



Class: Intro to IoT with Raspberry Pi!

Downloading this presentation

This repository Search Pull requests Issues Gist

acrobotic / Ai_Intro_RPi Navigate to: https://github.com/acrobotic/Ai_Intro_RPi

Unwatch 3 Star 1 Fork 0

Code, wiring diagrams, and presentation slides for introductory activities on the Raspberry Pi single-board computer. — Edit

8 commits 1 branch 0 releases 1 contributor

Branch: master +

Activity	Status	Time
activity_01	done	14 days ago
activity_02	done	14 days ago
activity_03	done	14 days ago
activity_04	adding activities 3 and 4	14 days ago
activity_05	activities completed	14 days ago
activity_06	activities completed	14 days ago
activity_07	activities completed	14 days ago
activity_08	done	14 days ago
presentation	.	14 days ago

Code Issues Pull requests Wiki Pulse Graphs Settings

HTTPS clone URL <https://github.com/>

You can clone with HTTPS, SSH, or Subversion. ®

Clone in Desktop Download ZIP

Class outline

Duration: 3–4 hrs; Difficulty: Low

Intro and Fundamental Concepts

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?

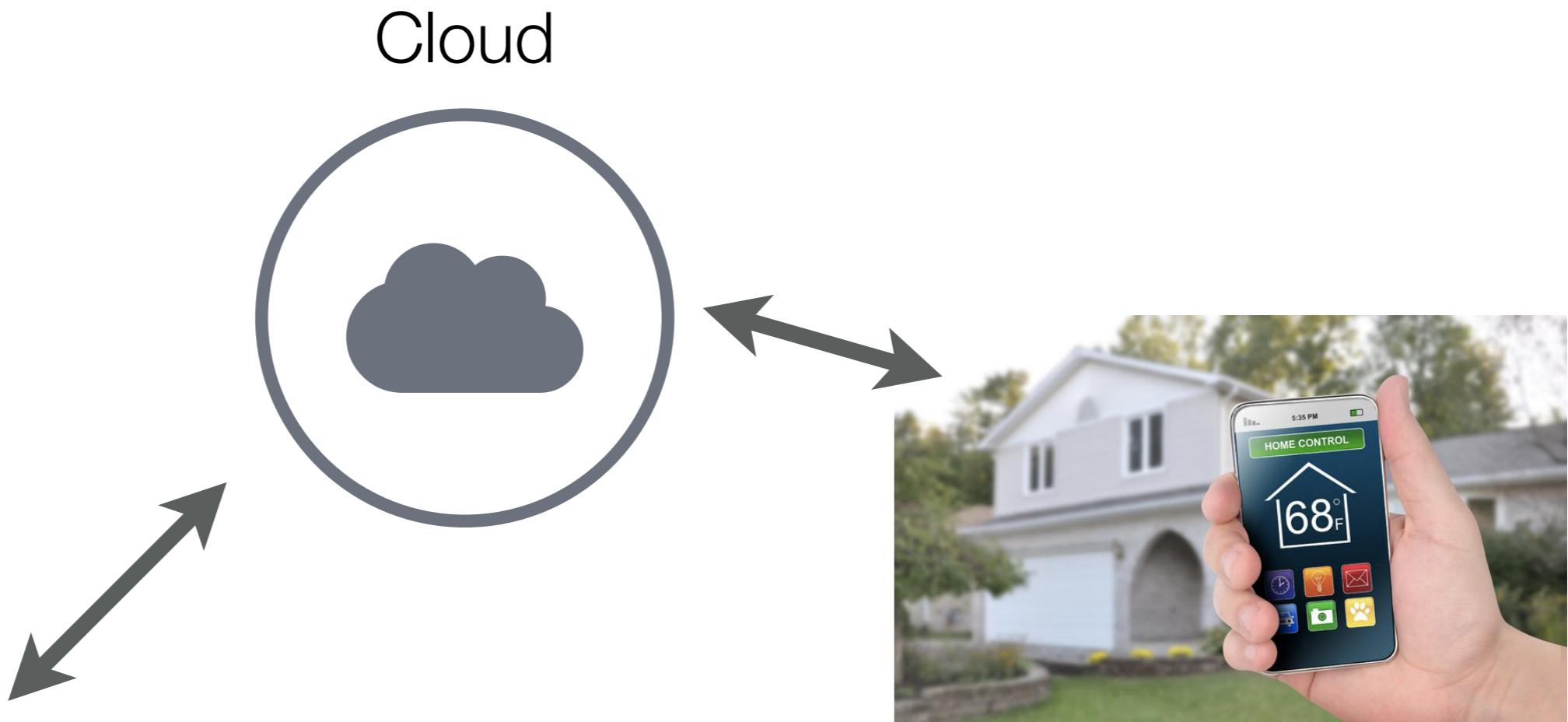
Extending the internet to include “everyday objects”.



System Overview



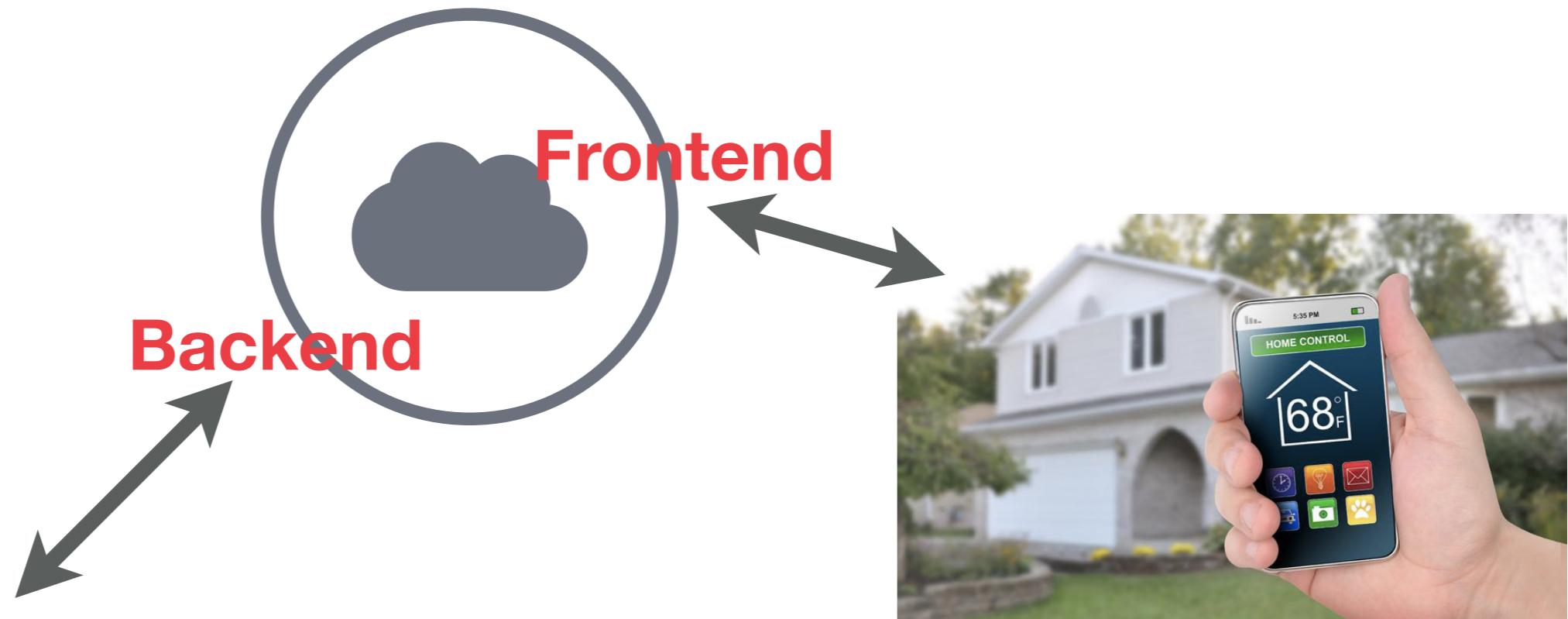
Things



Users
&
endpoints

System Overview

Electronics



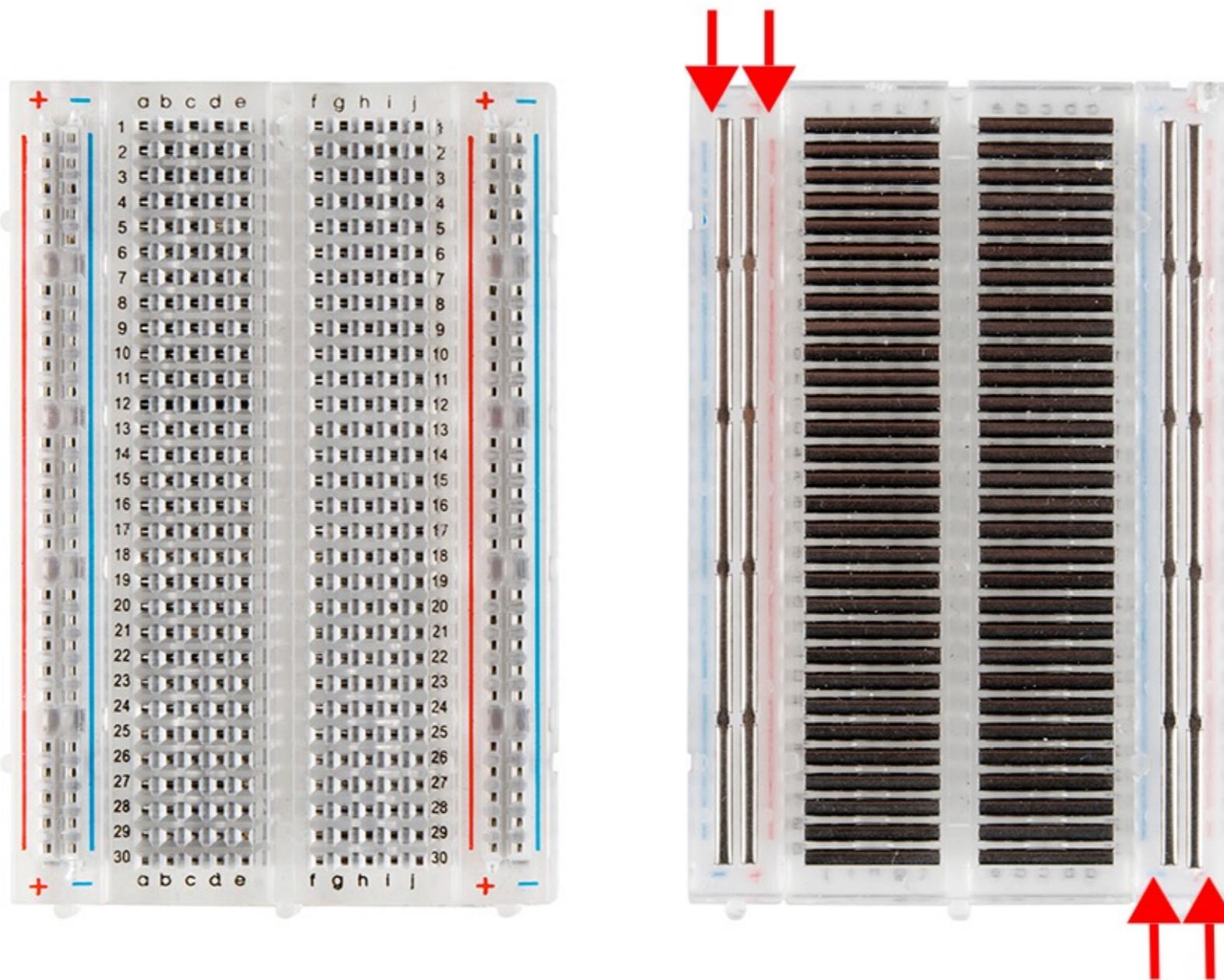
Fundamental Concepts

Hardware prototyping

Linux basic usage

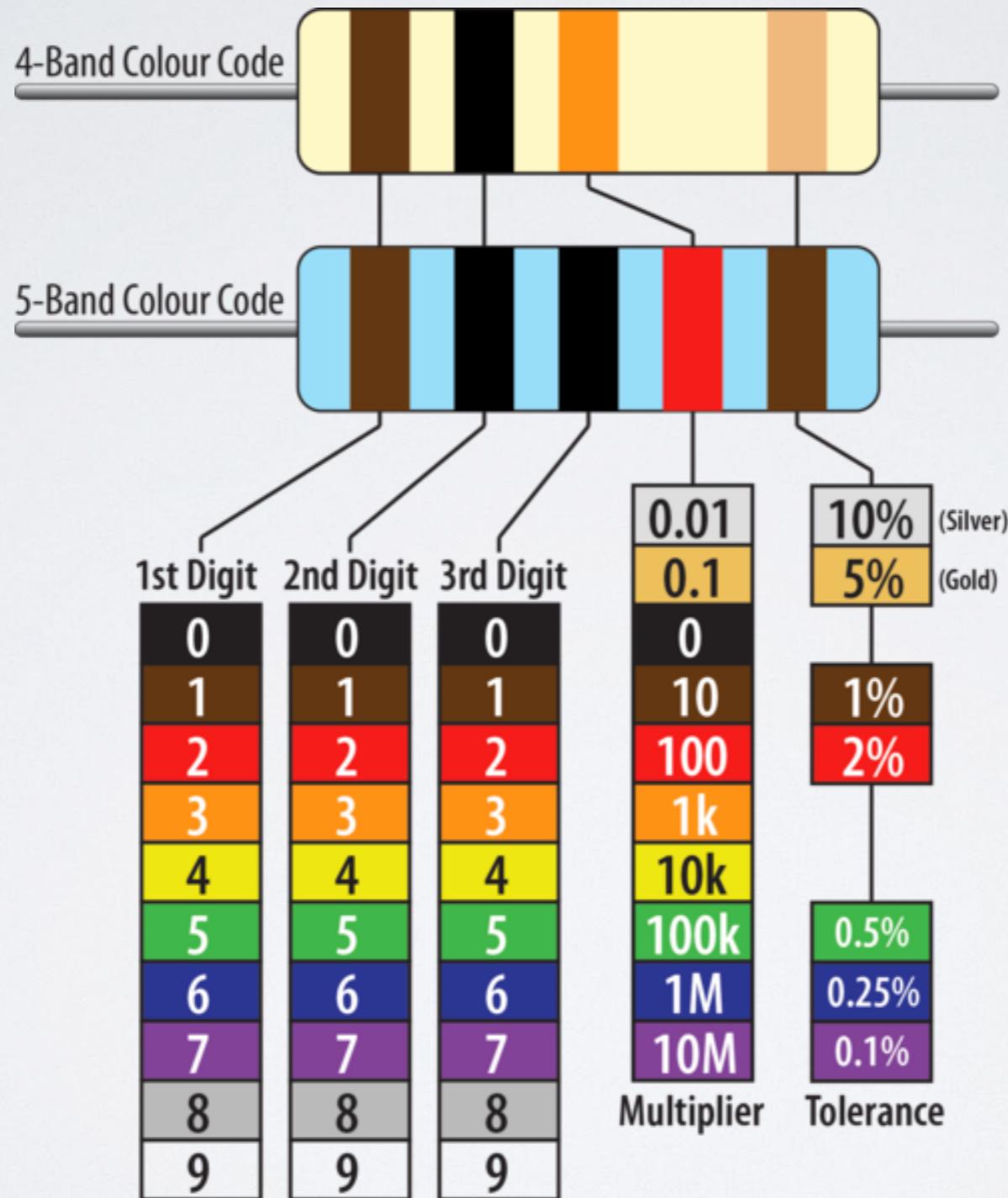
Electronic Fundamentals

How the solderless breadboard works:



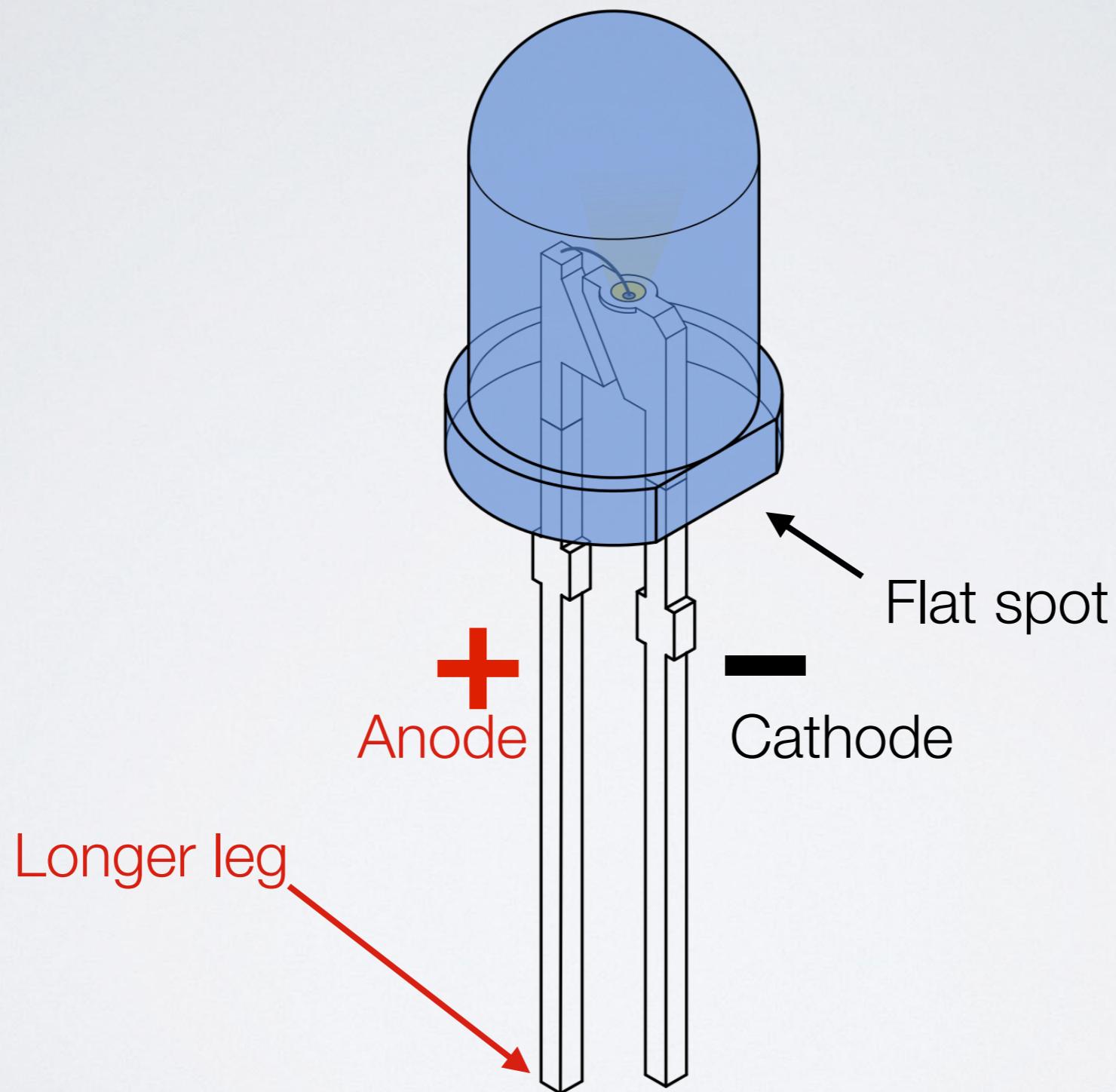
Electronic Fundamentals

Reading Resistor Codes:



Electronic Fundamentals

Reading LED polarity:



Raspberry Pi Fundamentals

Getting Started with Raspberry Pi:

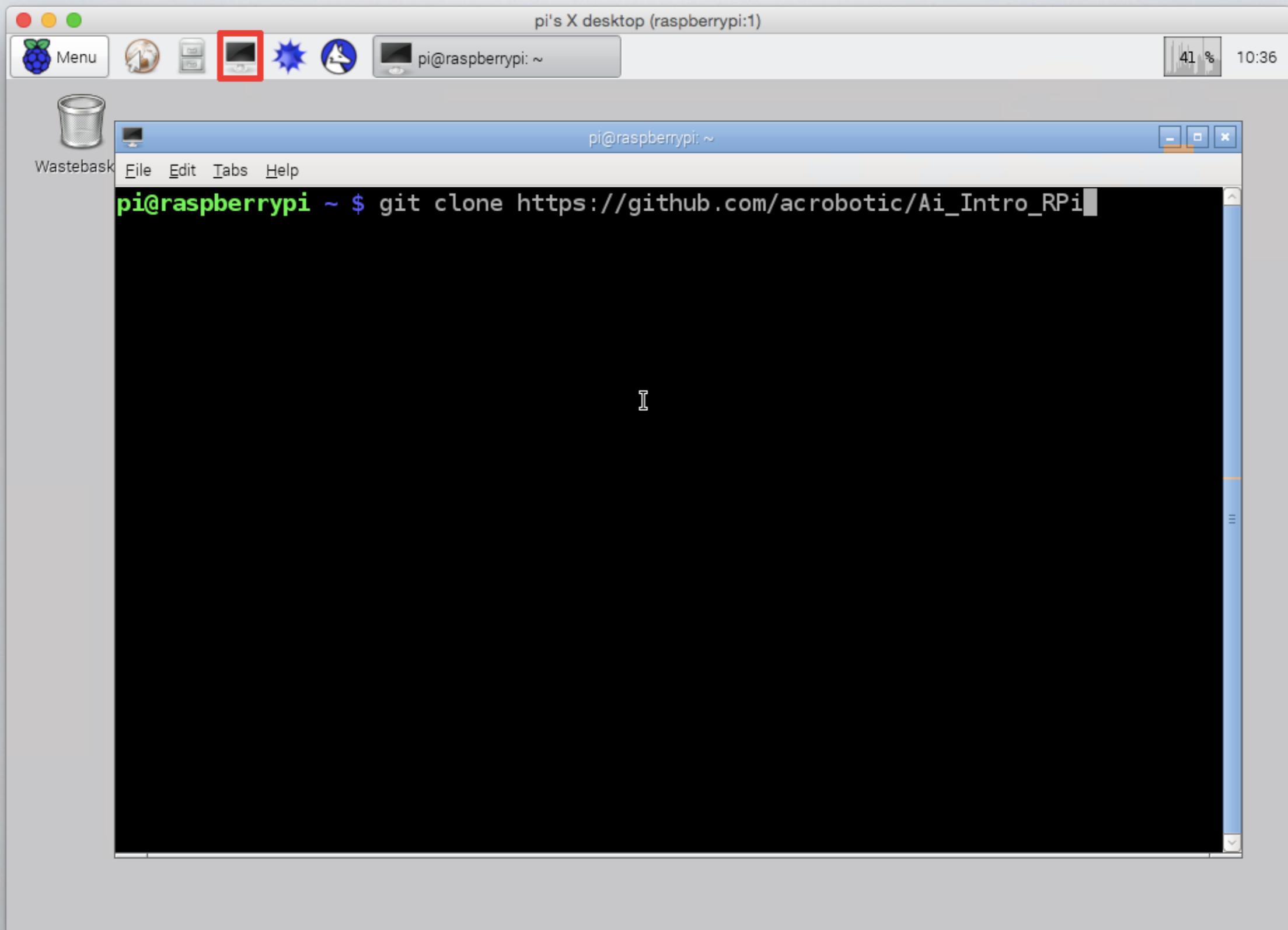


<http://learn.acrobotic.com>

<https://www.youtube.com/watch?v=ZJU7mns3juc>

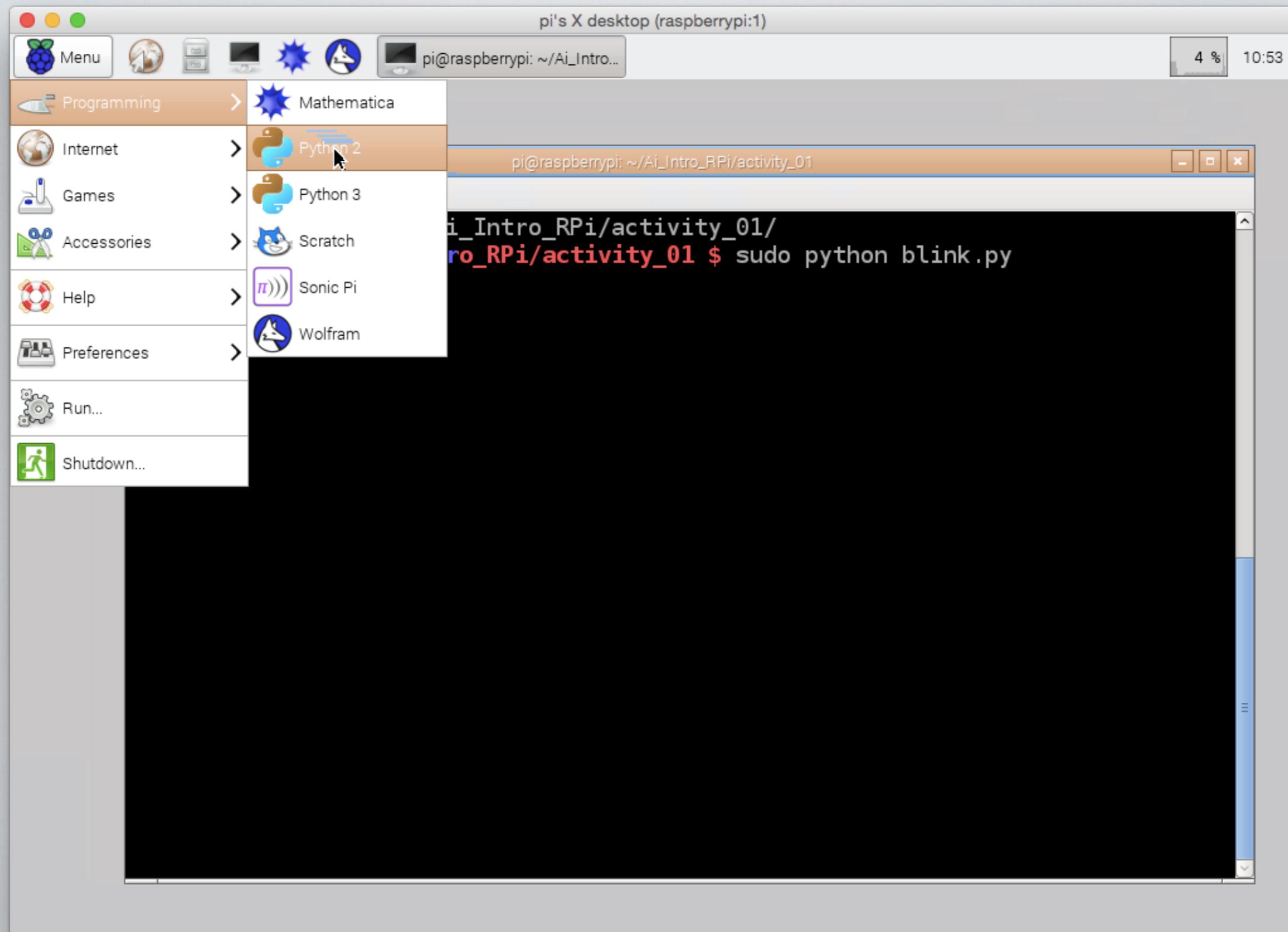
Linux Fundamentals

Accessing the Terminal and getting the activities code



Linux Fundamentals

Opening scripts in Python's “Integrated Development Environment”



Setting Up the Hardware

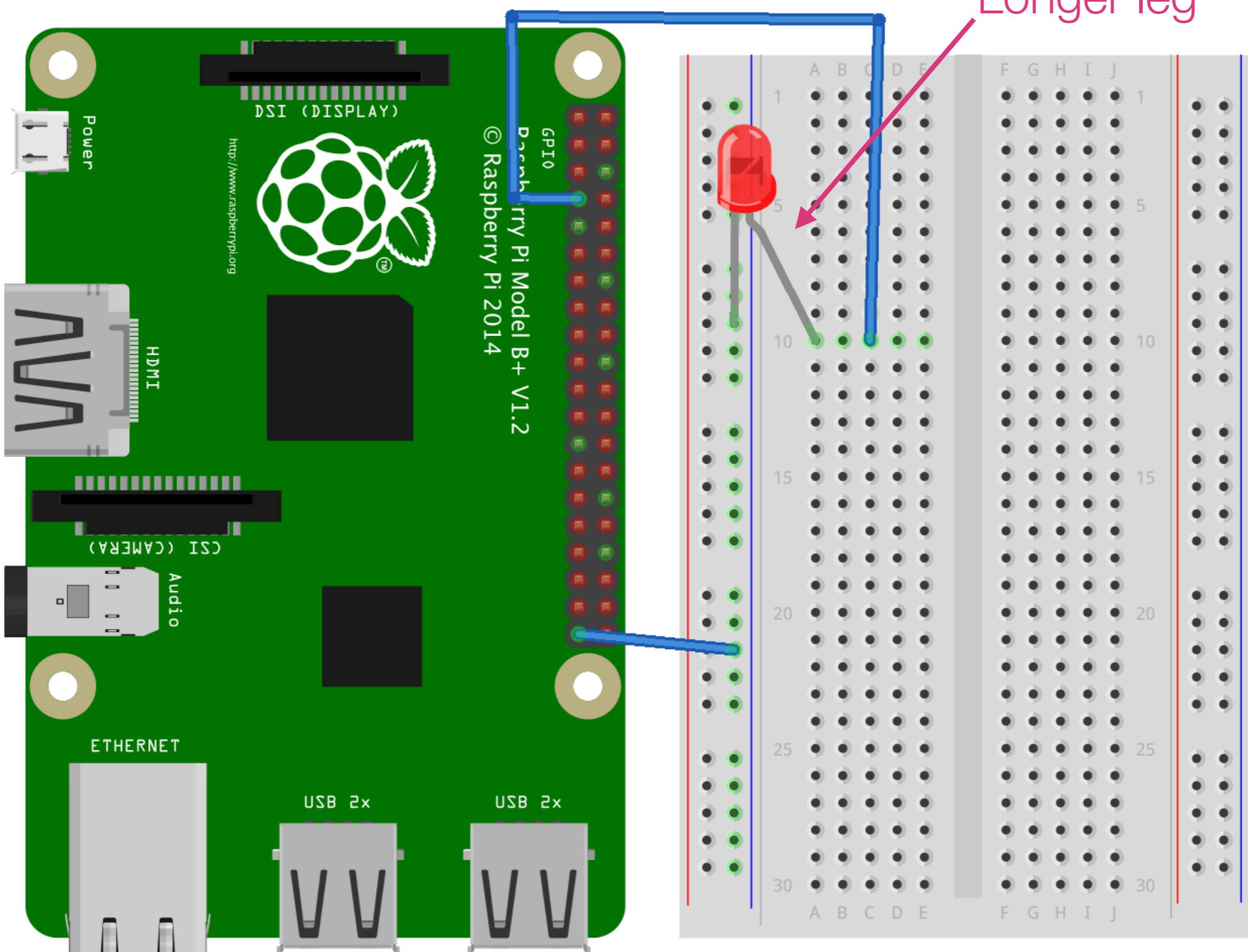
Access to the GPIO with Python

Wiring an LED and motion sensor

Using Python to control a USB camera

Setting Up the Hardware

Activity 1: Blinking an LED [blink.py]

