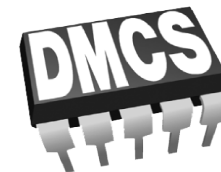




RTM Carrier Support

Presenter:
Aleksander Mielczarek, DMCS
Kacper Kłys, DMCS



2020-02-10, Łódź



FPGA Firmware



Factors Driving the Firmware Development

FPGA Firmware

- Need to control and monitor the LO RTM
- Changes in Piezo Control Device hardware
- Updates to ESS FPGA Firmware Framework
- Need to overcome problems with DDR3 memory



LO RTM Support

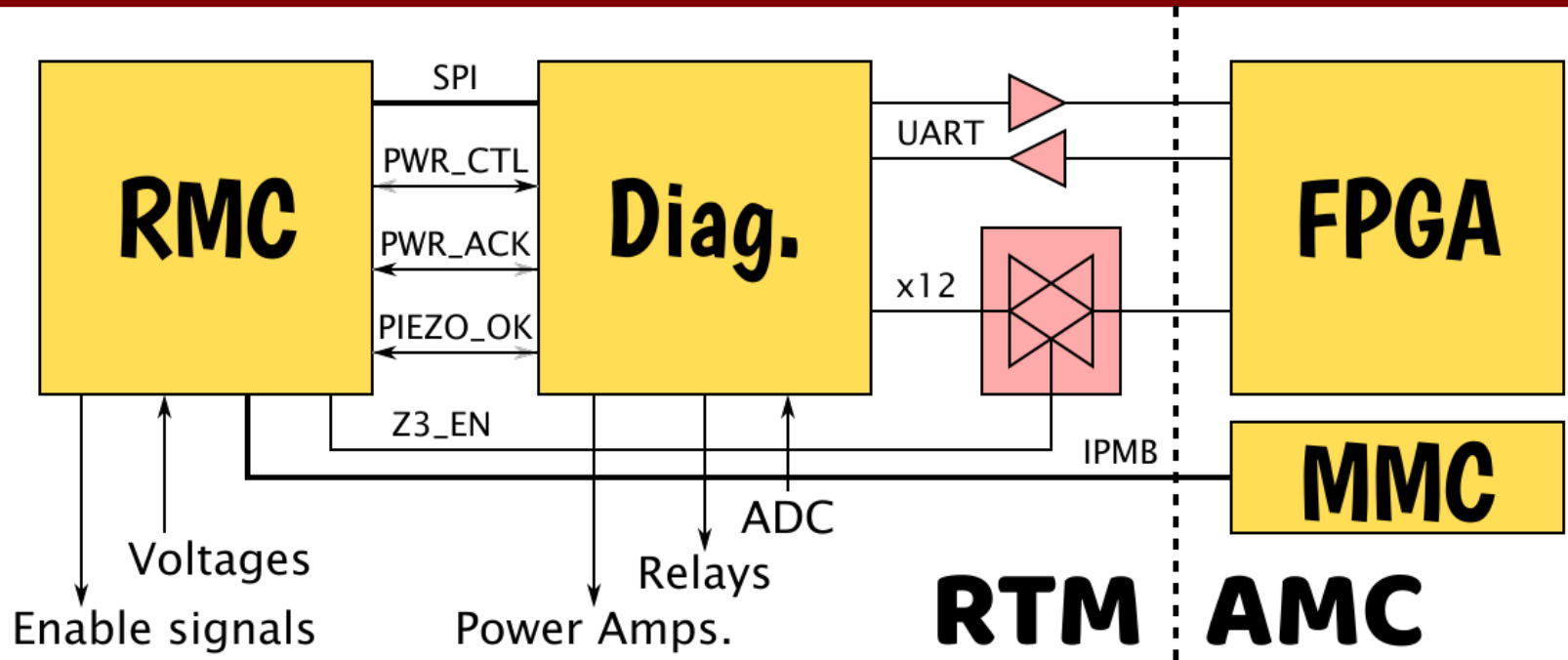
FPGA
Firmware

- LO RTM contains:
 - RF power measurement circuits
 - Adjustable attenuator
 - Configurable divider
 - Several controlled buffers
- Everything is accessible through SPI interface
- Additional measurement trigger and done signals



