Material List

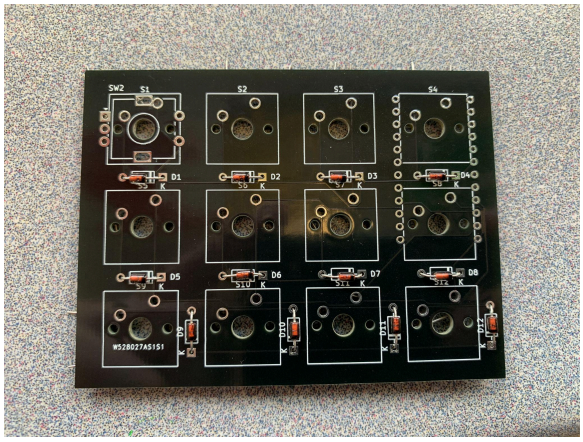
All of the materials needed should be included in your Macro-Pad Kit. If you are missing a part, please contact the Projects Executive or someone in the Projects Committee.

- 1x Top Plate
- 1x Bottom Plate
- 1x Board
- 11x Keyswitches
- 11x Keycaps
- 1x EC11 Rotary Encoder
- 1x Rotary Encoder Knob
- 12x 1N4148 Diodes
- 2x 0805 100nF Capacitor
- 1x Arduino Pro Micro and header pins
- 1x USB cable
- 6x Standoffs
- 12x Screws
- 4x Rubber feet

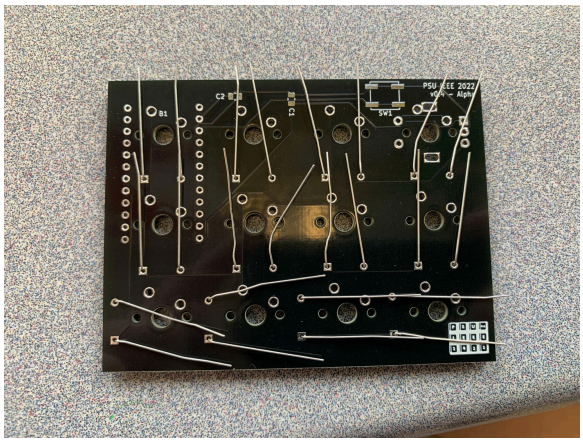
Soldering

Diodes

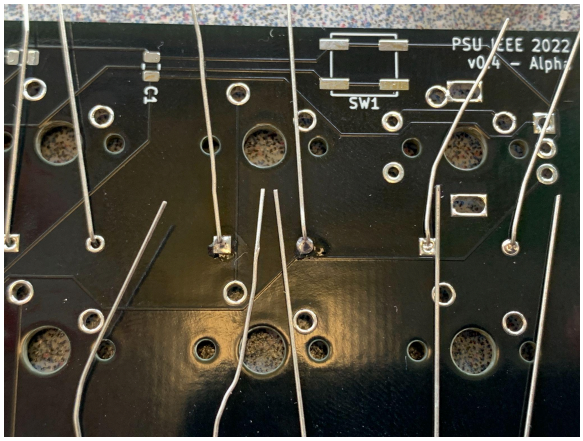
The first component we will solder and attach are the diodes. You should have 12 diodes, one for each switch. These diodes enable the microcontroller to determine what switch is being pressed through a Diode Matrix formation.



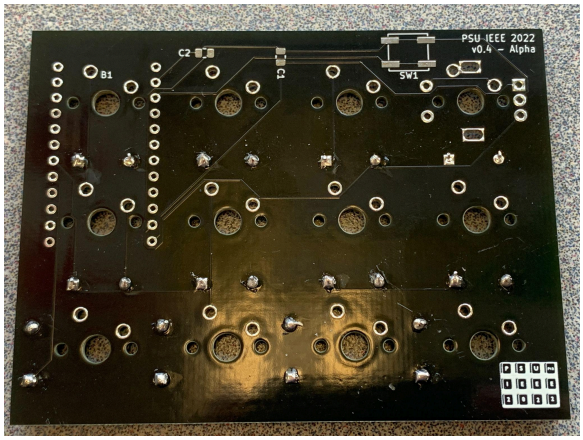
First, place the diodes in the through holes on the front of the PCB. These holes are marked with a rectangular diode symbol. Make sure to insert the diodes in the proper orientation. To do this, align the black stripe with the white stripe on the PCB.



