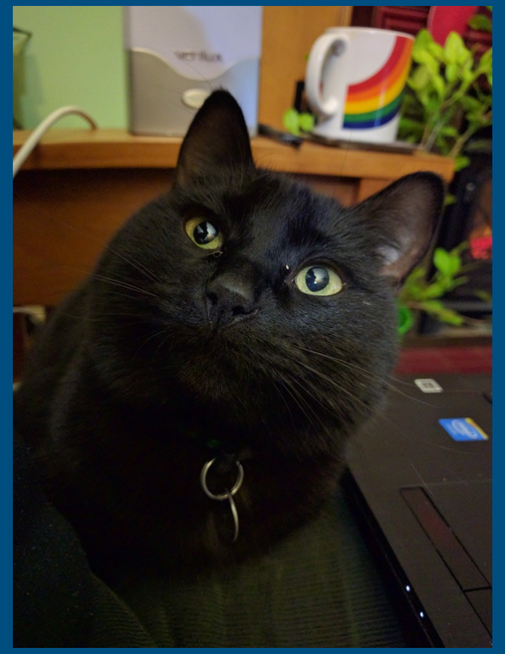
A Brief History of Micropython (also, cats)

Sev Leonard
@gizm0_0

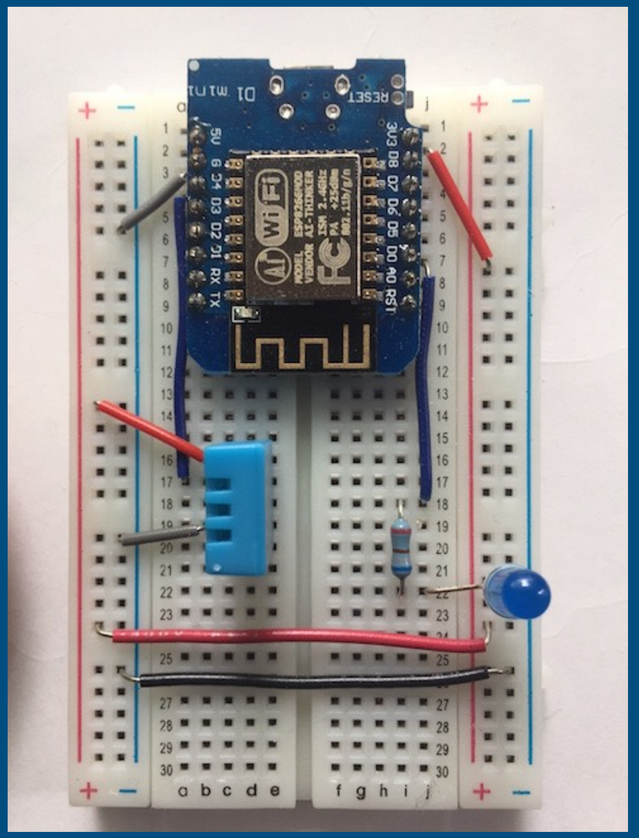
Hello!



- Pretzel afficionado
- He/Him
- Full stack technologist
- Oregon Health & Science University

Micropython Workshop!

- **Saturday April 28**
- **PDX Code Guild**
- **Hardware provided**
- **Laptop w/ USB required**



Tonight

- Micropython: Where did it come from?
- Micropython: How do I use it?
- Cats!
- Resources

Micropython - a history

“You don't need any software to program it: it behaves like a USB flash drive, and you just copy a Python text file to it and it runs”

- Damien George, theoretical physicist, Univ of Cambridge, 2013

http://www.wired.co.uk/article/micro-python

Kickstarter

- Rewrite of Python 3 for memory constrained environments
- Made for robots 
- REPL on board
- PyBoard

https://www.kickstarter.com/projects/214379695/micro-python-python-for-microcontrollers

