

Pre Set Up

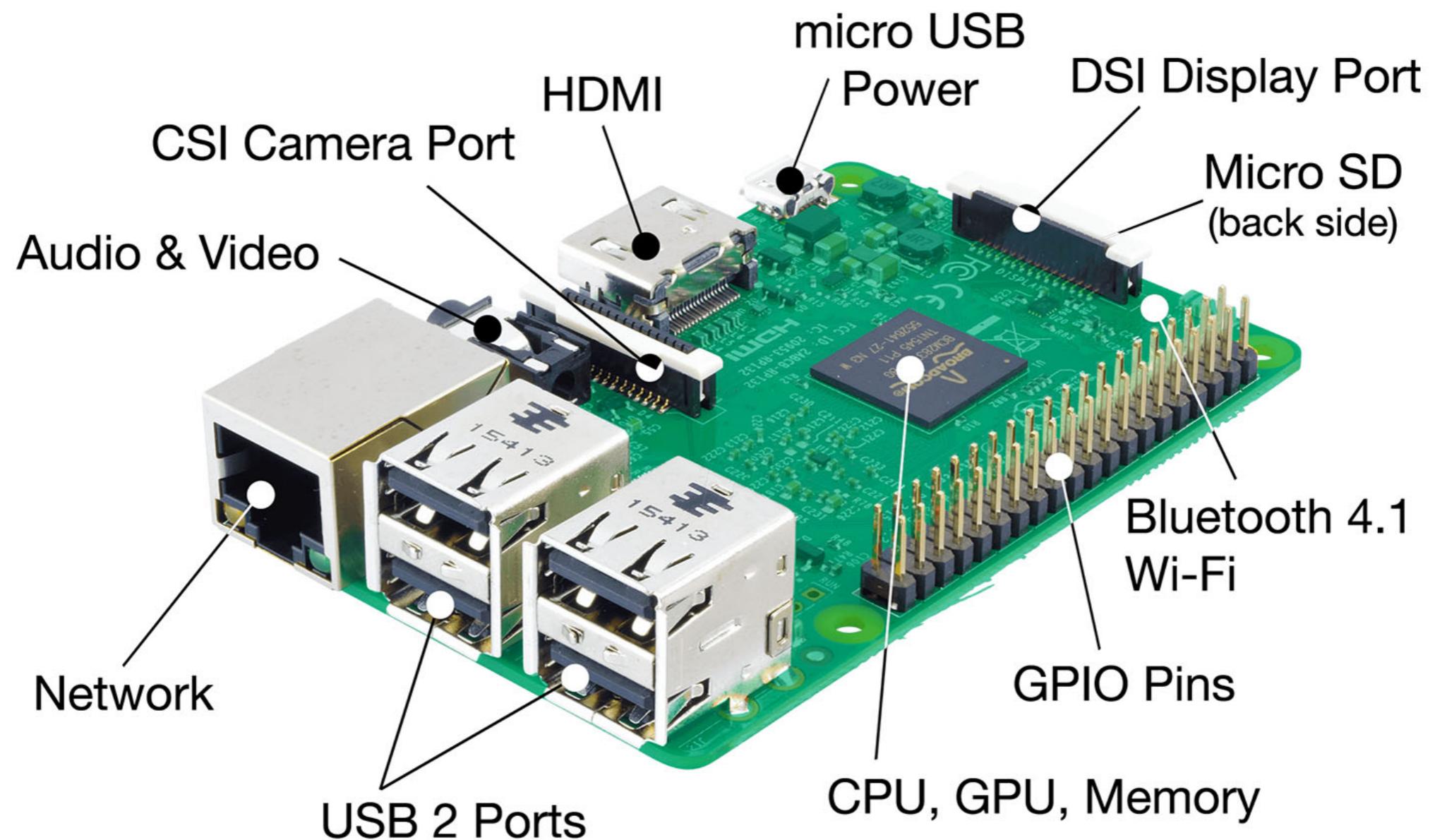
- HDMI Cable to HDMI slot
- Keyboard to one USB slot
- Mouse to another USB slot
- Power to micro USB Power Input

Coding Workshop

Outline

- Raspberry Pi
- Python on Pi
- Circuit design
- Programme the Gadgets
- Mini Project - Smart Home

What is Raspberry Pi?



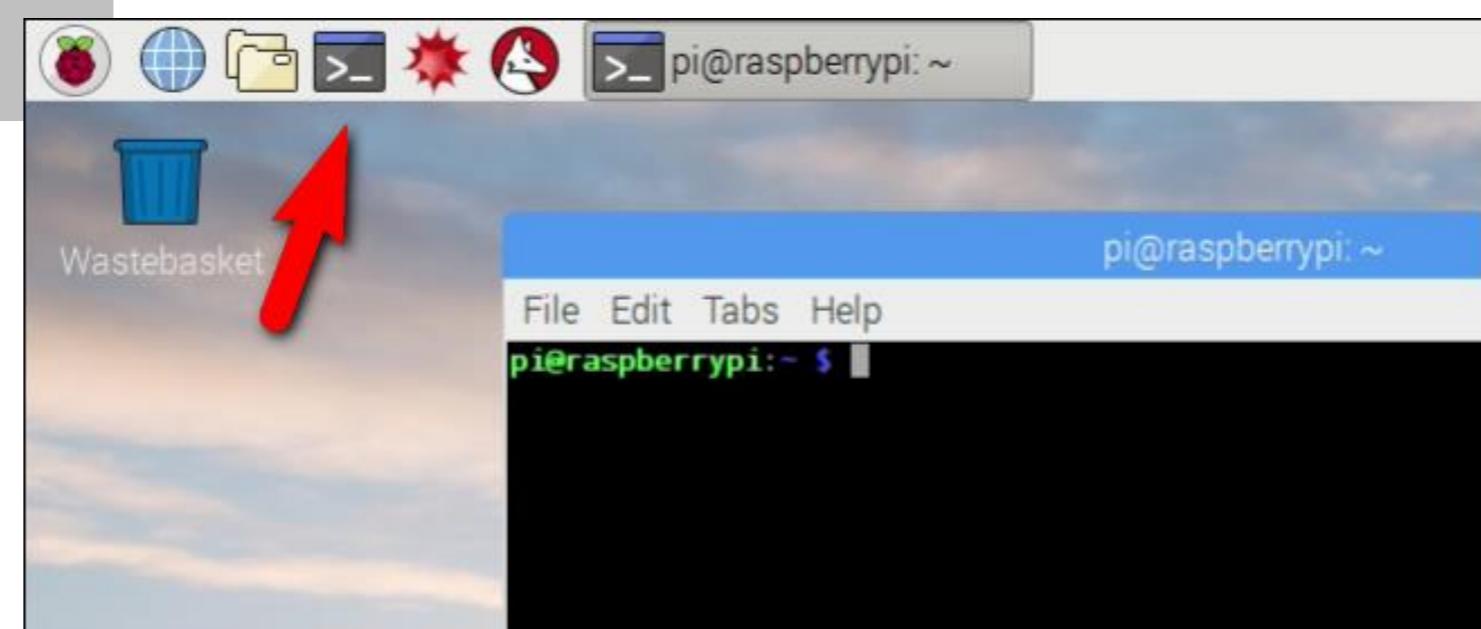
1. “The Raspberry Pi is a **credit-card sized computer..**”
2. “able to interact with the outside world” => sensors, electronic gadgets
3. How does the interaction happen?
 - OS - SD card. Raspbian OS
 - I/O interface. GPIO

Python on Pi

Python Programming on Pi

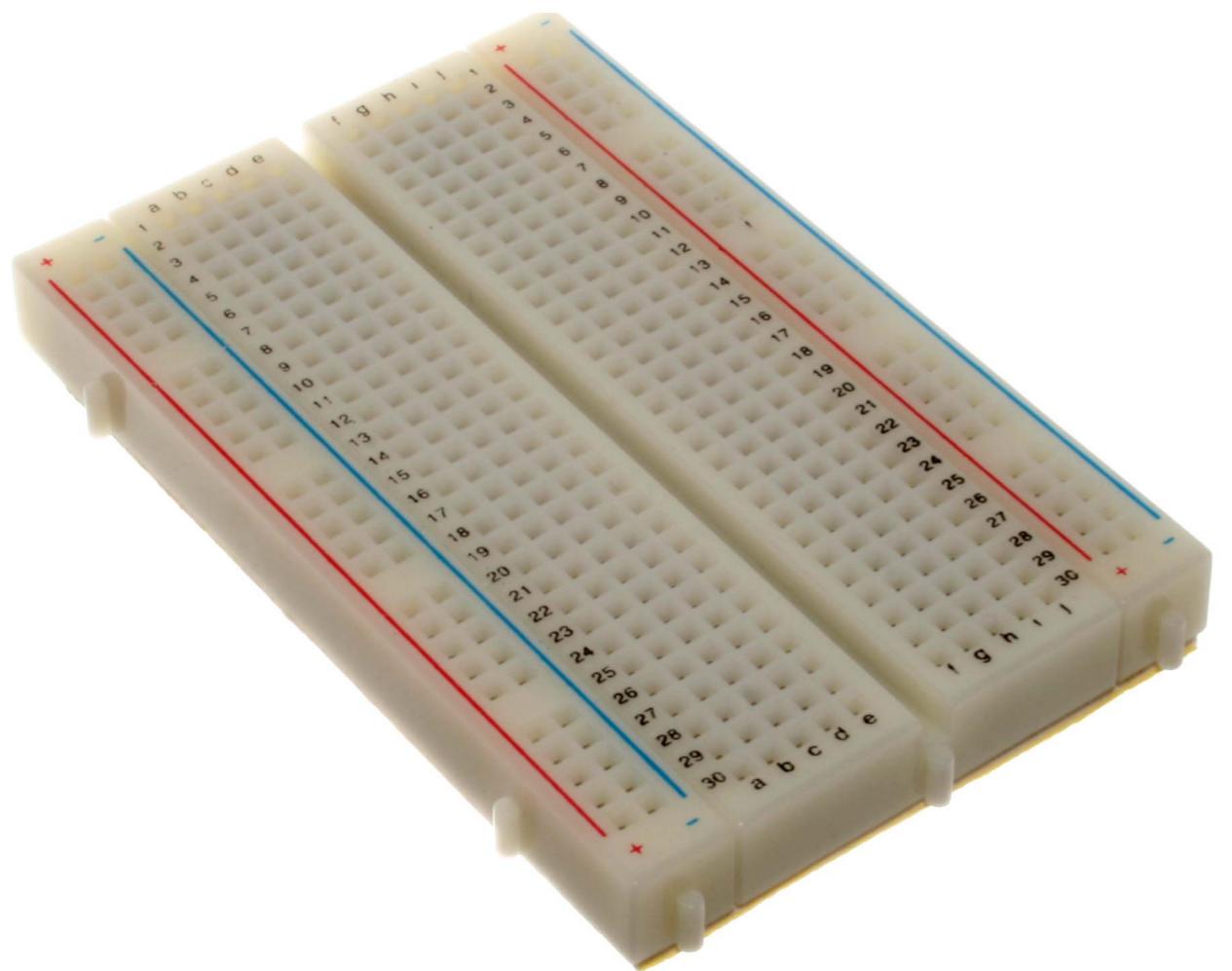


```
cd Desktop  
cd CodingWorkshop  
python helloworld.py
```



