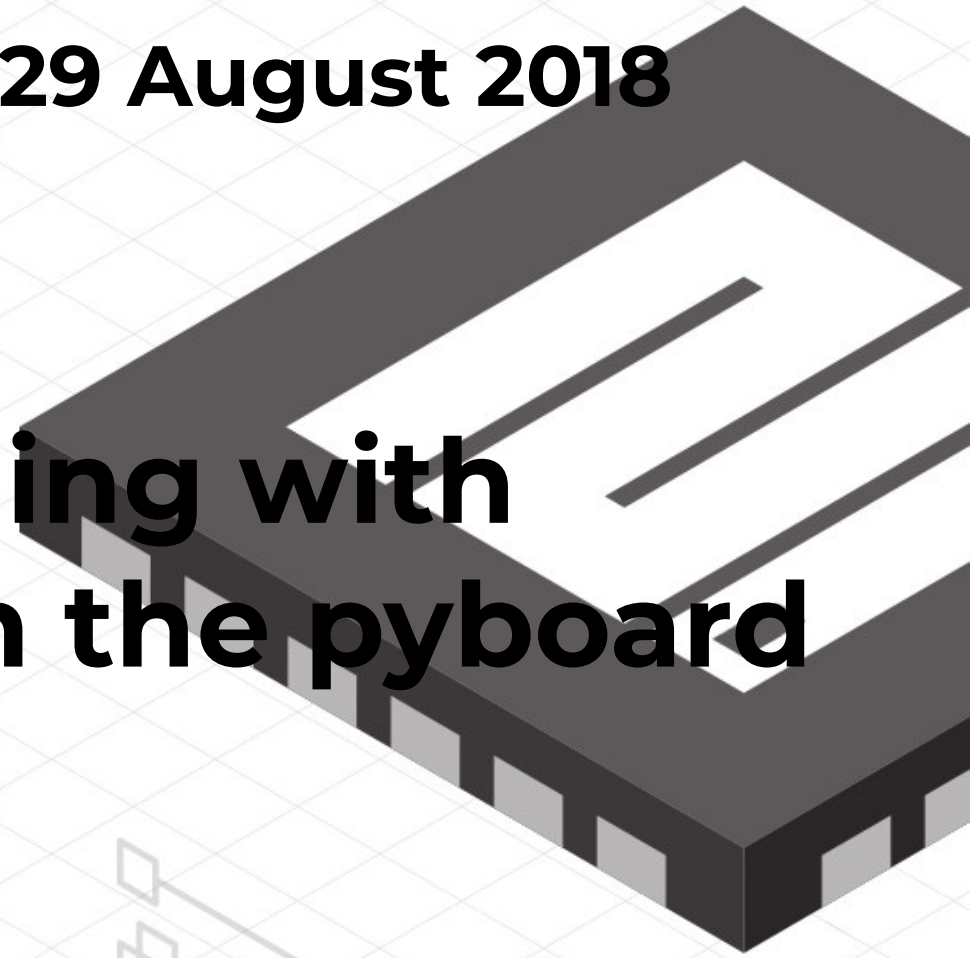
EDUCODE, Brussels 27-29 August 2018



# Programming with MicroPython on the pyboard

Christine Spindler – George Robotics Ltd
The company behind MicroPython

# Agenda

(1) What is MicroPython?

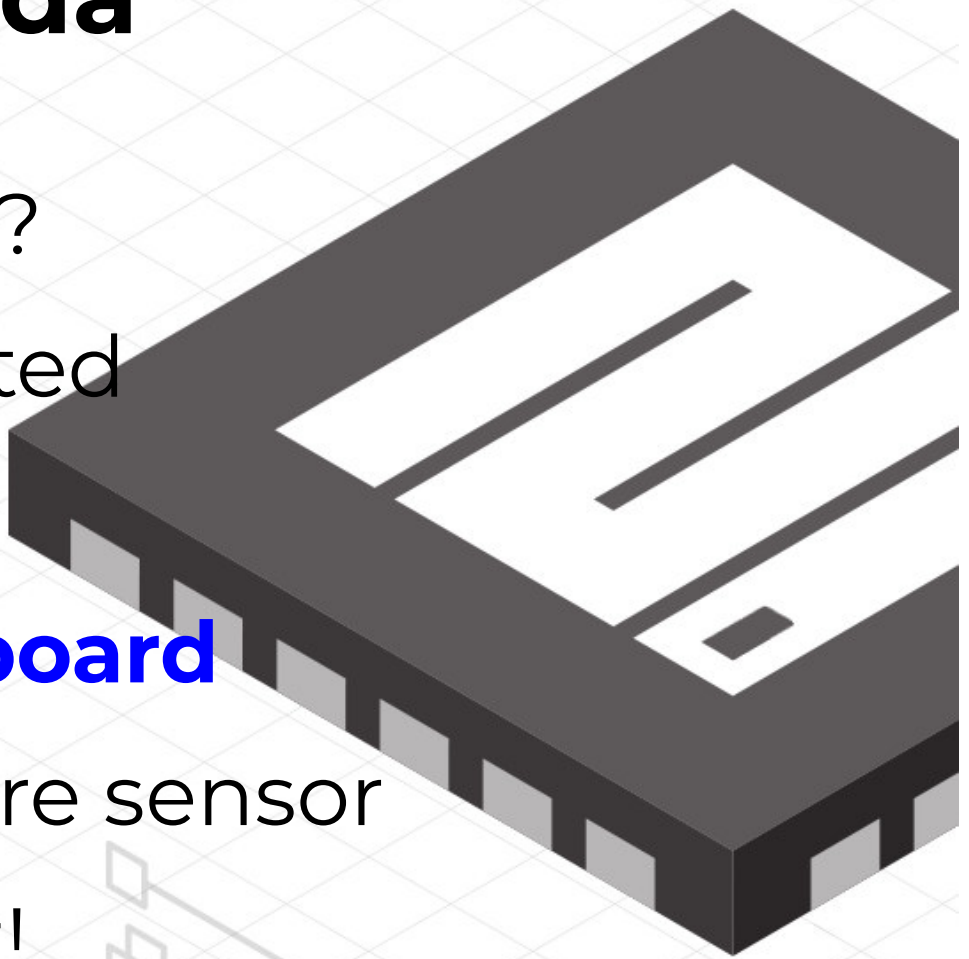
(2) How everything started

(3) First program with

**MicroPython on pyboard**

(4) Dive into Temperature sensor

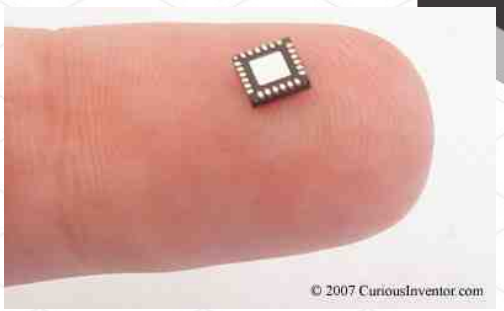
(5) What is coming next!

