Middleware interconnecting ROS/ROS2 with EtherCAT protocol

Legged Robots Team / Control Systems Lab / School of Mechanical Engineering
National Technical University of Athens

Aristotelis Papatheodorou
Research Associate

Athanasios Mastrogeorgiou
PhD Student

Konstantinos Machairas
PhD Student

Konstantinos Koutsoukis
PhD Student

Evangelos Papadopoulos
Professor
Table of Contents

• What is EtherCAT?
• Why EtherCAT?
• EtherCAT Functional Principle
• Typical EtherCAT Slave
• Intro to EtherLab
• Why EtherLab?
• ether_ros2 – ROS2 wrapper for Etherlab
  • Overview
  • A simple example (led blinking)
  • Adding a new slave type
  • YAML network config
• Case study: Laelaps II quadruped by CSL Legged Robots Team
• Next steps
What is EtherCAT?

**EtherCAT Technology**
EtherCAT is a high-performance, low-cost easy to use Industrial Ethernet technology with a flexible topology.

**Key features:**
- Full Duplex Communications.
- Master-Slave Architecture.
- Fastest industrial Ethernet technology.
- Suitable for both centralized & decentralized system architectures.
- Wide protocol support (CAN / SERCOS over EtherCAT)
- Proven technology, widely used in many fields (Robotics Power Plants, Wind Turbines, Medical devices etc.).
Why EtherCAT?
Typical EtherCAT Slave

Application Layer (μC)

Physical Device Interface

Data Link Layer (ESC)
Intro to EtherLab

EtherLab is an Open Source Toolkit for rapid realtime application development under Linux

- Utilizes the **native** Ethernet driver
- Supports **any realtime environment** through independent architecture
- Has **seamless integration** in any GNU/Linux distribution
- Provides virtual read-only **network interface** for debugging and traffic monitoring purposes
- Provides **userspace command-line tool** 'ethercat'
- **Conforms** to IEC/PAS 62407
Similar ROS packages for EtherCAT Master

- Deprecated ROS Industrial SOEM
- SOEM package
- ANYbotics forked SOEM package

Why EtherLab?

- Designed for hard realtime applications
- Complete integrability of EtherCAT protocol
- Kernel Space runtime, with user-space API vs. user-space runtime of SOEM (and its latency drawbacks)
- Advanced Diagnostic features
Overview of ether_ros2 (1/2)

Highlights

• Composable, Lifecycle Node
• Optimized intra-process communications for minimum latency
• Spinlock protected inter-thread data transactions
• Generic design that can support any application
• Easy to use, modifiable package
Overview of ether_ros2 (2/2)

**Topics**
- /ether_ros2/transition_event
- /pdo_in
- /pdo_listener
- /pdo_out
- /pdo_raw

**Services**
- /ether_ros2/change_state
- /ether_ros2/describe_parameters
- /ether_ros2/get_available_states
- /ether_ros2/get_available_transitions
- /ether_ros2/get_parameter_types
- /ether_ros2/get_parameters
- /ether_ros2/get_state
- /ether_ros2/get_transition_graph
- /ether_ros2/list_parameters
- /ether_ros2/set_parameters

**Messages**
- ModifyPdoVariables.msg
- PdoInX.msg
- PdoOut.msg
- PdoRaw.msg
Simple Example of ether_ros2

- ether_ros2 & ether_ros2_manager run inside the same container (high priority)
- joy_node runs as another process with lower priority
Adding a New Slave Type in ether_ros2

Create & Populate Input PDOs' message file
Advantages

- Runtime configuration of the EtherCAT network
- User-Friendly slave configuration method
- No source code modifications
Laelaps II Quadruped
by legged robots’ team @ CSL [nereus.mech.ntua.gr/legged]

Decentralized Architecture

- Increased Modularity
- Low Cost
- Higher overall processing power (in parallel)
- Expandability / Flexibility
- Reduced cable count
Laelaps II Architecture
Full Walking Experiment (1/2)

Experiment’s Details

- Gamepad controlled motion
- Accurate synchronization and steady gait cycles
- Intensive, high throughput operation
- Maximum achieved velocity during trotting: 0.1 [m/s]
Full Walking Experiment (2/2)

Trajectory Planning

- Toes' trajectories controlled & synchronized by the ether_ros2

Reference trajectory

Actual trajectory
Next Steps

- **Integrate Quality of Service (QoS)** policies to optimize communication among the nodes

- **Profile & Measure the latencies** of the package to identify and eliminate any potential bottlenecks

- **Model the package’s behavior** to be used in the control design process

- Create **tutorials** for easy integration of the package to an arbitrary application

- **Make it available to the community** as a complete & all-in-one EtherCAT Master solution
Members of the legged team involved in the project

Aristotelis Papatheodorou
Research Associate
aristotpap@gmail.com

Athanasios Mastrogeorgiou
PhD Student
mastrogeorgiou.gr
arnast@central.ntua.gr

Konstantinos Machairas
PhD Student
www.kmachairas.com
kmach@central.ntua.gr

Konstantinos Koutsoukis
PhD Student
kkoutsou@central.ntua.gr
THANK YOU

check our work @
csl-ep.mech.ntua.gr/legged

This work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant” (Project Number: 2182).