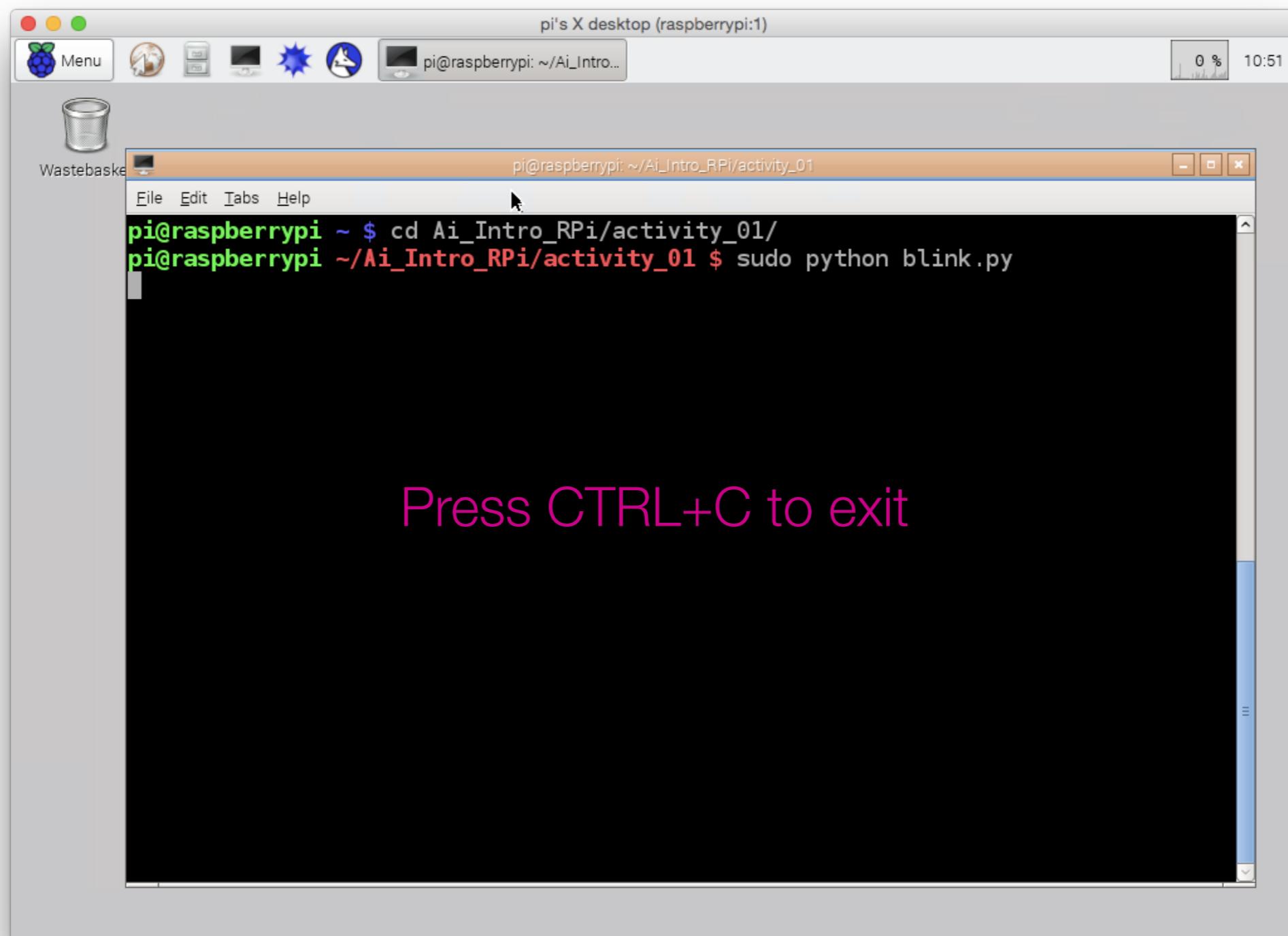


Running the Code

Activity 1: Blinking an LED [blink.py]

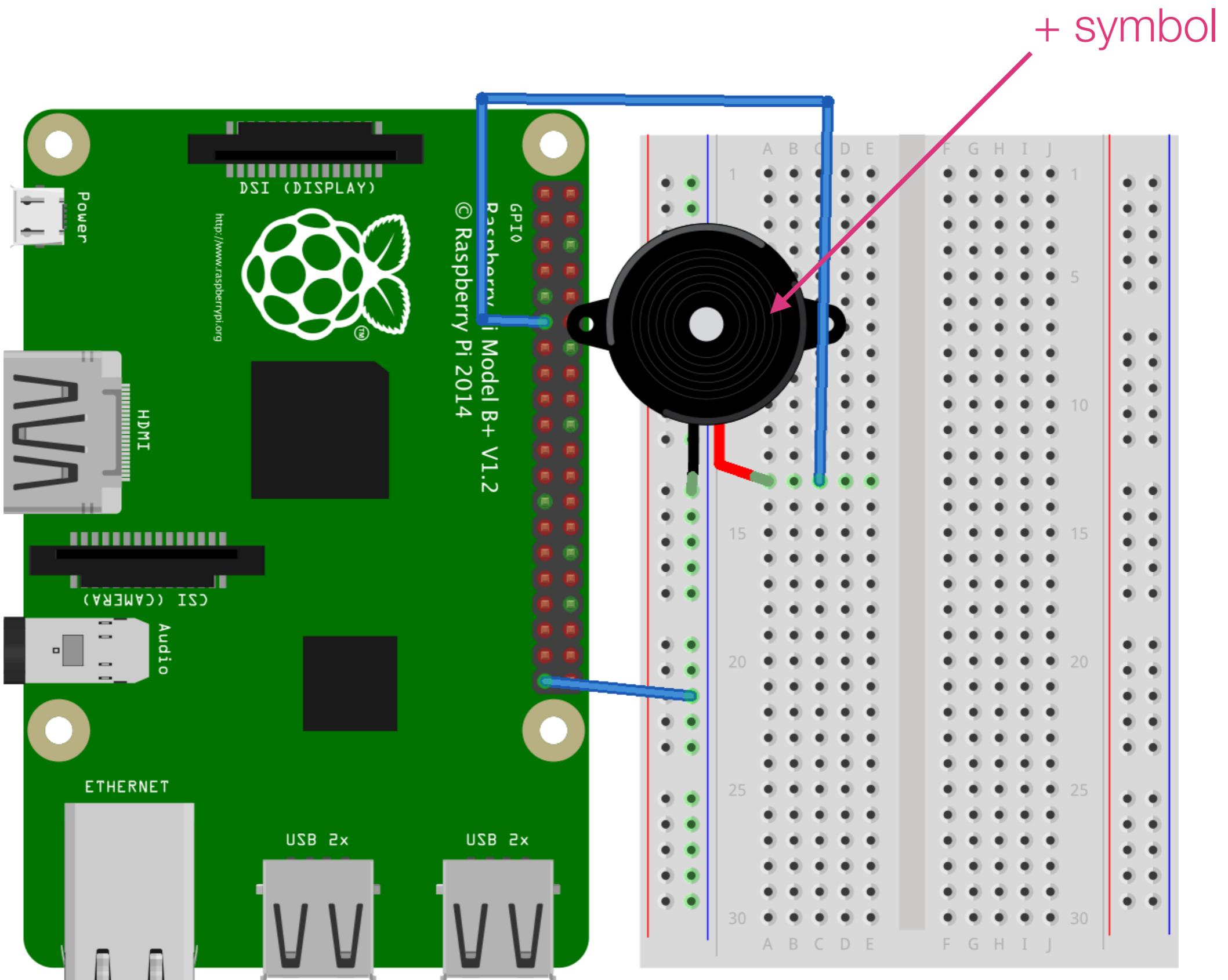
```
cd Ai_Intro_Rpi/activity_01
```

```
sudo python blink.py
```



Setting Up the Hardware

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

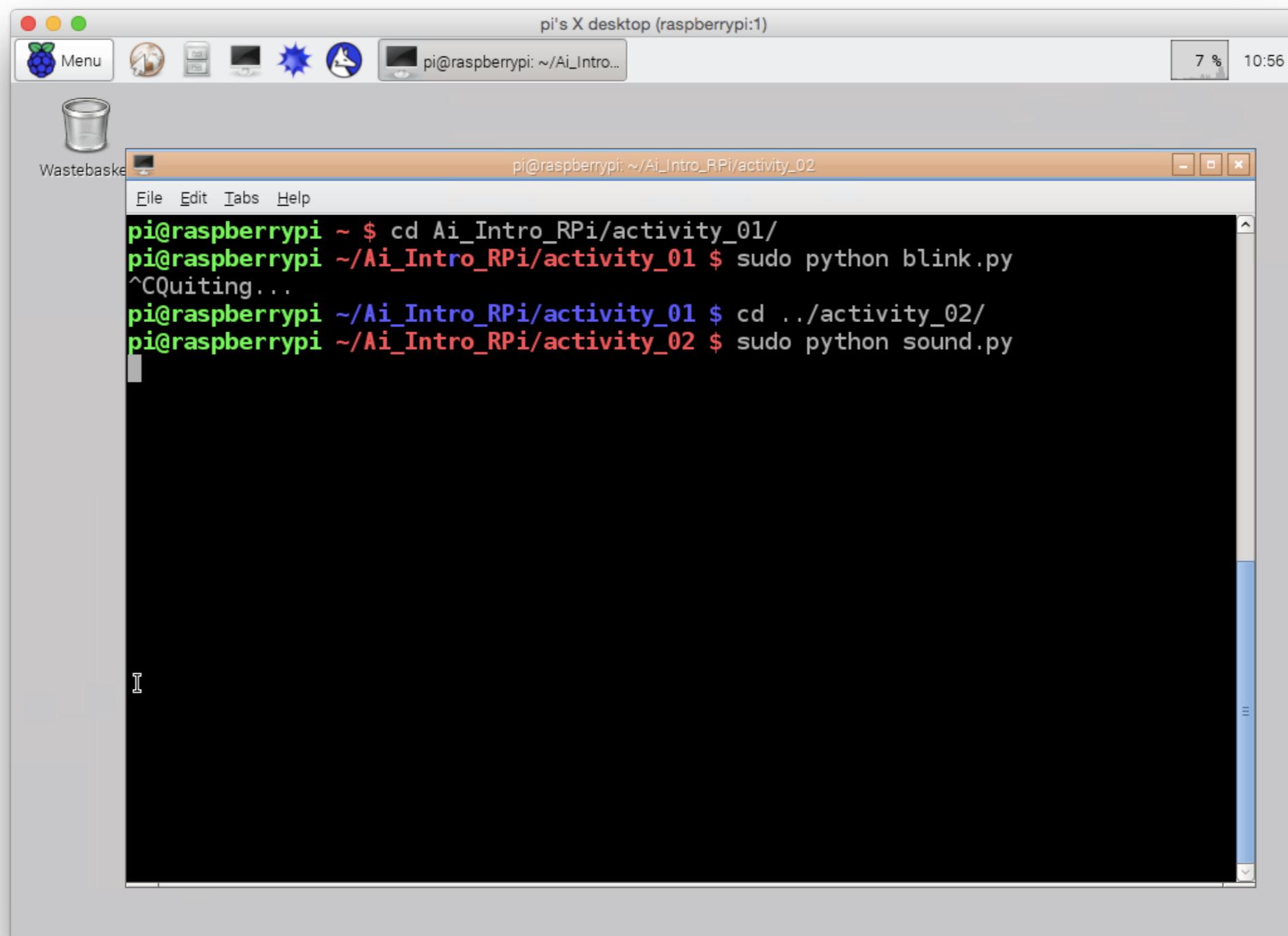


Running the Code

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

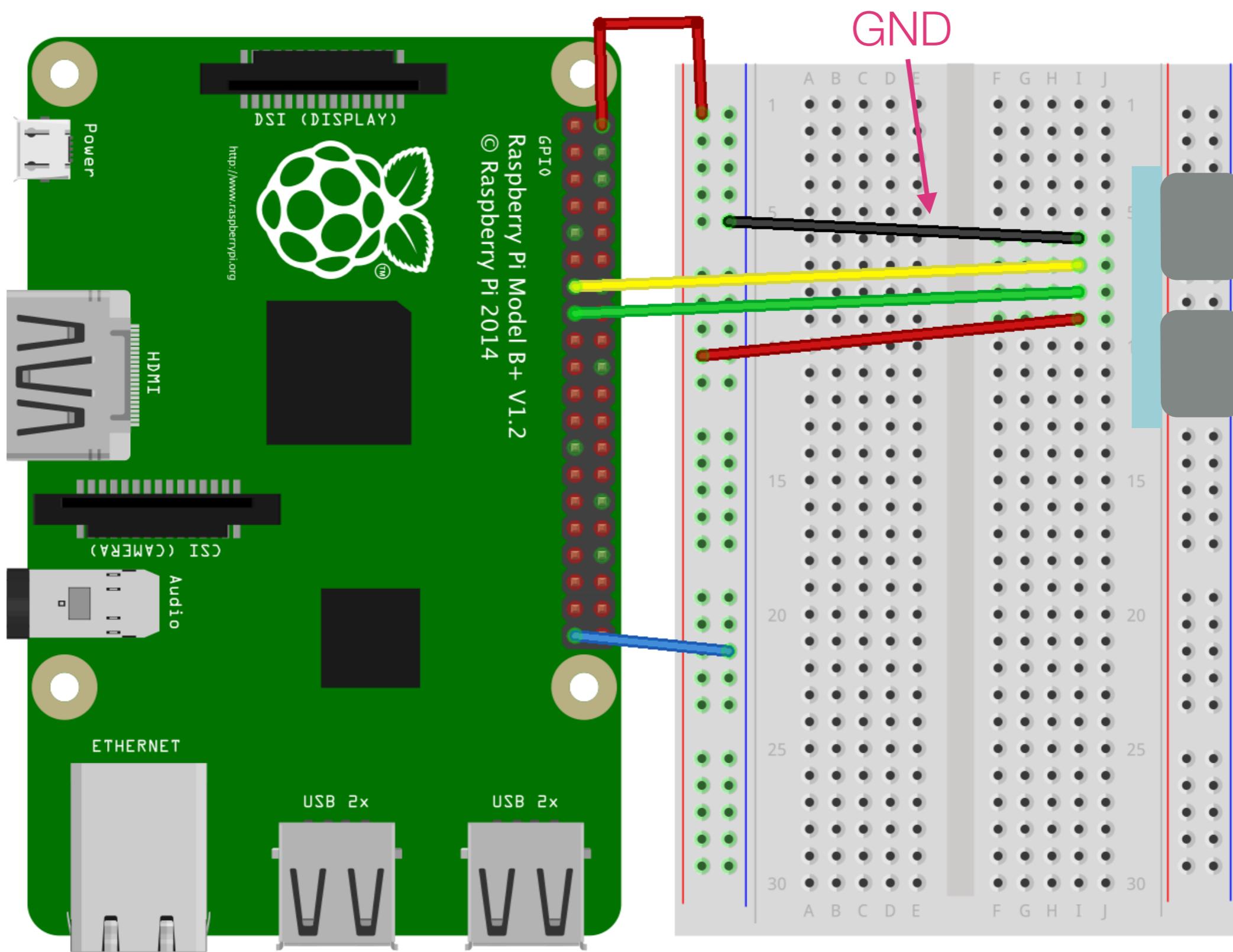
```
cd ../activity_02
```

```
sudo python sound.py
```



