Virtual Reality Glove with Thermal, Haptic, and Force Feedback

A Senior Design (EET4166C) Project by:
Brad Alvarez and Yeshaya Zweig

1. Introduction

Augmented reality and virtual reality are reality technologies that either enhance or replace a real-life environment with a simulated one. Augmented reality (AR) augments your surroundings by adding digital elements to a live view, often by using the camera on a smartphone. Virtual reality (VR) is a completely immersive experience that replaces a real-life environment with a simulated one, by using a VR headset worn over the eyes.

In AR, a virtual environment is designed to coexist with the real environment, with the goal of being informative and providing additional data about the real world. For example, industrial AR apps could offer instant troubleshooting information when a handset is aimed at a piece of failing equipment. VR encompasses a complete environmental simulation that replaces the user's world with an entirely virtual world. This can put the user in an entirely simulated environment, like walking on the moon, or operating a tank.

Both VR and AR have been used in education, medicine, gaming, and entertainment. In medicine, AR has been providing surgeons with visual information of surgical anatomy throughout procedures by utilizing AR and superimposing a 3D model of the patient's organs onto the lenses. Many tools have been made to work alongside VR technologies to improve training programs and increase the facilitation of information. One of the most popular aspects many people consider when they consider VR is head-mounted displays (HMDs.) The industry has developed many types, including mobile, wired, and wireless. Companies such as Oculus and HTC produce these solutions and are two of the most popular companies in the market today.

There are two crucial feedback mechanisms required for AR/ VR systems. One is visual feedback, and the other is sensory feedback. HMDs receive signals from the embedded computer within the headset for visual feedback. Visual feedback is the heart of the user's experience and as the headset tracks the position of the head in 3D space, it allows for fully convincing visual feedback, as the user moves throughout the virtual environment. Another relevant tool manufactured by the same company is the controller many of these HMDs come with. These are used in conjunction with the display to allow the user to interact with the game or environment. While most popular systems use handheld controllers, gloves that provide better immersion and Haptic feedback are sold on the market today. Haptic feedback gloves are very similar to the controller, except they provide wearable force feedback with multi-finger interaction.

What we propose is a VR and AR glove for the uses mentioned above. What makes our gloves unique however, is the features our glove possesses. Right now, the best VR and AR gloves on the market either have force feedback, or some temperature response, or basic haptic motor feedback. Our glove has thermal response, with heating and cooling spread across the palm. This immerses the user deeply when a can from the fridge is cold or petting your virtual dog feels warm. Additionally, our product has force feedback. Force feedback is where the hand is restricted from closing by some external force when it grabs a virtual object. This acts as if the hand is holding the virtual object in real life, and the immersion is increased. Lastly, our glove has haptic feedback to indicate when an object is interacted with, in this case a small vibration motor on each fingertip. This is a famous and long since used way to convey user to object interaction in most feedback devices.

These features are not available together in any commercial product. Right now, the only commercially available thermal response device is called the TouchDIVER, where only the tips of three fingers feel the temperature change, at a whopping cost of 8,000 dollars for a pair. With force feedback, there are a couple products that have only this feature, however the cost, about 5,000 dollars per pair, is daunting for the commercial audience. Our glove not only has these features but can be produced for a cost of about 350 dollars for a pair (bought with commercially available products). With current top of the line headsets costing between 500 and 3,500 for a headset only, this price is very competitive for the next level immersion some use cases require.

2. Motivation

VR and AR have impacted many industries, such as policing, medical, and military. Since 2019, the NYPD has been using VR to train for active shootings and real-life scenarios. In medicine, AR has been providing surgeons with visual information of surgical anatomy throughout procedures. For the military, it helps for training exercises that are too rare, too expensive, or too dangerous to be done in real life. Training simulation in the military field fosters the combat skills of small-scale units or single soldiers by simulating actual vehicles, soldiers, and combat environment to better prepare their personnel. There are also applications for immersive gaming and education and travel uses, which there is an even larger market for.

The global Augmented Reality and Virtual Reality market size accounted for 38.3 billion in 2022 and is estimated to achieve a market size of 394.8 billion by 2030. Additionally, over 171 million users use these devices at least once a month in the US alone. As we can see, there are a lot of use cases for this technology, and there is a demand for accurate replicability of the real world through a virtual one. With our glove, all police, healthcare and military training will be far more accurate when they can feel the scalpel or touch the hot car or learn how to react in cold climates, when they can both see and feel the training environment. Besides usefulness, having greater immersion in any experience whether in a game or learning about an event is invaluable to the user's perception of the environment. This glove can heighten that immersion considerably to benefit all of these fields.

3. How we decided/ranked our product

3.1. **Glove Ranking**

1. **Haptic, Thermal, & Force Feedback VR Glove**: This version has DC vibration motors for haptic feedback, Peltier pads for thermal feedback, and servos to restrict hand movement for force feedback.
    - Slightly more complex
    - More immersive
    - Moderate cost
    - Moderate size/weight
2. **Haptic Feedback & Thermal Feedback VR Glove:** This version has DC vibration motors for haptic feedback, and servos to restrict hand movement for force feedback.
    - Moderate complexity
    - Less immersive
    - Moderate cost
    - Moderate size/weight
3. **Force Feedback VR Glove**: This version has servos for force feedback, in addition to hardware hand tracking.
    - Low complexity
    - Not very immersive
    - Low cost
    - Smaller/Lighter size/weight
4. **Advanced VR Glove**: This version has DC vibration motors for haptic feedback, Peltier pads for thermal feedback, servos to restrict hand movement for force feedback, and additional features such as touch sensors for finer tactile feedback, and biometric sensors for tracking physiological responses (like heart rate or skin conductance).
    - Most complex
    - Very immersive
    - High cost
    - Bigger/Higher size/weight