Micropython vs Arduino

Arduino - an Ecosystem

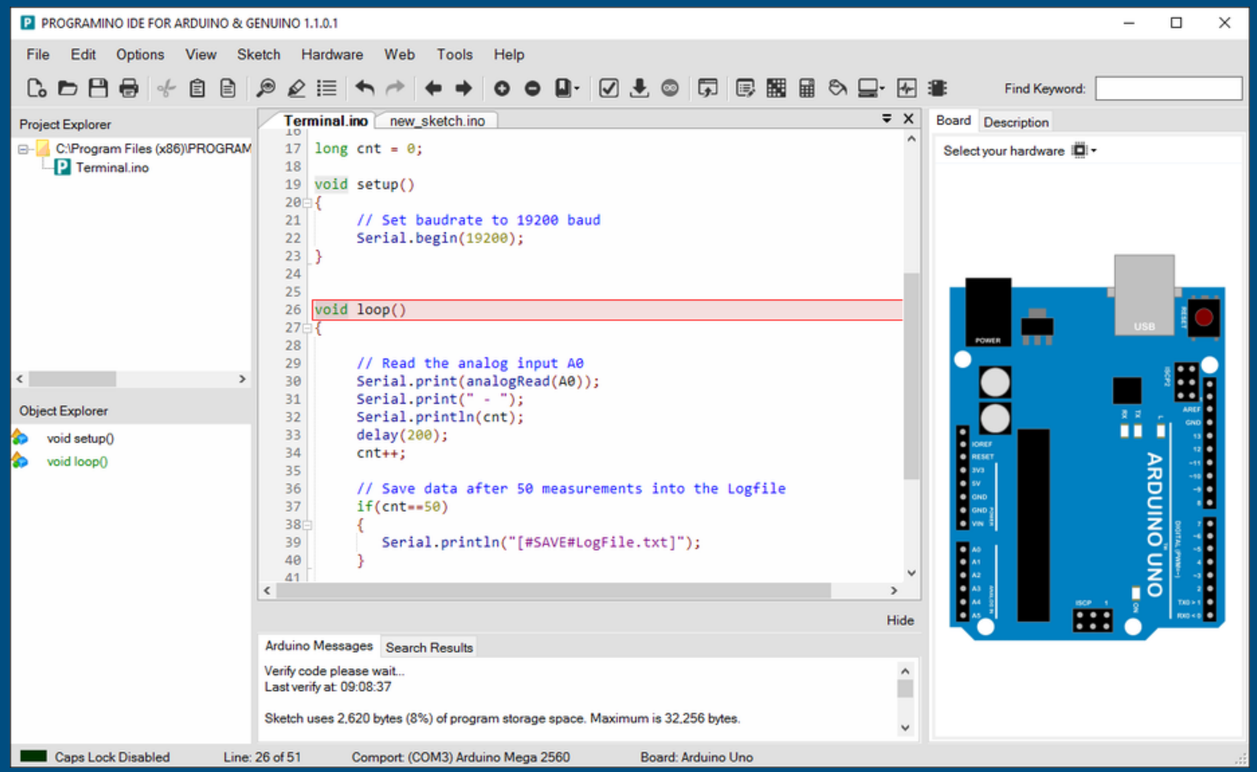


image source: http://programino.com/ide-for-arduino.html

https://www.arduino.cc/en/Guide/Introduction

Micropython - a Language

boot.py | *dht_functions.py*

```
1  import dht
2  import time
3  my_dht = dht.DHT11(machine.Pin(2))
4
5  def measure_humidity(outfile, poll_time_s):
6      while True:
7          my_dht.measure()
8          humidity = my_dht.humidity()
9          with open(outfile, 'a') as f:
10             f.write(str(humidity) + ",")
11         print("humidity: ", humidity)
12         time.sleep(poll_time_s)
```

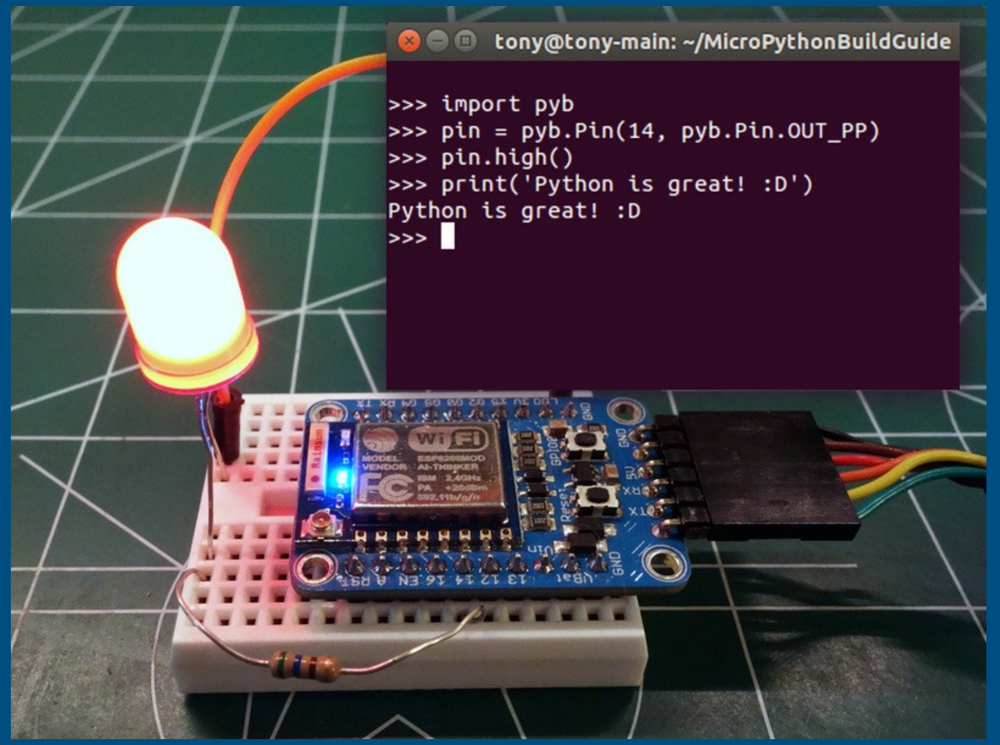


image source: https://learn.adafruit.com/building-and-running-micropython-on-the-esp8266/overview

