GRIP

Glove for Real-time Imitation and Performance

Department of Electrical Engineering and Computer Science
University of Central Florida
EEL 4914 Senior Design I
Group 21

Group Advisor

Dr. Saleem Sahawneh

Review Committee

Dr. Wei Sun, Dr. John Adeo, Dr. Sudeshna Pal

Group Members

Kane McAgy, EE

Christopher Marcoccia, EE

Rian Lowery, CPE

Griseld Xhengo, EE

Table of Contents

Chapter 1 Executive Summary....... 1
Chapter 2 Project Description....... 2
2.1 Motivation and Background....... 2
2.2 Prior Related Work.......2
2.2.1 Intelligent Programmable Prosthetic Arm.......2
2.2.2 NASA R5 Valkyrie....... 3
2.2.3 Shadow Robot Dexterous Hand.......3
2.3 Project Goals and Objectives.......3
2.4 Project Features and Functions....... 5
2.5 Project Specifications.......5
2.6 Hardware Diagram.......8
2.7 Software Design.......9
2.8 House of Quality.......11
Chapter 3 Research and Investigation.......12
3.1 Technology Comparison....... 12
3.1.1 MCU or FPGA....... 12
3.1.1.1 Arduino Nano....... 13
3.1.1.2 Arduino Mega 2560.......14
3.1.1.3 Arduino Nano 33 IoT....... 14
3.1.1.4 ESP32 Dev Kit....... 14
3.1.1.5 STM32 Nucleo....... 15
3.1.1.6 Teensy 4.1.......15
3.1.1.7 Raspberry Pi Pico....... 15
3.1.2 Smart Servos or Micro Linear Actuators....... 16
3.1.2.1 MG996R.......18
3.1.2.2 Micro PWM Servos....... 18
3.1.2.3 Digital PWM Servos....... 18
3.1.2.4 TTL Bus Smart Servos.......19
3.1.2.5 RS-485/Can....... 19
3.1.3 Transceiver....... 20
3.1.3.1 nRF24l01.......21
3.1.3.2 ESP-NOW....... 21
3.1.3.3 Bluetooth Low Energy....... 22
3.1.3.4 XBee....... 22
3.1.3.5 LoRa.......23
3.1.5 MOSFET....... 24
3.1.5.1 BSS138....... 25
3.1.5.2 2N7002.......25
3.1.5.3 AO3400/A.......26

3.1.5.4 BSS84/A....26
3.1.5.5 TPS229xx....27
3.1.6 Batteries....28
3.1.6.1 18650 Li-ion....29
3.1.6.2 Lithium-Polymer Pack....30
3.1.6.3 LiFePO4 Cells....30
3.1.6.4 USB Power Bank....31
3.1.6.5 NiMH....31
3.1.7 Sensors....33
3.1.7.1 sEMG Electrodes....33
3.1.7.2 IMU Based Finger Pose....34
3.1.7.3 Why the ZD10-100 Flex Strips....34
3.2 Technology Selection....36
3.2.1 MCU....36
3.2.2 Transceiver....36
3.2.3 Motor....36
3.2.4 Sensors....37
3.2.5 MOSFET....37
3.2.6 Batteries....38
Chapter 4 Related Standards and Design Constraints.... 38
4.1: Industrial Standards....38
4.1.1 Biomedical Device Standards....38
IEC 61326-1 – EMC for Lab Equipment....39
ISO 13485....39
ISO 10993....39
IEC60601-1/2 Medical Electrical Safety & EMC....39
Robotics and Automation standards:....40
Health and Safety – User Bystander Protection....40
ISO 10218 -1: 2025 – Robotics Safety requirements Part 1:....40
IEC 62366-1 Usability Engineering....40
ISO /TS 15066: 2016 – Collaborative robots....40
ISO 13482: 2014....41
IEEE 1872 – 2015....41
IEEE 7007-2021....41
4.1.2 PCB Standards....41
4.1.3 Sensor and Signal Standards....41
4.1.4 Actuator....43
4.1.5 Communication and Control Standards....43
4.2 Practical Constraints....44
4.2.2 Economic....44
4.2.3 Environmental....45

4.2.4 Sensor and signal Constraints.... 45
4.2.5 Ethical.... 46
4.2.6 Sustainability.... 46
Chapter 5 Chat GPT Comparisons and Usage....47
5.1 Introduction.... 47
5.2 Case Study 1 — Protocol Selection: SPI (nRF24L01) vs BLE.... 47
5.3 Case Study 2 — Firmware Mapping and Curve-Shaping Techniques....48
5.4 Case Study 3 — System Block-Diagram Refinement....49
5.5 Case Study 4 — Visualization Tool Selection (C# Simulation).... 49
5.6 Case Study 5 — Safety Control: Servo Limits & Fail-Safe Behavior.... 50
5.7 Case Study 6 — Code Documentation & Readability....51
5.8 Case Study 7 — Literature & Prior-Art Review.... 51
5.9 Limitations of ChatGPT....51
5.10 Lessons Learned.... 52
5.11 Conclusion....52
Chapter 6 Hardware Design.... 53
6.1 Transmitter circuit design....53
6.2 Receiver circuit design....56
Chapter 7 Software and System Integration.... 58
7.1 Overview.... 58
7.1.1 Abstract.... 58
7.1.2 Software Objectives.... 59
7.1.3 Software Role in the System....59
7.2 Software Architecture and System Layers....59
Software Architecture Overview....59
7.2.1. Data Acquisition Layer (Glove Subsystem)....59
7.2.2. Processing & Mapping Layer.... 60
7.2.3. Communication & Control Layer (Robotic Arm Subsystem).... 60
7.2.4. System Flow Description....60
7.3 Updated System Overview.... 61
7.3.1. Purpose....61
7.3.2 Subsystems at a Glance.... 61
7.4 Updated System Architecture.... 64
7.5 Updated Software Flow....65
7.6 Preliminary Results & Anticipated Challenges....67
7.7 System Integration and Testing Plan (Next Semester).... 68
7.7.1. Overview.... 68
7.7.2. Current SD-1 Functional Scope.... 68
7.7.3. Software System Architecture....68
7.7.4. Receiver-Side Interpretation.... 69
7.7.5. Firmware Flow....69

7.7.6 C# Visualization Module....70
7.7.7 Data Structures & Packet Format.... 70
7.7.8 Safety Features....71
7.7.9 Testing and Verification.... 71
7.7.10. SD-2 Planned Expansion (Shoulder Integration).... 71
7.7.11. Summary.... 71
7.8 Summary and Conclusion....72
7.8.1. Summary of Software Design.... 72
7.8.2. Accomplishments to Date.... 72
7.8.3. Next Semester Objectives....72
7.8.4. Anticipated Impact....73
7.8.5. Personal Reflection.... 73
7.8.6. Conclusion.... 73
Chapter 10 Project Management and Communication Plan.... 73
10.1. Overview.... 73
10.2. Management Approach....74
10.3 Budget and Financing.... 74
10.4. Team Communication Methods.... 75
10.5. Project Documentation....75
10.6. Risk Management and Contingency.... 75
10.6. Expected Outcome.... 75
10.7. Summary of Software Design.... 76
10.8. Accomplishments to Date.... 76
10.9. Next Semester Objectives.... 76
10.10. Anticipated Impact.... 76
10.11. Personal Reflection.... 76
10.12. Conclusion.... 77

List of Tables

Table 2.1 Project Goals Overview.......... 4
Table 2.2 Arm Specifications.......... 5
Table 2.3 Glove Specifications.......... 6
Table 2.4 3D Printing Specifications.......... 7
Table 3.1.1 MCU vs. FPGA.......... 13
Table 3.1.2 MCU Comparison.......... 16
Table 3.1.3 Smart Servos vs. Micro Linear Actuators.......... 17
Table 3.1.4 Servo motor comparisons.......... 20
Table 3.1.5 Transceiver comparisons.......... 23
Table 3.1.6 MOSFET comparisons.......... 28
Table 3.1.7 Batteries.......... 32
Table 3.1.8 Sensors.......... 35
Table 4.1 Flex Sensor Device Data.......... 42
Table 4.2 Budget Table.......... 45
Table 5.5.1 ChatGPT generated comparisons.......... 50
Table 7.7.1 Verification options.......... 71
Table 10.3.1 Expenses.......... 74
Table 10.5.1 Meeting strategies.......... 75

List of Figures

Figure 2.1 Hardware Block Diagram.. 9
Figure 2.2 Software Diagram.. 10
Figure 2.3 House of Quality..11
Figure 6.1.1 flex sensors with arduino connections................................. 54
Figure 6.1.2 nRF24 to voltage regulator connections.............................. 55
Figure 6.1.3 example of BSS138 MOSFET level shifter circuit................56
Figure 6.2.1 receiver circuit (servos not included)...................................57
Figure 6.2.2 mg996r servos with arduino ports....................................... 58
Figure 7.2.1 Software flowchart for whole system...................................61
Figure 7.4.1 Firmware flowchart..65
Figure 7.5.1 Updated Firmware Flowchart.. 66

Chapter 1 Executive Summary

Glove for Real-time Imitation and Performance (GRIP) is a system being developed as a senior design project for the University of Central Florida. GRIP is a design that allows for remote operation of a human-like hand. The operator would control the hand with a glove placed on their own hand with the remote hand mimicking the movements of the glove. With the demand for exploration in environments that do not support current methods of human exploration, a system that can mimic the movements of a human hand GRIP is the start of a solution that could provide a system that would both be precise and not place humans in danger.

GRIP uses wireless transmission from the glove the operator is wearing to the hand. The hand would be powered by a battery which would allow for remote operation independent of the location of the user. The signals from the glove would be processed by a microcontroller. The hand would have motors in order for dexterity in each of the fingers and the thumb. The glove would have flex sensors for the exact position of the user's hand to translate to the remote hand. There will be optimization in order to reduce latency between the glove and the hand so that the hand would be usable.

GRIP is intended for deployment to austere environments which would benefit from a human-like system. Its design however has room for further development in applications like prosthetics or can be upgraded into a full arm or robotic system that is modular to fill the needs of the end user and the environments they are in.

This report describes the process taken to design GRIP. It includes the theory and technology that is used in both the hand and the glove. The parts selection and the benefits along with upgrades that could be included depending on the situation the product would be operational in and the budget that is available. The hardware and software used will be detailed to provide an inside look on how they are customized for GRIP. The standards for the components and design will also be outlined. Final conclusions for use and improvements will also be presented.

Chapter 2 Project Description

2.1 Motivation and Background

There are many places in the world we explore despite the harsh conditions of the area. Whether it's the crushing depths of the ocean, the vacuum of space, or simple treacherous terrain, mankind has developed many different technologies to explore these places in person and remotely. With many incredible feats on our belts, there are still some limitations, specifically with remote vehicle exploration, when interaction with the environment is limited to what a machine is programmed to do. Many examples of this kind of technological interaction with exterior environments can be seen today, such as Mars rovers with collection trays and storage for specimens or remotely piloted submersibles with controllable appendages. Most of these, however, are either operated solely by the machine's protocols or are controlled by joysticks and keyboards. The Glove for Real-time Imitation and Performance (GRIP) arm project attempts to give remote operators more direct interactions with the environment they are studying. Whether it's attached to a study vehicle or in a stationary setting, by having GRIP be controlled by a human arm, more decisive movements can be made by the remote operator that can lead to better results. The capability and vision of using this in laboratories, perhaps for medical prosthesis technology and any application that can mitigate danger of a human real arm. Instead the use of the GRIP is a great alternative to most applications where harmful substances or environments are in play. Ultimately, the prototype aims to solve real world problems in a spectrum of possibilities.

2.2 Prior Related Work

2.2.1 Intelligent Programmable Prosthetic Arm

There were many existing products and one project that were similar to the idea of our project. The project was for UCF senior design in 2014 and is called the Intelligent Programmable Prosthetic Arm (IPPA). This project had many of the same ideas and components that we thought of for our arm, but with a few key differences. The IPPA focuses exclusively on the hand, whereas our project is planned to be a whole arm as a stretch goal. Having a whole arm option to connect the hand to would allow our project to have more versatility than the IPPA. The IPPA also seems to be controlled by an app to make it move, where our arm will mimic the user's arm via a glove. Overall, this previous project gives a good starting point and can provide ideas for components and mechanisms that can be used to improve our design.

2.2.2 NASA R5 Valkyrie

An advanced example was the NASA R5 also known as Valkyrie. This was a complete humanoid robot but we focused on the arms. The arms and hands have joints which use actuators to control movement. We are trying to have more advanced movement as one of our stretch goals especially in the fingers so that each one can move on its own. The R5 provides a good idea of what is needed in the hardware to make this movement possible. This is a very polished and expensive example so we may have trouble using the same exact components or methods NASA used but we might be able to find a work around using the R5 as inspiration. The full weight of the robot is three hundred pounds and the extent is full bodied and polished by experts. Our design will appear more crude than this to its full extent.

2.2.3 Shadow Robot Dexterous Hand

Focusing in on the glove that we are planning to have control of the arm there is a product called the Shadow Robot Dexterous Hand Series. This product mainly focuses on the hand but there is an option to add an arm. The hand is a copy of a human hand with four fingers and a thumb and is able to function with good precision. We are hoping to see how this hand interacts with the glove that controls it as we are trying to do the same thing. The communication between the hand and glove and how to have as little latency as possible will be something that this product can help with.
There are many more examples of robotic hands and arms but the three above are what we are focusing on and show that the concept is achievable and may offer solutions we have not thought about or ideas on how we may design our system. The ideas in our design remain original and cost-effective in premise. The latency is one of the biggest drawbacks that we want to improve upon. Our algorithm will be within a few seconds of delay and rotation, see the house of quality table for a visual. The shadow project was on the higher end of the projects due to higher-quality materials. Despite the simplicity of the device being a glorified arm piece, ours will include a shoulder mechanism as well.

2.3 Project Goals and Objectives

The GRIP team will have several basic and advanced goals to work towards and a singular stretch goal that will push the team to their limits. Each goal will be composed of multiple objects that must be completed in order for the goal to be achieved. Each objective will test a certain aspect of the goal being worked towards. For example, the homing sequence will have each joint of the arm reach its limit and take note of the range, with each joint's homing functionality being an objective. Each of these small objectives will work towards a bigger goal, which, once all goals are complete, will result in a device that can be felt as an extension of the user rather than a tool that they simply wield. The basic goals basically ensure that the hand and glove can communicate and that the hand is functional as a basic human hand that can open and close with the ability of holding everyday objects.

The advanced goals are what we were really aiming to achieve when we were visualizing the project. It would allow the hand to function almost exactly as a human hand would and the latency being minimized would allow for realistic use for the end user. For the stretch goal there would be an arm that the hand is attached to. This is the natural progression of the hand and would allow for more mobility and use. If the arm is incorporated the hand could move freely and if the movement goals are met it would replicate a human arm and hand.

Table 2.1 Project Goals Overview

Goal type	Description
Basic	• Create a device with a series of sensors that will track the user's movements and translate them to GRIP, allowing it to mimic the user with high accuracy. • The hand will be able to have an open or closed grip as well as other standard positions
Advanced	• The system will be able to sustain a low-latency connection with moderate traffic between the user's hand and GRIP, allowing the hand to move with the user with negligible delay. • The hand portion of GRIP will have articulated fingers that are bound to the user's fingers, allowing for more intricate interactions with the environment. Basic movements such as pointing and pinching will be doable.
Stretch	• Incorporate an arm attached to the hand that will be able to mimic the user's arm movement. • The arm's range of motion will be able to be tested by a computer program without the glove. • A homing algorithm will be made to calibrate soft and hard stops for the arm, allowing use from different-sized users after the calibration sequence.

2.4 Project Features and Functions

The project will have several critical features that will function in tandem with each other. The glove itself will have a series of sensors along the upper arm that will take note of the position of the arm relative to the user being upright. The glove will send these positions wirelessly to a receiver on the bionic arm's PCB, which will then instruct it to move to the positions of the user. The glove will be powered with a battery attachment that can be changed out, along with a power switch. A calibration mode will be available to acquire the user's full range of motion before connecting to the bionic arm. Software-wise, GRIP will have algorithms in place that will keep it from moving in a way that is considered out of range, but if that fails, the arm will have hard stops for the servos in order to guarantee operation in the functional range. The GRIP arm will also have a homing sequence on power start to log its ranges.

2.5 Project Specifications

The project is split into two main components: the arm and the glove worn by the user. As such, each will have its own specifications that are tailored to its workload and design. Each part should be able to function independently of the other for testing purposes, with the ultimate goal of the glove controlling the arm being achieved once the connection between the two is made. The tables below illustrate the specs of both components separately.

Table 2.2 Arm Specifications

Parameter	Specification
Pinch force	≥ 10 N continuous
Finger closing time	≤ 300 ms
Control latency	≤ 15 ms at 200 Hz
Position repeatability	+/- 3 mm at fingertips over 10 cycles
Runtime on battery	≥ 1.5 hr

Forearm dimensions	0.79m x 0.047m x 0.172m
Bicep dimensions	9.75” x 3.25” x 3.25”
Palm dimensions	0.061m x 0.07m x 0.089m
Index finger dimensions	0.114m x 0.018m
Middle finger dimensions	0.124m x 0.018m
Ring finger dimensions	0.115m x 0.018m
Pinkie dimensions	0.094m x 0.018m
Thumb dimensions	0.102m 0.018m
Arm weight	≤ 30 lbs

Table 2.3 Glove Specifications

Parameter	**Specification**
Mass	≤ 150 g
Runtime	≥ 6 hr

Latency	≤ 10 ms
Link reliability	Packet loss ≤ 0.5% at 5 m LOS
Fit	One size
3-axis accelerometer	ICM-20948 9DoF IMU
Wireless transmitter	nRF24l01 wireless module
Finger	ZD10-100
MCU	Arduino NANO

Table 2.4 3D Printing Specifications

Parameter (Bambu Lab P1P)	**Specification**
Technology and Motion	Process FDM/ material extrusion. Top speed 500 mm/s Build volume: 256 x 256 x 256
Materials	Used PLA
Cameras and Monitoring	Chamber monitoring camera support: 1280x720 @0.5 fps
Connectivity and Control	Wi Fi, Bluetooth, Bambu-Bus, Micro SD(what we used) and the bambu handy app
Power, Size, Mass	Input 100-240 VAC, 50/60 Hz. Max power = 1000 Watts@ 220V

	Dimensions 386 x 389 x 458 mm Weight = 9.65 kG
Nozzle/Flow	All-metal hotend, max 300 °C. 0.4 mm nozzle included; optional 0.2/0.6/0.8 mm. (Max flow ~32 mm^3/s
Heated Bed and Plates	Max bed temperature = 100 degrees Celcius Plates are cool engineering high temperature type compatible

We fabricate glove and hand fixtures on bambu lab P1P, 3D printed with the above specifications. The P1P with PLA filament for all parts.
The PLA used is 0.25 kg spool weight, 1.75mm diameter, net weight of the entire spool is 1 KG.

2.6 Hardware Diagram

This diagram covers both the glove and the actual arm connections. We intend to implement servos to mimic joint and shoulder movement. The intention is the use of flex sensors and PCB designs that are acceptable sizes for the average human. For the glove diagram below the main components are the sensors, wireless communication, and controller. The rest of the components are to support those components or to provide power. For the hand or arm portion it is similar. There is more of a demand for power so there would need to be a larger power source and there is an increased demand for current that the motors that will move the fingers need. This will need to be taken into account when designing the PCB for the hand.

