BASICS OF HARDWARE WITH A PYRULER

Brought to you by Kira Hartlage

ENTER

INTRODUCTION

Hi! I'm Kira Hartlage.

- Mechanical engineering background
- Consumer appliance design
- Self-taught software engineer
- Embedded software development

DESIRED OUTCOMES

EMBEDDED

Embedded firmware, software, systems?!

CIRCUITPYTHON

You don't need to learn C!

PYRULER

Let's make something happen

RESOURCES

You can do it too

EMBEDDED SYSTEMS

01

02

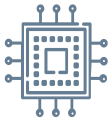
03

04

"An embedded system is a computerized system that is purpose-built for its application."

- Elecia White

HARDWARE



MICROCONTROLLERS

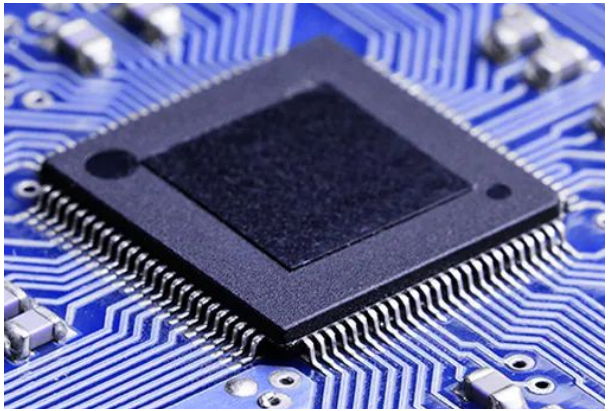
An integrated circuit with a processor, memory, and various input/output peripherals

PERIPHERALS

Various parts of the microcontroller that interface with the outside world

HARDWARE - MICROCONTROLLERS

1. Instructions - what the microcontroller is able to do
   a. Instructions operate on numbers that are stored in registers and memory
2. Registers - fast storage that the micro has that the instructions can use
3. Memory - storage for memory, but slower than registers



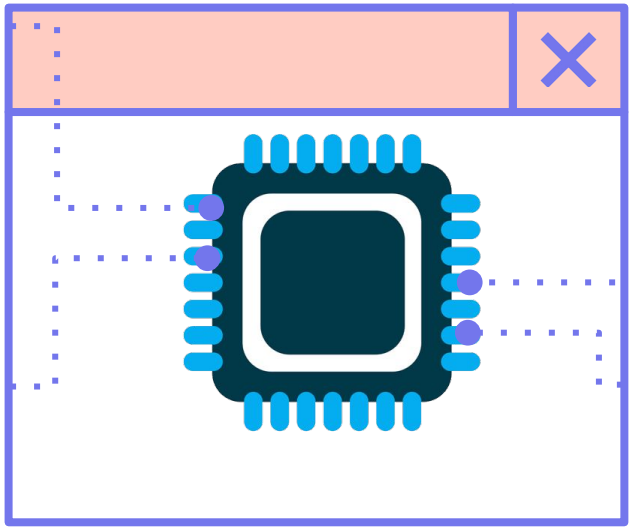
HARDWARE - PERIPHERALS

GPIO

General purpose input/output

ADC

Analog to digital control



PWM

Pulse-width modulation

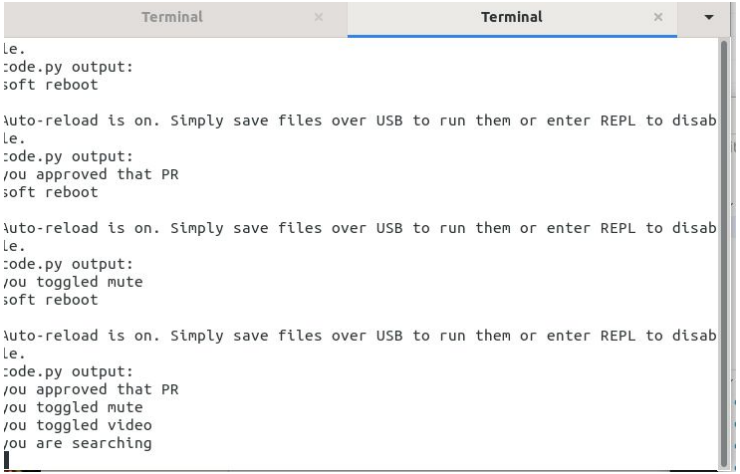
UART

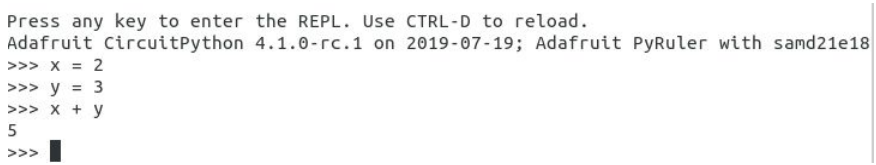
Universal asynchronous receiver-transmitter

