Document No: PD-3061-9374	Issue 3	Date: 7th December 2017

Document Title:

FlexComms Probe Adaptor User Guide

Summary / Scope:

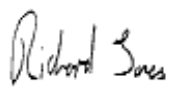
Describes high-level functions and comms address map for the FlexComms probe Adaptor (FCA).

Top level board revision A-3061-0815-02
FPGA factory version 1.0
FPGA application version v2.0

Reason for Issue / Nature of Change :
See revision sheet on page 2

Distribution:
\\renishaw\global\GB\PLC\CMMPD\Data\CMM\Projects\3061 - REVO probes\Controlled Documents - BLUE\Project Docs (PD)\FlexComms Adaptor

07/12/2017

X 

Author: Richard Jones
Senior Electronics Design Engineer
Signed by: Richard Jones 2

07/12/2017

X 

Reviewed: Ellie Burnett
Development Test Engineer
Signed by: Ellie Burnett

Contents

Revision sheet

Issue	Detail	Author	Date
1	First issue. Code version v1.0	Richard Jones	17th May 2016
2	Added in additional probe X, Y and Z objects with increased range. Added registers for current calibration and discussion (section 6). Added registers for Hardware ID and board temperature. Code version v1.9 (application) and v1.0 (factory)	Richard Jones	20th April 2017
3	Corrected reference to REVO-2 Probe Interface specification. Added DPR object 0x092A EEPROM Ident Added new DPR objects 0x0997, 8 and 9 for raw current sensing Application code version changed to 2.0	Richard Jones	7th December 2017

References

[1] R. Jones, “PD-3061-9364-02 FlexComms Adaptor Specification,” August 2015.

[2] R. Toller, “PD-5759-9016-01 REVO-2 Processing Board FPGA Datasheet,” February 2013.

[3] R. Toller, “PD-5759-9010-03 REVO-2 Probe Interface Specification,” April 2013.

[4] Renishaw plc., “H-1000-7560-03-A High speed communications cable wiring instructions,” January 2015.

[5] R. Jones, “C-3061-0810-01 FCA Schematic,” July 2015.

[6] R. Toller, “PD-3007-9131-10 Serial Communications Scheme,” August 2015.

[7] R. Toller, “PD-3007-9176-01 FlexComms Node Datasheet,” July 2015.

[8] Betatherm, “10K3A542i Datasheet,” May 2012. [Online]. Available: http://compeng/Manufacturer-Datasheets/Betatherm/Datasheets/Betatherm_DS_10K3A542i.pdf.

[9] R. Toller, “PD-3007-9161-02 Gyro/PH20 Stator FPGA Datasheet,” November 2010.

[10] Altera Corporation, “Altera ASMI Parallel IP Core User Guide,” December 2014. [Online]. Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_altasmi_parallel.pdf.

[11] Altera Corporation, “Altera Remote Update IP Core User Guide,” 17th November 2015. [Online]. Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_altremote.pdf. [Accessed 15th April 2016].

[12] R. Price, “PD-3061-9100-10 Active Head Probe Types and Serial Numbers,” July 2015.

[13] Linear Technology Corporation, “LTC2945 Wide Range I2C Power Monitor LT 0915 rev. B,” 2012. [Online]. Available: http://cds.linear.com/docs/en/datasheet/2945fb.pdf. [Accessed August 2016].

[14] Bourns, “CRL Series - Low Value Chip Resistors,” November 2015. [Online]. Available: http://www.bourns.com/docs/Product-Datasheets/CRL.pdf. [Accessed August 2016].

1 Introduction

The FlexComms Probe Adaptor (FCA) [1] is a test rig which interfaces to all types of active head or REVO probes. It is based on the REVO-2 processing board electronics and facilitates probe testing without the need for a CMM head. A photograph of the FCA is shown in Figure 1.

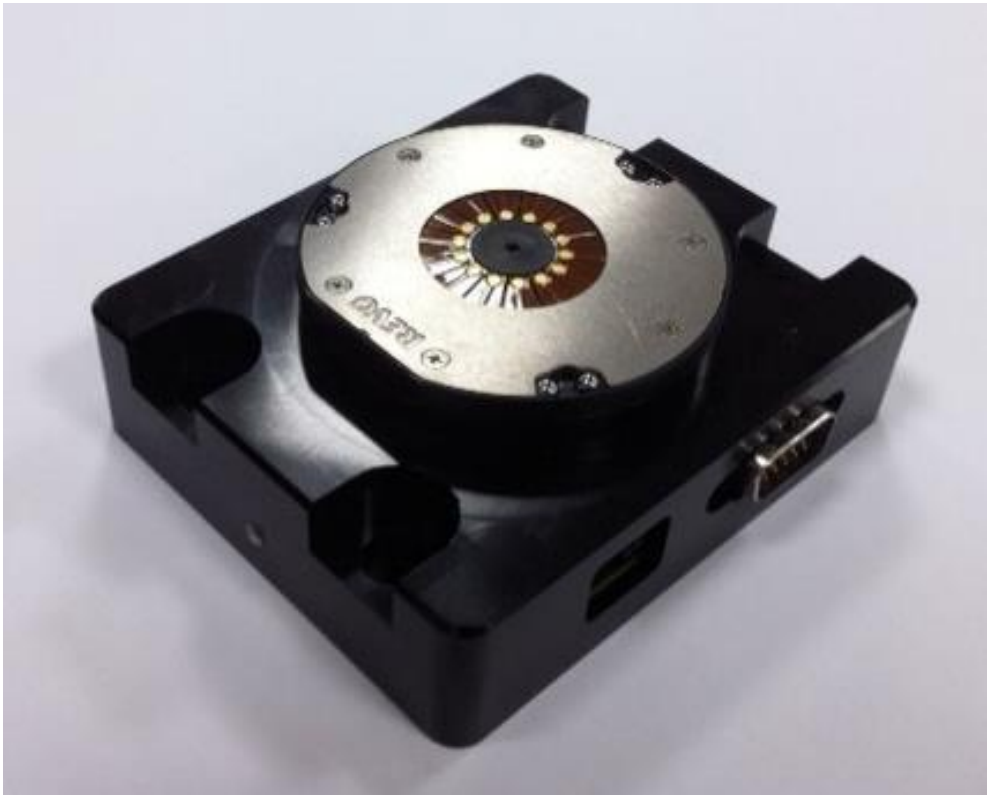


Figure 1: The FlexComms Probe Adaptor (FCA)

This document describes the functionality of the FCA and is intended to be a comprehensive user guide. It does not provide information on how it works or the rationale behind the design. It is largely based on [2].

2 External Connections

A diagram showing the external interfaces is shown in Figure 2, while an explanation is given in Table 1. Note that aside from the address switch and LED there is no user interface as the FCA is controlled entirely through its FlexComms connection.

Figure 2: FCA External Interfaces

Interface	Description
Status LED	Bi-colour red/green LED which is controlled through FlexComms (see section 5.20)
Probe Interface	Interface to a REVO probe (see [3] for information).
FlexComms Master	Connects to a FlexComms manager node using a standard orange cable [4].
Upstream FlexComms	Connects to a Downstream FlexComms port of another FCA using a standard RJ45 ethernet cable. Either this or the FlexComms Master connection must be present for the FCA to operate.
JTAG Programming Port	Enables programming and diagnostic access to the FPGA. Uses a REVO-2 programming connector (P-CN10-0070, see [5])
Downstream FlexComms	Connects to another FCA using a standard RJ45 ethernet cable
Address Switch	Sets the address of the FCA. User configurable.

