

PCM 2.0

(Based on PHOS FEE Board Controller 0.1)

Version 2.0

Qingxia Li, Feifei Zou

Department of Electronics and Information



華中科技大學
Huazhong University of
Science and Technology

August 2006, Wuhan, China



Preface

The purpose of this document is to give information related to the board controller of FEE in PHOS.

Revision Record

Version 2.0: This document version corresponds to BC version hex 0020 (firmware name of BC is PCM2_0.sof)

Third version, August 2006, Qingxia Li, Feifei Zou

Second version, April 2006, Qingxia Li, Feifei Zou.

First version, August 2005, Qingxia Li, Feifei Zou.



Contents

1	LOCATION OF PCM IN PHOS ELECTRONICS	3
2	FUNCTIONAL BLOCKS OF PCM.....	4
3	FUNCTION DESCRIPTION FOR PCM.....	5
3.1	INFORMATION STORAGE.....	5
3.1.1	Version Number of PCM	5
3.1.2	FEE Card Serial Number Record and Check.....	6
3.1.3	FEE Card address read and record	6
3.2	CONTROL AND MONITORING	6
3.2.1	General control and monitoring	6
3.2.2	Clock control.....	7
3.2.3	Temperature/power control and monitoring	7
3.3	COMMUNICATION	9
3.3.1	Communication with USB Chip	9
3.3.2	Communication with Altro Chip	9
3.3.3	Communication with DCS through RCU	9
3.4	CONFIGURATION OF FPGA.....	10
4	REGISTER TABLE OF PCM	10
5	SCRIPTS ON DCS FOR COMMUNICATION WITH PCM.....	20
5.1	READ BC VERSION.....	20
5.2	SET HIGH VOLTAGE FOR APD BIAS	21
5.2.1	Register and CSP Numbers Correspondence.....	21
5.2.2	Set 1 Channel.....	22
5.2.3	Set 2 channels	22
5.2.4	Set 3 channels	23
5.2.5	Set 32 Channels.....	23
5.3	READ TEMPERATURE AND VOLTAGES FROM AD7417	26
APPENDIX.....	29
APPENDIX A	ABBREVIATIONS AND ACRONYMS	29
APPENDIX B	REFERENCES.....	29

1 Location of PCM in PHOS Electronics

The PHOS Control and Monitoring (PCM) is located in FEE card. It communicates with DCS through RCU. The scripts running in DCS send commands and data to PCM and read back data from PCM.

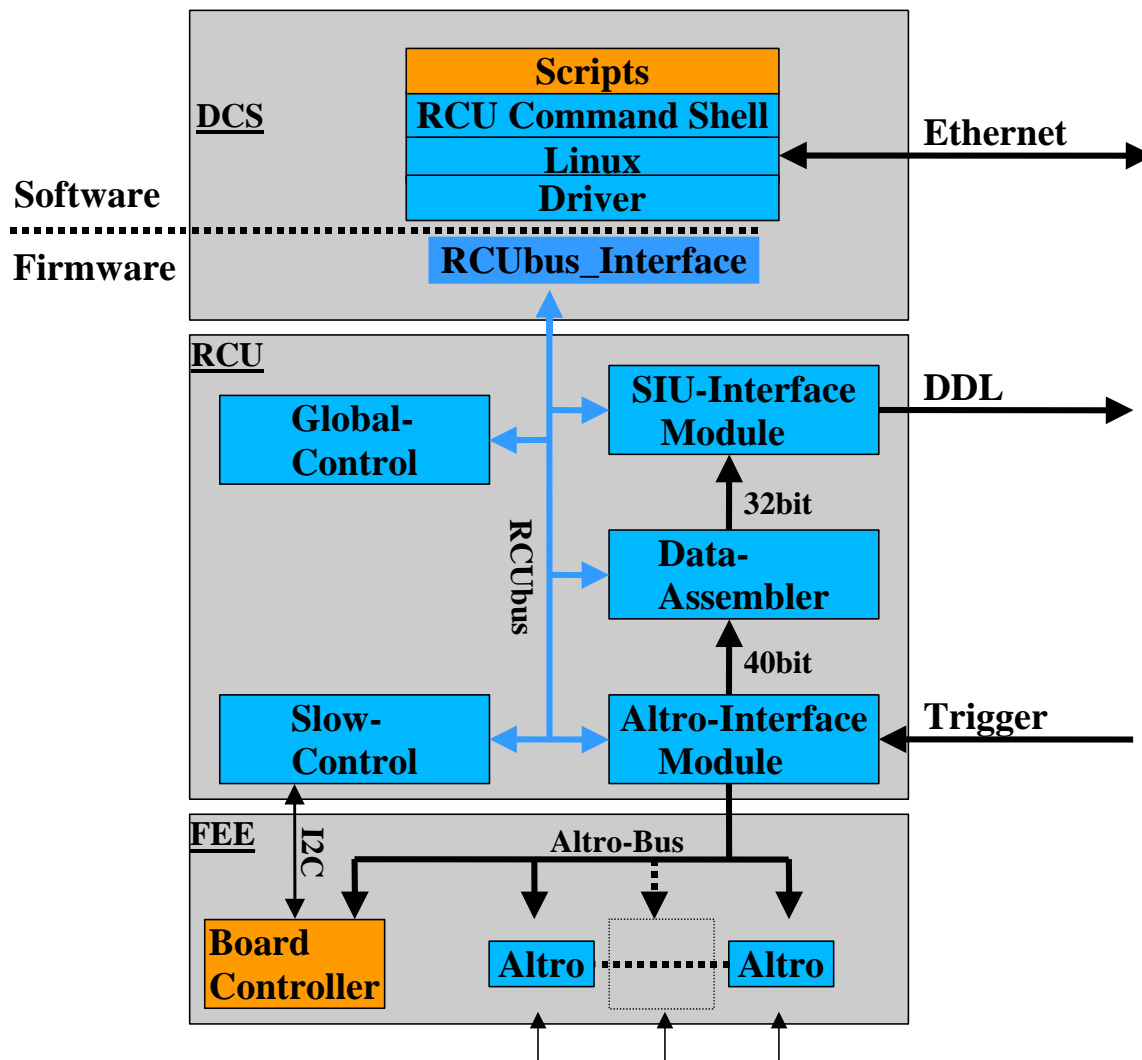


Figure 1 Location of PCM in Front-End Electronics of PHOS

The orange-colored block inside the bottom block FEE is the PHOS Control and Monitoring (PCM). The orange-colored block inside top block DCS includes scripts running in DCS for communication with PCM.