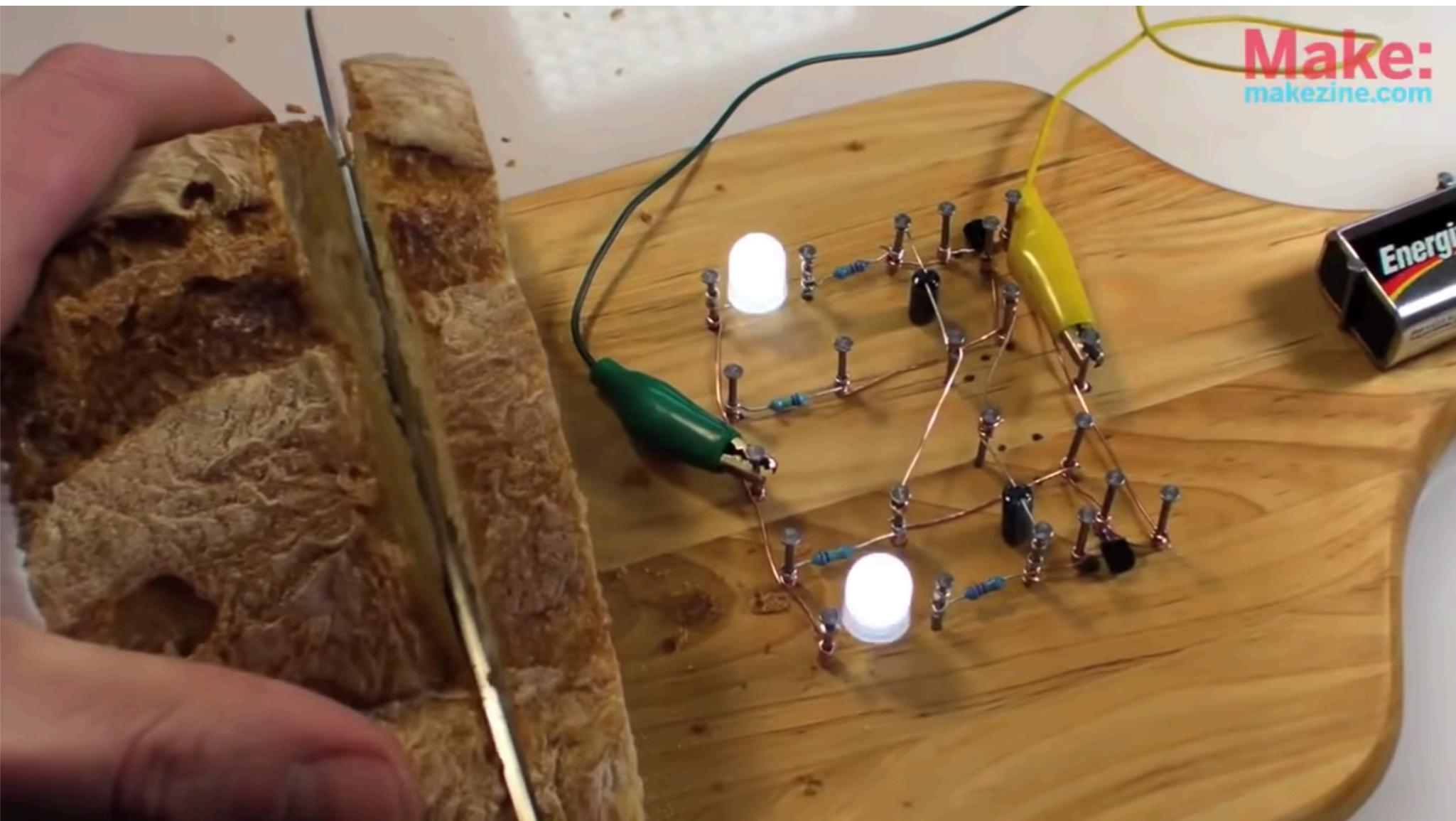
Circuit Design

- Bread Board
- Circuit Design Process
- GPIO T-extension
- Jumper wire



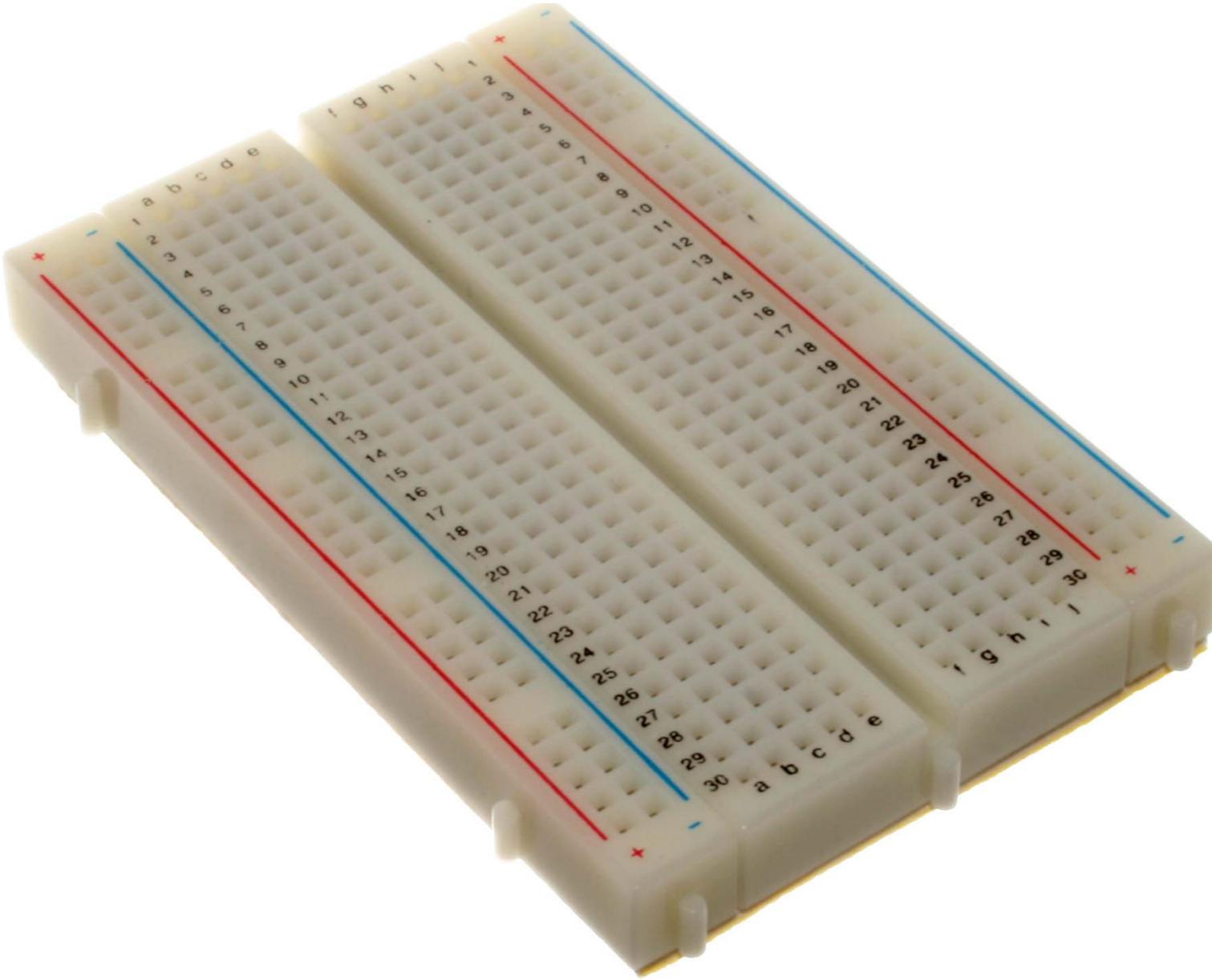
Bread Board?

Left: By oomlout - BREB-01 (Breadboard), CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=19867043>
Right: By Evan Swigart from Chicago, USA - Homemade White Bread with Strawberry Jam, CC BY 2.0, <https://commons.wikimedia.org/w/index.php?curid=11626953>



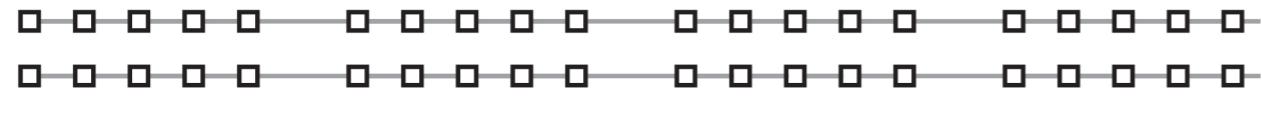
Bread Board

Collin's Lab: The REAL Breadboard, <https://www.youtube.com/watch?v=HrG98HJ3Z6w&gl=SG&hl=en-GB>



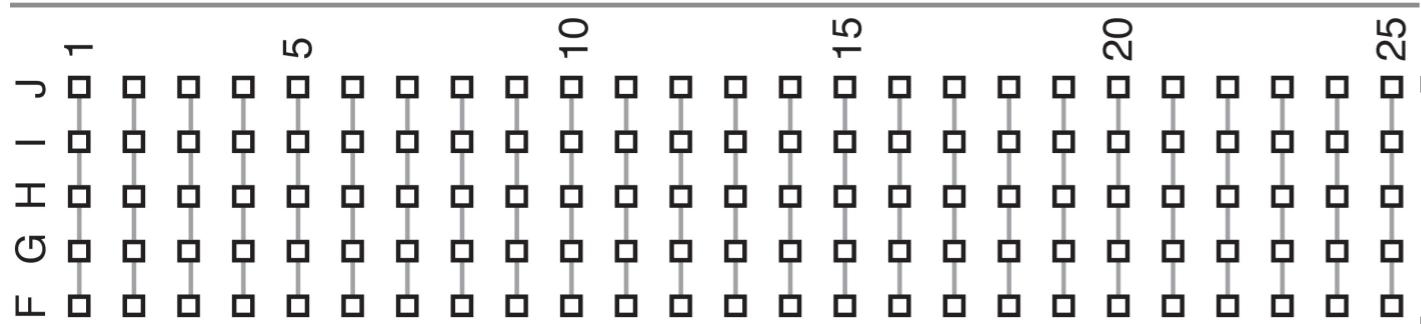
Solder-less Bread Board

By oomlout - BREB-01 (Breadboard), CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=19867043>

