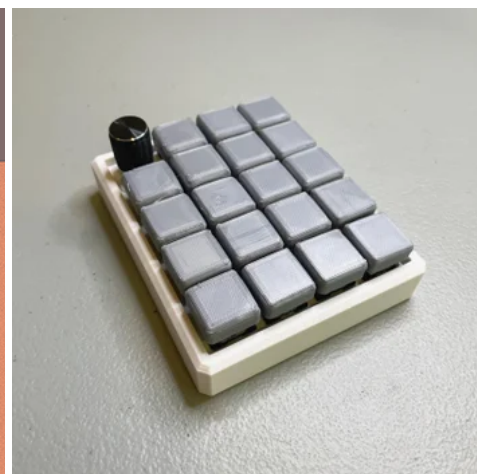
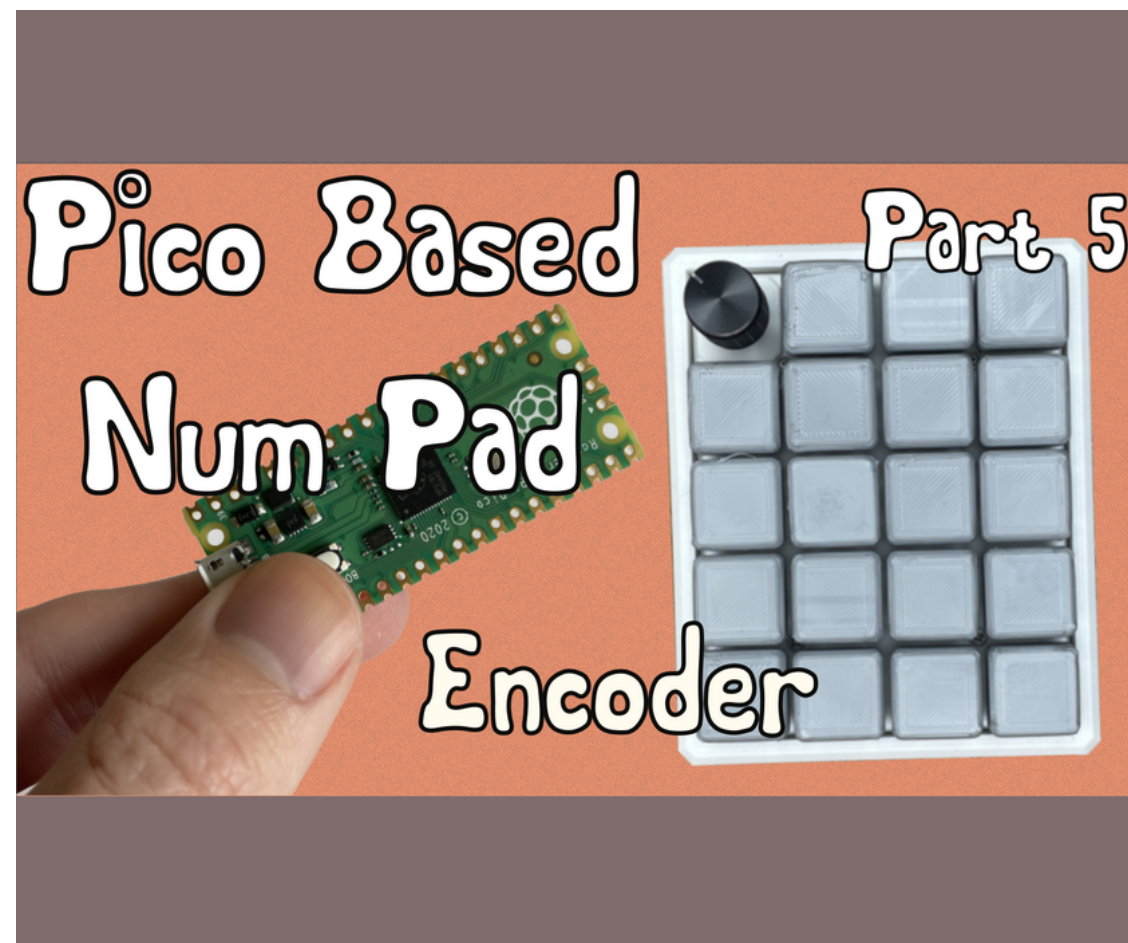


Number Pad Part 5 - Encoder

By [tinyboatproductions](#) in [CircuitsRaspberry Pi](#)



Introduction: Number Pad Part 5 - Encoder



Hey, its me that guy who keeps Instructables on a number pad powered by KMK.

I've been working to make this number pad '*work*' for me, and by that I mean trick it out with all the features I want.

