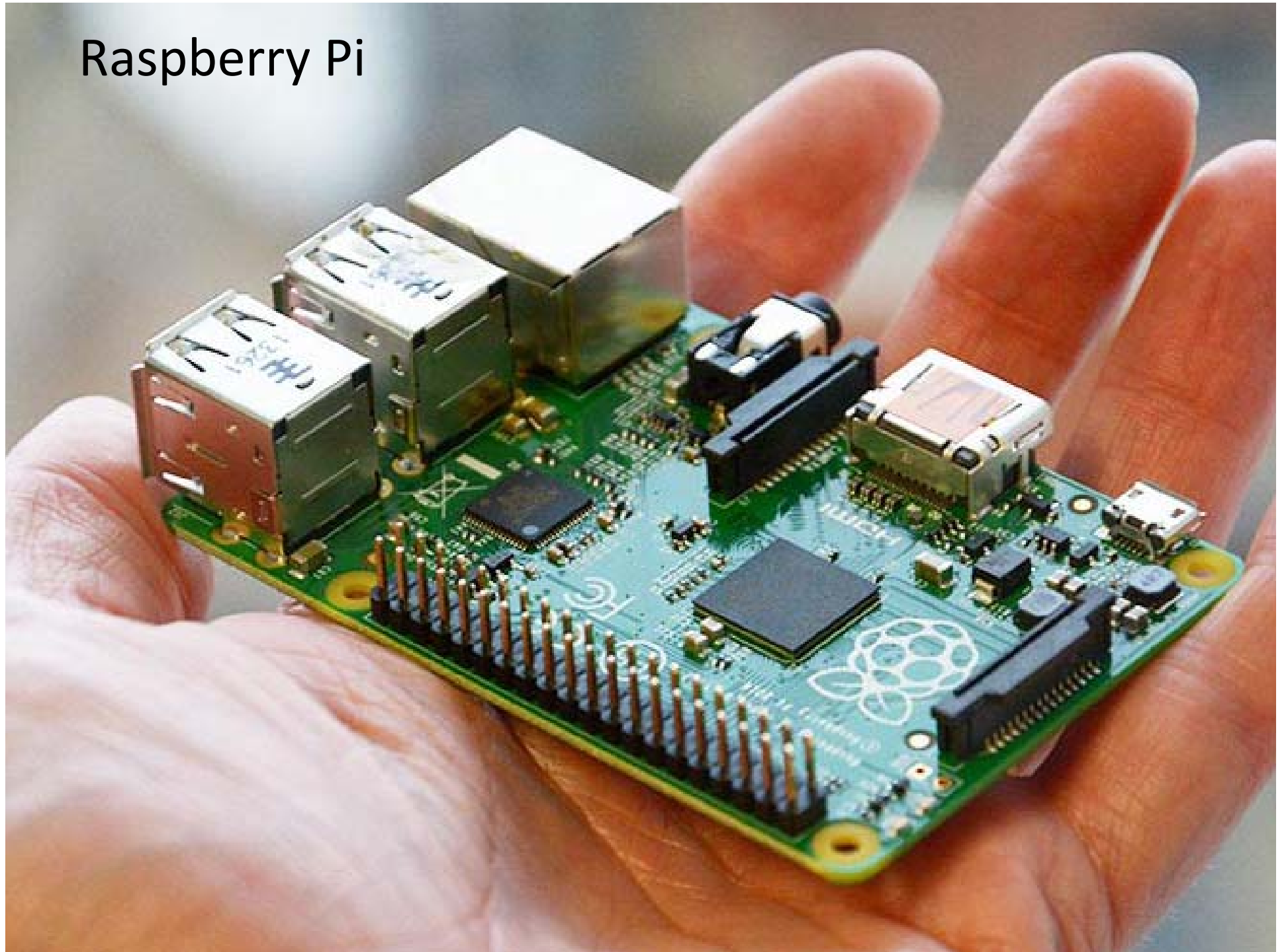
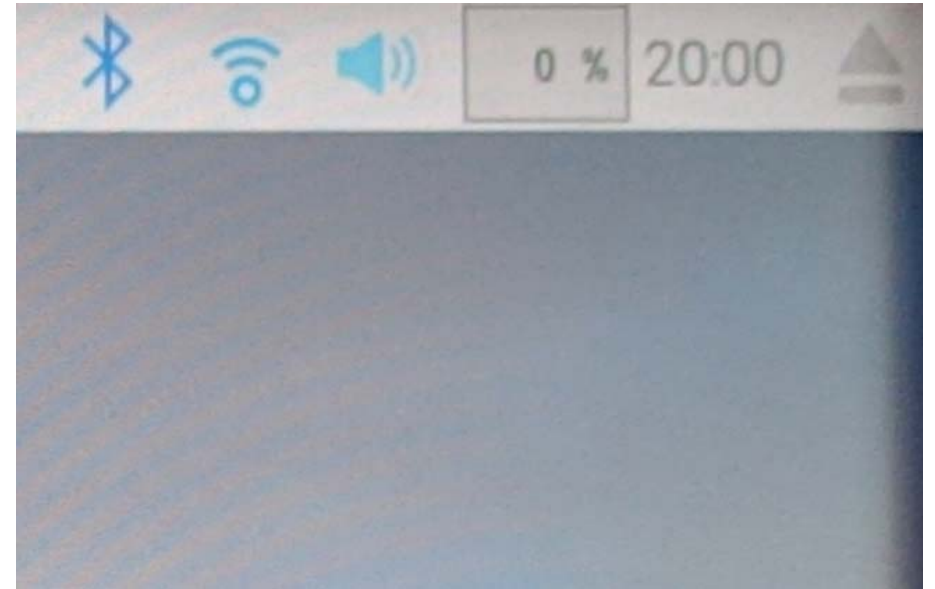
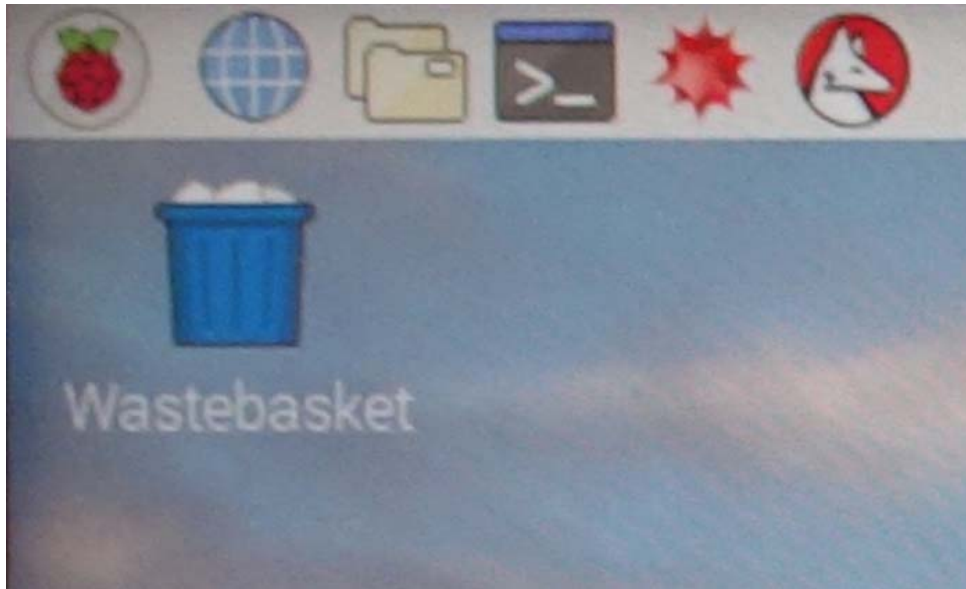
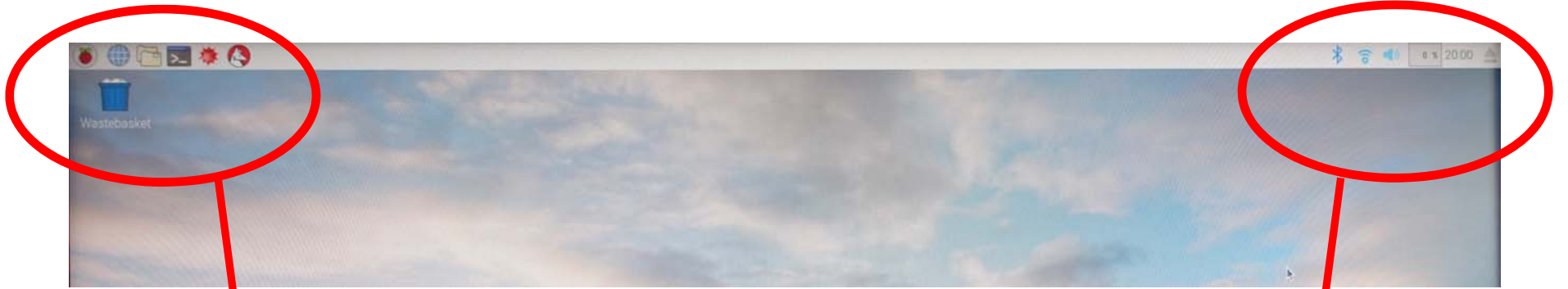


Raspberry pie

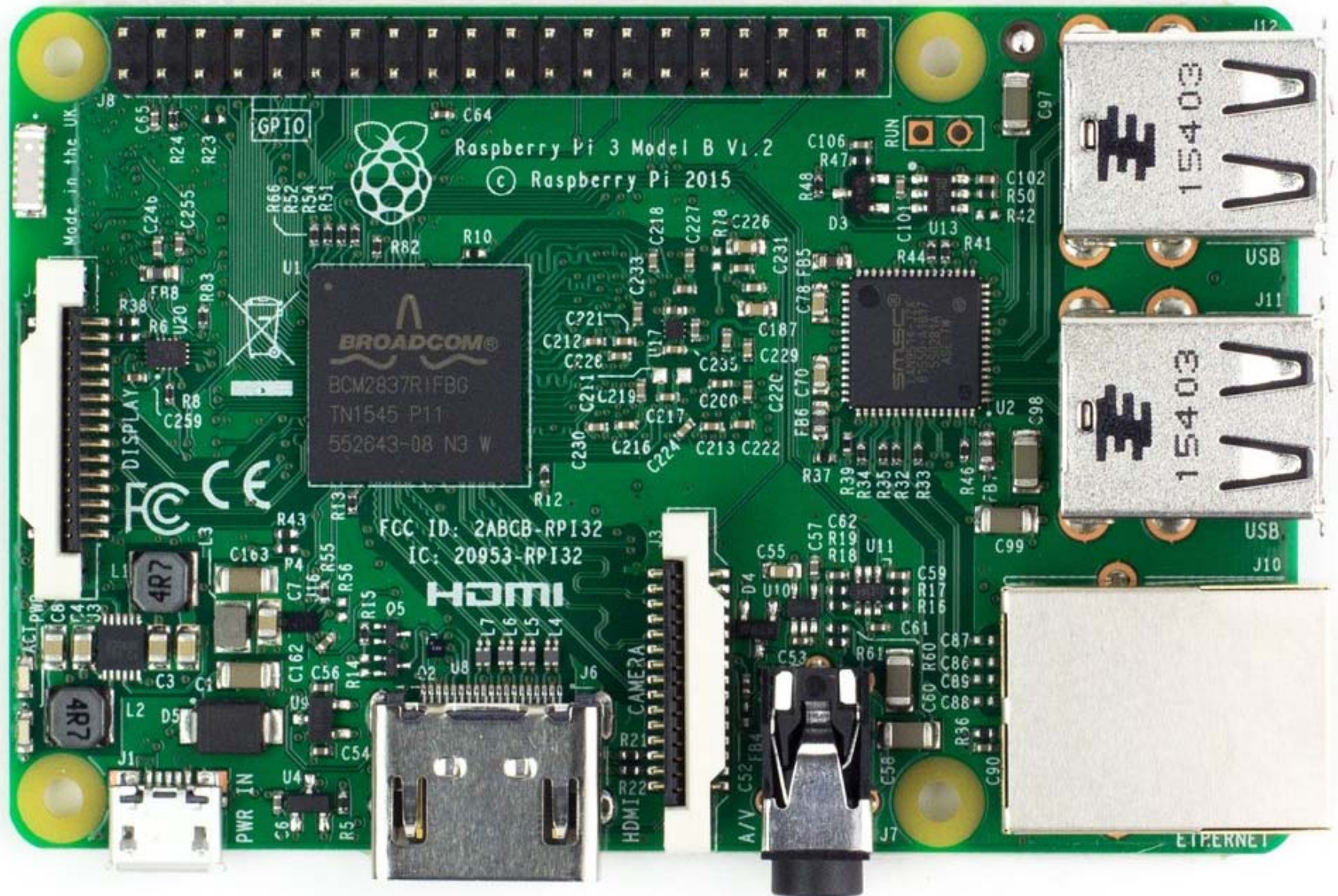


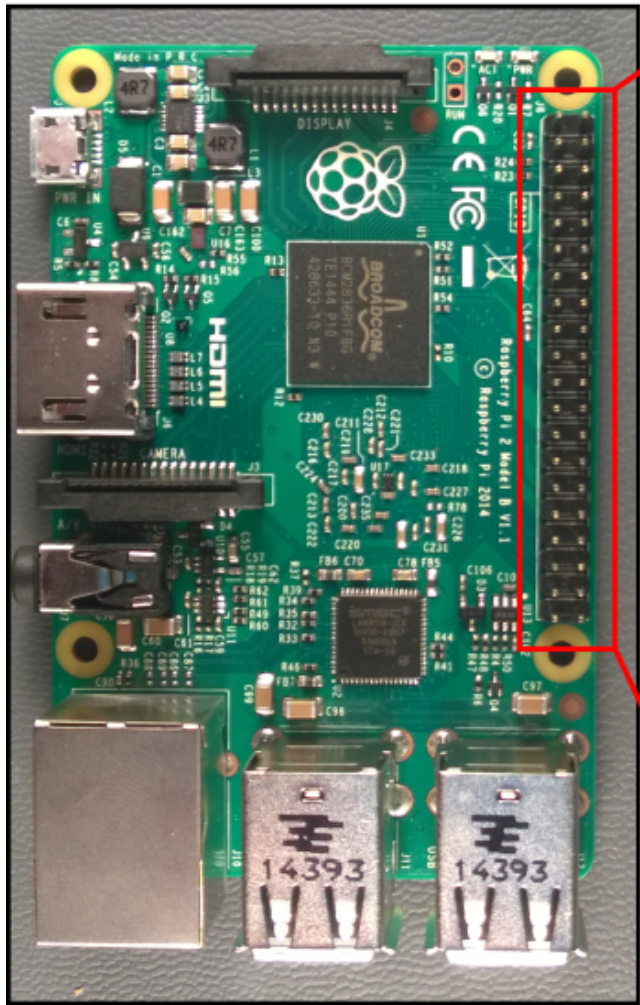
Raspberry Pi



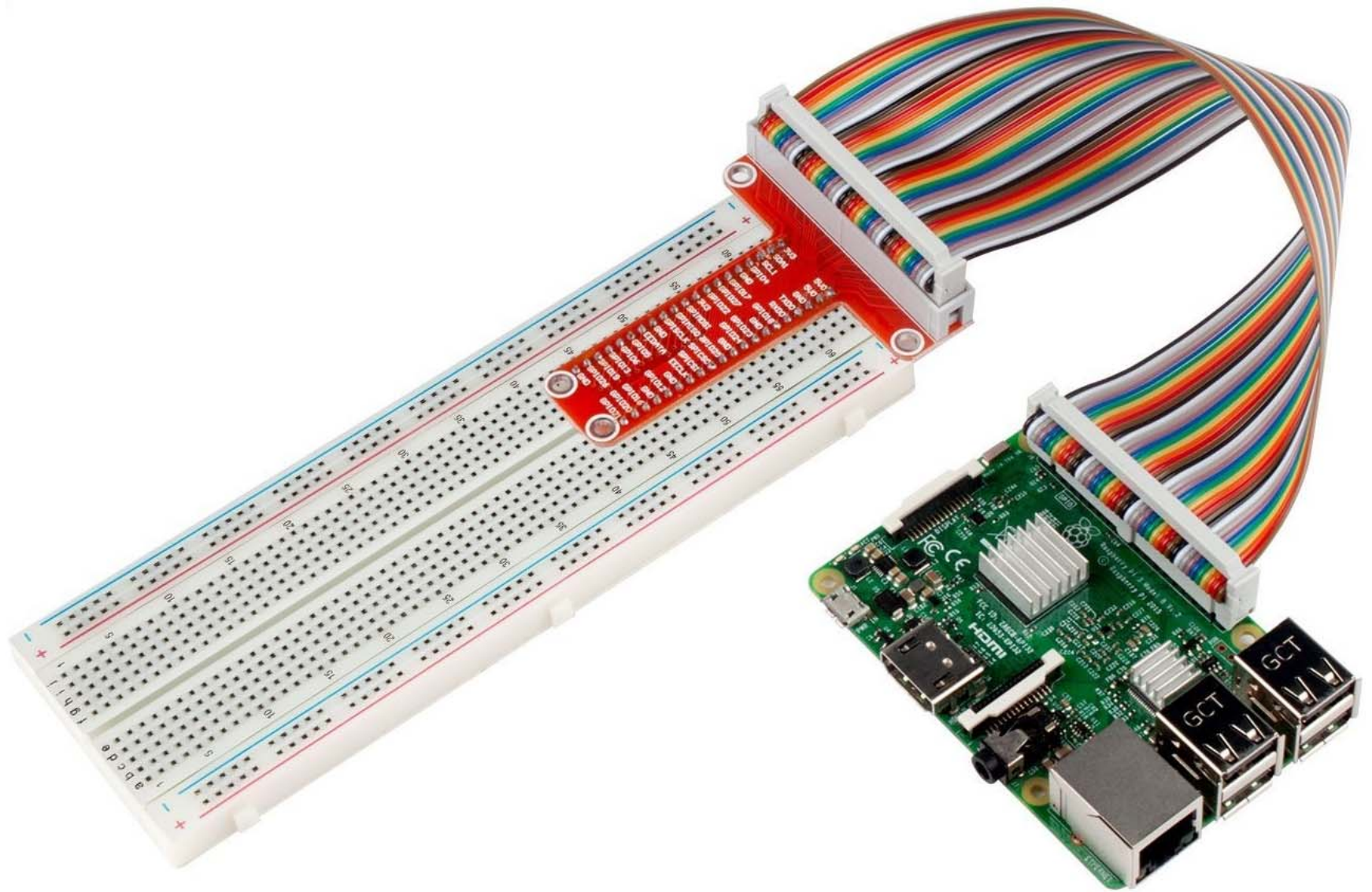


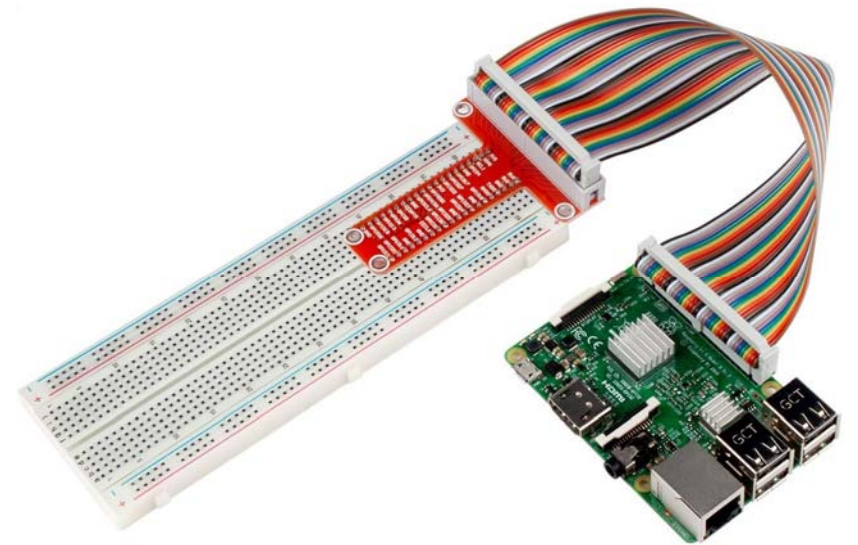
GPIO: General Purpose I/O



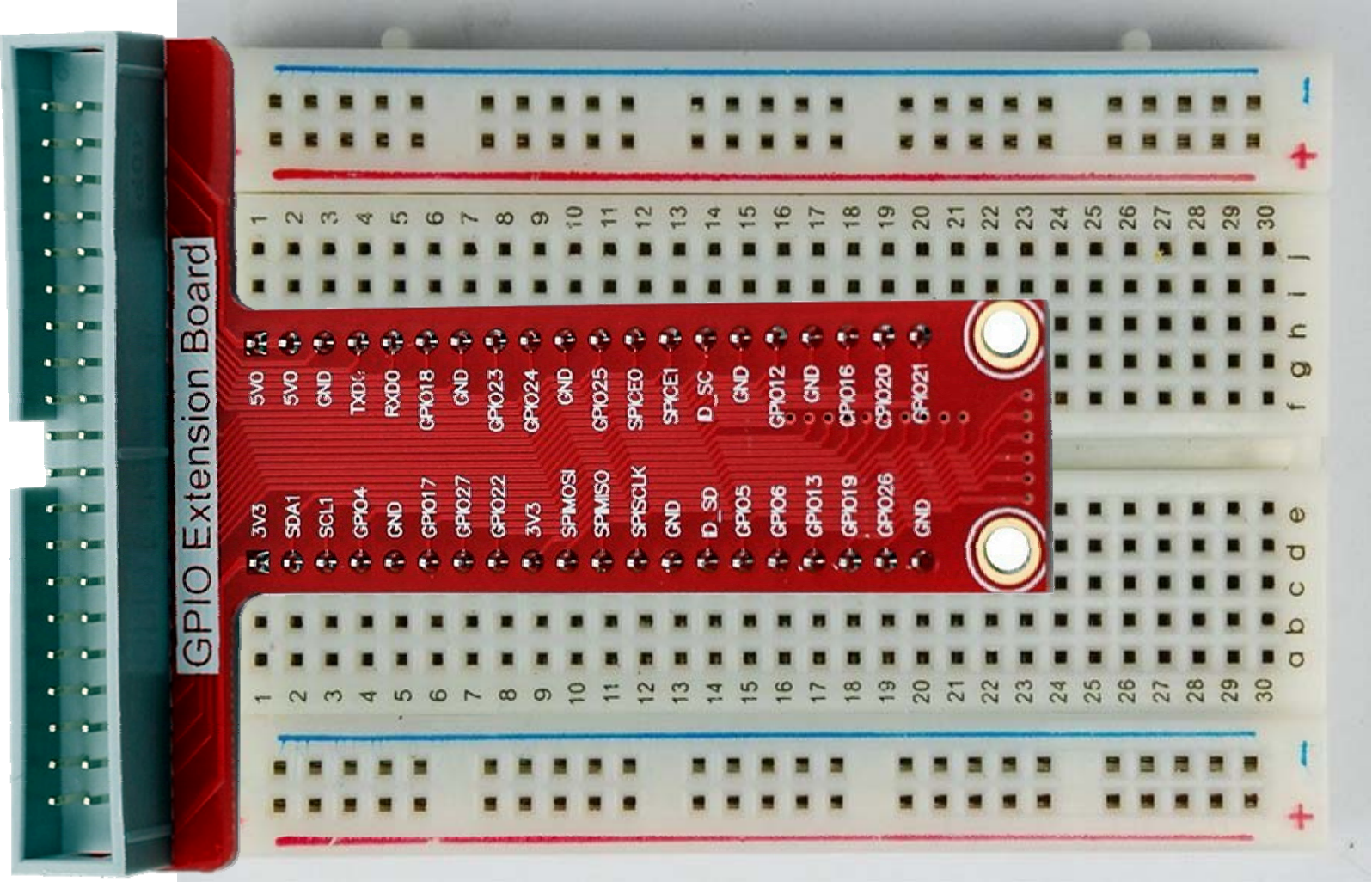


Alternate Function					Alternate Function	
	3.3V PWR	1		2	5V PWR	
I2C1 SDA	GPIO 2	3		4	5V PWR	
I2C1 SCL	GPIO 3	5		6	GND	
	GPIO 4	7		8	UART0 TX	
	GND	9		10	UART0 RX	
	GPIO 17	11		12	GPIO 18	
	GPIO 27	13		14	GND	
	GPIO 22	15		16	GPIO 23	
	3.3V PWR	17		18	GPIO 24	
SPI0 MOSI	GPIO 10	19		20	GND	
SPI0 MISO	GPIO 9	21		22	GPIO 25	
SPI0 SCLK	GPIO 11	23		24	GPIO 8	SPI0 CS0
	GND	25		26	GPIO 7	SPI0 CS1
	Reserved	27		28	Reserved	
	GPIO 5	29		30	GND	
	GPIO 6	31		32	GPIO 12	
	GPIO 13	33		34	GND	
SPI1 MISO	GPIO 19	35		36	GPIO 16	SPI1 CS0
	GPIO 26	37		38	GPIO 20	SPI1 MOSI
	GND	39		40	GPIO 21	SPI1 SCLK

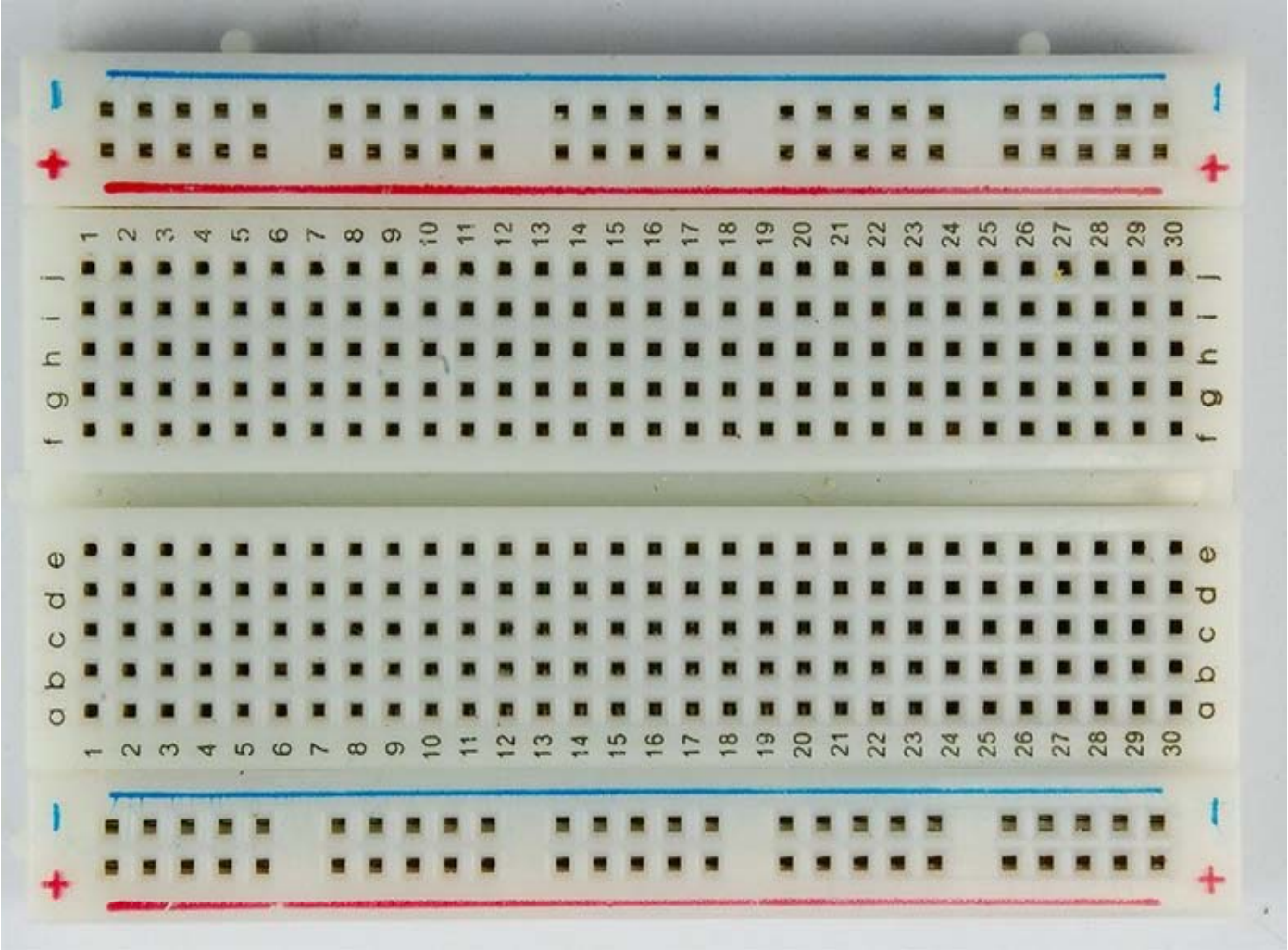


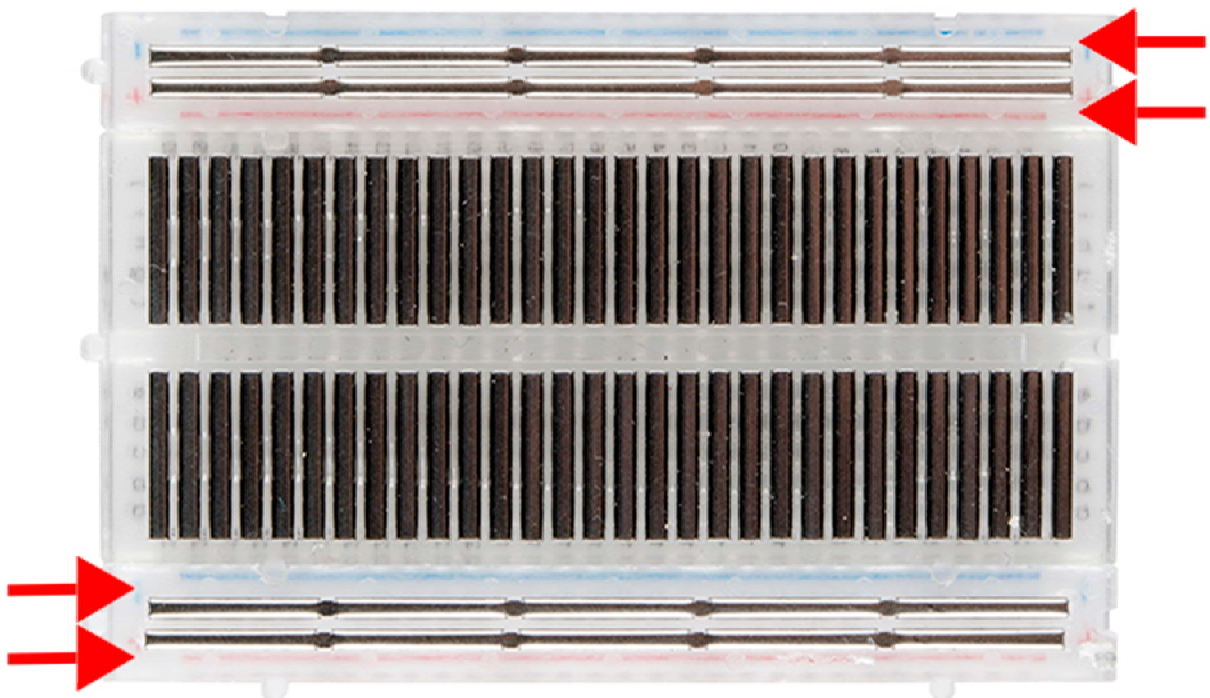
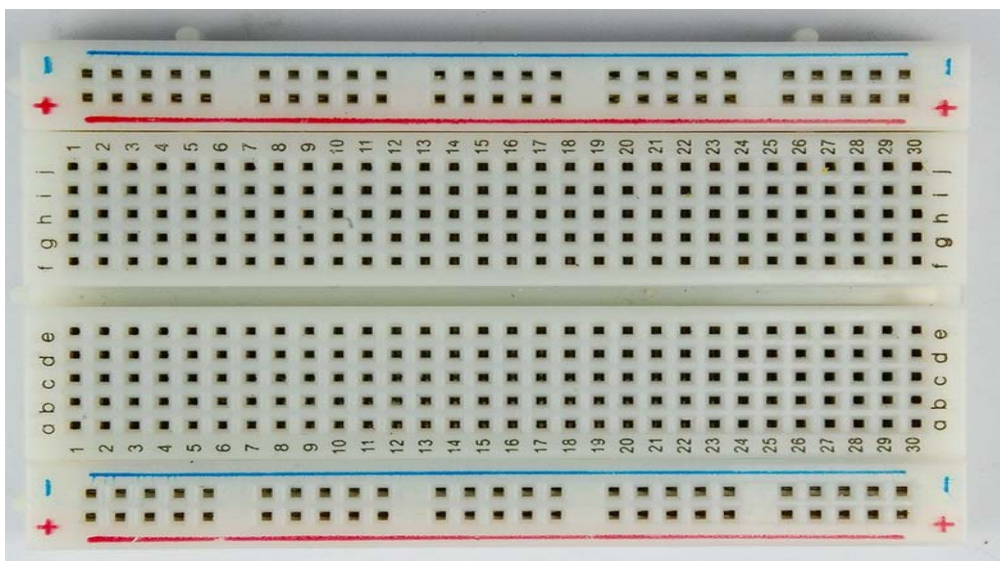


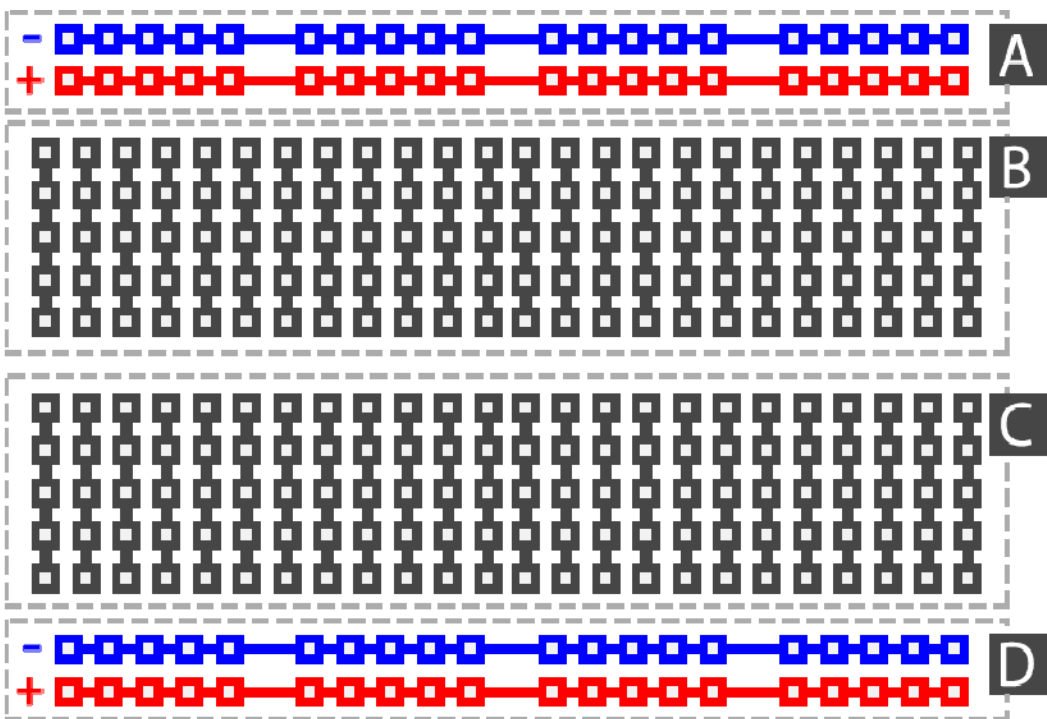
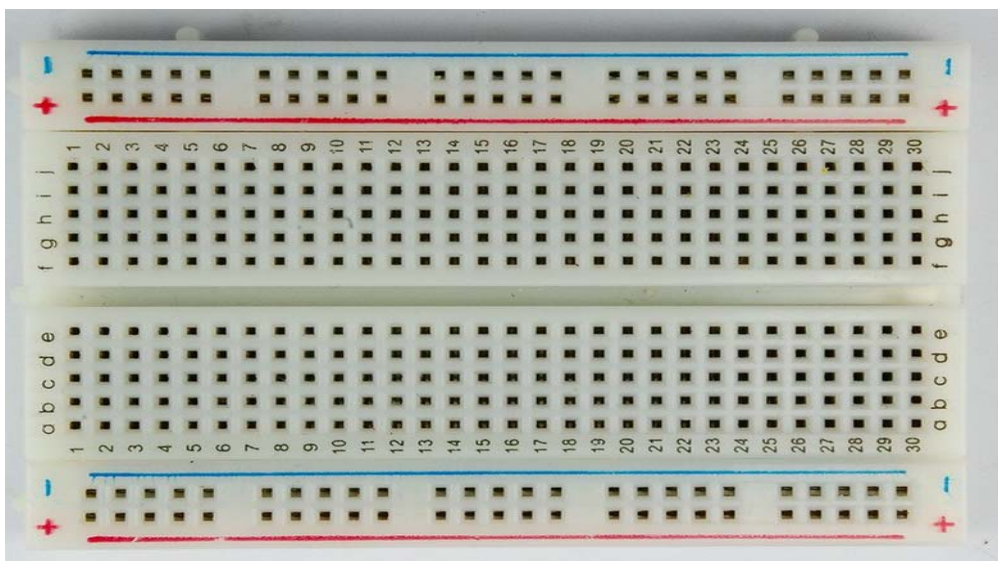
Breadboard

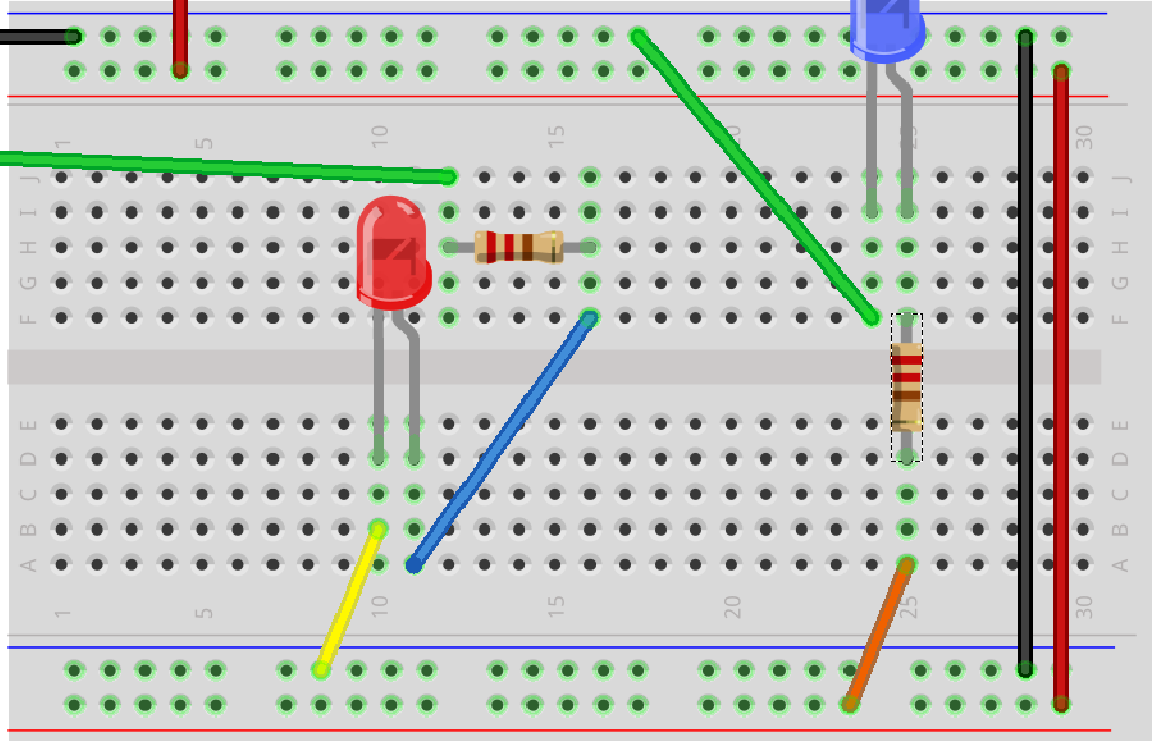
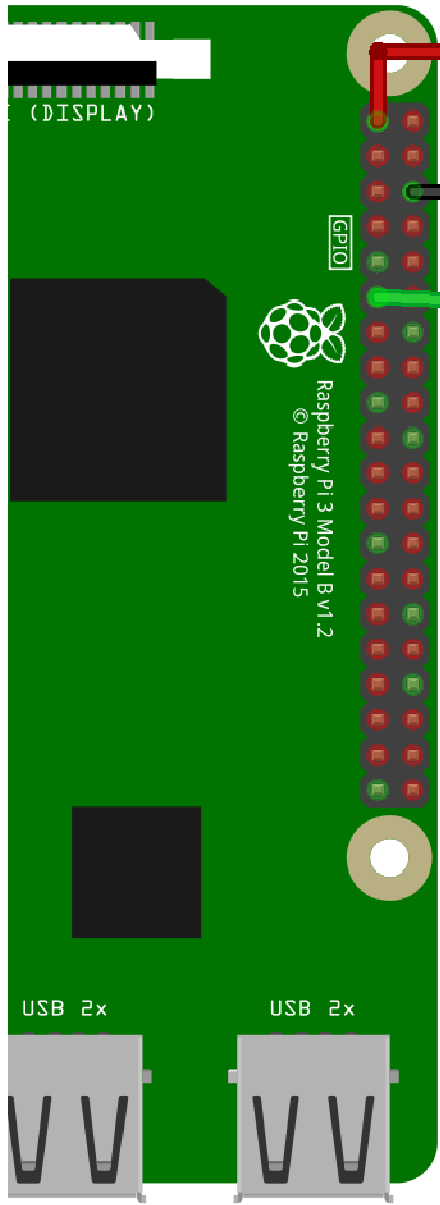


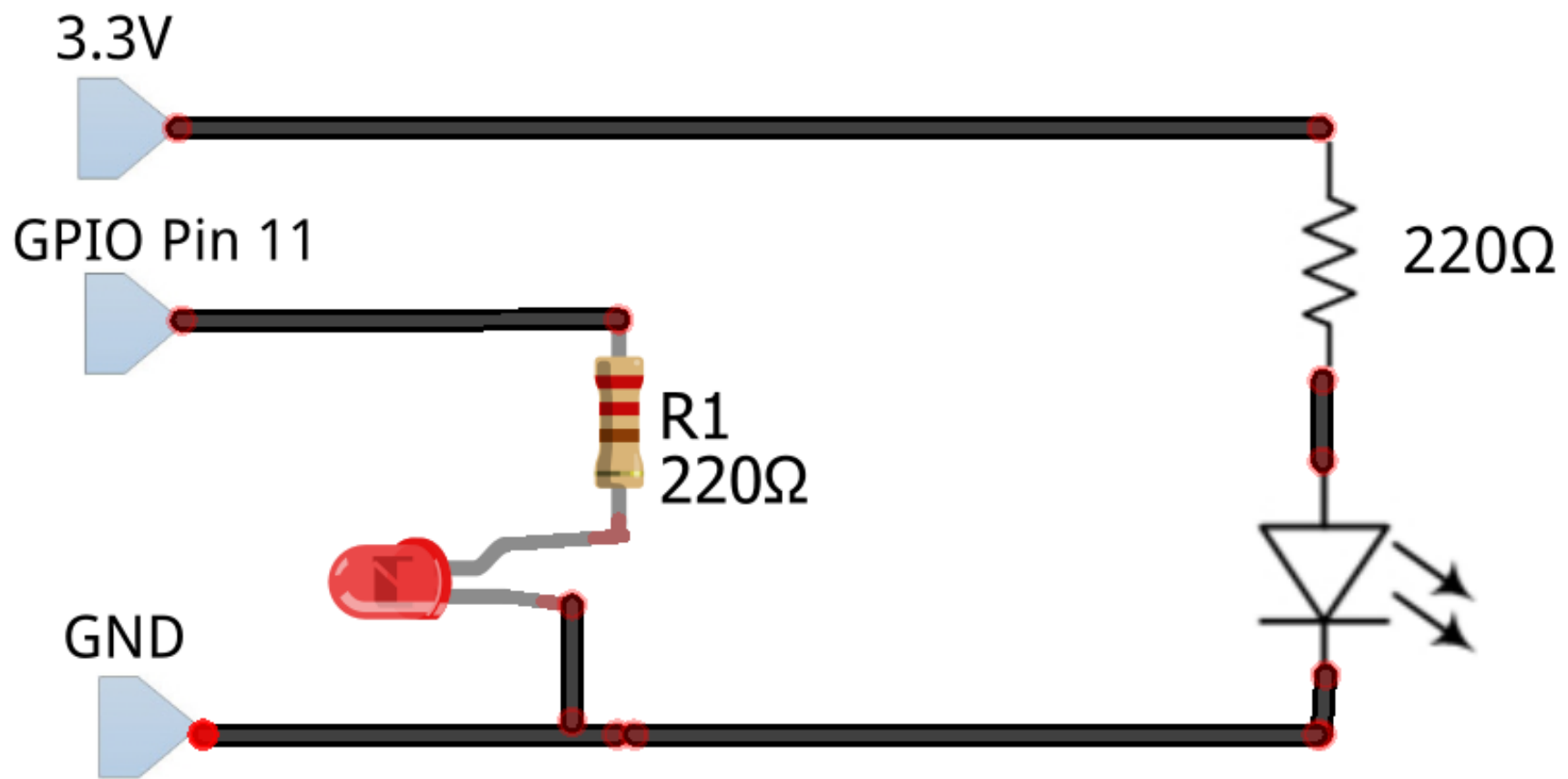
Breadboard



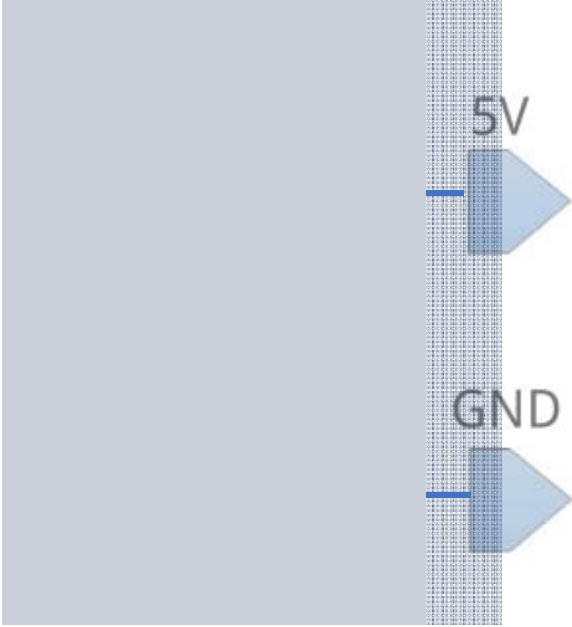




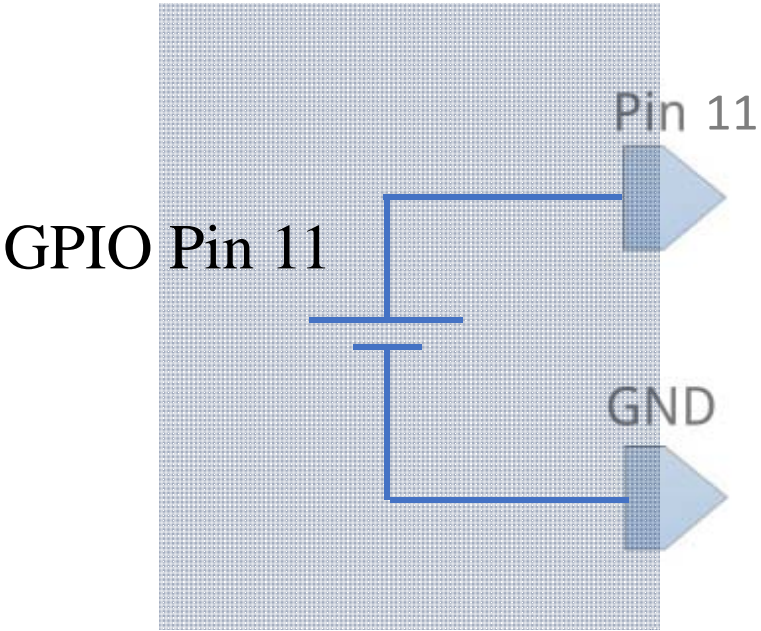




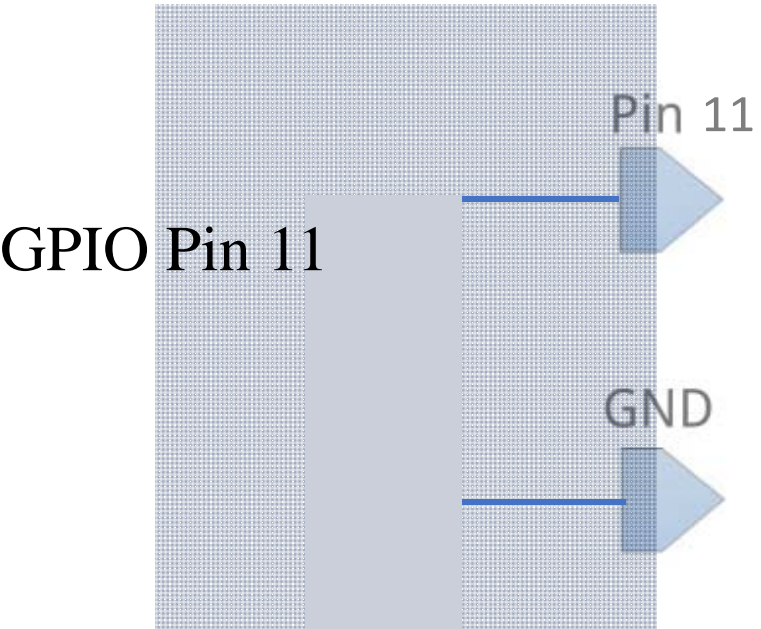
fritzing



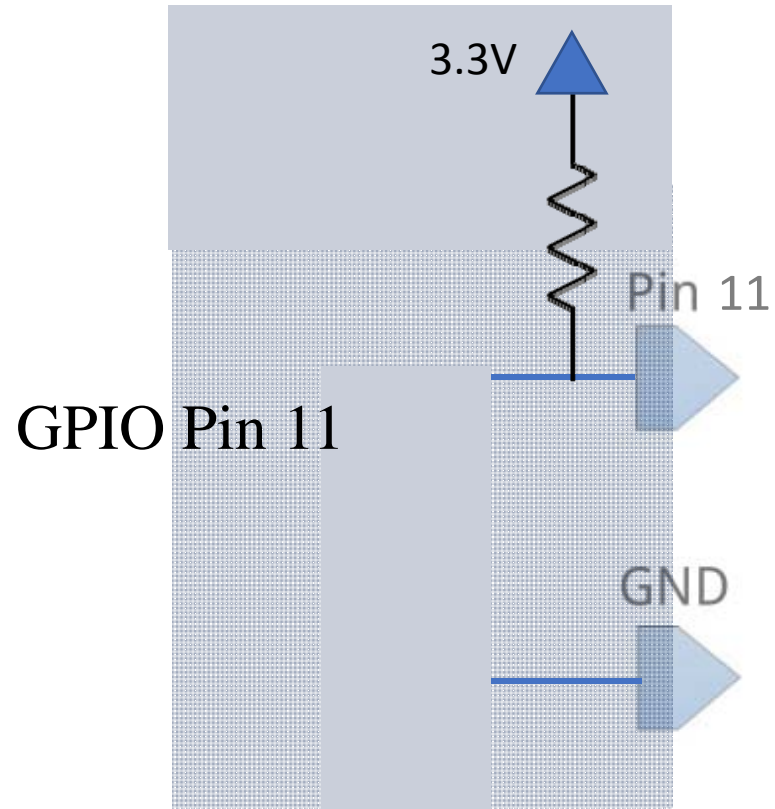
GPIO configured as OUTPUT

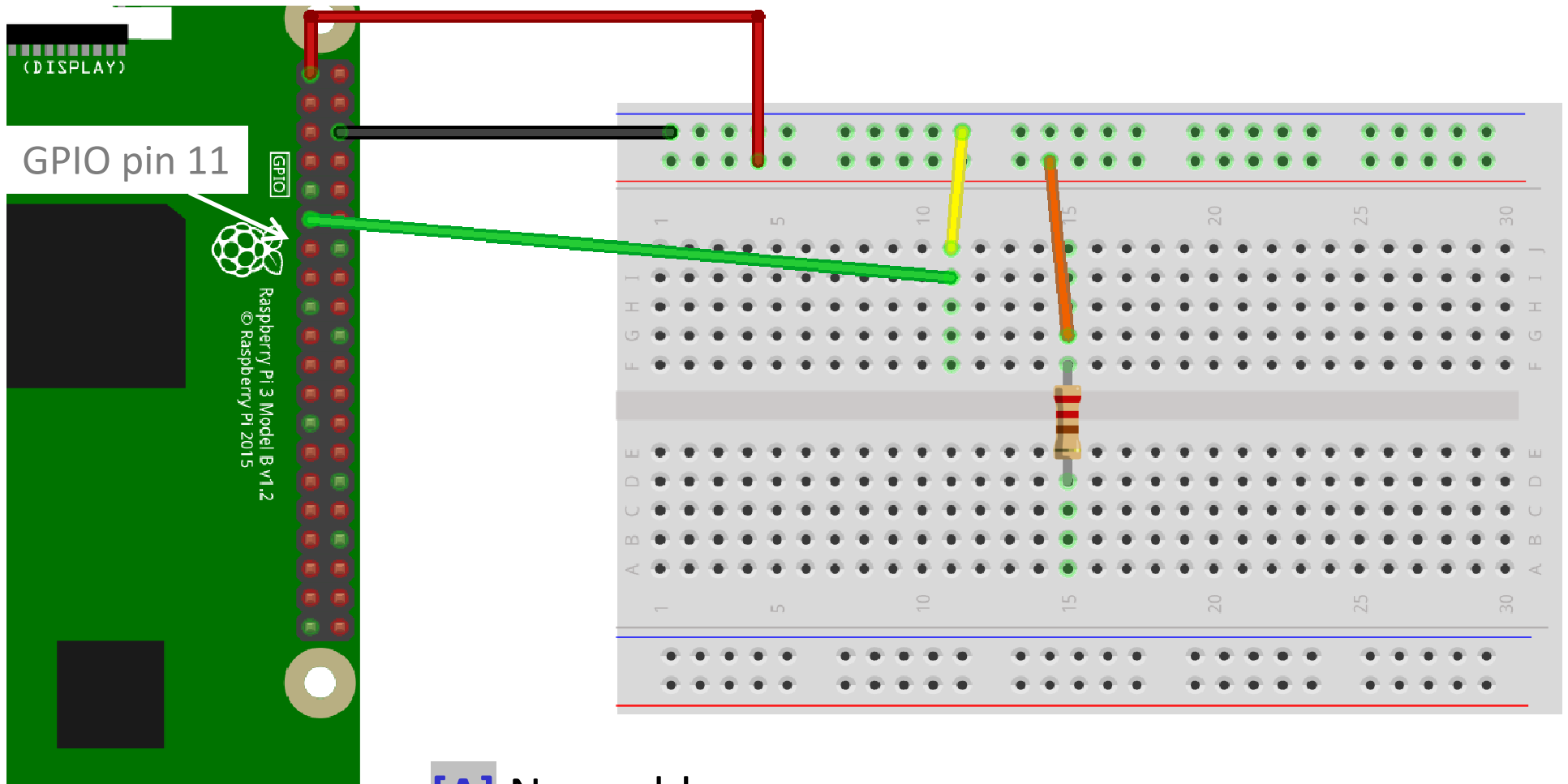


GPIO configured as INPUT



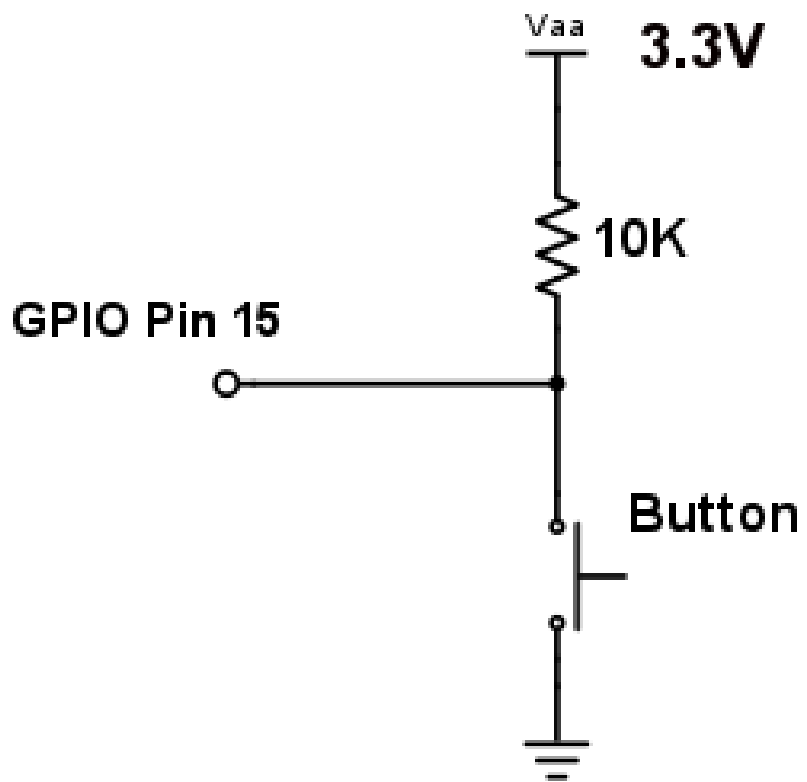
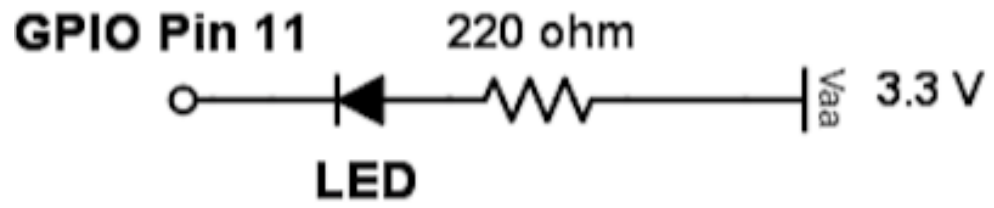
GPIO configured as INPUT with
internal PULLUP





The way the circuit is connected is:

- [A] No problem
- [B] A bad idea if Pin 11 is configured as an output
- [C] A bad idea if Pin 11 is configured as an input
- [D] A bad idea regardless of how Pin 11 is configured
- [E] I don't know



```
import RPi.GPIO as GPIO
import time
```

```
LedPin = 11
```

```
def setup():
```

```
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(LedPin, GPIO.OUT)
    GPIO.output(LedPin, GPIO.HIGH)
```

```
# pin11
```

```
# Numbers GPIOs by physical location
# Set LedPin's mode is output
# Set LedPin high (+3.3V) to turn off led
```

```
def loop():
```

```
    while True:
        GPIO.output(LedPin, GPIO.LOW) # led on
        time.sleep(0.5)
        GPIO.output(LedPin, GPIO.HIGH) # led off
        time.sleep(0.5)
```

```
def destroy():
```

```
    GPIO.output(LedPin, GPIO.HIGH) # led off
    GPIO.cleanup() # Release resource
```

```
if __name__ == '__main__':
```

```
    setup()
    try:
        loop()
    except KeyboardInterrupt:
        destroy()
# Program starts here
# When 'Ctrl+C' is pressed, execute destroy()
```