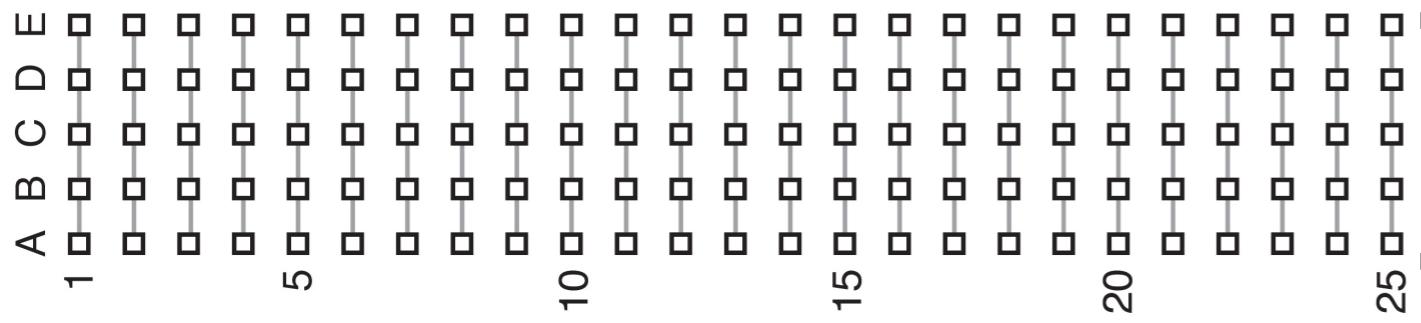


↙ All the holes in this row are connected

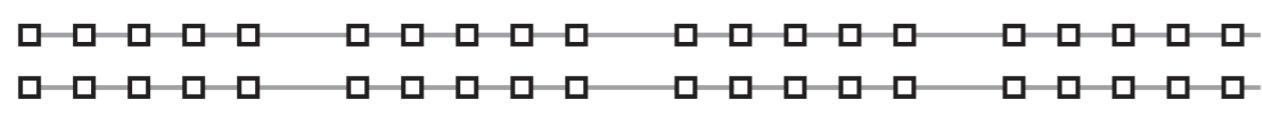
↙ All the holes in this row are connected



} All the holes each column
are connected

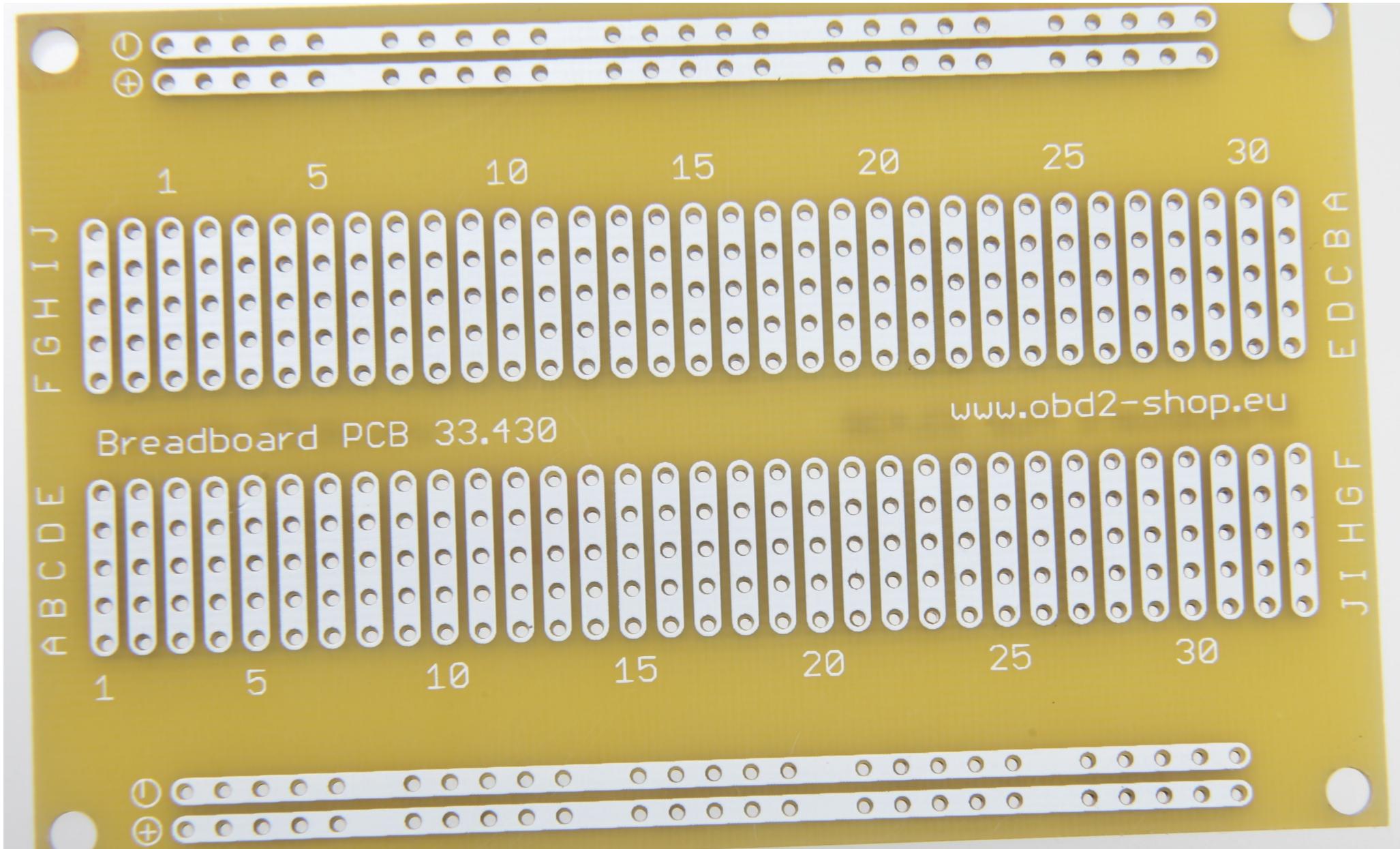


} All the holes each column
are connected



↙ All the holes in this row are connected

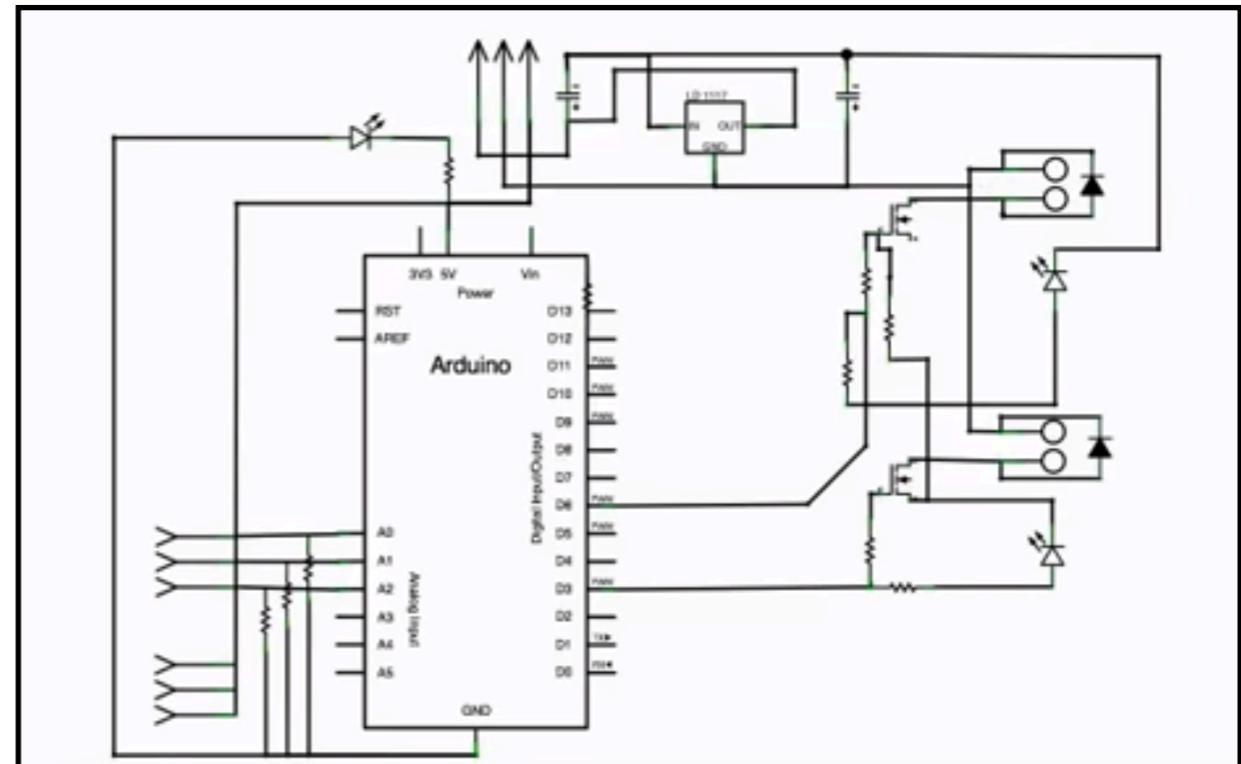
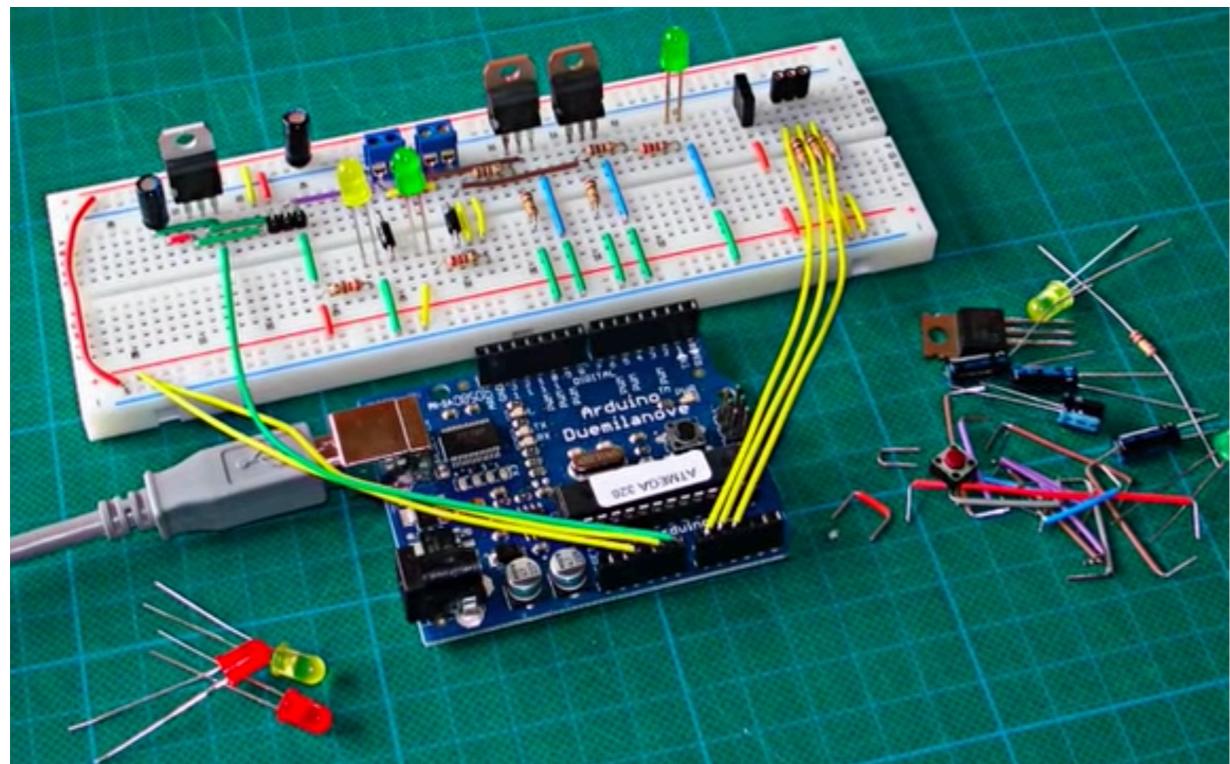
↙ All the holes in this row are connected



