DAQ Software Status

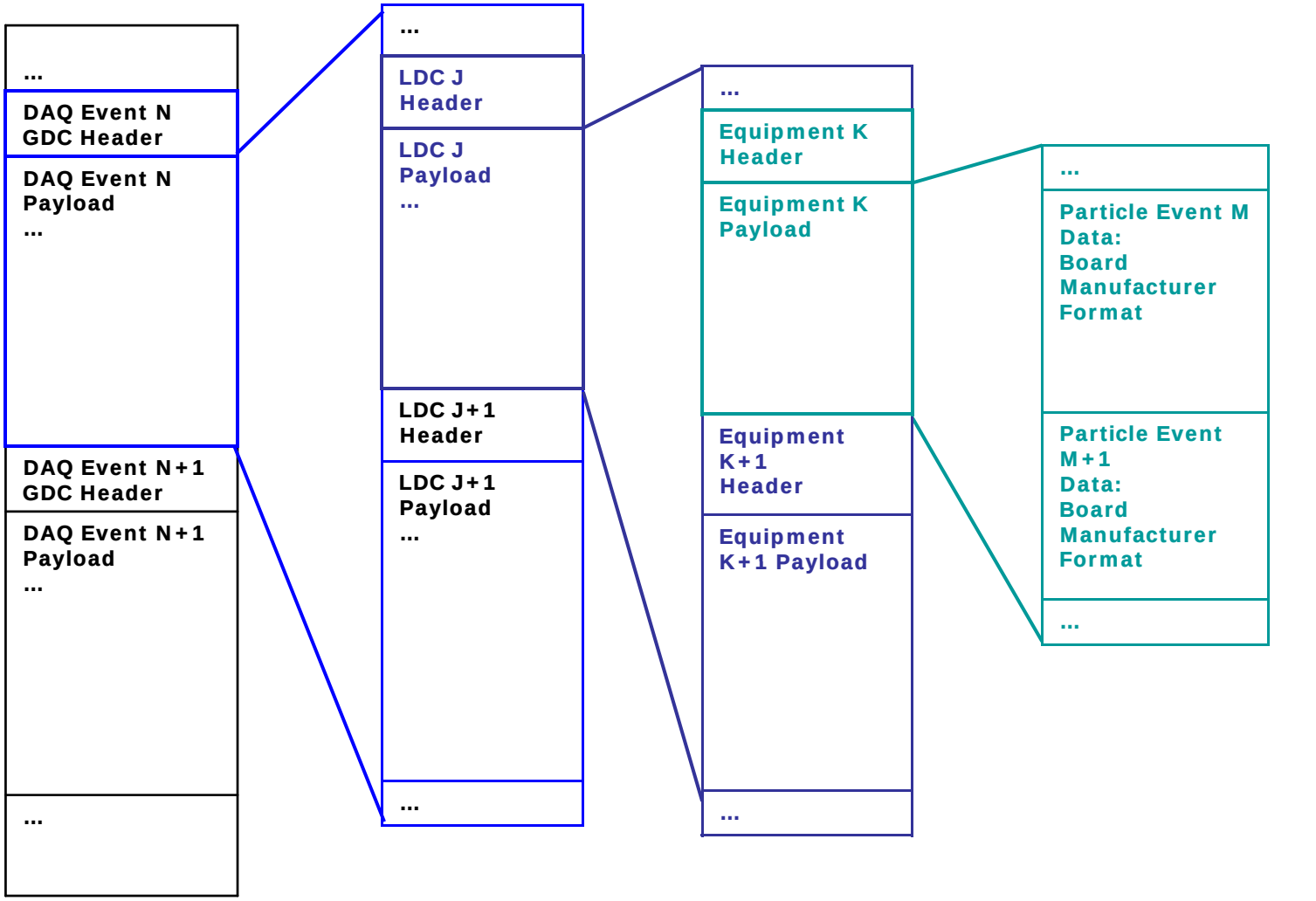
- DATE Readout
- Data Decoding
- Data Monitoring
- Next steps
- Summary

