MicroPython: Embedded programming for everyone

Jim Mussared
1 MicroPython

Python for microcontrollers
What is MicroPython?

- It’s Python 3... but micro!
  - Able to run on CPUs like the Cortex M0
  - 256kiB flash, 16kiB RAM, 16MHz
- Compilation happens on-device
- “Pythonic” access to low-level hardware (e.g. GPIO)
2  BBC micro:bit
BBC micro:bit

- LED matrix
- Accelerometer
- Magnetometer (compass)
- Temperature
- Buttons
- Bluetooth
- LiPo Battery header
- GPIO (+ SPI, I2C, UART, ADC)

Images: © BBC micro:bit https://www.microbit.co.uk/about
BBC micro:bit

- AUD $20-$25 each
- Compare to:
  - Arduino Esplora (~ $80)
  - Arduino Zero (~ $80)
  - Arduino Leonardo (~ $40)
  - Raspberry Pi Zero (~ $10 - $35)
  - Raspberry Pi (~ $60)
  - pyboard (~ $40)

* Prices are highly variable across countries and stores. Not including Arduino clones.
BBC micro:bit

- Microsoft TouchDevelop
- Microsoft Block Editor
- CodeKingdoms Javascript
- *and...* MicroPython!
micro:bit accessories

- Motorized buggy
- Motor drivers
- Starter kits
- Prototyping boards
- Edge connector

Image: © BBC micro:bit https://www.microbit.co.uk/about
micro:bit - more info

- [http://microbit-micropython.readthedocs.io/](http://microbit-micropython.readthedocs.io/)
- [https://www.microbit.co.uk/](https://www.microbit.co.uk/)
- [https://developer.mbed.org/platforms/Microbit/](https://developer.mbed.org/platforms/Microbit/)
- [https://www.kitronik.co.uk/bbc-micro-bit-accessories.html](https://www.kitronik.co.uk/bbc-micro-bit-accessories.html)
Demo
ESP8266

IoT for $5
ESP8266 features

- Tiny. Cheap.
- Once configured, very easy to use
  - Program over WiFi
  - Interactive REPL (serial + WiFi)
- Library support for external components
  - e.g NeoPixels (LED strips)
- WiFi! Both AP and Station.
  - IoT all the things!
ESP8266 gotchas

- Hardware variations (ESP-01, ESP-12a, etc)
  - Hardware documentation is hard to find
- Flash size
  - Beware of 4Mbit models. Need 8Mbit for MicroPython
- Programming
  - Some boards have USB, otherwise there are programmers
- Easy-to-use models (eg. ESP-01) have very few pins
- No binary distributions of MicroPython (yet)
- and...
ESP8266 gotchas

3.3V

!!!!!!!!
ESP8266 gotchas

- Beware of “3.3V” FTDI boards
- Get a programmer or an all-in-one board
- Adafruit’s “Feather Huzzah” is a great example
  - [https://www.adafruit.com/products/2821](https://www.adafruit.com/products/2821)
ESP8266

- Angus Gratton: Free as in cheap gadgets: the ESP8266
  - linux.conf.au 2016
  - https://www.youtube.com/watch?v=Du1MtqsySy4
- Damien George: Scripting the Internet of Things
  - Pycon Australia 2016
- Nick Moore: ESP8266 and MicroPython
  - PyCon Australia 2016
Demo
Building course content

Grokking out
Grok Learning

- Grok teaches coding using structured, interactive courses.
  - Year-round courses
  - Competitions (courses with leaderboards)
- Interactive notes and slides
- Comprehensive auto-marking
- Python 3, C++, Java, Bash, SQL, HTML/CSS, ...
- All school levels, university and professional.
Demo
Simulating the micro:bit

- Grok provides a sandbox to run your Python code
- Now does the same for micro:bit
  - Runs the real micro:bit firmware
  - Interactive notes
  - Marker can interact with your code running on simulated device
Hello, micro:bit!

Welcome to the "Introduction to micro:bit" course.

We're going to start with the "Hello, world" of embedded programming - turning on an LED. However, your micro:bit has 25 of them, so let's be a bit more adventurous and make them show a picture instead.

Don't worry if you don't understand the code right now - the important thing to start with is the button. This will turn on the device and run your code.

```python
from microbit import *
display.show(Image.HEART)
```

That's all there is to it! You just programmed an embedded device!
Hello, micro:bit!

programming - turning on an LED. However, your micro:bit has 25 of them, so let’s be a bit more adventurous and make them show a picture instead.

Don’t worry if you don’t understand the code right now - the important thing to start with is the ▶ button. This will turn on the device and run your code.

```python
from microbit import *
display.show(Image.HAPPY)
```

That’s all there is to it! You just programmed an embedded device!
This question hasn’t been released yet.

Another common situation is where we want to do one thing while a button is pressed, and another while the button is released.

For this problem, we’re going to write a program that shows a picture of a happy face while the button is down, and a sad face while the button is released.

Have a look at the micro:bit MicroPython Image documentation for the full list of images.

```python
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.GIRAFFE)
    else:
        display.show(Image.SAD)
    sleep(280)
```

Submissions:

- #5 Failed (1 test passed) a few seconds ago
- #4 Failed (1 test passed) 2 months ago
- #3 All tests passed!

Instructions:
from microbit import *

while True:
    if button_a.is_pressed():
        display.show(Image.BUTTERFLY)
    else:
        display.show(Image.GIRAFFE)
    sleep(10)
Demo
Grok Code Quest

- Starts October 31st 2016
- Newbies, Beginners, Intermediate *and* micro:bit
ANY QUESTIONS?

Web: http://groklearning.com/
Twitter: @groklearning
Email: jim@groklearning.com