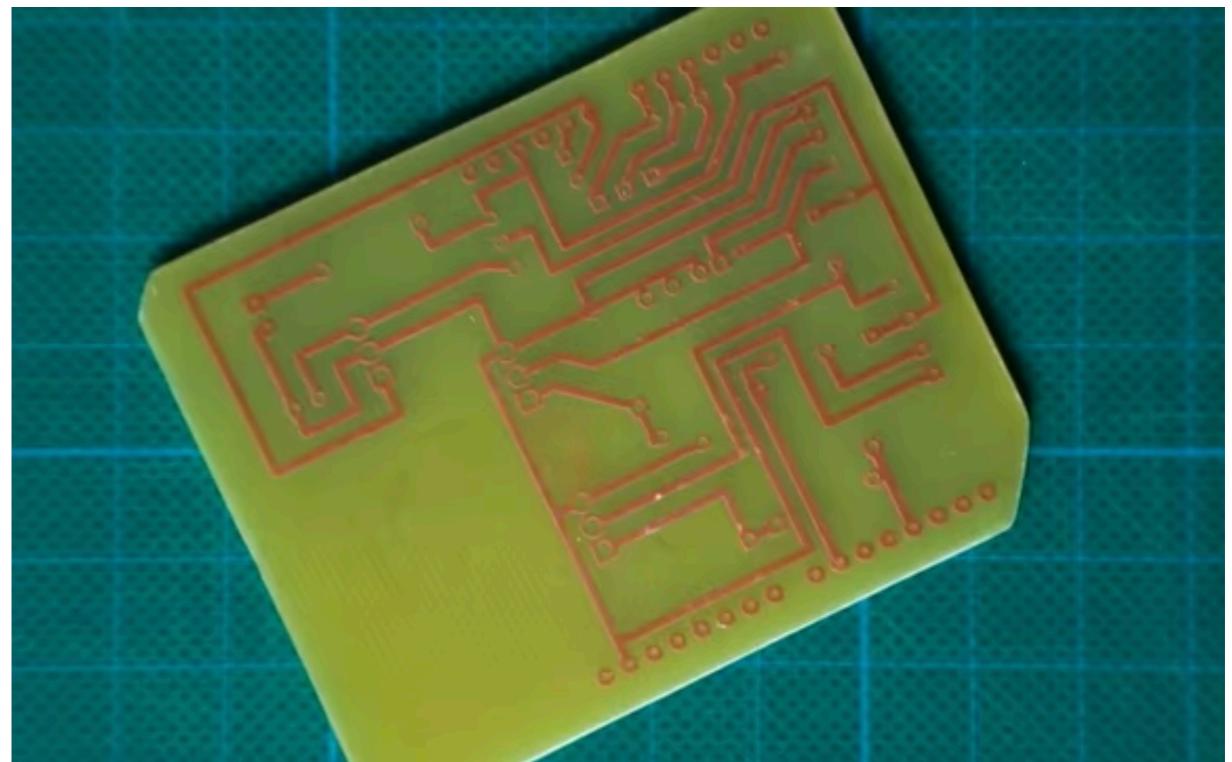
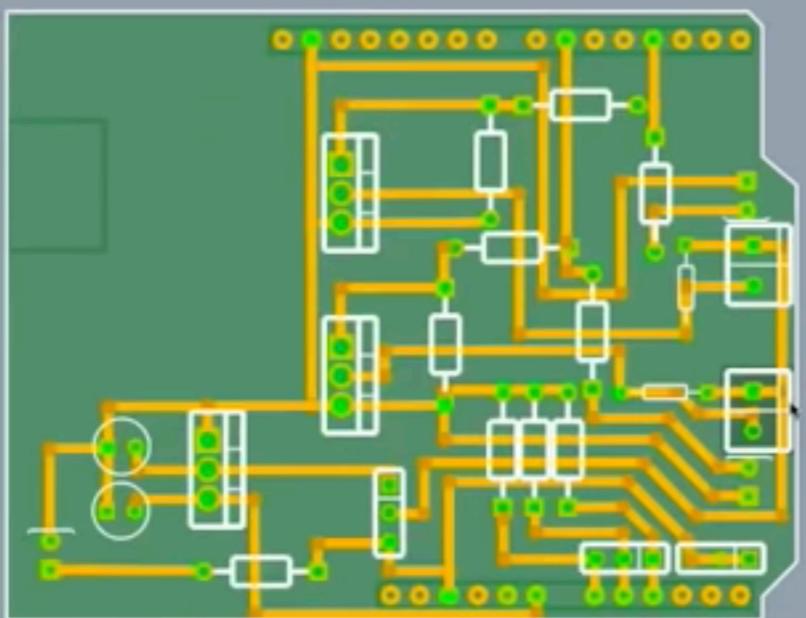
Printed Circuit Board

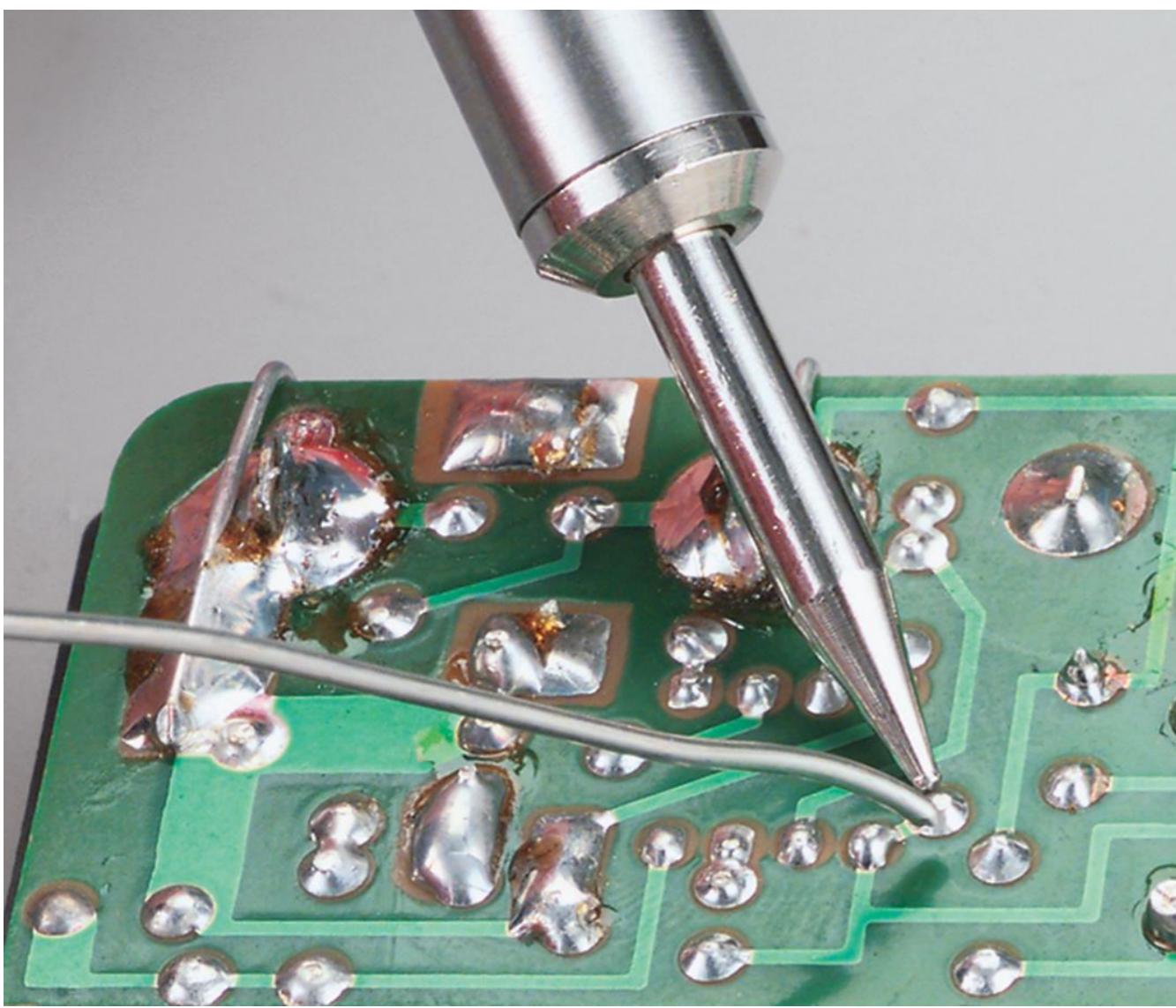
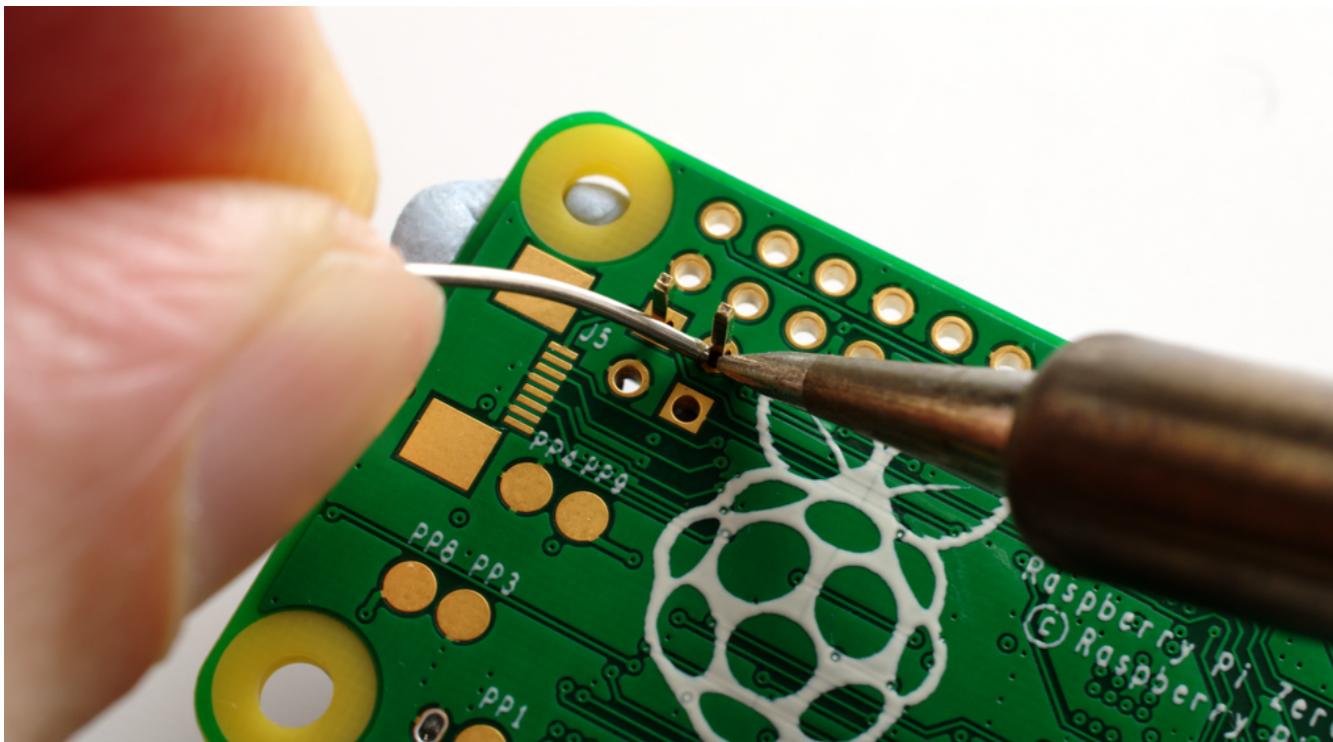
By Florian Schäffer - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=45535472>

