Embedded System Design Techniques™

Designing IoT Sensor Nodes using the ESP8266

Session 3: Interfacing Sensors to the ESP8266

July 12th, 2017

Jacob Beningo

Presented by:

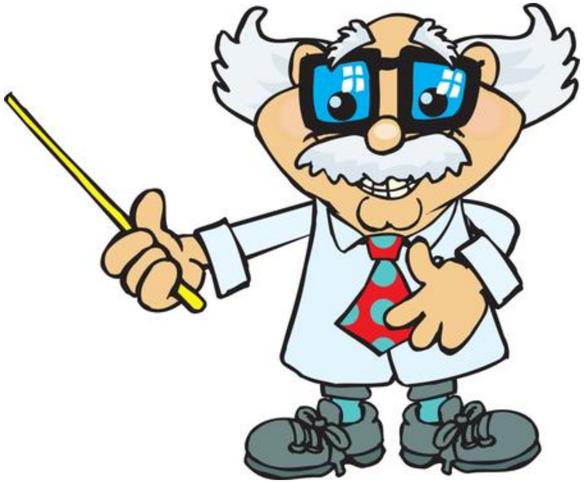
Course Overview

Topics:

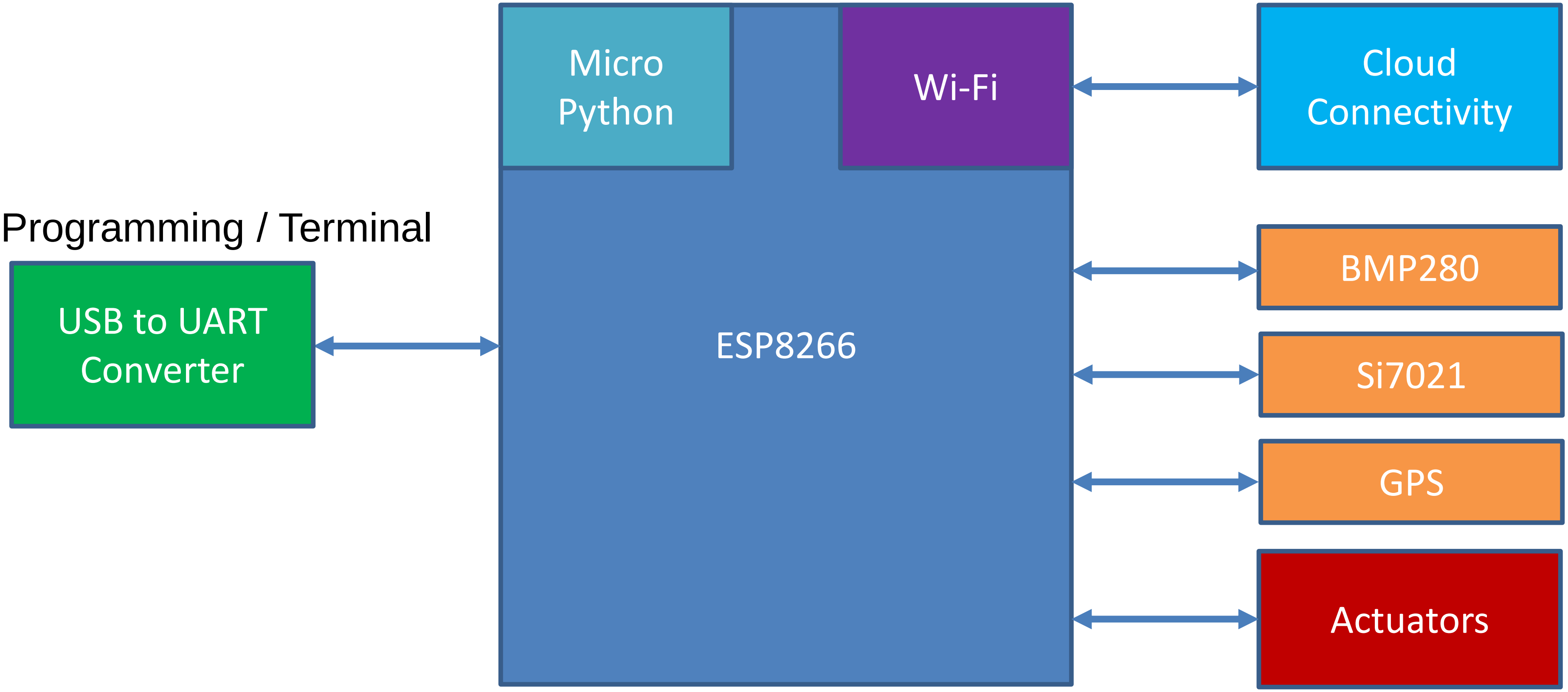
- The IoT Architecture
- Getting Started with the ESP8266
- **Interfacing Sensors to the ESP8266**
- Connecting the ESP8266 to the internet
- Device Management and the Automated Universe

Session Overview

- The IoT Sensor Node
- BMP280 Barometric Pressure Sensor
- Si7021 Temperature and Humidity Sensor
- DHT(Digital Humidity & Temperature)
- Analog Conversion