Piezo Control Device Management

FPGA
Firmware

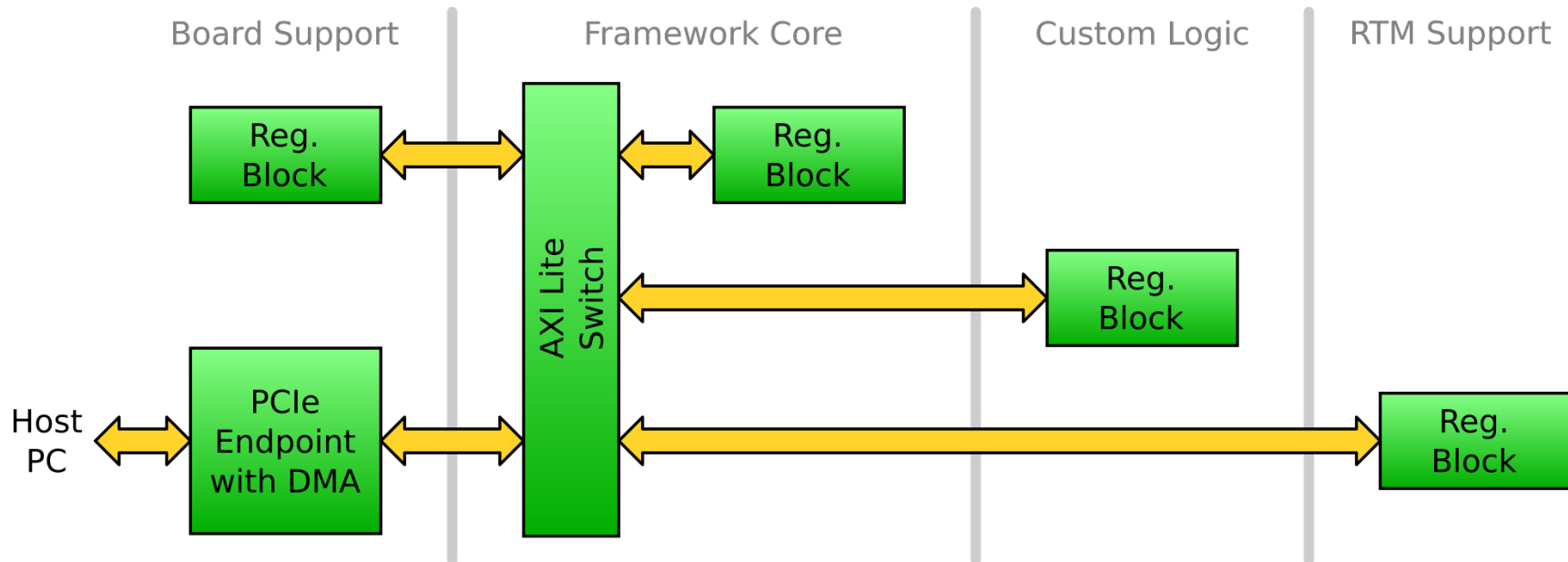


- UART communication is no longer used for switching channel modes between sensor and actuator.
- The control is now implemented with 4 lines per ch.:
ACTUATOR – request to PCD,
IS_ACTUATOR – indication of current mode,
PROT_TRIPPED – indication of an unwanted event,
PROT_RESET – request to return to operation.



