METROLAB Instruments SA

PDI 5025

PRECISION DIGITAL INTEGRATOR

User Manual

Version 2.2

(Revision 2.0)

January 2000

1. INTRODUCTION 1

2. THEORY OF OPERATION 3

2.1 Transfer Equations 4
2.1.1 Description 4
2.1.2 Model 5150 Integrator and signal conditioning 5
2.1.3 Microprocessor calculations 7

3. DESCRIPTION OF THE PDI 5025 PLUG-IN MODULES 9

3.1 5150 Integrator Module 9
3.1.1 Preamplifier 9
3.1.2 Bargraph Display 9
3.1.3 Voltage to Frequency Converter (VFC) 10
3.1.4 Integration Counters 10
3.1.5 Manual offset adjustment 10
3.2 5140 Trigger Module 12
3.2.1 Timer 12
3.2.2 Incremental Encoder 12
3.2.3 Motor control 12
3.3 5130 Interface Module 15
3.3.1 Microprocessor 15
3.3.2 RS 232 Port 17
3.3.3 IEEE-488 Port 17
3.4 5120 Power Supply Module 18

4. SPECIFICATIONS 19

4.1 Inputs 19
4.2 Outputs 20
4.3 Accuracy 20

5. INSTALLATION AND CONNECTIONS 25

5.1 5150 Integrator Module Analog Part 25
5.2 5150 Integrator Module Digital Part 26
5.3 5140 Trigger Module - Encoder Connections 28
5.4 5140 Trigger Module - External Synchronisation 29
5.5 5140 Trigger Module - Motor 29
5.6 5130 Interface Module 29
5.7 5130 Interface Module IEEE 488 30
5.8 5130 Interface Module RS 232 C 30

6. INSTRUMENT SETTINGS 33

6.1 Power On Sequence 33
6.2 Channel Selection 33
6.3 Gain Selection and FNC Button 33
6.4 Offset adjustment 34
6.5 Integrator Mode 35
6.6 Programming The Integration Intervals 35
6.6.1 Trigger Source: Timer 37
6.6.2 Trigger Source: Encoder 37
6.6.2.1 Rotational Encoder with Index 37
6.6.2.2 Linear Encoder with Index 38
6.6.2.3 Linear or Rotational Encoder without Index but with External Synchronisation. 38

6.6.2.4 Linear or Disk Encoder without Index or External Synchronisation. 39
6.6.3 Trigger Source: External 39
6.6.4 Fluxmeter - Autonomous/Talker only mode 39

7. PROGRAMMING THE PDI 5025 41

7.1 The PDI 5025 Status Registers 41
7.1.1 STATUS 1: Measurement Status (1 byte) 41
7.1.2 STATUS 2: Error Status (1 byte) 43
7.1.3 STATUS 3: Trigger Source Module 5140 (1 byte) 44
7.1.4 STATUS 4: PDI module 5130 (1 byte) 45
7.1.5 STATUS 5 and 6: Autotest (2 bytes) 46
7.1.6 STATUS 7: Acquisition status 47
7.2 Data Acquisition 47
7.2.1 Individual data storage (CUM,0) 48
7.2.2 Cumulated data storage (CUM,1,S) 48
7.2.3 Last cumulated data storage (CUM,1,L) 48
7.2.4 Direct Data Transfer (IMD,1) 48
7.2.5 Block Data Transfer (IMD,0) 49
7.2.6 End of Data (EOD) 50
7.2.7 Interrupting a measurement cycle (BRK) 50
7.2.8 Automatic Use of a Motor (MOT,A) 51
7.2.9 Autotest (TST and AUT) 51
7.2.10 Internal Short Circuit (ISC) 52
7.2.11 Synchronisation by the Host Computer (SYN) 53
7.2.11.1 RS 232 C 53
7.2.11.2 IEEE 488 53
7.3 The IEEE 488 Interface Commands 54
7.3.1 The "REMOTE" Command 54
7.3.2 The "LOCAL LOCKOUT" Command 54
7.3.3 The "GO TO LOCAL" Command 54
7.3.4 The "GROUP EXECUTE TRIGGER" Command 54
7.3.5 The "INTERFACE CLEAR" Command 54
7.3.6 The "DEVICE CLEAR" Command 54
7.3.7 Service Request (SRQ) 55
7.3.8 Setting the SRQ Masks 55
7.3.9 The SRQ Status Register 56
7.3.10 SRQ Masks command (MSK) 57

8. APPENDIX I. PDI 5025 COMMANDS 59

9. APPENDIX II. COMMAND REFERENCE GUIDE 67

10. APPENDIX III. PDI 5025 JUMPERS AND MICROSWITCHES 69

10.1 5150 INTEGRATOR MODULE: DIGITAL BOARD: JUMPERS 69
10.2 5150 INTEGRATOR MODULE: ANALOG BOARD: JUMPER 70
10.3 5140 TRIGGER MODULE: JUMPERS 71
10.4 5130 INTERFACES MODULE: JUMPERS 72
10.5 5130 INTERFACES MODULE: MICROSWITCHES 73
10.6 5130 MODULE: MICROSWITCHES: BAUD RATE SELECTION 73

11. APPENDIX IV. QUICK REFERENCE TO STATUS REGISTERS 75

12. APPENDIX V. SOFTWARE EXAMPLES 77

1. INTRODUCTION

The Precision Digital Voltage Integrator Model PDI 5025 forms part of a system for the measurement and analysis of magnetic fields.

In this system the PDI 5025 is connected to one or two sense coils and **a host computer that may be a minicomputer, a personnel computer (PC) or a simpler device such as the pocket computer by Psion.** The host computer initializes the PDI 5025 for a definite series of measurements, then collects the results and performs the required data analysis.

The main functions of the PDI 5025 are the:

- integration of the pick-up coil voltage relative to the internal time base, or mechanical preset increments (angular or linear) generated by the motion of the pick-up coil.

- steering a DC motor, for rotational or linear motion of the coil, and monitoring its position with an incremental encoder.

- transmission of the results to the external computer.

The PDI 5025 front panel houses all the instrument's connectors (with the exception of the power connector), the Gain and Level displays, the buttons for local commands and the microswitches for setting the communications parameters with the host computer. All other commands must be generated from the host via the computer interface.

The PDI 5025 can be packaged either in a bench top case or a 19" 3U modular chassis for rack mounting. Internal sub-functions are performed in modular plug-in units equipped with electronic cards in Euroboard format 100x160mm.

The internal communication between cards is based on the G-64 bus (Gespac); this solution allows for the future development of this system.

Although developed to perform magnetic field measurements the PDI 5025 is in fact a "Digital Voltage Integrator" and thus the results and equations in this document are expressed in Volt x Seconds (Vs).

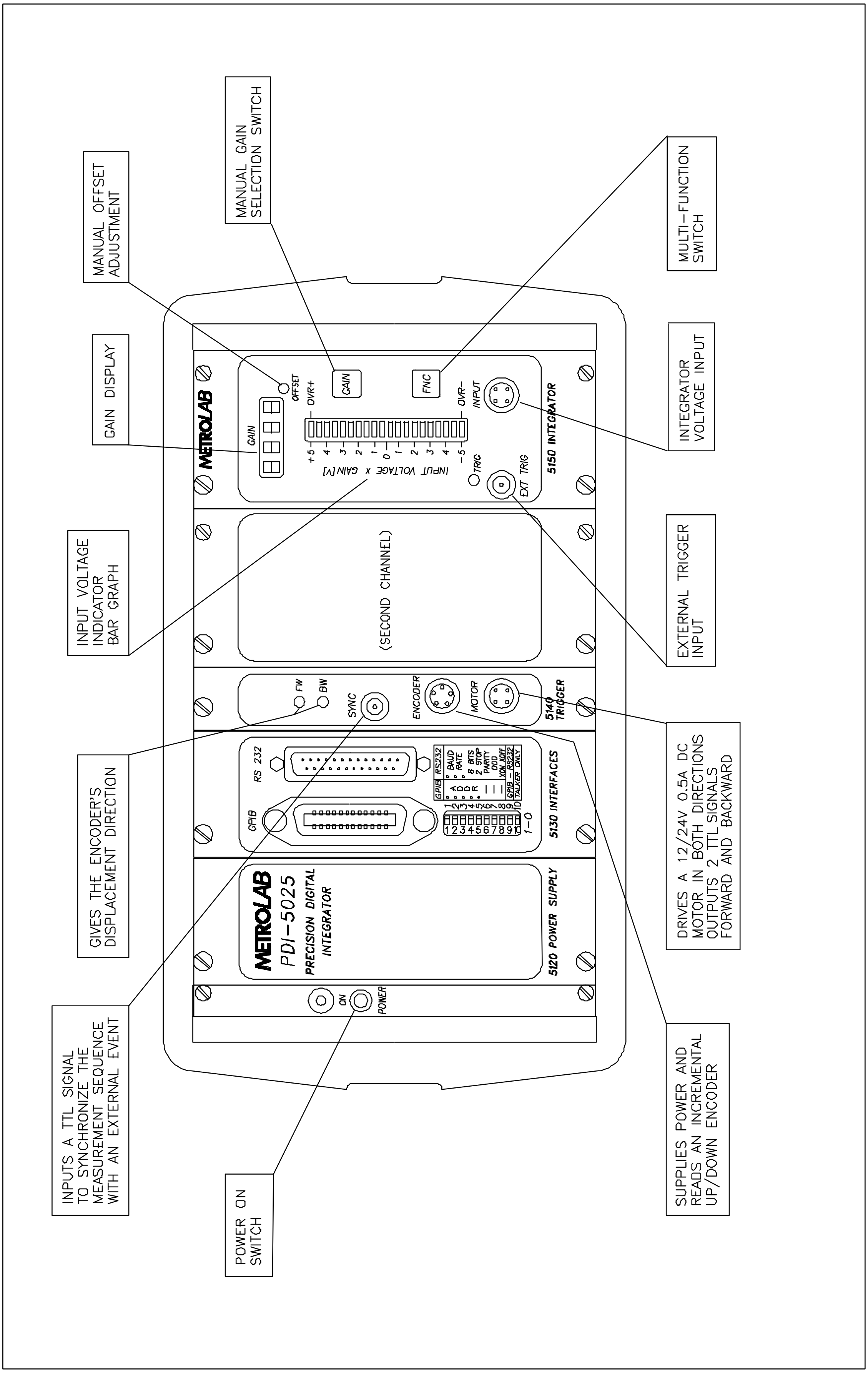


figure 1: PDI 5025 Front Panel

2. THEORY OF OPERATION

The PDI 5025 Precision Digital Integrator functions in the following manner (the block diagram is shown in figure 2):

- The input voltage is conditioned by a programmable gain preamplifier in order to deliver an output in the ±5V range.

- An **Overrange** detector monitors the output of the preamplifier to ensure that it stays within the ±5V limit. This output voltage is displayed on an indicator bargraph allowing the gain to be adjusted easily.

- The signal is then shifted by +5V (high precision reference) in order to ensure a positive voltage for all measurement conditions.

- This voltage, in the 0-10V range, is fed to a precision voltage to frequency converter (VFC) which has an output frequency we shall call F. In parallel, a reference oscillator generates pulses at a fixed frequency (Fr) corresponding to 4 times the output of the VFC with +5V input; i.e. with a 0V preamplifier input voltage.

- The F and Fr pulse trains are sent to two 32 bit counters during the measuring period. At the end of each period, they are switched to two alternative 32 bit counters. This permits the non active counters to be read without loss of counts.

- The local microprocessor picks up the contents of each pair of counters at the end of each measurement period, calculates the difference which is proportional to the integral of the coil voltage during this period, then processes the data to generate the final result which is stored in memory prior to transfer to the host computer. This data is expressed in 10^{-8}Vs units

- A trigger circuit controls measurement sequences either with an internal programmable timer with a 1ms resolution or an external incremental encoder that indicates the motion of the coil during measurements.

- Finally, a DC motor control circuit delivers a ±12V or ±24V/0,5A power signal to drive a motor in either direction. For motors which need more power, 2 TTL signals are provided for forward (FW) and backward (BW) commands of an external supply. **These outputs can be used as inputs if short circuited to ground to control the internal motor power supply in order to perform a manual conditioning of the mechanical system before an automatised sequence is undertaken.**

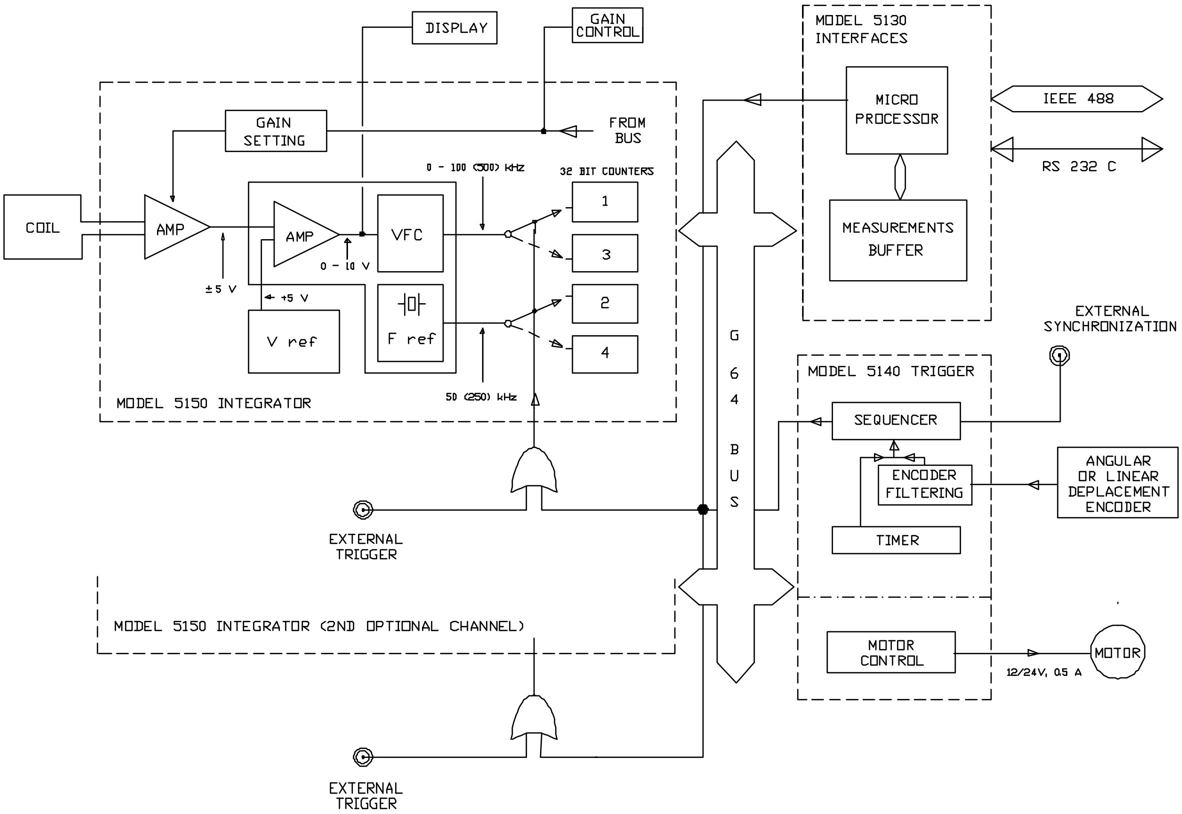


figure 2: PDI 5025 block diagram.

A local microprocessor controls the measuring sequences (Trigger generation), motor power, reception and decoding of commands from the host computer, front panel commands , display functions and the transfer of data to the host. The instrument can store up to 5200 measurement results before a transfer to the host is necessary.

This local microprocessor may drive one or two integrator channels working synchronously.

2.1 Transfer Equations

2.1.1 Description

In this section there is a brief description of the way the output measurements are obtained via the measurement chain, so that the user may clearly understand the transfer function of each element of the PDI 5025 and the software calculations it performs.

The overall transfer function of the PDI 5025 is to send measurements to the host computer in a 32 bit digital signal with a sign bit, in numerical integer form, expressed in 10^{-8} Volt*seconds units.

In the case of a magnetic field measurement with a coil of **N** turns and a surface area of **S** in m^2, the voltage generated is given by the formula:

$$Vi = -N \cdot d\Phi/dt$$

where $\Phi = B \cdot S$ (Scalar product).

The next section describes the successive operations of the 5150 Integrator unit, and the software calculations performed by the instrument.

2.1.2 Model 5150 Integrator and signal conditioning

The coil voltage Vi is first amplified in the preamplifier whose gain will be called G.

Then the voltage V1 (=Vi·G), which must be within the ± 5V range, is shifted by a + 5V reference value in order to ensure that the resulting voltage V_2 is always positive.

V_2 is applied to the VFC, whose input range is 0 to 10V for the output frequency of 0 to Fn. Fn, the maximum frequency of the VFC, corresponding to a 10V input, may have 3 values depending on the different configurations of the 5150 Integrator unit: 100KHz , 500KHz or 1MHz.

We shall call the current output frequency of the VFC, corresponding to the V_2 input, F, having a value between 0 and Fn hertz.

In parallel, the VFC generates a reference frequency Fr whose value is 2Fn. This frequency represents 4 times the VFC output frequency for a 5V input voltage V_2 which corresponds to a 0V coil voltage Vi.

The two pulse trains F & Fr are sent in the two active counters during the programmed period (δT) or motion.

At each trigger signal, the content of each counter (N and Nr) is transferred to the microprocessor, while 2 other counters resume counting. The values N and Nr will be then be processed by the microprocessor, according to the rules defined in next section, before being transferred to the host computer.

The transformation of the input signal in the PDI Module may be expressed by the following equations:

$V_1 = G \cdot Vi$ in the preamplifier (G = 1,...,1000)

The DC shift of +5V is expressed by:

$$V_2 = G \cdot Vi + 5$$

If we call C the transfer function of the VFC expressed in Hz/Volt, the voltage value for obtaining F, we can write:

$$F = C \cdot (G \cdot Vi + 5)$$

$$F = C \cdot G \cdot Vi + 5 \cdot C$$

The expression $5 \cdot C$ which represents the frequency for which $Vi = 0$ has a value equal to 1/4 of Fr, described above, since $C = Fn/10$ and $Fr = 2Fn$; this leads to the final expression:

$$F = C \cdot G \cdot Vi + \frac{1}{4} \cdot Fr$$

If T_1 and T_2 are the times of the start and end of a counting period, between two trigger signals, the count results N and Nr will represent the integral of F and Fr between T_1 and T_2. As a consequence the former expression, when integrated, can be expressed by:

$$\int_{T1}^{T2} F \cdot dt = C \cdot G \cdot \int_{T1}^{T2} Vi \cdot dt + \frac{1}{4} \cdot \int_{T1}^{T2} Fr \cdot dt$$

We know that:

$$N = \int_{T1}^{T2} F \cdot dt$$

and that:

$$Nr = \int_{T1}^{T2} Fr \cdot dt$$

and if we call R the result of the integration equation (ie the integral between T_1 and T_2 of the input voltage Vi), N can be expressed as:

$$N = C \cdot G \cdot R + \frac{1}{4} \cdot Nr$$

or

$$R = (N - \frac{1}{4} \cdot Nr)/C \cdot G$$

This is the relation binding the output counts of the module and the desired result. It depends on gain value of the preamplifier, configuration of the VFC and flux variations.

To get an absolute measurement of the Voltage integration, the microprocessor reads the fixed parameters (C,G) of the 5150 Integrator module in the measuring situation, and performs the calculations described below.

2.1.3 Microprocessor calculations

As explained in the introduction, the desired result is R in Volt·Seconds or a multiple thereof, despite the successive modifications of the signal.

The information available on the G-64 bus are the counts N and Nr, the Gain value, and a code defining Fn.

Using these data, the microprocessor calculates the result by employing the following formula, directly derived from the previous ones:

$$R = (4 \cdot N - Nr)/(4 \cdot C \cdot G)$$

In order to only manipulate integers, N is multiplied by 4 and R is expressed in units of 10^{-8} Volt.Seconds because of the different possible values of the 4·C·G coefficient.

So, in any measuring configuration, the microprocessor:

- firstly, multiplies N by 4

- secondly, subtracts Nr, and stores the result in the RAM, ready for a next acquisition and precalculation

- thirdly, when it has time, it applies a multiplying factor to the subtraction result, depending on the 4·C·G value. Namely, this coefficient:

$$K = 10^8 / (4 \cdot C \cdot G)$$

The result is also stored in the memory, whilst awaiting transmission to the host controller.

Depending on the possible values and combinations of C and G, there are 30 coefficients stored in EPROM selected according to the effective C·G product.

Thus results are always expressed in the same units, despite different Gain settings.

The consequence, due to the maximum capacity of the N and Nr counters, is that the maximum measuring range and resolution depend on the value of the C·G product. It is interesting to note that the 32 bit capacity is equivalent in both counters; since F is expressed in form 0 - Fn a 31 bit number plus sign, and Fr in form of 2Fn a 32 bit number, with 4 times higher resolution as described above.

The next table gives the values of maximum range and resolution for each possible C and G combination.

		G : possible values of preamplifier Gain									
VFC	C	1	2	5	10	20	50	100	200	500	1000
100KHz	10000	40 [10000]	40 [5000]	40 [2000]	40 [1000]	40 [500]	40 [200]	40 [100]	20 [50]	40 [20]	20 [10]
500KHz	50000	40 [2000]	40 [1000]	40 [400]	40 [200]	40 [100]	40 [40]	40 [20]	20 [10]	40 [4]	20 [2]
1MHz	100000	40 [1000]	40 [500]	40 [200]	40 [100]	20 [50]	40 [20]	20 [10]	10 [5]	20 [2]	10 [1]

MEASURING RANGE & RESOLUTION TABLE

- Range is expressed in Volt.second in the upper number
- Resolution is expressed in 10^{-8} Vs in [..]

3. DESCRIPTION OF THE PDI 5025 PLUG-IN MODULES

The plug-in units of the PDI 5025 correspond to the different main functions described in the previous chapter. In this section, we shall describe in more detail the principal functions of each unit. For a better understanding, it is possible to refer to the general diagram of figure 1 and to the corresponding block diagram of each unit.

3.1 5150 Integrator Module

The 5150 Integrator module is recognized on the internal G-64 Bus by a hardware address set by positioning jumpers J2 located on the digital circuit board (see section 10.1). The position of this jumper may assign the 5150 module either to channel A or B. This assignment is displayed on the left digit of the Gain display, except when a gain of 1000 is selected. One or two of these modules may be installed.

Following the sense signal from the coil, we shall describe the elements that transform the signal up to the final pair of counters accessed by the microprocessor via the G-64 bus.

3.1.1 Preamplifier

The preamplifier is a precision instrumentation differential amplifier that may work with a balanced or unbalanced input. The cable's screen can either be linked to the floating ground or to a differential ground which is at the mid point between the two input voltages. In this case the two inputs **must** be isolated form the ground of the connector. Thus care should be taken when using a coaxial cable to connect the coil. Selection is made by jumpers (see description of jumpers in section 10.1).

The gain of the preamplifier may be set from 1 to 1000 in a 1, 2, 5 progression. The gain may be controlled manually by use of the front panel GAIN button or by the host computer via the external interface. In either case the value of gain is displayed on the front panel GAIN LED display.

All the analog parts of the integrator are fed by a floating ±15V supply in order to allow a common mode of input coil voltage reaching ±300V from ground. As a consequence, all commands (eg gain commands) and outgoing signals are transmitted via photocouplers.

3.1.2 Bargraph Display

On the front panel a LED bargraph display monitors the amplitude of the amplified signal from the coil as output from the preamplifier. The outer LEDs of this display correspond to the +5V and -5V levels and are the plus and minus overrange indicators. In the case of a signal exceeding the limit they remain lit until cleared by the host computer, by manually pressing the FNC button or by changing the gain.

