# 

**Physics Division -- *Fast* Electronics Group**

**Description and Instructions**

**For LDRD\_QIS Firmware Version 0x3800**

**Mar 2021**

**Hai Dong**

# 1.0 Introduction

This document describes the firmware of the EFADC250\_DAQ. It has 7 unique functions as follows:

1. Read Out Processing
2. Host Interface
3. Play Back
4. IC Configuration
5. Miscellaneous

The IC Configuration and Host Interface functions are running at 125 MHz to support the 1GBits Ethernet.

The remaining functions are running nominally at 250MHz clock rate and all the time references listed in this document are Number of Samples \* 4 nS. However the firmware with minor modification can also be ran at slower clock. In this case all the time references listed in this document are Number of Sample \* 1/clock.

**Read Out Processing**

# 2.0 Read Out Processing

**Experiment Connection:**

**1 Gbits Ethernet**

**Trigger**

**Analog**

**0-5mV**

8

1

Host PC

Laser

Detector

EFADC250

\*

Pre-AmpMaybe needed \*\*

50 Ohm

**Digital**

**0**-**4V (low: 0-0.2V**

**High: 2-4V)**

\*

Need to change amp gain to accommodate 0-5mV detector output.

Won’t be able to use full range of 1V of ADC IC; best is 50mV with 5mV in.

\*\*

Might need to improve signal/noise ratio. Need to place close to detector

**Figure 1: Signal Processing**

Tf

Tr

Vp

IW

TETHI

0

5mV

2-4V

LUT

LUT

LUT

T#1

T#2

T#1

Twin

win

TETLO

There are 3 mode of operations: Analysis, Verifying, Streaming. Each mode starts when a Collect On command is received. Also at this time a timer (counter) is started and the count at the time Trigger signal is stored to be sent along with processed data.

**Analysis Mode (Figure 1):**

On rising edge of Trigger signal, if an X user programmable (NSAT) number of ADC samples in a channel is larger than Trigger Threshold (TET) and occurs within Look Up Time (LUT) time:

* A Y user programmable (PedNumOfAdcSamp) number of samples starting at Trigger time will be averaged to be reported as Pedestal for this trigger. Y has to be 1,2,4, or 8. If any of these ADC sample is above TETHI, the Pedestal quality bit will set.
* The samples (including this sample) in that channel will be summed until a sample goes below TETLO (sample < threshold is not included). In other words, the samples within integration window (IW) will be summed. The largest ADC count and the time when this occurs within IW will be reported. The Overflow bit of the ADC will not be included in the Sum. If a sample that is in the SUM is overflow (1 1111 1111 1111) the Overflow (Ov) bit of the Sum will be set. If a sample that is in the SUM is underflow (1 0000 0000 0000) the underflow (Uv) bit of the Sum will be set. Sum is terminated either at next trigger or 32 uS (Twin).
* The time that the 1st sample is above threshold (Tr) and the time that the 1st sample falls below threshold (Tf) will be reported. This time is relative to rising edge of the trigger signal.
* The peak ADC sample (Vp) and the time (Tp) will be reported.
* If after a Trigger signal, no NSAT number of samples is larger than TET within LUT time, report only the time of the trigger.
* A user programmable number of samples PedNumOfAdcSamp after Trigger will be averaged and report as pedestal. If an ADC sample within PedNumOfAdcSamp is greater than user programmable value (MAXPED), MaxPedDetect bit is set.
* If a pulse started within PedNumOfAdcSamp, PulseInPed is set
* Twin must be less than minimum time between pulses. If Twin is greater than time between pulses, the second pulse will not be recognized.
* If there are more than one pulse within Twin, only the parameters of the first pulse will be processed.
* The difference between TETHI and TETLO is essentially hysteresis. As such it should not be equaled. The difference should be set to be equaled the noise of the signals.

Of the three triggers shown in Figure 1, only T#1 has data to report. T#2 does not have data to report because samples cross TETHI after LUT time. T#3 does not have data to report because no or not enough sample(s) cross TETHI.

**Verifying mode:**

User can use this mode to verify that the Analysis Mode produces correct result. In this mode, result if any from Analysis Mode and a Z user programmable number (NumSampToWrOut) of ADC samples starting from trigger time will be reported. NumSampToWrOut must be even.

**Streaming mode:**

In this mode, if X number of ADC samples in a channel is larger than TET, Z number of ADC samples is reported.

For every Trigger signal, the following data will be reported for Analysis Mode:

* Trigger Time, (header)
* Trigger Number
* Chan 0 result if any.
* Chan 1 result if any.
* Chan 2 result if any.
* Chan 3 result if any.
* Chan 4 result if any.
* Chan 5 result if any.
* Chan 6 result if any.
* Chan 7 result if any,
* Chan 8 result if any.
* Trailer (end of trigger report)

For every Trigger signal, the following data will be reported for Verifying Mode:

* Trigger Time, (header)
* Trigger Number
* Chan 0 ADC samples.
* Chan 0 result if any
* Chan 1 ADC samples.
* Chan 1 result if any
* Chan 2 ADC samples.
* Chan 2 result if any
* Chan 3 ADC samples.
* Chan 3 result if any
* Chan 4 ADC samples.
* Chan 4 result if any
* Chan 5 ADC samples.
* Chan 5 result if any
* Chan 6 ADC samples.
* Chan 6 result if any
* Chan 7 ADC samples,
* Chan 7 result if any
* Chan 8 ADC samples.
* Chan 8 result if any
* Trailer (end of trigger report)

For every Trigger signal, the following data will be reported for Streaming mode:

* Trigger Time, (header)
* Trigger Number
* Chan 0 ADC samples if any.
* Chan 1 ADC samples if any.
* Chan 2 ADC samples if any.
* Chan 3 ADC samples if any.
* Chan 4 ADC samples if any.
* Chan 5 ADC samples if any.
* Chan 6 ADC samples if any.
* Chan 7 ADC samples if any.
* Chan 8 ADC samples if any.
* Trailer (end of trigger report)

**Specification :**

FPGA (ADC) clock is 250 MHz.

* Twin (Maximum ADC samples for each trigger) : 13 bits
* NumSampToWrOut must be even : 13 bits
* NSAT (number of sample above threshold 1-4) : 2 bits
  + 0 => 1 sample
  + 1 => 2 samples
  + 2 => 3 samples
  + 3 => 4 samples
* PedNumOfAdcSamp\_request (number of pedestal sample to average) : 2 bits
  + 1 => 1 sample
  + 2 => 2 samples
  + 4 => 4 samples
  + 8 => 8 samples
  + Other values are default to 1
* Trigger Time : 48 bits
* Trigger Number : 17 bits
* IW Sum (7500 samples \* 4095 max count per sample) : 25 bits
* LUT (NSAT must occurs within this time) (32.768 uS) : 12 bits
* Vp Peak amplitude of pulse within IW : 13 bits
* Tr time (in relative to trigger) when 1st sample >= TET : 13 bits
* Tf time (in relative to trigger) when last sample <= TET : 13 bits
* Tp time (in relative to trigger) of Vp : 13 bits
* TETHI sample(s) >= TETHI starts summing till sample <
  + TETLO : 12 bits : 13 bits
* MAXPED : 12 bits
* Pedestal : 12 bits

When the pulse is beyond the next trigger, the sum is ended at the next trigger.

Tf

Tr

Vp

IW

TET

0

5mV

5V

LUT

LT

LT

T#1

T#2

T#1

**Host Interface**

# 3.0 Host Interface

The firmware uses UDP Ethernet protocol for communication with the host computer for register configuration as shown in Appendix 1 Table 2. UDP Ethernet protocol transactions consists of data packets. Each UDP packet starts with 0x5A 0x5A following by Opcode as shown in Table1 and Table 2 and finally data for Opcode that consists of data. The data is for setting the registers (shown in Appendix 3), the LCD display, Play Back, and Ethernet Parameters. Appendix 1: “UDP Data Packet from Host to EFADC” and Appendix 2: “UDP Data Packet from EFADC to Host” specify the format and show an example for each Opcode. For every UDP Packet from the host there will be an “acknowledge good” or an “acknowledge bad” response from EFADC. If a command is requesting data such as “read register”, the EFADC will also send back the registers content. The packet is being clocked out at 8 nS per byte.

The firmware uses TCP Ethernet protocol to send processed data to host computer. TCP packet size is 1470 bytes and the time out is 200 uS. The number of bytes for each trigger depend on the mode and the present of pulse. Since TCP protocol will send the packet when the number of byte equaled to 1470 bytes or at time out, data of two or more triggers might be sent in same TCP packet. A packet contains less than 1460 bytes will be sent if a time out of 200 uS occurs. The user parse the packet using packet format shown in Appendix 4: TCPIP Data Format from EFADC. The TCPIP data packet for each trigger from EFADC depends on the mode of operation. The packet is being clocked out at 8 nS per byte.

