

Designing an Autonomous

Robot-Player for Connect-4

|  |
| --- |
|  |

  
  
Submitted to: Fontys University of Applied Sciences

June 2023  
Boris Ivanov

**Document data:**

Author: Boris Ivanov  
Student number: 2969300  
E-mail: [357544@student.fontys.nl](mailto:357544@student.fontys.nl)   
Date: 19 April 2023  
Place: Eindhoven, the Netherlands

**Company Data:**

Name: ALTEN Nederland  
Address: Hurksestraat 45, 5652 AH Eindhoven  
Company supervisor: Michael van der Velden  
Telephone: 0402563080  
E-mail: [Michael.van.der.velden@alten.nl](mailto:Michael.van.der.velden@alten.nl)

**University Data:**

Name: Fontys University of Applied Science  
Address: Nexus Building (ER, De Rondom 1, 5612 AP Eindhoven)  
School supervisor: Michal Mikołajczyk  
Telephone: 0618592672  
E-mail: [m.mikolajczyk@fontys.nl](mailto:m.mikolajczyk@fontys.nl)

**Approved and signed by the company supervisor**

Date:

Signature:

# Foreword

[PLACEHOLDER]  
This is an internship report on ‘Designing an Autonomous Robot-Player for Connect-4’. This project has been realized at ALTEN by Boris Ivanov on behalf of educational program Electrical & Electronic Engineering at Fontys University of Applied Sciences in Eindhoven. The project and this report were realized in the period of February 2023 – June 2023.

I was guided by my mentor Michael van der Velden.

[Continue]

**Table of Contents**

[Foreword 2](#_Toc130297905)

[Summary 4](#_Toc130297906)

[List of abbreviations 4](#_Toc130297907)

[List of figures & tables 4](#_Toc130297908)

[I. Introduction 5](#_Toc130297909)

[II. About the Company 5](#_Toc130297910)

[1. Background information 5](#_Toc130297911)

[III. Project description and assignment 6](#_Toc130297912)

[1. Project background 6](#_Toc130297913)

[2. Problem description 7](#_Toc130297914)

[3. Assignment 7](#_Toc130297915)

[4. Project scope 8](#_Toc130297916)

[5. Boundary condition 8](#_Toc130297917)

[6. Project approach: 8](#_Toc130297918)

[i. Development phases 8](#_Toc130297919)

[ii. Verification method (V-model) 8](#_Toc130297920)

[IV. Research 9](#_Toc130297921)

[1. Research objectives 9](#_Toc130297922)

[2. Main and sub-questions 9](#_Toc130297923)

[3. Research approach 9](#_Toc130297924)

[4. Results 9](#_Toc130297925)

[5. Conclusions 9](#_Toc130297926)

[V. Specification 10](#_Toc130297927)

[VI. System Design 11](#_Toc130297928)

[VII. Detailed Design/ Module Design 0](#_Toc130297929)

[VIII. Realization 0](#_Toc130297930)

[IX. Verification and validation 1](#_Toc130297931)

[1. Test set-up 1](#_Toc130297932)

[2. Test results 1](#_Toc130297933)

[X. Result analysis 1](#_Toc130297934)

[XI. Conclusions 1](#_Toc130297935)

[XII. Recommendations 1](#_Toc130297936)

[Evaluation 1](#_Toc130297937)

[Bibliography 2](#_Toc130297938)

[Attachments 2](#_Toc130297939)

[A. Original assignment 2](#_Toc130297940)

[B. Project plan 2](#_Toc130297941)

[C. Originality Declaration 2](#_Toc130297942)

[D. Confidentiality Declaration (optional) 2](#_Toc130297943)

[E. SRD, System Requirements Document (optional) 2](#_Toc130297944)

[F. SDD, System Design Document (optional) 2](#_Toc130297945)

[G. MDD, Module Design Document (optional) 2](#_Toc130297946)

