

Durham E-Theses

Thermal conductivity studies of $YBa(2)Cu(3)O(7-\delta)$

Delap, Martin Richard

How to cite:

Delap, Martin Richard (1990) Thermal conductivity studies of $YBa(_2)Cu(_3)O(_7-\delta)$, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/9301/

Use policy

 $The full-text\ may\ be\ used\ and/or\ reproduced,\ and\ given\ to\ third\ parties\ in\ any\ format\ or\ medium,\ without\ prior\ permission\ or\ charge,\ for\ personal\ research\ or\ study,\ educational,\ or\ not-for-profit\ purposes\ provided\ that:$

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders. Please consult the full Durham E-Theses policy for further details.

> Academic Support Office, The Palatine Centre, Durham University, Stockton Road, Durham, DH1 3LE e-mail: e-theses.admin@durham.ac.uk Tel: +44 0191 334 6107 http://etheses.dur.ac.uk

The copyright of this thesis rests with the author. No quotation from it should be published without his prior written consent and information derived from it should be acknowledged.

Thermal Conductivity Studies of YBa₂Cu₃O₇₋₆

by

Martin Richard Delap

A Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Department of Physics

The University of Durham 1990



2 8 AUG 1991



To my Parents

.

· . `:

*

•

.



• • •

17114

and the second

· . · :

.

. :

. .

.

. To take

÷

1 4 1

÷ .

:



Yasıdı. s.

6 8

Abstract

. .

. 21 .

11 1

The stand is a Apparatus to measure the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ at temperatures between 20K and 120K has been designed and constructed. The thermal conductivity is measured using a longitudinal steady state heat flow tech-nique. Thermal conductivity measurements have been performed upon a sample of YBa₂Cu₃O_{7- δ} which has been subjected to a series of heat treatments in order to remove oxygen from the material. The measurements show conclusively that the thermal conductivity of YBa₂Cu₃O_{7- δ} is very strongly influenced by the oxygen content of the material. A reduction of the oxygen content of the material results in a substantial lowering of the thermal conductivity. To explain this result, a quantitative model has been constructed. The model demonstrates that consideration of the changes in phonon interactions alone cannot account for the differences in the behaviour of the thermal conductivity of $YBa_2Cu_3O_6$ and $YBa_2Cu_3O_7$. In addition; the model shows that there must be a significant carrier contribution to the thermal conductivity in both the normal and superconducting states. A physical process has been proposed which provides the required large carrier contribution below T_c. Further studies have been performed on a series of samples of YBa₂Cu₃O_{7- δ} which were sintered at slightly different temperatures. Qualitative $\frac{1}{2}$ analysis of the physical properties of these samples has been performed.

the physical properties of these samples and the second second second second second second second second second 112hi y trans s ser alla suer l'all'as robers all'arteristic an an tara ta

Acknowledgements

I am grateful to a large number of people, within the field of Physics and beyond, who have supported me during the course of this work. I would like to thank all of those who have educated me up to this time. Many members of the Durham Physics department have contributed to making this study enjoyable. I am particularly indebted to the following whom I would like to take this oppportunity to thank

Professor A.D.Martin, Professor A.W.Wolfendale for the provision of the facilities of the Physics department. Dr W.D.Corner and Professor B.K.Tanner for enabling me to use the facilities of the Solid State Physics Research Group.

Dr. N.R.Bernhoeft, who supervised this work. I am particularly appreciative of his advice and encouragement. I hope that his infectious enthusiasm for Physics will remain with me.

Cookson Group Plc who have provided the financial support for this work. Dr A.J.Bell, Dr T.P.Beales and Mr J.S.Rogers from Cookson Group Plc for their guidance and assistance, especially with the work contained in chapter VII.

Dr D.K.Thomas and Mrs P.Bransden from the University Industrial Research Laboratories who have overseen and run the Cookson contract.

Mr P.J.Allen, Mr S.Cockerton, Miss S.M.Thompson and Mr P.Warren for their endless cheerful support and advice, and for teaching me in the ways of the VSM and the university computer system. Mr C.I.Gregory for his assistance in obtaining the magnetisation curves.

Dr D.Evans and Dr D.P.Hampshire for useful discussions.

Mr P.Foley and Mr J.Scott for their assistance and technical support in the dayto-day running of the equipment.

Mr D.Jobling and his staff in the Student Workshop, Mr P.Armstrong, Mr G.Teasdale and Mr M.Greener for their hard work and formidable ability at turning my brief sketches into precise pieces of equipment.

Mr T.Jackson and the staff from the Electronics Workshop, Mr C.Mullaney, Mr D.Stockdale, Mr T.Adamson and Mr C.Moore for their seemingly endless assistance, and for producing the sample mounting boards for the resistance measurements.

Mrs P.Russell in the Drawing Office for her advice with diagram design, for the enthusiasm and skill with which she labelled all of the diagrams and for drawing the more complicated diagrams.

Mr M.Lee and Miss V.Greener in the Audio Visual Workshop for the production of the optical micrographs, binding the soft copies of this thesis and, along with Mrs P.Russell, teaching me the finer points of using the photocopier.

Miss J.Outson, Mrs N.Bingham, Mrs C.Webster for typing various letters, helping with all matters secretarial, and placing all manner of orders for stationery and equipment.

Mr R.Hardy from the Geology Department at Durham for his enthusiastic help with the X-ray powder diffraction.

I greatly appreciate the support given to me by my brothers, Stephen and Graeme.

Thanks to Miss J.D.Guilliard for her continual encouragement and advice, I cannot express how much her support has meant.

Finally, I am especially indebted to my parents for all their support, morally and financially - Thank you.

Publications

Thermal conductivity studies of $YBa_2Cu_3O_{7-\delta}$. M.R.Delap, N.R.Bernhoeft, T.P.Beales, J.Rogers. Accepted for publication in Chemtronics.

Declaration

I declare that the work contained in this thesis has not been submitted for a degree at this University or any other. All work presented was conducted by the author unless stated otherwise.

Copyright © 1990 by Martin Richard Delap

)

)

¢

)

The copyright of this thesis rests with the author. No quotation from it should be published without the author's prior written consent. Information derived from this thesis should be acknowledged.

Contents

)

)

)

)

)

1	Super	rconductivity
	1.1	Historical
	1.2	Basic Phenomena
		1.2.1 Persistent Currents
		1.2.2 Meissner-Ochsenfeld Effect
		1.2.3 Critical Magnetic Field
		1.2.4 Critical Current Density
	1.3	Magnetic Penetration Depth
	1.4	Coherence Length
	1.5	Type I and Type II Superconductors
	1.6	Energy Gap
	1.7	Isotope Effect
	1.8	Fluctuation Effects
	1.9	Practical Superconductivity
		1.9.1 Depairing Critical Current Density
		1.9.2 Flux Flow Resistivity
		1.9.3 Granular Superconductors
	1.10	Thermal Conductivity of Superconductors
2	High	Temperature Superconductivity
	2.1	Introduction
	2.2	Range of High T_c Materials
		2.2.1 La-Sr-Cu-O
		2.2.2 Bi-Sr-Ca-Cu-O
		2.2.3 Tl-Ca-Ba-Cu-O
	2.3	Y-Ba-Cu-O
		2.3.1 Structure
		2.3.2 Oxygen Content
		2.3.3 Twinning
		2.3.4 Preparation of Single Crystal $YBa_2Cu_3O_{7-\delta}$
	2.4	Preparation of Polycrystalline $YBa_2Cu_3O_{7-\delta}$
	2.5	Physical Properties of $YBa_2Cu_3O_{7-\delta}$

.

		2.5.1	Isotope Effect
		2.5.2	Anisotropy
		2.5.3	B_{c2} and the Coherence Length $\ldots \ldots \ldots \ldots \ldots 29$
		2.5.4	B_{c1} and the Magnetic Penetration Depth
		2.5.5	Fluctuation Effects
	2.6	Thern	nal Conductivity of $YBa_2Cu_3O_{7-\delta}$
		2.6.1	Porosity
		2.6.2	Thermal Conductivity Maximum
		2.6.3	Interpretation of Data
3	Mate	rial Cl	haracterisation
-	3.1	X-rav	Powder Diffraction
		3.1.1	Outline
		3.1.2	Experimental Details
	3.2	Electr	ical Resistivity
		3.2.1	Temperature Dependence
		3.2.2	Application of a Magnetic Field
		3.2.2	Experimental Details
	3.3	Induct	tance Probe
		3.3.1	Outline
		3.3.2	Experimental Details
	3.4	Magne	etisation Curves
		3.4.1	Inter-grain Currents
		3.4.2	Intra-grain Currents
		3.4.3	Bean Critical State Model
		3.4.4	Vibrating Sample Magnetometer
		3.4.5	Experimental Details
4	Ther	nal Co	anductivity Experiments
Ŧ	4 1	Rasic	Measurement Methods 58
	T 1 T	A 1 1	Non-steady State Methods 59
		412	Steady State Methods 59
			4 1 2 1 Longitudinal Heat Flow 59
			4.1.2.2 Badial Heat Flow 60
	49	Design	Considerations 60
	3.4	Design	

.

.

		4.2.1 Temperature Stability
		4.2.2 Sample Heater
		4.2.3 Residual Gas Conduction
		4.2.4 Radiation Losses
		4.2.5 Measurement of Temperature Difference
		4.2.6 Thermocouple Wires
	4.3	Experimental Arrangement
		4.3.1 Cryostat
		4.3.2 Sample Mounting
		4.3.3 Constant Current Power Supply 69
		4.3.4 Data Acquisition System
	4.4	Test Data
5	Therr	nal Conductivity Study I – Effect of Oxygen Content
	5.1	Introduction
	5.2	Sample Preparation
	5.3	Preliminary Characterisation
		5.3.1 X-ray Powder Diffraction
		5.3.2 Electrical Resistivity Measurements
		5.3.3 Inductance Measurements
		5.3.4 Oxygen Content
	5.4	Thermal Conductivity Measurements
	5.5	Quantitative Model
		5.5.1 Formulation
		5.5.2 Obtaining the Phonon Density of States
		5.5.3 Umklapp Processes
		5.5.4 Effect of Different Phonon Density of States 91
		5.5.5 Phonon-carrier Scattering
		5.5.6 Point Defect Scattering
	5.6	Boundary Scattering
	5.7	Electronic Contribution
		5.7.1 Wiedemann-Franz Law
		5.7.2 Model for Electron Scattering
	5.8	Conclusions

)

)

)

O	1 der	mai Conductivity Study II - Enect of Sintering
	Temp	erature
	6.1	Sample Preparation
	6.2	X-ray Powder Diffraction
	6.3	Electrical Resistivity Measurements
	6.4	Magnetisation Measurements
	6.5	Optical Microscopy
	6.6	Thermal Conductivity Measurements
	6.7	Model for Connectivity 123
	6.8	Conclusions
7	Ther	mal Conductivity Study III – Cookson Group Materials
	7.1	Introduction
	7.2	Material Preparation
	7.3	Electrical Resistance Measurements
	7.4	Magnetisation Measurements
	7.5	Thermal Conductivity Measurements
	7.6	Conclusions
8	Conc	lusions and Suggestions for Further Work
	8.1	Summary 138
	8.2	Future Work
		8.2.1 Models
		8.2.2 Experiments
		8.2.3 Improvements to the Experimental Apparatus 140
	8.3	Conclusion
	Bibli	ography
A	The	Boltzmann Equation 152
в	Itera	tive Solution of Matrix Equations
С	Com	puter Program to Model Thermal Conductivity 160
ח	Com	nuter Program to Model Connectivity
J	СОШ	Pure
E	Com	puter Program for Data Acquisition

)

)

J

)

)

ī.

.

Chapter I

Superconductivity

One should always be a little improbable — Oscar Wilde. Phrases and Philosophies for the use of the Young

1.1 Historical

Unlike many other physical properties of materials, the phenomenon of superconductivity was not observed until the early part of the 20^{th} century. In 1908, Kammerling Onnes successfully overcame immense experimental difficulties and managed to liquefy helium. Three years later Onnes discovered that mercury exhibited a dramatic fall in electrical resistance just above the boiling point of liquid helium, such that the resistivity at 4.2K was below the limit of his measuring apparatus (figure 1.1). [Onnes (1911)¹] Since those early days thousands of materials, ranging from pure elements to alloys and oxides, have been found to undergo a similar phase transition at sufficiently low temperatures. Table 1.1 shows the transition temperatures of the known superconducting elements and some of the highest critical temperature superconducting materials discovered before 1986.

¹ Accounts of the early superconductivity work can be found in Gorter (1964) and Mendelssohn (1964).

Figure 1.1 — First observation of superconductivity.

Onnes' (1911) plot of resistance of a mercury specimen as a function of temperature. Reproduced from Kittel (1976).

)

)

)

)

)

)



Table 1.1 — Superconducting elements and compounds.

Compiled by Allen (1991). Original data sources Matthias et al (1963), Savitskii et al (1981) and Philips (1989).

Elements				
Element	$T_c(K)$	Element	$T_c(K)$	
Al	1.14	Re	1.40	
Cd	0.56	Ru	0.51	
Ga	1.09	Sn	3.72	
Hf	0.12	Ta	4.48	
Hg	4.15	Tc	7.77	
In	3.40	Th	1.36	
Ir	0.14	Ti	0.39	
La	6.00	Tl	2.39	
Lu	0.10	U	0.20	
Мо	0.92	v	5.38	
Nb	9.50	w	0.48	
Os	0.66	Zn	0.88	
Pa	1.40	Zr	0.54	
Pb	7.49			

Compounds		
Formula	$T_c(K)$	
Nb3Ge	23.2	
Nb3Ga	21.0	
Nb ₃ Sn	18.1	
NbN	17.2	
V ₃ Si	17.2	
V3Ga	16.8	
Ta3Pb	16.0	
Mo ₆ Pb _{0.9} S _{7.5}	15.2	
Mo3Re	15.0	
MoC	14.3	
LuRh ₄ B ₄	11.7	
La ₃ In	10.4	
ZrV ₂	8.8	
LaSn ₃	6.55	

1.2 Basic Phenomena

1.2.1 Persistent Currents

The complete disappearance of (dc) electrical resistivity is most elegantly demonstrated by the observation of a persistent current. In such an experiment an electrical current is applied to a ring of superconducting material, the current source is removed and due to the absence of a decay process the current will continue to flow indefinitely. File and Mills (1963) established a lower bound of 100,000 years on the decay time of a supercurrent using a nuclear magnetic resonance technique. Figure 1.2 shows typical data obtained from such experiments.

1.2.2 Meissner-Ochsenfeld Effect

Some years after Onnes' discovery, Meissner and Ochsenfeld (1933) observed that a specimen of superconducting material excludes magnetic flux when it is placed in a magnetic field and cooled below its transition temperature. This effect is not predicted simply by perfect conductivity. In a perfect conductor, current would flow in the material so as to oppose the existing magnetic flux in the material, the flux would be 'trapped' in the material when the field was removed. In the Meissner-Ochsenfeld effect the magnetic flux is actually excluded from the sample.

1.2.3 Critical Magnetic Field

The magnetic field applied to a superconductor is opposed by screening currents which are set up to cancel the magnetic field inside the material. If the applied field is sufficiently large a state will be reached where it is energetically favourable for the material to return to the normal state in preference to increasing the screening currents. The field at which this occurs is known as the thermodynamic critical magnetic field, $B_c(T)$, and may be determined by equating the difference in free energy between the normal and superconducting states with the magnetic energy per unit volume.

$$\frac{B_c^2(T)}{2\mu_0} = f_n(T) - f_s(T) \tag{1.1}$$

4

Figure 1.2 — Decay of persistent currents.

File and Mills' (1963) data showing the decay of a persistent current determined by a nuclear magnetic resonance technique.



1.2.4 Critical Current Density

Maxwell's equations show that a magnetic field is associated with an electrical current. Increasing the electrical current applied to a superconductor eventually results in the magnetic field associated with the current exceeding the critical magnetic field. At this point the material will return to its normal state. The current density at which this occurs is known as the critical current density $J_c(T)$, and the phenomenon known as the Silsbee effect. [Silsbee (1916)]

1.3 Magnetic Penetration Depth

The screening currents, which act to expel magnetic flux, flow within the surface region of the sample to a depth which may be characterised by a length known as the magnetic penetration depth, λ . Hence a real superconductor does not exhibit flux expulsion from the entire sample. Schoenberg (1952) has shown experimentally that in very thin films, of thickness less than the penetration depth, there is appreciable deviation from perfect diamagnetism.

London and London (1935) postulated that the superconducting state can be described by some simple additions to Maxwell's equations.

$$\frac{\partial \mathbf{J}}{\partial t} = \frac{\mathbf{E}}{\mu_0 \lambda^2}$$
 and $\nabla \times \mathbf{J} = \frac{-\mathbf{B}}{\mu_0 \lambda^2}$ (1.2)

Simple algebraic manipulation of these equations leads to the expression for the penetration of magnetic field inside a superconductor

$$\nabla^2 \mathbf{B} = \frac{\mathbf{B}}{\lambda^2} \tag{1.3}$$

which, for **B** parallel to the x axis, has the solution

$$B = Be^{-x/\lambda} \quad \text{if } B \neq 0 \tag{1.4}$$

These postulates indicate that the magnetic field inside a superconductor falls off exponentially from the surface, and provide a simple mathematical description of the effect, although there is perhaps little physical insight which can be gained from application of the equations.

1.4 Coherence Length

In 1950 Ginzburg and Landau produced a phenomenological theory of superconductivity, the importance of which was not initially widely appreciated. [Ginzburg and Landau (1950), Ginzburg (1955)] They introduced the idea of a pseudo wave function, Ψ , which was chosen as an order parameter for the superconducting electrons such that $|\Psi|^2$ gave the value of the local number density of 'super' electrons, n_s . Ψ must then satisfy the so-called Ginzburg-Landau equation

$$\frac{1}{2m^*}(-i\hbar\nabla - e\mathbf{A})^2\Psi + \beta|\Psi|^2\Psi = -\alpha(T)\Psi$$
(1.5)

This equation is very similar in form to the Schrödinger equation, with the addition of a non-linear term in Ψ^3 . The Ginzburg-Landau theory leads to the introduction of a length scale which is a characteristic of the spatial distribution over which Ψ can vary without an 'undue' increase in energy, the Ginzburg-Landau coherence length ξ_{GL} .

$$\xi_{GL}(T) = \frac{\hbar}{[2m^*\alpha(T)]^{1/2}}$$
(1.6)

Near the superconducting transition the solution of the Ginzburg-Landau equation predicts the same temperature dependence for the magnetic penetration depth and the coherence length, thus it is commonplace to characterise superconducting materials by their temperature independent Ginzburg-Landau parameter

$$\kappa_{GL} = \frac{\lambda_{GL}}{\xi_{GL}} \tag{1.7}$$

The idea of a coherence length was proposed, independently of the Ginzburg-Landau theory, by Pippard (1953). The Pippard coherence length, ξ_0 , arises from an appeal to basic physical insight. For a material which undergoes some phenomenon at an onset temperature T_c , only electrons within an energy kT_c of the Fermi energy can play a major role. These electrons will have a typical momentum of $\Delta p \simeq kT_c/v_f$ where v_f is the Fermi velocity. From the uncertainty principle $\Delta x \simeq \hbar v_f/kT_c$. In a pure superconductor well below T_c , $\xi_{GL} = \xi_0$ whereas near T_c , the term $\alpha(T)$ in equation (1.6) tends towards the value of $(T - T_c)$ hence $\xi(T)$ diverges as $1/(T - T_c)^{\frac{1}{2}}$.

1.5 Type I and Type II Superconductors

In fields below B_c , the existence of reversible, perfect diamagnetic behaviour indicates that the material is in a thermodynamically stable state of lowest free energy that can be reached by lowering the temperature and applying a magnetic field. As the applied field is increased it may become energetically favourable to allow magnetic flux to penetrate the material along an ordered lattice of lines of normal material. Such a state where normal and superconducting regions exist together is known as the mixed state and is a characteristic of a type II superconductor. Superconducting materials which do not form this state are classed as type I superconductors.

Type II superconductors have two characteristic magnetic fields, a lower critical magnetic field, $B_{c1}(T)$, up to which a full Meissner-Ochsenfeld effect is seen, and a second critical magnetic field, $B_{c2}(T)$, where the number of regions of normal material has increased to the extent that all of the material is in the normal state. A comparison of the magnetisation curves for 'ideal' type I and type II superconductors is given in figure 1.3. Abrikosov (1957) showed that the Ginzburg-Landau parameter could be used to determine whether a superconducting material was type I or type II; materials with $\kappa_{GL} < 1/\sqrt{2}$ are type I, whereas those with $\kappa_{GL} > 1/\sqrt{2}$ are type II. A further result of Abrikosov's work was that in the mixed state, such that $B_{c1}(T) < B < B_{c2}(T)$, the magnetic flux is quantised and penetrates the material in a regular array of flux tubes. Each flux tube carries a flux quantum

$$\Phi_0 = \frac{h}{2e} \tag{1.8}$$

Around each flux tube the field decays in an exponential manner as described by the London equation. Magnetic decoration techniques have been used to confirm Abrikosov's prediction of a regular array of flux tubes (figure 1.4).

1.6 Energy Gap

The transition from the normal to superconducting state involves no change in the latent heat of the system, consequently the phase transition is of second order. However, at the superconducting transition there is a discontinuity in the specific heat capacity of the system which is indicative of the superconducting state





Figure 1.4 — Decoration of the magnetic flux line lattice.

1

1

.

,

1

The flux lines form an ordered triangular array. Reproduced from Träuble and Essmann (1968).



having a lower entropy than the normal state. A detailed analysis of the transition shows that the electronic contribution to the specific heat capacity is altered in a manner which is characteristic of a system with excited energy levels separated from the ground state by an energy gap, conventionally denoted 2Δ . The BCS theory of superconductivity proposed by Bardeen, Cooper and Schrieffer (1957) suggests that in the superconducting state the electrons experience an attractive potential and form a paired ground state. Within this framework the electron pair has a ground state energy lower, by an energy 2Δ , than that of two single electron ground states, thus the energy gap is indicative of the strength of the electron pairing.

1.7 Isotope Effect

BCS theory suggests that the electrons in the superconducting state form a paired ground state, the pairing arising from some attractive interaction. If this pairing is caused by an electron-phonon interaction the superconducting transition temperature of the material should be dependent upon the isotopic mass of the atoms comprising the crystal lattice according to the relationship

$$M^{\alpha}T_{c} = \text{constant} \tag{1.9}$$

For conventional 'low temperature' superconductors such as lead, niobium and mercury, α has the value $\frac{1}{2}$ to within experimental error. [Meservy and Schwartz (1969)] Elements such as zirconium, molybdenum and osmium have a value of α in the range 0.00 to 0.39. It is widely thought [Tinkham and Lobb (1989)] that such departures from $\alpha = \frac{1}{2}$ show only that the simple BCS picture must be generalised to account for the electron-electron interactions as well as the electron-phonon terms.

1.8 Fluctuation Effects

Thermodynamic fluctuations in a material may lead to local breakdown of superconductivity which will cause the existence of an astronomically small, but finite, resistivity in a superconducting material below T_c . Conversely, it is also possible for the fluctuations to reveal superconducting behaviour in a sample at a temperature fractionally above T_c . [Glover (1967)] As the coherence length is a measure of the distance over which the superconducting order parameter may vary without loss of superconductivity it follows that in materials with a large coherence length, greater than a few interatomic distances, the electrons may interact coherently with each other and result in a sharp transition with little evidence of fluctuation effects. Conversely, materials with a short coherence length are expected to exhibit strong evidence for fluctuation effects.

1.9 Practical Superconductivity

1.9.1 Depairing Critical Current Density

The number density of electrons which carry a supercurrent in a conventional superconductor is constant for a given material at a fixed temperature. Increasing the current applied to a superconductor results in an increase in the electron velocity until the kinetic energy of the electrons is at least equal to the energy gap of the superconductor, at this point excitations may occur which destroy the superconductivity - effectively the superconducting pair is broken. The critical current density at which this phenomenon occurs is known as the depairing critical current density.

1.9.2 Flux Flow Resistivity

When a current is applied to a type II superconductor in the mixed state the flux lines experience a force $J \times \Phi$ which will tend to make them move. This flux motion will lead to an induced voltage (Lenz' law) which is in effect a resistive term. Any real material will contain inhomogeneities of many kinds e.g. twin boundaries, grain boundaries, point defects, chemical inhomogeneities. These regions will not be locally superconducting and therefore they may act as centres for flux pinning. In such a case, the flux will flow until it is pinned. As there is no longer flux motion, there will no longer be flux flow resistivity. A direct consequence of this flux pinning is that the superconductor will now be able to sustain a supercurrent up to a critical current density which can be one or more orders of magnitude greater than the defect-free material. In practical applications of superconductors, $J_c(T)$ may be limited by the onset of flux flow resistivity rather than the Silsbee effect or the depairing critical current density. [Das Gupta et al (1978)]

1.9.3 Granular Superconductors

A granular superconductor is a material which contains regions of superconducting material which are separated by some non-superconducting regions in such a manner that there is a weak coupling between the superconducting regions. These boundaries may take many forms, they may be insulating oxide layers or simple defects which are less than or of comparable size to the coherence length of the superconducting material. The superconducting transition of a granular system can be divided into three main regions. In the first region, the resistance of each of the grains of material will fall due to the onset of superconductivity within the grain at some temperature T_c^{onset} , however bulk superconductivity will not be observed as thermal fluctuations will result in loss of coherence of the superconducting wave function between grains. As the temperature is reduced further, a state is reached where the coupling between the grains will overcome the thermal fluctuation effects. Finally, at a sufficiently low temperature, a percolation path will be established and a zero resistance state attained. Thus there may be considerable broadening of the transition due entirely to weak coupling between grains. [Pellan et al (1972), Raboutou et al (1980)]

1.10 Thermal Conductivity of Superconductors

BCS theory affirms that when a material undergoes a phase transition into the superconducting state the electrons which condense into the paired state are unable to scatter phonons and transport entropy. As a consequence of this there is a reduction in the electronic contribution to the thermal conductivity as the material is cooled through T_c . If the dominant contribution to the transport of heat is from the electrons, there will be a significant reduction in the thermal conductivity of the material below T_c . This is the case for many conventional superconductors because the phonon contribution is usually low in the region of T_c (figure 1.5).

If phonon scattering is dominant, for example in a material with a high concentration of defects, there will be a reduction in the electron-phonon scattering resulting in a longer effective phonon mean free path, therefore there will be a rise in the thermal conductivity below T_c . Very few conventional superconductors exhibit this effect because the phonon contribution to the thermal conductivity is small at low temperatures. Some conventional superconductors which have very high defect concentrations do show this enhancement in the thermal conductivity below T_c , for example lead doped with 30% bismuth (figure 1.6).

)

)

)









Chapter II

High Temperature Superconductivity

'Curiouser and curiouser!' cried Alice — Lewis Carroll. Alice in Wonderland

2.1 Introduction

Since Onnes' discovery, progress in finding materials which have higher and higher T_c values has been extremely slow. Before 1986, the highest known value of T_c was around 23.2K for Nb₃Ge. Bednorz and Müller (1986) suggested that a lanthanum based compound has a superconducting transition temperature in excess of the highest T_c for any superconductor then known, this was the first of the so-called 'high temperature' superconductors and for this work they were awarded a Nobel Prize for Physics. Since that time there has been a vast increase in the number of materials which are known to exhibit bulk, 'high temperature' superconductivity, all of these materials are oxides. Table 2.1 compares the critical temperature of some of the higher T_c oxide superconductors known before 1986, with those of the new 'high temperature' oxide superconductors.

Following the discovery of 'high temperature' superconductivity an immense amount of research resource and effort has been devoted to the subject resulting in the publication of tens of thousands of papers. A great many review articles and books have also been written, [see for example Bednorz and Müller (1988), Ginsberg (1989), Philips (1989), Yvon and François (1989), Jorgensen et al (1990)] the reader is initially referred to these for details of material properties not given in

Tal	ble	2.1	- 1	Superc	ondu	cting	oxides.
-----	-----	-----	-----	--------	------	-------	---------

Compiled by Allen (1991). Original data sources: Matthias et al (1963), Savitskii et at (1981) and Philips (1989).

Low T_c oxides			
Formula	$T_c(K)$		
NbO	1.25		
SnO	3.81		
TiO	0.68		
TiReO	5.74		
Pd _{0.285} Zr _{0.61} O _{0.105}	2.09		
Ti _{0.573} Rh _{0.287} O _{0.14}	3.37		
Zr _{0.611} Rh _{0.285} O _{0.105}	11.8		
Zr ₃ V ₃ O	7.5		
$Mo_2PbS_6O_2$	11.7		
$Mo_2Cu_2S_6O_2$	9.0		
Tl _{0.3} WO ₃	2.14		
K _{0.33} WO ₃	6.3		
Na _{0.33} WO ₃	3.0		
Ba _{0.13} WO ₃	1.9		
SrTiO3	0.39		
Sr _{0.7} Ca _{0.3} TiO ₃	0.6		
(Ba _{0.1} Sr _{0.9})TiO ₃	0.6		
(Ca _{0.31} Sr _{0.69})TiO ₃	0.6		
Li ₂ TiO ₄	13.7		
$BaPb_{1-x}Bi_xO_{3-y}$	13.2		

High T_c oxides		
Formula	$T_c(K)$	
$(Ba_{1-x}K_x)BiO_3$	30	
La _{2-z} Ba _z CuO ₄	35	
$La_{2-x}Sr_xCuO_4$	38.5	
$La_{2-x}Ca_xCuO_4$	18	
(Nd,Ce,Sr)2CuO4	28	
$Nd_{2-x}Ce_{x}CuO_{4}$	24	
$YBa_2Cu_3O_{7-x}$	93	
YBa ₂ Cu ₄ O _{8-x}	80	
$Y_2Ba_4Cu_7O_{15-z}$	40	
$\operatorname{Bi}_2(\operatorname{Sr}_{2-x}\operatorname{Bi}_x)\operatorname{CuO}_6$	12	
Bi ₂ Sr ₂ CaCu ₂ O ₈	90	
(Bi,Pb) ₂ (Sr,Ca) ₄ Cu ₃ O ₈	110	
$Tl_2Ba_2CuO_6$	92	
$Tl_2Ba_2CaCu_2O_8$	110	
Tl2Ba2Ca2Cu3O10	125	

this chapter. An obvious consequence of the immense quantity of published material is that the articles cited in this chapter are not necessarily primary references and do not imply any priority of publication for a given result, although in a few cases such citations are possible and these are indicated in the text.

2.2 Range of High T_c Materials

2.2.1 La-Sr-Cu-O

Lanthanum strontium copper oxide was the first 'high temperature' superconductor to be discovered, its structure is identical to that of layered, undoped La₂CuO₄ which consists of a plane of copper atoms surrounded by 4 strongly bonded oxygen neighbours and a further 2 weakly bonded oxygen atoms sited above and below the copper atom. These layers are in turn intercalated by two La-O layers as shown in figure 2.1. Stoichiometric La₂CuO₄ is not itself superconducting, the material must be doped in order to produce a superconducting compound. Considerable work has been carried out by doping the system with a variety of elements, the highest T_c so far discovered for this system is near 39K for La_{2-x}Sr_xCuO_y with x = 0.15.

2.2.2 Bi-Sr-Ca-Cu-O

The Bi-Sr-Ca-Cu-O class of materials were discovered by Maeda et al (1988) and Michel et al (1987) after the discovery of the La-Sr-Cu-O and Y-Ba-Cu-O systems. The Bi-Sr-Ca-Cu-O system is a family of superconducting materials which follow the general formula $\operatorname{Bi}_m\operatorname{Ca}_{n-1}\operatorname{Sr}_2\operatorname{Cu}_n\operatorname{O}_x$, each member of the series consists of n Cu-O₂ layers separated by Ca layers containing no oxygen, these layers are intercalated by two Sr-O layers and m Bi-O layers. The structure determination of each of the materials was complicated by the difficulty in producing monophasic samples, caused in part by the large, layered, unit cell (a stacking fault along the caxis may give rise to a second phase) and the fact that the material is quinternary. Although some members of this class of materials are non-superconducting, the transition temperature of the superconducting phases varies from 12K for the n = 1m = 2 phase up to 110K for the n = 3 m = 2 phase.

Figure 2.1 — The structure of $La_{2-z}Sr_zCuO_y$ Reproduced from Schuller and Jorgensen (1989).



2.2.3 Tl-Ca-Ba-Cu-O

The Tl-Ca-Ba-Cu-O system of superconductors was discovered by Sheng and Hermann (1988) and is very similar to the Bi-Sr-Ca-Cu-O system. The generic formula is $Tl_mCa_{n-1}Ba_2Cu_nO_x$ and the materials again consist of n Cu-O layers separated by Ca layers which are in turn intercalated by two Ba-O layers and m Tl-O layers. The precise structures have again proved difficult to determine, for similar reasons to the bismuth based material. The highest transition temperature (125K) for any superconductor found to date belongs to this family for the n = 3 m = 2and n = 4 m = 1 phases. [Parkin et al (1988)] Figure 2.2 depicts schematically the structure of the thallium and bismuth based systems.

2.3 Y-Ba-Cu-O

2.3.1 Structure

Following the discovery of $La_{2-x}Sr_xCuO_y$, a new system of superconducting oxides was discovered by Wu et al (1987), that of RBa₂Cu₃O_{7- δ} where R is one of the lanthanide elements (except Ce,Pm,Tb). There are 12 known superconducting compounds which have this RBa₂Cu₃O_{7- δ} composition, [Maple et al (1987)] most of these are metallic in character above T_c and, bar lanthanum, have a superconducting transition temperature of between 90K and 93K (figure 2.3). All have the same layered structure which is closely related to the perovskite structure. [Le Page et al (1987)] A surprising result of investigations into the RBa₂Cu₃O_{7- δ} systems has revealed that a large number of the superconducting properties such as T_c, B_{c1}, B_{c2} and J_c are roughly independent of the rare earth element.

Various research workers have shown that the Y-Ba-Cu-O material is composed of two basic structures. [Khachaturyan et al (1988), Beno et al (1987), Schuller et al (1987), Van Tendeloo et al (1987), Greedan et al (1987)] The first is an orthorhombic structure based upon the stoichiometric composition $YBa_2Cu_3O_7$ (figure 2.4), whereas the second has a tetragonal structure based upon the composition $YBa_2Cu_3O_6$. The unit cell of orthorhombic $YBa_2Cu_3O_7$ consists of planes of $Cu-O_2$ separated by a Y layer and intercalated with two Ba-O planes and one plane containing Cu-O. The removal of oxygen from this plane results in the formation of the tetragonal semiconductor $YBa_2Cu_3O_6$. [Jorgensen et al (1990)]

Figure 2.2 — The structure of $Tl_m Ca_{n-1} Ba_2 Cu_n O_x$

The corresponding bismuth compound has the same structure with thallium replaced by bismuth. Reproduced from Ginsberg (1989).





TI

71

Cu Cu

Cu

Ca

Cu

B۵

Π

TI

n #]

n = 1

n = 1

Figure 2.3 — Electrical resistivity of RBa₂Cu₃O_{7- δ}

The electrical resistivity is normalised to its value at 120K (80K for R=La) for RBa₂Cu₃O_{7- δ} compounds with

(a) R = Y, La, Nd, Sm, Eu and Gd (b) R = Dy, Ho, Er, Tm, Yb and Lu. Reproduced from Maple et al (1987).



RBa2Cu3O

.



2.3.2 Oxygen Content

As the orthorhombic phase exhibits superconducting behaviour and the tetragonal phase exhibits semiconducting behaviour, the oxygen vacancy concentration is very important in the YBa₂Cu₃O_{7- δ} system. The lattice parameters of the unit cell depend upon the oxygen vacancy concentration, with the *c* axis parameter increasing as the oxygen concentration decreases (figure 2.5). Cava et al (1987) and Beyers et al (1987) have measured the effect of oxygen concentration upon the superconducting transition temperature and have shown that the $\delta = 0$ material has a T_c of \simeq 92K, which decreases by a small amount as δ increases to \simeq 0.2. As the oxygen concentration is decreased from $0.2 \leq \delta \leq 0.35$ there is a sharp drop in T_c followed by a plateau region with T_c \simeq 60K for 0.35 $< \delta <$ 0.45. This is illustrated in figure 2.6. T_c then falls rapidly, reaching 0K at $\delta \simeq$ 0.65.

The removal of oxygen from the material is dependent upon the temperature of the material and the partial pressure of oxygen above it. If the oxygen partial pressure is sufficiently high, the tetragonal phase transforms into the orthorhombic phase upon cooling. Jorgensen et al (1990) have shown that on lowering the partial pressure of oxygen from 1013mbar to 0.1mbar the temperature at which the orthorhombic to tetragonal transformation occurs is reduced from 740°C to 500°C. This has serious implications for production of materials for applications which require treatment by conventional semiconductor device processing. Tu et al (1989) have shown that the oxygen diffusion rate is extremely slow, $\simeq 0.035 \text{cm}^2 \text{sec}^{-1}$, hence it is difficult to ensure that the bulk material has a homogeneous oxygen content.

2.3.3 Twinning

The unit cell of the YBa₂Cu₃O_{7- δ} system has lattice parameters with the relationship $a \simeq b$ and $c \simeq 3a$. As the a = b symmetry is lost by the tetragonal to orthorhombic transition, some regions form a state where a > b, others into states where a < b, the change in lattice parameter in each case is only very small resulting in very little energy cost being involved in production of a twinned structure. This can have an appreciable influence upon the superconducting properties of a specimen of YBa₂Cu₃O_{7- δ}.


Figure 2.5 — Lattice parameters of YBa₂Cu₃O_{7- δ} as a function of oxygen content. Reproduced from Jorgensen et al (1990).



Figure 2.6 — T_c of YBa₂Cu₃O_{7- δ} as a function of oxygen content. Reproduced from Cava et al (1987).



2.3.4 Preparation of Single Crystal YBa₂Cu₃O₇₋₆

Single crystals of YBa₂Cu₃O_{7- δ} have been grown by a variety of methods [Kaiser et al (1987), Müller et al (1988), Takeya et al (1988), Taylor et al (1987), Wanklyn et al (1988)] although to date none of the methods have led to the production of large (10x10x10mm³) homogeneous single crystals. A popular method is that of crystal growth using a flux such as KCl or CuO. Crystals are produced by this method by heating the mixture of YBa₂Cu₃O_{7- δ} and the flux to a temperature above the melting point of YBa₂Cu₃O_{7- δ}. The melt is slowly cooled, at typically 3°C/hr, whereupon crystal growth takes place, aided by the flux. The production of crystals with the correct stoichiometry and with low impurity concentrations is dependent upon the initial stoichiometry, quantity and type of flux, and a certain amount of good fortune. Once cool, the crystals must be carefully cut out from the surrounding material.

Even crystals which have the correct $YBa_2Cu_3O_{7-\delta}$ stoichiometry and low impurity concentrations suffer greatly from the problems connected with the oxygen content. The as-grown crystals almost invariably do not have the $YBa_2Cu_3O_7$ composition and require an additional oxygen treatment. Owing to the low oxygen diffusion rate, large crystals of close to theoretical density require an impractically long time for oxygen to diffuse throughout the sample. This problem may be partially overcome by using crystals which have one dimension which is very small, of the order of a few μ m. [Salama et al (1989)] Whilst this is satisfactory for some measurements, for a great many experiments, larger crystals are required.

2.4 Preparation of Polycrystalline $YBa_2Cu_3O_{7-\delta}$

Polycrystalline YBa₂Cu₃O_{7- δ} may be produced by a variety of routes, ranging from precipitation techniques to solid state reaction of precursor compounds of yttrium, barium and copper. As YBa₂Cu₃O_{7- δ} is a quaternary material the phase diagram is complex, this exacerbates the problem of achieving the correct stoichiometric ratio of the elements throughout the entire material; local regions which contain too little copper result in formation of a tetragonal, green-coloured, semiconducting phase Y₂BaCuO₅; if a region is barium rich or yttrium deficient then phases such as BaCuO₂ or Ba₂CuO₃ may form. A close check upon the stoichiometry of the material is highly desirable. One of the most straightforward preparation routes is that of solid state reaction of precursor compounds of the metallic elements. These are often yttrium oxide Y₂O₃, barium carbonate BaCO₃ and copper (II) oxide CuO. The precursor materials are ground together to produce a homogeneous mixture, loaded into a crucible (typically alumina) and calcined. In order to minimise the contamination from the crucible material, the crucible may be filled with some existing YBa₂Cu₃O_{7- δ} and heated to the melting point of the YBa₂Cu₃O_{7- δ}, thereby leaving a strongly bonded coating of YBa₂Cu₃O_{7- δ} on the crucible surface. The calcination process is typically performed for 24 hours at a temperature above 900°C but less than 1010°C at which point any YBa₂Cu₃O_{7- δ} formed will start to undergo incongruent melting. This temperature ensures that the compounds fully react together and that the BaCO₃ is fully decomposed to BaO and CO₂.

After calcination the material is powdered and pressed into the desired shape, at this stage it consists of very poorly connected crystallites. In order to improve the links between the individual crystallites, the compacted material must undergo a sintering process. The sintering takes place at a high temperature. At approximately 920°C any BaCuO₂ present will melt, this could result in the grains being coated with BaCuO₂ which will inhibit the coherence of the superconducting order parameter between grains, as described in §1.9. Once the sintering process is complete, the material will probably be of low oxygen content. The final processing stage is therefore to add oxygen to the system. This is achieved by heating the sintered material to approximately 450° C and passing flowing oxygen over the samples for between 24 and 72 hours.

2.5 Physical Properties of $YBa_2Cu_3O_{7-\delta}$

2.5.1 Isotope Effect

The possible existence of an isotope effect in YBa₂Cu₃O_{7- δ} has been investigated by Batlogg et al (1987) and Bourne et al (1987). Such experiments, in common with all others made upon the YBa₂Cu₃O_{7- δ} system, are critically dependent upon the preparation route used for the samples as care must be taken to ensure that the results are not masked simply by differences in oxygen concentration. The samples used by Batlogg et al and Bourne et al were produced by diffusively replacing the ¹⁶O with ¹⁸O. Their data suggests that no change in T_c is observed, and that $\alpha = 0 \pm 0.02$. The results of Leary et al (1987) and Morris et al (1988) disagree slightly with this result, suggesting α has a small positive value. From these measurements it follows that if there is a phonon mediated interaction, it is dominated by an electron or other non-phonon interaction and as stated in §1.7 it does not necessarily mean that the idea of a BCS type paired electron state is not applicable in YBa₂Cu₃O_{7- δ}.

2.5.2 Anisotropy

The layered structure of YBa₂Cu₃O_{7- δ} leads to the material exhibiting highly anisotropic behaviour, the *a* and *b* axis properties are similar due to the orthorhombic structure of the unit cell; the *c* axis, being perpendicular to the Cu-O₂ planes has substantially different properties. Anisotropy in the electrical resistivity, ρ , has been observed by a variety of groups (figure 2.7). [Tozer et al (1987), Konczykowski et al (1988), Vinnikov et al (1988)] The resistivity along the *a* and *b* axes shows metallic behaviour down to T_c with a magnitude of approximately 0.20m Ω cm which is similar to the electrical resistivity obtained in good polycrystalline samples. Along the *c* axis, the resistivity is some 2 orders of magnitude higher than ρ_{ab} and shows evidence of semiconducting behaviour - the resistivity increases as the temperature falls, in a manner rather akin to the resistivity behaviour of polycrystalline samples with a poor oxygen content.

2.5.3 B_{c2} and the Coherence Length

With the determination of any physical quantity it is important to be clear which property of the material is being probed by the experiment and how that probe may affect the result. In §1.9 it was stated that the critical current density may be determined in a variety of ways- each of which involves a slightly different definition of J_c ; the point at which the quasi-particles are depaired; the onset of flux-flow resistivity and the current at which the self-field exceeds the critical magnetic field. Similarly there are a variety of *practical* definitions of $B_{c2}(T)$ each of which may be obtained by experimental observation.

The value of $B_{c2}(T)$ may be evaluated directly by applying a small probing current to a sample and detecting the onset of a voltage drop across the maFigure 2.7 — Electrical resistivity of single crystal YBa₂Cu₃O_{7- δ}. The two $\rho_{ab}(T)$ curves are from different crystals. The dashed curve through the ρ_c data is a fit to A/T + BT. Reproduced from Penney et al (1988).



terial. In this case the resistance and voltage criterion used will substantially affect the results. This particular problem is compounded further by noting that $YBa_2Cu_3O_{7-\delta}$ is a granular superconductor which results in a broadening of the transition as magnetic field is applied to the sample. B_{c2} may also be defined as the point at which the magnetisation curve exhibits reversible behaviour, corresponding to the onset of flux-flow resistivity, although it should be noted that at this point the material may still exhibit a strong diamagnetic signal. Magnetisation methods again suffer from problem of the broadening of the transition of granular superconductors. For materials with a high T_c , flux creep caused by thermal depinning may also affect the results.

A variety of research workers have determined B_{c2} for YBa₂Cu₃O_{7- δ} using a range of measurement techniques upon polycrystalline and single crystal specimens. Table 2.2 shows the values of dB_{c2}/dT near T_c. The value of $B_{c2}(0)$ may be obtained from this data by extrapolation of dB_{c2}/dT near T_c using the WHH formula in the 'dirty' ($\lambda/\xi \gg 1$) limit with no paramagnetic limiting. [Wertharmer et al (1966), Hake (1967), Collings (1986)]

$$B_{c2}(0) = -0.68T_c \frac{dB_{c2}}{dT}$$
(2.1)

The values of $B_{c2}(0)$ obtained are distributed over a large range of values (table 2.3). The data indicates that $B_{c2}(0)$ is extremely large and shows a marked anisotropy consistent with the layered structure of the material. The physical meaning of the higher values obtained is not at all clear owing to the omission of the effect of the paramagnetic susceptibility which would be significant at very high fields.

Representative values for the coherence length may be determined from $B_{c2}(0)$ through the mathematical relationship

$$B_{c2} = \frac{\Phi_0}{2\pi\xi^2}$$
(2.2)

The high values of $B_{c2}(0)$ lead to the result that the coherence length in the YBa₂Cu₃O_{7- δ} system is particularly small, ~30Å perpendicular to c and ~5Å parallel to the c axis. These values are comparable to the magnitude of the lattice parameters hence twin boundaries, dislocations or small impurities may substantially

and the second se				
Group	Method	$B \perp_c$	$B\ _{c}$	Bav
Ayache (1987)	Magnetic			2
Gallagher (1987)	Magnetic	3.8	0.54	
Iye (1987)	Resistance	3	0.9	
Muto (1990)	Resistance			2.4
Takita (1987)	Resistance	3.8	2.6	
Takabatake (1987)	Resistance			1.27
Welp (1989)	Magnetic	10	1.8	

Table 2.2 — $-dB_{c2}/dT$ at T_c in Tesla/Kelvin.

Table 2.3 — $B_{c2}(0)$ in Tesla.

Group	Method	B⊥ _c	$\mathbf{B}\ _{c}$	Bav
Ayache (1987)	Magnetic			120
Gallagher (1987)	Magnetic	240	34	
Iye (1987)	Resistance	190	56	
Muto (1990)	Resistance			150
Nakao (1988)	Magnetic	110 ± 10	40 ± 5	
Takabatake (1987)	Resistance			80
Takita (1987)	Resistance	240	160	
Welp (1989)	Magnetic	620	110	

affect the bulk superconducting properties of a given specimen of $YBa_2Cu_3O_{7-\delta}$. The coherence length may also be obtained from the observation of fluctuation effects described in §1.8, typically by observation of electronic contribution to the specific heat capacity. The value of $\xi(0)$ estimated in this manner is comparable to the values obtained by magnetic methods. Typical values obtained for $\xi(0)$ are given in table 2.4.

Group	Method	$\xi \perp_c$	ξ c	ξαυ
Fruchter (1988)	Magnetic	25 ±5	6.4 ±2	
Gallagher (1988)	Magnetic	31	4.3	
Inderhees (1988)	Heat capacity			7 ± 5
Iye (1987)	Resistance	24	7.0	
Oh (1988)	Resistance	16 ± 2	2.2 ± 1	
Muto (1990)	Resistance			15
Umezawa (1988a)	Magnetic	31	4.8	
		22	6.7	

Table 2.4 — $\xi(0)$ in Å.

2.5.4 B_{c1} and the Magnetic Penetration Depth

A large number of measurements of B_{c1} for YBa₂Cu₃O_{7- δ} have been made on polycrystalline specimens and single crystals using a variety of techniques. B_{c1} may be assessed by analysing the initial magnetisation curve and detecting the first deviation from a linear region. This particular method, whilst in principle easy to perform, is particularly susceptible to the problem of determining an appropriate criterion for the point at which the curve deviates from linear. A second method of determining B_{c1} is by analysis of the difference between field-cooled and zero-fieldcooled magnetisation data. A precise measurement of B_{c1} may still be difficult to obtain, particularly in samples which are inhomogeneous. Fruchter et al (1988) have used the onset of irreversibility of torque magnetometry signals to define B_{c1} , although this method is prone to the same errors as other magnetic techniques.

Table 2.5 gives representative values of $B_{c1}(0)$ for YBa₂Cu₃O_{7- δ} which, like $B_{c2}(0)$, show a large spread of values. In contradistinction to the values of B_{c2} the values of B_{c1} are small, ~20mT perpendicular to the *c* axis and ~100mT parallel to the *c* axis, indicating that YBa₂Cu₃O_{7- δ} is an extreme type II superconductor.

Group	Method	$B \perp_c$	$\mathbf{B}\ _{c}$	Bav
Fruchter (1988)	Torque curve	24 ± 3	110 ± 10	
Gallagher (1987)	M deviation	< 50	500	
Krusin Elbaum (1989a)	Direct magnetic	18	53	
McGuire (1987)	M deviation	20	400	40
Sridar (1989)	Microwave	25		
Umezawa (1988b)	M deviation	8 ± 1	37.5 ± 0.5	
Yeshurun (1988)	M relaxation	25 ± 5	90 ± 10	

Table 2.5 — $B_{c1}(0)$ in milliTesla.

The value of the magnetic penetration depth may be calculated from the values of B_{c1} , B_{c2} and ξ using the relationship

$$B_{c1} = \frac{\Phi_0}{\lambda^2} \ln\left(\frac{\lambda}{\xi}\right) \tag{2.3}$$

From this relationship the magnetic penetration depth of $YBa_2Cu_3O_{7-\delta}$ is found to be of the order of a few thousand angstroms, significantly larger than the coherence length. Direct measurement of the magnetic penetration depth may be made by using the technique of muon spin relaxation. With this technique, the sample is bombarded by a beam of polarised muons. The muons decay by emission of a positron which is emitted preferentially along the polarisation axis. By measurement of the angle of emission of the positron, the local magnetic field may be determined. In practice, a statistical average is taken and the muon depolarisation rate is related to the penetration depth by a theoretical model; it is the validity of this model which determines the validity of the data. Unlike other magnetic measurements, muon spin relaxation has the advantage of probing the local magnetic field of the material, hence voids in the material do not influence the results. Table 2.6 indicates the range of values obtained for $\lambda(0)$.

2.5.5 Fluctuation Effects

A representative value for the coherence length in high T_c materials is about 10Å, it follows that the superconductivity in high T_c materials will be much more

Group	Method	$\lambda \perp_c$	$\lambda \parallel_c$	λ_{av}
Forgan (1990)	Muon spin relaxation	12000	2100	5020
Fruchter (1988)	Indirect magnetic	5000	680	
Gallagher (1987)	Indirect magnetic	1800	270	
Harshman (1987)	Muon spin relaxation			1400
Krusin-Elbaum (1989b)	Direct magnetic	4200	1400	
Monod (1988)	Indirect magnetic			1900
Srid ar (1989)	Microwave	1400		
Umezawa (1988b)	Indirect magnetic	8400	900	
Uemura (1988)	Muon spin relaxation			1656
Yeshurun (1988)	Indirect magnetic	4600	1000	

Table 2.6 — $\lambda(0)$ in Å.

sensitive to small scale structural or chemical imperfections than conventional superconductors which have coherence lengths in the range 10^2 Å to 10^4 Å. These effects have been observed by Lægreid et al (1989) when performing high resolution heat capacity experiments in the region of T_c. The effects have also been observed in the electrical conductivity of YBa₂Cu₃O_{7- δ} by Vidal et al (1988), Gordon et al (1989) and Lægreid et al (1987).

2.6 Thermal Conductivity of YBa₂Cu₃O₇₋₆

In sharp contrast to many other physical parameters the database of the thermal conductivity of YBa₂Cu₃O_{7- δ} is rather small. This is in part due to the relative complexity of the measurement process and partly the requirement for large samples of well defined geometry. In particular the number of published measurements made upon single crystals is minimal.

Figure 2.8 illustrates the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ relative to a variety of materials at low temperatures. The magnitude of the thermal conductivity in the normal state is comparable to that of a metal with a poor thermal conductivity such as stainless steel. This is roughly consistent with the picture pre-

Figure 2.8 — The thermal conductivity of a range of materials Reproduced from White (1979). YBa₂Cu₃O_{7- δ} data taken from Jeżowski (1989). Perspex data taken from Burgess and Greig (1975).



sented by the electrical resistivity of YBa₂Cu₃O_{7- δ} which exhibits strong metallic behaviour in its normal state temperature dependence but which has a magnitude which suggests that it is a poor metal.

Early thermal conductivity measurements made upon YBa₂Cu₃O_{7- δ} disagreed strongly about the absolute magnitude of the thermal conductivity in the normal state, just above T_c (figure 2.9). The data of Jeżowski et al (1987) placed the thermal conductivity at around 5Wm⁻¹K⁻¹ at 100K, with the thermal conductivity decreasing slightly as the temperature is increased. Uher et al (1987) obtained a thermal conductivity of 3.5Wm⁻¹K⁻¹ at 100K with the thermal conductivity increasing slightly with temperature. The data of Morelli et al (1987) revealed a significantly smaller thermal conductivity, 0.5Wm⁻¹K⁻¹ at 100K; remaining roughly constant with increasing temperature! This latter data was obtained upon samples with a high electrical resistivity, $10m\Omega cm$ at 100K, and it is therefore probable that the sample suffered from a high porosity or poor inter-grain coupling.

2.6.1 Porosity

The porosity of a sample influences the thermal conductivity in two ways. Firstly the voids in the material act as scattering centres, this enhances the defectcarrier and defect-phonon scattering which tend to lower the thermal conductivity. Secondly, in a material with a high porosity (and hence of low relative density) a large portion of the apparent sample volume will consist of voids which act as barriers to the heat flow. In this instance the thermal conductivity calculated will be smaller than the 'true' value as the cross sectional area of the sample used in the calculation will be too large. This effect will manifest itself at all temperatures and, assuming the volume expansivity is small, will reduce the measured thermal conductivity by a constant proportion throughout the entire temperature range. The data of Heremans et al (1988) places the thermal conductivity of YBa₂Cu₃O_{7- δ} again at approximately 0.5Wm⁻¹K⁻¹. However, the density of their sample was only 43% of the theoretical single crystal value, in contrast to the 80% of theoretical density of the samples of Jeżowski et al.

Figure 2.9 — Early thermal conductivity measurements of $YBa_2Cu_3O_{7-\delta}$

• ... Jeżowski (1987) • ... Uher (1987) • ... Bayot (1987)

× ... Morelli (1987) + ... Heremans (1988)



2.6.2 Thermal Conductivity Maximum

All of the thermal conductivity data obtained upon superconducting samples of YBa₂Cu₃O_{7- δ} exhibits an increase upon cooling through the superconducting phase transition.¹ This increase in thermal conductivity continues as the sample is cooled until the thermal conductivity reaches a maximum at some temperature, T_{max} , before falling away. From figure 2.9, it can be seen that there is no universal value for T_{max} , some materials having the peak at approximately 60K, others at approximately 40K, the precise value depending upon the interplay between the scattering mechanisms.

Jeżowski et al (1989) have measured the thermal conductivity of three separate samples of YBa₂Cu₃O_{7- δ}, each of different oxygen content. Their data suggests that the absolute magnitude of the thermal conductivity in the normal state decreases as the oxygen content of the samples lowered. Unambiguous interpretation of this data is not possible because the data was obtained upon separate samples; the samples may all have an appreciably different microstructure which could influence the results. This problem afflicts all but a small number of the thermal conductivity studies of YBa₂Cu₃O_{7- δ} to date.

2.6.3 Interpretation of Data

The peak in the thermal conductivity below T_c has been attributed, by a number of research groups [Fischer et al (1988), Uher (1989)] to a decrease in the phonon-carrier scattering below T_c . To understand this picture one needs to understand the various processes which contribute to the thermal conductivity. In a conventional picture there is a contribution to the thermal conductivity from the phonon-phonon interactions, phonon-carrier interactions and carrier-carrier interactions. For the purposes of this discussion phonon-defect scattering is considered as a separate interaction although it is a particular case of phonon-phonon scattering arising from the non-ideal properties of the material. Many research groups have used the Wiedemann-Franz law to estimate the contribution to the thermal

¹ There is one exception to this. Uher et al (1988) drastically increased the number of point defects in their samples by irradiating them with neutrons and found that the peak became 'washed out' at high neutron fluxes.

conductivity from the charge carriers. This treats the carriers as a system of independent particles and relates the electrical resistivity, ρ , to the carrier thermal conductivity, κ_c .

$$\kappa_c = \frac{L_0 T}{\rho} \tag{2.4}$$

 L_0 (1.11 × 10⁻⁸ W Ω K⁻²) is the Lorenz constant. From this relationship, they estimate that the value of κ_c is small and hence that the carriers themselves do not play a major role in the heat conduction mechanism in the normal state. If this is the case, the thermal conductivity will be dominated by the phonon transport mechanisms; the carrier-phonon, phonon-phonon and defect-phonon scattering.

Below T_c it is assumed that the charge carriers are in a condensed state. In this state they carry no entropy and are unable to scatter phonons unless the phonon energy is sufficient to overcome the condensation energy. Thus below the transition temperature the thermal conductivity of the material increases as more and more carriers condense into the superconducting state and the carrier-phonon scattering relaxation time increases. The reduction in carrier-phonon scattering is offset by a reduction in the ability of the phonons to carry heat as the temperature falls. The interplay between the enhancement of the thermal conductivity due to the reduction in carrier-phonon scattering and the reduction in the phonon density as the phonons are 'frozen out' leads to a peak in the thermal conductivity below T_c .

Very few thermal conductivity studies have concentrated upon one particular sample which has undergone a variety of treatments. Zavaritskiĭ et al (1989a, 1989b) have measured the influence of oxygen content upon the same superconducting sample after subjecting it to a number of heat treatments. They found that the normal state thermal conductivity of the sample decreased as the oxygen was removed from the sample, in addition the peak in the thermal conductivity disappeared. However when the sample was heated to 675° C for 24 hours, producing a sample of nominal composition YBa₂Cu₃O₆, the normal state thermal conductivity dramatically increased (figure 2.10).

The initial decrease in the thermal conductivity was attributed to an increase in the number of defects in the sample as the oxygen was removed, hence there was Figure 2.10 — Thermal conductivity as a function of oxygen content.

+ ... x = 7.00 o ... x = 6.70 o ... x = 6.53o ... x = 6.31 - ... x = 6.77 o ... x = 6.00Reproduced from Zavaritskiĭ et al (1989a).



Temperature (K)

a greater increase in the phonon-defect scattering resulting in a lower thermal conductivity. When YBa₂Cu₃O₆ was formed, the sample was assumed to have taken on a more ordered form and there is therefore a reduction in the phonon-defect scattering resulting in a higher thermal conductivity. The carrier contribution was assumed to be negligible. However, Zavaritskiĭ et al did not re-oxygenate their sample after forming YBa₂Cu₃O₆. It is possible that the high temperature vacuum annealing caused the sample to undergo an irreversible change in its microstructure which led to an increase in the thermal conductivity. This argument is supported by noting that the thermal conductivity of YBa₂Cu₃O₆, at low temperatures, is actually lower than that of YBa₂Cu₃O₇. As the YBa₂Cu₃O₇ material is also wellordered and at low temperatures is expected to have a low carrier contribution to the thermal conductivity (Cooper pairs carry no entropy) this situation should not arise. The conventional picture of the thermal conductivity of YBa₂Cu₃O_{7- δ} is therefore left with a considerable difficulty. Part of the work in this thesis attempts to address this issue.

Chapter III

Material Characterisation

Experience is the name everyone gives to their mistakes. — Oscar Wilde. 'Lady Windermere's Fan'

.

3.1 X-ray Powder Diffraction

3.1.1 Outline

One method of determining the crystallographic structure and lattice parameters of a crystalline material is the method of powder diffraction. With this method a collimated, monochromatic beam of X-rays, of wavelength λ , is incident upon a finely powdered specimen which has effectively a continuous range of crystallite directions. Peaks in the intensity of the scattered X-rays will occur at the angles which satisfy the Bragg condition.

$$n\lambda = 2d_{hkl}\sin\theta_{hkl} \tag{3.1}$$

The crystallographic structure of $YBa_2Cu_3O_{7-\delta}$ is now well known, consequently by comparing the observed peak positions with the expected peak positions, the peaks may be indexed according to their Miller indices. The lattice parameters of the material may then be calculated to a precision and accuracy which depends critically upon the angular resolution of the detector. An exact determination of the lattice parameters requires the use of much more sophisticated experiments; for example neutron high resolution powder diffraction. X-ray powder diffraction may also be used to provide some indication of the phase purity of a known material by comparison of the powder diffraction curve with the expected curves for the material and its potential impurity phases. Quantitative analysis of the data may theoretically be possible, although the process is complicated by the influence of the resolution function of the detector, multiple scattering, absorption, statistics and structure factors. In practice, quantitative analysis must be performed by some other method.

3.1.2 Experimental Details

X-ray powder diffraction was performed using the Philips X-ray diffractometer housed in the Geology department at Durham. The experimental arrangement of this instrument is simple in concept and is depicted schematically in figure 3.1. A collimated X-ray beam, produced from a copper target, is incident upon a rotating sample stage whilst a detector rotates around the sample stage and makes an angle of 2θ relative to the incident beam. The output from the detector is amplified and sent to a ratemeter which in turn sends a signal to a chart recorder. The entire process is automated with the detector sweeping through an angle of 1° per minute.

The main diffraction lines for YBa₂Cu₃O_{7- δ} and the impurity phases BaCO₃, Y₂O₃, CuO, Y₂BaCuO₅, Ba₂CuO₃ and BaCuO₂ have been calculated by Bell (1990). These values were used to identify the crystallographic reflections corresponding to each diffraction peak. From the observed reflections the lattice parameters of the YBa₂Cu₃O_{7- δ} phase were calculated using a least squares technique, [Press et al (1988)] the *c* axis being computed first from the *d* spacing of the (001) to (007) reflections.

3.2 Electrical Resistivity

3.2.1 Temperature Dependence

The electrical resistivity of a material as a function of temperature is a simple yet powerful characterisation tool. Metallic materials are considered to have electrons in (unfilled) conduction bands even at absolute zero; consequently they tend to have a low electrical resistivity which decreases linearly as the temperature falls. In contrast to this, semiconductors rely upon thermal excitation of charge carriers

Figure 3.1 — X-ray powder diffraction apparatus.



into the conduction band, resulting in an increase in the electrical resistivity as the temperature decreases.

In the previous chapter it was noted that $YBa_2Cu_3O_{7-\delta}$ with a low oxygen content (high value of δ) is a semiconductor. The temperature dependence of the electrical resistivity of randomly oriented, polycrystalline specimens of $YBa_2Cu_3O_{7-\delta}$ serves as a rough screening tool for determining their integrity, any observation of semiconducting behaviour will suggest that a significant quantity of the material is of low oxygen content. The absolute magnitude of the electrical resistivity also serves as an indicator of the connectivity and the oxygen content of the material. A low resistivity may indicate that the grains are well connected or that they have a high overall oxygen content.

3.2.2 Application of a Magnetic Field

In a polycrystalline sample with very poor inter-grain connections, even very small magnetic fields may exceed the critical magnetic field for the grain boundaries. In this instance the number of inter-granular links which are able to sustain a supercurrent will be reduced as the magnetic field is increased and the observed resistance transition will broaden out. Resistivity measurements as a function of applied magnetic field are therefore a valuable way of determining the strength of the inter-grain coupling.

The zero resistance state is dominated by the percolation threshold – where there is a continuous zero resistance path throughout the sample. In order to counteract the influence of this effect, it is customary to measure the width of the transition by taking the difference between the temperatures at which the resistivity has fallen to 90% and 10% of the normal state value. Similarly T_c is often defined as the temperature at which the resistivity has fallen to 50% of the normal state resistivity.

3.2.3 Experimental Details

The superconducting transition temperature of each sample was determined using a standard four probe resistance technique, thereby eliminating effects due to the contact resistance to the sample. Figure 3.2 shows a schematic of diagram of





the experimental arrangement. Bar shaped samples (approximately 3x3x15mm³) of material were cut from each pellet and mounted onto a small printed circuit board using GE varnish. The circuit board was attached firmly to a small copper block which in turn was mounted upon a stainless steel rod. Electrical contact between the copper tracks on the circuit board and the sample was achieved using electrically conducting silver paint dissolved in heptan-2-one, this gave contact resistances of the order of a few ohms. The current and voltage leads were soldered to the appropriate tracks of the circuit board, shielded and fed continuously through the top plate of the cryostat insert. The temperature of the sample was monitored using a copper-constantan thermocouple. One arm of the thermocouple was mounted adjacent to the sample whilst the second arm was fed through the top plate of the sample insert and immersed in liquid nitrogen. In order to minimise spurious thermal emfs the thermocouple was made from continuous lengths of wire bonded only at the copper-constantan junctions. The thermocouple voltage was read using a Fluke 8840A calibrated microvoltmeter and the sample temperature determined by interpolation of a copper-constantan thermocouple reference table. [Landolt-Börnstein (1984)]

A measuring current of typically 2.8mA was supplied by a constant current power supply (described in §4.3). The current was measured with a calibrated Keithley 175 multimeter and the the voltage drop across the sample, typically a few tens of microvolts, was measured by a Tinsley 6050 nanovoltmeter. Considerable care was taken to ensure that no ground loops existed and that the voltage leads were shielded from electromagnetic interference. No corrections were made to compensate for the small magnetoresistance of YBa₂Cu₃O_{7- δ}. Hall effect voltages were computed out by averaging the signal with the current in both forward and reverse directions. Owing to the limited space available within the cryostat, the magnetic field applied to the sample was always perpendicular the the direction of the current flow. The measuring current was frequently varied to ensure that the voltage drop was directly proportional to the applied current when the sample was in the 'normal' state. The existence of the superconducting state was verified by the absence of a voltage drop across the sample even when the current was increased. The cryostat was mounted between the pole pieces of an electromagnet and the magnetic field varied by applying a direct current to the electromagnet coils. A calibrated Bell 610 gaussmeter measured the applied magnetic field. The sample chamber of the cryostat was evacuated with a rotary pump and the sample cooled by radiative heat loss to the cryostat heat exchanger. An Oxford Instruments 3120 temperature controller regulated the temperature of the heat exchanger to ± 0.1 K.

A common problem with many experiments which attempt to determine the variation of a physical property as a function of temperature is that there may be thermal hysteresis caused by the sample not being in thermal equilibrium with the temperature sensor. During the course of the electrical resistivity experiments no such hysteresis was observed hence it was assumed that the sample and mounting block were in thermal equilibrium. Figure 3.3 shows a typical set of data obtained close to the superconducting transition. The broadening of the transition as the applied field is increased is clearly visible. Fluctuations in the applied current and voltage drop across the sample were negligible, the largest source of error was the uncertainty in the separation of the voltage wires. The silver paint contacts were approximately 1mm in diameter, this serves as an upper bound for the uncertainty in the voltage wire separation. The uncertainty in the absolute resistivity was approximately 10%.

3.3 Inductance Probe

3.3.1 Outline

When a material undergoes a transition to the superconducting state there is a marked change in the response of the material to an applied magnetic field (See §1.2), this may be used to determine the transition temperature of a superconducting material. Consider a coil of wire which surrounds a specimen of superconducting material above T_c . The magnetic flux, Φ , through the coil is given by the expression

$$\Phi = B.A = L.I \tag{3.2}$$

where B is the applied magnetic field, A is the cross sectional area intersected by the B field, L is the inductance of the coil and I the current applied to the coil. Upon passing into the superconducting state, the material expels magnetic



Figure 3.3 — Electrical resistivity transition of sample D980.

flux from the sample and screens it from further applied field, the effective crosssectional of the coil is reduced and the inductance of the coil decreases. It is important to note that with this technique the signal is dominated by screening currents not flux expulsion, so it does not follow that the volume fraction of superconducting material may be determined by this method. The drop in inductance does not, by itself, provide conclusive proof of the existence of a superconducting state – merely that there is some change in the response of the material to applied magnetic fields.

3.3.2 Experimental Details

A coil of approximately 2500 turns was wound onto a PTFE former 15mm in diameter and 25mm in length. The coil was attached to a stainless steel tube which was slowly lowered into the neck of an open helium dewar. The temperature of the coil was monitored using a copper-constantan thermocouple referenced to liquid nitrogen and the same experimental procedure for temperature determination followed as for the resistivity transition. The inductance of the coil (typically 12mH) was monitored using a Wayne-Kerr inductance bridge operating at 1kHz. Figure 3.4 shows data obtained for a sample of superconducting material, D980. The transition is broader than that determined from electrical resistivity measurements because the magnetic measurements are not as dramatically affected by the existence of a single percolative path, full shielding occurs when complete surface screening currents are established.

3.4 Magnetisation Curves

3.4.1 Inter-grain Currents

The currents flowing within a polycrystalline superconductor can be considered to consist of two components. The first current is the inter-grain current or transport current which flows through the links between superconducting grains, associated with this current will be an inter-grain critical current density J_c^t which is the current density at which a bulk material appears to lose its superconductivity. J_c^t is often measured by direct methods – applying an increasing current to the sample until a voltage drop across the sample is observed. The critical current density obtained in this manner depends upon the coupling between grains,





consequently values obtained for simple pressed powders, sintered materials and melt-textured materials will vary considerably.

3.4.2 Intra-grain Currents

The second current flowing within a polycrystalline superconductor is the intragrain current or local current which flows within the individual grains. Again, this current will have an associated critical current density, J_c^g . The intra-grain currents can only be determined by indirect methods. One of the most popular methods is to perform magnetisation measurements as a function of applied field upon powdered samples and from the magnetic hysteresis attempt to deduce the critical current density using a critical state model.

3.4.3 Bean Critical State Model

An elementary model to calculate the critical current density of a superconductor from hysteresis curves was proposed by Bean. [Bean (1962,1964] This model is frequently used on the grounds that a value of J_c can be rapidly found from the magnetic hysteresis. Bean produced the expression

$$J_c(\mathrm{Acm}^{-2}) = \frac{3 \times 10^4 \Delta M}{d} \tag{3.3}$$

where ΔM is the hysteresis of magnetisation per unit volume $(JT^{-1}m^{-3})$ and d is the scaling length (cm). With the application of any critical state model the scaling length used is important. For polycrystalline materials which have only been powder pressed the hysteresis arises as a result of flux pinning within the grains hence the appropriate scaling length will be the average grain size (typically a few μ m). For sintered samples the currents flow through the surface of the bulk sample and the appropriate scaling length is the sample dimension.

The Bean model makes a few assumptions, which must be borne in mind when assessing the values of $J_c(B)$ derived from the model. The assumptions Bean made are as follows.

1. The superconductor is capable of sustaining a lossless macroscopic current up to critical current density J_c , which is zero at B_{c2} .

- 2. The magnetic field is shielded up to the value B_{c1} .
- 3. The shielding currents flow to the full amount J_c only to the depth required to reduce the magnetic field to B_{c1} .

3.4.4 Vibrating Sample Magnetometer

In a vibrating sample magnetometer (VSM) a sample is vibrated in a direction perpendicular to an applied magnetic field which is uniform over the volume of the sample. With the sample fixed, the applied field induces a magnetic moment in the sample which in turn gives rise to a constant magnetic flux through the detection coils. By vibrating the sample, the magnetic flux through the detection coils changes periodically, and this induces a voltage which is proportional to both the magnetisation of the sample and the amplitude of vibration. Attachment of a small permanent magnet to the drive mechanism, with its own pickup coils, provides a reference signal which, in addition to the signal from the sample, can be fed into a phase sensitive detector (PSD). In this way the sample magnetisation can be measured with reasonably high sensitivity. Absolute measurements of magnetisation may be obtained by by calibration against a suitable standard.

A great strength of magnetisation measurements is their high sensitivity to magnetic impurities in a sample, hence a magnetisation curve of $YBa_2Cu_3O_{7-\delta}$ may provide evidence that the material contains a small amount of magnetic impurities. These impurities may have been present in the precursor materials, or may have been introduced during the materials processing stage. As ferromagnetic impurities may affect the superconducting properties, magnetisation measurements may be used to assess the viability of using a particular source of precursor materials or of using a particular preparation route.

3.4.5 Experimental Details

The Durham VSM has been described in detail by Hoon and Willcock (1988), consequently only a bare outline of its operation is provided here. The Durham VSM utilises an electrical motor to drive a 180° double throw crank assembly which in turn vibrates the sample. The head mechanism is pneumatically cushioned from the support mechanism in order to reduce noise. The sample is attached to

Figure 3.5 — The Durham VSM.



a borosilicate glass rod which is centred, within an Oxford Instruments CF1200 gas flow cryostat, by a PTFE bush. The magnetic field is applied by a Newport Instruments type D electromagnet and monitored with a Hall probe. Figure 3.5 illustrates the experimental arrangement.

In order to produce absolute measurements of the magnetisation, the residual signal from the sample holder and glass rod must first be subtracted from the data. Once this is complete the signal output from the PSD is calibrated using a nickel reference crystal which has a well known magnetisation as a function of temperature and applied field. Such a calibration procedure is only valid if the demagnetisation factor of the nickel reference crystal is taken into account.

An example of a magnetisation curve obtained from a sample of $YBa_2Cu_3O_{7-\delta}$ is given in figure 3.6. At fields below B_{c1} the sample is strongly diamagnetic, but when the applied field is increased above B_{c1} flux lines start to penetrate the sample and the material enters the mixed state. Above a field B_{irr} the magnetisation is reversible showing that there is free flux flow in the sample. The point at which this reversibility occurs corresponds to the onset of flux flow resistivity (discussed in §1.9) and the sample is no longer capable of sustaining a transport current.





Chapter IV

Thermal Conductivity Experiments

'pon my word, Watson, you are coming along wonderfully. You have really done very well indeed. It is true that you have missed everything of importance, but you have hit upon the method.

-Sir Arthur Conan Doyle.

'A Case of Identity'

4.1 Basic Measurement Methods

The flow of heat in a solid is governed by the equation

$$h_i = -\kappa_{ij} \frac{\partial T}{\partial x_j} \tag{4.1}$$

where κ_{ij} is the thermal conductivity tensor, h_i is the rate of heat flow per unit area perpendicular to direction *i* and *T* is the temperature. The values of h_i will usually be independent of the size and shape of the sample although in certain extreme cases (e.g. when the phonon mean free path is of comparable size to the sample) the thermal conductivity will be sample dependent. Such effects are not usually seen until very low temperatures although artificial sapphire has been known to exhibit this effect at $\simeq 30$ K. [Berman et al (1955)]

The measurement of thermal conductivity falls into two main categories - steady state and non-steady state methods.

4.1.1 Non-steady State Methods

Non-steady state methods make use of the idea that an arbitrary temperature gradient imposed upon a sample will change with time at a rate governed by the thermal diffusivity of the sample. If the relaxation time of the system is sufficiently small, a periodic temperature profile may be imposed upon the system and 'lockin' or data averaging techniques may be used. Non-steady state methods are particularly useful at low temperatures where systems have a low heat capacity and a short relaxation time. At high temperatures the relaxation time may increase to several hours. making accurate determination of the time constant difficult. A review of the various non-steady state methods is given by Touloukhian (1970).

4.1.2 Steady State Methods

Steady state methods rely upon applying a constant rate of heat input to a sample, thereby establishing an equilibrium temperature gradient across the sample. There are a variety of steady state methods which may be used. A detailed discussion of the methods available is given in Touloukhian (1970), Berman (1961) and Berman (1976). The majority of steady state techniques are variants of the longitudinal heat flow and radial heat flow techniques.

4.1.2.1 Longitudinal Heat Flow

With these methods the thermal conductivity is measured in an analogous way to the four probe measurement of electrical resistance. In the thermal case the heat flow corresponds to the electrical current and the temperature difference is analogous to the electrical potential difference. A popular experimental arrangement is to hold one end of a sample rod at a constant temperature whilst a constant rate of heat input (\dot{Q}) is applied to the other. At equilibrium this results in a temperature gradient being set up which may be measured using suitable temperature sensors. Longitudinal heat flow methods generally have the advantages of being relatively simple to set up and of allowing direct computation of the thermal conductivity through the mathematical relationship:

$$\kappa = \frac{\dot{Q}l}{A\Delta T} \tag{4.2}$$

where ΔT is the temperature difference measured by the temperature sensors sited a distance *l* apart. The time taken for such a system to reach 1% of its equilibrium value is given by the expression due to Reese (1966).

$$\tau = \frac{4Cl^2}{\pi^2\kappa} \tag{4.3}$$

where C is the heat capacity per unit volume, κ is the thermal conductivity and l is the length of the sample. This time constant may become very long for samples which have a very low thermal conductivity or which are particularly long. For a sample of YBa₂Cu₃O_{7- δ} 20mm long, and using a thermal conductivity of a few Wm⁻¹K⁻¹, the time constant will be of the order of a few minutes at 100K.

4.1.2.2 Radial Heat Flow

In order to reduce the effective length of the sample, and thereby decrease the equilibrium time, the sample may be cut into the form of a disc and a radial heat flow technique employed. With this technique the centre of the sample is held at a constant temperature and a heater wrapped uniformly around the circumference of the disc. A constant heat input to the heater will therefore establish a radial temperature gradient throughout the sample. With this geometry the thermal conductivity is given by the expression

$$\kappa = \frac{\dot{Q}\ln(r_1/r_2)}{2\pi l \Delta T} \tag{4.4}$$

where r_1 and r_2 are the radii at which the temperature difference is measured. Radial heat flow methods also have the modest advantage of being dependent upon the ratio of two radii, thereby eliminating errors due to thermal expansion. Use of this method can lead to a reduction in the equilibrium time of an order of magnitude.

4.2 Design Considerations

Any experimental arrangement to measure thermal conductivity is the result of compromise because all connections to the sample incur the penalty of acting as a heat shunt. As a consequence of this the design of a system is strongly
dependent upon the properties of the materials to be measured, hence the system described here represents only one attempt at the solution of the design problems encountered.

Early measurements performed upon $YBa_2Cu_3O_{7-\delta}$ (see §2.6) indicated that the thermal conductivity is comparable to that of a poor metal, ie in the region of 0.1 to 10 Wm⁻¹K⁻¹, thus the system was designed to measure thermal conductivities within this range. As this was the first thermal conductivity measurement system to be set up in the Durham Physics department, a detailed description of the consideration of errors and factors influencing the system design is given here.

The samples of superconducting material to be investigated were produced as 20mm diameter pellets. However these suffered from a tendency to crack as they were removed from the die. This made radial heat flow methods impractical. After sintering the pellets were easily cut into bars of approximate dimensions 20x5x3mm³, hence the choice of a longitudinal heat flow technique was a natural one.

4.2.1 Temperature Stability

For a steady state technique to be effective, one has to ensure that the temperature gradient across the sample has reached an equilibrium value and that the gradient has not arisen from any temporary drift in temperature. For this reason it is important that the temperature of the cold tip of the cryostat is maintained at a constant value. Fortunately, attachment of the temperature sensor and cold tip heater to an automatic temperature controller makes possible the holding the temperature of the cold tip constant to better than ± 10 mK.

The temperature of the cold plate of the cryostat study was regulated by a Lakeshore Cryotronics DR91C temperature controller fitted with a Lakeshore Cryotronics GaAlAs diode sensor. The sensor was mounted inside a copper block which was in turn screwed into the central arm of the sample clamp, a good thermal contact between the sensor and the copper block was ensured by thermally anchoring the sensor leads to the block. Calibration of the sensor was performed by the manufacturers according to the American NBS standard and initially enabled absolute temperature to be measured to ± 10 mK over the entire 20K-120K range. Unfortunately, during the course of the experiments, the sensor performance degraded, possibly due to the mounting adhesive used by the cryostat manufacturers slowly attacking the sensor housing (the explanation was given by the sensor importers) and a second sensor had to be used. The characteristics of this GaAlAs diode were compared against a calibrated rhodium/iron resistor and the resulting calibration curve fitted to a 12^{th} order chebyshev polynomial using a standard least squares algorithm. [Press et al (1986)] The temperature reproducibility of this second sensor was approximately ± 10 mK and did not appreciably affect the stability of the cold plate.

With a temperature difference, $T_1 - T_2$, established across the sample, the measured thermal conductivity is an average of the values between T_1 and T_2 . Berman (1976) has shown that if the material has a thermal conductivity which varies as sharply as T^3 then the difference in apparent and true conductivities is less than $\frac{1}{4}\%$ even for $(T_1 - T_2)/T_1 \simeq 1/10$. In this study, the temperature gradients used varied between 0.5K and 1.5K depending upon the temperature of the sample.

4.2.2 Sample Heater

In order to limit the errors in the value of the thermal conductivity data obtained, the heat input to the free end of the sample must be known. Ideally, there should be no contact to the sample heater at all; the entire heat loss would then be due to the black body radiation emitted by the heater. Such a system could be achieved by the use of a powerful light source whose output is channelled to the end of the sample by a fibre optic cable. This arrangement is feasible for situations where relative changes in heater power are required and indeed this method has been exploited by Stokka and Fossheim (1988) when performing high resolution 'a.c.' heat capacity measurements. However, absolute measurements of thermal properties by this method are difficult as the efficiency of the transmission cable and emissivity of the sample must be known at all temperatures.

Use of a simple resistance heater requires a compromise to be reached between the necessity to have leads connected to the heater (which shunt heat out of or into the sample) and the resistance of the current leads which dissipate power by ohmic heating. If the wire is too thin or too long the resistance of the leads could be a significant proportion of the heater resistance, whereas if the leads have too great a cross sectional area, a significant amount of heat may be shunted into or out of the sample, resulting in an inaccurate value for κ . A discussion of this aspect of the design compromise may be found in Berman (1961) and White (1979).

The sample heater used in this study was a standard RS 120Ω strain gauge. This had the twin advantages of being compact, measuring only 8x3x0.25mm³, and of having a low residual resistance ratio, such that the resistance fell only slightly over the 20K-120K range. Conduction of heat to, or away from, the sample by the heater leads was limited by 20mm lengths of 46swg manganin wire. These wires had the effect of adding approximately 1.5Ω to the heater resistance at room temperature thereby leading to an uncertainty in the heater power of approximately 1.5%. The current was supplied to the heater by 32swg copper wires which were thermally anchored to the central post of the sample clamp. A further two 32swg copper wires attached in the same way allowed direct measurement of the voltage drop across the strain gauge and manganin wire, thus enabling the power developed across the heater to be known accurately.

4.2.3 Residual Gas Conduction

At low pressures, gas molecules have a large mean free path and will tend to travel from one surface to another without colliding into other molecules. The transfer of heat under such conditions is given by the equation [White (1979)]

$$\dot{Q}_{gas} = \alpha \; \frac{\gamma + 1}{\gamma - 1} \frac{\sqrt{R}}{\sqrt{8\pi}\sqrt{MT}} P(T_2 - T_1) \tag{4.5}$$

where γ is the ratio C_p/C_v , P is the gas pressure in Pa, R is the universal gas constant, M is the molecular weight of the residual gas and α is an 'accommodation coefficient' which is related to the corresponding coefficients for each surface. Assuming that the residual gas in the system is helium, the surfaces are poor ($\alpha = 1$) and the pressure gauge is at room temperature, the above expression reduces to

$$\dot{Q}_{gas} \simeq 0.03 P(T_2 - T_1) \quad \text{Wcm}^{-2}$$
 (4.6)

For particularly poor surfaces and a temperature difference between the sample and radiation shield of 2K, the heat loss due to residual gas conduction from a sample 2x0.5x0.3cm³ and a strain gauge 0.8x0.3x0.025cm³ is $0.226 \times P$ Watts where P is the residual gas pressure in Pascals. In all of the thermal conductivity experiments discussed in this thesis the sample was held in a vacuum better than 10^{-8} Pa, hence for an applied heater power of 5mW the effect of residual gas conduction was negligible.

4.2.4 Radiation Losses

A black body at a temperature T emits radiant energy at a rate $\dot{Q} = \epsilon \sigma T^4$ per unit area, where σ is Stefan's constant, $(5.7 \times 10^{12} \text{ Wcm}^{-2} \text{K}^{-4})$ and ϵ is the emissivity of the material. If the sample or heater is at 100K and is surrounded by material at 4.2K, the losses may be as high as 0.6mWcm^{-2} . Such losses can appreciably affect the experimental measurements. In order to keep these losses to a minimum, the sample heater was covered with clean aluminium foil which has an emissivity of much less than 1, reducing the losses by a factor of 50 or so. [White (1979)] The cryostat design was such that the vacuum can was thermally connected to the sample cold plate, ensuring that the sample was always surrounded by a material at a similar temperature.

4.2.5 Measurement of Temperature Difference

Accurate measurement of small temperature differences is extremely difficult, requiring the use of either two well-characterised sensors (typically calibrated to better than 5mK) or the use of a direct method of measuring the difference. A thermocouple has obvious potential for the task of direct measurement of a temperature difference, although great care must be taken to ensure that offset currents and spurious emfs are eliminated. A thermocouple made from gold+0.03% iron vs chromel has the advantage of a (roughly) constant and high sensitivity $(\simeq 13\mu V/K)$ at temperatures from 4.2K to 300K [Berman et al (1964)] whereas many other thermocouples suffer from loss of sensitivity as the temperature decreases, e.g. copper-constantan has a sensitivity of $\simeq 12\mu V/K$ at 300K which falls to only $\simeq 2.5\mu V/K$ at 4.2K. [Berman et al (1965)] The measured thermocouple voltage may be readily converted to a temperature difference by use of standard reference tables and appropriate interpolation algorithms. A disadvantage to the use of a differential thermocouple is that the absolute temperature of the sample is not measured, it is therefore common practice to use a second thermocouple or other temperature sensor to measure absolute temperature of the sample. With the cryostat used in this study, no fixed reference temperature was available, hence the use of a thermocouple to determine absolute temperature was not possible. A second high resolution temperature sensor with suitable characteristics was not available, however the samples were by necessity short, temperature gradients were kept to a minimum and the control sensor was thermally anchored to the cold tip at a point as close as possible to the sample, thus the absolute temperature sensor. The temperature gradients of 0.5K-1K used for each measurement provides an estimate for the uncertainty in the sample temperature. For accurate work at helium temperatures this state of affairs is most unsatisfactory although it is adequate for the measurements made in this study which were at temperatures above 20K.

The measurement of small temperature differences by use of a direct thermocouple is complicated by the presence of offset currents which arise principally from the finite input impedance of the nanovoltmeter. The effect of these currents is that even when the thermocouple junctions are thermally anchored to the same point, the nanovoltmeter will indicate a finite voltage. During the course of these experiments, the offset currents were nulled by a control on the nanovoltmeter which injected a current in opposition to the input offset current. Offset voltages may also arise from strain, inhomogeneities or local temperature gradients in the thermocouple wires. Such voltages may be compensated for by use of superconducting reversing switch which enables measurement of the thermocouple voltage in a positive and negative sense. Unfortunately, the cryostat design did not allow the use of such a switch, hence the offset voltages were computed by removing the heat input to the free end of the sample and allowing the thermocouple arms to reach equilibrium. Further checks were made by measuring the thermal conductivity for different sample heater powers. This is analogous to measuring the electrical resistivity of a sample by applying various currents to the sample. Figure 4.1 shows a typical set of data for the thermocouple voltage measured across a sample as the heater power is increased, in the presence of an offset voltage on the thermocouple. At high heater powers the function will deviate from linear





owing to the influence of radiation losses and the gradual change in the thermal conductivity of the sample along the direction of the temperature gradient.

