



Common code for TCT

Sharing is caring

Why common code?

- We all start from same file format
 - Given by Particulars DAQ system
- A common framework saves a lot of time
- It makes results comparable
- Eases the usage for newcomers
- Write manual/twiki/... together → less work
- No re-invention of the wheel
- Use improvements from everyone
- User driven → implement what is needed by the community

GitHub.com

- “GitHub is a Web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.”
- Git → Command line tool for revision control
- GitHub → where the repo is
- Create a local copy, develop a feature, pull request → revision by owner of the repo, accept pull request

GitHub



CMake

- User CMake to create make files
- Makes framework platform independent
- Works on Unix, Windows, Mac!
- Easy inclusion of external libraries (sort of ;))
 - Include Particulars DAQ lib
 - Include LeCroyReader lib
 - Include ROOT



TCT analysis framework

- Compile (rather than running macros)
- Steer analysis using configuration files
- Get it from GitHub
<https://github.com/DESY-FH-ELab/TCT-analysis>

either: `> git clone url`
to get your local copy of the master and use it

or “fork” from the master to develop new features



TCT analysis framework

DESY-FH-ELab / TCT-analysis

Unwatch 2

Star 0

Fork 4

a common analysis framework for (e)TCT pulses — Edit

103 commits

1 branch

1 release

3 contributors



Branch: master

TCT-analysis / +



Merge pull request #2 from garankonic/master

TCThenna authored 25 days ago

latest commit d3c305b30b

build	added out-of-source build directory	a year ago
cmake	Small remarks	2 months ago
doc/doxygen	removed "EUDAQ" legacy	a year ago
include	Corrections after code review	26 days ago
results	added results folder	a year ago
src	Fixes	25 days ago
testanalysis	Fixes	25 days ago
testdata/S57/295K/500V	added testfiles and installation instructions	a year ago
testsensor	add testsensor	a year ago
.gitignore	cosmetics	a year ago
CMakeLists.txt	Adding CMAKE options for external libraries	2 months ago
LICENSE	added licence file LICENCE	a year ago
README.md	README.md updated	29 days ago

Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

Settings

HTTPS clone URL

https://github.c

You can clone with HTTPS, SSH, or Subversion.

Download ZIP



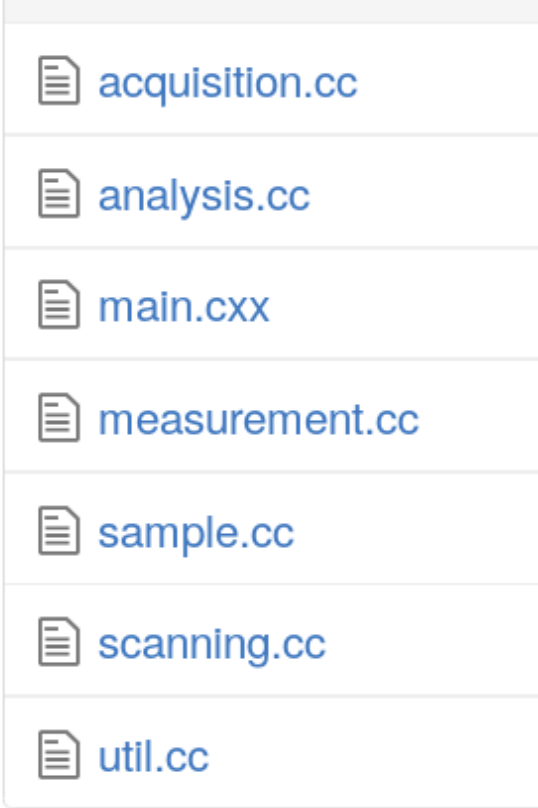
Structure

- /build: build project from here
 - > cmake ..
 - > make install
- /src: where the code is
- /include: where the headers are
- /external: put external code/libraries here
- Two config-files, “sample card” and “analysis-card”:
 - 1) For the sensor (sample name, width, doping, ...)
 - 2) For the analysis (file format, optical axis, cuts, ...)
 - Easy to share analysis without c&p macros
 - Re-do exact analysis with own data
- Output in ROOT files



