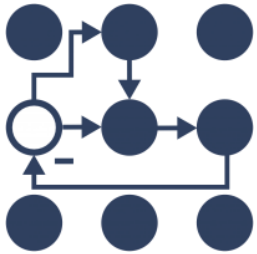


b»robotized

ros2_control goes industrial

CANopen

BECKHOFF TwinCAT ADS

ctrlX AUTOMATION

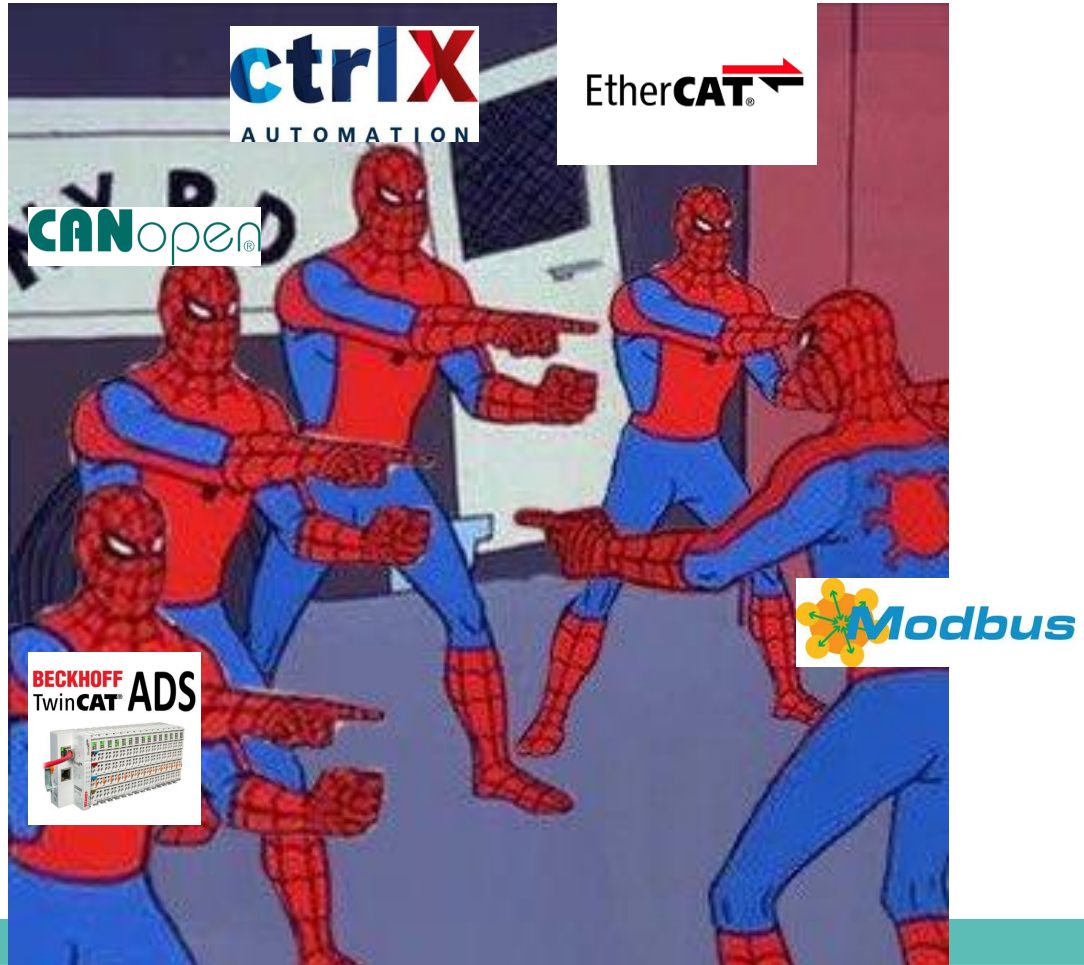
EtherCAT

Modbus

ROSCon DE & FR 2025 Strasbourg

Nikola Banović

Why are you here?



