Raspberry Pi Camera Trap: In this lab, we will use the Raspberry Pi to take input from a IR motion sensor and activate the Raspberry Pi camera to capture images of whatever endotherm activated the motion sensor.

Ewok costume: $69
Chewbacca costume: $129
Dressing up to mess with your neighbor's trail cam: Priceless.
ifunny.co

Here's an example of camera trap footage in a backyard:
https://www.youtube.com/watch?v=hFZFjoX2cGg&t=3s

NOTE: This project has the potential to be used in many creative ways, and capture human as well as animal movements and behavior and gain insight to both. As with any photography, please exercise discretion to ensure that the photos don't cause harm or violations of privacy, etc.

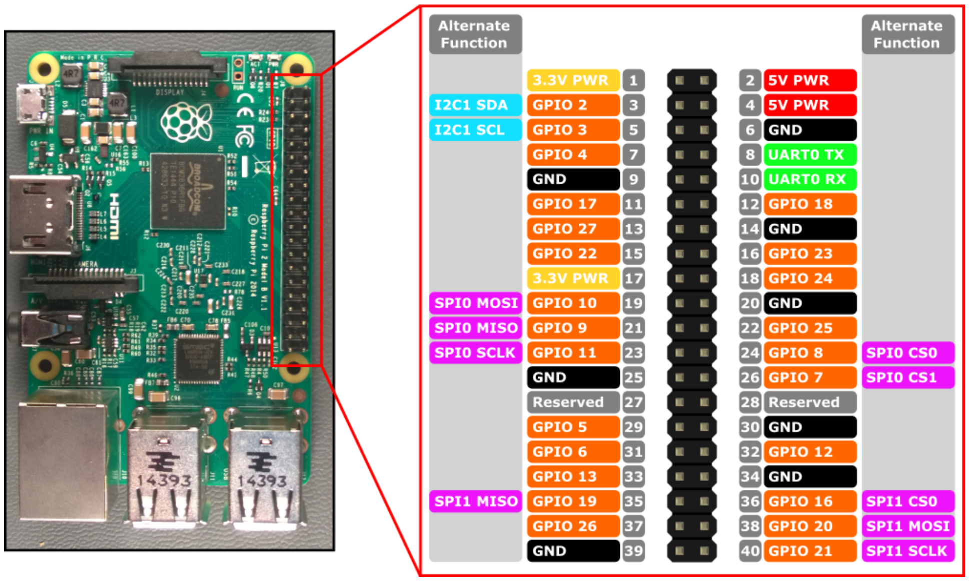

Figure 1: Schematic of the Raspberry Pi and its General Purpose Input/Output (GPIO) pins.

Wire up the Motion sensor

1. Connect the VCC(+5V), GND, and OUT on the IR motion sensor to the corresponding pins on the Raspberry Pi using your breadboard. If you have questions about this, ask me. Note, I recommend not using GPIO2 or GPIO3.
2. Test that python can talk to the motion sensor by querying the GPIO port and printing the result to the screen. Use this tutorial to set up the GPIO pins. https://goo.gl/6LaCjf. Write a short python script on the pi using nano to test the functionality.
3. When you observe the changing binary state of the motion sensor, you have it set up and can move on.

Work with your group to develop an algorithm for the camera trap.

4. Think through the logical sequence of events that need to take place to check whether an endotherm passed in front of the motion sensor, and then capture and write an image. **Write down your algorithm and submit it as part of your writeup.** If you did it on the whiteboard, take a picture.

Connect the Raspberry Pi Camera

5. Insert the wire tape to the port labeled Camera on the Pi board. I can explain how to secure this.
6. **[This should already be done, so can skip this step.]** Enable the Camera driver on the Raspberry Pi. “sudo raspi-config” navigate to peripherals and follow the instructions to enable the camera. This will turn on the SPIO ports that are assigned to the camera.

7. The Python PiCamera library explains how to interact with the camera. https://picamera.readthedocs.io/en/release-1.13/

Write your code

8. Send a copy of picam_trap.py to the Raspberry Pi using SFTP or Putty. You will do this the same way as sending the serialpie.py file from the In-class exercise.
9. Translate your algorithm to python code and add this to the picam_trap.py using nano.
10. Add a time stamp to the filename when saving. You can use the datetime library to complete this task.

Deploy and recover the camera trap

1. Find a location where you are likely to encounter endotherms, but where the trap is not likely to be damaged.
2. Estimate the likely deployment time using your knowledge of DC circuits and the battery specifications. Assume the Pi draws 250 mA of power when it is on.
3. Launch your python script from the command line. Use ‘&’ to have the script run in the background, even after you log out:

   ```
   $ sudo python /home/pi/picamtrap.py &
   ```

4. **Note:** You can also add this line to /etc/rc.local to have the program execute on boot of the Pi.
5. Charge the battery or plug in the external power supply and deploy.
6. Recover the camera trap after it has run out of battery. Download the recorded images using Putty or sftp.
7. Examine the images and curate the best ones where the subject is depicted most clearly.

What to turn in?

Turn in a .py script showing your code for the camera trap.

Make a PDF (**not a Word doc**) that includes your algorithm for the camera trap. Explain and justify the deployment location with coordinates, date/time, images, etc. Describe how/why this location was chosen and what you anticipated capturing with the camera trap. Include a selection of the best images and describe what is depicted. Rate your deployment success and what you would change during a second deployment.