FTM Firmware Specifications

Patrick Vogler[1], Quirin Weitzel

v3.0 - January 2011

[1]Contact for questions and suggestions concerning this document: patrick.vogler@phys.ethz.ch

Contents

Chapter 1

Introduction

The FTM (FACT Trigger Master) board collects the trigger primitives from all 40 FTU boards (FACT Trigger Unit) and generates the trigger signal for the FACT camera. The trigger logic is a 'n-out-of-40' majority coincidence of all trigger primitives. Beside the trigger, the FTM board also generates a trigger-ID (see chapter 2). It is controlled by the main control software via ethernet. Two auxiliary RS-485 interfaces are also available.

In addition to the trigger, the FTM board also generates the other fast control signals: Time-Marker (TIM), DRS [1] reference clock (CLD) and reset. These four fast control signals are distributed to the FAD (FACT Analog to Digital) boards via the two FFC (FACT Fast Control) boards. The FTM board also provides via the TIM line the signal for the DRS timing calibration. In order to generate the CLD DRS reference clock, as well as the time-marker signal for DRS timing calibration, the FTM board uses a clock conditioner [2].

The FTM board has two counters, the 'timestamp counter' and the 'on-time counter'. While the 'timestamp counter' runs continously (counting up, resetted by e.g. a 'start run'), the 'on-time counter' only counts when the camera trigger is enabled.

The FTM board further serves as slow control master for the 40 FTU boards. The slow control of the FTU boards and the distribution of the trigger-ID to the FAD boards are performed via dedicated RS-485 buses. Because the FAD as well as the FTU boards are arranged in crates of 10 boards each, the FTM board has four connectors, one for each crate. Running over these connectors there are two RS-485 buses (one for FTU slow control and one for the trigger-ID) besides the busy signal from the FAD boards and the crate reset.

In addition, the FTM board controls the two FLPs (FACT Light Pulser) via four LVDS signals each. Light pulser 1 is located in the mirror dish, light pulser 2 inside the camera shutter. There are also digital auxiliary in- and outputs according to the NIM (Nuclear Instrumentation Module) standard, for example for external triggers and veto, and to have the signals accessible.

The main component of the FTM board is a FPGA (Xilinx Spartan XC3SD3400A-4FGG676C), fulfilling the main functions within the board. The purpose of this document is to provide specifications needed for the development of the firmware of this FPGA and the software (called 'main control' in the following) controlling the FTM board. For further information about the FTM board hardware please refer to [3].

Chapter 2

Trigger-ID

For each processed trigger the FTM board generates a unique trigger-ID to be broadcasted to all FAD boards and added to the event data. This trigger-ID consists of a 32 bit trigger number, a two byte trigger type indicator and a checksum. The transmission protocol for the trigger-ID broadcast is shown in table 2.1.

byte no	content
0	Trigger-No first byte (least significant byte)
1	Trigger-No second byte
2	Trigger-No third byte
3	Trigger-No forth byte (most significant byte)
4	Trigger-Type 1
5	Trigger-Type 2
6	CRC-8-CCITT (checksum)

Table 2.1: The transmission protocol to broadcast the trigger-ID to the FAD boards

A Cyclic Redundancy Check (CRC) over byte 0 - 5 is used to evaluate the integrity of the trigger-ID. An 8-CCITT CRC has been chosen which is based on the polynomial x^8+x^2+x+1 (00000111, omitting the most significant bit). The resulting 1-byte checksum comprises the last byte of the trigger-ID. The transmission of the trigger-ID to the FAD boards is done by means of dedicated RS-485 buses (one per crate).

In the first byte of the trigger type indicator (see table 2.2) n0 - n5 indicate the number of trigger primitives required for a trigger, thus the 'n' of the 'n-out-of-40' majority coincidence. The two flags 'external trigger 1' and 'external trigger 2', when set, indicate a trigger from the corresponding NIM inputs. See also section 4.1 and table 4.9 for further information.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
n5	n4	n3	n2	n1	n0	external trigger 2	external trigger 1

Table 2.2: Trigger-Type 1

The 'TIM source' bit in 'Trigger-Type 2' (see table 2.3) indicates the source of the timemarker signal: a '0' indicates the timemarker being produced in the FPGA while a '1' indicates the

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TIM source	LP_set_3	LP_set_2	LP_set_1	LP_set_0	pedestal	LP_2	LP_1

Table 2.3: Trigger-Type 2

timemarker coming from the clock conditioner. The flags 'LP_1' and 'LP_2' are set when the corresponding lightpulser has flashed while the 'pedestal' flag is set in case of a pedestal (random) trigger. An event having none of these flags set indicates a physics event. The bits 'LP_set_0' to 'LP_set_3' are used to code information about the light pulser settings. They only have a meaning in case of calibration events.

