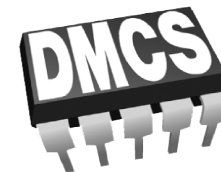




RTM
Carrier
Status

RTM Carrier HW/FW/SW Status

Presenter:
Aleksander Mielczarek, DMCS



2019-03-22, Lund



RTM
Carrier
Status

RTM Carrier Status

**Part 1 of 3:
Hardware**



RTM Carrier Hardware Status

RTM Carrier Status

- Evaluated 2 prototype revisions,
- Small improvements in MMC area,
- No changes to FPGA surroundings,
- Prepared production of 50 boards,
- Assembly – next month,
- Major components already at NCBJ.





RTM
Carrier
Status

RTM Carrier Status

**Part 2 of 3:
Firmware**



RTM Carrier Firmware – Overview

RTM Carrier Status



 `feature-piezo_rtm`

 `origin/feature-piezo_rtm`

 `origin/develop`

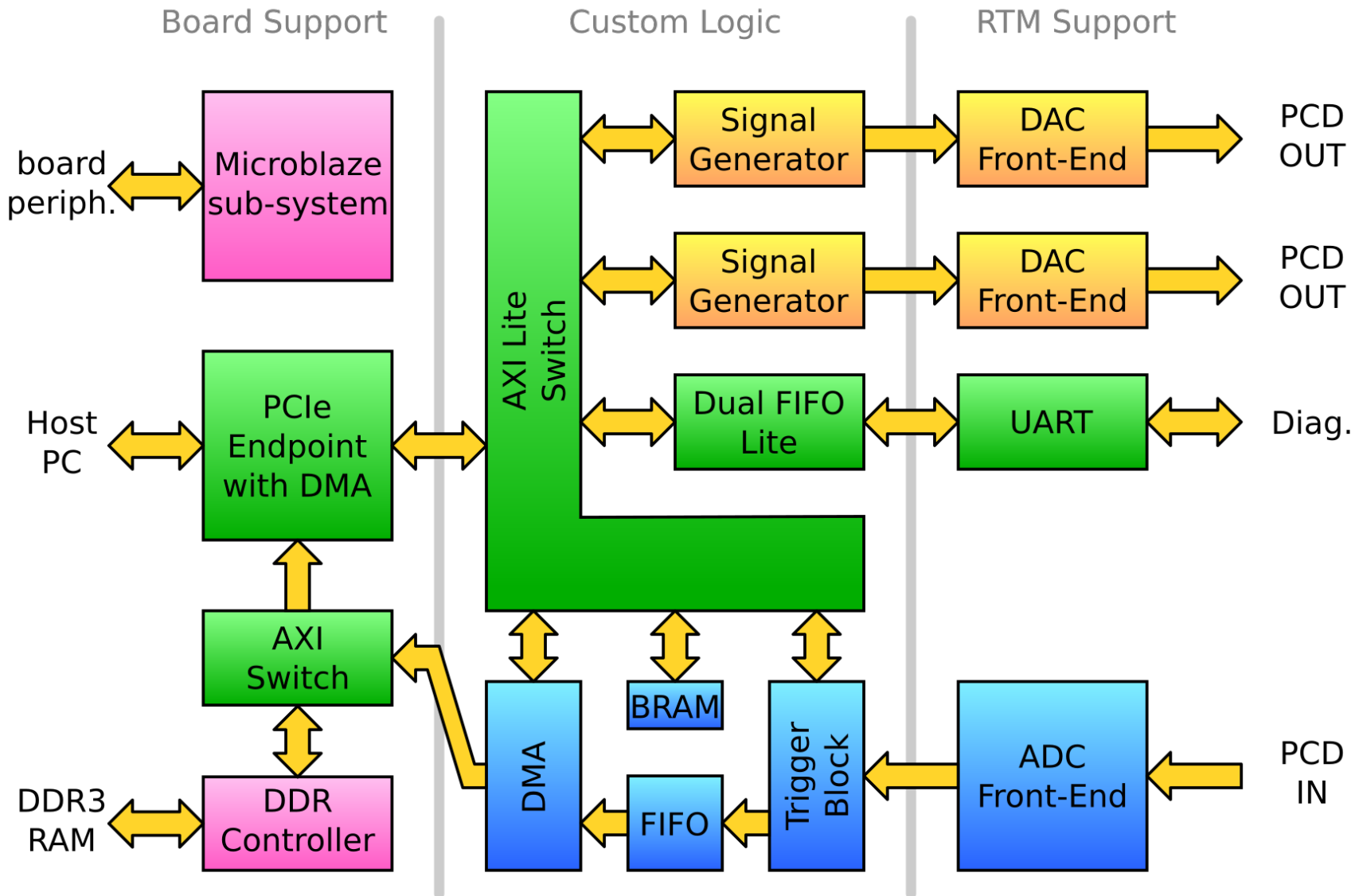
 `origin/feature-rtm_carrier_bsp`

- Code is based on the FPGA Firmware Framework,
- It is stored in branch **feature-piezo_rtm** of:
https://bitbucket.org/europeanspallationsource/ess_fpga_framework/src/
- A bit behind the development branch,
- The firmware is divided into 4 parts:
 - Framework core (ESS),
 - RTM Carrier BSP (NCBJ),
 - Piezo Control Device RSP (DMCS),
 - RTM Carrier PCD Application (DMCS).



RTM Carrier Firmware – Block Diagram

RTM Carrier Status





RTM Carrier Firmware – Next Steps

RTM Carrier Status

- Use the external trigger in place of the “oscilloscope trigger”
- Use DDR3 memory for samples storage
- Integrate with current development branch
(how to generate framework_core_regbank/register_bank.cfg?)
- Update the BSP (requests to NCBJ):
 - Provide working memory controller
 - Fix the M-LVDS signal receiver



