LESSON B-13

ACTIVITY #3:
ADDING THE IR LINE SENSOR

In this activity, you will add the Infrared Line Sensor to the circuit from Activity #2. You will also modify the program from Activity #2 to add the ability to sense input from the line sensor.

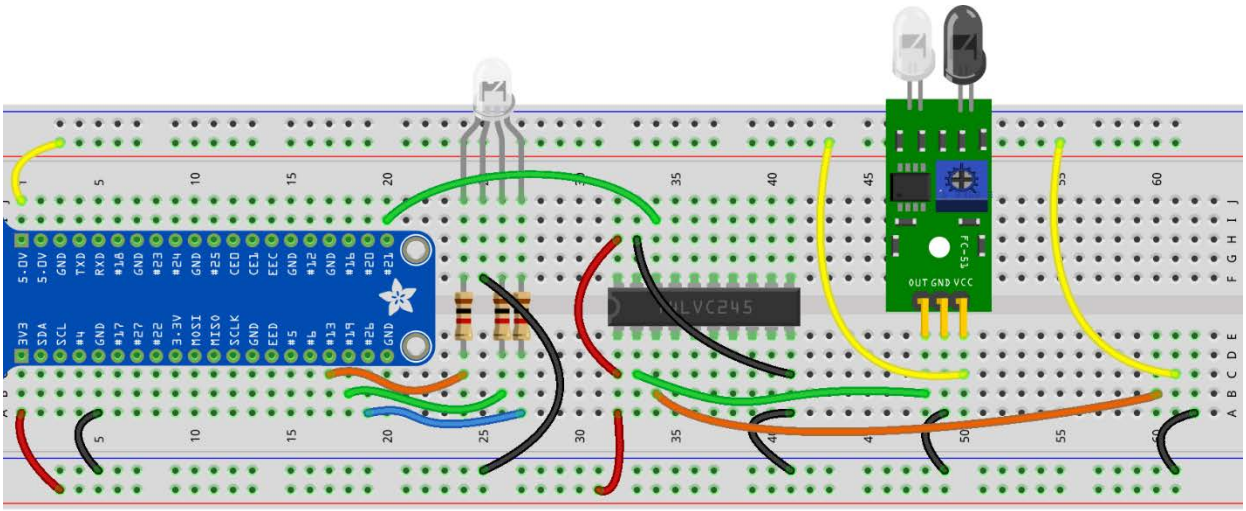
Step #1

With your Pi powered off, add the connections that will be required for the Infrared Line sensor. Add the following jumper wires to your circuit:

Ground – Short jumper wire from N1-61 to A62

5V – Long jumper wire from C61 to P2-55

Output – Long jumper wire from B60 to B34



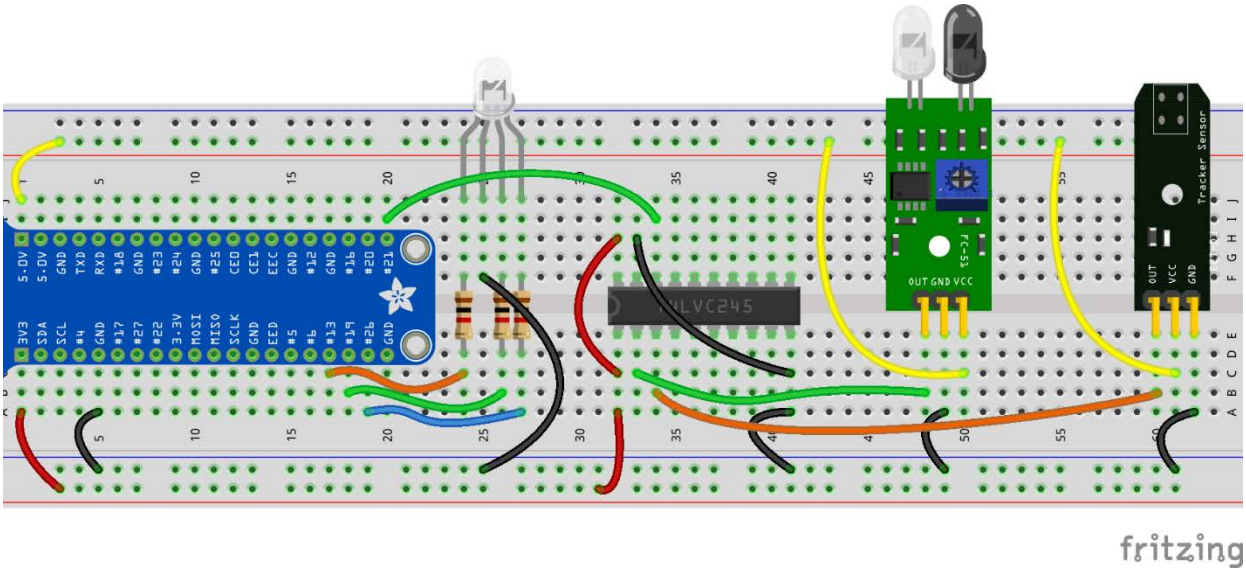
Step #2

The line sensor is now ready to be installed. In the last step you made connections to rows 60, 61, and 62 that will now be used for the sensor. Check the markings directly above the pins on the sensor, and insert the sensor into the breadboard in column E so that:

OUT is in E60

VCC is in E61

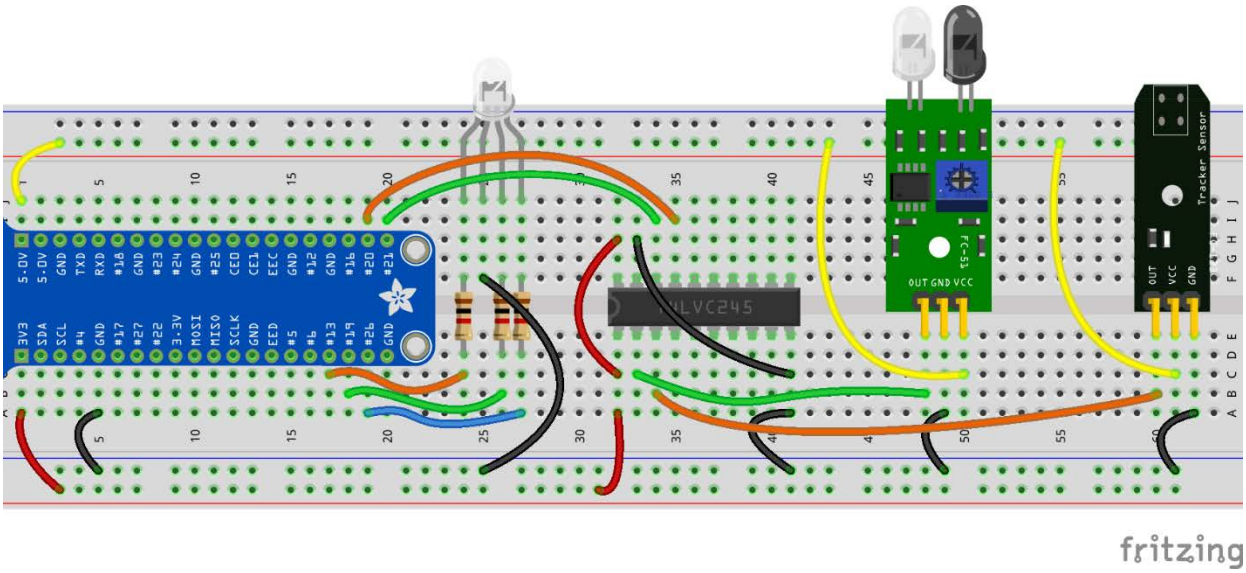
GND is in E62



This will mean that the LEDs on the line sensor are facing away from you when viewed with column A closest to you. This is the correct orientation to match the wiring above.

Step #3

The output of the IR Line Sensor is connected to channel 2 of the level shifter, but the output of channel 2 isn't connected to anything. Connect the output of channel 2 to **GPIO20**. Using a long jumper wire, connect I35 to I19:



Step #4

Power on your Raspberry Pi and open the program from Activity #2. You will make modifications to this program to add the input from **GPIO20** as well as program some additional LED behavior based on this input.

Setup **GPIO20** as an input just below the setup line for **GPIO21**. The additional line is highlighted below:

```
GPIO.setup(21, GPIO.IN)
GPIO.setup(20, GPIO.IN)


try:
```

Step #5

You will use the blue LED color to indicate when the line sensor does not see an infrared reflection. Sensing a reflective object will cause the output from the obstacle sensor to stay high. If this reflection is not present, like the lack of reflection caused by a black line, then the sensor output will be low. This low will flow through the level shifter and into **GPIO20**.

Add an **elif** statement below the **if** statement that will check to see if **GPIO20** is low or False. If so, that block will turn off the green and red elements and turn on the blue element. Turning off the blue element will also be added to the **if** block. The code additions are highlighted below:

```
try:
    while True:
        if GPIO.input(21) == False:
            GPIO.output(green, GPIO.LOW)
            GPIO.output(blue, GPIO.LOW)
            GPIO.output(red, GPIO.HIGH)
        elif GPIO.input(20) == False:
            GPIO.output(green, GPIO.LOW)
            GPIO.output(red, GPIO.LOW)
            GPIO.output(blue, GPIO.HIGH)
        else:
            GPIO.output(red, GPIO.LOW)
```

Step #6

Everything is looking good so far except that the **`else`** statement is not yet addressing the blue LED. If the blue LED gets turned on by the line sensor there is no code to turn it back off.

Add a line of code to the else statement that will turn off the blue LED. The addition is highlighted below:

```
        else:
            GPIO.output(red, GPIO.LOW)
            GPIO.output(blue, GPIO.LOW)
            GPIO.output(green, GPIO.HIGH)
        time.sleep(.1)
```

Here is the entire program with the line sensor additions:

```
import RPi.GPIO as GPIO
import time

rgb = [13,19,26]
red = 13
green = 19
blue = 26

GPIO.setmode(GPIO.BCM)
GPIO.setup(rgb, GPIO.OUT)
GPIO.setup(21, GPIO.IN)
GPIO.setup(20, GPIO.IN)

try:
    while True:
        if GPIO.input(21) == False:
            GPIO.output(green, GPIO.LOW)
            GPIO.output(blue, GPIO.LOW)
            GPIO.output(red, GPIO.HIGH)
        elif GPIO.input(20) == False:
            GPIO.output(green, GPIO.LOW)
            GPIO.output(red, GPIO.LOW)
            GPIO.output(blue, GPIO.HIGH)
        else:
            GPIO.output(red, GPIO.LOW)
            GPIO.output(blue, GPIO.LOW)
            GPIO.output(green, GPIO.HIGH)
        time.sleep(.1)

except KeyboardInterrupt:
    GPIO.cleanup()
```

Step #7

Run the program. Placing white paper near the line sensor will cause the RGB LED to turn green. Use a black marker to make a line at least ½ inch wide on the paper. Moving the paper across in front of the sensor will allow the sensor to indicate where the edges of the black line are located by turning the RGB LED blue. Verify that your obstacle sensor is still turning the LED red when an obstacle is in its path.

You can now see how obstacle and line sensors can help a robot, and its program, interact with the environment around it. Additional obstacle and line sensors can be added to further enhance this capability.

Leave the circuit assembled for use in Lesson B-14.