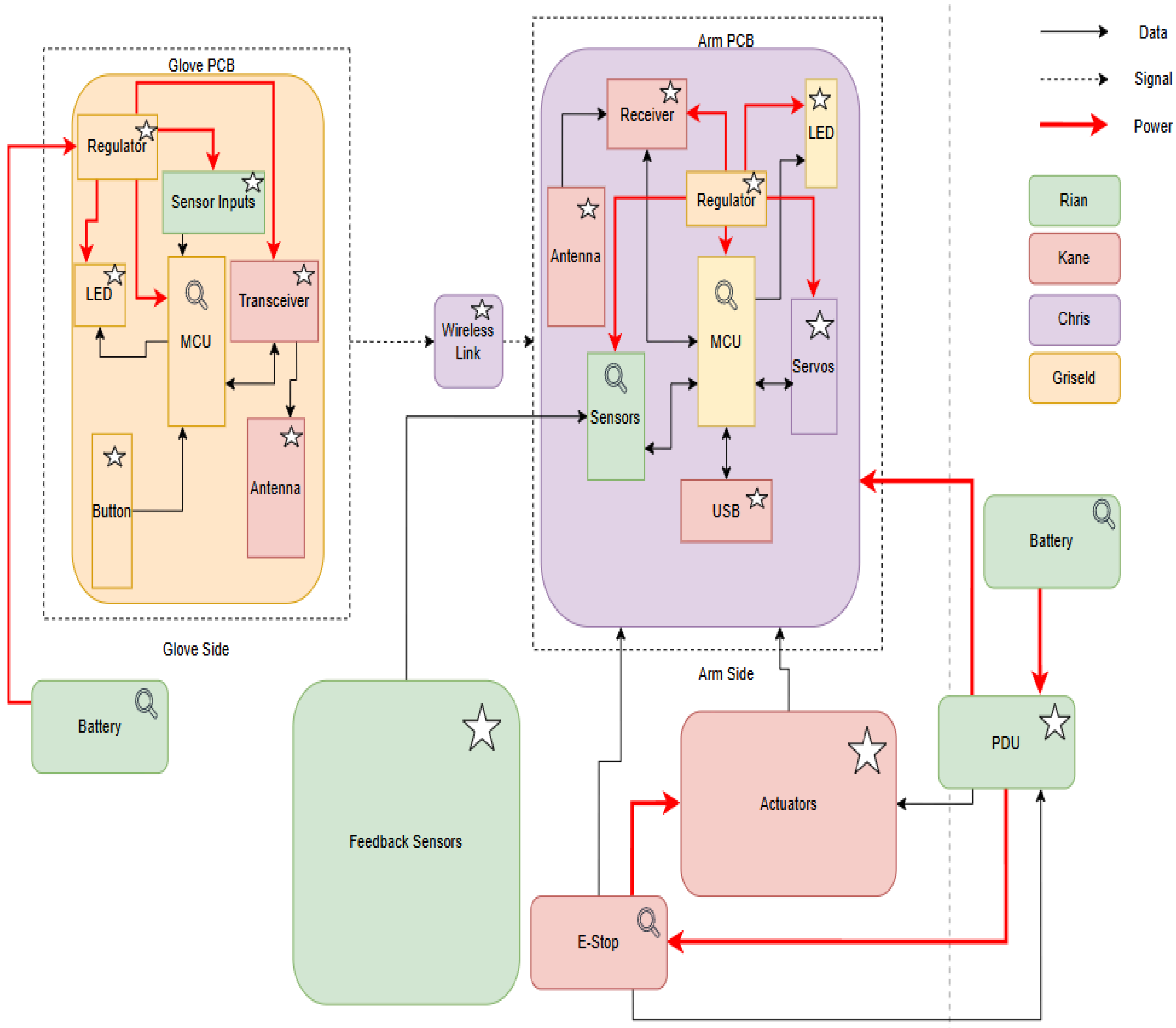


Figure 2.1 Hardware Block Diagram

2.7 Software Design

The GRIP software converts finger-bend measurements from the glove into coordinated servo motion on the robotic hand. When powered on, the glove's Arduino Nano initializes and begins acquiring analog readings from the flex sensors. A short calibration routine establishes each user's neutral and fully bent reference positions. The system then filters the raw signals and maps them to estimated joint-angle values. These values are packaged into a wireless data message, protected with a CRC checksum, and transmitted via the NRF24L01.

On the receiver side, the second Arduino Nano continuously listens for packets, verifies their integrity, and extracts the joint-angle commands. Before driving the servos, the firmware applies safety limits to prevent motion beyond mechanical bounds. The final angles are output as PWM signals to five finger servos, reproducing the user's hand pose in real time. A communication watchdog enforces a fail-safe state, returning the hand to neutral if transmission is lost. This modular software pipeline—acquisition, conditioning, mapping, transmission, validation, and actuation—ensures smooth and responsive glove-to-hand control for SD1. Shoulder control will be added in SD2.

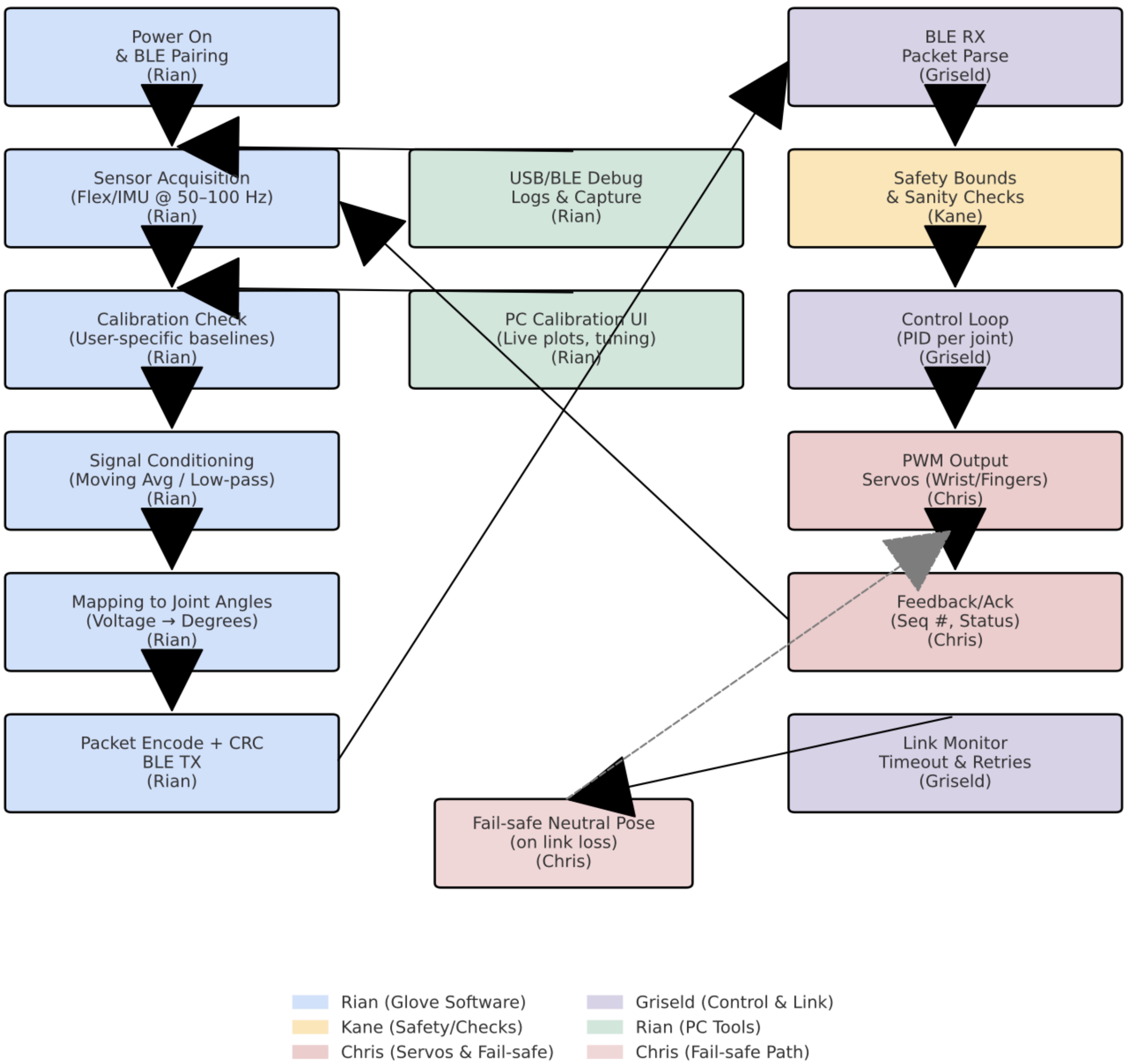
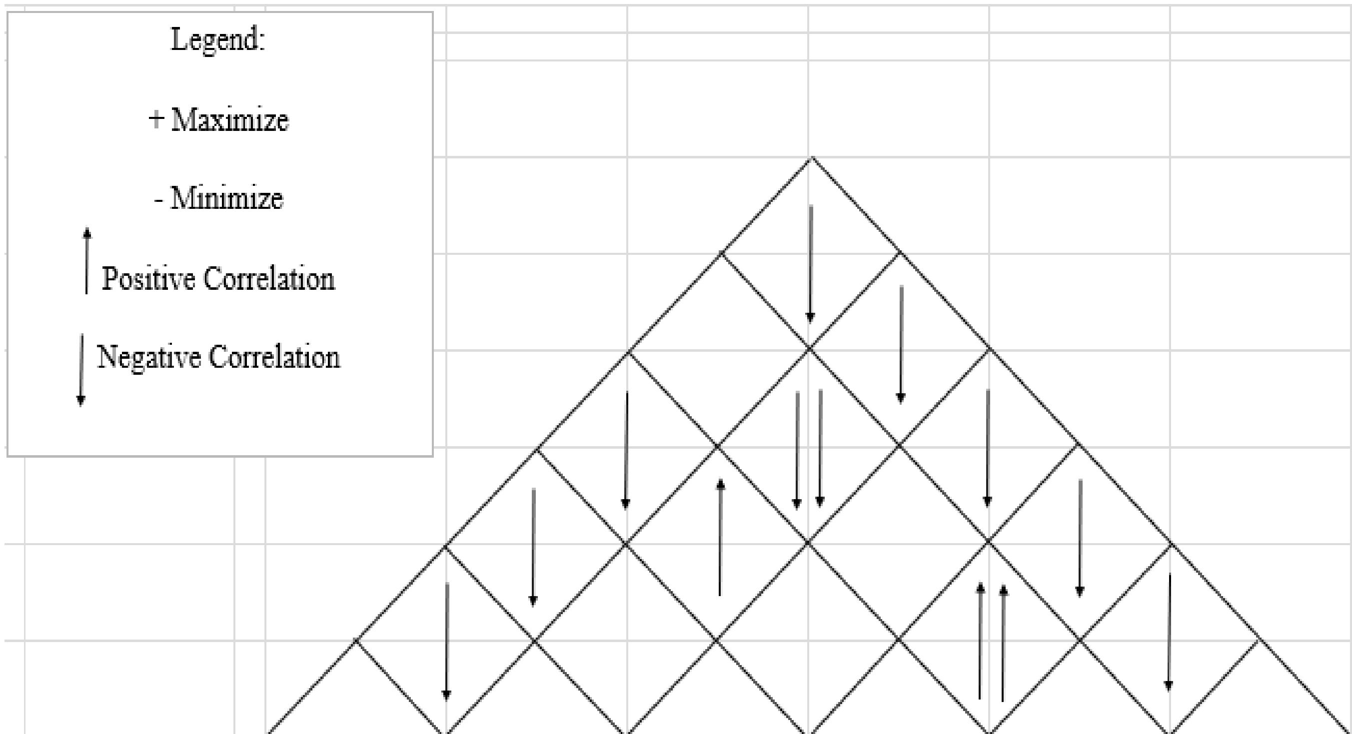


Figure 2.2 Software Diagram

2.8 House of Quality



Engineering Requirements / Marketing Requirements	Direction of Desirability	Battery Life	Cost	Weight	Latency	Accuracy	Temperature
Direction of Desirability	+	+	-	-	-	+	-
Portability	+	↑	↓	↓↓			
Cost	-	↓		↓	↓↓	↓	↓
Ease of Use	+	↑	↓	↑↑	↑↑	↑	↑
Data Accuracy	+		↓		↑	↑↑	
Latency	-					↑	
Battery Life	+		↓				
Target		> 30 Min	< 1500	< 25 lbs	< 2 seconds	± 5% from reference	No Overheating

Figure 2.3 House of Quality

Chapter 3 Research and Investigation

3.1 Technology Comparison

3.1.1 MCU or FPGA

Microcontroller units (MCU) can be thought of as a small computer on a chip. They consist of a processor, memory, general purpose input/output pins (GPIO), and I2C/SPI/UART. With all of those the MCU can control hardware directly. They can run control with low latency and power consumption.

MCUs are effective when real-time input and output is needed along with efficiency. They can for example read sensors, control wireless communication, and drive actuators which is useful for this project. MCUs are often cost effective compared to more powerful options. The benefit versus the cost makes them very effective for most projects that require a controller.

In terms of use MCUs are very user friendly. Since they are a popular choice for any project there is plenty of documentation to refer to in any use case and are available from a wide range of companies. When it comes to programming them they usually use well established languages that are either well documented or widely known so when it comes to setting them up with the appropriate code or debugging it is simple. Since we most likely will have some experience with some form of MCU between all the members of the group they will be easy to work with. All of this means that a MCU would be appropriate for use in this project since it covers all of the bases that we have.

Another option would be Field Programmable Gate Arrays (FPGA). FPGAs are chips that allow digital control of hardware instead of the program in the code like the MCUs. It uses logic blocks that can be programmed using Verilog or VHDL to build the hardware circuits. FPGAs would be more capable than MCUs and would be excellent for timing and low latency.

The tradeoff is that they are more complex than MCUs. Designing the hardware in VHDL or Verilog has a steeper learning curve than the knowledge required for the MCUs. Also the verification required for the simulation can be very time consuming and troubleshooting small errors can be difficult. In short FPGAs are not as user friendly as a MCU would be. They also come at a higher cost than readily available MCUs. For this project a FPGA would provide greater power and control but it would be overkill for this application.

For this project we decided that a MCU was appropriate for the requirements of the project. Microcontrollers would give a good mix of real-time control, Input/output options, and not take too much power for the GRIP application. Since our project has time critical parameters the MCUs predictability is a positive. They also keep the latency low which was one of our major goals. Other alternatives would either under or over shoot our need for this project. In terms of cost MCUs also excel in that category as we are going to need two. One will be for the glove and one will be for the hand. There is also the ability to integrate our stretch goal since MCUs have the extra capability that the arm would need along with the hand.

Table 3.1.1 MCU vs. FPGA

	MCU	FPGA
Power consumption	X	
Ease of use	X	
Processing power		X
Cost	X	
Documentation	X	

3.1.1.1 Arduino Nano

The Nano is small so it would be easy to mount for our project. The programming is simple in the Arduino IDE with plenty of available libraries. The peripherals we would need are all on the board including I2C and GPIO which covers the tasks we would need. It is simple and boots instantly with tight control which means latency would be reduced. Since there is a large community support network for Arduino troubleshooting is made easy. If spares are needed and since we will need two they are easy to source and inexpensive.

There are some cons as well to accompany the pros. There is 32 KB of flash and 2KB of RAM which may limit some features or libraries that can be used. For the UART hardware it shares with the USB which can cause limitations. The CPU isn't ideal for very heavy calculations or any high rate telemetry. Since the IO is 5 volts while many sensors are 3.3 volts only we may need some extra components in order to make the Nano viable. For the 3.3 volt pin there is not enough current to power many peripherals so that might need to be boosted. There is also no built in wireless so we likely would have to add one in ourselves.

3.1.1.2 Arduino Mega 2560

The mega gives a larger number of I/O pins with multiple UARTs and more timers. With all these extra features it is simple to attach extra sensors, displays, and devices without overwhelming the controller or running out of pins. Since it is in the Arduino family the large amount of libraries and support makes it simple to use and troubleshoot. The extra memory offers more room for larger amounts of code.

Since this MCU has all of those extra features it means that the footprint is much larger. It will also be heavier meaning it is harder to place in compact spaces or mount onto a glove for example. It still uses the same 8-bits 5 volts as the Nano which limits its performance and would require a solution for 3.3 volt components. There is also a higher power draw and more power is required.

3.1.1.3 Arduino Nano 33 IoT

This board keeps the small footprint of the Nano which makes it ideal for our project. The upgrade comes in the form of a 32-bit core with more memory and built in Wi-Fi which makes wireless integration possible. Again since it is the Arduino family the support and ease of use is excellent. This also has 3.3 volts instead of 5 volts which means it will match many sensors without any additional work.

The downside of the 3.3 volts is that any 5 volt peripherals will now need translation before they can be used. Any heavy calculations or high rate control will still feel strained with this controller.

3.1.1.4 ESP32 Dev Kit

The ESP32 has a lot packaged onto one chip. It features dual cores, multiple UARTs, timers, and Wi-Fi enabled. THis makes it great for combining wireless and real-time control on a single board. The community support is very good and there are plenty of libraries that can be used. In terms of the cost it is very reasonable and there are multiple choices of boards depending on your requirements.

Some of the cons are there is an increase in complexity with the Wi-Fi stacks and RTOS scheduling. This controller is also 3.3 volt only so if 5 volt devices are going to be added there needs to be special consideration taken. RF layout, and sleep/wake behavior need special attention. Some libraries also require special attention when they are used more than they would in the Arduinos.

3.1.1.5 STM32 Nucleo

This board brings great support for peripherals with good timer sets and solid UART that can support lots of sensors or other connected components. The Nucleo can run flexible control loops with precise timing making for low jitter clean pipelines. The community is large which helps for programming and trouble shooting. There are plenty of reference designs for motor control, power conversion, and sensing so you would not have to start from scratch. It seems to be focused on the support of peripheral components and make it easy to connect them.

There is a steeper learning curve than the Arduino line of products. Getting the clock and interrupts right takes more time than it would with other MCUs. The Nucleo is also a larger size than the Arduino Nano for example which makes placing it in tighter applications difficult. Everything is 3.3 volts only which would require shifting for 5 volt peripherals. The complexity and size is the tradeoff for the extra power and features.

3.1.1.6 Teensy 4.1

This chip has large performance capability with a M7 core and plenty of RAM and flash memory while still supporting Arduino tools for iteration. It excels at control, signal processing, and telemetry. There are many ports and high speed I2C which means that this on board can consolidate what might need multiple controllers.

Since it has such high capability this board can be overkill for some systems. The board is also larger than the Nano boards which makes it less feasible to place in applications where footprint matters. It is 3.3 volt only so there will need to be care taken with 5 volt components. All of the extra features also might add integration that is not needed for some projects. Cost is higher than entry level MCUs that would work fine for most projects. Power consumption can climb if not properly managed. It allows for future upgrades to a project but it may also end up having complexity that is not used.

3.1.1.7 Raspberry Pi Pico

The Pico is low cost and widely available and provides good function despite the low cost. It has dual cores and plenty of RAM with programmable input/output blocks. This allows for custom wave forms and precise timing. The support and programming is good and has a low barrier of entry. There are plenty of resources for trouble shooting or libraries that can be used in order to not start from scratch.