3.2. **Hand Tracking Ranking**

1. **Hardware hand tracking:** Adds hardware tracking, requires glove to send data back to PC and requires hand tracking hardware. (Flex sensors or hall effect sensors)
    - More difficult implementation (hardware and software)
    - Higher cost
    - More accurate

2. **Headset hand tracking:** Utilizes built-in hand tracking provided by the VR headset.
    - Simplistic implementation
    - Lower cost
    - Not very accurate with gloves
    - Made for bare hands

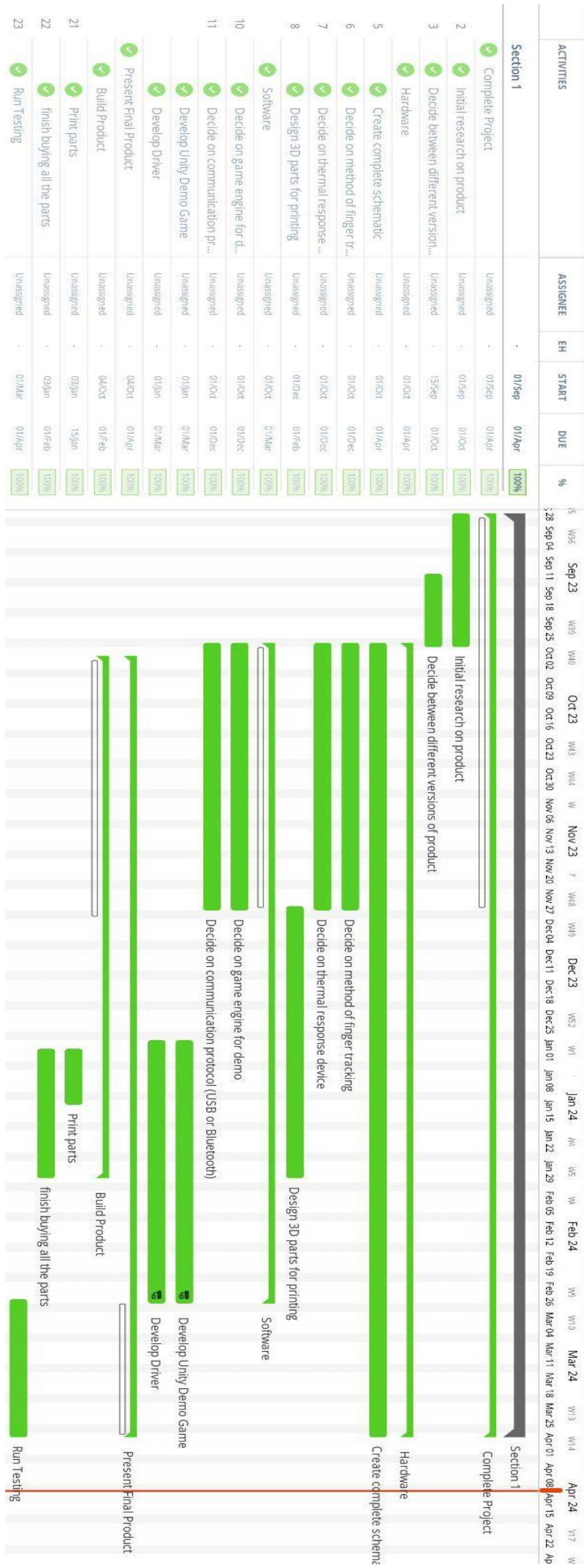
3.3. **Glove Design Decision Table**

Working Criteria	Points Available	#1	#2	#3	#4
Complexity	20	15	17	20	5
Immersiveness	35	30	25	15	35
Customer appeal	20	15	10	5	20
Size, weight	10	5	7	10	2
Cost	15	10	13	15	1
Totals	**100**	**75**	**72**	**65**	**63**

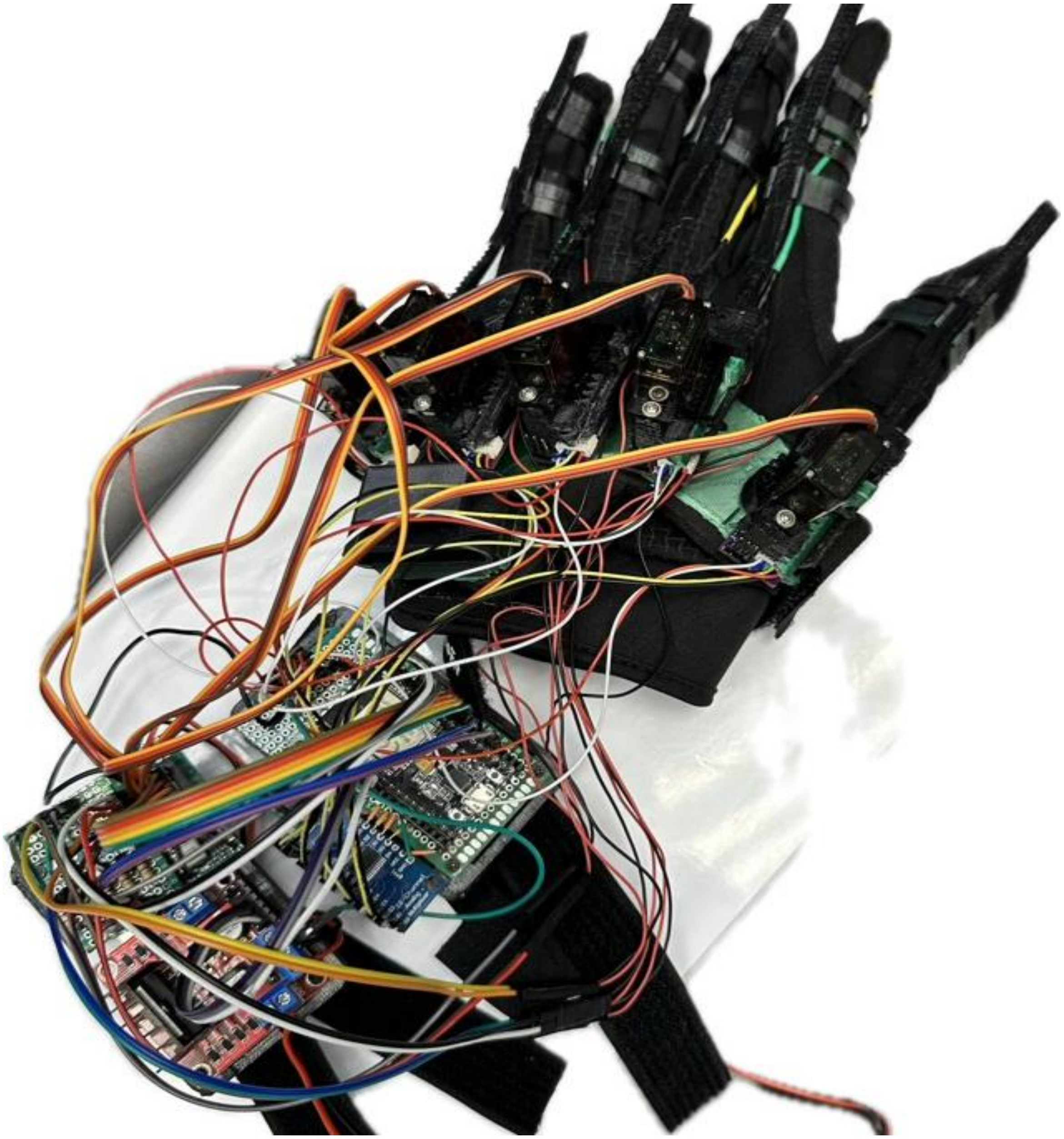
3.4. **Product Choice**

After considering the options, we chose the Haptic, Thermal, & Force Feedback VR Glove with hall effect sensor finger tracking. This version of the glove provides the most immersive and realistic experience for the user for its price and build complexity, as it can simulate different sensations such as vibration, temperature, and resistance. It also has a moderate cost and a slightly more complex design than the other options, which makes it practical for this project. We decided to use the hardware hand tracking, as it is the only way to get finger and hand tracking working with usable accuracy. Headset hand tracking is designed for bare hands, once we add all the VR glove hardware, headset hand tracking becomes very difficult and inaccurate.

4. Gantt chart



5. Product Image



6. Solution

6.1. How to use product

This product has the benefit of being very natural for people to use, as the user puts the glove on like a normal glove. After the glove is on the hand, the finger rings that keep the fingers attached to the racks are slid onto each finger and adjusted until a tight fit is formed. Then the user connects the microcontroller powering the glove to the computer via a wired USB connection. The next step is to open the demo that we made and ensure the virtual hand matches the finger positions of the user's actual hand. Once connected and displaying properly, the glove is functional and can be demonstrated.

6.2. Hardware

6.2.1. List of Parts

1. 3D printed parts
2. Flexible thermoelectric devices
3. Tiny magnets
4. Ball bearings
5. Multiplexers
6. Hall effect sensors
7. L298N motor driver
8. Bicycle gloves
9. Foam roll
10. Esp32 boards
11. Straps
12. Power blocks
13. Servos
14. Power supply 3.3v and 5v
15. Flexible wires
16. Mini vibration motors
17. 3.3v regulator

6.2.2. Explanation of Key Parts

Flexible Peltier Thermoelectric Heaters

These heaters use the Peltier effect, which is an electric heater with no circulating fluid or moving parts. The Peltier effect creates a heat flux at the junction of two different types of materials. When a current is passed one way through these devices one side heats and when it is reversed that same side cools and the reverse heats. The device we used is a flexible version, with high accuracy, giving us the thermal response we are looking for.

L298N Motor Controller

The L298N is a dual H-Bridge motor controller which allows speed and direction control of two DC motors at the same time. For our use, we used it to control the Temperature of the Peltier module by utilizing the controller's PWM input, as well as the heating/cooling. Since we need to control the temperature at a variable speed as well as switch the polarity of the current, we need to use a decent motor controller. We switch the polarity of the Peltier device the same way you would switch the direction of a regular DC motor, by switching the polarity through two outputs on the controller.

ESP32 Microcontroller

The Microcontroller we used for this product was the ESP32 development board. The ESP32 is a low-cost, low-power microcontroller that is controlling all of the actions of the glove, i.e. the haptic, thermal, and force feedback and calculating our finger position data. We decided to use this because of its dual core capabilities, which allows us to run two separate threads, one for calculating finger curl data and another for receiving haptic, thermal, and force feedback packets from our driver over serial.

Hall Effect Sensors

Hall effect sensor is a type of sensor which detects the presence and magnitude of a magnetic field using the Hall effect. The output voltage of a Hall sensor is directly proportional to the strength of the field. Using this output voltage, we can accurately measure the individual finger movements by connecting a gear with magnets to finger tracks, which are attached to the finger. Each finger has two hall effect sensors (sin and cosine). Utilizing both sensors allows us to accurately determine the rotation of the gear by mixing the signals. We do this by computing the arctangent quotient of the sin and cosine sensor values.

$$\theta = arctan2(y, x)$$

Arctan2 calculates the angle (θ) from the positive x-axis to a point. Plugging in our *sin* value from our sensor as *y* and our *cosine* value as *x* gives us the angle in radians between the positive x-axis and the point given by the coordinates. We then check the sign of the angle (crossing the 0° or 180° mark) and whether the *sine* value is greater than the *cosine* value to determine if a full rotation has occurred. We keep track and store the number of full rotations in a *totalRotations* variable (positive or negative). We then calculate the total angle by adding the raw angle to the total number of full rotations (*totalRotations*) times 2π.

$$angleRaw = arctan2(sinValue, cosValue)$$
$$totalAngle = angleRaw + (totalRotations \times 2\pi)$$

This leaves us with the total angle in radians of our finger gear from our two hall effect sensors which we can use later to get our finger curl percentage value.

DC Vibration Motors

We use small dc vibration motors to provide Haptic feedback. Haptic feedback (often shortened to just haptics) is controlled vibrations at set frequencies and intervals to provide a sensation representative of an in-game action. Examples include a video game controller vibrating and a smartphone screen providing a button-clicking sensation. We use vibration motors to vibrate when we interact with a virtual object to fully immerse the user by simulating the feeling of a physical object.

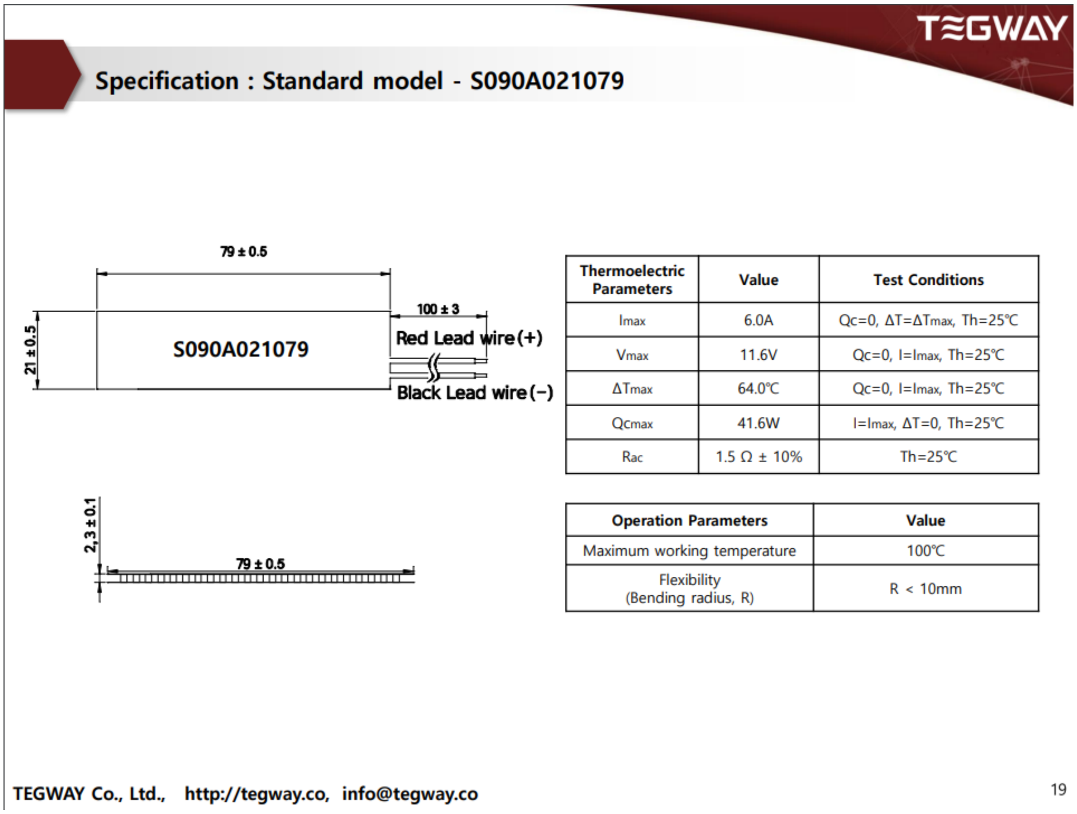
Micro Servo Motor MG90S

The Micro Servo we decided on (MG90S) is great for applications in low-cost robotics and automation. The MG90S can be powered directly from any 5.0V microcontroller. We use the motor to move a plastic tab that prevents the finger racks from fully extending precisely to the distance that we want the fingers closed. For different sized virtual objects, the servo motors are manipulated back and forth, maintaining a precise amount of flexing of the finger racks, preventing the fingers from fully extending.

6.2.3. Specifications of Key Parts

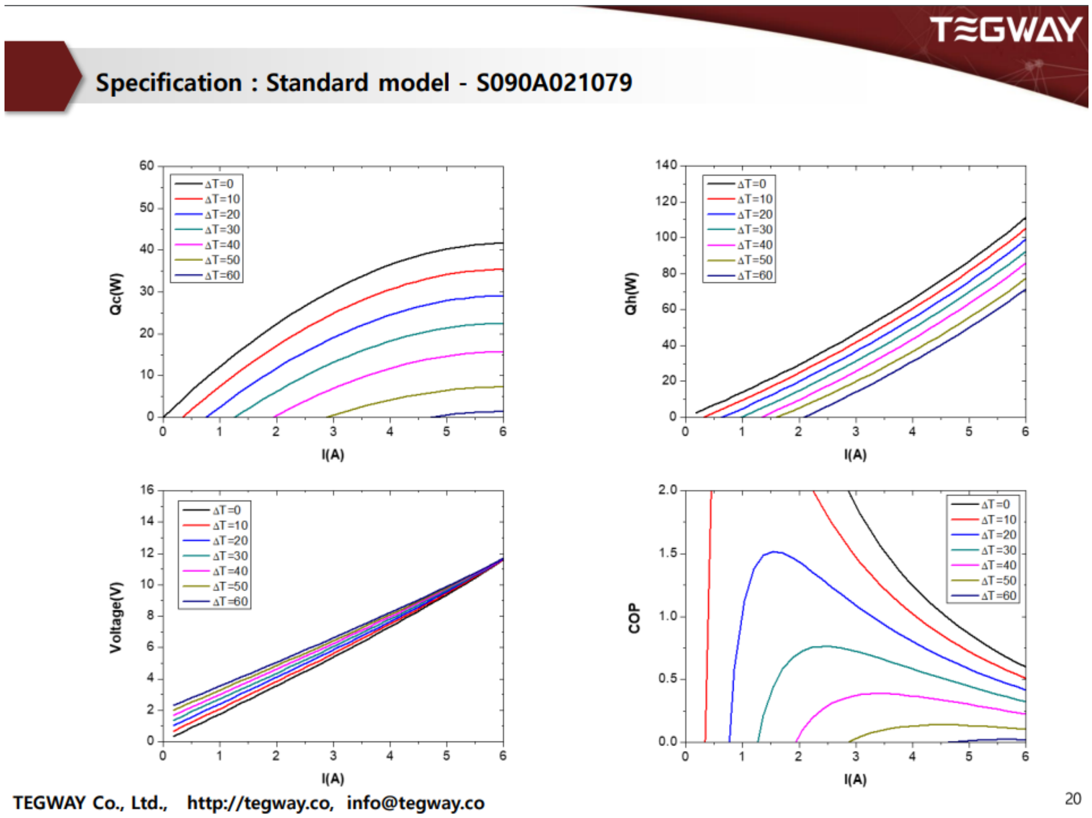
Flexible Peltier Thermoelectric Heaters

We used a TEGWAY thermoelectric Peltier device, as they are the only commercially available flexible Peltier modules in the world. Below are the specifications of the model we used.



Thermoelectric Parameters	Value	Test Conditions
I_{max}	6.0A	$Q_c=0$, $\Delta T=\Delta T_{max}$, $T_h=25$℃
V_{max}	11.6V	$Q_c=0$, $I=I_{max}$, $T_h=25$℃
ΔT_{max}	64.0℃	$Q_c=0$, $I=I_{max}$, $T_h=25$℃
Q_{cmax}	41.6W	$I=I_{max}$, $\Delta T=0$, $T_h=25$℃
R_{ac}	1.5 Ω ± 10%	$T_h=25$℃

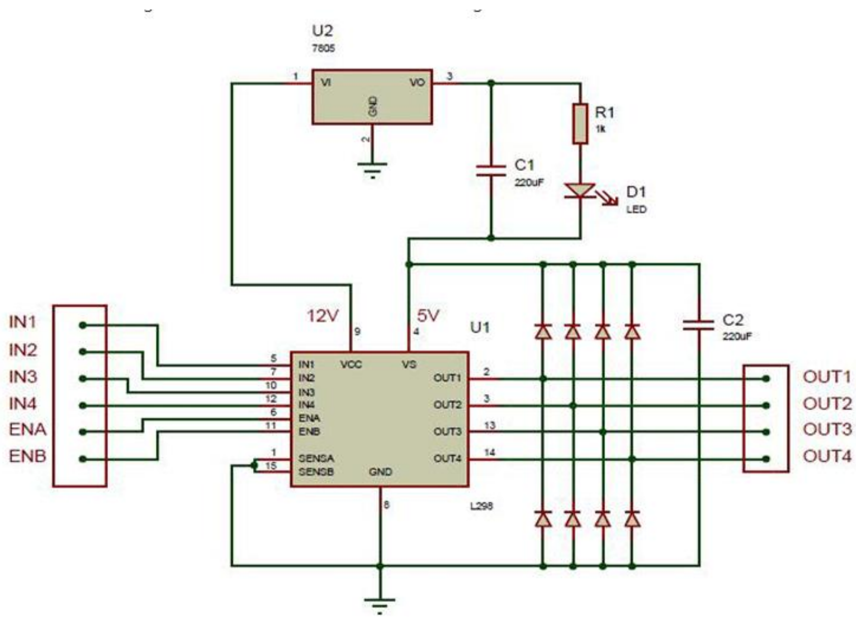
Operation Parameters	Value
Maximum working temperature	100℃
Flexibility (Bending radius, R)	R < 10mm



L298N Motor Controller

We decided to use a common L298N motor controller because it was one of the only models of motor controllers that could handle our current without being too impractically large.

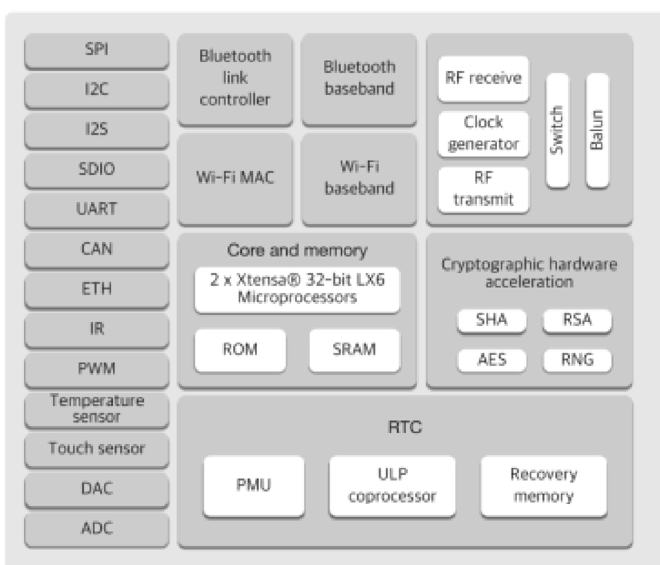
- Driver Model: L298N 2A
- Driver Chip: Double H Bridge L298N
- Motor Supply Voltage (Maximum): 46V
- Motor Supply Current (Maximum): 2A
- Logic Voltage: 5V
- Driver Voltage: 5-35V
- Driver Current:2A
- Logical Current:0-36mA
- Maximum Power (W): 25W
- Current Sense for each motor
- Heatsink for better performance
- Power-On LED indicator

