Control Software (CAT)

Introduction
USB Interface implementation

Calorimeter Electronics Upgrade Meeting

Frédéric Machefert
Wednesday 5th May, 2010

Introduction (I)

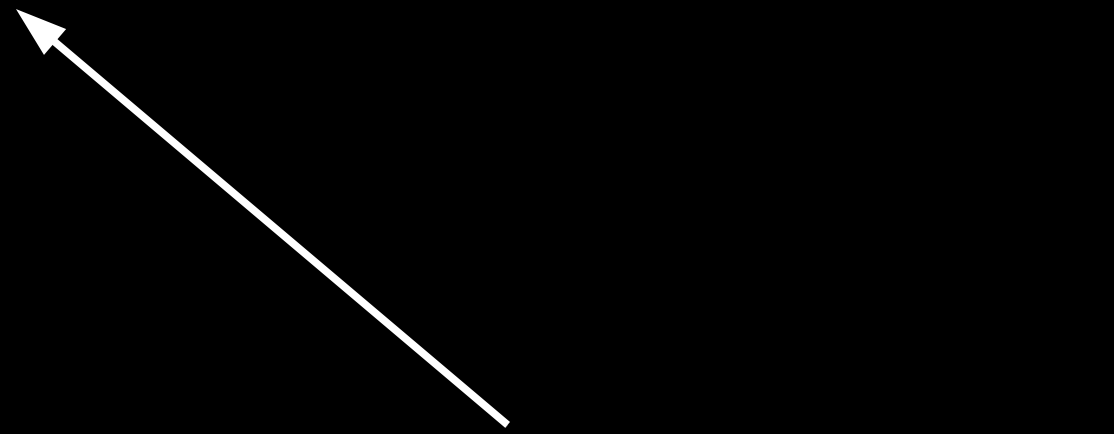
- LHCb uses PVSS for the control of the experiment
  - PVSS cannot rapidly control/monitor/run the electronics directly (slow)
- CAT is a software to
  - Configure our electronics
  - Monitor the electronics behaviour
  - Run automatized sequences (processes)
- CAT depends on other softwares
  - Which are LHCb standards
    - CMT (package management)
    - ROOT
    - (Python)
  - And graphical packages
    - WxWidgets (wxPython)
- CAT is made of two executables
  - A text program
  - A graphical interface

Introduction (II)

- What you need to / can do
  - Define your electronics (text file)
  - Configure it
    - You can apply configuration parameters to your electronics
      - SPECS
      - USB (see below)
    - You can read back the configuration
    - You may perform the configuration from a configuration file (text)
  - Run programs on the electronics
    - An automatic sequence is chosen and launched involving your electronics and the different protocols it requires (SPECS, USB)
    - You may define plots to store the information during the processing
  - CAT may show the results on a graphical windows
    - Useful for a rapid diagnostic / result
  - The data may be stored in directories (named from run numbers) for
    - archiving
    - Detailed analysis of the data
    - This last part is the one that uses ROOT (analysis framework)

