Embedded System Design Techniques™

Rapid Prototyping Embedded Systems using MicroPython

Session 3: Rapid Prototyping

May 4th, 2016
Jacob Beningo, CSDP

Presented by:

DesignNews

CEC CONTINUING EDUCATION CENTER

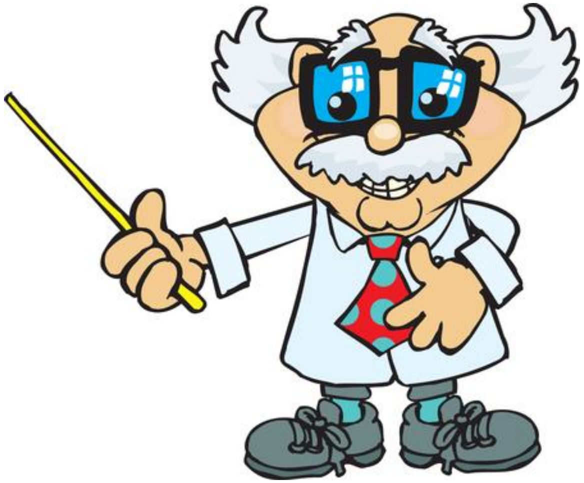
Digi-Key ELECTRONICS

Course Overview

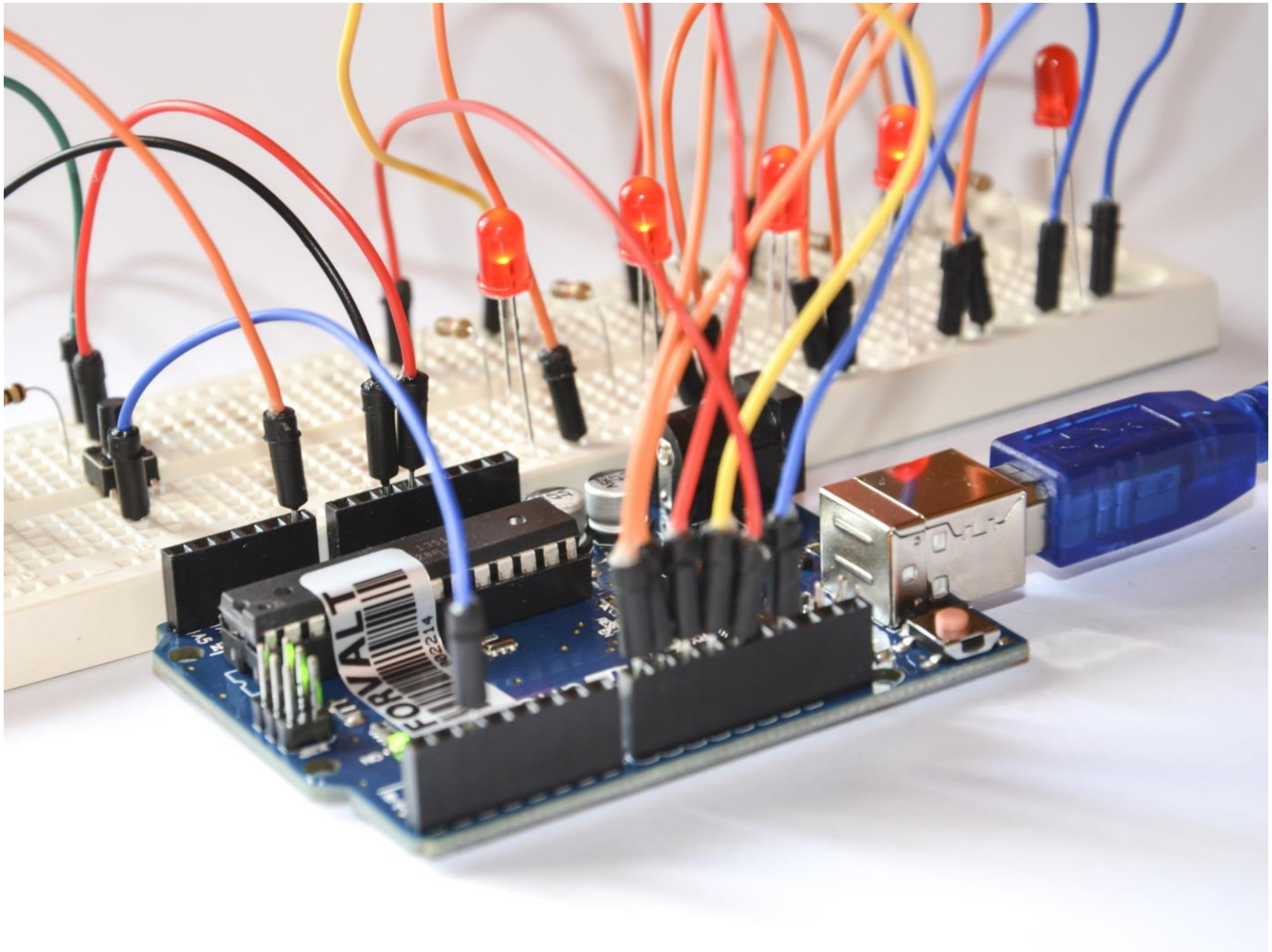
- Introduction to MicroPython
- Libraries and Peripheral Control
- **Rapid Prototyping**
- Building and Customizing Micro Python
- Python Scripting for Testing and Debug

