Magnetic precision has a name

Fast Digital Integrator FDI2056

User's Manual

Version 2.0

(Revision 1.2)

May 2014

REVISION HISTORY

v. 1.0 r. 1.0	June 2010	First release
v. 1.0 r. 1.1	June 2010	Update installation procedure
		Improve readability of screen-shots
		Correct section numbering in Chapter 4
		Correct formatting of specifications table
v. 1.1 r. 1.0	June 2010	Update for version 1.1 software
v. 1.1 r. 1.1	July 2010	Clarify Windows installation procedure
v. 1.1 r. 1.2	April 2011	Document Reset register
v. 2.0 r. 1.0	April 2012	Update for version 2.0 hardware and software
v. 2.0 r. 1.1	March 2013	SCPI commands added
v. 2.0 r. 1.2	May 2014	Fix front page graphics
		Technical specifications updated

CONTENTS

GETTING STARTED
1-Introduction

The Fast Digital Integrator FDI2056 is a high precision, high-speed voltage integrator, developed at the European Organization for Nuclear Research (CERN). Combined with an appropriate flux coil, the FDI2056 makes an extraordinarily flexible, fast, and sensitive magnetometer.

The CERN's original objective was to perform high-precision multi-pole measurements of dynamic field effects, in conjunction with a fast rotating-coil system. The FDI2056 is, however, a general-purpose instrument, capable of measuring low-level and rapid flux changes in all imaginable coil configurations: rotating coil, moving-coil, flip-coil, moving wire, static coil in an AC field, etc.

The FDI2056 communicates to any host computer using an ethernet VXI compliant protocol by default or using a serial PDI-5025 protocol when configured as such.

The software interface to the instrument is provided via the NI-VISA library. VISA (Virtual Instrument Software Architecture) is an industry standard, with NI-VISA being an implementation from National Instruments. VISA is supported by all major instrumentation manufacturers, and NI-VISA provides excellent cross-platform compatibility (Windows, Macintosh or Linux).

It is easy to develop custom software for the FDI2056, especially in LabVIEW, using Metrolab's FDI2056 Application Programming Interface (API).

Finally, keep your FDI2056 accurate and up to date by having it recalibrated at regular intervals. The recommended calibration interval is every 12 months. At this time, Metrolab will also install the latest available firmware.

You can also download the latest software and manual, free of charge. We post all updates on our website. The easiest way to be notified of updates is to sign up for our electronic newsletter, published twice a year; please see the Newsletter Subscription page of the News section of Metrolab's website, www.metrolab.com.

We hope the FDI2056 will help you perform your magnetic field measurements easily and accurately. If you have problems and your retailer cannot help you further, the Metrolab team is ready to help. Even if you don't have problems, we are always interested in knowing more about how our instruments are used. Feel free to contact us at any time at contacts@metrolab.com.

GETTING STARTED
2-Installation Guide

This chapter provides information regarding the various connections available on the instrument and other internal configurations available to the user.

2-1 FDI 2056 HOST INTERFACE PANEL AND CONNECTIONS

The DVI (Digital Visual Interface) is used to connect the host interface to a display device, such as a computer monitor.

The USB host interface is used to connect devices such as a keyboard, a mouse, etc.

Ethernet connectors: Connector number 2 must be used to connect the instrument to your host computer using a crossover cable or a straight cable to a switch or a hub.

The two LEDs signal the following information :

Top LED	Bottom LED
Off : No link Orange : 1 Gb Link Green : 100 Mb Link	Off : No Link Green blinking : activity

Three LEDs are present on the interface front panel.

- The LED named PG (Power Good) is green when the system's power is functioning as expected. When off, the system is sleeping. When the color of this LED turns to steady red or blinking continuously, it indicates a major failure.
- The GP LED is not dedicated to any particular hardware or firmware function. Nevertheless, a red blinking GP LED indicates a major failure.

- The LED named HD (Hard Drive) blinks each time an access is made to the internal hard drive or more generally, when any access is made to the SATA port.

2-2 FDI2056 FRONT PANEL AND CONNECTIONS

The 4-character display indicates the FDI2056's current gain or state.

The signal amplitude can be monitored on the LED bar-graph. The effect of changing the gain will be immediately visible. If the +OVR and –OVR lights come on, the measurement can be considered questionable.

Encoder input connector and LED: When using a rotational or linear encoder in your measurement system, this input will let you generate trigger signals which can be used to accurately synchronize your measurements to your mechanical system.

See section 2-2-1 for the connector pinning description.

The coil can be connected to the FDI2056 either via the two LEMO 00 connectors, "IN+" and "IN-", or via the single LEMO 0B connector.

The FDI2056 is supplied with a 10 m cable with a LEMO 0B connector. The FDI2056 coil input characteristics can be adapted for your application.

See Section 5-2 for details.

The "REF. CLK 20MHz" allows the FDI2056 time base to be checked with a precision counter. Please note that the reference oscillator is the one located on the board plugged into slot 2.

This yellow LED blinks each time a trigger event is internally generated when making a measurement.

The trigger "IN" is a 3.3V, 5V-tolerant TTL input. The software-selected edge of the Trigger "IN" signal determines when partial integrals are computed. Each trigger pulse ends the previous partial integral (if there was one) and starts a new one. Thus, a sequence of N partial integrals requires N+1 trigger pulses. The rise times should be as fast as possible to minimize timing jitter.

The trigger "OUT" signal allows you to monitor the activity of the trigger signal being used internally.

Please observe that the actual timing used in the system is not identical to the one that can be observed on this output. Using this signal to daisy-chain all boards together would result in measurements being very poorly synchronized. The system provides an internal channel synchronization designed for that purpose.

! CAUTION

⇒ The FDI2056 front-panel connectors are NOT protected against over-voltages or electrostatic discharge. Please be very careful, and verify your signal levels on an oscilloscope when making your connections.

2-2-1 *Description of the encoder input Micro-D connector.*

The FDI2056 is supplied with a 50-cm cable with a Micro-D connector. The color in the table below refers to this cable.

Pin n°	Cable color	Function
1	Black	$\overline{B}$
2	Brown	B
3	Red	$\overline{A}$
4	Orange	A
5	Yellow	5 [Volts] or 3.3 [Volts] (Factory default 5 [Volts]) A re-armable safety fuse prevents the system from providing more than 0.750 [Amps].
6	Green	Ground
7	Blue	Error Input

8	Violet	$\overline{\text{Index}}$
9	Grey	Index

The shell is connected to the chassis ground.

When using a single-ended encoder, all signals must be connected to the active high version of the signals (A, B, Index). When using differential signals, you must use the various pairs provided, which are then fed to a "differential to single ended driver". The output polarity of the driver follows the polarity of the active high signal (i.e., if A goes high and $\overline{A}$ goes low, the output of the driver will go high).

As stated in the table, it is possible to change the voltage used to power the encoder. A full step-by-step explanation describing this operations is available in the "Getting Started Manual".

2-3 COMMUNICATING WITH THE FDI2056

The FDI2056 communicates to any host computer using an Ethernet VXI compliant protocol by default or using a serial PDI 5025 protocol when configured as such.

2-4 INITIAL POWER-ON

1. After the FDI2056 is powered on, the message "INIT" will be displayed on all installed channels. This indicates that the FDI2056 is resetting. Once powered on, the FDI2056 takes several seconds before being able to communicate. Please remember that you must respect a warm-up time of at least 10 minutes before making any measurements. A significant drift might be measured if this precaution is not taken.

2. The first operation performed by the instrument is an internal status check of every integrator channel. During this operation, the text "TEST" is displayed on every tested channel. Should the instrument detect any problem, one of the following message could be displayed on the faulty channel: "!MEM", "!INP", "!CAL", "!COM", "!HWU". Actions that could be undertaken if such a situation should arise are described in chapter 4-11-9.

3. The "Status" indicator will then show a numerical value reflecting the current gain setting. Should the first channel display "A 10", this would indicate that the PDI 5025 emulation mode has been selected.
4. As long as the computer is internally in reset mode, the bright blue light at the right (bottom) of the FDI2056 front panel stays lit.

2-5 SOFTWARE DEVELOPMENT

- Insert the installation CD.
- Copy the Source folder to your hard drive. The API subfolder contains the Application Programming Interface.
- Modify the FDI2056 measurement software, or write a measurement system from scratch using the LabVIEW development system.

PROGRAMMING THE FDI2056
3-VXI & SCPI standard interfaces

3-1 HOST INTERFACE

The FDI2056 and its associated host software support an Ethernet interface. The native host interface protocol is based on the following standards, very widespread in the instrumentation industry:

- VXI-11: Ethernet device control,

- SCPI: command structure.

The native commands provide access to all the features of the FDI2056.

3-2 SCPI INSTRUMENT MODEL

The FDI2056 complies with the Standard Commands for the Programmable Instruments (SCPI) standard. SCPI uses a standard instrument model to organize the command structure. The diagram below shows the subsystems concerning the signal flow.

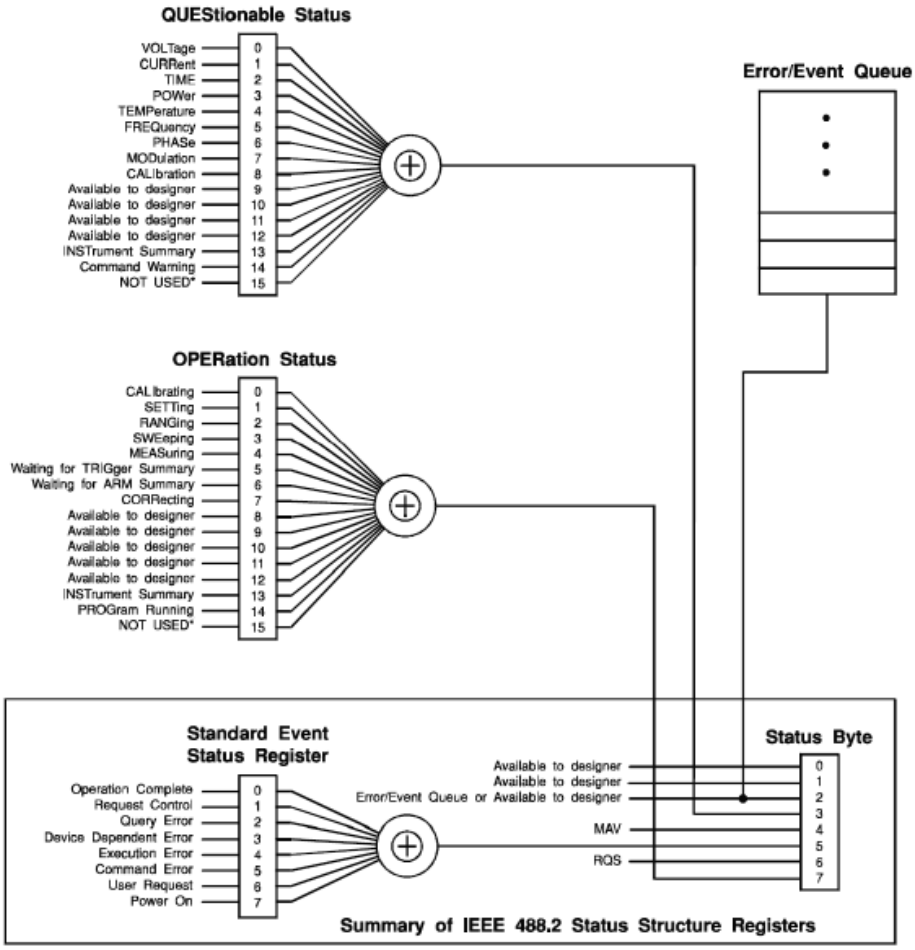
The following table provides a summary of the FDI2056 capabilities, organized according to the SCPI command reference – SCPI 1999, Volume 2 -. The supported commands include IEEE 488.2 "Common Commands" (start with "*") as well as SCPI commands – see Sections 3-5 through 3-7 for details. In addition to ASCII commands, the FDI2056 also supports certain VXI-11 commands, also noted in this table.

Functional Block	Command(s)	Function
Measurement Function	:MEASure …	Measure with standard settings. Equivalent to *RST;:READ …
	:READ …	Measure with current settings. Equivalent to :ABORt;:INITiate;:FETCh? …
	:FETCh …	Fetch measurement results previously acquired with READ or INITiate
CALCulate	:CALCulate…	Cumulative or partial-integrals mode.
CALibration	-	Not used in the FDI2056
CONTrol	:CONTRol	Quadrature signal decoder configuration
DIAGnostic	:DIAGnostic: …	Initiate firmware upgrade
DISPlay	:DISPlay	4-Digit display management

FORMat	:FORMat …	Set output format
HCOPy	-	Not used in the FDI2056
INPut	:INPut …	Configure gain and coupling
INSTrument	-	Not used in the FDI2056
MEMory	:MEM	Store / Delete user settings
MMEMory	-	Not used in the FDI2056
OUTput	-	Not used in the FDI2056
PROGram	-	Not used in the FDI2056
ROUTe	-	Not used in the FDI2056
SENSe	:SENSe …	Internal correction and measurement mode
SOURce	-	Not used in the FDI2056
STATus	*STB?, *SRE	Read / enable bits in Status Byte
	*ESR?, *ESE	Read / enable bits in Standard Event Status Register
	*OPC, *WAI	Detect and wait for operation to be complete
	:STATus …	Read / enable bits in OPERation and QUEStionable registers
	:SYSTem:ERRor…	Query error queue
SYSTem	*RST	Perform reset
	*TST?	Perform self-test
	*IDN?	Return Instrument ID
	*CLS	Clear status
	:SYSTem:VERSion	Return SCPI version
	:SYSTem:HELP …	Provide command help
TEST	-	Not used in the FDI2056
TRIGger	*TRG	Generate a trigger
	:ARM …	Arm Trigger signal
	:INITiate …	Enable triggers
	:ABORt	Abort triggers
	:TRIGger	Trigger configuration
UNIT	:UNIT:FLUX :UNIT:VOLTage :UNIT:TIMestamp	Specifies the fundamental unit of flux, voltage and timestamp.
VXI	-	Not used in the FDI2056

3-3 IEEE 488.2 / SCPI STATUS REGISTERS

IEEE 488.2 compliant instruments have at least two registers: the Status Byte and the Standard Event Status Register. SCPI adds the Operation Status Register, Questionable Status Register and Error/Event Queue. The diagram below, taken from the SCPI standard, provides a good summary. This section describes how the FDI2056 uses these status registers.