Updates to ESS FPGA Firmware Framework

FPGA Firmware

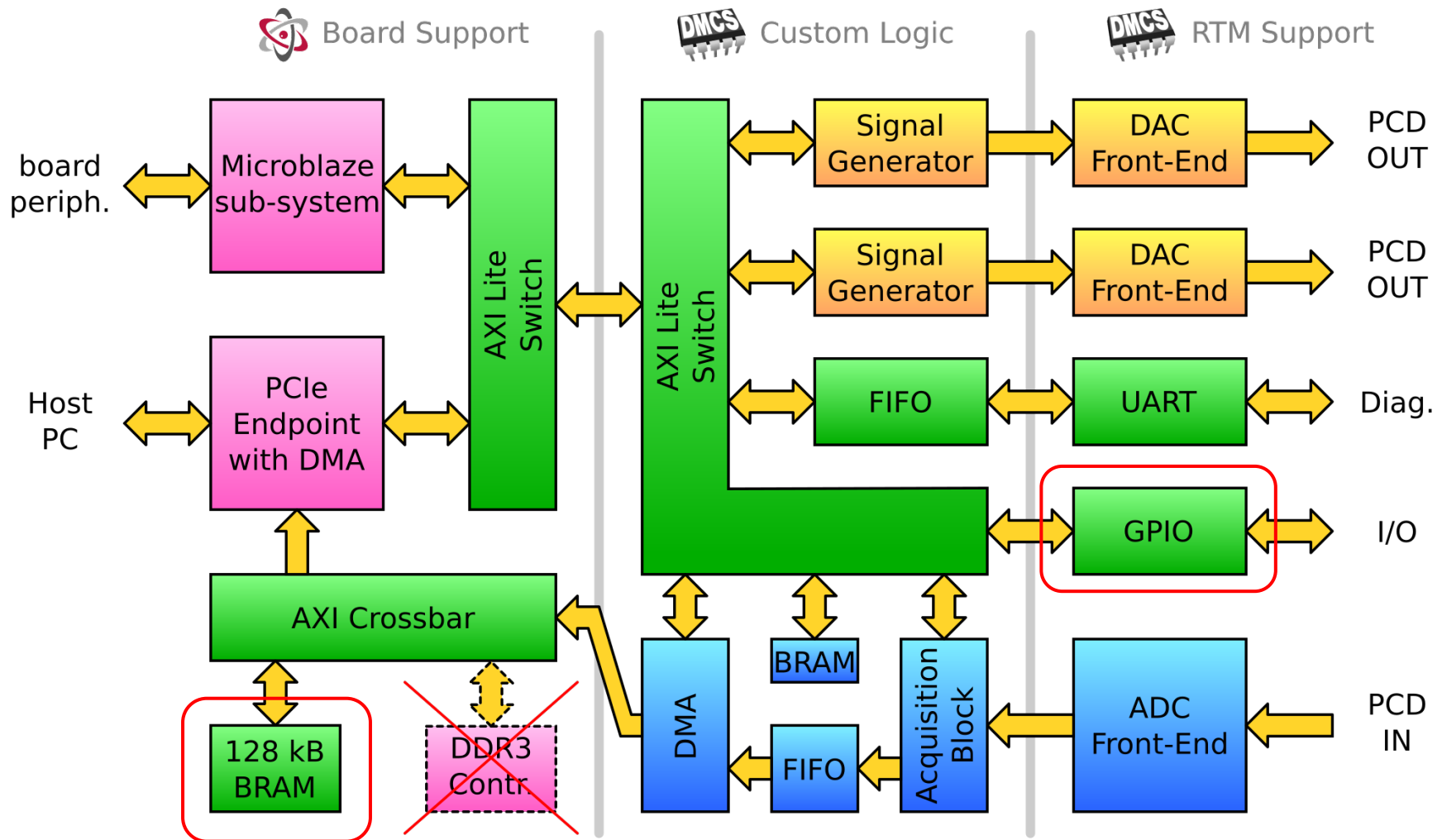


- Each component of the ESS Firmware Framework is now equipped with a register block.
- Each register block has an obligatory part containing unique component ID and version registers.



