



Class: Intro to IoT with Raspberry Pi!

Class outline

Duration: 3–4 hrs; Difficulty: Low

Intro

Overview of IoT and Raspberry Pi

Project description

Setting Up the Hardware

Access to the GPIO with Python

Controlling an LED, motion sensor, and
buzzer

Using Python to capture images from a USB
camera

Configuring the Backend

Raspbian and system administration basics

Running a webserver using Python and Flask

Building an API with Flask for LED control

Building a User Interface (Frontend)

Building a User Interface (UI) with HTML,
CSS, and JavaScript

Piecing Everything Together

Building a Wireless Alarm System

Future Enhancements (discussion)

Backend communication via WebSockets
Headless access to your Raspberry
PiRemote access to the system

Intro

Overview of IoT and Raspberry Pi

Project description

What is *the* Internet?

The Internet is a global system of interconnected computer networks that communicate with one another to link **billions** of devices worldwide.



Facebook's Map of World 'Friendship'

What devices are found on *the Internet*?

“interconnected computer networks...”



What is the goal *the* IoT?

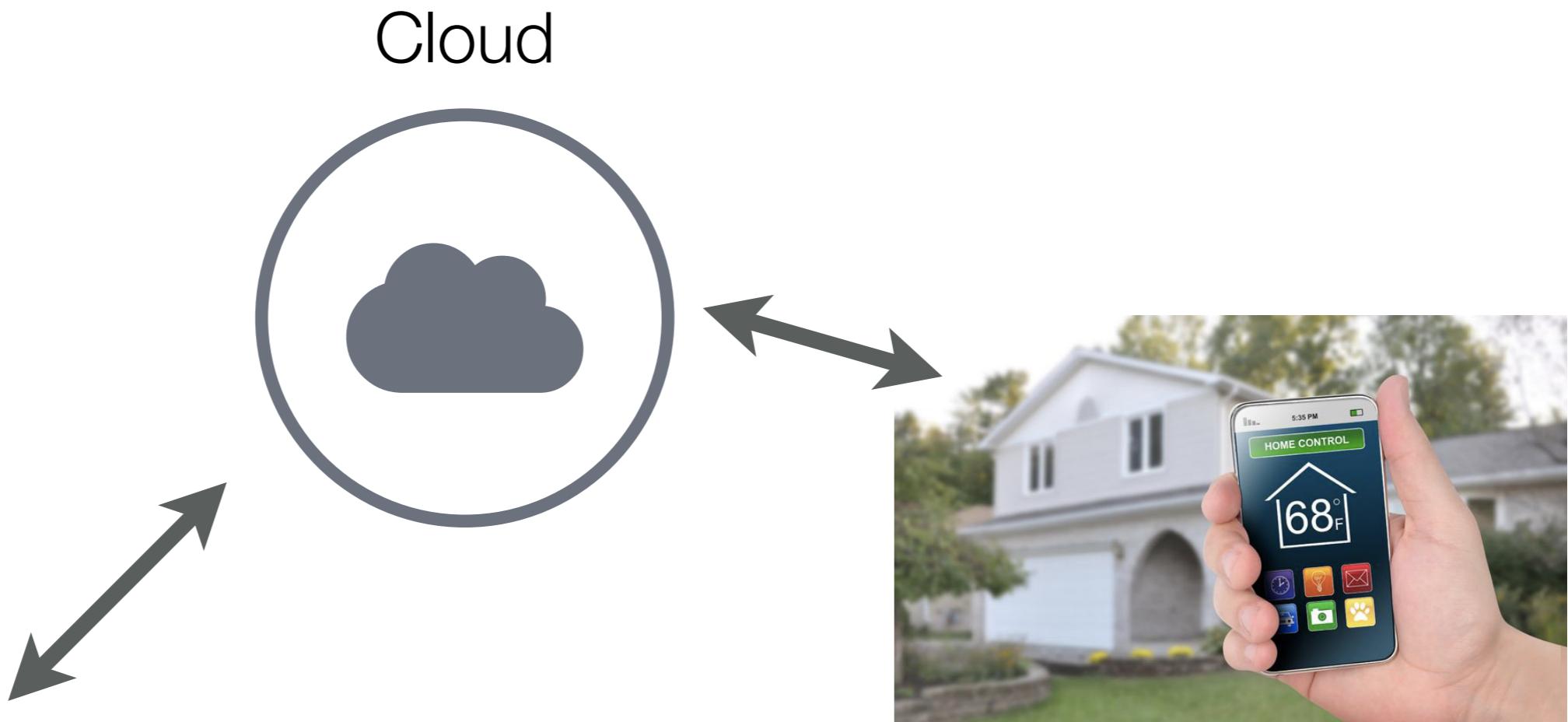
Extending the internet to include “everyday objects”.



System Overview



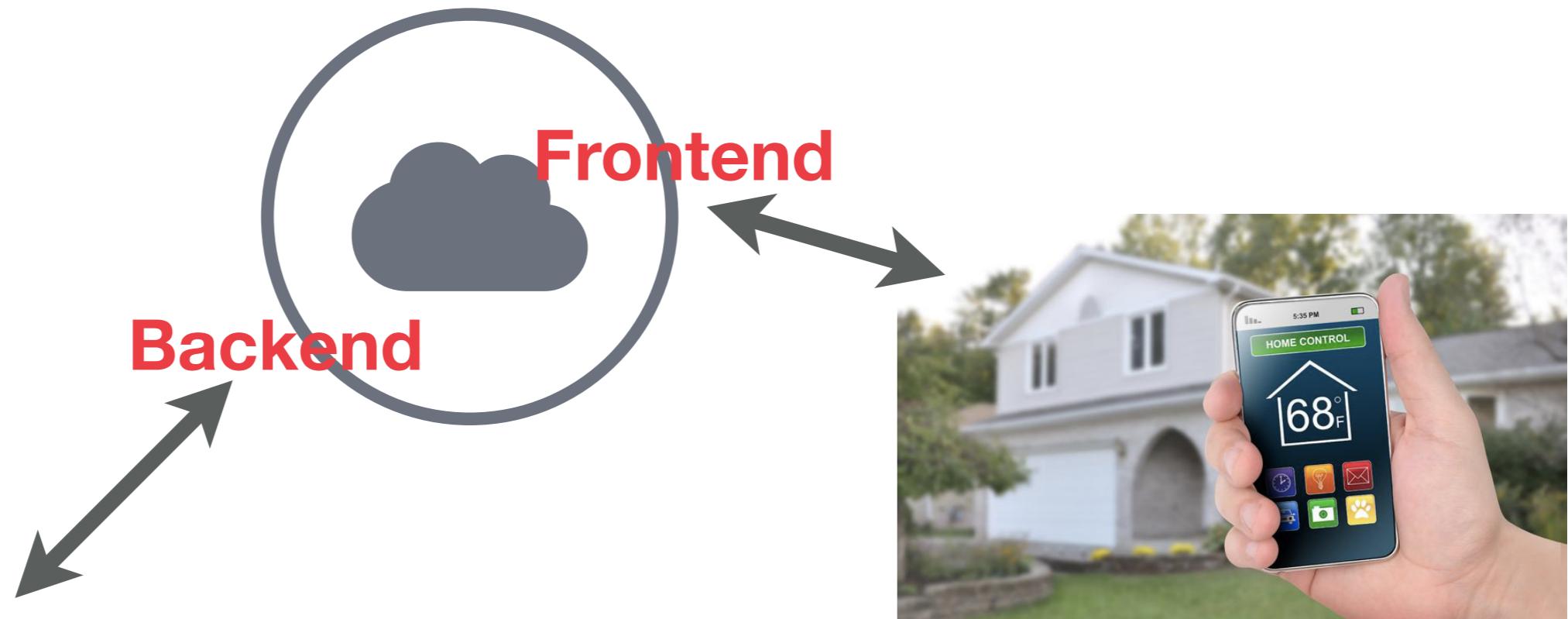
Things



Users
&
endpoints

System Overview

Electronics



Setting Up the Hardware

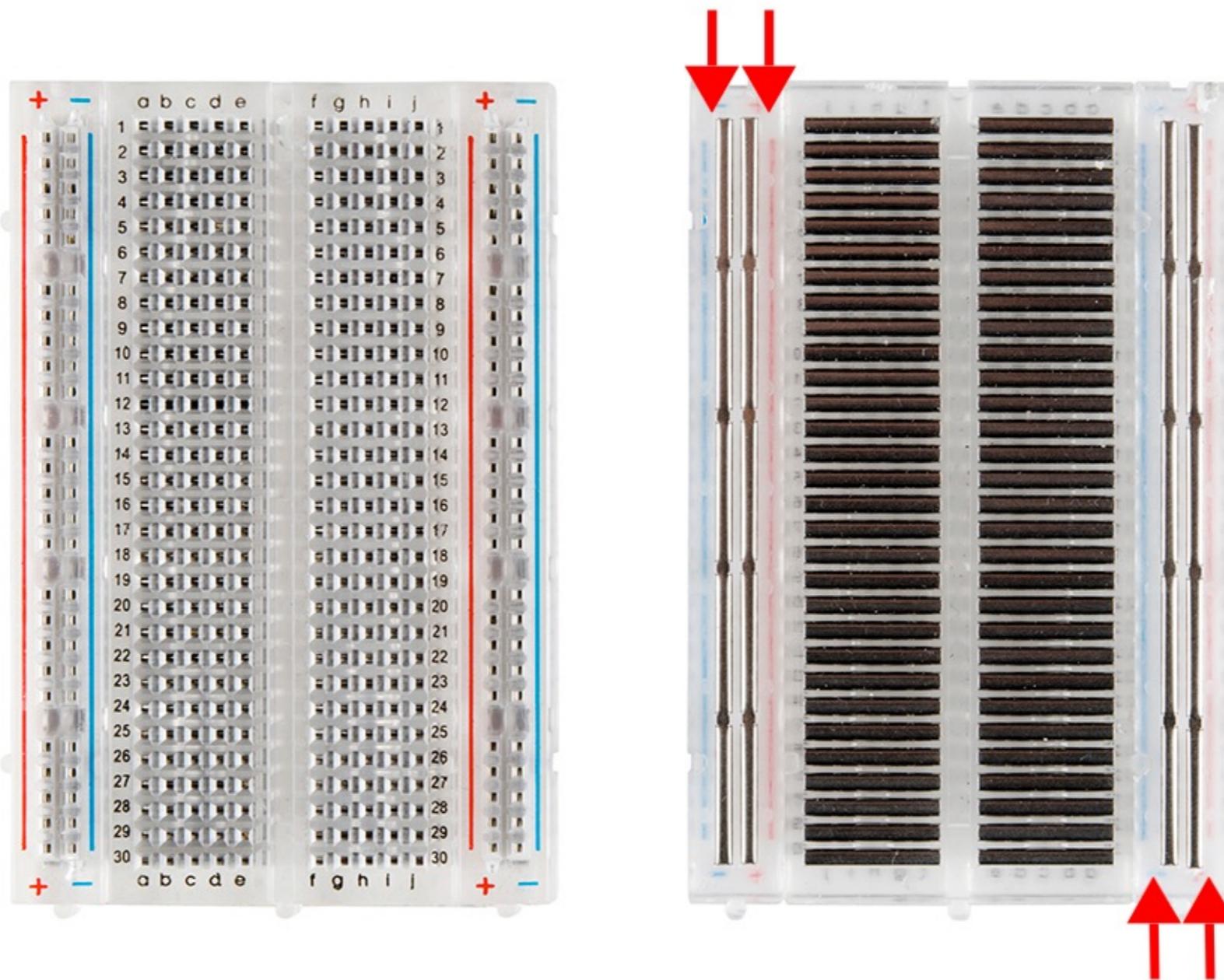
Access to the GPIO with Python

Wiring an LED and motion sensor

Using Python to control a USB camera

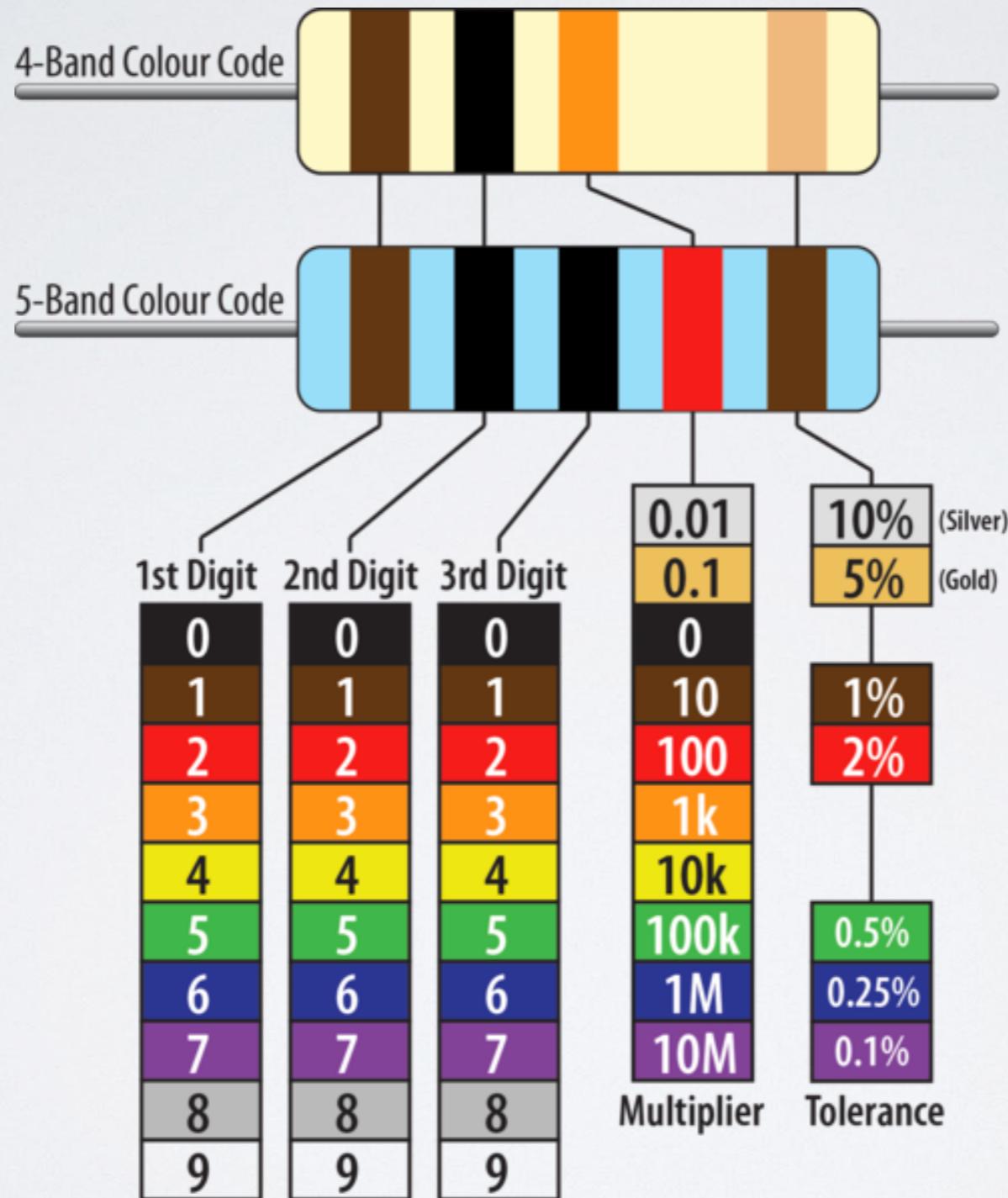
Electronic Fundamentals

How the solderless breadboard works:



