Intro to Hacking with the Raspberry Pi

Sarah Withee (@geekygirlsarah)

Titanium Sponsors

DATA BANK
powered by actifio

epiq SYSTEMS

PAIGE TECHNOLOGIES
INTELLIGENT PAIRING. PERPETUAL SUCCESS.

Platinum Sponsors

perceptivesoftware from Lexmark

ADAPTIVE SOLUTIONS GROUP

KU EDWARDS CAMPUS The University of Kansas

pinsight MEDIA+

VinSolutions
Make every connection count.

MULTI SERVICE
INNOVATION WHERE IT MATTERS

Gold Sponsors

DST SYSTEMS

LRS Consulting Services

NETCHEMIA
Transforming the way education works

ADVANTAGE TECH

KEYHOLE SOFTWARE

Balance Innovations

stackify

GARMIN

NEW DIRECTIONS

Bradford & Galt

OAKWOOD

Cerner

jack henry & ASSOCIATES INC.

UNITEDLEX

dsi

FitBark

Commerce Bank

BATS
Making Markets Better®

LIFERAY

imodules

eccoselect
PEOPLE | PROJECTS | PERFORMANCE

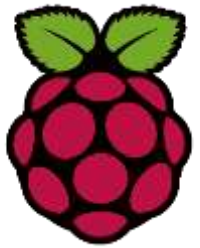
TRIPLE-I

CENTRIQ TRAINING

Breakdown

1. Raspberry Pi Information
2. Intro to Hardware
3. Project 1 (LEDs) – Cylon/Knight Rider Lights
4. Project 2 (LCD Screen) – LCD Twitter Feed
5. Project 3 (Sensors) – Wall-Avoiding Robot
6. Other Ideas
7. More Information and Conclusion

What is the Raspberry Pi?



“The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It also plays high-definition video. We want to see it being used by kids all over the world to learn programming.”

-- raspberrypi.org/faqs

What is “hacking”?

“We are taking back the term ‘Hacking’ which has been soured in the public mind. Hacking is an art form that uses something in a way in which it was not originally intended. This highly creative activity can be highly technical, simply clever, or both. Hackers bask in the glory of building it instead of buying it, repairing it rather than trashing it, and raiding their junk bins for new projects every time they can steal a few moments away.”

-- hackaday.com/about/

Questions

- How many of you own a Raspberry Pi?
- How many of you haven't done anything with it?
- How many of you have installed some server (web/file/media/etc.) and don't really touch it?
- How many of you have built projects with it?

Disclaimer

This IS a workshop…

› for integrating the Pi into fun hacking projects…

› for learning to program with sensors, LEDs, and more

› for learning to build sample projects to inspire you to build your own projects

Disclaimer

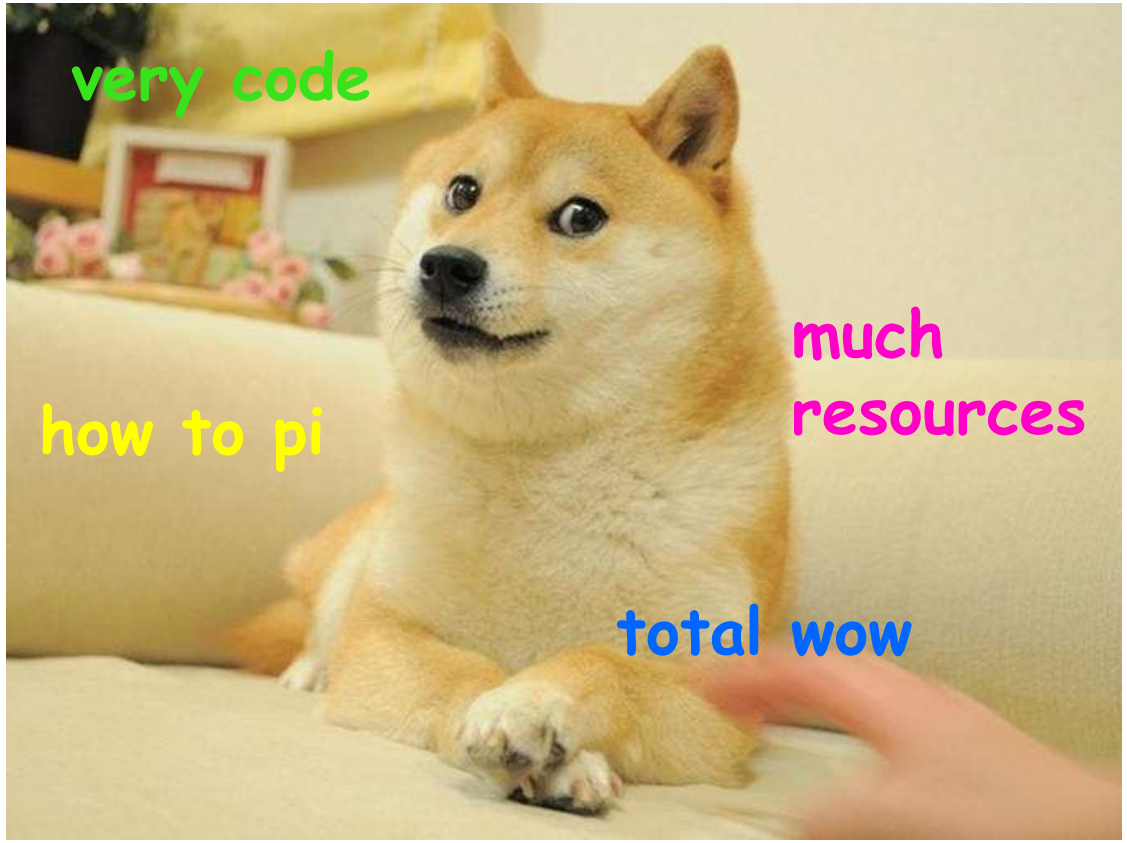
This is NOT a workshop…

› on how to use Linux

› on how to install a Raspberry Pi OS

› for advanced electronics

› on general programming



www.sarahwithee.com/raspberrypi

π

Raspberry Pi Information

Raspberry Pi Models





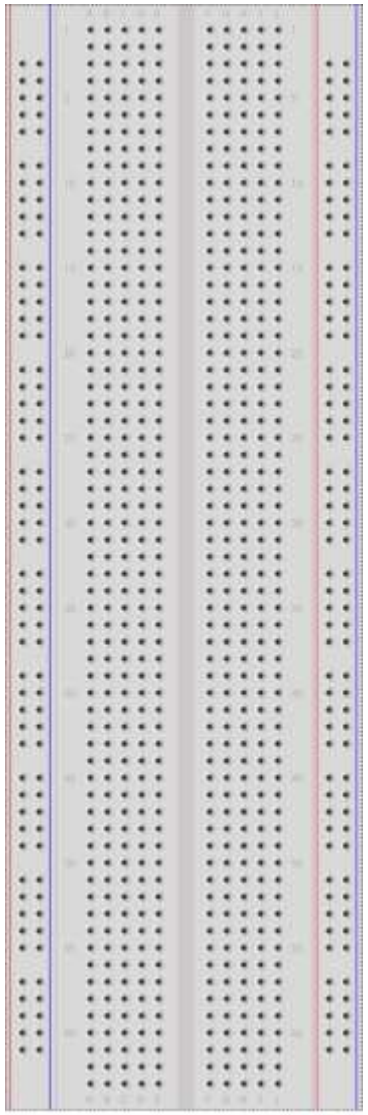
	Pi Model A	Pi Model B	Pi Model A+	Pi Model B+	Pi 2 Model B
Processor	700 MHz	700 MHz	700 MHz	700 MHz	4x 900MHz
Memory	256 MB	512 MB	256 MB	512 MB	1 GB
Expansion	Full SD	Micro SD	Full SD	Micro SD	Micro SD
USB Ports	1	2	1	4	4
Ethernet	No	10/100	No	10/100	10/100
A/V	HDMI, composite	HDMI, composite	HDMI, 3.5mm jack	HDMI, 3.5mm jack	HDMI, 3.5mm jack
GPIO Pins	26	26	40	40	40
Power	300mA*, 1.5W	700mA*, 3.5W	200mA*, 1W	600mA*, 3W	800mA*, 4W
Cost	$25	$35	$20	$30	$35

* Barebones Pi with no peripherals attached. 1.2A minimum power supply recommended.

Intro to Hardware

What Is a Breadboard?