4.2.6 Thermocouple Wires

A further possible source of error in the experiment is conduction of heat by the thermocouple wires, resulting in an artificially low temperature gradient being established along the sample. As the thermocouples are anchored to the sample clamp and the temperature difference always kept below 1K, this value is used to estimate the conduction 'error' from the thermocouple wire. At 100K, gold has a thermal conductivity of approximately $300Wm^{-1}K^{-1}$ and chromel has a thermal conductivity of approximately $20Wm^{-1}K^{-1}$. Each wire has a nominal diameter of 80μ m hence:

$$\dot{Q}_{AuFe} = \frac{300 \times (40 \times 10^{-6})^2 \times \pi}{l} = \frac{1.5}{l} \mu W$$
(4.7)

$$\dot{Q}_{Chromel} = \frac{20 \times (40 \times 10^{-6})^2 \times \pi}{l} = \frac{0.1}{l} \mu W$$
(4.8)

For a measuring power of only 1mW the heat conduction down 1 metre of each wire is respectively 0.15% and 0.01% of the total power, in practice the thermocouple wires were approximately 2m in length. In order to fit the wires into the sample space it was necessary to form them into loosely bound coils. This was achieved by carefully winding the each wire onto a stainless steel bar where and lightly holding it with very weak GE varnish. Once the varnish had set, the wires were carefully removed from the rod and small pieces of hair tied around each coil at three points in order to keep the coil in place. The GE varnish was then removed using methanol/toluene solvent.

The thermocouple junctions proved to be extremely difficult to make owing to the very large difference in melting points of the two alloys. They were eventually successfully welded by the following technique: the ends of wires to be joined were stripped of their enamel using a methanol/toluene solvent, the final 1.5cm of the wires were then twisted together and lightly clamped using a pair of fine stainless steel tweezers. A conventional car battery was used to charge two 33mF capacitors in parallel, the battery was disconnected and one terminal of the capacitors connected to the tweezers. The second terminal of the capacitors was connected to a small, clean copper plate. Spot welding of the wires was achieved by touching the ends of the wires on the copper plate – effectively shorting out the capacitors. Possible oxidation of the wires was impeded by performing the weld in a helium rich atmosphere. A visual check of each weld was made using an optical microscope to ensure that the two wires had fused together, a few attempts were necessary before an acceptable bond was made. Once the junction was made, the wires were carefully untwisted and covered with a thin layer of GE varnish to restore the electrical insulation.

With two thermocouple junctions successfully made, the three coils of wire were supported in the sample space by fine cotton thread to ensure good thermal isolation from the rest of the system. The free ends of the chromel wires were anchored to the clamp centre post and to a binding post situated on the copper cold plate. At this point the chromel wires were soldered to 0.2mm copper wires, coated in GE varnish and bound together. Some problems were encountered with the copper wires, as appreciable emfs ($\simeq 5\mu V$) were generated in some lengths of wire when they were immersed fully in liquid nitrogen and connected directly to a voltmeter. This problem was overcome by using lengths of high purity copper wire which gave emfs of less than $1\mu V$ when immersed in liquid nitrogen.

From the chromel-copper junctions the copper wires were taken to a second binding post on the cold plate before being led through a separate stainless steel tube to the top of the sample insert. Here they they were fed continuously through the top plate of the cryostat, the vacuum seal being made with Stycast 2850FT resin set with LV24 catalyst. Outside the cryostat, the wires were separately covered with insulating heat shrink and gently twisted together before being shielded by a copper braid. The shield, nanovoltmeter and cryostat were earthed via the copper water pipes in the laboratory, in this way the possibility of spurious thermal emfs and signals from radio-frequency noise were minimised.

4.3 Experimental Arrangement

4.3.1 Cryostat

The variable temperature insert of the cryostat used for the studies in this thesis is depicted schematically in figure 4.2. The sample and associated equipment

is attached to a copper plate at the end of an insert some 1.3m long and the insert lowered into the cryostat and screwed firmly into a heat exchanger. Radiation losses are reduced by a number of copper heat baffles placed at regular intervals along the tube. This particular cryostat design allows the vacuum seal to be made at room temperature, although it does suffer from the possibility of the sample and associated equipment being stressed as the insert is lowered and screwed into the heat exchanger. The sample is cooled by maintaining the bath of liquid helium at a slight overpressure, thereby forcing liquid helium through the needle valve and into the heat exchanger. A needle valve restricted the gas flow, further temperature regulation was achieved by use of a 25 Watt resistance heater mounted on the cold plate.

4.3.2 Sample Mounting

The samples were held by a simple clamp arrangement constructed from copper (see figure 4.3). The lower screw enables the sample to be loosely clamped, whilst the upper screw may be carefully tightened to ensure a firm grip. In order to insulate electrically the sample from the clamp, the clamp faces were covered with cigarette paper soaked in weak GE varnish. The thermal contact between the sample and copper clamp was aided by coating the clamp faces with a small quantity of vacuum grease. Two 0.3mm diameter holes were drilled into the samples in order to accept the thermocouple junctions, these were filled with vacuum grease in order to improve the thermal contact and allow a good bond length between the thermocouple junctions and the sample. [Kopp and Slack (1971)]

4.3.3 Constant Current Power Supply

In order to facilitate accurate and reliable measurements of both electrical resistance and thermal conductivity as a function of temperature, a constant current power supply was required. The power supply was designed to provide a constant current of 1mA to 10mA to the sample heater. A conventional car battery was used as the basic power source for the supply, thereby virtually eliminating 50Hz noise which is present to some degree in all mains powered supplies. The basic circuit is shown in figure 4.4.









Figure 4.4 — Circuit diagram of the constant current power supply.



An integrated circuit, a 334Z current stabiliser, provided a basic stabilising circuit. The output of this circuit was smoothed further by use of two 33mF capacitors in parallel. The current output from the supply was selected by altering the resistance input to the 334Z, achieved by use of a rotary switch selecting one from a range of resistors. Two further switches were also incorporated, the first permitted polarity reversal of the current (useful for resistance measurements), whilst the second enabled an ammeter to be included in the circuit. All of the resistors in the construction of the supply were of a metal oxide type which were chosen for their low inherent noise.

Figure 4.5 shows the typical output current of the supply to a strain gauge of 120 Ω resistance. The current is seen to be stable over a long period of time, to within 1.5 μ A over 15 hours, a change of 0.15%. This is of particular importance for the thermal conductivity measurements where the heater power has to be stable in order to establish and maintain an equilibrium temperature gradient. The drop in current was caused by the small drain on the car battery. The stability of the output current from the supply was checked at frequent intervals throughout the time experimental data was accumulated, no change in the stability was observed.

4.3.4 Data Acquisition System

The measurement of thermal conductivity was automated, initially by use of an Acorn BBC microcomputer with IEEE interface, and later by use of an 'IBM PC' compatible Elonex 286 PC-AT fitted with a Scientific Solutions IEEE interface card. All of the measurement instruments (except the nanovoltmeter) were fitted with IEEE cards to enable the two way flow of information from the PC which acted as an overall system controller. The nanovoltmeter could not be fitted with an IEEE card, however an analogue output from a precision amplifier was available and this output was fed into a Keithley 197 microvoltmeter which was fitted with an IEEE bus. The analogue output was found to correspond to the indicated voltage on the analogue display of the nanovoltmeter for ranges of 30μ V fsd and lower. As 30μ V corresponds to a temperature gradient of approximately 2K this range proved adequate for the measurement. A further limitation imposed by the nanovoltmeter was caused by the nanovoltmeter being unable to drive the analogue output continuously for more than a couple of hours without



ι

the internal batteries becoming exhausted. The computer therefore controlled a Bede Scientific Instruments MINICAM fitted with an IEEE interface which fired a relay connecting the nanovoltmeter to the mains power. The nanovoltmeter could then be electrically isolated before each reading and the integrity of the electrical grounding maintained. A schematic diagram of the system connection is given in figure 4.6.

The source code of the program controlling the entire system was written in BASIC for the BBC and in Turbo Pascal for the PC. Briefly, the program allows the operator to program a set of times at which the set temperature is to change. At each temperature the computer changes the PID values on the temperature controller to appropriate values to reflect the change in the response of the system as it cools. Data may be taken at intervals specified by the operator and plotted on screen, enabling an immediate display of the thermal conductivity as the temperature is altered. Automation of the data acquisition has the great advantage of allowing data acquisition to occur overnight, within certain restrictions brought about by the physical logistics of the cryostat (for example it is not possible to control the flow rate of helium gas using the computer). A complete list of the Turbo PASCAL source code of the control software is given in appendix E.

4.4 Test Data

In order to verify that the thermal conductivity measurement system was capable of performing accurate measurements upon a range of materials, the system was initially tested with a sample of perspex. Perspex has a thermal conductivity over an order of magnitude lower than that expected of the YBa₂Cu₃O_{7- δ} samples. If there were significant sources of error from radiation losses, residual gas conduction, conduction along the thermocouple wires or conduction along the sample heater wires, the measured thermal conductivity of perspex would be significantly higher than those of the literature values. Consequently if the value of a low thermal conductivity material was measured correctly it is reasonable to assume that the results for materials of higher thermal conductivity would be substantially correct.

A perspex sample, of dimensions 20x3x3mm³ was clamped to the cold tip of the cryostat and the thermal conductivity measured at temperatures just above 77K.





This served the secondary purpose of determining whether or not there were any anomalies in the thermal conductivity apparatus which might otherwise be falsely attributed to the superconducting nature of $YBa_2Cu_3O_{7-\delta}$. Owing to the long time constant of the system caused by the low thermal conductivity of perspex, the thermal conductivity was measured at each temperature after allowing 24 hours for the sample to reach equilibrium. The results in the region 80K to 100K are depicted in figure 4.7. The data of Burgess and Greig (1974) indicates that the thermal conductivity of of polymethylmethacrylate in this temperature range is $\simeq 0.16Wm^{-1}K^{-1}$. This is in agreement with the data obtained and therefore, as an initial test, showed that the thermal conductivity apparatus was capable of producing acceptable data.





Chapter V

Thermal Conductivity Study I Effect of Oxygen Content

The first moral of this story is just a practical one. Always test your general reasoning against simple models.

— John Bell.

Speakable and Unspeakable in Quantum Mechanics

5.1 Introduction

In §2.6 it was stated that the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ varies considerably with porosity of the sample. In addition, the detailed microstructure will influence the overall thermal conductivity in the normal and superconducting state. Very few thermal conductivity results have been published in which a property of the same sample of $YBa_2Cu_3O_{7-\delta}$ has been systematically altered. In this study a sample of $YBa_2Cu_3O_{7-\delta}$ was characterised before undergoing vacuum and oxygen annealing to alter the oxygen content of the material. This enabled the influence of oxygen content upon the thermal conductivity to be examined in a systematic way without the influence of variations in porosity, sample size or the detailed microstructure of the material.

5.2 Sample Preparation

The material was prepared by solid state reaction of precursor compounds obtained from Johnson Matthey Materials: 99.9999% yttrium oxide, 99.999% barium carbonate and 99.999% copper (II) oxide. The precursor materials were mixed in an agate mortar and placed in an alumina crucible. Calcination was performed at 990°C for 21 days using a Carbolite CTF12/65 tube furnace fitted with a Pt vs Pt+10%Rh thermocouple, solid state relay and Eurotherm 818P temperature programmer. The calcined material was ground in an agate mortar, placed in the crucible and heated to 985°C for 48 hours and cooled to room temperature at 40°C/hr. After the calcination process the material was finely ground in a Specamill 8000 agate mill for 30 minutes.

A 20mm diameter pellet of the material was produced by pressing in a Specac stainless steel die at a presssure of 405MPa (4kbar) using a uniaxial bench press. Sintering was performed at 970°C for 24 hours followed by cooling to room temperature at 40°C/hr. Oxygen annealing was performed at 450°C for 48 hours under flowing oxygen followed by a slow cool at 20°C/hr. This resulted in a material which was approximately 85% of the expected density of single crystal YBa₂Cu₃O₇. From the pellet, a bar of dimensions 20x5x3mm³ was cut using a fine bladed saw. After each set of measurements, the bar was subjected to a heat treatment in order to alter the oxygen content. The sequential list of heat treatments is given in table 5.1.

Reference	Temperature	Pressure	Time	T _c	
D970/1	450°C	1013 mbar	48 hrs	92 K	
D970/2	450°C	10 ⁻² mbar	24 hrs	68 K	
D970/3	450°C	10 ⁻² mbar	40 hrs	< 4.2 K	
D970/4	500°C	1013 mbar	48 hrs	90 K	

Table 5.1 — Sequential heat treatments of sample D970

5.3 Preliminary Characterisation

5.3.1 X-ray Powder Diffraction

A portion of the 'as-made' sample was powdered and analysed using the X-ray powder diffractometer described in §3.1. The material was found to consist of single phase YBa₂Cu₃O_{7- δ} with no evidence for the presence of Y₂BaCuO₅ or BaCuO₂. The diffraction lines for the (010) and (100) peaks were distinguishable indicating that a substantial quantity of the material had an orthorhombic structure. The c axis lattice parameter was determined to be 11.70 ± 0.01 Å, a rough estimate for the oxygen content of the sample may be obtained by combining this information with the correlation between c axis lattice parameter and oxygen content discussed in §2.3. From this relationship, the oxygen content of the sample was determined to be approximately 6.85 O atoms per Y atom.

5.3.2 Electrical Resistivity Measurements

The electrical resistivity as a function of temperature of a second bar cut from the 'as made' pellet is shown in figure 5.1. Use of a second bar ensured that the integrity of the sample used for the thermal conductivity experiments was maintained. The sample had a low electrical resistivity, $0.4m\Omega$ cm, at 100K, and exhibited a sharp transition at 92K in an applied field of less than 0.1mT. As the applied field was increased, the 90%-10% transition width broadened from 1K to 3.5K at 500mT. The low resistivity at 100K, the sharpness of the resistance transition and its resilience to change in relatively high magnetic fields is indicative of the sample having a high overall oxygen content and good inter-grain coupling. No resistivity measurements were performed upon the sample in each of the vacuum annealed states, however the resistance of the sample was measured after it had been re-oxygenated. The material exhibited evidence for semiconducting behaviour. This suggested that the oxygen content of the material had not been restored to its initial level.

5.3.3 Inductance Measurements

The inductance transition of the sample in each state was measured using the apparatus described in §3.3 As the measurements were performed using coils of different inductance, a 'normalised' inductance given by

$$L_{norm}(T) = \frac{L(T) - L(4.2K)}{L(100K)}$$
(5.1)

was defined, this allowed a direct comparison of the transitions to be made. The results are plotted in figure 5.3. The sample in the 'as made' condition had a T_c of 92K and a 90%-10% transition width of approximately 10K which is considerably

Figure 5.1 — Resisti	ity transition	of sample	in	state	D970	/1	•
----------------------	----------------	-----------	----	-------	-------------	----	---

0	<0.1mT	$+ \ldots 0.5 mT$	$\Delta \dots 5 mT$
0	$\dots 50 mT$	□ 500mT	



Figure 5.2 — Inductance transition of sample in states D970/1 to D970/4.

0	D970/1	+ D970/2	△ D970/3	٥	D970/4
---	--------	----------	----------	---	--------



broader than the resistivity transition. After the first vacuum anneal, no evidence for superconductivity was seen until approximately 68K when the inductance fell gently, further cooling revealed a sharp transition at 60K. After the second vacuum anneal, the sample did not superconduct even when placed in liquid helium, suggesting that much of the oxygen has been removed from the sample. After oxygen re-annealing, the sample showed a sharp drop in inductance at 90K with a transition width of 12K, slightly broader than that of the 'as made' sample.

5.3.4 Oxygen Content

The preliminary characterisation data may be used to estimate the oxygen content of the sample in each of the different states. This is achieved by comparison of the measured T_c values of the material in each state with the data of T_c as a function of oxygen content obtained from the published literature (see §2.3). The width of the superconducting transition for the sample in each state provides some indication of the spread of T_c values within the material. The estimated ranges of oxygen content of the sample in each annealed state is given in table 5.2.

Treatment	Method	T _c	ΔT_c	O _x
D970/1	Resistance	92K	1 K	x > 6.95
	Inductance		10K	x > 6.85
D970/2	Inductance	68K	10K	6.5 < x < 6.7
D970/3	Inductance	< 4.2K		6.0 < x < 6.3
D970/4	Resistance	90K	12K	x > 6.8
	Inductance	90K	12K	x > 6.85

Table 5.2 — Approximate oxygen content of sample D970.

5.4 Thermal Conductivity Measurements

The thermal conductivity of the sample as a function of temperature in the range 20K-120K was measured for the sample in each of the four states using the thermal conductivity apparatus described in chapter IV. A preliminary measurement of the thermal conductivity at temperatures down to 77K was made upon

the sample in the 'as made' state using liquid nitrogen as coolant. The sample was then warmed, removed and re-mounted before being cooled again using liquid helium. The measured thermal conductivity at temperatures down to 77K was the same, within experimental error, whether helium or nitrogen was used as coolant. Figure 5.3 shows the thermal conductivity of the sample in each of the four states. All states which exhibited a superconducting transition showed an enhancement of the thermal conductivity upon passing into the superconducting state, consistent with the results of many other research groups (see §2.6).

A second striking feature of the effect of the vacuum annealing treatment is that as the oxygen is removed from the material there is a dramatic decrease in the normal state thermal conductivity of the material. The thermal conductivity of sample D970/1 at 100K being approximately 40% higher than that of D970/3. The mild heat treatments given to the sample would not be expected to alter significantly the microstructure of the sample. When oxygen is partially restored to the material (D970/4), the thermal conductivity in the normal state increases and the enhancement of the thermal conductivity below T_c is again observed. These results provide conclusive evidence that the thermal conductivity of the material in both the normal and superconducting state is very strongly influenced by oxygen content.

5.5 Quantitative Model

5.5.1 Formulation

As the understanding of the physics of a problem and unambiguous interpretation of experimental results is difficult without a quantitative model with which to test various hypotheses, an elementary quantitative model has been developed. The basis of the model is the construction of the Boltzmann equation for a phonon 'gas' which is solved approximately by utilising the relaxation time approximation. The terminology and mathematical formulation of this approach is given in appendix A, although for clarity a brief outline of the formulation and the assumptions made are given here.

The Boltzmann equation is constructed by considering the 'heat current' which arises from the phonon modes, q_i . In the presence of a temperature gradient, the

Figure 5.3 — Thermal conductivity of sample in states D970/1 to D970/4.





distribution of the phonons will depart from the equilibrium distribution, $N^0(\mathbf{q}_i)$, given by the Bose-Einstein statistical factor

$$N^{0}(\mathbf{q}_{i}) = \frac{1}{e^{\hbar\omega_{i}(\mathbf{q}_{i})/k_{B}T} - 1}$$
(5.2)

There will therefore be a 'heat current' which arises as a result of the phonon drift if $N(\mathbf{q}_i) \neq N(-\mathbf{q}_i)$. In the steady state, the overall phonon distribution will be independent of time, consequently the phonon drift must exactly counteract all other processes which alter the phonon distribution – so called 'scattering' processes.

The relaxation time approximation assumes that the scattering processes will restore equilibrium at a rate which is proportional to the departure from equilibrium. This rate is characterised by a relaxation time, τ . By making this approximation, and considering only heat flow which is parallel to the direction of the temperature gradient the thermal conductivity is given by the expression

$$\kappa(T) = \frac{1}{3} k_B \int_{\omega} \mathbf{v}_g(\omega)^2 \tau(\omega) \left(\frac{\hbar\omega}{k_B T}\right)^2 \frac{e^{\hbar\omega/k_B T}}{(e^{\hbar\omega/k_B T} - 1)^2} D(\omega) d\omega$$
(5.3)

The density of states in the above expression involves integrating over all of q space i.e. averaging over all possible crystal orientations. This is accounted for in equation 5.3 by assuming that that an 'average' phonon group velocity may be found, for each phonon energy, which is capable of reproducing the phonon density of states, $D(\omega)$. A detailed discussion of this point may be found in the text of Grimvall (1986). Since the thermal conductivity measurements were made upon polycrystalline samples, it is reasonable to assume that, experimentally, a similar average has been performed and that the information based upon the phonon density of states is adequate. Partial phonon dispersion curves for YBa₂Cu₃O_{7- δ} have been measured by Reichardt et al (1989). Their results suggest that upon splitting the measured phonon density of states into appropriate energy intervals it is reasonable to reproduce the empirical phonon density of states using constant (i.e. frequency independent) average group velocities over each interval. With these approximations, equation 5.3 may be used to help elucidate the origin of

the temperature dependence of the thermal conductivity as the oxygen content is altered.

Each mechanism which scatters phonons may be characterised by a relaxation time and it is this relaxation time approximation which provides a mathematical description of the physics of the scattering mechanisms. Unfortunately, it is not possible to calculate exactly the overall relaxation time for the convolution of all of the phonon scattering mechanisms. It is therefore useful, as a first approximation, to consider each of the scattering processes as independent. This implies that the relaxation time may be written in terms of the relaxation times for each of the independent scattering mechanisms. This is an example of Mattheissen's 'rule'. [Mattheissen (1863)]

$$\tau^{-1} = \sum_{j} \tau_{j}^{-1} \tag{5.4}$$

A detailed discussion of the validity of the relaxation time approximation has been given by Carruthers (1961). He noted that the major problem is that the distribution function $N(\mathbf{q}_i)$ depends not only upon the occupation of the state being considered but upon all other states as well. Hence the characteristic relaxation time for each scattering mechanism will depend upon how far the system has deviated from equilibrium. For systems which are close to equilibrium, the approximation is likely to be a good one.

5.5.2 Obtaining the Phonon Density of States

In order to use equation 5.3 to compute the thermal conductivity, the 'bare' phonon density of states should be used. Measurement of the 'bare' phonon density of states by inelastic neutron scattering is made difficult due to the neutron data being weighted by a numerical factor which accounts for the 'strength' of neutron scattering by each atomic species. This factor depends upon the scattering cross section and the inverse of the mass of each atomic species. For Y, Ba Cu and O this factor is 0.08449, 0.0252, 0.1178 and 0.2647 barns/atom/amu respectively. As the higher energy phonon modes ($\hbar \omega > 30$ meV) may be attributed almost entirely to oxygen vibration states [Chaplot (1990), Reichardt (1989)], then unless these scattering factors are accounted for the measured phonon density of states will be weighted too strongly towards high energies.

5.5.3 Umklapp Processes

It is relatively straightforward to show that normal scattering processes preserve the crystal momentum and so will not restore an arbitrary phonon distribution to equilibrium, although they will influence the populations of each phonon mode when scattering mechanisms are present. [see, for example, Ashcroft and Mermin (1976)] Umklapp processes occur when two or more phonons scatter and the sum of the wave-vectors lies outside the first Brillouin zone. The onset of a significant contribution to the thermal conductivity from Umklapp processes may be estimated from the first peak in the phonon density of states. Figure 5.4 assists in understanding this idea. The phonon density of states is related to the group velocity $\nabla_{\mathbf{q}}\omega(\mathbf{q}_i)$

$$D(\omega) \propto \sum_{i} \int_{S_q} \frac{dS}{\nabla_{\mathbf{q}}\omega(\mathbf{q}_i)}$$
 (5.5)

Referring to figure 5.4, in region 1, the phonon modes have a large group velocity and low q, this will lead to a small contribution to the phonon density of states. The periodicity of the lattice requires that the dispersion curve should be zero at the zone boundary. In region 2, the phonon modes have a low group velocity and a high q, giving a large contribution to the phonon density of states. Near the edge of the first Brillouin zone (region 2) the periodic boundary conditions upon the lattice requires the gradient of the dispersion curve to become shallow, hence the phonon density of states will be high. As the energy is increased further (region 3), the modes will have a higher group velocity and a lower q leading to a fall in the phonon density of states. Hence the first peak in the phonon density of states is indicative of the point at which 2 phonons can just Umklapp scatter outside the first Brillouin zone. From the measured density of states of Strobel et al (1988), Rhyne et al (1987) and the calculated 'bare' density of states of Chaplot (1988), this peak is seen to occur at $\hbar\omega_u \sim 16 \text{meV}$ and is very nearly independent of the oxygen content of the sample. A characteristic temperature at which Umklapp processes become effective may be determined by assuming that a significant number of Umklapp processes occur when the typical phonon energy is half that at the zone boundary. If it is further assumed that the modes are





(b) Polyatomic lattice.





populated up to a typical energy $3k_BT$, then ω_u is given by

$$\frac{1}{2}\hbar\omega_u \simeq 3k_B T_u \tag{5.6}$$

As the temperature increases, the number of Umklapp processes occurring will increase and there will be a reduction in the phonon-phonon relaxation time. This fall in phonon relaxation time may be inserted into equation 5.3 by using a temperature dependent relaxation time. At low temperatures, the relaxation time is assumed to have a constant value. As the temperature increases, the relaxation time must decay at some rate which is controlled by a characteristic Umklapp temperature, T_u . A reasonable choice, which is used here, will be to select an exponential decay of the relaxation time τ_{pp} with temperature, thus

$$\tau_{pp} = \tau_{pp1} + d\tau_{pp} (e^{-T/T_u} - 1)$$
(5.7)

5.5.4 Effect of Different Phonon Density of States

By utilising the measured phonon density of states from neutron scattering experiments for $YBa_2Cu_3O_6$ and $YBa_2Cu_3O_7$, and using only equation 5.7 as the overall phonon relaxation time, it is possible to estimate the influence of the phonon density of states upon the thermal conductivity. Figure 5.5 shows the result obtained using the phonon density of states measured by Rhyne et al (1988) and the phonon density of states calculated by Chaplot (1988,1990). Chaplot's calculated phonon density of states for $YBa_2Cu_3O_6$ has a large peak at low energies which is not observed in any of the experimental data. This peak has the effect of further increasing the relative population of the low energy modes and increases the calculated thermal conductivity of $YBa_2Cu_3O_6$ relative to that of $YBa_2Cu_3O_7$.

It can be seen from the data that the 'bare' density of states of Chaplot produces a much 'flatter' curve owing to the higher number of low energy phonons. The experimental data of a number of groups [Rhyne et al (1988), Strobel et al (1988), Renker et al (1988)] shows that the phonon density of states for the insulating YBa₂Cu₃O₆ material is extremely similar to that of YBa₂Cu₃O₇. Rhyne et al observed that the phonon density of states undergoes only a slight renormalisation Figure 5.5 — κ computed using only Umklapp scattering. DOS --- O₆ — O₇ κ ++++ O₆ $\circ \circ \circ \circ \circ O_7$ $\tau_{pp}(\text{low T}) = 10\text{ps}, \tau_{pp}(\text{high T}) = 0.5\text{ps}, T_u = 20\text{K}$

(a) Rhyne et al (1988) 'neutron weighted' phonon DOS.





(b) Chaplot (1990) 'bare' phonon DOS.





as the temperature is decreased. Figure 5.5 shows that the phonon contribution to the thermal conductivity of $YBa_2Cu_3O_6$ is higher than that of $YBa_2Cu_3O_7$. This may be contrasted with the observed thermal conductivity which shows that $YBa_2Cu_3O_6$ has a lower thermal conductivity than $YBa_2Cu_3O_7$. The change in the phonon density of states cannot account for the large change in the thermal conductivity when oxygen is removed from the sample.

5.5.5 Phonon-carrier Scattering

As discussed in chapter II, YBa₂Cu₃O_{7- δ} which has a sufficiently high oxygen content to undergo a superconducting phase transition exhibits metallic behaviour in the normal state. From this it can be inferred that the material must contain free charge carriers which, in the normal state, will be able to scatter phonons. The phonon-carrier scattering mechanism may therefore be attributed a relaxation time, τ_{pe} . Within the confines of the relaxation time approximation this may be combined with the relaxation time for the Umklapp scattering by use of equation 5.4 thus

$$\tau = \frac{\tau_{pe}\tau_{pp}}{\tau_{pe} + \tau_{pp}} \tag{5.8}$$

As the material passes into the superconducting state, conventional models of superconductivity require that the carriers which condense into the superconducting state are unable to carry entropy and are unable to scatter phonons. If this is accepted, it follows that below T_c there will be an increase in the lifetime of the low energy phonons which are unable to break up the pairs. This process is accounted for in the model by using a τ_{pe} for phonons which have an energy $(\hbar\omega)$ above that of the superconducting energy gap which differs from τ_{pe} for phonons which have an energy $(\hbar\omega)$ below the superconducting energy gap.

$$\hbar\omega < 2\Delta \to \tau_{pe} = \tau_{pe1} * d\tau_{pe}$$
$$\hbar\omega > 2\Delta \to \tau_{pe} = \tau_{pe1}$$
(5.9)

The energy gap is temperature dependent and this dependence must be accounted for in the calculation. By measurement of the imaginary part of the conductivity of YBa₂Cu₃O_{7- δ}, Cohen et al (1987) and Porch et al (1988) have shown that the energy gap in YBa₂Cu₃O_{7- δ} has a temperature dependence which is close to the form required from BCS theory. The model does not require the BCS theory to be valid, merely that the superconducting mechanism causes the carriers to condense into a state such that the carriers are ineffective phonon scatterers. The temperature dependence of the energy gap may be approximated by the standard 'two fluid' model relation [Schoenberg (1952)]:

$$2\Delta = 3.52k_B T_c \left[1 - \left(\frac{T}{T_c}\right)^4 \right]$$
(5.10)

The calculations presented here assumed that, owing to the absence of carriers in the insulating material, the relaxation time of the phonon-carrier scattering in the insulating material is extremely long. Figure 5.6 shows the effect of adding in the electron-carrier scattering term as detailed above. As expected intuitively, the thermal conductivity of the superconducting material is depressed in the normal state by the carrier-phonon scattering. Upon passing through the superconducting phase transition there is a recovery of the thermal conductivity and it approaches the value of the insulating YBa₂Cu₃O₆ phase when the typical phonon energy falls within the superconducting energy gap. This recovery produces a peak in the thermal conductivity of the superconducting material in the normal state would be expected to be lower than that of the insulator, in sharp disagreement with experiment.

5.5.6 Point Defect Scattering

In chapter II it was stated that oxygen removed from the superconducting material is taken from the Cu-O chains rather than the Cu-O₂ planes, resulting in a tetragonal structure. As the orthorhombic $YBa_2Cu_3O_7$ structure is ordered, removal of oxygen atoms from the Cu-O chain can be considered to create a number of point defects in the material. These point defects will have the effect of acting as scattering centres for phonons and may be accounted for by the addition of a further relaxation time to equation 5.3. Klemens (1955,1969) has shown that the phonon relaxation time for scattering caused by a substitutional atom of different

Figure 5.6 — κ computed using Umklapp and carrier-phonon scattering.

Data obtained using the phonon density of states of Strobel et al (1988). $\tau_{pp}(\text{low T}) = 10\text{ps}, \tau_{pp}(\text{high T}) = 1.5\text{ps}, T_u = 20\text{K}$ $\tau_{pe} = 50,20,10,5,2,1\text{ps}$. The lower curve corresponds to $\tau_{pe} = 1\text{ps}$.



mass to a lattice composed of atoms of mass M is

$$\frac{1}{\tau} = \frac{a^3}{4\pi v^3} \left(\frac{\Delta M}{M}\right)^2 \frac{\omega^4}{G} \tag{5.11}$$

where v is the phonon group velocity given by linear dispersion ($\omega = qv$), G is the number unit cells in a volume containing one defect, a is the lattice constant and ΔM is the difference in mass between the solute and solvent atoms. This expression is arrived at by consideration of the perturbation energy of the lattice vibrations of a perfect crystal and may be thought of as simply an expression of Rayleigh scattering. The phonons scatter elastically and independently off point defects which are smaller than the phonon wavelength, leading to the characteristic ω^4 dependence. The expression may be applied to YBa₂Cu₃O_{7- δ} by treating the oxygen atom as a mass defect sited in an array of atoms of some 'average' mass M, and replacing a^3 by the volume of the unit cell. The various oxygen concentrations may now be treated, to a first approximation, by altering the number of unit cells per defect in equation 5.11. If YBa₂Cu₃O₇ is considered to be fully ordered then $G = \infty$, for O_{6.5} G=2, for O_{6.7} G=3.3 etc. YBa₂Cu₃O₆ has a fully ordered tetragonal structure, hence O_{6.3} is modelled with G=3.3 as it may be expected to have the same degree of disorder as O_{6.7}.

Inclusion of the point defect scattering term in equation 5.3, using equation 5.4, produces the results shown in figure 5.7. The addition of this term reduces significantly the thermal conductivity at higher temperatures. Raising the temperature of the material will result in an increased 'average' energy of the phonon modes. As the collision rate increases strongly with phonon energy, point defect scattering will become more important at high temperatures. The computed curves show that the effect of point defect scattering caused by oxygen vacancies is most effective at depressing the thermal conductivity for an oxygen content of 6.5, yet at this oxygen concentration the sample should still undergo a superconducting transition. The ordered states YBa₂Cu₃O₆ and YBa₂Cu₃O₇ remain unaffected by the presence of such a scattering mechanism. Hence increasing the concentration of oxygen defects as oxygen is removed from YBa₂Cu₃O₇ is unable to provide an adequate explanation of the observed thermal conductivity.
Figure 5.7 — κ computed using Umklapp and point defect scattering.

Data obtained using the phonon density of states of Strobel et al (1988). $\tau_{pp}(\text{low T}) = 10\text{ps}, \tau_{pp}(\text{high T}) = 1.5\text{ps}, T_u = 20\text{K}$

0	G = 2	(x = 6.50)	$+ \ldots G = 3.3$	(x = 6.70, 6.30)
Δ	G = 5	$(\mathbf{x} = 6.80, 6.20)$	$\diamond \ldots G = 10$	(x = 6.90, 6.10)
۵	G = 20	$(\mathbf{x} = 6.95, 6.05)$	* G = 100	(x = 7.00, 6.00)



5.6 Boundary Scattering

Further scattering mechanisms may be present within the material which will affect the thermal conductivity. One possible effect is that of boundary scattering; phonons will scatter at grain boundaries. Such scattering is important at low temperatures when the phonons have a large mean free path. The thermal conductivity limited by grain boundary scattering for the sample used in this study will not change as the oxygen is removed unless a radical re-arrangement of the grains takes place. The vacuum annealing temperatures used were very low, much less than the melting point of YBa₂Cu₃O_{7- δ} or any impurity phases, hence a drastic change in microstructure is highly unlikely. If the microstructure of the material had altered radically as the oxygen was removed, one must still explain the increase in the thermal conductivity when the sample was re-oxygenated.

It may be argued that a second boundary scattering mechanism exists which could explain the thermal conductivity data - scattering from twin boundaries. As the oxygen content of the material is reduced the material undergoes an orthorhombic to tetragonal phase transition. The occurrence of the phase transition gives rise to the formation of twin boundaries, such that there is 90° re-orientation of the local a direction. The data of Jorgensen et al (1990) indicates that material of oxygen content < 6.4 is tetragonal, consequently for the insulating material no twin boundaries should be present. If twin boundary scattering is important, one may only justifiably introduce this scattering mechanism for materials with $\delta < 0.6$ i.e. the materials which undergo a superconducting transition. The addition of a scattering mechanism cannot result in an increase in the phonon relaxation time hence there will be either no change or a reduction in the computed thermal conductivity. Again this can only lead to the situation where the thermal conductivity of the superconducting material would be lower than that of the insulating material. Scattering from twin boundaries cannot explain the reduction in thermal conductivity as the oxygen is removed from the sample.

5.7 Electronic Contribution

One important contribution to the thermal conductivity has been neglected in the model up to this point, the electronic contribution. YBa₂Cu₃O₆, being an insulator and therefore containing few free charge carriers will not be expected to have a significant carrier contribution to the thermal conductivity. This may be contrasted with $YBa_2Cu_3O_7$ which has a low resistivity and therefore has many free charge carriers.

5.7.1 Wiedemann-Franz Law

As discussed briefly in §2.6 the Wiedemann-Franz law [Wiedemann and Franz (1853)] provides one method of estimating the electronic contribution to the thermal conductivity of YBa₂Cu₃O_{7- δ} in the normal state.

$$\kappa_e = \frac{L_0 T}{\rho} \tag{5.12}$$

For material D970/1 whose normal state resistivity is approximately $0.4m\Omega cm$, the electronic contribution will be approximately $0.28 \text{ Wm}^{-1}\text{K}^{-1}$. This is less than 10% of the overall thermal conductivity.

It is not clear that the Wiedemann-Franz law should be valid at all for a material which undergoes a transition to a state where the electrons are highly correlated. The Wiedemann-Franz law assumes that the electrons scatter elastically hence both the charge (the carrier of 'electricity') and the energy (the carrier of 'heat') of the electron are conserved and the contribution to the electrical and thermal current possesses the same relaxation time. If the electrons scatter inelastically, charge will be conserved whereas the electron energy state will be altered and the Wiedemann-Franz law will fail.

For electrons which are scattered by phonons, inelastic collisions will occur when the 'average' phonon (determined from the consideration of the thermal energy and the density of states) has a wave-vector close to the Brillouin zone boundary. At very high temperatures (such that k_B T is much greater than the maximum phonon energy $\hbar\omega_{max}$), the electron scattering by phonons is effectively elastic and the Wiedemann-Franz law will hold. From the phonon density of states of YBa₂Cu₃O₇, $\hbar\omega_{max}/k_B \sim 900$ K, thus one would not expect the Wiedemann-Franz law to hold at the temperatures under observation in this study. Hence a different approach must be adopted to compute the carrier contribution to the thermal conductivity.

5.7.2 Model for Electron Scattering

In order to model the electronic contribution to the thermal conductivity it is assumed that the contribution will follow the general form of equation 5.3.

$$\kappa_e = \frac{1}{3} C_v v^2 \tau_e \tag{5.13}$$

where C_v is the electronic specific heat capacity, v is the 'typical' electron velocity which will be of the same order of magnitude as the Fermi velocity, and τ_e is the relaxation time for electron scattering processes. The electronic contribution to the heat capacity may be obtained from the heat capacity of the free electron gas. [see, for example, Ashcroft and Mermin (1976)]

$$C_{v} = \frac{\pi^{2}}{3} k_{B}^{2} T g(\epsilon_{F})$$
(5.14)

where $g(\epsilon_F)$ is the electronic density of states at the Fermi energy. Substituting for the various parameters, it can be shown that the the thermal conductivity is given by

$$\kappa_c = \frac{\pi^2}{3} \frac{k_B^2 T n_c}{m} \tau_e \tag{5.15}$$

where n_c is the number of 'normal state' carriers. At temperatures above T_c , it is assumed that the number of such carriers is constant whereas below T_c the number of such carriers will fall as the electrons condense into the superconducting state.

$$n_c = n_n - n_s(T) \qquad T < T_c$$

= $n_n \qquad T \ge T_c$ (5.16)

where $n_s(T)$ is the number of electrons which have condensed into the superconducting state at a temperature T.

As in the case for the phonon contribution, the relaxation time in equation 5.15 provides the mathematical description of the remaining physics of the problem. To a first approximation the relaxation time may be written in terms of the relaxation times of electron-electron and electron-phonon 'collisions'.

$$\tau_e = \frac{\tau_{ee}\tau_{ep}}{\tau_{ee} + \tau_{ep}} \tag{5.17}$$

It is further assumed that the relaxation time τ_{ep} is the same as that appropriate for phonons scattered by electrons τ_{pe} , which increases below T_c according to equation 5.9. An estimate for the functional form of τ_{ee} is now required.

At low temperatures few electrons will be excited to a level above the Fermi energy ϵ_F . If one considers an electron in an energy state ϵ above the Fermi surface, then as effectively only electron states below the Fermi surface are occupied any electron-electron scattering will occur by interaction with an electron which lies within a depth ϵ below the Fermi surface. The Pauli exclusion principle requires that the two electrons scatter only into unoccupied states – which are above the Fermi surface. It can be demonstrated [see, for example, Ashcroft and Mermin (1976)] that this leads to a scattering rate which varies as T², the relaxation time for such processes is thus represented in this calculation by the functional form

$$\tau_{ee} = \frac{\alpha}{T^2} \tag{5.18}$$

where α is some constant.

A complete model for the electronic contribution to the thermal conductivity must account for the increase in τ_{ee} and τ_{ep} as the temperature falls. At sufficiently low temperatures the electron-electron scattering relaxation time will be very large and the electron-phonon scattering will dominate. This may be contrasted with the phonon case where the phonon-electron scattering relaxation time becomes large and the phonon-phonon relaxation time is dominant.

The number of electrons which condense into the superconducting state may be described by the functional form of the 'two fluid' model. [Schoenberg (1952)]

$$n_{\theta}(T) = n_n \left[1 - \left(\frac{T}{T_c}\right)^4 \right]$$
(5.19)

As the number of normal state carriers estimated from 5.19 falls rapidly below T_c , estimation of the electronic contribution to the thermal conductivity of YBa₂Cu₃O_{7- δ} simply by the application of the above equations does not lead to sufficient enhancement of the thermal conductivity to overcome the depression of the thermal conductivity caused by the addition of the electron-phonon scattering.



101

In the above discussion, the relaxation time for electron-electron scattering is assumed to be independent of the number of electrons which are 'available' for scattering processes. However, this assumption may not be valid. As the electrons condense into the superconducting state there will be a reduction in the population of 'normal' electrons, if a large number of carriers have condensed into the superconducting state the probability of a collision with another 'normal' electron will be reduced and there will be an enhancement of the electron-electron relaxation time. It may be postulated that this physical description is represented mathematically by a simple power law. The relaxation time relevant to electronelectron scattering is then assumed to be (in ps)

$$\tau_{ee} = \frac{\alpha}{T^2} + \beta (T - T_c)^{\rho} \qquad \dot{T} < T_c$$
$$= \frac{\alpha}{T^2} \qquad T \ge T_c \qquad (5.20)$$

The choice of the parameters α , β and ρ is not entirely arbitrary. The parameters must be chosen such that the electron-electron relaxation time is physically reasonable. Typical relaxation times for electron-electron scattering are of the order of 10^{-14} seconds. Thus α , β and ρ must be chosen such that the overall electron-electron relaxation time is of this order.

Utilising only the phonon contribution to the thermal conductivity, the data obtained upon sample D970/3 was fitted using the parameters $\tau_{pp}(\text{low T}) = 35\text{ps}$, $\tau_{pp}(\text{high T}) = 1.5\text{ps}$, $T_u = 30\text{K}$, G = 20. Figure 5.8 shows calculated and experimental thermal conductivity curves. Utilising these parameters for the phonon contribution, the electronic contribution was then added to see if the model was capable of fitting the experimental data. Figure 5.9 shows the results obtained by performing the calculation with the addition of equations 5.8 and 5.13. The parameters used to fit the data were $\tau_{pe} = \tau_{ep} = 2.5\text{ps}$, $d\tau_{pe} = 1000$, $\alpha = 60$, $\beta = 4\times10^{-8}$, $\rho = 4$. The number of carriers per unit volume in the normal state was assumed to be $3\times10^{27}\text{m}^{-3}$, which is comparable to values found in the published literature. [Fischer et al (1988)]

The parameters relating to the electron-electron relaxation time lead to the relaxation time at 100K being 0.006ps. It is important to note that the additional

Figure 5.8 — Comparison of theoretical and experimental data for sample D970/3. $\tau_{pp}(\text{low T}) = 35\text{ps}, \tau_{pp}(\text{high T}) = 1.5\text{ps}, T_u = 30\text{K}, \text{G} = 20$



Figure 5.9 — Comparison of theoretical and experimental data for sample D970/4.

 $\tau_{pp}(\text{low T}) = 35\text{ps}, \tau_{pp}(\text{high T}) = 1.5\text{ps}, T_u = 30\text{K}, \text{G} = 20$ $\tau_{pe} = \tau_{ep} = 2.5\text{ps}, d\tau_{pe} = 1000, \alpha = 60, \beta = 4\text{x}10^{-8}, \rho = 4.$



term in the relaxation time is a parameterisation of a physical idea. There are a variety of theoretical ideas which may be able to explain the magnitude of the apparent carrier contribution to the thermal conductivity, some involving the idea that the carriers are bosons, holons or spinons, rather than the fermion approach adopted in this analysis. [see, for example, Mott (1990), Anderson and Ren (1990)]

Thermal conductivity measurements may be able to provide unique information about the nature of the normal state carriers below T_c . It has been suggested that as little as 10% of the normal state carriers eventually condense into the superconducting state. [Mott (1990)] Any theory which espouses this type of idea must therefore take into account the behaviour of the normal state carriers below T_c . Thermal conductivity measurements may be able to assist in the testing of such new theories.

5.8 Conclusions

Thermal conductivity measurements have been performed upon a sample of $YBa_2Cu_3O_{7-\delta}$ which has been subjected to a series of heat treatments in order to remove oxygen from the material. The measurements show that the peak in the thermal conductivity of the sample disappears as oxygen is removed and the sample becomes non-superconducting. Oxygen annealing of the sample restored the superconducting transition.

In order to assist the understanding of this behaviour, a quantitative model based upon the Boltzmann equation has been been constructed. The quantitative model demonstrates that consideration of phonon interactions alone cannot account for the behaviour of the thermal conductivity and that there must be a significant carrier contribution to the thermal conductivity in both the normal and superconducting states.

The electronic contribution estimated from ideas based around the Pauli exclusion principle has been shown to be unable to explain the large carrier contribution to thermal conductivity with the sample in the superconducting state. A physical process has been suggested which may account for the large carrier contribution below T_c . This process has been parameterised in order to produce quantitative results. The parameters required by the theory are bound by certain physical constraints. Consequently there are no entirely free parameters which can modify the theoretical fit to an arbitrary accuracy. The agreement of experiment and theory is good and it is possible that a more sophisticated refinement of the parameters may lead to an even closer correlation between experiment and theory.

Chapter VI

Thermal Conductivity Study II Effect of Sintering Temperature

A state without the means of some change is without the means of its conservation. — Edmund Burke. Reflections on the Revolution in France

6.1 Sample Preparation

The samples used in this study were produced from the same batch of calcined material used for the production of sample D970 described in §5.2. Equal masses of material were weighed and pressed into six 20mm diameter pellets at a pressure of 405MPa (4kbar). Each pellet was sintered at a slightly different temperature, D950 being sintered at 950°C, D960 sintered at 960°C and so on up to D1000. The furnace temperature was controlled using a Eurotherm 818P temperature controller fitted with a solid state relay which allowed the power input to the heating element to be continuously varied. The stability of the furnace quoted by the manufacturers is $\pm 2^{\circ}$ C. This may be used as an estimate of the error on the sintering temperature. The pellets were annealed under flowing oxygen at a temperature of 450°C for 48 hours and cooled slowly at 20°C/hr. All samples were stored in a desiccator when not in use. The samples were found to be approximately 85% of the expected density of single crystal YBa₂Cu₃O₇.

6.2 X-ray Powder Diffraction

X ray powder diffraction was performed upon each of the six samples using the X-ray powder diffractometer described in §3.1. The results show that the materials are all primarily YBa₂Cu₃O_{7- δ} with possible trace quantities of Y₂BaCuO₅. From the observed reflections the lattice parameters of the YBa₂Cu₃O_{7- δ} phase were calculated using a standard least squares fitting algorithm, these values are given in table 6.1. The large error bounds upon the *a* and *b* lattice parameters reflect the initial uncertainty of the *c* axis measurements and the low number of strong reflections available from which these parameters may be determined.

Sample	a (Å)	b (Å)	c (Å)	
D950	3.83 ± 0.02	3.88 ± 0.02	11.68 ± 0.01	
D960	3.84 ± 0.02	3.86 ± 0.03	11.69 ± 0.01	
D970	3.84 ± 0.02	3.87 ± 0.03	11.70 ± 0.01	
D980	3.85 ± 0.02	3.85 ± 0.03	11.69 ± 0.01	
D990	3.85 ± 0.03	3.86 ± 0.03	11.66 ± 0.01	
D1000	3.87 ± 0.03	3.87 ± 0.03	11.68 ± 0.01	

Table 6.1 — Structural parameters of samples D950 to D1000

6.3 Electrical Resistivity Measurements

The resistivity as a function of temperature and magnetic field was obtained for each of the six materials. The resistivity of the samples in the normal state did not alter as the applied magnetic field was increased. Figure 6.1 shows the temperature dependence of the resistivity (in zero field) for each of the samples. As the sintering temperature was increased above 950°C, the resistivity decreased slightly, reaching a minimum for sample D970. At sintering temperatures above 970°C the resistivity of the material increased considerably. None of the materials exhibited semiconducting behaviour.

The variation in the superconducting transition temperature, $T_c(50\%)$, in magnetic fields up to 500mT, is plotted in figure 6.2. Following the trend of the absolute



resistivity, sample D970 had the highest transition temperature in low field, 91.9K. The transition temperature of each sample was found to be approximately independent of applied field in the region 0 to 50mT although $T_c(50\%)$ did fall by approximately 0.8K as the field was increased to 500mT. In contrast to the behaviour of the remaining samples, D990 showed a significant depression of $T_c(50\%)$ in very low applied fields.

The width of the superconducting transition, $\Delta T_c(90\%-10\%)$, in the various magnetic fields is plotted in figure 6.3. As expected, the transition becomes broader at progressively higher applied fields, reflecting the granular nature of the samples. The variation of $\Delta T_c(90\%-10\%)$ as the sintering temperature is raised is particularly small, showing a slight sharpening of the transition as the sintering temperature is raised.

The resistivity transitions are expected to be affected significantly by two competing effects. The sintering was performed at high temperatures, above and below the initial calcination temperature, consequently it is possible that the samples have undergone some partial decomposition. This would have the effect of producing particularly 'dirty' grain boundaries which have an off-stoichiometric composition. At higher sintering temperatures one would expect the inter-granular connections to become stronger and this would tend to decrease the overall resistivity. The low resistivity at 100K of samples D950, D960 and D970 suggests that these materials have a high overall oxygen content and are likely to have particularly 'clean' grain boundaries. The increase in resistivity for samples D980, D990 and D1000 is indicative of the materials being 'degraded' by the sintering process, resulting in either a lower volume fraction of superconducting material or a decrease in the number of paths of low electrical resistance.

The small changes in T_c and ΔT_c as a magnetic field is applied to each sample is indicative of strong inter-granular coupling throughout each material. The very small changes in T_c and ΔT_c with applied field for all of the samples is curious. If samples D980 to D1000 have suffered from a larger number of impurities at the grain boundaries than samples D950 to D970, one might reasonably expect that the superconducting transition of sample D1000 should be much more broad than that of D950, with the difference increasing with applied field. One must

Figure 6.2 — $T_c(50\%)$ of samples D950 to D1000.

0	<0.1 mT	$+ \ldots 0.5 mT$	$\Delta \dots 5 mT$
0	50mT	• 500mT	



Figure 6.3 — $\Delta T_c(90\%-10\%)$ of samples D950 to D1000.

• <0.1mT	$+ \ldots 0.5 mT$	$\Delta \dots 5 mT$	

◇ ... 50mT □ ... 500mT



note, however, that electrical resistivity in superconducting systems is dominated by the percolation threshold [Shante and Kirkpatrick (1971)] consequently the resistivity provides information only about the strongest set of links which provide a continuous path through the material. Qualitative prediction of the depression of T_c and the broadening of the transition is difficult. For example, although sample D990 shows a lowering of T_c in low applied fields (unlike the other samples), sample D1000 may not show this effect simply because it contains a few very 'strong' superconducting paths – sample D1000 may have fewer percolative paths than sample D990, but each path may be able to support a greater electrical current than those of sample D990.

6.4 Magnetisation Measurements

Magnetic hysteresis curves were obtained at 77K and 40K using the experimental apparatus described in §3.4. Figure 6.4 shows the magnetisation curve of sample D980 at 40K. All of the magnetisation curves measured at a given temperature were similar in form, differing only in detail as the sintering temperature was raised. Table 6.2 indicates the main parameters which could be ascertained from each magnetisation curve.

	Parameter	Sample					
		D950	D960	D970	D980	D990	D1000
	$B_{irr}(\mathrm{T})$	0.89	0.82	0.96	0.86	1.10	0.65
77K	M $(J/T/kg)$ at 1.15T	0.218	0.216	0.248	0.176	0.203	0.173
	Peak M (J/T/kg)	0.943	1.030	1.054	0.962	1.150	0.920
40K	Peak M (J/T/kg)	4.92	5.28	4.30	4.42	4.38	4.88

Table 6.2 — Magnetisation curve parameters of samples D950 to D1000.

All samples exhibited strong diamagnetism at very low fields. At fields above B_{c1} the diamagnetic signal decreased as flux entered the sample. In all cases, at high magnetic fields(>0.6 Tesla) the magnetisation reached a value which was approximately field independent. This is characteristic of an extreme type II superconductor in the regime $B_{c1} \ll B \ll B_{c2}$. At 77K the magnetisation exhibited





irreversible behaviour below a field B_{irr} . The value of B_{irr} for samples D950 to D980 was similar; sample D990 had the highest value of B_{irr} whereas sample D1000 had the lowest value. The magnitude of B_{irr} is indicative of the strength of the screening currents a sample may sustain. An increase in B_{irr} may be achieved by an increase in the strength of the inter-grain coupling ('connectivity') such that the critical current, I_c , of the grain boundaries is increased. This is a possible explanation of the high value of B_{irr} of sample D990. Sample D1000 would be expected also to benefit from an increase in the connectivity, however, as indicated by the resistivity measurements, the effect of increased connectivity may be offset by a corresponding degradation of the sample.

At fields above B_{irr} the sample still exhibits a substantial diamagnetic behaviour, indicating that a significant part of the sample is in the superconducting state. Ando and Akita (1990) have suggested that as flux may freely flow within the sample, the magnetisation above B_{irr} is not dominated by bulk screening currents, hence the value of the magnetisation above B_{irr} provides an indication of the quantity of superconducting material. At 77K, the materials showed a slight decrease in the magnetisation at 1.15 Tesla as the sintering temperature was raised, although the value for sample D970 was significantly higher value than for the other materials. This trend is in broad agreement with the hypothesis that the samples degrade at sintering temperatures above 970°C and that sample D970 has a larger volume fraction of superconducting material than the other samples.

From the magnetic hysteresis, the transport critical current density of each material was determined by application of the Bean critical state model using the sample outer dimension as the scaling length (see §3.4). Figure 6.5 illustrates the variation of $J_c(B)$ with sintering temperature at 77K. At 77K, no discernible trend in $J_c(B)$ could be found as the sintering temperature increased, however the values of $J_c(B)$ for sample D990 were significantly higher than for the other samples, reflecting the higher value of B_{irr} found for this sample. Figure 6.6 illustrates the variation of $J_c(B)$ with sintering temperature at 40K. At this temperature $J_c(B)$ decreased as the sintering temperature was raised from 950°C to 970°C, and increased as the sintering temperature rose from 980°C to 1000°C.

Figure 6.5 — Bean model $J_c(B)$ of samples D950 to D1000 at 77K.



Figure 6.6 — Bean model $J_c(B)$ of samples D950 to D1000 at 40K.



An increase in hysteresis may be explained in two ways. Firstly, if higher $J_c(B)$ material contains a higher quantity of superconducting material, there will be a greater apparent flux expulsion from the material, leading to a higher magnitude of the magnetisation. If, in addition to this, there are are many sites which can strongly pin flux most of the flux lines within the sample will be pinned. Hence upon reducing the applied field the sample will exhibit a higher hysteresis than a material with a lower quantity of superconducting material. A second possible explanation of an increase in the hysteresis is that in the lower $J_c(B)$ material, there are insufficient pinning sites to pin the flux lines. In this latter case, a higher number of pinning sites of sufficient strength will lead to an increase in the observed hysteresis. It is not possible to state unequivocally which of these two explanations is appropriate for the samples in this study.

6.5 Optical Microscopy

Optical microscopy was performed on each of the specimens used for the resistivity measurements. The sample surfaces were prepared by abrading with fine grade emery paper followed by polishing using successive grades of Metadi diamond lapping compound (9 μ m, 6 μ m, 3 μ m and 1 μ m). The polished surfaces were washed with methanol. The samples were viewed under an optical microscope using polarised light reflected from the sample. No evidence for the green, Y_2BaCuO_5 phase was found. The optical micrographs obtained are shown in figures 6.7 and 6.8. Moving the sample underfocus verified that the dark, out of focus patches visible in the micrographs were voids in the material. The presence of voids is consistent with the porosity of each sample although some voids may have arisen by grains being pulled out of the material during the preparation of the surface. The small quantity of light coloured material may be attributed to $BaCuO_2$. The grain size varies widely throughout each sample, making the determination of an 'average' grain size difficult. However, it is clear that no significant variation in grain size or distribution occurs as one progresses from sample D950 to D1000. The optical micrographs show that, despite the high sintering temperature, no dramatic change in microstructure takes place.

























6.6 Thermal Conductivity Measurements

The thermal conductivity of samples D950 to D1000 was measured using the experimental apparatus described in chapter IV. The most significant source of uncertainty in the values obtained was the accuracy to which the dimensions of each sample could be measured, in particular the separation of the thermocouple wires (approximately 0.3mm in 6mm). For a series of measurements performed upon the same sample, a measurement error of this sort is reflected equally in all measurements and is less important than for measurements made upon different samples. When all of the uncertainties are considered, the thermal conductivity results given in this chapter are estimated to be within 8% of the true value.

Figure 6.9 illustrates the variation in thermal conductivity as a function of temperature for each of samples D950 to D1000. All of the curves show an increase in the thermal conductivity upon passing through the superconducting transition, reaching a peak at approximately 45K before falling sharply. The samples show a general increase in the normal state thermal conductivity as the sintering temperature is raised (see table 6.3). In contrast to the trend in the normal state, the thermal conductivity of the material at 60K is much more scattered, remaining very roughly constant with increasing sintering temperature, although the value for sample D990 is significantly higher than all of the others. Upon further cooling, the thermal conductivity of each sample reaches a maximum at approximately 45K, again there is a general increase in the value of this 'peak' thermal conductivitywith sample D990 having the highest value.

	D950	D960	D970	D980	D990	D1000
$\kappa(\text{peak})$	5.8	4.6	6.1	5.0	7.1	6.2
κ(100K)	4.1	3.2	4.3	3.8	5.0	4.9
κ(60K)	5.3	4.3	5.1	4.1	6.7	5.1

Table 6.3 — Thermal conductivity of samples D950 to D1000 at selected temperatures.

A simple explanation of the thermal conductivity data is made difficult by the effect of a change in sample connectivity. The data of chapter V suggests



Figure 6.9 — Thermal conductivity of samples D950 to D1000.

that regions of the sample which exhibit a superconducting transition will have a higher thermal conductivity, above and below T_c , than any regions of the sample which are oxygen deficient. One might expect that an increase in the normal state thermal conductivity, and the presence of an enhancement of the thermal conductivity below T_c , would suggest that a larger quantity of the material has a high oxygen content. However, as discussed previously, the strength of the intergranular coupling is expected to increase as the sintering temperature is raised. Unlike the electrical and magnetic data, the thermal conductivity will probe both the superconducting and non-superconducting regions of the material and will not be dominated by the percolation threshold. In the absence of a significant increase in the connectivity, analysis of these results using only the information obtained from the measurements of chapter V would suggest that samples D990 and D1000 have a much higher volume fraction of superconducting material than samples D950 to D980. This is in stark contrast to the picture presented by the other characterisation techniques which indicate that sample D1000 has degraded and has a relatively low volume fraction of superconducting material. This information suggests that an explanation of the thermal conductivity data will be incomplete without at least a rough estimate of the effect of increasing the inter-granular coupling.

6.7 Model for Connectivity

An elementary simulation of the effect of increasing the strength of the intergranular coupling has been produced. The computer simulation is based upon the assumption that a polycrystalline sample may be described as a matrix of individual grains. This matrix of grains is simulated by constructing a network of electrical conductances, one electrical conductance corresponding to the thermal conductance between two grains (see figure 6.10). The temperature of each grain is represented in the simulation by a node voltage and the heat flow between grains by an electrical current.

As the temperature of the sample is cooled below T_c , an increasing number of the inter-granular links will adopt a value appropriate to the superconducting state, this may be simulated by altering the ratio of the number of 'superconducting' links to 'normal' links. At temperatures much less than T_c , all of the links will be in





the superconducting state, whereas above T_c , all links will be in the normal state. Obviously the conductance of the links will vary with temperature and will differ in the superconducting and normal state values. This effect may be modelled, very approximately, from the experimental values of the thermal conductivity in both the normal state and at temperatures well below T_c . In the normal state the links were assigned a conductance which was constant with temperature, whereas the conductance of the superconducting links was described by a simple linear function.

The connectivity of the simulation is determined by eliminating a certain number of conductances, mimicing the effect of weak inter-granular coupling. Thus a connectivity of 0.9 is modelled simply by removing 10% of the total number of links. By keeping the number of node voltages (and therefore grains) constant and altering the number of links available for current flow, the effect of altering the connectivity may be modelled.

Once each network was constructed, the overall conductance was determined by imposing a voltage drop across two opposite sides of the network. Kirchoff's laws for electrical current were applied to each node in turn and the electrical potential of each node altered such that the net current flow into the node was zero. Alteration of the each node voltage was repeated until a self-consistent solution was found. In order to produce a realistic model of the material it is obviously necessary to use a large conductance network. The solution of large matrix equations (up to 50x50x50 nodes) using a reasonable amount of computing time is non-trivial. A detailed discussion of the solution of such equations is given in appendix B.

The results of the simulation for different values of the connectivity are shown in figure 6.11. As the connectivity of the system is decreased, the thermal conductance of the network is lowered in the normal and superconducting states and the peak in the thermal conductivity data becomes less distinct. The results of the simulation show that an increase in the connectivity of 20% will cause a corresponding increase in the normal state thermal conductivity, yet the magnitude of the thermal conductivity peak will not appear significantly sharper. The peak in the thermal conductivity only becomes 'washed out' when the connectivity is

Figure 6.11 — Effect of decreasing connectivity.

Connectivity parameters

0	1.00	+ 0.90	Δ 0.80
0	0.70	□ 0.60	* 0.50



reduced dramatically, hence it is reasonable to suppose that the increase in the normal state thermal conductivity of samples D990 and D1000 may be caused in part by the increase in the connectivity of the material, rather than a dramatic increase in the volume fraction of superconducting material.

6.8 Conclusions

A qualitative analysis of the relationship between the electrical resistivity, transport critical current density and thermal conductivity has been performed. Quantitative analysis of the thermal conductivity data is extremely difficult owing to the presently poor state of the understanding of the influence of the various processes which contribute to the thermal conductivity of a polycrystalline sample. However, by utilising the complementary information from the electrical resistivity and magnetic measurements, a reasonably coherent picture begins to emerge from the data. As the sintering temperature increases, there is an increase in the strength of the inter-granular coupling. This effect is most pronounced at sintering temperatures above the initial calcination temperature of the precursor materials. The increase in the connectivity manifests itself most clearly in the thermal conductivity data.

Chapter VII

Thermal Conductivity Study III Cookson Group Materials

The question which science typically asks is not 'What is it reasonable beforehand to suppose ?' but rather 'What have we evidence to think is actually the case ?' — John Polkinghorne.

One World

7.1 Introduction

A number of samples prepared by Cookson Group Central Research were made available for study. The purpose of this investigation was to identify whether or not the thermal conductivity of the material would be a useful characterisation tool, and whether it is possible to extract information about the superconducting nature of the material which is unobtainable from other, more conventional characterisation methods. The work presented in this chapter was conducted under the supervision of Dr T.P.Beales at Cookson Central Research, Oxford. Some of the material preparation and characterisation work was performed by Mr J.Rogers and the Cookson Analytical Laboratory.

7.2 Material Preparation

 $YBa_2Cu_3O_{7-\delta}$ powder was prepared by Cookson Central Research using a solid state reaction route from precursor oxides. Preliminary analysis of this calcined powder was conducted by Cookson Analytical Services. X-ray powder diffractometry indicated that the material consisted of primarily $YBa_2Cu_3O_{7-\delta}$, with $BaCuO_2$ and Y₂BaCuO₅ present at the 1% to 2% level. X-ray fluorescence analysis revealed that the material deviated slightly from optimum stoichiometry, the ratio Y:Ba:Cu ratio being 1:1.949:2.848 i.e. the material was approximately 2.5% barium deficient and 5% copper deficient. Before sintering, the calcined powder was ground and the particle size determined by sedigraph analysis to be $\simeq 2\mu m$ in diameter.

Samples of the calcined powder were weighed out and pressed into 19mm diameter pellets using a stainless steel die and hydrostatic bench press. Three pellets were sintered for 4 hours, in air, at nominal furnace temperatures of 950°C, 970°C and 990°C, followed by a slow cool at 60°C/hr to room temperature, these samples were labelled HM2-950, HM2-970 and HM2-990 respectively. The sintered pellets were heated to 400°C in flowing oxygen for 4 hours and slowly cooled to room temperature. The resulting pellets had a density of 90% of the theoretical density of single crystal YBa₂Cu₃O₇. X-ray powder diffraction experiments at Durham indicated that the the high temperature of the sintering process had not significantly altered the composition of any of the pellets, and that they were still primarily YBa₂Cu₃O_{7- δ}.

7.3 Electrical Resistance Measurements

The superconducting transition temperature of each sample was monitored by electrical resistance measurements which were performed by Mr J.Rogers at Cookson Central Research using a closed cycle cryostat. Each sample had approximately the same electrical resistance at room temperature $(1.3m\Omega)$. Figure 7.1 shows the normalised electrical resistance as a function of temperature in the vicinity of the superconducting transition. Sample HM2-950 had a very broad transition with an onset temperature of 92K, not reaching the zero resistance state until approximately 81K. As the sintering temperature was increased the superconducting transition sharpened considerably.

As discussed in $\S6.3$, the resistance transition is dominated by percolation effects such that the electrical current will flow through the path with the least electrical resistance. Consequently the sharpening of the resistance transition of the samples as the sintering temperature is increased may be due to *either* an increase in the connectivity of material with the 90K transition temperature *or* an increase in the actual volume of the high transition temperature superconducting material.





The electrical resistance transition alone cannot be explain unambiguously the change in the properties of the material as the sintering temperature is raised.

7.4 Magnetisation Measurements

Magnetic hysteresis curves were obtained for each sample at 72K using the apparatus at Durham described in §3.4. As expected, all of the samples showed partial flux penetration and irreversible behaviour at low fields (see figure 7.2) and no evidence of ferromagnetic impurities. Utilising the Bean critical state model described in §3.4 the intra-grain critical current density was obtained from the magnetic hysteresis and the sample dimensions. The critical current density of the material at 72K, as a function of applied magnetic field up to 0.6 Tesla is shown in figure 7.3.

Sample HM2-950 had the lowest critical current density of the three samples. As the sintering temperature was increased, the magnetic hysteresis and therefore the critical current density increased. In §6.4 it was noted that this behaviour can be explained in two ways: if, as the sintering temperature increases, there is a larger flux expulsion due to a change in the amount or distribution of superconducting material and if a large fraction of the flux lines are pinned, then the measured hysteresis at low fields is dominated by the magnitude of the flux expulsion. As an increase in the number of pinned flux lines also leads to a larger hysteresis, an increase in the pinning site density will also increase the apparent critical current density.

	HM2-950	HM2-970	HM2-990
Peak M (J/T/kg)	0.58	0.77	1.24
M(J/T/kg) at 1.1 T	0.109	0.140	0.263
<i>B_{irr}</i> (T)	0.82	0.94	1.15

Table 7.1 — Magnetisation curve parameters for HM2 samples.

The value of B_{irr} obtained for the samples decreased as the sintering temperature was raised (table 7.1). This trend was followed by the magnetisation at 1.1 Tesla (a field well above B_{irr}). Above B_{irr} , each material still exhibits a strong








diamagnetic signal which suggests that some of the material is in the superconducting state, despite the ability of flux to flow freely through the sample. Ando and Akita (1990) claim that that the magnetisation above B_{irr} is indicative of the quantity of the superconducting material present, although this information relates only to sections of the sample which have $B_{c2}(T)$ above that of $B_{irr}(T)$. Upon lowering the temperature of the sample below T_c , thermal depinning of the flux lines will become less effective (allowing a greater transport current to flow) and $B_{c2}(T)$ of each region of the sample will increase. At temperatures well below T_c , these two effects combine to increase B_{irr} beyond the limits of presently available experimental equipment. Magnetisation data may be able to provide some indication of the volume fraction of superconducting material close to T_c , however such data cannot be obtained from magnetisation data at temperatures well below T_c and in low magnetic fields.

7.5 Thermal Conductivity Measurements

The thermal conductivity of the samples as a function of temperature in the region 30K-100K is shown in figure 7.4. The most striking feature of the curves is that over the full temperature range the thermal conductivity of HM2-990 is considerably higher than that of the samples sintered at lower temperatures.

HM2-970 shows has only a very slight rise in the thermal conductivity at about 80K (corresponding to the zero resistance state). The curve for the HM2-950 is similar to HM2-970, with a shallow peak suggesting that only a small fraction of the material undergoes the transition to the superconducting state. The observed rise in the thermal conductivity is most pronounced in HM2-990, there is also evidence of a small secondary peak at 45K in this sample. Whilst this peak was reproducible on this sample, no evidence for the existence of such a peak has been observed in any of the other materials measured, either supplied by Cookson group, or as prepared at Durham. It is therefore unlikely that such a peak is caused by experimental error. The peak may be intrinsic to the superconducting phase of the material although the absence of such a feature on any of the other superconducting materials measured renders this explanation unlikely. A plausible explanation of the peak is that it is caused by the presence of a second phase of material or a substantial quantity of YBa₂Cu₃O_{7- δ} which has a low oxygen content. The





presence of low oxygen content $YBa_2Cu_3O_{7-\delta}$ would be detected by neither the electrical resistance nor the magnetisation measurements.

One possible explanation of the behaviour of the thermal conductivity of the samples above and below T_c is that there is a dramatic increase in the connectivity of the material when the sintering temperature is raised from 970°C to 990°C with a concomitant increase in the amount of superconducting phase present. Such an explanation is consistent with the sharpening of the resistance transition of the material as the sintering temperature is raised. This explanation also suggests that the increase in the magnetic hysteresis of the samples as the sintering is increased is caused by the increase in volume fraction of the superconducting material providing a larger initial flux expulsion from the material.

7.6 Conclusions

The electrical resistance transition of the HM2 samples was observed to sharpen as the sintering temperature of the material was raised. From the resistance data alone, it is not possible to determine whether this is caused by an increase in the volume fraction of 90K transition temperature material or by an increase in the strength of the inter-granular connections. In addition, no information about the superconducting material is obtained below T_c . By analysis of the magnetisation data it has been demonstrated that the transport critical current density increases as the sintering temperature is raised. This may be caused by an increase in the volume fraction of superconducting material, although verification of this at low fields and at temperatures well below T_c is not possible utilising magnetisation techniques.

The thermal conductivity data shows that samples HM2-950 and HM2-970 are very similar in their bulk properties and is consistent with the idea that they each sample contains a relatively small amount of superconducting material. HM2-990 has a much higher thermal conductivity than the other two samples, exhibits a much more pronounced peak when cooled below T_c , and provides evidence for a secondary peak at 45K. The thermal conductivity of HM2-990 provides evidence to suggest that the sample has a much higher volume fraction of 90K superconducting material with a higher strength inter-granular coupling. Thermal conductivity measurements have been used to provide information which complements the resistivity and magnetisation data and has been able to assist in understanding the simple physical properties of the material by providing a qualitative description of the properties of the material at a temperature well below T_c .

Chapter VIII

Conclusions and Suggestions for Further Work

The world is round and the place which may seem like the end may also be only the beginnning — Ivy Baker Priest.

'Parade'

8.1 Summary

Apparatus to measure the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ at temperatures between 20K and 120K has been designed and constructed. Measurements are performed on 20mm long bars, held under vacuum, using a longitudinal heat flow technique. A differential thermocouple is used to directly measure the temperature difference across the sample. Equilibrium times in this system are of the order of 15 minutes. The system is computer controlled thereby enabling data acquisition to occur throughout the entire cryostat 'run'.

A series of samples of $YBa_2Cu_3O_{7-\delta}$ have been prepared which have undergone sintering at slightly different temperatures. Preliminary characterisation techniques used have included electrical resistivity in applied magnetic fields, magnetisation measurements and X-ray powder diffraction. Similar measurements have been performed upon samples prepared by Cookson Central Research.

Using these complementary techniques a qualitative analysis of the relationship between the electrical resistivity, transport critical current density and thermal conductivity has been performed. As the sintering temperature increases, there is an increase in the strength of the inter-granular coupling. This effect is most pronounced at sintering temperatures above the initial calcination temperature of the precursor materials. The increase in the connectivity manifests itself most clearly in the thermal conductivity data. A complete quantitative analysis of the data would require a lengthy and exhaustive study owing to the complexity of the relationship between the oxygen content and the connectivity of the samples.

Thermal conductivity measurements have been performed upon a sample of $YBa_2Cu_3O_{7-\delta}$ which has been subjected to a series of heat treatments in order to remove oxygen from the material. The measurements show conclusively that the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ is very strongly influenced by the oxygen content of the material. A reduction of the oxygen content of the material results in a substantial lowering of the thermal conductivity.

A quantitative model, based upon the Boltzmann equation, has been constructed. This quantitative model demonstrates that consideration of the changes in phonon interactions alone cannot account for the differences in the behaviour of the thermal conductivity of YBa₂Cu₃O₆ and YBa₂Cu₃O₇. In addition, the model shows that there must be a significant carrier contribution to the thermal conductivity in both the normal and superconducting states. A physical process has been proposed which provides the required large carrier contribution below T_c . The parameterisation of this process, whilst not arbitrary, is only loosely based. Nevertheless, this model can provide a good fit to the observed thermal conductivity in both the superconducting and non-superconducting material. Improvements in the calculations and a greater understanding of the effect of free carriers below T_c could be achieved if this parameterisation was placed upon a more formal basis.

8.2 Future Work

8.2.1 Models

It is hoped that the model of the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ developed in this thesis provides some encouragement for the development of further theories which may be able to explain more fully the role of the 'normal' electrons in the transport properties of the superconducting state. Many qualitative

interpretations have been proposed to explain the behaviour of the thermal conductivity of YBa₂Cu₃O_{7- δ}. In contrast to this, the number of quantitative theories is extremely small. It seems difficult to build up a coherent picture of the major heat transport mechanisms without providing some 'hard' numerical evidence for each of the qualitative ideas proposed.

8.2.2 Experiments

The studies conducted in this thesis concentrated upon polycrystalline materials. It was stated in §2.5 that YBa₂Cu₃O_{7- δ} is highly anisotropic. Very few thermal conductivity measurements have been performed upon single crystal samples, owing largely to problems of obtaining suitable crystals. However large single crystals are becoming much more readily available. Determination of the anisotropy of the thermal conductivity, in particular with regard to the *c* axis direction, may be able to provide significant insight into the behaviour of the carriers in directions perpendicular and parallel to the oxygen chains.

The application of a magnetic field may strongly influence the behaviour of any 'normal state' carriers which may exist below T_c . Such experiments may be able to provide a reliable method of influencing the normal state heat carriers in a controllable manner, thereby yielding some useful insight into the behaviour of the materials

8.2.3 Improvements to the Experimental Apparatus

The experimental arrangement has a few minor failings with respect to the thermometry of the system. The sample clamping arrangement has proved to be adequate for the measurements obtained to date although more delicate samples may be affected by the pressure which is applied to achieve a good thermal contact. Large samples could be mounted by passing a copper bar through the sample, (in a similar manner to the thermocouple mounting). Rather than directly clamping the samples to the cold tip of the cryostat, the samples could be mounted first in a high thermal conductivity resin (such as Stycast) and firmly clamped to the cold tip. This would minimise the strain induced in the sample. The siting of the cold plate heater is not ideal. Rewiring the heater such that the heat flow is more evenly distributed over the surface of the cold plate will reduce any errors which may arise from the existence of temperature gradients across the cold plate. A second heater mounted on the central column of the sample clamp might assist the temperature regulation of the system, particularly at very low temperatures.

Radiation losses have been assumed to be small owing to the sample being sited in a can which is thermally linked to the cold plate. The use of a further radiation shield mounted on the cold plate, coupled with appropriate heating and sensing devices, will allow the temperature gradient of the radiation shield to more closely match that of the sample.

The present system of acquisition of data from the analogue nanovoltmeter is adequate but rather cumbersome as the range of the nanovoltmeter cannot be altered remotely. This problem also afflicts the constant current power supply. A programmable constant current supply and a nanovoltmeter fitted with a computer interface would provide an elegant, albeit expensive solution to these irritations.

8.3 Conclusion

A number of theories exist which attempt to elucidate the form of the interactions which give rise to the existence of superconductivity at high temperatures. The development of such theories will require an understanding of the thermal transport mechanisms above and below T_c . Electrical and magnetic measurements provide extremely useful information about the behaviour of the superconducting condensate but they are of limited use in assessing the behaviour of the normal state carriers. Information obtained from electrical and magnetic measurements is complementary to the information obtained from thermal measurements which probe quantitatively the low energy quasi-particle excitations.

Bibliography

All that mankind has done, thought, gained, or been, it is all lying in magic preservation in the pages of books. — Thomas Carlisle.

Abrikosov A.A. (1957) Soviet Phys. JETP 5 pp1174-1182

- Allen P.J. (1991) PhD Thesis, Durham University To be submitted
- Anderson P.W., Ren Y. (1990) High Temperature Superconductivity: Proceedings of a Symposium at Los Alamos ed: K.S. Bedell et al Publ: Addison Wesley ISBN 0-201-51249-1 pp3-34
- Ando Y., Akita S. (1990) Jap. Jnl. Appl. Phys. 29 ppL770-L771
- Ashcroft N.W., Mermin N.D. (1976) Solid State Physics Publ:Holt, Rhinehart and Wilson ISBN 0-03-049346-3
- Ayache C., Barbara B., Bonjour E., Calemczuk R., Couach M., Henry J.H., Rossat-Mignod J. (1987) Solid State Commun. 64 pp247-252

Bardeen J., Cooper L.N., Schrieffer J.S. (1957) Phys. Rev. 108 pp1175-1204

- Batlogg B., Cava R.J., Jayaraman A., van Dover R.B., Kourouklis G.A.,
 Sunshine S., Murphy D.W., Rupp L.W., Chen H.S., White A., Short K.T.,
 Mujsce A.M., Rietman E.A. (1987) Phys. Rev. Lett. 58 pp2333-2336
- Bayot V., Delannay F., Dewitt C., Erauw J-P., Gonze X., Issi J-P., Jonas A., Kinanay-Alaoui M., Lambricht M., Michenaud J-P., Minet J-P., Piraux L. (1987) Solid State Commun. 63 pp983-986
- Bednorz J.G., Müller K.A. (1986) Z. Phys. B 64 pp189-193
- Bednorz J.G., Müller K.A. (1988) Rev. Mod. Phys. 60 pp585-600
- Bell A.M.T. (1990) Supercond. Sci. Technol. 3 pp55-61
- Beno M.A., Soderholm L., Capone D.W., Hinks D.G., Jorgensen J.D., Grace J.D., Schuller I.K., Segre C.U., Zhang K. (1987) Appl. Phys. Lett 51 pp5759
- Berman R., Foster E.L., Ziman J.M. (1955) Proc. Roy. Soc. (London) A231 pp130-144
- Berman R. (1961) Experimental Cryophysics ed:Hoare, Jackson and Kurti Publ: Butterworths pp327-336
- Berman R., Brock J.C.F., Huntley D.J. (1964) Cryogenics 4 pp233-239
- Berman R., Brock J.C.F., Huntley D.J. (1965) Advances in Cryogenic Engineering 10 pp233-238
- Berman R. (1976) Thermal Conduction In Solids Publ: Clarendon Press, Oxford ISBN 0-19-851430
- Beyers R., Ahn B.T., Gorman G., Lee V.Y., Parkin S.S.P., Ramirez M.L., Roche K.P., Vazquez J.E., Gür T.M., Huggins R.A. (1989) Nature 340 pp619-621
- Bourne L.C., Crommie M.F., Zettl A., zur Loye H.-C., Keller S.W., Leary K.L., Stacy A.M., Chang K.J., Cohen M.L., Morris D.E. (1987) Phys. Rev. Lett. 58 pp2337-2339

Burgess S., Greig D. (1974) J. Phys. D.: Appl. Phys. 7 pp2051-2057

Carruthers P. (1961) Rev. Mod. Phys. 33 pp92-138

- Cava R.J., Batlogg B., Chen C.H., Rietman E.A., Zahurak S.M., Werder D. (1987) Nature 329 pp423-427
- Chaplot S.L. (1988) Phys. Rev. B 37 pp7435-7442
- Chaplot S.L. (1990) Phys. Rev. B 42 pp2149-2154
- Collings E.W. (1986) Applied Superconductivity, Metallurgy and Physics of Titanium Alloys ISBN 0-306-41690-5 and 0-306-41691-3
- Das Gupta A., Koch C.C., Kroeger D.M., Chou Y.T. (1978) Phil. Mag. B 38 pp367-380
- File J., Mills R.G. (1963) Phys. Rev. Lett. 10 pp93-97
- Fischer H.E., Watson S.K., Cahill D.G. (1988) Comments on Condensed Matter Physics 14 pp65-127
- Forgan E.M., Lee S.L., Sutton S., Wellhofer F., Abell J.S., Gough C.E., Cox S.F.J., Scott C.A. (1990) Supercond. Sci. Technol. 3 pp217-221
- Fruchter L., Giovannella C., Collin G., Campbell I.A. (1988) Physica C 156 pp69-72
- Gallagher W.J., Worthington T.K., Dinger T.R., Holtzberg F., Kaiser D.L., Sandstrom R.L. (1987) Physica B 148 pp228-232
- Ginsberg D.M. (1989) Physical Properties of High Temperature Superconductors Publ:World Scientific ISBN 9971-50-894-X
- Ginzburg V.L., Landau L.D. (1957) Zh. Eksp. i Teor. Fiz. 20 1064 Translation given in: ter Harr D. (1965) Men of Physics I : L.D.Landau Publ: Pergamon Press
- Ginzburg V.L. (1955) Il Nuovo Cimento 10 pp1234-1250
- Glover R.E. (1967) Phys. Lett. 25A pp542-544

- Gordon J.E., Tan M.L., Fisher R.A., Phillips N.E. (1989) Solid State Commun. 69 pp625-629
- Gorter C.J. (1964) Rev. Mod. Phys. 36 pp3-7
- Greedan J.E., O'Reilly A.H., Stager C.V. (1987) Phys. Rev. B. 35 pp8770-8773
- Hake R.R. (1967) App. Phys. Lett. 10 pp189-92
- Harshman D.R., Aeppli G., Ansaldo E.J., Batlogg B., Brewer J.H., Carolan J.F., Cava R.J., Celio M., Chaklader A.C.D., Hardy W.N., Kreitzman S.R., Luke G.M., Noakes D.R., Senba M. (1987) Phys. Rev. B 36 pp2386-2389
- Heremans J., Morelli D.T., Smith G.W., Strite S.C. (1988) Phys. Rev. B 37 pp1604-1610
- Hoon S.R., Willcock S.N.M. (1988) J. Phys. E Sci. Instrum. 21 pp772-785
- Inderhees S.E., Salamon M.B., Goldenfield N., Rice J.P., Pazol B.G., Ginsberg D.M., Liu J.Z., Crabtree G.W. (1988) Phys. Rev. Lett 60 pp1178-1180 and erratum Phys. Rev. Lett. 60 p2445
- Iye Y., Tamegai T., Takeya H., Takei H. (1987) Physica B 148 pp224-227
- Jeżowski A., Mucha J., Rogacki K., Horyń R., Bukowski Z., Horobiowski M., Rafałowicz J., Stępien-Damm J., Sułkowski C., Trojnar E., Zaleski A., Klamut J. (1987) Phys. Lett A 122 pp431-433
- Jeżowski A., Klamut J., Horyń R., Rogacki K. (1989) Supercond. Sci. Technol. 1 pp296-301
- Jorgensen J.D., Beno M.A., Hinks D.G., Soderholm L., Volin K.J., Hittermann R.L., Grace J.D., Schuller I.K., Segre C.U., Zhang K., Kleefisch M.S. (1987) Phys. Rev. B 36 pp3608-3616
- Jorgensen J.D., Veal B.W., Paulikas A.P., Nowicki L.J., Crabtree G.W., Claus H., Kwok W.K. (1990) Phys. Rev. B 41 pp1863-1877

- Kaiser D.L., Holtzberg F., Scott B.A., McGuire T.R. (1987) Appl. Phys. Lett. 51 pp1040-1042
- Kittel C. (1976) Introduction to Solid State Physics Publ: Wiley & Sons ISBN 0-471-49024-5
- Khachaturyan A.G., Semenovskaya S.V., Morris J.W. (1988) Phys. Rev. B 37 pp2243-2246
- Klemens P.G. (1955) Proc. Roy. Soc (London) A68 pp1113-1128
- Klemens P.G. (1969) Thermal Conductivity 1 pp1-68 Ed:R.P.Tye
- Konczykowski M., Rullier-Albenque F., Collin G. (1988) Physica C 153-155 pp1365-1366
- Kopp J., Slack G.A. (1971) Cryogenics 11 pp22-25
- Krusin-Elbaum L., Greene R.L., Holtzberg F., Malozemoff A.P., Yeshurun Y. (1989a) Phys. Rev. Lett. 62 pp217-220
- Krusin-Elbaum L., Malozemoff A.P., Yeshurun Y., Cronemeyer D.C., Holtzberg F. (1989b) Phys. Rev. B 39 pp2936-2939
- Lægreid T., Tuset P., Nes O.-M., Slaski M., Fossheim K. (1989)
 Proc. M²S Conference, Stanford. To be published in Physica C
- Lægreid T., Fossheim K., Sandvold E., Julsrad S. (1987) Nature 330 pp637-638
- Landolt-Börnstein (1984) New Series III/1 Publ: Springer, Berlin
- Leary K.J., zur Loye H.-C., Keller S.W., Faltens T.A., Ham W.K., Michaels J.N., Stacy A.M. (1987) Phys. Rev. Lett. 59 pp1236-1239
- Le Page Y., Siegrist T., Sunshine S.A., Schneemeyer L.F., Murphy D.W., Zahurak S.M., Waszczak J.V., McKinnon W.R., Tarascon J.M., Hull G.W., Greene L.H. (1987) Phys. Rev. B 36 pp3617-3621

London F., London H. (1935) Proc. Roy. Soc. (London) A149 pp72-88

- Maeda H., Tanaka Y., Fukutomi M., Asano T. (1988) Jap. Jnl. App. Phys. 27 ppL209-L210
- Maple M.B., Dalichaouch Y., Ferreira J.M., Hake R.R., Lee B.W., Neumeier J.J., Torikachvili M.S., Yang K.N., Zhou H., Guertin R.P., Kuric M.V. (1987) Physica B 148 pp155-162
- Matthias B.T., Geballe T.H., Compton V.B. (1963) Rev. Mod. Phys. 35 pp1-22
- Matthiessen A. (1863) Reports of the British Association for the Advancement of Science 32 pp125-161
- McGuire T.R., Dinger T.R., Freitas P.J.P., Gallagher W.J., Plaskett T.S., Sandstrom R.L., Shaw T.M. (1987) Phys. Rev. B 36 pp4032-4035
- Meissner W., Ochsenfeld R. (1933) Naturwissenschaften 21 pp787-788
- Mendelssohn K. (1964) Rev. Mod. Phys. 36 pp7-12
- Meservy R., Schwartz B.B. (1969) Superconductivity (Vol.1) ed: R.D.Parks, Publ: M.Dekker N.Y. pp117-191
- Michel C., Hervieu M., Borel M.M., Grandin A., Deslandes F., Provost J., Raveau B. (1987) Z. Phys. B 68 pp421-423
- Monod P., Dubois B., Odier P. (1988) Physica C 153-155 pp1489-1490
- Morelli D.T., Heremans J., Swets D.E. (1987) Phys. Rev. B 36 pp3917-3919
- Morris D.E., Kuroda R.M., Markelz A.G., Nickel J.H., Wei J.Y.T. (1988) Phys. Rev. B 37 pp5936-5939
- Mott N.F. (1990) Proc. LT19 Conference on High T_c Superconductors. To be published in Supercond. Sci. Technol. 4 ppS59-S66
- Müller P., Andres K., Groß F., Veith H., Hackl R. (1988) Physica C 153-155 pp421-422
- Muto Y., Kobayashi N., Watanabe K. (1990) Physica B 164 pp139-149

- Nakao K., Miura N., Tatsuhara K., Takeya H., Takei H. (1988) Phys. Rev. Lett. 63 pp97-100
- Oh B., Char K., Kent A.D., Naito M., Beasley M.R., Geballe T.H., Hammond R.H., Kapitulnik A., Graybeal J.M. (1988) Phys. Rev. B 37 pp7861-7864

Olsen J.L. (1952) Proc. Phys. Soc. (London) A65 pp518-532

- Onnes H.K. (1911) Akad. van Wetenschapen 14 113
- Parkin S.S.P., Lee V.Y., Engler E.M., Nazzal A.I., Huang T.C., Gorman G., Savoy R., Beyers R. (1988) Phys. Rev. Lett. 60 pp2539-2542
- Pellan P., Dousselin G., Cortés H., Rosenblatt J. (1972) Solid State Commun. 11 pp427-431
- Penney T., von Molnár S., Kaiser D., Holtzberg F., Kleinsasser A.W. (1988) Phys. Rev. B 38 pp2918-2921
- Philips J.C. (1989) Physics of High-T_c Superconductors Publ:Academic Press ISBN 0-12-553990-8
- Pippard A.B. (1953) Proc. Roy. Soc. (London) A216 pp547-568
- Raboutou A., Rosenblatt J., Peyral P. (1980) Phys. Rev. Lett. 45 pp1035-1039
- Rayleigh, Lord (1911) Proc. Roy. Soc. (London) A84 pp25-46
- Reese W. (1966) J. Appl. Phys. 37 pp864-868
- Reichardt W., Ewert D., Gering E., Gompf F., Pintschovius L., Renker B., Collin G., Dianoux A.J., Mukta H. (1989) Physica B 156 & 157 pp897-901
- Renker B., Gompff F., Gering E., Roth G., Reichardt W., Ewert D., Reitschel H., Mutka H. (1988) Z. Phys. B. 71 pp437-442
- Rhyne J.J., Neumann D.A., Gotaas J.A., Beech F., Toth L., Lawrence S., Wolf S., Osofsky M., Gubser D.U. (1987) Phys. Rev. B 36 pp2294-2297

- Richardson R.C., Smith E.T. (1988) Experimental Techniques in Low Temperature Physics Publ: Addison Wesley ISBN 0-201-15002-6
- Salama K., Selvamanickam V., Gao L., Sun K. (1989) Appl. Phys. Lett. 54 pp2352-2354
- Sample H.H., Brandt B.L., Rubin L.G., (1982) Rev. Sci. Instrum. 53 pp1129-1136
- Savitskii E.M., Baron V.V., Efimov Yu.V., Bychkova M.I., Myzenkova L.F. (1981) Superconducting Materials Publ: Plenum Press
- Schoenberg D. (1952) Superconductivity Chapter 5 Publ: Cambridge Univ. Press
- Schuller I.K., Hinks D.G., Beno M.A., Capone D.W., Soderholm L., Locquet J.-P., Bruynseraede Y., Segre C.U. Zhang K. (1987) Solid State Commun. 63 pp385-388
- Schuller I.K., Jorgensen J.D. (1989) MRS Bulletin Jan 1989 pp27-30
- Shante V.K.S., Kirkpatrick S. (1971) Advances in Physics 20 pp325-357
- Sheng Z.Z., Hermann A.M. (1988) Nature 332 pp55-58
- Silsbee F.B. (1916) J. Wash. Acad. Sci. 6 pp597-602
- Sridar S., Wu D.H., Kennedy W. (1989) Phys. Rev. Lett. 63 pp1873-1876
- Stokka S., Fossheim K. (1982) J. Phys. E Sci. Instrum. 15 pp123-127
- Strobel P., Monceau P., Tholence J.L., Currat R., Dianoux A.J., Capponi J.J., Bednorz J.G. (1988) Physica C 153-155 pp282-283
- Takabatake T., Ishikawa M., Nakazawa Y., Oguro I., Sakakibara T., Goto T. (1987) Jap. Jnl. App. Phys. 26 ppL978-L979
- Takeya H., Takei H. (1988) Physica C 153-155 pp413-414
- Takita K., Akinaga H., Katoh H., Uchino T., Ishigaki T., Asano H. (1987) Jap. Jnl. App. Phys. 26 ppL1323-L1325

- Taylor K.N.R., Cook P., Puzzer T., Matthews D.N., Russell G.J., Goodman P. (1988) Physica C 153-155 pp411-412
- Tinkham M. (1975) Introduction to Superconductivity Publ:R.E.Krieger, Malabar, Florida
- Tinkham M., Lobb C.J. (1989) Solid State Physics 42 pp91-133 Publ: Academic Press ISBN 0-12-607742-8
- Touloukhian Y.S., Powell R.W., Ho C.Y., Klemens P.G. (1970) Thermophysical Properties of Matter 1 pp3a-39a
- Tozer S.W., Kleinsasser A.W., Penney T., Kaiser D., Holtzberg F. (1987) Phys. Rev. Lett. 59 1768
- Träuble H., Essmann U. (1968) Phys. Stat. Sol. 25 pp373-393
- Tu K.N., Yeh N.C., Park S.I., Tsuei C.C. (1989) Phys. Rev. B 39 pp304-314
- Uemura Y.J., Emery V.J., Moodenbaugh A.R., Suenaga M., Johnston D.C., Jacobson A.J., Lewandowski J.T., Brewer J.H., Kiefl R.F., Kreitzman S.R., Luke G.M., Riseman T., Stronach C.E., Kossler W.J., Kempton J.R., Yu X.H., Opie D., Schone H.E. (1988) Phys. Rev. B 38 pp909-912
- Uher C., Kaiser A.B. (1987) Phys. Rev. B 36 pp5680-5683
- Uher C. (1989) Proc. of 3rd Annual Conference on Superconductivity and Applications Buffalo, New York
- Umezawa A., Crabtree G.W., Liu J.Z (1988a). Physica C 153-155 pp1461-1462
- Umezawa A., Crabtree G.W., Liu J.Z., Moran T.J., Malik S.K., Nunez L.H., Kwok W.L., Sowers C.H. (1988b) Phys. Rev. B 38 pp2843-2846
- Van Tendeloo G., Zandbergen H.W., Amelinckx S. (1987) Solid State Commun. 63 pp389-393
- Varga R.S. (1962) Matrix Iterative Analysis Publ:Englewood Cliffs

- Vidal F., Veira J.A., Maza J., Miguélez F., Morán E., Alario M.A. (1988) Solid State Commun. 66 pp421-425
- Vinnikov L.Y., Emelchenko G.A., Kononovich P.A., Ossipyan Y.A., Schegolev I.P., Buravov L.I., Laukhin V.N. (1988) Physica C 153-155 pp1359-1360
- Wanklyn B.M., Watts B.E., Haycock P., Pratt F., de Groot P.A.J., Rapson G.P. (1988) Solid State Commun. 66 pp441-443
- Welp U., Grimsditch M., You H., Kwok W.K., Fang M.M., Crabtree G.W., Liu J.Z. (1989) Physica C 161 pp1-5
- Werthamer N.R., Helfand E., Hohenberg P.C. (1966) Phys. Rev. 147 pp295-302
- White G.K. (1979) Experimental Techniques in Low Temperature Physics Publ:Clarendon Press, Oxford ISBN 0-19-85-1359-3
- Wiedemann G., Franz R. (1853) Annalen der Physik 89 497
- Wu M.K., Ashburn J.R., Torng C.J., Hor P.H., Meng R.L., Gao L., Huang Z.J.,
 Wang Y.Q., Chu C.W. (1987) Phys. Rev. Lett. 58 pp908-910
- Yeshurun Y., Malozemoff A.P., Holtzberg F., Dinger T.R. (1988) Phys. Rev. B 38 pp11828-11831
- Young D.M. (1971) Iterative Solution of Large Linear Systems Publ:Academic Press
- Yvon K., François M. (1989) Z. Phys. B 76 pp413-444
- Zavaritskii N.V., Samoliov A.V., Yurgens A.A. (1988a) JETP Lett. 48 pp242-245
- Zavaritskii N.V., Samoliov A.V., Yurgens A.A., Klochko V.S., Makarov V.I. (1988b) Physica C 162-164 pp562-563

Appendix A

The Boltzmann Equation

A.1 Basic Formulation

A phonon mode may be described in terms of a wave-vector, \mathbf{q} , and its polarisation, *i*. Every phonon mode \mathbf{q}_i which is excited has an energy $E(\mathbf{q}_i)$ and group velocity $\mathbf{v}_g(\mathbf{q}_i)$ associated with it. The total "heat current" due to a particular phonon mode is given by the product of the energy of the mode and the velocity of propagation. The overall "heat current" due to all modes is then

$$\mathbf{h} = \sum_{\mathbf{q}_i} N(\mathbf{q}_i) \mathbf{v}_g(\mathbf{q}_i) E(\mathbf{q}_i)$$
(A.1)

where $N(\mathbf{q}_i)$ is the number of modes with given \mathbf{q}_i and is a distribution function.