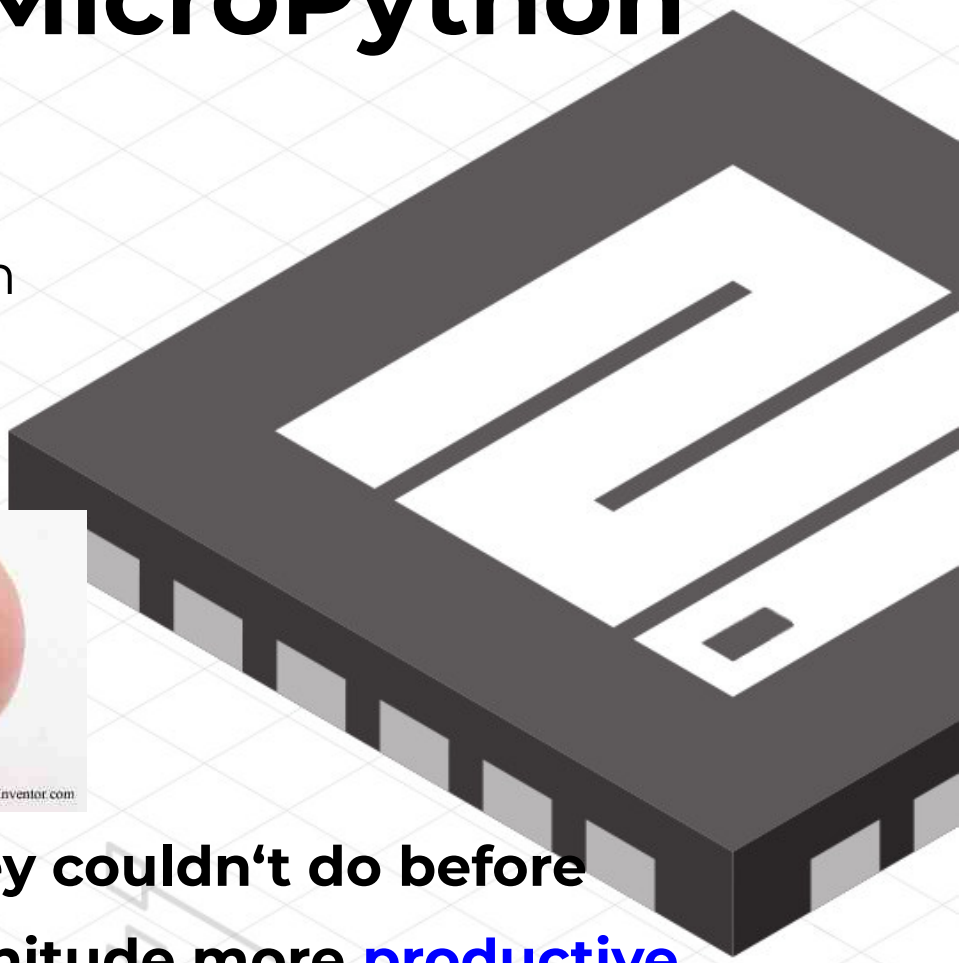


# Motivation for MicroPython

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

Need a way to **control** all these sophisticated devices.

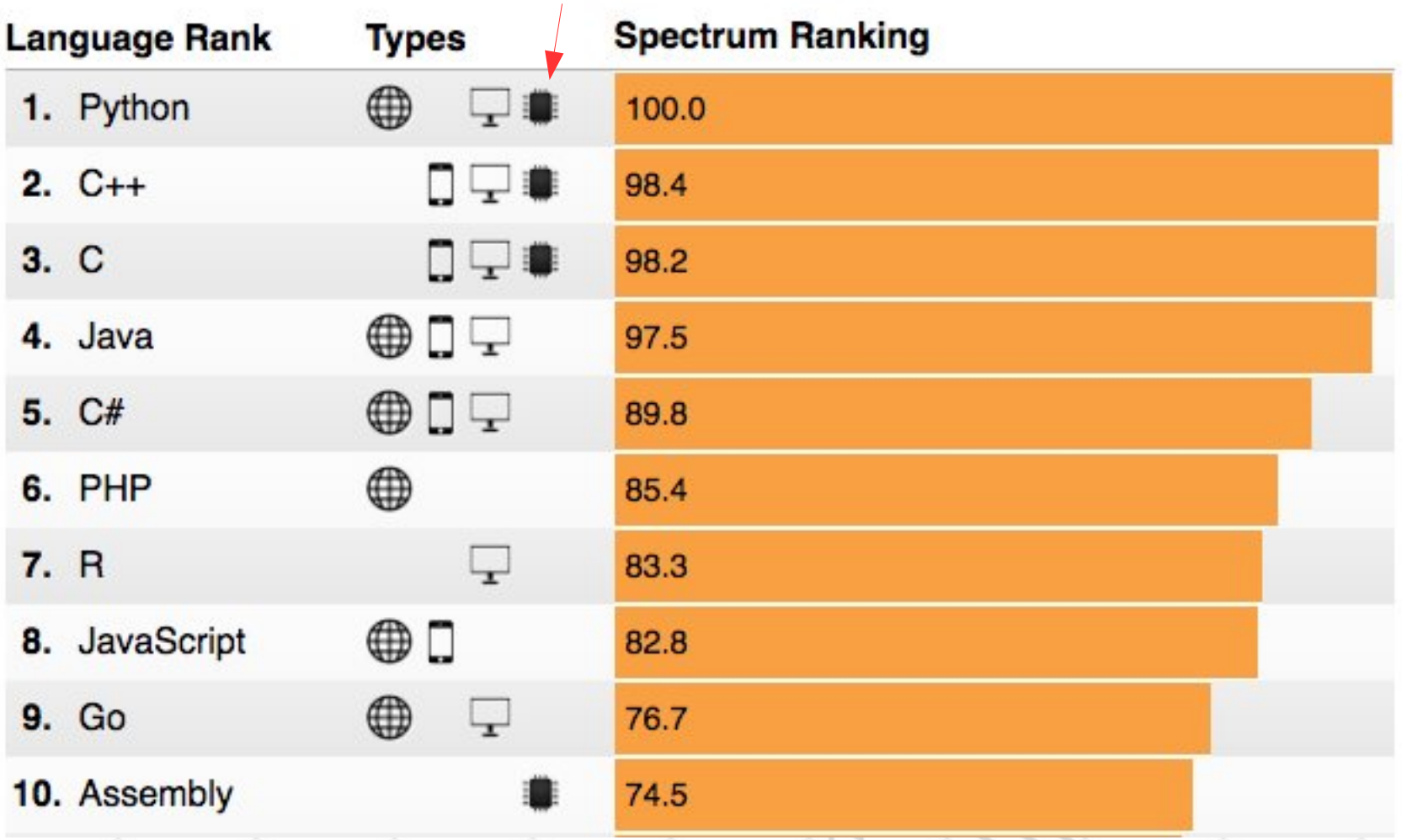


**MicroPython**

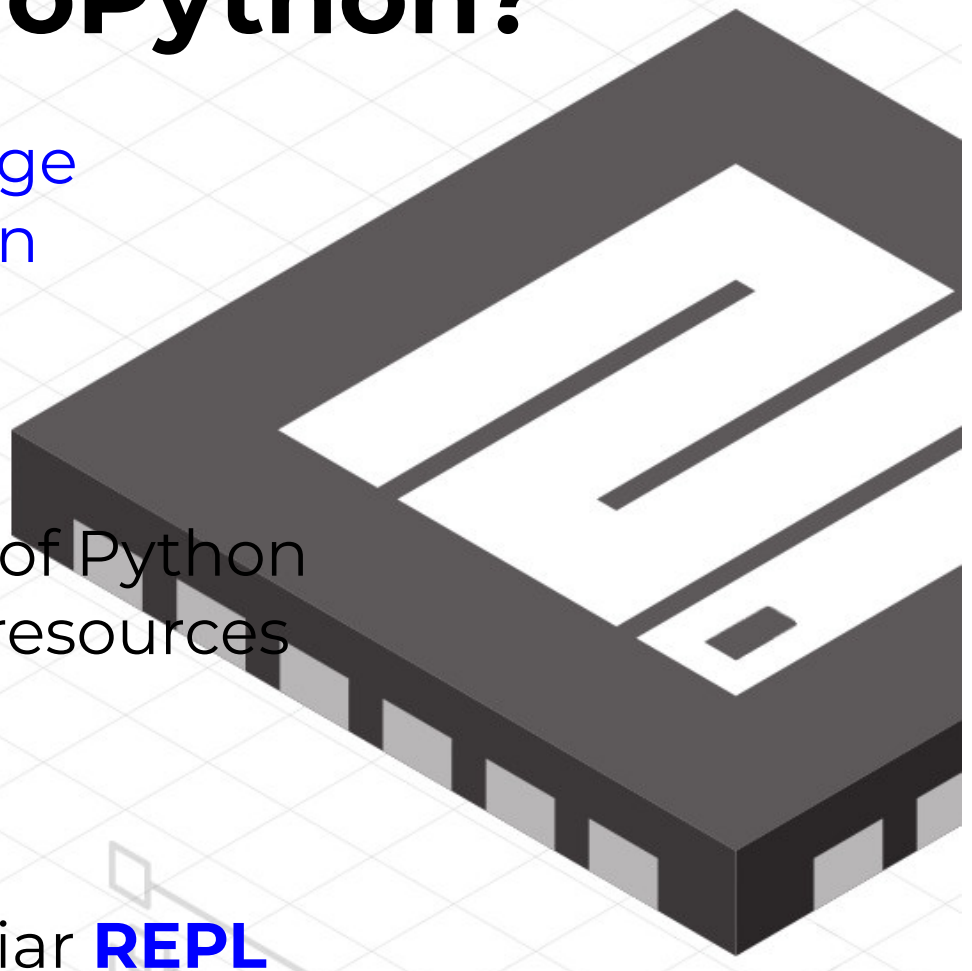
- **allows beginners to do things they couldn‘t do before**
- **Professionals be an order of magnitude more productive**
- **Building devices easier and more accessible**

# Python IEEE


Language Rank
Types
Spectrum Ranking
1. Python
100.0
2. C++
98.4
3. C
98.2
4. Java
97.5
5. C#
89.8
6. PHP
85.4
7. R
83.3
8. JavaScript
82.8
9. Go
76.7
10. Assembly
74.5

