University of Central Florida

Department of Electrical &
Computer Engineering

Senior Design Capstone Project

Industrial Robotic Animatronic

Group 21

Authors:

Austin Berg	Melvin Guzman	Tony Torres	Armando Diaz
Electrical Engineering	*Electrical Engineering*	*Electrical Engineering*	*Computer Engineering*

Sponsor:

Richard Kim, ROBOTIS

Reviewer Committee:

Dr. Mike Borowczak, ECE
Dr. Saleem Sahawneh, ECE
Mark Maddox, ECE

Mentor:

Dr. Lei Wei

Table of Contents:

Chapter 1: Executive Summary 1

Chapter 2: Project Scope 2

2.1 Background & Motivation: 2

2.2 Existing Products & Current Markets:.......................................3

2.3 Engineering Design Requirements, Goals & Objectives:.........................5

2.4 Features and Functionality: 7

2.5 Engineering Specifications: 8

2.6 Prototype Illustration: 9

2.7 High-Level Distribution of Work: 9

2.8 Hardware Diagram:...10

2.9 Software Diagram: 11

2.10 House of Quality: 12

Chapter 3: Research and Investigation 13

3.1 Technology Comparison: 13

3.1.1 Protection Systems and Programmable Logic Controllers 13

3.1.2 Human-Machine Interfaces (HMIs)..14

3.1.3 Microcontroller Units (MCUs), Field Programmable Gate Arrays (FPGAs), & Application-Specific Integrated Circuits (ASICs) 17

3.1.4 Motors & Motor Controller 24

3.1.5 Manufacturing Technologies 26

3.1.6 Sensors...27

3.1.7 Supervisory Control and Data Acquisition; Alarming and Faults.....................30

3.1.8 HMI Software...31

3.1.9 HMI Communication, Serial vs Ethernet, Modbus, Transistor-Transistor Logic, CAN Bus 33

3.1.10 Programming Languages..37

3.1.11 Fuses 41

3.1.12 Electrical Enclosures 41

3.1.13 Ethernet Switch 42

3.1.14 Circuit Breakers...46

3.1.15 PCB...47

3.1.16 Voltage Regulation........49
3.2 Part Comparison:........51
3.2.1 Programmable Logic Controllers (PLCs)........51
3.2.2 Human-Machine Interfaces (HMIs)........54
3.2.3 Microcontroller Unit........57
3.2.4 Motors & Motor Controller........58
3.2.5 Manufacturing Materials........59
3.2.6 Sensors........60
3.2.7 Peripherals........62
3.2.8 Ethernet Port........64
3.2.9 Ethernet Chip........65
3.2.10 Programming IDE........67
3.3.11 Fuses........68
3.2.12 Electrical Enclosures........69
3.2.13 Ethernet Switch........70
3.2.14 Circuit Breakers........71
3.2.15 PCB........72
3.2.16 Voltage Regulators........73
Chapter 4: Standards and Design Constraints........74
4.1 Industrial Standards:........74
4.1.1 Modbus........74
4.1.2 IEC 61131-3 (PLC Programming Standards)........75
4.1.3 Ethernet/IP Communication Protocol........76
4.2 Design Constraints:........77
4.2.1 Cost-Effectiveness........77
4.2.2 Scalability and Modularity........78
4.2.3 Real-Time Performance........79
4.2.4 Time Constraints........80
4.2.5 Environmental Constraints........81
4.2.6 Significant PCB Design........82
4.2.7 Manufacturing........82
4.2.8 Design Practicality........83

Chapter 5: Comparison of ChatGPT with other similar platforms 84

5.1 Introduction:..84

5.2 What is ChatGPT and how does it work?: 84

5.2.1 Pros 85

5.2.2 Cons 86

5.3 Other AI Platforms: 87

5.3.1 Claude 87

5.3.2 Microsoft Copilot..87

5.3.3 Gemini 88

5.3.4 Perplexity 88

5.3.5 Pi 88

5.3.6 AlphaCode 88

5.3.7 Jasper 88

5.4 How ChatGPT will be used in our project:...89

Chapter 6: Hardware Design 92

6.1 W5500 Ethernet Circuit: 92

6.2 Voltage Regulator Circuit: 94

6.3 MCU:...95

6.3.1: Clock 96

6.3.2: Programming and Debugging 96

6.3.3: USB Interface 96

6.3.4: UART and TTL 96

6.3.5: Input/Output Pins and Peripheral Signals 97

6.4 PLC Wiring:...97

6.5 USB-C Programming Port: 98

6.6 Motor Port: 99

6.7 Motor Control and TTL Circuit:..100

6.8 Status LEDs: 100

6.9 Power Requirements:...101

6.10 Existing Part Replacement:...102

Chapter 7: Software Design 104

7.1 C Programming for Robotics: 104
7.2 Motor Tuning: 105
7.3 Motor Control Programming, DYNAMIXEL Wizard 2.0 Configuration and Diagnostic Software: 107
7.3.1 Core Features and Capabilities 107
7.3.2 Integration with Project Workflow 108
7.3.2 Control Table 108
Chapter 8: System Prototype Construction 110
8.1 Animatronic: 110
8.2 Wires, Organization and Raceways: 112
Chapter 9: System Testing and Evaluation 113
9.1 HMI: 114
9.2 PLC: 114
9.2.1: Hardware 114
9.2.1: Software 115
9.3 Dynamixel Motors: 115
9.4 NEMA 17 Stepper Motor & A4988 Driver: 115
9.5 W5500: 116
9.6 SAMD21: 116
Chapter 10: Administrative Content 117
10.1 Budget & BOM: 117
10.2 Milestones: 120
10.3 Full Distribution of Work: 121
Chapter 11: Conclusion 122
Appendix A - Reference 123
Appendix B - Copyright Permission 124

List of Tables:

Table 2.1: Engineering Specifications 8
Table 2.2: Distribution of Work, High-Level 9
Table 3.1: Protection System Technology Comparison 14
Table 3.2: HMI Style Technology Comparison 15
Table 3.3: MCU Technology Comparison 23
Table 3.4: Motor Technology Comparison 25
Table 3.5: Manufacturing Technology Comparisons 27
Table 3.6: Sensor Technology Comparison 29
Table 3.7: Comparison of HMI Software Categories 32
Table 3.8: Technology Comparison, Serial & Ethernet 34
Table 3.9: Technology Comparison, Modbus 35
Table 3.10: Programming Language Technology Comparison 40
Table 3.11: Electrical Enclosure Technology Comparison 42
Table 3.12: Comparison of Ethernet Categories 43
Table 3.13: Ethernet Switch Selection Criteria and Evaluation 44
Table 3.14: Circuit Breaker Comparison 46
Table 3.15: PCB Technology Comparison 47
Table 3.16: PCB Software Comparison 48
Table 3.17: Voltage Regulators Comparison 51
Table 3.18: PLC Part Comparison 54
Table 3.19: Touch-Display HMI CPU Part Comparison 55
Table 3.20: Touch-Display HMI Part Comparison 55
Table 3.21: SAMD21 Part Comparison 57
Table 3.22: Manufacturing Materials Comparison 60
Table 3.23: Peripherals Part Comparison 62
Table 3.24: Peripherals Part Comparison 63
Table 3.25: Ethernet PCB Port Part Comparison 64
Table 3.26: Ethernet IC Part Comparison 66
Table 3.27: Programming IDE Comparison 68
Table 3.28: Electrical Enclosure Part Comparison 69
Table 3.29: Ethernet Switch Part Comparison 70
Table 3.30: Circuit Breaker Part Comparisons 72
Table 3.31: PCB Manufacturing Part Comparisons 72
Table 3.32: Circuit Breaker Part Comparisons 73
Table 5.1: ChatGPT Error Example 86
Table 6.1: Imperial Resistor Case-Code 102

Table 6.2: Metric Resistor Case Code 102
Table 6.3: Resistance Tolerance Values 103
Table 6.4: Capacitance Tolerance Values........ 103
Table 6.5: Parts substitutions made to OpenRB-150's design 103
Table 7.1: PID Tuning Ratios........ 106
Table 7.2: Example Control Table from DYNAMIXEL XC-330-T288-T Datasheet 109
Table 8.1: Animatronic Actuators Axis of Rotation 111
Table 10.1: Preliminary Budget & BOM as of 2.7.25........ 117
Table 10.2: Final BOM with accurate pricing........ 118
Table 10.3: Motors Provisioned by ROBITIS 119
Table 10.4: Total Cost for Product Deliverable 119
Table 10.5: Documentation, Design, and Implementation Milestone Tracker........ 120
Table 10.6: Distribution of Worktable 121

List of Figures:

Figure 2.1: E.T. Animatronic from Universal Studios Japan........3
Figure 2.2: Wolf Man Animatronic from Universal's Epic Universe........4
Figure 2.3: Prototype Diagram 9
Figure 2.4: Hardware Diagram 10
Figure 2.5: Software Diagram........11
Figure 2.6: House of Quality 12
Figure 6.1: W5500 Ethernet Circuit Schematic........92
Figure 6.2: LM2576HV Voltage Regulator Circuit Schematic........94
Figure 6.3: MCU Circuit Schematic 95
Figure 6.4: Proper Wiring Diagram for Micro820-AWB PLC 98
Figure 6.5: USB Program Port Circuit Schematic 99
Figure 6.6: TTL Circuit Schematic........ 100
Figure 6.7: Status LEDs Circuit Schematic........ 101
Figure 6.8: MCU Power Circuit Schematic 101
Figure 7.1: DYNAMIXEL Position PID Logic 106
Figure 8.1: Embedded Dynamixel XC-330-T288-T Motor........ 110
Figure 9.1: Test Bench Setup 113

Chapter 1: Executive Summary

The field of animatronics merges engineering, automation, and entertainment to create lifelike robotic figures capable of intricate movements and interactions. As technology advances, animatronics are becoming more sophisticated, yet accessibility remains limited due to proprietary systems and high costs. This project aims to develop an Industrial Robotic Animatronic that leverages industry-standard Programmable Logic Controllers (PLCs) and Human-Machine Interfaces (HMIs) to provide a cost-effective, scalable, and reliable solution for animatronic applications.

The animatronic system consists of multiple key components working together to achieve synchronized motion and interactivity. The control system, based on PLCs, ensures precise and repeatable movement execution, while the HMI interface provides an intuitive control panel for monitoring and user input. The motor control system integrates high-performance servos and sensors to enhance motion accuracy. A robust fault protection system detects and mitigates errors, ensuring operational reliability and safety.

Current animatronics in the market, primarily dominated by companies such as Disney and Universal, rely on proprietary designs that significantly increase costs and limit customization. This project addresses these challenges by utilizing open-integration methodologies, allowing developers, educators, and independent creators to implement high-quality animatronics without the constraints of expensive proprietary ecosystems.

With the implementation of open-source technologies in animatronics, costs can be reduced through innovation; thus, enabling collaboration within the engineering and maker communities. By designing an adaptable system, this project allows for easy customization and future expansions, ensuring its relevance in both commercial and research applications. The use of standard industrial protocols enhances compatibility with existing automation infrastructure, making it viable for integration into a variety of settings, including theme parks, museums, robotics research, and interactive installations.

The project is structured around three development phases: design and prototyping, integration of hardware and software, and testing for real-world applications. The ultimate goal is to create an animatronic system that not only meets industrial reliability standards but is also adaptable for various applications, including entertainment, education, and research.

This document outlines the design process, including technological selection, component integration, engineering constraints, and prototyping details. By demonstrating that industry-standard automation tools can be effectively applied to animatronics, this project presents a scalable and efficient alternative to existing high-cost solutions.

Chapter 2: Project Scope

In this chapter, we will broadly cover the project scope and goals.

2.1 Background & Motivation:

In the world of modern engineering and entertainment, animatronics stand at the intersection of creativity, technology, and innovation, captivating audiences and pushing the boundaries of what machines can achieve. From their origins in theme parks to their growing presence in museums, theatrical productions, and public displays, animatronics have become a symbol of storytelling brought to life through precision and artistry. These robotic creations, controlled by advanced systems, seamlessly blend lifelike movements with interactivity, creating immersive experiences that inspire wonder and curiosity.

The evolution of animatronics and robotics is rooted in decades of innovation within mechanical and electrical engineering. Early examples of mechanical automation date back to simple automata powered by springs and gears running on pneumatic or hydraulic power, which laid the groundwork for modern robotics. Over time, advancements in control systems, actuators, and sensors enabled the creation of more complex and realistic machines capable of replicating human and animal movements. The integration of microprocessors, digital control systems, and real-time programming revolutionized the field, allowing engineers to precisely choreograph intricate movements and responses. This progress highlights how animatronics has evolved as an engineering discipline, driven by the pursuit of greater precision, efficiency, and adaptability in both hardware and software design.

Our project aims to explore and contribute to this fascinating field by designing and developing an animatronic robot that integrates industry-standard technologies such as Programmable Logic Controllers (PLC) systems and Human-Machine Interfaces (HMI). These tools, widely utilized in industrial automation, offer robust control, precise motion coordination, and highly redundant safety and protection, making them ideal for robotic technology. By leveraging these systems, we can ensure high reliability, efficiency, and safety while enabling complex, synchronized movements and interactive capabilities. We aspire to push the boundaries of animatronic design by utilizing industry-grade tools and methods to create a system that is adaptable, cost-effective, and efficient.

The motivation behind this project stems from the desire to combine engineering expertise with creative expression to build a platform that not only entertains but also educates and inspires. Animatronics provide a unique opportunity to merge all types of engineering into a cohesive system; both technical and artistic.

2.2 Existing Products & Current Markets:

The field of animatronics has long been dominated by large-scale productions developed by industry leaders such as Disney and Universal. These companies have pioneered the use of animatronics to create immersive experiences in theme parks, with highly detailed and lifelike robots that captivate audiences worldwide. The animatronics used in their attractions are known for their sophisticated movements, extensive lifespans, and high reliability. However, these systems come with substantial costs due to proprietary designs, custom-built components, and advanced software tailored exclusively for their specific applications. While these animatronics set the gold standard for performance and realism, their high expense makes them largely inaccessible for smaller-scale productions, independent creators, or educational institutions.

The core of these systems often relies on proprietary or highly customized control systems that are designed in-house or through exclusive partnerships with major industrial automation companies. Brands like Allen-Bradley (Rockwell Automation) and Siemens are frequently involved in providing the Programmable Logic Controllers (PLCs) and related components that power the precise movements and control logic of these animatronics. These PLCs, while renowned for their reliability and advanced capabilities, come at a premium cost due to their robust design, high-performance features, and brand reputation. In addition, the proprietary nature of the control software and hardware integration used in these animatronics often locks buyers into a closed ecosystem, driving up costs further and limiting opportunities for cost-effective customization or expansion.

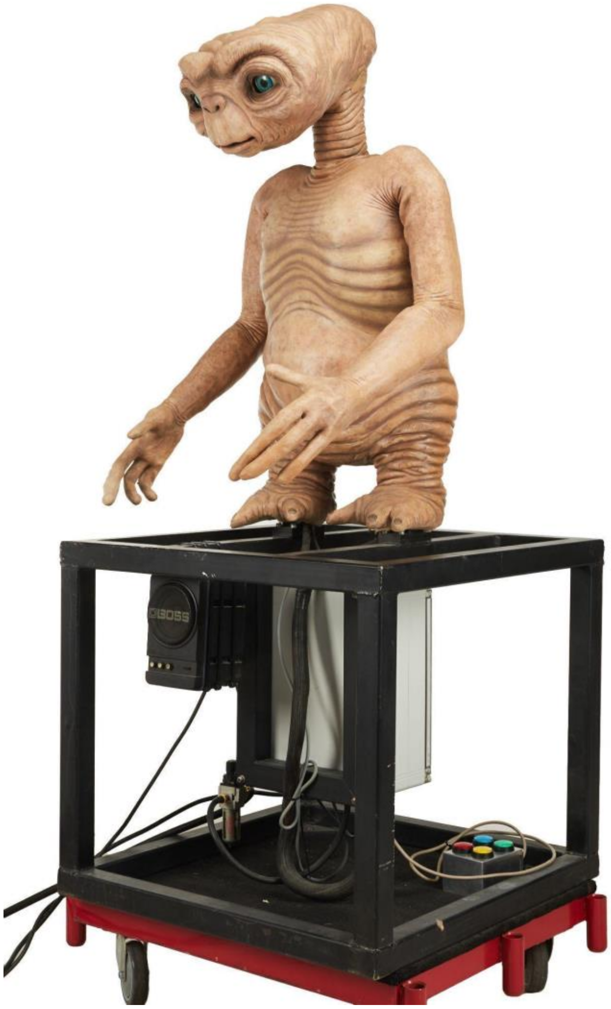


Figure 2.1: *E.T. Animatronic from Universal Studios Japan*
Image provided by Julien's Auctions

The current market for animatronics also includes a small number of companies that develop simpler, lower-cost systems intended for smaller venues, educational use, or hobbyists. However, these systems often lack the precision, reliability, and safety redundancies found in high-end systems, making them unsuitable for professional or demanding applications. This gap in the market leaves many potential users with limited options: either invest heavily in proprietary high-end systems or compromise on quality and reliability with more affordable but less capable alternatives.

The figure below showcases a brand new animatronic, which are said to cost over 1 million dollars per animatronic. These systems have the highest market quality for precision and speed, allowing for hyper realistic and quick movements, which in this case are used to add an element of fear to the ride, blurring the lines between fiction and reality.

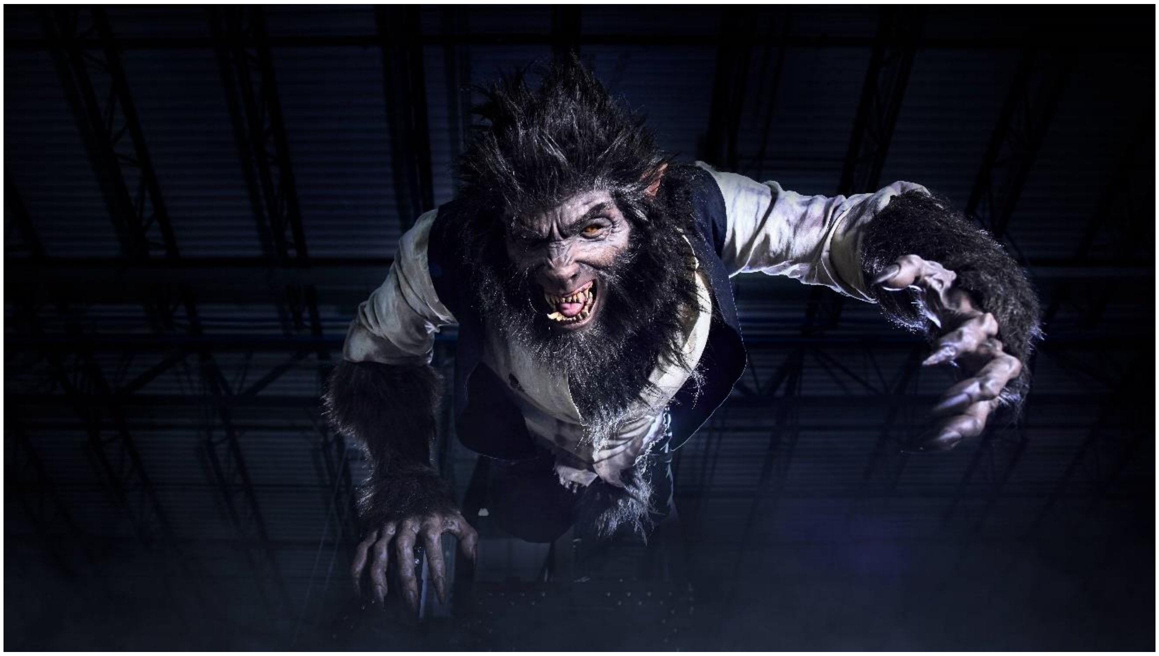


Figure 2.2: *Wolf Man Animatronic from Universal's Epic Universe*
Image provided by WDWMagic

Our project seeks to address this issue by designing an animatronic system that utilizes widely available, industry-standard PLCs and Human-Machine Interfaces (HMIs) while prioritizing cost-effectiveness and reliability. By focusing on open integration and avoiding proprietary constraints, we aim to demonstrate that advanced animatronic capabilities can be achieved without the excessive costs typically associated with current market offerings. This approach could make professional-grade animatronics more accessible to a broader range of applications, from education and research to smaller-scale commercial use.

2.3 Engineering Design Requirements, Goals & Objectives:

The primary focus of this project is to design and build a reliable, cost-effective animatronic system capable of delivering precise, synchronized movements while maintaining safety and efficiency. By leveraging industry-standard technologies such as PLCs and HMIs, the system will integrate mechanical, electrical, and software components into a cohesive unit. Below, the overall goals and objectives of the project are outlined, along with potential stretch goals for further enhancements.

Overall Goals:

- *Develop a Reliable and Cost-Effective Animatronic System:*
  - Design an animatronic system that prioritizes reliability, safety, and efficiency while minimizing production and operational costs.
- *Integrate Industry-Standard Technologies:*
  - Utilize PLCs, HMIs, and other automation technologies to ensure precise, synchronized control of the animatronic system's movements and interactivity.
- *Ensure Scalability and Modular Design:*
  - Create a system architecture that allows for easy customization, expansion, and adaptability to various applications and use cases.
- *Implement Advanced Safety and Fault Protection:*
  - Integrate real-time fault detection, predictive maintenance, and emergency shutdown capabilities to ensure safe and reliable operation in various environments.

Basic Goals:

- Execute a pre-determined control profile onto animation axes on demand.
- HMI can interface and display the status of the control system.
- At least 3 functioning axes of movement.

Advanced Goals:

- HMI can both display and write status to the control system.
- Control integration with fault protection for diagnosis and reset capabilities.
- All planned axes of movement function consistently with >80% up-time.

Stretch Goals:

- Implement Audio / Visual effects like sound and lighting.
- Ability to record and store new show profiles on demand.
- Control system has idle animation effects to run continuously.

Basic Goals:
To achieve the basic goals, the engineering design requirements must focus on establishing a reliable foundation for the animatronic system. Executing a pre-determined control profile onto animation axes requires integrating a Programmable Logic Controller (PLC) to handle precise movements. Industry-standard software, such as Siemens TIA Portal or Allen-Bradley Studio 5000, can be used for programming and optimizing control logic. Additionally, implementing an HMI (Human-Machine Interface) capable of interfacing with the control system and displaying its status will require a user-friendly graphical interface, such as a Siemens Comfort Panel or Rockwell Automation PanelView. Ensuring at least three fully functional axes of movement will involve robust servo motors and drives, such as those from ABB or Schneider Electric, calibrated for smooth and precise operations.

Advanced Goals:
Meeting the advanced goals will require a more sophisticated integration of the control and interface systems. Enhancing the HMI to allow both status display and write capabilities will necessitate bidirectional communication protocols, such as Modbus TCP/IP or Ethernet/IP. This will enable operators to modify system parameters or commands directly from the HMI. Fault protection systems must be integrated to detect, diagnose, and resolve errors efficiently, which can be achieved through advanced diagnostic modules and PLC programming with structured exception handling. To ensure consistent functionality with greater than 80% uptime, the design will include redundant components, such as dual power supplies, and regular system health monitoring using predictive maintenance software like Rockwell Automation's FactoryTalk Analytics.

Stretch Goals:
To address the stretch goals, the system must incorporate additional hardware and software features. Implementing audio/visual effects, such as synchronized lighting and sound, will require integration with audio processors and DMX lighting controllers. Software like QLC+ or Light-O-Rama can be utilized for creating dynamic lighting sequences. The ability to record and store new show profiles on demand will involve incorporating onboard memory or external storage solutions, such as SD cards or USB drives, and updating the PLC logic to handle these custom profiles. Finally, implementing idle animation effects will require continuous motion algorithms programmed into the PLC, enabling smooth transitions and lifelike movements during downtime. This will enhance the system’s realism and keep it engaging when not actively performing a pre-determined sequence.

2.4 Features and Functionality:

Review of similar animatronic systems, such as those found in theme parks or entertainment venues, indicate that intuitive user interfaces, precise control, and safety are paramount. Additionally, customers expressed a desire for robust fault-handling capabilities and the ability to expand functionality without incurring prohibitive costs. Comparisons with existing industry solutions revealed that proprietary designs, while powerful, often lock users into high-cost ecosystems. By addressing these pain points, this project aims to provide a versatile and accessible alternative.

Key Prototype Features:

- Human-Machine Interface (HMI):
    - The system includes a simplified yet powerful HMI panel that allows operators to monitor system status, execute control profiles, and manage animations. The HMI integrates an indicator LED for power/status, a selector switch for mode selection, and an emergency stop button for safety, ensuring ease of use even for non-expert operators.
- Integrated Motor Control:
    - The embedded electric motors, as depicted in the prototype, provide precise control over individual axes of movement. Feedback loops through the breakout board ensure accurate position tracking and smooth operation of each motor.
- Fault Protection and Diagnostics:
    - The control panel includes advanced fault logic for diagnosing and resetting system errors, a critical feature identified as a gap in comparable, low-cost systems. This ensures that the animatronic system operates reliably, even under demanding conditions.
- Expandability:
    - With separate protection and control circuits for motor feedback, the system is designed to accommodate additional axes or functionalities as needed. This modular approach aligns with customer needs for scalability without locking them into proprietary ecosystems.
- Safety Features:
    - The inclusion of an emergency stop button (E-stop) and robust circuit protection ensures operator and equipment safety at all times. This can also allow for features such as Lock-Out Tag-Out (LOTO) capabilities, allowing for safe maintenance in a de-energized state. These features are standard in industrial applications and align with industry regulations.

2.5 Engineering Specifications:

The engineering specifications define critical parameters across the Protection and Control System, Human-Machine Interface (HMI), and Animatronic systems. Among these, three key specifications will be demonstrated: fault response time, operational uptime, and axes functionality.

Table 2.1: *Engineering Specifications*
By Authors

Protection and Controls System	
Maximum Operating Voltage	24V DC
Maximum Operating Current	5A
Fault Response Time	<1 second
Continuous Fault Handling	256 (maximum)
Lock-out Tag-out capabilities	Via E-stop
Human-Machine Interface (HMI)	
Maximum Operating Voltage	12V DC
Maximum Operating Current	1A
Display Type	LCD or OLED
Communication Protocol	Ethernet, Modbus TCP, CAN
Interface Buttons	Selector Switch, Push Button, E-stop
Animatronic	
Maximum Operating Voltage	12V DC
Maximum Operating Current per Axis	2A
Number of Axes	3 (minimum)
Operational Uptime	>80%
Idle Power Consumption	<1W per Axis

The fault response time of less than 1 second is vital for ensuring system safety and reliability, as it showcases the ability of the Protection and Control System to quickly detect and address faults, minimizing damage and downtime. The operational uptime, exceeding 80%, highlights the Animatronic system's capability to maintain consistent functionality, demonstrating its robustness and reliability under normal operating conditions. Lastly, the functionality of multiple axes of either rotation or actuation will effectively demonstrate the capability of the system to function reliably and safely. It can also showcase its vast expandability to accommodate many different robotic features simultaneously while maintaining its overall high operational uptime and fast fault response time.

2.6 Prototype Illustration:

For a wide scope project such as this, it is important to have a general prototype illustration to allow for a clear understanding of exactly what the project scope contains. This can be used for both internal discussion of design and for use in client communications.

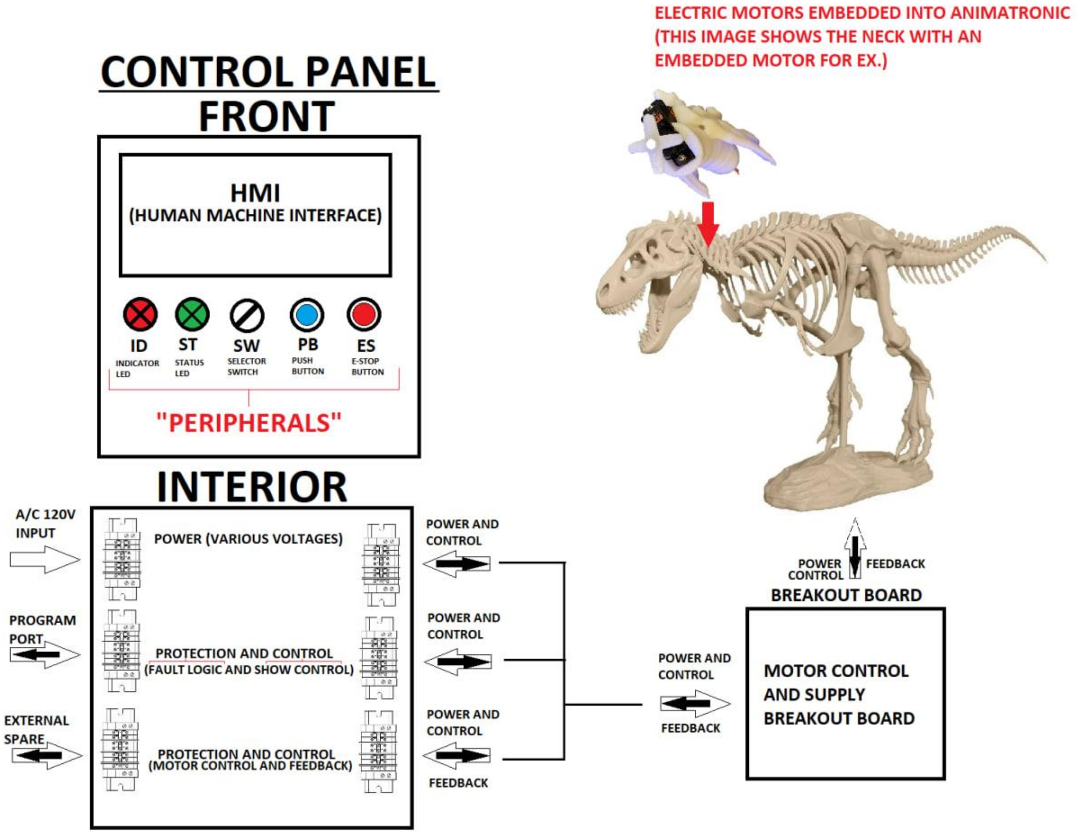


Figure 2.3: *Prototype Diagram*
By Authors

2.7 High-Level Distribution of Work:

Table 2.2: *Distribution of Work, High-Level*
By Authors

Austin Berg EE	Melvin Guzman EE	Tony Torres EE	Armando Diaz CpE
Protection Animatronic Team Lead	Controls Animatronic Motors / Sensors	PCB Design Power Flow Documentation	HMI Networking Peripherals

2.8 Hardware Diagram:

A hardware diagram is a high-level visual representation that outlines the physical components and connections within a robotic or industrial automation system. It typically includes elements such as microcontrollers or PLCs, sensors, actuators, power supplies, allow for a clear understanding of exactly what the project scope contains. This can be used communication interfaces, and any peripheral devices involved in control and monitoring. This diagram serves as a blueprint for understanding how hardware elements interact, how power and signals flow, and how each component contributes to the overall system operation. Hardware diagrams are essential because they provide a clear and organized overview of the system architecture, enabling engineers and technicians to design, troubleshoot, and maintain the system effectively. In the context of robotics and automation, a well-structured hardware diagram ensures compatibility between components, supports efficient system integration, and reduces the risk of wiring errors or misconfigurations during development and deployment.

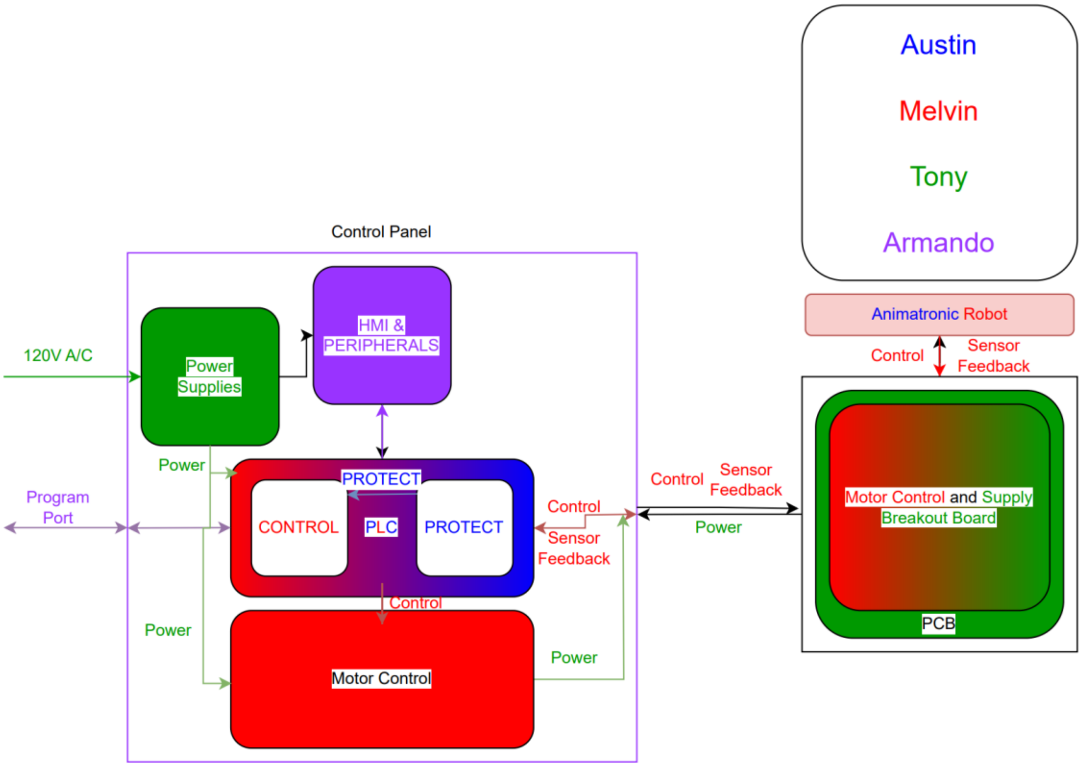


Figure 2.4: *Hardware Diagram*
By Authors

2.9 Software Diagram:

A software diagram is a high-level schematic that illustrates the structure, logic, and flow of software components within a robotic or industrial automation system. It maps out how different software modules, such as control algorithms, communication protocols, user interfaces, and data processing functions, interact with each other and with the hardware. In systems using PLCs, HMIs, or embedded controllers, the software diagram may also include programming environments, runtime behaviors, and task scheduling. This type of diagram is important because it helps developers visualize the overall software architecture, identify dependencies, and ensure smooth integration between software and hardware. For robotics, it supports clear organization of motion control logic, sensor feedback loops, decision-making processes, and user interaction. Ultimately, a software diagram improves system reliability, maintainability, and scalability by promoting structured design and aiding in debugging, updates, and collaborative development.

