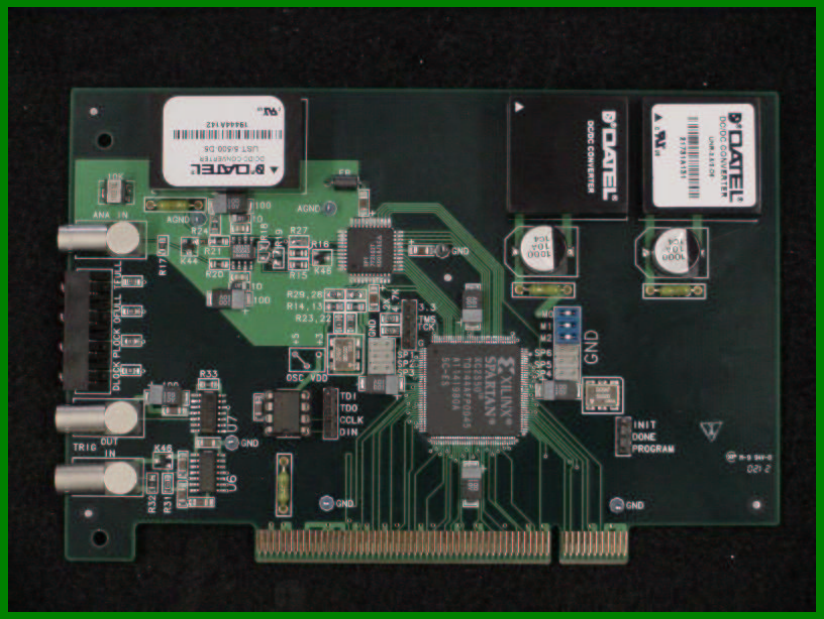
The Many Uses of the Indiana University fADC

Timothy Paul Smith
timothy.p.smith@dartmouth.edu
http://he3.dartmouth.edu
Dartmouth College

What is the fADC?



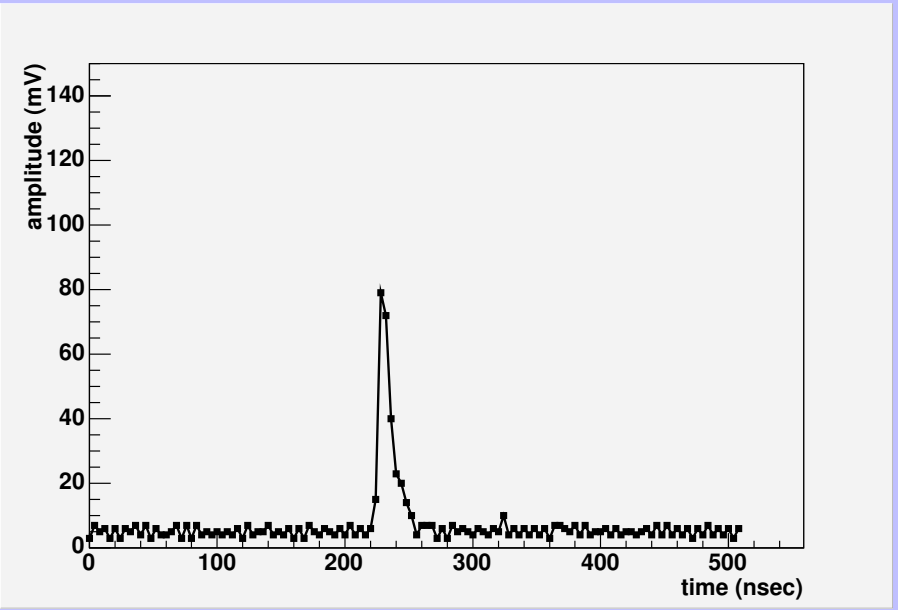
- Digitize signal every 4 ns - cached in ring buffer
- Generates Trigger
- Transfers data from Ring Buffer to Readout Buffer
- Accesses data via PCI-bus

Outline

- Things you could do with a digitized pulse
- Test Bed
- Root interface
- CODA Interface
- Future and Conclusion

Wild Ideas - Things You Could Do With A Digitized Pulse

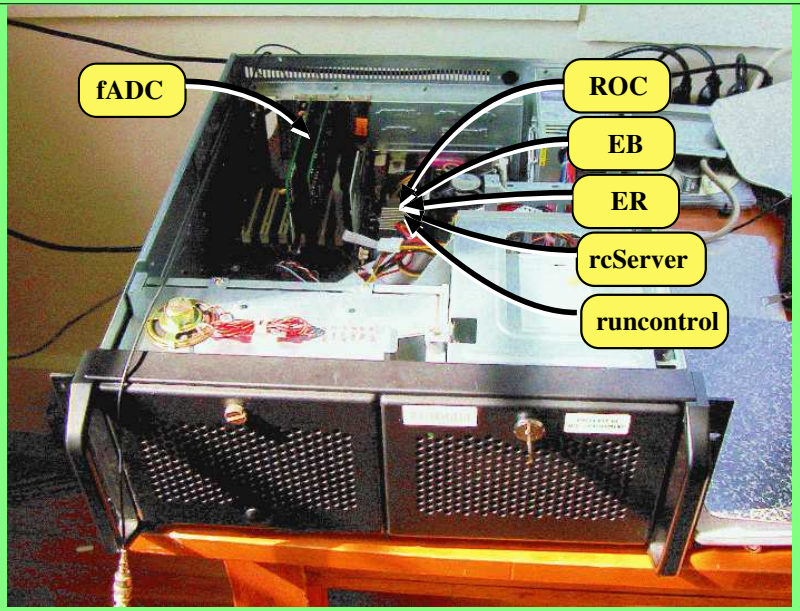
- **Integrate (sum) the pulse; classical ADC**
- **Subtract the pedestal**
- **Fit the Peak**
- **Timing ?**
- **Double hit resolution**
- **Use pulse shape for ...(?)**



Testbed

- To test our software we need signals
- Any simple Cosmic ray will do
- We have some scintillators, HV and computers from MIT-Bates

Testbed - All of CODA in a single Linux box



- ROC - Read-out-controller
- EB / ER / ET - Event Builder/Recorder/Transfer
- rcServer & runcontrol
- ... or we can read the fADC directly via. Root