3.1.3 Voltage to Frequency Converter (VFC)

The VFC circuit is a module containing a very high precision Voltage to Frequency Converter (F) it also contains the +5V DC shifter for the input signal and the Reference Oscillator (Fr).

3.1.4 Integration Counters

Two pairs of 32 bit counters are alternatively in charge of counting the F and Fr pulse trains. Following each trigger signal, counting is switched from one pair of counters to the other. Whilst one pair is counting, the contents of the other pair is transferred to the local microprocessor for preprocessing, the two counters are reset and made ready for the next trigger pulse which will switch counting back to them.

Surrounding the counters are the usual address decoders and latches necessary for data transfer also circuits required to condition the trigger signals for counter switching, counter reset, overrange transfer and front panel command processing.

Physically, the above mentioned functions are performed on two separate Euroboard cards respectively for the analog and digital parts. A third small card supports display components for the front panel. It handles command inputs and ensures data and power supplies transfers between the two main Euroboard cards.

3.1.5 Manual offset adjustment

A potentiometer accessible on the front panel allows the user to compensate the input offset voltage in order to lower the drift of the integrator. The offset adjustment has to be performed with no signal at the input, i.e static coil and static field or shorted plug at the input. The simultaneous depression of both buttons (GAIN and FNC) makes the offset adjustment easier by starting an automatic run and displaying input values (in multiples of the resolution values given in the table in section 2.1.3) on the front panel. Then adjust the offset potentiometer so that the displayed value is as close to zero as possible.

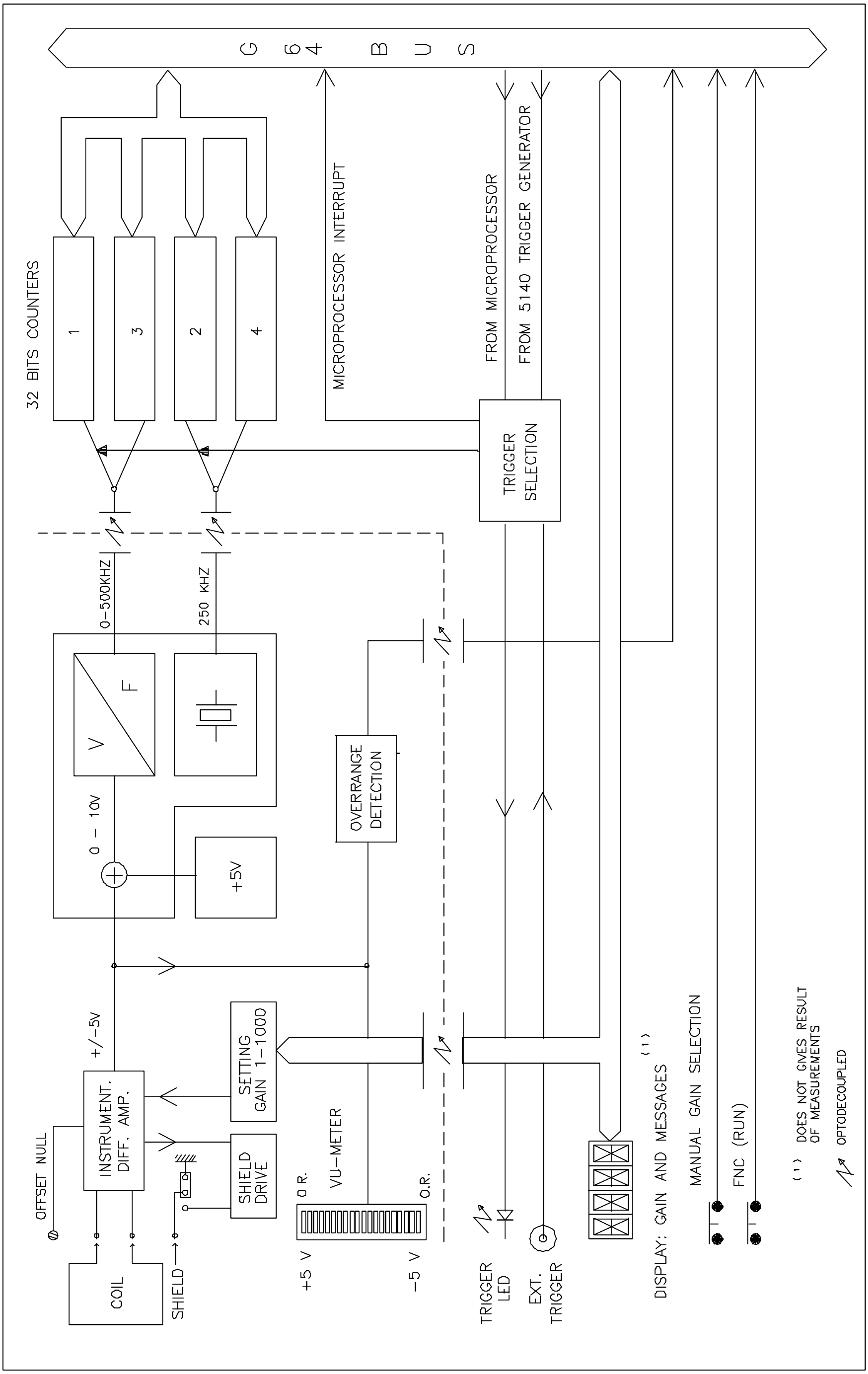


figure 3: PDI 5150 Integrator.

3.2 5140 Trigger Module

This unit consists of one Euroboard card and performs the functions described below.

3.2.1 Timer

The timer, driven by a precision quartz oscillator (±1ppm), is used to define the duration of the integration when measurements are performed with respect to time.

The resolution and minimum duration are both 1ms; although there is no maximum duration, one integration period (delimited by two triggers) has to be shorter than 2^{24} ms (about 4.6 hours).

3.2.2 Incremental Encoder

The encoder reading function is used to define the displacement of the coil when the measurements are performed with respect to space.

The circuit receives the two-phase signal from a bi-directional incremental encoder. In order to give maximum resolution, the circuit generates 4 pulses for each incoming encoder cycle. When programming the instrument, the user must remember to multiply by 4 the number of encoder cycles corresponding to the angle of displacement for the measurement.

The minimum time between triggers is 1ms.

An Index signal from the Encoder, if provided, is used to define the mechanical origin of the motion and reset the position counter in the initialisation operation. The encoder pulses, are sent to the same Trigger Source Counter used by the timer, so that the dynamic range is 2^{23} (ie more than 8 million pulses).

The 24th bit of the counter is used for the sign.

The SYNC input can be used to reset the counter in both encoder and timer modes and thus synchronise the measurement with an external event.

3.2.3 Motor control

The motor control function allows the host computer to drive a motor that is synchronised with the measurements.

On the "motor" socket, two pins correspond to the direct motor connection; according to the programmed sequence the output voltage between these pins may be 0, +12V or -12V for forward (FW) or backward (BW) motion. An internal jumper may convert these outputs in ±24V if required. The maximum output current is actively limited to 500mA.

If this current is insufficient two other pins in this socket, which deliver TTL signals for the forward (FW) or backward (BW) orders, may control an external power supply to drive the motor.

An additional feature of these pins is that they may be used (if short circuited to ground), as controls for the internal motor power supply. This allows manual control of the motor for mechanical trimming of the coil displacement mechanism.

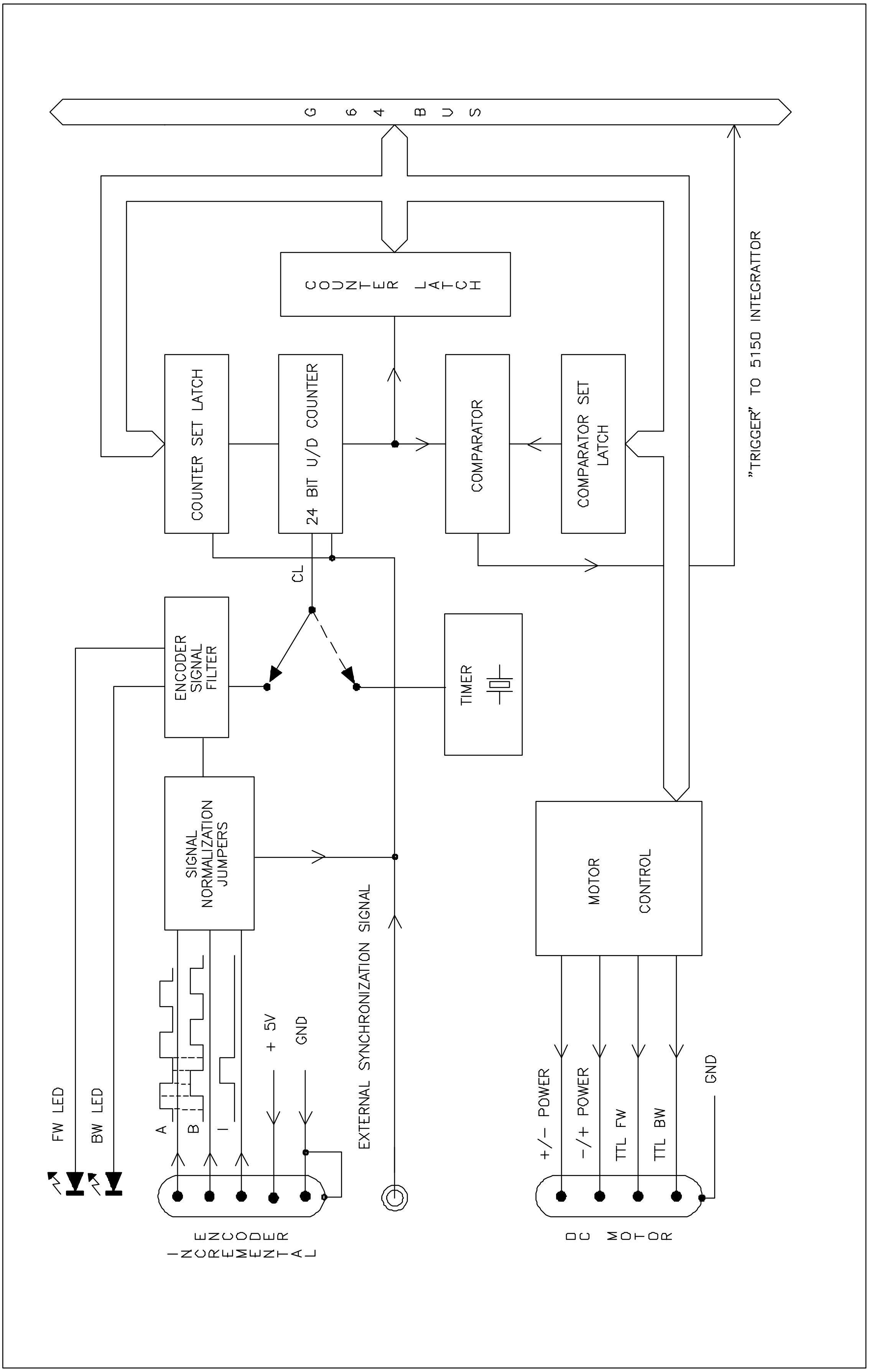


figure 4: PDI 5140 Trigger Generator.

3.3 5130 Interface Module

This unit comprises one Euroboard card containing the microprocessor, its peripherals, memory, G64 bus controls and the circuits driving the RS 232 C and IEEE 488 interfaces.

3.3.1 Microprocessor

The microprocessor is a Motorola 6809, with 32k bytes of RAM and a socket for a 32k byte EPROM.

The microprocessor handles communications with the host computer via the I/O ports , reception and decoding of commands, calculation, buffering and transmission of the results. Seven status registers can be interrogated to determine the current state of the instrument. See section 7.1 or appendix III for a detailed description of these registers.

Within the instrument, the microprocessor controls the different phases of measurement:

- steering of integrator counters; start, stop, reset and transfer of counts;

- position or time controls; mainly loading the set values and waiting for the comparator signal;

- motor commands and supply.

At start up, the microprocessor executes an autotest procedure which verifies the integrator parameters: zero offset, gain and overrange thresholds.

Ten microswitches situated on the front panel of the 5120 module control the configuration of the I/O ports.

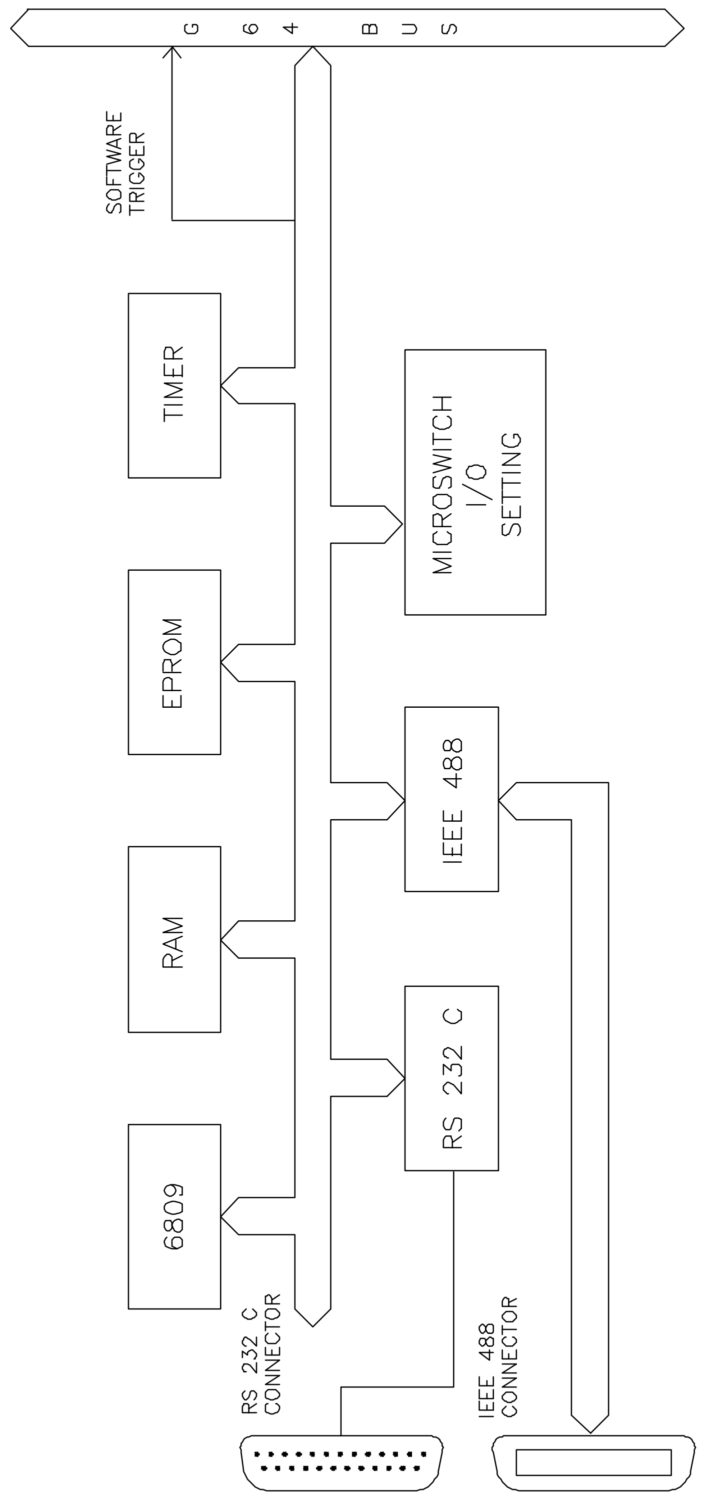


figure 5: PDI 5130 Microprocessor and Interfaces.

3.3.2 RS 232 Port

The RS 232 C interface can be used in a variety of configurations. The microswitch definitions can be found in section 5.8.

The principal functions that can be selected are:

- baud rates from 300 to 4800 (9600, 19200)[1]
- hardware handshake
- software handshake (X/ON, X/OFF)
- 7 or 8 bit word length
- number of stop bits
- presence and nature of parity bit.

The last microswitch selects the normal conversational mode or an autonomous mode in which the instrument outputs measurement data at is own rate. In this case, the trigger period is set to 300 ms in continuous mode.

The RS 232 works in Full Duplex mode.

3.3.3 IEEE-488 Port

The first five microswitches on the front panel are used to define the instruments IEEE Device Address. The microswitch definitions can be found in section 5.7.

The IEEE-488 interface conforms to IEEE standards. It can perform the following functions:

- SH1;AH1;TS;L4;SR1;RL0;PP0;DC1;DT1;C0;E2

The IEEE commands described below have the following effect on the PDI 5025:

- GET Group Execute Trigger generates a software SYNC signal to the 5140 Trigger module
- a SRQ can be generated by the PDI 5025
- SPE Serial Poll returns the SRQ Byte and clears its content.

The Parallel Poll (PPE) and Remote/Local (REN/NRE) commands are not used by the PDI 5025.

All unused commands will simply be ignored and will not affect the IEEE dialog.

---

1 See note in section 5.8.

3.4 5120 Power Supply Module

The Euroboard format plug-in power supply is a high efficiency switching power supply.

Its input voltage can be anything from 85 to 264 Volts Alternating (AC) and the line frequency between 45 and 440 Hz.

Output voltages are following:

- +5V/8A
- ±12V/2A

This power supply is thus adapted to the majority of mains voltages and does not require any setting before use of the instrument. The main fuse is located at the rear of this module.

4. SPECIFICATIONS

The following sections give the technical specifications of the PDI 5025.

4.1 Inputs

Integration channels:	1 or optionally 2 which are triggered simultaneously.
Input voltage range:	±5V divided by Gain (±5V to ±5mV depending on Gain setting).
Overvoltage protection:	±45V DC (will drain 28mA), ±600V during 100µs (will drain 1A).
Common mode voltage:	±12V vs connector shield; ±500V vs ground when the floating mode is set.
Input impedance:	balanced vs floating ground; 2 x 1MΩ; unbalanced 1000MΩ.
Boot strapping:	the connector shield can be driven at the average voltage of the + and - inputs.
Programmable gain	preamplifier gain is programmable in the steps: 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000.
Integration interval:	resolution (one or two channels): 1ms (timer) or 1 pulse (encoder). minimal time between two triggers: 1ms (all modes). maximum value: limited by counter capa-city (2^{23}ms or encoder pulses).
Ext. trigger:	TTL, not debounced, rising or falling edge with internal pull-down or pull-up resistors respectively (jumper selectable); parallel to the trigger generated by the 5140.
Synchronization:	TTL, debounced, rising or falling edge with internal pull-down or pull-up resistors respectively (jumper selectable).
Encoder:	2 UP/DOWN counter channels, 90° dephased and 1 index channel. Polarities are jumper selectable. The maximum frequency on the encoder input is 10^6 cycles/s.

4.2 Outputs

Computer interfaces: IEEE 488 and RS 232 C.

Output buffer: 5200 results can be accumulated during the measurement sequence prior to being transmitted to the host computer in 10^{-8} Vs units.

Encoder power: +5V and up to 500mA without short circuit protection.

Motor power: ±12V or ±24V actively limited internally to 500mA. Additional pins are allocated for external power unit control or motor manual control when short-circited to ground.

4.3 Accuracy

Setting time

Time to match an input step at 100ppm: 20µs (Gain = 1-200)
80µs (Gain = 1000).

Drift[1]

	Manual offset adjustment effect[2]		Temperature coefficient after 1h30 warm	
Gain	µV	ppm of max input range	µV/°C	ppm of max input range
1	±70	±7	+10	+1
10	±7	±7	+1	+1
100	±1.5	±15	+0.15	+1.5
1000	±1	±100	+0.1	+10

1 All values are referred to an input voltage.

2 Refer to an operator with an average skill level.

Drift variation versus time: max. per month 200µV
max. per year 500µV.
Divide by Gain in order to refer to input.

Gain

Gain linearity: max. Gain deviation from the straight line crossing the zero, in ppm of input range.

	VFC		
Gain	100 kHz	500 kHz	1 MHz
1-200	±10	±20	±60
1000	±50	±60	±100

Gain stability vs temp.: (max. temp. coefficient)
+7 ppm/°C (VFC 100kHz)
+9 ppm/°C (VFC 500kHz)
+25 ppm/°C (VFC 1MHz).

Gain stability vs time:
per month: ±30 ppm
per year: ±50 ppm.

Gain ratio error: max. ±100 ppm when interchanging the gain settings.

Gain adjustment: gain is factory set to 10 with ±10 ppm accuracy. The non linearity is balanced for the input voltage +0.4V and -0.4V.

Noise

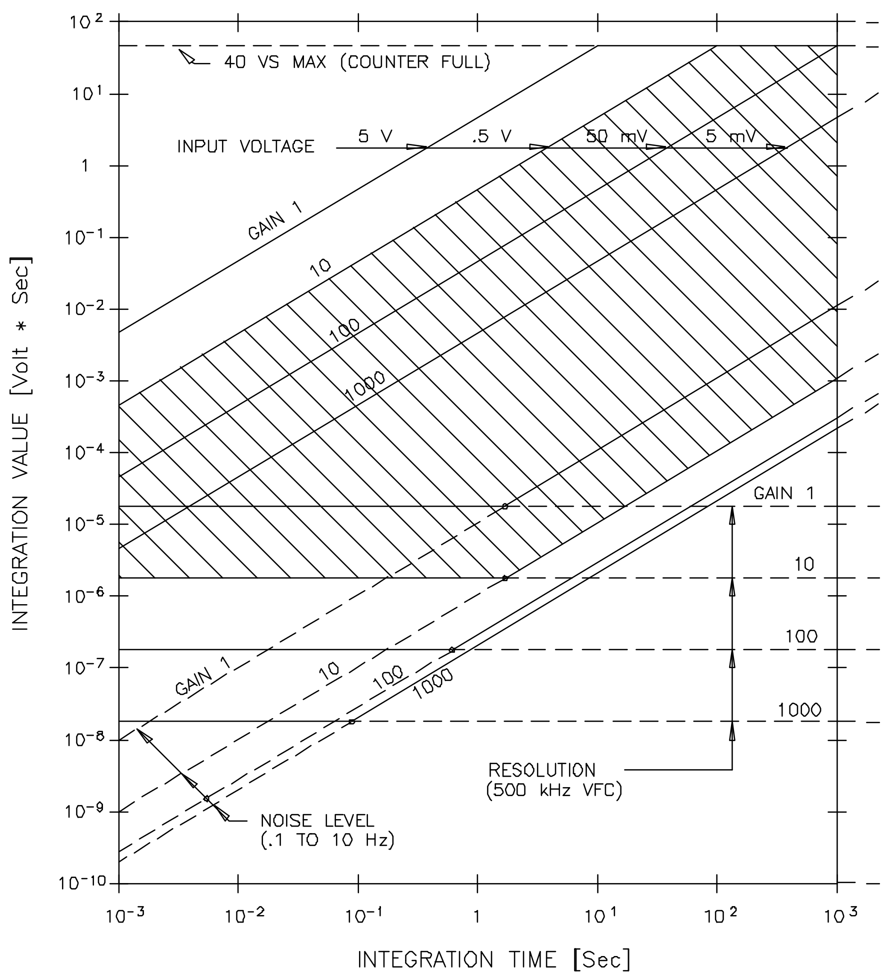
	Noise (0.1Hz to 10Hz; referred to input)	
Gain	(µV)	ppm of input range
1	±10	±1
10	±1	±1
100	±0.3	±3
1000	±0.2	±20

Resolution and maximum range:

		Gain									
VFC	C	1	2	5	10	20	50	100	200	500	1000
100KHz	10000	40 [10000]	40 [5000]	40 [2000]	40 [1000]	40 [500]	40 [200]	40 [100]	20 [50]	40 [20]	20 [10]
500KHz	50000	40 [2000]	40 [1000]	40 [400]	40 [200]	40 [100]	40 [40]	40 [20]	20 [10]	40 [4]	20 [2]
1MHz	100000	40 [1000]	40 [500]	40 [200]	40 [100]	20 [50]	40 [20]	20 [10]	10 [5]	20 [2]	10 [1]

In each box in the table, the lower number gives the resolution in units of 10^{-8}Vs and the upper number the maximum range in Vs. The resolutions only depend on the timing of the reading of the integration counters and on the calculation algorithm. To evaluate the global resolution of a measurement, the input noise must be taken into account.