From Prototype to Commercial Electronics



From Prototype to Commercial Electronics



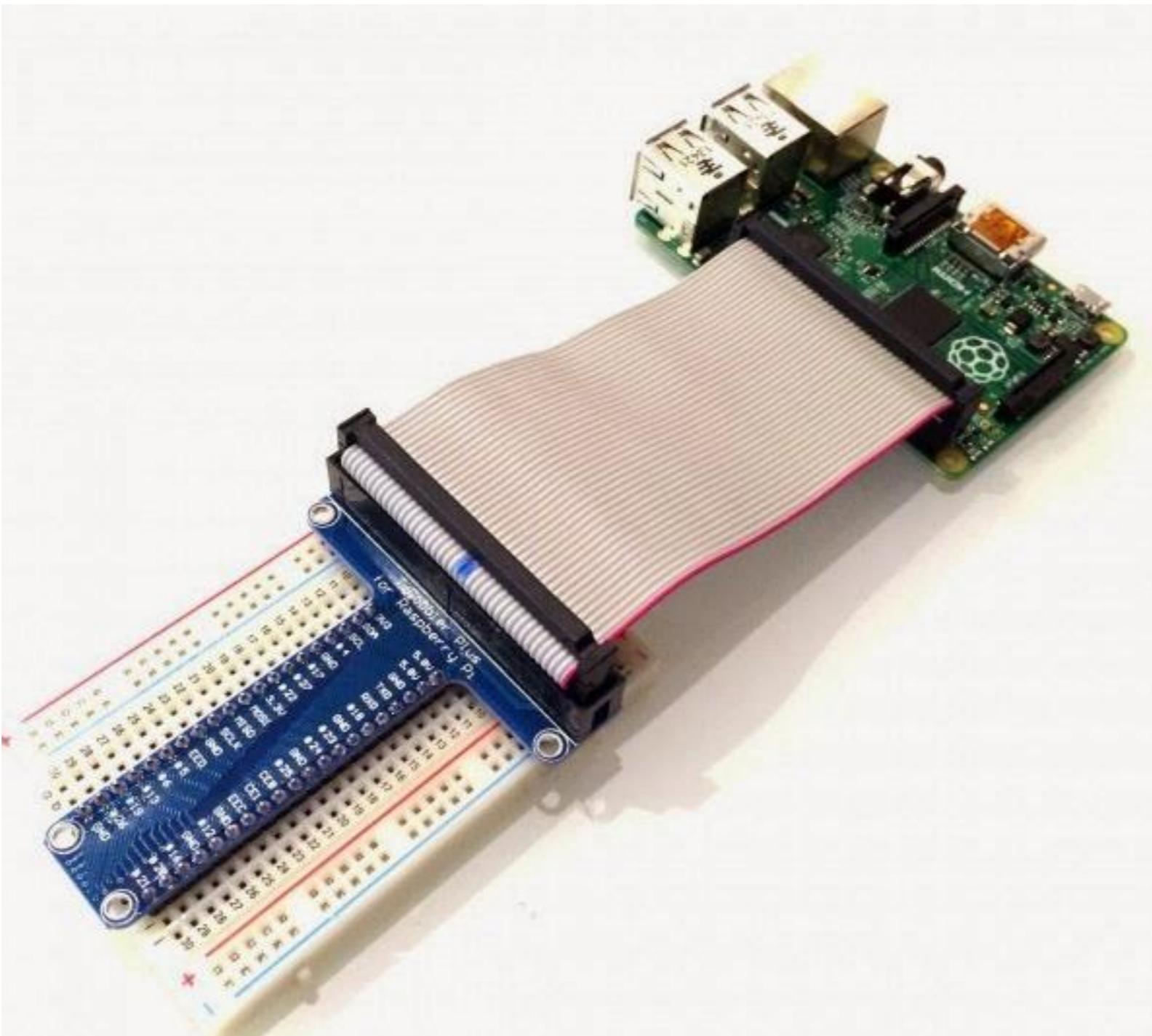


Top left: Gareth Halfacree from Bradford, UK [CC BY-SA 2.0 (<https://creativecommons.org/licenses/by-sa/2.0/>)]

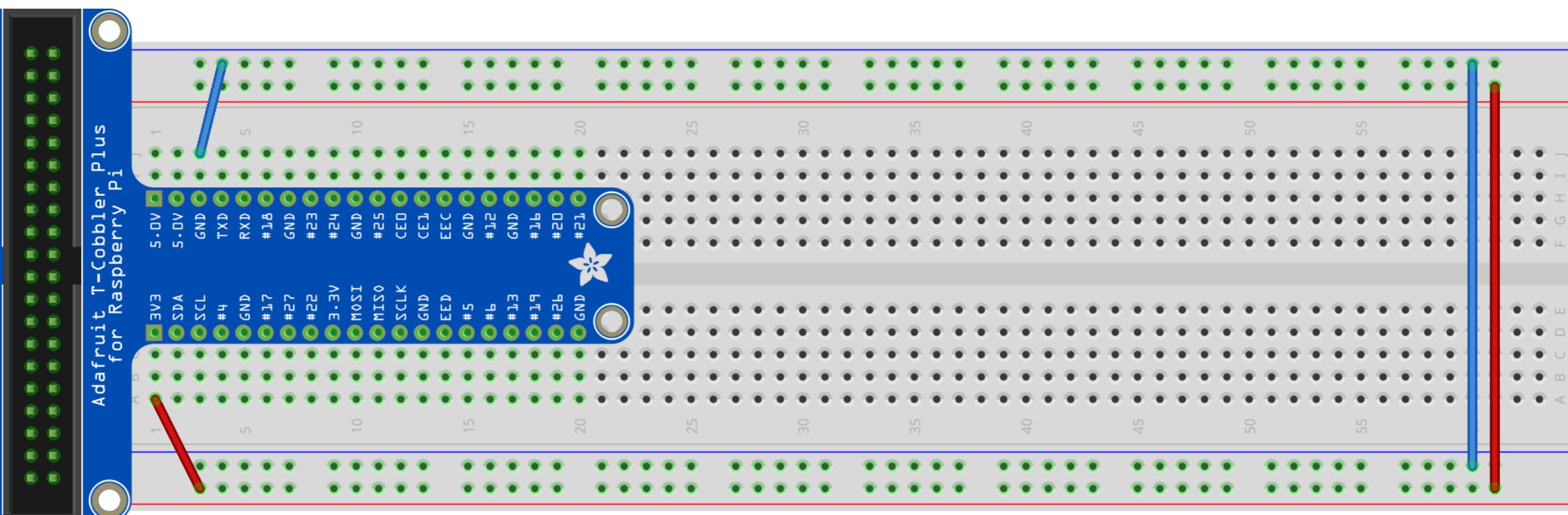
Botton left: <https://m.media-amazon.com/images/S/aplus-media/vc/a40c0904-3a4c-427f-86cd-95a04a4bbf87.jpg>