* The use of Bit 15 is not allowed since some controllers may have difficulty reading a 16 bit unsigned integer. The value of this bit shall always be 0.

3-3-1 *Status Byte*

- Contains a 1-byte status summary. The FDI2056 uses the following bits:

Bit	Name	Description
2	EAV	Error AVailable (in Error/Event Queue)
3	QSB	Questionable Summary Bit
4	MAV	Message AVailable: response ready to be read
5	ESB	Event Summary Bit
6	RQS	ReQuest for Service

Bit	Name	Description
7	OSB	Operation Summary Bit

3-3-2 *Standard Event Status Register*

- Memorizes certain standardized events. The FDI2056 uses the following bits:

Bit	Name	Description
0	Operation Complete	*OPC has flagged operation complete
2	Query Error	Error in preceding query
3	Device Dependent Error	Errors specific to the FDI2056, including internal errors
4	Execution Error	Error detected during command execution
5	Command Error	Error in preceding command
6	User Request	A parameter has been changed
7	Power On	FDI2056 has been powered up

3-3-3 *QUEStionable Status*

- Indicates conditions that may reduce the quality of the measurement. The FDI2056 sets the following bits:

Bit	Name	Description
0	Voltage	The input voltage generated an over-range.
8	Calibration	The recommended calibration period has been exceeded.
9	Trigger	Trigger signal is too fast.
10	Integration	A calculation overflow occurred in the integration subsystem.
11	Encoder	The pulse count at the index is not correct.

3-3-4 *OPERation Status*

- Captures conditions which are part of the instrument's normal operation. The FDI2056 uses the following bits:

Bit	Name	Description
4	MEASuring	Measuring voltage or flux
5	Waiting for TRIGger	Waiting for trigger

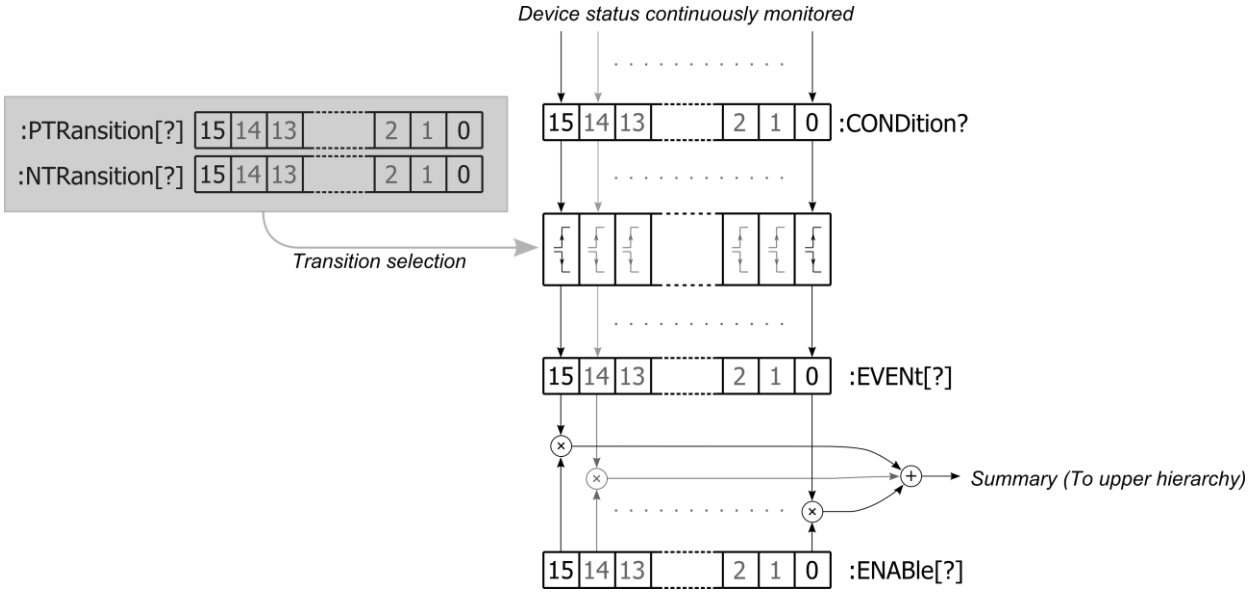
Bit	Name	Description
6	Waiting for ARM	Waiting for arm
7	CORRecting	Correcting the offset or the slope
8	Parameters	Parameter has changed
9	Data available	Internal data buffers are not empty
10	Encoder index	Index has been detected

3-3-5 *OPERation:PARameters*

- This is a fan-out for bit 8 of the OPERation register, a device-specific summary bit. It indicates subsystems whose configuration has changed.

Bit	Subsystem name
0	ARM
1	CALCulate
2	CONFigure
3	CORRection
4	DISPlay
5	FORMat
6	INPut
7	SENSe
8	STATus
9	SYSTem
10	TRIGger
11	UNIT
12	MEMory

As shown in the figure below, taken from the IEEE 488.2 standard, each of the registers above is actually a set of three registers:



- Condition Register (CONDition)
  Read-only register that is constantly updated to reflect the current state of the instrument.
- Event Register (EVENt)
  Transitions in a Condition Register are memorized in the corresponding Event Register. Most FDI2056 registers only latch transitions from 0 to 1, some are configurable using the Positive and Negative Transition Register set. Event Registers are cleared when read.
- Event Enable Register (ENABle)
  This is a mask indicating what bits in the Event Register are included in the Summary bit. The enable mask of the Status Byte is called the Status Enable register, and it determines which bits cause an RQS (ReQuest for Service).

Note that some Condition Register bits signal permanent conditions – for example Power On – and others signal transient events – for example, Command Error. Reading a Condition Register will provide no indication of transient events; only the corresponding Event Register will record such events.

Also note that for efficiency reasons, status register updates are deferred to the extent possible, and are normally performed only at the end of each sequence of commands separated by semicolons. This also means that a ReQuest for Service (RQS) is normally generated only at the end of such a sequence. There are exceptions to this deferral rule, for example if the command sequence includes a status-query command.

.

3-4 IEEE 488.2 CONTROLS

The following low-level functions are defined by the VXI-11 protocol. Excluded are the functions that are fundamental to the operation – e.g. in VXI-11: create_link, destroy_link, device_write, device_read, create_intr_chan, destroy_intr_chan, device_enable_srq, device_intr_srq, and device_abort.

Historically, the functions listed below correspond to dedicated hardware signals in IEEE 488.1 (HPIB or GPIB). Not all functions are supported in VXI-11. In addition, some of these functions are not supported on the FDI2056; the Description column will note whether the function has no effect or whether it returns an error.

VXI-11	Description
device_clear	Clears the device input and output buffers
device_trigger	Assert bus trigger
	Requests service from host
device_readstb	Read status byte
device_remote	Remote Enable – no effect
device_local	Enable local controls – no effect
	Disable local controls – no effect
device_lock	Acquire or release the device's lock
device_unlock	
device_docmd	Do special command – unsupported, returns error

3-5 IEEE 488.2 COMMON COMMANDS

As for any IEEE 488.2 compliant instrument, the FDI2056 supports the following commands.

Command	Name	Description
*CLS	Clear status	Clear all event registers and queues (not enable registers) and error buffer
*ESE <value>	Program event enable	Program standard event enable register
*ESE?	Event enable query	Read standard event enable register
*ESR?	Event status query	Read standard event register and clear it

Command	Name	Description
*IDN?	Identification query	Returns the following information: manufacturer; model; serial number; and version of electronics and firmware. Note that this query returns "Arbitrary ASCII Response Data" (see IEEE488.2 standard) and cannot be followed by another query in the same command sequence.
*OPC	Set operation complete	Set the operation complete bit in the standard event register after all commands have been executed
*OPC?	Operation complete query	Returns an ASCII "1" after all commands have been executed
*RST	Reset	Reset device to power-on configuration
*SRE <value>	Program status enable	Program status enable register. Important: you must also enable service requests on the host. See Section 3-8 for details.
*SRE?	Status enable query	Read status enable register
*STB?	Status byte query	Read status byte register
*TRG	Trigger	Generate bus trigger
*TST?	Self-test Query	Perform complete self-test, return 0 if successful, 1 if not
*WAI	Wait-to-Continue	Wait until previous commands have completed

3-6 SCPI COMMAND SYNTAX

In the command definitions below, the following conventions are used:

[ ] optional keywords or parameters

< > value

channel number. When omitted, the command assumes all channels to be selected.

| Each optional value is separated by this symbol.

The abbreviated form of each command is written in capital letters. For example, the "MEASure" command can be written as "MEASURE" or "MEAS", or, since capitalization doesn't matter, "measure" or "meas".

The following special parameters are recognized by most commands

MINimum, MAXimum, DEFault and OPTions

The command syntax

<command>? MAXimum | MINimum[,…]

can be used to query the maximum or minimum values for all parameters of a command.

Command parameters may also be set to their default values by omitting them. To insert a parameter after an omitted parameter, be sure to maintain the commas as placeholders.

Numeric parameters sometimes require units. Analogously, the values returned by queries contain units, as specified by the UNIT command. In addition, some units can have prefixes:

N = nano (10^{-9})

U = micro (10^{-6})

M = milli (10^{-3})

K = kilo (10^{3})

MA = mega (10^{6})

G = giga (10^{9})

The FDI2056 recognizes the following units:

Base unit	Multiplier	Description
WB	M, U, N	Weber
S	M, U, N	Seconds
V	M, U, N	Volts
Hz	G, MA,K	Hertz

However, it is possible to add specific units which are directly related to the flux measurement. As an example, if one knows the area of a coil, it may add a Tesla unit by specifying the multiplying factor needed to convert one Weber [Wb] to one Tesla [T].

3-7 SCPI COMMAND DESCRIPTION

Commands either set of query values. Some commands both set and query, some only set, and some only query. Particular commands might generate internal events.

This section follows these conventions:

- No query form exists for commands identified as "Set only".

- A question mark (?) appended to the command and "Query only" indicates query-only commands.
- A command generating an internal event is identified as "Event".
- All commands are listed in alphabetic order, excluding IEEE 488.2 commands starting with an asterisk (*) which are described in section 3-5 and excluding STATus command which are described in section 3-3.

3-7-1 *:ABORT (Event)*

Stops the execution of a command

Syntax `ABORT`

Example `ABORT`

3-7-2 *:ARM#[:SEQuence][:LAYER]:SOURce*

Selects the source for the arm event detector. Only one source may be specified at a time.

Syntax `ARM#:SOURce IMMediate|SYNChro|ENCoder|EXTernal|INTernal|TIMer|BUS`

`ARM#:SOURce? [DEFault|OPTions]`

Default `IMMediate`

Arguments

`IMMediate` Specifies an immediate exit of the arm state.

`SYNChro` Specifies that the arm source is fed by the synchronization signal found on the encoder connector. The slope to be used is configured using the command `ARM#[:SEQuence][:LAYER]:SYNChro[:POLarity]`.

`ENCoder` Specifies that the arm source is fed by the signal extracted from the encoder. The configuration of this extraction is made using the command `CONTRol#:ENCoder:CONFigure`.

`EXTernal` Specifies that the arm source is fed by the trigger-in signal found on the front panel. The slope to be used is configured using the command `ARM#[:SEQuence][:LAYER]:EXTernal[:POLarity]`.

`INTernal` Specifies that the arm source is fed internally. Used only on an instrument having multiple channels which need to work synchronously.

`TIMer` Specifies that the arm source is fed by the internal periodic signal source. The rate at which this source ticks is set using the command `TRIGger#[:SEQuence]:TIMer[:RATE]`.

`BUS` Specifies that the arm source is software triggered. The command generating this trigger is `*TRG`.

Example `ARM:SOUR IMM`

Sets the arm event to IMMediate.

`ARM:SOUR? OPTions`

The above command will return the following string:

`IMMediate|SYNChro|ENCoder|EXTernal|INTernal|TIMer|BUS`

3-7-3 *:ARM#[:SEQuence][:LAYER]:ENCoder*

Sets the encoder position at which the arm detection layer is satisfied, letting the system enter the trigger layer.

Syntax `ARM#:ENCoder numerical value (no units)`

`ARM#:ENCoder? [DEFault]`

Default 0

Arguments

The *numerical value* represents the absolute position of the encoder expressed as a number of internal pulses. Four (4) pulses are generated for every encoder line. A 1024-line encoder will generate 4096 internal pulses.

Example `ARM:ENCoder 127`

As soon as the absolute position 127 is reached, the system will instantly process the trigger-related instructions.

3-7-4 *:ARM#[:SEQuence][:LAYER]:SYNChro[:SLOPe]*

Configures the transition detector of the synchronization signal to generate an arm-condition on a specific slope.

Syntax `ARM#: SYNChro POSitive|NEGative|EITHer|NONe|DEFault`

ARM#:SYNChro? [DEFault|OPTions]

Default NONe

Arguments

POSitive A transition from low to high of the synchronization signal found on the encoder connector is used to leave the arm level.

NEGative A transition from high to low of the synchronization signal found on the encoder connector is used to leave the arm level.

EITHer Is a conjunction of both POSitive and NEGative conditions.

NONe Disables the synchronization function to the benefit of the error signal that might be found on some encoders.

Example ARM:SYNC POS

Configures the transition detector to generate an event when a low to high condition occurs on the synchronization signal.

