

# Set Up

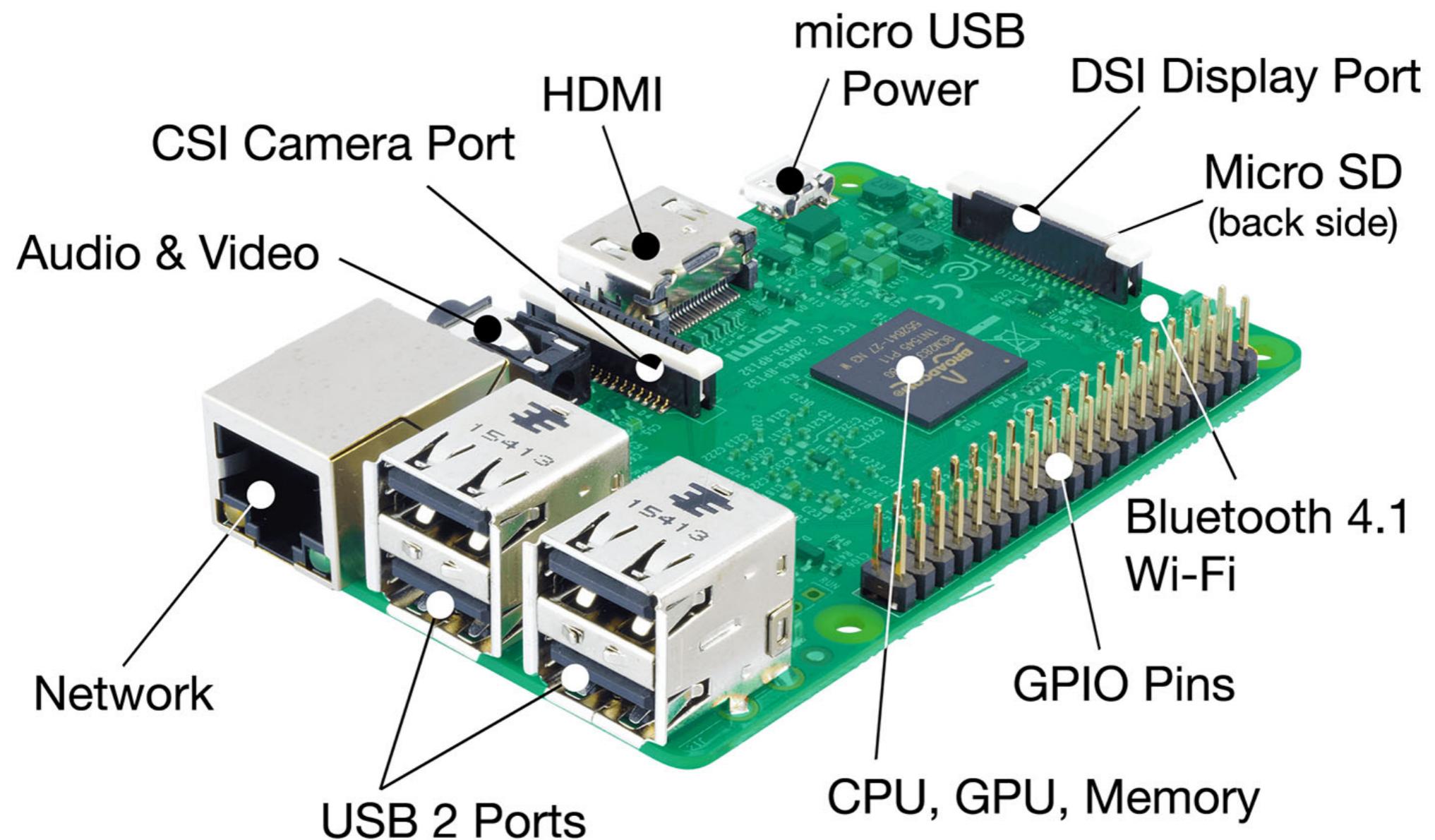
- HDMI Cable to HDMI slot
- Keyboard to one USB slot
- Mouse to another USB slot
- Power to micro USB Power Input

# Coding Workshop

# Outline

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

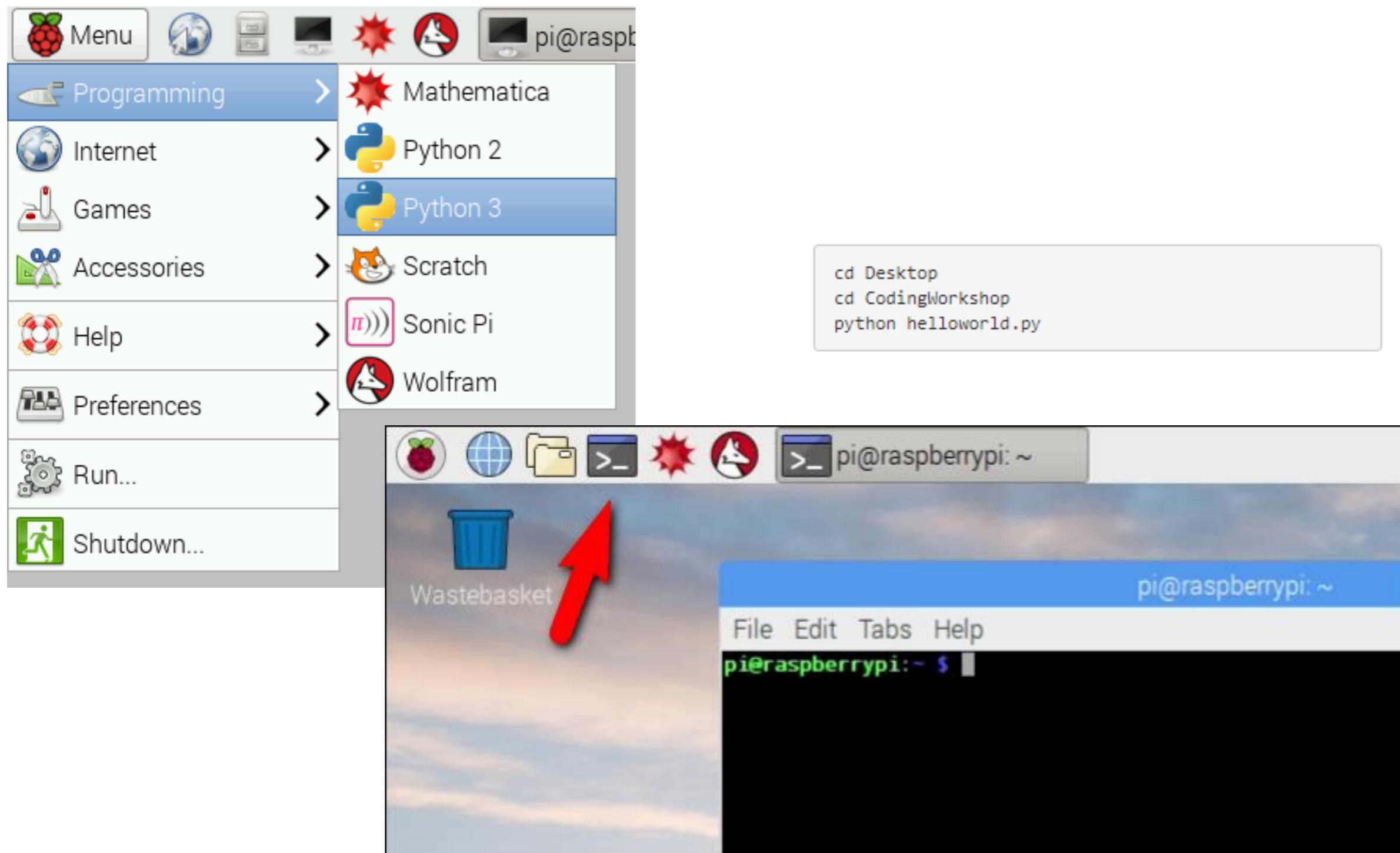
# What is Raspberry Pi?



1. “The Raspberry Pi is a **credit-card sized compute..**”
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. GIPO

