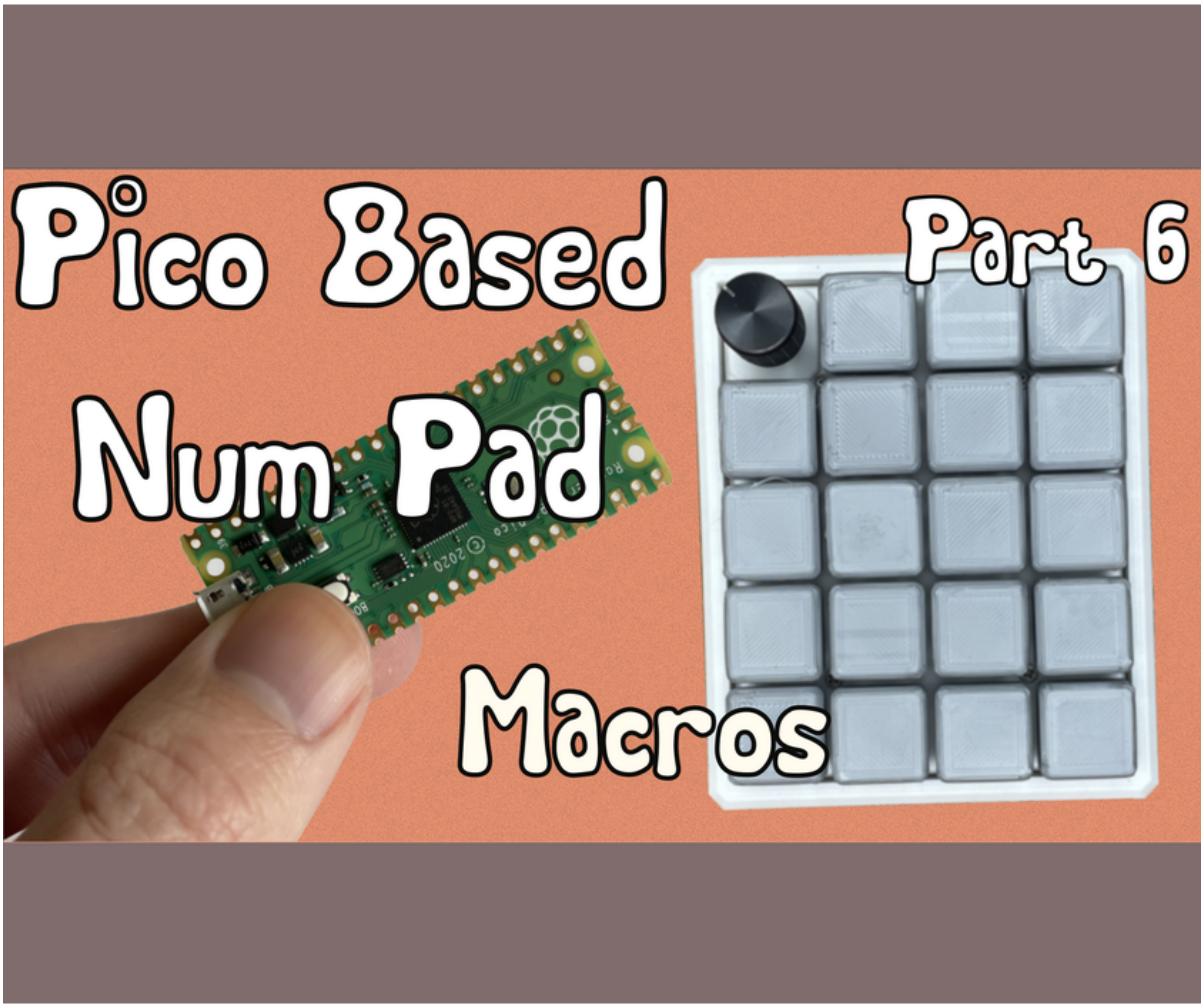
Pico Powered Number Pad - Part 6 - Macros

By tinyboatproductions in CircuitsComputers

Introduction: Pico Powered Number Pad - Part 6 - Macros



Hey y'all,

I'm back with another part of this number pad series. This has been an on going project for me for a while. I finally decided it was time to add macros, as well as an auto-clicker to this to make it into a macro pad.

As longtime readers will know, macro pads were some of my first projects.