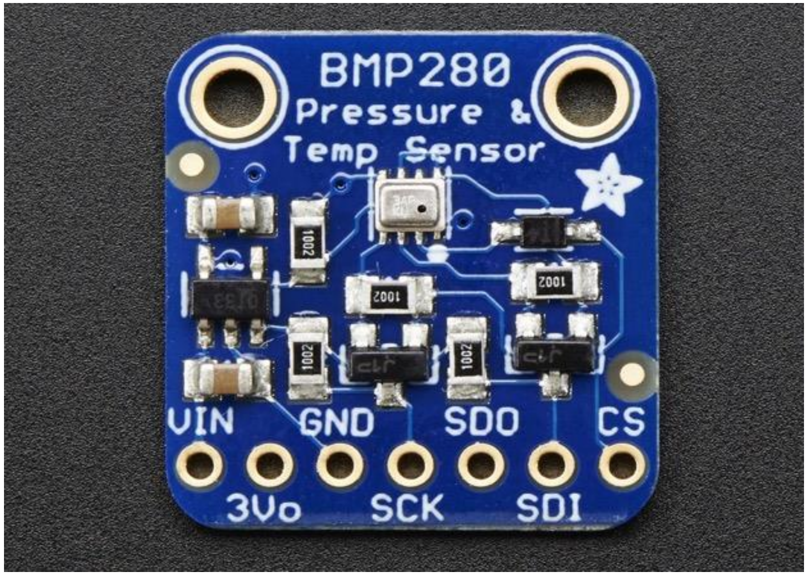


The IoT Sensor Node

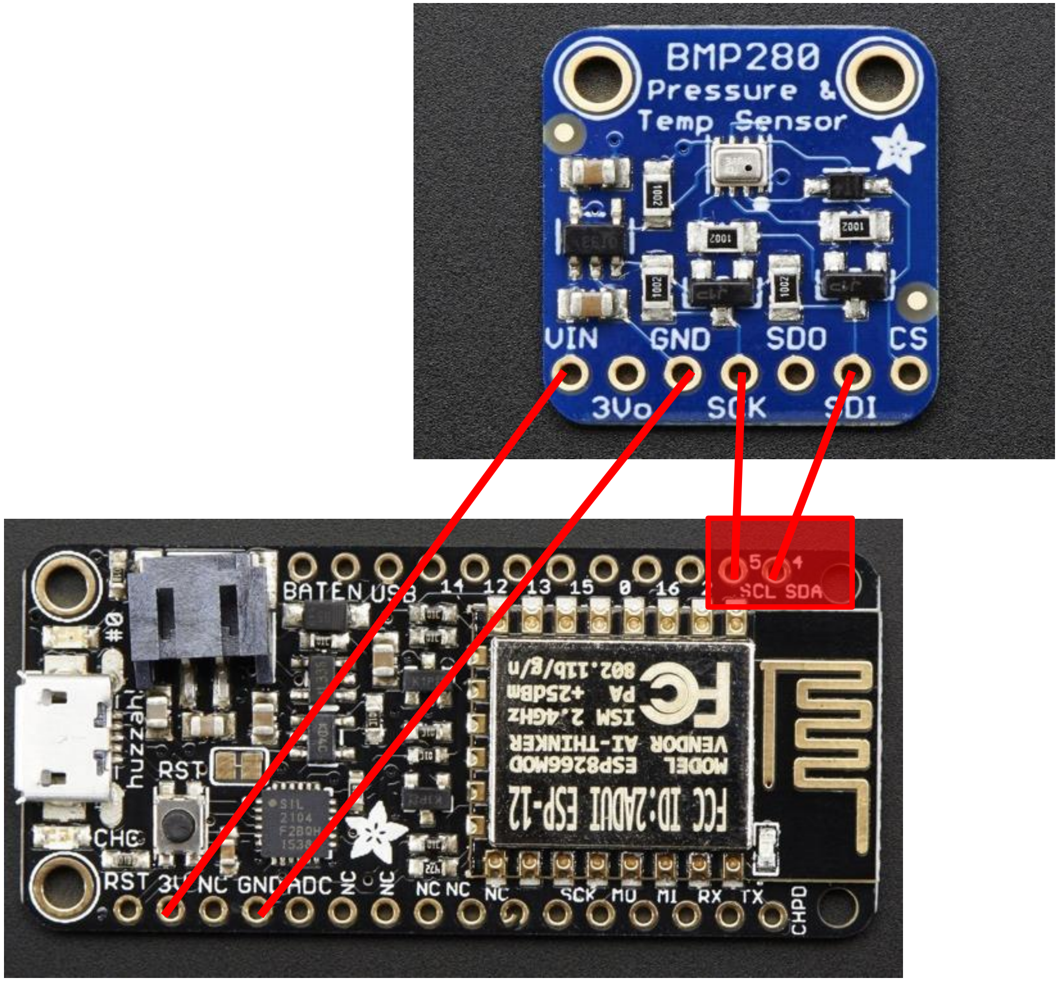


The BMP280

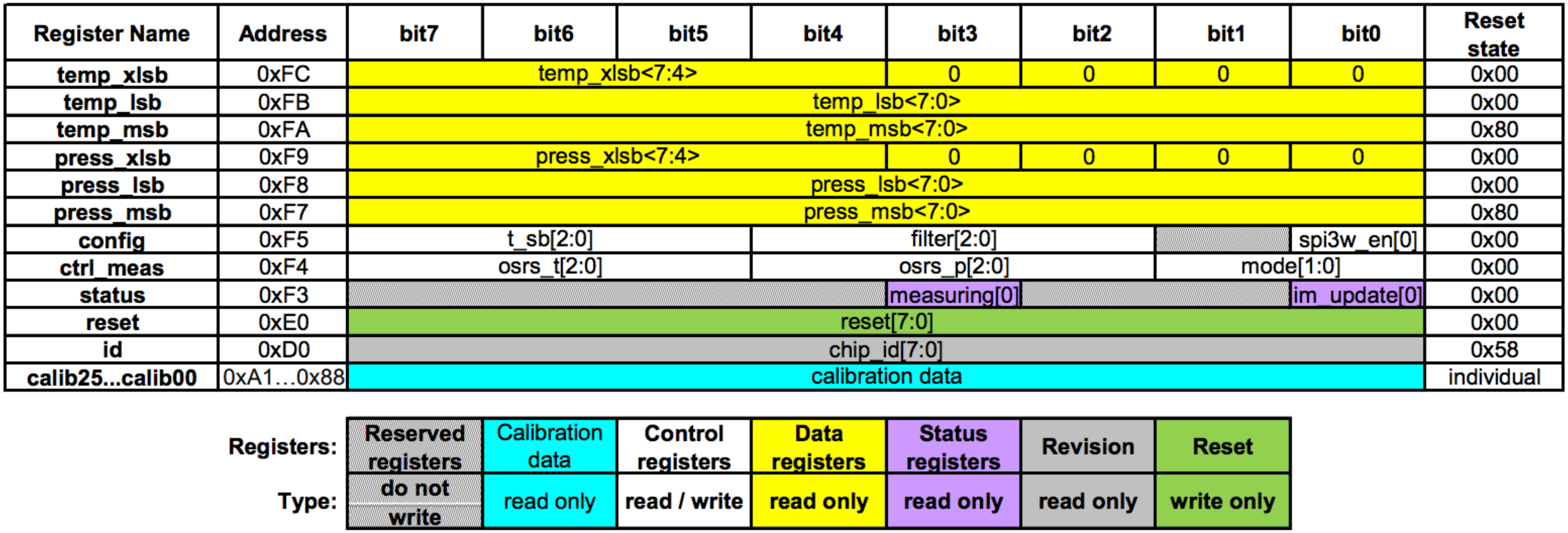
- Environmental sensor containing
  - Temperature Sensing
  - Barometric Pressure Sensor



The BMP280


BMP280
Pressure &
Temp Sensor
VIN
GND
SDO
CS
3Vo
SCK
SDI
BAT
EN
USB
14
12
13
15
0
16
5
4
SCL
SDA
#0
huzzah
RST
CHG
RST
3V
NC
GND
ADC
NC
NC
SCK
MO
MI
RX
TX
CHPD
802.11b/g/n
PA +25dBm
ISM 2.4GHz
VENDOR AI-THINKER
MODEL ESP8266MOD
FCC ID:2ADUIESP-12

The BMP280

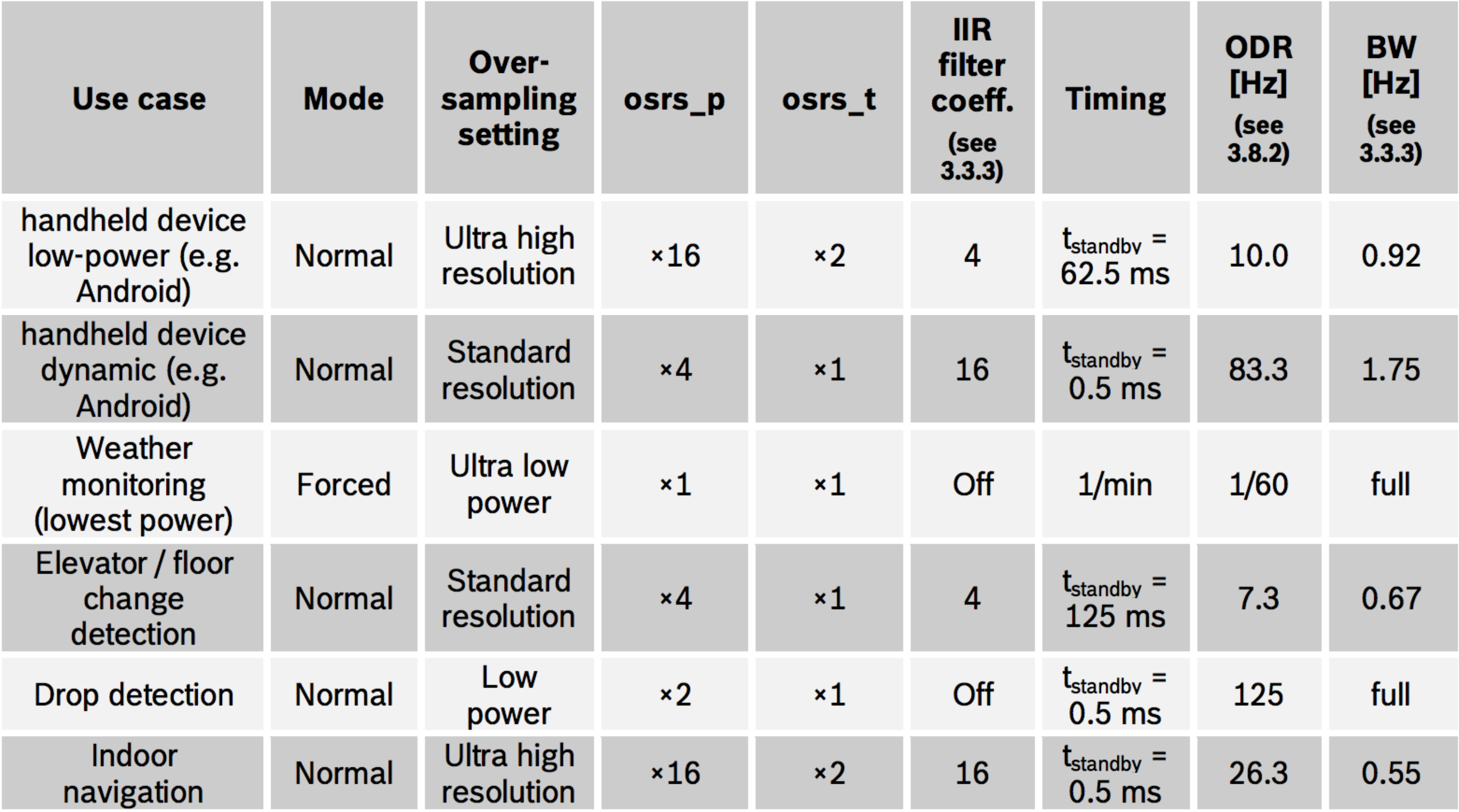


Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00
temp_msb	0xFA	temp_msb<7:0>								0x80
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00
press_lsb	0xF8	press_lsb<7:0>								0x00
press_msb	0xF7	press_msb<7:0>								0x80
config	0xF5	t_sb[2:0]			filter[2:0]			▨	spi3w_en[0]	0x00
ctrl_meas	0xF4	osrs_t[2:0]			osrs_p[2:0]			mode[1:0]		0x00
status	0xF3	▨				measuring[0]	▨		im_update[0]	0x00
reset	0xE0	reset[7:0]								0x00
id	0xD0	chip_id[7:0]								0x58
calib25...calib00	0xA1…0x88	calibration data								individual

Registers:	Reserved registers	Calibration data	Control registers	Data registers	Status registers	Revision	Reset
Type:	do not write	read only	read / write	read only	read only	read only	write only

Presented by:

The BMP280



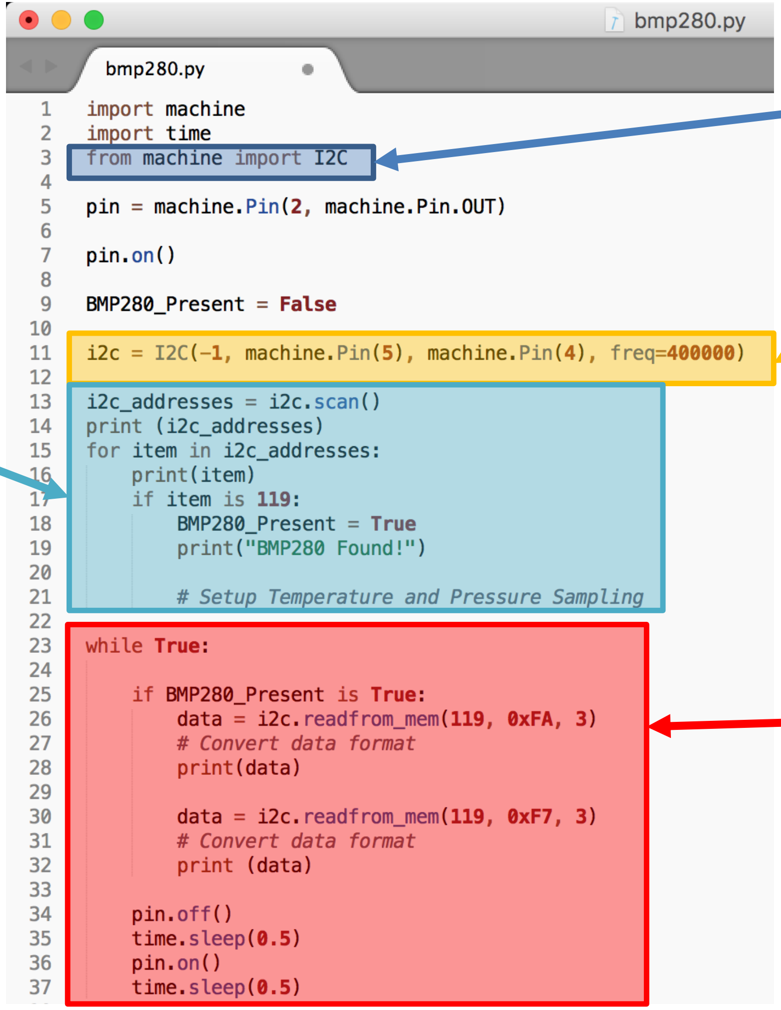
Use case	Mode	Over-sampling setting	osrs_p	osrs_t	IIR filter coeff. (see 3.3.3)	Timing	ODR [Hz] (see 3.8.2)	BW [Hz] (see 3.3.3)
handheld device low-power (e.g. Android)	Normal	Ultra high resolution	×16	×2	4	$t_{standby}$ = 62.5 ms	10.0	0.92
handheld device dynamic (e.g. Android)	Normal	Standard resolution	×4	×1	16	$t_{standby}$ = 0.5 ms	83.3	1.75
Weather monitoring (lowest power)	Forced	Ultra low power	×1	×1	Off	1/min	1/60	full
Elevator / floor change detection	Normal	Standard resolution	×4	×1	4	$t_{standby}$ = 125 ms	7.3	0.67
Drop detection	Normal	Low power	×2	×1	Off	$t_{standby}$ = 0.5 ms	125	full
Indoor navigation	Normal	Ultra high resolution	×16	×2	16	$t_{standby}$ = 0.5 ms	26.3	0.55

Micro Python I2C

```
from machine import I2C

i2c = I2C(-1, machine.Pin(5), machine.Pin(4), freq=400000)

data = i2c.readfrom_mem(119, 0xD0, 1)
print(data)

>>>b ‘X’
```

X is 0x58!

