Katherine

Ethernet Embedded Readout Interface for Timepix 3

Command Set

Petr Burian

burianp@kae.zcu.cz

Faculty of Electrical Engineering, University of West Bohemia
Institute of Experimental and Applied Physics, Czech Technical University in Prague

Content

1 COMMUNICATION DESCRIPTION 2

1.1 FUNDAMENTAL COMMUNICATION 2
1.2 COMMANDS 2
1.2.1 ACQUISITION TIME SETTING 3
1.2.2 BIAS SETTING 3
1.2.3 INTERNAL DAC SETTINGS 3
1.2.4 SENSOR REGISTER SETTINGS 4
1.2.5 ACQ. MODE SETTING 5
1.2.6 ECHO CHIP ID 5
1.2.7 GET BIAS VOLTAGE 6
1.2.8 NUMBER OF FRAMES SETTING 6
1.2.9 GET HW/READOUT TEMPERATURE 6
1.2.10 GET SENSOR TEMPERATURE 6
1.2.11 GET READOUT STATUS 7
1.2.12 HW COMMAND START 7
1.2.13 INTERNAL DAC SCAN 7
1.2.14 GET ADC VOLTAGE 9
1.2.15 DIGITAL TEST 9
1.2.16 ACQUISITION START 10
1.2.17 ACQUISITION STOP 10
1.2.18 ACQUISITION SETUP 10
1.2.19 SET ALL PIXEL CONFIG 11
1.2.20 GET COMMUNICATION STATUS 13
1.2.21 TOA CALIBRATION SETUP 13

2 MEASUREMENT DATA 13

2.1 PIXEL MEASUREMENT DATA 14
2.2 NEW FRAME ESTABLISHED 15
2.3 CURRENT FRAME FINISHED 15
2.4 PIXEL TIMESTAMP OFFSET (ONLY FOR DATA-DRIVEN MODE) 15
2.5 START OF FRAME TIMESTAMP LSB & END OF FRAME TIMESTAMP LSB 15
2.6 START OF FRAME TIMESTAMP MSB & END OF FRAME TIMESTAMP MSB 15
2.7 NUMBER OF LOST PIXELS 16
2.8 MEASUREMENT ABORTED NOTICE 16

1 Communication Description

1.1 Fundamental Communication

Command:

PC => Readout: 8-byte UDP datagram on port 1555

63..48	47..0
Command ID	**Command Data (CD[47..0])**

All commands return response – data or acknowledge.

Response on Command:

Readout => PC: 8-byte UDP datagram on port 1555

63..48	47..0
Command Response ID	**Command Response Data (CRD[47..0])**

Acknowledge**:** Command Response ID = Command ID, CRD = 0x00000000

Measurement Data:

Readout => PC: UDP datagram on port 1556

1.2 Commands

Command ID	Description
0x01	Acquisition Time Setting - LSB
0x02	Bias Setting
0x03	Acquisition Start
0x04	Internal DAC Settings
0x05	Seq. Readout Start
0x06	Acquisition Stop
0x07	HW Command Start
0x08	Sensor Register Setting
0x09	Acquisition Mode Setting
0x0A	Acquisition Time Setting - MSB
0x0B	Echo Chip ID
0x0C	Get Bias Voltage
0x0D	Get ADC Voltage
0x0E	Get Back Read Register
0x0F	Internal DAC Scan
0x10	Set Pixel Config
0x11	Get Pixel Config

Command ID	Description
0x12	Sett All Pixel Config
0x13	Number of Frames Setting
0x14	Get All DAC Scan
0x15	Get HW/Readout Temperature
0x16	LED settings
0x17	Get Readout Status
0x18	Get Communication Status
0x19	Get Sensor Temperature
0x20	Digital Test
0x28	ToA Calibration Setup
0x29	Set the number of tokens

1.2.1 Acquisition Time Setting

LSB of acq. time is 10ns, it follows:

ACQT = required_time[s] / 10ns

Setting Time LBS:

Command ID = 0x01: CD[31..0] = ACQT[31..0]

Setting Time MBS:

Command ID = 0x0A: CD[31..0] = ACQT[63..32]

Response:

ACK

1.2.2 Bias Setting

Bias value is sent as float value. There is no recalculation.

Note: Katherine Mini implements the only bias voltage – bias_id=0.

Command ID = 0x02

CD[39..32] = bias_id (0-8)

CD[31..0] = float_value_bias

Response:

ACK

1.2.3 Internal DAC Settings

It sets internal DAC of Timepix3.

