FRM II
Forschungs-Neutronenquelle
Heinz Maier-Leibnitz

MLZ
Heinz Maier-Leibnitz Zentrum

Benefits from standards in instrument control

Jens Krüger

MLZ is a cooperation between:

JÜLICH FORSCHUNGSZENTRUM

Helmholtz-Zentrum Geesthacht
Zentrum für Material- und Küstenforschung

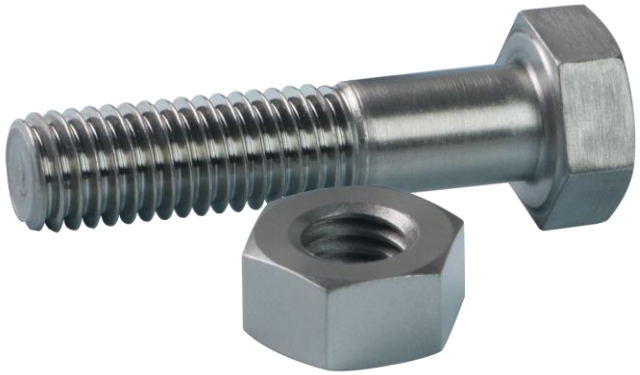
Technische Universität München

Introduction

- Importance of standards in instrument control
- 2 components of instrument control
  - Hardware
  - Software

Standards in Mechanics

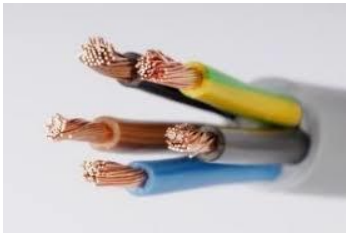
Standards in Mechanics



Standards in Mechanics

- Materials
  - Steel
  - Stainless steel
  - Aluminum
  - Brass
- Length
- Thickness
- Shape of head or nut

Standards in Electrics



Why standards in instrument control

- There are no common standards in instrument control
- A lot of neutron facilities around the world
- Each with their own standard, if exists
- Synchrotron facilities in the same situation

A way to define standards in instrument control

- Example: Motors and motor controllers

A way to define standards in instrument control

- Example: Motors and motor controllers
  - Select a pair of motor and motor controllers
  - One pair is simple, but normally many pairs
  - Wide range of application → matrix of options
  - Select a controller, which may drive more than one motor type
  - Majority of the applications
  - From remaining select next controller/motor(s) pairs and so on
  - Set of motor controllers as „standard“
  - Selection of controllers is facility specific, but look around

Standards in instrument control software

- Follow some rules
- Abstract hardware devices in software
- Define interface after requirements
- Example: Motor
  - Move from A to B
  - Stop movement
  - Set and Read Speed
  - Read position
  - …
- Hide specialties
- Don‘t mix operation with configuration

Components of instrument control

- PLC interface
- Entangle
- SeCOP
- NICOS

PLC interface

- Access via ModbusTCP / ADS (Beckhoff)
- Detection via Magic Word
- Catalog describing implementation, number and types of available devices
- HW-Abstraction into high level devices
- Addressing of predefined IO structures
- (Analog/Discrete In/Output with/out parameters, …)
- Set of predefined parameters for each device type
- Support for auxiliary strings (for humans)

Entangle

- Uses Tango (http://www.tango-controls.org)
- „device-oriented control toolkit for controlling any kind of hardware or software and building SCADA systems“
- Framework with well defined device interfaces
- Interfaces are stable since years
- About 20 interfaces:
  - Motor
  - Temperature Controller
  - Sensor
  - ...
- Implemented for a large number of hardware
- Hiding the specialties of concrete hardware
- Support for the PLC interface

Sample **E**nvironment **Co**mmunication **P**rotocol (SeCOP)

- International coordinated protocol (International Society for Sample Environment, ISSE)
- Interaction of sample environment from different facilities or vendors with local instrument control software
- Catalog describing modules and their properties (including hints for visibility and access level)
- Abstraction into high level devices
- Addressing of predefined IO interfaces via names
- (Readable, Movable, Drivable) with parameters
- Set of predefined parameters and properties
- Multi client support
- Synchronous/Asynchronous communication via TCP
- Devices can be stacked
- Detection via ‘*IDN‘
- First successful test at ILL with a furnace from MLZ

NICOS

- Networked Integrated Control System (http://www.nicos-controls.org)
- Standard user interface
- Standard components in GUI, devices, and commands
- Flexible configuration
- Script control
- Additional services
  - Data logging with history function
  - Electronic logbook
  - Status monitor (GUI and HTML)
  - Watchdog
  - ….

Conclusion

- Standards in instrument control are simple to define

Benefits

- Simple way to install instrument control
- High modularity
- Hardware abstraction makes implementation simpler
- Reduction of ‘unexpected‘ behavior
- Generic programming possible
- Better support for users and instrument responsible s
- Less specialized knowledge
- Simple way to add, remove, or exchange components

Benefits

- Hopefully, help to perform experiments with fantastic results