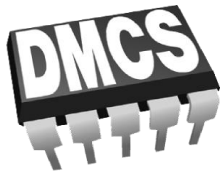
MTCA.4 RTM Carrier Board integration
Realization status and open tasks

Presenter:
Aleksander Mielczarek, DMCS
Kacper Kłys, DMCS



2020-10-07, Łódź

MTCA.4 RTM Carrier Board Integration

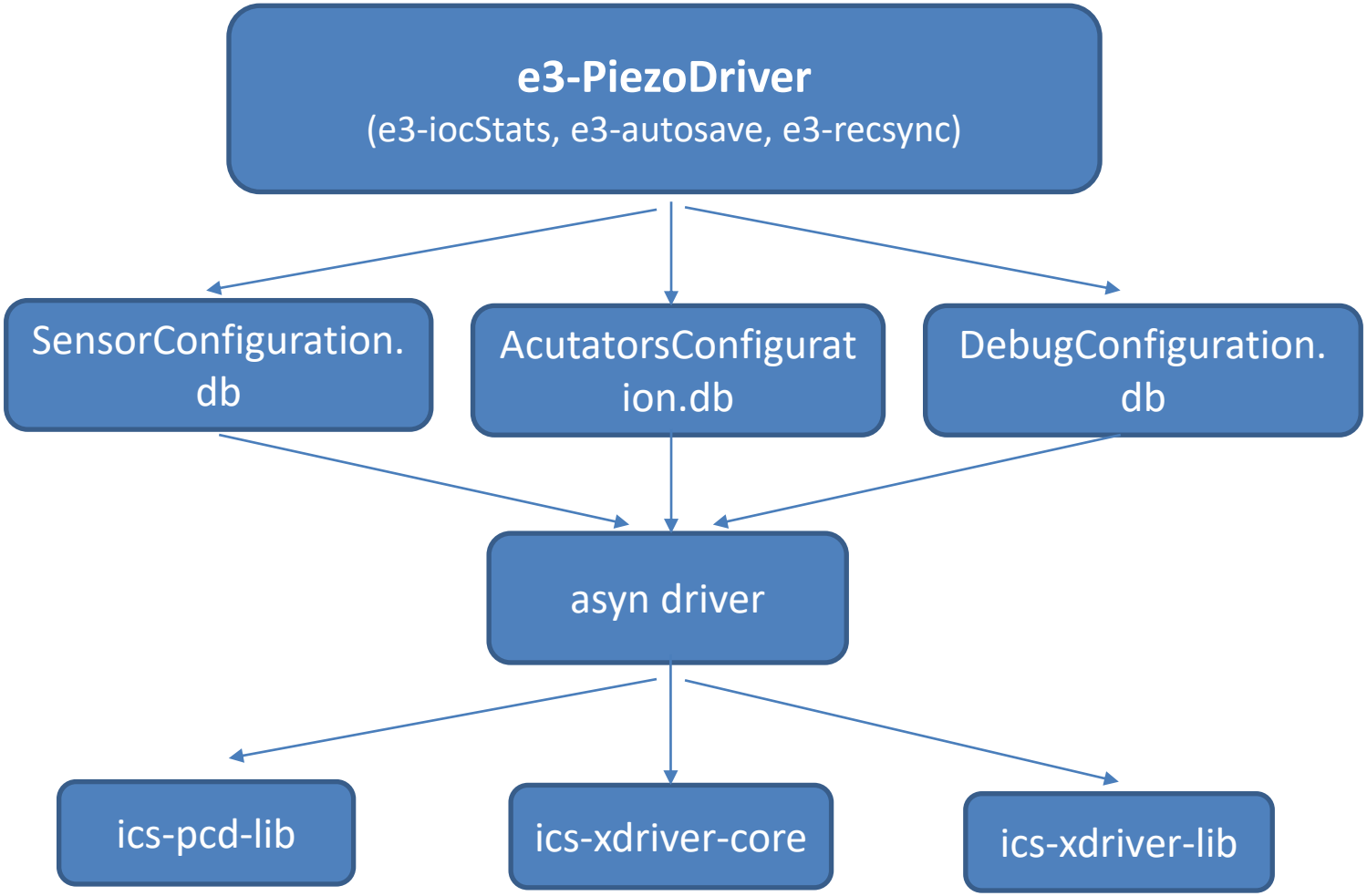
Realization Status:

- Removed projects from main ESSFFW repository and put them into two new repositories.
- Updated LO RTM library and application in order to work with updated LO RTM CPLD firmware.
- Corrected bug in driver, where two IOCTLs were given the same identifier.

Open Tasks:

- Actually make the piezo driver firmware – which is not possible as we still do not have a framework-compliant BSP with working DDR3 from NCBJ.

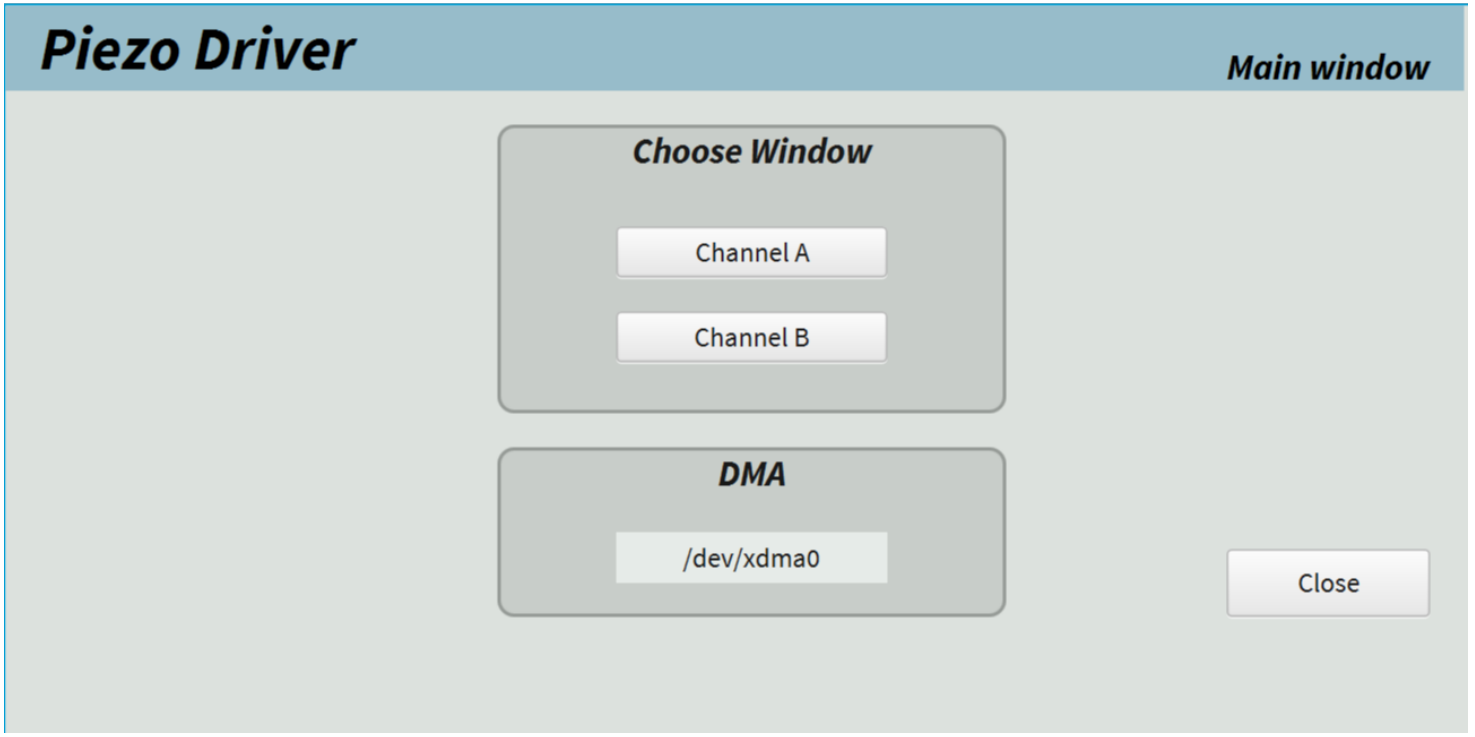
IOC for Piezo Driver - Structure



IOC for Piezo Driver - Structure

- Uses asyn driver based on API from Piezo Control device library (git submodules)