Micro Python Script Example



```python
import machine
import time
from machine import I2C

pin = machine.Pin(2, machine.Pin.OUT)

pin.on()

BMP280_Present = False

i2c = I2C(-1, machine.Pin(5), machine.Pin(4), freq=400000)

i2c_addresses = i2c.scan()
print (i2c_addresses)
for item in i2c_addresses:
    print(item)
    if item is 119:
        BMP280_Present = True
        print("BMP280 Found!")

        # Setup Temperature and Pressure Sampling

while True:

    if BMP280_Present is True:
        data = i2c.readfrom_mem(119, 0xFA, 3)
        # Convert data format
        print(data)

        data = i2c.readfrom_mem(119, 0xF7, 3)
        # Convert data format
        print (data)

    pin.off()
    time.sleep(0.5)
    pin.on()
    time.sleep(0.5)
```

Object Instantiation

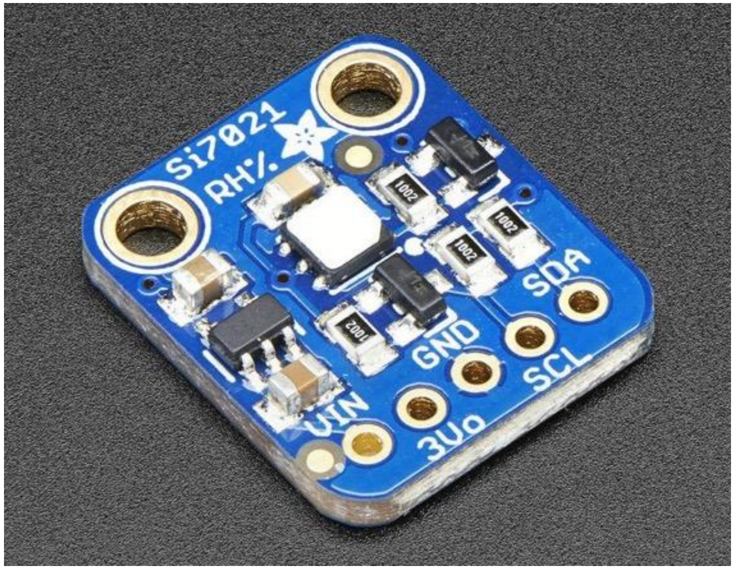
Initialization

Main script

Presented by:

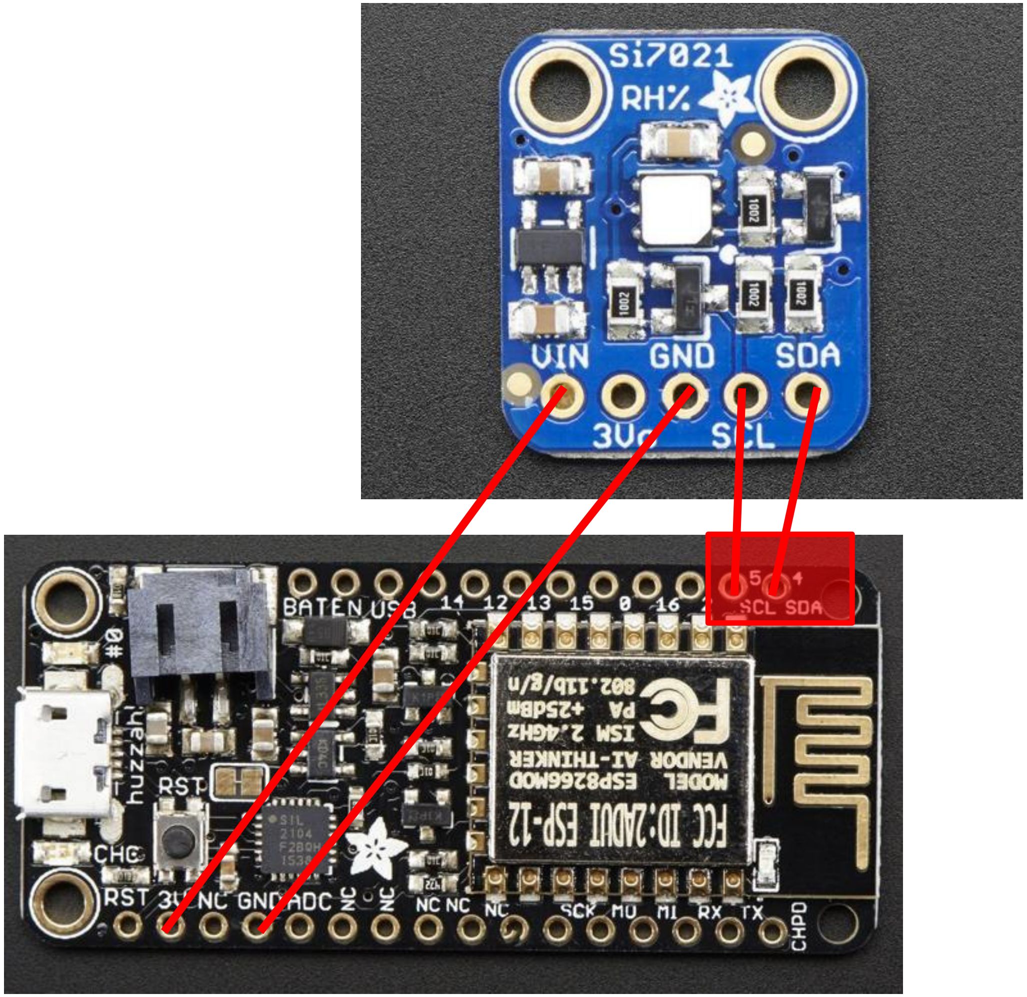
The Si7021

- Environmental sensor containing
  - Temperature Sensing
  - Humidity Sensor



Presented by:

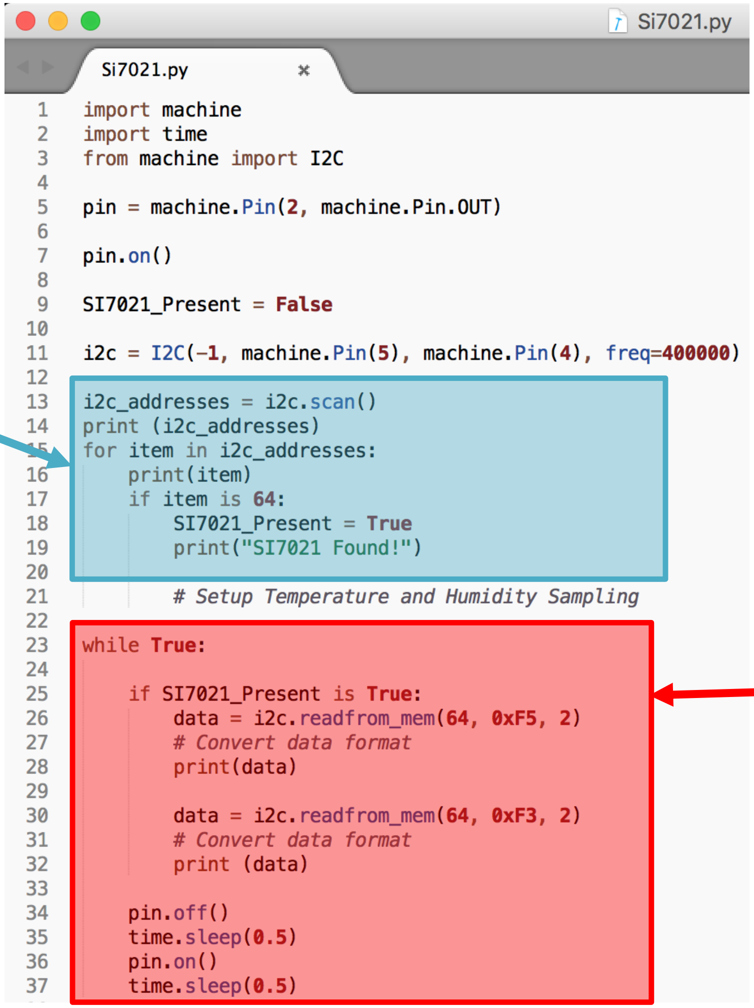
The Si7021



Presented by:

The SI7021

Initialization



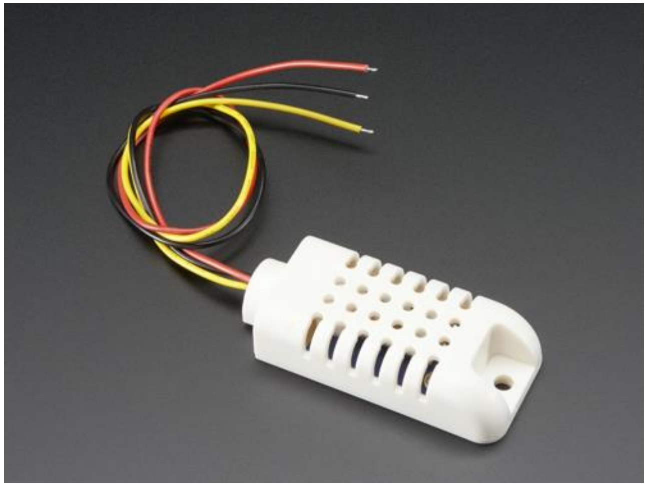
Si7021.py

```
import machine
import time
from machine import I2C

pin = machine.Pin(2, machine.Pin.OUT)

pin.on()

SI7021_Present = False

i2c = I2C(-1, machine.Pin(5), machine.Pin(4), freq=400000)

i2c_addresses = i2c.scan()
print (i2c_addresses)
for item in i2c_addresses:
    print(item)
    if item is 64:
        SI7021_Present = True
        print("SI7021 Found!")

        # Setup Temperature and Humidity Sampling

while True:

    if SI7021_Present is True:
        data = i2c.readfrom_mem(64, 0xF5, 2)
        # Convert data format
        print(data)

        data = i2c.readfrom_mem(64, 0xF3, 2)
        # Convert data format
        print (data)

    pin.off()
    time.sleep(0.5)
    pin.on()
    time.sleep(0.5)
```