Firmware for Piezo Control Device

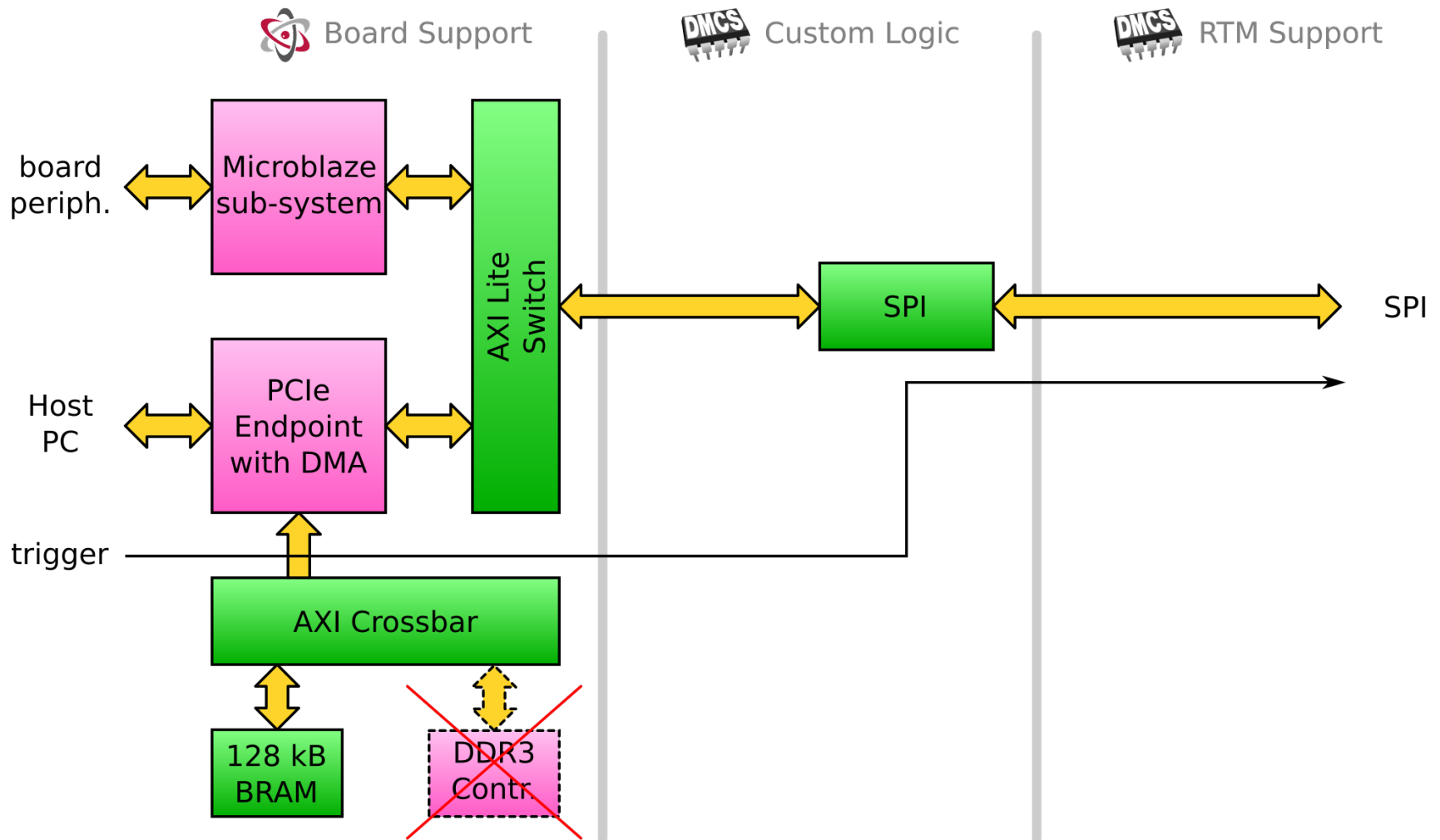
FPGA Firmware





Firmware for LO RTM


FPGA Firmware





Wish List

FPGA Firmware

- Dear NCBJ, please provide: 
- Working memory controller
- Fix for the M-LVDS signal receiver



Low-Level Software



Low-Level Software Support

Low-Level
Software



<https://bitbucket.org/europeanspallationsource/ics-xdriver-core/src/master/>

<https://bitbucket.org/europeanspallationsource/ics-xdriver-lib/src/master/>

<https://bitbucket.org/europeanspallationsource/ics-pcd-lib/src/master/>

<https://bitbucket.org/europeanspallationsource/ics-lo-lib/src/master/>



Changes to the Low-Level Software



Low-Level Software

- Provided new library for LO RTM.
- Extended PCD library with functions handling the GPIO communication.
- Extended driver support library with identification of system components.



LO RTM Support



Low-Level
Software

● Provided new library for LO RTM

```
int lo_version_get      (xildev *dev, uint16_t *ver);

int lo_config_get      (xildev *dev, uint8_t *value);
int lo_config_set_bits (xildev *dev, uint8_t set_mask);
int lo_config_clear_bits(xildev *dev, uint8_t clear_mask);

int lo_atten_get       (xildev *dev, uint8_t *value);
int lo_atten_set       (xildev *dev, uint8_t value);

int lo_power_get       (xildev *dev, uint16_t *clk_pow,
                        uint16_t *lo_pow, uint16_t *ref_pow);

float lo_power_in_dbm   (uint16_t power_word);
```



Piezo Control Device Support



Low-Level Software

- Extended PCD library with functions handling the GPIO communication

```
int pcd_mcu_set_dir(xildev *dev, int channel, bool actuator);  
int pcd_mcu_get_dir(xildev *dev, int channel, bool *actuator);  
  
int pcd_mcu_is_actuator(xildev *dev, int channel, bool *actuator);  
  
int pcd_mcu_get_prot(xildev *dev, int channel, bool *tripped);  
  
int pcd_mcu_reset_prot(xildev *dev, int channel);
```



