PC3000

EUROTHERM CONTROLS

Technical summary

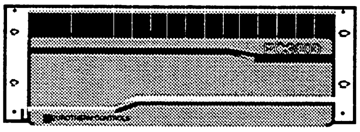
EUROTHERM CONTROLS

System 3000
Technical Summary

CONTENTS

EUROTHERM System 3000

A **versatile**, highly **integrated**, **modular** control system configured using **high productivity** programming tools and conforming to the International Standard IEC 1131/3.



The PC3000 incorporates

Precision Analogue Control

- Direct sensor inputs with over 40 linearisation tables and 14 bit resolution.
- Highly configurable PID control.
- Self tune and the latest technology Least Squares Adaptive tuning.

Digital Control

- Boolean and combinational logic functions.

Sequential Control

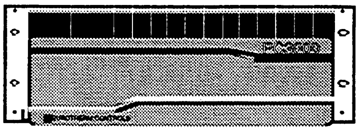
- Using Sequential Function Charts (sometimes referred to as Grafcet) and described in the International Standard, IEC 1131/3.

Statistical Control

- In-built calculation of SPC variables, standard deviation, mean, upper and lower control limits etc.
- Ability to use this information for real time Statistical Quality Control (SQC).

Communications

- Two User defined ports plus one programming port as standard.
- Expandable to 16 x RS485 plus 2 x RS422 plus 5 x RS232.
- Many protocols
    - Eurotherm EI BiSync
    - Modbus/ Siemens 3964R/ Allen Bradley/ Toshiba_EX
    - Simple DIY driver for printers, gauges etc.



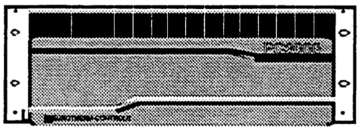
PC3000 data manipulation includes

Data types

- Reals (floating point), Inegers and Booleans
- Time and Time of Day
- Date and Date & Time
- Strings

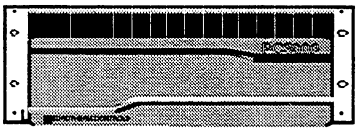
Operators and functions

- Most mathematical operators
- Full floating point arithmetic and mathematical functions
- Boolean operators and functions
- Date, Time and Time-of-Day arithmetic



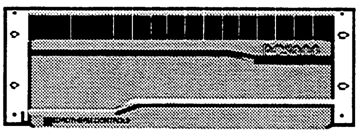
PC3000 Operator Interfaces include

- Simple two line panel, through hole or surface mounted.
- Monochrome and colour terminals
- Supervisory systems.



The PC3000 is easy to apply

- All of the above functionality of PC3000 may be configured to provide he solution to your application. The application is created using "Function Blocks" and software "Wiring" in conjunction with Sequential Function Charts" on an easy-to-use, PC based, Programming Station.



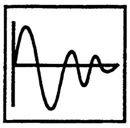
PC3000 has already been applied in a wide range of industries:

- Integrated extrusion control - pipe, cable, fibre, film profile and compounders
- Heat treatment furnaces - integrated furnace control, vacuum furnace, heat presses, tunnel kilns
- Biochemical and Pharmaceutical - Fermentation, freeze drying,
- Semi-conductor - Diffusion furnaces, MBE
- Food - Baking ovens

and many more

Key Features

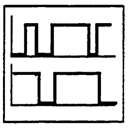
Analogue I/O



A range of Analogue Input modules to cover a wide spectrum of applications. Designed with high accuracy, high stability and high resolution (15 bit) for thermocouples (integral CJC for every channel, linearisation etc.), resistance thermometers and most types of transducer for measuring pressure, speed, position etc.(0-10V, 4-20mA etc). Modules are multi-channel with interchannel isolation and all channels isolated from the rest of the system. There is a special module with a high impedance channel (>100Mohms) for Zirconia probes, and a module with 2 Frequency-to-Voltage channels (max 10kHz).

Analogue Output modules provide high stability, high resolution (12 bit) analogue signals (0-10V, -10 to +10V, 4-20mA etc) for driving actuators, drives, thyristor stacks etc. Modules are multi-channel with all channels isolated from the rest of the system.

Digital I/O

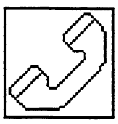


A range of Digital Input modules for low voltage and high voltage, ac and dc applications as well as modules with integral power supply for volt free contacts. Modules are multi-channel with all channels isolated from the rest of the system but not from each other. Channels on different modules are isolated from each other.

A range of Digital Output modules to provide logic level signals and relay contact switching at low and high volts and ac or dc. The relay modules have 12 normally open contacts arranged in 3 sets of four sharing a common. Logic modules offer npn open collector outputs, also arranged as 3 sets of 4 sharing a common. The logic output module does NOT provide the ouput power - it is an open collector switch only and the external devices, eg thyristor units, will equire an external 12 volt supply. Logic Output channels are isolated from the rest of the system but not from each other in its group of 4. Relay Output channels are inherently isolated from the system and from the other groups of 4.

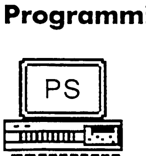
A pulse input module is available with 2 channels, each with 2 quadrature inputs. These inputs are capable of counting up to 200kHz. Local ouputs on the module can be set when a count value is reached.

Communications



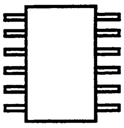
PC3000 comes with 3 communications ports as standard and a range of supporting system software for various protocols. Further ports can be added, as I/O modules, to provide a maximum of 23 ports (a combination of RS232 and RS485). These can be used as masters to communicate to PLCs, discrete controllers, motor drives, intelligent transducers, gauging equipment, weighing machines, printers and the like. They can also be configured as slaves for communication with host computers, supervisory systems etc. This very powerful communications capability allows PC3000 to act as a data acquisition unit, data concentrator and communications gateway in a complex network.

Programming



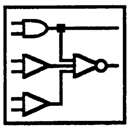
PC3000 programs are written off-line using Programming Software that will run on a personal computer under DOS. This may be run as a DOS application under Windows or OS/2. The software is an integrated set of editors providing the programmer with a highly interactive set of tools for creating and editing his program, adding comments, program file management, compilation, download and on-line, real-time debug. Hard copy of the program can be output to a printer together with cross reference listings and graphical output of the Sequential Function Charts. Context sensitive help screens are provided to assist with program generation.

Function Blocks



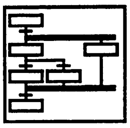
Pre-packaged functionality is provided in software blocks called Function Blocks, as described in the international specification IEC1131/3 These objects can be thought of as software 'chips' or more accurately as 'soft instruments'. The library of standard Function Blocks provided with the PC3000 include three term controllers, timers, counters, shift registers, a basic set of communication protocols, I/O support.and many more. Additionally more specialist Function Blocks are included such as further communications protocols, Statistical Process Control, Recipe Management and others.

Soft Wiring



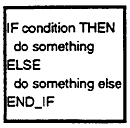
Function blocks may be linked together with language statements in the same way that discrete instruments or chips can be wired together to construct a control system. This 'Soft Wiring' need not be simply point to point but can include arithmetic, functions, decisions and other language constructs.

Sequence Programs



Whilst the Function Blocks provide the mechanism for implementing a wide range of continuous control functions, a deterministic sequencing engine and a graphical editor provide the method of describing time and event dependent actions. The language conforms to the **Sequential Function Chart** language described in IEC1131/3. A powerful benefit of this approach is that, instead of executing the entire program at every scan, only that small part of the program relevant to the current machine state is executed. The language includes mechanisms for describing alternative paths and parallel execution.

Powerful Programming Language

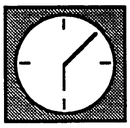


A high level, textual language is the common thread running through all of the PC3000 programming mechanisms. It provides the means of linking together Function Blocks (soft wiring) in exactly the same way one would wire together discrete instruments. It is used to describe the actions to be taken at each step of the sequence program and also the conditions under which the sequence program moves on to the next step. The language is **Structured Text** and is as specified in IEC1131/3 standard.

Functions

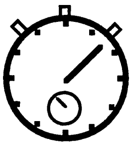
A wide variety of mathematical operators and functions are supplied to support the Structured Text language. These can be used in any Structured Text statement in either soft wiring or sequence program. Operators include the normal arithmetic operators as well as boolean and comparison operators. Functions include mathematical functions (SIN, COS, LOG, MAX etc), type conversion, time arithmetic, string handling (for communications and Operator Interfaces).

Real Time Clock



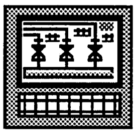
An integral Real Time Clock, completely accessible from the program provides a powerful mechanism for programming real time actions, preventative maintenance procedures, time stamping events and alarms. Functions are provided for real time arithmetic as part of **Structured Text**.

Deterministic Performance



The Operating System of the PC3000 supplies the Function Blocks and the Sequencing Engine with heart beat 'ticks' which determine the instant of execution of the various elements of the user program. These are defined as Tasks in the IEC1131/3 standard. This means that program execution time and, more importantly, I/O response is completely determined irrespective of program size and complexity. This is completely unlike the normal PLC mode of execution the rate of which is determined by I/O count and program size.

Operator Panels

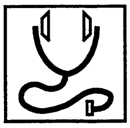


The Europanel2, a 40 character x 2 line display, is available and is fully supported by PC3000 Function Blocks. The communications protocols offered in PC3000 make a wide range of proprietory industrial operator staions available to the user, for example the Xycom range of 9" monochrome and 12" colour CRTs. They provide flexible and programmable user interfaces at the process level. The PC3000 can support multiple Operator Stations thereby enabling the presentation of information to be matched to the location and level of the operator.

Supervisory Software

EUROTHERM CONTROLS offers the WIZCON supervisory software to run under DOS and WIZCON for Windows to run under Windows®95/NT based personal computers. These system provide a graphical user interface with recipe management, alarm handling, data logging, chart plotting and reporting. One such system can communicate, not only with many PC3000s, but also with other EUROTHERM CONTROLS equipment, such as the 900 range of discretes and EUROTHERM DRIVES AC and DC motor speed controllers as well as third party equipment such as PLCs.PC3000 has also been used with many other popular SCADA packages such as FIXMACS™ from Intelution and INTOUCH™.from Wonderware. The protocol usually used is MODBUS.

Diagnostics

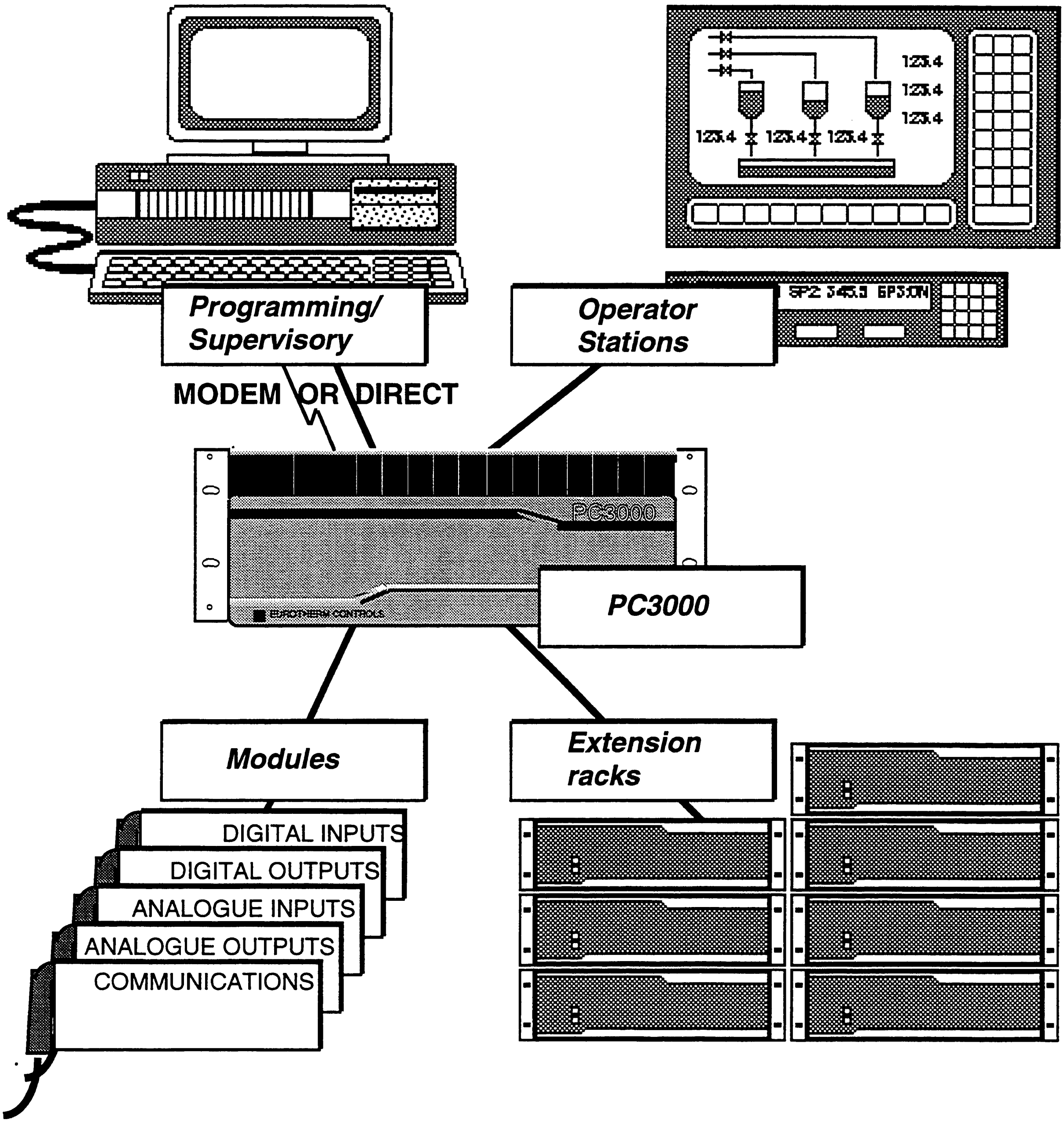


All modules have LED indication of health, communication activity and I/O state. Function Blocks all have status information which is available to the programmer and which can be used to provide fault tolerant programs. This status information can provide very specific information about hardware condition, program errors, maths errors, communication faults and program execution.

For program testing on the plant, every I/O point can be separately disconnected, in software, from the real world so that input values can be simulated and output response viewed without affecting the plant or the plant wiring at all.

The Programming Software can be used on-line to provide read/write access to all of the parameters in the program. In addition, the Sequential Function Chart is highlighted to show the currently active step providing, therefore, a real time 'mimic'.

The PS includes MODEM support for PC3000 systems in remote locations. All the PS features are available to the remote user via the modem link, just as if the PC3000 was directly connected. The modems must be Hayes compatible and should be capable of 9600 baud.

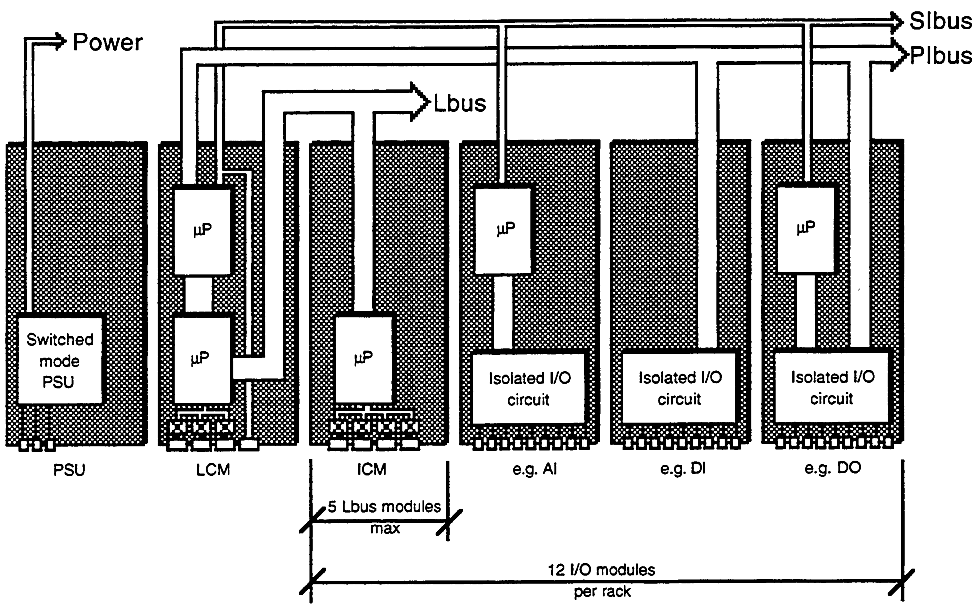


The System 3000 family of products includes hardware modules for low level I/O. Power supplies, communications and processor modules. Racks. Operator Stations. High level programming software operator panels and supervisory packages.

The following pages describe all of these products in some detail. For more information on any particular item and for application notes relevant to a particular industry, consult Eurotherm Controls.

Hardware

Description



I/O Modules

Data acquisition and control at the plant level is carried out through I/O modules which provide low level interfaces to transducers and actuators. Up to 12 I/O modules can be accommodated in a single rack and, in the maximum possible system of 8 racks, 96 modules in all. Although the number varies from module to module, all I/O modules have several channels, every one of which is isolated from the system. Some modules (e.g. Analogue Input) have inter-channel isolation as well.

Some modules have a micro-processor on board to handle the I/O task. An example is the Analogue Input for which the processor controls the dual slope A to D and provides calibration, scaling, limits checking, linearisation and communication.

Some modules have no processing capability, an example being the Digital Input range of modules.

Other modules can be used for either Digital functions or Analogue functions. An example here is the three channel Relay Output module which can be used as a simple digital switch or can be used, in Time Proportioning mode, as an Analogue control output. This module has a processor on board which provides the time proportioning functionality if required. The mode in which an output channel will operate is completely software configured, no hardware changes being required.

I/O Communication

The I/O modules plug into a rack, the back plane of which carries the power rails and the communication bus termed IObus. IObus has two constituents; a serial bus (SIbus) and a parallel bus (PIbus). PIbus is an 8 bit wide, 1Mz bus communicating with those modules that are Digital Inputs or Outputs or are configured as such. SIbus is a 375KHz bus communicating with those modules that are Analogue Inputs or Outputs or are configured as such.

All of the low level tasks associated with reading input channels and writing to output channels as well as determining module health etc. are handled by a dedicated micro-processor , termed the I/O Controller or IOC, which resides in the Local Controller Module (see next paragraph).

Local Controller Module (Replacement only) and LCM-Plus

The Local Controller Module (LCM) is a 68000 based single board computer forming the 'heart' of the PC3000. It has an EPROM module used for the system firmware with the current Function Block Library, a Floating Point co-processor , a Real Time Clock and 256Kbytes of battery backed CMOS RAM for the application program. This can be extended by the addition of a 128k or a 512k RAM module.

It transmits and receives data to and from the I/O modules through the IOC (see previous paragraph). It is equipped with two RS422 and one RS485 communications ports through which it can transmit and receive data to and from remote equipment such as slave instruments, intelligent transducers, weighing machines and supervisory systems. One port must be made available for the PS programming station, the other two may be software configured to the desired protocols.

The LCM contains a multi-tasking operating system which controls the execution of Function Blocks, Sequence Program, external communication and internal housekeeping activities.

The LCM-Plus is an enhanced LCM based on the 68020 processor with integral co-processor which offers enhanced performance particularly for mathematically intensive applications, eg multiloop Auto- tune PID, SPC, etc. The LCM-Plus has 512k of base RAM and the three ports may be individually set up as RS422 or RS485, and two of them as RS232. The RAM can be extended by the addition of a 128k or a 512k RAM module.