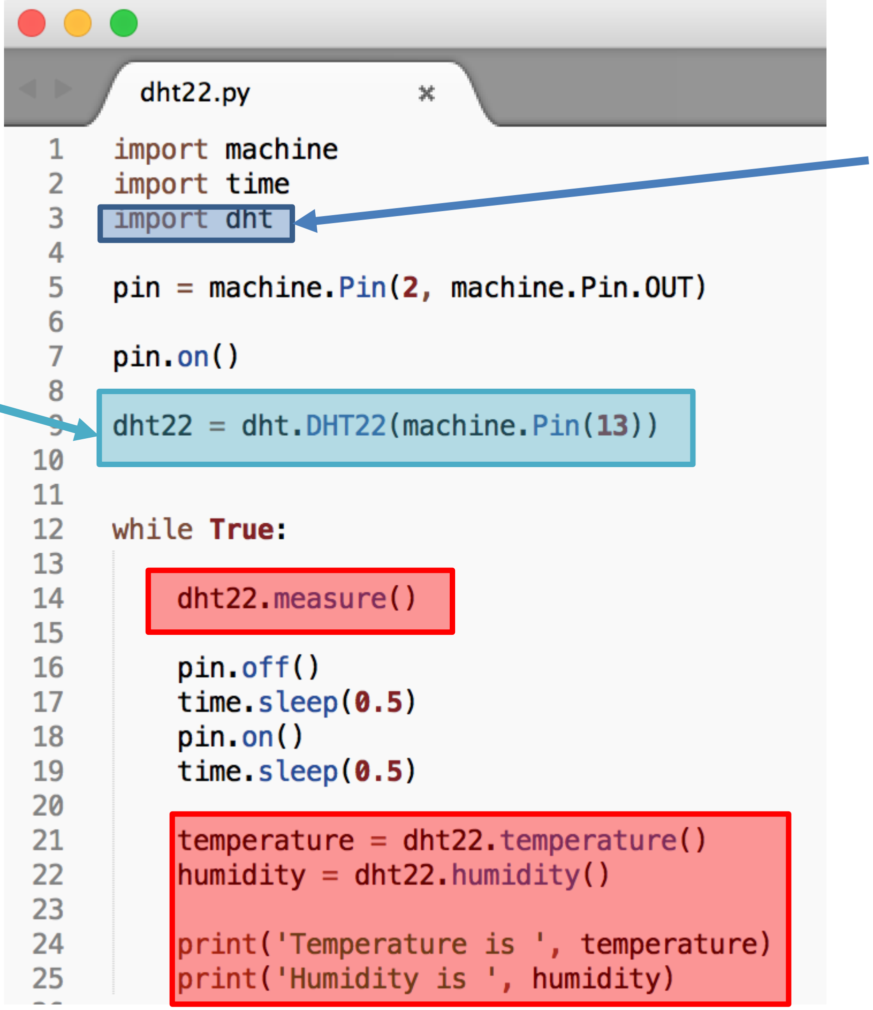
Main script

DHT (Digital Humidity & Temperature)

- DHT sensors are low cost digital sensors with capacitive humidity sensors and thermistors to measure the surrounding air.
- They feature a chip that handles analog to digital conversion and provide a 1-wire interface.



DHT 1-Wire Python Script

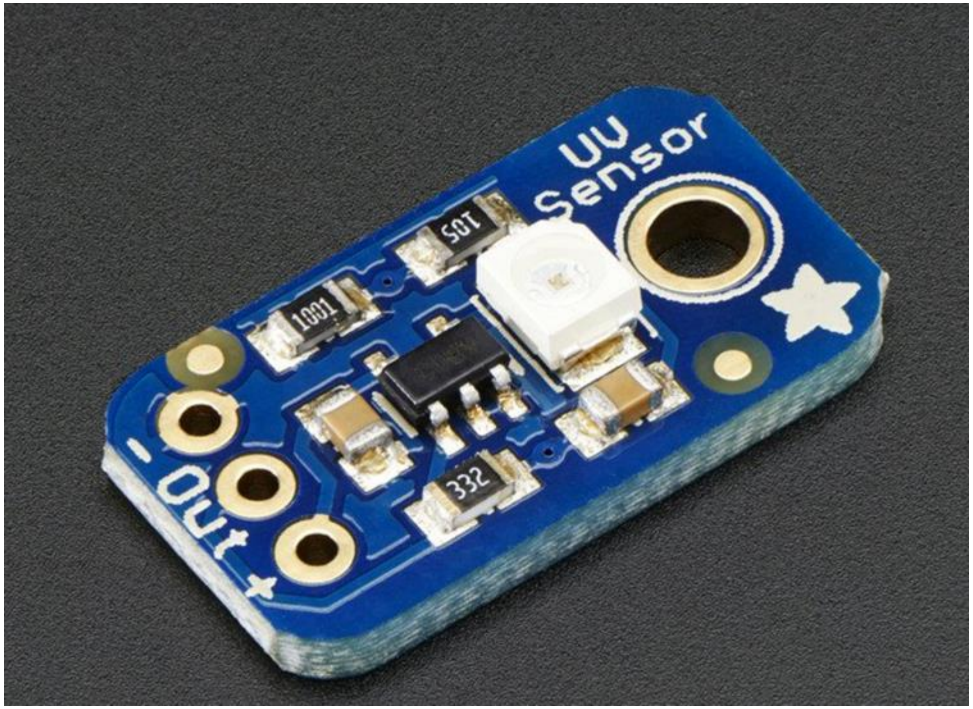


```
import machine
import time
import dht

pin = machine.Pin(2, machine.Pin.OUT)

pin.on()

dht22 = dht.DHT22(machine.Pin(13))


while True:

    dht22.measure()

    pin.off()
    time.sleep(0.5)
    pin.on()
    time.sleep(0.5)

    temperature = dht22.temperature()
    humidity = dht22.humidity()

    print('Temperature is ', temperature)
    print('Humidity is ', humidity)
```

