Parking Sensor

For my final project, I decided to make a sample parking sensor using Raspberry pi. It will work as a parking sensor. The idea of this park sensor is to show green when you have plenty of room to pull your car forward in parking lot, and then turn red when you should stop. We're going to build this system with our Raspberry Pi, and use some distances that we can easily test.

Components used

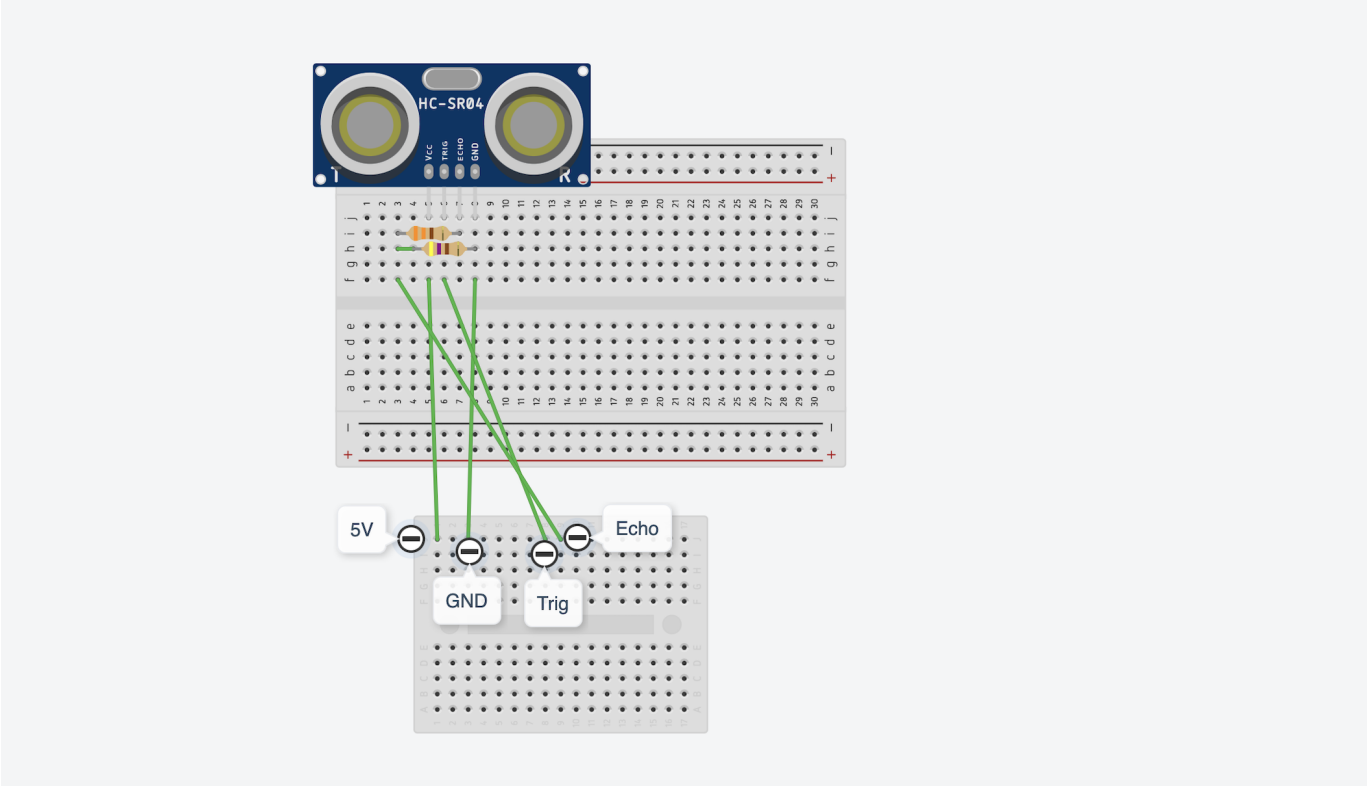
The following components will be used in this project other than Raspberry Pi setup.