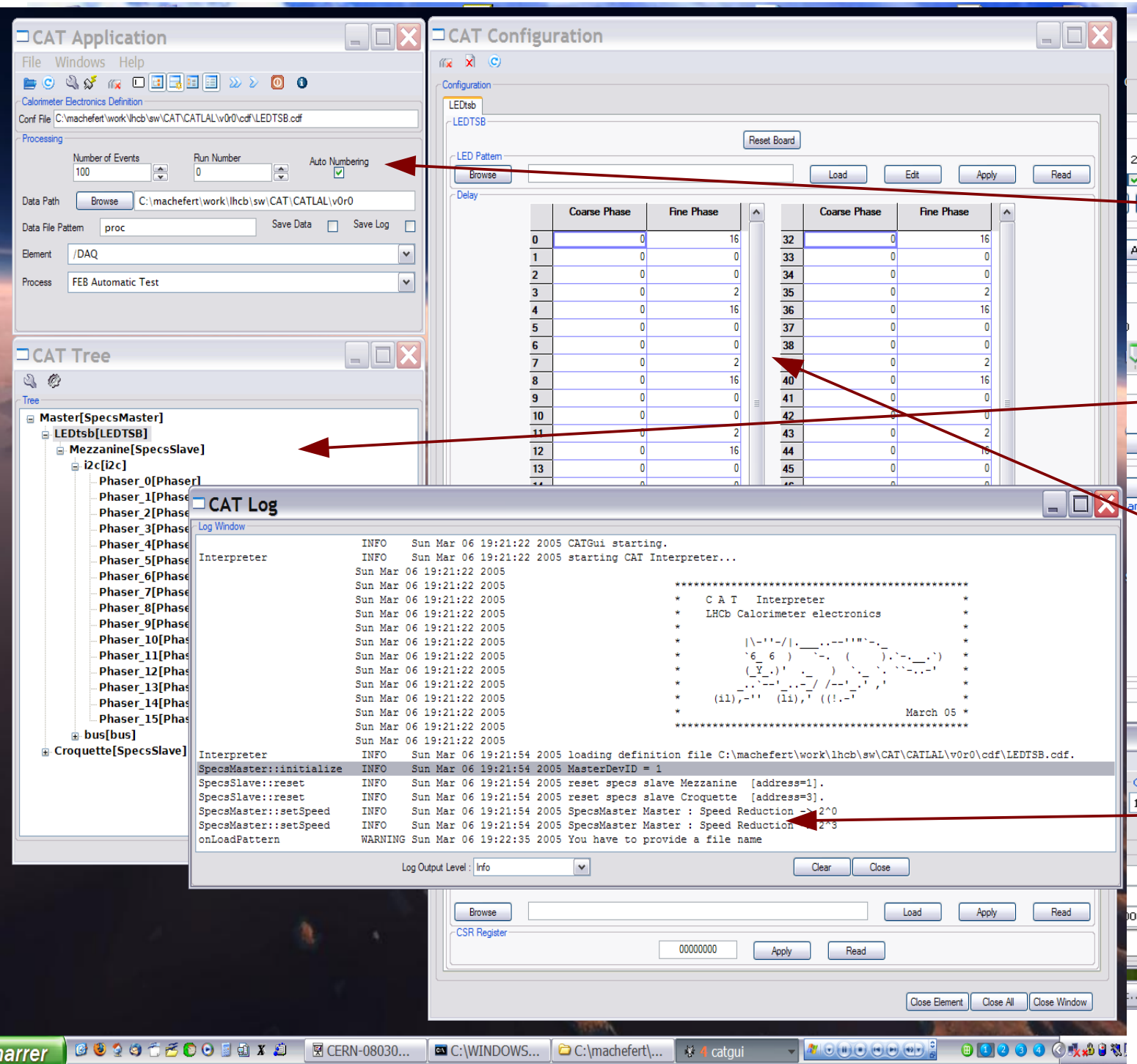
Text interface

```
716:~/LHCb/PyCAT/CAT/v1$ python -i python/cat.py -s
ProcDataBase              INFO    Thu Apr 29 16:06:43 2010 Building Processus DataBase.
Thu Apr 29 16:06:43 2010
Thu Apr 29 16:06:43 2010                          *********************************************
Thu Apr 29 16:06:43 2010                          *              C      A      T              *
Thu Apr 29 16:06:43 2010                          *                                           *
Thu Apr 29 16:06:43 2010                          *          |\-''-/|.___..--''"`-._          *
Thu Apr 29 16:06:43 2010                          *          `6_ 6  )   `-.  (     ).`-.__.`)   *
Thu Apr 29 16:06:43 2010                          *          (_Y_.)'  ._   )  `._ `. ``-..-'    *
Thu Apr 29 16:06:43 2010                          *        _..`--'_..-_  /  /--'_.' ,'          *
Thu Apr 29 16:06:43 2010                          *     (il),-''  (li),' ((!.-'                 *
Thu Apr 29 16:06:43 2010                          *********************************************
Thu Apr 29 16:06:43 2010
Thu Apr 29 16:06:43 2010 Application running on frederic@nb-machefert2 (Linux - 2.6.31-20-generic)
Application:loadHistor... INFO    Thu Apr 29 16:06:43 2010 Recovering last run number : 78
nb-machefert2[Computer] >
```