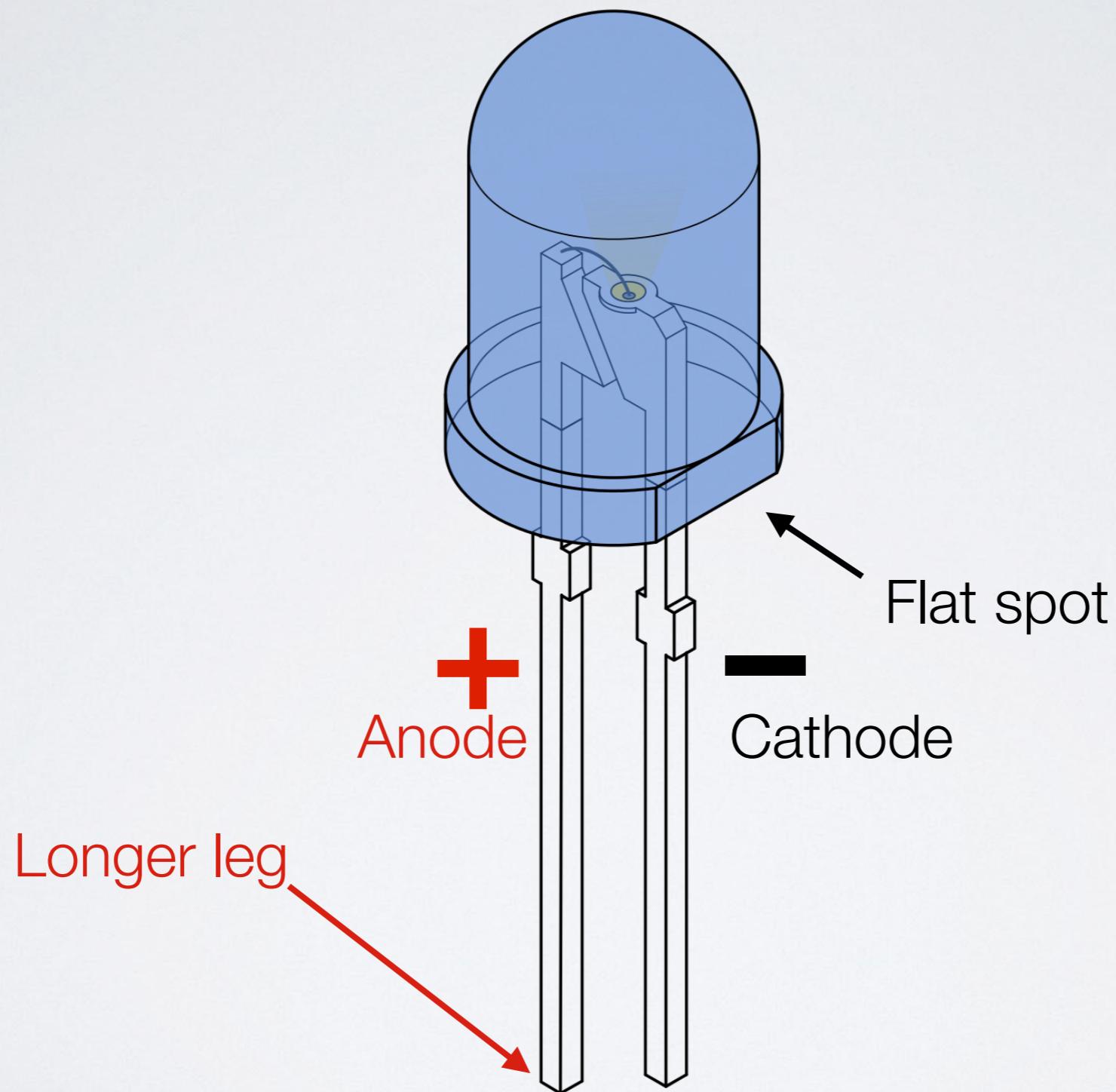
Electronic Fundamentals

Reading Resistor Codes:



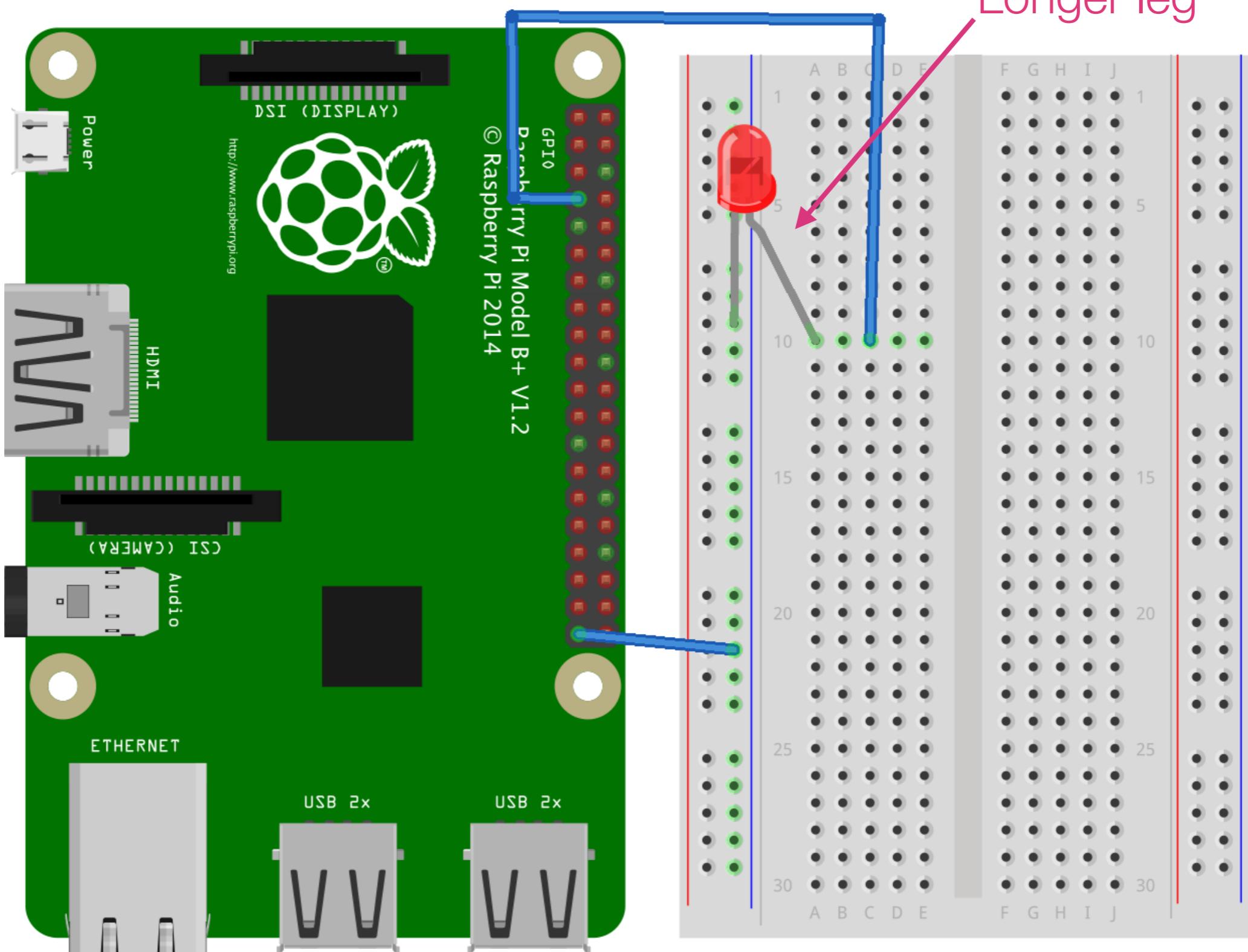
Electronic Fundamentals

Reading LED polarity:



Setting Up the Hardware

Activity 1: Blinking an LED [blink.py]