Analysis mode:

* Header Word (one 32 bits)
* Trigger Time (two 32 bits)
* ADC1 Pulse Parameters (four 32 bits) if there is a pulse
* ADC2 Pulse Parameters (four 32 bits) if there is a pulse
* ADC3 Pulse Parameters (four 32 bits) if there is a pulse
* ADC4 Pulse Parameters (four 32 bits) if there is a pulse
* ADC5 Pulse Parameters (four 32 bits) if there is a pulse
* ADC6 Pulse Parameters (four 32 bits) if there is a pulse
* ADC7 Pulse Parameters (four 32 bits) if there is a pulse
* ADC8 Pulse Parameters (four 32 bits) if there is a pulse
* Event Trailer (one 32 bits)

Verifying mode:

* Header Word (one 32 bits)
* Trigger Time (two 32 bits)
* ADC1 Window Raw Data (1+ NumSampToWrOut/2 32 bits)
* ADC1 Pulse Parameters (four 32 bits) if there is a pulse
* ADC2 Window Raw Data (1+ NumSampToWrOut/2 32 bits)
* ADC2 Pulse Parameters (four 32 bits) if there is a pulse
* ADC3 Window Raw Data (1+ NumSampToWrOut/2 32 bits)
* ADC3 Pulse Parameters (four 32 bits) if there is a pulse
* ADC4 Window Raw Data (1+ NumSampToWrOut/2 32 bits)
* ADC4 Pulse Parameters (four 32 bits) if there is a pulse
* ADC5 Window Raw Data (1+ NumSampToWrOut/2 32 bits)
* ADC5 Pulse Parameters (four 32 bits) if there is a pulse
* ADC6 Pulse Parameters (four 32 bits) if there is a pulse
* ADC6 Pulse Parameters (four 32 bits) if there is a pulse
* ADC7 Window Raw Data (1+ NumSampToWrOut/2 32 bits)
* ADC7 Pulse Parameters (four 32 bits) if there is a pulse
* ADC8 Window Raw Data (1+ NumSampToWrOut/2 32 bits)
* ADC8 Pulse Parameters (four 32 bits) if there is a pulse
* Event Trailer (one 32 bits)

Streaming mode:

* Header Word (one 32 bits)
* Trigger Time (two 32 bits)
* ADC1 Window Raw Data (1+ NumSampToWrOut/2 32 bits) if there is pulse
* ADC2 Window Raw Data (1+ NumSampToWrOut/2 32 bits) if there is pulse
* ADC3 Window Raw Data (1+ NumSampToWrOut/2 32 bits) if there is pulse
* ADC4 Window Raw Data (1+ NumSampToWrOut/2 32 bits) if there is pulse
* ADC5 Window Raw Data (1+ NumSampToWrOut/2 32 bits if there is pulse)
* ADC6 Window Raw Data (1+ NumSampToWrOut/2 32 bits if there is pulse)
* ADC7 Window Raw Data (1+ NumSampToWrOut/2 32 bits) if there is pulse
* ADC8 Window Raw Data (1+ NumSampToWrOut/2 32 bits) if there is pulse
* Event Trailer (one 32 bits)

**Play Back**

# 4.0 Play Back

User defines pulses maybe injected into the processing pipeline using a playback feature. Play Back stores 32, 13-bit ADC values in RAM and cycles through 32 ADC values on the rising edge of Trigger signal. There are 16 Play Back, one per ADC Channel. All 512 ADC values are written into memory using set playback command. Set Play Back Data in Appendix 1 shows an example of UDP packet to store 512 ADC values into RAM. When **bit 7 of Config 1** is set, Play Back outputs (instead of ADC IC outputs) are applied to all ADC processing functions. The data from Play Back is shown below (See Set Play Back Data in Appendix 1 for the relation between byte location in the packet and ADC Chan Play Back Word #

Play Back Word #

ADC Chan

C

1

C

2

3

3

1

30

31

32

1

2

C

2

3

3

1

30

31

32

1

3

C

2

3

3

1

30

31

32

1

4

C

2

3

3

1

30

31

32

1

5

C

2

3

3

1

30

31

32

1

6

C

2

3

3

1

30

31

32

1

7

C

2

3

3

1

30

31

32

1

8

C

2

3

3

1

30

31

32

1

Trigger

C

**IC Configuration**

# 5.0 IC Configuration

1. **ADC IC AD9230**

The ADC AD9230 ICs are needed to be configured after power up.

