Project plan

BECKHOFF PLC ROBOT ARM

Date	**:**	**23-03-2023**
Version	**:**	**0.1**
Status	**:**	**Setup**
Authors	**:**	**Luuk Aarts, Rik van Heesewijk, Dimitar Dyulgerov, Jorn Kersten**

Figaroa, Oswald

Fontys Hogescholen

Version

Version	Date	Author(s)	Amendments	Status
0.1	23-03-2023	Luuk Aarts, Rik van Heesewijk, Dimitar Dyulgerov, Jorn Kersten	Initialization project plan	

Communication

Version	Date	To
0.1	23-03-2023	Oswald Figaroa, Luuk Aarts, Rik van Heesewijk, Dimitar Dyulgerov, Jorn Kersten

Index

1. Project Assignment

1.1 Context

We will be creating this project for Fontys HBO-ICT in Eindhoven. The goal of the assignment is to have a robot arm between two machines in a factory and move a product from one machine to the next. The machines are filling bottles with fluid, afterwards the fluid is processed and put in a correct slot.

1.2 Goal of the project

The goal of this project is to create software that controls an Igus robot arm. The robot arm is going to be used to transport payload between two stations. The robot arm's movements ought to be smooth, with little-to-no overshoot. Along with the PLC software, a digital twin needs to be developed in order to simulate and visualize the hardware in a digital environment. A digital conveyor belt is going to be used to provide coordinates for the robot arm to move to. The robot arm is going to be controlled by a Beckhoff PLC. The digital twin is going to be connected to the PLC and be controllable by a user interface. Upon completion of the project, a higher degree of automation can be achieved in the production process.

1.3 The assignment

The assignment consists of creating PLC software to control an Igus robot arm, a digital twin, and a digital conveyor belt. The digital twin must be able to send commands to the hardware robot arm and vice-versa. The movement of the robot arm must have minimal overshoot. The robot arm will move in accordance with the directions given by the digital twin/digital conveyor belt.

1.4 Scope

The project includes:	The project does not include:
1 Moving the robot arm through a Beckhoff PLC, using ST 2 Digital twin 3 Digital conveyor belt 4 CANopen communication between PLC and Robot arm 5 Communication between visualization and hardware 6 Sending position commands to the hardware through the digital twin	1 Robot arm grabbing a physical object 2 Physical conveyor belt 3 Detection of physical items on a conveyor belt

1.5 Conditions

The preconditions at the time of writing are as follows: A Beckhoff plc needs to be used alongside with TwinCAT and structured text to program an Igus robot arm. The communication between the robot arm and the PLC need to communicate via CANopen.

1.6 **Finished products**

1.6.1 Beckhoff plc

On the Beckhoff plc we reconstruct the current project written in C# with TwinCAT in structured text. Then the plc will have to connect via a CANopen protocol with the Igus robot arm.

1.6.2 Visualisation

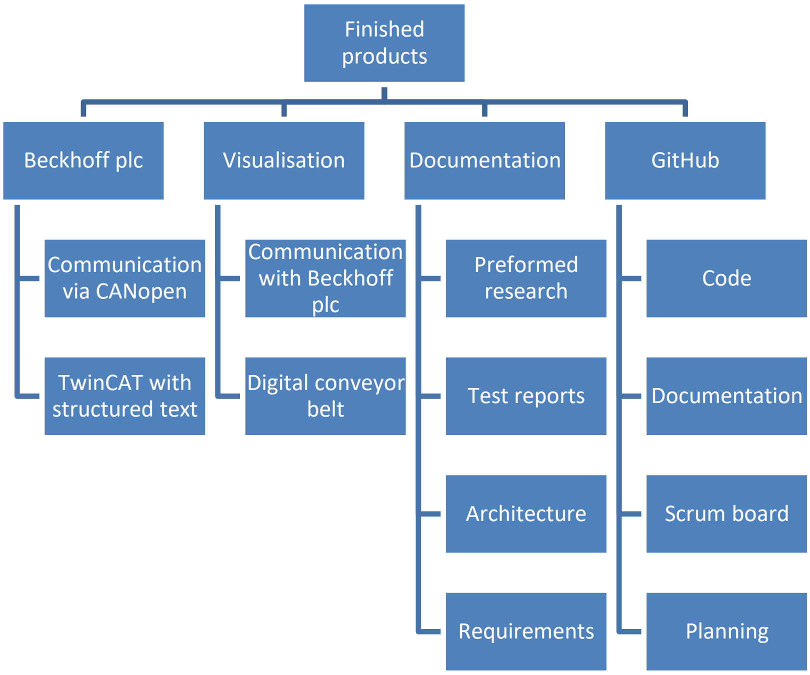
A visualisation of a conveyor belt will have to be created. This visualisation will be a 3D model in a program of choice, and it will communicate via a communication of choice with the Beckhoff plc.

1.6.3 Documentation

Documentation is going to be written during the project. All documents will be available to the stakeholders. The main documents that will be created are research, test reports, project architecture, and project requirements.

