PADME RawEvent Reference Manual

v.1.0.0 – 2020-04-05

Emanuele Leonardi (emanuele.leonardi@roma1.infn.it)

TRawEvent

Run and Event Number

```
Int_t  GetRunNumber()
UInt_t GetEventNumber()
```

Run and Event number of this event.
Event Number is assigned by PadmeMerger when the event is assembled.

Event Time

```
TTimeStamp GetEventAbsTime()
ULong64_t  GetEventRunTime()
```

EventAbsTime is the wall clock time (UTC) when the event was assembled by PadmeMerger.
EventRunTime is the low level time of the event expressed in number of 80MHz clock ticks since the start of the run, as counted by the Trigger board.

Trigger mask

```
UInt_t GetEventTrigMask()
Bool_t EventTrigMaskGetBit(UChar_t)
```

Trigger mask contains the trigger bit pattern of the event (8 bits), i.e. which triggers were active when the event was collected. Bits 0 to 5 correspond to the LEMO plugs 1 to 6 on the Trigger board front panel. Bits 6 and 7 are software triggers generated by the Trigger board itself.

Bit 0: BTF trigger
Bit 1: Cosmics trigger
Bit 2: ECal radioactive source trigger
Bit 3: Dual-timer trigger (~50Hz)
Bit 4,5: Reserved
Bit 6: Delayed (off-beam) software trigger
Bit 7: Random software trigger

Event Status

```
UInt_t GetEventStatus()
Bool_t EventStatusGetBit(UChar_t)
```

16 bit pattern encoding the status of the event.

Bit 0: Event is complete (all ADC boards were acquired)
Bit 1: Auto-pass is active
Bit 2-15: Reserved

Missing ADC boards map

```
UInt_t GetMissingADCBoards()
Bool_t MissingADCBoardsGetBit(UChar_t)
```

If event is incomplete (Event Status bit 0 set to 0), store the map of missing boards. Boards 0 to 28 correspond to bits 0 to 28. Bit is set if corresponding board (a) was included in the DAQ and (b) was not found in the final event.

Trigger information

```
TTriggerInfo* TriggerInfo()
```

Get handle to Trigger information structure for this event (see TTriggerInfo documentation below).

ADC Boards information

```
UChar_t GetNADCBoards()
TADCBoard* ADCBoard(Int_t)
```

Get handle to ADC boards information (see TADCBoard documentation below). Use GetNADCBoards to get the number of ADC boards available for this event.

TTriggerInfo

Trigger counter

`UInt_t GetTriggerCounter()`

Trigger board internal event counter (8 bit, i.e. 0 to 255)

Trigger time

`ULong64_t GetTriggerTime()`

40 bit Trigger board clock counter (80MHz). Reset at start of run.

Trigger status pattern

`UInt_t GetTriggerPattern()`

32 bit Trigger board status bit pattern for this event.

Bit 0-8:	Event Trigger mask (see TRawEvent for detailed documentation)
Bit 9-15:	Reserved
Bit 16:	Trigger FIFO status (not relevant for DAQ)
Bit 17:	Trigger AUTO bit (* WARNING * 0: auto-pass ON; 1: auto-pass OFF)

TADCBoard

Board ID

`UChar_t  GetBoardId()`

ID of this ADC board (0 to 28). Current ADC board map is:

Board 0-9: ECal right (door) side
Board 10-12: PVETO
Board 13: HEPVETO
Board 14-23: ECal left (wall) side
Board 24-26: EVETO
Board 27: SAC
Board 28: Target

Board serial number

`UInt_t GetBoardSN()`

Serial number of CAEN board V1742. This is directly obtained from the board when DAQ is started.

Board LVDS pattern

`UShort_t GetLVDSPattern()`

16 bit pattern latched on the V1742 LVDS I/O connector when trigger arrives.
This connector is currently not used and this word is always 0.

Board Status

`UInt_t  GetBoardStatus()`
`UChar_t GetStatus()` Old status pattern (do not use). Kept for backward compatibility.

Board status 10 bit pattern.

Bit 0: Event data content [0=is empty|1=has data]
Bit 1: DRS4 corrections staus [0=not applied|1=applied]
Bit 2: Zero suppression mode [0=rejection, 1=flagging]
Bit 3: Board fail (BF) flag
Bit 4: Event auto-pass flag [0=auto-pass OFF, 1= auto-pass ON]
Bit 5-9: Reserved

When autopass is ON, zero suppression is forced to flagging mode (Bit 2 = 1).

