ROBOCON-RT – Kickoff Document

Contents

1 DESCRIPTION

The end product for this project will provide a framework that facilitates the design and construction of remote controllable mobile robot platforms that are based on multiple motors and their real-time coordinated control. The originality and innovation for this project will come from its modularity, execution efficiency and its reliance on more modern control protocols such as EtherCAT.

Currently available examples and tools usually require custom development of a large number of components to deploy a robot platform and are hence quite prohibitive due to the large initial investment required.
The primary differences of this project from existing examples will hence be its performance, modularity and the provision of existing components to facilitate robot design and deployment. Expected users of this product will be the programmers who are interested in robotics.

2 MASTER FEATURE LIST

- **MF1 -** A low-level, real-time motor control software interface.
- **MF2 -** A 2-wheeled robot base platform that can perform basic mobility tasks using this real-time motor control software interface.
- **MF3 -** Customized real-time Yocto Linux environment for EtherCAT communication.
- **MF4 -** Fine-grained, customizable control over the framework.
- **MF5 -** Robust, real-time EtherCAT-based communications over motor drives.
- **MF6 -** Easy to use modular interface for developers.
- **MF7 -** High-performance data exchange inside the system.
- **MF8 -** Integration with the ROS framework.
- **MF9 -** Easily integrated with additional sensors and actuators.
- **MF10 -** Secure communication between the single board computer and the remote OCU.
- **MF11 -** An Open Source license.
- **MF12 -** Support for more generic hardware using the CANopen interface

3 WORKPACKAGES

WP #	Term	WP Title	Person-Months
1	491	Construction of a 2-wheeled mobile robot base	4
2	491	Choice, acquisition and integration of an EtherCAT-capable embedded CPU and the associated real-time Yocto Linux infrastructure	4
3	491	Porting and modernization of the RHexLib software	4
4	491	Design and implementation of an abstract EtherCAT motor control interface	4
5	492	Design and implementation of a robust network communication software subsystem for robot controlled systems	4
6	492	Overall system integration and scenario tests	6
7	492	Integration with ROS[1]	4
8 ⋆	492	Integration of additional sensors and capabilities	3
		Total:	33

Table 3.1: Workpackages

4 DETAILED DESCRIPTIONS OF HIGH-LEVEL WORKPACKAGES

4.1 WP1 - CONSTRUCTION OF 2-WHEELED MOBILE ROBOT BASE

In this workpackage, the following functionalities / features / work items will be implemented:

1. Choice of appropriate motors, peripherals and battery.
2. Mechanical design of the topology of the robot base.
3. Acquisition of other necessary materials of the robot base.
4. Integration of the mobile base with the board chosen in 4.2.

⋆: Bonus

4.2 WP2 - Choice, acquisition and integration of an EtherCAT-capable embedded CPU and the associated real-time Yocto Linux infrastructure

In this workpackage, the following functionalities / features / work items will be implemented:

1. Choice and acquisition of a single board computer with high performance, multi-core CPU supporting EtherCAT[2] technology.
2. Setup of Yocto Linux[3] and real-time patch of Yocto Linux on the selected single board computer.
3. Choice and customization of layers of the embedded Linux environment.

4.3 WP3 - Porting and modernization of the RHexLib software

In this workpackage, the following functionalities / features / work items will be implemented:

1. Understanding the core functionalities of RHexLib.
2. Porting the RHexLib software to the board chosen in WP2.
3. Modernization of the software according to the modern C++ standard.

4.4 WP4 - Design and implementation of an abstract EtherCAT motor control interface

In this workpackage, the following functionalities / features / work items will be implemented:

1. Understanding the basics of the EtherCAT protocol.
2. Implementing the abstract motor control interface.
3. Testing the interface on an example motor device.
4. Establishing the communication between the master and slave modules using EtherCAT.