The following diagram shows the measurement domains at different gains (hatched: G=10) and the noise levels.



General

Storage temperature:	-20°C to +70°C.
Operature temperature:	0°C to 40°C.
Max. DC field:	0.05 Tesla.
Power Supply:	voltage 85-264 VAC frequency 45-440 Hz power ~60W.
Packaging:	bench or 19" rack mounted (3 units high)
Dimensions:	(W x H x D, bench) 260 x 148 x 260 mm.
Weight:	5.5 kg (bench unit, one channel).

5. INSTALLATION AND CONNECTIONS

This chapter describes the installation of the PDI 5025 and its various modes of operation.

The PDI 5025 forms part of a complete measurement system. When it is used to measure magnetic fields using a moving coil the elements of the measurement system will comprise:

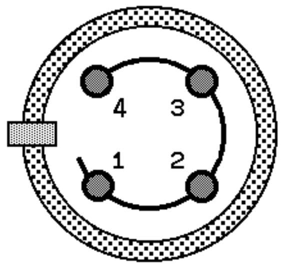
- one or two rotating coils
- an incremental encoder
- a DC motor with a reduction gear box
- a host computer (this is obligatory).

A set of jumpers allows the PDI 5025 to be adapted to different measurement environments. The following sections define, on a module by module basis, the function of each jumper. **To access the jumpers the user should remove the module from the chassis by unscrewing the retaining screw at the front of the instrument.** (Refer to appendix III for jumpers location.)

5.1 5150 Integrator Module Analog Part

The measurement coil is connected to the differential preamplifier by a four pin Lemo connector. Pin connections are shown below:

- pins 1 and 2: + input
- pins 3 and 4: - input



Lemo Connector for the Coil Input (Front View)

The analog ground can be left floating (±300V) or tied to ground with jumper J1.

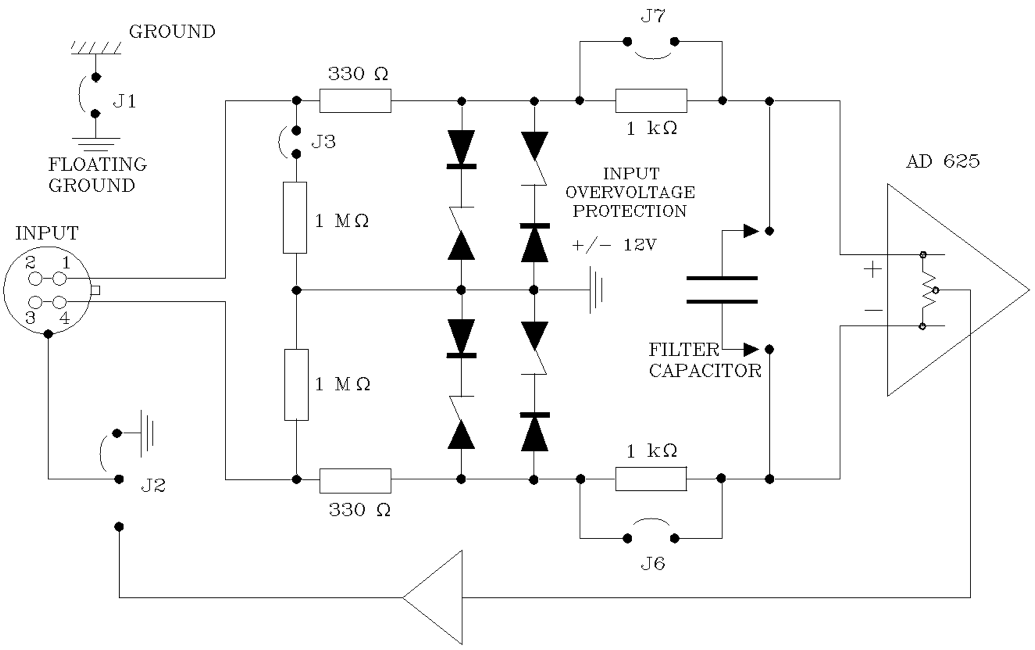
The screening of the cable connecting the coil to the instrument can be connected directly to analog ground by placing jumper J2 between pins 2 and 3. However the screen can also be "set" at a voltage which is half way between the two input voltages. This is achieved by placing jumper J2 between pins 1 and 2. **In this case, in order to avoid a short circuit it is essential that the coil connections are left floating and are not connected to**

the shield of the cable, earth or the ground of the magnet etc. (See figure 6)

If jumper J3 is in place (factory default) the input is balanced with an input impedance of 2MΩ. If J3 is removed then the input is unbalanced but with an input impedance of 1000MΩ.

An optional low pass RC filter may be inserted before the input amplifier by removing jumpers J6 and J7 and by connecting a capacitor between pads TP8 and TP9. The value of the capacitor can be calculated on the basis of an R value of 2660Ω. (J6 and J7 short circuit the resistors of the RC filter).

Note: J1,J2 and J3 are located on the solder side of the circuit board. The shield board must be removed to access these jumpers.



INTEGRATOR INPUT CONFIGURATION

5.2 5150 Integrator Module Digital Part

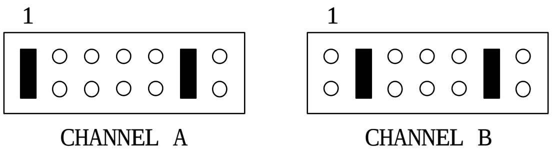
A TTL signal can be connected to the coaxial Lemo connector marked **EXT. TRIG**. The jumper J1 selects the active edge for the external trigger signal. This input is not debounced so the input signal must be clean.

J1 on 1-2: Trigger on rising edge (40KΩ pull down resistor)

J1 on 3-4: Trigger on falling edge (40KΩ pull up resistor)

The time between two active trigger edges must not be less than 1ms.

Jumper J2 is used to select the address of the module either as channel A or as channel B.



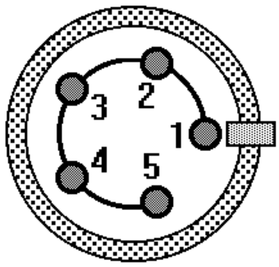
Any other jumper combination is invalid.

In section 3.1, it was stated that 5150 Integrator module is assigned to Channel A or B by positioning the internal jumper. If the PDI 5025 is delivered with only one 5150 Integrator module, then it is factory set to channel A. If the instrument is delivered with two 5150 Integrator modules they are assigned to channels A and B respectively. This assignment is independent of their location in the chassis. The effective assignment can be seen just after "Power on" on the GAIN display, following the autotest.

If necessary the user may change the assignment of each 5150 Integrator module by repositioning the internal jumper J2 (See section 10.2). The other jumpers must not be moved as this may make the module inactive (**the red jumpers must not be moved**, only the blue ones are user configurable).

5.3 5140 Trigger Module - Encoder Connections

The 5 pin Lemo connector marked **ENCODER** is used to connect an incremental encoder to the module. The pinouts are shown below:



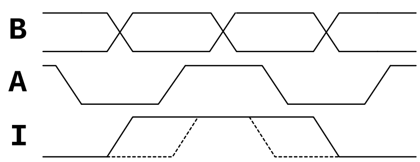
Front view

- pin 1: ground
- pin 2: channel A, TTL or open collector
- pin 3: channel B, TTL or open collector
- pin 4: index I, TTL or open collector
- pin 5: +5V, 500mA output (not limited!)
- shield: ground

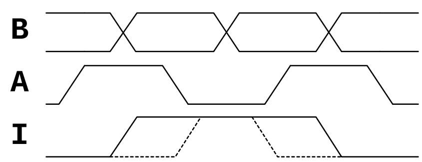
Warning: the current consumption is not limited. Care has to be taken to avoid over-current and short circuit conditions.

Channel A must be, by definition, that which is in phase with the index (I) pulse (see below).

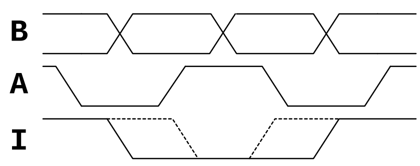
Channels A and B are dephased by ± 90° depending on the sense in which the encoder is rotating. The index pulse supplied to channel I fixes the reference point for the movements which follow. Jumpers J1 and J2 are used to normalise the signals of most common encoders found on the market. The timing diagrams below give the positions of these jumpers for the four possible configurations of the three input signals.



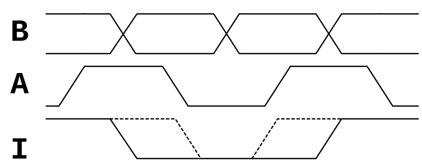
J1 on 1-2 ; J2 on 1-2



J1 on 2-3 ; J2 on 1-2



J1 on 1-2 ; J2 on 2-3



J1 on 2-3 ; J2 on 2-3

If an index signal is not used then pin 4 of the Lemo connector **must** be left open and jumper J2 placed on pins 2-3. The position of jumper J1 is not important.

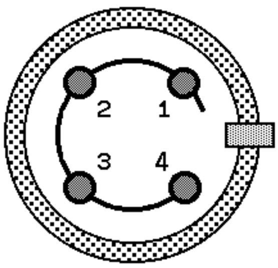
5.4 5140 Trigger Module - External Synchronisation

The SYNC input allows measurement sequences to be started by an external TTL or open collector signal. This input is "debounced" so that the signal can be generated by a mechanical switch. The jumper J4 is used to select the active edge of the synchronisation pulse:

J4 on 1-2	Synchronisation on falling edge (40kΩ pull up resistor)
J4 on 2-3	Synchronisation on rising edge (40kΩ pull down resistor)

5.5 5140 Trigger Module - Motor

The 4 pin Lemo connector marked **MOTOR** is used to power an optional DC motor or to control an external power supply which in turn drives the motor.



Front view

- pin 1: TTL + out; Motor command + in
- pin 2: TTL - out; Motor command - in
- pins 3 and 4: ±12V or ±24V between pins 3 and 4, limited up to 500mA
- shield: ground

If the motor is connected to pins 3 and 4, connecting pin 1 to ground will drive the motor in the positive sense while connecting pin 2 to ground will drive the motor in the negative sense.

Warning: To be coherent with the rotational sense of the encoder and thus allow the motor to be driven automatically by the PDI 5025 software the user must connect the motor so that the forward (FW) LED is lit when the motor is driven in the positive sense (command: MOT,+ or Lemo connector pin 1 connected to GND).

Jumper J7 selects the operating voltage for the motor. With jumper 7 on pins 2-3 the motor is driven at ±12V. If it is placed on pins 1-2 the operating voltage for the motor is ±24V. For the position of jumper 7 see the layout diagram in section 10.3.

5.6 5130 Interface Module

The 5130 module is used to interface the PDI 5025 to the host computer. Two digital interfaces are supported, RS 232 C and IEEE 488. The choice of the interface and its parameters are set by microswitches on the front of this module. The handshake mode for the RS 232 C interface is defined by 3 jumpers situated on the 5130 circuit board.

The PDI 5025 is equipped with a "watchdog" which automatically resets the internal microprocessor in the case of an eventual software crash. The "watchdog" can be disabled by removing jumper J4 on the 5130 circuit board.

5.7 5130 Interface Module IEEE 488

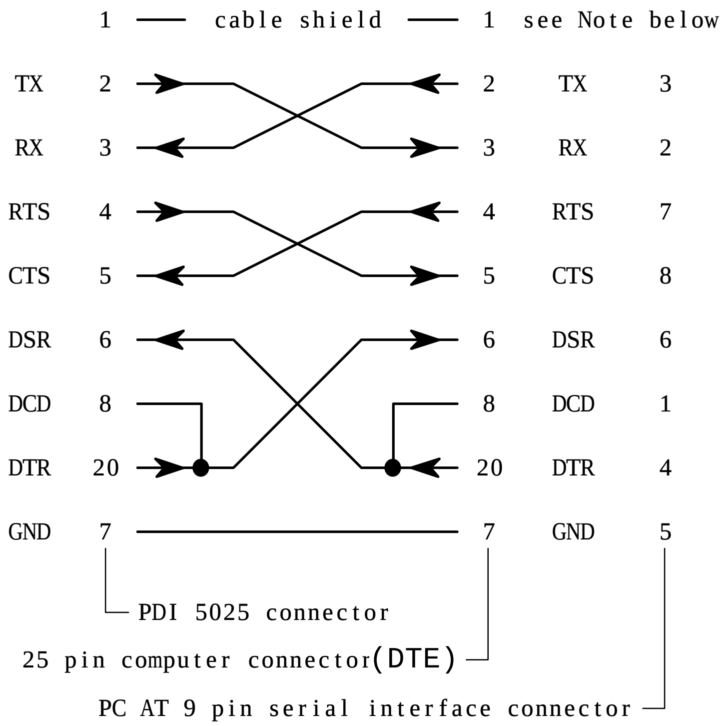
A standard IEEE 488 cable is used to connect the instrument to the host computer. The table below shows the functions assigned to each microswitch.

SW	"1"	"0"
1	DEVICE ADDRESS	
2		
3		
4		
5		
6	Always "0"	
7	Always "0"	
8	"CR/LF" Not sent	"CR/LF" Sent
9	IEEE 488	RS 232 c
10	Talker Only	Talker Listener

When microswitch 8 is set to "0" the carriage return ("CR") and line feed ("LF") characters are sent to the host at the end of each message. The EOI signal is made active at the end of every message. The "Talker only" microswitch sets the PDI 5025 in "Autonomous/Talker only mode" (see section 6.6.4).

5.8 5130 Interface Module RS 232 C

The PDI 5025 is a DCE device. The serial connection is a standard sub-D 25 pin male connector. We recommend the following cable layout to connect to the host computer. The pinouts are also given for an IBM PC or compatible with a 9 pin connector.



Note: Pin 1 of the RS 232 C connector on the Interface module supplies +10.5V at 100mA for the PSION Organiser handheld computer (modified by METROLAB). When connected to ground this power supply is automatically disabled.

The table below shows the functions assigned to each microswitch for the RS 232 C interface.

SW	"1"	"0"
1	BAUD	
2	RATE	
3	SELECTION	
4	8 bits	7 bits
5	2 stop bits	1 stop bit
6	Parity	No Parity
7	Even	Odd
8	XON/XOFF	No XON/XOFF
9	IEEE-488	RS 232 c
10	Autonomous	Conversational

The baud rate definition for microswitches 1,2 and 3 are given below.

Switches			BAUD RATE	
3	2	1		
0	0	0	300	
0	0	1	600	
0	1	0	1200	
0	1	1	2000	
1	0	0	2400	
1	0	1	4800	
1	1	0	9600	limited use*
1	1	1	19200	

*** Warning**: Limited use of 9600 and 19200 baud. When using RS 232 C interface at these speeds the PDI 5025 may lose some of the characters transmitted by the host computer during a **TRI** sequence. In this case the instrument will not recognise the command and it must be repeated by the host computer. The problem which may occur is when the host requests the instrument's status but the instrument fails to understand the request due to a lost character. A situation of "deadly embrace" will be reached as the PDI is waiting for the host and the host is waiting for the reply from the PDI. Care must be taken when designing the host software to retransmit a message to the PDI 5025 if it does not reply within a few seconds.

The "Talker only" microswitch sets the PDI 5025 in "Autonomous/Talker only mode" (see section 6.6.4).

6. INSTRUMENT SETTINGS

6.1 Power On Sequence

The PDI 5025 may be powered from any alternating mains supply giving between 85 and 264V, with a line frequency from 45 to 440Hz. The mains input socket is to be found at the rear of the chassis.

When powered on the first operation performed by the PDI 5025 is an autotest which takes 5 seconds. During the autotest sequence 5 different voltages are integrated each for 1 second. These voltages are given below:

Voltage	Gain
0	10
+0.494	10
-0.494	10
+0.494	5
-0.494	5

The results of the above measurements are verified as well as the correct operation of the over range indicators. Status bytes 5 and 6 indicate any error conditions that have been detected. The results of the autotest made be accessed via the **AUT** command. During the autotest the word "Test" is displayed on the 5150 module. At the end of the test the 5150 will either show a gain of 10 with the channel indicator ("A" or "B") or if an error has been detected the word "Fail" will be displayed. During the autotest the PDI 5025 does not process any communications with the host. Thus any messages sent will be ignored.

6.2 Channel Selection

The PDI 5025 can be equipped with a second channel (the standard unit is with a single channel). These two channels are always triggered in parallel. Measurements can be performed with either channel A, channel B or both channels simultaneously. The **CHA** command selects the active channel or channels.

A Command Error (bit 5 of status 1 set to 1) will be generated if the PDI 5025 is equipped with only one channel. In other words, this command is valid only for PDI 5025 with two channels.

6.3 Gain Selection and FNC Button

The gain of each channel can be independently set to any of the following values:

-- 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000

It is selected either by use of the GAIN button on the front panel or by the host computer with the **SGA** (**S**et **GA**in) command:

example:	SGA,B,100	set gain to 100 on channel B
	SGA,20	set gain to 20 on the active channel(s).

The Gain button is used to increment or decrement the Gain value. Press the button once to increment the gain. Press twice in rapid succession to decrement the gain. If the button is held down the gain will increment continuously until the button is released. We recommend that the gain is set in such a manner that the voltage displayed on the bargraph display is the highest possible without an over range being detected during the measurement cycle.

An over range condition is indicated by the OVR+ or OVR- LEDs on the bargraph display.

The **FNC** button performs several functions that are listed below:

- starts a measurement sequence
- if one over range indicator is on, the first depression of the button clears the over range indicator, the second will start the measurement sequence
- the button may also be programmed to generate an SRQ to the host computer (IEEE 488 interface).

When two channels are active the FNC buttons are connected in parallel.

6.4 Offset adjustment.

The user has the possibility to reduce the linear drift of the integrator by adjusting the voltage offset at the input. The voltage offset depends greatly of the Gain value. Therefore, the offset should be adjusted each time a new Gain is set. The OFFSET ADJUSTMENT mode facilitates this operation. This mode puts the integrator into an "infinite trigger sequence" using a 300 ms continuous integration period with the values being shown on the front panel display. The values are displayed in units of resolution of the integrator for the selected gain and VFC (for gain 1 to 100) or twice this resolution (for gain 200 to 1000). Therefore, for the 500 kHz VCO, the displayed unit is :

Gain	Units*10^{-8} Vs
1	2000
2	1000
5	400
10	200
20	100
50	40
100	20
200	20
500	8
1000	4

The OFFSET ADJUSTMENT mode is selected either by simultaneously depressing both buttons (GAIN and FNC) or by the host computer sending the ADJ command. Only one channel at a time can be set in the OFFSET ADJUST mode . The other channel, if present, is disabled and cannot be accessed. The selected channel does not respond to such commands as SGA and BRK. To leave the OFFSET ADJUST mode either push one of the two buttons or use the ADJ command from the host computer. The integrator then returns to its previous state. **Warning: in the case of an encoder with an index, the absolute position of the index is lost and has to be reinitialized with the IND command**.

6.5 Integrator Mode

The Integrator-Timer mode (**TRS,T**), which is the default at power-on, is generally used with fixed coil configurations to analyse time dependent flux variations. The Integrator-Displacement (encoder) mode (**TRS,E**) is used with moving coil configurations to analyse space dependant flux variations. The time interval between triggers can be as short as 1ms.

In this mode, channels A and B can be selected or deselected individually by **CHA,x** command (details of the syntax are given in appendix I).

The trigger source must be selected by the **TRS,...** (trigger source) command if the default value TRS,T is not desired.

Finally, a **TRI,..** sequence must be programmed, defining a sequence of integration intervals.

After the execution of the **TRI,..** command, the **RUN** command starts the measuring sequence. The **TRI,..** sequence is memorized in the PDI 5025 and can be repeated as many time as required by the **RUN** command.

If needed, the parameters of the current **TRI,...** sequence may be returned to the controller by issuing a **TRI,?** command.

6.6 Programming The Integration Intervals

The **TRI,...** is used to define the integration intervals which are delimited by "Trigger" events. This command is explained below:

$TRI,s,a/n_1,C_1/...../n_i,C_i{}^{CR}{}_{LF}$

where:

s = + or -	is the sense of the movement (default +)
a = value	absolute position of the first TRIGGER (default 0)
n_i = 1 to 65535 or *	number of intervals of C_i pulses or counts. '*'means infinite number of intervals.
C_i = 1 to 2^{23}	number of encoder pulses or time counts between TRIGGERs (always positive)
i = 1 to 20	number of different measurement parameters pairs.

An infinite number of integration intervals can be selected in the TRI sequence (n_i=*). This means that once the run is started it will continue until a break command (**BRK**) or a buffer full condition occurs. Any length of integration period (within the limits 1 - 2^{23} pulses or ms) can be associated with an infinite sequence. Any finite sequence can precede an infinite sequence in the TRI command, but the infinite sequence has to be the last one of the series. If the reading of the measurement results is performed faster than triggering, the buffer will never get full and the run will last for ever. If the LAST CUMULATED STORAGE mode (CUM,1,L) is selected, the PDI 5025 works like a fluxmeter.

To demonstrate the use of the **TRI** command some examples are given below:

$TRI,-,+500/4,15/10,2/1,100^{CR}{}_{LF}$	
$TRI,,/5,200^{CR}{}_{LF}$	which is equivalent to $TRI,+,0/5,200^{CR}{}_{LF}$
$TRI,,50^{CR}{}_{LF}$	this command will move to an absolute position "a" without performing any measurement.
$TRI,,89/10,250/*,1000^{CR}{}_{LF}$	start at the absolute position of 89 then perform 10 sequences of 250 pulses then an infinite sequence of 1000 pulses.

6.6.1 Trigger Source: Timer

The PDI 5025 has an internal quartz controlled time base generator set at 1 kHz which feeds the Trigger Source Counter. The **TRI** command defines the integration intervals in milliseconds (ms). The counter can be activated in two different ways:

a) by the **RUN** command or by pressing the FNC button. In the later case the TRS command (which must have been issued before) must be **TRS,T**.

b) by the first external synchronisation signal on the SYNC input which follows the **RUN** command issued by the host or by pressing the FNC button. All following signals on the SYNC input are ignored. In this case the **TRS** command must be **TRS,T,S**

Note: the synchronisation precision is better than 0.5µs.

6.6.2 Trigger Source: Encoder

The 5140 module has been designed to accept the signals generated by a linear or rotational incremental encoder. These signals are squared and filtered before being fed to the Trigger Source Counter with 4 pulses representing 1 encoder cycle (multiplication by 4). The maximum counting frequency is 10^6 cycles/s.

The **TRS,E,..** command will depend on the type of encoder used.

6.6.2.1 Rotational Encoder with Index

TRS,E,dddd

where dddd is the number of cycles per rotation (expressed as a decimal number). Note for an encoder with 360 cycles (TRS,E,360) actually gives 1440 pulses per rotation.

The use of the index pulse allows the measurements to be made absolute in relation to a fixed point. The index point must be passed at least once to initialise the measurement sequence. This may be done manually or by using the **IND,s** (where s = + or -) which will turn the motor in the indicated sense until the index pulse is detected. From this point on all measurements are absolute and are verified on each successive index pulse.

It follows from the above that the "a" values in the **TRI,...** command can not be greater than the number of pulses per encoder rotation, nor can they be negative. The value of "C_i" is also limited to the number of pulses per encoder rotation.