CM20 Vassil Verguilov

DATE Data Readout

- DATE readout code completed for the fADC (CAEN V1724)
- New fADC firmware from CAEN
  - Zero Suppression implemented
  - The fADC can acquire more then 1 trigger per event
- Data taking process is tested with cosmics

Data Unpacking



DATE Event header Format

Event Header

Event Size
Sync. Word
Header Size
Header Version
EventType
RunNb
Event Id[0]
Event Id[1]
TriggerPattern[0]
TriggerPattern[1]
DetectorPattern[0]
DetectorPattern[1]
Attribute[0]
Attribute[1]
Attribute[2]
LDC Id
GDC Id
TimeStamp[0]
TimeStamp[1]

Same header for Super- and Sub-Events (only the attribute value is different)

DATE Equipment Header Format

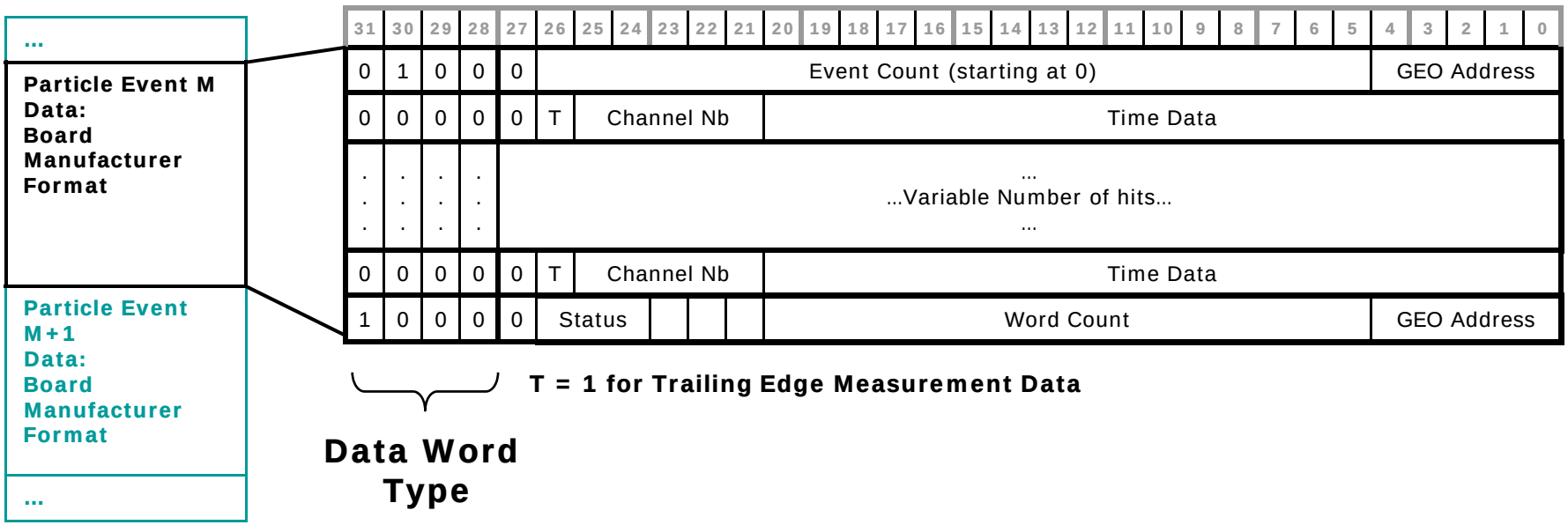
Equipment Header

- **Equipment Data Size**
- **Equipment Type**
- **Equipment User Id**
- **Equipment Attribute[0]**
- **Equipment Attribute[1]**
- **Equipment Attribute[2]**
- **Equipment Word Size**

Conventional Table of Equipment Type:

