Light readout software and QA/QC

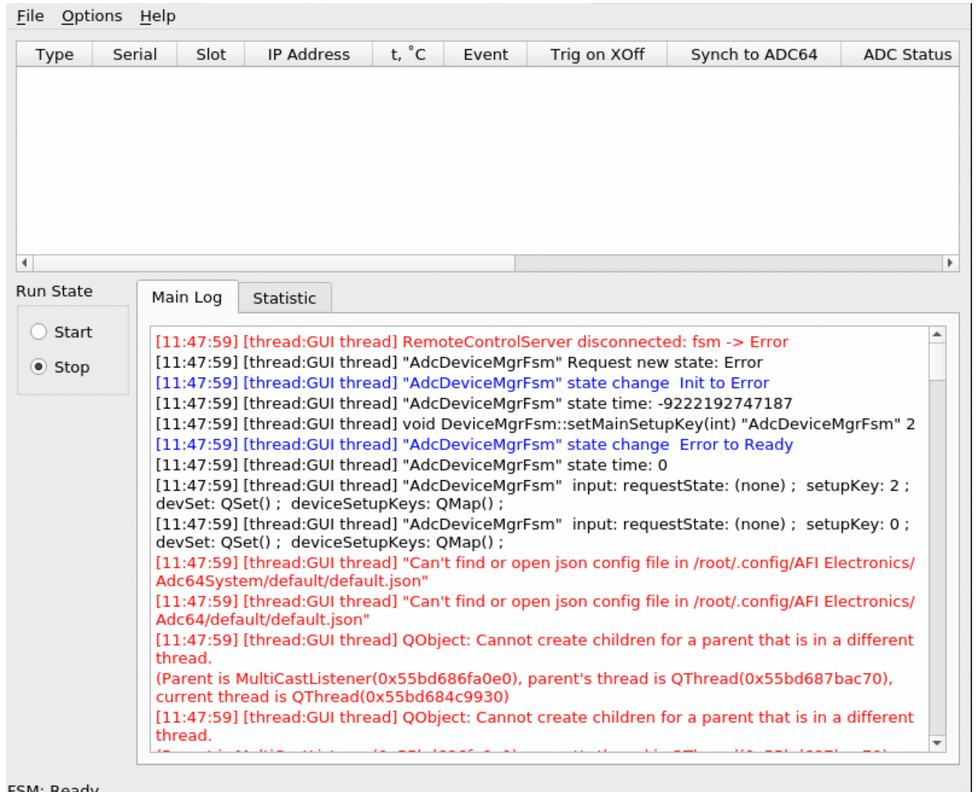
ArgonCube 2x2 Electronics and Readout Integrations, 25 August, 2022

Alexander Selyunin

Installed software

- ACD-SRV02 is used for software installation
- DHCP configured on srv02 on enp7s0f0 interface
- Docker container for ADC GUI software :
  - ksu (Root required to run the docker)
  - docker run -it --rm -p5900:5900 --network host quay.io/kozhukalov/adc64-gui
- VNC to access the ADC GUI from remote PC
  - ssh -L5900:localhost:11590 acdemo@acd-gw01.fnal.gov (ssh tunnel to gw01)
  - ssh -L11590:localhost:5900 acd-srv02 (ssh tunnel to srv02 from gw01)
  - VNC connection to localhost:5900
    - cd build/adc64
      - ./afi-adc64 (ADC oscilloscope gui)
    - cd build/adc64-system
      - ./afi-adc64-system (data acquisition application)

ADCs need to be powered up and connected to the network switch in order to check that we can readout ADC!!!



Installed software

HV control software

1. SSH to the RPi which serves the SiPM PS control board

2. Activate software environment
   - **source .supplr_venv/bin/activate**

Use the following cli commands to control it:

supplr list of all available commands
supplr board-status print out the connected board (defines board number which is 0 for Module 1)
supplr set-channels-volt --board <board number> --voltage <voltage value> set the same voltage for all channels [128 channels in total]
supplr reset --board <board number> turn all the voltages off
supplr set-channel-volt --board <board number> --mez <mez number> --mezch <mez channel number> --voltage <voltage value>
set the voltage [up to 120V*] for a particular mezzanine [0-3] and mezzanine channel number [0-31]
supplr read-hv-supply-voltage --board <board number> reads the voltage value which is set on the laboratory power supply [SiPM bias]

Installed software

VGA control software

There are two options:

1. The client part is already installed on the Raspberry Pi. Use ssh to connect the RPi on VGA control board and then use the command line interface: **gainr set -c config.yaml**. The config format is given below.
2. You can install the client part of `Gainr` software doing the following:

```
git clone https://git.jinr.ru/greenlab/gainr.git
cd gainr
python3 -m venv .venv
source .venv/bin/activate
poetry install
```

Configure

```
cat > config.yaml <<EOF
endpoint: http://<raspberry-py-ip>:5000
# Channel levels must be integers from 0 to 240
channels:
  0: 0
  1: 0
  2: 0
  3: 0
  4: 0
  5: 0
  6: 0
  7: 0
EOF
```

240 corresponds to 24dB amplification factor
0 corresponds to 0dB amplification factor

Set the output levels on VGA control

```
gainr set -c config.yaml
```

QA/QC procedures

- Connectivity check of the power lines for pre-amps
    - Power the E-PCBs to check the current draw -> 1.06A for all 16 E-PCBs or 265mA per VGA adapter card
- Channel by channel signal check -> LED system implementation required:
    - Power the LED by pulse Generator and provide TTL trigger for ADC
    - Power the SiPMs (~56V for all channels)
    - Run ADC GUI to check if all signals present