The command **MOT,A** starts the motor in the sense defined by the 's' in the **TRI,...** command immediately following the reception of the **RUN** command and stops the motor at the end of the measurement sequence.

Programming example of PDI 5025 with encoder and index:

TRS,E,360	Indicate a disk encoder with 360 cycles per rotation with an index.
IND,-	Turn the motor in the backwards sense (BW) until the index pulse is detected.
TRI,-,140/8,10/3,1000	prepare a sequence of 11 measurements in the BW sense, the first TRIGGER is at absolute position 140 (pulses, not cycles).
MOT,A	Set motor to automatic
RUN	Start the motor, perform the measurements then stop the motor.

6.6.2.2 Linear Encoder with Index

TRS,E,S

This case is identical to the one above except for the following points:

the absolute position can range from -2^{23} to $+2^{23}$ either side of the index point

The value of "**a**" in the TRI,... can be negative

6.6.2.3 Linear or Rotational Encoder without Index but with External Synchronisation.

TRS,E,S

The external synchronisation signal must be connected to the SYNC input. It is used to define the "zero" position of the encoder when it first occurs after the **IND,s** command. If **S** is specified the motor is moved until the first SYNC pulse is received. If **s** is not specified the instrument waits at the present position for the SYNC pulse.

After this the PDI 5025 behaves in the same way as the linear encoder with an index.

Warning: the Index input of the encoder connector must not be wired and left open.

6.6.2.4 Linear or Disk Encoder without Index or External Synchronisation.

TRS,E

The TSR counter is cleared by the RUN command. Warning: the sign of **a** must be the same as that of **s** in the TRI... command.

6.6.3 Trigger Source: External

The PDI 5025 can also accept external "TRIGGER" signals. In this case the **TRI,...** command does not serve any purpose as the "trigger" pulses are not generated by the 5140 module.

The external trigger source (the active edge of which can be set by jumper) must be connected to the **TRIG** input of the 5150 module.

Warning

- If two channels are used care must be taken to connect the trigger signal to both channels (i.e. in parallel).
- Trigger signals must not occur during the autotest as this will cause the autotest to fail.

The **TRS,X** command must be executed to enable the external trigger.

Following the reception of the **RUN** command each trigger signal defines the start of a new integration period. The measurement sequence is terminated by the **BRK** command, after which the external trigger signals are ignored. The bit 3 of Status 1 (End of Run) is set to "1".

During a measurement cycle the external trigger can be disabled by the command **FPT,0**. To reenable the external trigger the command **FPT,1** must by executed.

Note: the **FPT** command does not terminate the measurement that is in progress as does the **BRK** command.

In the same way as the Timer mode a measurement sequence can be initiated by an external synchronisation signal. To select this mode use the **TRS,X,S**. This works in manner described above except that the trigger signals are disabled until the first external synchronisation pulse occurs (this is fed to the **SYNC** input of the 5140 module) that follows the **RUN** command. Any following external SYNC signals are ignored.

6.6.4 Fluxmeter - Autonomous/Talker only mode

In the "Fluxmeter - Autonomous/Talker only" mode the PDI 5025 performs a continuous measurement with a defined period, at the end of which a cumulated flux measurement is available. This value is updated continuously. This mode can be used to get relative magnetic field measurements.

In this mode, pressing the FNC button sets the cumulated measurement to zero.

An over range condition causes the PDI 5025 to stop its measurement cycle. To resume the measurements, the RUN command must be issued by the host computer or the FNC button must be pressed twice (the first time to clear the over range error condition and the second time to restart the measurements).

Note: if the command **NBO,1** has been sent, the occurrence of an overrange will not stop the measurement.

This mode can be used either with a fixed coil to measure time dependent flux variations, or by a moving coil going from one field region to another one in order to perform field measurements or comparisons.

The Fluxmeter mode (which is a particular case of the Integrator mode) is selected with the following commands :

TRS,T

TRI,+,0/*,300

CUM,1,L

RUN

The integration period can be chosen to be any value (except when limited by the CUM,1,L command, see section 7.2.3). The Fluxmeter mode is terminated by sending a BRK command from the host computer.

The Autonomous/Talker only mode is selected using the microswitch 10 on the 5130 module. This mode is equivalent to a Fluxmeter mode with a 300 ms integration period but the last processed value is sent by the PDI 5025 as soon as it is available. It must be noted that in this mode the PDI 5025 cannot receive commands from the host computer. To stop the Autonomous/Talker only mode microswitch 10 must be set to position 0.

7. PROGRAMMING THE PDI 5025

This chapter describes the way in which the PDI 5025 stores and transmits the measured data. The internal status registers are also defined. The differences between the two host computer interfaces are given.

7.1 The PDI 5025 Status Registers

The PDI 5025 has seven internal status registers that can be accessed by the user. These registers can be used to interrogate the instrument on its current state. Each status register contains eight bits and can be read by the host computer in two different formats either in hexadecimal or binary. The registers are numbered 1 to 7

To read a status register the host computer must either of the commands shown below:

STH,n where n (n=1 to 7) is the number associated with the status register, if n is omitted then the value of 1 is assumed. This command generates a two character reply representing the value of the status register in **hexadecimal**.

STB,n where n (n=1 to 7) is the number associated with the status register, if n is omitted then the value of 1 is assumed. This command generates an eight character reply (0 or 1) representing the value of the status register in **binary**. The first character in the string corresponds to the most significant bit of the status register and the eight character to the least significant bit.

The detail of the seven status registers is given in the following sections.

7.1.1 STATUS 1: Measurement Status (1 byte)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
STATUS 2 FLAG	Always 0	Command Error	Overrang e Error	End of Run	Data Ready	Trigger	Synchro

This status byte is cleared when it has been read except for bit 7 which is only cleared once STATUS 2 has been read.

bit 7: STATUS 2 flag

This bit is set to 1 to indicate that STATUS 2 is not zero and that it should be read.

bit 6

This bit is always set to 0.

bit 5: command error

This bit is set to 1 each time that a command sent by the host is not syntactically correct, either because the mnemonic is unknown or because one or more values are incorrect. This error is also generated when a command is issued at a time when it is out of context.

bit 4: overrange error

This bit is set to 1 each time that the input voltage, of channel A or B, reaches the positive or negative overrange threshold.

Note: an overrange error has no effects before the first trigger (when waiting for a Synchro. signal or the absolute value of the first trigger for example). After that an overrange condition aborts the measurement cycle immediately and corresponds to the receipt of a **BRK** command (see section 7.2.7). The run abort can be disabled with the command **NBO**. Status register 4 indicates which overange has occurred.

bit 3: End of RUN

This bit is set to 1 at the end of each measurement sequence, or after the receipt of a **BRK** command when the Trigger Source is External. It does not mean that the conversion of all the measured data has been completed.

bit 2: Data Ready

The precise instant at which this bit is set depends on the transmission mode selected by the **IMD** command:

a) In the continuous transmission mode (**IMD,1**) this bit is set each time a value is available. When STATUS 1 is read this bit is cleared but is immediately set to 1 if there is still data in the buffer.

b) In the block transmission mode (**IMD,0**) this bit is set to 1 as soon as **all** the measured data have been converted and are thus available to the host.

bit 1: Trigger

This bit is set to 1 on receipt of the trigger signal occurring during a measurement cycle. The trigger can be internally generated by the 5140 module or external.

bit 0: Synchro

This bit is set to 1 each time a signal is detected on the SYNC input (of the 5140 module) or on the encoder index input. This can be used to synchronise the host computer software with the pulse from the index of an encoder.

7.1.2 STATUS 2: Error Status (1 byte)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
0	0	FNC button	Power ON or Reset	Autotest failed	Encoder Count Error	Data Buffer Full	Overlap Timeout

This status byte is cleared once it has been read, this also clears bit 7 of STATUS 1.

bit 7:

Always 0.

bit 6:

Always 0.

bit 5: FNC Button

The FNC button of the 5150 module can be programmed (by the **FNC,0** command) to set this bit to 1 when it is pressed. If the FNC button has been programmed to start a RUN (**FNC,1**) this bit is inactive.

bit 4: Power on Reset

This bit is set at power on or by the return of power after a power failure. It is also set by the reset provoked by the watchdog following a software crash.

bit 3: Autotest Failed

This bit is set to 1 if the autotest detects an error in the instrument. The exact cause of the error can be found by examining status registers 5 and 6.

bit 2: Encoder Count Error

This bit is only valid if the PDI 5025 is used with an encoder encorporating an index. Each time the index is detected the instrument verifies if the pulses which are counted are correct. If an error is found this bit is set.

bit 1: Measurement Buffer Full

The PDI 5025 has a memory buffer that can hold 5200 values. When the buffer is full this bit set to indicate this error.

Note: If this error happens the measurement cycle is aborted (as if a **BRK** had been received see section 7.2.7).

bit 0: Overlap Timeout

The PDI 5025 has been designed to receive a trigger once every millisecond. However if an external trigger is used, or if the encoder turns too quickly, and the interval between triggers is less than 1 millisecond this bit will set to indicate the problem.

Note: If this error happens the measurement cycle is aborted (as if a **BRK** had been received see section 7.2.7).

7.1.3 STATUS 3: Trigger Source Module 5140 (1 byte)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
<- T R S mode ->			Infinite Sequence	Run Active	Forwards or Backwards	Motor +	Motor -

This status byte represents the state of the PDI 5025 at the time of reading.

bit 7, 6, 5: TRS mode

These three bits indicate which trigger source has been selected (**TRS** command).

bits: 7	6	5	Mode	
0	0	0	Offset adjustment	ADJ,1
0	0	1	Timer without Synchro	TRS,T
0	1	0	Timer with Synchro	TRS,T,S
0	1	1	Encoder without index or Synchro	TRS,E
1	0	0	Encoder with Ext. Synchro or Linear with Index	TRS,E,S
1	0	1	Rotating Encoder with index	TRS,E,dddd
1	1	0	External Trigger	TRS,X
1	1	1	External Trigger with Synchro	TRS,X,S

bit 4: Infinite Sequence

This bit is set to 1 to indicate that an infinite trigger sequence is in progress. An infinite sequence is set using n_i=* in the TRI command.

bit 3: Measurement in progress (RUN Active)

This bit is set to 1 to indicate that the PDI 5025 is in the process of measuring. The bit is set by the **RUN** command or by pressing the FNC button; it is cleared when the sequence is terminated.

bit 2: Forward/Backward

A 1 indicates motion in the FORWARDS sense (FW LED lit), a 0 refers to the BACKWARDS sense (BW LED lit).

Note that in the timer mode this bit is always set to 1 (FW lit).

bit 1: Motor +

This bit indicates that the motor has been activated in the positive sense (corresponds to FW if cabling is correct), see the **MOT,+** command for more details.

bit 0: Motor -

This bit indicates that the motor has been activated in the negative sense (corresponds to BW if cabling is correct), see the **MOT,-** command for more details.

7.1.4 STATUS 4: PDI module 5130 (1 byte)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
<- V F C type ->		Overrange -	Overrange +	<- V F C type ->		Overrange -	Overrange +
Channel B				Channel A			

This status byte is divided into two identical parts, bits 0 to 3 refer to Channel A and bits 4 to 7 to Channel B.

bit 7, 6 and 3, 2: VFC type

These bits show if the channel is active and the type of Voltage To Frequency Converter which has been used in the module.

bits: 7/3	6/2	
0	0	Channel Inactive
0	1	VFC 100 kHz
1	0	VFC 500 kHz
1	1	VFC 1 MHz

bit 5 and 1: Overrange -

This bit indicates the state of the negative overrange LED on the bargraph display. When the negative overrange is lit this bit is set to 1.

bit 4 and 0: Overrange +

This bit indicates the state of the positive overrange LED on the bargraph display. When the positive overrange is lit this bit is set to 1.

7.1.5 STATUS 5 and 6: Autotest (2 bytes)

STATUS 5 relates to Channel A while STATUS 6 refers to Channel B. If only one 5130 module is placed in the instrument the status bits of the missing channel are always zero.

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
0	0	0	Error in Analog Measure	OVR - Not ON ->Error	OVR + Not ON ->Error	OVR - Not OFF ->Error	OVR + Not OFF ->Error

These bits are set during the autotest.

bit 7,6,5: Unused.

Always 0.

bit 4: Error in the analog measurement.

This bit is set to 1 when one or more values measured during the autotest are out of tolerance. The **AUT** command allows the host computer to access the values measured by the autotest (see section 7.2.9).

bit 3: Negative Overrange not ON Error

This bit shows that the deliberate negative overrange was not detected during the autotest. This is tested with an input voltage of -0.494V and a gain of 10.

bit 2: Positive Overrange not ON Error

This bit shows that the deliberate positive overrange was not detected during the autotest. This is tested with an input voltage of 0.494V and a gain of 10.

bit 1: Negative Overrange not OFF Error

This bit is the opposite of bit 3. If it is set to 1 the overrange condition was detected when it was not present. This is first tested with an input voltage of 0V and a gain of 10, then with -0.494V and a gain of 5.

bit 0: Positive Overrange not OFF Error

This bit is the opposite of bit 1. If it is set to 1 the overrange condition was detected when it was not present. This is first tested with an input voltage of 0V and a gain of 10, then with +0.494V and a gain of 5.

7.1.6 STATUS 7: Acquisition status

This status represents the state of the PDI 5025 at the time of reading.

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
0	0	0	Infinite Sequence Active	Run Active	Data Transfer Mode	Storage Mode	

bits 7,6,5: Unused

Always zero.

bit 4: Infinite Sequence Active

This bit is set to 1 when an infinite trigger sequence is in progress. An infinite sequence is selected with n_i = * in the TRI command. This bit is similar to bit 4 of Status 3.

bit 3: Run Active

This bit is set to 1 to indicate that the PDI 5025 is taking measurements. This status bit is similar to bit 3 of Status 3.

bit 2: Data Transfer Mode.

This bit is set to 1 to indicate that the immediate data transfer mode (IMD,1) is selected.

A 0 indicates that the block transfer mode (IMD,0) is selected.

bits 1&0: storage mode.

These bits indicate the selected data storage mode (command CUM).

bits: 1	0	Mode
0	0	Individual values
0	1	Cumulated values
1	0	Only last cumulated value is available.

7.2 Data Acquisition

During a measurement cycle the PDI 5025 stores raw data in an internal buffer. This raw data is then converted by the local microprocessor and stored. Following this operation, data are available to the host computer. There are three types of storage for converted data: individual, cumulated and last cumulated. The CUM command selects the storage mode. The processed data may be read in two different ways, either one after another or all in one block. The IMD command selects the data transfer mode.

7.2.1 Individual data storage (CUM,0)

This is the default data storage mode. Measured data are stored separately. The data represent the integrated value during a period delimited by two triggers. The number of values equals the number of integration periods.

Bits 1 and 0 of Status 7 are set to 0.

7.2.2 Cumulated data storage (CUM,1,S)

The value cumulated from the start of the measurement is stored in memory at the end of each integration interval. If the command **NBO,1** has been sent previously to the PDI 5025, the cumulated value is cleared to 0(zero) at each occurrence of an overrange. At the end of a run, the number of available data is identical to the number of integration periods.

Bits 1 and 0 of Status 7 are set respectively to 0 and 1.

7.2.3 Last cumulated data storage (CUM,1,L)

This mode stores only the last cumulated value. That means when a reading is performed, the integrated value from the beginning of the run until the last trigger preceding the reading time is transferred. Pressing the FNC button resets the cumulated value.

Since the local processor needs time to perform calculations on raw data, the trigger rate must not be too fast. If this is the case the last cumulated available value is not the last measured value. When this happens, the '*' character appears in the read string between the last digit and the channel descriptor (example 12546*A).

This mode is incompatible with the block data transfer (**IMD,0**) and forces an immediate data transfer mode (**IMD,1**)

This mode, with an adequate TRI sequence, allows the user to use the PDI 5025 as a fluxmeter.

7.2.4 Direct Data Transfer (IMD,1)

This is the default data transfer mode. Measured data are read individually. As soon as the measured data has been processed the DATA-READY bit in STATUS 1 is set to 1.

This happens even if the measurement cycle has not been completed. Thus it is possible to collect data as the measurement proceeds and so free up space in the internal data buffer. The total number of measured values can therefore be greater that 5200.

To read the next value the host computer must send the **ENQ** command if connected via the RS 232 C interface or address the PDI 5025 as a Talker when using the IEEE-488 bus. The returned value is expressed in 10^{-8}Vs followed by a space and then the channel identifier (e.g. 45982900 A). If an overrange occur during the measurement with the command **NBO,1** previously sent to the PDI 5025, the returned value is 0 with an exclamation mark between the last digit and the channel descriptor (e.g. **0!A**). To read the next value the host must send the **ENQ** or address the instrument as a Talker. The values are output in the same order as they are measured. If two channels are active then Channel B is sent before Channel A. If the values are read during a measurement cycle and the ENQ command (or the Talker state) is performed before the next data value is available (i.e. DATA-READY not yet set to 1) then the instrument returns the characters carriage return and line feed (CR LF) which corresponds to a null string.

When all the values have been read by the host computer and the internal buffer is empty the instrument will reply to a request for more data with the End Of Data string which by default is set to Ctrl Z (ASCII 26). This string may be programmed by the user, see section 7.2.6.

7.2.5 Block Data Transfer (IMD,0)

This mode allows the internal data buffer to be emptied in one single block transfer. Therefore it is not possible to obtain data whilst the measurement is in progress, the host computer must wait until all the data have been taken and converted. As soon as the last value has been converted the DATA-READY bit is set to 1 (bit 2 STATUS 1)

To initiate the block transfer the host computer must send the **ENQ** command for the RS 232 C interface or address the instrument as a Talker for the IEEE-488 bus. The PDI 5025 will then transmit all the data in the order that it was measured. Each value is separated by the CR LF pair of characters. If two channels are active the value for Channel B is sent before the value for Channel A.

The End Of Data string (default Ctrl Z) is emitted immediately after the last data value. The End Of Data string can be programmed with the **EOD** command (see section 7.2.6).

Note: If the External Trigger is used then the measurement cycle must by terminated by the **BRK** command which will end the cycle and allow the data to be read in the Block Transfer mode.

Note: this mode is incompatible with **CUM,1,L** and therefore cannot be executed in this case.

Warning: The Block Data Transfer is the most rapid way to transfer data form the PDI 5025 to the host computer but the user should verify that the host is capable of accepting the quantity of data to be transferred.

7.2.6 End of Data (EOD)

As mentioned in sections 7.2.4 and 7.2.5 the End of Data string, sent by the instrument when the buffer is empty and the host requests to read data, can be programmed by the host computer. By default the End of Data string is the character Ctrl Z (ASCII 26). The EOD command allows the End of Data string to be modified to contain one or more characters. The EOD command parameters are decimal ASCII codes separated by commas.

Example: following the reception of the command

EOD,69,109,112,116,121,13,10

the PDI 5025 will reply

Empty

The maximum length for the End of Data string is 20 characters. If the **EOD** command is executed without any parameters then the default string (Ctrl Z) is assumed.

7.2.7 Interrupting a measurement cycle (BRK)

It may be necessary to stop the measurement cycle once it has begun. This may be because of a programming error in the cycle itself or simply to restart a new cycle without waiting for the present one to end (the **RUN** command being ignored during the measurement cycle). To abort the measurement cycle the host should execute the **BRK** command.

This command is also used to stop measurements when external triggers are used (**TRS,X**). In this case the bit 3 of the Status 1 (End of Run) is set to "1".

The **BRK** command has the following effects:

- the measurement cycle is terminated
- the motor (if it is operational) is stopped irrespective of the commands used to start it (**MOT,+**, **MOT,-** or **MOT,A**)
- disables the external trigger input
- all the data measured up to the instant that the **BRK** was received are valid and can be read by the host.

The PDI 5025 can also execute a **BRK** itself if any of the following errors occurs:

- Overrange error (bit 4 STATUS 1)
- Overlap timeout (bit 0 STATUS 2)
- Measurement buffer full (bit 1 STATUS 2)

7.2.8 Automatic Use of a Motor (MOT,A)

As described in section 5.5, a DC motor can be connected to the 5140 module to control the movements of the measurement coil. Apart form the three commands **MOT,+**, **MOT,-** and **MOT,S**, which are used to start and stop the motor, there is also the possibility to control the motor automatically during a measurement cycle. Thus the motor starts when the **RUN** command is received (or the FNC button is pressed) and stops when the measurement cycle is complete.

If **MOT,A** is executed before the **RUN** command the motor will start after the **RUN**. If **MOT,A** is sent after a **RUN**, the motor will start immediately following the reception of the **MOT,A**. This function remains active and it is not necessary to repeat the **MOT,A** command before each measurement cycle. The function is cancelled by the the reception of the command **MOT,S**.

7.2.9 Autotest (TST and AUT)

Each time the PDI 5025 is powered up it performs an autotest, which lasts five seconds and performs five preset measurements and verifies the results. During the autotest the display shows the word "Test". Once finished, if the autotest was successful, the display will show the channel to which it is assigned and the current value of the gain. Should the test proves incorrect the display will show the word "Fail". The user may obtain detailed information on the problems encountered during the autotest by examining status registers STATUS 5 and STATUS 6 (see sections 7.1.5 and appendix IV).

The autotest may also be executed by the host computer by issuing a **TST** command.

Note: Following the autotest the PDI 5025 returns to its previous state. In case of an encoder with index, the index position is lost and has to be reinitialized with the IND,s command.

During the autotest the PDI 5025 does not respond to any host computer commands sent via the RS 232 C interface while IEEE 488 commands will be executed once the autotest has finished.

The command **AUT** allows the host computer to access the values measured by the autotest. The instrument transfers this data in one block in a format similar to that shown below:

Autotest Data:	Corresponding Input Voltage [V]	Gain
450 B	0.00	10
500 A	0.00	10
49420500 B	0.494	10
49399850 A	0.494	10
-49410150 B	-0.494	10
-49400200 A	-0.494	5
49410800 B	0.494	5
49399700 A	0.494	5
-49408050 B	-0.494	5
-49390300 A	-0.494	5

In the case of the IEEE 488 interface the above data in the left hand column (including the title) are not separated by a CR-LF but semicolons (;).

Warning: It is essential that during the autotest nothing is detected on or fed to the TRIG input of the 5150 modules.

7.2.10 Internal Short Circuit (ISC)

To adjust the offset of the 5150 module, it is possible to short circuit the measurement coil. In the PDI 5025 this can be achieved by executing the **ISC** command.

The format of this command is shown below:

ISC,c,n

where "c" is the channel A or B "n" is 0 to open circuit the coil or 1 to short circuit it.

"c" can be omitted, in this case the command effects all active channels.

Examples:

ISC,B,1	short circuit the input of channel B
ISC,0	open circuit the input on all active channels.

The short circuit is made by a semi-conductor analog switch and does not represent a real short circuit (due to the small residual resistance in the switch). This fact must be taken into consideration when working with very high precision measurements.

7.2.11 Synchronisation by the Host Computer (SYN)

As already mentioned in section 6.6 it is possible to synchronise a sequence of measurements with an external event. This can be either a signal supplied to the SYNC input of the 5140 module or it may be a command generated by the host computer. This command differs for the RS 232 C and IEEE 488 interfaces.

Note: The synchronisation delay for the SYNC input is very short (less than 500ns). Whereas the delay for a software SYNC depends on the transmission speed of the command and the software overhead for decoding the message. It may, therefore be in the order of several milliseconds.

7.2.11.1 RS 232 C

To perform the software synchronisation via the RS 232 C interface the host just sends the **SYN** command. Thus when the user has defined the source of the trigger with external synchronisation, (**TRS,T,S**, **TRS,E,S** or **TRS,X,S**) the instrument will, once it has received the **RUN** or **IND** commands, wait for the external signal on the SYNC input or the **SYN** command to start the measurement cycle.

