

Matrix Keypad

Created by Kattni Rembor



Last updated on 2018-11-12 09:12:26 PM UTC

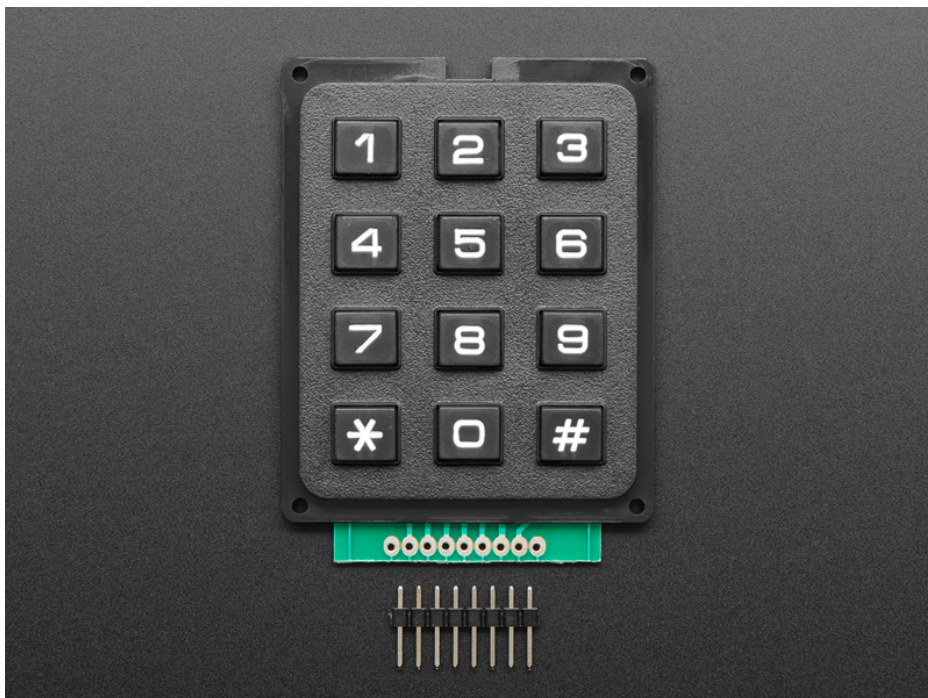
Guide Contents

Guide Contents	2
Overview	3
Python & CircuitPython	5
CircuitPython Microcontroller Wiring	5
Python Computer Wiring	5
CircuitPython Installation of Matrix Keypad Library	5
Python Installation of Matrix Keypad Library	6
CircuitPython & Python Usage	6
Full Example Code	7
Python Docs	10

Overview



Hey, Jenny, I've got your number! And I'm going to punch it into one of these nice matrix keypads. We have a bunch of different options in 3x4- and 4x4-grid styles. These are an easy way to add some input options to your project!



The keys are connected into a matrix, so for the 3x4 matrix keypads, you only need 7 microcontroller pins (3-columns and 4-rows) to scan through the pad. On the 4x4 matrix keypads, you only need 8 microcontroller pins (4-columns and 4-rows) to scan through the pad.

We have libraries in Arduino and CircuitPython to make getting started super simple and fast.



So wire it up to your project, and punch in your secret key to open all the possibilities!

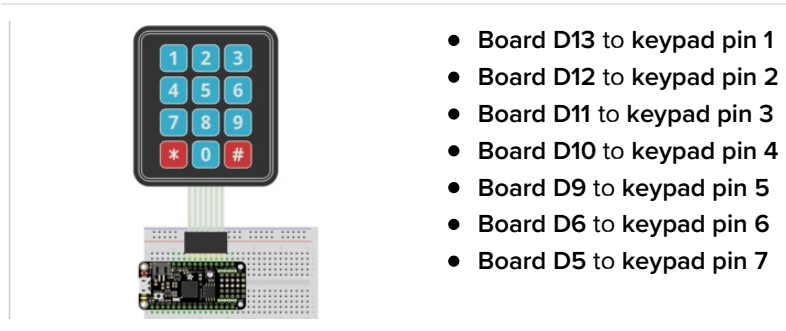
Python & CircuitPython

It's easy to use a matrix keypad with Python or CircuitPython and the [Adafruit CircuitPython Matrix Keypad \(https://adafru.it/CUo\)](https://adafru.it/CUo) module. This module allows you to easily write Python code that reads the button presses from the keypad.

You can use this with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to [Adafruit_Blinka, our CircuitPython-for-Python compatibility library \(https://adafru.it/BSN\)](https://adafru.it/BSN).

CircuitPython Microcontroller Wiring

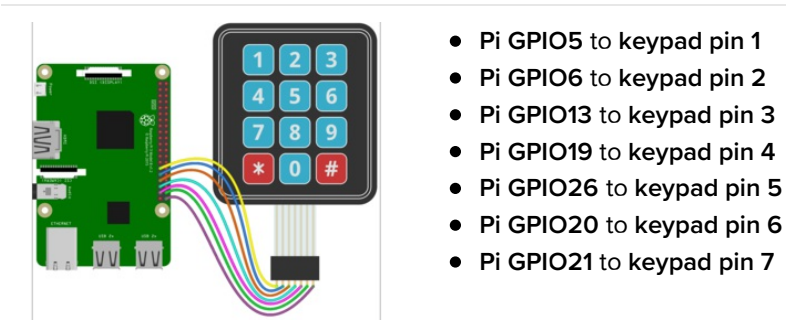
First wire up a matrix keypad to your board. Here's an example of wiring a 3x4 matrix to a Feather M0. You can use any free digital I/O pins. The keypad pins are referenced left to right if the keypad is oriented upright and facing you, i.e. pin 1 is the first pin on the side of the header closest to the 1-column on the keypad.



Python Computer Wiring

Since there's *dozens* of Linux computers/boards you can use we will show wiring for Raspberry Pi. For other platforms, [please visit the guide for CircuitPython on Linux to see whether your platform is supported \(https://adafru.it/BSN\)](https://adafru.it/BSN).

Here's an example of a 3x4 matrix keypad wired to the Raspberry Pi. You can use any free digital I/O pins. The keypad pins are referenced left to right if the keypad is oriented upright and facing you, i.e. pin 1 is the first pin on the side of the header closest to the 1-column on the keypad.



CircuitPython Installation of Matrix Keypad Library

You'll need to install the [Adafruit CircuitPython Matrix Keypad \(https://adafru.it/CUo\)](https://adafru.it/CUo) library on your CircuitPython board.

First make sure you are running the [latest version of Adafruit CircuitPython \(https://adafru.it/Amd\)](https://adafru.it/Amd) for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from [Adafruit's CircuitPython library bundle \(https://adafru.it/uap\)](https://adafru.it/uap). Our CircuitPython starter guide has [a great page on how to install the library bundle \(https://adafru.it/ABU\)](https://adafru.it/ABU).

For non-express boards like the Trinket M0 or Gemma M0, you'll need to manually install the necessary libraries from the bundle:

- `adafruit_matrixkeypad.mpy`

Before continuing make sure your board's lib folder or root filesystem has the `adafruit_matrixkeypad.mpy` file copied over.

Next [connect to the board's serial REPL \(https://adafru.it/Awz\)](https://adafru.it/Awz) so you are at the CircuitPython >>> prompt.

Python Installation of Matrix Keypad Library

You'll need to install the Adafruit_Blinka library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. [Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready \(https://adafru.it/BSN\)](https://adafru.it/BSN)!

Once that's done, from your command line run the following command:

- `sudo pip3 install adafruit-circuitpython-matrixkeypad`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython & Python Usage

To demonstrate the usage of a matrix keypad, we'll initialise it and read the button presses from the board's Python REPL.

First run the following code to import the necessary libraries:

```
import time
import digitalio
import board
import adafruit_matrixkeypad
```

Next we're going to identify what pins are associated with the rows and columns, and set up the order of the keys:

```
cols = [digitalio.DigitalInOut(x) for x in (board.D9, board.D6, board.D5)]
rows = [digitalio.DigitalInOut(x) for x in (board.D13, board.D12, board.D11, board.D10)]
keys = ((1, 2, 3),
        (4, 5, 6),
        (7, 8, 9),
        (*, 0, '#'))
```

Then we initialise the matrix:

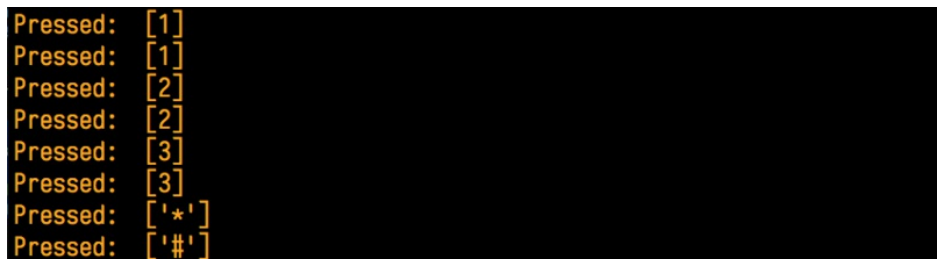
```
keypad = adafruit_matrixkeypad.Matrix_Keypad(rows, cols, keys)
```

Now you're ready to utilise the keypad with the following property:

- **pressed_keys**: An array containing all detected keys that are pressed from the initialised list-of-lists passed in during creation.

For example, to print all the key presses:

```
while True:
    keys = keypad.pressed_keys
    if keys:
        print("Pressed: ", keys)
    time.sleep(0.1)
```



```
Pressed: [1]
Pressed: [1]
Pressed: [2]
Pressed: [2]
Pressed: [3]
Pressed: [3]
Pressed: ['*']
Pressed: ['#']
```

That's all there is to using a matrix keypad with CircuitPython!

Full Example Code

For CircuitPython microcontroller boards:

```

import time
import digitalio
import board
import adafruit_matrixkeypad

# Membrane 3x4 matrix keypad - https://www.adafruit.com/product/419
cols = [digitalio.DigitalInOut(x) for x in (board.D9, board.D6, board.D5)]
rows = [digitalio.DigitalInOut(x) for x in (board.D13, board.D12, board.D11, board.D10)]

# 3x4 matrix keypad - Rows and columns are mixed up for https://www.adafruit.com/product/3845
# Use the same wiring as in the guide with the following setup lines:
# cols = [digitalio.DigitalInOut(x) for x in (board.D11, board.D13, board.D9)]
# rows = [digitalio.DigitalInOut(x) for x in (board.D12, board.D5, board.D6, board.D10)]

keys = ((1, 2, 3),
        (4, 5, 6),
        (7, 8, 9),
        (*, 0, #))

keypad = adafruit_matrixkeypad.Matrix_Keypad(rows, cols, keys)

while True:
    keys = keypad.pressed_keys
    if keys:
        print("Pressed: ", keys)
        time.sleep(0.1)

```

For Raspberry Pi:

```

import time
import digitalio
import board
import adafruit_matrixkeypad

# Membrane 3x4 matrix keypad on Raspberry Pi -
# https://www.adafruit.com/product/419
cols = [digitalio.DigitalInOut(x) for x in (board.D26, board.D20, board.D21)]
rows = [digitalio.DigitalInOut(x) for x in (board.D5, board.D6, board.D13, board.D19)]

# 3x4 matrix keypad on Raspberry Pi -
# rows and columns are mixed up for https://www.adafruit.com/product/3845
# cols = [digitalio.DigitalInOut(x) for x in (board.D13, board.D5, board.D26)]
# rows = [digitalio.DigitalInOut(x) for x in (board.D6, board.D21, board.D20, board.D19)]

keys = ((1, 2, 3),
        (4, 5, 6),
        (7, 8, 9),
        (*, 0, #))

keypad = adafruit_matrixkeypad.Matrix_Keypad(rows, cols, keys)

while True:
    keys = keypad.pressed_keys
    if keys:
        print("Pressed: ", keys)
        time.sleep(0.1)

```


Python Docs

[Python Docs \(https://adafru.it/C97\)](https://adafru.it/C97)