Low-Level Software

- Extended driver support library with identification of system components

```
$ ics-xdriver-lib$ ./xil-rw /dev/xdma1 -i
```

```
User BAR size 16777216
```

```
Framework ID: 0x101  
Framework version: 0.3.0
```

```
Project ID: 0x101  
Project build number: 61  
Project GIT hash: cfaec9dc
```

```
BSP ID: 0x6401  
BSP version: 1.0.0
```

```
RSP ID: 0x6501  
RSP version: 1.0.0
```

```
CSTMLOG ID: 0x6701  
CSTMLOG version: 1.0.0
```

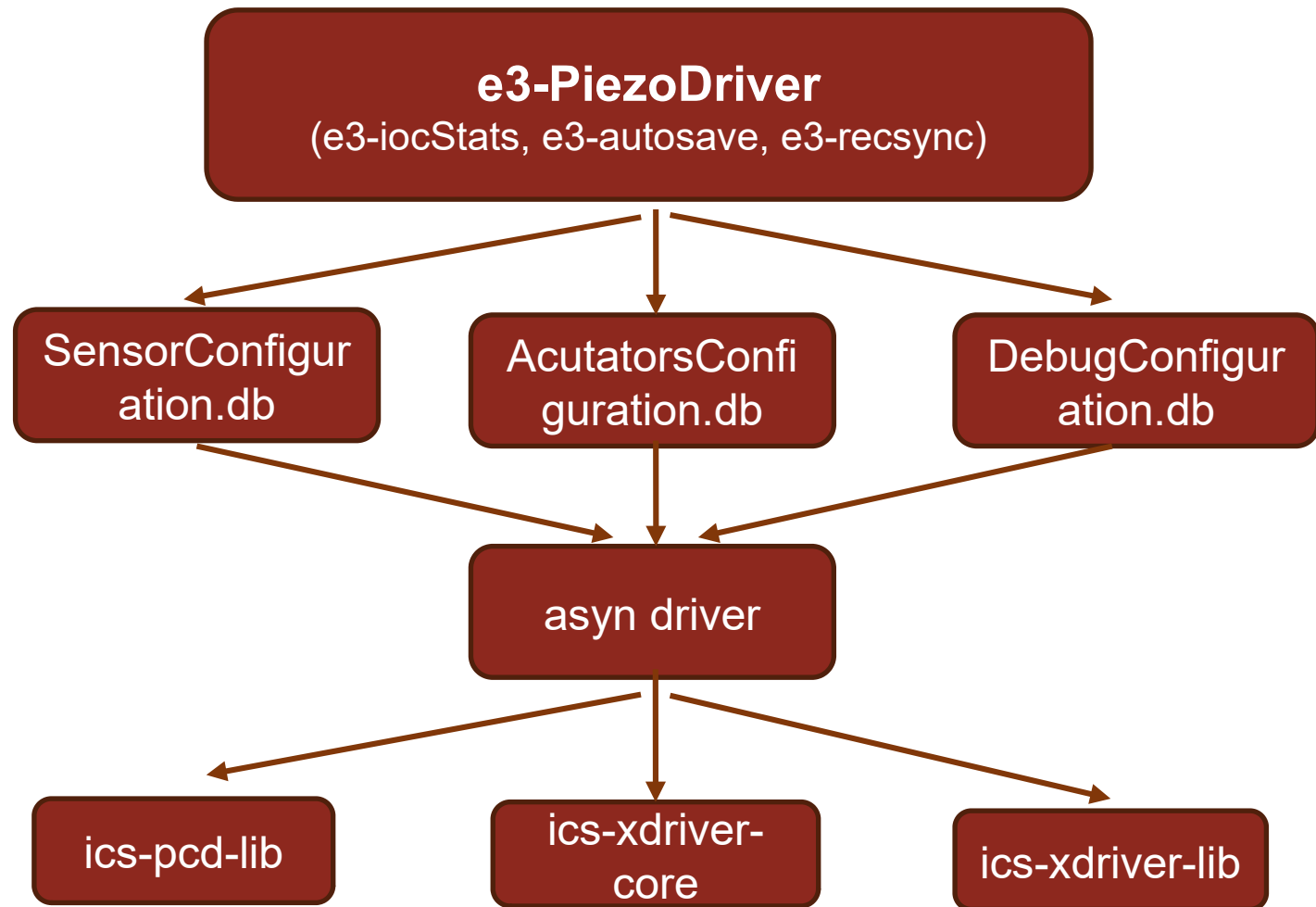


EPICS Software for Piezo Control Device



IOC Structure

EPICS
Software





IOC Structure

EPICS Software

- Uses asyn driver based on API from Piezo Control device library
(git submodules)
- Made of three .db files, one for each mode and last for general settings
- Working in two threads:
 - one acquiring data (in the future it will detect trigger signal),
 - second synchronizing parameters changes
- Attached all necessary epics modules: autosave, iocStats, recsync