# What is MicroPython?

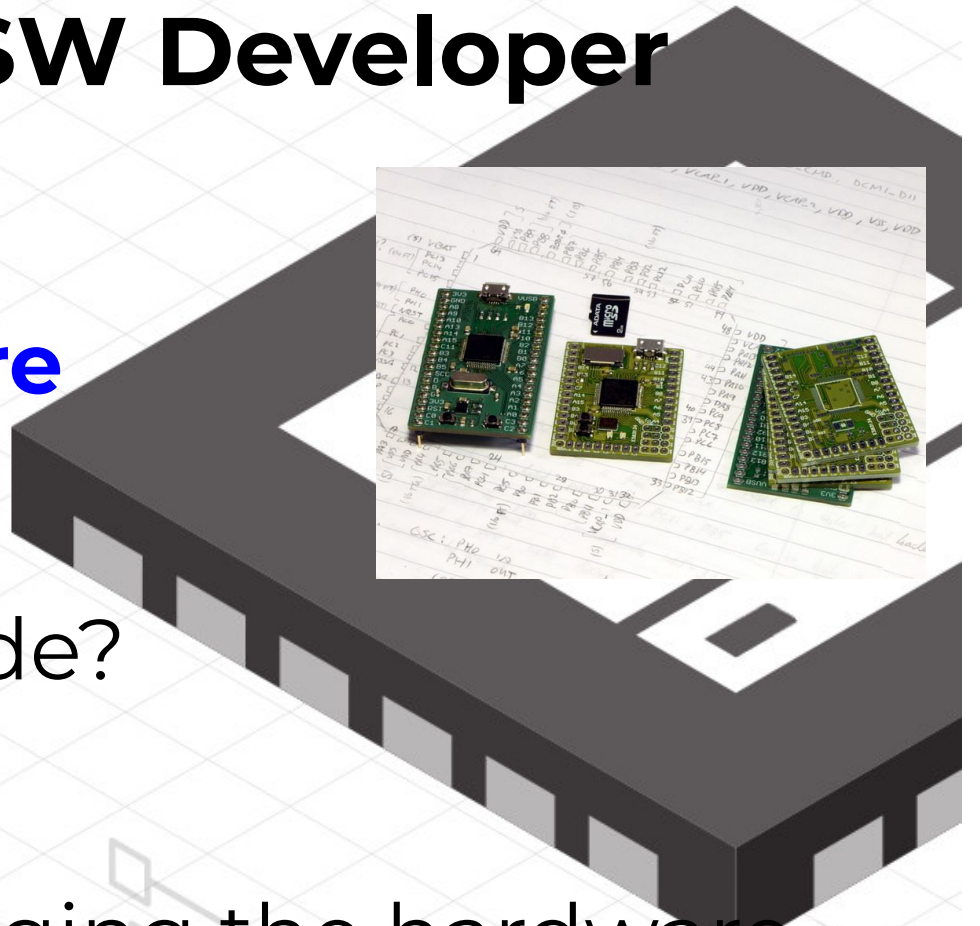
A **powerful** and **modern** language
large community made to run on
constrained/embedded systems



MicroPython is
- a complete reimplementation of Python
- 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

# Difference Embedded SW Engineer & SW Developer

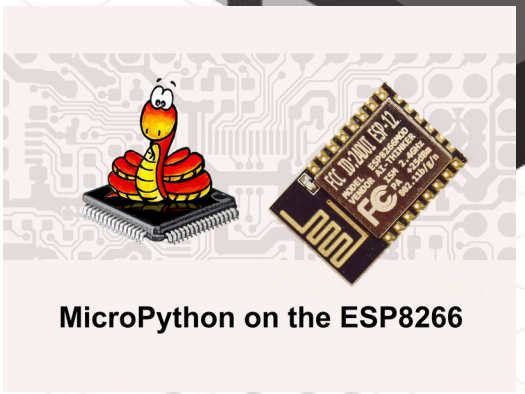


- Knowing the **hardware**
- **PC** vs. **PCB**
- How many lines of code?
- Different debugging
- Controlling and managing the hardware

# 2013 Crowdfunding via Kickstarter

# 2016 MicroPython in school

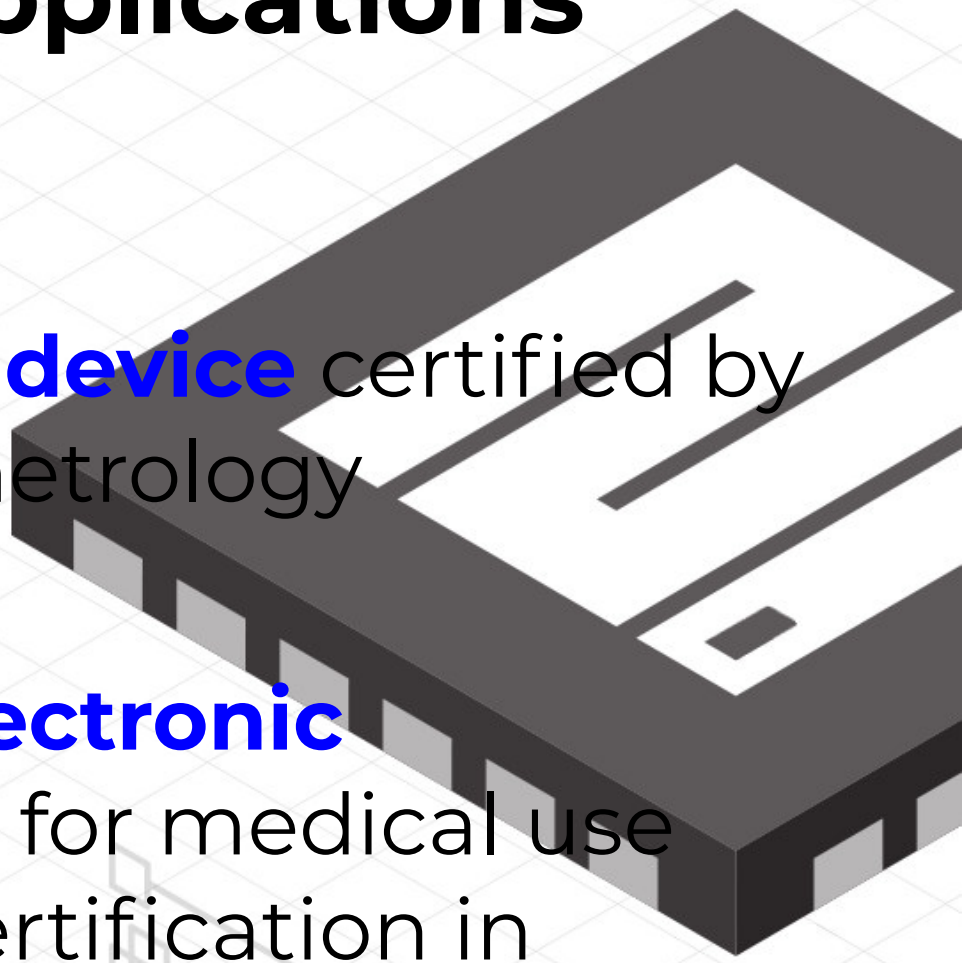
- **BBC micro:bit** project brought 1 Mio units to 7 year old children in the UK
- 2nd Kickstarter ESP8266 support
- New Logo

# Facts & Figures

- MicroPython is a public project on GitHub with **6900** stars
- ranking in the top of **100** most popular C/C++ projects
- **MIT license**
- Contributions come from many people (200+) with many different systems leads to: more robust code and build system
- more Features and supported hardware
- active forum