- Made of three .db files, on for each mode and last for general settings
- Working in one main thread in the infinite loop:
  - acquiring data
  - synchronising parameters changes
- Attached all necessary e3 modules: autosave, iocStats, recsync

IOC for Piezo Driver - OPI



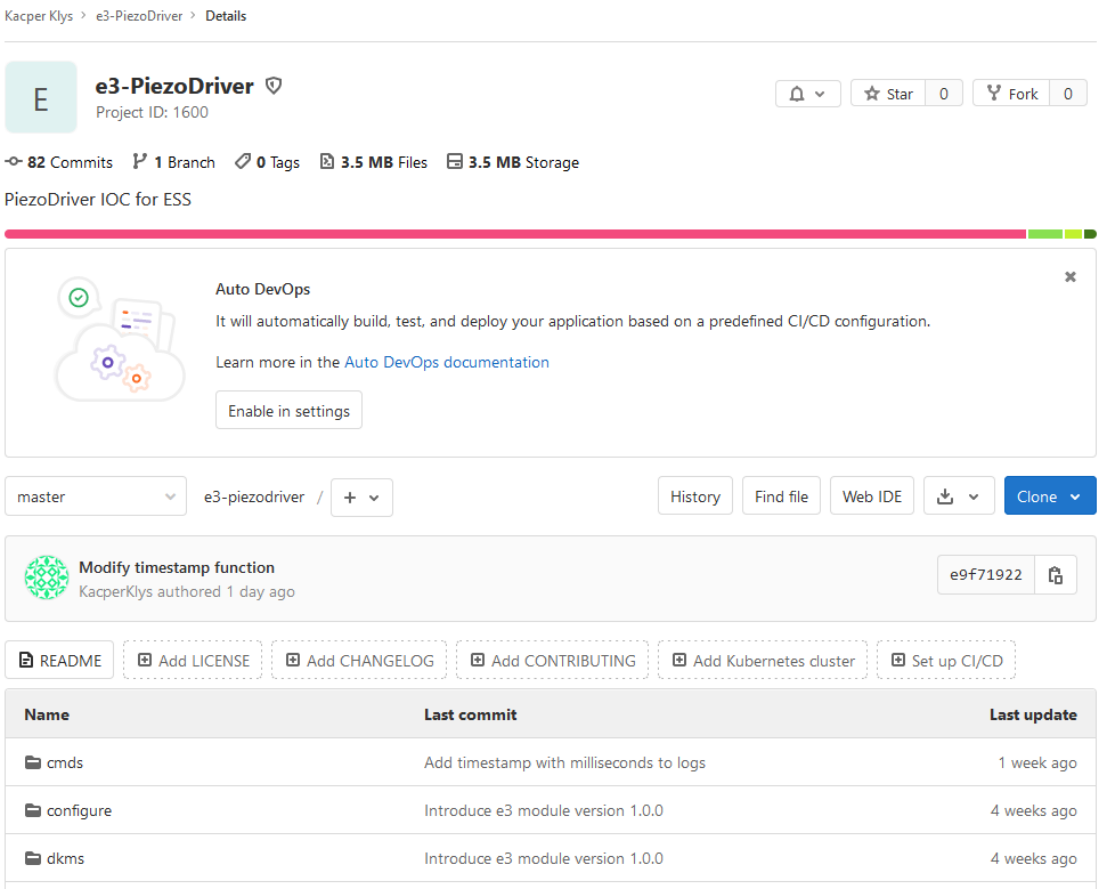
- Composed of three windows:
  - Main panel
  - Channel A
  - Channel B

IOC for Piezo Driver - OPI

- Preview of generated signal (from file or from OPI)
- Data from ADCs (possible saving to file)
- Trigger configuration (slope, delay etc.)