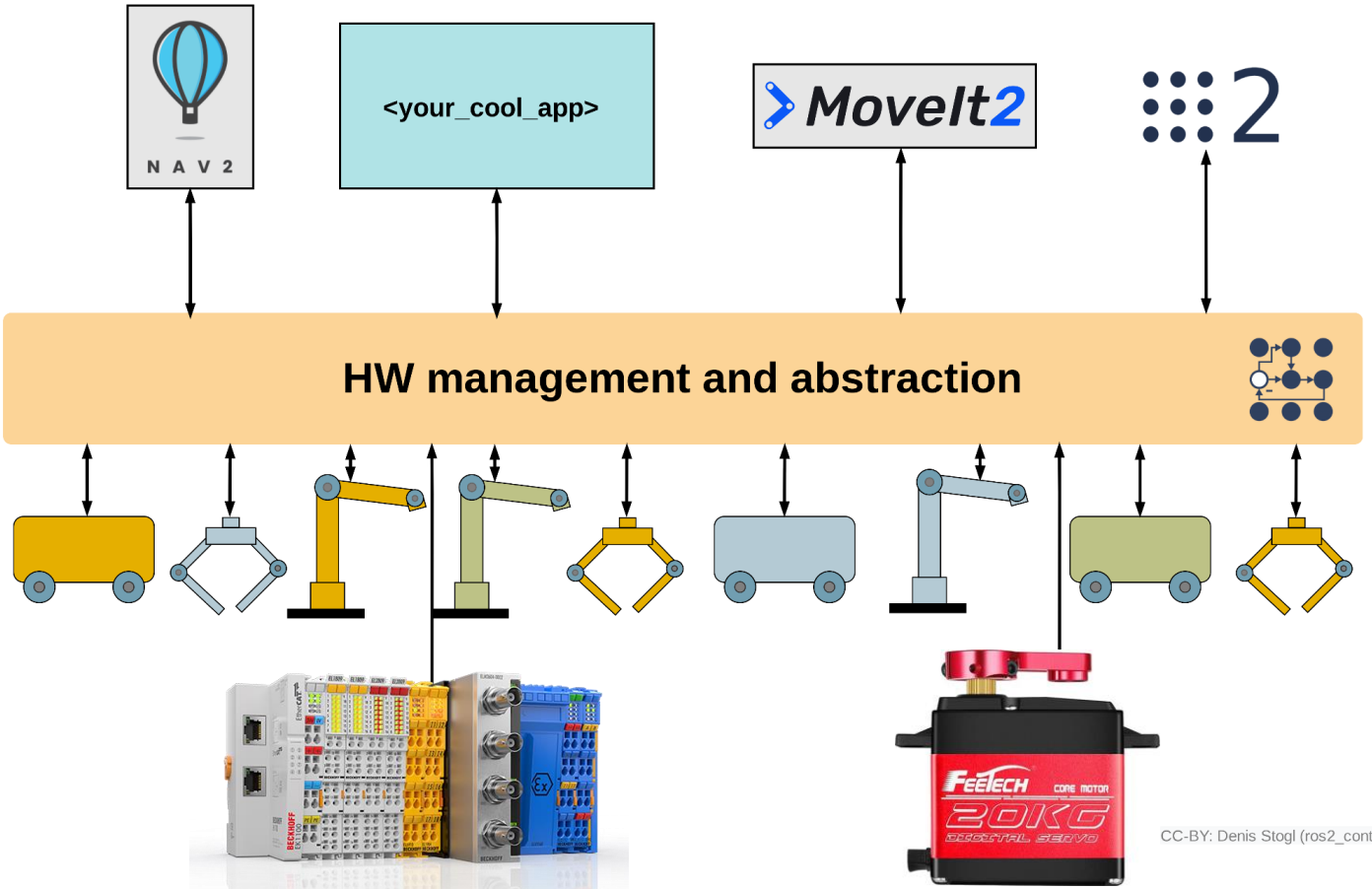
Why am I here?

» Reuse, reuse, reuse

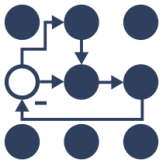
$whoami

- b»robotized group
  - Consulting and Development Services (sw+hw) for Robotics applications with ROS 2
- Nikola Banović
  - Robotics Engineer at **b»robotized** group (Croatia)
  - M.Sc. Robotics and Control Engineering from Univ. Zagreb
  - ros2_control contributor, author multiple driver packages

Why ros2_control?