RTM
Carrier
Status

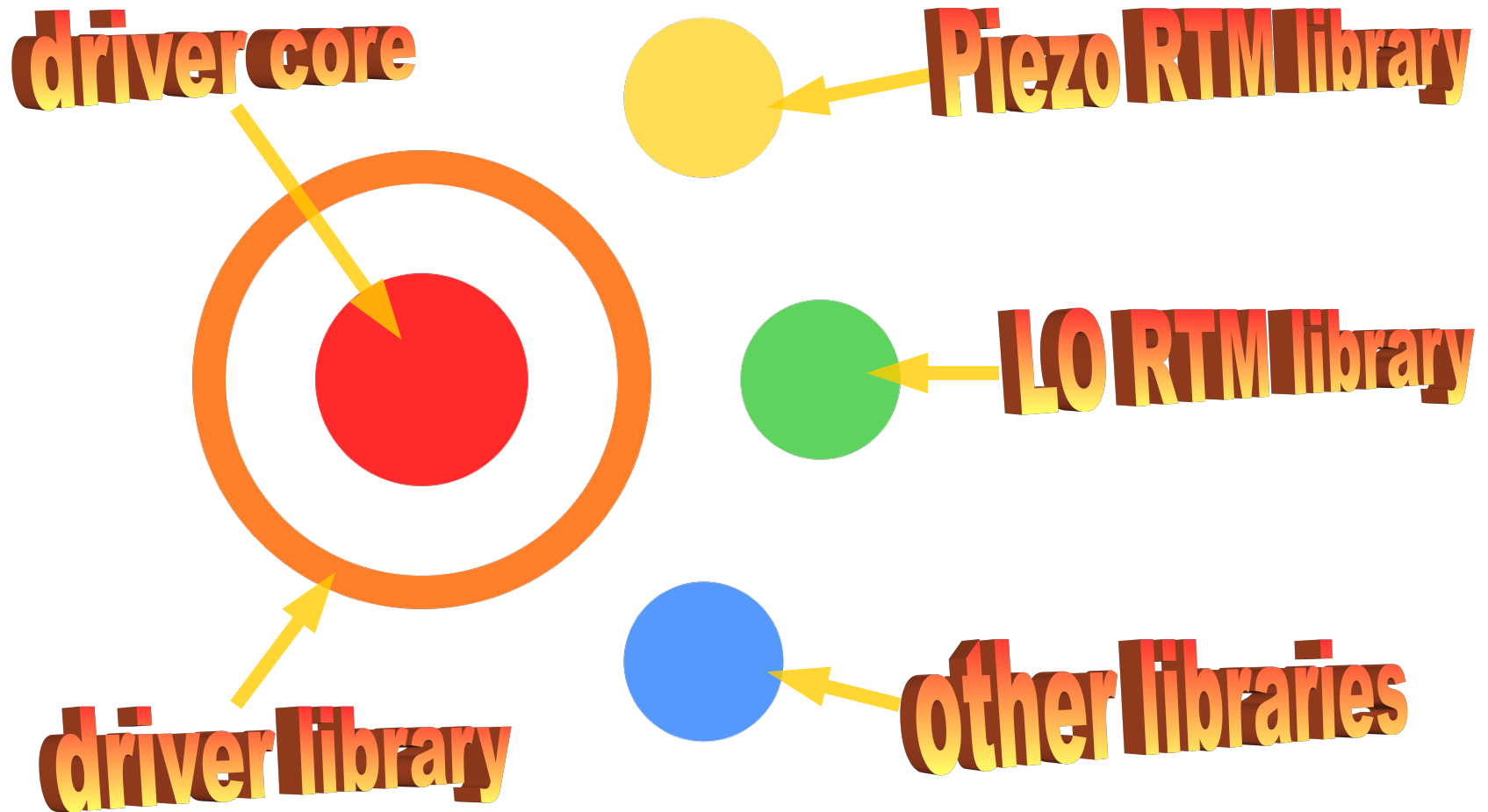
RTM Carrier Status

**Part 3 of 3:
Software**



RTM Carrier Software

RTM
Carrier
Status



<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/>



RTM Carrier Software – Driver



RTM Carrier Status

- Base code – Xilinx XDMA driver:
 - GPL-licensed
- Only two modifications:
 - Getting the address space size (useful for mmap)
 - Support of seeking in the register space



RTM Carrier Status

- The driver library:
 - Manages open file descriptors
 - Handles register access
 - Provides DMA file descriptors
 - Includes simple register read/write application



RTM Carrier Status

- `xildev * xil_open_device(const char *device_base_name);`
- `int xil_close_device(xildev *dev);`
- `unsigned int xil_get_bar_size(xildev *dev);`

- `int xil_get_c2h_fd(xildev *dev, unsigned int channel);`
- `int xil_get_h2c_fd(xildev *dev, unsigned int channel);`

- `int xil_read_reg(xildev *dev, uint32_t address, uint32_t *value);`
- `int xil_write_reg(xildev *dev, uint32_t address, uint32_t value);`



RTM Carrier Status

```
$ ./xil-rw -h
ESS Xilinx Driver register read/write utility
Read : ./xil-rw <dev_name> -r <address> [<count>]
Write: ./xil-rw <dev_name> -w <address> <value>
```

```
Example: ./xil-rw /dev/xdma0 -r 0x80000 16
```

```
$ ./xil-rw /dev/xdma0 -r 0x80000 4
User BAR size 4194304
