pyladies Berlin IoT and MicroPython workshop

Thank you!

Amazon Development Center Berlin

ThoughtWorks®

George Robotics Ltd
Developers of MicroPython

Agenda I

(1) IoT – The Internet of Things

(2) Challenges to program a Microcontroller

(3) Different languages for the IoT: Lua, Ruby, JerryScript

(4) MicoPython

Agenda II

(5) Workshop

(6) The pyboard layout

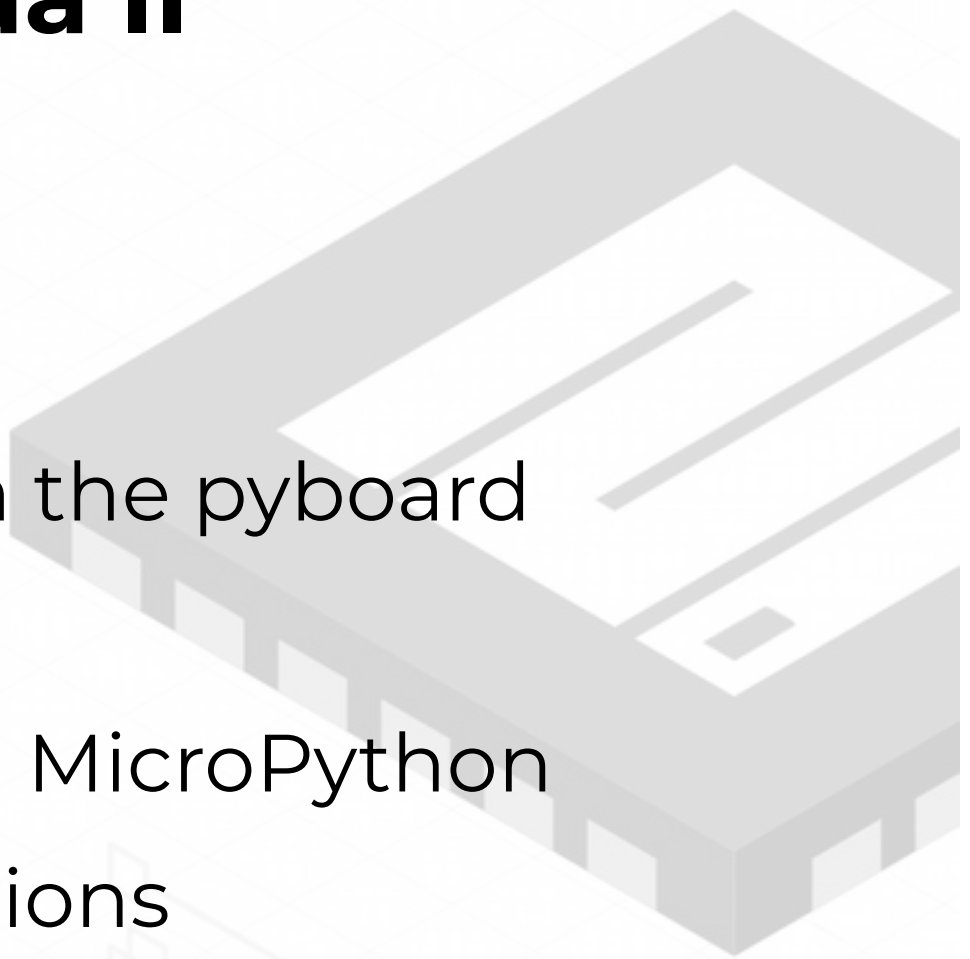
(7) How to interact with the pyboard

(8) Demos!

(9) Hands-on: exploring MicroPython

(10) Real-world applications

(11) What’s next?



The Internet of Things

Network of physical devices, vehicles, home appliances and other embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data.

2015 to 2016: increased 30% to 8.4 billion devices

In 2020 there will be 30 billion devices

These devices collect useful data with the help of existing technologies, autonomously and flow the data between other devices

IoT

The Internet of **making Things work**

The Internet of **useful Things**

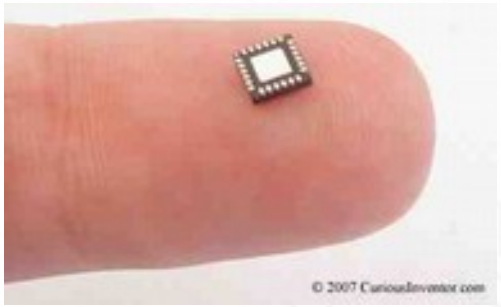
IoT = Microcontroller + (wireless) communication

Complexity of Hardware

Data Sheets! A few thousand pages for a single Microcontroller

High-level scripting languages allow:

- Easier to read/write and understand code
- Abstraction of HW
- Rapid prototyping
- More portable code
- Library reuse



Difference Embedded SW Engineer & SW Developer

- Knowing the hardware
- **PC vs. PCB**
- How many lines of code?
- Different debugging
- Controlling and managing the hardware

Lua and eLua

Pros: simple language, light-weight, fast

Cons: no native bitwise ops, no integers

Uses in IoT: NodeMCU ESP8266 board

Games, web applications and image processing

Ruby

```ruby
# Die Begrüßungsklasse
class Greeter
  def initialize(name)
    @name = name.capitalize
  end

  def salute
    puts "Hallo #{@name}!"
  end
end

# Erstelle ein neues Objekt
g = Greeter.new("Welt")

# Ausgabe "Hallo Welt!"
g.salute
```

Pros: popular language, lots of feature and libraries

Cons: no proper support for Microcontrollers – yet

JerryScript JavaScript for Microcontrollers

JavaScript engine for the Internet of Things

Pros: very popular language, large community

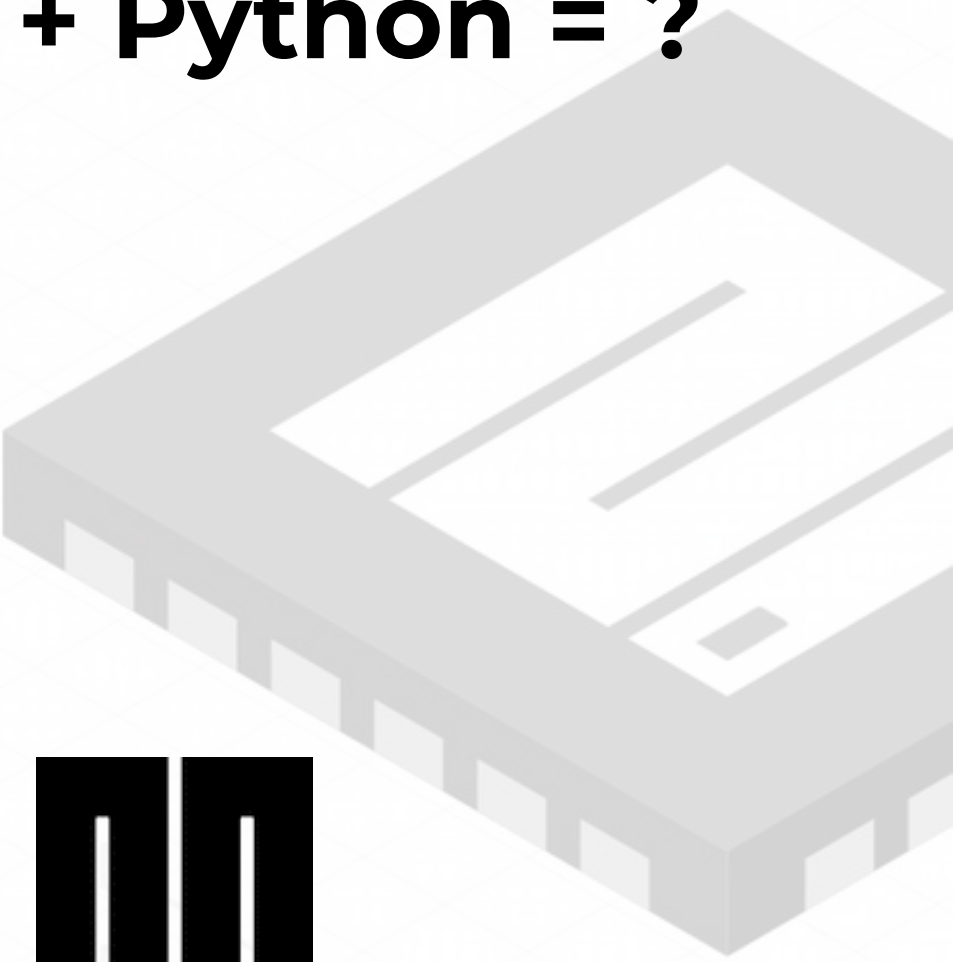
```
setInterval(function() {
        digitalWire(LED1, Math.random()>0.5);
}, 20)
```