While you insert the diodes, bend the legs to keep the diodes in place. This will not prevent them from falling out, but it will help. The direction you bend the legs does not matter.



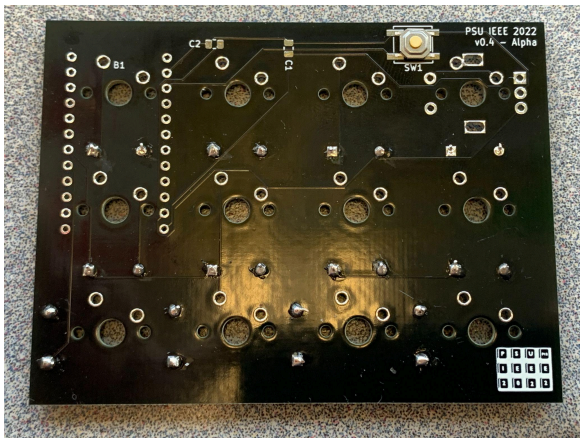
Once you have all of the diodes in place, you can begin soldering the diodes from the back of the PCB.



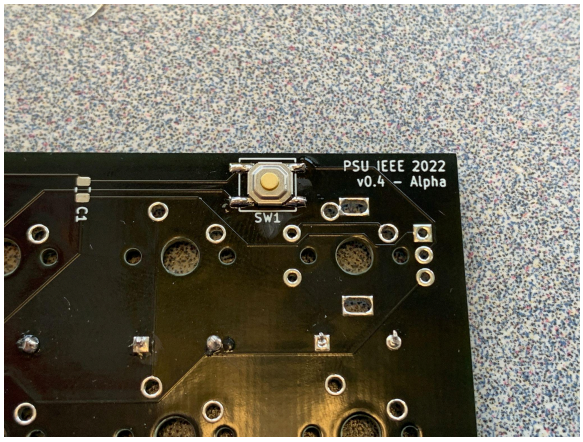
Once all of the legs are soldered, you can trim the legs off. Try not to trim the solder you applied.

Reset Button

Next, we will be soldering the reset button. This button allows us to reconfigure the layout. Reference the software section for more information.



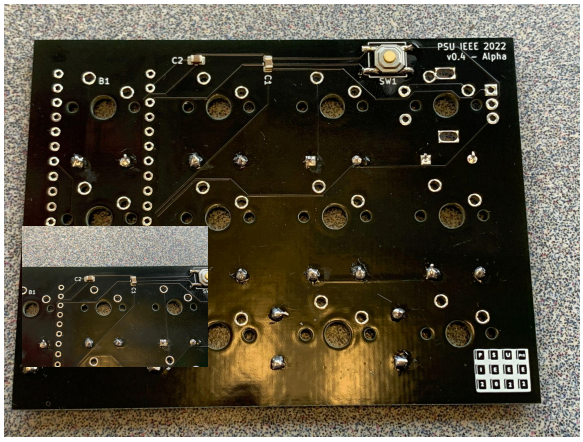
Place the reset button on the back of the PCB as shown. The orientation of the button does not matter. As long as the legs of the button align with the solder pads, it is in the proper location.



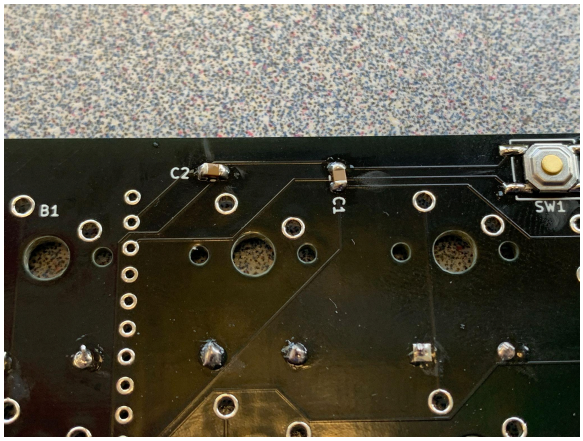
Carefully solder the button in place. Be sure that each leg makes a solid connection with the corresponding pad.

Capacitors

Once the reset button is soldered, we will solder the capacitors. These capacitors are required for the rotary encoder to function properly. You should have 2 100nF capacitors in your kit.



