The RFPI main function Software layer – library and CLI

Wojciech Tylman

About Me

Wojciech Tylman

D.Sc. In Computer Science, Ph.D. in Electronics

Role

C++ software engineer

Relevant experience

25+ years of experience in C++ programming

Design and implementation of SIL-4 software for rail traffic control and signaling, numerous projects.

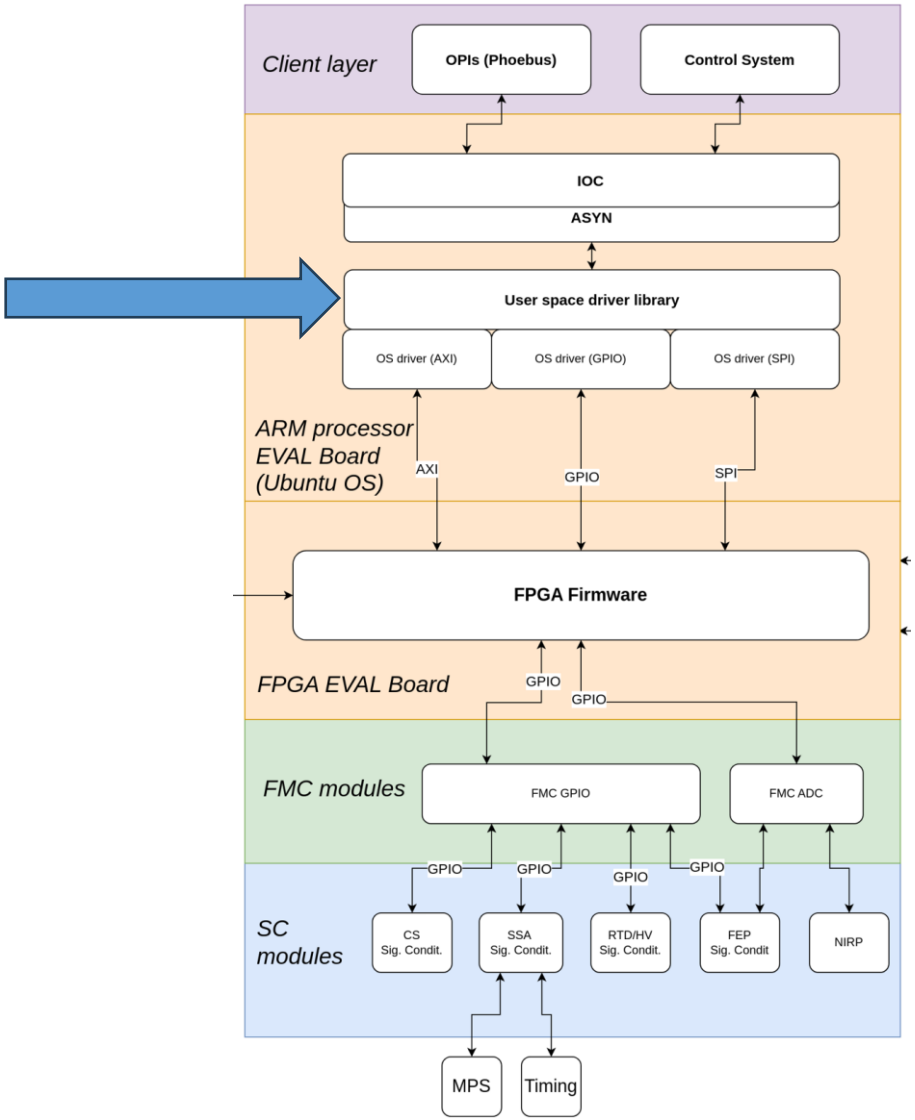
Agenda

- The main requirements,
- Specification and scope,
- The functionality and design details,
- Implementation.

The main requirements

- Software Layer acts as a middle layer between the firmware/hardware and IOC
  - Is the only way that IOC may interact with the system
  - Makes the IOC independent of the hardware details
- Unlike in the management software, no logic must be implemented, the software is mainly a pass-through interface
- Compile-time configuration of addresses, bit numbers etc. should be possible in a way to minimize mistakes
- Where run-time configuration is required, the software writes/reads hardware registers, software itself is stateless
- Command Line Interface (CLI) tool is welcome, to facilitate development and testing

The main requirements

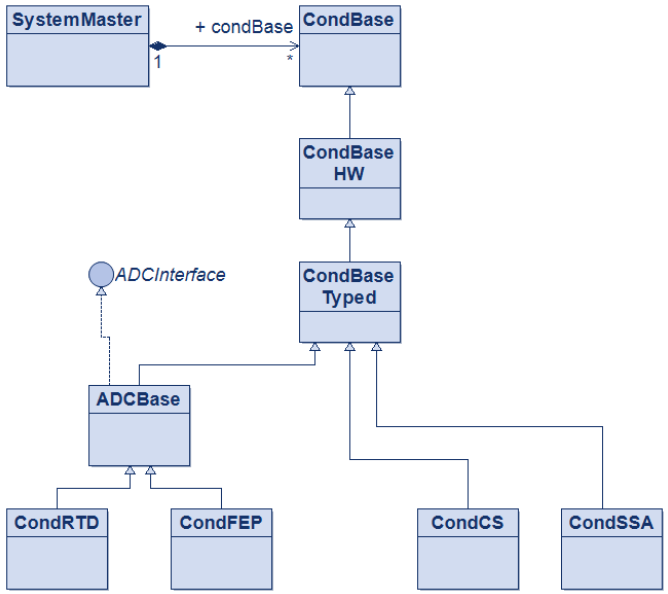


Specification and scope

- Targets the current design of the extension boards

- Structure allowing easy expansion of the software should new extension boards appear

Design details – class structure



Design details – CLI commands – selection

```
Commands common for all boards:
  -r --read          read register
  -w --write         write (reg:val) register
  --reset
SSA commands:
  -s --state         set state of the control register
  -c --clear         clear state of the control register
  --get-llrf-per     get low level rf permit
  --get-ssaf-per     get SSA permit flag
  --get-ssadc-per    get SSA DC permit flag
  --get-ssa-rdy      get SSA ready flag
RTD commands:
  --reset-prot       reset latched state
  --reset-count      reset counters
  --adc-cfg          get/set ADC configuration
  --dac              get/set DAC value
  --pga              get/set PGA value
  --hv-volt          HV coupler voltage
  --hv-curr          HV coupler current
  --temp             temperature
  --hv-volt-mask     HV coupler voltage mask
  --hv-curr-mask     HV coupler current mask
  --temp-mask        temperature mask
  --hv-cpl-v-thr     HV coupler voltage thresholds
  --hv-cpl-c-thr     HV coupler current thresholds
  --temp-thr         temperature thresholds
  --hv-cpl-v-scale   HV coupler voltage scaling
  --hv-cpl-c-scale   HV coupler current scaling
  --temp-scale       temperature scaling
  --hv-cpl-v-trip-cnt  HV coupler voltage trip count
  --hv-cpl-c-trip-cnt  HV coupler current trip count
  --temp-trip-cnt      temperature trip count
  --hv-cpl-v-trip-span  HV coupler voltage trip span
  --hv-cpl-c-trip-span  HV coupler current trip span
  --temp-trip-span      temperature trip span
CS commands:
  --get-he-prs       get helium pressure bit
  --get-cplr-v       get coupler vacuum
```

Implementation – libraries, tools

- g++ native compilation – ARM (Ubuntu)