Cons: callback-based, all numbers are floats

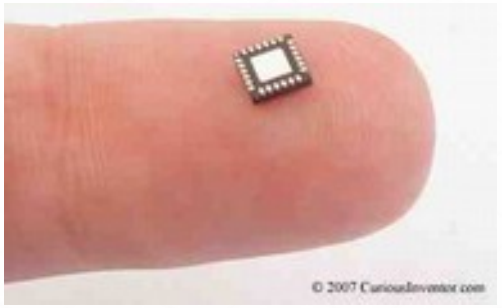
Uses in IoT: Espruino boards, ESP8266, Tessel boards

Jerryscript.net **www.tessel.io** **www.espruino.org**

Microcontroller + Python = ?



+

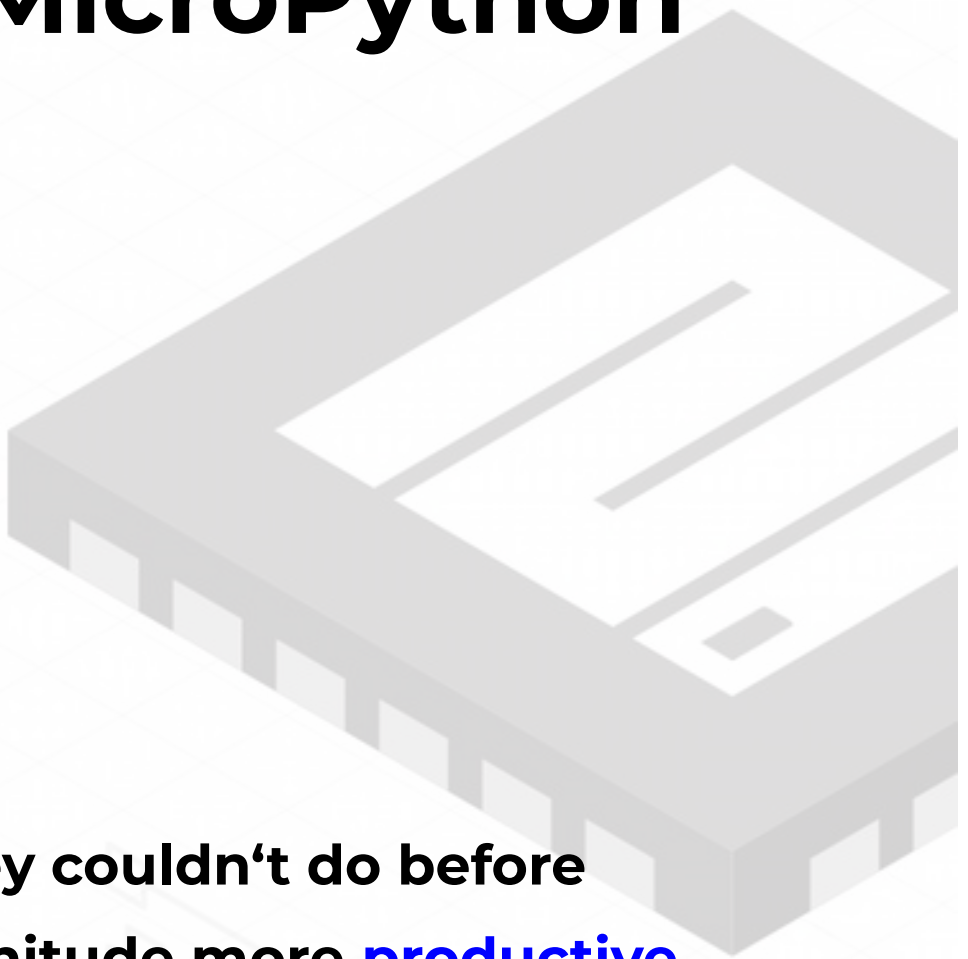


=

Motivation for MicroPython

Electronics circuits now pack an enormous amount of functionality in a tiny package

Need a way to control all these sophisticated devices.



MicroPython

- **allows beginners to do things they couldn‘t do before**
- **Professionals be an order of magnitude more productive**
- **Building devices easier and more accessible**

What is MicroPython?

A powerful and modern language
large community made to run on
constrained/embedded systems

MicroPython is
- complete reimplementation of a subset of Python 3.x
- designed to be efficient with resources
- designed to run bare metal

MicroPython includes
- a compiler, runtime and familiar REPL
- Support for basic libraries (modules), most with an ‘u’
- Extra modules to control hardware

The man behind MicroPython

- **Damien P. George** was born in Melbourne, Australia
- Bachelor in Eng & Science & PhD in theoretical Physics
- Writing embedded software for many Microcontrollers
- 6 years as a theoretical physicist incl. Cosmology and Higgs boson
- Now working full time to improve the MicroPython ecosystem