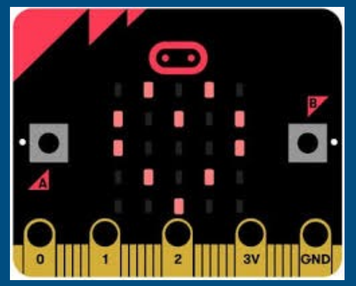
Arduino

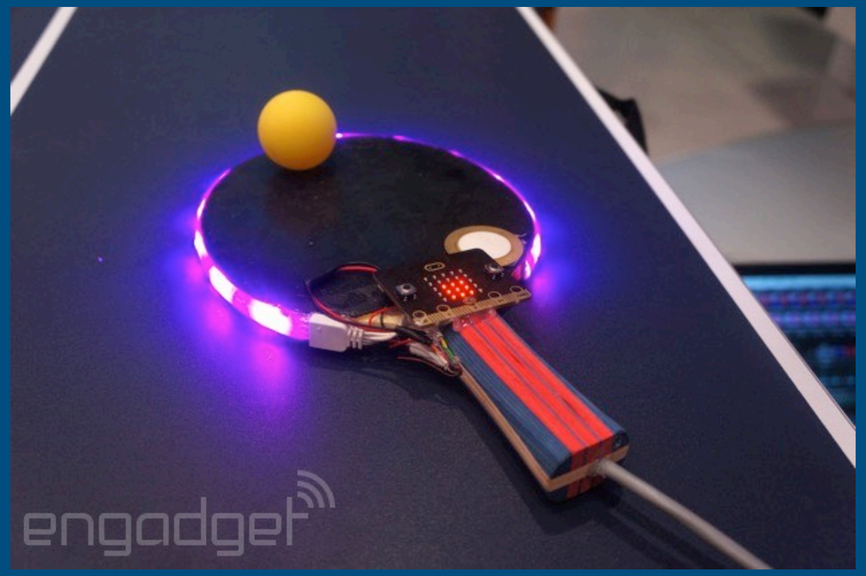
- Ecosystem
- C/C++ based
- Code compiled into machine language
- Company backed
- Large community
- Open source

Micropython

- Language
- Python!
- Code interpreted or precompiled
- Volunteer maintained
- Growing community :)
- Open source
- Can program on the device directly!

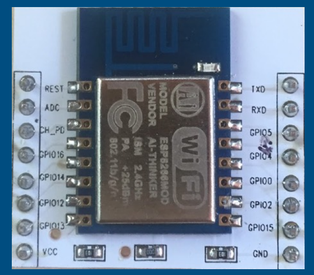
Micropython in the wild



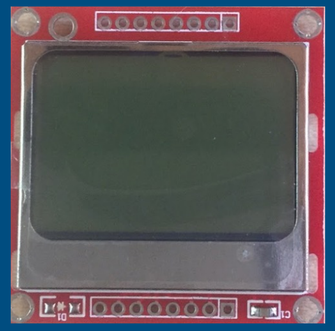


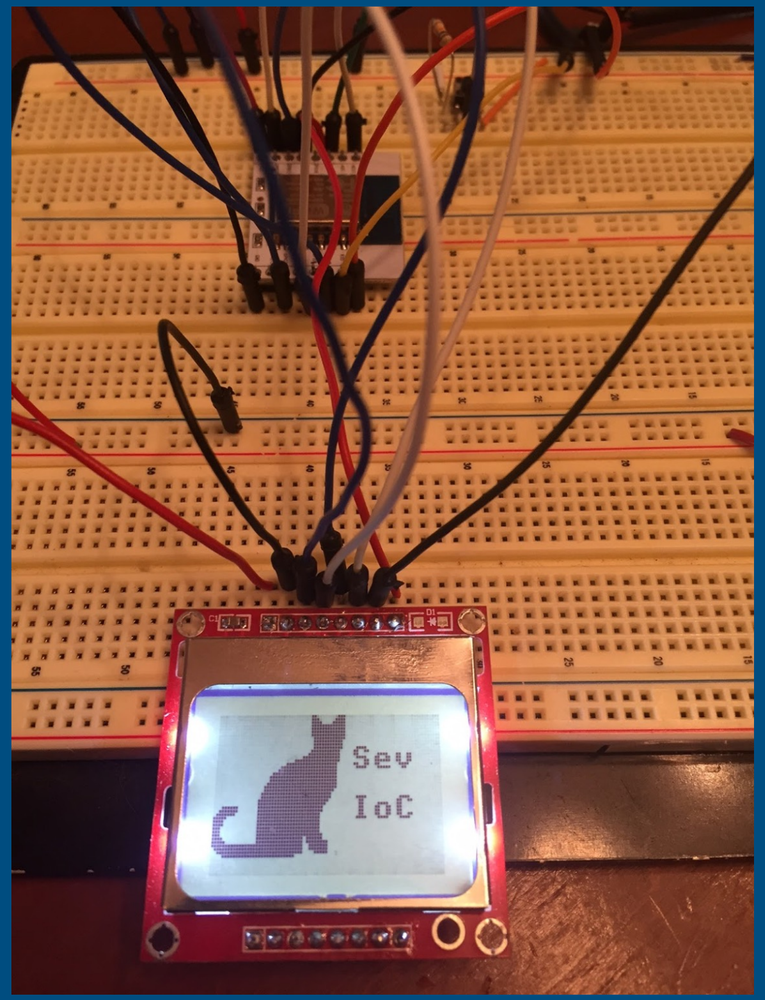
Or Internet of Cats! etc…

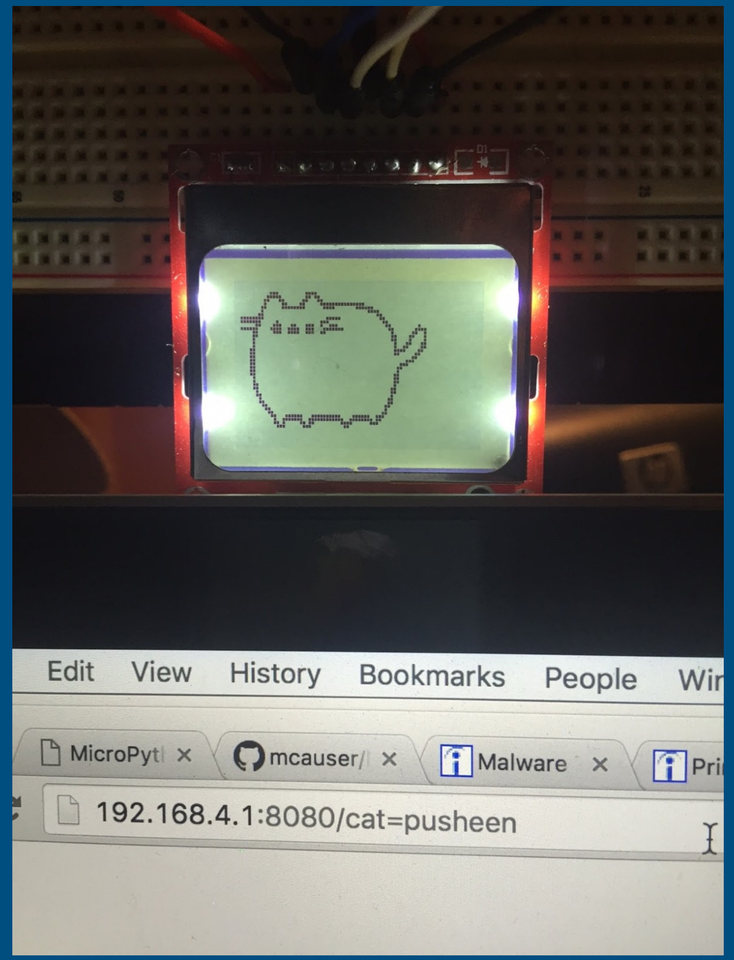
ESP8266-12



Nokia 5110

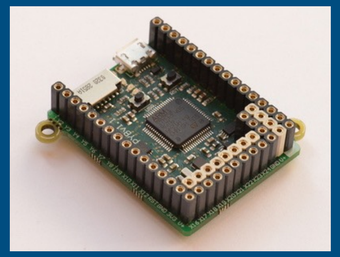






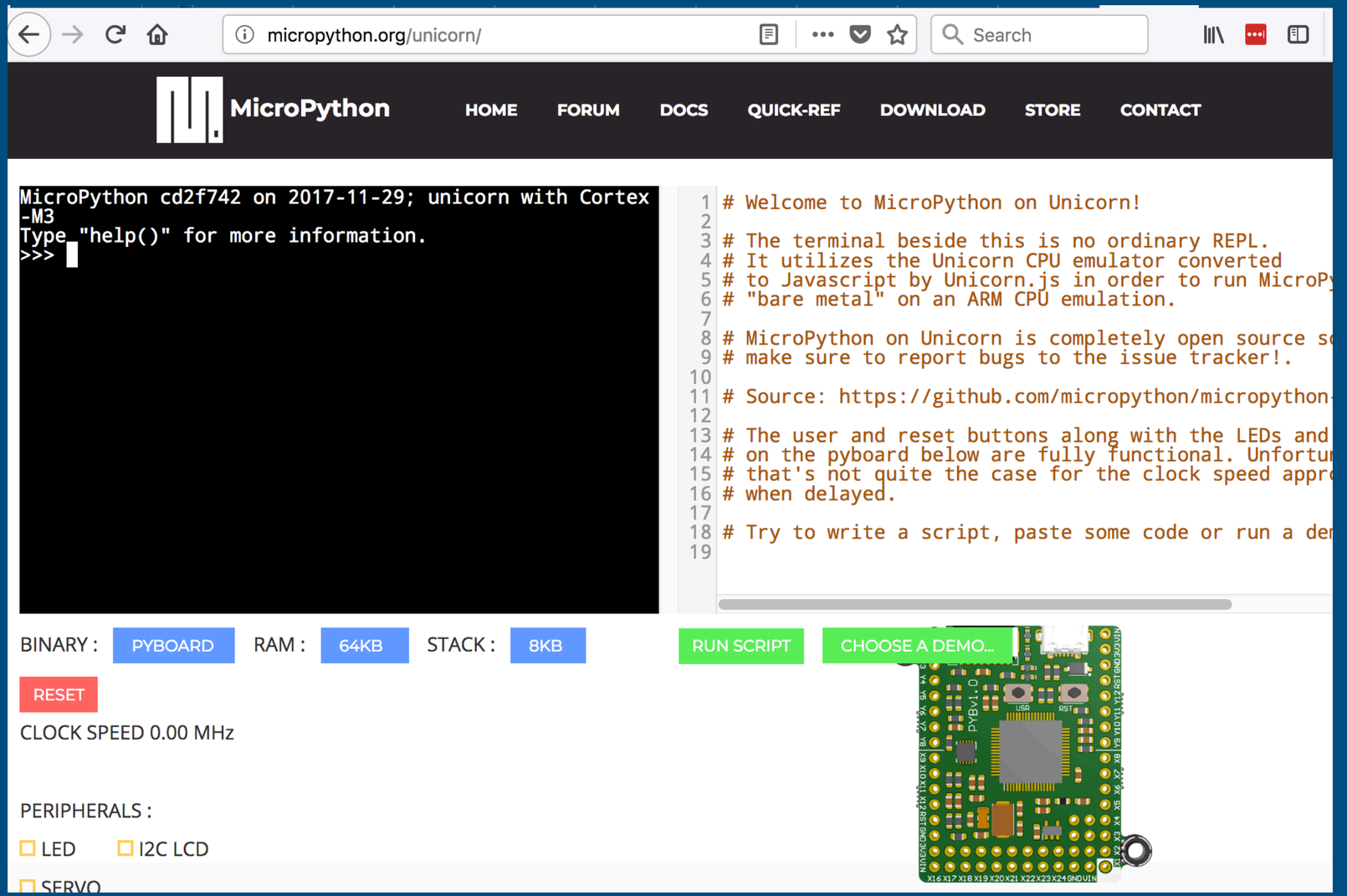
Using Micropython

PyBoard - $30 micropython.org

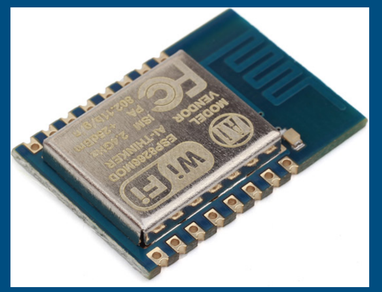


WiPy - $33 Adafruit
*can be ESP8266 based

ESP8266 - $2 - $15


micropython.org/unicorn/
Search
MicroPython
HOME
FORUM
DOCS
QUICK-REF
DOWNLOAD
STORE
CONTACT
MicroPython cd2f742 on 2017-11-29; unicorn with Cortex
-M3
Type "help()" for more information.
>>>
Welcome to MicroPython on Unicorn!
The terminal beside this is no ordinary REPL.
It utilizes the Unicorn CPU emulator converted
to Javascript by Unicorn.js in order to run MicroPy
"bare metal" on an ARM CPU emulation.
MicroPython on Unicorn is completely open source s
make sure to report bugs to the issue tracker!.
Source: https://github.com/micropython/micropython
The user and reset buttons along with the LEDs and
on the pyboard below are fully functional. Unfortu
that's not quite the case for the clock speed appr
when delayed.
Try to write a script, paste some code or run a de
BINARY :
PYBOARD
RAM :
64KB
STACK :
8KB
RUN SCRIPT
CHOOSE A DEMO...
RESET
CLOCK SPEED 0.00 MHz
PERIPHERALS :
LED
I2C LCD
SERVO

Get firmware



micropython.org/download

github.com/themadinventor/esptool/

MicroPython downloads

For the MicroPython source code, please visit github.com/micropython/micropython.

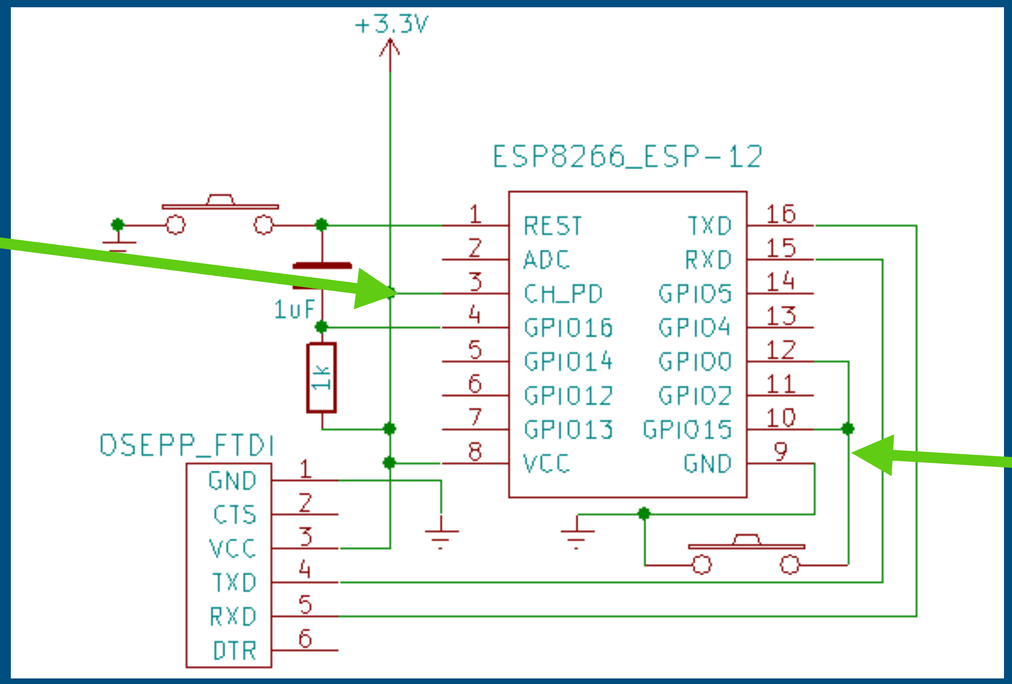
Daily dumps of the GitHub repository are available from this server:

- micropython-master.zip
- pyboard-master.zip

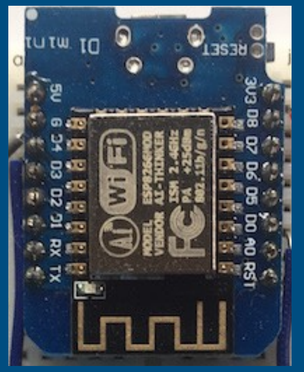
Links to firmware below: pyboard | WiPy | ESP8266 | other

