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:...4

2.4 Features and Functionality: 6

2.5 Engineering Specifications: 7

2.6 Prototype Illustration: 8

2.7 Distribution of Work: 8

2.8 Hardware Diagram:..9

2.9 Software Diagram: 9

2.10 House of Quality: 10

Chapter 3: Research and Investigation 11

3.1 Technology Comparison: 11

3.1.1 Protection Systems and Programmable Logic Controllers 11

3.1.2 Human-Machine Interfaces (HMIs)..11

3.1.3 Microcontroller Units (MCUs), Field Programmable Gate Arrays (FPGAs), & Application-Specific Integrated Circuits (ASICs) 14

3.1.4 Motors & Motor Controller 20

3.1.5 Manufacturing Technologies 22

3.1.6 Sensors...23

3.1.7 Supervisory Control and Data Acquisition; Alarming and Faults..................................25

3.1.8 HMI Software...26

3.1.9 HMI Communication 27

3.1.10 Programming Languages..27

3.1.11 Fuses 30

3.1.12 Electrical Enclosures 30

3.2 Technology Selection: 32

3.2.1 Protection Systems and Programmable Logic Controllers 32

3.2.2 Human-Machine Interfaces (HMIs)..32

3.2.3 Microcontroller Units (MCUs)..32

3.2.4 Motors 32
3.2.5 Manufacturing Technologies 32
3.2.6 Sensors 32
3.2.7 Supervisory Control and Data Acquisition; Alarming and Faults 32
3.2.8 HMI Software 32
3.2.9 HMI Communication 33
3.2.10 Programming Languages 33
3.2.11 Fuses 33
3.2.12 Electrical Enclosures 33
3.3 Part Comparison: 34
3.3.1 Programmable Logic Controllers (PLCs) 34
3.3.2 Human-Machine Interfaces (HMIs) 37
3.3.4 Motors & Motor Controller 39
3.3.6 Sensors 40
3.3.11 Fuses 42
3.3.12 Electrical Enclosures 43
3.4 Part Selection: 43
3.4.1 Programmable Logic Controllers (PLCs) 44
3.4.2 Human-Machine Interfaces (HMIs) 44
3.4.4 Motors and Motor Control 44
Chapter 4: Standards and Design Constraints 45
4.1 Industrial Standards: 45
4.1.1 Modbus 45
4.1.2 IEC 61131-3 (PLC Programming Standards) 46
4.1.3 Ethernet/IP Communication Protocol 47
4.2 Design Constraints: 48
4.2.1 Cost-Effectiveness 48
4.2.2 Scalability and Modularity 49
4.2.3 Real-Time Performance 50
4.2.4 Time Constraints 51
4.2.5 Environmental Constraints 52

Chapter 5: Comparison of ChatGPT with other similar platforms 53
5.1 What is ChatGPT and how does it work?: 53
  5.1.1 Pros: 54
  5.1.2 Cons: 55
5.2 Other AI Platforms: 56
  5.2.1 Claude 56
  5.2.2 Microsoft Copilot 56
  5.2.3 Gemini 57
  5.2.4 Perplexity 57
  5.2.5 Pi 57
  5.2.6 AlphaCode 57
  5.2.7 Jasper 57
5.3.1 How ChatGPT will be used in our project: 58
Chapter 9: System Testing and Evaluation 60
9.1 TBD: 60
  9.1.1 Motor Tuning 60
Chapter 10: Research and Investigation 61
10.1 Budget & BOM: 61
10.2 Milestones: 62

List of Tables:

tbd

List of Figures:

tbd

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

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, allowing for both technical and artistic growth. Moreover, as these systems continue to evolve, they could offer new applications in education or healthcare; beyond traditional entertainment.

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.

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.

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:

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 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.

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.

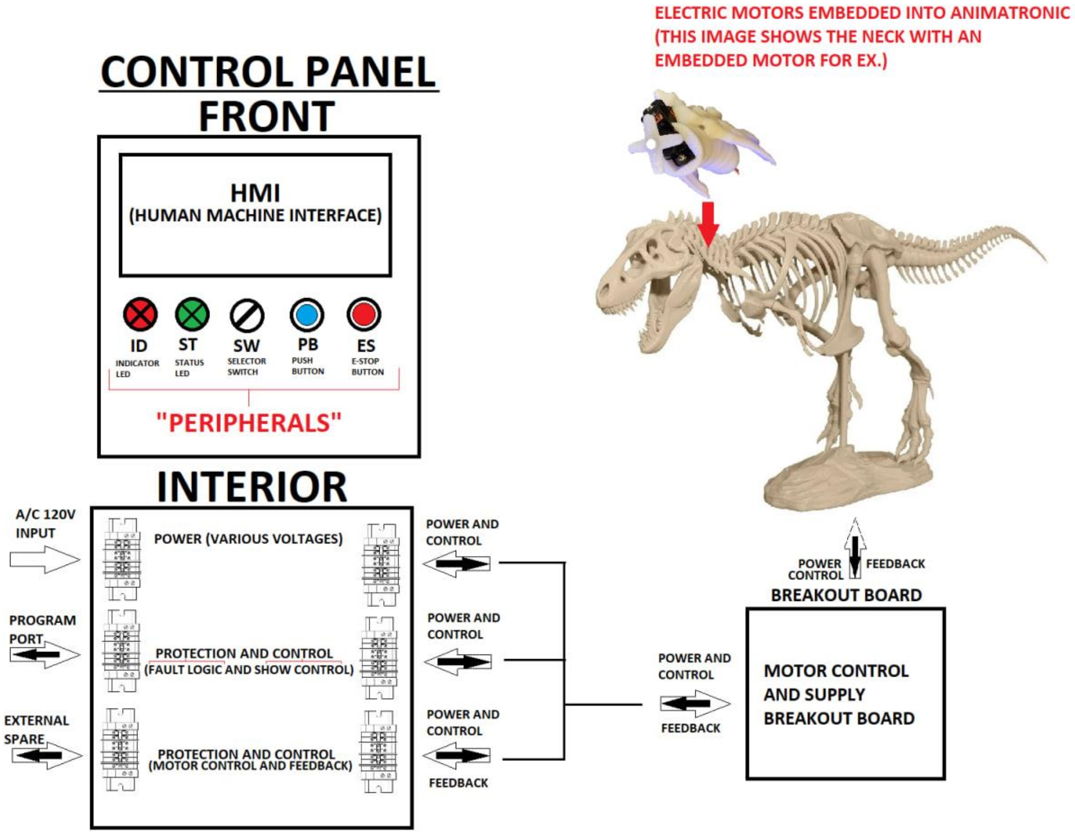


Figure 2.1: Prototype Diagram by Authors

2.7 Distribution of Work:

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:

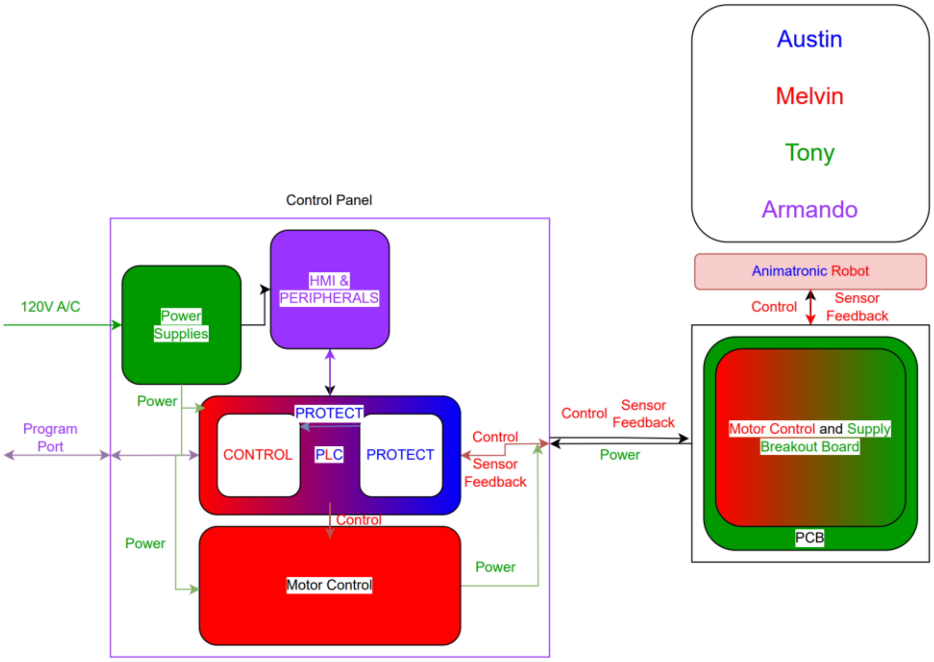


Figure 2.2: Hardware Diagram by Authors

2.9 Software Diagram:

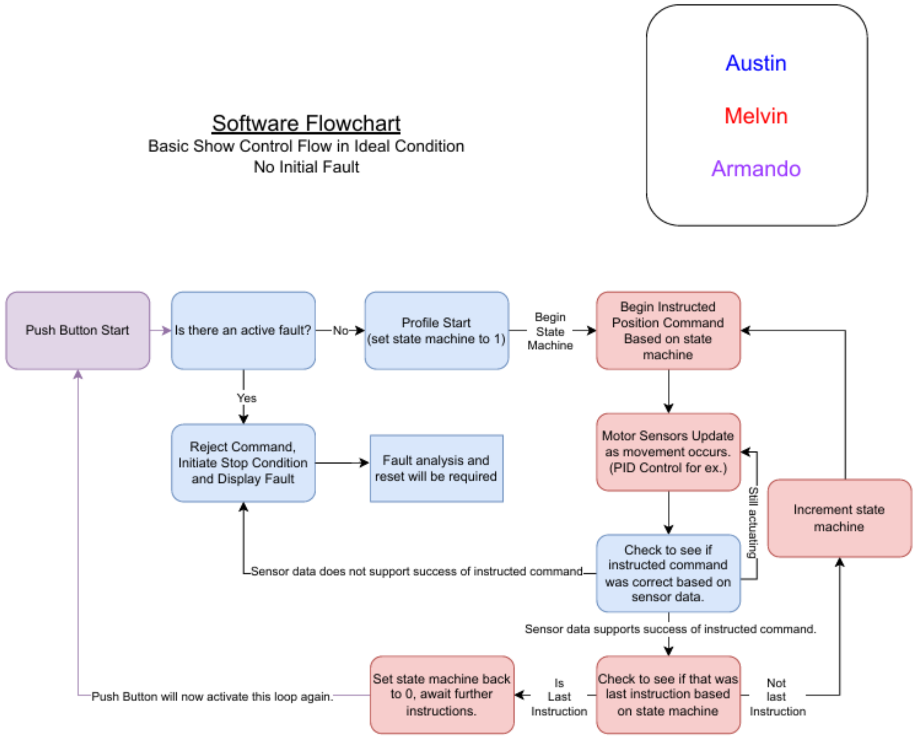
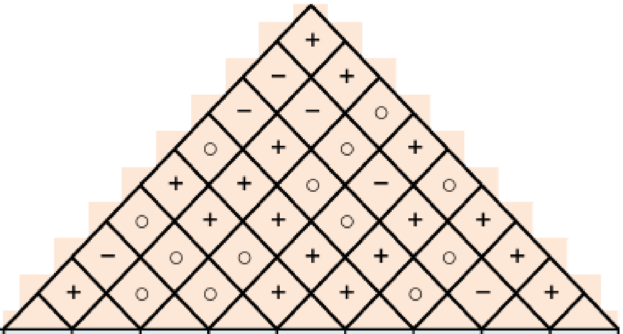

Austin
Melvin
Armando
Software Flowchart
Basic Show Control Flow in Ideal Condition
No Initial Fault
Push Button Start
Is there an active fault?
No
Profile Start (set state machine to 1)
Begin State Machine
Begin Instructed Position Command Based on state machine
Yes
Reject Command, Initiate Stop Condition 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.
Push Button will now activate this loop again.
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

Figure 2.3: 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 2.4: House of Quality by Authors

Chapter 3: Research and Investigation

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 and limitations.

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.

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
    - Easier Programming

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.

When looking for an interface to use, the options aren't so cut and dry. An HMI (Human Machine Interface) doesn't necessarily define a specific technology or item, but rather it merely describes that within the system there is a medium where a human user can operate to interact with the system. So, the technology decided here would define the main implementation of our control design for the system. Of the many used in the automation industry, the following three choices gathered with their qualities tabled below: a Touch-Display panel, the use of Remote Devices, and a Web Browser Application.

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

Table 3.1: HMI Technology Comparison
By Authors

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

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.

3.1.4 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.

To clarify, both servo and stepper motors may be used for the project. Servos are preferable for most functions for their fine control, but stepper motors may be used for functions that do not require such fine control.

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).

Programmable logic controller integration is essential to the project scope. Therefore, the motors should have the ability to interface with a Human Machine Interface (HMI) to look at real-time data and allow for local control.

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.

By leveraging these 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 velocity. 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.

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

Due to our use of a PLC, the software used for our design had to be compatible with said PLC, which happened to narrow what software were available to be used. So, the software used for PLCs aim to accomplish one, if not multiple, aspects of PLC use: Industrial Automation, Product Manufacturing, and Process Control. Now each of these serve a drastically different purpose compared to the other, each being somewhat self-explanatory. Industrial Automation being the use of machinery to accomplish tasks that would normally be completed by a person. The level of logic needed to perform such tasks does meet what is desired for our design but may be high enough to be considered excessive for what the animatronic is expected to do. Product Manufacturing would consist of the monitoring of machine processes as they perform the same action. Though monitoring is ideal for our animatronic, in particular to read the states of motors when idle or active, the monitoring used in this case is more a data collection to give a preview of how much product was produced. Lastly, Process Control is a way to manage the performance and output of a given process by adjusting parameters to achieve the desired result. This falls most in line with what is intended for our design. The animatronic would consist of several motors and having the ability to fine tune how these motors work is essential to optimize its performance and to be able to troubleshoot any issues. As stated previously, many PLC software mean to accomplish a combination of the objectives for PLCs. So, it is ideal that the PLC software used in our design is best for Industrial Automation, Process Control, or the combination of both.

3.1.9 HMI Communication

For HMI communication, there are two types of connections used: serial and ethernet. Serial communication is best when two devices are in conversation to only each other for that connection. Ethernet, on the other hand, is best when there are multiple devices in conversion, especially if the data being sent needs to be transferred to several of these connected devices.

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

Table x.x: Comparison between Serial and Ethernet
By Authors

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. Java applications can be run on different hardware and operating systems using the Java Virtual Machine (JVM), making it ideal for web-based automation interfaces, control dashboards, and enterprise automation systems.

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.

3.1.11 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).

Properly selecting and configuring an electrical enclosure ensures that an industrial control system remains reliable, safe, and easily maintainable, making it an essential component of animatronic and automation applications.

3.2 Technology Selection:

This section presents our technology selections for each applicable area of the animatronic system. The chosen technologies are based on a thorough evaluation of their advantages, limitations, and alignment with the project’s goals of reliability, cost-effectiveness, scalability, and performance. Each selection is justified by its ability to meet the system’s requirements and deliver the desired functionality.

3.2.1 Protection Systems and Programmable Logic Controllers

We will be utilizing a PLC as our protection system.

3.2.2 Human-Machine Interfaces (HMIs)

We will be utilizing an HMI that integrates with a PLC as our protection system.

3.2.3 Microcontroller Units (MCUs)

We will be utilizing a SAMD 21 MCU as our motor controller.

3.2.4 Motors

We will be utilizing stepper motors and servo motors.

3.2.5 Manufacturing Technologies

We will be utilizing additive manufacturing in the form of 3d-printing for the animatronic figure.

3.2.6 Sensors

We will be using limit switches, proximity sensors using hall effect magnet sensing technology, current sensors and angular position sensors.

3.2.7 Supervisory Control and Data Acquisition; Alarming and Faults

We will be using an HMI software with SCADA, alarming and fault reporting capabilities.

3.2.8 HMI Software

As discussed before, 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.

3.2.9 HMI Communication

We will be using Ethernet TCP/IP as our HMI communication protocol.

3.2.10 Programming Languages

We will be coding our MCU for motor control is C, and our PLC in Structured Text and Ladder Logic.

3.2.11 Fuses

We will be using Fuses of varying amperages to protect our devices in the case of power spikes or failures.

3.2.12 Electrical Enclosures

We will be using a NEMA and IP-rated electrical enclosure to provide a safe and secure environment for the main control system.

3.3 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.

3.3.1 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. The comparison is based on factors such as cost, performance, scalability, ease of use, and software pricing.

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.

Feature / Criteria	Allen-Bradley Micro800 Series	Allen-Bradley CompactLogix 5370 Series	Allen-Bradley ControlLogix Series	Siemens S7-1200 Series
Cost	Low	Medium	High	Medium
Processing Power	Moderate	High	Very High	High
Memory Capacity	Limited	High	Very High	High
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
Ease of Use	High	Medium	Low	Medium
Scalability	Moderate	High	Very High	High
Software Cost	Free	Paid	Paid	Paid

Table x.x: PLC Comparison
By Authors

3.3.2 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.

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

Table x.1: Touch-Display HMI Technology Comparison.
By Authors

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 Table x.x. 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 too undesirable and no longer considered.

	# of Cores	Core Speed	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

Table x.2: HMI Part Selections
By Authors

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.

	MCU	RAM	Storage	Screen Size	Operating Voltage	Cost
ED-HMI3010-101C	Pi 5	4-8GB	64GB	10.1”	5VDC	$157
ED-HMI3020-101C	Pi 5	4-8GB	256GB	10.1”	5VDC	$215
PPC-CM5-101	Pi 5	4-8GB	256GB	10.1”	15VDC	$329

Now with the several Raspbian based HMI, the three presented in Table x.x 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.

3.3.4 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.

3.3.6 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.

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.

#	Fuse #1	Fuse #2
Quantity	3	1

Price	$10.15	$8.55

Table x.x: Fuse Comparisons
By Authors

3.3.12 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 x.x: Electrical Enclosure Price Comparisons
By Authors

3.4 Part Selection:

This section presents the final selection of parts and components for the animatronic system, based on the comparisons and evaluations conducted in the previous sections. The chosen components are selected to meet the project's goals of reliability, cost-effectiveness, scalability, and performance while adhering to the design constraints. Each selection is justified by its alignment with the project's requirements and its ability to deliver the desired functionality.

3.4.1 Programmable Logic Controllers (PLCs)

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.

3.4.2 Human-Machine Interfaces (HMIs)

We have decided that the HMI3010 is the HMI that we will be using for our design.

3.4.4 Motors and Motor Control

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 motors to use for the project. ROBOTIS was able to

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:

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:

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.

Hardware Requirements:
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.

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.

Chapter 5: Comparison of ChatGPT with other similar platforms

Introduction:

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).

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.

5.1 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.

5.1.1 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:

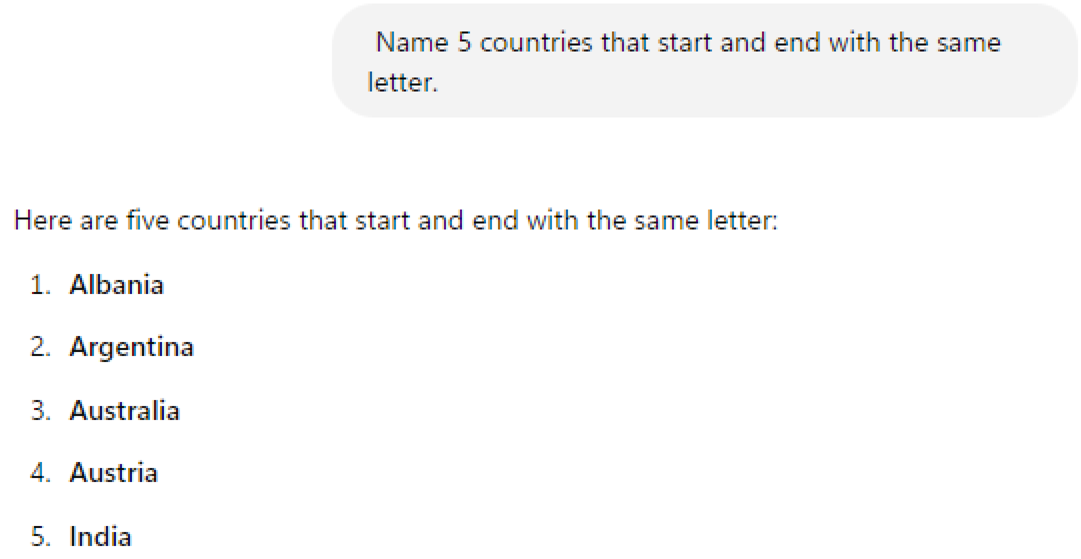
1. Brainstorming and idea generation
2. Assist in debugging code
3. Mathematical assistance

5.1.2 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.”



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**

Figure 5.1: ChatGPT Error Example
By Authors

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.

5.2 Other AI Platforms:

5.2.1 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.

5.2.2 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 datasheets, 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.

5.2.3 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.

5.2.4 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.

5.2.5 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.

5.2.6 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.

5.2.7 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.

5.3.1 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. ChatGPT could be invaluable in providing sample code in C or python for microcontrollers such as the ESP32 or the MSP430FR6989. 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. In addition to this, the touchscreen HMI interface also needs to be coded. 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.

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 9: System Testing and Evaluation

9.1 TBD:

9.1.1 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.

	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

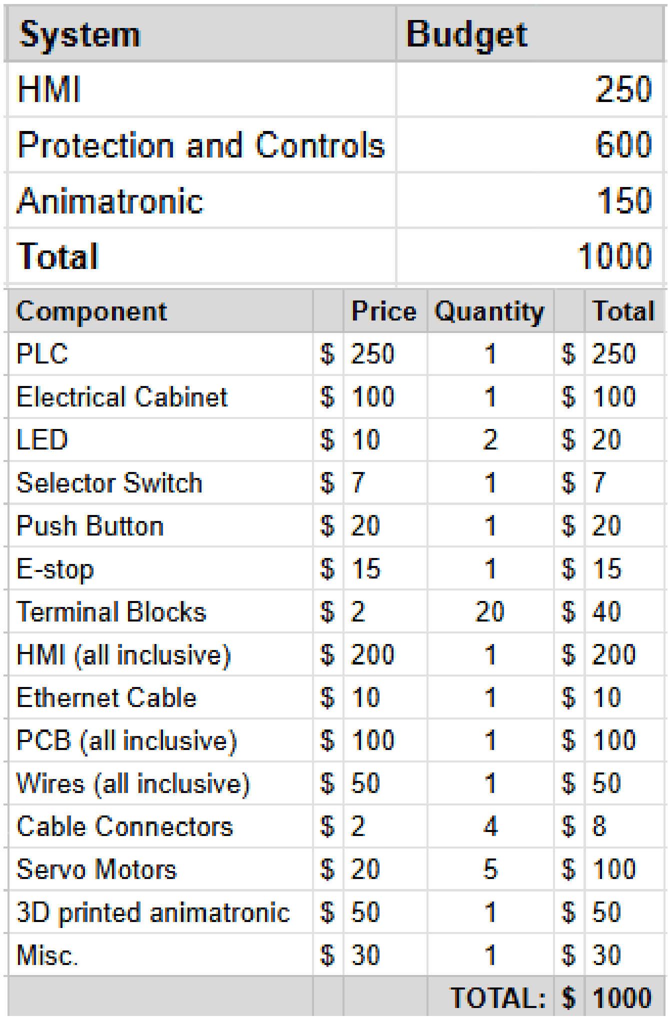
Table x.x PID Tuning
By Authors

Chapter 10: Research and Investigation

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. 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.



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 10.1: Preliminary Budget & BOM as of 2.7.25 by Authors

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.

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.

Table 10.3: Documentation, Design, and Implementation Milestone Tracker By Authors