**Real-time clock-based data logging subsystem in a 5 DOF robotic arm system**

Tál Dominik

Neptun code: RHNFAE

Lecturer: Dr. Magyar Attila

**Table of contents**

[Task description 2](#_Toc4967668)

[Introduction 3](#_Toc4967669)

[System and software requirements 4](#_Toc4967670)

[System plan 4](#_Toc4967671)

[System requirements 4](#_Toc4967672)

[Software requirements 4](#_Toc4967673)

[Implementation 5](#_Toc4967674)

[Hardware implementation 5](#_Toc4967675)

[File system and data structure 5](#_Toc4967676)

[Software implementation 5](#_Toc4967677)

[Summary 6](#_Toc4967678)

[Future plans 6](#_Toc4967679)

# Task description

Real-time clock-based data logging subsystem in a 5 DOF robotic arm system

The student shall develop a data logging subsystem based on an Arbotix-M robocontroller, a 5 degrees of freedom robotic arm, a micro SD card shield and a real time clock module.

The newly implemented subsystem should be able to log all the parameters of the robotic arm in real time, with at least a 100 ms resolution. The created log files’ format should be chosen with a focus on simple data analysis. The data structure should be designed by the student after negotiation with the lecturer. Detailed testing documentation should be included too in the semester closing documentation.

Required activities during the semester:

* choose the proper hardware for being able to determine the actual time and date
* choose the proper hardware for interfacing a micro SDHC card
* develop an interface to the Arbotix-M Robocontroller to be able to send out the internal values stored in its RAM
* design the data structure of the log files
* implement the secure data logging to the micro SDHC card
* execute and document the testing procedure

# Introduction

Nowadays in industrial systems data logs are one of the biggest treasure what mankind can gather to be able to develop processes and make them more effective.

TBC

# Data logging in real-time systems - 2 page (measurements, ping of the system)

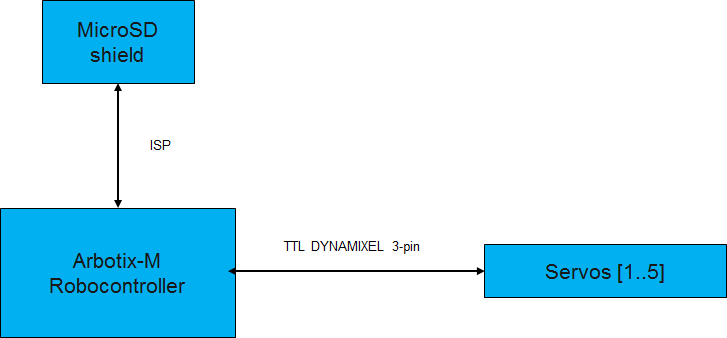
# TBC

# System and software requirements

In this chapter I will introduce the schematic of the system plan, the system level requirements and the software level requirements.

It is important to divide these two groups of requirements as later the two layers can be modified independently.

## System plan



## System requirements

## Software requirements

# Implementation

## Hardware implementation

TBC

## File system and data structure

The used interface can handle complex folder structures, but representing this is not the aim of my project. So I just put a file in the root folder. I have chosen .csv as it is simple, can be automatically analízed with many tools, and Microsoft Excel supports it too. The filename is automatically generated from the RTC module’s date and time data. The secure file handling is provided by the implemented file opening and closing functions.

I have faced a challenging techical problem while I sarted desinging the data structure. The DS1302 real time clock module can only output seconds as the biggest time resolution. Therefore I needed a more precise time data too. I have discovered in the Arduino documentation that the microcontroller is counting the elapsed milliseconds since the last restart. This can be requested by the call of Millis() function.

After solving the timestamp issue I could start defining which data is necessary from the robotic arm. First of all, the system voltage should be continuously monitored. Then the encoder values of each servos. Additionally the 3D coordinates can be stored using the calculations presented in the attached project laboratory documentation.

## Software implementation

TBC

# Testing

## Software test

## TBC

## System test

## TBC

# Summary

TBC

# Future plans

This project has been planned and executed as a crucial part of my thesis written in the following two semesters. As the logging subsystem has been integrated successfully in the system, now I can go on developing a so called learning function in the software. That means after a specific command the robotic arm will be released and can be moved freely. The SD module will log the movement and it will be able to replay these predefined movements. This plan has many challenges too, I will have to define the speed of replay, or maybe make it depend on a parameter. A filtering will be needed too in the recorded data.

As this system from hardware point of view is complete and robust I am not planning to modify it, only the software will be radically changed.

With the upper mentioned learning function my project should support our life in many fields. The robotic arm’s structure is the same as the human hands, so it could memorize physiotherapic exercises to help the rehabilitation of the injured. Another interesting field would be recording movements and replay them million times in industrial fields. Of yourse, you can not move a huge robotic arm on your own, but my project is scalable up to the mid size robotic arms which are the most common in co-robot workplaces and processes.

# Bibliography