pip install esptool

Load firmware



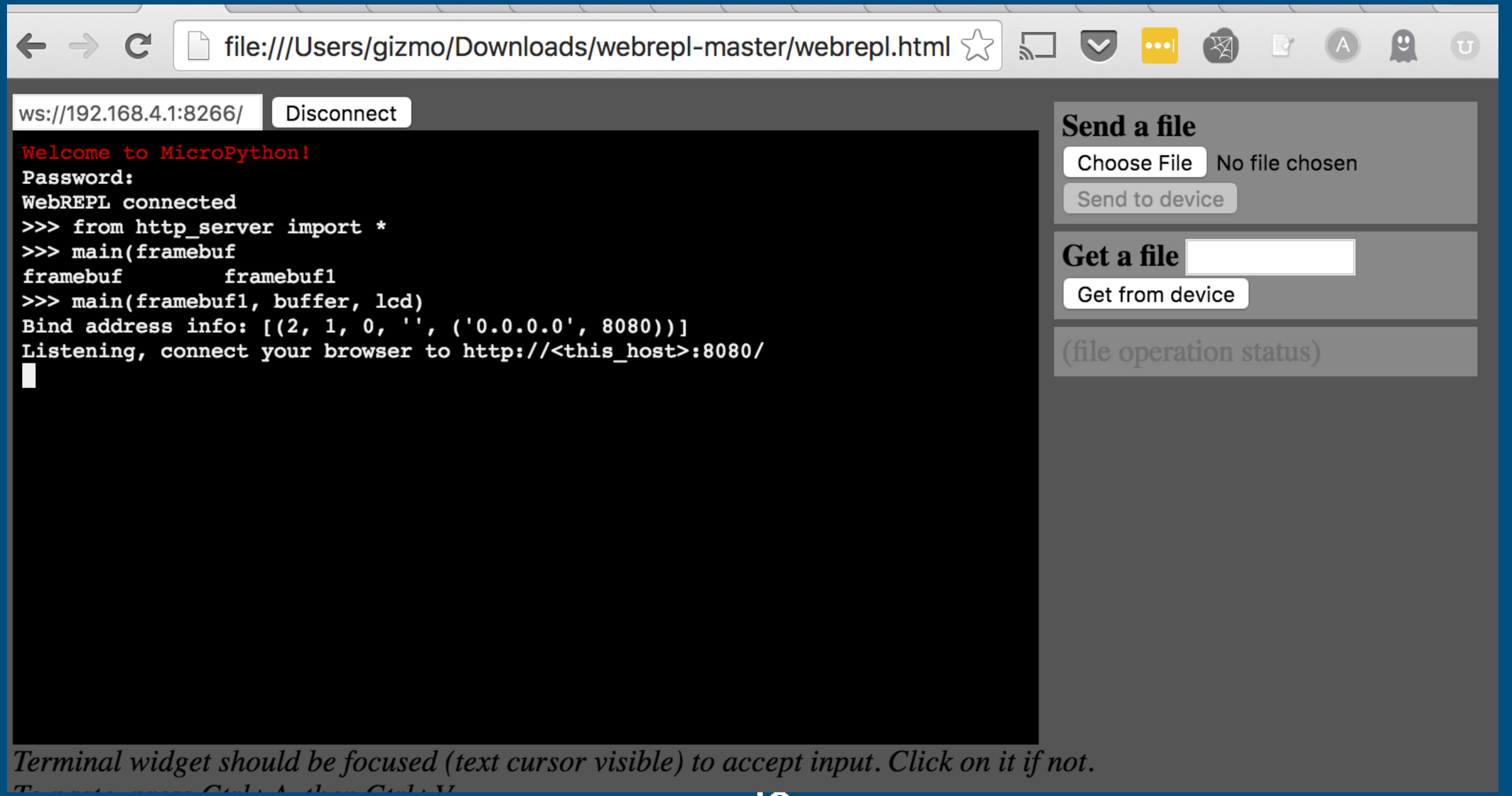
OR



Python!

> screen /dev/ttyUSB0 115200

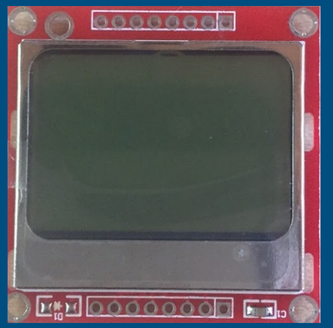
```
MicroPython v1.8.3-24-g095e43a on 2016-08-16; ESP module with ESP8266
Type "help()" for more information.
>>> print('Hello world!')
Hello world!
>>>
```


file:///Users/gizmo/Downloads/webrepl-master/webrepl.html
ws://192.168.4.1:8266/
Disconnect
Welcome to MicroPython!
Password:
WebREPL connected
>>> from http_server import *
>>> main(framebuf
framebuf framebuf1
>>> main(framebuf1, buffer, lcd)
Bind address info: [(2, 1, 0, '', ('0.0.0.0', 8080))]
Listening, connect your browser to http://<this_host>:8080/
Send a file
Choose File
No file chosen
Send to device
Get a file
Get from device
(file operation status)
Terminal widget should be focused (text cursor visible) to accept input. Click on it if not.

Connect the LCD

```
spi = HSPI(baudrate=80000000, polarity=0, phase=0)
RST = Pin(4)
CE = Pin(5)
DC = Pin(12)
BL = Pin(16)
lcd = upcd8544.PCD8544(spi, RST, CE, DC, BL)

width = 84
height = 48
pages = height // 8
buffer = bytearray(pages * width)
framebuf1 = framebuf.FrameBuffer1(buffer, width, height)
```



https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

Draw some monochrome cats

First, find a cat (pic)

- Converts easily to B/W
- Aspect ratio close to 2:1
- Not super detailed

Process the image

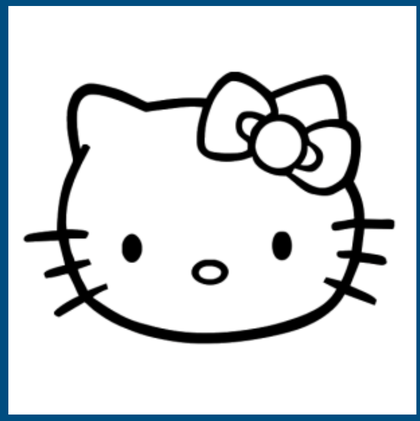
Get our cat onto a white background

```
convert hello_kitty.png  -background white -
alpha remove hello_kitty_white.png
```

Scale our cat

Original size: 503/503 ~ → 48/48

```
convert hello_kitty_white.png -resize 48x48
small_cat_paws.png
```