CC-BY: Denis Stogl (ros2_control)



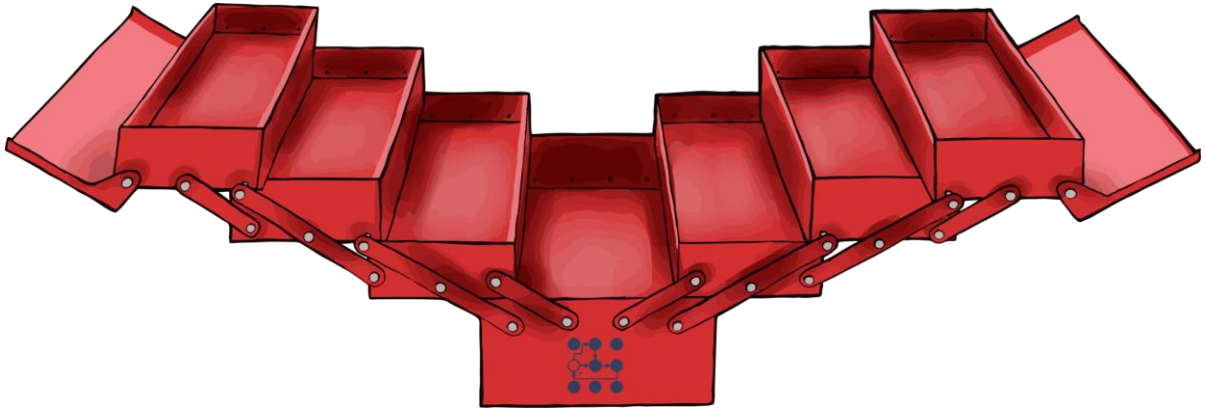
b»robotized

ros2_control

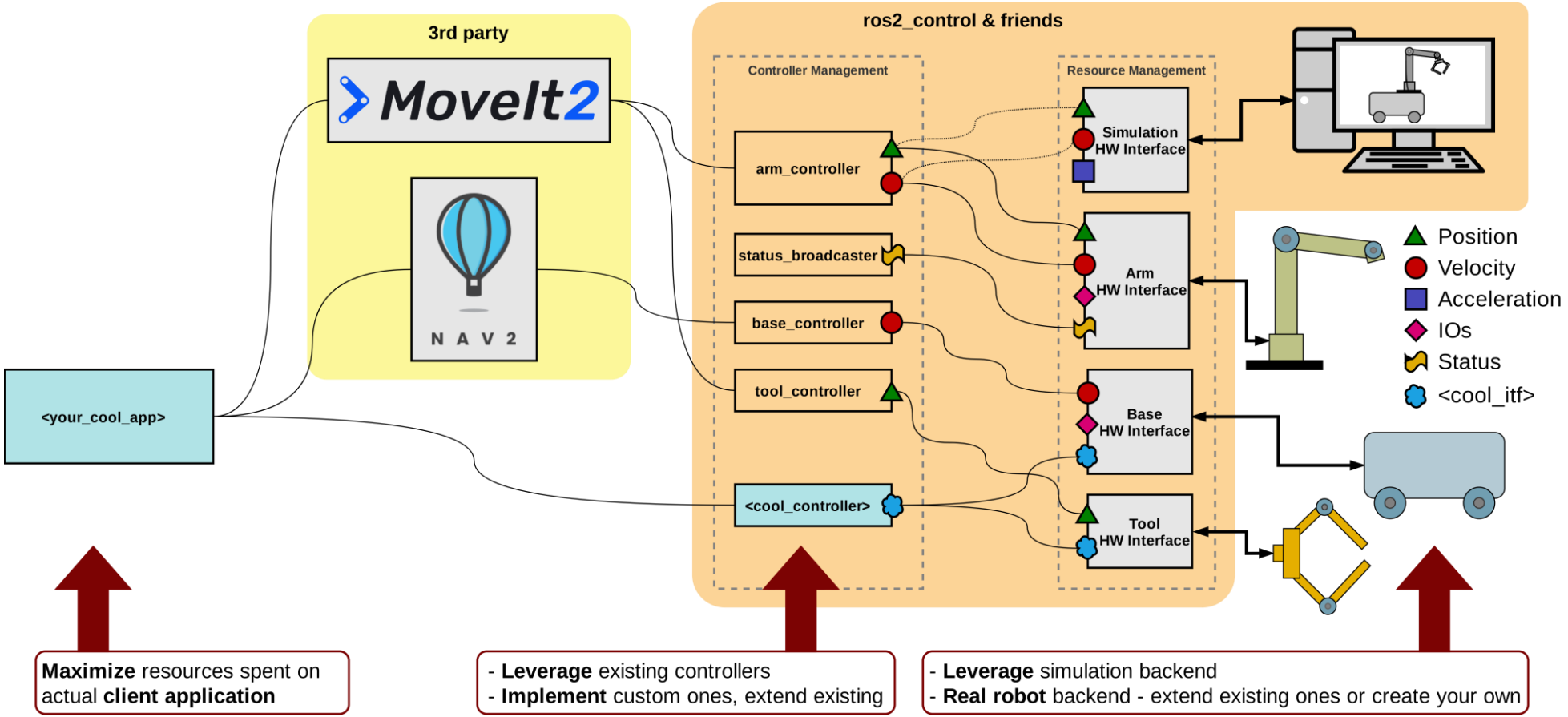
- Multiple, different robots, sensors, actuators... And they might change!
- I'd love to use an existing control algorithm
- I want to change something without a long downtime
- I don't want to be locked to a vendor with my solution

