DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

ARCHITECTURAL DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
SPRING 2024

ROBO CREW
RV8 WORKCELL

AKSHAY PALURI
AMEEN MAHOUCH
MUHAMMAD ANAS
HYUN HO KIM
KUNDAN SINGH MAHATO

Revision History

Revision	Date	Author(s)	Description
0.1	02.12.2024	AM, MA, AP, HK, KM	document creation

Contents

List of Figures

List of Tables

1 Introduction

This product is a high level working prototype of an industrial level robot which serves the purpose of spraying, coating, painting, and more uses due to their high flexibility and protection. The prototype will spray paint to the specific area designated to spary with paint ball marker. The robot arm is connected to a PLC controller and the arm is set on to an additional axis (i.e. linear rail) which is also connected to a servo amplifier present in the PLC. This PLC controller is connected to a PC containing various software to control the robot arm. Welcome to the future of seamless and superior surface finishing, where our robot not only meets but exceeds the demands of the most intricate and challenging projects.

2 System Overview

This section defines the top-level logical view of the design and describes the detail architectural strategy for the RV8 work cell system flow. Three separate layers make up the structure of the system: input, processing, and output. With the help of these layers, which are all essential to the system, the robot is able to carry out dynamic tasks that are dependent on the processing of sensor data. A high-level block diagram that shows the connections and interactions between different layers is included in this part to provide readers a thorough understanding of the architecture of the entire system. A robotic system's input layer, which makes use of sensors like a gate sensor and an E-stop is essential for task execution. E-stops offer an emergency shutdown signal that allows the system to be stopped right away. The robot arm's speed is controlled by the Gate sensor, which also communicates with the controller to find out the cage gate's present state. Efficiency and safety are prioritized in this design, which allows the robot to respond and adapt to a variety of scenarios while integrating security features like gate status monitoring and emergency stops. With the integration of sensors will perform the operation such as splashing multi-color paint to the desire target with great level of precision. PC and the trigger switch will be inter-connected to the PC to perform the necessary communication to perform the given task.

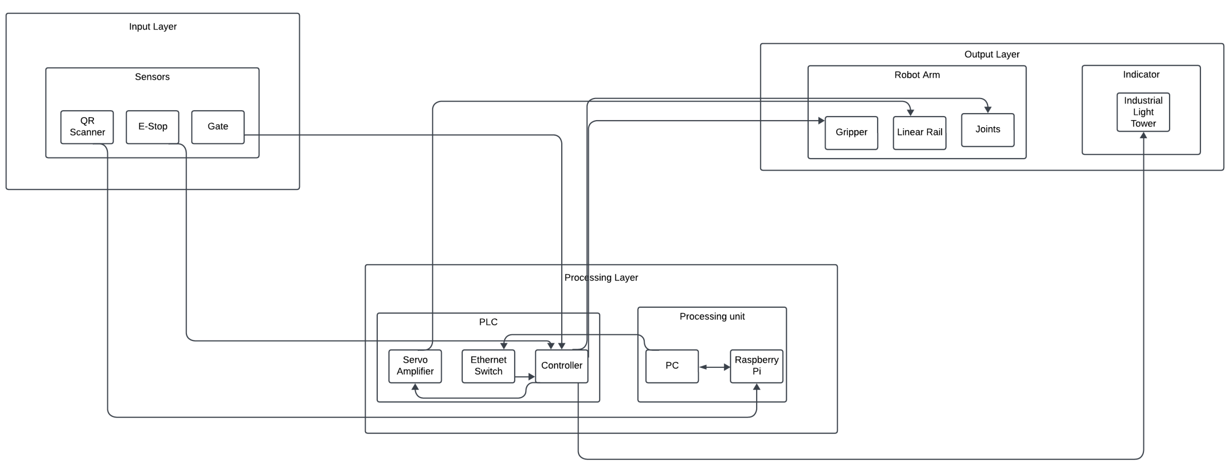

Figure 1: System architecture

3 Input Layer Subsystems

These sensors will enhance the safe and effective use of the RV8 robot for users. Firstly, the vision sensor assists the robot in aiming at targets. Secondly, the E-stop feature halts the system whenever a user presses the button or it's triggered by the computer. Lastly, the Gate sensor indicates whether the case gate is open or closed. Hence, all the inputs will be transferred to the PLC to initiate actions.