Setting Up the Hardware

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

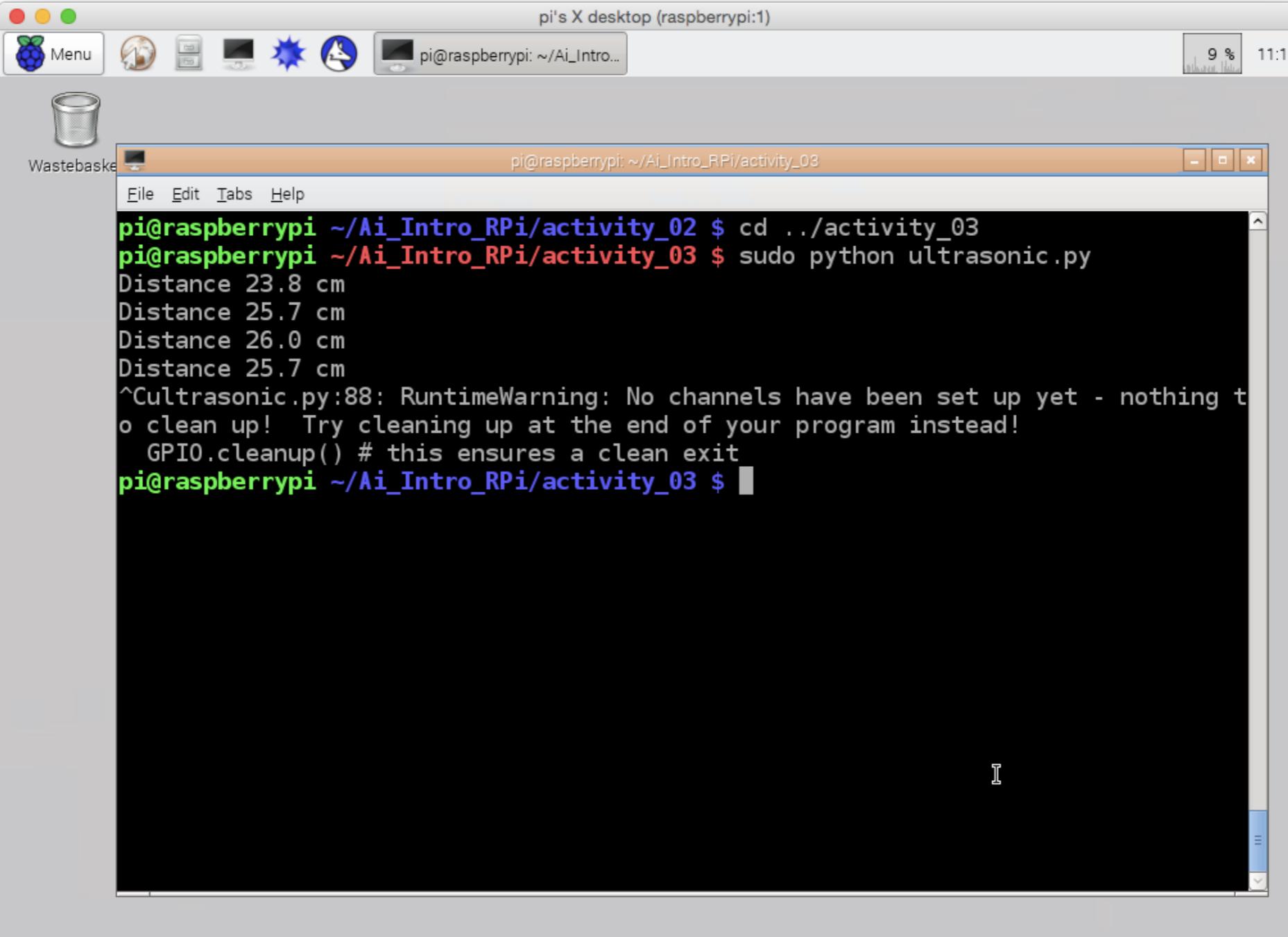


Running the Code

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

```
cd ../activity_03a
```

```
sudo python ultrasonic.py
```

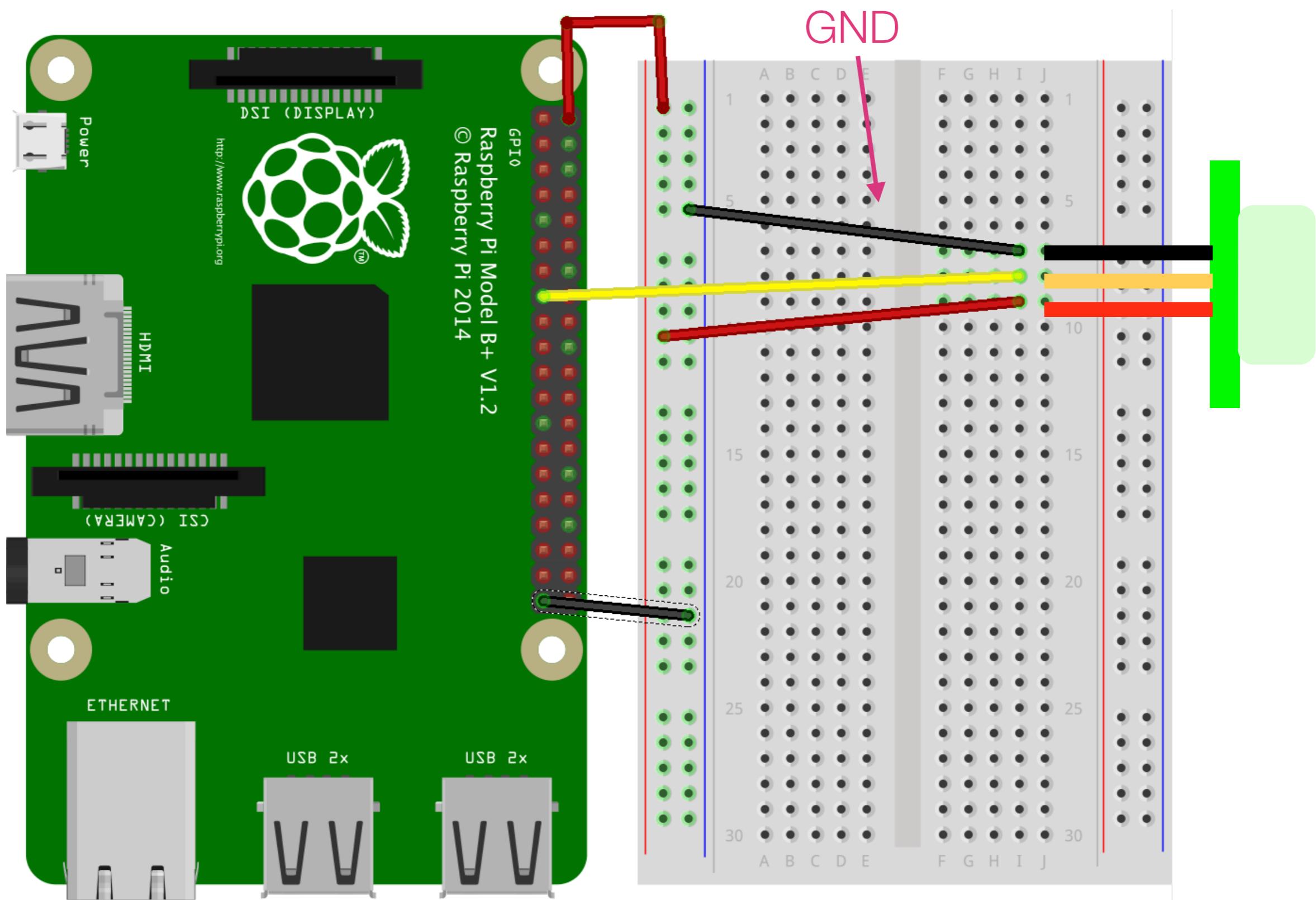


The screenshot shows a Raspberry Pi desktop environment with a terminal window open. The terminal window title is "pi@raspberrypi: ~/Ai_Intro_RPi/activity_03". The terminal output shows the command "sudo python ultrasonic.py" being run, followed by four distance measurements: "Distance 23.8 cm", "Distance 25.7 cm", "Distance 26.0 cm", and "Distance 25.7 cm". A warning message follows: "^Ultrasonic.py:88: RuntimeWarning: No channels have been set up yet - nothing to clean up! Try cleaning up at the end of your program instead!". The command "GPIO.cleanup() # this ensures a clean exit" is also visible. The desktop background is light grey, and there are icons for a trash can and a menu bar.

```
pi@raspberrypi ~/Ai_Intro_RPi/activity_03 $ cd ../activity_03a
pi@raspberrypi ~/Ai_Intro_RPi/activity_03 $ sudo python ultrasonic.py
Distance 23.8 cm
Distance 25.7 cm
Distance 26.0 cm
Distance 25.7 cm
^Ultrasonic.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 ~/Ai_Intro_RPi/activity_03 $
```

Setting Up the Hardware

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

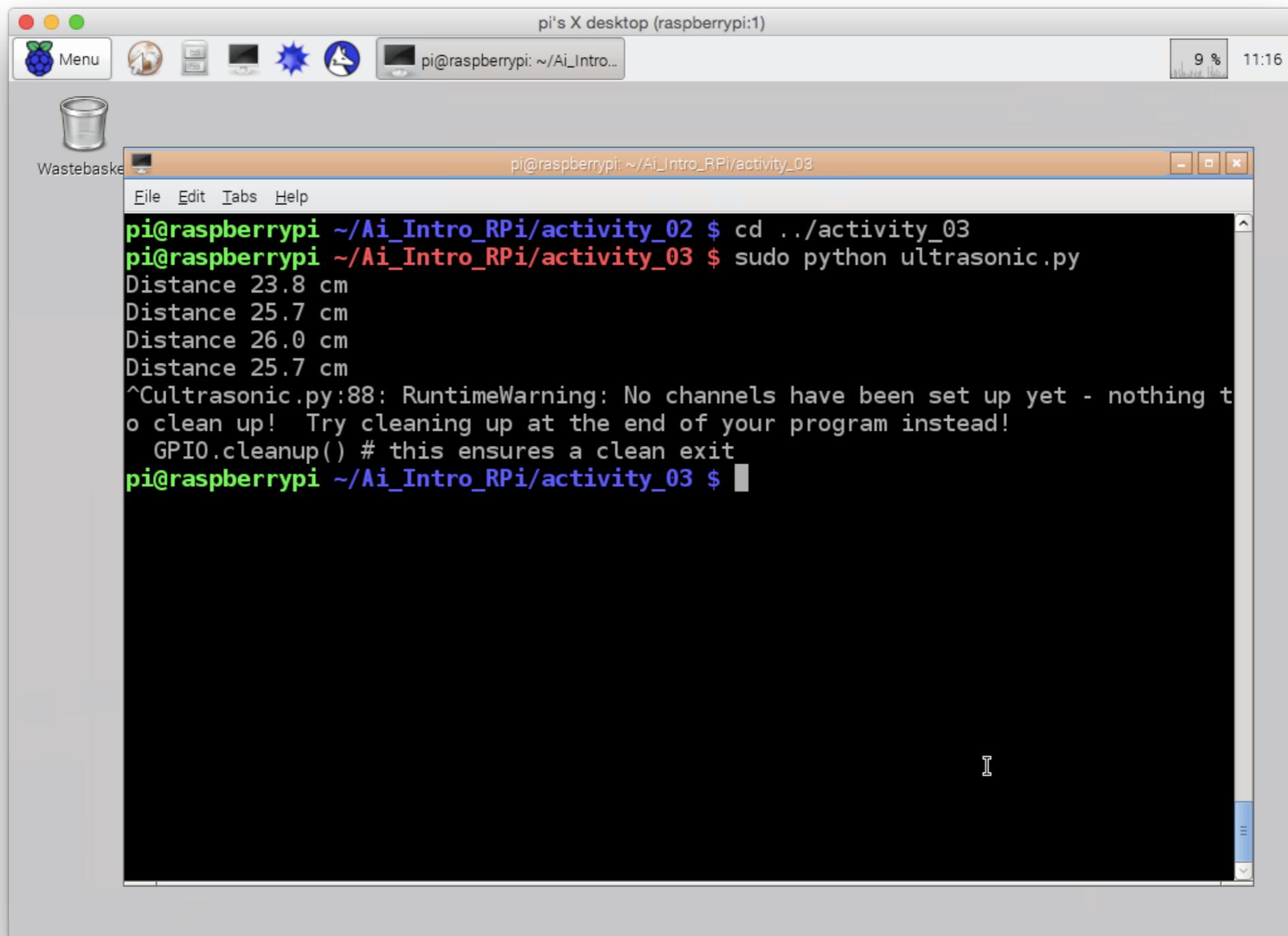


Running the Code

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

```
cd ../activity_03b
```

```
sudo python motion.py
```



The screenshot shows a Raspberry Pi desktop environment with a terminal window open. The terminal window title is "pi@raspberrypi: ~/Ai_Intro_RPi/activity_03". The terminal output shows the command "sudo python ultrasonic.py" being run, followed by four distance measurements: "Distance 23.8 cm", "Distance 25.7 cm", "Distance 26.0 cm", and "Distance 25.7 cm". A warning message follows: "^Ultrasonic.py:88: RuntimeWarning: No channels have been set up yet - nothing to clean up! Try cleaning up at the end of your program instead!". The terminal prompt "pi@raspberrypi ~/" is visible at the bottom.

