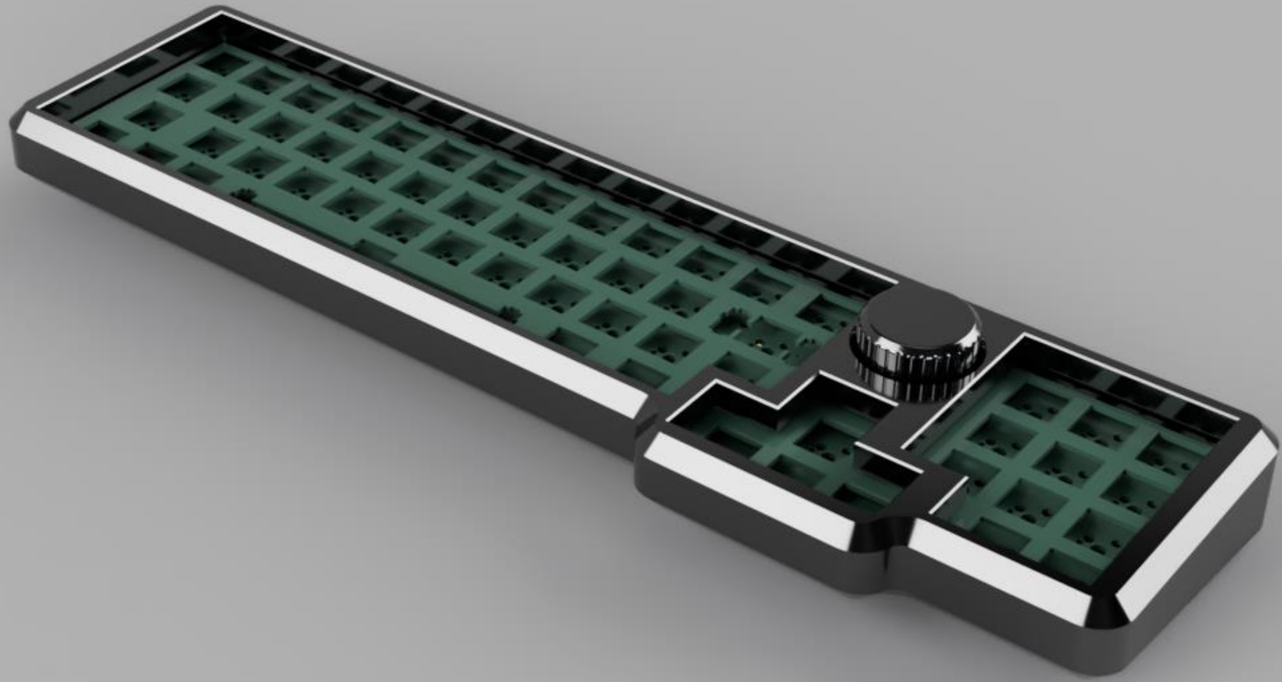
TrashTruck



By: Skycode22

https://github.com/Skycode22/HIDs

TrashTruck supplies

Supplies needed (other than PCB, 3D printed microcontroller holder, and case):

-Elite-C microcontroller

-Kailh hotswap sockets x58

-Diodes x58 (either THT 1N4148 or SMD SOD-123)