Session Overview

- Rapid Prototyping
- Accelerometer
- SD Card
- Sensor Interfacing
- Bluetooth

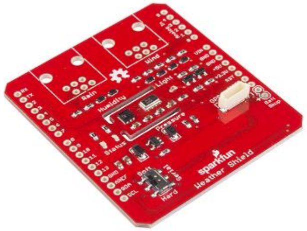


Rapid Prototyping

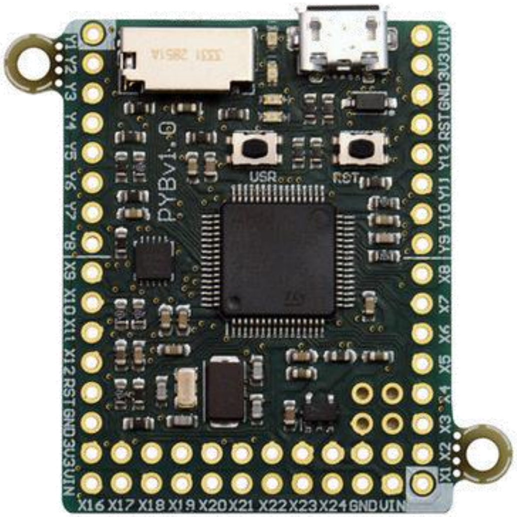


Presented by:

Rapid Prototyping - Hardware

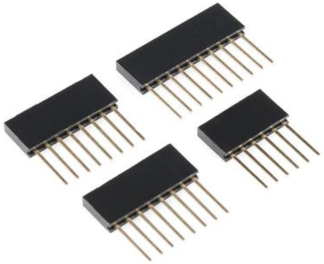


DEV-12081



PyBoard

RN42



PRT-11417

Presented by:

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

Accelerometer

Example:

```
# Accelerometer object
Accelerometer = pyb.Accel()

# Check the Blue LED state
   x = Accelerometer.x()

   if x > 15:
      pyb.LED(LED_BLUE).on()
   else:
      pyb.LED(LED_BLUE).off()

   print (”X-Axis data is ", x, "\n")
```

Create Object

Sample the x-axis

if conditional

debug info in terminal

Presented by:

SD Card

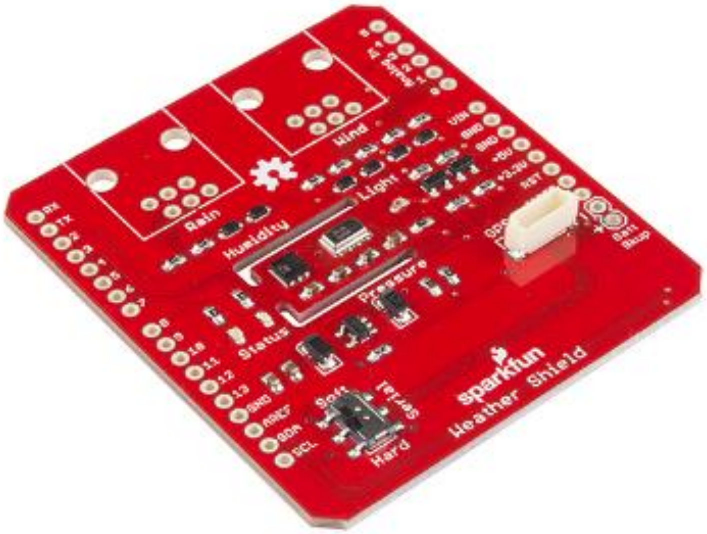
File System Control
- /Flash (max 300 kB)
- /SD (> 4 GB)