3.1 Layer Hardware

The sensors, including the vision sensor, e-stops, and gate, are connected to the programmable logical controller (PLC) via wires. They continuously transmit input values to the PLC, enabling the system to make informed decisions about subsequent actions.

3.2 Layer Operating System

RT ToolBox 3 Pro can be installed on both Windows 11 and Windows 10 operating systems.

3.3 Layer Software Dependencies

RT ToolBox 3, which controls the robot, will be used to facilitate the movement of the robot arm.

3.4 Vision Sensor

Vision sensors serve multiple purposes, including aiming at a target and tracing its movement. Additionally, it incorporates a safety feature: if there is any movement detected by the camera, it will automatically halt the program to ensure safety. This sensor will be mounted on the robot arm, enabling it to adjust its position by manipulating the robot arm's movements.

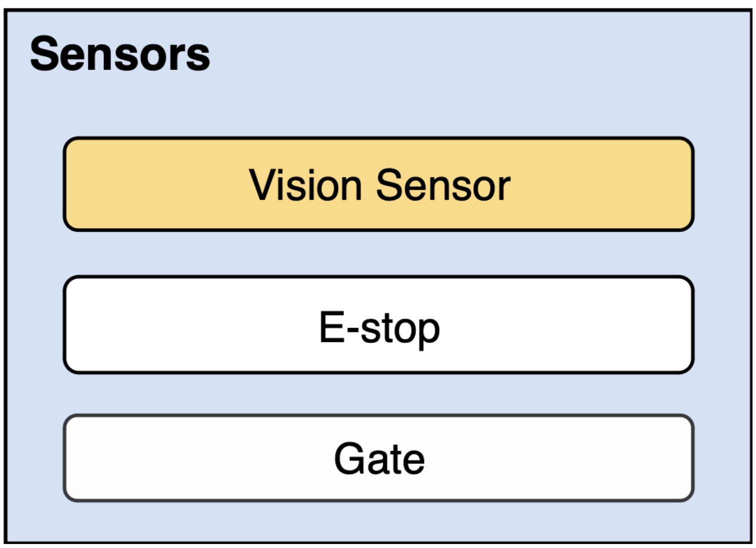

Figure 2: Vision subsystem diagram

3.4.1 Subsystem Hardware

A vision sensor will be mounted on the RV8 robot arm with a paintball gun.

3.4.2 Subsystem Operating System

Window 10 or 11

3.4.3 Subsystem Software Dependencies

Vision sensors will require software for configuration, calibration, data visualization, or integration with the robot control system, RT ToolBox 3 is a suitable option.

3.4.4 Subsystem Programming Languages

MELFA-BASIC IV programming language that is used by the Mitsubishi robots in the RT Toolbox 3 software.

3.4.5 Subsystem Data Structures

The information obtained from the vision sensor will be transmitted through the sensor to the PLC and to the connected computer.

3.4.6 Subsystem Data Processing

Extracts information from images or video streams captured by vision sensors.

3.5 E-stop

There will be multiple emergency stop (E-stop) buttons strategically placed throughout the system. These buttons will be located next to the case gate, on the controller, within an application (such as RT Toolbox) inside the cage, and alongside the linear rail. When any of these E-stop buttons are pressed, the entire system will immediately halt, ensuring rapid response to any safety concerns or emergencies.