2 Functional Blocks of PCM

The PCM includes the following blocks:

- Register Block and decoder
- Interface to RCU (Altro bus and slow control bus)
- Interface to chip of temperature sensor and voltage monitor (I^2C bus)
- Interface to high voltage bias circuit (SPI bus)
- Interface to voltage regulators for voltage control and monitoring
- Interface to USB chip (I^2C bus and FIFO)

Interface to USB chip is not implemented in this PCM version. Other blocks are all implemented in this version.

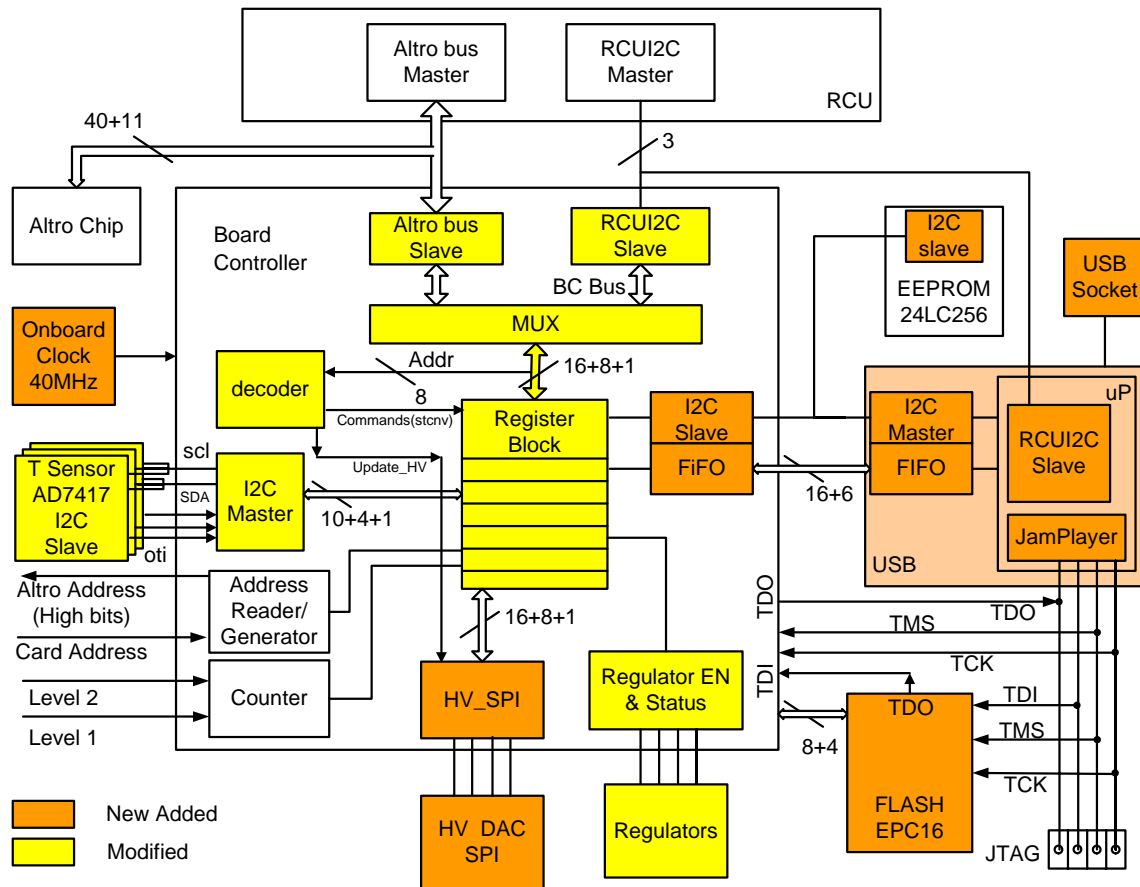


Figure 2 Functional Block Diagram of PCM

This version of PCM is based on TPC BC. PCM modified blocks in TPC BC and Added some blocks. PHOS PCM increase internal address bus from 5 to 8 lines, and increase registers from 24 to 84 registers. In figure 2, The orange-colored blocks are new added, and yellow blocks are modified.

HV_SPI block is silent when there is no “StartHV” command from register block. No clock is

running inside the block. When receiving “StartHV” command from register block, HV_SPI block enables the internal clock. Then it read data from HV_DATA register block (address is from hex40 to hex 5F) and store them in DAC_RAM. Third, it sends data in DAC_RAM bit by bit to 4 DACs in high voltage bias circuit, meanwhile reads back the data shifted back from 4 DACs and compares with the data sent. The comparison result is stored in registers HV_FB1 and HV_FB2, then transferred to register block through data bus. In the whole process, HV_SPI block give SPIBusy signal to register block. After the whole process is finished, HV_SPI block is silent again, waiting for next “StartHV” command from register block. The functional block diagram of HV_SPI is shown below.

