Programmable Trackpad

Final Presentation

Group 18
Taylor Barnes - CpE
Jonah Halili - CpE
Brian Modica - CpE
Bradley Vanderzalm - CpE

Project Motivation

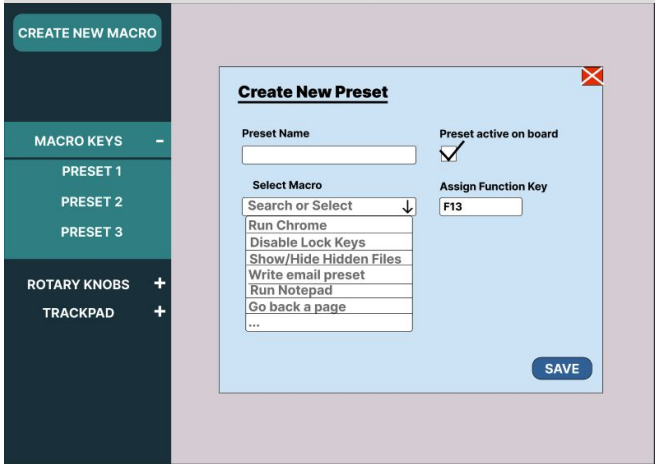
- The main goal of the programmable trackpad is to increase a user's productivity while using a computer
- The device will allow for usage of convenient short cuts
- Similar devices exists for the computer mouse, but nothing for the trackpad
- The intention is for this device to completely replace the default laptop trackpad
- Create a User Friendly Application software to easily pick and customize their own macro shortcuts to be used on the device's keypads and rotary knobs.

+

= ?

Project Concept

- Hardware device that acts as a fully-functioning trackpad.
- Buttons and rotary encoders on the device are assigned to macros that the computer executes.
- The user customizes macros via a graphical user interface.



Goals and Objectives

Goal	**Objective** (how we achieved said goal)
Reduce common and repetitive tasks	Add buttons with macro key capabilities that are programmable.
Convenient	Manage and run all Hotkey macros within a Graphical User Interface and have them operational even after termination of the application.
Ergonomic	Support ambidextrous users.
Low Learning Curve	Application with user-friendly interface to program macro keys. Only have to run one executable file, the user doesn't need Python or AutoHotKey installed.
Customizable for user	**Hardware** - Ability to easily remove keys to the user's liking. **Software** - Application should be able to create and store to run on the device

Functions

Function	Description
4 Mechanical Keys	Capable of macro and keybind function.
3 Rotary Encoders	Capable of audio mixer, adjusting windows, etc. (per-application functionality).
USB Connection	For charging the battery and communicating with the PC.
Touchpad	Mouse replacement offering ergonomics.
4 Mouse Buttons	Availability changes based on dominant hand usage.
Power Switch	Turn the device on or off.
Application User Interface	Main ability to program and customize hardware keys with macros

Requirements

Requirement	Planned Device Specifications	Actual Device Specifications
Device Dimensions	5” x 5” x 2”	6.5” x 4.5” x 1.079”
Device Weight	≤ 1 lb	7.9 oz
Device Latency	≤ 48 ms	~10 ms
Hot-swappable Switches	Yes	Yes
4 Mechanical Switch Inserts	Yes	Yes
3 Rotary Encoders	Yes	Yes
USB Connectivity	Yes	Yes
Bluetooth Connectivity	Yes	No
Battery Lifetime	≥ 10 hours	Was not able to verify
Battery Charging Time	< 3 Hours	90 Minutes

Market Analysis

Apple Magic Trackpad

- Price: $129.99
- Similarities to our device:
  - Small form factor
  - Fits conveniently on a desk
- Differences from our device:
  - No physical buttons
  - Little to no customizability
  - No Windows support

Mousetrapper Advance 2.0

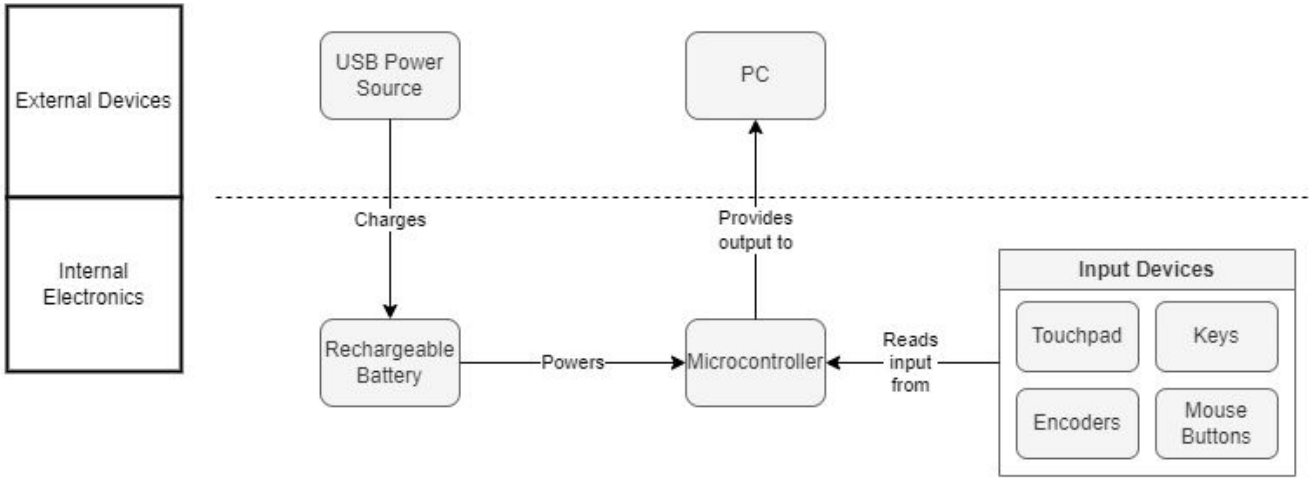
- Price: $200 - $300
- Similarities to our device:
  - Physical and customizable buttons
- Differences from our device:
  - Much larger
  - Wired functionality only

Keymecher MANO-703

- Price: $39.99
- Similarities to our device:
  - Has macro keys
  - Fits conveniently on a desk
- Differences from our device:
  - Macro key functions are hard coded, not customizable

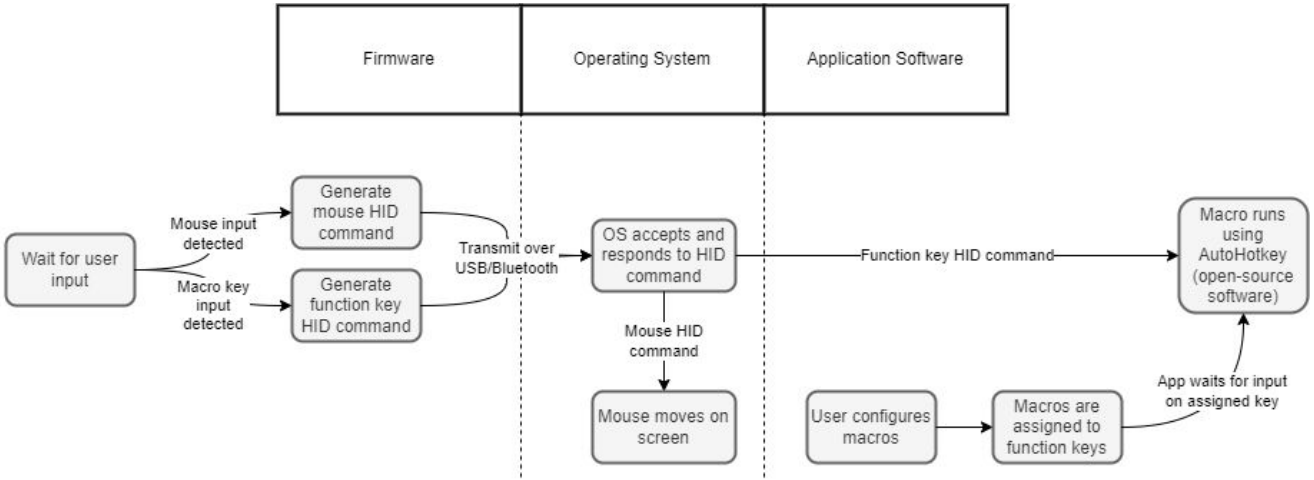
Hardware Overview

- The hardware device consists of a power system and an I/O system.
- A microcontroller handles data and transmits it to the user's PC.