I have two encoders my Avalanche keyboard and it's so useful. Also when I made my two macro pads they both had encoders. They are super useful.

If you watch the video you'll also know we are going to install mouse and media keys in this part. If you haven't watched it, uh, spoilers I guess? But I will be doing all the code at once.

Let's get into it.

Supplies

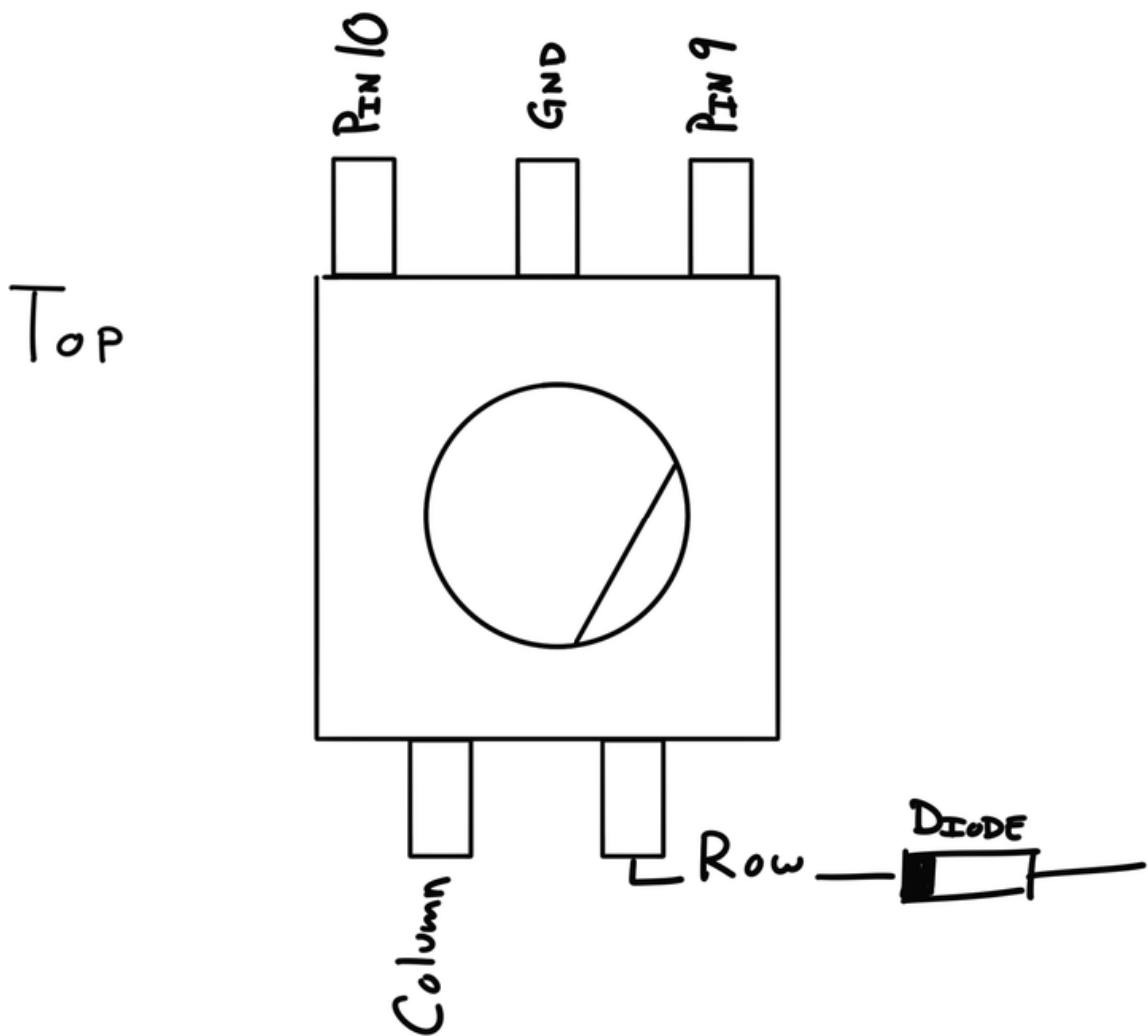
To build this you will either need an existing number pad if you've been following along. I would recommend starting with part one as I do assume some knowledge later.

But the only differences hardware wise is you will need an encoder. I am using one of [these](#), de-soldered from the breakout board.

Also I am using a new case - check it [here](#)

But if you don't want to make a whole new case, try this adapter piece - [here](#)

Step 1: Soldering



From an existing number pad, remove one of the switches and install the encoder in its place using the adapter piece, and wire the side with two leads into the key mesh - see the diagram.