Both LCMs have a 128k RAM slot available for DOWNLOADABLE FUNCTION BLOCKS. These are specialist, custom or new function blocks over and above the standard library. These are only available from Eurotherm Controls, Worthing. In this slot the LCM-Plus can use a 512k RAM card, 128k for downloadables, and 384k extension for the user program.

Unused or spare RAM can also be used as a File System, storing data or text files.

A buffered version of the 68000/68020 data/ address bus, termed Lbus, is available in the first five slots of the main PC3000 rack backplane for the Intelligent Communications Module (ICM).

Communications Module

The Intelligent Communications Module, ICM, plugs into one of the Lbus slots in the rack and provides four additional communications ports, three RS485/RS422 and one RS232. The module is 68000 based so that all of the low level communications handling can be off-loaded from the LCM. The ports are software configured to the desired protocol(s).

Extension Racks

The serial port (SIbus) of the IObus is brought to the front panel of the LCM where it can be used for multi-drop communication with extension I/O racks. In an extension rack, the LCM slot is taken by a Rack Interface Module (RIM) which locally generates a complete IObus. All I/O module types, excepting those using the Lbus, can thus be accommodated in this rack. Up to seven such racks can be multi-dropped from the LCM providing, therefore, a maximum module count of 96. The I/O modules in the extension rack are scanned at exactly the same rate as the main rack and the extension racks can be located so that there is no more than 200m from the main rack to the final extension rack.

Specifications

Environmental

Operating Temperature	0	→50	°C
Storage Temperature	-10	→70	°C
Relative Humidity	5	→95	% non-condensing

Power Supply Module (PSU)

Voltage range	85	→132	Vac (no link fitted)
	170	→264	Vac (with link fitted)
Frequency range	48	→62	Hz
Rating		120	VA
System requirement		1	per rack
Auxilary supply (option)	24V	300mA	for Europanel2

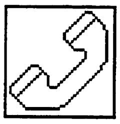
Local Control Module (LCM)

RAM	256k base RAM	slot 1	128k or 512k extension
		slot 2	128k for special FB
Communications ports	1 port for PS	2	user ports
		1	extension rack port
Standard	Cannot be multidrop slave	RS 422	(ports A & B)
	May be multidrop slave	RS 485	(port C)
Baud rates	300	→38.4 k	(ports A & B)
	300	→115k	(port C)
		375k	(ext rack)
Protocols	Software selectable		(2 of ports A, B & C)
System requirement		1	per system

Enhanced Local Control Module (LCM-Plus)

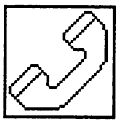
RAM	512k base RAM	slot 1	128k or 512k extension
		slot 2	128k for special FB
		or slot 2	512k=384k ext, 128k FBs.
Communications ports	1 port for PS	2	user ports
		1	extension rack port
Standard	RS422, RS485 or RS232		(ports A & B)
	RS422, RS485		(port C)
Baud rates	300	→38.4 k	(ports A & B)
	300	→115k	(port C)
		375k	(ext rack)
Protocols	Software selectable		(2 of ports A, B & C)
System requirement		1	per system

Intelligent Communications Module (ICM)



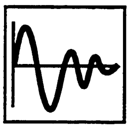
Communications ports		4	
Standard	3 off	RS485	(ports A, B & C)
	1 off	RS232	(port D)
Baud rates	300	→38.4 k	(ports A, B & C)
	300	→19.2K	(port D)
Protocols	Software	selectable	(all ports)
System capability		up to 5	(Slots 1 to 5, main rack)

Rack Interface Module (RIM)



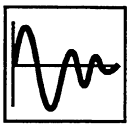
Inter rack ports	2	*Used only for inter-rack connections.*
Standard	RS 422	
Baud rate	375k	
System requirement	1	per extension rack up to 7

Analogue In (AI4) Standard Modules



No of channels		4	
Range	(mV4, mA4)	0-100mV(max), 0-20mA, Thermocouple	*three different modules provide for these requirements*
	(RT4)	PT100 Resistance Thermometer	
	(V4)	0-10V(max)	
Isolation		Channel to channel and channel to system	
Calibrated accuracy		± 0.25	%
Temperature coefficient (gain error)		typ < 5	ppm / °C
Resolution		14 bits	15 effective
Linearisation tables		>40	incl J, K, R, S, B, Pyrometer, Sq root
CJC rejection		>30:1	
Input impedance		>1MΩ	mV ranges
		>100kΩ	V ranges
External Burden resistor		5Ω	mA ranges
System capability		up to 96	total I/O modules

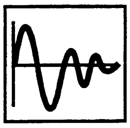
Analogue In (AI4) Special Modules



Special module - mV4 with one high impedance channel for Zirconia probes.		
No of channels	4	3 as mV4 module
Hi Z Range	-1V to 1V, 2V, 5V or 10V	
Isolation	Channel to channel and channel to system	
Input Impedance	> 100	Mohm
Temperature coefficient (gain error)	typ 25	ppm / °C
	Resolution	14 bits 15 effective
System capability	up to 96	total I/O modules

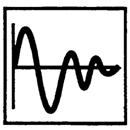
Special module - V4 with 2 channels 0 to 10kHz frequency to voiltage inputs.		
No of channels	4	2 as V4 module
Frequency	10Hz to 1, 2, 5, 10kHz	
Isolation	Channel to channel and channel to system	
Input Impedance	> 100	kohm
Miinimum input signal	100	mV
Maximum input signal	70	V rms
Resolution	14 bits	15 effective
System capability	up to 96	total I/O modules

Analogue In (AI2)



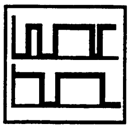
No of channels	2	fully configurable.
Range (V2, mA2)	TC, RT, mV, V, mA	
Isolation	Channel to channel and channel to system	
Calibrated accuracy	± 0.25	%
Temperature coefficient (gain error)	typically < 40	ppm / °C
Resolution	14 bits	15 effective
Linearisation tables	>40	incl J, K, R, S, B, Pyrometer, Sq root
CJC rejection	>30:1	
Input impedance	>1MΩ	mV ranges
	>100kΩ	V ranges
Burden resistor	5Ω	mA ranges
System capability	up to 96	total I/O modules

Analogue Out (AO4)



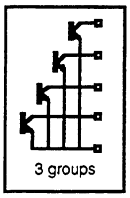
No of channels	4	each o/p may be mA or V
Range (V4	0-5V, 0-10V, -10 to +10V etc	
(mA4)	4-20mA, 0 -20mA etc.	
Isolation	Channel to system	
Calibrated accuracy	± 0.25	%
Resolution	12 bits	
Burden	<600Ω	mA ranges
Protection	Output current limited	
System capability	up to 96	total I/O modules

Digital In (DI14)



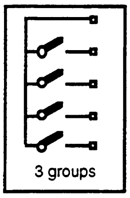
Range	No of Channels	
Digital Input Low (5LL14)	5Vdc	14
Digital Input Low (24LL14)	24Vdc	14
Digital Input Low (ACLL14)	24Vac	14
Digital Input High (HL14)	85-264 Vac	14
Digital Input Contact (CC14)	Volt free	14
Hysteresis	Set at 65%, reset at 35%	
Isolation	Channel to system	
System capability	up to 96	total I/O modules

Digital Out - Open Collector Logic (LGC12)



No. of channels	12	External PSU required!
Configuration	3 groups of 4 npn open collector pull down	
Range (LGC12)	up to 30V, 100mA max	
Isolation	Group to group, groups to system	
System capability	up to 96	total I/O modules

Digital Out - N/O Relay (RLY12)



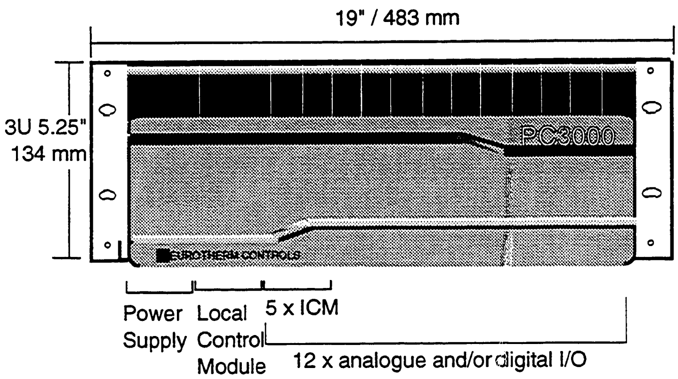
No. of channels	12	3 groups of 4
Configuration	normally open contacts	
Range (RLYSTD12)	264 Vac (30 Vdc)/1A	
Isolation	Group to group and groups to system	
System capability	up to 96	total I/O modules

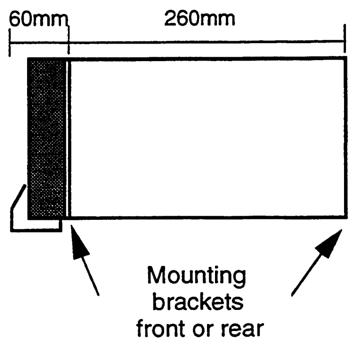
Pulse Input (5Q2)

No. of channels		2	2 quadrature inputs each.
Encoder Supply		5V	Internal jumper for 12V
Frequency			200kHz max
Max count		24bits	
Inputs	Phase A & B, reset, enable	Logic	5V or 12 V as above
Outputs	Compare	Logic	Open collector
Isolation	Both channels to system		
System capacity		up to 96	total I/O modules

Rack

Size	19" x 3U (5.25") x 320mm		
Capacity	Power supply module	1	
	Local Controller Module	1	in main rack
	OR Rack Interface Module	1	in extension racks
	I/O modules	up to 12	
	Communications modules	<= 5	in main rack only uses I/O slot
System connections	via back plane DIN 41612 connectors		
Field wiring	Front of rack on removable headers		
Rack spacing	for wiring and cooling	>=3U	
Mounting	Bulkhead (rear) or front panel		
System capability		up to 8	1 main rack plus up to 7 extension racks





Software

IEC1131/3

Since 1979 an international working group of users and manufacturers from the plc and control industry has been developing a new standard for programmable controller languages. The first draft was produced in 1982 and it was decided that the standard needed to be broken up into 5 sections:

Part 1	General Information
Part 2	Equipment and Test requirements
Part 3	Programming languages
Part 4	User Guidelines
Part 5	Messaging Services.

