

Integrating Control Applications Into Different Control Systems.

The MTCA4U Control System Adapter



Martin Killenberg

M. Hierholzer, C. Schmidt, *DESY, Hamburg, Germany*

S. Marsching, *aquenos GmbH, Baden-Baden, Germany*

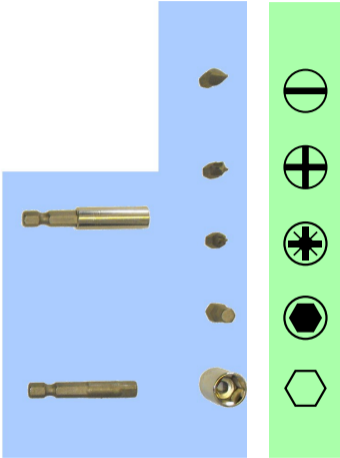
A. Piotrowski, *FastLogic Sp. z o.o., Łódź, Poland*

J. Wychowaniak, *Łódź University of Technology, Łódź, Poland*

20th October 2015

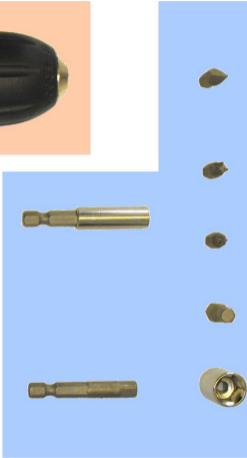
ICALEPCS 2015, Melbourne, Australia







Device



Adapter



Control System



Task

Complex control algorithms should be used with different control systems.

Task

Complex control algorithms should be used with different control systems.

Requirements For Abstraction

- Keep application code control system independent
- The algorithm must interact with the control system
- Use functionality provided by the control system
- Minimise device-dependent code on the control system side

Additional Requirements:

- Thread-safety
- Real-time capability
- Must not copy large data objects (arrays)

Task

Complex control algorithms should be used with different control systems.

Requirements For Abstraction

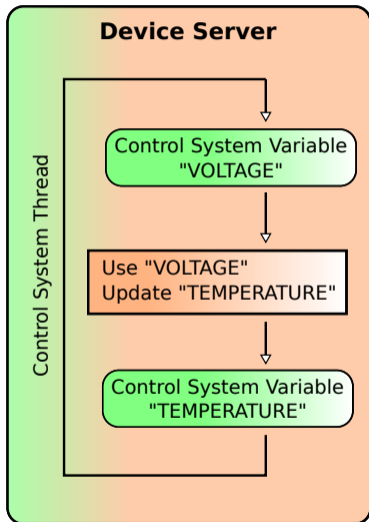
- Keep application code control system independent
- The algorithm must interact with the control system
- Use functionality provided by the control system
- Minimise device-dependent code on the control system side

Additional Requirements:

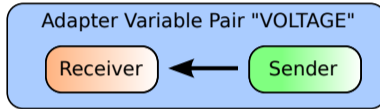
- Thread-safety
- Real-time capability
- Must not copy large data objects (arrays)