Chapter 3

FTM Commands

The communication between the FTM board and the main control, including the corresponding commands, protocols and data, is based on 16-bit words and big-endian. This is to facilitate the data-transmission over the Wiznet W5300 ethernet interface [4].

The basic structure of all commands is the same and given in table 3.1. After a start delimiter, the second word identifies the command. Next there is a parameter further refining the command, e.g. what to read. The fourth and fifth words are spares and should contain zeros. Starting from the sixth word, an optional data block of variable size is following. This data block differs in length and content depending on command and parameter. In case of 'read' instructions, the corresponding data block is sent back.

So far six different commands are foreseen: 'read', 'write', 'start run', 'stop run', 'ping FTUs' and 'crate reset' (see table 3.2). The command parameters of the 'read' command are shown in table 3.3. For the 'write' command there is no option because the static data block is the only data that can be written to the FTM board.

In table 3.4 the parameters to start a run are listed. The type of the run is fully described in the FTM configuration (static data block, see section 4.1), which always has to be sent by the main control before starting a run. Therefore the only option is to start an "endless" run or to take X events instead. In the latter case X is defined by a two words (32 bit) long unsigned integer, making up the command data block. The 'start run' command enables the transmission of trigger signals (physics, calibration or pedestal) to the FAD boards and resets the trigger and time counters. There is no parameter for stopping a run. If a number of events has been specified ('take X events'), the run will terminate if either the 'stop run' command is received or the requested number of events is reached.

In case of a 'ping FTUs' command the FTM will address the FTUs one by one and readout their DNA. The results are collected in the FTU list (see section 4.3), which is sent back to the main control. There are no parameters for this command. With the 'crate reset' command the FPGAs of a particular crate can be rebooted, where the command parameter defines the crate number (see table 3.5).

word no	content
0	start delimiter (e.g. '@')
1	command ID
2	command parameter
3	spare: containing 0x0000
4	spare: containing 0x0000
5	data block (optional and of variable size)
...	...
X	data block

Table 3.1: FTM command structure

command-ID: bits									
15 ... 8	7	6	5	4	3	2	1	0	command
0	0	0	0	0	0	0	0	1	read
0	0	0	0	0	0	0	1	0	write
0	0	0	0	0	0	1	0	0	start run
0	0	0	0	0	1	0	0	0	stop run
0	0	0	0	1	0	0	0	0	ping all FTUs
0	0	0	1	0	0	0	0	0	crate reset

Table 3.2: FTM command ID listing

command parameter: bits										
15 ... 8	7	6	5	4	3	2	1	0	command	data block
0	0	0	0	0	0	0	0	1	read static data block	no
0	0	0	0	0	0	0	1	0	read dynamic data block	no

Table 3.3: Command parameters for the 'read' command

command parameter: bits										
15 ... 8	7	6	5	4	3	2	1	0	command	data block
0	0	0	0	0	0	0	0	1	start run	no
0	0	0	0	0	0	0	1	0	take X events	number of events X

Table 3.4: Command parameters for the 'start run' command

command parameter: bits										
15 ... 8	7	6	5	4	3	2	1	0	command	data block
0	0	0	0	0	0	0	0	1	reset crate 0	no
0	0	0	0	0	0	0	1	0	reset crate 1	no
0	0	0	0	0	0	1	0	0	reset crate 2	no
0	0	0	0	0	1	0	0	0	reset crate 3	no