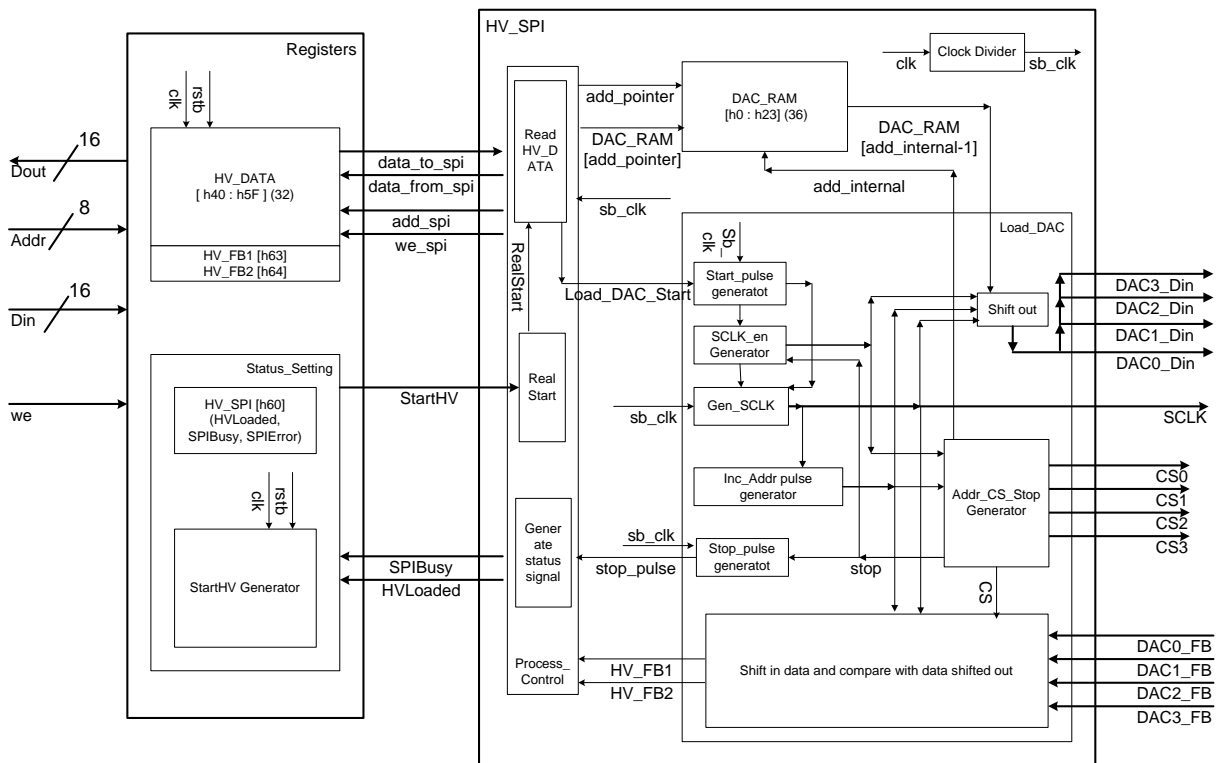


Figure 3 Functional Blocks of SPI Block for High Voltage Bias

3 Function description for PCM

3.1 Information Storage

3.1.1 Version Number of PCM

The PCM stores the version number of itself in register of address hex 20. The version number can be read out by running a script in DCS to read data in register 20 of PCM.



(The version number for this version is hex 20)

3.1.2 FEE Card Serial Number Record and Check

The FEE card serial number is permanently written in the EEPROM via an external programmer when programming the program into EEPROM for the USB chip. (The EEPROM chip can be unplugged, programmed and plugged back). There is a register (FEC_SN) in the PCM to store the FEE card serial number. Every time the FPGA is configured or reconfigured, the USB chip sends the serial number to the FPGA via the I²C bus. The PCM regularly checks the serial number. If the serial number is wrong, this means the USB block is not properly working and a status bit is set in a status register.

3.1.3 FEE Card address read and record

Every time the FPGA is configured or reconfigured, it reads the geographic address of the FEE card from the local GTL bus and stores in a register (csr2, address is hex 13).

3.2 Control and Monitoring

3.2.1 General control and monitoring

3.2.1.1 Reset locally and globally

The FPGA has a reset input which can be activated either via the RCU over a reset signal from GTL backplane, or manually via the reset button on the FEE card. The effect of Reset is that all PCM state machines are reset to default state. Note: the reload pushbutton reloads the PCM configuration into the FPGA.

3.2.1.2 PCM interrupt

The PCM generates an interrupt signal for the RCU via the PCM_INT signal level on backplane. There are three interrupt reasons:

- Flags of power supply occurs
- voltage or current values on the FEE card are out of the allowed range
- FEE card temperature exceeds a specified value.

3.2.1.3 Trigger Counting

The PCM counts the Altro strobe signals L1 and L2 from the GTL bus and stores both numbers in registers (level-1 trigger number in register L1CNT, and level2 trigger number in register L2CNT).

3.2.2 Clock control

3.2.2.1 SCLK control

The PCM can switch on and off the ADC sampling clock (SLCK) provided by the GTL bus by enabling the SCLK driver on the FEE card. (Control line is RDOCLK_EN).
Note: SCLK is also connected to the FPGA for monitoring purposes.

3.2.2.2 RCLK control

The PCM can switch off the readout clock (RCLK) provided by the GTL bus by switching off the RCLK driver on FEC card. (Control line is ADCCLK_EN) Note: RCLK is also monitored by FPGA.

3.2.2.3 Local clock control and monitoring

For debugging purposes, a local 40MHz backup oscillator is provided to the PCM. A separate logical block monitors the local clock and RCLK. In case RCLK fails, it enables the local clock to drive the entire logical blocks inside FPGA.

3.2.3 Temperature/power control and monitoring

3.2.3.1 Temperature reading and recording

When receiving a command, the PCM reads the temperature from three temperature sensors through the i2C interface of the PCM. One sensor is located in the voltage regulator region, the second sensor is located near the FPGA and the third sensor is located on the bottom layer of PCB. If one of the temperature sensors gets beyond a programmable temperature limit, it sends an interrupt to the PCM.

