

Embedded System Design Techniques™

Rapid Prototyping Embedded Systems using MicroPython

Session 1: Introduction to MicroPython

May 2nd, 2016

Jacob Beningo, CSDP

Course Overview






- **Introduction to MicroPython**
- Libraries and Peripheral Control
- Rapid Prototyping
- Building and Customizing Micro Python
- Python Scripting for Testing and Debug

The Lecturer – Jacob Beningo



Jacob Beningo
Principal Consultant

Social Media / Contact

-  : jacob@beningo.com
-  : 248-719-6850
-  : Jacob_Beningo
-  : Beningo Engineering
-  : JacobBeningo

EDN : Embedded Basics

CONSULTING

- Secure Bootloaders
- Code Reviews
- Architecture Design
- Real-time Software
- Expert Firmware Analysis

EMBEDDED TRAINING



www.beningo.com

Courses Overview

CEC 2013 – 2015

Fundamentals of Embedded Software (2013)

Mastering the Software Design Cycle (2014)

Python for Embedded Systems(2014)

Software Architecture Design (2014)

Baremetal C (2015)

Mastering the ARM Cortex-M Processor (2015)

Writing Portable and Robust Firmware in C (2015)

Design Patterns and the Internet (2015)

CEC 2016

Bootloader Design for MCUs
January 2016

Rapid Prototyping w/
Micro Python
May 2016

Bootloader Design for MCUs
July 2016

Side Topics 2016

Real-Time Software using Micro Python

PROMO-PYTHON

Embedded Bytes Newsletter

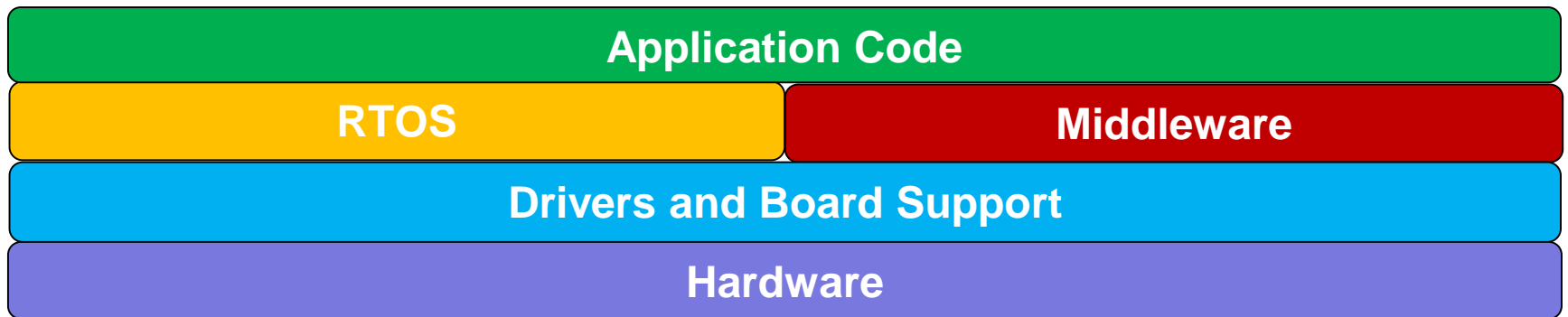
<http://bit.ly/1BAHYXm>

Session Overview

- Embedded System Design
- Python vs MicroPython
- Python in Embedded Systems
- Introduction to Python
- PyBoard Overview
- Writing a first script