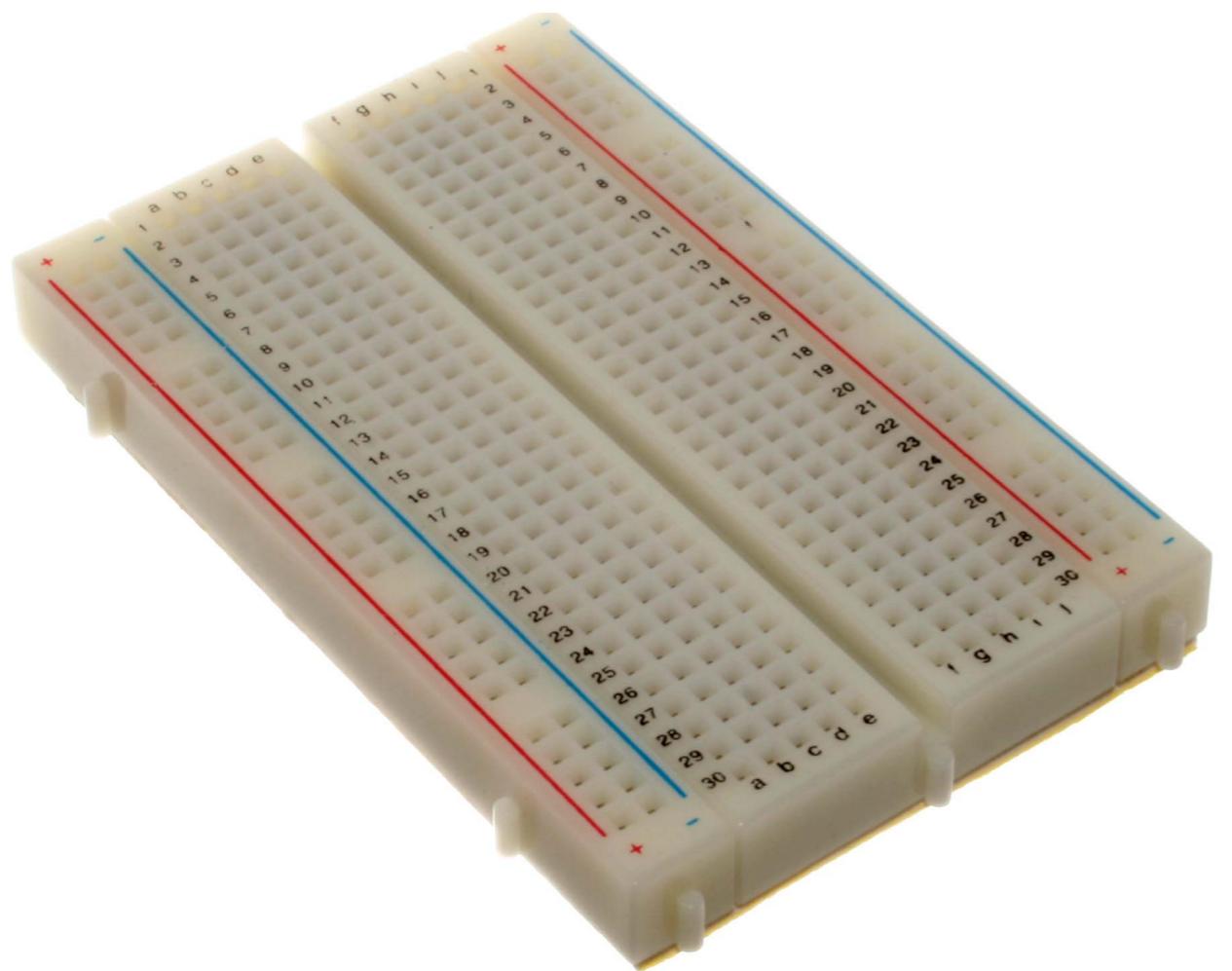
# **Python on Pi**

# Python Programming on Pi



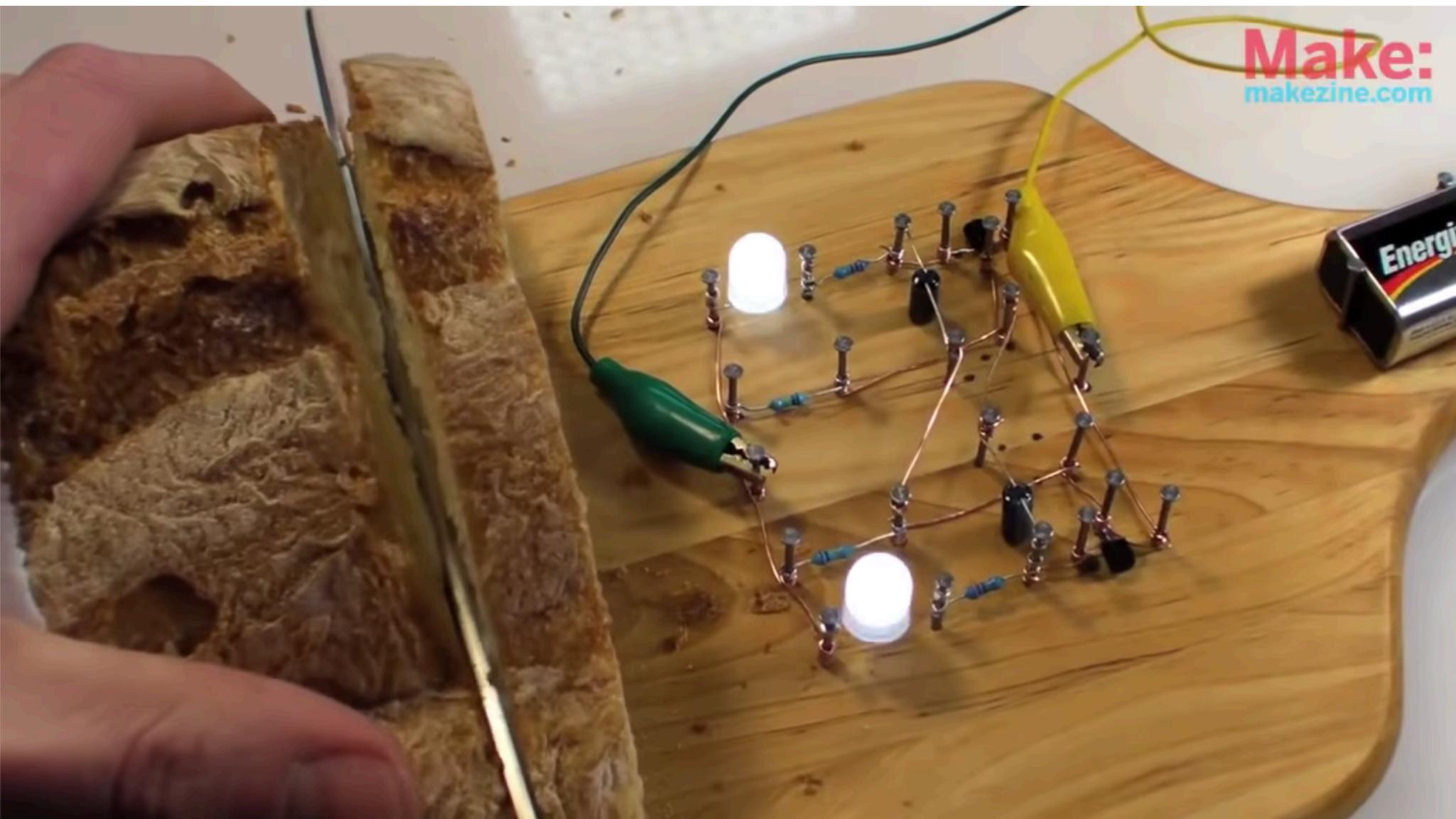
# Circuit Design

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



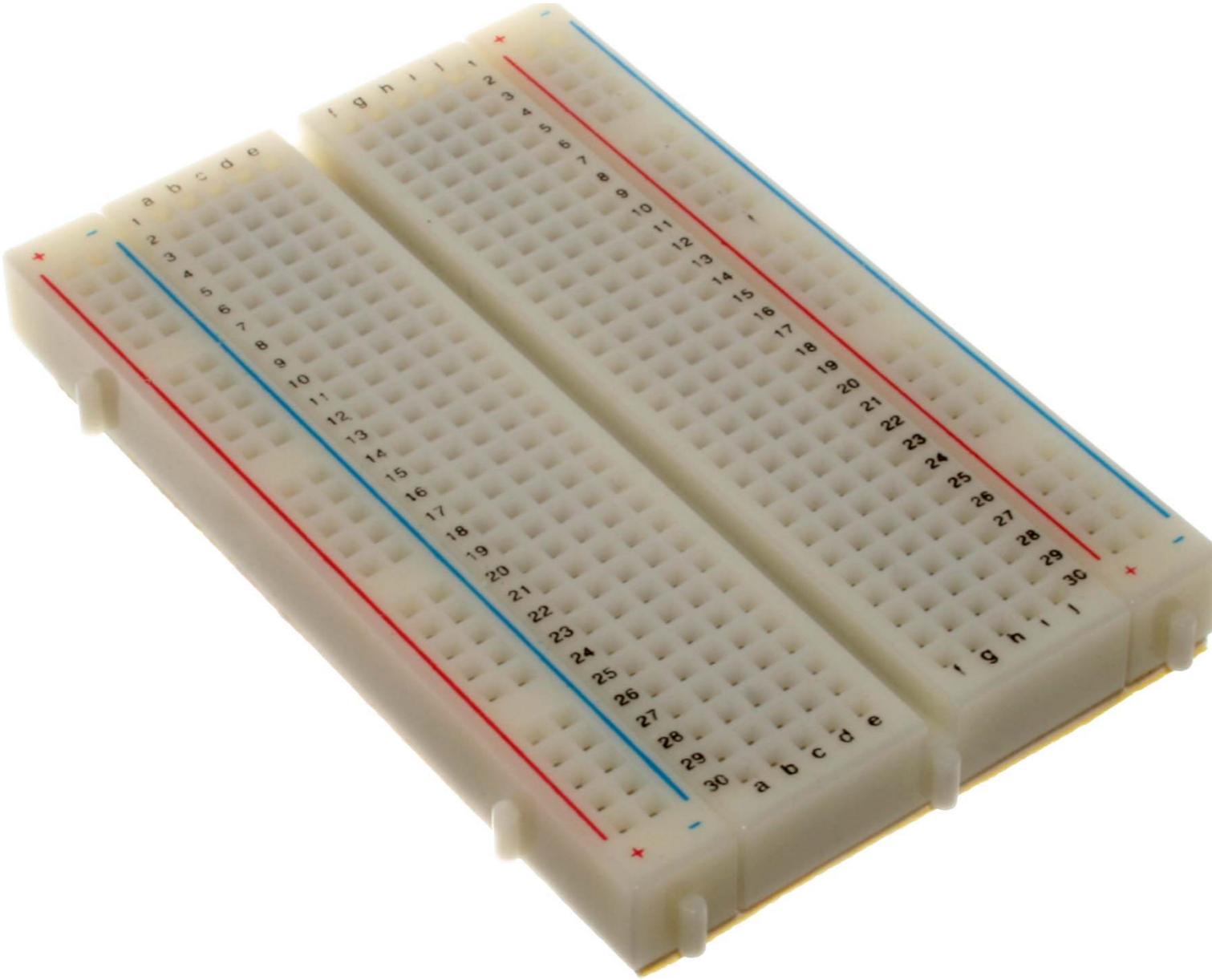
# Bread Board?

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>



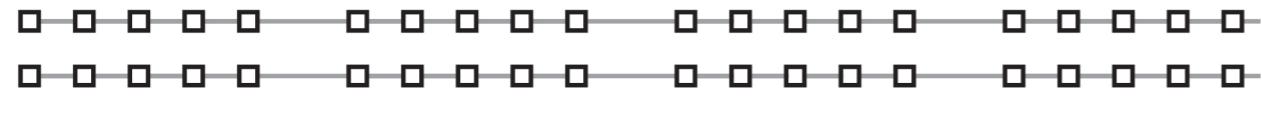
# Bread Board

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



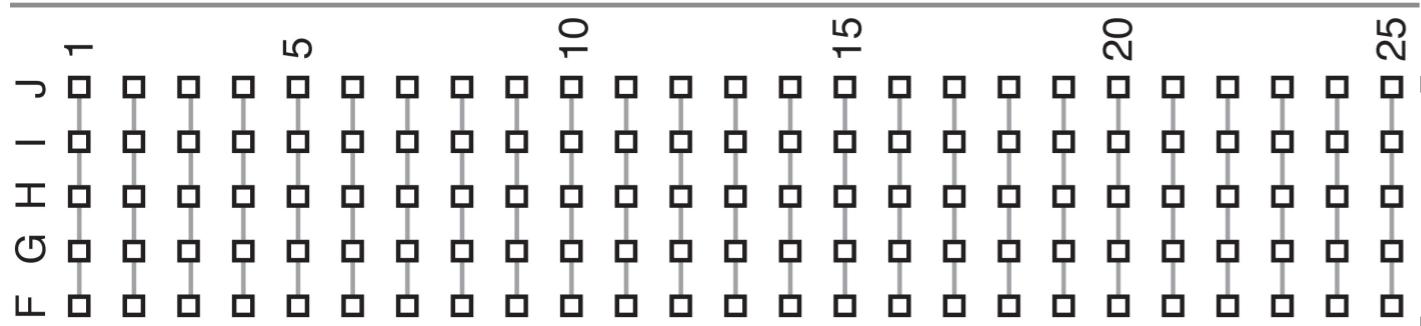
# Solder-less Bread Board

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

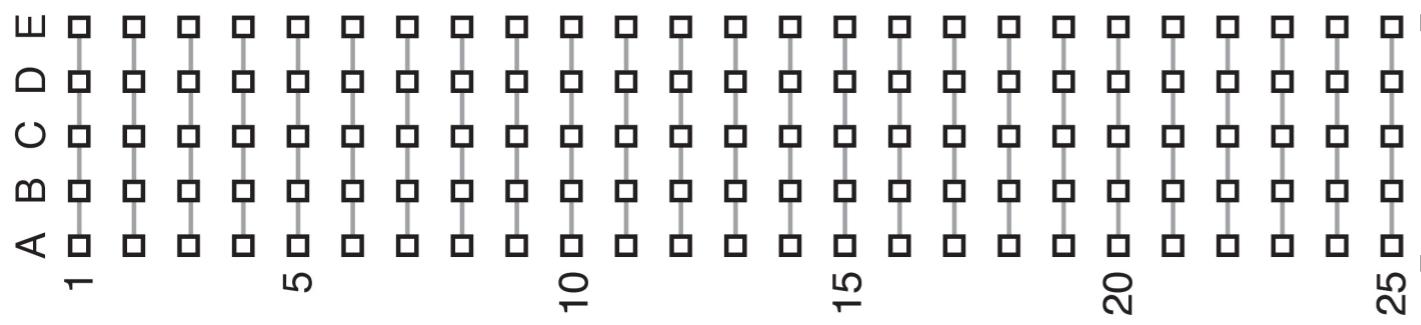


↙ All the holes in this row are connected

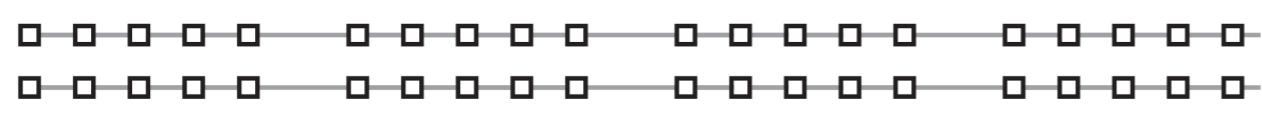
↙ All the holes in this row are connected