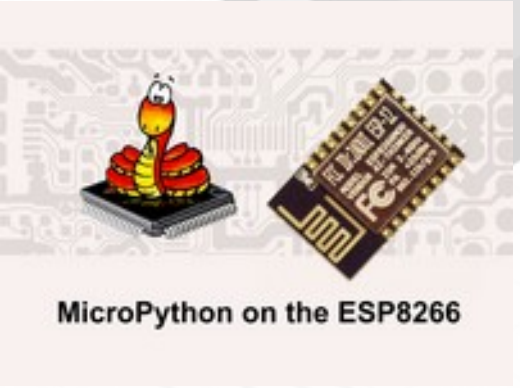
2013 Crowdfunding via Kickstarter

2015 MicroPython went to Space

ESA is funding to make
the language more robust
for critical embedded systems

2016 MicroPython went to school

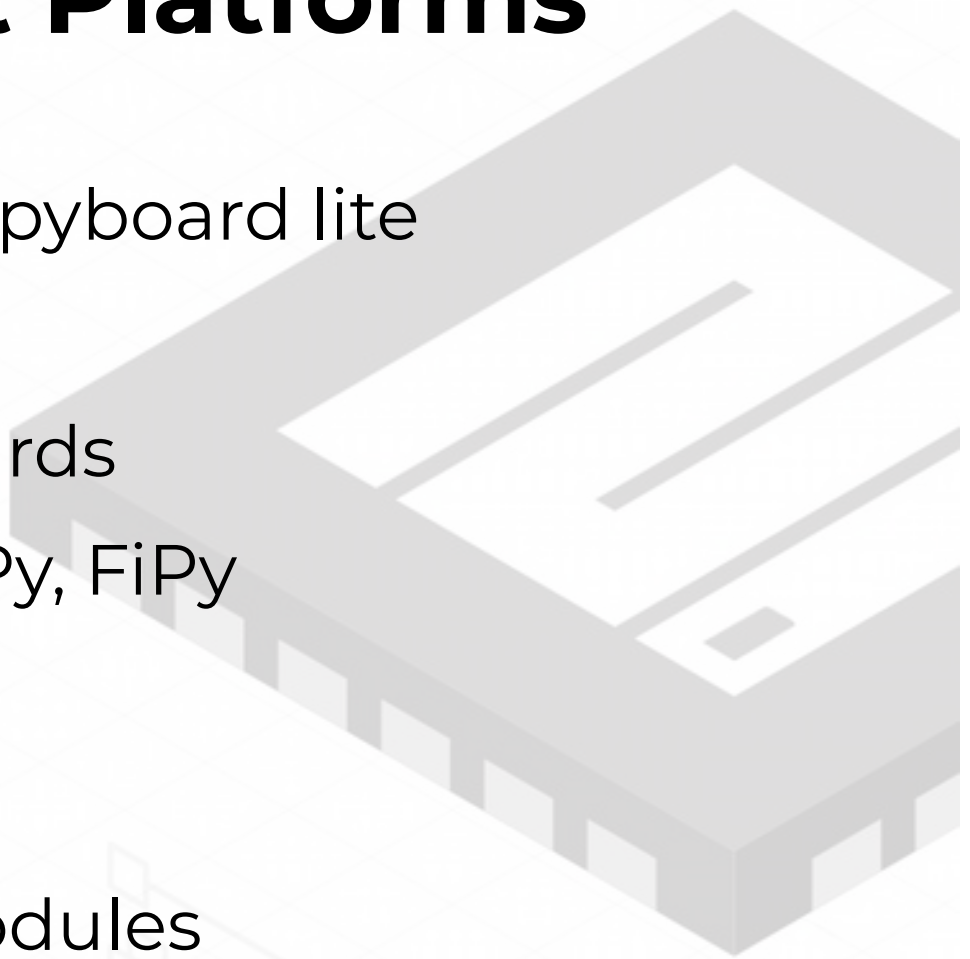
- BBC Micro:Bit project
  brought 1 Mio units to
  7 year old children in the UK
- 2nd Kickstarter ESP8266 support
- New Logo



Facts & Figures

- MicroPython is a public project on GitHub with 6000+ stars
- ranking in the top of 100 most popular C/C++ projects
- MIT license
- Contributions come from many people (190) with many different systems leads to: more robust code and build system
- more Features and supported hardware
- active forum

Development Platforms

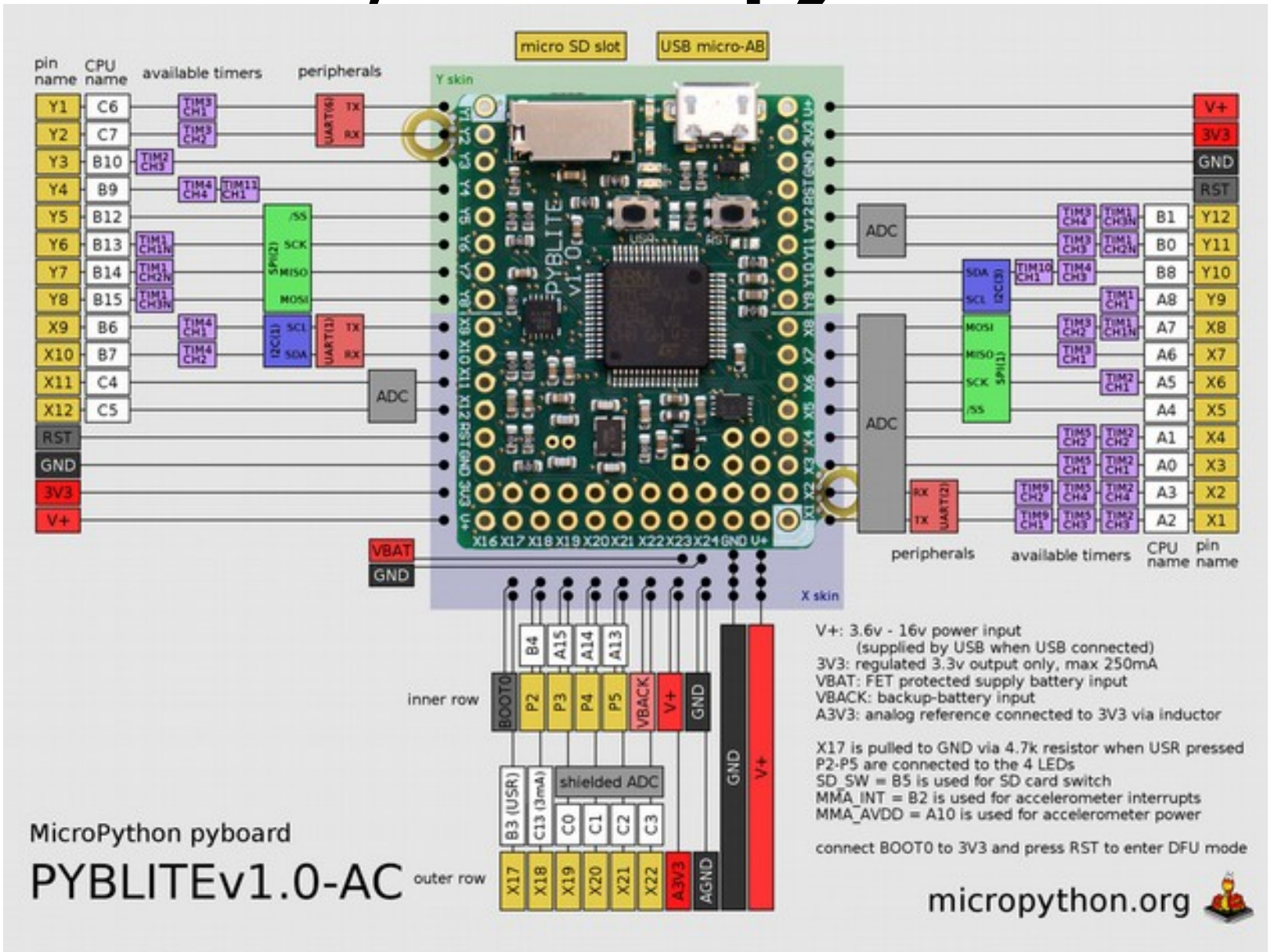


- MicroPython pyboard and pyboard lite
- MicroPython skins
- Adafruit CircuitPython boards
- PyCom Modules: LoPy, WiPy, FiPy
- ESP32 and ESP8266
- ST WiFi Module
- Digi International Xbee modules
- OpenMV Camera for Machine Vision