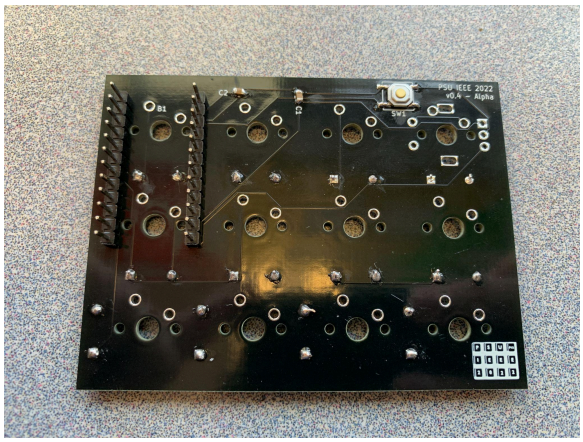
Place the two capacitors on the corresponding solder pads as shown in the image. The direction of these capacitors does not matter. Using tweezers to align the capacitors can be helpful.



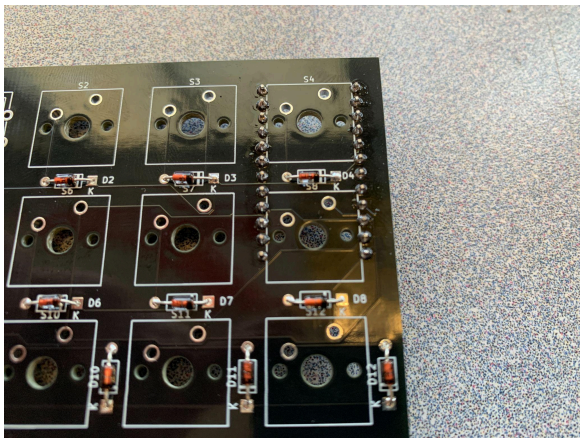
Carefully solder the capacitors onto the PCB. It is helpful to use tweezers to hold the capacitors while you solder. If you are having trouble, try tinning the tip of the soldering iron (put solder onto the tip of the iron). Then while holding the capacitor in place, put the soldering iron's tip onto the contacts of the capacitor and the pad.

Header pins

Next we will solder on the header pins for the microcontroller.



Place the header pins into the corresponding holes in the back of the PCB as shown. Make sure to put the shorter end of the header into the holes.

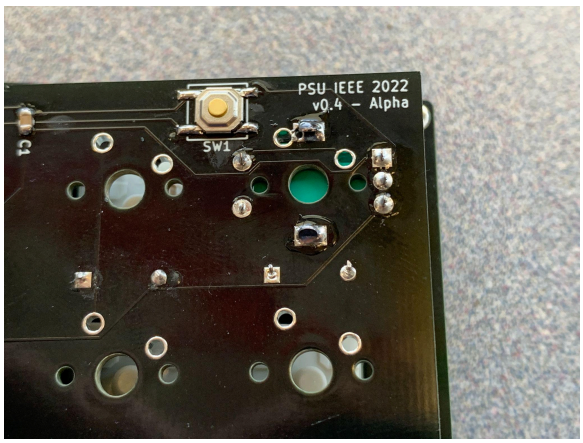


Solder the header pins from the front of the PCB. Make sure that the pins are as perpendicular to the PCB board as possible.

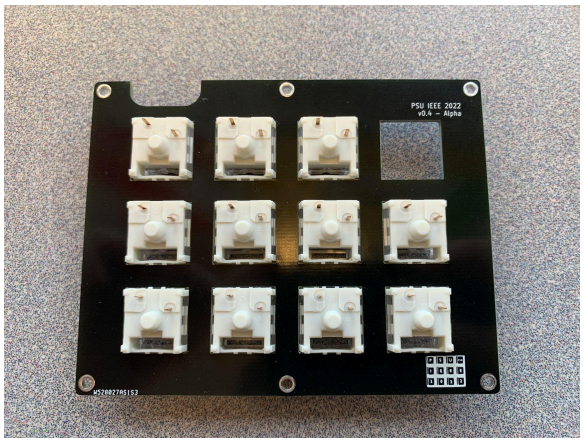
Switches and Encoder

The next things we will attach are the switches and the encoder. The switches will be the buttons that make up the matrix. The encoder will allow you to implement a scrolling function to your macro-pad.