In a state of thermal equilibrium, the distribution function will be given by some $N^0(\mathbf{q}_i)$ which will depend only upon the phonon frequency. In this case, the total heat current must be zero as for each mode \mathbf{q}_i there is a further mode $-\mathbf{q}_i$ which has

$$E(\mathbf{q}_i) = E(-\mathbf{q}_i)$$
$$\mathbf{v}_g(\mathbf{q}_i) = -\mathbf{v}_g(-\mathbf{q}_i)$$

The following assumptions are now made:

- 1. Net phonon transport is in the direction of the temperature gradient.
- 2. At a time t the number of phonons in mode q_i is given by $N(q_i)$

After a time δt , the phonons which were originally in state \mathbf{q}_i will have moved a distance $\mathbf{v}_g(\mathbf{q}_i)\delta t$ and have been replaced by the phonon density appropriate to $\mathbf{v}_g(\mathbf{q}_i)\delta t$. Hence the rate of change of the distribution function, $N(\mathbf{q}_i)$, due to this drift is given by

$$\frac{\partial N(\mathbf{q}_i)}{\partial t}\Big|_{drift} = \frac{\partial N(\mathbf{q}_i)}{\partial T} \nabla T \cdot \mathbf{v}_g(\mathbf{q}_i) \tag{A.2}$$

In steady state this phonon drift must be balanced by other processes which are referred to here as scattering processes.

$$\frac{\partial N(\mathbf{q}_i)}{\partial t}\Big|_{drift} + \frac{\partial N(\mathbf{q}_i)}{\partial t}\Big|_{scatt} = 0 \qquad (A.3)$$

$$\frac{\partial N(\mathbf{q}_i)}{\partial t}\Big|_{scatt} - \frac{\partial N(\mathbf{q}_i)}{\partial T} \nabla T \cdot \mathbf{v}_g(\mathbf{q}_i) = 0 \tag{A.4}$$

This is a form of the Boltzmann equation.

A.2 Relaxation Time Approximation

The Boltzmann equation may be solved, approximately, by a variety of methods. One of the most common methods is to invoke the relaxation time approximation which makes the following assumptions:

- 1. The scattering processes restore equilibrium at a rate proportional to the departure from the equilibrium value.
- 2. The system is close to the equilibrium distribution. ie $\frac{\partial N^0(\mathbf{q}_i)}{\partial t} \simeq \frac{\partial N(\mathbf{q}_i)}{\partial t}$
- 3. The return to equilibrium is deemed to be characterised by some relaxation time, denoted τ .

Mathematically, these approximations may be expressed as

$$\frac{\partial N(\mathbf{q}_i)}{\partial t}\Big|_{scatt} = \frac{N^0(\mathbf{q}_i) - N(\mathbf{q}_i)}{\tau(\mathbf{q}_i)}$$
(A.5)

hence

$$\frac{N^{0}(\mathbf{q}_{i}) - N(\mathbf{q}_{i})}{\tau} = -\frac{\partial N(\mathbf{q}_{i})}{dT} \nabla T \cdot \mathbf{v}_{g}(\mathbf{q}_{i})$$
(A.6)

The heat capacity of a phonon mode $C(\mathbf{q}_i)$ is given by $\partial U(\mathbf{q}_i)/\partial T \equiv E(\mathbf{q}_i).\partial N(\mathbf{q}_i)/\partial T.$

$$N^{0}(\mathbf{q}_{i}) - N(\mathbf{q}_{i}) = -(\mathbf{v}_{g}(\mathbf{q}_{i}) \cdot \nabla T) \tau(q) \frac{C(\mathbf{q}_{i})}{E(\mathbf{q}_{i})}$$
(A.7)

From equation A.1

$$\mathbf{h} = \sum_{\mathbf{q}_i} -(\mathbf{v}_g(\mathbf{q}_i) \cdot \nabla T) \tau(\mathbf{q}_i) \frac{C(\mathbf{q}_i)}{E(\mathbf{q}_i)} E(\mathbf{q}_i) \mathbf{v}_g(\mathbf{q}_i)$$
(A.8)

$$h_{z} = \sum_{\mathbf{q}_{i}} -(\mathbf{v}_{g}(\mathbf{q}_{i}) \cdot \nabla T) \tau(\mathbf{q}_{i}) C(\mathbf{q}_{i}) |\mathbf{v}_{g}(\mathbf{q}_{i})| \cos \theta \qquad (A.9)$$

Taking the option of the heat flow parallel to ∇T

$$\kappa = \sum_{\mathbf{q}_i} -\mathbf{v}_g(\mathbf{q}_i)^2 \tau(\mathbf{q}_i) . C(\mathbf{q}_i) \cos^2 \theta$$
(A.10)

$$= \int_{\omega_i} -\mathbf{v}_g(\omega_i)^2 \tau(\omega_i) C(\omega_i) \cos^2 \theta d\omega_i \qquad (A.11)$$

$$= \int_{\omega} -\mathbf{v}_g(\omega)^2 \tau(\omega) \frac{(\hbar\omega)^2}{k_B T^2} \frac{e^{\hbar\omega/k_B T}}{(e^{\hbar\omega/k_B T} - 1)^2} D(\omega) \cos^2\theta d\omega \qquad (A.12)$$

Provided τ and \mathbf{v}_g are independent of θ , averaging over all available angles results in the replacement of $\cos^2\theta$ by 1/3.

Appendix B

Iterative Solution of Simultaneous Equations

B.1 Basic Concept

Consider a node at which the potential is v_{ij} . This potential should be set such that Kirchoff's law is obeyed, i.e. the net current flow into the node is equal to the sum of the externally applied currents. Thus where there are no externally applied currents, the net current flow is zero. With an iteration procedure one solves for the elements of the voltage vector by setting up equations of the form:

$$v_{ij}^{n+1} = f(v_{ij}^n)$$
 (B.1)

There are many ways of re-arranging simultaneous equations into this form, with each one having its own advantages and disadvantages. The more common types are given here, with appropriate modifications to deal with the particular problem which had to be solved.

Consider the matrix equation $\mathbf{G}.\mathbf{V} = \mathbf{C}$ where \mathbf{G} is the matrix of conductances, \mathbf{V} is the matrix of node potentials (unknown) and \mathbf{C} is the matrix of applied current. This corresponds to a series of linear simultaneous equations:

$$g_{11}v_1 + g_{12}v_2 + g_{13}v_3 + g_{14}v_4 = c_1$$

$$g_{21}v_1 + g_{22}v_2 + g_{23}v_3 + g_{24}v_4 = c_2$$

$$g_{31}v_1 + g_{32}v_2 + g_{33}v_3 + g_{34}v_4 = c_3$$

$$g_{41}v_1 + g_{42}v_2 + g_{43}v_3 + g_{44}v_4 = c_4$$
(B.2)

This set of equations may be expressed in matrix form G.V = I, where g_{ij} are the coefficients of the conductance matrix G, v_i are the coefficients of the voltage matrix V (to be found), and c_i are the coefficients of the current matrix C.

B.2 Gauss Jacobi Method

The matrix G may be broken into three matrices, a diagonal matrix (D), a lower triangular (L) and an upper triangular matrix (U).

$$\mathbf{G} = \mathbf{D} + \mathbf{L} + \mathbf{U} \tag{B.3}$$

$$\mathbf{D} = \begin{pmatrix} g_{11} & 0 & 0 & 0 \\ 0 & g_{22} & 0 & 0 \\ 0 & 0 & g_{33} & 0 \\ 0 & 0 & 0 & g_{44} \end{pmatrix} \mathbf{L} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ g_{21} & 0 & 0 & 0 \\ g_{31} & g_{32} & 0 & 0 \\ g_{41} & g_{42} & g_{43} & 0 \end{pmatrix} \mathbf{U} = \begin{pmatrix} 0 & g_{12} & g_{13} & g_{14} \\ 0 & 0 & g_{23} & g_{24} \\ 0 & 0 & 0 & g_{34} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$
(B.4)

so now

$$(\mathbf{D} + \mathbf{L} + \mathbf{U})\mathbf{V} = \mathbf{C}$$
$$\mathbf{V} = \mathbf{D}^{-1}[\mathbf{C} - (\mathbf{L} + \mathbf{U})\mathbf{V}]$$
$$\mathbf{V} = \mathbf{V} - \mathbf{D}^{-1}[(\mathbf{L} + \mathbf{U} + \mathbf{D})\mathbf{V} - \mathbf{C}]$$
(B.5)

which suggests the iteration procedure:

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \mathbf{D}^{-1}[(\mathbf{L} + \mathbf{U} + \mathbf{D})\mathbf{V}^n - \mathbf{C}\mathbf{V}^n]$$
(B.6)

 D^{-1} is easily invertible as it is a diagonal matrix, so one has a simple iterative procedure for finding the voltage vector V. For the example set of equations, they are solved by the procedure:

$$v_{1}^{n+1} = v_{1}^{n} - (g_{11}v_{1}^{n} + g_{12}v_{2}^{n} + g_{13}v_{3}^{n} + g_{14}v_{4}^{n} - c_{1})/g_{11}$$

$$v_{2}^{n+1} = v_{2}^{n} - (g_{21}v_{1}^{n} + g_{22}v_{2}^{n} + g_{23}v_{3}^{n} + g_{24}v_{4}^{n} - c_{2})/g_{22}$$

$$v_{3}^{n+1} = v_{3}^{n} - (g_{31}v_{1}^{n} + g_{32}v_{2}^{n} + g_{33}v_{3}^{n} + g_{34}v_{4}^{n} - c_{3})/g_{33}$$

$$v_{4}^{n+1} = v_{4}^{n} - (g_{41}v_{1}^{n} + g_{42}v_{2}^{n} + g_{43}v_{3}^{n} + g_{44}v_{4}^{n} - c_{4})/g_{44} \qquad (B.7)$$

This is the Gauss-Jacobi method. Despite its simplicity this method is not often used as it is usually far too slow. A simple modification to this method, known as the Gauss-Siedel method leads to a great increase in speed.

B.3 Gauss Siedel Method

With this method one uses the 'new' values of v_i^n as they are computed. For example, in the equations above when v_2^{n+1} is to be computed the value of v_1^{n+1} would be inserted instead of v_1^n . This produces a speed increase of the order of 50%. For the example set of equations, one therefore solves with the procedure:

$$v_{1}^{n+1} = v_{1}^{n} - (g_{11}v_{1}^{n} + g_{12}v_{2}^{n} + g_{13}v_{3}^{n} + g_{14}v_{4}^{n} - c_{1})/g_{11}$$

$$v_{2}^{n+1} = v_{2}^{n} - (g_{21}v_{1}^{n} + 1 + g_{22}v_{2}^{n} + g_{23}v_{3}^{n} + g_{24}v_{4}^{n} - c_{2})/g_{22}$$

$$v_{3}^{n+1} = v_{3}^{n} - (g_{31}v_{1}^{n} + 1 + g_{32}v_{2}^{n} + 1 + g_{33}v_{3}^{n} + g_{34}v_{4}^{n} - c_{3})/g_{33}$$

$$v_{4}^{n+1} = v_{4}^{n} - (g_{41}v_{1}^{n} + 1 + g_{42}v_{2}^{n} + 1 + g_{43}v_{3}^{n} + 1 + g_{44}v_{4}^{n} - c_{4})/g_{44} \qquad (B.8)$$

In matrix form this is represented:

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \mathbf{D}^{-1}[\mathbf{L}\mathbf{V}^{n+1} + \mathbf{U}\mathbf{V}^n + \mathbf{D}\mathbf{V}^n - \mathbf{C}\mathbf{V}^n]$$
(B.9)

Thus the 'correction' to the value of \mathbf{V}^n is

$$\mathbf{D}^{-1}[(\mathbf{L}\mathbf{V}^{n+1} + \mathbf{U}\mathbf{V}^n + \mathbf{D}\mathbf{V}^n - \mathbf{C}\mathbf{V}^n]$$
(B.10)

Note that although V^{n+1} appears on both sides of the equation, the fact that on the right hand side it is multiplied by L, ensures that the equation is consistent.

B.4 Simultaneous Over Relaxation

The method of simultaneous over relaxation (SOR) follows the same principles as the Gauss-Siedel method. However in this case an attempt is made to accelerate towards the solution. Effectively an 'over-correction' in the iterative procedure is made by multiplying the correction term by an acceleration parameter, ω . The solution procedure is now

$$\mathbf{V} = \mathbf{V}^n - \omega \mathbf{D}^{-1} [\mathbf{L} \mathbf{V}^{n+1} + \mathbf{U} \mathbf{V}^n + \mathbf{D} \mathbf{V}^n - \mathbf{C}]$$
(B.11)

Varga (1962) has shown that the SOR method is only convergent for $0 < \omega < 2$. In order to obtain the optimum value of ω for a particular system, the solution must already be known. As the boundary conditions of the problem determine the value of ω , once an optimum ω has been found for a particular system, this value will be the same for all similar problems.

B.5 Chebyshev Acceleration

Chebyshev acceleration takes the SOR method one step further. This system makes use of the idea that although there is one particular value of ω which is optimal when close to the solution, this may not be the best value to choose when far away from the solution. To understand how this works one must look at the rate of convergence of the Gauss-Siedel method. In fact it is easier to consider the Gauss-Jacobi method and follow on from there.

In the Gauss-Jacobi method, the new vector V is found by multiplying the old vector by $D^{-1}(L+U)$ and then adding on a constant. This matrix has eigenvalues which indicate by what factor the modulus of the eigenvalue of the residual vector is suppressed. Clearly these eigenvalues must be less than 1 otherwise the residual will increase. The rate of convergence is therefore controlled by the slowest decaying eigenmode (the largest eigenvalue of $D^{-1}(L+U)$). This value is called the spectral radius ρ_s , in general it will increase as the matrix size increases. Varga (1962) has shown that the spectral radius for the Gauss-Siedel iterative process is simply the square of ρ_s for the Gauss-Jacobi method. He also demonstrated that the optimal choice of ω is given by

$$\omega = \frac{2}{1 + \sqrt{(1 - \rho^2)}}$$
(B.12)

and that the spectral radius for SOR is

$$\rho_{SOR} = \left(\frac{1-\rho}{1+\sqrt{1-\rho^2}}\right)^2 \tag{B.13}$$

However the value of ρ must still be known before the formula is applied, this involves knowing the solution! As ω is related to ρ , the value of the parameter ρ (not ω) was varied and an optimal one found, this was then used for all calculations utilizing that particular matrix size.

Chebyshev acceleration begins the iteration procedure with a particular value of ω (usually 1) and then update this value after each iteration. The particular

implementation used here is due to Young (1971). The matrix G is subdivided into both 'black' (B) and 'white' (W) points, rather like a chess board. The 'black' points are updated first, followed by the 'white' points. In the implementation used here, the matrix G is sub-divided as follows:

$$\begin{pmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{pmatrix} = \begin{pmatrix} W & B & W & B \\ B & W & B & W \\ W & B & W & B \\ B & W & B & W \end{pmatrix}$$
(B.14)

With the matrix sub-divided, one now uses one acceleration parameter, ω_n , for the 'black' points, and a different one, ω'_n , for the 'white' points. With Chebyshev acceleration these parameters are allowed to vary, according to the formulae:

$$\omega_{1} = 1$$

$$\omega_{n} = (1 - \frac{1}{4}\omega'_{n-1}\rho^{2})^{-1}$$

$$\omega'_{1} = (1 - \frac{1}{2}\rho^{2})^{-1}$$

$$\omega'_{n} = (1 - \frac{1}{4}\omega_{n}\rho^{2})^{-1}$$
(B.14)

Thus as $n \to \infty$ the ω tend towards the value

$$\omega_{\infty} = \frac{2}{1 + \sqrt{1 - \rho^2}} \tag{B.15}$$

In the final implementation of the computer program, the optimum value of ρ was found empirically. The optimum values for ρ used were

No of nodes	matrix order	no of iterations	ρ
10x10x10	1000 x 1000	28	0.984
20x20x20	8000 x 8000	38	0.9954
30x30x30	27000 x 27000	70	0.9982
40x40x40	64000 x 64000	90	0.9993

Appendix C

Program to Model Thermal Conductivity

The program is written in FORTRAN 77 and conforms to the ANSI standard. Functions and subprograms are utilised wherever they may assist with the clarity of structure. Global data is passed to each subroutine or function by use of COM-MON blocks which are by necessity large. Meaningful variable names are used throughout the program wherever possible. The routines to perform the numerical integration utilise Simpson's rule and are adapted from the set of routines to be found in Press et al (1988).

NOILY NOL

After P. 160 to End.

•

医子子 医子宫 化合金 化合金 化合合合合合合合合合合合合合合合合合合合合合合合合合合合合合	Call DEFAC
	WRITE (6,*)'Calculated DOS prefactors '
GRAM TREPH	
***************************************	CALL DSIMP (MMIN, MMAX, RVAL) PRIMT * ' 'Done numerical pormalisation. RVAL = '. RVAL
	RNORM = 3. / RVAL
t of parameter names and meanings	U
	WRITE (6,*)'Do you wish to save the density of states ?'
phonom irequency M _ whome framemory at start of internation	In PORMAT (1x) ANS
M - puonou involuency at end of integration	IF ((ANS .EO. 'Y') .OR. (ANS .EO. 'V')) THEN
	WRITE (6,*)'Enter the filename to save DOS to ? '
PP = phonon phonon collision time in 10e-12 seconds	READ (5, 80) DOSFIL
PP1 = effective phonon phonon collision time at low T	20 FORMAT (1A)
1PP2 = effective phonon phonon collision time at high T	OPEN (8, FILE-DOSFIL)
P = difference between TAUPP1 and TAUPP2	WRITE (8, *) 'Density of states for'
	IF (TC .GT. 0) THEN
122 = electron phonon collision time in JUS-12 Seconds 1291 - climetron channes collicion electrons	WAITE (8,")'UXYGEN / (SUDERCONDUCTING)' Eter
1641 - Gigetton phonon cullision time iot unpelet sier clone PD - change in Alectron change time halow emergy men	MBTTR (R 4)/(runnen 6 (americanduction))
And Interna actor parts included increases int phrases - an	END IF
/ = pi / lattice parameter	1, 7
- phonon group velocity	MRITE (8, *) //
	30 CONTINUE
	WRITE (8, *)'Phonon energy (meV)'
	WRITE (8,*)'DOS (arbitrary units)'
EUSION NEIM(30), WHT(30), MPCM(30)	
ENSION ALIM (30), AHT (30), APOM (30)	DO 40 RL - MMIN, MAXK, WYEE
RENSION BLIM (30), BHT (30), BPOM (30) Rensions at any (30), 1/101	MAITE (6,") RL, DOS(RL) / RVAL 40
RENSTON ALADI. P(30) Penston Aladi. P(30)	TO CONTROCT NRTTE (6.*)'Done DOS'
ENSION TEPLO(10), TPPHI(10), TTU(10), TEPI(10), DTTEP(10)	JI ONS
ENSION ALPHA(10), BETA(10), RHO(10)	U
RACTER*12 FILOUT, DOSFIL	WRITE (6,") Do you wish to use the electron terms Y/N ?
RACTER ¹ I ANS, ELTERM	READ (5,50) ELTERM
RACTER ⁺ 12 TIMER	50 PORMAT (IA)
RACTER'IG DATE	
nom / Block/ Milth, Wat, Brow, Nilms Josef / P. S.	
num / Fakk/ A, B unu / Bibe/ e an ubib byb biv	(bararatatatatatatatatatatatatatatatatata
RVM FEANS/ 1, 10, 20045, AVM, AVM, AVV NON /PANS/ PAUPPI. DTPP. TAURPI. DTEP. TU	
MOR / NS/ MMTH. MAAY. MSTEP	<pre>[IRITE (6.*)'Enter the data as requested.'</pre>
MON /DEFECT/ HM, G, DM, ACUBED	WEITE (6,*)'You will be prompted for data so that you may'
MON /LADDER/ NLAD, V, NLAD	WRITE (6,*)'overlay up to 6 graphs.'
	WRITE ($6, +$)'If you do not wish to enter any more data'
ur = 6.25g-34 / (2*3.141592654)	WRITE (6,*)'enter a negative relaxation time'
1 = 1.30E-23	WRITE ($6, *$)'Enter 0 to use the previous value.'
/ = 3.141592654 / 3.85	υ
BED = 3.85E-10 * 3.85E-10 * 11.67E-10	NUMBER = 1
- 0.312	
L SETDOS	WRITE (0, m) Enter the phonon-phonon relaxation time',
	T . SIC TOM 1. 7.
***************************************	IF (TPPLO(NUMBER) . EO. 0) TPPLO(NUMBER) = TPPLO(NUMBER - 1)
ate the prefactors for the DOS	IF (TPPLO(NUMBER) LE. 0) GO TO 70
te the integrals of DOS and normalise them	
no degrees of freedom)	WRITE (6,*)'Enter the phonon-phonon relaxation time',
rrite to disc if required	1 'at high T 2'
<u></u>	READ (5,*) TPPHI (NUMBER)

•

U

υ

$ \begin{array}{c} \label{constraints} (a) \ (b) \ (c) \ (c$	Tri (s, v) (s) (s) (s) (s) (s) (s) (s) (s) (s) (s	U) TEFHI(NUMBER) = TEFHI(NUMBER - 1) 0) GO TO 70 WRITE (7,*)' Therma	Conductivity)'
0. Final State 0. Final Final Final Final Final Final Fi	(1) (camperature at which Umlapp processes',	
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	Circle State Circle State<	ALL CODE ON C.	医弗里氏试验检尿解液的 医化化化 化化化化化化化化化化化化化物 医化物医尿体 医单体 化化丁二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二乙二
Unitable Unita	(1, 2) (2, 2)<	$\frac{1}{1+1} \int \frac{1}{1+1} \int \frac{1}$	Ing the calculation a.12 a tettettettet
(1) (1) <td>(4) (4)<td>60 T0 70 C 44444444444 T IN</td><td>V ****************</td></td>	(4) (4) <td>60 T0 70 C 44444444444 T IN</td> <td>V ****************</td>	60 T0 70 C 44444444444 T IN	V ****************
10.1 Texture the select confision of a mark of a mark of a select confision of a mark of a selec	(*) (electron-phonon relaxation time for',	***************************************
1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	10.1 The Anti-Name of Sector	:one? '	
0.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	(5.) 'The intermediation of the interval of t		e defect coefficient G'
(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	(1, 2): Test the induced in the inducty op 1: (1, 2): Test the induc) TELI(NUMBER) = TEPI(NUMBER) = 1) Di TELI(NUMBER) = TEPI(NUMBER) = 1)	r #=6.5' / 6 6'
(4) '1 Control (1 = monty just) (4) '1 Control (1 = monty just) (4) '1 Control (1 = monty just) (5) '1 Control (1 = monty just) (1) Control (1 = monty just) (2) Control (1 = monty just) (5) '1 Control (1 = monty just) (1) Control (1 = monty just) (2) Control (1 = monty just) (5) '1 Abbalance (1 = monty just) (1) Control (1 = monty just) (2) Control (1 = monty just) (1) '1 Abbalance (1 = monty just) (1) Control (1 = monty just) (2) Control (1 = monty just) (1) '1 Abbalance (1 = monty just) (1) Control (1 = monty just) (2) Control (1 = monty just) (1) '1 Abbalance (1 = monty just) (1) Control (1 = monty just) (2) Control (1 = monty just) (1) '1 Abbalance (1 = monty just) (1) Control (1 = monty just) (2) Control (1 = monty just) (1) Control (1 = monty just) (2) Control (1 = monty just) (2) Control (1 = monty just) (2) Control (2) Contr	(4): (1): (1): (1): (1): (1): (1): (1): (1	y oc oc phance in e-bk relaxation time' MRITE (6.*)/d=3.3 €	r zm6.3.6.7 (approx)'
(5.) (5.)	(5,) There excritions (5,) There excrit	of energy less than the energy gap ? ' WRITE (6,*)'G=10000	for x=6,7'
True for the former in the	TTER (FORMER) 1.0.0000 - 0.0001 TTER (FORMER) 0.000000 (5.4) KERNE (FE A/TY 2) 7.00000 (5.4) KERNE (FE A/TY 2) 7.000000 (5.4) KERNE (FE A/TY 2) 7.0000000 (5.4) KERNE (FE A/TY 2) 7.00000000 (5.4) KERNE (FE A/TY 2)	() READ (5,*) G	
Transmitter Transmitter Transmitter Transmitter (1) Transmitter Transmitter Transmitter (2) Transmitter Transmitter Transmitter	(5, 1) Trans (1000) Trans (1000) (5, 1) Trans (1000) Trans (1000) <td>0) DTTEP (NUMBER) = DTTEP (NUMBER - 1) TC = TC * 0.08614</td> <td></td>	0) DTTEP (NUMBER) = DTTEP (NUMBER - 1) TC = TC * 0.08614	
C. Y. A. Manualy and the stand of the st	(x, y) Justic = TPP_2(LOD) (x, y) Justic = TPP_2(TD) (y, y) Justic = TPP_2(TD) (y, y) Justic = TPP_2(TD) (y, y) Justic = TPP_2(TD)	0) GO TO 70 DO 100 LOOP = 1, NU	BER
Constant	Characterization Construction Construction Characterization Constructin Construction		
(4, 1) (5, 0)	Device a control of the control of t	V TIZPEA (NUMBER) – ALPEA (NUMBER) – 1) TADPA (NUMBER) – ALPEA (NUMBER – 1)	
(4, 1) These confictions for N(10-1) The confictions for N(10-1) 20, 100,000,000,000,000,000,000,000,000,0	(\$,') Exer. conficient for K(Torent), Standards 120, 00 TO 70 Standards 120, 00 TO 70 Standard		, 202
(5.) Instructions() Truction() (5.) Instruction() Truction() (5.1)	(5.) The structure (100) (5.) <	ficient for b/ (Tc-T) '	8
The intermediation is the intermediatis the intermediatio	TXX (UNMERX) . 20. 0) EXA (UNMERX - 1) TO = TT(1000) TXX (UNMERX) . 20. 0) EXA (UNMERX - 1) TO = TT(1000) TXX (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA(LOOP) (5.*) EXA (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA(LOOP) (5.*) EXA (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA(LOOP) (5.*) EXA (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA(LOOP) EXA (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA(LOOP) EXA (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA(LOOP) EXA (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA(LOOP) EXA (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA(LOOP) EXA (UNMERX) . 20. 0) EXA (UNMERX - 1) TXX = ALMA (TXX = ALMA, TXX = ALMA (TXX = ALMA, TXX = ALMA (TXX = ALMA	(acor) 43410 - 4310	
(5, 1) TU - TU, 0) TU - TU, 0) TU - TU, 0) (5, 1) Exercised TU - TU, 0) TU - TU, 0) (5, 1) Exercised Exercised Exercised (5, 1) Exercised	(5.) Ref = rest condition (5.) Ref = rest (coop) (6.) Ref = rest (coop) (7.) Ref = rest (rest () BETA(NUMBER) = BETA(NUMBER - 1) TU = TTU(LOOP)	
(5, 1) There confictions for (T-T) ⁻¹ ((5, 1) There confictions for (T-T) ⁻¹ ((5, 1) Nonumeran ((5, 1) Nonumeran ((1)	(5, *) Bart a confisient for (Te-T) ¹ * (All = Alle)(DOP) (5, *) Bart a confisient for (Te-T) ² * (All = Alle)(DOP) BG (NUMBER) . T. 0) CO 70 70 * EXA(LOP) BG (NUMBER) . T. 0) CO 70 70 * EXA(LOP) BG (NUMBER) . T. 0) CO 70 * EXA(LOP) BG (F = 2, 1) DG (NUMBER) . T. 0) CO 70 * EXA(LOP) R = 100 * EXA) GO TO 70 TU = TU = 0.08614	
(5,1) Non (number) Set = SETALLOOP) (6,1) Non (number) Non (number) (1,1) Non (number) Non (number) <	(5,1) BOO(UNDEER) - 1) EFT = BETA (LOOD) UNDERER 1. T. 0) CO TO 70 ENE = 32.7 UNDERER 1. T. 0) CO TO 70 ENE = 3.15-3 UNDERER 1. T. 0) CO TO 60 ENE = 3.12-3 UNDERER 1. T. 0) CO TO 60 ENE = 3.12-3 UNDERER 1. T. 0) CO TO 60 ENE = 3.12-3 UNDERER 1. T. 0) CO TO 60 ENE = 9.12-3 UNDERER 1. T. 0) CO TO 60 ENE = 9.12-3 UNDERER 1. T. 0) CO TO 60 ENE = 0.06614 UNDERER 1. T. 0) CO TO 60 ENE = 0.06614 UNDERER 1. T. 0) CO TO 60 ENE = 0.06614 UNDER 1. T. 0) CO TO 60 ENE = 0.06614 UNDER 1. TO TOTO 1. 10 C COMPACIANCE CONTINUESD UNDER 1. 10 C CONTINUESD UNDER 1. 10 C CO TO	fficient for (Tc-T)^n ' ALPHA(LOOP)	
0 (UNRERS), Z0, () (BO (UNRERS) - BO (UNRES)	UG (NUMERER). J. () RO (NUMERER) = NO (NUMERER) - 1) MA = NUMER. J. () OO TO 70 A = NUMER. J. () OO TO 70 A = NUMER. J. () OO TO 60 MA = A = NUM. A A = NUMER. J. () OO TO 60 MA = A = NUM. A A = NUMER. J. () OO TO 60 A = NUM. A A = NUMER 1 A	BET - BETA(LOOP)	
0.0(numers)X. 0) 00 TO 70 00 NE = 227 0.0 (numers)X. 0) 00 TO 70 00 NE = 227 0.0 (numers)X. 0) 00 TO 70 00 NE = 227 0.0 (numers)X. 0) 00 TO 70 00 NE = 227 0.0 (numers)X. 0) 00 TO 70 00 NE = 227 0.0 (numers)X. 0) 00 TO 70 00 NE = 227 0.0 (numers)X. 0) 00 TO 70 00 NE = 227 0.0 (numers)X. 0) 00 TO 70 00 NE = 727 0.0 (numers)X. 0) 00 TO 70 00 NE = 724 0.0 (numers)X. 0) 00 TO 70 00 NE = 724 0.0 (numers)X. 0) 00 TO 70 11 requested 1.1 requested 11 requested 1.1 requested 11 requested 1.1 requested 11 requested 1.1 requested 12 requested 1.1 requested <	MRE 3.227 MRE -9.12-31 MRE -9.12-3 MRE -9.12-3 MRE -9.12-3 MRE -12-3 MRE -12-4 MRE -12-4 MR	REIG (NUMBER) = REIG (NUMBER - 1)	
A = WORER + 1 MAGE + 1 MARE - 1 MARE - 1	R = NUMERK + 1 R = Start NGERK - 1 R = NUMERK + 1 NGERK - 1 R = Start NGERK - 1 R = NUMERK - 1 NGERK - 1 <	60 10 10 C 2010	
MARK * 1 Constant	UBER: I.K. 6) GO TO 60 EPRE: F(RE*5.14*3.14*3.14*3.14*3.14*3.14*3.14*3.14*3		
Operation to disc New Y O 00 Test = Tr(Y, 100614 An enumers	NOE Real = 7: C D 0 0 FM = TMIN, TATE UNERS. EQ. 1 00 FO 60 D 0 0 FM = TMIN, TATE D 0 0 FM = TMIN, TATE UNERS. EQ. 1 00 FM = TMIN, TATE D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE OF INFORMETION D GLAC D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE OF INFORMETION D GLAC D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE OF INFORMETION D GLAC D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE OF INFORMETION D GLAC D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE OF INDOR D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE OF INDOR D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE OF INDOR D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE TATE D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE TATE D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE TATE D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE TATE D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE TATE D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE TATE D 0 FM = TMIN, TATE D 0 FM = TMIN, TATE TATE D 0 FM = TMIN, TATE <td>00 60 EPREF = (NUE-3.14*</td> <td>.14*RKB*RKB*1E-12) / (3*RME)</td>	00 60 EPREF = (NUE-3.14*	.14*RKB*RKB*1E-12) / (3*RME)
MERR. 50. 1) 60 TO 60 7 = WIN, TAX, 15TE A = WOMER 1 0 90 TEM = TMIN, TAX, 15TE A = WOMER 1 0 90 TEM = TMIN, TAX, 15TE A = WOMER 1 0 90 TEM = TMIN, TAX, 15TE A = WOMER 1 0 90 TEM = TMIN, TAX, 15TE A = WOMER 1 0 90 TEM = TMIN, TAX, 15TE A = WOMER 1 0 90 TEM = TMIN, TAX, 15TE A = WOMER 1 0 90 TEM = TMIN, TAX, 15TE A = WOMER 1 0 90 TEM = TMIN, TAX, 15TE (6, 1) FLOOT 1 frequenced (6, 1) FLOOT 1 frequenced (7, 1) FLEFTLOOT 1 frequence (7, 1) FLEFTLOOT 1	0.0644 TCK = TC / 0.0644 R = NUNGER = 1 0.0644 A = NUNCHARDON Compute electronic contribution (5,9) FLIDON (5,9) FLIDON (5,9) FLIDON (5,9) FLIDON (5,9) FLIDON (5,9) FLIDON (7,9) FLIDEPELDON (7,9) FLIDEPELDON	REMAX = 0	
<pre>A = WOMERA = 1 A = WOMERA = 1 F = TRM = 0.06614 F = TRM = 0.06614 (5,9) FLOUT (5,9) FLOUT (5,9) FLOUT (5,9) FLOUT (5,9) FLOUT (5,9) FLOUT (5,9) FLOUT (5,9) FLOUT (7,9) FLO (7,9) FLO (7,9) FLO (7,9) CAL (7,9) CAL (7,10) CAL (7,10) CAL (7,10) CAL (7,10) CAL (7,1</pre>	R = NUMERR - 1 D0 00 TEM = TMIN, TWAX, TSTEP der information to disc T = TEM = 0.06614 der information to disc C compute electronic contribution (6,1)*Inter the filaname to ave data to 7' C compute electronic contribution (6,1)*Inter the filaname to ave data to 7' C compute electronic contribution (1,6)*Inter the filaname to ave data to 7' C compute electronic contribution (1,7)*ILEFILOUT) If (2,0, THEN) (1,7)*ILEFILOUT) TAUE = ALPH / (TEW TEN) (1,7)*ILEFILOUT) TAUE = TAUE = TAUE + I(TCK - TEN) (1,7)*ILEFILOUT) TAUE = TAUE + TAUE (1,7)*ILEFILIE TAUE = TAUE + TAUE (1,7)*	TCK = TC / 0.08614	
<pre>be: ifformation to disc f(s')'Encar the filense to avve data to ?' (s')'Encar the filense to avve data t</pre>	der informetion to diac 00 00 TEM = TEM, TAM, TSTEP der informetion to diac 1,500 FILOUT (6,1) "Enter the filename to ave data to 7' 1,500 FILOUT (6,1) "Enter the filename to ave data to 7' 1,500 FILOUT (7,0) FILOUT 1,500 FILOUT (7,0) FILOUT 1,500 FILOUT (7,1) FILE=FILOUT) 1,712 FILE=FILOUT) (7,1) FILE=FILOUT) 1,712 FILE (7,1) FILE=FILOUT) 1,712 FILE (7,1) FILE=FILOUT) 1,712 FILE (7,2) FILE 1,712 FILE <		
<pre>Mar information to disc (6,*)*Enter the fileness to ave data to 7 (7,FileFiloUT) (7,FileFileFileFileFileFileFileFileFileFile</pre>	dr information to disc compute electronic contribution (5,*)*Enter the filename to mave data to 7 f. fequented (5,*)*Enter the filename to mave data to 7 f. fequented (5,*)*Enter the filename to mave data to 7 f. fequented (5,*)*Enter the filename to mave data to 7 f. (ELTERM .ED. Y) THEM (5,0) FILOUT f. (ELTERM .ED. Y) THEM (5,0) FILOUT f. (ELTERM .ED. Y) THEM (1,1) f. (ELTERM .ED. Y) THEM (1,1) f. (ELTERM .ED. Y) THEM (1,2) f. (ELTERM .ED. Y) THEM (1,2) f. (ELTERM .ED. Y) THEM (1,2) f. (F. CE. TC) THEM (1,2)	DO 90 TEM = TMIN, 	WAX, TSTEP
<pre>C compute electronic contribution (5,1'Enter the filename to ave data to 7' (3,00) FILOT (1,00) FILOT (1,1) (</pre>	<pre>Compute electronic contribution (6,*)*Enter the filename to ave data to 7 (1,*)*ILB=FILOUT (7,*)*TLB=FILOUT (7,*)*TLB=FI</pre>		
<pre>(6,*)*Encer the filoneme to ave data to 7' (7,FILE=FILOUT) 7 (1, FILE=FILOUT) 7 (1, FILE=FILOUT) 7 (1, FILE=FILOUT) 7 (7, FILOUT) 7 (7, FILE=FILOUT) 7 (7, FILOUT) 7</pre>	<pre>(6,*)'Enter the filenese to ave data to 7' (1,60) FILOUT T (1,1) (7,FILE=FILOUT) (7,FILOUT) (7,F</pre>	transferteresessessessessessessessessessessessess	ribution
<pre>(6,')'Enter the filences to ave data to ?' (',File=FILOUT (',File=FILOUT) (',File=FILE=FILOUT) (',File=FILE=FILE=FILOUT) (',File=FILE=FILE=FILOUT) (',File=FILE=FILOUT) (',File=FILE=FILE=FILOUT) (',File=FILOUT) (',File=FILE=FILE=FILOUT) (',File=FILE=FILE=FILE=FILE=FILE=FILE=FILE=FILE</pre>	<pre>(6, *) 'Exter the filename to avve data to 7 ' (*, *) 'Extern the filename to avve data to 7 ' (*, *) 'Extern the filename to avve data to 7 ' (*, *) 'FILE=FILOUT) (*, *) 'FILOUT) (*, *) 'FILOUTER) (*, *) 'FILOUT) (*, *) 'FILOUT) (*, *) 'FILOUTER) (*, *) 'FILOUTER) (*, *) 'FILOUT) (*, *) 'FILOUTER) (*) 'FILOUT) (*, *) 'FILOUT) (*, *) 'FILOUT) (*, *) 'FILOUT) (*) 'FILOUT) (*, *) 'FILOUT) (*) 'FILOUTI) (*, *) 'FILOUT) (*) 'FILOUT) (*, *) 'FILOUT) (*) 'FILOUT) (*) 'FILOUT) (*) 'FI</pre>	C if requested	
[3, 00) FILOUT If (LITERM - EQ. 'Y') THEM [7, 11.3] If (1.3) [7, 11.3] If (1.3) [7, 11.3] TAUEE = 2 ELEE = 2 FAUEE = 2 TAUEE = 150 TAUEE = 2 TAUEE = 2 TAUE = 150 TAUE = 2 TAUE (7.*) TAUE (7.*)	T (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1	filename to mave data to ? '	
<pre>IF (T. GZ. TC) THEN TAUEZ = ALEN / (TEN*TEN) ERRE2 = EPREF * TEN ERRE2 = EPREF * TEN TAUEZ = ALEN / (TEN*TEN) ERRE2 = EPREF * TEN = ((T/TC)**4) TAUEZ = TAUEZ + BET* ((T/TC)**4) ERRE2 = EPREF * TEN = ((T/TC)**4) (7,*) * Teupp (High T) = ', TEPLIG(NUMBER) (7,*) * Teupp (High T) = ', TEPLIG(NUMBER) (7,*) * TEUPD (LOW T) = ', TEPLIG(NUMBER) (7,*) * TEPLIG(NUM</pre>	<pre>IF (T. CE. TC) THEM TAURE = ALPH / (TEM+TEM) EPRE2 = EPREF * TEM TAURE = ALPH / (TEM+TEM) EPRE2 = EPREF * TEM * (TCK - TEM)* TAURE = TAURE + BET * (TCK - TEM)* TAURE = TAURE + DAU TAURE = TAURE + TAU TAURE = TAURE + TAU (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', TPPLO(1), ' TO ', TPPLO(NUMBER) (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', TPPLO(1), ' TO ', TPPLO(NUMBER) (7,*) T E = ', EPRE2 * TAUE (7,*) T E = ', TPPLO(1), ' TO ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' TO ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' TO ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' TO ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' TO ', TPPLO(NUMBER) (7,*) T E = ', TPPLO(1), ' TO ', TPPLO(NUMBER) (7,*) T E = ', TPLO(1), '</pre>	IF (ELTERM .EQ. ') THEN
TAUEE = ALPH / (TEM*TEM) EPRE2 = EPREF * TEM EPRE2 = EPREF * TEM TAUEE = ALPH / (TEM*TEM) ELLE / (TEM*TEM) TAUEE = ALPH / (TEM*TEM) TAUEE = TAUEE + BEF * (TCC - TEM) **MH) EPRE2 = EPREF * TEM * ((T/C)**4) EPRE2 = EPREF * TEM * ((T/C)**4) (7,*) Taupp (High T) = ', TPPLO(NUMBER) (7,*) Taupp (High T) = ', TPPLO(NUMBER) (7,*) Taupp (High T) = ', TPPLO(NUMBER) (7,*) Teupp (High T) = ', TPPLO(NUMBER) (7,*) TEPL T	<pre></pre>	IF (1. 02: TC) T	20
2 2	<pre>2</pre>		TTEM * TTEM)
= 150 = 2 TAUEE = ALPH / (TEM*TEM) THE(21, 0, DATE) THE(2, 0, DATE) TAUEE = ALPH / (TEM*TEM) TAUEE = TAUEE = ALPH / (TEM*TEM) THE + TAUEE = TAUEE + TAUE1) TAUE = TAUEE = TAUEE + TAUE1) TAUE = TAUEE = TAUEE + TAUE1) TAUE + TAUE1) TAUE = TAUEE + TAUE1) TAUE + TAUE1) TAUE = TAUEE + TAUE1) TAUE + TAUE1) TAUE + TAUE1) TAUE + TAUEN + TAUE1) TAUE + TAUEN + TAUE1) TAUE + TAUEN + TAUE1) TAUE + TAUEN +	<pre>= 150 = 2 TAUEE = ALPH / (TEN*TEM) TAUEE = ALPH / (TEN*TEM) TAUEE = ALPH / (TEN*TEM) TAUEE = TAUEE + BET* ((TCK - TEM)* TAUE = TAUEE + BET* ((TCK - TEM)* TAUE = TAUEE + BET* ((TCK - TEM)* TAUE = TAUEE + TAUE (),*) Tc = ', TC (),*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),*) Te = ', TC (),*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),*) Te = ', TC (),*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),*) Te = ', TPLO(1), ' to ', TPPLO(NUMBER) (),*) Te = ', TPLO(1), ' to ', TPV(NUMBER) (),*) Te = ', TPLO(1), ' to ', TPV(NUMBER) (),*) Te = ', ALPHA(1), ' to ', ALPHA(NUMBER) (),*) Beta = ', BETA(1), ' to ', BETA(NUMBER) (),*) Beta = ', BETA(1), ' to ', BETA(NUMBER) (),*) Te = ', RETA(NUMBER) (),*) TE = ',</pre>	EPRE2 = EPRE7 =	
- 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2	- 2 - 2 TIME(2), 0, DATE) TIME(2), 0, DATE) TIME(4, 0, TIMER) (7,*) DATE, TIMER (7,*) DATE, TIMER (7,*) TC = ', TC (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPU(NUMBER) (7,*) T(unklapp) = ', TPU(1), ' to ', TPU(NUMBER) (7,*) Tauph = ', ALPHA(1), ' to ', NUMBER) (7,*) Beta = ', BETA(1), ' to ', BETA(NUMBER) (7,*) Beta = ', BETA(1), ' to ', BETA(NUMBER)	ELSE	
TAURE = TAURE + BET * ([TCK - TEM) ** HH) TIME(4, 0, FIMER) (7,*) DATE, TIMER (7,*) DATE, TIMER (7,*) Taupp (ind t) = ', TC (7,*) Taupp (ind t) = ', TPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (High T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (High T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (High T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (High T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (High T) = ', TPU(1), ' to ', TPPLO(NUMBER) (7,*) Taupp (High T) = ', TPU(1), ' to ', TPPLO(NUMBER) (7,*) T (unklapp) = ', TPU(1), ' to ', DTTEP (NUMBER) (7,*) T (unklapp) = ', TPU(1), ' to ', TPU(NUMBER) (7,*) T (unklapp) = ', TPU(1), ' to ', TPU(NUMBER) (7,*) T (unklapp) = ', TPU(NUMBER) (7,*) T (unklapp) = ', TPU(NUMBER) (7,*) T (unklapp) = ', TPU(1), ' to ', TPU(NUMBER) (7,*) T (unklapp) = ', TPU(NUMBER) (7,*) T (unklapp) = ', TPU(NUMBER)	$ TAUEE = TAUEE + BET * ((TCK - TEM) * TAUEE = TAUEE + BET * ((TCK - TEM) * TEM (0, TIMER) \\ TIME(4, 0, TIMER) \\ (7,*) DATE, TIMER \\ (7,*) DATE, TAUER \\ (7,*) TAUE = TAUER \\ (7,*) Taupp (Low T) = ', TPPHI(1), ' to ', TPPHI(NUMBER) \\ (7,*) Taupp (Bigh T) = ', TPPHI(1), ' to ', TPPHI(NUMBER) \\ (7,*) Taupp (Bigh T) = ', TPPHI(1), ' to ', TPPHI(NUMBER) \\ (7,*) Taupp (Bigh T) = ', TPPHI(1), ' to ', TPPI(NUMBER) \\ (7,*) Taupp (Bigh T) = ', TPPHI(1), ' to ', TPPI(NUMBER) \\ (7,*) Taupp (Bigh T) = ', TPPHI(1), ' to ', TPVI(NUMBER) \\ (7,*) Taupp (Bigh T) = ', TPPHI(1), ' to ', TPVI(NUMBER) \\ (7,*) Taupp (Bigh T) = ', TPPHI(1), ' to ', TPVI(NUMBER) \\ (7,*) Taupp (Ainther A) = ', DTTEP(1), ' to ', TPVI(NUMBER) \\ (7,*) Tauph = ', ALPHA(1), ' to ', BETA(NUMBER) \\ (7,*) Beta = ', BETA(1), ' to ', BETA(NUMBER) \\ (7,*) Beta = ', BETA(1), ' to ', BETA(NUMBER) \\ (7,*) TAUT = 0. \\ (7,*) TAUTA = 0. \\ (7,*) TAUTA = 0. \\ (7,*) TAUPA(1), ' to ', BETA(NUMBER) \\ (7,*) TAUTA = 0. \\ (7,*) TAUPA(1), ' to ', BETA(NUMBER) \\ (7,*) TAUTA = 0. \\ (7,*) TAUPA(1), ' to ', BETA(NUMBER) \\ (7,*) TAUTA = 0. \\ (7,*) TAUTA = 0. \\ (7,*) TAUTA = 0. \\ (7,*) TAUPA(1), ' TO ', TAUTA = 0. \\ (7,*) TAUPA(1), ' TO ', TAUTA = 0. \\ (7,*) TAUTA$	TAUEE = ALPH /	TEM*TEM)
TIME (21, 0, DATE) EPRE7 = EPREF * TEM * ([T/TC)**4) TIME (4, 0, FINER) END IF (7,*) DATE, TIMER C (7,*) Tauep (it) *, TC (7,*) Tauep (it) *, TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Tauep (it) *, TC (7,*) Tauep (it) *, TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Tauep (it) *, TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Tauep (it) *, TEPI(1), ' to ', TPPLO(NUMBER) (7,*) Tauep (it) *, TPLO(1), ' to ', TPPLO(NUMBER) (7,*) Tauep (it) *, TPLO(1), ' to ', TPPLO(NUMBER) (7,*) Tauep (it) *, TU(1), ' to ', TPPLO(NUMBER) (7,*) Tauep (it) *, TU(1), ' to ', TPPLO(NUMBER) (7,*) Tauep (it) *, TU(1), ' to ', TTU(NUMBER) (7,*) T (unklapp) *', TU(1), ' to ', TU(NUMBER) (7,*) T (unklapp) *', TU(1), ' to ', TU(NUMBER) (7,*) T (unklapp) *', TU(1), ' to ', TU(NUMBER) (7,*) T (unklapp) *', TU(1), ' to ', TU(NUMBER) (7,*) T (unklapp) *', TU(1), ' to ', TU(NUMBER) (7,*) T (unklapp) *', TU(1), ' to ', TU(1), ' to ', ' TU(1),	$ \begin{array}{llllllllllllllllllllllllllllllllllll$	TAUEE - TAUEE - TAUEE	BET * ((TCK - TEM) **RH)
TAUE (* 0, TIMER) C END IF (7,*) DATE, TIMER C TAUE = TAUEF + TAUEF1 / (TAUEE + TAUEF1) (7,*) Taupp (High T) = ', TPPHI(1), ' to ', TPPHI(NUMBER) RKE = EPRE2 * TAUE (7,*) Taupp (High T) = ', TPPHI(1), ' to ', TPPHI(NUMBER) C ELSE (7,*) Taupp (High T) = ', TPPHI(1), ' to ', TPPI(NUMBER) C ELSE (7,*) T (umklapp) = ', TTU(1), ' to ', TTPPI(NUMBER) C ELSE (7,*) T (umklapp) = ', TTU(1), ' to ', TTU(NUMBER) RKE = 0 (7,*) T (umklapp) = ', TTU(1), ' to ', TTU(NUMBER) END IF (7,*) T (umklapp) = ', TTU(1), ' to ', TTU(NUMBER) C (7,*) T (umklapp) = ', TTU(1), ' to ', TTU(NUMBER) C (7,*) * ALPHA(1), ' to ', MUMBER) C	THME(4, 0, TIMER) C END IF (7,*) DATE, TIMER C TAUE = TAUEF1 / (TAUEE + TAU (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) C TAUE = TAUEF1 / (TAUEE + TAU (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) C TAUE = TAUEF1 / (TAUEE + TAU (7,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) C EPAF2 * TAUE (7,*) Taupp (Low T) = ', TPPLI(1), ' to ', TPU(NUMBER) C ELSE (7,*) ' E-ph Tau = ', TPPLI(1), ' to ', TTU(NUMBER) C ELSE (7,*) ' Taupp (I = ', ALPHA(1), ' to ', TTU(NUMBER) C ELSE (7,*) ' Taupp = -', ALPHA(1), ' to ', ALPHA(NUMBER) C ELSE (7,*) ' Beta = ', BETA(1), ' to ', BETA(NUMBER) C RKVAL = 0.	EPRE2 = EPRE5	TEM * ((T/TC)**4)
(/,*) UATE, TANK (/,*) TE = ', TE (),*) TE = ', TE (),*) Taupp (High T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),*) Taupp (High T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),*) Taupp (High T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),*) Teph Tau = ', TPU(1), ' to ', TPPLO(NUMBER) (),*) T (umklapp) = ', TTU(1), ' to ', TPU(NUMBER) (),*) ALPHA(1), ' to ', ' TPU(NUMBER) ()	(/,*) UATE, TANK (/,*) Tare, TAUE = TAUEE * TAUE = TAUEE * TAUE + TAU (/,*) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLI(NUMBER) (/,*) Taupp (Hauh T) = ', TPPLI(1), ' to ', TEPLI(NUMBER) (/,*) ' E-ph Tau) = ', DTTEP(1), ' to ', DTEP(NUMBER) (/,*)' T (umklapp) = ', TTU(1), ' to ', DTEP(NUMBER) (/,*)' T (umklapp) = ', TTU(1), ' to ', DTEP(NUMBER) (/,*)' T (umklapp) = ', TTU(1), ' to ', TTU(MBER) (/,*)' T (umklapp) = ', TTU(1), ' to ', TTU(MBER) (/,*)' Beta = ', BETA(1), ' to ', BETA(NUMBER) (/,*)' Beta = ', BETA(1), ' to ', BETA(NUMBER)	JI DUS	
(),)) Taupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (),)' Taupp (Low T) = ', TPPHI(1), ' to ', TPPHI(NUMBER) (),)' E-ph Tau = ', TEPI(1), ' to ', TEPI(NUMBER) (),'' C (E-ph Tau = ', TTU(1), ' to ', DTEP(NUMBER) (),'' T (umklapp) = ', TTU(1), ' to ', TTU(NUMBER) (),'' A) TAU = ', ALPHA(1), ' to ', TTU(NUMBER) (),'' A) TAU = ', ALPHA(1), ' to ', TTU(NUMBER) (),'' A) TAU = ', ALPHA(1), ' to ', TTU(NUMBER) (),'' A) TAU = ', ALPHA(1), ' to ', 'TTU(NUMBER) (),'' A) TAU = ', ALPHA(1), ' to ', 'TTU(NUMBER) (),'' A) TAU = ', ALPHA(1), ' to ', 'TTU(NUMBER) (),'' A) TAU = ', ALPHA(1), ' to ', 'TTU(NUMBER)	(7,*) Tupp (Low T) = ', TPPLO(1), ' to ', TPPLO(NUMBER) (7,*) Tupp (Low T) = ', TPPHI(1), ' to ', TPPHI(NUMBER) (7,*) ' Tupp (High T) = ', TPPHI(1), ' to ', TEPHI(NUMBER) (7,*) ' d(E-ph Tau) = ', DTTEP(1), ' to ', DTEP(NUMBER) (7,*) ' T (umklapp) = ', TTU(1), ' to ', DTEP(NUMBER) (7,*) ' T (umklapp) = ', TTU(1), ' to ', TTU(NBER) (7,*) ' Alpha = ', ALPHA(1), ' to ', ALPHA(NUMBER) (7,*)' Beta = ', BETA(1), ' to ', BETA(NUMBER) (7,*)' Beta = ', BETA(1), ' to ', BETA(NUMBER)		(CODING - GOLLEN / CODI
(7,*) Taupp (100 1) - , TFEU(1), CO , TEPHI(NUMBER) (7,*) Teph Tau = , TEPI(1), Co , TEPHI(NUMBER) (7,*) Teph Tau = , DTTEP(1), Co , DTTEP(NUMBER) (7,*) T(umklapp) = , TTU(1), Co , TTU(NUMBER) (7,*) T(umklapp) = , TTU(1), Co , TTU(NUMBER) (7,*) Alph = , ALPHA(1), Co , ALPHA(NUMBER) (7,*) Alph = , ALPHA(1), Co , MENNINER) (7,*) Alph = , ALPHA(1), Co , MENNINER)	(7,*)' Taupp (act 1) - ', TFLU(1),' Co', TPHI(NUMBER) (7,*)' Taupp (Heigh T) - ', TEPHI(1),' to', TEPHI(NUMBER) (7,*)' Tauph = ', DTTEP(1),' to', TEPH(NUMBER) (7,*)' T (umklapp) = ', TTU(1),' to', TTU(NUMBER) (7,*)' Alpha = ', ALPHA(1),' to', TTU(NUMBER) (7,*)' Beta = ', BETA(1),' to', BETA(NUMBER) (7,*)' Beta = ', BETA(1),' to', BETA(NUMBER)	- 1 AUVE = 1 AUVE - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1	UEFI / (IAUGE + IAUEFI) F
(7,*)* E-ph Tau - ', TEP1(1), ' to ', TEP1(NUMBER) (7,*)* d(E-ph Tau) - ', DTTEP(1), ' to ', DTTEP(NUMBER) (7,*)* T (umklapp) - ', TTU(1), ' to ', TTU(NUMBER) (7,*)* Alpha - ', ALPHA(1), ' to ', TTWAEPEN (7,*)* Alpha - ', ALPHA(1), ' to ', STWAEPEN (7,*)* Alpha - ', DTWAEPEN	(),*)' E-ph Tau = ', TEPI(I),' to', TEPI(NUMBER) (7,*)' d(E-ph Tau) = ', DTTEP(I),' to', DTTEP(NUMBER) (7,*)' T (umklapp) = ', TTU(I),' to', TTU(NUMBER) (7,*)' Alpha = ', ALPHA(I),' to', ALPHA(NUMBER) (7,*)' Beta = ', BETA(I),' to', BETA(NUMBER) (7,*)' Beta = ', BETA(I),' to', BETA(NUMBER)	1. 1. 1. IFLUD(1.). (0. 1. IFLUD(NUTER)) 1. 1. 1. TEPHI(1.). to '. TPPHI(NUENER)	
(7,*)'d(E-ph Tau) = ', DTTEP(I), 'to', DTTEP(NUMBER) (7,*)'T (umklapp) = ', TTU(I), 'to', TTU(NUMBER) (7,*)'Alpha = ', ALPHA(I), 'to', ALPHA(NUMBER) (7,*)'Alpha = ', ALPHA(I)', 'to', 'ALPHA(I)', 'to', 'to	<pre>(7,*)'d(E-ph Tau) = ', DTTEP(1), ' to ', DTTEP(NUMBER) (7,*)'T (umklapp) = ', TTU(1), ' to ', TTU(NUMBER) (7,*)'Alpha = ', ALPHA(I), ' to ', ALPHA(NUMBER) (7,*)'Beta = ', BETA(1), ' to ', BETA(NUMBER) (7,*)'Beta = ', BETA(1), ' to ', BETA(NUMBER)</pre>		
(7,*)'T (umklapp) = ', TTU(1), 'to', TTU(NUMBER) (7,*)'Alpha = ', AlphA(1), 'to', AlphA(NUMBER) 'T * ' * * * * * * * * * * * * * ********	(7,*)'T (umklapp) = ', TTU(1), 'to', TTU(NUMBER) (7,*)'Alpha = ', AlphA(1), 'to', AlphA(NUMBER) (7,*)'Beta = ', Beta(1), 'to', Beta(NUMBER)	u) = ', DTTEP(1), ' to ', DTTEP(NUMBER) RKE = 0	
(7,*)' Alpha = ', AlpHA(1), ' to ', AlpHA(NUMBER) '1 +!! =-! = ' ====!' ====!' =====!	(7,*)'Alpha = ', Alpha(1), ' to ', Alpha(NUMBER) (7,*)'Beta = ', Beta(1), ' to ', Beta(NUMBER)	p) = ', TTU(1), ' to ', TTU(NUMBER)	
	(1, -) parts - , parts (1), (0, parts inversel)	, ALPHA(1), ' to ', ALPHA(NUMBER) BEFREAT ' to ' BEFREAT WINNERED	

	Call. Perko (MMTN - MMAY - DMTAL)	c	
	DEVIAL - DEVIAL + 15-13 + DEV + 4534 + DIADM / 1	,	NTRNCTON
	DAVIAL - AAVALL - AD-14 - AVUD - UE40 - AVVAL - DAVIAL - DA		
	AN OWL - AN O		COMMON / BL
	IF (REVALL GT. BROAN - REVAL		COMMON /NS
υ	₩\$1175 (6,4)'T = ', T2M, ' K = ', R4C	U	
υ	WRITE (6,*)'TAUEE = ', TAUEE, ' TAUEP = ', TAUEPI		VV = GROUP
U	WRITE (6,*)'EPREF = ', EPREF, ' EPRE2 = ', EPRE2		XT - X / T
υ	WRITE (6,*)'TAUE = ', TAUE		IF ((XT .G
90	CONTINUE		R = 0.
	WRITE (6,*)'Kmax = ', RNMAX		ELSE
-	MRITE (7, *) - 1., -1.		BOL = EXP
100 201		U	
ថ			
5		C ###	ND STATES
υc		υ	
	UNCLION FULLY		
	isiiseedeesiiseedeesiiseedeeseedeeseedeeseedeeseedeeseedeeseedeesee		una Alatara
, L		L	
, ,	COMON (PARS) 7, TC, HBAR, RKB, RLV	00	
8	COMPON /TAUS/ TAUPP1, DTPP, TAUREP1, DTEP, TU	C ###	*******
ช	COMPON /WS/ WOLIN, WOLINK, WSTEP		FUNCTION 6
8	X00000 /DEFECT/HW, G, DM, ACUBED	υ	Calculates
U U		0	for phonon
5 1			
	M = 1.402-19 × 12-3 / MDAM / 1 + 11-210 / MDAM / 1 + 11-210 / MDAM / 1 + 11-210 / MDAM / 1 + 11 + 11-210 / MDAM / 1	J	DIMENSION
ι υ	MARK - TWORLE + ATEK - (EVE(-T'-T'-T')) - T		COMMON /1.4
,	T (T .GT . TC) THEN	U	
	TAUEP TAUEPI	,	LOOP = 1
	138		0 - 20
3	$GMP = TC + 3.52 + (1 - ((T/TC)^{++4}))$		0 - M
-	IF (W .LT. GAP) THEM	10	CONTINUE
1	TAUEP = TAUEP1 * DTEP		IF ((X .GT
7			1 THEN
	TAUEP - TAUEPI		01) / - ^(Io
- 1			
а с			1006 = 10
F ,	AUO = TAURE + T		END IF
υ			IF ((LOOP
2	$\mathbf{wumer} = \mathbf{G} + 4 + 3.1416 + \mathbf{VV} + \mathbf{VV} + \mathbf{VV}$	υ	PRINT *,
đ	NEVON = VICUBED + EM + HM + HM + HM + DM + DM		GROUPV = V
H	r2 = 1512 * Rauner / Denon	υ	
F	TAUD * T2 / (TAUD + T2)		RETURN
ΞĘ (21 = 12	C	END
<u>ي</u> ت		טט	
υ υ		0 ***	
υ			SUBROUT INE
0 ****		U	Enters ph
E.	FUNCTION RINTEG(X)	*** •	*******
ы U (Evaluates the integral function.	U	
ג ז ט נ	das to be a function in order to call from integration routines. Arich was than from MIMEDITAL raciosa		DIMENSION
	MILCI BARC VARAD IICUN NUTURALIALA VALAILAUS. 1944-1944-1944-1944-1944-1944-1944-1944		
· · · · · · ·			

```
s the group velocity from the ladder
ns of energy hx.
                                                                                                                                                                                                                                                                                                                                                                                                                                        T. NLAD (LOOP)) . AND. (X . LE. WLAD (LOOP + 1)))
                                                                                                                                                                                                                                                                                                        VV * XT * XT * BOL / ((BOL - 1)*(BOL - 1))

CAU(X) * DOS(X)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               'X = ', X,' MLAD = ', MLAD (LOOP),' VV = ', VV
VV
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           .LT. NLAD) .AND. (OK .NE. 1)) GO TO 10
                                                                                                                                                                                                                               is already * hbar/1.602e-19 *********
                                                                                                                                                                                                                                                   ********
                                                                                                                                                                                                                                                                       *********
                                                                                                                                   SE. 30) .OR. (X .GT. WMAX)) THEN
MIIM(30), WHT(30), WPCM(30)
LOCK/ WLIM, MHT, WPOM, NLIMS
ARS/ T, TC, HBAR, RKB, RLV
S/ MMIN, WMAX, WSTEP
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        WLAD(10), V(10), RLAD(10)
ARS/ T, TC, HBAR, RKB, RLV
NDDER/ WLAD, V, NLAD
                                                                                                                                                                                                                                                     / in inverse Angstroms
converted to meV
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           DDER/ WLAD, V, NLAD
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          MLAD(10), V(10)
                                                                                                                                                                                                                                                                                                                                                                                                                                                              ROUPV (X)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                (1 + 30
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       06 + 1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   LADD
                                                                                                 (X) V
                                                                                                                                                                                             (XT)
```

D rth Rlad /0., 16, 32, 64, 80, 96, 112, 0, 0, 0/ Nlad - 7	A(LOOP) = (WHT(LOOP + 1) - WHT(LOOP)) / ((WLIM(LOOP + 1) - NLIM(LOOP))**WPOM(LOOP)) B(LOOP) = WHT(LOOP) Else Else
DO 10 LOOP = 1, 10 WLAD (LOOP) = RLAD (LOOP) PRLNT *, LOOP, WLAD (LOOP) 10 CONTINUE	A(LOOP) = (WHT(LOOP) - WHT(LOOP + 1)) / ((WLIM(LOOP + 1) - 1 WLIM(LOOP)) ***WPOW(LOOP)) B(LOOP) = WHT(LOOP + 1) B(LOOP) = WHT(LOOP + 1) 200 IF 200 IF 20
DO 20 LOOP = 2, 10 V(LOOP) = (WLAD(LOOP) - WLAD(LOOP - 1)) * 1.602E-19 / (1E3*HEAR) V(LOOP) = V(LOOP) / (RLV*IE10) 20 CONTINUE	A (LOOP) = 0. B (LOOP) = 0. END IF
END	20 CONTINUE RETURN END
russination contraction contraction and contraction contractions of the contraction of states for a given onega	С С алахнаталалалалалалалалалалалалалалалалалала
DIMENSION MLIM(30), MHT(30), WPOM(30) Dimension A(30), B(30) Common /BLOCK/ Wlim, Wht, WPOW, Mlims	C ************************************
COMPANN / PREF/ A, B RDOS = 0 DO 10 LOOP = 1, (NLINS - 1) IF ((W.GT. WLIN(LOOP)) .AND. (W.LT. WLIN(LOOP + 1)) .AND. (<pre>C WPOM = 2, 2, 2 C The DOS follows an w^2 dependence from W=0 to M=30meV C with the value at M=30 being 1. C From W=30 to M=50meV the DOS falls from 1 to 0 with the C dependence also being w^2.</pre>
<pre>1 WPOW(LLOOP) .WE. 0)) THEN IF (WHT(LOOP) .LT. WHT(LOOP + 1)) THEN RDOS = A(LOOP) * ((W - WLIM(LOOP)) **WPOM(LOOP)) + B(LOOP) ELSE RDOS = A(LOOP) * ((WLIM(LOOP + 1) - W)**WPOM(LOOP)) + B(LOOP) END IF END IF IF ((W .E.O. WLIM(LOOP)) .AND. (WPOM(LOOP) .NE. 0)) THEN RDOS = B(LOOP) END IF IO (CMTINUE END IF</pre>	C DIMENSION WLIM(30), WHT(30), WPOW(30) DIMENSION WLIM(30), WHT(30), WPOW(30) DIMENSION RSCLIM(30), RSCHT(30), RNPOM(30) DIMENSION SSCLIM(30), SSCHT(30), SSCPOM(30) DIMENSION SSCLIM(30), SSCHT(30), SSCPOM(30) DIMENSION SSCLIM(30), SNHT(30), SSCPOM(30) DIMENSION CSCLIM(30), CSCHT(30), CSCPOM(30) DIMENSION CNLIM(30), CNHT(30), CSCPOM(30) CHARACTERFI ANS, DSET COMMON /BLOCK /WLIM, WHT, WPOM, NLIMS
DOS = RDOS END SUBROUTINE PREFAC Calculates prefactors for density of states	COMMON /PARS/ T, TC, HBAR, RKB, RLV COMMON /WS/ WMIN, WHAX, WSTEP C C ATTINGTON /WS/ WMIN, WHAX, WSTEP C ATTINGTON /WS/ WMIN, WHAX, WSTEP C ATTINGTON /WS/ WMIN, WSTEP C ATTINGTON /WS/ WMIN, WSTEP C ATTINGTON /WS/ WMIN, WSTEP C ATTINGTON /WS/ WMIN, WSTEP C ATTINGTON / WS/ WMIN, WSTEP C ATTINGTON / WS/ WMIN, WSTEP
DIMENSION MLIM(30), WHT(30), WPOM(30) DIMENSION A(30), B(30) COMMON /BLOCK/ MLIM, WHT, WPOM, NLIMS COMMON /PREF/ A, B COMMON /PREF/ A, B COMMON /PREF/ A, B COMMON /PARS/ T, TC, HBAR, RVB, RLV DO 20 LOOP = 1, (NLIMS - 1) IF (WLIM(LOOP) .NE. MLIM(LOOP + 1)) .AND. (MPOM(LOOP) .NE. 0.)) IF (WLIM(LOOP) .NE. MLIM(LOOP + 1)) .AND. (MPOM(LOOP) .NE. 0.)) IF (WHT(LOOP) .LE. WHT(LOOP + 1)) THEN	C DATA RSCLIM /00., 06., 12., 14., 20., 23., 25., 27., 30., 33., 1 35., 39., 41., 44., 48., 58., 62., 69., 71., 77., 81., 90., 2 0., 0., 0., 0., 0., 0., 0., 0./ DATA RSCHT /00., 04. 15., 15., 36., 30., 33., 15., 12., 17., 19., 1 0., 08., 20., 20., 20., 20., 34., 17., 15., 01., 0., 2 0., 0., 0., 0., 0., 0., 0./ DATA RSCPOW /1. 1., 1., 1., 1., 1., 1., 1., 1., 1., 1

C C ******** Data for normal phonon DOS in meV *********	1 36., 41., 46., 50., 53., 55., 59., 59., 61., 63., 67., 70., 2 72., 77., 80., 90., 91., 100., 0./
	DATA CSCHT /00., 05., 10., 41., 30., 24., 28., 25., 26., 18.,
DATA RNLIM /00., 06., 10., 16., 18., 24., 30., 33., 36., 44., 50., 1 58., 62., 70., 80., 90., 100., 0., 0., 0., 0., 0., 0., 0.	1 18., 14., 26., 12., 12., 10., 15., 14., 17., 15., 19., 12., 2 13 8 11 1 . 0 0 / /
2 0., 0., 0., 0., 0., 0./	DATA CSCPON /1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
DATA RNHT /00., 04., 15., 21., 19., 40., 26., 38., 25., 06., 06.,	1 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.
I I7., 12., 28., 18., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 2., 2., 2., 2., 2., 2., 2., 2., 2., 2	2 1., 1./ C CSCN. = 23
DATA RNPOW /1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,	$c_{\rm SCL} = 27$
1 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.	
Z V., V./ RNL = 17	C THATTATT DECENTED FOR DOLMEL PHONON UCS IN MEV TATATTATT C
	C
С. 4446444444444444444444444444444444444	DATA CNLIM /00., 05., 07., 10., 13., 15., 16., 18., 23., 25., 27.,
C Data of Strobel et al	1 30., 33., 37., 40., 53., 58., 61., 64., 66., 70., 76., 81.,
C Data for superconducting phonon DOS in meV	Z 83., 90., 93., 93., 100., 0., 0./ DATA CNHT /00., 20., 40., 22., 40., 30., 21., 31., 16., 24., 22.,
С инералиникальностальностичностичностичностичностичностичности. С	1 26, 19, 18, 10, 06, 13, 08, 14, 12, 14, 07, 04, 2 08 04 0 0 0 0 0
DATA SSCLIM /00., 05., 11., 12., 14., 15., 18., 20., 22., 24.,	DATA CNPOW /1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
1 25., 27., 30., 35., 40., 45., 50., 55., 60., 65., 70., 75.,	1 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.
Z 80., 83., 90., 93., 100., 1.00., 1.00., 0., 0., 0., 84., 74., 83., 71.,	2 1., 1./ CML = 25
1 67., 70., 57., 70., 60., 58., 43., 40., 36., 30., 22., 16.,	
2 16., 13., 11., 10., 01., 0., 0./	DATA RMIN, RWAX /0.01, 100./
DATA SSCPOM /1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,	NIMU = NIMU
2 1., 1./	C III
SSCL = 28	10 CONTINUE
C ######### Data for normal phonon DOS in meV ###################################	WRITE (6,*)'Select which DOS to use ' WRITE (6,*)'R Rhyne et al (neutron weighted)'
Ū	WRNTTE (6,*)'S Strobel et al'
DATA SNLIM /00., 05., 11., 12., 14., 15., 17., 18., 20., 22., 24., 	WRITE (6,*)'C Chaplot (bare, calculated)'
1 23., 29., 30., 35., 90., 95., 100./ 2 70 75 80 85. 90 95. 100./	NUKITZE (6,") Read (5,30) dset
DATA SNHT /00., 07., 30., 42., 39., 65., 60., 66., 60., 58., 69.,	IF ((DSET .NE. 'R') .AND. (DSET .NE. 'S') .AND. (DSET .NE. 'C'))
1 69., 62., 66., 52., 50., 55., 40., 35., 35., 33., 31., 32.,	1 60 10 10
2 34:2 21.2 20.1 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1.2 1	20 CONTINUE
1 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.	WRITE (6,*)'Calculate K for superconducting material Y/N ?'
SNL = 30	30 FORMAT (1,2) ANS
U	IF ((ANS .NE. 'Y') .AND. (ANS .NE. 'Y') .AND. (ANS .NE. 'N') .AND.
	1 (ANS .NE. 'n')) GO TO 20
C Data of Chaplot et al 'bare' phonon DOS	IF ((ANS . BQ. 'Y') . OR. (ANS . EQ. 'Y')) THEN
	SC =].
C Data for superconducting phonon DUS in mev C аааааааааааааааааааааааааааааааааааа	IF (DSET .EQ. 'W') THEN NLIMS = RSCL
. 0	JI CN3
C DATA CSCLIM /0., 07., 10., 13., 16., 24., 26., 28., 32., 36.,	IF (DSET . EQ. 'S') THEN
C I 38., 41., 44., 45., 50., 54., 57., 60., 66., 67., 73., 75., C 2 87.90.0.0.0.0.0.0.0.0.0	NLLMS = SSCL FND TF
C DATA CSCBT /00., 21., 50., 40., 44., 37., 36., 27., 21., 21.,	IF (DSET .EQ. 'C') THEN
C 15., C 1 1 1 1 2 1 3 1 3 1 3 1 2 1 6 1 2 0 9 0 01.	NLINS = CSCL FUN TF
	ELSE ELSE
DATA CSCLIM /0., 5., 7., 11., 15., 18., 22., 25., 27., 30., 32.,	SC # 0.

Integration routine for Thermal conductivity PAUSE 'Too many steps when calculating DOS.' S = 0.5 * (B - A) * (RINTEG(A) + RINTEG(B)) IF (ABS(S - OS) .LT. EPS*ABS(OS)) RETURN IF (N . EQ. 1) THEN S = 0.5 * (B - A) * (DOS(A) + DOS(B)) + (B - A) *SUM/TNM) S = 0.5 * (S + (B - A) * SUM/TNM)SUBROUTINE TTRAPZ (A, B, S, N) PARAMETER (EPS=1E-2, JMAX=14) PARAMETER (EPS=1E-2, JMAX=16) SUBROUTINE DSIMP (A, B, S) CALL DTRAP2 (A, B, ST, J) SUBROUTINE TSIMP (A, B, S) S = (4.*ST - OST) / 3. SUM = SUM + RINTEG(X) DEL = (B - A) / TNM DEL = (B - A) / TNM SUM = SUM + DOS (X) X = A + 0.5 + DELIF (N . BQ. 1) THEN X = A + 0.5 * DEL DO 10 J = 1, JMAX DO 10 J = 1, IT DO 10 J = 1, IT X = X + DELs = 0.5 * (s X = X + DEL IT = 2 * IT OST = -1.E30 IT = 2 + ITOS = -1.E30 10 CONTINUE CONTINUE TI - HNT SUM = 0. OST = ST TI = MNT SUM = 0. 10 CONTINUE 05 = SO IT = 1 II = IEND IF END IF RETURN RETURN ELSE ELSE GNG 2 υu υυ υ υ υ C υu

Integration routine for Density Of States SUBROUTINE DTRAPZ (A, B, S, N) WLIM(LOOP) = RSCLIM(LOOP) MLIM(LOOP) = CSCLIM(LOOP) WPOW(LOOP) = SSCPOW(LOOP) WPON (LOOP) = RSCPON (LOOP) WLIM (LOOP) = SSCLIM (LOOP) WPON (LOOP) = CSCPON (LOOP) WLIM (LOOP) = RNLIM (LOOP) MLIM (LOOP) = SNLIM (LOOP) MLIM(LOOP) = CNLIM(LOOP) WPOW (LOOP) = RNPOW (LOOP) NPON (LOOP) = CNPON (LOOP) NPOW (LOOP) = SNPOW (LOOP) WHT (LOOP) - SSCHT (LOOP) WHT (LOOP) = CSCHT (LOOP) WHT (LOOP) = RSCHT (LOOP) IF (DSET .EQ. 'S') THEN IF (DSET .EQ. 'C') THEN MHT (LOOP) = CNHT (LOOP) IF (DSET .EQ. 'R') THEN IF (DSET .EQ. 'R') THEN WHT (LOOP) - RNHT (LOOP) IF (DSET .EQ. 'S') THEN IF (DSET . BQ. 'C') THEN WHT (LOOP) - SNHT (LOOP) IF (DSET . 80. 'S') THEN IF (DSET . BQ. 'C') THEN IF (DSET . BQ. 'R') THEN DO 40 LOOP = 1, NLINS THEN IF (SC . 80. 1) THEN IF (SC .EQ. 1) NLIMS - RNL NLIMS = CHL NLINS = SNIL END IF CONTINUE TC = 90 END IF END IF END IF END IF END IF TC = 0END IF END IF END IF END IP END IF **JI ON3** ELSE ELSE END 9

υ

υυ

υ υυυ υ

OST = -1.E30 OS = -1.E30 DO 10 J = 1, JWAX CALL TTRAPZ(A, B, ST, J) S = (4.*ST - OST) / 3. IF (ABS(S - OS) .LT. EPS*ABS(OS)) RETURN OS = S OS = S 10 CONTINUE PAUSE 'Too many steps when calculating K.'
Appendix D

Program to Model Connectivity

The program is written in FORTRAN 77 and conforms to the ANSI standard. Functions and subprograms are utilised wherever they may assist with the clarity of structure. Global data is passed to each subroutine or function by use of COMMON blocks which are by necessity large. The program requires the use of the NAG subroutine library to generate a sequence of random numbers. Routine G05CCF is a subroutine which resets the seed for the random number generator to a nonrepeatable value. Function G05CAF returns a psuedo-random number of type REAL in the range 0 to 1. These NAG routines be replaced by an appropriate set of routines which generate pseudo random numbers. Meaningful variable name are used throughout the program wherever possible.

The overall conductance of the network is computed by the iterative procedure to be found in routine CHEBY. A description of the method of solution and a table of appropriate acceleration parameters may be found in appendix B.