MicroPython Beginner Set

- Pyboard lite with accelerometer and headers
- Aluminium case for the pyboard in colour of your choice
- Micro-USB cable for connecting to the PC
- LCD160CRv1.0 Display with resistive touch
- 2 Servo Motors with different wheels
- 1 Protoskin for your own ideas
- LED +Resistor for fading LED example
- Jumper Wire and header pins
- HDC1080 Temperature Sensor breakout board
- STICKERS

Hello, I’m the pyboard



Pyboard lite is Low Power

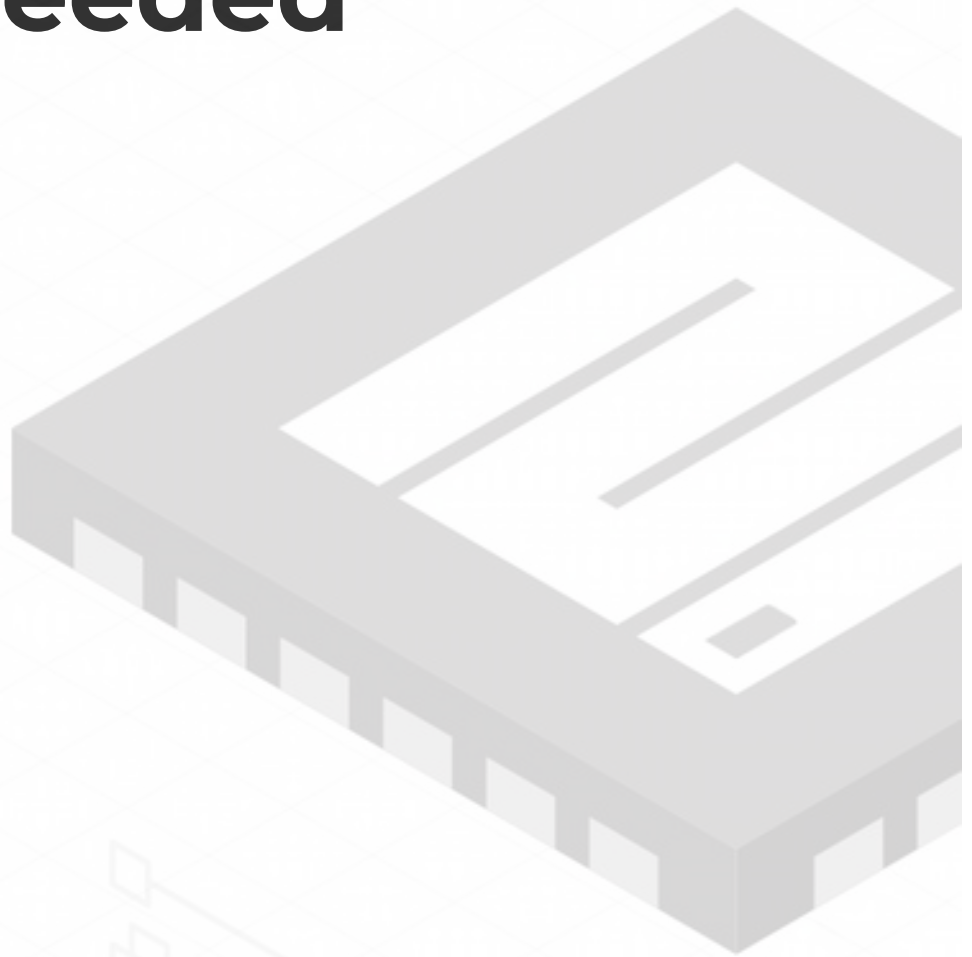
Power consumption	5.0 V
Running at 96 MHz	23 mA
Idling at 96 MHz	5 mA
Running at 48 MHz	13 mA
Idling at 48 MHz	4 mA
Sleep full RAM retention	180 uA
Deepsleep (backup retention only)	6 uA

- SD card reader
- Micro USB connector power and data
- No additional power supply needed
- 4 LED red, green, orange, blue
- 2 switches USR and RST
- Internal flash: 512k RAM
- Frequency 96MHz
- IO pins: 30
- 18 PWM
- 16 A/D
- 7 independent timers
- 3 UART
- 2 I2C
- 2 SPI
- Power supply input range on V+/VBAT: 3.6V–16V

NO IDE needed

3 ways to use a pyboard

- from file main.py
- Remote script
- REPL promt

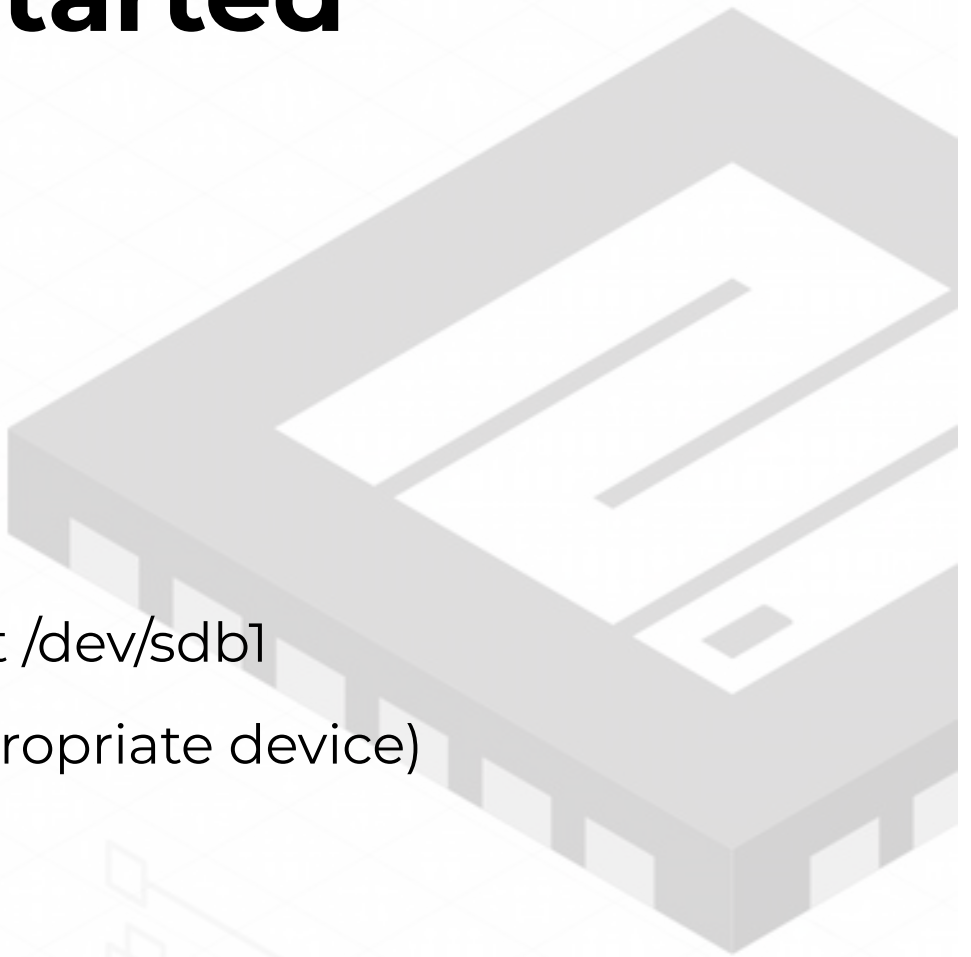


Plug it in

All machines

please connect your
pyboard with the MicroUSB cable

Getting started



LINUX

- Removable medium: PYBFLASH

Ubuntu: mount automatically and
pop-up with the pyboard folder

Manually mount
lsblk → list of connected drivers mount /dev/sdb1

(sdb1 needs to be replaced by the appropriate device)