-ribbon cable or any other wire to solder the pcb to the Elite-C

-Rotary encoder

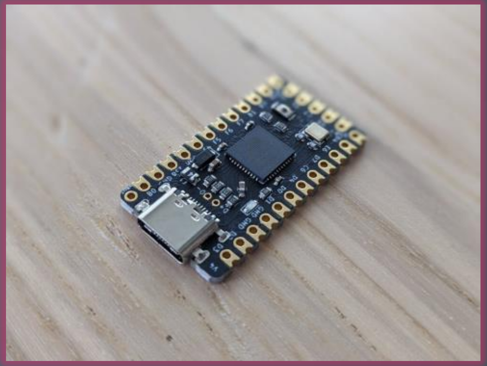
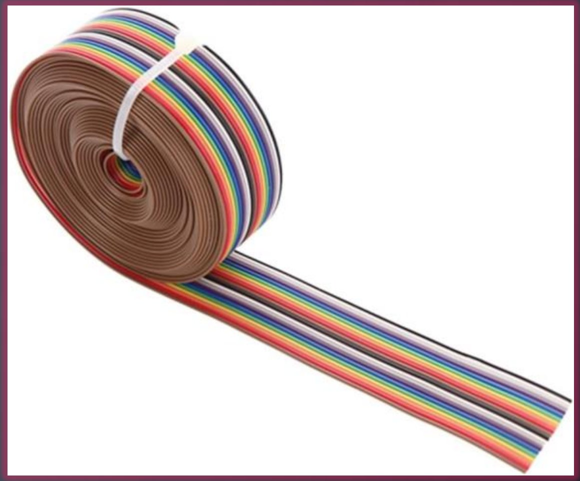
https://www.amazon.com/Cylewet-Encoder-Digital-Potentiometer-Arduino/dp/B07DM2YMT4/ref=sr_1_6?crid=3CTP70L8ADU7H&keywords=encoder&qid=1683396917&sprefix=encoder%2Caps%2C156&sr=8-6

Case screws:

https://www.amazon.com/gp/product/B09ZHHSY92/ref=ox_sc_act_title_1?smid=A3KEIBWKST18IS&psc=1

-x2 25mm 2M machine screws

-x2 15mm 2M machine screws

PCB AND CASE

- The gerber files are included in the repo; use the zipped file named “gerbers” (I used JLCPCB to fabricate mine)
- If you don’t like the silkscreen graphic on the pcb you can remove it in kicad and then regenerate the gerber, drill, and map files; Instructions for this can be found online
- The case can be found in the “production files” folder inside the “3-D” folder in the repo
- Depending on your fabrication process they will need either a STL or STEP file
- If you want to print this at home, there is a folder named “Elegoo Mars Printable” that has the case “cut” into 3 sections that can then be put together after printing *(be aware that the tolerances with 3D printing can vary from printer to printer, so be sure to check your measurements after the final model is printed)
- If the models are a little too long (which is the common error) then sand down the center pieces and measure frequently to achieve the correct size
- If you send the model as a whole to a fab house then you don’t have to worry about that error that commonly occurs when printing it at home