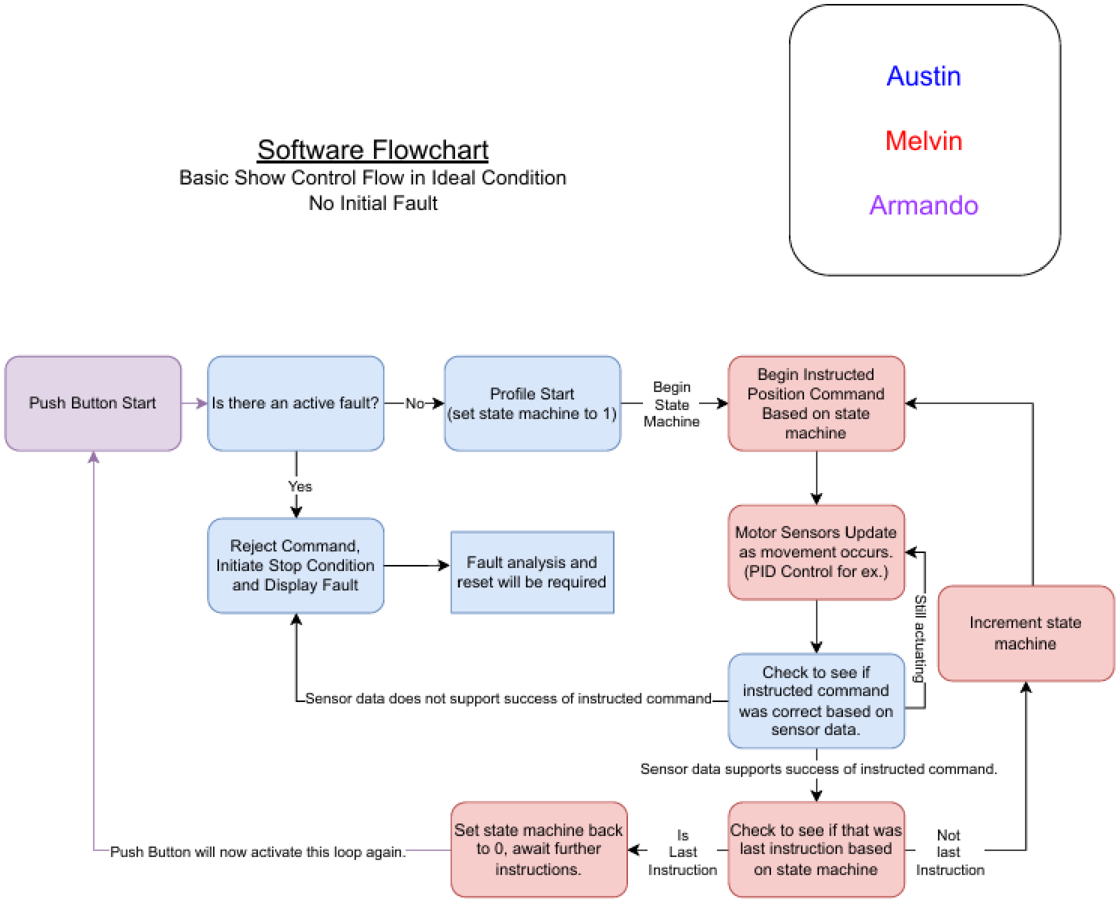
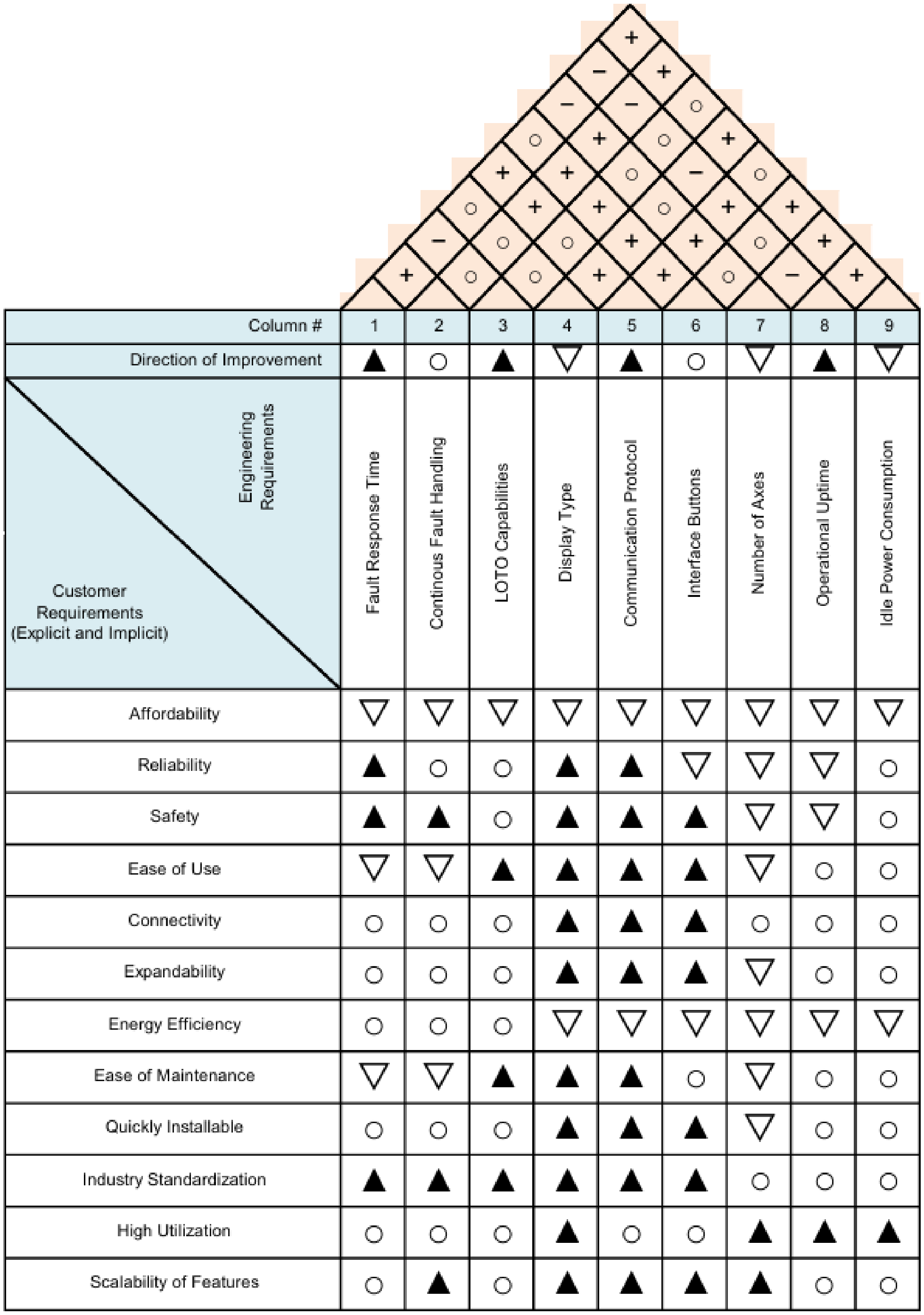

Software Flowchart
Basic Show Control Flow in Ideal Condition
No Initial Fault
Austin
Melvin
Armando
Push Button Start
Is there an active fault?
No
Profile Start
(set state machine to 1)
Begin
State
Machine
Begin Instructed Position Command Based on state machine
Yes
Reject Command, Initiate Stop Condition and Display Fault
Fault analysis and reset will be required
Motor Sensors Update as movement occurs. (PID Control for ex.)
Still actuating
Increment state machine
Check to see if instructed command was correct based on sensor data.
Sensor data does not support success of instructed command
Sensor data supports success of instructed command.
Set state machine back to 0, await further instructions.
Is
Last
Instruction
Check to see if that was last instruction based on state machine
Not
last
Instruction
Push Button will now activate this loop again.

Figure 2.5: *Software Diagram*
By Authors

2.10 House of Quality:

The House of Quality is important as it connects customer needs with technical requirements, ensuring products are designed to meet expectations effectively.



Column #	1	2	3	4	5	6	7	8	9
Direction of Improvement	▲	○	▲	▽	▲	○	▽	▲	▽
Engineering Requirements Customer Requirements (Explicit and Implicit)	Fault Response Time	Continous Fault Handling	LOTO Capabilities	Display Type	Communication Protocol	Interface Buttons	Number of Axes	Operational Uptime	Idle Power Consumption
Affordability	▽	▽	▽	▽	▽	▽	▽	▽	▽
Reliability	▲	○	○	▲	▲	▽	▽	▽	○
Safety	▲	▲	○	▲	▲	▲	▽	▽	○
Ease of Use	▽	▽	▲	▲	▲	▲	▽	○	○
Connectivity	○	○	○	▲	▲	▲	○	○	○
Expandability	○	○	○	▲	▲	▲	▽	○	○
Energy Efficiency	○	○	○	▽	▽	▽	▽	▽	▽
Ease of Maintenance	▽	▽	▲	▲	▲	○	▽	○	○
Quickly Installable	○	○	○	▲	▲	▲	▽	○	○
Industry Standardization	▲	▲	▲	▲	▲	▲	○	○	○
High Utilization	○	○	○	▲	○	○	▲	▲	▲
Scalability of Features	○	▲	○	▲	▲	▲	▲	○	○

***Figure 2.6**: House of Quality*
By Authors

Chapter 3: Research and Investigation

In this chapter, we will conduct a thorough analysis of the technologies and components used in robotic and industrial automation systems. The investigation begins with a technology comparison, where we evaluate different programming languages, control systems, and automation platforms to understand their suitability for real-time motor control, fault handling, and integration with industrial networks. Following this, we perform a part comparison, examining various hardware elements such as sensors, actuators, controllers, and communication modules. By comparing the performance, compatibility, and application of these technologies and components, we aim to identify the most effective solutions for building a reliable and efficient robotic control system. This research provides the foundation for informed design decisions in later stages of development.

3.1 Technology Comparison:

This section presents a comparison of various technologies that could be utilized for the project's key components. It explores potential solutions from both hardware and software perspectives, evaluating their advantages, limitations, and suitability for the intended application. These insights will then be used to guide the selection of the most appropriate technologies, ensuring optimal performance, reliability, and integration within the overall system design.

3.1.1 Protection Systems and Programmable Logic Controllers

Protection systems in industrial control environments are essential to ensure the safe and reliable operation of electrical and mechanical systems. These protections extend to programmable logic controllers (PLCs), safety relaying mechanisms, and other industrial control cabinet safeguards. PLCs serve as the core of automation and control systems, orchestrating inputs and outputs to manage industrial processes efficiently. Safety relays are crucial for implementing emergency stop mechanisms, preventing system failures from escalating into hazardous conditions. Industrial control cabinets incorporate a variety of protective measures, including circuit breakers, surge protectors, and thermal overload relays, to shield sensitive components from electrical faults. The integration of fault detection and predictive maintenance within these systems enhances operational longevity, reducing downtime and potential failures. Moreover, industrial enclosures are designed to comply with standards such as NEMA and IP ratings, ensuring environmental protection against dust, moisture, and temperature fluctuations.

Table 3.1: *Protection System Technology Comparison*
By Authors

Component	Function	Advantages	Limitations
Programmable Logic Controller (PLC)	Executes control logic, manages I/O, and handles automation processes	Modular, reliable, real-time capable, scalable	Not safety-rated unless certified (e.g. failsafe PLCs); higher cost
Failsafe PLC (Safety PLC)	Safety-certified PLC that handles both automation and safety logic	Certified to SIL/PL standards, supports advanced safety control	More expensive and complex to program; requires safety-rated inputs/outputs
Safety Relay	Dedicated logic device for safety circuits (e.g. E-stop, guard doors)	Simple, highly reliable, deterministic shutdown response	Limited logic capability; fixed functionality

As previously stated, PLCs serve as the central decision-making unit in most automated systems. They receive input from sensors, execute logic, and control outputs; all while being able to handle fault detection, diagnostics, and fail-safe routines. Safety relays operate parallel, ensuring that if a safety condition is triggered (e.g. an emergency stop or door open), the system reacts instantly and predictably. These mechanisms are backed up with standardized wiring practices, redundancy, and system diagnostics

3.1.2 Human-Machine Interfaces (HMIs)

Our intention with the control panel is to be in-line with common industry practices and have a design in which that is intuitive to use. So, for the control panel of our project, one of our requirements was to implement an interface that would allow a user to manage and operate the animatronic. For this interface, our options were abstract in the sense of implementation. These included remote devices, web-based interfaces, and HMIs.

Consideration of HMIs and other interface technologies:

- Remote Devices
    - Access from a distance
- Web Browser
    - Accessibility for different devices
    - Scalability for lower cost
- HMIs
    - Industry Standard
    - Direct Control
    - Response Time

For our design, an interface must be used for a user to be able to operate the system. For this interface, there are certain qualities that must be present. The interface, above all else, must be responsive and in-line current industry practices. As our main peripheral to the system is an animatronic, it is important that the system be able to operate and halt any actions in a timely manner; Moreso in the event of an emergency, the e-stop must be able to immediately stop the animatronic to ensure the safety of persons nearby. Other qualities that are to be considered for this interface are the ease of implementation, cost associated, and scalability to be consistent with the goals that we have established.

Selecting a suitable HMI technology involves more than identifying a single piece of hardware; it is a determination of the foundational mode by which users will interact with the system. The HMI serves as the operator's window into the functionality and behavior of the system and thus must be chosen with care. In the context of industrial automation, several technologies are commonly employed to facilitate this interaction. From the broad landscape of available methods, three interface technologies were selected for evaluation: a Touch-Display panel, Remote Devices, and a Web Browser Application. Each of these offers unique characteristics that merit consideration.

Table 3.2: *HMI Style Technology Comparison*
By Authors

	Responsiveness	Scalability	Mobility	Implementation	Cost
Touch-Display	+	+	-	+	+
Remote Devices	-	+	+	-	-
Web Browser	-	+	+	-	-

When deciding from the listed options, it was important to understand how each would be implemented and how it would affect the use of the system. A Web Browser Application requires the development and deployment of the application to a server, enabling users to interact with the system from common devices like phones or tablets. This wireless connectivity allows users to multitask while operating the system, and the ability to access it from any internet-connected device reduces the need for proprietary hardware, promoting scalability and cost efficiency. However, relying on a wireless connection for the main interface introduces significant concerns. For instance, the interface can become unresponsive due to internet traffic, especially when multiple users and systems are connected simultaneously, leading to delays that could hinder system performance. Additionally, if the internet connection is lost, the system becomes inoperable, potentially causing undesirable outcomes.

Despite its apparent advantages, a Web Browser Application may prove more costly upon closer examination. Since the application must be hosted on a server, ongoing monthly hosting fees are incurred, even after the system's development is complete. These recurring costs, combined with the potential for poor performance due to connectivity issues, make a Web Browser Application less desirable for our Human-Machine Interface (HMI) implementation. The need for reliable performance and cost-effectiveness outweighs the convenience of wireless accessibility, leading us to consider alternative solutions that better align with our system requirements.

Similar reasons can be seen with Remote Device implementation. The main difference with using Remote Devices or a Web Browser Application is the use of a local system and maybe the use of some proprietary devices. Yet these differences do not improve on the faults of having a Web Browser Application. While there may be no server to maintain, having a wireless connection may have an unreliable response time, due to external factors outside of the system's design, and ensuring responsiveness is key to the design. As for the devices that may be used in tandem with the system, it is open as to whether common devices, such as phones and tablets, can be used or a proprietary device one instead. Regardless of which, those devices would be tied with the particular system it is controlling, essentially needing a separate set of devices for each instance of the system that is used. This brings in an unnecessary cost of needing to bring several external devices. This option is more desirable than having a Web Browser Application, however it lacks the responsiveness that is needed.

This leaves the final option of using a Touch-Display. What makes using a Touch-Display different from using either of the previous implementations is the fact that the Display will be wired into the system directly. This lacks the mobility from the other options as the user will have to be where the main control box is stationed, however being a direct connection promotes the level of responsiveness desired for the system's design. Since the design itself would determine the bottleneck of the system, we would be able to determine the minimum speed the system would operate at and select parts accordingly. This combined with a physical interface such as switches and buttons would give a proper emergency shutoff mechanism that could effectively power down the animatronic as the user would be always by the control box. Having this sort of design would also limit several external factors that are associated with a wireless connection as there would be no communication traffic or signal interference, as well as being able to easily troubleshoot where certain connection issues would be occurring. This would also remove the cost of having additional devices as only one Touch-Display would be connected. For these reasons, we have decided to use a Touch-Display for our design.

3.1.3 Microcontroller Units (MCUs), Field Programmable Gate Arrays (FPGAs), & Application-Specific Integrated Circuits (ASICs)

In embedded system design, selecting the right processing architecture is crucial for balancing performance, flexibility, power efficiency, and cost. The three primary categories of processing units used in automation and control applications are Microcontrollers (MCUs), Field-Programmable Gate Arrays (FPGAs), and Application-Specific Integrated Circuits (ASICs). Each of these technologies serves a different purpose, with trade-offs in terms of customizability, processing power, and energy efficiency.

Microcontrollers (MCUs)

Microcontrollers are compact, self-contained computing units designed for embedded control applications. They typically include a central processing unit (CPU), memory, and peripherals such as timers, communication interfaces, and analog-to-digital converters (ADCs).

MCUs are widely used due to their ease of programming, affordability, and power efficiency. They operate on pre-written firmware, executing specific tasks such as sensor data processing, motor control, and real-time automation. Because MCUs have fixed hardware configurations, they are optimized for specific applications, ensuring low power consumption and deterministic behavior.

Advantages of MCUs:

- Low power consumption, making them ideal for battery-operated devices.
- Cost-effective for small to medium-scale applications.
- Ease of use, with widely available development tools and software support.
- Real-time operation, suitable for tasks that require precise timing control.

Disadvantages of MCUs:

- Limited processing power, making them unsuitable for complex signal processing or high-speed computation.
- Less flexibility, as hardware functions are predetermined and cannot be reconfigured dynamically.
- MCUs are best suited for applications such as industrial automation, robotics, IoT devices, and animatronics, where deterministic control and power efficiency are key concerns.

Field-Programmable Gate Arrays (FPGAs)

FPGAs are reconfigurable digital circuits that allow engineers to define custom hardware logic using Hardware Description Languages (HDLs) such as Verilog or

VHDL. Unlike MCUs, which run sequential code, FPGAs process multiple tasks in parallel, making them significantly faster for certain applications.

FPGAs excel in high-speed data processing, complex real-time control, and AI acceleration. Their programmable nature allows for dynamic hardware modifications, making them ideal for prototyping, signal processing, and specialized automation systems.

Advantages of FPGAs:

- High performance, enabling parallel processing for fast data handling.
- Hardware reconfigurability, allowing design changes without manufacturing new chips.
- Deterministic timing, essential for precise control applications.
- Disadvantages of FPGAs:
- Higher power consumption compared to MCUs, making them unsuitable for energy-sensitive applications.
- Complex programming requirements, requiring expertise in HDL coding and FPGA toolchains.
- Higher cost, especially for small-scale projects.

FPGAs are commonly used in high-performance robotics, aerospace control systems, and digital signal processing applications where speed and flexibility outweigh power constraints.

Application-Specific Integrated Circuits (ASICs)

ASICs are custom-designed chips created for a specific application or function. Unlike MCUs and FPGAs, which can be reprogrammed, ASICs have fixed hardware logic, making them highly optimized for performance and efficiency.

ASICs are the most power-efficient and high-performance solution, but they require extensive design and manufacturing costs, making them impractical for small-scale production. They are primarily used in consumer electronics, high-frequency trading systems, and AI accelerators, where large-scale production justifies the upfront cost of chip fabrication.

Advantages of ASICs:

- Highest performance and power efficiency for dedicated tasks.
- Optimized hardware, resulting in lower operational costs at scale.

Disadvantages of ASICs:

- High development cost, making them feasible only for mass production.
- Lack of flexibility, as changes require an entirely new chip design.

For the purpose of this project, we will be using an MCU as its tradeoffs are suitable for this project, and will therefore only compare MCU technologies:

For this animatronics project, microcontrollers serve as the primary control units, handling sensor input, motor actuation, communication, and real-time processing. The system relies on MCUs to process movement commands, synchronize actuators, and interface with external peripherals such as sensors and communication modules. Given the complexity of animatronic motion, the selected microcontrollers must provide precise timing control, efficient power management, and multiple input/output capabilities to ensure smooth and responsive operation.

Each MCU in the system is assigned specific roles based on its strengths. Some units handle low-level motor control, receiving position and speed commands while processing real-time sensor feedback to maintain smooth movements. Others are responsible for high-level logic processing, managing interactions between different subsystems and communicating with external devices such as PLCs and HMIs. Wireless connectivity may also be incorporated for remote monitoring and control, making certain MCUs with Wi-Fi or Bluetooth capabilities an integral part of the design.

Incorporating microcontrollers instead of FPGAs or ASICs provides the necessary flexibility, cost-effectiveness, and ease of programming needed for a modular and scalable animatronic system. The ability to program and reconfigure microcontrollers enables rapid prototyping and iterative development, allowing adjustments to movement profiles, sensor integrations, and communication protocols as the project evolves. Through careful selection and distribution of MCU responsibilities, the system ensures a balance between real-time control, efficiency, and expandability, making it well-suited for industrial animatronic applications.

3.1.3.1 Arduino Uno

The Arduino Uno is one of the most widely recognized microcontrollers in the world of embedded development. Based on the ATmega328P microcontroller, it is known for its simplicity, ease of programming, and strong open-source community support. The board features 14 digital input/output pins, 6 analog inputs, and operates at 16 MHz, making it suitable for projects that require basic sensor interfacing, motor control, and simple automation tasks.

One of the key advantages of the Arduino Uno is its accessibility for both beginners and experienced developers. The Arduino Integrated Development Environment (IDE) provides a user-friendly platform for writing and uploading code using the C++-based Arduino language. Its compatibility with a vast library of pre-written code allows for rapid prototyping and integration with numerous sensors, actuators, and communication modules.

However, the Arduino Uno has limitations in terms of processing power and memory. With only 32 KB of flash storage, 2 KB of RAM, and a single-core architecture, it is not well-suited for tasks that require complex computations, real-time processing, or network connectivity. Additionally, its lack of built-in Wi-Fi or Bluetooth restricts its use in IoT and remote-controlled applications. Despite these constraints, the Arduino Uno remains a highly effective choice for basic embedded systems and rapid prototyping.

3.1.3.2 Arduino Mega

The Arduino Mega 2560 builds upon the capabilities of the Uno, offering expanded input/output capabilities and increased memory, making it ideal for applications requiring multiple sensor inputs, actuators, or complex automation sequences. At its core, the Mega is powered by the ATmega2560 microcontroller, which operates at 16 MHz but provides significantly more resources, including 256 KB of flash storage, 8 KB of RAM, and 4 KB of EEPROM.

A defining feature of the Arduino Mega is its 54 digital I/O pins and 16 analog inputs, allowing it to handle a large number of peripherals simultaneously. This makes it particularly well-suited for animatronics, robotics, and industrial control systems where multiple motors, sensors, and communication interfaces need to be managed.

Despite its expanded capabilities, Mega shares some of the same limitations as the Uno, such as the lack of native network connectivity and limited processing power compared to more advanced microcontrollers. However, its compatibility with Arduino shields and libraries makes it an attractive option for developers who require additional I/O while maintaining ease of use and flexibility.

3.1.3.3 Raspberry Pi

Unlike traditional microcontrollers, the Raspberry Pi family provides a single-board computer (SBC) architecture, enabling it to run full-fledged operating systems like Linux-based Raspberry Pi OS. This significantly expands its capabilities beyond those of standard MCUs, allowing for tasks such as image processing, real-time networking, and advanced automation.

The Raspberry Pi 4 is among the most powerful models in the series, featuring a quad-core ARM Cortex-A72 processor, up to 8 GB of RAM, and multiple connectivity options, including Wi-Fi, Bluetooth, and Ethernet. This makes it an excellent choice for applications that require high computational power, such as machine learning, real-time video processing, or remote-controlled robotics. The Raspberry Pi Zero, a smaller and more power-efficient variant, provides similar capabilities in a compact form factor, making it suitable for embedded applications where space and energy consumption are concerned.

While Raspberry Pi boards offer exceptional processing power, they have limitations in applications requiring hard real-time processing. Unlike microcontrollers, which operate with deterministic timing, Raspberry Pi devices rely on an operating system, introducing latency and unpredictability in time-sensitive applications. However, for animatronics projects where data processing, multimedia integration, or network control are required, Raspberry Pi remains a powerful tool.

3.1.3.4 SAMD21

The SAMD21 is a 32-bit ARM Cortex-M0+ microcontroller, known for its low power consumption, high efficiency, and flexible peripheral options. This microcontroller is commonly found in the Arduino MKR series and offers a substantial upgrade over the ATmega-based Arduino boards in terms of processing capability.

Operating at 48 MHz, the SAMD21 provides 256 KB of flash storage and 32 KB of RAM, allowing for more complex applications than traditional 8-bit microcontrollers. It features multiple communication interfaces, including I2C, SPI, and UART, as well as a built-in USB interface, which simplifies integration with computer-based applications. Additionally, it includes advanced power management features, making it an excellent choice for battery-powered and energy-efficient designs.

Compared to the Arduino Uno and Mega, the SAMD21 offers better computational performance and peripheral flexibility, making it well-suited for projects requiring real-time processing, precise timing control, and low-power operation. While it lacks built-in network connectivity like the ESP32, it remains a strong contender for embedded control applications that need more processing power than traditional 8-bit microcontrollers.

3.1.3.5 ESP32

The ESP32 is a dual-core 32-bit microcontroller developed by Espressif Systems, widely known for its integrated Wi-Fi and Bluetooth connectivity. The ESP32 operates at clock speeds of up to 240 MHz and provides up to 520 KB of RAM, making it significantly more powerful than typical 8-bit microcontrollers.

One of the primary advantages of the ESP32 is its built-in wireless capabilities, allowing seamless integration with cloud-based platforms, wireless sensor networks, and real-time communication systems. Additionally, it includes hardware acceleration for encryption, making it suitable for secure data transmission.

Despite its high performance and network capabilities, the ESP32 consumes more power than low-energy MCUs like the SAMD21 or CC26xx series, which may be a limitation for battery-operated applications. However, for projects requiring wireless communication and computational efficiency, the ESP32 provides a well-balanced solution.

3.1.3.6 CC26xx Series

The CC26xx series from Texas Instruments is a family of ultra-low-power wireless microcontrollers, primarily designed for Bluetooth Low Energy (BLE) and Zigbee applications. These MCUs are optimized for minimal energy consumption, making them ideal for battery-powered devices and long-term sensor deployments.

With a 32-bit ARM Cortex-M3 processor, the CC26xx series offers sufficient computational power for low-power wireless applications while maintaining efficient energy usage. It is commonly used in smart home automation, medical devices, and industrial IoT systems, where prolonged battery life is a priority.

While the CC26xx series excels in wireless communication and low power consumption, it lacks the raw processing power of high-performance microcontrollers like the ESP32 or Raspberry Pi. However, for applications that require long-lasting, energy-efficient wireless communication, it remains a strong candidate.

3.1.3.7 MSP430 Series

The MSP430 series from Texas Instruments is a family of ultra-low-power 16-bit microcontrollers, specifically designed for applications where energy efficiency is a top priority. Unlike traditional 8-bit MCUs, the MSP430 operates with a 16-bit Reduced Instruction Set Computing (RISC) architecture, allowing for improved computational efficiency while maintaining low power consumption. These microcontrollers are widely used in battery-operated devices, industrial automation, and remote sensing applications, where power availability is limited, and long-term operation is required.

One of the standout features of the MSP430 series is its exceptionally low power consumption. These MCUs offer multiple low-power modes, allowing devices to enter deep sleep states and consume only microamps of current when idle.

This efficiency makes them ideal for wearable technology, medical devices, and wireless sensor networks that must operate for extended periods on a single charge. Additionally, the ability to wake from sleep mode in microseconds ensures that the system remains responsive while still maintaining energy efficiency.

The MSP430 series also includes a rich set of integrated peripherals, reducing the need for additional external components. Most variants come equipped with analog-to-digital converters (ADCs), digital-to-analog converters (DACs), communication interfaces such as UART, SPI, and I2C, as well as multiple timers. These features make the MSP430 suitable for real-time signal processing applications and embedded systems that require precise data acquisition and control. Some models also feature capacitive touch support, allowing them to be used in human-machine interfaces (HMIs) and touch-sensitive control panels.

Despite its many advantages, the MSP430 has some limitations. While its low-power architecture is beneficial for energy-sensitive applications, it comes at the cost of lower processing power compared to higher-end microcontrollers such as ARM Cortex-based devices. The 16-bit architecture, while more powerful than 8-bit MCUs, does not offer the same level of performance as 32-bit counterparts, making it less suitable for intensive computational tasks or high-speed real-time processing. Additionally, while Texas Instruments provides robust development tools and libraries, the learning curve can be steeper compared to more widely used platforms like Arduino, which benefits from an extensive open-source community and simpler development environments.

Overall, the MSP430 series is a highly efficient and reliable choice for applications that prioritize long battery life, low power consumption, and precise analog processing. While it may not be the best fit for high-performance automation tasks, it excels in sensor networks, industrial monitoring, medical instrumentation, and other embedded applications where energy efficiency is the primary concern.

3.1.3.7 MCU Summary

Below is a summarized table comparison, with our selection of MCU technology highlighted.

Table 3.3: *MCU Technology Comparison*
By Authors

MCU	Architecture	Clock Speed	RAM	Flash	I/O Pins	Connectivity
Arduino Uno	8-bit AVR	16 MHz	2 KB	32 KB	20 GPIO	USB, Serial
Arduino Mega	8-bit AVR	16 MHz	8 KB	256 KB	70 GPIO	USB, Serial
Raspberry Pi	64-bit ARM	Up to 1.5 GHz	Up to 8 GB	microSD	40 GPIO	Wi-Fi, Bluetooth, Ethernet
SAMD21	32-bit ARM	48 MHz	32 KB	256 KB	Varies (20-64 GPIO)	USB, I2C, SPI, UART
ESP32	32-bit Xtensa	Up to 240 MHz	520 KB	448 KB+	34 GPIO	Wi-Fi, Bluetooth
CC26xx Series	32-bit ARM	48 MHz	20 KB+	128 KB+	30 GPIO	BLE, Zigbee
MSP430 Series	16-bit RISC	Up to 25 MHz	0.5-4 KB	32 KB	Varies (16-32 GPIO)	UART, SPI, I2C

3.1.4 Motors & Motor Controller

The animatronic can be controlled and moved using electric motors. Each motor that performs a specific movement of the figure is referred to as a function, or axis, and are identified by a function name and number. These axes should all be individually coordinated to execute specific motions that, when played back all together at once, produce a desired motion profile. To deliver the creative intent desired, the motors should meet the following requirements:

- The figure should be able to produce believable, life-like motion.
- The motors should be able to stop upon command from the motor controller.
- The motors should be able to stop upon command from an external source.
- The motors should be able to home.

Selection of a motor considered several hard-set requirements and variables to reach a final decision. Factors included:

- Motor type
- Input voltage and current specifications
- Torque output
- Size
- Programmable logic controller integration
- Parameter customization
- Weight

Stepper motors and servo motors were the two motor types considered. A stepper motor moves in fixed angle increments by using electromagnets around the central gear. While this is useful for the animation profile, stepper motors lack fine control when at low top speeds while moving a heavier load. Each function will experience a different weighted loads from each other, therefore stepper motors may not be the best choice for most of the figure’s functions.

Servo motors use a DC motor for continuous rotation. The servo motor’s position is controlled via pulse width modulation (PWM). Using a built-in control circuit, a servo motor will move towards a commanded position depending on the width of the pulse and hold torque as the power stays on. Servo motors allow for finer control of position and velocity, making it a great choice for the project.

The input voltage and current of the motor chosen matters for the overall project goals. These values determine the maximum amount of power that the motors will be able to produce. To maintain simplicity of the power circuits, the input voltage should be kept as streamlined as possible in order to reduce the amount of control voltages needed to run the figure. Typical control voltages exist as 3.3V, 5V, 9V, 12V, and 24V. To emulate industry standards, the controls system of this project should predominantly consist of 12V and 24V components. In addition, having the right current requirement on the motors will mean that the motors can generate the necessary torque required to move a load while avoiding any issues such as overheating or instability.

The torque output is the measure of the rotational force a motor can produce. In terms of this project, the motor should be able to overcome several weighing loads present on the figure. Different functions or axes in the figure will experience different weight and force requirements, meaning that a motor must deliver sufficient torque to move the functions with heavier loads while maintaining consistent performance.

The motor size is important as it will affect the torque output of the motor as well as the overall look and design of the figure. A balance needs to be found with the motor size ass an undersized motor will be easy to incorporate into the figure but not produce enough torque, and an oversized motor will have no issue moving the heavier loads but will require additional designs to fit the figure. The weight of the motor is also important as some motor(s) will be driving a load that consists of other motor(s).

Table 3.4: *Motor Technology Comparison*
By Authors

Motor Type	**Precision / Control**	**Torque**	**Pros**	**Cons**
Stepper Motor	Moderate (good at low speed, may skip steps under load)	Low to moderate torque, varies with model	Inexpensive, easy to control, no feedback required	Loses torque at high speeds, prone to stalling, inefficient
Standard Servo Motor	High (accurate position hold, good for smooth motion)	Moderate torque with holding power	Great fine control, maintains torque when stationary	Typically, small scale, limited rotation range (unless modified)
Closed-Loop Servo Motor	Very high (real-time feedback with PID adjustment)	High torque, scalable depending on motor size	Real-time error correction, powerful, customizable PID tuning	Higher cost, more complex wiring and setup

Finally, the motors should have the ability to customize the way that they operate. Specifically, variables such as acceleration, velocity, proportional, integral, and derivate control tuning such that the proper tuning of these settings allow for the desired show playback. This project heavily utilizes the concept of proportional, integral, and derivative control (PID) to achieve the desired animation playback profile that will be used for the animatronic. Each motor utilized in the figure will have varying loads from function to function, therefore requiring different motor tuning that is specific to each motor. PID tuning essentially provides finer control over the motor by modifying the behavior of the motor when it is reaching closer to its goal position. As all control systems, a motor will have critical undershoot, undershoot, overshoot, and critical overshoot. In the case of a motor, over and under shooting will mean how much the motor may miss its programmed goal position. The motor will then reach a steady state once it settles into position, but the final position the motor takes on is determined by the steady state error.

3.1.5 Manufacturing Technologies

Manufacturing technologies play a pivotal role in the prototyping and production phases of industrial systems. Among the most prominent methods are 3D printing, injection molding, and other rapid prototyping techniques that allow for iterative design improvements and cost-effective production.

3D printing has revolutionized prototyping by enabling the rapid creation of complex geometries with minimal material waste. Technologies such as Fused Deposition Modeling (FDM) and Stereolithography (SLA) are commonly employed in the development of custom enclosures, brackets, and intricate mechanical components. The flexibility of additive manufacturing allows for rapid design iterations, making it an ideal choice for early-stage product development.

Injection molding remains the gold standard for mass production, offering high precision and repeatability for plastic components. While the initial tooling costs are substantial, the per-unit production cost decreases significantly with scale, making it the preferred choice for high-volume manufacturing. Material selection is crucial in injection molding, with polymers such as ABS, polycarbonate, and nylon providing varied mechanical and thermal properties suited for different application environments.