Useful Commands
- import OS
- os.mkdir
- open
- close
- os.sync

```
# Test code to write to a file
with open("Test1.txt", "a+") as f:
    f.write("Hello World!" + "\n")
    f.close()
```

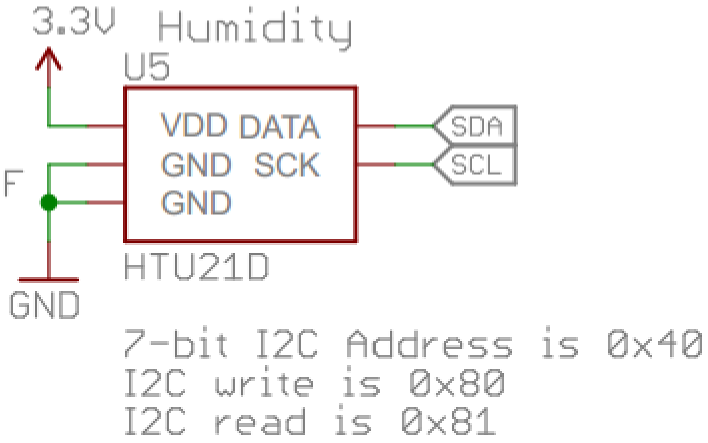
Sensor Interfacing

- DEV-12081
  - HTU21D humidity
  - MPL3115A2 barometric pressure
  - ALS-PT19 light sensors
  - Rain sensor (optional)
  - Wind sensor (optional)
  - GPS (optional)

