SEAS Computing Facility
Raspberry Pi Workshop 2: I/O Camera & Motion Sensor

October 21, 2017
Overview for Today

- Learn about new components
- Program a push-button ON/OFF input system for LED
- Connect and program camera
- Connect and program motion sensor with LED
- Combine camera, motion sensor and LED
Supplies

1. Monitor
2. Monitor power supply
3. HDMI–HDMI or HDMI–DVI cable
4. Keyboard
5. Mouse
6. Ethernet cable
7. Raspberry Pi power supply
8. Raspberry Pi
9. Breadboard
10. Ribbon cable and breakout board
11. 4 female–female jumper cables
12. 2 resistors
13. 1 red LED
14. 1 green LED
15. Button
16. Camera
17. Motion sensor
18. 3 male–female jumper cables
Components
Jumper Cables

Male

Female
Breadboard

Diagram from Tweaking4All
Push Button

- All four pins are **connected** when pressed on
- If off, the **left** and **right** are separated
Raspberry Pi Camera Module V2

- 1080p HD video at 30 frames/second
- 720p HD video at 60 frames/second
- 8 Megapixels for still photos (3280 x 2464)
- Fixed Focus Lens
- Connected to Raspberry Pi with 15-pin ribbon cable

https://www.amazon.com/Raspberry-Pi-Camera-Module-Megapixel/dp/B01ER2SKFS
PIR Motion Sensor Detector Module

- PIR: Passive Infrared
  - Senses infrared radiation from objects
- Range is adjustable up to 7 meters
- Viewing area is approximately a 120° cone

Button-Activated LEDs
Step 1: Connect the Push Button

- Insert the button into your breadboard.
- Connect one side of the button to 3.3V, and the other side to GPIO 26.
Step 2: Connect the LED

- Attach the LED to the breadboard.
- Connect a resistor from the negative side of the LED (shorter side) to ground.
- Connect the positive side of the LED to GPIO 21.
Step 3: Code
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 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

while True:  # Loop forever
    if GPIO.input(26) == True:  # If pin 26 (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 26 input is off
        GPIO.output(21, False)  # Set pin 21 (LED) off
```
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

Download the Python libraries for the Raspberry Pi Camera:
```
sudo apt-get update
sudo apt-get install python-picamera
```

Documentation for the picamera library can be viewed at: https://www.raspberrypi.org/documentation/usage/camera/python/README.md
Using the Camera Module with Python

Create Python file: `nano camera.py`

**Code:**

```python
# 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`
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
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

- 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
Step 3: Code (motion_detector.py)

```python
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:
        print "No intruders", i
        GPIO.output(19, 0)  # Turn OFF LED
        time.sleep(.05)
    elif i == 1:
        print "Intruder detected", i
        GPIO.output(19, 1)  # Turn ON LED
        time.sleep(.05)
```
Motion-Activated Camera
Combining PIR Sensor, LED and Camera

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

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT)  # red LED
GPIO.setup(16, GPIO.OUT)  # green LED
GPIO.setup(24, GPIO.IN)   # sensor input

j = 0
name = "image" + str(j) + ".jpg"
print "armed"
```
while True:
    i = GPIO.input(24)
    if i == 0:
        print "Nothing sensed"
        GPIO.output(16, GPIO.HIGH)  # turn on green light
        GPIO.output(23, GPIO.LOW)  # red light off
        time.sleep(0.1)
    elif i == 1:
        print "motion sensed"
        GPIO.output(16, GPIO.LOW)  # green off
        GPIO.output(23, GPIO.HIGH)  # red on
        time.sleep(0.1)
    if os.path.exists(name):  # checks to make sure you don't overwrite a file
        j += 1  # j = j + 1
        name = "image" + str(j) + ".jpg"
        nName = name.format()  # formats variable to string to pass to system
        os.system("raspistill -vf -hf -t 1 -o %s" % nName)  # take photo in 1 sec

List files: ls      View your image: gpicview <file_name>
Remove all images: rm image*.jpg
Combining PIR Sensor, LED and Camera (Alternative)

# !/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(23, GPIO.OUT) # red LED
GPIO.setup(16, GPIO.OUT) # green 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 green light
        GPIO.output(23, GPIO.LOW)   # red light off
        time.sleep(0.1)
    elif input == 1:
        GPIO.output(16, GPIO.LOW)  # green off
        GPIO.output(23, 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
Extension Ideas

Easy:

● Take video for 10 seconds after motion detected
● Time-lapse photography

Medium:

● Use button to activate or deactivate security system
● Have multiple LEDs represent how long since motion was detected
● Use secret code in terminal to deactivate security system
● Define your own functions in your code

Hard:

● Send email with photo of intruder (consider smtplib)
● Take photo upon receiving specific email