Other rapid prototyping techniques, such as CNC machining and vacuum casting, are often employed for producing durable prototypes and low-volume production runs. CNC machining provides high precision and is widely used for metal and plastic components requiring tight tolerances. Vacuum casting, on the other hand, is suitable for producing small batches of functional prototypes using silicone molds, bridging the gap between prototyping and full-scale manufacturing.

Table 3.5: *Manufacturing Technology Comparisons*
By Authors

Technology	Description	Pros	Cons
3D Printing (FDM/SLA)	Additive manufacturing using plastic filament (FDM) or resin (SLA)	Fast iteration, low waste, complex geometries, minimal tooling required	Lower strength than molded/metal parts, slower for large quantities
Injection Molding	Injecting molten plastic into a hardened steel mold	High precision, repeatability, low cost per part at scale	High upfront tooling cost, less flexible for quick design changes
CNC Machining	Subtractive manufacturing using computer-controlled cutting tools	High accuracy, suitable for strong materials, tight tolerances	Material waste, slower for complex geometries, expensive for prototyping

By leveraging manufacturing technologies, we can optimize the production process, ensuring that components are both functional and cost-effective while maintaining the flexibility to iterate on designs as needed.

3.1.6 Sensors

Although some motors, like ROBOTIS DYNAMIXEL motors, provide fine control over their motors, additional sensor technology is needed to meet the overall project goals. Sensors allow for the safe operation of the figure, maintaining safety for those around the figure and the figure’s components from any damage. Various sensors can be considered for use in the project, including:

- Proximity Sensors
- Limit Switches
- Optical Sensors
- Encoders
- Hall Effect Sensors
- Ultrasonic Sensors
- Temperature Sensors

Sensors are generally configured by the manufacturer to be either Positive Negative Positive (PNP) or Negative Positive Negative (NPN). PNP sensors have their output behave as a PNP transistor configuration, so the sensor sources current by connecting the output to the positive supply voltage. By definition this means that they are active high, since when the sensor is triggered, the output is a positive voltage.

On the contrary, an NPN sensor is set up as an NPN transistor on the output. When the sensor is triggered, the current sinks by the output being connected to ground. This means it is active low. There is no real advantage to using either sensor configuration, but rather it is important to keep in mind the differences to properly program the I/O on the main control unit.

Proximity sensors are used to detect whether an object is present in a particular position. For the figure, this would be useful to determine if a function has reached a certain position. In particular, it can be used to determine if a function is parked in a home position. Although the DYNAMIXEL servo motors can report their position, it is important to have a redundant sensor(s) that is independent of the motors.

Limit switches are a type of mechanical sensor that typically determines if an object has reached the end of its range of motion. Limit switches work by closing a switch whenever the object actuates some mechanical component, such as a lever or paddle. For the figure, it would be useful to determine if the function is reaching the end of its range.

Optical sensors use light to determine the position of an object. Some emit light that is reflected back to the sensor to read the position. However, for the case of the animatronic figure, an optical sensor would be used to determine a function’s position by having an emitter/receiver set up. This allows for either the figure or those around a figure to break the light barrier formed by the emitter/receiver optical sensor set up, which can initiate a safety stop.

Encoders take mechanical motion and turns them back into a digital signal. This allows for a variety of information to be processed such as position and speed. Encoders for the figure would allow for adding software limitations to the range of motion to prevent damage to the figure. The encoder can also be used to determine if the function is moving at the appropriate speed and to stop the figure if it is moving beyond the expected encoder count.

Hall effect sensors generate a voltage perpendicular to the flow of current once the sensor detects a magnetic field, with very high accuracy. These sensors are typically used for determining speed and position of motors.

Temperature sensors monitor the temperatures of the components for the figure. This is not just limited to the motors, but also the electrical enclosure housing other components within. If the enclosure begins to overheat, the system could turn off to prevent any serious damage to the components inside of the enclosure.

Ultrasonic sensors measure distance by bouncing high-frequency sound waves. More specifically, they work by emitting high-frequency sound waves (typically in the range of 40 kHz) and measuring the time it takes for the echo to return after bouncing off an object. This time-of-flight data allows the sensor to calculate the distance to the object with fairly high precision. These sensors are non-contact and can detect a wide range of materials, including solid objects, fluids, and some soft materials.

Table 3.6: *Sensor Technology Comparison*
By Authors

Sensor Type	Detection Method	Best Used For	Pros	Cons
Encoder	Mechanical rotation, digital pulses	Tracking motor position and speed	High resolution, great for closed-loop control	Requires calibration, cost varies
Proximity Sensor	Inductive or capacitive field changes	Detecting presence of metal or other materials at close range	Simple, reliable	Limited range, material-dependent
Limit Switch	Mechanical contact	Detecting end-of-range or physical contact	Fast, precise, good for safety applications	Prone to wear, requires physical actuation
Optical Sensor	Light beam interruption or reflection	Position detection or safety interruption		Sensitive to dirt, ambient light, alignment
Hall Effect Sensor	Magnetic field detection	Speed/position of motors, proximity sensing with magnets	Precise, contactless	Needs magnetic target, sensitive to electrical noise
Ultrasonic Sensor	High-frequency sound reflection	Proximity detection, user interaction, collision avoidance	Long-range, non-contact, works in many lighting conditions	Can be affected by noise, soft surfaces, or oblique angles

3.1.7 Supervisory Control and Data Acquisition; Alarming and Faults

In industrial automation and control systems, alarms and fault-handling mechanisms are essential for maintaining operational efficiency, safety, and reliability. These systems are deeply integrated with Programmable Logic Controllers (PLCs) and Human-Machine Interfaces (HMIs) to provide real-time diagnostics, automated responses, and operator alerts. Additionally, Supervisory Control and Data Acquisition (SCADA) systems play a crucial role in large-scale monitoring and control, enhancing the ability to detect, log, and respond to faults across multiple networked systems.

Alarms serve as critical indicators of system abnormalities, process deviations, or hardware failures. PLCs continuously monitor operational parameters such as voltage, temperature, pressure, and motion control. If a predefined threshold is exceeded, an alarm is triggered and sent to the HMI or SCADA system, ensuring operators are promptly notified.

Types of Alarms in Industrial Systems:

- **Warning Alarms:** Signal minor deviations that do not immediately impact operations but may require attention (e.g., voltage fluctuations or sensor drift).
- **Critical Alarms**: Indicate faults that could lead to performance degradation or system instability (e.g., overheating motors, failed communication with sensors).
- **Emergency Alarms:** Require immediate action, often leading to automated shutdowns to prevent catastrophic failures (e.g., PLC failure, power loss, or mechanical obstruction).

HMIs provide the graphical interface where alarms are displayed with detailed diagnostic information. Operators can view real-time alerts, acknowledge alarms, and access historical logs to analyze recurring issues. In more complex automation environments, SCADA systems aggregate alarm data from multiple PLCs, allowing for centralized monitoring across large facilities such as manufacturing plants, power stations, or water treatment facilities.

PLCs are responsible for detecting, isolating, and responding to system faults. When an issue arises, the PLC executes a structured fault-handling process:

- **Fault Detection:** Sensors and monitoring devices feed real-time data to the PLC, identifying deviations from normal operating conditions. For example, an actuator drawing excessive current may indicate a mechanical jam.
- **Fault Isolation:** The PLC determines the source and severity of the fault, ensuring that only the affected subsystem is taken offline while the rest of the system continues operating.

- **Automated Corrective Action:** Depending on the nature of the fault, the PLC may attempt an automatic resolution, such as reducing motor speed, adjusting parameters, or triggering a reset sequence.
- **Operator Notification via HMI/SCADA: If** the issue persists, the HMI displays a clear error message, guiding operators through troubleshooting steps. SCADA systems extend this functionality by providing a high-level overview of fault trends, enabling predictive maintenance strategies.
- **Fail-safe Activation:** If a fault presents a significant risk, the PLC can trigger emergency stop procedures or cut power to prevent further damage.

SCADA systems are critical for managing alarms and faults in large-scale automation environments. Unlike local HMIs that provide machine-level control, SCADA networks allow operators to monitor and control entire facilities remotely. These systems integrate with multiple PLCs, collecting and analyzing fault data across different processes.

SCADA Benefits for Fault Handling:

- **Centralized Monitoring:** Operators can oversee multiple PLC-controlled subsystems from a single interface.
- **Historical Data Analysis:** Logs all alarms and faults, enabling trend analysis to identify recurring issues.
- **Remote Control & Diagnostics:** Allows operators to acknowledge alarms, execute corrective actions, or adjust system parameters without being physically present.
- **Integration with Predictive Maintenance:** Uses data analytics and machine learning to anticipate failures before they occur, reducing downtime and maintenance costs.

By incorporating alarms and fault-handling mechanisms within PLCs, HMIs, and SCADA systems, industrial automation achieves greater safety, efficiency, and system longevity. These technologies work together to provide real-time monitoring, automated fault response, and predictive diagnostics, ensuring seamless and reliable operations.

3.1.8 HMI Software

With the hardware platform for the Human Machine Interface (HMI) established, attention turned to the selection of suitable software. Because the system relies on a Programmable Logic Controller (PLC) as its central control mechanism, the HMI software must be fully compatible with PLC communication protocols. This requirement significantly narrows the field of available software options, as many platforms are tailored to specific use cases or proprietary ecosystems.

In general, HMI software used within industrial settings can be classified into three overarching categories, based on its core application domain: Industrial Automation, Product Manufacturing, and Process Control. Each of these categories defines a different type of interaction between the system and its operator, and the distinction is critical when assessing which software most appropriately aligns with the objectives of this project.

Table 3.7: *Comparison of HMI Software Categories*
By Authors

Software Category	**Primary Focus**	**Strengths**	**Limitations for This Project**
Industrial Automation	Control and coordination of complex tasks usually handled by human operators	Detailed automation logic; supports complex workflows	Exceeds the complexity needed for animatronic control; introduces unnecessary overhead
Product Manufacturing	Monitoring repetitive machine activity and production metrics	Useful for tracking motor states and equipment status	Focused on productivity metrics lacks emphasis on active, real-time control
Process Control	Regulating active system parameters to achieve specific performance outcomes	Enables precise motor coordination, parameter tuning, and live fault diagnosis	Most closely aligned with project needs; essential for targeted and responsive system behavior

Industrial Automation software emphasizes the control and coordination of complex mechanical tasks typically performed by humans. While this approach accommodates detailed control schemes and extensive automation logic, its scope often exceeds the actual functional demands of our system. In contrast, software geared toward Product Manufacturing emphasizes the monitoring of repetitive machine behaviors, often focused on metrics such as unit output, defect rates, and efficiency tracking. While monitoring is indeed valuable, particularly in tracking the animatronics' motor behavior and idle states, this data-centric perspective misaligns with our objective, which is centered on direct system control rather than statistical analysis.

The most relevant category for our purposes is Process Control. Software within this category is designed to dynamically manage system behavior by regulating the parameters and outputs of active components. In the context of our animatronic, which features multiple motors requiring precise coordination and conditional activation, process control logic is essential. It supports the fine-tuning of performance, real-time adjustment, and the ability to diagnose misbehavior as it occurs. While many PLC-oriented software packages integrate features across all three domains, a strong process control foundation is necessary for effective deployment in our specific application.

It is best that the software used for our HMI is best suited for the tasks of Industrial Automation or Process Control. There are some top choices used in the industry, such as Ignition Edge, Vijeo Designer, and FactoryTalk. There is more software available than just these three, however this comes with a reason. Each program is made by a certain company: Ignition by Inductive Automation, Vijeo by Schneider Electric, and FactoryTalk by Rockwell Automation. Being developed by automation companies that have their own line of HMI products, it so happens that many of these software are locked; to only be compatible with a proprietary HMI.

This effectively limits the selection pool to only a few pieces of software that do not need a specific line of HMIs to function. The main two options to be considered are: Ignition Edge by Inductive Automation and InTouch by AVEVA. Both happen to be software that are compatible with third-party HMIs, which allows the use of the HMI selected and does not need a proprietary one which may exceed our budget. Ignition Edge proves to be the better software for our design, mainly for two reasons.

Half of the group working on this design happen to be familiar with Ignition. This familiarity may prove beneficial as less time would be needed to not only learn the software but to troubleshoot it as well. More importantly however, Ignition Edge offers a free license to be used. Normally, these software are usually priced by quotes that the developer would offer to companies in automation. And to ensure that the free license is actually usable, testing was done to ensure that it was compatible with our design.

3.1.9 HMI Communication, Serial vs Ethernet, Modbus, Transistor-Transistor Logic, CAN Bus

Determining the optimal communication method for device interconnection within the control system required a comprehensive comparison of the two most prevalent approaches: serial communication and Ethernet networking. Each method has distinctive advantages that make it more or less appropriate depending on the architecture and operational demands of the system. In evaluating these options, considerations such as data speed, physical connection distance, network topology, and communication protocol structure were central to identifying the most effective solution.

***Table 3.8:** Technology Comparison, Serial & Ethernet*
By Authors

Type	**Speed**	**Distance**	**Network**	**Data Transfer**	**Protocol**
Serial	Slower	Short	Point-to-Point	Bit	UART, I2C, SPI
Ethernet	Faster	Long	Large Networks	Packets	TCP/IP

Serial communication excels in applications where a single device must maintain a direct and exclusive connection to another. It is particularly effective over very short distances and in environments where communication simplicity is paramount. In such configurations, the data stream is linear and direct, minimizing the complexity of both hardware and software implementations. For compact systems, especially those where the human-machine interface (HMI) and control unit are located in close physical proximity, serial interfaces such as UART or SPI provide an efficient and space-conscious solution.

However, the design in question demands simultaneous communication between multiple system components, including the HMI, PLC, and motor controller. Achieving this through serial interfaces would necessitate a dedicated processing unit to serve as an intermediary, managing data requests and routing transmissions between devices. This additional layer of complexity introduces further software and hardware requirements, increasing the development burden without contributing meaningful value to the core control functionality.

Ethernet communication offers a more scalable and robust alternative, particularly when consistent data transfer speed and multiterminal connectivity are required. Unlike serial connections, Ethernet maintains high performance regardless of cable length, owing to the presence of switches and routers that actively manage packet integrity and transmission efficiency. These devices implement built-in error-checking mechanisms and ensure that packet delivery remains reliable across complex topologies.

This architectural advantage aligns precisely with the system's requirements. By employing Ethernet, multiple devices can interact through a shared network infrastructure without requiring point-to-point wiring or protocol conversion. Data flow between components is maintained at gigabit speeds, with minimal latency and strong resistance to signal degradation. Furthermore, the inclusion of industry-standard Ethernet interfaces facilitates straightforward integration with commercial components and diagnostic tools, enhancing both the development and maintenance processes.

Given the need for efficient, reliable, and easily expandable communication across several devices, Ethernet presents a clear advantage over serial methods.

With Ethernet established as the primary communication medium, selecting an appropriate protocol became essential to defining how devices exchange information within the system. The protocol governs data structure, transmission, and integrity, and directly impacts the scalability and performance of the design. Among common industrial protocols: Modbus TCP, Modbus RTU, RS-232, and RS-485, Modbus TCP was selected for its compatibility with Ethernet infrastructure and its robust handling of high-speed, multi-device communication.

Table 3.9: *Technology Comparison, Modbus*
By Authors

Protocol	Medium	Max Device Support	Speed & Reliability	Complexity
Modbus TCP	Ethernet	Hundreds	High speed: TCP/IP ensures reliable packet delivery	High
Modbus RTU	RS-485	~32	Moderate speed; limited error handling	Limited
Relay-Based I/O	Hardwired	N/A	Very fast for binary states	Not networked

Modbus TCP builds upon the familiar register-based structure of Modbus RTU while incorporating the advantages of the TCP/IP stack, including built-in packet verification and retransmission. This results in faster, more reliable communication between system nodes, which is critical given that the HMI, PLC, and motor controller must exchange data concurrently and without delay. Moreover, Modbus TCP does not require intermediary hardware for protocol conversion, simplifying integration and reducing potential failure points.

In contrast, serial protocols such as RS-232 and RS-485 impose constraints that make them unsuitable. RS-232 supports only point-to-point connections, while RS-485, though more flexible, suffers from slower speeds and increases susceptibility to noise. Both introduced added complexity through hardware addressing and custom communication handling. Similarly, relay-based I/O, while reliable for binary signals, lacks the data structure and diagnostics necessary for dynamic system control.

In conclusion, Modbus TCP offers the necessary speed, flexibility, and integration simplicity required by our system. Its Ethernet compatibility, robust communication model, and long-term scalability make it the optimal choice for PLC communication within the control architecture.

Transistor-Transistor Logic (TTL) is a class of digital logic circuits built from bipolar junction transistors (BJTs), resistors, and diodes. Transistors provide switching, turning on or off in response to input signals. Resistors limit current and help optimize voltage levels for the transistors. Diodes will stabilize the circuit by preventing current flow in the wrong direction.

TTL circuits are widely used in digital systems to perform standard logical operations including AND, OR, NOT, NAND, NOR, XOR, and XNOR gates. Typically, TTL logic gates are contained into one IC. TTL ICs can be identified via the IC having 4-digit numbers that begin with either a 74 or 54.

A TTL gate typically consists of transistors functioning both as switching devices and as amplifiers. The fundamental operation of TTL circuits involves using BJTs as switches. The switches operate by allowing current flow through the circuit when activated which can be referred to as logic high, or binary ‘1’, or preventing current flow when deactivated - logic low, binary ‘0’. Voltage levels define the different logical states:

- Logical high / binary ‘1’ - 2V to 5V, standard at 5V
- Logical low / binary ‘0’ - 0V to 0.8V
- Undefined - 0.8V to 2V.

TTL logic families include standard TTL, Low Power TTL (LPTTL), High-Speed TTL (HSTTL), Schottky TTL (STTL), and Advanced Schottky TTL (ASTTL), each offering unique benefits ranging from reduced power consumption to higher switching speeds, and more. In modern applications, standard TTL has mainly been replaced by advanced TTL variants or complementary metal-oxide-semiconductors (CMOS) due to lower power consumption and higher integration densities, among other reasons. However, TTL remains crucial in specific applications where robustness and reliability in noisy environments are essential.

The Controller Area Network (CAN) bus is a differential signaling communication protocol. It facilitates reliable data exchange between multiple devices, referred to as nodes, within a network, even under electrically noisy conditions.

CAN bus operates using two wires known as CAN High (CAN_H, colored yellow) and CAN Low (CAN_L, colored green), transmitting differential signals that effectively resist electromagnetic interference. Each device on a CAN bus communicates using messages rather than traditional addresses. These messages are uniquely identified by identifiers indicating priority; lower numerical identifiers indicate higher priority, ensuring critical messages receive immediate attention.

CAN communication is inherently robust due to its built-in error detection and correction mechanisms. Each message transmitted over a CAN network includes Cyclic Redundancy Check (CRC) data, ensuring message integrity. Error-detection capabilities include acknowledgment bits and error frames to identify and rectify corrupted messages automatically. Furthermore, CAN's arbitration mechanism ensures that if two nodes transmit simultaneously, the message with the higher priority continues transmission, while the lower priority node pauses and retries afterward.

3.1.10 Programming Languages

Programming languages play a crucial role in motor control, automation, and PLC programming, enabling precise system operation, real-time control, and seamless integration between hardware and software. Different languages are used depending on the system architecture, control requirements, and hardware specifications. Some languages, such as C and Python, are commonly used for embedded systems and automation logic, while others, such as Ladder Logic and Structured Text, are specifically designed for PLC-based industrial control. The selection of the appropriate programming language is essential for ensuring efficiency, maintainability, and real-time responsiveness in animatronics and industrial automation applications.

3.1.10.1 C

C is a low-level, high-performance programming language widely used for embedded systems, motor control, and industrial automation. It provides direct hardware access, allowing for precise manipulation of registers, memory, and I/O operations, which is essential for real-time motor control and sensor interfacing. C is particularly beneficial for applications that require fast execution speeds and deterministic timing, making it well-suited for microcontrollers, real-time operating systems (RTOS), and automation firmware.

3.1.10.2 C++

C++ is an extension of the C programming language that introduces object-oriented programming (OOP) features while retaining the low-level hardware access capabilities of C. This combination makes C++ a powerful choice for real-time embedded systems, motor control, and high-performance automation applications.

One of the key strengths of C++ is its ability to implement modular and reusable code through classes and objects, making it easier to manage complex automation projects. It also provides direct memory manipulation, allowing for precise control over hardware resources, making it highly suitable for motor controllers, industrial robots, and high-speed data acquisition systems.

However, compared to C, C++ has increased memory overhead and execution complexity, which may not be ideal for resource-constrained microcontrollers. In real-time control systems, developers often prefer C over C++ due to the need for absolute control over execution timing. Despite this, C++ remains a valuable choice for applications requiring both performance and modular software design.

One of the major strengths of C is its portability; code written in C can be compiled and executed on various platforms, from microcontrollers to industrial computers. Additionally, it has an extensive set of libraries and frameworks that facilitate motor control algorithms, PID control loops, and industrial communication protocols such as Modbus and CAN bus. However, C lacks built-in support for object-oriented programming (OOP) and higher-level abstractions, requiring more manual management of memory and system resources. Despite these limitations, its efficiency, reliability, and direct hardware access make it an industry-standard for motor control and automation systems.

3.1.10.3 Ladder Logic

Ladder Logic is one of the most commonly used PLC programming languages, designed to resemble relay-based electrical control diagrams. It is a graphical programming language that allows engineers to construct control logic using virtual relays, timers, counters, and logic gates, making it ideal for industrial automation and motor control applications.

One of the key advantages of Ladder Logic is its intuitive visual representation, making it accessible for electrical engineers who are familiar with relay-based control systems. It is primarily used for discrete control tasks, such as starting and stopping motors, activating solenoids, and implementing safety interlocks. PLCs executing Ladder Logic operate in a cyclic scan mode, continuously evaluating input conditions and executing control logic in a deterministic manner.

Despite its ease of use, Ladder Logic is not well-suited for complex mathematical operations or advanced data processing. It is best used for sequential control processes, safety mechanisms, and real-time industrial automation tasks where simplicity and reliability are paramount.

3.1.10.4 Structured Text (ST)

Structured Text (ST) is a high-level, text-based PLC programming language, standardized under IEC 61131-3, and commonly used for complex industrial automation applications. Unlike Ladder Logic, Structured Text follows a syntax similar to Pascal or C, allowing for conditional statements, loops, and function calls, making it more flexible for advanced control algorithms.

Structured Text is ideal for applications that require precise numerical computation, data handling, and complex logic implementation. It is often used in conjunction with PID control loops, mathematical modeling, and industrial process automation. The language enables modular programming, allowing engineers to create reusable function blocks and scalable control logic.

One of the main benefits of Structured Text is its ability to handle continuous and high-speed automation processes, which are difficult to implement in Ladder Logic. However, due to its text-based nature, debugging and troubleshooting can be more challenging compared to graphical programming methods like Ladder Logic or Function Block Diagram (FBD).

3.1.10.5 Python

Python is a high-level, interpreted programming language widely used in automation, robotics, and industrial control applications. It is known for its simple syntax, extensive libraries, and versatility, making it a valuable tool for data processing, machine learning, and system automation.

In industrial automation, Python is often used for high-level system control, data logging, and communication between different automation components. It supports Modbus, OPC UA, MQTT, and other industrial communication protocols, allowing seamless interaction with PLCs, sensors, and actuators. Additionally, Python’s integration with machine learning frameworks makes it useful for predictive maintenance, anomaly detection, and optimization of motor control parameters.

While Python offers significant advantages for rapid development and software integration, it is not a real-time language and lacks deterministic execution timing, making it unsuitable for low-level motor control tasks that require microsecond precision. However, it is commonly used for remote monitoring, industrial dashboards, and automation control scripts.

3.1.10.6 Java

Java is a high-level, object-oriented programming language known for its cross-platform compatibility, modularity, and scalability. It is commonly used in industrial SCADA systems, remote monitoring applications, and large-scale automation software. Java’s built-in networking capabilities make it an excellent choice for distributed control systems, cloud-based automation, and IoT integration.

One of the primary advantages of Java is its object-oriented architecture, which promotes code reusability, scalability, and maintainability. However, Java is not well-suited for low-level hardware control or real-time applications due to its dependency on a virtual machine, which introduces execution overhead and unpredictable timing delays. While Java is widely used for SCADA, factory automation, and enterprise integration, it is rarely used for real-time motion control or direct PLC programming.

3.1.10.7 Programming Language Summary

Below is a summarized table comparison, with our selection of programming language technology highlighted.

Table 3.10: *Programming Language Technology Comparison*
By Authors

Language	Type	Strengths	Weaknesses
C	Low-level	Fast execution, direct hardware access, portable, deterministic timing	No OOP, manual memory management, steeper learning curve
C++	Mid-level + OOP	Modular code via OOP, hardware-level control, high performance	Higher memory overhead, complex for resource-limited MCUs
Ladder Logic	PLC Graphical	Easy for electrical engineers, highly reliable, real-time cyclic scanning	Poor for complex math/data processing, not suitable for high-speed logic
Structured Text	PLC Text-based	IEC 61131-3 standard, supports loops/conditions, reusable blocks, compact for advanced logic	Harder to debug than visual languages, less intuitive for beginners
Python	High-level script	Easy syntax, rich libraries, supports many protocols (Modbus, MQTT, OPC UA), great for analytics	Not real-time, unsuitable for direct motor control or hard real-time systems
Java	High-level OOP	Cross-platform, scalable, strong networking support, good for cloud interfaces	JVM adds overhead, non-deterministic timing, not suitable for motor control

3.1.11 Fuses

Fuses are critical components in any electrical circuit because they provide essential protection against excessive current, which can lead to equipment damage or, in some cases, even a fire. A fuse consists of a thin wire designed to melt when the current flowing through it surpasses a safe level, causing an open circuit. This immediate break in the circuit prevents further current from reaching the load, ensuring that the device being powered is safeguarded from potentially dangerous electrical spikes.

3.1.12 Electrical Enclosures

Electrical enclosures serve as a protective housing for industrial control systems, ensuring the safety, organization, and reliability of critical electronic components. These enclosures provide physical protection from environmental hazards such as dust, moisture, and extreme temperatures, while also safeguarding personnel from electrical hazards by preventing direct contact with high-voltage components. Proper enclosure selection is essential for maintaining system integrity, extending equipment lifespan, and ensuring compliance with industrial safety standards.

The enclosure must accommodate essential electrical components, including a PLC, power supplies, breakers, relays, and more. Given the complexity of an animatronic control system, the enclosure must be designed with adequate space, efficient heat dissipation, and accessibility for maintenance and troubleshooting. Enclosure layouts typically include DIN rail mounting for modular components, cable management systems for clean wiring, and ventilation or active cooling systems to prevent overheating.

Electrical enclosures vary in design, material, and protection level, depending on the operational environment. Metal enclosures, such as those made from stainless steel or aluminum, are commonly used in industrial and outdoor settings where durability and resistance to corrosion are necessary. Plastic enclosures, often constructed from polycarbonate or fiberglass, are lightweight, cost-effective, and provide insulation against electrical shock, making them ideal for indoor and non-corrosive environments.

Enclosures are also rated based on their level of protection against dust, water, and mechanical impact. The National Electrical Manufacturers Association (NEMA) and the Ingress Protection (IP) rating system classify enclosures according to their environmental resistance. For example, NEMA 4 and IP66-rated enclosures are designed for outdoor use, protecting against heavy rain and washdowns, while NEMA 12 and IP54-rated enclosures provide adequate protection for indoor industrial applications.

Since PLCs, power supplies, and relays generate heat during operation, thermal management is a critical factor in enclosure design. Overheating can cause system failures, reduce component lifespan, and compromise overall performance. Depending on the heat dissipation requirements, enclosures may feature passive ventilation, forced air cooling, or even air conditioning systems. Heat sinks and fan-assisted cooling are commonly used for enclosures housing high-power automation components.

As automation systems evolve, electrical enclosures must be designed with scalability in mind. Modular enclosures allow for future expansion, accommodating additional PLC modules, sensors, or motor controllers as needed. Additionally, enclosures should provide conduit and cable gland entry points to facilitate organized wiring and reduce electromagnetic interference (EMI)002E

Table 3.11: *Electrical Enclosure Technology Comparison*
By Authors

Type	**Material**	**Protection Level**	**Advantages**	**Limitations**
Steel (NEMA 4X)	Steel	Waterproof, dustproof,	Outdoor rated	Heavy
Aluminum	Aluminum Alloy	Water-resistant	Lightweight	UV Degradation
Fiberglass	Fiberglass-Reinforced Plastic (FRP)	Waterproof, chemical resistant	High corrosion resistance and lightweight	Susceptible to damage if dropped

3.1.13 Ethernet Switch

With the selection of Ethernet as the core communication medium for the system, the next step in the network design process involved determining the appropriate Ethernet cable standard and the supporting switch hardware. The Ethernet switch serves as the central communication hub, linking key devices, namely the HMI, PLC, motor controller, and programming interface, into a unified network. Its role is essential in facilitating data transmission between components at high speed and with dependable reliability. Thus, both the cabling standard and the specific switch characteristics must be selected with a careful balance of performance, cost, and system compatibility.

The first consideration addressed the type of Ethernet cable to be used. Several categories are available, each offering different capabilities in terms of data rate, signal bandwidth, shielding, and cost. For this application, a data transmission speed of at least 1 Gbps was set as the minimum acceptable standard. This performance threshold effectively eliminates older categories such as Cat 5, which is limited to 100 Mbps and has become increasingly obsolete in modern network infrastructure. As a result, Cat 5e, Cat 6, and Cat 6a emerged as the remaining candidates.

Table 3.12: *Comparison of Ethernet Categories*
By Authors

Category	Max. Data Rate	Bandwidth	Max Distance	Standard
Cat 5	100 Mbps	100 MHz	100m	Fast Ethernet
Cat 5e	1 Gbps	100 MHz	100m	Fast Ethernet
Cat 6	1 Gbps	250 MHz	100m (10 Gbps @ 37m	Gigabit Ethernet
Cat 6a	10 Gbps	500 MHz	100m	Gigabit Ethernet

Although Cat 6a offers the highest data rates and bandwidth, it also introduces significant drawbacks for this application. The increased frequency used by Cat 6a cables leads to a higher likelihood of signal interference and electromagnetic noise, conditions that can destabilize communication in unshielded or electrically noisy environments. While Cat 6a cables typically include improved shielding to mitigate this issue, that very shielding adds to both the physical bulk and the overall cost of the cable. Moreover, achieving the full benefit of Cat 6a's 10 Gbps speed requires every device in the network, including the Ethernet switch and connected interfaces, to support the same high-speed standard. This not only adds cost but also imposes stricter requirements on all networking hardware.

By comparison, Cat 6 cables provide an optimal balance for the current system's needs. With support for gigabit speeds at 250 MHz and up to 100 meters in length, Cat 6 meets the performance threshold without incurring the cost penalties or signal management complexity of Cat 6a. Furthermore, Cat 6 is widely compatible with standard gigabit Ethernet switches and devices, which streamlines component selection and integration.

Following the selection of Cat 6 cabling, attention turned to the Ethernet switch itself. This device must meet three core criteria: it must include at least four ports to accommodate the system's PLC, HMI, motor controller, and programming interface; it must operate reliably at 24VDC, consistent with the control cabinet power supply; and it must support gigabit-speed communication to preserve data throughput across all links.

Several switch characteristics were also considered, including whether the device should be managed or unmanaged, whether it should support Power over Ethernet (PoE), and whether it should feature a modular or fixed configuration. Managed switches offer extensive control over network behavior, including traffic monitoring, security settings, and port prioritization. However, this level of control introduces setup complexity, additional cost, and unnecessary features for a system with fixed topology and predictable data flows.

Table 3.13: *Ethernet Switch Selection Criteria and Evaluation*
By Authors

Criteria	**Requirement**	**Rationale**	**Preferred Option**
Number of Ports	Minimum of 4 (preferably 5)	To support HMI, PLC, motor controller, and programming interface, with room for expansion	5-port switch
Operating Voltage	24V DC	Matches control cabinet supply, eliminates need for external converters	Native 24V DC compatibility
Data Transfer Speed	≥ 1000 Mbps (1 Gbps)	Ensures no communication bottlenecks and full use of Cat 6 cabling capabilities	Gigabit-speed support
Managed vs Unmanaged	Unmanaged	Reduces complexity, cost, and setup time; suitable for fixed and predictable network topologies	Unmanaged switch
Power over Ethernet (PoE)	Not required	All devices have dedicated power; PoE adds unnecessary cost and complexity	Non-PoE switch
Configuration Type	Fixed	Network design is finalized; modular features are not needed and would increase cost	Fixed-configuration switch

With the cable category established, the next critical decision involved selecting the appropriate Ethernet switch. The switch acts as the central interconnect for all networked devices, handling data traffic routing between system components. For our design, three major parameters guided the selection process: the number of ports, the supported data speed, and the required operating voltage. The system comprises four essential devices requiring Ethernet connectivity: the HMI, the PLC, the motor controller, and a programming interface, necessitating a minimum of four dedicated ports. To allow for future expansion or redundancy, a five-port switch is preferable.

Operational voltage was also a significant consideration. The control cabinet in which the switch will be installed operates at 24V DC. Selecting a switch that supports this voltage natively avoids the need for additional power conversion circuitry or external regulators, simplifying power distribution and reducing component count. Lastly, in keeping with the performance requirements of the selected Cat 6 cabling, the switch must support data transfer speeds of at least 1000 Mbps (1 Gbps) to ensure that no bottlenecks are introduced in the communication chain.

Additional evaluation criteria focused on features such as switch management capability, Power over Ethernet (PoE) functionality, and switch configuration (modular vs. fixed). Managed switches allow advanced control over network traffic, including port prioritization, VLAN setup, and traffic diagnostics. While valuable in larger or highly dynamic network configurations, these features are unnecessary in our relatively static system, where the network topology is fixed, and data flow is consistent. Managed switches also require more involved configuration, increase the learning curve, and are generally priced higher than unmanaged alternatives.

By contrast, unmanaged switches offer a straightforward, plug-and-play solution. These devices are preconfigured to route Ethernet traffic with default settings and require no setup or ongoing administration. This simplicity is highly beneficial in our case, where the focus is on reliable functionality rather than dynamic reconfiguration. An unmanaged switch reduces setup time, minimizes potential failure points, and aligns well with our objective to maintain a lean and efficient control system.

Based on these factors, the ideal switch for our application is an unmanaged, non-PoE, fixed-configuration unit supporting at least five ports, operating at 24V DC, and capable of gigabit-speed communication. This specification ensures full compatibility with our selected Ethernet cable standard, satisfies all electrical and mechanical constraints, and avoids the cost and complexity of unnecessary features.