no ros2_control

- I have my controller within PLC
- "I know control better than X" and have my industrial-grade controller already written
- "I have just one robot, why bother with this complexity?"



Why ros2_control?




CANopen
Modbus
BECKHOFF
TwinCAT ADS
ctrlX
AUTOMATION
EtherCAT
ros2_control

Anatomy of a hardware interface

CONNECT + MAP

1. make the hardware interface a participant of the network

2. map the data from the protocol to you system state/command

b»robotized

“You can represent any data with ros2_control as long as it is double!”

Dr. Denis (2023)

“You can repres any d with ros2_control a is double.

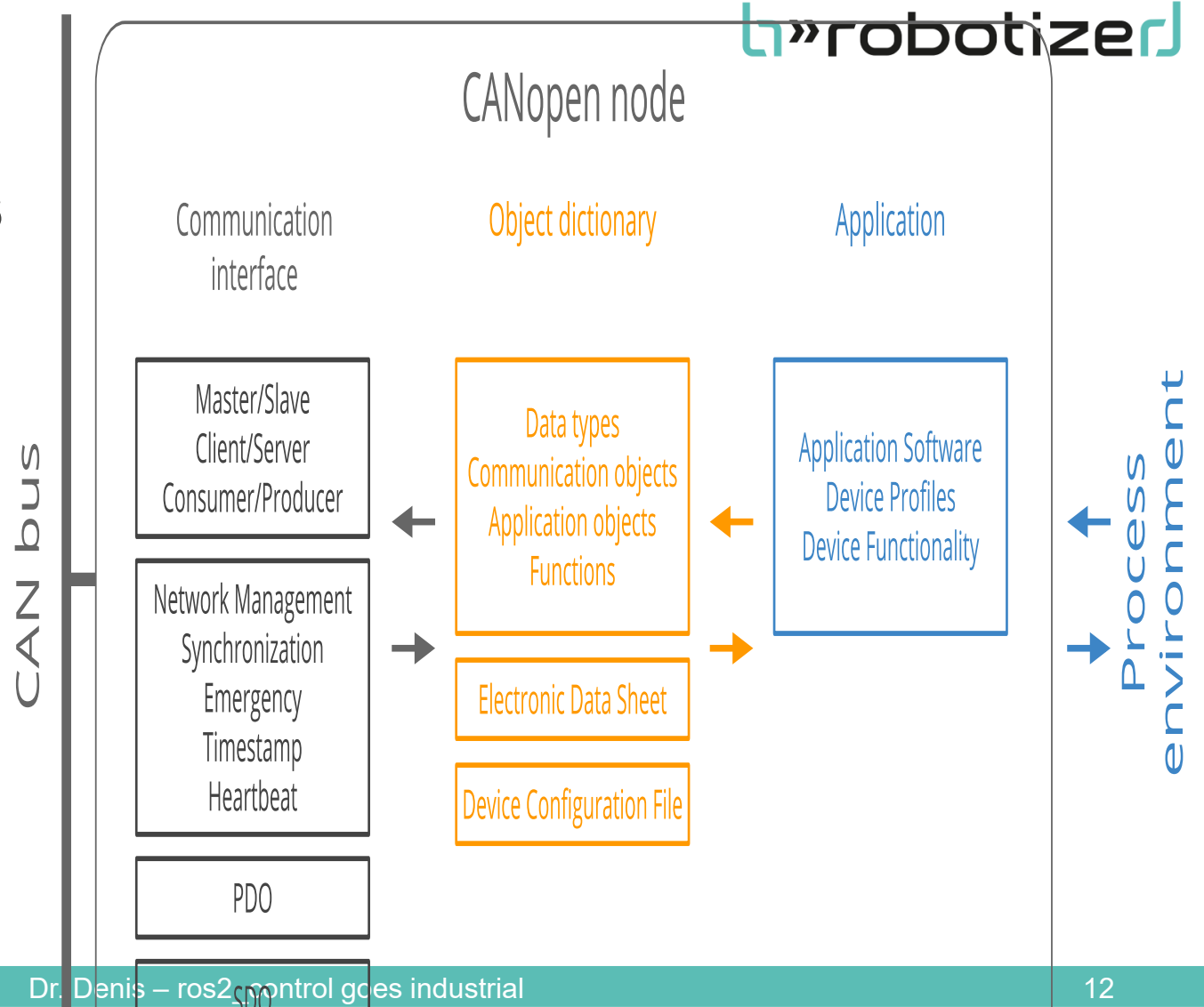
Dr. Denis (2023)