NOTE that the routine requires use of the NAG subroutine library C then open the file ready for output and put in the data heading C Read in the matrix size required, and the output file name the relative probabilities of B and C is passed into PP2 IF ((T.GT. (TC - DTC)) . AND. (T. LT. TC)) THEN WRITE (8, *) T, SOLN, CONN, RHO, ICOUNT, ITER, Subroutine to pick a value of the conductance The three values are passed in A, B and C. 'Failed - too many iterations' The probability of A is passed into PP1, WRITE (8,*) T, SOLN, CONN, ITER WRITE (6,*) 1. - PI, SOLN, ITER REAL FUNCTION SVAL (A, B, PP1, CONN) for the choice of a random number PI = 1 + (T/DTC) - (TC/DTC)IF (T. LE. (TC - DTC)) THEN SOLN - CHEBY (IMAXIT, ITER) IF (SOLM .EQ. - 1.) THEN ICOUNT - ICOUNT + ITER IF (RAN .LE. CONN) THEN IF (RAN .LT. PP1) THEN WRITE (8, *) ICOUNT, -1 RAN2 - GOSCAF (RAN2) RAN = GOSCAF (RAN) CALL VPRINT CALL SPRINT CALL SETUP SVAL - A SVAL - B ITER = 0SVAL = 0 P1 = 0 10 CONTINUE STOP . END IF END IF END IF TI ONI END IF RETURN ELSE ELSE ELSE GNG --U Ų υυ U υ υ v υ υ υ υ U υ υ υ υ υ υ υυ υ Start off the voltage condition as zero, after that use the 法大法 建水素 化合合 的复数主义主义 的复数有名的复数 化合合合合合合合合合合合合合合合合合合合合合合合合合 old solution, this speeds things up a bit. For testing the COMMON / VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT G05CBF sets the random number seed to a repeatable value resistors and capacitors. This attempts to simulate IMAXIT specifies max no of iterations before OMEGA=1 the effect of thermal conductivity V fixed, find I Go around loops, choosing a new matrix each time. SI,S2 and S3 are the possible conductance values P1 is the probability of S1 Program to calculate the effect of a matrix of program, reset the voltage condition each time. ISNUM-3 chooses the first 3 values of SS2 Set up the required arrays and variables DIMENSION V(50,50,50), S(50,50,3) COMMON /LIMIT/ START, END, STEP COMMON /VARS/ S1, S2, P1, CONN INPLICIT REAL(A - H,O - Z) IMPLICIT INTEGER (I - N) COMPANN / THRESH / THRESH WRITE (6,*)'Enter Tc ' WRITE (6,*)'Enter dTc' READ (5,*) DTC DO 10 T = 30, 120, 2 IF (T.GE. TC) THEN COMMON /MATRIX/ V, S COMMON /FILE/ FIL NPTS - M + M + L S2 = T + 0.001DIMENSION SS2 (4) CHARACTER*12 FIL CALL GOSCOF (10) READ (5,*) TC PF 1s P(S2)/P(S3) **IMAXIT - 300** INTEGER FLAG CALL HEADER SI = 0.05ICOUNT = 0ISNUM = 3 ICOUT - N **VTOT = 5.** PF = 0.5 ICIN = 1P1 = 1 END IF SI = 1. υυ 0000000000 υυ υ υ υ υ υ

C 42448444444444444444444444444444444444	υ
	WRITE (8,*)'Temperature (K)'
	MALLE 0, 7) CONQUELTIE
CHARMCTER*16 DATE	
CHARACTER*0 TIMER	U U
COMPANN /VARS/ 51, 52, P1, CONN	υ
COMMON /VAR2/ W, M, L, MPTS, NN, RHO, ICIN, ICOUT, VTOT	C ####################################
COMBON /FILE/ FIL	C This procedure uses a simultaneous over-relaxation method
CORPON /THRESH/ THRESH	C using a Gauss-Siedel type method and Chebyshev acceleration!
COMMON /LINIT/ START, END, STEP	C Needs a value of RHO to be defined before use,
WRITE (6,*)'Enter the file name to store the data'	
READ (5,10) FIL	FUNCTION CHEBY (IMAXIT, ITER)
10 FORMAT (1A)	REAL ONEGA
OPEN (8, FILE-FIL, STATUS-'NEW')	INTEGER FLAG
WRITE ($6, t$)'Enter number of rows, columns and planes (M, N, L)'	DIMENSION V(50,50,50), S(50,50,3)
READ (5.*) M. H. L	COMMON /MATRIX/ V. S
MRITE (6.9) Ster the threshold value'	COMMON /VAR2/ N. M. L. NPTS, MN. RHO, ICIN, ICOUT, VTOT
READ (5.*) THRESH	CONCON /THRESH/ THRESH
MATTRE (6.2) (Enter the value of Rho for this matrix size'	
	TURNEL
MATTER (S.*.)'STAGETERE CONNECTIVITY XIOM U CO I.	C INAMA 14 THE MAX NO OF ICEFALIONS DEFOCE CMPLAAT C INAMYO 14 PAR AND AD AN AN AN AN AND AN AND AN AND AND
	din future ta cina may no of the actions periode district of
	() 111111 111111 + 5
C Print out options so can be seen on batch output	
C Print out headings onto data file for GPLOT	INAXZ = INT(FLONT(IMAXI)*1.4)
υ	OMEGN = 1.
CALL TIME (21, 0, DATE)	υ
CALL TIME (4, 0, TIMER)	10 CONTINUE
U	FLAG = 0
WKITE (6,20) TIMER, DATE	υ
WERITE (6,30)	C Do Chebyshev with odd/even ordering
WRITE (6,40) FIL	υ
MRITE (6,50) M, N, L	DO 50 J = 0, 1
MRITE (6,60) RUO	ITER = ITER + I
MRITE (6,70) TERESH	
MRITE (6, 80) CONN	DO 40 IAX $= 1, M$
WRITE (6,90)	DO 30 IAY = 1, N
MALTE (6,100)	DO 20 IAZ = 1, L
υ	U
WRITE (6,20) TIMER, DATE	I = IAX + (IAY - I) + M + (IAZ - I) + MN
WRITE (6,30)	IF ((MOD(I + J, 2) .EQ. MOD(ITER, 2)) .AND. (I .NE. NPTS))
WERITE (8,40) FIL	1 THEN
WARITE (8,50) M, N, L	- 0 - MAS
WRITE (8,60) RHO	SONZ = 0.
MRITE (8,70) THRESH	υ
WRITE (6,80) CONN	C WORK OUT CO-ORDS OF V IN THE EQUATION
MRITE (8,90)	C CHECKING FOR OUT OF RANGE POINTS AT SAME TIME
WRITE (8,100)	C
	IX = IAX + J TY - TAY
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
30 FUNAT (Frogram : MATKIX.V')	
	LF (LT .GT. A) THEN
50 FORMAT (Matrix :, 113)	
	11 F 11 T 1 TT 170 CP 41 THPN
70 FORWAT (' Threshold :', LU.1) An someth ': Allocation's F6 31	LF (13 .447. N) 1060 TY = 1
BU FURMAT (CONDECTATION CONTRACTOR CONTRA	46 - 4 12 m 12 + 1
C. DURAN . J. MARING D. C.	
I DO FORMAT (*)	

-

AT ONS	AI CINE
C V=0 at IX=ICIM and V=VTOT at IX=ICOUT	C If all points are within THRESH of the last iteration, stop
C IF ((IX .BQ. ICIN) .OR. (IX .BQ. ICOUT)) GO TO 20	C Stop calculating if too many iterations done C
C SIM = SIM + S(IX.IV.IZ.]) + (V(IX +].IV.IZ) - V(IX.IV.IZ))	IF (ITER .GT. INAX2) THEN TTEP - TMAXIT
SUM2 = SUM2 + S(IX, IY, IZ, I)	CHEBY = -1.
SUM = SUM + S(IX - 1,IY,IZ,1) * (V(IX - 1,IY,IZ) - V(IX,IY,	RETURN
SUM2 = SUM2 + S(IX - 1, IY, I2, 1) C	IF (FLAG. EQ. 1) GO TO 10 C
IF (IY .LT. M) THEN	C Calculate the overall conductance, with different
SUM = SUM + S(IX,IY,IZ,2) * (V(IX,IY + 1,IZ) - V(IX,IY,IZ)) SUM2 = SUM2 + S(IX.IY.IZ.2)	C current vector this will be calculated differently C
	CURR = 0.
IF (IY .67. 1) THEN	DO 70 IY = 1, N
SUM = SUM + S(IX,IY - 1,IZ,2) * (V(IX,IY - 1,IZ) - V(IX,IY,	DO 60 IZ = 1, L DV - UPDOT - V/M - 1 IV IZV
$x = \frac{1}{2}$ SUM2 = SUM2 + S(IX, IY - 1, IZ, 2)	CURR = CURR + (DV*S(M - 1,IY,IZ,1))
END IF	60 CONTINUE
IF (IZ .12 .12 .12 .12 .12 .12 .12 .12 .12 .12	70 CONTINUE
SUM2 = SUM2 + S(IX, IY, IZ, 3)	U
END IF	CHEBY = CURR / VTOT
IF (12.67.1) THEN SIM - SIM - SITY IV 12 - 1 3) + JVITY IV 12 - 1) - JVITY IV	ITER - ITER / 2 C
III) = 200 + 2(14) + 1(14) + 1(2) = (1(14) + 1) + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 +	National
SUM2 = SUM2 + S(IX, IY, IZ - 1, 3)	QNZ
JI ONS	0.0
C IF (SUM2 . NE. 0) THEN	. ***********************************
SUM / SUM2	C Now set up the initial voltage condition
	C Each iteration will use the last one as a starting point
AI CM3	C unis may netp to speed unings up C théithéithéithéithéithéithéithéithéithéi
C SO NOW STODE OUD VALUE, AND MEN VALUE	C SUBBOOILTTHE RESERV
CORR = SUN * ONDERA	DIMENSION V(50,50), S(50,50,3)
IF (ABS (CORR) .GT. THRESH) THEN FIAG = 1	COMMON /MATRIX/ V, S Common /vars/ S1 S2 P1 Comm
	COMMON /VAR2/ M, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
V(IX, IY, IZ) = V(IX, IY, IZ) + CORR	
ZU CONTINUE 30 CONTINUE	DO ZU M = 1, L DO 10 I = 2, M - 1
40 CONTINUE	V(I, J, K) = VTOT / NPTS
50 CONTINUE	10 CONTINUE
C C Dia Chabuahaw acceleration to find new omega	V(1,J,K) = 0. V(M.J.K) = VTOT
C USE CIREDYSTEY SUCCESSION IN THIS HER SHEYS	20 CONTINUE
IF (ITER . EQ. 1) THEN	30 CONTINUE
OMEGA = 1. / (15*RHO*RHO) Et se	
IF (ITER .GE. IMAXI) THEN	END
OMEGA = 1.	υ
	C AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
OMEGA = 1. / (12)"UREGATKHUTKHU) End IF	С жет up the conductance attay and store it С анажанаталалаланыналалалалалалалалалалалалалалал

```
COMMON /VAR2/ N. M. L. NPTS, NN, RHO, ICIN, ICOUT, VTOT
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            COMMION /VARS/ S1, S2, P1, COMM
COMMION /VARZ/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
                                                                                             COMMON /VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
                                                                                                                                                                                 DO 10 IC = 1, 3
S(IX, IY, IZ, IC) = SVAL(S1, S2, P1, CONN)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 WRITE (6,10) (S(J,1Y,1Z,1),J=1,M - 1)
Format (8f12.8)
                                            DIMENSION V(50,50,50), S(50,50,3)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 DIMENSION V (50, 50, 50), S (50, 50, 3)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            DIMENSION V (50, 50, 50), S (50, 50, 3)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   DO 20 IY = 1, N
WRITE (6,10) (V(IX,IY,J),J-1,L)
FORMAT (8F12.8)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 COMMON /VARS/ S1, S2, P1, COMM
                                                                           COMMON /VARS/ S1, S2, P1, CONN
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              PRINT *, 'Horizontal'
DO 30 IZ = 1, L
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              COMMON /MATRIX/ V, S
                                                             COMMON /MATRIX/ V, S
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              COMMON /MATRIX/ V, S
                                                                                                                                                                                                                                                                                                                        S(1, IY, IZ, 2) = 0.
                                                                                                                                                                                                                                                                                                                                         S(1, IY, IZ, 3) = 0.
                                                                                                                                                                                                                                                                                                                                                         S(M, IY, IZ, 2) = 0.
                                                                                                                                                                                                                                                                                                                                                                        S(M, IY, IZ, 3) = 0.
                                                                                                                                                             DO 20 IZ = 1, L
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           SUBROUTINE SPRINT
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                SUBROUTINE VPRINT
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               DO 20 IY = 1, N
                                                                                                                                                DO 30 IY = 1, N
                                                                                                                                                                                                                                                                                                        DO 50 IZ - 1, L
           SUBROUTINE SETUP
                                                                                                                                DO 40 IX = 1, M
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    DO 30 IX = 1, M
                                                                                                                                                                                                                                                                                        DO 60 IY = 1, N
                                                                                                                                                                                                                     CONTINUE
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       10 FORMAT (8
20 CONTINUE
30 CONTINUE
                                                                                                                                                                                                                                     CONTINUE
                                                                                                                                                                                                                                                     CONTINUE
                                                                                                                                                                                                                                                                                                                                                                                          50 CONTINUE
60 CONTINUE
                                                                                                                                                                                                                                                                                                                                                                                           CONTINUE
                                                                                                                                                                                                                                                                      40 CONTINUE
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        RETURN
                                                                                                                                                                                                                                                                                                                                                                                                                                             RETURN
                                                                                                                                                                                                                                                                                                                                                                                                                                                               ONA
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          2
                                                                                                                                                                                                                     9 2 8
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  υ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    υυ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            υ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           U
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  υυ
                                                                                                                                                                                                                                                                                                                                                                                                                              υ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 U
                                                                                                                 υ
υ
                            υ
```

```
DO 50 IZ = 1, L
WRITE (6,40) (S(IX,J,IZ,2),J=1,N - 1)
FORMAT (8F12.8)
                                                                                                                                                                       DO 80 IY = 1, N
WRITE (6,70) (S(IX,IY,J,3),J=1,L - 1)
                                    PRINT *, 'Vertical'
DO 60 IX = 1, M
                                                                                                                                             PRINT *, 'Lateral'
DO 90 IX = 1, M
                                                                                                                                                                                                 FORMAT (8F12.8)
20 CONTINUE
30 CONTINUE
                                                                                                       CONTINUE
                                                                                                                                                                                                               CONTINUE
                                                                                                                  60 CONTINUE
                                                                                                                                                                                                                         90 CONTINUE
                                                                                                                                                                                                                                                     RETURN
                                                                                           <del>2</del> 2
                                                                                                                                                                                                  2 g
                                                                                                                                  υ
                                                                                                                                                                                                                                          υ
                           υ
```

ONE

Appendix E

Data Acquisition Program

The data acquistion program was written in Borland's Turbo Pascal V for an 'IBM PC' compatible computer. The IEEE routines referred to throughout the program were supplied with the Scientific Solutions IEEE card and were converted from the original BASIC to Turbo Pascal by Dr.D.B.Lambrick.

The program is divided up into several units which perform the following functions

THERMAL - main program

MRDGLOB - set up global variables.

MRDUTILS - general purpose input/output utility routines.

DEFAULTS - sets up the default settings for the IEEE devices and program variables.

DEVICES - IEEE device routines for inputting and outputting data to various devices attached to the IEEE bus.

Functions and subprograms are utilised wherever they may assist with the structure. Meaningful variable, procedure and function names are used throughout the program. The data is displayed graphically as it is acquired. The program was written to run on an 'IBM PC' compatible 286 AT machine with 640K RAM, and colour VGA graphics board, fitted with a Scientific Solutions IEEE bus. The program should run on non-AT machines and machines without colour VGA graphics although not all configurations have been tested.

\$K 30000,0,65000}	coeff[3]:=1.1101416234e-5;
	<pre>coeff[5]:=4.3326955522e-10; coeff[6]:=-7.0226932006e-13; coeffinition of contraction in the coeff[6]:=-7.0226932006e-13;</pre>
(*** Program TiteMAL.PAS	COMIT[/]:M4.2/404163390-16; X1:w1:
<pre>* obtains data for thermal conductivity experiments *</pre>	k2:=temperature;
(* M.R.Delap 1989 *)	sum:=w1*coeff[1]+x2*coeff[2];
{*************************************	for loop:=3 to 7 do begin
	x:=2*temperature*x2-x1;
program thermal;	aum:=aum+coeff[loop]*x; *1.=*3. *2.=*:
vaas dafaulta.mrdolob.mrdutila.ieee.devicea.crt.oraob.doo:	
	CONVERT Family = 3100.
const	
version = 2.07;	
version_date='20/09/90';	
1	procedure hold(instring:str255);
var dummy : real;	begin
	RestoreCrtMode;
	vertein (instring) /
i mesa tourar onvert mereen ine senaor groes and temperature Maint tha fifted chabushev noivonail coefficients	epacetarikada (srac)taria). Satisratikada (srac)taria) -
If the DR9IC controller is set temperature. these diode voltage conversions should be canted.	
Inscion convert_voit (voitage:real)	procedure header;
COLI: ATTAVILLE OF FORT	
T T T T T T T T T T T T T T T T T T T	II COLOUF LIGHT TEXTBACKGFOUND (NEG) /
top: integer;	
dum : teat;	If colour then TextColor(White);
IT (Voltage=0) then sum:=400	writein(Thermal Conductivity data collection program');
The set of	WINGLY, 'VERSION', VERSION', VERSION', VERSION', VERSION', ALL STON, GATE);
1960/27:17:4=:16:17:4=:16:17:4=:16:27:4:17:17:17:17:17:17:17:17:17:17:17:17:17:	WILLEIN(DY M. R. DELDY);
1979979714150111000 -/TA251994144111000 (TPTSP-ASBE-141141100)	II COLOUF THEN TEXTBACKGROUND (LIGhtGray);
COBIL[]]=20011593] COBIL[]]=111.0120] COBIL[]]=−133.20406] 66(10)90 10311910405]	IT COLOUR TAGN TAKTCOLOY (BLACK); Hitalicit A bo set.
· P-0-P-0-P-1-147300 //00/P4-1771473000 //74/AP-02/P4-1707147000	
±1.5−1.4 •2.5 ±1.51	united ()
118 - 141 2018) 118 - 141 2014 2014 2014 2014 2014 2014 2014	
eur loop - 3 de la 12 de benin 1001 - 3 de 12 de benin	
	procedure title:
sum = = sum + coeff [] cop] * x;	begin
x1:=x2; x2:=x;	header:
end ;	writeln;
ends	writeln('Ensure that the apparatus is set up correctly');
if aum>400 then aum:=400;	writeln('Cbeck ');
if sum-0 then sum:-0;	writeln('l. The sample is in place');
convert_volt:-sum;	writeln('2. The thermocouple wires are ok');
end;	writeIn('3. The sample space is evacuated');
	WITCHIN(* ALL MCCESS ACE CONNECTED TO THE IEEE DOTT); suiteloite the Minitian to been to only the IEEE DOTT);
function convert Temo(temoerature:real):real:	MITTELD: AND ALMANAN NO DECEMBER ()
var.	a)ace;
coeff : Array[112] of real:	header;
x,xl,x2,sum : real;	end; [title]
loop: integer;	
restrictions and the first of the second secon	first start start start starts
{COETILIENTS TAKEN ITOM IITTEN ITN OTGET CHEDYSNEY POLYNOMIAL) ^^_ffill.=! AltAbA7757^_Aff[7].=-0 37]]\$3A7846-4:	runction auto_min(max,min:real): real; var
(F_3F5.571717.5.6_~:[7]11300 [.51/]366014.1=:[1]11300	41

dum2:=newd*power_of(10, (Int(temp))); newd:=trunc(power_of(10, dummy)); if (newd-1) then dvs:=1; if (newd-2) then dvs:=2; if (newd-3) then dvs:=2;5; if (newd-9) and (newd<=7)) then dvs:=5; if (newd>=8) then dvs:=10; ans:=dvs"power_of(10,(Int(temp))); if (min+2*ane)*max then ans:=ans/10; Xmin:=1E34; Xmax:=-1E34; Ymin:=1E34; Ymax:=-1E34; for loop:=1 to point do temp:=log(Abs(min)); function auto_step(max,min:real): real; duany:=Frac(temp); if (min+3*ans)>max then ans:=ans/5;
if (min+4*ans)>max then ans:=ans/2; if min<>0 then sign:=min/Abs(min) newd:=trunc(power_of(10,dummy)); if (newd=0) then dvs:=1; if (dum2=max) then dum2:=max-1; : integer; dum2:-min; step, temp, dummy, dvs : real; : real: procedure find_max_and_min, else sign:--1; begin if point>2 then begin if sign>0 then begin : integer; <u>e 1 e e</u> dummy : =Frac (temp) ; step:=(mex-min)/8; temp,dummy : real; newd : integ sign, dum 2 : real; temp:=log(step); auto step:-ans; loop : integer; and; {auto step} auto min:-dum2; end; (auto_min) begin begin Dweu begin 808 VAF 1.00

{ акакананананананариекиди. Разалананананананананананананананананан

IEEE channel'); • .` • writeln('Device channel is ', device_default_chan[device]); device name ', nVcontrol.chan); ', Tcontrol.chan); writeln('Temperature controller ', Tcontrol.device,' ', nVcontrol.device, ', nVmeter.chan); ', nVmeter.device, ', Imeter.device, ', Vmeter.chan); ', Imeter.chan); ', Vmeter.device, writeln('Device name is ', device name[device]); procedure device_init_display(device: integer); for dummy:=1 to no_of_devices do
writeln(dummy,'....',device_name[dummy]); device no writeln('Device number is ',device); writeln('At device initialisation'); procedure device_init(device:integer); device_name[nVcontrol.device],' , device_name[Tcontrol.device],' if (device-dr91c) then init_dr91c; device_name[nVmeter.device],' if (device-k197) then init_k197; if (device-k175) then init k175; device_name[Vmeter.device],' device name[Imeter.device],' writeln('nV power controller (device init display(device)) procedure display_devices; writeln('Heater voltage and; {device_init_display} writeln{'Heater current writeln('Nanovoltmeter procedure display_meters; and; {diaplay_devices} writeln(' Meter dummy : integer; RestoreCRTmode; writeln; writeln; writela; apace; Cla: begin: begin begin begin end: N N N

if (device=solar) then init_solar;

if (device=fluke8860) then init fluke8860; if (device-weston) then init weston;

{device_init_display} end; {device_init}

if temp[loop]>Xmax then Xmax:=temp[loop]; if temp[loop]<Xmin then Xmin:=temp[loop]; if kval[loop]<Ymin then Ymin:=kval[loop]; if kval[loop]>Ymax then Ymax:=kval[loop];

end; (loop)

end: [if]

end; {find_max_and_min}

procedure initialise_devices; begin

init_ieee;

device init(Imeter.device);	if devices ok then
device init (Vmeter.device);	L begin
device init (nVmeter device);	write('Åre these meters and IEEE channels correct ? ');
device_init(Tcontrol.device);	device_setup_ok:=yes_or_no(true);
end; {initialise devices}	end
	elae
	begin
procedure device_defaults;	device_setup_ok:=false;
Var	writeln('These devices are incorrectly set up');
dumany : integer;	space;
begin	end;
zero_add;	header;
meter_defaults;	end;
(Note that meter defaults is external so that the may be easily	
changed without hunting out this section}	procedure change device parameters;
	var ok : boolean;
for dummy:=1 to no_of_devices do	being the second s
1001. davisidi.arim.utavice default chanidammul:	bacin back avtra-securptor, do
detta (dummu) inter-dette defuit community (header;
	display meters;
Imeter.chan:=device default chan[Imeter.device];	display_devices;
Vmeter.chan: -device_default_chan[Vmeter.device];	write('Enter the heater ammeter: 1 to ',no_of_devices:2,' ? ');
nVmeter.chan:~device_default_chan[nVmeter.device];	Imeter.device:=input_integer(Imeter.device,1,no_of_devices,false);
Tcontrol.chan:-device_default_chan[Tcontrol.device];	Lmeter.name:=device_name [Imeter.device];
nVcontrol.chan:=device_default_chan[nVcontrol.device];	Imeter.chan:=device_default_chan[Imeter.device];
Ineter . name : -devicename [Imeter . device] ;	beder /
Vmeter . name : =device_name [Vmeter . device] ;	display_meters;
nVmeter.name:=device_name[nVmeter.device];	display_devices;
Tcontrol.name:=device_name[Tcontrol.device];	writein(Heater anneter, device name (meter device));
nvcontrol.name:=device_name[nvcontrol.device];	WILCOID ('IEEE Channel'Imeter.chan);
end; {device_defaults}	WILCELDS Julie-Lite the TEPE channel of 2 'l'
	WITHOUT OF THE ATTOC CHARMENT ON : //
	OK: PAGE OF I TO (FLUER) (A fact c) that back
runction devices ok : boolean;	is not only the fame that the fame of the first of the fame of the fame of the fame fame of the fame fame of the f
Automatic sectors in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector is a sector is a sector in the sector in the sector is a sector in the sector in the sector in the sector is a sector in the s	Instant and the formation community of the set of the s
demmarytruct (f [tarter than-Wheter than) or [Tmeter chann/meter chan]	endz
ox (Tmeter chan-Tcontrol chan) the dumma '	header;
if (Wmeter.chan-NUmeter.chan) or (Vmeter.chan-Tcontrol.chan)	display meters;
or (nVmeter.cham-Tcontrol.chan) then dummy:=false;	display_devices;
if (Imeter.device=Vmeter.device) or (Imeter.device=nVmeter.device)	write('Enter the heater voltmeter: 1 to ', no_of_devices:2,' ? ');
or (Imeter.device-Tcontrol.device) then dummy:=false;	Vmeter device:=input_integer (Vmeter device, 1, no of devices, false);
if (Vmater.device=nVmater.device) or (Vmeter.device=fcontrol.device)	Vmeter name:sucorte name(weter device);
or (nVmeter.device-Tcontrol.device) then dummy stales;	Vmeter chan:=0evice_defauit_chan vmeter.device;
if (nvcontrol.chan=laster.chan) or (nvcontrol.chan*vmeter.chan)	
OF (NYCORIEOL.CHARTHOURGEGET.CHAR) OF (INCEEL.CHARTHOULEOL.CHART)	ulayiay_meetas uritala/Mastar u∩ltmater ′ device nume[Vmatar davice]}.
the distribution of the state o	HALLERI DECENT VALUECENT JULVALE INN VALUELIN Valtain/TRRE Thindelin''''''''''''''''''''''''''''''''''
ii (interticianty) force) or (numericianty) force) of then dummy:false; (frontrol-channer jees) or (numerer channer jees) then dummy:false;	Writein
devices okradiment	write('Is the IEEE channel ok ? ');
end: (meters commistered)	ok:=yes_or no(true);
	if (not ok) then begin
	write('Enter the ieee channel number: 1 to ', ieee no of channels:2,' ? ');
function device_setup_ok:boolean;	Vmeter.chan:=input_integer(Vmeter.chan,I, iono no of channels falsa).
diant of the second	header:
GISDLAY_meters;	

•

display meters;	devla [Vmeter.device].prim:=Vmeter.chan;
display_devices;	devta[Vmeter.device].prim:=Vmeter.chan;
write ("Enter the nanovoltmeter output device: 1 to ', of devices:2,' ? ');	devla[nVmeter.device].prim:=nVmeter.chan;
NVMMETER.coevice:wipur integer(numeter.cevice,1,no_or_gevices,talse); .utetter andreite americalandert davice).	Gevca(nvmeter.gevica).prim:=nvmeter.cnan; Aavis(Trontro) Aavica) oria:=Trontro) chan:
Avmeter.Atandevice.defanit.nverterj. Almater Atandevice.defanit.nhariaulmater davice).	devia (Trontrol device) primetrolicul chan; devia (Trontrol device) primetrol chan;
n mer en i trans. Fer vice "estant", fer vice i fer en i transfer i de vice i fer et el i	devla[nVcontrol.device].prim:=nVcontrol.chan;
display meters;	devta [nVcontro].device].prim:=nVcontrol.chan;
writeln'.'NVmeter device',device name[nVmeter.device]);	end; {alter_defaults}
<pre>writeln('IEEE channel,'n'meter.chan);</pre>	
writeln;	
write('is the IEEE channel ok ? ');	[#####################################
ok:=yea_or_no(true);	
if (not ok) then begin it.'.'.'.'.'.'.'.'.'.'''''''''''''''	arcondura usrishis defailte.
WIICE' EXCERTER STORE SAME AND A CONTRACTION AND AND AND AND AND AND AND AND AND AN	D. VEGULE VELIEVE VELOUIS
armeter.turn.turut_anteger.urger.turnets.turnets.false); idee no of channels.false);	duntime : time;
endi	dumstr : str20;
bender;	begin
display meters;	path:-default_path;
display_devices;	filename:-file_version(default_filename,path,'.DAT');
write('Enter the temperature control device: 1 to ', no_of_devices:2,' 7 ');	with min_time do begin
Tcontrol.device:=input_integer(Tcontrol.device,1,no_of_devices,false);	hour :=0;minute :=0;second :=1;day :=1;month :=1;year :=0;
Tcontrol.name:=device_name[Tcontrol.device];	end;
fcontrol.chan:-device_default_chan[fcontrol.device];	WICH max time do begin
beader; display meters;	hour:=23;minute:=59;second:=59;day:=31;month:=12;year:=99;
writein ('Temperature controller', device_name[Tcontrol.device]);	end;
writein ('IEEE channel', Tcontrol.chan);	
write ('ie the IEEE channel of 2 ');	neat reading time: main time;
territor of territorial territoria territorial territoria territoria territoria territoria territoria territorial territoria territoria territ	
II (not ok) then begin	
Write("Enter the idee channel number: 1 to ', leee no or channels:4,' f '); ••••••••1 **••••1 **•••••••••••••••••••	System_clock(oumtime); romentama_rlock(dumtime.clumatrl;
ICONCIOL.COMAN.TANGUCLICONCIOL.CON.LOL.CONCIOL.CONCIOL.CONCID. (Anos 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	compared to a transmission of the second s
	cost trading time:smit time:
	remerts remerts remerts and remerts
diacjav (avtess)	nv ontine: min time:
write("Enter the NY meter power switching device: 1 to '	point: -1;
no of devices:2, 2 ');	no of setpoints:=0;
nVcontrol.device:=input_integer (nVcontrol.device,1,no_of_devices,false);	<pre>setpoint_index:=0;</pre>
nVcontrol.name:-device_name[nVcontrol.device];	setpoint_time[0]:=min_time;
nVcontrol.chan:-device_default_chan[nVcontrol.device];	<pre>setpoint_temperature[0]:=1;</pre>
header; display_meters;	no more_setpoint_changes:=false;
Writein("N power switching, device_name[Icontrol.device]);	setpoints soften = talse;
writein'iEEE channel, nycontrol.chan); 	AV Old Scaule struge: Absorbing states and advents states
WILTG('IS TOP INGE CRADDEL OK / '); (b. team (t t(f)t))	thermocouple_iale:==tite:worowyie_usiautt_iiic; Var defaults:
if (not ok) then begin	Note that VAR DEFAULTS is in the unit DEFAULTS so that they may
write ('Enter the ieee channel number: 1 to ', ieee no of channels:2,' ? ');	be easily aletered without finding this routine!!!}
nVcontrol.chan:=input_integer(nVcontrol.chan,l, isse no of channels,false);	end; [variable_defaults]
end;	
end;	procedure display variable parameters;
lanain need to set device listen and talk addresses for IEEE routines, as	begin header;
they require a device number to be passed rather than listen and	writeln;
talk addresses.)	writeln{'Path for filename ', path);
devla[Imeter.device].prim:=Imeter.chan;	writeln('filename to store data ',filename);writeln;
devta[Imeter.device].prim:=Imeter.chan;	writein('Sample width * ', wigin:2,' mm');

	VAF
writeln("Thermocomple separation = "liendtheit", mm");writeln;	dumetr : atr20;
writein('Scale on nV meter in uV = ',nVscale:5:0,' uV');	default : time;
writeln('nv meter offset in uv	begin
writeln('Voltmeter fad on nV output = ',Vfad:8:6,' Volts');	header;
end; {display wariable parameters}	writeln;
	writeln('The system clock will run throughout the experiment');
	writein('Readings will be taken at constant intervals.');
runction Variable Darameters ok:boolean;	
	WILLEIN (ENCER LUE INCEVENT DELMEEN LEGGLINGS IN LUE TOIM MULTURY);
dispressions; dispressions;	dumatri = derault interval+null date;
WIIGLD, The start strength of the start of t	expand clock dumert perault);
WILLE ALE LIGE PLANELS ON CONTRACTOR	Anguage (action of the contract of the second of the secon
Variable parameters ok	writeln("Enter the delay after chanoing set point in the form HH.MM.SSY).
	Autoril michaelie actag arter cianying act point in the toim michaely i Anader-readenit annotaell dere cianying
	oumstr:edetaurcfbauernui.care; evenand /lock(humert lafeit);
successive successive states and the succes	tanut time(states) as as a state for a sta
barin Jarin	
ustration (NOT variable parameters ok) do	writeln:
begin	system clock(start time);
display variable parameters;	compress clock(start time, start str);
writeln	writeln('The start time of this program was ',start_str);
path:-get_pathname(path);	repeat
display_variable_parameters;	writeln('Enter the time at which data collection is to end');
writein;	writeln('Enter the time in the form HH:MM:SS use >24hrs if necessary');
filename:-get_filename(filename,path,'.DAT',start);	dumstr:=copy(start_str,9,9);
display variable parameters;	dumatr:=default_end_time+dumatr;
writeln	expand_clock(dumstr,default);
write('Enter the width of the sample: 0 to 50mm ? ');	input time (default, end time, false);
width:=input_real(width,0,50,false);	compress clock (end_time, dumstr) ;
display variable parameters;	writeln("The program will finish at ', dumstr);
writeln	until compare time(end_time,start_time);
write('Enter the thickness of the sample: 0 to 50mm 7 ');	space;
thick:=input_real(thick,0,50,false);	end; {enter_intervals}
display_variable_parameters;	
Write/1910; write/1910 the distance between the thermonomia wither 0 to 50mm 3 (1.	
Marten Estive traducture accessing for falacity with with a contract of the second state of	
viteln: Viteln:	
areassedidththththth	procedure display graphics variables;
display_wariable_parameters;	begin
writeln;	header;
write('Enter the scale on the nanovoltmeter: 0 to 1000 uV ? ();	writein;
nvscale:=input_real(nvscale,0,1000,false);	writein 'Auto rescale', auto scale);
display_variable_parameters;	writein (Minimum temperature value plotted 'Xmin:6:3' K);
WIIGUN Wiitelus the ennum (ffert on the seconditurber: _1000 to 1000 vV 2 1.	MILELN' MAXIMUM Cemperature Value Dioffed ',XMAX:0:3/' K'); (///ell.
willer bilder the menual virawit di the hallovothert svov tv avvo v	Willwini uritaint'Minianm K value sintted ''Ymin:6:3.' M/m/K');
ranicas Jontaverargusargusargusargus	writeln('Minfaum K value plotted ',Ymax:6:3.' M/m/K'):
writeln;	Writein
writeln('Enter the voltage for fad from the nVmmeter analogue output');	end;
write('From D to 10 volts ? ');	
Vfsd:=input_real(Vfad,0,10,false);	
header; and. [uhi]a]	function graphics_variables_ok : boolean; baria
	detail diadimentation unrishian.
	urplay_graphics_variaties, writeln;
	if ((Xmax <xmin) (ymax<ymin))="" or="" td="" then<=""></xmin)>
procedure enter_intervals;	begin

-

-

graphics variables ok:=false;	end; {initialise graphics variables}
end I I	
else	
begin	₹₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
write('Are these variables ok ? ');	
graphica_variables_ok:=yea_or_no(true);	
end;	procedure list_setpoints;
end; {graphics_variables_ok}	Var
	dummery : integer;
	Verstr : Brizo; Verst
	beader: writeln:
dummay : integer;	writeln('Point no Set time Set temp');
begin	for dummy:=1 to no_of_setpoints do begin
while (NOT graphice_variables_ck) do	if ((dummey mod no_of_lines)=0) then begin
begin	space;
writeln;	header; writeln;
write('Do you wish to have auto_scaling 7 ');	writeln('Point no Set time Set temp');
auto_scale: "yes_or_no (auto_scale);	end; [1f]
IT (NOT BUTC_SCALE) Chen	compress_clock(setpoint_time(dummy), setstr);
begin	writeln(' ',dummy,' ',setstr,' ',
display_graphics_variables;	setpoint_temperature[dummy]:7:3);
write("Enter the minimum display temperature: U to 300K 7 ');	end; (dumury)
XMLATE INDUCT FEAL(ARDIT, U. 50U, TALBE); Lateriated and the second s	II (no of setpoints mod no of lines)<>0) then space;
alapiat to and a strattas; sufficient to and a strattas;	
WILE(Sheet the maximum display temperature: 'Aminibit' to book ? '); Versi-taony sail(Wersi Versi 200 6:1).	eud; 11st_serbourcs}
Annas - Iniput Leal (Amas, Anni) Jou, Idiac() Aidaniau Atanhira astian)as:	
state("Store the minimum display value of K in W/m/K ? '):	procedure sort setucints:
Ymin:=inout real(Ymin.0.1000.false)	
display graphics variables:	loopl.loop2 : integer:
witte("Enter the maximum display value of K in W/m/K ? ');	dummy : str20;
Ymax:=input real(Ymax,Ymin,1000,false);	dummyreal : real;
end; {if}	dumtime : time;
end; (while)	<pre>{Noddy bubble sort - this can be made more efficient!}</pre>
if ((Xmin-Xmax)-0) then	begin
begin	writeln;
zir-rinxzwar	writeln('Sorting the setpoints into order this may take a while');
Xstep:=screen_X_axis/auto_step(Xmax,Xmin);	for loop1:=1 to no_of_setpoints do
	begin fersionst h. so of control the de
	ALL TOPPALET TO DO DI SELPOINTS OF
Same: = Same: - Sam	ty compare time [actroint time[loos], setroint time[loos]] then
Xstep:"screen_X axis/(Xmin-Xmax);	begin
end;	dumtime: "setpoint_time[loop1];
if ((Ymin-Ymax)=0) then	setpoint_time[loop1]:*setpoint_time[loop2];
Ystep:=screen_Y_axis/auto_step(Ymax,Ymin)	setpoint_time[loop2]:=duntime;
0.00	dummyreal:=eetpoint temperature[lop1];
I AND I A	serpoint temperature[100p1]:=serpoint temperature[100p2];
	end; end;
	end; {loop2}
procedure initialise_graphics_variables;	end; {loopl}
Dedin nealar-falaa.	set points sorted:#true;
al () w (ai te - tai we) Met o. = () . Yeary e 210 ()	
Autori-O; Autoaki-200, Ymin:=0; Ymaki=5;	
Xstep:=screen X axis/(Xmin-Xmax);	procedure get time(index : integer;var ok : boolean);
Ystep:=screen Y axis/(Ymin-Ymax);	

.

cnange graphics_variables; end; {initialise_graphics_variables}

	start_index : integer;
dumstr : str20;	loop_start_temp,loop_temp_interval : real;
default, answer : time;	dumey2 : real;
oer werpoint_change : time/	cumacr : stroughter : stroughte
headerj	header;
writeln;	writeln;
writeln('Set point change number :',index);	writeln('The temperature controller can be programmed to change the setpoint');
writeln('Enter the time in HH:MM:SS or -1 to end');	writein('at particular times. You must specify the times at which the');
normalise_clock(start_time);	writeln('controller is to change the setpoint');
compress_clock(start_time,dumstr);	writeln;
clock1:=copy(dumstr,1,8);	writein('Enter the times in the form HH:MM:SS, if you wish to go');
clock2:=copy(dumstr,10,8);	writeln('over to the next day, use hours greater than 241 ;
writeln('The starting time was ', clock!);	writeln('For stample,');
writein('The starting date was ', clock2);	Write-in('11.30pm is 23:30:00' 1.30am on the following day is 25:30:00'};
writein	WITCELD;
WICH der serpoint change do begin	Writein('rou may program the computer to change setpoints at regular');
nour: -desaurt four nour;	WILLELN('INCEIVAIS, UNIS EASES LYDING.');
まれつしてき、このをはして、きでいた。まれの1、まれの1、またので、「また」の1、「「」」」。 「「」」」」」 しょう	WILLERING Maintening and the anter the anteriat time of areasist time (and the second 2.1).
actorial activative sectorial sectorial	WILE VO YOU WHEN NO ENCER THE SECTOINT TIME AT CONSTANT TIME INTERVALS ();
CH4:=0: BOACH:=0: Year:=0;	CONSTANT_TIME:=Yee_Or_no(CONSTANT_LIMe);
end; 	
and the start time out setpoint change, default);	
input time (delault, answer, true);	
It answer nour-200 then okimise is a second se	
If ok then begin	start_index:=no_of_setpoints;
DOTMALISE CLOCK (ANSWER);	While (((no_of_setpoints+1) <max_setpoints) and="" do<="" ok)="" td=""></max_setpoints)>
setpoint time (index) == nswer;	
compress clock(setpoint time(index)/dumarr);	
WILE AL (DUBSER)	IT CONSTANT TIME) THEN
end; [get_time]	get time (no stateburg, ok);
	and: of then get_temperature (no_of_setpoints, ok, 100);
· [[15 (acres) by an arrival three
	the state of the s
autor : atr20:	usti diamav da bacin
	hour:=0; second:=0; day:=0; month:=0; vear:=0;
beader;	end;
writeln;	dummy.minute:= (no of setpoints-interval start index) *interval time;
writeln('Setpoint change number ', index);	add time threeval start time dummy setpoint time no of setpoints]);
compress clock(setpoint time[index],dumstr);	normalise clock(setpoint time[no of setpoints]);
writeln("Setpoint change time ', dumstr);	compress clock(setpoint time[no of setpoints);
writeln;	writeln('Next setpoint change is at ',dumstr);
write('Enter the setpoint at this time : 0 to 300K ? ');	dummy2:=((no_of_setpoints-interval_start_index)
<pre>setpoint_temperature[index]:=input_real(default,0,300,true);</pre>	*loop_temp_interval)+loop_start_temp;
<pre>if (setpoint_temperature[index]=-1) then ok:=false;</pre>	get temperature (no_of_setpoints, ok, dummy2) ;
end; [get_temperature]	end;
	if (constant_time and (NOT started)) then
araadiira aataa aataa nta.	utbegin untraffErter the interval (in minutea) between estructor charace 2 ().
procedure enter_artivutica;	MILICOL BUICH LUM LICE LICETVEL III MILUCCEO L'ECTRETE CLERINGE : r Interval fime-sinnui (interve (édésuit sinterve) time.
ur interval start index : integer:	
interval time : integer;	writeln;
dumuytime;	writeln('Enter the time for the interval loop to'.
answer : str8;	' start in NH:NM:SS');
ok, constant_time, started : boolean;	get_time (no_of_setpoints, ok);
clockstr : str20;	interval_start_index:=no_of_setpoints;
interval_start_time : time;	interval_start_time:=setpoint_time[no_of_setpoints];
clockl,clock2 : strl2;	no of setpoints:=no of setpoints-1;

beader: writeln:	
writeln('Loop start point is ', interval start index);	write("anter the point to change: 1 to '.no of setucions:3.' 7.').
compress clock(interval start time.dumstr):	dummovations interest() no of activities (state)
writeln("coop start time is "dumstr";	eternia - street treeder / / / / 10000 / 10000 / 10000 / 10000 / 10000 / 10000 / 10000 / 10000 / 10000 / 10000
uritalo:	and the second
Write(Futer the start temperature of the loop: U to 323K 7 7);	dumstr:=copy(dumstr,10,8);
loop_start_temp:=input_real(100,0,325,false);	writeln('Start date ',dumstr);
writein/	compress_clock (setpoint_time[dummy], dumstr);
write (Enter the temperature step of the loop: -325 to 325 K ');	write('point no = ',dummay,' Time = ',dummatr,
loop_temp_interval:=input_real(-1,-325,325,false);	<pre>/ setpoint ', setpoint_temperature[dummy]:7:3,' ? ');</pre>
end; {if}	flag:=yes_or_no(true);
header;	if ((dummy<=no of setpoints) and flag) then
end; (while)	begin
no of setpoints:=no of setpoints-1;	setpointe sorted:=false:
header:	aet time(dummv.ok);
if (start indexcome of setucints) then sort setucints:	- 1011 de rementer de la companya de la comp
is first controlled between our over over over over over over over ove	er remperature (ourmy, ok, 100);
II Wood activity for a solid of the activity act	end; [II]
Dieste serpoint canges raise	SOFT SECTOINTS;
end; [enter_setpoints]	end; [If]
	if (answermlist) then list_setpoints;
	until (answermo of choices);
procedure alter_setpoints;	1f (NOT setpoints sorted) then sort setpoints;
const	end; (alter setpoints)
delete = 1;	
add = 2;	
chance = 3:	
answer, dumany, loop : integer;	procedure input_gain_variation(index:integer;var ok : boolean);
no conces : inceger;	Degin
ok,fl ag : boolean;	header; writeln;
dumstr : str20;	writeln('Gain variation number ', index);
begin	write('Enter the lower temperature of the band: 0 to 325K ? ');
no_of_choices:=5;	temp_variation_lower[index]:=input_real(10,0,325,true);
repeat .	if (temp_variation lower[index]=-1] then ok:=false;
header; writeln;	if ok then begin
writeln('Setpoint options list');	write('Enter the upper temperature of the band: 0 to 325K 2 ');
writeln;	temo variation under[index]====================================
writeln('1 delete setpoint');	
writeln(/2add setwointe/).	setteletisticate the seteness of a setenes.
	WITCHING AND
	Write From , Min gainibit, to , max gaines?, 7);
	gain_eriation[index]:=input_real(i.).min_gain.max_gain,false);
WILCELN()	writein("Knter the rate variation for this band');
I I C C I LATIN ADDITION DOLLAR ADDITION	WILCO FROM AND FACOSSY, TO , MAX FACOSSY, 7);
answer:=inpur_inreger(),i,),raise);	rate_variation[index]:=input_real(0.0.min_rate,max_rate,false);
header;	writein('Enter the reset variation for this band');
	Write('From ',min_reset:5:2,' to ',max_reset:5:2,' ? ');
II (answermdelete) and (no of setpoints>=1)) then 	reset variation (index) : = input_real (0.0, min_reset, max_reset, false) ;
	Write('Enter the heater range for this band: '
write ("Enter the point to delete: I to ', no of setpoints:3,' ? ');	<pre>min_heater_power:2,' to ', max_heater_power:2,' ? ');</pre>
dummy:=input_integer(v,ino of_setpoints;sialse);	heater_range_variation[index]:=input_integer(4,min_heater_power,
	max_heater_power, false);
for loop:=dummy to no_of_setpoints do	end; [if]
begin	end; {input_gain_variation}
setpoint time [loop] :=setpoint time [loop+1];	
setpoint temperature[loop]:=setpoint_temperature[loop+1];	
	procedure list_gain_variations;
	Var
11 (answer=add) then enter_setpoints;	loop : integer;

Max temp 30.000 if (answer=change) then writeln; ok:=true: end; (loop) end; {if} end; (if) writeln('Min temp writeln(' 20.000 begin end; [if} begin begin writeln; writeln; writeln; header; header; writeln; writeln: writeln: space; repeat end; Heater power'); Heater power'); ', heater_range_variation[loop]); writeln('Temperature controller gain settings'); writeln('The temperature controller can be programmed to change the'); gain_variation[loop]:5:2,' ', rate_variation[loop]:5:2,' ', .` start_index:=no_of_gain_variations; while (((no_of_gain_variations+1)<max_gain_variations) and ok) do writeln('Enter -1 for the lower temperature limit to end.'); ((', loop, ', temp_variation_lower[loop]:7:3,'
temp_variation_upper[loop]:7:3,' writeln('At the prompt, enter the value you wish to use.'); if ((no_of_gain_variations mod no_of_lines)<>0) then space; writeln('Temperature controller gain settings'); writeln; Reset Reset no_of_gain_variations:=no_of_gain_variations+1; input_gain_variation(no_of_gain_variations,ok); end; (while) writeln('Band Min temp Max temp Gain Rate Rate no_of_gain_variations:=no_of_gain_variations=1; for loop:=1 to no_of_gain_variations do begin if ((loop mod no of lines)=0) then begin header; writeln; Gain reset_variation[loop]:5:2,' Max temp writeln; end; {if) space; procedure enter_gain_variations; procedure alter_gain_variations; : integer; answer, dummy, loop : integer; boolean; end; {enter_gain_variations) end; (list_gain_variations) writeln('Band Min temp : boolean; start index : integer; writeln(' ', loop, no_of_choices:=5; header; writeln; no of choices .4 end: {loop} delete = 1_j change = 3; ok:=true; header; writeln; writeln; writeln; header; ok, flag header; header begin add list begin begin const Var Var ĕ

writeln(dummy,' ',temp_variation_lower[dummy]:7:3,temp_variation_upper[dummy]:7: write('Enter the point to delete: 1 to ', no_of_gain_wariations:3,' ? '); write('Enter the point to change: 1 to ', no_of_gain_variations:3,' ? '); heater_range_variation[loop]:=heater_range_variation[loop+1]; reset_variation[dummy]:5:2,heater_range_variation[dummy]); temp_variation_lower[loop]:=temp_variation_lower[loop+1]; input_gain_variation(dummy,ok); rate reset heater range'); gain_variation[dummy]:5:2,rate_variation[dummy]:5:2, writeln('This allows you to make allowance for the change in'); writeln('time constant, sensor sensitivity and cooling power'); writeln('the temperature band within which it is operating.'); writeln('gain, rate, reset and heater power depending upon'); if (dummy<=no_of_gain_variations) and flag) then</pre> temp_variation_upper[loop]:=temp_variation_upper[loop+1]; writeln('Note that there are defaults already programmed'); if ((answer-delete) and (no_of_gain_variations>=1)) then reset_variation[loop]:=reset_variation[loop+1]; dummy:=input_integer(0,1,no_of_gain_variations,false); gain_variation[loop]:=gain_variation[loop+1]; rate_variation[loop]:=rate_variation[loop+1]; no of gain variations:=no of gain variations-1; 5'); dummy:=input_integer(0,1,no_of_gain_variations,false); write('Choose the option you require: 1 to 5 ? '); for loop:=dummy to no_of_gain_variations do 4.50 writeln('as the temperature falls. For example'); if (answer=list) then list_gain_variations; writeln('1... delete a temperature band'); writeln('3... change a temperature band'); if (answer-add) then enter_gain_variations; writeln('5... end this section'); writeln; writeln('2... add a temperature band'); 3.40 writeln('4... list temperature bands'); writeln('Temperature band selection'); answer:*input_integer(5,1,5,false); if (dummy<>0) then begin flag:=yes_or_no(true); gain 1.30 if (dummy<>0) then begin

	<pre>if (sensitivity=0) or (sensitivity<-100) or (sensitivity>100)</pre>
end: latter_dain_variations/	then sensitivity:"L";"L";"L";"L";"L";"L";"L";"L";"L";"L"
	end: {convert nv}
procedure set_gain_etc(temperature:real);	
V&E	
loop : integer;	function calculate_k(index:integer): real;
begin	begin
for loop:=1 to no_of_gain_variations do begin	if (dT[index]-D) then
if {{temperature>temp_variation_lower[loop]}	calculate_k:=0
and (temperture <ttemp_variation_upper[loop])) begin<="" td="" then=""><td>else calculate_k:=current[index]*V[index]*l]ength*</td></ttemp_variation_upper[loop]))>	else calculate_k:=current[index]*V[index]*l]ength*
write gain (Tcontrol.device, gain_variation [loop]);	IE-3/ (area*1E-6*dT[index]);
delay (100);	end; {calculate_k}
<pre>write_rate(Tcontrol.device,rate_variation[loop]); delinition</pre>	
detay(10)//	
WILLE FEBEC(ICONLYOL.GEVICE,FEBEC_VALIATION(LOOP)); Jain://o/	procedure open_ille;
ution() () () () () () () () () () () () () (187 6 • • • • •
MAILE BEAGE 1 41196 (10011101.0004109,10001101.0001)/	
	Cutilite Points / Fillename;
	Assign (f. oucrile);
600)	
	WITHCHAIR, JURA ACQUITED : ,SCAR SKI) Withchief (The 61)
	WITCHILL, UNTER TILE
	writern(r, Fach : 'pach);
	Writein(f. No radiation corrections');
· · · · · · · · · · · · · · · · · · ·	Writein(f,'nV offset uV : ',manual_offset:6:3);
[Following procedure looks up thermocouple table to convert dV to dT]	writeln(f,'nV scale uV : ',nVscale:7:3);
	writeln(f,'Vfad on meter : ',Vfad:7:6);
procedure load_thermocouple;	writein(f,'Sample size (mm) ');
Var	writeln(f,width:6:2,' ',thick:6:2,' ',llength:6:2);
f : text;	writeln(f,'Temperature (K)');
counter : integer;	writeln(f,'Thermal Conductivity (W/m/K)');
begin	close(f);
Assign(f,thermocouple_file);	end;
Reset (f) ;	
counter:=1;	
repeat	procedure store points (index:integer);
read(f, aufe[counter], aufe temperature[counter]);	Var
counter: =counter+1;	f : text;
until Eof(f);	dummv.loco : intecer:
Close (f);	bedin
end; (load thermocouple)	assign (f. outfile) ;
,	append (f) ;
	dummy:=index-no points in block+1;
function convert_nV(temperature,value:real):real;	for loop:=dummy to index do
Var	writeln(f,temp[loop]:9:4,' ',kval[loop]:9:5,' ',current[loop]*1E3:7:5,
counter : integer;	' ',V[loop]:9:6,' ',dV[loop]:9:6,' ',dV[loop]:8:4,
volt1,volt2 : real;	<pre>' ', setpt[loop]:8:3,' ', power[loop]:3:1,' ',</pre>
templ,temp2 : real;	gain(point):5:2,' ',rate(point):5:2,' ',
sensitivity : real;	rreset [point]:5:2);
begin	close(f);
counter:=1;	end; {store_points}
While (temperaturesaure_temperature(counter)) do	
courter:=courter+1; wolt:==ufactor:veta:	<pre>procedure plot_point(xval,yval:real);</pre>
VOLL1-244444	Var Y V · interati
<pre>voic: ************************************</pre>	dummer : integer;
temp2:maufe_temperature[counter]];	begin
<pre>sensitivity:= (volt1-volt2) / (temp1-temp2);</pre>	dummy:≖round((xvel-Xmin)*Xstep);

Y:=round((vval-Ymin)*Ystep);	procedure ger_next_reading_time; var
Y:=Y+yborder_bottom;	dumtime : time;
if colour then SetColor(Green);	clock : time;
Movero(X-3,Y); Linero(X+3,Y);	begin
Movero(X,Y-3); Linero(X,Y+3);	system clock (clock);
end; {plot_point}	add_time (clock, interval, dumtime);
	normalise_clock(duntime);
	<pre>if ((compare_time(dumtime,next_setpoint_time)) and</pre>
procedure get_setpoint_change_time;	(NOT no more setpoint changes) then
	accaliae clock/dimetime).
Clock : Line; fischad : Doolban:	is consist function (dimition, and time) then dumitime: wax time: (f consists time(dimition, and time) then dumitime:
	normalise clock (dumtime);
setpoint index:=0;	next reading time:-dumtime;
system clock(clock);	dumtime.minute:=duntime.minute-nV_minutes_on;
flagged:=false;	dumtime.second:=dumtime.second=nV_seconds_on;
{keep going until time now>set(index) but < (index+1)}	normalise_clock(dumtime);
repeat	nV_ontime:-dumtime;
begin being and being	end; [get_next_reading_time]
II (accpoint_indervio_or_accpoints) then bardin	
vey: naxt actorint temperature:=setnoint temperature[setnoint indext]];	procedure display text(X,Y;integer;instring:str255);
next setpoint time=setpoint time (setpoint index+1);	Yer.
end (if)	X1,Y1,X2,Y2 : integer;
else	high,long : integer;
begin	begin
setpoint_index:=no_of_setpoints;	SetTextStyle (DefaultFont, HorizDir, 1);
flagged:=true;	high:=TextHeight (instring);
no_more_setpoint_changes:=true;	long:=TextWidth(instring);
with next setpoint time do begin	
nour:=200;#hnufe:=200;#econd:=200;day:=200;day:=200;year:=200; 	XI:=X+LONG; IZ:=I; 4 YI-Y1aft than Y1Y1aft.
	AF ALANCET CLUER ALL ALANCET. AF YOSYTAAR THAN YOSHTAA
	if Y1 <ytop them="" y1:="Ytop;</td"></ytop>
if (compare time(clock,setpoint time[setpoint index])) and	if Y2>Ybottom then Y2==Ybottom;
(NOT compare time(clock, setpoint time[setpoint index+1])) then	SetViewPort (X1, Y1, X2, Y2, ClipOn);
flagged:"true;	ClearViewPort;
<pre>setpoint_index:=setpoint_index+1;</pre>	SetViewPort (0, 0, GetMaxX, GetMaxY, ClipOn);
until flagted and fastonich channe timel	OUCTEXCAT(X,I,INSTITUG); and: [display text]
procedure change_setpoint;	procedure set_nV_status(status,write_text:boolean);
begin	Var Var
if (next_setpoint_temperature>0.0) then begin	utility atr255
last_setpoint:=next_setpoint_temperature;	X,Y : integer;
<pre>write = setpoint(Tcontrol.device,next_setpoint_temperature); </pre>	begin if write text and colour thes Catrolor(White).
feast (roug) if anto nain than ast nain strinaxt satunint teamaratura):	is status then being used of a status that the status that be a status that being a status that be a status
	utoto on the dumstriedal (minicam address.minicam on);
if (NOT no more setpoint changes) then	dumatr:='On ';
	end
pause_flag:=true;	else begin
add_time (next_setpoint_time, pause, free_time); 	dumstr:=dal{minicam_address,minicam_off); dumetr:='Off'.
rat servint channe time:	
end: (channe settooint)	if write text then begin
	<pre>X:=xleft+(xborder left div 4);</pre>
	Y:=data line3;