} All the holes each column  
are connected

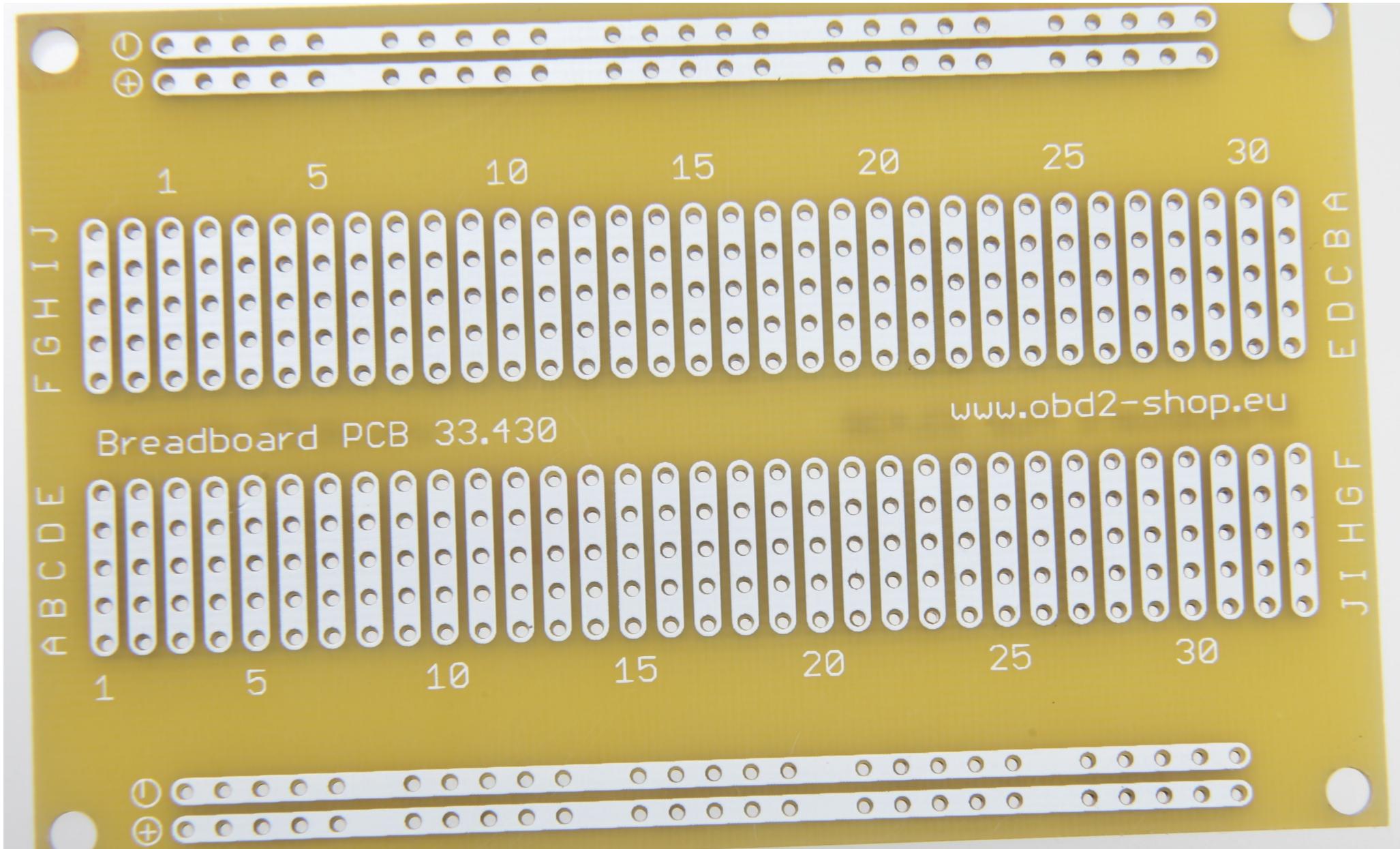


} All the holes each column  
are connected



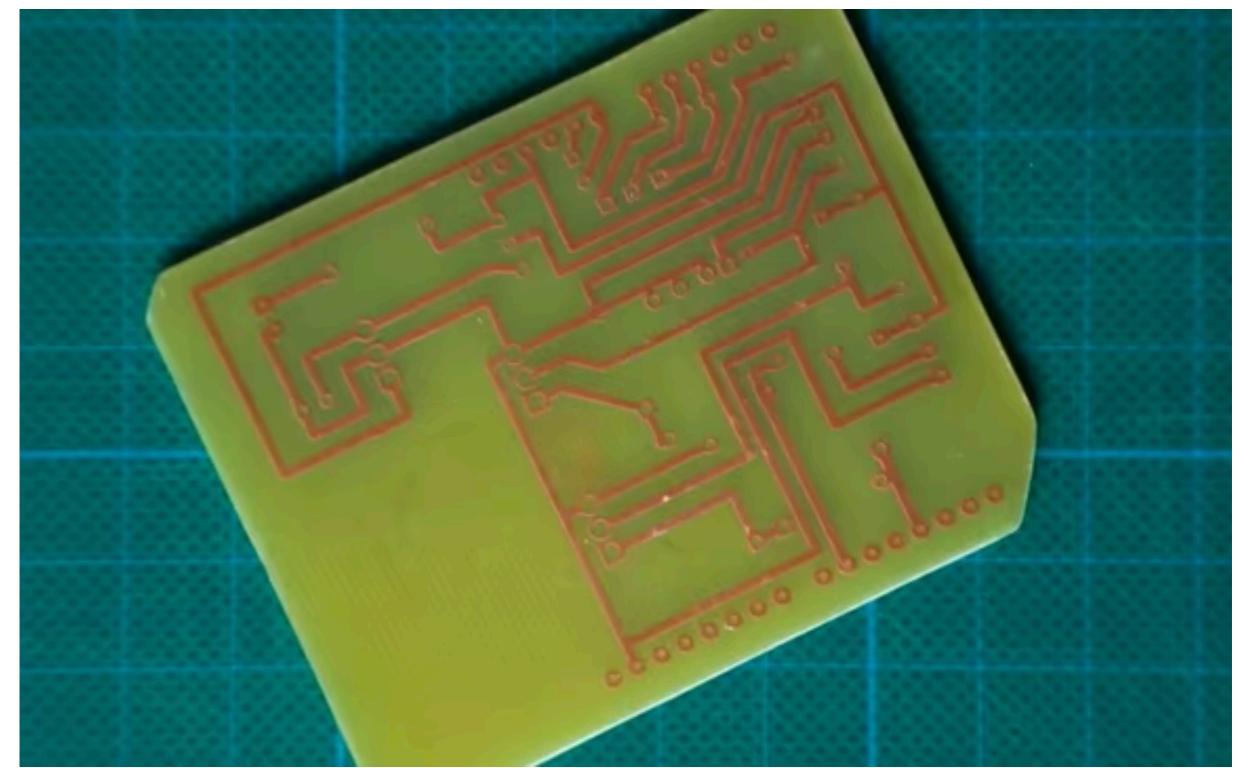
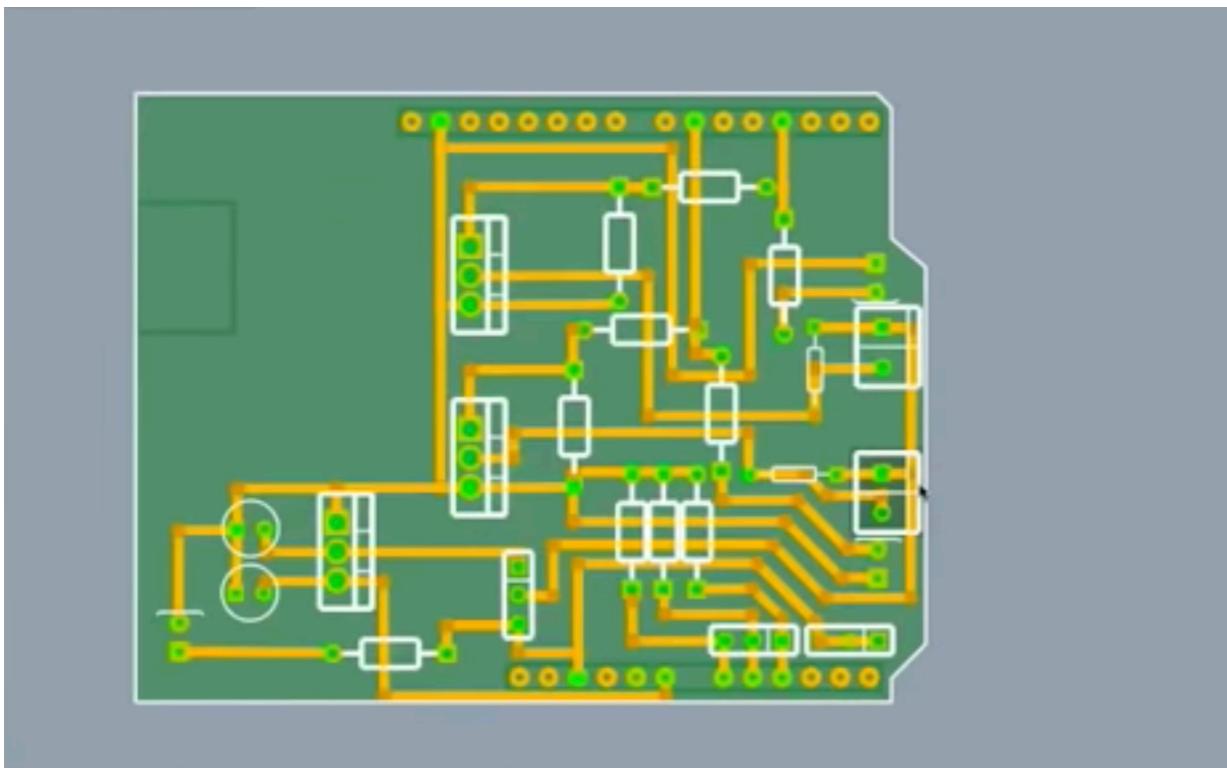
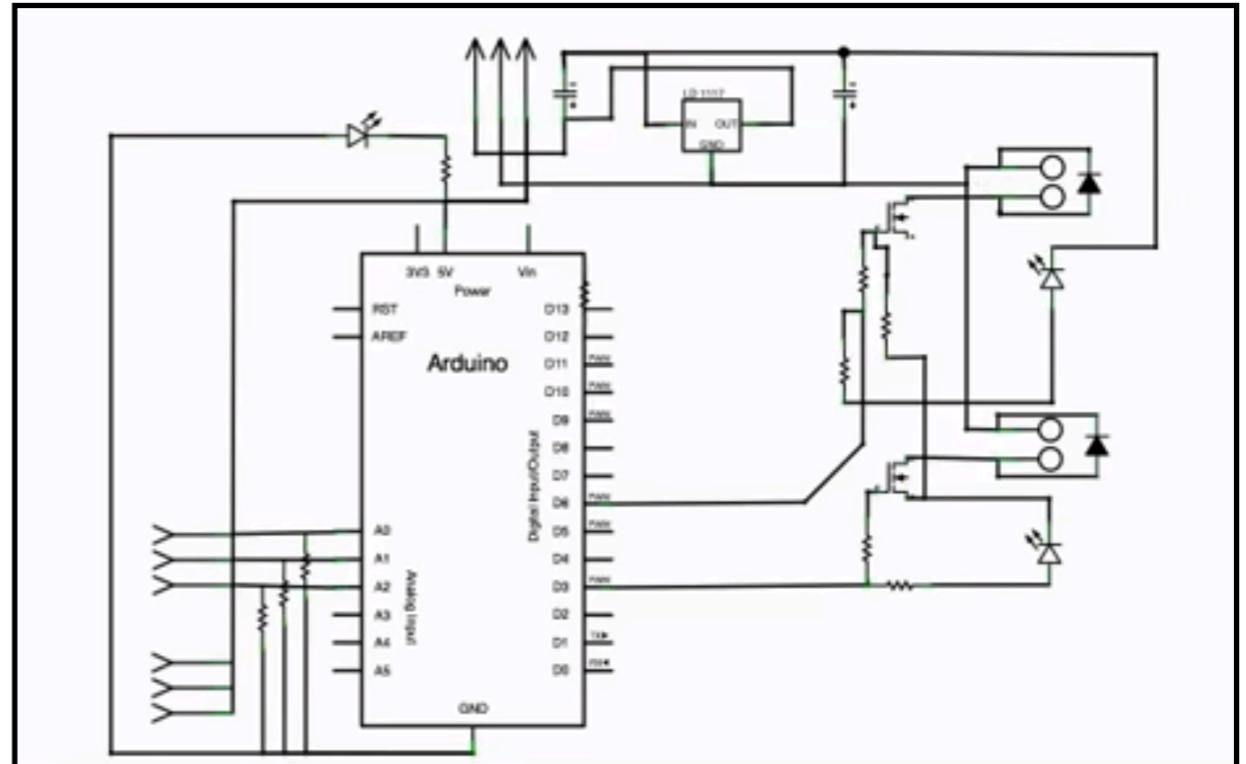
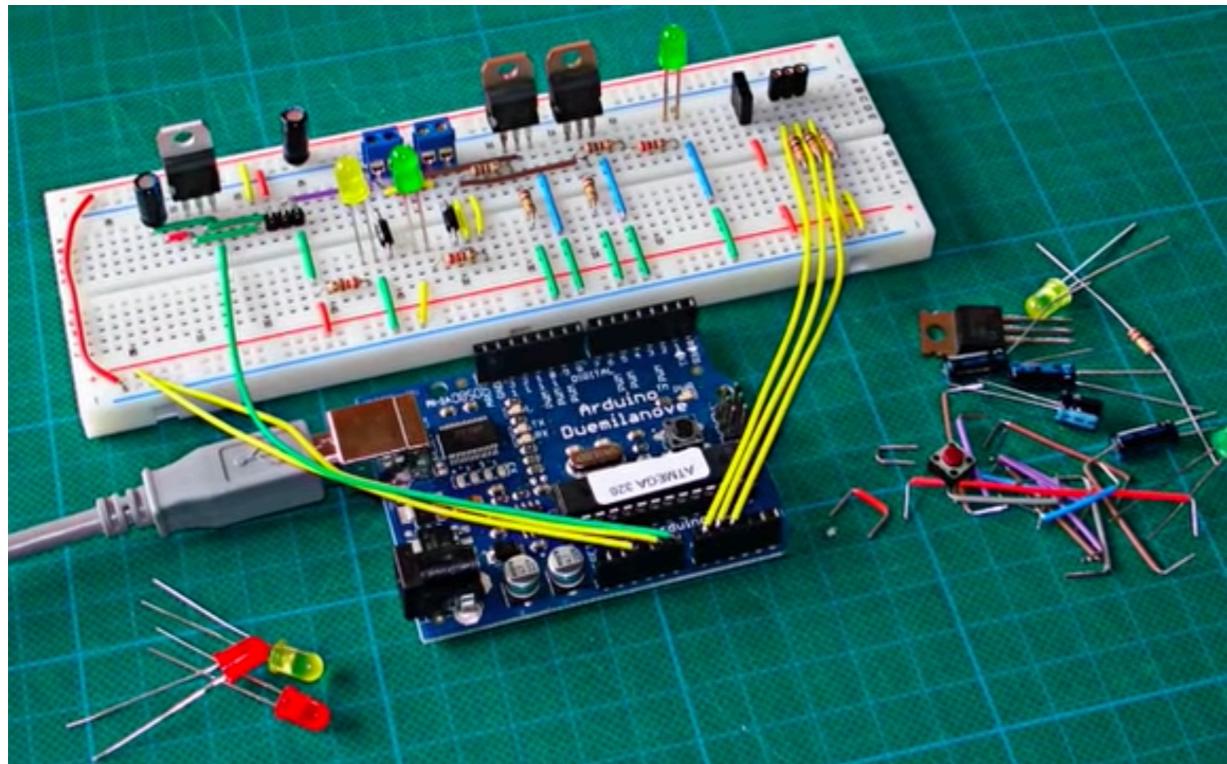
↙ All the holes in this row are connected

