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

<table>
<thead>
<tr>
<th>Language Rank</th>
<th>Types</th>
<th>Spectrum Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Python</td>
<td>![Icons]</td>
<td>100.0</td>
</tr>
<tr>
<td>2. C++</td>
<td>![Icons]</td>
<td>98.4</td>
</tr>
<tr>
<td>3. C</td>
<td>![Icons]</td>
<td>98.2</td>
</tr>
<tr>
<td>4. Java</td>
<td>![Icons]</td>
<td>97.5</td>
</tr>
<tr>
<td>5. C#</td>
<td>![Icons]</td>
<td>89.8</td>
</tr>
<tr>
<td>6. PHP</td>
<td>![Icons]</td>
<td>85.4</td>
</tr>
<tr>
<td>7. R</td>
<td>![Icons]</td>
<td>83.3</td>
</tr>
<tr>
<td>8. JavaScript</td>
<td>![Icons]</td>
<td>82.8</td>
</tr>
<tr>
<td>9. Go</td>
<td>![Icons]</td>
<td>76.7</td>
</tr>
<tr>
<td>10. Assembly</td>
<td>![Icons]</td>
<td>74.5</td>
</tr>
</tbody>
</table>

@IEEEESpectrum 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

Micro Python: Python for microcontrollers
by Damien George

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

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

Remote, wireless weather station network by Peter Hinch

Quadrocopter by Damien George
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/graphics/fxgcg50/

numworks.com
```python
# 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)
```
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

"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

<table>
<thead>
<tr>
<th></th>
<th>PYBv1.1 (168 MHz)</th>
<th>216 MHz (1.75 x PYBv1.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PYBv1.1</strong></td>
<td>Idle at 18 mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run at 55 mA</td>
<td></td>
</tr>
<tr>
<td><strong>PYBD</strong></td>
<td>Idle at 18 mA</td>
<td>Idle: 34 mA</td>
</tr>
<tr>
<td></td>
<td>Run at 55 mA</td>
<td>Run: 112 mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Light-sleep: 500 uA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Deep-sleep with RTC: 10 uA</td>
</tr>
</tbody>
</table>

### 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
Thanks to and credit for pictures:

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