| <pre>itr2); volt:=volt+read_volt(Vme</pre> | <pre>/tr2); volt:=volt+read_volt(Vmet
:r2; volt=volt+read_nV(nVmeter.de
end; [loop]
:r2); power[point]:=read_gain(Tcon
gain[point]:=read_gain(Tcon</pre> | <pre>itr2); volt:=volt+read_volt(Vmete</pre> | <pre>itr2); volt:=volt+read_volt(Vmete</pre> | <pre>/tr2); /tr2) /rolt:=volt:read_volt(Vmete /rolt:read_volt(Vmete.is2); /roop /roomer[point]:=read_peater_po /root /ro</pre> | <pre>itr2); volt:=volt+read_volt(Vmete</pre> | <pre>itr2); volt:=volt:=volttread_volt(Vmete
nV:=nV+read_nV(nVmeter.dev
end; {loop}
:r2); power[point]:=read_heater_pc</pre> | <pre>htt2); volt:=volt:=read_volt(Vmete:
r2; nV:=nV+read_nV(nVmeter.dev:
end; {loop} {loop} {c2; power[point]:=read heater power[power[point]:=read heater power[p</pre> | <pre>itr2); volt:=voltt=volttread_volt(Vmete:</pre> | <pre>itr2); volt:=wolt:=wolt:=wolt:=wolt:=wolt:=wolt:=wolt:wolt:
.r2; mv:=mv:=out_nv(nvmeter.dev
.r2);</pre> | <pre>htr2);</pre> | | \taktattttttttttttttttttttttttttttttttt | |
|--|--|--
--
--
--|--|--
---|--
--|--|--
--|--|
| <pre>voit:=voit=
voit:=voit=
atr+' ' +dumatr2;
dex]:4:1,dumatr2);
end; {loop}
dex]:4:1,dumatr2;
power[point]:
power[point]:</pre> | <pre>ndes1:3:0, dumatr2);</pre> | <pre>voict=rvoict voict=rvoict atr*^ * +dumatr2; atr*^ * +dumatr2; end; [loop] dex]:4:1,dumatr2); power[point]:= gain[point]:= dex]:4:1,dumatr2); power[point]:= dex]:4:1,dumatr2);</pre> | <pre>voict=rvoict voict=rvoict atr+' ' +dumatr2; atr+' ' +dumatr2; end; [loop] dex]:4:1,dumstr2); power[point]:</pre> | <pre>ndes1:3:0,dumatr2); voit:=voit= work=ad atr* ' +dumatr2; end; [loop] end; [loop] str* ' +dumatr2; gain[point]:= gain[point]</pre> | <pre>rout:=rout:
str+` ' +dumstr2;
dex]:4:1,dumstr2;
str+` ' +dumstr2;
power[point]:
dex1:4:1.dumstr2;
gain[point]:
dex1:4:1.dumstr2);</pre> | <pre>voict=rvoict voict=voict atr+' ' +dumatr2; nV:=nV+read idex]:4:1,dumatr2; end; [loop] cdex]:4:1,dumatr2; power[point]: </pre> | <pre>ndes1::0.dumatr2);</pre> | <pre>voit:=voit=
voit:=nvit=
wv:=nvit=voit=
wv:=vit=
voit=(iloop)
voit=(iloop)
voit=(iloop)
voit=(iloop)</pre> | voit:=voit=
atr+' '+dumatr2;
dex]:4:1,dumatr2);
end; [loop] | <pre>voic:=voict
str+' '+dumstr2;
dex]:4:1,dumstr2);
end; {loop}</pre> | | takteresteresteresteresteresteresterestere | |
| <pre>r+' + dumstr2;
0]:4:1, dumstr2;
end; [loop]
r+' + dumstr2;
r+' * dumstr2;</pre> | <pre>r+ '+dumatr2;
U:4:1,dumstr2;
r+ '+dumstr2;
power[point]:=read_pain
gain[point]:=read_gain</pre> | <pre>r+ '+dumatr2;
ul:4:1,dumatr2;
r+ '+dumatr2);
powr[point]:=read_heal
ul:4:1,dumatr2);
gain[point]:=read_gain
point[point]:=read_gain</pre> | <pre>r+ '+dumatr2;
ul:4:1,dumatr2;
r+' '+dumatr2;
power[point]:=read_peal
ul:4:1,dumatr2);
gain[point]:=read_gain</pre>
 | <pre>r+ '+dumatr2;
J:4:1,dumstr2);
c+' '+dumstr2);
power[point]:=read_pai
gain[point]:=read_gain</pre>
 | <pre>r+ '+dumatr2;
nV:=nV+read_nV(nVmetr
cj:4:1,dumatr2);
power[point]:=read_hea'
r1:4:1.dumatr2);
qain[point]:=read_qain</pre> | <pre>:+ ' + dumatr2;
[]:4:1, dumatr2;
c]:4:1, dumatr2);
power [point]:=read_hea
r+' '+dumatr2;</pre> | <pre>r+ '+dumatr2;
n):=nV+read_nV(nVmet)
t]:4:1,dumatr2);
r+' '+dumatr2;
power[point]:=read hea</pre>
 | <pre>nV:=nV*read_nV(nVmet) nV:net(i+:1,dumstr2); end; [loop] conter(netat):conter(base)</pre> | r+' '+dumstr2;
x]:4:1,dumstr2);
end; [loop]
 | :+' '+dumatr2; nV:nV+read nV(nVmet)
t]:4:1,dumstr2); end; {loop} | | ************************************* |
 |
power (point):=read hea power (point):=read hea power (point):=read hea	power[point]:=read_hea power[point]:=read_hea pdain[point]:=read_gdain	<pre>power [point]:=read hea power [point]:=read hea power [point]:=read gain dex]:4:1, dumatr2); </pre>	<pre>power [point] :=read hea power [point] :=read hea power [point] :=read hea power [2] : 4:1, dumatr2); power [point] :=read gain power [2] : 4:1, dumatr2); power [2] : 4:1, dumatr2]; power [2] : 4:1, dumatr</pre>	<pre>power[point]:=read_hea power[point]:=read_hea power[point]:=read_hea power[point]:=read_gein point]:=read_gein point]:=read_gein point]:=read_gein point]:=read_gein point];=read_gein poin</pre>	<pre>carry:''dumerr2; carry:''dumerr2; dain[point]:=read_dain dain[point]:=read_dain</pre>	power[point]:=read hea	power [point]:=read hea				adavi):3:0 dimestr2):		
and a factor of the second s	gain[point]:=read_gain(gain(point):=read_gain(gain(point):=read_gain(<pre>ndex]:4:1, dumatr2);</pre>	dain[point]:=read dain(ltstr+' '+dumstr2; power[point]:=read_heat	bower(point):=read heat	<pre>volt:=volt+read_volt(</pre>	<pre>index1:3:0,dumstr2);</pre>	<pre>currend currend c</pre>
941111 [V-111-1-1-441]; 9441111						gain[point]:=read_gain[rc	cainfoint is read can (To can the car and			TIBELT . TOURNELL ! TOURNELL ! TOURNELL !	<pre>index]:3:0,dumstr2); volt:=volt+read_volt(Vm tstr+' '+dumstr2; nV:=nV:read_nV(nVmster. end; {loop} end; {loop} control(1-nread); votar(root):=</pre>	<pre>index]:3:0,dumstr2); volt:=volt+read_volt(Vm tatr+' '+dumatr2; nV:=nV+read_nV(nVmeter. ndex]:4:1,dumatr2); end; [loop] voint[roop]</pre>	<pre>index]::0.dumstr2); index]:3:0.dumstr2); tetr+' '+dumstr2; nv::=nV+read_nV(nVmeter: end; [loop] end; [loop] voumer1:************************************</pre>
	utstrt''' toumstr <i>t!</i> index]:4:1,dumstr2); gain[point]:=read_gain[Tcon	usstr+``+aumatiz/ gain[point]:=read_gain[Tcont index]:4:1,dumatr2);	ustr+ + + dumst.z.) gain[point] = - reau_naure index] = 4 :1, dumst.2); 										
 | utstrt'''toumstrij
index]:4:1,dumstr2);
gain[point]:=read_gain[Toont
 | utstrt''' toumett?/
inderiteit.chmmetr?):
ainderiteit.chmmetr?): | | DITY. +40086171
 | | sates [setat]
 | | <pre>[index]:3:0, dumstr2); volt:=volt+read_volt (Vmet ntstr+'</pre> | <pre>[index]:3:0,dumatr2); volt:read_volt(Vmeter.de</pre> | <pre>currised_current.ta tidex1:3:0,dumstr2); volt:=volttred_volt(Vmete volt:=volttred_volt(Vmete volt:=volttred_volt(Nmeter.dev nv:=nV+read_nV(nVmeter.dev end; {loop}</pre>
 |
| <pre>power(point):=read_heater_
rear-read_heater_
rear-read_reader_
reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_
reader_
reader_
reader_
reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_reader_read</pre> | <pre>utetr+` '+dumstr2;
utetr+C '+dumstr2;
gain[point]:=read_gain[Tcon</pre> | utatr+* * +dumstr2;
jaii[point]:=read_heater_po
jain[point]:=read_pain[foint]
 | utstr+*
*+dumstr2;
joint]:=read_heater_po
joint]=read_gain(foint]=read_gain(foint]=read_gain(foint]=read_gain(foint]=read_gain(foint]=read_gain(foint]=read_gain(foint]=read_gain(foint)=read_gai | utstr+*
 | utstr+" * * * * * dumstr2;
index1: 4:1 | utatr+" "+dumstr2;
 | utstr+' '+dumstr2; | and the second |
 | | <pre>(index):3:0,dumstr2); volt:=volt+read_volt(Vmet utatr+'</pre> | <pre>([index]:3:0,dumstr2); volt:=volt=read_volt(Vmeter.de</pre>
 | <pre>curries_c</pre> |
| power (point):=read_heater_
 | <pre>utstr+ '+dumstr2; gain[point]:=read_gain[Tcon gain[point]:=read_gain[Tcon</pre> | <pre>utstr+' '+dumstr2;
gain[point]:=read_gain[Tcont
index]:4:1, dumstr2);</pre> | <pre>utstr+' '+dumstr2;
gain[point]:=read_gain[Tcont
index]:4:1, dumstr2);</pre> | <pre>outstr+ ' +dumstr2;</pre> | outstr+' '+dumstr2;
outstr+' '+dumstr2;
findex1:4:1.dnmstr2); | power[point]:=read_heater_point]:=read_heater_point]:=read_heater_point] | outstr+' '+dumstr2; | and the second | | | <pre>r[index]:3:0,dumstr2); volt:=volt+read_volt(Vmet</pre> | volt:=volt+read_volt(Vmeter.de
nV:=nV+read_nV(nVmeter.de | r[inder]:3:0,dumstr2);
volt:=volt+read_volt(Vmete
nV:=nV+read_volt(Nmeter.dev |
| power [power [power]
power [power]
utatt+ " the demarts" | uterst, ''the point is read heater power (point): sread heater power (point): sread fain (Tcon index): 4:1, dumstr2); | <pre>uttatt+' '+dumetr2;
utatt+' '+dumetr2;
jain[point]:=read_gain[Tcon]
gain[point]:=read_gain[Tcon]</pre> | <pre>utstr+' '+dumstr2';
utstr+' '+dumstr2';
gain[point]:=read_gain[Tcon]
index]:4:1, dumstr2);</pre>
 | outstr+' '+dumstr2;
outstr+' '+dumstr2;
s[index]:4:1,dumstr2);
 | <pre></pre> | power (point) := read heater p |
ratestrates | |
 | | <pre>volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmet nV:=nV+read_volt(Vmeter.de</pre> | r[index]:3:0,dumstr2);
outstr+' '+dumstr2; | <pre>curriedurieduriedurieduriedurieduriedurie</pre>
 |
| index]i4:1.dumstr2};
power[power[powth]:=read_heater
utatr+*********************************** | undex]:4:1,dumstr2);
utstr+* *tdumstr2;
gain[point]:=read_heater_p
gain[point]:=read_gain[Tcon | indexj:4:1,dumstr2);
utstr+* * +dumstr2;
idexj:4:1,dumstr2);
idexj:4:1,dumstr2); | <pre>index1:4:1,dumstr2); utstr+' '+dumstr2; utstr+' '+dumstr2; index1:4:1,dumstr2); index1:4:1,dumstr2);</pre>
 | [index]:4:1.dumstr2);
outstr+***dumstr2;
[index]:4:1.dumstr2);
 | [106ex];4:1.dumstr2);
power[point]:=read_heater_p
foutst:*f:dumstr2);
califocint]:=read_dain(Tcon | [index]:4:1.dumstr2];
power[point]:=read_heater_p | [index]:4:1.dumstr2);
outstr+' '+dumstr2:
 | (Lotex):4:1,000str2);
source(index): | [10dex]:4:1,dumstr2];
 | end; !:1; dumstr2);
end; !uoupr | <pre>volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmet v_+cead_volt(Vmet v_+cead_volt(Vmeter.de voltetr+' '+dumetr2;</pre> | <pre>x [index]:3:0, dumatr2);
outstr+' ' + dumatr2;
outstr+' ' + dumatr2;</pre> | urr=curr+read_current(
outstr+":3:0,dumetr2};
volt:=volt+read_volt(Vmet
nV:=nV+read_nV(nVmeter.de
 |
index]:4:1,dumstr2); power[point]:=read heater unter+* **dumstr2;	<pre>index1:4:1,dumstr2);</pre>	<pre>index]:4:1,dumstr2); end; [loop] utstr+' '+dumstr2; iutstr+' '+dumstr2; index]:4:1,dumstr2); index]:4:1,dumstr2); index[====================================</pre>	<pre>index1:4:1,dumstr2); end; [loop] utstr+' '+dumstr2; power[point]:=read_heater_p index1:4:1,dumstr2); index1:4:1,dumstr2]; index1:4:1,dumstr2];</pre>	<pre>[Index]:4:1,dumstr2); outstr+ ' +dumstr2; i[index]:4:1,dumstr2; gain[point]:=read_heater_p gain[point]:=read_gain[Tcon</pre>	<pre>[Index]:4:1,dumstr2); outstr+' '+dumstr2; foint]:=read_heater_p outstr+' '+dumstr2; findex1:4:1.dumstr2); findex1:4:1.dumstr2);</pre>	<pre>[index]:4:1,dumstr2); cutstr+' '+dumstr2; power[point]:=read_heater_p</pre>	<pre>[Index]:4:1.dumstr2); outstr+' '+dumstr2; power[point]:=read heater p</pre>	end; {loop} 	[index]:4:1, dumetr2); end; [loop]	[index]:4:1, dumstr2); end; [loop]	volt:=volt+read_volt(Vmet v:++++* * ******************************	volt:=volt+read_volt(Vmet nV:=nV+read_nV(nVmeter de nV:=nV+read_nV(nVmeter de	curriscurread_curread_current() vrient-10:0;4umetr2); nvient-10:0;4tread_volt(Vmet
<pre>end; (loop) end; (loop) power(point):=read_heater contentry fidumetr2; contentry fidumet</pre>	<pre>index1:4:1,dumstr2); end; [loop]</pre>	<pre>index]:4:1,dumstr2); end; [loop] end; [loop] intetr+' '+dumstr2; power[point]:=read_heater_j index]:4:1,dumstr2); index]:4:1,dumstr2]; index]:4:</pre>	<pre>index1:4:1, dumstr2); end; [loop] utstr+' '+dumstr2; power[point]:=read_heater_j index1:4:1, dumstr2); gain[point]=read_gain[Tcoint]</pre>	<pre>[index]:4:1,dumstr2); outstr+` '+dumstr2; findex]:4:1,dumstr2; findex]:4:1,dumstr2);</pre>	<pre>end; (loop)</pre>	<pre>[index]:4:1.dumstr2); power[point]:=read_heater_] </pre>	<pre>end; {loop} end; {loop} cutstr+' '+dumstr2; power[point]:=read heater ; }</pre>	end; {loop} 	[index]:4:1,dumstr2); end; {loop}	[index]:4:1, dumstr2); end; [loop]	volt:=volttrad_volt(Vmei	volt:=volt+read_volt(Wmetr2);	currited Curried Current (Wei voide 1:3:0, dumetr2); voit:=Voit+read_voit (Wei voit+read_voit (Wei voit+read_voit (Wei voit+read_voit voit+re
<pre>interior connects; index]:4:1,dumstr2); power[point]:=read_heater utstr+ '+dumstr2; power[point]:=read_heater utstr+ '+dumstr2;</pre>	uter: tidex]:4:1.dumstr2); power[point]:=read_heater_point]:=read_heater_point]:=read_heater_point]:4:1,dumstr2); point]:4:1,dumstr2);	<pre>index1:e1.dumstr2); index1:e1.dumstr2); power[point]:=read_heater_f index1:e1.dumstr2); index1:e1.dumstr2); point[point]:=read_gain(Tcor_gain[Tcor_gain[point]:=read_gain[Tcor_gain[Tcor_gain[point]:=read_gain[Tcor_gain[Tcor_gain[point]:=read_gain[Tcor_gain[Tcor_gain[point]:=read_gain[Tcor_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[point]:=read_gain[Tcor_gain[Tco</pre>	<pre>under: dumstr2); index]:4:1,dumstr2); power[point]:=read_heater_j index]:4:1,dumstr2); gain[point]:=read_gain[Tcor_ gain[point]:=read_gain[Tcor_ dumstr2);</pre>	<pre>outst::::</pre>	<pre>cutative remains { cutative remains { cutatitative remains { cutative remains</pre>	<pre>curation contact contact</pre>	<pre>(uddelt tunnett) end; loop} utstrt*' tunnett2); power[point]=read heater ; </pre>	<pre>dust: vummers; end; [loop] filoop] end; houst2);</pre>	(index):4:1, dumstr2);	[index]:4:1, dumetr2); end; {loop}	<pre>br[index]:3:0, dumstr2); volt+read_volt(Vmet volt+read_volt(Vmet)</pre>	r[index]:3:0,dumstr2);	curriscurried current (vertex); volt:=volt+read_volt (Vertex);
utstr+' + toumstr2} index]:4:1,dumstr2); power[aoint]:=read_heater utst+' 'tdumstr2';	utsert***********************************	<pre>utstr+ + tomstr2; index]:4:1, dumstr2); utst++ ' + dumstr2; index]:4:1, dumstr2; index]:4:1, dumstr2); gain[point]:=read_gain[Tcor gain[[tcor gain[[tcor]]:=read_gain[[tcor gain[[tcor]]:=read_gain[[tcor gain[[tcor]]:=read_</pre>	<pre>utstr+ + toumstr2; index]:4:1, dumstr2); utstr+' ' +dumstr2; index]:4:1, dumstr2; index]:4:1, dumstr2); gain[point]:=read_pain[Tcor</pre>	<pre></pre>	outsert ' toumstri) [[idex]:4:1,dumstr2]; pows[point]:=read_heater_] ['idex]:4:1.dumstr2); qain[[point]:=read_qain[Tor]	uterret found r23 (loop) [index]:4:1,dumstr2); outstr+' 'tdumstr2;	outstrt' 'toumstrz' outstrt' 'toumstrz'; outstrt' 'dumstr2'; power[point:=read heater ;	<pre>until the second of the second s</pre>	Intervention outsite formaticity admattch admat	uri-urited in the second of the second of the second of the second se	volt+read_volt(Vmet volt+read_volt(Vmet	r[index]:3:0,dumstr2); Volt:=volt+read_volt(Vmet	currand current (current curr
<pre>utstr+' '+dumstr2; index]:4:1,dumstr2); end; [loop] power[power[power[power]end:1]:rread_heater utstr+' '+dumstr2;</pre>	<pre>utetr+` '+dumetr2; index]:4:1,dumetr2); utetr+` '+dumetr2; utetr+` '+dumetr2; index]:4:1,dumetr2); gain[point]:=read_gain[Tcc</pre>	<pre>utstr+' ' +dumstr2; index]:4:1,dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2; index]:4:1,dumstr2); index1:4:1,dumstr2); index1:4:1,dumstr2);</pre>	<pre>utstr+' '+dumstr2; utstr+' '+dumstr2; index]:4:1,dumstr2; utstr+' '+dumstr2; power[point]:=read_heater index]:4:1,dumstr2; index]:4:1,dumstr2;</pre>										
 | <pre>-outstrt' '+dumstr2;
n[index]:4:1,dumstr2);
-outstrt' '+dumstr2);
power[point]:=read_pater_
gain[point]:=read_gain[Tcc</pre>
 | <pre>outstr+' '+dumstr2;
[index]:4:1,dumstr2);
power[point]:=read_heater_
[index1:4:1,dumstr2);
power[point]:=read_deater_</pre> | <pre>nv:=nv+read_nv(nvmeter.c nv:=nv+read_nv(nvmeter.c end; [loop] end; [loop] outstr+' '+dumstr2; </pre> | nV:=nV+read_nV(nVmeter.c
end; {loop}
outstr+' '+dumstr2;
 | uutstr+' '+dumstr2;
end; [loop]
Licter:!st:]. | <pre>nV:=nV+read_nV(nVmeter.c nU:=nV+read_nV(nVmeter.c end; [loop]</pre>
 | nV:=nV+read_nV(nVmeter.c
[index]:4:1,dumstr2);
end; {loop} | volt:=volt+read_volt(Vme | r[indax]:3:0.dumatr2]:
volt:read | current current of the second state of the second
 |
| <pre>utstr+' '+dumstr2;
index]:4:1,dumstr2);
power[point]:=read heater
'' '+dumstr2;</pre> | <pre>utstr+' '+dumstr2;
index]:4:1,dumstr2);
utstr+' '+dumstr2);
power[point]:=read_heater_
index]:4:1,dumstr2);
gain[fcc</pre> | <pre>utstr+' '+dumstr2;
index]:4:1,dumstr2);
utstr+' '+dumstr2);
utstr+' '+dumstr2;
index]:4:1,dumstr2);
index]:4:1,dumstr2);
index]:4:1,dumstr2);</pre> | <pre>utstr+' '+dumstr2;
index]:4:1,dumstr2);
utstr+' '+dumstr2);
utstr+' '+dumstr2);
gain[point]:=read_gain[Tcc
index]:4:1,dumstr2);
index]:4:1,dumstr2);</pre>
 | <pre>-outstr+' '+dumstr2;
n[index]:4:1,dumstr2);
-outstr+' '+dumstr2;
s[index]:4:1,dumstr2;
gain[point]:=read_pain[Tco
gain[point]:=read_gain[Tco</pre>
 | outstr+' '+dumst2;
[index]:4:1,dumst2);
outstr+' '+dumst2;
findex1:4:1.dumst2;
outstr+' '+dumst2;
findex1:4:1.dumst2);
gain[point]:=read dain[Tco | <pre>nV:=nV+read_nV(nVmeter.d</pre> | <pre>outstr+' '+dumstr2;
nV:=nV+read_nV(nVmeter.d
end; {loop}
outstr+' '+dumstr2;
power[point]:=read heater</pre>
 | nV:=nV+read_nV(nVmeter.d
[index]:4:1,dumstr2);
 | nV:=nV+read_nV(nVmeter.d_
[index]:4:1,dumstr2);
end; [loop]
 | nV:=nV+read_nV(nVmeter.d_
[index]:4:1,dumstr2); end; {loop} | - 11-1 | and) for heartform for |
 |
| <pre>utetr+' '+dmmstr2;
utetr+' '+dmmstr2;
index]:4:1,dumstr2);
end; [loop]
utetr+' '+dumstr2;
utetr+' '+dumstr2;
utetr+' '+dumstr2;</pre> | <pre>nV:=nV+read_nV(nVmeter.d
utstr+' '+dumstr2';
utstr+' '+dumstr2';
power[point]:=read_heater
index]:4:1,dumstr2);
gain[point]:=read_gain[Tcc</pre> | <pre>utar+' '+dumatr2;
utar+' '+dumatr2;
utatr+' '+dumatr2;
utatr+' '+dumatr2;
janex]:4:1,dumatr2;
janex]:4:1,dumatr2);
gan[foint]:=read_gan[foint]:=read_gan[foint]:=read_gan[foint]:=read_gan[foint];</pre> | <pre>utetr+' '+dumetr2;
utetr+' '+dumetr2);
andex]:4:1,dumetr2);
utetr+' '+dumetr2;
jouer[point]:=read_pain[rot]:=read_gain[rot];=read_gain[rot];=read_gain[rot];</pre>
 | <pre>nv:=nv+read_nv(nvmeter.c
outstr+' '+dumatr2;
n[index]:4:1,dumstr2);
power[point]:=read_heater
gain[point]:=read_gain[Tcc</pre>
 | nV:=nV+read_nV(nVmeter
outstr+' '+dumstr2;
end; [loop}
outstr+' '+dumstr2;
power[point]:=read_heater_
findex1:+1.thumstr2);
gain[point]:=read_heater_
dain[point]:=read_dain[To: | <pre>nv:=nv+read_nv(nvmeter.c) nv:=nv+read_nv(nvmeter.c) end; [loop] end; [loop] power[point]:=read_heater.c</pre> | <pre>nV:=nV+read_nV(nVmeter.c)</pre>
 | nV:=nV+read_nV(nVmeter.o
outstr+ '+dumetr2;
[index]:4:1,dumetr2);
end; [loop]
outstr2); | nV:=nV+read nV(nVmeter.
[index]:4:1,dumstr2);
end; [loop]
 | <pre>nV:=nV+read nV(nVmeter.c)</pre> | | |
 |
| <pre>nV:=nV+read_nV(nVmeter.
index]:4:1,dumstr2;
end; {loop}
power[point]:=read_heater
utstr+ '+dumstr2;</pre> | <pre>nV:=nV:read_nV(nVmeter.
nV:=nV:read_nV(nVmeter.
index]:4:1,dumstr2);
power[point]:=read_gain(Tc
index]:4:1,dumstr2);</pre> | <pre>uter+' '+dumetr2;
uter+' '+dumetr2;
index]:4:1,dumetr2;
utetr+' '+dumetr2;
index]:4:1,dumetr2;
jain[point]:=read_gain[point]:=read_gain[point]:=read_gain[point];</pre> | <pre>utetr+' '+dumstr2;
utetr+' '+dumstr2;
index]:4:1,dumstr2;
utetr+' '+dumstr2;
joint]:=read_gain[c
index]:4:1,dumstr2;
joint]:=read_gain[c</pre>
 | <pre>nV:=nV:read_nV(nVmeter.
outstr+' '+dumatr2;
n[index]:4:1,dumstr2);
eutstr+' '+dumstr2;
gain[point]:=read_gain[Tc</pre>
 | <pre>cutate+' ' dumatr2'; nV=nV+read_nV(nVmeter. i(undex):4:1, dumatr2); cutate+' ' +dumatr2); power[point]:=read_heater iindex1:4:1, dumatr2); gain[point]:=read_heater iindex1:4:1, dumatr2); gain[point]:=read_heater</pre> | outstr+' '+dumstr2'
end; [loop}
end: [loop}
power[point]:=read_heater | <pre>outstr+' 'dumstr2;
outstr+' 'dumstr2';
{index]:4:1,dumstr2';
outstr+' 'dumstr2';
outstr+' 'dumstr2'</pre>
 | N:=N+read_NV(NVmeter.
outstr+' '+dumstr2;
end; [loop]
curer!:-/orl:-/: | nV:=nV+read_nV(nVmeter
outstr+' '+dumstr2;
end; [loop]
 | utstr'' '+dumatr2;
[index]:4:1,dumstr2;
end; [loop} | | |
 |
| <pre>utstr+' ' +dumstr2;
utstr+' ' +dumstr2;
index]:4:1,dumstr2;
power[point]:=read_heater_
utstr+' ' +dumstr2;</pre> | <pre>utatres</pre> | <pre>utatr+' '+dmmstr2;
utatr+' '+dmmstr2;
index]:4:1,dmmstr2;
utatr+' '+dumstr2;
index]:4:1,dumstr2;
index]:4:1,dumstr2;
index]:4:1,dumstr2;</pre> | <pre>utities://dometr2/
utities// + dometr2/
index]:4:1, dometr2);
utities (point]:=read_nV(nVmeter.d
index]:4:1, dometr2);
index]:4:1, dometr2);
gain[point]:=read_gain[[coint]:=read_gain[[coint]]:=read_gain[[coint]];</pre> | <pre>sultates://outseter.com/subjecter.com/s</pre> | <pre>cutatr+' '+dumatr2;
outatr+' '+dumatr2;
[index]:4:1,dumatr2);
power[point]:=read_heater_
findex1:4:1,dumatr2);
gain[point]:=read_dain[To-</pre> | outstr+" '-dumetr2;
Nutstr+" '-dumetr2;
end; [loop]
power[puint]:=read_heater_
outstr+' '-dumetr2; | <pre>//interval_provements/
outstr+* '.tdumatr2;
end; [loop]
outstr+* '.tdumatr2;
power[point]:=read heater
outstr+* '.tdumatr2;</pre> | <pre>//inversion of the second second</pre> | <pre>cutative / dumatic/
nv:=nv+read nv(nvmeter.d
end; {loop}</pre> | ultarres | | | |
| <pre>[index]:3:0.dumatr2}; utatr+' '+dumatr2; index]:4:1.dumatr2; end; [loop] floop] power[point]:rread_heater</pre> | <pre>[index1:3:0,dumerr2);</pre> | <pre>[index]:s:0,dumstr2/; volc:=Volc+read_volc(vm volc:=Volc+read_volc(vm volc:=Volc+read_volc(vm volc:=Volc+read_volc(vm volc=Volc+read_volc(vm volc+read_volc(vm volc+read_vo</pre> | <pre>[index]:::U,dumstr2/; utstr+' ' +dumstr2/; utstr+' ' +dumstr2/; index]:4:1,dumstr2); end; [loop] utstr+' ' +dumstr2); power[point]:=read_pain[Tcc index]:4:1,dumstr2); index]:4:1,dumstr2);</pre> | <pre>pr[index]:s:0,dumstrl/;
coutstr+' 'fdumstr2;
outstr+' 'fdumstr2);
power[point]:=read_nv(nvmstr2);
power[point]:=read_pain(Tcc
gain[point]:=read_gain(Tcc</pre> | r[index]:3:0,dumatr2};
outatr+' '+dumatr2;
[index]:4:1,dumatr2);
power[point]:=read_heater_
[index]:4:1,dumatr2);
gain[point]:=read_gain[Tcc | r[index]:3:0,dumstr2};
outstr+' '+dumstr2;
[index]:4:1,dumstr2};
outstr+' '+dumstr2};
outstr+' '+dumstr2; | r[index::3:U,dumatr2};
outatr+' '+dumatr2;
outatr+' '+dumatr2;
outatr+' '+dumatr2};
outatr+' '+dumatr2; | r[index]:3:0,dumetr2);
outstr+' ' +dumetr2;
and: [loop]
_inder]:4:1,dumetr2);
_inder]:4:1,dumetr2); | <pre>r[index]:3:0,dumetr2}; outstr+' '+dumetr2; f[index]:41./dumstr2; end; [loop]</pre> | <pre>r[index]:s:0,dumetr2); nv:=nvtread_nv(nvmeter.d nv:=nvtread_nv(nvmeter.d index]:4:1,dumetr2); end; [loop]</pre> | | | |
| <pre>(index):3:0,dumatr2); volt=volt+read_volt(Vm vutetr+' + dumatr2; index]:4:1,dumatr2; end; [loop] findex1:4:1,dumatr2; power[point]:=read_heater vutetr+' + dumatr2; </pre> | <pre>(index):3:0,dumetr2); voit:=voit+read_voit(net vietr+' ' +dumetr2; vietr+' ' +dumetr2); end; [loop] = read_pater.de index]:4:1,dumetr2); gain[point]:=read_pain[Tcon index]:4:1,dumetr2);</pre> | <pre>(index):3:0,dumstr2); volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmet utatr+' ' +dumstr2; index]:4:1,dumstr2); end; [loop] utstr+' '+dumstr2); power[point]:=read_heater_p index]:4:1,dumstr2); index]:4:1,dumstr2);</pre> | <pre>[index]:3:0,dumstr2); volt:=volt+read_volt(Vmet utatr+' ' +dumstr2; inV:=nV+tead_vv(nVmeter.de nV:=nV+tead_nV(nVmeter.de index]:4:1,dumstr2; utstr+' '+dumstr2; index]:4:1,dumstr2; index]:4:1,dumstr2;</pre>
 | <pre>r[index]:3.0,dumstr2); volt:=volt+read_volt(n%met outstr+' ' +dumstr2; outstr+' ' ' +dumstr2; end; [loop] = end; [loop] = end; [loop] = read_gain[Toon tstr+' ' +dumstr2); elindex]:4:1,dumstr2); elindex]:4:1,dumstr2); </pre>
 | <pre>r[index]:3:0,dumstr2); voit:read_voit(met outstr+' '+dumstr2; [index]:4:1,dumstr2); outstr+' '+dumstr2); power[point]:read_hater_p outstr+' +dumstr2); point]:read_hater_p dain[point]:read_hater_p </pre> | <pre>r[index]:3:0,dumstr2); outstr+' '+dumstr2; [index]:41,dumstr2; cutstr+' '+dumstr2; cutstr+' '+dumstr2); cutstr+' '+dumstr2); cutstr+' '+dumstr2;</pre> | r[index]:3:0.dumetr2);
outstr+' '+dumetr2;
outstr+' '+dumetr2);
outstr+' *dumetr2);
power[point]:=read heater p
 | <pre>t[index]:3:0,dumetr2);</pre> | <pre>r[index]:3:0,dumstr2);</pre>
 | <pre>r[index]:3:0,dumetr2); volt:=volt+read_volt(Vmet vutstr+' '+dumetr2; iutstr+' '+dumetr2; end; [loop] end; [loop]</pre> | | |
 |
<pre>(index):3:0,dumatr2); volt:=volt+read_volt(Vme utatr+' '+dumatr2; index]:4:1,dumatr2; end; [loop] fower[power</pre>	<pre>(index]:3:0,dumstr2); volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmet vistr+` * dumstr2; end; [loop]</pre>	<pre>(index1:3:0,dumstr2); volt:=volt+read_volt(Vmet utatr+' '+dumstr2; utatr+' '+dumstr2; index1:4:1,dumstr2); power[point]:=read_heater_p utatr+' '+dumstr2); index1:4:1,dumstr2); index1:4:1,dumstr2);</pre>	<pre>[index]:3:0,dumstr2); volt:=volt+read_volt(Vmet utstr+^</pre>	<pre>br[index]:3:0,dumstr2); volt:=volt+read_volt(Vmet outstr+' '+dumstr2; nY:=nY+read_nV(nVmeter.de outstr+' '+dumstr2); power[point]:=read_heater_p entstr+' '+dumstr2); s[index]:4:1,dumstr2); </pre>	<pre>r[index]:3:0,dumstr2); volt:=volt+read_volt(Vmet outstr+' '+dumstr2; (index]:4:1,dumstr2); end; [loop] powr[point]:=read_heater_p index1:4:1,dumstr2); point[:=read_heater_p index1:4:1,dumstr2);</pre>	<pre>volt:=volt+read_volt(Vmet outet+' '+dumetr2; [index]:4:1,dumstr2); outett+' '+dumetr2); power[point]:=read_heater_p</pre>	<pre>r[index]:3:0,dumstr2); volt+read_volt(Vmet outstr+' '+dumstr2; findex]:4:1,dumstr2); end; [loop] outstr+' '+dumstr2); power[point]:=read heater p</pre>	<pre>r[index]:3:0,dumetr2); volt:=volt+read_volt(Vmet vutstr*' * +dumetr2; nV:=nV+read_volt(Nmeter.de nV:=nV+read_volt(Nmeter.de</pre>	<pre>r[index]:3:0,dumstr2);</pre>	<pre>volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmet vutstr+' '+dumstr2;</pre>			
<pre>####################################</pre>	<pre>refination for the second second s</pre>	<pre>Attract and a current[Im current]);</pre>	<pre>curr==curr+read_current[Im [index]:3:0.dumetr2); utatr+` ' dumetr2); utatr+` ' dumetr2); utatr+` ' dumetr2); utatr+` ' dumetr2); index]:4:1,dumetr2); power[point]:=read_heater_po gain[point]:=read_gain[cont]; index]:4:1,dumetr2);</pre>	<pre>r***********************************</pre>	<pre>r***********************************</pre>	<pre>trifits::::0.dumstr2); t[index]:3:0.dumstr2); volt:=volt+read_volt(Vmete volt:=volt+read_volt(Vmete voltstr4' '+dumstr2); f[index]:4:1.dumstr2); outstr4' '+dumstr2); volt::=volt+read_volt(Vmeter.dev voltstr4' '+dumstr2); voltstr4' '+dumstr2); voltstr4' '+dumstr2);</pre>	<pre>refinition for the second se</pre>	<pre>intitition::::::::::::::::::::::::::::::</pre>	<pre>r***********************************</pre>	<pre>inititititititititititititititititititi</pre>			
<pre>Antitation antitation and antitation antitation antitation antitation antitation and antitation antitation and antitation antitation and antitation and antitation and antitation antitation antitation antitation antitation and antitation antitation and antitation antitat</pre>	<pre>####################################</pre>	<pre>####################################</pre>	<pre>####################################</pre>	<pre>ssssssssssssssssssssssssssssssssssss</pre>	<pre>assessassessassessassessassessassessassessassessassessassessassessad r[index]:3:0, dumatr2); outatr+ ' +dumatr2); [index]:4:1, dumatr2); end; [loop] foutatr+ ' +dumatr2); power[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc dain[point]:=read_heater_pc</pre>	<pre>annitation curries (In curries (In curries (In curries (In curries (In curries (In volt:=curries (In volt:=volt:=volt:=volt:=volt:=volt:=volt:=volt:=volt:=volt:=volt(Vmete volt) (Vmete volt) (Indet (In the volt) (Indet vo</pre>	<pre>assistations assistation assistation assistation assistation assistation assistation assistation assistation as a curributed current (Imber as a curributed as a curributed as a curributed as a current as a curributed as a current as a</pre>	<pre>Antitation antitation and antitation antitation antitation antitation and antitati</pre>	<pre>Antitationsinterinterinterinterinterinterinterinter</pre>	<pre>initiality curr:=curr:read_current(Im c[index]::0,dumstr2); outstr+' '+dumstr2; [index]:4:1,dumstr2); end; {loop}</pre>			
<pre>interimentations is a current interimentation is a current is a current interiment is a current interiment is a current index]:3:0,dumstr2; inter+* * +dumstr2; index]:4:1,dumstr2; index]:4:1,dumstr2; inter+* * +dumstr2; i</pre>	<pre>international international internation internati</pre>	<pre>(index]:3:0, dumstr2; utstr+' ' +dumstr2; index]:4:1, dumstr2; index]:4:1, dumstr2; utstr+' ' +dumstr2; index]:4:1, dumstr2; utstr+' ' +dumstr2; index]:4:1, dumstr2; index]:4:1, dumstr2; index]:4:1, dumstr2;</pre>	<pre>index]:d.l.dumstr2; index]:d.l.dumstr2; i</pre>	<pre>riting in the second seco</pre>	<pre>runter::::::::::::::::::::::::::::::::::::</pre>	<pre>refine the second second</pre>	<pre>ritition in the second contraction in the second contraction is a second contract (im volt: "volt+read_volt(Vmeter) volt: "volt+read_volt(Vmeter) volt: "volt+read_volt(Vmeter) volt: "volt+read_volt(Nmeter) volt: "volt+read_volt(Nmeter) volt: "volt+read_volt(Nmeter) volt+read_volt(Nmeter) volt+read_volt+read_volt(Nmeter) volt+read_volt(Nmeter) volt+re</pre>	<pre>curr:=curr:read_current(Im read_read_current(Im volt:=volt:=volt+read_volt(Vmete: nV:=nV+read_nV(nVmeter.dev) inv:=nV+read_nV(nVmeter.dev) end; [lop]</pre>	<pre>curr:=curr:read_current(Im reintex): outstr+' '+dumstr2; [index]:4:1,dumstr2; end; [loop]</pre>	<pre>initial curvest (ine initial curvest) currection currection voit:=voit+read_current(ine voit:=voit+read_voit(vmeter.devi utetr+' ' +dumetr2); nV:=nV+read_nV(nVmeter.devi [index]:4:1,dumetr2); end; [loop]</pre>			
<pre>utatr+' 'dometr2' utatr+' 'dometr2'; utatr+' '+dometr2'; utatr+' '+dometr2'; utatr+' '+dometr2'; utatr+' '+dometr2'; utatr+' '+dometr2'; utatr+' '+dometr2'; utatr+' '+dometr2'; utatr+' '+dometr2'; utatr+' '+dometr2';</pre>	<pre>utstr+` + dumstr2; ************************************</pre>	<pre>utstr+ * + dumstr2; ************************************</pre>	<pre>utstr+ + 4dumstr2; utstr+ * + 4dumstr2; utstr+ * + 4dumstr2; utstr+ * + 4dumstr2; utstr+ * + dumstr2; utstr+ * + dumstr2; index]:4:1, dumstr2; utstr+ * + dumstr2; index]:4:1, dumstr2; power[point]:=read_pain(point]:=read_ga</pre>	<pre>uutstr+' + dumstr2: initiatiatiatiatiatiatiatiatiatiatiatiatia</pre>	<pre>uutstr+' + dumstr2; assassassassassassassassassassassassass</pre>	<pre>uutstr+' + dumstr2; uutstr+' * dumstr2; [index]:30, dumstr2; uutstr+' * dumstr2; [index]:4:1, dumstr2; uutstr+' * dumstr2; uutstr+' * dumstr2; uutstr+' * dumstr2;</pre>	<pre>DUTSIT* ' + OUMBELZ! Lumitanianianianianianianianianianianianiania</pre>	<pre>uutstrt' 'dumstr2; neessessessessessessessessessessessesses</pre>	uutstr+' * 400mmstr2; t[indem]:3:0,dummstr2); outstr+' * 4dummstr2); f[indem]:4:1,dummstr2); end; [loop] end; [loop]	<pre>uutstr** *dumstr2; tiatar**********************************</pre>	10012511, *40085121		rem:-remarked: remarked: remark
<pre>utetr+` '+dumstr2; ************************************</pre>	<pre>utetr+` '+dumstr2; ************************************</pre>	<pre>utatr+' '+dumatr2; ####################################</pre>	<pre>uttr+' '+dumstr2; ************************************</pre>	<pre>outstr' ' +dumstr2; tem:=temtresd_temp(Tcontro tem:=temtresd_temp(Tcontro curr:=curr+read_turrent(Im curr:=curr+read_current(Im volt:=volt+read_volt(Vmete voltsr+1 ' +dumstr2); index1:+ ' +dumstr2); power[point]:=read_heater_po (index1:+ ' +dumstr2); power[point]:=read_gain(Tcont</pre>	<pre>outstr+' ' +dumstr2; tem:=tem+read_temp(Tcontrc tem:=tem+read_temp(Tcontrc reint=scurt+read_current(In curr=scurt+read_current(In volt=scult+read_volt(Wmete volt=scult+read_volt+read_volt(Wmete volt=scult+read_volt+read_volt(Wmete volt=scult+read_volt+read_volt)=scalter_volt volt=scult+read_volt)=scalter_volt volt=scalter_volt+read_volt(Tconter_volt+read_volt)=scalter_volt volt=scalter_volt+read_volt)=scalter_volt volt=scalter_volt+read_volt(Tconter_volt+read_volt)=scalter_volt volt=scalter_volt+read_volt(Tconter_volt+read_volt)=scalter_volt volt=scalter_volt+read_volt(Tconter_volt+read_volt)=scalter_volt volt=scalter_volt+read_volt)=scalter_volt+read_volt(Tconter_volt+read_volt)=scalter_volt+read_volt(Tconter_volt+read_volt+read_volt)=scalter_volt+read_volt)=scalter_volt+read_volt(Tconter_volt+read_volt+read_volt+read_volt)=scalter_volt+read_volt</pre>	<pre>outstr' ' dumstr2; and transactions and the product of the pr</pre>	<pre>outstr+' ' +dumstr2;</pre>	<pre>Dutstr+' '+dumstr2; Anternationanternationanternationanternationanternation rearranternationanternationanternationanternation rearranternationanter2; outstr+' '+dumstr2; nV:=nV:read_nV(nVmeter.dev) end; [loop] rearranternation rearra</pre>	<pre>outstr+' '+dumstr2;</pre>	<pre>utstrf' ' + dumstr2; nutstrf' ' + dumstr2; nutstrf' ' + dumstr2; nutstrf' ' + dumstr2); nutstrf' ' + dumstr2); nvistrf' ' + dumstr2)</pre>	outstr+' '+dumstr2; 	tem:=temtread_temp(Tcontro	outstr+" '+dumstr2; temp(Tcontro
<pre>utstr+` '+dumstr2; tem:=tem+read_temp(Tcont tem:=tem+read_temp(Tcont index]:3:0,dumstr2); utstr+` '+dumstr2; index]:4:1,dumstr2; index]:4.1,dumstr2); power[point]:read_nv[nvmeter.d index]:4.1,dumstr2; index1:4.1,dumstr2; index1:4.1,dumstr2); index1:4.1,dumstr2; ind</pre>	<pre>utetr+` '+dumetr2;</pre>	<pre>utstr+' '+dumstr2; tem:=tem+read_temp(Tcontro tem:=tem+read_temp(Tcontro tindex1:=1:0,dumstr2); utstr+' '+dumstr2); index1:=4:1,dumstr2); utstr+' '+dumstr2); index1:=4:1,dumstr2); index1:=4:1,dumstr2); index1:=4:1,dumstr2); index1:=4:1,dumstr2);</pre>	<pre>utstr+` '+dumstr2; ************************************</pre>	<pre>outstr+' '+dumstr2;</pre>	<pre>outstr+' '+dumstr2; tem:=tem+read_temp(Tcontrc tem:=tem+read_temp(Tcontrc turt:=curr+read_current(Im volt=reult+read_volt+read_volt(Vmete volt=reult+read_volt(Vmete volt=reult+read_volt(Vmete volt=reult+read_volt+read_volt(Vmete volt=reult+read_volt+read_volt+read_volt(Vmete volt=reult+reult); volt=reult+reult+reult+reult+reult+read_volt+read_volt+reult+re volt=reult+reult</pre>	<pre>outstr+' '+dumstr2; tem:=tem+read_temp(fcontrc tem:=tem+read_temp(fcontrc</pre>	<pre>outstr+</pre>	<pre>Dutstr+* ' +dumstr2; ************************************</pre>	<pre>outstr+' '+dumstr2; temp(Tcontro ************************************</pre>	<pre>butstr+' '+dumstr2; tem:read_temp(Tcontrol initialized temp(Tcontrol initialized temp(Tcontrol initialized temp(Tcontrol curri-read_current(Ime volt:=rout+read_volt(Vmeter outstr+' '+dumstr2); findex]:4:1,dumstr2); end; [loop]</pre>	outstr+' '+dumstr2; tem:=tem+read_temp(Tcontr	outstr+' '+dumstr2;	tem:"tem:"tem:"tem?Tead temp[Tcontro
<pre>utetr+' +dumstr2; utetr+' +dumstr2; (index]:3:0,dumstr2;; utetr+' 'dumstr2; index]:4:1,dumstr2;; power[point]:-read_nv[nvmeter.c power[point]:-read_heater.c</pre>	<pre>utstr+' + dumstr2; itstr+' + dumstr2; iindex]:3:0,dumstr2); woit:=curr+read_current(I voit:=voit+read_nV(nVmeter de utstr+' + dumstr2); index]:4:1,dumstr2); power[point]:=read_gain(Tcon index]:4:1,dumstr2);</pre>	<pre>utstr+' + dumstr2; utstr+' + dumstr2; (index]:3:0, dumstr2); utstr+' ' + dumstr2); power [point]:=read_heater_po gain[point]:=read_heater_po gain[point]:=read_gain([cont); eread_heater_po</pre>	<pre>utetr+' '+dumstr2; utetr+' '+dumstr2; [index]:3:0,dumstr2); utetr+' '+dumstr2; utetr+' '+dumstr2; index]:4:1,dumstr2; utetr+' '+dumstr2; utetr+' '+dumstr2; utetr+' '+dumstr2; power[point]:=read_heater_po gain[point]:=read_gain[cont];</pre>	<pre>outstr+' ' +dumatr2; cutatr+' ' +dumatr2; r[index]:9:0,dumatr2; outstr+' ' +dumatr2; outstr+' ' +dumatr2; outstr+' +dumatr2; power[point]:=read_partr_po power[point]:=read_partr_po power[point]:=read_partr_po point]:=read_part(Tont</pre>	<pre>outetr+' '+dumstr2; cuttr+' '+dumstr2; r[index]:3:0,dumstr2); outetr+' '+dumstr2); outetr+' '+dumstr2); end; [loop] findex] +:1,dumstr2); power[point]:=read_heater_pc qain[point]:=read_heater_pc qain[point]:=read_heater_pc</pre>	<pre>outstr+' ' +dumstr2; assistation assistation assistence assistation assistence assistation assistence assiste</pre>	outstr+' '+dumstr2; ************************************	<pre>untstr+' ' +dumstr2; untstr+' ' +dumstr2; curr:=curr+read_current(Ime volt:=volt:=volt+read_volt(Vmete) untstr*' ' +dumstr2; nv:=nv:=ad_nv(nVmeter.dev) fidex1:4:1, dumstr2); end; [loop]</pre>	<pre>outstr+' '+dumstr2; outstr+' '+dumstr2; r[index]:3:0,dumstr2); outstr+' '+dumstr2); nV:=nV+read_nV(nVmeter.dev. [index]:4:1,dumstr2); end; [loop]</pre>	<pre>utstr* ' + dumstr2; utstr* ' + dumstr2; reastant and reastant and reastant and reastant (Ime r[index]:3:0, dumstr2); utstr* ' + dumstr2; [index]:4:1, dumstr2); end; [loop]</pre>	coutetr+' '+dumstr2; tem:=tem+read_temp(Tcontr	outetr' ' +dumstr2;	tem:=tem+read temp[Tcontro tem:=tem+read temp[Tcontro
<pre>contractory tem:=temtread_temp(Tcont utetr+' 'dumstr2; utetr+' 'dumstr2; utetr+' 'dumstr2; utetr+' 'dumstr2; index]:4:1,dumstr2; index]:4:1,dumstr2; index1:4:1,d</pre>	<pre>utstr://dumstr2/ utstr://dumstr2/ ten:reutrread_current(I volt:reolttread_volt(Vmet volt:rvolttread_volt(Vmet nV:=nV:read_nV(nVmeter.de nd; (loop) utstr+' '+dumstr2); power[point]:=read_gain(Tcon index]:4:1,dumstr2);</pre>	<pre>utatr+' 'dumatr2; utatr+' 'dumatr2; inter+************************************</pre>	<pre>test::.utstr; cumstr2; utstr::</pre>										
 | <pre>yrers.dumetr2;
cutatry: 'dumetr2;
r***********************************</pre>
 | <pre>voutetr+' +dumstr2;
vutetr+' +dumstr2;
r***********************************</pre> | <pre>y sets, cumests/;
tem:=tem+read_temp(fcontro
cutstransnamanananananananananananananan
t[index]:3:0,dumstr2);
volt:=volt+read_volt(Ymete
outstr+' '+dumstr2);
[index]:41,dumstr2);
power[point]:=read_heater_p;
power[point]:=read_heater_p;</pre> |
<pre>yrers.commercs;
outstr://idumstr2/
curried_comp(Tcontro)
r[index]:3:0,dumstr2);
wolt:=volt+read_volt(Weets
volt:=volt+read_volt(Weets
voltstr+/ 'dumstr2);
[index]:4:1,dumstr2);
outstr+' 'dumstr2);</pre> | <pre>rest.commett2;
cutatr*' +dumatt2;
restrantseries entering(Tcontrol
restrantseries entering);
restrantseries entering);
rest.commett2;
outatr*' +dumatt2;
commett2;
commett2;
commett2;
commett2;
commett2;
enter:
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
commett2;
c</pre> | <pre>/restructions
outstr:/ dumstr2;
restrainsentestrainsentestrainsentestrainsentestrain(Im
restrainsentestrainsentestrainsentestrainsentestrain)
restrainsentestrainsentestrainsentestrainsentestrain
volt=read_nv(nVmeter.dev.
[index]:4:1, dumstr2);
end; [loop]</pre>
 | <pre>/rest.ummetts;
butetr:/ +dumetts;
nationariantiantiantiantiantiantiantiantiantiant</pre> | y:s::.qumstr2;
outet+' '+dumstr2;
 | yrers.cumstral/
outstr+' '+dumstr2; | y:u:s,uumustzij
butstr+' '+dumustz2;
 |
| <pre>:si2,dumstr2);
tem:=d_temp(fcont
statts:statts:statts:statts:statts:statts:statts:statts:statts:
statts:statts:statts:statts:statts:statts:statts:statts:statts:
utstr+` '+dumstr2);
utstr+` '+dumstr2);
over[point]:=toop}
utstr+` '+dumstr2);
statts:
over[point]:=tood_heit.cont
over[point]:=tood_heit.cont
over[point]:=tood_heit.cont
over[point]:=tood_heit.cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
over[point]:=toot_cont
ov</pre> | <pre>cretion:::/dumetr2);
utetr+' + dumetr2)
tem:=tem+read_temp(Tcontr
intex::::::::::::::::::::::::::::::::::::</pre> | <pre>: ::::, dumatr2);
.:::::, dumatr2;
.::::::::::::::::::::::::::::::::::::</pre> | <pre>certimetrs);
utetr+ '+dumstr2;
utetr+' '+dumstr2;
index]:3:0,dumstr2);
utetr+' '+dumstr2);
utetr+' '+dumstr2;
index]:4:1,dumstr2;
utetr+' '+dumstr2;
index]:4:1,dumstr2;
index]:4:1,dumstr2;
index]:4:1,dumstr2;
index]:4:1,dumstr2;
index]:4:1,dumstr2;</pre>
 | <pre>rst.cumatr2;
butatr.dumatr2;
butatr.t "+dumatr2;
butatr.t "+dumatr2;
cut:=curt+read_current(Im
volt:=curt+read_volt(Ymete
volt:=volt+read_volt(Ymete
volt:r* '+dumatr2);
butatr+' '+dumatr2;
butatr+' '+dumatr2;
gain[point]:=read_gain[Tcont
gain[point]:=read_gain[Tcont</pre>
 | <pre>vist_cummstr2);
vutstr.** 'dummstr2);
r************************************</pre> | <pre>rsis.dumetr2);
utatr+' ' dumetr2;
utatranananatrainananananananananananananananan
curr=curr+read_current(In
volt=volt+read_voltVmete
utatr+' ' +dumetr2;
end; [loop]
utatr+' '+dumetr2;
uutatr+' '+dumetr2;</pre> | <pre>rst.cumstr2);
utstr+ ' +dumstr2;
intstrassistations.steat.teat_teat_feat_feat_feat_feat_feat_feat_</pre>
 | rest, dumatr2);
utetr*********************************** | res:,dumatr2);
butatr4: 'dumatr2;
intittatitatitatitatitatitatitatitatitat
 | risr | reis/dumetr2);
utstr+''+dumetr2;
tem:-tem+read_temp(Tcontr | retz:/dumatr2);
butatr+' '+dumatr2; | riti2,dumatr2);
butatr+ '+dumatr2;
 |
| <pre>i6:2, dimetr2);
tem:=remtread_temp(Tcont
tem:=remtread_temp(Tcont
tem:=remtread_temp(Tcont
tem:=remtread_temp(Tcont
volt:=volt=read_volt(Vme
volt:=volt=read_volt(Vme
volt:=volt=read_volt(Vme
volt:=volt=read_volt(Vme
volt:=volt=read_volt(Vme
volt:=volt=read_volt(Vme
volt:=volt=read_volt(Vme
volt:=volt=volt=volt=volt=volt=volt
volt:=volt=volt=volt=volt=volt=volt=volt=volt</pre> | <pre>56:2, dumstr2);
utstr+' + dumstr2;
autstr+' + dumstr2;
[idex] 33:0, dumstr2);
utstr+' ' + dumstr2);
utstr+' ' + dumstr2);
utstr+' ' + dumstr2);
power [point] := read_gain (Tcon
index] : 4:1, dumstr2);</pre> | <pre>:6:2, dumstr2);
utstr+' '+dumstr2;
utstr+' '+dumstr2;
index:):3:0, dumstr2);
volt==volt+rad_volt(Ymete
volt:=volt+rad_volt(Ymete
volt:=volt+rad_volt(Ymete
volt:=volt+rad_volt(Ymete
volt:=volt+rad_volt(Ymete
volt:=volt+rad_volt(Ymete
volt:=volt+rad_volt+rad_volt(Ymete
volt:=volt+rad_volt+rad_volt(Ymete
volt:=volt+rad_volt+rad_volt+rad_volt(Ymete
volt:=volt+rad_volt+rad_volt+rad_volt(Ymete
volt:=volt+rad_volt+rad_volt+rad_volt+rad_volt(Ymete
volt+rad_volt</pre> | <pre>:6:2, dumstr2);
utstr+' '+dumstr2;
utstr+' '+dumstr2;
idds:):3.4 dumstr2);
utstr+' '+dumstr2);
utstr+' '+dumstr2);
utstr+' '+dumstr2);
utstr+' '+dumstr2);
utstr+' '+dumstr2);
utstr+'
'+dumstr2);
jower[point]:=read_pain[point]:=read_gain[poin</pre> | <pre>:6:2,dumstr2);
uttr:+' + dumstr2);
uttr:+' - uttr:+' = currend_temp(Tcontro
curr:=currend_up(Tcont(Im
and = 1:3:0,dumstr2);
utatr+' + dumstr2);
nver[point]:=read_pater_pc
utstr+' + dumstr2);
utstr+' + dumstr2);
utstr+' + dumstr2);
jidex]:4:1,dumstr2);</pre>
 | :6:2,dumetr2);
utetr+' +dumetr2;
utetr+' +dumetr2;
(index]:3:0,dumetr2);
utetr+' +dumetr2);
utetr+' +dumetr2);
utetr+' +dumetr2);
index]:4:1,dumetr2);
end; [loop]
index]:4:1,dumetr2);
power[point]:=read_heater_pc
index1:4:1,dumetr2);
power[point]:=read_heater_pc | <pre>:6:2, dumstr2);
utstr+' '+dumstr2;
tem:=tem+read_temp(fcontro
curr:=curr+read_current(Ir
volt:=volt+read_volt(Vmeter.de
utstr+' '+dumstr2);
index]:4:1, dumstr2);
utstr+' '+dumstr2);
utstr+' '+dumstr2);</pre>
 | <pre>:6:2, dumstr2);
utetr+' + dumstr2;
a a a a a a a a a a a a a a a a a a a</pre> | :6:2, dumetr2);
utetr+' '+dumetr2;
************************************ | :6:2,dumstr2);
utetr+''-dumstr2;
utetr+''-dumstr2;
tates::::0,dumstr2);
utetr+'''+dumstr2);
utetr+'''+dumstr2);
onds]:4:1,dumstr2);
end; [loop]
 | :6:2,dumatr2);
utatr+ '+dumatr2;
tatatrassassassassassassassassassassassassass | :6:2,dumetr2);
utetr+' ^+dumetr2;
tem:=tem+read_temp(Tcontr | :6:2,dumetr2);
utetr+''+dumetr2;
 | :6:2,dumstr2);
utetr+^^ /+dumstr2; |
<pre>:6:2, dumstr2); utstr+` '+dumstr2; ************************************</pre>	<pre>:6:2, dumstr2); utetr+` '+dumstr2; itetr+` '+dumstr2; (idex]:3:0, dumstr2); utetr+` '+dumstr2); utetr+` '+dumstr2); power [point]:=read _pater_p power [point]:=read _gain[foont]:=read _gain[foont]:=read _gain[foont]:=read _gain[foont];=read _gain[foont];</pre>	<pre>:6:2, dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2); utitdr=):3:0, dumstr2); utitdr=):3:0, dumstr2); utitr+' '+dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2);</pre>	<pre>:6:2, dumstr2); utstr+' 'dumstr2); utstr+' 'dumstr2; eur:=curt+read_temp(Tcontro curt:=curt+read_current(Im volt:=volt+read_volt(Vmete nv:=v+read_nv(nVmeter.dev end; [loop] utst+' 'dumstr2; utst+' 'dumstr2; index]:4:1, dumstr2; index]:4:1, dumstr2;</pre>	<pre>:6:2, dumetr2); utetr+' 'dumetr2); utetr+' 'dumetr2; (index13:0, dumetr2); utetr+' 'dumetr2); utetr+' 'dumetr2); utetr+' 'dumetr2); utetr+' 'dumetr2); utetr+' 'dumetr2); utetr+' 'dumetr2); gain[point]:=read_gain[Tcont];</pre>	<pre>:6:2, dumstr2); utetr+` + dumstr2; ************************************</pre>	<pre>:6:2, dumetr2); utetr+' 'dumetr2); utetr+' 'dumetr2; utetr+' 'dumetr2); volt+read_volt(Vmete utetr+' 'dumetr2); nv:nv*read_nv(nVmeter.do; index]:4:1, dumetr2); utetr+' 'dumetr2); utetr+' 'dumetr2);</pre>	<pre>:6:2, dumetr2); utetr+' '+dumetr2; ************************************</pre>	<pre>:6:2, dumetr2); utetr+ ' +dumetr2; ************************************</pre>	<pre>:6:2, dumstr2); utstr+ ' +dumstr2; ************************************</pre>	:6:2, dumetr2); utetr: '+dumetr2); tutetr: '+dumetr2; (index]:3:0,dumetr2); nv:=nv+read_volt(Vmeter index]:4:1,dumetr2); end; [loop] end; [loop]	<pre>begin tear=tearread teap(Tcontr tear=tearread teap(Tcontr</pre>	:6:2,dumstr2); utstr+ '+dumstr2; tem:=tem+read_temp(Tcontr	:6:2,dumstr2); utstr+''+dumstr2;
<pre>65.dumstr2); tstr+' '4dumstr2; tstr+' '4dumstr2; tstr+' '4dumstr2; tstr+' '4dumstr2; utstr+' '4dumstr2; ndex1:3:0,dumstr2; ndex1:4:1,dumstr2; end; [loop] tstr+' '4dumstr2; power[point]:=read_heater.</pre>	<pre>feature::::::::::::::::::::::::::::::::::::</pre>	<pre>6:2,dumstr2); teatr+ ' +dumstr2); teatr+ ' +dumstr2); teatr+ ' +dumstr2); teatr+ ' +dumstr2); utetr+ ' +dumstr2); utetr+ ' +dumstr2); utetr+ ' +dumstr2); utetr+ ' +dumstr2); utetr+' ' +dumstr2); power[point]:=read_heater_po gain[point]:=read_heater_po gain[point]:=read_heater_po gain[point]:=read_heater_po</pre>	<pre>6:2.dumstr2); itstr+` '+dumstr2); itstr+` '+dumstr2); itstr+` '+dumstr2); ifact+` '+dumstr2); itstr+` '+dumstr2); itstr+` '+dumstr2); itstr+` '+dumstr2); power[point]:=read_psin[point]:=read_gsin[point]:=r</pre>	<pre>begin text://dumstr2); text://dumst</pre>	<pre>begin ttatr+' '+dumstr2'; ttatr+' '+dumstr2'; ttatr+' '+dumstr2'; itatr+' '+dumstr2'; itatr+' '+dumstr2'; nds:1:3:0,dumstr2'; itatr+' '+dumstr2'; end; [loop] itatr+' '+dumstr2'; end; [loop] inder1:4:1,dumstr2);</pre>	<pre>is::.dumstr2); itstr+' 'dumstr2; itstr+' 'dumstr2; itstr+' 'dumstr2; itstr+' 'dumstr2; itstr+' 'dumstr2; itstr+' 'dumstr2; itstr+' 'dumstr2; itstr+' 'dumstr2;</pre>	<pre>itstr+' '+dumstr2); itstr+' '+dumstr2); itstr+' '+dumstr2); itstr+' '+dumstr2); itstr+' '+dumstr2); itstr+' '+dumstr2); itstr+' '+dumstr2); power[point]:=read heater point]:</pre>	<pre>6:2,dumert2); tstr+' 'ddumert2; tstr+' 'ddumert2; tstr***********************************</pre>	<pre>is:2.dumstr2); itstr+``dumstr2); itstr+``dumstr2; itstr+``dumstr2; itstr+``dumstr2; itstr+``dumstr2; nV:=nV+read_nV(nVmeter.dev itstr+``+dumstr2; itstr+``+dumstr2; index]:4:1,dumstr2;</pre>	<pre>i6:2.dumst2); itst*' *dumst2); itst*' *dumst2; itst*' *dumst2; itst*' *dumst2; itst*' *dumst2; itst*' *dumst2; itst*' *dumst2; itst*' *dumst2; itst*' *dumst2; itst*' *dumst2; itst*' *dumst2;</pre>	6:2, dumatr2); tem:=tem+read_temp(Tcontr tem:=tem+read_temp(Tcontr	tem:=temtread_temp(Tcontre text:/ '+dumstr2'; text:/ '+dumstr2';	test
<pre>6:2,dumetr2); 6:2,dumetr2); tetr+' +dumetr2; tetr+' '+dumetr2; tetr+' '+dumetr2; index]:3:0,dumetr2; ndex]:4:1,dumetr2; end; [loop] tetr+' '+dumetr2; volt:=nv+read_nv(nVmeter.c end; [loop] tetr+' '+dumetr2; end; [loop]</pre>	<pre>rest.rest.rest.rest.rest.rest.rest.rest.</pre>	<pre>tet:-/ dumetr2); tet:/ dumetr2); tet:-/ dumetr2); tet:-/ dumetr2); tet:-/ dumetr2); tet:-/ 'dumetr2); tet:-/ 'dumetr2); tet:-/ 'dumetr2); tet:-/ 'dumetr2); tet:-/ 'dumetr2); tet:-/ 'dumetr2); tet:-/ 'dumetr2);</pre>	<pre>tet:-/ dumetr2); tet:/ '+dumetr2); tet:-/ '+dumetr2); index]:3:0,dumetr2); tetr+' '+dumetr2); tetr+' '+dumetr2); tetr+' '+dumetr2); tetr+' '+dumetr2); power[point]:=read_heater_po gain[point]:=read_heater_po gain[point]:=read_gain(point]:=re</pre>	<pre>weiter://</pre>	<pre>weit.dumetr2); teit.+ ' +dumetr2); teit.+ ' +dumetr2); teit.+ ' +dumetr2); index]:3:0,dumetr2); teit.+ ' +dumetr2); teit.+ ' +dumetr2); modex]:4:1,dumetr2); teit.+ ' +dumetr2); teit.+ ' +dumetr2); teit.+ ' +dumetr2); teit.+ ' +dumetr2); teit.+ ' +dumetr2); teit.+ ' +dumetr2);</pre>	<pre>weit2.dumetr2); tetz.dumetr2); tetz.dumetr2); tetz.dumetr2); tem:read_remp(%contrc tetz.dumetr2); index]:3:0,dumetr2); tetzr+ ' +dumetr2); mdex]:4:1,dumetr2); tetzr+ ' +dumetr2);</pre>	<pre>week_control week_control tetr+' + dumstr2; tetr+' + dumstr2; tetr+' + dumstr2; inder]:3:0,dumstr2; inder]:3:0,dumstr2; der+' + dumstr2; tetr+' + dumstr2; tetr+' + dumstr2; tetr+' + dumstr2; tetr+' + dumstr2;</pre>	<pre>certification control con</pre>	<pre>weit.dimetr2); tet:/dimetr2); tet:/dimetr2/ tem:=tem:trad_temp(Tcontro tem:=temtrad_temp(Tcontro text:/dimetr2); nV:=nV+read_nV(nVmeter.dev tetr+' '+dumetr2); end; [loop]</pre>	<pre>vert for the provide state of the provide stat</pre>	<pre>begin begin the structure of the st</pre>	<pre>ver_verter.times11; 6:2,dumetr2); tetr+' '+dumetr2;</pre>	<pre>verc_voir(setPr(inver]); bf://dumetr2); tetr+/ - dimetr2); tem:fem+read temp[Tcontrc</pre>
<pre>vert_voir(setpelinoex)); isi2, dumatr2); isi2, dumatr2); isi2, dumatr2; isi2, dumatr2; isi2, dumatr2; isi2, 'dumatr2); ndex1:4:1, dumatr2; isi2, 'dumatr2); power[poind]:read_heater.com isi2, (loop) isi2, 'dumatr2; power[poind]:read_heater.com isi2, dumatr2;</pre>	<pre>VVEC_C_Wold(secplindexi); isi:2,dumatr2); isi:2,dumatr2); isi:2,dumatr2); isi:2,dumatr2); isi:2,dumatr2); index]:3:0,dumatr2); index]:4:1,dumatr2); power[point]:rread_holter_p power[point]:rread_gain[Toon index]:4:1,dumatr2);</pre>	<pre>vver_volt(setpe(index)); i6:2,dumatr2); i6:2,dumatr2); if i</pre>	<pre>vver_volt(setpe(index)); i6:2,dumatr2); begin i1:::**********************************</pre>										
 | <pre>vver vout desp(index);;
isi2.dumatr2;
isi2.dumatr2;
isi2.dumatr2;
isi2.dumatr2;
isi2.dumatr2;
index]:3:0,dumatr2;
index]:4:1,dumatr2;
noti1:volt+read_volt(Numcter.dev
volt+read_volt+read_volt(Numcter.dev
volt+read_volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt(Numcter.dev
volt:volt+read_volt+read_volt(Numcter.dev
volt:volt+read_volt+read_volt+read_volt+read_volt+read_volt(Numcter.dev
volt:volt+read_volt+read</pre>
 | <pre>vver_vvir_setpe(index);;
vist_unstr2;
isis_dumstr2;
isis_dumstr2;
isis_odumstr2;
isis_odumstr2;
isis_vvir_ent_end_end_current(Im
volt:=volt+read_volt(Vmete
isis_vvir_entr2;
isis_vvir_entr2;
isis_vvir_entr2;
index1:4:1,dumstr2;
index1:4:1,dumstr2;
index1:4:1,dumstr2;
index1:4:1,dumstr2;</pre> | <pre>vver.volt(setPp(lindex));
is:2,dumatr2);
is:1,dimatr2);
is:1,dimatr2);
is:1,dimatr2);
is:1,dex]:3:0,dumatr2);
is:1,dumatr2);
is:1,dumatr2);
is:2,dumatr2);
is:2,dumatr2);
is:2,dumatr2);
is:2,dumatr2);</pre> | <pre>vver.vout(setpe(index));
its12.dumatr2);
tts12.dumatr2);
tts12.dumatr2);
tts12.dumatr2);
tts14.food(unatr2);
index]:30,dumatr2);
index1:
(-dumatr2);
end; [loop]
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];
tts14.food([loop]];</pre> | <pre>vver_vver_setp(index1);
i6:2,dumstr2);
i6:2,dumstr2);
tem:=tem+read_temp(Tcontro]
tem:=tem+read_temp(Tcontro]
testr*' *dumstr2);
itstr*' *dumstr2);
itstr*' *dumstr2);
itstr*' *dumstr2);
itestr*' *dumstr2);
itestr*' *dumstr2);</pre> | <pre>VMEr_ Volt(setpt[index]);
i6:2,dumatr2);
i6:2,dumatr2);
ifistr* '+dumatr2);
ifistr* '+dumatr2);
ifistr* '+dumatr2);
ifidex]:3:0,dumatr2);
ifidex]:4:1,dumatr2);
index]:4:1,dumatr2);
end; [loop]</pre>
 | <pre>VPerC_VOIC(Becpt[Index]);
6:0; dumatr2);
1:str************************************</pre> | <pre>vert_vour(setpt[incex]);</pre> | vert_voit(setpt[index]);
:6:2,dumstr2);
itstr+''+dumstr2; | Vert Volt(setPt[lnoex]);
begin [tel:/dumatr2];
tes:/routrostraced temp[Tcontro
 |
<pre>nvert Volt (effpt[index]); tem:=temtred_temp(Tcont uteirt+ '+dumstr2); tem:=temtread_temp(Tcont tem:=temtread_temp(Tcont temtreatread_temp(Tcont uteir+' +dumstr2); uteir+' +dumstr2); uteir+' +dumstr2); uteir+' +dumstr2); uteir+' +dumstr2); uteir+' +dumstr2); uteir+' +dumstr2);</pre>	<pre>rvert_voit (setp(lindex]); cst.fdumstr2); tst.fdumstr2); tst.fdumstr2) tst.fdumstr2) (index]:3:0,dumstr2); utstr+' +dumstr2); utstr+' +dumstr2); index]:4:1,dumstr2); power[point]:rread_pater_p gain[point]:rread_gain[Toon index]:4:1,dumstr2);</pre>	<pre>nvert voit (setpt[index]); isiz_dumstr2); utstr+ '+ '+dumstr2); utstr+ '+ '+dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); utstr+ '+dumstr2); utstr+ '+dumstr2); utstr+ '+dumstr2); utstr+ '+dumstr2); utstr+ '+dumstr2);</pre>	<pre>nvert Volt (setpt[index]); isiz_dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); power[point]:=read_pater_po gin[point]:=read_gin[point];=read_gin[point]:=read_gin[point]:=read_gin[point]:=read_gin[point];=read_gin[point</pre>	<pre>nvert_voit (setpt[index]); isiz_dumatr2); utstr.***********************************</pre>	<pre>nvert Voit (sectiindex)); isiz (umastr2); isiz (umastr2); isiz (umastr2); isiz (index):3:0, dumastr2); utatr+' + dumastr2; utatr+' + dumastr2; index1:4:1, dumastr2); index1:4:1, dumastr2); end; [loop] index1:4:1, dumastr2); power [point]:=read heater_pc qain[point]:=read deain[Tcont]</pre>	<pre>nvert Volt (setpt[lndex]); isiz.dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2);</pre>	<pre>nvert voit (setpt[index]); isi2, dumetr2); utit: dumetr2; ************************************</pre>	<pre>nvert voit (actpt[index]); begintemrtvl utatr:/ idumatr2; ************************************</pre>	<pre>nvert voit (actpt [index]); isi2, dumstr2); iutatr* '+dumstr2); intatr***********************************</pre>	<pre>nvert voit (setpt[index]); sis2, dumstr2); sist.dumstr2); sist.atist.sist.sist.sist.sist.sist.sist</pre>	<pre>nvert_voit (setpt[index]); test_voit (setpt[index]); begin ten:=tentread_temp(Tcontr tentread_temp(Tcontr tentread_temp(Tcontr tentread_temp(Tcontr tentread_temp(Tcontr tentread_temp(Tcontr tentread_temp(Tcontrad_temp(Tcont</pre>	nvert_voit (metpt[index]); :6:2,dummatr2); utatr+''+dummstr2; tem:=tem+read_temp(Tcontr	nvert Voit (eetptlindex]); bogin teist, vimmetr2); teist, vimmetr2);
<pre>nvert_Voit (setpt[index]); i = 1 to average_rea</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_read</pre>	<pre>nvert_voit (actpt[index]); :6:2, dumatr2); utatr* ' +dumatr2; nutatr* ' +dumatr2; index]:3:0, dumatr2); index]:3:0, dumatr2; index]:4:1, dumatr2; index]:4:1, dumatr2; index]:4:1, dumatr2; index]:4:1, dumatr2; index]:4:1, dumatr2; index]:4:1, dumatr2; index]:4:1, dumatr2;</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_readi sets_dumatr2; utstr+ ' +dumatr2); tem:=tem+read_temp(fcontro curr:=curr+read_temp(fcontro volt:=read_volt(Ymete NV:=nV+read_nV(nVmeter.dev end; [loop] utstr+' ' +dumatr2); utstr+'</pre>	<pre>snvert_Volt(setpt[index]); fist2.dumstr2); sutstr.dumstr2: sutstr.dumstr2: sutstr.dumstr2: sutstr.dumstr2); sutstr.t. *dumstr2; findex]:3:0,dumstr2); sutstr.t. *dumstr2; sutstr.t. *dumstr2; index]:4:1,dumstr2; sutstr.t. *dumstr2; findex]:4:1,dumstr2; gain[point]:=read_gain[Tcont</pre>	<pre>sourcert_Volt(setpt[index]); fiet2,dumstr2); begin begin begin begin begin tem:rtemtread_temp(Tcontro temp(Tcontro temp(Tc</pre>	<pre>sourcert_Volt(setpt[index]); suist:/dumstr2); begin begin tem:=temtread_temp[fcontre swiit:***********************************</pre>	<pre>Drvert_Volt(setpt[index]); fs:2,dumstr2); Dutstr+ ' +dumstr2; Dutstr+ ' +dumstr2; r[index1:3:0,dumstr2); r[index1:3:0,dumstr2); nvtstr+' +dumstr2); fundex1:+1:1,dumstr2); power[point]:=read heater point[index1];</pre>	<pre>sourcert_Volt(setpt[index]); fie:2,dumstr2); begin begin begin begin tem:=tem+read_temp(Tcontrol tem:=tem+read_temp(Tcontrol tem:=tem+read_temp(Tcontrol tem:=tem+read_temp(Tcontrol tem:=tem+read_temp(TechtIme volt:=volt+read_ov[t(whete:dev) outstr+' ' +dumstr2); nV:=nV+read_ov[t(whete:dev) temtrad_ev] t(nVmeter.dev) t(nVmeter.dev) to text:=volt=volt=volt=volt=volt=volt=volt=volt</pre>	<pre>sourcer_Volt(setpt[index]); /:6:2,dumstr2); butatr.' +dumstr2); sutatr.' +dumstr2; satatatatatatatatatatatatatatatatatatat</pre>	<pre>Snvert_Volt(setpt[index]); sist_dumstr2); begin snistriitintaitr2; snistriitintaitr2; snistriitintaitr2; snistriitintaitr2; solt:=volt+read_ovlt(Nmeter.devi volt:=volt+read_ovlt(Nmeter.devi sutstr*' * dumstr2; index]:4:1,dumstr2; findex]:4:1,dumstr2;</pre>	<pre>onvert_Volt(setpt[index]); fs:2,dumstr2); begin ten:=tentread_temp[contr tentread_temp[contr tentread_temp[contrad_temp[con</pre>	<pre>onvert_Volt(setpt[index]); fs6:2,dumstr2); bsgin tem:=tem+read_temp(Tcontrc tem:=tem+read_temp(Tcontrc tem=tem+read_temp(Tcontrc tem=tem+read_temp(Tcontrc tem=tem+tem+tem+tem+tem+tem+tem+tem+tem+tem+</pre>	<pre>onvert_Volt(aetpt[index]); f:6:2,dumatr2);</pre>
<pre>nvert_Volt(setpt[index]); for loop:=l to average_rea for loop:=nvelt=read_temp(Tcont volt:=volt=read_temp(Tcont volt:=volt=read_volt(Vm volt=read_volt(Vm volt=read</pre>	<pre>nvert_Volt(setpt[index]); for loop:=l to average_read tead tead teap(Tcontr begin tstr+' '+dumstr2); tindex]:3:0,dumstr2); tindex]:4:1,dumstr2); tott+' '+dumstr2); tutet+' '+dumstr2); tutet+'+dumstr2); tutet+' '+dumstr2); tutet+''+dumstr2); tutet+</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_readi</pre>	<pre>nvert_Volt(setpt[index]); i 6:2, dimstr2); begin i 6:2, dimstr2); begin utetr+ ' + dimstr2); begin (index]:3:0, dumstr2); utetr+ ' + dimstr2); utetr+ ' + dimstr2); utetr+' ' + dimstr2); utetr+' ' + dimstr2); utetr+' (index]:4:1, dimstr2); utet</pre>	<pre>snvert_volt(setpt[index]); sisi2,dumstr2); utstr1 ' +dumstr2); utstr1 ' +dumstr2: i(index]:3:0,dumstr2); sutstr4 ' +dumstr2); utstr4 ' +dumstr2);</pre>	<pre>invert_volt(setpt[index]); is:2,dumstr2); is:2,dumstr2); is:2,dumstr2); is:1,dumstr2); is:1</pre>	<pre>invert_Volt(setpt[index]); is is z_dumatr2); utstr+' '+dumatr2); utstr+' '+dumatr2; intitation: is index[::0,dumatr2); is index[::0,dumatr2); is index[::0,dumatr2); is index[::0,dumatr2); is index[::1,dumatr2); is index[:1,dumatr2]; is index[:1,dumatr2); is index[:1,dumatr2); is index[:1,dumatr2]; is index[:1,du</pre>	<pre>snvert_volt(setpt[index]); s:6:2,dumstr2); begin s:6:2,dumstr2); utstr+' *dumstr2; utstr+' '+dumstr2; sursed_temp(Tcontro' curri-curr+read_temp(Tcontro' sursert.employed_current(Im volt:=volt+read_volt(Vmete: nvient+r2); nvient+' '+dumstr2; end; [loop] power[point]:=read heater po uutstr+' '+dumstr2;</pre>	<pre>invert_volt(setpt[index]); is is dumatr2); begin tem:=tem+read_temp(Tcontrol tem:=tem+read_temp(Tcontrol tem:=tem+read_temp(Tcontrol tem:=tem+read_temp(Tcontrol tem:=tem+read_temp(Tcontrol tem=reatrined_temp(Tcontrol tem=reatrined_temp(Tcontrol tem=reatrined_temp(Tcontrol tem=read_temp(Tcontrol tem=read_temp(Tem temp(Tem tem=read_temp(Tem tem=read_temp(Tem temp(Tem tem=read_temp(Tem temp(Tem tem=read_temp(Tem temp(Tem tem temp(Tem temp(Tem temp(Tem te</pre>	<pre>invert_volt(setpt[index]); for loop:=1 to average_readi ::5:2,dumstr2); begin uutatr1, 'dumstr2); curread_temp(fcontro) initiatrationsing); curread_current(Im volt:=volt+read_volt(Nmeter voltstr4' 'dumstr2); end; (loop) (index]:4:1,dumstr2); end; (loop)</pre>	<pre>snvert_volt(setpt[index]); sist_dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2); intext+***********************************</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_read .:6:2,dumatr2); ten:=tentread_temp(Tcontr vutstr+' ^+dumatr2;</pre>	<pre>nvert_Volt(setpt[index]);</pre>	<pre>nnvert_Volt(setpt[index]);</pre>
<pre>nvert_Volt(setpt[index]); for loop:=1 to average_rea for loop for loop</pre>	<pre>nvert_Volt(setpt[index]); i6:2,dumstr2); i6:2,dumstr2); i1:e1:**********************************</pre>	<pre>nvert_Volt(aetpt[index]); for loop:=1 to average_readi =6:2,dumatr2); utatr+ '+dumatr2); utatr+ '+dumatr2; utatr+ '+dumatr2); (index]:3:0,dumatr2); utatr+ '+dumatr2); index]:4:1,dumatr2); utatr+ '+dumatr2); utatr+' '+dumatr2);</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_read; begin i:6:2,dumatr2); begin uttr+' + 4umatr2; tem:=tem+read_temp(Fcontro tem:=tem+read_temp(Fcontro tem:=tem+read_temp(Fcontro tem:=tem+read_volt(Vmete volt:=volt+read_volt(Vmete volt=volt+read_volt(Vmete volt+read_volt(Vmete volt=volt+read_volt(Vmete volt=vol</pre>	<pre>nnvert_Volt(setpt[index]); for loop:=1 to average_readi for loop:=1 t</pre>	<pre>nnvert_Volt(setpt[index]); s:6:2,dumstr2); s:6:2,dumstr2); utstr+' *dumstr2); utstr+' *dumstr2);</pre>	<pre>nvert_Volt(setpt[index]); sisis,dumatr2); utstr+` '+dumatr2); tem:=tem+read_temp(Tcontrc tem:=current(int i(index]::0,dumatr2); volt:=volt+read_volt(Vmete volt=read_volt(Vmeter.de i(index]::0,dumatr2); power[point]:=read_hoster_power[point]:=read_</pre>	<pre>nnvert_Volt(setpt[index]); for loop:=1 to average_readi ":6:2,dumatr2); utstr+' +dumatr2; utstr+' '+dumatr2; utstr+' '+dumatr2); utstr+' '+dumatr2; intert+' '+dumatr2;</pre>	<pre>nnvert_Volt(setpt[index]); for loop:=1 to average_readin :s6:2,dumatr2); begin tem:=tem+read_temp(Tcontrol utstr+' '+dumatr2); tem:=tem+read_temp(Tcontrol utstr+' '+dumatr2); nvi=nv+read_nv(nvmeter.devi tindex]:3:0,dumatr2); nvi=nv+read_nv(nvmeter.devi tindex1); nvi=nv+read_nv+read_nv(nvmeter.devi tindex1); nvi=nv+read_</pre>	<pre>snvert_volt(setpt[index]); si5:2_dumstr2); begin si5:2_dumstr2); sutstr**********************************</pre>	<pre>nvert_volt(setpt[index]); si6:2,dumstr2); utstr+' '+dumstr2); tem:=tem=tread_temp(fcontrol tem=traid_temp(fcontrol temp(fcontrol temp(fcontrol</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_read .:6:2,dumatr2); tem:read_temp(fcontr utstr+' ^+dumatr2;</pre>	<pre>nvert_Volt(setpt[index]);</pre>	for loop:=1 to average_read; v:6:2,dumatr2); utstr+' '+dumatr2;
<pre>nvert_Volt(setpt[index]); for loop:=1 to average_res for loop:=1 to average_res for loop:=1 to average_res for loop:=1 to average_res for utstr+' '+dumstr2); for loop:=1 to average_res for loop:=1 to average, for loop</pre>	<pre>nvert_Volt(setpt[index]); i6:2,dumstr2); i6:2,dumstr2); i1:2:,dumstr2); i1:2:,dumstr2); i1:1:2:::::::::::::::::::::::::::::::::</pre>	<pre>nvert_Volt(setpt[index]); sets2, dumatr2); utstr+* '+dumatr2); utstr+** '+dumatr2); utstr+** '+dumatr2); utstr+** '+dumatr2); utstr+***********************************</pre>	<pre>nvert_Volt(setpt[index]); i 6:2, dumetr2); i 6:2, dumetr2); i cit 1 oop:=1 to average_read; begin i i cit + ' + dumetr2); i cut + ' + ' dumetr2); i cut + ' + ' + ' + ' + ' + ' + ' + ' + ' +</pre>	<pre>nvert_volt(setpt[index]); :6:2,dumstr2); utstr+' *dumstr2; utstr+' *dumstr2; itindex]:3:0,dumstr2); curr:=curr+read_temp(Tcontro curr:=curr+read_temp(Tvete volt("v</pre>	<pre>nvert_Volt(setpt[index]); :=6:2,dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2; internationantionantionantionantionantionantion internationantionantionantionantion internationantionantion utstr+' '+dumstr2); [index]:+1.1,dumstr2); power[point]:=read_heater_po dain[point]:=read_heater_po dain[point]:=read_heater_po dain[point]:=read_heater_po dain[point]:=read_heater_po dain[point]:=read_heater_po</pre>	<pre>nvert_Volt(setpt[index]); i:6:2,dumstr2); iutstr+' *dumstr2); begin intatration = = = = = = = = = = = = = = = = = = =</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_readi .:6:2,dumatr2); utstr+` *dumatr2; .utstr+` *dumatr2; .utstr+` *dumatr2); utstr+` *dumatr2); uutst+` *dumatr2); </pre>	<pre>nvert_Volt(setpt[index]); :=6:2,dumstr2); uutstr+` '+dumstr2); uutstr+` '+dumstr2); uutstr+` '+dumstr2); uutstr*` '+dumstr2); uutstr*` '+dumstr2); uutstr*` '+dumstr2); outstr*` '+dumstr2);</pre>	<pre>nvert_volt(setpt[index]); for loop:=1 to average_readi .:6:2,dumatr2); utstr** '*dumatr2; utstr** '*dumatr2; inter** '*dumatr2; index]:3:0,dumatr2; inter** '*dumatr2; inter************************************</pre>	<pre>nvert_volt(setpt[index]); is:2,dumstr2); vutstr*' *dumstr2; is:2,dumstr2: is:1,dumstr2: is:1,dumstr2: is:1,dumstr2); is:1</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_read .:6:2,dumatr2); tem:read_temp(fcontr vutstr+' ^+dumatr2;</pre>	<pre>nvert_Volt(setpt[index]); rsc:2,dumstr2); ten:=tentread_tentr</pre>	<pre>nvert_volt(setpt[index]);</pre>
<pre>nvert_Volt(actpt[index]); i6:2,dumatr2); i6:2,dumatr2); utatr+' '+dumatr2); tem:=tem+read_temp(Tcont utatr+' '+dumatr2); utatr+' '+dumatr2); utatr+' '+dumatr2); utatr+' '+dumatr2); power[point]:=volt+read_volt(Vme volt:=volt+read_volt(Vme volt=volt+volt+read_volt(Vme volt=volt+volt+volt+volt+volt+volt+volt+volt+</pre>	<pre>nvert_Volt(aetpt[index]); for loop:=1 to average_read begin tem:=tem+read_temp(Tcontr itetr+` '+dumstr2; tem:=tem+read_temp(Tcontr itetr+` '+dumstr2; itetr>; '+dumstr2; itetr; '+dumstr2; itetr; '+dumstr2; itetr]; '-read_gain(Tcon itetr>; '+dumstr2; itetr]; '-read_gain(Tcon itetr>; '+dumstr2; itetr]; '-read_gain(Tcon itetr>; '+dumstr2; itetr]; '-read_gain(Tcon itetr>; '+dumstr2; itetr2; itetr]; '-read_gain(Tcon itetr>; '+dumstr2; itetr]; '-read_gain(Tcon itetr2; itetr>; '+dumstr2; itetr2; itetr2; itetr]; '-read_gain(Tcon itetr>; '+dumstr2; itetr2; itetr]; '-read_gain(Tcon itetr>; '+dumstr2; itetr2; itetr]; '-read_gain(Tcon itetr2; itetr2; itetr2; itetr2; itetr2; itetr2; itetr2; itetr3; itetr3;</pre>	<pre>nvert_Volt(setpt[index]); is:2, dumatr2); utetr+' + dumatr2); utetr+' + dumatr2); utetr+' + dumatr2); utetr+' - + dumatr2); utetr+' - + dumatr2); utetr+' - + dumatr2); index]:4.1, dumatr2); ind</pre>	<pre>nvert Volt (setpt[index]); i = 1 to average_readi begin i = 1 to average_readi tem=read_temp(Tcontro tem=read_temp(Tcontro</pre>	<pre>invert_Volt(setpt[index]); f=:2,dumstr2); p=:1 to average_read; p=:1 to average_read; p=:2,dumstr2); p=:1 to average_read; p=:1</pre>	<pre>invort voit (setpt[index]); invort voit (setpt[index]); is:2,dmmstr2); is:2,dmmstr2); intert+ ' +dumstr2); intert+ ' +dumstr2; int</pre>	<pre>inter: for loop:=1 to average_read; f:6:2,dumetr2); nutetr+' +dumetr2; nutetr+' +dumetr2; reinter: inter: int</pre>	<pre>invert_Volt(setpt[index]); invert_Volt(setpt[index]); is:2,dumstr2); intert+' +dumstr2; intert+' +dumstr2; intert+' +dumstr2; intert+read_current(Imter intert+' +dumstr2); intert+read_volt(vmteridev intert+' +dumstr2); in</pre>	<pre>nvert_Volt(setpt[index]); ris2,dumstr2); begin begin tem:=tem+read_temp(Tcontrol outstr+' '+dumstr2); nviintiintiintiintiintiintiintiintiintii</pre>	<pre>nvert_Volt(setpt[index]); for loop:=1 to average_readi.</pre>	<pre>nvert Volt(setpt[index]); ref2,dumstr2); utstr1 ' +dumstr2: intext1 ' +dumstr2: intext1:::utstr2: intext1:::utstr2); utstr4 ' +dumstr2); utstr4 ' +dumstr2); index]:4:1,dumstr2); end; [loop]</pre>	<pre>invert Volt(setpt[index]); for loop:=1 to average_read p:6:2, dumatr2); tean:read_teap(fontr tean:read_teap(fontr tean:read_teap(fontr); tean:read_teap(fontr); }</pre>	for loop:=1 to average_read: y:6:2,dumstr2); putstr+' '+dumstr2;	for loop:=1 to average_read: y:vert_Volt(setpt[index]); begin tem:=tem+read temp[fcontro tem:=tem+read temp[fcontro
<pre>come voltage conversion nvert_Voltage conversion utetr+' undtr2; tem:=tem+read_temp(Tcont tem:=tem+read_temp(Tcont tem:=tem+read_temp(Tcont volt:=volt+read_volt(Vme volt:=volt+read_volt(Vme volt:=volt+read_volt(Vme volt:=volt+read_volt(Vme volt:=volt=read_volt(Vme volt:=volt=read_volt(Vme volt:=volt=read_volt(Vme volt:=volt=read_volt(Vme volt:=volt=read_volt(Vme volt=volt=volt=volt=volt=volt=volt=volt=</pre>	<pre>cert_Volt detpt[index]); Avert_Volt detpt[index]); ist:/dmmstr2); utstr+' '+dumstr2); [index]:3:0.dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2); power[point]:rread_pater_p index]:4:1,dumstr2); power[point]:rread_gain(Tcon index]:4:1,dumstr2);</pre>	<pre>construction construction construction</pre>	<pre>out of the convertation // // // // // // // // // // // // //</pre>	<pre>code Youts (set [index]); invert_Volt (set p[index]); is 52 dumatr2); uutatr+' '+dumatr2); uutatr+' '+dumatr2); uutatr+' '+dumatr2); [index]:4:1, dumatr2); power [point]:=read_yout(Tymeter_Do power [point]:=read_gain(Tcont [index]:4:1, dumatr2); [index]:4:1, dumatr2);</pre>	<pre>code Yolt(setPt[index]); rist_tuntity); fist_tuntity); rist_tuntity); rist_tuntity); rist_tuntity); rist_ty, 'dumstr2); rist_ty, 'dumstr2); rist_ty, 'dumstr2); rist_ty, 'dumstr2); rist_ty, 'dumstr2); rist_ty, 'dumstr2); rist_ty, 'dumstr2); rist_ty, 'dumstr2); rist_ty, 'dumstr2); rist_ty, 'dumstr2);</pre>	<pre>code Yottage conversion instruction rists.dummatr2; rists.dummatr2; rutatr+ '+dummatr2; rutatr+ '+dummatr2; rutatr+ '+dummatr2; rutatr+' '+dummatr2; rutatr+' '+dummatr2; rutatr+' '+dummatr2; rutatr+' '+dummatr2; rutatr+' '+dummatr2; rutatr+' '+dummatr2;</pre>	<pre>code Volt(setpt[index]); is5:dumstr2); begin tem:=tempt[and:: butstr+`^+dumstr2); intratatatatatatatatatatatatatatatatatata</pre>	<pre>code Yout(setpt[index]); is6:dumstr2; is6:dumstr2; uutstr* '+dumstr2; uutstr* '+dumstr2; istaatustraatustraatustraatustraatustraatustraatustraatustraatustraatustraatust istaatustraatustraatustraatustraatustraatust istaatustraatustraatust uutstr*' +dumstr2; itdex]:s0,dumstr2); itdex]:s0,dumstr2); itdex]:s0,dumstr2); itdex]:s0,dumstr2);</pre>	<pre>code Yout(setpt[index]); invert_Volt(setpt[index]); is6:2,dumatr2); uutstr+' *hdumatr2; sutstr+' *hdumatr2); sutstr+' *hdumatr2); sutstr+' *hdumatr2); sutstr+' *hdumatr2); sutstr+' *hdumatr2); index]:4:1,dumatr2); [index]:4:1,dumatr2);</pre>	<pre>code Yottage control for loop:=1 to average_readin nvert_Vott(setpt[index]); for loop:=1 to average_readin begin tem:rtage_readin tem:rtage_readin begin tem:rtage_readin t</pre>	ode voltage conversion nvert Volt (setpt[index]); ri6:2,dumstr2); vutstr+' '+dumstr2;	<pre>.cod voltage conversion invert_Volt (sidex]); ifs2.dumatr2); utstr+' '+dumstr2;</pre>	<pre>code YoLtage conversion</pre>
<pre>ode voltage conversion************************************</pre>	<pre>ode voltage conversion************************************</pre>	<pre>ode voltage conversion************************************</pre>	<pre>ode voltage conversion************************************</pre>										
 | <pre>idoe voltage conversion************************************</pre>
 | <pre>idode voltage conversion************************************</pre> | <pre>idde voltage conversion************************************</pre> | <pre>idode voltage conversion************************************</pre>
 | <pre>lode voltage conversion************************************</pre> | <pre>lode voltage conversion************************************</pre>
 | <pre>(ode voltage conversion************************************</pre> | <pre>iode voltage conversion####################################</pre> | <pre>iode voltage conversion************************************</pre> | <pre>iode voltage conversion************************************</pre>
 |
| <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>idde voltage conversion************************************</pre>
 | <pre>idde voltage conversion************************************</pre> | <pre>idde voltage conversion************************************</pre> | <pre>idde voltage conversion************************************</pre>
 | <pre>lode voltage conversion************************************</pre> | <pre>lode voltage conversion************************************</pre>
 | <pre>(ode voltage conversion************************************</pre> | <pre>iode voltage conversion************************************</pre> | <pre>iode voltage conversion####################################</pre> | <pre>iode voltage conversion************************************</pre>
 |
| <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>[ode voltage conversion************************************</pre>
 | <pre>iode voltage conversion************************************</pre> | <pre>iode voltage conversion************************************</pre> | <pre>[ode voltage conversion************************************</pre>
 | <pre>lode voltage conversion************************************</pre> | <pre>lode voltage conversion************************************</pre>
 | <pre>(ode voltage conversion************************************</pre> | <pre>lode voltage conversion************************************</pre> | <pre>lode voltage conversion************************************</pre> | <pre>lode voltage conversion************************************</pre>
 |
| <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>ode voltage conversion************************************</pre>
 | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>ode voltage conversion************************************</pre> | ode voltage conversion************************************ | ode voltage conversion************************************ | ode voltage conversion************************************
 |
| <pre>ode voltage conversion************************************</pre> | <pre>de voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>ode voltage conversion************************************</pre>
 | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>ode voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>ode voltage conversion************************************</pre> | ode voltage conversion************************************ | <pre>ode voltage conversion************************************</pre> | ode voltage conversioninininininininininininininininininin
 |
| <pre>dd voltage conversion************************************</pre> | <pre>de voltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre> | <pre>dd voltage conversion************************************</pre>
 | <pre>dd voltege conversion************************************</pre>
 | <pre>dd voltese conversion************************************</pre> | <pre>dd voltage conversion************************************</pre> | <pre>de coltage conversion************************************</pre>
 | <pre>ode coltage conversion************************************</pre> | <pre>ode voltage conversion************************************</pre>
 | <pre>de voltage conversion************************************</pre> | <pre>dd voltage conversion************************************</pre> | uce of the conversion ************************************ | ucour rounder, NY:=0.0;
ode Voltage conversion####################################
 |
| <pre>utstr+ '+dumstr2) ode voltge conversion************************************</pre> | <pre>statt+' + dumatr;
det Voltage conversion************************************</pre> | <pre>utestr+ roumstr;
utestr+ roumstr;
nv:=0.0;
for loop:=1 to average_read;
for loop:=1 to average, loop;
for loop;
f</pre> | <pre>utstrt foundation
utstrt for conversion
over Voltage conversion
isi2, dumstr2);
utstrt ' +dumstr2);
utstrt ' +dumstr2);</pre> | <pre>utstr: 'toumstr:
ode voltage conversion************************************</pre> | <pre>utstr+ 'toumstr;
ode voltage conversion************************************</pre> | <pre>utstrt. 'toumstr) ode voltage conversion************************************</pre> | <pre>utstr: + dumatr;
ode voltage conversion************************************</pre> | <pre>utstr: + dumatr;
ode volt (set[index]);
if volt (set[index]);
utstr: * + dumatr2;
utstr: * + dumatr2;
utstr: * + dumatr2;
index]::0, dumatr2;
utstr: * + dumatr2;
index]::0, dumatr2;
volt:=volt+read_nv(nVmeter.devi
end; [loop]</pre> | <pre>utstr+ 'toumstr;
ode voltage conversion************************************</pre> | <pre>ucstr: ' - dumatr:
ode voltage conversion************************************</pre> | ucser+ + + connect;
ode voltage conversion************************************ | utstr+ - + dumstr;
ode voltage conversion************************************ | utstr+ - + qumstr;
ode voltage conversion************************************ |
| <pre>utstr+' '+dumstr;
ode voltge conversion************************************</pre> | <pre>volt:=0.0;
volt:=0.0;
oder Voltage conversion************************************</pre> | <pre>utetr+' + dumatr;
ode volt=0.0;
nver_voltge conversion************************************</pre> | <pre>uttr:+' ' + dumstr;
ode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
ultatr' 'dumatr;
lode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
lode volt(set[index]);
rist:dumstr2;
vist:+ '+dumstr2;
rist: dumstr2;
vutert+' '+dumstr2;
itidex]:3:0,dumstr2;
volt:=volt+read_volt(Vmete
volt:=volt+read_volt(Vmete
volt:=volt+read_volt(Vmete
volt:=ri:dumstr2;
itidex]:4:1,dumstr2;
cain[point]:=read_heater_pc
dain[point]:=read_heater_pc</pre> | <pre>volt:=0.0;
volt:=0.0;
for loop:=1 to average_read;
ris2,dumatr2);
vist:/dumatr2);
volt:=volt+read_temp(fcontrc
volt:=volt+read_volt(Vmete
volt:=volt+read_volt(Vmete.de
volt:=volt+read_volt(Vmete.de
volt:=volt+read_volt(Vmete.de
volt:=volt+read_volt(Vmete_red
volt:=volt+read_volt(Vmete_red
volt:=volt+read_volt(Vmete_red
volt:=volt+read_volt(Vmete_red
volt:=volt+read_volt(Vmete_red
volt:=volt+read_volt(Vmete_red
volt:=volt+read_volt(Vmete_red
volt=red_heate_red
volt=red_heater_red_volt(Vmete_volt));=red_heater_volt(Vmete_volt));</pre> | <pre>volt=0.0;
utstr' 'dumstr;
lode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
lode voltage conversion************************************</pre> | <pre>volt=0.0;
volt=0.0;
lode voltage conversion************************************</pre>
 | <pre>volts:=0.0;
voltscr:' 'dumstr;
nvert_Voltge conversion************************************</pre> | outstr' ' + dumstr;
ode voltage conversion************************************ | utstr*' * dumstr; volt=0.0;
lode voltage conversion************************************ | utstr' ' dumstr'
lode voltage conversion************************************
 |
| <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre> | <pre>statr+` '+dumstr; volt:=0.0; volt:=0.0;</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre> | <pre>outstr+ '+dumstr;
outstr+' '+dumstr;
lode volt@econversion************************************</pre> | <pre>outstr+ ' +dumstr; volt:=0.0; volt:=volt=vad_row (Tawater V, *dumstr2); volt:=volt=vad_row (Volt:=volt=vad_row (Vmete Volt:=volt=vad_row (Vmete Volt=V); volt=vad_row (Vmete Volt=V); volt=vad_row (Vmete Volt=V); volt=vad_row (Vmete Volt=V); volt=V); volt=vad_row (Vmete Volt=V); volt=V, *dumstr2; volt=V, *dums</pre> | <pre>outstr+' + dumstr;
outstr+' ' + dumstr;
fode voltage conversion************************************</pre> | <pre>outstr+ '+dumstr; volt:=0.0; volt:=0.0;</pre> | <pre>Dutstr+' '+dumstr;
Nutstr+' '+dumstr;
Nutstr+' Voltage conversion:
Dutstr+' '-dumstr2);
Nutstr+' '-dumstr2);</pre> | <pre>outstr+' '+dumstr; volt:=0.0;
ucde voltage conversion************************************</pre> | <pre>utstr+' '+dumstr; volt:=0.0;
ucde voltage conversion************************************</pre> | outstr+' '+dumstr; volt=0.0;
lode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************ | volt:=0.0;
voltage conversion************************************ |
| <pre>utstr+ (+dumstr;
ode voltage conversion************************************</pre> | <pre>start* '+dumstr; volt:=0.0; volt:(Vmet: utett=' '+dumstr2'; volt:=0.0; v</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre> | <pre>outstr+ '+dumstr; volt:=0.0;
idde voltage conversion************************************</pre> | <pre>outstr+ '+dumstr; volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+ '+dumstr; volt:=0.0; volt:=0.0;</pre> | <pre>butstr+' '+dumstr;
butstr+' '+dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' '+dumstr; volt:=0.0;
olde voltage conversion************************************</pre> | <pre>butstr+' '+dumstr;
ode voltage conversion************************************</pre> | outstr+' '+dumstr; volt:=0.0;
iode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
jode voltage conversion************************************ | outstr+ '+dumstr; volt:=0.0;
Jode voltage conversion#################################### |
| <pre>utetr+' +dumstr;
ode voltage conversion************************************</pre> | <pre>start* '+dumstr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' +dumstr;
ode voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv=0.0;
nvert_volt(setpt[index]);
y:6:2,dumstr2);
rean:=tem+read_temp(Tcontro
outstr+' '+dumstr2);
rean:=tem+read_temp(Tcontro
currimentemperst;
rean:=tem+read_volt(Nmeter
volt:=volt+read_volt(Nmeter.dev.
findex]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>untstr+' 'fdumstr;
ode voltage conversion************************************</pre> | outstr+' '+dumstr; volt=0.0;
lode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
jode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
iode voltage conversion************************************
 |
| <pre>volt:=0.0;
utetr+' +dumstr;
ode voltage conversion************************************</pre> | <pre>itstr+' 'dimatr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>outer++ ' +dumstr;
outer++ ' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>outetr+' +dumstr;
outetr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>untstr+' '+dumstr;
uode voltage conversion************************************</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv:=0.0;
onvert_Volt (setpt[index]);
rs:2,dumstr2);
rs:2,dumstr2);
rs:1,dumstr2);
rs:1:rs:1:dumstr2);
rs:1:rs:1:dumstr2);
rs:1:rs:1:dumstr2);
outstr+' '+dumstr2);
rs:1:rs:1:dumstr2);
outstr+' '+dumstr2);
findex]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utstr' 'dumstr;
utstr' 'dumstr;
nv:=0.0;
nv:=0.0;
utstr' 'dumstr2;
utstr' 'dumstr2;</pre> | outetr+' ' +dumstr; volt:=0.0;
iode voltage conversion************************************ | outstr+' ' +dumstr;
outstr+' ' +dumstr;
iode voltage conversion************************************ | outstr'' foundstry
dode voltage conversion************************************
 |
| <pre>utetr+' '+dimetr;
ode voltage conversion************************************</pre> | <pre>restry ' 'dumstry' ode voltage conversion************************************</pre> | <pre>utetr+' + dumatr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>outstr+' + dumstr;
outstr+' / dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+' + dumstr;
outstr+' + dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' + dumstr;
outstr+' + dumstr;
iode voltage conversion************************************</pre> | <pre>outstr:''toumstr;
outstr:''toumstr;
iode voltage conversion************************************</pre>
 | <pre>utstr' ' dumstr'
utstr' ' dumstr'
onert_volt(setpt[index]);
ris:2,dumstr2);
ris:2,dumstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);
ris::-utstr2);</pre> | <pre>outstr' ' +dumstr;
outstr' ' +dumstr;
nv:=0.0;
outstr' ' +dumstr2);
rs:2,dumstr2);
outstr' ' +dumstr2);
curr:read_curread_cenp(Tcontro)
curr:read_curread_cenp(Tcontro)
curr:read_curread_cent(Tweed)
outstr' ' +dumstr2);
outstr' ' +dumstr2);
curr:read_nv(nVmeter.dev.
fludex]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utst:+' ' dumst: '
utst:+' ' dumst: '
ode voltage conversion************************************</pre> | outstr+' 'tdumstr; volt=0.0;
iode voltage conversion************************************ | outstr:/ | utstr
 |
| <pre>utstr+' 'dumstr;
de voltage conversion************************************</pre> | <pre>start:' 'dumstr;
de voltage conversion************************************</pre> | <pre>utatr+' +dumatr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>coulds::+' + dumatr;
outst:+' + dumatr;
iods voltage conversion************************************</pre>
 | <pre>outstr:' 'dumstr;
outstr:' 'dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr:'' dumstr;
outstr:'' dumstr;
ioutstr:'' dumstr;
onvert Volt (setpt[index]);
y:6:2, dumstr2);
y:6:2,
dumstr2);
outstr:'' dumstr2;
outstr:'' dumstr2);
findex]:3:0, dumstr2);
moltsred_nv(nWeets: dev
end; [loop]
outstr:'' dumstr2);
power[point]:=read heater poi
outstr:''' dumstr2);</pre> | <pre>untertr' ' dumatr;
untertr' ' dumatr;
ode voltage conversion************************************</pre> | <pre>under:''''''''''''''''''''''''''''''''''''</pre>
 | <pre>vult:+/ :/ dumatr;
vult:+/ :/ dumatr;
ode voltage conversion************************************</pre> | <pre>outstr+' ' + dumatr;
outstr+' ' + dumatr;
iode voltage conversion************************************</pre> | outst:///dumst//
outst:///dumst//
iode voltage conversion************************************ | utstr///v/dumstr//
outstr// +dumstr/
ode voltage conversion************************************
 |
| <pre>utstr+' 'dumstr;
de voltage conversion************************************</pre> | <pre>start:' 'dumstr;
de voltage conversion************************************</pre> | <pre>utatr+' +dumatr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>coult:+'' + dumatr;
outstr+' ' + dumatr;
iode voltage conversion************************************</pre>
 | <pre>outstr:' 'dumstr;
outstr:' 'dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr:'' dumstr;
outstr:'' dumstr;
ioutstr:'' dumstr;
onvert Volt (setpt[index]);
y:6:2, dumstr2);
y:6:2,
dumstr2);
outstr:'' dumstr2;
outstr:'' dumstr2);
findex]:3:0, dumstr2);
moltsred_nv(nWeets: dev
end; [loop]
outstr:'' dumstr2);
power[point]:=read heater poi
outstr:''' dumstr2);</pre> | <pre>untertr' ' dumatr;
untertr' ' dumatr;
ode voltage conversion************************************</pre> | <pre>under:''''''''''''''''''''''''''''''''''''</pre>
 | <pre>vult:+/ :/ dumatr;
vult:+/ :/ dumatr;
ode voltage conversion************************************</pre> | outstr:/ ' + dumstr;
outstr:/ ' + dumstr;
iode voltage conversion************************************ | outst:///dumst//
outst:///dumst//
iode voltage conversion************************************ | utstr///v/dumstr//
outstr// +dumstr/
ode voltage conversion************************************
 |
| <pre>uter::::::::::::::::::::::::::::::::::::</pre> | <pre>de voltege converts;
itert'' + dumetr;
de voltge convertsion************************************</pre> | <pre>utatr+ '+dumatry
utatr+' +dumatry
ode voltege conversion************************************</pre> | <pre>utstr:/ 'dumstr;
utstr:/ 'dumstr;
utstr:/ 'dumstr;
utstr:/ 'dumstr);
nvert_volt(setpt[index]);
for loop:=1 to average_readi
for loop:=1 to average_readi
for</pre> | <pre>nucesi::::::::::::::::::::::::::::::::::::</pre> | <pre>idex1::';dumatr;
outer:':'dumatr;
iode voltage convertsion************************************</pre> | <pre>ndex_i:*;*;*dummer;;
outstr:* ' +dummer;;
iode voltage conversion************************************</pre> | <pre>ndex1:::::</pre> | <pre>read:r/:</pre> | <pre>des(:')'''''''''''''''''''''''''''''''''''</pre> | <pre>new1::</pre> | ndestriction destriction destr | nderi::: | newsi://s.cumatr//
outstr/* / tumatr/
jode voltage conversion#################################### |
| <pre>utstr+' 'dumstr;
de voltage conversion************************************</pre> | <pre>start:' 'dumstr;
de voltage conversion************************************</pre> | <pre>utatr+' +dumatr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>coult:+'' + dumatr;
outstr+' ' + dumatr;
iode voltage conversion************************************</pre>
 | <pre>outstr:' 'dumstr;
outstr:' 'dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr:'' dumstr;
outstr:'' dumstr;
ioutstr:'' dumstr;
onvert Volt (setpt[index]);
y:6:2, dumstr2);
y:6:2,
dumstr2);
outstr:'' dumstr2;
outstr:'' dumstr2);
findex]:3:0, dumstr2);
moltsred_nv(nWeets: dev
end; [loop]
outstr:'' dumstr2);
power[point]:=read heater poi
outstr:''' dumstr2);</pre> | <pre>untertr' ' dumatr;
untertr' ' dumatr;
ode voltage conversion************************************</pre> | <pre>under:''''''''''''''''''''''''''''''''''''</pre>
 | <pre>vult:+/ :/ dumatr;
vult:+/ :/ dumatr;
ode voltage conversion************************************</pre> | outstr:/ ' + dumstr;
outstr:/ ' + dumstr;
iode voltage conversion************************************ | outst:///dumst//
outst:///dumst//
iode voltage conversion************************************ | utstr///v/dumstr//
outstr// +dumstr/
ode voltage conversion************************************
 |
| <pre>utstr+' 'dumstr;
de voltage conversion************************************</pre> | <pre>start:' 'dumstr;
de voltage conversion************************************</pre> | <pre>utatr+' +dumatr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
iode voltage conversion************************************</pre>
 | <pre>outstr:' 'dumstr;
outstr:' 'dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr:'' dumstr;
outstr:'' dumstr;
ioutstr:'' dumstr;
onvert Volt (setpt[index]);
y:6:2, dumstr2);
y:6:2,
dumstr2);
outstr:'' dumstr2;
outstr:'' dumstr2);
findex]:3:0, dumstr2);
moltsred_nv(nWeets: dev
end; [loop]
outstr:'' dumstr2);
power[point]:=read heater poi
outstr:''' dumstr2);</pre> | <pre>untertr' ' dumatr;
untertr' ' dumatr;
ode voltage conversion************************************</pre> | <pre>under:''''''''''''''''''''''''''''''''''''</pre>
 | <pre>vult:+/ :/ dumatr;
vult:+/ :/ dumatr;
ode voltage conversion************************************</pre> | <pre>volt=0.0;
volt=0.0;
iode voltage conversion************************************</pre> | outst:///dumst//
outst:///dumst//
iode voltage conversion************************************ | utstr///v/dumstr//
outstr// +dumstr/
ode voltage conversion************************************
 |
| <pre>utstr+' 'dumstr;
de voltage conversion************************************</pre> | <pre>start:' 'dumstr;
de voltage conversion************************************</pre> | <pre>utatr+' +dumatr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>coutart+' +dumatr;
coutart+' +dumatr;
jode voltage conversion************************************</pre>
 | <pre>outstr:' 'dumstr;
outstr:' 'dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr:'' dumstr;
outstr:'' dumstr;
ioutstr:'' dumstr;
onvert Volt (setpt[index]);
y:6:2, dumstr2);
y:6:2,
dumstr2);
outstr:'' dumstr2;
outstr:'' dumstr2);
findex]:3:0, dumstr2);
moltsred_nv(nWeets: dev
end; [loop]
outstr:'' dumstr2);
power[point]:=read heater poi
outstr:''' dumstr2);</pre> | <pre>untertr' ' dumatr;
untertr' ' dumatr;
ode voltage conversion************************************</pre> | <pre>under:''''''''''''''''''''''''''''''''''''</pre>
 | <pre>vult:+/ :/ dumatr;
vult:+/ :/ dumatr;
ode voltage conversion************************************</pre> | <pre>volt=0.0;
volt=0.0;
iode voltage conversion************************************</pre> | outst:///dumst//
outst:///dumst//
iode voltage conversion************************************ | utstr///v/dumstr//
outstr// +dumstr/
ode voltage conversion************************************
 |
| <pre>utstr+' 'dumstr;
de voltage conversion************************************</pre> | <pre>start:' 'dumstr;
de voltage conversion************************************</pre> | <pre>utatr+' +dumatr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>coutart+' +dumatr;
coutart+' +dumatr;
jode voltage conversion************************************</pre>
 | <pre>outstr:' 'dumstr;
outstr:' 'dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr:'' dumstr;
outstr:'' dumstr;
ioutstr:'' dumstr;
onvert Volt (setpt[index]);
y:6:2, dumstr2);
y:6:2,
dumstr2);
outstr:'' dumstr2;
outstr:'' dumstr2);
findex]:3:0, dumstr2);
moltsred_nv(nWeets: dev
end; [loop]
outstr:'' dumstr2);
power[point]:=read heater poi
outstr:''' dumstr2);</pre> | <pre>untertr' ' dumatr;
untertr' ' dumatr;
ode voltage conversion************************************</pre> | <pre>under:''''''''''''''''''''''''''''''''''''</pre>
 | <pre>vult:+/ :/ dumatr;
vult:+/ :/ dumatr;
ode voltage conversion************************************</pre> | <pre>volt=0.0;
volt=0.0;
iode voltage conversion************************************</pre> | outst:///dumst//
outst:///dumst//
iode voltage conversion************************************ | utstr///v/dumstr//
outstr// +dumstr/
ode voltage conversion************************************
 |
| <pre>volt:=0.0;
utetr+' +dumstr;
ode voltage conversion************************************</pre> | <pre>itstr+' 'dimatr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>coutstr+' +dumatr;
outstr+' +dumatr;
jiode voltage conversion************************************</pre>
 | <pre>outetr+' +dumstr;
outetr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>untstr+' '+dumstr;
uode voltage conversion************************************</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv:=0.0;
onvert_Volt (setpt[index]);
rs:2,dumstr2);
rs:2,dumstr2);
rs:1,dumstr2);
rs:1:rs:1:dumstr2);
rs:1:rs:1:dumstr2);
rs:1:rs:1:dumstr2);
outstr+' '+dumstr2);
rs:1:rs:1:dumstr2);
outstr+' '+dumstr2);
findex]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utstr' 'dumstr;
utstr' 'dumstr;
nv:=0.0;
nv:=0.0;
utstr' 'dumstr2;
utstr' 'dumstr2;</pre> | <pre>coutstr+' +dumatr;
iode voltage conversion************************************</pre> | outstr+' ' +dumstr;
outstr+' ' +dumstr;
iode voltage conversion************************************ | outstr'' foundstry
dode voltage conversion************************************
 |
| <pre>utetr+' +dumstr;
ode voltage conversion************************************</pre> | <pre>start* '+dumstr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' +dumstr;
ode voltage conversion************************************</pre>
 | <pre>coutstr+ '+dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+' + dumstr;
outstr+' + dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv=0.0;
nvert_volt(setpt[index]);
y:6:2,dumstr2);
rean:=tem+read_temp(Tcontro
outstr+' '+dumstr2);
rean:=tem+read_temp(Tcontro
currimentemperst;
rean:=tem+read_volt(Nmeter
volt:=volt+read_volt(Nmeter.dev.
findex]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>untstr+' 'fdumstr;
ode voltage conversion************************************</pre> | <pre>coutstr+ '+dumstr; iiode voltage conversion************************************</pre> | outstr+' '+dumstr; volt:=0.0;
iode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
iode voltage conversion************************************
 |
| <pre>utstr+ ' +dumstr;
ode voltage conversion************************************</pre> | <pre>start* (+dumstr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>coutstr+ '+dumstr; volt:=0.0;
jiode voltage conversion************************************</pre>
 | <pre>outstr+ '+dumstr; volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' 'dumstr;
sovert_Volt(setpt[index]);
y:6:2,dumstr2);
y:6:2,dumstr2);
pedin
tem:=curr+read_temp(fcontro
tem:=curr+read_out(rent(in
volt:=volt+read_volt(Vmete
outstr+' 'dumstr2);
soutstr+' 'dumstr2);
[index]:4:1,dumstr2);
power[point]:=read_heate_point]:=read_heate_point]</pre> | <pre>outstr+ '+dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre>
 | <pre>Dutstr+' '+dumstr;
Dutstr+' '+dumstr)
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nvietr'' +dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' +dumstr2);
nvietr+' +dumstr2);</pre> | <pre>outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************</pre>
 | <pre>uutstr+' '+dumstr; volt:=0.0;
(ode voltage conversion************************************</pre> | <pre>coutstr+ ' +dumstr; volt:=0.0; iiode voltage conversion************************************</pre> | outstr+' '+dumstr; volt:=0.0;
jode voltage conversion************************************ | outstr+ '+dumstr; volt:=0.0;
iode voltage conversion####################################
 |
| <pre>utstr+ ' +dumstr;
ode voltage conversion************************************</pre> | <pre>start* (+dumstr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>coutstr+ '+dumstr; volt:=0.0;
idode voltage conversion************************************</pre>
 | <pre>outstr+ '+dumstr; volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' 'dumstr;
sovert_Volt(setpt[index]);
y:6:2,dumstr2);
y:6:2,dumstr2);
pedin
tem:=curr+read_temp(fcontro
tem:=curr+read_out(rent(in
volt:=volt+read_volt(Vmete
outstr+' 'dumstr2);
soutstr+' 'dumstr2);
[index]:4:1,dumstr2);
power[point]:=read_heate_point]:=read_heate_point]</pre> | <pre>outstr+ '+dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre>
 | <pre>Dutstr+' '+dumstr;
Dutstr+' '+dumstr)
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nvietr'' +dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' +dumstr2);
nvietr+' +dumstr2);</pre> | <pre>outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************</pre>
 | <pre>uutstr+' '+dumstr; volt:=0.0;
(ode voltage conversion************************************</pre> | <pre>coutstr+ ' +dumstr; volt:=0.0; iiode voltage conversion************************************</pre> | outstr+' '+dumstr; volt:=0.0;
jode voltage conversion************************************ | outstr+ '+dumstr; volt:=0.0;
iode voltage conversion####################################
 |
| <pre>utstr+ ' +dumstr;
ode voltage conversion************************************</pre> | <pre>start* (+dumstr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>outstr+ '+dumstr; volt:=0.0;
iode voltage conversion************************************</pre>
 | <pre>outstr+ '+dumstr; volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' 'dumstr;
sovert_Volt(setpt[index]);
y:6:2,dumstr2);
y:6:2,dumstr2);
pedin
tem:=curr+read_temp(fcontro
tem:=curr+read_out(rent(in
volt:=volt+read_volt(Vmete
outstr+' 'dumstr2);
soutstr+' 'dumstr2);
[index]:4:1,dumstr2);
power[point]:=read_heate_point]:=read_heate_point]</pre> | <pre>outstr+ '+dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre>
 | <pre>Dutstr+' '+dumstr;
Dutstr+' '+dumstr)
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nvietr'' +dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' +dumstr2);
nvietr+' +dumstr2);</pre> | <pre>outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************</pre>
 | <pre>uutstr+' '+dumstr; volt:=0.0;
(ode voltage conversion************************************</pre> | outstr+' '+dumstr; volt=0.0;
iode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
jode voltage conversion************************************ | outstr+ '+dumstr; volt:=0.0;
iode voltage conversion####################################
 |
| <pre>utstr+ ' +dumstr;
ode voltage conversion************************************</pre> | <pre>start* (+dumstr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>outstr+ '+dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+ '+dumstr; volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' 'dumstr;
sovert_Volt(setpt[index]);
y:6:2,dumstr2);
y:6:2,dumstr2);
pedin
tem:=curr+read_temp(fcontro
tem:=curr+read_out(rent(in
volt:=volt+read_volt(Vmete
outstr+' 'dumstr2);
soutstr+' 'dumstr2);
[index]:4:1,dumstr2);
power[point]:=read_heate_point]:=read_heate_point]</pre> | <pre>outstr+ '+dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre>
 | <pre>Dutstr+' '+dumstr;
Dutstr+' '+dumstr)
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nvietr'' +dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' +dumstr2);
nvietr+' +dumstr2);</pre> | <pre>outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************</pre>
 | <pre>uutstr+' '+dumstr; volt:=0.0;
(ode voltage conversion************************************</pre> | outstr+' '+dumstr; volt=0.0;
iode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
jode voltage conversion************************************ | outstr+ '+dumstr; volt:=0.0;
iode voltage conversion####################################
 |
<pre>utstr+ (+dumstr; ode voltage conversion************************************</pre>	<pre>start* '+dumstr; volt:=0.0; volt:(Vmet: utett=' '+dumstr2'; volt:=0.0; v</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>outstr' '+dumstr; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; outstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; volt:=0.0;</pre>	<pre>butstr+' '+dumstr; butstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; volt:=0.0; olde voltage conversion************************************</pre>	<pre>butstr+' '+dumstr; ode voltage conversion************************************</pre>	outstr+' '+dumstr; volt:=0.0; iode voltage conversion************************************	outstr+' '+dumstr; volt:=0.0; jode voltage conversion************************************	outstr+ '+dumstr; volt:=0.0; Jode voltage conversion####################################
<pre>utstr+ ' +dumstr; volt:=0.0; ode voltage conversion************************************</pre>	<pre>statr+` + dumstr; volt:=0.0; volt:=0.0;</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>coutatr+ '+dumatr; volt:=0.0; lode voltage conversion************************************</pre>	<pre>outstr' 'dumstr; volt=0.0; volt</pre>	<pre>outstr+' + dumstr; outstr+' / + dumstr; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; volt:=0.0;</pre>	<pre>butstr+ '+dumstr; butstr+' '+dumstr; clee voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; volt:=0.0; ude voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; volt:=0.0; lode voltage conversion************************************</pre>	<pre>coutstr+ ' +dumstr; volt:=0.0; iiode voltage conversion************************************</pre>	outstr+' '+dumstr; volt:=0.0; jode voltage conversion************************************	outstr+' '+dumstr; volt:=0.0; dode voltage conversion************************************
<pre>utstr+ ' +dumstr; volt:=0.0; ode voltage conversion************************************</pre>	<pre>statr+` + dumstr; volt:=0.0; volt:=0.0;</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>coutstr+ '+dumstr; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstr' 'dumstr; volt=0.0; volt</pre>	<pre>outstr+' + dumstr; outstr+' / + dumstr; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; volt:=0.0;</pre>	<pre>butstr+ '+dumstr; butstr+' '+dumstr; clee voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; volt:=0.0; ude voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; volt:=0.0; lode voltage conversion************************************</pre>	<pre>coutstr+ ' +dumstr; volt:=0.0; iiode voltage conversion************************************</pre>	outstr+' '+dumstr; volt:=0.0; jode voltage conversion************************************	outstr+' '+dumstr; volt:=0.0; dode voltage conversion************************************
<pre>utstr+ ' +dumstr; ode voltage conversion************************************</pre>	<pre>start* '+dumstr; volt:=0.0; start* '+dumstr; volt:=0.0; startvoltage conversion************************************</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>coutstr+ ' +dumstr; volt=0.0; lode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; outstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; volt:=0.0;</pre>	<pre>butstr+' '+dumstr; butstr+' '+dumstr; iode voltage conversion####################################</pre>	<pre>outstr+' '+dumstr; volt:=0.0; volt:=0.0</pre>	<pre>butstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>coutstr+ ' +dumstr; volt=0.0; iiode voltage conversion************************************</pre>	outstr+' '+dumstr; volt:=0.0; jode voltage conversion************************************	outstr+ '+dumstr; volt:=0.0; lode voltage conversion************************************
<pre>utstr+ (+dumstr; ode voltage conversion************************************</pre>	<pre>start* '+dumstr; volt:=0.0; volt:(Vmet: utett=' '+dumstr2'; volt:=0.0; v</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>coutstr+ ' +dumstr; volt=0.0; lode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; outstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; volt:=0.0;</pre>	<pre>butstr+' '+dumstr; butstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; volt:=0.0; olde voltage conversion************************************</pre>	<pre>butstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>coutstr+ ' +dumstr; volt=0.0; iiode voltage conversion************************************</pre>	outstr+' '+dumstr; volt:=0.0; jode voltage conversion************************************	outstr+ '+dumstr; volt:=0.0; Jode voltage conversion####################################
<pre>utstr+ (+dumstr; ode voltage conversion************************************</pre>	<pre>start* '+dumstr; volt:=0.0; volt:(Vmet: utett=' '+dumstr2'; volt:=0.0; v</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; outstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; volt:=0.0; volt:=0.0;</pre>	<pre>butstr+' '+dumstr; butstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; volt:=0.0; olde voltage conversion************************************</pre>	<pre>butstr+' '+dumstr; ode voltage conversion************************************</pre>	outstr+' '+dumstr; volt:=0.0; lode voltage conversion************************************	outstr+' '+dumstr; volt:=0.0; jode voltage conversion ¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹	outstr+ '+dumstr; volt:=0.0; Jode voltage conversion####################################
<pre>utstr+ ' +dumstr; ode voltage conversion************************************</pre>	<pre>start* (+dumstr; de voltage conversion************************************</pre>	<pre>utstr+ '+dumstr; ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; ode voltage conversion************************************</pre>										
 | <pre>outstr+ '+dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+ '+dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' 'dumstr;
sovert_Volt(setpt[index]);
y:6:2,dumstr2);
y:6:2,dumstr2);
pedin
tem:=curr+read_temp(fcontro
tem:=curr+read_out(rent(in
volt:=volt+read_volt(Vmete
outstr+' 'dumstr2);
soutstr+' 'dumstr2);
[index]:4:1,dumstr2);
power[point]:=read_heate_point]:=read_heate_point]</pre> | <pre>outstr+ '+dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre>
 | <pre>Dutstr+' '+dumstr;
Dutstr+' '+dumstr)
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nvietr'' +dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' '+dumstr2);
nvietr+' +dumstr2);
nvietr+' +dumstr2);</pre> | <pre>outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************</pre>
 | <pre>uutstr+' '+dumstr; volt:=0.0;
(ode voltage conversion************************************</pre> | outstr+' '+dumstr; volt=0.0;
iode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
jode voltage conversion************************************ | outstr+ '+dumstr; volt:=0.0;
iode voltage conversion####################################
 |
| <pre>utetr+' +dumstr;
ode voltage conversion************************************</pre> | <pre>start* '+dumstr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' +dumstr;
ode voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv=0.0;
nvert_volt(setpt[index]);
y:6:2,dumstr2);
rean:=tem+read_temp(Tcontro
outstr+' '+dumstr2);
rean:=tem+read_temp(Tcontro
currimentemperst;
rean:=tem+read_volt(Nmeter
volt:=volt+read_volt(Nmeter.dev.
findex]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>untstr+' 'fdumstr;
ode voltage conversion************************************</pre> | outstr+' '+dumstr; volt=0.0;
iode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************ | outstr+' '+dumstr; volt:=0.0;
iode voltage conversion************************************
 |
| <pre>utetr+' +dumstr;
ode voltage conversion************************************</pre> | <pre>start* '+dumstr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utstr+' +dumstr;
ode voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' '+dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
nvert_Volt (setpt[index]);
rist2,dumstr2);
outstr+' +dumstr2);
tem:=tem+read_temp(fcontro
curr:=curr+read_temp(fcontro
voltstr+' +dumstr2);
nvert=volt+read_volt(Vmete
outstr+' +dumstr2);
findex]:4:1,dumstr2);
power[point]:=read_heate_pc;</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
fode voltage conversion************************************</pre>
 | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv:=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);
nutstr+' '+dumstr2);</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv=0.0;
nvert_volt(setpt[index]);
y:6:2,dumstr2);
rean:=tem+read_temp(Tcontro
outstr+' '+dumstr2);
rean:=tem+read_temp(Tcontro
currimentemperst;
rean:=tem+read_volt(Nmeter
volt:=volt+read_volt(Nmeter.dev.
findex]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>untstr+' 'fdumstr;
ode voltage conversion************************************</pre> | <pre>outstr+' '+dumstr; volt=0.0; lode voltage conversion************************************</pre> | <pre>butstr+' ' +dumstr;
volt:=0.0;
fode voltage conversion************************************</pre> | outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************
 |
| <pre>volt:=0.0;
utetr+' '+dumstr;
ode voltage conversion************************************</pre> | <pre>statr+' + dumatr;
ode voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utetr+' +dumstr;
ode voltage conversion************************************</pre>
 | <pre>ubtstr+' +dumstr;
ucde voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' +dumstr;
nv=0.0;
nvert_volt(setpt[index]);
p:6:2,dumstr2);
putstr+' 'dumstr2;
nutstr+' 'dumstr2;
nutstr+' 'dumstr2;
nutstr+' 'dumstr2;
nutstr+' 'dumstr2;
outstr+' 'dumstr2;
findex]:4:1,dumstr2;
outstr+' 'dumstr2;
findex]:4:1,dumstr2;
gildex]:4:1,dumstr2;
nutstr+' 'dumstr2;</pre> | <pre>untstr+' +dumstr;
unde voltage conversion************************************</pre> | <pre>outstr+' '+dumstr;
outstr+' '+dumstr;
nv:=0.0;
preceversion************************************</pre>
 | <pre>uptstr+' '+dumstr;
uode voltage conversion************************************</pre> | <pre>outstr+' '+dumstr;
oude voltage conversion************************************</pre>
 | <pre>uutstr+`'idumstr;
uutstr+`'idumstr;
ode voltage conversion************************************</pre> | <pre>ultstr+' '+dumstr;
ude voltage conversion************************************</pre> | <pre>putstr+' '+dumstr;
putstr+' '+dumstr;
lode voltage conversion************************************</pre> | outstr+' '+dumstr; volt:=0.0;
lode voltage conversion************************************
 |
| <pre>volt:=0.0;
utetr+' +dumstr;
ode voltage conversion************************************</pre> | <pre>test:+' 'dimatr;
de voltage conversion************************************</pre> | <pre>utstr+ '+dumstr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' +dumstr;
jode voltage conversion************************************</pre>
 | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' + dimstr;
outstr+' + dimstr;
iode voltage conversion************************************</pre>
 | <pre>untstr+' '+dumstr;
uode voltage conversion************************************</pre> | <pre>outstr+' ' +dumstr;
oude voltage conversion************************************</pre>
 | <pre>utstr' 'dumstr;
utstr' 'dumstr;
nv:=0.0;
utstr' 'dumstr2);
utstr' 'dumstr2);
utstr' 'dumstr2;
utstr' 'dumstr2;
utstr' 'dumstr2;
utstr' 'dumstr2;
utstr' 'dumstr2;
utstr' 'dumstr2;
utstr' 'dumstr2;
utstr' 'dumstr2;
utstr' 'dumstr2;
utstr' 'dumstr2;</pre> | outstr+' '+dumstr; volt=0.0;
jode voltage conversion************************************ | outstr+' ' + dumstr;
outstr+' ' + dumstr;
oode voltage conversion************************************ | outstyf founder:
odde voltage conversion************************************
 |
| <pre>volt:=0.0;
utstr+' +dumstr;
ode voltage conversion************************************</pre> | <pre>volt:=0.0;
volt:=0.0;
de voltage conversion************************************</pre> | <pre>utet:+' + dumatr;
ode voltage conversion************************************</pre> | <pre>uttr+' 'dimatr;
ode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
iode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' ' dumstr;
outstr+' ' dumstr;
iode voltage conversion************************************</pre>
 | <pre>ultstr' ' +dumstr;
ulde voltage conversion************************************</pre> | <pre>outstr+' ' +dumstr;
outstr+' ' +dumstr;
nv:=0.0;
rist2,dumstr2);
rist2,dumstr2);
rist2,dumstr2);
rist2,dumstr2);
ristrintintintintintintintintintintintintinti</pre>
 | <pre>utstrf ' dumstr;
utstrf ' dumstr;
ode voltage conversion************************************</pre> | outstr+' 'tdumstr;
Jode voltage conversion************************************ | outstr.'' toumstr;
outstr.'' toumstr;
jode voltage conversion************************************ | utstr
 |
| <pre>volt:=0.0;
utetr+' 'dimetr;
ode voltage conversion************************************</pre> | <pre>restry ' 'dumstry' ode voltage conversion************************************</pre> | <pre>utetr+' + dumatr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
idde voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' ' dumstr;
outstr+' ' fumstr;
iode voltage conversion************************************</pre>
 | <pre>utstr' ' dumstr'
utstr' ' dumstr'
ode voltage conversion************************************</pre> | <pre>outstr' ' +dumstr;
outstr' ' +dumstr;
nv:=0.0;
outstr' ' +dumstr2);
rs:2,dumstr2);
outstr' ' +dumstr2);
curr:read_curread_cencf(ro
curr:read_curread_cencf(ro
curr:read_volt(read_volt(rometer);
outstr' ' +dumstr2);
curt:read_nv(nVmeter.dev;
[index]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utst:+' ' dumst:
utst:+' ' dumst:
ode voltage conversion************************************</pre> | <pre>outstr+' + dumstr;
outstr+' + dumstr;
iode voltage conversion************************************</pre> | outstr.'', tummstr;
outstr.'', tummstr;
jode voltage conversion************************************ | utstr./
 |
| <pre>volt:=0.0;
utetr+' 'dimetr;
ode voltage conversion************************************</pre> | <pre>restry ' 'dumstry' ode voltage conversion************************************</pre> | <pre>utetr+' + dumatr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
idde voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' ' dumstr;
outstr+' ' fumstr;
iode voltage conversion************************************</pre>
 | <pre>utstr' ' dumstr'
utstr' ' dumstr'
ode voltage conversion************************************</pre> | <pre>outstr' ' +dumstr;
outstr' ' +dumstr;
nv:=0.0;
outstr' ' +dumstr2);
rs:2,dumstr2);
outstr' ' +dumstr2);
curr:read_curread_cencf(ro
curr:read_curread_cencf(ro
curr:read_volt(read_volt(rometer);
outstr' ' +dumstr2);
curt:read_nv(nVmeter.dev;
[index]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utst:+' ' dumst:
utst:+' ' dumst:
ode voltage conversion************************************</pre> | outst+' ' + dumstr;
iode voltage conversion************************************ | outstr.'', tummstr;
outstr.'', tummstr;
jode voltage conversion************************************ | utstr./
 |
| <pre>volt:=0.0;
utetr+' 'dimetr;
ode voltage conversion************************************</pre> | <pre>restry ' 'dumstry' ode voltage conversion************************************</pre> | <pre>utetr+' + dumatr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>volts=0.0;
lode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' ' dumstr;
outstr+' ' fumstr;
iode voltage conversion************************************</pre>
 | <pre>utstr' ' dumstr'
utstr' ' dumstr'
ode voltage conversion************************************</pre> | <pre>outstr' ' +dumstr;
outstr' ' +dumstr;
nv:=0.0;
outstr' ' +dumstr2);
rs:2,dumstr2);
outstr' ' +dumstr2);
curr:read_curread_cencf(ro
curr:read_curread_cencf(ro
curr:read_volt(read_volt(rometer);
outstr' ' +dumstr2);
curt:read_nv(nVmeter.dev;
[index]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utst:+' ' dumst:
utst:+' ' dumst:
ode voltage conversion************************************</pre> | <pre>outstr+' + dumatr;
outstr+' + dumatr;
iode voltage conversion************************************</pre> | outstr.'', tummstr;
outstr.'', tummstr;
jode voltage conversion************************************ | utstr./
 |
| <pre>volt:=0.0;
utetr+' 'dimetr;
ode voltage conversion************************************</pre> | <pre>restry ' 'dumstry' ode voltage conversion************************************</pre> | <pre>utetr+' + dumatr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>volts=0.0;
lode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' ' dumstr;
outstr+' ' fumstr;
iode voltage conversion************************************</pre>
 | <pre>utstr' ' dumstr'
utstr' ' dumstr'
ode voltage conversion************************************</pre> | <pre>outstr' ' +dumstr;
outstr' ' +dumstr;
nv:=0.0;
outstr' ' +dumstr2);
rs:2,dumstr2);
outstr' ' +dumstr2);
curr:read_curread_cencf(ro
curr:read_curread_cencf(ro
curr:read_volt(read_volt(rometer);
outstr' ' +dumstr2);
curt:read_nv(nVmeter.dev;
[index]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utst:+' ' dumst:
utst:+' ' dumst:
ode voltage conversion************************************</pre> | <pre>outstr+' + dumatr;
outstr+' + dumatr;
iode voltage conversion************************************</pre> | outstr.'', tummstr;
outstr.'', tummstr;
jode voltage conversion************************************ | utstr./
 |
| <pre>volt:=0.0;
utetr+' 'dimetr;
ode voltage conversion************************************</pre> | <pre>restry ' 'dumstry' ode voltage conversion************************************</pre> | <pre>utetr+' + dumatr;
ode voltage conversion************************************</pre> | <pre>utetr+' '+dumstr;
ode voltage conversion************************************</pre>
 | <pre>volts:-0.0;
volts=0.0;
iode voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' ' dumstr;
outstr+' ' fumstr;
iode voltage conversion************************************</pre>
 | <pre>utstr' ' dumstr'
utstr' ' dumstr'
ode voltage conversion************************************</pre> | <pre>outstr' ' +dumstr;
outstr' ' +dumstr;
nv:=0.0;
outstr' ' +dumstr2);
rs:2,dumstr2);
outstr' ' +dumstr2);
curr:read_curread_cencf(ro
curr:read_curread_cencf(ro
curr:read_volt(read_volt(rometer);
outstr' ' +dumstr2);
curt:read_nv(nVmeter.dev;
[index]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utst:+' ' dumst:
utst:+' ' dumst:
ode voltage conversion************************************</pre> | <pre>outstr+' + dumatr;
outstr+' + dumatr;
iode voltage conversion************************************</pre> | outstr.'', tummstr;
outstr.'', tummstr;
jode voltage conversion************************************ | utstr./
 |
<pre>utstr+' 'ddmstr; utstr+' 'ddmstr; ode voltage conversion************************************</pre>	<pre>start:' 'dumstr; de voltage conversion************************************</pre>	<pre>utatr+' +dumatr; ode voltage conversion************************************</pre>	<pre>volt:=0.0; volt=0.0; ode voltage conversion************************************</pre>	<pre>volts:-0.0; iode voltage conversion************************************</pre>	<pre>outstr:' 'dumstr; outstr:' 'dumstr; jode voltage conversion************************************</pre>	<pre>outstr+' +dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre>	<pre>outstr:'' dumstr; outstr:'' dumstr; iode soltage conversion************************************</pre>	<pre>untertr' 'dumetr' untertr' 'dumetr' outert_Volt(setpt[index]); ris:2,dumetr2); rem:remtrad_temp(Tcontrol tem:remtrad_current(Ime volt:=volt=volt+read_volt(Nmeter.dev) nuterr' 'dumetr2); rentristion: remtrad_current(Ime volt:=volt=vead_volt(Nmeter.dev) end; [lobex]:4:1,dumetr2); end: [lobex]:4:1,dumetr2);</pre>	<pre>under:''''''''''''''''''''''''''''''''''''</pre>	<pre>volt:=0.0; vutstr+' 'dumstr; volt:=0.0; nvt=0.0; nvt=0.0; for loop:=1 to average_readin for loop:=1 to average_readin for</pre>	<pre>volt=0.0; volt=0.0; iode voltage conversion************************************</pre>	outst:///dumst// outst:///dumst// iode voltage conversion************************************	utstr///v/dumstr// outstr// +dumstr/ ode voltage conversion************************************
<pre>utstr+' 'ddmstr; dde voltage conversion************************************</pre>	<pre>start:' 'dumstr; de voltage conversion************************************</pre>	<pre>utatr+' +dumatr; ode voltage conversion************************************</pre>	<pre>volt:=0.0; volt=0.0; ode voltage conversion************************************</pre>	<pre>volts:-0.0; iode voltage conversion************************************</pre>	<pre>outstr:' 'dumstr; outstr:' 'dumstr; jode voltage conversion************************************</pre>	<pre>outstr+' + dumstr; outstr+' + dumstr; iode voltage conversion************************************</pre>	<pre>outstr:'' dumstr; outstr:'' dumstr; ioutstr:'' dumstr; onvert Volt (setpt[index]); y:6:2, dumstr2); y:6:2, dumstr2); outstr:'' dumstr2; outstr:'' dumstr2); inter:'' dumstr2); outstr:'' ' dumstr2); outstr:'' ' dumstr2); outstr:'''''''''''''''''''''''''''''''''''</pre>	<pre>untertr' 'dumetr' untertr' 'dumetr' outert_Volt(setpt[index]); ris:2,dumetr2); rem:remtrand_temp(Tcontrol tem:remtrand_temp(Tcontrol tem:read_temp(Tcontrol tem:read_volt(Ime volt:=volt=vead_volt(Wmeter outerr' 'dumetr2); nV:=nV:read_volt(Vmeter i(ndex]:3:0,dumetr2); end; [lop]</pre>	<pre>under:'' ' dumatr; unter:' ' fumatr; lode voltage conversion************************************</pre>	<pre>volt:=0.0; vutstr+' 'dumstr; volt:=0.0; nvt=0.0; nvt=0.0; for loop:=1 to average_readin for loop:=1 to average_readin for</pre>	<pre>volt=0.0; volt=0.0; iode voltage conversion************************************</pre>	outst:///dumst// outst:///dumst// iode voltage conversion************************************	utstr///v/dumstr// outstr// +dumstr/ ode voltage conversion************************************
<pre>utstr+' 'ddmstr; utstr+' 'ddmstr; ode voltage conversion************************************</pre>	<pre>start:' 'dumstr; de voltage conversion************************************</pre>	<pre>utatr+' +dumatr; ode voltage conversion************************************</pre>	<pre>volt:=0.0; volt=0.0; ode voltage conversion************************************</pre>	<pre>volts:-0.0; volts:-0.0; idode voltage conversion************************************</pre>	<pre>outstr:' 'dumstr; outstr:' 'dumstr; jode voltage conversion************************************</pre>	<pre>outstr+' +dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre>	<pre>outstr:'' dumstr; outstr:'' dumstr; iode soltage conversion************************************</pre>	<pre>untertr' 'dumetr' untertr' 'dumetr' outert_Volt(setpt[index]); ris:2,dumetr2); rem:remtrad_temp(Tcontrol tem:remtrad_current(Ime volt:=volt=volt+read_volt(Nmeter.dev) nuterr' 'dumetr2); rentristion: remtrad_current(Ime volt:=volt=vead_volt(Nmeter.dev) end; [lobex]:4:1,dumetr2); end: [lobex]:4:1,dumetr2);</pre>	<pre>under:''''''''''''''''''''''''''''''''''''</pre>	<pre>volt:=0.0; vutstr+' 'dumstr; volt:=0.0; nvt=0.0; nvt=0.0; for loop:=1 to average_readin for loop:=1 to average_readin for</pre>	<pre>volt=0.0; volt=0.0; iode voltage conversion************************************</pre>	outst:///dumst// outst:///dumst// iode voltage conversion************************************	utstr///v/dumstr// outstr// +dumstr/ ode voltage conversion************************************
<pre>volt:=0.0; utetr+' 'dimetr; ode voltage conversion************************************</pre>	<pre>restry ' 'dumstry' ode voltage conversion************************************</pre>	<pre>utetr+' + dumatr; ode voltage conversion************************************</pre>	<pre>utetr+' '+dumstr; ode voltage conversion************************************</pre>										