1. HC-SR04 Ultrasonic Distance Sensor
2. Led (X2)
3. 330Ω Resistor (X4)
4. Male-Female Jumper Wires
5. Breadboard

Stage 1 - Ultrasonic Distance Sensor

1. Connect ulttrasonic distance sensor to breadboard
2. Trigger for the distance sensor is GPIO 24
3. Echo for distance sensor is GPIO 23 using resistor whose one side should be pinned in echo and other side on a breadbord with no sensor pin.
4. Another resistor should be pinned below GND and the pin below the resistor pin of echo

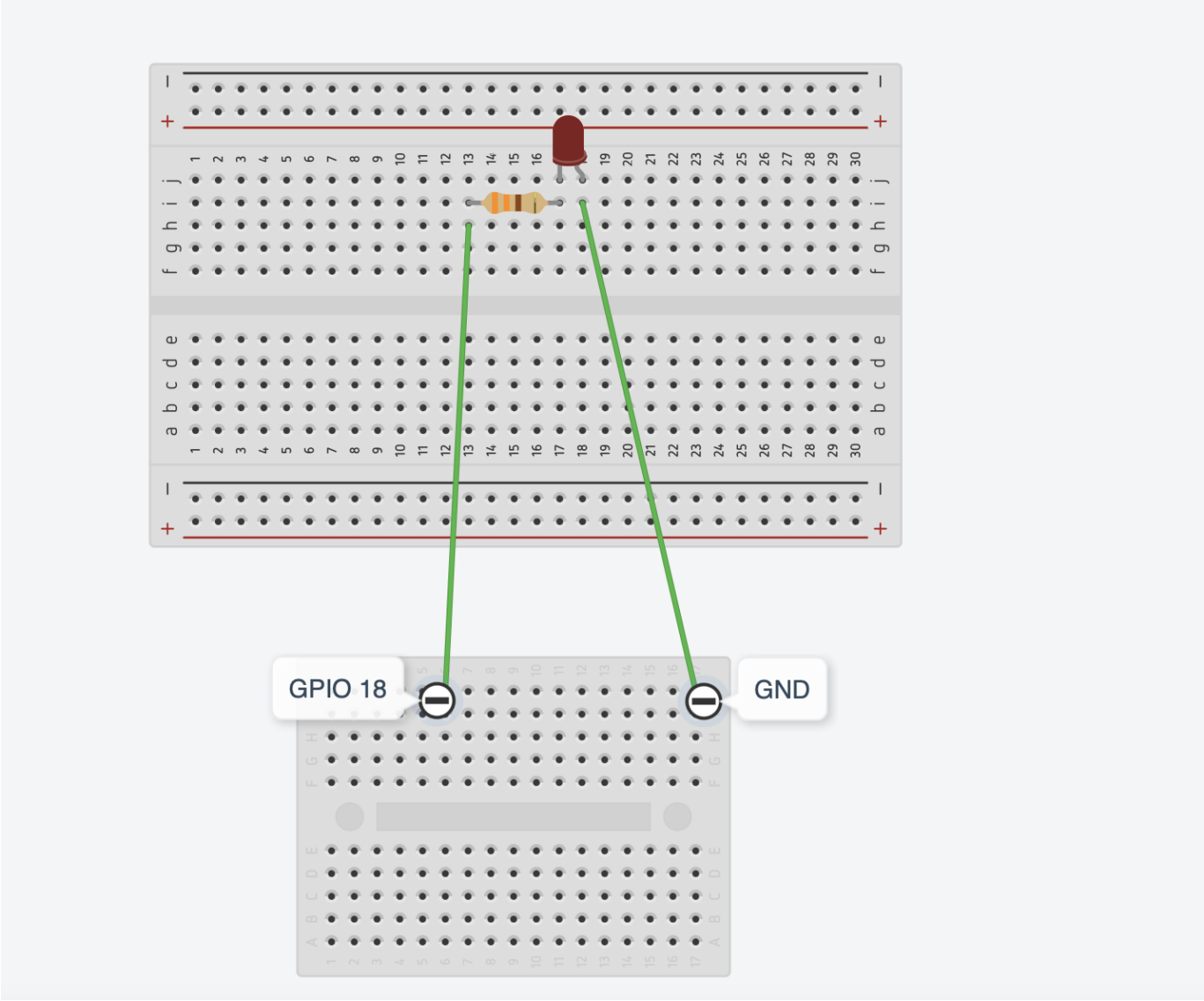
The figure below shows the circuit for ultrasonic distance sensor



