EDIT2013 Documentation

Release 0.1.0

Shota Takahashi

June 06, 2016

Contents

The Document for DAQ and ROOT macros used in EDIT2013 NEU course

Author Shota TAKAHASHI

Created 2013/03/12

Modified 2013/03/12

CHAPTER 1

How to obtain samples

Just download “sample.zip” and expand it in any place you want.

CHAPTER 2

What's inside

daq/	DAQ programs
data/	data
macros/	ROOT macros
plots/	save plots here if neccessary

CHAPTER 3

How to take data

DAQ programs are in `daq/` directory. Move to this directory first. Then execute program named **`adc`** or **`multiadc`**.

You need to specify `NEVENT` (event number) and `OFN` (output filename) as arguments. Output file is created in space-separated-text format.

Sample command 1

```
$ cd daq
$ ./adc NEVENT OFN (additional info).
```

Sample command 2 If you run without any argument, it will print usage.

```
$ ./adc
## --> usage will be printed
```

ADVANCED You can also set additional arguments if you want, like below

```
$ ./adc NEVENT OFN MPPC_ID BIAS_VOLTAGE
```

3.1 Save data

It is **REALLY** important to name data with easy-to-remember-and-handle convention. (If you named `test1.txt` , `test2.txt` , ... and so on,

you will get depressed when you came back to check data next week.)

My favorite way is to separate data by day. (it depends by purpose, though) I prefer `OFN` to be named like `../data/YMD/ANYTHING%03d.txt` , where

`YMD`	date (YYYYMMDD format)
`ANYTHING`	any words you want
`%03d`	run number, this makes easy to use ROOT macros below.

3.2 Overwrite data

DAQ program has non-overwriting feature to prevent **ACCIDENTAL** overwriting. In case of overwriting, remove that file first.

```
$ ./adc 100 SAME_NAMED_FILE
## --> Error: 'SAME_NAMED_FILE' already exists.

$ rm SAME_NAMED_FILE
$ ./adc 100 SAME_NAMED_FILE
```

3.3 Change ADC channnels

Number of ADCs and its channels are defined as variable(array) named `ModId` and `ChId` in `adc.cc` (`multiadc.cc`). Modify these numbers and re-compile.

Sample command

```
$ emacs adc.cc
$ ...(edit adc.cc)
$ make
```

3.4 Install CAMAC driver

If you restarted PC for some reason, you need to install CAMAC driver before excecuting DAQ. We use `camdrv` for CAMAC driver. Its source is in `/opt/hep/kinoko/drv/camdrv/`.

Sample command

```
$ cd /opt/hep/kinoko/drv/camdrv
$ su (ask passwd to lecturer)
# make install
# dmesg
# exit
```

CHAPTER 4

How to make plots

ROOT macro demos are in `macors` directory. Move to this directory first. Start ROOT CINT and load MACRO. Then execute as samples below.

*** IMPORTANT NOTICE *** These are just simple demos. I encourage you to look into the codes and improve it, or make your own. (If you know ROOT, you can do it =D)

Sample command 1 mppcTest.C to check one histogram

The function `histText()` is defined to return TH1D*. You need to create TH1D* and draw histogram.

```
$ root
root[] > .L mppcTest.C
root[] > TH1D *h1 = histText("h1", "../data/20130305/mppctest001.txt")
root[] > h1->Draw()
```

Sample command 2 adcCalibration.C to check three adc data at one time

The function "histText" is defined to return TCanvas*. You need to create TCanvas*. (Canvas will be drawn automatically)

```
$ root
root[] > .L adcCalibration.C
root[] > TCanvas *c1 = histText(32)
```

Sample command 3 tracker.C for eventdisplay

You don't need to load tracker.C, just type

```
$ root tracker.C
```

or, in case you want to start from certain run number,

```
$ 'root tracker.C(3)'
```

4.1 Change branch descriptor

All macros use method `TTree::ReadFile(FILENAME, BRANCH_DESCRIPTOR)()` to create TTree from text file. If you changed orders or number of columns of output text format, modify `BRANCH_DESCRIPTOR` as needed.

CHAPTER 5

How to use USB

Unfortunately, PCs used in EDIT2013 does not mount USB automatically. So please mount USB manually.

Sample command

```
$ dmesg
$ su (ask passwd to lecturer)
# mount /dev/sdc1 /mnt/usb (or /media/usb)
# ls /mnt/usb
# ... (rsync or cp files)
# umount /mnt/usb
# exit
```

CHAPTER 6

Indices and tables

- genindex
- modindex
- search