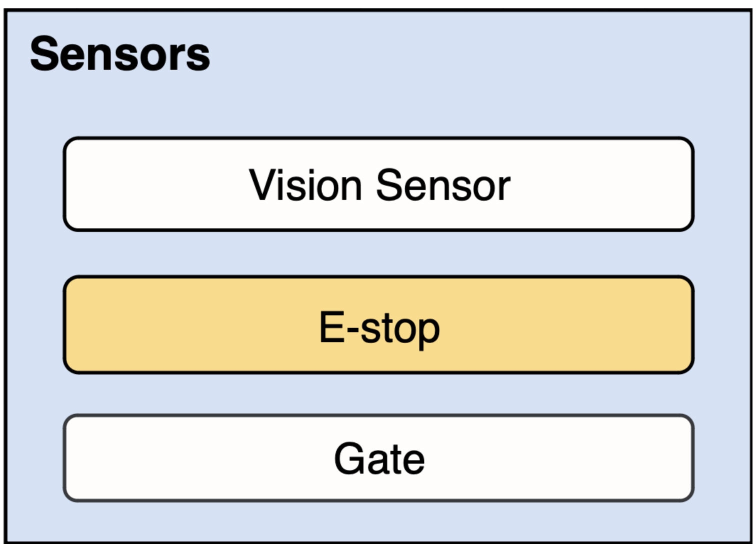

Figure 3: E-stop subsystem diagram

3.5.1 Subsystem Hardware

Inductive Proximity Sensors are used to limit the range of movement of the robot arm along the X-axis, E-stop sensors are utilized for emergency situations (True/False), and wires are employed to connect them to the PLC.

3.5.2 Subsystem Operating System

Window 10 or 11

3.5.3 Subsystem Software Dependencies

RT Toolbox 3 will be necessary to register the stop sensors.

3.5.4 Subsystem Programming Languages

MELFA-BASIC IV programming language

3.5.5 Subsystem Data Structures

True or False values from the physical stop sensor are sent to the PLC.

3.5.6 Subsystem Data Processing

capturing and analyzing signals generated when the emergency stop button or switch is pressed.

3.6 Gate

A contact sensor will be necessary to detect whether the gate is in closed or open position by physically making contact with the gate or its components. This is essential for safety measures to prevent the robot from damaging properties outside of the case.

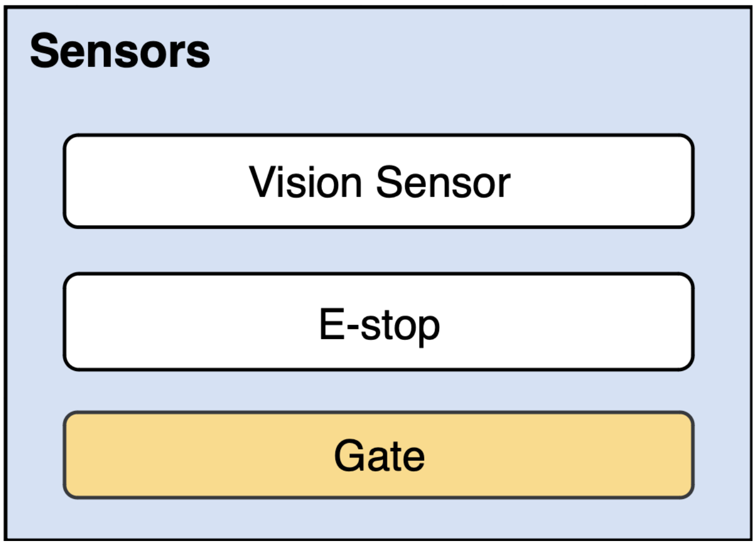

Figure 4: Gate subsystem diagram

3.6.1 Subsystem Hardware

Contact sensors and PLCs need to be connected via wires to receive the signal.

3.6.2 Subsystem Operating System

Window 10 or 11

3.6.3 Subsystem Software Dependencies

RT Toolbox 3 will be necessary to register the contact sensors.

3.6.4 Subsystem Programming Languages

MELFA-BASIC IV programming language

3.6.5 Subsystem Data Structures

True or false values will be measured through the sensor, and this value will be transmitted to the PLC via wires.

3.6.6 Subsystem Data Processing

detects the physical contact or proximity of an object to a specific area, typically through the closure or opening of an electrical circuit.

4 Processing Layer Subsystems

In this section, the processing layer's hardware and software design are delineated. The layer comprises a PLC, Raspberry Pi, and a PC interconnected to facilitate communication and decision-making processes. The PLC primarily governs the robotic arm's movements, including all joints and an additional axis, typically a linear rail. Conversely, the Raspberry Pi is tasked with data collection and transmission to the PC for decision-making regarding task execution on incoming objects, typically boxes. Communication between the PC and PLC ensures alignment of all arm joints for task performance. This section delves into implementation details, covering hardware components, programming languages, software dependencies, and operating systems relevant to the processing layer. Unnecessary details, such as purely software modules without specific hardware components, are omitted for conciseness. Adjustments to the organization, titles, and content are permissible to suit the project's requirements.

