ELECENG 4OI6A
ECE Capstone Progress Phase A
Ergonomic Computer Mouse for Individuals Lacking Hand Dexterity

Instructor: Dr. Abdelhadi
Section: C02
Group 10

Luke West – westl5 – 400310803
Ryan Xu – xur76 – 400309994
Matthew Yu – yum77 – 400322243
George Gill – gillg62 – 400327563

Date: January 17th, 2024

Table of Contents

Abstract and Summary of the Design

The computer mouse is a common household tool which has very straightforward operation for most of the population. But for those with limited finger and hand dexterity, or those lacking a hand entirely, the traditional design can be difficult to use. We propose a wrist band mounted design, which eliminates the use of the hand entirely, placing emphasis on use of the forearm instead. Our design aims to accomplish all normal functions of the traditional mouse, namely cursor movement, clicking, and scrolling. The mouse movement is achieved by moving the forearm in the horizontal plane, clicking is achieved via rotation about the forearm axis, and scrolling is achieved via pitch angle of the forearm. Our device is controlled via ESP32 microcontroller, which takes input from a PixArt PAJ7025R3 multiple objects tracking IR sensor, and MPU 6050 inertial measurement unit, and sends mouse input commands to the user's computer over Bluetooth (or UART via USB-C). The device also includes tactile clicking feedback via a haptic LRA motor.

Current Status

Implementation

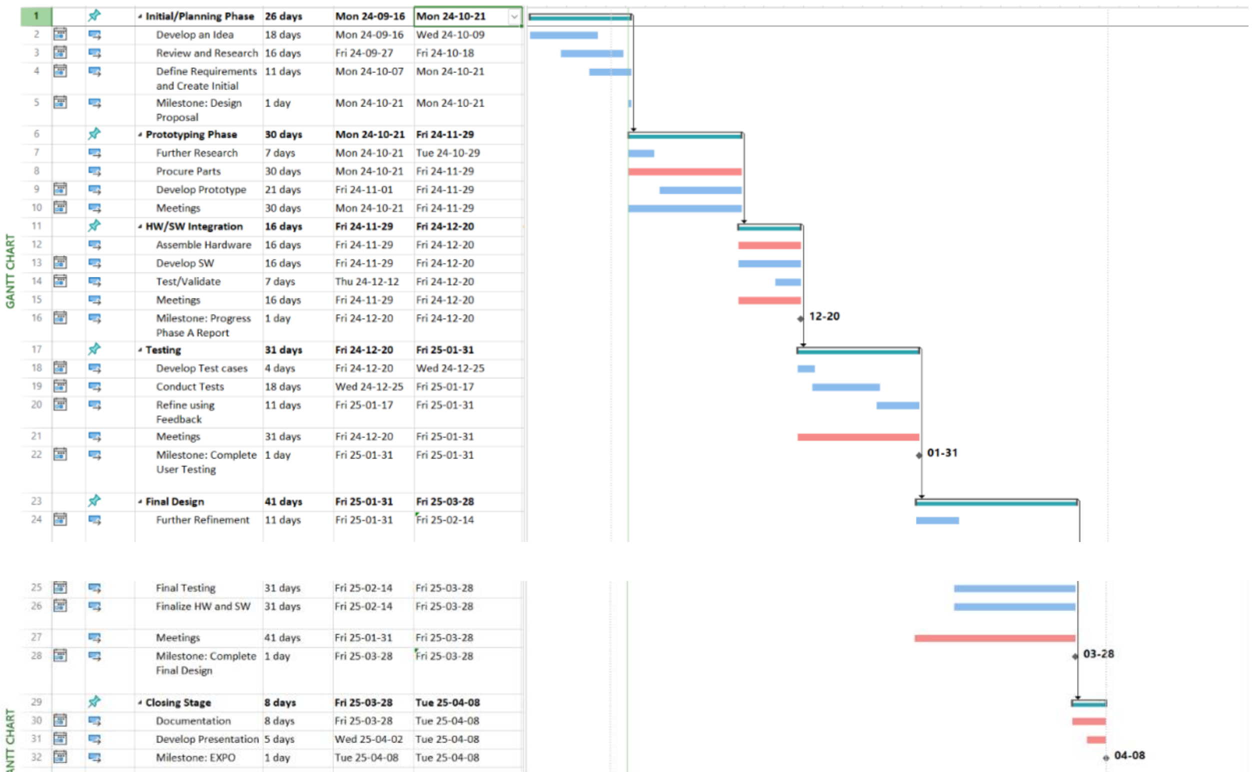
The following is a list of notable achievements towards the project in phase A.

- Procurement of main components which are shown in the expenditure section
- Design and acquisition of PCB to interface with PixArt PAJ7025R3 sensor
- Successful communication with PixArt PAJ7025R3 sensor and 2D visualization of tracked objects
- Completion of overall circuit design and schematic
- Simulation of mouse cursor tracking in Python
- Demonstration of haptic feedback based on rotation of IMU
- Prototype 3D models of wrist band to hold all components, as well as an initial four-point 3D-model of structure to hold four IR LEDs
- Testing with ESP-32 Bluetooth capabilities, including moving a mouse cursor with pre-programmed movements

Scheduling

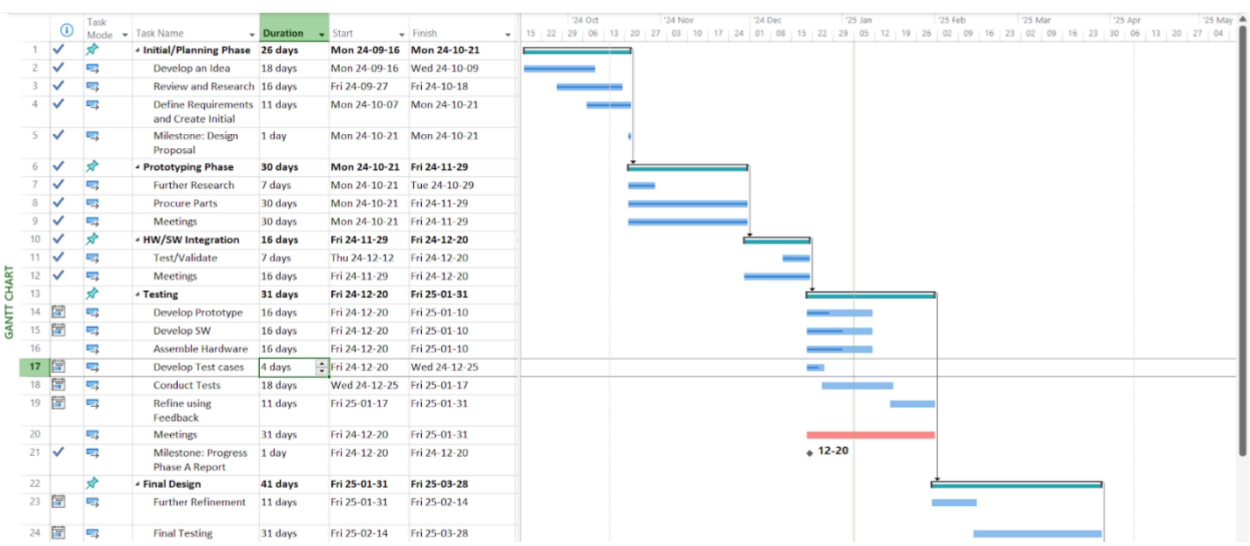
The Gantt chart illustrates the project's progress over the last quarter of 2024. We are generally on schedule, with a few minor setbacks that have altered the forthcoming schedule. Setbacks from procuring parts and some issues interfacing with the PixArt sensor caused a significant delay in creating a working prototype. Additionally, this also halted the hardware/software assembly tasks.

This will cause a change in the testing stage, as we will cut into this phase to crash the uncompleted tasks. This is reflected in the new Gantt chart, where tasks "Develop prototype," "Assemble Hardware," and "Assemble Software" have been moved to the testing phase, resulting in less time for the original testing phase tasks. Fortunately, these tasks should be quick to complete, as the challenges have been addressed, such as the created PCB for interfacing with the PixArt sensor.



	Task Name	Duration	Start	Finish
1	**Initial/Planning Phase**	**26 days**	**Mon 24-09-16**	**Mon 24-10-21**
2	Develop an Idea	18 days	Mon 24-09-16	Wed 24-10-09
3	Review and Research	16 days	Fri 24-09-27	Fri 24-10-18
4	Define Requirements and Create Initial	11 days	Mon 24-10-07	Mon 24-10-21
5	Milestone: Design Proposal	1 day	Mon 24-10-21	Mon 24-10-21
6	**Prototyping Phase**	**30 days**	**Mon 24-10-21**	**Fri 24-11-29**
7	Further Research	7 days	Mon 24-10-21	Tue 24-10-29
8	Procure Parts	30 days	Mon 24-10-21	Fri 24-11-29
9	Develop Prototype	21 days	Fri 24-11-01	Fri 24-11-29
10	Meetings	30 days	Mon 24-10-21	Fri 24-11-29
11	**HW/SW Integration**	**16 days**	**Fri 24-11-29**	**Fri 24-12-20**
12	Assemble Hardware	16 days	Fri 24-11-29	Fri 24-12-20
13	Develop SW	16 days	Fri 24-11-29	Fri 24-12-20
14	Test/Validate	7 days	Thu 24-12-12	Fri 24-12-20
15	Meetings	16 days	Fri 24-11-29	Fri 24-12-20
16	Milestone: Progress Phase A Report	1 day	Fri 24-12-20	Fri 24-12-20
17	**Testing**	**31 days**	**Fri 24-12-20**	**Fri 25-01-31**
18	Develop Test cases	4 days	Fri 24-12-20	Wed 24-12-25
19	Conduct Tests	18 days	Wed 24-12-25	Fri 25-01-17
20	Refine using Feedback	11 days	Fri 25-01-17	Fri 25-01-31
21	Meetings	31 days	Fri 24-12-20	Fri 25-01-31
22	Milestone: Complete User Testing	1 day	Fri 25-01-31	Fri 25-01-31
23	**Final Design**	**41 days**	**Fri 25-01-31**	**Fri 25-03-28**
24	Further Refinement	11 days	Fri 25-01-31	Fri 25-02-14
25	Final Testing	31 days	Fri 25-02-14	Fri 25-03-28
26	Finalize HW and SW	31 days	Fri 25-02-14	Fri 25-03-28
27	Meetings	41 days	Fri 25-01-31	Fri 25-03-28
28	Milestone: Complete Final Design	1 day	Fri 25-03-28	Fri 25-03-28
29	**Closing Stage**	**8 days**	**Fri 25-03-28**	**Tue 25-04-08**
30	Documentation	8 days	Fri 25-03-28	Tue 25-04-08
31	Develop Presentation	5 days	Wed 25-04-02	Tue 25-04-08
32	Milestone: EXPO	1 day	Tue 25-04-08	Tue 25-04-08

Figure 1 - Old Gantt Chart



	Task Name	Duration	Start	Finish
1	**Initial/Planning Phase**	**26 days**	**Mon 24-09-16**	**Mon 24-10-21**
2	Develop an Idea	18 days	Mon 24-09-16	Wed 24-10-09
3	Review and Research	16 days	Fri 24-09-27	Fri 24-10-18
4	Define Requirements and Create Initial	11 days	Mon 24-10-07	Mon 24-10-21
5	Milestone: Design Proposal	1 day	Mon 24-10-21	Mon 24-10-21
6	**Prototyping Phase**	**30 days**	**Mon 24-10-21**	**Fri 24-11-29**
7	Further Research	7 days	Mon 24-10-21	Tue 24-10-29
8	Procure Parts	30 days	Mon 24-10-21	Fri 24-11-29
9	Meetings	30 days	Mon 24-10-21	Fri 24-11-29
10	**HW/SW Integration**	**16 days**	**Fri 24-11-29**	**Fri 24-12-20**
11	Test/Validate	7 days	Thu 24-12-12	Fri 24-12-20
12	Meetings	16 days	Fri 24-11-29	Fri 24-12-20
13	**Testing**	**31 days**	**Fri 24-12-20**	**Fri 25-01-31**
14	Develop Prototype	16 days	Fri 24-12-20	Fri 25-01-10
15	Develop SW	16 days	Fri 24-12-20	Fri 25-01-10
16	Assemble Hardware	16 days	Fri 24-12-20	Fri 25-01-10
17	Develop Test cases	4 days	Fri 24-12-20	Wed 24-12-25
18	Conduct Tests	18 days	Wed 24-12-25	Fri 25-01-17
19	Refine using Feedback	11 days	Fri 25-01-17	Fri 25-01-31
20	Meetings	31 days	Fri 24-12-20	Fri 25-01-31
21	Milestone: Progress Phase A Report	1 day	Fri 24-12-20	Fri 24-12-20
22	**Final Design**	**41 days**	**Fri 25-01-31**	**Fri 25-03-28**
23	Further Refinement	11 days	Fri 25-01-31	Fri 25-02-14
24	Final Testing	31 days	Fri 25-02-14	Fri 25-03-28

Figure 2 - Updated Gantt Chart

Expenditure

Current Spending

To date, our group has acquired the following components and materials essential to the prototyping and testing phases of the project. For price break down please see the bill of materials.

- Reflective Tape: Used to test how accurate the PixArt sensor tracks reflections from the IR emitter.
- IR Emitters & Receiver Package: Transmits IR for tracking purposes with the PixArt sensor.
- ESP32: Controller used for interacting with other devices and processing sensor outputs.
- Motor Unit: Provides user feedback through vibrations for interaction with the system.
- Motor Driver: Controls the motor units for providing feedback.
- Motor Driver Cable: Connects the motor driver to the system for signal transmission.
- PixArt Test PCB: Connects the PixArt sensor to the system as it does not come with a pre-made connector.
- Heat Resistant Tape: Protects components during the soldering process to avoid heat damage.
- Flux Paste: Improves soldering quality, ensuring clean and strong connections.
- IR Emitter (860NM): Offers a better NM rating for the sensor to detect, compared to previous emitters.
- PixArt Sensors: Main sensor for tracking movement in the system.

The total expenditure amounts to $260.16 CAD for the prototyping and testing phase of the project.

Team Member Personal Contributions & School Resources

Several team members have provided personal materials that they already owned to support the project which has significantly reduced overall costs. These materials include:

- Helping Hand Station for Soldering
- Soldering Iron
- Solder Wire
- Breadboards
- Capacitors
- Resistors
- Wires
- 3D Printer + Filament
- Makerspace Materials

Projected Future Expenses

Our team estimates an additional $200 CAD for the following:

- Custom PCB for wrist mounting (Possible flexible PCB)
    - Purpose: To integrate all components into a compact, wrist friendly design. Flex PCB would improve comfort and durability.
- Smaller and more power efficient ESP32
    - Purpose: Reduce size and device and extend battery life
- Flexible and comfortable material for wrist strap
    - Purpose: Provide user-friendly and ergonomic strap so mouse can be used for prolonging periods of time.
- Battery
    - Purpose: Power device while maintaining portability and compactness.

Technical Implementation

Physical Design

Circuit Design and Hardware

The electrical schematic created in KiCAD outlines the configuration of the microcontroller, sensors, and other components. Acting as a map, it will provide significant assistance during the physical wiring of the device. There are two critical aspects of the schematic. Firstly, the design that will be located on the wrist. This includes the PixArt sensor, microcontroller, haptic motor drivers, motors, IMU, and a battery. The microcontroller interfaces with all components through its various GPIO pins and is powered by an external lithium-polymer battery or USB-C cable connected to the computer. The KiCAD schematic closely follows the rough schematic, with slight tweaks. Additionally, our group checked if the FireBeetle2 and Adafruit E-Feather have similar configurations due to their benefit of onboard voltage regulator and noticed that the FireBeetle 2 is almost a perfect fit.

At the current stage, all testing is being done using a Node MCU development board which does not have built in battery implementation. We determined that an implementation including the battery is optional for the completion of the project, as a wired design has its own benefits, and allows for testing without draining a battery which could get expensive. In the later stages of the project, we intend to include a battery and communicate using Bluetooth, as this should be more physically comfortable for the user.

Note: The microcontroller used in the KiCAD schematic is the FireBeetle 2 ESP32-E. The FireBeetle 2 has an onboard voltage regulator, reducing the input voltage from the selected 3.7 volts to the desired 3.3 volts. It also has a built-in PH2.0 lithium battery connector interface and an on-board charging circuit, allowing easy battery charging using the USB-C port. This eliminates the need for an external voltage regulation circuit shown on the original rough schematic which would be required using the current ESP32 microcontroller development board, reducing the footprint of the schematic.

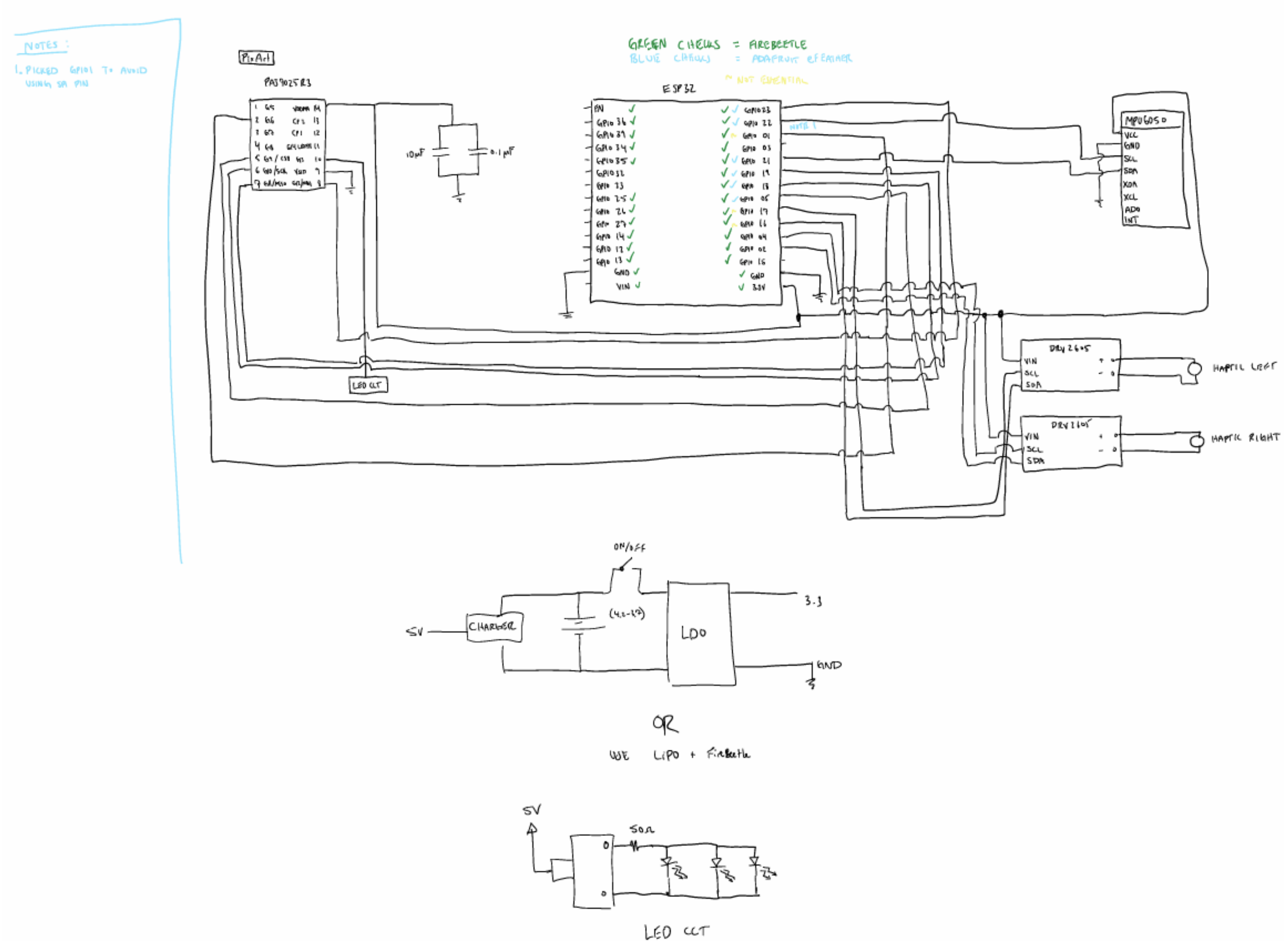


Figure 3 - Rough Schematic

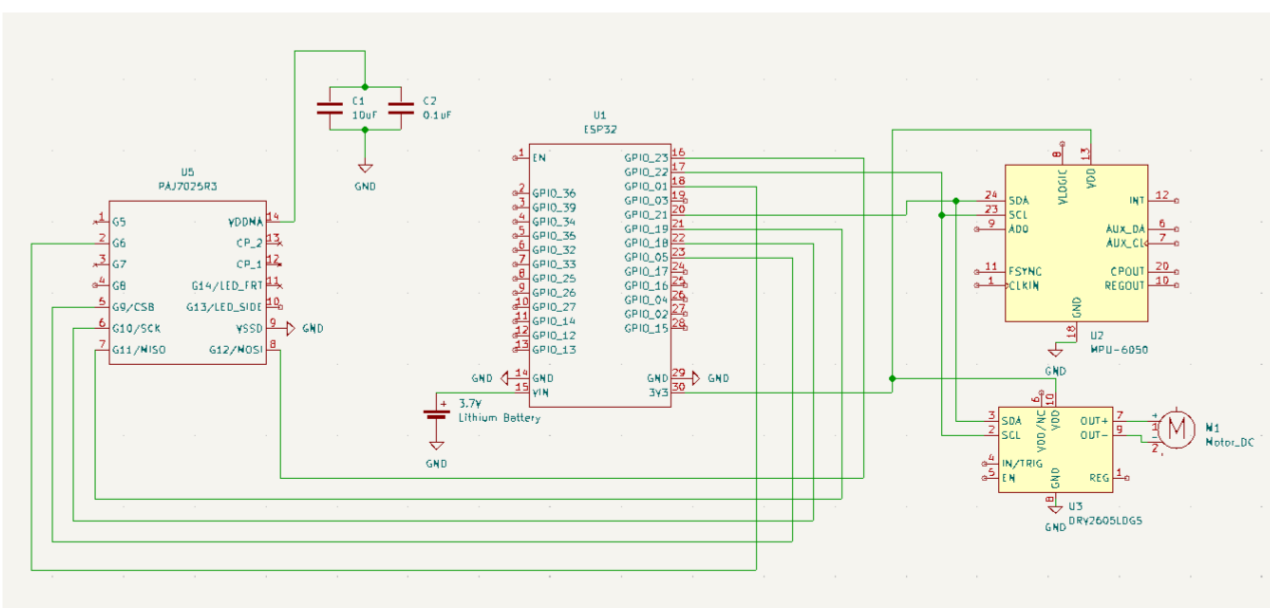


Figure 4 - Main Schematic on KiCAD

The second aspect of the schematic displays the IR LED circuit. The LEDs are configured in a parallel circuit and powered using an external lithium battery. Each diode has a 100-ohm resistor to limit the current to 8mA. The current can be altered using additional resistors if required. Lastly, a switch controls if the circuit is active, allowing the user to turn on/off the circuit accordingly. This schematic may undergo changes in the future following the prototyping stage.

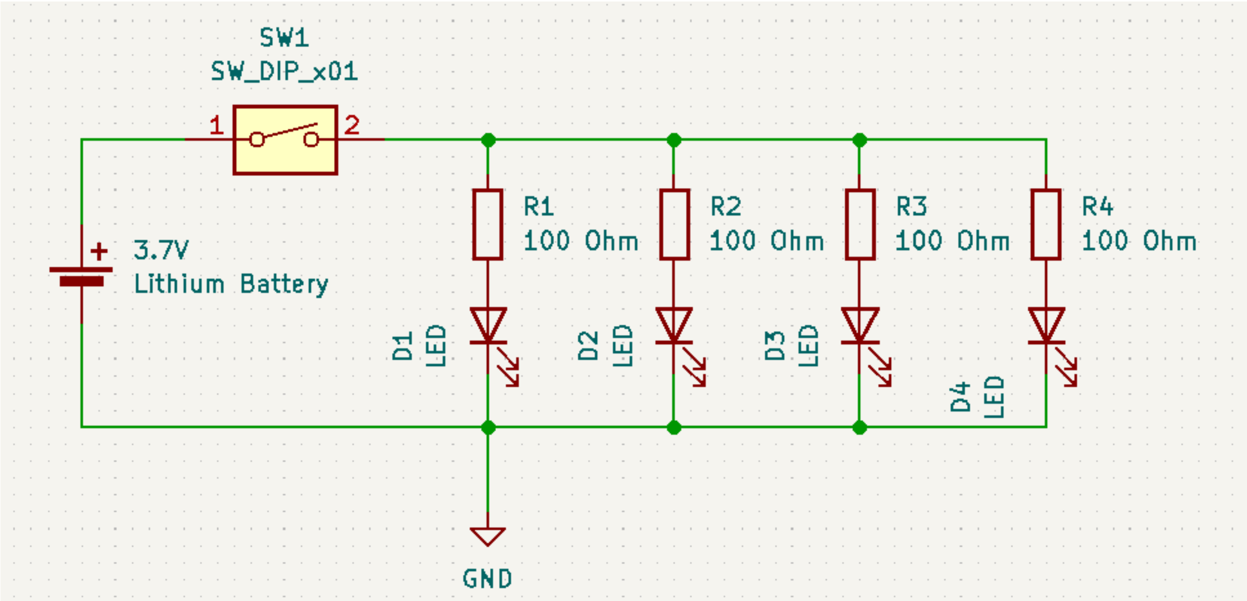


Figure 5 - Schematic of IR LED Circuit on KiCAD

PixArt Custom PCB Design

In the early stages, our group struggled to interface with the PixArt PAJ7025R3 sensor due to limited knowledge and a lack of pre-made connectors. The datasheet provided minimal guidance, leaving us uncertain about how to proceed. Eager to advance, we tried various methods, including soldering wires directly, creating a custom connector, and reaching out to companies for support. However, these attempts failed, and one sensor was damaged in the process. The long lead time for a custom PCB added further delays, but we ultimately determined that designing a custom PCB was the most reliable solution. A project by Bart Trzynadlowski using the PAJ7025R2 variant of our sensor was referenced in the design[1].

From doing research, our group decided to learn KiCAD to design the PCB as it is known as a beginner friendly software. The custom PCB that was created includes:

- Two-layer construction: A compact and efficient layout for seamless integration. Custom through-hole connector: Designed to provide a stable and secure connection to the sensor.
- Pin header connection: Precision-routed traces ensuring reliable communication between the sensor and the system.
- Clear labeling: Each trace and pin has been clearly labeled to improve assembly accuracy and facilitate future troubleshooting.

This process not only addressed the immediate need for a stable connection to the sensor but also significantly enhanced my skills in PCB design. Despite the initial setbacks, this solution enabled us to move forward with the project, exemplifying our ability to overcome technical challenges through proactive problem-solving and innovation. The experience has been invaluable in ensuring that our system is both reliable and scalable as we continue development. An image of the PCB design can be found below with more details of the design in the appendix.

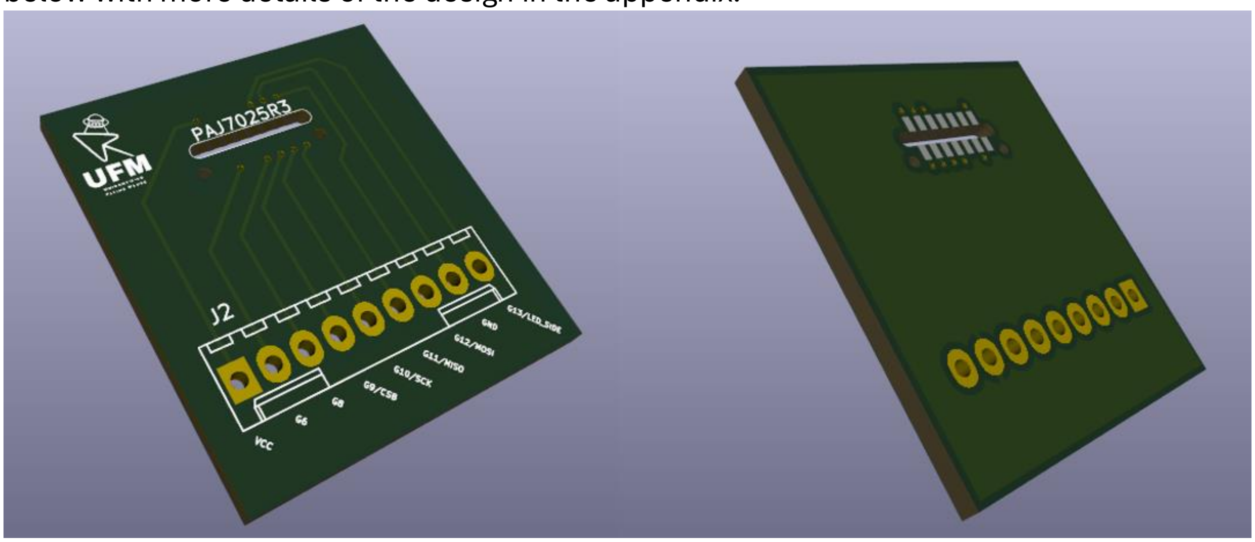


Figure 6 - Custom PixArt Sensor PCB Front and Back View Iteration 1

PixArt PCB Redesign

After a two-week lead time, the PCBs were delivered, and our team successfully soldered the sensor onto the custom design. This first iteration of the PCB was intentionally overdesigned to account for all potential pin configurations we deemed useful for the sensor. However, through extensive testing and prototyping, we identified several opportunities for optimization:

1. Pin Reduction: Of the original 9 pins, only 6 were necessary for the sensor's operation, allowing us to simplify the design and reduce unnecessary traces.
2. Proximity of Capacitors: Following recommendations from the sensor's datasheet, we ensured that decoupling capacitors were placed as close to the sensor as possible to improve signal stability and reduce noise. Proper placement of capacitors is critical for mitigating power supply fluctuations and ensuring consistent performance [2].
3. FFC/FPC Cables: To enhance flexibility and reduce weight, we transitioned from traditional pin headers to flat flexible cables (FFC) and flexible printed circuits (FPC). These components are significantly thinner and more compact, making them ideal for wearable applications where size and weight are critical considerations [3], [4].
4. Mounting Holes: Strategically placed mounting holes were added to secure the PCB within the 3D-printed housing, ensuring stability during operation [5].

Capacitor Selection

For the second iteration, we added two capacitors to the design: a 10 μF capacitor and a 0.1 μF capacitor. The selection process for these components was crucial to optimizing the PCB's performance. Here is an overview of the types of capacitors we considered and the reasoning behind our final choice:

- Ceramic Capacitors:
  - Pros: Compact, low cost, and excellent frequency response. They are ideal for decoupling applications due to their low equivalent series resistance (ESR).
  - Cons: Limited capacitance range, and their capacitance can vary with voltage (known as the DC bias effect).
- Tantalum Capacitors:
  - Pros: High capacitance in a small form factor, stable performance, and low leakage current. They are well-suited for applications requiring bulk storage and stability.
  - Cons: More expensive than ceramic capacitors and can be sensitive to voltage spikes, potentially leading to failure.

- Electrolytic Capacitors:
    - Pros: High capacitance values and relatively low cost. Suitable for bulk storage.
    - Cons: Larger size, shorter lifespan, and higher ESR compared to other types.
- Film Capacitors:
    - Pros: Very stable and reliable, with excellent tolerance.
    - Cons: Large size and relatively high cost, making them unsuitable for compact designs.

We ultimately chose a ceramic capacitor (0.1 μF) for decoupling due to its excellent frequency response and low ESR, which are crucial for filtering out high-frequency noise. For bulk capacitance, we opted for a tantalum capacitor (10 μF) due to its small form factor and stable performance. This combination balanced cost, size, and functionality [2].

Connector Selection: FFC vs. FPC

Transitioning to FFC/FPC connectors significantly improved the compactness and weight of the design. These connectors are widely used in applications requiring thin, flexible connections.

- Flat Flexible Cables (FFC): Comprised of flat and flexible ribbon-like cables, FFC connectors are simple to use and cost-effective[3], [4].
    - Pros: Low profile, highly flexible, and widely available in standard configurations.
    - Cons: Limited durability and can be prone to damage if handled improperly.
- Flexible Printed Circuits (FPC): Like FFC but more robust, FPC connectors use circuits printed directly onto a flexible substrate [3], [6].
    - Pros: Higher durability, better electrical performance, and customizable configurations.
    - Cons: More expensive and requires precise manufacturing processes.

For our application, FFC was chosen due to its simplicity and cost-effectiveness, aligning with the lightweight and compact nature of our design.

When it came to picking the type of FFC connector, there were a plethora of them to look at. Our team considered the pros and cons of each locking feature:

- Back lock: A simple, rear-entry locking mechanism where the cable is inserted and locked by engaging a mechanism at the back of the connector.
    - Pros: Provides a solid, secure lock with low chances of accidental disconnection. It’s also relatively straightforward and easy to implement.
    - Cons: In some designs, it might be harder to disengage or re-connect, especially in tight spaces.

- Flip lock: A locking mechanism where a flap or lever is flipped over the cable to lock it in place.
    - Pros: Simple to use and offers a quick connection. It’s also quite secure in some designs.
    - Cons: Can be bulky, and the flipping motion might wear out over time, especially in high-usage applications.
- Flip lock, Back lock: Combines the flip mechanism with a back-locking feature to secure the cable.
    - Pros: Provides both secure locking and ease of assembly/disassembly.
    - Cons: Combining both mechanisms can lead to a larger connector footprint, potentially affecting space and weight-sensitive designs.
- Latch Lock: A latch engages to lock the connector in place, often requiring the latch to be pulled or pressed to disengage the cable.
    - Pros: Ensures a firm connection, preventing the cable from accidentally detaching.
    - Cons: Can add complexity to the connector, potentially requiring more force to disconnect, and may not be as compact as other options.
- Rotary Lock: The cable is inserted into the connector, and a rotary mechanism is used to twist and lock the connector in place, ensuring a tight, secure fit.
    - Pros: The rotation provides a strong, mechanical engagement, making the connection extremely stable. It's resistant to vibrations and accidental disconnections.
    - Cons: It requires a bit more effort to engage and disengage compared to simpler mechanisms.
- Rotary Lock, Back lock: Combines the rotary mechanism with the back lock feature to create an even more secure connection.
    - Pros: Offers a very secure connection, combining the mechanical stability of the rotary lock with the added safety of the back lock. This ensures the cable remains firmly locked in place, even in high-movement or vibration-prone environments.
    - Cons: Can be bulkier than some other locking mechanisms, and the combined mechanism may require additional space or assembly effort.
- Slide Lock: A slide mechanism locks the connector by sliding a piece over the cable.
    - Pros: Quick to engage and disengage, with minimal effort required.
    - Cons: May not offer the same level of security as other locking mechanisms, especially in high-vibration environments.

We chose the Rotary Lock, Back lock mechanism for its combination of security, compactness, ease of use, and durability. Its rotary feature ensures a stable, tight connection that resists accidental disconnections from vibrations or mechanical stress, which is crucial for our wearable design. Despite its strength, it adds minimal bulk compared to other locking mechanisms, helping maintain a lightweight and unobtrusive design. The back lock adds an extra layer of security, ensuring the cable stays securely locked even under light pulls or movements. Additionally, its robust design ensures long-term reliability, reducing wear and tear from regular use [6].

NPT vs. PTH (Non-Plated Through-Hole vs. Plated Through-Hole)

When adding mounting holes, we considered the difference between NPT (Non-Plated Through-Hole) and PTH (Plated Through-Hole) designs:

- Non-Plated Through-Holes (NPTH):
    - Pros: Easier to manufacture and more cost-effective for applications where electrical connection is not required.
    - Cons: Not suitable for carrying signals or power.
- Plated Through-Holes (PTH):
    - Pros: Allow for electrical connections between layers and provide mechanical strength.
    - Cons: More expensive and complex to manufacture.

For our mounting holes, we chose NPTH since they only serve as mechanical supports and do not require electrical connectivity [5], [6].

Results

With these refinements, the second iteration of the PCB included:

- 1 x 10 μF tantalum capacitor
- 1 x 0.1 μF ceramic capacitor
- 1 x FFC connector
- 4 x NPTH mounting holes

By reducing the number of traces and using a smaller connector, we scaled the PCB down from 28 mm x 30 mm to 18 mm x 20 mm, even with the addition of two capacitors. This miniaturization represents a significant milestone in ensuring the device remains wearable and unobtrusive.

The iterative process not only improved the PCB's functionality and design but also highlighted the importance of component selection and user-driven refinement in achieving project goals. The redesign of the PCB along with 3D models of the electrical components are shown below with more information found in the appendix:

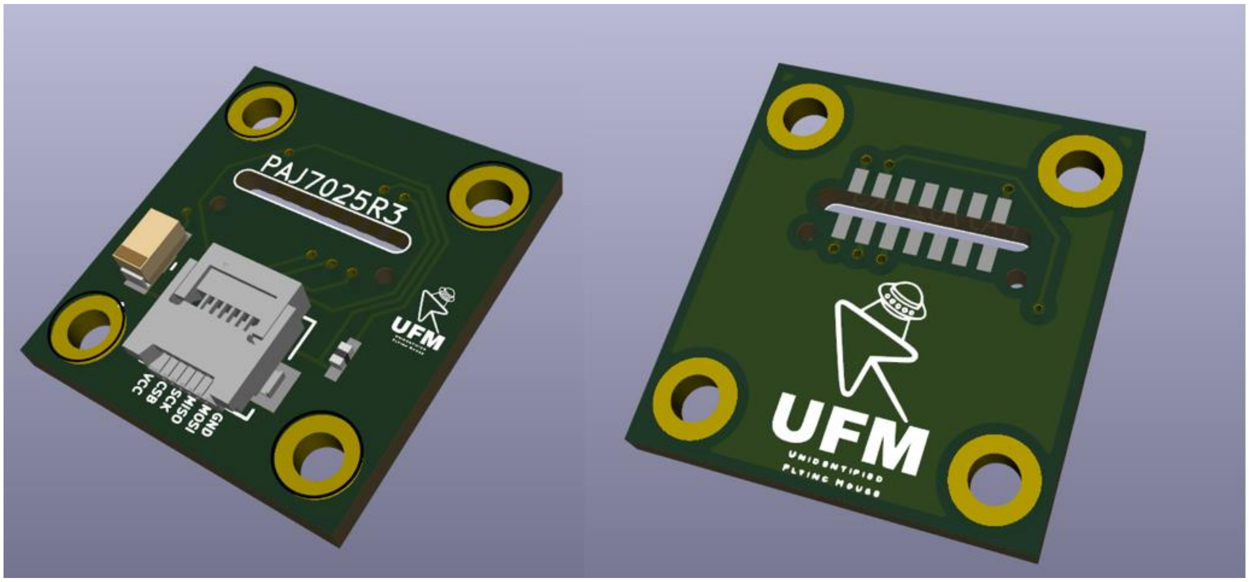


Figure 7 - 3D Render of Custom PixArt Sensor PCB Front and Back View Iteration 2

Procurement of Additional PAJ7025R3 Sensors

After damaging one of the two original sample sensors, our group decided to purchase more as a contingency if we damaged the second sensor. This proved more challenging that we initially thought, as the sensor is not carried by typical consumer level vendors such as Digikey or Mouser. After discussion with PixArt, they informed us that direct orders would only be available in bulk, with a minimum number of 90 sensors. Furthermore, we were informed that due to a change in manufacturing, ordering directly from PixArt would result in an order not being shipped until February. As this is well outside our price range, too long of a lead time and an unnecessary number of sensors, we investigated other alternatives. The options were sparse, and after communication with some vendors, we were eventually able to order a more appropriate sample size of 5 sensors from Codico, a company based in Europe. It required some communication with the McMaster ECE department as well to complete the order, as Codico only takes orders business to business. These additional sensors were delivered to McMaster, and our group was able to pick them up on January 8th.

Wrist Band Design

The wristband design focuses on creating two enclosures, one for the camera mounted on the side of the wrist and a second to hold most of the components on top of the wrist. We decided to separate the design into two compartments instead of a single enclosure, due to size regulations of the average wrist size. Firstly, we predefined the key needs to address this aim, such as ensuring the design is secure, comfortable, lightweight, and safe. The most important factor is comfortability, as the design should be ergonomic and accommodate prolonged use. Additionally, the design should be securely mounted on the wrist and durable to hold the fragile components, protecting it from minor wear and tears. Moreover, the design should be safe, with multiple holes for heat dissipation and no sharp corners. Lastly, the size and weight were considered. The wristband should be adjustable and accommodate wrists of 14 cm to 22 cm in circumference, utilizing a clamping method that can be mounted using a single hand. Typically, an average watch weighs around 120 grams. Our goal was to keep the weight constrained to 150 grams or less. The total weight of all our components is around 70 grams, however, to streamline the process we left out the battery and a haptic driver, resulting in the total weight of the components being only around 50 grams. This provides around 100 grams for the wristband itself.

In later stages of the project, the inclusion of a battery will require a small redesign of the enclosure. The added battery would be a 550 mAh lithium polymer battery, whose size roughly matches the dimensions of the existing enclosure (50.5 x 34 x 3 mm) and would only add around 10g to the weight of the wristband. At 550mAh, we estimate this would translate to roughly 3 hours of use before needing to recharge, assuming the ESP32 consumes a constant worst case ~200mA on average (150 mA for RF communications, 6mA for the MOT sensor, 4mA for the MPU6050, 30 mA for haptic motor, rounded up to 200mA) [7]. A more detailed component selection for the battery still needs to be made. There are also numerous power optimizations, such as sleep mode, which can improve battery life.

Next, a CAD prototype was created to visualize a potential design, constructed using Autodesk Inventor. The wrist enclosure is a simple (55mm x 36mm x 14mm) box. The enclosure features a two-piece assembly consisting of a box and a lid, fastened together with M2 fasteners and heatsinks. This small footprint was achieved by stacking the ESP32 on top of the other components, resting on four pillars. The IMU and Haptic Driver, sit under the microcontroller, with designated mounting holes to ensure they are snugly in place. The small opening at the bottom of the box is for the motor to sit in and directly contact the skin, for optimal haptic feedback. We considered adding a protective membrane between the user's skin and the motor, this may be implemented in the future depending on our feedback. Additionally, there are two slits in the box, one for the wires which interface with the camera, and another for connecting to the microcontroller itself. These slits also provide ventilation for heat to safely dissipate.

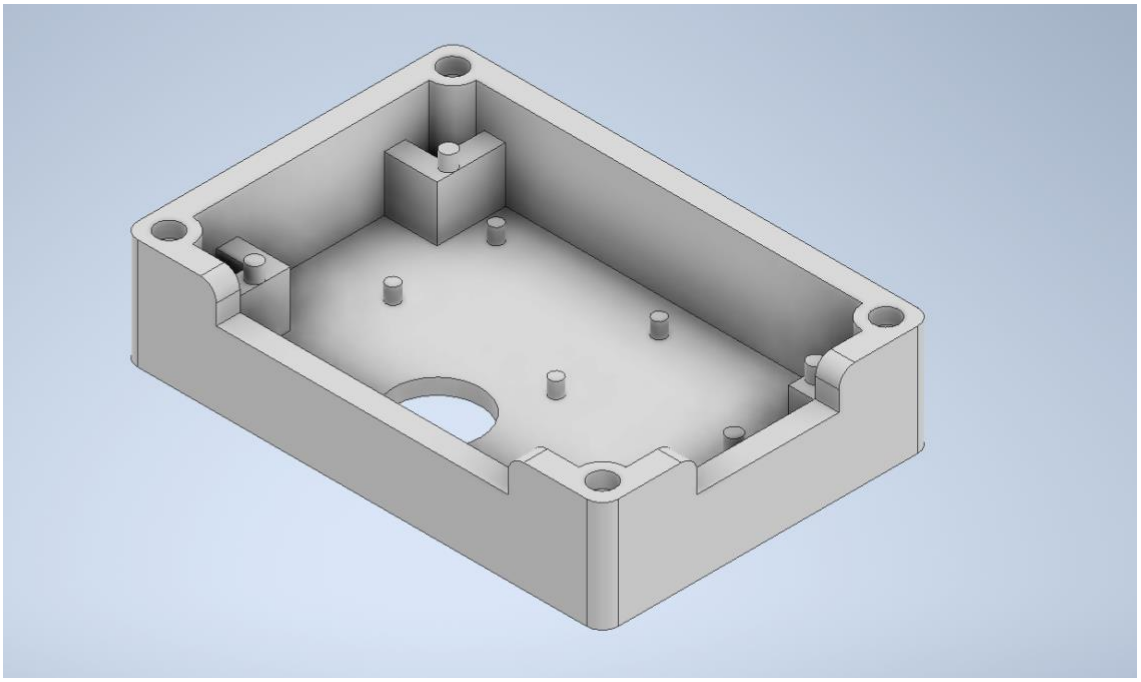


Figure 8 - CAD of Wrist Enclosure

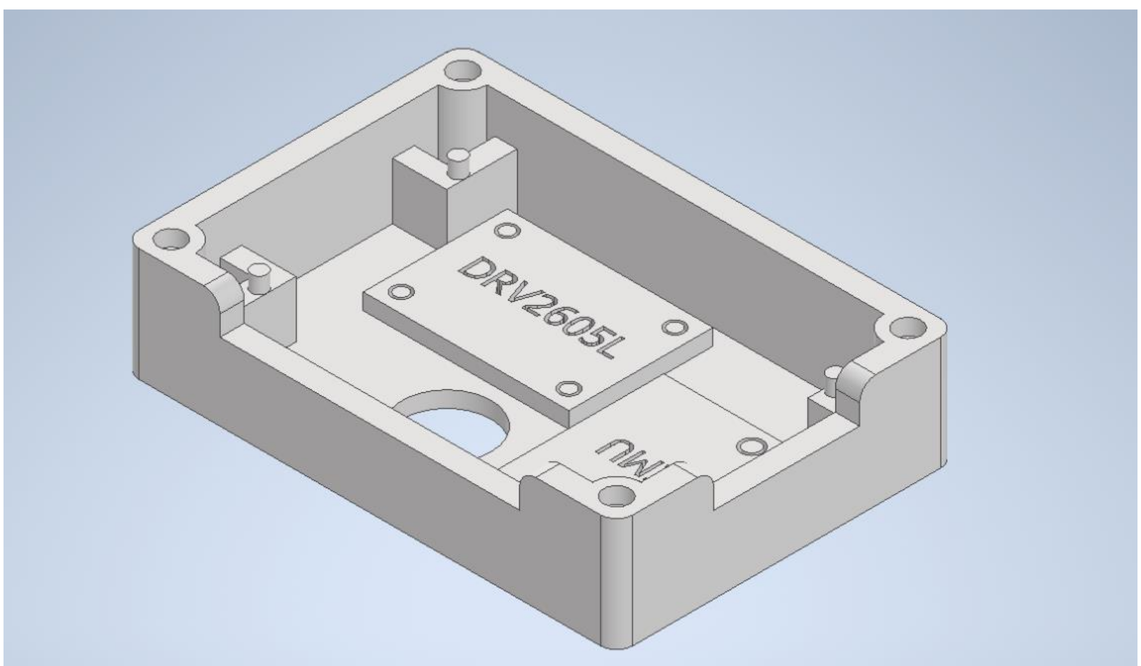


Figure 9 - Wrist Enclosure with IMU and DRV2605

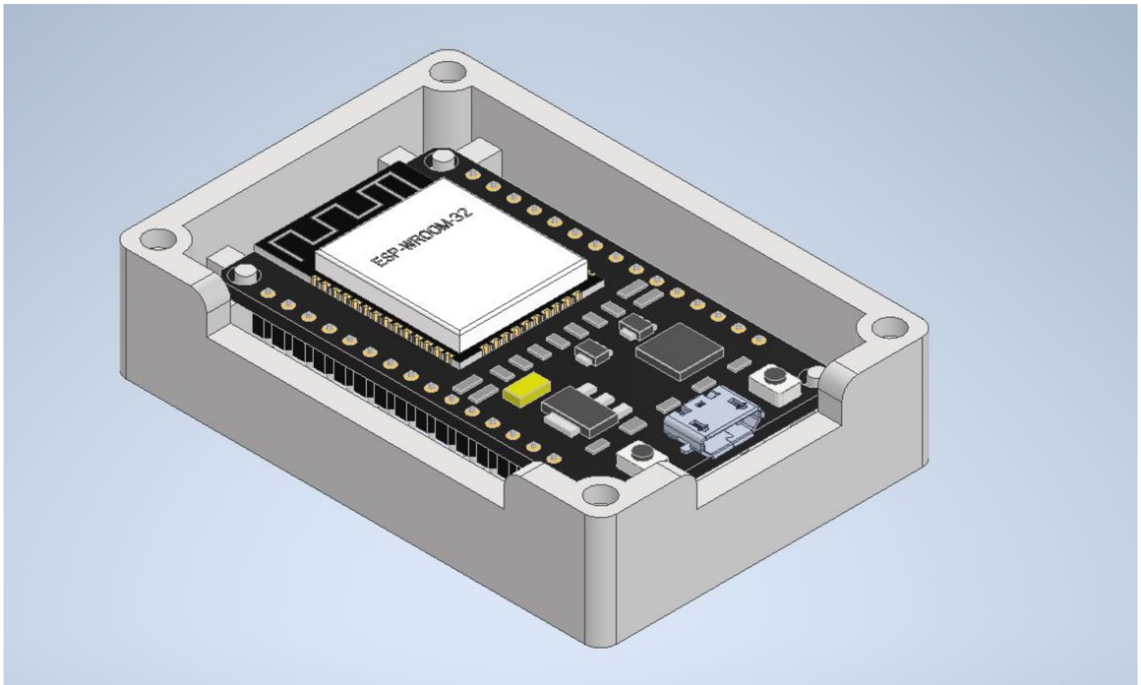


Figure 10 - Wrist Enclosure with IMU, DRV2605 and ESP-32

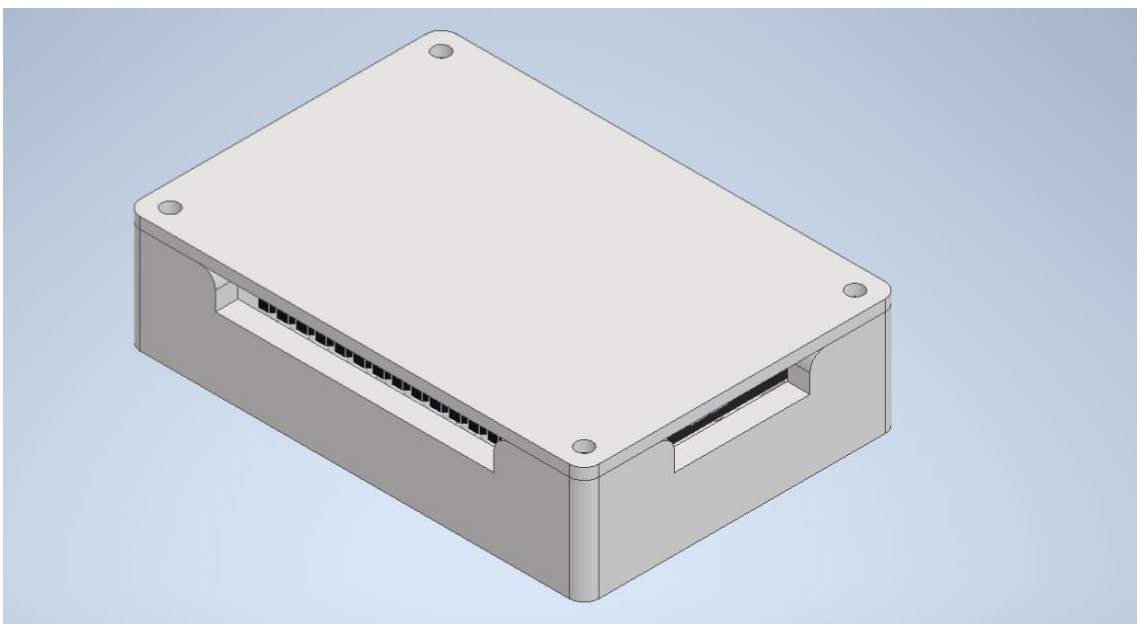


Figure 11 - Fully Assembled Wrist Enclosure

The camera enclosure follows a very similar route, where a box of size (36.7mm x 40.6mm x 12mm) encases the camera-mounted PCB. Similarly, this design also has designated holes for the heatsinks and M2 fasteners, for adjoining the lid to the box. Since our PCB does not have any through holes, we added a lip to the lid such that when the enclosure is assembled, the lip will apply pressure onto the PCB resulting in no movement. Each side has a hole, one for the camera to look out of, and another to connect the PCB to the wrist enclosure, while also functioning for heat dissipation.

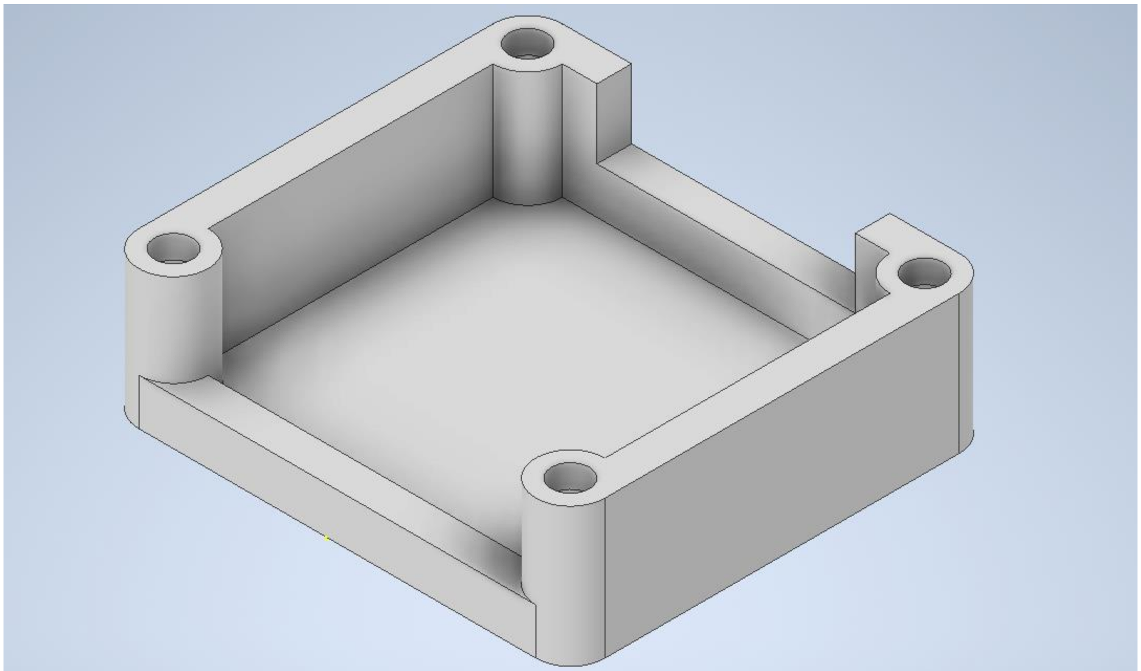


Figure 12 - CAD of Camera PCB Enclosure

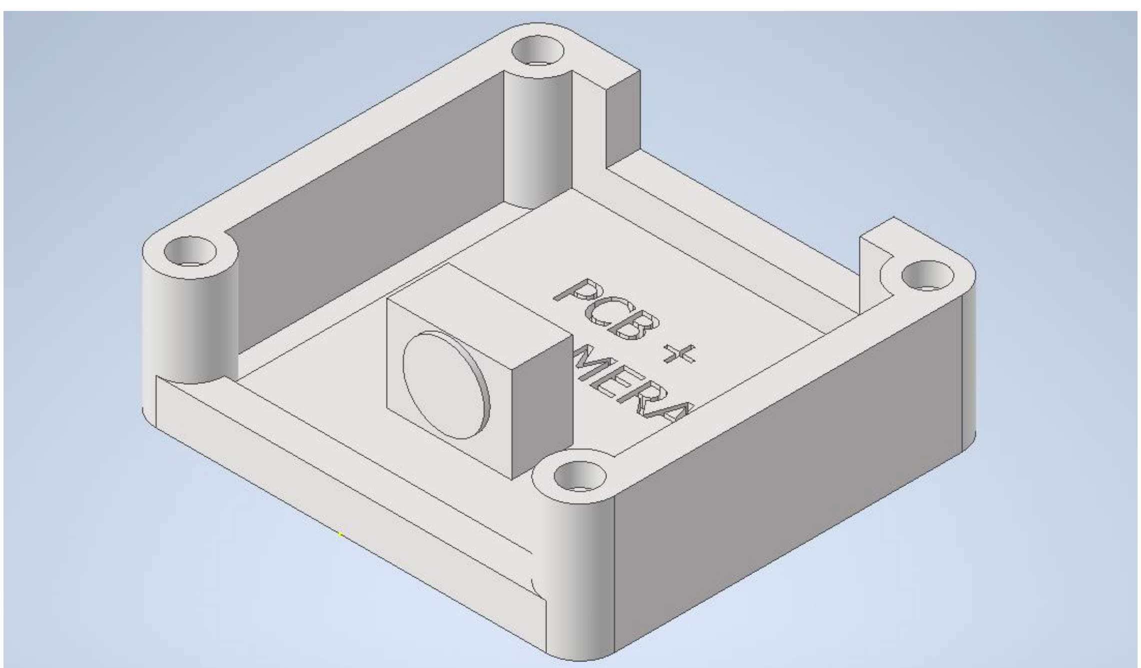


Figure 13 - Camera Enclosure with PCB

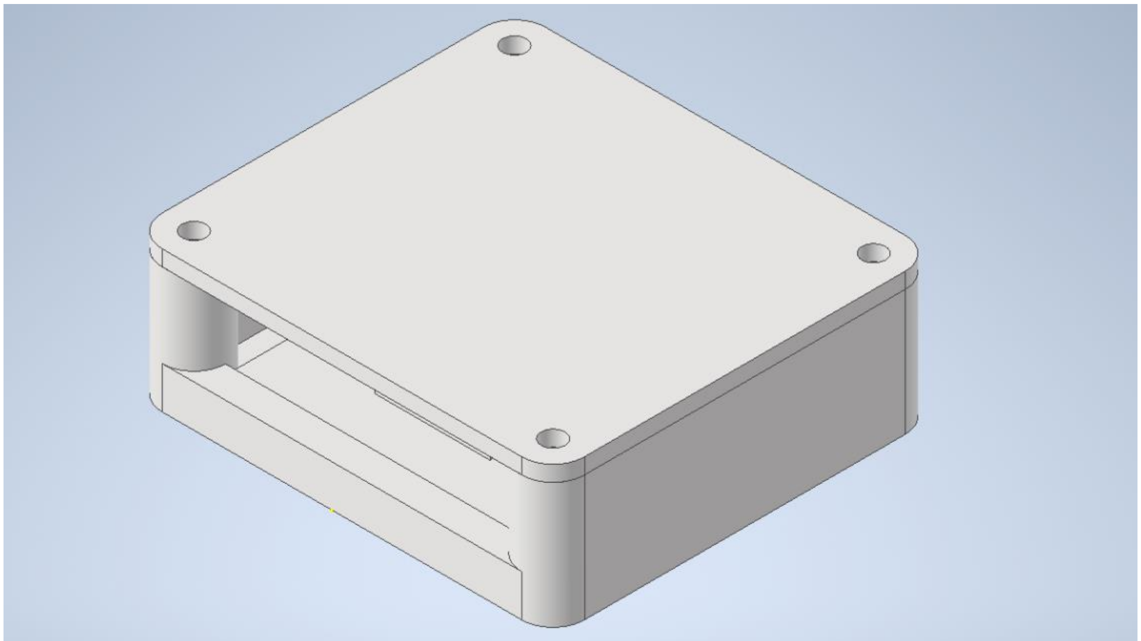


Figure 14 - Fully Assembled Camera Enclosure

Both enclosures will be securely mounted to a wristband, with a simple strapping mechanism. As seen in the figure below, the larger enclosure housing the ESP-32 sits on top of the user's wrist, and the camera enclosure sits to the left of it, with a ribbon of wires interfacing the two together. These wires will be long, providing enough slack for users with larger wrists. Initially, we decided to position the camera enclosure at the bottom of the wrist, however decided against it due to all the force being in a single direction. This positioning takes a little bit of stress off while maintaining a clear view of the LEDs below. The wristband design and strapping mechanism will be considered in the near future as we develop a better understanding of the design.

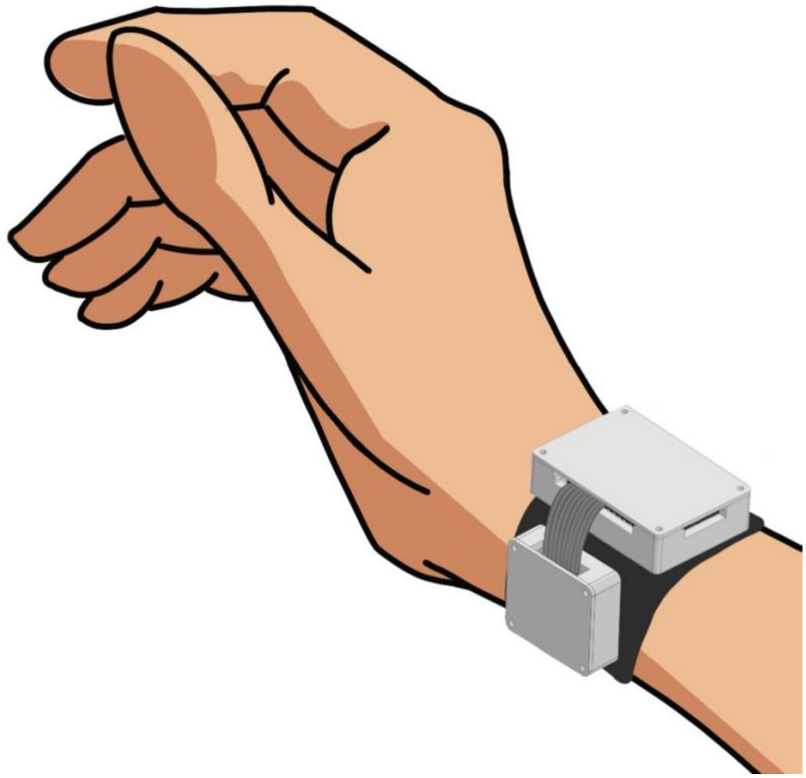


Figure 15 - Detailed Sketch of Wrist Band Design

A dedicated CAD model for the standalone LED circuit was also created using Autodesk Inventor. This model is a straightforward box with four holes in a square pattern for the IR LEDs. The distance between these was set at 100mm to ensure the LEDS do not interfere with each other. However, this distance may need to be increased/decreased depending on how the PixArt sensor interprets each instance. This enclosure holds the four IR LEDs, four resistors, and a battery. The only constraint for this model is the size, as we want to minimize the total footprint of our design for usability purposes.

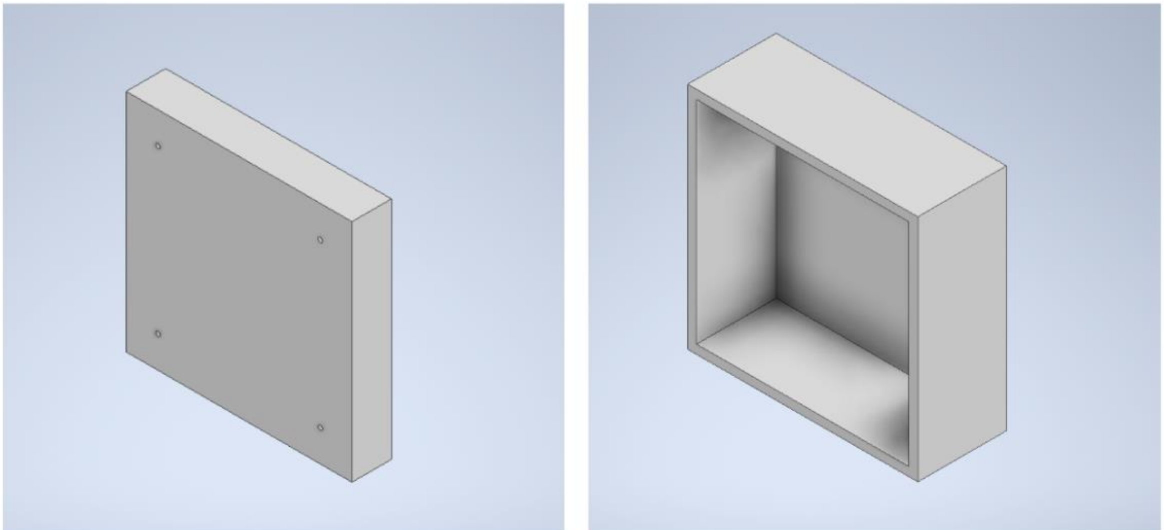


Figure 10 – CAD of IR LED Circuit Enclosure (base on right, Lid on left)

The next step is to 3D print this design and initiate testing to evaluate the requirements. This includes ergonomic, durability and comfortability testing. The feedback collected will adjust the design to meet the specified requirements more closely.

System Design

Flowchart/State Diagram

The following state diagram showcases the basic logic of the functionalities of the device. The device will keep a neutral roll angle of the wrist (as shown in Figure 15) to compare its current angle and depending on the angle deviation from the neutral roll angle, the device will enter different states to perform different functions. Scrolling can be performed via a different axis of movement, which we've picked to be the pitch the device. This means simply tilting your forearm to point at or away from the surface/desk could scroll the mouse up or down.

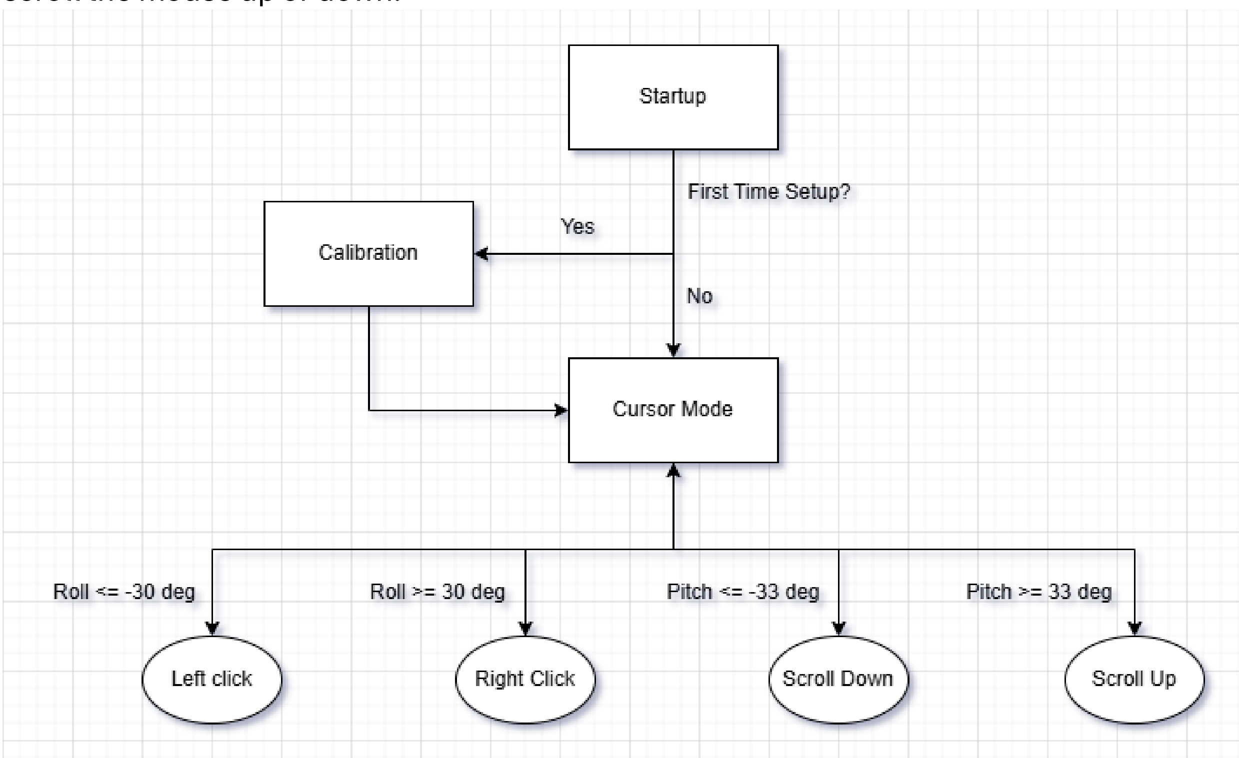


Figure 11 – Flowchart/State Diagram

The actual conditional values for the roll angles are only placeholder values currently, as more testing is needed to possibly determine better and more ergonomic values to enhance user experience. Calibration is also in a placeholder stage simple for ease of development, as the device will most likely require a calibration stage in later stages of development, as seen in the calibration of the IMU, that needs to take a base measurement first to determine the angular offsets correctly.

Monocular Point Tracking and Mouse Cursor Movement

Physical Setup and Working Principle

There are two main hardware components used to achieve mouse cursor motion: the PixArt PAJ7025R3 MOT (multiple objects tracking) sensor, and a cluster of 850nm IR LEDs. The MOT sensor contains a camera with an 800-900nm IR pass filter as well as built in hardware to track up to 16 objects. It assigns a number to each object and returns the coordinates of each point to the micro using SPI.

To obtain cursor motion, the 3D coordinates of the sensor relative to the cluster of LEDs must be obtained using only the 2D point coordinates obtained from the MOT sensor, and the intrinsic parameters of the sensor's camera. To accomplish this, multiple approaches were researched and explored. There are two projects documented on GitHub using the same MOT sensor which were used as main inspirations for this project: Retrosphere, which uses a stereo setup built into glasses to track reflective spheres for various VR and AR applications, and a 6dof demo by Bart Tryznadlowski which uses a monocular setup to model motion of a real ping pong paddle in 3D Click or tap here to enter text..

The main advantage of a stereo setup is that depth information is easier to obtain and more accurate, leading to more accuracy in the 3D tracking. In our project, a monocular approach is adopted, because the hardware setup is less complex and simpler to mount to a human wrist. Rather than using the known distance between two cameras, the known dimensions of a cluster of LEDs is used to determine camera pose in 3D. There are many approaches for achieving this that were explored, but eventually, the PnP solver in opencv was chosen [9].

This algorithm takes a set of known world coordinates (in our case the point cluster), a set of 2D points as seen in the camera, and the camera's intrinsic matrix and distortion coefficients. Then, it computes the rotation and translation required to correspond the 2D camera points to the 3D world points. This can then be used to recover the camera's pose in the world frame.

Simulation and Testing Results

As explained in the hardware design section of this report, our group faced challenges in interfacing with the PixArt MOT sensor. Thus, to test the theory required for the project, a simulation was written in Python using OpenCV. The camera is simulated using an intrinsic matrix created from the datasheet parameters of the PixArt PAJ7025R3. A 3D points cluster is created, and the points are projected onto a 2D plane using the $cv.projectPoints()$ function. Then, the camera pose is recovered using OpenCV's PnP (Perspective-n-Point) algorithms. The x and y coordinates of the camera position are then used for cursor movement, and a simulated cursor is plotted on a 1920x1080 screen. As an added feature, a function adding random gaussian noise to the camera points was also created to simulate real word noise and monitor its effects, but for the plots shown below, noise was neglected to show a clean result. This effect will be explored further while implementing the algorithm with the actual sensor.

Various point clusters were tested for accuracy of recovering the camera pose. At first, a rectangular pattern like that in Bart Tryznadlowski's project was used, however, because of the symmetry of the rectangle, mirror solutions are often returned [1]. These cases can be handled explicitly and will require more investigation. Furthermore, although the points being coplanar does decrease depth information of the object, this method does offer the advantage of a simpler setup that involves less LED occlusions (when one LED covers another LED) to consider. Because of this, this method will be the first method explored in the physical implementation, as it is the simplest to test. An example of a simulation using the rectangular structure is shown below:

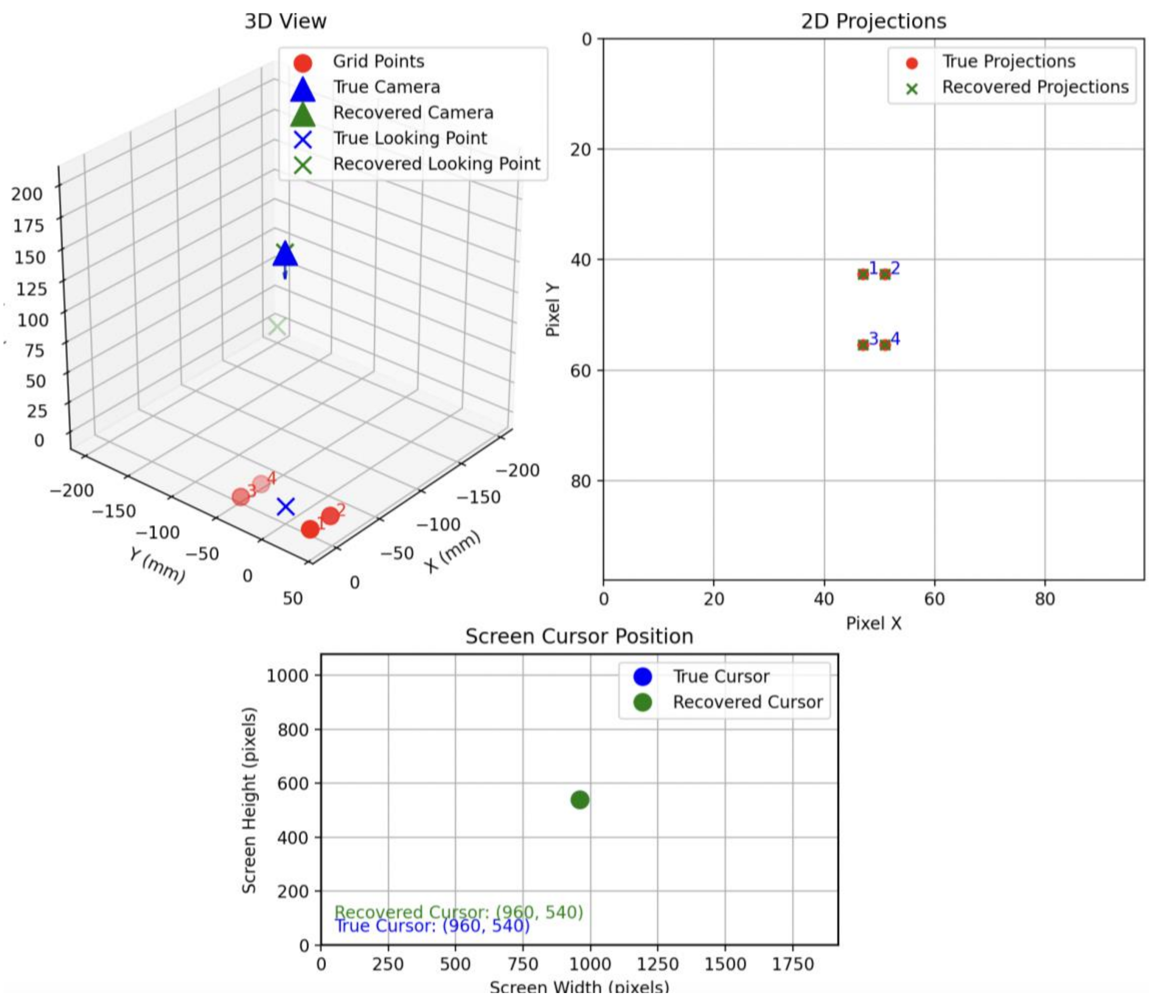


Figure 16 - Simulation Using Rectangular LED Object

To improve the accuracy, a 6 LED structure was also generated with a coplanar triangular base, and varying LED heights and locations. A 9 LED structure was also explored, using 5 coplanar points as the base and 4 other points with varied height, however the six LED configuration is preferred as the build complexity is reduced.

Varying the heights of the LEDs in the cluster helps provide depth information, but also decreases the range of motion, as occlusion occurs more frequently. Another thing to consider with the LED structure is that at lower height differences, it is more difficult to physically build, since our budget and build quality requires high error tolerance. Shown below is an example where the camera is directly above the Six LED structure:

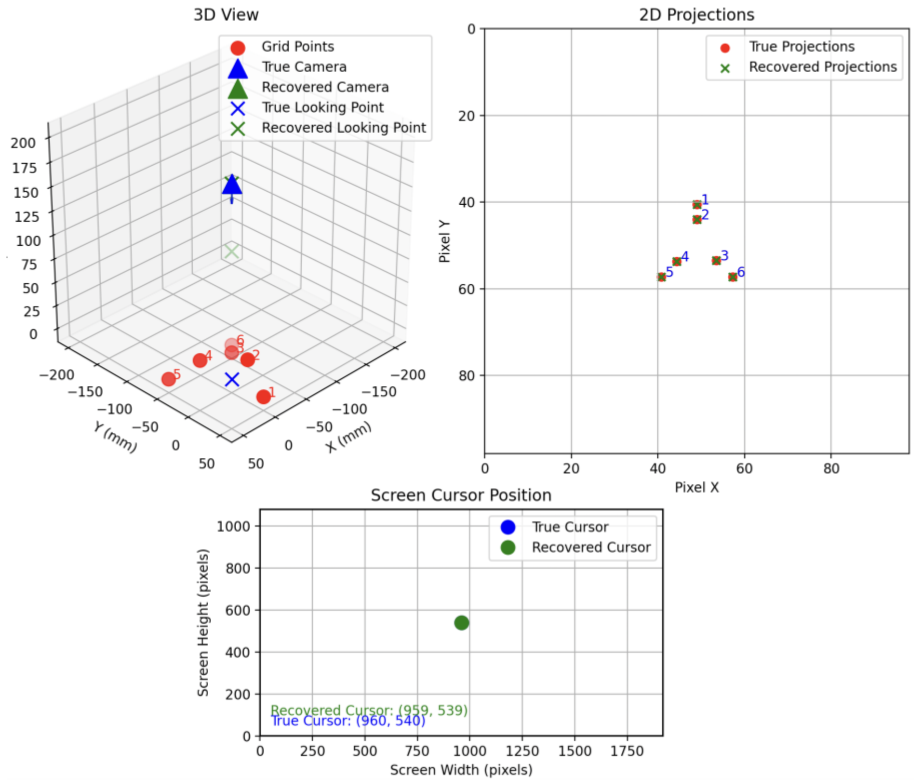


Figure 13 – Simulation Using Six-LED Object

Obstacles

There were several obstacles encountered when trying to create the simulation. First, learning about camera properties and the area of computer vision proved challenging, as this area was well outside my initial scope of knowledge and included a lot of math which was difficult to understand at first. Furthermore, methods from scratch based on the algorithms presented in various papers were attempted, such as BKP Horn's method for absolute orientation and a three-point head pose tracking algorithm for tracking LEDs mounted on glasses with a camera [10], [11]. However, this proved time consuming, and in the end, it was much more beneficial to use the OpenCV library, which contains a plethora of methods to use in both python and C++. Nonetheless, reading the literature was very beneficial overall learning, and helped in understanding the scope of what is required for the project. It may also prove useful when optimizing the computation time on the microcontroller if the OpenCV methods prove to be too computationally expensive.

One upcoming area which is anticipated to be a challenge is utilizing the methods from the simulation in the actual physical sensor and microcontroller implementation. Firstly, it may prove challenging to run on ESP32, as the computation may be time consuming enough to cause noticeable lag. Furthermore, converting the code from python to C/C++ should be feasible, but will present some differences. Ideally, all computations should be done on the microcontroller, however in a worst-case scenario, the ESP32 could simply be used to transmit data to the user's PC to perform the computations on a more robust device.

Another potential challenge is handling camera distortion. OpenCV models distortion using a polynomial with coefficients for radial (k1, k2, ..., k6) and tangential distortion (p1 and p2). From the PixArt datasheet, the -30% distortion was assumed to mean radial distortion only, so the first order radial coefficient k1 was set to -0.3. This may not necessarily be the case, and this issue can be solved by calibrating the camera to retrieve its intrinsic properties. This is typically done by taking several images of a grid pattern, however, since the PixArt sensor returns coordinates of up to 16 IR objects, a different calibration method will need to be used. This may need to be explored to accurately find the sensors position.

Next Steps

The following are the steps required to achieve a functional mouse cursor:

- Improve recovery of camera looking direction, as this becomes more inaccurate as the camera rotation increases. If this is not possible, rotation will be handled entirely with the IMU, while the PixArt sensor will handle 3D position.
- Python was used for easy plotting; however, it cannot be used on ESP32. Instead, convert to code which can run on ESP-32 to do computation on the microcontroller

(C/C++/Arduino), and interface with actual sensor and IR LEDs. Since OpenCV is a c-built library, this should be possible.

- Currently, occlusion is handled by setting limits which remain inside a zone where the LED are not occluded. Rather than avoiding occlusion, try to handle the cases where it occurs. This was done using a neural-network approach in the Retrosphere project, so it is an area which should be investigated for robustness of the design.
- The design should better handle the case where one or more LEDs are not visible to the camera.
- The case where the sensor loses track of an LED must be explicitly handled.
- The minimum amount of current through the LEDs while still producing a bright enough light shall be explored in the real implementation.
- Other methods of camera calibration compatible with the PixArt sensor should be explored and tested.

PixArt MOT Sensor Testing and 2D Visualization

Interfacing with the PixArt sensor proved more complicated than initially expected, as the sensor has numerous registers which need to be written to and read from over SPI to both initialize the sensor but also continuously read object data. Using code from Bart Trzynadlowski's code as a base, firmware was developed to communicate with the sensor using the ESP-32 and send object data over serial UART [1]. The code initializes the sensor according to explicit steps in the datasheet (our group has signed an NDA and is therefore unable to share the datasheet for the PixArt PAJ70253 sensor). The code was meticulously analyzed line by line to ensure all steps were made according to the datasheet. The code was commented to show functionality of each register and what each step does according to the datasheet.

The first testing step was to read one of the test images pre-stored in the sensor. These images were helpful to determine whether the sensor worked without needed any LED circuit. Once the sensor was confirmed to be functional, a circuit using a single OSRAM SFH 4356 860nm IR emitter was quickly made on a breadboard and supplied using a DC voltage from the AD2. Once it was determined that the sensor could detect the LED, multi-LED configurations were explored.

Mouse movement has seen its initial stages with great difficulty, however, hugely with the help of the hardware implementations, we are now able to start tracking objects with the Pixart sensor. The sensor was successfully interfaced with via SPI communication, and sensor data can now be extracted in real-time. Using this data, a 2D visualization/monitoring tool has been developed to assist in the next steps of development in conjunction with the simulations and theoretical implementations to finally bring mouse movement inputs to the device. An example is shown below of this implementation. The circuit contains four IR-LEDs in parallel with an 100ohm resistor in series with each, all supplied with 2V from the AD2. An example of one of our test setups with dimensions of 75mm x 23 mm is shown below:

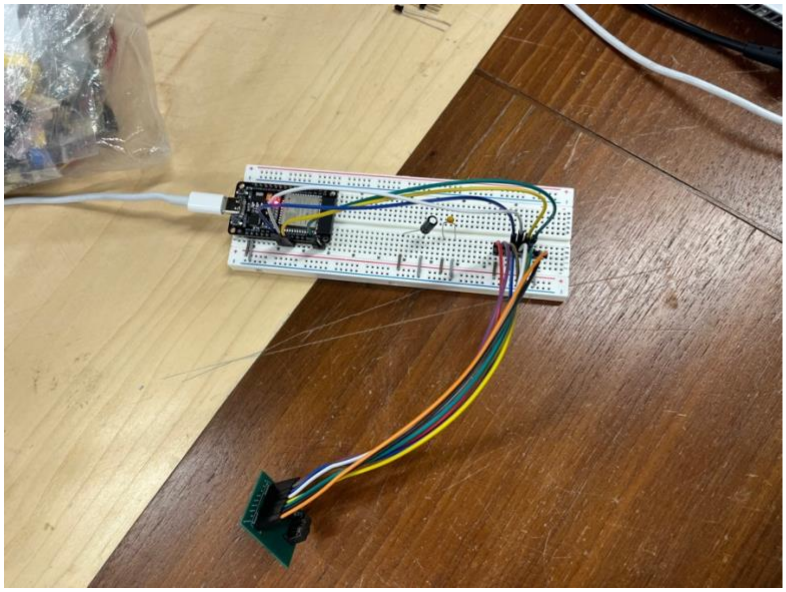


Figure 17 - PixArt Sensor and ESP32 Setup on Breadboard

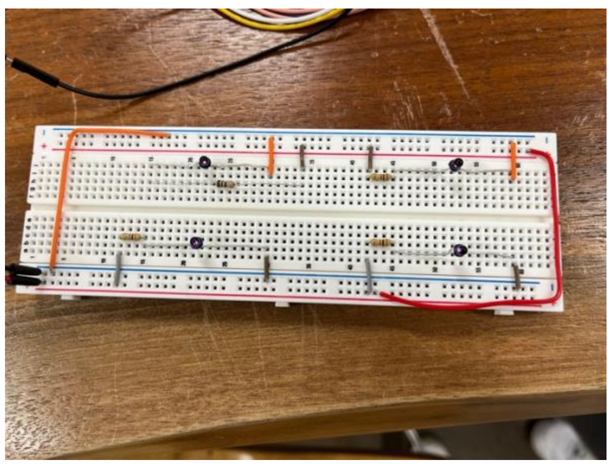


Figure 18 - Example of LED Circuit Test Setup

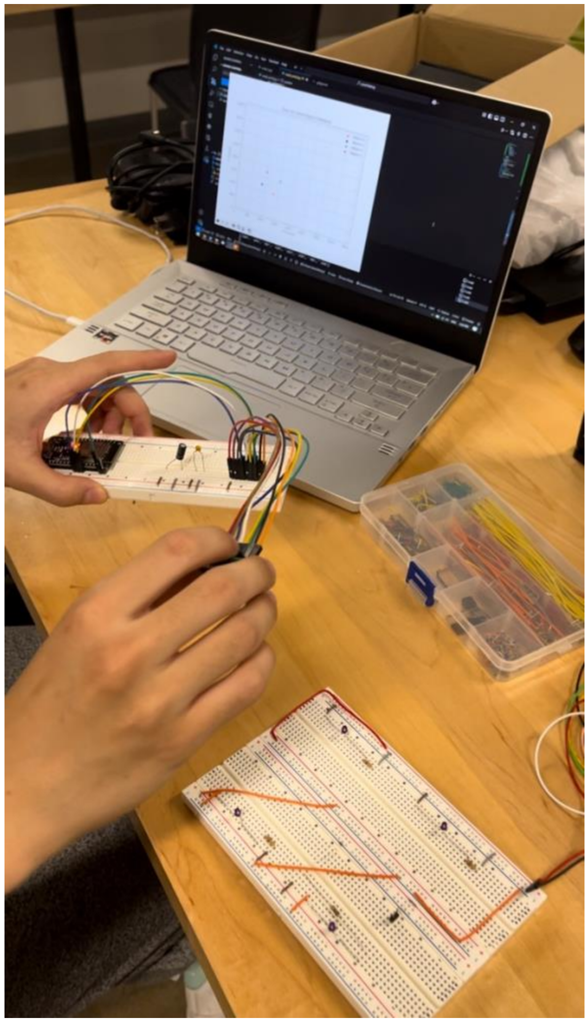


Figure 19 - Example of Complete Setup for 2D Visualization

For the ESP32 to emulate mouse inputs as an HID, the Espressif IDF toolkit has included examples of BLE usage and several HID demos, which were used in reference to implement our functionality. Note, that there also exists an existing BLE mouse emulation library for the ESP32, that could prove to be very useful for streamlining development and optimizing runtime.

Mouse Clicking Functions and Haptic Feedback

The mouse clicking and haptic feedback functions are achieved with a coin vibration motor (VCLP1020B002L) and an Inertia Measuring Unit. The selected motor is an LRA (linear resonant actuator), which is used for its compact size and higher efficiency when compared to other haptic motors such as ERM's (eccentric rotating mass) [12]. Piezoelectric actuators were also considered; however, they were found to be quite expensive, had less options to choose from, and often had abnormal shapes. The Vybronics VCLP1020B002L was selected because it has a low current requirement of 30 mA maximum, meaning it would never exceed the maximum ESP32 GPIO current rating of 40mA. This low power comes at the cost of lower vibration force at only 0.75 Grms, however this is still recognizable to the user, and we determined lower current consumption is of greater importance.

With the motors, we also included the Adafruit DRV2605 drivers, which were then soldered to the motor units to drive the motors. The motor driver has many preset functions which change the feel of the motor vibration. This was attractive to us as it opens the door for more complex feedback signals for a better user experience. The IMU is interfaced onboard the ESP32, where we extract all the sensor data to determine the current state of the device. The clicking function uses the extracted IMU data to determine when to send mouse-click inputs to the host computer.
So far, the IMU has been successfully interfaced via I2C to extract pitch, roll, and angular data from the sensor, and we were also able to mimic mouse inputs to the host computer via Bluetooth on the ESP32 using the BleMouse library.

This was tested using the IMU, Drivers/Motors, and a breadboard. The accelerometer and gyroscope values can be extracted by modifying the provided code from the Adafruit MPU6050 library (please see appendix). Initially, the gyroscope, measuring the rotational velocity along the three axes, was used for the clicking motion. Through trial and error, we noticed that measuring the gyroscope value in the y-axis at a threshold of -/+ 8m/s provided an accurate clicking motion when the wrist was quickly rotated. A right click was performed when the wrist was rotated clockwise and exceeded +8m/s, whereas a left click was performed when the wrist was rotated counter-clockwise when exceeding -8m/s. This value did not capture minimal rotation, which is ideal as we did not want to capture unwanted clicks. However, we noticed that repeated clicks may not be too ergonomic as each click required a quick jolt which may cause strain on the user. Instead, we opted to use the angle, where if the angle (measured by the roll value) exceeds a certain threshold a click is performed. If the angle exceeds -30 degrees, a left click is performed, and if the angle exceeds +30 degrees a right click is performed. If a click is performed, the motor buzzes signifying the user the click was successfully registered. The scrolling motion follows a similar pursuit using the angles captured from the pitch. Calculated using the accelerometer values, the pitch provides an angle range from -90 to +90 degrees. Thus, we set a threshold value that, if exceeded, performed a downward or upward scroll. Again, this value was found using trial and error. We settled at 33 degrees, where an upwards degree was performed if the imu was tilted to exceed +33 degrees, and a downwards scroll was performed if the imu was tiled to exceed -33 degrees. Additionally, we added a calibration function to gather 500 samples and calculate the offsets from the stationary IMU, where this offset was subtracted from values found in the actual simulation.

Bill of Materials (Matt)

The current Bill of Materials (BOM) can be found below:

Item	Desc	Cost	Tax	Total	Link
ESP32(x2)	x1	$19.99	$2.60	$22.59	ESP 32 ESP-WROOM-32 NodeMCU Development Board, USB Type-C, WLAN WiFi + Bluetooth Dual Core Microcontroller for Arduino IDE, Silicon CP2102 USB Bridge, 2PCS (30 PINS)
Motor Unit	Vibration User Feedback x5	$23.92	$3.11	$27.03	https://www.digikey.ca/en/products/detail/vybronics-inc/VCLP1020B002L/10285888
Motor Driver	x2	$23.36	$3.04	$26.40	https://www.digikey.ca/en/products/detail/adafruit-industries-llc/2305/5356831?s=N4IgTCBcDaKHAEBGArGAHAWgQZgCwDZ0A5AETgAI4wsAGJcgIQHkBBAJWLIAlmAFAFQCSAYTIBZRn0asyQvgBlpxVgDUweWiAC6AXyA
Motor Driver Cable	x4	$5.60	$0.73	$6.33	https://www.digikey.ca/en/products/detail/adafruit-industries-llc/4209/10230003
PixArt Test PCB	x5	$21.78	$2.83	$24.61	jlcpcb
Heat Resistant Tape	x1	$12.99	$1.69	$14.68	https://www.amazon.ca/dp/B07PTQ16Y8?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1
Flux Paste	x1	$13.33	$1.73	$15.06	https://www.amazon.ca/dp/B00425FUW2?ref=ppx_yo2ov_dt_b_fed_asin_title&th=1
IR Emitter (Proper Range)	x1	$5.01	$0.65	$5.66	https://look.ams-S.com/m/4963b77e9d10bec8/original/SFH-4356.pdf
PixArt PAJ7025R3 Sample Kit	x5	$104.25	$13.55	$117.80	https://www.codico.com/en/paj7025r3-sample-kit-cdc

Datasheets (Ryan)

- ESP32: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- VCLP1020B002L Coin Vibration Motor: https://www.vybronics.com/wp-content/uploads/datasheet-files/Vybronics-VCLP1020B002L-datasheet.pdf
- Adafruit DRV2605 Motor Driver: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-drv2605-haptic-controller-breakout.pdf

- MPU6050 IMU: https://compoindia.com/wp-content/uploads/2021/01/datasheet.pdf
- PixArt PAJ7025R3: *Confidential*
- Infrared LED: https://look.ams-osram.com/m/4963b77e9d10bec8/original/SFH-4356.pdf

The following are components which have not yet been ordered:

- Potential Battery: https://www.amazon.ca/Rechargeable-Suitable-Electronic-Products-Replacement/dp/B0CKRCXNP9/ref=sr_1_1?crid=37WAF8XO91G74&dib=eyJ2IjoiMSJ9.hbWp8DFW8W9b19aw6TSWTpnq2AX_T-oiXZ3mtF2Qrm2vp50NdYIbE0SmXewNYC9yMW05aa6jvLl70Q24ZW95WgOVPmCri3zTMR6RcJwUxuynwSAw_wGPq9B_RsXpgfP8z8RPKdNjlh_aHtPfoTGxkxAh0GLaUibtfQrLGosQsuE-D065zHPz3rkfCp8UArc1JxNUlKMYPn6rSLQ3MvSJA0t4KM5-4e08J1OSIdO0AneLPXK6us4alcaBHE_lgPtsAxD5HeTC3Bei3dVqBxSNGNefayFPEDrpm3McVZ8DTyPDugPa_Ww1KDqsAeDULA1cgos419kICLzPEQyjdPxH2gOAQghrcDb1NdrFRrcVqLY.lNyNYMLdU74FgF_WuGsXiWqDmGCkLaQjUWIixzw2N34&dib_tag=se&keywords=550+maH+lipoly+battery&qid=1737157342&s=electronics&sprefix=550+mah+lipoly+battery%2Celectronics%2C109&sr=1-1
- Firebeetle 2 microcontroller: https://www.dfrobot.com/product-2837.html

Prediction for Future Progress (George)

As the updated Gantt Chart illustrates, we were impacted by a few unforeseen challenges. These challenges had set us off track with the initial schedule; however, these challenges have been addressed and should bring us back on course. In our current state, we believe we have completed a third of the overall project, where the core elements of our project have been developed, as the purpose, design, and materials are now firmly in place.

The next step is to assemble the hardware and interface it with the software to develop a rudimentary prototype. Providing us with the ability to perform a wide range of tests, from functionality testing to ergonomic testing, over the next few months. We predict the design will be refined multiple times until an optimal design that satisfies all our requirements is found. Additionally, if time allows, we want to develop a GUI to interface with the device, enabling more customization for the user's preferences.

As this project proceeds, we may face more setbacks, such as interfacing the software with the PixArt sensor. To be cautious, we have the IMU as a fallback if significant problems arise with the other sensor.

Self-Assessment

Performance ratings use the following three levels: excellent, satisfactory, or room for improvement)

George Gill (gillg62)

Tasks and Roles

- Researched and contributed to the general selection of materials.
- Researched different IMUs and helped in the selection for our purpose.
- Assisted in creating conceptual designs for the housing of the sensors and controllers with Matt.
- Developed CAD Models and Assemblies, using Autodesk inventor
  - Ensured precision and functionality, while adhering to constraints.
- Assist in 3D printing with Matt.
- Assisted in creating the rough electrical schematic with Luke.
  - Used KiCAD to design the final schematic.
- Integrated the IMU, Haptics, and Microcontroller to enable mouse functions such as scrolling and clicking.
  - Performed research on different methods of capturing consistent motions.
  - Programmed the captured motions into mouse functions.
  - Debugging

Performance Assessment

Overall Self-rating: Satisfactory

- I have actively contributed to the development of our project. Assisting in various places but have mostly focused on the layout, interfacing with the IMU and Haptics, and the physical modeling of our design. These contributions have been beneficial to the overall project, bringing us closer to the final solution. However, there is room for improvement by spending more time on these, or future activities. Since my semester has a lighter course load than last term, I plan on spending more time helping the group conclude this project.

Matthew Yu (yum77)

Tasks and Roles

- Research and develop custom PCB for sensor using KiCad.
  - Use KiCad to design schematic diagrams, custom footprints, PCB layouts.
- Redesign custom PCB based on feedback of components needed to be removed or added to PCB.
  - Incorporate team feedback to adjust PCB design.
  - Revise component selection and layout based on function and size constraints.

- Selection of optimal capacitors, connectors, and cables needed for our systems design.
    - Verify compatibility with PCB design and system requirements.
- Reach out to PCB manufacturers to ensure PCB designs can be properly made.
- Research methods for soldering our sensor and other small electrical components.
- Solder sensor to custom PCB and solder any other components needed such as motor driver etc.
- Verify that all solder connections are properly made, the solder joints are of good quality, and no short circuits are present.
- Assist George in prototyping 3d printed housing for sensor and controllers.
- Start 3D printing and assembling physical prototypes along with George.
- Collaborate with team in discussing different ways sensor can be mounted along with input on how to arrange all components to fit in enclosure for lightest and smallest design.

Performance Assessment

Overall Self-rating: Satisfactory

- I put a lot of time and effort into learning how to design PCBs, including using KiCAD for creating custom schematics, footprints, and layouts. I also focused on improving my skills in soldering small SMD components, making sure everything was put together accurately. In addition, I learned how to test and troubleshoot components to quickly identify and fix any issues. Although I successfully completed many important tasks on my own, I see room for improvement in working more closely with the team. I recognize that my focus on independent work limited my opportunities for collaboration with my groupmates. I believe that more active collaboration could have opened up new perspectives and generated better ideas. By sharing knowledge and brainstorming together, the team could have benefited from different points of view, which would have enhanced our overall design and approach. In the future, I plan to make a more concerted effort to engage with my teammates, exchange ideas, and seek feedback throughout the process. I believe that increased teamwork will not only improve my individual contributions but will also lead to more innovative and well-rounded solutions for the project.

Ryan Xu (xur76)

Tasks and Roles

- Research and acquisition of key components of the project including the microcontroller, IMU, and PixArt sensor.
- Logo Design and overall branding of the project.

- Established communications with PixArt to obtain sample sensors for initial prototyping and development, along with negotiations regarding documentation confidentiality.
- Conceptual development of early-stage system/stack design with the team.
- Assisted Luke in some parts of the development of the python simulations.
- Assisted in the development of the software interfacing process with the Pixart sensor.
- Development of the 2D IR LED object tracking visualization tool.
- Research and early-stage idealizations of user-software UI/UX (Low Priority)

Performance Assessment

Overall self-rating: Satisfactory

- Higher productivity near the earlier stages of the project, trailing off in time investments near the time of writing this report. Non-consistent efforts, although meeting demands with good quality in bursts instead of a consistent workflow. Improvements must be made in more consistent work ethics and proactiveness to keep up with and push the speed of development further.

Luke West (westl5)

Tasks and Roles

- Research and development of Python simulation to model the PixArt sensor and test algorithms for localization of the camera with respect to reference world points
- Testing of communication with physical PixArt sensor over SPI alongside Ryan
- Helped Ryan with test 2D visual implementation of IR LED object tracking
- Visualizations of physical wrist band implementation, using George's 3D models to develop concept sketches
- Development of initial schematic alongside George ensuring pin for pin correspondence with the components
- Selection of haptic motor driver and motor components
- Selection of IR LEDs
- Selection of more power efficient ESP-32 board option (Firebeetle 2), and investigation into battery types, required power conversion and charging
- Communication with vendors and McMaster to secure additional PixArt sensors
- General support to the team in discussions and meetings involving project design

Performance Assessment

Overall self-rating: Satisfactory to Excellent

- There was a lot of time which went into researching the math for the simulation, and cameras in general, it took a while for me to even find the right resources to learn from as there is a lot available online, and it is an area I previously did not know a lot about. The python simulation which I built is sufficient for proof of concept and the visualizations were helpful in determining how our project could work. However, it still does not consider all cases and the algorithm for localization will need to be improved in the actual implementation. Although we were unable to interface with the PixArt sensor until recently, we were able to achieve communication with it using SPI which enabled us to visualize the tracked objects in 2D. I also provided help to groupmates when needed for various aspects of the design, such as the schematic and physical implementation, and set most meeting times for the group at regular intervals throughout the term. In terms of my own effort, I put in spread out short bursts of concentrated work throughout the last term. This work style led to some added stress as I would focus on my other classes then switch to high intensity focus on the capstone project when I started to feel behind. Looking forward to this term, I will be aiming to work more consistently each week rather than in short intense bursts, as I believe this will reduce stress, and improve productivity toward the project.

References

[1] B. Trzynadlowski, “PixArt PAJ7025R2 6dof Tracking Demo with Arduino and Windows,” GitHub. Accessed: Jan. 03, 2025. [Online]. Available: https://github.com/trzy?tab=repositories&q=PixArt&type=&language=&sort=

[2] “Types of Capacitors: Pros, Cons, & Applications,” Circuit Crush. Accessed: Jan. 16, 2025. [Online]. Available: https://www.circuitcrush.com/types-of-capacitors-applications/

[3] “What is the Difference Between FFC and FPC? ,” Quadrangle Products. Accessed: Jan. 16, 2025. [Online]. Available: https://www.quadrangleproducts.com/what-is-the-difference-between-ffc-and-fpc/

[4] “FFC vs FPC: Understanding Flexible Cable Technologies,” Newhaven Display International. Accessed: Jan. 16, 2025. [Online]. Available: https://newhavendisplay.com/blog/ffc-vs-fpc-understanding-flexible-cable-technologies/

[5] A. Brown, “Plated vs Non-Plated PCB Through Holes in PCB Desigs,” epec Engineered Technologies. Accessed: Jan. 16, 2025. [Online]. Available: https://blog.epectec.com/plated-vs-non-plated-pcb-through-holes-in-pcb-designs

[6] “What is a FPC/FFC Connector?,” Omron. Accessed: Jan. 16, 2025. [Online]. Available: https://components.omron.com/eu-en/products/connectors/board-to-fpc-connectors/board-to-fpc-connector_features?utm_source=chatgpt.com

[7] “ESP32 Thing Hookup Guide,” Sparkfun. Accessed: Jan. 16, 2025. [Online]. Available: https://learn.sparkfun.com/tutorials/esp32-thing-hookup-guide/hardware-overview#:~:text=The%20ESP32%20can%20pull%20as,while%20actively%20transmitting%20over%20WiFi.

[8] A. N. Balaji, C. Kimber, D. Li, S. Wu, R. Du, and D. Kim, “RetroSphere,” *Proc ACM Interact Mob Wearable Ubiquitous Technol*, vol. 6, no. 4, pp. 1–36, Dec. 2022, doi: 10.1145/3569479.

[9] “Perspective-n-Point (PnP) pose computation,” OpenCV. Accessed: Jan. 03, 2025. [Online]. Available: https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html

[10] B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *Journal of the Optical Society of America A*, vol. 4, no. 4, p. 629, Apr. 1987, doi: 10.1364/JOSAA.4.000629.

[11] S. Meers, K. Ward, and I. Piper, “Simple, robust and accurate head-pose tracking using a single camera,” Jan. 2006, [Online]. Available: https://ro.uow.edu.au/articles/conference_contribution/Simple_robust_and_accurate_head-pose_tracking_using_a_single_camera/27792744

[12] “Haptic Technology 101: A Beginner’s Guide to the Different Types of Vibration Motors,” TItan Haptics. Accessed: Jan. 16, 2025. [Online]. Available: https://titanhaptics.com/haptic-technology-101-a-beginners-guide-to-the-different-types-of-vibration-motors/

Appendix

KiCAD: PixArt Custom PCB Iteration 1

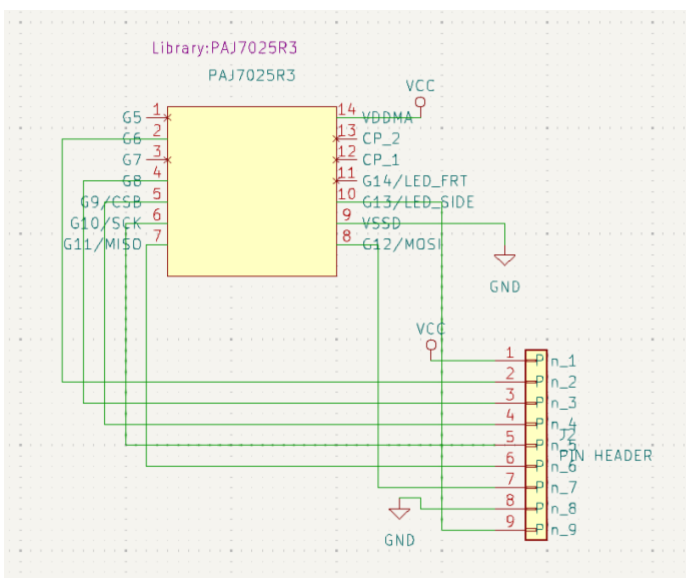
Custom Symbol

Description: The custom symbol was created to represent the PixArt PAJ7025R3 sensor in the schematic. Since the sensor did not have a pre-existing symbol in the KiCad library, it was necessary to design a unique symbol for it.
Purpose: This custom symbol allows for easy integration of the sensor into the schematic, ensuring accurate representation and connectivity in the design.
Details: The symbol includes all relevant pins and is designed to match the pinout of the PixArt sensor.

Custom Schematic

Description: The schematic diagram was designed using the custom symbol and other necessary components. It illustrates the wiring and connections between the PixArt sensor, ESP32 controller, motor units, and other key components.

Purpose: The schematic serves as a blueprint for the entire circuit, ensuring that each component is connected correctly and will function as intended. Schematic will also generate a netlist to ensure correct connections in the PCB editor.

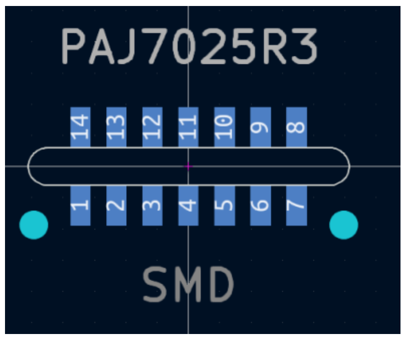


Custom Footprints

Description: The custom footprint was created for the PixArt PAJ7025R3 sensor, as it did not have a predefined footprint in the KiCad library. This footprint defines the physical layout of the sensor's pins and pads for the PCB design.

Purpose: The footprint ensures that the PixArt sensor can be accurately placed on the custom PCB, with the appropriate pad sizes and distances.

Details: The footprint includes through-hole pads for the sensor's connections, with proper alignment and size based on the sensor's specifications.



Description: Custom pin header found online

Purpose: Connect traces to external components such as capacitors and microcontroller.

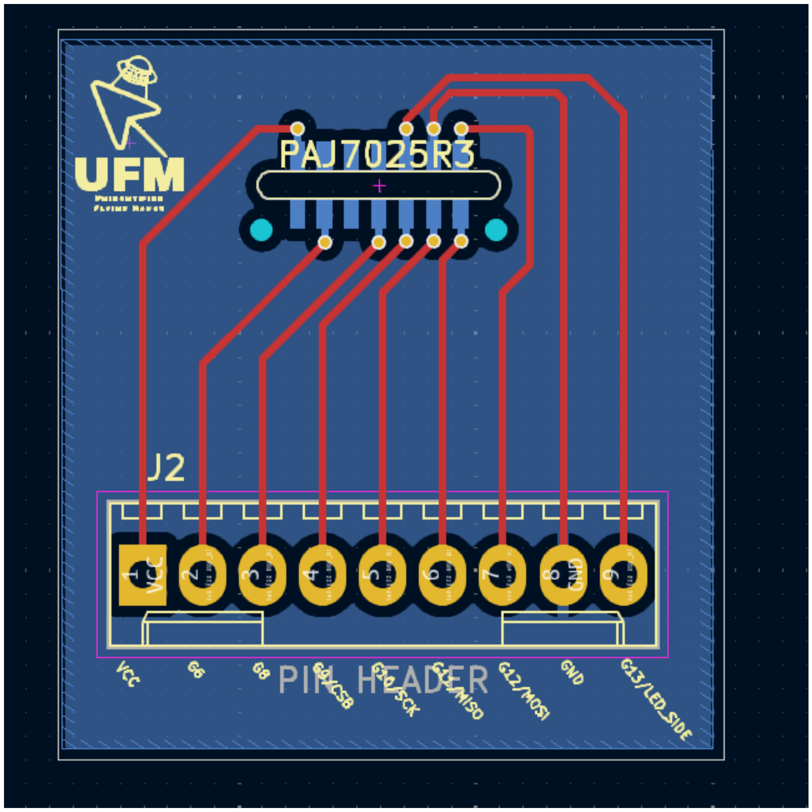
Details: Taken from KiCad GitHub: https://github.com/KiCad/Connectors_Molex.pretty

Custom PCB

Description: The final custom PCB design integrates all components, including the PixArt sensor, ESP32, motor drivers, and other critical elements, onto a compact 2-layer board.

Purpose: This PCB physically supports and connects all components, allowing them to work together in the system. It was designed to ensure that the PixArt sensor could interface properly with the rest of the system.

Details: The PCB was designed in KiCad with careful attention to trace routing, component placement, and electrical considerations like power distribution and signal integrity.

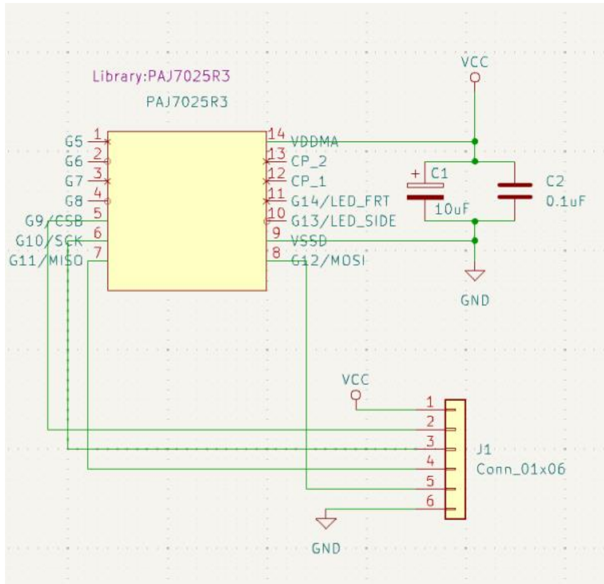


This section provides a detailed breakdown of the KiCad design process for the custom PCB. By creating the custom symbol, schematic, footprint, and final PCB, we were able to successfully integrate the PixArt PAJ7025R3 sensor into our system, overcoming the challenges posed by the lack of predefined components.

KiCad: PixArt Custom PCB Iteration 2

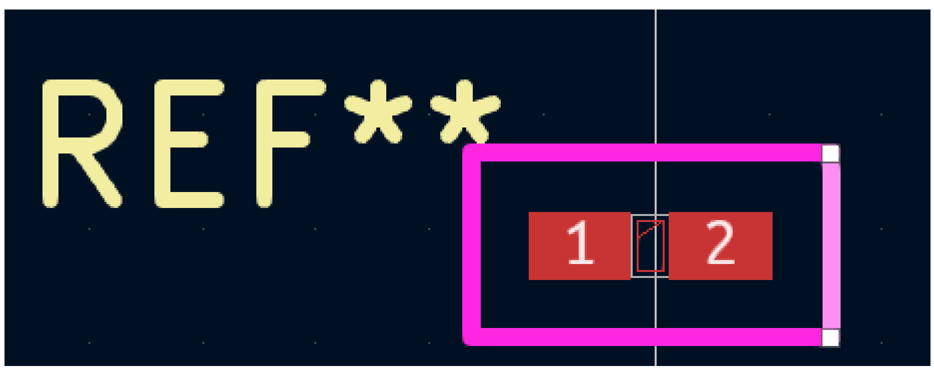
Redesign Schematic

Description: The schematic diagram was edited to reduce the number of pins, add the 0.1uF and 10uF capacitors, and change the connector to a 6 pin one.

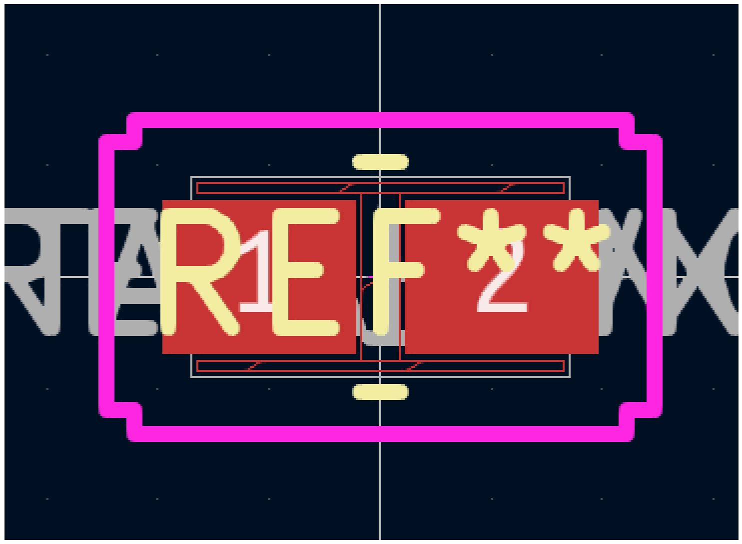


Added Footprints

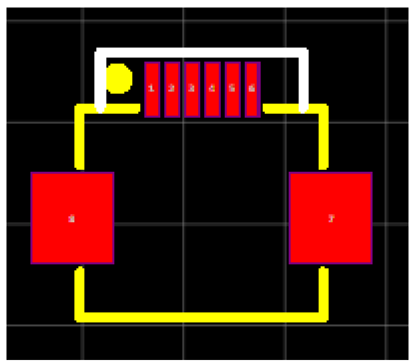
Description: Footprints and 3D models provided by Digikey based on part numbers were used to accurately represent electrical components needed to be placed on the capacitor.



0.1 uF Capacitor



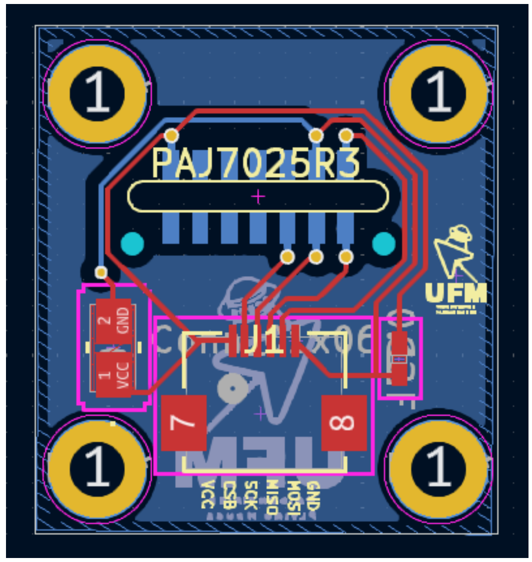
10 uF capacitor



FFC connector

Redesign PCB

Description: The final custom PCB design integrates all components needed to be changed or added from the 1st iteration.



Repositories

Main UFM Repository: https://github.com/westl5/UFM

- All code can be found in the *code* directory
- All PCB design information can be found in the *pcb* directory
- Other relevant documents such as datasheets, photos and videos can be found in the *docs* directory