IOC for Piezo Driver - Links

- Code on ESS gitlab: https://gitlab.esss.lu.se/kacperklys/e3-piezodriver



IOC for Piezo Driver - Links

- Documentation on confluence (will be updated soon): https://confluence.esss.lu.se/display/HAR/Piezo-driver+integration+-+IOC

How to import .bob files

How to install e3 module

How to install kernel module

5. Before installing module one has to initialize and download submodules from external repositories:

update submodules

```
$ cd/e3-PiezoDriver
$ git submodule init
$ git submodule update
```

6. In the e3-PiezoDriver directory and type commands to install kernel module with dkms:

install kernel module

```
$ make dkms_add
$ make dkms_build
$ make dkms_install
```

7. To install module, in the same directory type following commands:

install whole module

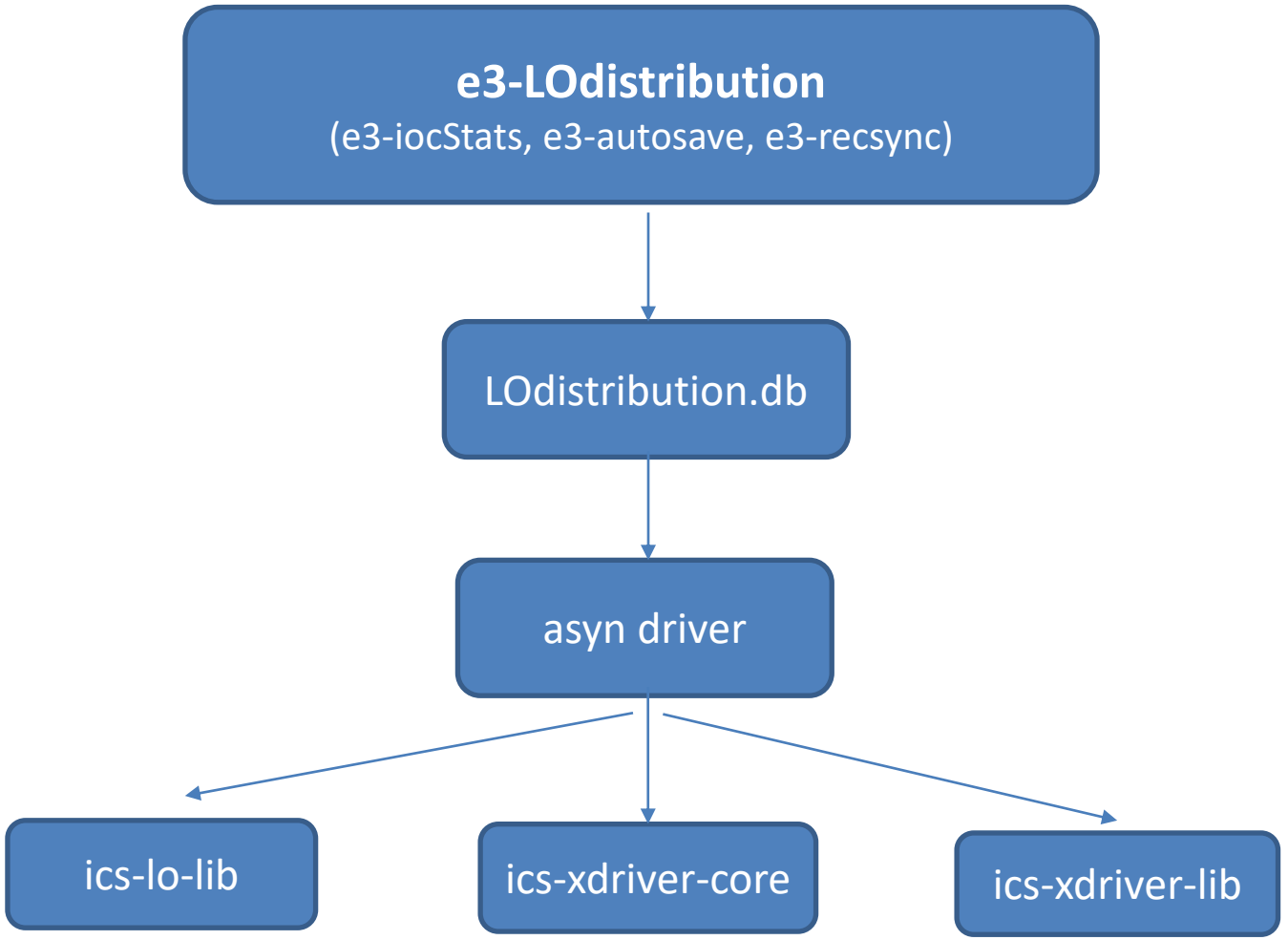
```
$ make build
$ make install
```