4.1 Layer Hardware

The hardware involved with the processing layer include a MELSEC Programmable Logic Controller, issued by Mitsubishi. The PLC is a specialized computer used in industrial automation and is paired with software to make decisions based on sensor data like temperature, pressure, or motor position. Finally, the PLC is equipped with an ethernet switch that enables communication between the Host PC and PLC.

4.2 Layer Operating System

The operating systems required by the layer are dependent on the specific components. The PLC runs a real-time operating system (RTOS), while the Raspberry Pi utilizes a Linux-based operating system Raspbian, and the host PC runs Windows 10.

4.3 Layer Software Dependencies

There are currently no software dependencies.

4.4 Host PC

The host PC acts as a powerful processing unit for in-depth analysis, decision making, and control. It receives pre-processed data from the Raspberry Pi, including data from the QR code, and it performs trajectory planning and higher level decision making.

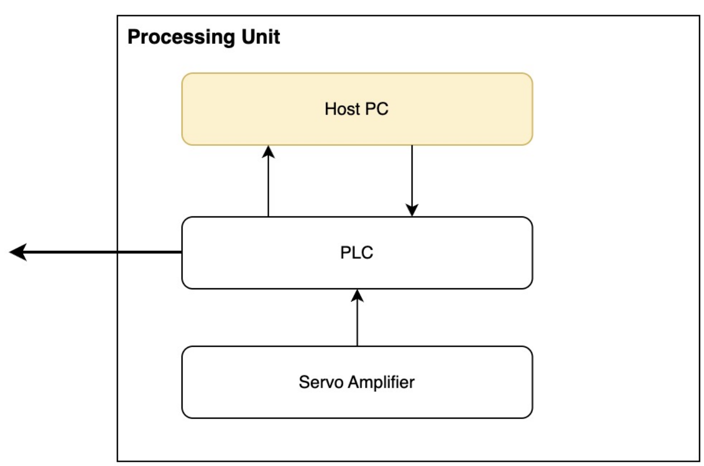

Figure 5: Host PC Subsystem

4.4.1 Subsystem Hardware

This subsystem consists of a Dell desktop computer with USB connections to the RV-8CRL controller.

4.4.2 Subsystem Operating System

The PC runs Windows 10.

4.4.3 Subsystem Software Dependencies

The PC runs several software for the system including RT ToolBox3, GXWorks, MRConfigurator2, and Creality.

- RT ToolBox is a software developed by Mitsubishi used for visual programming of the robotic arm, acting as a 3D simulator.
- GXWorks is a software suite provided by Mitsubishi Electric for programming and configuring PLCs. It is a comprehensive development environment that supports various programming languages such as ladder logic, function block diagrams (FBD), and structured text (ST).
- MRConfigurator2 is a software tool used for programmign Mitsubishi's servo drives and motion controllers.
- Creality is used for 3D modeling and design. Primarily, this software is used for designing mechanical components.

4.4.4 Subsystem Programming Languages

The Host PC uses Python scripts for network communication. The RT ToolBox software supports MELFA-BASIC-VI programming language.

4.4.5 Subsystem Data Processing

Upon receiving preprocessed data from the Raspberry Pi, the Host PC engages in thorough data analysis. This includes examining sensor data performing image processing if necessary. Algorithms for noise reduction, edge detection, and image segmentation may be employed to extract relevant information from sensor readings or images. The Host PC manages communication with the PLC using Ethernet/IP. Data packets are assembled and parsed according to the protocol specifications, ensuring reliable and efficient data exchange between the PC and the PLC.

4.5 Additional Axis Servo

This subsystem focuses on controlling the additional axis, such as a linear rail, to facilitate precise movements of the robotic arm.

4.5.1 Subsystem Hardware

The subsystem consists of a RollOn linear axis on a Megadyne belt. The servo amplifier is a MELSERVO MR-J4 series. Connections between the servo amplifier and PLC occur through a USB-A.

