OmniMouse

Final Presentation

An ECE 445 Senior Design Project by Group 74
Jie Jia, Yifei Li, and Zhengfeng Wu

The Inspiration

As Computer and Electrical Engineering students, we use our computers on a daily basis. Oftentimes, we interface with the computer through a mouse and keyboard. We have experienced that extended use of our mice has brought a lot of strain and pain on our hands and wrists.

Source: Logitech

Source: Delux

Objective

Our goal is to create a mouse that minimizes strain and injury for the user through varied non-repetitive motions and lessened hand pressures. We plan to address the various points of stress for using a mouse individually to create a product that will help prevent and reduce the prevalence of stress due to repetitive mouse use.

Source: ThePurlingCat.com

What's the OmniMouse?

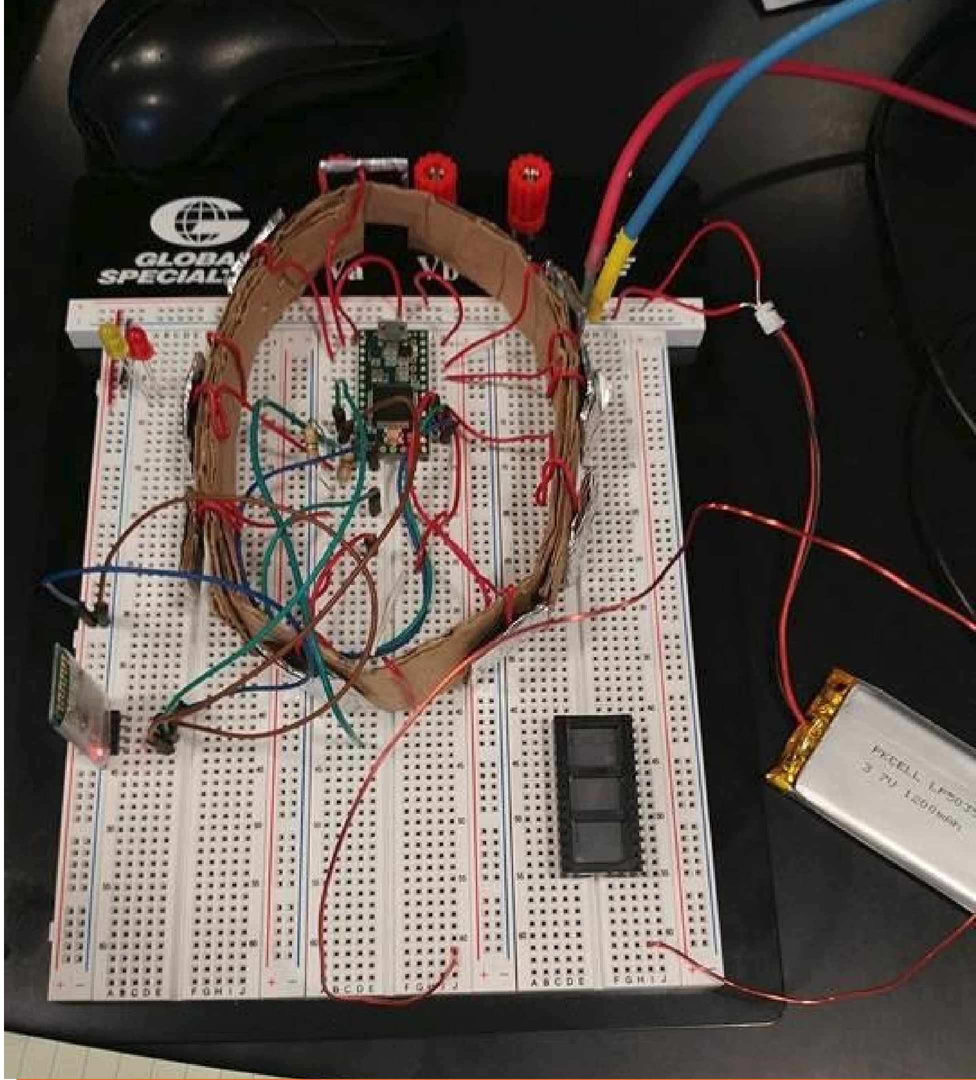
- OmniMouse is a new type of mouse device that uses capacitive sensors to sense the user's hand and give input to the computer.
- We place these sensors circularly, processing the user input into mouse movement.
- Using this method, we can eliminate a lot of the movement in the hand/arm of traditional mouses, therefore decreasing the amount of stress placed on the wrist.