[H. TRD, Test Report Document (optional) 2](#_Toc130297947)

# Summary

[written at the end]

# List of abbreviations

[continually updated]

|  |  |
| --- | --- |
| Acronym | Description |
| IT | Information Technology |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# List of figures & tables

[continually updated]

[Figure 1: The Connect-4 Robot [PLACEHOLDER] 6](file:///D:\University\Fontys\Year5\Internship\Documents\2.%20Documents\5.%20Final%20Report\Graduation%20Report.docx#_Toc130287566)

[Figure 2: OVERVIEW OF THE ROBOT PLAYER 11](#_Toc130287567)

[Figure 3: Level 1 of the software architecture 12](file:///D:\University\Fontys\Year5\Internship\Documents\2.%20Documents\5.%20Final%20Report\Graduation%20Report.docx#_Toc130287568)

[Figure 4: Level 2 of the software architecture for both cores 13](file:///D:\University\Fontys\Year5\Internship\Documents\2.%20Documents\5.%20Final%20Report\Graduation%20Report.docx#_Toc130287569)

[Figure 5: Level 3 of the software architecure for Cortex-M4 14](#_Toc130287570)

[Table 1: Project boundaries 8](#_Toc130287644)

# Introduction

[written at a later stage]

# About the Company

ALTEN is a global technology consulting and engineering firm. They provide research projects for technical and information systems divisions in the industrial, telecommunications, and service sectors. Their focus is on the conception and research for the technical divisions. Additionally, ALTEN provides networks and telecom architectures, as well as the development of IT systems for the information departments [1].  
As far as industries that rely on ALTEN for their business include, but are not limited to, telecommunications, computer systems, networking, multimedia, energy & life sciences, finance, defence, aviation, and information systems [2].

Diagram

Description automatically generated

Figure : Organogram

## Background information

Established in France in 1988, ALTEN is a global engineering and technology consulting firm with locations in 30 nations. ALTEN had 54,100 employees and earned 3.78 billion euros in revenue in 2022. 45% of the group's business is conducted in the French market [1].

Within the Netherlands, their expertise is in the following categories: ALTEN IT, Technical Software and Mechatronics, with the “Connect-4” project falling within the Mechatronics department.   
Technical software focuses on embedded systems, simulation & modelling, monitoring & control, and business critical systems. This includes anything from banking systems to traffic light control [3]. ALTEN provides end-to-end software engineering solutions, including software design, development, testing, integration, and maintenance, to its clients across industries.  
Mechatronics supports its clients by developing and improving its products with the latest improvements in technology. ALTEN's mechatronics services include designing and prototyping complex systems, simulation and modelling, control systems development, system integration, and testing and validation. The company has a team of experienced engineers who work closely with clients to understand their requirements and develop custom solutions that meet their needs [4].

This project is part of ALTEN’s in-house projects, which are often used to develop new skills for consultants or the ones of interns. Since ALTEN wants to demonstrate their competence in the field of motion systems it wanted to create a demonstrator around this. The Connect-4 (Four Up, 4-in-a-row) robot was developed for demos at trade fairs and open days at universities. The robot game is meant to demonstrate the knowledge of the consultants at ALTEN, and it is therefore developed with industrial components.

# Project description and assignment

*In this chapter you give all the details about the project assignment, in clear sections. The reader comes to know everything about the following areas:*

* *What is the initial situation? What is there to miss? Why is that a problem? What are the unintended consequences?*
* *What is the purpose of the project? What does the client achieve with it? What is the desired end situation?*
* *What is, by virtue of the two previous points, the precise assignment description?*

*In this chapter you clearly describe what does belong to and does not belong to your assignment. So if the company wants you to use a certain design method or apply a specific technique (FPGA, microcontroller, protocols, etc.) the reader can find it here.*

*If the assignment in the course of the project changes, this will be explained in this chapter.*

*So again:*

1. *Successful problem definition, means clear goal of the project*
2. *Defining boundaries of the project. Boundaries are more conditions that must be met. E.g. if there is enough budget.*
3. *What is out of scope? For example, out of scope is: sw development is not part of the project.*
4. *Make sure that the final results/solutions could be verified if possibleЮda,*

## Project background

A picture containing indoor

Description automatically generated  
My graduation internship for Fontys Hogeschool will be conducted at ALTEN, with my task being to realize as many as possible software blocks and verify an embedded software architecture on an STM32H7 processor. This is for a Connect-4 robot player, shown in figure 1, which has had its architecture designed by another graduate student.   
The game is fairly simple to play. There is a seven-by-six rack board, with slots at each side for the two players to enter their tokens. A red one for the robot player and a yellow one for the human player. The first player to connect four tokens in a row in any direction wins.

Figure : The Connect-4 Robot [PLACEHOLDER]

The whole process, of playing the game, should be completely autonomous. After the player token has been placed in the idle robot, it can decide its next move based on a difficulty setting. To be able to play the game, the 4-in-a-row robot is equipped with numerous parts that help it achieve its task. The big ones are the two motors for movement in the X and Z direction, together with their encoders and home/end switches. Additionally, it has two servos, one to rotate the end-effector and another to open the board for resetting the game state. Also, the robot has an RGB sensor and a flipper to be able to sort and distribute the tokens to the correct sides. The robot’s end-effector is equipped with a vacuum pump, vacuum sensor, and valve to be able to pick up tokens. Finally, the machine can detect when and where a token is dropped, through a series of IR sensors on the entrance of the board.

The project has existed for several years, and several major changes have occurred during its existence. The one that concerns the current state, is the change of processor used in the system. Before, the system used a single-core processor, but with constant improvements in functionality and new additions, the system started to become slower and unresponsive. Therefore, it was decided that a new processor will be put into the system. The dual-core STM32H755 is more powerful than its predecessor and fits with the newer requirements. The initial idea was that one core would be responsible for the real-time processing, while the other core would be the “primary” core and it will delegate tasks and take care of the higher-level logic like the game decisions, displaying results and more.

## Problem description

With the newly added dual-core processor the system had to undergo a major restructure of its software architecture and its PCB design. These two tasks were undertaken by previous interns. However, the software architecture wasn’t realized or verified due to time constraints. Several “demos” were made to showcase some parts of the architecture working together, and in theory it should follow that the rest should work as well. However, neither demonstration code has been extensively verified. More information about the demonstration projects will be included in Chapter IV.   
The above-mentioned software projects have had slightly different goals. One is the initialization procedure, another one is low-level code about different peripherals, and finally the communication between the two cores.

## Assignment

The assignment, therefore, is to combine and review if the previous systems work as intended. Write additional code that supports the operation of the Connect-4 robot player. Including, but not limited to: designing high-level logic for different system sub-modules from the previously designed architecture, designing libraries for sensors (RGB sensor, IR sensor) and peripherals (GPIO, motors, encoders, etc.), implementing low-level logic ( EXTI, NVIC, HSEM, etc.). To sum up, the task is to bring the robot to an operational level by designing and implementing the necessary elements and validating the previous work done.  
Additionally, ethernet communication with the systems should be investigated. This would be the starting ground for future upgrades of the system. This would have to facilitate communication with the internet, the transfer and receiving of data to keep high scores, current player’s turn, a human machine interface, etc.