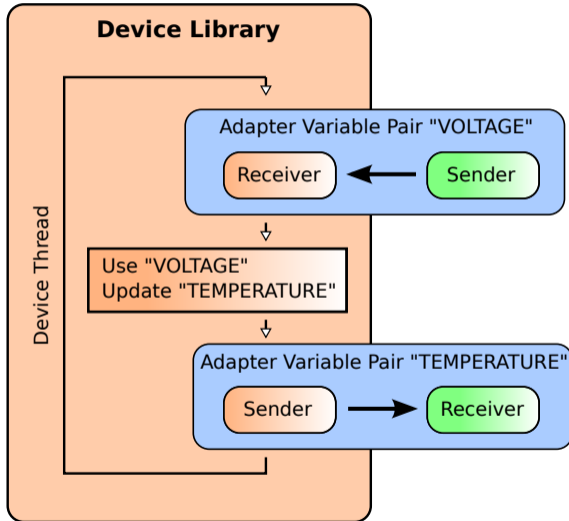
First Implementation

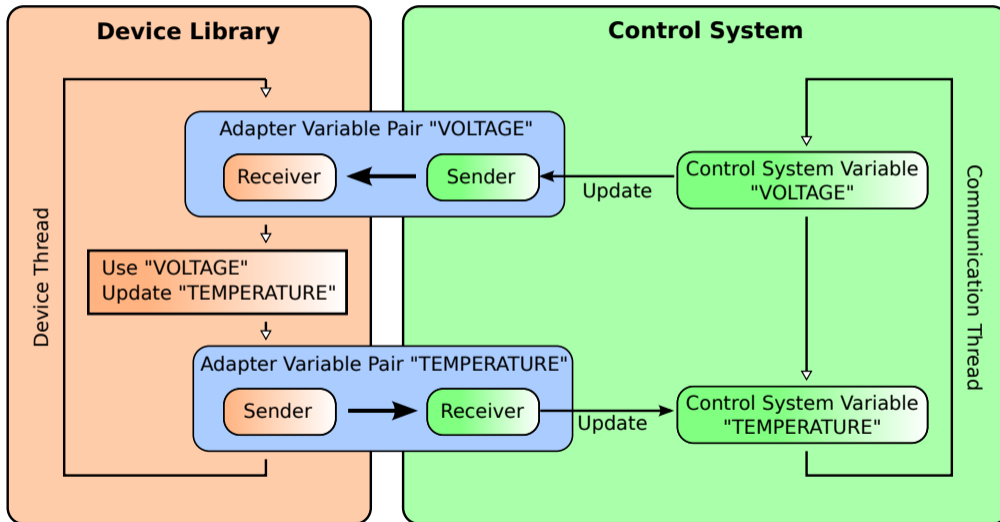
- Process variables to transfer data to/from the control system

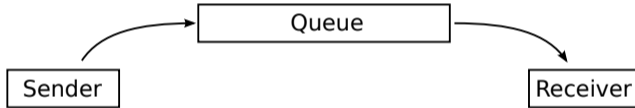


- Control system data types used inside the algorithm
- Control system variables can be locking/blocking
- Control system variables might not be thread safe
- Threading often handled by control system

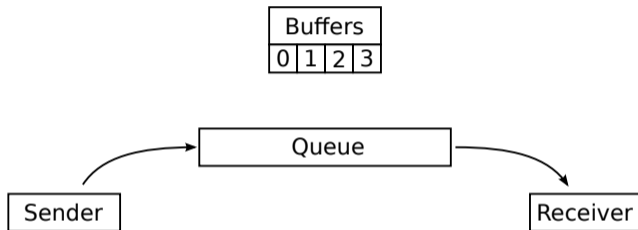




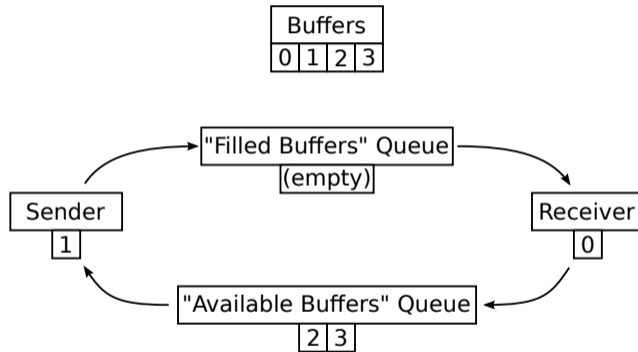




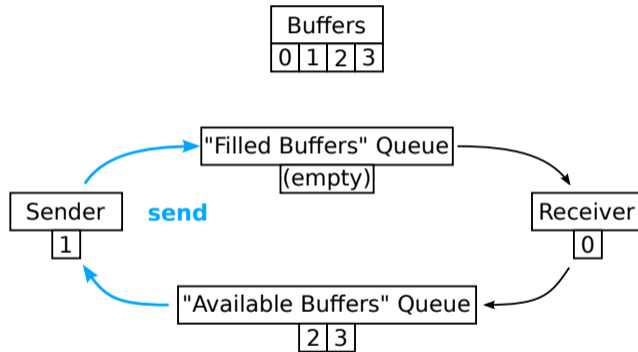
- Lock-free queue



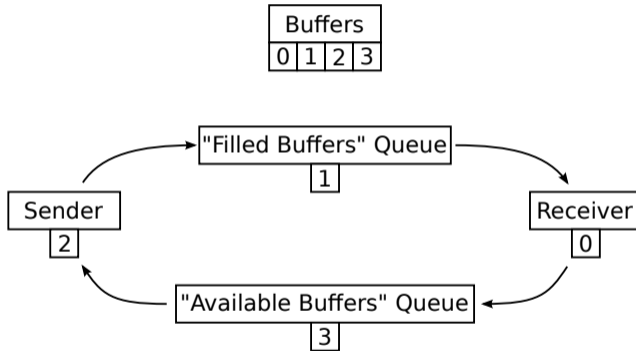
- Lock-free queue
- Pre-allocated buffers for arrays



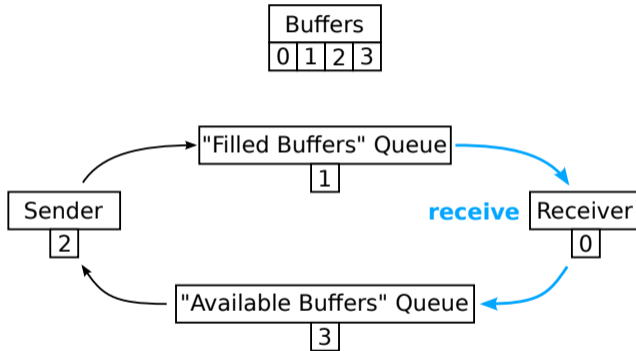
- Lock-free queues
- Pre-allocated buffers for arrays
- Copy references, not buffers



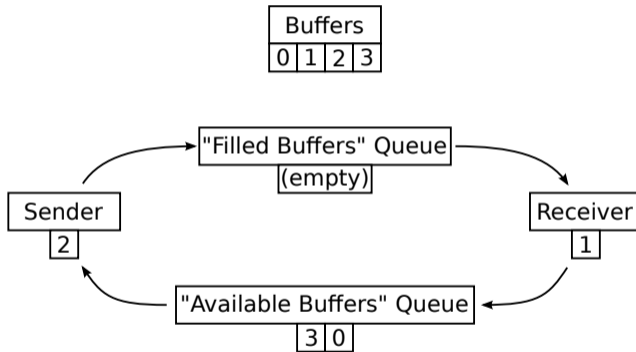
- Lock-free queues
- Pre-allocated buffers for arrays
- Copy references, not buffers



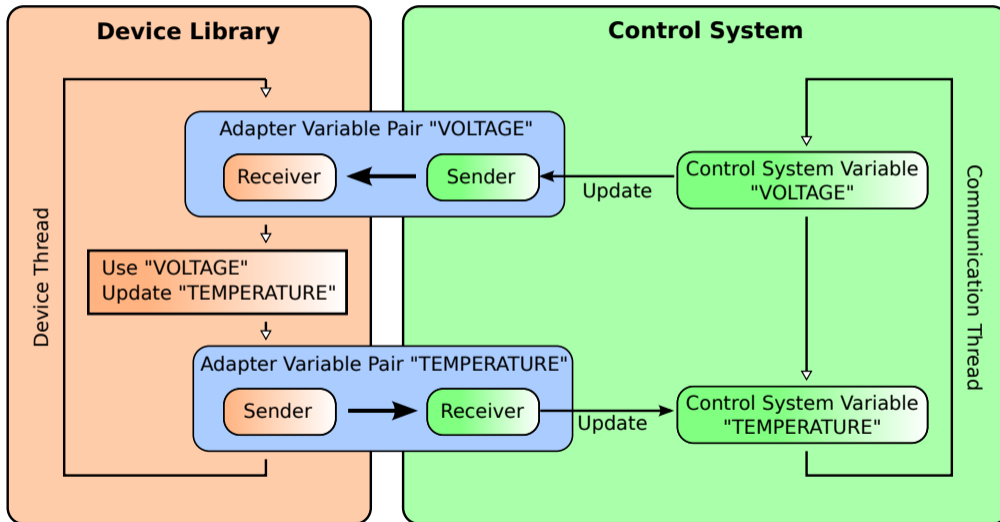
- Lock-free queues
- Pre-allocated buffers for arrays
- Copy references, not buffers

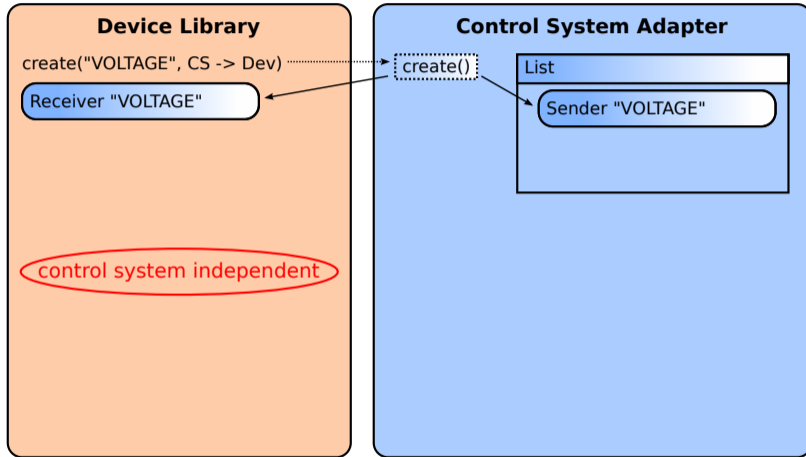


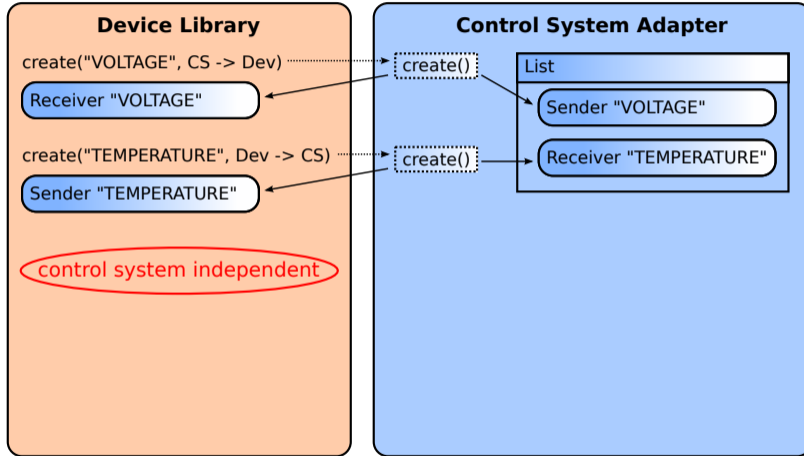
- Lock-free queues
- Pre-allocated buffers for arrays
- Copy references, not buffers

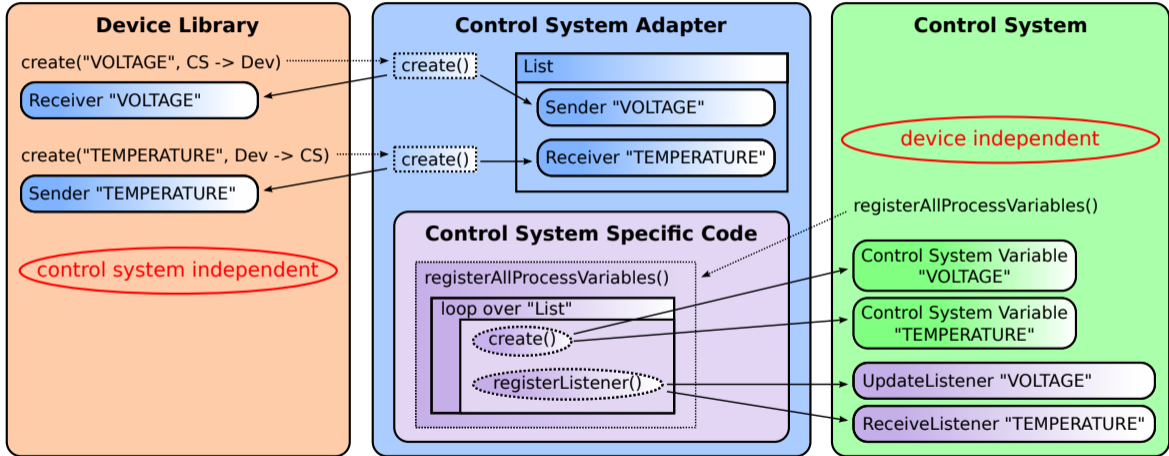


- Lock-free queues
- Pre-allocated buffers for arrays
- Copy references, not buffers









Adapter for process variables

- Generic part
- Control system specific part
 - Implementations for DOOCS and EPICS

Design Goals

- Control system independent process variables ✓
- Thread safety ✓
- Real time capability ✓
- Minimise copying ✓
- Minimise device-dependent code on control system side (✓)

Access to control system features

- Limits
- History
- Engineering units

Implementations are very different in the various control systems!

- Discussions how to put this into the adapter

MTCA4U Control System Adapter

- Adapter to use device logic with different control systems
- Implementations for DOOCS and EPICS exist
- Planned: support for OPC-UA

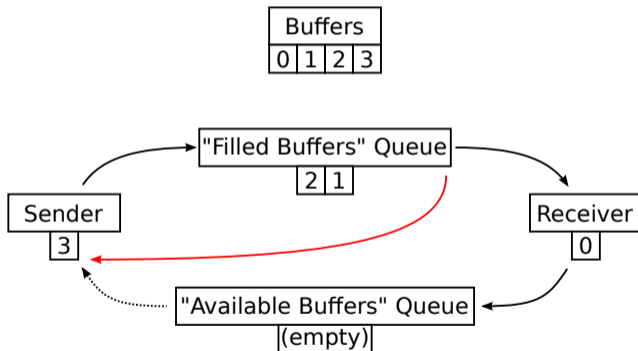
Software Repositories

All software is published under the GNU General Public License.

- MTCA4U Control System Adapter: <https://svnsrv.desy.de/public/mtca4u/ControlSystemTools/>
- EPICS extension: <http://oss.aquenos.com/svnroot/epics-mtca4u/>
- DOOCS extension: https://svnsrv.desy.de/desy/mtca4u_applications/D00CS_Adapter/

Backup

Update the queue if the receiver is slow/down



- No free buffers for the sender
- Overwrite the oldest buffer
- Pop the head of the "filled buffers" queue (buffer 1)
- Send the buffer which has just been filled (buffer 3)