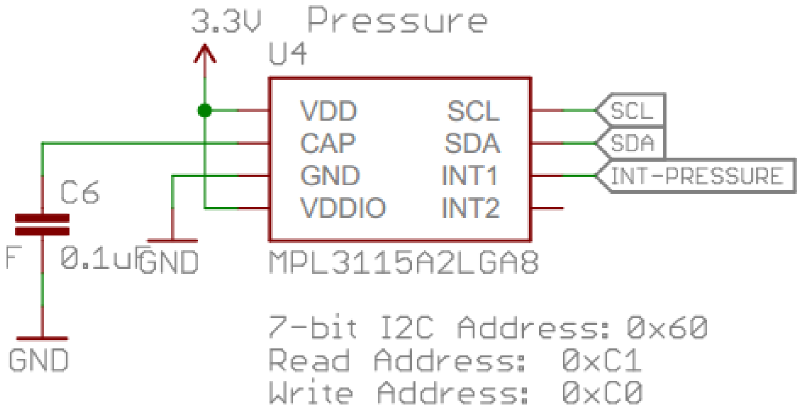


Sensor Interfacing

Digital

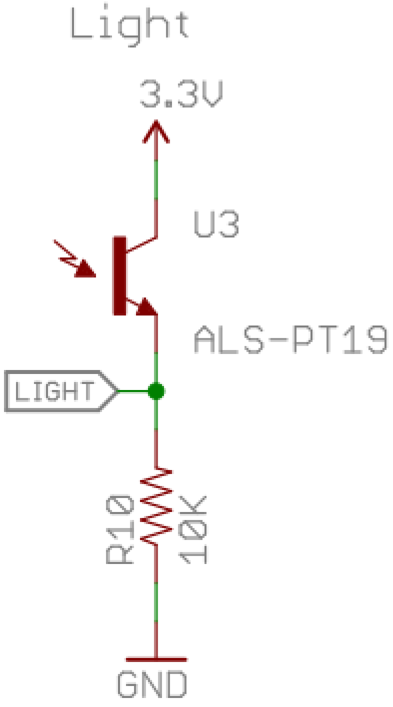


7-bit I2C Address is 0x40
I2C write is 0x80
I2C read is 0x81



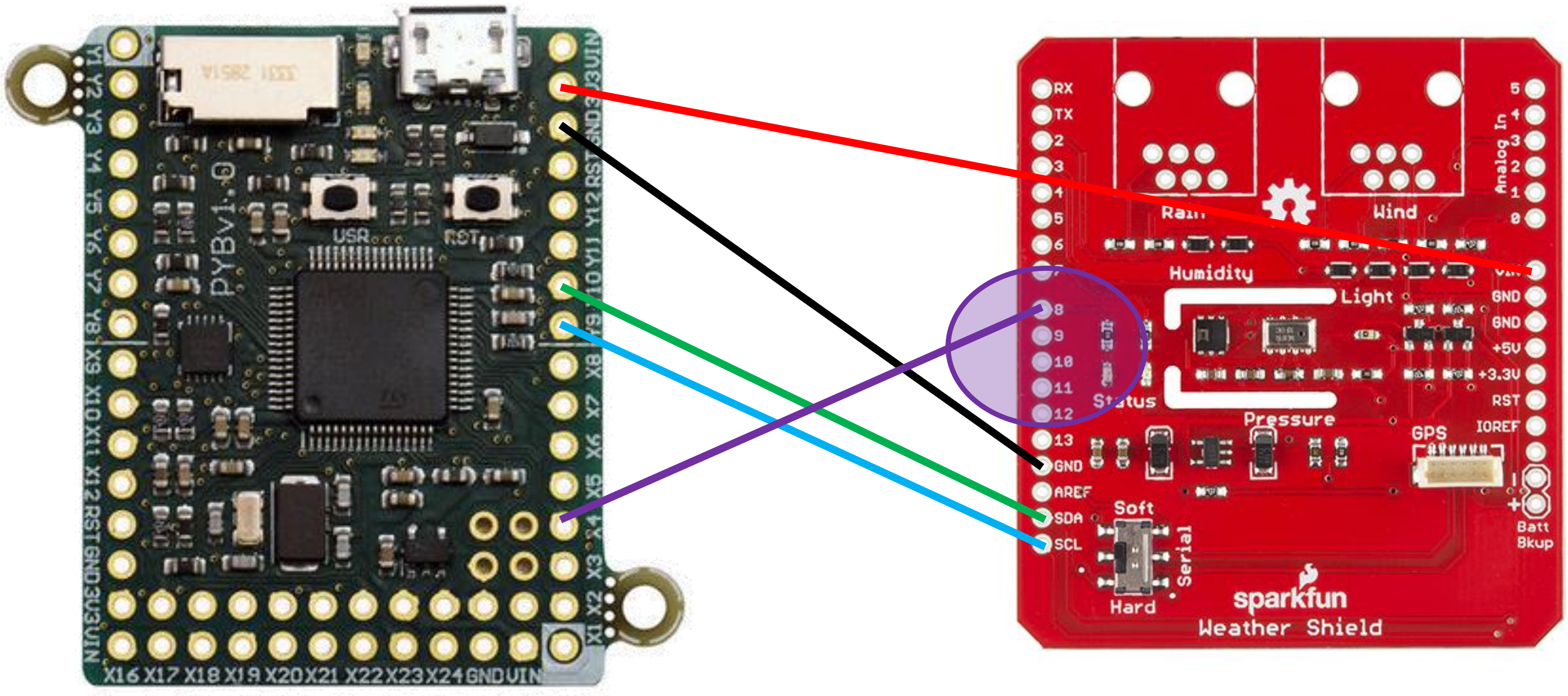
7-bit I2C Address: 0x60
Read Address: 0xC1
Write Address: 0xC0

Analog



Presented by:

Sensor Interfacing



Presented by:

Sensor Interfacing - Initialization

```
from pyb import I2C
```

```
GlobalTemp = 0.0
GlobalHumidity = 0.0

X4_State = 0
```

Variable Initializations

```
# initialize I2C peripheral 2
I2C2 = I2C(2,I2C.MASTER, baudrate=100000)

# start the first sample conversion
I2C2.send(0xF5, 0x40)

x4 = pyb.Pin.board.X4
x4.init(pyb.Pin.OUT_PP, pyb.Pin.PULL_NONE, -1)
```

I2C initialization
GPIO X4 Init

```
while True:
        SensorSample()
        pyb.delay(1000)
```

Main loop

Sensor Interfacing - Sampling

```
def SensorSample():
    global GlobalTemp
    global GlobalHumidity

    ToggleX4()
    RxData = I2C2.recv(3, 0x40)
    test = int(RxData[1]) & 0x2

    if(test != 0x2):
        Temperature = (RxData[0]<<8) | (RxData[1])
        TempAdjusted = (175.72*Temperature/(65536))-46.85
        TempAdjusted = TempAdjusted * 1.8 + 32
        GlobalTemp = TempAdjusted

        I2C2.send(0xF5, 0x40)
    else:
        Humidity = (RxData[0]<<8) | (RxData[1])
        HumidityAdjusted = (125*Humidity/(65536))-6
        GlobalHumidity = HumidityAdjusted
        I2C2.send(0xF3, 0x40)
```

Access global data

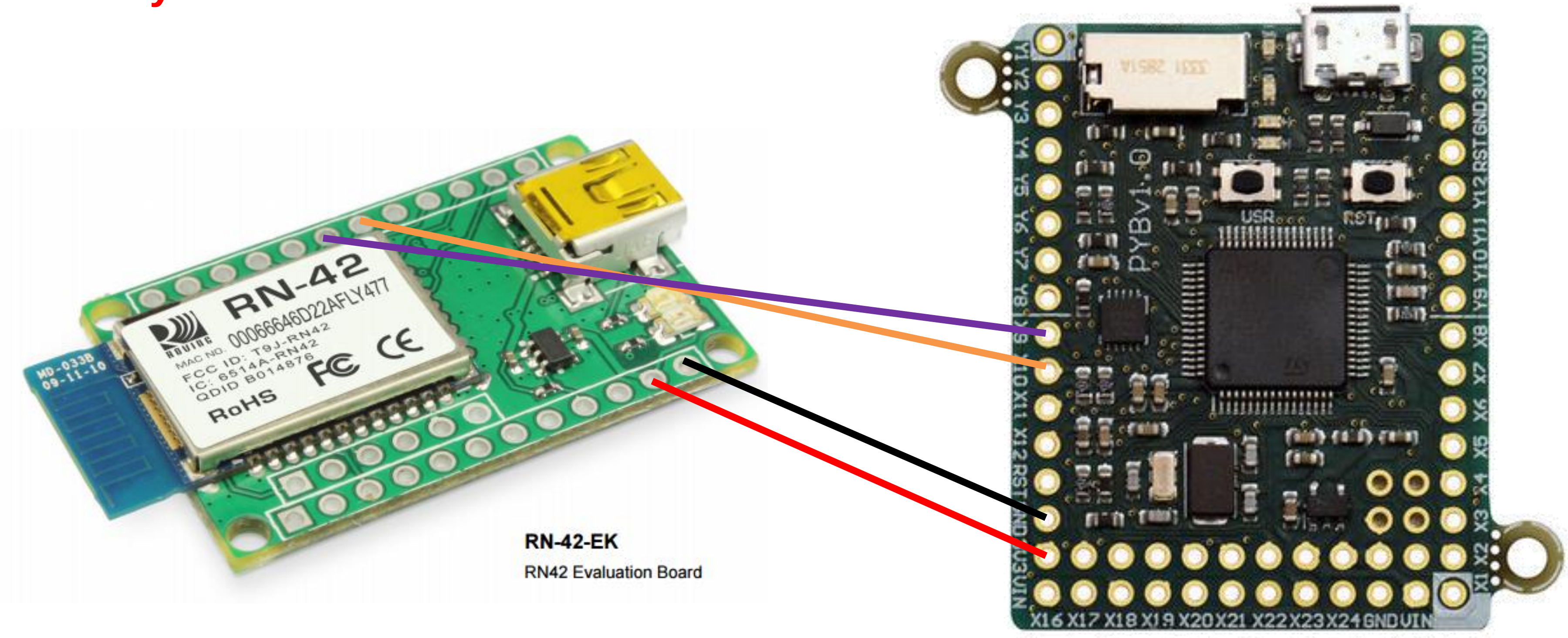
Blink LED
Receive Data

Temp
Sensor

Humidity
Sensor

Bluetooth

Verify connection from datasheet!


