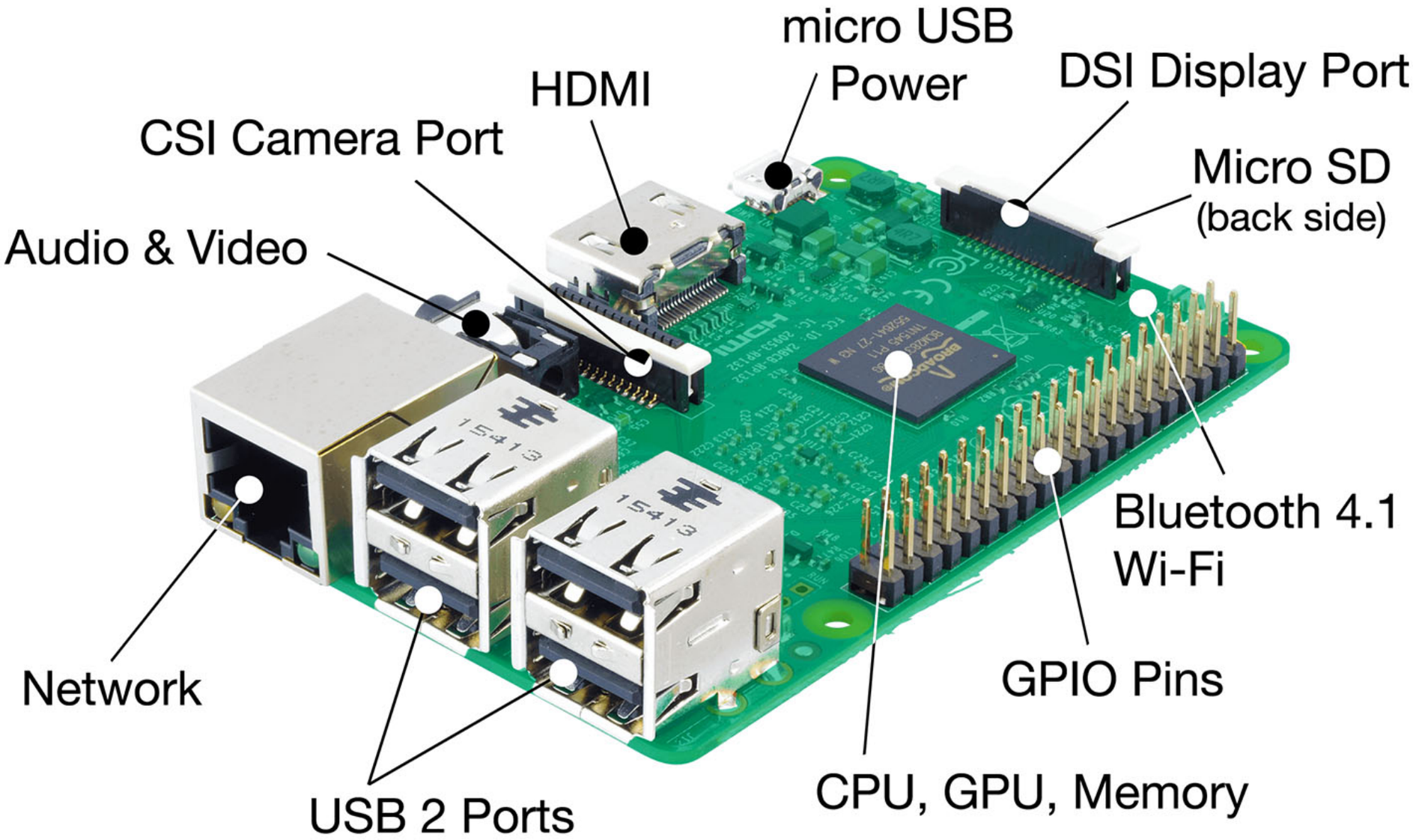
Coding Workshop

Outline

- Raspberry Pi
- Run 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?