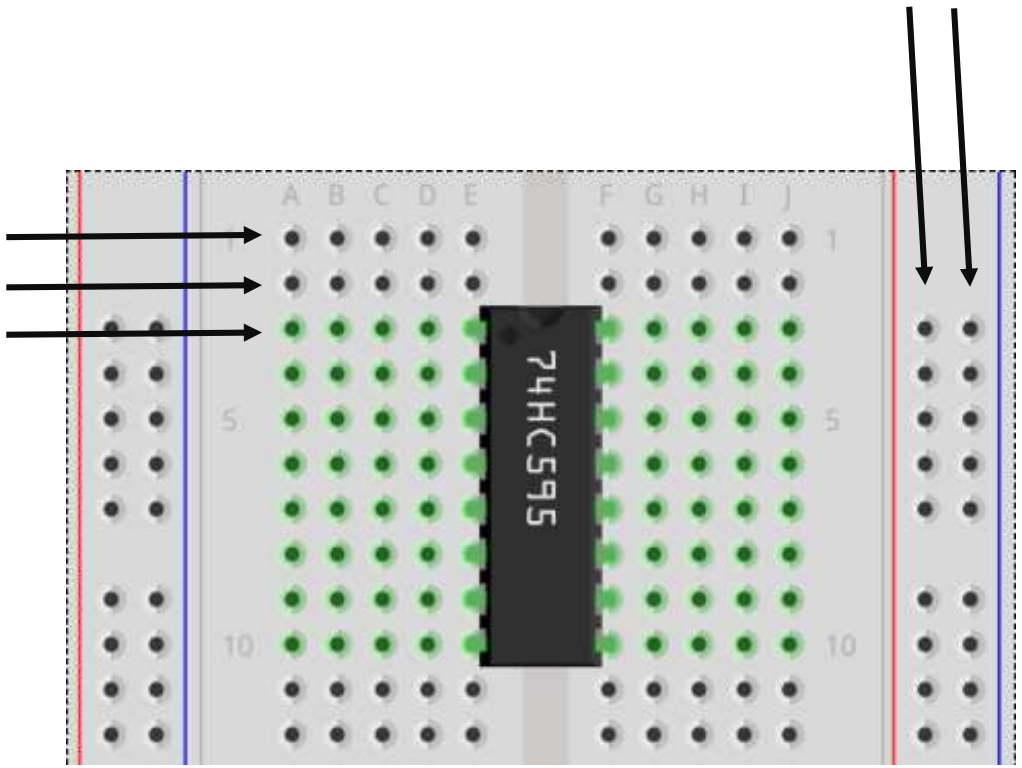
- A type of board used for prototyping
- Each row of 5 is connected
- Each column of 50 is connected
- Columns on side are power rails:
  - Red is power
  - Blue is ground



What Is a Breadboard?

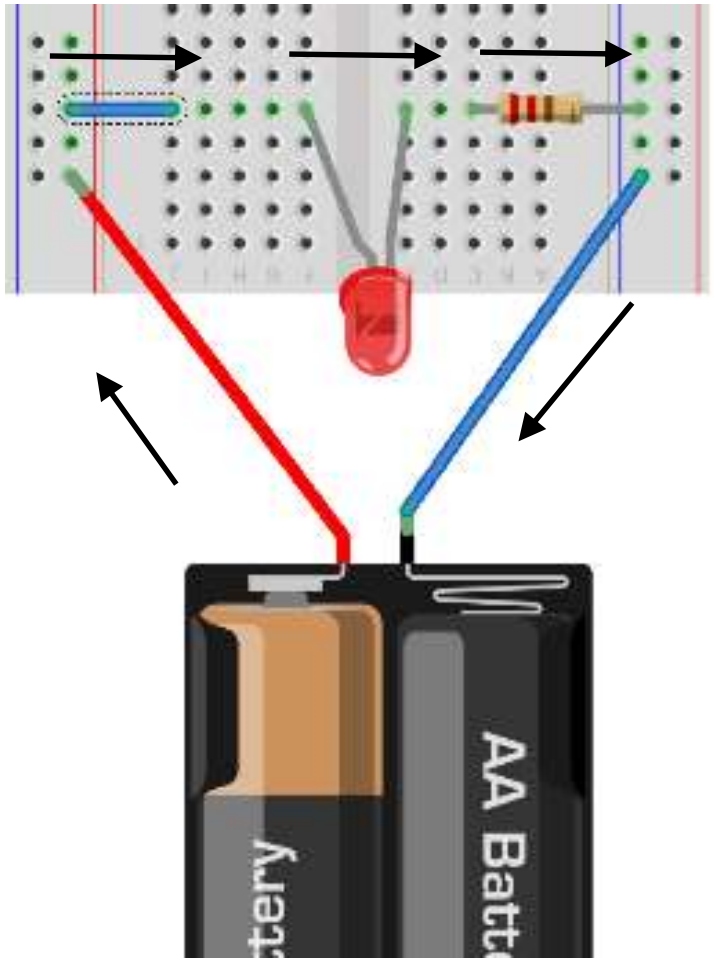
Each column of holes are connected together

Each row of 5 holes are connected to each other



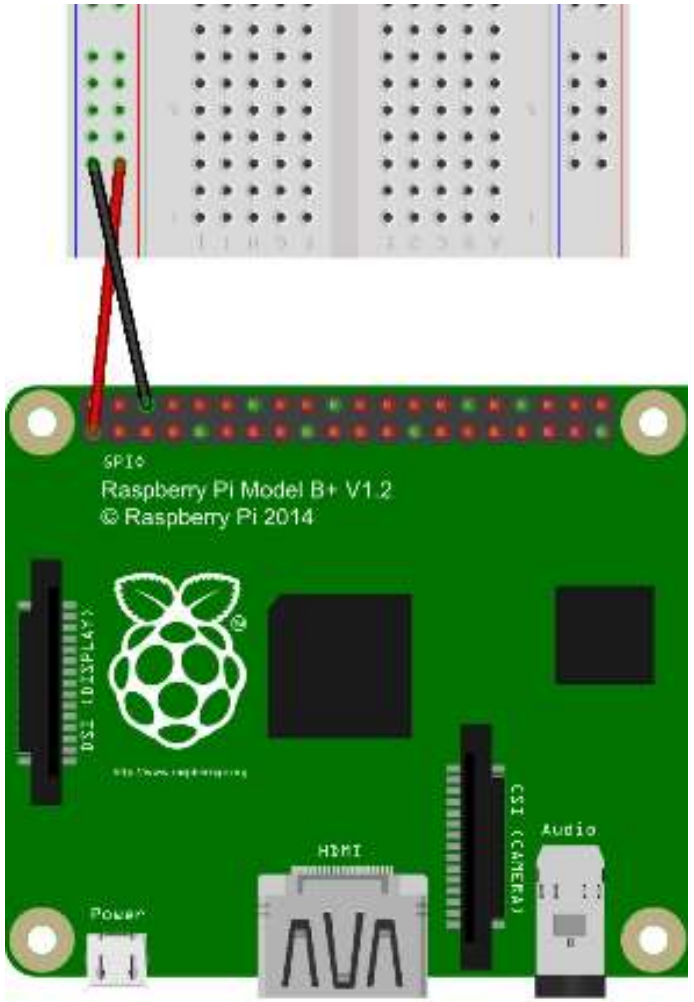
Making a Circuit

- Just like middle school circuitry
- Electricity must flow in a full circle from source, through elements, back to the source
- Source of power is called positive terminal
- Must return back to the source at negative terminal



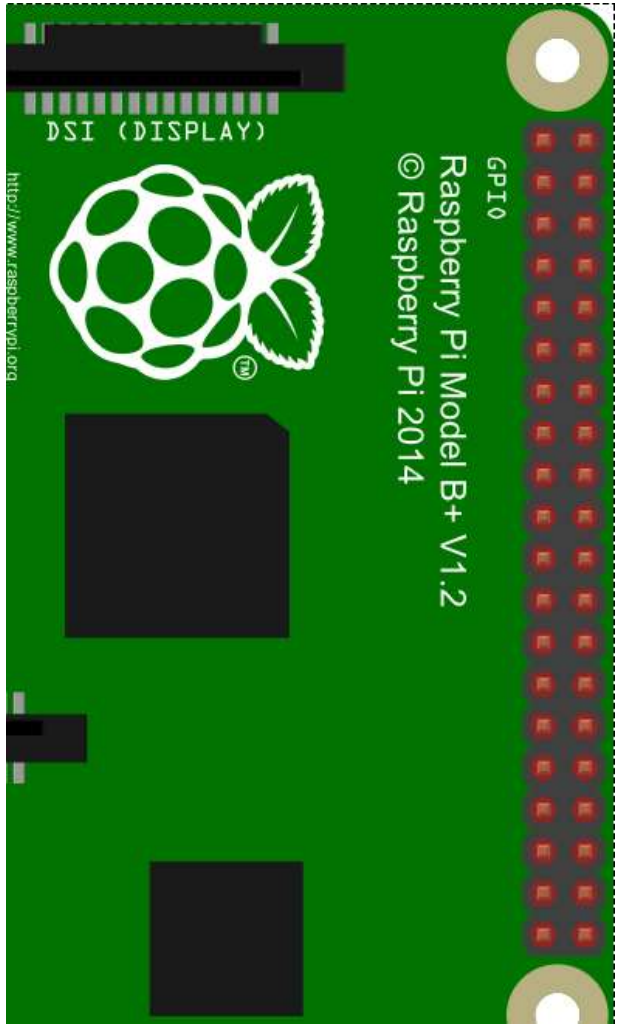
Powering Circuits

- Outgoing Power:
  - Pi provides 3.3 volts (V) and 5 volts
  - Labelled as +3.3V, +5V, or Vcc
  - Like positive terminal on battery
- Return Power:
  - Labelled Ground or GND
  - Like negative terminal on battery
- Typically we wire +3.3 or +5 to red rail and GND to blue rail on breadboard