CANOPEN

Protocol on the CAN BUS

» **Data model:**

Object Dictionary - index/subindex for each data entry



in ROS: ros2_canopen

- CANOPEN master + slave device representation
- Use CiA 301 or CiA 402 (motion profile)

- If not there is a „CustomMappingCanopenSytem“

- https://github.com/ros-industrial/ros2_canopen

```
<xacro:macro name="DiffDrive_ros2_control" params="
            name bus_config master_config can_interface_name master_bin">

  <ros2_control name="${name}_CanOpen" type="system">
    <hardware>
      <plugin>canopen_ros2_control/CustomMappingCanopenSystem</plugin>
      <param name="bus_config">${bus_config}</param>
      <param name="master_config">${master_config}</param>
      <param name="can_interface_name">${can_interface_name}</param>
      <param name="master_bin">"${master_bin}"</param>
      <param name="enable_canopen_interfaces">true</param>
    </hardware>
    <joint name="${prefix}left_wheel">
      <param name="node_id">0x17</param>  <!--23-->
      <xacro:my_motor_mapping inverse_direction="true"/>
    </joint>
    <joint name="${prefix}right_wheel">
      <param name="node_id">0x1C</param>  <!--28-->
      <xacro:my_motor_mapping inverse_direction="false"/>
    </joint>
  </ros2_control>

</xacro:macro>

<xacro:macro name="my_motor_mapping" params="inverse_direction:=false">
  <!-- RPDO 1 -->
  <!-- Control Word -->
  <command_interface name="drive_enable" data_type="bool">
    <param name="data_type">bool</param>
    <param name="index">0x2100</param>
    <param name="subindex">0x0</param>
  </command_interface>
  <command_interface name="velocity" data_type="int16_t">
    <param name="data_type">int16_t</param>
    <param name="index">0x2110</param>
    <param name="subindex">0x0</param>
    <param name="unit">rpm</param>
    <param name="to_hw_scaling_factor">${inverse_factor * 1.0 * gear_ratio}</param>
  </command_interface>
  <state_interface name="fault" data_type="bool">
    <param name="data_type">bool</param>
    <param name="index">0x2115</param>
    <param name="subindex">0x0</param>
  </state_interface>
</xacro:macro>
```