3.2.3.2 Monitoring voltage values

The PCM monitors the following voltage and current values and stores them in PCM registers with addresses and mnemonics given below:

- 4.2V digital input voltage and current (for 3.3V digital output) [Voltage Register: D4V2 at address 07, Current Register: D4V2C at address 08]
- 3.3V digital input voltage and current (for 2.5V digital output to Altro chip and FPGA) [Voltage Register: D3V3 at address 09, Current register D3V3C at address 0A]

- 4.0V analog input voltage and current (for 2.5V analog output to Altro chip) [Voltage Register: A4V0 at address 29, Current register A4V0C at address 2A]
- 13.5V analog input voltage and current (for 12V analog output to Shaper) [Voltage Register: A13V5 at address 2B, Current register A13V5C at address 2C]
- Input –6V voltage and current (for –5V output) [Voltage register: AM6V at address 39, Current register AM6VC at address 3A]
- 6V input voltage and current (for 5V output to analog circuits) [Voltage Register: A6V at address 3B, Current register A6VC at address 3C]

3.2.3.3 Monitoring regulator working status

The PCM monitors the working status of the LV regulators which power the following sections on the FEE card

- Shaper and Analog sums section: Check output of regulators (+5V output analog) by monitoring the logical level of ShaperPS_ERROR [Register: CSR1[5]/VTS[1]]
- Altro section: Check output of regulators (2.5V analog and 2.5V digital) by monitoring the logical level of AltroPS_ERROR [Register: CSR1[6]/VTS[3]]
- Digital logic sections: Check output of regulator (2.5V digital) for FPGA and GTL+ bus transceiver as well as regulator (3.3V digital) for temperature sensors, memories, USB chip etc, by monitoring the logical level of ALLPS_ERROR [Register: CSR1[6]/VTS[0]]
- HV bias control section: Check output of regulator (5V analog and digital combined) by monitoring logical level of BiasPS_ERROR [Register: CSR1[6]/VTS[2]]

Note: - 5V regulator output (CSP and gain-2 buffer) cannot be checked in this way.

3.2.3.4 Control voltage supply

The PCM enables the following LV voltage supply sections as function of power being enabled of the digital part. The digital power is enabled either via a dedicated GTL bus line driven by the RCU, or in case of RCU absence if a USB cable is connected.

- Shaper, CSP and analog sums: +12V and –5V power supply to CSP; +5V and –5V power supply to shaper and Analog summing blocks (+5V only) (by control line: ShaperPS_EN)
- Altro: +2.5V analog and +2.5V digital power supply to Altro chips (by control line: AltroPS_EN)
- HV-Bias: +5V analog and digital combined power supply to HV bias control

block. (by control line: BiasPS_EN)

3.2.3.5 High Voltage Bias (+400V) control

FEE card provides 200-400V voltage power supply to the APD of each channel. The Bias voltage values are set via 32 registers for HV bias control in the PCM. Each register contains 10 bit data corresponding to a bias voltage of one APD diode. [Register block: HV_DATA, address from hex 40 to hex 5F]

3.3 Communication

3.3.1 Communication with USB Chip

The PCM communicates with the USB chip via the I2C bus and a FIFO. The I2C link is for command transfers and the FIFO serves for data transfers.

The USB chip is monitored by a watch dog. It can monitor the FPGA working status by regularly checking the following registers:

- Register [FEC_SN] for FEE card serial number
- FPGA status register [FPGA_Status]
- Reconfiguration enable register [recfg_EN].

Whenever the FPGA is not working, the USB processor initiates reconfiguration of the PCM FPGA.

3.3.2 Communication with Altro Chip

The PCM is connected to the Altro chip via the Altro bus (the PCM can use the Altro bus only when the RCU allow to do so) and to test controller lines. The test controller line includes

- TMS_N for test mode selection (one pin is needed for each Altro chip, 4 pins are needed). On the revised FEE card V1 there is a switch for test mode.
- ADC_ADD0 for ADC select in test mode.
- ADC_ADD1 for ADC select in test mode.

By default, the PCM sets all Altro chips in normal mode (not test mode)

3.3.3 Communication with DCS through RCU

The PCM can receive commands and data from DCS and feed these back data to DCS through RCU.

The communication protocol between the PCM slave and the RCU master is the

RCU-I2C, a variant of I2C, and the Altro bus protocol. The RCU of the ALICE TPC project can be used for PHOS since the communication and bus protocols are identical for PHOS-FEE and TPC-FEC electronics.

3.4 Configuration of FPGA

FPGA can be configured by three ways.

1. JTAG off the shelf
2. Configuration device (EPC16)
3. RCU configuration on shelf

The way to reconfigure the Altera ACEX FPGA is through the JTAG bus. The USB chip receives configuration data from the RCU via RCU I2C bus, and then it writes the data to the 16 Mbit EPC16 configurations Prom via the JTAG interface. The EPC16 shares most JTAG pins with the FPGA. After the data transfer is completed, the board controller sets a register [Register: recfg_EN] to indicate the end of the data transfer. The USB chip monitors the recfg_EN register. Once the USB chip finds the register is set, it enables the FPGA configuration process. The FPGA is then reconfigured via its configuration pins.