3.1.14 Circuit Breakers

A circuit breaker is a type of automatic electrical switch designed to protect electrical circuits from damage caused by too much current or short circuits. A circuit breaker works by detecting an unusual level of current flow and interrupts the electrical connection to prevent overheating or just general damage to the electrical components. Unlike a fuse, which needs to be replaced after too much current goes through it, a circuit breaker can be reset after it trips, which makes it reusable and more convenient for protection. In our project we will be using 1A double pole circuit breakers which we will implement to provide overcurrent protection for components such as the MCU, HMI interface, and possibly an audio output module. The robot will also include higher-current components such as servo motors and power supplies, however the logic-level electronics are more sensitive to electrical faults and require finer protection. Using 1A breakers in these branches ensures that if something goes wrong like a short circuit, reverse polarity, or component failure, the breaker will trip and protect the components without damaging the system. The circuit breakers will be mounted on a din rail inside the electrical enclosure, where they will be integrated into the power distribution system. We will be using double pole circuit breakers which are designed to simultaneously disconnect both the hot and neutral lines in a circuit. This type of breaker is important to use because we will be working with high voltage AC power.

***Table 3.14:** Circuit Breaker Comparison*
By Authors

Circuit Breaker Type	Description	Advantages	Limitations
MCB	Protection from overloads and shorts	Small, cheap, and easy to reset	Low current rating
MCCB	High-Power overload protection	Adjustable, handles large loads	Bulky and expensive
Double Pole Breaker	Disconnects both hot wires	Full isolation, safe at high voltages	Uses up more space
Ground Fault Circuit Interrupter	Trips on ground faults	Shock protection in wet areas	Nuisance tripping and no arc protection
Arc Fault Circuit Interrupter	Detects electrical arcs	Prevents fires and makes wires safer	Expensive, false trips are possible

3.1.15 PCB

Printed Circuit Boards, also known as PCB's, are a foundational element in pretty much every modern electronic device. A PCB is essentially a flat board made of an insulating material such as fiberglass-reinforced epoxy with conductor copper traces laid out in a specific pattern to facilitate electrical connections between different components. This is in stark contrast with Point-to-Point wiring which is basically components being wired directly to each other. This method of wiring together components is bulky and can oftentimes lead to more mistakes.

The main advantage of using a PCB is that it provides a compact and organized reliable way to interconnect electronic components such as the MCU, resistors, and capacitors. In addition to these numerous benefits, it is also helpful to troubleshoot errors because of the ability to isolate issues by checking specific traces and components. This also makes it much easier to reproduce a new PCB if needed due to the visual nature of the electronic components and their connections.

The most common type of PCB technology used is the FR4 rigid PCB. It is made of fiberglass epoxy, it's cost effective, reliable, and supports analog and digital circuit designs. In our project the FR4 is useful in providing the structural support needed to mount the MCU, voltage regulators, and motor drivers. Rigid PCBs are, as the name suggests, not meant to bend. Rigid-flex PCBs on the other hand have both rigid and flexible parts. Flex PCB's is the last option to consider, and it is made from a flexible base. Since these PCB's can bend, they are able to withstand vibration and be molded into different shapes.

Table 3.15: *PCB Technology Comparison*
By Authors

	Material	**Cost**	**Applications**
Rigid FR4	Fiberglass Epoxy	Low	General Electronics
Rigid-Flex	FR4 + Polyimide	Very High	Embedded Systems
Flex	Polyimide	High	Foldable Designs

In our project, the PCB will be the housing for the MCU, as well as interfaces for various components such as servo motor drivers, power input designs, communication lines, and possibly an audio output module. The PCB design process will begin with a schematic. A schematic is a diagram that shows how each component is connected. We will likely use Eagle (Fusion 360) in order to build our PCB, but other software tools could be used as well such as KiCad or Altium Designer.

Table 3.16: *PCB Software Comparison*
By Authors

Software Tools	**Ease of Use**	**Cost**
Fusion360	Low Learning Curve	Free (For UCF Students)
KiCad	Moderate Learning Curve	Free
Altium	High Learning Curve	High-Cost

Both Fusion360 and KiCad are free software tools to use so for our project we will be using either one of those as Altium is made more for working professionals rather than students and thus requires a subscription. Whether we use Fusion360 or KiCad, either way we will eventually need to translate this schematic into a physical layout. This would mean we need to place the electrical components in specific positions by routing copper traces to form a proper connection. Power traces, for example, carry higher current and thus need to be made wider to handle the load. We might also include features such as hole mounting, which is when we secure the PCB to a chassis using hold drilled through the board at the corners, in order to allow for screws or fasteners. This essentially allows us to mount the PCB to an enclosure or surface.

Once the design is completed, we will need to export the files needed to manufacture the PCB. These files are called Gerber files. Once these Gerber files are made available then we will be able to produce the PCB board according to our specifications by sending it to a PCB fabrication house. Once we receive the PCB, we will then solder the components needed onto the board ourselves. This step will most likely be completed in its entirety in Senior Design 2. From there, the PCB essentially becomes the control hub of the robot, distributing power and signals precisely where needed. In terms of implementation, the PCB will be housed inside of the animatronics electrical enclosure. All input and output devices, such as the servos to the touch screen HMI, will connect to the PCB via headers or screw terminals. By centralizing control on the PCB, we ensure all of the subsystems work together in a coordinated way. The PCB will be protected by using a 10A fuse to protect sensitive components from an overload of current. Voltage regulators or converters that step-down voltage from 12V to 5V will also be implemented for the microcontroller and logic circuits.

In summary, the PCB actively defines the structure, layout, and reliability of the entire electrical system. Careful design and planning will allow us to make a PCB that meets our performance goals and simplifies integration.

3.1.16 Voltage Regulation

A DC-to-DC regulator is a vital part of any electronic system where different components require different voltage levels to operate consistently without the voltage levels fluctuating. In our project we will need to design a voltage regulator that is able to step voltage down from 12V to 5V to ensure the safe operation of the microcontroller unit (MCU). The MCU is the brain of our system. Its functions include things such as processing inputs, controlling the servo motors, and managing communication. Ensuring that it receives a stable and clean voltage is a top priority.

Voltage regulators work by maintaining a constant output voltage regardless of variations in input voltage or load current. DC-to-DC regulators generally fall into two categories: linear regulators and switching regulators.

Linear Regulators

Linear regulators are generally considered to be much simpler when compared to switching regulators. A linear regulator works by using a series element like a transistor that can dynamically adjust its resistance to drop the excess input voltage. Internally, it compares the output voltage to a stable internal reference voltage, and through the process of negative feedback, it adjusts the transistor or whatever pass element it’s using and adjusts it to maintain a consistent output. For example, if the input voltage were to rise, the regulator would then increase the resistance of the pass element to maintain the same output voltage. Linear voltage regulators are simple and effective for filtering out noise, but when the difference between the input and output voltage is too large, they become inefficient. When choosing a linear voltage regulator, it is important to keep this in mind. To better understand the performance characteristics of linear voltage regulators, it is helpful to look at some equations that quantify their behavior:

$$\text{Power Dissipation} = (V_{in} - V_{out}) * I_{out}$$
$$I_{out} = (\text{Power Dissipation}) / (V_{in} - V_{out})$$
$$\text{Efficiency } (\eta) = (V_{out} * I_{out}) / [V_{in} * (I_{cc} + Iout)]$$
$$V_{out} = [V_{ref} * (R_1 + R_1)] / (R_1)$$

These formulas allow us to evaluate how much power is being dissipated as heat, how efficient the regulator is under different load conditions, and how the output voltage is determined in adjustable regulator configurations. By applying these formulas, we can make informed decisions about thermal management, power budgeting, and component selection in our design.

Switching Regulators

Switching regulators on the other hand, operate using a completely different working principle - energy storage and transfer. Instead of dissipating excess voltage as heat, switching regulators use high frequency switching to transfer energy from input to output with the use of inductors, capacitors and diodes. The main types of switching regulator types include Buck Converter, Boost Converter, or a combination both, a Buck-Boost Converter. A buck converter, also known as a step-down converter, is used when the output voltage needs to be lower than the input voltage. It is one of the most commonly used types of switch regulators due to its efficiency and simplicity.

The buck converter works by rapidly switching a transistor (like a MOSFET) on and off. When the switch is on, the current flows from the input through an inductor to the load, which stores energy in the inductor's magnetic field. When the switch is off, the inductor will continue to supply current to the load by releasing the stored energy through a diode or rectifier, maintaining a continuous flow of energy. A capacitor is implemented across the output to smooth the voltage and provide a clean and steady DC output. To control the output voltage, the duty cycle (time switch is on during each cycle) of the switch must be adjusted. The duty cycle can be calculated by simply dividing the input voltage by the output voltage:

$$\textit{Duty Cycle} = V_{out} / V_{in}$$

A boost converter, or step-up-converter, is designed to produce an output voltage higher than its input voltage. It is especially useful in battery-powered systems where the input voltage can be lower than what is required by the load. In a boost converter, the switch connects the inductor to the ground, allowing it to store energy while the load is temporarily powered by the output capacitor. When the switch opens, the energy stored in the inductor is released and added to the input voltage, which results in a higher output voltage. Including a diode in the design ensures that current can only flow in one direction. A boost converter is useful in our project if we need to maintain operation as battery voltage drops.

A buck-boost converter has a flexible topology that can either increase or decrease the input voltage depending on operating conditions. This makes it ideal in situations where the input voltage may vary above or below the desired output voltage. The buck-boost converter circuit combines features of both the buck and boost converters in the sense that it uses a similar inductor-based energy storage and switching technique. It differs however in its topology; each topology being suited for different requirements such as isolation and polarity. In a basic inverting buck-boost converter, the output voltage is negative with respect to the input voltage, but non-inverting configurations could also be designed. A buck-boost converter would be helpful to use in the case of fluctuating input voltage levels.

Table 3.17: *Voltage Regulators Comparison*
By Authors

Parameters	**Buck**	**Boost**	**Buck-Boost**	**Linear Voltage Regulators**	**Switching Regulators**
Output Voltage	Step-Down	Step-Up	Step-Up or Step-Down	Step-Down	Step-Up or Step-Down
Energy Efficiency	85-95%	85-95%	75-90%	40-70%	80-95%
Heat Dissipated	0.5-2 W	0.5-2 W	1-3 W	2-10 W (depends on load drop)	0.5-2 W
Design Complexity (by number of components)	Moderate	High	High	Low	High
Cost	$1-$3	$1-$4	$3-$7	$0.30-$1.00	$2-$5

3.2 Part Comparison:

This section presents a comparison of various parts and components that could be utilized for the project's needs. The knowledge and research gained in the previous section leads the way to deciding on the best parts to use, deliberating on their advantages and limitations. After determining the most suitable approach, specific components can be selected to implement each chosen technology.

3.2.1 Programmable Logic Controllers (PLCs)

Programmable Logic Controllers (PLCs) are a critical component of the animatronic system, responsible for executing control logic, managing inputs and outputs, and ensuring precise and synchronized movements. This section compares several major PLCs, including the Allen-Bradley Micro800 series and other Allen-Bradley models, to help determine the most suitable option for the project.

Allen-Bradley Micro800 Series

The Micro800 series is a cost-effective and versatile line of PLCs designed for small to medium-sized applications. It is particularly well-suited for projects with budget constraints and moderate complexity, such as the animatronic system.

Advantages:

- Affordable pricing, making it ideal for cost-sensitive projects.
- Integrated programming environment (Connected Components Workbench) that supports ladder logic and function block diagrams.
- Built-in communication ports (Ethernet, USB, and serial) for easy integration with HMIs and other devices.
- Compact design, saving space in control panels.
- Scalable I/O options, allowing for future expansions.

Limitations:

- Limited processing power compared to high-end PLCs, which may restrict its use in highly complex applications.
- Smaller memory capacity, which could be a constraint for large programs or data logging.

Allen-Bradley CompactLogix 5370 Series

The CompactLogix 5370 series is a mid-range PLC offering higher performance and scalability compared to the Micro800 series. It is suitable for applications requiring advanced control capabilities and real-time performance.

Advantages:

- Powerful processing capabilities, suitable for complex control logic and real-time applications.
- Integrated Ethernet/IP supports high-speed communication with other devices.
- Scalable with additional I/O modules and communication options.
- Comprehensive programming environment (Studio 5000 Logix Designer) with support for multiple IEC 61131-3 languages.

Limitations:

- Higher cost compared to the Micro800 series, which may not align with the project’s budget constraints.
- Steeper learning curve for beginners due to the complexity of the Studio 5000 software.

Allen-Bradley ControlLogix Series

The ControlLogix series is a high-end PLC designed for large-scale and complex automation applications. It offers advanced features and robust performance, making it suitable for demanding environments.

Advantages:

- Exceptional processing power and memory capacity, ideal for highly complex applications.
- Supports advanced communication protocols, including Ethernet/IP, ControlNet, and DeviceNet.
- Highly scalable with modular I/O and communication options.
- Comprehensive programming environment (Studio 5000 Logix Designer) with advanced diagnostic and troubleshooting tools.

Limitations:

- High cost, which may be prohibitive for budget-conscious projects.
- Requires significant expertise to program and configure effectively.

Siemens S7-1200 Series

The Siemens S7-1200 is a mid-range PLC known for its robust performance and flexibility. It is widely used in industrial automation and is suitable for applications requiring moderate to high complexity.

Advantages:

- Powerful processing capabilities, suitable for complex control logic and real-time applications.
- Integrated PROFINET support for high-speed communication with other devices.
- Scalable with additional I/O modules and communication options.
- Comprehensive programming environment (TIA Portal) with support for multiple IEC 61131-3 languages.

Limitations:

- Higher cost compared to the Micro800 series, which may not align with the project’s budget constraints.
- Steeper learning curve for beginners due to the complexity of the TIA Portal software.

***Table 3.18**: PLC Part Comparison*
By Authors

Feature / Criteria	**Allen-Bradley Micro800 Series**	**Allen-Bradley CompactLogix 5370 Series**	**Allen-Bradley ControlLogix Series**	**Siemens S7-1200 Series**
Cost	$200-$500	$1,500-$3,000	$5,000-$10,000	$1,000-$2,500
Memory Capacity	64-512 KB	2,000-4,000 KB	10,000+ KB	2,000-4,000 KB
Communication via Port	Ethernet, USB, Serial	Ethernet/IP, Serial	Ethernet/IP, ControlNet, DeviceNet	PROFINET, Ethernet
Programming Software	Connected Components Workbench	Studio 5000 Logix Designer	Studio 5000 Logix Designer	TIA Portal
Software Cost	Free	Paid	Paid	Paid

The Allen-Bradley Micro800 series is a strong candidate for the animatronic system due to its affordability, ease of use, and free programming software. While it may lack the advanced features and processing power of higher-end PLCs like the CompactLogix 5370 or ControlLogix, it provides sufficient performance for the project's requirements. The CompactLogix 5370 offers a good balance of performance and scalability for more complex applications, while the ControlLogix series is ideal for large-scale systems with demanding requirements. The Siemens S7-1200 is also a viable option, particularly for applications requiring high-speed communication and advanced control capabilities. By carefully evaluating these factors, the team can select the most suitable PLC to ensure reliable and efficient operation of the animatronic system.

The Micro820-AWB is the PLC that we will be using in our design.

3.2.2 Human-Machine Interfaces (HMIs)

When selecting for a Touch-Display HMI, three types were available on Digikey which were defined by the CPU used and the Operating System installed. These would consist of an Intel CPU with a Windows OS, an ARM CPU with a Linux OS, and a Raspberry Pi with a Raspbian OS. With these pairings, the comparison was easily broken into 3 factors: CPU performance, OS usability, and cost.

Table 3.19: *Touch-Display HMI CPU Part Comparison*
By Authors

Processor Type	CPU	OS	Cost
Windows/Intel	+	-	-
Linux/ARM	+	+	-
Raspbian/Raspberry Pi	+	+	+

Taking a look first at the OS for these HMIs, there are really two main choices: Windows and Linux (as Raspbian is part of the Linux family). As part of our software considerations, something that is open-sourced is favorable as documentation would be abundant when implementing and troubleshooting our design. This is available in a Linux based OS, whereas in a Windows OS this is not. Furthermore, having a Linux based OS would allow more fine-grained control to our system for optimization; something the Windows OS restricts greatly. So having a Linux based OS would be beneficial to our design. The choice in OS isn't the only reason why a Windows based HMI wasn't selected, however. Our main decision factor is the cost of the HMI, as our budget for such is $250. And with the budget we have made, a comparison was made for the Windows based HMIs. On Digikey, the cheapest Windows (HMI B) based was listed for $581, far exceeding our budget. Furthermore, the specifications of the CPU were very limited in relation to the cheapest HMI (HMI A) found on Digikey as shown in the table. HMI A has nearly twice the qualities of HMI B, while being less than half the cost. This trend is further exacerbated by the comparison of HMI A with HMI D. While maintaining similar CPU qualities HMI D is more than three times the price of HMI A. With having a poor cost to performance ratio and a second choice OS, Windows based HMIs were undesirable not considered.

Table 3.20: *Touch-Display HMI Part Comparison*
By Authors

Models	Cores	Clock	OS/CPU	Cost
ED-HMI3020-070C-04320-P (A)	4	2.4GHz	Raspbian/Raspberry Pi	$205
UTC-207GP-ATO0E (B)	2	1.1GHz	Windows/Intel	$581
CS12800PX101A (C)	4	1.5GHz	Linux/ARM	$249
UPOS-211DP-BST50 (D)	4	2.42GHz	Windows/Intel	$766

This leaves the Linux based HMIs and the Raspbian based HMIs. Since these two types of HMIs have similar operating systems, the next would be to compare the qualities of their CPUs. It just so happens that the Raspberry Pi on HMI A utilizes an ARM A76 processor, which is similar to that of the Linux based HMIs running their own ARM processors. However, this means that any HMI using the Raspberry Pi will have the same ARM A76, hence the same CPU characteristics as HMI A. Comparing HMI A to HMI C, it can be concluded that any Raspbian based HMI may outperform any Linux based HMI for less of a cost, making Raspbian based HMI the most cost-effective HMI without having to sacrifice performance. Despite the intended workload for the HMI to be relatively simple, it would be best for the part to be able to run the software efficiently.

Now with the several Raspbian based HMI, the three presented in the above table were taken into consideration for our design. To meet our design's specifications, certain elements for HMI were established. First would be power considerations. It was ideal that the HMI operated at a voltage of five volts DC. This is to maintain a lower power consumption of the overall system and it would be fairly simple to implement a five-volt regulator into the PCB design. Our next consideration would be the screen size of the HMI. Most of the HMIs within our budget range fall between five inches to around ten inches. With the smaller of the dimensions, implementing the desired UI made end up being too crowded or smaller when constrained to a smaller screen. So, a larger screen would be beneficial so that the UI can be easily seen and operated with. As for system storage, this is where a crossover with the intended software happens. For the software that may run on the HMI, the system requirements fall around to having ten to fifteen gigabytes of storage and at least four gigabytes of RAM memory.

Now where our selections differ are in the additional features that each may provide, which for the purposes of our design may end up not being utilized. These features include several I/O ports such as serial and audio but as well as PCIe and M.2 expansion slots. The features were prevalent in both the HMI3020 and the CM5. It was also further investigation that the CM5 actually had an operating voltage of fifteen volts DC. With its price exceeding our budget, the CM5 was not considered. This leaves the HMI3010 and the HMI3020. Now the HMI3010 is a lesser model of the 3020, found after the making of Table x.2, but still retains the core aspects of the 3020 that was found desirable for our design. Both the 3010 and the 3020 use the same Raspberry Pi 5 CPU and allow for the same level and size of RAM, four to eight gigabytes of DDR4 memory. They both as well meet the operating voltage and screen size parameters that we have established. Additionally, both systems meet the storage requirement that is projected for the software that will be used. The main difference between these HMIs is, again, the additional features mentioned previously which are seen in the 3020. The 3010 is also significantly more cost effective than the 3020 as the 3010 is three quarters the price of the 3020.

3.2.3 Microcontroller Unit

The Microchip SAMD21 series comprises low-power microcontrollers built on the 32-bit ARM Cortex-M0+ core, operating at up to 48 MHz. These MCUs are available in various pin configurations, notably the 24-pin (SAMD21E), 48-pin (SAMD21G), and 64-pin (SAMD21J) variants, each offering a balance between size and functionality.

The Microchip SAMD21 series offers a range of microcontrollers tailored to various application needs, differing primarily in pin count and I/O capabilities. The 24-pin SAMD21E is optimized for compact applications where space is limited. Despite its smaller footprint, it maintains core features such as up to 256 KB of Flash memory and 32 KB of SRAM. However, the reduced pin count limits the number of available I/O interfaces, which may constrain peripheral connectivity in more complex designs. The 48-pin SAMD21G serves as a middle ground, providing additional I/O pins compared to the SAMD21E, thus offering greater flexibility for interfacing with various peripherals. It retains the same core specifications, including the 48 MHz operating frequency and memory capacities, making it suitable for a broader range of applications that require moderate complexity. For applications demanding extensive peripheral interfacing, the 64-pin SAMD21J offers the highest number of I/O pins among the three. This variant supports up to 52 programmable I/O pins, accommodating complex designs with multiple peripherals. The larger package size, however, necessitates more PCB space, which should be considered during the design phase.

In terms of package types, the SAMD21 series offers various options, including TQFP, QFN, and WLCSP, providing designers with flexibility based on their specific requirements for size, thermal performance, and manufacturing considerations.

Table 3.21: *SAMD21 Part Comparison*
By Authors

Feature	**SAMD21E**	**SAMD21G**	**SAMD21J**
I/O Pins	20	38	52
ADC Channels	10	14	20
Package Types	TQFP, QFN	TQFP, QFN, WLCSP	TQFP, QFN, UFBGA
Thermal Resistance (θJA)	64.7°C/W (TQFP)	63.6°C/W (TQFP)	60.9°C/W (TQFP)

3.2.4 Motors & Motor Controller

With the criteria established, several candidate brands can be compared against what is needed. To simplify motor selection, the smallest motors will be examined and compared as the small motors will output the finer motions, while larger motors would not need to be as accurate in motion but rather have the ability to drive a potentially heavy load.
The following brands were chosen to compare:

- Hyulim Robot
- HobbyKing
- Beffkkip
- ROBOTIS

Hyulim Robot is a brand that offers several smart motors. Their smallest offering is the HerkuleX DRS-0101, which is a servo type motor. This motor takes an input voltage of 7.4VDC and has a stall torque of 12kgf.cm at rated voltage. It is 45mm x 24mm x 31mm. It has the ability to be integrated with programmable logic controllers and allows for PID tuning. It weighs a total of 45g.

While the DRS-0101 is not a bad option, there are several factors it falls short on when looking at the project needs. The motor takes an input voltage of 7.4VDC, which falls short of the 12V or 24V that would be needed for the project. Assuming the use of a 12VDC or 24VDC power supply, a resistor would need to be used just to power the DRS-0101. The cost of a single DRS-0101 is $70. This makes Hyulim Robot out of reach due to budget constraints. The DRS-0101 is their smallest motor, and larger motors cost upwards of $300.

HobbyKing offers a large selection of servo motors. The HXT900 is their smallest offering and takes anywhere between 3-6VDC. It provides 1.6kg-cm of torque, is 23mm x 12mm x 23mm, and weighs 9.8g. While its weight is a great benefit, the HXT900 lacks programmable logic controller integration as it does not contain any built-in circuitry that can communicate data. This also means that it lacks the ability to customize its acceleration, velocity, and PID tuning.

The Beffkkip SG90 is somewhat similar to the HobbyKing servo motor. It operates between 4.8V-6V, an abnormal choice of motor input voltages. It can provide 1.2-1.4 kg/cm of torque at 4.8V, measures at 31.8mm x 12.5mm x 32mm, and weighs 9g. Like the HobbyKing servo, the SG90 lacks PLC HMI integration and any parameter customization.

The last brand to compare is ROBOTIS. With ROBOTIS, they offer smart servos and steppers similar to those of Hyulim Robot. ROBOTIS' smallest, most affordable option is the DYNAMIXEL XL330-M077-T. It takes an input voltage of 5V that, albeit not 12V or 24V, is a standard control voltage that would be simple to supply. It provides 0.1075N of rated torque, measures 20mm x 34mm x 26mm, and is easily able to be integrated to a programmable logic controller and has customizable parameter tuning including PID, velocity, acceleration, and more. This option is much more affordable compared to Hyulim Robot as well, with the XL330-M077-T being priced at $23.90.

Ultimately, ROBOTIS is the chosen brand of motors that will be supplied in the final product. The team was able to secure a sponsorship with ROBOTIS, who provided several Dynamixel Servo motors to use for the project. As part of our sponsorship agreement, ROBOTIS provided the project team with the following motor equipment:

- DYNAMIXEL XC-330-T288-T (Qty: 2)
- DYNAMIXEL 2XL430-W250-T
- DYNAMIXEL XL430-W250-T (Qty: 2)

For this project, the animatronic motors will be utilizing TTL for communication. These will leverage TTL's straightforward and reliable signaling approach. The logic levels in TTL circuits are typically standardized, with voltages ranging around 5 V for logical "high" and close to 0 V for logical "low." This clear delineation between signal states ensures robust and error-resistant operation within animatronic control systems. While CAN provides excellent noise reduction, some of the motors we have chosen do not have CAN capability and would require additional work done to make them compatible which is added complexity and would be beyond this project's scope.

3.2.5 Manufacturing Materials

In selecting the appropriate filament for fabricating animatronic components, key factors such as strength, durability, ease of printing, and environmental resistance must be considered. The three most common FDM materials, PLA, PETG, and ABS, each offer different trade-offs.

PLA (Polylactic Acid) is widely used due to its ease of printing, low warping, and good surface finish. It is biodegradable and suitable for detailed parts with minimal stress or exposure to heat. However, PLA can become brittle over time and has a relatively low heat resistance, making it less ideal for high-stress or outdoor applications. For indoor, lightweight animatronic structures, PLA is a practical and efficient choice.

PETG (Polyethylene Terephthalate Glycol) offers greater strength and flexibility than PLA, along with better chemical and temperature resistance. It's less brittle and more durable, which makes it suitable for moving parts and components exposed to friction or mild environmental stress. PETG is also easy to print, though it may require tuning to avoid stringing.

ABS (Acrylonitrile Butadiene Styrene) is strong, impact-resistant, and heat-tolerant, often used in professional and industrial applications. However, ABS is more difficult to print due to warping and the need for an enclosed chamber. It also releases fumes during printing, requiring good ventilation. While suitable for rugged animatronic applications, its printing challenges make it less ideal for simpler projects.

Table 3.22: *Manufacturing Materials Comparison*
By Authors

Property	**PLA**	**PETG**	**ABS**
Strength	Moderate	High	High
Flexibility	Low (brittle)	Medium	Medium
Heat Resistance	Low (~60°C)	Moderate (~80°C)	High (~100°C)
Surface Finish	Smooth, detailed	Slightly glossy, less detailed	Matte, can be smoothed with acetone
Ideal Use Case	Prototypes, decorative, indoor	Functional parts, mild outdoors	Rugged use, high-stress parts

For this project, PLA is selected due to its ease of use, dimensional accuracy, and sufficient strength for lightweight, indoor animatronic components.

3.2.6 Sensors

When it comes to proximity sensors, there were two candidates:

- GoHz M12 Inductive Proximity Sensor Unshielded
- Twidec M12 Approach Sensor Inductive Proximity Switch

The GoHZ M12 proximity sensor includes 4 proximity sensors all using NPN configuration. It takes a standard supply voltage of 12-24 VDC and has a 4mm trigger distance. This sensor has a major drawback in its cable as it is unshielded. With motors involved, the figure can be heavily susceptible to noise interference. This can cause the figure to behave abnormally or to not work at all.

With the Twidec M12 prox sensor, it is configured to be a PNP sensor, and it has a supply voltage of 6-36VDC. It can detect objects up to 4mm, and the cable is shielded. This improves the noise immunity.

For limit detection, two limit switches were looked at:

- HiLetgo V-156-1C25 Lead Limit Switch SPDT
- Joinfworld Micro Limit Switch with Hinge Lever

The HiLetgo limit switch is a single-pole double-throw (SPDT) switch that supports up to 125 VDC. It measures to be 28mm x 10mm. This was the chosen sensor for the project due to cost and having a desired input voltage.

The Joinfworld Micro Limit Switch is rated for 12V DC. It supports both a Normally Open (NO) and Normally Closed (NC) contacts. It is also a Single Pole Double Throw switch and happens to be waterproof. This ability makes the limit switch more expensive than the prior.

For the photoelectric sensor:

- Taiss Photoelectric Sensor 30cm Adjustable M18.

This optical switch features a normal open and normally closed terminal and is configured to be NPN. It is important for the optical switch to be NPN. It was stated prior that in general there are no benefits to using either an NPN or PNP configuration, but that aspect has now changed. The optical sensor should be NPN as it will be serving a safety functionality by detecting if there is a foreign object in figure's base. If an object is detected by breaking the beam, the figure should stop. By having the sensor set up as NPN, if the sensor were to ever lose power it would cause an automatic system to stop by sending a high value, since the system is already active low.

All DYNAMIXEL servo motors used in this project have integrated drives that allow for finer control over the motor's movements and controls. This means that there is no need to use a hall effect sensor or an encoder. The DYNAMIXEL motors already include speed and velocity tracking as well as an encoder. This information is fed to the main control unit and the programmable logic controller. With the other sensors considered, there are more than enough redundant safety measures that justifies the lack of need for the hall effect and redundant encoder.

For the temperature sensor:

- WWZMDiB DS18B20 Temperature Sensor

It has a standard input supply voltage range of 3.0 - 5.25V. It comes with multiple probes so not just the figure can have a temperature record outside of the motors as they already record temperature, but rather that the enclosure can be kept cooled down enough.

An embedded controller supplied by DYNAMIXEL, named OpenRB-150, is the central hub for the motor power and communication. This chip allows for the interfacing between the computer and the motor for programming purposes. This will also be integrated into the final control system, to be controlled by the programmable logic controller.

DYNAMIXEL provides software to enable this granular level of control, which is appropriately named DYNAMIXEL Wizard. Once the software is installed, the motor can be set up by connecting a 3-pin included cable from the motor to the OpenRB-150. The controller is then connected to a computer using a USB-C to USB-C cable. Before programming, the motor must be scanned by the software. Each motor can be given a unique identifier (motor ID) ranging from 1 to 252 to individualize each motor, as well as motors having different baud rates used for communication. By default, all motors ship with a default motor ID of 1, with a varying baud rate. The XL330-M288 is able to be scanned and connected to by scanning ID 1 and having a set baud rate of 57600 bps.

Table 3.23: *Peripherals Part Comparison*
By Authors

Component	Specs	Pros	Cons
GoHz M12 Inductive (NPN)	12-24 VDC, 4mm range, unshielded	Standard voltage, 4 sensors included	Unshielded cable, noise issues
HiLetgo V-156-1C25 (SPDT)	Up to 125 VDC, 28×10mm	Cost-effective, compatible voltage	Not waterproof
Taiss M18 Photoelectric Sensor (NPN)	30 cm range, adjustable, NPN, NO/NC terminals	Safety feature: failsafe via active-low setup	Prone to false flagging, can be affected by environmental conditions

3.2.7 Peripherals

Within the Human-Machine Interface (HMI), the inclusion of a manual control element is essential for issuing final operator confirmations. This component must provide durability, clarity, and reliability under a range of electrical and environmental conditions. Several options were evaluated, including push buttons from Taiss, industrial switches from McMaster-Carr, and a selector switch from Allen-Bradley, the latter of which ultimately fulfilled all critical criteria.

***Table 3.24:** Peripherals Part Comparison*
By Authors

Component	Action Type	State Retention	Key Strengths
Taiss Push Button (19mm)	Momentary	No	Compact size, LED illumination, durable stainless-steel build
Taiss Push Button (22mm)	Momentary	No	Better tactile feedback than 19mm; otherwise, similar strengths
Allen-Bradley 800T-16HGH2KB6DX (30mm)	Maintained (2-position)	Yes	State retention, NO/NC contacts, robust construction, suited for harsh environments

The Taiss buttons, available in 19mm and 22mm diameters, offer LED illumination, stainless steel construction, and IP65 protection. These momentary-action switches return to their default state upon release, which limits their ability to retain manual commands. This makes them less suitable for functions requiring persistent state control, such as manual overrides. While the larger 22mm model provides improved tactile feedback, both models would require supplemental logic for state retention, adding complexity. Furthermore, their environmental sealing, while sufficient for general use, falls short of industrial-grade requirements.

In contrast, the Allen-Bradley 800T-16HGH2KB6DX selector switch provides a maintained two-position action, allowing it to physically retain its state without external logic. It supports both NO and NC contacts and includes an integrated 120VAC green LED for clear visual feedback. Rated at 10A and designed for 30mm mounting, the switch offers robust mechanical stability and simplified wiring flexibility.

Crucially, its IP66 rating ensures protection against dust and high-pressure water, making it appropriate for more demanding environments. The tactile selector mechanism also enhances user awareness, reducing the risk of unintended resets during maintenance or operation. Unlike momentary buttons, its maintained action offers clear state visibility and operational safety. Additionally, the switch's compatibility with standard industrial enclosures and its modular construction streamline integration. The inclusion of both status indication and control within a single device also reduces overall part count and panel complexity.

Considering all factors, mechanical retention, electrical versatility, environmental protection, and user clarity, the Allen-Bradley selector switch was selected as the final manual confirmation interface. Its industrial reliability and intuitive operation make it the most appropriate choice for this system's HMI.

3.2.8 Ethernet Port

To complete the Ethernet interface for the embedded system, the selection of a printed circuit board (PCB)-mountable RJ45 Ethernet jack was required. The connector had to provide reliable electrical performance, mechanical durability, and appropriate support for signal conditioning features such as integrated magnetics. Additional considerations included Power over Ethernet (PoE) compatibility, mechanical mounting style, LED indicators for link and activity status, and environmental resilience. Three RJ45 connectors were evaluated for this purpose: the Phoenix Contact 1337243, the HALO Electronics HCJ11-804SK-L12, and the Same Sky (formerly CUI Devices) CRJ048-LG-TH.

Table 3.25: *Ethernet PCB Port Part Comparison*
By Authors

Connector	**Integrated Magnetics**	**PoE Support**	**Strengths**
Phoenix Contact 1337243	Yes	Full PoE+ (IEEE 802.3at)	Industrial-grade robustness, EMI shielding, wide temp range
HALO HCJ11-804SK-L12	Yes	PoE (IEEE 802.3af)	Balanced features, simplified PCB layout, cost-effective, good diagnostics
Same Sky CRJ048-LG-TH	No	None	Low cost, compact form factor

The Phoenix Contact 1337243 offers a comprehensive feature set suitable for harsh industrial environments. It includes integrated magnetics, dual-color LED indicators, EMI shielding fingers, and full support for PoE+ (IEEE 802.3at). Its through-hole design provides strong mechanical stability, and its operating temperature range spans from -40°C to +85°C, making it ideal for field-deployed systems exposed to environmental extremes. However, the enhanced capabilities of this connector come at a higher price and a more complex PCB footprint. For embedded systems located in controlled environments, such as the interior of a control cabinet, these industrial-grade features may exceed what is necessary.