DAC Index	Internal DAC
0	Ibias_Preamp_ON

DAC Index	Internal DAC
1	Ibias_Preamp_OFF
2	VPreamp_NCAS
3	Ibias_Ikrum
4	Vfbk
5	Vthreshold_fine
6	Vthreshold_coarse
7	Ibias_DiscS1_ON
8	Ibias_DiscS1_OFF
9	Ibias_DiscS2_ON
10	Ibias_DiscS2_OFF
11	Ibias_PixelDAC
12	Ibias_TPbufferIn
13	Ibias_TPbufferOut
14	VTP_coarse
15	VTP_fine
16	Ibias_CP_PLL
17	PLL_Vcntrl

Command ID = 0x04

CD[39..32] = dac_index

CD[15..0] = dac_value

Response:

ACK

1.2.4 Sensor Register Settings

It sets internal registers of Timepix3. Note: Layout does not correspond with Timepix3 manual.

This command only saves data to readout HW. Use HW command with ID=0.

Register Index	Register
0	Test Pulse Period
1	Number Test Pulses
2	Out Block Config
3	PLL Config
4	General Config

Register Index	Register
5	SLVS Config
6	Power Pulsing Pattern
7	SetTimer 15..0
8	SetTimer 31..16
9	SetTimer 47..32
10	Sense DAC Selector
11	Ext DAC Selector

Command ID = 0x08

CD[39..32] = register_index

CD[31..0] = register_value

Response:

ACK

1.2.5 Acq. Mode Setting

It defines acq. mode of Timepix3.

Command ID = 0x09

CD[1..0] = acq_mode (“00” => ToA & ToT; “01” => only ToA; “10” => Event & iToT; “11” => prohibited state);

CD[8] = fast_vco_en (‚0‘ => disabled; ‚1‘ => enabled)

Response:

ACK

1.2.6 Echo Chip ID

It returns chip ID of readout. This command can be used as echo.

Command ID = 0x0

Response:

Command Response ID = 0x0B

CRD[31..0] = timepix3_chip_id

Chip ID decoding example:

```
char* chipID2string(int chip_id_int)
{
char *temp;

int x = (int)((chip_id_int & 0xF)-1);
```

```
int y = (int)((chip_id_int >>4)&0xF);
int w = (int)((chip_id_int >> 8) & 0xFFF);

temp = (char *)malloc(15 * sizeof(char));

sprintf(temp, "%c%d-W000%d", 65+x, y, w);

return temp;
}
```

1.2.7 Get Bias Voltage

It returns bias voltage.

Command ID = 0x0C

CD[39..32] = bias_id (0-8)

Response:

Command Response ID = 0x0C

CRD[31..0] = bias_voltage (float value)

1.2.8 Number of Frames Setting

This command sets the number of frames for frame-based measurement.

Command ID = 0x13

CRD[31..0] = number_required_frames

Response:

ACK

1.2.9 Get HW/Readout Temperature

It returns the current temperature of readout HW.

Command ID = 0x15

Response:

Command Response ID = 0x015

CRD[31..0] = hw_temperature_float_value

1.2.10 Get Sensor Temperature

It returns the current temperature of readout chip (sensor).

Command ID = 0x19

Response:

Command Response ID = 0x019

CRD[31..0] = sensor_temperature_float_value

1.2.11 Get Readout Status

It returns the status information of readout HW.

Command ID = 0x17

Response:

Command Response ID = 0x017

CRD[7..0] = hw_type (0x1 => Katherine Mini Ethernet Readout, 0x02 => next devices…..)

CRD[15..8] = hw_revision (number of HW revision)

CRD[31..16] = hw_serial_number (serial number of readout)

CRD[47..32] = fw_version (version of firmware)

1.2.12 HW Command Start

The command starts HW internal command in readout.

Only a few of them make sense for final user and application SW.

Command ID = 0x07

CD[7..0] = readout_hw_command_id

HW Command ID	Description
0	Sensor Config Registers Update
1	Internal DAC Update
2	Internal DAC Back Read
3	Timer Read
4	Timer Set
5	Reset Matrix Sequential
6	Stop Matrix Command
7	Load Column Test Pulse Register
8	Read Column Test Pulse Register
9	Load Pixel Register Configuration
10	Read Pixel Register Configuration
11	Read Pixel Matrix Sequential Setting
12	Read Pixel Matrix Data-Driven Setting
13	Chip ID Read
14	Output Block Config Update
15	Digital Test

Response:

ACK

1.2.13 Internal DAC Scan

The command returns analog value of internal DAC of Timepix3.