What should it do?

- Process data at > 60hz for smooth mouse input
- Allow button input for click/scroll events
- Have both Bluetooth and USB connections
- Be powered off a battery, lasting at least 4 hours
- Be a functional alternative of a standard computer mouse.

Omnimouse Prototype

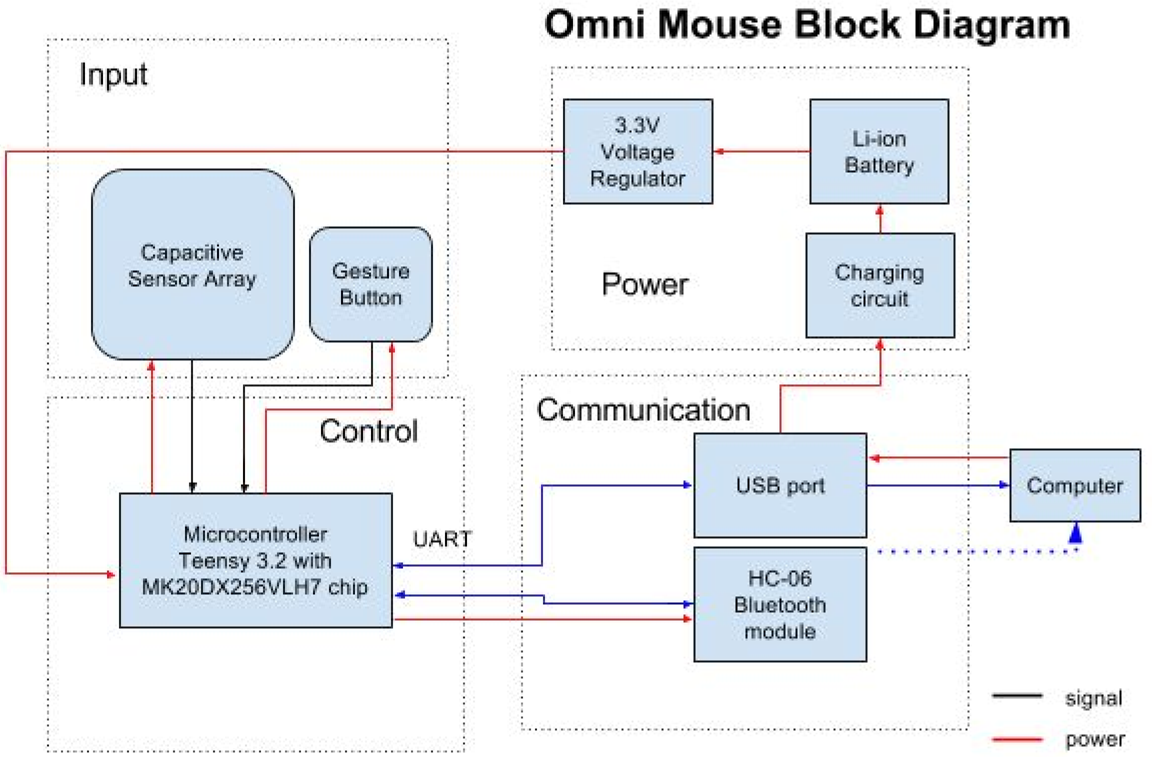
- Cardboard squares wrapped with tinfoil as sensors.
- Powered via USB
- Proof of concept to test the idea



The Final Omnimouse

- Acrylic Outer Shell
- Hard Foam Base
- Inner Mounted Copper Sensors
- USB + Bluetooth Connectivity