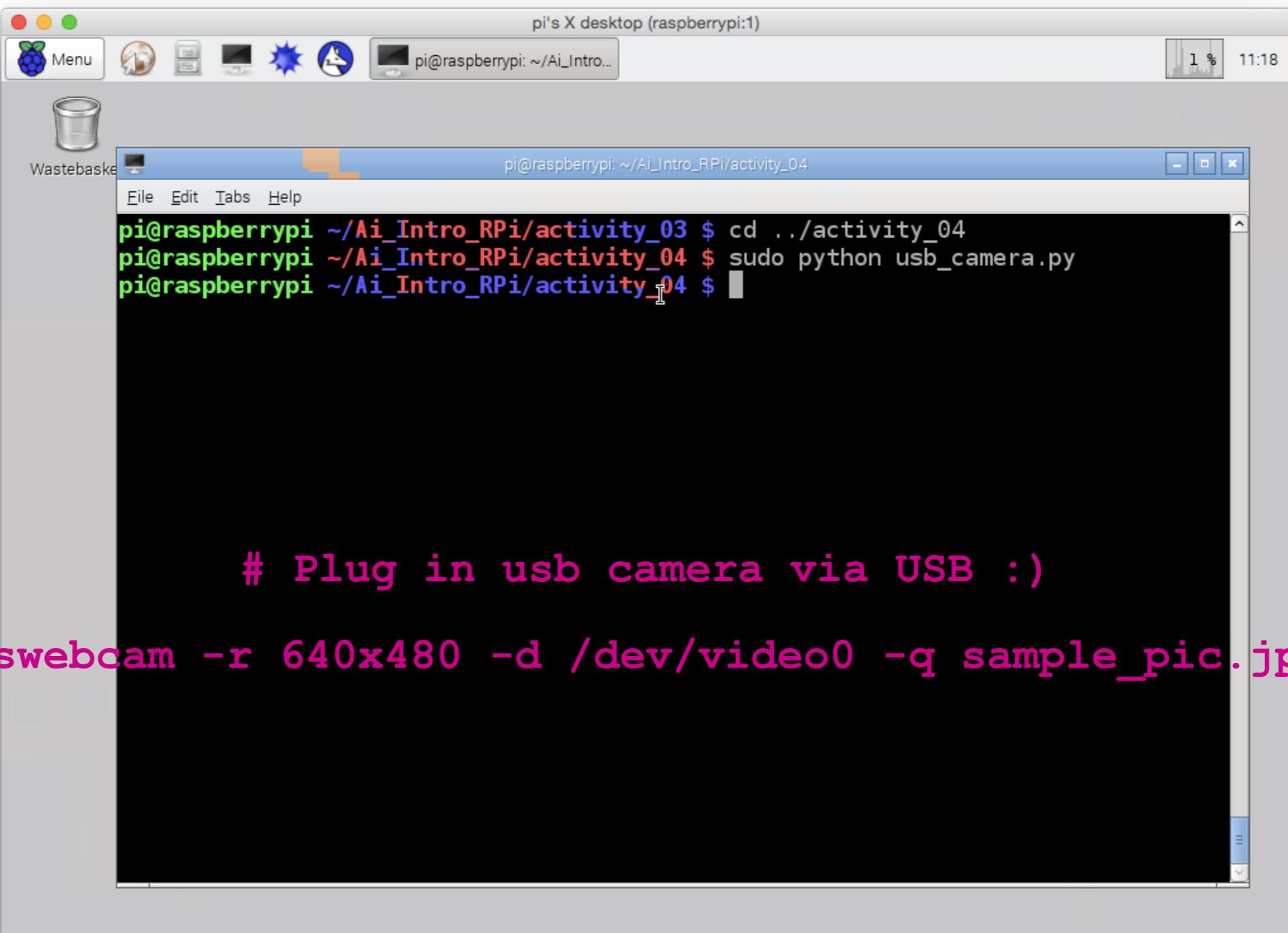
```
pi@raspberrypi ~/Ai_Intro_RPi/activity_03 $ cd ../activity_03b
pi@raspberrypi ~/Ai_Intro_RPi/activity_03 $ sudo python ultrasonic.py
Distance 23.8 cm
Distance 25.7 cm
Distance 26.0 cm
Distance 25.7 cm
^Ultrasonic.py:88: RuntimeWarning: No channels have been set up yet - nothing to clean up! Try cleaning up at the end of your program instead!
    GPIO.cleanup() # this ensures a clean exit
pi@raspberrypi ~/Ai_Intro_RPi/activity_03 $
```

Setting Up the Hardware

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

```
sudo apt-get update
```

```
sudo apt-get install fswebcam
```



The screenshot shows a Raspberry Pi desktop environment with a terminal window open. The terminal window title is "pi@raspberrypi: ~/" and the command being run is "fswebcam -r 640x480 -d /dev/video0 -q sample_pic.jpg". The terminal output includes a note to plug in the USB camera and a timestamp of 11:18.

```
pi's X desktop (raspberrypi:1)
pi@raspberrypi: ~/Ai_Intro_RPi/activity_04
File Edit Tabs Help
pi@raspberrypi ~/Ai_Intro_RPi/activity_04 $ cd ../activity_04
pi@raspberrypi ~/Ai_Intro_RPi/activity_04 $ sudo python usb_camera.py
pi@raspberrypi ~/Ai_Intro_RPi/activity_04 $ 

# Plug in usb camera via USB :)

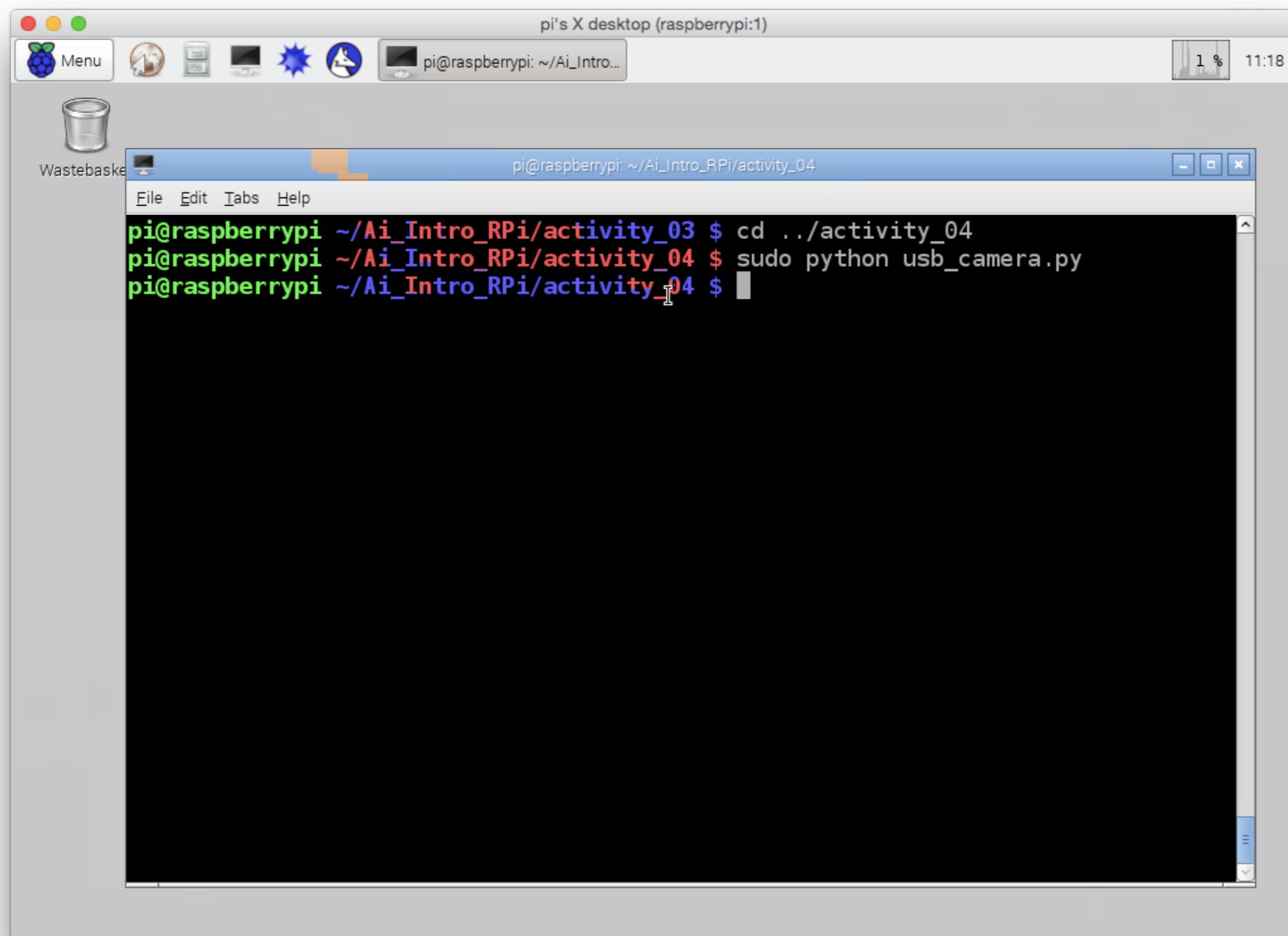
fswebcam -r 640x480 -d /dev/video0 -q sample_pic.jpg
11:18
```

Running the Code

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

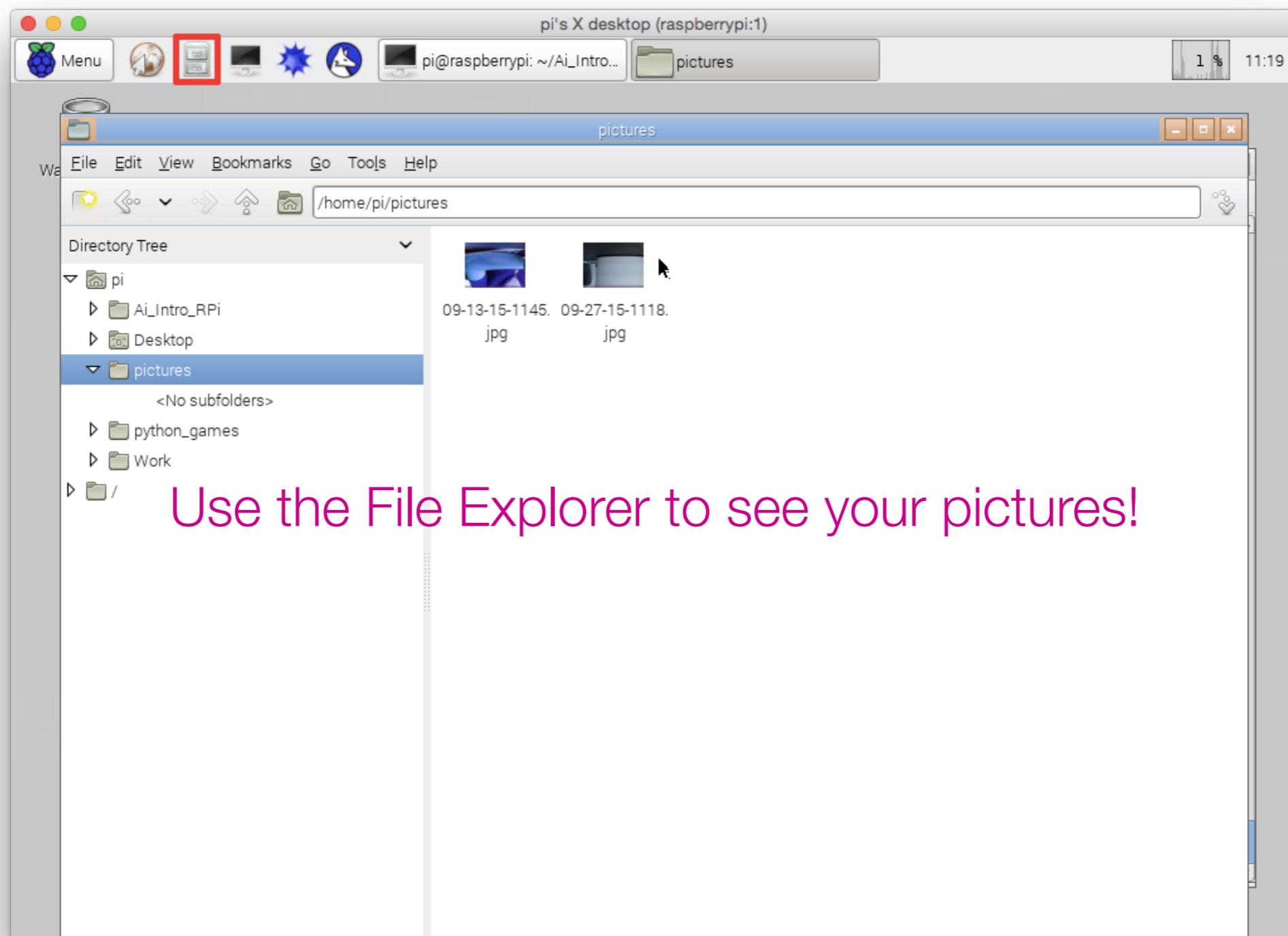
```
cd ../activity_04
```

```
sudo python usb_camera.py
```