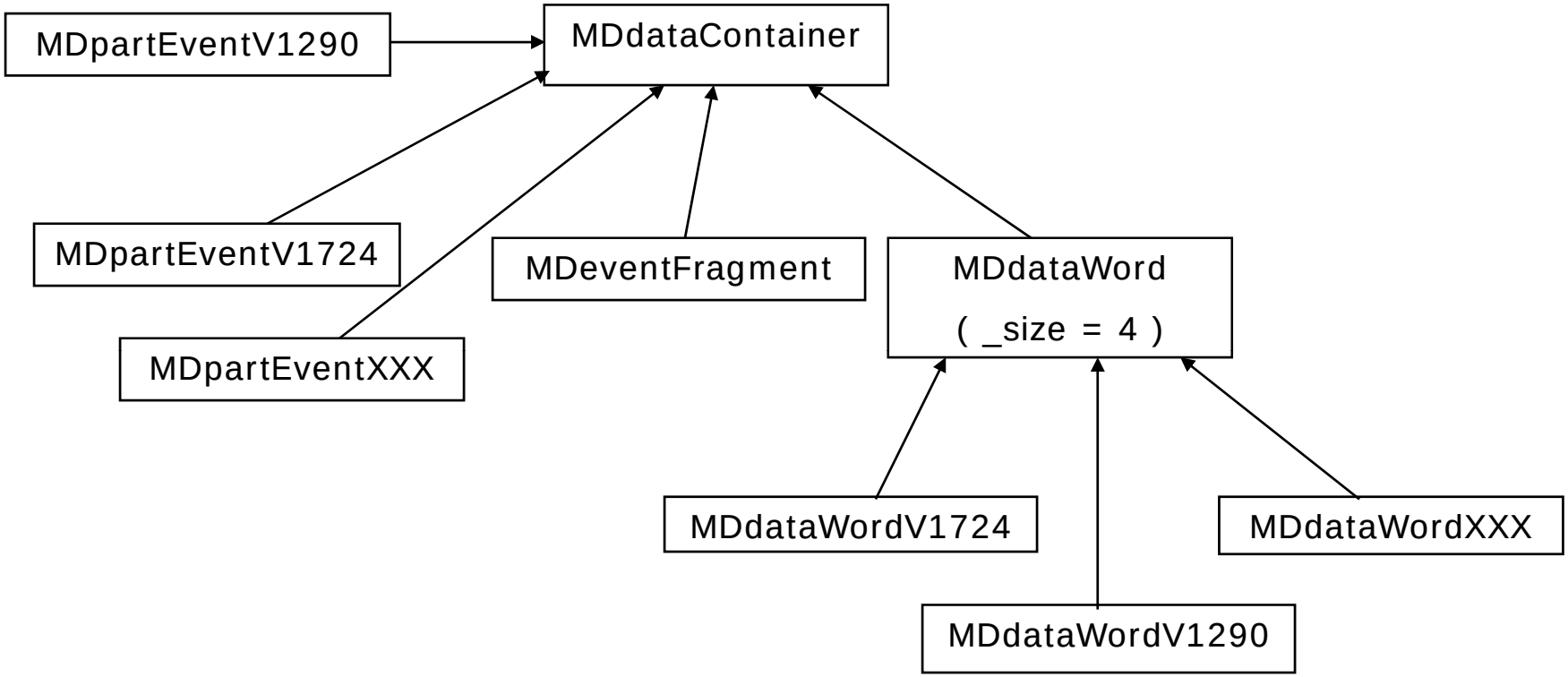
Type	Equipment
0	Random Generator
100	V2718
101	Trigger Receiver
102	TDC V1290
104	VLSB
110	Trailer
111	Sclaer V830
120	fADC V1724

Particle Data Format Example - CAEN V1290 TDC



TDC V1290

Data Unpacking Classes



Each class
- Has its own unpacking member functions
- Has a Dump() virtual member function