# Real World applications



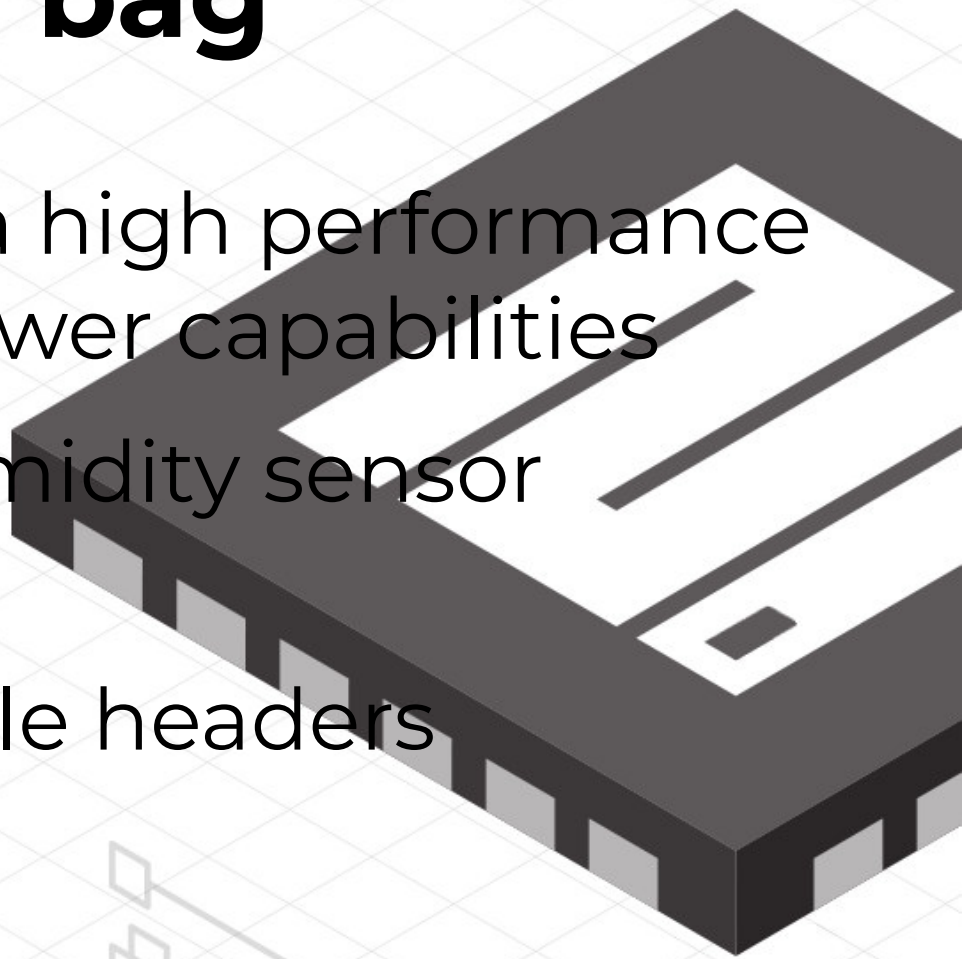
- **Traffic management device** certified by national institute of metrology (state: in production)
- **Contact-free opto-electronic** measurement system for medical use (state: international certification in progress)

# Maker Projects

Quadrocopter

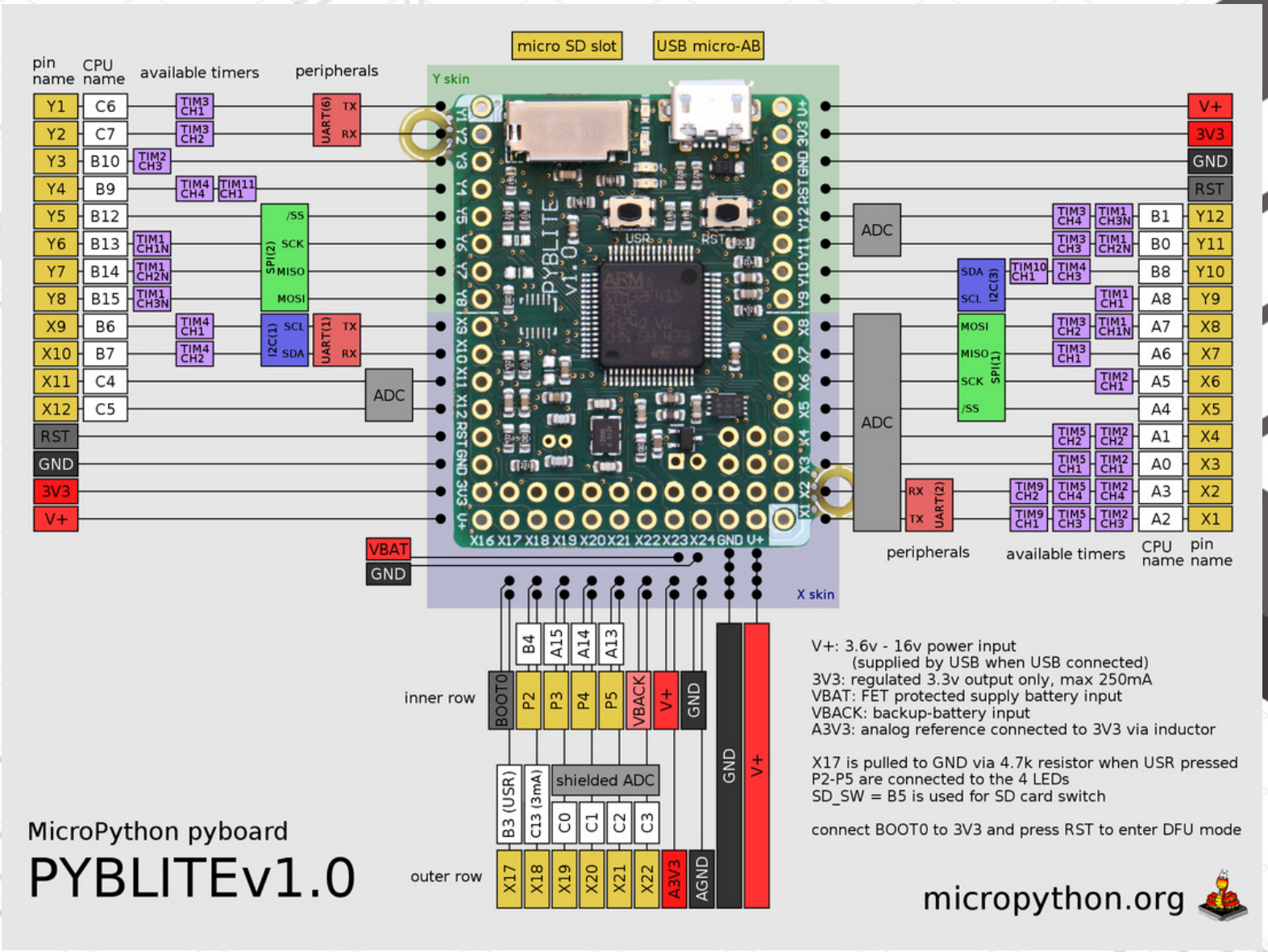
Remote, wireless
weather station network

# In your bag



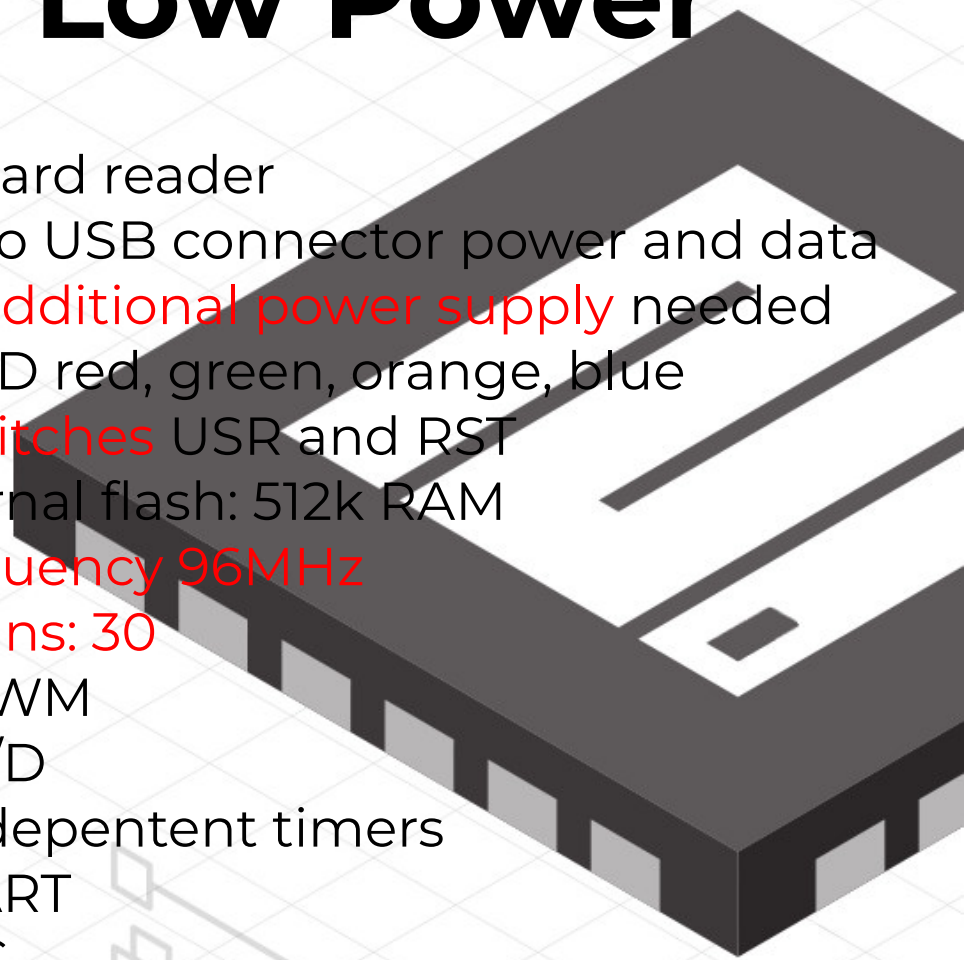
- **pyboard** lite enables a high performance processor with low power capabilities
- Red Temperature/Humidity sensor **HDC1080**
- Jumper Wire with male headers
- MicroUSB cable
- STICKERS