Table 1: List of External Interfaces

3 Connecting Multiple FCAs Together

The FCA contains separate downstream and upstream FlexComms ports which enable multiple units to be connected to and controlled through the same FlexComms bus. A diagram which shows how this is done is shown in Figure 3.

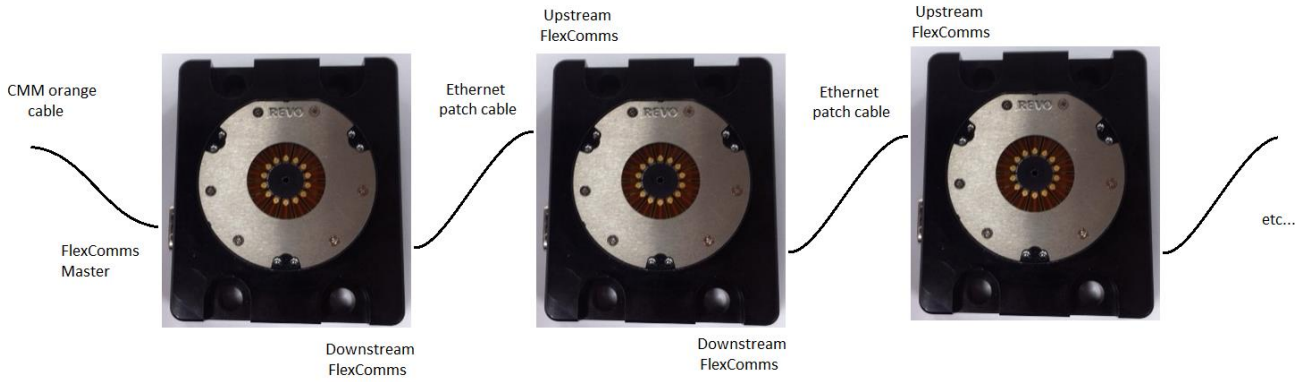


Figure 3: Connections between Multiple FCAs

Up to ten FCAs can be connected in a single chain, providing each unit has its address switch set to a different position. In addition, if digital probes are used, each probe needs to have a different address which must not coinside with any FCA address.

When creating a system comprising multiple FCAs and probes, make sure that the total power requirement does not exceed the maximum available. Table 2 lists the potential components of a system and the amount of power each item produces or consumes.

Source	Power Available (W)
FlexComms Master (either 5-axis PCI card or FlexComms USB Interface)	30.0
Sink	**Maximum Power Consumption (W)**
FlexComms Probe Adaptor	1.0
Low Voltage Probes (e.g. RSP2)	3.5
High Voltage Probes • Surface Finish Probe 2 (SFP2)	10.0 3.3

Table 2: List of power sources and sinks to be considered when designing a system.

4 FlexComms Object Address Map

A system comprising multiple FCAs and probes is controlled through a single FlexComms connection. For more information on how this interface works refer to [6].

This section lists the object addresses in the FCA and describes their functionality. Objects are accessed as 16-bit words.

Register addresses marked with an asterisk (*) are not available in the factory version of the firmware.

Table 3: FCA object address map. * ro = read only, rw = read/write.

address	object name	r/w*
0x0000	NODE_TYPE = 0x000F	ro
0x0001	FPGA_VERSION_MAJOR	ro
0x0002	FPGA_VERSION_MINOR	ro
0x0003	reserved	-
0x0004	FPGA_DELAY_D1	ro
0x0005	FPGA_DELAY_D2	ro
0x0006	FPGA_DELAY_D3	ro
0x0007	RX_BUFFER_SIZE	ro
0x0008	TX_BUFFER_SIZE	ro
0x0009	CLOCK_FREQUENCY_HI	ro
0x000A	CLOCK_FREQUENCY_LO	ro
0x000B	CYCLIC_RESPONSE_DELAY_HI	rw
0x000C	CYCLIC_RESPONSE_DELAY_LO	rw
0x000D	SYNC_ACQUISITION_DELAY	rw
0x000E	COMMS_CONTROL	rw
0x000F	RX_COUNT_HI	ro
0x0010	RX_COUNT_LO	ro
0x0011	TX_COUNT_HI	ro
0x0012	TX_COUNT_LO	ro
0x0013	ERROR_COUNT_HI	ro
0x0014	ERROR_COUNT_LO	ro
0x0015	reserved	-
0x0016	reserved	-
0x0017	reserved	-
0x0018	FORCE_ERROR_COUNT	rw
0x0019	PD_DEST_ADDRESS	rw
0x001A	TX_BLOCK_SIZE	rw
0x001B	RX_BLOCK_BUFFER_ADDRESS	rw
0x001C	TX_BLOCK_BUFFER_ADDRESS	rw
0x001D	CMD_RESPONSE_DELAY_HI	rw
0x001E	CMD_RESPONSE_DELAY_LO	rw
0x001F	reserved	-
	transmit process data node address list	
0x0020	transmit node 0 node address	rw
0x0021	transmit node 1 node address	rw
…		
0x005F	transmit node 63 node address	rw
0x0060	transmit node 0 transmit process data map address	rw
0x0061	transmit node 1 transmit process data map address	rw
…		
0x009F	transmit node 63 transmit process data map address	rw
	transmit process data maps	

	transmit node 0 process data map	
0x00A0	transmit process data object 0	rw
0x00A1	transmit process data object 1	rw
…	….	
0x00AF	transmit process data object 15	rw
0x00B0…	**transmit node 1 process data map…**	
0x00C0…	**transmit node 2 process data map…**	
…		
0x0490…	**transmit node 63 process data map…**	
	receive process data node address list	
0x04A0	receive node 0 node address	rw
0x04A1	receive node 1 node address	rw
…		
0x04DF	receive node 63 node address	rw
0x04E0	receive node 0 receive process data map address	rw
0x04E1	receive node 1 receive process data map address	rw
…		
0x051F	receive node 63 receive process data map address	rw
	receive process data maps	
	receive node 0 process data map	
0x0520	receive process data object 0	rw
0x0521	receive process data object 1	rw
…		
0x052F	receive process data object 15	rw
0x0530…	**receive node 1 process data map…**	
0x0540…	**receive node 2 process data map…**	
…		
0x0910…	**receive node 63 process data map…**	
0x0920	SERIAL_NUMBER_CHAR_01	rw
0x0921	SERIAL_NUMBER_CHAR_23	rw
0x0922	SERIAL_NUMBER_CHAR_45	rw
0x0923*	VPROBE VOLTAGE	ro
0x0924*	VPROBE CURRENT	ro
0x0925*	+5VL VOLTAGE	ro
0x0926*	+5VL CURRENT	ro
0x0927*	-5V VOLTAGE	ro
0x0928*	-5V CURRENT	ro
0x0929	HARDWARE_IDENT	ro
0x092A	EEPROM IDENT	rw
0x092B	STATUS	ro
0x092C	latched STATUS	ro
0x092D	CONTROL	rw
0x092E	EEPROM_CONTROL	rw
0x092F	spare	
0x0930	BOARD_TEMPERATURE	ro
0x0931	1V2 SENSE	ro
0x0932	3V3 SENSE	ro
0x0933	V_THERMISTOR	ro
0x0934	2V5 SENSE	ro
0x0935*	PROBE_X_EXTENDED_RANGE	ro
0x0936	spare	
0x0937*	PROBE_Y_ EXTENDED _RANGE	ro
0x0938	spare	