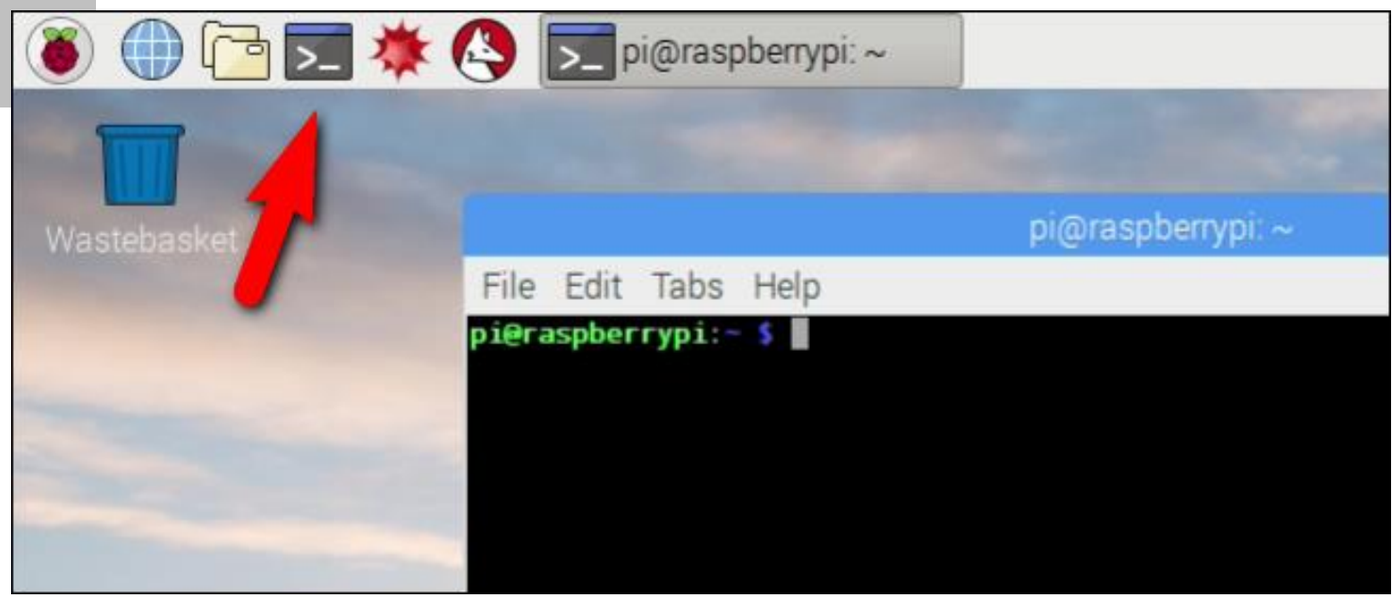
    - OS - SD card. Raspbian OS

    - I/O interface. GPIO

Python on Pi

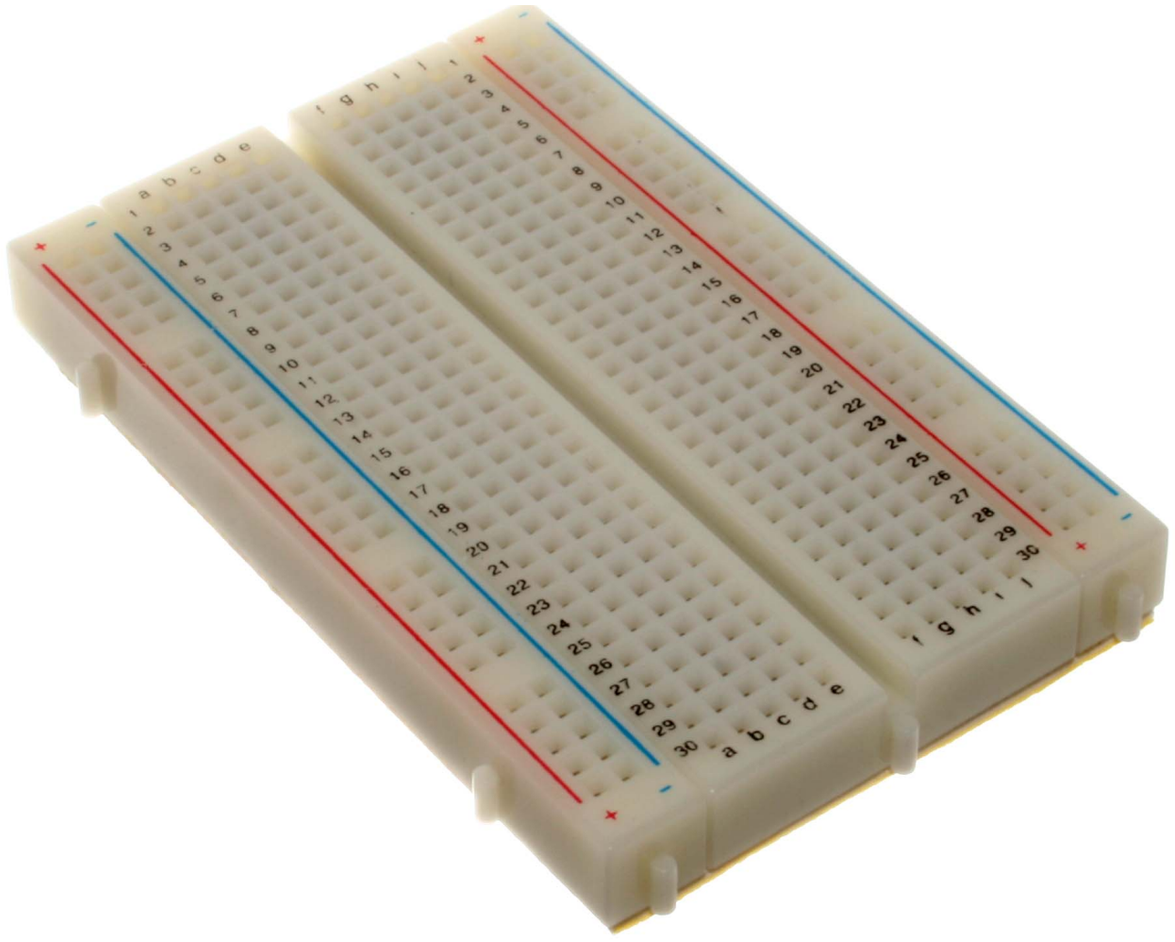
Python Programming on Pi


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



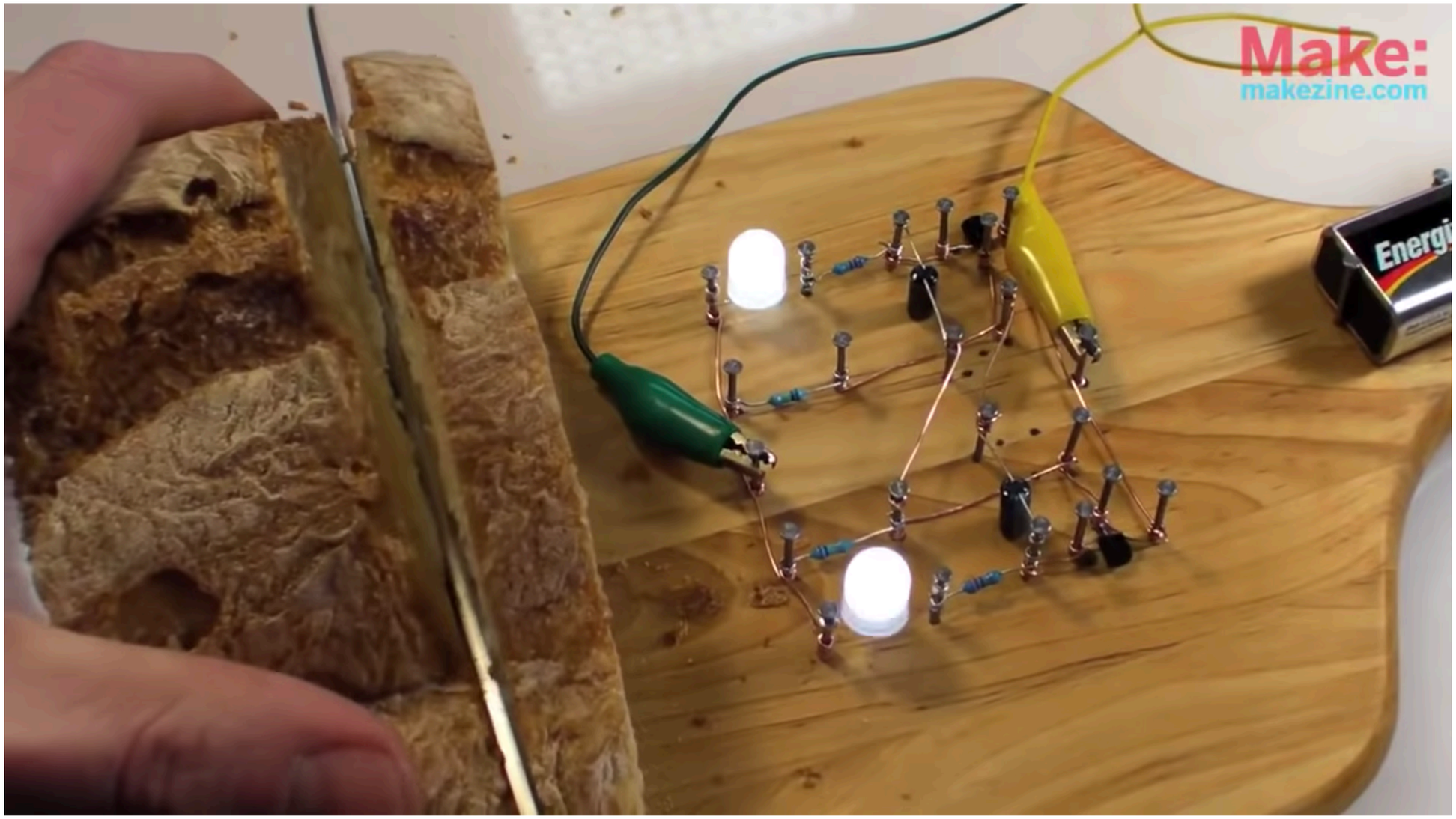
Circuit Design

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



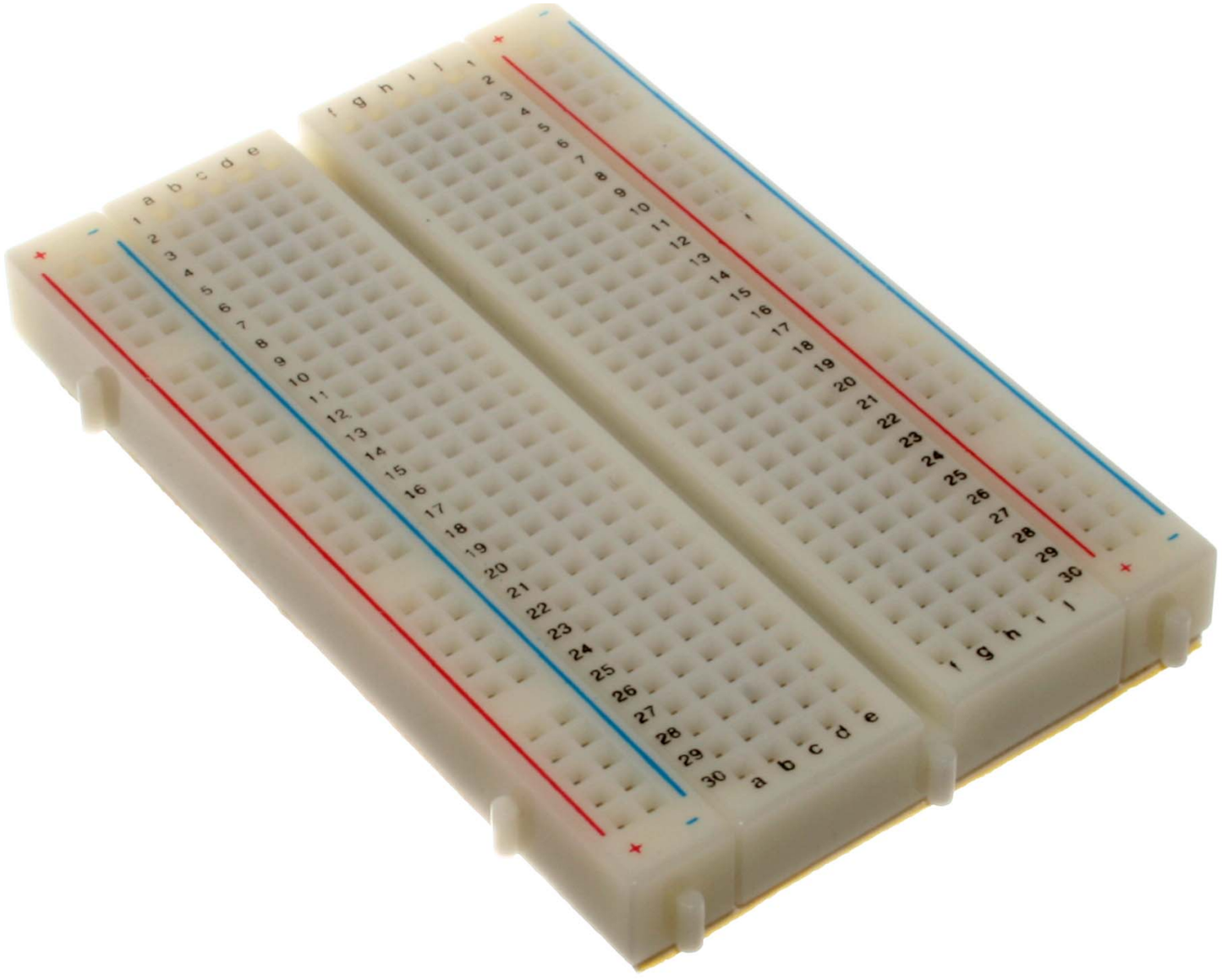
Bread Board?