- Opening the pyboard USB drive
- Editing main.py
- Resetting the pyboard

Getting started

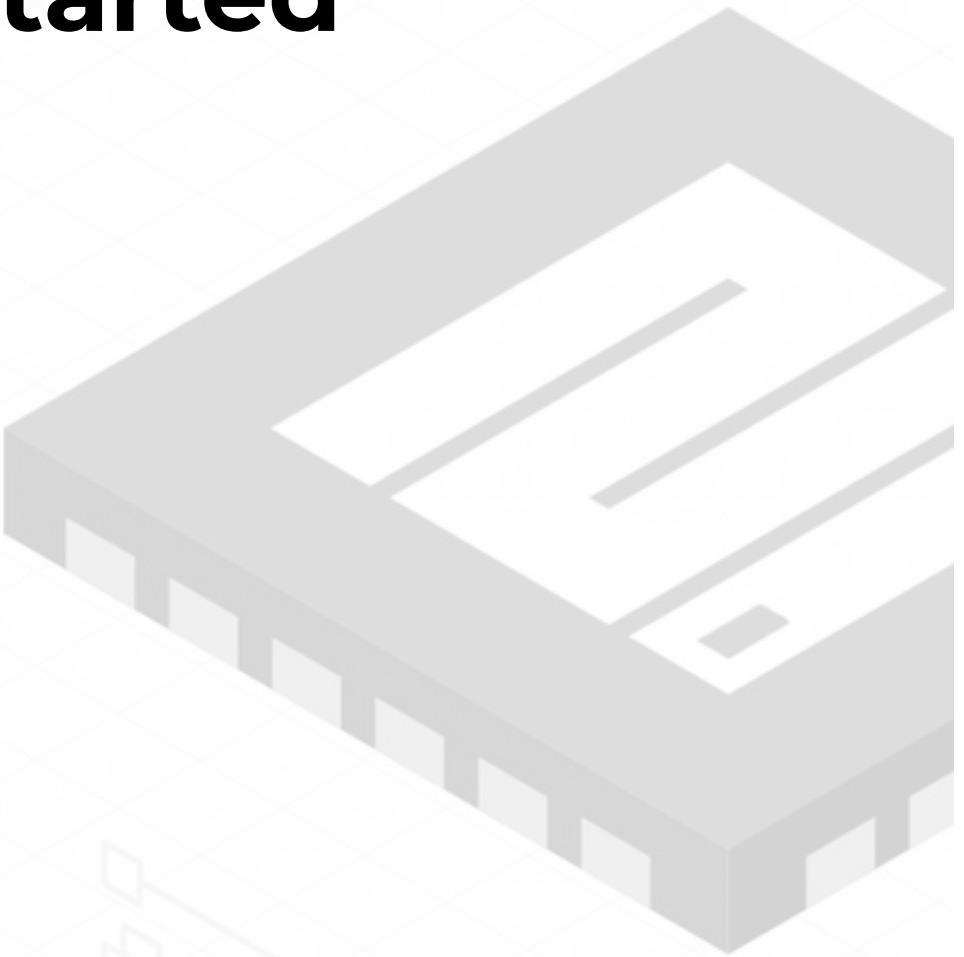
MAC

Removable disc on the desktop

PYBFLASH

→ click to open pyboard folder

- Opening the pyboard USB drive
- Editing main.py
- Resetting the pyboard



Getting started

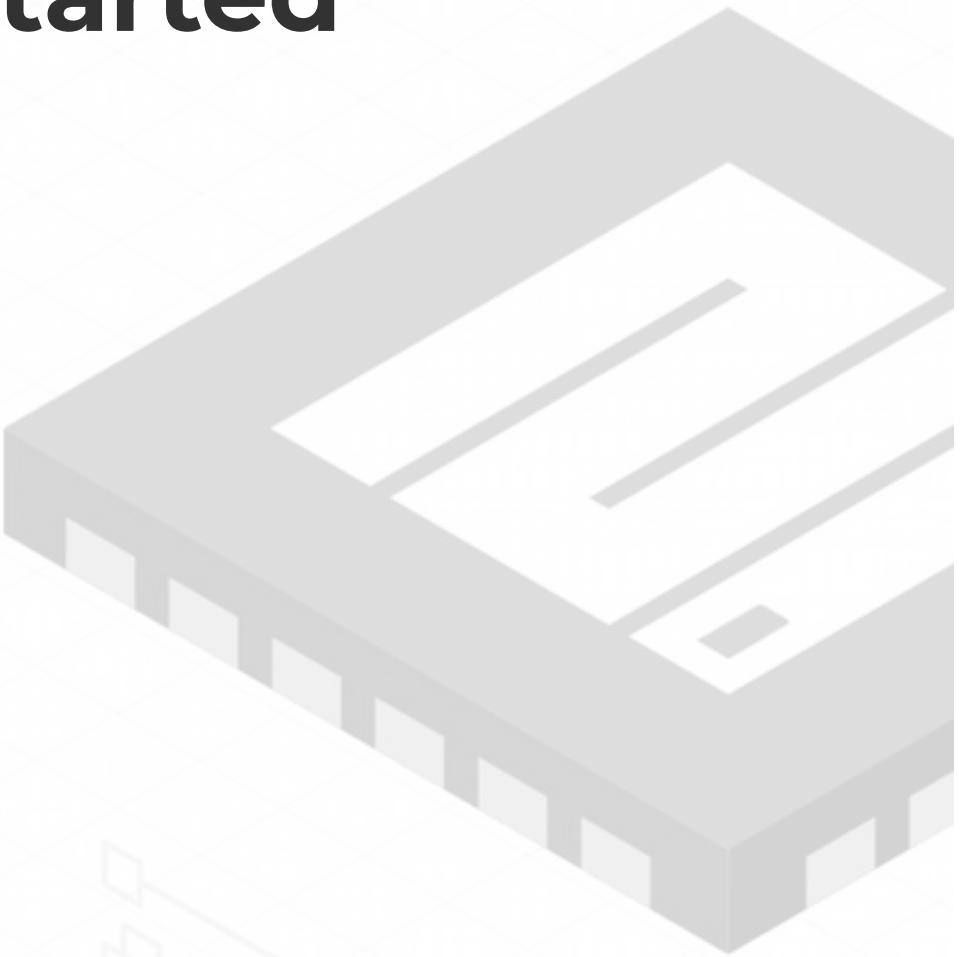
MAC

Removable disc on the desktop

PYBFLASH

→ click to open pyboard folder

- Opening the pyboard USB drive
- Editing main.py
- Resetting the pyboard



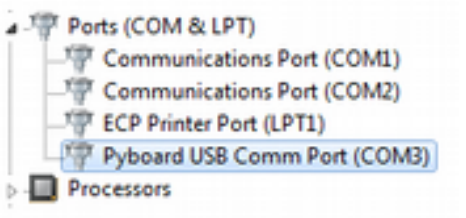
Getting a MicroPython REPL promt

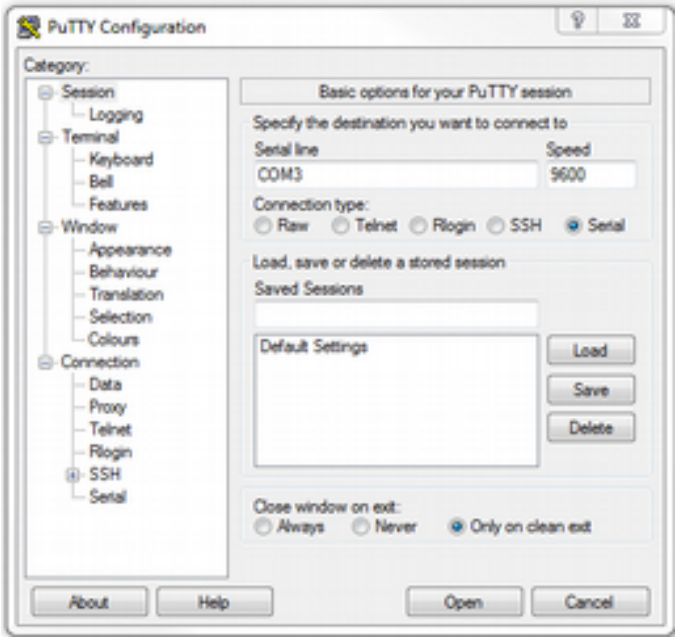
- Easy testing of your code

  WINDOWS

  Go to the device manager

- find pyboard in list of devices
- right click on the
  pyboard device
  for the COM port (e.g. COM 3)





MAC & LINUX OS

MAC

- open a terminal
- screen /dev/tty.usbmodem*
- CTRL-A CTRL- for exit screen