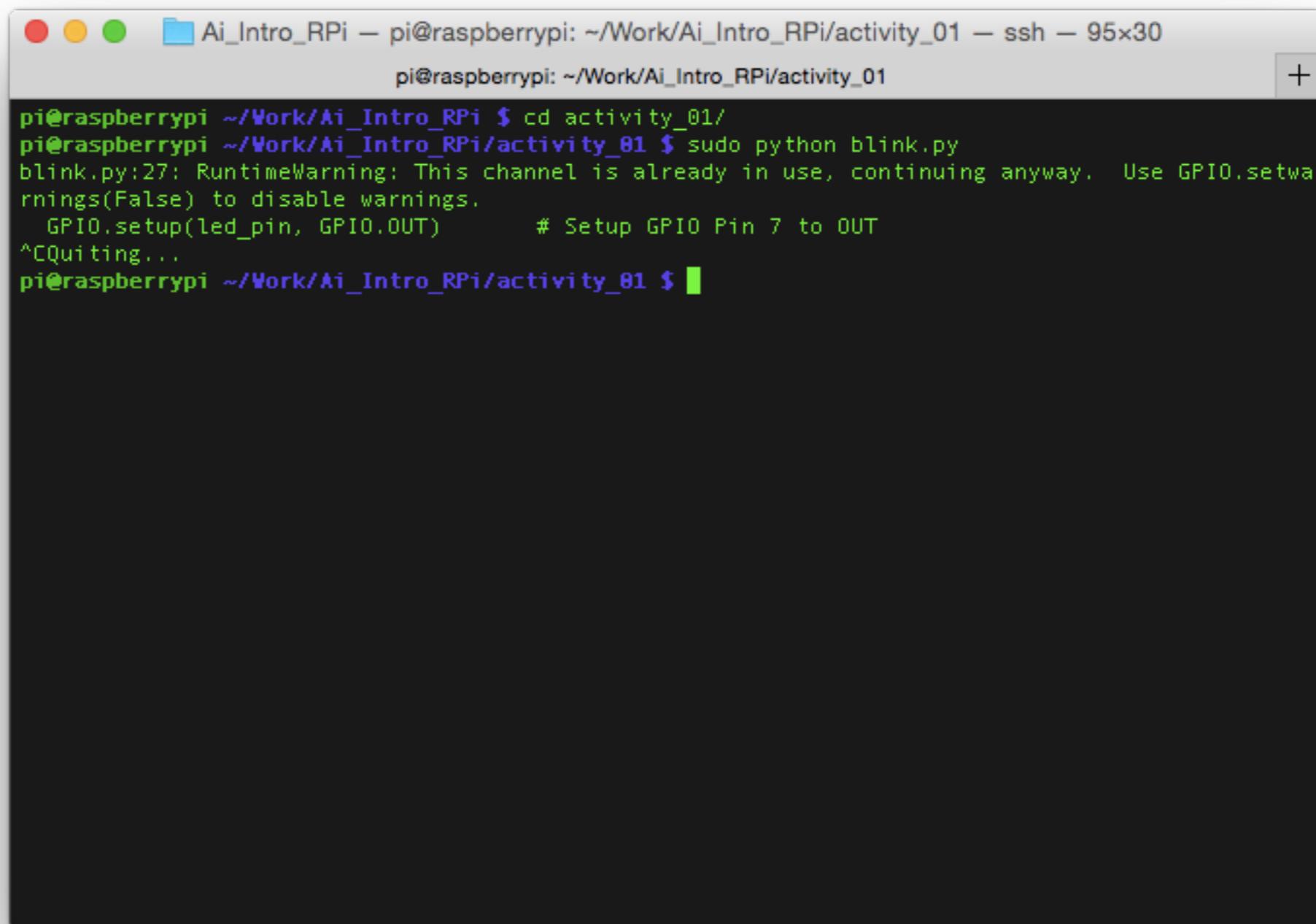


Running the Code

Activity 1: Blinking an LED [blink.py]

```
cd activity_01
```

```
sudo python blink.py
```

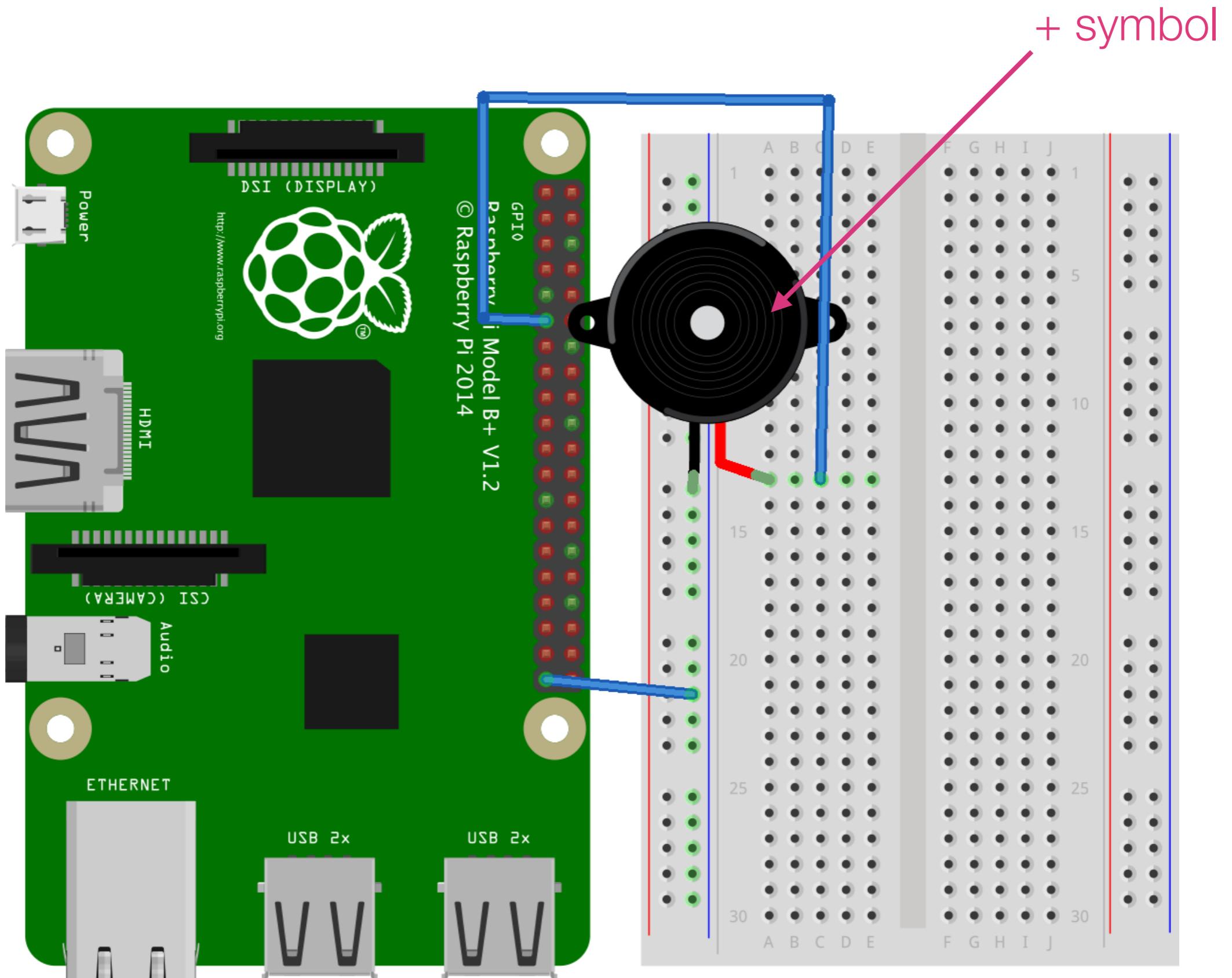


A screenshot of a terminal window titled "Ai_Intro_RPi – pi@raspberrypi: ~/Work/Ai_Intro_RPi/activity_01 – ssh – 95x30". The window shows the command line output of running the "blink.py" script. The output includes the command entered, a warning about a GPIO channel being already in use, and the final command to quit.

```
pi@raspberrypi ~$ cd activity_01/  
pi@raspberrypi ~$ sudo python blink.py  
blink.py:27: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.  
    GPIO.setup(led_pin, GPIO.OUT)      # Setup GPIO Pin 7 to OUT  
^CQuiting...  
pi@raspberrypi ~$
```

Setting Up the Hardware

Activity 2: Generating sound with PWM [sound.py]