DAC ID	DAC

DAC ID	DAC
1	Ibias_Preamp_ON
2	Ibias_Preamp_OFF
3	VPreamp_NCAS
4	Ibias_Ikrum
5	Vfbk
6	Vthreshold_fine
7	Vthreshold_coarse
8	Ibias_DiscS1_ON
9	Ibias_DiscS1_OFF
10	Ibias_DiscS2_ON
11	Ibias_DiscS2_OFF
12	Ibias_PixelDAC
13	Ibias_TPbufferIn
14	Ibias_TPbufferOut
15	VTP_coarse
16	VTP_fine
17	Ibias_CP_PLL
18	PLL_Vcntrl
28	BandGap output
29	BandGap_Temp
30	Ibias_dac
31	Ibias_dac_cas

Command ID = 0x0F

CD[7..0] = dac_ID

Response:

CRD[31..0] = float_voltage_value

1.2.14 Get All DAC Scan

It returns voltage values of all internal DACs in the only sequence.

Command ID = 0x14

CD = 0x0000

Response:

There are 22 responses with all dac scan values.

Order	DAC ID	DAC
1	1	Ibias_Preamp_ON
2	2	Ibias_Preamp_OFF
3	3	VPreamp_NCAS
4	4	Ibias_Ikrum
5	5	Vfbk
6	6	Vthreshold_fine
7	7	Vthreshold_coarse
8	8	Ibias_DiscS1_ON
9	9	Ibias_DiscS1_OFF
10	10	Ibias_DiscS2_ON
11	11	Ibias_DiscS2_OFF
12	12	Ibias_PixelDAC
13	13	Ibias_TPbufferIn
14	14	Ibias_TPbufferOut
15	15	VTP_coarse
16	16	VTP_fine
17	17	Ibias_CP_PLL
18	18	PLL_Vcntrl
19	28	BandGap output
20	29	BandGap_Temp
21	30	Ibias_dac
22	31	Ibias_dac_cas

CRD[31..0] = float_voltage_value

1.2.15 Get ADC Voltage

It returns voltage of ADC periphery. Just for debugging.

Command ID = 0x0D

CD[7..0] = adc_ch_id (0-255; id of ADC channel)

Response:

Command Response ID = 0x0D

CRD[31..0] = voltage (float value)

1.2.16 Digital Test

It starts testing of communication between readout and sensor.

Command ID = 0x20

CD[31..0] = 0

Response:

Command Response ID = 0x20

CRD[7..0] = response (64 => Ok, otherwise => test failed)

1.2.17 Acquisition Start

It starts acquisition - measurement.

Command ID = 0x03

CD[0] = readout_mode (‘0’ => Sequential readout mode, ‘1’ => Data-driven readout mode)

Response:

Command Response ID = 0x03

Measurement data…

1.2.18 Acquisition Stop

It quits/breaks current measurement.

Command ID = 0x06

CD[0] = readout_mode (‘0’ => Sequential readout mode, ‘1’ => Data-driven readout mode)

Response:

Command Response ID = 0x06

Measurement data…End of Frame

1.2.19 Acquisition Setup

Sets conditions for start and end of acquisition.

Command ID = 0x21

CD[15..0] see table:

CD bit index	Meaning	Note
0	Acquisition Start	0 => Start immidiately, 1 => start by hw event (trigger)
1	Channel of hw event (trigger) - start	Channel of trigger for start.
2		
3		
4	Edge setup - start	0 => rising edge of trigger, 1 => falling edge of trigger
5	Delayed start by HW event	0 => Start of acq. immidiately, 1 => Start of acq. is delayed
6	-	unused
7	-	unused
8	Acqusition Finish	0 => by timer, 1 => by hw event (trigger)
9	Channel of hw event (trigger) - finish	Channel of trigger for end.
10		
11		
12	Edge setup - finish	0 => rising edge of trigger, 1 => falling edge of trigger
13	-	unused
14	-	unused
15	-	unused

CD[40..33] = 0x05

Else CD bits in low.

1.2.20 Set All Pixel Config

The command sends configuration for all pixels to readout HW (not to sensor). To load config by chip, you have to perform following HW commands: 5, 9. In future, these steps will be accomplished by HW automatically and command will return result of configuration verification.

Command ID = 0x12

CD = null

After Command datagram, readout expects 65536 bytes with configuration.