4.5.2 Subsystem Software Dependencies

Servo amplifier parameters are dependent on MRConfigurator2, where they can be specified through read/write operations.

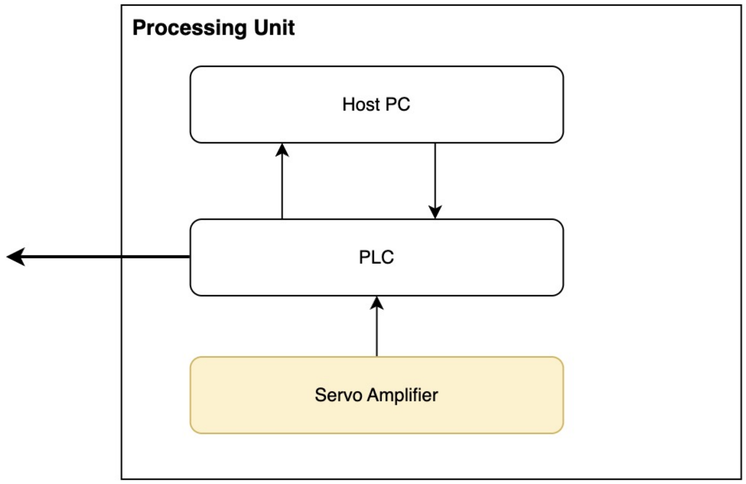

Figure 6: Servo Subsystem

4.5.3 Subsystem Data Processing

The servo amplifier receives control signals from the controller and translates them into precise voltage and current outputs to drive a servo motor. It continuously monitors feedback from sensors to maintain accurate motor position and velocity, adjusting operation as needed to minimize discrepancies. Additionally, servo amplifiers incorporate safety features to prevent damage and ensure safe operation of the motor system.

4.6 Programmable Logic Controller

The PLC acts as the central component of the processing layer, responsible for coordinating and executing control tasks for the robotic arm and additional axis. This subsection outlines the hardware, software, and data processing aspects specific to the PLC.

4.6.1 Subsystem Hardware

The PLC consists of a MELSEC Programmable Logic Controller manufactured by Mitsubishi, equipped with an integrated ethernet switch for communication with the host PC.

4.6.2 Subsystem Operating System

Operating on a real-time operating system (RTOS), the PLC ensures precise timing and reliable execution of control logic.

4.6.3 Subsystem Software Dependencies

Software tools such as GXWorks and MRConfigurator2, provided by Mitsubishi Electric, enable the programming and configuration of the PLC for control tasks.

4.6.4 Subsystem Programming Languages

Programming languages supported by GXWorks, including ladder logic, function block diagrams, and structured text, are utilized for developing control algorithms tailored to the application. RT ToolBox is written in MELFA BASIC VI

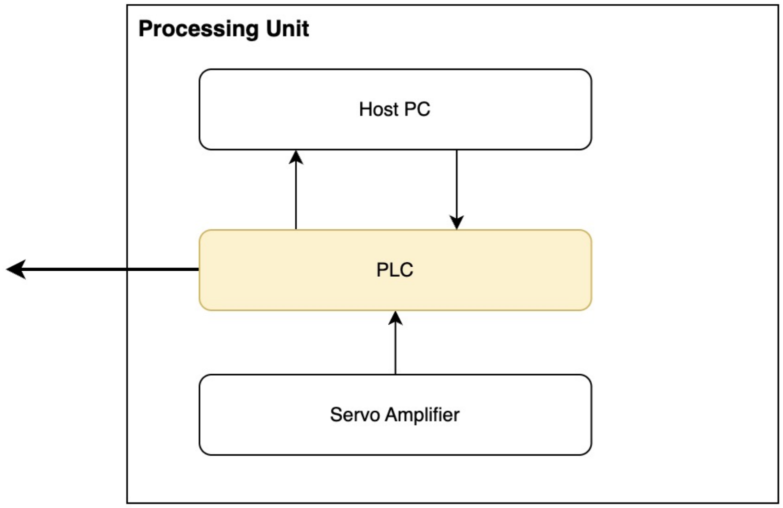

Figure 7: PLC Subsystem

4.6.5 Subsystem Data Processing

The PLC processes sensor feedback and command signals in real-time, executing control logic to coordinate the movements of the robotic arm and additional axis. Communication with the host PC via Ethernet/IP facilitates efficient data exchange for seamless system operation.