↙ All the holes in this row are connected



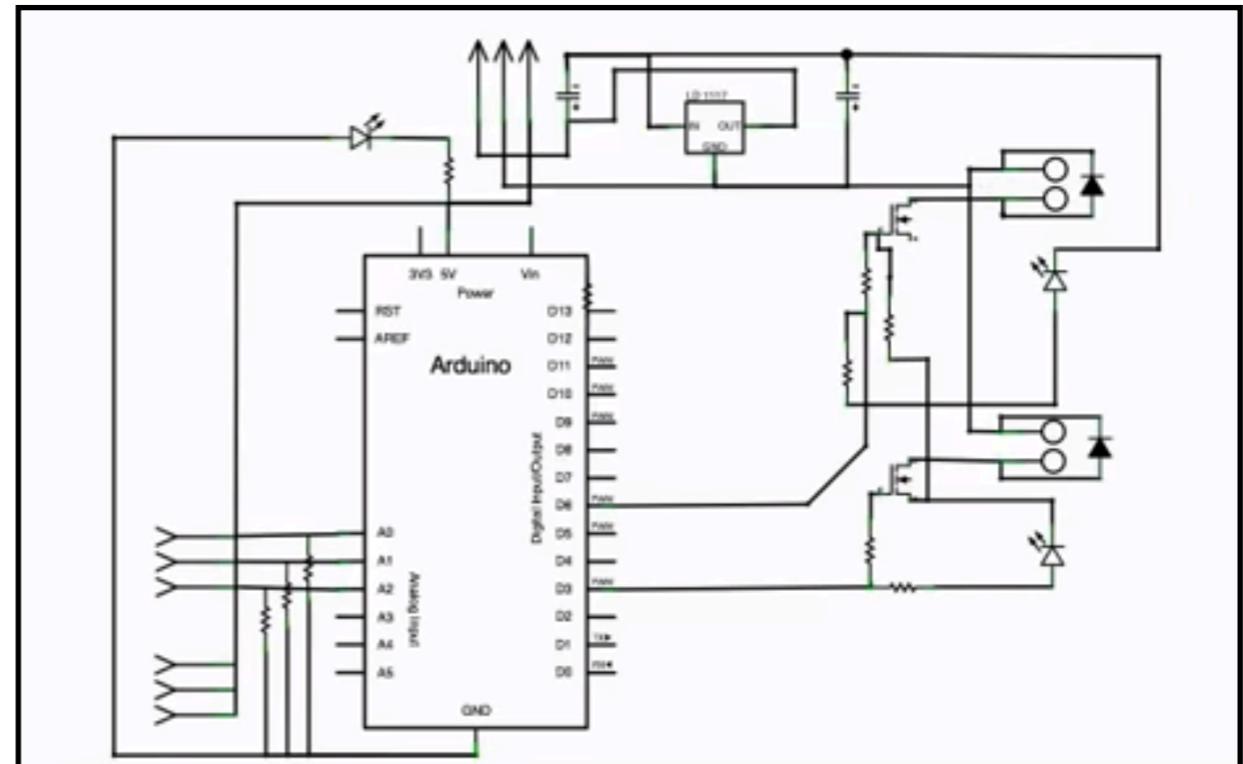
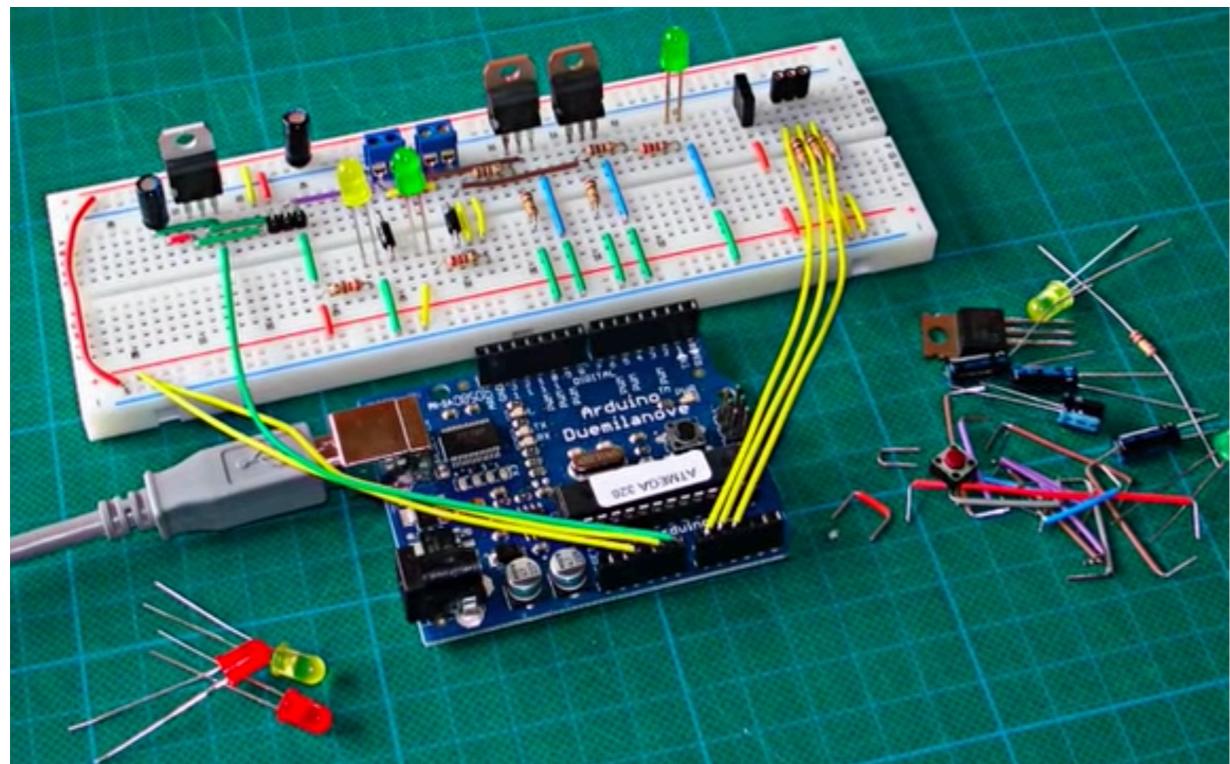
# Printed Circuit Board