Part 3, Programming Languages, was published as a standard by the IEC in 1992 under the chairmanship of Dr James H Cristensen of the Allen Bradley Company, Inc. This standard is referred to as IEC 1131/3.

Generally an application program must handle the following:

-**Interlocks** that control the conditions under which certain activities or processes can operate; eg a steam valve to a heat exchanger may not open unless there is proof of flow of the heat transfer fluid.

-**Alarms** that are triggered when certain boundary conditions are exceeded; eg when the temperature of a process vessel exceeds normal working temperatures.

-**Closed Loop control** that maintains processes parameters at their optimal condition; eg a furnace temperature kept within an acceptable range during loading, heat up and soaking.

-**Sequencing** that initiates and terminates key phases of a batch process through well defined states; eg idle, standby, pre-inoculation, cultivation, downstream processing of a fermentation system.

-**Optimisation** that views a total process and manipulates process settings through the entire system in order to optimise process yield; eg in extrusion using data from gauge(s) to maintain diameter and wall thickness by regulating extruder speed, mass flow and haul off speeds. Also the use of on-line statistical process control.

The traditional PLC uses ladder logic to achieve all of these functions in a single rung which is scanned at a fixed rate. Not all the functions outlined above are best addressed by the use of ladder logic.

The IEC 1131/3 offers a range of programmable controller languages so that the different parts of an application can be programmed in the most appropriate language and yet they can all be linked into a single executable program. To improve performance the Standard also introduces the concept of **Tasks** or managers to which the various parts of the application can be allocated. The IEC1131/3 standard also incorporates modern software engineering principles including structured, hierarchical programming and abstract data types.

One of the objectives is the make the application source code intelligible to process engineers themselves rather than just to the software implementers.

The suite of languages covered by the IEC1131/3 standard include related textual and graphical programming languages:

-Instruction List "IL". A low level language that is analogous to a microprocessor assembler in which only one operation, such as storing a value in memory, is allowed per line.

-Structured Text "ST". A high level language which can be used to express complex analogue and digital statements using English words such as IF, THEN, ELSE, SQRT() etc. Analogous to "C" code.

-Sequential Function Charts "SFC". A graphical representation of sequences comprising STEPS which contain actions and TRANSITIONS which contain tests. This is based on Grafcet and supports both alternative sequence selection and parallel sequences.

-Ladder Diagram "LD". A graphical language that uses the standardised set of ladder programming symbols.

-Function Block Diagram "FBD". A graphical language which allows program elements (eg soft instruments like PID) to be "wired" together in a form analogous to a circuit diagram.

Designed to be applied where the majority of control is analogue and then sequencing is associated with batching, **System 3000** includes the following IEC languages:

TASKS

STRUCTURED TEXT

FUNCTION BLOCK DIAGRAMS

SEQUENTIAL FUNCTION CHARTS

PC3000 Programming Software (PS)

PC3000 is programmed, off-line, using the PC3000 Programming Software tool kit which runs on a Personal Computer under DOS . A minimum of 2Mbyte of RAM is required, (4Mb for the LCM-Plus). The PS will run as a DOS application under Windows and OS/2. The software is a set of interactive editors and management tools providing facilities for

- Hardware declaration and configuration
- Function block declaration and configuration
- Soft wiring and combinational logic.
- Sequence program generation
- Compile and build
- Download and upload
- Subsidiary file creation including communications tables.
- Program file management
- On line debug
- User defined screens to assist on line facilities.

It is supplied with all disks and a complete A5 set of Reference Manuals. Training is available, contact your local Eurotherm Controls office.

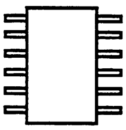
Hardware Management

The Hardware Management editor provides the programmer with the functions necessary for declaring, naming and configuring the hardware modules that will be present in a particular application. Note that this does not require a PC3000 to be present since all programming is performed off-line.

Once declared and named, using meaningful names of up to eight characters, the hardware I/O channels are represented as Function Blocks and are handled in precisely the same way as any other Function Block. Configuration is simply a matter of setting parameter values in the associated Function Blocks.

The Hardware Management editor provides facilities for moving previously declared modules to different positions in the rack and for moving channels from one module to another, even between modules of a different type provided they are functionally equivalent e.g. low voltage digital input and high voltage digital input.

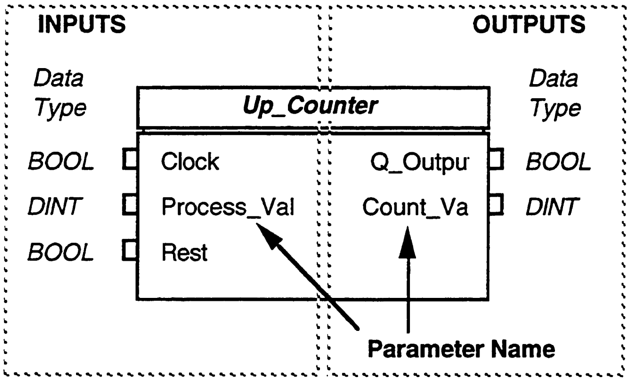
Function Blocks



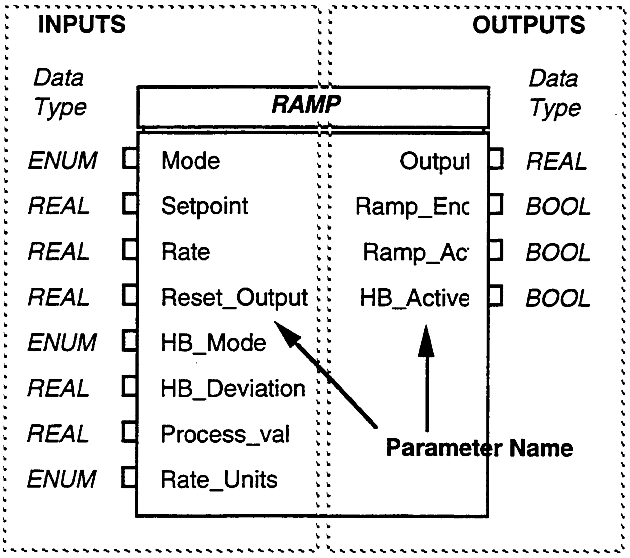
The PC3000 Function Block library contains a set of high level 'Software Instruments' which are analogous to the discrete instruments that one may purchase and wire together e.g. PID Controller, Timer, Counter etc. Unlike other systems, the number of Function Blocks is not limited and the programmer can declare as many of each type as is required to implement the control system. Practical limits are, of course, imposed by memory constraints and execution time which are described in a later section. The Function Block concept is described in IEC 1131/3 and the PC3000 conforms closely to this standard.

All Function Blocks have **Inputs** which may be written to from the Programming Software, from an Operator Station, from a Supervisory system or from the PC3000 itself by way of 'soft wiring' or the sequence program. They also have **Outputs** which are calculated by the function block itself. Many Function Blocks have **Configuration Parameters** which are set on start up and never change. These parameters are used to specify the exact way in which the Function Block will perform. An example might be the Linearisation of an Analogue Input since it would be very unusual to change the transducer type at run time. Most of these parameters are, however, treated by the PC3000 in the same way as Input parameters and can, therefore, be changed at run time or by the program itself.

A simple example of a function block is the **Up_Counter** and a slightly more complex example is the Ramp. These are shown diagrammatically below.



The Up_Counter is a simple uni-directional counter. **Count_Val** will increment every time **Clock** changes from 0 to 1 . When it reaches the value of **Process_Val**, **Q_Output** becomes true which can then be used to trigger an action. The **Reset** input sets **Q_Output** false and **Count_Val** to zero.



A Ramp Function Block, often used for setpoint programming of control loops, will ramp the **Output** towards the **Setpoint** at a rate determined by the **Rate** input parameter. This **Output** can then be connected, for instance, to the **Setpoint** parameter of a PID function block. The Ramp also supports a **Hold-back** mode whereby the **Output** can be forced to track the **Process_Val** within a given deviation. This might be used in the situation where the rate of change of the plant might not keep up with the programmed ramp rate. Further outputs from the function block indicate the operating conditions.

For ease of programming the Function Block Library is catagorised in **CLASS** groups in which a number of similar **Types** of function blocks are listed. The library is constantly being extended especially in the area of function blocks for specialist applications. This is a representative list of the current Library.

The initial CLASSES listed here are the common blocks whch are most likely to be used in every application.

CLASS	Type	

Used to control the functions of the LCM.

SYSTEM	PcsSTATE	LCM, Start up strategy, alarms etc.
	Task	No of tasks, task intervals, priorities.
	Messages	PS debug messages
	RT_Clock	LCM Real Time Clock

Associated with the HARDWARE inputs to the system.

INPUTS	Digital_In	PV, Status, module status, Input forcing.
	Debounce_In	DI with SW time before i/p change of state is recognised.
	Analog_In	PV, Status, Module status, Inpup forcing.
	Analog_In_R	AI for PT100 etc.

Associated with the HARDWARE outputs from the system.

OUTPUTS	Digital_Out	Relay, logic; module status, Output forcing.
	Analog_Out	% op, status, output forcing.
	T_Prop_Out	%op for time proportioning relay or logic. Includes fan and water cooling algorithms.

