

# PocketDAQ output files and related

**Marko Starič**



Belle II collaboration



Jožef Stefan Institute, Ljubljana

TOP software meeting

## Files: root vs. binary

Benefits of using Belle II standard root files:

- File reader exists: RootInput module
- Files are organized on event-by-event basis
- Raw data contained within objects (classes)
- No big-endian/little-endian issue
- Raw data unpackers are mostly available
  - TOPUnpacker module (needs update to new waveform data format)

PocketDAQ output root file

- event data contained in RawDataBlock
- have to be converted to RawDetector objects (RawTOP, RawCDC,...)
- module Convert2RawDet provided for this

## Event execution flow

|                |  |
|----------------|--|
| RootInput      | runXYZ.root (output from PocketDAQ)    |
| Convert2RawDet | RawDataBlocks → RawTOPs                |
| TOPUnpacker    | RawTOPs → TOPRawWaveforms or TOPDigits |
| ...            | (other modules follow)                 |

How to include TOPCAF in this scheme:

- Design a module that converts TOPRawWaveforms to relevant TOPCAF dataobjects
  - not much coding needed
- Attach it after TOPUnpacker and add other TOPCAF modules.

## class TOPRawWaveform

private members:

```
int m_moduleID = 0;           /**< module ID */
int m_pixelID = 0;           /**< software channel ID */
unsigned m_channel = 0;       /**< hardware channel number */
unsigned short m_scrodID = 0; /**< SCROD ID */
unsigned short m_scrodRevision = 0; /**< SCROD revision number */
unsigned m_freezeDate = 0;    /**< protocol freeze date (YYYYMMDD in BCD) */
unsigned short m_triggerType = 0; /**< trigger type (bits 0:7) */
unsigned short m_flags = 0;   /**< event flags (bits 0:7) */
unsigned short m_referenceASIC = 0; /**< reference ASIC window */
unsigned short m_segmentASIC = 0; /**< segment ASIC window (storage window) */
std::vector<unsigned short> m_data; /**< waveform ADC values */
unsigned m_electronicType = 0; /**< electronic type (see ChannelMapper::EType) */
std::string m_electronicName; /**< electronic name */
```

- Class stores data of a single ASIC window (64 sample values).
- Basically all information from raw data available also here.
  - maybe need to add something more (adjust to new format)

## Automatic data format detection

Some time ago I proposed (shown also at BPAC):

- Finesse data record begins with a header word that encodes format type, format version and SCROD ID
  - type: additional formats can be defined in the future
  - version: format can be changed while preserving backward compatibility
- automatic data format detection possible by TOPUnpacker
- Words that follow are the data from one SCROD

| word | 31:24                    | 23:16   | 15:8     | 7:0 |
|------|--------------------------|---------|----------|-----|
| 0    | format type              | version | SCROD ID |     |
| 1    | data from a single SCROD |         |          |     |
| ...  |                          |         |          |     |
| N-1  |                          |         |          |     |

Buffer size (N) is kept within RawTOP class

# To do

- Update TOPUnpacker with the new data format (Marko)
- Design a module in topcaf for conversion of TOPRawWaveforms to TOPCAF dataobjects (Jan)
- Provide current data format description (Lynn)
- Always copy also root file to KEKCC (CRT shifters, Matt)