

# Project Report\*

\*As a fulfilment to the course: "Project course name", D7039E & E7032E. Lecturer: Jan van Deventer.

Martin Blaszczyk, Edward Cedegård, Niklas Dahlquist, Edward Källstedt, Albin Martinsson, Måns Norell  
*Computer Science, Electrical and Space Engineering Dept.*

*Luleå University of Technology*

Luleå, Sweden

{marbla-6, edwcde-4, nikdah-6, edwkll-7, mnsnor-5, albmar-6}@student.ltu.se

*Abstract—*

## I. INTRODUCTION

### A. Background and Motivation

Arrowhead is an initiative from Luleå University of Technology to create a unifying framework which can enable embedded devices to intergrate and interoperate services in an open-network environment. This framework and its approach will strongly contribute to the reduction of design and engineering efforts in the industry.

As part of the Engineering Project course for Master's students in Computer Science, Control and Electrical Engineering at Luleå University of Technology the goal is to implement and demonstrate how the Arrowhead Framework can be used in a factory setting using autonomous ground robots. This project report aims to give an overview of the thought process, workflows and results of the project.

The proposed solution is a robot which by implementing machine vision algorithms can navigate the factory floor while also being able to pick up object using it's arm and gripper. It utilizes the Arrowhead framework to integrate with the other parts of the model factory.

### B. Contributions

Our proposed solution is an example of how the Arrowhead framework is utilized in a scaled factory setting together with a ground robot. The design is small and modular making it easy for future improvements and easy evaluation of the Arrowhead framework alongside third-party open-source solutions. Secondly we propose a solution consisting of lines and QR-codes and how the robot can flow a path using a RGB-camera instead of the conventional sensors. With this camera solution the possibilities of navigating the factory floor increase compared to the basic conventional following solutions.

### C. Structure

This report is divided into...

## CONCEPTUAL DESIGN

### D. Mechanical structure

The design of the robot is influenced by the LEGO®Mindstorms®EV3 set. While the original set

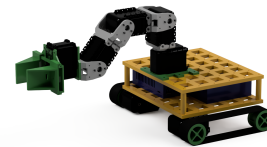


Fig. 1. Rendering of concept design.

has a movable base it was too small and without an arm and manipulator. It also does not incorporate the Dynamixel AX-12A Smart Servos used in the final design so the decision to redesign the robot with design inspiration from the LEGO®EV3 design. By developing and designing a new platform it gave the possibility to adapt the dimensions and fastenings. It also gave experience in manufacturing methods such as 3D-printing and Laser cutting. The final design is shown in Fig. 1 Consisting of seven Dynamixel®AX-12A Smart Servos for actuation, two on the base for locomotion and four on the arm and one for the gripping tool the robot has seven degrees of freedom. The main advantage of using the AX-12As is the possibility of connecting them in series which enables parallel control of all the joints. Additionally with the built in sensors the motors return feedback of the joint angles, angular speed, current draw, temperature to name a few.

### E. Electrical components

While the mechanical components are those visible it's the underlying electrical components that make the big difference. The internals in the Dynamixel®AX-12A Smart Servo motors give feedback on the state of the motors, the 1000mAh LiPo battery supplies power to the motors and surrounding electronics. For the computations the NVIDIA®Jetson Nano is used as it runs Ubuntu natively which makes the implementation of the underlying Robotic Operating System (ROS) [1] faster.

### F. Robotic Operating System

Because the robot and the task consists of many different parts it is preferred to utilize an easy to use and well docu-

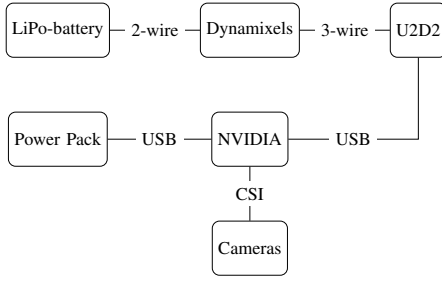


Fig. 2. Diagram of electrical components

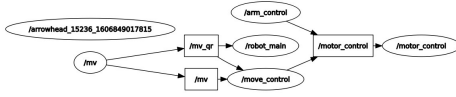


Fig. 3. Rosgraph of the robot.

mented framework for internal communication on the robot. Because of its many uses and popularity within the reasearch community this project makes use of the Robotic Operating System (ROS) as its underlying framework. ROS utilizes an internal TCP communication protocol where different parts can communicate and be synchronized which enables real-time performance. The rosgaph is shown in Fig. 3.

## II. PROJECT STRUCTURE

### A. Project planning

The projects course started in August 2020 and continued until mid January 2020 with the project deadline in the middle of December. To have a good structre and of the things to be done and synchronizing all the parts a plan was made to have a good overview of the project, shown in Table I. This plan enabled the group to be eflexible while still have a good long-term structure of the project. Every month a more detailed time plan was made to facilitate the small changes, dlays etc.

For the group members to what task were to be done, the built in function of Issues on the souce control platform Github® enabled all the project group members to see what had to be done every week. A Milestone was created for each week and populated with issues. When an issue was finished it coul simply be closed. If the issue was delayed it showed clearly in the Issue overview which tasks had to be prioritized.

### B. Source control

To keep track of the different software implementation the projects souce control implements Git in one common

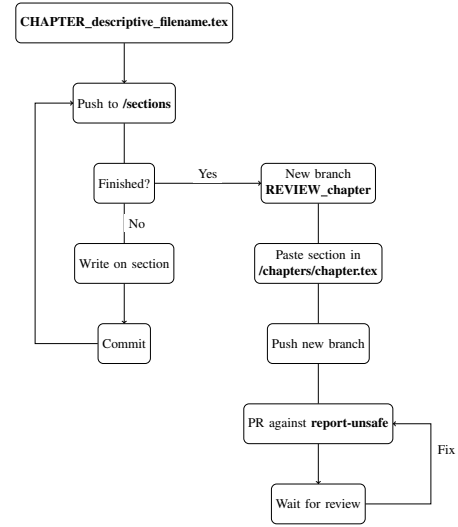
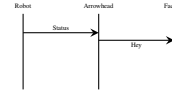


Fig. 4. Report writing flowchart



repository [2]. The repository is where all source code and relevand 3D-files are located. This report is written in LaTeX with Git as source control. To make sure it's easy for all members to write their designated sections a workflow was designed to to minimiza merge conflicts while writing drafts as show in Fig. 4. Without this flow the group members would experience merge conflicts after every push which would make it more complex and time consuming.

### C. Meetings

The group had two weekly meetings covering the previous week and the status of the project. This structure gave the students a great deal of responsibility to do work for each meeting while still maintaining a good structure of the project and encouraged discussions.

## REFERENCES

- [1] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system:" [Online]. Available: <https://www.ros.org>
- [2] Students, "D7039e," <https://github.com/kottz/D7039E>, 2020.

TABLE I  
OVERALL TIME PLAN

Sep	Oct	Nov	Dec
Concept generation	Evaluation	Evaluation	
Theory	Prototyping	Evaluation	Finishing up
Simulation	Evaluation	Evaluation	
Prototyping	Final Design	Evaluation	