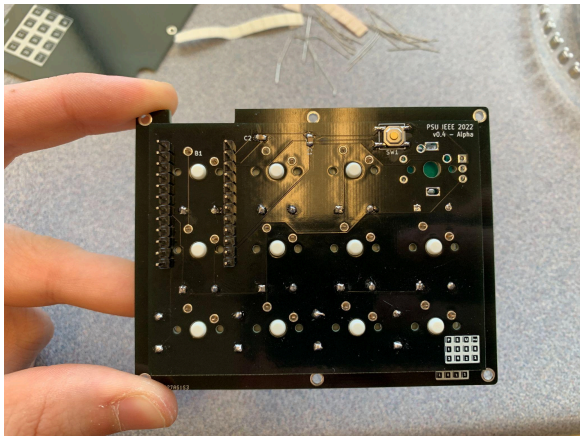
Place the encoder in the holes at the top left of the front of the PCB. The encoder should only go in one way.



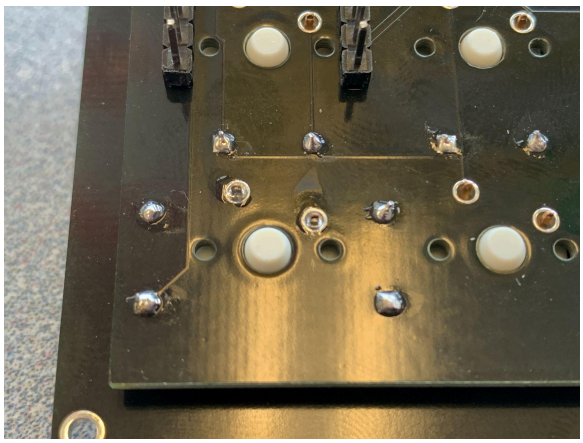
While ensuring the encoder does not move, solder the pins on the back of the PCB. Note that the cluster of two and the cluster of three pins are functional pins. The two larger pins are used to secure the encoder to the PCB. Therefore, the two larger pins do not need to be electrically sound.



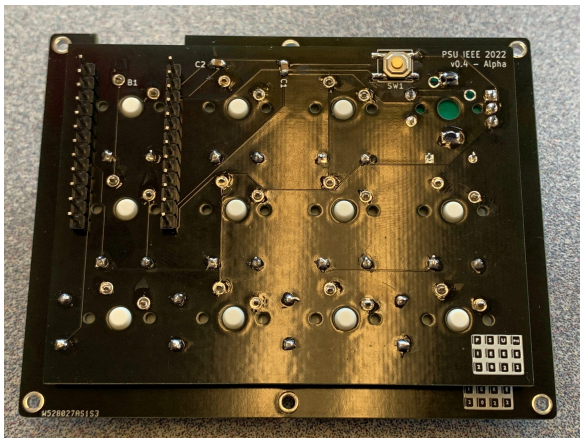
Next, place your switches into the front plate. They should slot into place. This requires significant force. If the switches fall out then you are inserting the switches in the wrong direction. Take note of the orientation of the switch. This is important for the switches to line up with the electrical PCB. Also, take note of which way the top plate is facing. Slot the switches in the top plate as in the picture, with the slot in the top left. Leave the top right switch-slot open for the rotary encoder to fit through.



Next, align the button pins with the holes in the electrical PCB.



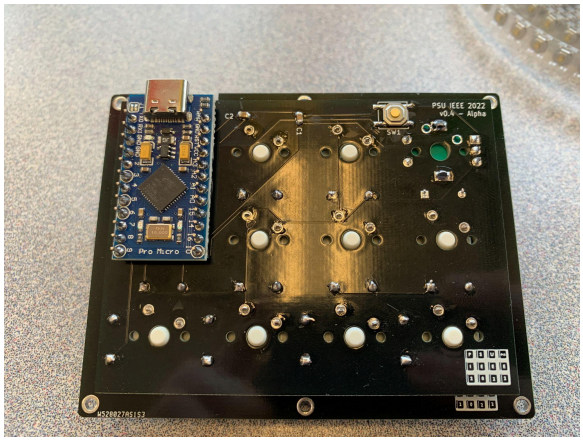
Begin soldering the switches. It may be difficult to keep the front plate and the PCB together. To make soldering easier, you can use a technique called “tacking”. This is when you solder components on the edge to secure the components together. Solder a switch on one edge of the PCB and one on the other side. This will prevent the top plate and switches from falling out.