* 1. To configure all ADC ICs at one time
     1. Poll bit 15 of Status 0 for a one. This indicates firmware is ready to accept command.
     2. Write 0 to bit 7 of Config 4. Rising edge firmware sends data to AD9230
     3. Select register of AD9230 to write to by writing to bits 15-8 of Config 5. Write data to be written to register of AD9230 by writing bits 7-0 of Config 5.
     4. Set bits 7,6 and reset Bit 5 of Config 4. Bit 6 tells firmware to write to AD230. Bit 5 tells firmware to write to all AD9230
  2. For Example to configure all ADC to convert negative going signal:
     1. Configure AD9230 delay clock
        1. Poll bit 15 of Status 0 for a one.
        2. Reset bit 7 of Config 4
        3. 0x17 to bits 15-8 of Config 5 to select AD9230 ADC\_CLK\_OUT\_DELAY\_REG
        4. 0x9E to bits 7-0 of Config 5. Data to write to ADC\_CLK\_OUT\_DELAY\_REG 0x9E. Delay clock b
        5. Set bit 7 and 6, reset bit 5 of Config 4. This tells firmware to write to AD9230
        6. Poll bit 15 of Status 0 for a one
        7. Reset bit 7 of Config 4
        8. 0xFF to bit 15-8 of Config 5 to select ADC\_MASTER\_TO\_SLAVE\_REG
        9. 0x01 to bit 7-0 of Config 5. Data to write to ADC\_MASTER\_TO\_SLAVE\_REG. Tell AD9230 to execute delay clock setting.
        10. Set bit 7 and 6, reset bit 5 of Config 4
     2. Configure AD9230 to run in CML mode
        1. Poll bit 15 of Status 0 for a one.
        2. Reset bit 7 of Config 4
        3. 0x0F to bits 15-8 of Config 5 to select AD9230 ADC\_AIN\_CONFIG\_REG
        4. 0x02 to bits 7-0 of Config 5. Data to write to ADC\_AIN\_CONFIG\_REG. Run in CML mode
        5. Set bit 7 and 6, reset bit 5 of Config 4. This tells firmware to write to AD9230
        6. Poll bit 15 of Status 0 for a one
        7. Reset bit 7 of Config 4
        8. 0xFF to bit 15-8 of Config 5 to select ADC\_MASTER\_TO\_SLAVE\_REG
        9. 0x01 to bit 7-0 of Config 5. Data to write to ADC\_MASTER\_TO\_SLAVE\_REG. Tell AD9230 to execute delay clock setting.
        10. Set bit 7 and 6, reset bit 5 of Config 4
     3. Tell AD9230 to turn off test mode
        1. Poll bit 15 of Status 0 for a one.
        2. Reset bit 7 of Config 4
        3. 0x0D to bits 15-8 of Config 5 to select AD9230 ADC\_TEST\_REG
        4. 0x00 to bits 7-0 of Config 5. Data to write to ADC\_TEST\_REG. Turn off test mode
        5. Set bit 7 and 6, reset bit 5 of Config 4. This tells firmware to write to AD9230
        6. Poll bit 15 of Status 0 for a one
        7. Reset bit 7 of Config 4
        8. 0xFF to bit 15-8 of Config 5 to select ADC\_MASTER\_TO\_SLAVE\_REG
        9. 0x01 to bit 7-0 of Config 5. Data to write to ADC\_MASTER\_TO\_SLAVE\_REG. Tell AD9230 to execute delay clock setting.
        10. Set bits 7 and 6, reset bit 5 of Config 4

1. **BIAS DAC AD5516ABC-1**

This function allows configuring the 16 Bias DAC on the EFADC-16 board. The bias DAC provide the pedestal (base line) values. Pedestal is the ADC sample values when there is no input signal. To configure the Bias DAC:

* 1. Read Bit 14 of Status 2 for a one to indicate this function is ready
  2. Write Config. 5
     1. Bits 15..12 🡪 Select which Bias DAC to write to.
     2. Bits 11.0 🡪 Value to be written to Bias DAC.
  3. Write 1 to bit 15 of Config 1
  4. Write 0 to bit 15 of Config 1

1. **LCD**

This function allows user to write to “NHD-C0216CZ\_FSW\_FBW-3V3” LCD. This LCD is 2 lines of 16 characters for each line. Set LCD Data 1St Line and Set LCD Data 2nd Line in Appendix 1 show the UDP data packet to write to LCD

**Miscellaneous**

# 6.0 Miscellaneous

1. **FPGA Die Temperature**

The temperature of the FPGA die can be read at register STATUS3 (Die Temp). The Celsius temperature is calculated as follow:

DieTemp\_C = ((float)(STATUS3) \* 503.975/1024) - 273.15;

1. **Firmware Version Number**

Firmware version number can be read at register STATUS1 bits 14 to 0. Bits 15-8 is 0x38 and bits 7-0 indicates the code revision

1. **Board Serial Number**

Board serial number can be read at register STATUS2 bits 8 to 0.

1. **Time Stamp**

Time Stamp keep track of time while Collecting Data (after receiving Collect On command). It freezes when not collecting data (after receiving Collect Off command). It is reset at power up or when bit 6 of Config 1 is set.

1. **Trigger Number**

Trigger number is the number of trigger that has processed. If triggers come faster than processing time, these triggers are not processed and won’t be counted. This condition can happen in Verifying and Streaming modes when the time to send NumSampToWrOut samples is faster than the Ethernet link or the host can receive. The firmware checks that there is sufficient buffer before accepting another trigger.