There is no built-in wireless so if communication is required there must be a separated component added. Logic is 3.3 volts only so there would need to be special care taken for 5 volt components. There is a new programming model that can be tricky if you are unfamiliar with it beforehand. The board footprint is larger than a Nano if size is of concern. If compute heavy tasks are required a more powerful platform would be required.

Table 3.1.2 MCU Comparison

	Arduino Nano	Arduino Mega	Nano 33	ESP32	STM32	Teensy	Pi Pico
Wireless			X	X			
Ease of use	X	X	X	X		X	
Power	X		X				X
Size	X		X				
USB	X	X	X	X	X	X	X
Cost	X	X		X			X
3.3V friendly			X	X	X	X	X
5V tolerant	X	X					

3.1.2 Smart Servos or Micro Linear Actuators

Smart servos are all in one with a motor, gearbox, sensors, and motor drive in a single housing. It also includes a digital control bus for control of those components. Setpoints will be sent to the motor and telemetry will be received so wiring would stay clean since it is an all in one package. There is a horn on the servo that a mimic tendon can be attached to in order to allow the motor to control the design it is attached to. The servo takes simple commands and can hold position and on some units can report status. The wiring is minimal, set up is fast, and from the small footprint they can be placed in applications that require tight spacing. The torque and speed specs are easy to modify and there is a range of off the shelf parts such as horns and brackets if modification is needed to make the motor fit a specific application.

Some of the tradeoffs are backlash and thermal limits. There is also the question of how many servos will need to be used to achieve the actuation that is desired. Holding high grip force for longer periods of time can also cause small servos to heat up. If there are multiple servos linked together there is a power burst that will need to be accounted for.

Linear actuators convert motor rotation into straight line motion through a screw or rack and pinion. This works by pulling the simulated tendon in a straight path. These can be bought or built if customization is required. The advantage to this is having a preloaded screw controlling the movement gives a more solid feel over a rotary gear. This also makes calibration simple because of the smooth path of travel and would keep the simulated tendons that control the fingers tidy.

The costs are size, complexity, and tuning. Even the smallest footprint of these motors would be larger than the servo motors and if customized there would be other parts to account for. There would also be more extra work required because each actuator would need a driver and feedback with more control work than an all in one package like the servos. The price would also be higher and power draw can be significant along with wear on the components if they are not properly assembled, aligned, and lubricated.

With all this in mind the smart servos is what we will use. They come in a complete package with simple controls and can still be modified if required. The cost is reasonable and it is an established component in many projects which means any trouble shooting and documentation is available.

Table 3.1.3 Smart Servos vs. Micro Linear Actuators

	Smart Servos	Micro Linear Actuators
Wiring	X	
Packaging	X	
Stiffness		X
Control	X	
Power bursts	X	
Cost	X	

3.1.2.1 MG996R

This is a common standard size hobby servo with a metal gear train and a simple three wire interface including power, ground, and signal. It accepts a pulse width command and has an internal potentiometer so there is no need for a motor driver or encoder. It is inexpensive and widely available. There is a large support network in case of trouble shooting requirements or documentation. It delivers useful stall torque for the tendon spools that will be used. There is also a large selection of horns, brackets, and gears for customization and repair purposes.

The tradeoffs are what can be expected from a hobby servo motor. There can be some gear backlash, motor whine, and variance from unit to unit in deadband and holding behavior and deadband. There is no host side telemetry so you would have to design relying on limit switches or mechanical stops. Currents can be high for short bursts so the voltage rail must be stiff and the wiring short. Grounds must be accurate especially when using multiple servos such as in our case. Long term clamping of the hand would also increase heat and wear on the motors.

3.1.2.2 Micro PWM Servos

Micro servos shrink the servo layout into a very small package that would be effective in situations that require a small footprint.They are good for lightweight degrees of freedom and short simulated tendons. Integration is simple, only requiring a single signal wire per motor. The power distribution will be 5 volts. Their low mass reduces inertia giving better motion to the fingers. The cost is very low which makes having spares easy in case of wear from long term clamping or heavy loads.

The compromises are torque and durability. Many micro servers use plastic gears and smaller less powerful motors so high grip forces or impacts will wear out the components faster. They provide no data back to the controller so there is calibration drift and gear wear. Power distribution can also be touchy and as in the case with this project a handful of motors will brown out a weak power source.

3.1.2.3 Digital PWM Servos

Digital PWM servos keep the same three wire interface but run a faster internal control loop and higher up update rate to the motor which translates to tighter holding torque and crisper transient response. Many of these in this class add metal geartrains, stronger motors, and configurable endpoints. If you want the simple wiring of hobby servos but want firmer grip as in our project and more repeatable positioning this class is a good upgrade with minimal code changes.

There are still PWM only and feedback blind to the host so the wiring scales linearly and the current, temperature , or position telemetry for safety or logging . Peak currents are higher than micro or standard servos so a rail from 6-7.4 volts is required. Bulk capacitance is also required to absorb any transients. Thermal limits also apply with long holds near stall. They will heat the case which limits the amount of time the grip can be active. There is also still backlash but less than some of the cheaper servos

3.1.2.4 TTL Bus Smart Servos

These integrate a motor, gearbox, and position sensor with a digital bus that is UART like so power and two signal lines will be routed through a string of actuators and address them by ID. The largest benefits are built in feedback and commanded motion profiles which simplify synchronization and safety. Because the command signal is digital rather than pulse widths, timing is robust and jitter free and the harness inside the compact hand stays cleaner than running separate lines to everything.

Costs include each vendor using its own unique protocol and lower torque density than larger units at the same price point. There would need to be a separate transceiver added to the controller board and attention will need to be paid to the grounding and stub lengths on the chain to avoid communication issues. While feedback helps absolute encoder resolution and gearbox quality still set the repeatability floor and backlash will still be present

3.1.2.5 RS-485/Can

At the higher end these smart servos move to 12-24 volt rails and absolute encoders with robust packets, error checking, and comprehensive telemetry. They expose configurable PID gains, software limits, and profile generators making syncing multi joint motion and safety interlocks far easier. Chaining together a few actuators over a proper bus means a single pair can serve an entire finger cluster and running at the voltage it does lowers current for the same mechanical power which improves voltage stability.

The trade offs are cost, size, and ecosystem commitment. These units cost more and can be larger than other servos and the budget will be higher for proper connectors and hubs. Bus setup is an extra bring up step and though backlash is typically lower thanks to better gears and bearings but it is not perfect.

Table 3.1.4 Servo motor comparisons

	MG996R	Micro PWM	Digital PWM	TTL Bus Smart	RS-485
Control signal	PWM	PWM	PWM	UART	RS-485
Voltage family	5-6 V	5-6 V	6-7.4 V	5-7.4 V	12-24 V
Feedback				X	X
Torque density			X		X
Backlash	Metal gears	Plastic gears	Mixed materials		Best in class
Holding accuracy			X	X	X
Telemetry				X	X
Setup	X	X	X		
Scalability				X	X
Cost	Low	Very low	Mid	Mid	High
Power stress					X

3.1.3 Transceiver

A transceiver is a compact device that combines signal generation, modulation, and power amplification stages needed to transmit data with the low noise amplification and demodulation stages required to receive it. This presents a simple digital interface to the host controller. In wireless variants the RF front end is coupled to a baseband modem that implements the chosen layout along with framing and error checking. Contemporary parts integrate hardware, addressing, CRC, and automatic acknowledgement so the host exchanges payloads over UART while the transceiver manages the other details. Wired transceivers perform an important role at the physical level translating logic level signals into balanced line drivers with fail safe biasing and termination options.

Operational behavior is governed by band selection, link budget, and protocol timing. Devices operating in the 2.4 GHz ISM band offer small antennas and higher data rates in exchange for coexistence with Wi-Fi and Bluetooth. Sub-GHz parts prioritize penetration and range at lower throughputs. The link budget is set by transmit power, receiver sensitivity, and antenna efficiency. This determines practical range and resilience while the protocol timing model sets latency and jitter. Together these characteristics define whether a transceiver is better suited to low-latency control traffic, bulk telemetry, or long-reach monitoring.

From a system perspective transceivers are compact building blocks that abstract complex analog and RF concerns into repeatable digital behaviors. They are typically powered from a regulated 3.3 V rail and draw short current bursts during transmission or receive and require modest board level accommodation such as local decoupling, a clear keep-out around the antenna region for modules, and clean reference routing for their digital interface. Once configured channel, data rate, addressing, and optional security the device presents a predictable packet service to the firmware: short payloads are queued, transferred across the link with hardware assistance, and delivered with integrity checks and delivery status allowing the application to treat the transceiver as a reliable conduit within the overall control architecture

3.1.3.1 nRF24l01

The nRF24L01 is a tiny 2.4 GHz transceiver that talks over SPI and pushes short packets with auto-ack/auto-retransmit, hardware FIFOs, and selectable on-air data rates. It’s purpose-built for low-latency point-to-point or star links at a few meters to tens of meters of sight perfect for a glove streaming pose data at 100 to 200 Hz. The register model is simple addressing pipes that make multiple node setups manageable, and modules are cheap and compact. With a decent packet format, you can get very smooth deterministic updates.

Some of the tradeoffs are it is 2.4 GHz, so it shares spectrum with Wi-Fi/BLE and can see interference. To solve this, pick a clean channel and consider the 250 kbps mode for better link budget. Most breakout boards are 3.3 V only on VCC and I/O so level-shift if your MCU is 5 V and add good decoupling right at the module. Long range variants can help but they draw more current and need antenna clearance and grounding attention. Security isn’t built in beyond addressing and CRC.

3.1.3.2 ESP-NOW

ESP-NOW is a peer-to-peer protocol built into ESP32 radios that provides very low overhead packet exchange without associating them with a Wi-Fi access point. It supports broadcast and unicasts and works with PCB or external antennas and runs on hardware that already offers multiple UART interfaces along with ample RAM for logs and diagnostics. For glove to hand control, it can deliver snappy updates while keeping firmware simple on both ends.

The trade-offs are power, and stack complexity compared to a tiny transceiver. An ESP32 is a full SoC radio so idle and transmit currents are higher than an nRF24L01 and you must manage tasks and radio activity to avoid control-loop jitter. All I/O is 3.3 V. RF layout is more sensitive, and the protocol is vendor specific rather than generic. It is attractive when you also want convenient logging, over-the-air features, or a path to Wi-Fi networking for the narrow job of fast control packets alone, a dedicated packet radio remains simpler

3.1.3.3 Bluetooth Low Energy

BLE provides standardized phone friendly connectivity with low peripheral power, mature stacks, and widely available modules. It is attractive for user interfaces and calibration via a mobile app and field debugging. Modern BLE 5 options such as 2M and coded PHYs let you bias toward throughput or range and pairing establish a baseline for link security without extra application work.

For continuous teleoperation BLE's connection intervals, supervision timeouts, and GATT transaction model introduce more latency and jitter than a purpose-built packet link. While tuning can improve performance, holding a deterministic 100–200 Hz command stream is harder than with simple 2.4 GHz packet radios. Many modules are 3.3 V only and the API surface is broader, so firmware effort rises if BLE is used as the primary real-time control link.

A pragmatic approach is to treat BLE as an auxiliary channel. Use it for configuration, calibration, and diagnostics while the primary control runs over a lower-latency radio. If you must rely on BLE for control, choose a short connection interval and minimize GATT overhead with a custom characteristic carrying a compact binary payload and measure actual end-to-end timing under realistic interference before finalizing the design.

3.1.3.4 XBee

XBee modules encapsulate IEEE 802.15.4 radios in a convenient form factor with options for transparent serial bridging and full Zigbee networking stacks. The 802.15.4 physical layer provides low-power operation, modest data rates, and robust short-range performance, while higher-layer profiles on certain models enable mesh capabilities such as routing and network addressing. The module ecosystem includes through-hole and surface-mount packages, integrated or external antennas, and standardized pinouts.

Most XBee families operate from a 3.3 V supply and expose UART and sometimes SPI interfaces, with configurable parameters via AT commands or a binary API. Radio variants cover both 2.4 GHz and sub-GHz bands, allowing selection based on range, penetration, and regulatory region. Power consumption and latency characteristics depend on the specific stack enabled, with transparent mode minimizing overhead and mesh features adding store-and-forward behavior.

This platform is frequently used where simple cable-replacement behavior is desired at the outset, with the option to scale to small networks using addressing and routing when needed. The combination of mature documentation, certification options, and accessory ecosystem makes it a stable choice for embedded links that prioritize reliability over raw throughput.

3.1.3.5 LoRa

LoRa devices implement a chirp spread-spectrum modulation in sub-GHz ISM bands such as 868/915 MHz, achieving very high receiver sensitivity and long link distances with small payloads. The physical layer supports configurable bandwidths, spreading factors, and coding rates, enabling a tunable trade-off between data rate and robustness. Common transceiver ICs present an SPI interface with registers for modem configuration, packet framing, and RSSI/SNR reporting.

Electrical characteristics are typical of sub-GHz radios: a regulated 3.3 V rail, discrete front-end filtering and matching, and antenna systems sized by the longer wavelength. Current draw rises with transmitted power and decreases with higher spreading factors, and duty cycle or dwell time regulations may apply depending on region. Modules vary from compact integrated solutions with onboard matching to larger designs accommodating SAW filters and external antennas.

The technology is well matched to low-throughput telemetry and monitoring where coverage and building penetration are prioritized over latency and bandwidth. It underpins many low-power wide-area networks and point-to-point links that require long-range operation, periodic reporting, and resilience to multipath and attenuation in cluttered environments.

Table 3.1.5 Transceiver comparisons

	nRF24L01	ESP-NOW	BLE	XBee	LoRa
Operating band	2.4 Ghz	2.4 GHz	2.4 GHz	2.4 GHz or sub	Sub GHz
Host	SPI	ESP32	UART/SPI	UART/SPI	SPI

interface					
Supply level	3.3 V	3.3 V	3.3 V	3.3 V	3.3 V
On air rate	X	X			
Latency suitability	X	X			
Indoor range	Short	Short	Short	Medium	Long
Packet assist in hardware	X	X			
Complexity	X				
Antenna choices	X	X	X		
Cost	X	X	X		

3.1.5 MOSFET

A metal oxide semiconductor field effect transistor (MOSFET) is a voltage controlled semiconductor switch in which an electric field at the insulated gate modulates the conductivity of a channel between source and drain terminals. The device is fabricated on a doped silicon substrate with a thin gate oxide separating the gate electrode from the channel region. Applying a gate-to-source voltage above the device's threshold attracts carriers into the channel and forms a low-resistance path. Removing that voltage returns the device to a high impedance state. Two polarities are available: N channel devices conduct with electrons as majority carriers and turn on for positive gate bias, while P channel devices conduct with holes and turn on for negative gate bias relative to the source. Most power and signal switches used in low-voltage electronics are enhancement-mode, where the default state is off.

Key electrical parameters describe how a MOSFET behaves in circuits. The drain–source voltage rating sets the maximum blocking voltage in the off state, and the continuous and pulsed drain current ratings reflect conduction limits with specified thermal conditions. Gate-to-source threshold voltage defines the onset of conduction but does not guarantee low loss; on-resistance is specified at higher gate drive levels such as 2.5 V, 4.5 V, or 10 V and determines conduction losses when the device is on. Dynamic behavior is governed by gate charge and the device capacitances between terminals, which influence how quickly the device can switch for a given driver. Every MOSFET includes an intrinsic body diode between body and drain regions; its recovery behavior and forward voltage are important in switching applications that reverse current.

3.1.5.1 BSS138

BSS138 is a small-signal N channel enhancement MOSFET commonly supplied in SOT-23 packaging. It is intended for low-voltage applications and features a low gate threshold, allowing the device to turn on with a logic-level gate drive. Typical maximum ratings include a drain-source voltage in the 50–60 V class and a continuous drain current in the few hundred milliampere range, reflecting its role as a signal-level or light-load switch.

Electrical characteristics emphasize modest on-resistance at low gate voltages, with datasheets specifying RDS at gate drives such as 2.5 V and 4.5–5 V. The device presents low gate charge, enabling fast transitions with minimal drive energy and reduced loading of the control pin. Body diode characteristics follow the usual N-MOSFET behavior, and the device's small die size supports compact layouts with short trace lengths.

From a packaging and logistics perspective, BSS138 is widely second-sourced, inexpensive, and available in tape-and-reel for automated assembly. Variants exist under multiple vendor part numbers with closely matching electrical limits. The combination of low gate threshold, low gate charge, and small outline has made it a common choice in logic interfacing and light switching roles where board area and cost are constrained. Packages and construction span a wide range to suit current and thermal requirements. Small signal parts commonly appear in SOT-23 and similar outlines for currents in the hundreds of milliamperes, while power devices use packages

3.1.5.2 2N7002

2N7002 refers to a family of small-signal N-channel MOSFETs also offered in SOT-23 and similar packages. Like BSS138, these parts target low-to-moderate current switching with logic-level drive, and they typically carry drain-source voltage ratings around 60 V with continuous drain currents in the few hundred milliampere range. The device family is broadly manufactured, so sourcing options are extensive.

Electrical behavior differs primarily in gate threshold distribution and on-resistance curves at low gate voltages. Many 2N7002 variants specify RDS at VGS 4.5–5 V, with some providing data at 2.5 V for low-voltage logic. Gate charges are small, facilitating rapid switching in signal and low power converters, and parasitic capacitances are low enough to make these devices predictable in high-impedance control environments.

From a mechanical and assembly standpoint, footprints and pinouts align with common SOT-23 conventions, easing substitution among vendors. The broad ecosystem of equivalents simplifies second-source planning. Designers often view the 2N7002 class and BSS138 class as interchangeable small-signal switches, with selection guided by specific RDS and threshold curves in the intended gate-drive range. such as SO-8, DFN/PowerPAK, TO-220, and D2PAK with exposed pads or tabs to conduct heat to copper planes or heatsinks. Datasheets characterize thermal resistance from junction to ambient or case and provide safe operating area curves that bound permissible combinations of voltage, current, and time. Additional features may be specified, including avalanche energy capability, logic level gate drive compatibility, and in some product families integrated functions such as gate protection diodes or load-switch control, allowing selection of devices that match the intended voltage range, switching speed, and thermal design.

3.1.5.3 AO3400/A

AO3400-class parts represent higher-current, logic level N channel MOSFETs in compact SOT-23 packaging. Compared with small-signal switches, these devices offer substantially lower RDS at gate drives such as 2.5 to 4.5 V, enabling amperes of continuous current when thermal conditions are managed. Typical drain-source voltage ratings are in the 25–30 V range, appropriate for low-voltage power rails.

Key electrical parameters include very low on-resistance, moderate gate charge, and robust pulsed current capability. Transfer characteristics are optimized for true logic-level operation, and datasheets commonly provide curves down to 2.5 V gate drive. Package thermal resistance becomes the limiting factor; copper area and vias under the source tab are used to spread heat in compact layouts.

Availability is broad with multiple vendors offering pin-compatible devices in the same performance class. These MOSFETs are frequently chosen when a small outline is required but current and conduction loss targets exceed what small-signal parts can support. The blend of low RDS at low VGS and a ubiquitous SOT-23 footprint makes the AO3400 family a common baseline for compact low voltage switches.

3.1.5.4 BSS84/A

BSS84 class devices are small-signal P-channel MOSFETs supplied in SOT-23 packages, intended for low voltage applications requiring a high side device with negative gate drive relative to the source. Typical maximum drain-source voltage ratings are around 50 V, and continuous drain currents fall in the few hundred milliampere range, placing these parts in the signal and light load category.

