Internal note: HEN-479

# The Nikhef/RU Nijmegen digitizer


## Abstract

This note describes the four channel low-power 200 Msps digitizer that was built to be used at the Pierre Auger Observatory to record signals of the radio antennas that detect cosmic rays. The digitizer uses fast ADCs, an FPGA, a GPS receiver and a small embedded PC board to convert, store and transmit data to the central data acquisition system. The functionality is explained, as well as the various internal registers and the structure of the messages that are sent by the embedded PC to and from the FPGA.


Authors: T.A.M. Wijnen, Radboud University Nijmegen.
H. Verkooijen, Nikhef Amsterdam.

Keywords: Fast digitizer, programmable digital oscilloscope, fast ADC, Pierre Auger, Nikhef Amsterdam, Radboud University Nijmegen, GPS, VOIPAC.

Note: HEN - 479
Version: 2.2.
Date: 17 December 2019.
URL: http://www.hef.ru.nl/∼thei/Auger/

# Document Change Record

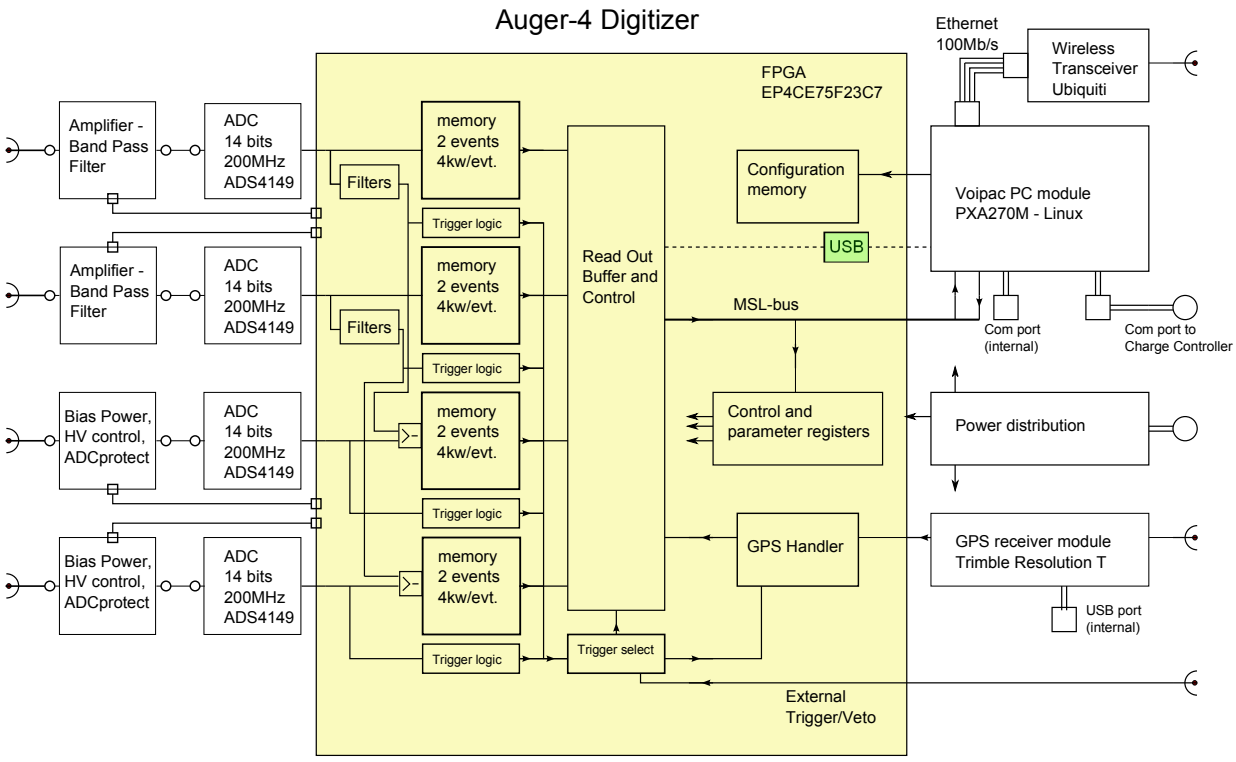
| Revision | Date | Reason for change |
|---|---|---|
| 1.00 | 22-May-2009 | First version digitizer (USB interface) |
| 1.2 | 28-Aug-2010 | Version 2 Digitizer (Embedded PC / ethernet interface) |
| 1.3 | 04-Feb-2011 | Version 3 Digitizer (new type of ADC) |
| 1.4 | 25-Aug-2011 | Several changes to the firmware of the FPGA |
| 2.0 | 24-Feb-2014 | Added a Linear Predictor filter in the trigger channels |
| 2.1 | 04-Jul-2014 | Added GPS UTC flags information |
| 2.2 | 17-Dec-2019 | Added option polarity selection of ADC signal for triggering |

# 1 Introduction

Nikhef Amsterdam and Radboud University Nijmegen have produced a four channel low-power rugged 200 Msps digital oscilloscope. The unique properties of this device are a fully programmable trigger logic, combined with remote readout and steering capabilities. On a trigger, the 4 input channels are digitized and time stamped by on board GPS module. The obtained relative time precision is less than 3ns. This makes it possible to merge the data of several digitizers within a certain small time window, no matter where these digitizers are located on the globe.

The communication to the digitizer uses standard ethernet, and uses the Voipac PXA270 Module as an interface between the ethernet communication and an FPGA which serves as the main readout steering module. The PXA270 Module takes an active part in the triggering and read out of the digitizer through internet. Moreover it can remotely handle the uploading of new firmware for the FPGA and software for the on board data acquisition system. The combination of an FPGA and a CPU creates an extremely flexible data acquisition module which is easily configurable by the user. Therefore, this module is an ideal unit in a research environment. The development of this unit is aimed at a distributed setup in Argentina, the purpose of which is to measure the radio signals between 30 and 80 MHz of cosmic-ray induced showers. This measurement requires a huge amplification of the measured signal (about 50 dB) as well as a smart noise reduction algorithm. Furthermore the location of the experiment, away from power lines at 1400 meter altitude, makes the use of solar power the only viable option, thus requiring a low power readout device. The digitizer includes a wireless transceiver which enables standalone field operation.

Even though this digital oscilloscope has been developed with a clear research project in mind, it is not hard to envision other solitary sensor readout projects requiring fast readout or data logging in which this oscilloscope on the internet could play a major role. The main specifications are: four channels 200 Msps, channel input bandwidth 100 MHz, input range +/- 100 mV tot-top, accuracy of 3 ns on GPS time stamp, power consumption 12V / 6 Watt, fully programmable trigger logic via FPGA firmware, fast reloading of firmware and fast readout via ethernet.