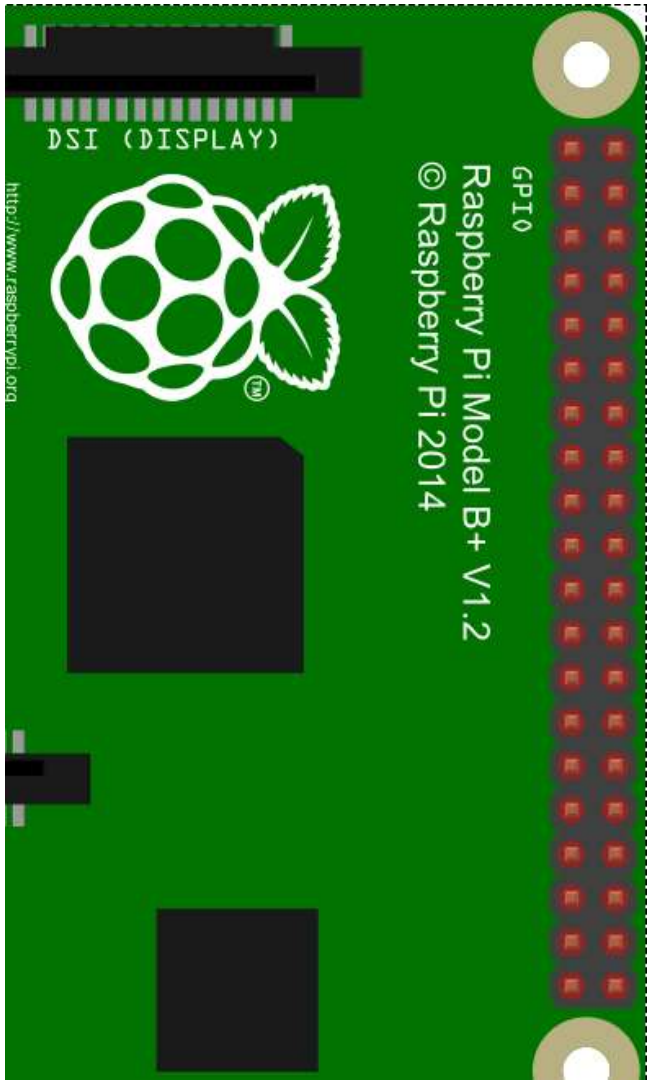


General Purpose Input/Output (GPIO)

- Raspberry Pi provides:
  - A or B: 26 pins
  - A+, B+, or 2B: 40 pins
- 40 pins are backward compatible with 26
- Multiple power and ground pins
- All others are “general purpose” and are programmable
- Output pins send +3.3V
- Some pins have special purposes and can’t be swapped



Powering Circuits



Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Programming GPIO Pins

- For this talk, we'll use Python for readability
- Python libraries are already build in

- At start, must set output mode:
  - Board – physical pin #
  - BCM – GPIO pin #
- At end, must exit cleanly:

```
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)
GPIO.setmode(GPIO.BCM)



GPIO.cleanup()
```

Programming GPIO Pins

- For each GPIO pin used, you must set it up as input or output first:

```
GPIO.setup(3, GPIO.OUT)
GPIO.setup(4, GPIO.IN)
```

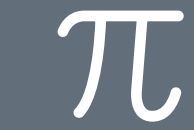
- Write out to the output pins:

```
GPIO.output(3, True) # +3.3V
GPIO.output(3, GPIO.HIGH)

GPIO.output(3, False) # 0V
GPIO.output(3, GPIO.LOW)
```

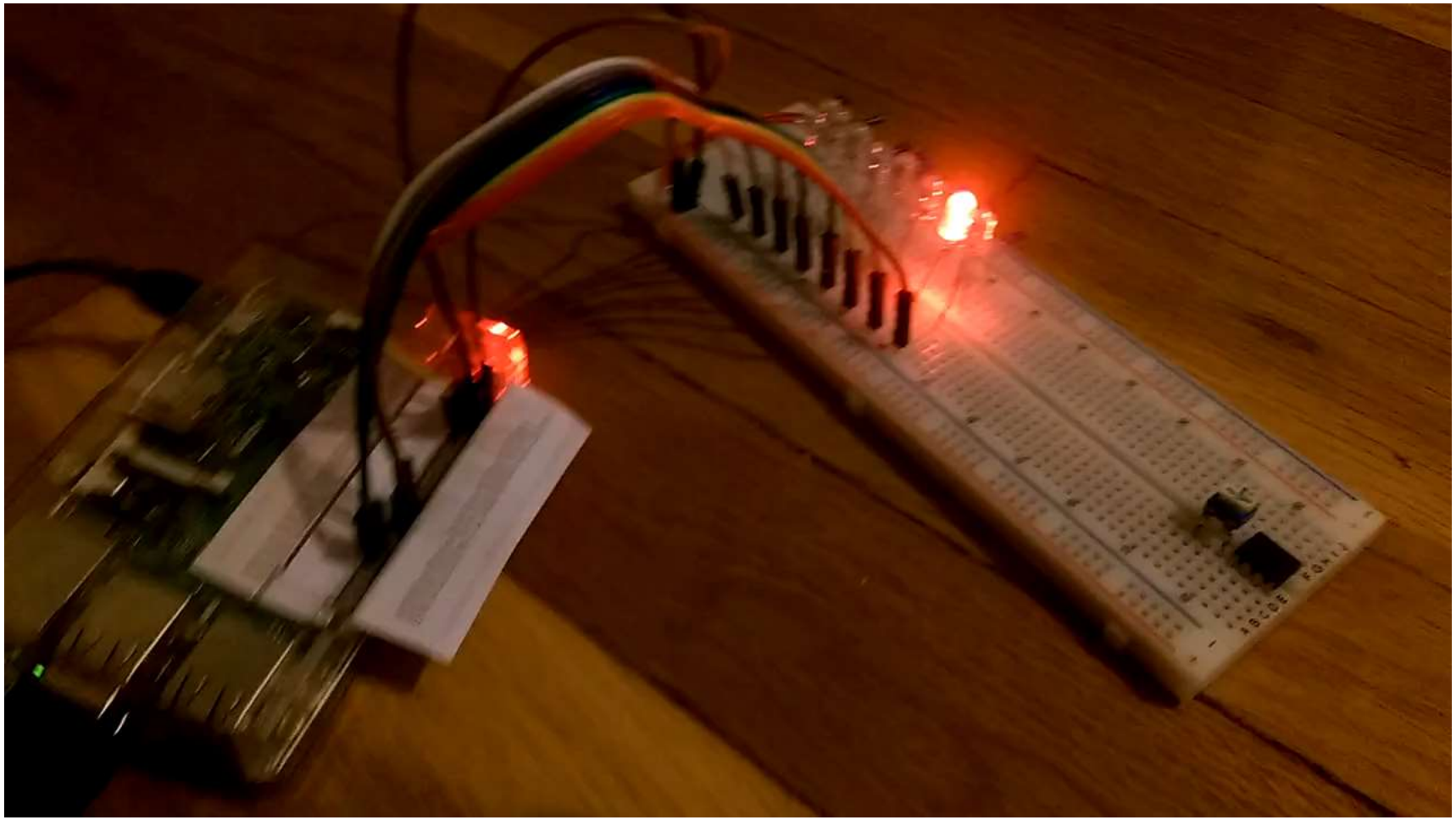
- Read in from input pins:

```
x=GPIO.input(4) # Bool or 0/1
```