Beckhoff ADS

Automation device specification

Similar to TCP/IP

» **Data model:**

PLC variables, accessed by symbolic name

in ROS: beckhoff_ads_driver

https://github.com/b-robotized/beckhoff_ads_driver

```
<xacro:macro name="state_interface_PLC" params="name PLC_symbol PLC_type n_elements:=1 index:=''">
  <state_interface name="${name}">
    <param name="PLC_symbol">${PLC_symbol}</param>
    <param name="PLC_type">${PLC_type}</param>
    <param name="n_elements">${n_elements}</param>
    <xacro:if value="${index != ''}">
      <param name="index">${index}</param>
    </xacro:if>
  </state_interface>
</xacro:macro>

<xacro:macro name="beckhoff_bot_ros2_control" params="name">
  <ros2_control name="${name}" type="system">
    <hardware>
      <plugin>beckhoff_hardware_interface/BeckhoffSystem</plugin>
      <param name="plc_ip_address">192.168.122.218</param>
      <param name="plc_ams_net_id">192.168.122.218.1.1</param>
      <param name="plc_ams_port">851</param>
      <param name="local_ams_net_id">192.168.122.1.1.1</param>
    </hardware>
    <gpio name="robot_out">
      <xacro:command_interface_PLC name="enable" PLC_symbol="MAIN.Enable" PLC_type="BOOL" initial_value="1"/>
      <xacro:command_interface_PLC name="GoToID" PLC_symbol="MAIN.goToID" PLC_type="UDINT" initial_value="5"/>
      <xacro:command_interface_PLC name="pos0" PLC_symbol="MAIN.bPos0" PLC_type="BOOL" n_elements="2" index="0" initial_value="1"/>
      <xacro:command_interface_PLC name="pos1" PLC_symbol="MAIN.bPos1" PLC_type="BOOL" n_elements="2" index="1" initial_value="1"/>
    </gpio>
    <sensor name="robot_sensor">
      <xacro:state_interface_PLC name="status" PLC_symbol="MAIN.bStatusError" PLC_type="BOOL"/>
      <xacro:state_interface_PLC name="batt_voltage" PLC_symbol="MAIN.BatteryVoltage" PLC_type="REAL"/>
      <xacro:state_interface_PLC name="curr_pos_0" PLC_symbol="MAIN.currentPos" PLC_type="LREAL" n_elements="2" index="0" />
      <xacro:state_interface_PLC name="curr_pos_1" PLC_symbol="MAIN.currentPos" PLC_type="LREAL" n_elements="2" index="1" />
    </sensor>
  </ros2_control>
</xacro:macro>
```

Modbus (client)