Table 3.5: Command parameters for the 'crate reset' command

Chapter 4

FTM data blocks

The trigger master features two main data blocks, named 'static data block' and 'dynamic data block' in the following. They are implemented in the firmware as block-RAM. In addition, there is the so-called 'FTU list', which is generated only on request ('ping FTUs' command). If any of these blocks is sent to the main control (either automatically or on demand), a header with a size of eleven words is added. This header is identical for all data blocks and contains solely read-only information: the FTM board ID (57-bit Xilinx device DNA [5, 6, 7, 8]), a firmware ID and the readings of the trigger counter and time stamp counter. The header structure is summarized in table 4.1.

word no	content
0x000	board ID bits 63...48
0x001	board ID bits 47...32
0x002	board ID bits 31...16
0x003	board ID bits 15...0
0x004	firmware ID
0x005	Trigger counter at read-out time bits 31...16
0x006	Trigger counter at read-out time bits 15...0
0x007	Time stamp counter at read-out time bits 47...32
0x008	Time stamp counter at read-out time bits 31...16
0x009	Time stamp counter at read-out time bits 15...0
0x00A	spare

Table 4.1: Header structure for sending a data block

4.1 Static data block

The static data block contains all the settings needed to configure and operate the FTM. It has to be written by the main control each time before a run is started or, in general, some component has to be reprogrammed. Single register access is not foreseen for the moment. In addition, whenever the FTM board receives a new static data block, it performs a complete reconfiguration including a reprogramming of the FTUs. Table 4.2 summarizes the static data block. More details about the individual registers can be found in the subsequent tables.

word no	content
0x000	general settings
0x001	on-board status LEDs
0x002	light pulser and pedestal trigger frequency
0x003	ratio between LP1, LP2 and pedestal triggers
0x004	light pulser 1 amplitude
0x005	light pulser 2 amplitude
0x006	light pulser 1 delay
0x007	light pulser 2 delay
0x008	majority coincidence n (for physics)
0x009	majority coincidence n (for calibration)
0x00A	trigger delay
0x00B	timemarker delay
0x00C	dead time
0x00D	clock conditioner R0 bits 31...16
0x00E	clock conditioner R0 bits 15...0
0x00F	clock conditioner R1 bits 31...16
0x010	clock conditioner R1 bits 15...0
0x011	clock conditioner R8 bits 31...16
0x012	clock conditioner R8 bits 15...0
0x013	clock conditioner R9 bits 31...16
0x014	clock conditioner R9 bits 15...0
0x015	clock conditioner R11 bits 31...16
0x016	clock conditioner R11 bits 15...0
0x017	clock conditioner R13 bits 31...16
0x018	clock conditioner R13 bits 15...0
0x019	clock conditioner R14 bits 31...16
0x01A	clock conditioner R14 bits 15...0
0x01B	clock conditioner R15 bits 31...16
0x01C	clock conditioner R15 bits 15...0
0x01D	spare
0x01E	spare
0x01F	spare
0x020	enables patch 0 board 0 crate 0
0x021	enables patch 1 board 0 crate 0
0x022	enables patch 2 board 0 crate 0
0x023	enables patch 3 board 0 crate 0
0x024	DAC_A board 0 crate 0
0x025	DAC_B board 0 crate 0
0x026	DAC_C board 0 crate 0
0x027	DAC_D board 0 crate 0
0x028	DAC_H board 0 crate 0
0x029	Prescaling board 0 crate 0
0x02A	enables patch 0 board 1 crate 0
0x02B	enables patch 1 board 1 crate 0

0x02C	enables patch 2 board 1 crate 0
0x02D	enables patch 3 board 1 crate 0
0x02E	DAC_A board 1 crate 0
0x02F	DAC_B board 1 crate 0
0x030	DAC_C board 1 crate 0
0x031	DAC_D board 1 crate 0
0x032	DAC_H board 1 crate 0
0x033	Prescaling board 1 crate 0
...	...
0x1A6	enables patch 0 board 9 crate 3
0x1A7	enables patch 1 board 9 crate 3
0x1A8	enables patch 2 board 9 crate 3
0x1A9	enables patch 3 board 9 crate 3
0x1AA	DAC_A board 9 crate 3
0x1AB	DAC_B board 9 crate 3
0x1AC	DAC_C board 9 crate 3
0x1AD	DAC_D board 9 crate 3
0x1AE	DAC_H board 9 crate 3
0x1AF	Prescaling board 9 crate 3
0x1B0	active FTU list crate 0
0x1B1	active FTU list crate 1
0x1B2	active FTU list crate 2
0x1B3	active FTU list crate 3