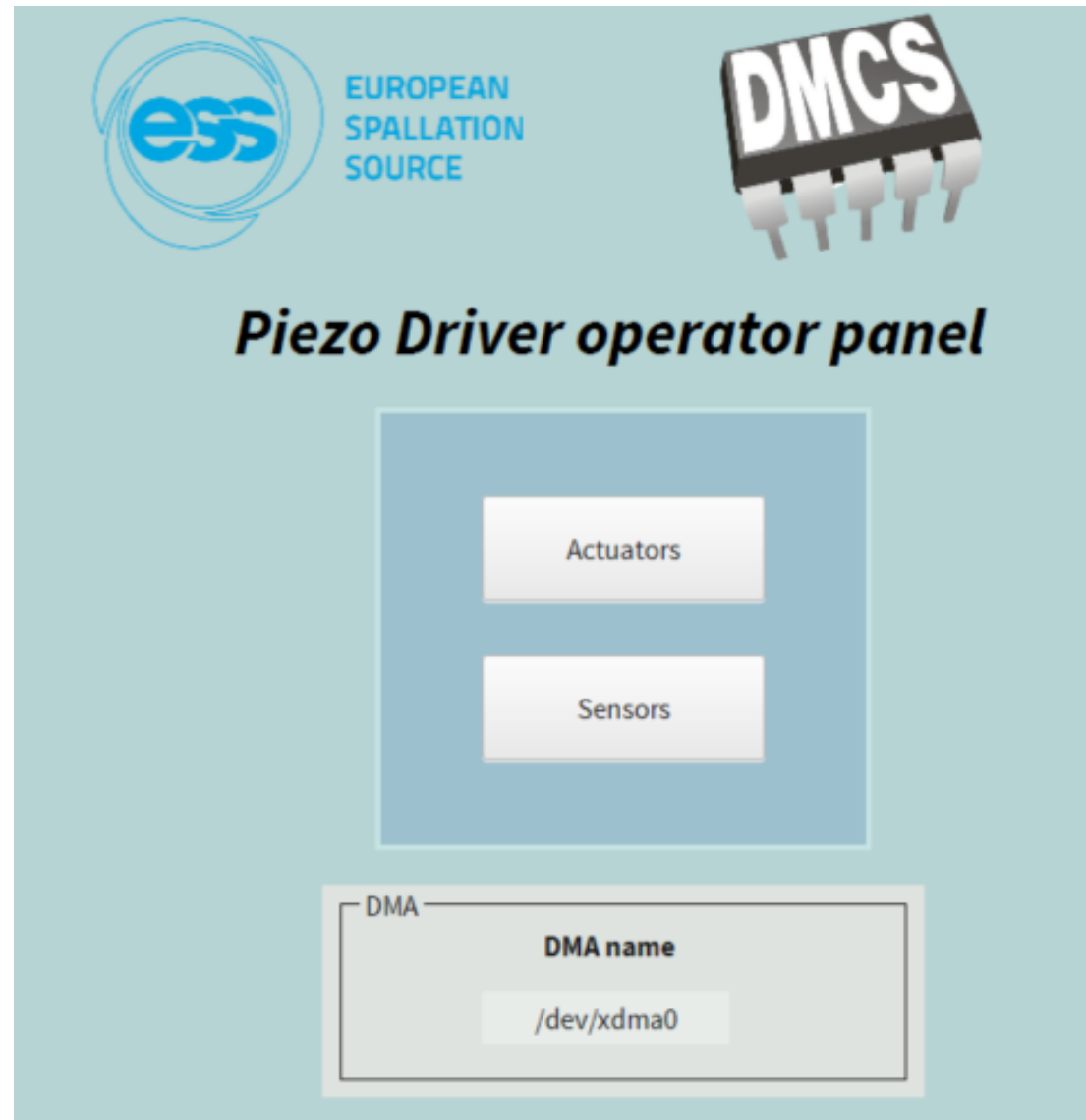


IOC Panels

EPICS
Software

● Composed of three windows:

- Main panel
- Sensor panel
- Actuators panel





Sensor panel

EPICS Software

- Possibility of changing mode of the specific channel
- Current data received from piezo (continuous or single mode)
- Saving acquired data to .csv file
- Setting all parameters concerning sensor mode (number of samples, trigger type etc.)