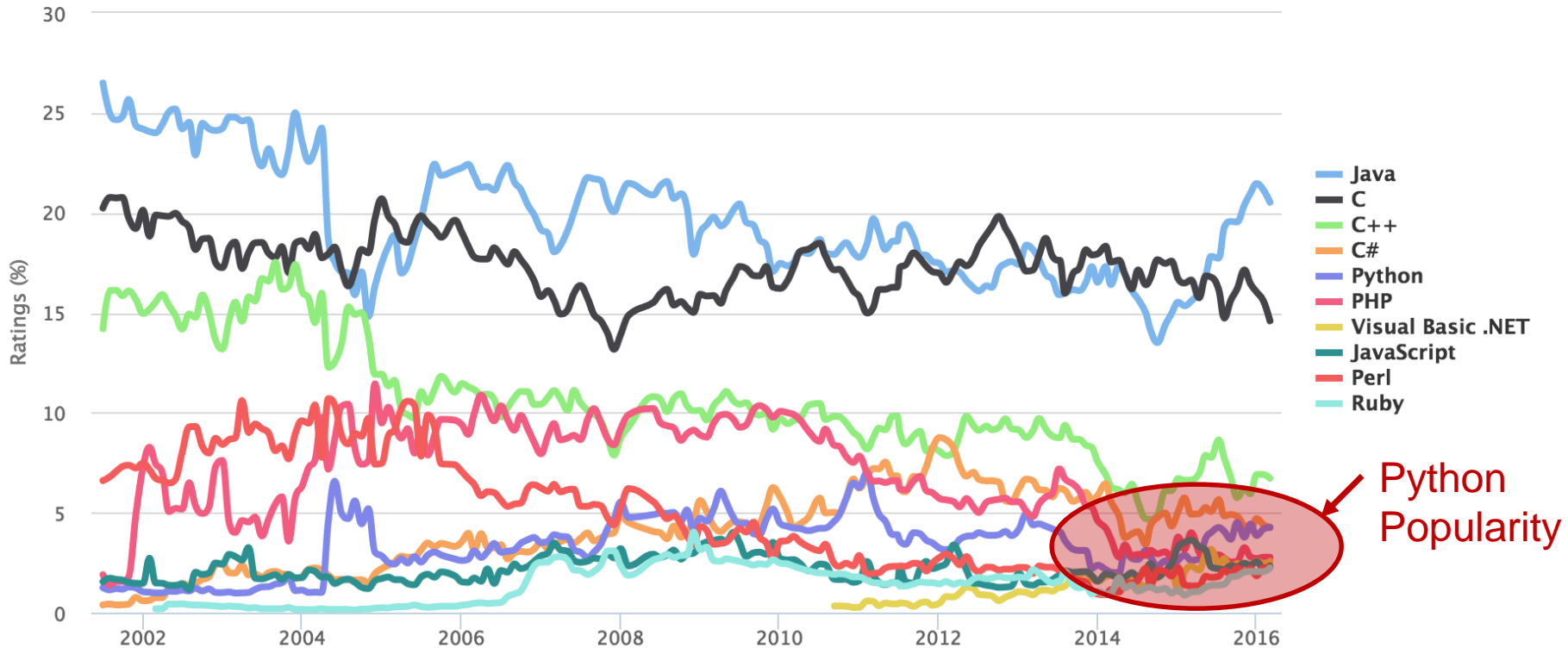
Embedded System Design



Survey of Programming Languages

TIOBE Programming Community Index

Source: www.tiobe.com



Python

- What is Python?
 - Interpreted language
 - Interactive
 - Object Oriented
- Why use Python?
 - Entry level language that is easy to learn
 - Is portable
 - Supports a large set of libraries