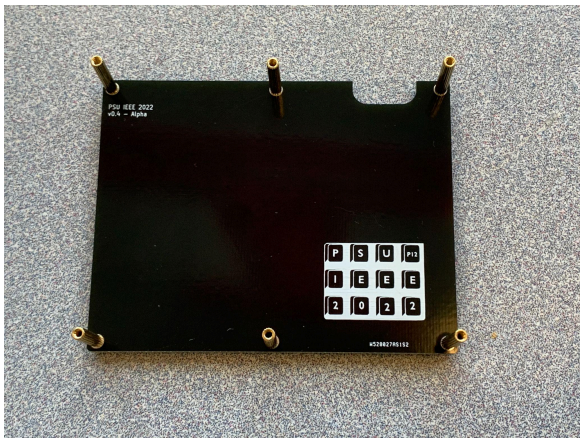
Continue to solder all of the switches from the back of the PCB.

Microcontroller

Next, we will solder on the microcontroller. This is a knock-off version of the Arduino pro-micro. This particular version is crucial because it allows the microcontroller to appear as a keyboard to any computer it connects to. This is what makes the macro-pad work.



Place the microcontroller on the header pins. Orient the microcontroller as in the image. Ensure the electrical components are on the top of the microcontroller. Then solder the microcontroller on.



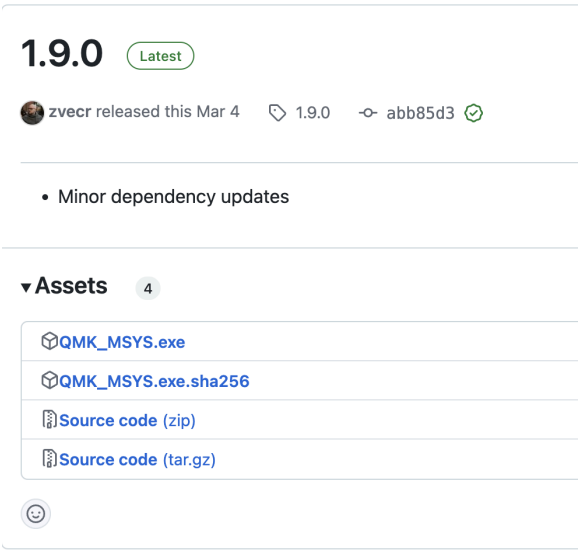
Finally, assemble the macro-pad. Screw the stand-offs into the back plate with the included screws. Then, you can place the front plate and the PCB on top of the stand-offs and secure it in the same way.

Once the macro pad is fully assembled, put your key caps on your switches and the dial on your rotary encoder.

Software setup

Prerequisites

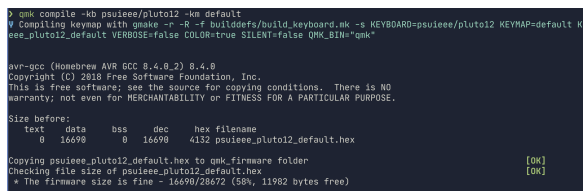
- QMK
    - On Windows, download the latest version of QMK MSYS from https://msys.qmk.fm/ (not sha)



    - On Mac, open your terminal and run “brew install qmk/qmk/qmk”
        - Note that you need to install HomeBrew first if you do not have it. Follow the instructions here, https://brew.sh, to do that then run the command above.

Setup QMK

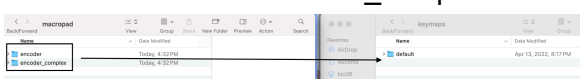
- Open up the terminal where you can run QMK commands (see bullet points directly below for what this is on Mac and Windows)
    - On Windows, open QMK MSYS.Windows defender will pop up. Press more info and then run anyway. Then save it where you want it. You only need to install the drivers and add to windows terminal. Then press install.
    - On Mac, open Terminal
- Type “qmk setup” and press enter
- Answer “y” to all of the prompts by typing the letter and pressing enter
- Test your build environment by running “qmk compile -kb psuieee/pluto12 -km default”
    - You will know if is successful if the final portion of the output says “Firmware size fine”