Electrical characteristics include on-resistance specified at gate-source voltages such as 2.5 V and 4.5 V, with gate thresholds tailored for logic-level control in low-voltage systems. Gate charge is low, supporting fast transitions and modest drive requirements, while body-diode behavior follows the P-channel topology. As with their N-channel counterparts, parasitic capacitances are small, which aids predictable switching when driven from simple logic sources.

From a sourcing perspective the BSS84 family is widely available and second-sourced, making it straightforward to maintain alternatives. The small package and standard pinout ease placement on dense boards, and the parts are often paired with complementary N-channel SOT-23 devices to implement compact high side or load disconnect functions where a P-channel polarity is preferred.

3.1.5.5 TPS229xx

TPS229xx refers to a family of integrated load switch devices that combine a low RDS MOSFET with gate control, slew rate management, and protection features in a small outline. Rather than exposing a bare gate, these ICs present an enable pin and, in many variants, flags for fault reporting, reverse current blocking, or quick output discharge. Voltage ratings and on-resistance are targeted at low voltage rails, with options spanning a wide current range.

Electrically, the devices are characterized by controlled turn-on behavior to limit inrush, well defined quiescent currents, and specified thermal performance in small packages such as WCSP, X2SON, or SOT variants. Many members of the family are optimized for 1.1 to 5.5 V operation and publish RDS at low gate voltages since the internal driver is designed for logic control. Some options include good power outputs and undervoltage lockout thresholds that simplify system sequencing.

From a system-integration standpoint, these parts are used when a clean, self-contained load switch is preferred over a discrete MOSFET and gate network. The consolidation of control and protection reduces external component count and can improve repeatability across units. Product availability covers multiple vendors and footprints, enabling selection by rail voltage, current rating, and required features without changing the overall approach.

Table 3.1.6 MOSFET comparisons

	BSS138	2N7002	AO3400	BSS84	TPS229xx
Polarity	N-ch	N-ch	N-ch	P-ch	N-ch
Package	SOT-23	SOT-23	SOT-23	SOT-23	WCSP/X2SON/SOT
VDS rating	50-60 V	60 V	20-30 V	50 V	1.1-5.5 V
Logic level	X		X	X	X
RDS class			X		X
Continuous ID			X		X
Gate charge	X	X		X	X
High side suitability				X	X
Added functions					X
Thermal performance					X
Sourcing	X	X	X	X	X

3.1.6 Batteries

A rechargeable battery is an electrochemical storage device that converts chemical energy to electrical energy through redox reactions at two electrodes separated by an electrolyte. Cell chemistry sets the nominal voltage, energy density, internal resistance, and safety profile; common chemistries in portable electronics include lithium-ion families (lithium-cobalt oxide, nickel-manganese-cobalt, lithium iron phosphate), nickel-metal hydride, and, less commonly now, sealed lead-acid for higher power at low cost. Cells are manufactured in cylindrical, prismatic, and pouch formats, each balancing mechanical robustness, thermal behavior, packaging efficiency, and manufacturability.

Key performance attributes include specific energy and specific power, continuous and burst discharge capability, cycle life, and calendar aging. Internal resistance and temperature strongly influence voltage sag and heat generation under load, while protective elements govern usable operating windows. Lithium-ion cells are typically charged by a constant current or constant voltage profile to a defined upper cutoff and discharged to a lower cutoff to maintain longevity. Nickel chemistries use different termination methods and tolerate overcharge and deep discharge differently. Datasheets specify limits for charge and discharge currents, temperature ranges, and storage conditions that determine practical service life.

Most packs incorporate electronics to manage safety and interface expectations. A battery management system or simpler protection circuit monitors cell voltage and current, enforces over-current, over/under-voltage, and over-temperature cutoffs, and may provide balancing in multi cell series stacks. Consumer USB power banks add a boost converter to present regulated 5 V along with state of charge indication and input charging control. Cylindrical formats such as 18650 are widely second sourced with consistent mechanical standards, whereas pouch and prismatic formats offer higher packaging efficiency at the expense of mechanical rigidity and may rely more on the enclosure for support and heat spreading.

3.1.6.1 18650 Li-ion

An 18650 lithium-ion cell is a standardized cylindrical format approximately 18 mm in diameter and 65 mm in length, widely used in portable electronics and electric vehicles. Typical chemistries include nickel-manganese-cobalt (NMC) and nickel-cobalt-aluminum (NCA), yielding nominal voltages around 3.6 to 3.7 V per cell with energy densities that are high for their size. Mechanical robustness is a strength of the metal can construction, which helps with handling, spot-welding of tabs, and thermal conduction to the enclosure.

Electrical characteristics are defined by capacity classes ranging from roughly 2,000 to 3,500 mAh and continuous discharge ratings that vary from low drain cells intended for consumer devices to high drain cells designed for power tools and traction packs. Internal resistance and thermal behavior depend on the specific model. Datasheets specify maximum continuous and pulse currents, recommended temperature windows, and charge profiles using constant current or constant voltage to a fixed upper cutoff. Cycle life and calendar aging are influenced by depth of discharge, charge voltage setpoint, and temperature.

Cells are available from multiple manufacturers with consistent dimensions and a broad ecosystem of holders, protection circuits, and pack-building accessories. Variants include “protected” versions that integrate a small protection PCB under the sleeve, and bare cells intended for assembly into packs with external battery management systems. Traceability and authentic sourcing are important due to market prevalence and performance variability among brands.

3.1.6.2 Lithium-Polymer Pack

Lithium-polymer pouch cells use laminated foil enclosures rather than rigid cans, allowing thin, lightweight shapes and custom footprints that can be stacked or arranged to fit irregular volumes. Nominal cell voltage is again around 3.6 to 3.7 V, with capacities and C-ratings tailored to applications from small wearables to high-power model aircraft packs. The flexible packaging enables high packing efficiency but places more responsibility on the surrounding mechanical design for support and protection.

Electrical behavior spans low to very high discharge capability depending on electrode design. Datasheets specify continuous and burst C-rates, internal resistance, and temperature ranges for charge and discharge. Charging uses the same constant current and constant voltage method as other lithium-ion chemistries, with attention to cell balancing when multiple cells are configured in series. Thermal characteristics benefit from large surface area but require uniform compression and avoidance of sharp edges or point loads to maintain reliability.

Commercial offerings range from single cells with flying leads and protection boards to multi-cell packs with balanced connectors and embedded protection. The absence of a rigid shell makes weight low and form factor options broad, while also making the cell more sensitive to swelling if overstressed or aged. Handling practices emphasize mechanical support, controlled clamping pressure, and enclosure features that mitigate puncture risk.

3.1.6.3 LiFePO4 Cells

LiFePO4 cells use an olivine structured cathode that provides a nominal voltage near 3.2 to 3.3 V per cell, lower than cobalt-rich chemistries but with a flat discharge curve and strong thermal stability. Energy density is typically lower on a volumetric basis compared with NMC/NCA, yet specific power and safety margins under abuse conditions are recognized advantages. Cycle life can be long, with datasheets often citing thousands of cycles under moderate depth of discharge.

Electrical specifications include continuous and pulse discharge ratings suitable for many portable power roles, with internal resistance generally higher than high energy NMC cells but stable across a wide temperature range. Charging follows a constant current/constant voltage profile to a lower terminal voltage than other lithium-ion chemistries, and the chemistry's tolerance to high cycle counts favors applications where longevity outweighs absolute capacity per volume.

Cylindrical LiFePO4 formats mirror the accessory ecosystem of other cells, making holders, tabs, and pack assembly familiar. The chemistry's lower nominal voltage affects series/parallel configurations for given rail targets, and the flat discharge plateau simplifies state of charge estimation in some battery management approaches. Availability spans primary manufacturers and many secondary sources with varying discharge capabilities and capacity classes.

3.1.6.4 USB Power Bank

A consumer USB power bank combines one or more lithium-ion cells with a protection and control PCB and a DC-DC boost stage to present regulated output ports. The most common interface supplies 5 V over USB-A or USB-C, with newer models implementing power-delivery negotiation for additional fixed voltage steps. Enclosures typically include charge-state indicators, on/off control, and protection against over-current, over-voltage, under-voltage, and over-temperature conditions.

Internally, packs often use one to three cells in parallel at a single series count, relying on a boost converter to produce the regulated output. The battery management circuitry supervises charging via a dedicated charge controller, balances parallel cells implicitly through common connection, and enforces cutoff thresholds to protect cell health. Efficiency and maximum continuous output current depend on the boost stage design and thermal constraints of the enclosure.

Product families span compact single-cell units optimized for portability to larger multi cell enclosures designed for higher capacity and extended runtime. Certification markings, port counts, advertised watt-hours, and supported fast-charge protocols differentiate models. Because the output is regulated and standardized at the port, integration involves treating the power bank as a self-contained source rather than a raw cell, with the battery management and charging logic embedded inside the unit.

3.1.6.5 NiMH

NiMH cells in AA format provide a nominal voltage of approximately 1.2 V per cell and are arranged in series to meet target voltages. The chemistry is known for mechanical robustness, tolerance to repeated charge/discharge, and moderate self-discharge characteristics that have improved in “low self-discharge” formulations. Cylindrical cans and standardized dimensions make sourcing and replacement straightforward.

Electrical characteristics include moderate internal resistance and C-rates that are typically lower than high drain lithium-ion cells but adequate for many consumer and industrial products. Charging methods differ from lithium systems, employing techniques such as negative deltaV detection, temperature rise monitoring, or timed charge to terminate correctly and avoid overcharge. Performance at lower temperatures is generally acceptable, with capacity and voltage response dependent on discharge rate.

Packs can be assembled from off the shelf cells with holders or welded tabs, and they do not require the same level of electronic protection as lithium chemistries, though fuses and thermal sensors are common in higher power applications. Energy density is lower relative to lithium-ion cells, resulting in larger and heavier packs for equivalent watt hours. The chemistry's handling simplicity and wide availability make it a stable option where standardized cell formats and straightforward charging infrastructure are valued.

Table 3.1.7 Batteries

	18650	Li-ion	LiFePO4	USB power bank	NiMH
Voltage per cell	3.6-3.7 V	3.6-3.7 V	3.2-3.3 V	5 V	1.2 V
Energy density	X	X			
Specific power	X	X			
Cycle life			X		
Thermal stability			X		
Form factor		X			
Protection	X	X	X	X	
Output regulation				X	
Capacity range	2-3.5 Ah	Wide range	1-1.6 Ah	Wide range	2-3 Ah
Availability	X	X	X	X	X
Packaging	X		X	X	X

3.1.7 Sensors

A sensor is a transducer that converts a physical stimulus into a measurable electrical quantity according to a defined transfer characteristic. Common classes include resistive elements that change resistance with strain, light, or temperature; capacitive and inductive structures that vary impedance with displacement or proximity; piezoelectric materials that generate charge under stress and inertial devices that report acceleration and angular rate.

Key attributes are sensitivity, range and linearity over the operating span, hysteresis and repeatability across loading cycles, bandwidth for time varying signals, and noise and temperature coefficients that bound resolution and stability. Many sensors incorporate conditioning features such as thin-film resistive networks, shielding, or integrated bridges to improve signal integrity. Flex and bend sensors, such as resistive polymer strips that increase resistance with curvature, fall within the strain sensing family and are specified by nominal resistance, resistance curvature curve, mechanical endurance, and allowable bend radius; their performance is typically characterized under defined test radii and temperature to establish scaling behavior and drift

3.1.7.1 sEMG Electrodes

The sEMG or surface electromyography measures the small bioelectric signals at the skin using microvolts that respond to muscle activation. The intended design considered would be to have electrode tech on all main muscle fibers that are needed to sense. The Pros of the technology was evaluated because it captures intent before actual motion occurs so it would remove the need for physical manipulation of a sensor. Also this type of electrode can control even when a finger motion is restricted.

Ultimately, I decided not to choose this device because of a few factors including noise, safety overhead, crosstalk complications, and cost effectiveness. The electronics require a high gain and low noise analog front ends, some motion suppression and will be delicate when paired with the PWM motor environment. Essentially violating IEC rule 61326-1 substantially with the EMC burden.

Also the user error and bodily functions of human beings like sweat and hair can dampen the effectiveness of the sensing capabilities. The day to day variability of having different users with all different hair and sweat introduces too much risk to justify accurate utility on a recurrent basis.

On the software level, consuming time to code and having independent finger control would require many channels and high level classifiers that ultimately prove to be unnecessary.

Bottom line is sEMG is great for intent decoding but adds a lot of hardware and regulatory challenges to complexity we do not need to concern ourselves with, considering our economic and time constraints.

3.1.7.2 IMU Based Finger Pose

Small IMU's mounted to measure piece by piece orientation, having about three per finger it is possible to triangulate joint angles and drive the actuators. While a model like this would yield high rate orientation data and enable rich motion capture, and good repeatability we did not choose it because of the cost, computation and fusion, environment sensitivity and mechanical integration.
The cost of this would imply 10-15 IMU's total per hand.This would strain the I2C bandwidth and power budget. The power budget would theoretically be beyond what an arduino nano can handle. When also considered that the gyro drift would accumulate and suffer interference not to mention permanently having to do a calibration with every use of the design, it is easy to not choose this method. The demand is simply too much for the controller and it complicates manufacturability.

3.1.7.3 Why the ZD10-100 Flex Strips

The flex sensor couples elegantly with the Arduino Nano so it's the best choice, featuring two tire resistive strips and it changes with applied normal force. It only reads positive absolute values so the hand will never be bending backwards in an unshapely manner. The form of the strip also sits kindly on every finger with little thought into it. Performance for control response time is less than 10 ms and the usable force window matches up with human finger bends even through fabrics such as the glove material layer. Because the strip is passive (no powered electrodes), our EMC problem is smaller and compliance with IEC 61326-1 practices (shielding, grounding, RC input filters) is straightforward. The glove materials and adhesives can be selected/verified against ISO 10993 endpoints (cytotoxicity/irritation/sensitization) without patient-connected electronics at the skin. Usability under IEC 62366-1 is also simpler: users just don't calibrate, move on skin prep, gels, or multi-IMU pairing.

In terms of cost and manufacturability, one sensor per finger keeps BOM and wiring modest, improves serviceability (easy replacement), and supports our IPC-anchored PCB approach for clean analog front ends. It fits our time and budget constraints while delivering the control fidelity we need to validate the hand’s mechanics, safety (max grip force / safe-open), and user workflow.

In short: the ZD10-100 path gives us fast, simple, and robust intent capture that integrates cleanly with the Nano, meets our standards-aware constraints (EMC, biocompatibility, usability), and leaves headroom to iterate the actuator safety and mechanical subsystems—the real heart of the prototype. When those are proven, we can revisit sEMG or IMU-rich variants as a second-generation upgrade, perhaps a flex goal.

Table 3.1.8 Sensors

	sEMG	IMU	ZD10-100
Per finger hardware	2-3	2-3	1
MCU interface	Analog	I2C/SPI	ADC
Latency	X		
Repeatability			X
Calibration			X
Sensitivity			X
EMC			X
Processing			X
Wiring			X
Power demand	X		X
Cost			X
Manufacturability			X
Regulatory			X
Robustness			X
Drift	X		X

3.2 Technology Selection

3.2.1 MCU

Arduino Nano was selected because it provides the shortest path to a stable control platform with exactly the interfaces this project needs in a compact footprint. Its integrated ADC, I2C/SPI, timers/PWM, and USB serial make sensor sampling, packet framing, and actuator command generation straightforward without additional glue hardware. The mature Arduino ecosystem, abundant example code, and wide community support reduce schedule risk and keep bring-up focused on the application rather than driver development. Instant boot and low power suit a wearable/glove context, and the small board size aligns well with tight enclosures.

The trade-offs are limited RAM/Flash, a single hardware UART, and 5 V logic in a 3.3 V centric world are acceptable for this scope because the control loop and packet sizes are modest and the design tolerates lightweight protocols. Level compatibility and memory constraints are well understood and manageable within the intended sampling rates and feature set, so Nano's simplicity and availability outweigh the benefits of moving to a more complex 32-bit platform for this iteration.

3.2.2 Transceiver

nRF24L01 was chosen for the glove to hand link because it is purpose built for short, deterministic packets at high update rates with hardware assist for acknowledgments, retries, and CRC. The SPI interface and small modules keep the electrical and mechanical footprint minimal, and selectable data rates allow a practical balance between robustness and latency. Broad, inexpensive module availability and stable libraries help ensure predictable performance across iterations and spares.

Expected limitations are operation in the busy 2.4 GHz band, 3.3 V only power IO, and minimal built in security are reasonable in the context of short-range control traffic and a contained operating environment. The device's feature set aligns directly with the requirement for steady, low latency pose updates without the complexity of a full Wi-Fi or BLE stack, making it an appropriate choice for this phase.

3.2.3 Motor

MG996R was selected as the actuator class for the hand because it combines adequate torque in a standard package with simple three wire PWM control and a well-supported ecosystem of horns, brackets, and spares. The integrated position loop and metal gear drivetrain provide a straightforward path to tendon-spool actuation without external drivers or encoders, and the cost profile enables multiple degrees of freedom within a student budget. Availability from multiple vendors reduces supply risk and eases replacement.

Known compromises are backlash in the geartrain, audible operation, and lack of host side telemetry are acceptable for a first pass grasping hand where the priority is functional motion and repeatable positioning rather than precision metrology. Current surges at high load are characteristic of this class but are manageable within typical 6 V rail designs, so the overall balance of capability, price, and integration simplicity favors MG996R for the initial build.

3.2.4 Sensors

ZD10-100 flex strips were selected because they provide a direct, passive measurement of finger bend as a simple resistance change that can be read with basic analog circuitry. Response time is fast enough for control loops in the tens to hundreds of hertz, and the strip format conforms well to finger geometry and glove materials. The absence of powered skin contacts simplifies electromagnetic compatibility considerations and reduces biocompatibility and usability overhead compared with electrode-based approaches.

While resistive bend sensors exhibit nonlinearity, material aging, and some temperature dependence, these effects are predictable and acceptable for capturing finger posture in this context. The one-sensor per finger model keeps wiring and bill of materials modest and supports straightforward maintenance or replacement, aligning with cost and schedule constraints while delivering the fidelity needed for reliable pose capture.

3.2.5 MOSFET

BSS138 was chosen as the utility MOSFET for light switching and level translation roles because it turns on at logic-level gate voltages, has very low gate charge, and is widely available in a compact SOT-23 package. Its electrical characteristics are well matched to small-signal loads and signal interfacing, allowing clean control with minimal drive effort and negligible board area impact. Multiple second sources and consistent pinouts simplify procurement and substitution.

The device's current capability is intentionally limited compared to power MOSFETs, but that aligns with its intended use as a signal-level switch or translator rather than a motor or rail switch. Within those boundaries it provides a reliable, inexpensive building block that avoids the overhead of larger packages or gate drivers for light duty tasks in the design.

3.2.6 Batteries

Standard 18650 cells were selected as the base element for the arm power source due to their high energy density, robust cylindrical construction, and broad second-sourcing ecosystem. The format supports a wide range of capacities and discharge ratings, and the metal can aid mechanical handling and thermal conduction within enclosures. Established CC/CV charging profiles and well-documented performance characteristics make system-level planning predictable.

Although multi-cell packs require protection and battery management provisions, these are routine for the format and are well supported by off-the-shelf electronics and pack hardware. The series/parallel flexibility of 18650 cells provides a clear path to meeting future voltage and current targets if the arm is pursued later, without committing to bespoke or hard to source form factors.