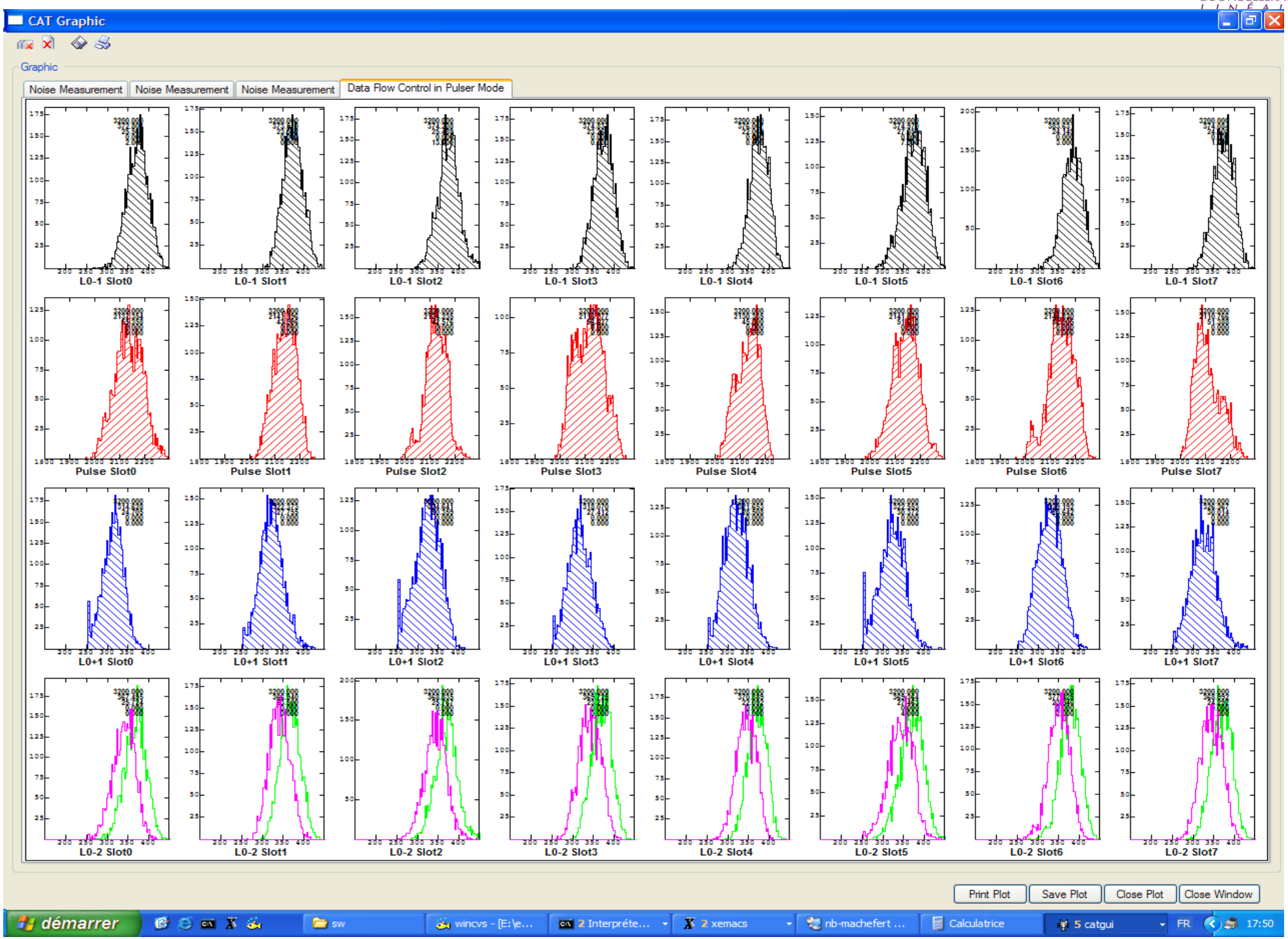
You can enter your commands here

Graphical Interface (I)

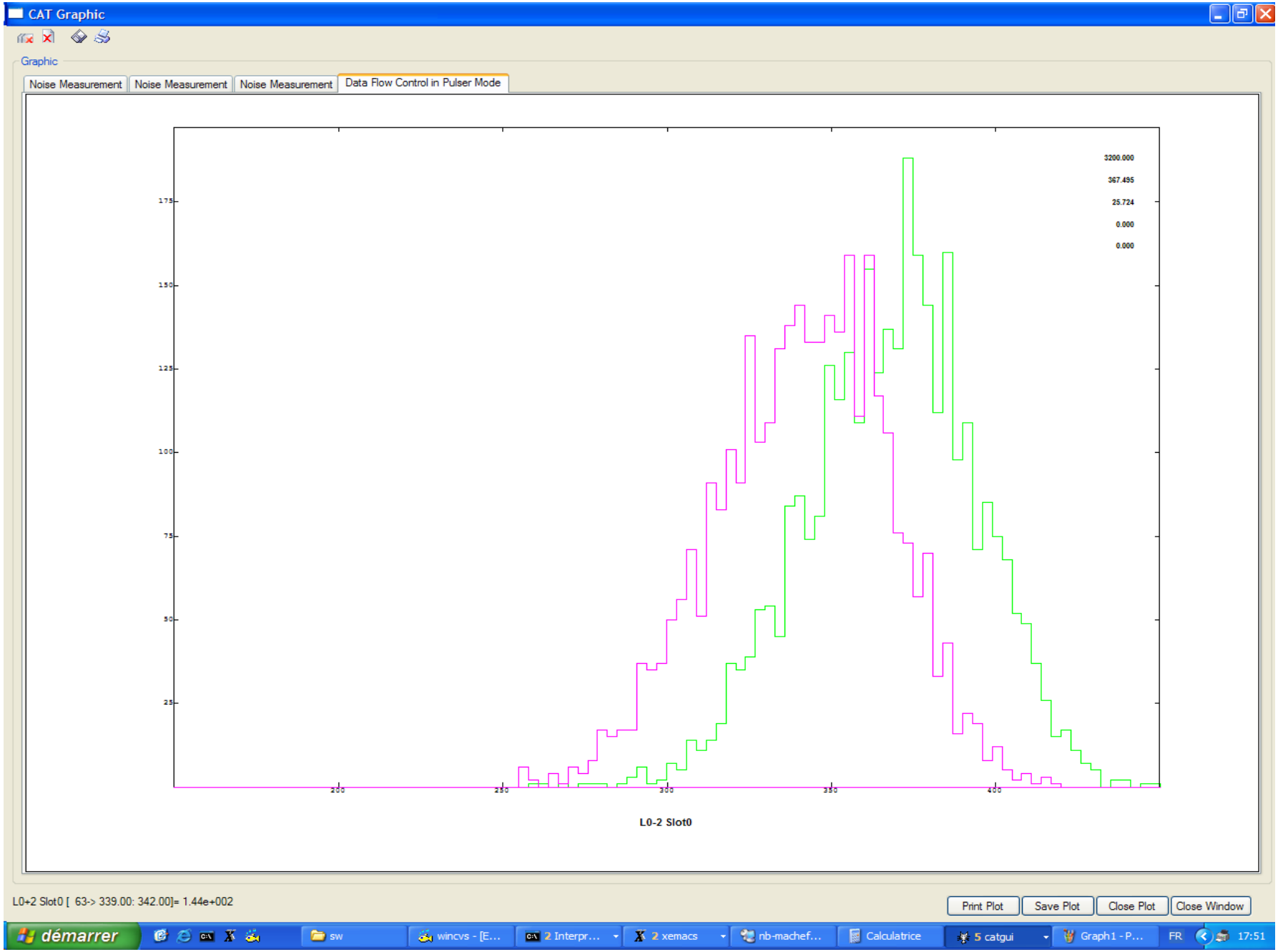


- 5 windows :
- Main control
  - Select processus to be run
  - Number of events
  - Data Storage
  - etc...
- Electronics tree
  - Full configuration loaded
- Configuration panel
  - LEDTSB example (FEB XCAL, CROC, Delay Chip, SPD/PRS board, Specs Slave and Master, DCU, etc...)
- Log window
  - Same output as in CATCmd (same code !)
- Graphic Output
  - Show histogram and previous run results