# The pyboard lite Pin layout



# pyboard lite is Low Power

| Power consumption | |
|---|---|
| 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 |

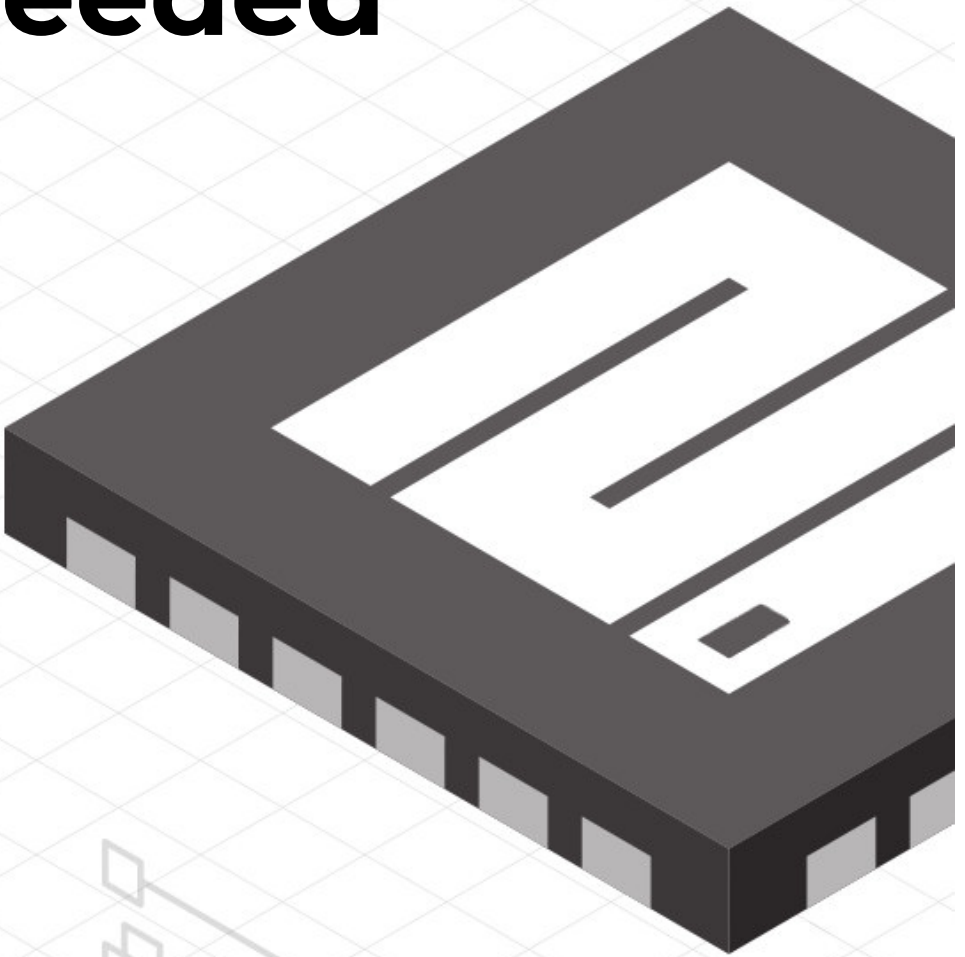


- SD card reader
- Micro USB connector power and data
- No additional power supply needed
- 4 LED red, green, orange, blue
- 2 switches USR and RST
- Internal flash: 512k RAM
- Frequency 96MHz
- IO pins: 30
- 18 PWM
- 16 A/D
- 7 independent timers
- 3 UART
- 2 I2C
- 2 SPI
- Power supply input range on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

# NO IDE needed

**3 ways** to use a pyboard

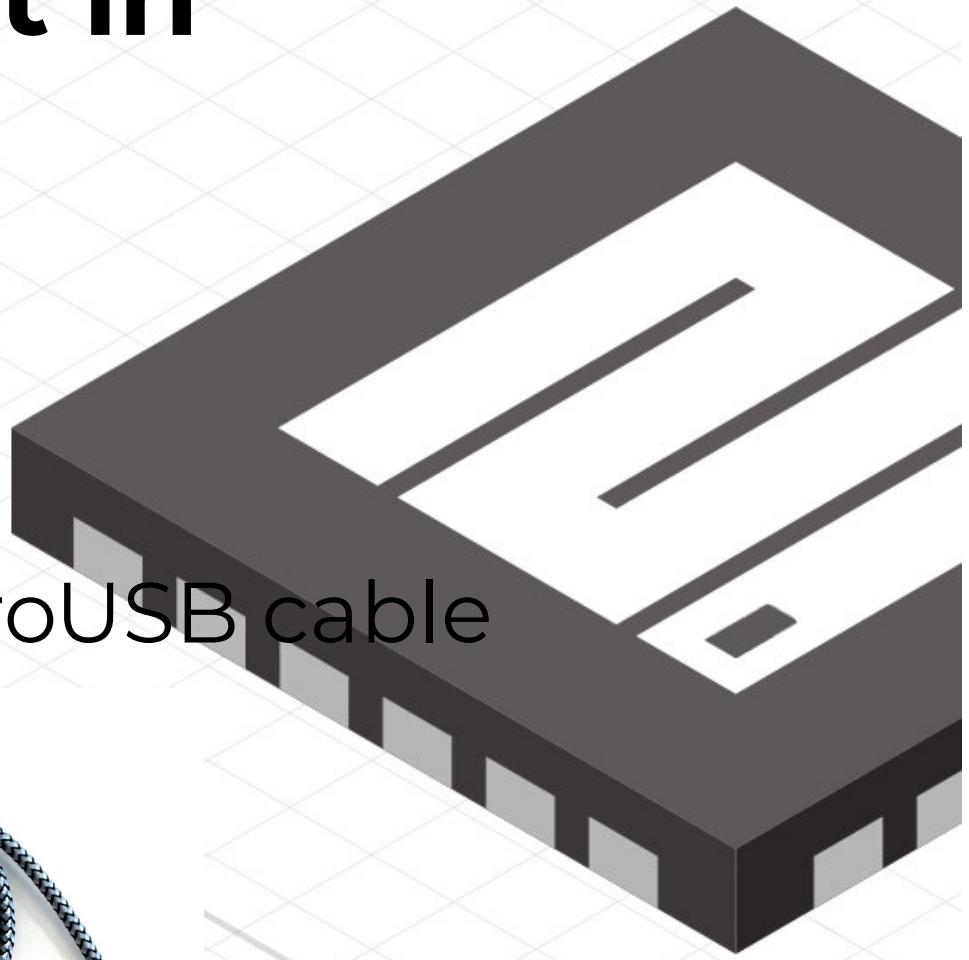
- from file **main.py**
- **Remote** script
- **REPL** promt



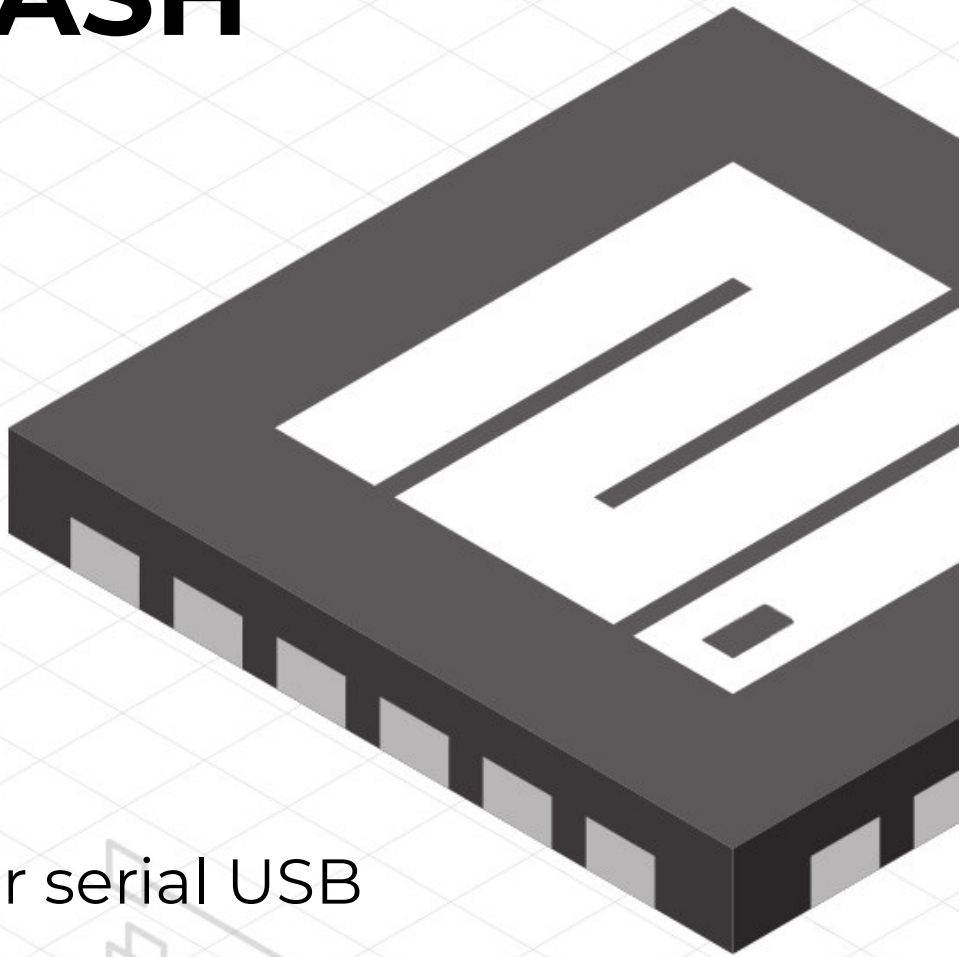
# Plug it in

All machines

please connect your
pyboard with the MicroUSB cable



# PYBFLASH



## 5 files

- boot.py
  executed when the
  pyboard boots up
- **main.py**
  contains your python program
- README.txt
- Pybcdc.inf – Windows driver for serial USB
- **HDC1080_logdata.py** – driver for the Temperature Sensor

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.
>>>
```

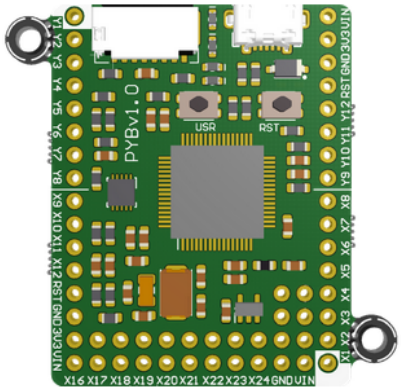
```
# four LEDS numbered 1 to 4