**APPENDIX 1**

**UDP Data Packet from Host to EFADC**

# 7.0 Appendix 1 UDP Data Packet from Host to EFADC

**The format for sending data to the EFADC250 is as follow:**

1. 5A
2. 5A
3. Op-Opcode as shown in Table 1
4. Data

**The format for “acknowledge good” response the EFADC250 is as follow:**

1. 5A
2. 5A
3. 00
4. 03
5. FA

**The format for “acknowledge bad” response the EFADC250 is as follow:**

1. 5A
2. 5A
3. 00
4. 03
5. FE

Table 1: OpCode from Host

|  |  |  |
| --- | --- | --- |
| OP-CODE | Function | Type Of Data |
| 01 | Set Registers, LCD, Play Back | 0x0000, the bytes followed are register data (see register files chart).  0x0002, the bytes followed are to be displayed on LCD (must be 64 bytes).  0x0003, the bytes followed are to be stored in Play Back Memory |
| 02 | Activate  Command | 00: Collect Off. Collect is OFF after power up.  01 : Collect Data On  02: Reserved  03: Read Back Registers (only one time).  04: Read Play Back Memory  05: Play Back all 512 samples once. Set bit 7 of Config1 to be ADC samples.  **06**: Write new IPv4\_ADDR, SUBNET\_MASK, GATEWAY\_IP, MAC\_ADDR to config ROM  **07**: Read new IPv4\_ADDR, SUBNET\_MASK, GATEWAY\_IP, MAC\_ADDR to config ROM |

**Examples of data from Host to EFADC-15:**

Turn Collect On (See Appendix A for Registers’ Definition)

5A --- header

5A

02 -- Opcode to turn Collect on, Regs Read back request, Reset ADC

01 -- 01 indicated turn Collect on.

Processed Data from ADC is sent to PC until Turn Collect off command is received.

Turn Collect Off (See Appendix A for Registers’ Definition)

5A --- header

5A

02 -- Opcode to turn Collect on, Regs Read back request, Reset ADC

00 -- 00 indicated turn Collect off.

Request Register and Status Read Back (See Appendix A for Registers’ Definition)

5A --- header

5A

02 -- Opcode to turn Collect on, Regs Read back request, Reset ADC

03 -- 03 indicated request all Registers to be sent back

Register and Status are sent to PC once. After PC sent this command, it should wait for registers and status data to arrive before sending another command. See example of register and status below.

Set Registers (See Appendix A for Registers’ Definition)

5A --- header

5A

01 -- Opcode to set Register, PLayBack, LCD

00 -- 0000 indicates data is for registers.

00

00 -- Config 1 Hi Byte

01 -- Config 1 Lo Byte

00 -- Config 2 Hi Byte

02 -- Config 2 Lo Byte

00 -- Config 3 Hi Byte

03 -- Config 3 Lo Byte

00 -- Config 4 Hi Byte

04 -- Config 4 Lo Byte

00 -- Config 5 Hi Byte

05 -- Config 5 Lo Byte

00 -- Config 6 Hi Byte

06 -- Config 6 Lo Byte

00 -- Config 7 Hi Byte

07 -- Config 7 Lo Byte

00 -- Config 8 Hi Byte

08 -- Config 8 Lo Byte

00 -- Config 9 Hi Byte

09 -- Config 9 Lo Byte

00 -- Config 10 Hi Byte

0A -- Config 10 Lo Byte

00 -- Config 11 Hi Byte

0A -- Config 11 Lo Byte

00 -- Config 12 Hi Byte

08 -- Config 12 Lo Byte

00 -- Config 13 Hi Byte

09 -- Config 13 Lo Byte

00 -- Config 14 Hi Byte

0A -- Config 14 Lo Byte

00 -- Config 15 Hi Byte

0A -- Config 15 Lo Byte

00 -- Config 16 Hi Byte

08 -- Config 16 Lo Byte

00 -- Config 17 Hi Byte

09 -- Config 17 Lo Byte

00 -- Config 18 Hi Byte

0A -- Config 18 Lo Byte

00 -- Config 19 Hi Byte

0A -- Config 19 Lo Byte

00 -- Config 20 Hi Byte

08 -- Config 20 Lo Byte

00 -- Config 21 Hi Byte

09 -- Config 21 Lo Byte

00 -- Config 22 Hi Byte

0A -- Config 22 Lo Byte

00 -- Config 23 Hi Byte

0A -- Config 23 Lo Byte

00 -- Config 24 Hi Byte

0A -- Config 24 Lo Byte

Set LCD Data 1St Line

5A --- header

5A

01 -- Opcode to set Register, PLayBack, LCD

00 -- 0002 indicated data is for LCD data

02

02 -- LCD CMD

01 -- Return Home, 1st CHAR

03 -- LCD Char

\*\* -- LCD Char 1 ASCII

03 -- LCD Char

\*\* -- LCD Char 2 ASCII

03 -- LCD Char

\*\* -- LCD Char 3 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 4 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 5 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 6 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 7 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 8 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 9 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 10 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 11 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 12 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 13 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 14 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 15 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 16 Lo Byte