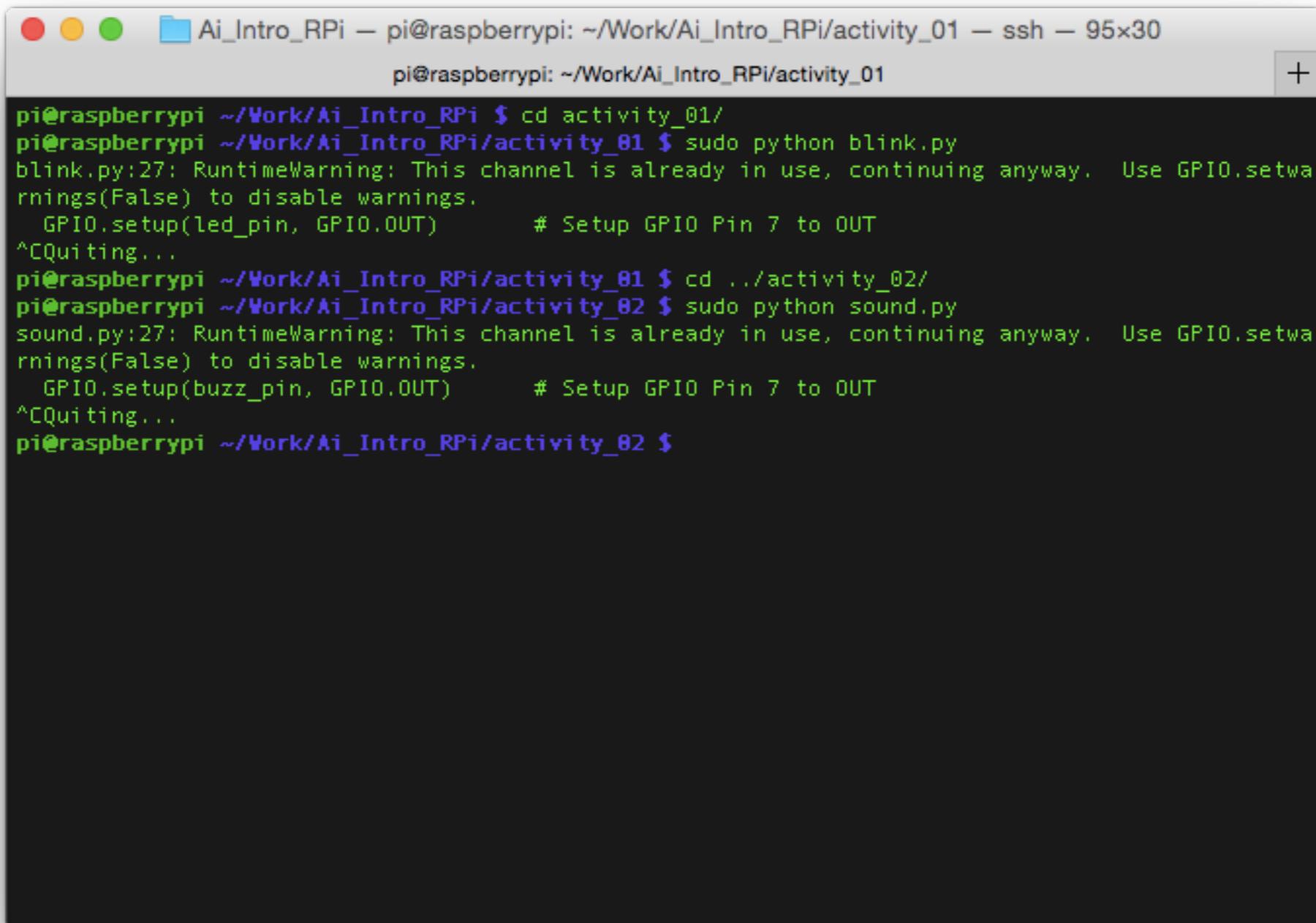


Running the Code

Activity 2: Generating sound with PWM [sound.py]

```
cd ../activity_02
```

```
sudo python sound.py
```

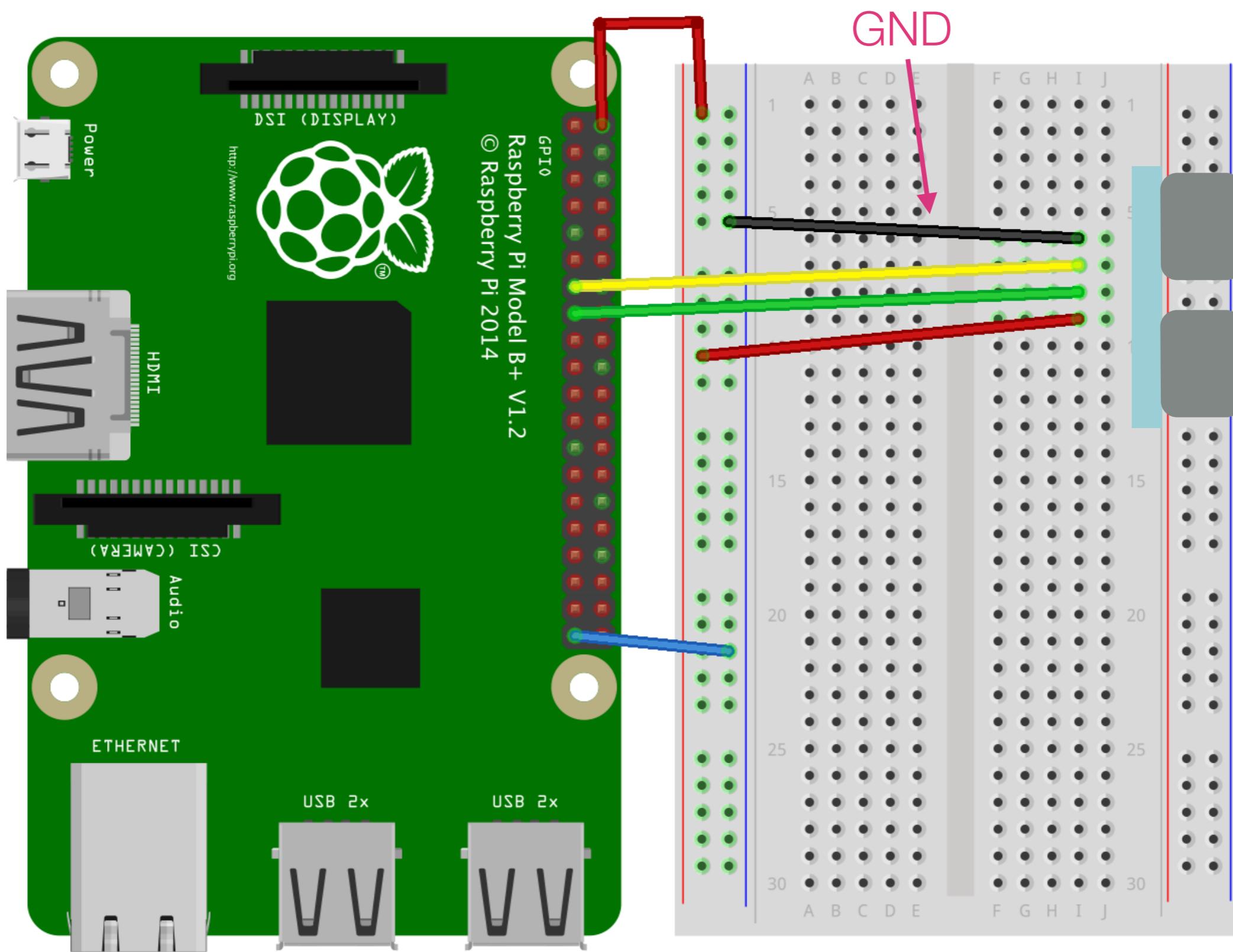


A screenshot of a terminal window titled "Ai_Intro_RPi – pi@raspberrypi: ~/Work/Ai_Intro_RPi/activity_01 – ssh – 95x30". The window shows the command line history:

```
pi@raspberrypi:~/Work/Ai_Intro_RPi$ cd activity_01/  
pi@raspberrypi:~/Work/Ai_Intro_RPi/activity_01$ sudo python blink.py  
blink.py:27: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.  
    GPIO.setup(led_pin, GPIO.OUT)      # Setup GPIO Pin 7 to OUT  
^CQuitting...  
pi@raspberrypi:~/Work/Ai_Intro_RPi/activity_01$ cd ../activity_02/  
pi@raspberrypi:~/Work/Ai_Intro_RPi/activity_02$ sudo python sound.py  
sound.py:27: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to disable warnings.  
    GPIO.setup(buzz_pin, GPIO.OUT)      # Setup GPIO Pin 7 to OUT  
^CQuitting...  
pi@raspberrypi:~/Work/Ai_Intro_RPi/activity_02$
```