Block Diagram

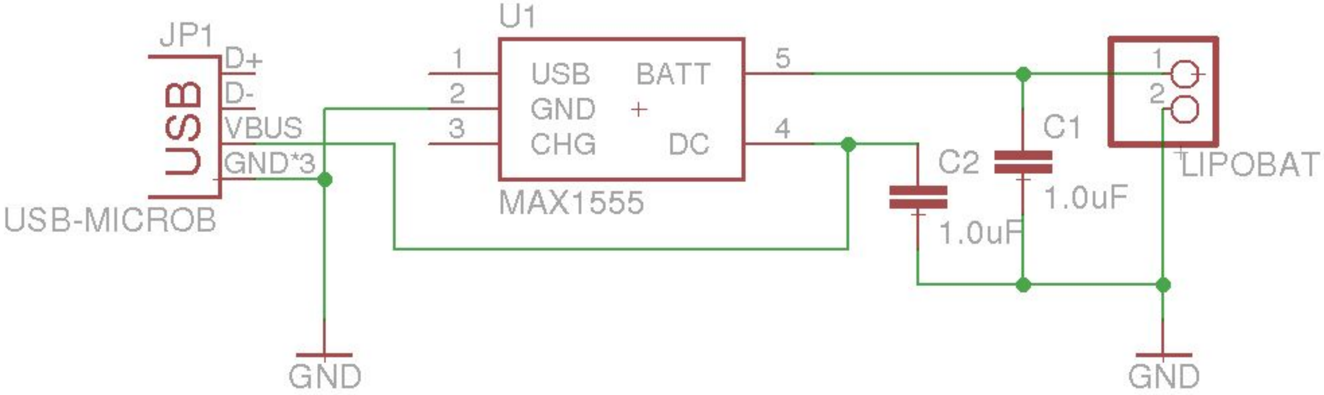


Block Diagram Descriptions

Our design has 4 main components:

- Power: Battery, charging circuit, and protection circuit. Ensures safe operation of the lithium ion battery.
- Input: 8 sensor array and gesture button. Gives input from our user into our controller and finally to the connected computer.
- Communication: We can communicate with either USB, or Bluetooth.
- Control: We utilize the Teensy 3.2 microcontroller to process sensor data and send the data to the computer for processing.