The screenshot shows the 'Sensor panel' interface. At the top, there are logos for 'ESS EUROPEAN SPALLATION SOURCE' and 'DMCS'. The title 'Sensor panel' is centered. Below the title, there are two channel mode controls: 'Channel A mode' and 'Channel B mode'. Each has a 'Sensor mode' indicator (green circle) and a 'Driver mode' indicator (dark green circle). A central graph displays 'Voltage/Current' vs 'Sample number' for four channels: Channel A current [A], Channel A voltage [V], Channel B voltage [A], and Channel B current [V]. The graph shows a flat line at approximately 10000 for Channel A current and Channel B current, and a flat line at 0 for Channel A voltage and Channel B voltage. On the left, there is a 'Save to file' panel with a 'Save set' checkbox, a 'Number of cycles to be saved' input field (set to 0), a 'Save to file' button, a 'Select directory' button, a 'Path to save file' input field, and a 'Type file name' input field. At the bottom, there is a control panel with 'Display mode' (Single/Advanced), a 'Capture signal' button, and various trigger parameters: 'Number of samples' (1024), 'Delay' (0 ms), 'Samples to drop' (0), 'Trigger type' (PCD_TR_FREE), 'Trigger length' (0.00 us), 'Trigger counter' (0), 'Number of triggers during 5s' (0.0), and a 'Calibrate trigger' button. A 'Main menu' button is in the bottom right corner.



Actuators panel

EPICS Software

- Possibility of changing mode of the specific channel
- Predefined waveforms (sine, triangle square)
- Loading waveform from .csv file
- Setting all parameters concerning actuators mode (frequency, offset, trigger type etc.)

The screenshot displays the 'Actuators panel' interface. At the top, there are logos for ESS (European Spallation Source) and DMCS. The interface is split into two columns for Channel A and Channel B. Each column has a mode selector (Channel A mode / Channel B mode) and an alarm indicator (Alarm channel A / Alarm channel B) showing 'No error'. Below the mode selectors are buttons for 'Sensor mode' and 'Driver mode'. The main part of each column is a graph showing the waveform: Channel A shows a sine wave, and Channel B shows a triangle wave. Below each graph is a control panel with the following parameters: Source of waveform (CSS), Amplitude (5.00 V), Frequency (1000.00 Hz), Offset (0.00 V), Wave type (W_SIN for Channel A, W_TRI for Channel B), Trigger type (PCD_TR_NONE), and Trigger delay (0.0000 ms). There are also buttons for 'Load wave to piezo', 'Reset circuit protection', and 'Main menu'.



Repository

EPICS
Software

Code on ESS gitlab:

<https://gitlab.esss.lu.se/kacperklys/e3-piezodriver>

e3-PiezoDriver Project ID: 1600

Star 0 Fork 0 Clone

50 Commits 1 Branch 0 Tags 2.2 MB Files

PiezoDriver IOC for ESS

master e3-piezodriver / +

History Find file Web IDE

Fix bug with destructor procedure b0c4f907
KacperKlys authored 1 month ago

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Enable Auto DevOps Add Kubernetes cluster

Set up CI/CD

Name	Last commit	Last update
cmds	Clean code and add debug printf	2 months ago
configure	Clean code and add debug printf	2 months ago
dkms	Temp Commit with some changes	2 months ago
docs	Init..e3-PiezoDriver	5 months ago
e3-PiezoDriver-loc/PiezoDriverApp	Fix bug with destructor procedure	1 month ago
ics-pcd-lib @ 8a55070b	added driver files as submodule	2 months ago
ics-xdriver-core @ 9975a71b	added driver files as submodule	2 months ago
ics-xdriver-lib @ f0d0c1ee	added driver files as submodule	2 months ago



Documentation

EPICS
Software

Documentation on confluence:

<https://confluence.esss.lu.se/display/HAR/Piezo-driver+integration+-+IOC>

● How to import .bob project

● How to install module

● How to use it

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
```



Thank you