Right: <https://pxhere.com/en/photo/1058819>

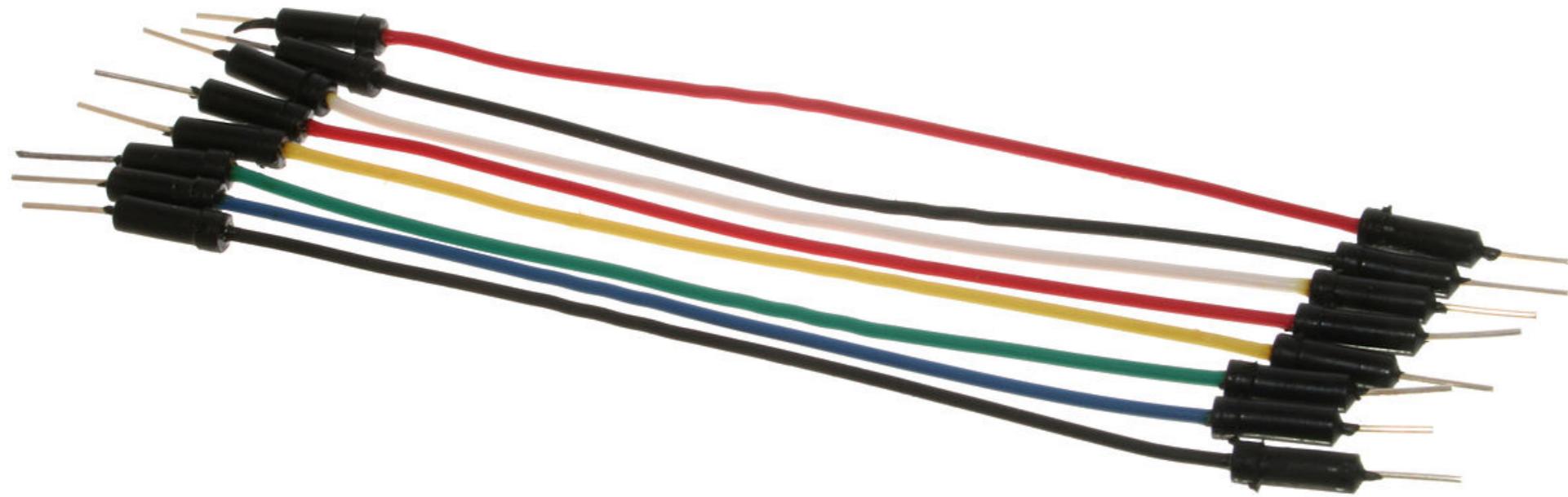
TODO Initial Set Up



Initial Power Connection



- Connect the 3.3V row to the red column and the GND to the blue column
- Connect both blue columns and both red columns so that they are the same



Jumper Wire

Top: By oomlout - A few Jumper Wires, CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=19866418>
Bottom: By oomlout - Jumper Wires with Crocodile Clips (x10) - JUMP-03 Uploaded by bomazi, CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=19866590>

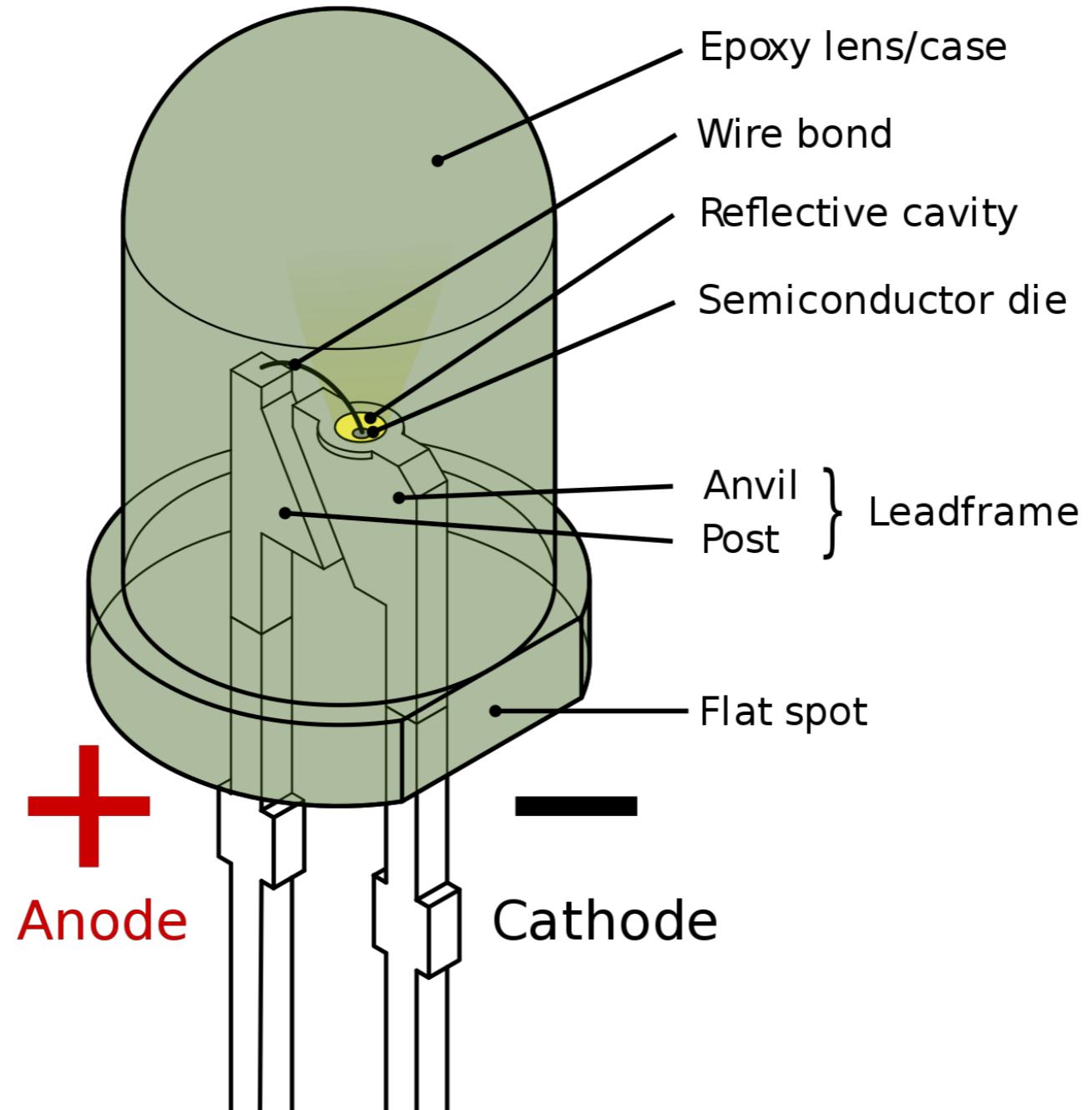


Jumper Wire

Top: By oomlout - A few Jumper Wires, CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=19866418>
Bottom: By oomlout - Jumper Wires with Crocodile Clips (x10) - JUMP-03 Uploaded by bomazi, CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=19866590>

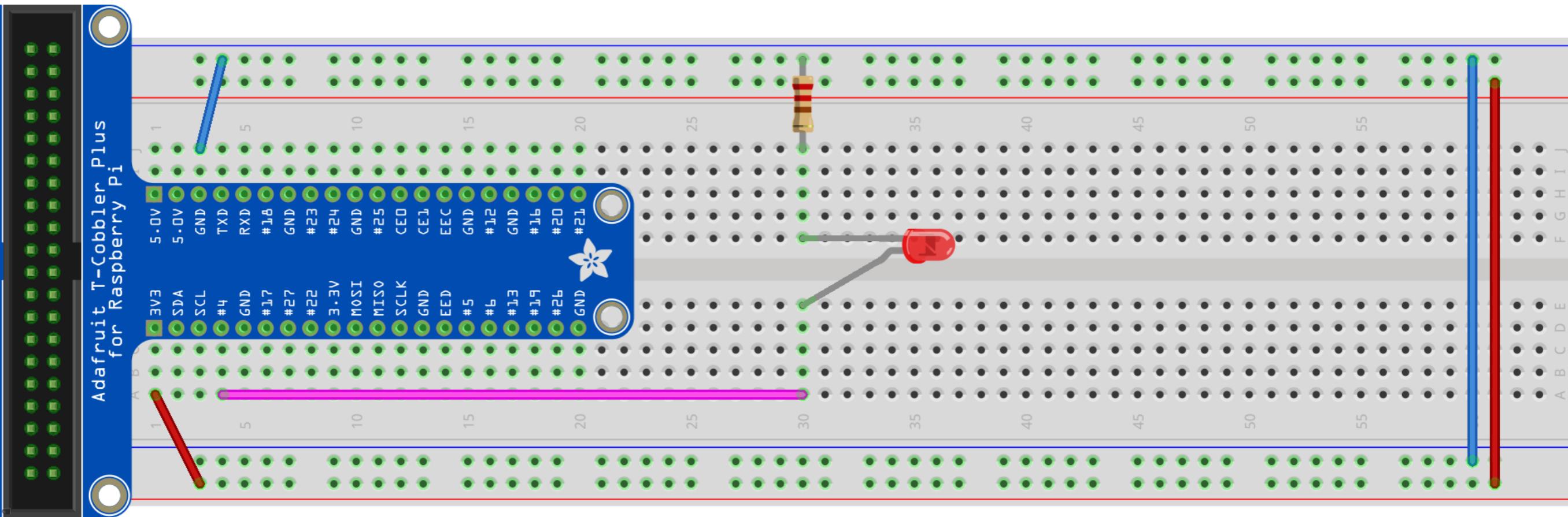
Gadgets

- LED (Light-emitting Diode)
- LDR (Light-dependant Resistor)
- Switch
- Buzzer [analog output]
- 7 Segment Display



Light-emitting Diode

LED - Circuit



- Resistor from ground to a row
- Wire from pin #4 to a different row
- LED connecting both the rows with the longer side connecting to pin #4

LED - Code

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

GPIO.setup(4,GPIO.OUT)