CIRCUITPYTHON

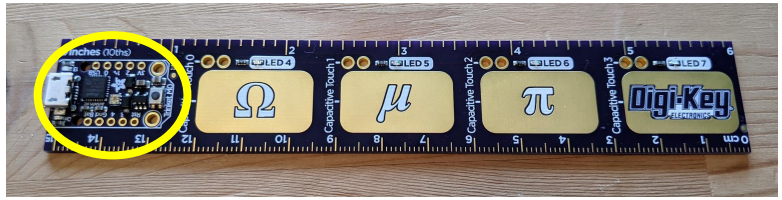
https://learn.adafruit.com/assets/49441

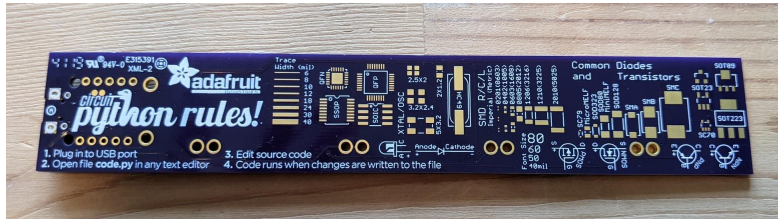
FILE STRUCTURE & SERIAL CONSOLE





PYRULER





PyRuler is a reference board with a Trinket M0 (a Cortex M0) microcontroller.

PYRULER PROJECT

01

02

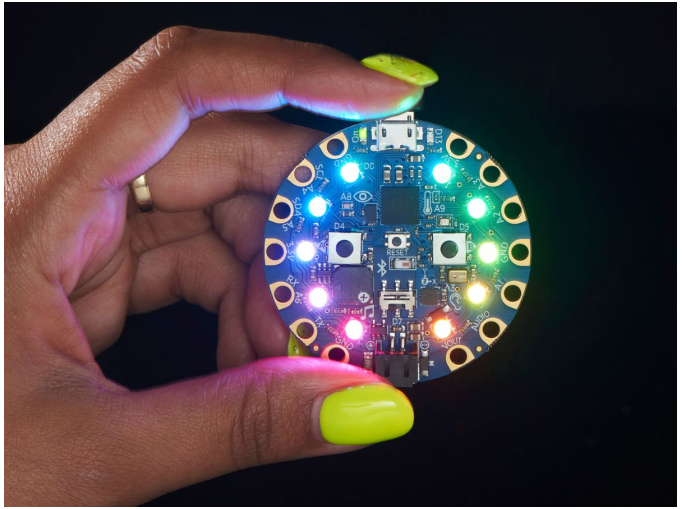
03

04

Adapted from an existing project here: https://learn.adafruit.com/PyRulerVideoPanic/code

CIRCUIT PLAYGROUND EXPRESS/BLUEFRUIT

- nRF52840 microcontroller
- Bluetooth Low Energy support for wireless connectivity
- 10 x mini NeoPixels, each one can display any color
- 1 x Motion sensor
- 1 x Temperature sensor (thermistor)
- 1 x Light sensor (phototransistor)
- 1 x Sound sensor (MEMS microphone)
- 1 x Mini speaker with class D amplifier
- 2 x Push buttons
- 1 x Slide switch
- And more!



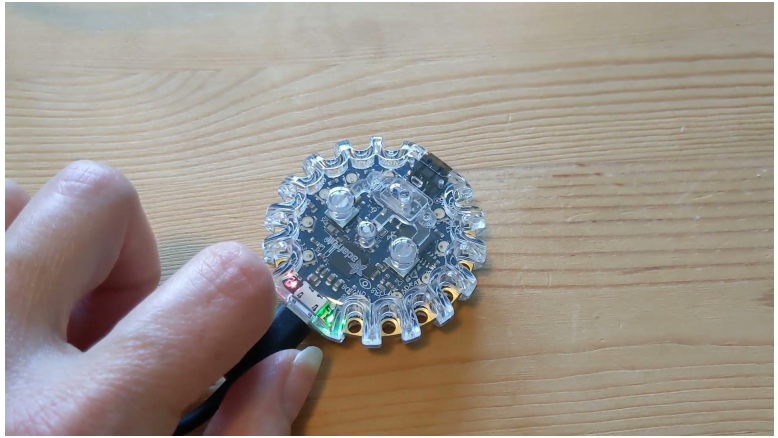
https://learn.adafruit.com/assets/80528

CIRCUIT PLAYGROUND EXPRESS/BLUEFRUIT

```
from adafruit_circuitplayground import cp
import time

cp.pixels.brightness = 0.1

while True:
    if cp.button_a:
        cp.play_mp3("radio-tune.mp3")
    if cp.button_b:
        cp.play_mp3("punch.mp3")
    if cp.loud_sound(sound_threshold = 250):
        cp.pixels.fill((50, 0, 50))
        time.sleep(0.2)
    else:
        cp.pixels.fill((0, 0, 0))
    cp.red_led = cp.switch
```



CIRCUIT PLAYGROUND EXPRESS/BLUEFRUIT PROJECT

Multi-tasking and scheduling can be difficult when you have one main loop.