Setting Up the Hardware

Activity 3: Detecting Motion with a Ultrasonic sensor [ultrasonic.py]

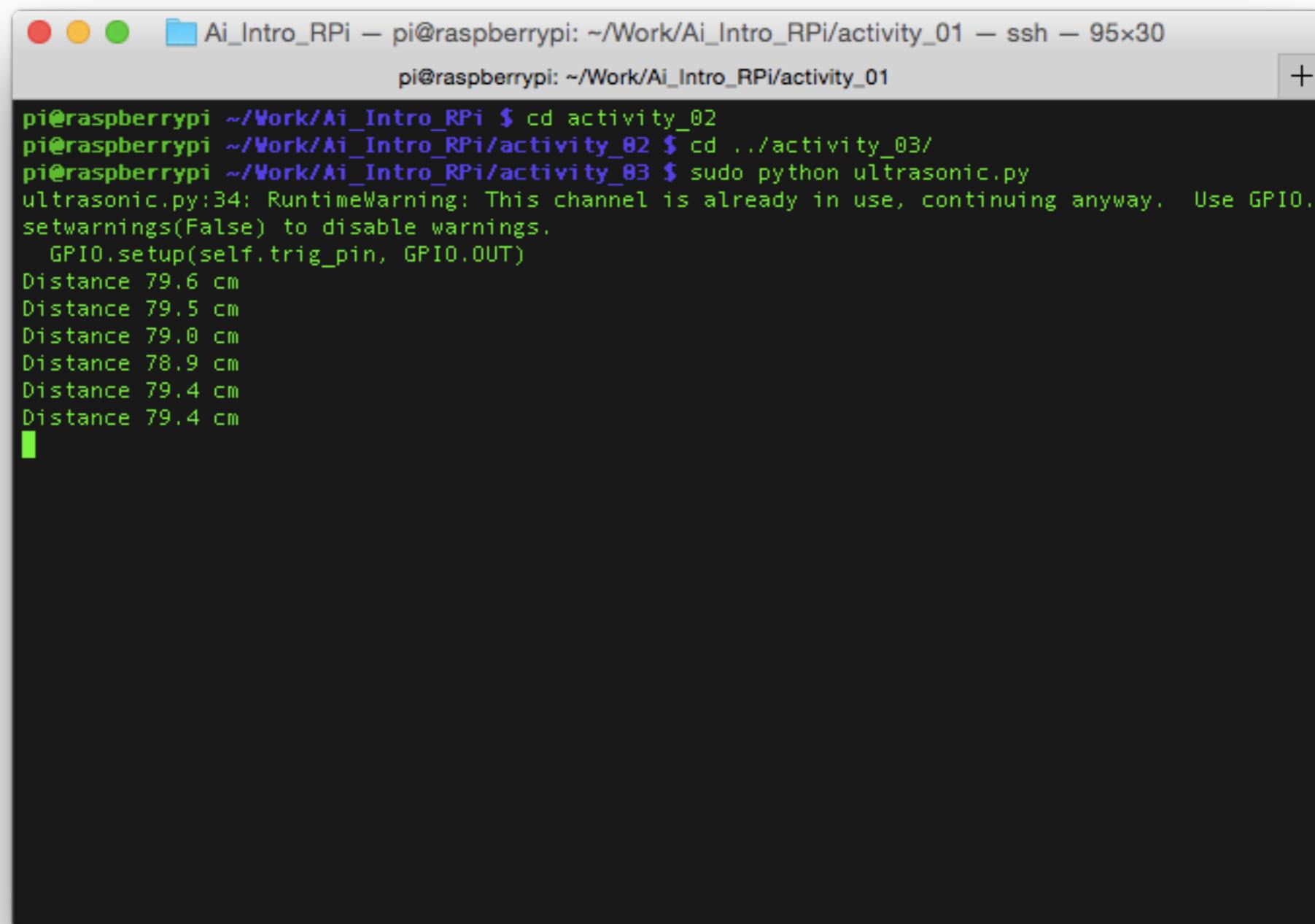


Running the Code

Activity 3: Detecting Motion with a Ultrasonic sensor [ultrasonic.py]

```
cd ../activity_03
```

```
sudo python ultrasonic.py
```



A screenshot of a terminal window titled "Ai_Intro_RPi – pi@raspberrypi: ~/Work/Ai_Intro_RPi/activity_01 – ssh – 95x30". The window shows the command line history and the execution of the "ultrasonic.py" script. The output indicates the distance measured by the ultrasonic sensor is consistently around 79.4 cm.

```
pi@raspberrypi ~$ cd activity_02  
pi@raspberrypi ~$ cd ..../activity_03/  
pi@raspberrypi ~$ sudo python ultrasonic.py  
ultrasonic.py:34: RuntimeWarning: This channel is already in use, continuing anyway. Use GPIO.  
setwarnings(False) to disable warnings.  
    GPIO.setup(self.trig_pin, GPIO.OUT)  
Distance 79.6 cm  
Distance 79.5 cm  
Distance 79.0 cm  
Distance 78.9 cm  
Distance 79.4 cm  
Distance 79.4 cm
```

Setting Up the Hardware

Activity 4: Capturing an Image from a USB camera [usb_camera.py]

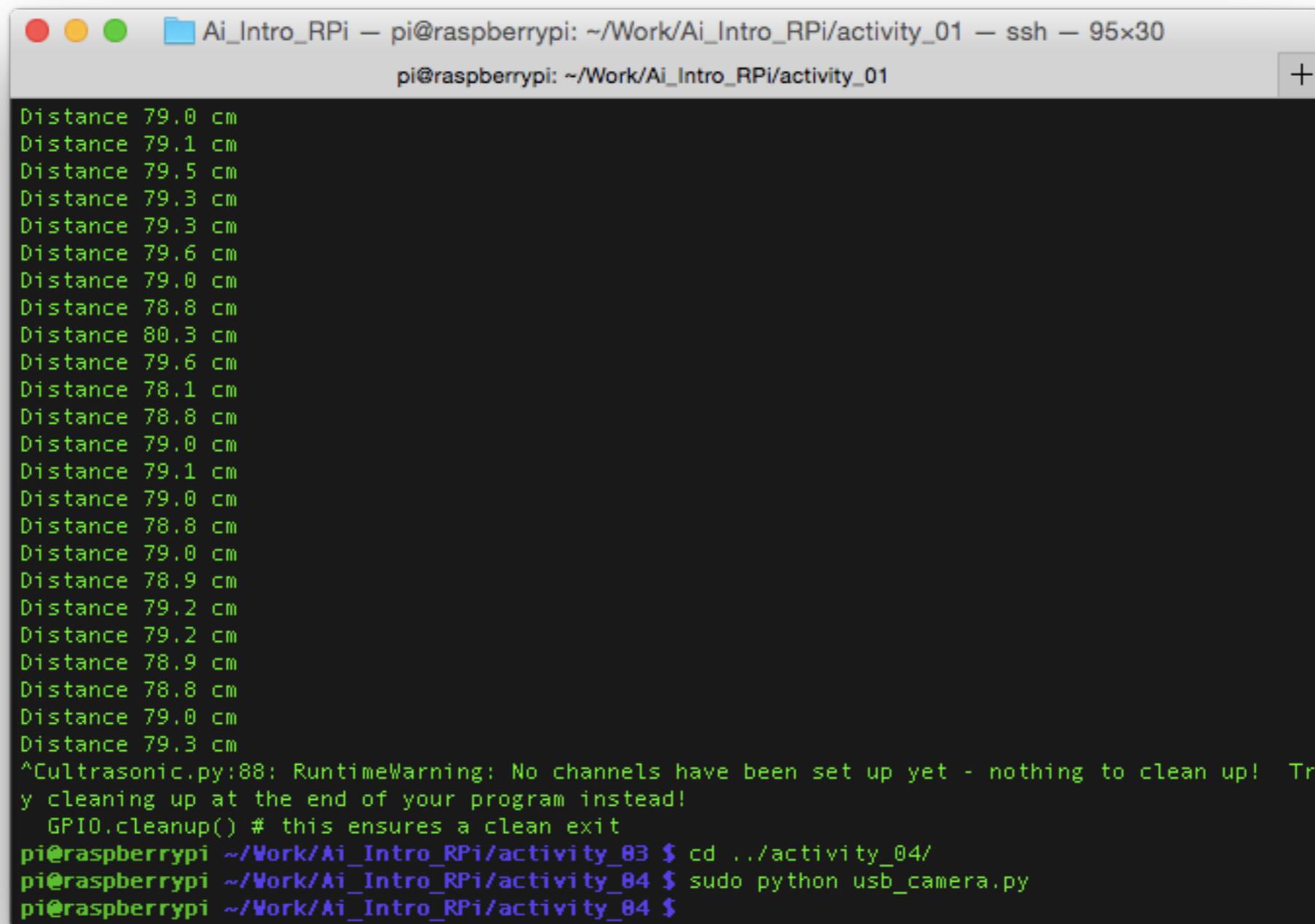
Plug in via USB :)

