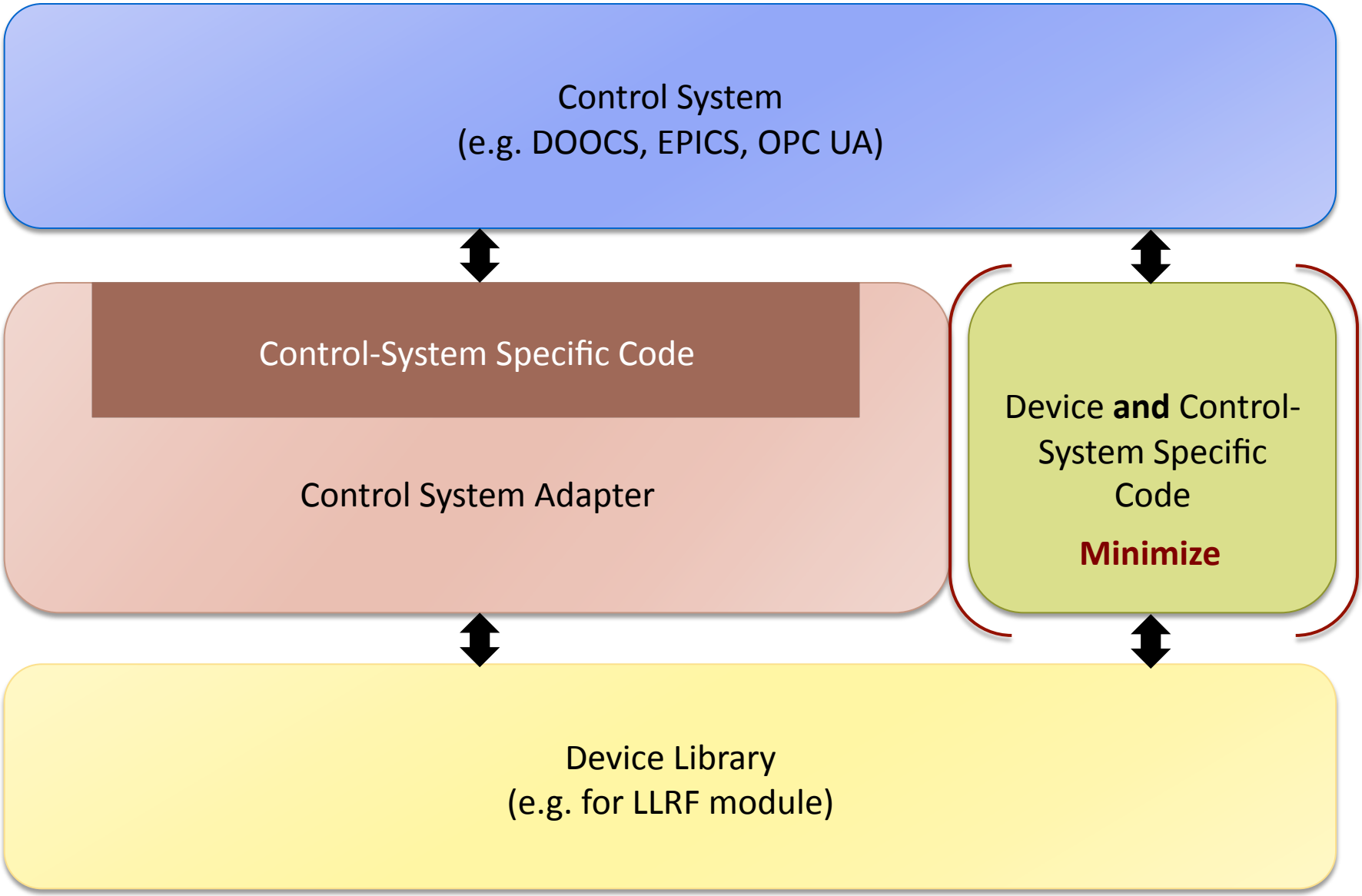
aquenos
software & more

Timing and Interrupt Issues for the MTCA4U Control System Adapter

ARD-ST3 Mini-Workshop: “SRF controls and CW operation”
November, 24th–25th 2015

Sebastian Marsching (aquenos GmbH)
Martin Hierholzer, Martin Killenberg (DESY)


Control System
(e.g. DOOCS, EPICS, OPC UA)
Control-System Specific Code
Control System Adapter
Device **and** Control-System Specific Code
Minimize
Device Library
(e.g. for LLRF module)

Different processing models

- Synchronous ⇔ Asynchronous
- Single threaded ⇔ Multi threaded
- Global mutex ⇔ Mutex per process variable

Control System Adapter PV Synchronization

- Asynchronous
- Semi multi-threaded
  - Up to one thread for the device library
  - Up to one thread for the control system
- Lock free
  - No risk of dead locks regardless of the control system's locking model

High Precision Timing

- Latency in non real-time Linux can be in the order of **seconds**!
- Real-time code can have microseconds latency, but required optimized code paths (including all involved hardware drivers)
- Reliable high-precision timing must be implemented in hardware!

Best Effort Timing

- Synchronization through software possible when best-effort latency is sufficient.
- Most events delivered within a few **milliseconds**.
- Some events will take much longer.
- Example:
  - Start of diagnostics data processing.
  - New data available notification.

Issues in the Control System Adapter

- Busy waiting needed (otherwise, we would depend on a mutex).

Latency ⇔ CPU load

- Alternative: Out-of-band notification (e.g. function call).
  - Increased the amount of code that depends on both device and control-system.
- How to synchronize different hardware components?

LLRF System Timing Issues

- Goal for the future: Get event directly from LLRF board
  - Required firmware and kernel driver changes
- Solution for now: Get event from separate hardware:
  - x1Timer / x2Timer
  - MRF EVR board
- RF pulse is already in hardware, issue only exists for readout and data processing triggers.

Summary

- Control system adapter uses a lock-free, asynchronous synchronization mechanism.
- Tradeoff between latency and CPU consumption.
- Out-of-band notification through function calls might be suitable for most applications.

Discussion