Project 1 (LEDs) – Cylon/Knight Rider Lights



Demo



Materials List

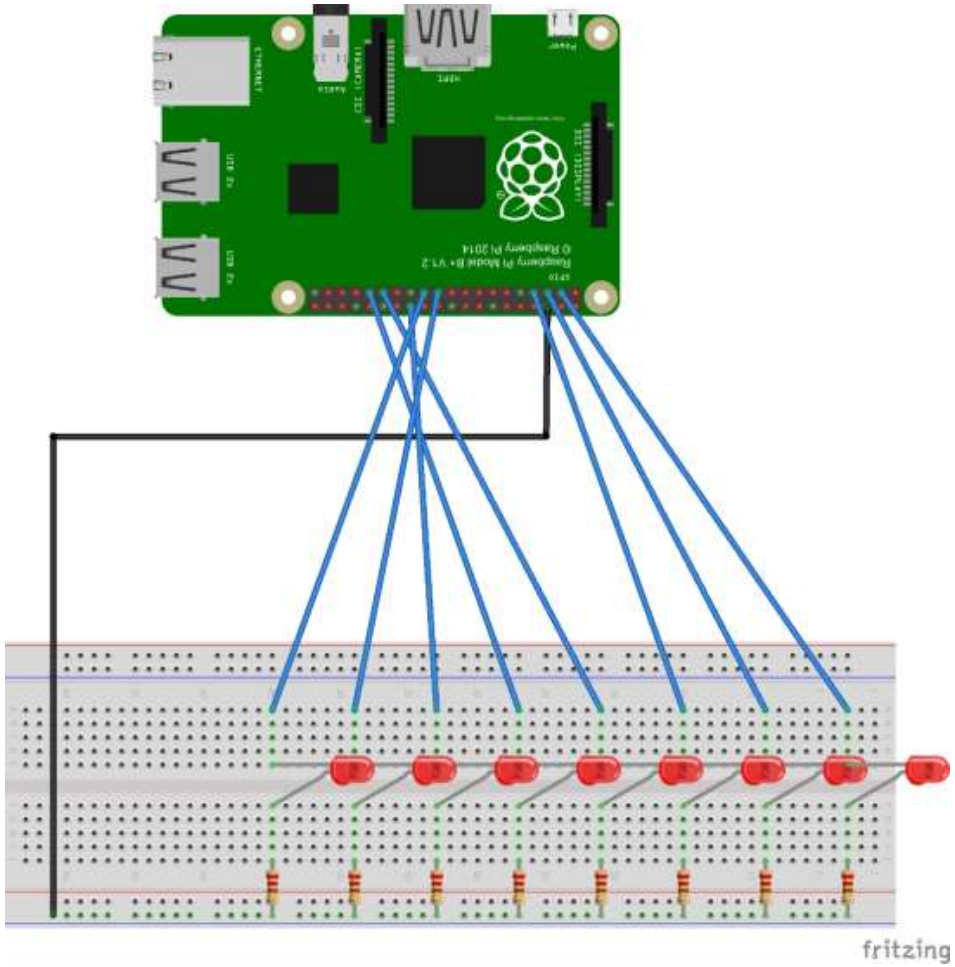
- Raspberry Pi (any model)
- Power supply (I recommend at least 1.5A for this project)
- 1 Breadboard
- 8 light emitting diodes*
- 8 resistors*
- 9 jumper wires

* I'll explain these in a minute

Light Emitting Diodes (LED)

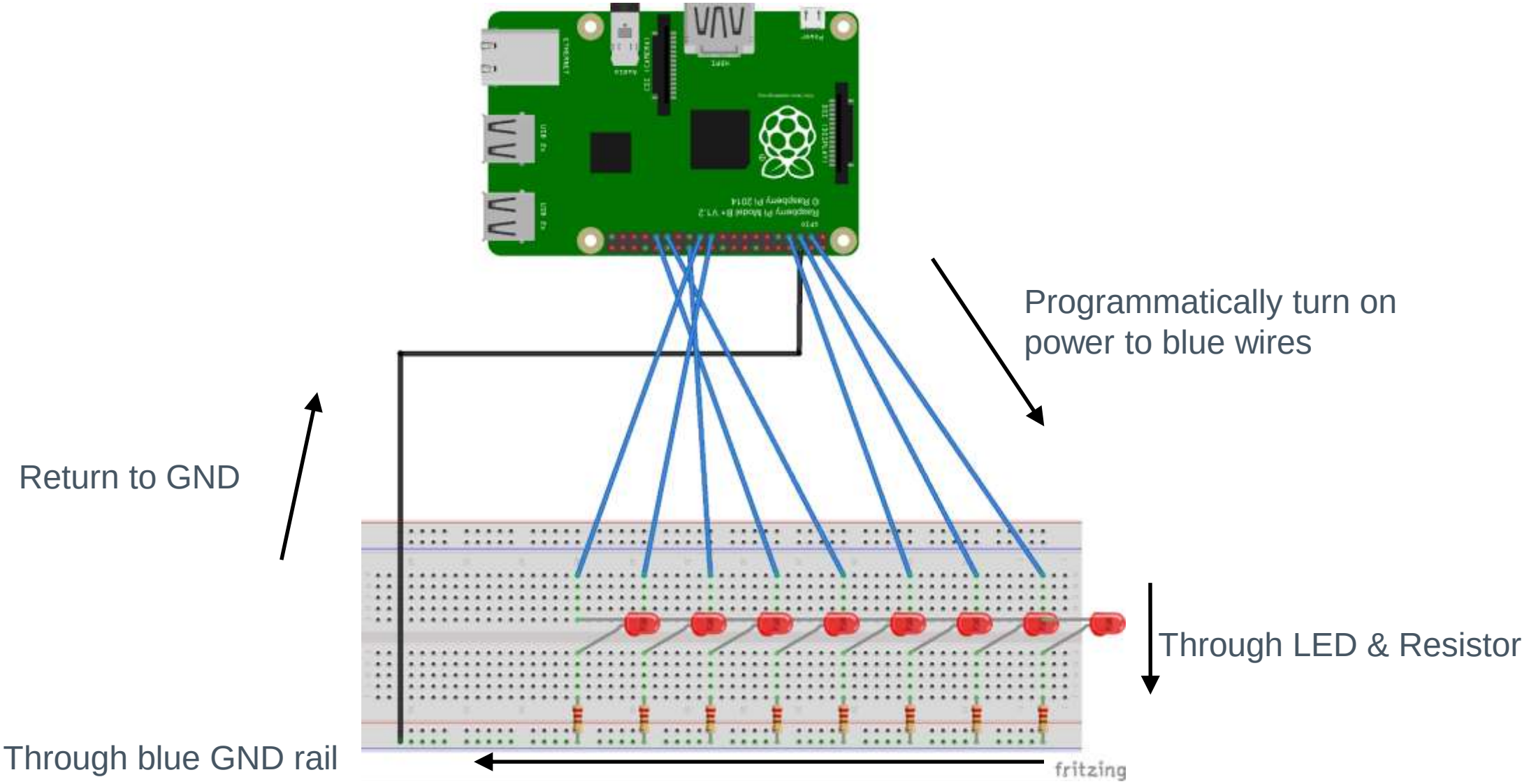
- Diode – Electronic device that allows electricity to flow in one direction
- Anode – longer wire (+)
- Cathode – shorter wire (-)
- When power is applied (correctly), it glows. Otherwise nothing
- Resistor needed otherwise LED will glow bright white then fry
- I recommend just buying LED packs that include resistors with them

Hardware Layout



1. Pi GND -> blue rail
2. Pi GPIO 2-10 -> LED’s anodes (longer wire)
3. LED’s cathode (shorter wire) -> resistor
4. Resistors -> blue rail

Hardware Layout



Code

```
1. # Import libraries
2. import time
3. import RPi.GPIO as GPIO
4.
5. # Create array of GPIO pins
6. pins = [3, 5, 7, 29, 31, 26, 24, 21]
7. # Speed lights will blink (secs)
8. speed = .2
9.
```