Data Unpacking Classes

- MDdataContainer - base class for all
- MDdataWord - base class for all (*SetDataWord(void * d)*)
- MDdataWordXXX - classes implementing the data format (at word-level) of each equipment
  - MDdataWordV1724: *GetSample*
  - MDdataWordV1290: *GetMeasurement, GetChannel, GetTDC, GetError, GetWordCount, GetBunchID, GetEventID*
- MDpartEventXXX - classes manipulating the data (at event level) from each equipment using corresponding MDdataWordXXX class
  - MDpartEventV1724: *GetPattern, GetChannelMask, GetTriggerTimeTag, GetSampleData*
  - MDpartEventV1290: *GetHitMeasurement, GetHitType, GetHitChannel, GetNHits*
- MDeventFragment - container for the particle events
- MDequipMap - Class using a hash to determine which object (MDpartEventXXX) can decode specific event, based on the Equipment Id of the event
- MDdateFile - IO routines for the DATE raw data file

Data Dump Example

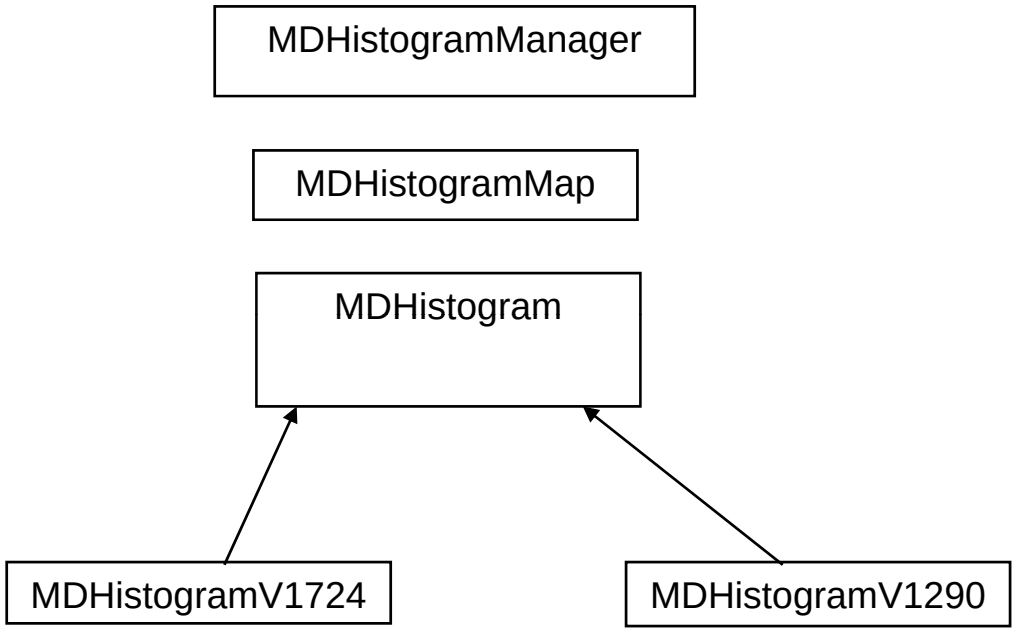
```
#include <ctype.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "MDevent.h"
#include "event.h"
#include "MDdateFile.h"

using namespace std;

/* Simple main */
int main( int argc, char **argv ) {

MDevent mdEvent;
MDdateFile dFile("59","/data/mice");
char *eventBuffer;
unsigned int eventCount(0);
if ( dFile.OpenFile() == DATE_FILE_OK ) {
                        while ( eventBuffer = dFile.GetNextEvent() ) {
                                        cout << "Event Count : " << eventCount++ << endl;
                        char* ptr = eventBuffer;
                                        mdEvent.SetDataPtr(ptr);
                                        mdEvent.Dump();
                        }
} else {
                        cout << "Error in opening the file. Aborted." << endl;
}
return 0;
} /* End of main */
```

No dependency from DATE! (uses only event.h)