Running the Code

Activity 4: Capturing an Image from a USB camera [usb_camera.py]

```
cd ../activity_04
```

```
sudo python usb_camera.py
```



A screenshot of a terminal window titled "Ai_Intro_RPi – pi@raspberrypi: ~/Work/Ai_Intro_RPi/activity_01 – ssh – 95x30". The window shows the command "pi@raspberrypi: ~/Work/Ai_Intro_RPi/activity_01" followed by the output of the "usb_camera.py" script. The script prints distance measurements in centimeters, ranging from 78.0 to 80.3 cm, with some values being 79.0 cm. At the bottom, there is a warning about Ultrasonic.py and a cleanup command.

```
Distance 79.0 cm
Distance 79.1 cm
Distance 79.5 cm
Distance 79.3 cm
Distance 79.3 cm
Distance 79.6 cm
Distance 79.0 cm
Distance 78.8 cm
Distance 80.3 cm
Distance 79.6 cm
Distance 78.1 cm
Distance 78.8 cm
Distance 79.0 cm
Distance 79.1 cm
Distance 79.0 cm
Distance 78.8 cm
Distance 79.0 cm
Distance 78.9 cm
Distance 79.2 cm
Distance 79.2 cm
Distance 78.9 cm
Distance 78.8 cm
Distance 79.0 cm
Distance 79.3 cm
^Cultrasonic.py:88: RuntimeWarning: No channels have been set up yet - nothing to clean up! Try cleaning up at the end of your program instead!
    GPIO.cleanup() # this ensures a clean exit
pi@raspberrypi ~$ cd ../activity_04/
pi@raspberrypi ~$ sudo python usb_camera.py
pi@raspberrypi ~$
```

Configuring the Backend

Raspbian and system administration basics

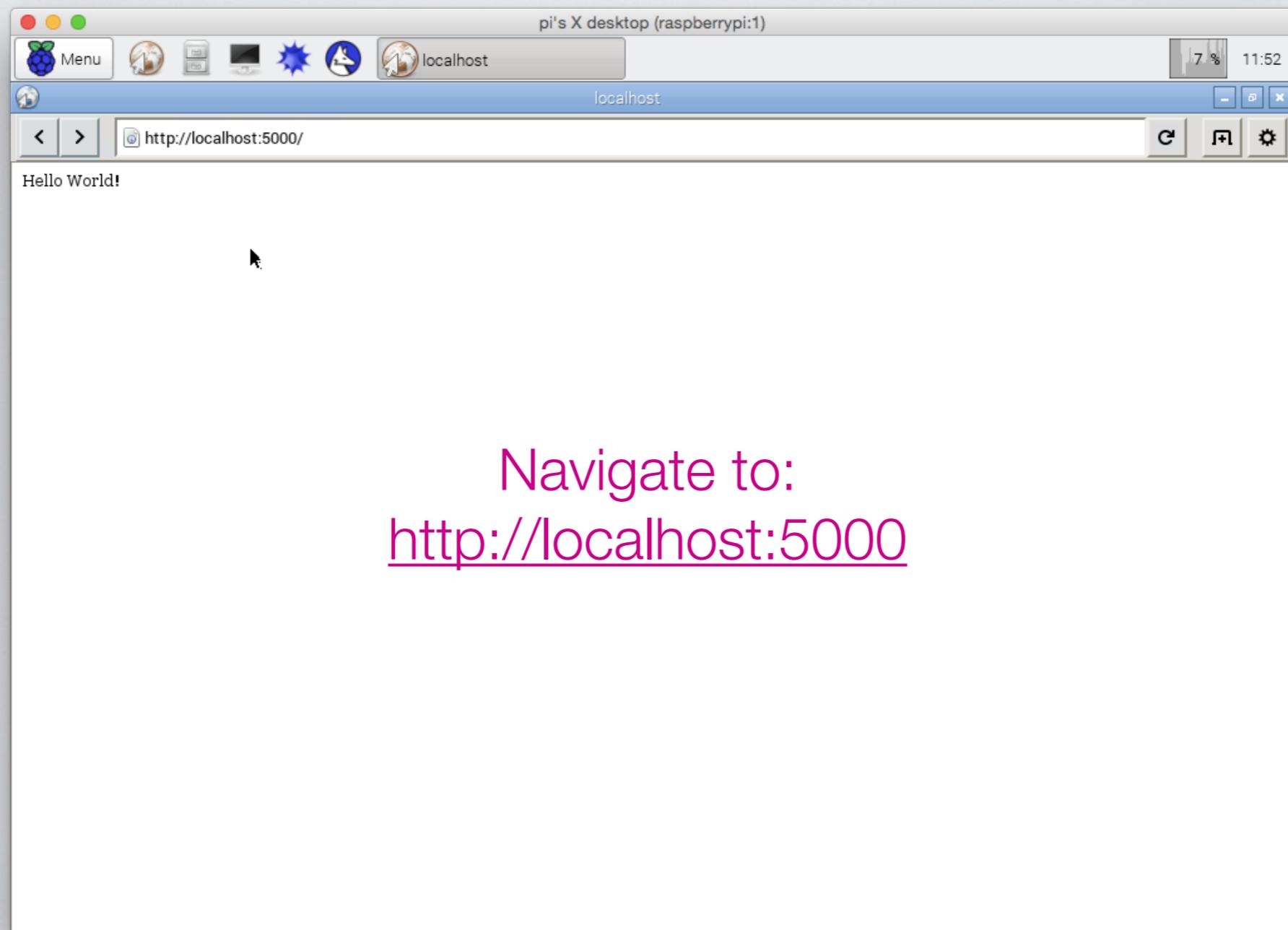
Running a webserver using Python and Flask

Configuring the Backend

Activity 5: Getting started with web frameworks [application.py]