Stage 2 - Red LED Light

1. Connect red LED light in the breadbord. Use resistor and cathode into GPIO 18 and anode to GND in Raspberry pi

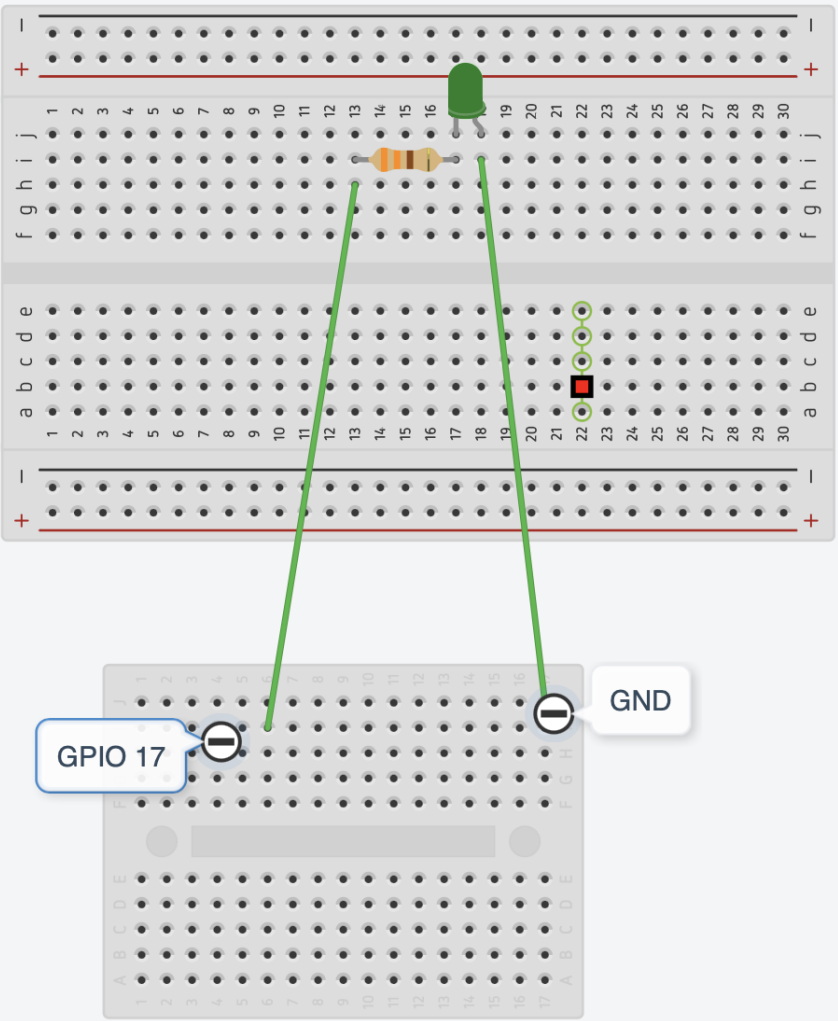
The figure below shows the connection for red LED light



Stage 3 - Green LED Light

1. Connect green LED light in the breadboard. Use resistor and connect cathode into GPIO 17 and anode to GND in Raspberry pi

The figure below shows the connection for green LED light



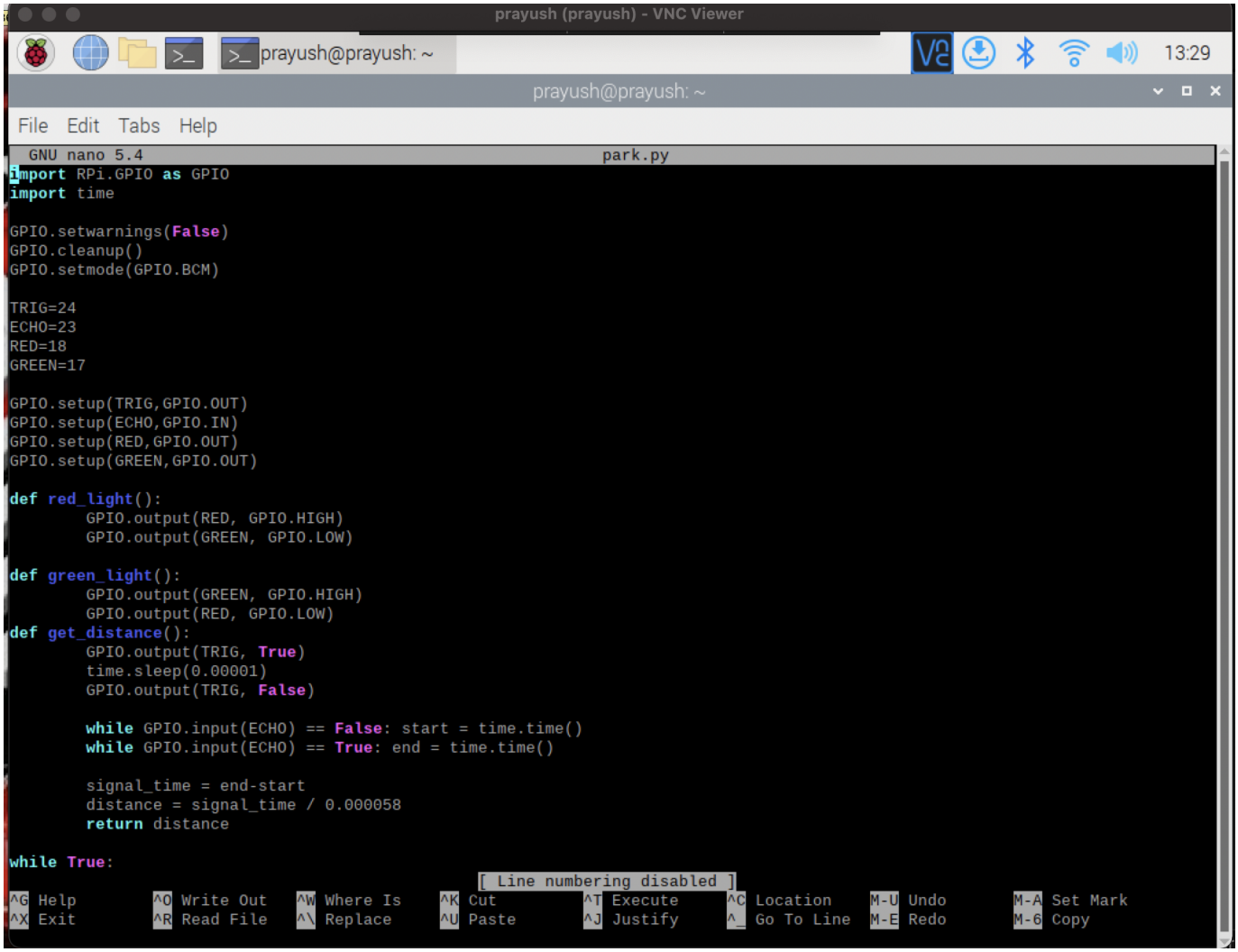
Final coding section

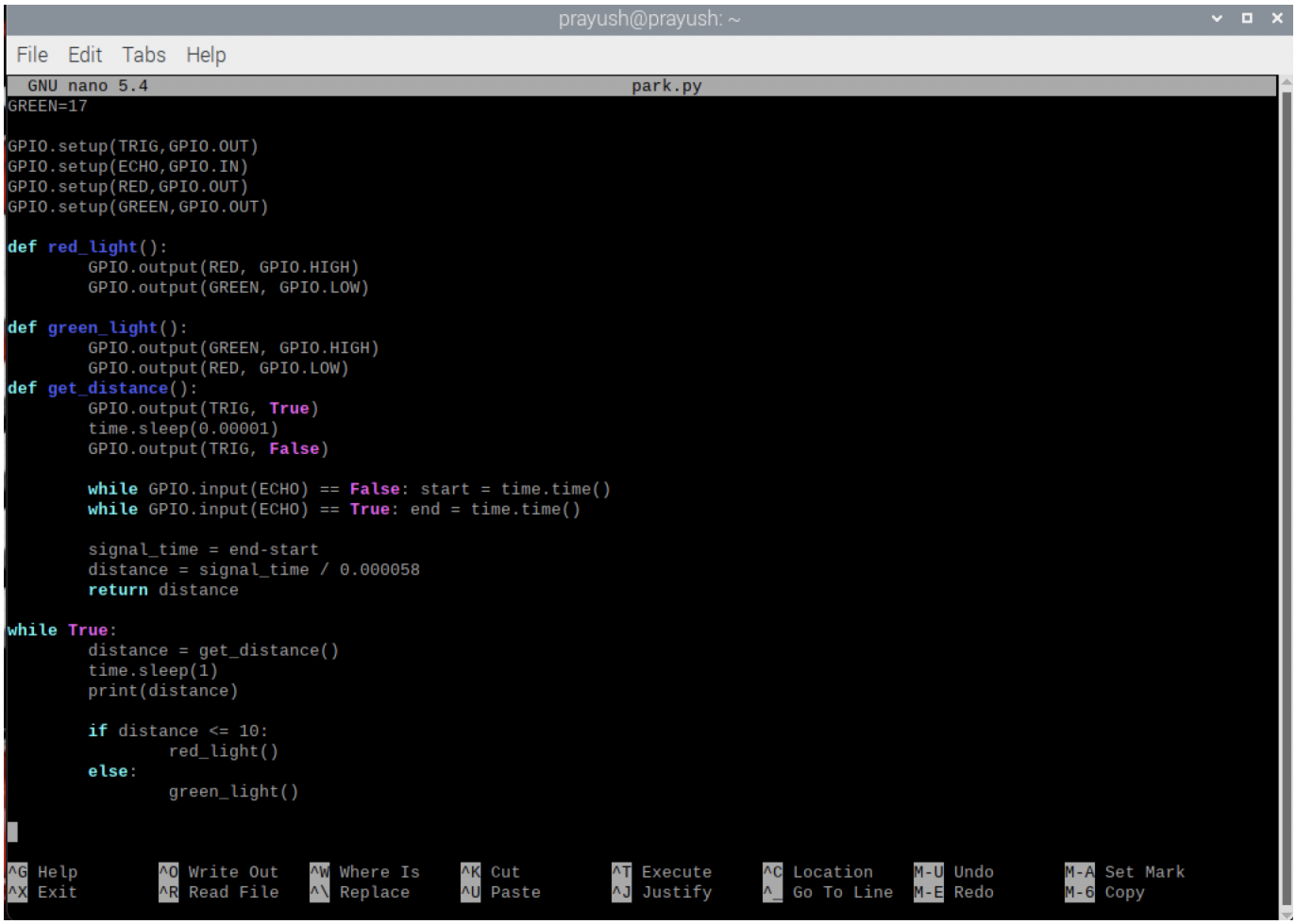
Open your Raspberry pi and inster the below code:

```
import RPi.GPIO as GPIO
import time

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

TRIG=24
ECHO=23
RED=18
GREEN=17

GPIO.setup(TRIG,GPIO.OUT)
```

```
GPIO.setup(ECHO,GPIO.IN)
GPIO.setup(RED,GPIO.OUT)
GPIO.setup(GREEN,GPIO.OUT)

def red_light():
        GPIO.output(RED, GPIO.HIGH)
        GPIO.output(GREEN, GPIO.LOW)
def green_light():
        GPIO.output(GREEN, GPIO.HIGH)
        GPIO.output(RED, GPIO.LOW)
def get_distance():
        GPIO.output(TRIG, True)
        time.sleep(0.00001)
        GPIO.output(TRIG, False)

        while GPIO.input(ECHO) == False: start = time.time()
        while GPIO.input(ECHO) == True: end = time.time()

        signal_time = end-start
        distance = signal_time / 0.000058
        return distance

while True:
        distance = get_distance()
        time.sleep(1)
        print(distance)

                if distance <= 10:
                red_light()
        else:
                green_light()
```

The image below shows the screenshot for the code





Output

If the object is too close to the sensor a solid red light flashes or the solid green light flashes.