

MicroPython – Python for Microcontrollers

Christine Spindler

C35C3

Saturday, 29. December 2018























Clarke



Outline

- (1) What is MicroPython**
- (2) Benefits of Scripting languages**
- (3) Maker Projects**
- (4) How MicroPython generates benefit**
- (5) Hardware & Software interaction**
- (6) Questions**

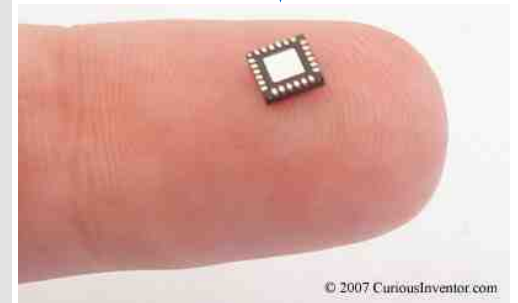
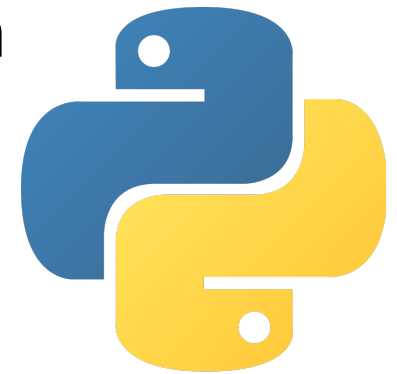
Top Programming Languages 2018

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

@IEEESpectrum 31.07.18

What is MicroPython?

- **lean and efficient** rewrite of Python
- Includes complete parser, compiler, virtual machine, runtime system and garbage collector
- **byte code** or **native machine code**
- Supports **inline assembler**
- **Compilation on the chip**
- **REPL** (read, evaluate, print – loop)



How everything started

KICKSTARTER

Micro Python: Python for microcontrollers

by Damien George

Home Updates **21** Backers **1,930** Comments **309** Cambridge, United Kingdom Hardware

Post update Last: 12/12/2013
Edit project
Check dashboard
View backer report
View messages
Send survey

Kickstarter School
Contact us
Creator FAQ
Log out

1,930 backers
£97,749 pledged of £15,000 goal
1 second to go

Back This Project
£1 minimum pledge

This project will be funded on Friday Dec 13, 10:09am GMT.

Share **1,898** Tweet Embed

The Python language made lean and fast to run on microcontrollers. For beginners and experts. control

5 years in

- **GitHub**

7 000+ Stars, 200+ contributors, 2 000+ forks
42 Releases v1.9.4 with code coverage 99.2%

- In the UK all 11-12 year old children got a **BBC Micro:Bit**

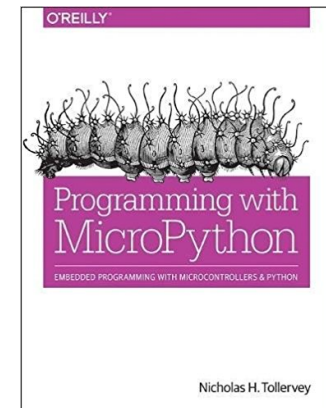
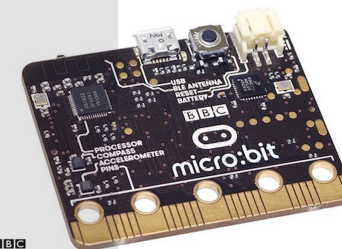
- **Development boards**

shipping with MicroPython pre-installed from different companies

Adafruit (CircuitPython), PyCom, OpenMV...

- **First** O'Reilly Book by Nicholas Tollervey

- **2nd** Generation of pyboards ready to launch



Benefits of Scripting Languages

- Initial acquaintance/**learnability**
- Rapid prototyping
- Time to market
- Easy **extensibility** by a user
- Security of extensibility by a user
- natural **sandbox**
- extension code, to maintain product integrity and protection against attack vectors.

Have you used MicroPython?

“**NO**, we use C because that’s what we do!”

“Scripting Languages are **interpreted**, so they are **slow** and use a lot of **Resources**, that’s why they are **not energy efficient**”

Well, MicroPython is **fast**!

If you look at the **DEVELOPMENT TIME**

Maker Projects



**Quadcopter
by Damien George**



**Remote, wireless
weather station
network
by Peter Hinch**

MicroPython Maker Projects



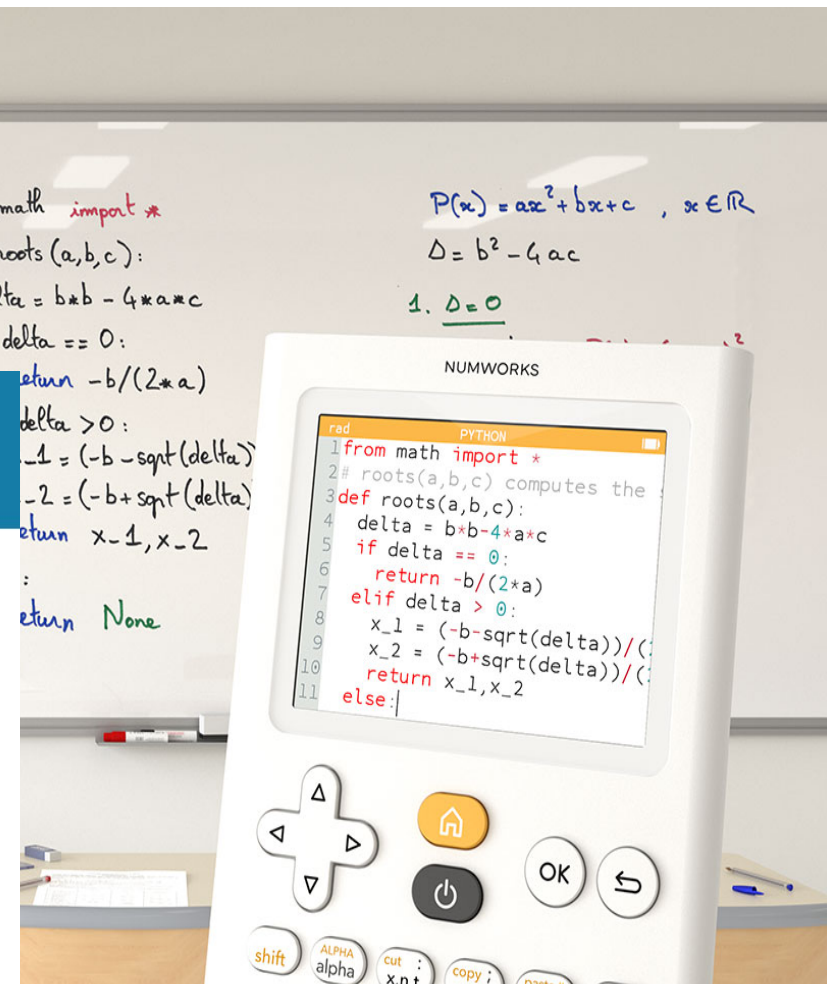
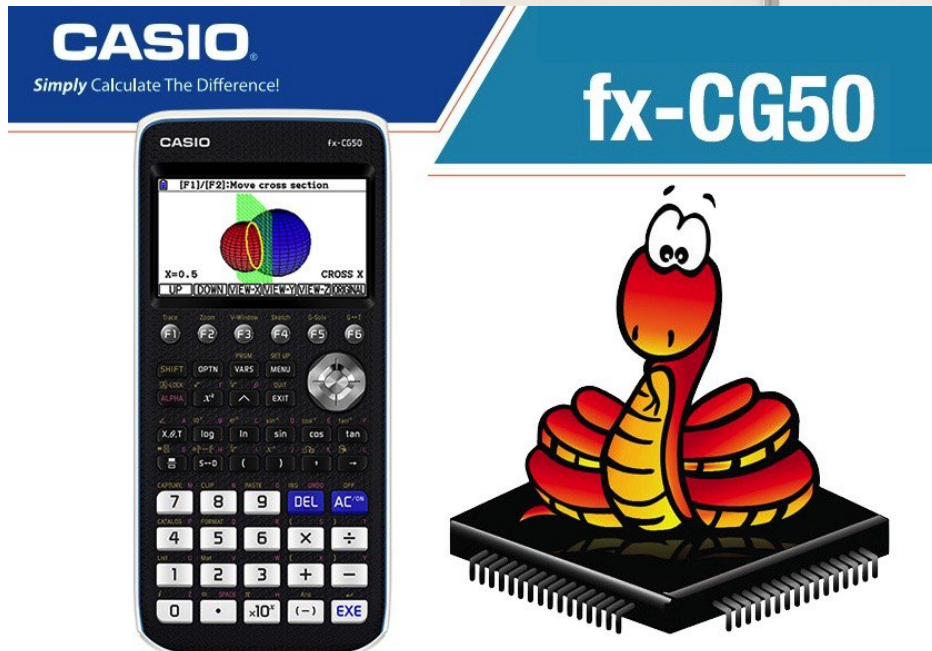
Tell us!

*“We came to **MPY** while searching for a lightweight python implementation for a linux based system **to get rid of bloated shell scripts**. For this first project it ended up with getting rid of linux replacing it with MPY.*

*Two years later we cover ultra low power systems (consuming 500nA with active REPL!) for **real-time image processing**, all with just **one development and runtime environment**.”*

“All implementations are a combination of a MPY framework and few (usually just one) specific (usually (very) small) C or even assembler modules”

MicroPython on calculators



<http://edu.casio.com/products/graphic/fxcg50/>

numworks.com

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

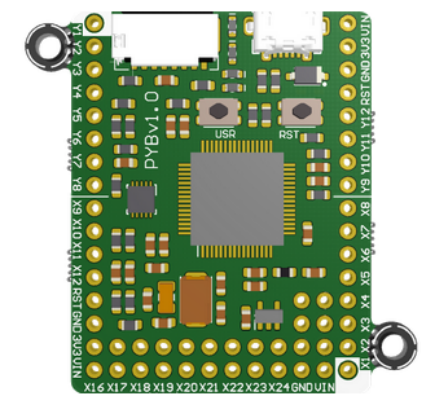
```
1 # four LEDs numbered 1 to 4
2
3 import time
4 import pyb
5
6 for i in range(1000):
7     pyb.LED((i%4) + 1).toggle()
8     time.sleep_ms(100)
9
```

Clock Speed 0.00 MHz

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

Run Script LEDs

- LED
- SERVO
- ADC



Companies using MicroPython



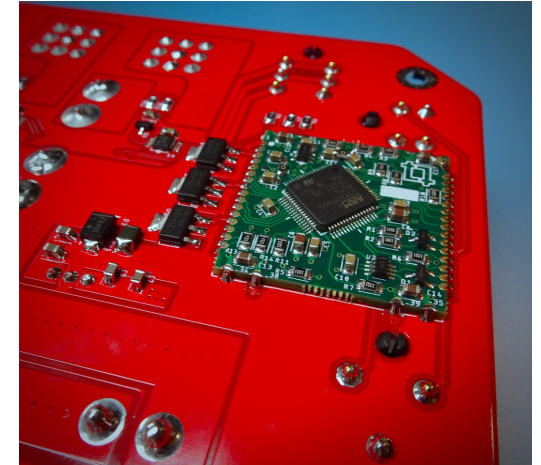
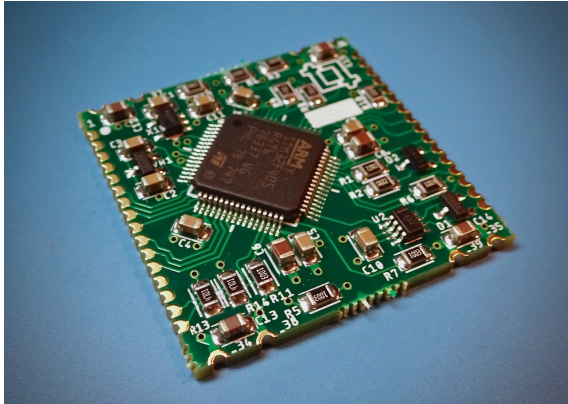
George Robotics The developers of MicroPython

“My background is theoretical physics, so I approach the design and development of MicroPython from a much more academic and research-oriented point of view, compared to simply engineering a solution to a problem.

I believe this has been part of the reason for the success of MicroPython”

— Damien P. George Creator of MicroPython & Director of George Robotics

Companies using MicroPython



TRAVIS TRAVELSTEAD
CONSULTING

"Mechanical, electrical and software design and development"

"The constant battle of finding components and tools that have ease of use, while also being capable for professional applications, is what drew me to Micropython. It allows me to design, build and iterate efficiently"



So it's the amazing software?

pyboard D!



MicroPython – Python for Microcontrollers

DEMO



Pyboard D

	PYBv1.1 (168 MHz) PYBD (120 MHz)	216 MHz (1.75 x PYBv1.1)
PYBv1.1	Idle at 18 mA Run at 55 mA	—
PYBD	Idle at 18 mA Run at 55 mA	Idle: 34 mA Run: 112 mA Light-sleep: 500 uA Deep-sleep with RTC: 10 uA

Downloading data to **PYBD 100 mA** around 800 Kbyte/sec

Uploading data out of **PYBD 140 mA** 1 Mbyte/sec

Listening HTTP server connected to WiFi router ~1mA

What MicroPython can't do

- really **small MCU's** use traditional C
- dynamically typed language
- **memory** fragmentation
- embedded Linux system for large projects

MicroPython for product development!

PRO

- productivity
- traceability
- testability
- portability
- licensing
- support

CON

- increased hardware resources
- lack of developer skills regarding scripting languages

micropython.org
forum.micropython.org
store.micropython.org
christine@micropython.org



Thanks to and credit for pictures:

- Damien George
- Viktoriya Skoryk
- Travis Travelstead
- Nicholas Tollervey
- Peter Hinch
- The Python Software Foundation
- GitHub
- Casio, Numworks
- The BBC Micro:Bit Foundation