## Project scope

The project is concerned with the re-evaluation (and if needed redesign) and implementation of the previously designed software architecture. The dual-core communication is worked out, but the rest of the modules have to be implemented (refer to point 3 in this chapter for examples). The programming language will be C.

|  |  |
| --- | --- |
| **Project boundaries** | **Within Scope ?** |
| Implement software modules | Yes |
| Redesign software modules | Yes |
| Research ethernet communication | Yes |
| Implementing ethernet communication | No |
| Redesign hardware/mechanics | No |
| Changes to the gameplay | No |

Table 1: Project boundaries

## Boundary condition

Boundary conditions are essential to ensure that a project is completed within the specified limits and to prevent any unwanted consequences. In the context of the Connect-4 robot player project, there are several boundary conditions that should be considered. These include hardware limitations, time constraints, and more.  
The project must be completed within the allotted timeframe, and deadlines for each stage of the project must be established to ensure timely completion. The nature of merging different software project at different points of completion is usually very time-consuming.  
The hardware limitations of the robot player must be considered during the design and implementation of the software. The STM32H7 processor has a limited amount of memory, there are a lot of other hardware components with varied points of failure.

## Project approach:

### Development phases

The project will follow the normal V-model development procedure. However, since the project has been under development for quite some time, a big part of the verification phase is complete. The system and the architecture of the said system have been designed, together with parts of the different lower-levelled modules and their software implementations.

A part of this project will be the validation of the already made design choices and system/sub-systems (refer to Chapter VI) , through different means of testing (unit, module, integration) and a varied assortment of techniques (black-box, white-box, happy-path) and through the designing of newer modules that are to be integrated into the system.

### Verification method (V-model)

Chart, diagram, funnel chart

Description automatically generatedThe V-model is a development model that emphasizes the importance of testing and verification throughout the development process. It is suitable for the Connect-4 project because it involves a complex system with multiple components that must be integrated and tested thoroughly.

Figure : The V-Model

# Research

*The research aspect of your assignment must clearly come into your report. A rule of thumb is that your internship for at least 20% and for a maximum of 80% must consist of research and development. We speak not of scientific research, but of applied research. In this part of the report you describe surveys, information gathering methods, used literature and other resources and comparisons of possible methodologies, techniques, tools and solutions (if applicable) for your project.*

*Start the chapter with a short “Intro” (few sentences) to tell the reader what you will do in this chapter.*

*Avoid ambiguities, be anywhere specific. Example: "This led to delays" is much too vague. In such a case, you write: "forty percent of customers has, in the past year, received the invoice only five months or longer after the purchase."*

*Finalize every main chapter with an “outro”: tell the reader what he has to learned from this chapter*

*In this chapter, as in all other chapters, you refer to the References at the end of the main part of the report (before the Attachments) in the IEEE way. Apply MsWord “Bibliography” for this.*

*Make sure that you write all in your own words to avoid plagiarism!*

## Research objectives

Validate the previously designed software architecture and figure out how to combine the existing legacy code with the code written to demonstrate the workings of the new architecture and PCB.

## Research

* STM32 HAL
* Existing Projects
  + Dual-Core Communication Blinky (Pascal)
  + Dual-Core Proximity & TaskGenerator & MotorX Code (Pascal)
  + Dual-Core Initialization (Laurens)
  + Legacy low-level code
* Hardware semaphores and their role in GameController-CM7, CM4TaskGenerator-CM7, TaskManager-CM4
* Mapping legacy code to new architecture (figuring out how to blend all the projects together)

## Results

## Conclusions

# Specification

*Make sure:*

1. *Clear definition of the test cases and their outcome*
2. *MoSCoW model for realistic overview*
3. *Can be described in a SRD (System Requirement Document) and attached*
4. *In this chapter you describe the outcome of the SRD*

# System Design

*Here you can e.g. introduce or start applying and the top-down structured design of your project.*

1. *Can be described in a SDD (System Design Document) and attached to the report*
2. *In this chapter you describe the outcome of the SDD*

In this chapter I will briefly introduce the system architecture that was designed when the project was handed over. About more detailed information, you may request access to the SAD [5]

The designer before me chose to describe the system by including different levels of abstraction to his work. In total there are 3 levels/layers to the software architecture, each of them describing the different modules needed to make the system functional. Level 1 has the highest abstraction, Level 3 the lowest. By designing and implementing from the lowest level, a clear path to completion is presented. Furthermore, by building up the lower-levelled blocks and through testing them, the stability of the system can be verified better, and debugging can be done more easily when building up the more complex blocks, which are comprised of the already mentioned “smaller” blocks.   
  
