# Scaler information in CLAS12 2018 data

Since January 2018, scaler banks are injected in CLAS12 raw data. These collect readings from different boards installed in the same crate (crate id 64), specifically:

* “helicity” scalers (bank tag 0xE125) and
* integrating scalers (bank tag 0xE115),

Scaler readout is triggered by the helicity strobe signal (30 Hz), which is also used as a trigger for the data acquisition, being one of the signals connected to the TS front panel. A scaler bank is therefore expected to be present in the data stream every 33 ms.

The signals recorded by both helicity and integrating scalers are the Faraday Cup, the Synchrotron Light Monitor (SLM) and a MHz clock.

The scaler banks recorded in the EVIO raw data file are written to the RAW::scaler hipo bank (id 20013).

## Helicity scalers

In the case of the helicity scalers, the 3 signals mentioned above are integrated over two time intervals for each helicity state: the 500 us settling time of the Pockel cell, which is used to randomly set the helicity, and the remaining of the helicity state duration, i.e. approximately 33 ms. For each of these time intervals, two sets of readings are reported by two boards, the first gated with the DAQ live signals and the second with no gating.

Each scaler reading written to the EVIO composite bank is a 32 bit integer with the following structure:

* bit 0-23: scaler value,
* bit 24-28: channel id,
* bit 29: integration time interval (0=500us or 1=33 ms),
* bit 30: the quartet signal (0 for the first helicity state of a quartet[[1]](#footnote-1) and 1 for the remaining three),
* bit 31: the helicity signal.

The java code used to decode this information and create the corresponding hipo bank is in the CodaEventDecoder class of CoatJava clas-detector/decode package. The most relevant lines are reported below for reference:

if(node.getTag()==57637) {

int helicity = DataUtils.getInteger(dataEntry, 31, 31);

int quartet = DataUtils.getInteger(dataEntry, 30, 30);

int interval = DataUtils.getInteger(dataEntry, 29, 29);

int id = DataUtils.getInteger(dataEntry, 24, 28);

long value = DataUtils.getLongFromInt(DataUtils.getInteger(dataEntry, 0, 23));

if(id < 3) {

DetectorDataDgtz entry = new DetectorDataDgtz(crate,num,id+32\*interval);

SCALERData scaler = new SCALERData();

scaler.setHelicity((byte) helicity);

scaler.setQuartet((byte) quartet);

scaler.setValue(value);

entry.addSCALER(scaler);

scalerEntries.add(entry);

}

}

An example of a hipo bank with helicity scalers is shown below (from run 3741):

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EVENT # 9779 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

------------------------+---------------------------+

>>>> GROUP (group= 20013) (name=RAW::scaler):

------------------------+---------------------------+

crate (BYTE): 64 64 64 64 64 64 64 64 64 64 64 64

slot (BYTE): 0 0 0 0 0 0 1 1 1 1 1 1

channel(SHORT): 32 33 34 0 1 2 32 33 34 0 1 2

helicity(BYTE): 0 0 0 0 0 0 0 0 0 0 0 0

quartet (BYTE): 0 0 0 0 0 0 0 0 0 0 0 0

value (INT) : 144 29998 32016 3 450 484 152 31239 33330 3 463 500

where, because of the encoding adopted, the meaning of slot and channel are as follows:

* slot 0/1 correspond to the gated/ungated boards,
* channel 0/1/2 and 32/33/34 correspond to fcup/slm/clock for the 33 ms and 500 us intervals, respectively.
  + Note, the bank printout above was made in a prior software/hardware configuration, when the two intervals were swapped

Before February 5 2018, the helicity scalers configuration was still being checked and part of the data collected are affected by the following issues:

* slot 0 was gated with the DAQ busy signal or was ungated,
* the clock signal was not connected.

## Integrating scalers

Since approximately February 5 2018, the readout of integrating scalers was added. These integrate fcup, slm and clock, starting from the beginning of a run, with both gated and ungated values.

The EVIO bank 0xE115 has always 72 words:

* block header,
* event header,
* scaler header,
* 16 TRG gated scaler values,
* 16 TDC gated scaler values,
* 16 TRG ungated scaler values,
* 16 TDC ungated scaler values,
* gated reference,
* ungated reference,
* DSC2 filler (always 0),
* VME filler,
* bock trailer.

Even if all 16 input channels are written to the bank, only the first 3 are used, corresponding to fcup, slm and clock. The gate signal comes from the TS busy.  
The TRG and TDC values correspond to different discriminators that can have in principle different thresholds: these are currently set to the same values so that the reported scaler values are expected to be very identical.

In the RAW::scaler hipo bank, the corresponding information appears as follows:

---------------------------+------------------------------+------------------------------+  
>>>> GROUP (group= 20013) (name=RAW::scaler):  
---------------------------+------------------------------+------------------------------+  
crate (BYTE): 64 64 64 64 64 64 64 64 64

64 64 64  
slot (BYTE): 64 64 64 64 64 64 64 64 64

64 64 64  
channel (SHORT): 0 1 2 16 17 18 32 33 34

48 49 50  
helicity (BYTE): -1 -1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1  
quartet (BYTE): -1 -1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1  
value (INT): 9960 1850780 4737052 9966 3668703 4737066 305106 52128164 212451068

305183 103257477 212451068  
---------------------------+------------------------------+------------------------------+

Helicity and quartet are to -1 (undefined) since the same structure created for the helicity scaler is used while this scaler board does not record the corresponding information.

Different scaler inputs are identified by the channel number as follows:

channel = i + 16 \* j

with:

* i = 0,1,2 -> FCUP, SLM, Clock,
* j = 0,1,2,3 -> gated TRG, gated TDC, ungated TRG, ungated TDC.

## Helicity FADC

The bank’s name is HEL::adc and has the same format as all other decoded adc banks. The sector and layer values should always be 1, and component is 1/2/3 for the helicity/sync/quartet signals. The state of each is contained in the ped variable, where the high state is around 3400 and the low state is around zero. See also <https://logbooks.jlab.org/entry/3531353>

1. Currently, the helicity sequence consists of a pseudo-random sequence of quartets of states whose pattern is 1001 or 0110. [↑](#footnote-ref-1)