MicroPython



3.4



<http://docs.micropython.org/en/latest/pyboard/>

Uses of Python in Embedded Systems



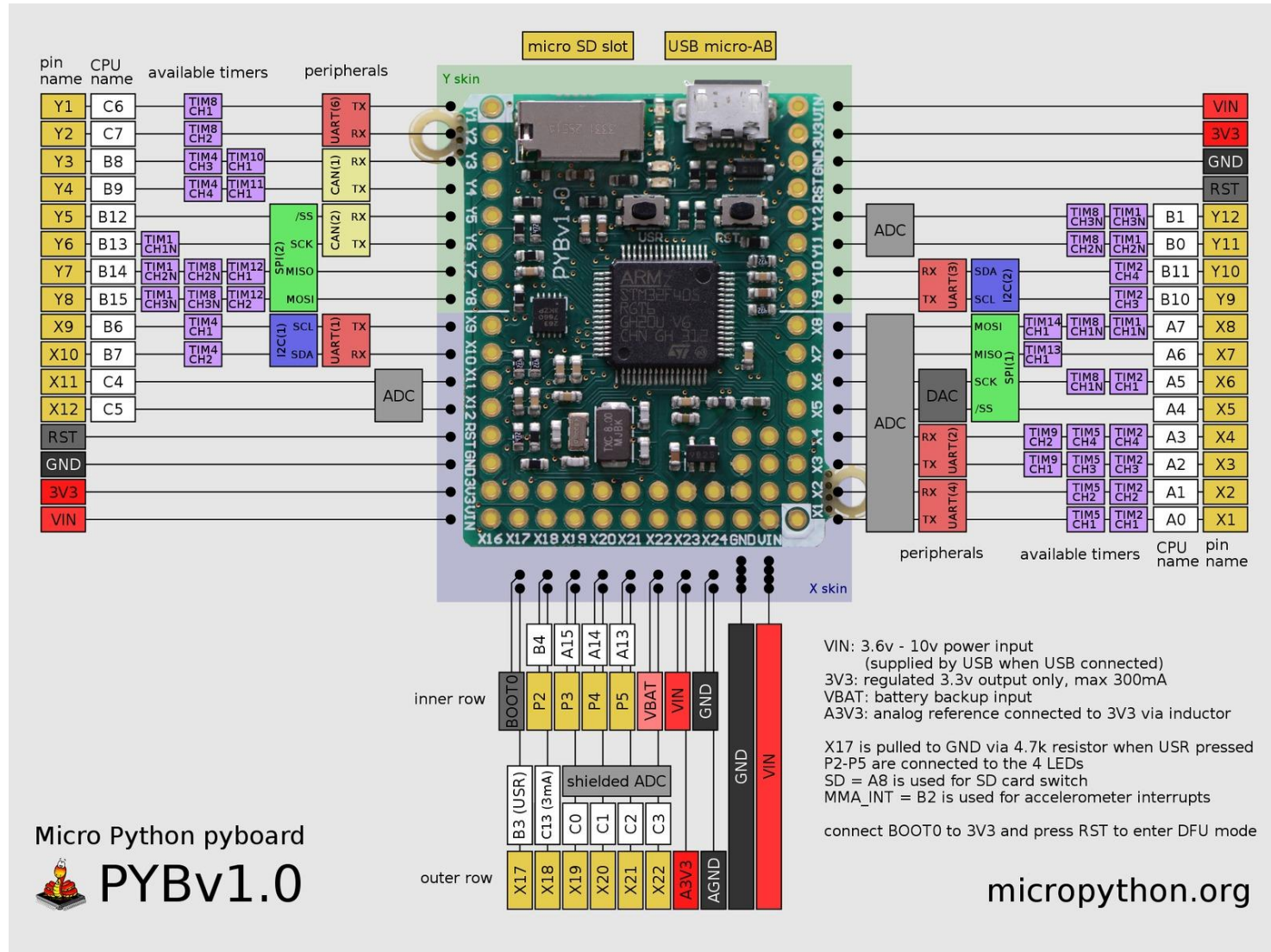
Python Keywords vs C

Python Reserved Words	C Reserved Words
and - as - assert - break -	auto - break - case - char - const -
class - continue - def - del -	continue - default - do - double -
elif - else - except -	else - enum - extern - float - for
finally - for - from - global -	goto - if - int - long - register -
if - import - in - is - lambda -	return - short - signed - sizeof -
Not - or - pass - print - raise -	static - switch - typedef - union -
return - try while - with - yield	unsigned - void - volatile - while

Python Data Types

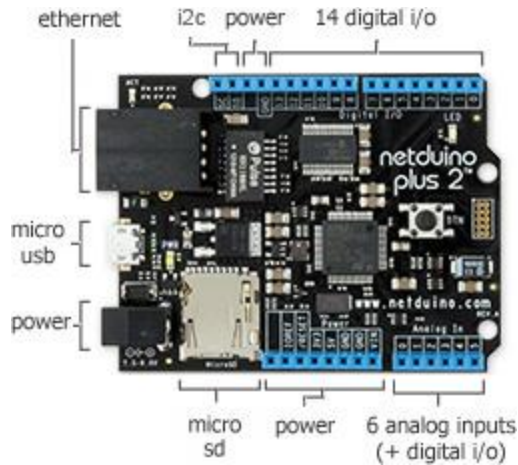
- Numeric
- String
- Lists
- Tuples
- Dictionaries
- Numeric
 - Int (14)
 - Long (134563)
 - Float (14.34)
 - Complex numbers (14.3 + 0.35j)
- String
 - Myname = 'Jacob Beningo'
 - S = Myname[0] # s = 'J'
 - S = Myname[0:4] # s = 'Jacob'

PyBoard Overview

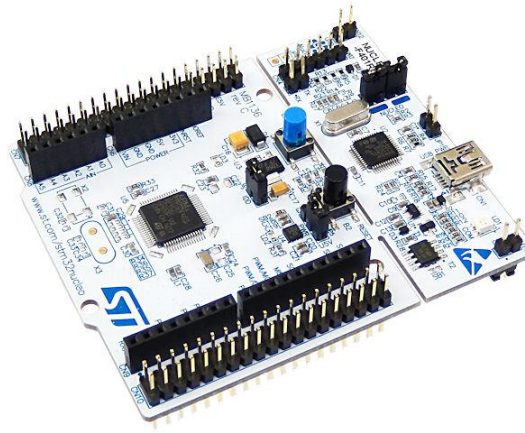


Alternative Hardware

Netduino plus 2



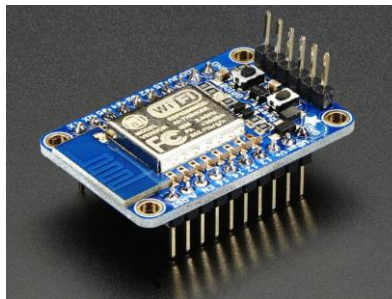
Nucleo Board



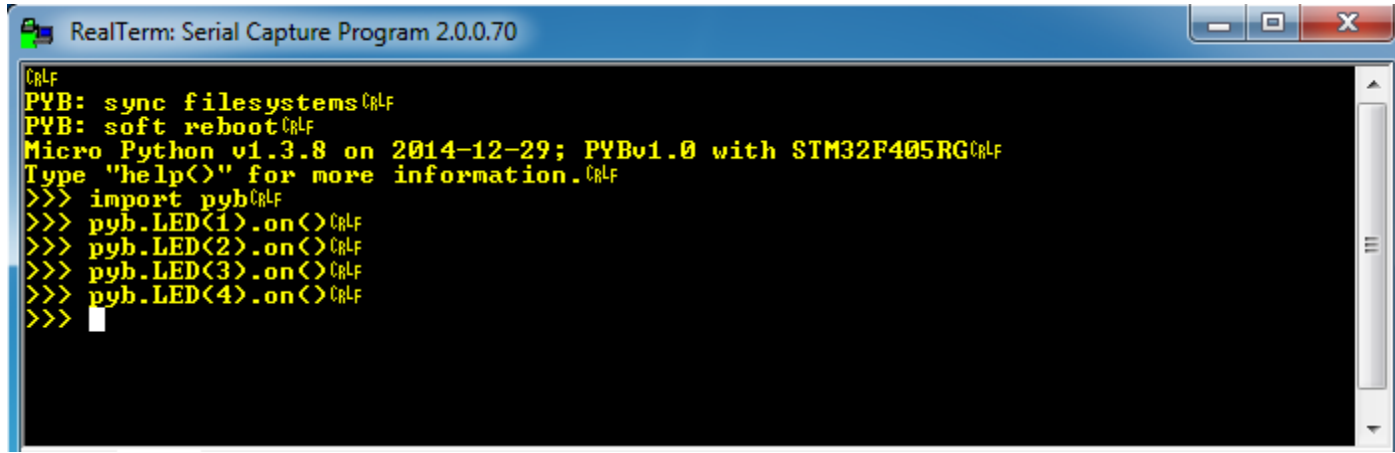
Discovery Board



HUZZAH ESP8266



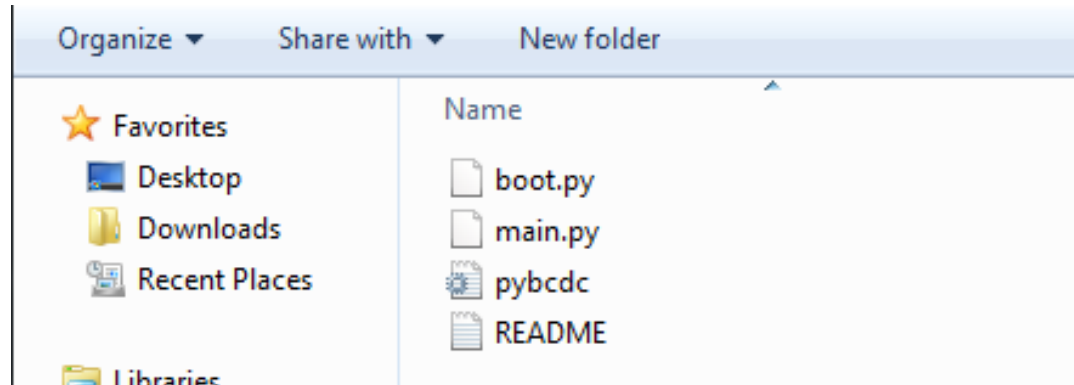
The REPL Interface



```
RealTerm: Serial Capture Program 2.0.0.70
CRLF
PYB: sync filesystems CRLF
PYB: soft reboot CRLF
Micro Python v1.3.8 on 2014-12-29; PYBv1.0 with STM32F405RG CRLF
Type 'help()' for more information. CRLF
>>> import pyb CRLF
>>> pyb.LED(1).on() CRLF
>>> pyb.LED(2).on() CRLF
>>> pyb.LED(3).on() CRLF
>>> pyb.LED(4).on() CRLF
>>>
```

Controls	Function
CTRL-A	Enter raw REPL mode
CTRL-B	Enter normal REPL mode
CTRL-C	Interrupt a running program
CTRL-D	Soft reset
help()	Displays information on pyb library

Scripting Mode



- boot.py – defines scripts to run on startup
- main.py – start of python scripting program
- pybcdc – windows serial driver
- README – misc board information

Writing a first script

Libraries and external classes



```
# main.py  
import pyb
```

Definitions and Initialization



```
# define LED color constants  
LED_RED = 1  
LED_GREEN = 2  
LED_YELLOW = 3  
LED_BLUE = 4  
  
# Defines the primary loop delay  
DELAY_1000MS = 1000  
  
# Create an Led object assigned to the green LED  
Led = pyb.LED(LED_GREEN)
```

Primary program loop

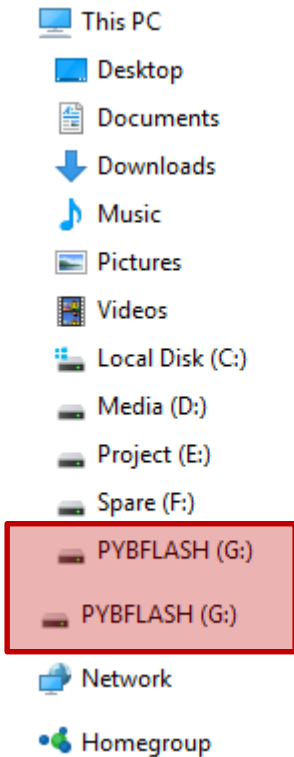


```
# Main execution loop  
# Toggle the LED every  
while True:  
    Led.toggle()  
    pyb.delay(DELAY_1000MS)
```

Loading and Running the Script

1 - Plugin

2 – Copy Scripts to drive



Name	Date modified	Type	Size
.fseventsd	4/6/2016 3:56 PM	File folder	
.Trashes	4/6/2016 8:23 AM	File folder	
._Trashes	4/6/2016 8:23 AM	TRASHES File	4 KB
boot.py	9/11/2004 2:00 AM	PY File	1 KB
main.py	9/11/2004 2:00 AM	PY File	1 KB
pybcdc.inf	9/11/2004 2:00 AM	Setup Information	3 KB
myscript.py	9/11/2004 2:00 AM	PY File	1 KB

3 – CTRL-D or Power Cycle*

```
COM5 - PuTTY
MicroPython v1.6-327-g61fa7c8 on 2016-04-04; NetduinoPlus2 with STM32F405RG
Type "help()" for more information.
>>>
PYB: sync filesystems
PYB: soft reboot
```

*May need to CTRL-C to interrupt application

Additional Resources

- Download Course Material for
 - Updated C Doxygen Templates (Sept 2015)
 - Example source code
 - Templates
- Microcontroller API Standard
- EDN Embedded Basics Articles
- Embedded Bytes Newsletter
 - <http://bit.ly/1BAHYXm>



From www.beningo.com under






- Blog > CEC Rapid Prototyping with MicroPython

The Lecturer – Jacob Beningo



Jacob Beningo
Principal Consultant

Social Media / Contact

-  : jacob@beningo.com
-  : 248-719-6850
-  : Jacob_Beningo
-  : Beningo Engineering
-  : JacobBeningo

EDN : Embedded Basics

CONSULTING

- Secure Bootloaders
- Code Reviews
- Architecture Design
- Real-time Software
- Expert Firmware Analysis

EMBEDDED TRAINING



www.beningo.com