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

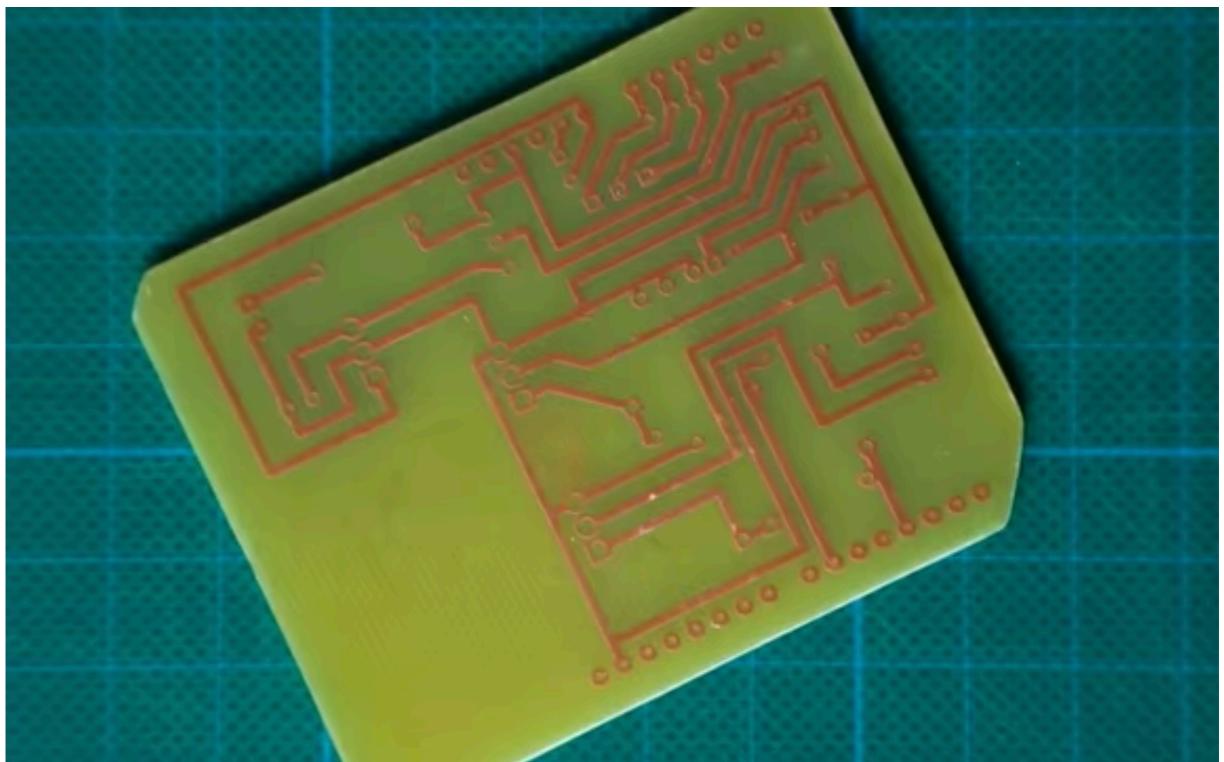
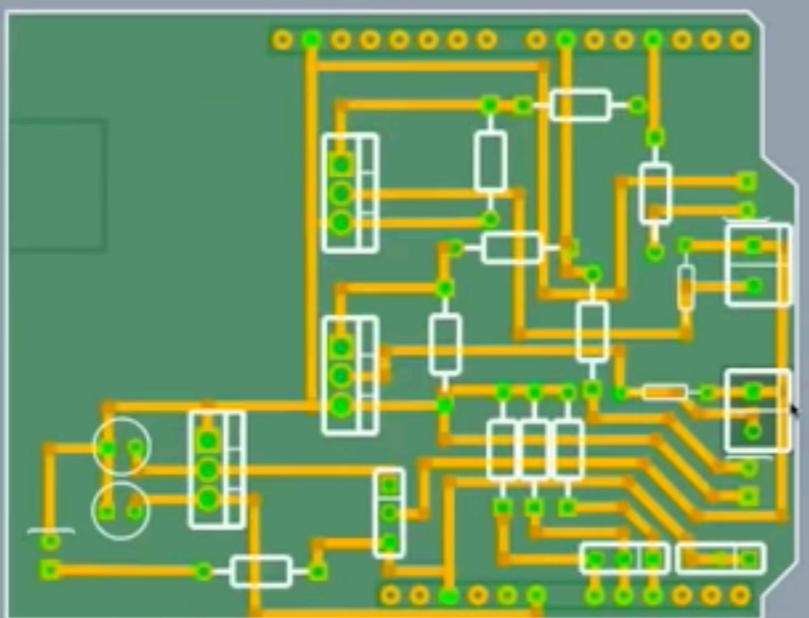


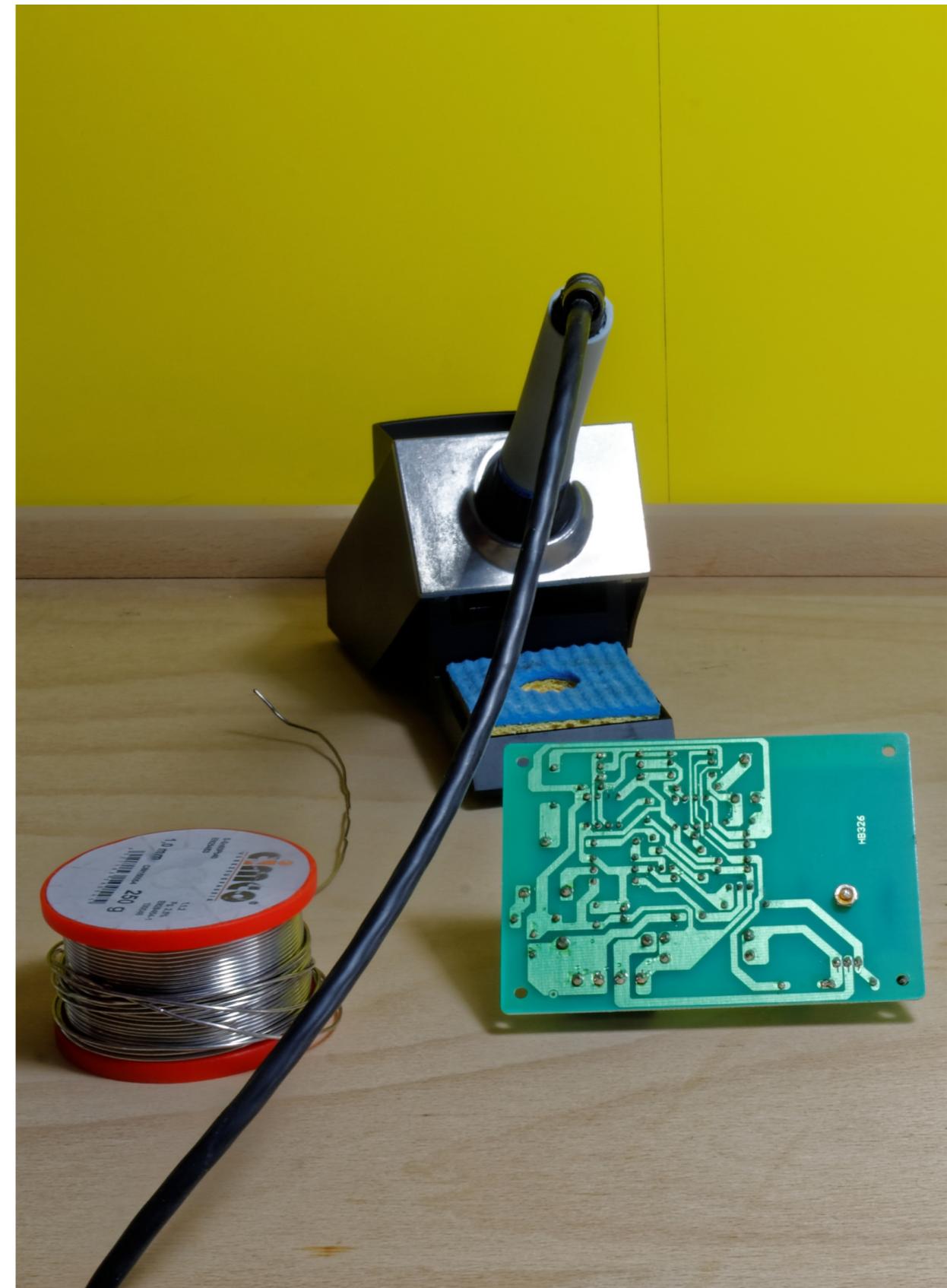
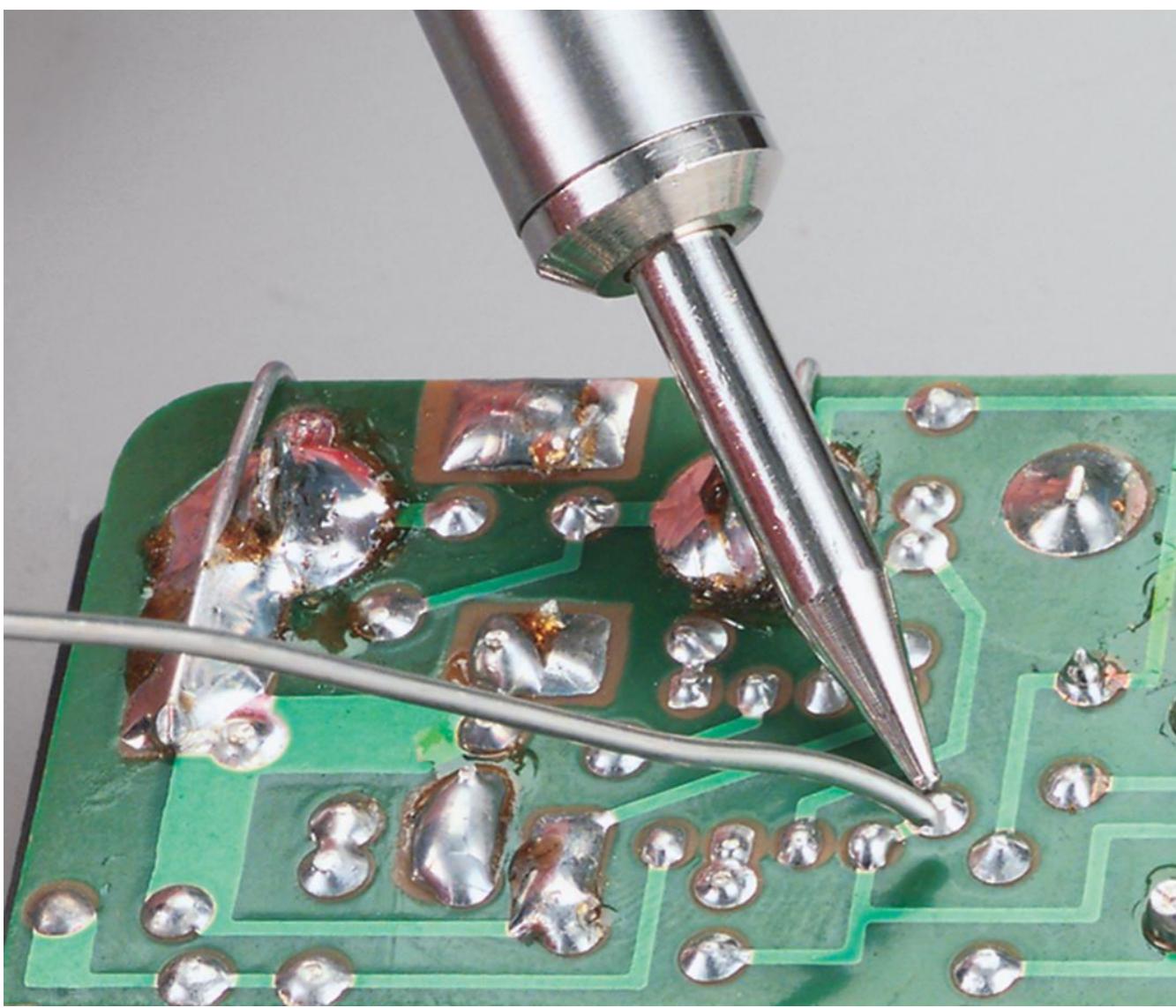
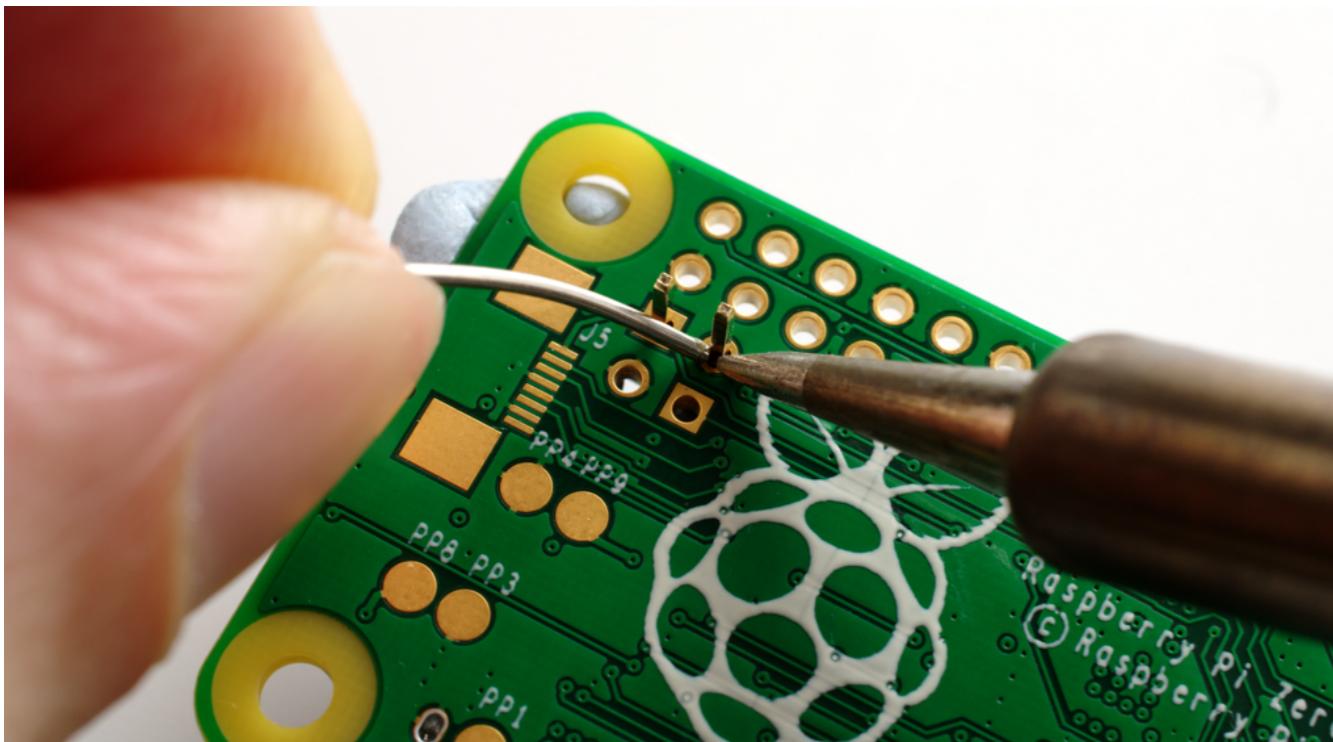
By: <http://fritzing.org/home/>