Graphical interface (II)



Graphical Interface (III)



Recent evolution of CAT

- CAT works well
  - but...
  - CAT was missing a USB interface
  - CAT interpreter is hard-coded
    - This is not flexible
    - This was hard to code
      - Could result in crash dump when users was making a “syntax error”
    - Could not perform any simple operation in the interpreter
      - “arithmetic” (2+2=4), “If” type conditions
  - CAT is heavy
    - The program (executable) uses everything
    - Includes the libraries coded by all the other groups
    - … and the bugs of all the groups
- I tried to resolve those problems in a new version
  - USB interface... see after
  - Interpreter → the new version of CAT is based on the python interpreter
    - Far easier for the person who codes (the interface is given for free)
  - Weight → new version is based on dynamic library loading
    - You include only what you need

USB Interfacing (I)

USB Interfacing (II)

USB Interfacing (I) : launching a sequence

```
521:~/LHCb/PyCAT/CAT/v1$ python -i python/cat.py -s
Linux system
ProcDataBase                INFO    Tue May  4 15:33:31 2010 Building Processus DataBase.
Tue May  4 15:33:31 2010
Tue May  4 15:33:31 2010                          ********************************************
Tue May  4 15:33:31 2010                          *              C      A      T             *
Tue May  4 15:33:31 2010                          *                                           *
Tue May  4 15:33:31 2010                          *          |\-''-/|.___.--''``-.            *
Tue May  4 15:33:31 2010                          *          `6_ 6  )   `-.  (     ).`-.__.`)   *
Tue May  4 15:33:31 2010                          *          (_Y_.)'  ._   )  `._ `. ``-..-'    *
Tue May  4 15:33:31 2010                          *        _..`--'_..-_/  /--'_.' ,'            *
Tue May  4 15:33:31 2010                          *     (il),-''  (li),' ((!.-'                 *
Tue May  4 15:33:31 2010                          ********************************************
Tue May  4 15:33:31 2010
Tue May  4 15:33:31 2010 Application running on frederic@nb-machefert2 (Linux - 2.6.31-20-generic)
Application:loadHistor... INFO    Tue May  4 15:33:31 2010 Recovering last run number : 86
nb-machefert2[Computer] > from libCATCore import * # import library in python
nb-machefert2[Computer] > loadDll("CATCore")       # import library in C++
DLLMgr::load                INFO    Tue May  4 15:33:38 2010 Loading dll CATCore.
CATCore                     INFO    Tue May  4 15:33:38 2010 Symbols found in dll CATCore: 6 Element(s) - 1 Processus.
DLL::print                  INFO    Tue May  4 15:33:38 2010 ================================================================================
DLL::print                  INFO    Tue May  4 15:33:38 2010 Element                    Description
DLL::print                  INFO    Tue May  4 15:33:38 2010               SpecsElement                                  Generic Specs Element
DLL::print                  INFO    Tue May  4 15:33:38 2010                   SpecsI2c                                  Specs I2c bus element
DLL::print                  INFO    Tue May  4 15:33:38 2010           SpecsParallelBus                          Specs parallel bus element
DLL::print                  INFO    Tue May  4 15:33:38 2010                SpecsMaster                 Specs Master interface description
DLL::print                  INFO    Tue May  4 15:33:38 2010                 SpecsSlave                  Specs Slave interface description
DLL::print                  INFO    Tue May  4 15:33:38 2010             UsbFTInterface                    USB FT245 interface description
DLL::print                  INFO    Tue May  4 15:33:38 2010 ================================================================================
DLL::print                  INFO    Tue May  4 15:33:38 2010 Processus                Target Type
DLL::print                  INFO    Tue May  4 15:33:38 2010        UsbFTInterfaceTest          UsbFTInterface
DLL::print                  INFO    Tue May  4 15:33:38 2010 ================================================================================
nb-machefert2[Computer] > create("usb","UsbFTInterface")
usb                         INFO    Tue May  4 15:33:45 2010 Todo : time efficiency by creating the buffer once for all ?
nb-machefert2[Computer] > usb=cat.computer().child("usb")
nb-machefert2[Computer] > usb.setSerialNum("Wilky_05")
nb-machefert2[Computer] > usb.setDeviceDesc("Carte Test_Wilky")
nb-machefert2[Computer] > usb.setWordSize(WordSize.U32)
nb-machefert2[Computer] > usb.init()
**********************************************
1 usb device(s) found
**********************************************
UsbFTInterface::init        INFO    Tue May  4 15:33:45 2010 Device with device description Carte Test_Wilky found.
UsbFTInterface::init        INFO    Tue May  4 15:33:45 2010 Try to open device with serial number Wilky_05.
**********************************************
Connected to the device with Serial Number Wilky_05
Opened with handle : 09A8EA58 - internal ID : 1
**********************************************
UsbFTInterface::init        INFO    Tue May  4 15:33:45 2010 Device opened with UsbId 1.
Latency timer : 2 ms
Device type : FT2XXBM
Desc. : Carte Test_Wilky
S/N : Wilky_05
Rx Queue : 0 bytes left
Tx Queue : 0 bytes left
Events : 0

UsbFTInterface::init        INFO    Tue May  4 15:33:45 2010 Usb device intialization done.

Reading data from EEPROM...Ok

UsbFTInterface::init        INFO    Tue May  4 15:33:45 2010 EPROM data :
UsbFTInterface::init        INFO    Tue May  4 15:33:45 2010   Device Description -> Carte Test_Wilky
UsbFTInterface::init        INFO    Tue May  4 15:33:45 2010   Serial Number      -> Wilky_05
<libCATKernel.StatusCode object at 0x9a6cdbc>
nb-machefert2[Computer] > r=[]
nb-machefert2[Computer] > w=range(int('12345678',16),int('12345678',16)+10)
nb-machefert2[Computer] > print w
[305419896, 305419897, 305419898, 305419899, 305419900, 305419901, 305419902, 305419903, 305419904, 305419905]
nb-machefert2[Computer] > usb.write(8,w)
<libCATKernel.StatusCode object at 0x9a6cd84>
nb-machefert2[Computer] > usb.read(8,20,r)
<libCATKernel.StatusCode object at 0x9a6cdbc>
nb-machefert2[Computer] > print r
[305419896, 305419897, 305419898, 305419899, 305419900, 305419901, 305419902, 305419903, 305419904, 305419905, 303174162, 303174162, 303174162, 303174162, 303174162, 303174162, 303174162, 303174162, 303174162, 303174162]
nb-machefert2[Computer] > usb.close()
UsbFTInterface::close       INFO    Tue May  4 15:35:49 2010 Close USB device usb [id=1]
nb-machefert2[Computer] >
```

