A Linux-Based Multi-Interface Real-Time Controller for Rapid Prototyping of Robotic Systems

Junyoung Kim* and Hyun-Joon Chung

Korea Institute of Robotics & Technology Convergence, South Korea

(Correspondence: Hyun-Joon Chung, hjchung@kiro.re.kr)

KOREA INSTITUTE OF ROBOTICS & TECHNOLOGY CONVERGENCE

Abstract

This study focuses on developing a **Linux-based real-time (RT) controller capable of operating with multiple communication interfaces**. A key contribution of this research is the simultaneous use of various interfaces commonly applied in the control of robotic and mechatronic systems, **facilitating rapid prototyping by decreasing the interface dependency of the system components**. To meet the demands of various system development, the controller integrates major communication protocols, such as EtherCAT, CAN, and DAQ devices widely used in robotics and mechatronics.

In addition to supporting these interfaces, **the controller offers a straightforward method for managing both RT and non-RT threads**. This is intended to enable **the implementation of a synchronized, multi-interface controller while supporting the integration of higher-level tasks**, such as a GUI, computer vision, AI-based services, and communication with external control networks. This flexibility allows the controller to handle a range of advanced functionalities without compromising real-time performance.

Experimental validation confirms that the developed controller maintains accurate sampling periods for all RT threads, even when non-RT tasks such as the GUI are running. **Low latency measurements further verify that the system preserves real-time performance, ensuring stable and reliable operation in multi-interface applications**.

Problem Statement

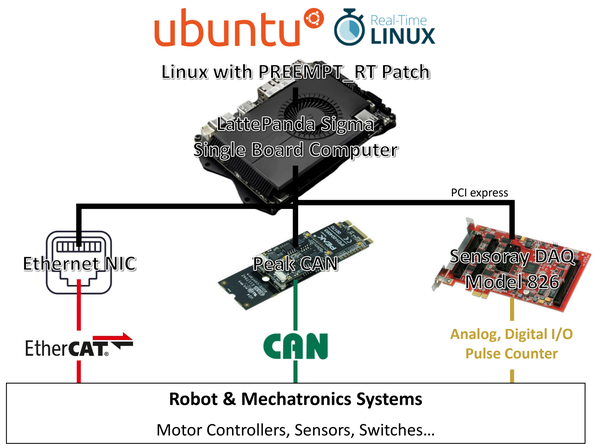
RT controller with multi-interfaces for rapid prototyping of the robotic system

- **Necessity:**
  - To minimize trade-offs in decision-making process when selecting components
  - To test various components and interfaces to find the optimal environment
- **Requirements:**
  - Includes major communication interfaces for robot systems: EtherCAT, CAN, DAQ
  - Allocates each interface on individual RT threads to simplify the integration procedure among the interfaces

Methods

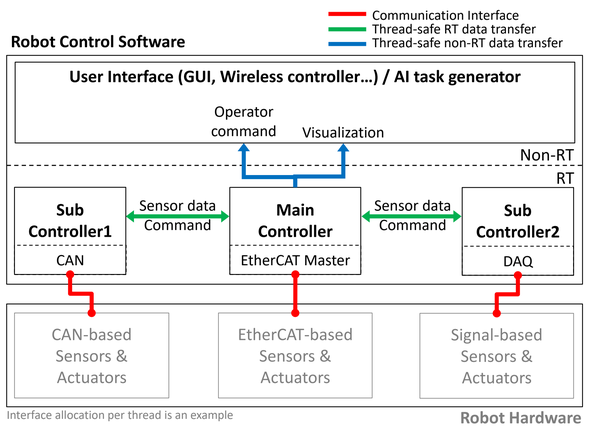
Environment & Controller Configuration

- SBC: Small size & low power consumption
  ➔ Compatible for robot (mobile)
- Preempt_RT: Low hardware dependency, Standard debugging and programming environment
  ➔ Suitable for testing & prototyping



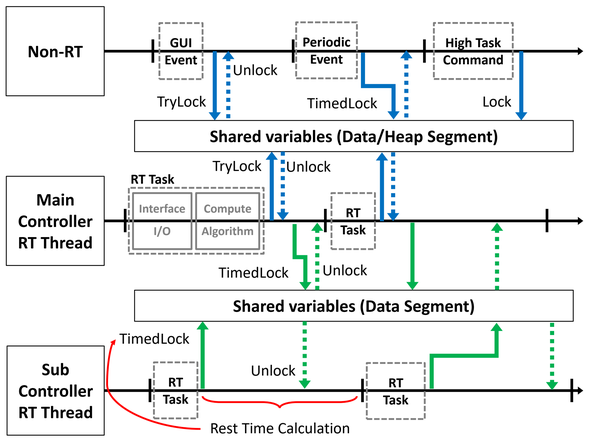
Controller S/W Architecture

- RT threads for each interface
  ➔ Simple to manage and expand by adding new interface
- Non-RT thread for utilizing GUI or high level tasks
- Mutex for thread-safe synchronization of data obtained from each interface



Threads Synchronization by Mutex

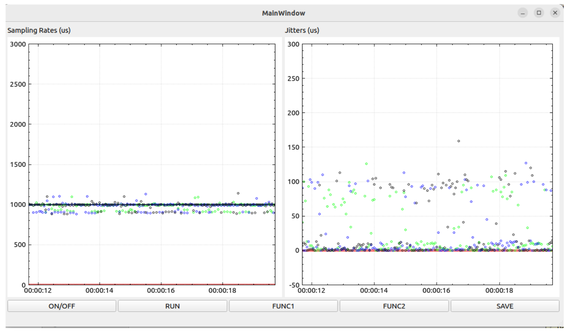
- RT controller proper sampling & locking mechanism
- **TimedLock**: Try to lock mutex, waiting for up to the rest of the current period (best effort to sync, not to corrupt RT)
- **TryLock**: Minimize load on the mutex (not necessary task)
- **Lock**: Wait until the command is delivered (Necessary task)



Results

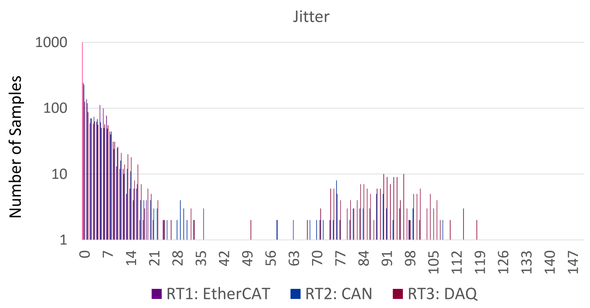
RT Performance Test with Jitter Analysis

- Non-RT GUI: Graph, Interaction button, and Real-time jitter logger
- With/Without load Injection on system with *stress* tool
- Sampling rate of 1ms (1kHz frequency) for all RT threads



Jitter analysis

- Affordable maximum jitter of RT threads
  : 128(EtherCAT), 116(CAN), 139(DAQ) *μs*
- Deterministic and timeliness RT performance with non-RT and other system loads



Conclusion

This study developed a **Linux-based real-time controller that supports multiple communication interfaces**, offering flexibility and efficiency in robotic and mechatronic applications.

The integration of major communication protocols, including EtherCAT, CAN, and DAQ, **may enable effective in reducing system component dependency, facilitating rapid prototyping across diverse systems**. Furthermore, the controller's design allows for **the synchronized management of both RT and non-RT threads, enabling the potential integration of advanced functionalities** such as GUIs, AI-based services, and external network communication in future implementations without compromising real-time performance.

Experimental validation demonstrated that **the controller maintains consistent sampling periods for RT threads and exhibits low latency, even with non-RT tasks in operation**. These results indicate that **the proposed controller can reliably operate in complex multi-interface environments while ensuring stability and precision.** Future work may explore the practical implementation of these advanced functionalities and further optimization of RT and non-RT task management to enhance performance and usability.

Acknowledgments

This work was supported by the Challengeable Future Defense Technology Research and Development Program through the Agency For Defense Development(ADD) funded by the Defense Acquisition Program Administration(DAPA) in 2024(No. 915052101).

Real Time Communications Conference and Expo at Illinois Tech
IEEE International Conference
IEEE RTC 2024 - October 8-10, 2024 - IIT, Chicago, IL, USA

IEEE