# From Prototype to Commercial Electronics



# From Prototype to Commercial Electronics



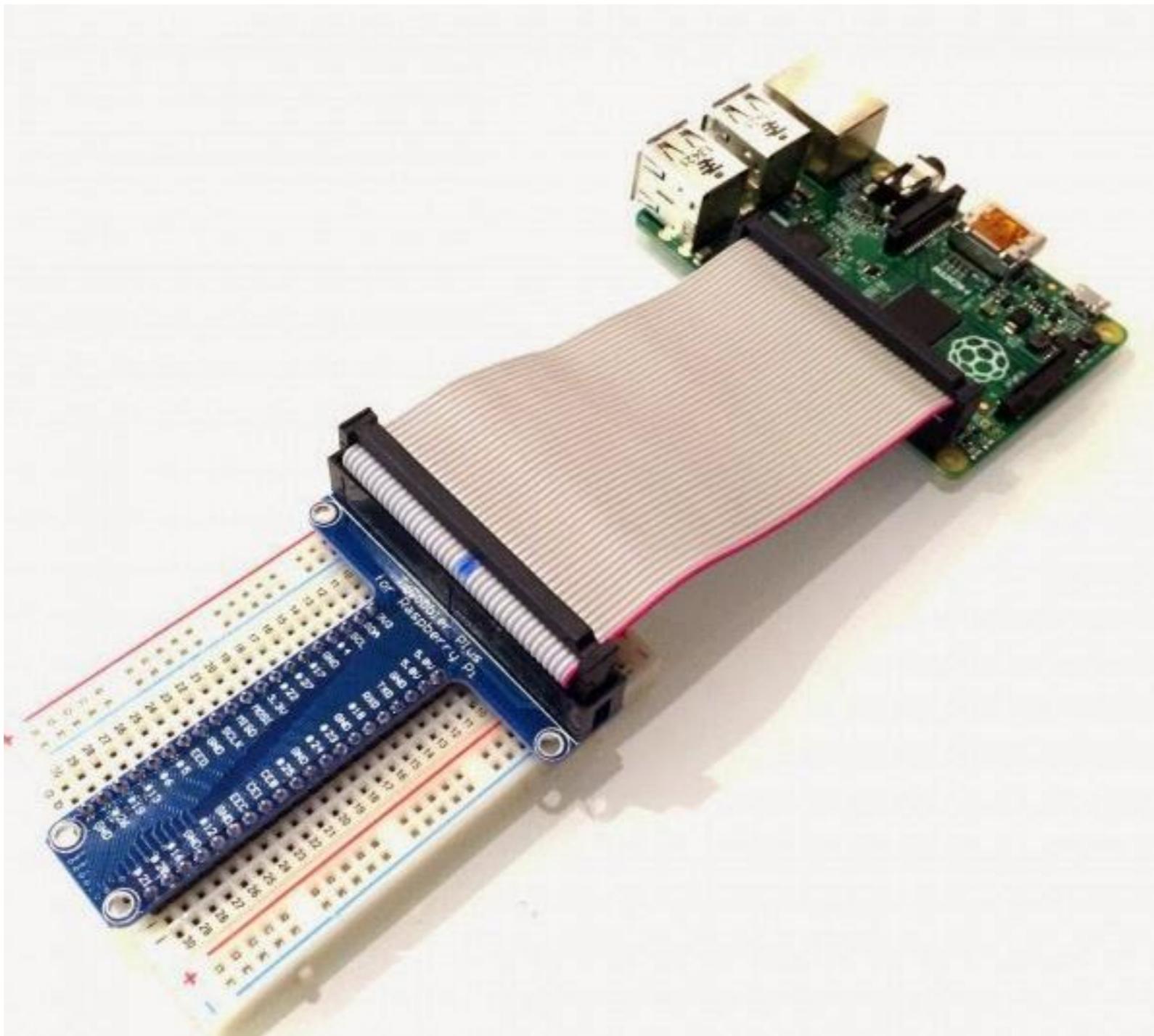


Top left: Gareth Halfacree from Bradford, UK [CC BY-SA 2.0 (<https://creativecommons.org/licenses/by-sa/2.0/>)]

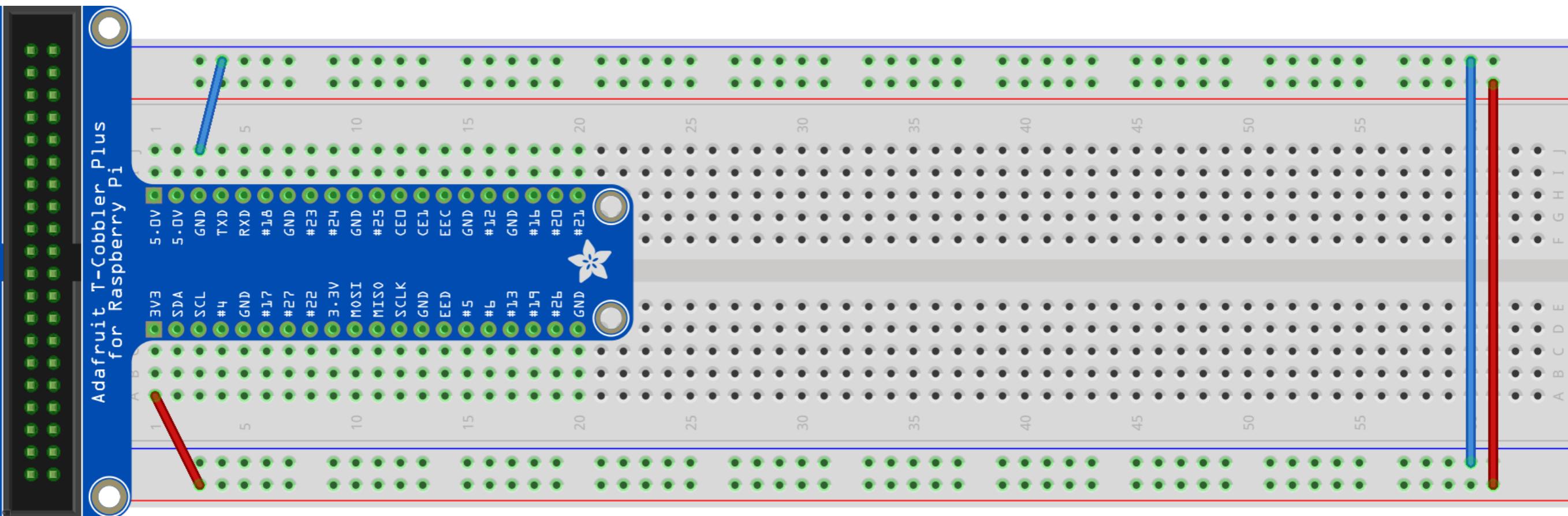
Botton left: <https://m.media-amazon.com/images/S/aplus-media/vc/a40c0904-3a4c-427f-86cd-95a04a4bbf87.jpg>

