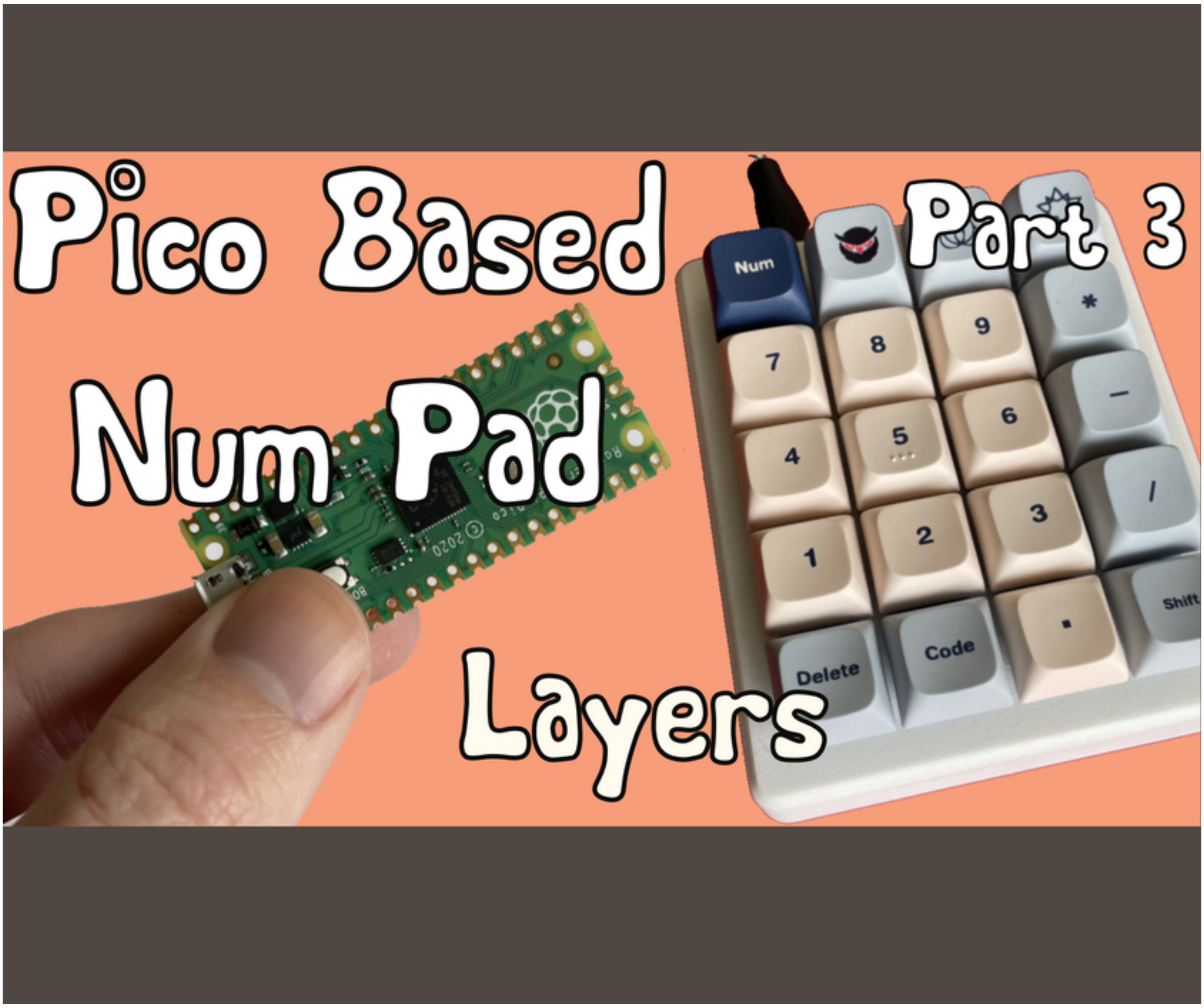
AUTODESK
Instructables

Pico Base Number Pad: Part 3 - Layers

By tinyboatproductions in CircuitsRaspberry Pi

Introduction: Pico Base Number Pad: Part 3 - Layers



Hey all,

This is the next part in my series about building a Pico Based Number Pad. In my last instructable, I build it and installed the code, Part 1, and Part 2.

This is part 3. In this part we will talk about adding layers to get even more out of the number pad.

Supplies

You only need 3 things for this part, the number pad I built last time, a usb cord, and a text editor.

I will be using notepad++, but you can also use notepad, or even VSCode to work on this.

If you do use VS code, check out the Pico-Go add on, it makes it so you can debug really easily. It is not needed but nice to have, and VS code is completely free.

A quick note before we get stuck in, if you build this back when the original instructable came out, I would highly suggest saving your code.py on your computer and reinstalling CircuitPython and KMK as a few of the variables have changed. This is due to KMK still being in development. Nothing major just a few small tweaks. Check out the previous instructable if you need to remember how to do that.

Also going to note up here at the top, there is KMK documentation. Take a look there for more information on everything I talk about. I’ll just link it up here to keep this instructable cleaner.

Step 1: Imports

First we need to import layers into our code so that we can use them.

Here is the code from last time:

```
print(“Starting”)

import board

from kmk.kmk_keyboard import KMKKeyboard
from kmk.keys import KC
from kmk.scanners import DiodeOrientation

keyboard = KMKKeyboard()

keyboard.col_pins = (board.GP0,board.GP1, board.GP2, board.GP3)    # try D5 on Feather, keeboar
keyboard.row_pins = (board.GP4, board.GP5, board.GP6, board.GP7, board.GP8)    # try D6 on Feather, keeboar
keyboard.diode_orientation = DiodeOrientation.ROW2COL

keyboard.keymap = [
    [KC.A,    KC.LSHIFT,  KC.TAB,     KC.KP_PLUS,
     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.KP_DOT,  KC.KP_ENTER,
    ]
]

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

We need to add this line below the imports:

```
from kmk.modules.layers import Layers()
```

So we have

```
…
from kmk.kmk_keyboard import KMKKeyboard
from kmk.keys import KC
from kmk.scanners import DiodeOrientation
from kmk.modules.layers import Layers()
…
```

This line just makes it so that we can use the layers module in our code.

Step 2: Add Layers

Now we need to add the layers to our keyboard so that we can actually use them. Right now the code knows they exist but the keyboard doesn’t. So lets tell it using:

```
keyboard.modules.append(Layers())
```

I add this bellow the diode line like this:

```
…
keyboard.diode_orientation = DiodeOrientation.ROW2COL

keyboard.modules.append(Layers())
…
```

Now the keyboard knows layers are something it can use.

Step 3: Helper Keys

Before going much farther I am going to create 2 helper keys. These keys simplify the code later and are just easier to work with.

The two keys are a transparent key and a raise key.

```
TRANS = KC.TRNS
RAISE = KC.MO(1)
```

The first one is a transparent key, basically a blank key. This make it so when we activate a layer we don’t have to define all the keys if we don’t want or need too.

The second, raise, gives us a way to access the layers we add. I have used KC.MO() this activates layer 1, the number we give it, until we let go. This is very similar to how the shift key allows you to type 6 and ^ on the same key. Think of it like a shift button.

Step 4: Add a Keymap

Ok now we have everything set to add the layer, we just need a key map. I like to copy an existing layer and change out the key codes that I need to make it work. Also, make sure to add a comma (,) after the base layer bracket].

The next layer I am going to make is going to be a navigation layer so I am also going to note that, it just makes it easier to read later. Here are my key maps, after adding layer 1.

```
  #LAYER 0: BASE
    [KC.A,      KC.LSHIFT,  KC.TAB,     KC.KP_PLUS,
     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.KP_DOT,  KC.KP_ENTER,
    ],
    #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,
    ],
```

Notice also that I added a comma after layer 1. This is incase you want to add more layers later.

Also a note here is that you want to make sure that on the layer you are going to, that key is transparent, so the upper left key on layer 1 should be transparent. This is just good practice and prevents any key conflicts.

Step 5: Activate the New Layer

With all of that done we just need a way to get to layer 1. Right now it’s all setup but we dont have any way to access it.

That is where the RAISE key comes in. I am going to replace the KC.A in the upper left with RAISE. This means I can press that key and go to layer 1 until I let go.

Step 6: Alternate Layer Shifting Behavior

Let’s also say, you dont want to have a momentary layer shift. There are a few other ways to access higher layers.

The first is to make basically a latching layer swap, like caps lock, it makes it so you can access all the capital letters until you turn it off.

To do that, we just need to swap our RAISE for the one below, and create a BASE helper to get back to the base layer.

Also be sure to add the BASE to your layer 1 key map otherwise you’ll be stuck on layer 1 until you restart the board.

```
RAISE = KC.DF(1)
BASE = KC.DF(0)
```

Here I swap the MO for DF. This changes the default layer, or makes a latching key.

The third option, the one I use is to make a tap/hold key. This is a key that, when tapped sends a key code but when held shifts layers.

```
RAISE = KC.LT(1, KC.A)
```

This will shift to layer 1 when held but will send an A when tapped. This behaves similar to the momentary shift only activating the layer when the button is held.

Note here that with any of these the number in parentheses is the layer we are going to. Also the layers are numbered starting at 0, so the base layer is the 0th layer and the nav layer is the 1st layer. That’s why I like to add the layer number and a note about what its for before each layer.

Step 7: Full Code

Here is the full code for this project, with layers.

```
print("Starting")

import board

from kmk.kmk_keyboard import KMKKeyboard
from kmk.keys import KC
from kmk.scanners import DiodeOrientation
from kmk.modules.layers import Layers

keyboard = KMKKeyboard()

keyboard.col_pins = (board.GP0,board.GP1, board.GP2, board.GP3)    # try D5 on Feather, keeboar
keyboard.row_pins = (board.GP4, board.GP5, board.GP6, board.GP7, board.GP8)    # try D6 on Feather, keeboar
keyboard.diode_orientation = DiodeOrientation.ROW2COL

#board modules
keyboard.modules.append(Layers())

#Helper key codes
TRANS = KC.TRNS
RAISE = KC.MO(1)
#RAISE = KC.DF(1)
#BASE = KC.DF(0)
#RAISE = KC.LT(1, KC.A)

keyboard.keymap = [
    #LAYER 0: BASE
    [RAISE,     KC.LSHIFT,  KC.TAB,     KC.KP_PLUS,
     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.KP_DOT,  KC.KP_ENTER,
    ],
    #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,
],

]

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

Step 8: Parting Thoughts

That’s all I have for this section.

Please feel free to leave a question or comment below. I will try my best to help if you run into issues.

There is also a KMK discord group you can join. They are really helpful.

Otherwise, go check out my YouTube channel and follow me on instagram.

Until next time!