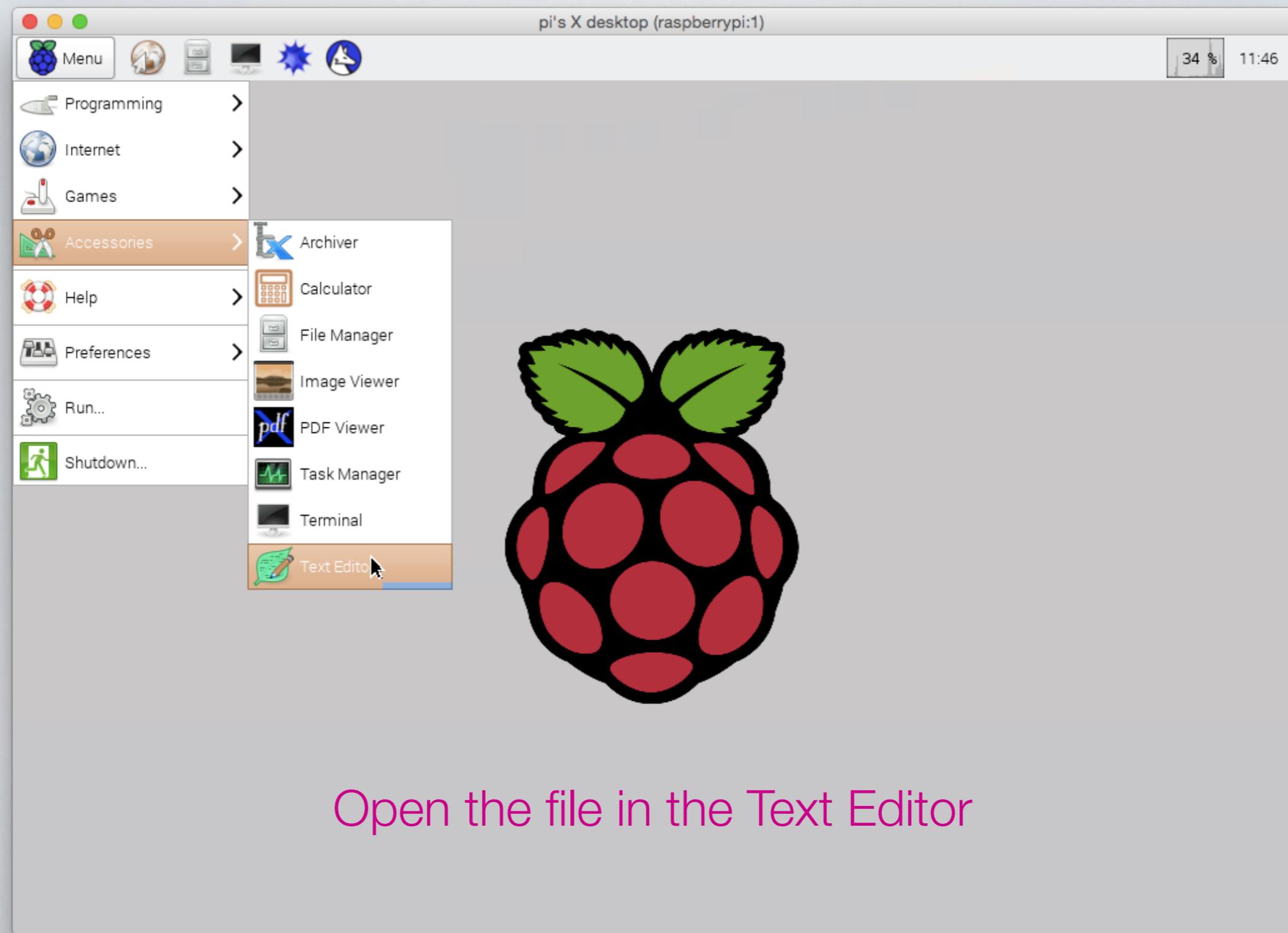
```
cd ../activity_05/simple
```

```
sudo python application.py
```



Configuring the Backend

Activity 5: Getting started with web frameworks [application.py]



Configuring the Backend

Activity 6: Building a simple API for LED control [application.py]

```
cd ../../activity_06  
sudo python application.py
```

Navigate to:

<http://localhost:5000/on>
<http://localhost:5000/off>

Building a User Interface (Frontend)

Building a User Interface (UI) with HTML, CSS, and JavaScript

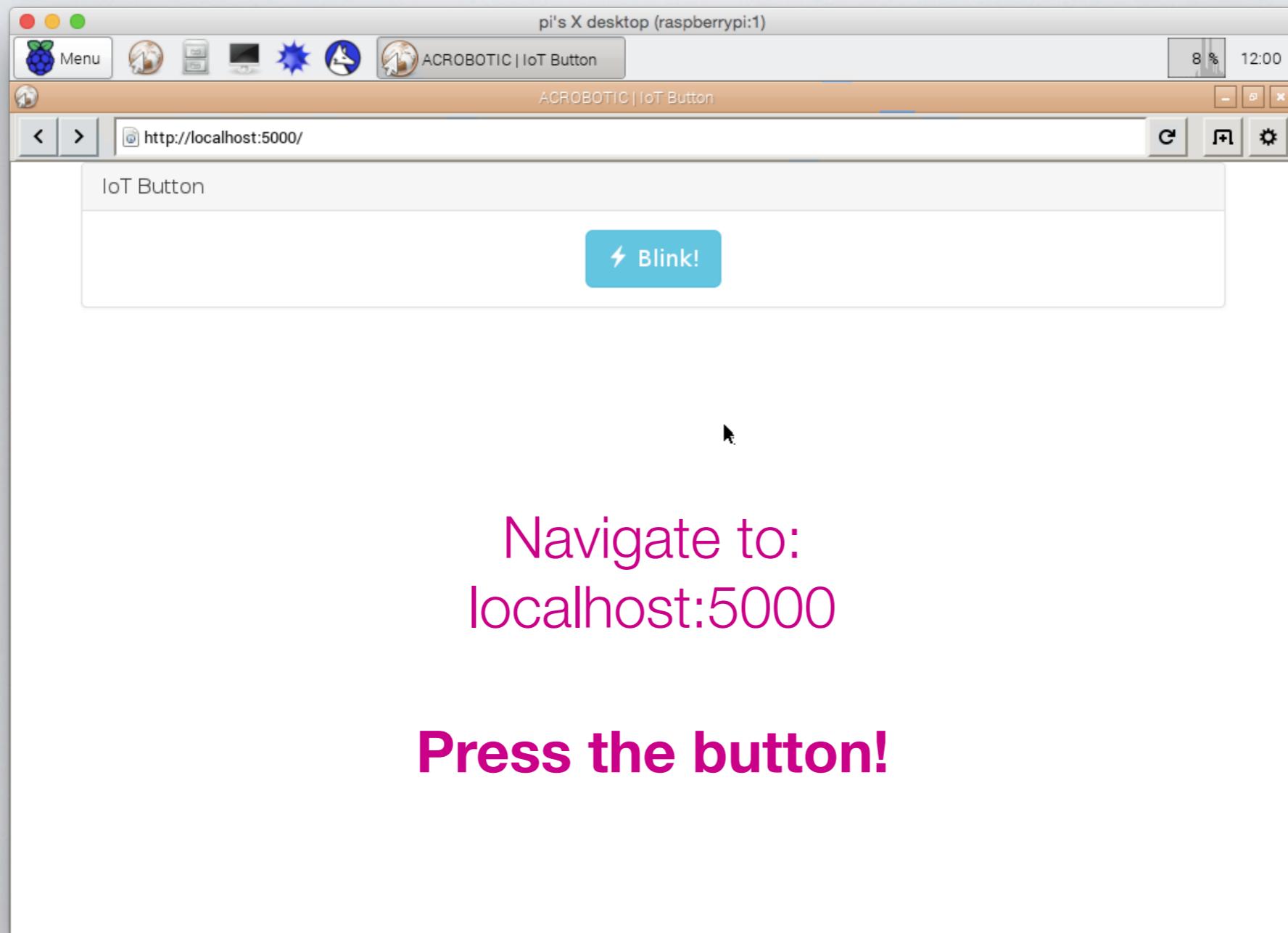
Communicating with the Backend through WebSockets

Building a User Interface

Activity 7: Creating a simple UI for LED control [application.py]

```
cd ../activity_07
```

```
sudo python application.py
```



Building a User Interface

Activity 8: Creating a simple UI for an Alarm System [application.py]

```
cd ../activity_08
```

```
sudo python application.py
```