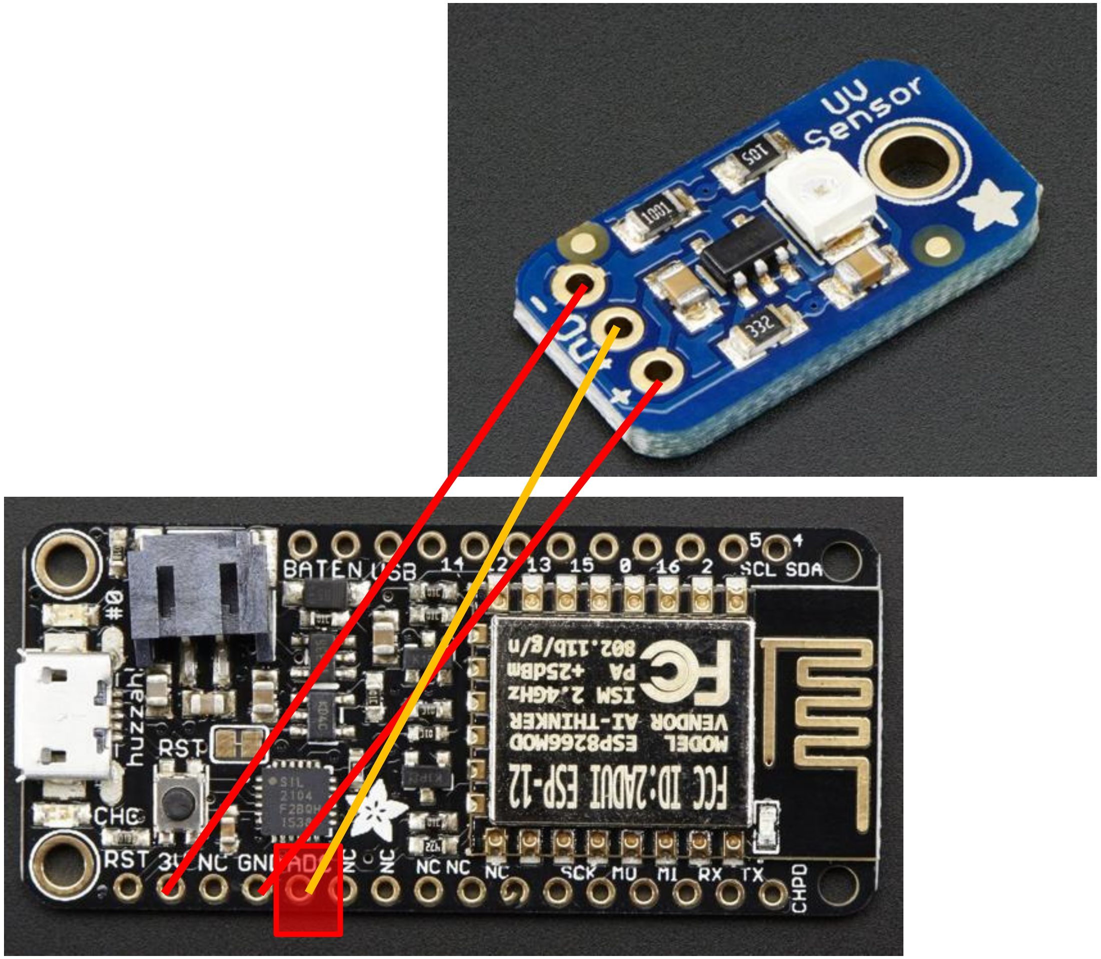
dht object

Analog UV 1918 Sensor

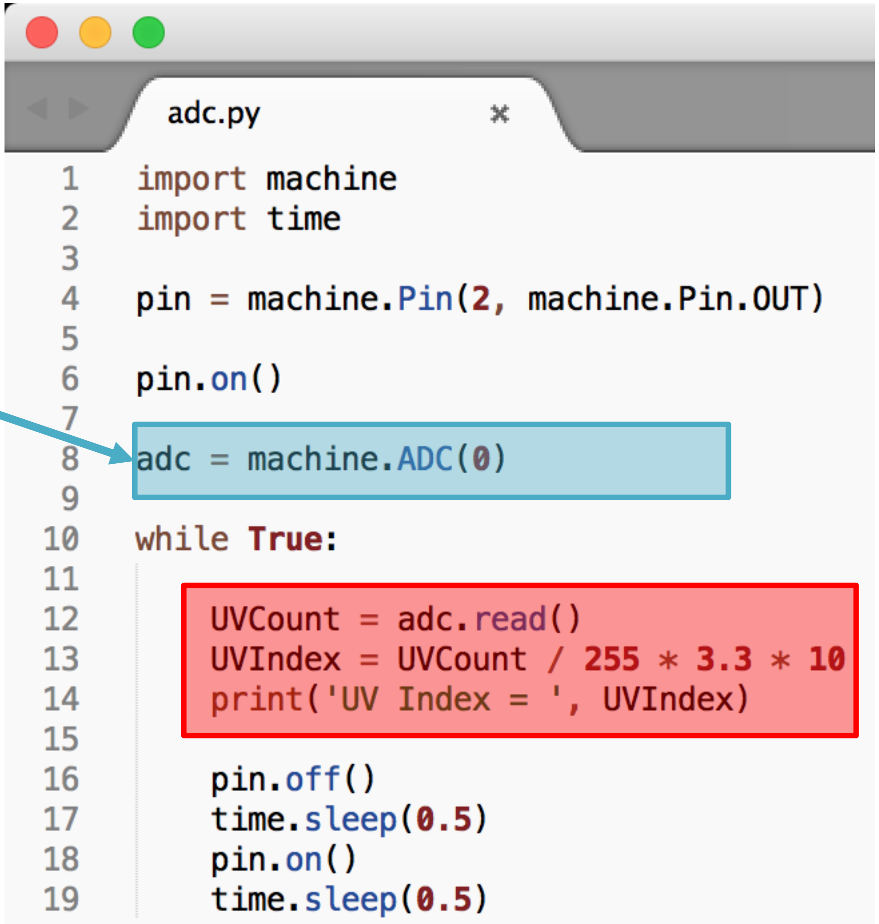
- Environmental Sensor
  - Analog
  - UV signal
  - UV Index = V / 0.1



Analog UV 1918 Sensor



ADC Python Script



```
import machine
import time

pin = machine.Pin(2, machine.Pin.OUT)

pin.on()

adc = machine.ADC(0)

while True:

    UVCount = adc.read()
    UVIndex = UVCount / 255 * 3.3 * 10
    print('UV Index = ', UVIndex)

    pin.off()
    time.sleep(0.5)
    pin.on()
    time.sleep(0.5)
```

adc object

Additional Resources

- Download Course Material for
  - Python Doxygen Templates
  - Example source code
  - Blog
  - YouTube Videos
- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

From www.beningo.com under

- Blog > CEC – Designing IoT Sensor Nodes using the ESP8266

The Lecturer – Jacob Beningo

Jacob Beningo

Principal Consultant

BENINGO EMBEDDED GROUP

Social Media / Contact

- E : jacob@beningo.com
- T : 810-844-1522
- : Jacob_Beningo
- f : Beningo Engineering
- in : JacobBeningo
- EDN : Embedded Basics

ARM Connected Community

Consulting

- Advising
- Coaching
- Content
- Consulting
- Training

www.beningo.com

DesignNews CEC CONTINUING EDUCATION CENTER

Presented by:

Digi-Key ELECTRONICS ROHDE&SCHWARZ