Load libraries

Define the USB device

Initialization

Write/Read using the interpreter

USB Interfacing (I)

```
Thu Apr 29 16:06:43 2010                    *                                          *
Thu Apr 29 16:06:43 2010                    *          |\-''-/|.___..--''"`-._          *
Thu Apr 29 16:06:43 2010                    *          `6_ 6  )   `-.  (     ).`-.__.`)  *
Thu Apr 29 16:06:43 2010                    *          (_Y_.)'  ._   )  `._ `. ``-..-'   *
Thu Apr 29 16:06:43 2010                    *        _..`--'_..-_  /  /--'_.' ,'         *
Thu Apr 29 16:06:43 2010                    *     (il),-''  (li),' ((!.-'                *
Thu Apr 29 16:06:43 2010                    **********************************************
Thu Apr 29 16:06:43 2010
Thu Apr 29 16:06:43 2010 Application running on frederic@nb-machefert2 (Linux - 2.6.31-20-generic)
Application:loadHistor... INFO    Thu Apr 29 16:06:43 2010 Recovering last run number : 78
nb-machefert2[Computer] > load("usb.py")
DLLMgr::load              INFO    Thu Apr 29 16:07:56 2010 Loading dll CATCore.
CATCore                   INFO    Thu Apr 29 16:07:56 2010 Symbols found in dll CATCore: 6 Element(s) - 1 Processus.
DLL::print                INFO    Thu Apr 29 16:07:56 2010 ================================================================================================
DLL::print                INFO    Thu Apr 29 16:07:56 2010 Element                    Description
DLL::print                INFO    Thu Apr 29 16:07:56 2010              SpecsElement                                        Generic Specs Element
DLL::print                INFO    Thu Apr 29 16:07:56 2010                 SpecsI2c                                         Specs I2c bus element
DLL::print                INFO    Thu Apr 29 16:07:56 2010          SpecsParallelBus                                  Specs parallel bus element
DLL::print                INFO    Thu Apr 29 16:07:56 2010               SpecsMaster                          Specs Master interface description
DLL::print                INFO    Thu Apr 29 16:07:56 2010                SpecsSlave                           Specs Slave interface description
DLL::print                INFO    Thu Apr 29 16:07:56 2010            UsbFTInterface                            USB FT245 interface description
DLL::print                INFO    Thu Apr 29 16:07:56 2010 ================================================================================================
DLL::print                INFO    Thu Apr 29 16:07:56 2010 Processus               Target Type
DLL::print                INFO    Thu Apr 29 16:07:56 2010        UsbFTInterfaceTest           UsbFTInterface
DLL::print                INFO    Thu Apr 29 16:07:56 2010 ================================================================================================
usb                       INFO    Thu Apr 29 16:07:56 2010 Todo : time efficiency by creating the buffer once for all ?
**********************************************
1 usb device(s) found
**********************************************
UsbFTInterface::init      INFO    Thu Apr 29 16:07:56 2010 Device with device description Carte Test_Wilky found.
UsbFTInterface::init      INFO    Thu Apr 29 16:07:56 2010 Try to open device with serial number Wilky_05.
**********************************************
Connected to the device with Serial Number Wilky_05
Opened with handle : 0B2CB948 - internal ID : 1
**********************************************
UsbFTInterface::init      INFO    Thu Apr 29 16:07:56 2010 Device opened with UsbId 1.
Latency timer : 2 ms
Device type : FT2XXBM
Desc. : Carte Test_Wilky
S/N : Wilky_05
Rx Queue : 0 bytes left
Tx Queue : 0 bytes left
Events : 0

UsbFTInterface::init      INFO    Thu Apr 29 16:07:56 2010 Usb device intialization done.

Reading data from EEPROM...Ok

UsbFTInterface::init      INFO    Thu Apr 29 16:07:56 2010 EPROM data :
UsbFTInterface::init      INFO    Thu Apr 29 16:07:56 2010   Device Description -> Carte Test_Wilky
UsbFTInterface::init      INFO    Thu Apr 29 16:07:56 2010   Serial Number      -> Wilky_05
cd: 1: can't cd to data/Run_79
Application::makeDir      INFO    Thu Apr 29 16:07:57 2010 Directory data/Run_79 created.
UsbFTInterfaceTest        INFO    Thu Apr 29 16:07:57 2010 ================================================================
UsbFTInterfaceTest        INFO    Thu Apr 29 16:07:57 2010    * UsbFTInterfaceTest *
Application::loop         INFO    Thu Apr 29 16:08:02 2010 [Processing evt 1000]
UsbFTInterfaceTest        INFO    Thu Apr 29 16:08:02 2010 ----------------------------------------------------------------
finalize                  INFO    Thu Apr 29 16:08:02 2010 Processed Run Number       : 79
finalize                  INFO    Thu Apr 29 16:08:02 2010 Number of events processed : 1000
finalize                  INFO    Thu Apr 29 16:08:02 2010 Number of Errors           : 0
finalize                  INFO    Thu Apr 29 16:08:02 2010 Number of App errors       : 0
finalize                  INFO    Thu Apr 29 16:08:02 2010 Elapsed time               : 0.880000
UsbFTInterfaceTest        INFO    Thu Apr 29 16:08:02 2010 ----------------------------------------------------------------
Data::print               INFO    Thu Apr 29 16:08:02 2010 Number of data streams : 4
Data::print               INFO    Thu Apr 29 16:08:02 2010  (  0)  TimeWrite -                          Time to write (    1000)
Data::print               INFO    Thu Apr 29 16:08:02 2010  (  1)   TimeRead -                           Time to read (    1000)
Data::print               INFO    Thu Apr 29 16:08:02 2010  (  2) valueWrite -                 Error - Written values (       0)
Data::print               INFO    Thu Apr 29 16:08:02 2010  (  3)  valueRead -                    Error - Read values (       0)
UsbFTInterfaceTest        INFO    Thu Apr 29 16:08:02 2010 ----------------------------------------------------------------
Application::svcRunning   INFO    Thu Apr 29 16:08:02 2010 Processus UsbFTInterfaceTest 'UsbFTInterface test' completed [1000 events]
usb[UsbFTInterface] >
```