7.2.11.2 IEEE 488

In this case the synchronisation command is the IEEE 488 Group Execute Trigger (GET). As this command can be sent to several devices on the bus it must be possible to enable and disable the synchronisation caused by the GET. The **SYN** command is thus used to enable or disable the GET, it is used as follows:

SYN,0	GET has no effect (disabled)
SYN,1	GET causes the synchronisation of the instrument.

The GET is disabled by default. Thus when the user has defined the source of the trigger with external synchronisation, (**TRS,T,S**, **TRS,E,S** or **TRS,X,S**) the instrument will, once it has received the **RUN** command, wait for the GET (the **SYN,1** having already been executed) or a signal on the SYNC input to start the measurement cycle.

7.3 The IEEE 488 Interface Commands

7.3.1 The "REMOTE" Command

The PDI 5025 is always in the REMOTE mode and does not therefore react to the REN line. Thus it is unnecessary to place the instrument in the REMOTE mode in order to send it messages.

7.3.2 The "LOCAL LOCKOUT" Command

The IEEE-488 command LOCAL LOCKOUT (LLO) disables the front panel buttons on the 5150 module, only the GTL (LOCAL) command can reactive these buttons.

7.3.3 The "GO TO LOCAL" Command

Normally the IEEE-488 Go To Local (GTL) command forces a device to return to its LOCAL state when it has been placed in the REMOTE state. As the PDI 5025 is always in the REMOTE state the GTL has no effect on the instrument other than to enable the front panel buttons that have been disabled by a previous LOCAL LOCKOUT (LLO) command.

7.3.4 The "GROUP EXECUTE TRIGGER" Command

The IEEE-488 Group Execute Trigger (GET) forces a synchronisation by the host computer. Before it can be used the GET must be enabled by the **SYN,1** command (see section 7.2.11.2).

7.3.5 The "INTERFACE CLEAR" Command

This command aborts the present communication and frees the bus. The instrument is unaddressed.

7.3.6 The "DEVICE CLEAR" Command

This command (DCL) resets the instrument to its power on state whether or not it is addressed.

Warning: As the instrument goes to its power on state it is completely reinitialised. During this time, about 2 seconds, it cannot process any communications! Thus following a **DCL** the host should wait 2 seconds before sending another command to the instrument.

7.3.7 Service Request (SRQ)

An important feature of the PDI 5025 is its ability to interrupt the host computer when certain conditions occur.

When a service request is sent, the controller must first determine which instrument instigated the demand. This operation is carried out by a serial poll (spoll) of each device, connected to the bus, capable of generating a Service Request. When the instrument is "polled" it replies with the contents of its SRQ status register which indicates if it is the originator of the Service Request and if so, the nature of the request. The above sequence supposes that the controller is programmed to receive the SRQ interrupts.

If one or more of the bits in the SRQ status register are set to 1 then the "Require Service" (RQS) bit will be set. This causes a request for service to the controller. However, as soon as the Require Service bit is set, the PDI 5025 puts the SRQ line of the IEEE bus true, which in turn causes an interrupt in the controller.

STATUS registers 1 and 2 seen through the SRQ masks are used to derive the SRQ status register.

7.3.8 Setting the SRQ Masks

The SRQ Masks can only be applied to bits 0 to 5 of STATUS 1 and STATUS 2. Each 1 in the SRQ Mask can be considered as a hole which allows the information in the STATUS 1 or 2 registers through to generate a 1 in the SRQ status register and thus generate a Service Request.

Example:

bit	7	6	5	4	3	2	1	0
STATUS 1	1	0	0	1	1	0	0	1
SRQ Mask Register			0	1	0	0	1	0
SRQ status register	1	1 RQS bit	0	1	0	0	0	0

To create a mask, firstly determine which conditions must generate interrupts. In this example the bits corresponding to the Trigger (bit 1) and the Overrange Error (bit 4) will be active. Next, determine the octal code containing two digits for these conditions. Finally, output the **MSK** message followed by the two octal digits (see also section 7.3.10).

If any of the bits from STATUS 2 once masked is non zero then bit 7 of the SRQ status register is set to 1, thus indicating that the interrupt is due to a condition in STATUS 2. The host computer must therefore read STATUS 2 to determine the exact cause of the interrupt.

When the value of the mask register is changed the SRQ status register is updated. Note that the RQS bit can only be cleared by a SPOLL.

7.3.9 The SRQ Status Register

A Service Request is generated by the PDI 5025 when bit 6 (RQS) is set to 1. This bit must be tested by the controller when it executes a serial poll. The remaining bits allow the controller to determine the nature of the Service Request, a 1 representing the active state. The SRQ status register is RESET to 0 after the controller has finished the serial poll. The bits have the following significance:

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
STATUS 2 FLAG	RQS bit	Command Error	Overrange Error	End of RUN	Data Ready	Trigger	Synchro

The SRQ status register is a copy of STATUS 1 but seen through the first mask register (Mask 1). These bits have the same meaning as those described in section 7.1.1 for STATUS 1. However bits 6 and 7 differ in the following manner.

<u>bit 7: STATUS 2 FLAG</u>

This bit is set to 1 to indicate that the masked version of STATUS 2 is not zero and that it should be read.

Example: If all the bits in STATUS 2 are masked then this bit will never be set irrespective of the bits in STATUS 2

<u>Unlike</u> STATUS 1 this bit (like all the others) is set to zero by reading the SRQ status register (ie by a serial poll).

<u>bit 6: RQS (Request Service bit)</u>

This bit is set to 1 if one or more bits of the SRQ status register are set to 1, to generate a Service Request.

7.3.10 SRQ Masks command (MSK)

The two mask registers are programmed by using the **MSK** command which has the following format:

MSK,x,no where

x is the number of the mask register (if omitted the default is 1)

n signifies the ASCII characters from 0 to 7 inclusive relating to the mask pattern, in octal, for bits 3,4 and 5.

o the ASCII characters from 0 to 7 inclusive relate to the mask pattern, in octal, for bits 0, 1 and 2.

This message allows the user to set up a mask for the SRQ register in order to prevent the PDI 5025 from interrupting the controller when a particular event occurs. On power up the default value for Mask 1 is 00 (octal) and 20 (octal) for Mask 2.

SRQ Mask register:

	bits 5	4	3		2	1	0	
"n" value				"o" value				
0	M	M	M	0	M	M	M	
1	M	M	A	1	M	M	A	
2	M	A	M	2	M	A	M	
3	M	A	A	3	M	A	A	M= masked
4	A	M	M	4	A	M	M	A= active
5	A	M	A	5	A	M	A	
6	A	A	M	6	A	A	M	
7	A	A	A	7	A	A	A	

Note that it is not possible to mask bits 6 and 7. Both bits are always active.

Example:

MSK,1,00 gives all bits masked on STATUS 1

MSK,2,77 gives all bits active on STATUS 2.

Note: See section 7.3.7 and 7.3.9 for more details on the SRQ register.

8. APPENDIX I. PDI 5025 COMMANDS

All the PDI 5025 commands are listed in tabular form in this section in their order of use. That is initialisation, measurement, data extraction and then miscellaneous. This does not, however, signify that they must be executed in this order!

SYNTAX	DEFAULT	DESCRIPTION	MNEMONIC
CHA,A	***	Select channel A.	CHAnnel
CHA,B		Select channel B.	
CHA,*		Select both channels A and B.	
	Note:	this command is valid only if the PDI 5025 is equipped with two channels.	
		Trigger source selection (see section 6.6).	
TRS,T	***	- Timer (without external synchronization).	TRigger Source, Timer
TRS,T,S		- Timer (with external synchronization). The SYNC connector (Lemo) on the 5140 module provides the input for the external synchronization.	TRigger Source, Timer, Synchro
TRS,E		- Incremental encoder without index signal. The encoder can be rotational or linear.	TRigger Source, Encoder
TRS,E,S		- Linear incremental encoder with index signal. This command can also be used with linear or rotational encoders which are not supplied with an index signal and for which the "zero" position must be defined externally (IND command).	TRigger Source, Encoder, Synchro
TRS,E,d..d		- Rotational incremental encoder with d..d cycles per turn and with index signal.	TRigger Source, rotational Encoder
	Note:	the 5140 module counts 4 pulses for each encoder cycle.	
TRS,X		External hardware trigger. The hardware trigger is fed into the appropriate Lemo connector of the 5150 module.	TRigger Source, eXternal
TRS,X,S		External hardware trigger with external synchronization.	TRigger Source, eXternal, Synchro
	Note:	When executing the TRS command, the previous TRI sequence is cancelled, the motor is stopped and the MOT,A becomes inactive.	

SYNTAX	DEFAULT	DESCRIPTION	MNEMONIC
TRI,s,a/n1,C1/n2,C2/.../n20,C20			TRIgger sequence
	+	s defines the sense of displacement	
	0	a defines the absolute position of the start (1st trigger): 0 to 2^{23} + sign. ni is the number of integration intervals of the "i" subsequence (ni = 1 to 65535 or ni=*); Ci is the "size" of the integration intervals of the "i" subsequence given in number of counts of the 5140 module (1 to 2^{23}). The max. number of subsequences is 20.	
	Note:	an integration interval is defined by two consecutive trigger signals. The second signal also starts the next interval, meaning that there is no dead time between each integration intervals.	
TRI,?		Returns a string containing the current values of the sequence.	
SGA,i,d..d		Sets the Gain of the 5150 module, channel "i" i = A, B or * (* means: both channels) d..d is the value of the Gain; the leading zeroes can be omitted.	Set GAin
SGA,10	***	The default Gain at power on is 10.	
	Note:	if the channel designation is omitted, the Gain is set on the active channel (or channels).	
ADJ,i,1:		set the selected channel in OFFSET ADJUST mode.	ADJust
ADJ,0 or ADJ,i,0		stop the OFFSET ADJUST mode on the activated (ADJ,0) or selected (ADJ,i,0) channel. If the selected channel is not in OFFSET ADJUST mode, the command is ignored. This command facilitates the adjustment of the input offset by running an infinite trigger sequence of 300ms and displaying the values on the front panel of the PDI 5025. See section 3.1.5 for a description of the displayed units. The OFFSET ADJUST mode can also be selected by pushing both buttons (GAIN and FNC) simultaneously. Bits 5-6-7 of Status 3 are set to 0-0-0 while the OFFSET ADJUST mode is selected.	
IND,s		s = + forward direction. s = - backward direction.	INDex
	***	if s is omitted the motor does not move. This command allows the encoder position to be initialized. The motor is driven in the selected direction until the index signal or an external synchronization pulse arrives; then the trigger source counter is reset, the motor is stopped and, finally, bit 0 of STATUS 1 is set to 1.	

SYNTAX	DEFAULT	DESCRIPTION	MNEMONIC
MOT,s		Starts the motor. s = + starts the motor in the Forward direction (FW) s = - starts the motor in the Backward direction (BW)	MOTor
MOT,S	***	stops the motor and cancels the MOT,A command.	MOTor, Stop
MOT,A		Automatic start of the motor. The motor starts automatically following the RUN command. The sense of rotation (or displacement) is defined by the TRI command. When the sequence of measurements is completed, the motor is stopped.	MOTor, Automatic
	Note:	the MOT,S command cancel MOT,A.	
WARNING		The motor and encoder wiring must be such that MOT,+ (MOT,-) starts the motor in the Forward (Backward) direction (respectively the FW and BW front panel LEDs are lit).	
RUN		starts the TRI sequence or stands-by the next external synchronization signal as defined by the TRS command (TRS,T,S or TRS,E,S or TRS,X,S). Having already performed the MOT,A command, the motor starts in the direction defined by the TRI command.	
	Note:	when executed before the completion of the sequence of measurements the RUN command generates a syntax error.	
BRK		Stops the measurement sequence and the motor. Terminate an external trigger measurement.	BReaK
EOD,a1,..,an	^Z	Defines an "end of data" string. a1,..,an decimal ASCII code of characters defining the "end of data". The string "a1,..,an" is returned following the ENQ command when the measurement buffer is empty.	End Of Data
	Note:	the ASCII Dec. 26 (^Z) character is the default "end of data" string at power on or if "a1,..,an" is omitted.	

SYNTAX	DEFAULT	DESCRIPTION	MNEMONIC
CUM,0	***	integrated values are stored separately.	CUMulative
CUM,1,S		the value cumulated from the start of the measurement is stored in memory at the end of each integration interval. If the command NBO,1 has been sent previously to the PDI 5025, the cumulated value is cleared to 0 at each occurrence of an overrange.	
CUM,1,L		only the last cumulated value is stored and available for reading. This mode is incompatible with the block data transfer (IMD,0) and forces an immediate data transfer mode (IMD,1). See section 7.2.3 concerning timing limitations.	
	Note:	It is not possible to change the storage mode during an active run.	
IMD,1	***	Flux values can be read during the measurement. The flux values can be read <u>before</u> the sequence of measurements has been completed. The values are read one by one. If reading is performed before a data is available, the PDI 5025 returns <CR><LF>. The "Data ready" bit in STATUS 1 (bit 2) being set to 1 indicates that a new measurement value is available.	IMmeDiate
IMD,0		Flux values are read as a block at the <u>end</u> of the sequence of measurements and as soon as the microprocessor has completed the calculations. The different values in the block are separated by a <CR><LF>. The bit 2 in STATUS 1 being set to 1 indicates that the block of data can be read. Note: this mode is incompatible with the CUM,1,L command.	
ENQ		Valid only for the <u>RS 232C</u> interface. This command allows data to read in the manner specified by the IMD command. When using the <u>IEEE 488</u> interface, this command is not valid and reading is carried out by addressing the PDI 5025 as a Talker. Data are transferred according to the IMD command setting.	ENQuiry
	Note:	- when the data buffer has been read and reset, a new read request will make the PDI 5025 return either the ASCII Dec. 26 character (^Z) or the string programmed with EOD. - This command is only used to read the measured values and must not be sent to read the status registers.	

SYNTAX	DEFAULT	DESCRIPTION	MNEMONIC
FPT,n	***	n = 1, enables the front panel trigger input. n = 0, disables the front panel trigger input.	Front Panel Trigger
	Note:	this command can only be used when TRS,X or TRS,X,S have been executed previously. The typical application is to disable external trigger pulses which are not wanted.	
STH,d		Returns the STATUS d in Hex (2 ASCII char.)	STatus Hex
STB,d		Returns the STATUS d in Binary (8 ASCII char.) d = 1 to 7	STatus Binary
	Note:	if "d" is omitted, the STATUS 1 is returned by default. The first and last characters are respectively the MSB and LSB.	
DSP,i,xxxx		sends string XXXX to the display of the selected channel. The string can be of any length but only the first four characters are displayed. This command overwrites previous displays and the new display remains on until an other command which sends characters to the display is executed (DSP, SGA, TST, ADJ etc.). If xxxx is omitted the normal display (Gain and channel) is shown i = A, B, * or nothing.(* means both channels and nothing means the active channel or channels). The commas have to be present in all cases.	DiSPlay
VER		Returns the software revision number.	VERsion
CVR,i		Clears overrange of channel i (i = A, B or *)	Clear oVeRrange
	Note:	the + or - overrange LED lit on the front panel bargraph display is switched off. If i is omitted, the command is executed on the active channel (or channels). The CVR command is automatically executed following the RUN and SGA commands.	
RGA,A		Returns the current Gain of channel A.	Read GAin
RGA,B		Returns the current Gain of channel B.	Read GAin
	Note:	- the Gains, which can be different for the two channels, can be modified manually using the appropriate switch on the 5150 front panel or via the SGA commands. - If no parameter is given in the RGA command, the Gain of the active channel is returned. If two channels are active, A is considered as default.	

SYNTAX	DEFAULT	DESCRIPTION	MNEMONIC
RCT		Reads the counters of the 5140 module . After reading the counters (3 bytes), the PDI 5025 converts the result to decimal and transfers it in the format +/- ddd..d. Leading zeroes are suppressed.	Read CounTer
	Note:	in the TRS,E,d..d mode, the counter values returned are always positive and represent the number of pulses counted following the index signal; there is no indication of the number of turns.	
ZCT	***	Resets the counters of the 5140 module. Automatic at power on.	Zero CounTer
<u>WARNING</u>		Using this command requires particular care! Use only for specific and well defined applications! Resetting the counters to zero during a RUN, will perturb the sequence of measurements!	
NBO,0	***	The occurrence of an overange condition will immediately stop the measurement in progress (if any) and set bit 4 of status 1 to one.	Not Brk Overrange
NBO,1		This command disables the break of a run caused by an overange condition. This command must be used with great care because it is not possible to determine when the overange condition occurred and for how long it lasted. The wrong data are strored in the buffer as any valid data! However, bit 4 of Status 1 is set to one. If an overrange occur during the measurement, the returned value is 0 with an exclamation mark between the last digit and the channel descriptor (e.g. 0!A).	
TST		This command allows the PDI 5025 channel (or channels) to be tested automatically. Following this command, the state prior to its execution is restored. In the case of an encoder with index, the position of the index is lost and has to be reinitialized with the IND,s command.	TeST
AUT		Returns test measurements. The command sends the block of 5 measurements calculated during the automatic test performed following the power on sequence or by the TST command.	AUTo test

SYNTAX	DEFAULT	DESCRIPTION	MNEMONIC
ISC,i,0	***	Internal short circuit of channel "i": OFF	Input Short Circ.
ISC,i,1		Internal short circuit of channel "i": ON i = A, B or * (* means: both channels)	
	Note:	if "i" is omitted, the command will be executed on the active channel (or channels). The ISC short circuits the input of the 5150 integrator channel. However, the short circuit is made using an analog switch which is not as effective as one made by directly shorting the inputs.	
FNC,1	***	Definition of the front panel FNC switch. FNC,1 defines the FNC button as equivalent to a RUN command, starting a measurement sequence.	FuNCtion
	Note:	if an overrange LED has been lit, the first depression of the FNC switch clears it and a second depression will start the measurement sequence.	
FNC,0		Following this command, the first depression of the FNC button will set bit 5 of STATUS 2 to 1 and generate an SRQ to the host when using the IEEE 488 interface providing this bit is not masked.	
		<u>RS 232 C only</u>	
LLO,0	***	Enables the front panel buttons.	Local LOckout
LLO,1		Disables the front panel buttons.	Local LOckout
	Note:	the LLO command can be executed only when using the RS 232C interface and it operates as the Local Lockout in IEEE 488.	
		<u>IEEE 488 only</u>	
MSK,a,oo		Mask setting.	MaSK
		Sets a mask on STATUS a (a = 1,2) to define the conditions required to generate a SRQ interrupt. "oo" is the mask definition in octals.	
	Note:	the "oo" default values for the mask patterns of STATUS 1 and STATUS 2 are "00" and "20" respectively.	

SYNTAX	DEFAULT	DESCRIPTION	MNEMONIC
		RS 232 C	
SYN		A synchronization signal is generated by the host.	SYNchronization
	Note:	the SYN command can be executed only if the trigger source has been previously defined with a synchronization pulse (TRS,T,S or TRS,E,S or TRS,X,S).	
		IEEE 488	
SYN,1		Enables the IEEE 488 GET command to generate a synchronization signal.	
SYN,0	***	Disables the IEEE 488 GET command from generating a synchronization signal.	
		BOTH INTERFACES	
	Note:	The software synchronization replaces the hardware synchronization which can be fed into the appropriate SYNC input of the 5140 module.	

9. APPENDIX II. COMMAND REFERENCE GUIDE

SYNTAX	DEFAULT	MNEMONIC	PAGES
CHA,A	***	CHAnnel	33,59
CHA,B			
CHA,*			
TRS,T	***	TRigger Source, Timer	37,59
TRS,T,S		TRigger Source, Timer, Synchro	37,59
TRS,E		TRigger Source, Encoder	39,59
TRS,E,S		TRigger Source, Encoder, Synchro	38,59
TRS,E,d..d		TRigger Source, rotational Encoder	37,59
TRS,X		TRigger Source, eXternal	39,59
TRS,X,S		TRigger Source, eXternal, Synchro	39,59
TRI,s,a/n1,C1/n2,C2/.../n20,C20		TRIgger sequence	35,60
TRI,?		Returns a string containing the current values of the sequence.	60
SGA,i,d..d		Set GAin	33,60
SGA,10	***	The default Gain at power on = 10.	
ADJ,i,1:		ADJust offset	34,60
ADJ,0 or ADJ,i,0		stop ADJust mode	
IND,s		INDex	37,38, 60
MOT,s		MOTor	51,61
MOT,S	***	MOTor, Stop	
MOT,A		MOTor, Automatic	
RUN		RUN	61
BRK		BReaK run	50,61
EOD,a1,..,an		End Of Data	50,61
CUM,0		storage mode,	48,62
CUM,1,S		CUMulative or	
CUM,1,L		individual	

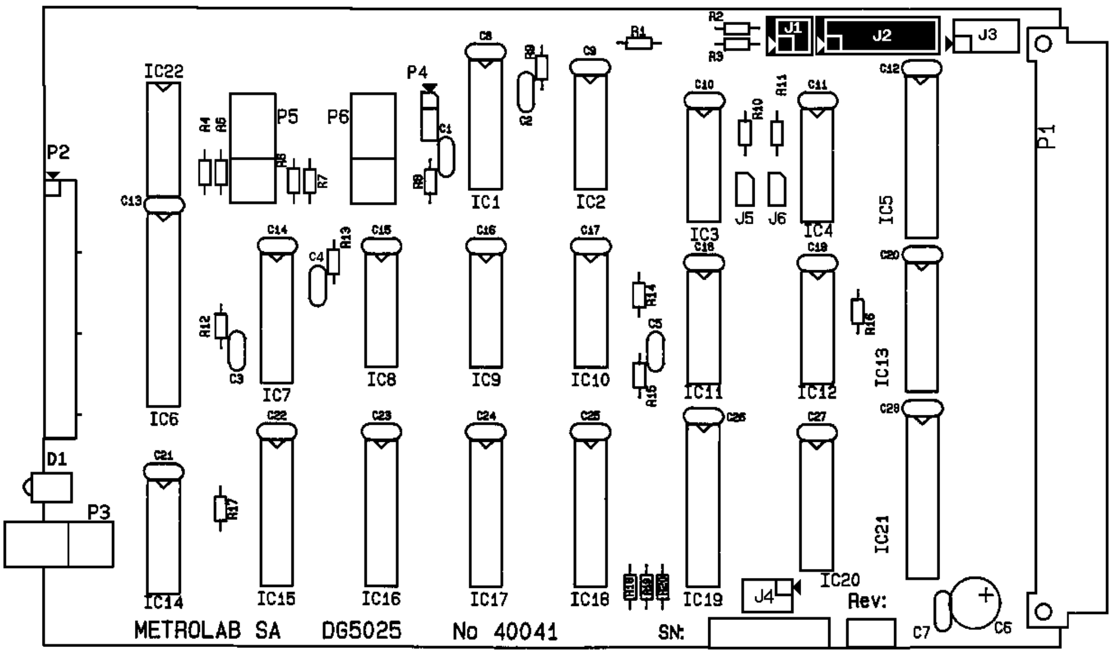
SYNTAX	DEFAULT	MNEMONIC	PAGES
IMD,1		IMeDiate or	48,62
IMD,0		block data trans.	49,62
ENQ		ENQuiry	62
FPT,n	***	Front Panel Trigger	39,63
STH,d		read STatus Hex.	41,63
STB,d		read STatus Bin.	
DSP,i,xxxx		send char. to DiSPlay	63
VER		software VERsion	63
CVR,i		Clear oVeRrange	63
RGA,i		Read GAin	63
RCT		Read CounTer	64
ZCT		Zero CounTer	64
NBO,0		Not Brk Overrange	64
NBO,1			
TST		perform TeST	51,64
AUT		read AUTo test data	51,64
ISC,i,0		Input Short Circ.	52,65
ISC,i,1			
FNC,1		button	33,65
FNC,0		FuNCtion	
LLO,0		Local LOckout	65
LLO,1		RS 232 C only	
MSK,a,oo		MaSK	55,65
		IEEE 488 only	
SYN		SYNchronization	53,66
		RS 232 C	
SYN,1		SYNchronization	53,66
SYN,0		IEEE 488	