Data Dump Example

```
- - - - - - - - - - - - - MDeventFragment Dump - - - - - - - - - - - - - - - -
Size:28 (header:28) Type:110 EquipId:0 BasicSize:0
Attributes: (00000000.00000000.00000000)
Event Count : 3
..... MDevent Dump .....
Size:8416 (header:68) Version:0x00030009 Type:PhysicsEvent
runNb:96 nbInRun:2 burstNb:0 nbInBurst:0 ldcId:1 gdcId:VOID time:Thu Feb  7 02:43:33 2008
Attributes:noAttr (00000000.00000000.00000000)
triggerPattern:80000000-00000020 detectorPattern:00000000[invalid]
- - - - - - - - - - - - - MDeventFragment Dump - - - - - - - - - - - - - - - -
Size:32 (header:28) Type:101 EquipId:0 BasicSize:4
Attributes: (00000000.00000000.00000000)
        0) 00000003 (....)
- - - - - - - - - - - - - MDeventFragment Dump - - - - - - - - - - - - - - - -
Size:52 (header:28) Type:102 EquipId:10 BasicSize:4
Attributes: (00000004.00000000.00000000)
 Decoding equipment V1290
 CAEN V1290 Global Header  : Event Count : 1; Geo Address : 0
 CAEN V1290 Leading edge Measurement: 47209 ; Channel: 1
 CAEN V1290 Leading edge Measurement: 47290 ; Channel: 0
 CAEN V1290 Trailing edge Measurement: 47327 ; Channel: 1
 CAEN V1290 Trailing edge Measurement: 47396 ; Channel: 0
 CAEN V1290 Global Trailer  : Word Count : 6; Geo Address : 0
- - - - - - - - - - - - - MDeventFragment Dump - - - - - - - - - - - - - - - -
Size:8236 (header:28) Type:120 EquipId:120 BasicSize:4
Attributes: (00000000.00000000.00000000)
 Decoding equipment V1724
 Init ch 0 : 100 ; 0x83b86ec
 Init ch 1 : 100 ; 0x83b8aec
 Init ch 2 : 100 ; 0x83b8eec
 Init ch 3 : 100 ; 0x83b92ec
 Init ch 4 : 100 ; 0x83b96ec
 Init ch 5 : 100 ; 0x83b9aec
 Init ch 6 : 100 ; 0x83b9eec
 Init ch 7 : 100 ; 0x83ba2ec
 ------------   CAEN V1724 Header    ------------
 Word Count        : 2052
 Geo               : 0 ; ZLE disabled ; Channel Mask : 0xff
 Event Counter     : 3
 Trigger Time Tag  : 558607545
 ------------ End of CAEN V1724 Header -----------
 ----------   Channel 0 ( Length = 256 ) ----------
 CAEN V1724 Data: 8423 ; 8424
 CAEN V1724 Data: 8424 ; 8421
 CAEN V1724 Data: 8422 ; 8422
 CAEN V1724 Data: 8424 ; 8422
 CAEN V1724 Data: 8420 ; 8422
 CAEN V1724 Data: 8420 ; 8424
 CAEN V1724 Data: 8422 ; 8424
 CAEN V1724 Data: 8426 ; 8424
 CAEN V1724 Data: 8423 ; 8422
```