LINUX

- picocom /dev/ttyACM0
- rshell

micropython.org/unicorn/

MicroPython HOME FORUM DOCS QUICK-REF DOWNLOAD STORE CONTACT

```
MicroPython f663b70 on 2017-09-10; unicorn with Cortex-M3
Type "help()" for more information.
>>>
```

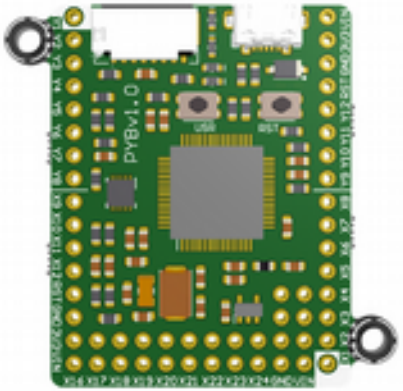
```
# four LEDS numbered 1 to 4

import time
import pyb

for i in range(1000):
    pyb.LED((i%4) + 1).toggle()
    time.sleep_ms(100)

```

Clock Speed 0.00 MHz

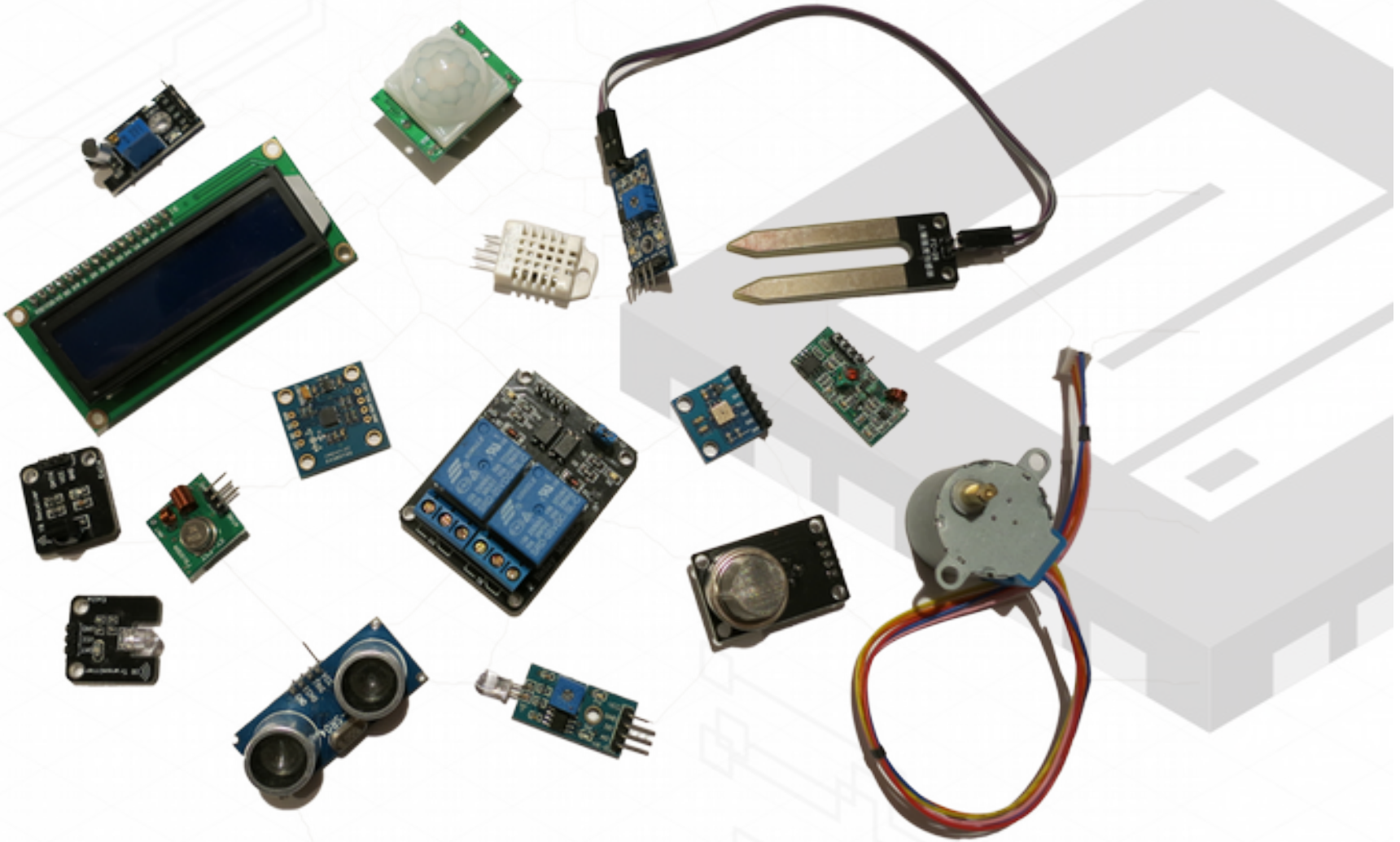
Binary: Pyboard Ram Size: 64KB Stack Size: 8KB Reset

Run Script LEDs

LED
SERVO
ADC



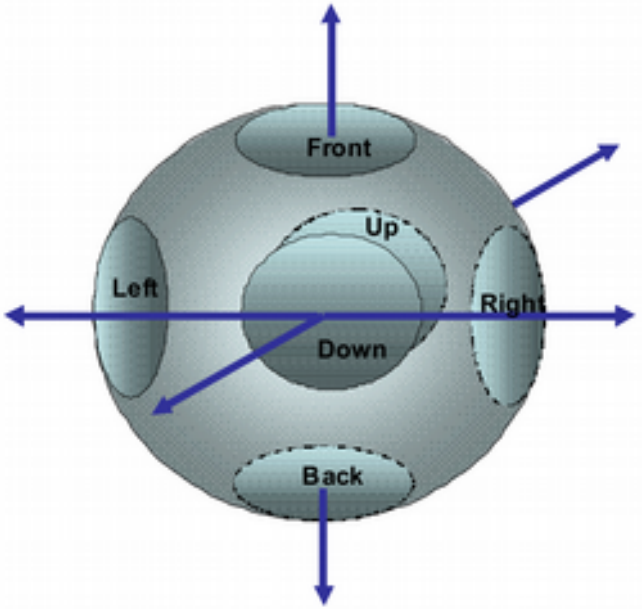
Sensors



https://www.mysensors.org/about/components

MMA7660 3-Axis Orientation/Motion Detection Sensor

- 6-bit digital value
- I2C interface on the pyboard
- I2C uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors. Typical voltages used are +5 V or +3.3 V
- X, y, z direction