RN-42
RoHS
PYBv1.0
USR
RST
Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10 X11 X12 RST GND 3V3 VIN
VIN 3V3 GND RST Y12 Y11 Y10 Y9 X8 X7 X6 X5 X4 X3 X2 X1
X16 X17 X18 X19 X20 X21 X22 X23 X24 GND VIN

RN-42-EK
RN42 Evaluation Board

Presented by:

Bluetooth Initialization

Bluetooth Module

- Need to pair with mobile device or PC
  - Putty, BlueSerialTerminal, etc
- Defaults
  - Bluetooth Slave mode
  - Serial 115200,8 bits, no parity, 1 stop bit
  - No flow control
  - Low power mode off

Detailed pairing instructions at
http://ww1.microchip.com/downloads/en/DeviceDoc/50002325A.pdf

```
# Configure Uart1 for communication
Uart1 = pyb.UART(1,115200)
Uart1.init(115200, bits=8, parity=None, stop=1)
```

Uart Configuration

Bluetooth Sensor Data Tx

```
def Uart1Rx():

    global GlobalTemp
    global GlobalHumidity

    if Uart1.any():
        temp = Uart1.readchar()
        print (chr(temp))
```

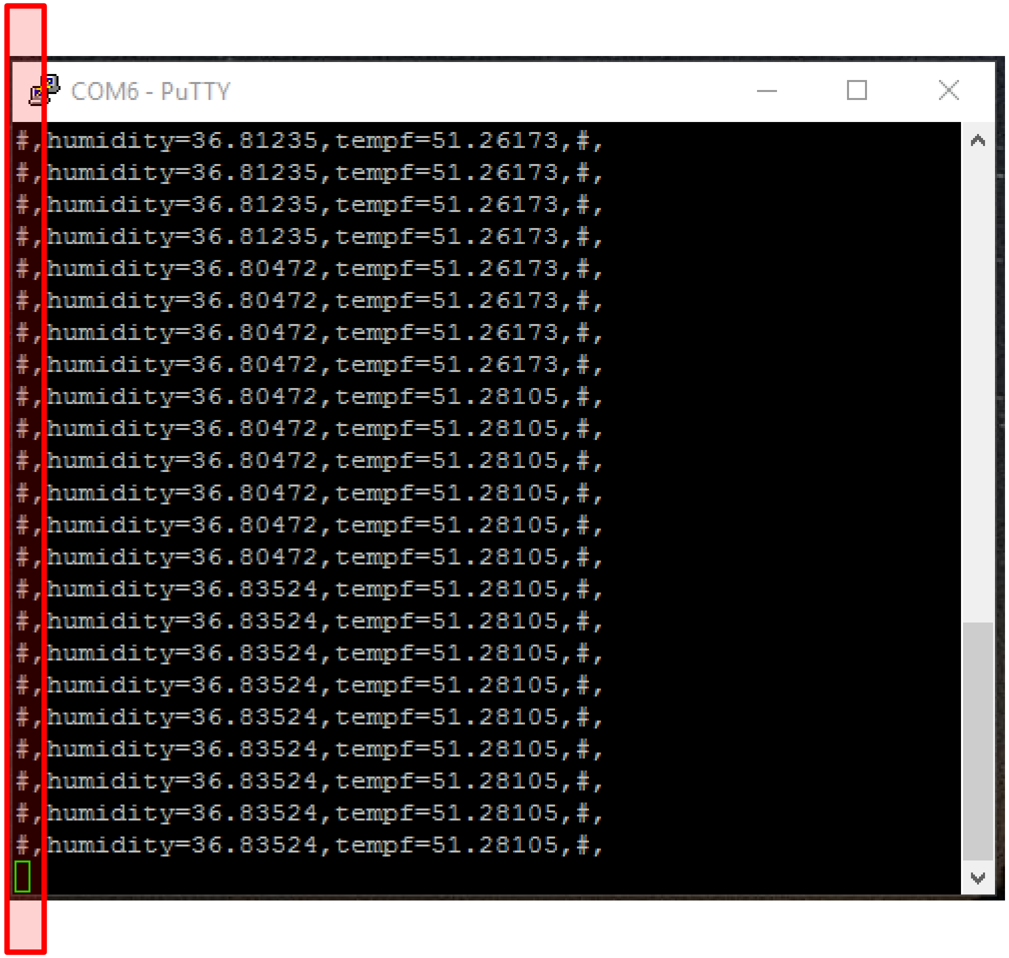
Read character data

```
        if temp == ord('#'):
            print("Transmitted")
            path = "#,humidity="+str(GlobalHumidity)+",tempf="+str(GlobalTemp-
                        )+",#,\n\r"
            Uart1.write(path)
```