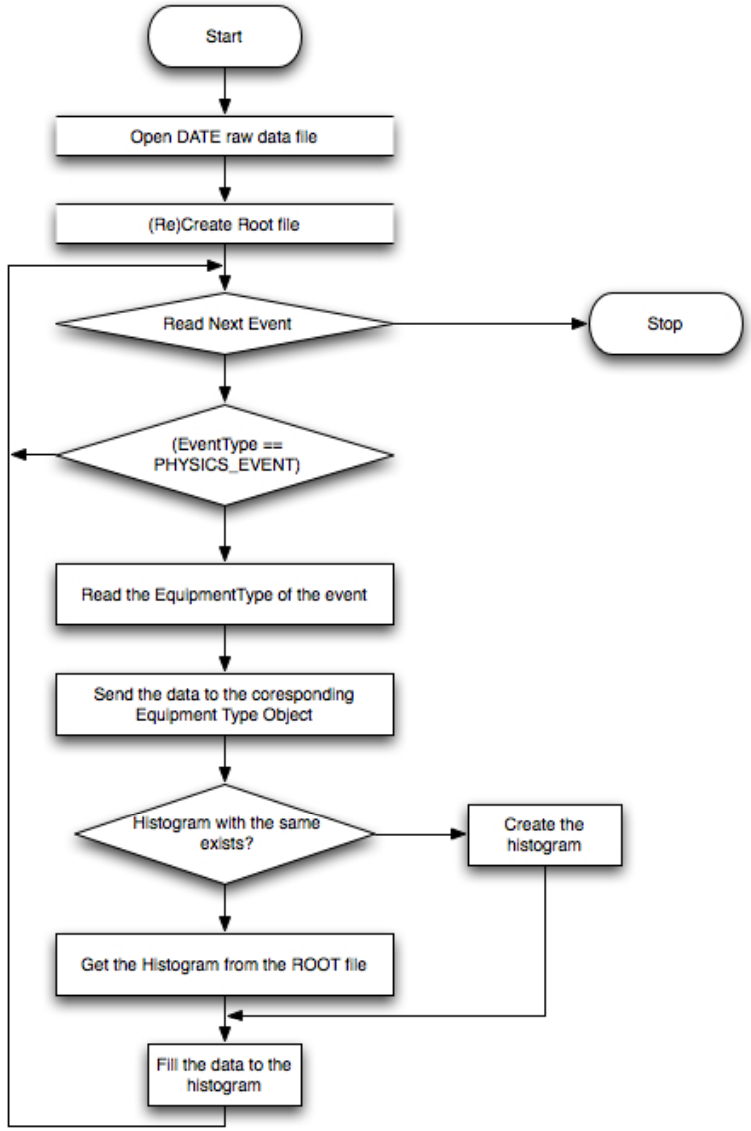
Data Monitoring Classes



Data Monitoring Classes

- MDHistogram - base abstract class for all the equipment histograms
- MDHistogramXXX - classes for creating histograms depending on the equipment type.
  - MDHistogramV1290: Creates 5 types of histograms
  - MDHistogramV1724: Creates 1 type of histogram
- MDHistogramMap - Manipulates a hash which contains 2 parameters - equipment type id (integer) and corresponding class, manupulating the equipment data.
- MDHistogramManager - manages the MDHistogramXXX classes and save the histograms to ROOT file.
- The classes are ROOT dependant and use the Data Decoding Classes