This time an automatized Run was launched :

- 256 bytes
- 1000 read/write Op.
- ~3s

You can store “ntuple” To analyse “online” from the interpreter

USB Interfacing (I)

```
Connected to the device with Serial Number Wilky_05
Opened with handle : 0983D990 - internal ID : 1
*********************************************
UsbFTInterface::init      INFO    Thu Apr 29 16:16:20 2010 Device opened with UsbId 1.
Latency timer : 2 ms
Device type : FT2XXBM
Device : Carte Test_Wilky
Serial : Wilky_05
Rx Queue : 0 bytes left
Tx Queue : 0 bytes left
Events : 0

UsbFTInterface::init      INFO    Thu Apr 29 16:16:20 2010 Usb device intialization done.

Reading data from EEPROM...Ok

UsbFTInterface::init      INFO    Thu Apr 29 16:16:20 2010 EPROM data :
UsbFTInterface::init      INFO    Thu Apr 29 16:16:20 2010   Device Description -> Carte Test_Wilky
UsbFTInterface::init      INFO    Thu Apr 29 16:16:20 2010   Serial Number      -> Wilky_05
<libCATKernel.StatusCode object at 0x98163ac>
nb-machefert2[Computer] > cd("usb")
usb[UsbFTInterface] > p=proc("UsbFTInterfaceTest")
usb[UsbFTInterface] > p.setAddress(8)
<libCATKernel.StatusCode object at 0x9816374>
usb[UsbFTInterface] > p.setParam(64,100.,20.) # 64 mots de 32 bits - mean=100 - sigma=20
<libCATKernel.StatusCode object at 0x98163ac>
usb[UsbFTInterface] > cat.run("UsbFTInterfaceTest",obj(),1000)
cd: 1: can't cd to data/Run_85
Application::makeDir      INFO    Thu Apr 29 16:16:27 2010 Directory data/Run_85 created.
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:27 2010 ================================================================================
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:27 2010     * UsbFTInterfaceTest *
Application::loop         INFO    Thu Apr 29 16:16:31 2010 [Processing evt 1000]
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:31 2010 --------------------------------------------------------------------------------
finalize                  INFO    Thu Apr 29 16:16:31 2010 Processed Run Number       : 85
finalize                  INFO    Thu Apr 29 16:16:31 2010 Number of events processed : 1000
finalize                  INFO    Thu Apr 29 16:16:31 2010 Number of Errors           : 0
finalize                  INFO    Thu Apr 29 16:16:31 2010 Number of App errors       : 0
finalize                  INFO    Thu Apr 29 16:16:31 2010 Elapsed time               : 1.050000
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:31 2010 --------------------------------------------------------------------------------
Data::print               INFO    Thu Apr 29 16:16:31 2010 Number of data streams : 4
Data::print               INFO    Thu Apr 29 16:16:31 2010  (  0)  TimeWrite -                          Time to write (    1000)
Data::print               INFO    Thu Apr 29 16:16:31 2010  (  1)   TimeRead -                           Time to read (    1000)
Data::print               INFO    Thu Apr 29 16:16:31 2010  (  2) valueWrite -                 Error - Written values (       0)
Data::print               INFO    Thu Apr 29 16:16:31 2010  (  3)  valueRead -                    Error - Read values (       0)
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:31 2010 --------------------------------------------------------------------------------
Application::svcRunning   INFO    Thu Apr 29 16:16:31 2010 Processus UsbFTInterfaceTest 'UsbFTInterface test' completed [1000 events]
<libCATKernel.StatusCode object at 0x981648c>
usb[UsbFTInterface] > p.setParam(68,100.,20.)
<libCATKernel.StatusCode object at 0x9816454>
usb[UsbFTInterface] > cat.run("UsbFTInterfaceTest",obj(),1)
cd: 1: can't cd to data/Run_86
Application::makeDir      INFO    Thu Apr 29 16:16:56 2010 Directory data/Run_86 created.
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:56 2010 ================================================================================
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:56 2010     * UsbFTInterfaceTest *
UsbFTInterface::usbWri... WARNING Thu Apr 29 16:16:56 2010 256 byte(s) written on USB interface usb out of 272 bytes to be sent.