Transmit string over bluetooth

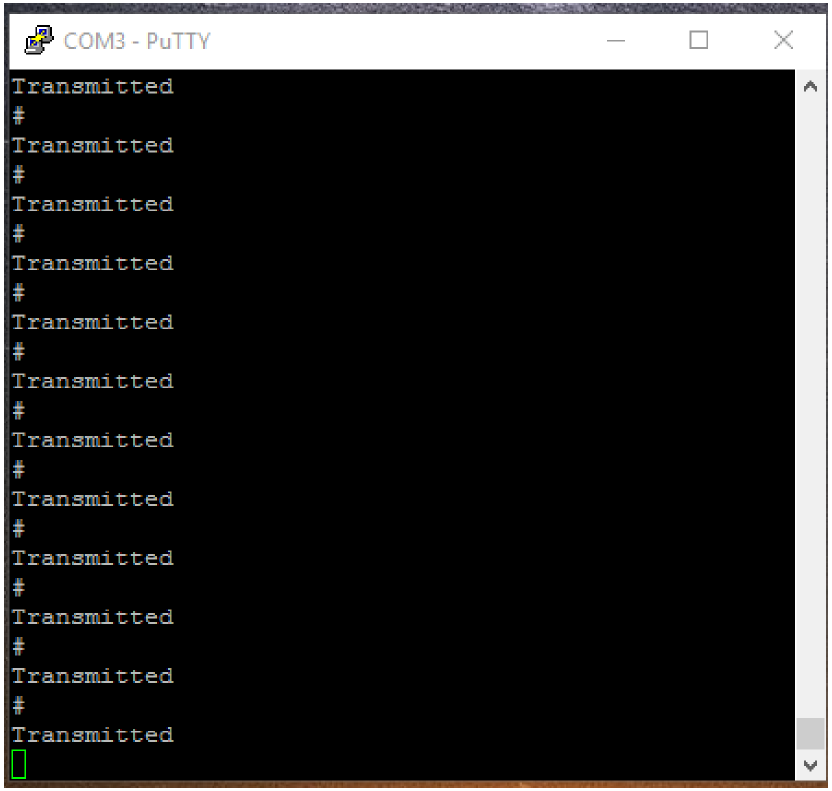
Results

Bluetooth Master



```
#,humidity=36.81235,tempf=51.26173,#,
#,humidity=36.81235,tempf=51.26173,#,
#,humidity=36.81235,tempf=51.26173,#,
#,humidity=36.81235,tempf=51.26173,#,
#,humidity=36.80472,tempf=51.26173,#,
#,humidity=36.80472,tempf=51.26173,#,
#,humidity=36.80472,tempf=51.26173,#,
#,humidity=36.80472,tempf=51.26173,#,
#,humidity=36.80472,tempf=51.28105,#,
#,humidity=36.80472,tempf=51.28105,#,
#,humidity=36.80472,tempf=51.28105,#,
#,humidity=36.80472,tempf=51.28105,#,
#,humidity=36.80472,tempf=51.28105,#,
#,humidity=36.80472,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
#,humidity=36.83524,tempf=51.28105,#,
```

MicroPython Terminal



```
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
#
Transmitted
```

Results

Rapid Prototyping of Bluetooth and two external sensors

- 15 minutes to wire everything up
- 30 minutes to write Python code (and get coffee)
- 15 minutes of debugging (loose serial wire)

Total Time = 60 minutes!

Where to go from here?

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

E : jacob@beningo.com
T : 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

DOULOS CERTIFIED TRAINING PARTNER

BENINGO EMBEDDED GROUP

www.beningo.com

Presented by: