ROOT based tools for DAQ data analysis of the PITZ gun operation

Y. Renier, G. Asova, I. Isaev, DESY, Zeuthen, Germany

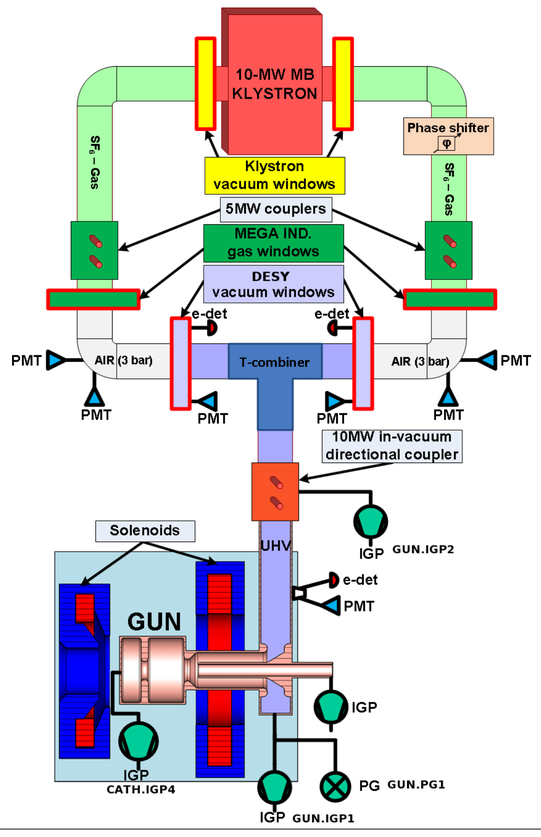
Abstract

Conditioning of electron gun is a lengthy process (typically >=3 months). The evaluation of the conditioning progress and of the reliability of the operation involves tens of sensors (RF measurements, photomultiplier signals, interlocks, pressure gauges, ...). In order to simplify the gathering of this large volume of data from the data acquisition system (DAQ) and its analysis, a ROOT library has been developed.

That library can take into account that measured physical properties can have complex history: the name of the variable within the control system, the scale or the offset can change over the period of interest.

Simple functions allows to retrieve the data from the DAQ, analyze and plot the results. The library is routinely used for different purposes such as to generate automatic shift summaries or to create operation overview for the run coordination meetings.

Gun 4.6 Setup



Gun 4.6 is operated with a 10Hz repetition rate. The RF pulse length are typically very long (nominal pulse length is 650us) to feed FLASH/XFEL superconducting linac.

Most of the RF transport is done in 2 waveguides, and the RF power is recombined in the tunnel just before the gun.

In the left figure, the RF feed system with the interlock sensors are shown. Most of theses are fast sensors measuring at 1Mhz during RF pulses. The associated interlocks can stop the RF within a few microsecond if the measurement exceed the threshold. These fast interlock sensors are:

- Photomultipliers (PMT)
- Electron detectors (e-dect)
- Reflected power measured by directional couplers

The vacuum measured with Ion getter pumps (IGP) and pressure gauges (PG) are slow signals (few Hz).

DAQ system

The PITZ control system is based on DOOCS (http://tesla.desy.de/doocs).

The DAQ system fetch fast sensor data with 10Hz and slow data with 5Hz from DOOCS and stores it in ROOT (https://root.cern.ch) files on band storage using dCache (https://dcache.org). The most recent files are kept on normal hard drive (DAQlive) for a few days for fast access.

There are 3 main types of data types:

- Float array (for fast sensors) with data points every micro-second during the millisecond when the RF pulse is.
- Float (single value) for each RF pulse.
- Flags which are an integer where each bit is a Boolean value representing for example the state of an interlock

The data saved represents about **8 To per month.**

ROOT library

A ROOT library has been developed to interact easily with DAQ data. The main functionality are:

- Configuration of sensor: data type, name and history managing address, scale factor and offset (storage bit and polarity for flags).
- Fetch data from DAQlive or dCache for the 3 types of data.
- Determination of intervals where data is missing and of shutdowns.
- Compression of float data: subsequent values with variation smaller than 1/100th of the total span are skipped. Automatic detection of linear or logarithmic scale is implemented. For flags data, the data is kept only when at least a bit changes.
- Saving data for faster future access.
- Analysis of data and plot generation.