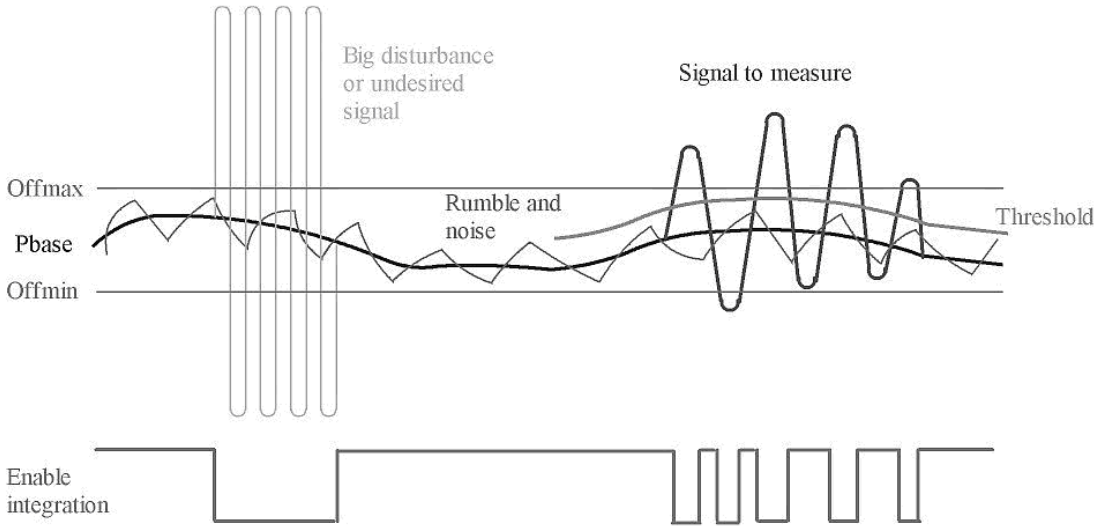

Auger-4 Digitizer
FPGA
EP4CE75F23C7
Amplifier - Band Pass Filter
Amplifier - Band Pass Filter
Bias Power, HV control, ADCprotect
Bias Power, HV control, ADCprotect
ADC 14 bits 200MHz ADS4149
ADC 14 bits 200MHz ADS4149
ADC 14 bits 200MHz ADS4149
ADC 14 bits 200MHz ADS4149
Filters
Filters
memory 2 events 4kw/evt.
memory 2 events 4kw/evt.
memory 2 events 4kw/evt.
memory 2 events 4kw/evt.
Trigger logic
Trigger logic
Trigger logic
Trigger logic
Read Out Buffer and Control
Trigger select
Configuration memory
USB
MSL-bus
Control and parameter registers
GPS Handler
Ethernet 100Mb/s
Wireless Transceiver Ubiquiti
Voipac PC module PXA270M - Linux
Com port (internal)
Com port to Charge Controller
Power distribution
GPS receiver module Trimble Resolution T
USB port (internal)
External Trigger/Veto

# 2 Digitizer operation

## 2.1 ADC offset, gain and baseline adjustment

The four ADC's of the four input channels can be individually adjusted in gain and offset by loading the respective parameters from the list commands. The default values for the gain correction is the center value 0x4000 (two bytes) and the default value for the offset adjust is the center value 0x80 (one byte). During the calibration sequence one can adjust these such that all four channels produce the same ADC value with zero Volt input and the same ADC value when an internal voltage ("full scale") is applied to the input amplifier.

The measured baseline of each channel can be integrated over a selectable time frame. This moving average will be subtracted from the ADC value before it is used in the trigger logic, i.e. the ADC input is compared with the threshold value for this channel. There are adjustable maximum and minimum ADC values that can be set to exclude big disturbances on the ADC inputs which would make a big error on the integrated baseline. The default for the maximum value ("basemax") is 0x2800 and the minimum value ("basemin")is 0x1800. These values represent limits of approximately 40 mV above and 40 mV below the baseline (0x2000) of the input signal. (See figure below.)



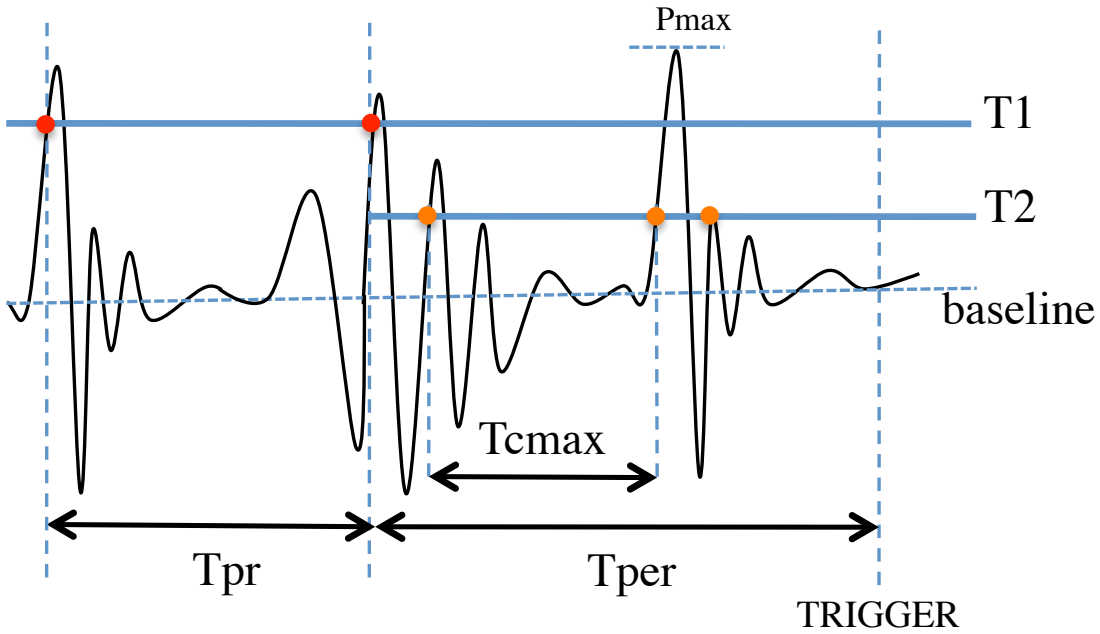

The integration time can be set with the "inttime" parameter. The value can be a number from 0 to 15. This number is the power of 2 and has to be multiplied with 10ns to roughly get the integration time. For example, a value of 9 sets the integration time to (2**9) * 10 ns = 5120 ns. There are exceptions for the values 0, 1 and 2. When the value is 0, the integration is turned off and the baseline is fixed to the center value of the ADC range, thus 0x2000. The parameter value 1 sets the integration time to 10 ns and the value 2 sets the integration time to 20 ns.

### 2.2 The channel trigger logic

In order to decide whether a signal from an antenna is interesting, the FPGA uses a set of parameters that are applied to the digitized signal. The figure below shows how these parameters are used to fire the channel trigger of the digitizer.

The orange dots show where the signal crosses the threshold levels. The number of positive going threshold crossings and the time in which they occur, have to meet several conditions as specified by the following parameters:

- Tprev (previous) is the time during which there are no T1-threshold (signal) crossings allowed. This time may be set to values ranging from 0 to 255 time 5 ns, so the maximum is 1275 ns.
- Tper (period) is the time during which the rising T2-threshold (noise) crossings are counted (in NC) starting from a valid T1-crossing. This time may be set to values ranging from 0 to 255 time 25 ns, so the maximum is 6375 ns.
- TCmax is the maximum time allowed between the T2-threshold crossings. This time may be set to values ranging from 0 to 255 time 5 ns, so the maximum is 1275 ns.
- NCmin is the minimum number of T2-threshold crossings. This number may be set to values ranging from 0 to 255.
- NCmax is the maximal number of T2-threshold crossings. This number may be set to values ranging from 0 to 255.




