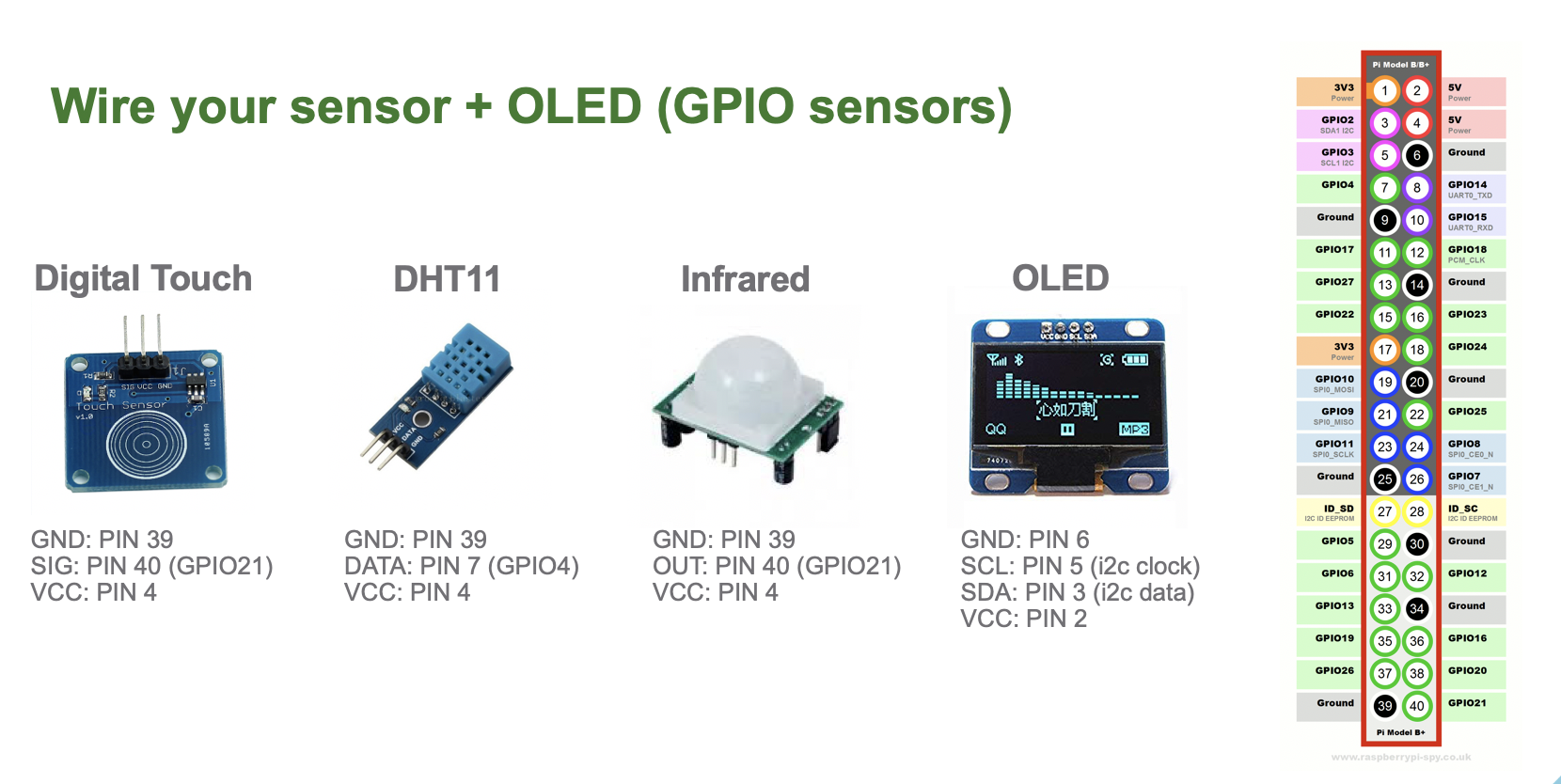
# Introduction

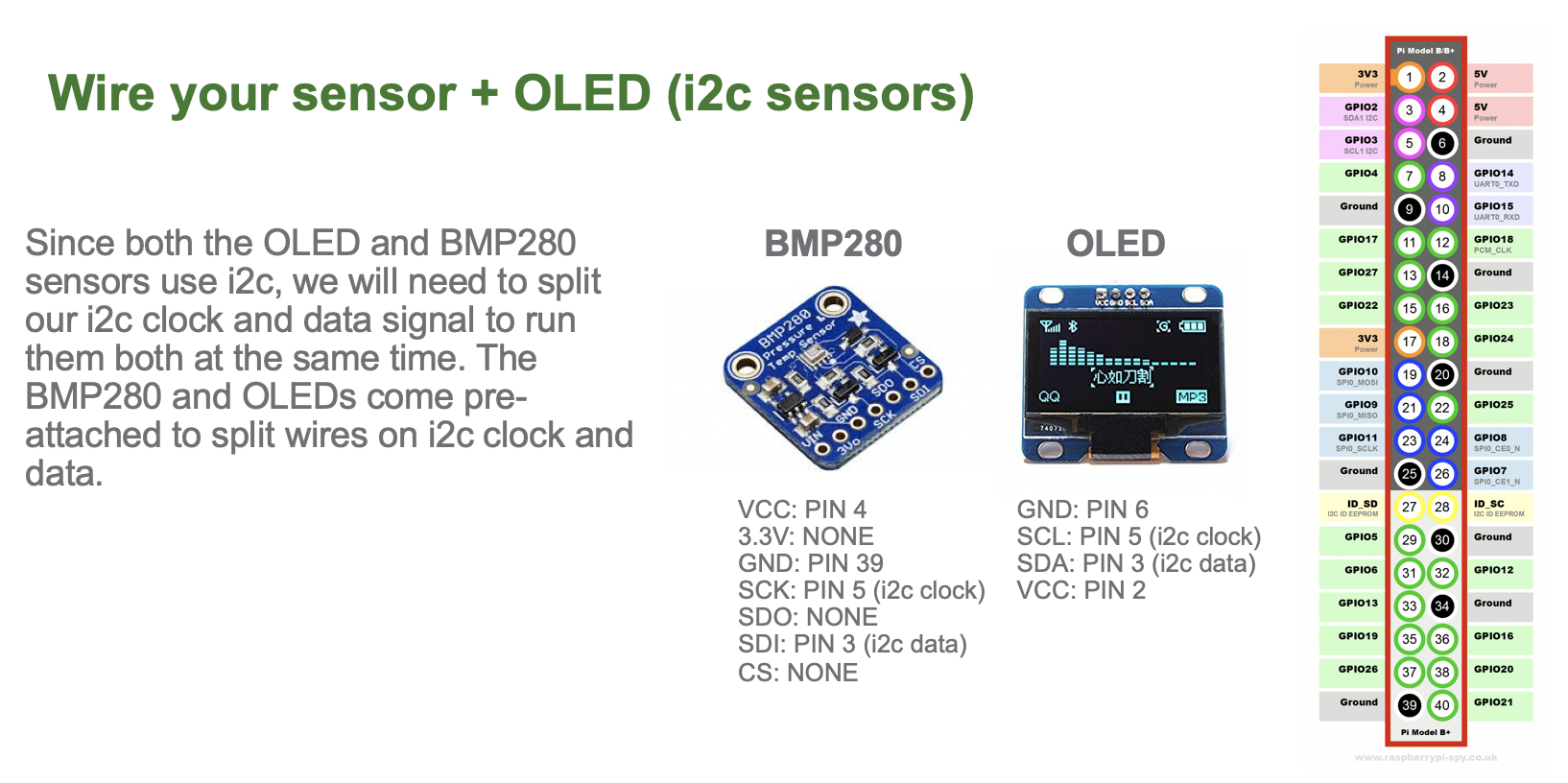
In this lab we will be covering how to wire, write and deploy Python code to run with DHT11, BMP280, Digital Touch, and Infrared sensors. The code reads sensor data and displays them on your OLED display.