An extensive documentation using doxygen (www.doxygen.org) is available.

Setup and basic usage

Download from SVN:

```
$ svn co
https://svnsrv.desy.de/desy/PITZ/doocs_measure_scripts/ROOT/ILs_database
$ cd ILs_database
```

Initialize environment variables:

```
$ source init_root.sh
```

Start ROOT:

```
$ root -l
```

Load the library:

```
[0] gSystem-> Load("libclasses_data.so");
```

Setup a time interval:

```
[1] time_interval period("2016/10/14
02:01:15", "2016/10/14 02:01:20");
```

Getting scalar data:

```
[2] scalar_data power_in_gun("power in
the gun [MW]",
"RfinfRF2Cavity10MWPower_20131023");
[3] power_in_gun.append_from_DAQ(period);
```

Plot the data:

```
[4] power_in_gun.Plot();
```

That is all !

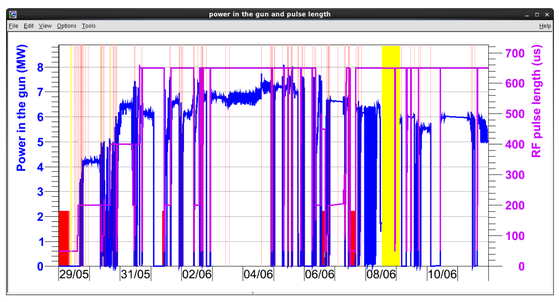
Class run_overview

run_overview class implemented with pre-configured properties (including history):

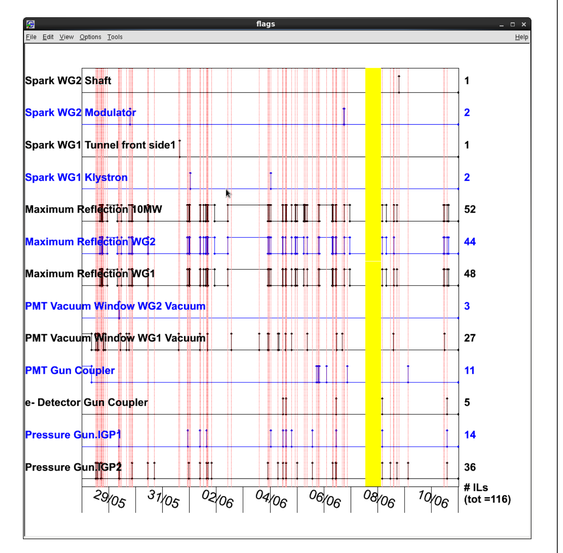
- 37 interlocks flags.
- Average power in the gun, forward and reflected power.
- 4 vacuum pressures (PG1, IGP1, IGP2, IGP4).
- Main solenoid current.
- Laser pulse length, mean charge at low.ICT1, …

Several plot implemented:

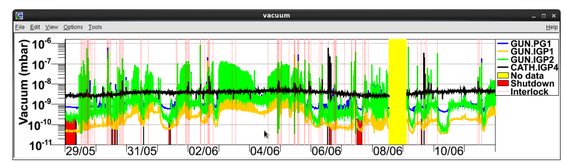
```
[0] gSystem->Load("libclasses_data.so");
[1] time_interval recond("2017/05/29",
"2017/06/12");
[2] run_overview recond_4p6;
[3] recond_4p6.append_from_DAQ(recond);
[4] recond_4p6.Plot_Pgun_pulse_length(
NULL,recond);
```




```
[5] recond_4p6.Plot_ILs(NULL,recond);
```