5 Robot Arm Layer Subsystems

This section consists of the robot arm, where the actual movement of the robot happens along with the application, which would be the paintball marker associated with it. The robot arm receives commands from the Controller, which gives it instructions on how it can move the joints, linear rail, and the pneumatic line for the trigger to shoot the paintball. These communication

5.1 Layer Hardware

There are multiple hardware components for this layer. This layer consists of the joints that are controlled by the program in the computer. There is also pneumatic portion which controls the air that will flow through the solenoid valves. The system is also integrated with a linear rail which will control the linear movement of the entire robot arm. These systems are connected to the robot controller which controls these movements and enable communication with the controller.

5.2 Layer Operating System

The robot arm uses MELFA Works Operating System.

5.3 Layer Software Dependencies

The software dependencies include the use of RT Toolbox3 software to program the movement of the joints, GX Works software to control the PLC and a USB connection to enable the communication between the robot arm and the controller.

5.4 Subsystem 1

Joints are a hardware component of the robot arm that gives the robot a specific orientation to carry out certain tasks. This will require communication from the robot controller and software, RT Toolbox3, which will give it specific orientation.

5.4.1 Subsystem Hardware

This subsystem has encoders which communicates the speed of joints back to controller, so that they can adjust. They also have automatic braking system embedded into them to quickly brake if E-stops are enabled.

5.4.2 Subsystem Operating System

Joints use the MELFA Works Operating System to control their orientation and movement.

5.4.3 Subsystem Software Dependencies

These joints require the use of RT Toolbox3 for their movement and orientation.

5.4.4 Subsystem Programming Languages

The programs of their movement is written in MELFA BASIC VI.

5.4.5 Subsystem Data Structures

Some data structures utilized by these joints are joint state and trajectory data structure which give information to the joints on where to move.

5.4.6 Subsystem Data Processing

We will be using the joints coordinates given in degrees in order to control their movements. They will be given in a tuple, and we will set boundaries on the limits on where the joints can move.

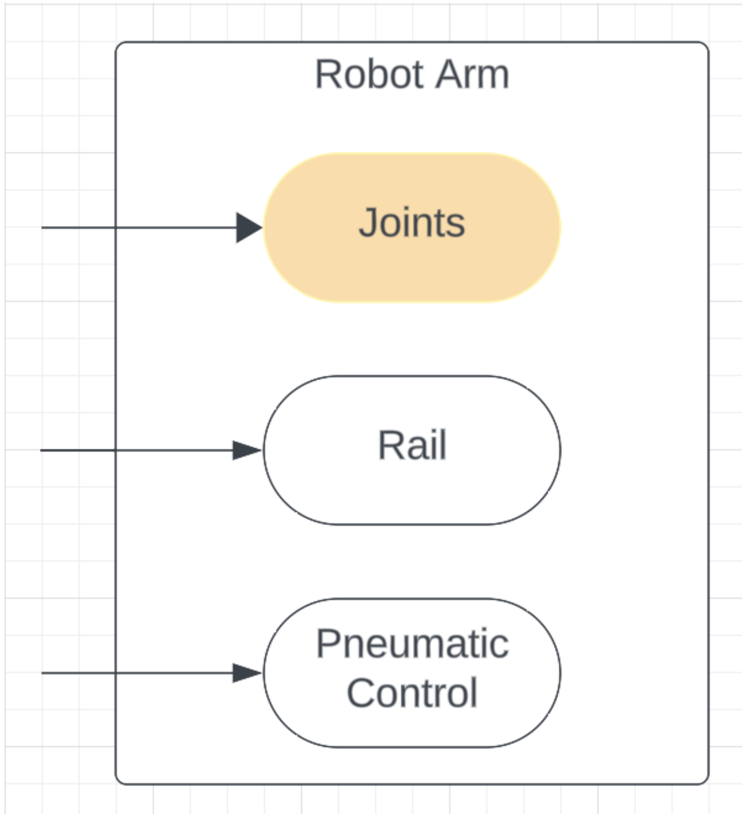

Figure 8: Joints Subsystem

5.5 Rail

Rail is component of the RV8 robot arm which can move all the joints on a linear plane. This gives the robot arm more space to operate and more room in order to perform tasks on a wider plane. The rail receives commands from the servo amplifier which communicates with the robot controller.