At the other end of the spectrum is the Same Sky CRJ048-LG-TH, a compact and cost-effective connector that provides basic shielding and LED indicators but lacks integrated magnetics. This omission requires external magnetics to be added to the board, increasing layout complexity and consuming additional PCB space. It also introduces potential signal integrity issues, particularly at gigabit speeds, due to longer trace routing and increased parasitic effects. While appropriate for budget-constrained applications, this part introduces design trade-offs that are undesirable in a tightly integrated, space-limited system.

Striking a middle ground, the HALO HCJ11-804SK-L12 includes integrated signal magnetics, green and yellow status LEDs, and standard 10/100/1000BASE-T compatibility. It supports IEEE 802.3af PoE, sufficient for the vast majority of embedded applications, while omitting support for higher-wattage PoE+ standards that are not needed in this design. The connector uses a through-hole mounting style for improved mechanical reliability and features a temperature range of 0°C to +70°C, appropriate for deployment inside climate-controlled panels or enclosures. Although it lacks the industrial robustness of the Phoenix Contact model, its feature set is more than adequate for this project's requirements and presents a cost-effective, integration-friendly solution.

By combining essential performance features with reduced cost and complexity, the HALO connector aligns well with the project's goals. Its integrated magnetics simplify PCB layout, reduce bill of materials (BOM) count, and improve long-term signal integrity. The inclusion of built-in status LEDs further supports rapid network diagnostics and monitoring during development and deployment.

In summary, while the Phoenix Contact Connector offers unmatched durability, and the Same Sky part provides an ultra-low-cost baseline, the HALO HCJ11-804SK-L12 delivers the optimal blend of functionality, manufacturability, and price. It is therefore selected as the Ethernet port for the embedded PCB design.

3.2.9 Ethernet Chip

As Ethernet was chosen for the system's communication infrastructure, an appropriate Ethernet controller needed to be integrated into the embedded design. This component is responsible for managing network data transfers between the host microcontroller and the broader network. The controller must handle TCP/IP protocols efficiently, offer reliable throughput, and interface cleanly with the selected microcontroller via a standard protocol such as SPI. To identify the best fit, three leading Ethernet chips were evaluated: the WIZnet W5500, the ASIX AX11001, and the Microchip ENC28J60. Each was assessed on protocol support, interface compatibility, memory architecture, data throughput, and ease of integration.

Table 3.26: *Ethernet IC Part Comparison*
By Authors

Chip	Protocol Support	Interface	Internal Buffer	TCP/IP Stack
WIZnet W5500	TCP, UDP, IPv4, ICMP, ARP, IGMP, PoE	SPI	32 KB (dedicated to Ethernet)	Full hardware-based stack
ASIX AX11001	TCP/IP	SPI, UART, I2C	32 KB + 128 KB Flash	Integrated in 8051 CPU
Microchip ENC28J60	None (relies on host MCU for stack)	SPI	8 KB	None

The WIZnet W5500 distinguishes itself as a high-performance hardwired TCP/IP controller designed specifically for SPI-interfaced embedded systems. It integrates a complete hardware-based TCP/IP stack along with a 10/100 Ethernet MAC/PHY and includes 32 KB of internal buffer memory. The full protocol stack, supporting TCP, UDP, IPv4, ICMP, ARP, IGMP, and PoE, is handled directly in silicon, offloading the host MCU and improving overall system responsiveness. This makes the W5500 ideal for communication-intensive applications, including those requiring multiple simultaneous connections. Its support for up to eight hardware sockets ensures robust performance in multitasking scenarios like real-time device control, diagnostics logging, or Modbus TCP communication.

In contrast, the ASIX AX11001 provides an all-in-one solution combining an 8051-based RISC processor with integrated Ethernet capabilities and a hardware-accelerated TCP/IP stack. While this eliminates the need for an external microcontroller, the trade-off is reduced flexibility. The internal CPU limits expansion possibilities, and the 8051 architecture poses compatibility challenges for systems already designed around modern 32-bit MCUs. Additionally, its embedded nature complicates modularity, making it less suitable for systems where logic control is already established on external processors.

The Microchip ENC28J60, meanwhile, offers a minimalist approach. It acts purely as an Ethernet MAC/PHY without a hardware TCP/IP stack, leaving all protocol handling to the host MCU. This significantly increases the processing burden on the microcontroller and consumes valuable memory resources, especially in applications requiring concurrent communication with multiple devices. With only 8 KB of internal buffer memory and a 20 MHz SPI interface, the ENC28J60 is constrained in both bandwidth and efficiency, making it ill-suited for real-time or high-load network environments.

The W5500’s superior data throughput is another decisive factor. It supports SPI clock rates up to 80 MHz, quadrupling the bandwidth available in the ENC28J60 and allowing high-speed data exchange even during frequent polling or packet bursts. This is especially valuable when managing a Modbus TCP connection with multiple endpoints, where performance consistency is essential. Furthermore, the W5500’s entire 32 KB buffer is dedicated solely to Ethernet traffic, unlike the AX11001, which must allocate its memory between code execution and data storage.

From a power standpoint, the W5500 includes efficient sleep and wake-on-LAN modes, making it suitable for both continuous and intermittent operation scenarios. These features align well with embedded systems designed for industrial automation, IoT monitoring, or distributed control, where power efficiency and reliability are both critical.

Considering all criteria, protocol handling, buffer memory, data transfer speed, power efficiency, and ease of integration, the WIZnet W5500 is the most well-rounded and technically robust option. It’s hardware-based protocol stack relieves the host microcontroller of unnecessary overhead, while its flexible SPI interface ensures compatibility with a wide range of embedded platforms. As a result, the W5500 is selected as the Ethernet controller for this project’s embedded communication system.

3.2.10 Programming IDE

When programming the SAMD21 microcontroller, developers can choose from several IDEs depending on their experience level, desired features, and project complexity. Common options include Arduino IDE, PlatformIO, and Atmel Studio.

Arduino IDE is the most beginner-friendly and widely used option for SAMD21 development. It offers a simple interface, extensive community support, and easy integration with a wide range of Arduino-compatible libraries. Its abstraction of hardware details allows rapid development and prototyping, making it an excellent choice for an animatronic project where development speed and simplicity are key. However, it lacks advanced debugging tools and deeper control over low-level hardware.

PlatformIO is a more advanced, modern IDE that runs as a plugin inside editors like VS Code. It supports multiple boards and frameworks, including Arduino, and provides powerful features such as integrated version control, task automation, and debugging tools. While it is more capable than the Arduino IDE, it has a steeper learning curve and may be overkill for simpler projects.

Atmel Studio (Microchip Studio) is the official IDE for programming SAMD21 using Atmel's low-level toolchain. It provides full control over the hardware, integrated debugging with Atmel ICE, and supports writing bare-metal C code. It is best suited for professional or complex embedded applications. However, it's Windows-only, relatively heavyweight, and not beginner-friendly.

Table 3.27: *Programming IDE Comparison*
By Authors

Chip	Debugging Tools	Library Ecosystem	Cross-platform	Ideal For
Arduino IDE	Limited (no built-in debugger)	Extensive (Arduino community)	Yes (Windows, Mac, Linux)	Prototyping, education, hobby projects
PlatformIO (VS Code)	Advanced debugging support	Uses Arduino & other libraries	Yes (Windows, Mac, Linux)	Advanced projects, automation, CI/CD
Atmel Studio / Microchip Studio	Integrated debugger (with ICE)	More limited to low-level APIs	Windows only	Professional, bare-metal, real-time apps

For this animatronic project, Arduino IDE is chosen due to its ease of use, compatibility with SAMD21 boards, and the availability of relevant libraries and examples for motor control and sensor integration.

3.3.11 Fuses

One of the biggest advantages of using fuses is their cost-effectiveness. They offer a simple yet highly reliable method of circuit protection, making them a widely used solution in electrical systems. In our project, we are incorporating a total of two 10A fuses. The first fuse will be positioned right after the 120VAC power supply, which connects directly to an outlet. This will serve as the primary safeguard against any overcurrent issues coming from the main power source. The second fuse will be placed after the 12VDC power supply, specifically to protect the PCB and its components from any potential overloads.

When it comes to selecting the right fuses, we are mainly looking for a model that features a push-button actuator for easy reset functionality, as well as compatibility with a 120VAC power source. This ensures convenience and reliability in case of a trip, allowing for quick restoration if we need to replace the fuse entirely. Below are the fuse options we are considering for this project.

We will be using 10A fuses throughout the design of the project to provide effective overcurrent protection for circuits that draw moderate amounts of current such as the motor drivers and actuators. A 10A rating offers a great balance between protecting the wiring and the components from too much current while allowing space for normal operating current.

3.2.12 Electrical Enclosures

When selecting an electrical enclosure, several key factors must be considered, including dimensions, price, durability, and ease of component installation. The first two options are from McMaster-Carr, a reputable supplier known for high-quality industrial components. The 18"x12"x4" enclosure is priced at $105.84, making it the most expensive option, and having a compact depth that may limit space for wiring and larger components such as inverters and relays. The second McMaster-Carr option, measuring 18"x18"x6", is larger and more spacious, while also being more cost-effective at $80.05. Its additional depth provides greater flexibility in mounting electrical components and ensuring proper airflow within the enclosure, reducing potential heat buildup, though not a main factor in our purchase decision. Both McMaster-Carr enclosures are expected to be well-constructed and suitable for industrial applications, ensuring a reliable and robust housing for the animatronics' electrical system.
The other two enclosures are generic brands and offer significantly larger internal volumes at competitive prices. The 20"x16"x8" enclosure, priced at $89.99, provides ample room for mounting the PLC, relays, and power distribution components, making it an attractive option in terms of the amount of room we have to work with. However, its increased size may present challenges in integration within the animatronic structure, requiring careful mounting considerations. The 16"x16"x8" option, costing $98.98, offers slightly less internal volume but maintains a deeper 8" depth, which can be beneficial for cable management and component accessibility. For each enclosure, factors such as heat dissipation and weather resistance are not primary concerns, as the electrical components will remain in a controlled indoor environment. Overall, the 18"x18"x6" McMaster-Carr enclosure appears to offer the best balance of cost, space, and reliability for housing the animatronics' control electronics.

***Table 3.28**: Electrical Enclosure Part Comparison*
By Authors

Ht, Wd, Dp.	Brand	Price
18x12x4"	McMaster Carr	$105.84
18x18x6"	McMaster Carr	$80.05
20x12x10"	VEVOR	$65.63

3.2.13 Ethernet Switch

With the functional criteria for the Ethernet switch clearly defined, namely, unmanaged configuration, non-PoE operation, fixed port count, 24V DC compatibility, and support for gigabit data rates, the final step was to evaluate and select a specific model that meets all design requirements. The system necessitates four Ethernet ports to service the Human-Machine Interface (HMI), the Programmable Logic Controller (PLC), the motor controller, and the programming interface. An additional fifth port was considered desirable for future expansion or redundancy, allowing for debugging, updates, or potential integration of additional sensors or logging tools.

Furthermore, to ensure seamless integration into the existing power infrastructure, the switch must operate at 24V DC, which is the standard voltage provided within the system's control cabinet. Compatibility at this level avoids the need for additional voltage regulators or DC-DC converters, both of which would introduce complexity and potential points of failure. In terms of performance, the switch must support data transmission at 1000 Mbps, ensuring no communication bottlenecks and maintaining full compatibility with the selected Cat 6 cabling standard.

Three switches were identified for comparison,

***Table 3.29**: Ethernet Switch Part Comparison*
By Authors

Brand	Operating Voltage	Transfer Speed	Cost
Centopto	12-48V	1000 Mbps	$36.99
GoodTop	5-60V	100 Mbps	$32.99
YuLinca	12-55V	100 Mbps	$29.99

Among the three options, the Centopto switch distinguished itself as the most suitable candidate. Supporting five ports, the switch satisfies the minimum required count while enabling future expansion. The voltage input range of 12-57V DC includes the required 24V DC mark, ensuring compatibility with our cabinet power supply. This wide input range also increases resilience to minor voltage fluctuations, which are common in industrial settings, thereby improving system stability. Most critically, the Centopto switch supports gigabit Ethernet, matching the performance of the selected Cat 6 cables and ensuring optimal data throughput between the system's components.

By contrast, the GoodTop and YuLinca switches both fall short of the required network performance. Despite having similar voltage compatibility and lower price points, both models are limited to 100 Mbps transfer rates, which introduces a hard cap on communication speed. This constraint could lead to data congestion during simultaneous communication between devices, reducing system responsiveness and potentially causing control delays, especially during high-frequency polling or feedback cycles involving the motor controller and HMI. These speed limitations directly conflict with the system's performance targets and therefore disqualify both options.

Furthermore, the Centopto model features a rugged DIN rail mounting configuration, which simplifies installation within the control cabinet and ensures mechanical security during operation. Its unmanaged nature requires no configuration, allowing it to be deployed immediately in a plug-and-play fashion. The lack of PoE capability further aligns with our system's design, avoiding unnecessary complexity and cost while preserving compatibility with the existing power layout.

Taking all of these elements into account, port count, voltage range, transfer speed, mechanical design, and ease of use, the Centopto switch offers the best balance of performance, reliability, and integration simplicity. Its selection ensures that the system's network backbone will operate without compromise, sustaining the real-time communication required for coordinated device operation and system monitoring.

3.2.14 Circuit Breakers

When choosing a specific circuit breaker to use for this project, a few things need to be taken into consideration. What is the rated current, the number of poles, the voltage rating, and the typical applications it is used for. An example of a high-quality industrial circuit breaker designed for precision protection of low current circuits would be the Allen-Bradley 1492-CB2-G010 circuit breaker. Rated at 1A and 480VAC, it is a double pole breaker, which means it can disconnect 2 hot wires at the same time, which is essential for our application which requires a high level of isolation and safety. Inside the breaker it features thermal magnetic tripping, which means it can detect both an overload of current and short circuits.

A similar option is the Siemens 5SY41 MCB. Like the Allen-Bradley model, it is rated at 1A and 480VAC and has thermal magnetic tripping built in. Siemens breakers are known for their quick response time to faults and their high breaking capacity, which makes them a strong candidate for consideration in our project. A third breaker to use would be the Schneider Electric Multi9 C60 2P 1A MCB. This model has technology built into it that helps to quickly stop arcs when it trips, which in turn protects the breaker from getting damaged and extends its lifetime.

***Table 3.30**: Circuit Breaker Part Comparisons*
By Authors

Specification	Allen-Bradley 1492-CB2 G010	Siemens 5SY41 MCB	Schneider Electric Multi9 C60
Rated Voltage/Current	480VAC/1A	480VAC/1A	480VAC/1A
Advantages	Compact and durable	Fast-Response	Arc quenching
Limitations	Limited Accessories	Bulky	High cost
Typical Application	Industrial Panels	Control Systems	Integrated Panels

3.2.15 PCB

JLCPCB is a popular manufacturing company based in China that has become widely known for offering affordable PCB fabrication services. JLCPCB is a service provider that makes it easy to go from a Gerber file to a physical circuit board. The platform offers services such as 2-6-layer PCBs, Surface Mount Technology, and even small-scale production. One of the biggest reasons to use JLCPCB is because of how cheap it is. Prototype boards can be ordered for just a few dollars, with fast international shipping. This could prove to be invaluable to us because shipping time is an important consideration when fabricating a PCB. Compared to other PCB manufacturers, such as PCBWay or OSH Park, JLCPCB stands out for its low cost and fast international shipping. It is important however to recognize the limitations with JLCPCB boards.

***Table 3.31**: PCB Manufacturing Part Comparisons*
By Authors

Feature	JLCPCB	PCBWay	OSH Park
Location	China	China	United States
Price (Relative to each other)	Low	Low to Moderate	High
Min. Order Quantity	5 Boards	5-10 Boards	No Minimum
Shipping Cost	Low	Low	Free
Strength	Cheapest option (On Average)	Good for low volume runs	Great for small, high-quality boards

3.2.16 Voltage Regulators

One of the most important aspects when choosing a voltage regulator is that is must be a DC-to-DC converter that is able to step down voltage from 12VDC to 5VDC. This conversion is crucial for safely powering out low-voltage logic components such as the MCU and the sensors. To meet these requirements, we have evaluated several voltage regulators, primarily focusing on buck converters because of their high efficiency and suitability for stepping down voltage while also maintaining high levels of efficiency.

A candidate we considered was the TPS564257. It is a synchronous step-down converter with a wide input range (3V to 17V) and can deliver up to 4A of output current which is more than what is needed for our systems 5V rail. Another great feature about it is that it is very efficient, with about a 93.9% efficiency rating. Another main benefit it has is that the IC cost is low, costing about $0.14 in bulk. The total BOM cost for this regulator is $0.72, offering excellent value for the performance it gives. In comparison, the TPS62142 is a high-efficiency, low-noise synchronous buck converter that has an output current of 2A. This converter is also a solid option, although it is slightly more expensive than the TPS564257. \ On the other hand, the LM2679 and LM2576 are designs that support a wider set of input voltage ranges and are built for slightly heavier-duty industrial applications. The LM2679 supports inputs from 1.2V to 37V and can deliver 5A of output current, which is great for powering the motor and actuator systems. The biggest drawback for this, however, is the obvious higher cost and decreased efficiency. The LM2576 offers a 7V-40V range and 3A of output current. The LM2576 offers proven reliability, wide input voltage support, and a robust 3A output capacity, which makes it perfectly suited for handling the motor drivers and logic circuits in our animatronic. Its simple design, fewer external components, and durability make it a strong choice for Senior Design 2. The LM2576 is very tolerant to wide variations in input voltage and design mistakes don’t affect it as much, which adds an extra layer of safety and robustness to the overall design.

***Table 3.32**: Circuit Breaker Part Comparisons*
By Authors

Regulator	**Voltage Range**	**Output Current**	**Efficiency**	**Cost**	**Topology**
TPS564257	3V-17V	4A	93.9%	$0.72	Buck
TPS62142	3V-17V	2A	88.8%	$0.93	Buck
LM2679	8V-40V	5A	84.7%	$4.22	Buck
LM2576	4V-40V	3A	82.7%	$1.79	Buck

Chapter 4: Standards and Design Constraints

Industrial automation and animatronic control systems must adhere to a variety of engineering standards and design constraints to ensure safety, reliability, and performance. These standards govern electrical safety, control system design, software architecture, and environmental considerations, ensuring compliance with regulatory guidelines and industry best practices. Additionally, various design constraints, such as power limitations, environmental conditions, and system scalability, must be addressed to optimize system functionality.

4.1 Industrial Standards:

Industrial standards are authoritative guidelines and specifications established to ensure the safety, reliability, and efficiency of engineering systems across various industries. These standards encompass technical definitions, rules, and procedures that promote uniformity and compatibility among components and processes. By adhering to these standards, engineers can design systems that meet consistent performance criteria, facilitate interoperability, and comply with regulatory requirements.

4.1.1 Modbus

Modbus is a widely used communication protocol in industrial automation systems, enabling communication between devices such as PLCs, HMIs, and sensors. It is particularly relevant to this project due to its simplicity, reliability, and open-source nature, which aligns with the goal of avoiding proprietary ecosystems. Modbus operates over serial (RS-232/RS-485) or Ethernet (Modbus TCP/IP) connections, making it versatile for various applications.

Modbus was developed in 1979 by Modicon (now part of Schneider Electric) as a simple and robust communication protocol for industrial automation systems. It was initially designed to enable communication between Modicon PLCs and other devices, such as HMIs and sensors, over serial communication lines (RS-232/RS-485). The protocol’s simplicity and open nature quickly made it a de facto standard in the industry, leading to its widespread adoption across various manufacturers and applications.

In the 1990s, with the rise of Ethernet in industrial environments, Modbus was adapted to support TCP/IP, resulting in Modbus TCP/IP. This version retained the core principles of the original protocol while enabling faster and more flexible communication over Ethernet networks. Today, Modbus remains one of the most widely used protocols in industrial automation due to its reliability, ease of implementation, and compatibility with a wide range of devices.
In the context of the animatronic system, Modbus TCP/IP will be utilized to facilitate communication between the HMI and the PLC. This protocol allows the

HMI to both read system status and write commands to the PLC, enabling bidirectional control. For example, the HMI can send commands to execute pre-determined control profiles or modify system parameters in real-time. Additionally, Modbus supports fault diagnostics by allowing the PLC to transmit error codes and status updates to the HMI, ensuring quick identification and resolution of issues.

The use of Modbus also supports the system's scalability, as additional devices (e.g., sensors, actuators) can be integrated into the network without significant reconfiguration. However, it is important to note that Modbus lacks built-in security features, which necessitates the implementation of additional measures, such as network segmentation and firewalls, to protect the system from unauthorized access.

4.1.2 IEC 61131-3 (PLC Programming Standards)

IEC 61131-3 is an international standard for programming programmable logic controllers (PLCs). It defines five programming languages: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL), and Sequential Function Chart (SFC). This standard ensures consistency and portability of PLC programs across different manufacturers and platforms, which is critical for the animatronic system's design.

The International Electrotechnical Commission (IEC) was founded in 1906 as a global organization responsible for developing and publishing international standards for electrical, electronic, and related technologies. The IEC's work has been instrumental in promoting interoperability, safety, and efficiency in various industries, including industrial automation.

IEC 61131-3, first published in 1993, is part of the IEC 61131 series of standards for programmable logic controllers (PLCs). It was developed to address the lack of standardization in PLC programming, which at the time varied significantly between manufacturers. The standard defines five programming languages: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL), and Sequential Function Chart (SFC). These languages provide engineers with a range of tools to develop control logic, depending on the complexity and requirements of the application.

Over the years, IEC 61131-3 has become the cornerstone of PLC programming, ensuring consistency and portability across different platforms. Its adoption has been further reinforced by the development of integrated development environments (IDEs) that support all five languages, such as Siemens TIA Portal and CODESYS.

For this project, Structured Text (ST) and Ladder Diagram (LD) will be the primary programming languages. ST will be used for complex control algorithms, such as fault detection and predictive maintenance, due to its flexibility and similarity to high-level programming languages. LD, on the other hand, will be employed for simpler logic, such as motor control and emergency stop functionality, because of its intuitive graphical representation.

The adoption of IEC 61131-3 also supports the system's modularity and scalability. For instance, reusable function blocks can be created for common tasks, such as motor control or fault handling, reducing development time and ensuring consistency across different parts of the system. Furthermore, compliance with this standard ensures that the system can be easily maintained or upgraded by engineers familiar with industry-standard practices.

4.1.3 Ethernet/IP Communication Protocol

Ethernet/IP (Industrial Protocol) is a widely used communication protocol in industrial automation, developed by Rockwell Automation and now managed by the Open DeviceNet Vendors Association (ODVA). It was introduced in the early 2000s as an extension of the standard Ethernet protocol, specifically designed for industrial applications. Ethernet/IP leverages the Common Industrial Protocol (CIP), which provides a unified communication framework for industrial devices, including PLCs, HMIs, servo drives, and vision systems.

Unlike Modbus TCP/IP, which is primarily focused on simple data exchange, Ethernet/IP supports both implicit (real-time) and explicit (message-based) communication. This makes it suitable for applications requiring high-speed data transfer and precise synchronization, such as motion control and vision systems. Ethernet/IP operates over standard Ethernet networks, allowing it to integrate seamlessly with existing IT infrastructure while providing the robustness and determinism required for industrial environments.

Key Features:

- Real-Time Communication: Ethernet/IP supports real-time data exchange through its implicit messaging capability, which is essential for applications like servo control and synchronized motion.
- Scalability: The protocol can handle large networks with hundreds of devices, making it ideal for complex systems with multiple axes of movement or distributed I/O.
- Interoperability: As part of the CIP family, Ethernet/IP ensures compatibility with a wide range of devices from different manufacturers, promoting flexibility and reducing vendor lock-in.
- Advanced Diagnostics: Ethernet/IP provides detailed diagnostic information, enabling quick identification and resolution of issues, which aligns with the project's fault protection and diagnostics goals.

While Modbus TCP/IP is the primary communication protocol for the animatronic system, Ethernet/IP may be used for specific applications requiring higher-speed communication and real-time performance. For example:

- Servo Drives: Ethernet/IP can be used to control servo motors with high precision, ensuring smooth and synchronized movements across multiple axes.
- Vision Systems: If the system incorporates vision-based feedback (e.g., for object detection or tracking), Ethernet/IP can facilitate fast data transfer between cameras and the PLC.
- Advanced Diagnostics: Ethernet/IP's diagnostic capabilities can enhance the system's fault detection and predictive maintenance features, providing detailed insights into device status and performance.

Implementation Considerations:

- Network Infrastructure: Ethernet/IP requires a robust network infrastructure, including industrial-grade switches and cables, to ensure reliable communication in harsh environments.
- Configuration Tools: Rockwell Automation's Studio 5000 or other CIP-compatible software tools will be needed to configure and manage Ethernet/IP devices.
- Cost: While Ethernet/IP offers advanced features, it may involve higher costs compared to Modbus TCP/IP, particularly for licensing and hardware.

4.2 Design Constraints:

Industrial standards are established protocols and guidelines that ensure safety, reliability, and interoperability within industrial automation systems. These standards encompass various aspects, including electrical safety, control system design, software architecture, and environmental considerations. By adhering to these standards, engineers and developers can create systems that comply with regulatory requirements and align with industry's best practices. Standards such as IEC 60204-1, IEC 61508, and IEC 61131-3 provide frameworks for designing and implementing safe and efficient control systems. Compliance with these standards not only enhances system performance but also facilitates integration with existing infrastructure and technologies. Ultimately, industrial standards serve as a foundation for developing robust and scalable automation solutions.

4.2.1 Cost-Effectiveness

Cost-effectiveness is a critical design constraint for the animatronic system, as it directly impacts the feasibility and accessibility of the solution. The goal is to minimize costs without compromising reliability, performance, or safety. This constraint influences decisions at every stage of the design process, from component selection to system architecture.

High-end components, such as servo motors from manufacturers like ABB or Schneider Electric, offer superior precision and performance but come at a significant cost. To balance performance and affordability, mid-range motors with adequate precision and torque will be selected. These motors provide sufficient performance for the animatronic system's requirements while keeping costs manageable. Similarly, the choice of PLCs and HMIs will prioritize cost-effective yet reliable options, such as Siemens S7-1200 or Allen-Bradley Micro800 series, which offer robust functionality at a lower price point compared to high-end models.

The use of open-source communication protocols like Modbus TCP/IP reduces licensing fees and avoids vendor lock-in, which is common with proprietary protocols. This not only lowers initial costs but also ensures flexibility for future expansions or integrations with third-party devices. However, cost-effectiveness must not compromise functionality. For example, while Ethernet/IP may involve higher costs due to licensing and hardware requirements, its advanced features (e.g., real-time communication) may justify the investment for specific applications, such as high-speed servo control or vision systems.

The system architecture must be designed to minimize costs while maintaining reliability. For instance, redundant power supplies and advanced fault detection systems may increase initial costs but are essential for ensuring reliable operation and minimizing long-term maintenance expenses. Predictive maintenance features, such as real-time monitoring of motor health, can reduce downtime and repair costs, further enhancing cost-effectiveness over the system's lifecycle.

Achieving cost-effectiveness often involves trade-offs. For example, while mid-range motors may not offer the same level of precision as high-end models, their performance is sufficient for the animatronic system's requirements. Similarly, the use of open-source protocols may require additional security measures, such as network segmentation, to mitigate the lack of built-in security features. These trade-offs are carefully considered to ensure that cost savings do not compromise the system's overall functionality or safety.

4.2.2 Scalability and Modularity

Scalability and modularity are essential design constraints to ensure that the animatronic system can adapt to future requirements and expansions. The system must be designed to accommodate additional axes of movement, enhanced features like audio/visual effects, or integration with new technologies without requiring significant reconfiguration or redesign.

A modular architecture is key to achieving scalability. For example, the PLC program will be structured using reusable function blocks, which can be easily modified or expanded to accommodate new functionalities. This approach reduces development time and ensures consistency across different parts of the system. Similarly, the HMI will support dynamic updates to its interface, allowing operators to add new controls or displays as needed.

Scalability also impacts the selection of communication protocols. Ethernet-based protocols like Modbus TCP/IP and Ethernet/IP are preferred over serial communication due to their ability to handle larger networks and higher data throughput. These protocols support the integration of additional devices, such as sensors, actuators, or vision systems, without requiring significant changes to the existing network infrastructure.

The power distribution system must be designed with spare capacity to support future expansions. For example, the power supply and distribution boards will include additional terminals or circuits to accommodate new motors or sensors. This ensures that the system can be expanded without requiring a complete overhaul of the power infrastructure.

The system’s expandability is further enhanced by its modular design. For instance, additional axes of movement can be added by integrating new servo motors and drives, while audio/visual effects can be incorporated using DMX lighting controllers and audio processors. The use of standardized components and protocols ensures that these expansions can be implemented quickly and cost-effectively.

4.2.3 Real-Time Performance

Real-time performance is a critical constraint for the animatronic system, as it directly impacts the precision and reliability of its movements and fault response capabilities. The system must operate with precise timing to ensure synchronized movements across multiple axes and respond to faults within the required timeframe of less than 1 second.

Achieving real-time performance requires the use of deterministic hardware, such as PLCs with real-time operating systems (RTOS). These PLCs ensure that control logic is executed with precise timing, minimizing delays and jitters. High-speed sensors and encoders are also essential for providing accurate feedback on motor position and speed, enabling the system to maintain precise control over its movements.

The choice of communication protocols plays a significant role in real-time performance. Ethernet-based protocols like Modbus TCP/IP and Ethernet/IP are preferred for their low latency and high data throughput. Ethernet/IP, in particular, supports implicit (real-time) messaging, which is essential for applications requiring high-speed data transfer and synchronization, such as servo control.

Real-time performance is especially critical for the fault detection system. The system must continuously monitor critical parameters, such as motor current, position, and temperature, and immediately trigger corrective actions when anomalies are detected. This requires dedicated processing units for fault handling, as well as high-speed communication between sensors, PLCs, and the HMI.

For applications involving multiple axes of movement, such as animatronic figures, precise synchronization is essential to ensure smooth and lifelike motions. This is achieved through real-time control algorithms and high-speed communication between the PLC and servo drives. The use of synchronized motion controllers, such as those from Delta or Yaskawa, further enhances the system's ability to coordinate complex movements.

4.2.4 Time Constraints

The development of the animatronic system is subject to a strict timeline, with only two academic semesters for design and planning, followed by a summer semester for building and testing. This compressed schedule presents several challenges that must be carefully managed to ensure the project's success.

During the first semester, the focus is on conceptual design, component selection, and system architecture. This phase involves extensive research, simulation, and prototyping to validate design choices and ensure feasibility. However, the limited time frame requires efficient prioritization of tasks and milestones. For example, critical components such as the PLC, HMI, and motors must be selected early to allow sufficient time for integration and testing.

The summer semester is dedicated to building the physical system and conducting rigorous testing. This phase includes assembling mechanical components, wiring electrical systems, and programming the PLC and HMI. Testing is critical to identify and resolve any issues before the system is finalized. However, the tight schedule leaves little room for unexpected delays, such as component shortages or design flaws. To mitigate this risk, contingency plans are developed, and critical tasks are prioritized to ensure that the system is operational by the end of the summer.

Strategies to Manage Time Constraints:

1. Parallel Development: Tasks are divided into parallel streams to maximize efficiency. For example, while the mechanical team works on assembling the animatronic structure, the electrical team can focus on wiring and

power distribution, and the software team can develop the PLC and HMI programs.

2. Rapid Prototyping: Quick prototypes are built to test key concepts and identify potential issues early in the design process. This reduces the risk of costly redesigns during the building phase.
3. Milestone-Based Planning: The project is divided into clear milestones with specific deliverables and deadlines. Regular progress reviews ensure that the project stays on track and any issues are addressed promptly.
4. Resource Allocation: Resources, including time, budget, and personnel, are allocated based on priority. Critical tasks, such as motor control and fault detection, receive the highest priority to ensure that the system's core functionality is achieved within the timeline.
5. Modular Design Approach: The system is designed in self-contained modules that can be developed, tested, and refined independently. This allows teams to work concurrently on different components and makes integration more efficient, reducing bottlenecks and facilitating easier troubleshooting.

4.2.5 Environmental Constraints

The development of the animatronic system is also influenced by environmental constraints, including limited access to advanced manufacturing technologies and materials. These constraints impact the design and fabrication of mechanical components, as well as the overall system architecture.

The project team does not have access to large-scale CNC machining or advanced fabrication technologies, such as 3D printing with high-performance materials. This limits the ability to produce complex or highly precise components in-house. To address this constraint, the design prioritizes simplicity and the use of off-the-shelf components wherever possible. For example, standard aluminum extrusions and prefabricated brackets are used for the animatronic structure, reducing the need for custom machining.

Silicone molding and other advanced material processing techniques are not available for creating life like animatronic skins or flexible components. As a result, the design focuses on mechanical functionality rather than aesthetic realism. For example, the animatronic figure may use rigid materials for its exterior, with movements designed to convey expressiveness through mechanical means rather than lifelike textures.

The project is developed in an academic setting, which may lack the specialized tools and facilities found in industrial environments. For example, testing facilities for high-precision motion control or environmental stress testing may be limited. To overcome this, the team leverages simulation software to validate designs and conducts incremental testing to identify and address issues early.

Strategies to Overcome Environmental Constraints:

1. Simplified Design: The system is designed to minimize the need for complex or custom components, relying instead on readily available parts and materials.
2. Collaboration with External Partners: Partnerships with local manufacturers or makerspaces are explored to access advanced fabrication technologies when necessary.
3. Creative Solutions: The team employs creative problem-solving to achieve desired functionality within the constraints. For example, 3D-printed components may be used for prototyping, even if they are not suitable for the final product.
4. Focus on Core Functionality: The design prioritizes the system's core functionality, such as precise motion control and fault detection, over aesthetic or non-essential features.

4.2.6 Significant PCB Design

The nature of this project requires a significant Printed Circuit Board (PCB) design. Prior to specifying the design in detail, it is important to note the constraints that need to be met. Per the significant PCB design guideline, the design of the PCB must adhere to the following requirements:

- Contain at least:
  - one (1) Main Controller Unit (MCU)
  - a power supply unit (PSU)
  - several peripherals integrated with the MCU

- Cannot contain:
  - Parts smaller than 0805
  - Solder-less prototyping board
  - Multiple development boards

Another project guideline point relates to the complexity of the functionality for the PCB. The guideline states that students are not expected to design PCBs that are used for advanced or complex tasks. Examples given are voltage regulation, motor driving, AD/DA conversion, etc. Rather, we are required to use existing circuit designs that can fulfill the functionality.