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



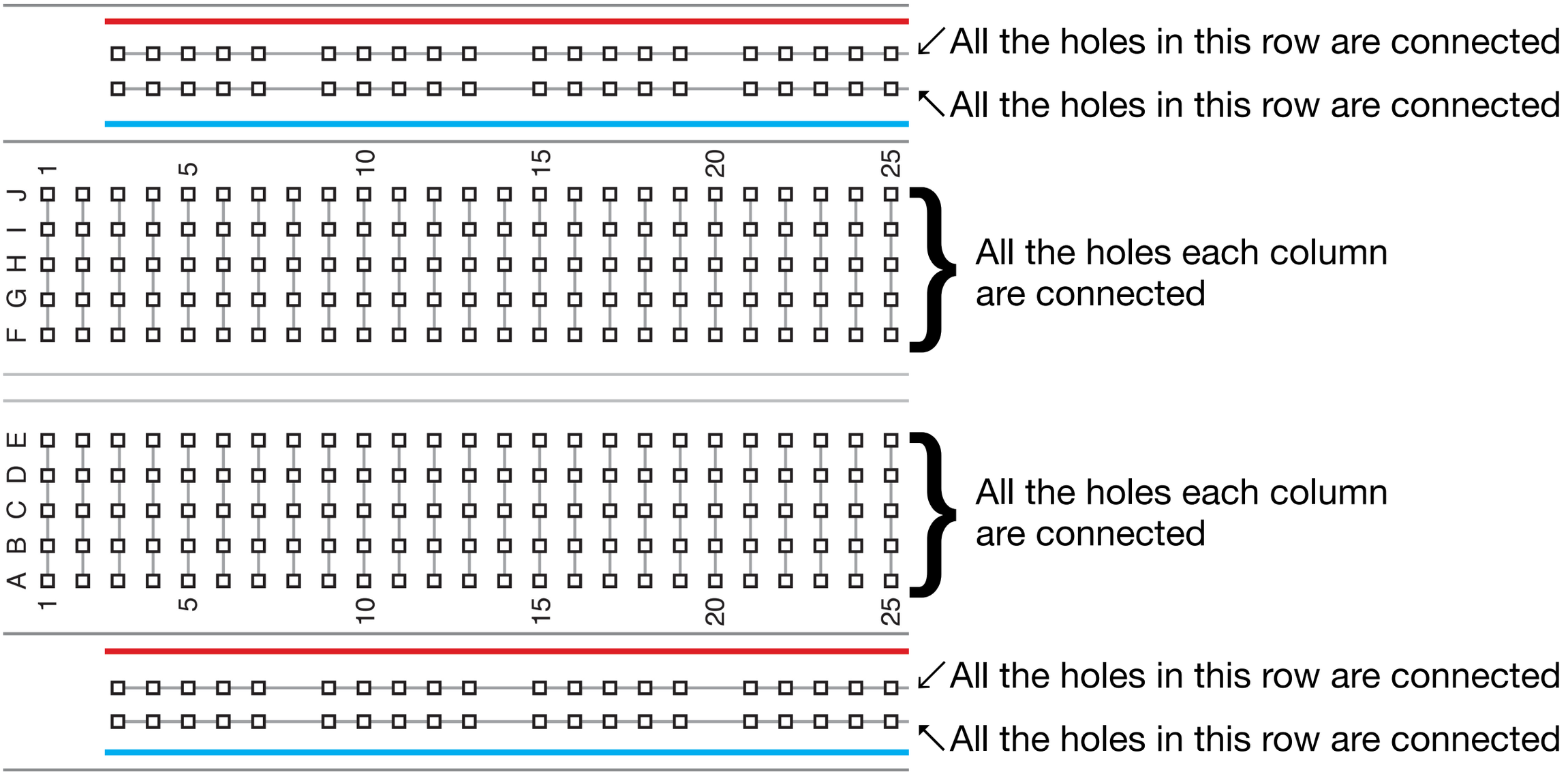
Bread Board

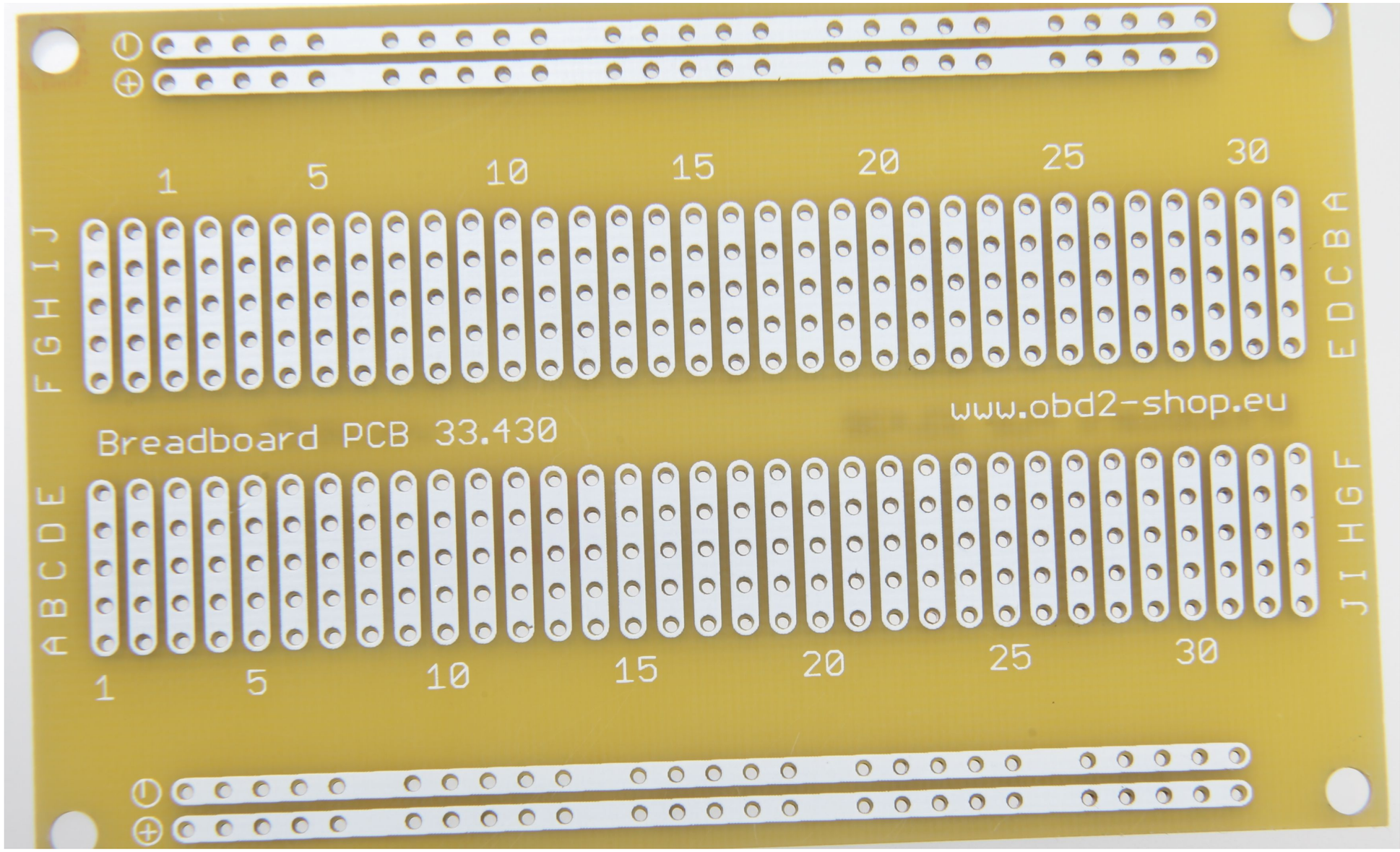
Collin's Lab: The REAL Breadboard, https://www.youtube.com/watch?v=HrG98HJ3Z6w&gl=SG&hl=en-GB



Solder-less Bread Board

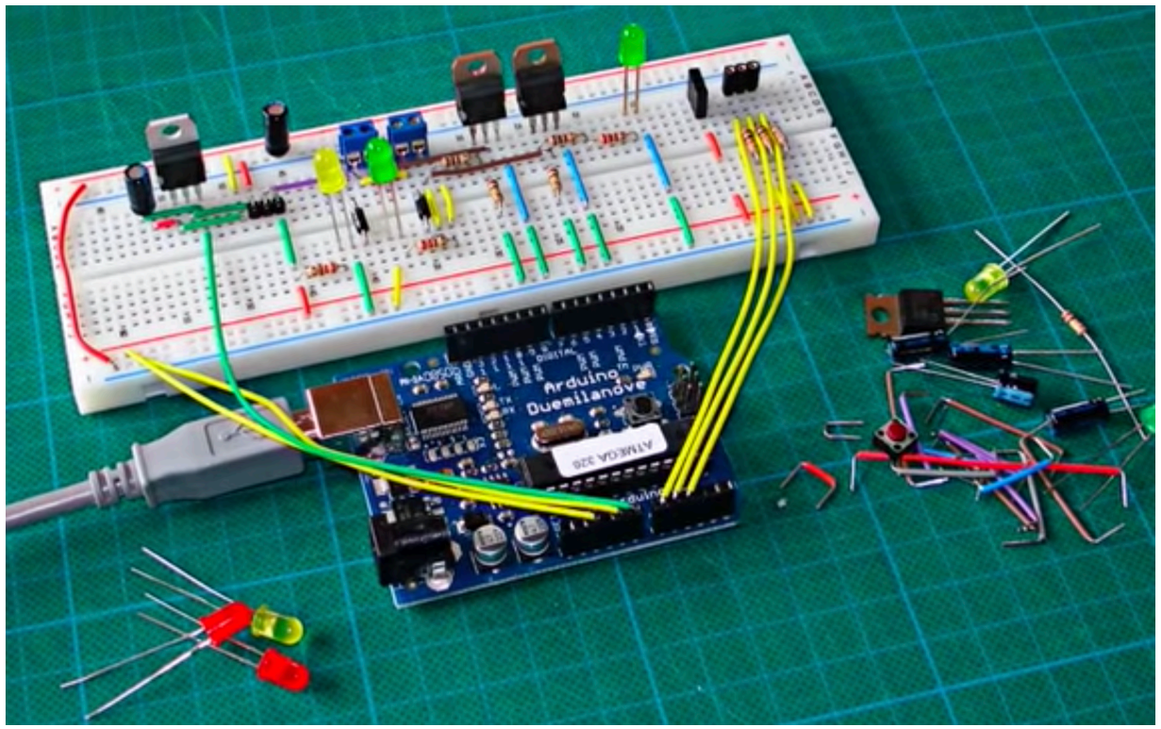
By oomlout - BREB-01 (Breadboard), CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=19867043

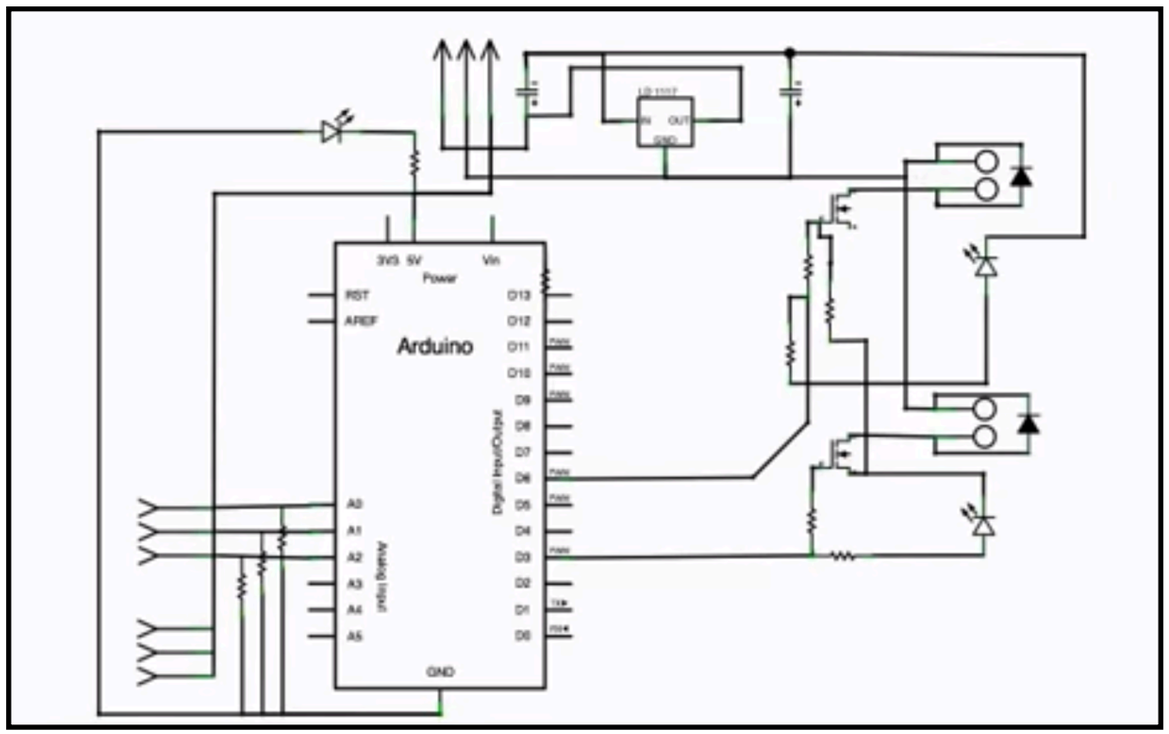

All the holes in this row are connected
All the holes in this row are connected
F G H I J
1
5
10
15
20
25
All the holes each column are connected
A B C D E
1
5
10
15
20
25
All the holes in this row are connected
All the holes in this row are connected



Printed Circuit Board

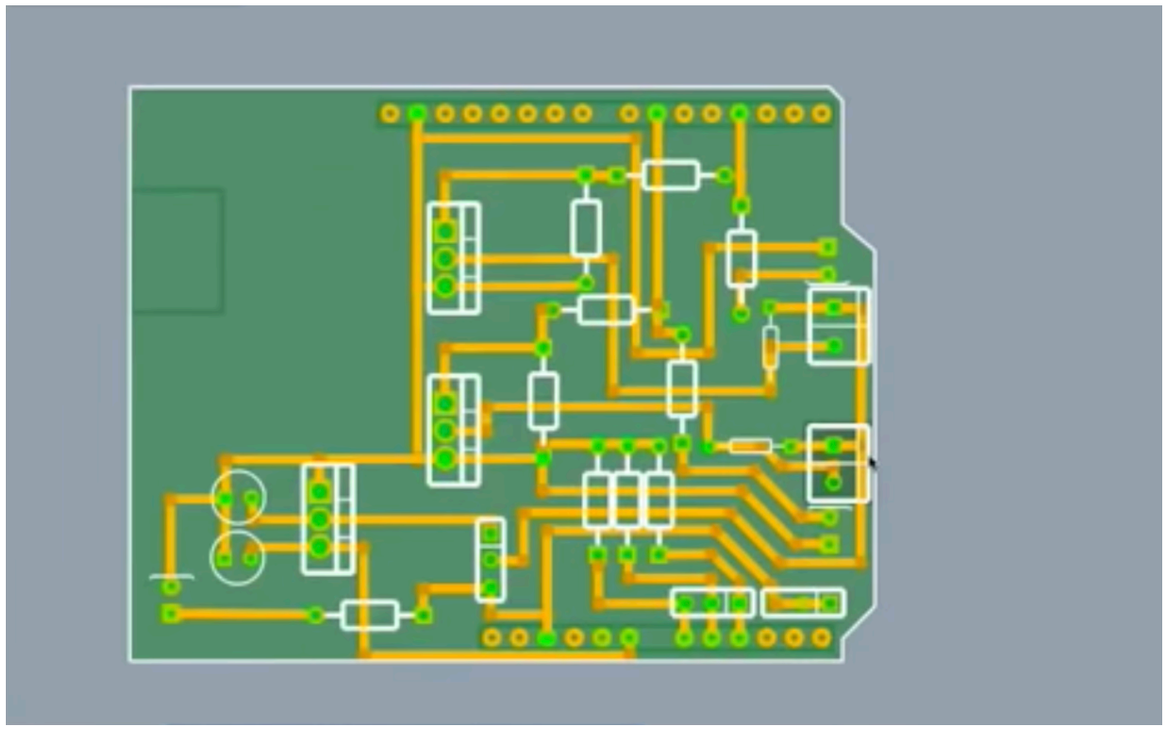
From Prototype to Commercial Electronics