A choice of Control Algorithms.

CONTROL	PID	Standard PID Control
	VP	Standard boundless Valve Positioning Control.
	PID_Auto	PID with Auto-tune, DRA, & LSAT adaptive tuning.
	PIDHeatCool	As auto-tune but with limited parameters visible.
	VP_Auto	VP control with AT, DRA and LSAT.

Internal Variables showing the different DATA types recognised.

USER_VAR	Boolean	0 (*Off*), 1 (*On*).
	Real	32 bit, floating point (up to $\pm 3.4 * 10^{38}$)
	Integer	±2,147,483,647
	Time	00d00h00m00s000ms to 23d23h59m59s
	Time_Of_Day	eg 13:57:15
	Date	eg 01-Jan-2000
	DateAndTime	eg 31-Dec-1999-23:59:59
	String	up to 80 characters.
	Long_String	up to 255 characters.

Input and Output Conditioning Function Blocks.(Selection only)

CONDITION	Scale	2 point straight line scaling.
	MaxMinAverg	Maximum, minimum, average of 16 inputs
	SwitchOver	Weighted scaling between 2 overlapping sensors eg thermocouple to pyrometer.
	Process_Dly	Real parameter value delayed by a set time
	RelHumidity	Gives RH from wet and dry PT100 and atmospheric pressure.
	Rate_Limit	Output will only slew at the set rate.
	Freq_Cnt	Computes speed from a pulse interval (>10mS)
	Totaliser	Totalises a real number.
	CustomLin	18 point straight line custom linearisation.

Used together these blocks provide detection, acknowledgement and storage of Alarms and their History.

ALARMS	Sensor	Time stamps alarms.
	Cur_Alms	Provides acknowledge for alarms.
	History	Logs the last 128 cleared alarms.
	Sensor_16	As Sensor, but for 16 points.
	Detector	Provides flags for Hi, Lo, DevHi, DevLo, DevBand.

Ramp and Ramp + Dwell generators based on time to target or rate to target.

PROGRAMMER	Ramp_Rate	Ramp to target specified by ramp rate.
	Ramp_Target	Ramp to target specified by time to target.
	Prog8Rate	8 off ramps (rate) to target + dwell
	Prog8Time	8 off ramps (time) to target + dwell

These CLASSES of Function Blocks are all associated with COMMUNICATIONS.

The Comms Drivers allocated to LCM or ICM ports.

COMMS	EI_Bisync_M	Eurotherm ASCII Master
	EI_Bisync_S	Eurotherm ASCII Slave
	Raw_Comms	A DIY driver for simple devices eg barcode.
	Siemens_M_S	Siemens 3964(R) Master & Slave.
	JBus_M	JBus, or Modbus, RTU or ASCII, Master.
	JBus_S	JBus, or Modbus, RTU or ASCII, Slave
	Toshiba_M	Toshiba EX250, EX500 & EX2000, Master.
	Euro_Panel2	Europanel 2x40 character display
	AllenB_M	Allen Bradley Master for PLC2-5. (LCM ports only).

Remote Variables with a Master Comms driver address a remote device as a slave. Modes can be Read Continuous, Write Continuous, Write on Change, or triggered On Demand.

Note a single Function Block can contain 8 or 64 variables with contiguous addresses.

REMOTE_VARS	Rmt_Bool	Individual Boolean flag
	Rmt_Real	Individual Real number
	Rmt_Dint	Individual (double) integer
	Rmt_Time	Individual time variable
	Rmt_Str	Individual string
	Rmt_Bool_8 (and _64)	Block of 8 (or 64) Booleans
	Rmt_Real_8 (and _64)	Block of 8 (or 64) Reals
	Rmt_Dint_8 (and _64)	Block of 8 (or 64) Integers
	Rmt_SW	Status Word - a single integer displayed bit by bit.

Slave variables with a Slave Comms driver being addressed by a remote master.

Note a single Function Block can contain 8 or 64 variables with contiguous addresses.

SLAVE_VARS	Slv_Bool	Individual Boolean flag
	Slv_Real	Individual Real number
	Slv_Dint	Individual (double) integer
	Slv_Time	Individual time variable
	Slv_Str	Individual string
	Slv_Bool_8 (and _64)	Block of 8 (or 64) Booleans
	Slv_Real_8 (and _64)	Block of 8 (or 64) Reals
	Slv_Dint_8 (and _64)	Block of 8 (or 64) Integers
	Slv_SW	Status Word - a single integer displayed bit by bit

These are blocks which allow immediate connection to Eurotherm Controls and Eurotherm Drives Products

REMOTE_INST	GenRmtInst	An instant link to 94, 808, 902, 900EPC controller
	GenRmtDrive	An instant link to one PNO block
	RmtDrive584	An instant link to a 584 inverter.
	RmtDrive590	An instant link to a 590 drive.
	RmtTU1400	An instant link to a TU series thyristor stack.

This section of Function Blocks are for the more complex applications.

Allows local recipe storage using the LCM RAM inside an application program.

RECIPE	RecipeMan	Manages a no. of variable blocks as recipes.
	StageMan	Manages a no. of RECIPES to provide a sequence.
	Bool_16x128	128 recipes of 16 Booleans
	Real_16x128	128 recipes of 16 Reals.
	Dint_16x128	128 recipes of 16 (double) integers.
	Str_1x128	128 recipes of 1 string.

These blocks allow spare LCM RAM to be used as a FILE store. This store may be formatted outside the application program area so that data is not lost on loading a new application.

FILESYSTEM	*FSStrAcs*	*File search by initial string per line.*
	FSStrAcs	*File search by initial number per line.*
	FSFormat	Used to format the RAM file store.
	FSFileHndl	File delete, copy etc.
	FSFreeSpce	Checks size of available file storage.
	FSAccess	Used to read and write a file.
	FSDirectory	Used to "dir" the file store.

These are simple single element arrays of 16 variables.

SELECT	Select_Real	Real extracted from array by index number.
	Select_Dint	
	Select_Time	
	Select_Bool	
	Select_Str	
	Set_Real	Real placed in array by index number.
	Set_Dint	
	Set_Time	
	Set_Bool	
	Set_Str	
	Regist_Real	Composite of Select and Set with index In and Out.
	Regist_Dint	
	Regist_Time	
	Regist_Bool	
	Regist_Str	

These are comphresive SPC blocks capable of producing all the standard diagnostic parameters.

STATISTICAL	SPC	SPC on a single Real variable, sampled by time.
	Histogram	The 24 segment histogram of a Real variable.
	SPC_Event	SPC on a single Real variable, sampled by Event.
	Statistics	Error, Max, Min, Std Dev of a single Real variable over a sample period.

These are for dummy loads which allow simulation.

FILTER	Lag	A first order low pass filter.
LOADS	Pid_Load	A configurable dynamic load for simulation.
	VP_Load	As above, for valve positioner outputs.

Finally a number of general purpose function blocks.

The Sequential Function Charts are made up of Function Blocks. Each step has a time associated with it showing how long it has been executing. This most often used instead of creating a specific timer. It is also indicates which steps have executed which is useful during acceptance testing.

STEPS	Macro	Shows time and whether the macro is executing or finished.
	Steps	Shows time and whether the step is executing
TIMERS	Pulse_Timer	Fixed duration pulse on receipt of rising edge.
	On_Delay	Fixed delay to set output after receipt of rising edge.
	Off_Delay	Fixed delay to reset output after recept of falling edge.
	Stopwatch	Input and reset as conventional stopwatch.
COUNTERS	Up_Counter	
	Dn_Counter	
	Up_Dn_Count	
OTHERS	Shift_16	16 bit shift register.
	Shift_Real	16 stage register for real values.
	Shift_Dint	16 stage register for integers.
	Alarm_Cntrl	Provides birth and death times and acknowledgement for an alarm point.
	Bistable_SD	Set dominant bistable.
	Bistable_RD	Reset dominant Bistable.

Tasks

The PC3000 has a multi-tasking environment with tasks that execute at predetermined rates. All of the function blocks, sequence program steps and transitions, communications and housekeeping activities must be assigned to one of a maximum of seven tasks allowed.

By default all Digital activities are allocated to a 10mSec Task, all Analogue activities and the Sequence program to a 100mSec Task, and Communications to either 10mSec (eg EI Bisync and Modbus) or 100mSec (eg Europanel2, Raw Comms).

With the default 10mSec Task ALL digital inputs are read, ALL the combinational logic etc is computed and ALL the digital outputs are written to WITHIN the 10mSec.

With the default 100mSec Task ALL analogue inputs are read, ALL the function blocks, functions etc are computed, the next stages of the sequence program are executed and ALL the analogue outputs are written to WITHIN the 100mSec.

This Tasking ensures deterministic performance - ie at the end of a 10 mSec tick, if 2 relays have been required to come on, they will be energised at the same instant.

As an application becomes larger there will come a time when the LCM cannot execute all the required code within the Task interval. This causes an "overrun". The occasional overrun is not critical. However continual overuns, particularly of the fastest task, requires a change to the application program.

The simplest option is just to slow the system down - eg change the Task intervals to 12 and 120mSec. However it may be required to keep the digital task at 10msec and generate a new task at, say, 500mSec and re-assign some of the less critical analogue functions to this new task.

Limits:
- Fastest Digital Task 5mSec
- Fastest Analogue Task 100mSec
- Fastest Task .... < 65msec
- Max number of Tasks 7

Each Task must be a whole number multiple (<20) of the previous Task.

Structured Text

```
IF condition THEN
 do something
ELSE
 do something else
END_IF
```

The language used to define all soft wiring and sequence program actions is **Structured Text** as defined in IEC 1131/3 This is a high level language designed for use in programmable controllers. It supports operations on all variable types :- Boolean, Integer and Real, includes string operations and operations on a variety of variable types associated with Time and Date.