Table 4.2: Overview of the FTM static data block

The FTM general settings register is detailed in table 4.3. The 'TIM_CLK' bit defines whether the time marker is generated by the FPGA ('TIM_CLK' = 0, default for physics data taking), or whether it is generated by the clock conditioner ('TIM_CLK' = 1, e.g. for DRS timing calibration). The 'ext_veto', 'ext_trig_1' and 'ext_trig_2' bits enable (1) or disable (0) the NIM inputs for the external veto and trigger signals, respectively. In order to select which trigger sources are active during a run, the bits 'LP1', 'LP2', 'ped' and 'trigger' are foreseen (0 disabled, 1 enabled). During a physics run, for example, 'LP1', 'ped' and 'trigger' should all be set to generate interleaved calibration and pedestal events as well as activate the 'n-out-of-40' trigger input. For a didicated pedestal run only 'ped' should be set, since in this case the FTM sends directly a trigger to the FADs. For calibration runs it depends on whether the external (LP1) or internal (LP2) light pulser is used: For the first case 'LP1' and 'trigger' have to be set, since here the full trigger chain is involved and the camera triggers based on G-APD signals. For the second case only 'LP2' is needed, because the shutter is closed and the FTM sends directly the trigger signal to the FADs (like for pedestal events). Bits 8 to 15 of the general settings register are not used up to now.

Bit	15...8	7	6	5	4	3	2	1	0
Content	x	trigger	ped	LP2	LP1	ext_trig_2	ext_trig_1	ext_veto	TIM_CLK

Table 4.3: FTM general settings register

The 'on-board status LEDs' register shown in table 4.4 allows to switch a total of eight LEDs on the FTM board for debugging purposes by setting the corresponding bit high.

Bit	15...8	7	6	5	4	3	2	1	0
Content	x	red_3	red_2	gn_1	ye_1	red_1	gn_0	ye_0	red_0

Table 4.4: 'on-board status LEDs' register

The frequency, with which light pulser and pedestal triggers are sent, is stored in the register at address 0x002 (see table 4.5). It is given in Hz and adjustable up to about 1 kHz (10 bit). The next register defines the ratio of LP1, LP2 and pedestal events (see table 4.6).

Bit	15 - 10	9	8	...	2	1	0
Content	x	FREQ_9	FREQ_8	...	FREQ_2	FREQ_1	FREQ_0

Table 4.5: Register for the frequency of calibration and pedestal events

Bit	15 - 12	11	...	8	7	...	4	3	...	0
Content	x	ped_R3	...	ped_R0	LP2_R3	...	LP2_R0	LP1_R3	...	LP1_R0

Table 4.6: Register defining the ratio between pedestal, LP1 and LP2 events

In order to define the amplitude and characteristics of the light pulses that are generated by the LP1 and the LP2 system, the registers 'LP1 amplitude' and 'LP2 amplitude' are used, respectively. These registers are presented in table 4.7 and table 4.8. In general the light pulser systems are controlled from the FTM by means of four control lines: The first line defines the amplitude of the calibration events by sending a gate/pulse with an adjustable length (bits 0 to 3 in the amplitude registers). With the second and third line additional LEDs can be switched on in the calibration systems (bits 13 and 14). The fourth line is used to overdrive the LP systems and to generate a very fast timing pulse. To do so, bit 15 has to be set to 1.

Bit	15	14	13	12...4	3	...	0
Content	FCP1	add_LEDs1_1	add_LEDs1_0	x	LP1A_3	...	LP1A_0

Table 4.7: Light pulser 1 amplitude register

Bit	15	14	13	12...4	3	...	0
Content	FCP2	add_LEDs2_1	add_LEDs2_0	x	LP2A_3	...	LP2A_0

Table 4.8: Light pulser 2 amplitude register

The different settings of the 'n-out-of-40' logic (physics or calibration events) are stored in two separate registers, which both have a structure according to table 4.9.