4.5 WP5 - Design and implementation of a robust network communication software subsystem for robot controlled systems

In this workpackage, the following functionalities / features / work items will be implemented:

1. Analyzing the requirements of the network communication subsystem.
2. Design of the network communication topology.
3. Implementation of the network subsystem.

[2] https://www.ethercat.org
[3] https://www.yoctoproject.org

4.6 WP6 - Overall system integration and scenario tests

In this workpackage, the following functionalities / features / work items will be implemented:

1. Design of scenarios to test the real-time performance of the platform.
2. Design of integration tests between the core RHexLib software and the EtherCAT module.
3. Benchmark of the network system.
4. Testing the basic functionalities of the 2-wheeled robot platform.

4.7 WP7 - Integration with ROS

In this workpackage, the following functionalities / features / work items will be implemented:

1. Understanding core ROS concepts.
2. Implementing an interface for communicating with ROS.
3. Integrating the system as a ROS node.

4.8 WP8 - Integration of additional sensors and capabilities

In this workpackage, the following functionalities / features / work items will be implemented:

1. Choice of the additional sensors to be supported.
2. Integration of the sensors with the system.

5 Overall Systems Architecture

Figure 5.1 includes a deployment diagram for the hardware of the project.

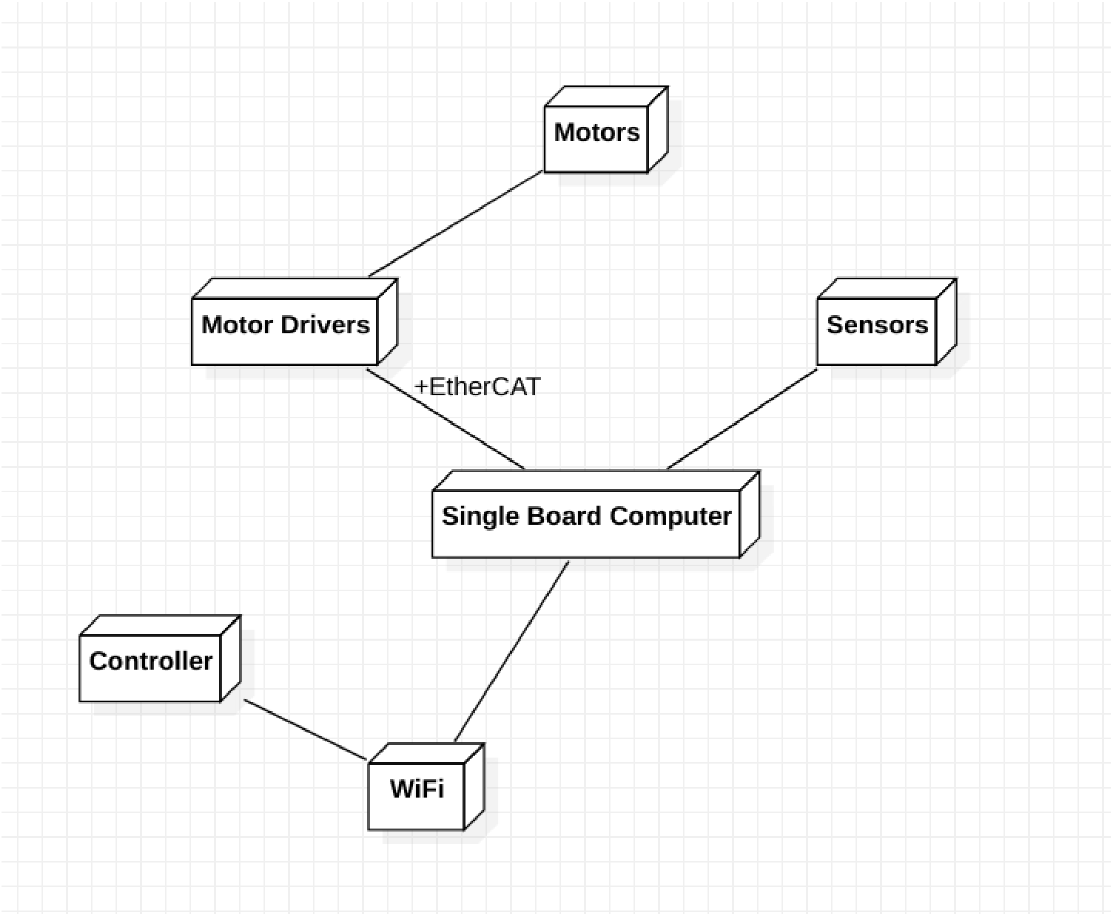

Figure 5.1: Project Deployment Diagram

The overall hardware topology consists of a single board computer, motor drivers, motors and additional sensors as key parts. This system will be controlled by a controller, which will be a laptop, through a Wi-Fi connection. The single board computer will communicate with motor drivers through the motor controller interface using the EtherCAT protocol and the motor drivers will drive the motors.

Figure 5.2 includes a component diagram for the software of the project.

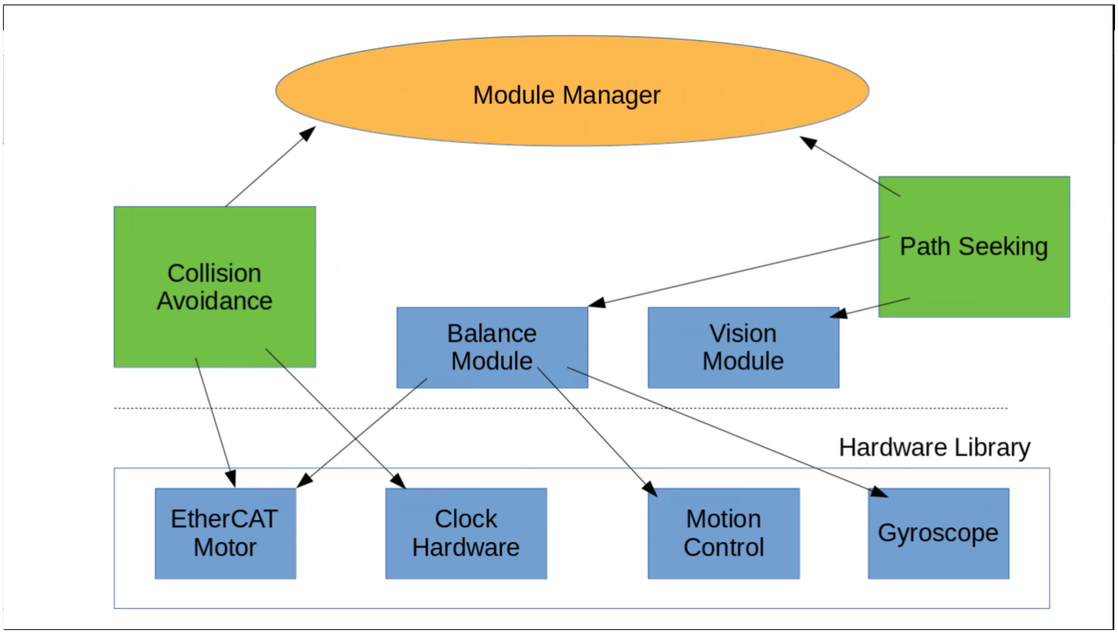

Figure 5.2: Project Component Diagram

On figure 5.2 the blue rectangles represent the generic modules that are provided by the framework itself. Green ones represent the User created modules that make use of the provided modules and the Module manager takes care of the linking and scheduling.

6 TIMELINE

Figure 6.1 includes a timeline of the project.