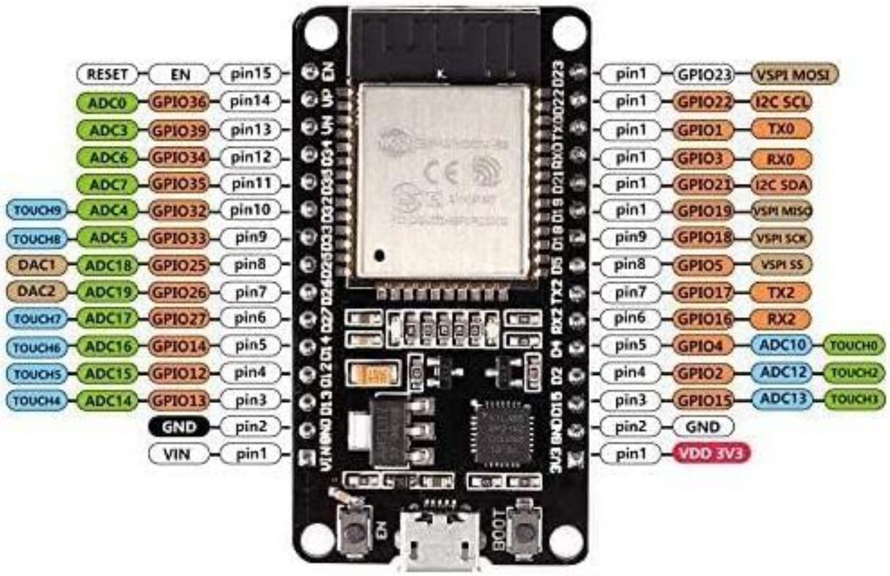


ESP32 Microcontroller

The Microcontroller we used for this product was the ESP-WROOM-32 ESP32 ESP-32S Development Board 2.4GHz Dual-Mode Wi-Fi + Bluetooth Dual Cores Microcontroller Processor Integrated with Antenna RF AMP Filter AP STA. Below are the specifications of the model we used.

1.5 Block Diagram

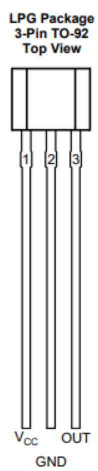




Hall Effect Sensors

Standard hall effect sensors, below are the specifications of the model we used.

- Principle of operation : Hall IC
- Signal output: Square wave digital output
- Operating voltage: 5v
- Operating current: 10 mA
- Operating temperature: -30◦C to +125◦C

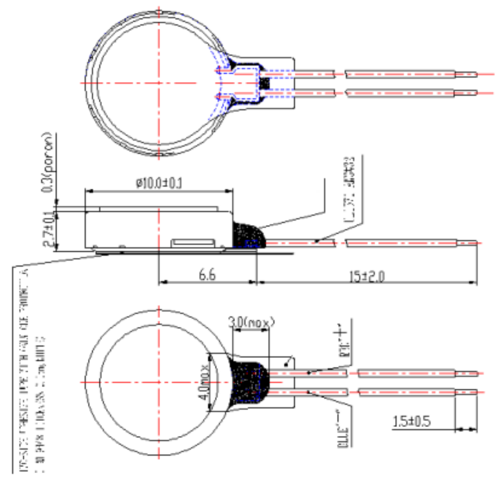


DC Vibration Motors

Standard vibration motors, below are the specifications of the model we used.

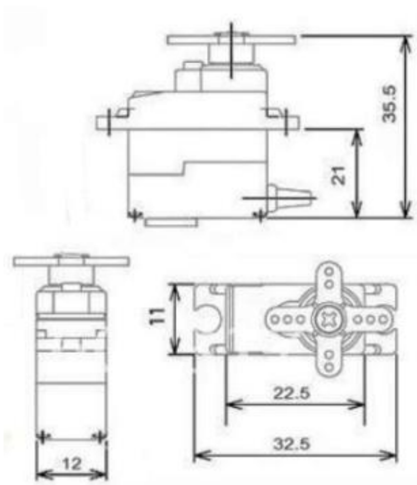
- Rated voltage: 3V;

- Rated current: 80mA max
- Rated speed: 12000 RPM
- Vibration motor body size: 10 x 2.7mm / 0.39" x 0.22" (D * T)
- Line length: 100mm/3.93"



Micro Servo Motor MG90S

Standard micro servo motor, below are the specifications of the model we used.



- Product Model : MG90S
- Maximum Angle :180 degrees

- Operating Voltage : 4.8V
- Operating Speed : 0.11 seconds / 60 degrees
- Temperature Range : 0 ℃ -- 55 ℃
- The dead-band setting : 5 microseconds
- Line Length : 300MM
- Control system : PWM

6.3. Software

6.3.1. Communication

Communication between our glove and the VR world centers around a driver. Using a modified version of the open-source VR glove driver, OpenGloves, we can send data to and from the glove via USB Serial and Named Pipes.

Glove to Driver Communication

The only data the glove is required to send to the driver is finger curl and gesture data. The firmware on the microcontroller encodes it using an alphanumeric string with a newline character as a delimiter. An encoded data packet may look like this: "A256B110C0D1023IJ\n." Each letter represents a different key (finger or gesture) paired with a key value (finger curl value or gesture status).