During Tper, the number of rising T2-threshold crossings is counted and the maximum pulse value (Pmax) is determined. If the signal has met the parameters above at the end of Tper, the number of T2-threshold crossings and maximum pulse value is known. If the number of T2-threshold crossings is higher than a certain minimum (NCmin) and

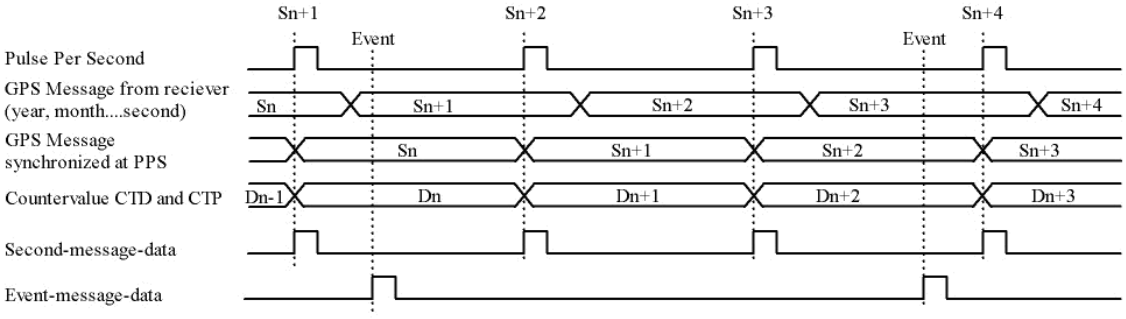
less than a certain maximum (NCmax), the number of T2-threshold crossings (NC) and maximum signal pulse value (Pmax) is latched for the next decision. Note that NC is set to one at the start of the period Tper.

After these requirements are met, the integrated baseline (Base) is subtracted from the maximum signal pulse value (Pmax) and divided by NC. The value of this division must also be higher than a minimum (Qmin) and less than a maximum (Qmax).

- Qmin is the minimum of (Pmax-Base)/NC. This number may be set to values from 0 to 255.
- Qmax is the maximum of (Pmax-Base)/NC. This number may be set to values from 0 to 255.

### 2.3 The calculation of the right event time

The FPGA generates "event message" data and "second message" data. The "event message" contains the sampled channel data accompanied by a GPS time stamp that will show the date and time (formatted as ddmmyyy hhmmss) when the event trigger occurred. To get the event time in nanosecond accuracy, we need a counter that records the time (in steps of approximately 5 ns because we use a 200 MHz clock) between two "one second pulses" (1PPS) of the GPS module. Each time that a 1PPS pulse arrives, the value of this counter is stored as the current CTP (Count Ticks between 1PPS signals) value. Each time that a trigger signal is set, the counter value is stored as the CTD value (Count Ticks Delta since last 1PPS signal). It is obvious that at the trigger time, one can only attach the previous CTP value to the data of this event trigger, because the counter is still running since the previous 1PPS pulse. The CTD value is also attached to the data of this event trigger. An example of the "data messages" and "second messages" from the FPGA to the PC is given here.

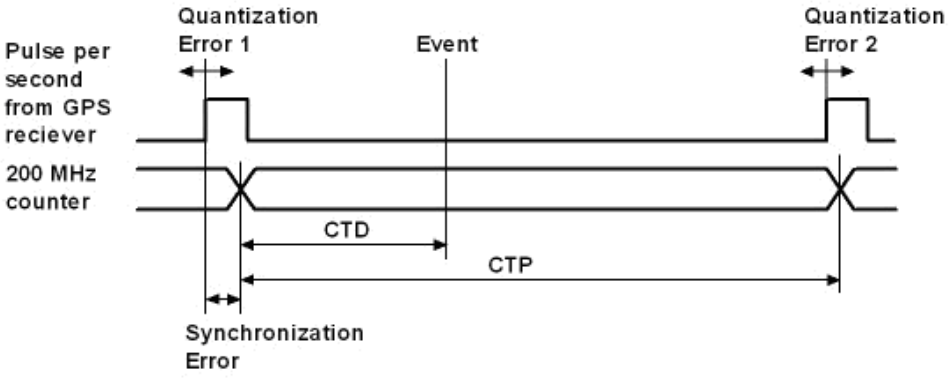



Successive data to PC

| Message toPC | Countervalue (200 MHz) | Second time label |
|---|---|---|
| Second-message-data | Endvalue CTP Dn-1 | Sn |
| Event-message-data | Momentary value CTD Dn | Sn |
| Second-message-data | Endvalue CTP Dn | Sn+1 |
| Second-message-data | Endvalue CTP Dn+1 | Sn+2 |
| Event-message-data | Momentary value CTD Dn+2 | Sn+2 |
| Second-message-data | Endvalue CTP Dn+2 | Sn+3 |

The CTP value belonging to the CTD value of trigger Sn, will be sent in the "second message" data of trigger Sn+1. The CTD/CTP ratio has to be multiplied with the "real"

second time expressed in nanoseconds. The time between two PPS signals (CTP) should be exactly one second, but unfortunately the PPS signals have a time error with respect to the "real" second time. These errors are called the quantization errors and can be read from the "second message" data. The Quantization Error 1 value for the "event message" data of trigger Sn is transferred in the "second message" data of trigger Sn+1. The Quantization Error 2 value is transferred in the "second message" data of trigger Sn+2.




$$\text{Event time [ns]} = \text{Synchr.Error} + \text{Quant.Error 1} + \left(\frac{\text{CTD}}{\text{CTP}}\right) * \left(1E^9 - \text{Quant.Error 1} + \text{Quant.Error 2}\right)$$

To calculate the exact event time we have to account for various errors in the time values. First the ratio between CTD and CTP has to be calculated. The ratio CTD/CTP is expressed in nanoseconds and it has to be added with a time offset, that is is the sum of the Quantization Error 1 and the Synchronization Error. The synchronisation the asynchroneous 1PPS signal with the 200 MHz clock, causes a synchronization uncertainty of one clock cycle, i.e. 5 ns. The sampling period is reduced to 2.5 ns by clocking the 1PPS signal on both the rising and falling edge of the 200 MHz system clock. The rising edge also clocks the CTP and CTD counter and it handles the 1PPS signal. The register bit that holds the falling edge result of the 1PPS is fed into bit 31 of the CTP register and it can be used to adjust the synchronization error by 2.5 ns.

### 2.4 The readout time windows

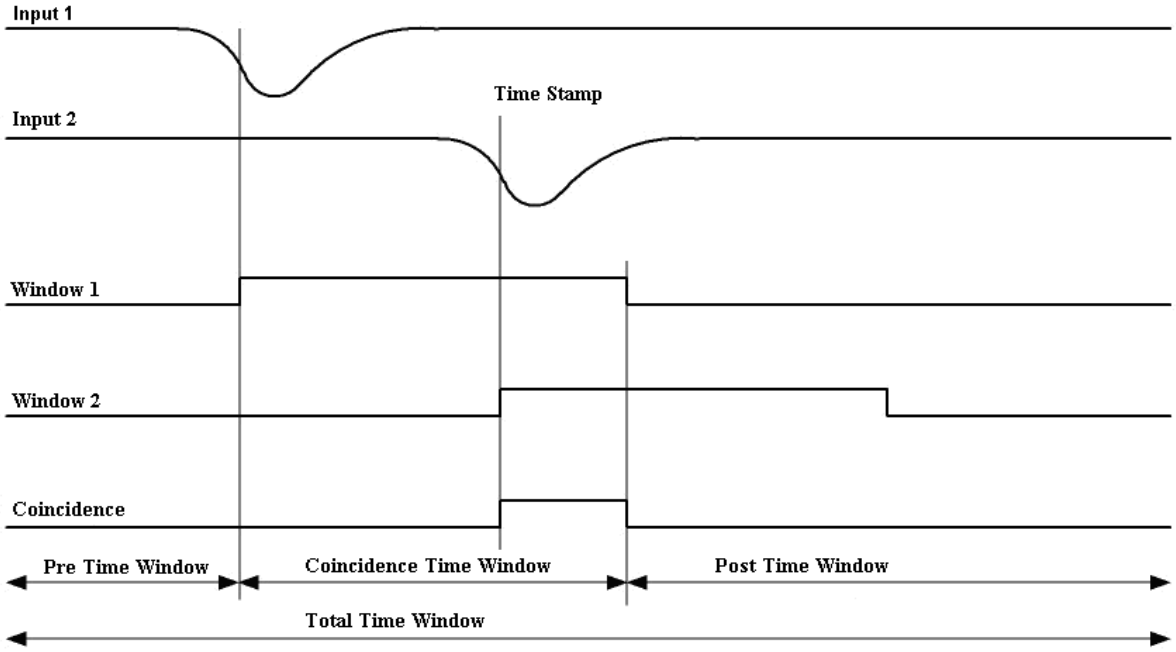
There are three time windows which control the sample of ADC values to store and read out of the channel memory. Concatenated they enclose the total time frame of the event data that is stored and read out.

