Microbit introduction

Goal

- Learning Python - mainly for data processing.
- Understanding flow control is an important aspect of programming

python

Programming microcontrollers

- Using microcontrollers to practice flow programming.
- What are microcontrollers?
  - A microcontroller is a miniature computer.
  - Dedicated to a single (or very few) task(s).

Waiting for user input. If button A is pressed, increase clock count. If button B is pressed, start heating program X...

Programming microcontrollers

- Used to be C, C++.
- Require special programming devices.

- However, now you can use micropython
  - Version of Python for selected microcontrollers.

MicroPython

Original author(s)	Damien P. George
Initial release	3 May 2014; 6 years ago
Stable release	1.13 / 2 September 2020; 3 months ago
Repository	github.com/micropython/micropython
Written in	C
Platform	ARM Cortex-M, STM32, ESP8266, ESP32, 16bit PIC, Unix, Microsoft Windows, Zephyr, javaScript
Type	Python implementation
License	MIT license[1]
Website	micropython.org

Microcontrollers in the wild...

Which microcontrollers use Python?

Different microcontrollers use different flavours of micropython. However, swapping boards should not be overly difficult. Also, adafruit sells other boards (using the same flavor as the playground express)

BBC:microbit

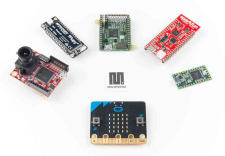
- The Micro Bit is an open source hardware ARM-based embedded system designed by the BBC for use in computer education in the United Kingdom.

Why do we use the microbit?

- There is a good online simulator for the microbit.
- That allows us to use it without buying the hardware.
- However, you could buy the microbit (or other device).

Beyond this course

- These devices are very versatile...



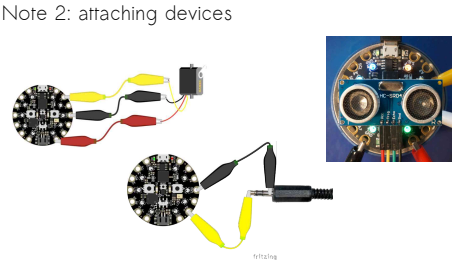
Beyond this course...

- The Playground Express is an interesting device in its own right.

Note 1: Stand-alone use

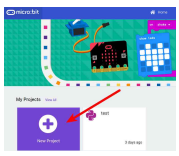
- The PGE needs a computer to be programmed.
- However, after that, whenever it is powered, it will run the onboard code.
- For example: you can power the device from a USB power bank

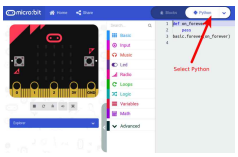
Note 2: attaching devices



Website

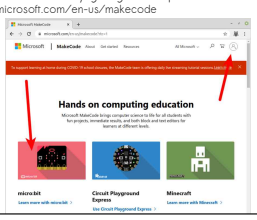
- https://makecode.microbit.org/



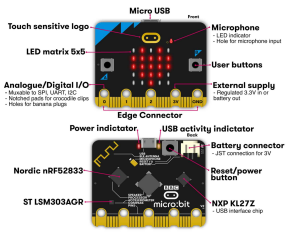


Making an account (optional)

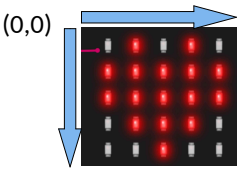
- You can make an account by going to the parent website:
  - https://www.microsoft.com/en-us/makecode



About the device



Led coordinates



Getting started

Initial code

- This is the initial Python code provided.
- This code is akin to a `while = True` loop
- Code in `def on_forever()` will be executed over and over.

```
def on_forever():
    pass

basic.forever(on_forever)
```

- You might replace this with:

```
while True:
    pass
```

Finding code

- The editor makes it easy to find code snippets.
- Find what you need in the left hand menu
- Drag to the code screen

Example 1

- Step 1: Blinking one of the LEDs (wait 100 ms between blinks)
  - Have one of the LEDs blink at a rate of ~10 Hz
- Step 2: Only blink whenever a button is pressed
  - Have one of the LEDs blink at a rate of ~10 Hz
  - but only when a button is pressed (held down)

Challenge 1: switching on/off LEDs

- Switch on all the LEDs.
- Switch on all the LEDs, wait 1 second, switch all of them off.

Challenge 2: Detect a single button press

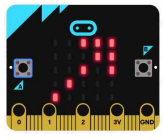
- How can we detect a single button press?

- Note there is another way of doing this:

```
def on_pin_pressed_p0():
    pass
input.on_pin_pressed(TouchPin.P0, on_pin_pressed_p0)
```

Challenge 3: Rotating arrow

- Start by drawing an arrow pointing up
- Every time you press a button, the arrow shifts to a next position



Challenge 4: make a counter

- Make a number counter, that goes up to 25.
- Extension: pressing button B resets the count.