Software Overview

- The device's software operates on three levels: the device firmware, the PC's operating system, and the application.
- The firmware processes user input and transmits to PC.
- The PC's operating system interprets HID commands (mouse and keyboard).
- The application executes user-defined macros.



Technology Investigation: Hardware

Battery technology comparison

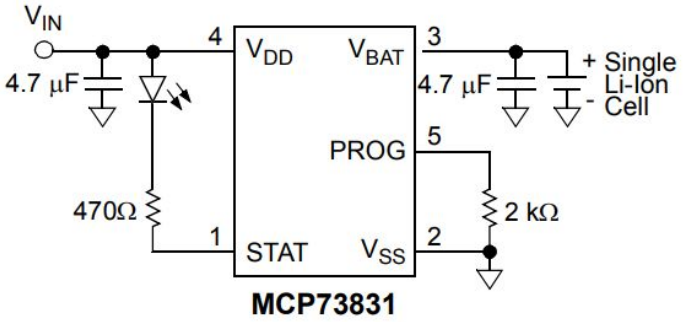
- LiPo batteries are the common choice for small devices like cell phones.
- Many microcontrollers and voltage regulators are designed to work with typical LiPo batteries.

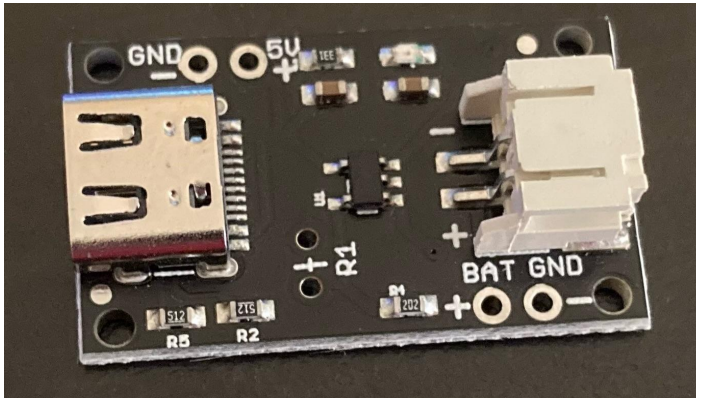
Technology	**Size**	**Capacity**	**Efficiency**	**Cost**	**Protection**
Lead-acid	Largest	Medium	Medium	Lowest	None
Lithium Ion	Medium	Medium	Best	Highest	None
Lithium Polymer	**Smallest**	**Medium**	**Best**	**Highest**	**Built-in**
Nickel Metal Hydride	Medium	Best	Medium	Medium	None

Technology Investigation: Hardware

Battery charger

- To charge the battery, the device needs a chip that will deliver constant current at the battery's rated voltage.
- The TP4056 and MCP73831 both accomplish this task, so they were both considered.
- The MCP73831 was chosen because it is simpler and has better documentation.





Technology Investigation: Hardware

Voltage regulation

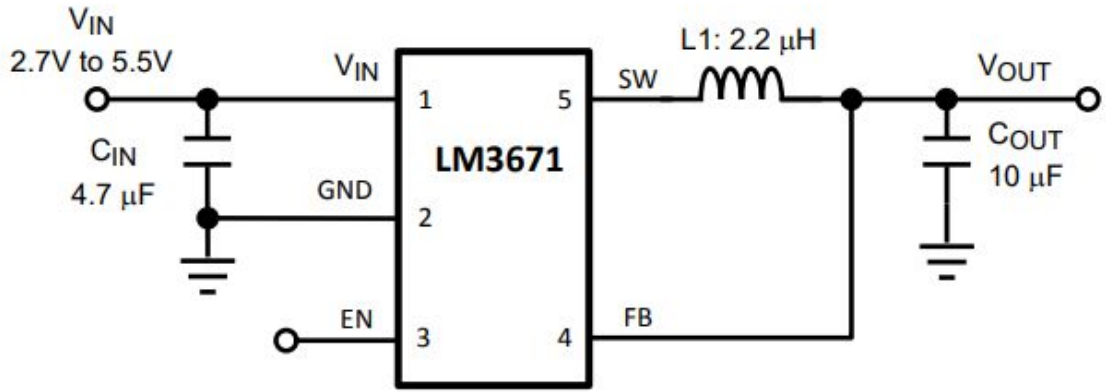
- The device electronics use three different voltages.
  - 5V for USB
  - 3.3V for microcontroller
  - 3.7-4.2V battery
- A voltage converter is needed to regulate the 3.3V nodes.
- The table shows comparisons between various considered parts.

3.3V Chip	Input Voltage	Current Output	Number of extra parts (inductors and capacitors) needed per documentation
TPS62203	2.5-6 volts	300 mA	3
AP63203	3.8-32 volts	2000 mA	4
LM3671	2.7-5.5 volts	600 mA	3

Technology Investigation: Hardware

Voltage regulation

- For the regulator, we chose the LM3671 because it is designed specifically for LiPo batteries.



Technology Investigation: Hardware

Input Units: Keyboard Switches

- Linear, tactile, clicky, silent
- Actuation force
- Sound level
- Price ($8 - $15 for a pack of 10)

Technology Investigation: Hardware

Switch	Actuation Force	Behavior	Sound Level
Gateron Red	45 grams	Linear	Quiet
Gateron Brown	55 grams	Tactile	Slightly Audible
Gateron Blue	60 grams	Clicky	Very Audible
Boba U4T	**62 or 68 grams**	**Tactile**	**Slightly Audible**
Gateron Black Ink v2	60 grams	Linear	Slightly Audible
Kailh Box Jade	65 grams	Clicky	Very Audible

Technology Investigation: Hardware

Input Units

Rotary Encoders

- Not many to choose from, generic ones did just the job
- We wanted haptic feedback between detents while turning the knob
- Potentially have another vertical input for more customizability
- Why not use a potentiometer
    - Did not achieve what we potentially wanted in terms of customizability

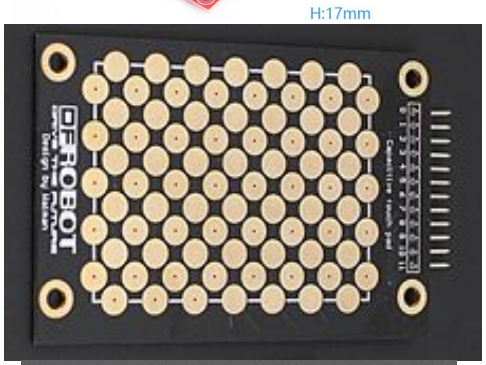
Trackpad/Touchpad Left and Right Click Switches