- The Pre Coincidence time window. It can be set from 0 to 16000 ns (0 to 3200 steps of 5 ns, default 5000 ns) via the parameter with identifier 0x30.
- The Coincidence time window. It can be set from 0 to 20000 ns (0 to 4000 steps of 5 ns, default 5000 ns) via the parameter with identifier 0x31.
- The Post Coincidence time window. It can be set from 0 to 480 ns (0 to 96 steps of 5 ns, default 240 ns) via the parameter with identifier 0x32.

The total time (sum of the three windows) must be less than or equal to 20480 ns (4096 steps of 5 ns) because of the limits of the internal channel memory. The analog input is

sampled with a 12 bit amplitude resolution and the ADC values are stored in a 12 bit memory. The data will be sent to the computer in byte (8 bit) quantities. This means that the total number of data bytes is 1.5 times the number of ADC samples per channel. Therefore the number of data bytes is six times (4 channels x 1.5) the total window size. The maximum number of ADC bytes in a transfer is therefore 4 (channels) times 4096 (samples) times 3/2 bytes = 24576 bytes.

One parameter list specifies the four pairs of "pre-coincidence" and "post-coincidence" read out windows for the individual channel 1...4. The "coincidence" read out window is common for all channels.




Note that the actual event time of a trigger is latched. The event time value consists of the GPS time stamp and the CTP (Count Ticks between 1PPS) time.

# 3 Digitizer setup and messages from PC to FPGA

The digitizer is very flexible in the way that a trigger can be generated, that starts to store the ADC values in the channel memory. Several parameters can be loaded into the FPGA registers that control the function of the digitizer. Some may be used to calibrate the four individual analog input channels and their ADC's, others to select the various trigger modes and to adjust the time windows of the data to be read out. The parameter list table shows all parameters, some contain one byte, others have two bytes. The whole list of parameters can be loaded with a single message. Individual parameters can be loaded with a message formatted as follows:

| Header | Identifier | Byte Count | Message info | End |
|---|---|---|---|---|
| 0x99 | 0xxx | total length of message (2 bytes) | two or more data bytes | 0x66 0x66 |

The Identifier is a number that identifies the list of parameters that is used for the setup of several functions in the FPGA. List number 0 is for read only and on request the FPGA returns some status information. Lists number 1 and 2 are used for general settings, lists number 8...F for channel specific settings and lists number 10...1F for trigger channel filters. The table shows the various lists and the number of data bytes in that list. The total number of bytes of the list is that number plus 6 for the header, identifier, the byte count and the end marker.

### 3.1 FPGA version and status (list 0x00)

This is a read only list and nothing can be written to this list except the request to read the list.

| List name | List number | bytes |
|---|---|---|
| Request FPGA Version and Status | 0x00 | 0 |
| Digitizer Mode Parameters | 0x01 | 12 |
| Readout Window Parameters | 0x02 | 16 |
| Command and Reset Parameters | 0x03 | 8 |
| SPI Sequence Parameters | 0x04 | 8 |
| Channel-1 Property Parameters | 0x08 | 12 |
| Channel-2 Property Parameters | 0x09 | 12 |
| Channel-3 Property Parameters | 0x0A | 12 |
| Channel-4 Property Parameters | 0x0B | 12 |
| Channel-1 Trigger Parameters | 0x0C | 12 |
| Channel-2 Trigger Parameters | 0x0D | 12 |
| Channel-3 Trigger Parameters | 0x0E | 12 |
| Channel-4 Trigger Parameters | 0x0F | 12 |
| Filter-1 Constants, Channel-1 | 0x10 | 16 |
| Filter-1 Constants, Channel-2 | 0x11 | 16 |
| Filter-2 Constants, Channel-1 | 0x12 | 16 |
| Filter-2 Constants, Channel-2 | 0x13 | 16 |
| Filter-3 Constants, Channel-1 | 0x14 | 16 |
| Filter-3 Constants, Channel-2 | 0x15 | 16 |
| Filter-4 Constants, Channel-1 | 0x16 | 16 |
| Filter-4 Constants, Channel-2 | 0x17 | 16 |
| Filter-5 Coefficents-1, Channel-1 | 0x18 | 16 |
| Filter-5 Coefficents-2, Channel-1 | 0x19 | 16 |
| Filter-5 Coefficents-3, Channel-1 | 0x1A | 16 |
| Filter-5 Coefficents-4, Channel-1 | 0x1B | 16 |
| Filter-5 Coefficents-1, Channel-2 | 0x1C | 16 |
| Filter-5 Coefficents-2, Channel-2 | 0x1D | 16 |
| Filter-5 Coefficents-3, Channel-2 | 0x1E | 16 |
| Filter-5 Coefficents-4, Channel-2 | 0x1F | 16 |

### 3.2 Digitizer mode parameters (list 0x01)

This list is used for general parameters that set the mode of operation, select trigger sources and preset the common coincidence read out time window.

| value | description | bytes |
|---|---|---|
| 0x0199 | header (99) and list number (01) | 2 |
| 0x0012 | byte count (includes all bytes from header upto end marker) | 2 |
| 0x0100 | Control register | 2 |
| 0x0030 | Trigger Enable Mask | 2 |
| 0x00 | Channel Read out enable mask (4 bits) | 1 |
| 0x00 | Test trigger rate divider | 1 |
| 0x0030 | Common Coincidence read out time | 2 |
| 0x0100 | spare-1 | 2 |
| 0x0030 | spare-2 | 2 |
| 0x6666 | end of list marker | 2 |

#### 3.2.1 Control register

The general control register enables the main functions in the FPGA.

| bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|
| Enable auto reboot (timeout) | unused | unused | Enable Filter nr.5 | Enable Filter nr.4 | Enable Filter nr.3 | Enable Filter nr.2 | Enable Filter nr.1 |
| **bit 7** | **bit 6** | **bit 5** | **bit 4** | **bit 3** | **bit 2** | **bit 1** | **bit 0** |
| DCO edge select | Fake ADC values | GPS program mode | unused | Filter Read out enable | Full Scale enable | 1PPS enable | DAQ enable |

- When data bit 0 of the control register is '0', all triggers are disabled and no measurement data will be sent to the PC. Set this bit to '1' when the PC is ready to accept data.
- With bit 1 of the control register, the 1PPS message is enabled to be sent to the PC. On default (after power up or after a soft reset), this bit is off and no 1PPS messages are sent to the PC. This ensures that no messages are sent while the PC is still initializing. Upon power up or a soft reset, the FPGA keeps the interface signals to the PC in an 'off' state for approximately 10 seconds.
- When bit 2 of the control register is set, the inputs of the ADC's are forced to the Full Scale value. This mode may be used during the calibration procedure of the ADC's.
- When bit 3 is on, the data read out on channels 3 and 4 is the filtered data of channels 1 and 2, otherwise the input of channel 3 and 4 is read out.
- With bit 5 of the control register, the programming of the GPS receiver can be enabled. After power up, bit 5 is turned off ('0') and it is not possible to program the GPS receiver.