Bit	15...6	5	4	3	2	1	0
Content	x	n5	n4	n3	n2	n1	n0

Table 4.9: Structure of the two majority coincidence (n-out-of-40) registers; the binary value in these registers is the number n of FTU trigger primitives required to trigger an event (physics or calibration)

In addtion, there are several registers in the static data block to define delays (e.g. for the trigger). Also a general dead time to be applied after each trigger can be set (to compensate for the delay of the busy line). The clock conditioner settings are specified at address 0x00D to 0x01C (LMK03000 from National Semiconductor, for more details see [2]). Starting at address 0x020, the FTU settings are stored. The FTM always holds the complete FTU parameters in the static data block. For the meaning of these registers, please refer to the FTU firmware specifications document [10]. In case not all FTUs are connected during e.g. the testing phase, or a FTU is broken, the 'active FTU list' registers can be used to disable certain boards.

4.2 Dynamic data block

The dynamic data block shown in table 4.10 contains permanently updated data stored inside the FTM FPGA. It contains the actual on-time counter reading, the board temperatures and the trigger rates measured by the FTUs. This data block is updated and sent periodically by the FTM. Thus the main control software receives periodically a corresponding data package via ethernet. Usually the shortest counting interval of the FTUs ('prescaling' register) defines the period. The on-board 12-bit temperature sensors are MAX6662 chips from Maxim Products. For more information about these components and their data see [9]. When sending the dynamic data block, the header defined in table 4.1 is added at the beginning.

word no	content
0x000	on-time counter at read-out time bits 47...32
0x001	on-time counter at read-out time bits 31...16
0x002	on-time counter at read-out time bits 15...0
0x003	temperature sensor 0: component U45 on the FTM schematics [3]
0x004	temperature sensor 1: U46
0x005	temperature sensor 2: U48
0x006	temperature sensor 3: U49
0x007	rate counter bit 29...16 patch 0 board 0 crate 0
0x008	rate counter bit 15...0 patch 0 board 0 crate 0
0x009	rate counter bit 29...16 patch 1 board 0 crate 0
0x00A	rate counter bit 15...0 patch 1 board 0 crate 0
0x00B	rate counter bit 29...16 patch 2 board 0 crate 0
0x00C	rate counter bit 15...0 patch 2 board 0 crate 0
0x00D	rate counter bit 29...16 patch 3 board 0 crate 0
0x00E	rate counter bit 15...0 patch 3 board 0 crate 0
0x00F	rate counter bit 29...16 total board 0 crate 0
0x010	rate counter bit 15...0 total board 0 crate 0
0x011	Overflow register board 0 crate 0
0x012	CRC-error register board 0 crate 0
0x013	rate counter bit 29...16 patch 0 board 1 crate 0
0x014	rate counter bit 15...0 patch 0 board 1 crate 0
0x015	rate counter bit 29...16 patch 1 board 1 crate 0
0x016	rate counter bit 15...0 patch 1 board 1 crate 0
0x017	rate counter bit 29...16 patch 2 board 1 crate 0
0x018	rate counter bit 15...0 patch 2 board 1 crate 0
0x019	rate counter bit 29...16 patch 3 board 1 crate 0

0x01A	rate counter bit 15...0 patch 3 board 1 crate 0
0x01B	rate counter bit 29...16 total board 1 crate 0
0x01C	rate counter bit 15...0 total board 1 crate 0
0x01D	Overflow register board 1 crate 0
0x01E	CRC-error register board 1 crate 0
...	...

Table 4.10: FTM dynamic data block

4.3 FTU list

When the FTM board receives the 'ping all FTUs' instruction, it sends a ping command to all FTU boards and gathers the FTU boards responses to a list. This list is called 'FTU list' and shown in table 4.11. The FTM only accepts a ping when no run is ongoing (defined by the 'start run' and 'stop run' commands). When the FTU list is complete, it is sent back via ethernet with the header defined in table 4.1.

address	content
0x000	total number of responding FTU boards
0x001	number of responding FTU boards belonging to crate 0
0x002	number of responding FTU boards belonging to crate 1
0x003	number of responding FTU boards belonging to crate 2
0x004	number of responding FTU boards belonging to crate 3
0x005	crate number and address of the first responding FTU board
0x006	DNA of first responding FTU board bit 63 ... 48
0x007	DNA of first responding FTU board bit 47 ... 32
0x008	DNA of first responding FTU board bit 31 ... 16
0x009	DNA of first responding FTU board bit 15 ... 0
0x00A	crate number and address of the second responding FTU board
0x00B	DNA of second responding FTU board bit 63 ... 48
0x00C	DNA of second responding FTU board bit 47 ... 32
0x00D	DNA of second responding FTU board bit 31 ... 16
0x00E	DNA of second responding FTU board bit 15 ... 0
...	...

