

CircuitPython and Mu Talk

CircuitPython

What is CircuitPython? Where did it come from?

CircuitPython is a port of MicroPython

MicroPython created in 2013 to run on micros with limited resources (RAM, storage, clock speed, etc)

Scott Shawcroft of Adafruit started CircuitPython in 2016 to port MicroPython to SAMD21

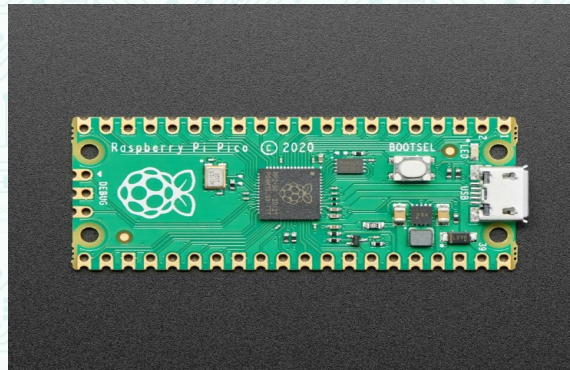
Based on Python 3.4 and beyond

Needs minimum 48MHz and above, 32K RAM (64K recommended), 256KB onboard flash memory

Blinka wrapper library to allow CircuitPython to run on SBC's such as Raspberry Pi's and BeagleBones

CircuitPython developer prioritizes boards that have built in USB – SAMDx.5, ESP32-

2



How did I find CircuitPython?

I discovered CircuitPython when I ordered an AdaBox subscription and the first one included a CircuitPlayground Express with CircuitPython pre-loaded – FUN!

Why Python for Microcontrollers?

Python is way easier to develop with than C or assembly

Has garbage collection and no pointers

Has built in exceptions

Has tons of libraries and examples for all sorts of sensors and devices

Supports wireless and wired ethernet for boards that have those

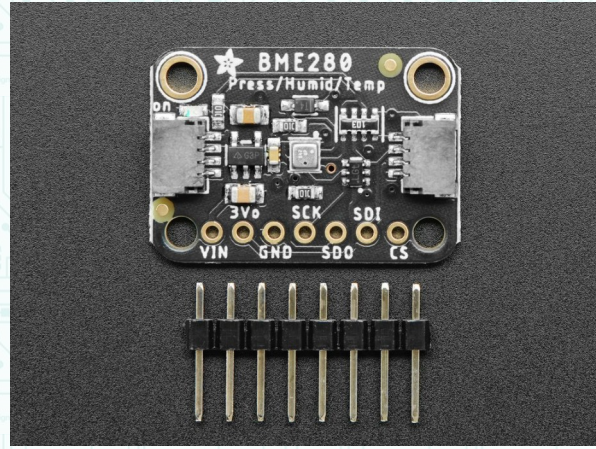
USB built in and just works

What can it do?

Libraries – supports tons of sensors, displays, and actuators

Examples – provides sample code for most if not all devices

**Supports USB out of the box –
Adafruit dropped support for the 8266 because of lack of on-chip USB support**



Documentation – CircuitPython.org, Adafruit.com,

Where can I get it?

CircuitPython is available at www.circuitpython.org

Run through download bootloader

**first, head over to circuitpython.org/downloads – Python support for your board
find the board that you want to use and download the uf2 bootloader file – over 240 board supported!**

install bootloader:

double-tap reset button, CIRCUITPY drive changes to a drive specific to your board

in my case, RPI-RP2

click on the downloaded .uf2 file and drag it to the USB drive

board will reboot itself, and you'll be back to having a CIRCUITPY drive

Download libraries – Python support for your sensors, displays, etc

Download the library bundle that corresponds to the version of CircuitPython .uf2 file you installed and extract them somewhere

Copy over libraries you'll need to CIRCUITPYTHON/lib/

Also might be handy to copy over example files as space on your device permits

Different Versions – 6.x vs 7.x are both

SBCs

<https://learn.adafruit.com/circuitpython-on-raspberrypi-linux>



Mu

“A simple Python editor for beginner programmers”

Mu is the preferred IDE according to the folks at CircuitPython

Easy to learn, easy to use, very simple

Available for just about any OS (not sure about FreeBSD...)