- With bit 6 of the control register turned on, the ADC input register is put in a test mode in which it will output a counting pattern (in stead of real ADC values). The counting pattern is stored as fake data and may be used for test purposes.
- When bit 7 of the control register is set, the FPGA will use the rising edge of ADC-DCO to store the ADC values, otherwise the FPGA will use the falling edge of ADC-DCO. The default value for this bit is off.
- When bit 8, 9, 10 or 11 of the control register is set, the IIR filter number 1, 2, 3 or 4 for the ADC channels 1 and 2 is turned on, before this data is fed into the trigger logic. The data read out from channel 1 and 2 is the raw ADC data. For test purposes, the filtered data can be read out via channel 3 and 4 in case bit 3 is turned on (see above).
- When bit 12 of the control register is set, the Linear Predictor (LP) filter for the ADC channels 1 and 2 is turned on, before this data is fed into the trigger logic. This filter is placed in series with and behind the IIR filters.
- Bit 15 enables the auto reboot feature that will reboot the PC if no triggers have been accepted for readout for more than 5 minutes.

### 3.2.2 Trigger enable mask

There is a parameter that should be used to enable all kinds of trigger sources, as shown in the table below. Bits 11 thru 8 enable the trigger logic for the input channels 1 thru 4. Note that a channel can not generate a trigger when it is not enabled for read out. Bit 7 enables a trigger when both input channel 1 and 2 cross the trigger threshold.
The Calibration trigger (bit 6) is generated by a very slow internal pulse generator ($\sim$ 9.4 Hz) and it is used by the software to get uncorrelated event triggers for the adjustment of the offset and gain settings for the individual ADC channel inputs. The “10 seconds” trigger (enabled by bit 5), generated once per 10 seconds by the 1PPS pulse from the GPS module, provides an uncorrelated event trigger that can be used to sample background noise on the input channels. The External trigger enable (bit 4) enables both the external input and the internal trigger pulse generator. The frequency of the internal pulse generator can be set via the Test Trigger Rate Divider to values of 0 Hz ... 2400 Hz.

| bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|
| unused | unused | unused | unused | Enable Channel 4 trigger | Enable Channel 3 trigger | Enable Channel 2 trigger | Enable Channel 1 trigger |
| **bit 7** | **bit 6** | **bit 5** | **bit 4** | **bit 3** | **bit 2** | **bit 1** | **bit 0** |
| Enable Ch.1 and Ch.2 trigger | Enable Calibration trigger | Enable 10 seconds trigger | Enable External trigger | unused | Enable Not Ch.1 and Ch.2 trigger | Enable Red. Ch1 and Ch.2 trigger | Enable Ch.3 and Ch.4 trigger |

Bit 2 enables the capability to trigger on the logical combination of ”NOT Channel 1 AND Channel 2”. Bit 1 enables the trigger on the condition that both Channel 1 AND Channel 2 fulfill the trigger requirement AND that the maximum signal on Channel 2 (E/W) must be higher then the maximum signal on Channel 1 (N/S).

### 3.2.3 Channel read out enable

The channel readout enable parameter should be used to select which channels are to be read out and thus which channels can be activated in the trigger logic. With bits 0 thru 3 of this parameter the read out of input channels 1 thru 4 are enabled for data read out.

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|
| unused | unused | unused | unused | Channel 4 read out enable | Channel 3 read out enable | Channel 2 read out enable | Channel 1 read out enable |

### 3.2.4 Test trigger rate divider

Once the external/internal trigger enable bit is set, a non-zero value loaded into the internal test trigger rate divider register, will generate random trigger at a predefined rate of 4800 / (N+1) Hz. When a value of zero is used for N, the internal triggers are stopped. The following table shows the frequency for a set of rate divider numbers.

| Rate Divider Register | Test Trigger Frequency (Hz) |
|---|---|
| 0x00 | 0 |
| 0x01 | 2400 |
| 0x02 | 1600 |
| 0x03 | 1200 |
| 0x04 | 960 |
| 0x05 | 800 |
| 0x07 | 600 |
| 0x0B | 400 |
| 0x17 | 200 |
| 0x2F | 100 |
| ... | ... |
| 0x5F | 50 |
| 0xBF | 25 |
| 0xFF | 18 |

### 3.2.5 Parameter for common coincidence read out time window

The Coincidence readot time window can be set from 0 to 20480 ns (0 to 4096 steps of 5 ns) via this parameter. The default is 5000 ns.

### 3.2.6 Parameters spare-1 and spare-2

There are two spare registers in this list and its function is not yet defined.

### 3.3 Digitizer read out time parameters (list 0x02)

The Pre Coincidenc read out time window and the Post Coincidenc read out time window can be set for each individual channel, while the Coincidence read out time window is common for all 4 channels. The windows can be set in units of 5 ns using the following list.

| value | description | bytes |
|---|---|---|
| 0x0299 | header (99) and list number (02) | 2 |
| 0x0016 | byte count (includes all bytes from header upto end marker) | 2 |
| 0x0100 | Pre coincidence read out time for Channel 1 (default: | 2 |
| 0x0030 | Post coincidence read out time for Channel 1 | 2 |
| 0x0100 | Pre coincidence read out time for Channel 2 | 2 |
| 0x0030 | Post coincidence read out time for Channel 2 | 2 |
| 0x0100 | Pre coincidence read out time for Channel 3 | 2 |
| 0x0030 | Post coincidence read out time for Channel 3 | 2 |
| 0x0100 | Pre coincidence read out time for Channel 4 | 2 |
| 0x0030 | Post coincidence read out time for Channel 4 | 2 |
| 0x6666 | end of list marker | 2 |

### 3.4 Command and Reset list (list 0x03)

The special Command/Reset list is written as list number 3 and the two data bytes are used for the Command register and the Reset register. A zero byte means no command or no reset signal.

| Header | Identifier | Byte Count | Message info | End |
|---|---|---|---|---|
| 0x99 | 0x03 | 0x0008 | command register, reset register | 0x66 0x66 |

#### 3.4.1 Command register

The Command register is written as the first byte in this list and an individual signal is pulsed on each data bit that is set in this register. In this implementation only bit 0 of the data byte is defined and it generates a single trigger pulse (width 400 ns) which is active only if the internal trigger is enabled. Other bits result in no operation.

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | Single Trigger |

#### 3.4.2 Reset register

The Reset register is written as the second byte in this list and an individual reset signal is pulsed when data bit 1 or 0 in this register is set. The Soft Reset will reset the whole FPGA and load the default values in all the parameter registers. The Hard Reset

will reset/reboot the PC. During the boot process, the FPGA will be re-configured and restarted and this will also load default values into all the parameter registers.

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Reload FilterCoeff. | 0 | 0 | Hard Reset | Soft Reset |

### 3.5 Digitizer SPI sequence list (list 0x04)

The ADC (AD9230) has an additional number of specific internal registers that are part of the SPI register map and these can be loaded using the SPI command registers. The parameters of the SPI command hold the SPI 8 bit address number (first data byte) and the SPI 8 bit data value (second data byte). This list is written as shown below.

| Header | Identifier | Byte Count | Message info | End |
|---|---|---|---|---|
| 0x99 | 0x04 | 0x0008 | SPI address, SPI data | 0x66 0x66 |

For further details of how to use the SPI commands, see the ADC 9230 documentation.

### 3.6 Channel property parameters (lists 0x08...0x0B)

The properties (e.g. offset and gain correction) of the four ADC channels can be set using lists 0x08...0x0B as shown below. The Channel Property list contains the following bytes:

| value | description | bytes |
|---|---|---|
| 0x0899 | header (99) and list number (8...B) | 2 |
| 0x0012 | byte count (includes all bytes from header upto end marker) | 2 |
| 0x4000 | gain correction | 2 |
| 0x0080 | offset correction (80), integration time (00) | 2 |
| 0x2800 | base maximum | 2 |
| 0x1800 | base maximum | 2 |
| 0x00 | PM voltage setting | 1 |
| 0x00 | Filter setting (unused) | 1 |
| 0x0000 | spare-1 (unused) | 2 |
| 0x6666 | end of list marker | 2 |

The four ADC's of the four input channels can be individually adjusted in gain and offset by loading the respective parameters in these lists. They are used to ensure that an ADC input signal of 0 Volt produces a zero value for storage in the event memory. When a test full scale signal is applied to the ADC (via the Control Register), one should apply a gain correction value such that all channels produce the same ADC value. The correction values can then be used during normal operation.

For the baseline adjustement, the respective parameters set the maximum and minimum ADC values to exclude big disturbances on the ADC inputs and the integration time for the moving average.

The PM voltage setting parameter is used for adjusting the high voltage supply for the Photo Multiplier tube that may be used as an input to the digitizer. The output voltage of the corresponding DAC is 300 mV + n * 5 mV. The Filter setting parameter can be used to select a filter on the input of this ADC channel. It is not yet used in the current version of the digitizer. The parameter spare-1 is not yet used.

## 3.7 Channel trigger parameters (lists 0x0C... 0x0F)

The Channel Trigger parameter list contains the following bytes:

| value | description | bytes |
|---|---|---|
| 0x0C99 | header (99) and list number (C... F) | 2 |
| 0x0012 | byte count (includes all bytes from header upto end marker) | 2 |
| 0x0190 | signal threshold (T1) | 2 |
| 0x00A0 | noise threshold (T2) | 2 |
| 0xC8 | tprev, previous time (n*5 ns) | 1 |
| 0x50 | tper, period time (n*25 ns) | 1 |
| 0x64 | tcmax, maximum time between threshold crossings (n*5ns) | 1 |
| 0xFF | ncmax, maximum number of threshold crossings | 1 |
| 0x00 | ncmin, minimum number of threshold crossings | 1 |
| 0xFF | qmax, maximum charge | 1 |
| 0x00 | qmin, minimum charge | 1 |
| 0x00 | options, (spare-3) | 1 |
| 0x6666 | end of list marker | 2 |

### 3.7.1 Parameters for trigger thresholds (signal and noise)

The threshold levels for the trigger logic can be set individually per channel and there are separate thresholds for the signal (T1 threshold) and the noise background (T2 threshold) for use in the trigger logic.

### 3.7.2 Parameters for trigger logic

- The parameter Tprev is the time during which there are no T1-threshold crossings allowed. The time = n * 5 ns, n=0... 255, so the time=0... 1275 ns.
- The parameter Tper is the time during which the rising T2-threshold crossings are counted (in NC), starting from a valid T1-crossing.
  The time = n * 25 ns, n=0... 255, so the time=0... 6375 ns.
- The parameter TCmax is the maximum time allowed between the T2-threshold crossings. The time = n * 5 ns, n=0... 255, so the time=0... 1275 ns.
- The parameter NCmin is the required minimum number of T2-threshold crossings (0... 255).
- The parameter NCmax is the required maximum number of T2-threshold crossings (0... 255).

- During Tper, the number of rising T2-threshold crossings is counted and the maximum pulse value (Pmax) is determined. The parameter Qmax is the required maximum of Pmax/NC and Qmin is the required minimum of Pmax/NC. Both can be set in the range 0...255.
- The parameter Options (spare-3) allows definitions of extra trigger options:
  bit 0: if 0, trigger on a positive ADC signal; if 1: trigger on negative ADC signal.
  other bits: unused.

### 3.7.3 Parameters for filter settings

The Filter Constants for the filters in the trigger channel logic, can be set for each channel separately using one of the list with identifier 0x10 ... 0x1F. The lists contain the following bytes.

| value | description | bytes |
|---|---|---|
| 0x1099 | header (99) and list number (10 ... 1F) | 2 |
| 0x0016 | byte count (includes all bytes from header upto end marker) | 2 |
| .. .. | filter constants for a specific frequency band | 16 |
| 0x6666 | end of list marker | 2 |

**Filter constants**

The (hex) values of the filter constants (A1,A2,B1...B6) for an IIR filter with central frequency and width are given in the table below:

| Freq | W. | A1 | | A2 | | B1 | | B2 | | B3 | | B4 | | B5 | | B6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MHz | byte: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 67.3 | 0.99 | 0B | A4 | 0C | 4F | 04 | 23 | 04 | 00 | 0B | E7 | 03 | EC | 0C | 5A | 03 | D8 |
| 39.6 | 0.99 | 02 | 01 | 0C | 4F | 0D | 6F | 04 | 00 | 02 | 8B | 03 | EC | 09 | C6 | 03 | D8 |
| | | | | | | | | | | | | | | | | | |

## 3.8 Get parameter list command (list 0xnn)

In order to read a parameter list back from the FPGA, one first has to send a request for that list, using a list command without any data bytes. The list consists of the header, the list identifier number, the byte count (fixed to be 6) and the end marker.

| Header | Identifier | Byte Count | End |
|---|---|---|---|
| 0x99 | 0xnn | 0x0006 | 0x66 0x66 |

The FPGA answers to this command by sending the list in the same format as it was written into the FPGA. This applies for lists 0x01...0x1F. List number 0x00 is special because it is read only and the get command returns a long list of FPGA version and status info, including the GPS data from the GPS receiver module. See the next chapter for a full description of bytes that are returned by the FPGA.

# 4 Digitizer read out and messages from FPGA to PC

The structure for the messages from the FPGA to the PC is the same as for messages from the PC to the FPGA. The message starts with a header (0x99), followed by an identifier for the type of the message. Next comes a byte count, followed by the data bytes and the message is terminated by an end marker (0x6666). Note that the Byte Count is equal to the total number of bytes including header, identifier, byte count, data and end marker.

| Header | Identifier | Byte Count | Message info | End |
|---|---|---|---|---|
| 0x99 | 0xxx | total length of message (2 bytes) | two or more data bytes | 0x66 0x66 |

The number of data bytes depends on the type. In case of a parameter list or a time stamp read out, it will be a few bytes and in case of measurement data it can be a few ten thousand bytes. The identifier byte gives the type of the message. the following paragraphs give detailed information about the contents of these messages.

## 4.1 Read Parameter list message

After a request by sending the "get parameter list" command, the FPGA will answer by sending the requested list in the same format as it was written into the FPGA. This applies for lists 0x01...0x1F and for details, please refer to the previous chapters. List number 0x00 is special because it is read only and the get command returns a long list of FPGA version and status info, including the GPS data from the GPS receiver module.