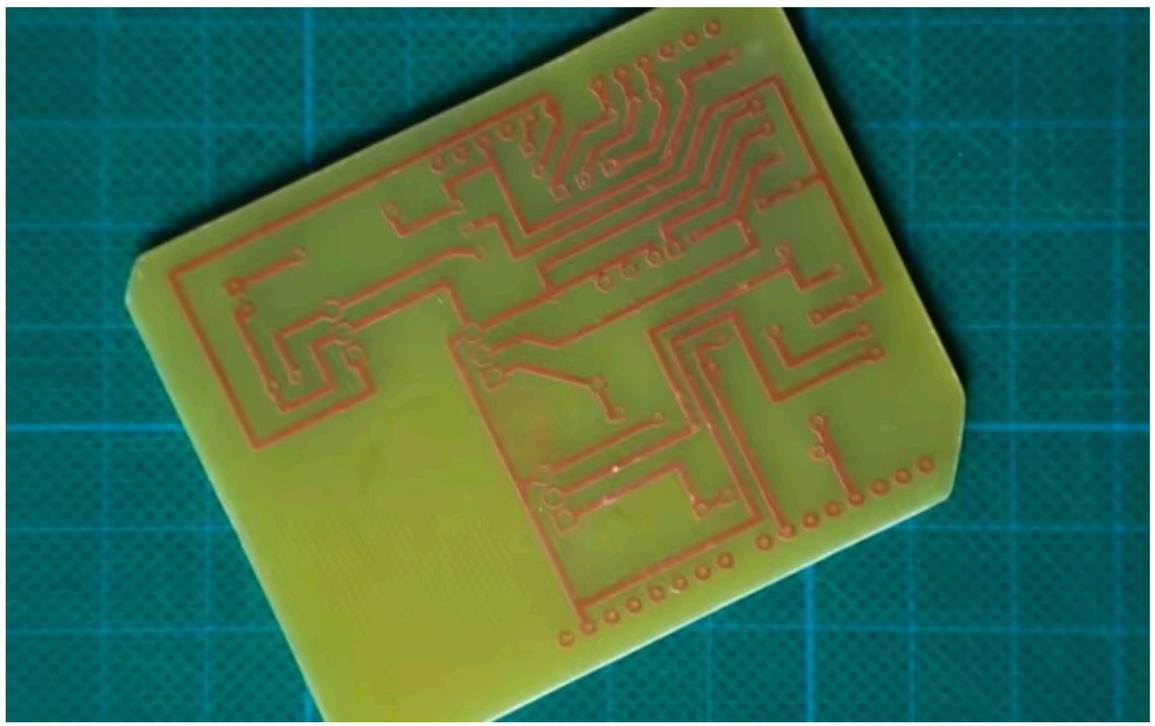




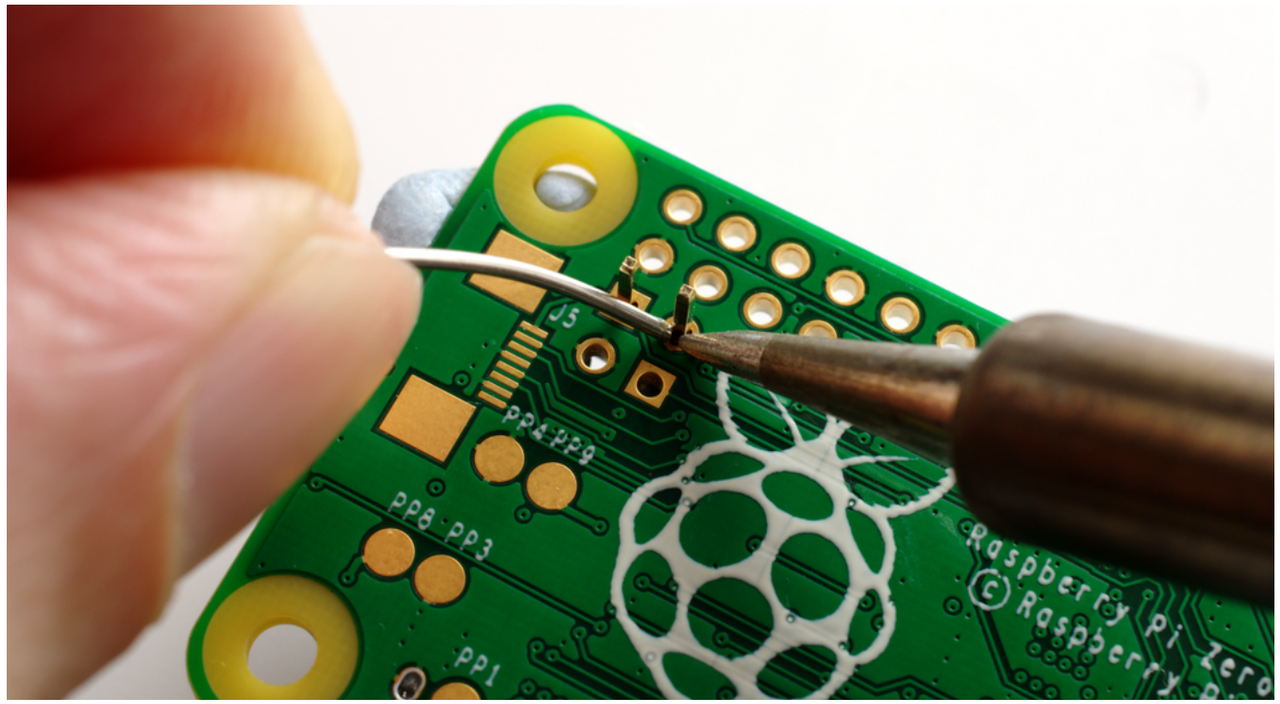
By: http://fritzing.org/home/

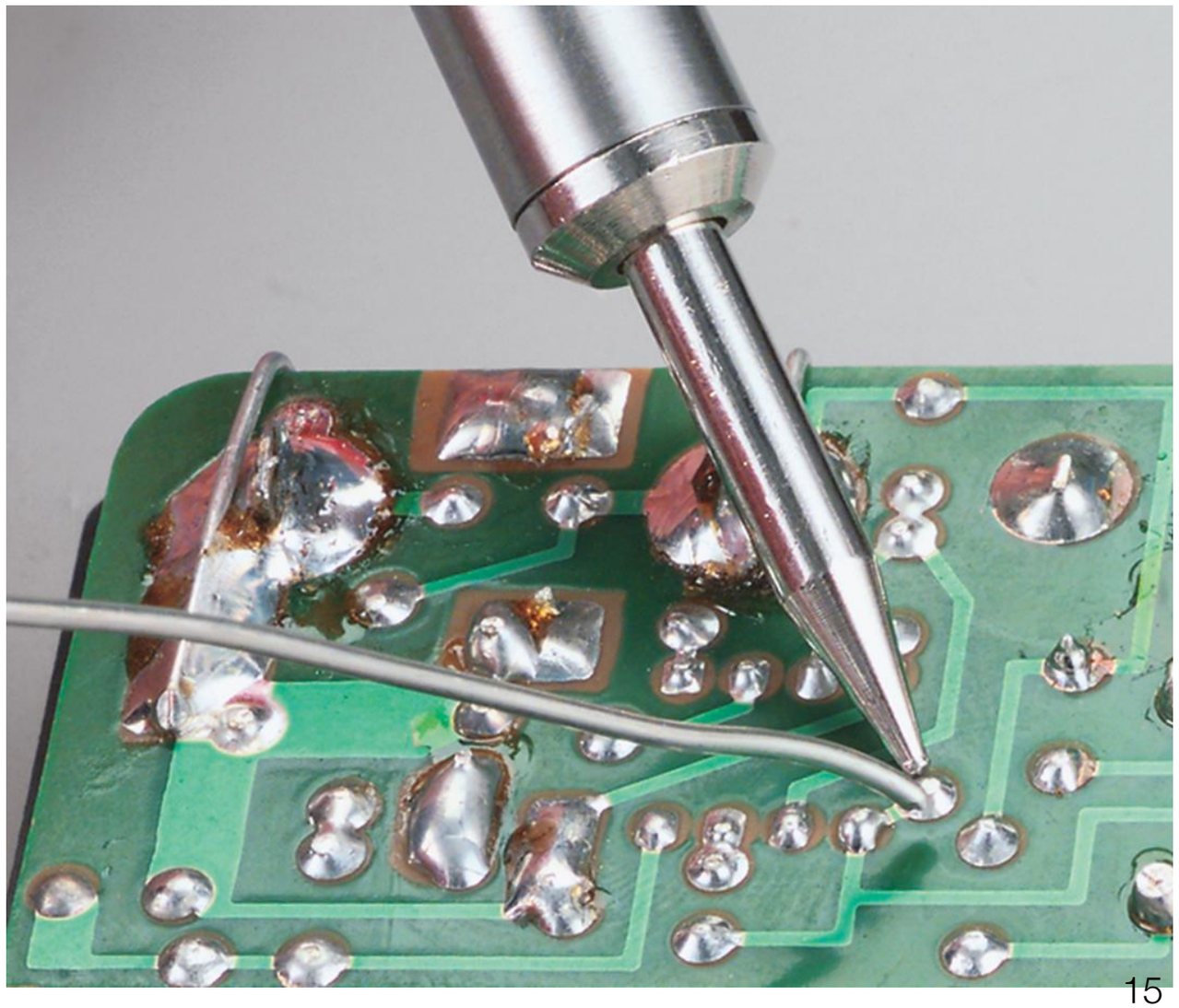
From Prototype to Commercial Electronics





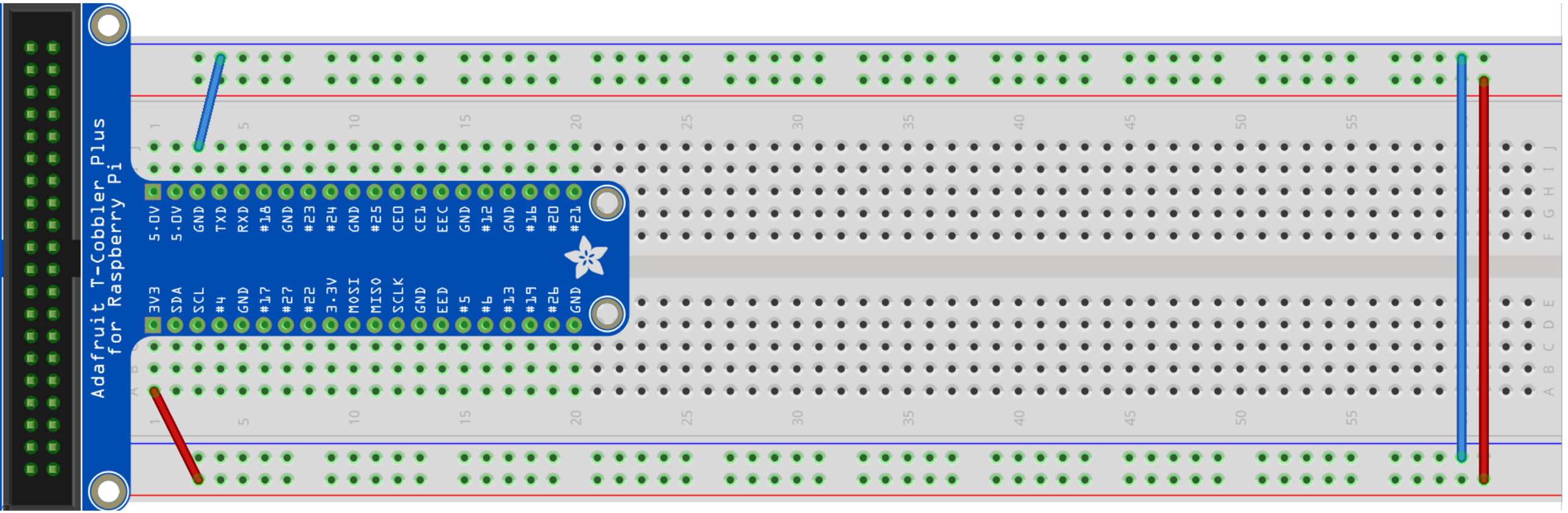
By: http://fritzing.org/home/



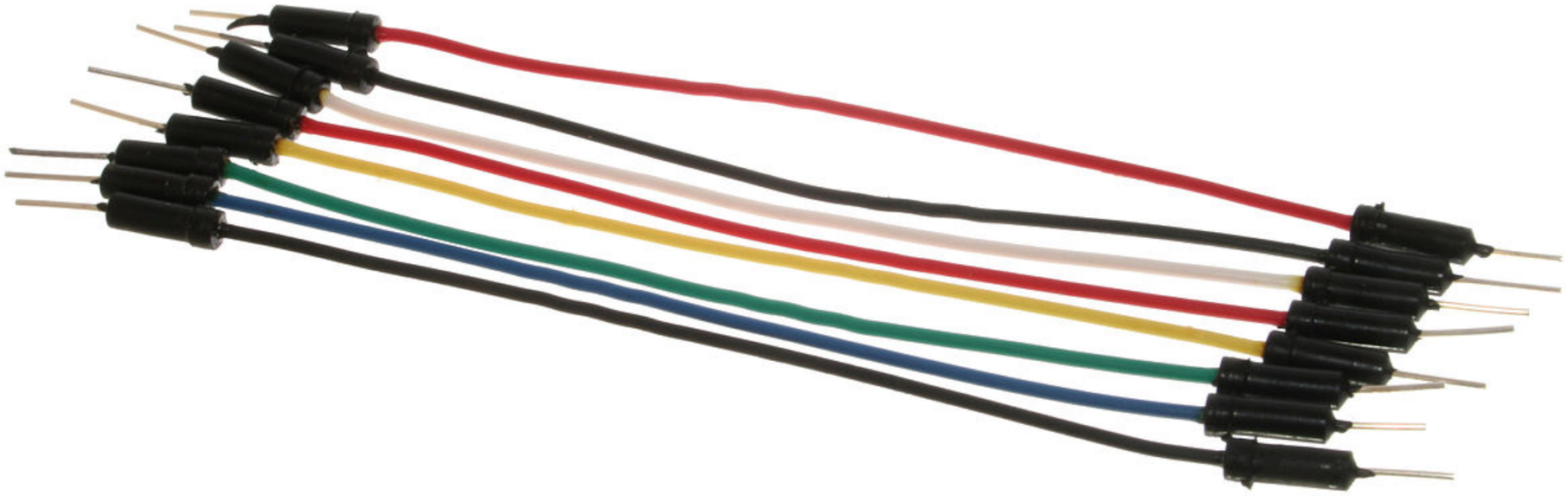


Top left: Gareth Halfacree from Bradford, UK [CC BY-SA 2.0 (https://creativecommons.org/licenses/by-sa/2.0)]
Botton left: https://m.media-amazon.com/images/S/aplus-media/vc/a40c0904-3a4c-427f-86cd-95a04a4bbf87.jpg
Right: https://pxhere.com/en/photo/1058819

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
- Extra step for transparent breadboard: Connect the middle



Jumper Wire

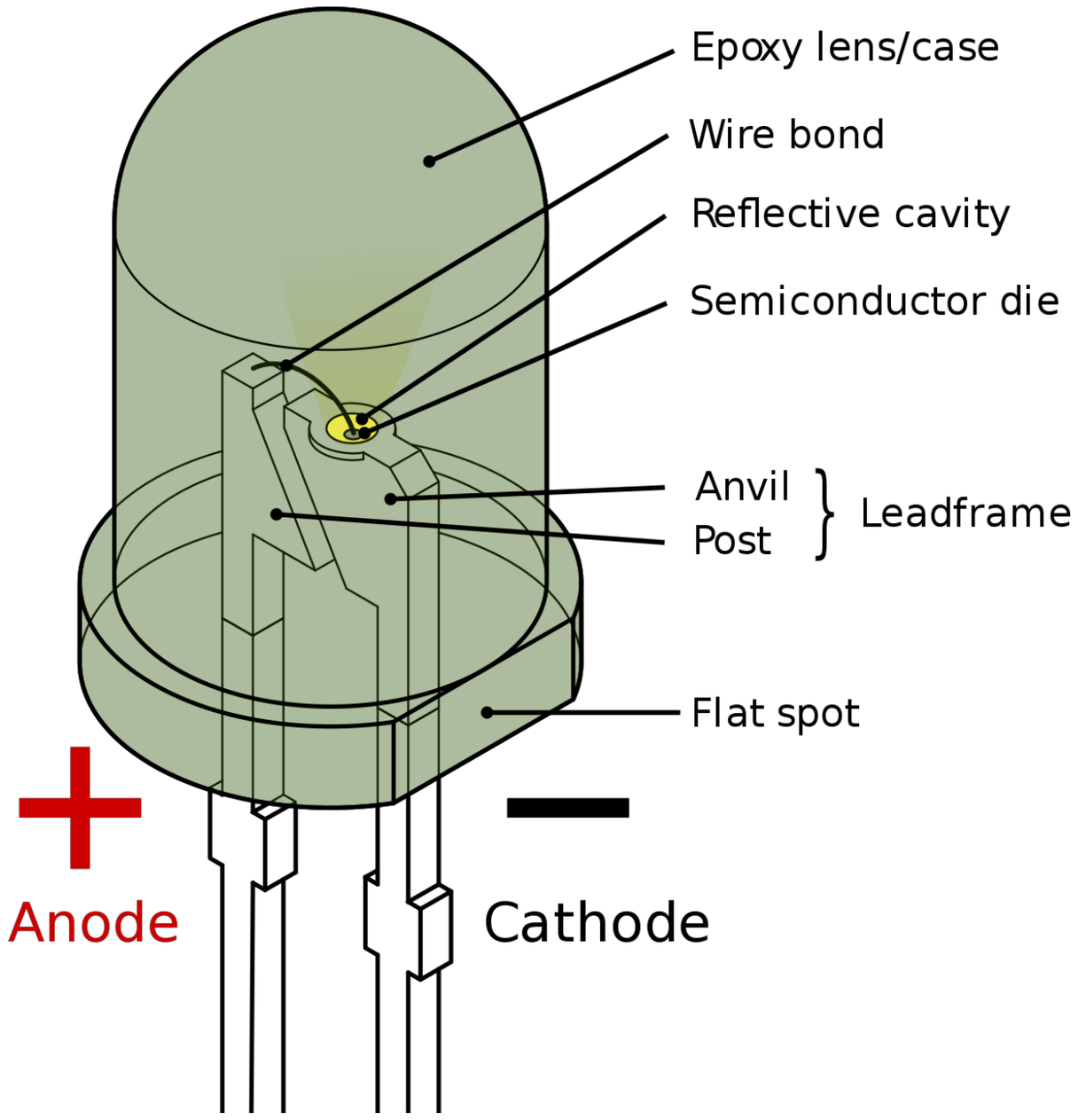
Top: By oomlout - A few Jumper Wires, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=19866418
Bottom: By oomlout - Jumper Wires with Crocodile Clips (x10) - JUMP-03Uploaded by bomazi, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=19866590

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-03Uploaded by bomazi, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=19866590

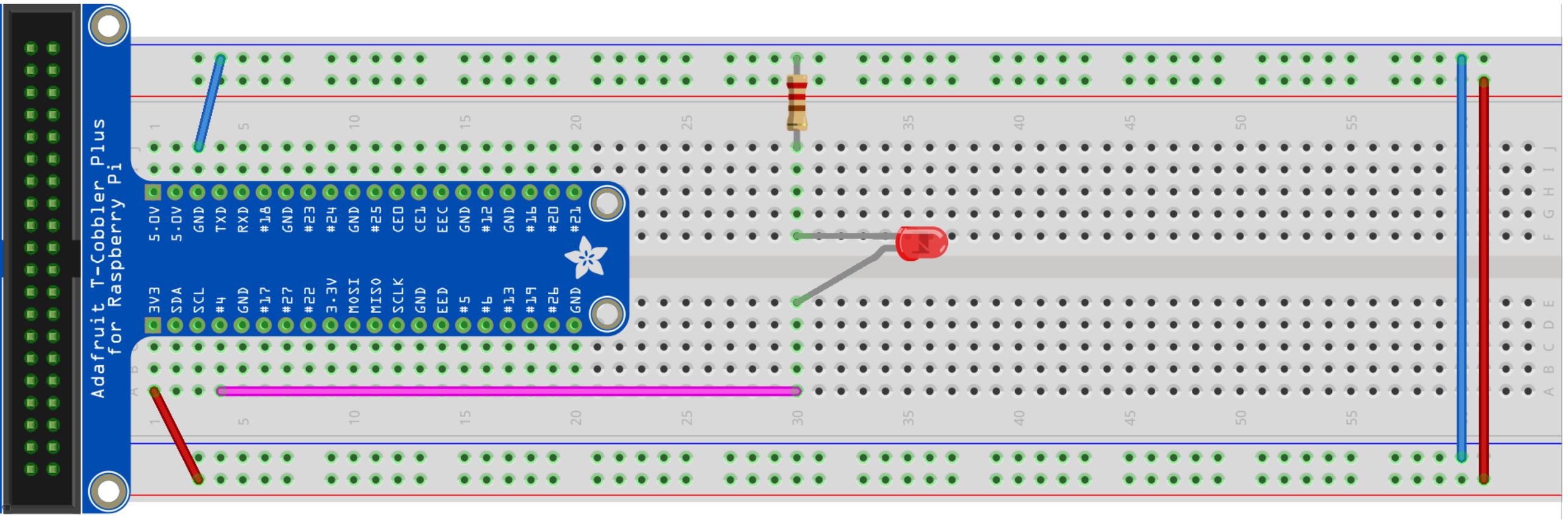
Gadgets

- 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

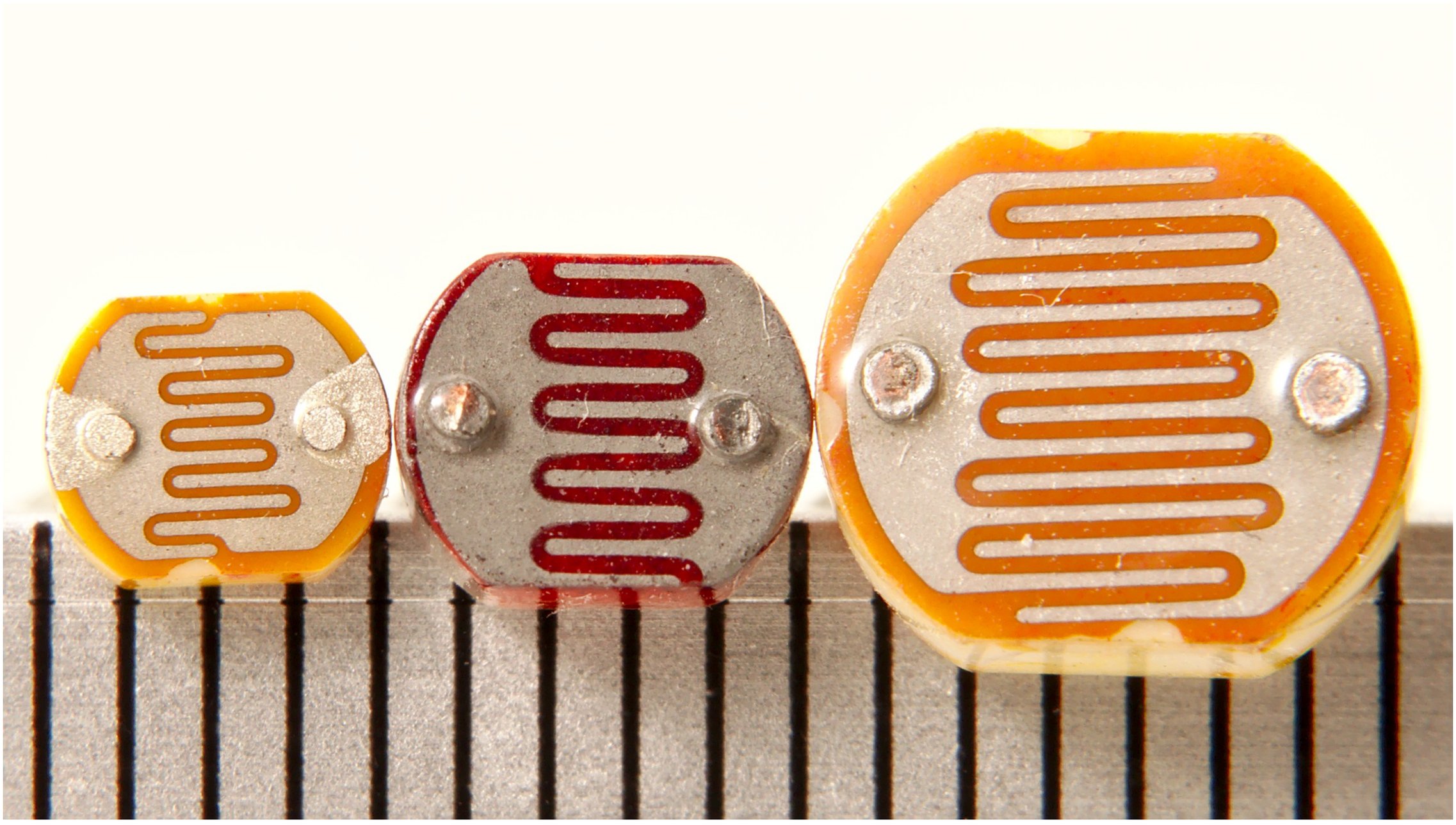
```
import RPi.GPIO as GPIO
import time

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 LED blink 5 times
2. Make the LED blink at different frequency
3. Make different LEDs blink at different frequency

4. 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

LDR - Code

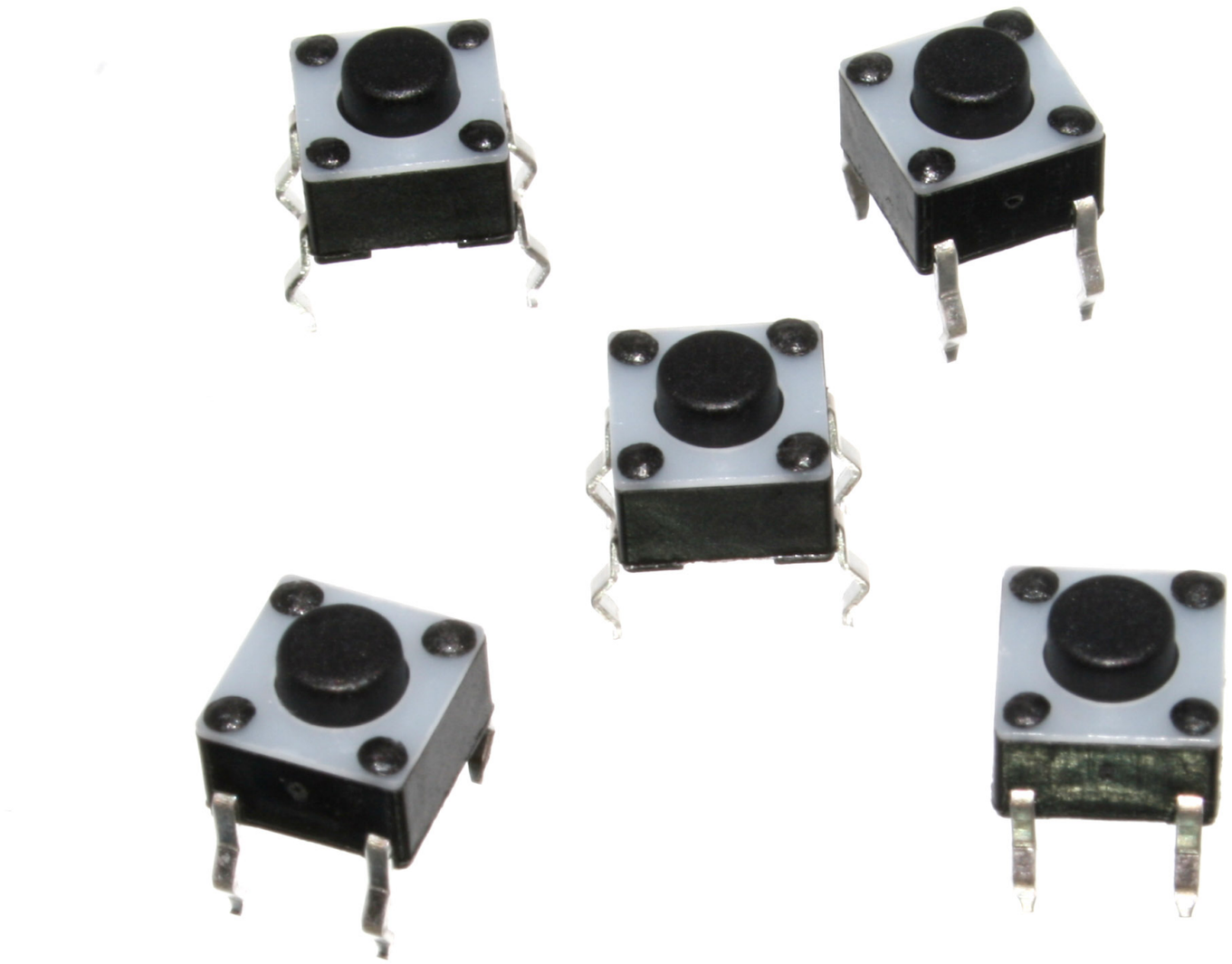
```
from gpiozero import LightSensor
ldr = LightSensor(4)
while True:
    print(ldr.value)
```

LDR - hard version