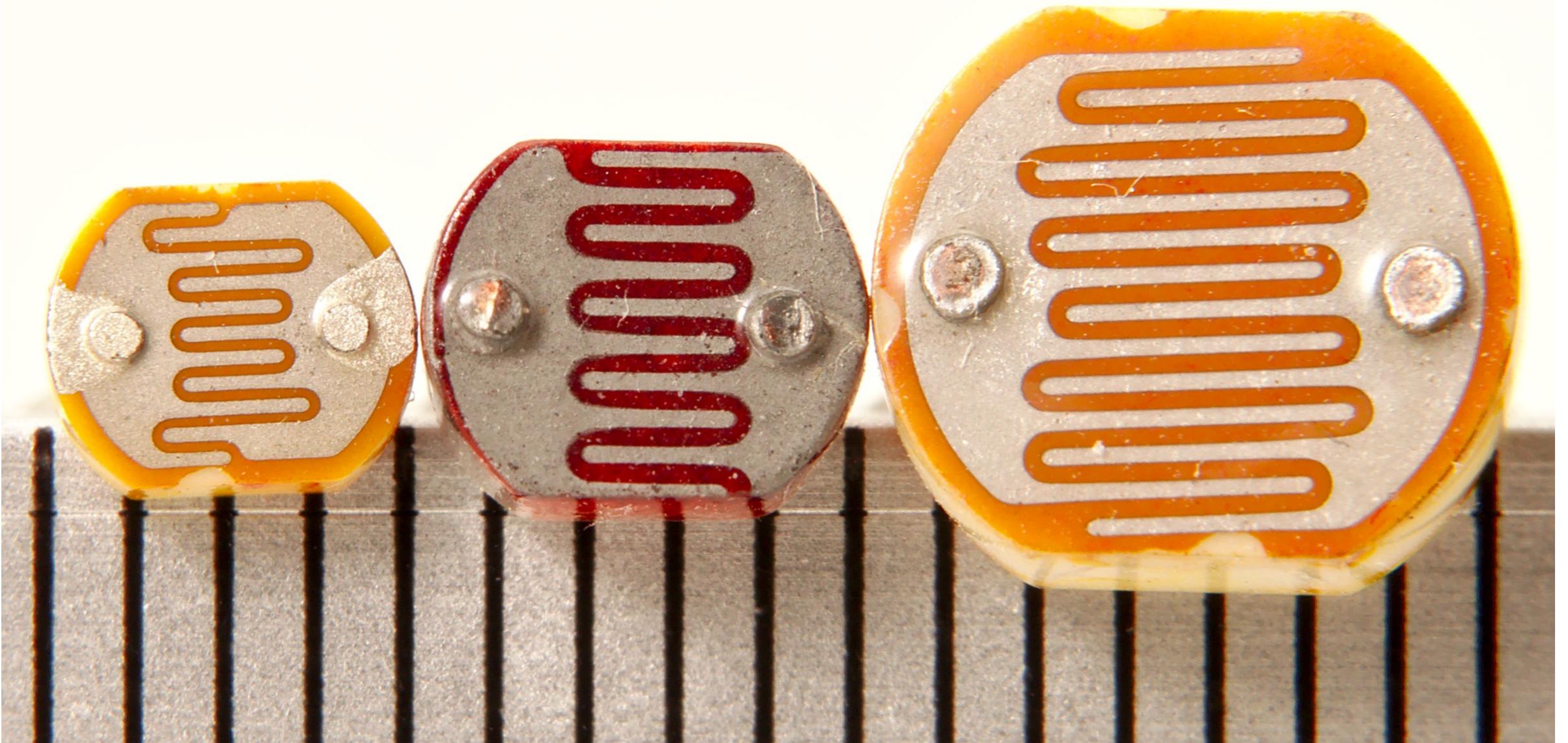
print('LED on')
GPIO.output(4,GPIO.HIGH)
time.sleep(0.5)

print('LED off')
GPIO.output(4,GPIO.LOW)
time.sleep(0.5)
```

QUIZ TIME ! _

- 1. Make the light blink 5 times**
- 2. Make the light blink at different frequency**
- 3. What are the resistance of our resistors?**

Resistor Color Code



Light-dependent Resistor

By Junkyardsparkle - Own work, CC0, <https://commons.wikimedia.org/w/index.php?curid=35977178>

QUIZ TIME ! _

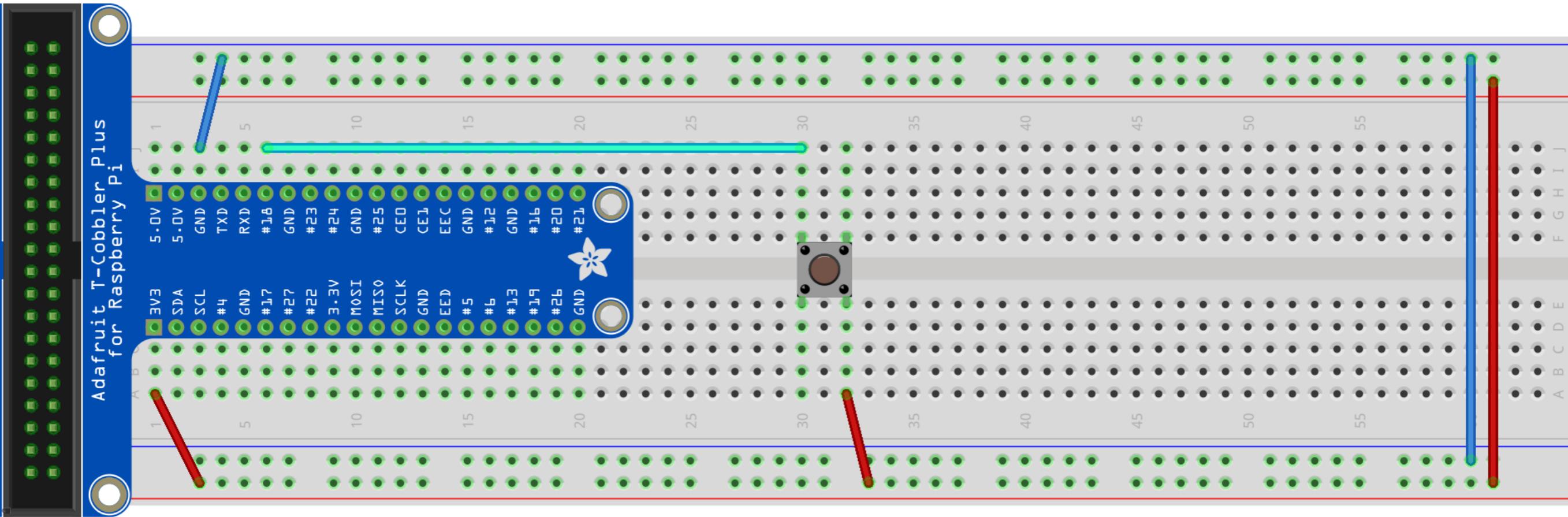
1. Replace the resistor with LDR
2. Control the LED - *LIKE MAGIC*



Switch

By oomlout - BUTT-01 (Pushbuttons), CC BY-SA 2.0, <https://commons.wikimedia.org/w/index.php?curid=23281155>

Switch - Circuit



- Plug in switch in the middle
- Wire from 3.3V (red line) to a row
- Wire from pin #18 to a different row

Switch - Code

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

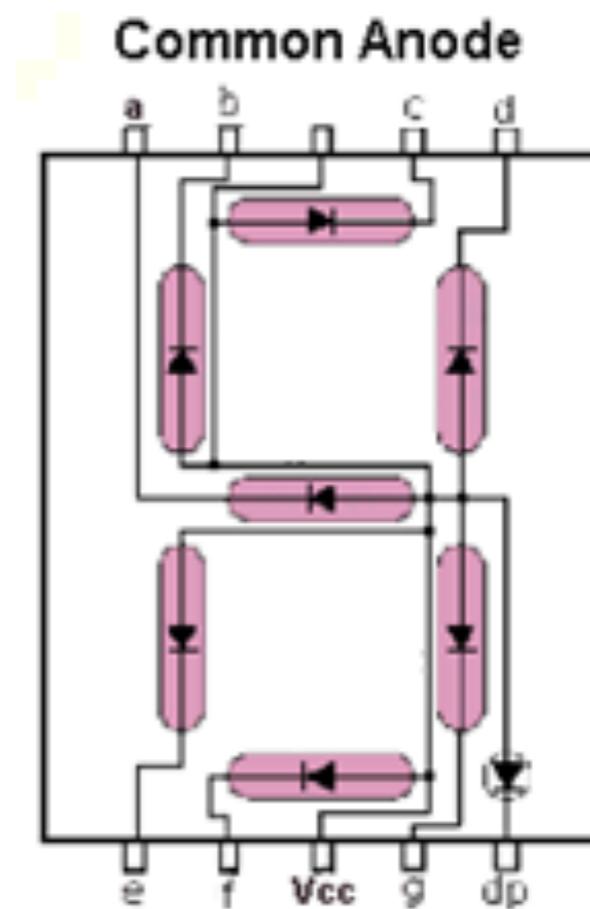
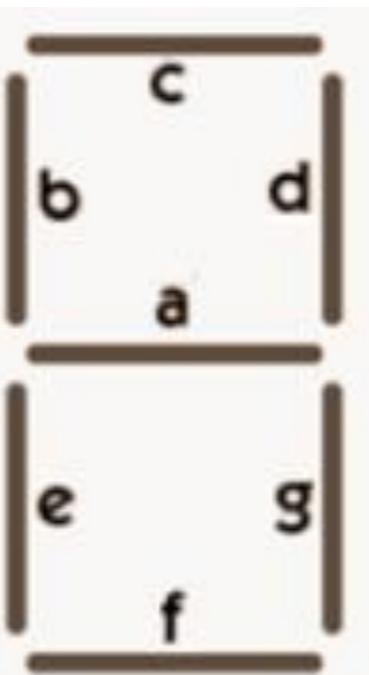
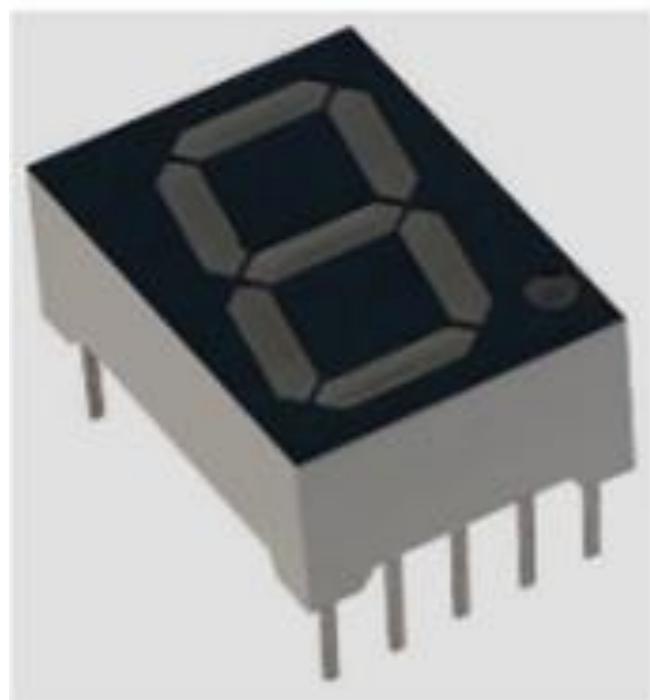
PORT_IN = 18
GPIO.setup(PORT_IN, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

while True:
    input_state = GPIO.input(PORT_IN)
    if input_state == 1:
        print('Button Pressed')
        time.sleep(0.2)
```

