Link to These Slides

seascf.seas.gwu.edu/workshops
What is a Raspberry Pie?

- A baked dish made of pastry dough and raspberry filling
What is a Raspberry Pi?

- Single-board computer
- Developed in the UK
- Several models
- Inexpensive ($5 for cheapest model, the Raspberry Pi Zero)
- Can be used with a computer monitor, keyboard, and mouse
What is Raspbian?

- Operating system optimized for the Raspberry Pi
- Based on the Linux kernel
- Can be used like a desktop computer or through the terminal
Raspberry Pi 3 Model B

- **Dimensions**: 85.6mm x 56mm x 21mm
- **40 Pin Extended GPIO**
- **Broadcom BCM2837 64bit Quad Core CPU at 1.2GHz, 1GB RAM**
- **On Board Bluetooth 4.1 Wi-Fi**
- **MicroSD Card Slot**
- **DSI Display Port**
- **Micro USB Power Input. Upgraded switched power source that can handle up to 2.5 Amps**
- **4 x USB 2 Ports**
- **10/100 LAN Port**
- **3.5mm 4-pole Composite Video and Audio Output Jack**
- **CSI Camera Port**
- **Full Size HDMI Video Output**
# Raspberry Pi vs. Arduino

## Raspberry Pi
- Allows graphical user interface
- Can be directly connected to Internet
- More powerful and more memory
- Can be used with more programming languages

## Arduino
- Low power consumption
- Can directly read analog inputs
- Requires less hardware (monitor, mouse, etc.) to get started
- No operating system needs to be installed
Connecting to the Internet

- **Wired Connection**: Registered Raspberry Pis can instantly connect to the GW network from the SEH Studio Labs using an ethernet cable. This is the easiest and fastest option.

- **GWWireless**: GWWireless does not support Raspberry Pis.

- **eduroam**: Raspberry Pis can be connected to eduroam by modifying two configuration files and running some commands.
  - Instructions: [seascf.seas.gwu.edu/eduroam-connection](seascf.seas.gwu.edu/eduroam-connection)
Keyboard Check

Open the web browser and type # and “ in the address bar

If the symbols do not appear as expected, follow these steps:

1. Click the Raspberry Pi icon in the upper-left corner, click Preferences, and click Raspberry Pi Configuration
2. Click Localisation, Click Set Keyboard, choose United States as country, select English (US) for variant, and click OK
Creating a Basic Python Program

1. Open Terminal

2. Type `nano helloworld.py` and press ENTER to open a new file in the nano text editor.

3. Type `print("Hello, World!")`

4. Use CTRL + O and ENTER to save.

5. Exit with CTRL + X.

6. Type `python helloworld.py` and press ENTER to run the program.

Breadboard Setup
Jumper Cables
Breadboard

Diagram from Tweaking4All
Connecting to a Breadboard
Connecting to a Breadboard
Connecting to a Breadboard

Note the red strip on the ribbon is on the opposite side of the Raspberry Pis from the USB ports
Basic LED Circuit
Light Emitting Diode

- A **diode** is a component that only allows flow of current in one direction.
- A **light emitting diode (LED)** emits light when current passes in the correct direction.
Circuit Basics

- Ohm’s Law: $V = IR$
  - $V$: Voltage (volts)
  - $I$: Current (amperes)
  - $R$: Resistance (ohms)
- LEDs have a maximum current
- Ohm’s Law (rewritten): $I = \frac{V}{R}$
  - To keep current ($I$) low, resistance ($R$) must be high enough
Building the LED Circuit

- Two jumper cables
- LED
- 220 Ω resistor
Building the LED Circuit

- Add a resistor to the breadboard
- Connect a wire from the red power rail to one end of a resistor
Building the LED Circuit

- Connect the anode (long end) of the LED to the resistor
Building the LED Circuit

- Connect a wire from the cathode (short end) of the LED to the blue ground rail of the breadboard
- The LED should glow!
Blinking an LED

- Connect the cable from the resistor to pin 21 of the breakout board
Blinking an LED

- Open Terminal
- Type `nano blinky.py` and press ENTER to open a new file in the nano text editor
- Type the code on the right
- Use CTRL + O and ENTER to save
- Exit with CTRL + X
- Type `python blinky.py` and press ENTER to run the program

Code:
```python
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(21, GPIO.OUT)
for i in range(0,100):
    GPIO.output(21, i % 2)
    time.sleep(0.25)
GPIO.cleanup(21)
```
Blinking an LED

Code:
import RPi.GPIO as GPIO # Import library for input/output
import time # Import library for time

GPIO.setmode(GPIO.BCM) # Allow Pi to recognize pin numbers

GPIO.setup(21, GPIO.OUT) # Set pin 21 as output

for i in range(0,100): # Run next code for i = 0,1,2,...,100
    GPIO.output(21, i % 2) # Set pin 21 to remainder of i/2
    time.sleep(0.25) # Wait for 0.25 seconds

GPIO.cleanup(21) # Unset pin 21 to avoid later issues
Button-Activated LEDs
Push Button

- All four pins are **connected** when pressed on
- If off, the **left** and **right** are separated
Adding a Button

- Insert the button into your breadboard
- Connect one side of the button to 3.3 V
- Connect the other side to GPIO pin 19
Adding a Button

In the terminal, enter `nano button.py` to create a new file, then type:

```python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)  # Use GPIO board pin numbers
GPIO.setwarnings(False)  # Disable warnings from no cleanup
GPIO.setup(19, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)  # Pin 19 will be input for button with default of low voltage
GPIO.setup(21, GPIO.OUT)  # Pin 21 will be output for LED

while True:  # Loop forever
    if GPIO.input(19) == True:  # If pin 19 (button) is on
        GPIO.output(21, True)  # Set pin 21 (LED) on
        time.sleep(5)  # Keep the LED on for 5 secs
    else:  # If pin 19 input is off
        GPIO.output(21, False)  # Set pin 21 (LED) off
```

- Use CTRL + O and ENTER to save
- Exit with CTRL + X
- Type `python button.py` and press ENTER to run the program
- Use CTRL + C to stop
Camera
Step 1: Connecting Your Camera

- Remove green cover if it’s still covering the camera lens
- Remove red cover if it’s still covering the ribbon cable port next to the HDMI port
- Fully insert the blue end of the ribbon cable into the port with the metallic part facing the HDMI port
Step 2: Using Your Camera

- To take a photo: `raspistill -o image.jpg`
  - Use flags `-vf -hf` to flip the image right-side-up
    `(raspistill -vf -hf -o cam2.jpg)`
  - By default the camera takes pictures after 5 seconds. Use `-t` to change it

- To take a 10s video: `raspivid -o video.h264 -t 10000`

- View video: `omxplayer video.h264`

- Note: using the .h264 format is sped up. To convert to mp4:
  - `sudo apt-get install -y gpac`
  - `MP4Box -add video.h264 video.mp4`

- View video: `omxplayer video.mp4`
Using the Camera Module with Python

Create Python file: nano camera.py

Code:
# Basic program to take a photo
import picamera
camera = picamera.PiCamera() # initialize the camera
camera.capture("image.jpg")

View your image: gpicview image.jpg
Delete your image: rm image.jpg
Sense HAT
Sense HAT

- Includes:
  - 8 x 8 RGB LED matrix
  - Five-button joystick
  - Gyroscope
  - Accelerometer
  - Magnetometer
  - Thermometer
  - Barometric pressure
  - Humidity
- Has been used on the International Space Station
Sense HAT Basics

Install the Sense HAT package

- `sudo apt-get install sense-hat`
- Ensure that you **unplug** your Pi
- Attach your Sense HAT
Setting up the code and sending text to the HAT

- Create a Python file: `nano hello_world.py`
- Create Sense HAT object:
  ```python
  from sense_hat import SenseHat
  sense = SenseHat()
  ```
- Have text scroll across the Sense HAT:
  ```python
  sense.show_message("Hello world")
  ```
- Run: `python hello_world.py`
Colors

- All colors humans see are mixtures of red, blue and green
- Computers store data as 0s and 1s, called bits
- Bits are often grouped in sets of 8 called bytes, which can represent values from 0 to 255
- The Sense HAT takes colors defined using the (R, G, B) pattern
  - Examples:
    - red = (255, 0, 0)
    - yellow = (255, 255, 0)
    - white = (255, 255, 255)
- Color mixer: [w3schools.com/colors/colors_rgb.asp](http://w3schools.com/colors/colors_rgb.asp)
Displaying Background Color on the Sense HAT

We can define global variables for red, green and blue values to quickly change later.

```python
from sense_hat import SenseHat

sense = SenseHat()

r = 255
g = 0
b = 0

sense.clear((r, g, b))
```
Changing Text Color on the Sense HAT

```python
from sense_hat import SenseHat

sense = SenseHat()

red = (255, 0, 0)
background = (255, 255, 255)

sense.show_message("Red Text", text_colour=red, back_colour=background)
sense.clear()
```
Displaying a single character on Sense HAT

```python
from sense_hat import SenseHat

sense = SenseHat()
sense.show_letter("T")
```
Setting a single pixel on the matrix

The Sense HAT’s LED matrix uses a coordinate system with an x- and a y-axis.

The blue pixel is at coordinates 0, 2.

The red pixel is at coordinates 7, 4.
Setting a single pixel on the matrix

```python
from sense_hat import SenseHat
sense = SenseHat()
blue = (0, 0, 255)
red = (255, 0, 0)
sense.set_pixel(0, 2, blue)
sense.set_pixel(7, 4, red)
```
Motion Detector
PIR Motion Sensor Detector Module

- **Time Delay Adjust**
  - Clockwise increases delay

- **Sensitivity Adjust**
  - Clockwise decreases range

- **Trigger Selection Jumper**
  - See next slide

- **Pins**
  - **Power**: Should be between 5 and 20 V input
  - **Ground**: Should be connected to ground
  - **Output**: Will be 3.3 V if activated, 0 if not

Diagram from [Henry’s Bench](http://henrysbench.com)
PIR Motion Sensor Detector Module

Single Trigger Mode – Time Delay is started immediately upon detecting motion. Continued detection is blocked.

Repeatable Trigger Mode – Time Delay is re-started every time motion is detected.

Diagram from Henry’s Bench
Step 1: Connect Motion Sensor

Connect:
- Power to 5V
- Ground to GND
- Output to 21
Step 2: Connect LED

- Connect negative (short) end of LED to GND
- Connect positive (long) end of LED to resistor
- Connect resistor to pin 19
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(21, GPIO.IN)  # Read output from PIR motion sensor
GPIO.setup(19, GPIO.OUT)  # LED output pin
while True:
    i = GPIO.input(21)
    if i == 0:
        # When output from motion sensor is LOW
        print "No intruders", i
        GPIO.output(19, 0)  # Turn OFF LED
        time.sleep(.05)
    elif i == 1:
        # When output from motion sensor is HIGH
        print "Intruder detected", i
        GPIO.output(19, 1)  # Turn ON LED
        time.sleep(.05)
Motion-Activated Camera
Motion-Activated Camera

- Connect negative (short) end of blue LED to GND
- Connect positive (long) end of LED to resistor
- Connect resistor to pin 16
Motion-Activated Camera

# !/usr/bin/python
import picamera
import time
import RPi.GPIO as GPIO

camera = picamera.PiCamera() # connect to the camera
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(19, GPIO.OUT) # red LED
GPIO.setup(16, GPIO.OUT) # blue LED
GPIO.setup(24, GPIO.IN) # sensor input

j = 0
print "armed"
while True:
    input = GPIO.input(24)
    if input == 0:
        GPIO.output(16, GPIO.HIGH)  # turn on blue light
        GPIO.output(19, GPIO.LOW)   # red light off
        time.sleep(0.1)
    elif input == 1:
        GPIO.output(16, GPIO.LOW)   # blue off
        GPIO.output(19, GPIO.HIGH)  # red on
        name = "image" + str(j) + ".jpg"
        j += 1
        camera.capture(name)
        time.sleep(5)  # wait 5 secs so we don’t take too many pictures

List files: ls      View your image: gpicview  <file_name>
Remove all images: rm  image*.jpg
Weather Station Twitter Bot
from sense_hat import SenseHat
import time
import sys

sense = SenseHat()
sense.clear()

temp = sense.get_temperature()
temp = 1.8 * round(temp, 1) + 32
print("Temperature F", temp)
humidity = sense.get_humidity()
humidity = round(humidity, 1)
print("Humidity :", humidity)
Twitter Bot

Run commands to get Twitter libraries:

- `sudo apt-get install python-setuptools`
- `sudo easy_install pip`
- `sudo pip install twython`
Twitter Bot

Under `import sys`, add this code:

```python
from twython import Twython

# API Key constant variables
CONSUMER_KEY = '******************YOUR DATA***********'
CONSUMER_SECRET = '******************YOUR DATA***********'
ACCESS_KEY = '******************YOUR DATA***********'
ACCESS_SECRET = '******************YOUR DATA***********'

# Copy of the Twython object with all our keys and secrets to allow easy commands
api = Twython(CONSUMER_KEY,CONSUMER_SECRET,ACCESS_KEY,ACCESS_SECRET)

Get the API keys and secrets at [go.gwu.edu/5al](go.gwu.edu/5al)

At the end, add this code:

```python
api.update_status(status='Temperature: {}
Humidity: {}%'.format(temp, humidity))
```
More Things to Try

- https://pimylifeup.com/raspberry-pi-weather-station/
  ○ See “Improving your weather station – Utilizing the LED Matrix”

- https://pimylifeup.com/raspberry-pi-twitter-bot/
  ○ See “Tweeting Webcam Photos via your Raspberry Pi Twitter Bot”
Link to These Slides

seascf.seas.gwu.edu/workshops

Survey: go.gwu.edu/fall2018survey