The list of keys:
"A" thumb curl - scalar
"B" index curl - scalar
"C" middle curl - scalar
"D" ring curl - scalar
"E" pinky curl – scalar
"L" grab - boolean
"M" pinch – boolean
"O" calibration button - boolean
"P" trigger value - scalar

These keys are strictly for finger data sent from glove to driver, a different set of keys is used for feedback data sent from driver to glove.
Finger curl values range from our microcontroller's max analog value (4095) to 0.

Driver to Glove Communication

The driver receives feedback data (haptic, thermal, and force) from the Game Engine (Unity) using Named Pipes and encodes the data into the same alphanumeric string with its own set of keys.

The list of keys:
"A" thumb curl limit - scalar
"B" index curl limit - scalar
"C" middle curl limit - scalar
"D" ring curl limit - scalar
"E" pinky curl limit – scalar
"I" thermal value - scalar
"J" thumb vibration – scalar
"K" index vibration – scalar
"L" middle vibration - scalar
"M" ring vibration – scalar
"N" pinky vibration – scalar

This data packet gets sent to the glove, which decodes the data and sets the corresponding values of its components.
Force feedback values are integers from 0 to 1000. Thermal feedback values are -1000 to 1000 (negative is cold, positive is hot). Haptic feedback values are integers from 0 to 10000 (in milliseconds).

Driver to Game Engine Communication

The driver receives finger tracking data from the glove and decodes it to emulate the virtual hand in-game. It can communicate with the Game Engine by utilizing a VR runtime. A VR runtime, such as OpenVR or OpenXR, is a software layer that provides a standard way for applications to interact with a variety of virtual reality hardware, allowing developers to create VR experiences without needing to code for specific devices. Our VR runtime of choice is OpenVR, and the driver can interact with the runtime to enable the use of the in-game hand and finger tracking, from the data sent from the glove.

6.3.2. Game Engine (Unity)

Our 3D virtual demo environment is created using the game engine Unity. It is known for its user-friendly interface, making it accessible while still offering advanced features. It also provides robust VR support with a VR runtime API interface that can interact with VR devices. This makes it easy to integrate our emulated hand in-game by using OpenVR.

Interaction System

Utilizing OpenVR allows us to use their robust interaction system. We can place objects anywhere in the world and grab, throw, and move them. However, we still need a way to send data from Unity to our driver when interacting with certain objects so we can trigger the corresponding components on our glove. We achieve this by using three Named Pipes between Unity and our driver, one for force, thermal, and haptic feedback. Named Pipes are a way to provide inter-process communication among processes running on the same machine. When data for a certain feedback system needs to be updated, we send an updated struct to the driver through its respective Named Pipe.

Example of force feedback struct:

```
struct FFBData
{
  int16_t thumbCurl;
  int16_t indexCurl;
  int16_t middleCurl;
  int16_t ringCurl;
  int16_t pinkyCurl;
};
```

The driver then parses this data, encodes it, and sends it to the glove's microcontroller.

Modes of Interaction

On Object Pickup: The glove provides feedback when the user picks up a virtual object. For example, when the user virtually picks up an object, such as a warm cup of coffee, several feedback mechanisms are triggered. The glove's servos adjust to mimic the shape of the object, the thermal elements activate to simulate the temperature of the object, and haptic feedback is initiated on fingers to signify the act of grabbing.

In Radius of Object: The glove also provides feedback when the user is not just holding an object, but also in radius of an object. For example, if the user is in the radius of a hot object, such as a campfire, the glove's thermal elements adjust their temperature based on the proximity of the user's hand to the virtual heat source. Furthermore, haptic feedback is also triggered on individual fingers when they come into contact with virtual objects.

6.3.3. 3D demo environment

Using Unity, we build a 3D demo VR environment to test and show off our glove's capabilities. The environment includes objects of varying sizes and shapes that can be grabbed, moved, or thrown. When an object is grabbed, the glove's servos engage, restricting hand movement, allowing the user to feel the object's presence. The environment also includes objects of varying temperature, which change the temperature of the glove's Peltier module, allowing the user to feel the object's temperature. Lastly, when any finger touches an object in the virtual environment, the corresponding haptic motors turn on briefly, allowing the user to feel like they are touching the object.

Our software implementation is also straightforward to implement into existing 3D virtual games and environments. For example, in Unity, developers can include our scripts (*FBManager* and *FBClient*) that interact with our driver and add the *ObjectToggles* class onto any game object that wants to interact with the glove.