10. APPENDIX III. PDI 5025 JUMPERS AND MICROSWITCHES

10.1 5150 INTEGRATOR MODULE: DIGITAL BOARD: JUMPERS

JUMPER	INSERTED	NOT INSERTED	USER SELECT.	FACTORY SELECT.	FACTORY SETTING
J1	1-2: external trigger on rising edge	3-4: external trigger on falling edge	*		1-2
J2	Board address			*	1-2 and 11-12 (chan. A) 3-4 and 11-12 (chan. B)
J3	Interrupt select			*	5-6 and 7-8
J4	VFC frequency code			*	1-2 (100 kHz) 3-4 (500 kHz) 5-6 (1 MHz)
J5		Open G96 line of hardware trigger		*	Inserted
J6		G96 bus normal line		*	Inserted

Note: Modifications must be implemented by qualified personnel.
The user can only change the jumpers indicated with a * in the USER SELECT. column. (Blue jumpers)
WARNING: the factory adjustments which are indicated with a * in the FACTORY SELECT. column must not
be changed. (Red jumpers)



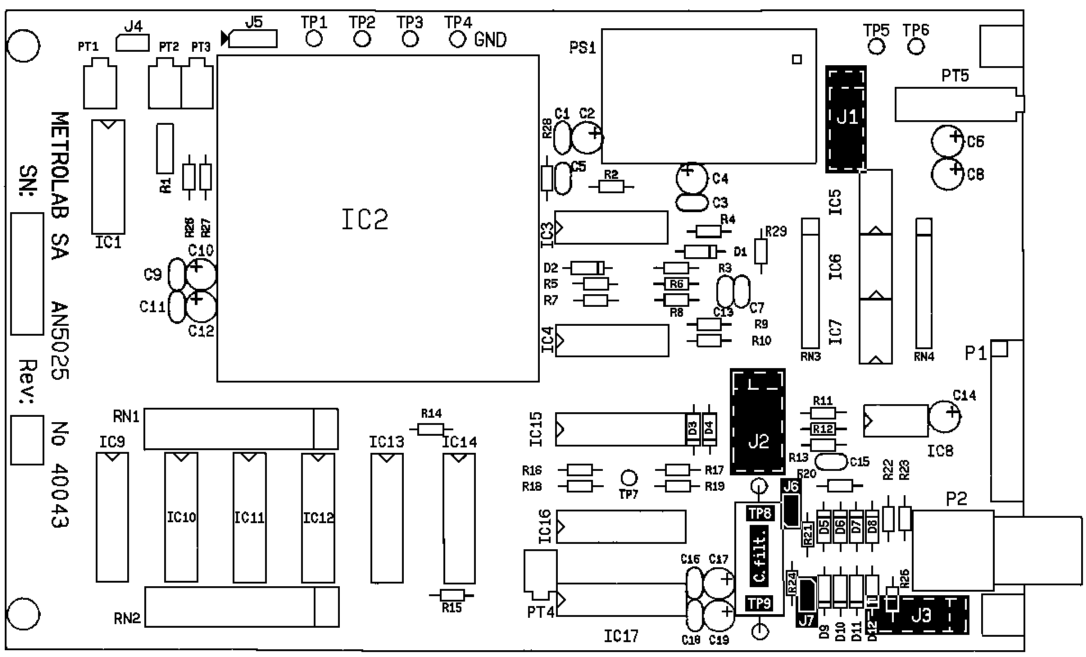
10.2 5150 INTEGRATOR MODULE: ANALOG BOARD: JUMPER

JUMPER	INSERTED	NOT INSERTED	USER SELECT.	FACTORY SELECT.	FACTORY SETTING
J1	Connects analog GND to general GND	Floating analog GND	*		Inserted
J2	1-2: coil shield at mid point of input voltage	2-3: coil shield to analog GND	*		2-3
J3	Balanced input Zin = 2 Megohms	Unbalanced input Zin = ~1000 Megohms	*		Inserted
J4	VFC adjustments			*	Inserted
J5	VFC adjustments			*	1-2
J6	Direct analog input	Input filter; 1 kohms + cap. !!	*		Inserted
J7	Direct analog input	Input filter; 1 kohms + cap. !!	*		Inserted

Note: Modifications must be implemented by qualified personnel.
The user can only change the jumpers indicated with a * in the USER SELECT. column. (Blue jumpers)

<u>WARNING</u>:

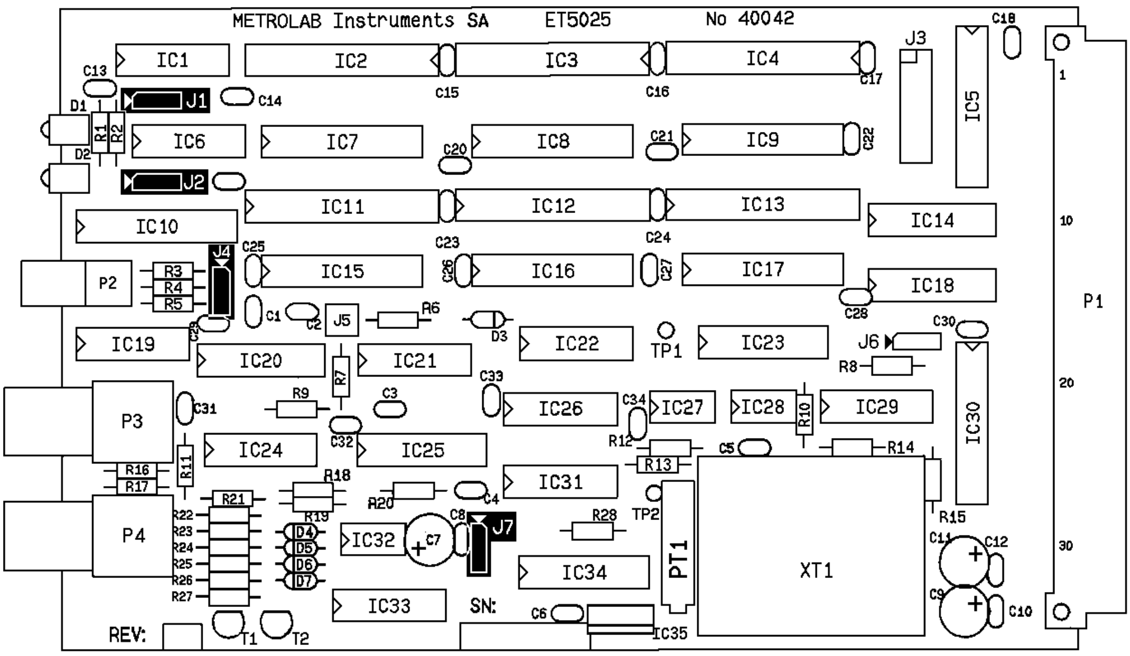
- The factory adjustments which are indicated with a * in the FACTORY SELECT. column must not be changed. (Red jumpers)
- The jumpers J1 to J3 are located on the solder side of the module and are covered by the shield plate, which must be removed to access the jumpers.



10.3 5140 TRIGGER MODULE: JUMPERS

JUMPER	INSERTED	NOT INSERTED	USER SELECT.	FACTORY SELECT.	FACTORY SETTING
J1	Encoder A channel 1-2: polarity +	Encoder A channel 2-3: polarity -	*		1-2
J2	1-2: Index polarity +	2-3: Index polarity -	*		1-2
J3	Board address			*	11-12 13-14
J4	Ext. synch. polarity 1-2: falling edge	Ext. synch. polarity 2-3: rising edge	*		1-2
J5	Reserved			*	none
J6	Interrupt selection			*	1-2
J7	1-2: motor power supply: 24 V	2-3: motor power supply: 12 V	*		2-3

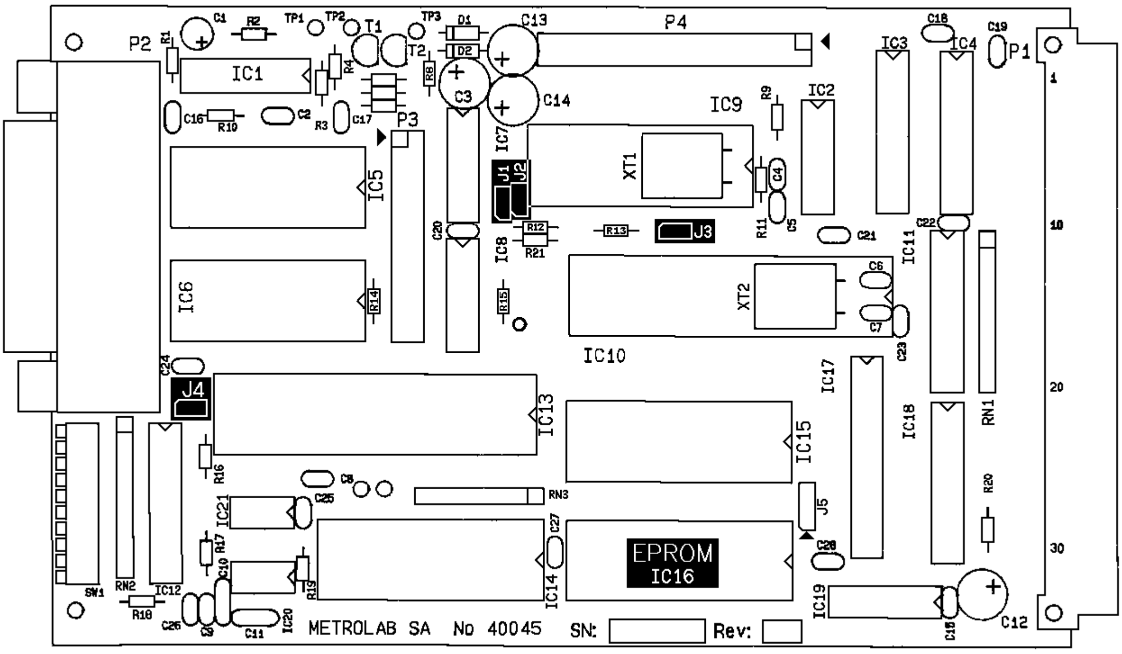
Note: Modifications must be implemented by qualified personnel.
The user can only change the jumpers indicated with a * in the USER SELECT. column. (Blue jumpers)
WARNING: the factory adjustments which are indicated with a * in the FACTORY SELECT. column must not
be changed. (Red jumpers)



<u>10.4 5130 INTERFACES MODULE: JUMPERS</u>

JUMPER	INSERTED	NOT INSERTED	USER SELECT.	FACTORY SELECT.	FACTORY SETTING
J1	RS 232 C DCD line disable	DCD enable	*		Inserted
J2	RS 232 C CTS line disable	CTS enable	*		Inserted
J3	RS 232 C DSR line disable	DSR enable	*		Inserted
J4	Watchdog enable	Watchdog disable	*		Inserted
J5	1-2: 27512 Eprom	2-3: 27256 Eprom		*	

Note: Modifications must be implemented by qualified personnel.
The user can only change the jumpers indicated with a * in the USER SELECT. column. (Blue jumpers)
<u>WARNING</u>: the factory adjustments which are indicated with a * in the FACTORY SELECT. column must not
be changed. (Red jumpers)



10.5 5130 INTERFACES MODULE: MICROSWITCHES

SWITCHES	IEEE-488 OPERATION "1" <-SWITCH POS.-> "0"		RS232c OPERATION "1" <-SWITCH POS.-> "0"	
1	DEVICE ADDRESS		BAUD RATE SELECTION	
2				
3				
4			8 bits	7 bits
5			2 stop bits	1 stop bit
6	Always "0"		Parity	No Parity
7	Always "0"		Even	Odd
8	NO "CR/LF"	"CR/LF"	XON/XOFF	No XON/XOFF
9	IEEE 488	--	--	RS 232 c
10	Talker Only	List./Talker	Autonomous	Conversational

10.6 5130 MODULE: MICROSWITCHES: BAUD RATE SELECTION

Switches 3	Switches 2	Switches 1	BAUD RATE
0	0	0	300
0	0	1	600
0	1	0	1200
0	1	1	2000
1	0	0	2400
1	0	1	4800
(limited use, see section 5.8)			
1	1	0	9600
1	1	1	19200

11. APPENDIX IV. QUICK REFERENCE TO STATUS REGISTERS

STATUS 1: Measurement Status (1 byte)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
STATUS 2 FLAG	Always 0	Command Error	Overrang e Error	End of Run	Data Ready	Trigger	Synchro

STATUS 2: Error Status (1 byte)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
0	0	FNC button	Power ON or Reset	Autotest failed	Encoder Count Error	Data Buffer Full	Overlap Timeout

STATUS 3: Trigger Source Module 5140 (1 byte)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
<- T R S mode ->			Infinite Sequence	Run Active	Forwards or Backward s	Motor +	Motor -

STATUS 4: PDI module 5130 (1 byte)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
<- V F C type ->		Overrang e -	Overrang e +	<- V F C type ->		Overrang e -	Overrang e +
Channel B				Channel A			

STATUS 5 (channel A) and 6 (channel B): Autotest (2 bytes)

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
0	0	0	Error in Analog Measure	OVR - Not ON ->Error	OVR + Not ON ->Error	OVR - Not OFF ->Error	OVR + Not OFF ->Error

STATUS 7: Acquisition status

bit 7	6	5	4	3	2	1	0
hex 80	40	20	10	8	4	2	1
0	0	0	Infinite Sequence Active	Run Active	Data Transfer Mode	Storage Mode	

12. APPENDIX V. SOFTWARE EXAMPLES

This appendix gives several listing to illustrate how the PDI 5025 can be programmed. Examples are for GW-Basic, HP Basic and 'C'.

HARDWARE	LANGUAGE	INTERFACE	DESCRIPTION	PAGES
PC,XT,AT	GW-BASIC	RS 232c	TIMER MODE, DIRECT DATA TRANSFER.	78
PC,XT,AT	GW-BASIC	RS 232c	TIMER MIDE, BLOCK DATA TRANSFER.	81
PC,XT,AT	GW-BASIC	RS 232c	INCREMENTAL ENDODER WITH INDEX, DIRECT DATA TRANSFER	85
PC,XT,AT	GW-BASIC	IEEE-488 (GPIB)	INCREMENTAL ENCODER WITH INDEX, DIRECT DATA TRANSFER	88
hp COMPUTER (200, 300 series)	HP BASIC	IEEE-488 (HPIB)	INCREMENTAL ENCODER WITH INDEX, DIRECT DATA TRANSFER	92
PC,XT,AT	"C" (QUICK-C)	RS 232c	INCREMENTAL ENCODER WITH INDEX, DIRECT DATA TRANSFER	95

MANRST.BAS

```
100 '************************************************************************
110 '*                                                                      *
120 '*                                      METROLAB INSTRUMENTS SA / GENEVA *
130 '*                                                      L.M. JUNE 1991 *
140 '*  PRECISION DIGITAL INTEGRATOR PDI 5025                               *
150 '*  =====================================                               *
160 '*                                                                      *
170 '*  LANGUAGE  :  GW-BASIC.                                              *
180 '*  INTERFACE :  RS232C.                                                *
190 '*                                                                      *
200 '*  PDI 5025  :  TIMER MODE,                                            *
210 '*               DIRECT DATA TRANSFER.                                  *
220 '*                                                                      *
230 '************************************************************************
240 'PDI'S FRONT PANEL MICRO-SWITCHES SETTING (1 to 10 ): 1011000000
250 'EXECUTES A RUN IN TIMER MODE.
260 'ACCEPT 1 OR 2 CHANNELS.
270 'PDI'S COMMANDS ARE SENT TO PDI IN TWO DIFFERENT WAYS :
280 '     1) COMMANDS WHICH DO NOT REQUIRE A RESPOND FROM PDI ARE EXECUTED
290 '        DIRECTLY BY WRITING CORRESPONDING STRING IN UNIT #1
300 '        (EXAMPLE : PRINT #1,"TRS,T").
310 '     2) COMMANDS WHICH REQUIRE A RESPOND FROM PDI ARE EXECUTED VIA
320 '        THE SUBROUTINE LINE 1190. IN THIS CASE THE COMMAND STRING IS
330 '        PASSED TO THE SUBROUTINE WITH THE STRING VARIABLE SEND$ AND
340 '        THE RESPOND OF THE PDI IS RETURNED IN THE VARIABLE IN$.
350 '        (EXAMPLE : SEND$="STB,2" : GOSUB 1190 : STB2$ = IN$).
360 'THE READING OF STATUS 1 IS DONE BY THE ROUTINE LINE 970. IN ADDITION
370 '  OF THE READING, THE SUBROUTINE CHECKS THE END OF RUN BIT AND UPDATES
380 '  THE VARIABLE ENDRUN IF NECESSARY. THE STATUS IS RETURNED IN
390 '  VARIABLE STB1$.
400 '
410 'INITIALIZATION OF LOCAL VARIABLES
420     IMES = 0 : ENDRUN = 0 : VSA# = 0 : VSB# = 0 : CLS
430 'INITIALIZATION RS232C : PORT 1, 4800 BAUDS, NO PARITY, 8 BITS, 1 STOP BIT
440     OPEN "COM1:4800,N,8,1,LF" AS #1
450 'RESET (READ) STATUS 1 AND 2 (WE DO NOT CARE OF CONTENT)
460     SEND$="STB,1" : GOSUB 1190
470     SEND$="STB,2" : GOSUB 1190
```

MANRST.BAS continued

```
480 ' SET PDI5025'S PARAMETERS :
490     PRINT #1,"CHA,*"                'BOTH CHANNELS SELECTED (DEFAULT)
500     PRINT #1,"SGA,A,5"              'SET GAIN CHANNEL A
510     PRINT #1,"SGA,B,200"            'SET GAIN CHANNEL B (IF CHANNEL B PRESENT)
520     PRINT #1,"TRS,T"                'TRIGGER SOURCE : TIMER  (DEFAULT)
530     PRINT #1,"TRI,,0/100,200"       'TRIGGER SEQUENCE
540     PRINT #1,"IMD,1"                'DIRECT TRANSFERT        (DEFAULT)
550     PRINT #1,"CUM,0"                'INDIVUDUAL VALUES       (DEFAULT)
560 'READ STATUS 1 AND CHECK COMMAND ERROR BIT
570     GOSUB 970
580     IF MID$(STB1$,3,1) <> "0" THEN GOTO 490
590     INPUT "PRESS ENTER TO START RUN",DUMMY
600 'START RUN WITH COMMAND RUN
610     PRINT #1,"RUN"
620 'INFORM USER THAT RUN STARTED
630     PRINT #1,"DSP,,RUN "
640     PRINT "RUN IN PROGRESS"
650 'READ STATUS 1 AND WAIT UNTIL DATA READY BIT IS 1
660     GOSUB 970
670     WHILE MID$(STB1$,6,1) <> "1"
680        GOSUB 970
690     WEND
700 'READ DATA UNTIL DATA READY BIT IN STATUS 1 IS 0
710     WHILE MID$(STB1$,6,1) = "1"
720 'EXECUTE ENQ COMMAND TO GET DATA
730       SEND$="ENQ" : GOSUB 1190
740 'CHECK IF "END OF DATA" : ^Z (ASCII 26).
750       IF INSTR(IN$,CHR$(26)) <> 0 THEN GOTO 920
760 'CHECK IF DATA FROM CHANNEL B
770       IF INSTR(IN$,"B")=0 THEN GOTO 810
780         VSB# = VAL(LEFT$(IN$,INSTR(IN$,"B")-2))  ' VALUE CHANNEL B
790         GOTO 880
800 'CHECK IF DATA FROM CHANNEL A
810       IF INSTR(IN$,"A")=0 THEN GOTO 880
820         VSA# = VAL(LEFT$(IN$,INSTR(IN$,"A")-2))  ' VALUE CHANNEL A
830         IMES=IMES+1                              ' # OF MEASURMENT
840 'PRINT VALUES
850       FORMAT$ = "####)  CHA A : ##########    CHA B : ##########   (* 10-8 V.s)"
860       PRINT USING FORMAT$;IMES,VSA#,VSB#
870 'READ STATUS 1 IN ORDER TO CHECK DATA READY BIT IN WHILE LOOP
880       GOSUB 970
890     WEND
900 'IF RUN NOT FINISHED, START WAITING AGAIN FOR DATA READY BIT
910     IF ENDRUN = 0 THEN GOTO 650
920 'DEFAULT DISPLAY ON PDI
930     PRINT #1,"DSP,,"
940     CLOSE #1
950     END
```

MANRST.BAS continued

```
960 '
970 'SUBROUTINE TO READ STATUS 1 AND CHECK END OF RUN BIT
980 '====================================================
990     SEND$="STB,1" : GOSUB 1190
1000     STB1$=IN$
1010 'CHECK END OF RUN BIT
1020     IF MID$(STB1$,5,1) <> "1" THEN RETURN
1030 'RUN FINISHED
1040     ENDRUN = 1
1050 'INFORM USER THAT RUN IS OVER AND READING CONTINUES
1060     PRINT "RUN FINISHED, READING IN CONTINUING"
1070     PRINT #1,"DSP,,READ"
1080 'CHECK OVERRANGE ERROR BIT
1090     IF MID$(STB1$,4,1) <> "1" THEN RETURN
1100 'INFORM USER THAT OVERRANGE OCCURED
1110     PRINT "ATTENTION : OVERRANGE OCCURED, RUN STOPPED"
1120     PRINT #1,"DSP,,OVER"
1130     INPUT "PRESS ENTER TO CONTINUE",DUMMY
1140 'CLEAR OVERRANGE
1150     PRINT #1,"CVR"
1160     GOTO 920
1170     RETURN
1180 '
1190 'SUBROUTINE TO EXECUTE A COMMAND WHICH REQUIRES A RESPOND FROM PDI
1200 '=================================================================
1210 'SEND TO PDI COMMAND SEND$
1220     PRINT #1,SEND$
1230     IN$=""
1240 'ACCEPT DATA UNTIL CARRIAGE RETURN AND LINE FEED FOUND OR ^Z
1250     WHILE (INSTR(IN$,CHR$(13)+CHR$(10))=0) AND (INSTR(IN$,CHR$(26))=0)
1260        NPASS=0
1270 'WAIT FOR DATA IN RS232 BUFFER
1280       WHILE EOF(1)
1290            NPASS=NPASS+1
1300 'IF WAIT TO LONG, TRY AGAIN (COMMAND NOT DECODED BY PDI 5025)
1310            IF NPASS>500 THEN GOTO 1220
1320       WEND
1330 'ADD INPUT CHARACTERS TO STRING IN$
1340       IN$=IN$+INPUT$(LOC(1),#1)      ' BUILD STRING "IN$"
1350    WEND
1360    RETURN
```