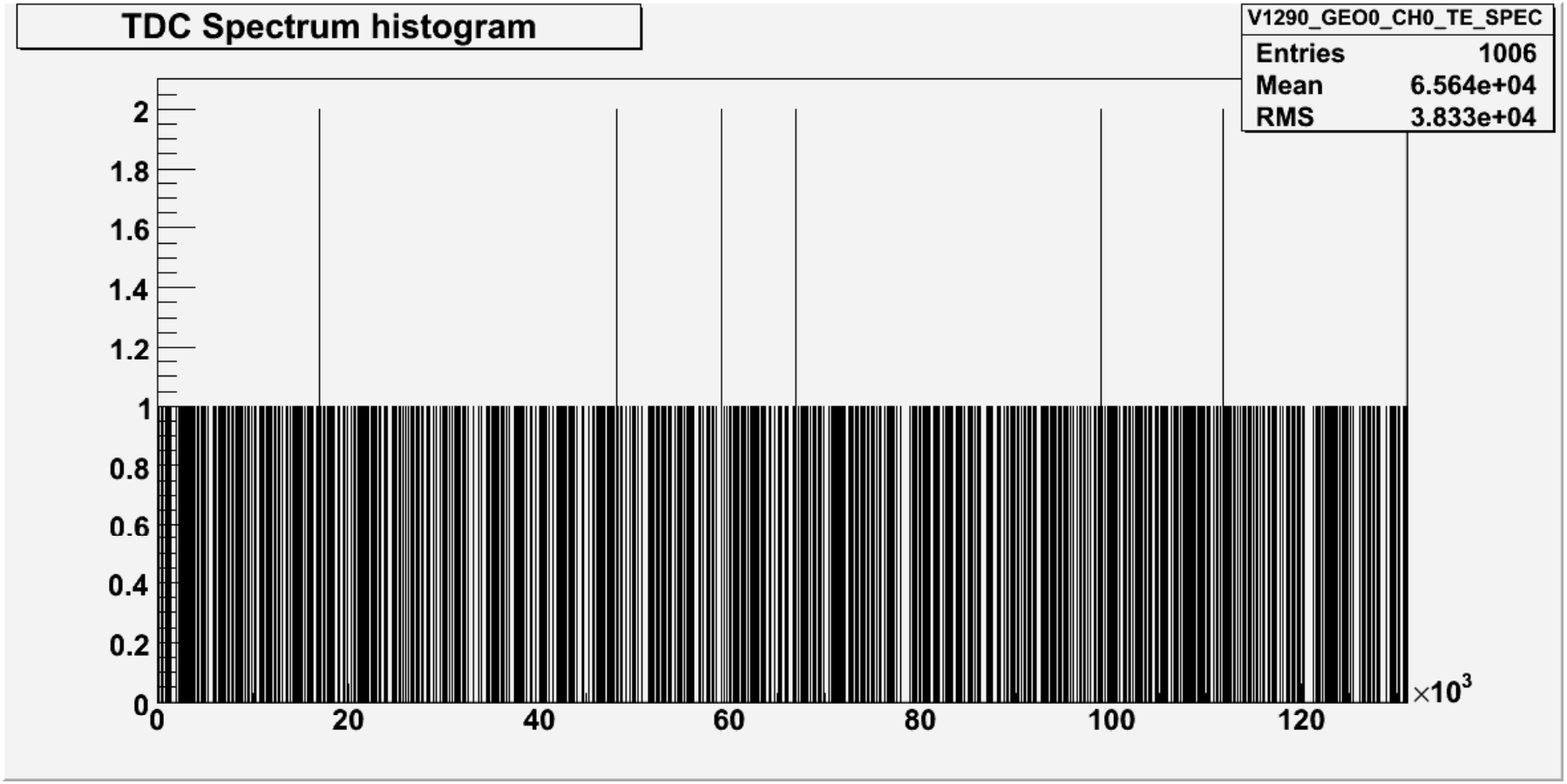
Data Monitoring Simple Application - Flow Chart