# Wiring

## For GPIO Sensors



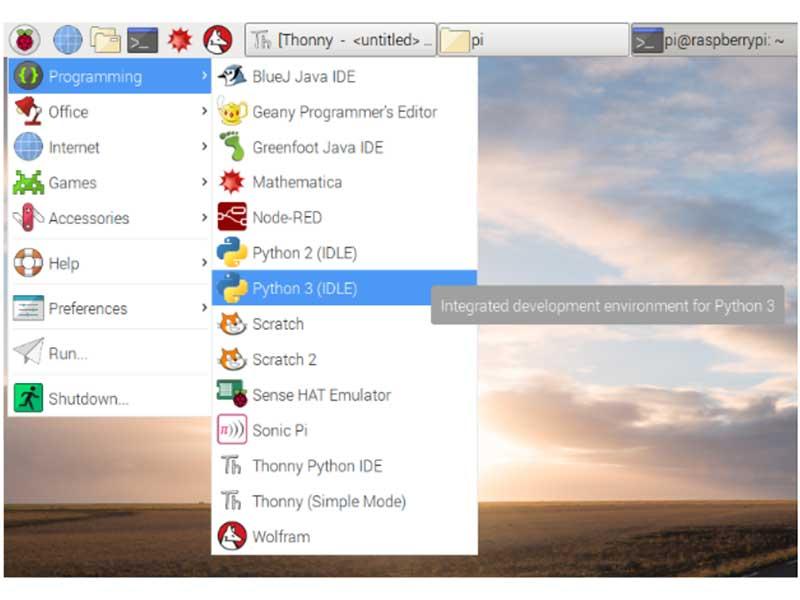
## For I2C Sensors



# Running Sensor Code

## IDLE as an Editor

Open IDLE Python IDE like below:



## Navigating to Directory With Code

At the top of the IDLE window click “open” and navigate to home/pi/Desktop/SECTIONTIME/sensortype

To run the code go to the top of the IDLE window and click Run -> Run Module

## 

# Appendix: Code for at home!

## DHT11 (Temperature + Humidity)

Reads Temperature and Humidity

### DHT11 - Run Before Start

$ pip install Pillow

$ pip install Adafruit\_SSD1306

$ pip3 install Adafruit\_DHT

$ pip install RPi.GPIO

### DHT11 - Python Code

import sys

import Adafruit\_DHT

import time

import Adafruit\_SSD1306

from PIL import Image

from PIL import ImageDraw

from PIL import ImageFont

#Setup Display

RST = 24

disp = None

def setup\_display(address):

disp = Adafruit\_SSD1306.SSD1306\_128\_32(rst=RST, i2c\_address=address)

disp.begin()

return disp

try:

disp = setup\_display(0x3C)

except OSError:

disp = setup\_display(0x3D)

#Clear Display

disp.clear()

disp.display()

#Create blank image for drawing with 1 for 1 bit color

width = disp.width

height = disp.height

image = Image.new('1', (width, height))

#Make draw object draw on image

draw = ImageDraw.Draw(image)

#Clear the image

draw.rectangle((0, 0, width, height), outline = 0, fill = 0)

#Load default font

font = ImageFont.load\_default()

# First define some constants to allow easy resizing of shapes.

padding = 2

top = padding

bottom = height-padding

while True:

#Display humidity and temperature

humidity, temperature = Adafruit\_DHT.read\_retry(11,4)

draw.rectangle((0,0,width,height), outline=0, fill=0)

draw.text((padding, -2), 'DHT11 Sensor', font=font, fill=455)

draw.text((padding, 8), 'Temperature: ' + str(temperature) + "C", font=font, fill=455)

draw.text((padding, 18), 'Humidity: ' + str(humidity) + "%" , font=font, fill=255)

# Display image.

disp.image(image)

disp.display()

#time.sleep(1)

## BMP280 (Pressure + Temperature + Altitude)

Reads pressure, temperature and altitude

### BMP280 - Run Before Start

$ pip install Pillow

$ pip install Adafruit\_SSD1306

$ pip3 install adafruit-circuitpython-bmp280

### 

### BMP280 - Python Code

import Adafruit\_GPIO.SPI as SPI

import Adafruit\_SSD1306

import time

import board

import digitalio

import busio

import adafruit\_bmp280

import PIL

from PIL import Image

from PIL import ImageDraw

from PIL import ImageFont

i2c = busio.I2C(board.SCL, board.SDA)

bmp280 = adafruit\_bmp280.Adafruit\_BMP280\_I2C(i2c)

bmp280.sea\_level\_pressure = 1013.25

def setup\_display(addr):

disp = Adafruit\_SSD1306.SSD1306\_128\_32(rst=RST, i2c\_address=addr)

disp.begin()

return disp

# Raspberry Pi pin configuration:

RST = 24

# 128x32 display with hardware I2C:

disp = None

try:

disp = setup\_display(0x3C)

except OSError:

disp = setup\_display(0x3D)

# Clear display.

disp.clear()

disp.display()

# Create blank image for drawing.

# Make sure to create image with mode '1' for 1-bit color.

width = disp.width

height = disp.height

image = Image.new('1', (width, height))

# Get drawing object to draw on image.

draw = ImageDraw.Draw(image)

# Draw a black filled box to clear the image.

draw.rectangle((0,0,width,height), outline=0, fill=0)

# Draw some shapes.

# First define some constants to allow easy resizing of shapes.

padding = 2

top = padding

bottom = height-padding

#Load default font

font = ImageFont.load\_default()

while True:

temperature = bmp280.temperature

pressure = bmp280.temperature

altitude = bmp280.altitude

# Write two lines of text.

draw.rectangle((0,0,width,height), outline=0, fill=0)

draw.text((padding, -2), 'Temperature: ', font=font, fill=455)

draw.text((75, -2), str(temperature), font=font, fill=255)

draw.text((padding, 8), 'Altitude: ', font=font, fill=455)

draw.text((55, 8), str(altitude) , font=font, fill=255)

draw.text((padding, 18), 'Pressure: ', font=font, fill=255)

draw.text((55, 18), str(pressure) , font=font, fill=255)

# Display image.

disp.image(image)

disp.display()

time.sleep(1)

## DIGITAL TOUCH

Reads touch

### DIGITAL TOUCH - Run Before Start

$ pip install Pillow

$ pip install Adafruit\_SSD1306

$ pip install RPi.GPIO

$ pip install adafruit-blinka

### DIGITAL TOUCH - Python Code

import RPi.GPIO as IO

import time

import Adafruit\_GPIO.SPI as SPI

import Adafruit\_SSD1306

from PIL import Image

from PIL import ImageDraw

from PIL import ImageFont

import board

from digitalio import DigitalInOut, Direction

#Setup Display

RST = 24

disp = None

def setup\_display(address):

disp = Adafruit\_SSD1306.SSD1306\_128\_32(rst=RST, i2c\_address=address)

disp.begin()

return disp

try:

disp = setup\_display(0x3C)

except OSError:

disp = setup\_display(0x3D)

#Clear Display

disp.clear()

disp.display()

#Create blank image for drawing with 1 for 1 bit color

width = disp.width

height = disp.height

image = Image.new('1', (width, height))

#Make draw object draw on image

draw = ImageDraw.Draw(image)

#Clear the image

draw.rectangle((0, 0, width, height), outline = 0, fill = 0)

#Load default font

font = ImageFont.load\_default()

while True:

IO.setmode(IO.BCM)

pad\_pin = board.D11

pad = DigitalInOut(pad\_pin)

pad.direction = Direction.INPUT

if pad.value:

draw.text((6, 0), "Digital Touch Sensor", font=font, fill=455)

draw.text((40, 15), 'PRESSED', font=font, fill=455)

else:

draw.text((6, 0), "Digital Touch Sensor", font=font, fill=455)

disp.image(image)

disp.display()

time.sleep(0.1)

disp.clear()

draw.rectangle((0, 0, width, height), outline = 0, fill = 0)

IO.cleanup()

time.sleep(0.1)

## INFRARED (Motion)

Detects motion

### INFRARED - Run Before Start

$ pip install Pillow

$ pip install Adafruit\_SSD1306

$ pip install RPi.GPIO

### 

### 

### INFRARED - Python Code

import RPi.GPIO as IO

import time

import Adafruit\_GPIO.SPI as SPI

import Adafruit\_SSD1306

from PIL import Image

from PIL import ImageDraw

from PIL import ImageFont

#Setup Display

RST = 24

disp = None

def setup\_display(address):

disp = Adafruit\_SSD1306.SSD1306\_128\_32(rst=RST, i2c\_address=address)

disp.begin()

return disp

try:

disp = setup\_display(0x3C)

except OSError:

disp = setup\_display(0x3D)

#Clear Display

disp.clear()

disp.display()

#Create blank image for drawing with 1 for 1 bit color

width = disp.width

height = disp.height

image = Image.new('1', (width, height))

#Make draw object draw on image

draw = ImageDraw.Draw(image)

#Clear the image

draw.rectangle((0, 0, width, height), outline = 0, fill = 0)

#Load default font

font = ImageFont.load\_default()

while True:

IO.setmode(IO.BCM)

IO.setup(21, IO.IN)

if IO.input(21):

draw.text((6, 0), "INFRARED SENSOR:", font=font, fill=455)

draw.text((16, 12), 'MOTION DETECTED', font=font, fill=455)

else:

draw.text((6, 0), "INFRARED SENSOR:", font=font, fill=455)

draw.text((10, 12), 'NO MOTION DETECTED', font=font, fill=455)

disp.image(image)

disp.display()

time.sleep(1)

disp.clear()

draw.rectangle((0, 0, width, height), outline = 0, fill = 0)

IO.cleanup()

time.sleep(1)