5.5.1 Subsystem Hardware

This subsystem has servo motor which gives the rail its motion. The servo motor communicates with the servo amplifier. The servo motor also consists of encoders which controls the speed of the rail. It also consists of a braking system which limits it from movement when the robot arm is powered off.

5.5.2 Subsystem Operating System

Linear rail use the MELFA Works Operating System to control their movement.

5.5.3 Subsystem Software Dependencies

The rail requires RT Toolbox3 to program its movements.

5.5.4 Subsystem Programming Languages

The programs of their movement is written in MELFA BASIC VI.

5.5.5 Subsystem Data Structures

Data structures utilized by these consist of movement data structures, especially those of kinetic and dynamic structures, position, which help enable the movement of the linear rail.

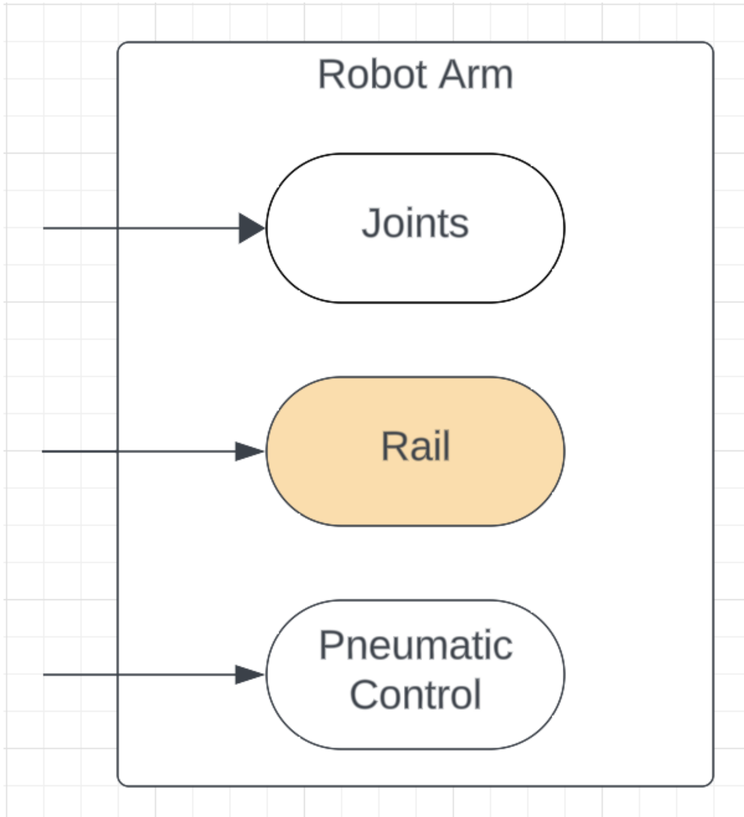

Figure 9: Rail Subsystem

5.5.6 Subsystem Data Processing

For data processing, the linear rail will use have a set origin, and can be configured according to the application. This will include adjusting its movement according the image that will be processed.

5.6 Pneumatic Control

Pneumatic system is connected to the RV8 robot and controls the air pressure that will travel through the robot. The wiring is connected to the PLC which, through the programs, will control the air pressure that will travel through the pipes. The pneumatic control will be connected through solenoid valves which will control the compressed air.

5.6.1 Subsystem Hardware

This subsystem has PLC which will be programmed in order to control the compressed air, and has solenoid valves will be connected to the PLC. This will help the compressed air to trigger the paintball marker.

5.6.2 Subsystem Operating System

Pneumatic Control use the MELFA Works Operating System to control their movement.