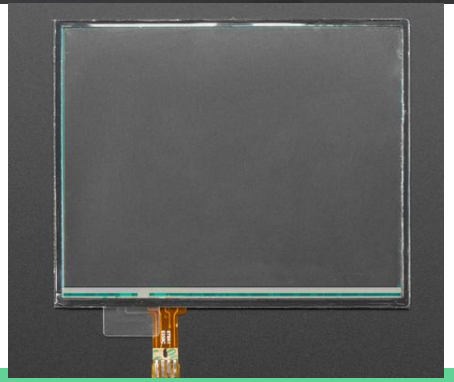
- Any tactile button that had enough height to reach the left and right button plates

Technology Investigation: Hardware

Touchpads

Touchpad	Size	Price	Integration	LCD Display	Power Drain
AliExpress RGB Touch Display Module	4.5” Diagonal	$15	9 Through-Hole Pins	Yes	High
DFRobot Capacitive Touch Kit	3.2” Diagonal	$20	12 Through-Hole Pins	No	Medium-Low
Adafruit Resistive Touch Screen	3.7” Diagonal	$6	4 Pin Flat Flex Cable	No	Low





Technology Investigation: Hardware

Microcontroller Requirements

- 15 GPIO pins
  - 6 encoders
  - 4 keys
  - 2 touchpad
  - 2 mouse buttons
  - 1 orientation switch
- Programmable with SWD
- Bluetooth-capable
- USB-capable
- 2 ADC pins for touchpad

Technology Investigation: Hardware

Microcontroller

- ATMega32: Meets GPIO requirements, but does not have built-in Bluetooth/USB interfaces.
- nRF52840: Has necessary GPIO, is capable of generating Bluetooth and USB messages, does not have a Bluetooth antenna.
- ESP32: Similar specs to nRF, slightly less powerful.
- MDBT50Q: Combines the nRF MCU with a Bluetooth antenna.

MCU	GPIO Pins	Bluetooth	USB	Programming Method
ATMega32	32	None	None	Arduino IDE, Microchip Studio
nRF52840	48	On-chip	On-chip	nRF5 SDK, Arduino IDE, CircuitPython
ESP32	34	On-chip	On-chip	Arduino IDE

Technology Investigation: Software

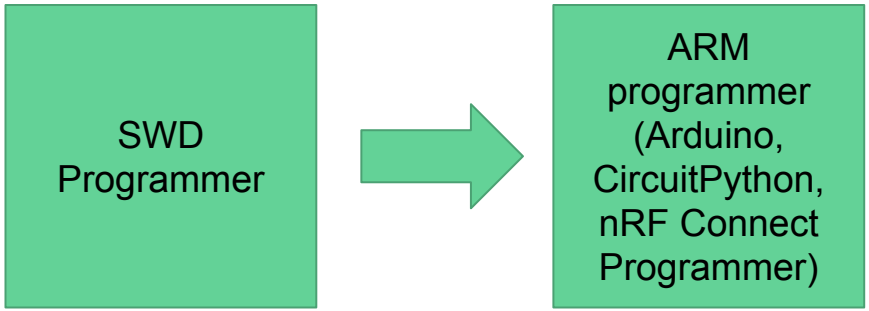
HID Libraries and Firmware

- CircuitPython to connect the I/O pins to respective input units
- USB and Bluetooth connection
- Keyboard, Rotary Encoder, X and Y coordinate reading
- Arduino also has libraries but CircuitPython was more involved with what we wanted
- KMK and QMK for keyboard firmware was also considered

Technology Investigation: Software

SWD Programmer and Connector

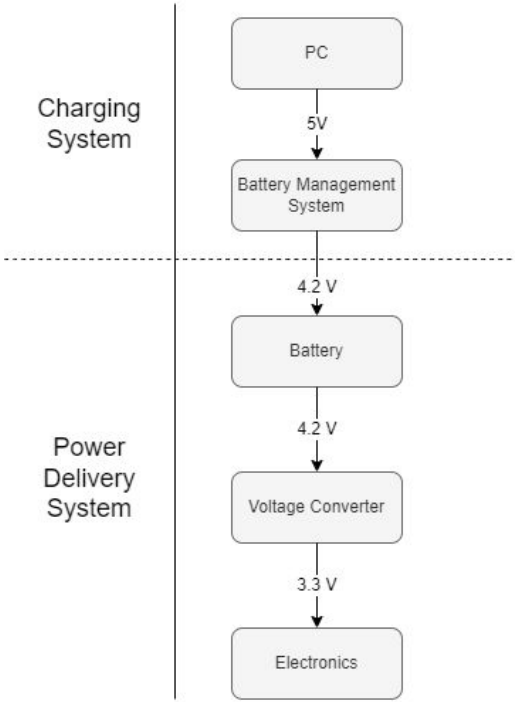
- This allows us to program the blank microcontroller
- Install the bootloader for the device where we can then edit through Arduino
- Two ways to go about uploading firmware
  - Arduino
  - CircuitPython



Design Details: Hardware

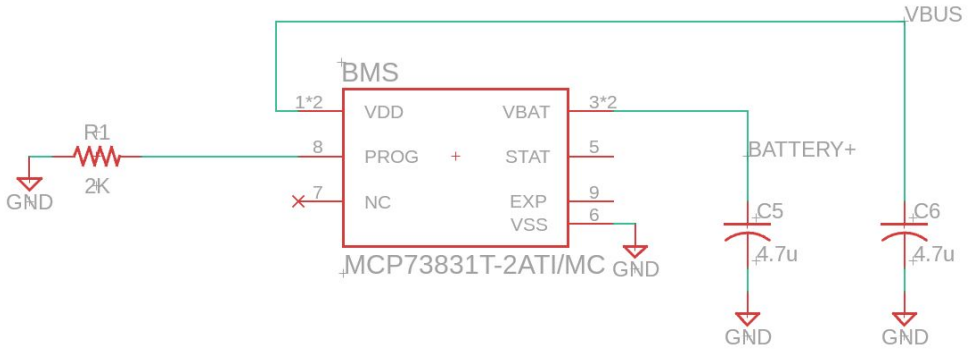
Power System

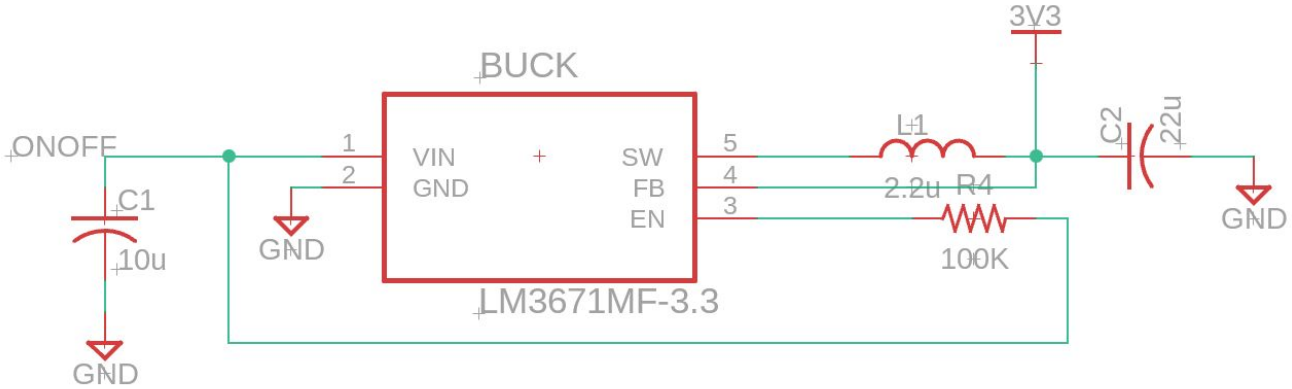
- The PC supplies 5V power.
- The 5V power activates the MCP73831, which charges the battery.
- The battery supplies 3.7-4.2V power, which is then regulated by the LM3671.
- The regulated 3.3V power activates the device electronics.

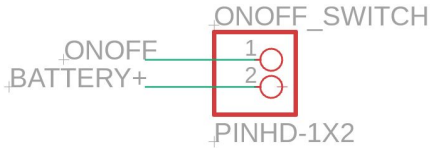


Design Details: Hardware

Power System Schematics

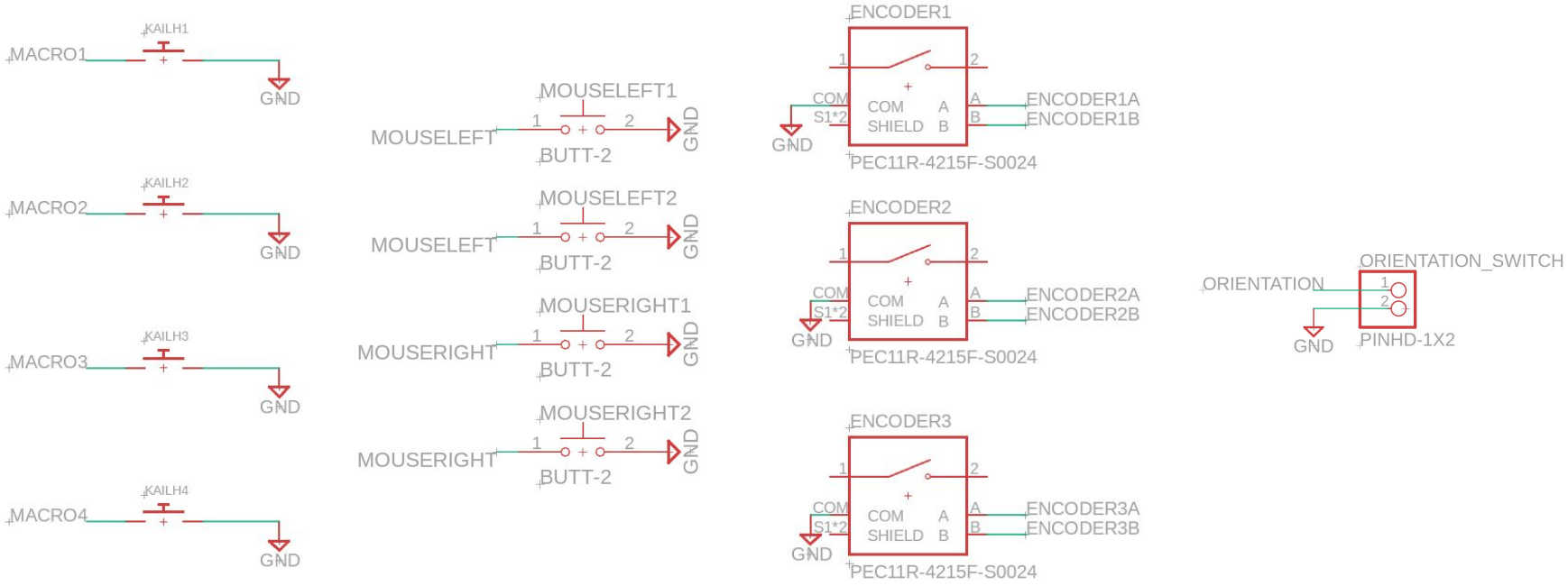






Design Details: Hardware

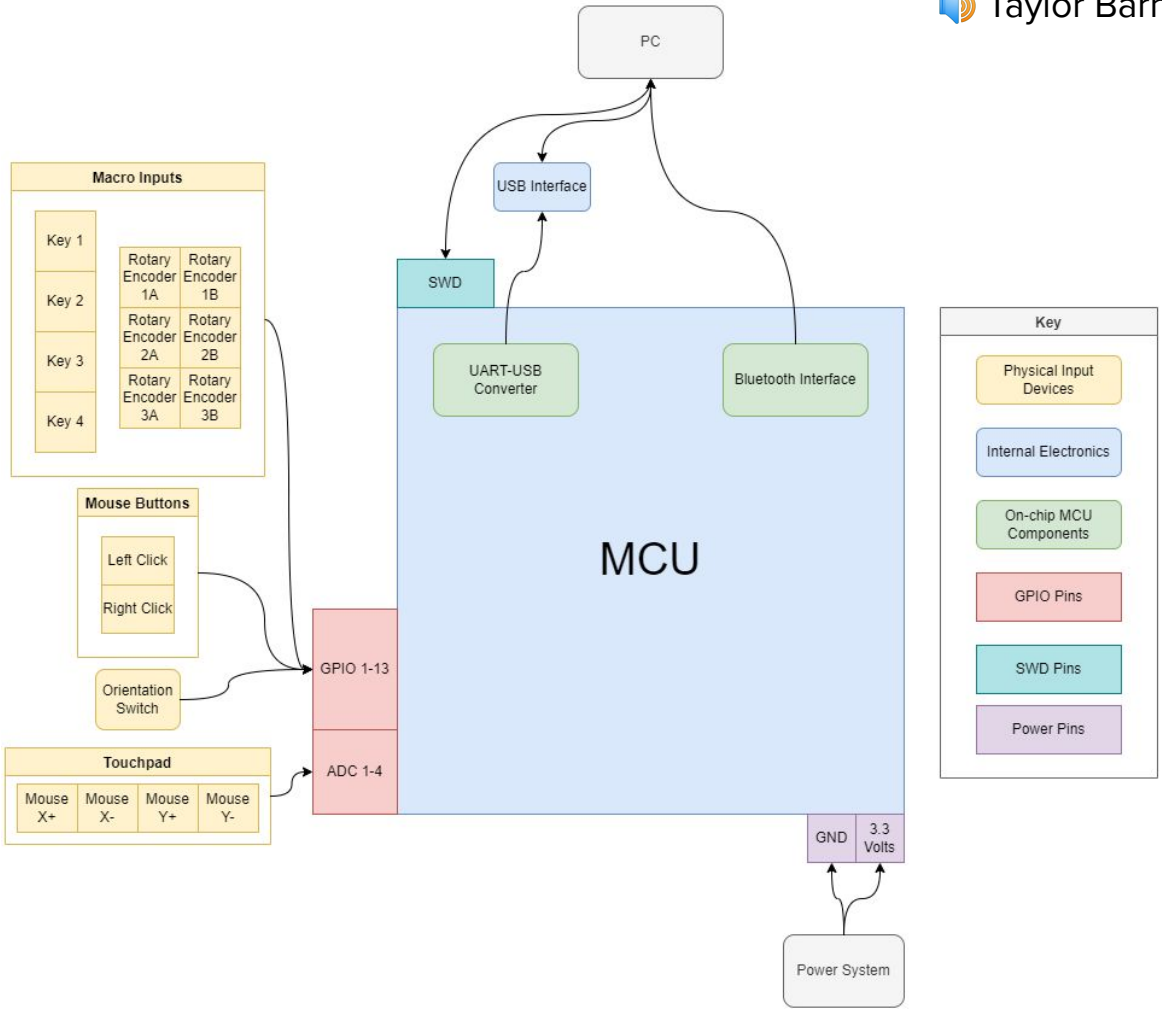
Input Unit Schematics



Design Details: Hardware

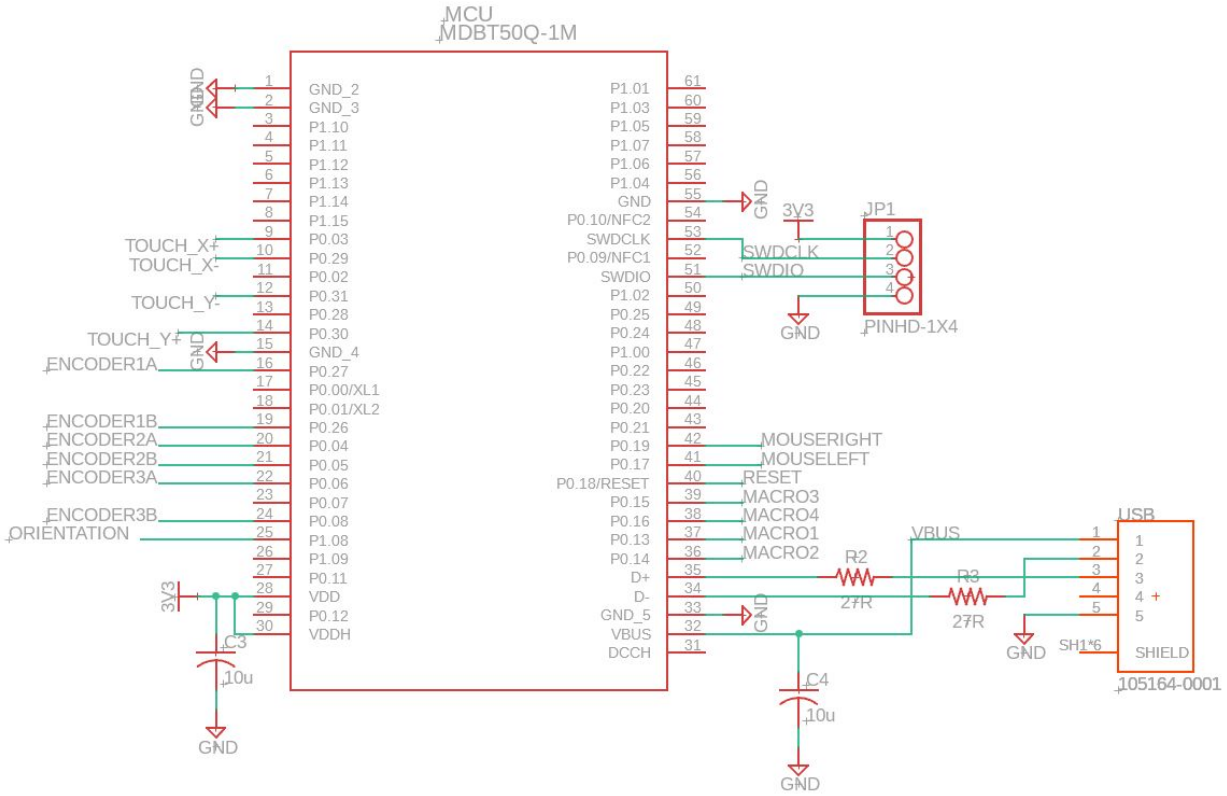
Microcontroller

- The MCU is powered with 3.3V from the power system.
- The input units connect to the MCU via GPIO pins.
- The MCU's Bluetooth and USB interfaces with the user's PC.
- The PC can be used to program the MCU on its SWD pins.



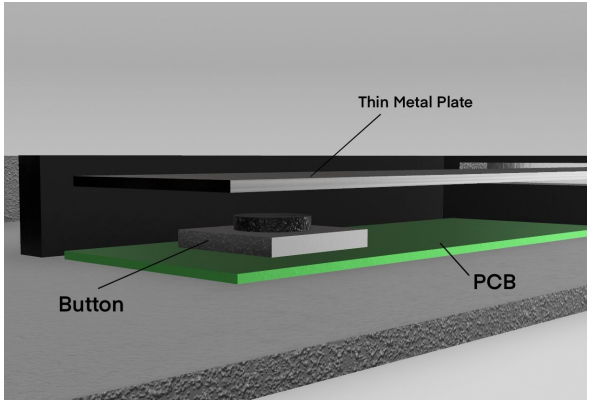
Design Details: Hardware

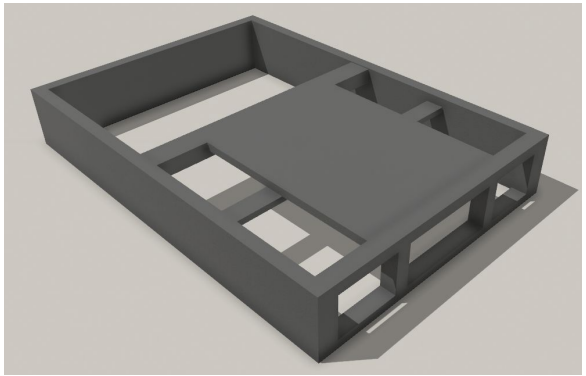
Microcontroller Schematic



Design Details: Chassis

- Open casing with platform on the top for the trackpad
- Backplate with standoffs to hold pcb
- Cutouts on the side to access on/off switch, orientation switch, and micro USB.
- Left and right click mechanism
    - Thin metal plate with some material under it to close the gap between the button and the plate



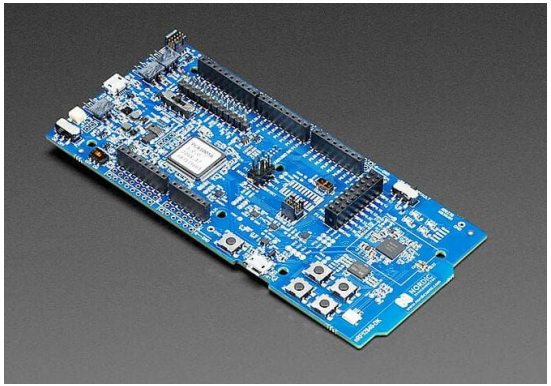


Design Details: Hardware

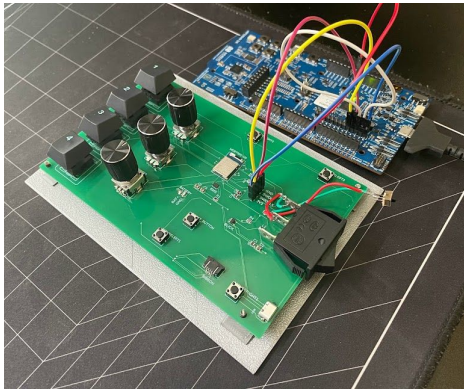
SWD Programmer

- Initially we had to work with a 3rd party programmer due to availability and cost restrictions, but it didn’t work
- Switched over to the PCA10056 as an alternative for J-Link programming

3rd party programmer



PCA10056

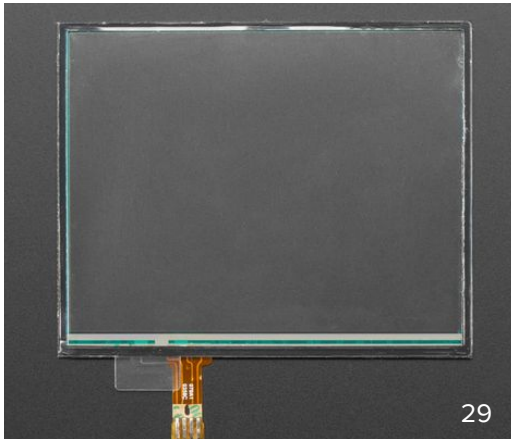


J-link connection

Design Details: Hardware

Adafruit Resistive Touch Screen

- This touchscreen passes X and Y coordinates to the MCU via a 4 pin ribbon cable
- The TouchScreen library for Arduino is used to take these coordinates and translate it to mouse movement
- Calculations within the code are made to reduce unintended jitteriness and jumps in mouse movement



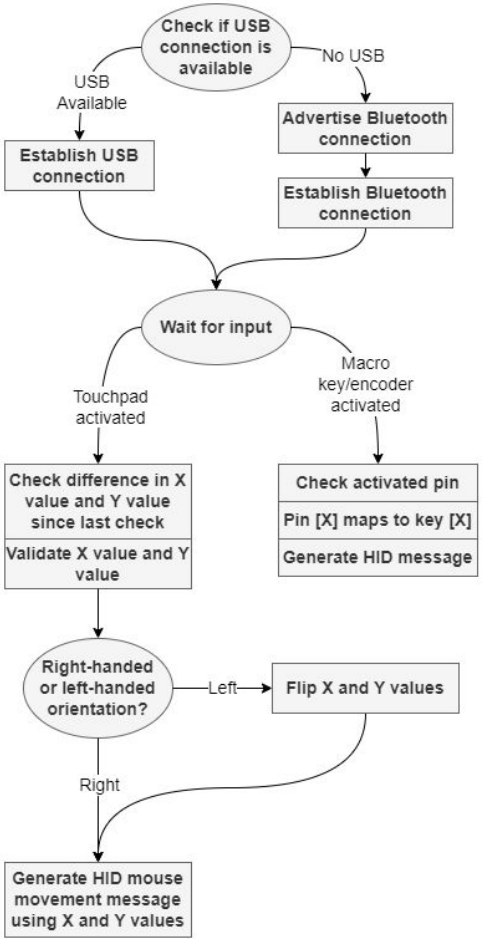
Firmware: C, Arduino, and CircuitPython

- Three methods to upload firmware to the microcontroller
  - Arduino: uploading bootloader then working with C++ through the Arduino IDE
    - Most viable contender, bluetooth was finicky/didn't work
  - CircuitPython: uploading bootloader for the microcontroller and the CircuitPython capabilities then editing the code.py directly
    - Most preferred but microcontroller wouldn't accept CircuitPython uf2 bootloader, everything including bluetooth worked
  - C: using nRF Connect through the vsCode extension to upload hex files with all the mapping
    - Worst method due to conversion of files to hex files to upload
- Ended up using Arduino for all firmware programming

Design Details: Firmware

The main functionalities we needed were: keyboard HID, mouse HID, rotary encoder controllers, and Bluetooth. We connect available GPIO pins to functions keys.

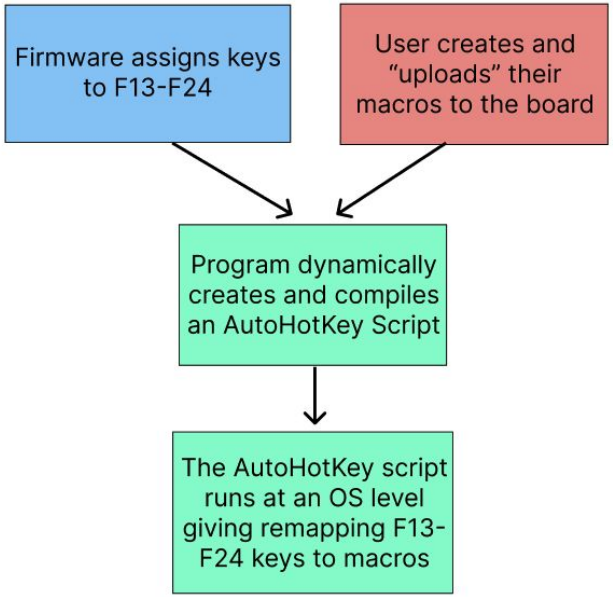
- Main idea was to set the keys to keys people wouldn't typically use (F13 - F17)
  - These would be set to macros or whatever the user wanted through the application
  - This would be done using the AHK scripting
- Rotary Encoders each need two function keys for each channel on the encoder
  - F18-F24
- 4 Left and Right buttons
  - F25-F28
- Trackpad coordinates are set to calculated mouse movements
  - Connected through analog pins



Design Detail: Application Software

A graphical user interface (GUI) is needed for the user to design macros and “upload” them to the device.

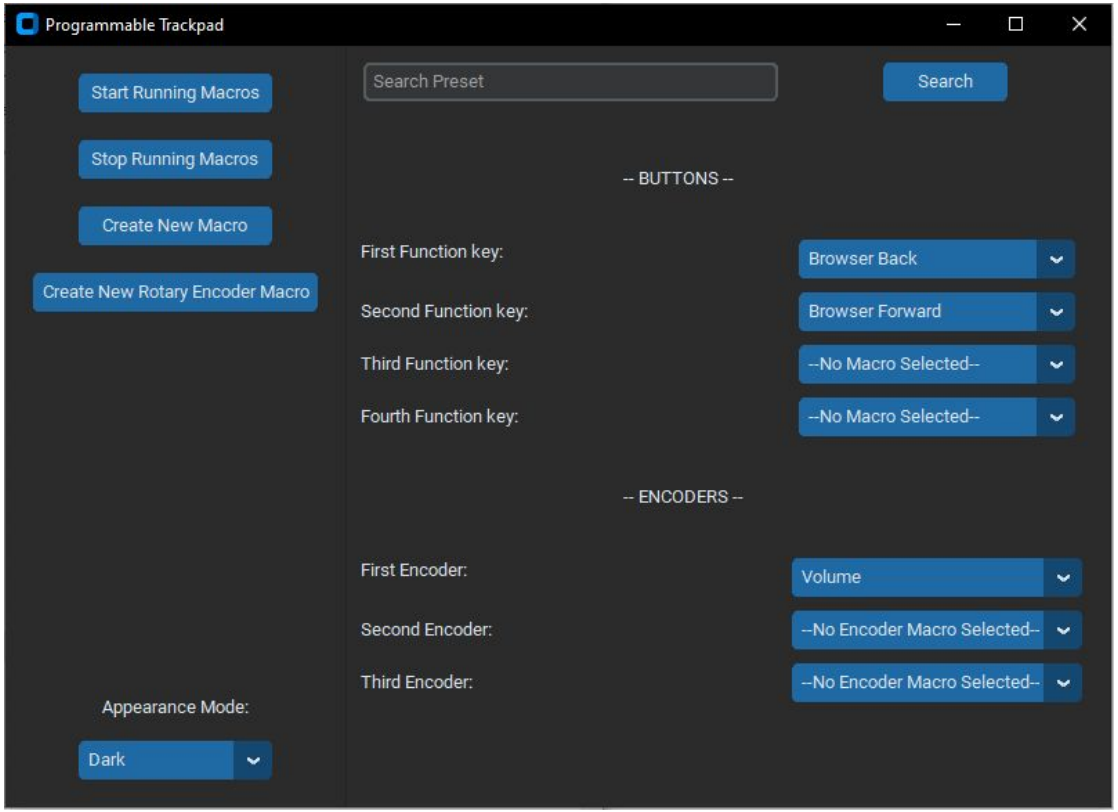
- User interface with forms and dropdowns for the user to design their own Macros.
- Dynamically create and compile one AutoHotkey script that communicates with Windows at the OS level to perform the macro.

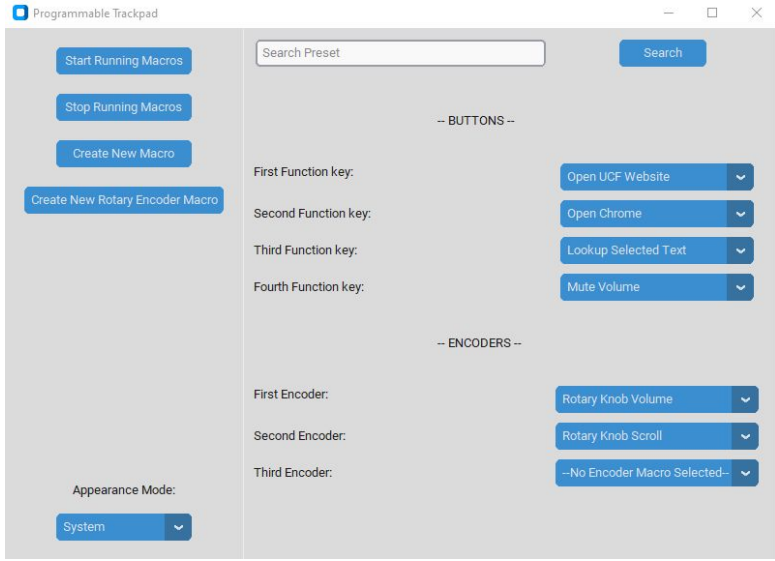


Design Detail: Technology Used in Application software

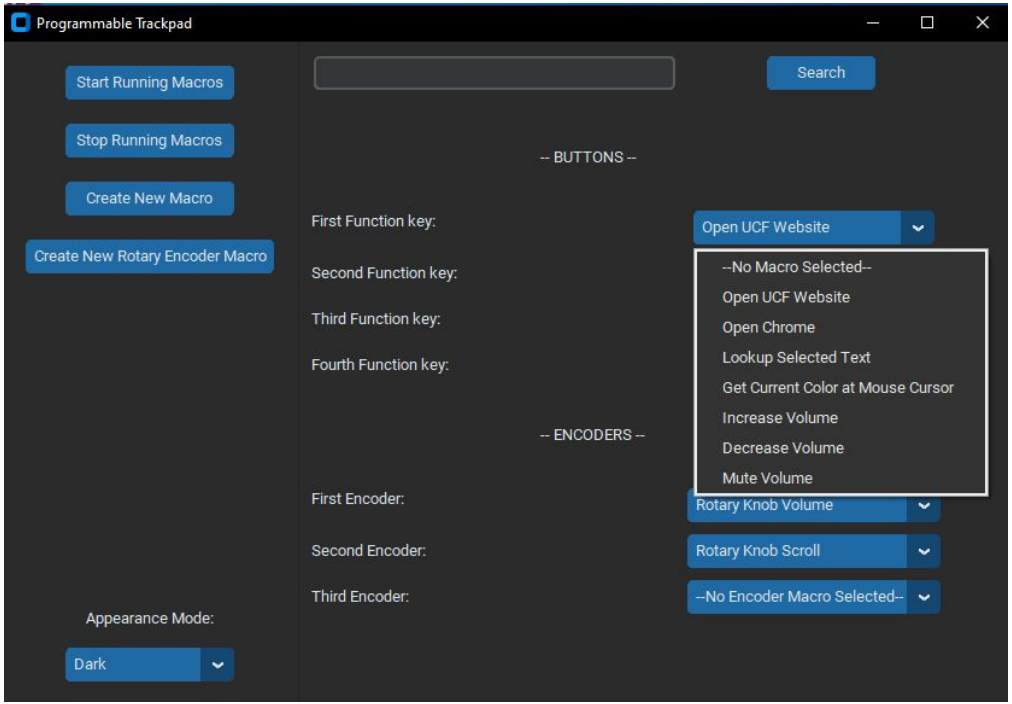
	Frontend	Backend
Technology	TKinter GUI Library	AutoHotKey (AHK), scripting language for windows
Coding Language	Python	Python and AHK macros performing at the OS Level
Features	Open source library, abundant documentation	Dynamically create AHK macro scripts based on the user's choice. Abundant documentation
Compatible with Windows OS?	Yes	Yes
Cost	Free	Free

Design Detail: GUI Design



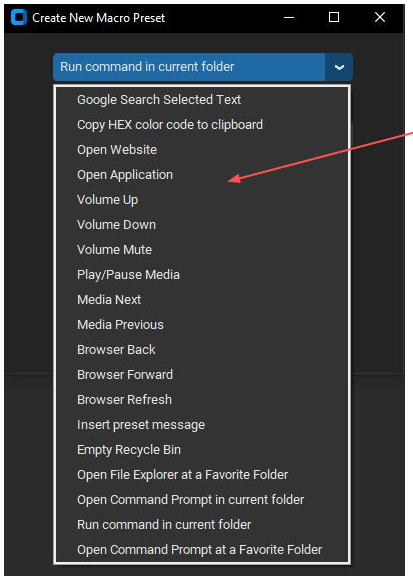


Design Detail - Home Page and Overall Functionality

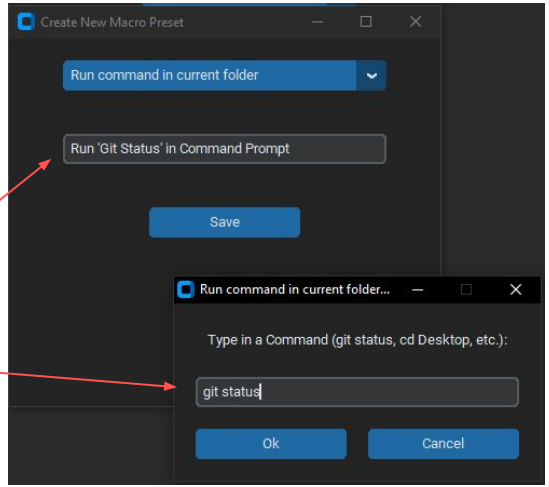


- Based on the dropdowns, it will dynamically create, compile, and execute an AutoHotkey Script that remaps F13 to F24 to the user's macro hotkeys.
- The user can manage their macros by doing the following
  - Create
  - Search
  - Edit
  - Delete

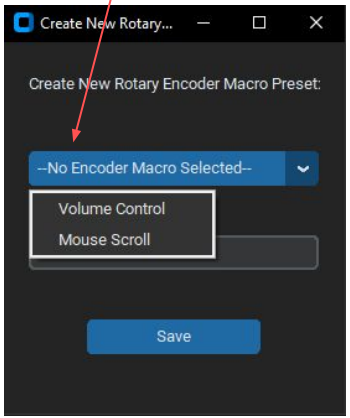
Design Detail: Create Macro Window



- Default Macros to choose from for buttons
- Give custom name along with any other user input (if needed)



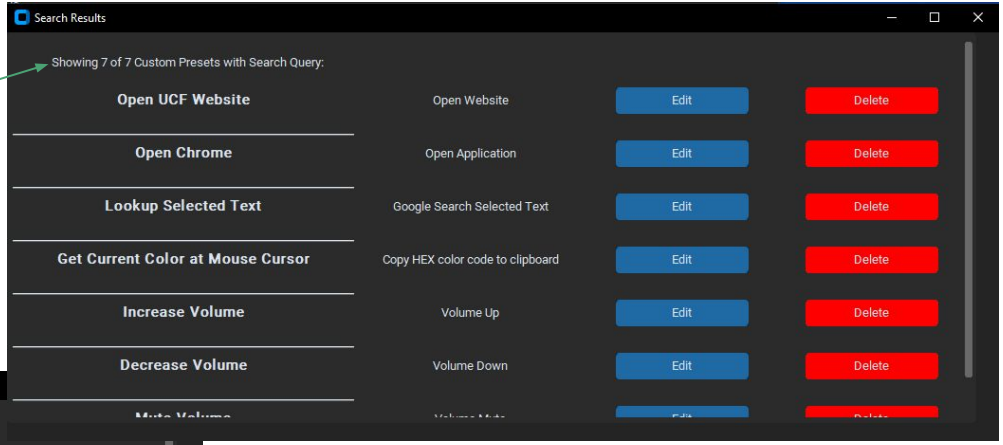
- Default macros that make sense only for Rotary Encoder Knobs

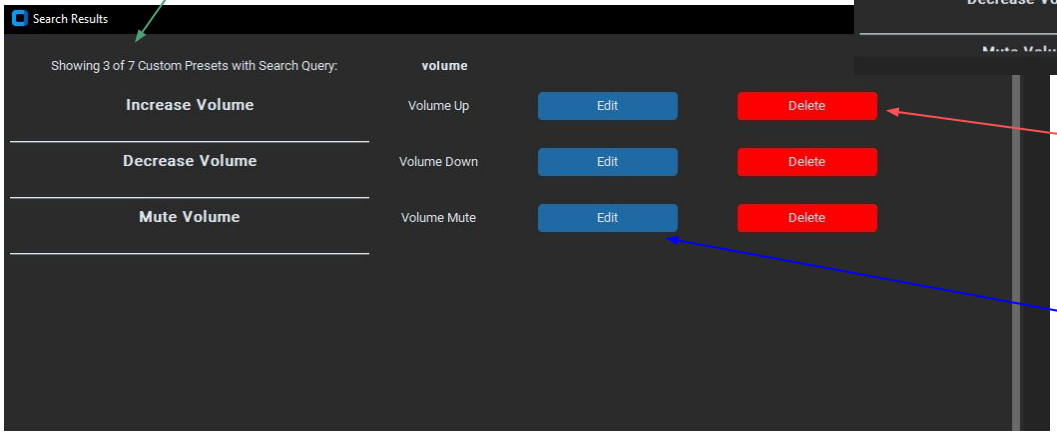


Some Macros have static functionality such as “*Google Search Selected Text*” but some allow user input such as “*Open Application*” where the user can choose any application.

Design Detail: Search Window

- Search based on macro Name and/or Type

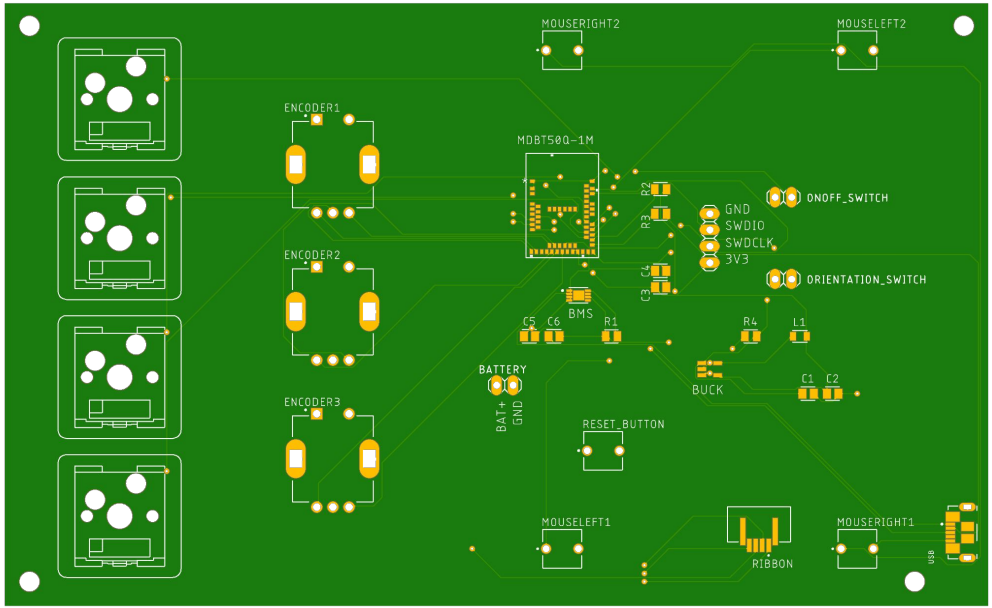




- Dynamic Delete - Immediately removes preset from table and the dropdowns on the home page
- Edit - Opens the same Create Macro window but with form data pre-filled

Design Details: PCB Design

- Designed in Fusion 360
- Large size accommodates for the size of the touchpad and macro keys
- Four holes in corners used to mount PCB in chassis
- Ground pour avoids the Bluetooth antenna



Further Strides future of the device

- USB-C upgrade from micro-USB
- Bluetooth capability
  - Exhausted all attempts to implement bluetooth: CircuitPython, Arduino community libraries, nRF forums, etc.
- Unit upgrades
  - Left and right click buttons
  - Trackpad upgrade, smoother mouse movement
- Chassis upgrade

Finance: Development Budget

- These materials were purchased as part of the prototyping phase of development.
- Total development budget: $317.80

Item	Price	Purpose
3.7" Touchpad	$11.45	
Touchpad to USB breakout board	$17.59	Prototyping touch functions
500 mAh LiPo battery	$6.99	
1700 mAh LiPo battery	$10.99	
18650 battery charger	$5.99	
USB-C Battery Management board	$8.99	
USB-C Battery Charging board	$12.99	
TPS61023 Development Board	$3.56	
LP3671 Buck Converter	$11.22	Prototyping power system
Rotary Encoders	$9.00	
Mech Switch Breakout Board	$3.25	Prototyping HID commands
MDBT50Q	$12.95	
Itsbitsy nRF52840 Express	$27.20	Prototyping microcontroller
SWD Programmer	$6.40	
nrf52840 DK	$57.18	Prototyping SWD programming
ESP32-WROOM-32D	$9.09	Exploring nRF alternatives
PCBs and Stencils	$102.96	Prototyping system integration
Total	$317.80	

Finance: Bill of Materials

- These are the materials used to manufacture one Programmable Trackpad.
- Total cost: $60.19

Item	Quantity	Price
MDBT50Q	1	$12.95
500 mAh LiPo battery	1	$6.99
LM3671 3.3V Converter	1	$1.61
MCP73831 Battery Management System	1	$0.77
3.7" Touchpad	1	$11.45
Rotary Encoder	3	$2.90
Kailh Socket Pack	1	$2.50
Boba U4T Key Switch Pack	1	$11.50
PCB	1	$2.00
Ribbon Cable Receptacle	1	$0.36
Micro USB Receptacle	1	$1.36
	Total	$60.19

Conclusion

- Our team set out to design a niche product that did not exist in the market. Through our research, we discovered a gap in the market and identified an opportunity to fill it with our design.
- We are proud to say that our final design is at a satisfactory point where further improvements can still be made. With the given time frame and financial support it met a majority of the engineering requirements we sought after. We believe that our product has the potential to make a significant impact in its industry.
- We are grateful for the support and contributions of our team members and IDEM systems who helped us along the way. We could not have accomplished this without their hard work and dedication.