Set LCD Data 2nd Line

5A --- header

5A

01 -- Opcode to set Register, PLayBack, LCD

00 -- 0002 indicated data is for LCD data

02

02 -- LCD CMD

01 -- Go to 2nd line 1st CHAR

03 -- LCD Char

\*\* -- LCD Char 1 ASCII

03 -- LCD Char

\*\* -- LCD Char 2 ASCII

03 -- LCD Char

\*\* -- LCD Char 3 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 4 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 5 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 6 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 7 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 8 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 9 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 10 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 11 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 12 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 13 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 14 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 15 Lo Byte

03 -- LCD Char

\*\* -- LCD Char 16 Lo Byte

Set Play Back Data

5A --- header

5A

01 -- Opcode to set Register, PLayBack, LCD

00 -- 0001 indicates data is for Play Back Data.

03 --

01 -- bit 12-8 of ADC 1 Play Back Wd 1

02 -- bit 7-0 of ADC 1 Play Back Wd 1

01 -- bit 12-8 of ADC 1 Play Back Wd 2

02 -- bit 7-0 of ADC 1 Play Back Wd 2

:

:

01 -- bit 12-8 of ADC 1 Play Back Wd 32

02 -- bit 7-0 of ADC 1 Play Back Wd 32

01 -- bit 12-8 of ADC 2 Play Back Wd 1

02 -- bit 7-0 of ADC 2 Play Back Wd 1

01 -- bit 12-8 of ADC 2 Play Back Wd 2

02 -- bit 7-0 of ADC 2 Play Back Wd 2

:

:

01 -- bit 12-8 of ADC 16 Play Back Wd 31

02 -- bit 7-0 of ADC 16 Play Back Wd 31

01 -- bit 12-8 of ADC 16 Play Back Wd 32

02 -- bit 7-0 of ADC 16 Play Back Wd 32

**APPENDIX 2**

**UDP Data Packet from EFADC to Host**

# 8.0 Appendix 2 UDP Data Packet from EFAD to Host

**The format of data from EFADC250 is as follows:**

1. 5A5A
2. Op-Opcode as shown in Table 2
3. Data

Table 2: OpCode from EFADC250

|  |  |  |
| --- | --- | --- |
| OP-CODE | Function | Type Of Data |
| 03 | Data to PC | 01: Reserved  03: Register and Normal (Running) Status values.  04: Play Back Memory data  05: Reserved  07: Register and (Ethernet Parameters) Status values  FA: CMD Received is good  FE: CMD Received is bad |
|  |  |  |

**Examples of data from EFADC-15 to Host:**

Registers and Status (See Appendix A for Registers’ and Status Definition) when COnfig 1 Bit 11 is zero

5A --- header

5A

03 -- Opcode to read Register, PLayBack, LCD, etc

03 -- 03 indicates data is for registers and Status is running (normal) condition

00 -- Config 1 Hi Byte

01 -- Config 1 Lo Byte

00 -- Config 2 Hi Byte

02 -- Config 2 Lo Byte

00 -- Config 3 Hi Byte

03 -- Config 3 Lo Byte

:

:

00 -- Config 22 Hi Byte

0A -- Config 22 Lo Byte

00 -- Config 23 Hi Byte

0A -- Config 23 Lo Byte

00 -- Config 24 Hi Byte

0A -- Config 24 Lo Byte

:

:

01 -- Status 0 Hi Byte

01 -- Status 1 Lo Byte

01 -- Status 1 Hi Byte

01 -- Status 1 Lo Byte

02 -- Status 2 Hi Byte

02 -- Status 2 Lo Byte

03 -- Status 3 Hi Byte

03 -- Status 3 Lo Byte

04 -- Status 4 Hi Byte

04 -- Status 4 Lo Byte

Read Play Back Data

5A --- header

5A

03 -- Opcode to read Register, PLayBack, LCD

04 -- 04 indicates data is for Pedestal Sums

01 -- bit 12-8 of ADC ch 0 Play Back Wd 0

02 -- bit 7-0 of ADC ch 0 Play Back Wd 0

01 -- bit 12-8 of ADC ch 0 Play Back Wd 1

02 -- bit 7-0 of ADC ch 0 Play Back Wd 1

:

:

01 -- bit 12-8 of ADC ch 0 Play Back Wd 31

02 -- bit 7-0 of ADC ch 0 Play Back Wd 31

01 -- bit 12-8 of ADC ch 1 Play Back Wd 0

02 -- bit 7-0 of ADC ch 1 Play Back Wd 0

01 -- bit 12-8 of ADC ch 1 Play Back Wd 1

02 -- bit 7-0 of ADC ch 1 Play Back Wd 1

:

:

01 -- bit 12-8 of ADC ch 15 Play Back Wd 30

02 -- bit 7-0 of ADC ch 15 Play Back Wd 30

01 -- bit 12-8 of ADC ch 15 Play Back Wd 31

02 -- bit 7-0 of ADC ch 15 Play Back Wd 31

**APPENDIX 3**

**Register Definition**

# 9.0 Registers Definition

**Register Definitions:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Access** | **Runs** | **Set Ethernet** |
| STATUS 0 | R | 15..0 🡪 FirmwareVersion (0x38\_ \_) | IPv4\_addr (31..16) |
| STATUS 1 | R | 15..0 🡪BdSerialNum (0x0003) | IPv4 addr (15..0) |
| STATUS 2 | R | 15 🡪 AD9230\_READY  14 🡪 AD5516 READY  13..0 🡪 zeroes | Subnet mask (31..16) |
| STATUS 3 | R | 9..0 🡪 DieTemp | Subnet mask (15..0) |
| STATUS 4 | R | 15..0 🡪PortNumber | MAC addr (47..32) |
| STATUS 5 | R | 15..0 🡪MAC addr (31..16) | MAC addr (15..0) |
| STATUS 6 | R | 12. 0 🡪ADC 0 Pedestal |  |
| STATUS 7 | R | 12. 0 🡪ADC 1 Pedestal |  |
| STATUS 8 | R | 12. 0 🡪ADC 2 Pedestal |  |
| STATUS 9 | R | 12. 0 🡪ADC 3 Pedestal |  |
| STATUS 10 | R | 12. 0 🡪ADC 4 Pedestal |  |
| STATUS 11 | R | 12. 0 🡪ADC 5 Pedestal |  |
| STATUS 12 | R | 12. 0 🡪ADC 6 Pedestal |  |
| STATUS 13 | R | 12. 0 🡪ADC 7 Pedestal |  |
| CONFIG 1 | R/W | 15 🡪 Rising edge write AD5516 reg to DAC  14..13 🡪 NSAT  0=1 NSAT;  1=2 NSAT;  2=3 NSAT;  3=4 NSAT,  12..10 🡪  9,8 🡪 Mode  00: **Analysis** mode  01: **Verifying** mode  10: **Streaming** mode  7 🡪 Test Mode 1 = Play Back  6 🡪 1 Reset Trigger Number and Time Stamp  5-1 🡪  0 🡪 Sel Ethernet Para for Status |  |
| CONFIG 2 | R/W | 11..0 🡪 LUT. NSAT sample must occurs within this time. | IPv4\_ADDR\_to\_ROM(31.. 16) |
| CONFIG 3 | R/W | 15-12 PedNumOfAdcSamp: The number of ADC samples to average for Pedestal value. Must be 1,2,4,8. Any other value will default to 1.  11-0 MaxPed When an ADC Samples is greater than this, quality Ped Sum is set | IPv4\_ADDR\_to\_ROM(15.. 0) |
| CONFIG 4  IDLAY Set | R/W | 15..12 🡪 Select which ADC receive IDELAY control bits and read back IDELAY comparator error  11 🡪 Idelay comparator reset  10 🡪 Increment IDELAY N delay value  9 🡪 Decrement IDELAY P delay value  8 🡪 Reset IDELAY  7 🡪 rising edge write to AD9230 ADC  6 🡪 1 write to all ADC  5 🡪 0 write to AD9230  1 read from AD9230 . Data is at Stat  4 🡪 1 Reset ADC  3..0 🡪 Select ADC to write to | SUBNET\_MASK\_to\_ROM(31.. 16) |
| CONFIG 5 | R/W | 15..8 🡪 Registers inside AD9230  7..0 🡪Data to write to register. | SUBNET\_MASK\_to\_ROM(15..0) |
| CONFIG 6 | R/W | 15..12 🡪 select DAC to write DAC value bits \  11..0 🡪 DAC value | BdSerialNum to be save to ROM(15..0) |
| CONFIG 7 |  | 12-0 🡪 Twin Must be less than minimum time between pulses |  |
| CONFIG 8 | R/W | 12-0 NumSampToWrOut in verifying and Streaming mode. **Must be less than** Twin **and even** | PortNumber\_to\_ROM(15..0) |
| CONFIG 9 | R/W | 11-0 ADC 0 TETHI | MAC\_ADDR\_to\_ROM(47..32) |
| CONFIG 10 | R/W | 11-0 ADC 1 TETHI | MAC\_ADDR\_to\_ROM(31..16) |
| CONFIG 11 | R/W | 11-0 ADC 2 TETHI | MAC\_ADDR\_to\_ROM(15..0) |
| CONFIG 12 | R/W | 11-0 ADC 3 TETHI | When = x"ABCD" save BdSerialNum to ROM |
| CONFIG 13 | R/W | 11-0 ADC 4 TETHI |  |
| CONFIG 14 | R/W | 11-0 ADC 5 TETHI |  |
| CONFIG 15 | R/W | 11-0 ADC 6 TETHI |  |
| CONFIG 16 | R/W | 11-0 ADC 7 TETHI |  |
|  |  |  |  |
| CONFIG 17 | R/W | 11-0 ADC 0 TETLO |  |
| CONFIG 18 | R/W | 11-0 ADC 1 TETLO |  |
| CONFIG 19 | R/W | 11-0 ADC 2 TETLO |  |
| CONFIG 20 | R/W | 11-0 ADC 3 TETLO |  |
| CONFIG 21 | R/W | 11-0 ADC 4 TETLO |  |
| CONFIG 22 | R/W | 11-0 ADC 5 TETLO |  |
| CONFIG 23 | R/W | 11-0 ADC 6 TETLO |  |
| CONFIG 24 | R/W | 11-0 ADC 7 TETLO |  |