Before we get into the 'how to' I want to touch on some topics I won't be covering. I won't be going over how to build the number pad I am working on. Checkout part 1 and 2 for more on that.

I also won't be covering some of the more basic modules on this like layers, check out part 3 for that.

In general, I think I have covered most of what is in the starting code in a previous part.

Also, this should work on any keyboard or HID device running KMK.

Finally, when in doubt, check the KMK docs, I'm recording/writing this in late 2023/early 2024. You can also leave a comment here or on YouTube.

Supplies

The supplies for this project are pretty minimal if you already have a keyboard/HID device running on KMK firmware.

If you don't already have one I can recommend, the FITZ_43 and the number pad I am putting this code onto.

You'll also a computer with text editor and usb cable.

Step 1: Simple String

Lets start with just sending a simple string of text. Before I show you how you have to promise you won't put a password in here. Storing your passwords in plan text is just a bad idea, don't do it.

With that out of the way, lets get started. Start by importing the send_string component to the keyboard at the top (don't worry, I'll have the full code attached at the bottom.)

```
from kmk.handlers.sequences import send_string
```

With that done, we don't have to attach this to the keyboard like we have for some other modules.

Now, I like to create an special area for my macros so they stay contained and keep my code easy to read. (If you want to go all out you can even create another file, and import that, but that's a whole other topic).

```
#Macros
aString = send_string("Your message here!")
```

I create a helper keycode with a name I can use later, in this case "aString".

Then add send_string("") with your message in the quotes.

And bam, add the helper code to your keymap, (I added a new layer for my macros), and you should be good to go!

Step 2: Simple Key Combo

But, what if you don't to just send text but key combos, I'll show two here, but these can be extended for more functionality. Check out the docs for some of that.

Again, we start with an import

```
from kmk.handlers.sequences import simple_key_sequence
```

First I am going to add the SAVE command or CTRL+S using the below.

```
SAVE = simple_key_sequence(
  (
    KC.LCTL(KC.S),
  )
)
```

Note here that the layout is very important. I don't know why but, if you don't do it like that it doesn't work. And don't forget the trailing comma, I spent way too long chasing my tail on that.

Again, add the SAVE keycode to your keymap and its ready.

Next, I'll add the save as command, CTRL+SHIFT+S

```
SAVEAS = simple_key_sequence(
  (
    KC.LCTL(KC.RSFT(KC.S)),
  )
)
```

This is almost the same as the SAVE command with just an extra layer of keycodes.

Step 3: Rapid Fire

I think this is the function I use the most as I got maybe too into cookie clicker recently.

This like the others starts with an import

```
from kmk.modules.rapidfire import RapidFire
```

Now we need to attach it to the keyboard

```
keyboard.modules.append(RapidFire)
```

Now we have access to the a single keycode KC.RF(). To use this we need to define what it does. I am going to add this to the macros section and add some settings, I'll discuss what they do after.

```
LMB_SPAM = KC.RF(KC.MB_LMB, timeout = 10, interval = 100)
```

OK, that's a lot of stuff what does it all mean?

First off, "LMB_SPAM" is a short hand for the key that I can add to my keymap, instead of putting the whole right part into my keymap.

"KC.RF(" is the keycode we added above, the rapidfire key.

"KC.MB_LMB' is the key I want to send, in this case it is the left mouse button. (I added mouse buttons in part 5)

"timeout = 10" is the time you need to hold the key before the key is fired, in this case I set it to 10 milliseconds.

"interval = 100" is the time between clicks. It is important to note here that 10 milliseconds is as low as you can go. I did try some lower values, and it just doesn't work, it sends the key once or twice and then stops. Also, if you run a click speed tester at 10 you will get a value of around 62 clicks per second instead of the expected 100. I don't know why, but I expect its a limitation either with the pico, KMK, Windows, or a combination.

Another option to add here is '"toggle = True" this would be added at the end, this turns this into a 'latching' key. Once you press it it will keep firing the key until you press it again. Kind of fun, but always be careful when telling your computer to hold down a key forever......(or until you unplug the pico).

There are a few other options to use so check out the docs for that.

Step 4: Full Code

This is just a step with the full code

Step 5: Final Thoughts

I though this was a great way to extend the capabilities of the number pad I built.

I like the idea of adding macros/rapidfire to my keyboards to make using my computer easier.

Thanks for reading!

Until next time ~