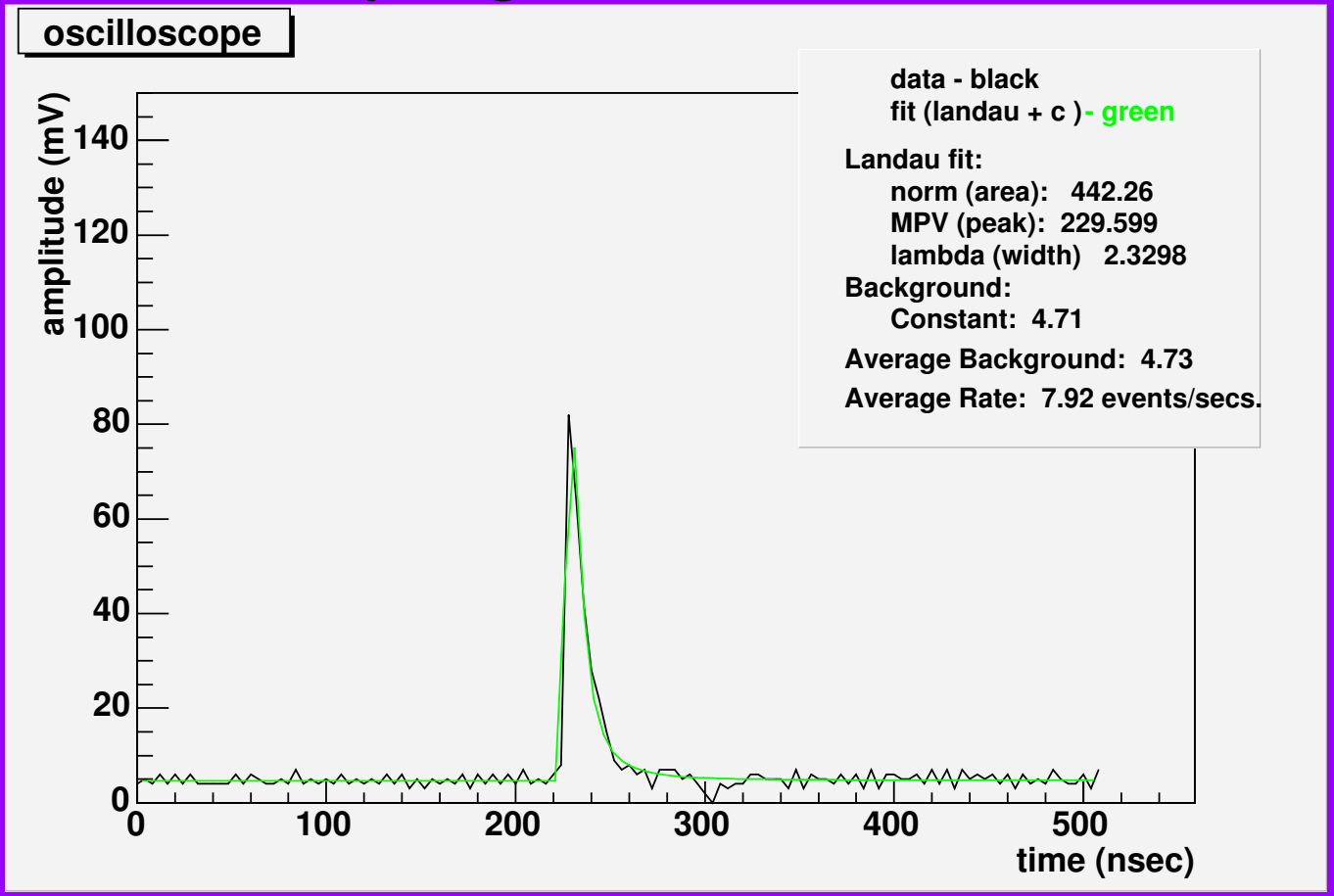
Root Interface

We have written a class which is essentially a wrapper to the IU software, and allows easy access to the fADC. With Root it is then easy to develop algorithms to use the information within the pulse shape.

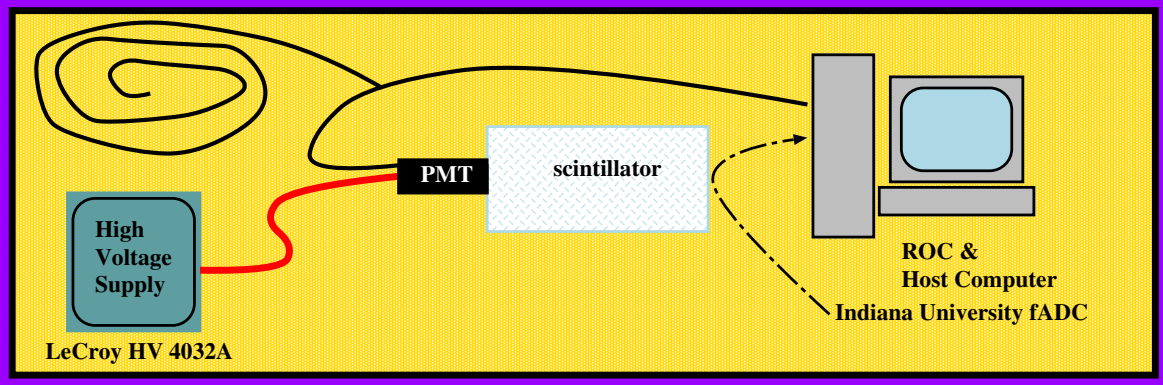
```
{
  int nEvents, nData, data[1000];
  gROOT->Reset();
  gSystem.Load("libFADCroot");
  TfADC *fadc = new TfADC();     // create fADC object, default setting

  fadc->SetTriggerLevel(20);     // set trigger level to 20 mV
  nEvents = adc1->NextEvent();  // read all events, return # of events
  for (int i=0;i<nEvent;i++)
      nData = adc1->GetData(i,data);  // event i into array data[]
}
```