import time
import pyb

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

```

Clock Speed 0.00 MHz

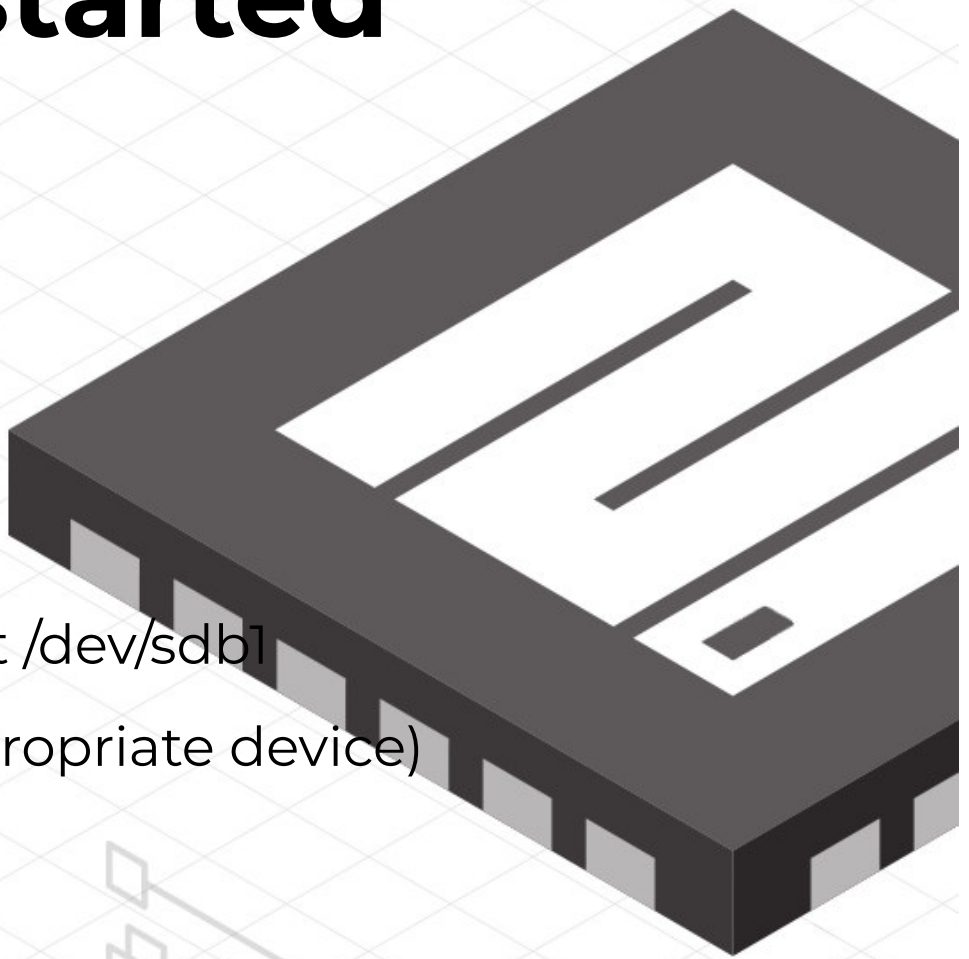
Binary: Pyboard Ram Size: 64KB Stack Size: 8KB Reset

Run Script LEDs

- LED
- SERVO
- ADC



# Getting started



**LINUX**

- Removable medium: PYBFLASH

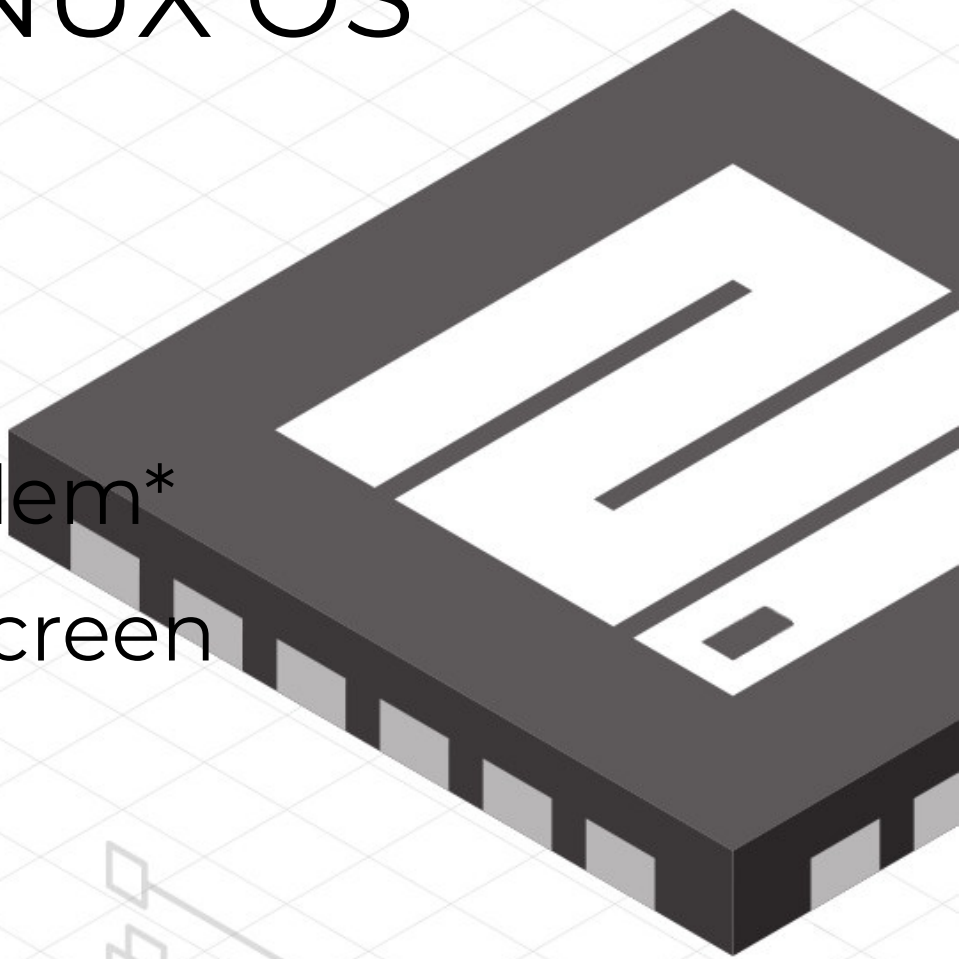
Ubuntu: mount automatically and
pop-up with the pyboard folder

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

# MAC & LINUX OS



**MAC**

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

**LINUX**

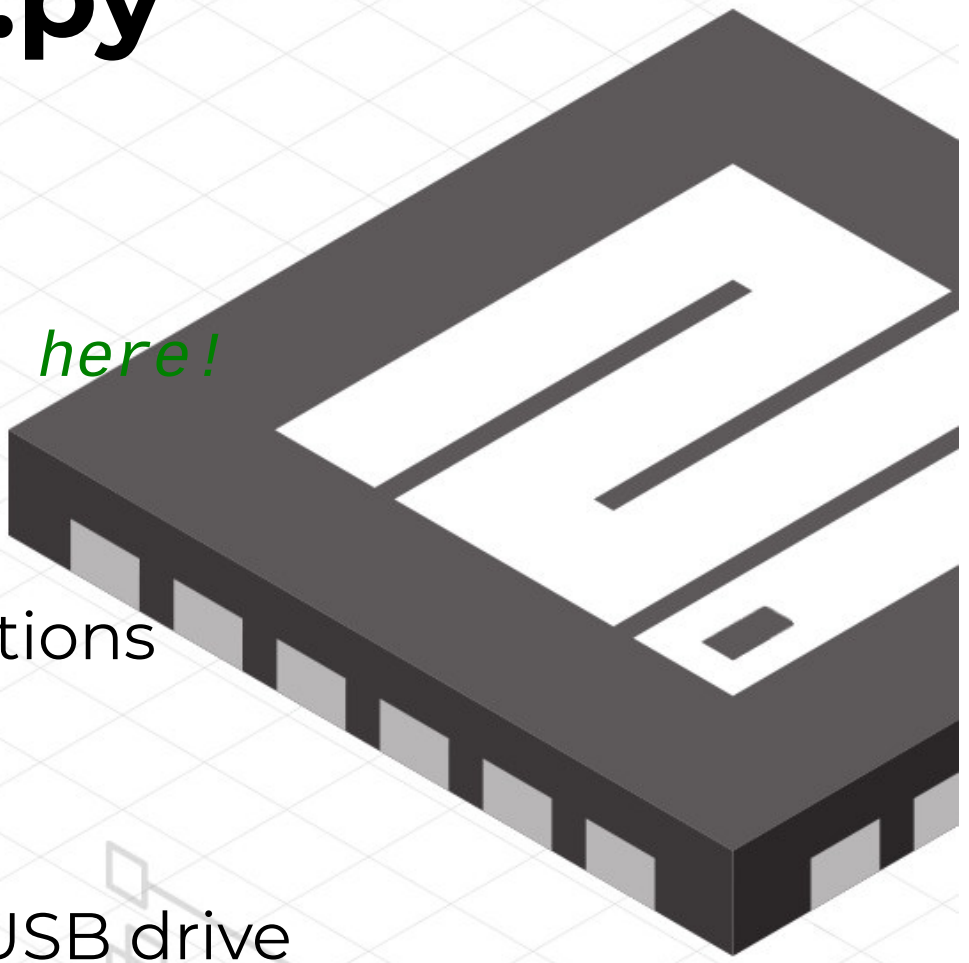
- picocom /dev/ttyACM0
- rshell

# main.py

- open main.py in text editor

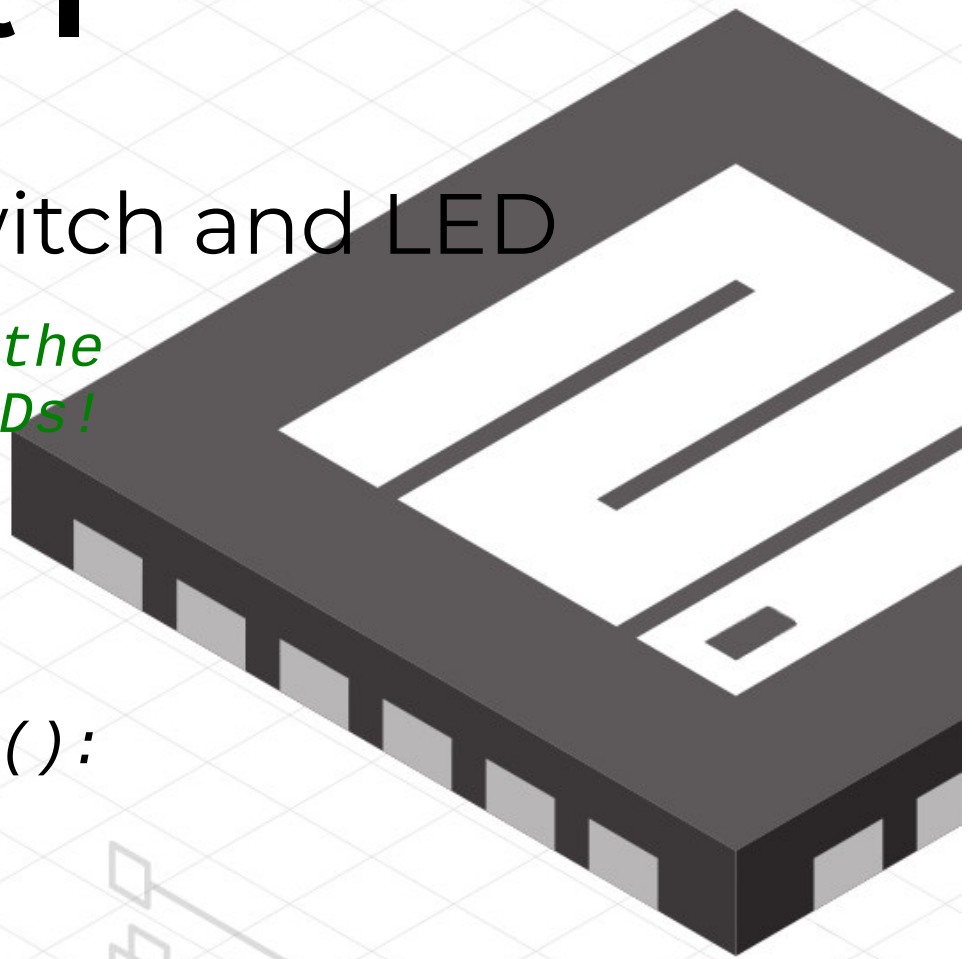
```
#main.py – put your code here!
import pyb
pyb.LED(4).on()
```

- pyb module contains all functions and classes of the pyboard
- save and close main.py
- eject/unmount the pyboard USB drive
- now press reset button



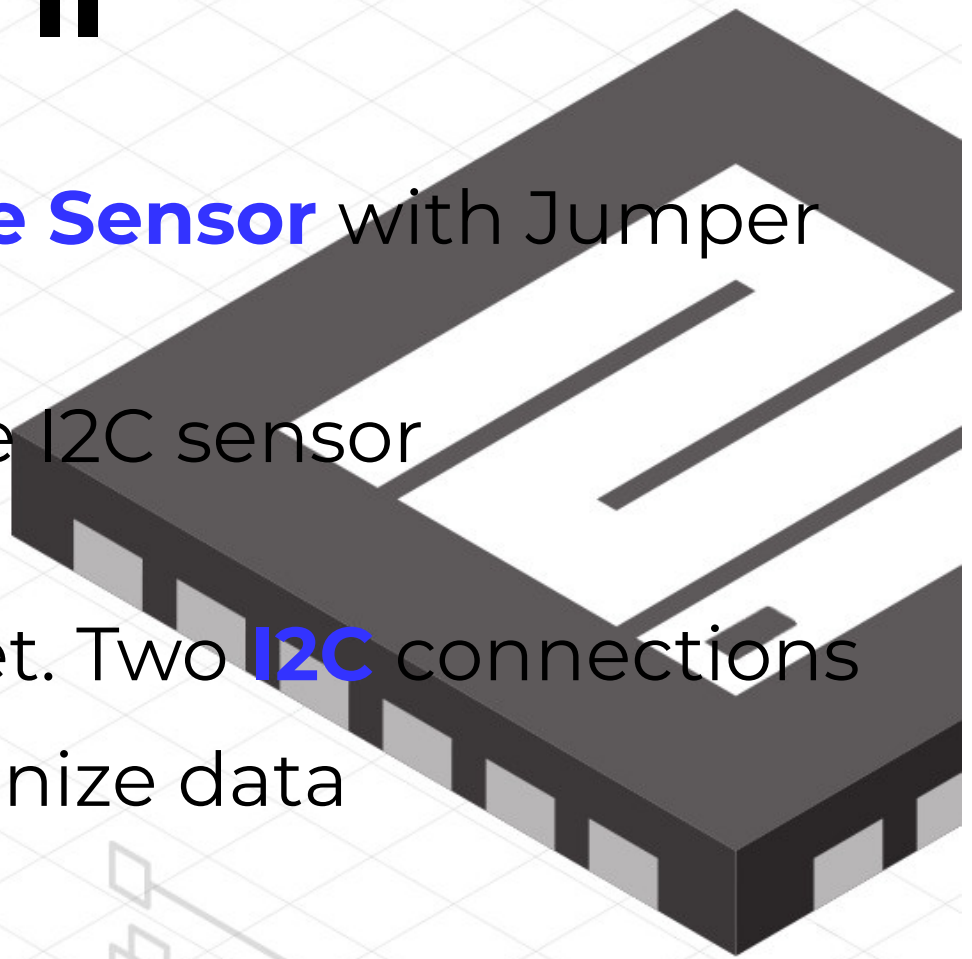
# Part I

## First program with Switch and LED