0x0939*	PROBE_Z_ EXTENDED _RANGE	ro
0x093A	STATUS LED CONTROL	rw
0x093B	spare	
0x093C*	PROBE_VOLTAGE_CONTROL	rw
0x093D	PROBE_STATUS	ro
0x093E*	VPROBE_CURRENT_GAIN	rw
0x093F*	VPROBE_CURRENT_OFFSET	rw
0x0940*	PLUS_5VL_CURRENT_GAIN	rw
0x0941*	PLUS_5VL_CURRENT_OFFSET	rw
0x0942*	NEG_5V_CURRENT_GAIN	rw
0x0943*	NEG_5V_CURRENT_OFFSET	rw
0x0944 *	CALIBRATION_WEEK	rw
0x0945*	CALIBRATION_YEAR	rw
0x0946	FLASH_ADDRESS_HI	ro
0x0947	FLASH_ADDRESS_LO	ro
0x0948	FLASH_CONTROL	rw
0x0949	FLASH_STATUS	ro
0x094A	FLASH_READ_BYTE	ro
0x094B	ASMI_STATUS	ro
0x094C	FLASH_PROTECT	rw
0x094D	FIRMWARE_UPDATE_CONTROL	rw
0x094E	RU_USER_DATA_HI	ro
0x094F	RU_USER_DATA_LO	ro
0x0950	RU_USER_PARAM_ETC	rw
0x0951	FLASH_APP_FIRMWARE_ADDRESS_HI	ro
0x0952	FLASH_APP_FIRMWARE_ADDRESS_LO	ro
0x0953*	PROBE_X	ro
0x0954	spare	
0x0955*	PROBE_Y	ro
0x0956	spare	
0x0957*	PROBE_Z	ro
0x0958	FIRST_FREE_FLASH_ADDRESS_HI	ro
0x0959	FIRST_FREE_FLASH_ADDRESS_LO	ro
0x095A	FLASH_NUMBER_OF_SECTORS	ro
0x095B	spare	
0x095C	FLASH_SECTOR_SIZE_HI	ro
0x095D	FLASH_SECTOR_SIZE_LO	ro
0x095E*	PROBE_SAMPLE_DELAY	ro
0x095F	NODE TYPE TO EEPROM	rw
0x0960	ADDRESS TO EEPROM	rw
0x0961	BYTE TO EEPROM	rw
0x0962	BYTE FROM EEPROM	ro
0x0963 – 0x098B	spare	
0x098C*	PROBE_EEPROM_BYTE01	ro
0x098D*	PROBE_EEPROM_BYTE23	ro
0x098E*	PROBE_EEPROM_BYTE45	ro
0x098F*	PROBE_EEPROM_BYTE67	ro
0x0990*	PROBE_EEPROM_BYTE89	ro
0x0991*	PROBE_EEPROM_BYTEAB	ro
0x0992*	PROBE_EEPROM_CONTROL	rw
0x0993	spare	
0x0994*	PROBE_EEPROM_ADDRESS	rw
0x0995*	PROBE_EEPROM_WRITE_DATA	rw
0x0996*	PROBE_EEPROM_READ_DATA	ro

0x0997	VPROBE_RAW_CURRENT	Ro
0x0998	PLUS_5VL_RAW_CURRENT	Ro
0x0999	NEG_5V_RAW_CURRENT	ro
0x099A	First Free Address	
…	…	…
0x199F	End of Dual Port RAM	

4.1 Object storage

Object storage in the FPGA is divided into three regions: registers, RAM and dual-port RAM (DPR). Register objects are read from addresses 0x0000-0x000A. RAM objects are accessed from address 0x000B-0x091F. DPR objects are accessed from address 0x0920 onwards. See Figure 4.

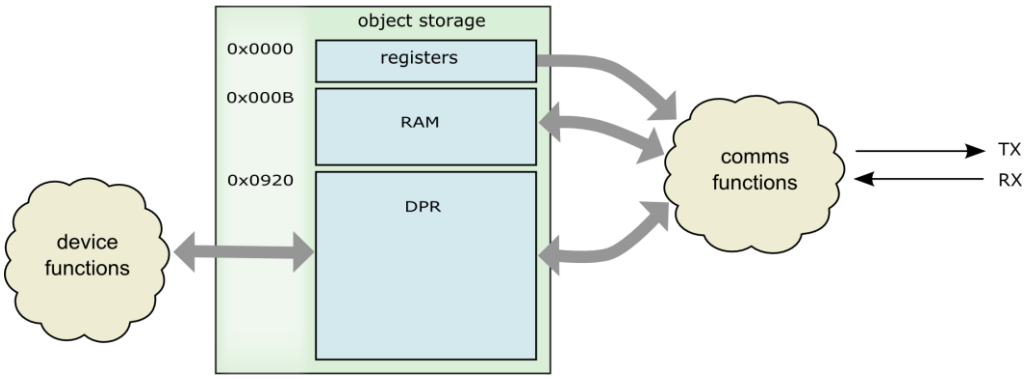


Figure 4: object storage

Objects stored in registers are read-only: they are defined as constants in the source code and cannot be modified. Examples of register objects are node type and clock frequency.
Objects stored in RAM are accessed by the comms functions only. Examples of objects stored in RAM are transmit and receive counts and all process data mapping objects. Objects stored in DPR are accessible by both device and comms functions.

4.2 FlexComms node addresses

The 'Register' and 'RAM' addresses, 0x0000 to 0x091F, are common to all FlexComms nodes. For a description of these addresses see reference [7].

5 Object descriptions

5.1 0x0920 – 0x0922 SERIAL_NUMBER_CHAR_xx

The FCA's serial number is stored as 6 ASCII character codes.

Table 4: head serial number

SERIAL_NUMBER_CHAR_01		SERIAL_NUMBER_CHAR_23		SERIAL_NUMBER_CHAR_45	
bits 31-16	bits 15-0	bits 31-16	bits 15-0	bits 31-16	bits 15-0
char 0	char 1	char 2	char 3	char 4	char 5

Example:
The serial number "4J0267" is stored as follows

Table 5: head serial number example

SERIAL_NUMBER_CHAR_01		SERIAL_NUMBER_CHAR_23		SERIAL_NUMBER_CHAR_45	
bits 31-16	bits 15-0	bits 31-16	bits 15-0	bits 31-16	bits 15-0
'4'=0x34	'J'=0x4A	'0'=0x30	'2'=0x32	'6'=0x36	'7'=0x37

The FCA serial number is stored in a serial EEPROM. The FPGA reads the serial number from the EEPROM on power-up. Writing a new serial number via objects SERIAL_NUMBER_CHAR_XX writes to the FPGA's copy. The FPGA's copy of the serial number is stored in the EEPROM via the EEPROM store command – see section 5.13.

5.2 0x0923 VPROBE VOLTAGE

The measured value of +VPROBE voltage, with a resolution of 25 mV and update rate of 0.3 s. This supply can be switched between 0 V, 5 V and 20 V (see section 5.21).

5.3 0x0924 VPROBE CURRENT

The calibrated +VPROBE current with a resolution of 0.01 mA and update period of 0.3 s. The calibrtation parameters can be read and set through registers 0x093E and 0x093F, see section 6.

5.4 0x0925 +5VL VOLTAGE

The measured value of +5VL voltage with a resolution of 25 mV and update period of 0.3 s. This supply can be switched on and off, see section 5.12.

5.5 0x0926 +5VL CURRENT

The calibrated +5VL current with a resolution of 0.01 mA and an update period of 0.3 s. The calibrtation parameters can be read and set through registers 0x0940 and 0x0941, see section 6.

5.6 0x0927 -5V VOLTAGE

The measured value of the -5V voltage with a resolution of 25 mV and update period of 0.3 s

5.7 0x0928 -5V CURRENT

The calibrated -5V current with a resolution of -0.01 mA and update period of 0.3 s. The calibrtation parameters can be read and set through registers 0x0942 and 0x0943, see section 6.

5.8 0x0929 HARDWARE_IDENT

Returns the hardware identifier for currnet PCB. This is programmed by four 0Ω links and is set to 0x1 on the '-02' version of the PCB.

5.9 0x092A EEPROM Identifier

Returns the current EEPROM or variant identifier for the FPGA (eight-bits). It is set during during commissioning to one of the values show in Table 6. It can be updated by using the EEPROM control word

Table 6: Value definition for the EEPROM variant

Value (hex)	Definition
0	01 issue PCB
1	01 issue PCB which has been modified
2	02 issue PCB

5.10 0x092B STATUS

Status bits are defined in Table 7.

Table 7: status bits

bit	description
0	probe in analogue mode
1	probe in digital mode
2	probe present
3	probe laser stable
4	EEPROM write in progress
5	latched probe error
6	spare ('0')
7	application/factory firmware
8..15	spare ('0')

5.10.1 STATUS bit 0: probe in digital mode

Indicates when the FlexComms Probe Adaptor is in digital mode. At power-up this is controlled automatically, but it can be overridden using bits 0 and 1 of the CONTROL word.

5.10.2 STATUS bit 1: probe in analogue mode

Indicates when the FlexComms Probe Adaptor is in analogue mode. At power-up this is controlled automatically, but it can be overridden using bits 0 and 1 of the CONTROL word.

5.10.3 STATUS bit 2: probe present

The *probe present* bit is set to '1' when the REVO-2 head has detected that a probe is attached, and '0' otherwise. See [2], section 6.

5.10.4 STATUS bit 3: probe laser stable

A PROBE_OK signal is generated in an analogue probe (e.g. RSP2V2, RSP3): '1' indicates "probe OK", '0' indicates "probe not OK". The *laser stable* bit is a modified version of the probe OK signal, derived as follows:

laser stable bit = PROBE_OK signal ***and*** PROBE_LASER_ENABLE ***and*** PROBE_PRESENT.

PROBE_LASER_ENABLE is an internal FPGA signal set via control word bits 2 and 6 (see section 5.12), and is '1' when the probe has been enabled and '0' otherwise.

In other words, the *laser stable* bit is forced to '0' when the probe laser is not switched on.

5.10.5 STATUS bit 4: EEPROM write in progress

The *EEPROM write in progress* bit is set to '1' during an EEPROM write and '0' otherwise. EEPROM writes are described in section 5.13.

5.10.6 STATUS bit 5: latched probe error

The *latched probe error* bit is set to '1' when the PROBE_OK signal is '0'. When this condition is not present, it is cleared to '0' when '1' is written to the *cleared latch probe errors* bit in the CONTROL word (5.12.6).

5.10.7 STATUS bit 7: application/factory firmware

The *application/factory firmware* bit is set to '0' if the FPGA is configured with the factory default firmware image and '1' if configured with an in-field programmed 'user' image. In all normal circumstances the FPGA will be configured with a user image. If an in-field update fails, the FPGA configuration will revert to the factory image.

5.11 0x092C Latched STATUS

Latched STATUS is the value of the status register latched *S* + *d1* clock cycles after reception of a sync frame, where *S* is the SYNC_ACQUISITION_DELAY and *d1* is the FPGA_PROCESS_DELAY.

5.12 0x092D CONTROL

CONTROL bits are defined in Table 8.

Table 8: CONTROL bits

bit	description
0	enable analogue mode
1	enable digital mode
2	probe enable request
3	spare
4	initialise probe
5	spare
6	probe disable request
7	clear latched probe errors
8..14	spare
15	reset comms counters

All of the control word bits are self-clearing – i.e. they do not need to be set to '0' after having been set to '1'.

5.12.1 CONTROL bit 0: enable analogue mode

Setting the *enable analogue mode* bit to '1' overrides the automatic probe type detection and puts the FCA into analogue mode. It can only be returned to automatic mode by power cycling.

5.12.2 CONTROL bit 1: enable digital mode

Setting the *enable digital mode* bit to '1' overrides the automatic probe type detection and puts the FCA into digital mode. It can only be returned to automatic mode by power cycling.

5.12.3 CONTROL bit 2: probe enable request

Setting the *probe enable request* bit to '1' switches on the laser in an analogue probe (e.g. RSP2). Setting bit 6 (*probe disable request*) to '1' switches off the laser.

5.12.4 CONTROL bit 4: 14nitialize probe

Setting the *initialise probe* bit to '1' causes the *probe initialisation complete* bit in the probe status word to be reset to '0' and the probe serial number to be read from the probe EEPROM – see section 5.51. Once the serial number has been read, the *probe initialisation complete* bit is set to '1'.

5.12.5 CONTROL bit 6: probe disable request

Setting the *probe disable request* bit to '1' switches off the laser in an analogue probe (e.g. RSP2).

5.12.6 CONTROL bit 7: clear latched probe errors

Setting the *clear latched probe errors* bit to '1' clears the *latched probe error* bit in the status register – see section 5.10.5.

5.12.7 CONTROL bit 15: reset comms counters

Writing '1' to *reset comms counters* resets the FlexComms transmit and receive frame counts and error counters to zero.

5.13 0x092E EEPROM_CONTROL

The EEPROM_CONTROL register is used to write parameters to non-volatile EEPROM. Bits are automatically cleared after writing.

Table 9: EEPROM_CONTROL bits

bit	description
0	write new node type
1	write new serial number
2	write new current calibration values
3	write new calibration date
4	write new EEPROM identifier
5	spare
6	write byte to EEPROM
7	read byte from EEPROM

5.13.1 EEPROM CONTROL bit 0: write new node type

Setting this to '1' writes the value stored in 0x095F NODE TYPE TO EEPROM to EEPROM. This defines the node type of the device.

5.13.2 EEPROM CONTROL bit 1: write new serial number

Writes the value stored in 0x0920 – 0x0922 SERIAL_NUMBER_CHAR_xx to the EEPROM.

5.13.3 EEPROM CONTROL bit 2: write new current calibration values

Writes the current gain and offset values stored in registers 0x093E to 0x0943 to EEPROM.

5.13.4 EEPROM CONTROL bit 3: write new current calibration date

Writes the calibration date stored in registers 0x0944 and 0x0945 to EEPROM.

5.13.5 EEPROM CONTROL bit 4: write new EEPROM identifier

Writes the EEPROM identifier byte stored in registers 0x092A to EEPROM.

5.13.6 EEPROM CONTROL bit 6: write byte to EEPROM

Writes the value stored in location 0x0961 BYTE TO EEPROM to the location specified by 0x0960 Address to EEPROM.

5.13.7 EEPROM CONTROL bit 7: read byte from EEPROM

Reads the EEPROM data from address specified in 0x0960 ADDRESS TO EEPROM and puts it in location 0x0962 BYTE FROM EEPROM.

5.14 0x0930 BOARD_TEMPERATURE

Returns the PCB temperature as measured by an on-board MAX6629 digital temperature sensor. The result is in a 12-bit + sign format with a resolution of 0.0625°C/LSB and an accuracy of ±1°C. Examples of different temperatures are shown in

Table 10: Board temperature format

Temperature (°C)	Register contents
25	0x0190
0.0625	0x0001
0	0x0000
-0.0625	0x1FFF
-25	0x1E70

5.15 0x0931 1V2 SUPPLY MONITOR

Returns the voltage of the 1.2V power supply, with a resolution of 1 millivolt/LSB. This supply is used to power the FPGA core.

5.16 0x0932 3V3 SUPPLY MONITOR

Returns the voltage of the 3.3V power supply, with a resolution of 1 millivolt/LSB. This supply is used to power the digital parts of the board including the FPGA IO, transceivers and non-volatile memory.

5.17 0x0933 THERMISTOR VOLTAGE

Returns the voltage, in millivolts, generated by the probe's internal thermistor and associated amplifier in the FCA. The voltage output is linked to resistance by the formula below from which the temperature can be found using reference [8].

$$R_T = \frac{10V_T + 1.903}{5.423 - V_T}$$

5.18 0x0934 2V5 SUPPLY MONITOR

Returns the voltage of the 2.5V analogue power supply, measured in millivolts. This supply is used to power the FPGA's analogue supply.

5.19 0x935, 0x937, 0x939 V PROBE EXTENDED RANGE

This returns the synchronized analogue voltage from the probe amplifier divided by two, giving double the range. For more information see section 5.41.

5.20 0x093A STATUS LED CONTROL

The Status LED control register is used to control the colours and states of the LED on the front face of the FCA.

Table 11: LED control register bits

15..5	4	3	2	1	0
unused	LED colour		LED state		

LED states and colours are specified in Table 12 and Table 13.

Table 12: LED colour

msb	lsb	LED colour
0	0	off
0	1	red
1	0	green
1	1	amber

Table 13: LED state

msb		lsb	LED state
0	0	0	off
0	0	1	not used
0	1	0	flash 4Hz, start with off
0	1	1	flash 4Hz, start with on
1	0	0	flash 0.5Hz, start with off
1	0	1	flash 0.5Hz, start with on
1	1	0	not used
1	1	1	on

When a new value is written to the LED control register to change the flashing state of the LED, the FPGA applies the change synchronously with the 4Hz flash signal, to avoid the appearance of glitches in the LED. This functionality is taken from the PH20 Stator LED [9].

5.21 0x093C PROBE_VOLTAGE_CONTROL

The PROBE_VOLTAGE_CONTROL object allows the automatic control of the probe voltage switches to be overridden.

Table 14: Probe Voltage Control Bits

bit	description
0	probe voltage enable
1	high voltage enable
2..6	reserved
7	enable probe voltage control
8..15	reserved

Table 15: Probe Voltage Control Truth Table

Bit 7	Bit 1	Bit 0	Probe Voltage
0	X	X	automatic control
1	X	0	0V (off)
1	0	1	5 V
1	1	1	20 V (+VHEAD)

Note: some probes, for example RSP2, are damaged when +VHEAD is applied to them. Be sure that the probe attached can tolerate +VHEAD (nominally +20V) before enabling the high voltage setting (20V.)

5.22 0x093D PROBE_STATUS

Probe status bits are defined in Table 16.

Table 16: probe status bits

bit	description
0	spare
1	probe initialisation complete
2	probe EEPROM error
3	probe power on
4	probe high voltage on
5	probe serial number is valid
6..15	spare ('0')

5.22.1 PROBE_STATUS bit 1: probe initialisation complete

When a probe is attached, the FPGA sets the *probe initialisation complete* bit to '0' and carries out an initialisation sequence – see [2], section 6. When the initialisation is complete, the *probe initialisation complete* bit is set to '1'.

5.22.2 PROBE_STATUS bit 2: probe EEPROM error

The *probe EEPROM error* bit is set to '1' when an error is detected reading data from the probe EEPROM – see section 5.45. It is cleared to '0' at the start of the process that reads data from the probe EEPROM. The process that reads data from the probe EEPROM is initiated automatically at power-up, and after that may be started by writing '1' to CONTROL word bit 4. See section 5.51.

5.22.3 STATUS bit 3: probe power on

Indicates whether the probe power supply is currently enabled.

5.22.4 STATUS bit 4: probe high voltage on

When the probe power supply is switched on, this status bit indicates whether it is currently supplying a high voltage (+20V) or normal voltage (+5V).

5.22.5 STATUS bit 5: probe serial number is valid

The *probe serial number is valid* bit is set to '1' if each byte in the probe serial number is an ASCII character in the range 0-9, A-Z (i.e. upper case letters and numeric digits only). See [2], section 6.

5.23 0x093E VPROBE_CURRENT_GAIN

12-bit value that the raw VPROBE current reading gets multiplied by to correct any linear errors. One bit corresponds to a factor of 2^{-8}. This is stored in EEPROM and is updated by setting bit 2 in the EEPROM_CONTROL object. The nominal value is 3200 or 0xC80.

5.24 0x093F VPROBE_CURRENT_OFFSET

8-bit offset applied to the VPROBE current reading to correct for any offset errors. The default value is 128 or 0x80 which allows for negative values of offset to be programmed. This value is stored in EEPROM and is updated by setting bit 2 in the EEPROM_CONTROL object.

5.25 0x0940 PLUS_5VL_CURRENT_GAIN

12-bit value that the raw PLUS_5VL current reading gets multiplied by to correct any linear errors. One bit corresponds to a factor of 2^{-8}. This is stored in EEPROM and is updated by setting bit 2 in the EEPROM_CONTROL object. The nominal value is 3200 or 0xC80.

5.26 0x0941 PLUS_5VL_CURRENT_OFFSET

8-bit offset applied to the PLUS_5VL current reading to correct for any offset errors. The default value is 128 or 0x80 which allows for negative values to be set. This value is stored in EEPROM and is updated by setting bit 2 in the EEPROM_CONTROL object.

5.27 0x0942 NEG_5V_CURRENT_GAIN

12-bit value that the raw NEG_5V current reading gets multiplied by to correct any linear errors. One bit corresponds to a factor of 2^{-8}. This is stored in EEPROM and is updated by setting bit 2 in the EEPROM_CONTROL object. The nominal value is 1946 or 0x79A.

5.28 0x0943 NEG_5V_CURRENT_OFFSET

8-bit offset applied to the NEG_5V current reading to correct for any offset errors. The default value is 128 or 0x80 which allows for negative values to be set. This value is stored in EEPROM and is updated by setting bit 2 in the EEPROM_CONTROL object.

5.29 0x0944 CALIBRATION_WEEK

Stores the week number of when the unit was last calibrated as an eight bit number. This value is stored in EEPROM and is updated by setting bit 3 in the EEPROM_CONTROL object.

5.30 0x0945 CALIBRATION_YEAR

Stores the two-digit year of when the unit was last calibrated as an eight bit number. This value is stored in EEPROM and is updated by setting bit 3 in the EEPROM_CONTROL object.

5.31 0x0946, 0x0947 FLASH_ADDRESS_HI, FLASH_ADDRESS_LO

The FLASH_ADDRESS objects specify the high and low words of the 19-bit flash address.

Table 17: Flash address objects

FLASH_ADDRESS_HI						**FLASH_ADDRESS_LO**						
bit 15	…	bit 4	bit 3	bit 1	bit 0	bit 15	bit 14	….	….	…..	bit 1	bit 0
not used			**flash address (0x00000-0x7FFFF)**									

The flash address is used in flash read, write, write protect and erase operations, and should be written before the flash operation is initiated.

5.32 0x0948 FLASH_CONTROL

Flash operations (and the remote update parameter read) are initiated by a write to the FLASH_CONTROL object. Writing a '1' to a flash control bit initiates the operation. The flash control bits are 'self-clearing' – i.e. it is not necessary to write a '0' once the operation is complete. While the flash operation is in progress the *ASMI busy* bit in the flash status word is set to '1'. The *ASMI busy* bit resets to '0' when the flash operation is complete. (For the remote update parameter read, the *remote update busy* bit in the STATUS word behaves in the same way.)

Table 18: FLASH_CONTROL bits

bit	description
0	sector erase
1	page write
2	byte read
3	page read
4	status read
5	write protect
6	remote update parameter read
7..15	reserved

5.32.1 FLASH_CONTROL bit 0: sector erase

Writing '1' to the *sector erase* bit initiates a flash sector erase operation to set all of the bits in the specified flash sector to the value '1' (all bytes set to 0xFF). The sector which contains the address specified by the FLASH_ADDRESS objects is erased, provided the sector has not previously been write-protected. An attempt to erase a write-protected sector will result in the *illegal erase* bit in the FLASH_STATUS word being set when the status word *ASMI busy* bit returns to '0'. A sector erase may take up to 3 seconds to complete.

5.32.2 FLASH_CONTROL bit 1: page write

Writing '1' to the *page write* bit initiates a flash page write operation. The number of 16-bit words specified in the TX_BLOCK_SIZE object are written from RX_BLOCK_BUFFER_ADDRESS to the flash address specified by the FLASH_ADDRESS objects. An attempt to write to a write-protected sector will result in the *illegal write* bit in the FLASH_STATUS word being set when the *ASMI busy* bit returns to '0'. A page write may take up to 5 milliseconds to complete.
Flash write operations change the state of flash bits from '1' to '0', and not vice versa. A flash sector must be erased (set to all '1's) before the pages in it are written.

5.32.3 FLASH_CONTROL bit 2: byte read

Writing '1' to the *byte read* bit initiates a flash byte read operation. After the *ASMI busy* bit has returned to '0' the byte at the address specified by the FLASH_ADDRESS objects can be read from the FLASH_READ_BYTE object

5.32.4 FLASH_CONTROL bit 3: page read

Writing '1' to the *page read* bit initiates a flash page read operation. After the *ASMI busy* bit has returned to '0', the number of 16-bit words specified in the TX_BLOCK_SIZE object are read from flash, starting at flash address specified by the FLASH_ADDRESS objects and written to DPR locations beginning at the address specified by the TX_BLOCK_BUFFER_ADDRESS.
In other words, a flash page read causes $N \times 2$ 8-bit bytes to be copied from flash to N consecutive 16-bit words in DPR, where N is the transmit block size.

5.32.5 FLASH_CONTROL bit 4: status read

Writing '1' to the *status read* bit initiates a flash status read operation. After the *ASMI busy* bit has returned to '0' the flash status byte can be read from the FLASH_READ_BYTE object.

5.32.6 FLASH_CONTROL bit 5: write protect

Writing '1' to the *write protect* bit initiates a flash write protect operation. Block protect bits BP3, BP2 and BP1 in the flash protect register indicate which flash sectors to protect. When a sector is protected, subsequent writes or erases in this sector will be unsuccessful, indicated by *illegal write* or *illegal erase* bits in the flash status word.

5.32.7 FLASH_CONTROL bit 6: remote update parameter read

Writing '1' to the *remote update parameter read* bit initiates a read of a parameter from the Altera remote update megafunction (ALTREMOTE_UPDATE). Information about the parameter to read is specified in the RU_USER_PARAM_ETC object. See section 5.39 for more details.

5.33 0x0949 FLASH_STATUS

FLASH_STATUS indicates the status of the flash interface state machine. Table 19 provides details of the individual bits in the status word.

Table 19: FLASH_STATUS bits

bit	description
0	ASMI busy
1	illegal write
2	illegal erase
3	operation complete
4..15	reserved ('0')

5.33.1 FLASH_STATUS bit 0: ASMI busy

The *ASMI busy* bit set to '1' when the Active Serial Memory Interface (ASMI) begins executing a function, following a write to the flash control word. The bit is set to '0' when the function is complete. Note that there is a delay between sending a command to initiate a flash operation and the reading a '1' in the flash status word *ASMI busy* bit due to the delay incurred by writing the status word to the DPR. A safer way to determine completion of a flash operation is to use the *operation complete* bit (bit 3) – see below.

5.33.2 FLASH_STATUS bit 1: illegal write

The *illegal write* bit is set to '1' when an attempt is made to write to a protected sector in flash. The bit is reset to '0' following a successful write.

5.33.3 FLASH_STATUS bit 2: illegal erase

The *illegal erase* bit is set to '1' when an attempt is made to erase a protected sector in flash. The bit is reset to '0' following a successful erase.

5.33.4 FLASH_STATUS bit 3: operation complete

The *operation complete* bit is set to '1' when an ASMI operation has completed. The bit is reset when the flash status word has been read (e.g. via the FlexComms comms service channel).

5.34 0x0946 FLASH_READ_BYTE

Bits 7 to 0 of the FLASH_READ_BYTE object contain the byte read from flash in a flash read operation.

Table 20: Flash read byte object

bit 15	…	bit 8	bit 7	…	bit 0
'0'			byte read from flash		

5.35 0x0949 ASMI_STATUS

ASMI_STATUS provides access to the 8-bit status register in the FPGA's flash configuration memory. See ref [10] for details of the status register contents.

5.36 0x094C FLASH_PROTECT

FLASH_PROTECT allows writes to the block protect bits of the flash status register (Table 21).

Table 21: FLASH_PROTECT bits

bit 15	…	bit 3	bit 2	bit 1	bit 0
not used			BP2	BP1	BP0

The block protect bits allow part of the flash to be configured as read-only, as shown in Table 22.