Grandad protocol

» **Data model:**

16-bit registers + addresses that point to them

in ROS: modbus_hardware_interface

https://github.com/b-robotized/modbus_hardware_interface

https://www.youtube.com/watch?v=iEiZje8imeU

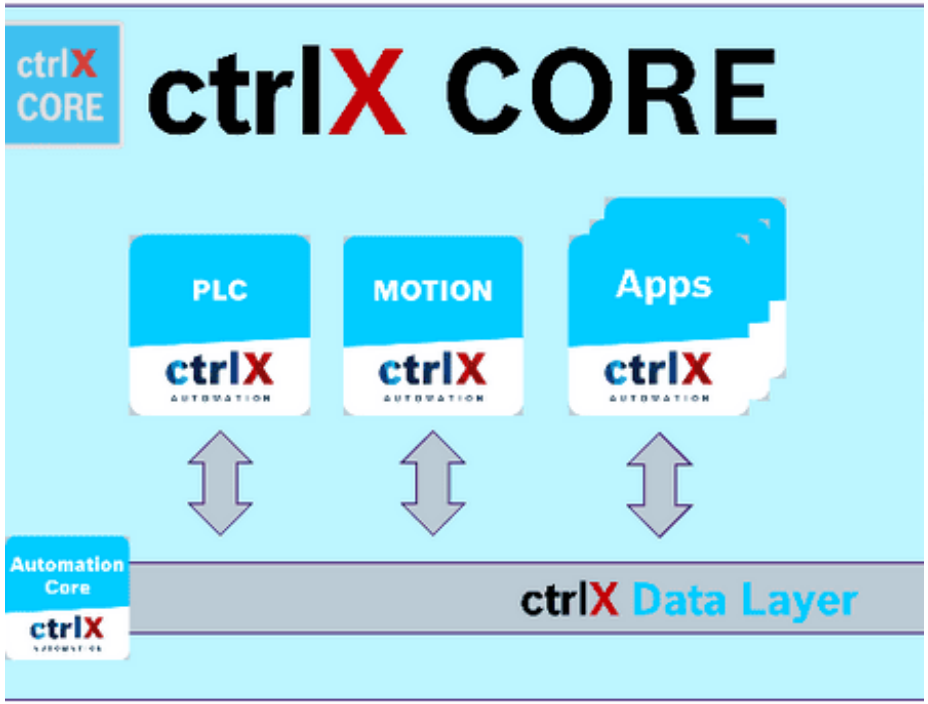
```
<xacro:macro name="my_actuator_ros2_control" params="name">
  <ros2_control name="${name}" type="system">
    <hardware>
      <plugin>modbus_hardware_interface/ModbusHardwareInterface</plugin>
      <param name="modbus_server_ip">10.28.28.28</param>
      <param name="modbus_server_port">502</param>
      <param name="use_persistent_connection">tue</param>
    </hardware>
    <joint name="${prefix}revolute_joint">
      <command_interface name="move_left">
        <param name="register">0x0802</param>
        <param name="bits_to_read">8</param>
        <param name="conversion_fn">to_int_to_modbus</param>
        <param name="write_function">bits</param>
      </command_interface>
      <command_interface name="move_right">
        <param name="register">0x0804</param>
        <param name="bits_to_read">8</param>
        <param name="conversion_fn">to_int_to_modbus</param>
        <param name="write_function">bits</param>
      </command_interface>
      <state_interface name="position">
        <param name="register">0x0017</param>
        <param name="bits_to_read">16</param>
        <param name="conversion_fn">to_int_to_float</param>
        <param name="read_function">input_register</param>
      </state_interface>
      <state_interface name="min_pose">
          <param name="register">0x0026</param>
        <param name="bits_to_read">8</param>
        <param name="conversion_fn">to_int_to_float</param>
        <param name="read_function">input_bits</param>
      </state_interface>
      <state_interface name="max_pose">
        <param name="register">0x0024</param>
        <param name="bits_to_read">8</param>
        <param name="conversion_fn">to_int_to_float</param>
        <param name="read_function">input_bits</param>
      </state_interface>
    </joint>
  </ros2_control>
</xacro:macro>
```

CtrlX Data Layer

Data sharing in CtrlX OS platform

» **Data model:**

Variables and metadata addressed via path + name



in ROS: rexroth_datalayer_driver

```xml
<xacro:macro name="state_interface_DataLayer" params="name DL_address DL_variable initial_value:=''">
  <state_interface name="${name}">
    <param name="DL_address">${DL_address}</param>
    <param name="DL_variable">${DL_variable}</param>
    <xacro:if value="${initial_value != ''}">
      <param name="initial_value">${initial_value}</param>
    </xacro:if>
  </state_interface>
</xacro:macro>
<xacro:macro name="command_interface_DataLayer" params="name DL_address DL_variable initial_value:=''">
  <command_interface name="${name}">
    <param name="DL_address">${DL_address}</param>
    <param name="DL_variable">${DL_variable}</param>
    <xacro:if value="${initial_value != ''}">
      <param name="initial_value">${initial_value}</param>
    </xacro:if>
  </command_interface>
</xacro:macro>

<xacro:macro name="gripper_ros2_control" params="name">

  <ros2_control name="${name}" type="system">
    <hardware>
      <plugin>ctrlx_hardware_interface/DataLayerHardwareInterface_NRT</plugin>
      <param name="device_ip">192.168.1.1</param>
      <param name="device_user">boschrexroth</param>
      <param name="device_password">boschrexroth</param>
    </hardware>
    <gpio name="in1">
      <xacro:state_interface_DataLayer name="gripper_opened"
        DL_address="fieldbuses/ethercat/master/instances/Gripper/realtime_data/input"
        DL_variable="EL1008/Channel_1"
        initial_value="0.0"/>
      <xacro:state_interface_DataLayer name="gripper_closed"
        DL_address="fieldbuses/ethercat/master/instances/Gripper/realtime_data/input"
        DL_variable="EL1008/Channel_2"
        initial_value="0.0"/>
    </gpio>
    <gpio name="out1">
      <xacro:command_interface_DataLayer name="open_gripper"
        DL_address="fieldbuses/ethercat/master/instances/Gripper/realtime_data/output"
        DL_variable="EL2008/Channel_1"/>
      <xacro:command_interface_DataLayer name="close_gripper"
        DL_address="fieldbuses/ethercat/master/instances/Gripper/realtime_data/output"
        DL_variable="EL2008/Channel_2"/>
    </gpio>
  </ros2_control>
</xacro:macro>
```

