FEA slow control

Romain Gaïor

July 12, 2024

Abstract

Description and operation of the slow control system used for the Field Array Emission setup at IPMU Kamioka branch.

1 Introduction

The system controls:

- two CAEN HV power supplies
- one cryogenic controller

1.1 Description

It's implemented using a communication protocol called MQTT [https://mqtt.org/]. MQTT is a protocol based on message publishing and subscribing and a central broker which distribute the message from the publisher to the relevant subscribers. The control of the two instruments works the same way: one python script query the data on the ethernet link from the PC to the device and publishes the result on a given topic. Then there are other scripts which are subscriber of the topic and that either log the data in the data base or write them in a file or just display the message. See the sketch in Fig.1.
The code is available on github: https://github.com/rgaior/fea_sc.git.

1.2 Installation

PC

- distribution: LMDE 6 (faye)
- CPU intel I7 64 bit, RAM 8GB
- username: xenon
- IP: `10.240.102.252`

Device addresses

- CAEN1 (PID 58374): `192.168.0.250`
- CAEN2: (PID14133): `192.168.0.251`
- CTC100: `192.168.0.252`

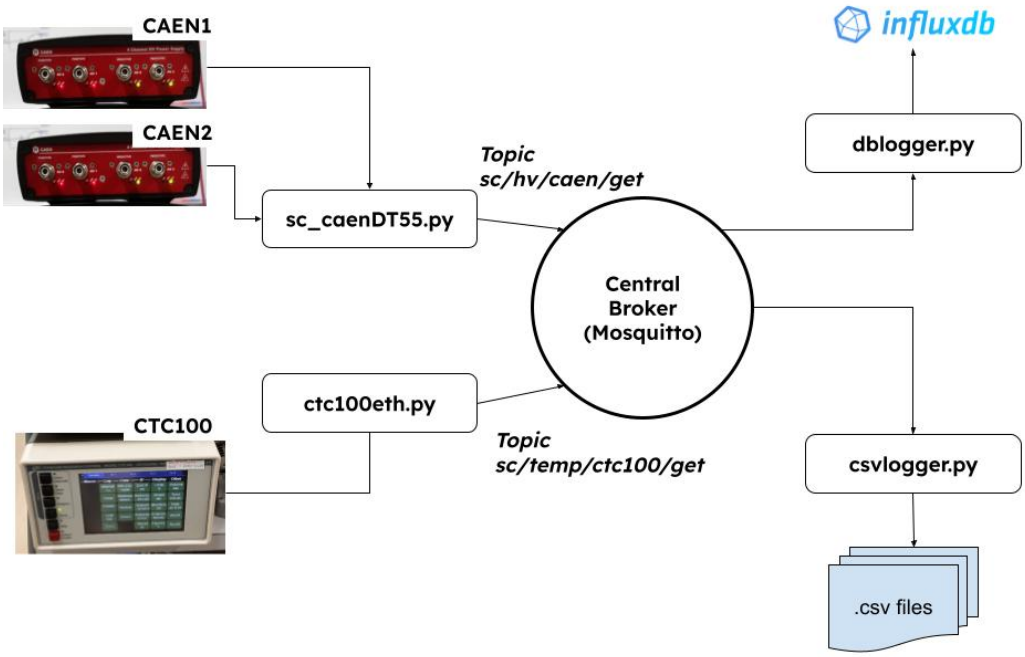

Figure 1: general sketch of the slow control system

influxdb

- local installation
- `InfluxDB v2.7.6 (git: 3c58c06206) build_date: 2024-04-12T21:51:21Z`
- account : username = ipmu ; pw = same as the computer
- UI: http://10.240.102.252:8086/
- bucket: `FEA_SC`
- org: IPMU
- token:

```
u0aOHcX1oQHJUg69ucy9iWnStCeNBuuC_S-3BMtKkWH7B9pjWhf-nw3hgvpnVHuvekZqWO_I_-eNlZGWR7ggyg==
```

code location and environment The main repository is `/home/xenon/slowcontrol/`, it contains the data (`/datalogs/`), the logs (`/logs/`), the environment(`/sc_venv/`) and the code (`/fea_sc/`).

In order to activate the environment, type:

```
source /home/xenon/slowcontrol/sc_venv/bin/activate
```

2 Operation

2.1 Logging

The code are python script that are run in interactive mode, so if you want to run only the data query of the ctc100 for instance you should type:

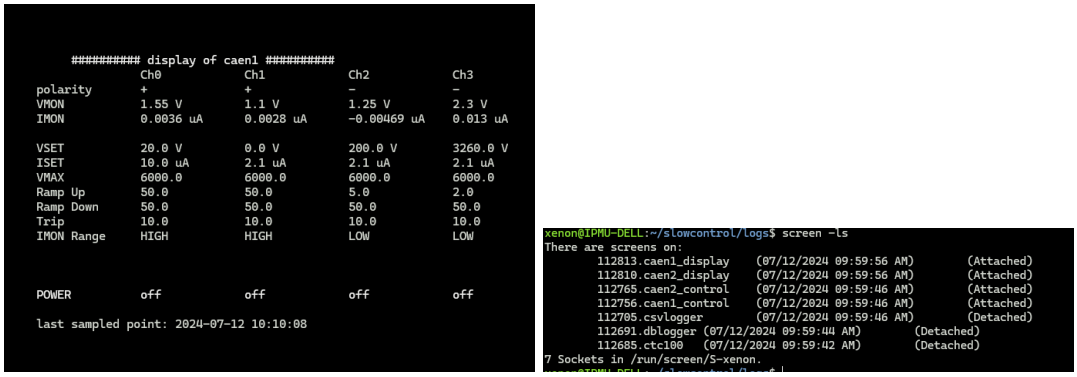

Figure 2: display of the CAEN HV information

```
python -i ctc100/ctc100eth.py
```

The CAEN HV take the argument (caen1 or caen2) to run:

```
python -i caendt55/sc_caendt55.py caen1
```

This will run a python shell and start querying the data from the device in a separate thread. Similarly if you want to log the data you need to run the dedicated codes:

```
python -i datalogger/csvlogger.py
python -i datalogger/dblogger.pypython
```

2.2 CAEN Control

For the control of the CAEN, in the python shell opened with `sc_caendt55.py` code, one can simply type

```
caenhv.set(channel, PARAM, value)
```

where channel is 0,1,2,3 PARAM are listed in the CAEN power supply manual (see the screen shot in Fig. 4) and value is the value you want to set.

2.3 CAEN display

In the case of the CAEN we want to log, control and display the output values. These operations are done with separate scripts (`sc_caendt55.py` for the control and `caen_display.py` for the display). The display is run with

```
python -i caendt55/caen_display.py caen1
```

This will open a python shell with the information of the CAEN displayed

2.4 Scripts

There are some bash script to open automatically the monitoring parts in different `screen` sessions:

- `/slowcontrol/start_ctc.sh` : start the ctc logging
- `/slowcontrol/start_caen.sh`: start the CAEN logging, control and display and puts that in a TMUX window Fig. 3
- `/slowcontrol/start_logger.sh` : start the code to push data in db and in csv file.
- `/slowcontrol/start_sc.sh` : start the above scripts

At the end, when all the script are executed one should see a list of screens like in Fig. 2 (right).
The CAEN control and display window are in a TMUX so you need to know the related commands.

Figure 3: CAEN TMUX window for control and display

- to detach from the TMUX window: `ctrl-b d`
- to list the TMUX session: `tmux ls`
- to attach from the TMUX window: `tmux attach -t <name of the session>` or `tmux a` if there is only one session.
- To navigate between the panes with `ctrl-b` ←→↑↓

So you can set a value in the CAEN control part and see the result instantaneously in the CAEN display part.

2.5 logs

- The logs of action are recorded in the folder: `/slowcontrol/logs/`
- The csv file are store in the folder: `/slowcontrol/datalogs/`

A set commands

Set Commands

$CMD:SET,CH:XX,PAR:VSET,VAL:XXX.XX	VSET value
$CMD:SET,CH:XX,PAR:ISET,VAL:XXXX.XX	ISET value
$CMD:SET,CH:XX,PAR:IMRANGE,VAL:LOW	IMON range low
$CMD:SET,CH:XX,PAR:IMRANGE,VAL:HIGH	IMON range high

$CMD:SET,CH:XX,PAR:RUP,VAL:XXX	RAMP UP value
$CMD:SET,CH:XX,PAR:RDW,VAL:XXX	RAMP DOWN value
$CMD:SET,CH:XX,PAR:TRIP,VAL:XXX.X	Set TRIP time value
$CMD:SET,CH:XX,PAR:PDWN,VAL:RAMP	POWER DOWN ramp mode
$CMD:SET,CH:XX,PAR:PDWN,VAL:KILL	POWER DOWN kill mode
$CMD:SET,CH:XX,PAR:ON	Set Ch ON
$CMD:SET,CH:XX,PAR:OFF	Set Ch OFF
$CMD:SET,CH:XX,PAR:ZCDTC,VAL:ON	If ON, it stores the present IMon value (IMonZero) into memory for "zero current compensation" purposes (see description below); if OFF, the unit is ready to store IMon as IMonZero. After IMonZero is stored, the parameter returns to OFF
$CMD:SET,CH:XX,PAR:ZCDTC,VAL:ON	If ON, it stores the present IMon value (IMonZero) into memory for "zero current compensation" purposes (see description below); if OFF, the unit is ready to store IMon as IMonZero. After IMonZero is stored, the parameter returns to OFF
$CMD:SET,CH:XX,PAR:ZCADJ,VAL:EN	The stored IMonZero value via ZCDetect option is subtracted from the measured, "non compensated" IMON value. The returned "compensated" IMON value will be then the difference between measured and stored values;
$CMD:SET,CH:XX,PAR:ZCADJ,VAL:DIS	The returned IMON value is not compensated

Figure 4: