ROS - FIELD BUSES INTEGRATION REVIEW

Ludovic DELVAL
ludovic.delval@ipa.fraunhofer.de

Source: http://www.lamonde.com/acatalog/CLICK_PLCLCPU_Units.html
Field buses usage (Feb 2018)

Industrial Ethernet: 52% (46)
Annual growth: 22% (22)

PROFINET 12%
EtherCAT 7%
Modbus-TCP 4%
POWERLINK 4%
EtherNet/IP 15%
Other Ethernet 10%
WLAN 4%
Bluetooth 1%
Other Wireless 1%
Modbus-RTU 6%
CC-Link 6%
CANopen 4%
DeviceNet 4%
Other Fieldbus 10%
PROFIBUS DP 12%
Fieldbus: 42% (48)
Annual growth: 6% (4)
Wireless 6% (6)
Annual growth: 32% (32)

Content

- CANopen
- PROFINET
- EtherCAT
- Ethernet/IP
- Modbus-TCP
- PowerLink
- OPC-UA
ROS_CANopen:

Github stack:

https://github.com/ros-industrial/ros_canopen

ROS documentation:

http://wiki.ros.org/ros_canopen

Status: Maintained

Maintainer: Mathias Lüdtke - m0athias.luedtke@ipa.fraunhofer.de
Author: Florian Weisshardt - fmw@ipa.fhg.de
License: LGPL
ROS_CANopen

- **CAN drivers for motors**

  - canopen_402 – Implement the CiA Profile 402 (Tested on Schunk devices)
  - canopen_motor_node – Interface for canopen_402 and ros control
  - canopen_master – Implement the CiA DS 301 protocol
  - canopen_chain_node – ROS Node that manage a canbus
  - socketcan_interface – Generic CAN interface and socketcan driver (tested with PEAK-USB)
KaCanOpen
ROS // CANopen bridge
https://github.com/KITmedical/kacanopen (Maintained)
Documentation:
https://kitmedical.github.io/kacanopen/

CANopen for Python
Python implementation for CANopen
https://github.com/christiansandberg/canopen (Maintained)
Siemens_Experimental:
Github stack:  
https://github.com/ros-industrial/siemens_experimental
ROS documentation:  
http://wiki.ros.org/siemens_cp1616
Status: Maintained

Maintainer: Frantisek Durovsky – frantisek.durovsky@smartroboticsys.eu
Author: Frantisek Durovsky
License: Apache 2.0
Siemens_cp1616

- HARDWARE Dependant
- Use CP1616 driver available here: https://drive.google.com/file/d/0B5F9t8AjEoHyTnIxSjJtQlJDSVk
- Tutorial for setup:
  - http://wiki.ros.org/siemens_cp1616/Tutorials
- Easily setup and good examples for controller and device mode
PROFINET Python Implementation
Easy to use Python implementation for Profinet
https://github.com/devkid/profinet (Unmaintained)
Pr2Ethercat_drivers:

Github stack:

https://github.com/PR2-prime/pr2Ethercat_drivers

ROS documentation:

http://wiki.ros.org/pr2Ethercat_drivers?distro=kinetic

Status: Unmaintained

Maintainer: David Feil-Seifer - dave@cse.unr.edu
Author: Rob Wheeler – wheeler@willowgarage.com
Derek King – dking@willowgarage.com

License: BSD
Pr2_ethercat_drivers

- HARDWARE Dependant!

- Compatible with willow garage hardware:
  - WG005 – Main motor controller for Pr2 motion
  - WG006 – Main motor controller for Pr2 gripper
  - WG014 – EtherCAT Hub
  - WG021 – LED projector

- Drivers based on the SOEM / SOES library
  - https://github.com/OpenEtherCATsociety/SOEM
  - https://github.com/OpenEtherCATsociety/SOES
OROCOS RTT_SOEM

- Available here: https://github.com/orocos/rtt_soem (Unmaintained)

- Use the OROCOS tools chain to build a ethernet driver

- Create a EtherCAT Master
  - Edit the file test.ops by adding:
    - (Change interface if needed) `Master.ifname = "eth1" // will change the default interface`
    - `Master.configure() // configure the interface and sockets`
    - `Master.setPeriod(0.05) // Topics publishing periods`
    - `Master.start() // Start the EtherCAT Master`
    - (Example) `stream("Master.Slave_1002.bits", ros.topic("DI")) // will publish on the topic /DI the data`
ODVA_EthernetIP:

Github stack:

https://github.com/ros-drivers/odva_ethernetip

ROS documentation:

http://wiki.ros.org/odva_ethernetip

Maintainer status: Unmaintained

Maintainer: Mike Purvis – mpurvis@clearpathrobotics.com
Author: Kareem Shehata - kareem@shehata.ca
License: BSD
ODVA_EthernetIP

API:

- IO Scanner (WARNING - default port wrong)
- Session – Create a TCP and an UDP sockets
  - Services (TCP) – Call set/get single attribute
  - Receive/Send IO_packet (UDP)
Driver for Omron laser Scanner:

http://wiki.ros.org/omron_os32c_driver (Maintained)

OpENer:
OpENer is an EtherNet/IP stack for I/O adapter devices. It supports multiple I/O and explicit connections and includes objects and services for making EtherNet/IP-compliant products as defined in the ODVA specification.

https://github.com/EIPStackGroup/OpENer (Maintained)

CIPSter:
C++ porting of OpENer,
https://github.com/liftoff-sr/CIPster (Maintained)
Modbus:

Github stack:

https://github.com/HumaRobotics/modbus

ROS documentation:

http://wiki.ros.org/modbus

Status: Maintained

Maintainer: Sven Bock - sb@generationrobots.com
Wagdi Ben yaala - wb @generationrobots.com

Author: Sven Bock
License: BSD
Modbus

Prerequisites

Pymodbus

```
$ sudo apt-get install python-pymodbus
$ sudo apt-get install python-pyasn1 python-twisted-conch
```

Quickstart:

Client

```
rosrun modbus_wrapper modbus_client.py _ip:="localhost" _port:=1234
```

Server

```
rosrun modbus_wrapper modbus_server.py _port:=1234
```

Warning: Default port for Modbus (502) need admin right
Modbus

- Address / Types are hardcoded

- Example client:
  - Read/Write the HoldingRegister
    - Read (address 40000-40019)
    - Write (address 40020-40040)

To change register:
- client.write_coils (address, input)
- client.write_holding_registers (address, input)
- client.write_input_registers (address, input)
Pymodbus:
https://github.com/riptideio/pymodbus (Maintained)

Documentation:
Ros_powerlink:

Github stack:

https://github.com/SmartRoboticSystems/ros_powerlink

ROS documentation:

--------- Not released ---------

Status: ?

Maintainer: Jan Bacik - jan.bacik@smartroboticsys.eu
Author: Jan Bacik
License: BSD
ros_powerlink

- Dedicated to I/O modules
- Single node:
  - Create two topics
    - powerlink/data_to_plc (Subscriber)
    - powerlink/data_from_plc (Publisher)
ros_opcua_communication:

Github stack:

https://github.com/iirob/ros_opcua_communication

ROS documentation :

http://wiki.ros.org/ros_opcua_impl_freeopcua?distro=kinetic

Status: Maintained

Maintainer: Denis Štogl - denis.stogl@kit.edu
Author: Denis Štogl
License: LGPLv3
ros_opcua_communication

- Client: based on openlib FreeOPCUA (https://github.com/FreeOpcUa/freeopcuA)
  - C++ based
  - Create a list of services

- connect - Connect to the OPC UA server defined in request.
- disconnects - Disconnects from the OPC UA server.
- list_node - List child nodes from the node ID request. (Just one level under)
- call_method - Call OPC UA method.
- read – Read a value from node in OPC UA server.
- write - Write a value to the node in OPC UA server.
- subscribe - Subscribe to data-change event for the node in OPC UA server. Topic for publishing event will be automatically created.
- unsubscribe - Unsubscribe from a node in the OPC UA server.
Server: based on openlib Python-opcua ([https://github.com/FreeOpcUa/python-opcua](https://github.com/FreeOpcUa/python-opcua))

- Python based
- Create a server with topics as nodes / services as methods
# ROS Implementation

<table>
<thead>
<tr>
<th>Type</th>
<th>Repository</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANopen</td>
<td><a href="https://github.com/ros-industrial/ros_canopen">https://github.com/ros-industrial/ros_canopen</a></td>
<td>Maintained</td>
</tr>
<tr>
<td>EtherCAT</td>
<td><a href="https://github.com/PR2-prime/pr2_ethercat_drivers">https://github.com/PR2-prime/pr2_ethercat_drivers</a></td>
<td>Unmaintained</td>
</tr>
<tr>
<td>EthernetIP</td>
<td><a href="https://github.com/ros-drivers/odva_ethernetip">https://github.com/ros-drivers/odva_ethernetip</a></td>
<td>Unmaintained</td>
</tr>
<tr>
<td>Modbus TCP</td>
<td><a href="https://github.com/HumaRobotics/modbus">https://github.com/HumaRobotics/modbus</a></td>
<td>Maintained</td>
</tr>
<tr>
<td>Powerlink</td>
<td><a href="https://github.com/SmartRoboticSystems/ros_powerlink">https://github.com/SmartRoboticSystems/ros_powerlink</a></td>
<td>Not Released</td>
</tr>
<tr>
<td>OPC-UA</td>
<td><a href="https://github.com/iirob/ros_opcua_communication">https://github.com/iirob/ros_opcua_communication</a></td>
<td>Maintained</td>
</tr>
</tbody>
</table>
Fielbuses Open Library

- [https://github.com/oss-fieldbus](https://github.com/oss-fieldbus) - List most of open source library for fieldbuses
- [https://github.com/christiansandberg/canopen](https://github.com/christiansandberg/canopen) - CANopen python lib
- [https://github.com/CANopenNode/CANopenNode](https://github.com/CANopenNode/CANopenNode) - CANopen C lib
- [https://github.com/devkid/profinet](https://github.com/devkid/profinet) - Python Profinet Implementation
- [https://github.com/OpenEtherCATsociety/SOEM](https://github.com/OpenEtherCATsociety/SOEM) - EtherCATMaster C lib
- [https://github.com/OpenEtherCATsociety/SOES](https://github.com/OpenEtherCATsociety/SOES) - EtherCATSlave C lib
- [https://github.com/EIPStackGroup/OpENer](https://github.com/EIPStackGroup/OpENer) - EthernetIP C lib
- [https://github.com/liftoff-sr/CIPster](https://github.com/liftoff-sr/CIPster) - EthernetIP C++ lib
- [https://github.com/riptideio/pymodbus](https://github.com/riptideio/pymodbus) - Modbus Python lib
- [https://github.com/OpenAutomationTechnologies/openPOWERLINK_V2](https://github.com/OpenAutomationTechnologies/openPOWERLINK_V2) - POWERLINK C lib
- [https://github.com/FreeOpcUa/freeopcua](https://github.com/FreeOpcUa/freeopcua) - OPC-UA C++ lib
- [https://github.com/FreeOpcUa/python-opcua](https://github.com/FreeOpcUa/python-opcua) - OPC-UA Python lib