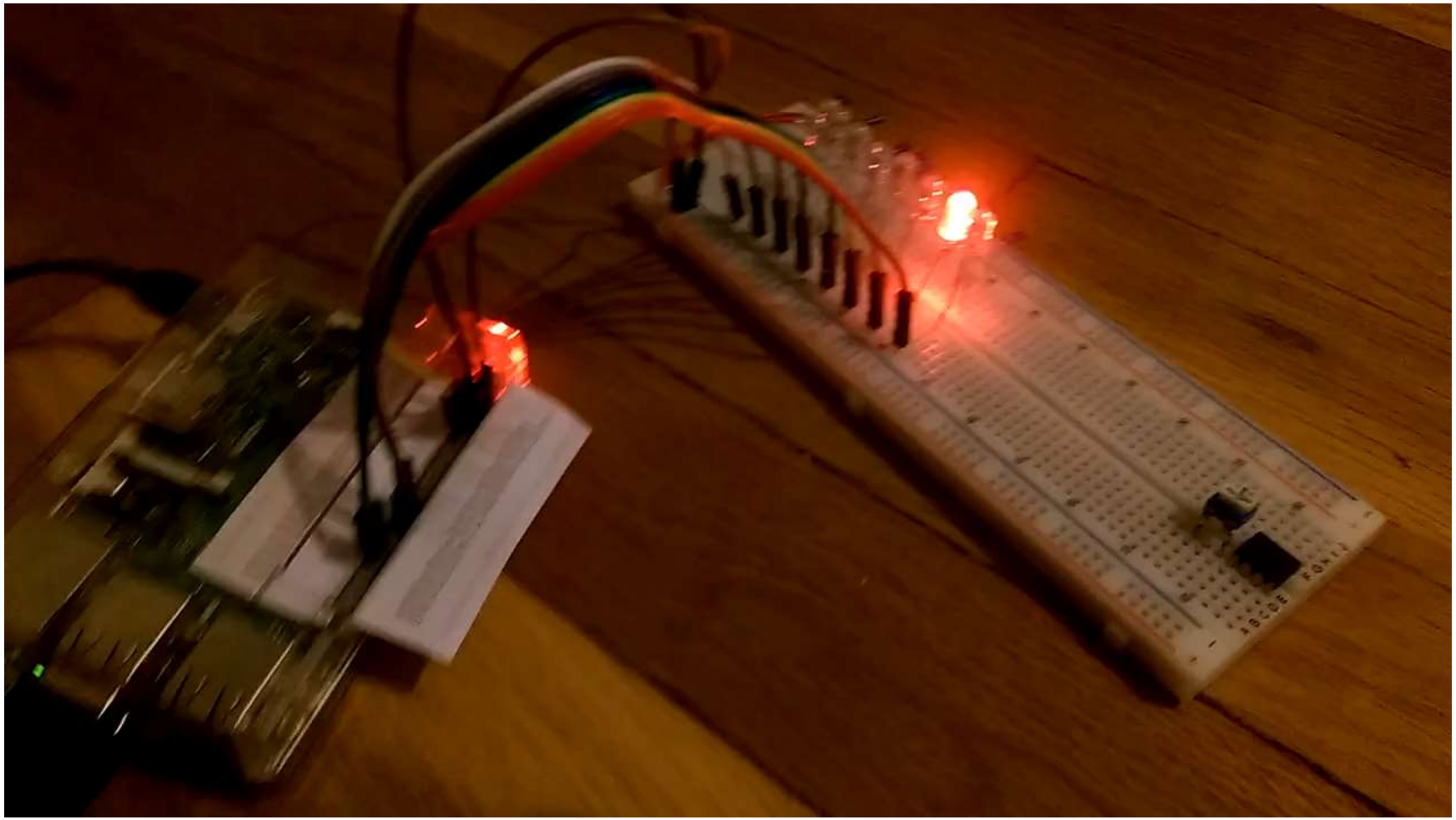
Code

```
10.# Set GPIO pins to board (physical pins) mode
11.GPIO.setmode(GPIO.BOARD)
12.
13.# Set up all pins as output pins
14.for i in pins:
15.  GPIO.setup(i, GPIO.OUT)
16.
```

Code

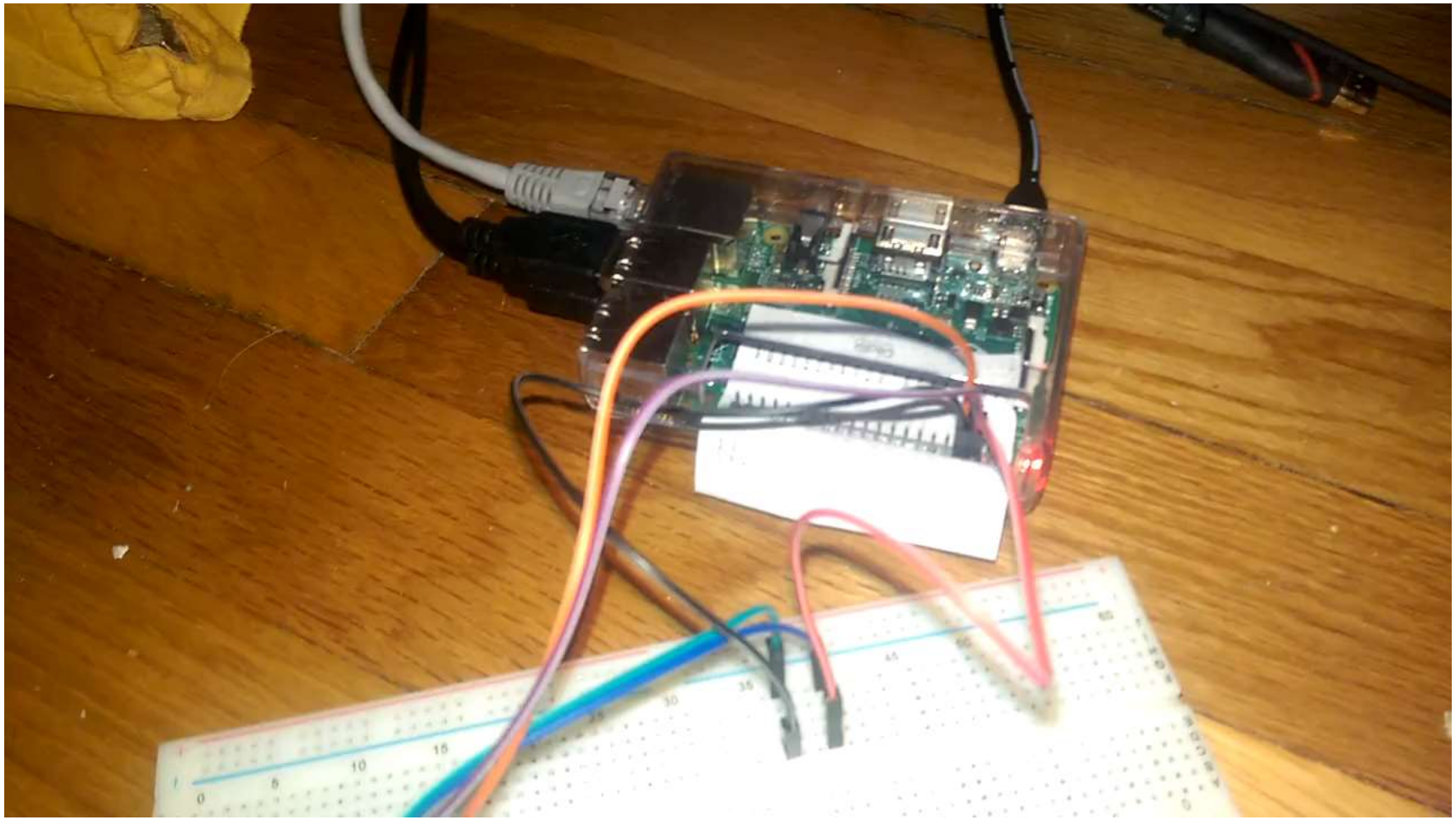
```
17.while True:
18.  for i in range(len(pins)):
19.    GPIO.output(pins[i], True)
20.    time.sleep(speed)
21.    GPIO.output(pins[i], False)

22.  for i in range(len(pins)-1, -1, -1):
23.    GPIO.output(pins[i], True)
24.    time.sleep(speed)
25.    GPIO.output(pins[i], False)
```

Demo



Project 2 (LCD Screen) – LCD Twitter Feed

Demo



Materials List

- Raspberry Pi and power supply
- 4 jumper wires
- 1 Breadboard
  (needed if you don't have right wires)
- 1 LCD screen
  - They vary in size/shape/color/etc. Use one with I2C or Serial capabilities

LCD Screen

- Uses 4x20 characters per screen
- Each character is 8x5 pixels
- Can add custom characters
- Really easy to use with serial or I2C chip on the back
- Does have small delay when writing/erasing screen

I^2C

- Inter-Integrated Circuit
- Communications protocol for circuitry
- Uses master/slave bus
- Each device has an address and can be contacted by it
- Communication shares the same 2 wires for all devices connected: SDA, SCL
- A lot of support is built into various hardware (Raspberry Pi, Arudinos, Beaglebones, etc.)