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:56 2010 --------------------------------------------------------------------------------
finalize                  INFO    Thu Apr 29 16:16:56 2010 Processed Run Number       : 86
finalize                  INFO    Thu Apr 29 16:16:56 2010 Number of events processed : 1
finalize                  INFO    Thu Apr 29 16:16:56 2010 Number of Errors           : 0
finalize                  INFO    Thu Apr 29 16:16:56 2010 Number of App errors       : 0
finalize                  INFO    Thu Apr 29 16:16:56 2010 Elapsed time               : 0.000000
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:56 2010 --------------------------------------------------------------------------------
Data::print               INFO    Thu Apr 29 16:16:56 2010 Number of data streams : 4
Data::print               INFO    Thu Apr 29 16:16:56 2010  (  0)  TimeWrite -                          Time to write (       1)
Data::print               INFO    Thu Apr 29 16:16:56 2010  (  1)   TimeRead -                           Time to read (       1)
Data::print               INFO    Thu Apr 29 16:16:56 2010  (  2) valueWrite -                 Error - Written values (       0)
Data::print               INFO    Thu Apr 29 16:16:56 2010  (  3)  valueRead -                    Error - Read values (       0)
UsbFTInterfaceTest        INFO    Thu Apr 29 16:16:56 2010 --------------------------------------------------------------------------------
Application::svcRunning   INFO    Thu Apr 29 16:16:56 2010 Processus UsbFTInterfaceTest 'UsbFTInterface test' completed [1 events]
<libCATKernel.StatusCode object at 0x9816374>
usb[UsbFTInterface] >
```

Using the interpreter it is
Easy to tune a processus

Here I ask to read 272 bytes

I tried to write more
Than 256 in a frame

Test Sequence Initialisation

- CAT is made to be as simply as possible for the user
- Each Sequence is divided in three steps
  - Initialisation
  - Execution
  - termination

Open the root file

```
36 //=============================================================
37 StatusCode UsbFTInterfaceTest::initialize ( ) {
38   openRootFile ();
39   m_write=new TH1D( "Write" , "Write" , 32 , 0. , 256. );
40   m_read =new TH1D( "Read"  , "Read"  , 32 , 0. , 256. );
41   m_error=new TH1D( "Error" , "Error" , 32 , 0. , 256. );
42   m_rnd =new TRandom();
43   addDataStream("TimeWrite","Time to write");
44   addDataStream("TimeRead","Time to read");
45   addDataStream("valueWrite","Error - Written values");
46   addDataStream("valueRead","Error - Read values");
47   return StatusCode::SUCCESS;
48 }
49
50 //=============================================================
```

Define Histo

Define data to be stored (vectors)

Test Sequence execution

```
51  //  Virtual function execute
52  //=============================================================================
53  StatusCode UsbFTInterfaceTest::execute ( ) {
54
55    // prepare vector to be written
56    std::vector<U32> write,read;
57    write.reserve(m_size);
58    read.reserve(m_size);
59    for (unsigned int i=0; i<m_size; ++i){
60      write.push_back((int)((U32)(m_rnd->Gaus(m_mean,m_sigma))));
61    }
62
63    // fetch the usb interface
64    UsbFTInterface *usb=dynamic_cast<UsbFTInterface*>( element() );
65    usb->setWordSize(UsbFTInterface::WS_DWord);
66
67    // write / read and measure times
68    float tw,tr;
69    clock_t start;
70    start=clock();
71    usb->usbWrite(m_address,write);
72    tw=elapsedTime(start);
73    start=clock();
74    usb->usbRead(m_address,m_size,read);
75    tr=elapsedTime(start);
76
77    dataFill("TimeWrite",tw);
78    dataFill("TimeRead",tr);
79
80    // store values and check errors
81    float w,r;
82    for (unsigned int i=0; i<m_size; ++i){
83      w=write[i];
84      r=read[i];
85      m_write->Fill(w);
86      m_read->Fill(r);
87      if (w!=r){
88        m_error->Fill(w);
89        dataFill("ValuesWrite",w);
90        dataFill("ValuesRead",r);
91      }
92    }
93    return StatusCode::SUCCESS;
94  }
```

Usb interfacing

Store the parameters you want to keep

Fill the histograms

The rest is what you want you code to do...