A consumer USB power bank was selected for the glove because it integrates lithium-ion cells, protection circuitry, charging control, and a regulated 5 V output in a single certified enclosure. The standardized port interface, state of charge indication, and internal safeguards reduce the number of custom power elements and simplify logistics such as charging and replacement. Market availability is broad, allowing capacity and size to be matched to endurance goals.

While the boost converter and enclosure introduce efficiency overhead and fixed form factor constraints, these are outweighed by the benefit of a turnkey, self-contained supply with predictable behavior and built in protections. For the glove's power demands and operating model, the convenience and safety integration of a power bank align well with time and budget priorities.

Chapter 4 Related Standards and Design Constraints

4.1: Industrial Standards

4.1.1 Biomedical Device Standards

Electromechanical biomedical device for prosthetic uses, as well as laboratory usage. The Bionic hand classification is in part electromechanical, it has to comply with relevant standards to ensure the pillars of safety, reliability and performance are present. This is especially necessary for human interfacing systems. Relevant standards are necessary to ensure the hand design meets requirements for human interaction, electrical safety, and mechanical durability.

Since the device interfaces directly with a human test subject the design must comply with all medical standards for human to robot interaction, as well as intermediate production safety. The below standards ensure that the project bionic hand is not only manufacturable but robust, safe and compliant for human use under any FDA , IEEE and IEC regulations. In addition to the biomedical interpretation of the project we also accounted for the proper laboratory robotics and automation standards and constraints. The vision is to have this device accessible in laboratories for experiments as well as medical capabilities of its design.

IEC 61326-1 – EMC for Lab Equipment

The system we intend to use intertwines low level flex sensor signals with PWM motor currents, in this case the compatibility of the electromagnetism is essential. The rule sets immunity and emissions requirements for lab equipment enabling glove accurate reads even near other RF sources. We intend to leave all walkie talkie and any other possible equipment that is harmful to the readings of the equipment left at a safe distance away from experimentation. Also frequent and consistent records of measurement design and compliance readings for the project. One thing that is possible is to test in hopes of protecting against a charged item of clothing or something that can inadvertently damage the equipment.

ISO 13485

This specification highlights the quality management system for medical devices. Ensure consistent design, and servicing to meet regulatory requirements.

ISO 10993

Covers the biocompatibility of materials that are used in medical devices that come in contact with human skin, ensuring plastics silicone, and three 3D printed parts do not cause irritation or harmful effects like toxicity.

IEC60601-1/2 Medical Electrical Safety & EMC

Creates general safety and performance standards for medical electrical equipment this standard discusses, power limits and electrical isolation. This means that using basic casing and protective wire connectors and proving the system remains safe under all conditions. Effectively we simplify lab approvals and lets teammates handle debugging of all parts including the glove and controller components. Also to meet this standard its best we keep all hot surfaces and drivers below the regular temperature limits.

Robotics and Automation standards:

Additionally, standards must fall under the general umbrella of robotics safety and Automation due to its sensitive and multifunctional capability. Baseline bullets for criteria for robot architecture, motion control, force pressure/ limits in human interaction and finally system risk reduction in shared workspaces.

Health and Safety – User Bystander Protection

The device will have a maximum grip force, all measurements are taken and pinch points will be minimized. We plan to implement an emergency stop wire or functional button to cut off all communication with the sensors and glove. In other words we will incorporate a “before putting on the glove” checklist for all users including the design team. In addition adding a two person rule for first time users to make everything more safe.

ISO 10218 -1: 2025 – Robotics Safety requirements Part 1:

Lists safety requirements for industrial robotics. This particular requirement addresses it as partly completed machinery. It's necessary to have clearly defined sop behaviors programmed for emergencies and the code will include this, to satisfy this rule we will have all accurate documentation on the stopping behavioral procedures and experiment on what those will look like. We have a built in safety function already with designed performance levels, refer to flex sensor specs for movement, less than 20 grams of weight on the sensor will never move. Intuitively, the sensor has precautions for sudden readings as well with the limits in flex sensor specifications.

IEC 62366-1 Usability Engineering

The human error component of this design is prevalent even with proper safety procedures present. To combat this we will design mapped task flows and test for confusion or unsafe readings before each use. Good usability here is also safety, fewer inadvertent grasps and mindful motion of the equipment. It must be stated that each user will be properly informed on procedural guidelines before each use. Another way to combat this is to have an itemized list of things to know before putting on the glove.

ISO /TS 15066: 2016 – Collaborative robots.

International organization for standardization. Applying safety rated monitors stops safe speeds to servos and integrated guidance derived for allowable contact forces. While not deploying a full robot the hand has human proximity with the device and proper speed and separations and basic limits anyone can grasp have to be implemented. We intend to select contact limits from specifications and constrain the torque and speed of the servos. Also verifying a monitored stop is incorporated in firmware.

ISO 13482: 2014

Robotics and robotic devices – Safety and personal care robots. Addresses service/ assistant robots interacting with humans. This helps define acceptable hazards and protective measures for including contact scenarios in non medical settings.

IEEE 1872 – 2015

IEEE standard ontologies for Robotics and Automation, defines ontology for robotics and automation systems. Design compliance approaches structure our software messages and logs to label tasks states and entities all the time. The debugging helps while enabling downstream tools to reason about behavior. It also makes requirements and some cases more explicit for testing.

IEEE 7007-2021

Ontological standard for ethically driven robotics and automation, focuses on ethical framework and robotics. We will ensure human in the loop oversight and the user can always override the device and cannot lock a grip or anything of that nature.

4.1.2 PCB Standards

The central key hardware elements of the device relies on Flex sensors, microcontrollers and motor servos that are interchangeable and integrated on PCB. W will anchor design to IPC 2221 and IPC 6012 to ensure manufacturability and repeatability.

4.1.3 Sensor and Signal Standards

Flex sensors used for motion detection follow the IEEE standards, including IEEE 1451 for smart transducers. This standard covers the calibration, digital communication and identification methods for sensors. The chosen flex sensors are the ZD10-100, a FSR style resistive strip. Applying force changes its resistance values so the user can have full operational control of the hand portion of the device with one flex sensor over each finger.

The Sensors are finger mounted to generate repeatable pressure on the sensor. Yielding a 0-500 gram rand is great for gloves since the curling of the finger under a pad can generate a lot of weight in that range and always clearing eh less than 20 gram activation threshold. In tandem with the software the fast response of less than 10 ms supports low latency control and the cycles rating is great for frequent usage that will last a long time.

In terms of the safety of the sensors the sensor values are designed to only read absolute values of the sensor so even if the user were to make irregular movement with their model hand, the device would just open and close its hand without any backwards joint motion or unsafe usage.

Flex Sensor (ZD10 -100) Design parameters

Sensor raw signals are prone to noise, drift, and variability. These problems are handled by the processing layer:

- **Calibration:** Flex sensor outputs are normalized for varying hand sizes and natural resting states by taking baseline readings for each user at startup.
- **Filtering:** A low-pass filter removes high-frequency noise from the IMU, while a moving-average filter smoothes out abrupt oscillations.
- **Mapping Functions:** Valuable joint angles are derived from sensor measurements. Servo angles of **0 to 90°**, for instance, are mapped to a flex sensor range of 0.5 to 3.5 V. When joint motion is not exactly proportional to voltage, nonlinear mapping functions are employed.
- **Data Packaging:** Finger angles, wrist orientation, and metadata like sequence numbers are all included in the small data packets that are created from the processed values.

Table 4.1 Flex Sensor Device Data

Parameters	Data
Length x Width:	100 x 10 mm
Thickness	< 0.25 - 0.30 mm
Durability:	> 1,000,000 actuations
Measurable Range:	0-500 grams normal force

Response/ Recovery:	< 20g response point

4.1.4 Actuator

The Servos and motors controlling finger motion fall under the NEMA and IEC standards umbrella:

NEMA 17 or NEMA 14 micro stepper/ servo motors are used for compact torque control.

ISO 10218 -2 – if the bionic hand is mounted to a robot arm or otherwise interacts in a cell this governs integration and operational maintenance.

IEC 60034-1 for the mechanical performance/ thermal limits and duty cycles.

The intention of the actuator subsystem is to convert glove flex sensor into finger motion for the current prototype we target the tendon driven fingers via flex sensors, powered via rotary servos around 5-6 Volts. This is a great choice due to its budget friendliness and compact arduino functionality. This architecture is fail safe to open hand when power is removed aligned with iec 80601 -2 -78 expected performance expectations.

4.1.5 Communication and Control Standards

Internal control system utilizing the I2C and SPI communication between the microcontroller and servos. The microcontroller of the robotic arm interprets incoming packets and uses PWM signals to drive servo motors:

- **PID Controllers:** A PID loop is used by each joint to minimize overshoot and guarantee precise positioning.
- **Safety Constraint:** Software limits keep the servos from going beyond their mechanical ranges, which is a safety constraint. Values are clipped to safe limits if data is received that is out of range.
- **Fail-Safe Procedures:** The arm returns to its neutral rest position after 500 milliseconds if no valid packet is received. Before putting on the glove procedures.

- **Servo Synchronization:** To produce organic, coordinated movement, motion is interpolated across several joints.

The following tests are carried out to guarantee performance:

- **Latency Tests:** End-to-end response measured between finger movement and arm response (<100 ms target).
- **Stress Testing:** Continuous operation for 30+ minutes without overheating or communication dropouts.
- **Error Injection:** Artificially corrupted packets verify CRC detection and recovery routines.
- **Unit Tests:** Validate calibration, mapping functions, and PID controller logic.
- **Integration Tests:** Full glove-to-arm operation across multiple users to confirm reliability.

4.2 Practical Constraints

Time remains a major constraint of the project since a senior design capstone project development is limited to 2 semesters along other classes. By May spring 2026 the concept validation, circuit design and coding need to be done along with mechanical assembly. Some delays we can run into is the PCB shipping/ manufacturing and the constant adjustments to 3D prints and fittings.

We regularly have changed our design or even something as small as one finger over five times. To mitigate the stress of this we established weekly milestones and we emphasize work in pieces. The tasks get completed well in a timely manner thanks to a great group of engineers.

4.2.2 Economic

- The total budget allocated by the sponsor or university is $500 - $800.
- Sensor Choice: Flex sensors are ($12-20) limit redundancy in design.
- Microcontroller selection: Arduino Nano

- Actuators: Mini servos instead of high end brushless motors

Table 4.2 Budget Table

Hardware Expenses		
Part	Cost ($)	Quantity
ZD10-100 flex sensor 500g	8.81	5
MCP1700 3.3V regulator (10pcs)	8.99	1
mg996r servo motors (4 pcs)	15.98	1
nRF24l01 (4pcs)	7.89	1

4.2.3 Environmental

Several environmental constraints exist, temperature range 0 -40 degrees for normal operation. Humidity < 70% non-condensing to prevent corrosion Dust Resistance including Plastic or silicone casing to protect joints and circuitry.

4.2.4 Sensor and signal Constraints

Flex Sensor constraints would be the electrical range resolution, nonlinearity, noise, Mechanical integration, durability and firmware. Starting with the electrical resolution the Nano has an ADC of 10 bits with a 5 V reference, with respect to its internal 1.1 V reference the dividers will hardly ever grab the full 1.1 V reference. The team has considered using an op amp gain to improve this resolution, and source impedance considering the glove material to be less than 10 k ohms.

Film based flex devices like the flex sensors in most cases are highly nonlinear and tend to experience drift with longevity stricken conditions. Also the servos and motors add some unwanted noise to the equation. We physically want to separate the sensor wiring from motors.

The Arduino Nano Constraints (ATmega328P)

Basic specification of the microcontroller:

Classic Nano = 16 MHz

2KB SRAM

ADC performance

Throughput of ADC = 10 kS/s

ADC clock can raise 50-77 kS/s but this costs some accuracy which we intend to not need to risk in any case. The I/O pins feature 8 analog channels (A0-A7), PWM with 6 channels at a timer dependent 490-980Hz. Worth noting intention to design regulators for use on the side of the PCB design to keep within a good voltage range. Power and thermal respect 20mA per pin with a maximum of 40 mA equalling 200 mA total MCU current.

4.2.5 Ethical

Sourcing factors into many ethical constraints when conducting research, some companies use sickening or otherwise disgusting practices to conduct research and manufacture products. Our project doesn't intend to take advantage of third parties or exploit any work. All companies chosen that went into our project are reputable in their business practices and have good standing in the general engineering community.

4.2.6 Sustainability

The design in how it is envisioned allows for interchangeability with the sensors and motors. Interchangeability allows for reusability and more room to modulate or play around with the framework. In addition to the casing mentioned in other sections we intend to have safe grounded casing for the device to always have for protection and longevity.

Chapter 5 Chat GPT Comparisons and Usage

5.1 Introduction

In November 2022 Chat GPT, was released and has evolved at an exceptional rate. Our team was able to work with this AI to verify our documentation, enhance our vision of the project and gather useful information with excellent efficiency. In general the case between members usage varies while the software design would benefit from help with chat GPT, our usage of it was limited.

Chat GPT has helped us all clarify technical concepts and its integration with its google search engine was beneficial as a trusted source of information. Some parameters and extended knowledge was gained with excellent bibliography to quickly get to our sources. We will comment from a standpoint of efficiency and productivity ChatGPT is a great asset in almost any technical space.

The rapid design framing is great as well for scoping and turning the glove into hand into a concrete plan of action. It can draft block diagrams and it was helpful for us to visualize how we want the components to set us on the hand. The catch is sometimes it can be confident in wrong responses so its best to keep its use somewhat limited for extremely high level tasks.

5.2 Case Study 1 — Protocol Selection: SPI (nRF24L01) vs BLE

Early in the semester, our team had not yet decided which wireless interface would connect the glove subsystem to the robotic hand. Two primary options emerged:

1. Bluetooth Low Energy (BLE)
2. nRF24L01 SPI Radio Modules

Our initial assumption favored BLE because it is widely supported, used in wearables, and easy to pair with consumer devices. However, when we asked ChatGPT to compare BLE with the nRF24L01 in a point-to-point real-time control system, the tool provided several insights:

- BLE requires pairing and acknowledgment layers that increase latency.

- BLE is optimized for low-bandwidth, intermittent communication rather than fast servo updates.
- nRF24L01 modules offer low-latency (<1–5 ms) packet transmission and configurable data-rate profiles.
- BLE is more prone to interference when operating in crowded 2.4-GHz environments.

Using ChatGPT's analysis, we validated our final communication choice by reviewing multiple datasheets and community hardware reports from robotics forums. These sources reinforced concerns about BLE latency and connection stability. The nRF24L01-based SPI interface was therefore chosen based on:

- Faster continuous packet transmission
- Fewer software layers between the application and RF hardware
- Bidirectional payload support with ACK ability
- Simpler firmware implementation

This early decision was crucial because communication choice determined how firmware, packet structure, and timing loops would be designed. ChatGPT helped accelerate the research process by summarizing trade-offs and helping us articulate justification to our professor. Ultimately, the decision aligned well with our performance goals and supported a cleaner first-semester implementation.

5.3 Case Study 2 — Firmware Mapping and Curve-Shaping Techniques

One of the most important steps in the GRIP system is mapping raw flex-sensor voltages into usable joint-angle commands. Our first implementation used linear mapping between ADC values (0–1023) and servo angles (0–180°). This worked mechanically but produced abrupt motion near low-bend regions and poor precision during fine hand movements.

We consulted ChatGPT for alternative mapping methods. Several options were discussed:

- Exponential curve shaping
- Polynomial approximation
- Logistic compression
- Piecewise mapping
- Moving-average smoothing

From these, ChatGPT recommended a tunable exponential mapping function such as:

$$\theta = (norm)^k * 180\, degrees,\ where\ k \approx 4 - 5$$

This form improved sensitivity for smaller bends and provided controllable smoothness. After testing, we selected power ≈ 4.2 as a balance between responsiveness and stability. Additionally, ChatGPT suggested averaging windows of 3–5 samples to reduce jitter. We implemented a small median-plus-mean smoother to further stabilize noisy readings.

This approach dramatically improved movement quality. Even before integrating the robotic hand, we achieved responsive visualization inside our C# arm-simulation tool. The assistance from ChatGPT accelerated our transition from naïve mapping to a more principled signal-conditioning pipeline.

5.4 Case Study 3 — System Block-Diagram Refinement

Our early block diagrams were very detail-heavy, including explicit ADC-conversion boxes, register-stage mapping, and other low-level operation layers. During design review, peers warned that this might appear overly granular, especially in SD1 where we had not yet implemented all modules.

We asked ChatGPT whether ADC conversion must appear explicitly. The tool emphasized that while ADC is necessary for analog flex sensing, it is typically implied within the MCU's data-acquisition block. Only when conversion timing, sampling strategy, or precision trade-offs are important should it be diagrammed separately.

Given that SD1's primary focus was the glove–hand pipeline (not ADC theory), our diagrams were simplified:

Old (too detailed):

Sensor → Voltage Divider → ADC → Filter → Exponential → Mapping → Packetization

Revised SD1 diagram:

Sensor → Pre-processing + Mapping → Packetization → Transmission

This simplification better matched the professor's guidelines for SD1, keeping emphasis only on the most functionally meaningful components. ChatGPT's guidance ensured our diagrams were leaner, more readable, and aligned with higher-level system requirements.

5.5 Case Study 4 — Visualization Tool Selection (C# Simulation)

Before full mechanical integration, we wanted to test joint-angle behavior using a simulated robotic arm. Initially, we considered MATLAB, Python (MatPlotLib + Tkinter), Unity, and C#.

ChatGPT outlined the following trade-offs:

Table 5.5.1 ChatGPT generated comparisons

Platform	**Pros**	**Cons**
MATLAB	Fast prototyping, built-in math tools	License, less portable
Python	Simple, flexible, highly accessible	GUI rendering slower
Unity	Powerful graphics	Overkill, steep overhead
C#	Strong UI tools, IK libraries, lightweight	Requires structuring

Based on these comparisons — and our preexisting familiarity with C# — we selected a C#-based desktop simulation that implemented both Forward Kinematics (FK) and Inverse Kinematics (IK).

This tool became one of the most useful outcomes of SD1:

- Allowed real-time angle visualization
- Enabled drag-based IK testing
- Allowed UI display of θ_1 & θ_2 values
- Provided initial validation without hardware risk

ChatGPT also provided useful pseudo-code guidance for 2-link IK geometry.

5.6 Case Study 5 — Safety Control: Servo Limits & Fail-Safe Behavior

Real-time servo control poses safety risks: sudden angle jumps can damage hardware. When consulting ChatGPT about motor-control reliability, it recommended several best-practice techniques:

- Angle limiting (software clipping)
- Neutral-pose fallback when communication is lost
- Slew-rate limiting where $\Delta\theta$ per cycle is capped
- Interpolation when packets are dropped

We implemented two of these during SD1:

1. Angle clipping ensured that no command exceeded the physical servo range.
2. Timeout fallback puts the servo into a neutral posture if >500 ms passed without valid data.

Implementing these behaviors early produced more stable and predictable motion. It also established safety patterns that will be extended next semester when PID and shoulder DOF are added.

5.7 Case Study 6 — Code Documentation & Readability

We also used ChatGPT to help refine firmware structure and documentation. While ChatGPT did not write code for us, it helped reformat messy blocks into cleaner functions (e.g., separating readSensors(), applyMapping(), and sendPacket()), which improved readability.

The assistant also helped suggest comments and header structure, resulting in more maintainable code.