[0x00080000] = 0x00000000
[0x00080004] = 0x00000000
[0x00080008] = 0x00000004
[0x0008000C] = 0x00000000
```



RTM Carrier Status

- The PCD library:
 - Provides API for piezo actuator
 - Provides API for piezo sensor
 - Provides functions to communicate with the diagnostic processor on the RTM
 - Includes two applications: one for generator mode, one for sensor mode



RTM Carrier Software – PCD Library



RTM Carrier Status

- `int pcd_gen_set_active(xildev *dev, int channel, bool enabled);`
- `int pcd_gen_get_active(xildev *dev, int channel, bool *enabled);`
- `int pcd_gen_set_amplitude(xildev *dev, int channel, float amplitude);`
- `int pcd_gen_get_amplitude(xildev *dev, int channel, float *amplitude);`
- `int pcd_gen_set_offset(xildev *dev, int channel, float offset);`
- `int pcd_gen_get_offset(xildev *dev, int channel, float *offset);`
- `int pcd_gen_set_frequency(xildev *dev, int channel, float freq);`
- `int pcd_gen_get_frequency(xildev *dev, int channel, float *freq);`
- `int pcd_gen_set_trigger(xildev *dev, int channel, bool external);`
- `int pcd_gen_get_trigger(xildev *dev, int channel, bool *external);`
- `int pcd_gen_set_delay(xildev *dev, int channel, float delay);`
- `int pcd_gen_get_delay(xildev *dev, int channel, float *delay);`
- `int pcd_gen_load_samples(xildev *dev, int channel, int16_t *samples, unsigned num_samples);`