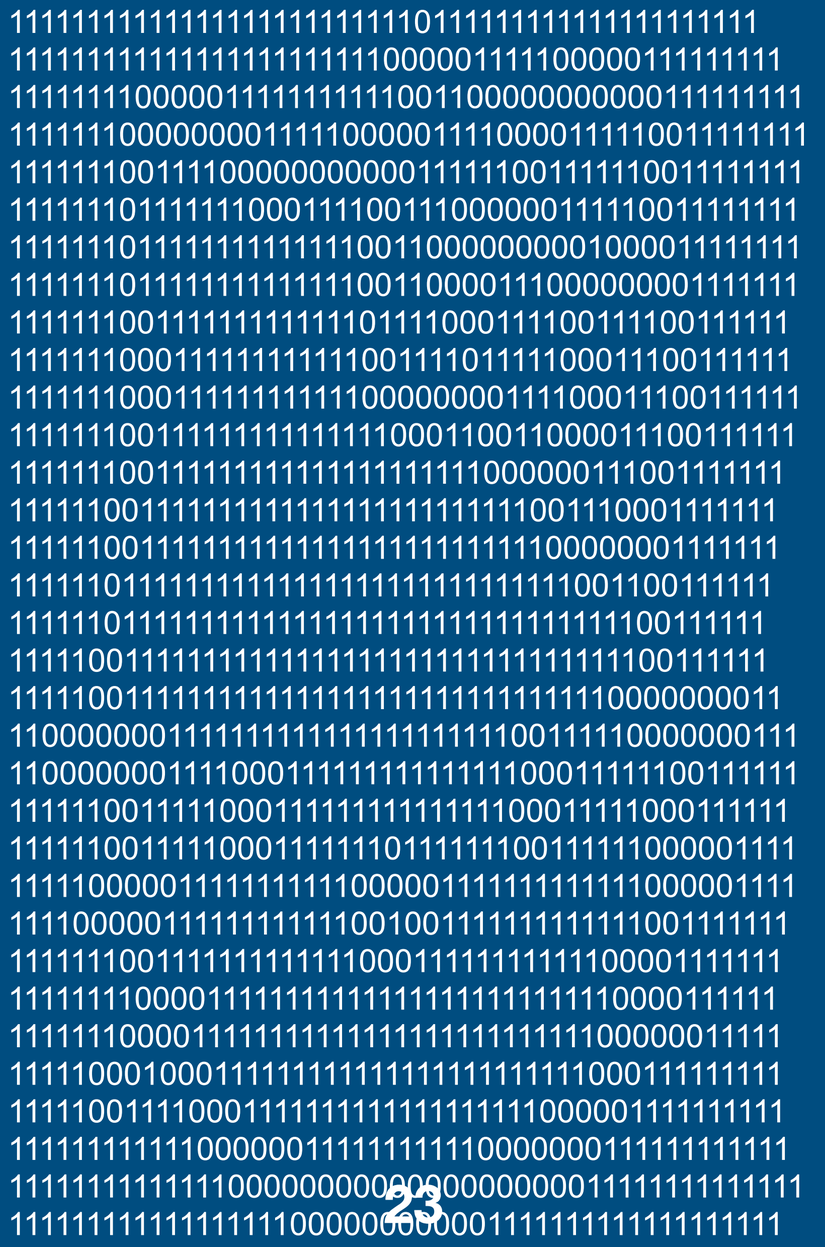


Convert cat to bitmap

Use convert_png.py based on github.com/garybake/upython_wemos_shields

```
python convert_png.py > hello_kitty.txt
```

Woo more Python!


23

Framebuf reference

```
def fill(self, col):
    self.framebuf.fill(col)

def pixel(self, x, y, col):
    self.framebuf.pixel(x, y, col)

def scroll(self, dx, dy):
    self.framebuf.scroll(dx, dy)

def text(self, string, x, y, col=1):
    self.framebuf.text(string, x, y, col)
```

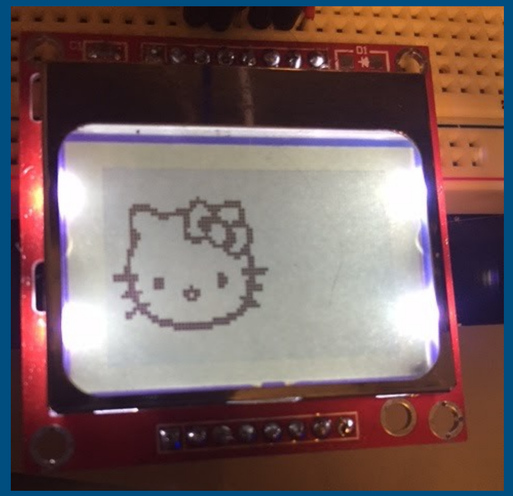
github.com/micropython/micropython/blob/master/drivers/display/ssd1306.py

Rendering the image

```
1 for line in f:
2     for char in line:
3         if char == '0':
4             framebuf1.pixel(y,x,1)
5             gc.collect()
6 lcd.data(buffer)
7 gc.collect()
```

Draw Hello Kitty!

```
from setup_lcd import *                          # import lcd
from draw_image_from_text import *               # draw_image
draw_image('hello_kitty.txt', framebuf1, buffer, lcd)
```