3-7-5 *:ARM#[:SEQuence][:LAYER]:EXTernal[:SLOPe]*

Configures the transition detector of the external input trigger signal to generate an arm-condition on a specific edge.

Syntax ARM#:EXTernal POSitive|NEGative|EITHer|DEFault

ARM#:EXTernal? [DEFault|OPTions]

Default POSitive

Arguments

POSitive A transition from low to high of the synchronization signal found on the encoder connector is used to leave the arm level.

NEGative A transition from high to low of the synchronization signal found on the encoder connector is used to leave the arm level.

EITHer Is a conjunction of both POSitive and NEGative conditions.

Example TRIG:EXT POS

Configures the transition detector to generate an event each time a positive transition occurs on the external input trigger signal.

3-7-6 *:CALCulate#:FLUX[:CUMulative]*

Enables or disables the accumulation of partial integrals values.

Syntax `CALC#:FLUX[:CUMUlative] Boolean|DEFault`

`CALC#:FLUX[:CUMUlative]? [DEFault]`

Default `OFF`

Arguments

0 Disables the accumulation.

1 Enables the accumulation.

Example If a constant voltage of 1 volt is fed to the system configured to measure a partial integral every 1 second, every partial integral will evaluate to 1 [WB]. Then, issuing the command

`CALC:FLUX 1`

will result in the system reporting for the three first value :
`…; 1.000e0 WB, …; 2.000e0 WB, …; 3.000e0 WB, ….`

If the accumulation mode is unset (`CALC:FLUX 0`), the system will report for the first three values :
`…; 1.000e0 WB, …; 1.000e0 WB, …; 1.000e0 WB, ….`

3-7-7 *:CALCulate#:TIMestamp[:CUMulative]*

Enables or disables the accumulation of the timestamp.

Syntax `CALC#:TIMestamp[:CUMUlative] Boolean|DEFault`

`CALC#:TIMestamp[:CUMUlative]? [DEFault]`

Default `OFF`

Arguments

0 Disables the accumulation.

1 Enables the accumulation.

Example Assuming that the internal time trigger is set at 1 [kHz], a trigger will take place every 0.001 second. Then, enabling the timestamp accumulation by issuing the following command

`CALC:TIM 1`

will result in the system reporting for the first three timestamp values :
`1.00e-3 S;…, 2.00e-3 S;…, 3.00e-3 S;…, ….`

If the accumulation mode is unset, the system will report for the first three values :
`1.00e-3 S;…, 1.00e-3 S;…, 1.00e-3 S;…, ….`

3-7-8 *:CONTRol#:ENCoder:CONFigure*

Configures how the quadrature signals fed to the front panel connector must be decoded.

Syntax `CONTRol#:ENCoder:CONFigure "<Configuration>"`

`CONTRol#:ENCoder:CONFigure? [DEFault]`

Default `"SINGle,A:B:INDex:/ERRor,ROTational:1024"`

Arguments

The <Configuration> quoted string is made of three different fields separated by a comma (,).

First field : `SINGle|DIFFerential`

Specifies if the encoder provides differential or single-ended signals. An internal differential to single-ended buffer is located in the system and is enabled when needed.

Second field : `A|/A:B|/B[:INDex|/INDex][:ERRor|/ERRor]`

The slash symbol (/) preceding the signal name states that the signal must be negated before use. If either or both of the `INDex` / `ERRor` signals are omitted, the configuration will consider these signals as absent and will behave accordingly.

Third field : `ROTational|LINear:<numerical value>`

Specifies if a `LINear` or a `ROTational` encoder is used. The numerical value specifies the number of lines found on the encoder. Please note that a 1024 lines encoder will internally be interpreted as a 4096 pulses encoder.

Example `CONTR:ENC:CONF "DIFF,A:/B:IND,ROT:256"`

This will configure the system to decode the signal fed from the encoder connector using a differential to single-ended circuitry. Once this adaptation is made, the A signal is used directly, the B signal is inverted before use. The encoder provides an index signal which is used directly. No error signal is present in the system. The encoder is rotational and will generate 256 pulses per turn.

It is to be noted that
- using a differential system, physically permuting a pair of signal
or
- setting the same pair to be inverted by software
is the same operation.

Notes When the decoder is configured to use an index, the first time the index is reached the internal counter keeping track of the current position will be set to 4*<numerical value> -1 or 0 depending on the rotational direction. The value of this counter is available by using the command `CONTR#:ENC:POS?`

When configuring the FDI2056 to use an encoder, it is of upmost importance to check that the rotational direction is in accordance with the two LEDs denoted FW and BW located on the front panel of the channel into which the encoder is plugged.

3-7-9 *:CONTRol#:ENCoder[:POSition]*

Used to set the zero of a linear or rotational encoder having no index. When used as a query, this command returns the current position of the encoder.

Syntax `CONTRol#:ENCoder[:POSition] <numerical value>`

`CONTRol#:ENCoder[:POSition]?`

Arguments

The <numerical value> is used to set the current position of the encoder.

Example `CONTR:ENC?`

Returns the current position of the encoder.

3-7-10 *:DATA#:COUNt? (Query only)*

Returns the amount of data available in the instrument fifo. This value may be directly used to fetch this amount of remaining data.

Syntax `DATA#:COUNt?`

Example `DATA:COUN?`

Returns the amount of data available in the instrument fifo. This value may be equal to zero.

3-7-11 *:DIAGnostic:UPGRade:[INITiate]* *(Event)*

Initiates a firmware upgrade.

Syntax `DIAGnostic:UPGRade[:INITiate]`

NOTICE
The :DIAGnostic:UPGRade:[INITiate] command is intended for use by the manufacturer only. It can cause your FDI2056 to become nonoperational.

3-7-12 *:DISPlay#:TEXT[:DATA]*

Displays some text on the 4-digit display located on the channel front panel.

Syntax `DISPlay#:TEXT[:DATA] <string>`

`DISPlay#:TEXT[:DATA]?`

Arguments

string Only the first four (4) ASCII characters will be displayed. Each time the system needs to write on the display, the user value is withdrawn.

Example `DISP:TEXT HELLO`

Displays HELLO on every instrument's display.

3-7-13 *:FETCh#:ARRay?* *(Query only)*

Fetches data values acquired during last `READ` or `INITiate`.

Syntax `FETCh#:ARRay? <size>[,<digits>]`

Default `digits = 6`

Arguments

size The amount of data being fetched into the system.

digits The number of digits that must be returned to the user.

Example `FETCh:ARR? 1023,6`

Read 1023 values, each having 6 digits.

Related commands

FORMat#:DATa

FORMat#:TIMestamp[:ENABle]

FORMat#:READings:ALL

FORMat#:UNIT[:ENABle]

UNIT#:FLUX

UNIT#:VOLTage

UNIT#:TIMestamp

3-7-14 *:FORMat#[:DATa]*

Sets format for returned measured data.

Syntax FORMat#[:DATa] ASCii|INTeger|DEFault

FORMat#[:DATa]? [DEFault|OPTions]

Default ASCii

Arguments

ASCii User-readable output.

INTeger Returns an IEEE488.2 definite-length block. The block starts with a sharp sign (#) followed by a 1-digit number ranging from 1 to 9. This number specifies how many digits the number which follows contains.

Example

#3123 means a number having 3 digits. That number is 123.

#15 means a number having 1 digit. That number is 5.

#123 means a number having 1 digit. That number is 2, the 3 which follows is part of the data.

Encoded data are 32-bit little-endian floating-point values. When timestamps are enabled, data are formatted by pairs constituted by a timestamp followed by a measured value. If timestamps are disabled, all data are measured values. Enabling or disabling the timestamp is controlled by the command FORMat#:TIMestamp[:ENABle].

Related commands

FETCh#:ARRay?

READ#?

MEAS#:VOLT?

FORMat#:TIMestamp[:ENABle]

```
SENSe#:FUNCtion
```

3-7-15 *:FORMat:TIMestamp[:ENABle]*

Determines if the timestamp must be sent when reading measurement results.

Syntax `FORMat:TIMestamp[:ENABle] <Boolean>|DEFault`

`FORMat:TIMestamp[:ENABle]? [DEFault]`

Default 1

Arguments

0	Disables the timestamp readout.
1	Enables the timestamp readout.

3-7-16 *:FORMat:READings:ALL*

Determines the behavior of the readings when using commands addressing multiple channels.

Syntax `FORMat:READings:ALL <Boolean>|DEFault`

`FORMat:READings:ALL? [DEFault]`

Default 1

Arguments

0	Only data coming from channel 1 will be read out. A warning is sent to the user if the value is not the same on every channel.
1	All values are read back, each value is preceded by the string "CH<Number>:", <Number> refers to the number of the channel.

Example

```
FORM:READings:ALL 0; INP:GAIN?
```

This command might return

```
10\n
```

```
FORM:READings:ALL 1; INP:GAIN?
```

This command might return

```
CH1:10, CH2:100\n
```

3-7-17 *:FORMat:UNIT[:ENABle]*

Determines if the measurement results must be followed by the unit.

Syntax `FORMat#:UNIT[:ENABle] <Boolean>|DEFault`

`FORMat#:UNIT[:ENABle]? [DEFault]`

Default `1`

Arguments

0	Disables the unit readout.
1	Enables the unit readout.

Example `FORM:UNIT 0; :READ:VOLT?`

This command might return

`0.567 \n`

`FORM:UNIT 1; :READ:VOLT?`

This command might return

`0.567 V \n`

3-7-18 *:INPut#:COUPling*

Defines how the input signal is fed to the sense system.

Syntax `INPut#:COUPling VREF|DC|GND|DEFault`

`INPut#:COUPling? [DEFault|OPTions]`

Default `GND`

Arguments

VREF	The internal voltage reference associated with the current gain is fed to the sense system.
DC	Selects a DC coupling.
GND	Short-circuits the input.

Example `INP:COUP DC`

Direct coupling of the input.

3-7-19 *:INPut#:GAIN*

Defines which gain is used to amplify the input signal.

Syntax `INPut#:GAIN MINimum|MAXimum|UP|DOWN|DEFault|<value>`

`INPut#:GAIN? [DEFault|OPTions]`

Default `0.1`

Arguments

`MINimum`	Sets the input gain to its minimal value.
`MAXimum`	Sets the input gain to its maximal value.
`UP`	Increases the gain using the next valid value.
`DOWN`	Decreases the gain using the next valid value.
`value`	Valid values are 0.1, 0.2, 0.5, 1.0, 2.0, 4.0, 5.0, 10.0, 20.0, 40.0, 50.0, 100.0. Any other value will generate an error.

Example `INP:GAIN 10`

Sets the gain of all channels to 10.

3-7-20 *:MEASure#[:SCALar]:VOLTage? (Query only)*

Instantly measures a single non-averaged voltage value.

Syntax `MEASure#[:SCALar]:VOLTage? [<digits>]`

Default `digits = 6`

Arguments

`digits`	The number of digits that must be returned to the user.

Example `MEAS:VOLT? 3`

This command might return

`0.123 V \n`

Related commands

`FORMat#:DATa`

`FORMat#:READings:ALL`

`FORMat#:UNIT[:ENABle]`

`UNIT#:VOLTage`

3-7-21 *:MEMory :STORe[:STATe] (Event)*

Store the current user's settings in local hard-drive.

Syntax `MEMory:STORe[:STATe]`

Example `MEM:STOR`

Store the current system state into the local hard-drive

Related commands

`MEMory:DELete`

`*RST`

`SYSTem:PRESet`

3-7-22 *:MEMory :DELete (Event)*

Deletes the current user's settings from local hard-drive.

Syntax `MEMory:STORe[:STATe]`

Example `MEM:DEL`

Delete all user's settings, keeping only factory defaults.

Related commands

`MEMory:STORe[:STATe]`

`*RST`

`SYSTem:PRESet`

3-7-23 *:READ#:ARRay? (Query only)*

Read data values. Equivalent to the sequence :ABORt;:INIT;:FETCh:ARR?

Syntax `READ#:ARRay? <size>[,<digits>]`

Default `digits = 6`

Arguments

`size` The amount of data being read into the system.

`digits` The number of digits that must be returned to the user.

Example `READ:ARR? 555,6`

Read 555 values, each having 6 digits.

Related commands

FORMat#:DATa

FORMat#:TIMestamp[:ENABle]

FORMat#:READings:ALL

FORMat#:UNIT[:ENABle]

UNIT#:FLUX

UNIT#:VOLTage

UNIT#:TIMestamp

3-7-24 *:SENSe#:CORRection:ALL[:ACQuire]* *(Event)*

Proceeds to an offset and slope correction over all available gain on a channel. When proceeding to the offset correction, the input is not short-circuited. It's up to the user to proceed to that operation if needed.

Syntax SENSe#:CORRection:ALL[:ACQuire]

Example SENS:CORR:ALL

Automatically proceeds to an offset and slope correction over all available gain.

Related commands

INPut#:COUPling

SENSe#:CORRection:MEMorized

3-7-25 *:SENSe#:CORRection:ZERo[:ACQuire]* *(Event)*

Proceeds to an offset correction on a channel. When proceeding to the offset correction, the input is not short-circuited. It's up to the user to proceed to that operation if needed.

Syntax SENSe#:CORRection:ZERo[:ACQuire]

Example SENS:CORR:ZER

Starts the offset correction.

Related commands

INPut#:COUPling

SENSe#:CORRection:MEMorized

3-7-26 *:SENSe#:CORRection:SLOPe[:ACQuire] (Event)*

Proceed to an offset and slope correction on a channel. When proceeding to the offset correction, the input is not shorted. It's up to the user to proceed to that operation if needed.

Syntax `SENSe#:CORRection:SLOPe[:ACQuire]`

Example `SENS:CORR:SLOP`

Starts an offset and a slope correction.

Related commands

`INPut#:COUPling`

`SENSe#:CORRection:MEMorized`

3-7-27 *:SENSe#:SAMPler[:RATe]*

Defines at which rate the sampler must acquire the incoming signal.