I²C

- Must be enabled in Raspberry Pi first
- Run:

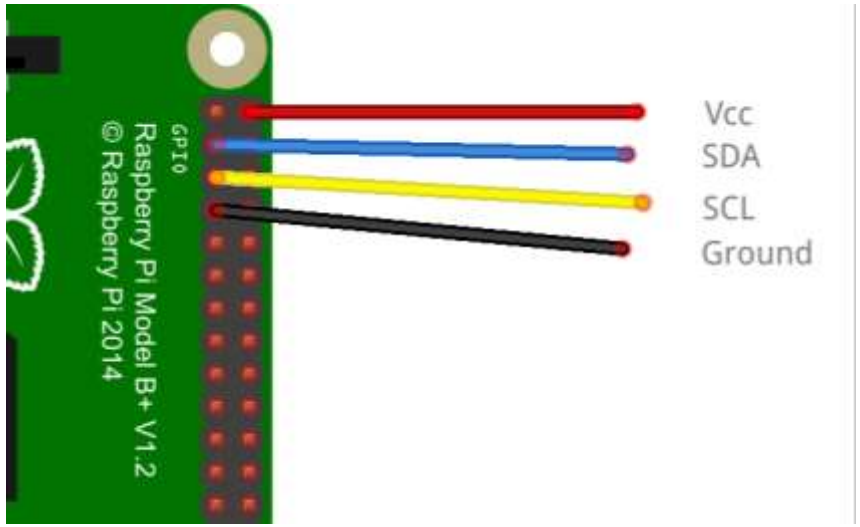
```
sudo raspi-config
```

  - Choose Advanced options
  - Choose I2C
  - Choose to enable it
  - Choose to enable it on startup

Wiring

- No breadboard needed if using female-to-female wires
- Use breadboard as jumping point otherwise

Raspberry Pi	LCD Screen
+5V	Vcc
SDA (pin 3)	SDA
SCL (pin 5)	SCL
Ground	Gnd



Code

```
1. # Requires RPi_I2C_driver.py for I2C and LCD
2.
3. import RPi_I2C_driver
4. from time import *

5. # Create the object
6. mylcd = RPi_I2C_driver.lcd()
```

Code

```
7. # Print happy welcome message on lines 1 and 3
8. mylcd.lcd_display_string("Welcome to the", 1)
9. mylcd.lcd_display_string(“Kansas City Dev”, 3)
10.mylcd.lcd_display_string(“Conference!!", 4)
11.sleep(2) # 2 sec delay

12.# Erase screen
13.mylcd.lcd_clear()
14.
```

Code

```
14.x = 0
15.y = 0
16.changex = 1
17.changey = 1
18.
```

Code

```
14.while True:
15.    x = x + changex
16.    y = y + changey
17.    if x < 1:
18.       changex = 1
19.       x = 1
20.    if x >= 4:
21.       changex = -1
22.    if y < 1:
23.       y = 0
24.       changey = 1
25.    if y >= 19:
26.       changey = -1
```

Code

```
27.   mylcd.lcd_clear()

28.   mylcd.lcd_display_string_pos("o", x, y)

29.   sleep(.5)
```

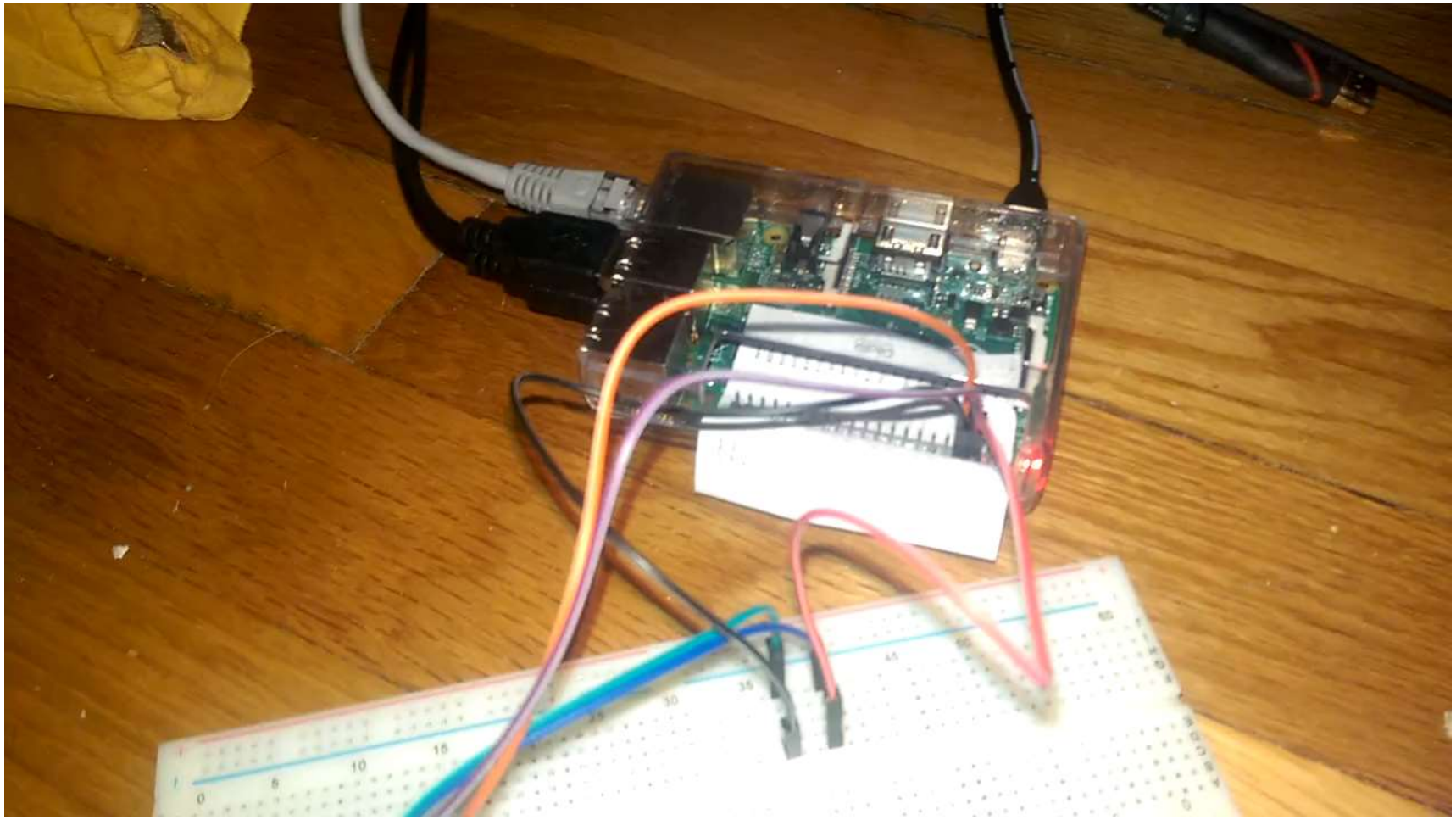
LCD Twitter Feed

```
1. from twython import Twython
2.
3. APP_KEY = '...'
4. APP_SECRET = '...'
5.
6. twitter = Twython(APP_KEY, APP_SECRET,
   oauth_version=2)
7. ACCESS_TOKEN = twitter.obtain_access_token()
8.
9. twitter = Twython(APP_KEY, access_token=ACCESS_TOKEN)
```

LCD Twitter Feed

```
10. tweets = twitter.search(q='#KCDCpi')

11. print(tweets["u'statuses'"][0]["u'user'"]["u'screen_name'"])
12. print(tweets["u'statuses'"][0]["u'text'"])
```