https://github.com/b-robotized/rexroth_datalayer_driver

in ROS: rexroth_datalayer_driver

```xml
<xacro:macro name="state_interface_DataLayer" params="name DL_address DL_variable initial_value:=''">
  <state_interface name="${name}">
    <param name="DL_address">${DL_address}</param>
    <param name="DL_variable">${DL_variable}</param>
    <xacro:if value="${initial_value != ''}">
      <param name="initial_value">${initial_value}</param>
    </xacro:if>
  </state_interface>
</xacro:macro>
<xacro:macro name="command_interface_DataLayer" params="name DL_address DL_variable initial_value:=''">
  <command_interface name="${name}">
    <param name="DL_address">${DL_address}</param>
    <param name="DL_variable">${DL_variable}</param>
    <xacro:if value="${initial_value != ''}">
      <param name="initial_value">${initial_value}</param>
    </xacro:if>
  </command_interface>
</xacro:macro>

<xacro:macro name="gripper_ros2_control" params="name">

  <ros2_control name="${name}" type="system">
    <hardware>
      <plugin>ctrlx_hardware_interface/DataLayerHardwareInterface_NRT</plugin>
      <param name="device_ip">192.168.1.1</param>
      <param name="device_user">boschrexroth</param>
      <param name="device_password">boschrexroth</param>
    </hardware>
    <gpio name="in1">
      <xacro:state_interface_DataLayer name="gripper_opened"
        DL_address="fieldbuses/ethercat/master/instances/Gripper/realtime_data/input"
        DL_variable="EL1008/Channel_1"
        initial_value="0.0"/>
      <xacro:state_interface_DataLayer name="gripper_closed"
        DL_address="fieldbuses/ethercat/master/instances/Gripper/realtime_data/input"
        DL_variable="EL1008/Channel_2"
        initial_value="0.0"/>
    </gpio>
    <gpio name="out1">
      <xacro:command_interface_DataLayer name="open_gripper"
        DL_address="fieldbuses/ethercat/master/instances/Gripper/realtime_data/output"
        DL_variable="EL2008/Channel_1"/>
      <xacro:command_interface_DataLayer name="close_gripper"
        DL_address="fieldbuses/ethercat/master/instances/Gripper/realtime_data/output"
        DL_variable="EL2008/Channel_2"/>
    </gpio>
  </ros2_control>
</xacro:macro>
```

https://github.com/b-robotized/rexroth_datalayer_driver

Not REAL TIME!

EtherCAT

High-performance Ethernet protocol

» **Data model:**

Process Data Image – a continuous block of memory maintained by the master node

in ROS2: ethercat_driver_ros2

» *(We @ b-robotized use DataLayer for this)*

» **Cudos to ICUBE Laboratory!**

(we didn‘t contribute!)

» https://github.com/ICube-Robotics/ethercat_driver_ros2

What is next?

Use, use, use...

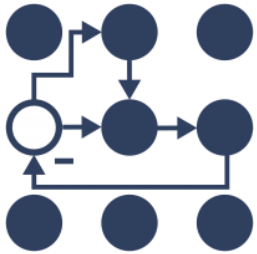
Help us polish existing and add new protocols to ros2_control!

Use, use, use...

» Standardize scaling in a header-only library?

» Standardize interface descriptions

» Use ros2_control variant types -> not only double!



b»robotized

ros2_control goes industrial

CANopen

BECKHOFF TwinCAT ADS

ctrlX AUTOMATION

EtherCAT

Modbus

ROSCon DE&FR 2025 Strasbourg

Nikola Banović