Board Group Mask

`UChar_t GetGroupMask()`

4 bit pattern, 1 bit per group.
If 1, the group is active and contains active channels.
If 0, the group is either not active or does not contain active channels.

Board Group Error Mask (not implemented yet)

```
UChar_t GetGroupErrorMask()
Bool_t  GetGroupError(UChar_t);
```

4 bit pattern, 1 bit per group.
If 1, group StartIndexCell is not coherent with the other groups.
Other error conditions might be added in future.

Board Event Counter

`UInt_t GetEventCounter()`

22 bit counter with progressive number of event (number of triggers) within the run.

Board Time Tag

`UInt_t GetEventTimeTag()`

Event Time Tag: time of event since last reset (counter incremented at each sampling clock hit).
WARNING: this time is related to the digitizing process of the event and is not precise. Use Trigger Time Tag instead (see Trigger Time Tag in TADCTrigger below).

Board Zero Suppression Algorithm

`UChar_t Get0SuppAlgrtm()`

4 bit (0-15) code of algorithm used for zero suppression.

0: no 0-suppression algorithm applied
1: event is accepted a group of consecutive samples above a given threshold is found
2: event is accepted if RMS of the samples is above a given threshold
3-15: reserved

Active Channel Mask

`UInt_t GetActiveChannelMask()`

32 bit mask encoding which channels were active (i.e. DAQ read them) in current event.

Accepted Channel Mask

UInt_t GetAcceptedChannelMask()

32 bit mask encoding which channels were accepted (i.e. passed the zero suppression algorithm) in current event.

ADC Channels Information

UChar_t GetNADCChannels()
TADCChannel* ADCChannel(Int_t)

Get handle to ADC channels information (see TADCChannel documentation below). Use GetNADCChannels to get the number of ADC channels (0 to 32) available for this board.

ADC Triggers Information

UChar_t GetNADCTriggers()
TADCTrigger* ADCTrigger(Int_t)

Get handle to ADC triggers information (see TADCTrigger documentation below). Use GetNADCTriggers to get the number of ADC triggers (0 to 4) available for this board.

TADCChannel

Channel Number

```
UChar_t GetChannelNumber()
```

Number of this channel (0-31) within the ADC board.

Channel Sample Values

```
UShort_t GetNSamples()
Short_t  GetSample(Int_t)
Short_t* GetSamplesArray()
```

Number of samples is always 1024.
User can retrieve either the value of a single sample (GetSample), or an array with all the 1024 sample values (GetSamplesArray).
Sample values can exceed the 12 bit (0-4095) range due to CAEN V1742 correction algorithm. They are therefore encoded into a signed 16 bit variable (Short_t).

TADCTrigger

Group Number

UChar_t GetGroupNumber()

Number of this group (0-3) within the ADC board.

Start Index Cell

UShort_t GetStartIndexCell()

Start Index Cell (SIC) is used to map sample indexes to the corresponding physical capacitor on the DRS4 chip, e.g. SIC=27 means that sample 0 was read from capacitor number 27 in the DRS4 capacitor array, sample 1 from capacitor 28, and so on. This information is mainly used to apply capacitor-related corrections to the raw readings from the V1742 and is usually not needed for off-line analysis.

Frequency

UChar_t GetFrequency()

2 bit (0-3) code for frequency at which data was sampled.

0: 5 GHz
1: 2.5 GHz
2: 1 GHz
3: not used (will be used for 800 MHz sampling frequency when available)

Trigger Signal Readout

Bool_t GetTriggerSignal()

Trigger signal sampling enable status (true: enabled, false: disabled). It is usually enabled.

Trigger Time Tag

UInt_t GetTriggerTimeTag()

Trigger arrival time since last ADC board reset. Time is in 8.5ns clock counts.

Trigger Sample Values

UShort_t GetNSamples()
Short_t GetSample(Int_t)
Short_t* GetSamplesArray()

If Trigger Signal Readout is enabled, number of samples is 1024, 0 otherwise.

User can retrieve either the value of a single sample (GetSample), or an array with all the 1024 sample values (GetSamplesArray).
Sample values can exceed the 12 bit (0-4095) range due to CAEN V1742 correction algorithm. They are therefore encoded into a signed 16 bit variable (Short_t).