### 4.1.1 List of read only registers (list 0x00)

| Identifier | Description | bytes |
|---|---|---|
| 0x0099 | Header (99) and list number(0) | 2 |
| 0x002E | byte count (46 bytes) | 2 |
| .. | FPGA version register | 4 |
| .. .. | GPS time stamp (date and time) | 7 |
| .. | Status electronics | 1 |
| .. | GPS position longitude in rad. (1 double) | 8 |
| .. | GPS position latitude in rad. (1 double) | 8 |
| .. | GPS position altitude in meter (1 double) | 8 |
| .. | Temperature electronics in degrees (1 float) | 4 |
| 0x6666 | end of list marker | 2 |

The GPS information in this list is a copy of the data that is received from the GPS receiver. The FPGA version register has a length of four bytes and consists of two parts: a software and a hardware part. The eight highest bits (31 ... 24) are always logical zero. The bits 23 ... 12 represent a 12 bits firmware version number of the code of the field programmable gate array (FPGA). Bits 11 ... 9 denote a 3 bit subversion number. The nine lower bits (8 ... 0) represent a hardware serial number. Each unit has its own serial number / address and it is set by jumpers on the board.

**FPGA version register**

| bit 31 . . . 24 | bit 23 . . . 12 | bit 11 . . . 9 | bit 8 . . . 0 |
|---|---|---|---|
| always "00000000" (8 bit) | firmware version (12 bit) | subversion number (3 bit) | serial number (9 bit) |

### 4.1.2 Firmware versions

| version | subversion | date | info |
|---|---|---|---|
| 0x070 | 0 | 20121101 | first version for Digitizer 4 |
| 0x070 | 1 | 20121101 | first version for Digitizer 4 |
| 0x070 | 2 | 20121101 | first version for Digitizer 4 |
| 0x070 | 5 | 2013xxxx | separate IIR for for channel 1 and 2. |
| 0x070 | 6 | 2013xxxx | trigger enables for "NOT Channel 1 AND Channel 2" and "channel 1 and 2 and power of channel 2(E/W) higher than power of channel 1 (N/S). |
| 0x070 | 7 | 20130927 | added auto-reboot, corrected UTC byte order |
| 0x071 | 0 | 20140312 | Linear Predictor filter in trigger channels |
| 0x071 | 1 | 20140611 | optional generate of Linear Predictor filter |
| 0x071 | 2 | 20140924 | added BulletOff bit for BulletM5 wifi transceiver |
| 0x071 | 3 | 20141010 | default auto-reboot off; original (HV) BlockCoinc code |
| 0x071 | 4 | 20141103 | synchronized BlockCoinc to proper clock, removed BulletOff |
| 0x071 | 5 | 20150211 | corrected error in FifoSelect (BlockCoinc)<br>trigger on negative ADC signal for scintillators (Auger) |
| 0x071 | 6 | 20190817 | trigger on positive ADC signal for power adapter (Grand) |
| 0x071 | 7 | 20190821 | selectable (via parameters) trigger on positive or negative ADC signal |
| 0x072 | 0 | 20191217 | Corrected bug in firlp (no trigger in ch.1/2) and posttime handling |

## 4.2 One Second message (1PPS)

The "One Second" message is a message that will be sent to the PC after the reception of the time stamp message from the GPS receiver. The time stamp message from the GPS receiver will be renewed every second and that information along with a copy of all parameter lists will be sent to the PC. The format of the GPS time/date stamp is shown below. The CTP (Count Ticks between 1PPS) is a counter value that represents the number of clock periods of the 200 MHz clock between two 1PPS signals. The counter will be set to one on a 1PPS signal and it counts (in 5 ns steps) until the next 1PPS signal. On a 1PPS signal the counter value is stored and the counter is set to one again. The highest bit of CTP (bit31) is set when there is a synchronization error. The Quantization Error represents the PPS quantization error in units of seconds. (See GPS receiver documentation for more information.) The GPS Time stamp has the following format.

### GPS Time Stamp and Status register

| example value | description (for 27-jan-2013 12:34:56) | bytes |
|---|---|---|
| 0x07dd | Year (from GPS receiver) | 2 |
| 0x01 | Month (from GPS receiver) | 1 |
| 0x1B | Day (from GPS receiver) | 1 |
| 0x0C | Hours (from GPS receiver) | 1 |
| 0x22 | Minutes (from GPS receiver) | 1 |
| 0x38 | Seconds (from GPS receiver) | 1 |
| 0x01 | Status of electronics | 1 |

At the moment only two bits of the status register are implemented while the others are zero. If bit 0 is set, a GPS receiver module has been detected. If bit 1 is set, the trigger is temporarily inhibited because both front end memory buffers are filled. As soon as one buffer has been read out, the flag is automatically cleared.

### GPS flags

The 1PPS message contains 4 additional bytes that provide status and flags from the GPS receiver.

| name | bytes | info |
|---|---|---|
| UTC Offset | 2 | leap second offset between GPS and UTC |
| GPS Timing flags | 1 | details: see Resolution-T GPS user guide pages 77... 81 |
| GPS Decoding flags | 1 | details: see Resolution-T GPS user guide pages 77... 81 |

The GPS Timing flags are defined as:

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|
| n.u. | n.u. | n.u. | 0: time from GPS | 0: have UTC info | 0: time is set | 0: GPS PPS | 0: GPS time |
| n.u. | n.u. | n.u. | 1: time from user | 1: no UTC info | 1: time is not set | 1: UTC PPS | 1: UTC time |

The UTC Offset (signed integer value) is the current leap second offset between GPS and UTC according to the relationship: Time(UTC) = Time(GPS - UTC Offset. The UTC Offset information is reported to resolution-T by the GPS system and can take up to 12.5 minutes to obtain. More details: see Resolution-T GPS user guide pages 77... 81.

The GPS Decoding status byte may have one of the following values:

- 0x00 : doing fixes
- 0x01 : do not have GPS time
- 0x03 : PDOP (Position Dilution of Position error) is too high
- 0x08 : no usable satellites
- 0x09 : only 1 usable satellite
- 0x0A : only 2 usable satellites
- 0x0B : only 3 usable satellites
- 0x0C : the chosen satellite is unusable
- 0x10 : TRAIM (TimingReceiverAutonomousIntegrityMonitor) rejected the fix

**Status of electronics register**

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | Trigger inhibited | GPS detected |

Note that all parameter lists are read out, but only the actual parameters appear in the list, while their respective headers and end marker are omitted. List number 0 is a special list and its contents will be shown in the next paragraph.

**One second message (1PPS) structure**

| value | description | bytes |
|---|---|---|
| 0xC499 | header (99) and identifier (C4) | 2 |
| 0x0xxx | byte count (includes all bytes from header upto end marker) | 2 |
| .. .. | GPS Time Stamp and Status | 8 |
| .. | CTP (tick count between 1PPS) | 4 |
| .. | GPS Quantization error | 4 |
| .. | GPS UTC Offset | 2 |
| .. | GPS UTC Timing flags | 1 |
| .. | GPS Decoding Status | 1 |
| .. | Trigger Rate during last second | 2 |
| .. .. | list #0x00: FPGA Version and Status | 40 |
| .. .. | list #0x01: Digitizer Mode Parameters | 12 |
| .. .. | list #0x02: Readout Window Parameters | 16 |
| .. .. | list #0x08: Channel-1 Property Parameters | 12 |
| .. .. | list #0x09: Channel-2 Property Parameters | 12 |
| .. .. | list #0x0A: Channel-3 Property Parameters | 12 |
| .. .. | list #0x0B: Channel-4 Property Parameters | 12 |
| .. .. | list #0x0C: Channel-1 Trigger Parameters | 12 |
| .. .. | list #0x0D: Channel-2 Trigger Parameters | 12 |
| .. .. | list #0x0E: Channel-3 Trigger Parameters | 12 |
| .. .. | list #0x0F: Channel-4 Trigger Parameters | 12 |
| .. .. | list #0x10: Filter-1, Channel-1 Constants | 16 |
| .. .. | list #0x11: Filter-1, Channel-2 Constants | 16 |
| .. .. | list #0x12: Filter-2, Channel-1 Constants | 16 |
| .. .. | list #0x13: Filter-2, Channel-2 Constants | 16 |
| .. .. | list #0x14: Filter-3, Channel-1 Constants | 16 |
| .. .. | list #0x15: Filter-3, Channel-2 Constants | 16 |
| .. .. | list #0x16: Filter-4, Channel-1 Constants | 16 |
| .. .. | list #0x17: Filter-4, Channel-2 Constants | 16 |
| .. .. | list #0x18: Filter-5, Channel-1 Coefficients-1 | 16 |
| .. .. | list #0x19: Filter-5, Channel-1 Coefficients-2 | 16 |
| .. .. | list #0x1A: Filter-5, Channel-1 Coefficients-3 | 16 |
| .. .. | list #0x1B: Filter-5, Channel-1 Coefficients-4 | 16 |
| .. .. | list #0x1C: Filter-5, Channel-2 Coefficients-1 | 16 |
| .. .. | list #0x1D: Filter-5, Channel-2 Coefficients-2 | 16 |
| .. .. | list #0x1E: Filter-5, Channel-2 Coefficients-3 | 16 |
| .. .. | list #0x1F: Filter-5, Channel-2 Coefficients-4 | 16 |
| 0x6666 | end of list marker | 2 |