```
# 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)
```

# Part II



- Connecting **Temperature Sensor** with Jumper Wires
- Four wires to connect the I2C sensor temperature board
- Look at the pyboard sheet. Two **I2C** connections

  **SCL**: clock line to synchronize data

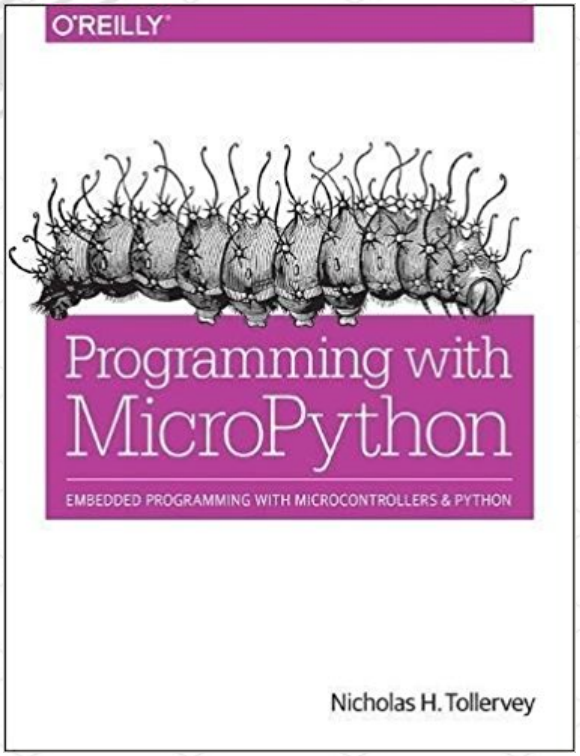
  **SDA**: data line

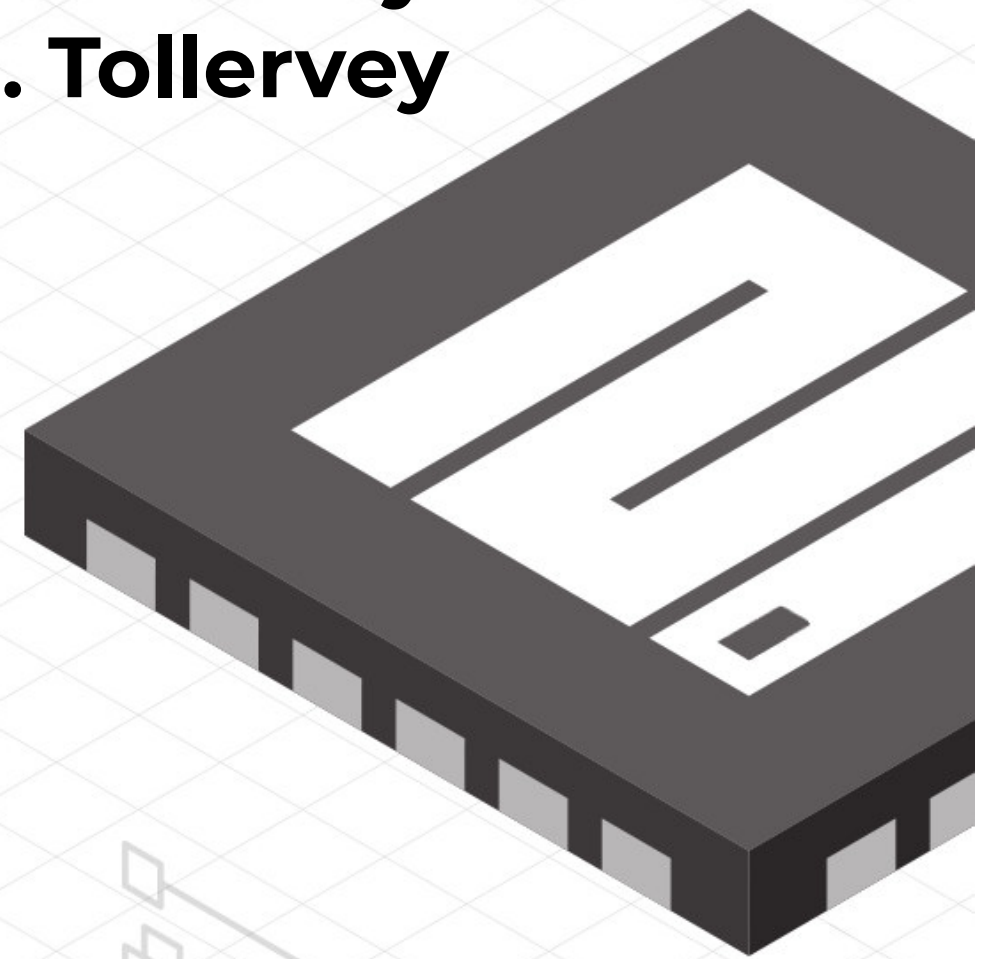
  **GND** and **VCC** to 3V

# Temperature & Humidity

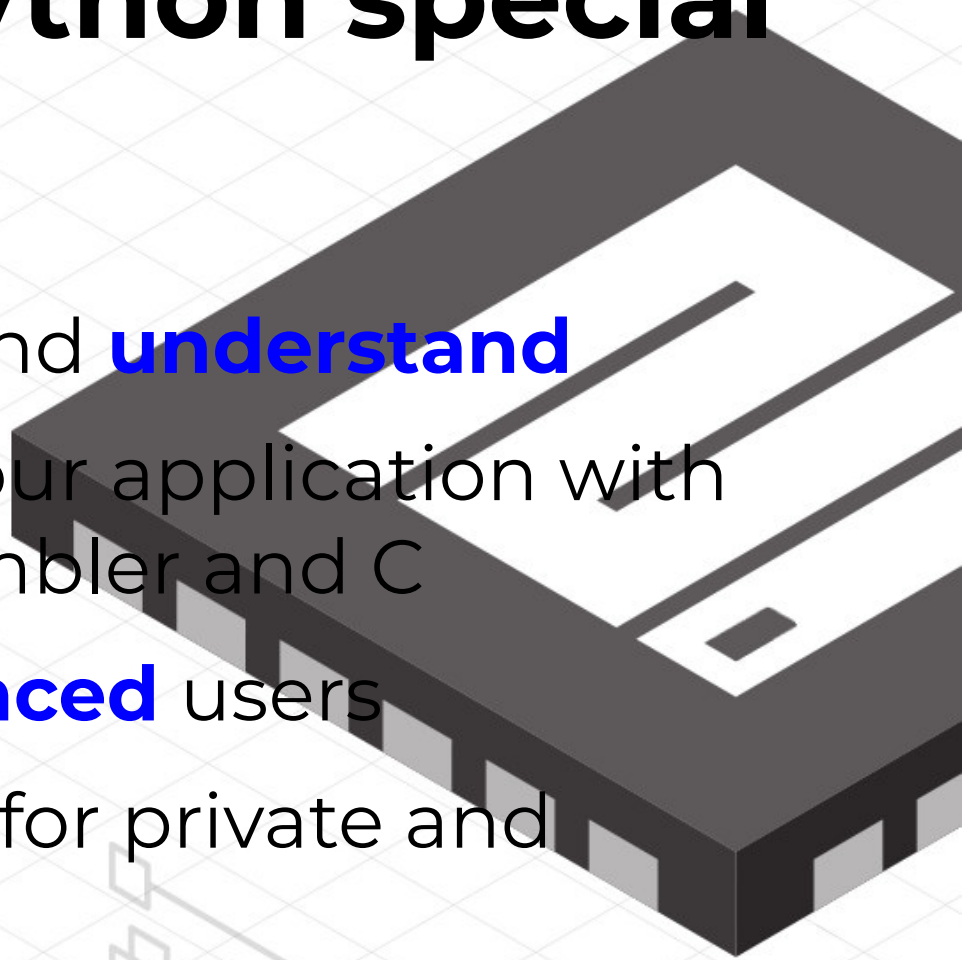
- Logging Temperature and Humidity data in a .txt file on the pyboard internal flash
- Or to an SD-card

# Programming with MicroPython by Nicholas H. Tollervey



# 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
- **Tools** instead of toys for teaching

micropython.org
docs.micropython.org
**christine@micropython.org**
forum.micropython.org
store.micropython.org