All four mathematical operators, +, - , *, and / are supported both for integer and floating point arithmetic. The remainder operator, **MOD**, is also available.

Four logical operators are supported. These are **AND**, **OR**, **XOR** and **NOT**.

The set of available operators is completed by the comparison operators <>, =, >, <, >= and <=.

The conditional construct is also supported by use of the key words **IF**, **THEN**, **ELSE**, **ELSIF** and **END_IF.**

Functions

```
SIN, COS,
LOG, EXP,
MAX, MIN,
etc.
```

A large number of functions are provided to support the Structured Text operators and permit complex operations. They are grouped into classes according to function.

NUMERIC

Includes SQRT, LOG, LN, EXP, ABS, trig functions and inverse trig functions to enable complex mathematical expressions to be constructed. The floating point maths co-processor, an integral part of the LCM, provides the engine for these functions and means that execution is extremely fast.

SELECTION

Includes MAX and MIN and functions which select one of two values depending on the value of a boolean expression. These are more efficient than using the IF-THEN-ELSE construct.

STRINGS

Used for formatting messages on operator panels and includes the BASIC like operations of LEN, LEFT, RIGHT, MID, editing functions such as CONCAT(enation), INSERT, DELETE, REPLACE and justification operations.

TYPE_CONV

Provides for conversion between different types of variables e.g. TIME_TO_REAL, REAL_TO_TIME etc.

STRING_CONV

Provides for conversion of internal number representation into ASCII strings and vice versa. Used for operator panels and some communications protocols.

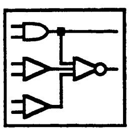
TIME_ARITH

Provides functions to add and subtract time, time of day and date to calculate, for example, the interval between two times of day or to add an interval to a time of day to arrive at another time of day. Used extensively for calculations involving the real time clock.

COMPACT

A set of functions which enable strings to be used as an efficient form of numeric arrays. Arrays containing different variable types can be accommodated.

Soft wiring



If Function Blocks are thought of as discrete instruments, the connections between them are implemented by 'soft wiring' which is directly analogous to the wiring between hardware instruments. The connections are made by language statements using the Structured Text language.

It is important to remember that statements in soft wiring are executed repeatedly at the same rate as the Function Block containing the destination parameter. It should therefore be used for continuously changing variables like a process variable and NOT for a variable which may only change once in a while - that should be handled by the Sequential Function Charts.

The soft wiring also determines the order in which Function Blocks are executed so that the destination always receives current data. Mechanisms are available to enable the programmer to construct elaborate control systems with feed-back paths and still maintain the correct execution order.

Conditional wiring may be as simple or as complex as required and enables great flexibility to be achieved. By suitable use of the conditional construct and/or **SELECT** functions (described above) dynamic re-wiring can be configured whereby a totally different control scenario may be initiated upon a set of conditions becoming true or an event occurring as a program is running.

The language editor is more than just a text editor. Language statements can be typed in or, for less experienced programmers and for large applications, a pop-up window picking mechanism is implemented which assists fault free programming. Data type checking, syntax checking and numeric over-range and under range checks are carried out dynamically. Errors are reported as soon as they are discovered which is often as soon as they are made. These features increase the speed of program generation by making sure that errors are kept to a minimum and that any errors that are made are reported quickly rather than later on after compilation.

The editor has sophisticated cut and paste, search and replace and copying facilities. Comments may be freely inserted into the language statements thus enhancing readability without sacrifice of execution speed. Readability is also enhanced by the ability to use sensible names for all I/O, function blocks and variables. These names are also not restricted to the English Language. .

There follows some examples of wiring statements using Structured Text operators and functions.

Example 1

The simple connection between an analogue input and a PID would appear as

```
LOOP.Process_Value       := INPUT.Process_Value;
```

Example 2

A slightly more complex example would involve taking the average of two inputs. The following example shows the use of brackets to force the order of evaluation.

```
LOOP.Process_Value       := (INPUT1.Process_Value + INPUT2.Process_Value) / 2.;
```

Example 3

A simple example of a digital output being true if either of two inputs is true would appear as:

```
OUTPUT.Process_Value     :=  INPUT1.Process_Value OR INPUT2.Process_Value;
```

Example 4

An alarm output (say) triggered when an analogue input exceeds a limit would be wired as follows.

```
OUTPUT.Process_Value     :=  INPUT.Process_Value > 500.;
```

Example 5

The following examples are a case of dynamic re-wiring and shows the input to a PID Function Block being either of two Analogue Inputs depending on which one has a GO status. This is an example of redundancy to protect against, for instance, open circuit thermocouples. If, for some reason, both inputs are NOGO, the PID input is forced to retain its previous value.

The first implementation conforms exactly to IEC 1131/3 and uses SELECT functions

```
LOOP1.Process_Val        := SELECT_REAL( G = LEVEL1.Status = 1(*GO*) ;
                                IN0 := SELECT_REAL( G = LEVEL2.Status = 1(*GO) ;
                                   IN0 := LOOP1.Process_Val ;
                                   IN1 := LEVEL2.Process_Val );
                                IN1 := LEVEL1.Process_Val );
```

The second implementation uses a syntax that is not described in IEC 1131/3 but is permitted on the PC3000 on the grounds that it is often more readable than the equivalent implementation using SELECT functions.

```
LOOP1.Process_Val        := IF LEVEL1.Status = 1(*GO*) THEN
                                LEVEL1.Process_Val
                             ELSE
                                IF LEVEL2.Status = 1(*GO) THEN
                                   LEVEL2.Process_Val
                                ELSE
                                   LOOP1.Process_Val
                                END_IF
                             END_IF;
```

Example 6

Functions can be used in soft wiring as in selecting the larger of two values.

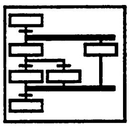
```
LOOP1.Process_Val          := MAX_REAL( IN1:= INPUT1.Process_Value,
                                        IN2:= INPUT2.Process_Value);
```

Example 7

Functions may be nested as in the following example which will put a message on the programming station message line including the current value of an analogue input.

```
Messages.PM                := CONCAT( IN1 := "Measured value of Input 1 = ",
                                        IN2 := REAL_TO_STRING( IN1:= INPUT1.Process_Value ) ) ;
```

Sequential Function Charts



Sequential Function Charts are described in IEC1131/3 as a Graphical Organisation Language rather than a true language. They provide an easily understandable way to describe time and event dependent activities and to break the control problem down into manageable portions.

The language has, basically, two constructs, a **Step** and a **Transition**.

A **step** is displayed as a box and contains **a set of instructions**, expressed in Structured Text.

A **transition** is displayed as a horizontal line and contains **a test**, again written in Structured Text. The expression can be as simple or as complicated as necessary as long as, when evaluated, it yields either the value *True* or the value *False*.

In execution, the instructions in the first step are obeyed and then the program waits until the succeeding transition becomes true, at which point the instructions in the next step are executed. Obviously, while the program is waiting, all the Function Blocks and associated wiring together with all the communications and other activities are being taken care of. The only activity associated with the sequence program is the evaluation of the expression contained in the transition, not a very large overhead.

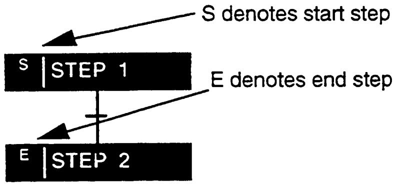
Steps can be defined as Single Shot which execute only once when active or as Continuous which execute continuously until the following transition becomes true. A step can also be defined as a **Macro** which, instead of being a set of actions expressed in Structured Text, is another complete Sequential Function Chart. In this way, charts can be nested up to 20 levels deep providing an understandable heirarchy for large programs.

These two very simple constructs are supported by a number of other mechanisms for describing alternative paths, parallel paths, multi-level charts, abort sequences and the like.

The program editor includes facilities for cut and paste, moving, copying and renaming. It is designed such that it is difficult to draw syntactically incorrect charts and, even if this is achieved, the editor's built-in syntax checker will trap any error before the programmer can leave the editor.

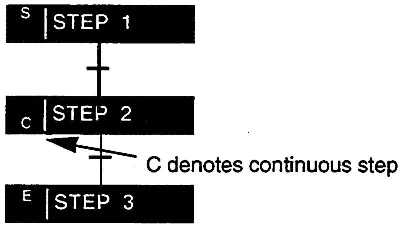
A few examples will illustrate the capability of the language.

Example 1



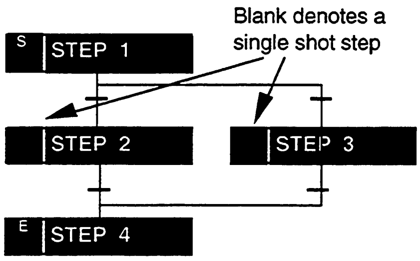
STEP_1 is executed **once** and the transition evaluated continuously until it becomes true. STEP_2 is then executed **once** and the program terminates.

Example 2



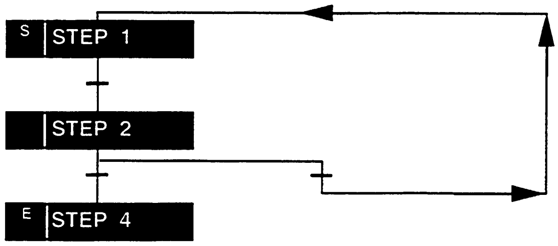
STEP_1 is executed **once** and the following transition evaluated continuously until true. Execution then passes to STEP_2 which executes **continuously** until the following transition becomes true. STEP_3 is then executed and the program terminates.

Example 3



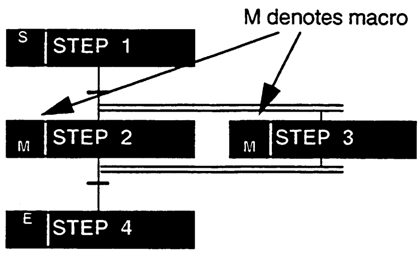
STEP_1 is executed once. The following transitions are continuously evaluated, strictly left to right, until one becomes true. The succeeding step is then executed, either STEP_2 or STEP_3 depending upon which transition was found to be true. Execution passes to STEP_4 once the relevant transition becomes true.