What can be seen in figure X is the overview of the system, with the different communication methods required by each peripheral. Blue signifies that the module needs to be verified. Red  
means that this is not implemented on the system and will not be worked upon during this project. The objects in the blue-dashed blocks are the different sub-modules of the system. Each inner block of the sub-modules describes the hardware used to achieve the task, coloured in darker yellow.

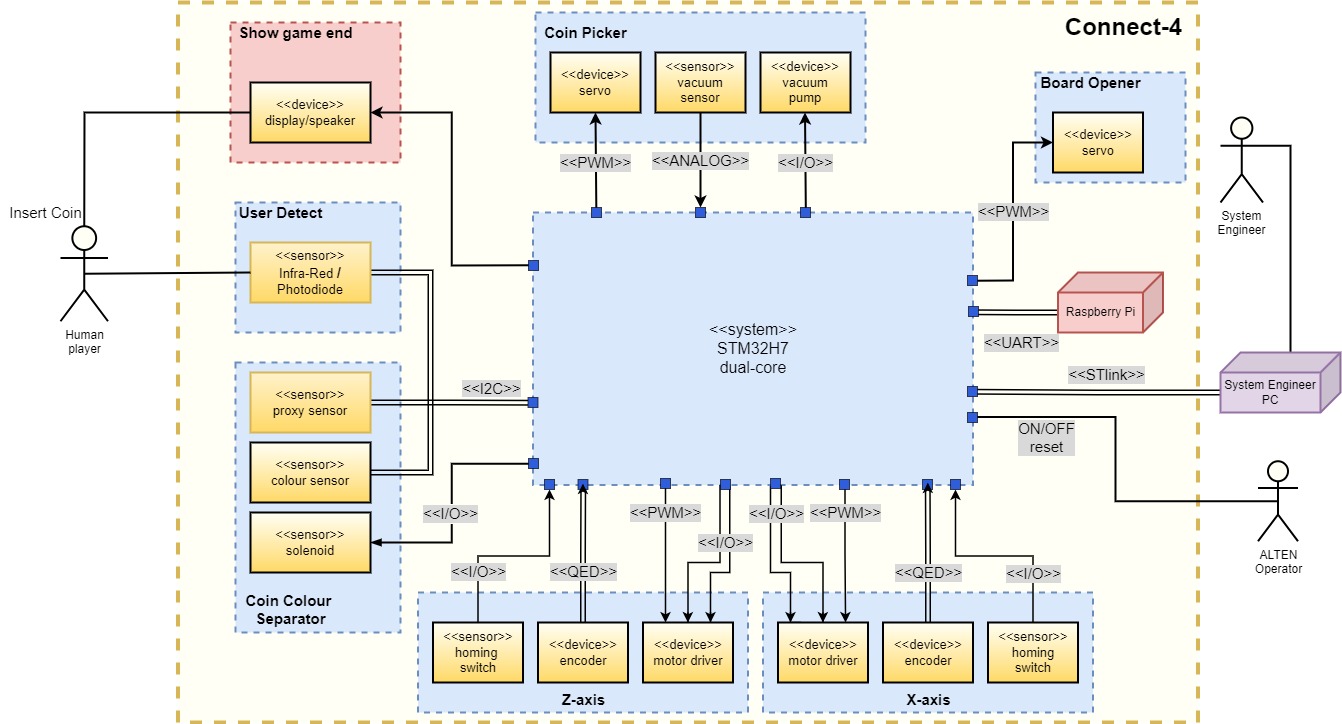


Figure 4: OVERVIEW OF THE ROBOT PLAYER

Diagram

Description automatically generated

The first level of the architecture describes that the core Cortex-M7 (referred to as M7) will take care of the “higher-level” logic like the next-move decision, delegating tasks, and the bulk of the upgrades for the future will be done on this core. It will be the primary core of the system, while Cortex-M4 (referred to as M4) will be the secondary core of the system. It will take care of the real-time processing and it will act upon tasks given from Cortex-M7. The core will drive the motors, separate and pick the tokens.