```
import RPi.GPIO as GPIO
import time

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

def RCtime (RCpin):
    reading = 0
    GPIO.setup(RCpin, GPIO.OUT)
    GPIO.output(RCpin, GPIO.LOW)
    time.sleep(0.1)
    GPIO.setup(RCpin, GPIO.IN)
    # This takes about 1 millisecond per loop cycle
    while (GPIO.input(RCpin) == GPIO.LOW):
        reading += 1
        return reading

while True:
    print RCtime(4) # Read RC timing using pin #4
```

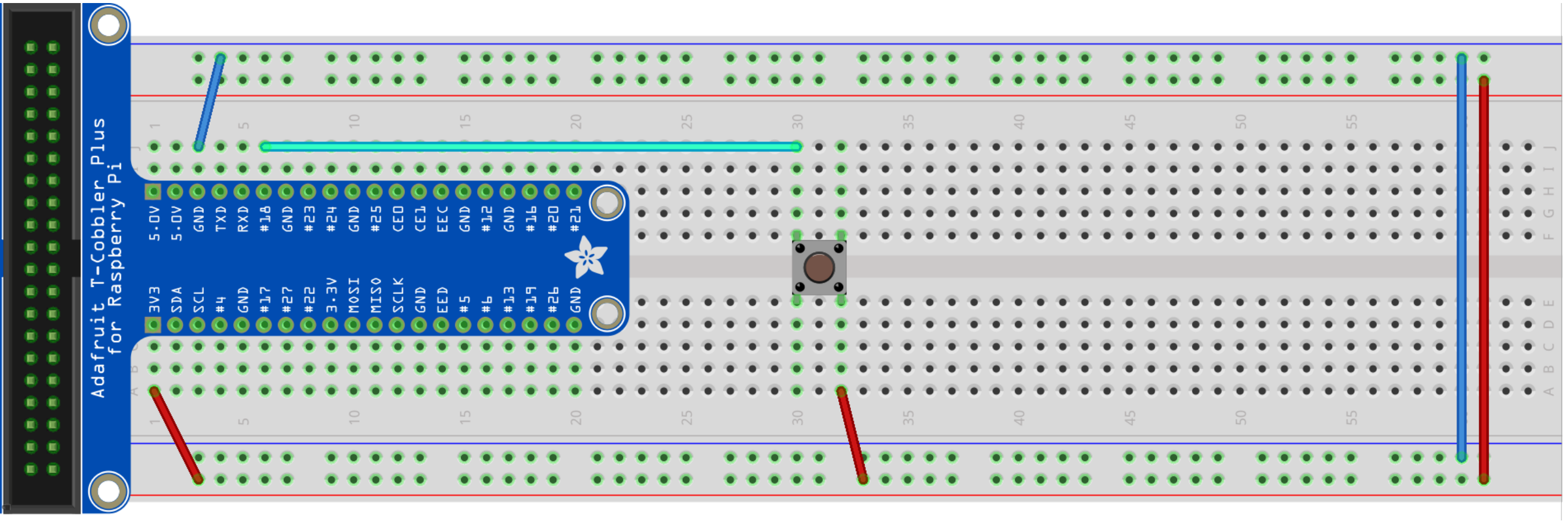
QUIZ TIME!_

1. Use the reading to control LED
2. Use the reading to control different LEDs



Switch

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

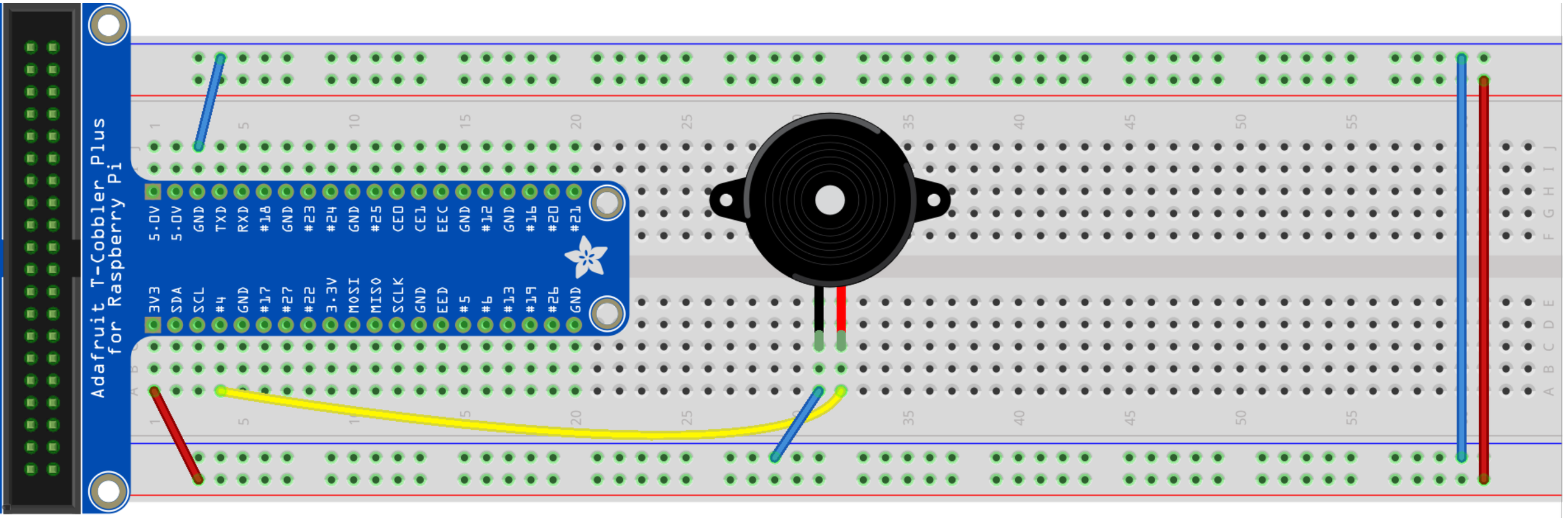
```
import RPi.GPIO as GPIO
import time