Response:

ACK

*Example code – converting config data from PIXiet format (`filename` - *.bpc 65kB binary file) to "Katherine" format (`fpga_config_mem_order` – 16384 32-bit unsigned words):*

```
byte[] pixel conf bytes = new byte[1];
pixel_conf_bytes = System.IO.File.ReadAllBytes(filename);

byte[] reverse_array = { 0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15 };

UInt32[] fpga_config_mem_order = new UInt32[16384];

byte[,] manual_format = new byte[256, 256];

for (int i = 0; i < pixel_conf_bytes.Length; i++)
{
    byte tmp = pixel_conf_bytes[i];

    uint y = (uint)(i / 256);
    uint x = (uint)(i % 256);

    tmp = (byte)((tmp & 0x21) | (reverse_array[((tmp & 0x1E) >> 1)] << 1));

    manual_format[y, x] = tmp;

}

for (int i = 0; i < pixel_conf_bytes.Length; i++)
```

```
{

   uint y = (uint)(i / 256);
   uint x = (uint)(i % 256);

   byte tmp = manual_format[y, x];

   y = 255 - y;

   uint tmp1 = (64 * x) + (y >> 2);

   uint tmp2 = (8 * (3 - (y % 4)));

   fpga_config_mem_order[(64 * x) + (y >> 2)] |= (UInt32)((tmp << (int)(8 * (3 - (y % 4)))));
}
```

Example code – converting config data from BMC to "Katherine" format (`fpga_config_mem_order` – 16384 32-bit unsigned words). The parameter of function called matrixoconfigdata stands for byte contain of file.:

```
public void ChipConfig(byte[] matrixconfigdata)
  {
    UInt32[] fpga_config_mem_order = new UInt32[16384];

   for (int i = 0; i < (256 * 256); i++)
   {
     uint y = (uint)(i / 256);
     uint x = (uint)(i % 256);
     byte tmp = (byte)matrixconfigdata[i];

     y = 255 - y;
    fpga_config_mem_order[(64 * x) + (y >> 2)] |= (UInt32)((tmp << (int)(8 * (3 - (y % 4)))));
   }

  }
```

Example code – send the whole matrix config to the readout:

```
public void SetAllConfigRegister(UInt32[] matrix_config)
{
        int MessageID = 0x12;

        Byte[] senddata = new Byte[8];

        for (int i = 0; i < senddata.Length; i++)
           senddata[i] = 0;

        senddata[6] = (Byte)MessageID;

        udpClient.Send(senddata, senddata.Length);

        byte[] byteArray = matrix_config.SelectMany(BitConverter.GetBytes).ToArray();

        MemoryStream mem_stream = new MemoryStream(byteArray);

        using (BinaryReader reader = new BinaryReader(mem_stream))
        {
           for (int i = 0; i < 64; i++)
              udpClient.Send(reader.ReadBytes(1024), 1024);
        }

}

Note: It is preferable to less UDP datagrams with bigger payload.
```

BMC file describes the matrix configuration by 65536 bytes. Each byte corresponds with the pixel.

7	6	5	4	3	2	1	0

unused	unused	Test bit	Loc Th 3	Loc Th 2	Loc Th 1	Loc Th 0	Mask bit

1.2.21 Get Communication Status

It enables or disables on-line correction of ToA values in corrupted columns or in situation when shifted clock are applied to pixel matrix.

Command ID = 0x18

Response:

Command Response ID = 0x018

CRD[7..0] = comm_lines_mask (It shows active lines)

CRD[15..8] = total_data_rate (this value * 5 = Total Data Rate [Mbs] between readout HW and sensor chip)

CRD[23..16] = chip_deteced_flag (0 => chip not found, 1=> chip found)

1.2.22 ToA Calibration Setup

It returns the status of communication between readout HW and sensor chip.

Command ID = 0x28

CD[0] = toa_calibration_enabled (‘0’ => disabled, ‘1’ => enabled)

Response:

ACK

1.2.23 Set the number of tokens

It sets the number of tokens being used for frame-based mode. For Katherine device, value range from 1 to 17 is relevant. The higher value the faster readout process of frame-based data.

Reason for using: on badly designed HW, low number of tokens can reduce current peaks on digital power supply. When standard chipboard is used it is recommended to value of 17 to accelerate reading process.

Command ID = 0x29

CD[0] = number_tokens

Response:

ACK

2 Measurement Data

Measurement data are formed as 6-byte vector. According to header data part they can contain pixel measurement data or specific data describing acquisition.

47..44	**43..0**
Header	**Data**

Header	Description
0x7	New Frame Established
0x4	Pixel Measurement Data
0x5	Pixel Timestamp Offset (only for data-driven mode)
0xC	Current Frame Finished
0x8	Start of Frame Timestamp LSB
0x9	Start of Frame Timestamp MSB
0xA	End of Frame Timestamp LSB
0xB	End of Frame Timestamp MSB
0xD	Number of Lost Pixels
0xE	Measurement Aborted Notice

Measurement process – Frame-based readout mode:

1) Client sends *Acquisition Start* command to the readout.
2) The readout starts first frame and sends *New Frame Established* frame to client
3) Measurement is in progress. The client has to wait for the end of frame.
4) If there were some hits the readout sends *Pixel Measurement Data* frames with hits. If not this stage is omitted.
5) The readout sends Start/End frame timestamp frame; this means frames with headers 0x8, 0x9, 0xA and 0xB are sent.
6) Finally, *Current Frame Finished* data frame is sent. If the required number of frames is achieved, the measurement is over. If not, the stage 1) goes again.