Running the Code

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



Configuring the Backend

Raspbian and system administration basics

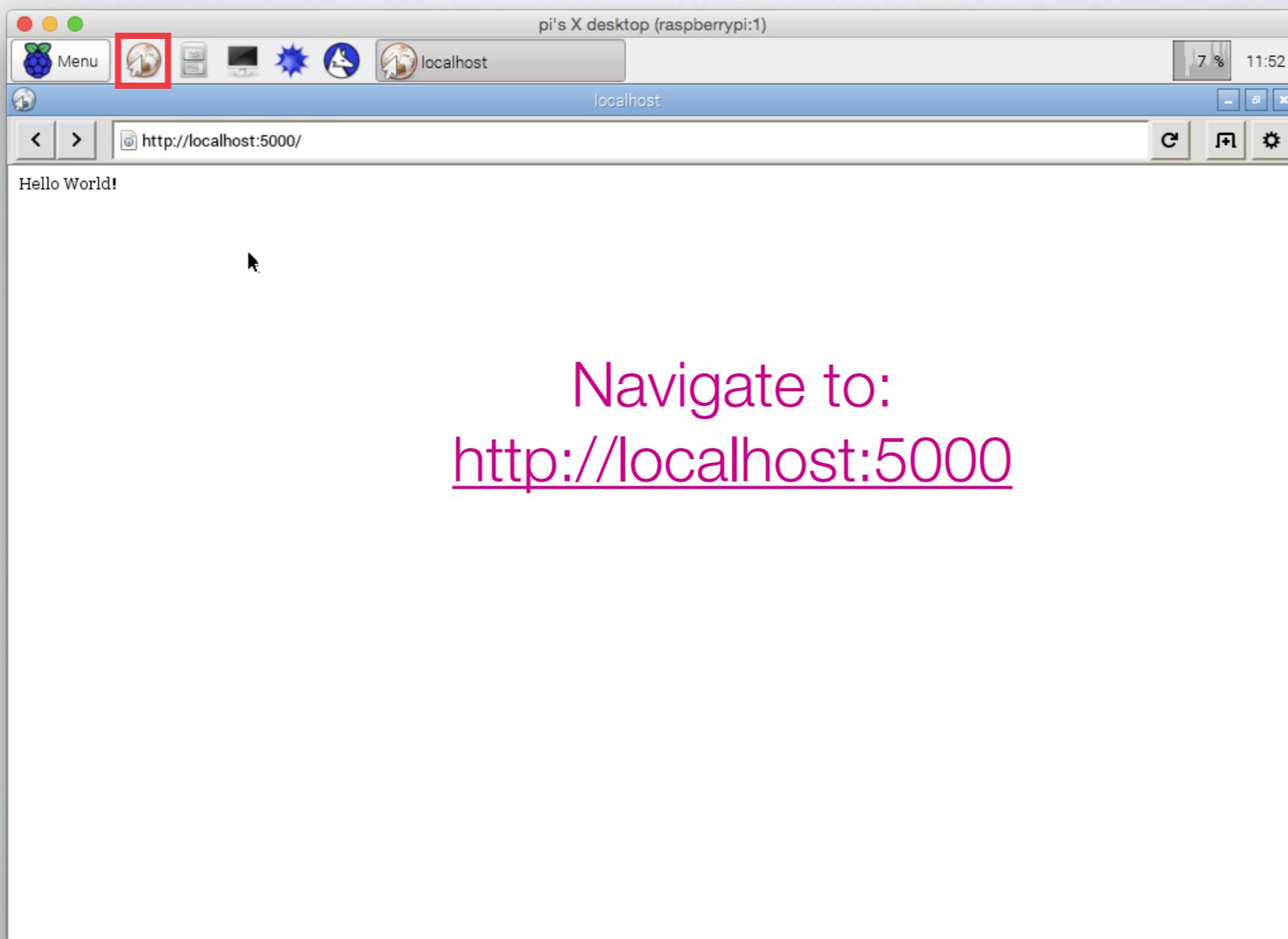
Running a webserver using Python and Flask

Building an API with Flask for LED control

Configuring the Backend

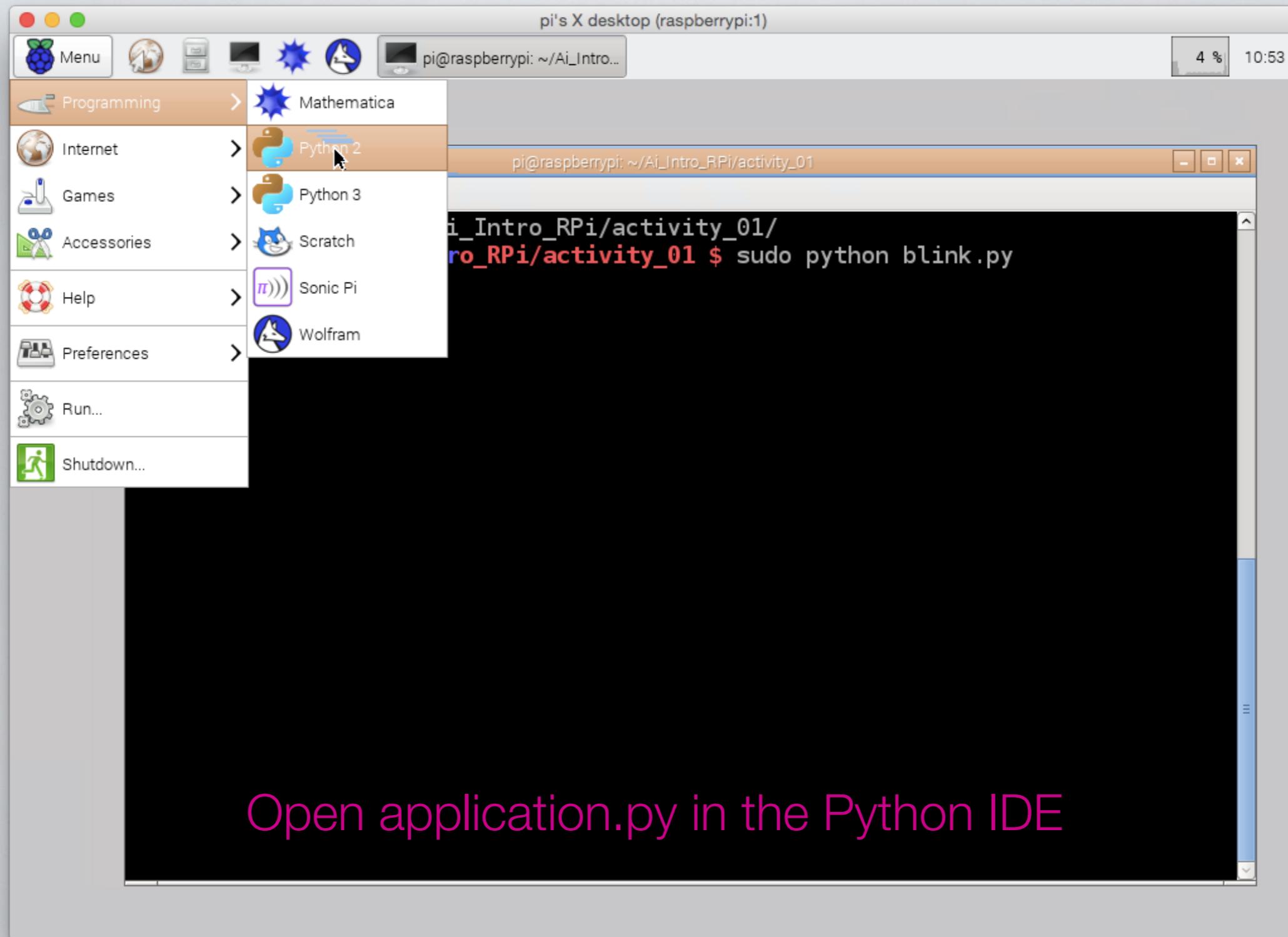
Activity 5: Getting started with web frameworks [application.py]

```
cd ../activity_05/simple  
sudo apt-get install python-pip  
sudo pip install flask  
sudo python application.py
```



Configuring the Backend

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



Configuring the Backend

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

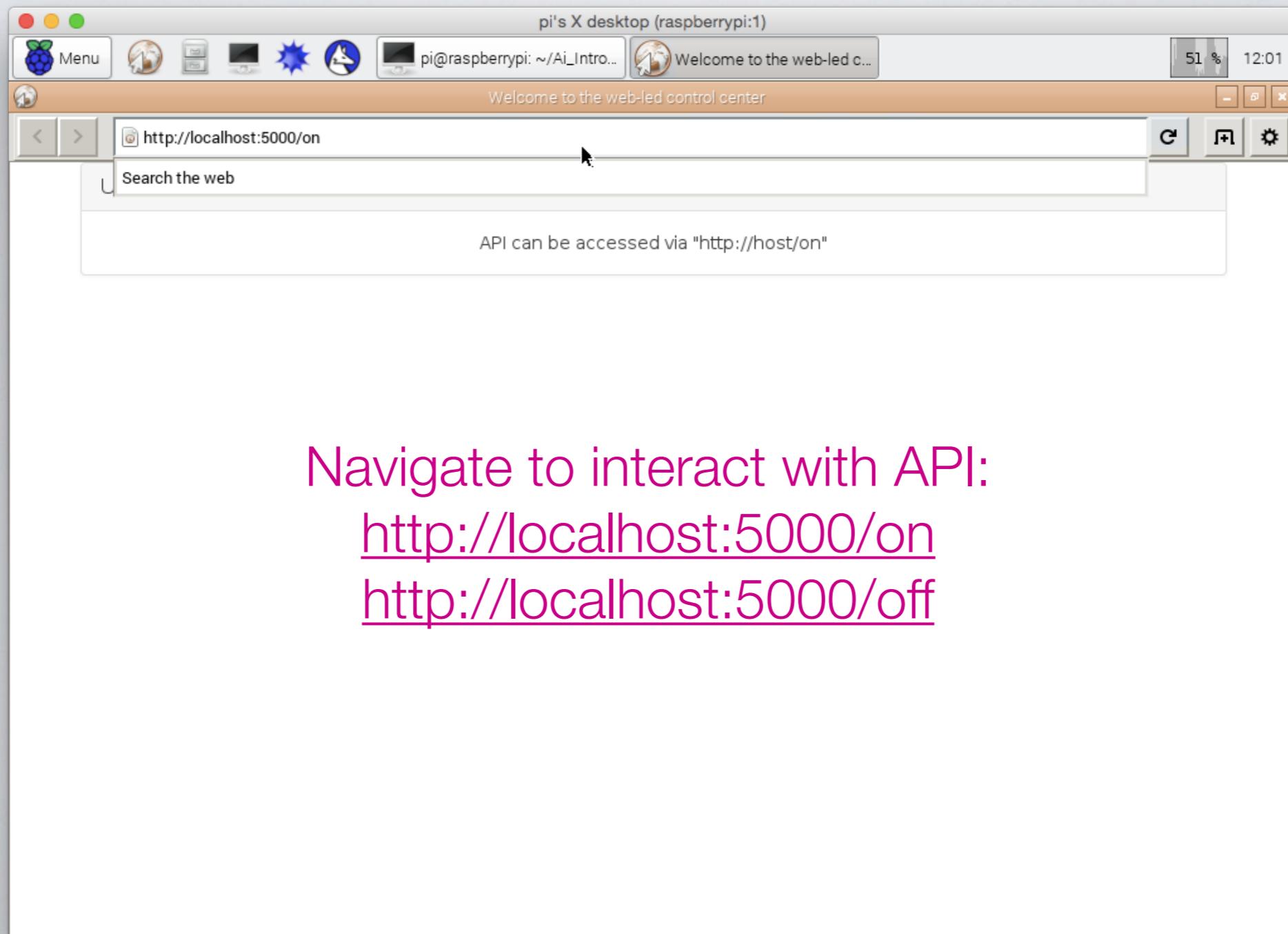
Q&A

Configuring the Backend

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

```
cd ../../activity_06
```

```
sudo python application.py
```



Navigate to interact with API:

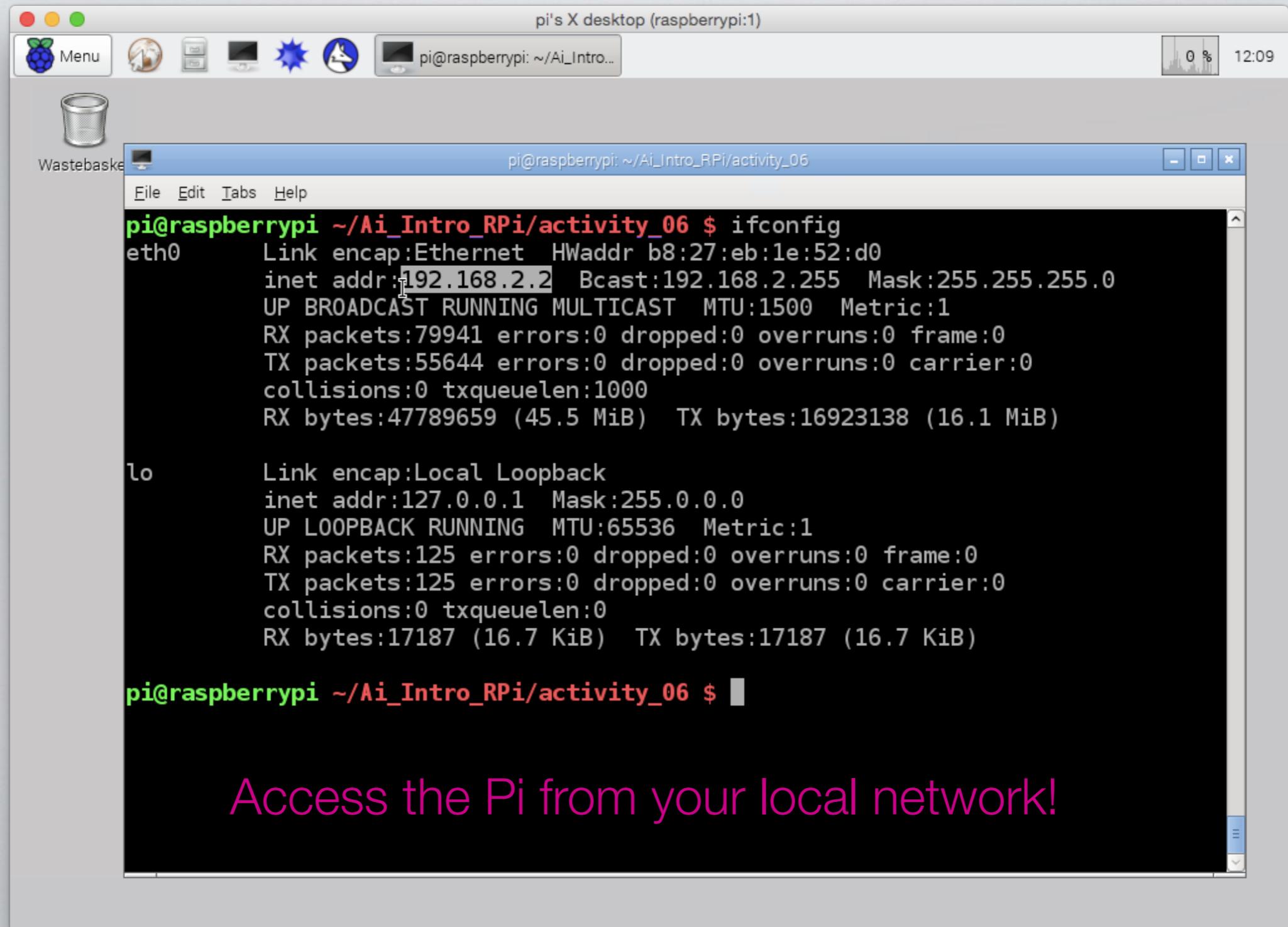
<http://localhost:5000/on>

<http://localhost:5000/off>

Configuring the Backend

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

`ifconfig`



The screenshot shows a Xfce desktop environment on a Raspberry Pi. A terminal window is open, displaying the output of the `ifconfig` command. The terminal window title is `pi@raspberrypi: ~/Ai_Intro_RPi/activity_06`. The desktop background shows icons for a trash can and a waste basket. The top bar shows the window title `pi's X desktop (raspberrypi:1)`, the user `pi@raspberrypi: ~/Ai_Intro...`, battery level `0 %`, and time `12:09`.

```
pi@raspberrypi ~/Ai_Intro_RPi/activity_06 $ ifconfig
eth0      Link encap:Ethernet HWaddr b8:27:eb:1e:52:d0
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
                  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                  RX packets:79941 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:55644 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:47789659 (45.5 MiB)  TX bytes:16923138 (16.1 MiB)

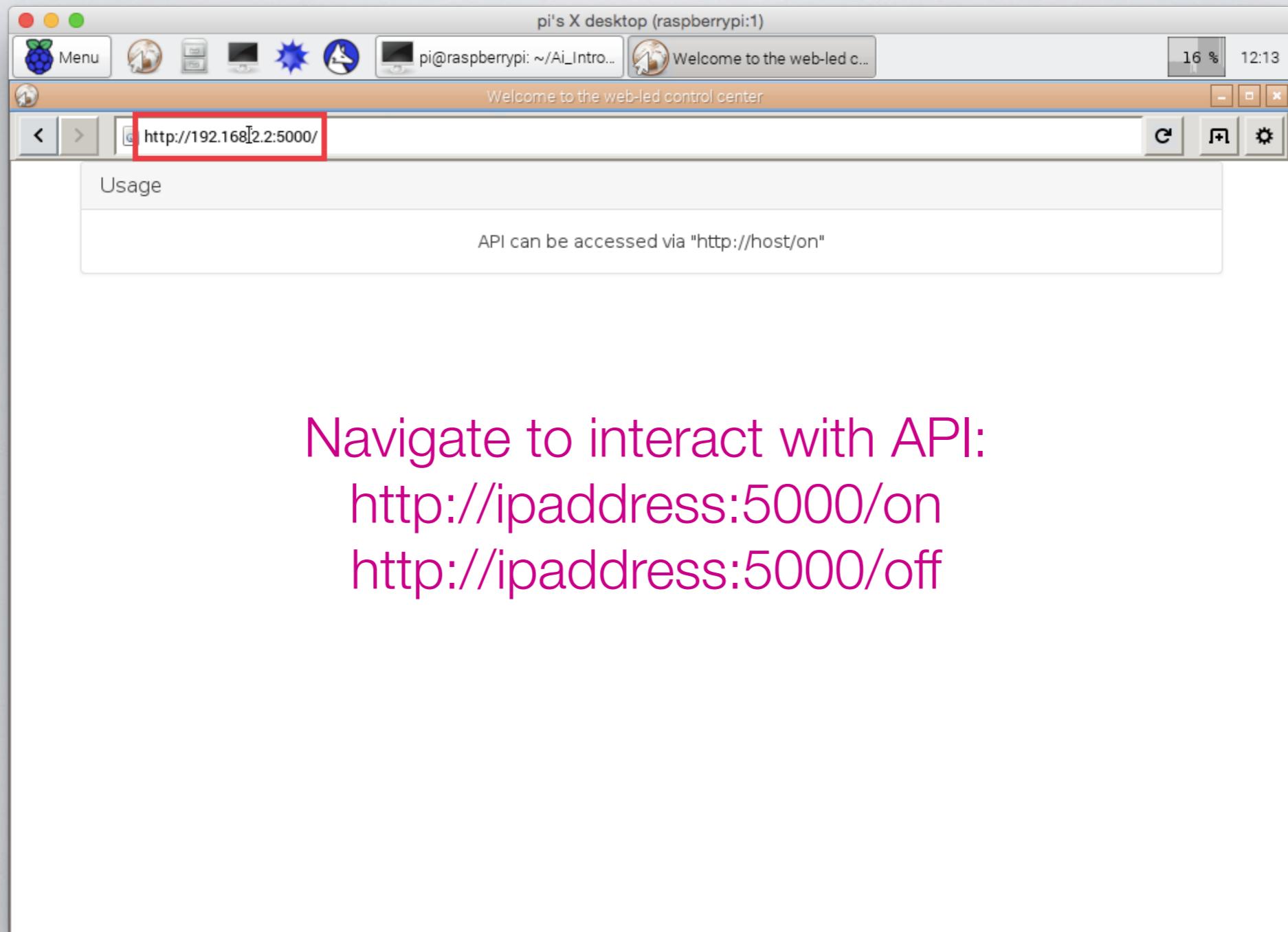
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
                  UP LOOPBACK RUNNING  MTU:65536  Metric:1
                  RX packets:125 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:125 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:17187 (16.7 KiB)  TX bytes:17187 (16.7 KiB)

pi@raspberrypi ~/Ai_Intro_RPi/activity_06 $
```

Access the Pi from your local network!

Configuring the Backend

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



Navigate to interact with API:
<http://ipaddress:5000/on>
<http://ipaddress:5000/off>

Configuring the Backend

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

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

Q&A

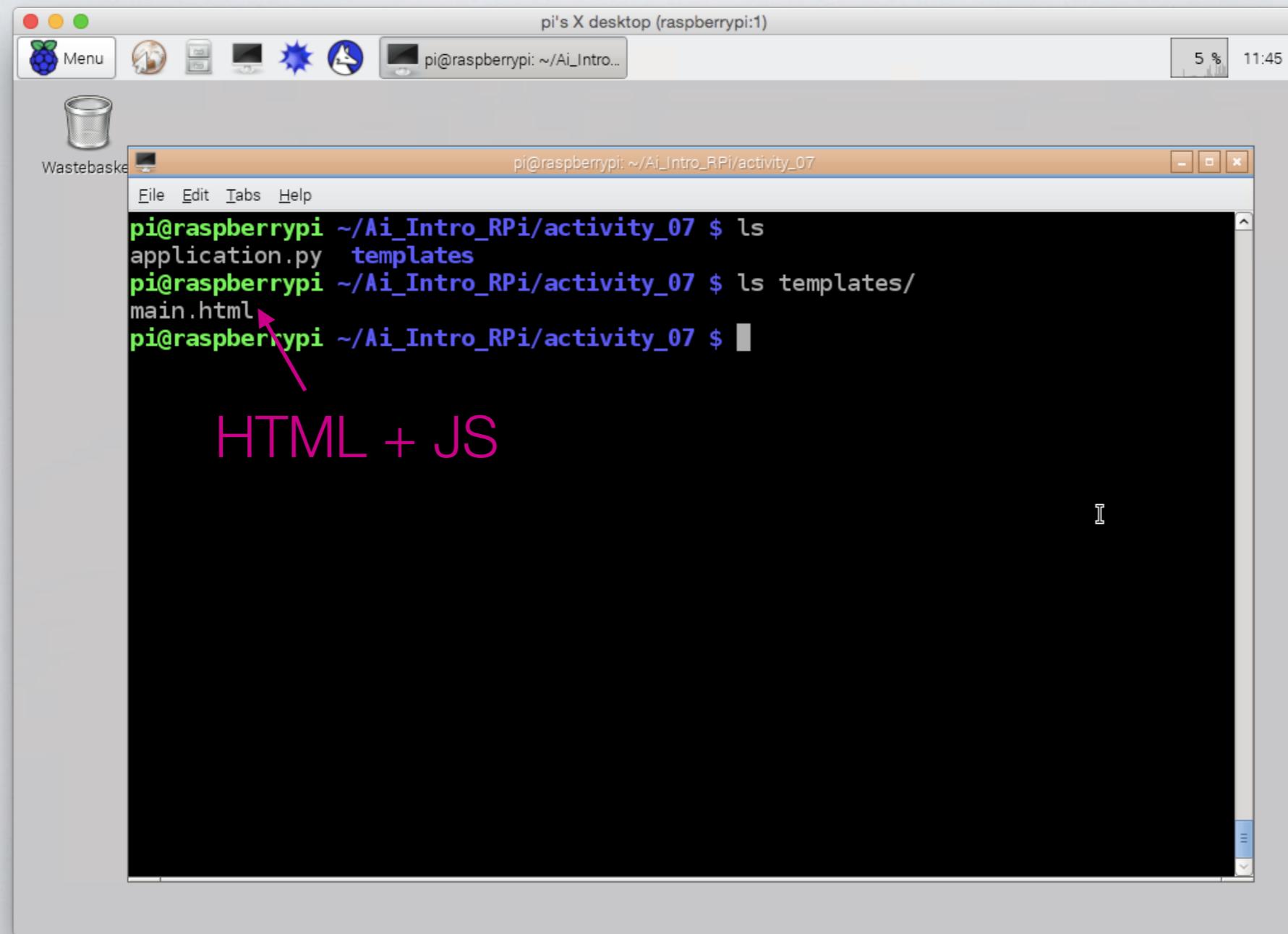
Building a User Interface (Frontend)

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

Building a User Interface

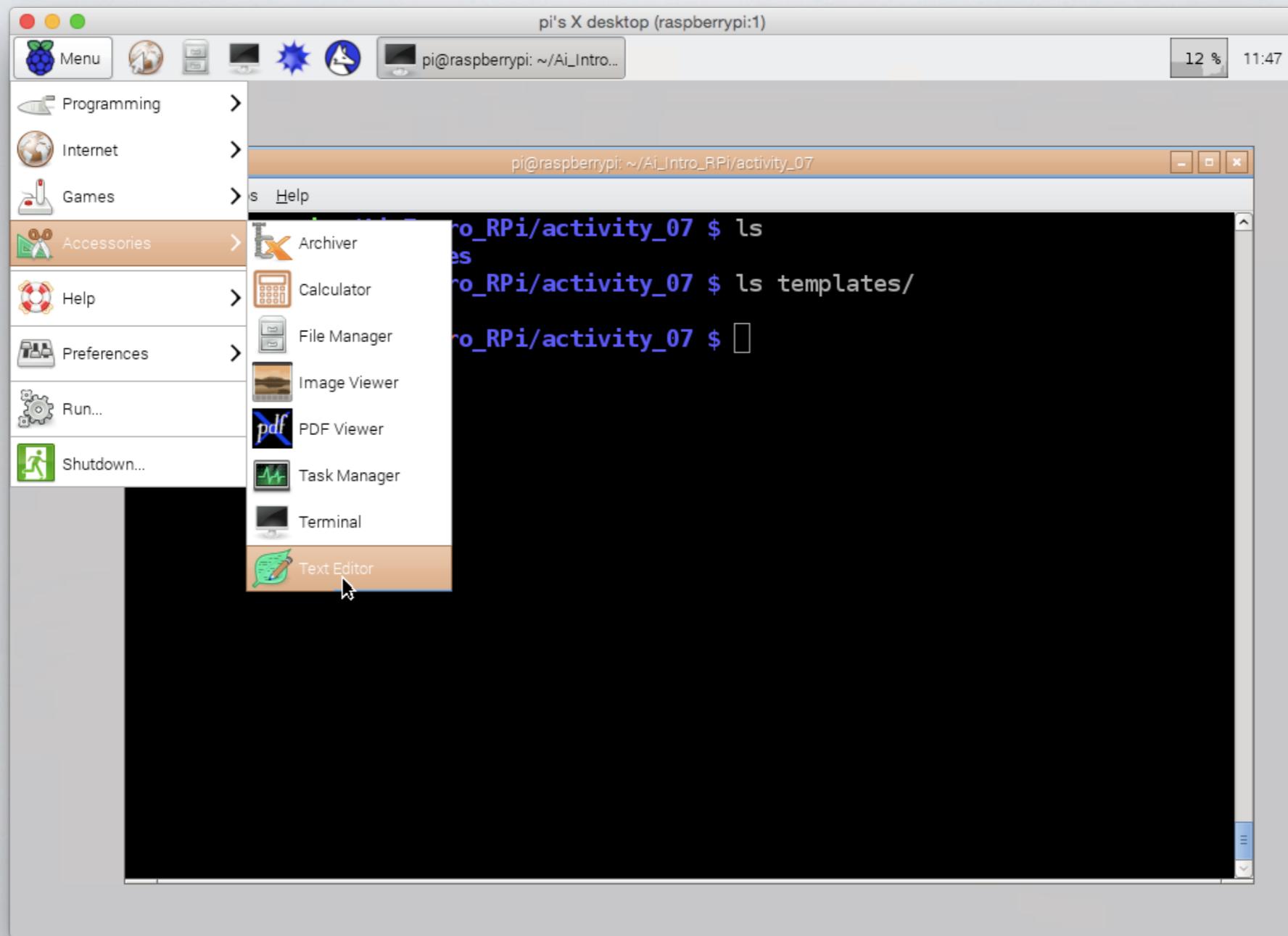
Activity 7: Creating a simple UI for LED control [application.py]

```
cd ../activity_07
```



Building a User Interface

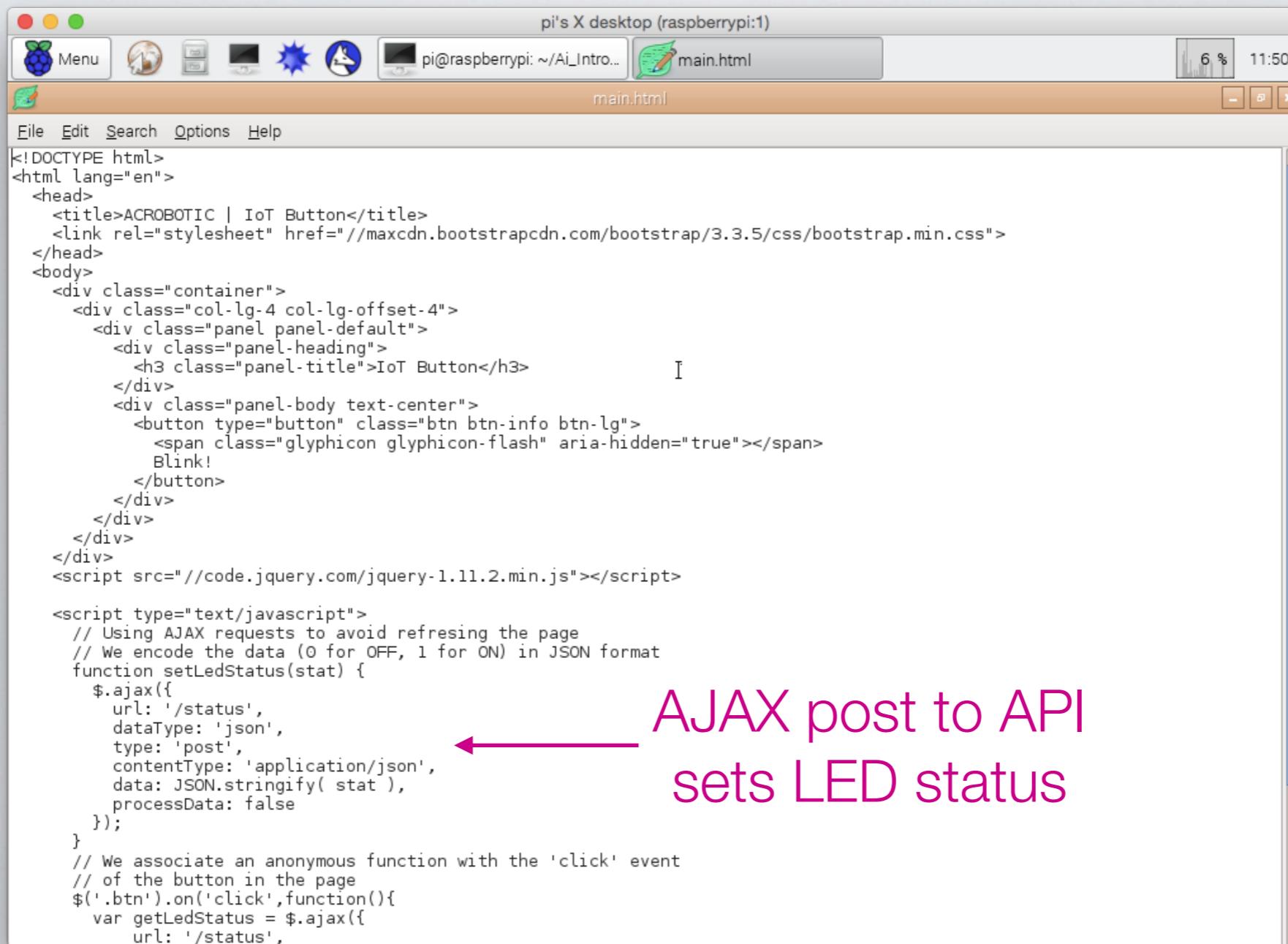
Activity 7: Creating a simple UI for LED control [application.py]