Table 22: flash block protect bits

BP2	BP1	BP0	protected area
0	0	0	none
0	0	1	upper eighth (sector 7)
0	1	0	upper quarter (sectors 6 and 7)
0	1	1	upper half (sectors 4, 5, 6 and 7)
1	0	0	all
1	0	1	all
1	1	0	all
1	1	1	all

5.37 0x094D FIRMWARE_UPDATE_CONTROL

FIRMWARE_UPDATE_CONTROL contains bits used in FPGA firmware update functions. Refer to [2], section 9 for details of the flash interface and remote firmware updates.

Table 23: FIRMWARE_UPDATE_CONTROL bits

bit	description
0	enable firmware update
1	reconfigure now
2	disable watchdog stroke
3..15	reserved ('0')

5.37.1 FIRMWARE_UPDATE_CONTROL bit 0: enable firmware update

Set the *enable firmware update* bit to '1' to allow flash writes and erases. Flash write and erase instructions have no effect when the enable firmware update bit is '0'.

5.37.2 FIRMWARE_UPDATE_CONTROL bit 1: reconfigure now

Writing '1' to the *reconfigure now* bit causes the FPGA to reconfigure itself. If a valid user image is present in flash at address 0x80000, the FPGA will be configured from the user image; if not, the FPGA will be configured from the factory default image at flash address 0x00000.

5.37.3 FIRMWARE_UPDATE_CONTROL bit 2: disable watchdog stroke

Setting the *disable watchdog stroke* bit to '1' prevents the FPGA logic from automatically toggling the remote update control reset input, so causes the FPGA to reconfigure after 6.5ms.

5.38 0x094E RU_USER_DATA_HI and 0x094F RU_USER_DATA_LO

The 29-bit RU_USER_DATA is valid when the *remote update busy* bit has returned to '0' following a remote update parameter read (initiated by writing '1' to flash control word bit 6).

Table 24: RU_USER_DATA words

RU_USER_DATA_HI						RU_USER_DATA_LO						
bit 15	bit 14	bit 13	bit 12	…	bit 0	bit 15	bit 14	….	….	…..	bit 1	bit 0
0	0	0	remote update data									

5.39 0x0950 RU_USER_PARAM_ETC

Bits 0 to 4 of RU_USER_PARAM_ETC are used to interface to the Altera remote update Megafunction (ALTREMOTE_UPDATE). Refer to section [11] for more details.

Table 25: RU_USER_PARAM_ETC bits

bit 15..5	bit 4	bit 3	bit 2	bit 1	bit 0
not used	read source		param		

5.40 0x0951 FLASH_APP_FIRMWARE_ADDRESS_HI and 0x0952 FLASH_APP_FIRMWARE_ADDRESS_LO

FLASH_APP_FIRMWARE_ADDRESS returns the address in flash of the FPGA application firmware image. This address is the start of the sector immediately above the last sector used to store the factory firmware image.

5.41 0x0953 PROBE_X, 0x0955 PROBE_Y, 0x0957 PROBE_Z

The FCA digitises voltages from analogue probes (e.g. RSP2) using three AD7687 16-bit ADCs. The digitized signals are read from the PROBE_X, Y and Z objects. The ADC conversions are triggered SYNC_ACQUISITION_DELAY clock cycles after the sync frame has been received.
The time taken to perform the ADC conversion is given by the 0x095E PROBE_SAMPLE_DELAY object.

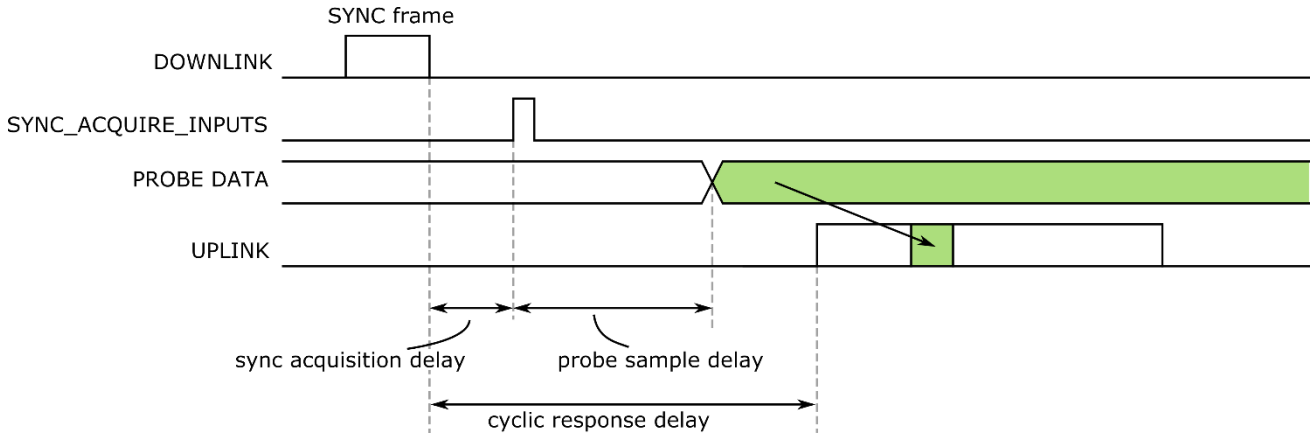


Figure 5: probe ADC conversion timing

If probe data from the most recent conversion is required in the process data response frame (uplink), the cyclic response delay must be set appropriately, as shown in Figure 5.
See ref [3] for details of the probe interface.

Note that in the FCA, only 15 bits of the ADC is used as the remaining bit is used to give increase range (see section 5.19). This means that the least significant bit is always read as zero.

The relationship between differential probe voltages V+ and V- and ADC count is given in Table 26. This is copied directly from the REVO-2 processing board datasheet and does not account for the fact that only 15 bits of the converter are used.

Table 26: relationship between probe voltage and ADC count

V+ (V)	V-(V)	(V+)-(V-) (V)	ADC_IN+(V)	ADC_IN-(V)	ADC count
2.707	0.093	2.614	4.096	0	0x7FFF
1.4000199	1.3999801	0.000039886	2.04800494	2.0479506	0x0001
1.400	1.400	0	2.048	2.048	0x0000
0.093064	2.706936	-2.613872	0.00003125	4.0959687	0x8001
0.093	2.707	-2.614	0	4.096	0x8000

5.42 0x0958 FIRST_FREE_FLASH_ADDRESS_HI and 0x0959 FIRST_FREE_FLASH_ADDRESS_LO

FIRST_FREE_FLASH_ADDRESS returns the address of the start of free space in flash. Addresses below FIRST_FREE_FLASH_ADDRESS are used for firmware storage.

5.43 0x095A FLASH_NUMBER_OF_SECTORS

FLASH_NUMBER_OF_SECTORS returns the number of sectors in the configuration flash.

5.44 0x095C FLASH_SECTOR_SIZE_HI and 0x095D FLASH_SECTOR_SIZE_LO

FLASH_SECTOR_SIZE returns the number of bytes in each flash sector.

5.45 0x095E PROBE_SAMPLE_DELAY

PROBE_SAMPLE_DELAY gives the number of clock cycles between the acquisition time and valid probe data being available, as shown in Figure 5.