4.2.7 Manufacturing

The chosen manufacturer for the project is JLC PCB. Therefore, JLC PCB's standards apply and are taken into consideration in the PCB design process. With JLC PCB, their capabilities include the production of 1-32 layers per PCB, which we will only need 2 of. With a 2-layer board, JLC PCB can produce PCBs with a minimum dimension of 3x3mm and a maximum of 670 x 600mm FR4 material PCB, with a tolerance of +/- 0-.1%. FR4 PCBs can be 0.4/0.6/0.8/1.0/1.2/1.6/2.0 mm thick for up to 12 layers, and a tolerance of +/- 10%. The holes on the PCB can be drilled anywhere between 0.15-6.3mm. Through-holes have a tolerance of +0.13 / -0.08mm and press fit can be +/- 0.05mm. Minimum via hole size/diameter is 0.15mm / 0.25mm. For the traces, they can be a minimum of 0.10mm wide and 0.10mm spaced out at 1 oz; 0.16 / 0.16mm at 2 oz with +/- 20% tolerance.

4.2.8 Design Practicality

While the animatronic is developed using a set of mechanical drawings, structured control logic, and detailed digital models, the transition from conceptual design to physical implementation introduces the possibility of unforeseen challenges. Despite careful planning, real-world constraints such as manufacturing tolerances, mechanical flexing, and the non-linear behavior of both the materials used and the motors can significantly impact the system's actual performance.

One of the primary concerns is the potential for unexpected collisions or misalignments between moving parts - especially in complex systems like an animatronic, where multiple degrees of freedom must operate in close coordination. For example, an axis that appears unobstructed in a CAD model might, in a practical real-world setting, intersect with another axis due to slight inaccuracies in assembly or unaccounted movement dynamics. These issues are often not apparent until the system is physically constructed and tested under operational conditions.

To address these challenges, the design must incorporate both hardware and software solutions. On the hardware side, adjustable mounts, physical travel limits, and modular assemblies can help compensate for mechanical variability. On the software side, having a “soft stop” so each function cannot reach its hard stops, real-time sensor feedback, and having a proper profile can help ensure safe and accurate actuation during operation. In some cases, reprogramming motion profiles or reassigning control logic might be necessary to maintain fluid movement while avoiding unintended interactions.

The possibility of these design constraints emphasizes the importance of an iterative development approach. Prototyping, testing, and debugging are not just steps in the workflow but are essential to refining the design and ensuring it performs reliably in real-world conditions. This iterative process also demands flexibility and a willingness to revisit assumptions made during the initial design phase. Remaining adaptive and receptive to feedback allows the team to balance both engineering functionality and creative expression, something that is crucial to a multidisciplinary project like animatronics in the career setting, where form and function must align seamlessly.

Chapter 5: Comparison of ChatGPT with other similar platforms

In this chapter, we will first dive into the potential benefits as well as the disadvantages of using AI software such as ChatGPT. We will take a deep dive into how ChatGPT works fundamentally and compare it with other AI tools such as Claude, Microsoft Copilot, Gemini, as well as other Large Language Models (LLMs).

5.1 Introduction:

The use of ChatGPT in our animatronic project could prove to be very useful in providing general knowledge while also functioning as a learning assistant. ChatGPT is very useful for understanding the concepts of how our project will work at the end. While not perfect, ChatGPT might also be able to assist in things such as power calculations, or just complex algebra that would be done faster using AI instead of software like Symbolab for example. We will first discuss what ChatGPT is and how it works, as well as some of its pros and cons. We will then delve into other tools similar to ChatGPT and discuss which LLM would be better suited for our project.

5.2 What is ChatGPT and how does it work?:

ChatGPT works in basically 5 steps:

1. Training on Massive Data Size
2. Understanding Natural Language Processing (NLP)
3. Generates a Response
4. Filters Responses
5. Continuous Learning

ChatGPT first needs to be trained on a large amount of data, including information from websites, books, articles, online forums, Wikipedia, and more. This helps the model recognize patterns in human communication. During the Pre-Training Phase, ChatGPT analyzes trillions of words, known as “tokens,” from these datasets. The model uses a transformer architecture, which allows it to learn the relationships between words, sentences, and ideas rather than memorizing specific facts. Using Natural Language Processing (NLP) techniques, it breaks down user inputs to understand context and identify the most relevant information. The first step in NLP is tokenization, where a message is divided into smaller parts, such as words or phrases, enabling the model to process and generate responses effectively. Additionally, ChatGPT considers the entire conversation history to maintain context and coherence.

To ensure the responses are helpful and safe, ChatGPT undergoes fine-tuning to avoid generating harmful, biased, or inappropriate content. For example, it is trained to refrain from answering questions like “What’s my address?” or addressing illegal topics. While ChatGPT is continuously learning, it does not do so in real-time. Instead, it learns in phases whenever OpenAI updates the model. These updates often involve Reinforcement Learning from Human Feedback (RLHF), where AI trainers manually review, and rank responses based on criteria like accuracy, clarity, and helpfulness. This process helps refine the model’s performance over time, ensuring it provides high-quality and reliable answers.

5.2.1 Pros

There are some electrical components we need to use for this project that we have not directly learned about in lectures. A relay is a good example of this. If someone wanted to find out more about relays they might do a quick google search and look up more information about that. An alternative is to understand the concept of relays using ChatGPT, which could in turn help us make better project design choices.

Example #1:

“Define simply what a relay is and explain it in a few sentences.”

Output #1:

“A relay is an electrically operated switch that allows a small electrical signal to control a larger electrical load. It works by using an electromagnet to open or close a set of contacts, turning a circuit on or off. This allows low-power devices, like sensors or microcontrollers, to safely control high-power devices such as motors, lights, or heaters. Relays are commonly used in automation, electrical protection, and remote-control systems.”

This is a fast and simple way to understand clearly what a transistor is. We can see that a relay works by using an electromagnet to act as a switch. You could have also easily googled “What is a relay” and got an answer like: “A relay is an electrically operated switch that connects and disconnects circuits. Relays are used in many applications, including automotive, industrial, and telecommunications.” However, using ChatGPT, we were able to learn more of the concept behind how relays actually work.

Regardless of which LLM (Large Language Model) is used, AI has multiple benefits, including but not limited to:

1. Brainstorming and idea generation
2. Assist in debugging code
3. Mathematical assistance

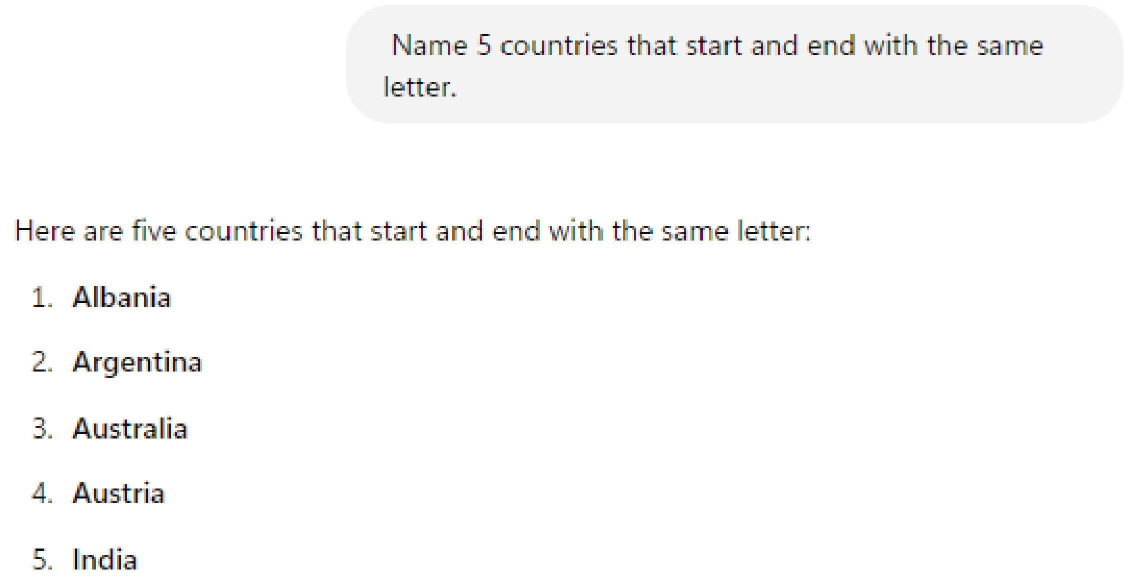
5.2.2 Cons

1. Lack of real-time learning and adaptation
2. Potentially outdated information
3. Lack of Creativity
4. Cannot do complex engineering simulations
5. Can lead to weak critical thinking

While ChatGPT has numerous benefits, it is also important to list the cons in using it. ChatGPT lacks real-time learning and adaptation. This means it does not learn in real time and therefore cannot adjust its responses based on new data published online. Troubleshooting, testing, and design alterations can sometimes need to rely on new data or even just real-time simulation results (e.g., MATLAB, LTSpice, etc.) which ChatGPT cannot do. This can also lead to ChatGPT giving incorrect or outdated responses. Another issue less frequently discussed is that relying too much on ChatGPT can lead to weak critical thinking and even a lack of creativity. During this project we will need to use strong analytical and problem-solving skills and using ChatGPT too much may give a shortcut for solving problems without fully understanding the reasoning behind the solutions. Not only does this make the student lack creativity in solving engineering problems that may arise, but also it takes away the understanding of fundamental engineering concepts which are useful for real-world challenges.

ChatGPT is far from perfect and could lead to harmful consequences. Many times, ChatGPT can give false positives. To demonstrate a simple case of this, I will ask ChatGPT, "Name 5 countries that start and end with the same letter."

***Table 5.1:** ChatGPT Error Example*
By Authors



Name 5 countries that start and end with the same letter.

Here are five countries that start and end with the same letter:

1. **Albania**
2. **Argentina**
3. **Australia**
4. **Austria**
5. **India**

As we can see, a simple question like this ChatGPT can get wrong. It got the first 4 countries correct but missed the last one. ChatGPT relies on patterns in data rather than direct comprehension and so it is possible that it could flag useful information as problematic or confidently provide incorrect answers. When using ChatGPT or other AI platforms, it is important to cross check its response with reliable sources and understand that even though ChatGPT is a powerful tool, it is not infallible.

5.3 Other AI Platforms:

5.3.1 Claude

Claude AI is similar to ChatGPT in things such as natural, text-based conversations, summarizations, decision-making, etc. However, Claude AI would be recommended to use for creative work. This is why Claude is considered a more helpful coding assistant. Claude AI and ChatGPT also have a significantly fundamental difference in their training philosophy as well. Whereas Claude AI follows a set of predefined ethical principles encoded in its responses, ChatGPT uses Reinforcement Learning from Human Feedback (RLHF), meaning it adapts based on user feedback.

5.3.2 Microsoft Copilot

Microsoft Copilot’s main difference with ChatGPT is that Copilot has real-time internet access, whereas ChatGPT relies on pre-trained data, which may not always be current or correct. Another but less significant difference in the two platforms is that Copilot is less creative than ChatGPT, meaning if coding support is needed, ChatGPT would be the better option to use. However, during our project, if real-time information and fact-checking were needed, Microsoft Copilot would be the best source to use in that situation. A major distinction between Microsoft Copilot and ChatGPT lies in their ability to access information. Copilot offers real-time internet access, enabling it to retrieve the most current data directly from the web. In contrast, ChatGPT relies on a pre-trained dataset, which, while extensive, may not always reflect the most up-to-date or accurate information.. This makes Copilot a more dependable choice when real-time data (such as the latest technical standards, component data sheets, or cutting-edge industry research) is needed. Another difference, though less significant, is their approach to creativity. ChatGPT generally excels in generating innovative ideas, making it ideal for brainstorming, refining code, or exploring unconventional solutions. On the other hand, Copilot is better suited for delivering precise, factual, and current information. For our senior design project, selecting between these two tools should depend on the specific task. Microsoft Copilot is invaluable for obtaining accurate and up-to-the-minute information for research, fact-checking, or ensuring compliance with industry standards. Meanwhile, ChatGPT’s strengths lie in tasks such as debugging code, crafting complex scripts, or brainstorming alternative designs. By utilizing the unique strengths of each platform, we can maximize efficiency and stay well-informed throughout the project.

5.3.3 Gemini

Gemini is similar to Copilot in the sense that it has real-time web access. Gemini is always using Google to search for its information though, whereas ChatGPT uses Google only in browsing mode. Gemini is also less creative than ChatGPT, but it is still a strong contender. Overall, Gemini is better for live fact-checking and for analyzing images and videos, while ChatGPT is stronger for creativity and coding.

5.3.4 Perplexity

Perplexity AI stands out as a fundamentally different tool from ChatGPT in how it approaches generating responses. One of the most significant distinctions is Perplexity AI's accountability in its outputs. This platform consistently provides sources and citations for the information it shares. For every response, users have the ability to verify the origins of the data and check the credibility of the information, ensuring a higher level of trust in its outputs. This feature is especially valuable in situations where precision and reliability are critical. Perplexity is a more suitable choice for academic inquiries where conducting in-depth research is needed.

5.3.5 Pi

Pi AI is also very different from ChatGPT in its responses. Pi AI's primary focus is on emotional intelligence and friendly conversation. It is built on an LLM called Inflection-1 and it uses similar language processing techniques as ChatGPT such as NLP, deep learning, and reinforcement training. Pi AI is more tailored to a personalized dialogue which makes it an inadequate tool in conducting research.

5.3.6 AlphaCode

AlphaCode and ChatGPT are both LLM's developed for distinct purposes, but as with every LLM AI, they do share some foundational similarities. AlphaCode is specifically designed to excel in competitive programming and code generation. The main focus is to solve coding challenges and output efficient code. ChatGPT on the other hand is tailored more towards NLP tasks such as conversational dialogue and generating ideas. In essence, AlphaCode is very narrow in its versatility whilst ChatGPT is more versatile, having a broader range of applications other than just coding.

5.3.7 Jasper

Jasper AI and ChatGPT are both LLMs designed for generating human-like text, but they do have some differences to them. Jasper AI is more tailored towards marketing, content creation, and things such as business applications. It's more adept at making things such as offering templates and tools for generating blog posts and making social media content. Jasper AI integrates with SEO tools making it ideal for businesses. Both ChatGPT and Jasper use Natural Language Processing (NLP), however ChatGPT is more flexible for a broader range of users, including technical or coding assistance.

5.4 How ChatGPT will be used in our project:

ChatGPT has many benefits and drawbacks, as we have discussed. We have looked at a few benefits of ChatGPT and examined its limitations. We have also researched different LLMs and compared them with ChatGPT. Now we will discuss how ChatGPT will be used in this project.

Since we are essentially building an animatronic robot, several key components are going to need to be designed correctly to make it all work in the end. We will utilize components such as servo motors, fuses, breakers, touch-screen HMI, a custom designed PCB and perhaps even an audio system so that the dinosaur is able to make noises. In designing this complex system, we plan to use ChatGPT extensively as an assistant to accelerate development and to improve the quality of the designs.

ChatGPT will be very useful to our project especially in the early stages because it can serve as a brainstorming partner during the early design stages of our project. I can describe high-level functions that I want the robot to do and ChatGPT can break that down into detailed steps, suggest layouts on how to make it, and recommend which components to use. For example, when determining the correct servo motor to use based on torque and rotation angle requirements, I can load preliminary data, and ChatGPT will be able to assist in estimating the appropriate size and model. By having this assistant, it saves time that would otherwise be spent looking through datasheets.

Custom PCB design is a core part of our project and as such we need to make sure our designs address important things such as capacitor placement, trace-width requirements for high current paths, or just overall debugging a PCB design, ChatGPT would be able to double-check our math and possibly prevent a faulty design. Additionally, ChatGPT can assist with making schematic layouts. Although some PCB circuits will be coming from TI WEBENCH, ChatGPT can provide example circuits for things such as motor control, power regulation, and microcontroller integration. If I describe a component I'm using, such as a DF3 audio player, ChatGPT could help generate a rough draft of a schematic or simply a proper pinout configuration.

ChatGPT is also very helpful for us to potentially troubleshoot issues with respect to the hardware and software portions of the project. The animatronic robot will require embedded programming for real-time control of the servo motors. For example, I could describe what I want the firmware to do, such as reading a potentiometer or adjusting a servo position, and ChatGPT can provide complete code examples explaining every line of code. ChatGPT can help design menus, interpret button press logic, and structure the software in a modular type of way. If we use a coding language like C++ in our design, then we can also use

ChatGPT to assist with object-oriented design principles in order to keep the code organized.

Power distribution is another critical component of any electromechanical system, and ours is no exception. We need to be able to design this animatronic so that the servo motors, controller, and various electronic components operate simultaneously. In our project, we are getting power from a receptacle which means 120VAC is being supplied to the robot. This voltage is then stepped down and converted to 24VDC via a transformer and rectifier system. To ensure the safe and reliable operation of this system, 10A fuses and 1A circuit breakers will be used throughout the system. ChatGPT has played an important role in the design, analysis, and troubleshooting of this power system.

In addition, ChatGPT will prove to be very useful in validating the overall architecture of this power distribution network. When designing the conversion from the 120VAC to 24VDC, I was able to consult ChatGPT for recommendations on transformer sizing, rectifier types, and voltage regulation techniques. ChatGPT helped me understand the differences between linear and switching power supplies, and how to be more efficient when powering servo motors and ladder logic circuits without introducing unnecessary electrical noise.

ChatGPT is also very helpful in the design and placement of the 1A MCB's and the 10A fuses. It has helped me in calculating current demands and appropriately size these protective components. By inputting estimates for how much current things such as the servo motors, the HMI, and the PCB, I was able to use ChatGPT to double check that 10A fuses would handle the combined load without tripping randomly, and yet still offering short-circuit protection. Additionally, ChatGPT assisted in the placement of the 1A circuit breakers, where more sensitive components are used.

Another important area that ChatGPT can support our project is in wire sizing and distribution layout. In sizing the wire we need, we can input current ratings and expected length of wire runs and ask ChatGPT to help determine the appropriate wire gauge to minimize voltage drop and prevent overheating. With the correct wire gauge chosen, we will need to be able to fit all of the wire and all of the electrical components(other than the PCB) inside of an industrial electrical enclosure. When putting all of the components together inside of our electrical enclosure, ChatGPT can help provide best practices for grounding and how to safely isolate high and low voltage sections and how to properly fuse branches that split off from the main 24VDC din rail.

ChatGPT can also assist with mechanical considerations involved in the design of the animatronic structure. When designing linkages and joint mechanisms that will translate the rotational motion of the servo motors into lifelike movement of the dinosaur's limbs, ChatGPT can offer insight into different mechanical configurations such as 4-bar linkages, gear ratios, or pulley systems. These

mechanical designs are crucial to ensuring that the movements of the animatronic are smooth, realistic, and mechanically sound.
ChatGPT also supports documentation efforts, a key aspect of any engineering project. When writing technical documentation, user manuals, or safety guidelines, ChatGPT can help structure content, provide templates, or even offer recommendations on what sections to include. This allows the team to spend less time formatting documents and more time on development. Additionally, ChatGPT can assist in preparing formal presentations by generating bullet points, slide outlines, or even sample diagrams that explain technical content in a way that's accessible to both engineers and non-engineers, such as faculty reviewers or potential stakeholders.

Another important area is regulatory and standards compliance. While we are not necessarily bound by strict commercial safety certifications, it’s still beneficial to follow best practices that align with common industrial and consumer safety standards. ChatGPT can summarize relevant portions of standards such as IPC-2221 (for PCB design), NEC (for electrical installations), or ISO guidelines related to robotics and machinery. By reviewing this information, we can design with greater confidence that our system is not only functional but also aligned with real-world safety and reliability expectations.

Moreover, during system integration, ChatGPT can help identify common pitfalls that occur when combining hardware and software components. It can offer advice on power sequencing, ensuring that certain parts of the system power on or off in a specific order to prevent damage or faults. It can also help us implement watchdog timers, fault recovery protocols, and modular startup routines in firmware, ensuring that our embedded system boots up and operates reliably. These tips are especially valuable during field testing, where subtle issues can arise that are difficult to trace without structured debugging strategies.

In summary, ChatGPT plays a multifaceted role in the development of our animatronic robot and provides critical support across all phases of the engineering process. From the early stages of concept development and system architecture to detailed tasks such as assisting in PCB design, power distribution planning, and firmware development, ChatGPT assists as a versatile engineering assistant. It can help to validate critical component choices, generate example circuits and code, troubleshoot hardware and software related issues, and even assist with documenting our progress. Whether it’s advising on trace routing in our custom PCB design or guiding the safe implementation of a 24VDC power system with fuses and circuit breakers that need to be sized and placed properly, ChatGPT’s ability to provide instant, context-aware responses, make it an indispensable tool to assist us in the timely completion of this project.

Chapter 6: Hardware Design

In this chapter, we will discuss the design theory and of the hardware implementation for the animatronic, motor control, and the PLC/HMI protection system enclosure.

6.1 W5500 Ethernet Circuit:

The schematic provided from Arduino outlines the integration of the W5500 Ethernet controller and Ethernet port into a customized circuit board based on the Arduino MKR ETH Shield design. The schematic is methodically arranged to ensure optimal operation of the Ethernet controller by maintaining signal integrity, precise voltage regulation, and effective communication protocols. It primarily includes two critical sections: the W5500 chip itself and the Ethernet port interface, with careful attention paid to power distribution by separating input voltages into distinct analog (3V3A) and digital (3V3D) domains. This segregation minimizes interference between analog and digital circuits, ensuring reliable and stable system performance.

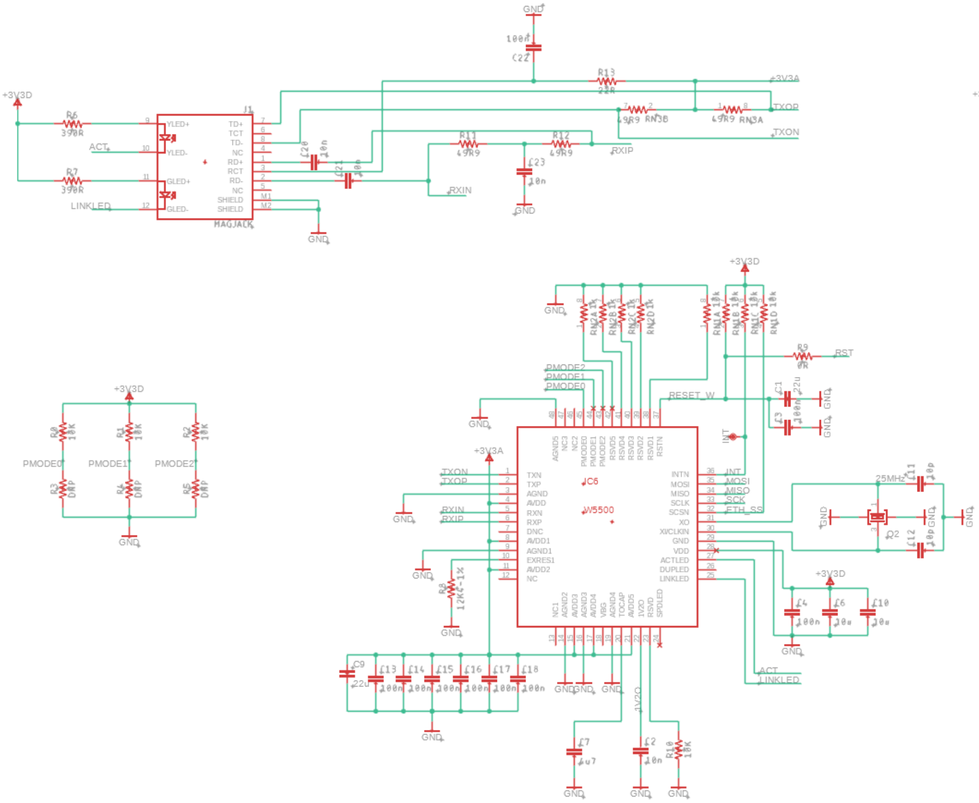


***Figure 6.1:** W5500 Ethernet Circuit Schematic*
By Authors

The W5500 is a hardwired TCP/IP Ethernet controller that integrates both MAC and PHY layers, simplifying system design by reducing external components.

Supporting protocols such as TCP, UDP, IPv4, ICMP, ARP, IGMP, and PoE, it effectively handles various embedded Ethernet applications. Communication between the W5500 and the host microcontroller occurs via an SPI (Serial Peripheral Interface), capable of speeds up to 80 MHz.

Essential SPI pins for the W5500 include MOSI (Master Output Slave Input), MISO (Master Input Slave Output), SCLK (SPI Clock), and SCSn (Chip Select). MOSI and MISO facilitate bidirectional data transfer, while SCLK synchronizes the timing of SPI communications. The SCSn pin activates the chip when driven low, enabling multiple devices to share the SPI bus efficiently.

Power integrity within the schematic is meticulously managed by isolating analog and digital voltages. The analog domain (3V3A) supplies critical PHY circuits, while the digital domain (3V3D) powers the controller's digital logic. Multiple decoupling capacitors (each 100nF) are positioned near power pins to stabilize voltages, mitigate electrical noise, and provide rapid current response during transient conditions.

A precise external resistor (12.4 kΩ, 1% tolerance) connected to the EXRES1 pin establishes the necessary bias for the chip's internal analog reference circuits. Correct resistor values ensure accurate PHY operations and consistent network performance. Additionally, capacitors connected to TOCAP and 1V2O pins stabilize internal voltage references, reducing ripple and supporting reliable PHY functionality.

A 25MHz crystal oscillator is included between the XI and XO pins, aligning with standard Ethernet frequency requirements for 10/100 Mbps operation. Two accompanying capacitors (10 pF each) form a feedback loop with the crystal, ensuring stable oscillations and accurate timing.

The schematic incorporates status indicators through dedicated LEDs (ACTLED, LINKLED, DUPLED, and SPDLED), providing visual feedback on network activity, link status, duplex mode, and connection speed. These LEDs significantly assist with monitoring and diagnostics during system operation.

Ethernet PHY signals (TXP, TXN, RXP, RXN) connect directly to an RJ45 connector through integrated magnetics, which include isolation transformers for galvanic isolation. These transformers protect the circuit from electrical interference and ensure correct impedance matching (typically 100 ohms) for standard Ethernet cables.

Series termination resistors (49.9Ω each) on differential Ethernet pairs minimize signal reflections and maintain proper impedance, enhancing communication reliability. Additional filtering elements (390Ω resistors and 10nF capacitors) further improve signal quality and guard against transient voltage spikes.

The RJ45 connector wiring adheres to Ethernet standards, using specific pins for transmit (TD+, TD-) and receiving (RD+, RD-) signals. Additional connector pins facilitate functions required by Ethernet standards, including spare pairs and termination references.

PHY operation modes are configured via the PMODE[0:2] pins, using pull-up or pull-down resistors to set auto-negotiation, communication speed, and duplex mode during system startup. The RSTn pin ensures proper reset functionality, driven high through a pull-up resistor to guarantee correct initialization. The INTn pin provides interrupt notifications to the microcontroller for critical network events, ensuring timely software responses.

Overall, this carefully organized schematic ensures robust and efficient Ethernet communication through meticulous pin-level consideration, comprehensive signal integrity management, precise timing, and efficient power distribution. The removal of unnecessary elements, such as the SD card from the original Arduino design, further streamlines the schematic, enhancing reliability and reducing complexity for dedicated Ethernet applications.

6.2 Voltage Regulator Circuit:

Our PCB design uses the LM2576HV step-down voltage regulator to efficiently convert a 12V input signal to a regulated 5V output signal. The input signal is stabilized by 2 capacitors that have a value of 220uF which are there to help filter out high-frequency noise and smooth and ripple from the power supply. These input capacitors are critical in ensuring that the regulator receives a steady input voltage, which in turn improves performance and reliability.

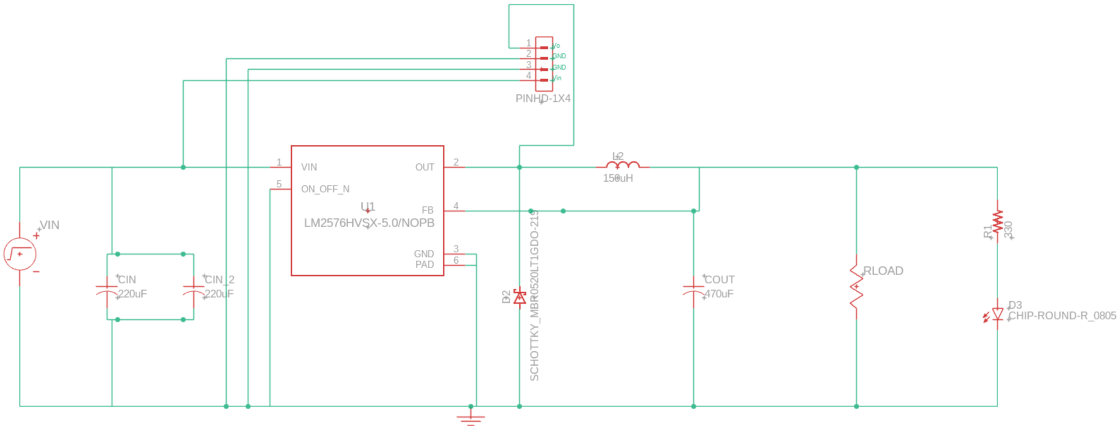


Figure 6.2: *LM2576HV Voltage Regulator Circuit Schematic*
By Authors

Another critical component in this design is the 150uH inductor. It is placed between the output pin and the load to store and transfer energy efficiently during the switching operation. This inductor helps to maintain a stable current output and it reduces the ripple voltage. In addition, the Schottky diode (D1) is placed across the inductor and output which is essential for when the regulator is turned off. The diode needs to have a low forward voltage to maintain high efficiency and avoid heating.

6.3 MCU:

The MCU that will be used in the PCB is the ATSAMD21G18A Cortex-M0+, a 32-bit ARM-based MCU (SAMD21). It is the same MCU used in the OpenRB-150. The SAMD21 supports a range of comms protocols to enable seamless interaction between sensors, motors, and other peripherals.

- Serial Communication: includes UART, SPI, and I2C. These will be used to connect to the peripheral devices.
- Digital and Analog I/O: the general-purpose I/O (GPIO) pins will be leveraged for the sensor data. This project will mainly focus on the use of the digital pins, keeping analog for potential future use beyond this project.
- USB interface: A key feature of the MCU - native USB interfacing allows for communication between a device of choice and the PCB without requiring the use of any adapters or external serial controller.

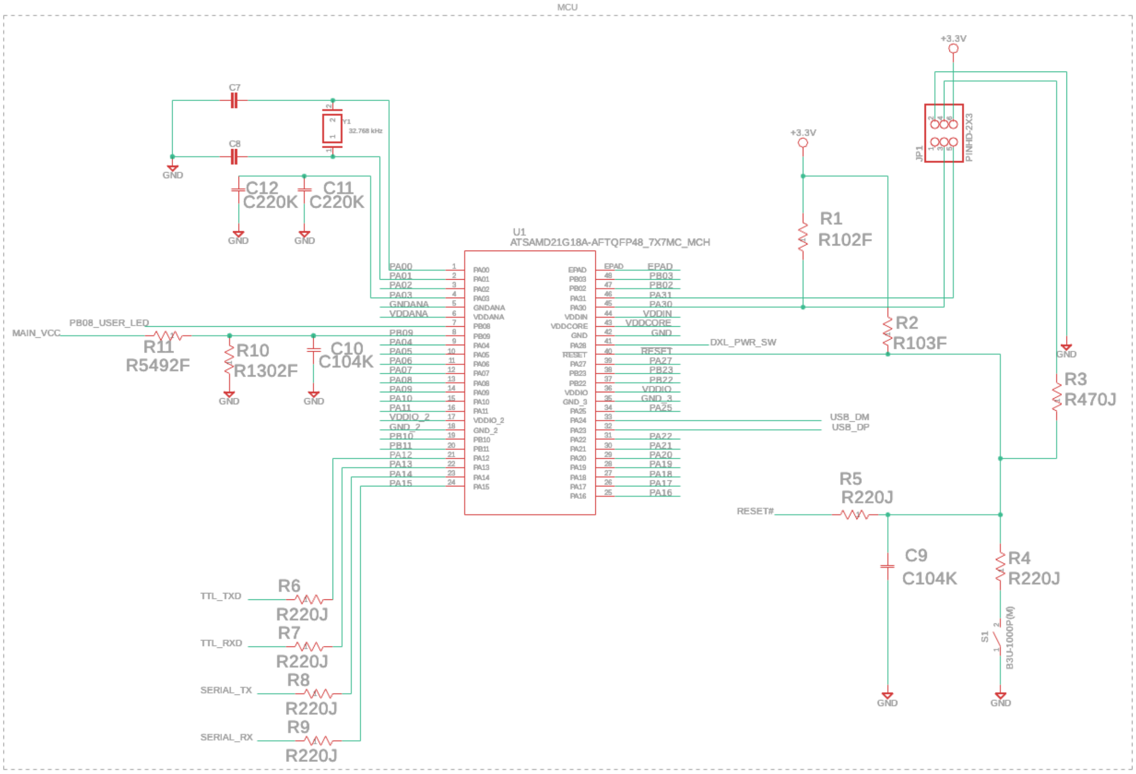


***Figure 6.3**: MCU Circuit Schematic*
By Authors

6.3.1: Clock

To maintain timing accuracy, the SAMD21 features a 32.768 kHz crystal oscillator at pins PA00 and PA01 serving as the timing reference for the internal timers and sleep modes built into the controller. Each connection for the crystal is paired with load capacitors connected to ground to form a stable oscillating circuit. The lack of capacitors can result in the timing becoming off over time. The capacitor's ability to hold a charge and discharge it will stabilize the signal.

6.3.2: Programming and Debugging

Programming and debugging capabilities are available via a 3x2 0.1" header, allowing for in-system programming.

- SWCLK (PA30): Serial Wire Clock
- SWDIO (PA31): Serial Wire Data
- RESETN: Manual reset input

Pull-up resistors ensure clear logic levels at SWD as well as managing reset signal behavior. The reset line is connected to a physical, tactile switch. When the switch is pressed down, which pulls the RESETN line to low and a capacitor gives a debounce effect delay and avoid a false trigger. The resistor prevents a floating input state and maintains valid logic levels. Although the OpenRB-150 calls for the header to be "DNP", or "Do Not Place", we have opted to include it in the project for the added flexibility.

6.3.3: USB Interface

This is where the SAMD21's native USB functionality is involved. The USB_DM (PA24) and USB_DP, which are Data Minus and Data Plus, are routed to the USB-C Programming Port circuit.

The USB interface allows for the firmware deployment to the MCU and the usage of the Arduino IDE to program the profiles for the animatronic. It also enables the use of the DYNAMIXEL Wizard 2.0 software for configurating the motors and real-time data view and diagnostics. Electrostatic discharge diodes are added to the USB lines to protect the MCU against any voltage spikes or static discharge.

6.3.4: UART and TTL

Two UART channels are configured for TTL-level serial communication. The lines used for the servo motors are:

- TTL_TXD / TTL_RXD (PA12 and PA13)
- SERIAL_TX / SERIAL_RX (PA14 / PA15)

Both UART lines are routed from the MCU to the TTL transceivers with auto TX enable capability. Each TX and RX line has 220 Ohm resistors that serve two purposes. First, it limits the inrush current that can be generated during transmission. Second, it protects against a potential voltage mismatch.

6.3.5: Input/Output Pins and Peripheral Signals

The SAMD21 features a flexible I/O structure that allows multiple roles to be assigned to various pins. For example:

- PB08: drives a programmable status LED through a current-limiting resistor as well as a pull-down resistor.
- PA02 - PA07: additional analog inputs
- PA08 / PA09: I2C interface
- PA16 - PA19: SPI interface including MOSI, MISO, SCK, and SS
- PA10 - PA-23, PB10 - PB11, PB22 - PB23: Timer and Counter outputs, and additional UART

6.4 PLC Wiring:

In industrial automation, proper I/O (Input/Output) wiring configuration is essential for ensuring reliable communication between sensors, actuators, and the Programmable Logic Controller (PLC). The diagram provided (Figure 6.4) illustrates the wiring for a Micro820 PLC (2080-LC20-20AWB) using a 2080-PSAC12W power supply. The system is configured with sinking inputs and sourcing outputs, a widely accepted industrial standard due to enhanced noise immunity and compatibility with a wide range of field devices.

Sinking vs. Sourcing I/O:

- **Sinking (NPN):** The input or output device provides a path to ground (0V). In this configuration, the PLC input terminal is internally connected to the negative side of the power supply, and current flows into the input terminal when a sensor switches the positive voltage.
- **Sourcing (PNP):** The input or output device supplies positive voltage (+VDC) to the load. The PLC output terminal is internally connected to the positive side of the power supply, and current flows out of the output terminal toward the load.

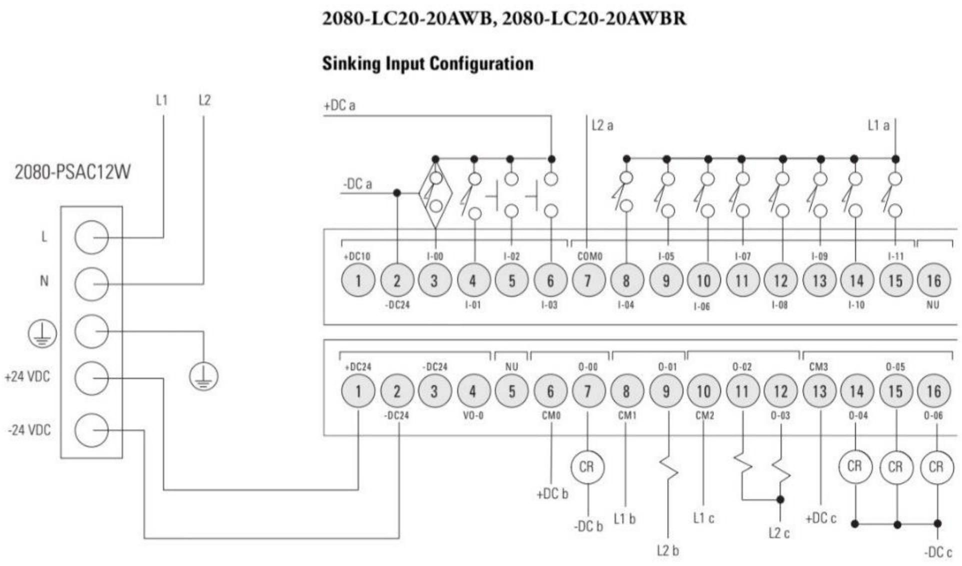


Figure 6.4: *Proper Wiring Diagram for Micro820-AWB PLC*
Image Provided by Rockwell Automation

System Configuration:

- **Inputs (Sinking Configuration)**: In this setup, external field devices such as limit switches, push buttons, or proximity sensors are connected between the positive DC rail and the PLC input terminals. When a device closes the circuit, current flows from +24VDC through the device into the PLC input terminal, which is internally referenced to 0VDC. This sinking configuration is shown in the top portion of Figure 6.4.
- **Outputs (Sourcing Configuration)**: The lower portion of Figure 6.4 demonstrates sourcing output wiring. The PLC outputs supply +24VDC to field devices (e.g., relays, indicator lights, solenoids) when energized. Current flows from the PLC output terminal through the device to the negative rail. This method is preferred in industrial environments due to its compatibility with grounded enclosures and minimized risk of short-circuits on signal lines.

This clear division and direction of current flow ensures robust operation and simplifies debugging and system maintenance. Our choice of sinking inputs and sourcing outputs aligns with established automation practices and maximizes system compatibility with industrial sensors and actuators.

6.5 USB-C Programming Port:

The USB-C port acts as both a power input and data communication channel. It is directly connected to the USB_DM and USB_DP lines of the SAMD21. The port supports plug-and-play functionality for development. The USB-C port is what allows us to upload the Arduino sketches, communicate with the SAMD21 via serial terminal, and interface with the motors through DYNAMIXEL Wizard 2.0

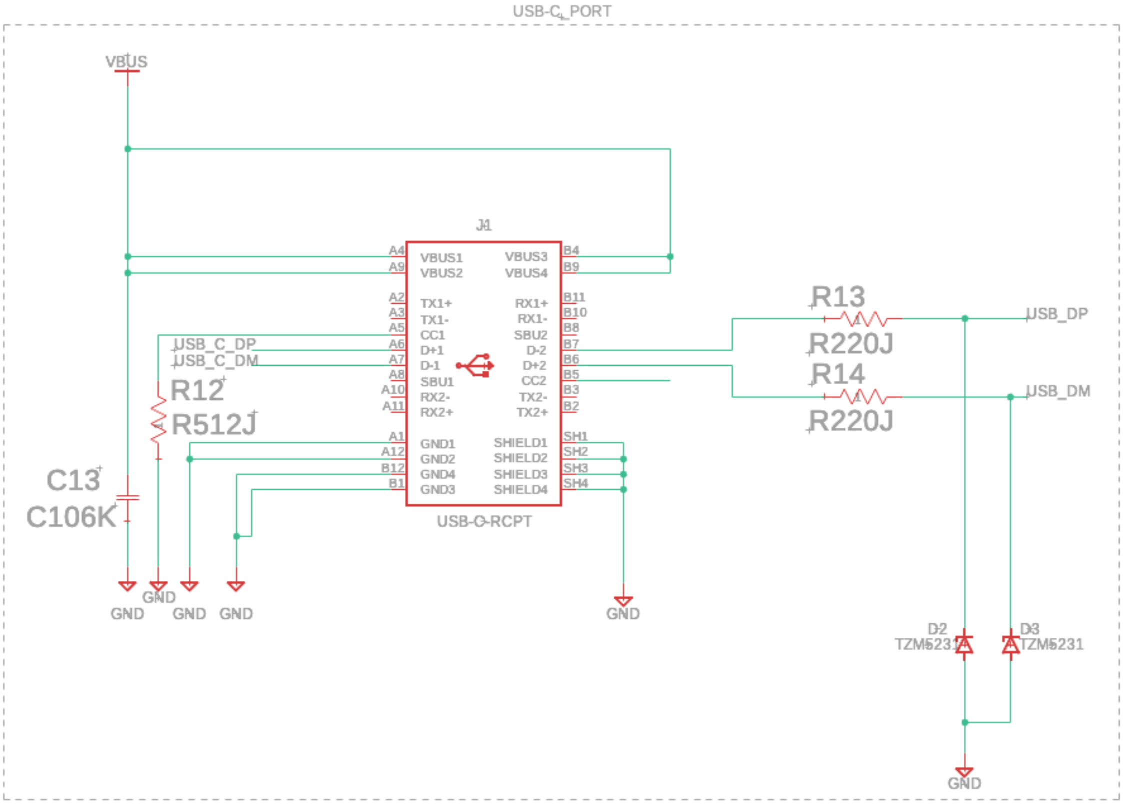


***Figure 6.5**: USB Program Port Circuit Schematic*
By Authors

6.6 Motor Port:

This PCB includes two dedicated motor ports - one for servos using TTL communication, and the other for the stepper motor driver.

- DXL_TTL: half-duplex and bi-directional serial comms line that combines transmit and receive into a single wire.
- DXL_PWR: dedicated power line for motors, able to be daisy chained motor to motor.
- GND: ground signal

The stepper motor is controlled using a separate set of GPIO signals. It does not rely on TTL but uses direct digital control from the GPIO pin to the driver.

6.7 Motor Control and TTL Circuit:

The servo motor port connects to TTL ICs, utilizing the logic levels (0V logic low; 5V logic high). It features Auto TX Enable, meaning that the SAMD21 is capable of automatically switching the transmission line on and off. The motor port consists of:

- TTL_TXD: Transmit Data - send instructions over TTL from MCU to motors.
- TTL_RXD: Receive Data - receive instructions from motors to MCU including position, torque, and status.

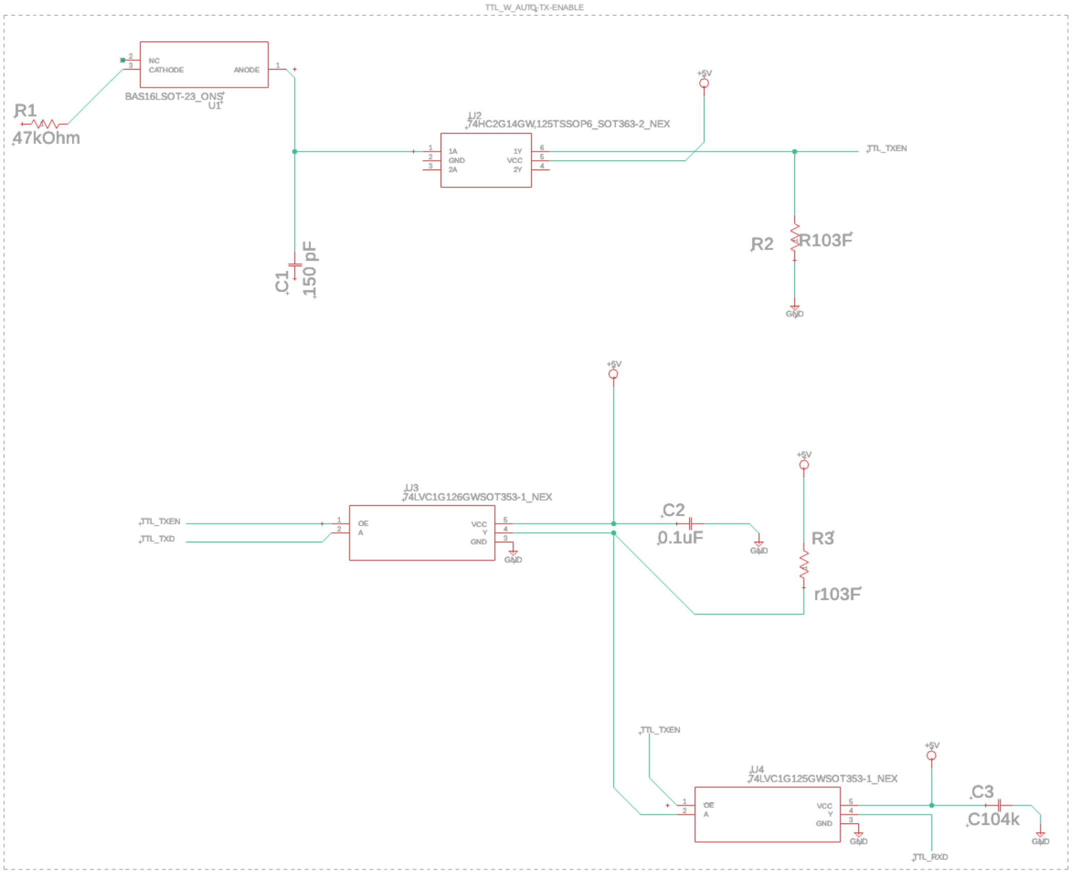


***Figure 6.6**: TTL Circuit Schematic*
By Authors

6.8 Status LEDs:

The purpose of the status LEDs is to provide real-time feedback of what the PCB's status. Resistors are present to limit the current draw and prevent damage to the LEDs. There are three LEDs in the circuit.

- Heartbeat: green LED, indicates that the MCU is turned on and in a healthy state.
- Communication Activity: orange LED, meant to indicate if active communication is occurring.
- Fault: red LED, indicates if a motor has a fault status on it.

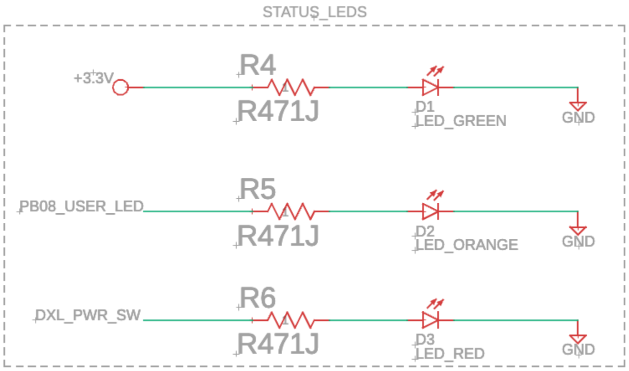


***Figure 6.7**: Status LEDs Circuit Schematic*
By Authors

6.9 Power Requirements:

By specification, the OpenRB-150 supports an input voltage requirement ranging from 3.7V to 12.6V, as well as 5V from the USB-C port. For our custom PCB, we will have a voltage input of either 5V from the USB-C port or a standard 12V input terminal. Two regulators are required for the circuit power - a 3.3V regulator and a 5V regulator. The regulators are daughter boards that are connected to headers on the PCB. To power the MCU, 3.3V is applied to the VDDIO and VDDIN pins, which power the digital I/O pins and the internal voltage regulators respectively. VDDCORE is a core power supply and VDDANA is analog supply.

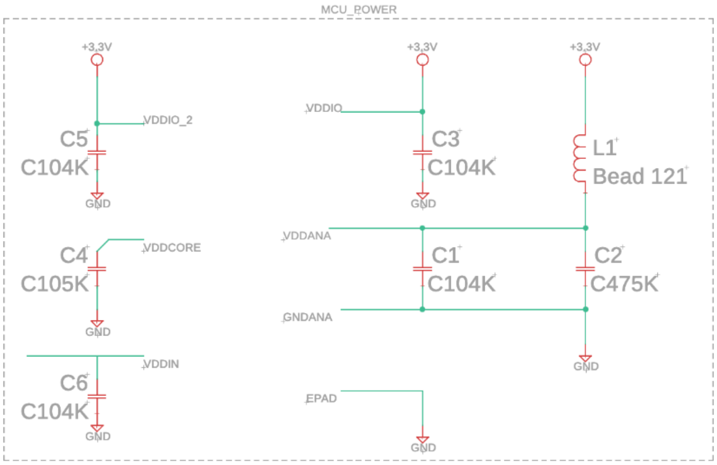


***Figure 6.8**: MCU Power Circuit Schematic*
By Authors

6.10 Existing Part Replacement:

Due to predetermined project requirements, the printed circuit board must contain parts smaller than the 0805-resistor case code. Therefore, many components needed to be substituted to meet this requirement. 0805 is an imperial unit measuring at approximately 0.08in x 0.05in. Its metric equivalent is 2012, which measures 2.0mm x 1.25mm. Most existing parts provided by DYNAMIXEL measure at 1005. Since the project requirements call for the sizes to increase, substitutions are easier to make. An increased amount of surface area per each component will allow for more than enough overhead in power requirements as power dissipation is improved. A table with sizes is provided to better understand size requirements. Any size greater than or equal to 2012 metric or 0805 imperial will be used in this project.

***Table 6.1**: Imperial Resistor Case-Code*
By Authors

Resistor Case Code	Approx. Length (in)	Approx. Width (in)	Power (W)
01005	0.016	0.008	0.031
0201	0.02	0.01	1/20 (0.05)
0402	0.04	0.02	1/16 (0.062)
0603	0.06	0.03	1/10 (0.1)
0805	0.08	0.05	1/8 (0.125)
1206	0.125	0.06	1/4 (0.25)
1210	0.125	0.1	1/2 (0.5)
1812	0.18	0.125	3/4 (0.75)
2010	0.2	0.1	3/4 (0.75)
2512	0.25	0.125	1

***Table 6.2**: Metric Resistor Case Code*
By Authors

Resistor Case Code	Approx. Length (mm)	Approx. Width (mm)	Power (W)
0402	0.4	0.2	0.031
0603	0.6	0.3	1/20 (0.05)
1005	1	0.5	1/16 (0.062)
1608	1.6	0.8	1/10 (0.1)
2012	2	1.25	1/8 (0.125)
3216	3.2	1.6	1/4 (0.25)
3225	3.2	2.5	1/2 (0.5)
4532	4.5	3.2	3/4 (0.75)
5025	5	2.5	3/4 (0.75)
6332	6.3	3.2	1

The OpenRB-150 schematic calls for generic code for each component. To decipher each is straightforward. Resistors start with R followed by a 3-digit code and a letter. The first two digits represent significant numbers, and the last digit is the number of zeros to append. The letter at the end indicates the tolerance.

Table 6.3*: Resistance Tolerance Values*
By Authors

Letter Code	Tolerance
F	+/- 1%
G	+/- 2%
J	+/- 5%
K	+/- 10%
M	+/- 20%

Likewise, the OpenRB-150 schematic also provides identifiers for the capacitors.

Table 6.4*: Capacitance Tolerance Values*
By Authors

Letter Code	Tolerance
B	+/- 0.10 pF
C	+/-0.25 pF
D	+/- 0.5 pF
F	+/- 1%
G	+/- 2%
J	+/- 5%
K	+/- 10%
M	+/- 20%
Z	+80% / - 20%

Parts substitutions were made as follows:

Table 6.5*: Parts substitutions made to OpenRB-150's design*
By Authors

Substitutions			
Generic Case Code	Old Size	New Part	New Size
R473J	1005	RMCF0805JT47K0	0805
BAS16L	1006	BAS16LT1G	0805
C104K	1005	KGM21NR71H104KT	0805
R103F	1005	RC0603FR-0710KL	0805
C151K	1005	C0402C151K5RACTU	0805
R470J	1608	RPC2512JT47R0	0805

Chapter 7: Software Design

In this chapter, we will discuss the design theory and control flow of the software implementation for both the animatronic and the PLC protection system.

7.1 C Programming for Robotics:

The C programming language is a foundational tool for embedded systems, particularly in robotics. Understanding C's essential characteristics, including efficiency, real-time control, hardware interfacing, portability, and modularity, is critical to addressing the specific demands of robotic applications.
Embedded systems function as specialized computing units performing dedicated tasks within larger mechanical or electrical assemblies. In robotic systems specifically, embedded components handle crucial functions such as real-time data processing, sensor management, actuator control, and inter-device communications. Consequently, selecting an appropriate programming language is vital, requiring efficiency, low-level hardware control, and reliability. The C language is particularly suited due to its direct access to hardware components, enabling precise management and optimization of system resources.

A significant factor contributing to C's suitability for embedded robotic systems is its inherent efficiency. The language allows meticulous management of memory allocation and processor utilization, crucial in environments constrained by limited memory and computational resources typical of microcontroller-based robotic systems. By generating compact, highly optimized code, C ensures the smooth execution of applications, minimizing resource overhead and thus enhancing robotic system performance and responsiveness.

Real-time performance is another critical advantage of C, particularly within robotic applications that demand immediate response to sensory input and strict adherence to timing constraints. Unlike higher-level languages, C provides deterministic behavior, free from unpredictable delays such as garbage collection pauses. This ensures that control algorithms execute consistently within their required time frames, significantly contributing to system stability and reliability, especially in scenarios demanding precise timing.

C's capability for direct hardware interfacing is exceptionally beneficial in robotics applications. Developers leveraging C can directly interact with microcontroller registers and peripheral devices, streamlining the development of specific hardware drivers for sensors, actuators, motors, and communication interfaces. This granular level of control facilitates tailored, high-performance solutions precisely aligned with application-specific requirements, essential for optimizing behavior and achieving peak performance in robotic systems.

The portability offered by C represents another key advantage for embedded robotic systems. Due to the language's widespread adoption across various microcontroller architectures, numerous compilers and development tools are readily available, allowing for smoother adaptation of existing code across different hardware platforms. This broad availability simplifies scaling processes, reduces development time when transitioning between systems, and enhances maintainability. Furthermore, extensive community support and numerous libraries bolster this advantage, providing robust resources for developing complex robotic systems.

Beyond hardware efficiency and control, C's structured syntax promotes clarity in system operations, significantly aiding debugging, maintenance, and future system upgrades. The modular nature of C programming supports systematic testing and validation of discrete system components, essential in complex robotic architectures composed of multiple interacting subsystems. This modularity enhances reliability, eases troubleshooting, and facilitates ongoing system improvements.

Additionally, utilizing C aligns effectively with educational and research efforts in robotics. Learning and employing C provides foundational knowledge in embedded system principles, real-time operations, and hardware interactions. Many educational frameworks and open-source initiatives use C to introduce these concepts, offering students and researchers practical experience in realistic applications. Such hands-on learning is invaluable, fostering the necessary skills to contribute effectively to innovation within robotics.

In summary, the intrinsic properties of the C programming language (its efficiency, predictable real-time performance, low-level hardware access, broad portability, and modularity) establish it as a prime candidate for robotic embedded system development. These characteristics support the creation of responsive, reliable, and scalable robotic systems that effectively address real-time operational demands and resource limitations. As the robotics field continues to progress, C's foundational role in embedded system design remains critically relevant, underscoring its sustained significance and practicality.

7.2 Motor Tuning:

Proportional gain tuning will correct the position depending on the setting. Adjusting the gain to be low will cause a slow response by the motor, barely moving into its goal position. When the gain is set to a high number, it will prolong the steady state error as the motor keeps overcorrecting itself. Integral tuning will consider the accumulated error and keep adjusting itself until it reaches the final position. Low integral gain results in not reaching the target position, and high integral gain will cause overshoot and oscillate until reaching steady state. Finally, derivative tuning affects the system’s overshoot and oscillation.

Table 7.1: *PID Tuning Ratios*
By Authors

	Low	High
Proportional Gain	Slow response by motor	Motor will overshoot the goal position and over-correct
Integral Gain	Motor will not reach goal position	Motor will overshoot and oscillate until steady state
Derivative Gain	Motor will overshoot its goal position	Motor will move attempt to move to goal position slowly

The DYNAMIXEL position controller operates similarly to its velocity counterpart, but its main objective is to precisely control the actuator's position rather than speed. The process starts with the user sending commands to the actuator via RS-485 or TTL communication, specifically the "Goal Position" parameter, which defines the desired target angle or position. This input is passed through a trajectory generator (or profile), which shapes the motion using user-defined parameters for velocity and acceleration. These profiles help generate smooth motion paths that reduce mechanical shock and enhance control accuracy.

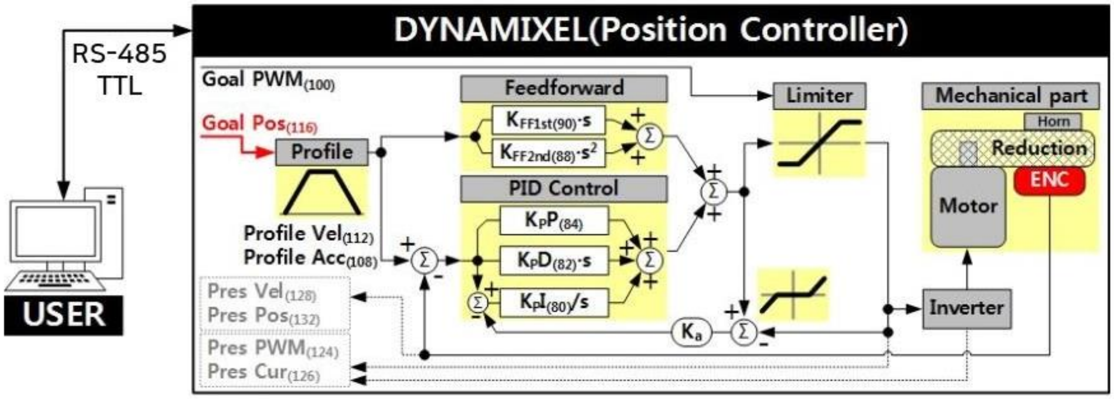


Figure 7.1: *DYNAMIXEL Position PID Logic*
Image Provided by ROBOTIS

The output from the profile generator feeds into a PID control system combined with a feedforward mechanism. The feedforward path uses two gains: K_{FF1st} (velocity-based) and K_{FF2nd} (acceleration-based), which help anticipate the motion's needs and reduce the burden on the feedback loop. These feedforward terms add predictive elements to the control output, making the motor respond more effectively, especially during fast or complex movements.

The PID control block refines the motion using three classic control terms. The proportional gain (K_P) addresses current position error, the derivative gain (K_D) reacts to the rate of position change to counter overshooting, and the integral term (implicitly handled or disabled in some DYNAMIXEL models) would typically correct long-term steady-state errors. The combined PID and feedforward signals are then processed by a limiter to constrain the command within safe bounds, preventing hardware damage or excessive torque output.

After limiting it, the signal is directed to the inverter, which drives the actuator’s motor. The motor is mechanically linked to a reduction gearbox and an encoder (ENC). This encoder provides real-time position feedback, which is vital for the control system to continuously adjust and refine the motor's behavior to match the goal position accurately.

Feedback values such as present position, velocity, current, and PWM output are continuously monitored and can be read by the user. These feedback signals are crucial for diagnostics, system health monitoring, and fine-tuning the controller for different loads or performance requirements. Proper tuning of control parameters like K_P, K_D, and the feedforward gains is essential. High proportional gain improves responsiveness but can cause oscillations, while derivative gain enhances stability by damping motion. The feedforward terms should be used to reduce control lag during acceleration and deceleration without destabilizing the system.

7.3 Motor Control Programming, DYNAMIXEL Wizard 2.0 Configuration and Diagnostic Software:

To configure, monitor, and validate the performance of the DYNAMIXEL servo motors used in the animatronic, the project utilized DYNAMIXEL Wizard 2.0, an official software suite developed by ROBOTIS. This cross-platform tool allows for comprehensive control over motor behavior, real-time data plotting, firmware management, and diagnostic testing. It is an essential tool during both the development and the debugging phases.

7.3.1 Core Features and Capabilities

DYNAMIXEL Wizard 2.0 enables seamless communication with motors via USB-based serial converters. The software provides an intuitive GUI with access to a range of functionalities, including:

- Motor Discovery and Scanning: Users can scan for connected DYNAMIXEL motors over selectable communication ports and baud rates. Each detected motor is categorized by model and ID and displayed in a side panel for further access (see Figure 7.1).
- Control Table Access: Each DYNAMIXEL has a unique control table that defines its configuration and runtime behavior. Parameters like Goal Position, Operating Mode, and Torque Enable can be modified directly in the GUI (see Figure 7.2). This control enables fast iteration over motion settings.

- Live Graphing: Real-time plots of control values such as Present Position and Goal Position provide insight into system performance and enable tuning of PID gains and motion profiles. The software supports adjustable logging intervals down to 1 millisecond (see Figure 7.3).
- Packet Monitoring: A packet window allows users to construct and transmit protocol-specific instruction packets, enabling low-level testing of DYNAMIXEL commands and return data parsing (see Figure 7.4).
- Firmware Utilities: Firmware updates and recovery operations are built in. The software checks for outdated versions and allows safe flashing, or manual recovery in case of corruption.
- Backup and Restore Functions: Configuration data from the EEPROM and selective RAM fields can be saved and restored using .ctd files. This feature streamlined repetitive setup tasks and preserved motor states across sessions.
- Self-Diagnostics: Diagnostic routines analyze motor health indicators such as voltage, temperature, and communication integrity, while allowing positional calibration and performance testing (see Figure 7.5).

7.3.2 Integration with Project Workflow

Throughout the project, DYNAMIXEL Wizard 2.0 was used extensively for the following tasks:

- Assigning unique motor IDs to avoid conflicts on the TTL communication bus.
- Monitoring live position feedback to compare theoretical and actual movement.
- Adjusting PID gain value to minimize jitter and ensure smooth motion.
- Saving and restoring working configurations for easy reinitialization.

7.3.2 Control Table

The control table provided by ROBOTIS maps various motor parameters to specific memory addresses. These allow us to read and write data for monitoring and controlling the motors in real time. Each entry in the control table is associated with an address, data size, access type, initial value, defined operating range, and the unit. The control table is divided into configuration parameters, control parameters, and status monitoring. These settings are critical for tuning motor behavior, setting software safety thresholds, and integrating the motor with the programmable logic controller.

By writing specific addresses within the control table, developers can adjust key characteristics such as PID gains, acceleration profiles, and torque limits. Reading from the addresses provides real-time feedback necessary for motion planning, diagnostics, and stability. The control table was utilized to configure the motors to match the desired movement profile of the axis.

Table 7.2: *Example Control Table from DYNAMIXEL XC-330-T288-T Datasheet Provided by ROBOTIS*

Address	Size(Byte)	Data Name	Access	Initial Value	Range	Unit
0	2	Model Number	R	1,060	-	-
2	4	Model Information	R	-	-	-
6	1	Firmware Version	R	-	-	-
7	1	ID	RW	1	0 ~ 252	-
8	1	Baud Rate	RW	1	0 ~ 7	-
9	1	Return Delay Time	RW	250	0 ~ 254	[μsec]
10	1	Drive Mode	RW	0	0 ~ 13	-
11	1	Operating Mode	RW	3	0 ~ 16	-
12	1	Secondary (Shadow) ID	RW	255	0 ~ 252	-
13	1	Protocol Type	RW	2	1 ~ 2	-
20	4	Homing Offset	RW	0	-1,044,479 ~ 1,044,479	[pulse]
24	2	Moving Threshold	RW	10	0 ~ 2,047	0.229 [rev/min]
31	1	Temperature Limit	RW	72	0 ~ 100	[°C]
32	2	Max Voltage Limit	RW	140	60 ~ 140	0.1 [V]
34	2	Min Voltage Limit	RW	60	60 ~ 140	0.1 [V]
36	2	PWM Limit	RW	885	0 ~ 885	0.113 [%]
44	4	Velocity Limit	RW	265	0 ~ 1,023	0.229 [rev/min]
48	4	Max Position Limit	RW	4,095	0 ~ 4,095	[pulse]
52	4	Min Position Limit	RW	0	0 ~ 4,095	[pulse]
60	1	Startup Configuration	RW	3	-	-
63	1	Shutdown	RW	52	-	-

Chapter 8: System Prototype Construction

In this chapter, we will discuss the system prototype construction and fabrication, including the PCB layouts.

8.1 Animatronic:

The 3d printed animatronic T-Rex will internally house the motors and their wiring. This will be done using CAD techniques to embed the motors into the STL files of the T-Rex directly, with added extrusions for routing of wiring internally. Since this is a skeleton T-Rex model, hiding all motors will not be feasible. However, the point of this project is to create a showcase for this technology, and with a fully visible yet still sleek design for motor placements, it will perfectly capture the use case of this product for future use. For example, an existing use case for a T-Rex animatronic with embedded actuators is the animatronic T-Rex in the Jurassic Park ride at Universal Studios Florida, Islands of Adventure. This animatronic employs a latex-polymer skin to cover the inside mechanism from the view of guests, creating a more realistic guest experience. Below is an example of one of our Dynamixel motors embedded in the neck, where it will meet the rear of the skull. This movement will allow for head rotation.

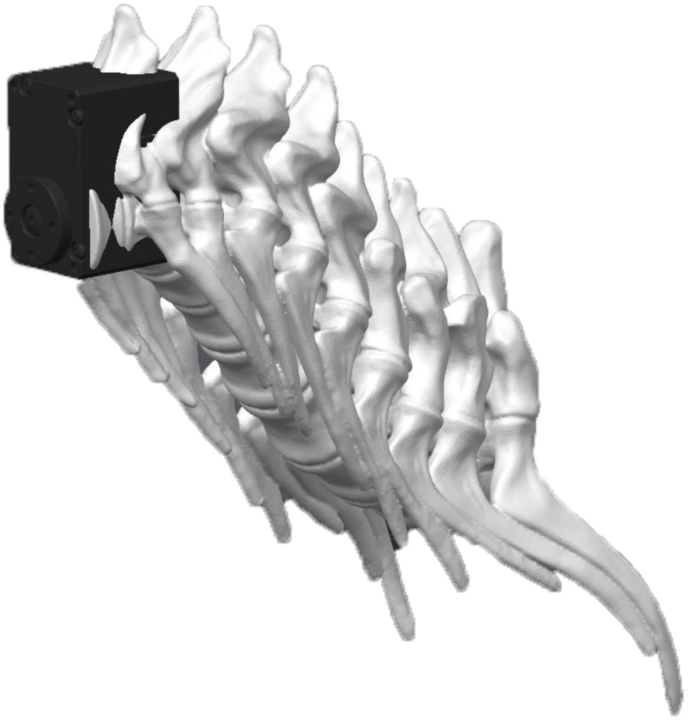


Figure 8.1: *Embedded Dynamixel XC-330-T288-T Motor*
By Authors

***Table 8.1:** Animatronic Actuators Axis of Rotation*
By Authors

Dynamixel XC-330-T288-T	Jaw (Open/Close)
Dynamixel XC-330-T288-T	Head Rotate (360 Degree Rotation)
Dynamixel 2XL430-W250-T	Neck Full Articulation (Left/Right, Up/Down)
Dynamixel XL430-W250-T	Body Articulation (Left/Right)
Dynamixel XL430-W250-T	Tail Articulation (Left/Right)
NEMA 17 Stepper Motor	Motion Base (Forward, Backward)

Jaw Actuation (Dynamixel XC-330-T288-T):
The jaw mechanism utilizes the compact Dynamixel XC-330-T288-T servo motor, measuring 20×34×26 mm and weighing just 23 grams. This actuator delivers a stall torque of up to 0.92 Nm and operates at a no-load speed of 65 RPM. Its high-resolution encoder provides 4,096 steps per revolution, allowing precise control over the jaw's open and close movements. The motor's small footprint facilitates seamless integration into the T-Rex's skull structure, enabling realistic biting motions without compromising the skeletal aesthetic.

Head Rotation (Dynamixel XC-330-T288-T):
For head rotation, another XC-330-T288-T servo is employed, capitalizing on its 360-degree rotation capability. This actuator's compact size and high torque output make it ideal for positioning within the neck assembly, allowing the T-Rex's head to rotate smoothly and continuously. The motor's precise control ensures lifelike scanning motions, enhancing the animatronics' interactive presence.

Neck Articulation (Dynamixel 2XL430-W250-T):
The neck's complex movements are managed by the innovative Dynamixel 2XL430-W250-T, which integrates two actuators within a single module. Each axis offers a stall torque of 0.28 Nm and operates at a no-load speed of 57 RPM. This configuration allows independent control of pitch and yaw, enabling the T-Rex to perform naturalistic nodding and turning motions. The module's compact design simplifies installation and reduces wiring complexity, making it an efficient solution for multi-axis articulation.

Body Articulation (Dynamixel XL430-W250-T):
The body sway is facilitated by the Dynamixel XL430-W250-T servo, which provides a stall torque of up to 1.5 Nm and a no-load speed of 61 RPM. Its dimensions of 28.5×46.5×34 mm and weight of 57.2 grams make it suitable for integration into the torso section. The actuator's high-resolution encoder ensures smooth and controlled lateral movements, simulating the subtle shifts of a breathing or alert creature.