Figure : Level 1 of the software architecture

Diagram, schematic

Description automatically generated

The second level is as deep as it goes for Cortex-M7, since the rest of the functionality is out of scope for this project and is for future upgrades. This layer describes the blocks needed to facilitate the primary function of the core, the ablility to give out tasks and set-up the whole system. The blocks describe the different functions that need to be designed to make the layer complete and functional, this is true for both cores.   
  
  
For Cortex-M4 the second layer describes another set of controllers, this time for the hardware of the robot. It is apparent that another level would be needed to explain the full functionality of these blocks.

Figure : Level 2 of the software architecture for both cores

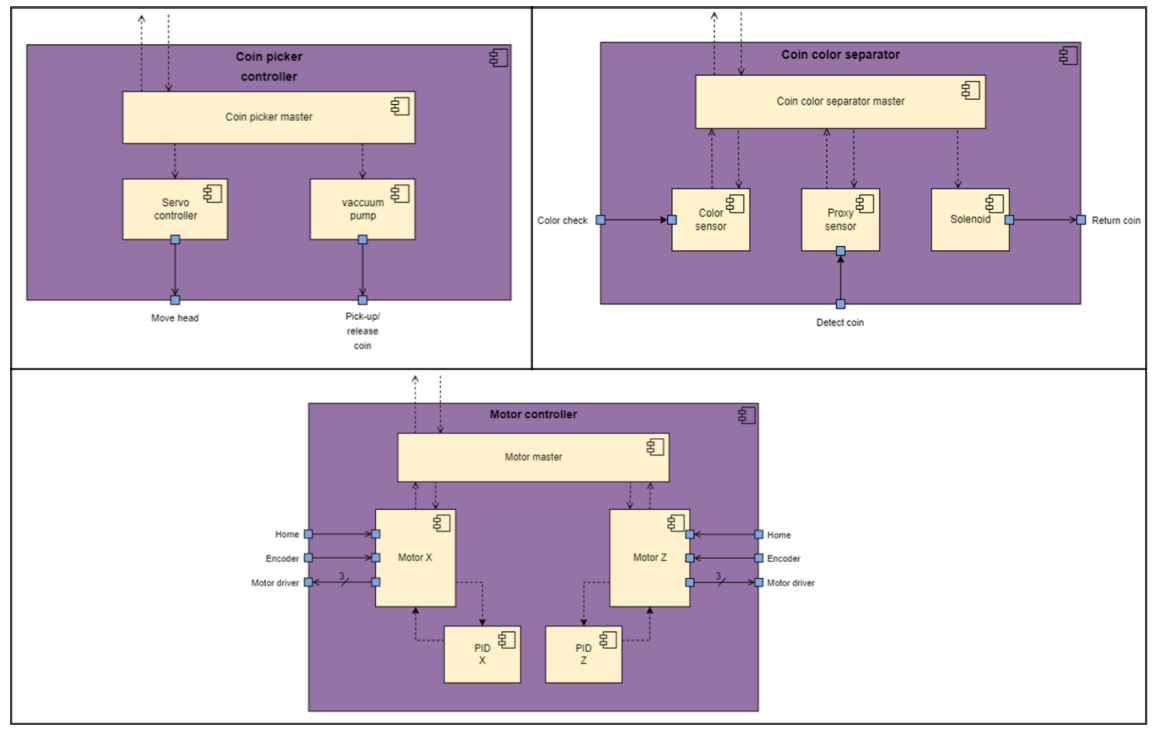
Layer 3 is the last one from the software architecture. The blocks here describe libraies that need to be designed to the different perephirals.