4 Register table of PCM (The postils are some notes of test results)

Table 1 Register Table for PCM

Reg. Addr.	Mnemonic	Reg. Name	Width	Acc. Mode	Allow Bcast	Meaning
00						
01	T1_TH	Temperature Thr.	10	R/W	Y	Maximum Temperature Threshold
02	D4V2_TH	D4V threshold	10	R/W	Y	Minimum 4V Digital Voltage Threshold
03	D4V2C_TH	D4VC threshold	10	R/W	Y	Maximum 4V Digital Current Threshold
04	D3V3_TH	D3V3 threshold	10	R/W	Y	Minimum 3.3V Digital Voltage Threshold
05	D3V3C_TH	D3V3C threshold	10	R/W	Y	Maximum 3.3V Digital Current Threshold
06	TEMP1	Temperature	10	R	N/A	Temperature Value
07	D4V2	4.2V Digital Voltage	10	R	N/A	4V Digital Voltage Value
08	D4V2C	4.2V Digital Current	10	R	N/A	4V Digital Current Value
09	D3V3	3.3V Digital Voltage	10	R	N/A	3.3V Digital Voltage Value



PHOS Control and Monitoring

0A	D3V3C	3.3V Digital Current	10	R	N/A	3.3V Digital Current Value
0B	L1CNT	L1 Counter	16	R	N/A	Number of L1 Trigger Received
0C	L2CNT	L2 Counter	16	R	N/A	Number of L2 Trigger Received
0D	SCLKCNT	Sampling clk counter	16	R	N/A	Sampling Clock counter
0E	DSTBCNT	Data Strobe Counter	8	R	N/A	Number of Data Strobe in the last ReadOut
11	CSR0	Configuration Status 0	11	R/W	Y	Interrupt - Mask Register
12	CSR1	Configuration Status 1	14	R/W	N/A	Error Status Register
13	CSR2	Configuration Status 2	16	R/W	Y	Card Configuration Status Register
14	CSR3	Configuration Status 3	16	R/W	Y	Card Configuration Status Register

<i>Reg. Addr.</i>	<i>Mnemonic</i>	<i>Reg. Name</i>	<i>Width</i>	<i>Acc. Mode</i>	<i>Allow Bcast</i>	<i>Meaning</i>
16	CNTLAT	Counters Latch	-	W	Y	Latch L1, L2, SCLK counters
17	CNTCLR	Counters Clear	-	W	Y	Clear L1, L2, SCLK counters
18	CSR1CLR	Conf. St. Reg 1 Clear	-	W	Y	Clear Error Status Register
19	ALRST	ALTRO Reset	-	W	Y	Reset all the ALTROs
1A	BCRST	BC Reset	-	W	Y	Set default values in registers of BC
1B	STCNV	Start Conversion mADC	-	W	Y	Start Conversion / Read Out Monitor ADC
1E	Update_HV	Update_HV	-	W	Y	Start to update high voltage
1F						

<i>Reg. Addr.</i>	<i>Mnemonic</i>	<i>Reg. Name</i>	<i>Width</i>	<i>Acc. Mode</i>	<i>Allow Bcast</i>	<i>Meaning</i>
20	BC_version	BC version				Version of Board Controller
21	T2_TH	Temperature Thr.	10	R/W	Y	Maximum Temperature Threshold
22	AM6V_TH	AM6V threshold	10	R/W	Y	Minimum -6V Analog Voltage Threshold
23	AM6VC_TH	AM6VC threshold	10	R/W	Y	Maximum -6V Analog Current Threshold
24	A13V5_TH	A13V threshold	10	R/W	Y	Minimum 13V Analog Voltage



PHOS Control and Monitoring

						Threshold
25	A13V5C_TH	A13VC threshold	10	R/W	Y	Maximum 13V Analog Current Threshold
26						
27						
28	TEMP2	Temperature	10	R	N/A	Temperature Value
29	A4V0	4.0V Analog Voltage	10	R	N/A	4.0V Analog Voltage Value
2A	A4V0C	4.0V Analog Current	10	R	N/A	4.0V Analog Current Value
2B	A13V5	13.5V Analog Voltage	10	R	N/A	13V Analog Voltage Value
2C	A13V5C	13.5V Analog Current	10	R	N/A	13V Analog Current Value
2D						
2E	ADC_ID	ADC ID number	2	R	N/A	Temperature/Voltage sensor ID number
2F	TestReg	Test register	16	R	N/A	For test

30						
31	T3_TH	Temperature Thr.	10	R/W	Y	Maximum Temperature Threshold
32	A4V0_TH	A4V0threshold	10	R/W	Y	Minimum 4.0V Analog Voltage Threshold
33	A4V0C_TH	A4V0C threshold	10	R/W	Y	Maximum 4.0V Analog Current Threshold
34	A6V_TH	A6V threshold	10	R/W	Y	Minimum 6V Analog Voltage Threshold
35	A6VC_TH	A6VC threshold	10	R/W	Y	Maximum 6V Analog Current Threshold
36						
37						
38	TEMP3	Temperature	10	R	N/A	Temperature Value
39	AM6V	-6V Analog Voltage	10	R	N/A	-6V Analog Voltage Value
3A	AM6VC	-6V Analog Current	10	R	N/A	-6V Analog Current Value
3B	A6V	6V Analog Voltage	10	R	N/A	6V Analog Voltage Value
3C	A6VC	6V Analog Current	10	R	N/A	6V Analog Current Value
3D						
3E						
3F						

40	HV_DATA[32]	High Voltage Data	16	R/W	Y	High Voltage Values for writing
----	-------------	-------------------	----	-----	---	---------------------------------



PHOS Control and Monitoring

~ 5F		[0-31]				to HV DAC
60	HV_SPI	HV SPI/DAC Status	16	R	N/A	Status of SPI Block for High Voltage Bias
62						
63	HV_FB1	HV Feedback 1 Status	16	R	N/A	HV DACs Feedback status for channels 1-16
64	HV_FB2	HV Feedback 2 Status	16	R	N/A	HV DACs Feedback status for channels 17-32
65	VTS	Voltage Temperature Status	16	R	Y	Voltage and Temperature Status Register
68						
69						
6A						
70	Serial Number_1	Card_number	16	R/W		The serial number of every FEE card written by usb
71	Serial Number_2	serial_number	16	R		The serial number of each FEE card fixed in firmware