Test Sequence Termination

- You may use this part for
  - Specific printout
  - Evaluating counters on the run
  - Etc...
- But the rest is done for you

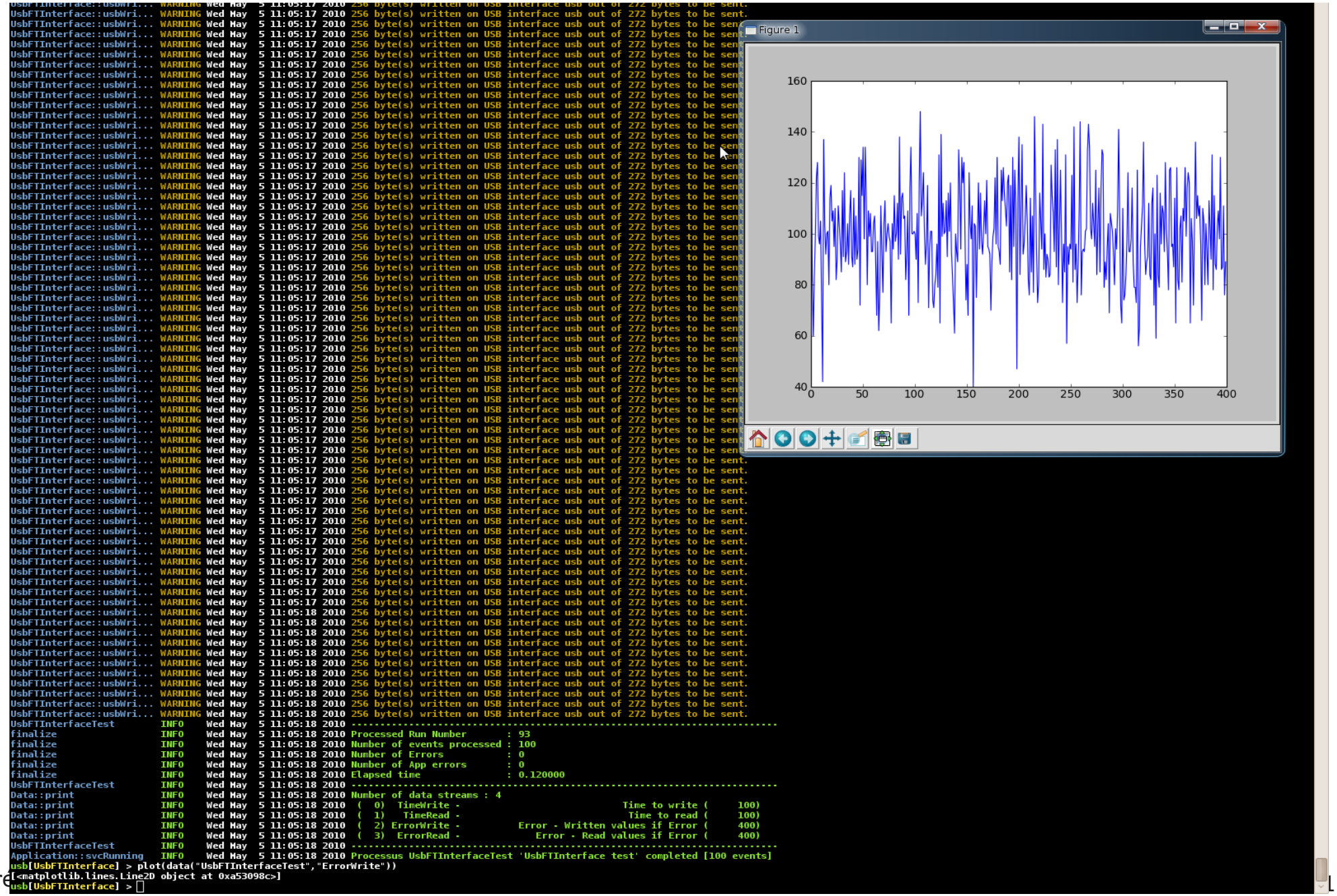
Close the Root file
Will soon disappear → done for you)

```
//=============================================================================
//  Virtual function finalize
//=============================================================================
StatusCode UsbFTInterfaceTest::finalize ( ) {
  // Plot settings

  closeRootFile(); // Deletes the histos -> do not make it yourself !!!
  return StatusCode::SUCCESS;
}
```

Plot from the interpreter

- You may directly access the data you stored in the vectors



Conclusion

- CAT is already extensively used and works well
  - I am ready to give a hand to install it on any system
- I started to work on a new version
- The USB interfacing exists both in the old and the new version
  - Don't have to wait to start working
- The graphical interface of the new version is still preliminary
  - No specific difficulties (time...)
  - Should permit the user to make simpler graphical interfaces
- The python interpreter works well
  - Allows a direct access from the interpreter to ALL the C++ commands
  - Don't need to “code” your own interpreter (was a nightmare)
- The USB interface works fine in Linux
  - It should work on windows...
    - But I still experience some difficulties (rapid attempt)