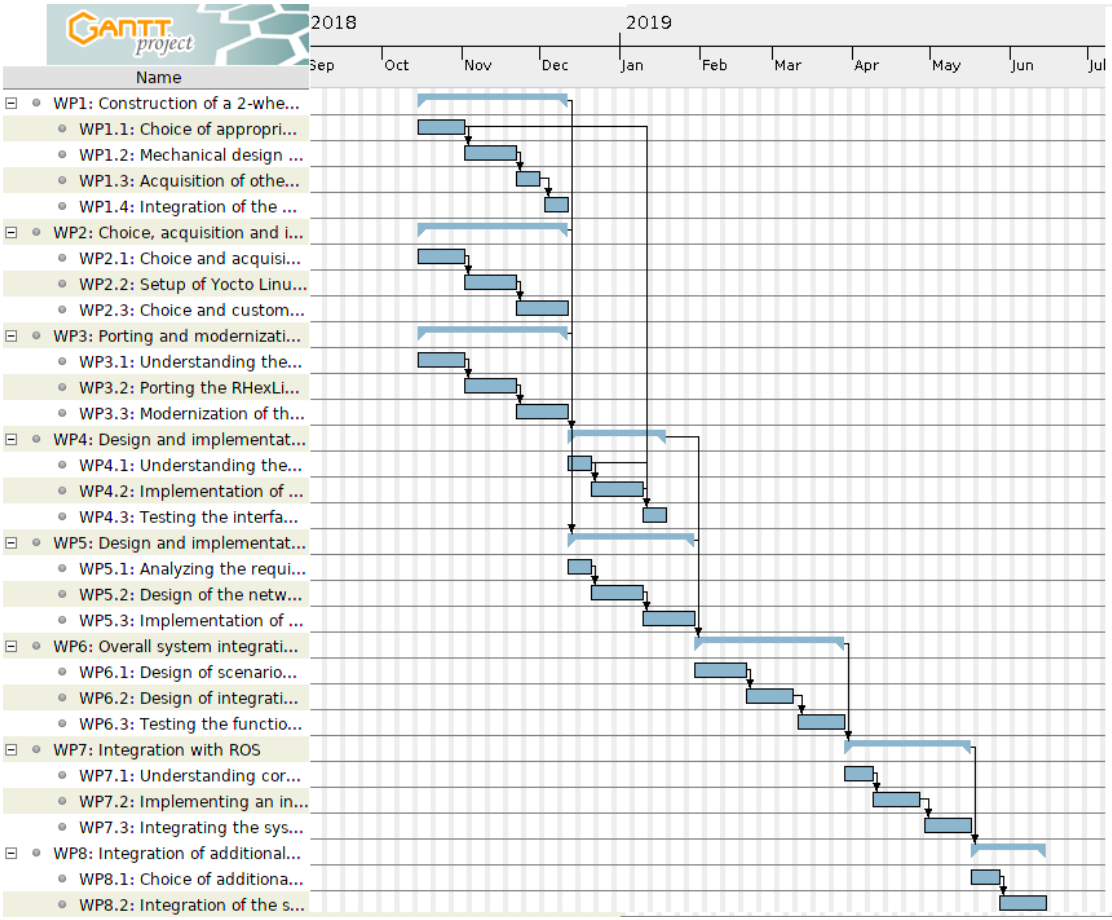

Figure 6.1: Project Timeline

7 Risk Assessment

Risk	Description	Possible solution(s)
Hardware failure due to misuse	Hardware components could be misused in several ways, such as exposure to high voltage and physical misuse.	Being careful in setting up hardware and checking configuration settings reduces the risk.
Problems in hardware acquisition	Some of the important hardware components of this project have to be imported abroad.	The project proposing company (Teknolus) provides the infrastructure for the rapid acquisition of components, as well as facilities for mechanical manufacturing when necessary.
Reliance on external code	The project relies on external code and resources which are susceptible to changes (such as in license)	The imported components have some viable alternatives that we can switch to (or write ourselves for used external code components).
Security issues	Communications between the system and the remote controller could be compromised by external parties.	The data transferred between the controller and system shall be encrypted symmetrically.

Table 7.1: Table of possible risks