Syntax `SENSe#:SAMPler[:RATe] <value HZ> | DEFault`

`SENSe#:SAMPler[:RATe]? [DEFAult]`

Default `500 KHZ`

Example `SENS:SAMP 100 KHZ`

Set the analog-to-digital converter sampling rate to 100 [kHz].

3-7-28 *:SENSe#:FUNCTion*

Defines how the sense subsystem must consider the incoming signal.

Syntax `SENSe#:FUNCTion OFF|VOLTage|FLUX|DEFault`

`SENSe#:FUNCTion? [DEFAult|OPTions]`

Default `FLUX`

Arguments

`OFF`	Turns the sense subsystem off.
`VOLTage`	Senses the input voltage.
`FLUX`	Integrates the input voltage.

Example `SENS:FUNC OFF`

This command turns the sense subsystem off. No value can be read from that channel in that mode.

3-7-29 *:SYSTem:HELP:HEADers? (Query only)*

Returns all commands known by the system.

Syntax `SYSTem:HELP:HEADers?`

Example `SYST:HELP:HEAD?`

Lists all available commands.

3-7-30 *:SYSTem:HELP:SYNTax? (Query only)*

Returns the syntax of the specified command

Syntax `SYSTem:HELP:SYNTax <command_header>`

Example `SYST:HELP:SYNT? INP`

The above command returns

```
INPut#:COUPling     GND|DC\n
INPut#:COUPling?    [OPTions|DEFault]\n
INPut#:GAIN         MINimum|MAXimum|
                    UP|DOWN|DEFault|<value>\n
INPut#:GAIN?        [OPTions|DEFault]\n
```

3-7-31 *:SYSTem:CHAnnel#:CDATe? (Query only)*

Returns the calibration date of the specified channel.

Syntax `SYSTem:CHAnnel#:CDATe?`

Example `SYST:CHA1:CDAT?`

This command might return

```
Tue Jan 22 14:03:45 2013\n
```

3-7-32 *:SYSTem:CHAnnel#:MDATe? (Query only)*

Returns the manufacturing date of the specified channel.

Syntax `SYSTem:CHAnnel#:MDATe?`

Example `SYST:CHA1:MDAT?`

This command might return

```
Thu Jan 01 13:53:19 2009\n
```

3-7-33 *:SYSTem:CHAnnel#:HWVERsion? (Query only)*

Returns the hardware version of the specified channel.

Syntax `SYSTem:CHAnnel#:HWVERsion?`

Example `SYST:CHA1:HWVERsion?`

This command might return

```
CORE v1.3-PXI|DISPLAY v1.0|MEMORY v1.0\n
```

3-7-34 *:SYSTem:CHAnnel#:SERial? (Query only)*

Returns the serial number of the specified channel.

Syntax `SYSTem:CHAnnel#:SERial?`

Example `SYST:CHA1:SER?`

This command might return

```
2056915 \n
```

3-7-35 *:SYSTem:FWVERsion? (Query only)*

Returns the firmware version of the instrument. The returned value is the same as the fourth parameter returned by the `*IDN?` command.

Syntax `SYSTem:FWVERsion?`

Example `SYST:FWVER?`

This command might return

```
1.0.2\n
```

3-7-36 *:SYSTem:LANGuage*

Select the command set used by the instrument.

Syntax `SYSTem:LANGuage SCPI|PDI5025`

`SYSTem:LANGuage? [DEFault|OPTions]`

Arguments

`SCPI` No effect.

`PDI5025` The language switch is made immediately, leading to an instant loss of communication, as the PDI5025 idiom is carried through a serial communication port.

Example `SYST:LANG?`

This command will return

`SCPI\n`

3-7-37 *:SYSTem:SERial? (Query only)*

Returns the serial number of the instrument. The returned value is the same as the third parameter returned by the `*IDN?` command.

Syntax `SYSTem:SERial?`

Example `SYST:SERial?`

This command might return

`2056-0001\n`

3-7-38 *:SYSTem:POWER*

Shuts the system down. This is the best way to turn the system off.

Syntax `SYSTem:POWer ON|OFF`

`SYSTem:POWer? [DEFault|OPTions]`

Arguments

`ON` No effect.

`OFF` Properly shuts the system down.

Example `SYST:POW OFF`

Shuts the system down.

3-7-39 *:SYSTem:PRESet (Event)*

Reinitializes the system using user's settings.

Syntax `SYSTem:PRESet`

Example `SYST:PRES`

Reinitializes the system using users setting.

3-7-40 *:SYSTem:CHAnnel[:COUNt]? (Query only)*

Returns the number of channels available in the system.

Syntax `SYSTem:CHAnnel[:COUNt]?`

Example `SYST:CHA?`

Returns the number of channels available in the system.

3-7-41 *:TRIGger#[:SEQuence]:SOURce*

Selects the source for the trigger event detector. Only one source may be specified at a time.

Syntax `TRIGger#[:SEQuence]:SOURce IMMediate|ENCoder|EXTernal|INTernal|TIMer|BUS`

`TRIGger#[:SEQuence]:SOURce? [DEFault|OPTions]`

Default `TIMer`

Arguments

`IMMediate` Specifies an immediate exit of the trigger state. No measurements can be tanken in that mode.

`ENCoder` Specifies that the trigger source is fed by the signal extracted from the encoder. The configuration of this extraction is made using the command `CONTRol#:ENCoder:CONFigure`.

`EXTernal` Specifies that the trigger source is fed by the trigger-in signal found on the front panel. The slope to be used is configured using the command `TRIGger#[:SEQuence]:EXTernal[:POLarity]`.

`INTernal` Specifies that the trigger source is fed internally. Used only on instruments having multiple channels which need to work synchronously.

`TIMer` Specifies that the trigger source is fed by the internal periodic signal source. The rate at which this source ticks is set using the command `TRIGger#[:SEQuence]:TIMer[:RATE]`.

`BUS` Specifies that the trigger source is software triggered. The command generating this trigger is `*TRG`.

Example `TRIG:SOUR TIM`

Sets the trigger event to `TIMer`.

3-7-42 *:TRIGger#[:SEQuence]:COUNt*

Determines the number of triggers needed to perform a full measurement.

Syntax `TRIGger#[:SEQuence]:COUNt <numerical value (no unit)>`

`TRIGger#[:SEQuence]:COUNt? [DEFault]`

Default `2`

Arguments

The numerical value given expresses the number of triggers that must occur before a measurement can be completed.

Example `TRIG:COUN 1024`

Ask for 1024 trigger events before finishing a complete measurement.

3-7-43 *:TRIGger#[:SEQuence]:ECOUNt*

ECOunt specifies a particular number of occurrences of the same event that must be recognized.

Syntax `TRIGger#[:SEQuence]:ECOUNt <numerical value (no unit)>`

`TRIGger#[:SEQuence]:ECOUNt? [DEFault]`

Default `1`

Arguments

`ECOunt` acts as a prescaler. The numerical value must be a positive value of 1 or greater.

Example `TRIG:ECOUN 10`

The tenth count of the source signal will generate a trigger pulse.

3-7-44 *:TRIGGER#[:SEQuence]:EXTernal[:SLOPe]*

Configures the transition detector of the external input trigger signal to generate a trigger-condition on a specific edge.

Syntax `TRIGger#:EXTernal POSitive|NEGative|EITHer|DEFault`

`TRIGger#:EXTernal? [DEFault|OPTions]`

Default POSitive

Arguments

POSitive A transition from low to high of the trigger-in signal located on the front-panel is used to exit the trigger level.

NEGative A transition from high to low of the trigger-in signal located on the front-panel is used to exit the trigger level.

EITHer Is a conjunction of both POSitive and NEGative conditions.

Example ARM:EXT EITH

Configures the transition detector to generate an event each time a transition occurs on the external input trigger signal.

3-7-45 *:TRIGGER#[:SEQuence]:ENCoder[:DIRection]*

When the trigger source is set to ENCoder, this subcommand helps to specify whether the encoder direction of interest is FORward or BACKward.

Syntax TRIGger#:ENCoder FORward|BACKward|DEFault

TRIGger#:ENCoder ? [DEFault|OPTions]

Default FORward

Arguments

FORward The forward definition must be set in accordance with the LED FW located on the front panel of the FDI channel.

BACKward The backward definition must be set in accordance with the LED BW located on the front panel of the FDI channel.

Example TRIG:ENC BACK

Configures the trigger system to react only to pulses generated when the encoder is travelling backwards.

3-7-46 *:TRIGger#[:SEQuence]:TIMer[:RATe]*

Sets the frequency of an internal periodic signal source. Its value affects the trigger system only when it is selected as the SOURce for the event detector.

Syntax TRIGger#:TIMer[:RATe] <value HZ> | DEFault

TRIGger#:TIMer[:RATe]? [DEFAult]

Default 100 kHz

Example TRIG:TIM 1KHZ

Sets the internal periodic signal source to generate a pulse every 1 [ms].

3-8 PROGRAMMING HINTS

Note that National Instruments' "Measurement & Automation Explorer" (part of the LabVIEW package) provides a very useful tool to explore the command set. Under "System / Peripherals & Interfaces / VISA TCP/IP Resources" or "… / USB Resources", select the FDI2056 and click the "Open VISA Test Panel" icon. This opens a window from which you can try all functions available through NI-VISA.

Here are a few notes on how the native command set is intended to be used:

- Use the `UNIT` command to set the units in which the results are returned, and the "digits" parameter to control the number of significant figures returned (assuming `FORMat` is `ASCii`).
- Use the `TRIGger:SEQuence` commands to control the measurement timing. As trigger source, you can select an internal timer, a trigger command sent from an encoder, or an external Trigger In signal. After setting up the trigger source, you initiate a measurement using the `INITiate` command.
- Use the `CALCulate:FLUX[:CUMulative]` command to control the integration mode.
- Use the `FETCh:ARRay?` command to retrieve all remaining data corresponding to a preceding `READ:ARRay?` or `INITiate` command.
- Using the `*OPC` command, you can also generate a ReQuest for Service (RQS) when a measurement (or any other action) is complete. Set bit 0 of the Standard Event Enable register and the ESB (Event Summary Bit) in the Status Enable register. Now, the execution of an `*OPC` command will generate an RQS.
- Be sure to check the status after every command. The Standard Event Status register provides a general idea of what went wrong, and the message on the Error/Event Queue (retrieved by `SYSTem:ERRor?`) provides a detailed diagnostic. See the SCPI standard for the exact interpretation of these error messages. It may be convenient to set up the Enable bits to generate a ReQuest for Service (RQS) when an error is encountered.

1.1.1 Error codes

Error codes returned by the native host interfaces are numbered according to the SCPI standard. Sections 21.8.9 through 21.8.16 of Volume 2 of the SCPI manual, "Command Reference", provide a generic description of all possible error codes. In general, the codes are between -800 and 300. This chapter describes only the error codes produced by the FDI2056, and the circumstances that might produce each error.

0	**NO ERROR**	Zero indicates no error.
-100	**COMMAND ERRORS**	
-102	Syntax error	The command header did not match any of the known commands.
-104	Data type error	A parameter within the command was of a type invalid for the command.
-115	Unexpected number of parameters	The wrong number of parameters was given in the command.
-123	Exponent too large	The command contains a numeric parameter that was too large to be stored internally. This occurs if the value has an exponent greater than ±43.
-151	Invalid string data	The parameters in the command contain an unmatched single or double quote.
-171	Invalid expression	The parameters in the command contain an unmatched bracket.
-200	**EXECUTION ERRORS**	
-221	Settings conflict	Indicates that a legitimate program data element was parsed but could not be executed due to the current device state.
-222	Data out of range	Indicates that a legitimate program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the device.
-225	Out of memory	The device has insufficient memory to perform the requested operation.
-300	**DEVICE-DEPENDENT ERRORS**	
-310	System error	System error.
-363	Input buffer overrun	The internal acquisition buffer was overrun. This happens when in `TIMer` triggered mode and when the host did not `FETCh` the data in time.
-400	**QUERY ERRORS**	
-400	Query error	Generic query error.
-410	Query INTERRUPTED	The host has sent a new command before finishing reading the response to a preceding query.

-420	Query UNTERMINATED	The host is trying to read a response without having sent a complete query.
-440	Query UNTERMINATED after indefinite response	Indicates that a query was received in the same program message after a query requesting an indefinite response was executed. On the FDI2056, the only command returning an indefinite response ("Arbitrary ASCII Response Data") is *IDN.
100	**INSTRUMENT-DEPENDENT COMMAND ERRORS**	
101	Invalid value in list	One or more values in a numeric list parameter are invalid, e.g. floating point when not allowed.
102	Wrong units for parameter	A parameter within the command has the wrong type of units for the command.
103	Invalid number of dimensions in channel	The channel list provided refers to more layers of multiplexers than allowed.
105	Numeric suffix invalid	The channel number provided is not valid.
200	**INSTRUMENT-DEPENDENT EXECUTION ERRORS**	
200	Software Error	The firmware has encountered an unexpected error.
201	Data not all available	The user is trying to fetch more data than was acquired.
202	Data buffer was overrun	No probe plugged in.
203	Unit already exists	The user is trying to register a unit which already exists in the system.
204	Unknown unit	The user is requesting information regarding an inexistent unit.
205	Invalid encoder configuration	The encoder configuration given is incorrect.
206	Operation in progress	The instrument is currently operating and cannot accept the command.
207	Channels don't share the same configuration	The command requested the value on the default channel, but this value is not shared by all channels present in the system.
208	Parameter value has been rounded or modified to meet internal constraints	The user has specified a value for a parameter that cannot be completely satisfied. The value has been rounded to the nearest value achievable by the system or the value resulted in a change in the system configuration.

USING THE PDI EMULATION MODE
4-Software interface

4-1 COMPATIBILITIES ISSUES

The PDI emulation mode offers a way to help customers who have already extensively used the PDI 5025 in the past, to integrate easily this new instrument in their existing installation. However, the FDI2056 is based on a completely different system and therefore some functionality present in the PDI 5025 have been transformed or removed. The following list enumerates all the differences between the two systems:

		PDI 5025	FDI2056
Communication	RS232	Hardware configurable	Software configurable
	IEEE488	Available	Unavailable
	Ethernet	Unavailable	Available
Motor control		Available	Unavailable[1]
Gain selection		1, 2, 5, 10, 20, 50, 100, 200, 500 1000	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50 100
Timestamp		Unavailable	Available[2]
Acquisition buffer		Limited to 5200 measures[3]	Limited to 512k measures[3]
Front panel FNC and Gain buttons		Available	Unavailable[4]
Channels		1 or 2	1 up to 9
Autonomous / Talker only mode		Available	Unavailable

1) Although a physical driver motor is not present in the system, a software layer is made available to help you to start and stop the motor with the same set of commands that were provided on the PDI 5025. Please read section 4-3.
2) Using an extended command (IMD,2)
3) This is the lower limit. If the data are read at a sufficient speed, there is no limitation.
4) The FNC and Gain buttons were used on the PDI 5025 to:
- Change the gain,
- Set the unit in offset compensation mode,

- Send a software trigger,
- Start a measurement.

During the following explanations, the red flag (⚑) symbol, signifies a change between the PDI 5025 original behavior and the new emulation one.

4-2 CONFIGURATION

Before using the FDI2056 in its PDI 5025 emulation mode, some parameters must be set according to the user's specific hardware interface and bi-phase encoder.

All parameters are located in the FDI2056 operating system registry. On 32-bit systems, the root key is:

HKLM\Software\Metrolab\FDI2056\PDI5025EmulationParameters

And on 64-bit systems, the root key is:

HKLM\Software\Wow6432Node\Metrolab\FDI2056\PDI5025EmulationParameters

(This is because the PDI 5025 emulator application has been targeted for 32-bit systems.)

The following tables describe all parameters, their usage, and specify the various values or numerical ranges. All keys described are relative to the root key as previously specified. All values are capital-insensitive.

Channels		
FDI2056_**SLOT2..11**	Up to 9 channels	
ChannelName	Letter	A,B,C,D,E,F,G,H,I
PDI channel name, must be unique for each integrator channels.		
DefaultGain	Numeric	1,2,5,10,20,50,100,200, 500,1000
Default PDI gain set at startup.		
ExternalTriggerEdge	Edge enumeration	disabled rising falling both
External trigger input configuration. This input is located on the front panel of each integrator channel.		
Gain_1	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Substitution values for each PDI analog input gain to map FDI gain values. Especially useful for PDI gain greater than 100 and FDI gain lower than 1. When the RGA command is issued, the returned gain value is the one given during the SGA		

command, not the substituted value.

Gain_2	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Gain_5	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Gain_10	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Gain_20	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Gain_50	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Gain_100	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Gain_200	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Gain_500	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
Gain_1000	Numeric	0.1, 0.2, 0.4, 0.5, 1, 2, 4, 5, 10, 20, 40, 50, 100
InternalTimeTriggerRate	Numeric	Default : 1000. Up to 500'000.

Internal time trigger rate, expressed in Hertz. The PDI default value is 1000 Hz. When changing this parameter, you must not exceed the sampling rate of the system.

SamplingRate	Numeric	Default : 100'000. Up to 500'000

Sampling rate used by the analog front end.

VFCType	VFC enumeration	Inactive 100kHz 500kHz 1MHz

This value is only provided as a means to fill the status register 4. It is not used otherwise.

Communication

Baudrate	Numeric	75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 28800,

			38400, 57000, 57600, 115200, 128000, 230400
	ByteSize	Numeric	7, 8
	FlowControl	Flow control enumeration	none xonxoff ctsrts
	Parity	Parity enumeration	no odd even mark
	Port	Port name	COM1 to last system COM.

This value must be set carefully; if the port doesn't exist the PDI emulator application will exit and reboot the system, resulting in an endless startup / reboot loop.

	StopBit	Stop bit values	1.0 1.5 2.0

QuadratureSignalConfiguration

	HasIndex	Boolean	true false

When set to true, indicates to the system that the encoder connected to the system provides an Index signal. This information is used to define the way the decoder will behave when an index search is performed.

With index: the decoder waits until the index is found and sets its internal counter to zero.

Without index: the decoder is set to zero immediately after the reception of the IND command.

	InvertA	Boolean	true false

All InvertX parameters indicate whether the input signal is used in its direct form or in its inverted version. This way, the direction detected by the bi-phase decoder can be configured. The configured version of the signals A, B and Index are then directly used to create a 90° index which must be positive.

	InvertB	Boolean	true false
	InvertError	Boolean	true

			false
	InvertIndex	Boolean	true false
	SingleEnded	Boolean	true false
	Sync	Edge enumeration	disabled rising falling both

The SYNC and ERROR signals share the same connector pin. When enabled, the SYNC input takes precedence over the ERROR function.

4-3 MOTOR SOFTWARE INTERFACE

Since the FDI2056 does not incorporate a motor driver, and since the PDI 5025 (which di provide such interface) offered a set of command to handle the motor, we have decided to offer a software layer to help our customers wishing to continue to use the synchronization capability offered by the original command set.

Be warned, this is a very rough interface and we do not handle precise positioning.

Basically, this software interface is a Dynamic Linked Library (DLL) which offers four entries:

- `void Start(unsigned int Direction);`

The emulator will call this procedure each time the motor must be started. The Direction parameters must be set to 1 to specify a forward/positive direction and to 0 to specify a backward/negative direction.

The definition of forwards and backwards must respect the definition of the direction you have set during the configuration of the Bi-Phase decoder. See previous chapter.

- `void Stop();`

The emulator will call this procedure each time the motor must be stopped.

- `void Initialize();`

The emulator will call this procedure when it starts. You may proceed in this procedure to all initializations needed by your motor driver.

- `void Release();`

  The emulator will call this procedure when it stops. You may proceed in this procedure to the cleanup needed to restore the system on the same state as before the call of `Initialize`.

A complete but empty project, written for and using Microsoft Visual Studio 2010©, is available on the software distribution CD and in the folder C:\Program Files\Metrolab\FDI2056\MotorInterface of the FDI2056. Once completed to fulfill your motor driver requirements, the generated DLL, named "MotorInterface.dll" must be placed in the folder C:\Program Files\Metrolab\FDI2056\ where the default DLL, which was generated using this empty project, is placed.

4-4 POWER ON SEQUENCE

When powered on, the first operation performed by the FDI2056 is an autotest which takes several seconds.

Should one integrator channel fail, Status bytes 5 and 6 will indicate any error conditions that raised this situation.

During the autotest the word "TEST" is displayed on the module. At the end of it, the display will either show a gain of 10 with the channel identification letter ("A", "B" up to "I") or, if an error has been detected, one of the following status indicators will be displayed : "!CAL", "!MEM", "!INP" or "!RNG". Section 4-11-9 describes the corrective action that may eventually take place if such a situation arises.

This autotest sequence will automatically follow when PDI 5025 emulation mode is entered. During autotest, no communications may be established with the host. Thus, any messages sent to the FDI2056 will be ignored.

4-5 CHANNEL SELECTION

The FDI2056 can be equipped with as many as nine channels (the standard unit is equipped with a single channel). All channels are always triggered in parallel. Measurements can be performed on an individual channel or on all channels simultaneously. The **CHA** command selects the active channel or channels.

If issued on a unit equipped with a single channel, this command will generate a Command Error (bit 5 of status 1 set to 1).

4-6 GAIN SELECTION

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

1, 2, 5, 10, 20, 40, 50, 100

It is selected 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).

We recommend that the gain be 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.

4-7 OFFSET ADJUSTMENT.

The instrument has the possibility of reducing the linear drift of the integrator channel by adjusting the voltage offset at the input. The voltage offset depends greatly on the gain value. Therefore, as a best practice recommendation, the offset adjustment should be requested each time a new gain is set. This adjustment is made each time the **ADJ** command is sent.

The **ADJ** command can be sent to any channels in the system at the same time.

It's up to the user to decide whether the input will be short-circuited by the internally provided circuit or by using an external circuit. Should you decide to use the internal short-circuitry; the **ISC** command will help you to do so.

4-8 INTEGRATOR MODE

The Integrator-Timer mode - **TRS,T** -, which is the default mode at power-on, is generally used with fixed coil configurations to analyze time dependent flux variations. The Integrator-Displacement or Encoder mode - **TRS,E** - is used with moving coil configurations to analyze space dependent flux variations.

In these modes, every channel can be selected or deselected individually by using the **CHA** command.

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 and can be repeated as many times as required by the **RUN** command.

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

4-9 PROGRAMMING THE INTEGRATION INTERVALS

The **TRI** command is used to define the integration intervals which are delimited by successive trigger events. The syntax of this command is explained below:

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

where:

s = + or -	the direction of the movement (default +)
a = value	the absolute position of the first trigger condition (default 0)
n_i = 1 to 65535 or *	the number of intervals of C_i pulses or counts. '*' means infinite number of intervals.
C_i = 1 to 2^{23}	the number of encoder pulses or time counts between triggers (always positive)
i = 1 to 20	the 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 milliseconds) 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 the triggering, the buffer will never get full and the run will last forever. If the "last cumulative storage" mode – **CUM**,1,L – is selected, the instrument will behave as 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 $^{CR}_{LF}$	equivalent to TRI,+,0/5,200 $^{CR}_{LF}$
TRI,,50 $^{CR}_{LF}$	this command will roughly move the motor to the specified position without performing any measurement. The accuracy of the positioning is very low.
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.

4-9-1 *Trigger Source: Timer*

The internally quartz controlled time base generator is set by default to operate at a frequency of 1 [kHz] and is used to feed the FDI2056 Trigger Factory. The **TRI** command defines the integration intervals in terms of number of periods – As the default frequency is 1 [kHz], you may consider this integration interval as the time spent for each partial integral expressed in milliseconds. If you would like to proceed to a faster integration time, change the default time base generator (see section 1-).

The counter can be activated in two different ways:

a) by the **RUN** command. In this case the TRS command - which will have to have been issued before - must be **TRS,T**.

b) by the first external synchronization signal on the SYNC input which follows the **RUN** command issued by the host. All following signals on the SYNC input will be ignored. In this case the **TRS** command must be **TRS,T,S**

4-9-2 *Trigger Source: Bi-Phase Encoder*

The Bi-Phase decoder module, located on slot 2 of the instrument, has been designed to accept the signals generated by linear or rotational incremental encoders. These signals are formated and filtered before being fed to the Trigger Factory with 4 pulses representing 1 encoder cycle (multiplication by 4).

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

4-9-2-1 Rotational Encoder with Index

TRS,E,dddd

Field dddd expresses in decimals the number of cycles per rotation.

We remind you that since the decoder tracks the number of edge by cycle, for an encoder with 360 cycles (TRS,E,360) you will actually end up with 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 initialize the measurement sequence. This may be done manually or by using the **IND** command which will turn the motor in the indicated direction until the index pulse is detected. This point is taken as a reference, and therefore, all measurements are absolute to it and are verified on each successive index pulse.

It follows from the above that the absolute value given in a **TRI** command cannot be greater than the number of pulses per encoder rotation, neither can it 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 direction 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 $^{CR}_{LF}$	Indicates a disk encoder with 360 cycles per rotation with an index.
IND,- $^{CR}_{LF}$	Turn the motor in the backwards direction (BW) until the index pulse is detected.

TRI,-,140/8,10/3,1000 $^{CR}_{LF}$	Prepares a sequence of 11 measurements in the backward direction, the first TRIGGER is at absolute position 140 (pulses, not cycles).
MOT,A $^{CR}_{LF}$	Sets motor to automatic
RUN $^{CR}_{LF}$	Starts the motor, perform the measurements then stop the motor.

4-9-2-2 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^{30} to $+2^{30}$, on either side of the index point
- The absolute position in the **TRI** programming sequence of the command can be negative

4-9-2-3 Linear or Rotational Encoder without Index, but with External Synchronization

TRS,E,S

The external synchronization 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 current position for the SYNC pulse.

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

4-9-2-4 Linear or Disk Encoder without Index or External Synchronization

TRS,E

The TRS counter is cleared by the RUN command.

Warning: The mathematical sign (+,-) of the absolute position **a** must be the same as that of the direction **s** in the **TRI** command.

4-9-3 *Trigger Source: External*

A trigger signal can also be fed externally and is selected by sending the **TRS,X** or **TRS,X,S** command.

In this case the **TRI** command does not serve any purpose since the triggering sequence is not controlled internally.

The external trigger source (the active edge of which can be selected by software – Please refer to chapter 4-2) must be connected to the input denoted "*External Trigger: IN"* of the integrator channel.

Note: If the option "external trigger is dispatched" is selected, the trigger signal will be fed to all integrator channels present in the instrument. In this particular case, the trigger must be fed to the first integrator channel located on slot 2 only. By choosing this option, you alleviate yourself the burden of having to wire all integrator channels with your trigger source. See section 4-2 for more details regarding this option.

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 "End of Run" of STATUS 1 (bit 3) is then set.

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

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

A measurement sequence can be initiated by an external synchronization signal. To select this mode, use **TRS,X,S**. This works in the manner described above, except that the trigger signals are disabled until the first external synchronization pulse occurs after the **RUN** command. Any following external SYNC signals are ignored. The external synchronization shares the same pin as the ERROR signal found on the bi-phase trigger input connector.

4-10 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 about its current state. Each status register contains eight bits and can be read by the host computer in two different formats, in either hexadecimal or binary. The registers are numbered 1 to 7.

To read a status register, the host computer must send 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 eighth character to the least significant bit.

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

STATUS 1	**Measurement Status (1 Byte)**
	This status byte is cleared as soon as 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 indicate that STATUS 2 is not zero and that it should be read.
Bit 6	This bit is always read as 0.
Bit 5	Command Error This bit is set 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 not appropriate.
Bit 4	Over-range Error This bit is set each time the input voltage of any channel reaches the positive or negative over-range threshold. *Note:* An over-range error has no effects before the first trigger

	(when waiting for a synchronization. signal or the absolute value of the first trigger, for example). After this first trigger, an over-range condition aborts the measurement cycle immediately and has a similar effect to the reception of a **BRK** command (see section 4-11-7). The run abort can be disabled with the command **NBO**. STATUS 4 indicates which over-range has occurred.
Bit 3	End of RUN This bit is set 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 setting of this bit 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 as soon as all the measured data have been converted and are thus available to the host.
Bit 1	Trigger This bit is set on receipt of the trigger signal occurring during a measurement cycle. The trigger can be internally or externally generated.
Bit 0	Synchro This bit is set each time a signal is detected on the SYNC input or on the encoder index input. This can be used to synchronize the host computer software with the pulse from the index of an encoder.

Table 4-1 STATUS 1: Measurement Status

STATUS 2	**Error Status (1 Byte)** This status byte is cleared as soon as it has been read, this also clears bit 7 of STATUS 1.
Bit 7	This bit is always read as 0.

Bit 6	This bit is always read as 0.
Bit 5	This bit is always read as 0.
Bit 4	Power on reset This bit is set at power on.
Bit 3	Autotest failed This bit is set if the autotest detects an error in the instrument. The exact cause of the error can be found by examining STATUS 5 and STATUS 6.
Bit 2	Encoder count error This bit is only valid if used with an encoder providing an index. Each time the index is detected, the instrument verifies if the amount of pulses counted from the previous occurrence is correct. If an error is found this bit is set.
Bit 1	Measurement buffer full When the instrument's internal buffer is full, this bit set to indicate this error. This error will stop the current measurement cycle (as if **BRK** had been received; see section 4-11-7).
Bit 0	Overlap timeout This error happens when the trigger rate is faster than the sampling rate. This error will stop the current measurement cycle (as if **BRK** had been received; see section 4-11-7).

Table 4-2 STATUS 2: Error Status

STATUS 3	**Trigger source (1 Byte)**	
Bits 7..5	TRS Mode	
000	Offset adjustment	ADJ,1
001	Timer without synchronization	TRS,T

010	Timer with Synchronization	TRS,T,S
011	Encoder without index nor synchronization	TRS,E
100	Encoder with external synchronization	
	Linear with Index	TRS,E,S
101	Rotating Encoder with index	TRS,E,dddd
110	External Trigger	TRS,X
111	External Trigger with synchronization	TRS,X,S

Bit 4 Infinite sequence

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

Bit 3 Measurement in progress

This bit is set to indicate that the FDI2056 is in the process of measuring. The bit is automatically set by the **RUN** command; it is automatically cleared when the sequence is terminated.

Bit 2 Forwards / Backwards

A value of 1 indicates motion in the forward direction (FW LED lit), a value of 0 refers to the backward direction (BW LED lit).

Note that when no encoder is in use, this bit has no meaning.

Bit 1 Motor +

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

This bit is set or cleared even if the software motor interface is unused; see chapter 4-3 for more details.

Bit 0 Motor -

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

This bit is set or cleared even if the software motor interface is unused; see section 4-3 for more details.

Table 4-3 STATUS 3: Trigger source

STATUS 4	**Integrator Channel (1 Byte)**	
	Access to integrator channel status for channel C to I are accessed using the **CHA** command.	
Channel B **Channel D** **Channel F** **Channel H**	Bit 7..6	VFC Type These bits show if the channel is active. The reported Voltage To Frequency type depends on the value set during the configuration. Please read section 4-2. 00 : Channel B is inactive 01 : VFC 100 kHz 10 : VFC 500 kHz 11 : VFC 1 MHz
	Bit 5	Negative over-range This bit reports the instantaneous negative over-range status. When the negative over-range LED on the bargraph display of the channel is lit, this bit is set.
	Bit 4	Positive over-range This bit reports the instantaneous positive over-range status. When the positive over-range LED on the bargraph display of the channel is lit, this bit is set.
Channel A **Channel C** **Channel E** **Channel G** **Channel I**	Bit 3..2	VFC Type These bits show if channel is active. The reported Voltage To Frequency type depends on the value set during the configuration. Please read section 4-2. 00 : Channel A is inactive 01 : VFC 100 kHz 10 : VFC 500 kHz 11 : VFC 1 MHz
	Bit 1	Negative over-range This bit reports the instantaneous negative over-range status.

	When the negative over-range LED on the bargraph display of the channel is lit, this bit is set.
Bit 0	Positive over-range This bit reports the instantaneous positive over-range status. When the positive over-range LED on the bargraph display of the channel is lit, this bit is set.

Table 4-4 STATUS 4: Integrator Module

STATUS 5

STATUS 6 Autotest (1 Byte)

STATUS 5 relates to Channel A (or C/E/G/I), while STATUS 6 refers to Channel B (or D/F/H). If only one integrator channel is placed in the instrument, the status bits of the missing channel are always zero.

Access to Integrator Channel Status for Channel C to I are accessed using the **CHA** command.

Bit 7	This bit is always read as 0.
Bit 6	This bit is always read as 0.
Bit 5	This bit is always read as 0.
Bit 4	Error in 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 4-11-9).
Bit 3	Negative over-range not reported This bit shows that a deliberate negative over-range was not detected during the autotest and reports this error.
Bit 2	Positive over-range not reported This bit shows that a deliberate positive over-range was not detected during the autotest and reports this error.
Bit 1	Negative over-range erroneously reported

	An over-range was detected in the presence of a negative input signal which should not have produced this report.
Bit 0	Positive over-range erroneously reported An over-range was detected in the presence of a positive input signal which should not have produced this report.

Table 4-5 Status 5 & 6: Autotest

STATUS 7	**Acquisition (1 Byte)**
Bit 7	This bit is always read as 0.
Bit 6	This bit is always read as 0.
Bit 5	Timestamp Transfer Mode When set, this bit indicates that the timestamp will be transferred with the data (**IMD,x,T**).
Bit 4	Infinite sequence active When set, this bit indicates that an infinite trigger sequence is in progress. An infinite sequence is selected with ni = * in the **TRI** command. This bit is similar to bit 4 of STATUS 3.
Bit 3	Run active This bit is set to indicate that the FDI2056 is taking measurements. This status bit is similar to bit 3 of STATUS 3.
Bit 2	Data Transfer Mode When set, this bit indicates that the immediate data transfer mode (**IMD,1**) is selected. A value of 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).

00	Individual values
01	Cumulative values
10	Only the last cumulative value is available.

Table 4-6 STATUS 7: Acquisition

4-11 DATA ACQUISITION

During a measurement cycle the FDI2056 stores all partial integrals and time stamps in an internal buffer. This raw material 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, cumulative and last cumulative value. 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.

4-11-1 *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.

4-11-2 *Cumulative 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 FDI2056, the cumulative value is cleared to 0 (zero) at each occurrence of an over-range. At the end of a run, the number of available measurements data is identical to the number of integration periods.

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

4-11-3 *Last cumulative data storage (CUM,1,L)*

This mode stores only the last cumulative 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.

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

With an adequate TRI sequence, this mode allows the user to use the FDI2056 as a fluxmeter.

4-11-4 *Direct Data Transfer (IMD,1)*

This is the default data transfer mode. Measured data are read individually. As soon as the measured data have been processed, the "Data-Ready" bit in STATUS 1 is set.

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.

To read the next value, the host computer must send the **ENQ** command.

The returned value is expressed in 10^{-8} [Vs] followed by a space and then the channel identifier (e.g. 45982900 A).

If an over-range occurs during the measurement with the command **NBO,1** having been previously sent to the instrument, 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** command. 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 is performed before the next data value is available (i.e. "Data-Ready" not yet set) then the instrument will return the characters "carriage return" and "line feed" (CR LF). This 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 4-11-6.

4-11-5 *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 (bit 2 STATUS 1)

To initiate the block transfer, the host computer must send the **ENQ** command. The instrument will then transmit all the data in the order that it was measured.

The returned value is expressed in 10^{-8} [Vs], followed by a space and then the channel identifier (e.g. 45982900 A).

Each value is separated by the CR LF pair of characters. If more than one channel is active, the values are sent in decreasing channel order (i.e. for a two-integrator channel instrument, Channel B values will be sent before the values 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 4-11-6).

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

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

4-11-6 *End of Data (EOD)*

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 $^{CR}_{LF}$

the FDI2056 will reply

Empty

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

4-11-7 *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 bit 3 of STATUS 1 (End of Run) is set.

Using 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** -.
- The external trigger input is disabled.
- All data measured up to the instant that the **BRK** was received are valid and can be read by the host.

The FDI2056 can also execute a **BRK** automatically if any of the following errors occurs:

- Over-range error (STATUS 1 : bit 4)
- Overlap timeout (STATUS 2 : bit 0)
- Measurement buffer full (STATUS 2: bit 1)

4-11-8 *Automatic Use of a Motor (MOT,A)*

As the FDI2056 does not incorporate a motor driver, this command calls an external library that can be tailored by the end user. Please read the section 4-3 for more details.

Assuming the user has provided the adequate software layer, the instrument will behave as follows:

Apart from the three commands **MOT,+**, **MOT,-** and **MOT,S**, which are used to start and stop the motor, there is also the possibility of controlling the motor automatically during a measurement cycle. Thus the motor starts when the **RUN** command is received and stops when the measurement cycle is completed.

This automated mode is entered after the reception of the **MOT,A** command and exited at the reception of the **MOT,S** command.

4-11-9 *Autotest (TST and AUT)*

Each time the FDI2056 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 prove incorrect, the display will show one of the following words "!CAL", "!MEM", "!INP" or "!RNG".

	Message	Cause
	!CAL	The internal compensation file is corrupt. You may either : Proceed to a full compensation by sending the **ADJ,2** command which will perform the measurements needed and update the user compensation parameters. or Restore the factory compensation parameters by sending the **ADJ,3** command which will proceed to the replacement of the user compensation parameters with those generated during the integrator channel production. If the problem persists after one of these corrective operation attempts, please contact us.
	!MEM	The internal system memory is corrupt. This is a major issue, please contact us.

	!INP	The analog input circuitry doesn't perform as it should. This is a major issue, please contact us.
	!RNG	The over-range detection mechanism does not perform as expected. This is a major issue, please contact us.
	!HWU	A firmware upgrade, including FPGA reconfiguration, is needed.

The user may obtain detailed information on the problems encountered during the autotest by examining status registers STATUS 5 and STATUS 6.

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

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:

To be defined.

4-11-10 *Offset compensation (ISC, ADJ)*

To adjust the offset of the acquisition module, the input signal is measured during at least two seconds at a very high sampling rate (500 kSPS) and the mean is used to zero the offset. This operation is performed as soon as the **ADJ** command is sent to the instrument. This automatic adjustment is made for the current input gain only.

You may either decide to use the internal switch to short-circuit the input coil or to use your own external short-circuit taking into account thermocouple and other effects.

To short-circuit the measurement coil, you must issue the **ISC** command.

4-11-11 *Synchronization by the Host Computer (SYN)*

As already mentioned in section 6.6, it is possible to synchronize a sequence of measurements with an external event. This can either be a signal supplied to the SYNC input of the acquisition module or it may be the **SYN** command generated by the host computer.

When the user has defined the source of the trigger with external synchronization, (**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 before starting the measurement cycle.

4-12 PDI 5025 EMULATED COMMANDS

All the PDI 5025 emulated commands are listed in tabular form in this section in their order of use, beginning with initialization, measurement and data extraction commands followed by miscellaneous commands. This does not, however, signify that they must be executed in this order!

SYNTAX		DESCRIPTION	MNEMONIC
CHA,A	*Default*	Select channel A.	CHAnnel
CHA,B		Select channel B.	
CHA,*		Select both channels A and B.	
	Note:	This command is valid only if the FDI2056 is equipped with two channels.	
Trigger source selection (see section 6.6).			
TRS,T	*Default*	Timer (without external synchronization).	TRigger Source, Timer
TRS,T,S		Timer (with 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

SYNTAX	DESCRIPTION	MNEMONIC
TRS,E,d..d	Rotational incremental encoder with d..d cycles per turn and with index signal. Note: The decoder module counts 4 pulses for each encoder cycle.	TRigger Source, rotational Encoder
TRS,X	External hardware trigger. The hardware trigger is fed into the appropriate input of the acquisition module.	TRigger Source, eXternal
TRS,X,S	External hardware trigger with external synchronization. Note: When executing the TRS command, the previous TRI sequence is cancelled, the motor is stopped and the MOT,A becomes inactive.	TRigger Source, eXternal, Synchro

SYNTAX	DESCRIPTION	MNEMONIC
TRI,s,a/n1,C1/n2,C2/.../n20,C20		TRIgger sequence
	s The direction of displacement : + or -	
	a The absolute position of the start (1st trigger): -2^{31} to $2^{31}-1$.	
	ni The number of integration intervals of the "i" subsequence (ni = 1 to 65535 or ni=*; the sign "*" stands for "infinity").	
	Ci The number of events received before the sending of a trigger to the integrator. The integration intervals of the "i" sub-sequence is given in number of input events (1 to 4095). The maximum number of sub-sequences is 20.	
TRI,+,1	*Default*	
	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 interval.	
TRI,?	Returns a string containing the current values of the sequence.	
SGA,i,d..d	Sets the Gain of the 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	*Default* 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	Performs a measurement immediately to compensate the offset.	ADJust
	Note : The input will not be short-circuited. It is therefore up to the user to either externally short the input or send the ISC command before this operation. Failing to ensure clean initial conditions may lead to poor measurement results and high drift.	
	When the selected integrator channel enters the Offset Adjust Mode, it automatically performs an average of the input signal during 2 seconds. This averaged measurement is used to zero the offset.	
	Once the mean is achieved, the instrument	

	SYNTAX		DESCRIPTION	MNEMONIC
			automatically leaves the adjustment mode and asserts the "End Of Run" bit in the STATUS 1. Each time you perform a compensation using one of the ADJ commands, the parameters will remain active until a new use of an ADJ command. These compensation parameters are automatically lost at POWER OFF.	
☞	ADJ,0 or ADJ,i,0		Stop the OFFSET ADJUST mode on the selected channel. If the selected channel is not in OFFSET ADJUST mode, the command is ignored.	
☞	ADJ,i,2		Performs a full compensation of both the offset and the gain for the current gain. This operation is performed for the current gain.	
☞	ADJ,i,3		Same as ADJ,i,2 but, in addition, store the parameters in the integrator channel's non-volatile memory. This difference in parameter storage is of capital importance, because ADJ,i,3 is the only command which allows compensation parameters to be retained by the system after POWER OFF.	
	IND,s		s = + :forward direction. s = - : backward direction. 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.	INDex
	Motor interface: Please read section 4-3 We remind you that the FDI2056 doesn't incorporate a motor driver.			
	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	*Default*	stops the motor and cancels the MOT,A command.	MOTor, Stop
	MOT,A		Automatically starts the motor. The motor starts automatically following	MOTor, Automatic

SYNTAX		DESCRIPTION	MNEMONIC
		the RUN command. The direction of rotation (or displacement) is defined by the TRI command. When the sequence of measurements is completed, the motor is stopped.	
	Note:	the MOT,S command cancel MOT,A.	
WARNING		The motor and encoder wiring must be such that MOT,+ starts the motor in the forward direction and MOT,- in the backward direction (respectively the FW and BW front panel LEDs are lit).	
RUN		Starts the programmed measurement sequence. 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. Terminates an external trigger measurement.	BReaK
EOD,a1,..,an		Defines an "end of data" string. a1,..,an decimal ASCII code of characters defining the "end of data".	End Of Data
	Default	^Z	
		When the measurement buffer is empty, the string "a1,..,an" is returned following the ENQ command.	
	Note:	At power on or if "a1,..,an" is omitted, the ASCII Dec. 26 (CTRL-Z) character is used as the default "end of data" string.	
CUM,0	*Default*	Integrated values are stored separately.	CUMulative
CUM,1,S		The cumulative value from the start of the measurement is stored in memory at the end of each integration interval. If the command NBO,1 has previously been sent to the FDI2056, the cumulative value is cleared to 0 at each occurrence of an over-range.	
CUM,1,L		Only the last cumulative value is stored and available for reading. This mode is	

	SYNTAX		DESCRIPTION	MNEMONIC
			incompatible with the block data transfer (IMD,0) and forces an immediate data transfer mode (IMD,1). See section 4-11 which explains timing limitations.	
		Note:	It is not possible to change the storage mode during an active run.	
	IMD,1	*Default*	Flux values can be read during the measurement. The flux values can be read before the sequence of measurements has been completed. The values are read one by one. If reading is performed before data is available, the PDI 5025 returns CR LF. When "Data ready" bit in STATUS 1 (bit 2) is set, it indicates that a new measurement value is available.	IMmeDiate
	IMD,0		Flux values are read as a block at the end 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 sequence. When "Data ready" bit in STATUS 1 (bit 2) is set, it indicates that the block of data can be read.	
		Note:	This mode is incompatible with the CUM,1,L command.	
	IMD,i,T		Flux values with associated timestamps can be read during the measurement. This mode is the same as IMD,1 or IMD,0 but each measurement is followed by a comma and the associated timestamp.	IMmedDiate with Timestamps
	ENQ		This command allows data to be read in the manner specified by the IMD command.	ENQuiry
		Notes:	When the data buffer has been read and reset, a new read request will make the instrument return either the default string (CTRL-Z) or the one defined with the EOD command. This command is only used to read the measured values and must not be sent to read the status registers.	
	FPT,1 FPT,n	*Default*	n = 1: enables the front panel trigger input. n = 0: disables the front panel trigger input.	Front Panel Trigger

SYNTAX		DESCRIPTION	MNEMONIC
	Note:	this command can only be used when TRS,X or TRS,X,S has been previously executed. A typical application of this command is to disable unwanted external trigger pulses.	
STH,d		Returns 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, STATUS 1 is returned by default. The first and last characters are respectively 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 another 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 all 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 over-range of channel i (i = A, B,.. or *)	Clear over-range
	Note:	the + or - over-range LED lit on the front panel bargraph display is switched off. If it 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,i		Returns the current Gain of channel i (i=A, B, … or *).	Read GAin
RGA,B		Returns the current Gain of channel B.	Read GAin
	Note:	If no parameter is given in the RGA command, the Gain of the active channel	

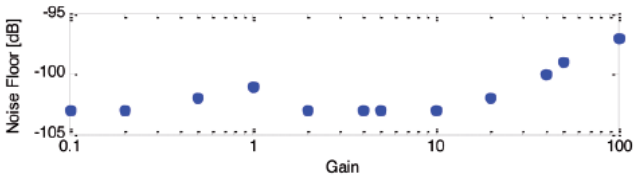
SYNTAX		DESCRIPTION	MNEMONIC
		is returned. If many channels are active, A is the active channel by default.	
RCT		Reads the counter of the Trigger Factory. After reading the counters, the FDI2056 converts the result to decimals and transfers it in the format +/- ddd..d. Leading zeroes are suppressed. 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.	Read CounTer
ZCT	*Default*	Resets the counters of the Trigger Factory. Automatic at power on. Warning: Using this command requires particular care! Use only for specific and well-defined applications! Resetting the counters to zero during RUN, will perturb the sequence of measurements!	Zero CounTer
NBO,0 NBO,1	*Default*	The occurrence of an over-range condition will immediately stop the measurement in progress (if any) and set bit 4 of STATUS 1 to one. By default, when an over-range condition occurs, the measurement sequence is stopped. This command disables the instrument's default reaction to an over-range condition. This command must be used with great care because it is not possible to determine when the over-range condition occurred and for how long it lasted. The wrong data are stored in the buffer as would be any valid data! However, bit 4 of STATUS 1 is set to one. If an over-range occurs during the measurement, the returned value is 0 with an exclamation mark between the last digit and the channel descriptor (e.g. 0!A).	Not Break Over-range
TST		This command allows the FDI2056 channel (or channels) to be tested automatically. Following this command, the state prior to its execution is restored.	TeST
AUT		Returns test measurements. The command sends the block of 5 measurements calculated during the	AUTo test

	SYNTAX		DESCRIPTION	MNEMONIC
			automatic test performed following the power on sequence or following the TST command.	
	ISC,i,0	*Default*	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 integrator channel.	
	FNC		*Command no longer applicable, will be ignored by the system.*	FuNCtion
	LLO		*Command no longer applicable, will be ignored by the system.*	Local LOckout
	MSK		*Command no longer applicable, IEEE488 specific, generates a command error.*	MaSK
	SYN		Generate synchronization.	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). The software synchronization replaces the hardware synchronization which can be fed into the appropriate SYNC input of the Trigger Factory.	
	EMU,V ¤		Returns the current version of the PDI emulator	
	EMU,P		Turns the system off. As soon as it is safe to operate the power switch to "off" state, you will be prompted on the display with the text "Off.".	
	EMU,L		Leave the emulator and enters in the VXI communication mode. The RS232 communication will be lost to the benefit of this Ethernet protocol.	
	EMU,F		Set the boot mode to VXI FDI mode. The next time you turn the system on, the instrument will enter the VXI communication mode. This command does not require you to leave the PDI mode.	

REFERENCE
5-Technical Details

5-1 TECHNICAL SPECIFICATIONS

DIGITIZER

Gain	0.1, 0.2, 0.4, 0.5, 1.0, 2, 4, 5, 10, 20, 40, 50, 100	–
Dynamic range	±10 ÷ Gain	V
Input overvoltage protection	±15 ÷ Gain	V
Max common mode voltage	12 ÷ Gain	V
Max input bandwidth	250 @ Gain ≤ 10, decreasing to 25 @ Gain 100	kHz
Noise floor (@ 1 kHz bandwidth)		
Input impedance	200	kΩ
Gain accuracy	10	ppm
Digitizer resolution	18	bit
Max sample rate	500	kS/s
Nonlinearity: Single Tone	-105	dBc
Nonlinearity: Dual Tone	-95	dB

INTEGRATOR

Timer resolution	12.5	ns
Time base stability over temperature	±0.075 (0 to 60° C)	ppm
Time base stability over time	$< 5 \times 10^{-4}$ (30 s) ± 0.7 (1 year)	ppm
Drift	10^{-5}	FS / min (1)
Drift variation	typical < Noise Floor ÷ 5	Vs / s

COMMON

Trigger sources		External, timer, encoder, software, multichannel	
Trigger rate		0.02 to 500k	Hz
Encoder input:			
	Voltage	3.3 or 5	V
	Current protection	750 (Hold), 1500 (Trip)	mA
	Signal type	Single-ended or differential	
	Index type	None, or 90° – 270°	
Memory capacity		1M	PI (2)

SYSTEM

Industrial computer	Intel x86 architecture, Windows OS, 16 GB RAM, 32 GB Flash drive, Ethernet, USB 2.0 (3)	
Number of channels	1 – 3	
Ethernet interface	VXI-11 (IEEE 488.2), SCPI compliant	
Max transfer rate (Ethernet)	1000 (3)	PI / s (2)
RS-232 Interface	PDI5025 compatibility mode	
Power requirements	100 – 240 V, 50 – 60 Hz, 80 A inrush current max	
Operating temperature	0 – 40	°C
Size and weight	445 x 130 x 245 mm (19" x 3U), 7.2 kg max	
Mounting	Horizontal or vertical, optional rack-mount kit	
Recommended calibration interval	12	months

(1) FS = Full Scale.
(2) PI = Partial Integral, including timestamp.
(3) Subject to change; contact Metrolab for exact specifications.

5-2 INTEGRATOR CHANNEL REGISTERS DESCRIPTION

Memory mapping (unless mentioned, all registers are readable / writable)

	Offset	0x0	0x1	0x2	0x3
BAR2	0x000	Read : Data, Write : Trigger configuration FIFO.			
BAR3	0x100	Status	Mode	Interrupt enable	Interrupt pending
	0x104	Gain configuration	Signal path configuration & Voltage reference selection	Trigger input configuration	Bi-Phase decoder configuration
	0x108	Data processing reported events	Data acquisition reported events	FIFO status	
	0x10C	PXI Star Trigger: configuration and enabling system.			
	0x110	Bi-Phase decoder: current position [Edge count]			
	0x114	Bi-Phase decoder: maximum count - 1 [Edge count]			
	0x118	Bi-Phase decoder: target count [Edge count]			
	0x120	Sampling Rate Period (Integrator mode only).			
	0x124	Time Trigger Rate Period			
	0x128	Gain Factor (Double precision IEEE 754 floating point – 64 bits –)			
	0x12C				
	0x130	Offset (Signed integer – 32 bits –)			
	0x134	JTAG Configuration			

	0x13C (Read only)	Version	Revision	Frequency Multiplier	Frequency Divider
BAR4	0x200-0x27C	Calibration parameters			
BAR5	0x300	Display char 1	Display char 2	Display char 3	Display char 4

Trigger configuration Fifo (BAR2 : 0x000)		
Bits 31..28	Trigger source	0000 : External 0001 : PXI 0010 : Internal Time Trigger 0011 : Software 0100 : Bi-Phase Index 0101 : Bi-Phase Pulses positive direction 0110 : Bi-Phase Pulses negative direction 0111 : Bi-Phase Pulses 1000 : Bi-Phase Target
Bits 27..16	Trigger prescaler.	
Bits 15..0	Triggers count	A value of 0 will be interpreted as infinite count.

The trigger configuration uses a 16-Words fifo. Once the acquisition is started, the first line of the FIFO is used to select and count the trigger events as configured by the user. As soon as the programmed count in the line is exhausted, the system automatically tries to read a new line. The acquisition will be marked as completed once no more line is available. Since the fifo can be monitored at any time by reading the bit 2 of the Status register, it is possible for the user to add a new trigger configuration line each time a line count is exhausted, letting an infinite number of configuration possible.

Status Register (BAR3 : 0x100)		
Bit 0	Acquisition complete	The acquisition is complete, the Trigger FIFO having been exhausted.
Bit 1	Data FIFO Event	The detailed condition which generated this event is available in the "FIFO status" register, bits 0 to 3.
Bit 2	Trigger FIFO Event	The detailed condition which generated this event is available in the "FIFO status" register, bits 4 and 5.
Bit 3	Calculation Overflow	The detailed condition which generated this event is available in the "Data processing reported events" register.
Bit 4	Trigger Overrun	More than one trigger event was detected between two data acquisitions.
Bit 5	System error	ADC or Encoder error. More details on the reasons which

		generated this event are available in the "Acquisition processing reported events" register.

Mode (BAR3 : 0x101)		
Bit 0	Enable acquisition	Enable the acquisition process.
Bit 1	Enable trigger	Enable the triggering system.
Bit 2	Enable internal time trigger	Enable the internal time trigger to operate.
Bit 3	*unused*	
Bit 4	Float64 results	1 => Results are Float64, 0 => Results are Float32
Bit 5	Sampler	1 => Sampler, 0 => Integrator
Bit 6	Mean	1 => Mean is measured for 2^20 samples.
Bit 7	Loopback	For memory diagnostic purpose. Data written to BAR2 may be read back on the same access port.

Acquisition and Trigger must be started before any triggers can be sent. It means also that the Self Trigger cannot be started simultaneously with the two others. Otherwise, the first partial integral might be wrong. Doing this initialization by writing two successive values will guarantee enough time for the system to settle.