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

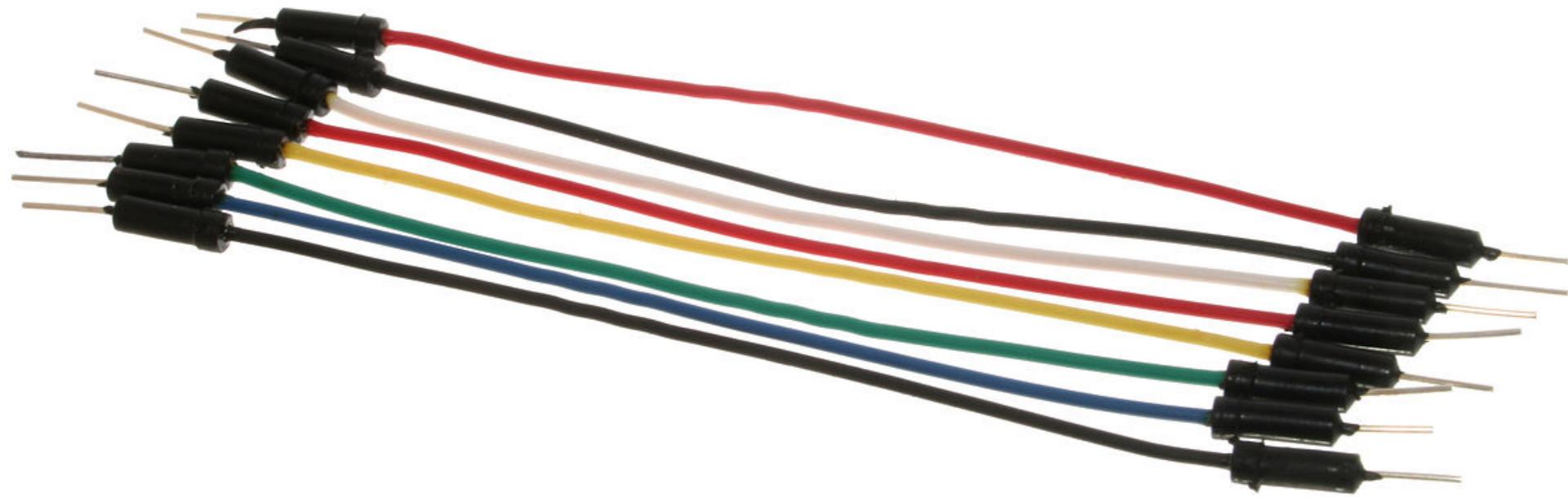
# Initial Set Up



# Initial Power Connection



fritzing



# 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>*

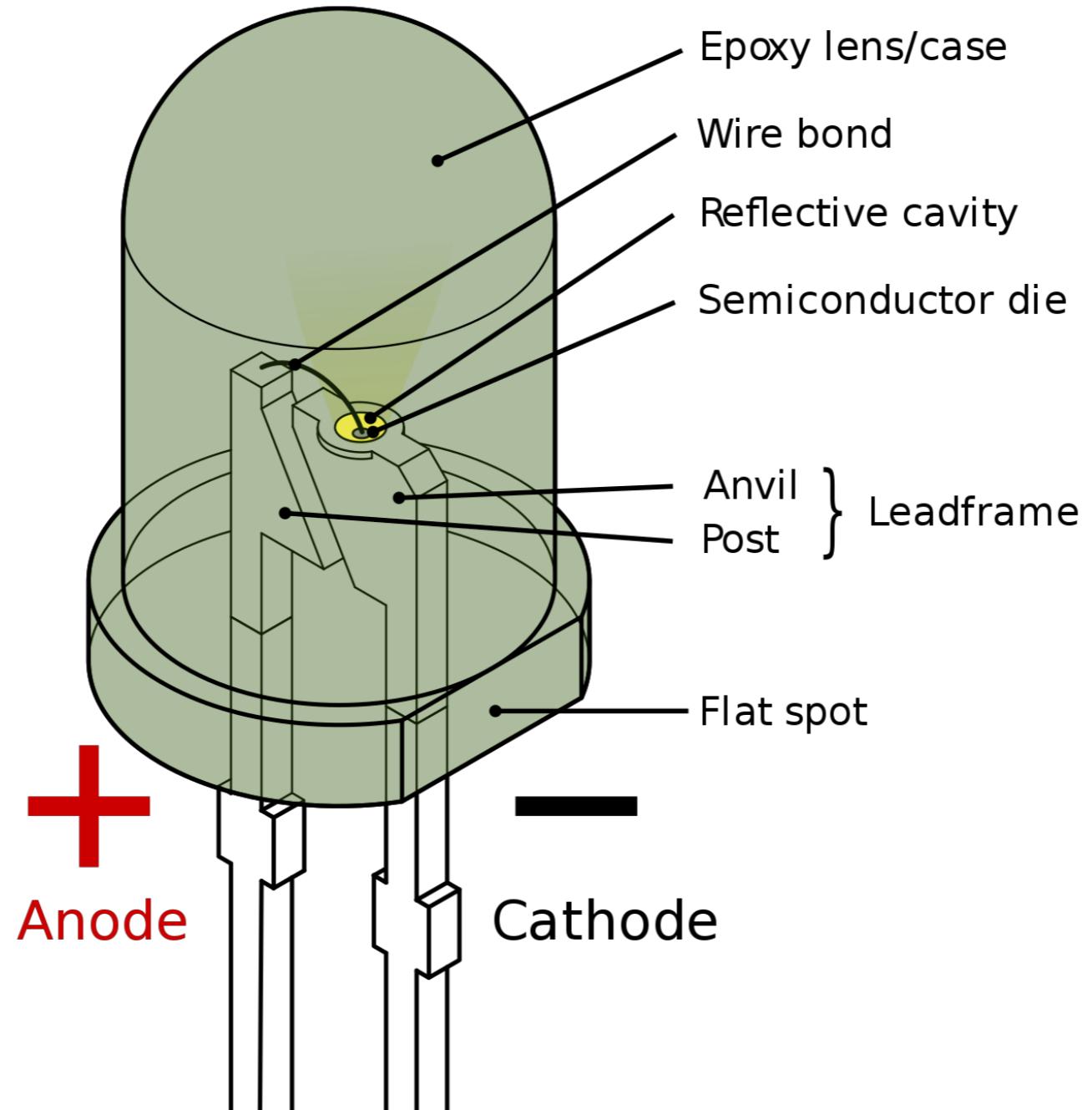


# Jumper Wire

*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>*

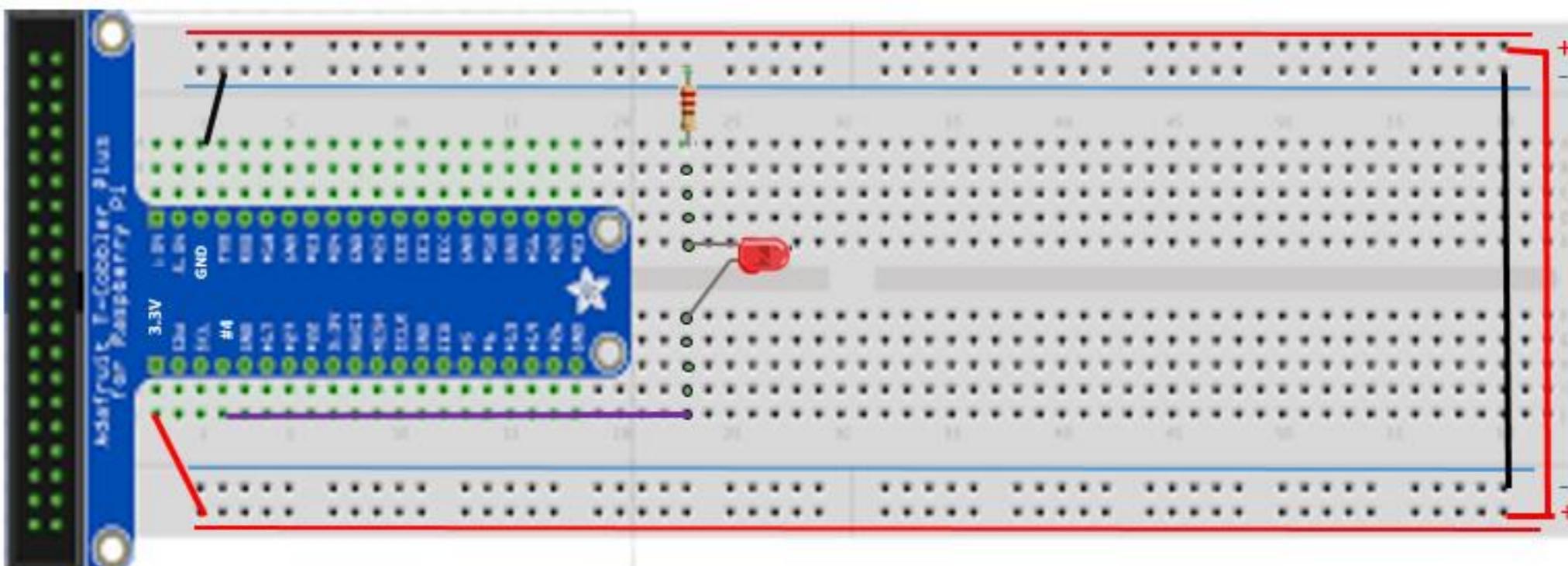
# Gadgets

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



# Light-emitting Diode

# LED - Circuit



```
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(1)
print "LED off"
GPIO.output(4,GPIO.LOW)
```