Building a User Interface

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

Q&A



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>ACROBOTIC | IoT Button</title>
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
</head>
<body>
    <div class="container">
        <div class="col-lg-4 col-lg-offset-4">
            <div class="panel panel-default">
                <div class="panel-heading">
                    <h3 class="panel-title">IoT Button</h3>
                </div>
                <div class="panel-body text-center">
                    <button type="button" class="btn btn-info btn-lg">
                        <span class="glyphicon glyphicon-flash" aria-hidden="true"></span>
                        Blink!
                    </button>
                </div>
            </div>
        </div>
    </div>
    <script src="//code.jquery.com/jquery-1.11.2.min.js"></script>

    <script type="text/javascript">
        // Using AJAX requests to avoid refresing the page
        // We encode the data (0 for OFF, 1 for ON) in JSON format
        function setLedStatus(stat) {
            $.ajax({
                url: '/status',
                dataType: 'json',
                type: 'post',
                contentType: 'application/json',
                data: JSON.stringify( stat ),
                processData: false
            });
        }
        // We associate an anonymous function with the 'click' event
        // of the button in the page
        $('.btn').on('click',function(){
            var getLedStatus = $.ajax({
                url: '/status',
                type: 'get'
            });
        });
    </script>

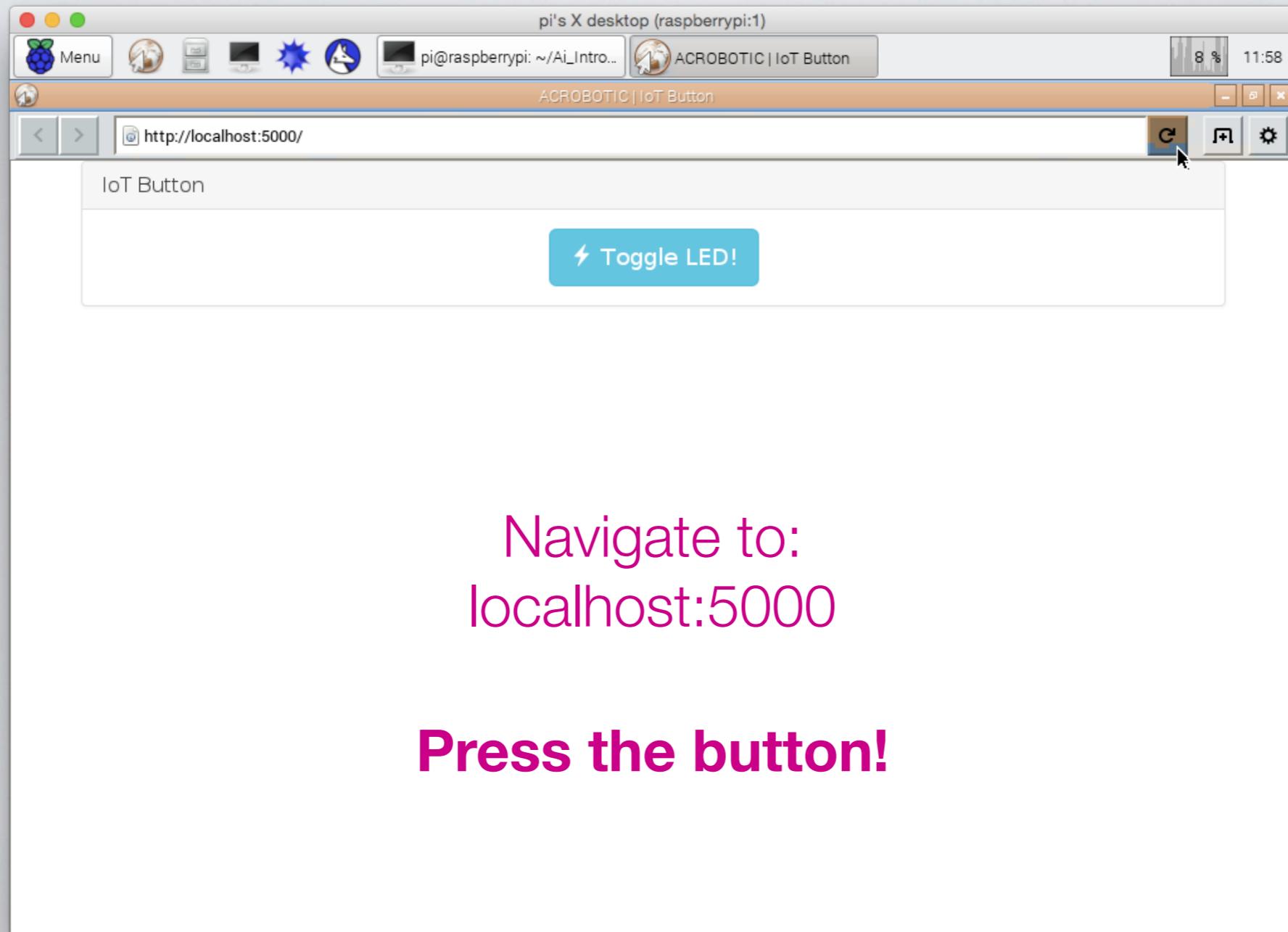
```

AJAX post to API
sets LED status

Building a User Interface

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

```
sudo python application.py
```

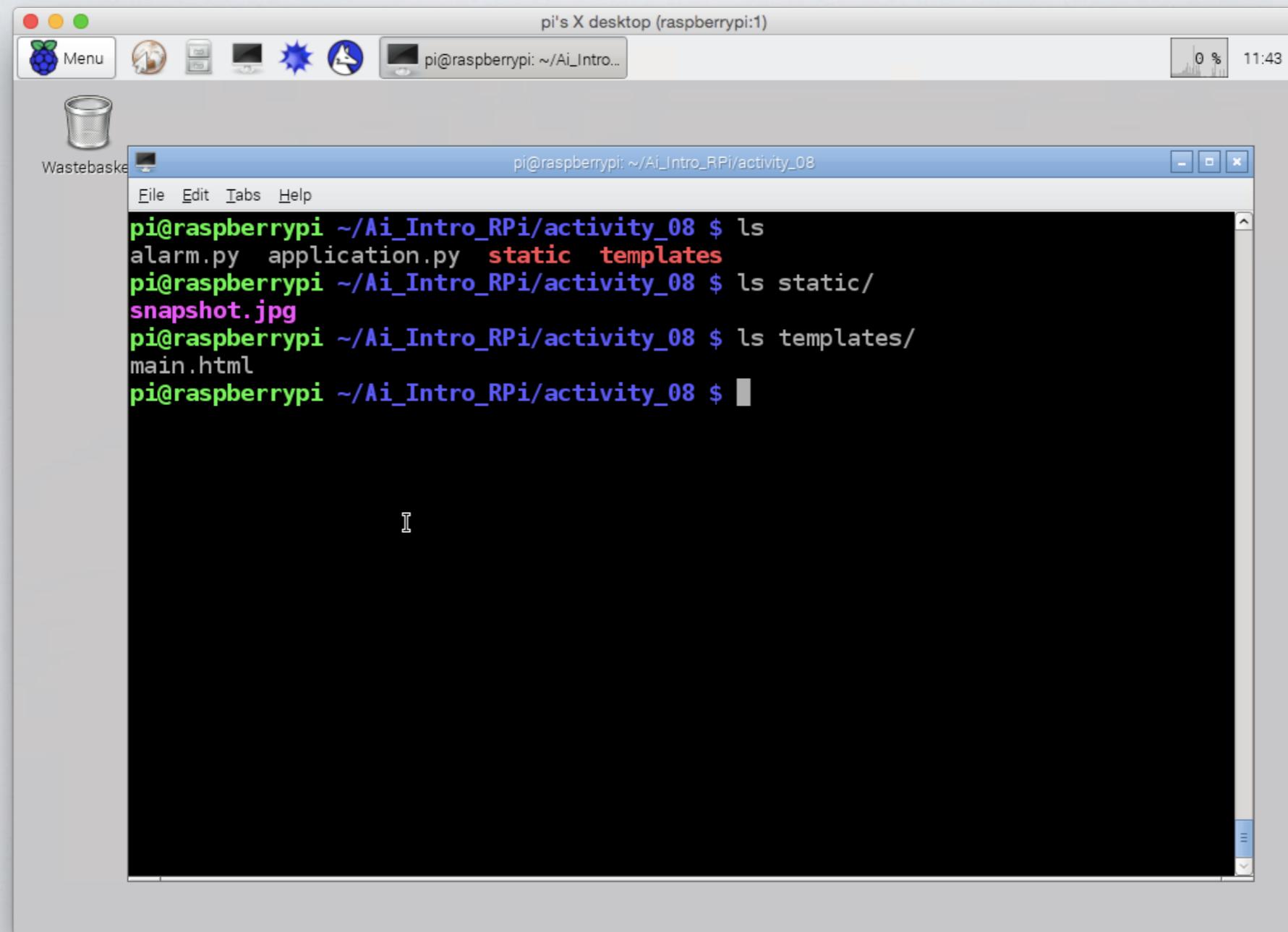


Building a User Interface

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

```
cd ../activity_08
```

```
sudo python application.py
```

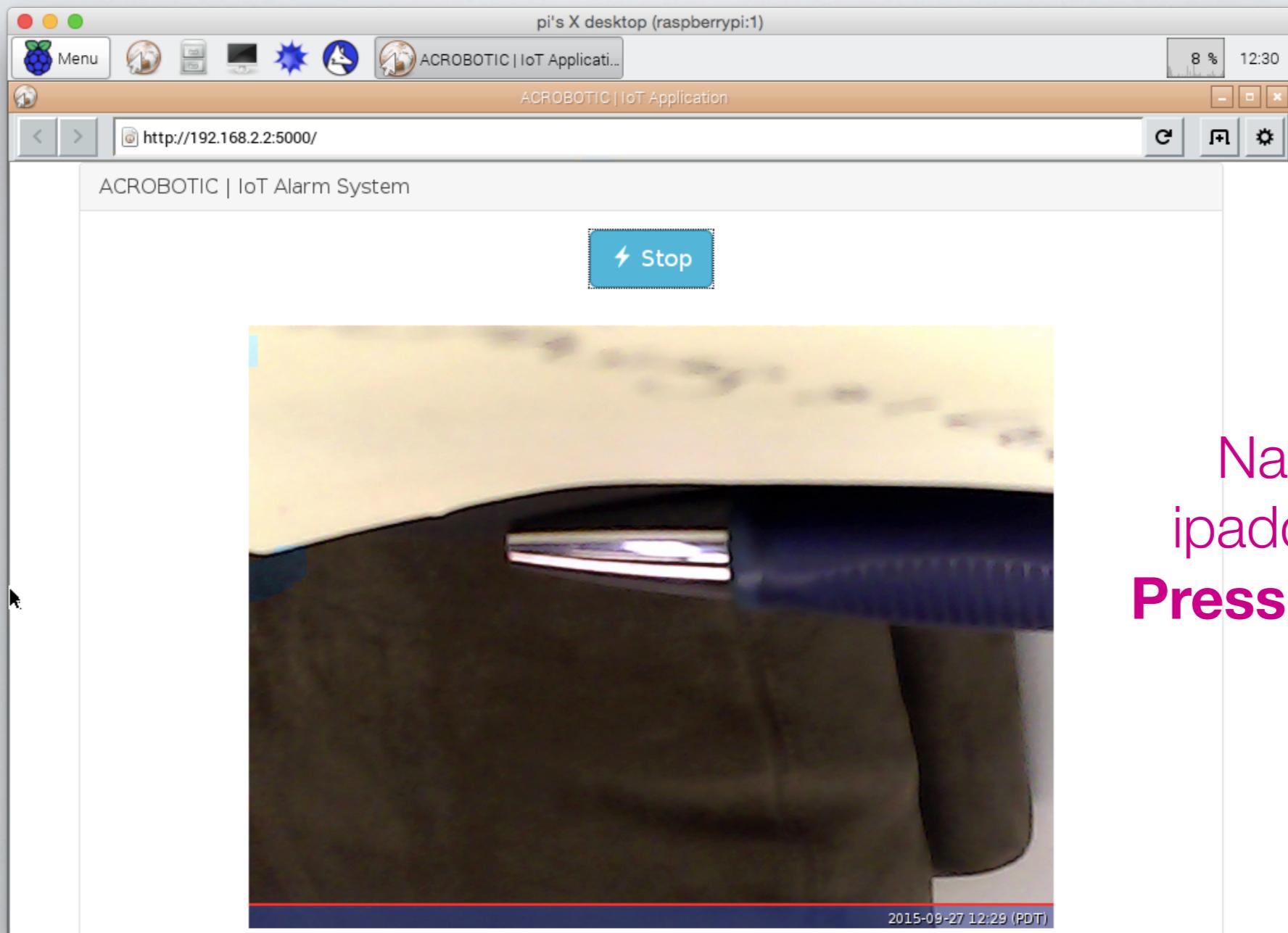


Building a User Interface

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

```
cd ../activity_08
```

```
sudo python application.py
```



Navigate to:
ipaddress:5000
Press the button!