RTM Carrier Software – PCD Library



RTM Carrier Status

- `int pcd_osc_trigger_arm(xildev *dev, enum PcdTriggerType trig);`
- `int pcd_osc_set_pre_trigger(xildev *dev, bool enabled);`
- `int pcd_osc_get_pre_trigger(xildev *dev, bool *enabled);`
- `int pcd_osc_set_sample_count(xildev *dev, uint16_t sample_count);`
- `int pcd_osc_get_sample_count(xildev *dev, uint16_t *sample_count);`
- `int pcd_osc_set_trigger_channel(xildev *dev, uint8_t channel);`
- `int pcd_osc_get_trigger_channel(xildev *dev, uint8_t *channel);`
- `int pcd_osc_set_trigger_delay(xildev *dev, uint8_t trig_delay);`
- `int pcd_osc_get_trigger_delay(xildev *dev, uint8_t *trig_delay);`
- `int pcd_osc_set_sample_dropping(xildev *dev, uint8_t samples_to_drop);`
- `int pcd_osc_get_sample_dropping(xildev *dev, uint8_t *samples_to_drop);`
- `int pcd_osc_set_trigger_level(xildev *dev, int16_t level);`
- `int pcd_osc_get_trigger_level(xildev *dev, int16_t *level);`
- `int pcd_osc_read_data(xildev *dev, SensorSample *storage, uint32_t num_samples);`



RTM Carrier Status

- `int pcd_init(xildev *dev);`
- `const char * pcd_error_string(int error_code);`

- `int pcd_mcu_clear(xildev *dev);`
- `int pcd_mcu_tx(xildev *dev, uint8_t *tx_data, uint16_t num_bytes);`
- `int pcd_mcu_rx(xildev *dev, uint8_t *rx_data, uint16_t max_num_bytes);`

- `int pcd_dma_reset(xildev *dev);`
- `int pcd_dma_write_descriptor(xildev *dev, uint8_t desc_no, uint64_t addr, uint32_t length);`
- `int pcd_dma_start(xildev *dev, uint8_t num_descriptors);`
- `int pcd_dma_stop(xildev *dev);`
- `int pcd_dma_is_running(xildev *dev, bool *result);`



RTM Carrier Status

```
./pcd-gen -h
```

ESS Piezo Control Device - generator test utility

Usage:

```
./pcd-gen [-d <dev_file>] [-m <command>] [-c <channel>] [-p] [-ADFOPSTW <param>]
```

Arguments:

- c <channel> - Use specified device channel: A or B
- d <dev_file> - Use the specified device file, default is /dev/xdma0
- m <command> - Exchange serial data with the diagnostic MCU
- p - Print current channel config
- A <ampl> - Set the desired amplitude (volts)
- D <time> - Set the desired delay after trigger (ms)
- F <freq> - Set the desired repetition frequency (Hz)
- O <offset> - Set the desired offset voltage (volts)
- P <power> - Enable or disable the channel (on/off)
- S <mode> - Switch selected channel to desired mode (sen/act)
- T <trigger> - Set the desired trigger type (int/ext)
- W <name> - Load the selected waveform (sin/tri/sqr)

Examples:

- Getting diagnostic MCU status:
./pcd-gen -m s
- Getting diagnostic MCU readouts:
./pcd-gen -m csv



RTM Carrier Status

```
./pcd-osc -h
```

ESS Piezo Control Device - oscilloscope test utility

Usage:

```
./pcd-osc [-d <dev_file>] [-c <file_name>] [-p] [-CDgGLPRST <param>]
```

Arguments:

- c <file_name> - Use the provided CSV file name, default is out.csv
- d <dev_file> - Use the specified device file, default is /dev/xdma0_user
- p - Print current config
- C <channel> - Set the trigger channel, range 0..3
- D <time> - Number of 64-cycle periods to wait after the trigger
- g - Try to print captured data using gnuplot
- G <count> - Try to read given number of samples from buffer
- L <level> - Set the trigger level, expressed in ADC counts
- P <enable> - Enable or disable the pretrigger (on/off)
- R <number> - Set the number of samples to drop between two stored ones
- S <count> - Set the number of samples to capture after trigger
- T <type> - Arm the trigger, type can be: R,F,B,I

Supported trigger types:

- R - rising edge
- F - falling edge
- B - both edges
- I - immediate



RTM Carrier Software – Next Steps

RTM
Carrier
Status

Open questions:

- Trigger scheme:
how to provide delayed triggers for piezo?
- Periodic signals generation:
is it needed, probably no?
- Packaging:
can we use plain RTM package based on makefile?



RTM
Carrier
Status

That's it.