Example 4



The transitions after STEP_2 will determine whether the program loops back to execute again or terminates. This construct might used to perform a set of operations a given number of times with a counter being incremented in STEP_1 and its value being tested in the alternate transitions. It would also be used to perform a set of operations until a condition becomes true or while a condition is true.

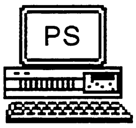
Example 5



STEP_1 is executed and, when the following transition becomes true, **both** STEP_2 and STEP_3 are executed. When both macros have executed to their respective end steps and the succeeding transition becomes true execution passes to STEP_4 and the program terminates.

This is an example of **parallel** execution and most control systems use this concept extensively. At the top level, a structure like this is used to divide the problem into Macros concerned with different aspects of the problem e.g. control, alarms, operator panel etc.

Programming



The Programming Software (PS) is an integrated environment for writing programs using the languages described in the preceding paragraphs, compiling the result, downloading it to a PC3000 and monitoring the execution of it in real time with on-line debug support. It is menu driven although the experienced user will find that he can type directly, if preferred. Language statements are checked for syntax as they are entered so that program errors can be flagged as soon as they are made instead of having to wait for the compiler to find them.

The software provides a directory of all programs and the ability to load, save, edit, rename and compile programs. Programs can also be merged so that they can be developed in small sections, possibly by different people, and subsequently joined together when the different sections have been tested in isolation.

The PS can produce documentation output in the form of a Structured Text listing of your program, including comments and Sequential Function Chart drawings, and a cross reference table of all variables.

It can also produce output about your program which is of use to Supervisory software packages by identifying the communications addresses of every parameter in your program in a flat DOS text file format. These files can be post-processed to automate the generation of 'gate' files for these systems.

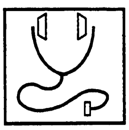
The PS will operate in real-time providing user access to all parameters in the program. Parameters can be read and changed on-line providing a powerful commissioning and debug tool. The execution of the sequence program is indicated by highlighting the active steps. Placing the cursor on any variable displays its current value.These de-bug facilities are considerably enhanced by the use of **User Screens**.

User Screens

The on-line capabilities of the Programming Software are considerably enhanced by the ability to define User Screens as part of the programming environment. Normally, in order to read or modify parameters from the PS, the Function Block parameter screen must be displayed. A Function Block Parameter screen provides all information concerning one particular function block. When commissioning and debugging, however, it is usual to require access to a variety of parameters from different function blocks and the User Screen provides this facility.

Any number of such screens can be defined using the normal processes of direct typing or menu pick lists. Any number of parameters of whatever type may be displayed on a User Screen together with explanatory text thus effectively producing a custom built commissioning tool for diagnostics and debug. The parameters on a User Screen retain all of their read/write access, formatting, units etc. that they have on their relevant Function Block Parameter screen.

Diagnostics



Hardware

All I/O modules, including communications modules have LED indication of I/O state and/or communications activity visible from the front of the rack.

Status information

All I/O function blocks, including communications function blocks, have Status_Info parameters. These are multi-value parameters giving a precise indication of the condition of the I/O channel. For an Analogue Input, for instance, the Status _Info provides the following diagnostic information.

Reset	The channel is in Reset.
Ok	The channel is functioning normally
No_Mod	There is no module in the rack at this position.
Wrg_Mod	The module at this position is not an Analogue Input module
Init	The module has just been powered up and is initialising itself.

No_Cal	The channel has not been calibrated or the data has been corrupted.
Calib	The channel is being calibrated.
Over_R	The input is over range
Under_R	The input is under range

This information can be used during commissioning for debug purposes and later on for fault diagnosis but the program can also use this data to issue meaningful alarm messages, implement redundancy or to define recovery mechanisms in the event of failure.

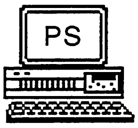
On-line program monitoring

When used in on-line mode, the PS is a powerful diagnostic tool. Any parameter may be viewed and modified, if necessary. Important parameter groups may be viewed simultaneously on a User Screen. The sequence program execution may be monitored by virtue of the highlighting that indicates active steps and parameter values may be interrogated to discover, for instance, the transition being tested and the value causing it to remain false.

There is also an on-line program monitor which operates automatically to log system events and errors. The last 50 such events can be inspected and will include such events as Start Up, WatchDog and floating point maths errors (such as divide by zero).

The program execution is also monitored to provide an indication of the processor utilisation. This is very useful, especially for large applications, when re-assigning parts of the program to slower tasks, in order to achieve optimal performance.

Programming Station



The PS is relatively insensitive to the computer type used but the following is an outline specification for a suitable machine.

Machine type .... 386 / 486 personal computer or equivalent

Hard disk.......................... >20M byte free space

Floppy disk..................... Single 3.5" drive

Memory .... 3M byte extended i.e 4M byte total.

Graphics........................... VGA

Monitor .... VGA monitor (colour preferred)

Communications .... At least 1 serial port for program down load

Operating system .... DOS version 6.2

Full details for running as a DOS application under WINDOWs or OS/2 is provided in the documentation.

Note: The serial communications port on the computer will normally be RS232 whereas the PC3000 standard LCM protocol is RS422. In order for the two to communicate, an RS232/422 converter and the necessary cables will be required. The Eurotherm Model 261 is a suitable converter and the cables are also available from Eurotherm. The LCM-Plus has an RS232 port so can be connected directly with a suitable cable.

User Interfaces

Euro Panel

For applications requiring a simple operator interface the Europanel2 provides 2 lines of 40 characters, numeric keypad, up/ down and scroll keys and 3 function keys. There are 5 LED indicators along the left hand edge. It communicates with the PC3000 through one of the RS485 ports. It is not possible to multi-drop the panel and so one PC3000 port is required for each panel in a system.

The Euro_Panel2 function block provides PC3000 program support and handles the functions of data entry, cursor movement, screen updates etc. The programmer merely provides the function block with the information concerning the parameters to display, their format and whether they are read/ write or read only and the function block does the rest.

In many cases, this panel is all that is required but it can also perform the function of a local panel with limited functionality in the situation where a more complex terminal is the primary operator interface but some distributed capability is also required.

Eurotherm Supervisory Package (Wizcon for DOS or Wizcon for Windows®)

Wizcon runs on any appropriate personal computer under DOS or W95/NT and provides the ability to construct a sophisticated operator interface as well as including supervisory functions such as report generation, alarm and event logging, trend logging and chart drawing. It is supported by a powerful, English like, logical language which enables complex supervisory control functions to be created.

Wizcon is provided with a wide range of communication drivers to talk to virtually any PLC thus enabling the integration of equipment of differing origins into an overall control scheme.

Slimline and Baseline versions of Wizcon with restricted numbers of gates and drivers provide very competitive solutions for smaller applications.

System Capability

Hardware

I/O Modules per rack 12
Max no. of racks per system 8
Max RAM per LCM ....756k + 128k
Max RAM per LCM-Plus ....1408k +128k

Software

The three limiting factors on the size of a PC3000 program are

- Maximum of 96 I/O modules.
- available memory space
- The time taken in execution.

The memory requirement of a program is determined by the sum of the individual requirements for

- The number of function blocks
- The size of the Structured Text soft wiring
- The number of SFC steps
- The size of the Structured Text statements in SFC steps
- The number of SFC transitions
- The size of the Structured Text conditions in SFC transitions

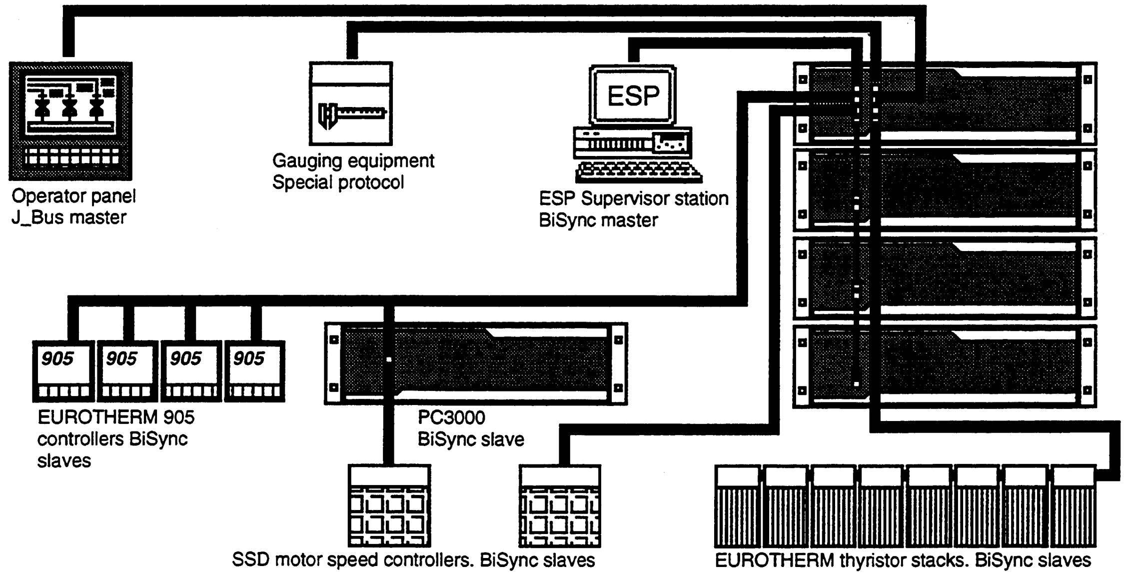
Whereas the memory requirement for each Function Block and each instance of an SFC step or transition can be determined, the structure and complexity of the Structured Text statements is entirely under the control, of the programmer and cannot easily be predicted.