 | <pre>volt:=0.0;
volt:=0.0;
idde voltage conversion************************************</pre>
 | <pre>volt:=0.0;
volt:=0.0;
iode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr+' ' dumstr;
outstr+' ' fumstr;
iode voltage conversion************************************</pre>
 | <pre>utstr' ' dumstr'
utstr' ' dumstr'
ode voltage conversion************************************</pre> | <pre>outstr' ' +dumstr;
outstr' ' +dumstr;
nv:=0.0;
outstr' ' +dumstr2);
rs:2,dumstr2);
outstr' ' +dumstr2);
curr:read_curread_cencf(ro
curr:read_curread_cencf(ro
curr:read_volt(read_volt(rometer);
outstr' ' +dumstr2);
curt:read_nv(nVmeter.dev;
[index]:4:1,dumstr2);
end; [loop]</pre>
 | <pre>utst:+' ' dumst:
utst:+' ' dumst:
ode voltage conversion************************************</pre> | <pre>outstr+' + dumstr;
outstr+' + dumstr;
iode voltage conversion************************************</pre> | outstr.'', tummstr;
outstr.'', tummstr;
jode voltage conversion************************************ | utstr./
 |
| <pre>utstr+' 'ddmstr;
utstr+' 'ddmstr;
ode voltage conversion************************************</pre> | <pre>start:' 'dumstr;
de voltage conversion************************************</pre> | <pre>utatr+' +dumatr;
ode voltage conversion************************************</pre> | <pre>volt:=0.0;
volt=0.0;
ode voltage conversion************************************</pre> | <pre>outstr:''dumstr;
outstr:''dumstr;
outstr:''dumstr;
souvert_Volt(setpt[index]);
y:6:2,dumstr2);
outstr:''dumstr2);
outstr:''dumstr2);
outstr:''dumstr2);
t[index]:3:0,dumstr2);
outstr:''dumstr2);
findex]:4:1,dumstr2);
outstr:''dumstr2);
outstr:''dumstr2);
findex]:4:1,dumstr2);
gain[point]:=read_gain[Tont</pre> | <pre>outstr+' 'dumstr;
outstr+' 'dumstr;
jode voltage conversion************************************</pre> | <pre>outstr+' +dumstr;
outstr+' +dumstr;
iode voltage conversion************************************</pre> | <pre>outstr:'' dumstr;
outstr:'' dumstr;
iode soltage conversion************************************</pre> | <pre>untertr' 'dumetr'
untertr' 'dumetr'
outert_Volt(setpt[index]);
ris:2,dumetr2);
rem:remtrad_temp(Tcontrol
tem:remtrad_current(Ime
volt:=volt=volt+read_volt(Nmeter.dev)
nuterr' 'dumetr2);
rentristion:
remtrad_current(Ime
volt:=volt=vead_volt(Nmeter.dev)
end; [lobex]:4:1,dumetr2);
end: [lobex]:4:1,dumetr2);</pre> | <pre>under:''''''''''''''''''''''''''''''''''''</pre> | <pre>volt:=0.0;
vutstr+' 'dumstr;
volt:=0.0;
nvt=0.0;
nvt=0.0;
for loop:=1 to average_readin
for loop:=1 to average_readin
for</pre> | utstr*' *dumstr?
dode voltage conversion************************************ | outstriverounder)
outstriverounder)
dode voltage conversion#################################### | utstr///v/dumstr//
outstr// +dumstr/
ode voltage conversion************************************ |
| <pre>utstr+' + dumstr;
utstr+' + dumstr;
ode voltage conversion************************************</pre> | <pre>de voltes conversion************************************</pre> | <pre>utatr+ '+dumatr;
utatr+ '+dumatr;
ode voltage conversion************************************</pre> | <pre>user:: ' +dumstr;
ode voltage conversion************************************</pre>
 | <pre>inter()''''''''''''''''''''''''''''''''''''</pre>
 | <pre>interfit' 'dumetr'
outetr'' 'dumetr2);
nvert_Volt(setpt[index]);
fs12,dumetr2);
fs12,dumetr2);
fs12,dumetr2);
nutetr+' 'dumetr2);
nutetr+' 'dumetr2);
outetr+' 'dumetr2);
findex]:3:0,dumetr2);
outetr+' 'dumetr2);
findex]:4:1,dumetr2);
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,dumetr2;
findex]:4:1,d</pre> | <pre>under:'' 'dumatry'
under:'' 'dumatry'
ode voltage conversion************************************</pre> | <pre>interfit://idumetr/
outstr//idumetr/
convert_voltage.conversion************************************</pre>
 | <pre>untart:' 'dumatr'
untart'' 'dumatr'
ode voltage conversion************************************</pre> | <pre>interfit.outstrf' 'dumatr'
uutstrf' 'dumatr'
outstrf' 'dumatr'
prest_dec conversion************************************</pre>
 | <pre>interf://interfice/operatory/
outstr://interfice/operatory/
oode voltage conversion************************************</pre> | <pre>destr=""""""""""""""""""""""""""""""""""""</pre> | <pre>dest:://dumatr/
outst://tdumatr/
outst://tdumatr/
pvoltectpt[index])/
for loop:=1 to average_read
/:6:2,dumatr2)/
ten:=tem+read_temp(Tcontre
outst://tdumatr2/</pre> | <pre>deal::::::::::::::::::::::::::::::::::::</pre>
 |
| <pre>uter::::::::::::::::::::::::::::::::::::</pre> | <pre>de voltes converts;
itert'' + dumetr;
obe voltege converts
nvert Volt (metpt[index]);
volt = 0.0;
nvert Volt (metpt[index]);
itert' ' + dumetr2);
itert' ' + dumetr2);
index]: 41, dumetr2);
power [point]: = read_gain(Tcont
index]: 41, dumetr2);
index]: 41, dumetr2);
power [point]: = read_gain(Tcont);
index]: 41, dumetr2);</pre> | <pre>utatr+ '+dumatry
utatr+' +dumatry
ode voltege conversion************************************</pre> | <pre>utstr:+' 'dumstr;
utstr:+' 'dumstr;
utstr:+' 'dumstr;
utstr:+'' dumstr]);
nvert_volt(setpt[index]);
for loop:=1 to average_readi
is:2,dumstr2);
utstr:+' 'dumstr2);
utstr:+'' +dumstr2);
utstr:+'' +dumstr2);
utstr:+''' +dumstr2);
utstr:+''''''''''''''''''''''''''''''''''''</pre> | <pre>des():*:, 'dumatr;
uutstr:':, 'dumatr;
ode voltage conversion************************************</pre> | <pre>utstr+ ' +dumstr) utstr+ ' +dumstr) utstr+ ' +dumstr) utstr+ ' +dumstr) utestr+ ' +dumstr2) nvert_volt(setpt[index]); for loop:=l to average_read; for loop:=l to average, for loop:=l to avera</pre> | <pre>utstr+' '+dumatr;
volt:=0.0;
volt:=0.0;
volt:=0.0;
invert_volt(setpt[index]);
r:6:2,dumatr2);
r:e:2,dumatr2);
vutstr+' '+dumatr2);
vutstr+' +dumatr2);
vutstr+' +dumatr2);
vutstr+' +dumatr2);
vutstr+' '+dumatr2);
vutstr+' '+dumatr2);
vutstr+' '+dumatr2);</pre> | <pre>user::::::::::::::::::::::::::::::::::::</pre> | <pre>utstr:+ ' dumatr);
utstr:+ ' dumatr);
ode voltage conversion************************************</pre> | <pre>newsist</pre> | <pre>utstr:/ 'dumatr;
uutstr:/ 'dumatr;
ode voltage conversion************************************</pre> | <pre>udest1'::</pre> | ucer1:::: | utstr://dumstr/
utstr://tdumstr:
ode voltage conversion#################################### |
| <pre>dest::rs.dumatr);
uteatr+' + dumatr;
ods voltage conversion************************************</pre> | <pre>de voltesconterry
lest+' ' +dumetry
de voltesconversion************************************</pre> | <pre>dex(:):*,dumarr);
utatr:*,dumarr);
ode voltege converion:************************************</pre> | <pre>def:::::::::::::::::::::::::::::::::::</pre>
 | <pre>ndex(::':,dumstr);
outstr':'.dumstr);
dode voltage conversion************************************</pre>
 | <pre>ndex(::':,.dumetr);
outst:-'' +dumetr);
outst:-'' +dumetr);
wolt:=0.0;
y:6:2,dumetr2);
y:6:2,dumetr2);
uutst:+''+dumetr2);
outst:+''+dumetr2);
index]:3:0,dumetr2);
outst:+''+dumetr2);
findex]:4:1,dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);
outst:+''+dumetr2);</pre> | <pre>ndex(:'':,dumstr);
outstr'':dumstr);
for log:=1 to average_read;
y:6:2,dumstr2);
y:6:2,dumstr2);
outstr+''+dumstr2);
outstr+''+dumstr2);
outstr+''+dumstr2);
findex1:41,dumstr2);
outstr+''+dumstr2);
outstr+''+dumstr2);
outstr+''+dumstr2);
findex1:41,dumstr2);
outstr+''+dumstr2);</pre> | <pre>ndexj:':a,dumstr);
outstr' 'dumstr);
outstr'' 'dumstr);
for loop:=1 to average_readi
pren:
temtread_temp(Tcontro);
y:6:2,dumstr2);
y:6:2,dumstr2);
tems: temtread_temp(Tcontro);
curr:=curr:read_temp(Tcontro);
curr:=curr:read_rolt(Vmete:
nv:=nv:read_rolt(Vmete:
nv:=nv:read_rolt(vmeter.dev.
end; [loop];
outstr'' 'dumstr2);
power[point]:=read heater poi
outstr'' 'dumstr2);</pre> | <pre>ndex(:::s,dumstr);
outstr+' '+dumstr);
ode voltage conversion************************************</pre> | <pre>dex(:'?', dumstr);
outstr'' 'dumstr);
robits=Conversion************************************</pre>
 | <pre>ndex():7:, dumatr);
utstr+' ' dumatr;
ode voltage conversion************************************</pre> | ncexi:':*,dumstr);
outstr' '+dumstr;
ode voltage conversion************************************ | ndexj://s.gummetr/?
outstr/ 'dummetr; volte=0.0;
outstr/ voltage coversion************************************ | ndexi:/?4.dumstr/?
outstr+ (+dumstr;
jode voltege conversion************************************
 |
<pre>dex(:/:4.dumstr); utstr+' + dumstr); ods voltege conversion************************************</pre>	<pre>der:::/.t.dumetr; der voltage conversion************************************</pre>	<pre>dex!:/:4.dumstr); utstr:^uv; ode voltage conversion************************************</pre>	<pre>dex1:/f4.dumstr); utstr+' 'fdumstr; utstr+' 'fdumstr; nvert Volt (etpt[index]); rvert Volt (etpt[index]); nvert Volt (etpt[index]); nvert vit (etpt[index]); nvert vit (etpt[index]); nvert vit (etpt[index]); vutetr+' 'fdumstr2); utetr+' 'fdumstr2);</pre>	<pre>ndex[:':4,dumstr); outstr'' +dumstr); outstr'' +dumstr); y:6:2,dumstr2); y:6:2,dumstr2); poutstr'' +dumstr2); cutri=cutriesd_temp(Tcontro cutri=cutriesd_temp(Tcontro cutri=cutriesd_volt(Ymete nv::numstr2); findex]:4:1,dumstr2); gain[point]:=read_gain[Tcont gain[point]:=read_gain[Tcont power[point]:=read_gain[Tcont power[point]:=read_gain[Tcont power[point]:=read_gain[Tcont power[point]:=read_gain[Tcont power[point]:=read_gain[Tcont power[point]:=read_gain[Tcont power[point]:=read_gain[Tcont power[point]:=read_gain[Tcont</pre>	<pre>ndex[:':4,dumstr); outstr'':4,dumstr); outstr'' +dumstr); got loop:=1 to average_readi y:6:2,dumstr2); tem:=tem+read_temp(Tcontro y:6:2,dumstr2); outstr+' +dumstr2); index]:3:0,dumstr2); outstr+' '+dumstr2); outstr+' '+dumstr2); findex]:4:1,dumstr2); outstr+' '+dumstr2); findex]:4:1,dumstr2); outstr+' '+dumstr2); goutstr+' '+dumstr2); findex]:4:1,dumstr2); goutstr+' '+dumstr2); goutstr+' '+dumstr2); goutstr+' '+dumstr2); goutstr+' '+dumstr2); goutstr+' '+dumstr2); goutstr+' '+dumstr2); goutstr+' '+dumstr2); goutstr+' '+dumstr2); goutstr+' '+dumstr2);</pre>	<pre>ndex[:/:4,dumstr); outstr*/ *dumstr); idde voltage conversion************************************</pre>	<pre>ndex[:':4, dumstr); outstr'' ' dumstr); outstr'' ' dumstr); outstr'' ' dumstr2); outstr'' ' dumstr2); for loop:=1 to average_readi begin tem:=temtread_temp(Tcontro tem:=temtread_temp(Tcontro tem:=temtread_temp(Tcontro tem:=temtread_out(Imtei volt:=toltr=', ' dumstr2); findex]:3:0, dumstr2); findex]:4:1, dumstr2); outstr+' ' dumstr2);</pre>	<pre>ndex(:':4,dumstr); utstr:':4,dumstr); lode voltage conversion************************************</pre>	<pre>dex[:'74,dumstr); outstr'' 'dumstr); convert_voltage.comversion####################################</pre>	<pre>dexj::4,dumstr); uutstr+''+dumstr; code voltage conversion************************************</pre>	ndex[:/:4.dumstr); volt:=0.0; outstr/ '+dumstr; volt:=0.0; otde voltege convertion************************************	<pre>ndex[:/:4.dumstr); outstr+ ('tdumstr); volt:=0.0; voltage conversion************************************</pre>	ndex[:/:4.dumstr); curri-v.v; outstr+ (*dumstr; iode voltage conversion************************************
<pre>dex1:7:4,dumstr); utstr+' +dumstr); utstr+' +dumstr); nv=t_0.0; nv=t_volt(eetpt[index]); s:s.2,dumstr2); utstr+' +dumstr2); utstr+' +dumstr2);</pre>	<pre>dex[:7:4,dumstr); lest+' '+dumstr; odt=0.0; det Voltgeconversion************************************</pre>	<pre>dex!:7:4,dumatr); utstr: '.tdumatr); ode voltage conversion************************************</pre>	<pre>dex1:714.dumstr); utstr+ '+dumstr); utstr+ '+dumstr); volt.=0.0; nver Voltge conversion************************************</pre>	<pre>ndex[:7:4,dumstr]; outstr+' +dumstr]; iode voltage conversion************************************</pre>	<pre>ndex[:7:4,dumstr]; outstr+''+dumstr]; outstr+''+dumstr2; outstr+''+dumstr2; for loop:=1 to average_read; y:6:2,dumstr2); for loop:=1 to average_read; pegin findex]:3:0,dumstr2); findex]:3:0,dumstr2); findex]:4:1,dumstr2); findex]:4:1,dum</pre>	<pre>ndex1:7:4, dumstr); outstr' ' +dumstr); outstr' ' +dumstr2); y:6:2, dumstr2); putstr' ' +dumstr2); y:6:2, dumstr2); outstr' ' +dumstr2); outstr' ' +dumstr2); findex1:4.1, dumstr2); outstr' ' +dumstr2); outstr' ' +dumstr2); outstr' ' +dumstr2);</pre>	<pre>ndex]:7:4, dumstr; uoutstr; ' + dumstr; iode voltage conversion************************************</pre>	<pre>idex[:1:4,dumstr]; ultstr*('+dumstr); ultstr*('+dumstr2); ide voltge conversion************************************</pre>	<pre>dex[:7:4,dumstr]; utstr=u.u; yutstr=u.u; yutstr=u.u; y:6:2,dumstr2); y:6:2,dumstr2); utstr+' 'dumstr2); utstr+' 'dumstr2); utstr+'dumstr2); utstr+'dumstr2); utstr+'dumstr2); utstr+'dumstr2); utstr+'dumstr2); utstr+'dumstr2);</pre>	<pre>idex]:?s/dumstr); utstr:''+dumstr); ode voltage conversion************************************</pre>	<pre>ndex1:1:4,dumstr); outr=0.0; outetr' ' +dumstr; outetr' ' +dumstr; nv=1:eout=0.0; iode voltage conversion####################################</pre>	<pre>ndex1:7.4, dumstr); outstr+' ' + dumstr; outste conversion************************************</pre>	ndex]:7:4,dumstr); curri=U.U; outstr+' +dumstr; jode voltage conversion************************************
<pre>dex]:7:4,dumstr; utstr+' +dumstr; ode voltage conversion************************************</pre>	<pre>dex]:7:4,dumstr); tetr:=0.0; tetr:=0.0; tetr:* '+dumstr; ode volt==0.0; volt==0.0; volt==0.0; volt==0.0; volt==0.0; volt==0.0; tem:read_temp(fontr nvert_volt(expt[index]); tem:read_temp(fontr volt==0.1; volt==0.0; volt=0.0; volt=0.</pre>	<pre>dex]:?:4,dumatr); utetr+' +dumatr); utetr+' +dumatr) ode volt=0.0; nv=t_voltge conversion************************************</pre>	<pre>dex]:7:4,dumstr); uttr:=0.0; ode voltage conversion************************************</pre>	<pre>ndex]:?:4,dumstr; curr:=0.0; outstr+' '+dumstr; n0.0; iode voltage conversion************************************</pre>	<pre>ndex]:?:4,dumstr; outstr+' 'dumstr; outstr+' 'dumstr; iode voltage conversion************************************</pre>	<pre>ndex]:7:4,dumstr); outstr+' 'dumstr); outstr+' 'dumstr; outstr+' 'dumstr2; sovert_Volt(setpt[index]); sovert_Volt(setpt[inde</pre>	<pre>ndex]:7:4,dumstr); ndex]:7:4,dumstr); outstr+' 'dumstr2; outstr+' 'dumstr2); outstr+' 'dumstr2);</pre>	<pre>idex[:7:4,dumstr]; udex[:7:4,dumstr]; uutstr* '+dumstr; lode voltage conversion************************************</pre>	<pre>ndex]:7:4,dumstr; outstr*' *dumstr; utstr*' *dumstr; nv=0.0; for loop:=1 to average_readin for loop:=1 to average_readin</pre>	<pre>idex]:7:4,dumstr); idex]:7:4,dumstr); utstr+' '+dumstr; utstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>ndex]:7:4,dumstr); outstr+' '+dumstr; outstr+' '+dumstr; iode voltage conversion************************************</pre>	<pre>ndex]:?:4,dumstr); outstr+' ' +dumstr; iode voltage conversion************************************</pre>	<pre>ndex]:7:4,dumstr; outstr' ' +dumstr; outstr' ' +dumstr; oode voltage conversion************************************</pre>
<pre>dex]:7:4,dumstr; utstr+' +dumstr; ode voltage conversion************************************</pre>	<pre>dex]:7:4,dumstr; tstr*' *dumstr; ode volte=0.0; wolt=0.0; wort_Volt(seconversion************************************</pre>	<pre>dex]:7:4,dumatr); utetr+' +dumatr; utetr+' +dumatr; ode volt=0.0; nver_volt(setpt[index]); i.62,dumatr2); utetr+' +dumatr2); utetr+' +dumatr2);</pre>	<pre>dex]:7:4,dumstr); uttr:+' +dumstr; ode voltage conversion************************************</pre>	<pre>ndex]:?:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; iode voltege conversion************************************</pre>	<pre>ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; iode voltege conversion************************************</pre>	<pre>ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre>	<pre>ndex]:7:4,dumstr; outstr+' 'dumstr; outstr+' 'dumstr; onvert_Volt(setp[index]); sient_entextrain y:s:c_dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2);</pre>	<pre>idex]:7:4,dumstr); udex]:7:4,dumstr; uutstr+' '+dumstr; nvert_Voltage conversion************************************</pre>	<pre>ndex]:7:4,dumstr; udex]:7:4,dumstr; utstr+' '+dumstr; note voltage conversion************************************</pre>	<pre>idex]:7:4,dumstr); idex]:7:4,dumstr); utstr+' '+dumstr; utstr+' '+dumstr); iode voltage conversion************************************</pre>	<pre>ndex]:1:4,dumstr); outstr+` '+dumstr; outstr+` '+dumstr; volt:=0.0; iode voltage conversion************************************</pre>	<pre>ndex]:?:4.dumstr); curr:=0.0; outstr+' ' +dumstr; outstr+' ' +dumstr; outstr=0.0; iode voltage conversion************************************</pre>	<pre>ndex]:7:4,dumstr); curr:=0.0; outstr' '+dumstr; outstr' '+dumstr; outstr' '+dumstr; outstr' '+dumstr; nY:=0.0; iode voltage conversion************************************</pre>
<pre>dax]:7:4,dumstr; utstr+' +dumstr; ode voltage conversion************************************</pre>	<pre>dex1:7:4,dumstr; statr+` +dumstr; odt:=0.0; statr+` +dumstr; odt:=0.0; statr+` +dumstr2; statr+` +dumstr2; itstr+` +dumstr2; itstr+` +dumstr2; index1:3:0,dumstr2; index1:4:1,dumstr2; index1:4:1,dumstr2; index1:4:1,dumstr2; index1:4:1,dumstr2; index1:4:1,dumstr2; index1:4:1,dumstr2; index1:4:1,dumstr2;</pre>	<pre>dex::7:4, dumstr); utstr+' '+dumstr); ode voltage conversion************************************</pre>	<pre>dex!:7:4,dumstr; utstr+' +dumstr; ode voltage conversion************************************</pre>	<pre>dex1:7:4,dumstr; dex1:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; source conversion************************************</pre>	<pre>idex]:7:4,dumstr; idex]:7:4,dumstr; outstr+ '+dumstr; ide voltage conversion************************************</pre>	<pre>idex]:7:4,dumstr); idex]:7:4,dumstr); outstr+' +dumstr; outstr+' +dumstr2); iode voltage conversion************************************</pre>	<pre>dex1:7:4,dumstr); dex1:7:4,dumstr); outstr+' +dumstr; outstr+' +dumstr2; ioofe voltage conversion************************************</pre>	<pre>dex]:7:4,dumstr; dex]:7:4,dumstr; butstr+``+dumstr; butstr+``+dumstr; nv:=0.0; nv:=0.0; nv:=1:0:0; for loop:=1 to average_readir for loop:=1 to average</pre>	<pre>dex]:7:4,dumstr; dex]:7:4,dumstr; butstr+ '+dumstr; tode voltage conversion************************************</pre>	<pre>dex]:7:4.dumstr); udex]:7:4.dumstr); udet+' '+dumstr); nv:=0.0; nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); for loop:=1 to average_readin for loop:=1 to average_readin findex]:4:1,dumstr2); end; [loop]</pre>	<pre>dex]:7:4,dumstr; outstr+' '+dumstr; outstr+' '+dumstr; fode voltage conversion************************************</pre>	<pre>dex[:7:4,dumstr); outstr+' '+dumstr; outstr+' '+dumstr; outstr+' '+dumstr; yole voltage conversion************************************</pre>	<pre>idex1:7:4.dumstr; idex1:7:4.dumstr; outstr+</pre>
<pre>dex]:7:4,dumstr; dex]:7:4,dumstr; utstr* '+dumstr; ode voltage conversion************************************</pre>	<pre>dex:[7:4, dumatr; test:+ ' +dumatr; de voltege conversion************************************</pre>	<pre>dex]:7:4,dumstr; dex]:7:4,dumstr; ode voltage conversion************************************</pre>	<pre>utstr:/ dumstr:/ utstr:/ dumstr:/ utstr:/ dumstr:/ ode voltage conversion************************************</pre>										
 | <pre>outs::: dumatr:;
ndex]::.dumatr:;
outst:+' 'dumatr:;
outst:+' 'dumatr:
iode voltage conversion************************************</pre>
 | <pre>outer: ' 'dumatr);
ndex]:?.f.dumatr);
outer: ' 'tdumatr);
outer: ' 'tdumatr);
outer: Yoit (mote
viet: dumatr2);
read: [index]:3:0,dumatr2);
cutatr: ' 'dumatr2);
cutatr: ' 'dumatr2);
cutatr: ' 'dumatr2);
outer: ' 'dumatr2);
cutatr: ' 'dumatr2);
cutatr: ' 'dumatr2);
cutatr: ' 'dumatr2);
cutatr: ' 'dumatr2);
cutatr: ' 'dumatr2);
cutatr: ' dumatr2);
cutatr: ' dumatr2);</pre> | <pre>dex1:?., dumstr);
ndex1:?., dumstr);
outstr+' 'dumstr);
outstr+' 'dumstr);
y:0:2, dumstr2);
y:0:2, dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
findex1:3:0, dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);</pre> | <pre>dex[:?.4,dumstr]
ndex[:?.4,dumstr]
outstr:* 'fdumstr]
sutstr:=0.0;
ide voltage conversion************************************</pre>
 | <pre>dex[:7:4,dumstr);
dex[:7:4,dumstr);
untstr*('4dumstr);
untstr*('4dumstr);
nv=0.0;
nvert_volt(setpt[index]);
prist_dumstr2);
rist_dumstr2);
rist_dumstr2);
rist_dumstr2);
nutstr*('4dumstr2);
untstr*('4dumstr2);
interr*(*4dumstr2);
nutstr*('4dumstr2);
interr*(*4dumstr2);
interr*(*4dumstr2);
interr*(*4dumstr2);
interr*(*4dumstr2);
interr*(*4dumstr2);
interr*(*4dumstr2);
interr*(*4dumstr2);
interr*(*4dumstr2);</pre> | <pre>dex17:4,dumstr;
dex17:4,dumstr;
outstr*' *dumstr;
nv:=0.0;
volt:=0.0;
nv:=0.0;
nv:=2.0;
res:=diumstr2);
res:_dumstr2);
res:=curread_temp(Tcontro
tem:=tem+read_temp(Tcontro
tem:=tem+read_temp(Tcontro
tem:=tem+read_temp(Tcontro
tem:=tem+read_volt(Nmeter.dev
volt:=volt+read_volt(Nmeter.dev
nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=</pre> |
<pre>dex1:7:4,dumstr;
dex1:7:4,dumstr;
uutstr* '*dumstr;
ode voltage conversion************************************</pre> | outer: ' tdumstr);
outer: ' tdumstr);
outer: ' tdumstr);
oode voltage conversion************************************ | outstry roumstry
outstry' fumstry;
outstry' fumstry;
outstry' fumstry;
outstry' fumstry;
onvert_Volt (setpt[index]);
y:6:2,dumstr2);
outstr+' ' fumstr2;
tem:=tem+read_temp(Tcontre | <pre>outsity voumstry
adex1:7:4,dumstr);
outsity '+dumstry;
utsity dee voltage conversion************************************</pre>
 |
<pre>dec;17:4.dumstr; dec;17:4.dumstr; utstr* '-dumstr; ode voltage conversion************************************</pre>	<pre>Jostr+' + dumstr; Jostr+' + dumstr; Jostr+' + dumstr; Jostr+' + dumstr; Jostr+' + dumstr]; Jostr+' + dumstr2); Jostr+' + dumstr2); Jostr+' + dumstr2); Jostr+' + dumstr2); Jostr2)</pre>	<pre>utstr: + idumstr); utstr: + idumstr); utstr: + idumstr); ode voltage conversion************************************</pre>	<pre>ucstr:* * dumstr; dex]:7:4,dumstr); utstr:* * fdumstr); ode voltage conversion************************************</pre>	<pre>outst:: ' dumst:; .outst:: ' foumst:; idea voltage conversion************************************</pre>	<pre>outstr: ' dumstr; outstr: ' dumstr; outstr: ' fdumstr; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; idex[7:4,dumstr]; outstr+' 'fdumstr] ided voltage conversion************************************</pre>	<pre>outstr: ' + dumstr; ndex[:]' ; 4 dumstr); outstr: ' + dumstr); volt:=0.0; ioutstr: ' + dumstr] outstr: ' + dumstr2); y:6:2, dumstr2); outstr: ' + dumstr2); outstr: ' + dumstr2); inter::::::::::::::::::::::::::::::::::::</pre>	<pre>outstr:* fumstr; dex[:r*, fumstr; outstr** fumstr; outstr** fumstr; nv:=0.0; nv:=0.0; nvert_volt(setpt[index]); ris:2,dumstr2); ris:2,dumstr2); ris:ris:numstr2); nutstr** fumstr2); nutstr** fumstr2);</pre>	<pre>outstry + dumstry des[:7:4,dumstry]; outstry' + dumstry outstry' + dumstry prest_voltage conversion************************************</pre>	<pre>Ducstry *Oumstry dex[:7:4,dumstr]; vutstry * *dumstry ode voltage conversion************************************</pre>	outst:/.t.dumstr; dex]:?:4.dumstr); outst:/.t.dumstr); outst:/.t.dumstr]; outst:/.t.dumstr]); outst:/.tout(setpt[index]); outst:/.t.dumstr2); outst:/.t.dumstr2; outst:/.t.dumstr2;	outstr: ' toumstr; ndex]:?!4,dumstr; outstr:' (dumstr); volt:=0.00; iode voltage convertyoit (setpt[index]); y:5:2,dumstr2); outstr:' 'tdumstr2; tem:=temtresd_temp(Tcontre	oucstr+ + dumstr; ndex]:7:4,dumstr); outstr+ * +dumstr) iode voltage conversion************************************
<pre>dec;17:4.dumstr; dec;17:4.dumstr; utstr* '-dumstr; ode voltage conversion************************************</pre>	<pre>Jostr+' + dumstr; Jostr+' + dumstr; Jostr+' + dumstr; Jostr+' + dumstr; Jostr+' + dumstr]; Jostr+' + dumstr2); Jostr+' + dumstr2); Jostr+' + dumstr2); Jostr+' + dumstr2); Jostr2)</pre>	<pre>utstr: + idumstr); utstr: + idumstr); utstr: + idumstr); ode voltage conversion************************************</pre>	<pre>ucstr:* * dumstr; dex]:7:4,dumstr); utstr:* * fdumstr); ode voltage conversion************************************</pre>	<pre>outstr: ' dumstr; outstr: ' dumstr; outstr: ' dumstr; outstr: ' dumstr; outstr: ' dumstr; suist. ' dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); outstr: ' dumstr2); outstr: ' dumstr2); outstr: ' dumstr2); outstr: ' dumstr2); outstr: ' dumstr2); gain[point]:=read_gain[foont]; outstr: ' dumstr2);</pre>	<pre>outstr: ' dumstr; outstr: ' dumstr; outstr: ' fdumstr; iode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; idex[7:4,dumstr]; outstr+' 'fdumstr] ided voltage conversion************************************</pre>	<pre>outstr: ' + dumstr; ndex[:]' ; 4 dumstr); outstr: ' + dumstr); volt:=0.0; ioutstr: ' + dumstr] outstr: ' + dumstr2); y:6:2, dumstr2); outstr: ' + dumstr2); outstr: ' + dumstr2); inter::::::::::::::::::::::::::::::::::::</pre>	<pre>outstr:* fumstr; dex[:r*, fumstr; outstr** fumstr; outstr** fumstr; nv:=0.0; nv:=0.0; nvert_volt(setpt[index]); ris:2,dumstr2); ris:2,dumstr2); ris:ris:numstr2); nutstr** fumstr2); nutstr** fumstr2);</pre>	<pre>outstry + dumstry des[:7:4,dumstry]; outstry' + dumstry outstry' + dumstry prest_voltage conversion************************************</pre>	<pre>Ducstry *Oumstry dex[:7:4,dumstr]; vutstry * *dumstry ode voltage conversion************************************</pre>	outestr+ remark; ndex]:7:4, dumastr); ndex1:7:4, dumastr]; volt:=0.0; volt	outstr: ' toumstr; ndex]:?!4,dumstr; outstr:' (dumstr); volt:=0.00; iode voltage convertyoit (setpt[index]); y:5:2,dumstr2); outstr:' 'tdumstr2; tem:=temtresd_temp(Tcontre	oucstr+ + dumstr; ndex]:7:4,dumstr); outstr+ * +dumstr) iode voltage conversion************************************
<pre>dex]:7:4.dumstr; dex]:7:4.dumstr; dex]:7:4.dumstr; dev]:7:4.dumstr; de voltage conversion************************************</pre>	<pre>retput::-v.v. dex]:?:4,dumstr; dex[:?:4,dumstr]; volt:=0.0; de voltgeconversion************************************</pre>	<pre>utstr: ' + dumstr; utstr: ' + dumstr); utstr: ' + dumstr); ode voltage conversion************************************</pre>	<pre>utstr+' foumstr; dex]:?.4(umstr); dex]:?.4(umstr); ode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; ndex[:+' '+dumstr); outstr+' '+dumstr); ndex[:+' +dumstr]; notex[:+' +dumstr]; outstr+' +dumstr2); y:6:2,dumstr2); outstr+' +dumstr2); outstr+' +dumstr2); [index]:4:1,dumstr2); [index]:4:1,dumstr2); [index]:4:1,dumstr2); [index]:4:1,dumstr2); [index]:4:1,dumstr2); [index]:4:1,dumstr2); [index]:4:1,dumstr2);</pre>	<pre>undex1:f+' +dumstr; ndex1:f+' +dumstr; ndex1:f+' +dumstr; outstr+' +dumstr; outstr+' +dumstr2); y:0:2,dumstr2); y:0:2,dumstr2); y:0:2,dumstr2); y:0:1,dumstr2); undex1:f+' +dumstr2); undex1:f+' +dumstr2); outstr+' +dumstr2); giddex1:f+' +dumstr2); outstr+' +dumstr2); giddex1:f+' +dumstr2);</pre>	<pre>outstr+' + dumstr; does voltage conversion************************************</pre>	<pre>outstrt' + dumstr; outstrt' + dumstr; ndex] 7:4, dumstr; outstrt' + dumstr; sviit:=0.0; volt:=0.0; volt:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi for loop:=1 to average_readi for loop:=1 to average_readi for loop:=1 to average_readi funder]:3:0, dumstr2; volt:=voltread_volt(Wmeter volt:=voltread_volt(Wmeter volt:=voltread_volt(Wmeter volt:=roit); for loop: outstrt' + dumstr2; outstrt' + dumstr2; volter! - to unstr2; volter! - to unstr2; volter! - to unstr2; volter! - to unstr2;</pre>	<pre>Dutstr: ' dumstr; dowsir:' ' dumstr; Dutstr: ' dumstr; Dutstr: ' dumstr; Dutstr: ' dumstr2); r::2dumstr2); r::2dumstr2); r::1dowstr2); putstr: ' dumstr2); r::1dowstr2); putstr: ' dumstr2); putstr: ' dumstr2); r::1dowstr2); butstr: ' dumstr2); butstr: ' dumstr2); butstr: ' dumstr2); cutstr: ' dumstr2); butstr: ' dumstr2); cutstr: ' dumstr2); butstr: ' dumstr2); cutstr: ' dumstr2);</pre>	<pre>outstr* ' + dumstr; ddex[:r* / dumstr; outstr*' ' + dumstr; outstr*' ' + dumstr; substr2'; ref:2, dumstr2); ref:2, dumstr2); ref:2, dumstr2); ref:2, dumstr2); ref:4, dumstr2, dumstr2); ref:4, dumstr3, dumstr3, dumstr3, dumstr4, d</pre>	<pre>JULG STT: * * JULNET STT: * * YOL STT: * * O.O; JULNET * * * * * * * * * * * * * * * * * * *</pre>	oucstr+ '-tumetr; ndex]:14.4umetr); ndex]:*t-4umetr]; tode voltege conversion************************************	outstrt' toumstr; ndex]:7:4,dumstr; outstrt' toumstr; outstrt' toumstr; outstrt_Vitage conversion************************************	outstr+' + dumstr; ndex]:7:4,dumstr); outstr+' (+dumstr) iode voltage conversion************************************
<pre>decs::?:4.dumstr; decs::?:4.dumstr; dets:r:4.dumstr; de voltage conversion************************************</pre>	<pre>lecpn::=v.u; dex]:?:4,dumstr; de voltege conversion************************************</pre>	<pre>utstr: ' + dumstr); utstr: ' + dumstr); utstr: ' + dumstr); ode voltage conversion************************************</pre>	<pre>ucstr+ '-toumstr; dex]:7:4,dumstr); dex]:7:4,dumstr); devoltage conversion************************************</pre>	<pre>outstrt 'tdumstr; outstrt' 'tdumstr); outstrt' (dumstr); outstrt' (dumstr); outstrt' (dumstr); outstrt' (dumstr); y:6:2,dumstr2); y:6:2,dumstr2); outstrt' 'dumstr2); outstrt' 'dumstr2); outstrt' 'dumstr2); (index]:4:1,dumstr2); (index]:4:1,dumstr2); (index]:4:1,dumstr2); (index]:4:1,dumstr2); (index]:4:1,dumstr2); (index]:4:1,dumstr2); (index]:4:1,dumstr2);</pre>	<pre>outstr: ' dumstr; outstr: ' dumstr; outstr: ' fdumstr; outstr: ' fdumstr; suistr: ' dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); outstr: ' dumstr2); suistr: ' fdumstr2); suistr: ' fdumstr2); gidds1:3:0,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2); gidds1:4:1,dumstr2);</pre>	<pre>outstr+' + dumstr; deekir:=0.0; outstr+' + dumstr); outstr+' + dumstr); outstr+' + dumstr2; y:6:2, dumstr2); outstr+' + dumstr2);</pre>	<pre>outstrt' + dumstr; ndex[:1:4,dumstr; ndex[:7:4,dumstr]; volt:=0.0; volt:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi fudex[:4:1,dumstr2]; outstrt' + dumstr2]; power[point]:=read heater po outstrt' + dumstr2];</pre>	<pre>Dutstry ' dumstry dex[:7: ' dumstry' Dutstry' ' dumstry' Dutstry' ' dumstry' nv:=0.0; nvert_volt(setpt[index]); rist_dumstr2);</pre>	<pre>outstr* ' + dumstr; ddex[:7' + dumstr]; utstr*' ' + dumstr] outstr*' ' + dumstr] ; volt=0.0; y:6:2, dumstr2); y:6:2, dumstr2); rem:=tem+read_temp(Tcontro) cutstr*' + dumstr2); utstr*' + dumstr2); utstr*' + dumstr2); utstr*' + dumstr2); utstr*' + dumstr2); index]:4:1, dumstr2); end; [loop]</pre>	<pre>Jutestry 'volumetry Jutestry' 'volumetry Jutestry' (Jumetry); So Ioop:=1 to average_readin for Ioop:=1 to average_readin for Joop:=1 to average_readin</pre>	outstr+ '-tumetr; ndex]:7.4.dumetr); ude voltege conversion************************************	outstrt' toumstr; ndex]:7:4, dumstr; outstrt' toumstr; outstrt' toumstr; outstrt Vitage conversion************************************	outstr+' + dumstr; ndex]:7:4,dumstr); outstr+' (+dumstr) iode voltage conversion************************************
<pre>detr:/ 'dumstr; dets:/'f.4.dumstr; tdst/f.4.dumstr); udst/f.4.dumstr); de voltage conversion************************************</pre>	<pre>Listr+' ^ dumstr; be:]?:4,dumstr; be:]?:4,dumstr; be:]?:4,dumstr; be:]?:0, itstr+' ^ dumstr]); volt(etpt[index]); volt(etpt[index]); itstr+' ^ dumstr2); item:read_volt(Vmet volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmetc.de end; [lop] vietr+' ^ dumstr2); power[point]:=read_gain(Tcon gain[point]:=read_gain(Tcon</pre>	<pre>utstr+ '+dumstr; dex!??4,dumstr); utstr+ '+dumstr); ode voltage conversion************************************</pre>	<pre>utstr+' 'dumstr; dex]?f4,dumstr; dex]?f4,dumstr); utstr+' 'dumstr]; utstr+' 'dumstr]; utstr+' dumstr2); nvert_volt(setpt[index]); for loop:=1 to average_readi for loop:=1 to average_rea</pre>	<pre>outstr:/ 'dumstr; outstr:/ 'dumstr; ndex]:?.4,dumstr); outstr:/ 'dumstr] nowert Yolt (setpt[index]); y:5:2,dumstr2); outstr:/ 'dumstr2); outstr:/ 'dumstr2); outstr:/ 'dumstr2); outstr:/ 'dumstr2); outstr:/ 'dumstr2); gain[point]:=read_gain[foont]; outstr:/ 'dumstr2); gain[point]:=read_gain[foont]; gain[point]:=read_gain[foont]; data[dumstr2];</pre>	<pre>outstr+' +dumstr; adex1;+' +dumstr; idex1;+' +dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' + dumstr; ndex :7:4, dumstr); outstr+' + dumstr); notstr+' + dumstr); onvert_Volt(setpt[index]); y:6:2, dumstr2); outstr+' + dumstr2); outstr+' + dumstr2); outstr+' + dumstr2); outstr+' + dumstr2); findex]:4:1, dumstr2); outstr+' + dumstr2); outstr+' + dumstr2); outstr+' + dumstr2); outstr+' + dumstr2);</pre>	<pre>outstr+' + dumstr; mdex :7:4, dumstr); outstr+' + dumstr); y outstr=0.0; y outstr+' + dumstr]); y outstr+' + dumstr2; y:6:2, dumstr2; outstr+' + dumstr2; outstr+' + dumstr2; y outstr+' + dumstr2;</pre>	<pre>ultstr* ' dumstr; dex[7:4 dumstr; - dumstr; ultstr* ' dumstr; -0.0; ultstr* ' dumstr]; outett*' ' dumstr2; rsi2 dumstr2; rs</pre>	<pre>outstr+' 'dumstr; dex!:^'.dumstr; outstr+' 'dumstr; outstr+' 'dumstr); volt=0.0; volt=0.0; volt=equilient. pret_volt=ge conversion************************************</pre>	<pre>utstr*' *dumstr; dex]??&dumstr); utstr*' *dumstr); utstr*' *dumstr); ood voltage conversion************************************</pre>	outstr+' 'dumstr; ndex]:7:4,dumstr); outstr+' 'dumstr; outstr+' 'dumstr; outstr+' 'dumstr); for loop:=1 to average_read y:6:2,dumstr2); tem:rted_temtred_temp(fortz); tem:rtentred_temp(fortz);	outstr+' ' + dumstr; ndex]: 7:4, dumstr); outstr+' ' + dumstr; outstr+' (+ dumstr; outstr+' (+ dumstr; y:5:2, dumstr2); w:5:2, dumstr2); outstr+' ' + dumstr2; tem:=tem+read_temp(Tcontre	outstr+' '+dumstr; ndex]:?:4,dumstr); outstr+' '+dumstr); iode voltage conversion************************************
<pre>detsit:/ 'dumstr; dex]:7:4.dumstr; utestr* 'dumstr); ode volte=0.0; ode volte=0.0; nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]);</pre>	<pre>tetpnt:=v.v; detpl://t.dumetr; detvoltage conversion************************************</pre>	<pre>utatr+' ' +dumatr; utatr+' ' +dumatr; dex]:7:4,dumatr; ode voltage converion************************************</pre>	<pre>utstr:/ 'dumstr; dexj:?:4.dumstr; utstr:/ 'dumstr; ode voltege conversion************************************</pre>	<pre>outstr+' 'dumstr; dowsi7:4, dumstr; ndexi7:4, dumstr); outstr+' 'dumstr); yode voltage conversion************************************</pre>	<pre>outstr+' 'dumstr; dex]:?!4,dumstr; outstr+' 'dumstr); outstr+' 'dumstr); outstr+' 'dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2);</pre>	<pre>outstr+' 'dumstr; ndex]:?!4,dumstr; ndex]:?!4,dumstr); outstr+' 'dumstr); volt:=0.0; volt:=0.0; volt=ecpt[index]); volt=ecpt[index]); volt=ecpt[index]]; volt=read_temp[fcontro tindex]:4.1,dumstr2); outstr+' 'dumstr2); volt=read_volt[Vmeter.de outstr+' 'dumstr2); volt=read_heater_p; outstr+' 'dumstr2);</pre>	<pre>outstr+' 'dumstr; dex]:?:4,dumstr); udex[:?:4,dumstr); outstr+' 'dumstr); yode voltage conversion************************************</pre>	<pre>utstr' 'dumstr; dex!:?4dumstr; utstr+' 'dumstr; utstr+' 'dumstr; utstr+' 'dumstr; nv:=0.0; rst.dumstr2); nv:=0.0; for loop:=1 to average_readin rst.trutt); for loop:=1 to average_readin for loop:=1 to average_readin readingtr=' 'dumstr2'; outstr+' 'dumstr2'; end: (loop) for loop:=1,0; nutstr+' 'dumstr2'; end: (loop)</pre>	<pre>outstr*' 'dumstr; dex]:7:4,dumstr); outstr*' 'dumstr); volt:=0.0; volt:=0.0; ren:=rent:=0; ren:=rent:====================================</pre>	<pre>utstr+' 'dumstr; idex::?:4,dumstr; utstr+' 'dumstr; outstr+' 'dumstr; outstr+' 'dumstr; nverLyolt(setpt[index]); r:6:2,dumstr2); utstr+' 'dumstr2; utstr+' 'dumstr2; utstr+' 'dumstr2; index]:3:0,dumstr2; utstr+' 'dumstr2; index]:4:1,dumstr2; findex]:4:1,dumstr2; index]:4:1,dumstr2;</pre>	outstr* '+dumstr; ndex]:?:4.dumstr); outstr* '+dumstr; oude voltage conversion************************************	outstr'' 'dumstr; setput:u'u'; setput:u'u'; setput:u'u'; setput:u'u'; setput:u'u'; setu: 'devisi':'d', dumstr; suit:-0.0; voit:-0.0;	<pre>outstr' ' + dumstr; ndex]:?:4, dumstr); outstr+ ' + dumstr); outstr+ ' + dumstr; outstr+ ' + dumstr2); for loop:=1 to average_readi for loop:=1 to average_readi for loop:=1 to average_readi begin = tem+read temp[Tcontrc outstr+ ' + dumstr2);</pre>
<pre>detpnt:*' 'dumstr; dex]:7:4,dumstr; utestr*' 'fdumstr); ods voltege conversion************************************</pre>	<pre>tcstr+' 'dumstr; dex]:?;4,dumstr; dex]:?;4,dumstr; setr=-0.0; test+' 'fdumstr; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; for loop:=1 to average_read begin test+' 'fdumstr2); utstr+' 'dumstr2); utstr+' 'dumstr2); power[point]:=read_gain[Toon gain[point]:=read_gain[Toon</pre>	<pre>utetr+' '+dumetr; utetr+' '+dumetr; dex]:7:4,dumetr); utetr+' '+dumetr; ode voltage converion************************************</pre>	<pre>utstr:' 'dumstr; dex]:?:4.dumstr; utstr:' 'dumstr; ods voltege conversion************************************</pre>										
 | <pre>outstr+' 'dumstr;
outstr+' 'dumstr;
idex]:?:4,dumstr);
outstr+' 'dumstr);
outstr+' 'dumstr2);
y:6:2,dumstr2);
y:6:2,dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
findex]:4:1,dumstr2);
gain[point]:=read_parter_power[point]:=read_parter_popurt];
outstr+' 'dumstr2);
findex]:4:1,dumstr2);</pre>
 | <pre>outstr+' 'dumstr;
outstr+' 'dumstr;
idex[:7:4,dumstr);
outstr+' 'dumstr);
outstr+' 'dumstr2);
power_Volt(setpt[index]);
r:::dmmstr2);
outstr+' 'dumstr2);
i(index]:3:0,dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);</pre> | <pre>outstr+' 'dumstr;
dex[:7:4,dumstr);
outstr+' 'dumstr);
voutstr=0.0;
y:0:2,dumstr]);
y:6:2,dumstr2);
y:6:2,dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);
outstr+' 'dumstr2);</pre> | <pre>outstr' 'dumstr;
adax]:?:4,dumstr);
udex[:?:4,dumstr);
outstr' 'dumstr);
ude voltage
conversion************************************</pre> | <pre>outstr*' '+dumstr;
dex]:?4,dumstr);
outstr*' '+dumstr);
outstr*' '+dumstr);
iode voltage conversion************************************</pre> | <pre>ultstr' 'dumstr;
dex!:7:4,dumstr);
ultstr+' 'dumstr);
volt:=0.0;
volt:=0.0;
nv:=0.0;
for loop:=1 to average_readi
for loop:=1 to average_readi
volt:=-currated_current(Im
volt:=-currated_volt(Imeter
outstr+' 'dumstr2);
findes]:3:1,dumstr2);
end; [loop]</pre>
 | <pre>utstr+' 'dumstr;
dex!:?:4,dumstr;
utstr+' 'dumstr;
outstr+' 'dumstr;
outstr+' 'dumstr;
prec.2,dumstr2);
r:6:2,dumstr2);
utstr+' 'dumstr2;
utstr+' 'dumstr2;
intexr:' 'dumstr2;
utstr+' 'dumstr2;
utstr+' 'dumstr2;
utstr+' 'dumstr2;
index]:4:1,dumstr2;
findex]:4:1,dumstr2;
intexr2;
intexr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2;
interr2</pre> | <pre>outstr' ' dumstr;
ndex]:?:4.dumstr);
outstr' ' dumstr);
outstr' ' dumstr;
outstr' ' dumstr;
outstr' ' dumstr2);
y:6:2.dumstr2);
outstr+' ' dumstr2);
tem:=temtred_temp[findex]</pre> | outstr+''+dumstr; setpht:=U.U;
ndex]:?:4,dumstr); curr:=0.0;
outstr+''+dumstr; N:=0.0;
outstr-''+dumstr; N:=0.0;
voltage conversion************************************ | <pre>outstr+' ' +dumstr;
outstr+' ' +dumstr;
outstr+' ' +dumstr;
outstr+' ' +dumstr;
outstr+' ' +dumstr;
fof ecpt[index]);
fof ecpt[index]);
onvert_Uot(setpt[index]);
begin = to average_readi
outstr+' ' +dumstr2);
begin = tem+read temp[Tcontrc
outstr+' +dumstr2);</pre>
 |
<pre>utstr+' +dumstr; dex1:7:4,dumstr; utstr+' +dumstr; ods voltage conversion************************************</pre>	<pre>stort:^(.); dex[:7:4.dumstr; dex[:7:4.dumstr]; stort=0.0; ode voltage_conversion************************************</pre>	<pre>utatr+' '+dumatr; dex!:7:4,dumatr); dex1:7:4,dumatr); ode voltage conversion************************************</pre>	<pre>utstr:' 'dumstr; dex]:??.dumstr; dex]:??.dumstr; utstr:' 'fdumstr; mv:=0.0; nvsrt Volt(setpt[index]); nvert Volt(setpt[index]);</pre>	<pre>outstr' 'dumstr; outstr' 'dumstr; idex[:?4,dumstr); outstr' 'dumstr); outstr' 'dumstr); yol:=0.0; yol:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; for loop:=1 to average_readi begin to nv:=nv:=nv:=1:0; hegin to nv:=nv:=nv:=0; hegin to nv:=nv:=nv:=0; hegin to nv:=nv:=nv:=0; hegin to nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=</pre>	<pre>outstr+' +dumstr; outstr+' +dumstr; idex[:?4,dumstr); outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr2); s:s:dimstr2); cutstr+' +dumstr2); outstr+' +dumstr2);</pre>	<pre>outstr' 'dumstr; outstr' 'dumstr; ndex]:?:4,dumstr); outstr' 'dumstr); outstr' 'dumstr); y.ot.ol.0; y:6:2,dumstr2); y:6:2,dumstr2); outstr' 'dumstr2); outstr' 'dumstr2); outstr' 'dumstr2); outstr' 'dumstr2); outstr' 'dumstr2); outstr' 'dumstr2); outstr' 'dumstr2); outstr' 'dumstr2);</pre>	<pre>outstr' ' +dumstr; outstr' ' +dumstr; iddx]: 7.4, dumstr); outstr' ' +dumstr); outstr' ' +dumstr2); y: 6:2, dumstr2); y: 6:2, dumstr2); poutstr' ' +dumstr2); outstr' ' +dumstr2);</pre>	<pre>Dutstr+ '+dumstr; dex]:?.4,dumstr); uutstr+ '+dumstr; uutstr+ '+dumstrs; iode voltge conversion************************************</pre>	<pre>outstr' '+dumstr; outstr' '+dumstr; dex]:?f,dumstr); outstr' '+dumstr); volt:=0.0; volt:=0.0; nv:=0.0; for loop:=1 to average_readi nv:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi for loop:=1 to average_readi for loop:=1 to average_readi nv:=0.0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=1:=0; nv:=nv:=nv:=nv:=1:=0; nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=nv:=</pre>	<pre>Dutstr+' '+dumatr; dex]:7:4,dumstr); uutstr+' '+dumstr); putstr+' '+dumstr); putstr+' '+dumstr2); preiz,dumstr2); preiz,dumstr2); preiz,dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2); putstr+' '+dumstr2); findex]:4:1,dumstr2); end; [loop]</pre>	<pre>outstr+ ' +dumstr; outstr+ ' +dumstr; ndex]:?.4.dumstr); outstr+ ' +dumstr; outet+ ' +dumstr; outet Volt (setpt[index]); y:6.2,dumstr2); outstr+ ' +dumstr2;</pre>	<pre>outstr+ ' +dumstr; setpnt:=0.0; ndex]:?4,dumstr); outstr+ ' +dumstr; volt:=0.0; jode voltage conversion************************************</pre>	<pre>outstr+' '+dumstr; setpnt:=0.0; ndex]:?4,dumstr; voinstr: outstr+' '+dumstr; voil:=0.0; outstr+' '+dumstr; noisenannannannannannannannannannannannannan</pre>
<pre>utetr+ '+dumstr; dex]:7:4,dumstr; utetr+ '+dumstr; ode voltage conversion************************************</pre>	<pre>start* '+dumstr; dex]:7:4.dumstr; dex[:7:4.dumstr]; dex[:7:4.dumstr]; det log:=0.0; det Voltge conversion************************************</pre>	<pre>utatr+' '+dumatr; dex]:7:4,dumatr); dex]:7:4,dumatr); dex]:7:4,dumatr); ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; dex]:7:4.dumstr); utstr+' '+dumstr; dex[:7:4.dumstr]; utstr+' '+dumstr]; utstr+' '+dumstr]; nvert_Volt(eetpt[index]); s:2.dumstr2); utstr+' '+dumstr2); utstr+' '+dumstr2);</pre>	<pre>outerr' 'dumerr; outerr' 'dumerr); outerr' 'dumerr); outerr' 'dumerr); outerr' 'dumerr); yol:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0:=1 to average_readi begin tem:=temtread_temp(Tcontro begin begin temerr2; outerr' 'dumerr2; outerr' 'dumerr2; outerr' 'dumerr2; jober [point]:=read_peint[:=read_gein(Tcont gain[point]:=read_gein(Tcont gain[point]:=read_gein(Tcont power [point]:=read_gein(Tcont power [point]:=read_gein(Tcont gain[point]:=read_gein(Tcont geint[point]:=read_gein(Tcont geint[point]:=read_gein(Tcont geint[point]:=read_gein(Tcont geint[point]:=read_gein(Tcont geint[point]:=read_gein(Tcont geint[point]:=read_geint[point]:=read_geint[point]:=read_geint[point]:=read_geint[point]:=read_geint[point[point]:=read_geint[point]:=read_geint[point[point]:=read_geint[point]:=read_geint[point]:=read_geint[point[point]:=read_geint[point[point[point]]:=read_geint[point[point[point]]:=read_geint[point[point[point[point[point[point[point[point[point[point[point[point[point[poi</pre>	<pre>outstr+' 'dumstr; outstr+' 'dumstr; index]:?:4,dumstr; outstr+' 'dumstr; outstr+' 'dumstr; onvert_Volt(setpt[index]); y:6.2,dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2);</pre>	<pre>outstr' 'dumstr; outstr' 'dumstr; ndex]:?4,dumstr); outstr' 'dumstr); outstr' 'dumstr); yode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; outstr+' '+dumstr; idde voltage conversion************************************</pre>	<pre>Dutstr+ '+dumstr; dex]:7:4,dumstr; udst]:7.4,dumstr; udstr-' '+dumstr; uode voltge conversion************************************</pre>	<pre>outstr+ '+dumstr; setpnt:=0.0; dex]:7:4,dumstr; vdumstr; vo: outstr+' '+dumstr; vo: corr:=0.0; vo:ti=0.0; for loop:=1 to average_readi p:c2,dumstr2); p:c2,dumstr2); vist.************************************</pre>	<pre>Dutstr+' '+dumstr; dex]:7.4,dumstr); udst:+' +dumstr); udst.+' +dumstr); putstr+' +dumstr; poe voltage conversion************************************</pre>	<pre>outstr+ ' +dumstr; outstr+ ' +dumstr); utstr+ ' +dumstr); outstr+ ' +dumstr); outstr+ ' +dumstr); outstr+ ' +dumstr); utstr+ ' +dumstr2); v:6:2, dumstr2); tem:rted_tempt</pre>	<pre>outstr+' ' +dumstr; outstr+' ' +dumstr); outstr+' ' +dumstr; outstr+' ' +dumstr; outstr+' ' +dumstr; y de conversion************************************</pre>	<pre>outstr+' '+dumstr; setpnt:=0.0; ndex]??4,dumstr; vunstr); outstr+' '+dumstr; volt:=0.0; outstr+' '+dumstr; volt:=0.0; sovert_Volt(setpt[index]); for average_readi ouvert_Volt(setpt[index]); begin to average_readi outstr+' '+dumstr2); tem:tem+read temp[Tcontrc outstr+' '+dumstr2);</pre>
<pre>utstr+ '+dumstr; dex]:7:4,dumstr; utstr+ '+dumstr; utstr+ '+dumstr; ode voltage conversion************************************</pre>	<pre>stpnt:=0.0; dex]:7:4,dumstr; dex]:7:4,dumstr; suit:=0.0; suit:=0.0; de voltege conversion************************************</pre>	<pre>utstr+ '+dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; utstr+'+dumstr; utstr+'+dumstr; utstr+'+dumstr); utstr+'+dumstr2); for loop:=1 to average_readi nv:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi</pre>	<pre>utstr+ ' +dumstr; dex]:7:4,dumstr); utstr+ ' +dumstr); utstr+ ' +dumstr); utstr+ ' +dumstr); ode voltage conversion************************************</pre>	outsEr+ '+dumstr;metpnt:=0.0;ndex]:?:4,dumstr;uerr:=0.0;ndex]:?:4,dumstr;volt:=0.0;outsEr+ '+dumstr;volt:=0.0;iode voltage conversion************************************	outstr+' + dumstr;metpnt:=0.0;ndex]:7:4, dumstr;volt:=0.0;outstr+' + dumstr;volt:=0.0;outstr+' + dumstr2;volt:=0.0;idde voltage conversion************************************	<pre>outstr+' +dumstr; dex]:?4,dumstr; ndex]:?4,dumstr; outstr+' +dumstr; outstr=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; west_ent_ent[index]; y:6:2,dumstr2); outstr+' +dumstr2); wilt=volt+read_volt(Vmete.dev outstr+' +dumstr2); wilt=volt+read_volt(Vmete.dev outstr+' +dumstr2); power[point]:=read_heater_p; outstr+' +dumstr2);</pre>	outstr+' +dumstr;setpnt:=0.0;idex]:7:4,dumstr;curr:=0.0;iddex]:7:4,dumstr;volt:=0.0;outstr+' +dumstr;volt:=0.0;outstr+' +dumstr;n/:=0.0;outstr+' +dumstr2;for loop:=1 to average_readioutstr+' +dumstr2;for loop:=1 to averageoutstr+' +dumstr2;for loop:=1 to average <t< td=""><td><pre>Dutstr+' +dumstr; idex]:7:4,dumstr; udsti:7:4,dumstr; uutstr+' '+dumstr; uode voltage conversion************************************</pre></td><td><pre>outstr+' +dumstr; dex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; utstr+' +dumstr; nv=0.0; nvert_Volt(setp[index]); p:6:2,dumstr2);</pre></td><td><pre>putstr+ '+dumstr; dex]:7:4,dumstr); udstr+ '*dumstr); putstr+ '*dumstr); pode voltage conversion************************************</pre></td><td><pre>outstr+ '+dumstr; ndex]:7:4,dumstr; cutrt+' '+dumstr; volt=0.0; iode voltage conversion************************************</pre></td><td><pre>outstrt ' + dumstr; ndex]:?:4,dumstr; outstrt ' + dumstr; outstrt ' + dumstr; iode voltage conversion************************************</pre></td><td><pre>outstr+' +dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; for loo; for loop:=l to average_read; for loop:=l to average_read; for loop:=l to average_read; outstr+' +dumstr2); begin</pre></td></t<>	<pre>Dutstr+' +dumstr; idex]:7:4,dumstr; udsti:7:4,dumstr; uutstr+' '+dumstr; uode voltage conversion************************************</pre>	<pre>outstr+' +dumstr; dex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; utstr+' +dumstr; nv=0.0; nvert_Volt(setp[index]); p:6:2,dumstr2);</pre>	<pre>putstr+ '+dumstr; dex]:7:4,dumstr); udstr+ '*dumstr); putstr+ '*dumstr); pode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; ndex]:7:4,dumstr; cutrt+' '+dumstr; volt=0.0; iode voltage conversion************************************</pre>	<pre>outstrt ' + dumstr; ndex]:?:4,dumstr; outstrt ' + dumstr; outstrt ' + dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' +dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; for loo; for loop:=l to average_read; for loop:=l to average_read; for loop:=l to average_read; outstr+' +dumstr2); begin</pre>
<pre>utstr+ '+dumstr; dex]:7:4,dumstr; utstr+ '+dumstr; utstr+ '+dumstr; ode voltage conversion************************************</pre>	<pre>stpnt:=0.0; dex]:7:4,dumstr; dex]:7:4,dumstr; suit:=0.0; suit:=0.0; de voltege conversion************************************</pre>	<pre>utstr+ '+dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; utstr+'+dumstr; utstr+'+dumstr; utstr+'+dumstr); utstr+'+dumstr2); for loop:=1 to average_readi nv:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi</pre>	<pre>utstr+ ' +dumstr; dex]:7:4,dumstr); utstr+ ' +dumstr); utstr+ ' +dumstr); utstr+ ' +dumstr); ode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; index]:?!4,dumstr; outstr+' +dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre>	outstr+' + dumstr;metpnt:=0.0;ndex]:7:4, dumstr;volt:=0.0;outstr+' + dumstr;volt:=0.0;outstr+' + dumstr2;volt:=0.0;idde voltage conversion************************************	<pre>outstr+' +dumstr; dex]:?4,dumstr; ndex]:?4,dumstr; outstr+' +dumstr; outstr=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; mv:=0.0; formstr=0.0; mv:=volt=read_temp(fcontro begin currierdicationtri b</pre>	outstr+' +dumstr;setpnt:=0.0;idex]:7:4,dumstr;curr:=0.0;iddex]:7:4,dumstr;volt:=0.0;outstr+' +dumstr;volt:=0.0;outstr+' +dumstr;n/:=0.0;outstr+' +dumstr2;for loop:=1 to average_readioutstr+' +dumstr2;for loop:=1 to averageoutstr+' +dumstr2;for loop:=1 to average <t< td=""><td><pre>Dutstr+' +dumstr; idex]:7:4,dumstr; udsti:7:4,dumstr; uutstr+' '+dumstr; uode voltage conversion************************************</pre></td><td><pre>outstr+' +dumstr; dex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; utstr+' +dumstr; nv=0.0; nvert_Volt(setp[index]); p:6:2,dumstr2);</pre></td><td><pre>putstr+ '+dumstr; dex]:7:4,dumstr); udstr+ '*dumstr); putstr+ '*dumstr); pode voltage conversion************************************</pre></td><td><pre>outstr+ '+dumstr; adex]:?:4,dumstr; adex]:7:4,dumstr; cutr:=0.0; volt:=0.0; iode voltage conversion************************************</pre></td><td><pre>outstrt ' + dumstr; ndex]:?:4,dumstr; outstrt ' + dumstr; outstrt ' + dumstr; iode voltage conversion************************************</pre></td><td><pre>outstr+' +dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; for loo; for loop:=l to average_read; for loop:=l to average_read; for loop:=l to average_read; outstr+' +dumstr2); begin</pre></td></t<>	<pre>Dutstr+' +dumstr; idex]:7:4,dumstr; udsti:7:4,dumstr; uutstr+' '+dumstr; uode voltage conversion************************************</pre>	<pre>outstr+' +dumstr; dex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; utstr+' +dumstr; nv=0.0; nvert_Volt(setp[index]); p:6:2,dumstr2);</pre>	<pre>putstr+ '+dumstr; dex]:7:4,dumstr); udstr+ '*dumstr); putstr+ '*dumstr); pode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; adex]:?:4,dumstr; adex]:7:4,dumstr; cutr:=0.0; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstrt ' + dumstr; ndex]:?:4,dumstr; outstrt ' + dumstr; outstrt ' + dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' +dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; for loo; for loop:=l to average_read; for loop:=l to average_read; for loop:=l to average_read; outstr+' +dumstr2); begin</pre>
<pre>utstr+ '+dumstr; dex]:7:4,dumstr; utstr+ '+dumstr; utstr+ '+dumstr; ode voltage conversion************************************</pre>	<pre>stpnt:=0.0; dex]:7:4,dumstr; dex]:7:4,dumstr; suit:=0.0; suit:=0.0; de voltege conversion************************************</pre>	<pre>utstr+ '+dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; utstr+'+dumstr; utstr+'+dumstr; utstr+'+dumstr); utstr+'+dumstr2); for loop:=1 to average_readi nv:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi</pre>	<pre>utstr+ ' +dumstr; dex]:7:4,dumstr); utstr+ ' +dumstr); utstr+ ' +dumstr); utstr+ ' +dumstr); ode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; .ndex]:?:4,dumstr); outstr+' +dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre>	outstr+' + dumstr;metpnt:=0.0;ndex]:7:4, dumstr;volt:=0.0;outstr+' + dumstr;volt:=0.0;outstr+' + dumstr2;volt:=0.0;idde voltage conversion************************************	<pre>outstr+' +dumstr; dex]:?4,dumstr; ndex]:?4,dumstr; outstr+' +dumstr; outstr=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; wilt=0.0; mv:=0.0; formstr=0.0; mv:=volt=read_temp(fcontro begin currierdicationtri b</pre>	outstr+' +dumstr;setpnt:=0.0;idex]:7:4,dumstr;curr:=0.0;iddex]:7:4,dumstr;volt:=0.0;outstr+' +dumstr;volt:=0.0;outstr+' +dumstr;n/:=0.0;outstr+' +dumstr2;for loop:=1 to average_readioutstr+' +dumstr2;for loop:=1 to averageoutstr+' +dumstr2;for loop:=1 to average <t< td=""><td><pre>Dutstr+' +dumstr; idex]:7:4,dumstr; udsti:7:4,dumstr; uutstr+' '+dumstr; uode voltage conversion************************************</pre></td><td><pre>outstr+' +dumstr; dex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; utstr+' +dumstr; nv=0.0; nvert_Volt(setp[index]); p:6:2,dumstr2);</pre></td><td><pre>putstr+ '+dumstr; dex]:7:4,dumstr); udstr+ '*dumstr); putstr+ '*dumstr); pode voltage conversion************************************</pre></td><td><pre>outstr+ '+dumstr; adex]:?:4,dumstr; adex]:7:4,dumstr; cutr:=0.0; volt:=0.0; iode voltage conversion************************************</pre></td><td><pre>outstrt ' + dumstr; ndex]:?:4,dumstr; outstrt ' + dumstr; outstrt ' + dumstr; iode voltage conversion************************************</pre></td><td><pre>outstr+' +dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; for loo; for loop:=l to average_read; for loop:=l to average_read; for loop:=l to average_read; outstr+' +dumstr2); begin</pre></td></t<>	<pre>Dutstr+' +dumstr; idex]:7:4,dumstr; udsti:7:4,dumstr; uutstr+' '+dumstr; uode voltage conversion************************************</pre>	<pre>outstr+' +dumstr; dex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; utstr+' +dumstr; nv=0.0; nvert_Volt(setp[index]); p:6:2,dumstr2);</pre>	<pre>putstr+ '+dumstr; dex]:7:4,dumstr); udstr+ '*dumstr); putstr+ '*dumstr); pode voltage conversion************************************</pre>	<pre>outstr+ '+dumstr; adex]:?:4,dumstr; adex]:7:4,dumstr; cutr:=0.0; volt:=0.0; iode voltage conversion************************************</pre>	<pre>outstrt ' + dumstr; ndex]:?:4,dumstr; outstrt ' + dumstr; outstrt ' + dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' +dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; for loo; for loop:=l to average_read; for loop:=l to average_read; for loop:=l to average_read; outstr+' +dumstr2); begin</pre>
<pre>det_sif_s_dumstr; dex]i7.4_dumstr; dex]i7.4_dumstr; utetr+' +dumstr; ode voltage conversion************************************</pre>	<pre>setpnt:=0.0; setsrt:' + dumstr; dex]:7:4,dumstr; tstrt' + dumstr; ode volte=0.0; volt:=0.0; volt:=0.0; volt:=0.0; for loop:=1 to average_read for loop:=1 to average_</pre>	<pre>utstr+ '+dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; utstr+' +dumstr; utstr+' +dumstr; utstr+' +dumstr; nver_Uolt(setpt[index]); nver_Uolt(setpt[index]); for loop:=1 to average_readi for loop:=1 to average_readi for</pre>	<pre>utetr+' +dumstr; dex]:7:4,dumstr; detr=0.0; utetr+' +dumstr; ode voltage conversion************************************</pre>	outstr+' +dumstr;metpht:=0.0;outstr+' +dumstr;volt:=0.0;index]:7:4,dumstr;volt:=0.0;outstr+' +dumstr;volt:=0.0;iode voltage conversionvolt:=0.0;iode voltage conversionvolt:=0.0;iode voltage conversionvolt:=0.0;iode voltage conversionvolt:=0.0;iode voltage conversionvolt:=0.0;volt:scdumstr2;volt=voltvolts:/dumstr2;volt:scdumstr2;voltage voltage conversionvolt:scdumstr2;voltage voltage voltagevoltvoltage voltagevoltagevoltage voltagevoltagevoltage voltage <td><pre>outstr+' +dumstr; ndex]:?.4,dumstr; ndex]:?.4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre></td> <td><pre>outstr+' 'dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' 'dumstr; soutstr+' 'dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); putstr+' 'dumstr2); findex]:3:0,dumstr2); outstr+' 'dumstr2); power[point]:=read_heater_power[point]:=re</pre></td> <td><pre>outstr+' 'dumstr; ndex]:?:4,dumstr; ndex]:?:4,dumstr; outstr+' 'dumstr; outstr+' 'dumstr; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; for loop:=l to average_readi for loop:=l to average_readi f</pre></td> <td><pre>ultstr+' ^dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; ulestr+' ^dumstr; ulestr+' ^dumstr; nvert_volt=ecpt[index]); for loop:=l to average_readir for loop</pre></td> <td><pre>outstr+' '+dumstr; dex]:7:4,dumstr; ndex]:7:4,dumstr; curr:=0.0; ulstr+' '+dumstr; outstr+' '+dumstr2); putstr*' '+dumstr2); rest_dumstr2); rest_dumstr2); rest_dumstr2); rest_dumstr2); rest_dumstr2); rest_dumstr2); nutstr+' '+dumstr2); rest_</pre></td> <td><pre>utstr* ' +dumstr; utstr* ' +dumstr; dex]: 7:4, dumstr; utstr* ' +dumstr; utstr* ' +dumstr; utstr* ' +dumstr2); ric:2, dumstr2); ric:2, dumstr2);</pre></td> <td><pre>outstr+' +dumstr; .ndex]:7:4,dumstr; .ndex]:7:4,dumstr; outstr+' +dumstr; .outstr+' +dumstr; .outstr+' 'dumstr2); .outstr+' 'dumstr2; .outstr+' 'dumstr2;</pre></td> <td><pre>outstr+' + dumstr; outstr+' + dumstr; ndex]:?.4, dumstr; outstr+' + dumstr; outstr+' + dumstr; iode voltage conversion************************************</pre></td> <td><pre>outstr+' ' dumstr; outstr+' ' dumstr; ndex]:7:4,dumstr; outstr+' ' dumstr; outstr+' ' dumstr; sutstr+' ' dumstr2); onvert Volt(setpt[index]); begin outstr+' ' dumstr2); tem:tem+read temp[Tcontrc fem:tem+read temp[Tcontrc</pre></td>	<pre>outstr+' +dumstr; ndex]:?.4,dumstr; ndex]:?.4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' 'dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' 'dumstr; soutstr+' 'dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); putstr+' 'dumstr2); findex]:3:0,dumstr2); outstr+' 'dumstr2); power[point]:=read_heater_power[point]:=re</pre>	<pre>outstr+' 'dumstr; ndex]:?:4,dumstr; ndex]:?:4,dumstr; outstr+' 'dumstr; outstr+' 'dumstr; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; for loop:=l to average_readi for loop:=l to average_readi f</pre>	<pre>ultstr+' ^dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; ulestr+' ^dumstr; ulestr+' ^dumstr; nvert_volt=ecpt[index]); for loop:=l to average_readir for loop</pre>	<pre>outstr+' '+dumstr; dex]:7:4,dumstr; ndex]:7:4,dumstr; curr:=0.0; ulstr+' '+dumstr; outstr+' '+dumstr2); putstr*' '+dumstr2); rest_dumstr2); rest_dumstr2); rest_dumstr2); rest_dumstr2); rest_dumstr2); rest_dumstr2); nutstr+' '+dumstr2); rest_</pre>	<pre>utstr* ' +dumstr; utstr* ' +dumstr; dex]: 7:4, dumstr; utstr* ' +dumstr; utstr* ' +dumstr; utstr* ' +dumstr2); ric:2, dumstr2); ric:2, dumstr2);</pre>	<pre>outstr+' +dumstr; .ndex]:7:4,dumstr; .ndex]:7:4,dumstr; outstr+' +dumstr; .outstr+' +dumstr; .outstr+' 'dumstr2); .outstr+' 'dumstr2; .outstr+' 'dumstr2;</pre>	<pre>outstr+' + dumstr; outstr+' + dumstr; ndex]:?.4, dumstr; outstr+' + dumstr; outstr+' + dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' ' dumstr; outstr+' ' dumstr; ndex]:7:4,dumstr; outstr+' ' dumstr; outstr+' ' dumstr; sutstr+' ' dumstr2); onvert Volt(setpt[index]); begin outstr+' ' dumstr2); tem:tem+read temp[Tcontrc fem:tem+read temp[Tcontrc</pre>
<pre>utstr+' 'dumstr; dex]??4,dumstr; dex]??4,dumstr; ode voltage conversion************************************</pre>	<pre>setpnt:=0.0; setpnt:=0.0; lest:/ 'dumstr; bex]??4,dumstr; bevltage conversion************************************</pre>	<pre>utstr+''+dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; dex]??4,dumstr; dex]??4,dumstr; ode voltage conversion************************************</pre>										
 | <pre>volt:=0.0;
volt:=0.0;
udex]:?:4,dumstr;
adex]:?:4,dumstr;
iddex]:?:4,dumstr;
volt:=0.0;
ilode voltage conversion************************************</pre>
 | <pre>detpnt:=0.0;
detpnt:=0.0;
ndex]:7:4,dumstr;
ndex]:7:4,dumstr;
outstr+ '+dumstr;
iode voltage conversion************************************</pre> | <pre>under:'' 'dumatr;
under:'' 'dumatr;
under:'' 'dumatr;
under:'' 'dumatr;
under:''' dumatr;
under:''''dumatr2);
y:6:2,dumatr2);
y:6:2,dumatr2);
y:6:2,dumatr2);
under:''''dumatr2);
under:''''dumatr2);
under:'''''dumatr2);
under:''''''dumatr2);
under:''''''''''''''''''''''''''''''''''''</pre> | <pre>ndex]:7:4,dumstr;
outstr+'
'dumstr;
ndex]:7:4,dumstr;
ndex]:7:4,dumstr;
ndex]:7:4,dumstr;
ndex]:7:4,dumstr;
nutstr+' 'dumstr2);
y:6:2,dumstr2);
nutstr+' 'dumstr2);
nutstr+' 'dumstr2);
nutstr+' 'dumstr2);
nutstr+' 'dumstr2);
nutstr+' 'dumstr2);
nutstr+' 'dumstr2);
nutstr+' 'dumstr2);
nutstr+' 'dumstr2);</pre> | <pre>untstr*' *dumstr;
dex]:7:4,dumstr;
dex]:7:4,dumstr;
untstr*' *dumstr;
untstr*' *dumstr;
untstr*' *dumstr];
fof:2,dumstr2);
untstr*' *dumstr2);
untstr*' *dumstr2);</pre> | <pre>dexj:?:4.dumstr;
dexj:?:4.dumstr;
dexj:?:4.dumstr;
dexj:?:4.dumstr;
butstr+`'+dumstr;
ude voltage conversion************************************</pre> | <pre>recent:://dematr;
outstr+'
*dumatr;
dets]?:4,dumatr;
udet]?:4,dumatr;
udet'?:4,dumatr;
outstr+' *dumatr;
nv:=0.0;
volt:=0.0;
nvert_volt(setpt[index]);
res_dumatr2);
nutstr*' *dumatr2;
nutstr*' *dumatr2;
utstr*' *dumatr2;
utstr*' *dumatr2;
utstr*' *dumatr2;
utstr*' *dumatr2;
index]:4:1,dumstr2);
end; [loop]</pre> | <pre>outstr+' ' + dumstr;
outstr+' ' + dumstr;
ndex! i? 4, dumstr;
outstr+' ' + dumstr;
iode voltage conversion************************************</pre> | <pre>index]:?:4,dumstr;
outstr+' 'dumstr;
ndex]:?:4,dumstr;
outstr+' 'dumstr;
outstr+' 'dumstr;
outstr+' 'dumstr;
ide conversion************************************</pre> | <pre>number://dumetr/
outstr/'/dumetr/
ndex]:7:4/dumetr/
dex]:7:4/dumetr/
outstr+' +dumetr/
sutstr+' +dumetr/
outstr+' /dumetr2);
for loop:=1 to average_read/
pristr(index]);
begin temrteatteatteatteatteatteatteatteatteattea</pre>
 |
<pre>currently utetr+' 'dumetry dex]:?:j.dumetry dex]:?:dumetry utetr+' 'dumetry ode voltage conversion************************************</pre>	<pre>acwi:v:v; acwi:v:dumatr; istr:' 'dumatr; istr:' 'dumatr; devoltage conversion************************************</pre>	<pre>dex1:/:3.dumstr); utstr:*' 'dumstr); dex1:7.4.dumstr); utstr:*' 'dumstr); ode voltage conversion************************************</pre>	<pre>dem::r:, dumerr; utetr+ ' +dumerr; utetr+ ' +dumerr; ode voltage conversion************************************</pre>	<pre>ndex1::/:j.gummerr;; outetr:/ 'dummerr; ndex1:?://ummerr; outetr:/ 'dummerr; outetr:/ 'dummerr; iode voltage conversion************************************</pre>	<pre>ndex1::':, dumatr); outstr+' ' dumatr; ndex1: ' + dumatr; outstr+' ' + dumatr; outstr+' ' + dumatr; iode voltage conversion************************************</pre>	<pre>ndex[:::</pre>	<pre>ndex1::1.3,dumstr); outstr+' +dumstr; ddat17:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; for loop:=1 to average_readi for loo</pre>	<pre>ndex1:::,dumatr; outst:+ '+dumatr; ddex1:*,dumatr; ddex1:*,dumatr; ndex1::,dumatr; ndex1::-0.0; nuetr-Yolt(setpt[index]); pi6:2,dumatr2); pi6:2,dumatr2); nutst:+ '+dumatr2); nutst:+ '+dumatr2); nutst:+</pre>	<pre>ndex1::1.3.oumstr; outstr:''-dumstr; ddex1:*.'.dumstr; ddevoltage conversion************************************</pre>	<pre>dexj:::</pre>	<pre>ndex1:/:s,dumstr); outstr/ 'dumstr; ndex1:?t./dumstr; cutr=0.0; volt=0.0; iode voltage conversion************************************</pre>	<pre>nderl://</pre>	<pre>ndex1:r13,dumstr); setEnt=0.0; adex1:r13,dumstr; adex1:r4,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; soutest-' +dumstr; adex1:r4,dumstr; butestr' +dumstr; adex1:r4,dumstr; butestr' +dumstr; begin contect_outcetrc temstr2; temstr2; temstr2; </pre>
<pre>currently dex]:?:j.dumatr); dex]:?:dumatr; dex]:?:dumatr; dex]:rel0.0; utetr+' 'dumatr; ode voltage conversion************************************</pre>	<pre>acw1:*':'dumatr; isetpnt:=0.0; isetpnt:=0.0; isetpnt:=0.0; de voltage conversion************************************</pre>	<pre>dex1:/:3.dumstr); utstr:* '+dumstr); dex1:7.4.dumstr); utstr:* '+dumstr); ode voltage conversion************************************</pre>	<pre>des::r:.dumetr; utetr+ '+dumetr; utetr+ '+dumetr; des::r:.dumetr); utetr+ '+dumetr; de voltage conversion************************************</pre>	<pre>ndex1::/:j.gummerr;; outetr:/ 'dummerr; ndex1:?://ummerr; outetr:/ 'dummerr; outetr:/ 'dummerr; iode voltage conversion************************************</pre>	<pre>ndex1::':, dumatr); outstr+ ' +dumatr; ndex1: ' +dumatr; outstr+ ' +dumatr; outstr+ ' +dumatr; iode voltage conversion************************************</pre>	<pre>ndex[:::</pre>	<pre>ndex1::1.3,dumstr); outstr+' +dumstr; ddat17:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; for loop:=1 to average_readi for loo</pre>	<pre>ndex1:::,dumatr; outst:+ '+dumatr; dex1:*,dumatr; dex1::*,dumatr; ndex1::*,dumatr; nuert_volt(setpt[index]); pi6:2,dumatr2); pi6:2,dumatr2); pi6:2,dumatr2); rem:rtem+read_temp(Tcontro) curr:-curritead_current(Inv volt:-volt+read_nv(nVmeter.dev) end; [lop]</pre>	<pre>ndex1::1.3.oumstr; outstr:''-dumstr; ddex1:*.'.dumstr; ddevoltage conversion************************************</pre>	<pre>dexj:::</pre>	<pre>ndex1:/:s,dumstr); outstr/ 'dumstr; ndex1:?t./dumstr; cutr=0.0; volt=0.0; iode voltage conversion************************************</pre>	<pre>nderl://.j.gummetr; outstr/ ' dummetr; outstr/ ' dummetr; nderl:??i.gummetr; outstr/ ' dummetr; iode voltage convertion************************************</pre>	<pre>ndex1:r13,dumstr); setter1:0; ndex1:r13,dumstr; ndex1:r4,dumstr; outstr+' +dumstr; outstr+' +dumstr; ndex1:r4,dumstr; nd</pre>
<pre>dex]:7(3,dumstr); tem='u'u; dex]:7(4,dumstr) dex]:7(4,dumstr); text+' '+dumstr); utstr+' '+dumstr); de voltage conversion************************************</pre>	<pre>dex1:713.dumstr; lettr+' +dumstr; lettr+' +dumstr; dev2172.4_dumstr; de voltage conversion************************************</pre>	<pre>desl:7:3,dumatr); desl:7:3,dumatr); utstr: ' +dumatr); utstr: ' +dumatr); ode voltage conversion************************************</pre>	<pre>dex1:77.3,dumstr); utstr+ ' +dumstr; dex1:7.4,dumstr); utstr+ ' +dumstr); dev1:7.4,dumstr); dev1:7.4,dumstr); dev1:7.4,dumstr); cutri=0.0; nvert_Volt(setpt[index1]); dev1:0.0; nvert_Volt(setpt[index1]); dev1:2.2,dumstr2); utstr+ ' +dumstr2); utstr+ ' +dumstr2);</pre>	<pre>ndex]:7:3,dumatr); outstr+' +dumatr; outstr+' +dumatr); outstr+' +dumatr); outstr+' +dumatr); nv:=0.0; iode voltage conversion************************************</pre>	<pre>ndex1:7.3.dumatr); outstr+' 'dumatr); ndex1:7.4.dumatr); ndex1:7.4.dumatr); outstr+' 'dumatr); sutstr+' 'dumatr2); y:6:2.dumatr2); y:6:2.dumatr2); rem:ntmantmantmantmantmantmantmantmantmantma</pre>	<pre>ndex]:7.3,dumstr); outstr+' 'dumstr); outstr+' 'dumstr); outstr+' 'dumstr); outstr+' 'dumstr); iode voltage conversion************************************</pre>	<pre>ndex]:7.3,dumstr; ndex]:7.3,dumstr; outstr:' 'dumstr; ndex]?*4,dumstr; ndex]?*4,dumstr; ndex]?*4,dumstr; ndex]?*4,dumstr; ndex]=0.0; nv:=0.0;</pre>	<pre>idex[:1:3,dumstr]; idex[:1:3,dumstr]; idex[:1:4,dumstr]; idex[:1:4,dumstr]; idex[:1:4,dumstr]; idex[:1:4,dumstr]; inter('idex[]); inter('</pre>	<pre>idex]:'1.3, dumstr); idex]:'1.3, dumstr); idex]:'.4, dumstr); idex[:7:4, dumstr); idex[:7:4, dumstr); idex[:7:4, dumstr); idex[:7:4, dumstr]; idex[:7:4, dumstr]; idex[:7:4, dumstr2]; idex[:7:4, dumstr2]; idex[:7:4, dumstr2]; idex[:7:4, dumstr2]; idex[:4:4, dumstr2]; idex[:4:4,</pre>	<pre>idex]:7:3.dumstr); idex]:7:3.dumstr); utstr+''+dumstr; utstr+''+dumstr); utstr+''+dumstr2; idex[:0.0; idex]:2.dumstr2; invert_volt[adex]); if(idex]:2.dumstr2; invert_volt[adex]); if(idex]:2.dumstr2; invert_volt[adex]]; invert[index]]; invert[invert</pre>	<pre>ndex1:7:3, dumatr); outstr+' ' dumatr; outstr+' ' dumatr; outstr+' ' fumatr; volt=0.0; iode voltage conversion************************************</pre>	<pre>ndex1:7:3,dumstr); ndex1:7:4,dumstr; outstr+' '+dumstr; ndex1:7:4,dumstr; ndex1:7:4,dumstr; ndex1:7:4,dumstr; noutstr+' '+dumstr; noutstr; noutstr; noutstr+' '+dumstr2); convert2nt; convert2,; convert2,;</pre>	<pre>ndex1:7:3,dumstr); ndex1:7:4,dumstr; ndex1:7:4,dumstr; ndex1:7:4,dumstr; ndex1:7:4,dumstr; ndex1:7:4,dumstr; ndex1:7:4,dumstr; noutestr; ndex1:7:4,dumstr; nv:=0.0; noutestr; nv:=0.0; nvert_voit(eetpt[index]); begin nvert_voit(mestr2); nuestr2); tem:fem+read temp[Tcontrc nuestr2); </pre>
<pre>dex]:7:3,dumstr); dustr+' + dumstr) dustr+' + dumstr) dustr+' + dumstr); utstr+' + dumstr); utstr+' + dumstr); de voltage conversion************************************</pre>	<pre>dex]:7:3,dumstr; istrt' 'dumstr; istrt' 'dumstr; dev vites convertion************************************</pre>	<pre>dex]:7(3,dumatr); dex]:7(3,dumatr); utstr+' +dumatr) utstr+' +dumatr); utstr+' +dumatr); de voltege conversion************************************</pre>	<pre>dex]:7:3,dumstr); utstr+' +dumstr; utstr+' +dumstr; dex]?f.4,dumstr); utstr+' +dumstr; dev lots; de voltage conversion************************************</pre>										
 | <pre>ndex]:7:3,dumstr;
outstr+' +dumstr;
outstr+' +dumstr;
ndex]:7:4,dumstr;
ndex]:7.4,dumstr);
ndex]:7.4,dumstr);
noter[=0.0;
nv:=0.0;
nv:=0.0;
nv:=0.0;
nv:=0.0;
nv:=0.0;
nv:=0.0;
nv:=0.0;
nv:=0.0;
nv:=0.0;
nv:=1.0;
nv:=1.4,dumstr2);
nutstr+' +dumstr2;
nutstr+' +dumstr2);
power[point]:=read_pain[foint]:=read[foint]</pre>
 | <pre>ndex]:7:3,dumstr;
outstr:* 'dumstr;
ndex]:7:4,dumstr;
outstr:* 'fdumstr;
outstr:* 'fdumstr;
iode voltage conversion************************************</pre> | <pre>ndex]:7:3,dumstr;
ndex]:7:3,dumstr;
outstr+' +dumstr;
ndex]:7:4,dumstr;
ndex]:7.4,dumstr;
ndex]:7.4,dumstr;
ndex[r+' +dumstr]
ndex[r+' +dumstr]
ndex]:2.0,dumstr2);
y:6:2,dumstr2);
nutstr+' +dumstr2);
nutstr+' +dumstr2);
nutstr+' +dumstr2);
nutstr+' +dumstr2);
nutstr+' +dumstr2);
nutstr+' +dumstr2);
nutstr+' +dumstr2);
nutstr+' +dumstr2);
nutstr+' +dumstr2);</pre> | <pre>ndex]:7:3,dumstr;
ndex]:7:3,dumstr;
outstr+'
'dumstr;
ndex]:7:4,dumstr;
ndex]:7:4,dumstr;
ndex]:7:4,dumstr:0;
volt:=0.0;
volt:=0.0;
nv::0:1=0.0;
nv::0:1=0.0;
nv::0:1=0.0;
nv::0:1=0.0;
nv::0:1=0.0;
nv::0:1=0.0;
nv::0:1=0.0;
nv::0:1=0.0;
nv::0:1=0;
nv::0:1=0;
nv::0:1=1:1:dumstr2;
nv::0:1=1:1:dumstr2;
nv::0:1=1:1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu::1=1:dumstr2;
nu:</pre> | <pre>idex]:7:3,dumstr;
idex]:7:4,dumstr;
outstr*' *dumstr;
outstr*' *dumstr;
outstr*' *dumstr;
outstr*' *dumstr;
inter*' *dumstr2);
putstr*' *dumstr2;
nutstr*' *dumstr2;
nutstr*' *dumstr2;
outstr*' *dumstr2;
inter*' *dumstr2;</pre> | <pre>ndex]:?:3,dumstr;
ndex]:?:4,dumstr;
outstr*' *dumstr;
uutstr*' *dumstr;
volt=0.0;
volt=0.0;
nV:=0.0;
nV:=0.0;
nV:=0.0;
nV:=0.0;
nV:=0.0;
nV:=0.0;
rem:=tem+read_temp(Tcontro
tem:=tem+read_temp(Tcontro
tem:=tem+read_temp(Tcontro
tem:=tem+read_volt(Tem
volt:=volt+read_volt(Tem
volt:=volt+read_volt(Tem
volt:=volt+read_volt(Tem
volt:=volt+read_volt(Tem
volt:=volt+read_volt(Tem
volt:=volt+read_volt(Tem
volt:=1,dumstr2);
nutstr+' *dumstr2;
findex]:4:1,dumstr2);</pre> | <pre>idex]:7:3,dumstr;
idex]:7:4,dumstr;
uutstr*
'*dumstr;
uutstr* '*dumstr;
uutstr* '*dumstr;
uutstr* '*dumstr;
oode voltage conversion************************************</pre> | <pre>ndex]:7:3,dumatr); tem:=0.0; setpnt:=0.0; ndex]:7:4,dumatr; outstr+' +dumatr; iode voltage convertion************************************</pre> | <pre>ndex]:7:3,dumstr;</pre> | <pre>ndex]:7:3,dumstr;
outstr+' 'dumstr;
outstr+' 'dumstr;
ndex]:7:4,dumstr;
outstr+' 'dumstr;
outstr+' 'dumstr;
fof voltage conversion************************************</pre>
 |
<pre>dex]:7:3,dumstr; dex]:7:3,dumstr; dutstr*(*dumstr; dutstr*(*dumstr); utstr*(*dumstr); utstr*(*dumstr); dutstr*(*dumstr2); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvertr2); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nvert_Volt(setpt[index]);</pre>	<pre>dex]:7:3,dumstr);</pre>	<pre>dex]:7:3,dumatr); utestr+' +dumatr; utestr+' +dumatr; utestr+' +dumatr; ode voltage conversion************************************</pre>	<pre>dex]:7:3,dumetr]; utetr+' + dumetr]; utetr+' + dumetr; dex]:7:4,dumetr); utetr+' + dumetr]; utetr+' + dumetr]; nv=10.0; nvert Volt(setpt[index]); nvert Volt(setpt[index]); nvert Volt(setpt[index]); nvert Volt(setpt[index]); nvert voltse for loop:=1 to average_reading; nvert voltse for loop:=1 to average_reading; note::2; nvert voltse for loop:=1 to average_reading; note::2; nvert voltse for loop:=1 to average_reading; note::2; nvert voltse for loop:=1 to average_reading; note::2; nvert voltse for loop:=1 to average_reading; note:=2; nvert voltse for loop:=1 to average_reading; note:=2; nvert voltse for loop:=1 to average_reading; nvert vo</pre>	<pre>ndex1:7:3,dumstr; outstr+' +dumstr; outstr+' +dumstr; ndex1:7:4,dumstr); ndex1:7:4,dumstr); ndex1:7:4,dumstr); ndex1:7:4,dumstr); ndex0.0:0; ndex1:7:4,dumstr); ndex1:7:4,dumstr2); ndex1:7:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2); ndex1:4,dumstr2);</pre>	<pre>ndex]:7:3,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; soutstr+' +dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); outstr+' +dumstr2; findex]:3:0,dumstr2); outstr+' +dumstr2; findex]:4:1,dumstr2); outstr+' +dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2); gindex]:4:1,dumstr2);</pre>	<pre>ndex]:7:3,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr:0.0; volt:-0.0; volt:-0.0; volt:-0.0; volt:-0.0; volt:-0.0; volt:-0.0; volt:-0.0; volt:-0.0; volt:-voltread_tem[foontrc curri-current[I volt:-voltread_volt[Vmete nv:-nv+read_volt[Vmete_vol (vmetr2); volt:-voltread_volt[Vmete_vol volt:-voltread_volt[Vmete_vol volt:-voltread_volt[Vmete_vol volt:-voltread_volt[Vmete_vol voltr+' +dumstr2); voltr+' +dumstr2);</pre>	<pre>ndex]:7:3,dumstr; outstr+' 'dumstr; outstr+' 'dumstr; dod voltage conversion************************************</pre>	<pre>idex]:7:3,dumstr; idex]:7:4,dumstr; uutstr+' ^dumstr; uutstr+' ^dumstr; uutstr+' ^dumstr; nutstr+' ^dumstr; inter+' ^dumstr2); putstr+' ^dumstr2; nutstr+' ^dumstr2;</pre>	<pre>ndex]:7:3,dumstr); tem:=0.0; outstr*' *dumstr; ndex]:7:4,dumstr; curr:=0.0; ndex]:7:4,dumstr; ndex]:7:4,dumstr]; volt==0.0; volt==0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=nv:=1+read_temp(Tcontro tem:=tem+read_temp(Tcontro tem:=tem+read_temp(Tcontro tem:=tem+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt(Tmm volt:=-volt+read_volt+read_volt+read_volt(Tmm volt:=-volt+read_volt+read_volt(Tmm volt:=-volt+read_volt+read_volt(Tmm volt:=-volt+read_volt+read_volt+read_volt+read_volt(Tmm volt:=-volt+read_volt+read_volt+read_volt+read_volt+read_volt+read_volt+read_volt(Tmm volt:=-volt+read_volt+r</pre>	<pre>idex]:7:3,dumstr; idex]:7:3,dumstr; utstr+' 'dumstr; idex]:7:4,dumstr; idex]:7:4,dumstr; idex]:7:4,dumstr; idex]:7:4,dumstr; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; for loop idex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2); iddex]:4:1,dumstr2);</pre>	<pre>index]:7:3,dumatr); ioutatr; ioutatr; 'dumatr; ioutatr; 'dumatr; ioutatr; 'dumatr; ioutatr; 'dumatr; ioutatr; 'dumatr; ioutatr; 'dumatr]; ioutatr; 'dumatr2); ioutatr; 'dumatr; 'dumatr2); ioutatr; 'dumatr2); ioutatr; 'duma</pre>	<pre>ndex]:7:3,dumstr; tem:=0.0; tem:=0.0; actpnt:=0.0; a</pre>	<pre>ndex]:7:3,dumstr; tem:=0.0; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; surr:=0.0; outstr+' +dumstr; for loop:=1 to average_read; for loop:=1 to average_read; putstr, dumstr2); outstr+' +dumstr2);</pre>
<pre>dex]:7:3,dumatr); utatr+' +dumatr; utatr+' +dumatr; utatr+' +dumatr; ode voltage conversion************************************</pre>	<pre>dex]:7:3,dumatr); itstr+' +dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; devoltage conversion************************************</pre>	<pre>dex]:7:3,dumstr); utstr+' '+dumstr); utstr+' '+dumstr); dex]:7:4,dumstr); dex]:7:4,dumstr); utstr+' +dumstr); ode voltage conversion************************************</pre>	<pre>dex]:7:3,dumtr); utstr+' 'dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; utstr+' 'dumstr; nvet Voltge conversion************************************</pre>	<pre>index]:?:3,dumstr); coutstr+' '+dumstr); coutstr+' '+dumstr); coutstr+' '+dumstr); coutstr+' '+dumstr); coutstr+' '+dumstr2); wy:6:2,dumstr2); wy:6:2,dumstr2); coutstr+' '+dumstr2); coutstr+' '+dumstr2); coutstr+' '+dumstr2); coutstr+' '+dumstr2); coutstr+' '+dumstr2); power[point]:=read_parter_point]:=read_part[ront]; coutstr+' '+dumstr2); power[point]:=read_parter_point]:=read_part[ront]; coutstr+' '+dumstr2); power[point]:=read_parter_point]:=read_part[ront]; coutstr+' '+dumstr2); power[point]:=read_parter_point]:=read_part[ront]; coutstr+' '+dumstr2);</pre>	<pre>ndex]:7:3,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; ndex]:7:4,dumstr; ndex]:7.4,dumstr; outstr+' +dumstr; outstr+' +dumstr2; sovert_Volt(setpt[index]); y:62,dumstr2); outstr+' +dumstr2; soutstr+' +dumstr2; outstr+' +dumstr2; outstr+'</pre>	<pre>ndex]:7:3,dumstr; outstr+' +dumstr; outstr+' +dumstr; outstr+' +dumstr; ndex]:7.4,dumstr; outstr+' +dumstr; outstr+' +dumstr2); y:6:2,dumstr2); outstr+' +dumstr2); outstr+' +dumstr2); findex]:4.1,dumstr2); outstr+' +dumstr2); outstr+' +dumstr2); outstr+' +dumstr2); outstr+' +dumstr2);</pre>	<pre>ndex]:7:3,dumatr); outstr+' +dumatr; outstr+' +dumatr; ndex]:7:4,dumatr; ioutstr+' +dumatr; iode voltage conversion************************************</pre>	<pre>idex]:7:3,dumstr); idex]:7:4,dumstr); butstr+' +dumstr); idex]:7:4,dumstr); butstr+' +dumstr); inv:=0.0; butstr+' +dumstr2); if if index]); if isit,dumstr2); if index]:3:0,dumstr2); inv:=urstr==================================</pre>	<pre>ndex]:7:3,dumstr; utstr+' '-dumstr; utstr+' '-dumstr; utstr+' '-dumstr; utstr+' '-dumstr; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=volt+read_temp(Tcontro tem:=temtread_temp(Tcontro tem:=temtread_temp(Tcontro towstr2); volt:=volt+read_v(n%meter.dev. (index]:4:1,dumstr2); voltstr+' '-dumstr2); (index]:4:1,dumstr2); voltstr+' '-dumstr2); voltstr+' '-dumstr2); volts</pre>	<pre>idex]:7:3,dumstr); idex]:7:3,dumstr); utstr+' '+dumstr); idex]:7:4,dumstr); idex]:2.4,dumstr); idex of tage conversion************************************</pre>	<pre>.ndex1:7:3,dumstr); .outstr+' '+dumstr; .outstr+' '+dumstr); .outstr+' '+dumstr); .outstr+' '+dumstr); .outstr+' '+dumstr); .outstr+' '+dumstr2); .outstr+' '+dumstr2); .outstr+' '+dumstr2);</pre>	<pre>ndex]:7:3,dumatr; outstr+' 'dumatr; outstr+' 'dumatr; ndex]:7:4,dumatr; outstr+' 'dumatr; outstr+' 'dumatr; orvert_Volt(setpt[index]); y:6:2,dumatr2); outstr+' 'dumatr2;</pre>	<pre>ndex]:7:3,dumstr); ndex]:7:4,dumstr; outstr+` +dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+` +dumstr; potestr+` +dumstr; nvert_Volt(setpt[index]); povert_Volt(setpt[index]); potestr+` +dumstr2); putstr+` +dumstr2);</pre>
<pre>dex]:7:3,dumatr; utatr+' +dumatr; utatr+' +dumatr; utatr+' +dumatr; out volt=0.0; utatr+' +dumatr); utatr+' +dumatr2); volt=0.0; volt=0.0; volt=0.0; volt=0.0; volt=read_temp(Tcont tem:r=eurr+read_temp(Tcont volt=rvoit+read_volt(Vmeeter.c index]:3:0,dumatr2); utatr+' +dumatr2; utatr+' +dumatr2; utatr+' +dumatr2; utatr+' +dumatr2; utatr+' +dumatr2; utatr+' +dumatr2; utatr+' +dumatr2;</pre>	<pre>dex::7:3,dumstr; itstr+' +dumstr; dex::7:4,dumstr; dex::7:4,dumstr; dex:0.0; det voltage conversion************************************</pre>	<pre>dex]:7:3,dumstr); utstr+' '+dumstr); utstr+' '+dumstr); dex]:7:4,dumstr); utstr+' '+dumstrs; ode volt=0.0; volt=0.0; volt=0.0; vv:=0.0; nv:=0.0; for loop:=1 to average_readi nv:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi fo</pre>	<pre>des1:7:3,dumatr; utstr+' +dumatr; des1:7:4,dumatr; des1:7:4,dumatr; des1:7:4,dumatr); utstr+' +dumatr; nv:=0.0; nv:=0.0; nv:=0.0; nv:=0.0; nv:=1:0,0; for loop:=1 to average_readi for lo</pre>	<pre>index1:7:3,dumstr; coutstr+ '+dumstr; coutstr+ '+dumstr; coutstr+ '+dumstr; coutstr+ '+dumstr; coutstr+ '+dumstr; conversion************************************</pre>	<pre>ndex1:7:3,dumatr; ndex1:7:3,dumatr; outstr+'+dumatr; ndex1:7:4,dumatr; ndex1:7:4,dumatr; ndex1:7:4,dumatr; outstr+'+dumatr; ndex1:2:4,dumatr2); soutstr+'+dumatr2); outstr+'+dumatr2); outstr+'+dumatr2); ndex1:3:0,dumatr2); outstr+'+'dumatr2); findex1:3:0,dumatr2); outstr+'+'dumatr2); findex1:4:1,dumatr2); outstr+'+'dumatr2); findex1:4:1,dumatr2); findex1:4:1,dumatr2); outstr+'+'dumatr2); findex1:4:1,dumatr2); findex1:4:1,dumatr2); findex1:4:1,dumatr2); findex1:4:1,dumatr2); findex1:4:1,dumatr2); findex1:4:1,dumatr2);</pre>	<pre>ndex[:7:3,dumstr); ndex[:7:4,dumstr); outstr+ '+dumstr); ndex[:7:4,dumstr); outstr+ '+dumstr); ndex[:7:4,dumstr); ndex[:7:4,dumstr]; node voltage conversion************************************</pre>	<pre>idex]:7:3,dumstr; idex]:7:4,dumstr; outstr+' +dumstr; idex]:7:4,dumstr; idex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr2; inv:=0.0; ifor loop:=1 to average_readin pount:** inv:=0.0; ifor loop:=1 to average_readin inv:=0.0; ifor loop:=1 to average_readin inv:=0.0; ifor loop:=1 to average_readin inv:=0.0; ifor loop:=1 to average_readin inv:=0.0; ifor loop:=1:4:1,dumstr2; outstr+' +dumstr2; ifor loop:=1:4:1,dumstr2; ifor lo</pre>	<pre>dex]:7:3,dumstr; udex]:7:4,dumstr; udex]:7:4,dumstr; udex]:7:4,dumstr; udex[:7:4,dumstr; udex[:7:4,dumstr]; udex[r*, *dumstr]; ior loop:=1 to average_readir for loop:=1 to average_r</pre>	<pre>dex]:7:3,dumstr; dex]:7:3,dumstr; uutstr+ '+dumstr; uutstr+ '+dumstr; dev2:1:24,dumstr; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; for loop:=1 to average_readi nv:=0.0; for loop:=1 to average_readi nv:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi volt:=-0: volt:=-0: volt:=volt+read_volt(Immetr) volt:=rvolt+read_volt(Immetr.dev volt:=rvolt+read_volt(Immetr.dev volt:=rvolt+read_volt(Vmmetr.dev volt:=rvolt+read_volt(Vmmetr.dev volt:=volt+read_volt(Vmmetr.dev volt:=volt+read_volt(Vmmetr.dev volt:=volt+read_volt(Vmmetr.dev volt:=volt+read_volt(vmmetr.dev volt:=volt+read_volt(vmmetr.dev volt:=volt+read_volt+read_volt(Vmmetr.dev volt:=volt+read_volt(Vmmetr.dev volt:=volt+read_volt+read_volt+read_volt+read_volt(Vmmetr.dev volt:=volt+read_volt(vmmetr.dev volt:=volt+read_volt(vmmetr.dev volt:=volt+read_volt+read_volt(vmmetr.dev volt:=volt+read_volt(vmmetr.dev volt:=volt+read_volt+read_volt+read_volt(vmmetr.dev volt:=volt+read_volt+read_volt+read_volt+read_volt(vmmetr.dev volt:=volt+read_volt+read_volt+read_volt+read_volt+read_volt(vmmetr.dev volt:=volt+read_volt+re</pre>	<pre>dex]:7:3,dumatr; dex]:7:3,dumatr; udex]:7:4,dumatr; udex]:7:4,dumatr; udex]:7:4,dumatr; iode voltage conversion************************************</pre>	<pre>index1:7:3,dumstr); index1:7:3,dumstr); cutstr+' 'dumstr); indextr+' 'dumstr); cutstr+' 'dumstr); indextr+' 'dumstr); indextr+' 'dumstr); indextr+' 'dumstr2); vutstr+' 'dumstr2); cutstr+' 'dumstr2);</pre>	<pre>ndes]:7:3,dumstr; ndes]:7:3,dumstr; outstr+' +dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; iode voltage conversion************************************</pre>	<pre>idex1:7:3,dumstr); idex1:7:3,dumstr); idex1:7:4,dumstr; outstr+`+dumstr; idex1:7:4,dumstr; outstr+`+dumstr; outstr+`+dumstr; idex0:fill(idex1); idex1:fill(idex1); idex1:fill(idex1</pre>
<pre>dex]:7:3,dumstr; tem:=0.0; utstr+' +dumstr; dex]?7:4,dumstr; utstr+' +dumstr; utstr+' +dumstr; ode voltage conversion************************************</pre>	<pre>dex[17:3, dumatr]; itstr+' '+dumatr]; dex]:7:4,dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dex[7:4,dumatr]; det[=0.0; det[=0.0; det[=0.0; det[=0.0; det[=0.0]; rv::===================================</pre>	<pre>desl:7:3 dumatr): utstr+ '+dumatr); utstr+ '+dumatr); dex]:7:4,dumatr); utstr+ '+dumatr); utstr+ '+dumatr); utstr+ '+dumatr2); for loop:=1 to average_readi. nver volt=0.0; nver volt=0.0; for loop:=1 to average_readi. for loop:=1 to average_r</pre>	<pre>decs::7:3,dumatr; utstr+' +dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dev1:7:4,dumatr; utstr+' +dumatr; nv:=0.0; ode voltage conversion************************************</pre>	<pre>current. index1;7:3, dumstr); outstr+ '+dumstr); outstr+ '+dumstr; outstr+ '+dumstr; index1;7:4, dumstr); outstr+ '+dumstr; index1;7:4, dumstr); index1;7:4, dumstr2; inv:6:2, dumstr2); outstr+ '+dumstr2; index1;3:0, dumstr2); outstr+ '+dumstr2; index1;4:1, dumstr2); power[point]:=read_pain[point]:=read_pain[foint]:=read[foint]:=read[foint]:=read[foint]:=read[foint]:=read[foint]:=read[foint]:=read[foint]:=read[foint]:=</pre>	<pre>index[:r:3, dumatr); outstr+' + dumatr); outstr+' + dumatr); ndex[:r:4, dumatr); ndex[:r:4, dumatr]; ndex[:r:4, dumatr]; outstr+' + dumatr]; iode voltage conversion************************************</pre>	<pre>idex[:7.3, dumstr); idex[:7.3, dumstr); outstr+ '+dumstr); iddex[:7.4, dumstr]; outstr+ '+dumstr; idde voltage conversion************************************</pre>	<pre></pre>	<pre>idex]:7:, dumstr); idex]:7:, dumstr); idex]:7:4,dumstr); idex]:7:4,dumstr); idex]:7:4,dumstr); idex]:7:4,dumstr); idex]:7:4,dumstr); intextr' 'dumstr); intextr' 'dumstr2); intextr' 'dumstr2); intextr'</pre>	<pre>dust:// 'dumst:); dust:// 'dumst:); dex]:?:4,dumst:); dex]:?:4,dumst:); dex]:?:4,dumst:); ulest:// 'dumst:); ulest:// 'dumst:2); putst:// 'du</pre>	<pre>dex]:7:3,dumstr); udex]:7:3,dumstr); udex]:7:4,dumstr); udex]:7:4,dumstr); udex[:7:4,dumstr]; udex[:7:4,dumstr]; udex[:7:4,dumstr]; udex[:2:2,dumstr2]; r=6:2,dumstr2]; utextr+' *dumstr2]; utextr+' *dumstr2]; utext+' *dumst</pre>	<pre></pre>	<pre>dest::7:3, dumatr); ndex1:7:3, dumatr); outetr+' '+dumatr); ndex1:?4, dumatr); outetr+' '+dumatr); outetr+' '+dumatr); outetr+' '+dumatr); y:62, dumatr2; outetr+' '+dumatr2;</pre>	<pre></pre>
<pre>dex]:7:3.dumstr; dex]:7:4.dumstr; tem:=0.0; dex]i7.4.dumstr; dex]i7.4.dumstr; dex]i7.4.dumstr; dex[i7.4.dumstr]; curr:=0.0; volt:=0.0; volt:=0.0; volt=0.0; for loop: hegin tem:=temtread_temp(Tcont tem:=temtread_temp(Tcont utetr+' +dumstr2); utetr+' +dumstr2; utetr+' +dumstr2; volt:=volt+read_volt(Wme volt:=volt+read_volt(Wme volt:=volt+read_volt(Wme volt:=rool, loop); volt:=rool, dumstr2; volt:=rool, volt:=rool, volt:=rool, volt: volt:=rool, volt:=rool, volt volt:=rool, volt volt:=rool, volt volt:=rool, volt volt.=rool, volt volt.=rool, volt volt</pre>	<pre>dex[:::</pre>	<pre>dex1:7:3,dumatr; utstr:+ '+dumatr; dex1:7:3,dumatr); utstr:+ '+dumatr; dex1:7:3,dumatr); utstr:+ '+dumatr; wutstr:+ '+dumatr2); utstr:+ '+dum</pre>	<pre>uter:=0.0; uter:=************************************</pre>	<pre></pre>	<pre>cutatry * dumatr); ndex]:?.3 dumatr); ndex]:?.4 dumatr); ndex]:?.4 dumatr); ndex]:?.4 dumatr); ndex]:?.4 dumatr); outatr+ * fdumatr; iode voltage conversion************************************</pre>	<pre>outstry founstry dex]:7:4,dumstry outstry' founstry adex]:7:4,dumstry outstry' founstry adex]:7:4,dumstry outstry' founstry soutstry' founstry iode voltage conversion************************************</pre>	<pre>cutatr: 'dumatr; ndex]:7.3,dumatr; ndex]:7.3,dumatr; ndex]:7.4,dumatr; ndex]:7.4,dumatr; nv:=0.0; sutr:=0.0; volt:=0.0; volt==0.0; volt==0.0; volt==0.0; power[point]:=read heater po outatr: ' 'dumatr2; sutr:=read heater po power[point]:=read heater po power[point]:=read heater po</pre>	<pre>dex1:7:3,dumstr); dex1:7:3,dumstr); dex1:7:3,dumstr); dex1:7:3,dumstr); dex1:7:3,dumstr); dex1:7:3,dumstr); utstr+' +dumstr; note voltage conversion************************************</pre>	<pre>dex1:7:4 dumstr; dex1:7:3 dumstr); dex1:7:4 dumstr; dex1:7:4 dumstr; dex1:7:4 dumstr; butstr+ '+dumstr; butstr+' '+dumstr; sutstr+' '+dumstr2); pr6:2 dumstr2); pr6:2 dum</pre>	<pre>dex]:7:3,dumstr; dex]:7:4,dumstr); dex]:7:4,dumstr); butstr+' 'dumstr; dex]:7:4,dumstr); butstr+' 'dumstr]; nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); putstr+' 'dumstr2); nutstr+' 'dumstr2); findex]:3:0,dumstr2); mvert*' 'dumstr2); nutstr+' 'dumstr2); mvert*' 'dumstr2); mvert*'dumstr2); mve</pre>	<pre>outstr:/ foumstr); .ndex]:7:3,dumstr); outstr:/ (dumstr); .ndex]:7:4,dumstr; .ndex]:7:4,dumstr; .ndex]:7:4,dumstr); outstr:/ 'dumstr); .outstr:/ 'dumstr2); .outstr:/ 'dumstr2); .outstr:/ 'dumstr2);</pre>	<pre>cutatr: roumstr; outstr: roumstr; outstr: 'dumstr; ndex]:?.4,dumstr; ndex]:?.4,dumstr; outstr: 'dumstr; outstr: 'dumstr; yode voltage conversion************************************</pre>	<pre>outstr: voumstr; dex1:7:3,dumstr); utex1:*' *dumstr); dex1:7:4,dumstr; outstr*' *dumstr; outstr*' *dumstr; sutstr*' *dumstr; for loop:=1 to average_read; for loop:=1 to average_read; potstr,dumstr2); begin temstr2</pre>
<pre>dex]:7:3.dumstr; dex]:7:4.dumstr; dex]?7:4.dumstr; dex]?7:4.dumstr; dex]?7:4.dumstr; dex]?7:4.dumstr; dex]?7:4.dumstr; curr=0.0; dev0ltage conversion************************************</pre>	<pre>curried.0; dex]:1:3.dumstr; dex]:1:4.dumstr; dex]:7:4.dumstr; dex]:7:4.dumstr; dex]:7:4.dumstr; dex]:7:4.dumstr; dex]:1:4.dumstr; dex]:1:4.dumstr; dex[:2.dumstr2]; fordex]:3:0.dumstr2; fordex]:3:0.dumstr2; fordex]:4:1.dumstr2; fordex]:</pre>	<pre>utstr: ' dumstr); utstr: ' dumstr); utstr: ' dumstr); utstr: ' dumstr); utstr: ' dumstr); utstr: ' dumstr); utstr: ' dumstr2); utstr: ' dumst</pre>	<pre>uter:-0.0; dex]:7:3,dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; iten:-0.0; utetr+' 'dumatr; nv=r_0.0; nv=r_volt=0.0; nv=r_volt=0.0; for loop:=1 to average_readi for loop:=1 to averag</pre>	<pre>coulds:??.?.dumatr; index]??.?.dumatr); outstr+' *dumatr; outstr+' *dumatr; outstr+' *dumatr; index]??.4,dumatr; outstr+' *dumatr; index]??.4,dumatr; index]??.4,dumatr; index]??.4,dumatr2); index[solution]); convert Volt(setpt[index]); index[solution]); convert Volt(setpt[index]); index[solution]; convert Volt(setpt[index]); convert Volt(setpt[index]); conver</pre>	<pre>crutic::::::::::::::::::::::::::::::::::::</pre>	<pre>outstr+ '+dumstr; ndex]:?;4,dumstr); setpnt=0.0; ndex]:?;4,dumstr); outstr+' '+dumstr; outstr+' '+dumstr; outstr+' '+dumstr2); soutstr+' '+dumstr2); rem:=temtread_temp(fcontre tem:=cutr+read_out[tymete_ outstr+' '+dumstr2); soutstr+' '+dumstr2); findex]:4:,dumstr2); soutstr+' '+dumstr2); power[point]:=read_heate_pc; outstr+' '+dumstr2);</pre>	<pre>cvutatry + dumatry ndex]:7:4,dumatry ndex]:7:4,dumatry ndex]:7:4,dumatry ndex]:7:4,dumatry ndex]:7:4,dumatry ndet[:7:4,dumatry ndet[:0:0; nutatry + dumatr2); nutatry + dumatr2);</pre>	<pre>utatry roumstry dex]:7:3,dumstry outstry' dumstry utatry' dumstry outstry' dumstry nutatry' dumstry fof:2,dumstry routetry' dumstr2); sutstry' dumstr2); nutatry' dumstr2); sutstry' dumstr2); sutstry</pre>	<pre>cverts dex]:7:4,dumstr; dex]:7:4,dumstr; deta1:7:3,dumstr); setpnt:=0.0; udex]:7:4,dumstr; butstr+''+dumstr; outetr+''+dumstr2); putstr+''+dumstr2); putstr+''+dumstr2); putstr+''+dumstr2); putstr+''+dumstr2); findex]:4:1,dumstr2); cutstr+''+dumstr2); mvientr2); findex]:4:1,dumstr2); end; [loop]</pre>	<pre>utstr+ '+dumstr); dex]:7:3,dumstr); utstr+' '+dumstr); utstr+' '+dumstr); utstr+' '+dumstr); utstr+' '+dumstr2; nvert_Volt(setpt[index]); nvert_Volt(setpt[index]); nuetr+' '+dumstr2; rem:=tem+read_temp(fcontrol curr==curr+read_temp(fcontrol curr==curr+read_volt(Nmeter.devi nutstr+' '+dumstr2); nutstr+' '+dumstr2); utstr+' '+dumstr2);</pre>	<pre>outstr: foumstr); .ndex]:7:3,dumstr); outstr: 'fdumstr); .ndex]:7:4,dumstr); .ndex]:7:4,dumstr); .ndex[:7:4,dumstr); .ndex[:7:4,dumstr]; .outstr: 'fdumstr2); .outstr: 'fdumstr2); .outstr: 'fdumstr2); .outstr: 'fdumstr2;</pre>	<pre>outstr+ ' + dumstr; ddex]:7:4, dumstr); setEnt=0.0; outstr+' ' + dumstr; outstr+' ' + dumstr; idde voltage conversion************************************</pre>	<pre>outstr+ * toumstr; dex1:7:3,dumstr); ndex1:7:3,dumstr); ndex1:7:4,dumstr); dex1:7:4,dumstr; outstr+ * +dumstr; sutstr+ * +dumstr; outstr+ * +dumstr2); for loop:=1 to average_readi for loop:=1 to average_readi for loop:=1 to average_readi begin temstr2);</pre>
<pre>dex]:7:3,dumstr; dex]:7:3,dumstr; utstr+' '+dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; dex]:0.0; utstr+' '+dumstr; ode voltage conversion************************************</pre>	<pre>ristr+' 'dumstr; bes:)?:4,dumstr; lstr+' 'dumstr; lstr+' 'dumstr; retr=0.0; retr=</pre>	<pre>utetr+' 'dumetr; dex]:7:3,dumetr); dex]:7:4,dumetr); utetr+' 'dumetr); utetr+' 'dumetr); utetr+' 'dumetr2); ode voltage conversion************************************</pre>	<pre>utetr+' 'dumetr; dex:]7:3,dumetr; dex:]7:3,dumetr); utetr+' 'dumetr; utetr+' 'dumetr); utetr+' 'dumetr); ode voltage conversion************************************</pre>	<pre>outstr:/ 'dumstr: index1:7:,dumstr): coutstr:/ 'dumstr): outstr:/ 'dumstr: noutstr:/ 'dumstr: coutstr:/ 'dumstr: iode voltage conversion************************************</pre>	<pre>outstr+' 'dumstr; ndex]:7:3,dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr); outstr+' 'dumstr2); ndex]); suttr+' 'dumstr2); nutstr+' 'dumstr2); outstr+' 'dumstr2); findex]:3:0,dumstr2); nutstr+' 'dumstr2); nutstr+' 'dumstr2); nu</pre>	<pre>outstr+' ' dumstr; ndex[:7:3, dumstr); ndex[:7:4, dumstr); ndex[:7:4, dumstr); outstr+' ' dumstr); outstr+' ' dumstr2); outstr+' ' dumstr2); y:6:2, dumstr2); potet[index]]:0, dumstr2); outstr+' ' dumstr2); findex]:3:0, dumstr2); outstr+' ' dumstr2); findex]:4:1, dumstr2); power[poth]:=read_heater_power[poth]:=read_heater_power[poth]:=read_heater_power[poth]:=read_heater_poth_poth_poth_poth_poth_poth_poth_poth</pre>	<pre>outstr+' ' dumstr; dex]:7:4 dumstr); ndex]:7:4 dumstr); setpn:=0.0; outstr+' ' dumstr) outstr+' ' dumstr) outstr+' ' dumstr2); outstr+' ' dumstr2); setpn:==0.0; volt:=0.0; volt:=0.0; volt:=emplemented_emplement(Im volt:=volt+read_volt(Meeter dev outstr+' ' dumstr2); outstr+' ' dumstr2); outstr+' ' dumstr2); outstr+' ' dumstr2); outstr+' ' dumstr2); outstr+' ' dumstr2);</pre>	<pre>Degin utatr*' 'dumatr; dex1:7:3,dumatr); utatr*' 'dumatr); dex1:*.dumatr); utatr*' 'dumatr); pi6:2,dumatr2); pi6:2,dumatr</pre>	<pre>ultatr' ' dumatr; dex]:?i.j.dumatr; ndex]:?i.dumatr; ndex]:?i.dumatr; ndex]:?i.dumatr; ndex]:?i.dumatr; udevoltage conversion************************************</pre>	<pre>utstr+' 'dumstr; dex]:7:3,dumstr); utstr+' 'dumstr); dex]:7:4,dumstr); utstr+' 'dumstr) devoltage conversion************************************</pre>	<pre>outstr+' 'dumstr; .ndex]:7:3,dumstr); outstr+' 'dumstr); .outstr+' 'dumstr; .outstr+' 'dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' ' dumstr; ndex]:7:3, dumstr); outstr+' ' dumstr); outstr+' ' dumstr); outstr+' ' dumstr); outstr+' (dumstr); vol:=0.0; nV:=0.0; nV:=0.0; nV:=0.0; nV:=1.00; nV:=1</pre>	<pre>outstr+' ' dumstr; ndex]:7:3,dumstr; setpn:=0.0; outstr+' ' dumstr; outstr+' ' dumstr; outstr+' ' dumstr; soutstr+' ' dumstr]; soutstr+' ' outstr2); outstr+' ' dumstr2); outstr+' ' dumstr2); begin outstr+' ' dumstr2);</pre>
<pre>uter:* ' + dumstr; dex]:7:3, dumstr; dex]:7:3, dumstr; dex]:7:4, dumstr; detx]:7:4, dumstr; detx]:7:4, dumstr; detx]:7:4, dumstr; ode voltage conversion************************************</pre>	<pre>legin dex;:?t.dumstr; dex;:?t.dumstr; itstr+`'+dumstr; itstr+`'+dumstr; itstr+`'+dumstr; de voltage conversion************************************</pre>	<pre>utatr+' '+dumatr; des]:7:3,dumatr); des]:7:4,dumatr); utatr+' +dumatr); utatr+' +dumatr); utatr+' +dumatr); ode voltage conversion************************************</pre>	<pre>utatr+' 'dumatr; dex]:7:3,dumatr); utatr+' 'dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dex]:7:4,dumatr; dex]:4:1,dumatr; nvert_Volt(setpt[index]); dev]:4:1,dumatr2); utetr+' 'dumatr2); utetr+' 'dumatr2); utetr+'' dumatr2); utetr+'' 'dumatr2); utetr+'' 'dumatr2); utetr+'' 'dumatr2);</pre>	<pre>outstr+ ' +dumstr; ndex]:7:3,dumstr); outstr+ ' +dumstr); outstr+ ' +dumstr); outstr+ ' +dumstr); outstr+' +dumstr); outstr+' +dumstr); ionvert_Volt(setpt[index]); y:6:2,dumstr2); y:6:2,dumstr2); outstr+' +dumstr2); outstr+' +dumstr2); outstr+' +dumstr2); outstr+' +dumstr2); findex]:4:1,dumstr2); power[point]:=read_pain[point]:=read_gain[fcont]; outstr+' +dumstr2); gain[point]:=read_gain[fcont]; power[point]:=read_gain[fcont]; power[point]:=read_gain[fcont]; power[point]:=read_gain[fcont]; power[point]:=read_gain[fcont];</pre>	<pre>outstr+' +dumstr; outstr+' +dumstr; ndex]:7:3,dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr); outstr+' +dumstr2); y:6:2,dumstr2); nv:ert_uo; nv:ert=outstread_volt(vete curr:=currteed_temp(fcontro begin begin begin volt:=volt+read_volt(vete_ nv:ert2); nv:ert2); nutstr+' +dumstr2); nutstr+' +dumstr2); nuts</pre>	<pre>outstr+' +dumstr; outstr+' 'dumstr; idex]:7:3,dumstr; outstr+' 'dumstr; outstr+' 'dumstr; outstr+' 'dumstr; ide voltage conversion************************************</pre>	<pre>outatr+ '+dumstr; outatr+' '+dumstr; ndex]:7:3,dumstr; outstr+' '+dumstr; outstr+' '+dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr2; outatr+' '+dumstr2; outetr+' '+dumstr2; outetr+' '+dumstr2; nds]:4:1,dumstr2; outetr+' '+dumstr2; nds]:4:1,dumstr2; outetr+' '+dumstr2; nds]:4:1,dumstr2; outetr+' +dumstr2; nds]:4:1,dumstr2;</pre>	<pre>begin utstr+' 'dumstr; idex]:1:3,dumstr); udex]:7:4,dumstr); udex[:7:4,dumstr); udex[:7:4,dumstr); utstr+' 'dumstr]; utstr+' 'dumstr2); putstr+' 'dumstr2); ref2,dumstr2); utstr+' 'dumstr2); utstr+' 'dumstr2);</pre>	<pre>begin utstr+' +dumstr; dex]::3,dumstr); utstr+' +dumstr); utstr+' +dumstr); utstr+' +dumstr); utstr+' +dumstr2); putstr+' +dumstr2); res_,dumstr2); res_,dumstr2); rest-resd_temp(fcontro); res</pre>	<pre>begin utstr+' 'dumstr; idex];7:3,dumstr); utstr+' 'dumstr); utstr+' 'dumstr); utstr+' 'dumstr); vutstr+' 'dumstr2); nvert_volt(setpt[index]); for loop:=1 to average_readin volt:=0.0; nv:=0.0; for loop:=1 to average_readin volt:=0.0; for loop:=1 to average_readin volt:=volt+read_current(ime volt:=volt+read_volt(vmeter.devi inds]:4:1,dumstr2); findex]:4:1,dumstr2);</pre>	<pre>outstr+' 'dumstr; ndex]:7:3,dumstr); outstr+' 'dumstr); outstr+' 'dumstr); outstr+' 'dumstr); ndex]:7:4,dumstr); outstr+' 'dumstr); notesontesonternessionternessionternession iode voltage conversionternessionternession iode voltage conversionternessionternession iode voltage conversionternessionternession voltactr+' 'dumstr2); voltactr+' 'dumstr2); tem:=temtred_temp(Tcontr</pre>	<pre>outstr+ ' +dumstr; begin outstr+' ' +dumstr; tem:=0.0; outstr+' ' +dumstr; setpnt:=0.0; outstr+' ' +dumstr; setpnt:=0.0; outstr+' ' +dumstr; ny:=0.0; outstr+' ' +dumstr; ny:=0.0; y:6:2,dumstr2); for loop:=1 to average_read; begin tem:=tem+read_temp(Tcontre (contre</pre>	<pre>outer:' ' +dumstr; outer:' ' +dumstr; ndex]:7:3,dumstr; rem:=0.0; ndex]:7:4,dumstr; outer:' ' +dumstr; nde voltage conversion************************************</pre>
<pre>utatr+ '+dumatr; dex]:7:3,dumatr; dex]:7:3,dumatr; dex]:7:4,dumatr; utatr+' +dumatr; ode voltage conversion************************************</pre>	<pre>istr:' '+dumstr; becirt' '+dumstr; istr:' '+dumstr; betint:=0.0; istr:' '+dumstr; betint:=0.0; istr:=0.0; betint:=0.0; volt:=0.0; volt:=0.0; inver_Volt(eetpt[index]); istr:=0.0; inver_Volt(eetpt[index]); istr:=0.0; inver_Volt(eetpt[index]); volt(eetpt[index]); istr:=volt(eetpt[index]); volt(eetpt[index]); istr:*' '+dumstr2); index]:41,dumstr2); index]:41,dumstr2); power[point]:=read_gain(Tcon gain[point]:=read_gain(Tcon</pre>	<pre>utstr+ '+dumstr; dex]:7:3,dumstr); tetm:=0.0; utstr+ '+dumstr); utstr+ '+dumstr); utstr+ '+dumstr); utstr+ '+dumstr); ode voltage conversion************************************</pre>	<pre>utstr+' '+dumstr; dex]:7:3,dumstr); utstr+' +dumstr; utstr+' +dumstr; dex]?f.4,dumstr); utstr+' +dumstr; dev lige conversion************************************</pre>	<pre>outstr+ ' +dumstr; ndex]:7:3,dumstr); outstr+ ' +dumstr); outstr+' +dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr); ndex]:7:4,dumstr); noutstr+' +dumstr); y:6:2,dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); nutstr+' +dumstr2); nutstr+' +dumstr2); nutstr+' +dumstr2); power[point]:=read_pain[point]:=read_pain[Tcont power[power[po</pre>	<pre>outstr+' 'dumstr; outstr+' 'dumstr; ndex]:7:4,dumstr; outstr+' 'dumstr); ndex]:7:4,dumstr; outstr+' 'dumstr); outstr+' 'dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); y:6:2,dumstr2); outstr+' 'dumstr2; rem:=temread_volt(Vmete outstr+' 'dumstr2); (index]:3:0,dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2); outstr+' 'dumstr2);</pre>	<pre>outstr+' + dumstr; dex]:7:3,dumstr; ndex]:7:3,dumstr; outstr+' + dumstr; outstr+' + dumstr; outstr+' + dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:4,dumstr; ndex]:4,dumstr; ndex]:4,dumstr; ndex]:4,dumstr2; nutstr+' + dumstr2; nutstr+' + dumstr2;</pre>	<pre>outstr+ '+dumstr; dex]:7:3,dumstr; ndex]:7:4,dumstr; outstr+' '+dumstr; outstr+' '+dumstr; ndex]:7:4,dumstr); outstr+' '+dumstr); for loop:=1 to average_readi noter_Uol(eetpt[index]); y:6:2,dumstr2); outstr+' '+dumstr2); outstr+' '+dumstr2); outstr+' '+dumstr2); power[point]:=read heater point]; power[point]:=read heater point];</pre>	<pre>begin utstr+' +dumstr; dex]:1:3,dumstr); utstr+' +dumstr; utstr+' +dumstr; utstr+' +dumstr; utstr+' +dumstr; outstr+' +dumstr2); putstr+' +dumstr2); nv:=0.0; rem:rem+read_remp(Tcontrol curr:-curriread_current(Ime volt:-volt+read_nv(nVmeter.dev) end; [lop]</pre>	<pre>begin utstr+' +dumstr; idex]:?:3,dumstr; idex]:?:4,dumstr; utstr+' +dumstr; utstr+' +dumstr; idex]:?:0,dumstr; idex]:?:0,dumstr2; inv:=0.0; ivelt==0.0; volt:=0.0; volt:=0.0; ivelt==0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=read.eur;(fontro: volt:=volt+read_volt(Vmeter.dev: (index]:4:1,dumstr2); end;(loop)</pre>	<pre>begin utstr+' 'dumstr; idex]:7:3,dumstr); utstr+' 'dumstr); utstr+' 'dumstr); utstr+' 'dumstr); utstr+' 'dumstr); putstr+' 'dumstr2); nvert_voltage conversion************************************</pre>	<pre>outstr+ ' +dumstr; begin ndex]:7:3,dumstr); tem:=0.0; outstr+' +dumstr) outstr+' +dumstr); volt=0.0; ndex]:7:4,dumstr); volt=0.0; idot voltage conversion************************************</pre>	<pre>outstr+ ' +dumstr; begin ndex]:7:3,dumstr; tem:=0.0; outstr+' ' +dumstr; setpnt:=0.0; outstr+' ' +dumstr; setprt:=0.0; outstr+' ' +dumstr; ny:=0.0; outstr+' ' +dumstr; ny:=0.0; y:6:2,dumstr2); for loop:=1 to average_read; y:6:2,dumstr2); tem:=tem+read_temp(Tcontre (contre)</pre>	<pre>outstr+' + dumstr; ndex]:7:3,dumstr; setpnt:=0.0; outstr+' + dumstr; outstr+' + dumstr; setpnt:=0.0; outstr+' + dumstr; outstr+' + dumstr; setref(index]); setpn=1 to average_readi potstr+' + dumstr2); begin outstr+' + dumstr2;</pre>
<pre>utetr+' + dumatr; dex]:7:3,dumatr; dex]:7:3,dumatr; dex]:7:4,dumatr; utetr+' + dumatr; utetr+' + dumatr; ods voltage conversion************************************</pre>	<pre>Latr+' '+dumstr; bex]:7:3,dumstr); tatr+' '+dumstr; tatr+' '+dumstr; bev]:7:4,dumstr; bev]:7:4,dumstr; bev]:7:4,dumstr; tatr=0.0; tatr=0.0; volt=0.0; nver_Volt(eetpt[index]); volt(eetpt[index]); tem:read_temp(Tcontr tem:read_temp(Tcontr volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmet volt:=volt+read_volt(Vmetr.de end; [lop] tater+' '+dumstr2); power[point]:=read_gain(Tcon gain[point]:=read_gain(Tcon</pre>	<pre>utstr+ '+dumstr; dex]:7:3,dumstr); utstr+ '+dumstr; utstr+ '+dumstr; utstr+ '+dumstr; utstr+ '+dumstr; ode voltage conversion************************************</pre>	<pre>utetr+' '+dumetr; dex]:7:3,dumetr); utetr+' '+dumetr; dex]:7:4,dumetr; dex]:7:4,dumetr; dex]:7:4,dumetr; dev1:7:4,dumetr); utetr+' '+dumetr2); nvert Volt(setpt[index]); nvert Volt(setpt[index]); nuetr+' '+dumetr2); nuetr+' '+dumetr2); nu</pre>	<pre>outstr+ '+dumstr; index]:7:3,dumstr); outstr+ '+dumstr; outstr+ '+dumstr; index]:7:4,dumstr); outstr+ '+dumstr; index]:7:4,dumstr); index]:7:4,dumstr); index]:7:4,dumstr); index]:7:4,dumstr); index]:7:4,dumstr); index]:7:4,dumstr); index]:7:4,dumstr2); intervention interve</pre>	<pre>outstr+' + dumstr; ndex]:7:3,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' + dumstr; ndex]:7.4,dumstr; ndex]:7.4,dumstr; ndex]:7.4,dumstr; ndex]:7.4,dumstr; iode voltage conversion************************************</pre>	<pre>outstr+' + dumstr; dex]:7:3,dumstr; ndex]:7:4,dumstr; outstr+' * dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr2); vi6:2,dumstr2); nutstr+' * dumstr2); nutstr+' * dumstr2);</pre>	<pre>outatr+ '+dumatr; dex]:7:3,dumatr; ndex]:7:4,dumatr; outatr+' '+dumatr; outatr+' '+dumatr; ndex]:7:4,dumatr); outatr+' '+dumatr2); for loop:=1 to average_readi noise:_dumatr2); y:6:2,dumatr2); outatr+' '+dumatr2); outatr+' '+dumatr2);</pre>	<pre>begin utstr+' +dumstr; idex]:7:3,dumstr; idex]:7:4,dumstr; utstr+' +dumstr; utstr+' +dumstr; utstr+' +dumstr; inv:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; iverLyolt(setpt[index]); iverLyolt(setpt[index]); iverLyolt(setpt[index]); ivert_dumstr2; ivertiantstrates ivertiantstrates volt:=volt+read_volt(wetes volt:=volt+read_volt(wetes volt:=volt+read_volt(wetes index1:3); index1:3);</pre>	<pre>begin outstr+' +dumstr; dex]:7:3,dumstr); butstr+' *dumstr; outstr+' *dumstr; ndex]:7:4,dumstr; curr=0.0; volt=0.0; volt=0.0; volt=0.0; volt=etpt[index]); pret_volt=etpt[index]); pret_volt=etpt[index]); pret_volt=etpt[index]); pret_volt=etpt[index]; outstr+' *dumstr2); outstr+' *dumstr2); end; [loop]</pre>	<pre>utstr+' '+dumstr; dex]:7:3,dumstr); utstr+' '+dumstr; utstr+' '+dumstr; utstr+' '+dumstr; dex]:7:4,dumstr); utstr+' '+dumstr; oode voltage conversion************************************</pre>	<pre>outstr+ ' +dumstr; .ndex]:7:3,dumstr); outstr+' ' +dumstr; .ndex]:7:4,dumstr; .ndex]:7:4,dumstr); .ndexl:? ' +dumstr); .ndexl:? *dumstr); .ndexl:? *dumstr); .ndexl:? *dumstr); .outstr+' ' +dumstr2); .tem:read_temp[forh]; .tem:read_temp[forh];</pre>	<pre>outetr+' ' +dumstr; ndex]:7:3,dumstr; outstr+' ' +dumstr; outstr+' ' +dumstr; outstr+' ' +dumstr; idex 0ltage conversion************************************</pre>	<pre>outstr+' + dumstr; ndex]:7:3,dumstr; outstr+' ' + dumstr; outstr+' ' + dumstr; ndex]:7:4,dumstr; outstr+' + dumstr; outstr+' + dumstr; outstr+' + dumstr; iet v (tumstr2); outstr+' + dumstr2); outstr+' + dumstr2); outstr+' + dumstr2);</pre>
<pre>begin utetr+' +dumetr; dex]:7:3,dumetr); tem:=0.0; dex]:7:4,dumetr); utetr+' +dumetr); utetr+' +dumetr); ods volte=0.0; volt:=0.0; nvert_Volt(setpt[index]); nvert_Volt(s</pre>	<pre>begin tear:+' '+dumstr; bex]:7:3,dumstr; tear:=0.0; bes1:7:3,dumstr; bes1:7:4,dumstr; bes1:7:4,dumstr; bes1:7:4,dumstr; bes1:7:4,dumstr; corr==0.0; bes1:7:4,dumstr; corr==0.0; nv:=0.0; for loop:=1 to average_read for loop</pre>	<pre>utetr+ '+dumetr; dex]:7:3,dumetr); utetr+ '+dumetr; utetr+ '+dumetr); utetr+ '+dumetr; dex]:7.4,dumetr); utetr+ '+dumetr; nvert_Volt(setpt[index]); ifor loop:=1 to average_readi. for loop:=1 to average_readi. fo</pre>	<pre>utetr+' '+dumatr: dex]:7:3,dumatr); utetr+' '+dumatr); utetr+' '+dumatr); utetr+' '+dumatr); utetr+' '+dumatr); utetr+' '+dumatr2); nvert Volt (setpt[index]); invert Volt (setpt[index]); utetr+' '+dumatr2); utetr+' '+dumatr2);</pre>	<pre>outstr+' + dumstr; ndex]:7:3,dumstr; ndex]:7:4,dumstr; outstr+' ' + dumstr; outstr+' ' + dumstr; ndex]:7:4,dumstr; ndex[:7:4,dumstr]; outstr+' ' + dumstr; nv:=0.0; nv:=0</pre>	<pre>outer+' ' +dumatr; outer+' ' +dumatr; ndex]:7:3,dumatr; ndex]:7:4,dumatr; outer+' ' +dumatr; outer+' ' +dumatr; outer+' ' +dumatr; ione voltage converteron ione voltage converteron ione voltage converteron ione voltage converteron ione voltage converteron ione voltage is 0.0; volt:=volt=read_volt(Vmete outer+' ' +dumatr2; index]:4:1,dumatr2; outer+' ' +dumatr2; index]:4:1,dumatr2; index]:4:1,dumatr2; index]:4:1,dumatr2; index]:4:1,dumatr2; index]:4:1,dumatr2; index]:4:1,dumatr2; index]:4:1,dumatr2; index]:4:1,dumatr2;</pre>	<pre>outstr+' +dumstr; dex]:7:3,dumstr; ndex]:7:4,dumstr; outstr+' 'dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; inv=0.0; volt==0.0; volt==0.0; volt==0.0; volt==0.0; volt==0.0; volt==0.0; volt==0.0; volt==0.0; volt==0.0; volt==volt+read_temp[font:=read_heate_pc] outstr+' +dumstr2; index]:4:1,dumstr2; volt==volter_volter_pc] volt==volter_vo</pre>	<pre>outstr+' '+dumstr; dex]:7:3,dumstr; ndex]:7:4,dumstr; outstr+' +dumstr; outstr+' +dumstr; ndex]:7:4,dumstr); outstr+' +dumstr); y dod voltage conversion************************************</pre>	<pre>untarr+' +dumstr; idex]:7:3,dumstr; idex]:7:3,dumstr; untstr+' *dumstr; untstr+' *dumstr; untstr+' *dumstr; reit+0.0; nv:=0.</pre>	<pre>putstr+' +dumstr; dex]:7:3,dumstr; dex]:7:4,dumstr; uutstr+' *dumstr; detal:7:4,dumstr; detal:7:4,dumstr; detal:7:4,dumstr; retr=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=dumstr2; volt:=volt+read_uv(nVmeter.dev. outstr+' *dumstr2; uutstr+' *dumstr2; volt:=volt+read_uv(nVmeter.dev. end; [loop]</pre>	<pre>utatr' 'dumstr; dex]:7:3,dumstr; dex]:7:4,dumstr; utstr+ 'dumstr; utstr+ 'dumstr; dev]:0:0; utstr+ 'dumstr; oode voltage conversion************************************</pre>	<pre>outstr+' ' + dumstr; ndex]:7:3, dumstr); outstr+' ' + dumstr; outstr+' ' + dumstr; ndex]:7:4, dumstr); outstr+' ' + dumstr; notes voltage convertion************************************</pre>	<pre>outstr+' '+dumstr; ndex]:7:3,dumstr; setpnt:=0.0; outstr+' '+dumstr; outstr+' '+dumstr; ndex]:7:4,dumstr; outstr+' '+dumstr; outstr+' '+dumstr; y:6:2,dumstr2); w:6:2,dumstr2); tem:=tem+read_temp(Tcontre (Tcontre</pre>	<pre>outstr+' ' + dumstr; ndex]:7:3, dumstr; cen:=0.0; outstr+' ' + dumstr; ndex]:7:4, dumstr; outstr+' + dumstr; outstr+' + dumstr; outstr+' + dumstr; nvert_Volt(setp[index]); fect [index]); outstr+' + dumstr2); cottst.' + dumstr2); cottst.' + dumstr2);</pre>
<pre>begin utstr+' + dumstr; dex]:7:3,dumstr); utstr+' + dumstr; utstr+' + dumstr; utstr+' + dumstr); utstr+' + dumstr2); utstr+' + dumstr2); nv=0.0; for loop:=1 to average_res for lo</pre>	<pre>tarr: ' +dumstr; begin tem:=0.0; besl:7:3,dumstr; besl:7:3,dumstr; besl:7:4,dumstr; besl:7:4,dumstr; besl:7:4,dumstr; besl:7:4,dumstr; besl:7:4,dumstr; corr=0.0; begin test:+' +dumstr2); tem:=tem+read_temp(fcontr volt:=volt+read_volt(Nmeter de power[point]:=read_gain(fcon index]:4:1,dumstr2); power[point]:=read_gain(fcon gain[point]:=read_gain(fcon</pre>	<pre>utetr+' + dumetr; dex]:7:3,dumetr); utetr+' + dumetr; dex]:7:4,dumetr); utetr+' + dumetr); dex]:7:4,dumetr); dex]:7:4,dumetr); utetr+' + dumetr2); utetr+' + dumetr2);</pre>	<pre>uttr:+ ' +dumstr; dex]:7:3,dumstr; utstr+' +dumstr; utstr+' +dumstr; dex]:7:4,dumstr; dex]:7:4,dumstr; utstr+' +dumstr; wolt(=0.0; nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nver_Volt(eetpt[index]); nuetr+' +dumstr2); nuetr+' +dumstr2); nuetr+' +dumstr2); nuetr+' +dumstr2); nuetr+' +dumstr2); nuetr+' +dumstr2);</pre>	<pre>outstr+' + dumstr; ndex]:7:3, dumstr; ndex]:7:4, dumstr; outstr+' + dumstr; ndex]:7:4, dumstr; ndex]:7:4, dumstr; noutstr+' + dumstr; noutstr+' + dumstr2); y:6:2, dumstr2); poutstr+' + dumstr2); r[index]:4:1, dumstr2); power[point]:=read_pain[foont]:=read_pain[foont]; nutstr+' + dumstr2); power[point]:=read_pain[foont]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; power[point]:=read_pain[foont]; pain[point]:=read_pain[foont]; pain[point]:=read_pain[foont]; pain[point]; pain[p</pre>	<pre>outstr:' ' dumstr; dex]:7:3,dumstr; ndex]:7:4,dumstr; outstr:' ' dumstr; outstr:' ' dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; outstr:' ' dumstr2) sourt Volt (setpt[index]); y:6:2,dumstr2); outstr:' ' dumstr2); outstr:' ' dumstr2);</pre>	<pre>outstr+' + dumstr; dex]:7:3, dumstr; ndex]:7:4, dumstr; outstr+' + dumstr; ndex]:7:4, dumstr; ndex]:7.4, dumstr); outstr+' + dumstr); yode voltage conversion************************************</pre>	<pre>outstr+' '-dumstr; dex]:7:3,dumstr; dex]:7:4,dumstr; outstr+' '-dumstr; outstr+' '-dumstr; ddex]:7:4,dumstr; iode voltage conversion************************************</pre>	<pre>ultativ' 'dumatr; dex]:7:3,dumatr); dex]:7:4,dumatr); ultativ' 'dumatr); ultativ' 'dumatr); ultativ' 'dumatr2); ultativ' 'dumatr2);</pre>	<pre>untatry' + dumatr; dex]:7:3,dumatr; dex]:7:3,dumatr; uutatr' '+dumatr; uutatr' '+dumatr; uutatr' '+dumatr; uutatr' '+dumatr; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; nv:=0.0; for loop:=1 to average_readi volt:=0.0; nv:=0.0; volt:=0.0; volt:=reurtread_temp(Tcontro tem:=temtread_temp(Tcontro tem:=temtread_temp(Tcontro tem:=temtread_temp(Tcontro tem:=temtread_temp(Tcontro tem:=temtread_out(Immetro volt:=rvolt+read_uV(nVmeter.dev. findex]:4:1,dumatr2); findex]:4:1,dumatr2);</pre>	<pre>utatr+' 'dumatr; dex]:7:3,dumatr; dex]:7:4,dumatr; utatr+' 'dumatr; utatr+' 'dumatr; utatr+' 'dumatr; putatr+' 'dumatr; pot loge voltage conversion************************************</pre>	<pre>outstr+' + dumatr; ndex]:7:3,dumatr; ndex]:7:3,dumatr; outstr+' + dumatr; outstr+' + dumatr; notext: 7:4, dumatr; notext: 7:4, dumatr; notextr+' + dumatr; notest voit (setpt[index]); y:5:2,dumatr2); outstr+' + dumatr2; tem:=tem+read_temp(fcontr</pre>	<pre>desl:r:3.dumstr; ndex]:7:3.dumstr; ndex]:7:4.dumstr; outstr+' 'dumstr; ndex]:7:4.dumstr; ndex]:7.4.dumstr; ndex]:7.4.dumstr; outstr+' 'dumstr; volt=0.0</pre>	<pre>outstr+' ' dumstr; begin dex]:7:3,dumstr; dex]:7:4,dumstr; outstr+' ' dumstr; ndex]:7:4,dumstr; ndex]:7:4,dumstr; outstr+' ' dumstr; potetr' (dumstr2); potetr' (dumstr2); begin outstr+' ' dumstr2);</pre>
<pre>iter:' 'dumerr; tem:=0.0; tem:=0.0; uter:' 'dumerr; dex]:7:3,dumerr; uter:' 'dumerr; uter:' 'dumerr); uter:' 'dumerr2); uter:' 'dumerr2);</pre>	<pre>raise or dumatr; iter:' ' dumatr; iter:' ' dumatr; iter:' ' dumatr; dex]:7:3, dumatr; dex]:7:4, dumatr; dex]:7:4, dumatr; dex]:7:4, dumatr; dex]:7:4, dumatr; dex]:7:4, dumatr; dex]:7:4, dumatr; dex]:7:4, dumatr; rem: rem: rem: rem: rem: rem: rem: rem:</pre>	<pre>action of an action and a set of a</pre>	<pre>ration of dumetr) deal 77:3, dumetr) tem:=0.0; tem:</pre>	<pre>cast1::y,dumstr; outdatr* '+dumstr; outdatr* '+dumstr; outdatr* '+dumstr; outdatr* '+dumstr; outdatr* '+dumstr; ndex]:?:s,dumstr; ndex]:?:s,dumstr; outdatr* '+dumstr; poutdatr* '+dumstr2); r[index]:s1:,dumstr2); r[index]:s1:,dumstr2); power[point]:=read_pain[Fornt]; power[point]:=read_pain[Fornt]; power[point]:=read_pain[Fornt]; power[point]:=read_pain[Fornt];</pre>	<pre>cutat::' 'dumat:); outat:' 'dumat: dex]:?.dumat: outat:+' 'dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]:?.dumat: ndex]</pre>	<pre>cutstry ' dumatry outstry' ' dumatry dex]:7:3,dumatry outstry' ' dumatry ndex]:7:4,dumatry ndex]:7:4,dumatry ndex]:7:4,dumatry outstry' ' dumatry y:6:2,dumatr2); y:6:2,dumatr2); outstry' ' dumatr2); outstry' ' dumatr2);</pre>	<pre>cutstr' ' +dumstr; begin dex]:?;4,dumstr; outstr+' +dumstr; butstr' +dumstr; ndex]:?;4,dumstr; ndex]:?;4,dumstr; cutr:=0.0; volt=0.0; nv:=0.0; volt=0.0; volt=0.0; volt=0.0; volt=0.0; volt=0.0; volt=0.0; volt=0.0; volt=1:3:0,dumstr2); volt=1:4:1,dumstr2); outstr+' +dumstr2); power[point]:=read heater poi outstr+' +dumstr2);</pre>	<pre>sedic:::</pre>	<pre>autstr' 'dumstr; begin dex1:7:3,dumstr; bethatr: dex1:7:4,dumstr; butstr' 'dumstr; butstr' 'dumstr; begin putstr' 'dumstr2); prect_volt(setpt[index]); prect_volt(setpt[i</pre>	<pre>sex1:0:0, dumatr; uutstr+' + dumatr; dex]:7:3,dumatr; uutstr+' + dumatr; uutstr+' + dumatr; uutstr+' + dumatr; ode voltage conversion************************************</pre>	<pre>cast1::y, dumatr; outdtr+' ' dumatr; ndex1:7:3, dumatr; outstr+' ' dumatr; ndex1:7:4, dumatr; outstr+' ' dumatr; retr=0.0; ide voltage conversion************************************</pre>	<pre>cation.cumatr; cutatr:'.to.umatr; outatr:'.to.umatr; dex]:?:4,dumatr; outatr:'.to.umatr; outatr:'.toumatr; idex]:?:4,dumatr; ndex]:?:4,dumatr; cutr:=0.0; volt:=0.0; volt:=0.0; for loop:=1 to average_read pegin tem:=temtread_temp(Tcontr</pre>	<pre>conter://</pre>
<pre>state::</pre>	<pre>xs1:st.c.dumstr;; itatr+''+dumstr; itatr+''+dumstr; itatr+''+dumstr; bex]:7:3,dumstr; bex]:7:4,dumstr; bex]:7:4,dumstr; bex]:7:4,dumstr; ber]:7:4,dumstr; ber]:7:4,dumstr; ber]:7:4,dumstr; ber]:7:4,dumstr2); curr=curr+read_remp(Tcontr volt==0.0; for loop:=1 to average_read for loop:=1 to average_read for</pre>	<pre>extracts; cunners; cunners; cunners; cunners; cunners; cunners; cunners; cunters; cunten</pre>	<pre>exists the interact is a set of the inter</pre>	<pre>cdex(:s), dumatr); outatr+' ' +dumatr); begin ddex[:7:4, dumatr); outatr+' ' +dumatr); outatr+' +dumatr); outatr+' +dumatr); idde voltage conversion************************************</pre>	<pre>deajist oumstr); begin dealist: ddex]:?:4,dumstr); bettr:=0.0; outstr+' +dumstr); ddex]:?:4,dumstr; ndex]:?:4,dumstr; ndex]:?:4,dumstr; ndex]:?:4,dumstr; soutstr+' +dumstr2; iode voltage conversion************************************</pre>	<pre>curst::</pre>	<pre>currant::: currant:: currant::</pre>	<pre>sexj:s:s.dumstr; begin dexj:7:3,dumstr; dexj:7:3,dumstr; detart' 'dumstr; detart' 'dumstr; nutstr' 'dumstr; devoltage conversion nutstr' 'dumstr2); sutstr' 'dumstr2) nutstr' 'dumstr2); sutstr' 'dums</pre>	<pre>setist:::</pre>	<pre>exj:::</pre>	<pre>dex(1:1), dumatr); outatr+' ' dumatr; dex[:7:3,dumatr); outatr+' ' dumatr; ndex[:7:4,dumatr); outatr+' ' dumatr; ndex[:7:4,dumatr); outatr+' ' dumatr; voutatr+' ' dumatr2); voutatr+' ' dumatr2); voutatr+' ' dumatr2); tem:=tem+read_temp(tontr ' tem:=tem+read_temp(tontr ' tem:=tem+read_temp(tontr');</pre>	<pre>dex1:st.ordumetr; outstr+' '+dumetr; hdex1:7:s.fumetr; outstr+' '+dumetr; outstr+' '+dumetr; ndex1:7:4,dumetr; outstr+' '+dumetr; idde voltage conversion************************************</pre>	<pre>dexls::</pre>
<pre>ax1:8:5,dumatr); dex1:7:3,dumatr); utstr+' '+dumatr; begin utstr+' '+dumatr; dex1:7:4,dumatr; begin utstr+' '+dumatr; corr=0.0; volt=0.0; volt=0.0; volt=0.0; volt=0.0; for loop: begin utstr+' '+dumatr2; utstr+' '+dumatr2; utstr+' '+dumatr2; utstr+' '+dumatr2; utstr+' '+dumatr2; volt=0.0; volt=volt+read_volt(Vme volt:=volt+read_volt(Vme volt=volt+volt+read_volt(Vme volt=volt(Vme volt=volt+volt(Vme volt=volt+volt(Vme volt(Vme volt=volt(Vme volt(Vme volt=volt(Vme volt(Vme volt(V</pre>	<pre>xx1 s1:6, dumstr); itstr+' + dumstr; begin itstr+' + dumstr; bex] i7:4, dumstr); bex] i7:4, dumstr; bex] i7:4, dumstr; bex] i7:4, dumstr; bes[i i i i i i i i i i i i i i i i i i i</pre>	<pre>ex1:8:5,dumatr); utatr: ' +dumatr); begin dex1:7:3,dumatr); utatr: ' +dumatr); utatr: ' +dumatr); dex1:7:4,dumatr); utatr: ' +dumatr); utatr: ' +dumatr2); utatr: ' +dumatr2);</pre>	<pre>ex(:8:6, dumatr); utatr+ ' + dumatr); utatr+ ' + dumatr); tem:=0.0; utatr+ ' + dumatr); tem:=0.0; dex]:7:4, dumatr); tem:=0.0; dety1:7:4, dumatr); utatr+ ' + dumatr2; volt:=0.0; nvert_volt(eetpt[index]); volt:=0.0; for loop:=1 to average_readi nvert_volt=eetpt[index]); volt:=0.0; for loop:=1 to average_readi for loop:=1 to average_readi nvert_volt=eetpt[index]); tem:=temtread_readi volt:=volt+read_volt(Vmete nvert+' + dumatr2); utatr+' + dumatr2); utatr+' + dumatr2); utatr+' + dumatr2); utatr+' + dumatr2); utatr+' + dumatr2); utatr+' + dumatr2); index]:4:1, dumatr2); utatr+' + dumatr2); index]:4:1, dumatr2); index]:4:1, dumatr2);</pre>	<pre>dex]8:6,dumetr); outatr+' 'dumetr); dex]7:3,dumetr); mdex]7:3,dumetr); mdex]7:4,dumetr); outatr+' 'dumetr); mdex]7:4,dumetr); mdex]7:4,dumetr); outetr+' 'dumetr2); outetr+' 'dumetr2); outetr+' 'dumetr2); mdex]5:1,dumetr2); mdex]5:1,dumetr2); findex]5:1,dumetr2); gain[point]5=read_gain[Tcont gain[point]5=read_gain[Tcont</pre>	<pre>dex1(8:6,dumetr); outatr+' 'dumetr); dex1(7:3,dumetr); mdex1(7:3,dumetr); ndex1(7:3,dumetr); uutatr+' 'dumetr); ndex1(7:4,dumetr); outatr+' 'dumetr2); uutatr+' 'dumetr2); outatr+' 'dumetr2); (index1(3:0,dumetr2); (index1(3:0,dumetr2); outatr+' 'dumetr2); (index1(3:1,dumetr2); (ind</pre>	<pre>dex1st.(dumetr); outstr+' + dumetr); begin 0: uutstr+' + dumetr); dex1:7:3,dumetr); uutstr+' + dumetr); ndex1:7:4,dumetr); uutstr+' + dumetr2; uutstr+' + dumetr2; y:6:2,dumetr2); uutetr+' + dumetr2); uutetr+' + dumetr2); uutetr+' + dumetr2); uutetr+' + dumetr2); uutstr+' + dumetr2); uutstr+' + dumetr2); uutstr+' + dumetr2);</pre>	<pre>dex1:8:6,dumatr); outstr+' +dumatr); hdex1:7:4 dumatr); begin ndex1:7:4 dumatr); betn:=0.0; ndex1:7:4 dumatr); ndex1:7:4 dumatr); ndex1:7:4 dumatr2); outstr+' +dumatr2); nv:=0.0;</pre>	<pre>dex]:85.dumatr); utatr+' 'dumatr); begin utatr+' 'dumatr); utatr+' 'dumatr); utatr+' 'dumatr); utatr+' 'dumatr; butatr+' 'dumatr); utatr+' 'dumatr2); utatr+' 'dumatr2);</pre>	<pre>dex]:8:6,dumatr); begin outstr*' 'dumatr); mdex]:7:4,dumatr); terpn:=0.0; betpn:=0.0; utstr*' 'dumatr); butstr*' 'dumatr2); putstr*' 'dumatr2</pre>	<pre>dex1:6.dumstr); utstr+''+dumstr); begin utstr+''+dumstr); utstr+''+dumstr); utstr+''+dumstr); begin vutstr+''+dumstr); volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=0.0; volt:=volt+read_current(Ime volt:rvolt+read_volt(Vmeter.devi utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2); utstr+''+dumstr2);</pre>	<pre>deal:8:6,dumatr); outatr+' 'dumatr; begin ndex]:7:3,dumatr; ndex]:7:4,dumatr; ndex]:7:4,dumatr; outatr+' 'dumatr; outatr+' 'dumatr]; iode voltage conversion************************************</pre>	<pre>dex1:8:c,dumetr); outatr+' 'dumetr); begin begin begin dex1:7:3,dumetr); outatr+' 'dumetr); ndex1:7:4,dumetr); ndex1:7:4,dumetr); outatr+' 'dumetr); yode voltage conversion************************************</pre>	<pre>dex1sf5.dumatr); dex1sf5.dumatr); begin begin dex1:1:3.dumatr); dex1:7:4.dumatr); dex1:7:4.dumatr); ndex1:7:4.dumatr); outstr+' +dumatr); outstr+' +dumatr); nv:=0.0; outstr+' +dumatr2); for loop:=1 to average_readi for loop:=1 to average_readi begin outstr.' +dumatr2); begin</pre>

procedure take reading(plot data:boolean);	Y1:=round((Y-Ymin)*Ystep)+yborder_bottom;
Var	if colour then SetColor(White);
saveblock, dummy : integer;	<pre>ine (xborder_ight, Y1, xborder_left, Y1);</pre>
dumreal : real?	
	II COLOUT THEM SECTORITEIOW); Chemistry(sisteri(servis)]aff thistramets)dis 3) Yl dimetri.
reacting; 16 milor date than havin	V.=V+V aris stor: V.=V+V aris stor:
at Part, data times ogian at	
(*****)jode voltace Convergion************************************	if colour then SetColor (Cyan);
dunreal:-convert Volt(temp[point]);	OutTextXY (acentre,
if plot data then plot point (dumreal, Abs (kval[point]));	<pre>yborder_bottom-((yborder_bottom_thickness*2) div 3).'Temperature (K)');</pre>
{	SetTextStyle (DefaultFont, VertDir, 1);
end;	OutTextXY((xleft+xborder_left_thickness div 3),
<pre>saveblock:=(point mod no points_in_block);</pre>	ycentre, Thermal Conductivity W/m/K') ;
if (saveblock=0) then store_points(point);	if colour then SetColor(White);
reading taken: =true;	SetLineStyle(Solidin,0,NormWidth);
nVpower:=false;	SetTextStyle (DefaultFont, HorizDir, 1);
set_nV_status(nVpower, plot_data);	SetTextJustify [LeftText, BottomText);
point:=point+1;	X1:-xleft+(xborder_left div 4);
if (point>-max_points) then	Y1:#title_linel/
begin	Setviework (zief, ytop, zright, ybot tom, Clipon);
	IT COLOUT THEN SECOLOT (LIGHTGZAY); Nort Att of the
dummy's support in block; 	GIRDIAN CAXL(X1, X1, 1186 NAVA A CAKA AC
	X1:#x1aft+(xhordar]aft div di cuarya ac oor co x //
accepturestories.	Al - maistra Associat atv 41/ Y1 - mit1a - 14ma2 -
	displaytext(XI.YI.'Temp. K I(mA) V '+
	<pre>distant in the second in the second is a second if the second is the second is a second is the second is a se</pre>
and. (take reading)	if colour then SetColor(White);
	end; {draw_scales}
procedure draw_scales;	procedure graph_mode;
	ter Prostingt British : interes:
	A LADOLLAND I THREECT A LADOLLAND ALL AND A
XI, YI : Integer;	Interant (sreputiver, sraphmode, graph, jach);
begin Last and the market of the second	DITUTOR - ALABATIC - A
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	
X_AXIB Stephenton Stephenton Stephenton, Amax, Amar); Y ====================================	uegin RestoraCrtMode:
×	header:
lasio uveria.com terpiumas, main. Vartandia.() Mar V value Mar V valua.():	wateln/(Granhics Error: '.GranhErrorMsd(ErrorCode)); witteln/
setTextStvle(SmallFont, Horizdir,4);	writeln('Program aborted');
SetTextJustify(CenterText);	Halt(1);
X:=Xmin;	end;
while (X<=Xmax) do	colour:=(GetPaletteSize=16);
begin	graphset:=true;
X1:=round((X-Xmin)*Xstep)+xborder_left;	end: {graph_mode}
if colour then SetGolor(White);	
line(X1,yborder_top,X1,yborder_bottom);	
str(X:6:2,dumstr);	procedure initialise_graph(tlag:integer);
11 COLUCE Then SetColor(Tellow); Amruvusit. bb	Var loom : integer:
ULTERTIAL JOOGGET DOLLOW-LYDUGGET COLLYW HALANGOS VI JJYWWWYLYY Vervy faf afar	JUOP JINEYSLA Animmuv Praal:
ATTATATATATATATATATATATATATATATATATATA	vuunnege
CIC/ THILLE/	if NOT graphset then graph mode
rimuri while (Y<=Ymax) do	else SetGraphMode (GraphMode);

·

<pre>transmission of the state of the state</pre>	X value: MetKaxi V value:	var dumav.loop : intener:
<pre> the off carefulcation</pre>	restructions. The second sector of the sector of the second sector second second second second second second second second s	setpt, newsetpt : real;
<pre>status as to a status a status as to a status</pre>	else top extra lines:=1;	gain, rate, rreset : real;
<pre>creation or construction of construction</pre>		newgain, newrate, newrreaet : real;
<pre>state and state and s</pre>	er tiltsknedenes are calculated right edge minus left edge.	begin
<pre>stant stant s</pre>	COP GOGE MINUS DOCCOM GOGE. INIS WAY CAE LACT CHAT U.U.15 Laft doer antertar)	nadori nain-read nainfrontrol dauirat.
<pre>interior of interior of interior i</pre>	Tari doed you warrant	garni-resource frontrol.device); rate:=read rate (Tcontrol.device);
<pre>M (M (M (M (M (M (M (M (M (M (</pre>	_lines:=top_no_of_lines+top_extra_lines;	rreset :-read_reset (Tcontrol .device) ;
<pre></pre>	:=0; 	setpt:=read_setpoint(Tcontrol.device); / *******/cdmolt.cocomment(_cet************************************
<pre>start % start start % start start % start start % start %</pre>		<pre>interiode to coversion interior in</pre>
<pre>Construction::::::::::::::::::::::::::::::::::::</pre>	201 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101 - 101	ertri-lookert_voltsertri. srifeln('Set roint = ' setri.o'):
<pre>() df (httms::round(httm)</pre>	er top thickness:=-1*(pixel line height*total lines);	
<pre>(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(</pre>	<pre></pre>	write('Enter the new setpoint: 0 to 325K 7 ');
<pre>(Tetrinit the balance: results): (Tetrinit the balance: resul</pre>	erright thickness:=round((xright-xleft)/10);	newsetpt:=input_real(setpt,0,325,true);
<pre>clobic clobic clob</pre>	er_left.=xleft+xborder_left_thickness;	header;writeln;
<pre>creation thread in the interval of the transmission of transmission of the transm</pre>	sr_right:=xright-xborder_left_thickness;	writeln('Gain = ',gain:6:2);
<pre>rements in the interval i</pre>	sr_bottom_thickness:=round((ytop-ybottom)/10);	write('Enter the new gain from ', min_gain:5:2,' to ', max_gain:5:2,' ? ');
<pre>c.potter.Potter.Potter. Media:Transfile. T.Potter.Pot</pre>	er_top:=ytop-yborder_top_thickness;	pewgain:=input_real(gain,min_gain,max_gain,true);
<pre>(r=renord:</pre>	er_bottom:=ybottom+yborder_bottom_thickness;	beder; writein;
<pre>instruction: Take: above the second in the second interval in</pre>	ht:=yborder_top-yborder_bottom; 	writein("Kate = ',rate:5:2); urita/"Enter the new rite from ' min rate.5.2 ' to ' mer rete.5.2 ' 2 ').
<pre>image:retrients interformer.int; interiments its interiments interiments its interiments interiments its interiments its interiments its interiments interiments interiments interiments interiments interiments interime</pre>	■1010年1011(1110年~1111)) 1111日として日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレンドは、日本・ドレ	Miller ziter ite iter var ilter juler julijateriji. Davrstaterikovit tasi[riterenin riter jurijitateriji.
<pre>seriptical: factor (yourder_top: (iteral_iner)) intel: recond (yourder_top: (yourder_top: iteration: interest: factor: interest: factor: interest: factor: interest: interest: factor: interest: interest</pre>	turitandotent targut anosteriaturi ≠11entek / (Teatritan):	nersons - anten - see traction - sector and -
<pre>surfacture: fact in factor in f</pre>	s == who then the second se	writeln('Remet = '.rreset:6:2):
<pre>jint:::end(tyberder_fop_thickness*(text_liness); fiel::reund(tyberder_fop_thickness*(text_lines); fiel::reund(tyberder_fop_thickne</pre>	e:=zborder left+(xlength div 2);	write("Enter the new reset from ", min reset:5:2," to ", max reset:5:2," ? ");
<pre>ind ::round(yborder_top_thice.meetp(); ind ::round(yborder_top_thice.meetp();</pre>	linel:=round(yborder_top+(yborder_top_thickness*(total_lines-1))	newrreaet:"input_real(rreset,min_reset,max_reset,false);
<pre>int := round (pocker_top_thickness* (rcal_lines);</pre>	/total_lines);	if (setpt<>newsetpt) then write_setpoint(Tcontrol.device,newsetpt);
<pre>lime2:=round(yborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.tyborder_top.thermes': immail:=round(yborder_top.tyborder_top.thermes')/total_limes); immail:=round(yborder_top.tyborder_top.thermes')/total_limes); immail:=round(yborder_top.tyborder_top.thermes')/total_limes); immail:=round(yborder_top.tyborder_top.thermes')/total_limes); immail:=round(yborder_top.thermes')/total_limes); immail:=round(yborder_top.total_limes)//total_limes); immail:=round(yborder_top.thermes')/total_limes); immail:=round(yborder_top.thermes')/total_limes); immail:=round(yborder_top.thermes')/total_limes); immail:=round(yborder_top.thermes')/total_limes); immail:=round(yborder_top.thermes')/total_limes); immail:=round(yborder_top.thermes')/total_limes); immail:=round(yborder_top.thermes')/total_limes); immail:=round(yborder_top.thermes'); immail:=round(y</pre>	linel:=round(yborder_top+(yborder_top_thickness*(total_lines-2)) /total_lines);	delay(100); if (rate<>newrate) then write rate(Tcontrol.device.newrate):
<pre>(recisi lines = 7 + 0° estra lines)/(cotal lines); (recisi lines = 7 + 0° estra lines)/(cotal lines); (recisi lines = + 0° estra lines)/(cotal lines); (recisi lines + rog estra lines)/(cotal lines); (recisi lines); (recisi lines); (recisi lines + rog es</pre>	line2:=round(vborder too+(vborder too thickness*	
<pre>ins2:=round(yborder_top_thicheaie ine3:=round(yborder_top_thicheaie</pre>		if (qain<>newgain) then write gain(Tcontrol.device,newgain);
<pre>Interiment in the interiment in the interiment int</pre>	ine2:=round(yborder top+(yborder top thickness*	delay(10);
<pre>Jimal:=round(yborder_top_thickneas* imal:=round(yborder_top_thickneas* icolar:</pre>	[total_lines-4-top_extra_lines)/total_lines);	<pre>if (rreset<>newrreset) then write reset(Tcontrol.device, rreset);</pre>
<pre>(cotal_lines'='top_atra_lines)/total_lines); inelines (cotal_lines'='top_atra_lines)/total_lines); clas: (cotal_lines'='top_atra_lines)/total_lines); clas: (cotal_lines'='top_atra_lines)/total_lines); procedure break_into_program; vari integer; begin</pre>	line3:=round(yborder_top+(yborder_top_thickness*	delay(10);
<pre>integer. op.trailing='-top_etra_linge)/(ctal_ling=)/(ctal_ling='-top_etra_ling=)/(ctal_ling='-top_etra_ling=)/(ctal_ling='-top_etra_ling=)/(ctal_ling='</pre>	(total_lines-5-top_ertra_lines))/total_lines);	end; [reading]
<pre>cclea; cclas; ag1) then bg1in for loop:=1 to point-1 do for loop:1 do for loop:1 loop: contact</pre>	ine3:=round (yborder top-tyborder top thickness	
<pre>agril then begin be</pre>	(colai_times')/(colai_times's'-top_erta_times)//(colai_times)/	procedure break into program:
<pre>begin for loop:=1 to point-1 do for loop:=1 to point-1 do begin begin umwy.r=convert_voit(temp[loop]); begin display_data(point-1); end; end; display_data(point-1); end; end; display_data(point-1); end; end; display_data(point-1); end; end; display_data(point-1); end; end; display_data(point-1); end; end; end; end; end; end; end; end</pre>	action: actions	
<pre>for loop:=1 to point-1 do</pre>		ans : integer;
<pre>"Diode voltage conversion************************************</pre>	for loop:=1 to point-1 do	look : charj
<pre>begin begin dumwy:=convert_Volt(temp[loop]); plot_point(dummy,Abs(kval[loop])); end:</pre>	*Diode voltage conversion************************************	maxoptions : integer;
<pre>dumy:=convert_Volt(temp[loop]); dumy:=convert_Volt(temp[loop]); end: end: end: display_data(point-1); display_data(point-1); end: initalise_graph) itialise_graph) itialise_graph) writeln('1 alter devices used (change keithley for fluke, etc)'); writeln('1 alter devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devi); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used (change keithley for fluke, etc)'); writeln('1 alter graphics devices used keithley</pre>	begin	begin
<pre>plot_point(dummy,Abs(kval[loop])); maxoptions:=9; ans:=9; look:=ReadRey; look:=ReadRey; display_data(point-1); display_data(point-1); display_data(point-1); look:='x'; header; writeln: look:='x'; header; writeln('1 alter devices used (change keithley for fluke, etc)'); writeln('1 alter setpoints'); writeln('1 alter graphice data'); </pre>	dummy:=convert_Volt(temp[loop]);	RestoreCRTMode;
<pre>end; end; display_data(point-1); end; end; end; initialise_graph) initialise_graph) writeln('1 alter devices used (change keithley for fluke, etc)'); writeln('1 alter veriable parameters (nV offsets etc)'); writeln('1 alter graphics data'); writeln('1 alter graphics data');</pre>	plot_point (dummy, Abs (kval[ioop]);	maxortions:=9;
<pre>display_data(point-1); discreased (look='i') or (look='i') or (ans<>maxoptions) do end; end; end; noti='x'; begin look:='x'; header; writeIn': nitialise_graph) writeIn': writeIn('1 alter variable parameters (nY offsets etc)'); writeIn('2 alter graphics data'); writeIn('1 alter graphics data');</pre>	end;	
<pre>display_data(point-1);</pre>		look:=ReadKey;
<pre>intralise_graph) nitialise_graph) writeln; writeln('1 alter devices used (change keithley for fluke, etc)'); writeln('2 alter variable parameters (nV offsets etc)'); writeln('3 alter setpoints'); writeln('4 alter graphics data');</pre>	display_data(point-1); and:	while ({look='i') or (look='i')} or (ans<>maxoptions) do begin
<pre>nitialise_graph) writeln; writeln('1 alter devices used (change keithley for fluke, etc)'); writeln('2 alter variable parameters (nV offsets etc)'); writeln('3 alter setpoints'); writeln('4 alter graphics data');</pre>		look:='x';
<pre>writeIn; writeIn('1 alter devices used (change keithley for fluke, etc)'); writeIn('2 alter variable parameters (nV offsets etc)'); writeIn('3 alter setpoints'); writeIn('4 alter graphics data');</pre>	nitiallse_graph)	header;
writeln('1 alter devices used (change keithley for fluke, etc)'); writeln('2 alter variable parameters (nV offsets etc)'); writeln('3 alter setpoints'); writeln('4 alter graphics data');		writeln;
witchn(i. PASTATATATATATATATATATATATATATATATATATAT		<pre>writeln('l alter devices used (change keithley for fluke, etc)'); </pre>
writeln('4 alter graphics data');		Wittein('2 aiter Vailadie parameters (nV oilsets etc)'); Writein('3 aiter setpoints');
		ritel'ar (stor grachic of ar

/ cases used the date date date ()	if auto gain then set gain etc(last setpoint);
writeln('6 take a data point');	get next reading time;
writein('7 alter the controller data');	get_setpoint_change_time;
writein(e end the program);	repeat
writein('9 go back to the graph and start collecting data again');writein;	system_clock(now_time);
write ('Choose the option you require: 1 to 9 ? ');	
ans:=input_integer(maxoptions,1,maxoptions,false);	
if (ans-1) then change_device_parameters;	(*************************************
if [ans-2] then change Variable_parameters(false);	
II (BUBWO) CIDEN DOGOTO	
atter Berpointes; ant astrointes;	II ((COMPARE TIME (NOW TIME, DEXT READING TIME)) and (NOT TIME flac)) that basin
ust_set_ctrotst_transg_trans,	and INCI paraginagin chen degin
if (anaed) than church resubics wariablas:	Lake realing (rue) :
it (more than the start starts.	Ger Jext Keasting Lines
is (allow)) tiggt arts (gots gots) is (starts) that starts (starts)	
TI UNIGATO LINEI LANGE FEGULIAINAGUI	
II (ADDM) / THEN ALLON CONTROLLET MANUALLY:	II ((compare_time(now_time,next_setpoint_time))
11 (mild-of tildi undin tritels/tritels/'Rad strates 2.''.	and (wor no work serpoint changes)) then change serpoint;
iteactisticalitatical programment ; ; if (see or sofficial) than beda	is formation that from the seas that would be a seasoff it.
1 (3 () () () () () () () () () () () () ()	190194591-00054 1913 //0011-0011-50110415011041
	if licomerate timeloom time of outimeli
halt:	and (NOT DYSUE)) the contract of the contract
end:	outra at the second sec
and: [whi]e]	and
entry lenses destrictions	
ALLOCATION(MARTIN)	is (Mangerand) than brack (at)
	LL (Neyrressed) then Dreak_Linto_program; dienlav time:
	until compare time(now time.end time):
{+++++++++++++++++++++++++++++++++++++	RestoreCRTMode;
(* main program *)	Nindow (1, 1, 80, 25) :
	ClrScri
	end.
Parten Lock (start cine);	
cominge devices parameters; sstathe devices	
clarge var.eure jacking clarge i	
gain valaalion gesaulta; 11445 min sein sein sein seinen se	
aler servicel	
restriction and the section of the s	
interalize graphics variables;	
Usatatione voitore fonderaionatatatatatatatatatatatatatatatatatatat	
write setooint (Tcontrol, device, duamy);	
last settooint = 1;	
{	

Ŧ	-	*	*****	

rograms			*******	
a for pi			******	
definition			*******	
- global	Delap		*******	
mrdglob	90 M.R.		******	
TINU	3/8/		****	

unit mrdglob;

interface

year : integer; month : integer; day : integer; hour : integer; minute : integer; mecond : integer; eeod; type time = record

namfil = string[12]; str2 = string[3]; str3 = string[4]; str5 = string[4]; str5 = string[6]; str14 = string[10]; str14 = string[12]; str14 = string[12]; str15 = string[16]; str20 = string[40]; str255 = string[40]; str255 = string[40]; str255 = string[40]; str255 = string[40]; str355 = string[40];

,

Var

GraphMode : integer;

	second: =second mod 60;
(* Unit MRDUTILS PAS *)	end;
<pre>" obtains date for thereal conductivity experiments "</pre>	lf (minute>39) then begin how
1. A.A. DELAÇ 1. NUVERNUCL 1993 1. A.A. MAY ARARARARARARARARARARARARARARARARARARA	nout: - nout : - nout v and of 50; minute mod 60;
•	end;
unit WRDUTILS;	if (hour>23) then begin
	day:=day+(hour div 24);
interface	hour:=hour mod 24;
the form of the second the	tanast harda
uses mrogloo, crc, dos, graph;	repeat begin of atrus:
<pre></pre>	flag:"false;
function zero(instring:strl2):atrl2/	if ((month=4) or (month=6) or (month=9) or (month=11)) then
runcium structure	bedin
procedure compress clock(intime:time:tar.var result:str20);	flac:=true:
function input strings.	if (day>30) then begin
procedure input time(dfault:time:var result:time:minus one:boolean);	Boath: Tourth: South: Sou
function input integer(default.min.maxintegersminus one:boolean);integer;	dav:=dav-30;
function input real(default, min, marreal, mine one: boolean) : real;	ok:=false;
function log(value:real):real;	end;
function power_of(a,b:real):real; {calculates a^b}	end;
procedure find_char(st:str255;ch:char;var pointer:integer);	if (month=2) then begin
procedure space;	flag:=true;
procedure cls;	if (day>28) then begin
procedure dashed_line;	month:=month+1,
function yes_or_no (default : boolean) : boolean;	day:=day-28;
<pre>procedure expand_clock(intime:str20;var result:time);</pre>	ok:=false
function compare(numl,num2:integer):integer;	end;
function compare_time(timel,time2:time): boolean;	end;
procedure system_clock(var result: time);	if (flag-false) then
procedure add_time(intime1,intime2:time;var result:time);	begin
function file_exists[filename,path:str255]:boolean;	if (day>31) then begin
function get_pathname(oldpath:str255):str255;	month:=month+1;
function file_version(name:str12;path:str255;stension:str4): str12;	day:=day-31;
function get_filename(oldfile:stri2;var path:str25;extension:str4;new_file:boolean)	ok:=false;
procedure houd (instring str235);	end;
procedure display directory(path:str233);	end; (lr) is (arrth:13) than taria
	II (MOULL/IZ) LUTEN JOGIN While month>12 do
ime)amentation	bedin
	year:=vear+1;
function zero(instring:stri2; stri2;	month:=month-12;
Var	end;
dummy : integer;	end;
begin	end; [repeat]
dummy:-Length (instring);	until ok;
II dumny-of then zerodu/; // dimensional according to the service of the servi	end;
AI GLUMMY=I. THEN ZETO:= U' THRELING; ff dimmuny] then servi=instring;	
at dummy/s trent retoring/	
	procedure compress_clock(intime:time;var result:str20);
	Var
procedure normalise_clock(var intime:time); var	outstr,dumstr : strzu; dummuv : intecer:
flac.ok : boolean;	beain · · · · · · · · · · · · · · · · · · ·
begin	<pre>{ normalise clock(intime);)</pre>
with intime do begin	with intime do begin
if (second-59) then bed in	result:='';

<pre>str(minute,dumstr); result:=result+zero(dumstr)+':';</pre>	accept:=false;
str(second,dumstr);	write (keystr);
str(day,dumstr);	euq
str (montr), duestr) /result:=result+zero (duestr) +'/';	IT accept then begin
str(year,dumstr);	QUEST: = QUEST: - Keyst:
dumy: -Length (dumstr);	
it dummay=1 then dumstri='0'+dumstr	WEILG (REVECT) ;
else dumstr:=copy(dumstr; (dumwy-1),2);	
result := result + dumstr ;	until (Regarment(13)) or (councer>=<3)); de (councer>=<5)
	<pre>is Counter=234) then dumstr:=dumstr+cnr(13); </pre>
	ant central and constants
procedure expand clock(intime:str20;var result:time);	
(input HH: NMH: SS-DD/NMH/YY)	
var	procedure input_time(default:time;var result:time;minus_one:boolean);
dumstr : str20;	
dummy,start,stop,len : integer;	[NB the value returned is the time only - the date returned is the
flag,ok : boolean;	same as that for default. Also, if -1 is true on entry then if
	- Is succeed then result. Winder 200 on sait
nied to superior and the superior of the super	
start:=1; find char(intime,':',stop);len:=stop=start;	loop, duumay, colon, len : integer;
dumstr:=copy(intime,start,len);	COMPACE CURRENT : STRIDG
	Jetter : sting(1)/
start:metopti, ting coart(intime, : 'stop)/ten:=stop-start;	
dumstr:=copy(intime,start,len);	
val (dumatr, minute, dummy);	
statt:sstory=1; lind_chart(intime,, stop);ien:=stop=statt; i==+	
dumstr =copy(intime, start, isn);	
Val (oursets/second ourset);	ALLAND THE
start:=stop+1; find_char(intime,'', stop);ien:=stop-start;	
dumstreeopy (intime start, ien);	dumerrs=copy(dumerrmer.tre);
$\nabla = \left\{ \begin{array}{ccc} 1 & 0 \\ 0 & 0 $	while (w) valad do begin Litelin'strate definite do 'h' dimentel'.
statt:scopting into contract interes / / scop)/iten:-scop-scart/	Cristing, 20 Marti indexe dari tradity
QUBBCK:=COPY(INCLER, BCAKC, IEU); 	
	TT (COMPACETCER(12)) CLART CELLICE
start:=stop+1; 	erse value:
ounsers = copy (incluse, start, /); still dates tast i dates (incluse);	
· / []	
	len:=lencth(dimmir):
	for loop:-1 to lend o begin
	accept:=false;
function input_string: str255;	letter:=copy(dumstr,loop,1);
Var	if ((letter='0') or (letter='1') or (letter='2') or (letter='3') or
counter, dummy, colon : integer;	<pre>(letter='4') or (letter='5') or (letter='6') or (letter='7') or</pre>
dumstr : str255;	(letter='8') or (letter='9')) then accept:=true;
keyatr : char;	if ((letter='-') and lastcolon) then accept:=true;
accept : boolean;	II accept then lastcolon:=failed; is intere
	IT (TRECERT :) SUG (NOT TERCERT)) LUED REALT
counter:=0; reneat	lattoion: Artus:
kevstr:=ReadRev;	end;
accept:#true;	if accept then dumstr2:=dumstr2+letter;
if ((keystr=chr(8)) and (counter>=1)) then	end: {loop}
begin	<pre>if (({minus_one and ((dumstr<>'-1'+chr(13)) and (colon<>2))) or</pre>
dumstr:≖copy(dumstr,l,counter-1);	({NOT minus_one} and (colon<>2))) and

writein;	GotoXY (x,y);
writeln('That input time is invalid - try again');	readln(dumstr);
valid:"false; writein;	val (dumstr, ans, error);
end	If dumstrey then ans: default;
else	until ((ans>=min) and (ans<=max)) or ((ans==1) and minus_one);
valid:=true;	input_real:=ens;
end; [while]	end;
compress_clock (default, dumdate) ;	
dundate : =copy (dumdate , 9, 10) ;	
dumstr2:=dumstr2+dumdate;	function log(value:real):real;
if (dumstr=('-1'+chr(13))) then	begin
begin	if value<=0 then log:=0
result.=default;	else log:-Ln(value)/Ln(l0);
result.hour:=255;	end; (log)
end	
else begin	
If (dumstr=chr(13)) then begin	function power_of(a,b:real):real; {calculates a^b}
result:-default;	Degin and the second seco
normalise_clock(result);	
	end; (power_of)
erpand clock dustry result) /	
normalise_clock(result)/	procedure IInd_char(st:str/JJ;ch:char/var pointer:integer);
	Vai Jan - İntener:
	rent construction (
function innut interes/default min.max.interes.mainum one:boolean):interes:	oofnter: woolnter!]:
	until (st[pointer)-ch) or (pointer>-len):
	<pre>if (cointer>=len) and (st[len]<>ch) then pointer:=0;</pre>
dumstr str255	end;
x,y : byte;	
begin	
x:=WhereX;	procedure space;
y:=wherey;	var dummy : char;
repeat	begin
GotoXY (x,y);ClfEol;	writeln ('Press SPACE to continue');
GotoXY {x, y}; #rite(default:6);	repeat
Got oXY (x, y) ;	dumny : -ReadKey ;
readl n (dumetr) ;	until (dummrys' ');
val (dumstr, ans, error) ;	end; [space]
if dumstrant then anso-default;	
until (dansvembr) and (ans<-max)) or ((ans1) and minus_one); Jacut Jatacatiertein	procedure cla; havia
	end; (cla)
function input real(default,min.max:real;minus one:boolean):real;	procedure dashed line;
Var	begin
ans : real;	witeln (`''''''''''''''
error : integer;	end;
dumatr : str255;	
ary : byte: And	function ves or no (default:boolean); boolean;
s:=WhereX;	var
y:=WhereY;	ans,defn : char;
repeat	x,y : byte;
GotoXY (x, y);ClrEol;	begin

·

	end:
if default then defn:"'Y'	<pre>if (timel.hour=time2.hour) and (timel.minute=time2.minute)</pre>
else defn:"'N';	and (timel.second-time2.second) and (timel.day-time2.day)
GotoXY(x,y);ClrEol;	and (time1.wonth=time2.month) and (time1.year=time2.year)
GotoXY(x,y):write(defn);	then ans:«false;
GotoXY (x, y) ;	compare time:=ans;
repeat	end;
ans:"Readkey;	
until (ans='Y') or (ans='y') or (ans='N') or (ans='n') or (ans=chr(13));	
if (ans='Y') or (ans='Y') then yes_or_no:=true;	
if (ans=chr(13)) then yes_or_no:=default;	procedure system_clock(var result:time);
if (ans='N') or (ans='n') then yes_or_no:≖false;	var
end; (yes_or_no)	hr,min,sec.100 : word;
1	yr,mmth,dy,dayofwk : word;
{\	begin
	GetTime (hr.min, sec, sec100);
function Int_to_str(i:integer):string;	GetDate(yr,mnth,dy,dayofwk);
var	result.hour:-hr;
s : strl2;	result.minute:
begin	result.second:-sec;
etr(1, e);	result.dey.=dy,
Int_to_str:=s;	result.month:_muth;
end; ●nd;	result .year:-yyc-1900;
	normelise_clock(result);
	end;
function compare(numl,num2:integer):integer;	
If (numl/shume) then compare:=U;	procedure add_time(intime1,intime2;time;var result;time);
II (UUMIREDUMAL) TREE COMPARES II) is finite (Street Compares in the Street St	
	rest from the former of the fo
	reault muta -intrinet.modit.court.nctmet.nour. reault mitter-mittimel minteri.totimet
function commare time(time).time): boolean: {	rodestration - patranstration - strategy - s
ust de la conferencia de la contraction de la co	realt favelor (mailed) javin (mailed)
and · boolean: · the second of	realt morth triangle that and the set of the first morth.
	really that a first first state of the first state
dusenversetet dusenversetetemel vear.time2.vear):	
if (dumm-0) then ans strue:	function file exists(filename.path:str255):boolean:
if (dummy-1) then begin	Var
dumny : =compare (time1.month, time2.month) ;	fil : str255;
if (dummy=0) then ans:=true;	f : file;
if (dummy=1) then begin	begin
dummy:=compare(time1.day,time2.day);	fil:mpath+^/ +filename;
if (dummy-0) then ans:struc;	{\$I-}
if (dummy-1) then begin	Assign (f, fil);
dummy:=compare(timel.hour,time2.hour);	React (f) /
11 (dumyru) then answerve;	Close (1);
it (dummyrt) Tren begin minite time? atomite). Airman archaerstrimed minite time? atomite).	1414) 1415)
tering a construction of the construction of	riterations of the second second statement of the second s
AI (dummive)) then bench	
compare (time: second, time2, second);	
if (dummy=0) then ans:=true;	function get pathname(oldpath:str255):str255;
end;	Var
end;	answer,dumstr : str255;
end; ●	letter : string[1];

tartpt,len,loop,counter : integer;	if (len>8) then name:=copy(name,l,8);
111 t:=false:	€1]enumber:=0;
hile (NOT ok) do begin	repeat
writeln('Enter the name of the directory path (including drive name)');	dash = -0 ; dash :0 ;
writeln('The old path was ', oldpath);	find char(name,'-',dash);
writeln('Press RETURN to use this path');	dumstr:='';
answer:-input_string;	dumstr2:=''''
if (answer=chr(13)) then ok:=true;	filenumber:=filenumber+1;
dunstr:='';	atr (filenumber, dumatr);
startpt:=0;	if (dash<>0) then dumstr2:=copy(name,1,dash-1)
started:=falge;	else begin
accept:=true;	len:=Length (name) ;
counter : =0;	if (lenc=6) then dumstr2:-name
had_colon:-false;	else dumstr2:=copy(name,1,6)
len:=Length(answer);	end;
ok:=true;	fname:=dumstr2+'-'+dumstr>extension;
for loop:-1 to len do begin	failed:=file_exists(fname, path);
accept:=true;	if (filenumber-99) then failed:"false;
letter:=copy(answer,loop,l);	until (NOT failed);
if ((letter-'') or (letter-chr(13))) then accept:-false;	file_version:=fname;
<pre>if (letter=':') and (counter<>startpt+1) then accept:=false;</pre>	end; [file_version]
II ((LECLERT'!') and accept and (NOT had colon)) then had colon:"true;	
if (accept and (NOT started)) then begin	function get_filename(oldfile:str12;var path:str255;extension:str4;new file:boolean)
startpt:=counter;	
started: "true;	answer,dumstr : str255;
end;	len : integer;
if accept then begin	dumetr2,dumetr3 : str12;
dumstr:=dumstr:letter;	dumchar : string[1];
counter:=counter+1;	loop : integer;
end:	ok, okz : boolean;
end; {loop} is //www.trd colory ord //conversion/) the body	dot : integer;
IY ([WOT had colon) and (anawer/your(13))) then begin -t	
WITCHIN (100 MUST SPECITY & GIIVE!);	
	C) TS.T.
nd: (while)	display directory(math):
f (answer-chr(13)) then get pathname:=oldpath	writeln("Enter the name of the file you wish to use');
else get pathname:=dumstr;	writeln('Enter RETURN to use the old name, DIR to change directory');
[get_pathname]	writeln('Old file was ',oldfile);
	writeln('The ', extension,' extension is automatically added');
tion file version/seme.str]2.neth.str]55.extension.str4]. str]2.	answer:=input arrial; then somer-reconstructure 1 feorth(somer-1-1). 16 [somerch_rial]; then somer-reconstructure 1 feorth(somerch_1)];
file bited antioned (memorial bited for the file file file file file for the file file file file file file file fil	<pre>All distance.com.start.pr) then social-could distance.com.start.pr).desg(intercourt.pr); if (social-court-pr)[]]) then social-could(intercourt.pr).desg(intercourt.pr);</pre>
str : str2; {Adds the version number and the }	if (answer='DIR') of answer='DIR') or (answer='DIR') or (answer='dir')
atr2 : atr6; (extension)	or (answer='diR') or (answer='diR') or (answer='DiR') or
enumber : integer;	(anwer='dlr') then beein
led : boolean;	ok :=false;
me : strl2;	path:=get_pathname(path);
h : integer;	end;
. integer:	until ok;
: Integer;	len:=length(answer);
n. 	IT (len≻IZ) then answer:⊐copy(answer,1,9); ∆ot∩.
strictions ' dat'.	duction find charlenears'' doth'
.nd_cidat(name, ., uoc); * [dot<20] then name:=conv(name.].dot-1]	
else begin	dumstr2:=/ ';

else begin len:=Tenoth(answer):
if (len>8) then answer:=copy(answer,l,8);
end;
dumstr2:=answer+extension;
ok2:=true;
if new_file then get_filename:=file_version(dumstr2,path,extension)
if NOT file exists (dumstr2.path) them ok2:-false
else get filename:-dumstr2
end
until. ok2;
end;
{*************************************
procedure hold(instring:str255);
begin
RestoreCrtMode;
writeln;
writeln(instring);
writeln;
writeln('If using THERNAL CONDUCTIVITY program, change the'+
' meter to the correct scale.'};
writeln('Then press SPACE');
writeln('The graphics display will then need to be re-drawn');
writeln('To do this, press i and then RETURN');
space;
SetGraphMode (GraphMode) ;
end;
<pre>procedure display_directory(path:str255); {Stolen from Roger!! 11/01/90}</pre>
Var
test_key, fnkey : char;
z block, y block, recnum, last rec, last row, last blk colmn : byte;
ntav tar - buta.

no_of_files_per_line:=(screen_width div filename_length);
i := 1; FindFirst(dumstr, Archive + Directory, dir_search);
dir_rec[i] := dir_search;
while DosError = 0 do array[1..225] of SearchRec; no_of_files_per_line : integer; integer; SearchRec; oyue; string[2]; boolean; filename_length : integer; screen_width : integer; : integer; NameStr; ExtStr; DirStr; : str255; filename_length:=13; dumstr:=path+'*.*'; screen width:=80; dir_search dir_rec prev_rec x_string filename dumetr begin begin dir ext Ť, Ř × 5

GotoXY (filename_length*j+2,i); FSplit (dir_rec[(i-1)*no_of_files_per_line+j+1].name, dir, filename, ext) FSplit(dir_rec[(i-1)*no_of_files_per_line+)+1].name,dir,filename,ext); for] := 0 to no_of_files_per_line-1 do begin GotoXY(filename_length*j+2, i); write(filename,ext); for j := 0 to last_blk_colmn-1 do FindWext (dir_search); dir_rec[i] := dir_search; write(filename, ext); while i <> last_row do i := last_row; end; 1 := 1 + 1; begin i := 1; end; writeln; begin end; end; end;

end.

.H 06/8/6 +)	R.Delan	
		•
************	*****	·
unit defaults;		
interface		
uses mrdglob;		
const		
max_setpoints	= 100; {Set	is max number of setpoint changes)
max_devices	= 10; (Sel	s max number of IEEE devices)
	(Not	ce 10 is max allowed by IEEE routines)
		aning to do Witzn mei / a ann annhan af adimta atanna (a Baw)
max points		s maker of rein chances stored in NMT) se number of rein chances i
max_gain_variacions min_rain		
	(W) 2.66 m	ux and minimum permitted values)
min rate	- 0.0:	
	. 66 -	
min reset	- 0.0	
max reset	- 99.;	
min_heater power	- 0; {S	et max and min heater ranges}
max heater power	- 5;	
nV minutes on	= 1; {Si	st how long nV meter is switched on}
nV seconds on	q) :0 -	efore a reading is taken
ieee no of channels	= 31;	
no_of_lines	= 17;	
more than	- 0:	codes returned from compare routines)
equal	- 1;	
less than	- 2;	
pc ieee	- 1;	(IEEE address of PC)
default_interval	<pre>- '00:04:59';</pre>	{Default time between readings}
default_pause	- '00:24:59';	(Default pause after set point change)
default_path	- C:\MARTIN\	Thermall's
default_filename	- 'DATA';	
graph_path	- 'C:\TP5\GRA	на, :
default_end_time	- '96:00:01';	
null_date	,00/00/00-, =	
minicam_on	- '128' ;	
minicam_off	:,0, =	
top_no_of_lines	- 7;	
pixel_line_height	- 10;	
default_setpoint_ho	our = 0;	
default_setpoint_mi	n = 1;	
default_setpoint_se	cond = 0;	
default_setpoint_in	iterval_time = 1	
thermocouple_defaul	L_file = 'C:	MARTIN\AUFE.CAL';
dr9lcinitetr1 = 2	COMITO'; (EOI st	atus, remote mode, terminator = LF CR}
aryteinitstra =	FURFINESAS ; } {	UTLES = NeLVIN, FESOLUCION = XX.XXX)
dr9lcinitatr2 = 'FIS	SF3A4'/ (unit	s = Volts resolution = X.XXXX}
kl97initstr = 'G	ToV) ; CTODOTS	cs, U-2V]
kl75initstr = 'G	SOKODOT4'; (Amp	5, 0-20maA]

	: integer;	: integer;	: integer;	: integer;	: real;	: array [1.	: array [1.	: boolean;	: boolean;	: integer;	: boolean;	: array [].	: text;	: integer:
: integer: : str8; : integer;	fluke8860,k197	iolar, weston, dr91c	Ę	lode			emperature.	jain	cale	je_readings	1	ĥ	1.	inel.date line2

-

record		
device : integer;		
name : str8; 		
cnan : inceger; end;		
VAL		
bdeal, fluke8860, k197	: inte	ger;
k175, solar, weston, dr91c	: inte	ger;
minicam	: inte	ger;
GraphMode	: inte	jger;
	: real	
aure fa tamaarataa	arra	THEAT TO LOCATE IT IN THE TO LOCATE IT.
aute comperatore auto cein		Yaar: Aar:
auto gcale	lood :	
average readings	: inte	
colour	: bool	.ean;
current	: arra	y [1max_points] of real;
datafile	: text	
data_linel, data_line2	: inte	ger;
data_line3	: inte	ider;
data_time	: 3573	vy [1max_points] of str12;
device_name	: arra	y [lieee_no_of_channels] of str20;
device_default_chan	STLA	IV [11eee_no_of_channels] of integer;
av, dr ·		IV (L. Max Doinca) of real?
ena cuae filaname	s strl	2.
free time	: time	
gain variation	arra	y [1max gain variations] of real:
graph X limit	: inte	
graph_Y_limit	: inte	9 e r;
graphset	: bool	
heater_range_variation	: arra	<pre>iy [1max_gain_variations] of integer;</pre>
Imeter	: mete	5K8;
interval	: time	
kval	: arra	y [1max_points] of real;
last_setpoint	. real	••
ilengun mannal offaat		-
max time. min time	- time	
max X value. max Y value	: inte	06 E I :
minicam address	: str5	
next_reading_time	: time	
now time	: time	
no_more_setpoint_changes	lood :	lean;
no_of_devices	: inte	sger;
no_or_gain_variations	: Jnte	-der;
no_or_setpoints	: 1776 	99er;
nu points in piock numeter nuccetrol		
averet, avointot av ontime		
nV nower.nV old status		
n'Lower	i bool	
nVscale	: real	
next setpoint temperature	: real	
next setpoint time	: time	Ĩ
outfile, path	: str2	255,
old_time	: time	

pause flag	: boolean;	EXTER SPECIAL DEWARET UV NUT DAVE ANY LEEE GEVICE NUMBER 1 AS THE Computer IEEE routines assume that the computer is address 1}
point	: integer;	
power	: array [1max_points] of real;	begin
rate variation	: array [1max_gain_variations] of real;	
reading_taken	: boolean;	{ The following are 'physical' device numbers note that the parameter
reset_variation	: array []max gain variations] of real;	names need to be defined in the VAR statement at the top of this unit)
screen_X_axis	: integer;	· · · · · · · · · · · · · · · · · · ·
SCTOON I GAIS Setsoint indev	: interest	ILUKE0000:"L; P107.=2.
setuciate conted	· boolean:	110
aerpointa soitea aetooint tima	: DOUTERIN • arrau (0 max meteonintel of time:	
actorist tennerature	· array (v	
start time	· · · · · · · · · · · · · · · · · · ·	seston:=6:
start str	: str20;	minicam:=7;
setpt	: array []max points] of real;	no_of_devices:=7;
temp	: array [1max_points] of real;	1
temp_variation_lower	: array [1max gain variations] of real;	(default_chan gives the IEEE address of each particular device, they
temp_variation_upper ***^*	: array []max_gain_variations] of real; 	may be altered by the main program if the user vishes
title linel title line?	: 10140.001	minicam addresa:="42";
title line3	: integer;	device name[fluke8860]:='fluke 8860 ';
total lines	: integer;	device default chan[fluke8860]:=16;
top extra lines	: integer;	device_name[k197]:='Keithley 197 ',
gain, rate, rreset	: array [1max_points] of real;	device_default_chan[k197]:=2;
Tcontrol	: meters;	device_name[kl75]:-'Keithley 175 ', ;
timeout	: boolean;	device_default_chan[k175]:=3;
thermocouple_file	: str255 <i>i</i>	device_name[dr91c]:='DR91C controller ';
~	: array [1max_points] of real;	device default_chan[dr91c]:=4;
Vfsd	: real;	device_name[solar]:='Solartron';
Vmeter	: meters;	device_default_chan[solar]:=7;
width	: real;	device name (weston) := Meston ';
xborder_lerc	: Integer;	
xborder_right	: integer;	device name (nintem) := Minicam board
xborder_left_thickness	: integer;	device_default_chan[minicam]:=6;
xborder_right_thickness	: integer;	
xleft, xright, ytop, ybottom	: integer;	Imeter.device:=k175;
amin, amax, imin, imax		
Xstep, Ystep	: real;	
yporder_cop	: inceger; 	JUNCON COLLEGE TO PLOT
yourder too thickness	. integer: . tstacer:	LACOULD ALCONNENT AND AND ALCONNENT AND ALCONNENT AND ALCONNENT AND ALCONNENT AND AND ALCONNENT A
yourder bottom thickness	· interstary	
vheight vlenth		
vlabel position	: integer:	procedure gain variation defaults;
xlabel position	: integer;	
ycentre, xcentre	: integer;	(At different temperatures, the PID settings will need to vary
		this sets the default configurations
procedure meter_defaults;		If you add any more, make sure that you change
procedure gain variation det	faults;	no_gain variations at the end of this procedure}
procedure var_gerautts;		harin
		temp variation lower[1]:=10.;
implementation		temp_variation_upper[1]:=15.;
		<pre>gain_variation[1]:=0.2;</pre>
		rate_variation[1]:=0.;
procedure meter defaults;		Zeset_Variation[1]:=U.; haatar_ranna wariation[1]:=1.
(This procedure sets up the	default meter configurations	
and is called from DEVICE_	DEFAULTS in the main program	temp_variation_lower(2):=15.;

gain_variation[2]:=0.3; rate_variation[2]:=0.; remet_variation[2]:=0.; remet_variation[2]:=0.; temp_variation_lower[3]:=20.; temp_variation[3]:=0.0; rate_variation[3]:=0.0; heater_range_variation[3]:=4;

temp_variation_lower[4]:=30.; temp_variation_upper[4]:=40.; gain_variation[4]:=0.0; rate_variation[4]:=0.0; heater_range_variation[4]:=4; temp_variation_lower[5]:=40.; temp_variation_upper[5]:=50.; gain_variation[5]:=0.5; rate_variation[5]:=0.0; reset_variation[5]:=0.0; heater_range_variation[5]:=4; temp_variation_lower[6]:=50.; temp_variation_upper[6]:=60.; gain_variation[6]:=10; rate_variation[6]:=0.0; reset_variation[6]:=0.0; heater_range_variation[6]:=5; temp_variation_lower[7]:=60.; temp_variation[1]:=80.; gain_variation[7]:=20.; rate_variation[7]:=0.0; reaet_variation[7]:=0.0; heater_range_variation[7]:=5; temp_variation_lower[0]:=80.; temp_variation_upper[8]:=300.; gain_variation[8]:=20.; rate_variation[8]:=0.0; reset_variation[8]:=0.0; heater_range_variation[8]:=5; no_of_gain_variations:=8; end;

procedure var_defaults;

(This procedure sets up some of the sample and meter configurations and is called from VARIABLE_DEFAULTS in the main program)

begin width:=3; thick:=3; (sizes in millimetres) llength:=6; area:=width*thick;

N'scale:=30; Vfad:=0.98; N'power:=true; manual_offaet:=0; average_readings:=1; no_points_in_block:=1; auto_gain:=true; end;

end.

(DNIT DEVICES	function dal(madd,n : str5) : str255;
<pre>(Contains the routines for reading/writing to the DR9IC,Keithley 197£175)</pre>	procedure fluke8860_com(comstr:strl2);
(Schlumberger Solartron and Fluke 8840 DMMs	procedure creok;
(М.К.ЮЕТДР) 1 NOVEmber 1989 актеритернатернатернатернатернатернатернатерна	procedure init_fluke8860; function read fluke8860 : double:
-	function ave_fluke8860 : double;
unit devices;	**************************************
	implementation
[Note that the set temperature is in volts and that the conversion }	
<pre>{ routine for the new sensor is still here </pre>	
	procedure init_ieee;
interface	Degin cntrllr:∎true:
	my_flag:=false;
uses defaulte, ardg lob, ieee, mrdutils, crt;	my_addr:=1;
	intlatat:=0;
procedute intrijetej function droit cuedructratral, ven chari, stren.	init; and: [init incol
function dryle com write(filename:name), errefictionerrefreficherchar):errefic	
procedure init_dr9]c;	{ ************************************
function dr91c_read_sensor: real;	
function dr91c_read_setpoint: real;	function dr91c_com(comstr:str80;rep:char):str80;
function convert_temp(temp:real):real;	Var
procedure drylc write setpoint (temp:real); f	Lumstr : stroj
runction dryle read gain: real; 	Contract : real
procedure dryste write gain (value:real)/ function droit vand transt tal.	uibed
proceedure dryk reset(value:real); brocednre dryk write reset(value:real);	Intr_rever Acts=chr(10):
function dryle rade: real;	
procedure dr91c write rate(value:real);	dataatring:=comstr+chr(13);
function dr91c_read_heater_power: real;	wr_etr(dr91c);
procedure dr91c_write_beater_range(value:integer);	if (rep ^{_R}) then begin
function decipher_keithley(instrictr80; digits:integer): real;	read_str(dr91c);
runction KJ97 cometristr80; repicbar; var rangeistr3): real; 	dr91c_com:=datastring;
Procedure Init Mile Viewer (Artandad.ers]. Nociana. Function 1137 Teams (Artandad.ers2). Nociana.	
ruction k175 conformatristR00; recider 2012); Doureau; function k175 conformatristR00; recider var randertr3); real;	ETRE OTATC COMPLET ; ;
function k175 range check(intended:str1): boolean;	
procedure init_k175,	
function solar_com(comstr:str80; rep:char): real;	function dr91c_com_write(filename:namfil;comstr:str80;rep:char):str80;
procedure init solar!	Var , , , , , , , , , , , , , , , , , , ,
runicion wescon come comentiatravity ireal; ir	
riveduate initiamendi. function read temp(device:integer):seal:	cummy : real/ heafn
function read aetpoint (device integer : real;	tinit lees
procedure write_eetpoint (device:integer; value:real);	eois:=chr(10);
function read_gain(device:integer): real;	iors:='i';
procedure write_gain(device:integer; value:real);	datastring:=comstr+chr(13);
runction read refe (device:integr): real; 	wr_str(dr91c);
Procedure write fate(devrice:integor/value:real); f	if (rep='R') then begin
unicton faq Teat (aevice:integr): faal; nroodning unita inter raaf (device:integr): faal;	read to file(filename, dr91c); Leni of the filename (dr91c);
procedure mille from trading from the f	OFFLC_COM_WEILE:=Galastring; and
punction read_meter range (device:integer; value:integer);	enu else dr91c com write:=' ':
function read current (device:integer): real;	end;
function read_volt(device:integer): real;	
function read_nV(device:integer): real;	

var dumstr : str80;

dumstr:==dr91c_com(dr91cinitstr1,'N');
dumstr:==dr91c_com(dr91cinitstr2,'N'); begin end;

.

function dr91c_read_sensor : real; dumatr:=dr91c_com('N0','R'); dumstr:=copy (dumstr, 1, 7); dr91c_read_sensor:=dummy2; val (dumstr, dumny2, dumny); dummy : integer; dumstr : str80; dummy2 : real; begin Var

function dr91c_read_setpoint: real;

end;

dumstr:=dr91c com('W0','R'); dr91c_read_setpoint:=dummy2; dumstr:=copy (dumstr, 19, 6) ; val (dumetr, dummy2, dummy); dummy : integer; dumstr : str80; dummy2 : real; begin end; Var

function convert_temp(temp:real):real;

{Coefficients taken from fitten 7th order chebyshev polynomial} coeff[5]:=4.3326955522e-10; coeff[6]:=-7.0226932006e-13; coeff[2]:=-9.3711538784e-4; coeff[3]:=1.1101416234e-5; coeff[4]:=-1.2055536247e-7; sum:=x1*coeff[1]+x2*coeff[2]; coeff : Array[1..12] of real; coeff[7]:=4.2746418359e-16; sum:=sum+coeff[loop]*x; for loop:=3 to 7 do begin coeff[1]:=1.4154967767; x:=2*temp*x2-x1; x,xl,x2,sum : real; xl:=x2; x2:=x; convert_temp:=sum; loop: integer; x2:=temp*;* xl:=l; begin end; end; TON

procedure dr9lc_write_setpoint(temp:real);

dumstr:=dr91c_com('S'+dumstr,'N') volt:=convert_Temp(temp); str(temp:5:2,dumstr); str(volt:6:4, dumstr); dumstr : str80; volt : real; begin

end;

function dr91c_read_gain : real; dumstr:-dr91c_com('M3','R'); dumstr:=copy (dumstr, 1, 3) ; val (dumstr, dumny, error); dr91c_read_gain:-dummy; : integer; dumstr : str80; : real; error dummy begin end; Var

procedure dr91c_write_gain(value:real); dumstr:=dr9lc_com('P'+dumstr,'N'); str (value:4:1, dumstr); dumstr : str80; begin end; TRA

function dr91c_read_reset : real; dumstr:=dr91c com('W3','R'); dumstr:=copy (dumstr, 9, 3) ; val (dumatr, dumny, error); dr91c_read_reset:=dummy; error : integer; dumstr : str80; dummy : real; begin end; VAL

procedure dr9lc_write_reset(value:real); dumstr:=dr9lc_com('I'+dumstr,'N'); str(value:5, dumstr); var dumstr : str80; begin end;

function dr91c_read_rate : real; dumstr : str80; Var

error : integer; dummy : real;

dumstr:=dr91c com('W3','R');	function k197 com(comstristr80: repichar: var range:str3) : real:
dumstr=copy (dumstr, 5, 3);	begin
val (dumstr,dumeny,error);	init jeee,
dr9lc_read_rate:-dummy;	eois:=chr(10);
end;	iors:='1';
	<pre>datastring:=comstr+'X'+chr(l3);</pre>
	Wr str (k197);
procedure drylc_wile_rate(value:real);	TATA COM:=C.C.
	TT VIAGUAR (A) COMPANY
	14444 ロビードメンタイプ ドード・ドード・ドード・ドード・ドー・ドー・ドー
11.1.441.44	KIN' COMPERCIPATINEY (GREASELINEY (GREASELING, J)
dumst:=atstc_com/ v turnett, v /;	range: =copy (datastring, 2, 3); =nd -
	end;
function drgle read heater nomer : real:	
	arocadita (ait 1107.
dummatr : str80;	
dummy : real;	dumerv : real:
error : integer;	dumstr : str3;
begin	begin
dumstr:=dr91c_com('M3','R');	sel_dev_rem(k197);
dumstr:=copy (dumstr, 15, 3);	dummy:=k197_com(k197initstr,'N',dumstr);
va.(dumetr.dummy,error);	end;
dr91c_read_beater_power:=dummy;	-
	runction Kly/_range_cneck(intended:strj) : boolean;
orocedure dr91c write heater rance(value:integer);	var disenv · real·
t value = 0 heater off	dumetr: str3:
1heater off	bertin
210e-3	dummy:-k197 com('G0','R',dumstr);
310e-2	if (dumstr<>intended) then k197 range check:=false
410e-1	else k197_range_check:=true;
5 ma x }	end;
Var	
dumatr : string[1];	
dumetr2 : str20;	function k175_com(comstr:str80; rep:char; var range:str3) : real;
	begin
	eols:=cnr(10);
	lotori 1. j Gataetring.morett'⊻'+rhrf]3).
	wr atr (k175);
<pre>(********Keithley 197 & 175 routines************************************</pre>	k175_com:=0.0;
	range:=' ';
	if (repervi) then begin
Var 4	read_str(K175); tiretertition(2);
diamay : feal, f	KI/J_COM:=Geclpner_Keltnley(Gatastring,9);
cummit : integer, dumeit : arra0:	Lange: -copy (Gardarring, 2, 3); and
begin	end;
dunmy2:=digits+6;	
dumstr:=copy(instr,5,dummy2);	
val (dumst; dummy; dumy2);	function k175_range_check(intended:str3) : boolean;
decipher_keithiey:=dummy;	Var
end;	dummy : real; dummy :
	CIBER S ALLOS

dumanv:=k175 com('','R', dumatr);	
if (dumstr<>intended) then k175_range_check:=false	-
else kl75_range_check:=true;	
end;	. In more meters are acced then the tollowing fourthes should be altered at to mode to the new device.
	* where consider the meters are asked to confirm the display units a * where consider, the meters are asked to confirm the display units *
	the second second of a Solution of the solution of the solutions that the
Proceeding intraviation	
287 	(中学家中学校学校学校学校学校学校学校学校学校学校学校学校学校学校学校学校学校学校
Outwing : 1494.1 Automote : 1447.1	-
ounst. stad	
egidev rem(k175); sejdev rem(k175);	function read temp(device:integer) : real;
dummov:=kl15 contkl75initstr,'N', dumstr);	begin
	if (device=dr91c) then read_temp:=dr91c_read_sensor =1== read_temp:=9090
function solar_com(comstr:str80; rep:char) : real;	
Var	function read_setpoint(device:integer) : real;
dummy : real;	begin
dummy2 : integer;	if (device-dr9ic) then read setpoint:-dr9ic_read setpoint
dumstr : strl2;	L-AV-1-THAT BARA TABA TABA TABA TABA TABA TABA TA
begin begin	
11116 Oois:=chr (13);	
	procedure write setpoint (device:integer; value:real);
datastring:=comstr;	var dumstr : str20;
wr_etr(solar);	begin
solar_com:=0.0;	if (device-dr91c) then dr91c_write_setpoint(value);
if (rep='R') then begin	end;
read strikesty ;	
	. say . tawadah animahi atau basa anitong
Vel (ourse)(r, ourse)(r, ourse)(r) strt(diverse)(r)	
st tourstatury,	dummv : real;
	beein
end;	if (device=dr91c) then dummy:=dr91c read gain;
	•
	read_gain:=dummy; and:
PLOCEDURE INIL_BOLAL?	
dummy : real;	
begin	procedure write_gain(device:integer;value:real);
sel_dev_rem(solar);	begin
dummy:=solar_com(solarinitstr,'N');	If (device=dryic) then dryic_write_gain(value);
end;	end;
{	
function weston_com(comstr:str80; rep:char):real;	function read_rate(device:integer) : real;
begin	Var
writein ('This routine is not written yet');	Lummy : real;
end;	begin if (device=dr91c) then dummy:=dr91c_read_rate;
arteredure fait weepon.	read rate:mdummy:
procedure Intr_weston; begin	end;
sel dev rem(weston);	
writeIn('Init_weston not written yet');	

begin	until range_ok;
if (device=dr91c) then dr91c_write_rate(value);	end;
end:	
	function read_volt(device:integer) : real; var
function read reset(device:integer) : real;	range ok : boolean;
Jean State	. dumatr : str3;
dummy : real;	begin
begin	repeat
if (device-dr91c) then dummy:-dr91c_read_reset;	begin
	range_ok:=true;
read_reset:-dummy;	if (device=kl97) then begin
end;	<pre>read_volt:=k197_com('','K',dumetr); range_ok:=k197_range_check('DCV');</pre>
	ipue
procedure write_reset (device : integer; value : real);	if (device-ki75) then begin
begin if (device=dr9ic) then dr9ic write reset(value):	read_volt:=El/2_com('','K',dumatr); range ok:=k175 range check('DCV');
	endi
end;	<pre>if (device=solar) then read_volt==solar_com('T0','R');</pre>
	if (NOT range ok) then read voltimeston fown ov , //
function read beater power(device:integer) : real;	hold('The voltage meter is not correctly set. '+
Var.	'Presently it is '+dumstr);
dummy : real <i>;</i>	end;
begin	until range_ok;
if (device=dr9lc) then dumny:=dr9lc_read_heater_power;	end;
read heater power:=duemu:	
	function read nV(device:integer) : real;
-	Var
procedure write heater range (device: integer; value: integer);	range ok : boolean;
begin	dumstr : str3;
if (device=dr9lc) then dr9lc_write_heater_range(value);	begin
end;	repeat hunda
	range ok merine 6 (Aunionaliji than barin
unction Kead_cuitent(Gevice:integet) : teat; ****	AI (GENERALS), CHEAN BOUGHT DOWN, ''', R' (HUMALT): Trad Ny:=k197 com('', 'R' (HUMALT):
rance of the bouleas:	rance off-rance check('DCV');
dumatr : strj:	end
begin	if (device-k175) then begin
repeat	read_N':=k175_com{'', ''R', dumatr);
begin	range_ok:=#175_range_check ('DCV');
range ok:*true;	end;
if (device-kl97) then begin	if (device=solar) then read_NY:=solar_com('TO','R');
read_current:≈k197_com('', 'R', dumstr);	if (device-weston) then read_nV:-weston_com('U0','K');
range_ok:=k197_range_check('DCA');	if (NOT range_ok) then
endi	hold (The nanovoitmeter meter is not correctly set.'+
if (device-k175) then begin	. Fresently IC 15 'toumstry';
read current: #1/3 com (, , , , dumerr);	
range_ok:=K1/3_range_cneck('UCA'); =nd:	uncii range ok; end:
if (device=solar) then read current:=solar com('TO','R');	
if (device-weston) then read current:-weston com('UO','R');	
if (NOT range ok) then	{ ************************************
hold('The current meter is not correctly set. '+	
'Presently it is, '+dumstr);	<pre>{* Minicam Routines - Version One *}</pre>
÷ Ŧ 5 /* Altered by M.R.Delap for use with thermal conductivity experiment
/* 24 November 1989 procedure minicam_com(comstr : str20); function dal(madd, n : str5) : str255; datastring:=comstr+chr(13); read_str(minicam); wr_str(minicam); my_flag:=false; my_addr:=l; eois:=chr(13); cntrllr:=true; intlstat:=0; iors:='s'; init; begin end:

function dal(madd,n : str5) : str255; var comstr : str20; begin comstr:='DAl,'+madd+','+n; comstr:='DAl,'+madd+','+n; dal:=datastr1ng; dal:=datastr1ng; end; procedure fluke8860_com(comstr:strl2); begin eois:=chr(10); iots:='i'; datstring:=comstr+chr(13)+chr(10); init; wr_str(fluke8860); wr_str(fluke8860); end;

procedure cr_eoi; begin

fluke8860_com('M2'); end;

procedure init_fluke8860; begin

egin eois:=chr(10); iors:='i'; tcsy;
dataEtring:=copy(datastring,l,length(datastring)-2);
val(datastring,dum,ec);
read_fluke8860:=dum; function read fluke8860; read str(fluke8860); ec : integer; var dum : double; eois:=chr(10); wr_byte(\$30); wr_byte(\$0d); wr_byte(\$0a); last:-false; last:=true; 1018:m'1'; set eol; gt sby; tcay; tcsy; begin end;

function ave_fluke8860; var sum : double; i : integer; begin sum:=0.0; for i:=1 to 10 do begin sum:read_fluke8860; sum:=sum+read_fluke8860; end: ave_fluke8860:=sum/10.0; end;

end;

end.

(* *	procedure wr str(devnum:int	eder);
(* UNIT ieee - Pascal routines for Scientific Solutions RevD board *)	procedure wr from file(film	am:namfil;devnum:integer);
(* 12/04/89 D.B.Lambrick *)	procedure rd byte;	•
	procedure wr byte (dat:integ	er);
(* Rd_Byte Amended 7/7/88 *)	procedure par_poll;	
(* Conversion to Turbo 5.0 21/3/89 - Unit conv. 12/04/89 *)	procedure unconfig;	
(*	procedure disab_poll (devnum	: integer);
(*************************************	procedure read bus stat;	1
	procedure sel dev clr (devnu	m: integer) ;
unit ieee;	procedure sel_dev_rem(devnu	m:integer);
	procedure ser_poll(devnum: h	nteger);
{	procedure pp_en(devnum:inte	ger);
	procedure all_loc;	
interface	procedure all_rem;	
	procedure trnsfr (devnum:int	eger);
uses mrdglob,mrdutils;	procedure my_la;	
	procedure my_ta;	
const Bdadd : integer = \$310;	procedure listen (devnum:int	eger);
	procedure talk (devnum:integ	er);
type	procedure unlisten;	
laddr - record	procedure untalk;	
prim : integer;	procedure tcay;	
sec : array[110] of integer;	procedure tasy;	
	procedure rcv_cntrl;	
devadd = array[110] of laddr;	procedure pass_cntrl(devnum	::integer);
stray = array[110] of string[255];	procedure set eoi;	
	procedure rdsup(devnum:inte	ger);
	procedure rd_arry(sar:stray	;sep,lastchar:char;devnum:integer);
intlatat, my_addr, lastinti, dat, poil_resp, 10005atus, srq : integer;	procedure zero_add;	
DIL; SADSe; l, l : Anteger;		
relations to chart and the second sec		
CICTLIF.LESC.MY.LING.LING.LINGLENCE.LVE: DOOLEAN; Listertier	1 mp/tenentation	
tortary perturbed a statut for		[Weite for the local huffer to sloer]
	procedure inclear;	(MALLS INF LINE INPUT PULIER TO CLORE) / / _ 2000 hitter)
	Tabanur : Aumma Jea	Jaiing 7670 .a.t
Conversion and the second s	utbac	
PCILLE : SCIL4/	dutany:=Fort[Bated+9] 	
	antes (manual and a) at	
procedure includes		
proceeding matrix/	aroadire unit12.	
	procedure waiting	
procedure recon la(gevoum:nreger); erroradire of the fricterent)	Ver W1200M : Integer;	
procedure (aft. procedure (aft.	ranat	
proceeding a stabut	w]2dnm:=Port[Rdadd+1];	
procedure go idle;	until (w12dum and 2) = 2;	
procedure dtscnt;	end:	
procedure abrt;		
procedure abort;	procedure tci (comm: integer);	
procedure devclr;	var dummy : integer;	
procedure getrigg(devnum:integer);	begin	
procedure devloc(devnum:integer);	ibclear;	
procedure llo;	<pre>Port [Bdadd+10] :=0;</pre>	
procedure my_rsp(devnum:integer);	Port [Bdadd+9] : =comm;	
procedure read_setup(devnum:integer);	repeat	
procedure read end;	dummy :=Port [Bdadd+12]	
procedure read ar deevum: nreger; /	until (dumany and 1) <> U;	
procedure read to titetrituam: namititucevnum; integer /;	cno;	

•

170502

.

8291A Aux mode B, enable cpt } while (devla[devnum].sec[j] <> 0) {or (j=1)} do [Read 8292 cont status reg] 8291A addr mode - bd is 1) 8291A serial poll mode) if devla[devnum].prim = my_addr then my_flag:=true Port [Bdadd+0] :=devla[devnum].sec[j]+\$60; 8291A intrr enble 1 } 8291A intrr emble 2 } Port [Bdadd+0] :=devla[devnum] .prim+\$20; 8291A Aux mode A) Wait for IBF low } { CIr 8292 err mask Disable addr 1 } Init T1 delay } Port [Bdadd+0] :=devta[devnum] . sec[j]+\$60; Reset 8292 } Reset 8291A } { Enable tci } Clear IBF) Port [Bdadd+0] :=devta[devnum].prim+\$40; while (devta[devnum].sec[j] <> 0) do until (dummy and 1) <> 0; Port[Bdadd+4]:=\$80; Port[Bdadd+5]:=\$0; var dummy : integer; dummy:=Port[Bdadd+9] Port [Bdadd+5] :=\$a1; dummy:=Port [bdadd]; Port [Bdadd+5] :=\$2; Port [Bdadd+5] :=\$25; Port [Bdadd+1] :=\$93; Port [Bdadd+6] :=\$e0; Port [Bdadd+8] :=\$a0; Port [Bdadd+9] :=\$f2; Port [Bdadd+5] :=\$80; Port [Bdadd+2] :=\$0;
Port [Bdadd+3] :=\$0; Port [Bdadd+6] :=\$1; procedure fetch la: j:=j+1; Port [Bdadd+8] :=\$0; procedure out ta; begin var j : integer; procedure gtsby; j:=1; end; var j :integer; procedure init; else begin end; tci (\$e6); j:=j+l; wait12; ibclear; ibclear; ibclear; wait12, begin repeat):=1; end; begin begin begin begin sud; end; end;

begin if cntrllr then abrt else Port [Bdadd+4] :=\$80; Port [Bdadd+5] :=\$0; Port [Bdadd+0] :=\$14; Port [Bdadd+0] :=\$3f; Port [Bdadd+0] :=\$3f; Port [Bdadd+0] :=\$11; Port [Bdadd+9] :=\$f9; fetch_la (devnum) ; Port [Bdadd+0] :=\$8; procedure go_idle; procedure getrigg; fetch_la (devnum) ; Port [Bdadd+0] :=1; procedure gtscnt; cntrllr:=true; procedure devclr; procedure devloc; procedure abort; procedure abrt; procedure llo; tci (\$f9); tci (\$f4); tc1(\$f1); ibclear; wait12; waitl2; wait12; wait12; wait12; begin wait12; begin begin end; begin begin begin begin begin end; end; end; end; ;pue ;pua; end; puq;

procedure my_rsp;

tci (\$f6);

out ta (devnum) :	read and:
waltl2;	end;
Port [Bdadd+0] : "my_addr+\$20;	
wait12;	procedure read_to_file;
Port [Bdadd+4] :=\$40;	var datout : text;
Port [Bdadd+5] : =0;	iobyt : integer;
tci (\$f6);	achar : string[1];
end	loop : integer;
	begin
procedure read setup;	1000:=0;
var dummuv : integer;	assion (datout, filnam) :
bedin	rewrite (datout);
if cois="" then dummy:=13 else dummy:=ord(eois);	read setup(devnum);
Port [Bdadd+7] : =dummy;	writeln('Done setup');
if iors='s' then Port[Bdadd+5]:=\$86 else Port[Bdadd+5]:=\$82;	space;
if my flag then my rsp(devnum);	intlstat:=Port[Bdadd+1];
if cutrilt then begin	while ((intistat and 1)=1) and ((intistat and \$11)<>\$11) do
wait12;	begin
Port [Bdadd+0] : =\$3f ;	loop:=loop+1;
my_rsp(devnum);	1obyt :=Port [Bdad4+0] ;
end;	achar:=chr(iobyt);
if not (my_flag or cntrllr) then repeat	write (datout, achar);
diumy := Port (Bdadd+4);	write(acher);
until (dummy and 4)=0;	if loop>=80 then begin
end;	loop:-1;
	write (datout, chr (13));
procedure read end;	
Variation integer;	LNTLBTAT:=#Fort (BOAQQ+L);
inguilte the market	
l coulil use acqui	
	c toge (datout);
	end;
Fort [Dataday] : "\$J: Data [] - [
1	
	Val a many : integar;
	Legan the state of
else fogin	II CULTIF Then begin
rorc [boacott]:=98U;] ==+1+1:	Valteli <i>j</i>
10441111111111111111111111111111111111	
autications. Autications.	Dorf [Reladd+0] : ##W addr+S40:
end;	Port [Bdadd+0] :=\$3f;
	fetch_la (devnum) ;
procedure read_str;	waitl2;
var lobyt : integer;	gtaby;
achar : string[1];	
	else begin
	Legrence - Bost (Dialized 1). Annews - Bost (Dialized 1).
recent any	until ({dummy and 2)=0):
intlstat:=port[Bdadd+1];	if not ((lastint1 and 2)=2) then wait12;
if ((intlstat and 1)<>0) then	end;
begin	end;
iobyt:=Port[Bdadd+0];	·
achar:=chr(iobyt);	procedure wr_str;
datastring:=datastring+achar;	var lobyt,strcnt : integer;
<pre>{ write(datafile,achar);}</pre>	schar : string[1];
end;	achar : char;

begin	if cntrllr then tci(\$fd);
write_setup(devnum);	end;
for strent:=1 to length(datastring)-1 do	arccadura rd huta.
regult 	
schar = copy detast ing, stront, i/;	
lobyt : -ord (achar) /	intistat:=Port[Bdadd+1];
Port [Bdadd+0] := iobyt;	until ((intistat and 1)=1) or ((intistat and 510)=510);
WAILIS	under under Date (ork der Jennigen) in
end	IT (Included and $T_{r} = (T_{r}) + T_{r} = T_{r}$
schar:=copy(datastring,length(datastring),1);	dat: "Port [Bdadd+0];
achar:=schar[1];	last: -true/
jobyt:=ord(achar);	end .
if iors="1" then begin	else
	begin
Port[Bdadd+0]:=iobyt;	dat:=256;
pue	last:=true;
else begin	end;
if acher <> eols then begin	
Port [Bdadd+0] := 1 obyt ;	else dat:=Port[Bdadd+0];
str (lobyt, dumstr) ;	
hold (dumatr); 	end;
	and the property of the second s
boott (butaddan)d (and an an	procedure w. Dyre;
	11 NOT Last Green wailing 16 over 1900 total states (christer)).
148611011:4101184467	IT NOT AGE CHEM WITH CONTINUES OF A
LI CULIII LUGU COY?	TT TEAL LINEN DOTALLY (GALT)
ena;	ena;
procedure wr from file:	procedure par poll;
var datin : File of char;	var dummy : integer;
schar : string[1];	begin
achar : char;	Port [Bdadd+4] :=\$40;
begin	Port (Bdadd+5) : =0;
assign (datin, film an);	repeat
reset (datin);	dummy :=Port (Bdadd+9) ;
write_setup(devnum);	until ((dumy and 2)<>2);
repear	Port (Bdadd+9) :=515;
read(datin,achar); 	repeat A
White Hot colleaning do	dummy -rot cloandari) :
Post Post [Bdadd+0] ==ord (achar) :	cont. Trans. Total Reladed): coll resc: Port [Reladed):
wait12;	Port [Bdadd+4] = \$80;
end:	Port [Bdadd+5] :=\$0;
until eof(datin);	end;
if lors='i' then begin	
Port [Bdadd+5] :=\$6;	procedure unconfig;
Port[Bdadd+0] :=ord(achar);	begin
end	wait12;
else Degin (f schar /) acts than hants	rott (bdadgtv):"?12; And
LL ACHAI VY COLA LHTH VYYL Port (Bdadd+0) :=ord (achar) :	
wait12;	procedure disab poll;
end;	begin
Port [Bdadd+0] :=ord(eois);	wait12;
end;	Port [Bdadd+0]:=\$3f;
close (datin);	fetch_la(devnum);

Port [Bdadd+0] :=\$70; Port [Bdadd+0] :=\$5; unconfig; wait12; end;

until not ((dummy and 1)=0); ieeestatus:=Port[Bdadd+8]; srg:=ieeestatus and 1; procedure read bus stat; dummy:=Port [Bdadd+9]; var dummy : integer; Port [Bdadd+9] :=\$e7; tci (\$e7); repeat begin end;

procedure sel_dev_clr; Port [Bdadd+0] :=\$3f; Port [Bdadd+0] :=\$4; fetch_la (devnum) ; wait12; wait12; begin ;pue procedure sel_dev_rem; Port [Bdadd+0] :=\$3f; } fetch_la (devnum) ; tci (\$f8); gtsby; } wait12; gtsby; tcsy; begin end;

until not ((dummy and 1)=0); poll_resp:=Port[Bdadd+0]; dummy:=Port [Bdadd+1]; Port [Bdadd+4] :=\$80; Port [Bdadd+5] :=\$40; Port [Bdadd+0] :=\$18; Port [Bdadd+0] :=\$3f; Port [Bdadd+4] :=\$40; Port [Bdadd+0] :=\$19; var dummy : integer; procedure ser poll; Port [Bdadd+5] :=0; out ta (devnum) ; tci (\$fd) ; wait12; wait12; wait12; wait12; gtsby; repeat begin

Port [Bdadd+0] :=\$60+bit+sense; Port [Bdadd+0] :=\$3f; Port [Bdadd+0] :=\$5; fetch_la(devnum); procedure pp_en; wait12; wait12; wait12; begin end;

procedure all_loc; tci (\$f7); begin :pue

procedure all_rem;

tci (\$f8); begin end;

if iors='s' then Port[Bdadd+5]:=\$87 else Port[Bdadd+5]:=\$83; out_ta(devnum);
if eois='' then dummy:=13 else dummy:=ord(eois); until not ((intlstat and \$10)=0); Port[Bdadd+0] := my_addr + \$20; intlatat :=Port [Bdadd+1]; Port [Bdadd+7] : =dummy; Port [Bdadd+4] :=\$40; Port [Bdadd+5] :=\$80; Port [Bdadd+5] :=\$0; Port [Bdadd+4] :=\$3; Port [Bdadd+5] :=\$0; if not my_flag then var dummy : integer; Port [Bdadd+0] :=\$3f; fetch_la(devnum); procedure trnsfr; tci (\$f6) ; tci (\$fd); wait12; } repeat wait12; begin end; begin end;

Port [Bdadd+5] := \$82; Port [Bdadd+4] := \$40; Port [Bdadd+5] := \$0; procedure my_la; last:"false; (wait12; begin end;

procedure my_ta;

ua(f13;	until not / (down and 21-01.
Port[Edadd+0]:= mv addr + 540;	comm:=Port[Bdadd+5]:
wait12	Port [Reladed+5] := Sf:
145t:=f5186; 145t:=f6186;	comm and 127
end;	if comm =9 then beein
• • • • • • • • • • • • • • • • • • •	Port [Bdadd+4] :=\$80;
procedure listen;	Port [Bdadd+5] :=\$0;
begin	cntrllr:=true;
fetch_la(devnum);	tci (\$fa) ;
end;	end.
	else begin
procedure taik;	hold('Error - undefined command');
begin	end <i>i</i>
out_ta(devnum);	end;
end;	end;
procedure unlisten:	procedure pass cutri:
best in the second s	
test: test:	if not cutrily then begin
wait12;	out ta (devnum) :
Port [Bdadd+0] :=\$3f;	wait12;
gtaby;	Port [Bdadd+0] :=\$9;
end;	wait12;
	Port[Bdadd+4]:=\$1;
procedure untalk;	Port [Bdadd+5] := \$a1;
begin	Port [Bdadd+5] :=\$0;
tcsy;	lastint1:=0;
waiti?	cntrllr:=false:
Port [Bd.dd+0] : =\$5f :	tci (Sf1):
qteby;	end;
end;	end;
procedure tcsy;	procedure set_eoi;
begin	begin
tci (Sfd) ;	Port [Bdadd+5] :=\$6;
	last:=true;
	end;
Port [Bdadd+] :=\$80;	
Port [Bdadd+5] : =\$0;	procedure rdsup;
	var dummy : integer;
	le un flan than an una (denum).
	if antilly that had
usian tei facili	AT VALLILE CHEM BEYAN Wait12:
Port [Bdadd+5]:=\$80;	Port (Bdadd+0) :=53f;
Port [Bdadd+5]:=\$3;	my_rap(devnum);
Port [Bdadd+4] :=\$80;	end;
Port (Bdadd+5) :=\$0;	if not (my_flag or cntrllr) then repeat
end;	dummy:=Port [Bdadd+4];
	until (dummy and 4)=0;
var dummy.comments	(DID
beain	procedure rd arry:
if not cutrilr then	var dummy, iobyt, i : integer;
begin	achar : atring[1];
if not ((lastint] and 120)=128) then repeat	begin
Intlastat:=Pert Bdadd+1/2 	Port (Bdadd+5) :=\$81;
	rosuptaevnum; ; vanait
repear	r epear

repert repert for the school of the school with the school of the school	<pre>repeat repeat dummy:=Port[Bdadd+1]; until not ((dummy and 1)-0); iobyt:=Port[Bdadd+0]; achar:=chr(iobyt); achar:=chr(iobyt); sar[i]:=sar[i]*dear); if (achar=sep) or (achar=lastchar); until (achar=sep) or (achar=lastchar); if (achar=lastchar);</pre>	<pre>ccedure zero_add; ir i : integer; gin ir i=1 to 10 do ir i=0; sec(1):=0; sec(2):=0; s</pre>	id;
---	---	--	---------

.

.

.