The byte count for the firmware is 0x0140 when filter-1 thru filter-4 are implemented; the byte count is 0x1C0 when filter-5 (Linear Predictor) is also implemented.

## 4.3 ADC Measurement data message

On a coincidence of input signals or upon reception of some other trigger signal, the ADC measurement data will be sent to the computer using the format shown below. The ADC produces 14 bit offset binary values which are converted to signed 14 bit values and this value is sign extended into a 16 bit number. The ADC values are thus transferred as signed short integer values of 2 bytes. The actual Byte Count depends on the number of channels and the size of the read out time windows. The maximum value for the number of ADC data bytes is 4 * 4096 * 2 = 32768 bytes. The maximum of the total message length is then 24 + 32768 = 24600 bytes.

### ADC Measurement data message structure

| value | description | bytes |
|---|---|---|
| 0xC099 | header (99) and identifier C0 | 2 |
| 0x0016 | byte count (includes all bytes from header upto end marker) | 2 |
| .. | Trigger Source pattern | 2 |
| .. .. | GPS Time Stamp and Status | 8 |
| .. | CTD : tick count from 1PPS to trigger (units of 5 ns) | 4 |
| .. | Total time or # of samples read out of Channel 1 | 2 |
| .. | Total time or # of samples read out of Channel 2 | 2 |
| .. | Total time or # of samples read out of Channel 3 | 2 |
| .. | Total time or # of samples read out of Channel 4 | 2 |
| .. | Signal Threshold of Channel 1 (T1) | 2 |
| .. | Noise Threshold of Channel 1 (T2) | 2 |
| .. | Signal Threshold of Channel 2 (T1) | 2 |
| .. | Noise Threshold of Channel 2 (T2) | 2 |
| .. | Signal Threshold of Channel 3 (T1) | 2 |
| .. | Noise Threshold of Channel 3 (T2) | 2 |
| .. | Signal Threshold of Channel 4 (T1) | 2 |
| .. | Noise Threshold of Channel 4 (T2) | 2 |
| .. .. | list #0x01: Digitizer Mode Parameters | 12 |
| .. .. | list #0x02: Readout Window Parameters | 16 |
| .. .. | ADC data Channel 1, 2, 3, 4 | .. |
| 0x6666 | end of list marker | 2 |

### Trigger Source pattern

| bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|
| 0 unused | 0 unused | 0 unused | 0 unused | Channel 4 Trigger | Channel 3 Trigger | Channel 2 Trigger | Channel 1 Trigger |
| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
| Ch1 AND Ch2 trigger | Calibration trigger | 10 sec. trigger | External trigger | Power (Ch1 + Ch2) trigger | 0 unused | 0 unused | 0 unused |

The Trigger Source pattern shows a bit pattern of the signals on which the final trigger or coincidence was made.

- Channel triggers. A trigger from an ADC channel can only make a coincidence if that ADC channel was enabled in the trigger mask and in the readout and if the analog signal meets the specified requirements. These 4 bits are read from the Channel trigger and read out enable register.
- External trigger. The external trigger is made by a rising edge of a digital signal from the external input. The bit is also set by internally generated triggers (by command or a free running trigger).
- The 10sec trigger. There is an internal counter which counts the 1PPS pulses from the GPS module. A trigger is generated each time the counter has counted ten 1PPS pulses (random time trigger every ten seconds).
- Calibration trigger. There is an internal counter from which a trigger is generated every 0.106 seconds. This trigger is needed for calibration purposes.
- Ch1 AND Ch2. This is a logical AND of the triggers from channel 1 and channel 2.
- Power of (Ch1 + Ch2). This is a trigger based on the power of the signals in channel 1 and channel 2.

The structure of the GPS time stamp within this message is the same as in the 1PPS message. The CTD gives you the measurement of the time between the preceeding 1PPS signal and the moment of this event trigger, measured with the 200 MHz clock (units of 5 ns). The combination of the total time per channel (units of 5 ns), the common coincidence read out time (from list #1) and the respective pre and post coincidence read out time, will determine which ADC samples belong to each of the readout channels. List #0x01 contains the channel read out enable pattern which tells you which channels have been read out. List #0x01 also contains the trigger enable mask which shows the various trigger types that were enabled.

### 4.4 Communication error message

If the FPGA receives a message that is does not understand, it will reply to the PC with a message with identifier 0xCE followed by the identifier of the misunderstood message. If the header byte is not detected, the first data byte will be 0x99. If an identifier of a non existing message is detected, the first data byte will be 0x89. If the end marker is not detected, the first data byte will be 0x66.

| Identifier | Description | bytes |
|---|---|---|
| 0xCE99 | Header (99) and identifier (CE) | 2 |
| 0x0008 | Byte Count | 2 |
| .. | data byte 1 | 1 |
| 0x00 | data byte 2 (always zero) | 1 |
| 0x6666 | end of list marker | 2 |

### 4.5 Special test messages

For test purposes and in case of lost byte sychronisation, the PC may send the following bytes to make sure that the FPGA returns to a known state.

- PC sends 0x66 0x66 ... : the FPGA will absorb these bytes, give no response and reset the internal read state machine.
- PC sends 0x99 0x66 ... : the FPGA will respond with the same bytes (echo 0x99 0x66) and reset the internal read state machine.

## 5 LEDs on the digitizer boards

There are several LEDs mounted on the digitizer board that display information about the status of the electronics. The table lists their names in several design files.

| function | Auger-4 LEDs | | |
|---|---|---|---|
| | Board LED# | FPGA name | Ease Schematics |
| GPS Detected | D2 | LED(2) | GPS_DET |
| One Sec Pulse (1PPS) | D3 | LED(3) | ONESEC |
| Ten Sec Trigger Pulse | D1 | LED(1) | TENSEC |
| Ext. or Int. Trigger | D8 | LED(8) | EXT_TR_OUT |
| Coincidence signal | D5 | LED(5) | COINCEND |
| Blocking Coincidence | D4 | LED(4) | BLOCKCO |
| Data transfer to PC | D7 | LED(7) | DATA_SND |
| Data transfer from PC | D6 | LED(6) | DATA_RCV |

# 6 References

**The Auger Engineering Radio Array**
The Auger Engineering Radio Array
http://www.auger.org/

The GPS receiver module

The ADC ADS4149 specification