Windows, MacOS, CentOS, Fedora, Raspberry Pi OS, Snap Store for Ubuntu, etc

Can be installed as a Python package via PIP

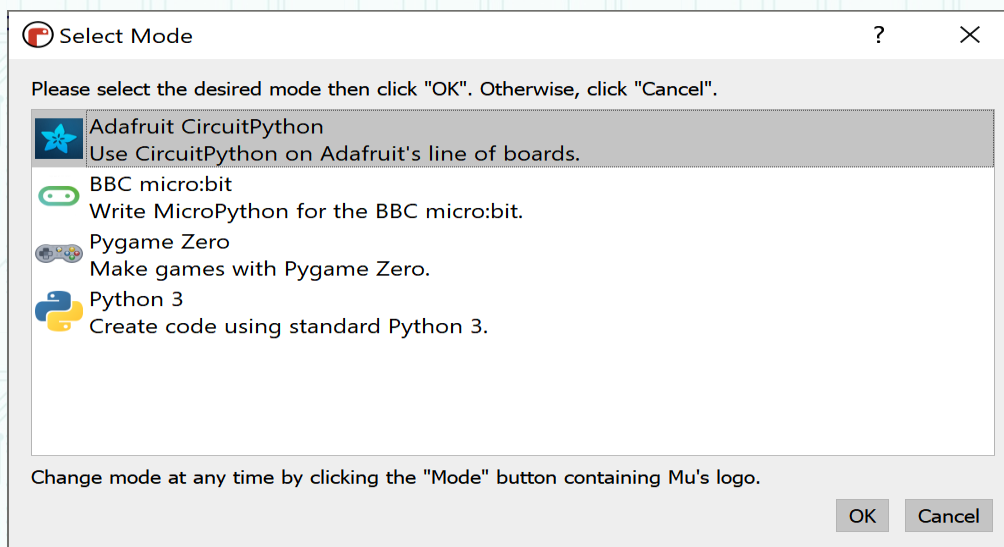
Raspberry Pi OS comes with Mu “Out of the Box”



Mu can be used for Python (uses JupyterHub), CircuitPython, MicroPython and even has a view for the Flask framework

Download from codewith.mu/en/download

After Mu starts, select Mode:

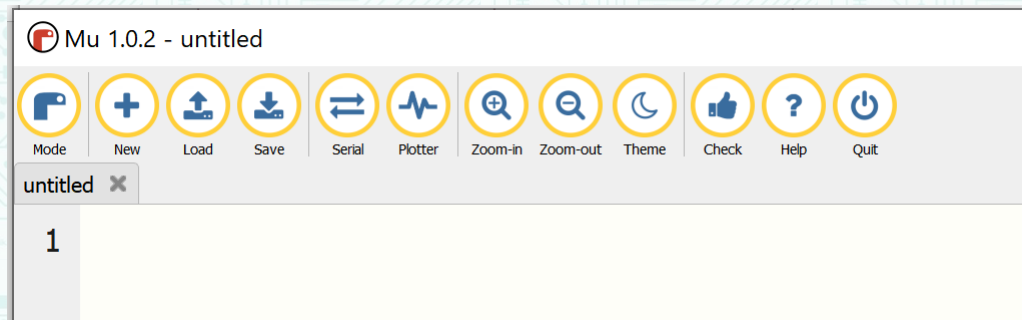


CircuitPython and Mu Together

Microcontroller appears as a small USB hard drive

make a Python program and save it to micro

Mu Control



New – start a new file

Load – Load a file already saved on your micro into the Mu editor

Save – Save a file from the Mu editor onto your micro

Serial – Open the REPL terminal (Read, Evaluate, Print, Loop)

Plotter – plots tuples onto a graph – scales automatically

Zoom in/out – fonts bigger or smaller

Theme – Light, Dark, or High Contrast

Check – check for syntax, also seems to check for best practices for formatting (spacing, etc)

Help – super simple tutorial

Note that the title bar will look slightly different depending on version and if you use one of the other modes. Example, Python 3 mode has a debugging button that doesn't apply to micros.

Let's look at some code:

diff CircuitPython & Arduino

Arduino Code for
Distance Sensor HC-0SR04

```
void setup() {  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  float duration, distance;  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  
  duration = pulseIn(echoPin, HIGH);  
  distance = (duration / 2) * 0.0344;  
  
  Serial.print(distance);  
  Serial.print(" cm \n");  
  delay(500);  
}
```

CircuitPython Code for
Distance Sensor HC-0SR04

```
import time  
import board  
import adafruit_hcsr04  
  
sonar = adafruit_hcsr04.HCSR04(trigger_pin=board.D0, echo_pin=board.D1)  
  
while True:  
    print((sonar.distance,))  
    time.sleep(0.1)
```



```
"""
```

Demonstrate a blinking LED on an ItsyBitsy M4 board

```
"""
```

```
import board
import displayio
import terminalio
import digitalio
import time
```

```
led = digitalio.DigitalInOut(board.D13)
led.direction = digitalio.Direction.OUTPUT
```

```
while True:
    led.value = True
    time.sleep(.5)
    led.value = False
    time.sleep(.5)
    pass
```

Shout out to Adafruit and Lady Ada