FF						Reserved for USB
----	--	--	--	--	--	------------------

T1_TH (01) Default value = 40 °C = 00 1010 0000 = hex A0

9	0
X	Low Temperature Threshold

D4V2_TH (02) Default value = 3.8 V = 01 1101 1000 = hex 1D8 = dec 472

9	0
X	Low Analog Voltage Threshold

D4V2C_TH (03) Default value = 0.36 A = 00 0000 1100 = hex C = dec 12

9	0
X	High Analog Current Threshold

D3V3_TH (04) Default value = 2.9 V = 01 1100 0010 = hex 1C2 = dec 450

	9	0
X	Low Digital Voltage Threshold	

D3V3C_TH (05) Default value = 0.73 A = 00 0001 0001 = hex 011 = dec 17

	9	0
X	High Digital Current Threshold	

TEMP1 (06)

Conversion Factor: 0.25°C / ADC count

Default value = 40 °C = 00 1010 0000 = hex A0

	9	0
X	Temperature Value	

D4V2 (07)

Conversion Factor: 8.04 mV / ADC count

Default value = 3.8 V = 01 1101 1000 = hex 1D8 = dec 472

	9	0
X	Analog Voltage Value	

D4V2C (08)

Conversion Factor: 29.8 mA / ADC count

Default value = 0.36 A = 00 0000 1100 = hex C = dec 12

	9	0
X	Analog Current Value	

D3V3 (09)

Conversion Factor: 6.44 mV / ADC count

Default value = 2.9 V = 01 1100 0010 = hex 1C2 = dec 450

	9	0
X	Digital Voltage Value	

D3V3C (0A)

Conversion Factor: **42.9 mA** / ADC count

Default value = 0.73 A = 00 0001 0001 = hex 11 = dec 17

	9	0
X	Digital Current Value	

L1CNT (0B)

15	0
Number of L1 Triggers Received	

L2CNT (0C)

15	0
Number of L2 Triggers Received	

SCLKCNT (0D)

15	0
Number of SCLK	

DSTBCNT (0E)

	7	0
X	Number of DSTB Asserted	

CSR0 (11) Default value = 7FF

10	9	8	7	0
cnv	Error Mask		Interrupt Mask	

- CSR0[10] = 1, ADC waits for the command for starting the conversion
- CSR0[10] = 0, ADC conversion continuously

CSR1 (12)

13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	---	---	---	---	---	---	---	---	---	---



bc_int	bc_error	SC instr error	ALTRO error	BC instr error	BC par error	missed sclk	LV-regulator error	LV regulator error	dc over th	dv under th	ac over th	av under th	temp over th
--------	----------	----------------	-------------	----------------	--------------	-------------	--------------------	--------------------	------------	-------------	------------	-------------	--------------

0: temp over th -----Temp-1, Temp-2, Temp-3

1: av under th ----- 4.0 V analog, +6.0 V analog, -6.0 V analog, +13.5 V analog

2: ac over th ----- 4.0 V analog, +6.0 V analog, -6.0 V analog, +13.5 V analog

3: dv under th ----- 4.2 V digital, 3.3 V digital

4: dc over th ----- 4.2 V digital, 3.3 V digital

5: LV regulator error ---- Shaper +6 V

6: LV-regulator error ---- Altro Supply, digital 4.2V+3.3V, Bias Supply

CSR2 (13) Default value = 000F

15	11	10	9	4	3	2	1	0
HADD		card self test	altro test mode			clock_en	paps_en	

paps_en

CSR2[0] -- altro_sw

CSR2[1] -- pasa_sw

clock_en

CSR2 [2] -- rdclk_en

CSR2 [3] -- adcclk_en

altro test mode

CSR2 [4] -- adc_add0

CSR2 [5] -- adc_add1

CSR2 [6] -- altro address

CSR2 [7] -- altro address

CSR2 [8] -- altro addresss

CSR2 [9] -- TSM ON/OFF

CSR3 (14) Default value = 2220

15	7	0
cnv end	X	rdclk / sclk warning ratio

TEMP2 (28)

Conversion Factor: 0.25°C / ADC count

Default value = 40 °C = 00 1010 0000 = hex A0

9

0



X	Temperature Value
----------	--------------------------

A4V0 (29)Conversion Factor: **6.4 mV** / ADC count

Default value = 3.6 V = 10 0010 1110 = hex 22E = dec 558

	9	0
X	Analog Voltage Value	

A4V0C (2A)Conversion Factor: **42.9 mA** / ADC count

Default value = 0.858 A = 00 0001 0100 = hex 14 = dec 20

	9	0
X	Analog Current Value	

A13V5 (2B)Conversion Factor: **26.8 mV** / ADC count

Default value = 13.1 V = 01 1110 1000 = hex 1E8 = dec 488

	9	0
X	Digital Voltage Value	

A13V5C (2C)Conversion Factor: **22.3 mA** / ADC count

Default value = 0.334 A = 00 0000 1111 = hex F = dec 15

	9	0
X	Digital Current Value	

TEMP3 (38)Conversion Factor: **0.25°C** / ADC count

Default value = 40 °C = 00 1010 0000 = hex A0

	9	0
X	Temperature Value	

AM6V0 (39)

Conversion Factor: **4.88 mV** / ADC count – 8.2V

Default value = - 6.4 V = 01 0111 0000 = hex 170 = dec 368

9	0
X	Analog Voltage Value

AM6V0C (3A)

Conversion Factor: **29.3 mA** / ADC count

Default value = 0.44 A = 00 0000 1111 = hex F = dec 15

9	0
X	Analog Current Value

A6V0 (3B)

Conversion Factor: **11.4 mV** / ADC count

Default value = 5.6 V = 01 1110 1000 = hex 1E8 = dec 488

9	0
X	Digital Voltage Value

A6V0C (3C)

Conversion Factor: **34.73 mA** / ADC count

Default value = 0.764 A = 00 0001 0110 = hex 16 = dec 22

9	0
X	Digital Current Value

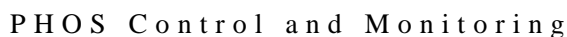
HV_DATA [0].. HV_DATA [31] (40-5F) Default value = hex 00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	X	X	DataBit9	DataBit8	DataBit7	DataBit6	DataBit5	DataBit4	DataBit3	DataBit2	DataBit1	DataBit0

DataBitx:one of 10-bit HV DATA; Bits 10-15 are not cared.