Download keymaps from GitHub

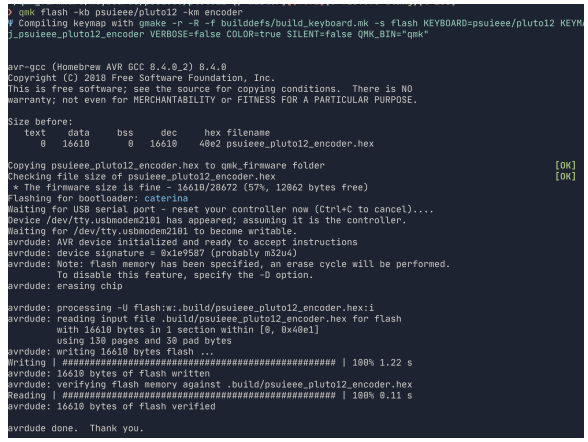
- Go to the PSU IEEE Projects GitHub repository here https://github.com/psuieee/projects
- Click “Code” then “Download ZIP”
- Extract the ZIP file on your computer
- In the extracted folder, navigate to “projects-main/spring2024/macropad”
- Open another window of File Explorer/Finder to navigate to the qmk_firmware folder
    - On Windows, this is under C:/Users/__YOUR_USER/qmk_firmware

    - On Mac, this is under “/Users/__YOUR_USER__/qmk_firmware”
- In the qmk_firmware folder, navigate to “keyboards/psuieee/pluto12/keymaps”
- Copy both the “encoder” and “encoder_complex” folders from the extracted GitHub repo (the original File Explorer window) into the “keymaps” folder from the qmk_firmware folder (the second File Explorer Window)
    - You should now have three folder in your “keymaps” folder: “default”, “encoder”, and “encoder_complex”



Flash your macropad

- In your terminal, run “qmk flash -kb psuieee/pluto12 -km encoder”
- After the software compiles, you will see that the software is waiting for a bootloader. Press the reset button on the macropad (underneath the PCB near the USB port) to enter bootloader mode
- The flash is complete when the progress bar for writing data reach 100%



```
❯ qmk flash -kb psuieee/pluto12 -km encoder
ℹ Compiling keymap with gmake -r -R -f builddefs/build_keyboard.mk -s flash KEYBOARD=psuieee/pluto12 KEYMA
j_psuieee_pluto12_encoder VERBOSE=false COLOR=true SILENT=false QMK_BIN="qmk"


avr-gcc (Homebrew AVR GCC 8.4.0_2) 8.4.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Size before:
   text    data     bss     dec     hex filename
      0   16610       0   16610    40e2 psuieee_pluto12_encoder.hex

Copying psuieee_pluto12_encoder.hex to qmk_firmware folder                          [OK]
Checking file size of psuieee_pluto12_encoder.hex                                   [OK]
 * The firmware size is fine - 16610/28672 (57%, 12062 bytes free)
Flashing for bootloader: caterina
Waiting for USB serial port - reset your controller now (Ctrl+C to cancel)....
Device /dev/tty.usbmodem2101 has appeared; assuming it is the controller.
Waiting for /dev/tty.usbmodem2101 to become writable.
avrdude: AVR device initialized and ready to accept instructions
avrdude: device signature = 0x1e9587 (probably m32u4)
avrdude: Note: flash memory has been specified, an erase cycle will be performed.
         To disable this feature, specify the -D option.
avrdude: erasing chip

avrdude: processing -U flash:w:.build/psuieee_pluto12_encoder.hex:i
avrdude: reading input file .build/psuieee_pluto12_encoder.hex for flash
         with 16610 bytes in 1 section within [0, 0x40e1]
         using 130 pages and 38 pad bytes
avrdude: writing 16610 bytes flash ...
Writing | ################################################## | 100% 1.22 s
avrdude: 16610 bytes of flash written
avrdude: verifying flash memory against .build/psuieee_pluto12_encoder.hex
Reading | ################################################## | 100% 0.11 s
avrdude: 16610 bytes of flash verified

avrdude done.  Thank you.
```

- You can now use your macropad!

Further customization

- Try flashing the encoder_complex keymap with “qmk flash -kb psuieee/pluto12 -km encoder_complex” using the same process as before
    - This keymap has more complex functionality such as layers
- In one of the keymap folders (such as “encoder” or “encoder_complex”), open “keymap.c”. Here you can customize what each key does
- For further customization, reference the QMK documentation here https://docs.qmk.fm/#/