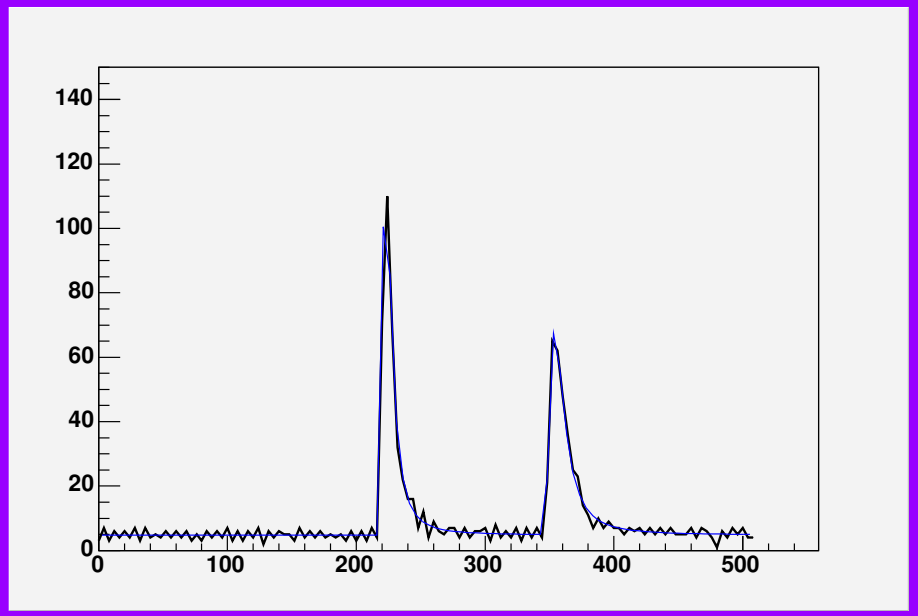
Curve fitting

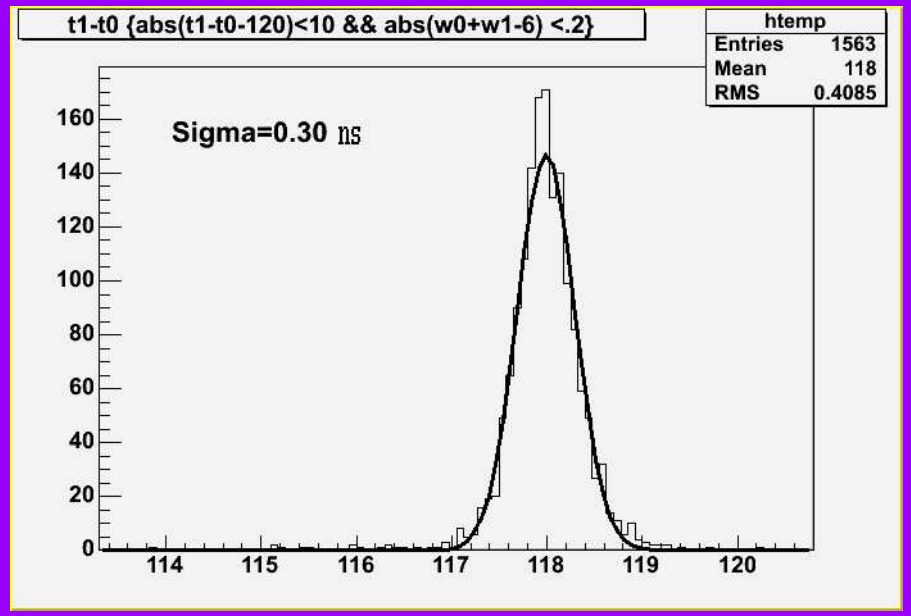
We can fit almost anything



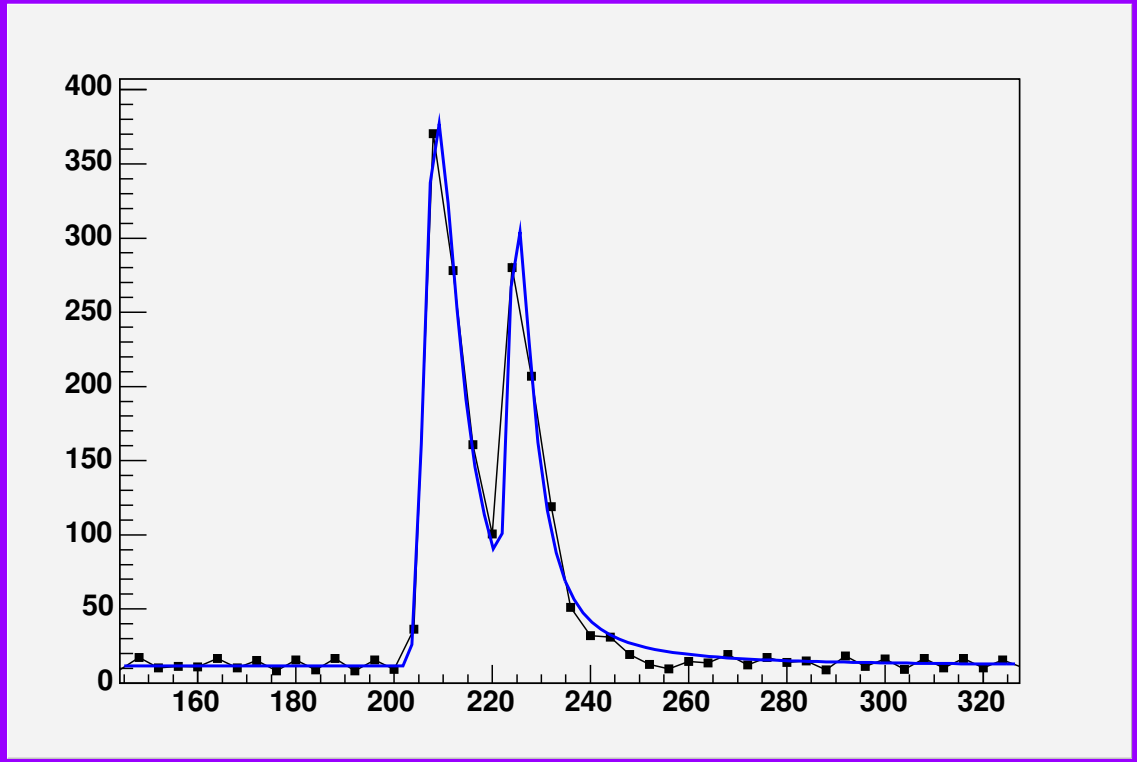
Timing - echo test







Peak Separation



We can separate and fit peak 20 ns apart - or better.

CODA Interface

We have written a library which can be called from the read-out-list (part of the *.crl file), and allows one to read these fADC as just another channel.

At present you can only read the sum of the pulse.

makelist test_fADC.crl native fadcCoda.o

```
begin trigger usrtrig

  variable data
  data = readFADC()
  open event type EVTYPE of BT_UI4
  output data
  close event

end trigger
```

Other DAQ Developments

We have been working with the CODA/DAQ group at JLab on a CAMAC-PCI to CODA interface. Since many people have "LabView" hardware, they could now run CODA.

We can actually make a hybrid system with CAMAC and fADC components!

Future Test

- **Add more feature to the coda ROL library - like reading time, or width or height**
- **Is there more information in the shape of the peak?**

Conclusion

The IU - fADC is a versatile board with a lot of possibilities. You can use it as a digital oscilloscope (one that is programmable and can fit anything), or as part of a CODA system, or as a small stand-alone DAQ system.

In GlueX the fADC will not be cards in the back of a Linux box, but dense boards in a crate. So we could transport the kilobyte/pulse to a Linux box someplace to be processed - but that would be a waste. We do have a powerful CPU in the crates, and we have the Xilex gate-array on the fADC board which could be more fully used.

There are a lot of innovative things which can be done in the front end of the DAQ system, but they will take time and playing - but they could lead to a more robust and better system for GlueX.