Apply Calibration

Wednesday, July 29
3

- Calibrate TOF raw items (calculate time using TDC channel)

➡ create new data class

➡ create new analysis task

➡ apply calibration parameters

Create data class R3BTofTimeItem

- r3broot/r3bdata/tofData/ R3BTofRawItem.h ….cxx
- Inherits from TObject
- Data members (private):
  - ➡ Int_t fChannelId;
  - ➡ Double_t fTime;
- public: Get… / Set… functions

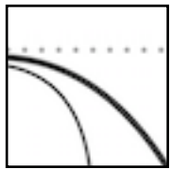
- r3broot/r3bdata
- Add tofData/R3BTofTimeItem.cxx file to CMakeLists.txt
- Add to R3BDataLinkDef.h

  ➡ #pragma link C++ class R3BTofTimeItem+;

- Recompile

Create analysis task R3BTofTcal

- This task will read TofRawItem and produce TofTimeItem by applying time calibration parameters



- cd r3broot/tof
- copy R3BTofFillTcal to R3BTofTcal
- rename

Declaration

- #include <map>
- class R3BTofTCalPar;

- Data members (private):
  - ➡ TClonesArray *fRawData; // Input data
  - ➡ TClonesArray *fTimeData; // Output data
  - ➡ Int_t fNTimeItems; // Counter for produced hits
  - ➡ R3BTofCalPar *fCal_Par; // Calibration parameters
  - ➡ std::map<Int_t, R3BTofTCalPar*> fMapPar; // Map for fast search: channelID —> calibration

- Additional overloaded functions (public):
  - ➡ void SetParContainers();
  - ➡ InitStatus ReInit();

Implementation

- Constructors (member initialization area):

```
fTimeData(new TClonesArray(“R3BTofTimeItem”)),
fNTimeItems(0)
```

Parameter Initialization

- #include “FairRunAna.h”

- void R3BTofTcal::SetParContainers()
- {

```
    // Get access to runtime database and
     FairRunAna* ana = FairRunAna::Instance();
     FairRuntimeDb* rtdb = ana->GetRuntimeDb();

     // Get pointer to parameter container
     fCal_Par = (R3BTofCalPar*)(rtdb-
>getContainer(“TofCalPar"));
```

- }

Re-initialization (in case Run ID has changed)

- InitStatus R3BTofTcal::ReInit()
  - ➡ SetParContainers();
  - ➡ return kSUCCESS;

Task Initialization

```
// Fill auxiliary map for fast access
R3BTofTCalPar* par;
for(Int_t i = 0; i < fCalPar->GetNumTCalPar(); i++)
{
    par = fCalPar->GetTCalParAt(i);
    fMapPar[par->GetBarId()] = par;
    par->Print();
}

// Get input data
FairRootManager* mgr = FairRootManager::Instance();
fRawData = (TClonesArray*) mgr->GetObject("TofRawItem");

// Register output data
mgr->Register("TofTimeItem", "Tof", fTimeData, kTRUE);

return kSUCCESS;
```

Event loop implementation

- void R3BTofTcal::Exec(Option_t *option)

```
Int_t nRawItems = fRawData->GetEntriesFast();
for(Int_t i = 0; i < nRawItems; i++)
{
    // Get pointer to raw hit
    R3BTofRawItem* rawItem= (R3BTofRawItem*) fRawData->At(i);
    Int_t tdc = rawItem->GetTDC();

    // Get pointer to relevant calibration
    Int_t channel = rawItem->GetChannelId();
    R3BTofTCalPar *par = fMapPar[channel];
    if(NULL == par)
    {
      continue;
    }

    // Convert TDC to [ns]
    Double_t time = tdc/1000.*par->GetTimeAt(0);

    // Produce time calibrated hit
    new ((*fTimeData)[fNTimeItems]) R3BTofTimeItem(channel,time);
    fNTimeItems += 1;
}
```

Reset after each event

- void R3BTofTcal::FinishEvent()

```
fTimeData->Clear();
fNTimeItems = 0;
```

Compilation

- CMakeLists.txt
  - ➡ set(SRCS
  - ➡ …
  - ➡ R3BTofTcal.cxx)
- TofLinkDef.h
  - ➡ #pragma link C++ class R3BTofTcal+;
- Recompile

- R3BTofTcal.cxx :

- #include “R3BTofTimeItem.h”

- cd r3broot/macros/r3b/unpack/tof

- new file run_time_calib.C

Create analysis macro

```
{
    // Create analysis run ----------------------------------------------------
    FairRunAna* run = new FairRunAna();
    run->SetInputFile(inputFileName);
    run->SetOutputFile(outputFileName);

    // ----- Runtime DataBase info --------------------------------------------
    FairRuntimeDb* rtdb = run->GetRuntimeDb();
    FairParRootFileIo* parIo1 = new FairParRootFileIo();
    parIo1->open(parFileName);
    rtdb->setFirstInput(parIo1);
    rtdb->setOutput(parIo1);
    rtdb->saveOutput();

    // Time calibration -------------------------------------------------------
    R3BTofTcal* tofTcal = new R3BTofTcal("TofTcal", 1);
    run->AddTask(tofTcal);

    // Initialize -------------------------------------------------------------
    run->Init();

    // Run --------------------------------------------------------------------
    run->Run();
}
```

- Now there are 2 instances of TCAL calibration for NeuLAND and TOF
- Algorithm and parameter containers are equivalent
- Create generic TCAL calibration task