5.6.3 Subsystem Software Dependencies

The pneumatic control requires GXWorks3 for programming

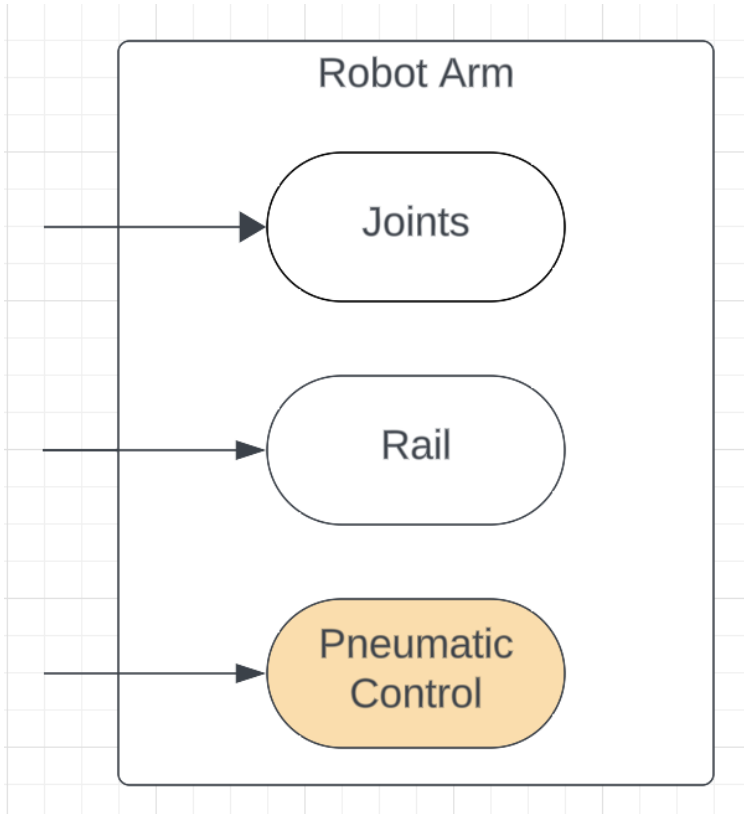

Figure 10: Pneumatic Control Subsystem

5.6.4 Subsystem Programming Languages

The programs of their movement is written in Ladder Logic.

5.6.5 Subsystem Data Structures

Data structures utilized by these consist of valve state data structures, which also consists pressure and flow data structures. These will be used in order to control the compressed air flow.

5.6.6 Subsystem Data Processing

For data processing, the pneumatic control will use to activate and deactivate solenoid valves. There are also going to be sensors that will transmit data regarding the proper functioning of the solenoid valves.

6 Indicator Layer Subsystems

6.1 Indicator Layer Subsystems

The indicator layer will be responsible for the output of the system, where the indicator subsystem consists of an industrial light tower, which displays the status of the working cell.

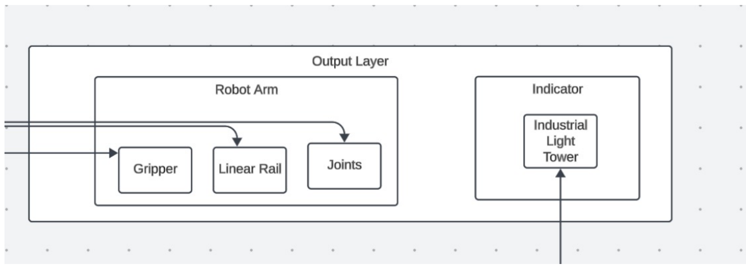

Figure 11: Indicator Layer Subsystem

6.1.1 Industrial Light Tower

An industrial light tower system, typically used in a working cell or industrial setting, serves several essential responsibilities to ensure the efficient and safe operation of the workspace. The specific responsibilities of an industrial light tower system may vary depending on the application. For our system, it serves as an inference to the work environment as it indicates different modes with colored LEDs as get input from the controller and outputs as LEDs.

6.1.2 Subsystem Operating System

No operating system required.

6.1.3 Subsystem Software Dependencies

No software dependencies required by the subsystem.

6.1.4 Subsystem Programming Languages

No programming Language required.

6.1.5 Subsystem Data Structures

No data structures required.

6.1.6 Subsystem Data Processing

No data processing require.

7 Appendix A

REFERENCES