Interrupt Enable (BAR3 : 0x102)	
Bit 0	Enable status bit 0 condition
Bit 1	Enable status bit 1 condition
Bit 2	Enable status bit 2 condition
Bit 3	Enable status bit 3 condition
Bit 4	Enable status bit 4 condition
Bit 5	Enable status bit 5 condition
Bit 6	Enable status bit 6 condition
Bit 7	Global interrupt enable

Interrupt Pending (BAR3 : 0x103)	
Bit 0	Status bit 0 condition is pending
Bit 1	Status bit 1 condition is pending
Bit 2	Status bit 2 condition is pending
Bit 3	Status bit 3 condition is pending
Bit 4	Status bit 4 condition is pending
Bit 5	Status bit 5 condition is pending
Bit 6	Status bit 6 condition is pending

Bit 7	*unused*

Gain configuration (BAR3 : 0x104)

Register value	Gain
0	0.1
1	0.2
2	0.4
3	0.5
4	1.0
5	2.0
6	4.0
7	5.0
8	10.0
9	20.0
10	40.0
11	50.0
12	100.0
13 to 15	0.1

Any other value could result in a gain change and should be avoided.

Signal path and voltage reference configuration (BAR3 : 0x105)

Bit 6	Override gain linking By default, the voltage reference is automatically set according to the value specified in the Gain Configuration Register, i.e. the voltage reference is linked to the gain. However, you may force the system to use the value written in this register by forcing this bit to '1'. This way, you may take control over the voltage fed to the input resistor ladder. Please note that this voltage is applied to the positive input of the resistor ladder and the same voltage, but negative, is applied to the negative input of the said resistor ladder.
Bit 5..2	Voltage reference value 0000 : 12.5 [Volts] 0001 : 12.5 [Volts] 0010 : 10.0 [Volts] 0011 : 10.0 [Volts] 0100 : 5.0 [Volts] 0101 : 2.5 [Volts] 0110 : 1.25 [Volts]

	0111 : 1.0 [Volts]	
	1000 : 0.5 [Volts]	
	1001 : 0.25 [Volts]	
	1010 : 0.125 [Volts]	
	1011 : 0.1 [Volts]	
	1100 : 0.05 [Volts]	
Bit 1..0	Select the signal that is fed to the input	
	00 : Direct coil connection	
	01 : Input is short-circuited	
	10 : Voltage reference	
	11 : Voltage reference	

External Trigger configuration (BAR3 : 0x106)

Bit 1..0	Active edge configuration	00 : Disabled
		01 : Rising edge
		10 : Falling edge
		11 : Both edges

Bi-Phase decoder configuration (BAR3 : 0x107)

Bit 7..6	Sync configuration	00 : Disabled
		01 : Rising edge
		10 : Falling edge
		11 : Both edges
		When set to the "Disabled" state, the Sync signal acts as the Error input of the decoder. The Error input polarity is set in bit 3
Bit 5	Bi-Phase encoder Index type	0 : No index present
		1 : Index present
Bit 4	Bi-Phase input signal mode	0 : Differential
		1 : Single ended
Bit 3..0	Bi-Phase input signal configuration	Bit 0 : A when set to 0, /A when set to 1
		Bit 1 : B when set to 0, /B when set to 1
		Bit 2 : I when set to 0, /I when set to 1
		Bit 3 : E when set to 0, /E when set to 1

Data processing reported events (BAR3 : 0x108, Read only)	
Bit 0	Partial integral accumulator has overflowed.
Bit 1	Partial timestamps accumulator has overflowed.
Bit 2	Float32 result has overflowed on its exponent part.
Bit 3	Float64 result has overflowed on its exponent part.

Acquisition reported events (BAR3 : 0x109)	
Bit 0	Acquisition done.
Bit 1	ADC was reset. Indicates a serious problem.
Bit 2	ADC sampling is too fast. A sampling request was received while the current conversion operation was still in progress.
Bit 3	Trigger rate is too fast. Your trigger rate is higher than the sampling rate.
Bit 4	Error detected on Bi-Phase encoder: the Error signal is active. Its activity state depends on the configuration. See register "Hardware configuration"
Bit 5	Error detected on Bi-Phase decoder: edge count incorrect at index. Writing a value again in the "Bi-Phase decoder: maximum count" register will reset this bit.
Bit 6	Positive over-range detected. A write cycle clears this bit.
Bit 7	Negative over-range detected. A write cycle clears this bit.

FIFO status (BAR3 : 0x10A, Read only)	
Bit 0	Data FIFO is not empty. Data must be collected.
Bit 1	16-Words Transfer may be performed. Data FIFO contains more than 16 32-Bit words. When this condition is set, you may try to get data by burst. However, with a PC computer, this is a very tedious task.
Bit 2	Data FIFO over-run. Data acquisition is faster than your data collecting process. Data have been lost.
Bit 3	Data FIFO under-run. You read more data than the FIFO content.
Bit 4	Trigger FIFO full. You must wait before providing a new trigger configuration.
Bit 5	Trigger FIFO overrun. You provided too much trigger configuration. Trigger configurations were lost.

FIFO status (BAR3 : 0x10B, Read only)	
Bit 0	Synchro
Bit 1	Trigger

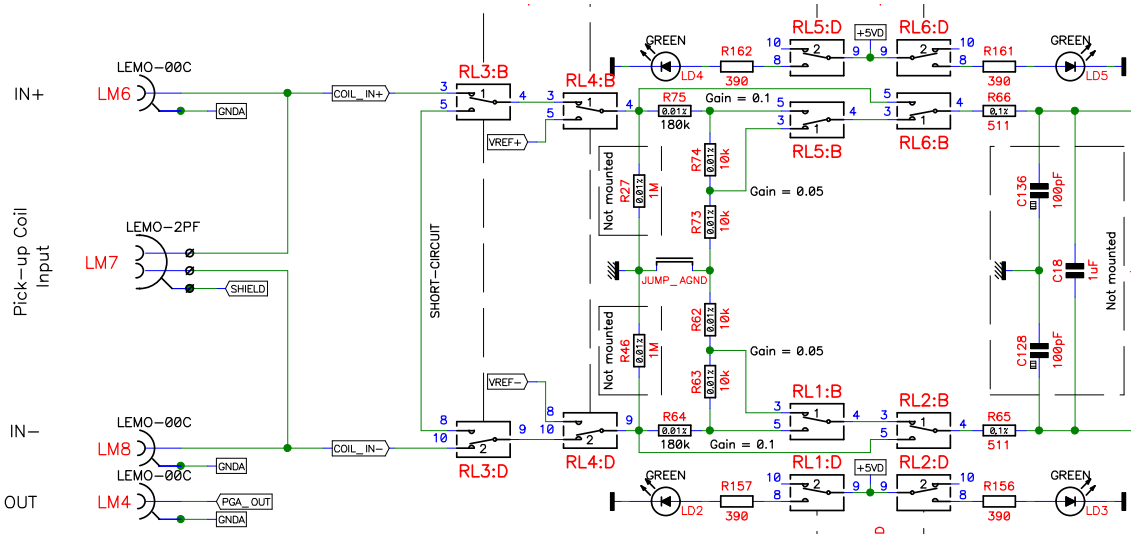
PXI Star trigger enable and configuration (BAR3 : 0x10C, For PXI Star only).		
Bit 12..0	Bit numbering is directly related to device slot numbering. When set, the related device will receive a trigger pulse from the STAR Trigger.	
Bit 17..16	PXI trigger delay / Sampler delay	For board configured as PXI Star: sampler delay. For board configured as PXI Device: PXI trigger delay.

Sampling rate period (BAR3 : 0x120)
Expressed as a multiple of the system clock.

Internal time trigger rate period (BAR3 : 0x124)
Expressed as a multiple of the system clock.

5-3 INPUT ADAPTATION

The "IN+" and "IN-" coil inputs can be optimized for your coil signals. The schematic diagram below shows an excerpt from the FDI2056 input circuitry.



- R27 and R46 can be mounted to tie IN+ and IN-, respectively, to ground. By default, these are not mounted, so that both are floating inputs.
- C18 can be used to filter common-mode HF noise. By default, this is used as an anti-aliasing filter for the ADC, with a cut-off at around 10 kHz.
- C136 and C128 can be used to filter HF noise on IN+ and IN-, respectively, relative to ground. By default, these are not mounted.

! CAUTION

⇒ These modifications are to be performed only by professional electronics technicians. Unprofessional workmanship may destroy your card and will void your warranty. Metrolab will happily perform any necessary modifications for a nominal service fee, and free of charge if you state your requirements at the time of ordering.

5-4 ADDING A SUPPLEMENTARY INTEGRATOR CHANNEL

The FDI2056 usually comes with one channel but can be equipped with up to three integrator channels on the standard Metrolab crate and up to nine integrator channels with the extended crate. To install an FDI2056 integration card in the crate, please follow the instructions bellow carefully.

! CAUTION

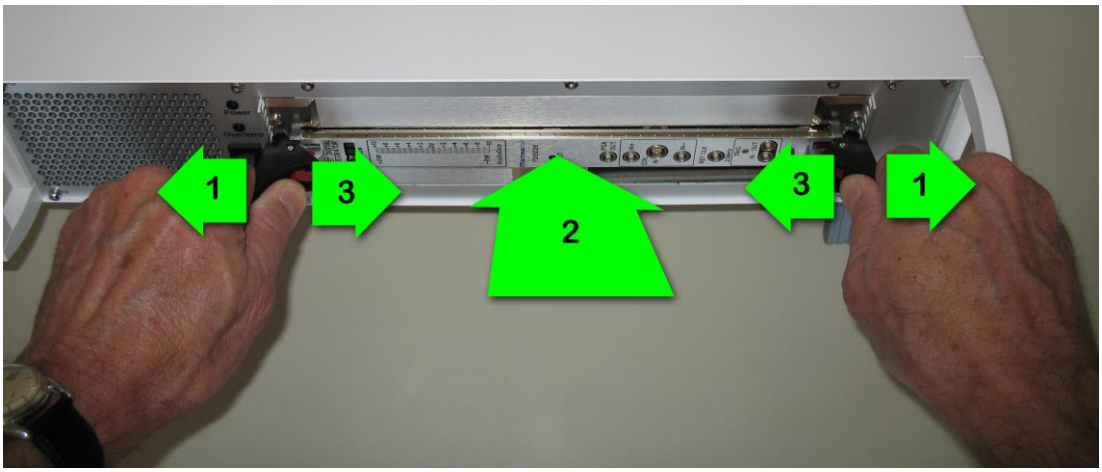
⇒ The FDI2056 electronics supports hot-swapping, but the firmware will not start up correctly. We recommend powering down the crate.

⇒ Handle the FDI2056 channel card with caution to avoid damage due to electrostatic discharge. Ground yourself before handling the card; the best procedure is to use a grounded wrist-strap during installation.

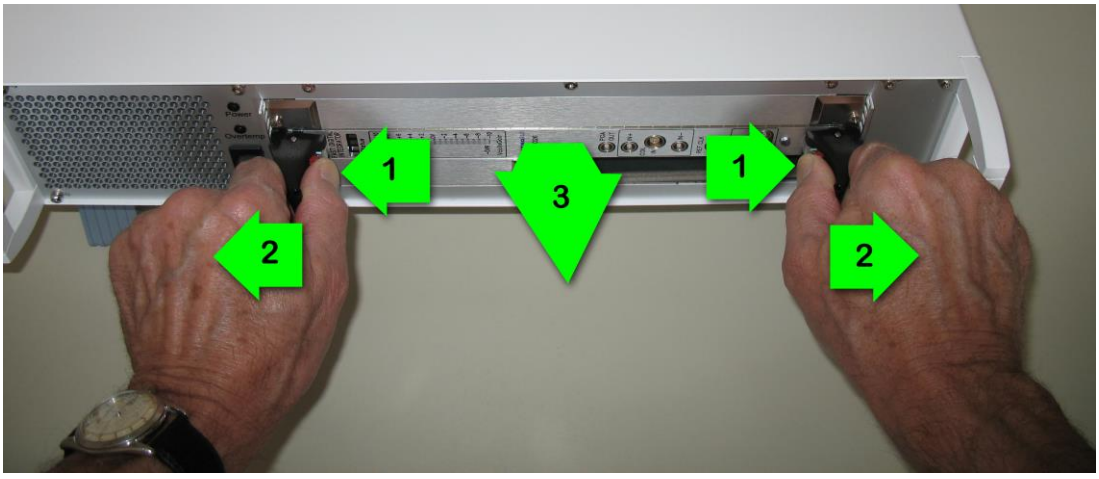
⇒ The FDI2056 channel card has components on the underside that just barely clear the faceplate of adjacent cards. Exercise extreme caution in order not to damage the card when you slide it into the crate. Slide it in slowly, and lift it slightly to keep the components from snagging.

NOTICE

⇒ PXI cards use a locking card extractor. When inserting the card: ①Push the levers outward; ② Push the card in until the black plastic extractor levers touch the crate; and ③ Push the levers inward until you hear both locks click.



⇒ When removing a card: ① Press the red lock levers with your thumbs; ② Push the black extractor levers outward to extract the card; and ③ Gently pull the card out of the slot. See Caution note above.

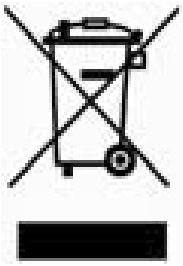


⇒ To provide additional assurance that the card does not vibrate loose, for example when shipping, you can tighten the screws behind the extractor lever.

5-5 WARRANTY, CALIBRATION, CERTIFICATION AND MAINTENANCE

Warranty	2 years
Recommended calibration interval:	12 months
Maintenance	None

NOTICE

⇒ 

This product conforms to the WEEE Directive of the European Union (2002/96/EC) and belongs to Category 9 (Monitoring and Control Instruments). For proper environment friendly disposal, you can return the instrument free of charge to us or our local distributor.