Tail Articulation (Dynamixel XL430-W250-T):
Mirroring the body articulation, the tail's side-to-side movements are powered by another XL430-W250-T servo. This actuator's robust torque and precise control enable dynamic tail motions that contribute to the animatronics' balance and expressiveness. Its compact size allows for discreet placement within the tail structure, maintaining the skeletal design while providing fluid motion.

Motion Base (NEMA 17 Stepper Motor):
The animatronics' mobility is achieved through a NEMA 17 stepper motor mounted on the base of an Ender 3 bed slinger platform. This motor typically offers a holding torque of around 44 N·cm and operates with a step angle of 1.8 degrees, providing precise linear movement. By repurposing the 3D printer's motion system, the T-Rex can glide forward and backward smoothly, enhancing its interactive capabilities during demonstrations.

8.2 Wires, Organization and Raceways:

Proper wire management is critical in automation and animatronic systems to ensure safety, maintainability, and professional-quality assembly. In this project, we are implementing industry best practices for organizing wires within control cabinets and enclosures, including the use of raceways, standardized wire color coding, and careful routing techniques.

To ensure safe and intuitive identification of circuits, we adhere to standardized wire color coding:

- **120V AC Wiring:**
  - Black - Line (Hot)
  - White - Neutral
  - Green or Green/Yellow - Earth Ground
- **24V DC Control Wiring:**
  - Red - Positive (+24V)
  - Blue - Common (To Differentiate from 120 AC)

By following these conventions, the cabinet remains compliant with industrial norms, reduces the risk of electrical accidents, and simplifies maintenance or expansion.

Chapter 9: System Testing and Evaluation

In this chapter, we will discuss the system testing, adjusting and evaluation of the prototyping and fabrication. To streamline development and testing, a dedicated test bench was assembled, as shown in Figure 9.1. This setup provided a centralized and organized platform for integrating and evaluating all system components under controlled conditions.

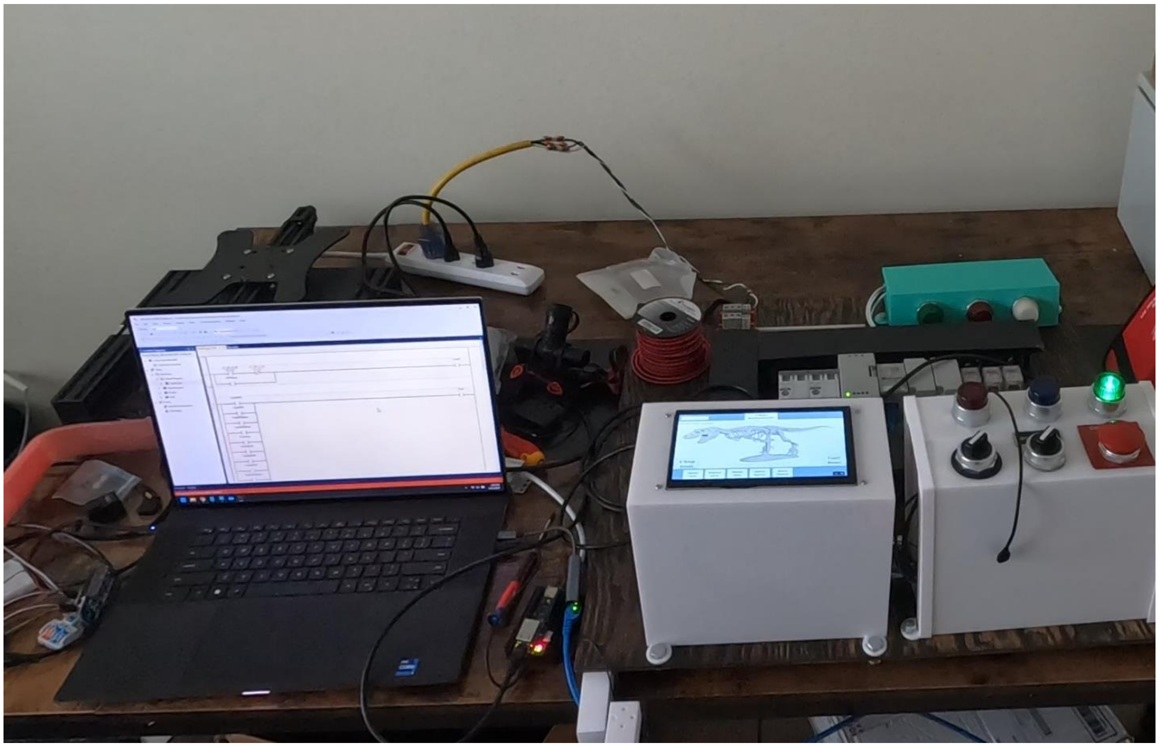


Figure 9.1: *Test Bench Setup*
By Authors

Key Features of the Testing Board:

- **HMI Panel**: The touch display and manual control box (with emergency stop, rotary selectors, and indicator buttons) allowed for comprehensive human-machine interaction testing.
- **PLC and I/O Modules**: Mounted on the DIN rail, the PLC and associated relays were tested for digital input/output response, electrical isolation, and reliability under load.
- **Power Distribution and Wiring**: Power modules and breakers were organized to simulate real-world operating conditions and ensure safe voltage regulation.
- **Network Connectivity**: Ethernet and USB interfaces facilitated communication between the W5500 module, HMI, and development laptop.

Integration Testing Procedures:

- **Full System Power-On Checks**: Verified that all subsystems powered on without faults and began proper operation cycles.
- **Communication Protocols Validation**: Ensured interoperability between HMI, PLC, Dynamixel motors, and other peripherals.
- **Fail-safe Mechanisms**: The emergency stop button was tested to ensure it correctly disabled critical systems without causing data corruption or hardware damage.

This physical setup plays a crucial role in enabling safe and repeatable testing of hardware and software integration before final deployment and greatly accelerates the troubleshooting process.

9.1 HMI:

The primary testing goals were:

- **Responsiveness**: Button presses, and mode selections were tested for latency. Results showed a consistent 150ms or less response time over Ethernet communication with the PLC.
- **Display Accuracy**: The touchscreen interface was tested in various lighting conditions. The interface remained clearly visible, with touch detection accuracy within a tight tolerance.
- **Communication**: Commands issued from the HMI to the PLC and feedback from sensors were correctly updated on-screen in real-time.

Issues encountered:

- Connectivity between PLC and Modbus; was resolved via internal tools inside Inductive Automation software.
- Proper installation of software; was resolved via official online forums and Inductive Automation instructions.

9.2 PLC:

The PLC served as the central controller for protection sequencing.

9.2.1: Hardware

- **Wiring**: Verified input connections from the HMI and outputs to relays controlling motor drivers and status LEDs.
- **Digital I/O Test**: All digital inputs and outputs were tested for correct voltage levels and timing. Each I/O pin was manually toggled and observed through both LED indicators and multimeter readings.

9.2.1: Software

- **Program Structure**: The PLC ladder logic was tested using simulation tools and then deployed to the physical controller.
- **Sequence Testing**: Startup, stop, emergency stop, and reset sequences were tested against expected behaviors.
- **Fault Handling**: Simulated input loss (e.g., disconnected sensor) to confirm PLC safely halted operations and flagged errors.

9.3 Dynamixel Motors:

The Dynamixel servo motors were tested for addressability, movement precision, and communication via UART.

- **Motor IDs**:
  - Neck Rotation: ID 10
  - Jaw Rotation: ID 20
  - Neck tilt: ID 40
  - Neck Swivel: ID 50
  - Body Rotate: ID 70
  - Tail Swing: ID 80
- **Testing Procedure**:
  - Each motor was commanded to move to set positions.
  - Torque feedback and position accuracy were logged.
  - Interference tests ensured that only the motor with the corresponding ID responded.
- **Results**:
  - Positional accuracy was within 1°, and no skipped commands occurred during 30-minute endurance cycles.

9.4 NEMA 17 Stepper Motor & A4988 Driver:

This subsystem was tested for motion reliability and driver protection.

- **Test Setup**: NEMA 17 motor controlled via A4988 on a breadboarded circuit with current limiting set to 0.9A.
- **Functionality Testing**:
  - Rotational direction and step size were controlled via microcontroller PWM output.
  - Step loss was measured by comparing expected vs actual rotations.
- **Observations**:
  - Micro stepping worked reliably up to 1/16 step.
  - Slight heating on the driver was mitigated with a small heatsink.

During breadboard testing, the A4988 driver frequently overheated due to poor thermal dissipation and unstable connections inherent to the temporary setup. This led to part failure in one case, prompting a design revision. To improve reliability and thermal performance, the driver was transitioned to a press-fit configuration on the final PCB, ensuring solid electrical contact and allowing the addition of a heatsink for passive cooling, and in the event of a part failure, easily hot-swappable driver boards.

Furthermore, another improvement to performance was the application of heat sink and a fan setup for active cooling. This reduced average temperatures, and allows for our average failure rate to decrease, especially due to repeated usage in a consistent time frame, which is the goal.

9.5 W5500:

The W5500 Ethernet module served as the networking interface for HMI and PLC communication.

- **Connection Test**: Validated IP address assignment via DHCP and successful ping responses.
- **Data Throughput**: Achieved a stable 10 Mbps throughput during JSON command exchange.
- **Latency**: Round-trip latency of ~20ms between HMI and controller.

No packet loss or connection dropouts were observed during sustained use.

9.6 SAMD21:

The SAMD21 microcontroller was evaluated as an intermediary controller and sensor processor.

- **Firmware**: Custom firmware was flashed over USB using Arduino framework.
- **Serial Communication**: Verified both UART and I2C communication.
- **GPIO Testing**:
  - Output toggling was verified with logic analyzer.
  - Analog inputs were tested with a potentiometer and multimeter comparison.

Test results showed stable operation with minimal current draw (~25 mA active).

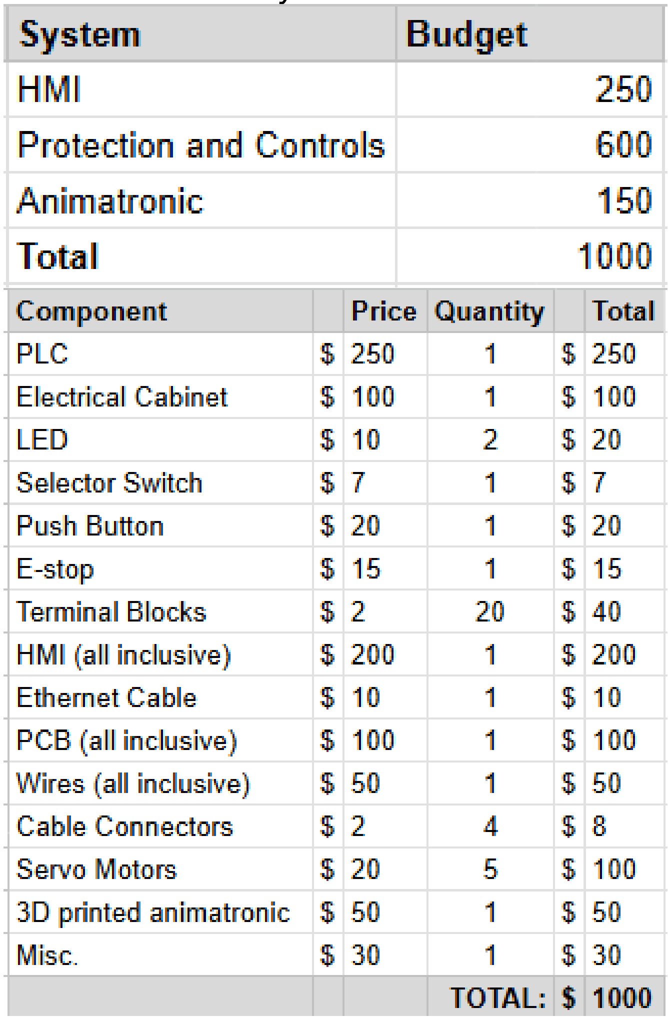
Chapter 10: Administrative Content

10.1 Budget & BOM:

One of the major goals of this project is to minimize costs while still providing industrial level quality by using industry-standard control systems. To do this, we have budgeted a total of $1000 dollars across all members, leading to a split of $250. This project has been sponsored by ROBOTIS to provide the servo motors used.

In preparation to hit this budget goal, some preliminary research must be conducted to assess the current market prices and create a bill of materials (BOM). The prices used in this BOM are based off real products, however they are kept generalized and prices are rounded to accommodate for changes in final product or a future sponsorship.

Table 10.1: *Preliminary Budget & BOM as of 2.7.25*
By Authors



System	Budget
HMI	250
Protection and Controls	600
Animatronic	150
Total	1000

Component		Price	Quantity		Total
PLC	$	250	1	$	250
Electrical Cabinet	$	100	1	$	100
LED	$	10	2	$	20
Selector Switch	$	7	1	$	7
Push Button	$	20	1	$	20
E-stop	$	15	1	$	15
Terminal Blocks	$	2	20	$	40
HMI (all inclusive)	$	200	1	$	200
Ethernet Cable	$	10	1	$	10
PCB (all inclusive)	$	100	1	$	100
Wires (all inclusive)	$	50	1	$	50
Cable Connectors	$	2	4	$	8
Servo Motors	$	20	5	$	100
3D printed animatronic	$	50	1	$	50
Misc.	$	30	1	$	30
			TOTAL:	$	1000

Table 10.2: *Final BOM with accurate pricing*
By Authors

Animatronic					
Component		**Price**	**Quantity**		**Total**
Base w/ Stepper Motor	$	58.58	1	$	58.58
Stepper Motor Driver	$	1.38	1	$	1.38
3d Printer PLA	$	20	1	$	20
Misc Hardware	$	10	1	$	10
A4988 Stepper Driver	$	1.21	1	$	1.21
Reed Switches	$	1.89	5	$	9.45
Total PCB Budget (Boards and Components)	$	50	1	$	50
Creality Ender 3 V2 Bed Slinger (NEMA 17 Stepper)	$	38.58	1	$	38.58
12VDC Power Supply	$	15.27	1	$	15.27
Enclosure					
Component		**Price**	**Quantity**		**Total**
USB to USB Cabinet Mount	$	6.69	2	$	13.38
Switch Power Socket 10A Fuse	$	3.37	1	$	3.37
AB Push Buttons w/ LED	$	8.46	4	$	33.84
AB Switch Lock 2-POS	$	12.78	1	$	12.78
AB Selector Switch 3-POS	$	26.63	1	$	26.63
AB LED	$	10.65	3	$	31.95
Ethernet Switch w/ 12VDC Supply	$	35.13	1	$	35.13
Ethernet To Ethernet Cabinet Mount	$	4.43	2	$	8.86
DIN Rail Terminal Blocks	$	30	1	$	30
Misc Wires	$	15.4	1	$	15.4
Rack Cable Raceways	$	19.16	1	$	19.16
VEVOR NEMA Steel Enclosure, 20 x 12 x 10'	$	65.63	1	$	65.63
Misc Ethernet Cables	$	20	1	$	20
HMI					
Component		**Price**	**Quantity**		**Total**
HMI ED-HMI3010-101C	$	174.2	1	$	174.2
AC Socket Socket Plug Panel	$	12.05	1	$	12.05
CanaKit 3.5A Raspberry Pi 4	$	9.99	1	$	9.99

Mini Industrial 5-Port Gigabit Ethernet Switch	$	34.8	1	$	34.8
Protection And Controls					
Component		**Price**	**Quantity**		**Total**
Allen-Bradley 1492-CB1 Circuit Breaker 1 Amp	$	4.21	3	$	12.63
Allen Bradley Micro820-AWB PLC & Micro800-PSU	$	133.13	1	$	133.13
			TOTAL:	***$***	***897.4***
		POST			***POST***
		TAX			***TAX***
		& SHIP			***& SHIP***

As previously stated, this project has been sponsored by ROBOTIS to provide Dynamixel servo motors for the animatronic. This has alleviated the total budget burden amongst team members, however it does still impact the overall budget of the project, as if this were a product on market, consumers would be required to pay the total price, for all other materials and the Dynamixel motors.

Table 10.3: *Motors Provisioned by ROBITIS*
By Authors

Dynamixel Motors					
Component		**Retail**	**Quantity**		**Total**
2XL430-W250-T	$	142.89	1	$	142.89
XL430-W250-T	$	54.89	2	$	109.78
XC330-T288-T	$	98.89	2	$	197.78
			TOTAL:	***$***	***450.45***

So, combining both BOMs, we would get our total deliverable cost for a prospective client:

Table 10.4: *Total Cost for Product Deliverable*
By Authors

Total Deliverable Cost	
Team BOM	*$897.4*
ROBOTIS BOM	*$450.45*
TOTAL COST:	**$1,347.85**

10.2 Milestones:

Keeping track of milestones is a key tool for success when completing long-term projects. Itemizing project documentation, design and implementation tasks into clear lists with set dates make the goal posts clear and organize workload for team cohesion.

Table 10.5: *Documentation, Design, and Implementation Milestone Tracker By Authors*

Documentation			
Start Date	**End Date**	**Task**	**Description**
1/6/25	1/14/25	Recruitment	Austin Berg, Melvin Guzman, Tony Torres and Armando Diaz were recruited.
1/15/25	1/24/25	D&C / Review Committee	Complete Divide and Conquer (D&C) and finalize review committee for project.
1/27/25	2/7/25	D&C Revision	After project approval, revise and submit the final D&C document.
2/8/25	3/24/25	Midterm Milestone Report	Complete 60-page mid-project report.
3/25/25	4/4/25	Midterm Milestone Report Revision	Revise and submit final 60-page mid-project report.
4/5/25	4/22/25	Final Report	Complete and submit the final 120-page report.
Design and Implementation			
Start Date	**End Date**	**Task**	**Description**
2/7/25	3/24/25	Final Component Selection	All team members should have all components selected for use.
2/7/25	4/22/25	Protection and Controls System Design	Main design should be completed.
2/7/25	4/22/25	HMI Design	Main design should be completed.
2/7/25	4/22/25	Power Flow and PCB Design	Main design should be completed.
4/22/25	7/22/25	Testing, programming and adjusting	Testing and adjusting of implemented designs to ensure working product.
4/22/25	7/22/25	Prototype Completion	Completed prototype with working systems and functions as specified.

10.3 Full Distribution of Work:

The worktable below outlines the task distribution among group members, with each individual assigned responsibility for a specific system. While primary accountability lies with designated members, the design and functionality of each system are inherently dependent on others. As a result, continuous collaboration is maintained across all stages of the project to ensure seamless integration between systems. This collaborative approach guarantees that all members remain informed and actively engaged in the development and implementation processes, regardless of direct responsibility.

Table 10.6: *Distribution of Worktable*
By Authors

Electrical Engineer	**Responsibilities**
Austin Berg	Protection System Design and Implementation
	Animatronic Design and Implementation
	Manufacturing and Mechanical Installation
	Project Lead
Electrical Engineer	**Responsibilities**
Melvin Guzman	Motor Control Design and Implementation
	Animatronic Design and Implementation
	Sensor Design, Implementation and Management
	PCB Design and Manufacturing
Electrical Engineer	**Responsibilities**
Tony Torres	Power Supply Design and Implementation
	Voltage Regulator Design
	Documentation Management and Collaboration
	PCB Design
Computer Engineer	**Responsibilities**
Armando Diaz	Human Machine Interface Design
	Networking System Design and Management
	Peripheral Design
	PCB Design

Chapter 11: Conclusion

This project marks a significant milestone in the development of an integrated control and communication system, combining mechanical actuation, electronic control, and human-machine interface protection into a cohesive and scalable design. Throughout the design process, thoughtful consideration was given to system architecture, component selection, and industry best practices: ensuring a foundation that is not only functional but robust, adaptable, and maintainable.

A key strength of this system is its alignment with widely accepted industrial standards. From the choice of input/output wiring configurations to the implementation of standardized communication protocols, our design emphasizes compatibility, reliability, and ease of integration within broader automation systems. These decisions promote interoperability and long-term flexibility, allowing for seamless adaptation to future applications or environmental demands.

While the current stage focuses on hardware development and design documentation, the groundwork laid here establishes a clear pathway to full implementation. The next phase will involve system assembly, testing, and refinement, validating functionality under real-world conditions and optimizing performance across all subsystems.

This effort not only addresses the immediate goals of the project but also contributes to a broader understanding of multidisciplinary design in modern engineering systems. It reflects the team's ability to balance theoretical knowledge with practical execution, leveraging collaboration and iterative development to tackle complex challenges.

As we advance toward implementation, our focus will shift to integrating the individual subsystems into a cohesive whole, ensuring that electrical, mechanical, and software components operate in unison. This phase will also allow us to validate design assumptions, identify areas for optimization, and gather critical performance data. The transition from concept to physical prototype is not only a test of our technical approach, but also a demonstration of the collaborative engineering process that underpins this project. With a clear roadmap, a solid foundation, and a commitment to continuous improvement, our team is well-equipped to bring this system to life and deliver a solution that meets both functional requirements and long-term reliability standards.

Appendix A - Reference

[1] Microchip Technology Inc. *SAM D21E / SAM D21G / SAM D21J Datasheet*. 2017. www.microchip.com/content/dam/mchp/documents/OTH/ProductDocuments/DataSheets/40001882A.pdf

[2] WIZnet Co., Ltd. *W5500 Datasheet*. Version 1.0.2, November 2013. https://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/W5500_datasheet_v1.0.2_1.pdf

[3] Rockwell Automation. *Micro800 Programmable Controllers Technical Data*. Publication 2080-TD001D-EN-P, 2023. https://literature.rockwellautomation.com/idc/groups/literature/documents/td/2080-td001_-en-p.pdf

[4] Texas Instruments. *LM2576 SIMPLE SWITCHER® 3A Step-Down Voltage Regulator Datasheet*. SNVS107C, Revised April 2013. https://www.ti.com/lit/ds/symlink/lm2576.pdf

[5] ROBOTIS. *DYNAMIXEL XC330-T288-T Manual*. https://emanual.robotis.com/docs/en/dxl/x/xc330-t288/

[6] ROBOTIS. *DYNAMIXEL 2XL430-W250-T Manual*. https://emanual.robotis.com/docs/en/dxl/x/2xl430-w250/

[7] ROBOTIS. *DYNAMIXEL XL430-W250-T Manual*. https://emanual.robotis.com/docs/en/dxl/x/xl430-w250/

[8] EDATEC. *ED-HMI3010-101C Datasheet*. Mouser Electronics, 2024. https://www.mouser.com/datasheet/2/1563/HMI3010_10datasheet_en-3446798.pdf

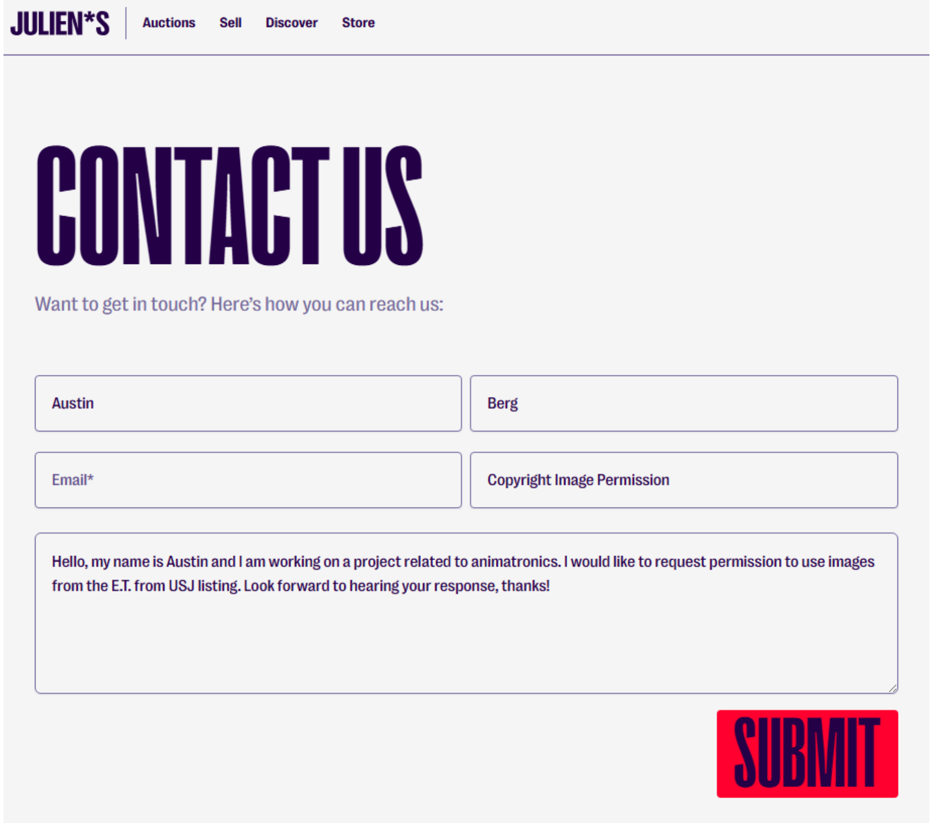
[9] Julien's Auctions. *E.T. the Extra-Terrestrial Universal Studios Japan "E.T. Adventure" Animatronic*. Lot 11, 2024. https://www.juliensauctions.com/en/items/100555/e-t-the-extra-terrestrial-universal-studios-japan-e-t-adventure-animatronic

[10] WDWMAGIC. *Animatronics – Monsters Unchained: The Frankenstein Experiment at Universal Epic Universe*. October 29, 2024. https://www.wdwmagic.com/wdwmagic-plus/universal-orlando-resort/gallery/29oct2024-animatronics---monsters-unchained-the-frankenstein-experiment-at-universal-epic-universe/57871.htm

[11] MakerBot. *T-Rex Skeleton*. Thingiverse Thing:275091, 2013. https://www.thingiverse.com/thing:275091

Appendix B - Copyright Permission

Figure 2.1: Requested



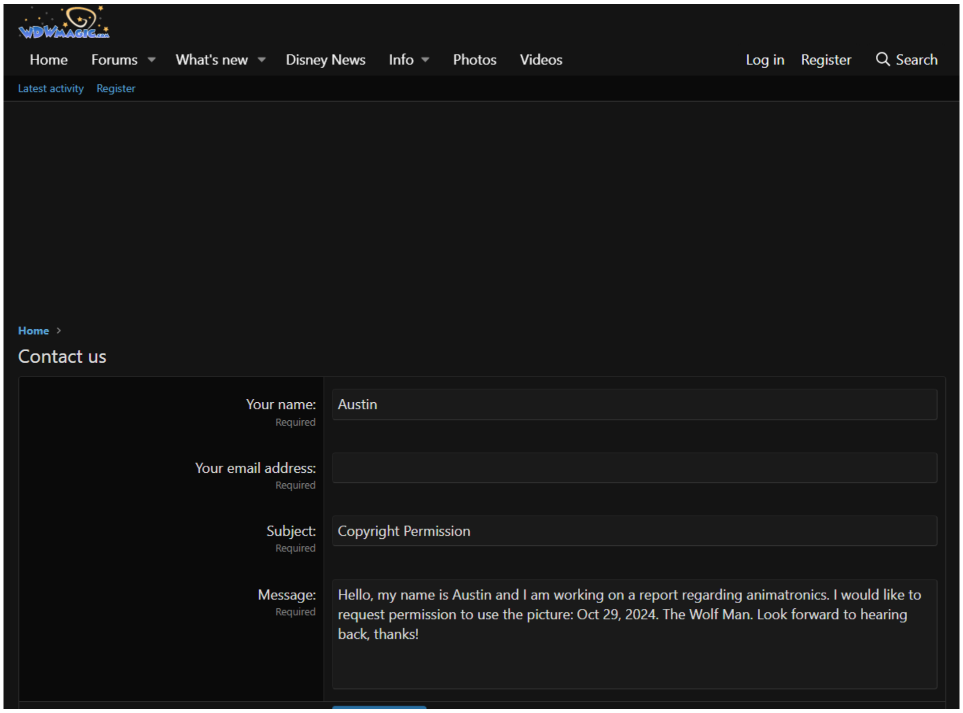
CONTACT US

Want to get in touch? Here's how you can reach us:

Austin	Berg
Email*	Copyright Image Permission

Hello, my name is Austin and I am working on a project related to animatronics. I would like to request permission to use images from the E.T. from USJ listing. Look forward to hearing your response, thanks!

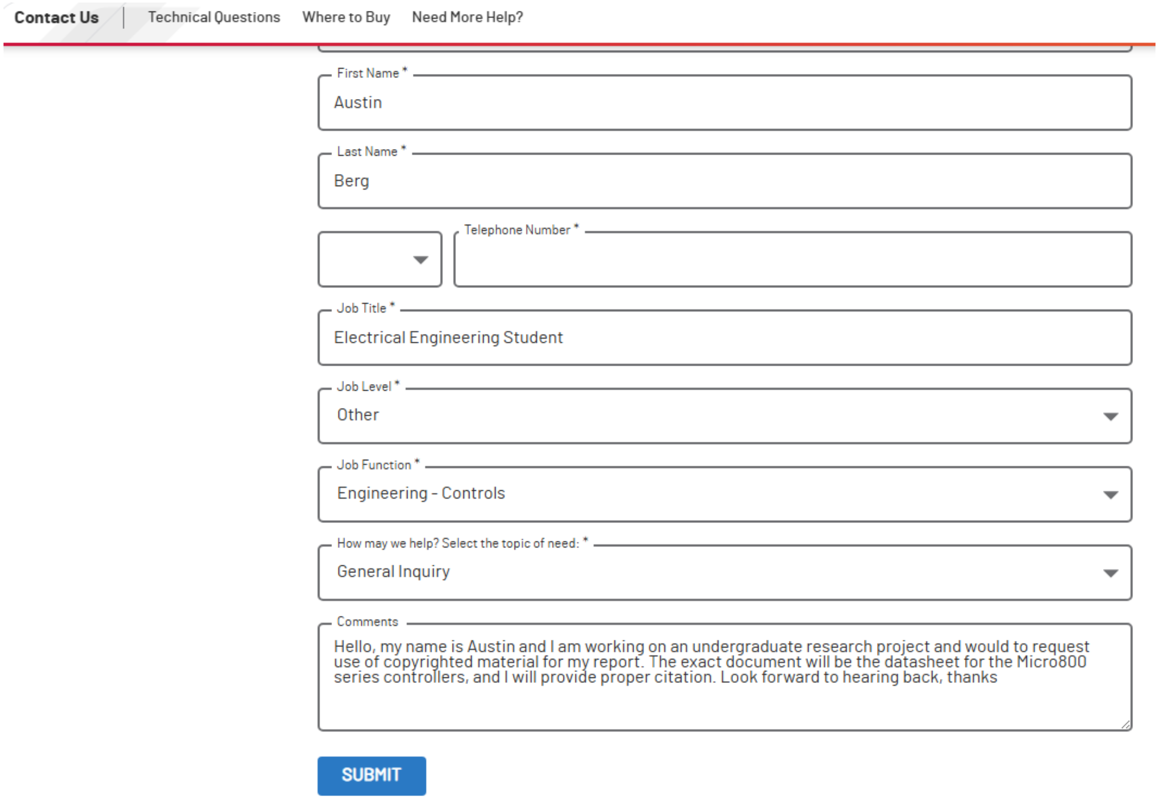
Figure 2.2: Requested



Contact us

Your name: Required	Austin
Your email address: Required	
Subject: Required	Copyright Permission
Message: Required	Hello, my name is Austin and I am working on a report regarding animatronics. I would like to request permission to use the picture: Oct 29, 2024. The Wolf Man. Look forward to hearing back, thanks!

Figure 6.4: Requested



First Name *: Austin

Last Name *: Berg

Telephone Number *:

Job Title *: Electrical Engineering Student

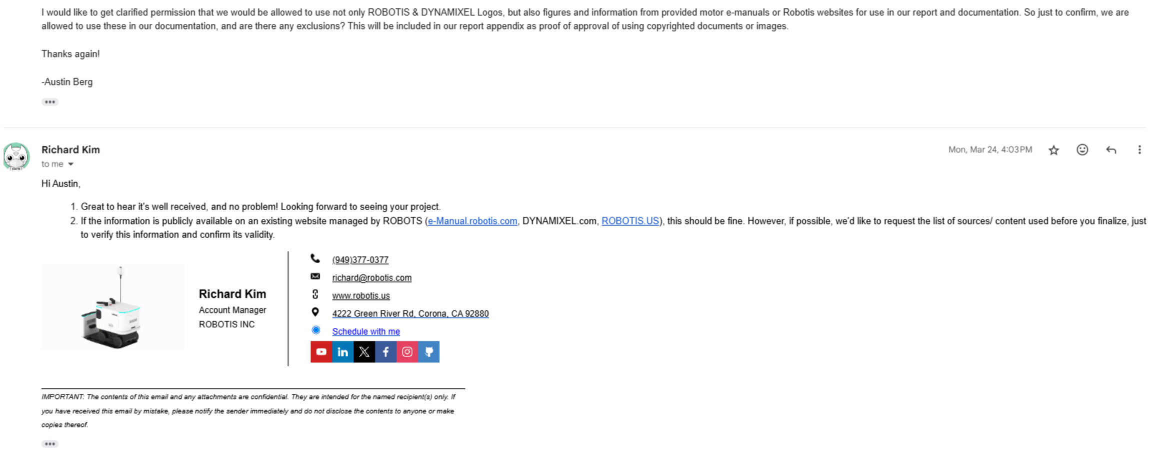
Job Level *: Other

Job Function *: Engineering - Controls

How may we help? Select the topic of need: *: General Inquiry

Comments: Hello, my name is Austin and I am working on an undergraduate research project and would to request use of copyrighted material for my report. The exact document will be the datasheet for the Micro800 series controllers, and I will provide proper citation. Look forward to hearing back, thanks

Figure 7.1: Approved



I would like to get clarified permission that we would be allowed to use not only ROBOTIS & DYNAMIXEL Logos, but also figures and information from provided motor e-manuals or Robotis websites for use in our report and documentation. So just to confirm, we are allowed to use these in our documentation, and are there any exclusions? This will be included in our report appendix as proof of approval of using copyrighted documents or images.

Thanks again!

-Austin Berg

Richard Kim
to me

Mon, Mar 24, 4:03PM

Hi Austin,

1. Great to hear it's well received, and no problem! Looking forward to seeing your project.
2. If the information is publicly available on an existing website managed by ROBOTS (e-Manual.robotis.com, DYNAMIXEL.com, ROBOTIS.US), this should be fine. However, if possible, we'd like to request the list of sources/ content used before you finalize, just to verify this information and confirm its validity.

Richard Kim
Account Manager
ROBOTIS INC

☎ (949)377-0377
richard@robotis.com
www.robotis.us
4222 Green River Rd, Corona, CA 92880
Schedule with me

IMPORTANT: The contents of this email and any attachments are confidential. They are intended for the named recipient(s) only. If you have received this email by mistake, please notify the sender immediately and do not disclose the contents to anyone or make copies thereof.