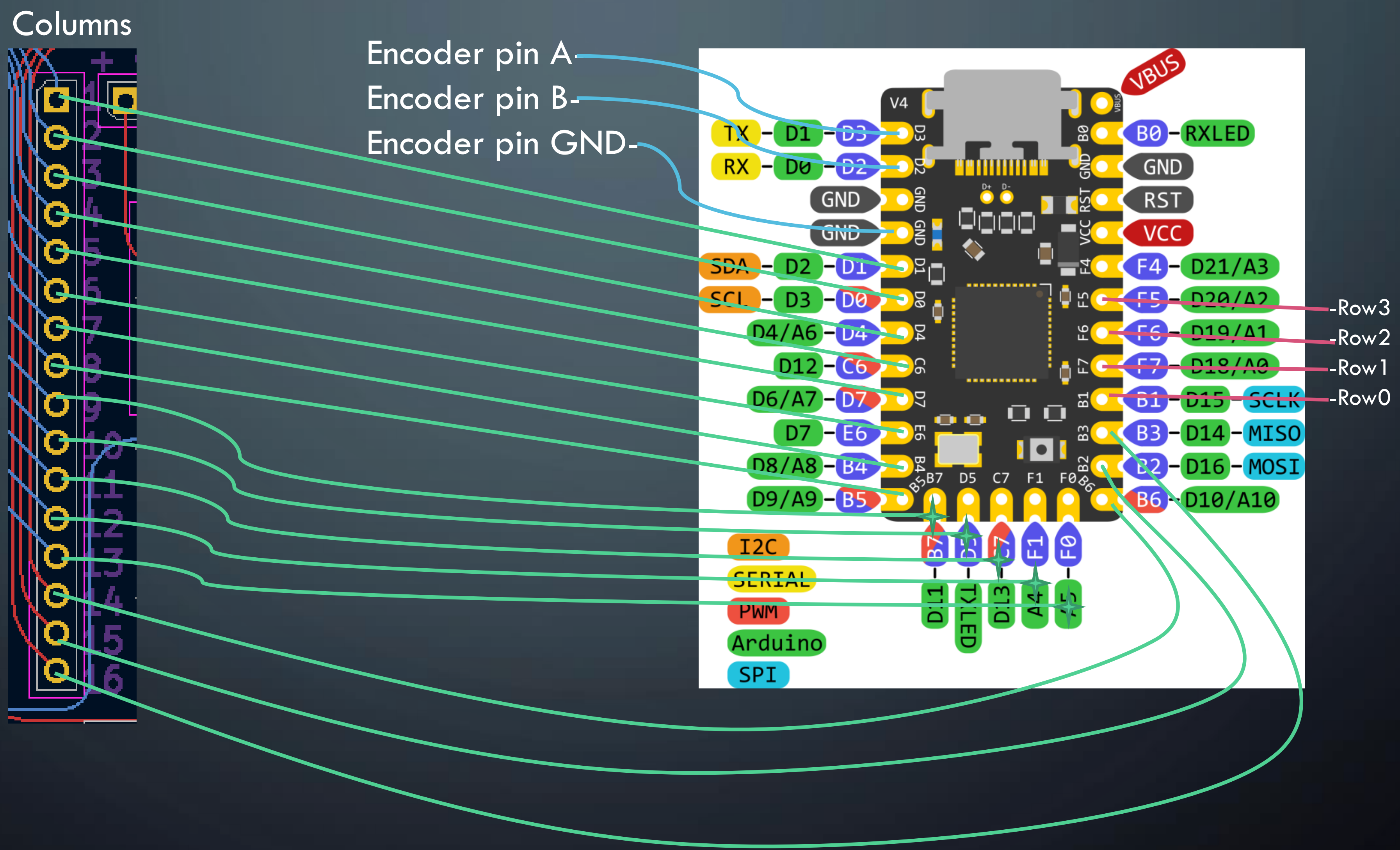
FIRMWARE INSTRUCTIONS

- Firmware does not currently have VIA or VIAL enabled, I plan on including that and/or a reset button to the V2 design
- The .hex file is included in the repo; use QMK toolbox to flash
- If you want to make modifications to the keymap then you can go into the keymap file to make your modifications, recompile the .hex file, and reflash
- The QMK firmware source code is in the JSON format which incorporates the .mk file, config, .c, and .h files all inside the info.json; you need the most up to date versions of the qmk toolbox and the QMK MSYS to recompile*
- Reference the documentation on QMK to see the process for compiling with MSYS; https://docs.qmk.fm/#/newbs_getting_started

OVERVIEW

1. Solder diodes
2. Solder Hotswap sockets
3. Solder encoder to pcb
4. Solder ribbon cables to pcb
5. Solder other end of ribbon cables to corresponding pins on the Elite-C
6. **Reference “Pinout Diagram” slide for wiring diagram**
7. **Reference “Case Assembly” slide for the rest of the build (make sure firmware is flashed to the Elite-C before assembling the rest of the build)**

PINOUT DIAGRAM



Columns

Encoder pin A-

Encoder pin B-

Encoder pin GND-

-Row3

-Row2

-Row1

-Row0

CASE ASSEMBLY

1. Place foam in the case, ensuring that a cut is made in the foam so the ribbon cable fits through
2. Put adhesive gaskets along the edges of the case where the pcb will make contact
3. Put Elite-C into the 3D printed holder, it should snap into place and if it doesn't you can secure it with a small drop of adhesive(hot glue works well)
4. Secure the Elite-C in its slot under the foam in the center of the case
5. Guide the ribbon cable along the channel until it meets up with the keyboard pcb
6. Place the pcb securely into the case
7. Put the top of the case on and screw in from the bottom with the machine screws