The execution time considerations are not easily determined but, usually will be limited by the number of Function Blocks executing in the fastest task. These are those blocks concerned with digital I/O, timers and counters and any soft wiring connecting them together. It should be remembered that the size of the SFC program has no bearing on the time taken to execute it since only those steps that are currently active will be executing at any given moment.

For large applications there is a trade off between the number of complex analogue Function Blocks which execute in a slow task and the number of simple digital Function Blocks which execute in a fast task. In order to increase the size of an application, it is possible to reassign Function Blocks and SFC steps to a slower task and, hence, increase the number of them.

These considerations demonstrate the flexibility of the PC3000 but they do, however, make it difficult to specify absolute limits for its capability. A real application example will serve to provide an indication.

Application example



The example has two PC3000s controlling different parts of a production line but communicating with each other. Four EUROTHERM 900EPC instruments perform local control and data acquisition on the same communications link. Motor speed control is by Eurotherm Drives 590 drives communicating with the PC3000s and the heaters are controlled via EUROTHERM TU1450 communicating thyristor stacks. There is some specialist gauging equipment which uses a custom protocol and a local operator station which is the 9" monochrome CRT terminal using Modbus protocol. The line is supervised by an ESP supervisor system. The PC3000 is not only performing as a Process Control system but also as a very effective communications gateway.

The main PC3000 is a four rack system which is the 'master' controller for the line, all other instrumentation being slaved from it. It is responsible for temperature control of 32 zones, long loop control of the line speed involving two control loops and the two motor speed controllers, start up and shut down sequencing, alarm monitoring and various other minor tasks.

The program dimensions for the main PC3000 are as follows.

- Digital Inputs 40
- Digital Outputs 26
- Analogue Inputs 44
- Analogue Outputs 4
- Communications ports .... 7 — 3 x BiSync master, 2 x BiSync slave, 1 x Raw_Comms, 1 x J_Bus slave
- No. of SFC charts (macros) 62
- No. of SFC steps 320
- No. of SFC transitions 390
- No. of function blocks .... 720 (includes 34 PIDs)
- Program size .... 170 Kbytes

The other PC3000 is a relatively simple, single rack system providing local control and data acquisition for the main PC3000.

This system operates with two tasks; all the digital I/O and function blocks execute in a 10ms task and the rest, including the Sequential Function Charts execute in a 100ms task. The system could be made considerably larger by selectively moving less critical parts of the program into slower tasks thus providing execution time for more program elements.

Product Coding

Note VERSION3 is CE compliant and all modules will fit in VERSION2 racks.

Programming software

PS Programming S/W on 3.5" discs and all Manuals

PC3000/PS/DOS/ENG/3INCH/*swvers*

swvers = **320, 312, 300** or other Function Block Library.

Manuals Only **PC3000/A5MANUALS**

PC3000 Hardware

Main Rack **PC3000/RACK/VERSION3/MAIN/***mounting/blanks*

Extension Rack **PC3000/RACK/VERSION3/EXT/***mounting/blanks*

mounting = **FRONT** or **REAR** (bracket position)

blanks = **NOBLANK, 1BLANK, 2BLANKS** etc

= number of unused module slots

Power Supply (one per rack) **PC3000/PSU/VERSION3/HAC/NONE**

Enhanced Local Control Module - 512k base RAM

PC3000/LCM-PLUS/VERSION3/slot1/slot2/swvers

slot 1 = **NOMEM, 128KRAM, 512KRAM** extension RAM

slot 2 = **NOMEM, 128KRAM, 512KRAM** for downloadable FBs.

swvers = **320** (only)

Standard Local Control Module - 256k base RAM. For replacement only

PC3000/LCM/VERSION3/*slot1/slot2/swvers*

slot 1 = **NOMEM, 128KRAM, 512KRAM** extension RAM

slot 2 = **NOMEM, 128KRAM** for downloadable FBs.

swvers = **320, 312, 300** or other Function Block Library.

Intelligent Communications Module

PC3000/COMM/VERSION3/PORTS4/NOMEM/*swvers*

swvers = **320, 312, 300** or other Function Block library.

Must match adjacent LCM!

Cables for LCM and ICM ports

PC3000/CABLE/LCM232/9PIN/3.0M	LCM-Plus Port to 9 pin RS232
PC3000/CABLE/COMM/25PIN/3.0M	Port to 25 pin D, RS422
PC3000/CABLE/COMM232/25PIN/3.0M	ICM Port to 25 pin D, RS232
PC3000/CABLE/COMM/9PIN/3.0M	Port to 9 pin D, RS42
PC3000/CABLE/COMM232/9PIN/3.0M	ICM Port to 9 pin D, RS232

Rack Interface Module for Extension Racks.

PC3000/RIM/VERSION3

Cable for LCM to RIM, or RIM to RIM

PC3000/CABLE/COMM/COMM/*length*

length = **1.5M, 4.0M, 6.0M.**

Europanel2 **PC3000/PANEL/VERSION2/EP2X40/***mounting*

mounting = PANELMNT, or SURFACEMNT

(PANELMNT = through hole mounting)

Cable **PC3000/CABLE/COMM/EP2X40/3.0M**

Note 24V aux supply required.

PC3000 Hardware - I/O Modules

Analogue Input **PC3000/AI/VERSION3/***type*

type = **mV4**	4 channels, for 0 to up to 100mV, or thermocouple
= **mA4**	same module as mV4 but with 4 off 5 ohm shunts
= **V4**	4 channels, for 0 to up to 10V
= **RT4**	4 channels, for PT100
= **mV3/HiZ1**	3 channels as mV4, 1 channel 100M ohm input impedance
= **V2/FV2**	2 channels as V4, 2 channels 0 to up to 10kHz frequency to voltage.
=**mA2**	2 channels t/c, mV, mA, V or PT100.

Analogue Output	**PC3000/AO/VERSION3/***type*		
	type	= **mA4**	4 channels, links set for mA
		= **V4**	4 channels, links set to Volts
		Same module, individual link per channel.	
Digital Input	**PC3000/DI/VERSION3/***type*		
	type	= **HL14**	14 channels, 85 to 264V AC
		= **ACLL14**	14 channels, 24V AC
		= **24LL14**	14 channels, 24V DC
		= **5LL14**	14 channels, 5V DC
		= **CC14**	14 channels, contact closure
Digital Output	**PC3000/DO/VERSION3/***type*		
	type	= **RLYSTD12**	12 n/open contacts (1 amp)
		= **LGC12**	12 open collector (100mA)
Pulse Input	**PC3000/PI/VERSION3//5Q2**		
		2 channels, quadrature inputs, 5V. Link change for 12V.	

Miscellaneous items

Extra 128k RAM	**SUB3K/SPARE/128K/VERSION3/PCB**
Extra 512k RAM	**SUB3K/SPARE/512K/VERSION3/PCB**

RS422/RS232 convertor/isolator	**261/230/00** (for 240V) **261/115/00** (for 110V)

Cable 261 to Personal Computer
PC3000/CABLE/9PIN/261/3.0M

INTERNATIONAL SALES AND SERVICE

AUSTRALIA
Eurotherm Pty. Ltd.
Telephone Sydney (+61 2) 96348444
Fax (+61 2) 96348555

AUSTRIA
Eurotherm GmbH
Telephone Vienna (+43 1) 7987601
Fax (+43 1) 7987605

BELGIUM
Eurotherm B.V.
Telephone Antwerp (+32 3) 3223870
Fax (+32 3) 3217363

DENMARK
Eurotherm A/S
Telephone Copenhagen (+45 31) 871622
Fax (+45 31) 872124

FRANCE
Eurotherm Automation SA
Telephone Lyon (+33 478) 664500
Fax (+33 478) 352490

GERMANY
Eurotherm Regler GmbH
Telephone Limburg (+49 6431) 2980
Fax (+49 6431) 298119
Also regional offices

HONG KONG
Eurotherm Limited
Telephone Hong Kong (+852) 28733826
Fax (+852) 28700148
Telex 0802 69257 EIFEL HX

INDIA
Eurotherm India Limited
Telephone Chennai (+9144) 4961129
Fax (+9144) 4961831

IRELAND
Eurotherm Ireland Limited
Telephone Naas (+353 45) 879937
Fax (+353 45) 875123

ITALY
Eurotherm SpA
Telephone Como (+39 31) 975111
Fax (+39 31) 977512
Telex 380893 EUROTH I

JAPAN
NEMIC-LAMBDA K.K.
Eurotherm Div.
Telephone Tokyo (+81 3) 34476441
Fax (+81 3) 34476442

KOREA
Eurotherm Korea Limited
Telephone Seoul (+82 2) 5438507
Fax (+82 2) 545 9758
Telex EIKOR K23105

NETHERLANDS
Eurotherm B.V.
Telephone Alphen a/d Ryn (+31 172) 411752
Fax (+31 172) 417260

NEW ZEALAND
Eurotherm International (NZ) Limited
Telephone Auckland (+64 9) 2635900
Fax: (+64 9) 2635901

NORWAY
Eurotherm A/S
Telephone Oslo (+47 66) 803330
Fax (+47 66) 803331

SPAIN
Eurotherm España SA
Telephone (+34 91) 6616001
Fax (+34 91) 6619093

SWEDEN
Eurotherm AB
Telephone Malmo (+46 40) 384500
Fax (+46 40) 384545

SWITZERLAND
Eurotherm Produkte (Schweiz) AG
Telephone Zurich (+41 55) 4154400
Fax (+41 55) 4154415

UNITED KINGDOM
Eurotherm Controls Limited
Telephone Worthing (+44 1903) 695888
Fax (+44 1903) 695666

U.S.A
Eurotherm Controls Inc.
Telephone Reston (+1 703) 4714870
Fax (+1 703) 7873436
Web www.eurotherm.com

ED 15

http://www.eurotherm.co.uk

Part No. HA022230 Issue 5.1