USB Charging Circuit



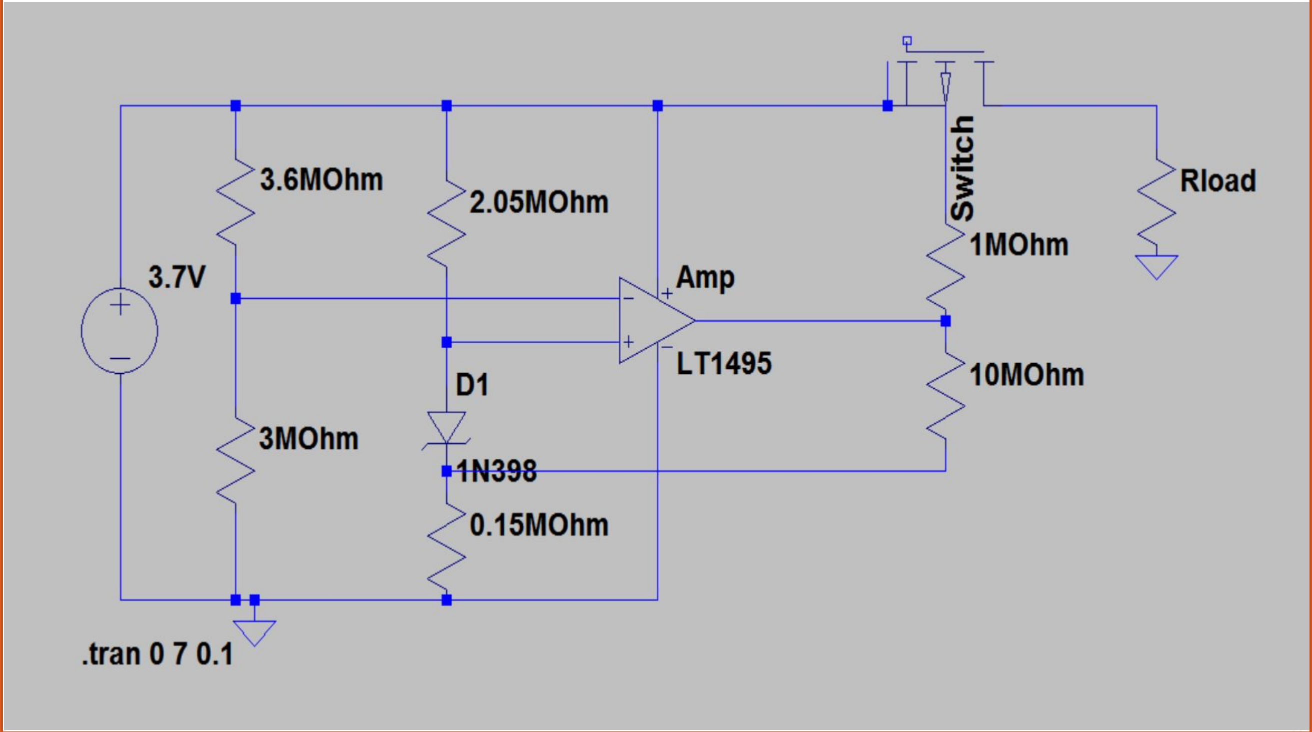
Power Consumption

Design draws 64 mA on average from our Li-Ion Battery. The battery has a capacity of 1200 mAh, giving us:

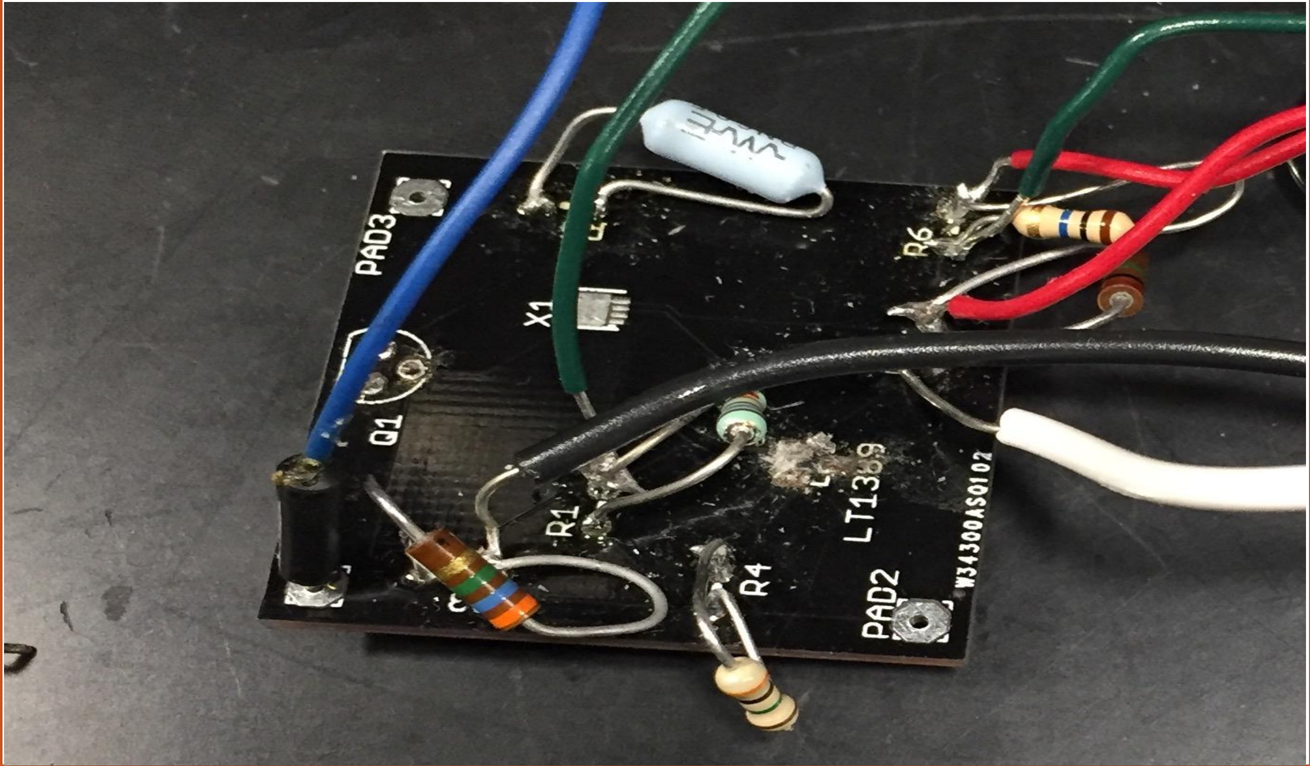
$$1200mAh/64mA=18.75hrs$$

We expect a total on-time of almost 19 hours! Much higher than expected.

Power Protection Circuit

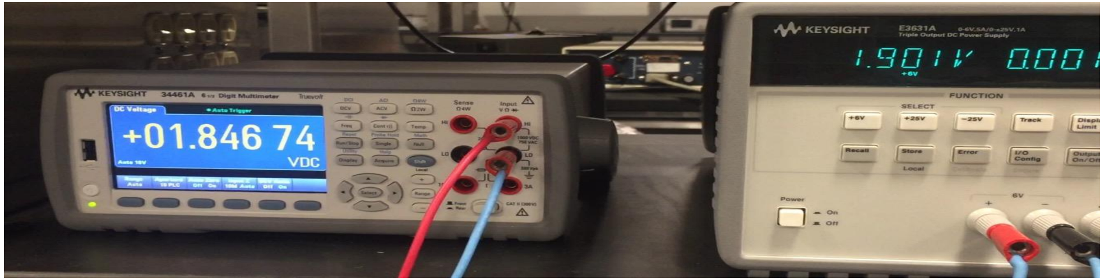


Power Protection Circuit

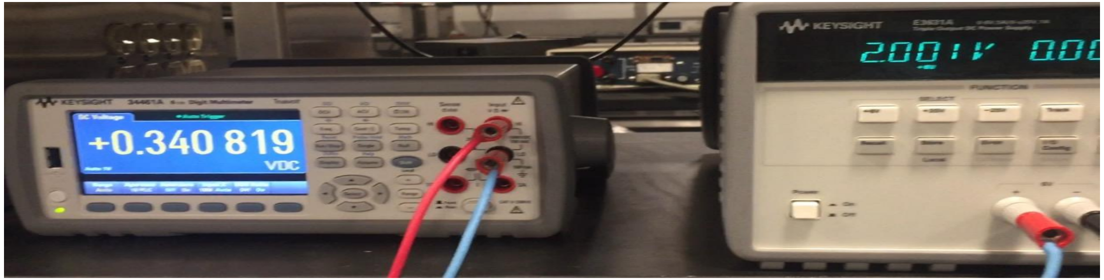


Results

Switch off:



Switch on:

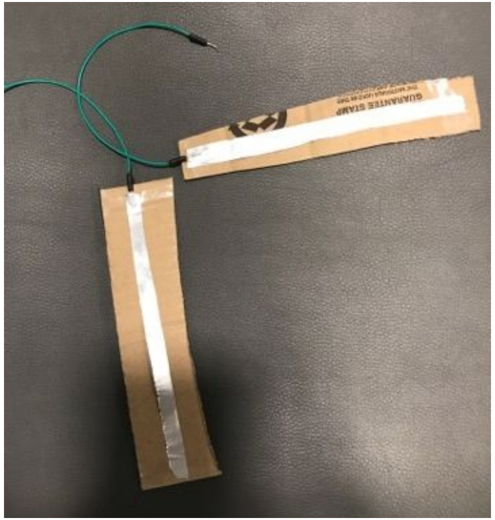


Hardware Practical Concerns

- Chip size:
  surface-mount vs through-hole
- Switch: mosfet selection

Sensor Design

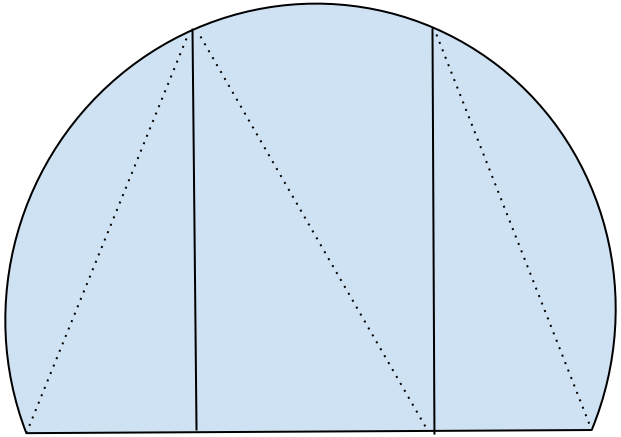
Design 1: Bar Sensor



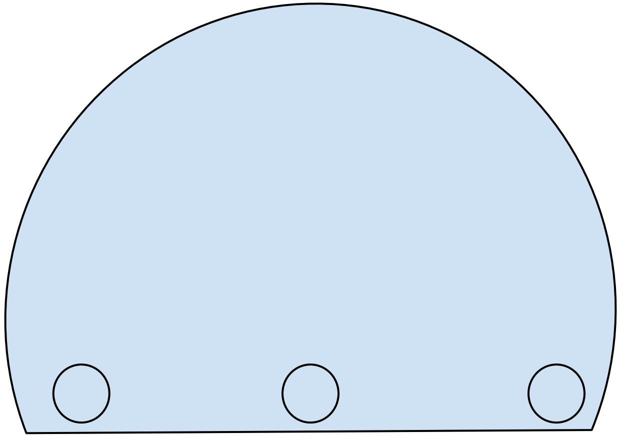
Design 2: X-Y Strip Sensor

Design 3: Triangular Strips

Sensor Design



Vertical Triangle Strips



Dot Array

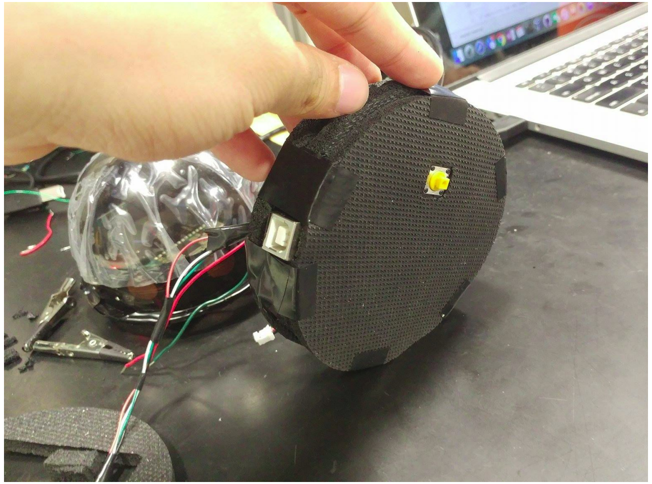
Sensor Algorithm

- For a single input, calculates the position and angle of the sensor.

- For two inputs, determines On/Off and calculates capacitive readings at the same time. Find the angle based on the position difference of sensors weighted by its capacitive reading.

- For multiple inputs, record capacitive readings in priority queue. Select three sensors with largest readings and perform two-input algorithm on each pair of sensor. Then combine outputs and calculate the angle.

Communication



OmniMouse USB-B and Bottom Mounted Button

HC-06 (Source: Amazon.com)

Sensor Packets

Touched sensor: first, second, third,
Angle: angle in deg
Direction: dirX,dirY,
Button/Gesture: O (No input)
L (Left click)
M (Middle click)
R (Right click)
D (Down scroll)
U (Up scroll)
Sensor Binary reading: S1, S2, S3, S4, S5, S6, S7, S8,

Challenges

- **Sensors**
  Sensors and each exposed point (solder point, bare wire, etc) was very susceptible to interference that caused errors in data.
- **Power**
  The surface-mount chips we chose were difficult to solder onto the PCB due to their size.
- **Bluetooth**
  We ran into a lot of issues with our bluetooth module. Its a basic module, which lacks a lot of features such as a Bluetooth HID profile.

Solutions

- ➔ **Sensors**
  Thoroughly insulated all wires and solder points, as well as positioning wires to be the best distance away from each other. Tweaked sensitivity settings and our code to compensate.
- ➔ **Power**
  Tried and tested a few boards. Ended up using a chip adapter which essentially "stretches" the chip legs.
- ➔ **Bluetooth**
  Our current implementation created a user-space mouse driver using Python. It takes in each of our packets from the mouse and processes them into mouse movements for the user.

Original Parts List:

Teensy 3.2 Microcontroller
1200 mAh Li-Ion battery
LD1117 3.3v Voltage Regulator
12x Copper Sensor Units
12x Acrylic Caps for Sensors
Acrylic Mouse Shell
HC-06 Bluetooth Module
MAX1555 USB Charging Chip
Omron B3F-4055
Top Layer PCB (Sensors, Modules)
Charging PCB
Power Protection PCB
Various Resistors, Capacitors, Wires, etc.

Final Parts List:

Teensy 3.2 Microcontroller
1200 mAh Li-Ion battery
LD1117 3.3v Voltage Regulator
12x Copper Sensor Units
Acrylic Mouse Shell
HC-06 Bluetooth Module
2 Sheets of Hard Foam (ECE-Store)
USB-B Female Port
USB-B to USB-A Cable
Micro USB-B Male Header
MAX1555 USB Charging Chip
Omron B3F-4055
Top Layer PCB (Sensors, Modules)
Charging PCB
Power Protection PCB
Various Resistors, Capacitors, Wires, etc.

What Changed?

- We can only connect one power source, either MicroUSB or the GND and VCC pins. Due to this, we had to separate the power and data; we used USB-B that splits our power and data from the computer into just data going into the Teensy, and power going into the power circuit.

- We realized the acrylic caps, originally used as a dielectric, was unnecessary since the dome sufficed for that purpose.

- Our PCB + Components took a little more space than expected, so we constructed a base for the mouse out of foam sheets, cut to form a layered circular base.

Results

- Both USB and Bluetooth Operational
- Handles input like Left/Right click and Scrolling Up/Down
- USB mode operates at 62.5 Hz (measured). Wireless mode operates at around 20Hz.
- Battery can be charged under USB mode and lasts over 18 hrs in battery mode.

Issues

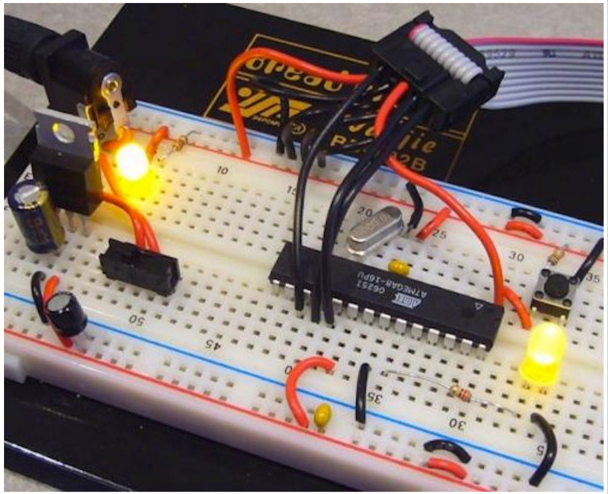
- Python can’t process sensor data quickly enough to get smooth movement.
- Button is sometimes hard to click.
- Somewhat of a learning curve

Future Work

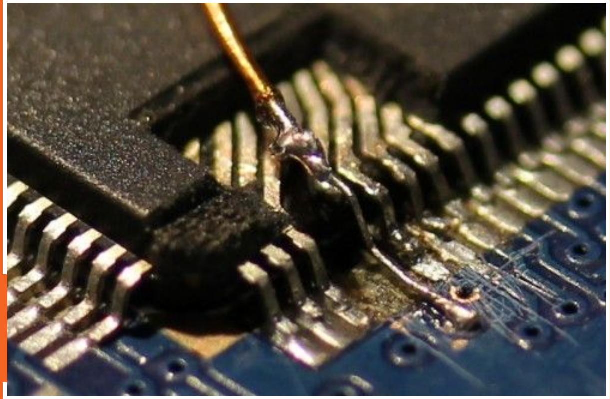
- Improve wireless connection by writing driver in C/C++ or change the bluetooth module.
- 3-D print or machine a sturdier base
- Address learning curve with a tutorial program
- Design a smaller PCB to fit in the acrylic shell

Retrospective

- Thoroughly test design on breadboard
- Always over-order
- Design early and iterate
- **Better Planning**



Source: SparkFun.com



Source: HackedGadgets.com

Thank You