Demo



Project 3 (Sensors) – Wall-Avoiding Robot

So…

- … it’s not fully finished... ☹
- … but I’ll go over it anyway… ☺

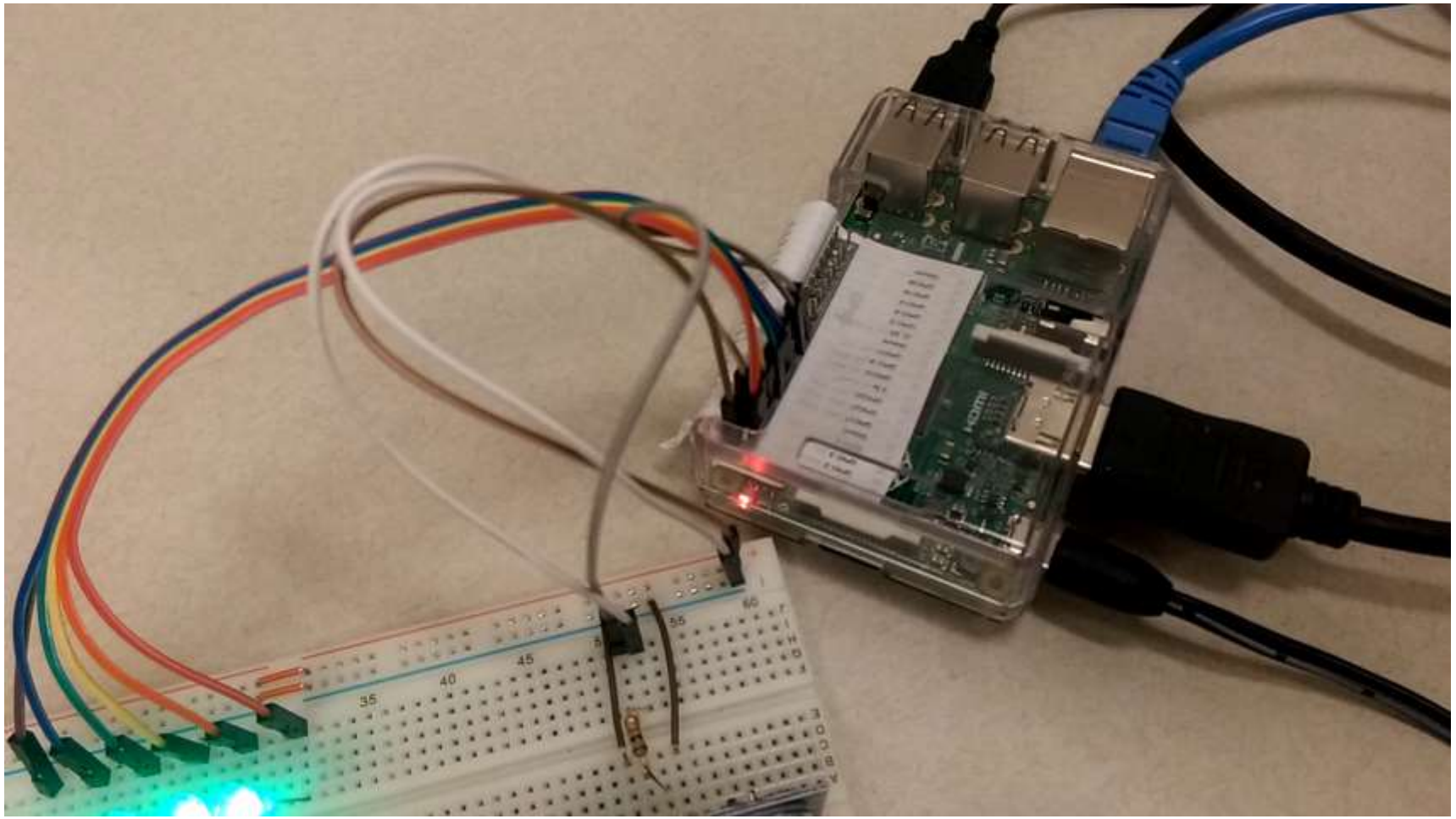
Materials List

- Raspberry Pi and power supply
- Chassis of some sort
- 1 Breadboard
- 1 ultrasonic sensors (HC-SR04)
- Motor controller
- Some form of battery pack for motors and Pi

Ultrasonic Sensor

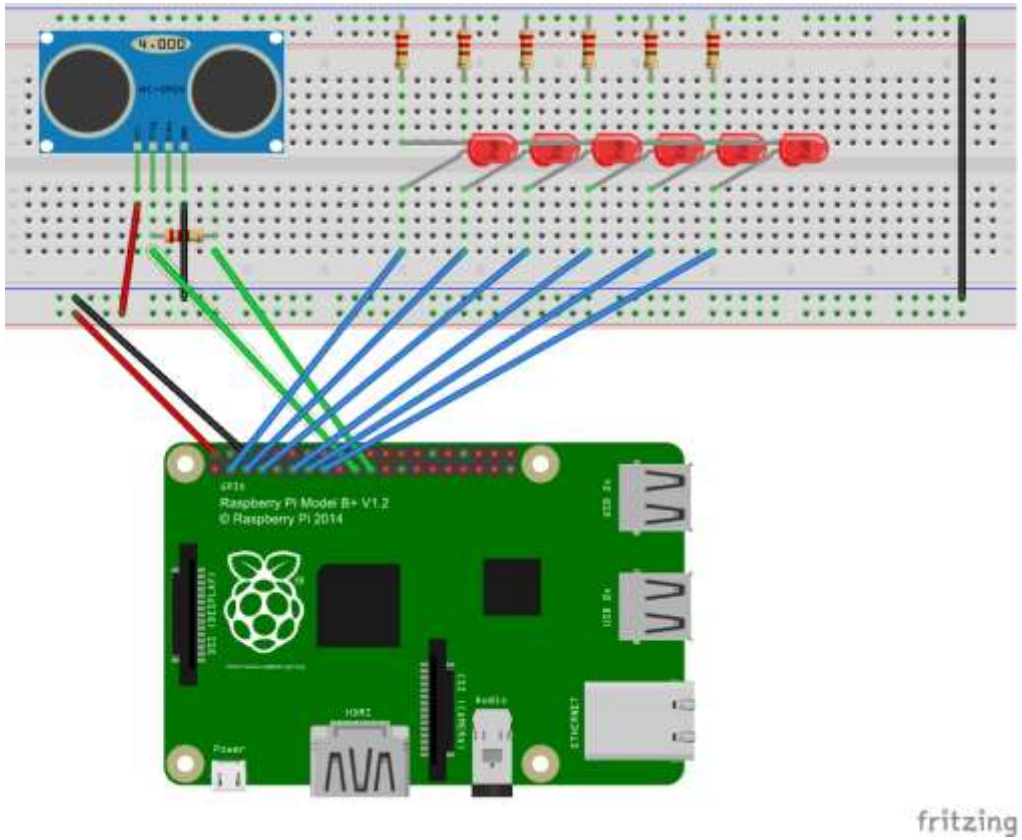
- Ultrasonic sensors are one of a variety of sensors you can use
- They detect distance with sound
- Ultrasonic = outside of human hearing range
- Pulses 40KHz signal from “T” side
- “R” side listens for it
- Time taken in between is time for sound to travel

Demo



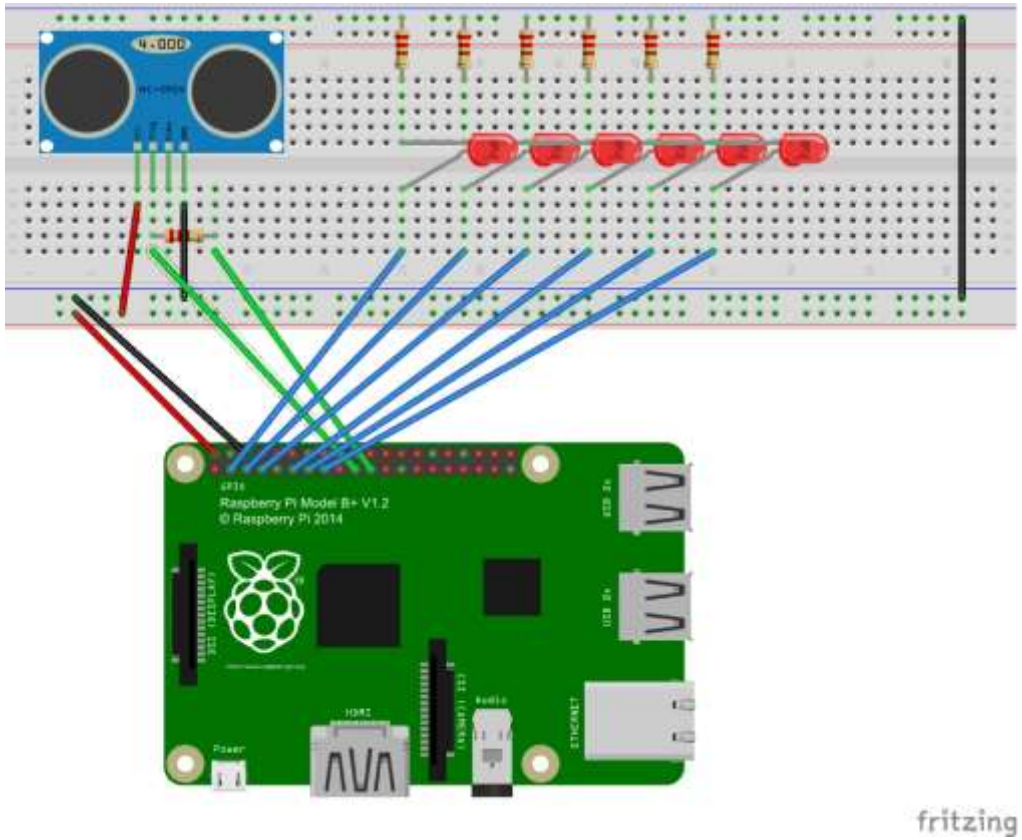
Wiring

- Breadboard Ground -> Pi Ground
- Breadboard +5V -> Pi +5V
- Sensor Gnd -> Gnd rail
- Sensor +5V -> +5V rail
- LEDs similar to Project 1