## **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**

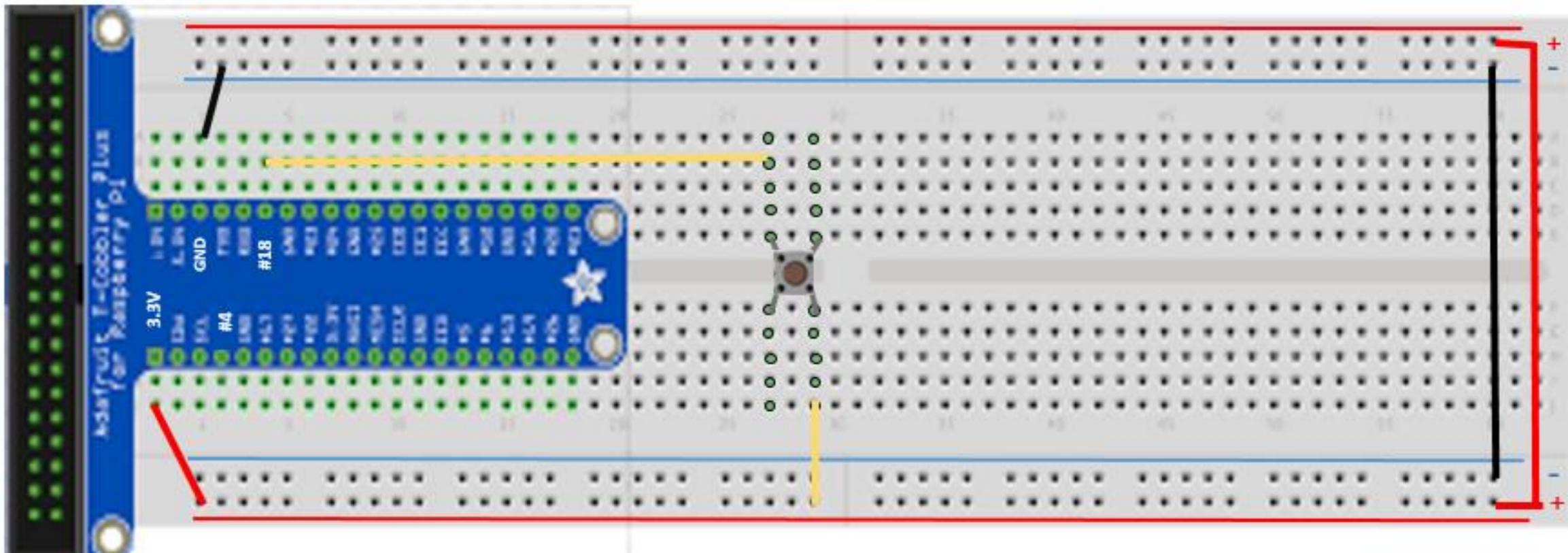
# **Light-dependent Resistor**

# **QUIZ TIME ! \_**

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

# Switch

# Switch - Circuit



```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

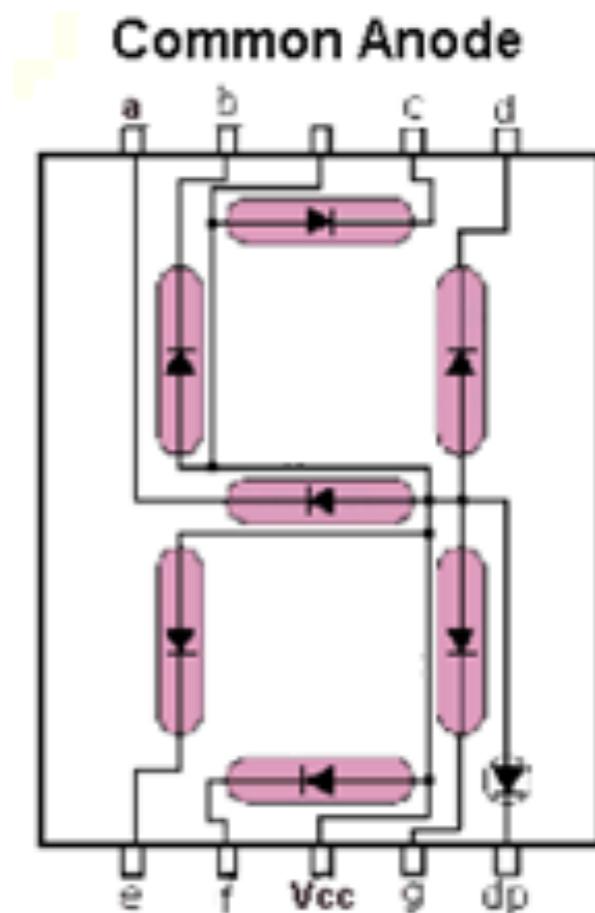
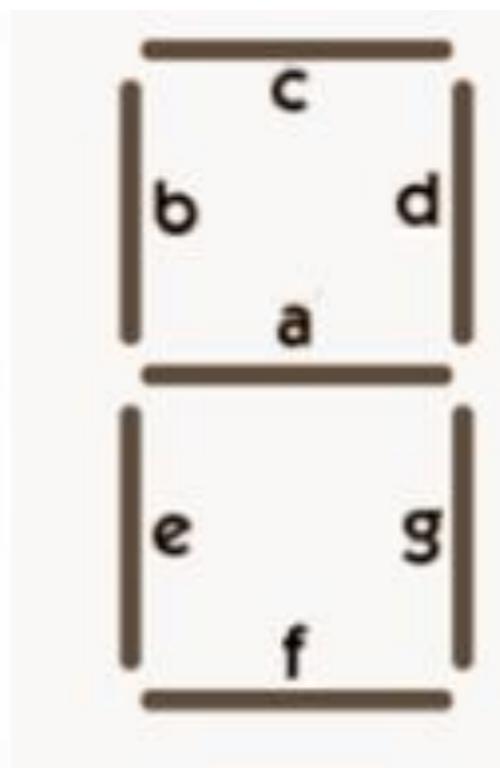
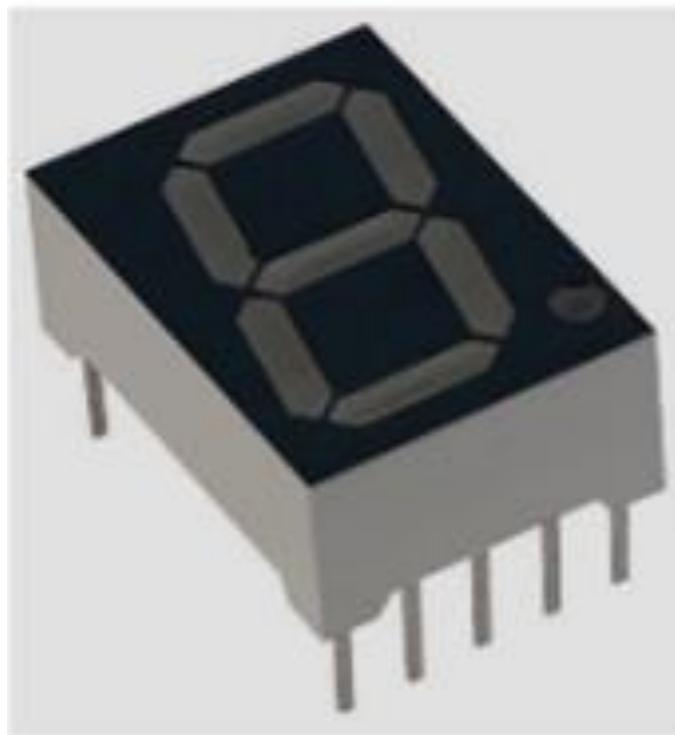
GPIO.setup(18, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

while True:
    input_state = GPIO.input(18)
    if input_state == 1:
        print('Button Pressed')
        time.sleep(0.2)
```

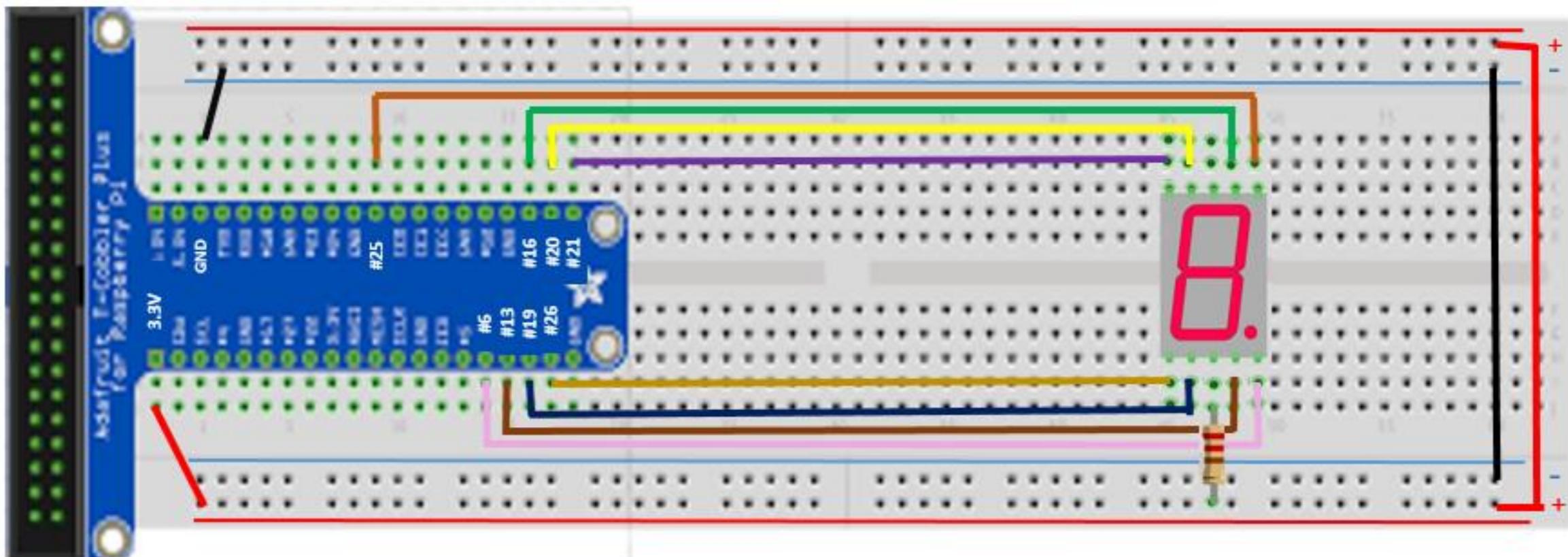
# **QUIZ TIME ! \_**

1. Count the number of times the button is pressed and print out on console
2. Make the light switch on when the button is pressed

# 7 Segment



# 7 Segment - Circuit



```
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

# Buzzer

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

[https://github.com/gumslone/raspi\\_buzzer\\_player/blob/master/buzzer\\_player.py](https://github.com/gumslone/raspi_buzzer_player/blob/master/buzzer_player.py)

# Buzzer - Circuit

**QUIZ TIME! \_**

**1. Play a song!**

# 4 Digit 7-Seg



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

# 4 Digit 7 Seg - Circuit

```
num = {  
    '0':(0,0,0,0,0,0,1),  
    '1':(1,0,0,1,1,1,1),  
    '2':(0,0,1,0,0,1,0),  
    '3':(0,0,0,0,1,1,0),  
    '4':(1,0,0,1,1,0,0),  
    '5':(0,1,0,0,1,0,0),  
    '6':(0,1,0,0,0,0,0),  
    '7':(0,0,0,1,1,1,1),  
    '8':(0,0,0,0,0,0,0),  
    '9':(0,0,0,0,1,0,0)  
}  
  
display = '8901'  
  
GPIO.output(23,1) # do not set dots  
try:  
    while True:  
        for digit in range(4):  
            for loop in range(7):  
                GPIO.output(segments[loop], num[display[digit]][loop])  
                GPIO.output(digits[digit], 1)  
                time.sleep(1)  
                GPIO.output(digits[digit], 0)  
finally:  
    GPIO.cleanup()
```

# **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

- Send me your personal GitHub account, to be added as a collaborator.
- Will send the notes by this weekend, take a set of hardware, try them at home
- Test and set up the remaining sets of hardware
- Write a demo page on Github

# 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>