SEAS Computing Facility
Raspberry Pi Workshop 4:
Sense HAT and Web Server

November 4, 2017
Overview

- Learn about the Sense HAT
- Create a Pong game to play on the Sense HAT using Python
- Set up the Raspberry Pi as a home web server
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  helloWorld.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")
  ```
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
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)
```
Displaying a single character on Sense HAT

```python
from sense_hat import SenseHat

sense = SenseHat()

t = sense.show_letter('T')
```
Sleep

Sometimes, for various reasons, you’ll want your program to pause in between lines of code. To do this use the sleep function use sleep and the time parameter -- in seconds.

```python
from sense_hat import SenseHat
from time import sleep

sense = SenseHat()
buff = (240, 220, 130)
blue = (0, 0, 255)
sense.show_letter("G", buff)
sleep(1)
sense.show_letter("W", blue)
sleep(1)
```
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

```
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)
```
Functions in Python

```python
x = 3                       # a global variable

def my_function(y):         # define my_function with parameter y
    global x               # reference to the global variable
    x += y

my_function(2)              # call the function
```
Make a Bat

Create a Python file: nano pong.py

- Define the color for your bat
- Create a global variable bat_y for the y-coordinate of the bat and set it to 4
- Define a function called draw_bat with no parameters
- In the function, set the pixel with coordinates (0, bat_y) to the color
- Also set the pixels above and below so that the bat is three pixels tall
- Call the function from the main code
Make a Bat

white = (255, 255, 255)

bat_y = 4

def draw_bat():
    sense.set_pixel(0, bat_y, white)
    sense.set_pixel(0, bat_y + 1, white)
    sense.set_pixel(0, bat_y - 1, white)

draw_bat()

Test: does the bat appear?
def move_up(event):
    global bat_y
    if event.action == 'pressed' and bat_y > 1:
        bat_y -= 1

while True:
    sense.stick.direction_up = move_up
    sense.stick.direction_down = move_down
    draw_bat()

Also define move_down which increases bat_y, but only up to 6

Test: does the bat move up and down?
while True:
    sense.stick.direction_up = move_up
    sense.stick.direction_down = move_down
    sense.clear(0, 0, 0)
    draw_bat()
    sleep(0.25)
Make a Ball

- Define a color for the ball
- Add global list variables:
  - ball_position = [3, 3]
  - ball_velocity = [1, 1]
- Define draw_ball as a function to set the pixel at (ball_position[0], ball_position[1]) to the ball color
Make a Ball

def draw_ball():
    sense.set_pixel(ball_position[0], ball_position[1], blue)

Call `draw_ball()` in the while loop

Test: does the ball appear on the LED matrix?
Move the Ball

def draw_ball():
    sense.set_pixel(ball_position[0], ball_position[1], blue)
    ball_position[0] += ball_velocity[0]

Test: does the ball move?
def draw_ball():
    sense.set_pixel(ball_position[0], ball_position[1], blue)
    ball_position[0] += ball_velocity[0]
    if ball_position[0] == 7 or ball_position[0] == 0:
        ball_velocity[0] = -ball_velocity[0]

Test: does the ball bounce?

Now do this for the y direction
Move the Ball

```python
if ball_position[1] == 7 or ball_position[1] == 0:
    ball_velocity[1] = -ball_velocity[1]
```

Test: does the ball move diagonally?
Collision with the bat

- When should the ball bounce?
  - x coordinates?
  - y coordinates?
Collision with the bat

Add to the `draw_ball` function:

```python
if ball_position[0] == 1 and (bat_y - 1) <= ball_position[1] <= (bat_y + 1):
    ball_velocity[0] = -ball_velocity[0]
```

Test: does the ball bounce properly?
Losing the game

- What happens if you the ball misses the bat?
- What should happen?
Losing the game

Add to the `draw_ball` function:
```python
    if ball_position[0] == 0:
        sense.show_message("Lose")
```

Test: do you lose the game when you should?
Feature ideas:

- Count the number of bat bounces and show the count at the end of the game
- Make moving the joystick left and right adjust the speed
- Give the player three lives
Creating and uploading pictures to a web server with Raspberry Pi
Setting up the push button to take pictures

- Insert the button into your breadboard.
- Connect one side of the button to 3.3V, and the other side to GPIO 26.
Connect 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
Making the button work with the camera

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

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

GPIO.setmode(GPIO.BCM)  # Use GPIO board pin numbers
GPIO.setwarnings(False)  # Disable warnings from not cleaning up
GPIO.setup(26, GPIO.IN)  # Pin 26 will be input for button
GPIO.setup(21, GPIO.OUT)  # Pin 21 will be output for LED
camera = picamera.PiCamera()
j=0
```
while True:  # Loop forever
    if GPIO.input(26) == True:  # If pin 26 (button) is on
        GPIO.output(21, True)  # Set pin 21 (LED) on
        Name = "image" + str(j) + ".jpg"
        J += 1
        camera.capture(name)
        time.sleep(5)  # Keep the LED on for 5 secs
    else:  # If pin 26 input is off
        GPIO.output(21, False)  # Set pin 21 (LED) off
Creating a web server with Raspberry Pi

Let’s install ‘Apache’, the main component of the project

- `sudo apt-get install apache2 apache2-doc apache2-utils`

We must then install PHP to get into the Pi

- `sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache`
Installing the database

We must then install the package for database connectivity (PHP->SQL):

- `sudo apt-get install php5-mysql`

Install the MySQL server which stores everything, our database for the Raspberry Pi web server:

- `sudo apt-get install mysql-server mysql-client`
Testing your web server

Find the ip address of your pi

- In terminal type: Hostname -l

Open a web browser and type the ip address or type in ‘localhost’

You can edit the webpage by going editing the index.html file in /var/www/html

Make sure you use sudo
Making pictures accessible on your server

Move your buttonCamera.py code to the /var/www/html directory
  - In terminal: sudo mv buttonCamera.py /var/www/html

To run it make sure to be a super user
  - In terminal: sudo python buttonCamera.py
Viewing your content as a web page on your server

Everything in the /var/www/html directory will be viewable through the browser.

Open your browser, type in your ip address or localhost then the name of the file or image

- In the url: localhost/image0.jpg
Other places to find Raspberry Pi projects

- http://makezine.com/category/technology/raspberry-pi/
- https://hackaday.io/projects/tag/raspberry%20pi