5.8 Case Study 7 — Literature & Prior-Art Review

ChatGPT helped us summarize relevant concepts, including:

- 2-link IK geometry
- Nonlinear control curves
- Wireless-packet CRC mechanisms
- Calibration routines

These summaries guided deeper manual research using vendor datasheets and robotics references.

5.9 Limitations of ChatGPT

While valuable, ChatGPT had several limitations:

- Sometimes provided inaccurate hardware current specs
- Occasionally suggested libraries not compatible with Arduino Nano
- Provided simplified examples requiring manual validation

- Needed cross-checking against datasheets and real testing
- Does not replace instructor approval, peer review, or experimentation

Thus, ChatGPT was most useful early in the design process — not as a replacement for technical evaluation.

5.10 Lessons Learned

1. ChatGPT accelerates research but must be validated with real testing.
2. Its greatest value is summarizing trade-offs when choices are unclear.
3. It improved our ability to articulate design justification.
4. It helped structure firmware more cleanly by suggesting modular organization.
5. Overuse can lead to vague justification — final reasoning must come from testing or literature.

5.11 Conclusion

In summary, ChatGPT played a meaningful but appropriately bounded role in the GRIP SD1 software workflow. By providing rapid conceptual comparison, architecture brainstorming, and code-organization suggestions, it helped us reach important decisions more quickly — particularly the selection of SPI-based radios and exponential mapping functions.

It did not replace engineering judgment, debugging, or experimentation. Instead, it functioned as a design accelerator, helping us narrow choices before validating them through hardware testing and instructor feedback.

The integration of AI-assisted research with hands-on firmware development contributed to a well-organized and functional SD1 software pipeline, positioning the GRIP team for successful expansion into shoulder DOF and PID control during SD2.

Chapter 6 Hardware Design

6.1 Transmitter circuit design

The transmitting side of the project is the glove worn by the user, where flex sensors are attached to the fingers and used to measure the curvature of them. The ZD10-100 is used for this purpose. It is a variable resistor that increases in resistance when bent. Each flex sensor has one pin to ground and the other pin is connected to 5V with a 10k ohm resistor between the pin and 5V source, creating a voltage divider circuit. The output of this circuit is attached to an analog Arduino pin, where the changes in voltage due to the varying flex sensor resistance are measured. This process will be repeated for the other 4 flex sensors and the analog values will be put through an algorithm to translate them into servo positions for each of the fingers.

After the analog signals are collected, they will need to be transmitted to the arm via the nRF24l01 transceiver module. However, this module runs on 3.3V logic while the Arduino runs on 5V logic, direct connections can damage the nRF24. To fix this, we utilize the BSS138 N-type MOSFET to shift from a 5V signal from the Arduino to a 3.3V signal the nRF can properly use.

Powering this circuit, the Arduino Nano will supply the 5V needed by the flex sensor circuits via the 5V pin on the board. However, the nRF24, despite only needing 3.3V to be powered, requires more current then the Arduino can provide so a voltage regulator will be used. Powered by the Arduino's 5V pin, the MCP1700 steps the input voltage down to 3.3V while producing 250mA of current to be used, more than enough to satisfy the nRF24 at peak loads and prevent potential brownouts.

Furthermore, capacitors are added to the Vin pins of the nRF24 and the output of the MCP1700 to smooth signals. The 0.1uF and 10uF capacitors used on the nRF24 are linked in parallel between the module's Vin and GND pins to mitigate noise. The 0.1uF capacitor reduces the high frequency, fast switching noise produced by the module's internal circuit, while the 10uF provides extra charge for low frequency dips during transmission spikes. The capacitors at the MCP1700 output are used to stabilize the output and help it respond quickly to sudden changes in current demand from the nRF24.

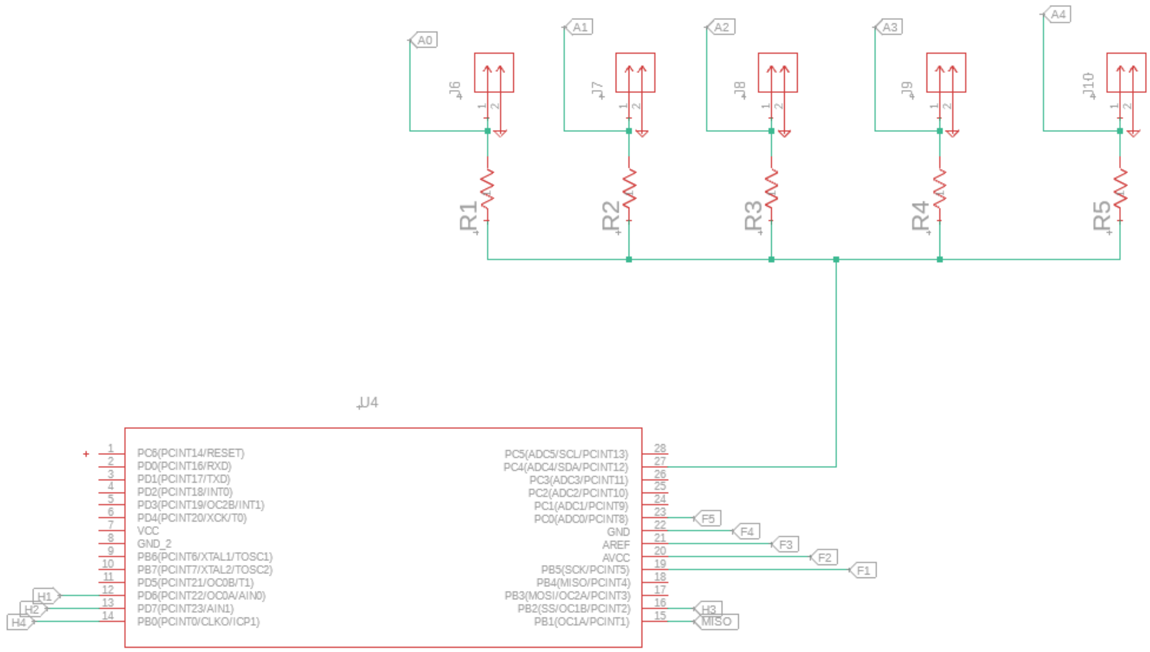


Figure 6.1.1 flex sensors with arduino connections

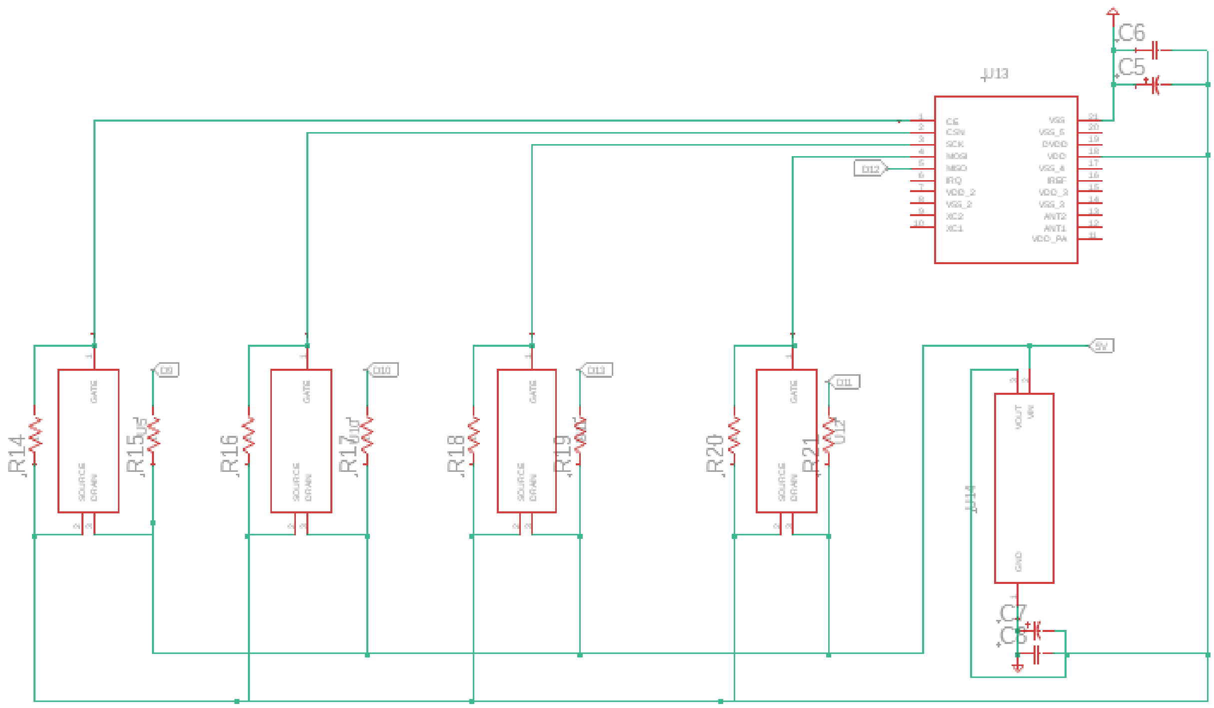


Figure 6.1.2 nRF24 to voltage regulator connections

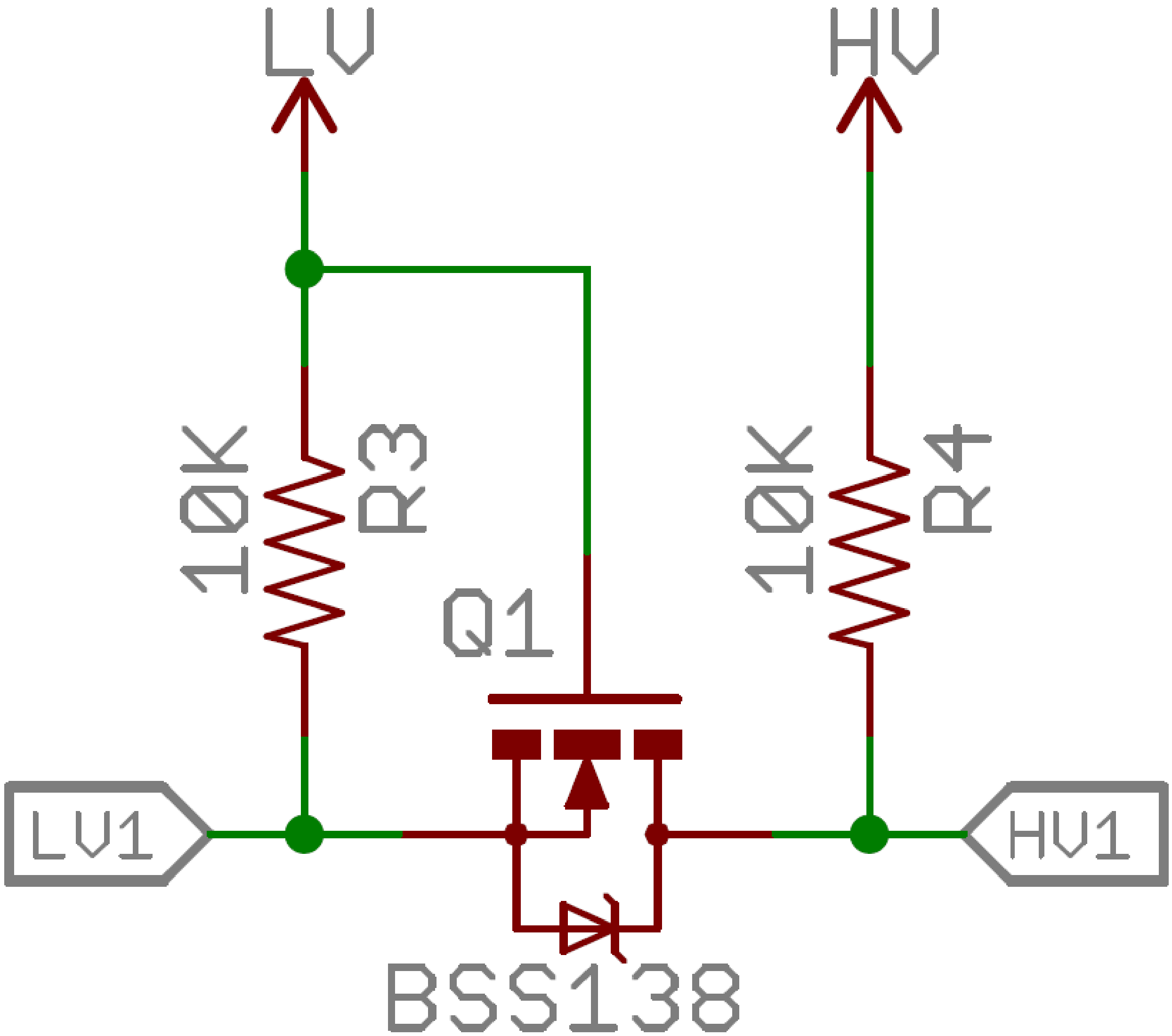


Figure 6.1.3 example of BSS138 MOSFET level shifter circuit

6.2 Receiver circuit design

The receiving side of the project is the arm itself, where the flex sensor values transmitted from the glove's nRF24 will be received by the arm's same module. These values will be translated into positions for the mg996r servo motors controlling the fingers. The Arduino will be powered via a portable power bank, while a 6 volt, 15 amp power supply will be chosen to power all motors simultaneously. Each motor can have a max current load of ~2.5A, with the potential for all of them to reach their max load at once, a minimum 12.5A is necessary for bare minimum requirements, but a 15A supply is chosen as a safety buffer in case of unexpected current draw. 6V was chosen because at this level the motors are able to deliver peak torque of approximately 12kg/cm.

Similarly to the transmitter circuit, the Arduino will be powered via USB and the 5V pin will give power to the high voltage input of the BSS138 circuit and the input of the MCP1700 regulator. The MCP, using the same capacitors mentioned in section 6.1.1, will provide a steady 3.3V to the low voltage input of the BSS and Vin of the nRF24l01 which is used to receive the packets sent by the glove. Packets received from the glove will be sent from the nRF24 to the logic shifter circuit from 3.3V to 5V. Then, after level shifting, the signals are sent to the Arduino which translates the flex sensor values into positions for the motors. All capacitors mentioned in section 6.1 are used for the same purpose in this circuit.

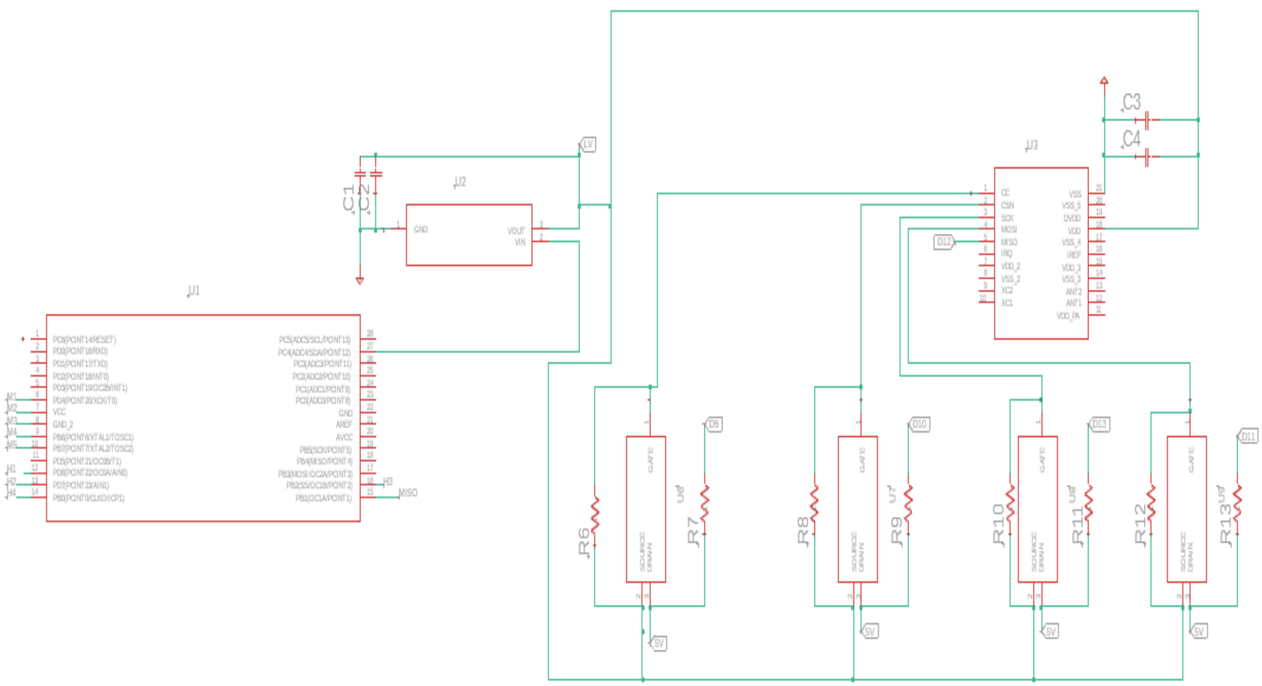


Figure 6.2.1 receiver circuit (servos not included)

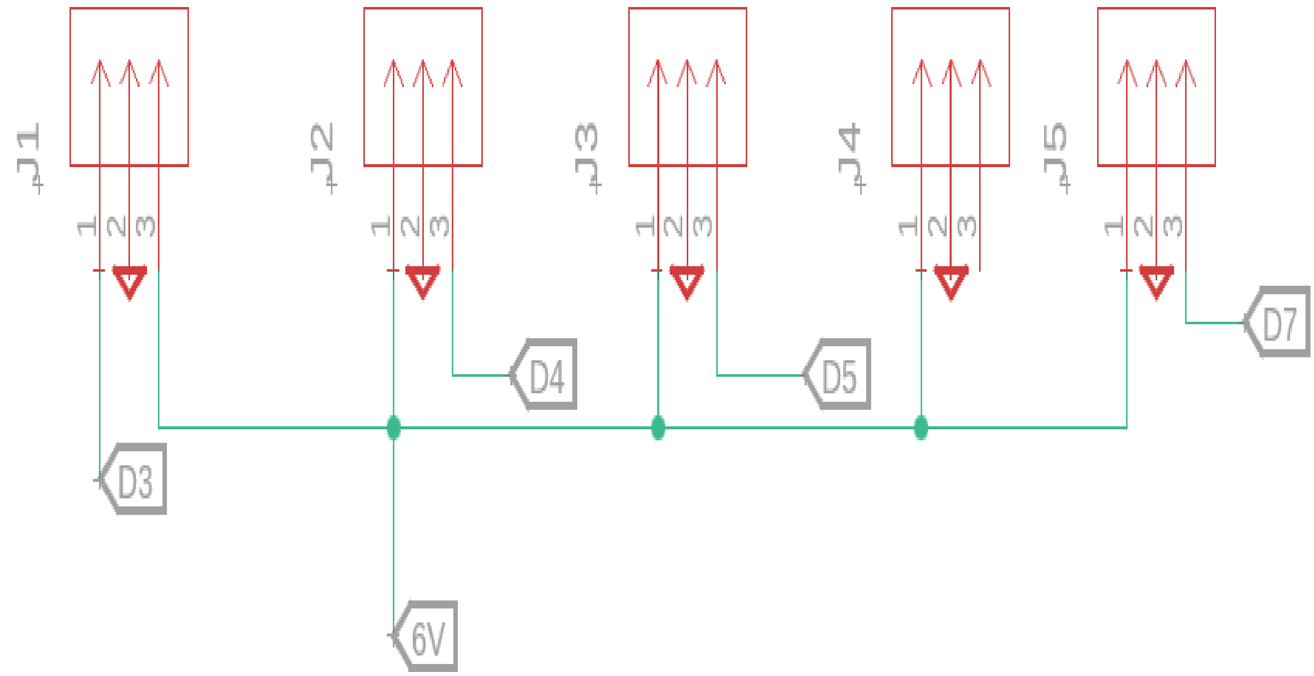


Figure 6.2.2 mg996r servos with arduino ports

Chapter 7 Software and System Integration

7.1 Overview

7.1.1 Abstract

The GRIP (Gesture-Responsive Integrated Prosthetic) system is a two-semester senior design project that focuses on developing a robotic arm capable of mirroring the natural motion of a human hand. This is achieved through a sensor-equipped glove that detects flexion and orientation, and a software-driven control system that interprets and transmits this data to a robotic arm in real time. As the software engineer for the team, my focus is on developing the embedded firmware, communication protocol, and calibration tools that allow seamless synchronization between the glove and the arm. The goal is to create a low-latency, reliable, and power-efficient system capable of smooth and precise movement replication.

7.1.2 Software Objectives

1. Design and implement the firmware for both the glove and robotic arm microcontrollers.
2. Develop a NRF24L01 2.4 GHz radio (SPI) communication protocol to transmit sensor data efficiently and securely.
3. Create a calibration and visualization tool for testing and debugging.
4. Implement a control algorithm to drive servo motors with realistic motion and minimal latency.
5. Ensure safe operation through fault detection and fail-safe software routines.

7.1.3 Software Role in the System

The GRIP software acts as the "brain" of the entire system. It connects the hardware subsystems — sensors, motors, and communication modules — into a unified, synchronized network. Data from the glove is continuously collected, processed, and transmitted to the robotic arm, which then interprets and replicates the user's motion. This process happens in under 100 milliseconds, allowing near real-time motion tracking. The software also provides a foundation for expandability, enabling future additions such as pressure sensors, advanced filtering algorithms, and adaptive grip control. By building a modular architecture, the system can evolve with new hardware and algorithmic improvements without major redesign.

7.2 — Software Architecture and System Layers

Software Architecture Overview

The GRIP software architecture follows a three-tier modular design that separates sensor data acquisition, signal processing and mapping, and communication / control logic. This layered structure ensures that each subsystem can be tested independently, simplifies debugging, and allows future upgrades (such as new sensors or improved algorithms) without redesigning the entire codebase.

7.2.1. Data Acquisition Layer (Glove Subsystem)

- Collects raw input from the *flex sensors* and the *IMU*.
- Each flex sensor outputs an analog voltage proportional to finger bend; these signals are digitized through the MCU's *10-bit ADC* at a *20-50 Hz* sampling rate.
- IMU data (3-axis acceleration + gyro) tracks wrist rotation and movement speed.
- Interrupt-driven sampling maintains precise timing and minimizes CPU load.

• A short calibration routine records baseline voltages for each finger at startup to normalize readings between different users.

7.2.2. Processing & Mapping Layer

• Converts sensor voltages into usable joint-angle values for the robotic arm.

• Implements a *low-pass filter* and *moving-average smoothing* to reduce noise and remove high-frequency spikes from the IMU.

• Applies a *non-linear mapping* so servo angles (0–90°) correspond naturally to finger bends (≈ 0.5–3.5 V).

• Packages processed data into compact frames containing: finger angles, wrist orientation(yaw + pitch), packet sequence ID, and CRC checksum for error detection.

7.2.3. Communication & Control Layer (Robotic Arm Subsystem)

• Uses an *NRF24L01+ 2.4 GHz radio link* (the radio is controlled over SPI on the MCU side) for wireless transmission between glove and arm using compact payloads(≈ 10-byte packets).

• Arm-side MCU decodes packets and updates servo targets using PWM signals.

• PID is planned for the next phase; the current prototype uses nonlinear mapping and optional slew limits to smooth motion.

• *Safety features:* angle clipping and a fail-safe that returns joints to a neutral pose if no valid packet is received for >500 ms.

7.2.4. System Flow Description

1) Sensors → MCU ADC (Glove) → 2) Filter / Map → 3) NRF24L01 + 2.4GHz Link → 4) Arm MCU Receives → 5) PID Control → 6) Servo Output → 7) Monitor errors & safety.

This cycle repeats every 10–20 ms to maintain real-time synchronization between human movement and robotic response.

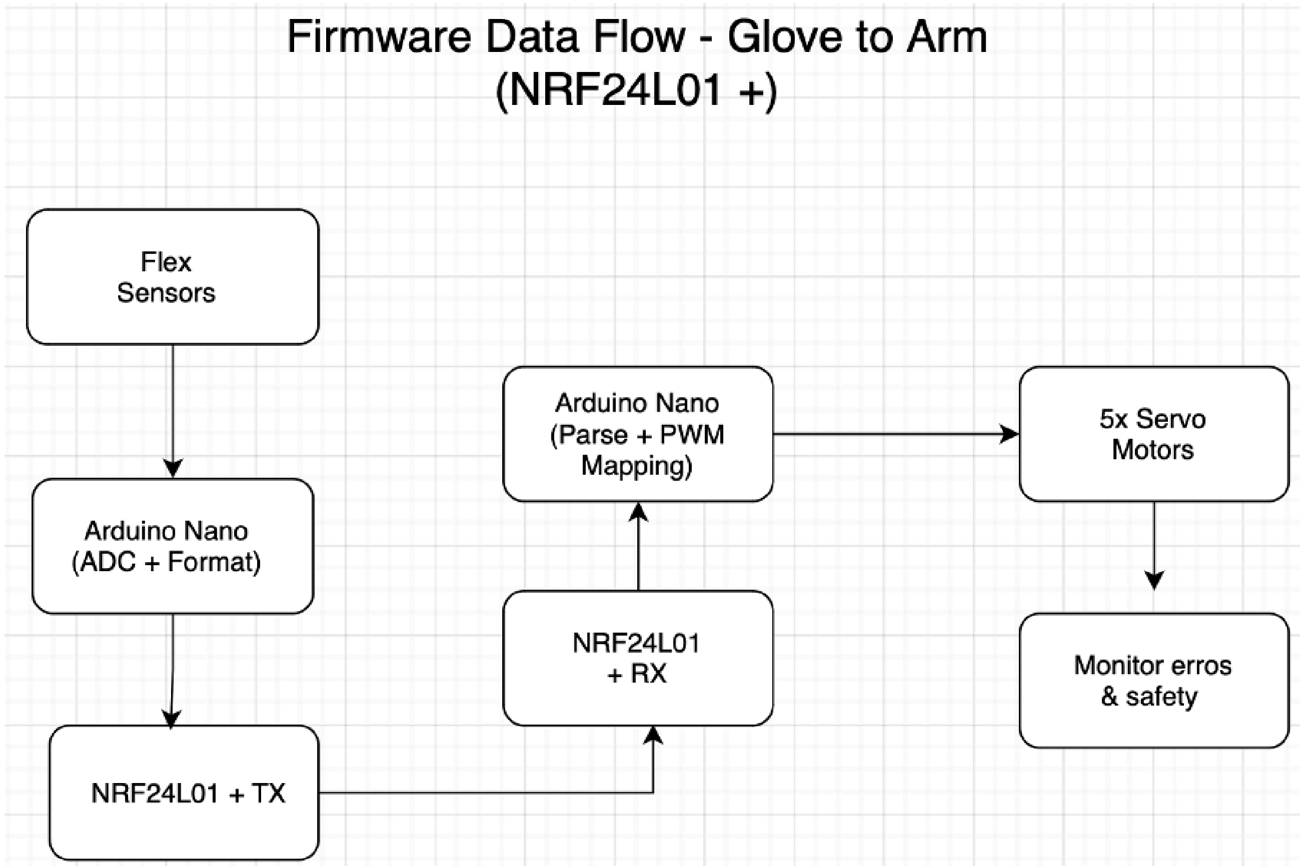


Figure 7.2.1 Software flowchart for whole system

7.3 Updated System Overview

7.3.1. Purpose

Mirror human hand/arm motion on a robotic arm with low latency, smooth motion, and safe behavior.

First-semester focus: prove the full signal pipeline from flex sensor → mapping → radio → servo/visualization and add a shoulder DOF.

7.3.2 Subsystems at a Glance

- Glove MCU: Arduino Nano (10-bit ADC, A0–A4).
- Radio Link: NRF24L01 2.4 GHz (SPI).
- Arm MCU: Arduino Nano/Uno (PWM to servos).
- Actuators: Servos for shoulder + elbow (fingers optional
- now).

- PC Tool: C# 2D arm (FK/IK) for visualization + testing.

A. End-to-End Data Path (Prototype)

1. Flex Sensing

- Five flex sensors in voltage dividers → ADC 0–1023.
- Typical rest values ~500–800 (varies by glove, hand, dividers).

2. Pre-Processing (Glove)

- constrain() to reject outliers; normalize to [0,1].
- Nonlinear curve (power ~4–4.5) to improve fine control near light bends.
- Optional moving average (window 3–5) to cut jitter (tunable).

3. Packetization (Glove)

- Fields (current prototype): [seq, f1…f5, flags, crc] (compact bytes).
- Optional (future): IMU-derived orientation (yaw/pitch) if an IMU is integrated
- Rate: ~20–50 Hz (balanced for smoothness & power).
- Behavior: If radio busy → retry next tick; seq wraps at 255.

4. Transport (NRF24)

- 2.4 GHz, short payloads, auto-ack available if enabled.
- Target link latency: <10–20 ms; end-to-end: <100 ms.

5. Receive + Mapping (Arm)

- Validate CRC/seq; drop duplicates; detect gaps.
- Map byte angles → servo degrees (0–180°) with range clipping.
- Optional slew-limit (e.g., max Δ°/cycle) to avoid step jumps.
- Fail-safe: If >500 ms without valid packet → neutral pose.

6. Visualization (PC, C#)

- Mirror the same θ values; FK draws the arm; IK when dragging target.
- HUD shows θ_1 (shoulder), θ_2 (elbow); clamps outside workspace.

B. Wiring & Channels (Prototype Snapshot)

Glove side (Nano):

- A0–A4: flex sensors (with 10 kΩ dividers).

- SPI: NRF24 (CE/CSN on D pins).
- 5 V rail for dividers; shared GND.

Arm side (Nano/Uno):

- PWM pins → Shoulder Servo, Elbow Servo (fingers optional).
- SPI: NRF24 receiver; 5–6 V external servo supply, shared GND.
- Decoupling at radio (e.g., 10–47 µF + 0.1 µF).

C. Calibration & Mapping Details

- Startup calibration (glove):
    - Record rest value per sensor (median of N samples).
    - Store offsets in RAM (EEPROM planned).
- Normalization: norm = (raw - rest) / span, clamped to [0,1].
- Curve shaping: curved = pow(norm, k); typical k = 4.0–4.5.
- Angle map: angle = round(curved * 180); per-joint min/max tables planned.
- Per-user profiles (planned): store rest + span + k per finger.

D. Safety & Robustness

- Angle clipping per servo to protect mechanics.
- Packet watchdog: neutral if timeout >500 ms.
- Sequence tracking: detect drops; hold-and-interpolate when ≤2 packets lost.
- Power notes: NRF24 peak ~115 mA; keep cap near module; separate servo supply.

E. Current Status (First Semester)

- Flex sensing → nonlinear mapping → servo response working on bench.
- Shoulder DOF added to firmware + C# model.
- C# FK/IK stable; unreachable targets clamp to boundary.
- Preliminary loop rate 20–50 Hz feels smooth; latency budget on track.

F. Known Limitations / Next Steps

- PID control planned; current control uses nonlinear mapping + optional slew limiting to avoid step changes.
- IMU integration future (not in the current control loop).
- NRF24 auto-ack/retry policy to be tuned for crowded RF spaces.
- Per-user calibration profiles in EEPROM; export/import via PC tool.

Expand from shoulder+elbow → fingers + wrist once mechanics are ready.

7.4 Updated System Architecture

The GRIP system includes four primary functional layers:

1) Input — Flex Sensor

The flex sensor measures bend, producing an analog value.

Raw values fluctuate between ~20 and 1023 depending on bend severity.

2) Processing — Arduino Firmware

The Arduino firmware:

- Reads sensor values
- Normalizes input to a 0.0–1.0 range
- Applies exponential curve to smooth fine control
- Maps the curved value into joint angle ranges
- Outputs PWM signals to servo motors

An exponential curve (power ≈ 4–5) provides better low-end sensitivity and prevents jitter.

3) Output — Servo Motors

Servo motors drive:

- Shoulder joint
- Elbow joint

Angles are constrained to remain within mechanical safe limits (0–180°).

4) Visualization / Testing — C# App

A 2-segment virtual arm is rendered onscreen using calculated joint angles.

This software supports:

- Forward Kinematics (FK)
- Inverse Kinematics (IK)
- Keyboard angle control
- Mouse end-effector drag control

The IK implementation uses standard 2-link geometry to compute shoulder + elbow angles based on x,y target position.

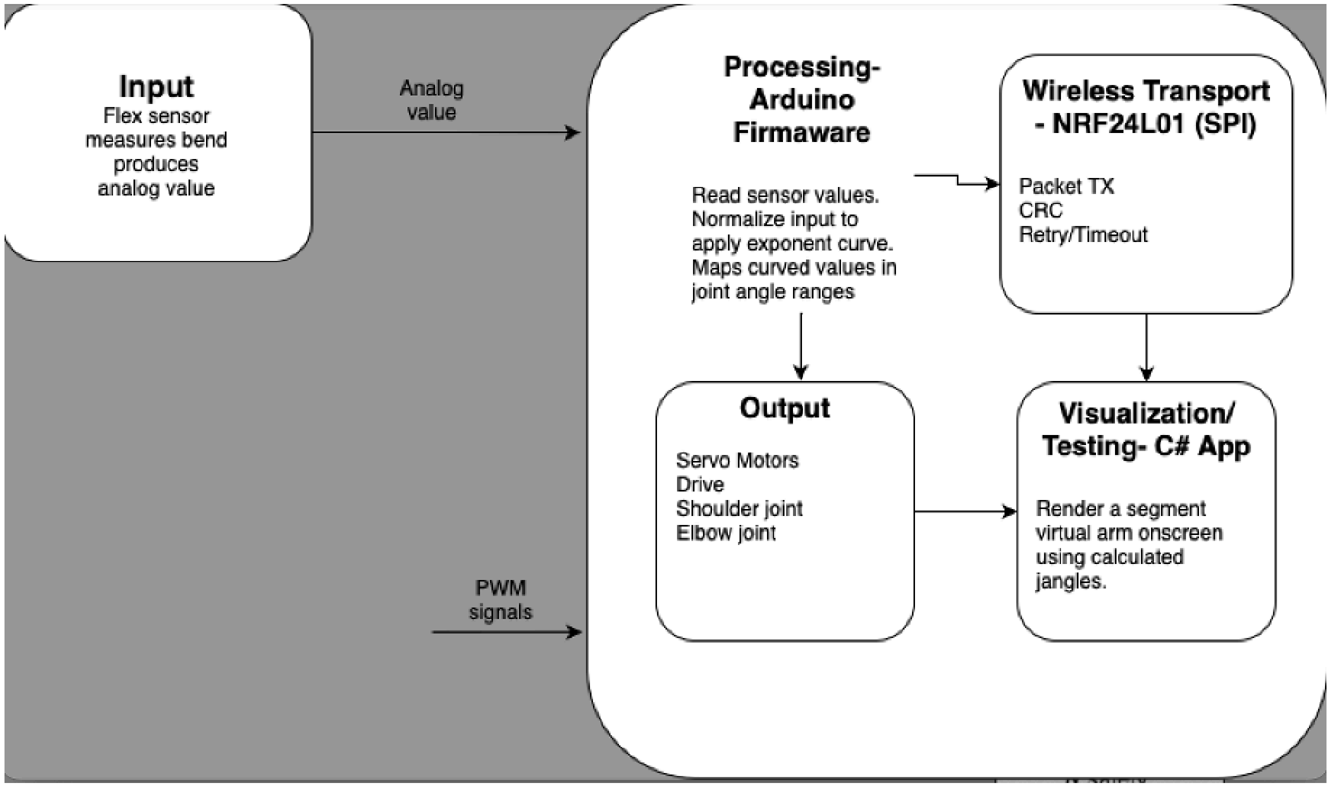


Figure 7.4.1 Firmware flowchart

7.5 Updated Software Flow

Firmware Flow Summary

1. Read analog input from flex sensor
2. Constrain values to stable region
3. Normalize between 0–1
4. Apply exponential curve shaping for smoother motion
5. Convert curved result → joint angle
6. Output via PWM to servo
7. Print serial telemetry for testing

This mapping algorithm offers more stable control than linear scaling.

Key Firmware Elements

- Use of pow() for exponential scaling

- constrain() prevents invalid values
- Servo commands through myServo.write(angle)
- Serial output aids calibration

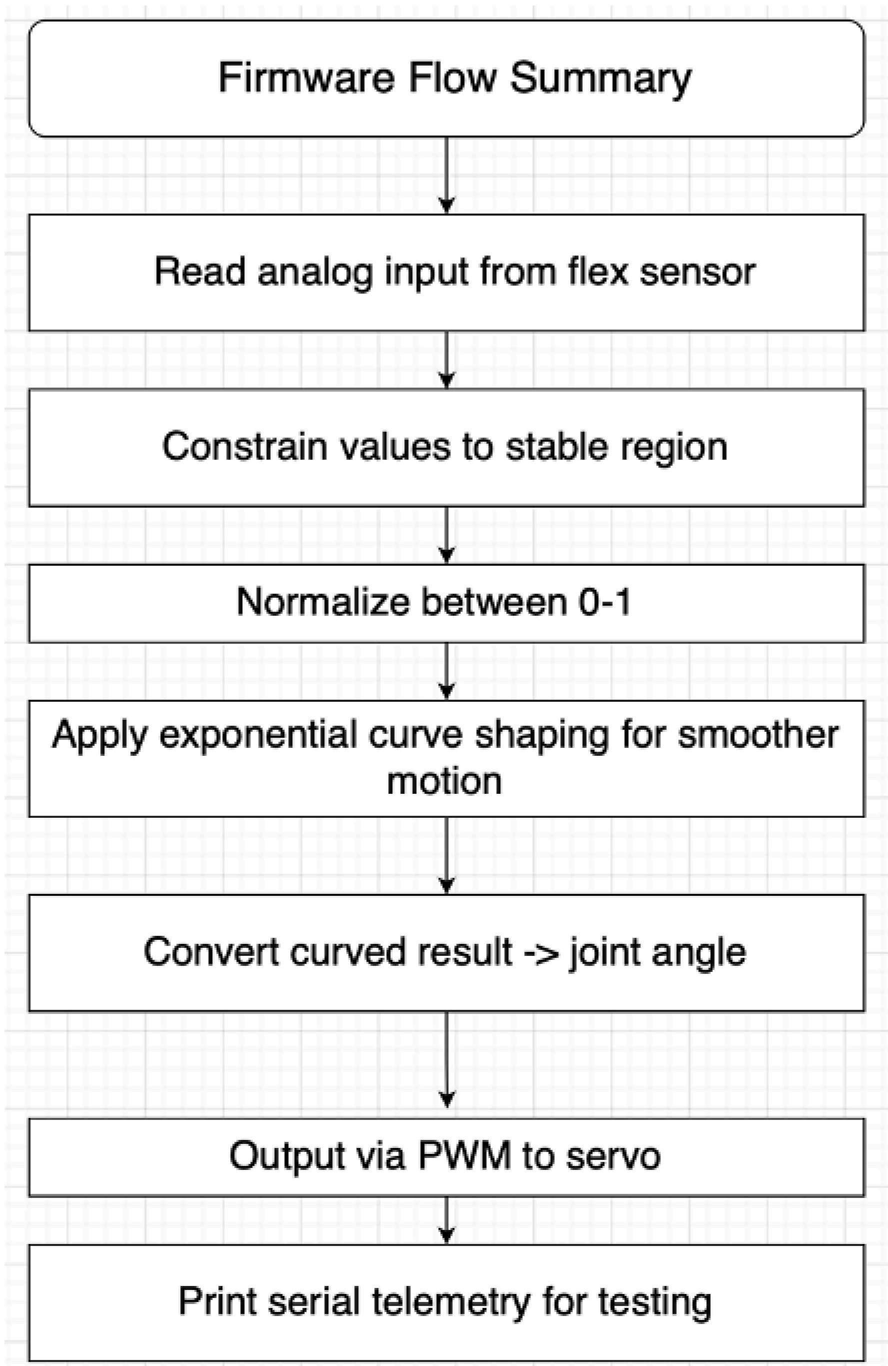


Figure 7.5.1 Updated Firmware Flowchart

C# Software Flow

1. Receive shoulder + elbow angles (or user input)
2. Compute FK → Determine elbow + hand coordinates
3. Render joints + segments graphically
4. If dragging hand → run IK solver, update $\theta 1$ + $\theta 2$
5. Display angles on screen

This allows real-time control and testing before hardware is finalized.