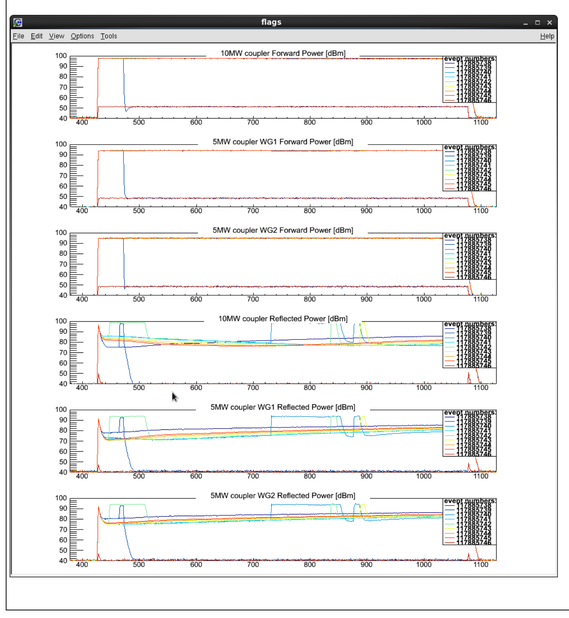



```
[6] recond_4p6.Plot_vacuum(NULL,recond);
```



Class interlock_data

```
[1] interlock_data ILdata;
[2] time_interval ILperiod("2017/06/10
02:52:00","2017/06/10 02:53:00");
[3] ILdata.append_from_DAQ(ILperiod);
[4] ILdata.Plot_LILI_couplers(NULL);
```

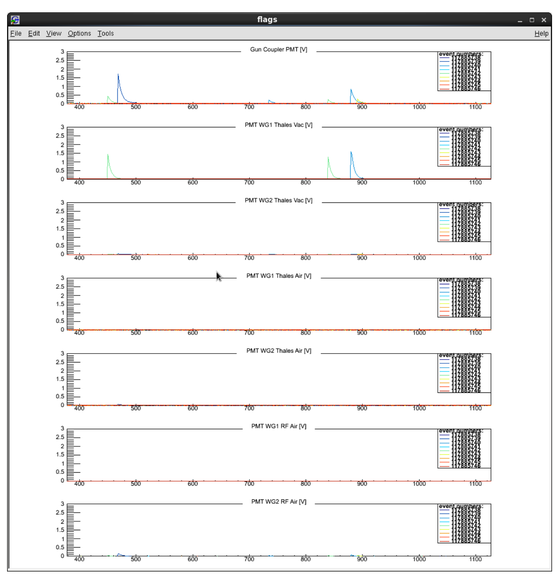


interlock_data class implemented with pre-configured properties (including history) for interlock analysis:

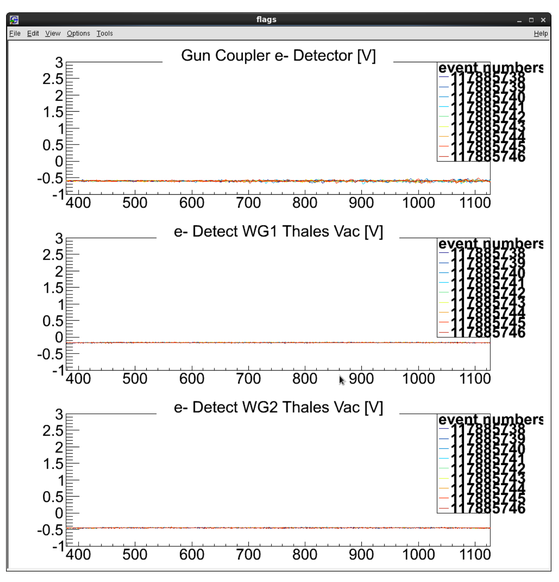
- All fast sensors (16) with full acquired float array.
- 21 Interlock flags.
- Average power in the gun, forward and reflected power.
- RF pulse rise-time, flattop, fall-time.
- Gun temperature, solenoids SP, polarity and read-back values.

No compression for spectra, care to have enough memory if trying to load long periods.
Can recognize most interlocks types and detect channels and events with interlock.

```
[5] ILdata.Plot_LILI_pmt(NULL);
```




```
[6] ILdata.Plot_LILI_edect(NULL);
```



Conclusion and Prospects

A ROOT library has been developed to interact easily with DAQ data. It takes into account that sensors may change name or calibration.

The main data types are handled (flags, scalar value and arrays), as well as missing data and shutdowns. Compression is implemented for scalar data and flags to help analysis and plotting.

Based on these low-level functionalities, user class with relevant sensor pre-configuration were created for operation statistics and interlock analysis.
Extensive documentation is included, and the user class can serve as model for developing new ones.