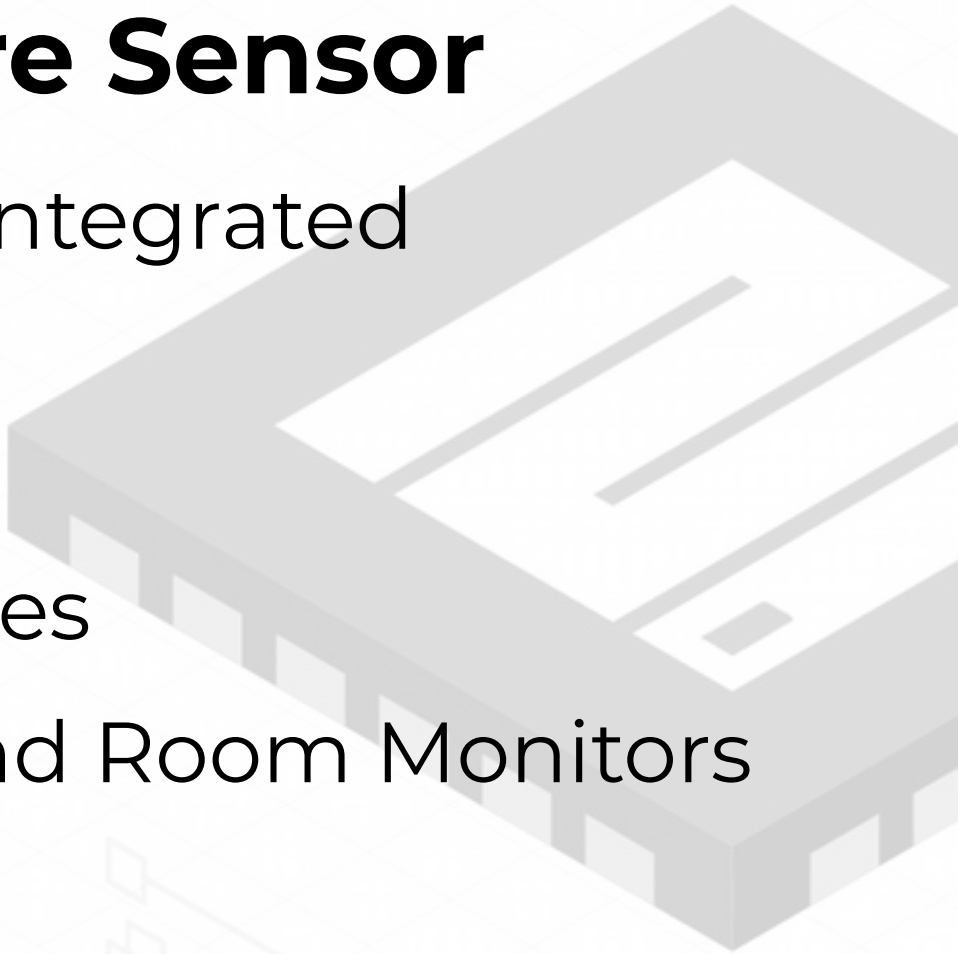


HDC1080 Humidity and Temperature Sensor

- Digital humidity and integrated temperature sensor
- I2C Interface

  Used in Medical Devices

  Smart Thermostats and Room Monitors



LCD160CRv1.0 Display Skin

- Controlled with intelligence and optimised for Python programming
- Integrated touch controller
- Low memory RAM footprint
- Library integrated in MicroPython

pyboard Demo

```
# four LEDS numbered 1 to 4

import time
import pyb
for i in range(1000):
    pyb.LED((i%4) + 1).toggle()
    time.sleep_ms(100)
```

```
# push the USR button on the
# pyboard to flash the LEDs!
import time
import pyb

while True:
    if pyb.Switch().value():
        pyb.LED(1).on()
    else:
        pyb.LED(1).off()
    time.sleep_ms(50)
```

```
# inline assembler

@micropython.asm_thumb
def asm_add(r0, r1):
    add(r0, r0, r1)
print(asm_add(1, 2))
```

```
led = pyb.LED(4)
intensity = 0
while True:
    intensity = (intensity + 1) % 255
    led.intensity(intensity)
    pyb.delay(20)
```

Demo with HDC1080

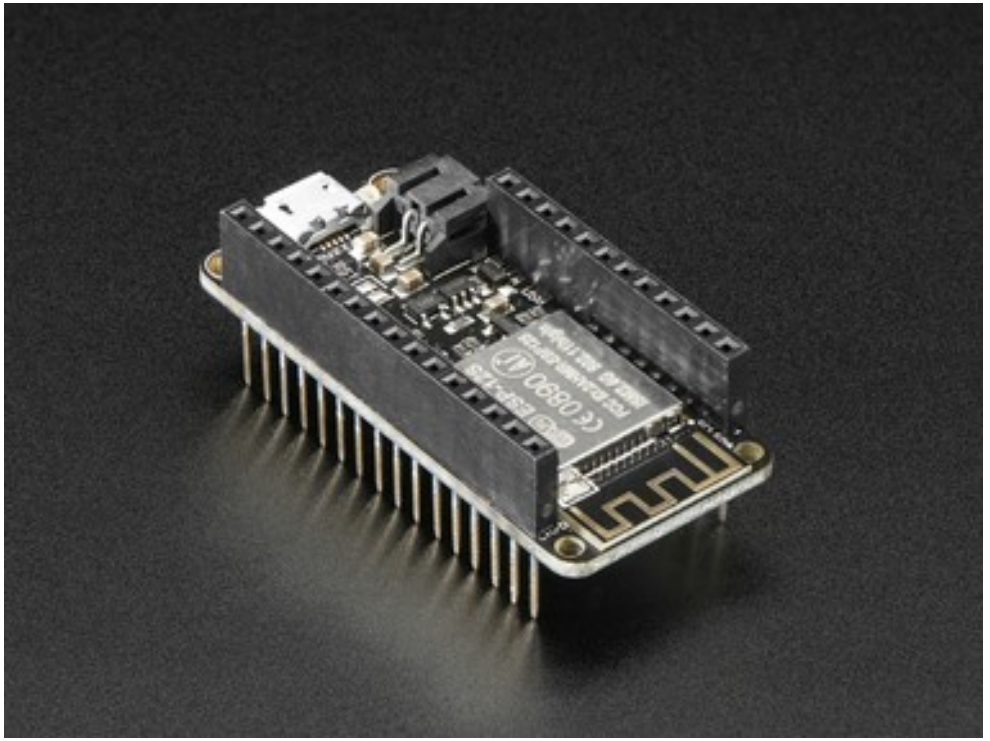
How to find an I2C sensor on the pyboard

```
>>> i2c = machine.I2C(sda=machine.Pin('X10'), scl=machine.Pin('X9'), freq=400000)
```

```
>>> i2c.scan()
```

returns the address of the I2C device!

