

Radmu GTT

Zynq Ultrascale+

Software Reference Manual

INFN Padova

By Lorenzo Castellani

The interface consists of a web page that implements a series of forms for setting the operating parameters, access to all the registers of the various chips and a WebSocket Server (RFC 6455) which responds on port 4444.

LabView library to communicate with the server is available
<https://www2.pd.infn.it/~caste/pub/WebSockets.zip>

Preliminary commands implemented.

Binary commands accepted by server are:

Commands implemented with WebSocket opt code = binary frame.

The format of the frames: [cmd] [arg1] [argn].

The first byte represents the command followed by the arguments.

The following structures describe the arguments for each command, UINT32 represents an integer 32bit (4 bytes) in little-Endian format, FLOAT64 an 8byte floating point in little-Endian format.

OuptutDisable, code 0x01

Disable output versus DAQ board.

BYTE cmd;

BYTE ch; //channel 0-7

BYTE disable; //1=disabled 0=enabled

Return :

BYTE cmd; //0x81

UINT32 t_out; //content of three state register

Enable Trigger, code 0x02

Enable trigger channel 0-7

BYTE cmd;

BYTE ch; //channel 0-7

BYTE enable; //0=disable 1=enabled

Return :

BYTE cmd; //0x82

UINT32 enable; //content of enable register

Increment L1A delay, code 0x05

Increment L1A input and output (L1A back to DAQ) delay channel 0-7, is used to compensate PCB different routers length

BYTE cmd;

BYTE ch; //channel 0-7

Return : the same frame

Decrement L1A delay, code 0x06

Decrement L1A input and output (L1A back to DAQ) delay channel 0-7, is used to compensate PCB different routers length

BYTE cmd;

BYTE ch; //channel 0-7

Return : the same frame

Monostable , code 0x07

Set trigger CFG

```
BYTE cmd;  
  
BYTE ch;    //channel 0-7  
  
BYTE mono;  //monostable delay (unit 40ns) 0=disabled
```

Return : the same frame.

Trigger LUT CFG, code 0x03

```
BYTE cmd;  
  
BYTE lutbit; //bit number 0-255  
  
BYTE val;    //value 0/1  
  
Return :  
  
BYTE cmd;    //0x83  
  
UNIT32 LUT[8]; //the GFG (all 256 LUT bits)
```

Status, code 0x84

```
BYTE cmd;  
  
Return :  
  
BYTE cmd;    //0x04  
  
UNIT32 LUT[8]; //the Trigger LUT GFG (all 256 LUT bits)  
  
UINT32 t_out; //content of three state register for the output  
  
UINT32 monot0123; //monostable delay  
  
UINT32 monot4567;  
  
UINT32 enretrig; //triger enable  
  
UINT32 delay01;  // L1A delay  
  
UINT32 delay23;  
  
UINT32 delay45;  
  
UINT32 delay67;  
  
UINT32 L1AMODE; // selezione outL1Amux bit 0 - 3, bit 4 selezione in extL1A
```

L1A Mode, code 0x08

BYTE cmd;

UINT32 L1AMODE; // selezione outL1Amux bit 0 - 3, bit 4 selezione in extL1A

Return : the same frame.

Read L1A Mode, code 0x88

BYTE cmd; //0x08

UINT32 L1AMODE; // selezione outL1Amux bit 0 - 3, bit 4 selezione in extL1A

TTC Broadcast, code 0x11

Enable channel

BYTE cmd;

BYTE brdcst; // 1=BcReset, 2=EventCnt Reset, 4=Testpulse

Return : the same frame.

Si5338 PLL status and Xilinx Pll, code 0x9D

Get Si5338 PLL status

BYTE cmd;

BYTE reset; //reset nLoseLock;

Return :

BYTE cmd; //0x1D

BYTE pllstatus; //0x00 = OK pll locked

INT32 nLoseLock; // Lose lock number

Return : the same frame **Return data on error** , code 0xFF:

BYTE cmd; //0xFF

INT32 errorcode; // Error code

Write configuration File, code 0x41

Save configuration file

```
BYTE cmd;  
  
CSTRING filename; //C string with termination char 0. If string is empty write default  
//configuration file  
  
BYTE data[size]; //the configuration data  
  
Return the same frame;
```

Read configuration File, code 0xC1

Read configuration file

```
BYTE cmd;  
  
CSTRING filename; //C string with termination char 0. If string is empty write default  
  
Return:  
  
BYTE cmd; //0x41  
  
CSTRING filename; //C string with termination char 0. If string is empty is default configuration  
  
BYTE data[size]; //the configuration data
```

Set configuration, code 0x42

Set configuration

```
BYTE cmd;  
  
BYTE data[size]; //the configuration data  
  
Return the same frame;
```

Get configuration, code 0xC2

Get configuration

BYTE cmd;

Return :

BYTE cmd; //0x42

BYTE data[size]; //the configuration data

Error code:

- 1 Not Authorized
- 22 Invalid Value
- 5 I/O Error
- 9 Unknow command (connection will be closed)
- 2 no such file
- 13 Permission denied
- 16 Busy

See linux c/c++ error base for undefined number (errno-base.h)

TEXT commands accepted by server are:

Temperature? Return the temperature

Version? Return the App version string

SiStatus? The Si5338 PLL status