7.6 Preliminary Results & Anticipated Challenges

Current Achievements

- Flex sensor input successfully detected and processed
- Nonlinear curve mapping improves control resolution
- Servo joints respond predictably to flex variation
- Addition of shoulder degree of freedom confirmed feasible
- C# application successfully simulates motion via FK + IK
- Joint visualization and dragging interface functioning

Anticipated Development Challenges

Because full hardware implementation is planned for next semester, these challenges are projected rather than observed:

1. Signal Noise / Drift

Flex sensors may produce jitter requiring smoothing or filtering.

2. Servo Torque Limitations

Shoulder lifting may require higher-torque motors.

3. Mechanical Stability

Precision hinges must withstand repeated load cycles.

4. Real-Time Synchronization

Wireless transmission and joint control must remain responsive.

5. IK Singularities / Edge Cases

Points outside the reachable radius require safety handling.

Planned Next-Semester Improvements

- Add additional sensing (multiple flex sensors or IMU)
- Add PID control loop for smooth joint motion
- Introduce battery-powered independent operation
- Improve mechanical shoulder assembly
- Perform full hardware testing + calibration

7.7 System Integration and Testing Plan (Next Semester)

7.7.1. Overview

The GRIP software stack translates human finger motion from a wearable glove into real-time robotic hand/elbow movement. In Senior Design 1 (SD1), the focus is on developing firmware to capture flex-sensor data, mapping this data to elbow/hand response, and transmitting it to the robotic assembly over wireless communication.

Shoulder control is not implemented in SD-1; it is modeled only in software and is planned for integration in Senior Design 2

7.7.2. Current SD-1 Functional Scope

The following functions are fully included in the SD-1 software effort:

- Flex sensor reading (up to 5 channels)
- ADC acquisition with smoothing
- Value mapping → servo angle
- Wireless transmission via NRF24L01
- Robotic hand + elbow actuation
- Serial output for calibration
- Desktop visualization of joint angles

The following are out of scope for SD-1 but planned for SD-2:

- Physical shoulder actuation
- PID-based servo stabilization
- Multi-sensor fusion (e.g., IMU integration)

This staged rollout ensures that core motion mapping is validated before expanding degrees of freedom.

7.7.3. Software System Architecture

The SD-1 software architecture is divided into three layers:

1) Data Acquisition (Glove MCU)

- Flex sensor signals enter the Arduino Nano's ADC ports
- Sensor's output ~0–1023 (10-bit)
- Baseline calibration normalizes finger bend differences
- Smoothing via moving average / low-pass filtering

2) Data Processing + Mapping (Glove MCU)

- Normalize ADC → 0.0–1.0
- Apply nonlinear exponential curve (power ≈ 4–5) for precision
- Convert mapped value to joint-angle range (0–180°)

3) Wireless Transmission (NRF24L01)

- Packet format includes:
    - finger bend values
    - Packet ID
    - CRC checksum
- Transmission rate: ~50–100 Hz

In SD-1, packets control only the elbow + hand.

Shoulder parameters are reserved but unused until SD-2.

7.7.4. Receiver-Side Interpretation

On the robotic-hand MCU:

- NRF24L01 receives packet
- Data validated using CRC + sequence ID
- Mapped joint angles forwarded to servo driver
- Debug output printed for calibration
- If packets fail or time out (>500 ms), the arm returns to a neutral safe position.

PID control will be added in SD-2; SD-1 uses direct mapping.

7.7.5. Firmware Flow

Glove Firmware:

- Read ADC values (F1–F5)
- Filter → moving average
- Normalize to 0–1
- Apply exponential shaping curve
- Map to angle range
- Pack & transmit
- Output debug serial stream
- Hand Firmware
- Receive packet

- Validate structure + CRC
- Convert angle fields → PWM signals
- Drive servo motors (elbow + hand)
- Idle/recover on timeout

7.7.6 C# Visualization Module

A custom desktop tool was developed to assist with debugging and visualization.

Current SD-1 features:

- Receives serial data from glove/receiver
- Displays elbow + hand joint angle values
- Draws a 2-link virtual arm (FK-based)
- Allows manual dragging of end-effector (IK testing)

Future (SD-2):

- Integrate true shoulder degree of freedom into the FK/IK model

7.7.7 Data Structures & Packet Format

The NRF24L01 packet contains:

Byte	**Meaning**
0	Finger 1
1	Finger 2
2	Finger 3
3	Finger 4
4	Finger 5
5	Packet ID
6	CRC

Shoulder joint fields are reserved for SD-2.

7.7.8 Safety Features

- Value clipping to keep joint motion within physical limits
- Servo command throttling to avoid sudden jumps
- Neutral pose on invalid / missing packets
- Debug output for monitoring

Full PID control and sensor redundancy will be added in SD-2.

7.7.9 Testing and Verification

Table 7.7.1 Verification options

Method	**Result**
Serial sensor monitoring	Valid flex readings
Mapping curve evaluation	Improved low-bend resolution
Wireless latency	Averaged < 100 ms
Servo control	Stable hand + elbow movement
Visualization	Accurate FK response

Shoulder motion is not tested in SD-1 — simulation only.

Next semester testing will include BLE reliability, torque testing, extended motion testing, and PID tuning.

7.7.10. SD-2 Planned Expansion (Shoulder Integration)

- Add shoulder servo motor
- Add PID control loops
- Improve visualization to 3-DOF
- Power optimization + battery integration
- Add IMU for orientation compensation
- This expansion will complete the GRIP system's upper-limb mobility.

7.7.11. Summary

SD-1 successfully establishes the core software framework required for real-time replication of hand/elbow movement via flex-sensor input.

The system currently supports:

- ADC acquisition
- Mapping + shaping curve
- Wireless packet transmission
- Elbow + hand actuation
- Real-time visualization

SD-2 will extend this core to include shoulder actuation, PID feedback control, and advanced sensor fusion.

The foundation built during SD-1 ensures that software scalability, modularity, and performance will support future development.

7.8 Summary and Conclusion

7.8.1. Summary of Software Design

The GRIP project's software system serves as the foundation that links human hand motion with robotic arm replication. Throughout the first semester, the software framework was developed and documented across multiple stages — including sensor data acquisition, calibration algorithms, SPI communication, and servo control logic. Each software module was designed with modularity to ensure independent testing and future scalability.

The architecture's three-tier design (Data Acquisition, Processing & Mapping, and Communication & Control) ensures scalability, easy maintenance, and real-time responsiveness. A C#FK/IK visualization tool provides early-stage verification before physical integration.

7.8.2. Accomplishments to Date

• Completed detailed software architecture and data flow documentation.

• Developed firmware structure for glove sensor acquisition and data transmission.

• Implemented mapping and calibration functions to convert sensor data into servo motion.

• Designed wireless communication packet format with error-checking and fail-safe logic.

• Created flowcharts, diagrams, and testing documentation for advisor review.

7.8.3. Next Semester Objectives

• Integrate glove and robotic arm firmware via NRF24L0.1 + 2.4 GHz radio (SPI controlled)

• Conduct full calibration, communication reliability, and latency testing.

• Evaluate servo accuracy, data consistency, and overall response time.

• Prepare final demonstration and comprehensive technical report.

7.8.4. Anticipated Impact

Once complete, the GRIP project will showcase real-time robotic motion replication based on wearable sensor input. This system could serve as a foundation for prosthetic research, teleoperation systems, and assistive robotics. The software component bridges the gap between human motion and machine behavior with high precision and reliability.

7.8.5. Personal Reflection

Working on the GRIP software component has provided valuable experience in embedded programming, real-time communication, and control system integration. This project strengthened my understanding of how firmware, hardware, and algorithms work together to create intelligent mechatronic systems that respond intuitively to human input.

7.8.6. Conclusion

The software framework built this semester establishes a solid foundation for completing the GRIP system in the next phase. Through disciplined development, collaborative teamwork, and iterative testing, the project is positioned to achieve its goal of creating an accurate, responsive, and user-friendly glove-controlled robotic arm.

Chapter 10 Project Management and Communication Plan

10.1. Overview

Effective project management and communication are essential for coordinating the GRIP team's efforts across software, electrical, and mechanical components. This plan ensures timely collaboration, clear documentation, and consistent progress throughout the second semester.

10.2. Management Approach

The GRIP project will follow an **iterative development model** — a cycle of design, integration, testing, and improvement. Each phase will conclude with internal reviews and actionable feedback to maintain alignment between subsystems.

Key Management Principles:

• Weekly Team Meetings: Track progress, resolve technical challenges, and assign tasks.
• Version Control: All firmware and documentation updates are maintained via GitHub.
• Task Ownership: Each subsystem lead is responsible for updating their assigned area.
• Progress Logs: Team members record work hours, milestones, and test outcomes.

10.3 Budget and Financing

The budget for this project will be limited to $1000. This is a self imposed limit agreed on by the group but as the project progresses the actual cost will become clear. If we are under budget some extra funds can be used to mitigate errors and have some back up components. There are costs that are unknown such as any changes that might need to be made to the component and PCB costs. Smaller components such as the LEDs and buttons will also need to be sourced. The extra left in the budget will be useful for those situations. This budget is a draft and is subject to change.

Table 10.3.1 Expenses

Component	Quantity	Unit Cost	Total
Controller	2	$10.00	$20.00
Flex Sensors	7	$9.42	$65.94
Servo Motors	10	$3.62	$36.20
Transceiver	1	$7.68	$7.68
Receiver	1	$4.95	$4.95
Power Source	1	$7.95	$7.95
Frame and 3-D Printed Components	1	$400.00	$400.00
Force Sensor	1	$4.95	$4.95

10.4. Team Communication Methods

To maintain consistent and transparent collaboration, the team will use the following tools:

- **Discord:** Daily updates, quick coordination, and informal communication.
- **Email:** Weekly summaries and confirmation of key decisions.
- **GitHub:** Issue tracking, commits, and collaborative firmware development.
- **Google Drive:** Shared documentation, test results, and final reports.

10.5. Project Documentation

All materials will be organized in shared cloud storage to maintain accessibility and accountability:

- Engineering Logs: Record calibration data, test results, and modifications.
- Software Repository: Contains source code, mapping functions, and build notes.
- Testing Sheets: Track performance metrics and anomalies by date.
- Reports & Diagrams: Maintain final formatted versions for presentations and submission.

Meeting Schedule

Table 10.5.1 Meeting strategies

Meeting Type	Frequency	Purpose	Participants
Team Check-In	Weekly	Review milestones and technical progress	Entire team
Advisor Meeting	Biweekly	Receive feedback and confirm progress	Faculty advisor + leads
Testing Session	As needed	Perform subsystem and integration testing	Assigned teams
Final Review	End of semester	Demonstration results and present documentation	All members + faculty

10.6. Risk Management and Contingency

Potential risks will be mitigated through proactive planning and communication:

- Component Delays: Backup parts will be ordered early.
- Firmware Instability: Maintain version backups and frequent Git commits.
- Communication Issues: Escalate unresolved challenges in advisor meetings.
- Hardware Damage: Keep spare IMU sensors, servos, and power boards on hand.

10.6. Expected Outcome

By following this management and communication framework, the team will ensure consistent coordination between subsystems and minimize project delays. This structure provides a clear path to a successful demonstration of the GRIP system next semester.

10.7. Summary of Software Design

The GRIP project's software system serves as the foundation that links human hand motion with robotic arm replication. Throughout the first semester, the software framework was developed and documented across multiple stages — including sensor data acquisition, calibration algorithms, SPI communication, and servo control logic. Each software module was designed with modularity to ensure independent testing and future scalability.

The architecture's three-tier design (Data Acquisition, Processing & Mapping, and Communication & Control) ensures scalability, easy maintenance, and real-time responsiveness. A C#FK/IK visualization tool provides early-stage verification before physical integration.

10.8. Accomplishments to Date

- Completed detailed software architecture and data flow documentation.
- Developed firmware structure for glove sensor acquisition and data transmission.
- Implemented mapping and calibration functions to convert sensor data into servo motion.
- Designed wireless communication packet format with error-checking and fail-safe logic.
- Created flowcharts, diagrams, and testing documentation for advisor review.

10.9. Next Semester Objectives

- Integrate glove and robotic arm firmware via NRF24L0.1 + 2.4 GHz radio (SPI controlled)
- Conduct full calibration, communication reliability, and latency testing.
- Evaluate servo accuracy, data consistency, and overall response time.
- Prepare final demonstration and comprehensive technical report.

10.10. Anticipated Impact

Once complete, the GRIP project will showcase real-time robotic motion replication based on wearable sensor input. This system could serve as a foundation for prosthetic research, teleoperation systems, and assistive robotics. The software component bridges the gap between human motion and machine behavior with high precision and reliability.

10.11. Personal Reflection

Working on the GRIP software component has provided valuable experience in embedded programming, real-time communication, and control system integration. This project strengthened my understanding of how firmware, hardware, and algorithms work together to create intelligent mechatronic systems that respond intuitively to human input.

10.12. Conclusion

The software framework built this semester establishes a solid foundation for completing the GRIP system in the next phase. Through disciplined development, collaborative teamwork, and iterative testing, the project is positioned to achieve its goal of creating an accurate, responsive, and user-friendly glove-controlled robotic arm. A multifaceted tool that can accomplish feats of engineering and meet standards.

Appendicies

Appendix A - Reference

[1] Arduino, “Nano | Arduino Documentation.” Accessed: Nov. 7, 2025. [Online]. Available: https://docs.arduino.cc/hardware/nano

[2] Microchip, “ATmega328/P Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://docs.arduino.cc/resources/datasheets/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

[3] Nordic Semiconductor, “nRF24L01+ Product Specification v1.0.” Accessed: Nov. 7, 2025. [Online]. Available: https://docs.nordicsemi.com/bundle/nRF24L01P_PS_v1.0/resource/nRF24L01P_PS_v1.0.pdf

[4] SparkFun, “nRF24L01+ Product Specification v1.0 (mirror).” Accessed: Nov. 7, 2025. [Online]. Available: https://cdn.sparkfun.com/assets/3/d/8/5/1/nRF24L01P_Product_Specification_1_0.pdf

[5] TowerPro/HandsOnTech, “MG996R Metal Gear Servo Motor—User Guide.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.handsontec.com/dataspecs/motor_fan/MG996R.pdf

[6] TowerPro (mirror), “MG996R High Torque Servo Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf

[7] Nexperia, “BSS138BK: 60 V, 360 mA N-channel Trench MOSFET—Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://assets.nexperia.com/documents/data-sheet/BSS138BK.pdf

[8] Nexperia, “2N7002: 60 V, 300 mA N-channel Trench MOSFET—Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://assets.nexperia.com/documents/data-sheet/2N7002.pdf

[9] onsemi, “NDS7002A / 2N7002—N-Channel Enhancement MOSFET—Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.onsemi.com/download/data-sheet/pdf/nds7002a-d.pdf

[10] Alpha & Omega Semiconductor, “AO3400A: 30 V N-Channel MOSFET—Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.aosmd.com/res/data_sheets/AO3400A.pdf

[11] onsemi, “BSS84: P-Channel MOSFET—Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.onsemi.com/download/data-sheet/pdf/bss84-d.pdf

[12] SparkFun, “Flex Sensor—Data Sheet (general resistive bend sensor).” Accessed: Nov. 7, 2025. [Online]. Available: https://cdn.sparkfun.com/assets/8/e/7/a/0/flex22.pdf
[13] Probots, “ZD10-100 Resistive Film Flex Sensor—Product Page (specs).” Accessed: Nov. 7, 2025. [Online]. Available: https://probots.co.in/0-500g-flex-sensor-resistive-film-pressure-sensor-zd10-100.html
[14] Texas Instruments, “TPS54331: 3-A, 28-V Input Step-Down Converter—Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps54331.pdf
[15] Texas Instruments, “TPS6216x: 3-V to 17-V, 1-A Step-Down Converters—Datasheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps62162.pdf
[16] Microchip, “MCP1700 Low-Quiescent LDO—Data Sheet.” Accessed: Nov. 7, 2025. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP1700-Low-Quiescent-Current-LDO-20001826E.pdf
[17] Espressif, “ESP-NOW—ESP-IDF Programming Guide.” Accessed: Nov. 7, 2025. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html
[18] Bluetooth SIG, “Bluetooth Core Specification v5.2 (overview).” Accessed: Nov. 7, 2025. [Online]. Available: https://www.bluetooth.com/bluetooth-resources/bluetooth-core-5-2-feature-overview/
[19] Digi, “XBee RR 802.15.4 RF Module—User Guide.” Accessed: Nov. 7, 2025. [Online]. Available: https://docs.digi.com/resources/documentation/digidocs/PDFs/90002522.pdf
[20] Semtech, “SX1276/77/78/79 LoRa® Transceivers—Product Page.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-connect/sx1276
[21] IEC, “IEC 61326-1:2020—EMC Requirements for Electrical Equipment for Measurement, Control and Laboratory Use.” Accessed: Nov. 7, 2025. [Online]. Available: https://webstore.iec.ch/en/publication/62
[22] ISO, “ISO 10993-1:2018—Biological Evaluation of Medical Devices—Part 1.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.iso.org/standard/68936.html
[23] IEC, “IEC 62366-1:2015/AMD1:2020—Application of Usability Engineering to Medical Devices.” Accessed: Nov. 7, 2025. [Online]. Available: https://webstore.iec.ch/en/publication/59980

[24] Castle Creations, “CC BEC—10 A Switching Regulator (spec page).” Accessed: Nov. 7, 2025. [Online]. Available: https://www.castlecreations.com/en/cc-bec-010-0004-00

[25] International Electrotechnical Commission (IEC), “IEC — International Standards and Conformity Assessment for Electrical, Electronic and Related Technologies.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.iec.ch/

[26] International Organization for Standardization (ISO), “ISO — International Organization for Standardization.” Accessed: Nov. 7, 2025. [Online]. Available: https://www.iso.org/

[27] IEEE Standards Association, “IEEE SA — Standards.” Accessed: Nov. 7, 2025. [Online]. Available: https://standards.ieee.org/