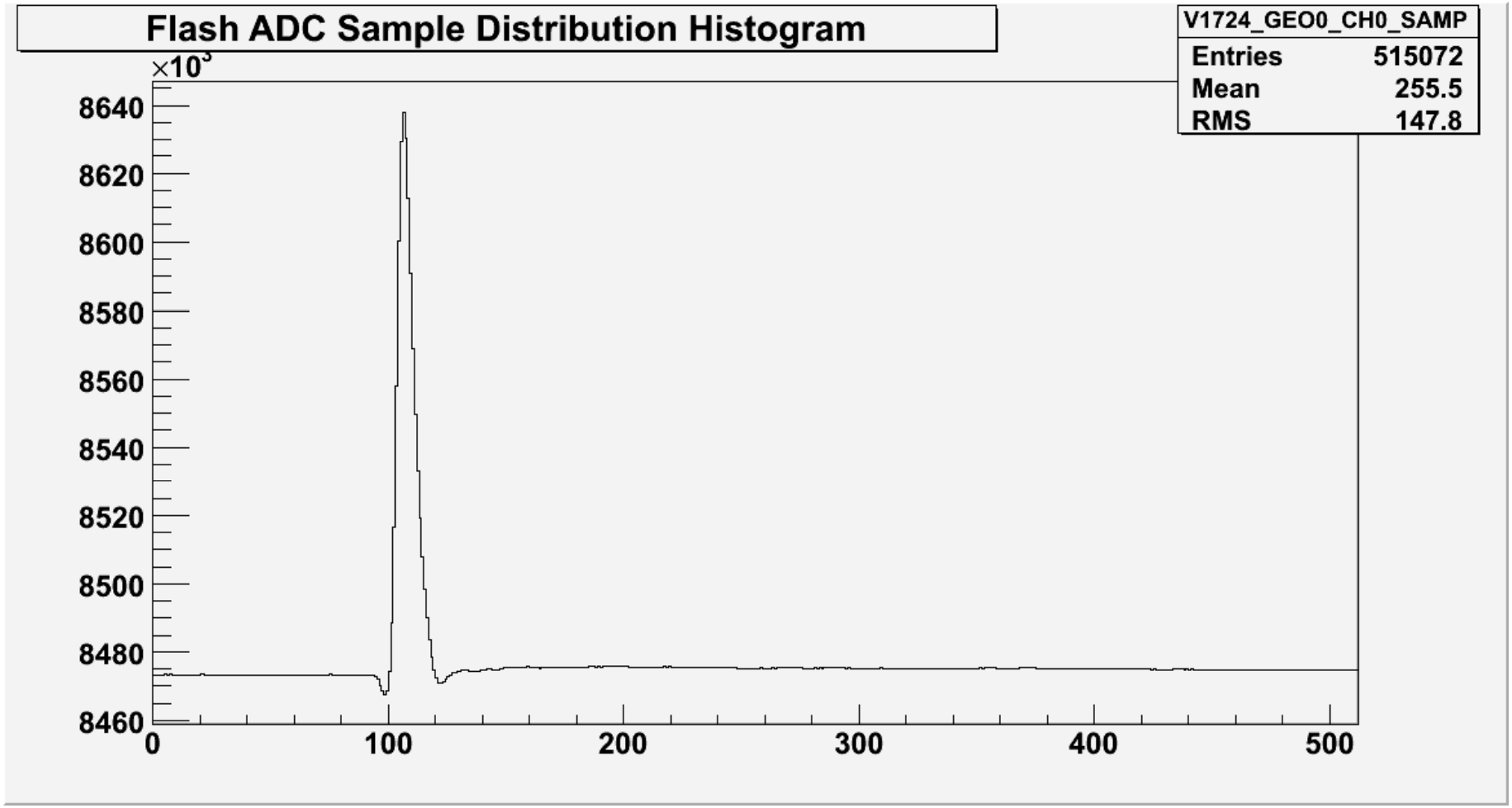
Data Monitoring Simple Application

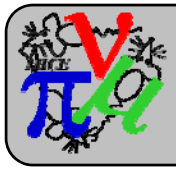
```
#include <iostream>
#include "MDevent.h"
#include "MDHistogramV1290.h"
#include "MDdateFile.h"
#include "MDHistogramManager.h"

int main(){

MDdateFile dFile( "96","/data/mice" );

if ( dFile.OpenFile() == DATE_FILE_OK ) {

                MDHistogramManager HistoManager( "histo76.root" );
                MDevent DaqEvent;
                unsigned int eventCount( 0 );

                while ( char * ptr = dFile.GetNextEvent() ) {
                        cout << "Processing event  " << dec << eventCount++ << " ..." << endl;
                        HistoManager.Process( (unsigned char *) ptr );
                        cout << "Done." << endl << endl;
                }

} else {
                cout << "Error opening the file. Aborted." << endl;
}

return 0;
}
```

TDC Sample Histogram



fADC Sample Histogram





Next Steps

- Optimize DATE readout code
- Implement ZS for the fADC
- Read all the readout parameters from DATE database
- Read the monitoring data remotely in real-time
- Automatic installation system for the DAQ PCs

Summary

- DAQ DATE Readout is finished
- Framework for data decoding is created
- Basic monitoring classes are developed
- Data readout, decoding and monitoring are tested with real data