Measurement process – Data-driven Readout Mode:

1) Client sends *Acquisition Start* command to the readout.
2) The readout starts "frame" (the whole measurement is expressed by one frame) and sends *New Frame Established* frame to client
3) If there are some hits the readout immediately sends *Pixel Measurement Data* frames with hits.
4) The time of measurement is ran out – shutter is close.
5) The readout sends Start/End frame timestamp frame; this means frames with headers 0x8, 0x9, 0xA and 0xB are sent. For this readout mode, start timestamp of frame is zero value.
6) Finally, *Current Frame Finished* data frame is sent. The measurement is over.

2.1 Pixel Measurement Data

Selected detector mode and Fast VCO functionality affects data frame with measurement pixel data. See the following self-explaining table. As it has been noted above, pixel data use 0x4 header.

Fast VCO	Mode	Measure Data Frame					
		47..44	***43..36***	***35..28***	***27..14***	***13..4***	***3..0***
Enabled	ToA & ToT	0x4	Coordinate Y	Coordinate X	ToA	ToT	FastToA
	Only ToA	0x4	Coordinate Y	Coordinate X	ToA	---	FastToA
	Event Count and iToT	0x4	Coordinate Y	Coordinate X	iToT	Event Counter	---

Disabled	ToA & ToT	0x4	Coordinate Y	Coordinate X	ToA	ToT	Hit Counter
	Only ToA	0x4	Coordinate Y	Coordinate X	ToA	---	Hit Counter
	Event Count and iToT	0x4	Coordinate Y	Coordinate X	iToT	Event Counter	Hit Counter

The LSB of ToA expresses 25ns; the LSB of FastToA means 1.5625ns. Total local timestamp is given by: Timestamp = ToA – FastToA.

2.2 New Frame Established

This data frame indicates the readout starts to measure new frame. During frame-based readout mode the readout device sends this frame before each frame. If data-driven mode is activated, the frame is sent only at the beginning of measurement (acquisition).

47..44 (Header)	43..0 (Data)
0x7	0x00000000000

2.3 Current Frame Finished

This data frame indicates the end of frame. During frame-based readout mode the readout device sends this frame after each frame. If data-driven mode is activated, the frame is sent only at the end of the whole measurement (acquisition). The frame also tells us how many pixel data frames were sent to client.

47..44 (Header)	43..0 (Data)
0xC	Number of Sent Pixels

2.4 Pixel Timestamp Offset (only for data-driven mode)

This data frames sends time offset for timestamp of pixels. The final timestamp (ToA) of hit is expressed as sum of this offset multiplied by constant of 16384 and ToA value. This frame is sent when pixel hit needs the new value of pixel offset. Also each UDP datagram with pixel data begins by this frame.

47..44 (Header)	43..32	31..0 (Data)
0x5	0x000	Time Offset

2.5 Start of Frame Timestamp LSB & End of Frame Timestamp LSB

These frames send LSB part of timestamp of start/end frame according to internal counter in Timepix3.

47..44 (Header)	43..32	31..0 (Data)
0x8	0x000	Start of Frame Timestamp LSB

47..44 (Header)	43..32	31..0 (Data)
0xA	0x000	End of Frame Timestamp LSB

2.6 Start of Frame Timestamp MSB & End of Frame Timestamp MSB

These frames send MSB part of timestamp of start/end frame according to internal counter in Timepix3.

47..44 (Header)	43..16	15..0 (Data)
0x9	0x0000000	Start of Frame Timestamp MSB

47..44 (Header)	43..16	15..0 (Data)
0xB	0x0000000	End of Frame Timestamp MSB

2.7 Number of Lost Pixels

This vector indicates the number of lost pixels during measurement (within the current frame). “Lost pixels” stands for how many hit pixels were discarded due to full internal buffers in the readout device. Pixel loss occurs when hit rate of detector is higher than Ethernet internet throughput rate.

47..44 (Header)	43..0 (Data)
0xD	Number of Lost Pixels

2.8 Measurement Aborted Notice

This message tells us that measurement was aborted before defined time limit because of user request (*Acquisition Stop Command*).

47..44 (Header)	43..0 (Data)
0xE	0x00000000000