

MicroPython with LoPy

Getting started workshop @ MakeZurich

2.2.2017

Christian Fässler

What is MicroPython

- Python for Microcontrollers
- Lean implementation of Python 3
- Subset of standard lib
- Needs
 - 256k code space
 - 16k RAM

What is a LoPy

- ESP32 Board
 - 160 MHz CPU
 - 512k RAM
 - 4 MB Flash for user code
 - I2C, SPI, UART, I2C
 - SD Card
 - 8 ADC Channels
 - 24 GPIOs
 - RTC
 - WS2812 LED
- Wireless connectivity
 - Bluetooth 4.2 (BLE & Classic)
 - LoRa (Semtech's SX1272)
 - WiFi

Tooling

- Batteries included!
 - Firmware upgrade tool
 - PyMakr IDE
 - PyMate mobile App
 - Based on MQTT
 - Soon: cloud service
- Code upload via
 - Built-in FTP/telnet Server
 - Serial connection

First thing to do

- Install Pycom updater
 - <https://www.pycom.io/support/supportdownloads>
- Firmware Upgrade!
 - Almost weekly releases with new features
 - Check <http://forum.pycom.io> Announcement topic

Connect to Device REPL

- PyMaker IDE
 - Settings/Preferences/Pycom Device
- Directly via serial port
 - 8N1 / 115200 Baud
 - screen /dev/tty... 115200
 - putty
- WiFi
 - SSID: lopy-wlan-XXXX
 - Password: www.pycom.io
 - Telnet 192.168.4.1

Hello World!

- REPL (read-evaluate-print-loop)
- Interactive interpreter (>>>)
- `pycom.rgbled(0xRRGGBB)`

```
>>> print('Hello World')
```

```
>>> import pycom  
>>> pycom.heartbeat(False)  
>>> pycom.rgbled(0xFF0000)
```

Primer on Python

- Variables

```
text = 'Dont forget your towel!'
number = 42
floatingnumber = 13.37
```

- Functions

```
def print_text(name, age):
    text = "My name is {0} and I'm {1} years old"
    print(text.format(name, age))
```


Primer on Python (math)

- Addition, Subtraction, Division, Multiplication, exponentiation

```
12+34
```

```
3*3
```

```
16/2
```

```
(3+1)*(4+5)
```

```
5**2 #5 to the power of 2
```

```
'spam'*3 #spamspamspam
```

Primer on Python (useful built-ins)

- `print(text)` – Print text
- `int(text)` – «cast» text to int
- `type(obj)` – Get type of object
- `input(prompt)` – Read from stdin return value as string

Primer on Python (containers)

- Lists

- `mylist = [1,2,3,4]`
- `mylist.append(5)`
- `mylist.remove(2)`

- Dictionaries

- `mydict = {'name': 'zaphod beebblebrox', 'heads':1}`
- `mydict['heads'] = 2`

Primer on Python (control structs)

```
if value == True:  
    print('value is true')  
else:  
    print('value is false')
```

```
while True:  
    do_whatever_is_to_do()
```

```
for i in range(10):  
    print('{0}-th run'.format(i))
```

Time Module

```
import time  
time.sleep(10) #sleep 10 seconds  
time.sleep_ms(200) #sleep 200 milliseconds
```

Exercises

- Create a function which changes the LED color from red to blue every 1s
- Create a function which asks for your weight and your body size and calculates $BMI = \text{weight (kg)} \div \text{size}^2 \text{ (m)}$

Primer on Python (modules)

- Files are treated as modules
- Stuff is imported from modules
- Stuff needs to be imported to be used
- File lib.py contains function connect()

```
# lib.py
def print_name(name):
    print('-----')
    print(name)
    print('-----')
```

```
# use lib
from lib import print_name
print_name()
```

More on Python

- Lots of tutorials
- <https://docs.pycom.io>
- <https://learnpythonthehardway.org/>
- <http://docs.python-guide.org/en/latest/>

GPIO output

```
from machine import Pin
# initialize ``P9`` in gpio mode and make it an output

p_out = Pin('P9', mode=Pin.OUT)
p_out.value(1)
p_out.value(0)
```

GPIO input

```
from machine import Pin
# initialize ``P10`` in gpio mode and make it an input

p_in = Pin('P10', mode=Pin.IN, pull=Pin.PULL_UP)
p_in.value() # get value, 0 or 1
```

GPIO with interrupts

```
from machine import Pin
# initialize ``P10`` in gpio mode and make it an input

p_in = Pin('P10', mode=Pin.IN, pull=Pin.PULL_UP)
p_in.callback(Pin.IRQ_HIGH_LEVEL, lambda pin: print("ON"))
p_in.callback(Pin.IRQ_LOW_LEVEL, lambda pin: print("OFF"))

# Pin.IRQ_HIGH_LEVEL | Pin.IRQ_LOW_LEVEL also possible
```

GPIO numbering

```
# Numbering with expansion board  
Pin.exp_board.G16  
led = Pin(Pin.exp_board.G16, mode=Pin.OUT)
```

```
# Numbering without expansion board  
# P10 is mapped to the Button  
Pin.module.P9  
led = Pin(Pin.module.P9, mode=Pin.OUT)
```

Pin Diagramms

- Expansion board
https://www.pycom.io/wp-content/uploads/2016/11/expansion_v02_pinout.pdf
- LoPy
https://www.pycom.io/wp-content/uploads/2016/11/lopy_pinout.pdf

Exercise

- React to button presses on P10 and turn on/off on P9
- Do the same with interrupts!

Boot process

- /flash/boot.py
- /flash/main.py

Nice to know

- Deploy your scripts via
 - telnet
 - FTP
- Safe boot mode
 - Firmware recovery

Hello ThingsNetwork!

- https://docs.pycom.io/pycom_esp32/pycom_esp32/tutorial/includes/lora-abp.html

```
from network import LoRa
import socket

# Initialize LoRa in LORAWAN mode.
lora = LoRa(mode=LoRa.LORAWAN)

# join a network using ABP (Activation By Personalization)
lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))

# create a LoRa socket
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)

# make the socket blocking
# (waits for the data to be sent and for the 2 receive windows to expire)
s.setblocking(True)

# send some data
s.send(bytes([0x01, 0x02, 0x03]))

# make the socket non-blocking
# (because if there's no data received it will block forever...)
s.setblocking(False)

# get any data received (if any...)
data = s.recv(64)
print(data)
```

Get DeviceEUI

```
import binascii
from network import LoRa

lora = LoRa(mode=LoRa.LORAWAN)
devEUI = binascii.hexlify(lora.mac())
```

Libraries

- <https://github.com/micropython/micropython-lib>
- Useful things
 - urllib
 - urequests
 - MQTT client

What is there?

- https://docs.pycom.io/pycom_esp32/library/index.html
- I2C
- ADC / DAC
- OneWire
- PWM
- SPI
- Watchdog (WDT)
- SD Card
- Timers
- RTC
- Bluetooth GATT
- Multithreading

Thanks!

- Hope I could whet your appetite ☺
- If you have any questions please contact me

Our online shop with pycom products

- <http://shop.adnexo.ch>

Code & slides

- https://github.com/adnexo-GmbH/lopy_ws

Contact

- christian.faessler@adnexo.ch
- www.adnexo.ch
- http://twitter.com/adnexo_gmbh