Table 4.11: FTU list

Because there are four connectors for the RS-485 buses, one for each crate, there will be a full scan for all addresses on every connector. In case there is no response to a 'ping' on a certain address, there will be up to two repetitions. A responding FTU board will get an entry into the FTU list (table 4.11) including the number of 'ping' sent until response. The crate number (connector number) and the number of pings are coded together with the FTU board address as shown in table 4.12. While 'Cr0' and 'Cr1' indicate the crate number (connector number), 'pings_0' and 'pings_1' contain the number of 'pings' until response of an FTU board, both coded in binary. The 'DNA' of the FTU board is the device DNA [5, 6, 7, 8] of the FPGA on the responding FTU board. This is a unique 57 bit serial number unambiguously identifying every Xilinx FPGA. In the most significant word (bit 63 ... 48) bits 63 town to 57 are filled with zeros.

Bit	15	14	13	12	11	10	9	8	7	6	5	...	0
Content	Cr1	Cr0	pings_1	pings_0	x	x	x	x	x	x	A5	...	A0

Table 4.12: Crate number and address of first responding FTU board

Chapter 5

FTU communication error handling

When the FTM board is communicating with a FTU board via RS-485, the FTU board has to respond within 5 ms. If this timeout expires, or the response sent back by the FTU board is incorrect, the FTM resends the datapacket after the timeout. If this second attempt is still unsuccessful, a third and last attempt will be made by the FTM board. An error message will be sent to the central control whenever a FTU board does not send a correct answer after the first call by the FTM board. This message (see table 5.1) contains, after the standard header (see table 4.1), the number of unsuccessful calls and the data packet sent to the FTU board in these unsuccessful calls. In order to avoid massive error messages for e.g. test setups with single FTUs, the 'active FTU list' can be employed to disable FTUs from the bus. In that case the FTM will not try to contact the corresponding boards.

word no	content
0x000	board ID bits 63...48
0x001	board ID bits 47...32
0x002	board ID bits 31...16
0x003	board ID bits 15...0
0x004	firmware ID
0x005	Trigger counter at read-out time bits 31...16
0x006	Trigger counter at read-out time bits 15...0
0x007	Time stamp counter at read-out time bits 47...32
0x008	Time stamp counter at read-out time bits 31...16
0x009	Time stamp counter at read-out time bits 15...0
0x00A	spare
0x00B	number of unsuccessful calls
0x00C ... 0x027	slow control data packet sent to FTU (28 byte)

Table 5.1: FTU communication error message

Bibliography

[1] Paul Scherrer Institut PSI. *DRS4 9 Channel, 5 GSPS Switched Capacitor Array*. datasheet.

[2] National Semiconductor Corporation. *LMK03000 Family Precision Clock Conditioner with integrated VCO*, 2008. datasheet.

[3] ETH Zürich, IPP. *FTM Schematics*, 2010.

[4] WIZnet Co.Ltd. *W5300 Fully Hardwired Network protocol Embedded Ethernet Controller*, 2008. datasheet.

[5] Xilinx. *Spartan-3AN FPGA Family Data Sheet*, 2009.

[6] Xilinx. *Spartan-3A DSP FPGA Family: Data Sheet*, 2009.

[7] Xilinx. *Advanced Security Schemes for Spartan-3A/3AN/3A DSP FPGAs*, 2007.

[8] Xilinx. *Security Solutions Using Spartan-3 Generation FPGAs*, 2008.

[9] Maxim Integrated Products. *12-Bit plus Sign Temperature Sensor with SPI-Compatible Serial Interface MAX6662*, 2001. datasheet.

[10] ETH Zürich, IPP. *FTU Firmware Specifications v3*, 2010.