Figure : Level 3 of the software architecture for Corex-M4

As mentioned before, this is the work of a previous assignment and further detail is saved due to brevety. [5]

# Detailed Design/ Module Design

1. *Can be described in a MDD (Module Design Document) and attached to the report*
2. *In this chapter you describe the outcome of the SDD regarding the design (calculations, simulations, schematics, software)*

# Realization

1. *Can be described in a MDD (Module Design Document) and attached to the report*
2. *In this chapter you describe the outcome of the SDD regarding the building of the prototype*
3. *The test plan and test report of the MDD can be used as source*

# Verification and validation

## Test set-up

## Test results

# Result analysis

# Conclusions

*The reader should be able to understand this chapter even when he, immediately after he has read the introduction and chapters, has skipped all intermediate: make sure you connect the content within the conclusion chapter!*

*The reader who has read the whole report, should encounter no new information in this last chapter, indeed: he must be able to predict what it says! In this chapter the results are compared with the initial assignment (requirements/specifications) 15*

*and conclusions are drawn. Do not draw conclusions that are not underpinned with previous mentioned results. Conclusions coming out of the blue are not acceptable!*

*Recommendations (could be a separate chapter) tell the reader what should be improved or still has to be done in order to complete the assignment*

*This last chapter has no figures or lists. The maximum length is one page.*

# Recommendations

# Evaluation

*This is not a chapter, and therefore has no number and no sections. Just like the foreword or preface the evaluation is a personal part of the report and you can write this component also in the ‘I’ form. You reflect on the experiences you have had during the project. You oversee the whole journey and you discuss what you've learned. You describe what you've found and what you remember as your most "teachable or valuable moments" i.e.: when did the error(s) or problem(s) occur and why; especially how you've solved the problems and again emphasize that!*

*This is not the place to settle outstanding accounts. But suppose there was a profound reorganization at your Department, where many people are transferred or dismissed, then of course this has influenced your work, and then you need to mention this. But do this carefully, without offending somebody.*

*Finally it is advised to take some time to look back at and evaluate your study. First compare your graduation time, subjects, needed skills, needed knowledge, etc. to that what you have learned at Fontys Engineering. Which subjects, courses, practical’s and projects were helpful or even indispensable. Also you could advice how to change the curriculum of Fontys Engineering from every possible view point. Adding or deleting subjects and/or courses, change practical’s, change the way of teaching, you name it. This will help Fontys Engineering to keep the curriculum updated and in that way Fontys Engineering is able to educate the engineer of the future!*

*To be clear: this part is not often written in (business) reports. But some universities do want this part to show your competences and your (positive) critical view on your education. Fontys Electrical Engineering is happy with this separate chapter as a learning experience for the study.*

# Bibliography

|  |  |
| --- | --- |
| [1] | “ALTEN - 2022 report,” 2022-11-02. |
| [2] | “Alten - Services,” ALTEN, [Online]. Available: https://www.alten.com/services/. [Accessed 03 2023]. |
| [3] | ALTEN, “Technical Software,” ALTEN, 2023. [Online]. Available: https://www.alten.nl/en/technical-software/. |
| [4] | ALTEN, “Mechatronics,” ALTEN, 2023. [Online]. Available: https://www.alten.nl/en/mechatronics/. |
| [5] | P. Faatz, “Software Architecture Document,” https://redmine.alten.nl/projects/in-a-row/repository/192/revisions/758/show/51.%20Software%20Engineering/1.%20Repo/trunk/1.%20Design/Dual-core%20low%20level%20design, 2022. |

# Attachments

## A. Original assignment

## B. Project plan

## C. Originality Declaration

## D. Confidentiality Declaration (optional)

## E. SRD, System Requirements Document (optional)

## F. SDD, System Design Document (optional)

## G. MDD, Module Design Document (optional)

## H. TRD, Test Report Document (optional)