8. To run IOC, one can type

run ioc

```
$ iocsh.bash cmds/PiezoDriver.cmd
```

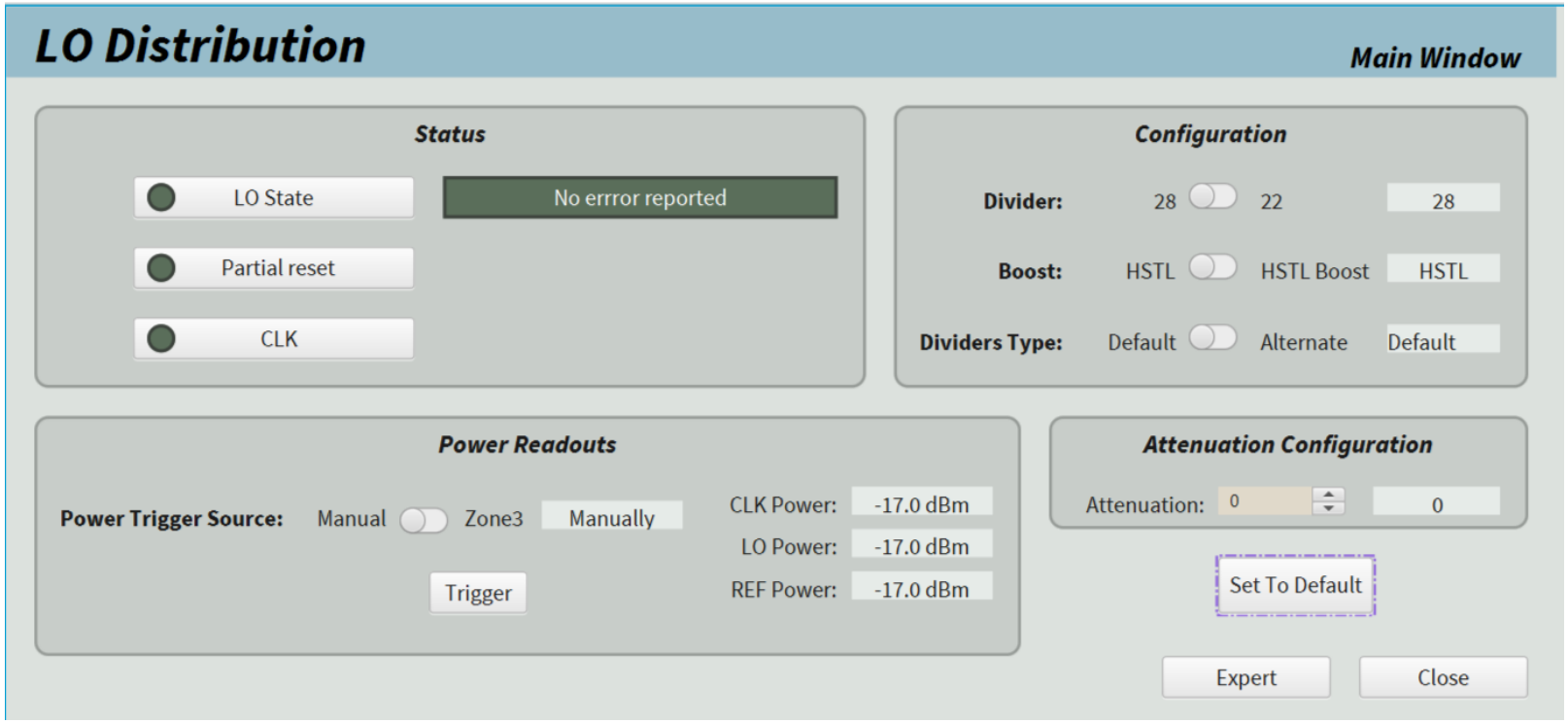
IOC for LO - Structure



IOC for LO - Structure

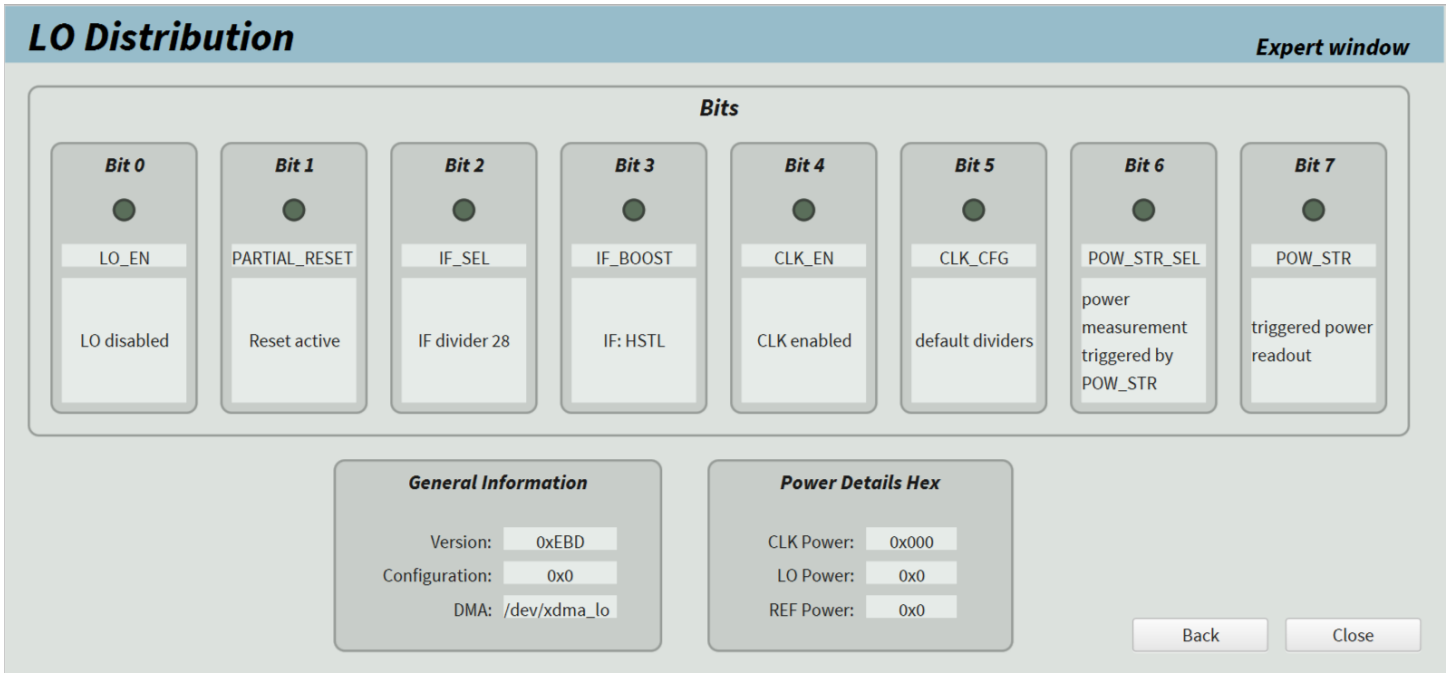
- Uses asyn driver based on API from LO library (git submodules)
- Made of one, main .db file
- Working in one main thread in the infinite loop synchronising parameters changes
- Attached all necessary e3 modules: autosave, iocStats, recsync