**APPENDIX 4**

**TCPIP Data format from EFADC**

# 10.0 TCPIP Data format from EFADC

**Event Header** (1) – indicates the start an event.

(31) = 1

(30 - 28) = 001

(27) = 0

(26 - 18) = Board Serial Number

(17 - 0) = trigger number

**Trigger Time** (2) – time of trigger occurrence relative to the most recent global reset. Time in the ADC data processing chip is measured by a 48-bit counter that is clocked by the 250 MHz system clock. The six bytes of the trigger time

Time = TA TB TC TD TE TF

are reported in two words (Type Defining + Type Continuation).

Word 1:

(31) = 1

(30 - 28) = 010

(27 - 24) = 0000

(23 - 16) = TD

(15 - 8) = TE

(7 - 0) = TF

Word 2:

(31) = 0

(30 - 28) = 010

(27 - 24) = 0000

(23 - 16) = TA

(15 - 8) = TB

(7 - 0) = TC

**Window Raw Data** (4) – raw ADC data samples for the trigger window. The first word identifies the channel number and window width. Multiple continuation words contain two samples each. The earlier sample is stored in the most significant half of the continuation word. Strict time ordering of the samples is maintained in the order of the continuation words. A *sample not valid* flag bit 13 will be set when PTW+1 is odd.

Word 1:

(31) = 1

1. - 28) = 011

(27 - 24) = channel number (0 – 15)

(23 - 13) = zeroes

(12 - 0) = NumSampToWrOut

Words 2 - N:

(31) = 0

1. - 28) = 3

(27 - 26) = 00

(25 - 13) = ADC sample x (includes overflow bit)

(12 - 0) = ADC sample x + 1 (includes overflow bit)

**Pulse Parameters** (9) – computed pulse parameters for detected pulses in a channel. The first word identifies the channel number, event number within the block, and pedestal information for the window. Multiple continuation word *pairs* contain information about the pulses detected. For a channel with hits detected:

Word 1: Channel ID and Pedestal information (reported *once* for a channel with hits)

(31) = 1

(30 - 28) = 100

(27 - 24) = channel number (0 – 15)

(23 - 15) = zeroes

(14) = PulseInPed

(13) = 0

(12) = MaxPedDetect

(11 – 0) = Pedestal

Word 2 : Integral of pulse within IW

(31) = 0

(30 - 28) = 4

(27) = 0

(26) = Ov One or more samples is overflow = 0x1FFF

(25) = Uv One or more sample is underflow = 0x1000

(24 - 0) = 25-bits sum of raw samples that constitute the pulse data set

Word 3 : Time Parameter and quality of pulse within IW

(31) = 0

(30 - 28) = 4

(27) = 0

(26) = Pulse extend beyond next trigger

(25-13) = Tf time (in relative to trigger) when last sample <= TET

(12-0) = Tr time (in relative to trigger) when 1st sample >= TET

Word 4 : Peak amplitude of pulse within IW.

(31) = 0

(30 - 28) = 4

(27 - 26) = 00

(25 - 13) = Tp time (in relative to trigger) of Vp

(12 - 0) = Vp Peak amplitude of pulse within IW

`**Event Trailer:** Indicate the end of an event.

(31) = 1

(30 - 28) = 0

(27 – 0) = zeroes