MANRSTBL.BAS

```
100 '*************************************************************************
110 '*                                                                       *
120 '*                                     METROLAB INSTRUMENTS SA / GENEVA *
130 '*                                                      L.M. JUNE 1991 *
140 '*  PRECISION DIGITAL INTEGRATOR PDI 5025                                *
150 '*  =====================================                                *
160 '*                                                                       *
170 '*  LANGUAGE  :  GW-BASIC.                                               *
180 '*  INTERFACE :  RS232C.                                                 *
190 '*                                                                       *
200 '*  PDI 5025  :  TIMER MODE,                                             *
210 '*               BLOCK DATA TRANSFER.                                    *
220 '*                                                                       *
230 '*************************************************************************
240 'PDI'S FRONT PANEL MICRO-SWITCHES SETTING (1 to 10 ): 1011000100
250 'EXECUTES A RUN IN TIMER MODE.
260 'ACCEPT 1 OR 2 CHANNELS.
270 'PDI'S COMMANDS ARE SENT TO PDI IN TWO DIFFERENT WAYS :
280 '     1) COMMANDS WHICH DO NOT REQUIRE A RESPOND FROM PDI ARE EXECUTED
290 '        DIRECTLY BY WRITING CORRESPONDING STRING IN UNIT #1
300 '        (EXAMPLE : PRINT #1,"TRS,T").
310 '     2) COMMANDS WHICH REQUIRE A RESPOND FROM PDI ARE EXECUTED VIA
320 '        THE SUBROUTINE LINE 1520. IN THIS CASE THE COMMAND STRING IS
330 '        PASSED TO THE SUBROUTINE WITH THE STRING VARIABLE SEND$ AND
340 '        THE RESPOND OF THE PDI IS RETURNED IN THE VARIABLE IN$.
350 '        (EXAMPLE : SEND$="STB,2" : GOSUB 1520 : STB2$ = IN$).
360 'THE READING OF STATUS 1 IS DONE BY THE ROUTINE LINE 1300. IN ADDITION
370 '  OF THE READING, THE SUBROUTINE CHECKS THE END OF RUN BIT AND UPDATES
380 '  THE VARIABLE ENDRUN IF NECESSARY. THE STATUS IS RETURNED IN
390 '  VARIABLE STB1$.
400 '
410 'INITIALIZATION OF LOCAL VARIABLES
420     IMES = 0 : ENDRUN = 0 : VSA# = 0 : VSB# = 0 : TRUE = -1 : FALSE = 0 : CLS
430 'INITIALIZATION RS232C : PORT 1, 4800 BAUDS, NO PARITY, 8 BITS, 1 STOP BIT
440     OPEN "COM1:4800,N,8,1,LF" AS #1
450 'OPEN TEMPORARY FILE IN CURRENT DIRECTORY
460     OPEN "PDIBLOCK.TMP" FOR OUTPUT AS #2
470 'RESET (READ) STATUS 1 AND 2 (WE DO NOT CARE OF CONTENT)
480     SEND$="STB,1" : GOSUB 1520
490     SEND$="STB,2" : GOSUB 1520
500 ' SET PDI5025'S PARAMETERS :
510     PRINT #1,"CHA,*"               'BOTH CHANNELS SELECTED (DEFAULT)
520     PRINT #1,"SGA,A,5"             'SET GAIN CHANNEL A
530     PRINT #1,"SGA,B,200"           'SET GAIN CHANNEL B (IF CANNEL B PRESENT)
540     PRINT #1,"TRS,T"               'TRIGGER SOURCE : TIMER  (DEFAULT)
550     PRINT #1,"TRI,,0/100,20"       'TRIGGER SEQUENCE
560     PRINT #1,"IMD,0"               'BLOCK TRANSFERT
570     PRINT #1,"CUM,0"               'INDIVUDUAL VALUES       (DEFAULT)
```

MANRSTBL.BAS continued

```
580 'READ STATUS 1 AND CHECK COMMAND ERROR BIT
590     GOSUB 1300
600     IF MID$(STB1$,3,1) <> "0" THEN GOTO 510
610     INPUT "PRESS ENTER TO START RUN",DUMMY
620 'START RUN WITH COMMAND RUN
630     PRINT #1,"RUN"
640 'INFORM USER THAT RUN STARTED
650     PRINT #1,"DSP,,RUN "
660     PRINT "RUN IN PROGRESS"
670 'READ STATUS 1 AND WAIT UNTIL DATA READY BIT IS 1
680     GOSUB 1300
690     WHILE MID$(STB1$,6,1) <> "1"
700        GOSUB 1300
710     WEND
720 'INFORM USER THAT TRANSFER BEGINS
730     PRINT "CALCULATION FINISHED, TRANSFERT IN PROGRESS"
740     PRINT #1,"DSP,,TRAN"
750     PAUSE = FALSE
760 'EXECUTE ENQ COMMAND TO GET DATA
770     PRINT #1,"ENQ"
780     IN$ = ""
790     NPASS = 0
800 'CHECK IF "END OF DATA" : ^Z (ASCII 26).
810     WHILE INSTR(IN$,CHR$(26)) = 0
820 'WAIT FOR DATA IN RS232 BUFFER
830       WHILE EOF(1)
840         NPASS=NPASS+1
850 'IF WAIT TO LONG, TRY AGAIN (COMMAND NOT DECODED BY PDI 5025)
860         IF NPASS>500 THEN GOTO 770
870       WEND
880 'INPUT STRING FROM RS232  AND PUT IT INTO TEMPORARY FILE
890       IN$ = INPUT$(LOC(1),#1)
900       PRINT #2,IN$;
910 'PAUSE TRANSFERT IF TO MANY CHARACTERS IN RS232 BUFFER
920       IF PAUSE THEN GOTO 970
930         IF LOC(1) < 128 THEN GOTO 1010
940         PAUSE = TRUE
950         PRINT #1,CHR$(19)
960         GOTO 1010
970       IF LOC(1) > 0 THEN GOTO 1010
980 'RESUME TRANSFERT IF RS232 BUFFER EMPTY
990         PAUSE=FALSE
1000         PRINT #1,CHR$(17)
1010     WEND
1020 'END OF DATA TRANSFERT : CLOSE TEMPORARY FILE
1030     CLOSE #2
```

MANRSTBL.BAS continued

```
1040 'DEFAULT DISPLAY ON PDI
1050     PRINT #1,"DSP,,"
1060 'RE-OPEN TEMPORARY FILE FOR READING
1070     OPEN "PDIBLOCK.TMP" FOR INPUT AS #2
1080     WHILE NOT EOF(2)
1090 'READ ONE LINE AT A TIME
1100       LINE INPUT #2,LIN$
1110 'CHECK IF DATA FROM CHANNEL B
1120       IF INSTR(LIN$,"B")=0 THEN GOTO 1160
1130         VSB# = VAL(LEFT$(LIN$,INSTR(LIN$,"B")-2))  ' VALUE CHANNEL B
1140         GOTO 1220
1150 'CHECK IF DATA FROM CHANNEL A
1160       IF INSTR(LIN$,"A")=0 THEN GOTO 1220
1170         VSA# = VAL(LEFT$(LIN$,INSTR(LIN$,"A")-2))  ' VALUE CHANNEL A
1180         IMES=IMES+1                              ' # OF MEASURMENT
1190 'PRINT VALUES
1200       FORMAT$ = "####)  CHA A : ##########    CHA B : ##########   (* 10-8 V.s)"
1210       PRINT USING FORMAT$;IMES,VSA#,VSB#
1220     WEND
1230 'CLOSE AND DELETE TEMPORARY FILE
1240     CLOSE #2
1250     KILL "PDIBLOCK.TMP"
1260     CLOSE #1
1270     END
1280 '
1290 '
1300 'SUBROUTINE TO READ STATUS 1 AND CHECK END OF RUN BIT
1310 '====================================================
1320     SEND$="STB,1" : GOSUB 1520
1330     STB1$=IN$
1340 'CHECK END OF RUN BIT
1350     IF MID$(STB1$,5,1) <> "1" THEN RETURN
1360 'RUN FINISHED
1370     ENDRUN = 1
1380 'INFORM USER THAT RUN IS OVER AND READING CONTINUES
1390     PRINT "RUN FINISHED, CALCULATION IN PROGRESS"
1400     PRINT #1,"DSP,,CALC"
1410 'CHECK OVERRANGE ERROR BIT
1420     IF MID$(STB1$,4,1) <> "1" THEN RETURN
1430 'INFORM USER THAT OVERRANGE OCCURED
1440     PRINT "ATTENTION : OVERRANGE OCCURED, RUN STOPPED"
1450     PRINT #1,"DSP,,OVER"
1460     INPUT "PRESS ENTER TO CONTINUE",DUMMY
1470 'CLEAR OVERRANGE
1480     PRINT #1,"CVR"
1490     GOTO 1260
1500     RETURN
```

MANRSTBL.BAS continued

```
1510 '
1520 'SUBROUTINE TO EXECUTE A COMMAND WHICH REQUIRES A RESPOND FROM PDI
1530 '===============================================================
1540 'SEND TO PDI COMMAND SEND$
1550     PRINT #1,SEND$
1560     IN$=""
1570 'ACCEPT DATA UNTIL CARRIAGE RETURN AND LINE FEED FOUND OR ^Z
1580     WHILE (INSTR(IN$,CHR$(13)+CHR$(10))=0) AND (INSTR(IN$,CHR$(26))=0)
1590        NPASS=0
1600 'WAIT FOR DATA IN RS232 BUFFER
1610       WHILE EOF(1)
1620            NPASS=NPASS+1
1630 'IF WAIT TO LONG, TRY AGAIN (COMMAND NOT DECODED BY PDI 5025)
1640            IF NPASS>500 THEN GOTO 1550
1650       WEND
1660 'ADD INPUT CHARACTERS TO STRING IN$
1670       IN$=IN$+INPUT$(LOC(1),#1)       ' BUILD STRING "IN$"
1680    WEND
1690    RETURN
```

MANRSE.BAS

```
100 '***************************************************************************
110 '*                                                                         *
120 '*                                       METROLAB INSTRUMENTS SA / GENEVA *
130 '*                                                        L.M. JUNE 1991 *
140 '*  PRECISION DIGITAL INTEGRATOR PDI 5025                                  *
150 '*  =====================================                                  *
160 '*                                                                         *
170 '*  LANGUAGE  :  GW-BASIC.                                                 *
180 '*  INTERFACE :  RS232C.                                                   *
190 '*                                                                         *
200 '*  PDI 5025  :  INCREMENTAL ENCODER WITH INDEX MODE                       *
210 '*               DIRECT DATA TRANSFER.                                     *
220 '*                                                                         *
230 '***************************************************************************
240 'PDI'S FRONT PANEL MICRO-SWITCHES SETTING (1 to 10 ): 1011000000
250 'EXECUTES A RUN IN ENCODER MODE.
260 'ACCEPT 1 OR 2 CHANNELS.
270 'PDI'S COMMANDS ARE SENT TO PDI IN TWO DIFFERENT WAYS :
280 '     1) COMMANDS WHICH DO NOT REQUIRE A RESPOND FROM PDI ARE EXECUTED
290 '        DIRECTLY BY WRITING CORRESPONDING STRING IN UNIT #1
300 '        (EXAMPLE : PRINT #1,"TRS,T").
310 '     2) COMMANDS WHICH REQUIRE A RESPOND FROM PDI ARE EXECUTED VIA
320 '        THE SUBROUTINE LINE 1320. IN THIS CASE THE COMMAND STRING IS
330 '        PASSED TO THE SUBROUTINE WITH THE STRING VARIABLE SEND$ AND
340 '        THE RESPOND OF THE PDI IS RETURNED IN THE VARIABLE IN$.
350 '        (EXAMPLE : SEND$="STB,2" : GOSUB 1320 : STB2$ = IN$).
360 'THE READING OF STATUS 1 IS DONE BY THE ROUTINE LINE 1100. IN ADDITION
370 '  OF THE READING, THE SUBROUTINE CHECKS THE END OF RUN BIT AND UPDATES
380 '  THE VARIABLE ENDRUN IF NECESSARY. THE STATUS IS RETURNED IN
390 '  VARIABLE STB1$.
400 '
410 'INITIALIZATION OF LOCAL VARIABLES
420     IMES = 0 : ENDRUN = 0 : VSA# = 0 : VSB# = 0 : CLS
430 'INITIALIZATION RS232C : PORT 1, 4800 BAUDS, NO PARITY, 8 BITS, 1 STOP BIT
440     OPEN "COM1:4800,N,8,1,LF" AS #1
450 'RESET (READ) STATUS 1 AND 2 (WE DO NOT CARE OF CONTENT)
460     SEND$="STB,1" : GOSUB 1320
470     SEND$="STB,2" : GOSUB 1320
480 ' SET PDI5025'S PARAMETERS :
490     PRINT #1,"CHA,*"               'BOTH CHANNELS SELECTED (DEFAULT)
500     PRINT #1,"SGA,A,5"             'SET GAIN CHANNEL A
510     PRINT #1,"SGA,B,200"           'SET GAIN CHANNEL B (IF CHANNEL B PRESENT)
520     PRINT #1,"TRS,E,1024"          'TRIGGER SOURCE : ROTATIONAL ENCODER (1024 STEPS)
530     PRINT #1,"TRI,+,0/32,128"      'TRIGGER SEQUENCE
540     PRINT #1,"MOT,A"               'AUTOMATIC START AND STOP FOR MOTOR
550     PRINT #1,"IMD,1"               'DIRECT TRANSFERT        (DEFAULT)
560     PRINT #1,"CUM,0"               'INDIVUDUAL VALUES       (DEFAULT)
570 'READ STATUS 1 AND CHECK COMMAND ERROR BIT
580     GOSUB 1100
590     IF MID$(STB1$,3,1) <> "0" THEN GOTO 490
```

MANRSE.BAS continued

```
600 'INITIALIZE ENCODER INDEX POSITION
610     PRINT "INITIALIZE ENCODER POSITION"
620     PRINT #1,"DSP,,INDX"
630     PRINT #1,"IND,+"
640 'READ STATUS 1 AND CHECK SYNCHRO BIT
650     GOSUB 1100
660 'WAIT FOR SYNCHRO BIT = 1  (INDEX FOUND)
670     WHILE MID$(STB1$,8,1) <> "1"
680       GOSUB 1100
690     WEND
700     PRINT "INDEX FOUND"
710     PRINT #1,"DSP,,"
720     INPUT "PRESS ENTER TO START RUN",DUMMY
730 'START RUN WITH COMMAND RUN
740     PRINT #1,"RUN"
750 'INFORM USER THAT RUN STARTED
760     PRINT #1,"DSP,,RUN "
770     PRINT "RUN IN PROGRESS"
780 'READ STATUS 1 AND WAIT UNTIL DATA READY BIT IS 1
790     GOSUB 1100
800     WHILE MID$(STB1$,6,1) <> "1"
810        GOSUB 1100
820     WEND
830 'READ DATA UNTIL DATA READY BIT IN STATUS 1 IS 0
840     WHILE MID$(STB1$,6,1) = "1"
850 'EXECUTE ENQ COMMAND TO GET DATA
860       SEND$="ENQ" : GOSUB 1320
870 'CHECK IF "END OF DATA" : ^Z (ASCII 26).
880       IF INSTR(IN$,CHR$(26)) <> 0 THEN GOTO 1050
890 'CHECK IF DATA FROM CHANNEL B
900       IF INSTR(IN$,"B")=0 THEN GOTO 940
910         VSB# = VAL(LEFT$(IN$,INSTR(IN$,"B")-2))  ' VALUE CHANNEL B
920         GOTO 1010
930 'CHECK IF DATA FROM CHANNEL A
940       IF INSTR(IN$,"A")=0 THEN GOTO 1010
950         VSA# = VAL(LEFT$(IN$,INSTR(IN$,"A")-2))  ' VALUE CHANNEL A
960         IMES=IMES+1                              ' # OF MEASURMENT
970 'PRINT VALUES
980       FORMAT$ = "####)  CHA A : ##########    CHA B : ##########   (* 10-8 V.s)"
990       PRINT USING FORMAT$;IMES,VSA#,VSB#
1000 'READ STATUS 1 IN ORDER TO CHECK DATA READY BIT IN WHILE LOOP
1010       GOSUB 1100
1020     WEND
1030 'IF RUN NOT FINISHED, START WAITING AGAIN FOR DATA READY BIT
1040     IF ENDRUN = 0 THEN GOTO 780
1050 'DEFAULT DISPLAY ON PDI
1060     PRINT #1,"DSP,,"
1070     CLOSE #1
1080     END
```

MANRSE.BAS continued

```
1090 '
1100 'SUBROUTINE TO READ STATUS 1 AND CHECK END OF RUN BIT
1110 '=====================================================
1120     SEND$="STB,1" : GOSUB 1320
1130     STB1$=IN$
1140 'CHECK END OF RUN BIT
1150     IF MID$(STB1$,5,1) <> "1" THEN RETURN
1160 'RUN FINISHED
1170     ENDRUN = 1
1180 'INFORM USER THAT RUN IS OVER AND READING CONTINUES
1190     PRINT "RUN FINISHED, READING CONTINUING"
1200     PRINT #1,"DSP,,READ"
1210 'CHECK OVERRANGE ERROR BIT
1220     IF MID$(STB1$,4,1) <> "1" THEN RETURN
1230 'INFORM USER THAT OVERRANGE OCCURED
1240     PRINT "ATTENTION : OVERRANGE OCCURED, RUN STOPPED"
1250     PRINT #1,"DSP,,OVER"
1260     INPUT "PRESS ENTER TO CONTINUE",DUMMY
1270 'CLEAR OVERRANGE
1280     PRINT #1,"CVR"
1290     GOTO 1050
1300     RETURN
1310 '
1320 'SUBROUTINE TO EXECUTE A COMMAND WHICH REQUIRES A RESPOND FROM PDI
1330 '==================================================================
1340 'SEND TO PDI COMMAND SEND$
1350     PRINT #1,SEND$
1360     IN$=""
1370 'ACCEPT DATA UNTIL CARRIAGE RETURN AND LINE FEED FOUND OR ^Z
1380     WHILE (INSTR(IN$,CHR$(13)+CHR$(10))=0) AND (INSTR(IN$,CHR$(26))=0)
1390       NPASS=0
1400 'WAIT FOR DATA IN RS232 BUFFER
1410       WHILE EOF(1)
1420            NPASS=NPASS+1
1430 'IF WAIT TO LONG, TRY AGAIN (COMMAND NOT DECODED BY PDI 5025)
1440            IF NPASS>500 THEN GOTO 1350
1450       WEND
1460 'ADD INPUT CHARACTERS TO STRING IN$
1470       IN$=IN$+INPUT$(LOC(1),#1)      ' BUILD STRING "IN$"
1480    WEND
1490    RETURN
```

MANIBE.BAS

```
100 '************************************************************************
110 '*                                                                      *
120 '*                                     METROLAB INSTRUMENTS SA / GENEVA *
130 '*                                                      L.M. JUNE 1991 *
140 '*  PRECISION DIGITAL INTEGRATOR PDI 5025                               *
150 '*  =====================================                               *
160 '*                                                                      *
170 '*  LANGUAGE  :  GW-BASIC.                                              *
180 '*  INTERFACE :  GPIB (IEEE-488)                                        *
190 '*                                                                      *
200 '*  PDI 5025  :  INCREMENTAL ENCODER WITH INDEX MODE                    *
210 '*               DIRECT DATA TRANSFER.                                  *
220 '*                                                                      *
230 '************************************************************************
240 'PDI'S FRONT PANEL MICRO-SWITCHES SETTING (1 TO 10) : 1011000010
250 'EXECUTES A RUN IN ENCODER MODE.
260 'ACCEPT 1 OR 2 CHANNELS.
270 'PDI'S COMMANDS ARE SENT TO PDI IN TWO DIFFERENT WAYS :
280 '     1) COMMANDS WHICH DO NOT REQUIRE A RESPOND FROM PDI ARE EXECUTED
290 '        VIA THE GPIB SUBROUTINE "IBWRT". THE COMMAND STRING IS PASSED
300 '        TO THE SUBROUTINE WITH THE STRING VARIABLE SEND$.
310 '     2) COMMANDS WHICH REQUIRE A RESPOND FROM PDI ARE EXECUTED VIA
320 '        THE SUBROUTINE LINE 1530. IN THIS CASE THE COMMAND STRING IS
330 '        PASSED TO THE SUBROUTINE WITH THE STRING VARIABLE SEND$ AND
340 '        THE RESPOND OF THE PDI IS RETURNED IN THE VARIABLE IN$.
350 '        (EXAMPLE : SEND$="STB,2" : GOSUB 1530 : STB2$ = IN$).
360 'THE READING OF STATUS 1 IS DONE BY THE ROUTINE LINE 1310. IN ADDITION
370 '  OF THE READING, THE SUBROUTINE CHECKS THE END OF RUN BIT AND UPDATES
380 '  THE VARIABLE ENDRUN IF NECESSARY. THE STATUS IS RETURNED IN
390 '  VARIABLE STB1$.
400 '
410 'INITIALIZATION HPIB
420   CLEAR   ,58077!
430   IBINIT1 = 58077!
440   IBINIT2 = IBINIT1 + 3
450   BLOAD "BIB.M",IBINIT1
460   CALL
IBINIT1(IBFIND,IBSTOP,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,I
BSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT)
470   CALL
IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRTF,IBWRTA,IBWRT,IBCMDA,IBCMD,IBRDF,IBRDA,IBRD,IBRPP,IBRSP
,IBDIAG,IBXTRC,IBSTA%,IBERR%,IBCNT%)
```

MANIBE.BAS continued

```
480  '------------------------------------------------------------------------
490  '|                                                                      |
500  '|  ADDRESS OF THE PDI5025 MUST BE SELECTED IN THE IBCONF.EXE SOFTWARE!!  |
510  '|  THE IDENTIFIER IS :    PDI                                          |
520  '|                                                                      |
530  '|  The program 'ADJUSTIB.BAS' start here. (Previous lines are for the  |
540  '|        National Instuments GPIB Board initialisation.)               |
550  '|                                                                      |
560  '------------------------------------------------------------------------
570  '
580 'Find PDI by identifier "PDI".Adress is given by configuration
590 'soft IBCONF.EXE
600 DEV$="PDI"
610 CALL IBFIND (DEV$,PDI)
620 IF PDI<0 THEN PRINT "ERROR IN INITIALISATION OF PDI":STOP
630 '
640 'INITIALIZATION OF LOCAL VARIABLES
650     IMES = 0 : ENDRUN = 0 : VSA# = 0 : VSB# = 0 : CLS
660 'RESET (READ) STATUS 1 AND 2 (WE DO NOT CARE OF CONTENT)
670     SEND$="STB,1" : GOSUB 1530
680     SEND$="STB,2" : GOSUB 1530
690 ' SET PDI5025'S PARAMETERS :
700     SEND$="CHA,*" : CALL IBWRT(PDI,SEND$)           'BOTH CHANNELS SELECTED (DEFAULT)
710     SEND$="SGA,A,5" : CALL IBWRT(PDI,SEND$)         'SET GAIN CHANNEL A
720     SEND$="SGA,B,200" : CALL IBWRT(PDI,SEND$)       'SET GAIN CHANNEL B (IF CHANNEL B
PRESENT)
730     SEND$="TRS,E,1024" : CALL IBWRT(PDI,SEND$)       'TRIGGER SOURCE : ROTATIONAL ENCODER
(1024 STEPS)
740     SEND$="TRI,+,0/32,128" : CALL IBWRT(PDI,SEND$)  'TRIGGER SEQUENCE
750     SEND$="MOT,A" : CALL IBWRT(PDI,SEND$)           'AUTOMATIC START AND STOP FOR MOTOR
760     SEND$="IMD,1" : CALL IBWRT(PDI,SEND$)           'DIRECT TRANSFERT        (DEFAULT)
770     SEND$="CUM,0" : CALL IBWRT(PDI,SEND$)          'INDIVIDUAL VALUES       (DEFAULT)
780 'READ STATUS 1 AND CHECK COMMAND ERROR BIT
790     GOSUB 1310
800     IF MID$(STB1$,3,1) <> "0" THEN GOTO 700
810 'INITIALIZE ENCODER INDEX POSITION
820     PRINT "INITIALIZE ENCODER POSITION"
830     SEND$="DSP,,INDX" : CALL IBWRT(PDI,SEND$)
840     SEND$="IND,+" : CALL IBWRT(PDI,SEND$)
850 'READ STATUS 1 AND CHECK SYNCHRO BIT
860     GOSUB 1310
870 'WAIT FOR SYNCHRO BIT = 1  (INDEX FOUND)
880     WHILE MID$(STB1$,8,1) <> "1"
890       GOSUB 1310
900     WEND
910     PRINT "INDEX FOUND"
920     SEND$="DSP,," : CALL IBWRT(PDI,SEND$)
930     INPUT "PRESS ENTER TO START RUN",DUMMY
940 'START RUN WITH COMMAND RUN
950     SEND$="RUN" : CALL IBWRT(PDI,SEND$)
960 'INFORM USER THAT RUN STARTED
970     SEND$="DSP,,RUN " : CALL IBWRT(PDI,SEND$)
980     PRINT "RUN IN PROGRESS"
```