HV_SPI (60) Default value = hex 00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---



SPIError: Set by SPI block; 1, Something Wrong in SPI/DAC/HV; 0, no error.

SPIBusy: Set by SPI block; 1, SPI is being used; 0, SPI bus is free.

HVLoaded: Set by SPI block; 1, the action of writing data to DAC finished.

HV_FB1 (63) Default value = hex 00

Correct HVx: Set by SPI block of BC; 1, Compared with HV_DATAx and is correct; 0, not compared or not correct.

HV_FB2 (64) Default value = hex 00

Correct HVx: Set by SPI block of BC; 1, Compared with HV_DATAx and is correct; 0, not compared or not correct.

VTs (65)

ALLPS_ERROR: All powers have problems.

ShaperPS_ERROR: There are problems with shaper power supply.

BiasPS_ERROR: There are problems with power supply to HV bias part

Serial Number_1 (70) Default value = 0000

Detector: This PCM is for PHOS and the 14-15 bits are 01.



Version number: The production version of FEE and here 11-13 bits are 001.

Reserved: This bit is 0.

FEE card number: 0-9 bits are from b000000001 to b111111111.

Serial Number_2 (71) Default value = 0000

15	14	13	11	10	9	4	3	2	1	0
Detector(PHOS)		Version number	Reserved	FEE card number						

Detector: This PCM is for PHOS and the 14-15 bits are 01.

Version number: The production version of FEE and here 11-13 bits are 001.

Reserved: This bit is 0.

FEE card number: 0-9 bits are from b000000001 to b111111111.

5 Scripts on DCS for Communication with PCM

The PHOS board controller can receive commands and data from DCS, and feed back data to DCS. This is realized by running scripts in DCS. This section explains some example scripts running in DCS for communication between DCS and board controller. The example scripts given in this section assumes that FEE card is at position 3 (geographic address is 3).

5.1 Read BC Version

The version number is stored in the BC register of address h20. The script for reading BC version is as follows.

```

w 0x7000 0x523020 (read data from FEE board 3 on Branch A, the BC register 20)
w 0x7001 0x390000 (End of instruction)
wait 1 us
w 0x0 0x0 (Execute instruction)
wait 1 us
r 0x7800 (read error status register)
r 0x6000 (read result memory)

```

If there is no error (The values in 0x7800 are all zero), the value read from 0x6000 is h0020 for this BC version.

5.2 Set High Voltage for APD bias

5.2.1 Register and CSP Numbers Correspondence

The 32 values for setting 32 channel high voltage biases which correspond to CSP numbers are stored in registers of addresses from hex 40 to hex 5F. This correspondence is listed in the table below.

Table 2 Register address and CSP number correspondence

Register Address (hex number)	CSP Number (decimal number)
40	23
41	22
42	21
43	20
44	19
45	18
46	17
47	16
48	0
49	1
4A	2
4B	3
4C	4
4D	5
4E	6
4F	7
50	8
51	9
52	10
53	11
54	12
55	13
56	14
57	15
58	31
59	30
5A	29
5B	28
5C	27
5D	26
5E	25
5F	24

5.2.2 Set 1 Channel

```
w 0x7000 0x623041      (means: send BC register address 41)
w 0x7001 0x70029A      (means: send data 29A to register of adder 41)
w 0x7002 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7003 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7004 0x62301E      (sending 00 to address 1E: a command to update high voltage )
w 0x7005 0x700000
w 0x7006 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7007 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7008 0x523041      (read data from BC register of address 41)
w 0x7009 0x390000      (end of this script)
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000                (display data read from the register 41)
```

This script sets DAC output to hex 29A, which corresponds to 3.25V DAC analog output, and 260V high voltage output. If different value is needed, just change 29A to a value desired.

5.2.3 Set 2 channels

```
w 0x7000 0x623040      (means: send BC register address 40)
w 0x7001 0x70029A      (means: send data 29A to register of adder 40)
w 0x7002 0x623041      (means: send BC register address 41)
w 0x7003 0x70029A      (means: send data 29A to register of adder 41)
w 0x7004 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7005 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7006 0x62301E      (sending 00 to address 1E: a command to update high voltage )
w 0x7007 0x700000
w 0x7008 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7009 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x700a 0x523041      (read data from BC register of address 41)
w 0x700b 0x390000      (display data read from the register 41)
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000                (display data read from the register 41)
```

5.2.4 Set 3 channels

```

w 0x7000 0x623040      (means: send BC register address 40)
w 0x7001 0x70029A      (means: send data 29A to register of adder 40)
w 0x7002 0x623041      (means: send BC register address 41)
w 0x7003 0x70029A      (means: send data 29A to register of adder 41)
w 0x7004 0x623042      (means: send BC register address 42)
w 0x7005 0x70029A      (means: send data 29A to register of adder 42)
w 0x7006 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7007 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7008 0x62301E      (sending 00 to address 1E: a command to update high voltage )
w 0x7009 0x700000
w 0x700a 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x700b 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x700c 0x523041      (read data from BC register of address 41)
w 0x700d 0x390000      (display data read from the register 41)
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000                (display data read from the register 41)