/src

- Acquisition:
 - Reads in the acqs from various formats
 - knows what the scope knows:
noise, amplitude, ... of **single** acq
- Analysis:
 - can filter single acqs
 - finds signal, creates average
 - reads analysis card
- Measurement:
 - prepares vector of acqs for analysis
- Sample: helper class to read the sample card
- Scanning:
 - Analysis of files from particular DAQ
 - Make use of lib from particular



acquisition.cc
analysis.cc
main.cxx
measurement.cc
sample.cc
scanning.cc
util.cc

/src

For single acqs

- Acquisition:
 - Reads in the acqs from various formats
 - knows what the scope knows:
noise, amplitude, ... of **single** acq
- Analysis:
 - can filter single acqs
 - finds signal, creates average
 - reads analysis card
- Measurement:
 - prepares vector of acqs for analysis
- Sample: helper class to read the sample card

For avg'ed acqs

- Scanning:
 - Analysis of files from particular DAQ
 - Make use of lib from particular

acquisition.cc

analysis.cc

main.cxx

measurement.cc

sample.cc

scanning.cc

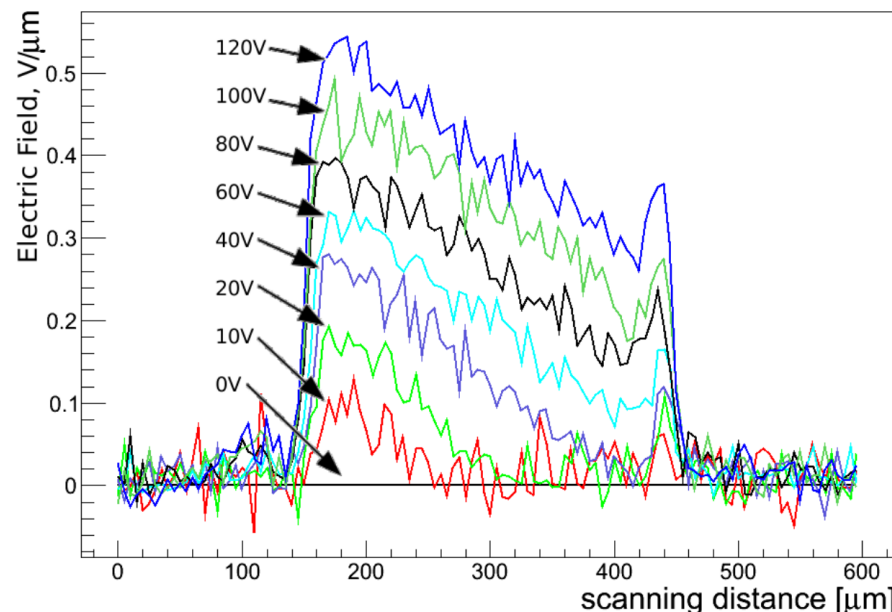
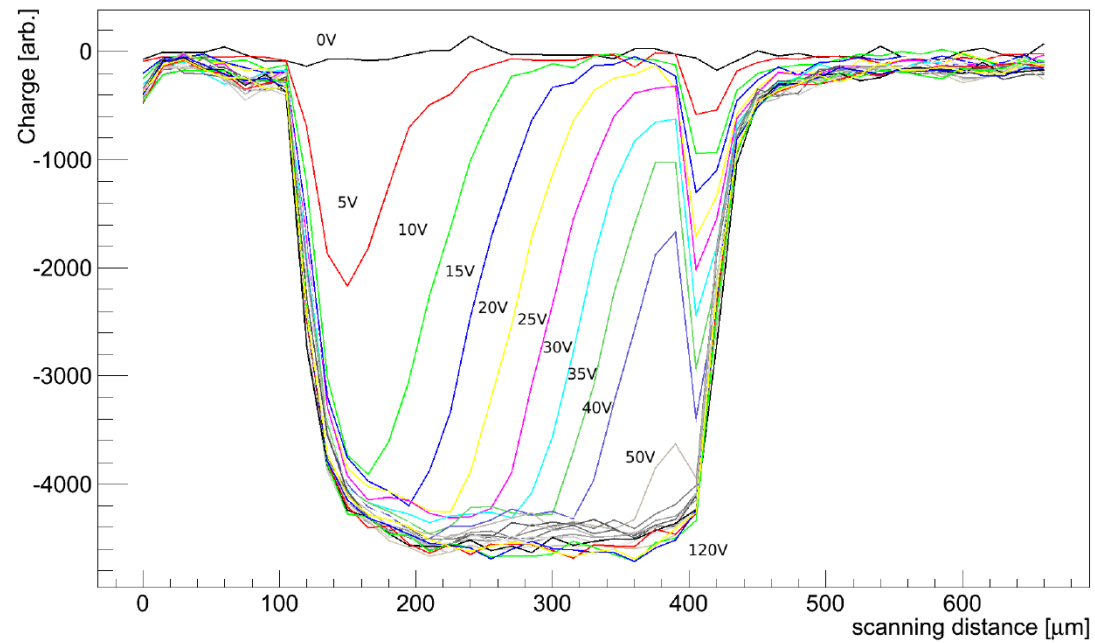
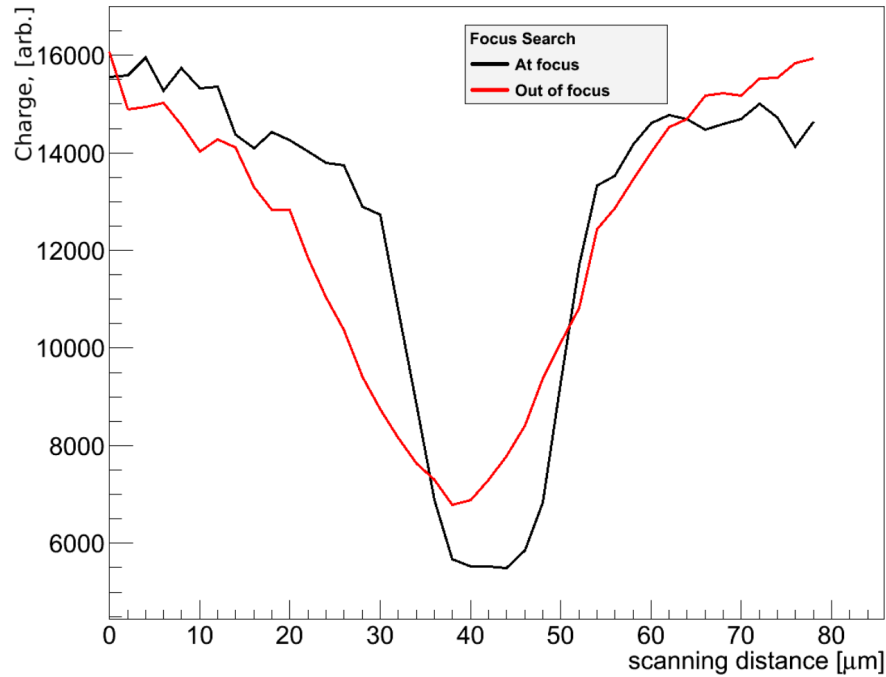
util.cc

Scanning

```
93  
94     if(ana->DO_focus() && ana->TCT_Mode()==0) DoTopFocus();  
95     if(ana->DO_focus() && ana->TCT_Mode()==1) DoEdgeFocus();  
96     //if(ana->DO_focus() && ana->TCT_Mode()==2) BottomDoFocus(); // FIXME needs implementation  
97     if(ana->DO_EdgeDepletion() && ana->TCT_Mode()==1) DoEdgeDepletion();  
98     if(ana->DO_EdgeVelocity() && ana->TCT_Mode()==1) DoEdgeVelocity();  
99     if(ana->CH_PhDiode()) LaserPowerDrop();  
100    if(ana->CH_PhDiode()) BeamSigma();
```

- Implement all “common scans” here
- Just include/exclude from analysis card
- No hard-coded values
 - Just add parameter to analysis card

Examples



Conclusion

- Platform independent
- Easy to share/contribute to code
- Easy cross-check and re-doing of analysis of others
- No re-invention of the wheel for each group
- Mode for single acqs and avg'ed acqs
- First documentation already available (summer student report)