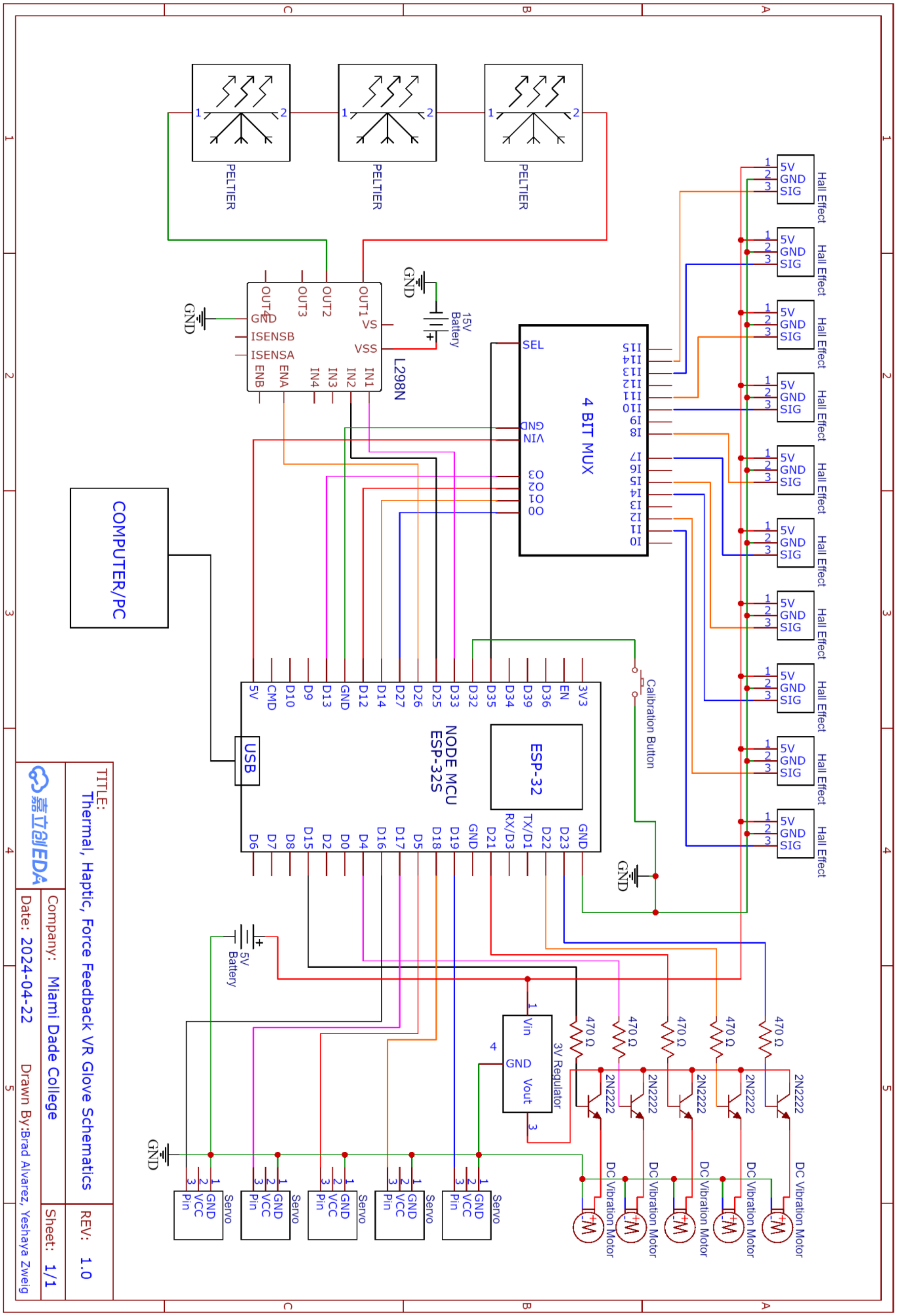
Example of ObjectToggles class in Unity:

```
public class ObjectToggles : MonoBehaviour
{
    bool isHot = false;
    bool isCold = false;
    short grabbedTemp = 0; //from -1000 to 1000
    float radiusTemp = 0.0f; //radius temp distance
    Vector3 radiusOffset = Vector3.zero;

    bool radiusHaptics = false; //haptics when touched
    bool grabbedHaptics = false; //haptics when grabbed
    short hapticTime = 150; //haptic time in ms
}
```

This class specifies how thermals and haptics should interact when an object is grabbed or in the radius of a user's hand. As for force feedback, it is applied to every interactable (object with intractable script) that has a custom hand pose created for it.

6.4. Schematic



7. Implementation

Here is a brief breakdown of all of the different parts of this project.

- For finger tracking, we have hall effect sensors and magnets to track the position of the fingers, using a 3D printed chassis.
- We also use a headset tracked controller attached to the glove to give us 3D spatial tracking in real time. This enables us to move around a room both in real life and virtually, to grab different objects.
- To restrict the fingers from closing when holding a virtual object, we use servo motors to prevent the fingers from extending.
- To emulate thermal response, we use flexible Peltier pads that can heat up or cool down the hand based on the 3D environment.
- To emulate touch, we vibrate small haptic devices attached to each finger when our virtual finger touches an object.
- Our glove's communication with the VR world centers around a custom driver which interacts with the glove's microcontroller and the game engine/VR runtime.

8. Safety

The glove has no environmental concerns, as it does not have a carbon footprint. In regard to danger to the user, the worst that can happen is that the heaters overheat, but this is heavily restricted in the software. The servo motors are not strong enough to cause any damage to the user, in case of malfunction.

9. Future plans

This version of a VR Glove is hopefully the first of many iterations of *the* next powerful way to interact with virtual environments. With all of these features at this low cost, this glove is part of the cutting edge of VR/AR technology today. Advanced sensors and features will hopefully be added eventually, like tactile feedback, although now the cost of entry is too great for the average consumer. This additional feature will ensure higher accuracy of training scenarios, as well as enhance immersion of recreational uses. Our vision is a future where this kind of glove technology is standard with every VR/AR headset sold.

10. Cost to produce/List of materials

Glove Parts	Quantity	**Cost**
Flexible thermoelectric devices	3	130
Tiny magnets	25	8.99
Ball bearings	10	7.65
Multiplexer	1	6.99
Hall effect sensors	10	11.99
3D printed parts		10
L298N motor driver	1	3.22
Bicycle gloves	1	17.74
Foam roll		11.99
USB C cables	2	17.98
ESP32 dev boards	1	13.99
Straps		12.86
Charger blocks	2	24.98
Servos	5	20.99
Power supply 3.3v 5v	2	12.83
Assorted wires		14.43
3.3v regulator	1	9.99
Mini vibration motors	5	8.99
470 ohm resistors	5	.20
2n2222 transistors	5	.10
Button	1	.01
Total:		**345.92**