Wiring

- LED anodes to pins 3, 5, 7, 11, 13, 15 (9 is a ground, skip it)
- Sensor Trigger to 19
- Sensor Echo to 21



Code

```
1. import time
2. import RPi.GPIO as GPIO
3.
4. # Set up the pins
5. uTrig = 19
6. uEcho = 21
7. leds = [3, 5, 7, 11, 13, 15]
8.
9. GPIO.setup(GPIO.BOARD)
10.for i in leds:
11.    GPIO.setup(i, GPIO.OUT)
12.    GPIO.output(i, GPIO.LOW)  # off
```

Code

```
13.def reading(trigger, echo):
14.  GPIO.setmode(GPIO.BOARD)
15.  GPIO.setup(trigger, GPIO.OUT)
16.  GPIO.setup(echo ,GPIO.IN)
17.  GPIO.output(trigger, GPIO.LOW)
18.
19.  time.sleep(0.3)
20.
21.
```

Code

```
21.  GPIO.output(trigger, True)
22.  time.sleep(0.00001)
23.  GPIO.output(trigger, False)

24.  signaloff = 0
25.  while GPIO.input(echo) == 0:
26.     signaloff = time.time()
27.
```

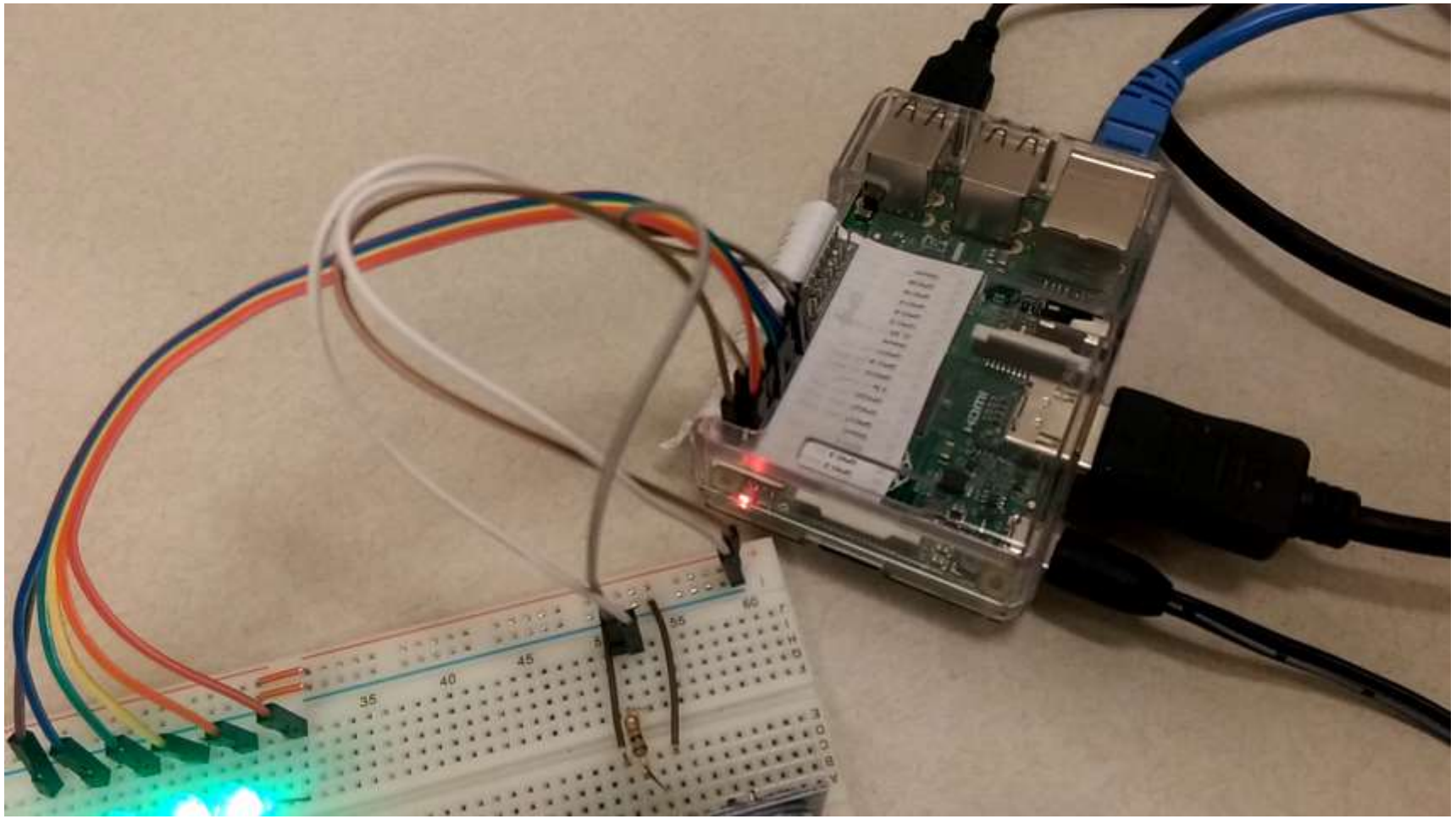
Code

```
28.  signalon = 0
29.  while GPIO.input(echo) == 1:
30.     signalon = time.time()
31.
32.  timepassed = signalon - signaloff
33.
34.  distance = timepassed * 17000
35.
36.  return distance
37.
```

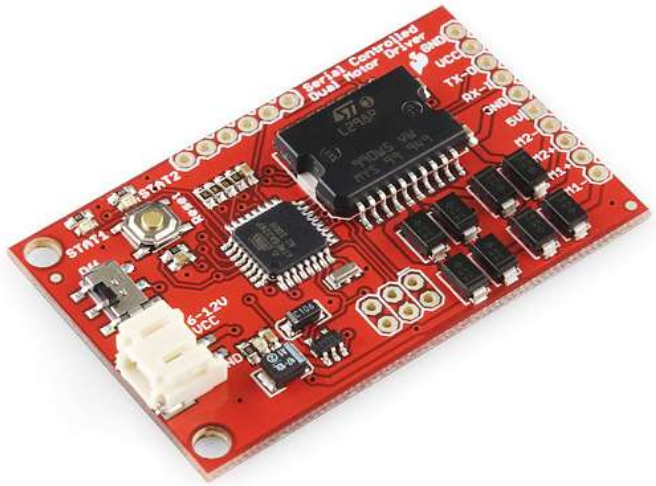
Code

```
38.# Main program
39.while True:
40.   # Grab a reading
41.   read = reading(uTrig, uEcho)
42.
43.   for i in range(0, 6):
44.      print(i)
45.      if read > i * 3:
46.         GPIO.output(leds[i], GPIO.HIGH)
47.      else:
48.         GPIO.output(leds[i], GPIO.LOW)
49.
```

Demo



Motor Driver/Controller



- Pi's pins don't provide enough power
- Need secondary power source
- Motor controller takes commands to control power to motors

More Information and Conclusion

More Project Ideas

- Customized cookie maker machine
- Tracking cat/dog door
- Show tweets/emails on LCD screen
- AI-programmed remote control cars
- Robots… of any variety…
- Turn on/off heater/AC when a room is occupied/unoccupied
- Ruby-based garage door opener
- Chicken coup protection device

More Project Ideas

- Customized cookie maker machine
- Tracking cat/dog door
- Show tweets/emails on LCD screen
- AI-programmed remote control cars
- Robots… of any variety…
- Turn on/off heater/AC when a room is occupied/unoccupied
- Ruby-based garage door opener
- Chicken coup protection device

› THE POSSIBILITIES ARE ENDLESS!!

Conclusion

- Raspberry Pi is more than just a PC, it can interface with other electronics
- You've seen some sample projects
- You've seen how to code items like sensors, LCD screens, and LEDs
- Hopefully you're inspired to use these ideas to do your own projects

Contact Me

Twitter: @geekygirlsarah

Email: sarah@sarahwithee.com

Slides/Resources: sarahwithee.com/raspberrypi

- Feel free to contact with questions or to show off projects
- Check out the projects before you leave!