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

Buzzer

Buzzer - Circuit

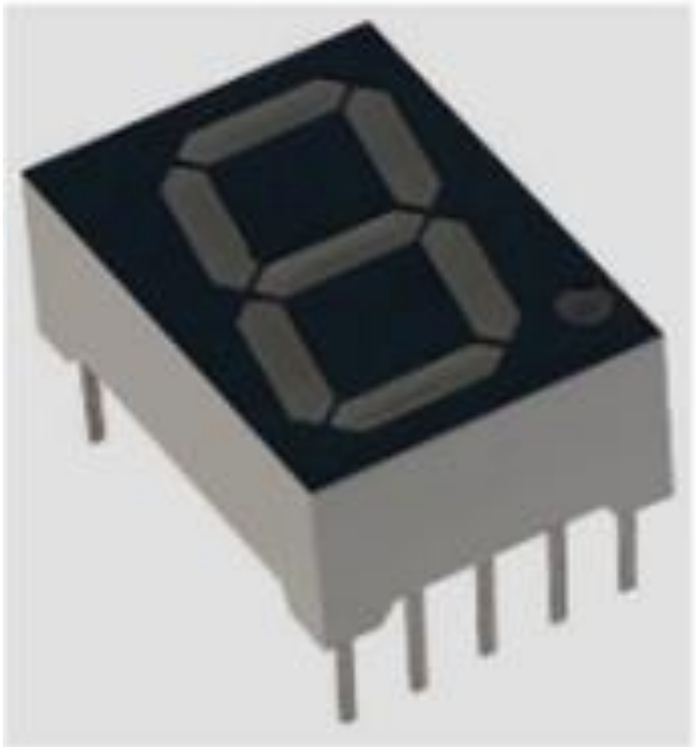


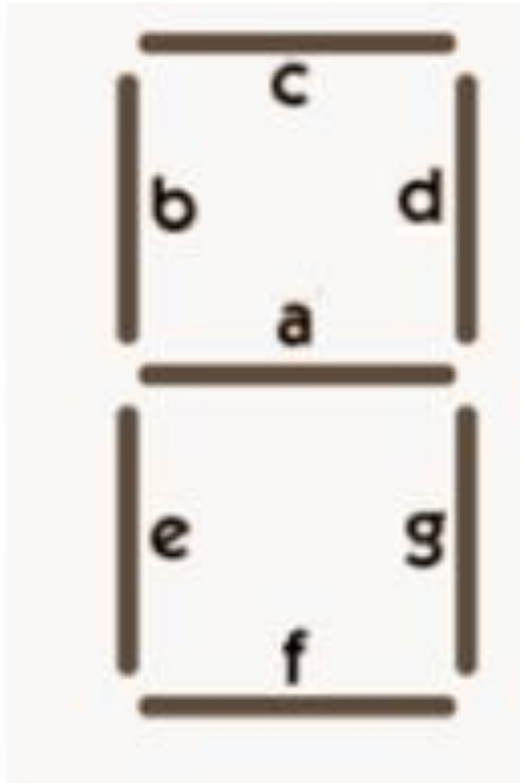
- Wire from #4 to longer leg
- Wire from ground to shorter leg

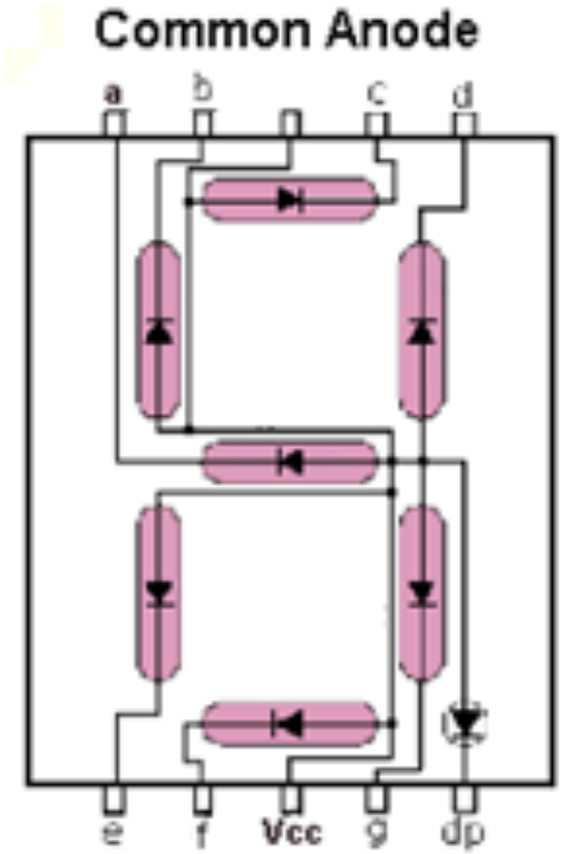
QUIZ TIME!_

1. Make some noise

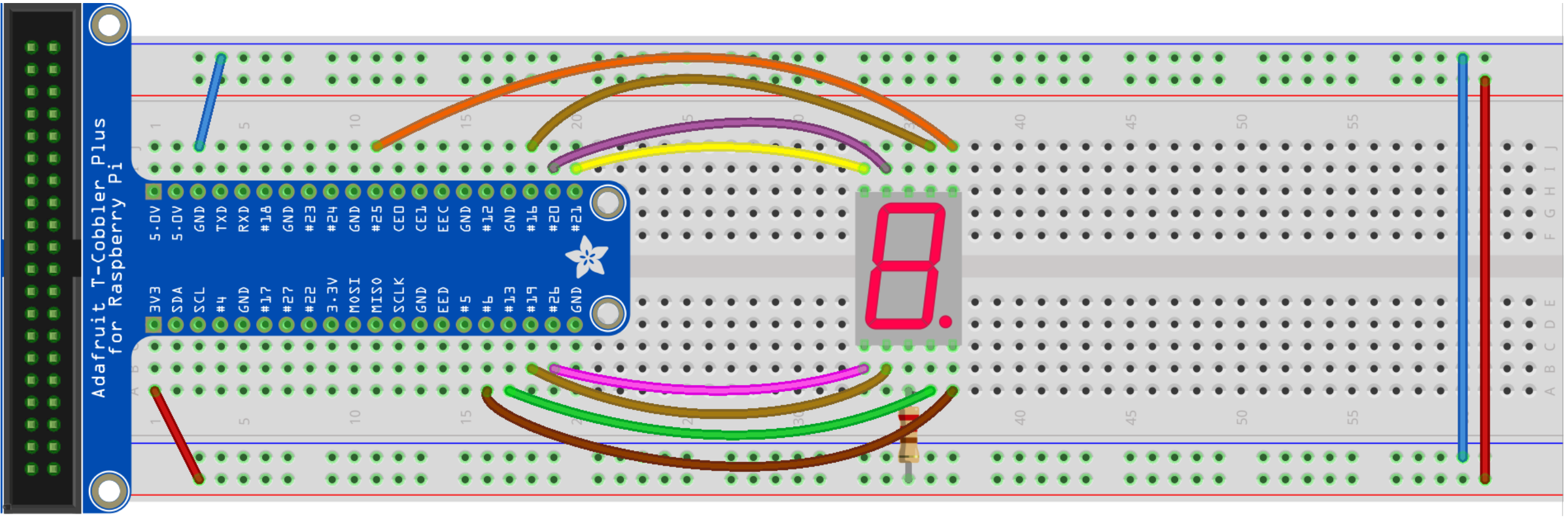
7 segment







7 Segment - Circuit



- 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

7 Segment - Code

1. Fix sevenSeg.py

    - Open the display.py file 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)
GPIO.setwarnings(False)

GPIO.cleanup()
display(0)
```

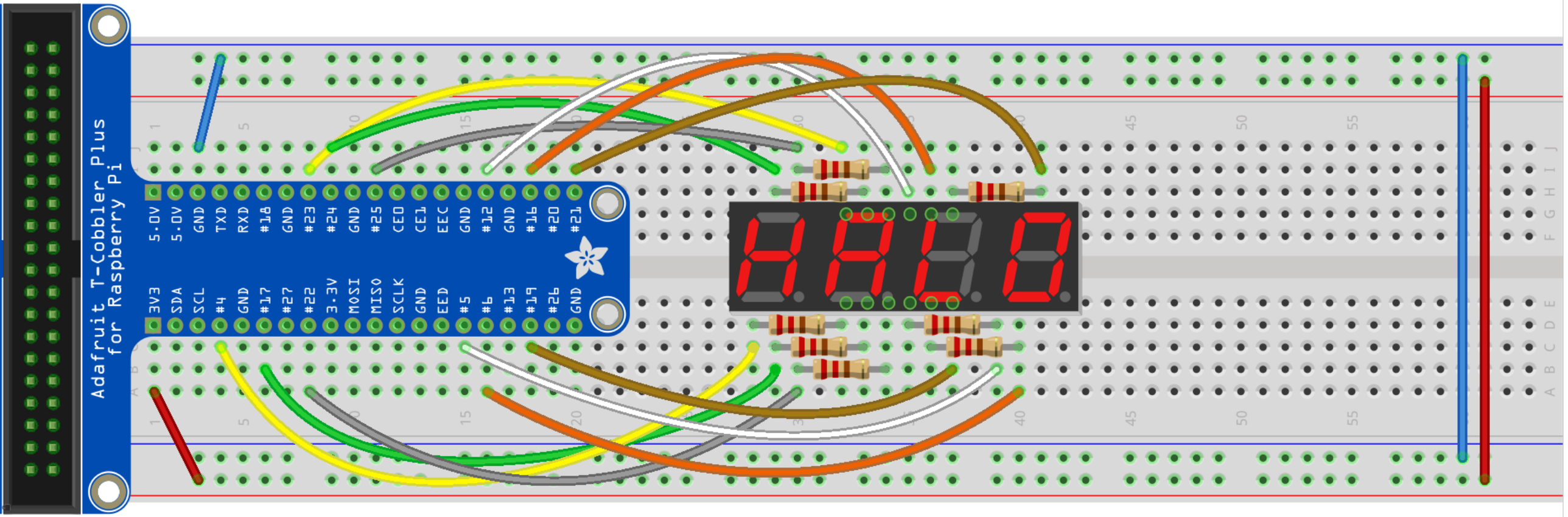
QUIZ TIME!_

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

4 Digit 7-Seg

https://raspi.tv/2015/how-to-drive-a-7-segment-display-directly-on-raspberry-pi-in-python

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

Example

- Example system design:
- Morning alarm system: alarm rings at 8AM, push alarm button to stop the alarm, and then light turns on automatically
- Buzzer rings when 7seg goes to 8, push button to stop the alarm, and the LED lights up automatically

Project

- https://lets-code-p.github.io/ => Gallery

- Photo of the project

- Short description of the project

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/