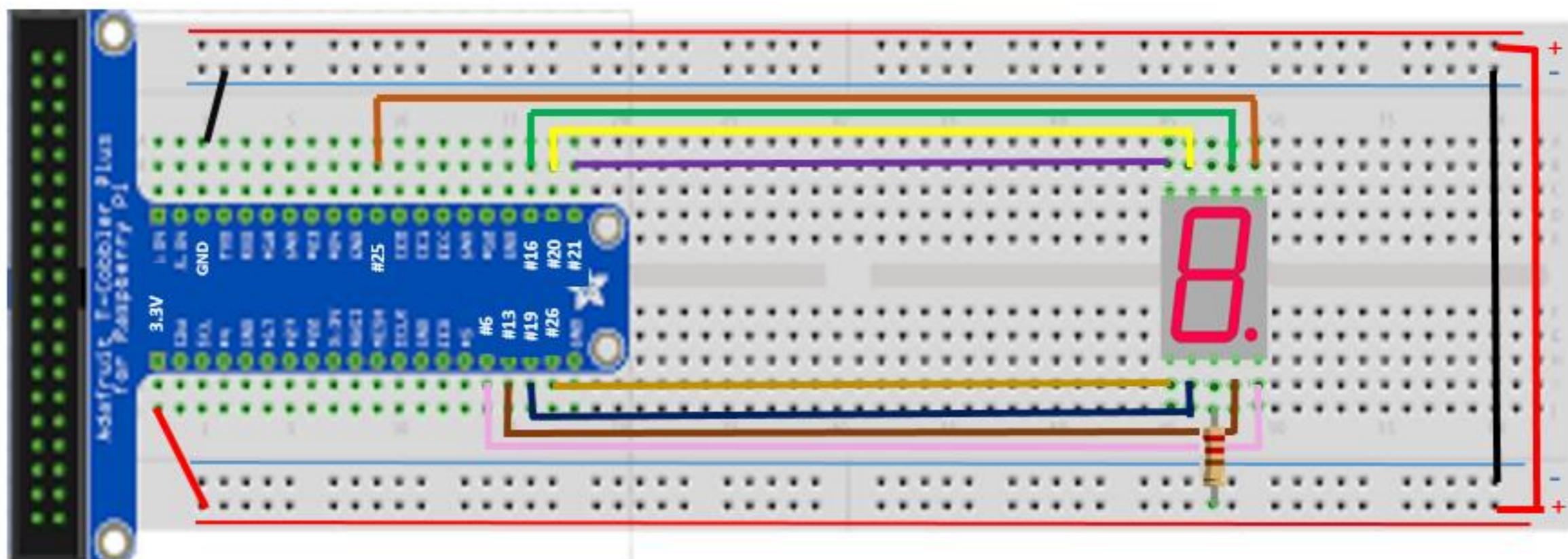
QUIZ TIME ! _

1. Count the number of times the button is pressed and print out on console
2. If we change code to: `pull_up_down=GPIO.PUD_UP`, how should we change the circuit accordingly?
3. Switch on the LED when the button is pressed

7 segment



7 Segment - Circuit



TODQ Segment - Code

Circuit Connections:

- Connect the middle bottom pin to the 3.3V + power line via a resistor
- Connect the other 8 side pins to any free pins on the raspberry pi

Python Code:

1. Fix display.py

Open the display.py file found in the CodingWorkshop folder.

Fix the values for a-h depending on which pins you connected them to.

Edit the values in the matrix num depending on the number to be displayed.

2. Code to display a digit

```
import RPi.GPIO as GPIO
import time
from display import display

GPIO.setmode(GPIO.BCM)
display(0)
```

QUIZ TIME ! _

1. Displace number increasingly/decreasingly
2. Count the number of times the button is pressed and display on 7-seg

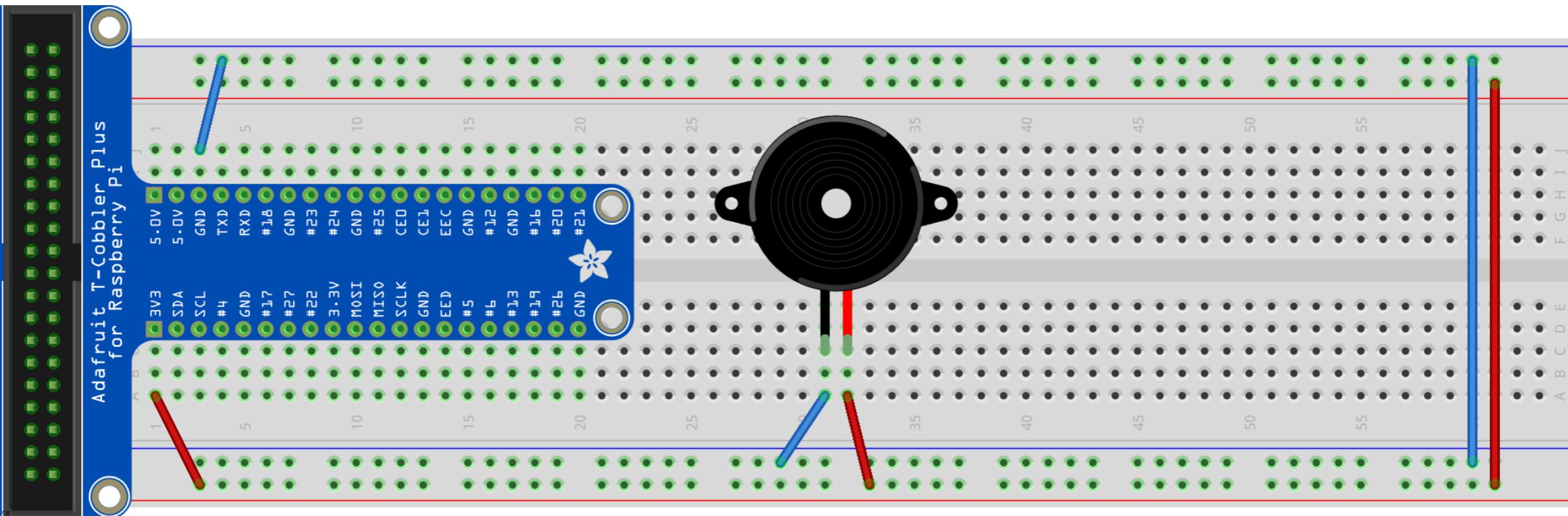
TODO

Buzzer

<https://www.youtube.com/watch?v=K8AnlUT0ng0>

https://github.com/gumslone/raspi_buzzer_player/blob/master/buzzer_player.py

Buzzer - Circuit



- Wire from 3.3V to longer leg
- Wire from ground to shorter leg

QUIZ TIME! _

1. Play a song!

Pin HIGH activates
this segment

Pins Explained

12 11 10 9 8 7



12

9

8

6

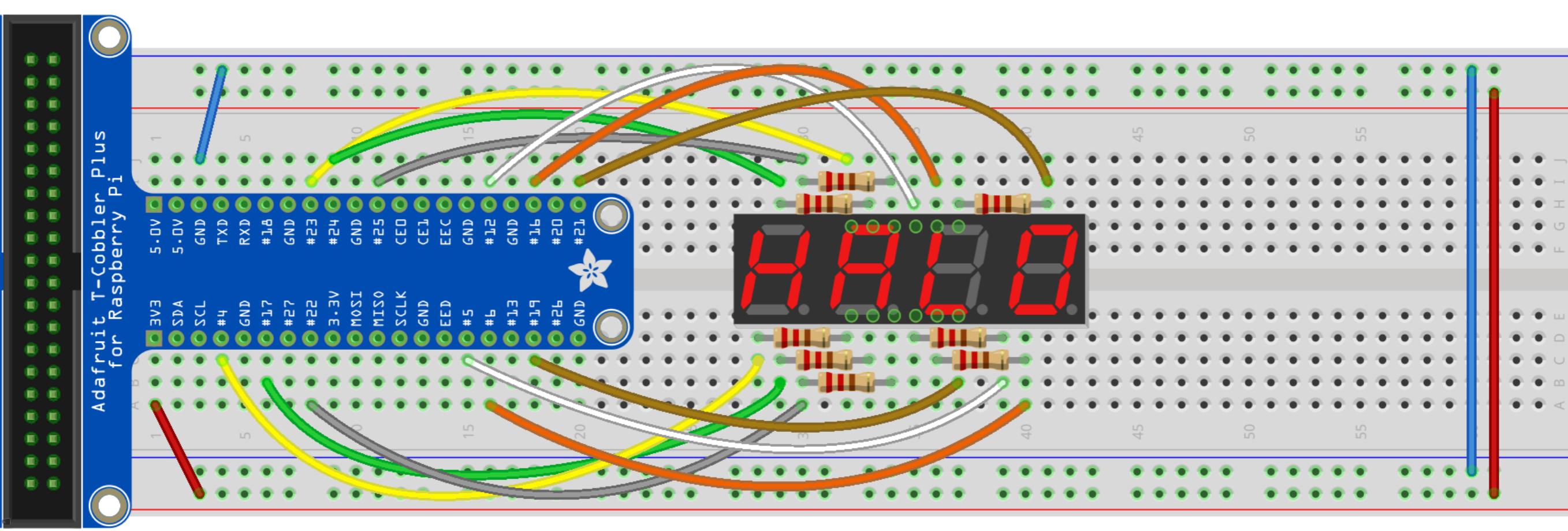
Grounding this pin, activates this digit

1 2 3 4 5 6

Common
cathode

4 Digit 7-Seg

4Digit 7Seg - Circuit



4 Digit 7 Seg - Circuit

Seg / digit	7Seg Pin	Resistor	BCM GPIO #
Bot left	1	Y	4
Bot centre	2	Y	17
Decimal pt	3	Y	22
Bot right	4	Y	5
Centre centre	5	Y	6
Digit 4	6	N	19
Top right	7	Y	21
Digit 3	8	N	16
Digit 2	9	N	12
Top left	10	Y	25
Top centre	11	Y	24
Digit 1	12	N	23

QUIZ TIME ! _

- 1.** Display 4 different numbers at the same time
- 2.** Make a clock with minutes and seconds

Project - Smart Home

- <https://lets-code-p.github.io/> => Gallery
- Photo of the project
- Short description of the project

Need help from volunteers

- GitHub account, to be added as a collaborator.
- Test and set up the remaining sets of hardware

Resources

- <https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started>
- <https://curriculum.raspberrypi.org/>
- fritzing: a software to document prototypes, and help to translate into schematics
- <http://fritzing.org/home/>
- <https://github.com/adafruit/Fritzing-Library>
- Resistor color code: <https://www.digikey.sg/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band> <https://romannurik.github.io/SlidesCodeHighlighter/>