One solution is using a software timer and handling events asynchronously.

RESOURCES

GITHUB REPO - with code and list of more links!

- https://github.com/kirakirakira/python-hardware-pyruler

ADAFRUIT

- https://learn.adafruit.com/

01

02

03

04

THANKS!

kira.hartlage@gmail.com
github: kirakirakira

PYRULER PROJECT

01

02

03

04

```python
# Modified from https://learn.adafruit.com/PyRulerVideoPanic/code
# Keyboard shortcuts work in Microsoft Teams

import os
import board
from digitalio import DigitalInOut, Direction
import time
import touchio
import adafruit_dotstar

leddot = adafruit_dotstar.DotStar(board.APA102_SCK, board.APA102_MOSI, 1)
leddot[0] = (128, 0, 128)
leddot.brightness = 0.3

# Set this to True to turn the touchpads into a keyboard
ENABLE_KEYBOARD = True

# Used if we do HID output, see below
if ENABLE_KEYBOARD:
    from adafruit_hid.keyboard import Keyboard
    from adafruit_hid.keycode import Keycode
    from adafruit_hid.keyboard_layout_us import KeyboardLayoutUS
    kbd = Keyboard()
    layout = KeyboardLayoutUS(kbd)
```

Import helpful libraries

Set up LED

Enable keyboard input

PYRULER PROJECT


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

touches = [DigitalInOut(board.CAP0)]
for p in (board.CAP1, board.CAP2, board.CAP3):
    touches.append(touchio.TouchIn(p))

leds = []
for p in (board.LED4, board.LED5, board.LED6, board.LED7):
    led = DigitalInOut(p)
    led.direction = Direction.OUTPUT
    led.value = True
    time.sleep(0.25)
    leds.append(led)
for led in leds:
    led.value = False
```

Set up D13 LED as output

Record touches

Turn on all LEDs then turn them off

PYRULER PROJECT

01

02

03

04

```
cap_touches = [False, False, False, False]


def read_caps():
    t0_count = 0
    t0 = touches[0]
    t0.direction = Direction.OUTPUT
    t0.value = True
    t0.direction = Direction.INPUT
    # funky idea but we can 'diy' the one non-hardware captouch device by hand
    # by reading the drooping voltage on a tri-state pin.
    t0_count = t0.value + t0.value + t0.value + t0.value + t0.value + \
        t0.value + t0.value + t0.value + t0.value + t0.value + \
        t0.value + t0.value + t0.value + t0.value + t0.value
    cap_touches[0] = t0_count > 2
    cap_touches[1] = touches[1].raw_value > 3000
    cap_touches[2] = touches[2].raw_value > 3000
    cap_touches[3] = touches[3].raw_value > 3000
    return cap_touches
```

Read cap touch inputs

PYRULER PROJECT

01

02

03

04

```
def type_alt_code(code):
    kbd.press(Keycode.CONTROL, Keycode.SHIFT)
    kbd.press(Keycode.U)
    kbd.release_all()
    kbd.send(Keycode.TWO)
    kbd.send(Keycode.ONE)
    kbd.send(Keycode.TWO)
    kbd.send(Keycode.SIX)
    kbd.send(Keycode.ENTER)
```

Send/type keyboard input for an ALT code

```
def toggle_mute():
    kbd.press(Keycode.CONTROL, Keycode.SHIFT)
    kbd.press(Keycode.M)
    kbd.release_all()
```

Toggle mute in Teams

```
def toggle_video():
    kbd.press(Keycode.CONTROL, Keycode.SHIFT)
    kbd.press(Keycode.O)
    kbd.release_all()
```

Toggle video camera in Teams

```
def go_to_search():
    kbd.press(Keycode.CONTROL)
    kbd.press(Keycode.E)
    kbd.release_all()
```

Go to search in Teams

PYRULER PROJECT

```python
while True:
    caps = read_caps()
    # light up the matching LED
    for i, c in enumerate(caps):
        leds[i].value = c
    if caps[0]:
        if ENABLE_KEYBOARD:
            go_to_search()
    if caps[1]:
        if ENABLE_KEYBOARD:
            toggle_video()
    if caps[2]:
        if ENABLE_KEYBOARD:
            toggle_mute()
    if caps[3]:
        if ENABLE_KEYBOARD:
            print("you approved that PR")
            layout.write('LGTM :+1:')
    time.sleep(0.1)
```

This is the main loop. It loops indefinitely and services the functions that are within it.

1. It reads the cap touch buttons.
2. Lights up the matching LED if the button is touched.
3. If cap 0 is touched, it will go to search in Teams.
4. If cap 1 is touched, it will toggle the video camera in Teams.
5. If cap 2 is touched, it will mute the video.
6. If cap 3 is touched, it will print “LGTM 👍” to wherever the keyboard input is placed if active.