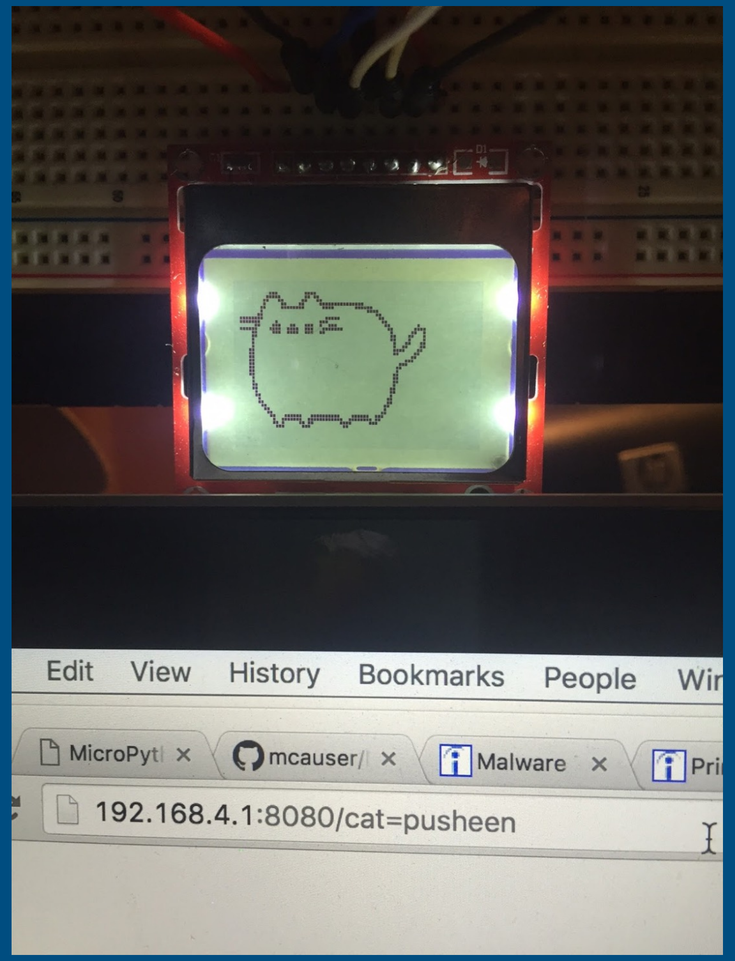


And now, the INTERNET!

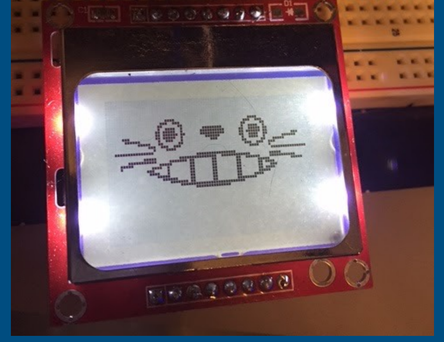
github.com/micropython/micropython/blob/master/examples/network/http_server_simplistic_commented.py

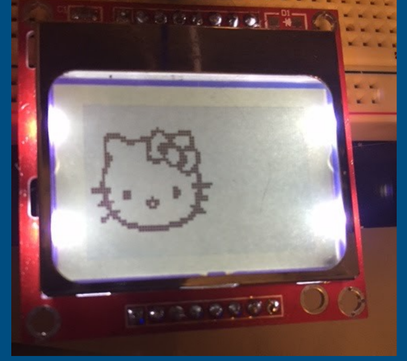
```
while true:
…
  req = client_stream.readline()
  req = str(req)
  print(req)
  if req.find('GET /cat=sitting') > 0:
     lcd_cat = 'sitting_cat.txt'
  draw_image(lcd_cat, framebuf1, buffer, lcd)
```

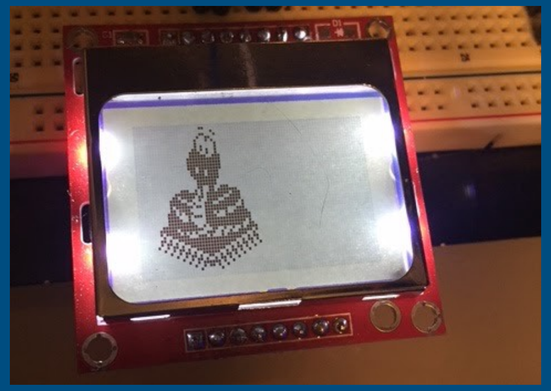
Request a cat! / pusheen

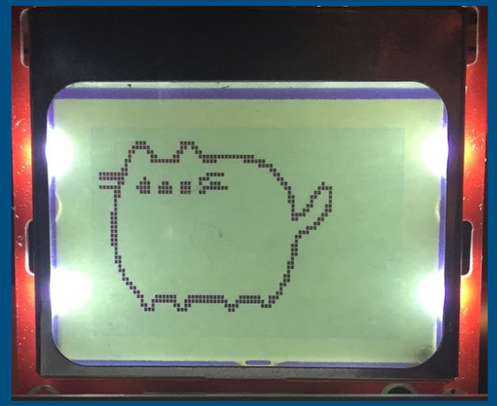


Collect them all!









Resources

- https://learn.adafruit.com/micropython-basics-what-is-micropython/overview
- https://micropython.org/unicorn/
- https://forum.micropython.org/
- http://microbit.org/

Github repos

- This talk: https://github.com/gizm00/pydx_upython
- PyCon Lab: https://github.com/gizm00/esp8266_micropython_lab
- PyCascades IoT Talk (no code) https://github.com/gizm00/pycascades_2018
- https://github.com/mcauser/MicroPython-ESP8266-DHT-Nokia-5110

Electronics buying

- WeMos D1 mini & DHT11: https://www.ebay.com/usr/alice1101983
- Breadboards: http://www.ebay.com/usr/chivazhu
- Resistors: http://www.ebay.com/usr/henpitts
- Nokia 5110 - got from a friend, typically available for $10

Thanks!

- @gizm0_0
- github.com/gizm00
- sev@thedatascout.com