MANIBE.BAS continued

```
990 'READ STATUS 1 AND WAIT UNTIL DATA READY BIT IS 1
1000     GOSUB 1310
1010     WHILE MID$(STB1$,6,1) <> "1"
1020        GOSUB 1310
1030     WEND
1040 'READ DATA UNTIL DATA READY BIT IN STATUS 1 IS 0
1050     WHILE MID$(STB1$,6,1) = "1"
1060 'ADDRESS THE PDI AS A TALKER TO GET DATA
1070        GOSUB 1580
1080 'CHECK IF "END OF DATA" : ^Z (ASCII 26).
1090       IF INSTR(IN$,CHR$(26)) <> 0 THEN GOTO 1260
1100 'CHECK IF DATA FROM CHANNEL B
1110       IF INSTR(IN$,"B")=0 THEN GOTO 1150
1120         VSB# = VAL(LEFT$(IN$,INSTR(IN$,"B")-2))  ' VALUE CHANNEL B
1130         GOTO 1220
1140 'CHECK IF DATA FROM CHANNEL A
1150       IF INSTR(IN$,"A")=0 THEN GOTO 1220
1160         VSA# = VAL(LEFT$(IN$,INSTR(IN$,"A")-2))  ' VALUE CHANNEL A
1170         IMES=IMES+1                              ' # OF MEASURMENT
1180 'PRINT VALUES
1190       FORMAT$ = "####)  CHA A : ##########    CHA B : ##########   (* 10-8 V.s)"
1200       PRINT USING FORMAT$;IMES,VSA#,VSB#
1210 'READ STATUS 1 IN ORDER TO CHECK DATA READY BIT IN WHILE LOOP
1220       GOSUB 1310
1230     WEND
1240 'IF RUN NOT FINISHED, START WAITING AGAIN FOR DATA READY BIT
1250     IF ENDRUN = 0 THEN GOTO 990
1260 'DEFAULT DISPLAY ON PDI
1270     SEND$="DSP,," : CALL IBWRT(PDI,SEND$)
1280     CLOSE #1
1290     END
1300 '
1310 'SUBROUTINE TO READ STATUS 1 AND CHECK END OF RUN BIT
1320 '==================================================
1330     SEND$="STB,1" : GOSUB 1530
1340     STB1$=IN$
1350 'CHECK END OF RUN BIT
1360     IF MID$(STB1$,5,1) <> "1" THEN RETURN
1370 'RUN FINISHED
1380     ENDRUN = 1
1390 'INFORM USER THAT RUN IS OVER AND READING CONTINUES
1400     PRINT "RUN FINISHED, READING CONTINUING"
1410     SEND$="DSP,,READ": CALL IBWRT(PDI,SEND$)
1420 'CHECK OVERRANGE ERROR BIT
1430     IF MID$(STB1$,4,1) <> "1" THEN RETURN
1440 'INFORM USER THAT OVERRANGE OCCURED
1450     PRINT "ATTENTION : OVERRANGE OCCURED, RUN STOPPED"
1460     SEND$="DSP,,OVER" : CALL IBWRT(PDI,SEND$)
1470     INPUT "PRESS ENTER TO CONTINUE",DUMMY
1480 'CLEAR OVERRANGE
1490     SEND$="CVR" : CALL IBWRT(PDI,SEND$)
1500     GOTO 1260
1510     RETURN
```

MANIBE.BAS continued

```
1520 '
1530 'SUBROUTINE TO EXECUTE A COMMAND WHICH REQUIRES A RESPOND FROM PDI
1540 '=================================================================
1550 'SEND TO PDI COMMAND SEND$
1560     CALL IBWRT(PDI,SEND$)
1570 'ADDRESSING THE PDI AS A TALKER TO READ IT (SUBROUTINE)
1580     IN$=SPACE$(50)
1590     CALL IBRD(PDI,IN$)  'DATA IN STRING IN$
1600     IN$=LEFT$(IN$,INSTR(IN$,CHR$(10))) ' BUILD STRING "IN$"
1610    RETURN
```

MANIBE.HPB

```
100 !**************************************************************************
110 !*                                                                        *
120 !*                                     METROLAB INSTRUMENTS SA / GENEVA *
130 !*                                                      L.M. JUNE 1991 *
140 !*  PRECISION DIGITAL INTEGRATOR PDI 5025                                 *
150 !*  ====================================                                  *
160 !*                                                                        *
170 !*  LANGUAGE  :  HP BASIC  (HEWLETT-PACKARD).                             *
180 !*  INTERFACE :  HP-IB.                                                   *
190 !*                                                                        *
200 !*  PDI 5025  :  INCREMENTAL ENCODER WITH INDEX,                          *
210 !*               DIRECT DATA TRANSFER.                                    *
220 !*                                                                        *
230 !**************************************************************************
240 !PDI'S FRONT PANEL MICRO-SWITCHES SETTING (1 to 10 ): 1001000010
250 !EXECUTES A RUN IN ENCODER MODE.
260 !ACCEPT 1 OR 2 CHANNELS.
270 !PDI'S COMMANDS ARE SENT TO PDI IN TWO DIFFERENT WAYS :
280 !     1) COMMANDS WHICH DO NOT REQUIRE A RESPOND FROM PDI ARE EXECUTED
290 !        DIRECTLY (EXAMPLE : OUTPUT Pdi;"TRS,T").
300 !     2) COMMANDS WHICH REQUIRE A RESPOND FROM PDI ARE EXECUTED VIA
310 !        THE SUBROUTINE LINE 1410. IN THIS CASE THE COMMAND STRING IS
320 !        PASSED TO THE SUBROUTINE WITH THE STRING VARIABLE SEND$ AND
330 !        THE RESPOND OF THE PDI IS RETURNED IN THE VARIABLE IN$.
340 !        (EXAMPLE : SEND$="STB,2" : GOSUB 1410 : STB2$ = IN$).
350 !THE READING OF STATUS 1 IS DONE BY THE ROUTINE LINE 1170. IN ADDITION
360 !  OF THE READING, THE SUBROUTINE CHECKS THE END OF RUN BIT AND UPDATES
370 !  THE VARIABLE ENDRUN IF NECESSARY. THE STATUS IS RETURNED IN
380 !  VARIABLE STB1$.
390 !
400 !INITIALIZATION OF LOCAL VARIABLES
410     DIM Format$[80]
420     Imes=0
430     Endrun=0
440     Vsa=0
450     Vsb=0
460     CLEAR SCREEN
470 !PDI 5025 ADDRESS
480     Hpib=7
490     Pdi=Hpib*100+9
500     ON INTR 7 GOSUB 1470
510     Mask=2                  !BIT 1 ENABLES SRQ INTERRUPTS
520     ENABLE INTR Hpib;Mask
530 !RESET (READ) STATUS 1 AND 2 (WE DO NOT CARE OF CONTENT)
540     Send$="STB,1"
550     GOSUB 1410
560     Send$="STB,2"
570     GOSUB 1410
```

MANIBE.HPB continued

```
580 ! SET PDI5025'S PARAMETERS :
590     OUTPUT Pdi;"CHA,*"                !BOTH CHANNELS SELECTED (DEFAULT)
600     OUTPUT Pdi;"SGA,A,5"              !SET GAIN CHANNEL A
610     OUTPUT Pdi;"SGA,B,200"            !SET GAIN CHANNEL B (IF CHANNEL B PRESENT)
620     OUTPUT Pdi;"TRS,E,360"            !TRIGGER SOURCE : TIMER  (DEFAULT)
630     OUTPUT Pdi;"TRI,,0/10,200"       !TRIGGER SEQUENCE
640     OUTPUT Pdi;"MOT,A"                !AUTOMATIC START AND STOP FOR MOTOR
650     OUTPUT Pdi;"IMD,1"                !DIRECT TRANSFERT        (DEFAULT)
660     OUTPUT Pdi;"CUM,0"                !INDIVUDUAL VALUES       (DEFAULT)
670     OUTPUT Pdi;"MSK,1,10"             !MASK : END OF RUN BIT ACTIVE
680 !READ STATUS 1 AND CHECK COMMAND ERROR BIT
690     GOSUB 1170
700     IF Stb1$[3;1]<>"0" THEN GOTO 590
710 !INITIALIZES INDEX POSITION OF ENCODER
720     OUTPUT Pdi;"IND,+"
730     OUTPUT Pdi;"DSP,,INDX"
740 !WAIT FOR SYNCHRO BIT = 1 IN STATUS 1
750     GOSUB 1170
760     WHILE Stb1$[8;1]<>"1"
770       GOSUB 1170
780     END WHILE
790     OUTPUT Pdi;"DSP,,"
800     PRINT "INDEX FOUND"
810     INPUT "PRESS ENTER TO START RUN",Dummy
820 !START RUN WITH COMMAND RUN
830     OUTPUT Pdi;"RUN"
840 !INFORM USER THAT RUN STARTED
850     OUTPUT Pdi;"DSP,,RUN "
860     PRINT "RUN IN PROGRESS"
870 !WAIT UNTIL END OF RUN
880     Npass=0
890     Npass=Npass+1
900     GOTO 890
910 !READ DATA UNTIL DATA READY BIT IN STATUS 1 IS 0
920     GOSUB 1170
930     WHILE Stb1$[6;1]="1"
940 !GET DATA FROM PDI 5025
950       ENTER Pdi;In$
960 !CHECK IF "END OF DATA" : ^Z (ASCII 26).
970       IF POS(In$,CHR$(26))<>0 THEN GOTO 1120
980 !CHECK IF DATA FROM CHANNEL B
990       IF POS(In$,"B")=0 THEN GOTO 1030
1000        Vsb=VAL(In$)                 ! VALUE CHANNEL B
1010        GOTO 1100
```

MANIBE.HPB continued

```
1020!CHECK IF DATA FROM CHANNEL A
1030      IF POS(In$,"A")=0 THEN GOTO 1100
1040        Vsa=VAL(In$)                ! VALUE CHANNEL A
1050        Imes=Imes+1                             ! # OF MEASURMENT
1060!PRINT VALUES
1070      OUTPUT Format$ USING "4D,"")  CHA A : "",12D,""   CHA B : "",12D,""  (* 10-8
V.s)"","";Imes,Vsa,Vsb
1080      PRINT Format$
1090!READ STATUS 1 IN ORDER TO CHECK DATA READY BIT IN WHILE LOOP
1100      GOSUB 1170
1110    END WHILE
1120    PRINT "READING FINISHED"
1130!DEFAULT DISPLAY
1140    OUTPUT Pdi;"DSP,,"
1150    STOP
1160!
1170!SUBROUTINE TO READ STATUS 1 AND CHECK END OF RUN BIT
1180!=====================================================
1190    Send$="STB,1"
1200    GOSUB 1410
1210     Stb1$=In$
1220 !CHECK END OF RUN BIT
1230     IF Stb1$[5;1]<>"1" THEN RETURN
1240 !RUN FINISHED
1250     Endrun=1
1260 !INFORM USER THAT RUN IS OVER AND READING CONTINUES
1270     PRINT "RUN FINISHED, READING IN PROGRESS"
1280     OUTPUT Pdi;"DSP,,READ"
1290 !CHECK OVERRANGE ERROR BIT
1300     IF Stb1$[4;1]<>"1" THEN RETURN
1310 !INFORM USER THAT OVERRANGE OCCURED
1320     PRINT "ATTENTION : OVERRANGE OCCURED, RUN STOPPED"
1330     OUTPUT Pdi;"DSP,,OVER"
1340     ENTER Pdi;In$
1350     INPUT "PRESS ENTER TO CONTINUE",Dummy
1360 !CLEAR OVERRANGE
1370     OUTPUT Pdi;"CVR"
1380     GOTO 1120
1390     RETURN
1400 !
1410 !SUBROUTINE TO EXECUTE A COMMAND WHICH REQUIRES A RESPOND FROM PDI
1420 !==================================================================
1430 !SEND TO PDI COMMAND SEND$
1440    OUTPUT Pdi;Send$
1450    ENTER Pdi;In$
1460    RETURN
1470 !
1480 !SRQ SERVICE ROUTINE
1490 !==================
1500    S=SPOLL(Pdi)
1510    IF S=0 THEN RETURN                  !NOT PDI GENERATED SRQ
1520 !INITIATES DATA READING
1530    GOTO 910
1540    RETURN
1550    END
```

MANRSE.C

```
/* ********************************************************************** */
/*                                        METROLAB Instruments SA / Geneva */
/*                                                        L.M. June 1991 */
/*  Precision Digital Integrator PDI 5025                               */
/*  =====================================                              */
/*                                                                      */
/*  Language  : QUICK-C (Microsoft).                                    */
/*  Interface : RS232C                                                  */
/*                                                                      */
/*  PDI 5025  : incremental encoder with index,                         */
/*              direct data transfer.                                   */
/*                                                                      */
/* ********************************************************************** */
/*  PDI's front panel micro-switches setting (1 to 10) : 1011000000     */
/*  Executes a run in encoder mode.                                     */
/*  Two functions are used to communicate with the PDI 5025 :           */
/*    - write_pdi() : send string in argument to the PDI 5025.          */
/*    - read_pdi()  : read data or parameters when necessary from PDI 5025. */
/*                    The pointer in argument points to the read string.    */
/*                    The function value is the length of the read string.  */
#include <stdio.h>
#include <stdlib.h>
#include <bios.h>

#define nbtrymax  200

void write_pdi(char *pwchr);            /* function writing to PDI 5025   */
int read_pdi(unsigned int *pointer);    /* function reading from PDI 5025 */
unsigned int port = 0;                  /* serial port 1 ! */

main()
{
   int i, nchar, imes=0;
   unsigned int etat;
   unsigned int *pointer;
   char dummy, endrun=0, flagA, *pchar;
   double valuea=0,valueb=0;

/* Initializes serial port 1 */
   if(!(_bios_serialcom(_COM_INIT,port,_COM_4800|_COM_NOPARITY|_COM_CHR8|_COM_STOP1) & 0x0030))
     {
       printf("serial port %d not initialized\n",port+1);
       exit(0);
     }
/* reserve memory for data read from PDI 5025 */
    if(!(pointer = (unsigned int *)malloc(20)))    /* 20 characters max. */
      {
      printf("memory for data not allocated\n");
      exit(0);
      }
```

MANRSE.C continued

```
/* reserve memory for string to be converted to double */
    if(!(pchar = (char *)malloc(20)))              /* 20 characters max. */
      {
      printf("memory for string not allocated\n");
      exit(0);
      }
/* reset (read) status 1 */
    do {  write_pdi("STB,1\r\n");
       }  while (!(nchar = read_pdi(pointer)));
/* reset (read) status 2 */
    do {  write_pdi("STB,2\r\n");
       }  while (!(nchar = read_pdi(pointer)));

/* set PDI 5025's parameters and check command error bit in status 1 */
    *(pointer+2) = '0';
    do {
         write_pdi("CHA,*\r\n");          /* both channel selected (default)    */
         write_pdi("SGA,A,5\r\n");        /* gain channel A                     */
         write_pdi("SGA,B,200\r\n");      /* gain channel B (if present)        */
         write_pdi("TRS,E,360\r\n");      /* trigger source : rotaional encoder */
         write_pdi("TRI,,0/30,12\r\n");   /* trigger sequence                   */
         write_pdi("MOT,A\r\n");          /* automatic start and stop for motor */
         write_pdi("IMD,1\r\n");          /* direct transfert                   */
         write_pdi("CUM,0\r\n");          /* individual values                  */
         do {  write_pdi("STB,1\r\n");    /* read status 1 */
            }  while (!(nchar = read_pdi(pointer)));
       } while (*(pointer+2) == '1');     /* if error try again */
/* initializes encoder index position */
    printf("initializes encoder position\n");
    write_pdi("DSP,,INDX\r\n");
    write_pdi("IND,+\r\n");
/* wait for synchro bit = 1 in status 1 */
    do {
         do {  write_pdi("STB,1\r\n");
            }  while (!(nchar = read_pdi(pointer)));
       } while (*(pointer+7) != '1');     /* synchro bit */
    printf("index found\n");
    write_pdi("DSP,,\r\n");
```

MANRSE.C continued

```
/* start run */
    printf("press ENTER to start run\n");
    scanf("%c",dummy);
    write_pdi("RUN\r\n");
    write_pdi("DSP,,RUN\r\n");
/* read status 1 */
    do {  write_pdi("STB,1\r\n");
       }  while (!(nchar = read_pdi(pointer)));
    if (*(pointer+4) == '1') endrun = 1;                /* end of run bit */
/* read data : one at a time */
    do {
/* wait for data ready bit = 1 in status 1 */
       do {
            do {  write_pdi("STB,1\r\n");
               }  while (!(nchar = read_pdi(pointer)));
            if (*(pointer+4) == '1') endrun=1;        /* end of run bit */
          }  while (*(pointer+5) != '1');             /* data ready bit */

/* read data until data ready bit = 0 in status 1 */
       while (*(pointer+5) == '1')
          {
            do {  write_pdi("ENQ\r\n");                /* ENQ command */
               }  while (!(nchar = read_pdi(pointer)));
/* convert characters into double value */
            flagA = 0;
            for (i=0;i<=nchar;i++)
               {
                 if (*(pointer+i)>='0'&& *(pointer+i)<='9')
                     *(pchar+i) = *(pointer+i);
                 else
                     *(pchar+i) = ' ';
                 if(*(pointer+i) == 'A' ) flagA = 1;
               }
            if (flagA)
               {
                 valuea = atof(pchar);
                 printf("%4d)\tCHA A = %10.0Lf\tCHA B = %10.0Lf  ( * 10-8 V.s)\n",
                        ++imes,valuea,valueb);
               }
            else
                 valueb = atof(pchar);
            do {  write_pdi("STB,1\r\n");               /* read status 1 */
               }  while (!(nchar = read_pdi(pointer)));
            if (*(pointer+4) == '1') endrun = 1;        /* end of run bit */
          }
       } while (endrun != 1);

    write_pdi("DSP,,\r\n");    /* default display */
    free(pointer);             /* free allocated memory */
}                              /* end of main */
```

MANRSE.C continued

```
/* function write_pdi : send command to PDI 5025 */
void write_pdi(char *pwchr)
{
  while ( *pwchr != '\0' )
     _bios_serialcom(_COM_SEND,port,*(pwchr++));
  return;
}

/* function read_pdi : read data or parameters from PDI 5025. */
/* the read string is put at the position "pointer".          */
/* the function value is the total number of characters read  */
int read_pdi(unsigned int *pointer)
{
   int nchar = 0,nbtry=0;
/* wait for data in serial port buffer */
   while( ((*pointer = _bios_serialcom(_COM_RECEIVE,port,0)) & 0xFF00)
       && (nbtry++ < nbtrymax));
   if (nbtry < nbtrymax) nchar = 1;
/* accept data until empty buffer */
   while(!((*(++pointer) = _bios_serialcom(_COM_RECEIVE,port,0)) & 0xFF00))
     nchar++;
   return nchar;        /* number of characters read */
}
```

PDI 5025 SUMMARY

Status Registers

	bit 7	6	5	4	3	2	1	0
	hex 80	40	20	10	8	4	2	1
1	STATUS 2 FLAG	Always 0	Command Error	Overrange Error	End of Run	Data Ready	Trigger	Synchro
2	0	0	FNC button	Power ON or Reset	Autotest failed	Encoder Count Error	Data Buffer Full	Overlap Timeout
3	<= T R S mode =>			Infinite Sequence	Run Active	Forwards or Backwards	Motor +	Motor -
4	<= V F C type =>		Overrange -	Overrange +	<= V F C type =>		Overrange -	Overrange +
	Channel B				Channel A			
5 & 6	0	0	0	Error in Analog Measure	OVR- Not ON => Error	OVR+ Not ON => Error	OVR- Not OFF => Error	OVR+ Not OFF => Error
7	0	0	0	Infinite Sequence Active	Run Active	Data Transfer Mode	Storage Mode	

Micro Switches

	IEEE-488		**RS232c**	
SW.	**"1" <=SWITCH POS.=> "0"**		**"1" <=SWITCH POS.=> "0"**	
1	DEVICE ADDRESS		BAUD RATE SELECTION	
2				
3				
4			8 bits	7 bits
5			2 stop bits	1 stop bit
6	Always "0"		Parity	No Parity
7	Always "0"		Even	Odd
8	NO "CR/LF"	"CR/LF"	XON/XOFF	No XON/XOFF
9	IEEE 488	--	--	RS 232 c
10	Talker Only	List./Talker	Autonomous	Conversational

Switches 3	2	1	BAUD RATE	
0	0	0	300	
0	0	1	600	
0	1	0	1200	
0	1	1	2000	
1	0	0	2400	
1	0	1	4800	
1	1	0	9600	limited
1	1	1	19200	use

Software Commands

SYNTAX	DE-FAULT	MNEMONIC	PAGES
CHA,A	***	CHAnnel	33,59
CHA,B			
CHA,*			
TRS,T	***	TRigger Source, Timer	37,59
TRS,T,S		TRigger Source, Timer, Synchro	37,59
TRS,E		TRigger Source, Encoder	39,59
TRS,E,S		TRigger Source, Encoder, Synchro	38,59
TRS,E,d..d		TRigger Source, rotational Encoder	37,59
TRS,X		TRigger Source, eXternal	39,59
TRS,X,S		TRigger Source, eXternal, Synchro	39,59
TRI,s,a/n1,C1/n2,C2/.../n20,C20		TRIgger sequence	35,60
TRI,?		Returns a string containing the current values of the sequence.	60
SGA,i,d..d		Set GAin	33,60
SGA,10	***	The default Gain at power on = 10.	
ADJ,i,1:		ADJust offset	34,60
ADJ,0 or ADJ,i,0		stop ADJust mode	
IND,s		INDex	37,38, 60
MOT,s		MOTor	51,61
MOT,S	***	MOTor, Stop	
MOT,A		MOTor, Automatic	
RUN		RUN	61
BRK		BReaK run	50,61
EOD,a1,..,an		End Of Data	50,61
CUM,0		storage mode,	48,62
CUM,1,S		CUMulative or	
CUM,1,L		individual	

SYNTAX	DE-FAULT	MNEMONIC	PAGES
IMD,1		IMeDiate or	48,62
IMD,0		block data trans.	49,62
ENQ		ENQuiry	62
FPT,n	***	Front Panel Trigger	39,63
STH,d		read STatus Hex.	41,63
STB,d		read STatus Bin.	
DSP,i,xxxx		send char. to DiSPlay	63
VER		software VERsion	63
CVR,i		Clear oVeRrange	63
RGA,i		Read GAin	63
RCT		Read CounTer	64
ZCT		Zero CounTer	64
NBO,0		Not Brk	64
NBO,1		Overrange	
TST		perform TeST	51,64
AUT		read AUTo test data	51,64
ISC,i,0		Input Short Circ.	52,65
ISC,i,1			
FNC,1		button	33,65
FNC,0		FuNCtion	
LLO,0		Local LOckout	65
LLO,1		RS 232 C only	
MSK,a,oo		MaSK	55,65
		IEEE 488 only	
SYN		SYNchronization	53,66
		RS 232 C	
SYN,1		SYNchronization	53,66
SYN,0		IEEE 488	