ESP8266 on the Feather Huzzah



Tutorial

https://github.com/tine3700/micropython/tree/master/docs/pyboard/tutorial

PART I

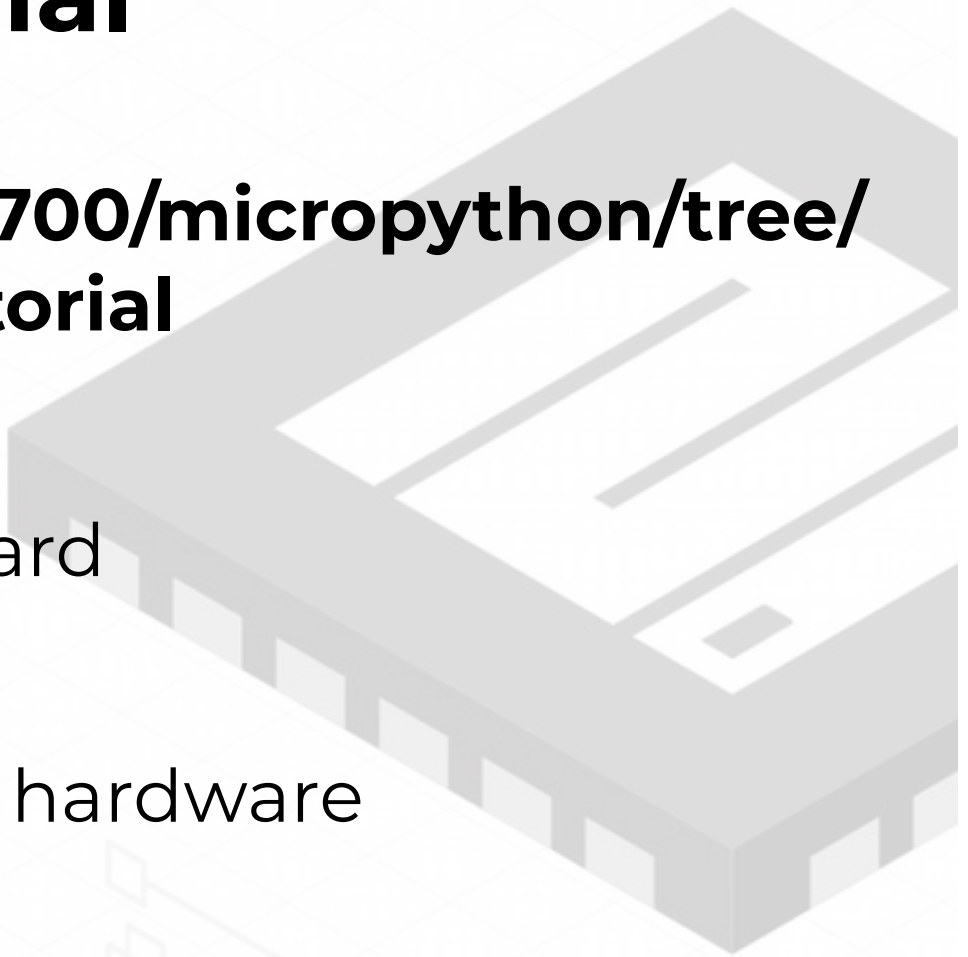
Basics to know your pyboard

PART II

Advanced with additional hardware

PART III

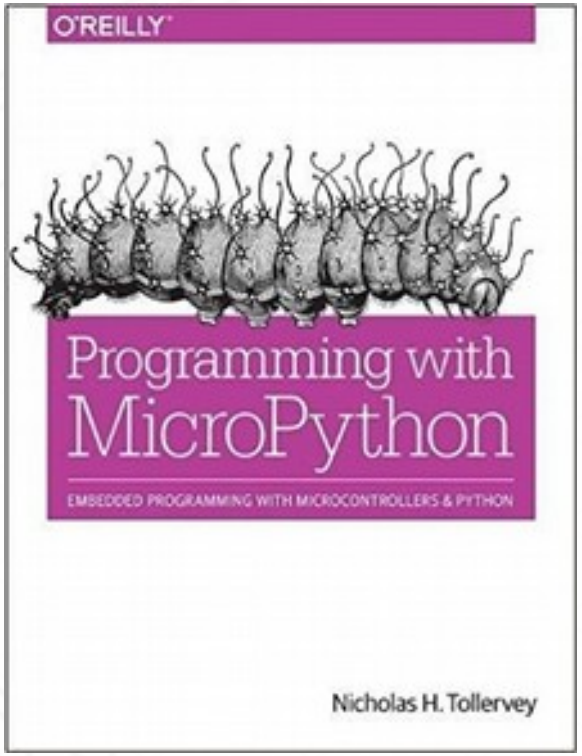
Building your own device



Why is MicroPython special

- Python is easy to learn and understand
- To get the most out of your application with mixing code, even assembler and C
- for beginners and advanced users
- MIT license – free to use for private and industrial projects

Programming with MicroPython by Nicholas H. Tollervey



The Zen of MicroPython

Code,
Hack it,
Less is more,
Keep it simple,
Small is beautiful,

Be brave! Break things! Learn and have fun!
Express yourself with MicroPython.

Happy hacking!

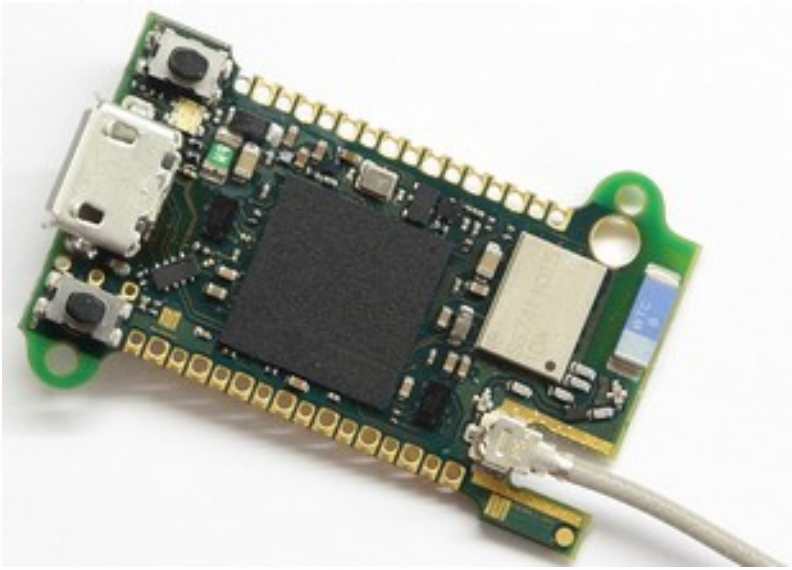
Real World applications

- Traffic management device certified by national institute of metrology (state: in production)
- Contact-free opto-electronic measurement system for medical use (state: international certification in progress)

What‘s next

- Schools/teaching
  (micro:bit, pyboard, high-schools and universities)
- Continued software/hardware development
- Easier embedding
- Development of new boards and pyboard skins
- New pyboard D-Series

Hello, I’m the new one



Cortex M7 runs at 216 MHz with 256k RAM
PCB size: 33.5 mm x 19 mm
Easy to use low power modes
Weight: 2.4 gramms