5.46 0x095F NODE TYPE TO EEPROM

At power on this contains the node type currently written in EEPROM. Writing a new value to this register and setting the appropriate bit in 0x092E EEPROM_CONTROL writes a new node type to EEPROM.

5.47 0x0960 ADDRESS TO EEPROM

Contains the address in EEPROM to either read from or write to depending on which command is issued (see 0x092E EEPROM_CONTROL).

5.48 0x0961 BYTE TO EEPROM

This location contains the byte that will be written to EEPROM when the appropriate bit in 0x092E EEPROM_CONTROL is set.

5.49 0x0962 BYTE FROM EEPROM

Following a read EEPROM command, this location contains the data byte from the EEPROM.

5.50 0x098C - 0x0991 PROBE_EEPROM_BYTES

When the Processing Board FPGA detects that a probe has been attached the FPGA attempts to read the first 13 bytes from the probe's EEPROM. The 13th byte is interpreted as a checksum of the previous 12 bytes, in the form of a bitwise modulo-2 sum (exclusive-OR) of bytes 0-11.

If the checksum fails, the FPGA stores the 13 bytes read on the first attempt and tries to read the 13 bytes from the probe again.

If the checksum passes, or all of the new 13 bytes are identical to the previous 13 bytes, the bytes are stored in the PROBE_EEPROM_BYTES objects; a checksum fail or a mismatch causes the process to be repeated. A maximum of sixteen attempts are made to read the data from the EEPROM.

If after sixteen attempts the checksum has failed and on no two successive reads were the bytes identical, then the *probe EEPROM error* bit (status word bit 5) is set to '1'.

The least significant byte in each PROBE_EEPROM_BYTES object contains the data from odd addresses in the EEPROM, the most significant byte contains data from even addresses.

Table 27: Probe EEPROM bytes

bit 15	…	bit 8	bit 7	...	bit 0
byte from EEPROM address n			byte from EEPROM address n+1		

See ref [12] for details of the data contained in the probe EEPROM bytes.

5.51 0x0992 PROBE EEPROM CONTROL

Controls read and write access to the probe's EEPROM. The individual functions are shown in Table 28: Probe EEPROM control Table 28. All the bits are self clearing.

Table 28: Probe EEPROM control

bit	description
0	write byte
1	read byte
2-7	spare

5.51.1 PROBE EEPROM CONTROL, Bit 0: write byte

Initiates a write command to the probe EEPROM

5.51.2 PROBE EEPROM CONTROL Bit 1: read byte

Initiates a read command to the probe EEPROM

5.52 0x0994 PROBE EEPROM ADDRESS

Specifies the probe EEPROM address to either read data from or write data to depending on the command. Probe EEPROM memory maps can be found in the relevant product's datasheet.

5.53 0x0995 PROBE EEPROM WRITE DATA

The bottom eight bits in this location contain the data byte to write to the probe EEPROM.

5.54 0x0996 PROBE EEPROM READ DATA

The bottom eight bits contain the data byte that has been read from the probe EEPROM following a read command.

5.55 0x0997 VPROBE_RAW_CURRENT

Vprobe current measurement directly from the external power monitoring device. Resolution is 0.125 mA/bit and the range is 0 to 512.0 mA.

5.56 0x0998 PLUS_5VL_RAW_CURRENT

+5VL current measurement directly from the external power monitoring device. Resolution is 0.125 mA/bit and the range is 0 to 512.0 mA.

5.57 0x0999 NEG_5V_RAW_CURRENT

-5V rail current measurement directly from the external monitoring device. Resolution is 0.076 mA/bit and the range is 0 to 311.3 mA.

6 Current Measurement Calibration

The FCA contains three power monitoring circuits which measure the voltage and current of the three probe power supplies (VPROBE, +5VL and -5V). Voltage readings are fed directly into the DPR, however the current measurements require correction factors to be applied to increase accuracy and convert the measurement into an easily readable resolution. Higher accuracy readings are required when setting up certain types of analogue probe such as RSP2.

The resolution of the 12-bit sensor is 0.025 mV/bit, so the range and resolution of the current readings are defined by the sense resistors used in the circuit. These have been chosen to maximize the resolution without restricting the maximum reading. The sense resistors chosen along with the resolution and range are shown in Table 29.

Supply	Sense resistor (Ω)	Resolution (mA/bit)	Range (mA)
VPROBE	0.2	0.125	512.0
+5VL	0.2	0.125	512.0
-5V	0.33	0.076	311.3

Table 29: Current measurement range and resolution

The accuracy of the measurements is based on the properties of the measurement sensor (LTC2945) and the current sense resistor. A summary of the different kinds of error is shown in Table 30. These are taken from the LTC2945 and resistor's datasheets [13] and [14].

Error Type	Magnitude
Offset Error	3.1 bits or 77.5 µV
Nonlinearity	3 bits or 75 µV
Gain Error	0.6 %
Input current	20 µA
Sense resistor tolerence	1.0 %
Total error (µA)	$\frac{152.5}{R_{sense}} + 20 + 1.6\%$

Table 30: Sources of current measurement error. Note that percentages are the proportion of the value being measured.

Since all of these errors, apart from non-linearity, are either constant or linear they can be calibrated out using simple linear and constant correction factors. This leaves a residual error (in µA) of $75/R_{sense}$. The raw and calibrated accuracy of the chosen sense resistors for each of the supplies is shown in Table 31.

Supply	Non-calibrated accuracy	Calibrated accuracy
VPROBE	783 µA + 1.6%	375 µA
+5VL	783 µA + 1.6%	375 µA
-5V	482 + 1.6%	227 µA

Table 31: Theoretical accuracy of the current measurements.

6.1 Calibrating the current sensor

The raw measurements from the current sense chip are corrected inside the FPGA with gain and offset constants that are stored in the external EEPROM. This correction serves two purposes: it corrects the measurement to remove linear and constant errors, and it converts the value into a human readable output with a resolution of 10 µA per bit. This is done according to the following formula:

$$Output = Input \times \frac{a}{256} + b - 128$$

Where a is the gain and b the offset. The factors of 256 and 128 are included to correct the ranges of the constants.

To calibrate the measurements, complete the following steps:

1) Program the nominal values of the coefficients into the EEPROM as shown in Table 32.

Register	DPR Address	Value Decimal	Value Hex
VPROBE_CURRENT_GAIN	93E	3200	C80
VPROBE_CURRENT_OFFSET	93F	128	80
PLUS_5VL_GAIN	940	3200	C80
PLUS_5VL_OFFSET	941	128	80
NEG_5V_GAIN	942	1946	79A
NEG_5V_OFFSET	943	128	80

Table 32: Default current measurement constants

For each of the supplies, complete the following:

2) With the supply enabled, load the supply with a number of different currents and record measurements from the FCA and a calibrated multimeter.

3) Plot the FCA measured values against multi-meter values and calculate the line of best fit using linear regression ($y = mx + c$).

4) Adjust the coefficients as follows:

$$NewGain = \frac{OldGain}{m} \qquad NewOffset = OldOffset - \frac{100 \times c}{m}$$

5) Check that the new constants work correctly by repeating step 3.

6) Update the calibration date week and year bytes in EEPROM