IOC for LO - OPI



- Main window:
  - Enable CLK, LO
  - LO configuration (divider, boost, attenuation)
  - Power readouts in dbm

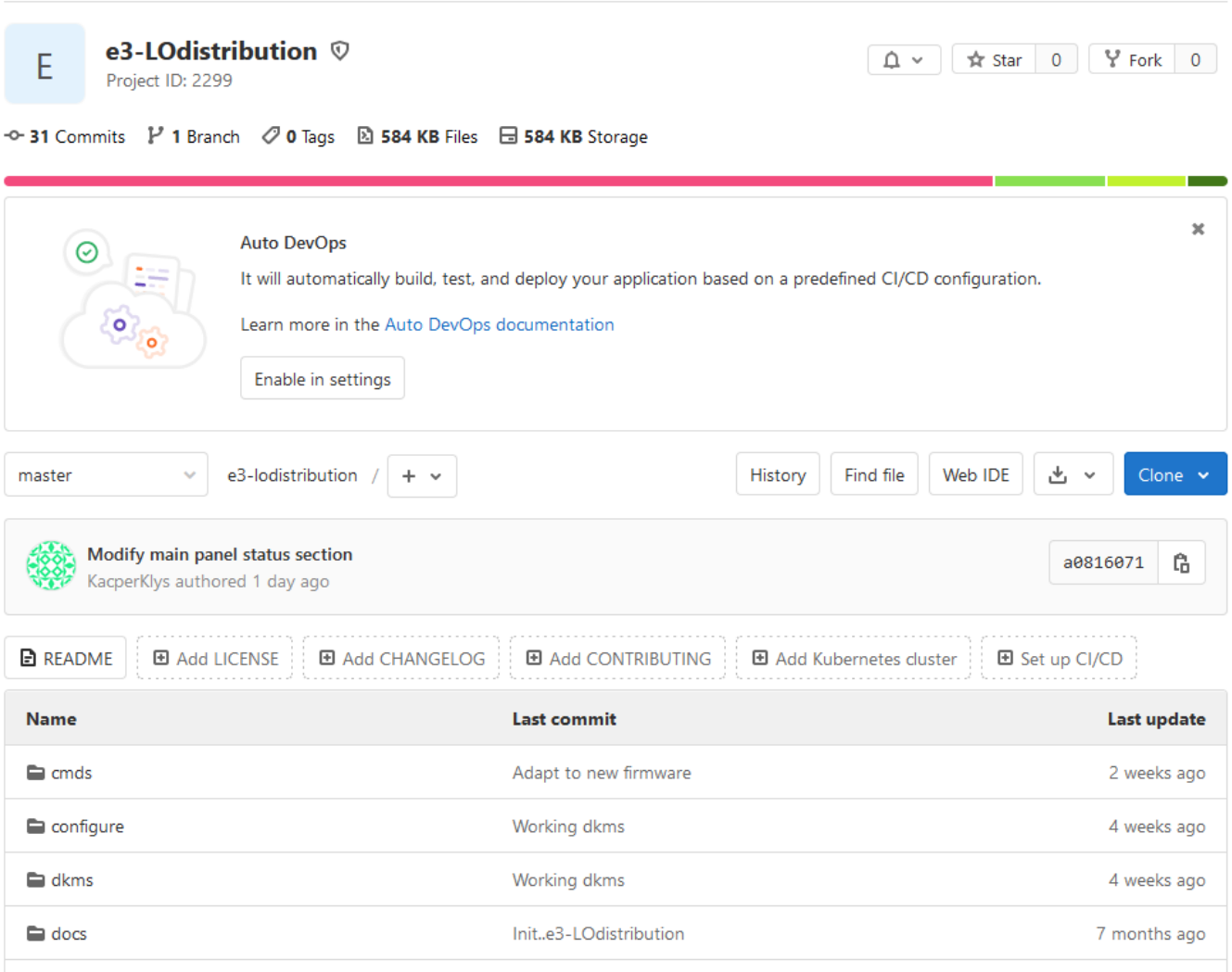
IOC for LO - OPI



- Expert window:
  - Bit descriptions and statuses
  - Software version
  - Power readouts in hex

IOC for LO Links

- Code on ESS gitlab: https://gitlab.esss.lu.se/kacperklys/e3-lodistribution



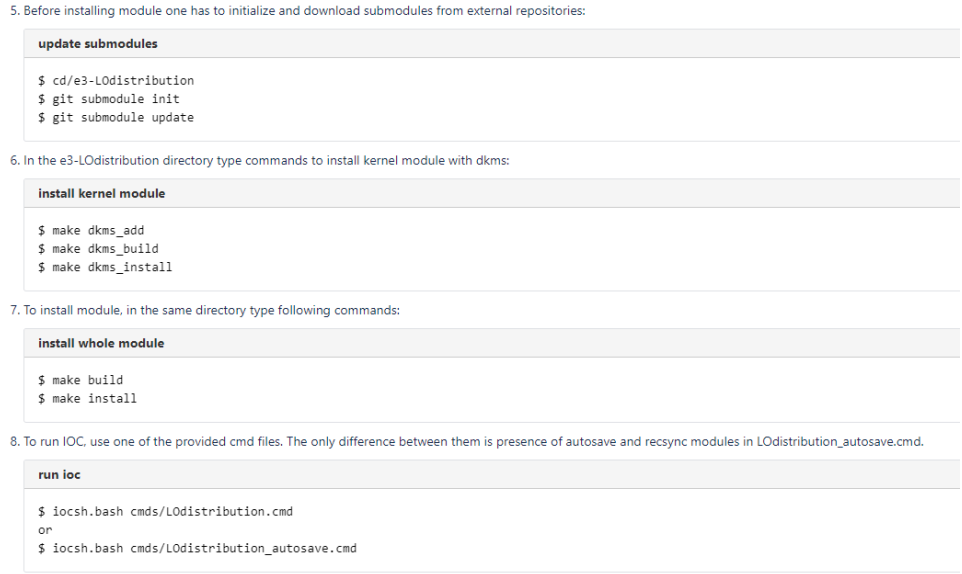
IOC for LO - Links

- Documentation on confluence (will be updated soon): https://confluence.esss.lu.se/display/HAR/LO+distribution+-+IOC

How to import .bob files

How to install e3 module

How to install kernel module



5. Before installing module one has to initialize and download submodules from external repositories:

update submodules

```
$ cd/e3-LOdistribution
$ git submodule init
$ git submodule update
```

6. In the e3-LOdistribution directory type commands to install kernel module with dkms:

install kernel module

```
$ make dkms_add
$ make dkms_build
$ make dkms_install
```

7. To install module, in the same directory type following commands:

install whole module

```
$ make build
$ make install
```

8. To run IOC, use one of the provided cmd files. The only difference between them is presence of autosave and recsync modules in LOdistribution_autosave.cmd.

run ioc

```
$ iocsh.bash cmds/LOdistribution.cmd
or
$ iocsh.bash cmds/LOdistribution_autosave.cmd
```