```

5.2.5 Set 32 Channels

```

w 0x8000 0x4          (means: switch on the power of FEE at position 3)
wait 1 us              (means: wait for 1 us., This is not necessary. Just used as delimiter)
w 0x7000 0x623040
w 0x7001 0x70029A
w 0x7002 0x623041
w 0x7003 0x70029A

w 0x7004 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x7005 0x3AFFFF      # RCU wait    0xFFFF clocks

w 0x7006 0x623042
w 0x7007 0x70029A
w 0x7008 0x623043
w 0x7009 0x70029A

w 0x700a 0x3AFFFF      # RCU wait    0xFFFF clocks
w 0x700b 0x3AFFFF      # RCU wait    0xFFFF clocks

```



```

w 0x700c 0x623044
w 0x700d 0x70029A
w 0x700e 0x623045
w 0x700f 0x70029A

w 0x7010 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x7011 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7012 0x623046
w 0x7013 0x70029A
w 0x7014 0x623047
w 0x7015 0x70029A

w 0x7016 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x7017 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7018 0x623048
w 0x7019 0x70029A
w 0x701a 0x623049
w 0x701b 0x70029A

w 0x701c 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x701d 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x701f 0x62304A
w 0x7020 0x70029A
w 0x7021 0x62304B
w 0x7022 0x70029A

w 0x7023 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x7024 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7025 0x62304C
w 0x7026 0x70029A
w 0x7027 0x62304D
w 0x7028 0x70029A

w 0x7029 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x702a 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x702b 0x62304E
w 0x702c 0x70029A
w 0x702d 0x62304F

```



w 0x702e 0x70029A

w 0x702f 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7030 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7031 0x623050

w 0x7032 0x70029A

w 0x7033 0x623051

w 0x7034 0x70029A

w 0x7035 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7036 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7037 0x623052

w 0x7038 0x70029A

w 0x7039 0x623053

w 0x703a 0x70029A

w 0x703b 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x703c 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x703d 0x623054

w 0x703e 0x70029A

w 0x703f 0x623055

w 0x7040 0x70029A

w 0x7041 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7042 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7043 0x623056

w 0x7044 0x70029A

w 0x7045 0x623057

w 0x7046 0x70029A

w 0x7047 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7048 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7049 0x623058

w 0x704a 0x70029A

w 0x704b 0x623059

w 0x704c 0x70029A

w 0x704d 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x704e 0x3AFFFF # RCU wait 0xFFFF clocks

```

w 0x704f 0x62305A
w 0x7050 0x70029A
w 0x7051 0x62305B
w 0x7052 0x70029A

w 0x7053 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x7054 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x7055 0x62305E
w 0x7056 0x70029A
w 0x7057 0x62305F
w 0x7058 0x70029A

w 0x7059 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x705a 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x705b 0x62301E
w 0x705c 0x700000

w 0x705d 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x705e 0x3AFFFF # RCU wait 0xFFFF clocks

w 0x705f 0x52305F
w 0x7060 0x390000
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000

```

This script sets each channel to hex 29A, which corresponds to 3.25V DACs analog output, and 260V high voltage output. If a different value is needed for a channel, just change 29A to a value desired.

5.3 Read temperature and voltages from AD7417

```

w 0x7000 0x62301B      (sending 00 to address 1B: a command to AD7417 to read temperature and
                        voltages )
w 0x7001 0x700000

w 0x7002 0x3AFFFF # RCU wait 0xFFFF clocks
w 0x7003 0x3AFFFF # RCU wait 0xFFFF clocks

```

```

w 0x7004 0x523006      (read data from BC register of address 06)
w 0x7005 0x523007
w 0x7006 0x523008
w 0x7007 0x390000
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000  3            (display 3 data read from the register 06, 07, 08)
wait 1 us
w 0x7000 0x523009      (read data from BC register of address 09)
w 0x7001 0x52300A
w 0x7002 0x390000
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000  2            (display 2 data read from the register 09, 0A)
wait 1 us
w 0x7000 0x523028      (read data from BC register of address 28)
w 0x7001 0x523029
w 0x7002 0x52302A
w 0x7003 0x390000
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000  3            (display 3 data read from the register 28, 29, 2A)
wait 1 us
w 0x7000 0x52302B      (read data from BC register of address 2B)
w 0x7001 0x52302C
w 0x7002 0x390000
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000  2            (display 2 data read from the register 2B, 2C)
wait 1 us
w 0x7000 0x523038      (read data from BC register of address 38)
w 0x7001 0x523039
w 0x7002 0x52303A
w 0x7003 0x390000
wait 1 us
w 0x0 0x0

```

```
wait 1 us
r 0x7800
r 0x6000  3          (display 3 data read from the register 38, 39, 3A)
wait 1 us
w 0x7000 0x52303B    (read data from BC register of address 3B)
w 0x7001 0x52303C
w 0x7002 0x390000
wait 1 us
w 0x0 0x0
wait 1 us
r 0x7800
r 0x6000  2          (display 2 data read from the register 3B, 3C)
```

5.4 Read Serial Number

The serial number is stored in the BC register of address h70. The script for reading serial number is as follows.

```
w 0x7000 0x523070    (read data from FEE board 3 on Branch A, the BC register 70)
w 0x7001 0x390000    (End of instruction)
wait 1 us
w 0x0 0x0 (Execute instruction)
wait 1 us
r 0x7800 (read error status register)
r 0x6000 (read result memory)
```

If there is no error (The values in 0x7800 are all zero), the value read from 0x6000 is like 0x4801 for the serial number. And the detail definition for the bits sees the register table for PCM.

Appendix

Appendix A Abbreviations and Acronyms

PHOS	PHOton Spectrometer
BC	Board Controller
PCM	PHOS Control and Monitoring

Appendix B References

- [1] TPC Slow Control
- [2] TPC board controller register table
- [3] DCS&RCU – How topdf
- [4] RCU Firmware Manual. pdf