On the side with three leads, wire the outer two directly to the pico and the center one to ground. Again, see the diagram.

Step 2: Imports

At the top of the code we need to add some imports. These will import parts of the code from KMK that we can use.

```
from kmk.modules.encoder import EncoderHandler
from kmk.extensions.media_keys import MediaKeys
from kmk.modules.mouse_keys import MouseKeys
```

Note if you don't plan to use the media or mouse keys you can exclude the relevant lines.

Be sure to check the documentation for the [Encoders](#), [Media Keys](#), and [Mouse Keys](#) from the kmk docs as things can change and this info could be out of date at some point.

Step 3: Define Stuff

We need to create a location for all of our encoders settings but luckily for us KMK has done the heavy lifting so we just need to add one line

```
encoder_handler = EncoderHandler()
```

Now we need to tell the keyboard to actually use the stuff we've defined and imported using the following lines

```
keyboard.modules.append(encoder_handler)
keyboard.modules.append(MediaKeys())
keyboard.modules.append(MouseKeys())
```

That's all the boring stuff lets get to the fun stuff.

Step 4: Tell the Encoder Where It Is

The code knows it should use an encoder now but it has no idea where it is or what to do with it.

We need to add the following line to give the code the pin numbers for the encoder. I used pins 9 and 10 on the pico, swap these for what ever you used.

```
encoder_handler.pins = ((board.GP9,board.GP10,None),)
```

I used the third option None because I wired the click function into the key mesh. Check the documentation if you are using an I2C encoder, or want to check the other options for encoders.

If you have more encoders you can keep adding them after the comma.

Step 5: Encoder Map!

We need to tell the keyboard what to do with the encoder when we move it.

This is like the keymap for the keys.

```
encoder_handler.map = [
    ((KC.VOLD, KC.VOLU),),
    ((KC.MW_UP, KC.MW_DN),),
]
```

Also make sure to swap any keys on the main keymap for the click function.

Step 6: Full Code

Below is the full code I am using all together.

```
import board

from kmk.kmk_keyboard import KMKKeyboard
from kmk.keys import KC
from kmk.scanners import DiodeOrientation
from kmk.modules.encoder import EncoderHandler
from kmk.extensions.media_keys import MediaKeys
from kmk.modules.mouse_keys import MouseKeys

from kmk.modules.layers import Layers

keyboard = KMKKeyboard()

keyboard.col_pins = (board.GP5,board.GP6, board.GP7, board.GP8) # try D5 on Feather, keeboar
keyboard.row_pins = (board.GP11, board.GP12, board.GP2, board.GP3, board.GP4) # try D6 on Feather, keeboar
keyboard.diode_orientation = DiodeOrientation.ROW2COL

encoder_handler = EncoderHandler()

keyboard.modules.append(Layers())
keyboard.modules.append(encoder_handler)
keyboard.modules.append(MediaKeys())
keyboard.modules.append(MouseKeys())

#Encoder
encoder_handler.pins = ((board.GP9,board.GP10,None),)

#HELPER KEYS
TRANS = KC.TRNS
#RAISE = KC.MO(1)
#RAISE = KC.DF(1)
#BASE = KC.DF(0)
RAISE = KC.LT(1, KC.ESC)

keyboard.keymap = [
    [#LAYER 0: BASE
    KC.MPLY, RAISE, KC.TAB, KC.PPLS,
    KC.N7, KC.N8, KC.N9, KC.KP_ASTERISK,
    KC.N4, KC.N5, KC.N6, KC.KP_MINUS,
    KC.N1, KC.N2, KC.N3, KC.KP_SLASH,
    KC.BSPC, KC.N0, KC.DOT, KC.PENT,
    ],
    [#LAYER 1: NAV
    TRANS, TRANS, TRANS, TRANS,#RIGHTS
    KC.HOME, KC.UP, KC.PGUP, TRANS,
    KC.LEFT, TRANS, KC.RIGHT, TRANS,
    KC.END, KC.DOWN, KC.PGDN, TRANS,
    TRANS, TRANS, TRANS, TRANS,
    ],
]

encoder_handler.map = [
    ((KC.VOLD, KC.VOLU),),
    ((KC.MW_UP, KC.MW_DN),),
]

if __name__ == '__main__':
    keyboard.go()
```

Step 7: Conclusion



Well, what are you waiting for, try it out. Don't forget to check out the official KMK documentation for other key codes or settings.

If you have any questions leave them below and I'll try and help

I want to thank you for reading. I post other projects over on my YouTube channel so go take a look over there too. I hope to be back with another project soonish. See you then!