1.6.4 GitHub

All created code and documentation will be uploaded to GitHub where all data for this project is going to be stored. As well as code and documentation, GitHub will also be used to keep track of our planning and scrum board.



1.7 Research questions

In order to start with our project, we are going to have to do some research first. The Beckhoff PLC needs to run on certain software and with the right input and control commands. For us to research this quickly and effectively, we will set up some questions we will have to answer before we start working on the project.
More questions might be added further into the project, the main questions will stay the same as they are the guideline for what we want to achieve with this project.

How can we control the movement of the Igus Robot Arm?

- Which program can we use to write the code for the Igus robot?
- What protocol are we going to use to communicate between the virtual and physical robot?
- How can we move a joint?
- How are we going to acquire the movements of each joint to get to a desired position? (Inversed kinematics, inversed geomatics)

How can we visualize the movement of the Igus Robot Arm?

- Which program can we use to visualize the Igus Robot movements.
- What protocol are we going to use to communicate between the virtual and physical robot?

2. Approach and Planning

2.1 Approach

We will make use of the Scrum method. We will start every day with a stand up where we discuss the things we will be working on, and the problems we are running in to. We will also be using sprints to work towards the final products in small steps. We will have 6 sprints where we will all be working on a small part of the project to get to finished product. The sprints will be 2 weeks long. At the start of every sprint, we will be discussing who will do what during that sprint. We will be using GitHub for our planning board. There we can all fill in things that need to be done and then we can pick up the problem and work on it during one of the sprints.

2.2 Research methods

Throughout this project we will do the necessary research following the DOT framework. The rules regarding this framework are explained at ictresearchmethods.nl. We will keep track of all the information we find by documenting our research and saving the sources for later use.

2.3 Breakdown of the project

Sprint 1: In the fist sprint we will be making the Project plan and installing the software
Sprint 2: Undergo research for visualisation and trying to move the robot arm with structured text
Sprint 3: Create a virtual conveyor belt, and have communication between the visualisation and the PLC. Try to get a connection between plc and robot arm via CANopen.
Sprint 4: Have movement of the arm by sending coordinates to the robot arm. If visualisation is connected to the PLC try to move arm via the visualisation.
Sprint 5: Try to complete the conveyor belt and other finishing touches.
Sprint 6: 3D modelling/bugfixes and other finishing touches.

3. Project Organization

3.1 Team members

Name + Phone + e-mail	Role/tasks	Availability
Contact details	*Mention the role or any specific tasks*	*What availability of the person is necessary (e.g. 3 days a week, during phase 2)*
Jorn Kersten – 06-20747022 – 483331@student.fontys.nl	Sprintmaster	2 days a week for the whole project
Dimitar Dyulgerov 479370@student.fontys.nl	PLC dev, digital twin help	2 days a week for the whole project
Rik van Heesewijk – 0616291271 – 483622@student.fontys.nl	Dev	2 days a week for the whole project
Luuk Aarts – 0622056181 – 488397@student.fontys.nl		2 days a week for the whole project

3.2 Communication

Communications between the team and the customer happen through email, Teams, and weekly meetings.
Communications between team members happen through WhatsApp, Teams, Discord, and weekly in-person meetings. In case a team member is unable to participate in a meeting in person, they can join online through one of the aforementioned platforms.

3.3 Testing approach and Test environment

Throughout the project, manual testing is going to be the most common testing method. Code reviews will be performed by a team member before the code is approved and marked as done.

Testing is going to be done in a digital environment, before deploying the code to the hardware, in order to avoid critical issues and hardware damage.

The digital twin or a barebones simulation is going to be used as a test environment.

3.4 Configuration management

The project is going to use GitHub as a GIT repository and version control. The main branches are

- main: The release branch, all code there is tested and reviewed, ready to be demonstrated
- dev: Used to merge other branches into and see if any conflicts arise. Also used for general development
- digital_twin: Development work on the digital twin
- plc_dev: Development work on the plc software.

More (sub-)branches might be added during the development cycle if necessary.

4. Finance and Risks

4.1 Risks and fall-back activities

<table>
<tr><th colspan="2">Risk</th><th>Prevention activities included in plan</th><th>Fall-back Activities</th></tr>
<tr><td colspan="2">1 Hardware damage</td><td>Deploy the code to a simulation before deploying to the hardware</td><td>Repair hardware damage</td></tr>
<tr><td colspan="2">2 Team member’s laptop damaged/lost data</td><td>Regular pushes to the GIT repository</td><td>Redoing the lost work</td></tr>
<tr><td colspan="2">3 Sudden leave of a team member</td><td>Timely inform the rest of the team and staff</td><td>Redistributing the work, the remaining team members take over</td></tr>
<tr><td>4 Discovery of a better approach during development</td><td>Research thoroughly before committing to a solution.</td><td colspan="2">Evaluate the benefits and drawbacks of switching to the newly discovered approach and proceed accordingly</td></tr>
</table>