Build a Light Sensor

Using Raspberry Pi and Python

April 6, 2018

Raspberry Pi – Light Sensor

- Today we are going to turn our Raspberry Pi into a sensor that will turn on a light when the room gets too dark.

- We will be using a photo resistor and a Raspberry Pi to measure the time it takes to charge a capacitor. The Python code we run will report a number which is the time it takes for the resistor to fully charge the capacitor. The larger this number is the darker the room is. We will explore this more when we talk about the Python code.

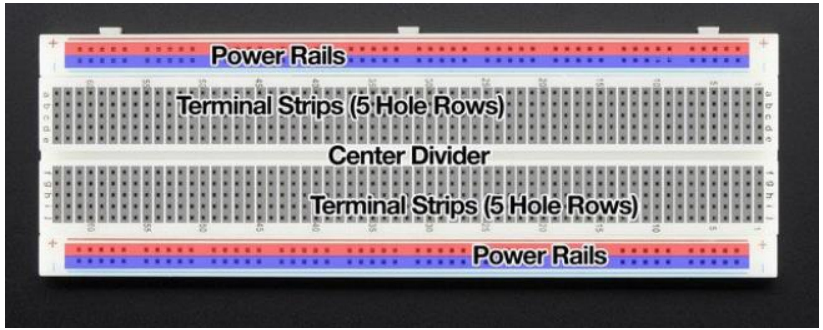
Our Equipment

- A Raspberry Pi and the GPIO pins
- 1 Breadboard
- 4 Male to Female DuPont Wires
- 1 jumper – to make the connection between the
- 1 1uF capacitor
- 1 10K resistor
- 1 LED Diode
- 1 LDR Sensor (Photo Resistor)

Circuit

- Circuit - Your circuit is the collection of all the connections you've made, using wires, resistors, LEDs, buttons, GPIO and other pins, etc. It can be closed and a signal is able to traverse from one end to the other or it can be open.
- An open circuit is like a long train of dominos, where you've removed 4 or 5 from the middle. You can try sending a signal from one end, but it's never going to bridge the gap.

The BreadBoard

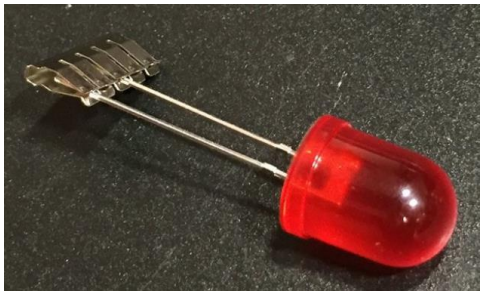


A breadboard is basically, a chunk of plastic with a bunch of holes. However, something special is going on inside. Although you can't see it there are many strips of metal that connect the rows and columns together.

If you look on the back of the breadboard, there's a yellow waxy paper covering some sticky foam. If you were to peel back that foam you'd see dozens of these metal rows.

There are two long rails on each side of the breadboard that provide power to the breadboard and parts installed on it and many terminal strips to connect parts to.

The inside of each terminal strip looks like this picture, these little teeth in the terminal strips grip onto electronic parts. When a part is pushed into the breadboard, the clip pushes open and grabs onto the metal leg. Any other part plugged into the other 4 teeth are thus electrically connected together.



The Raspberry Pi

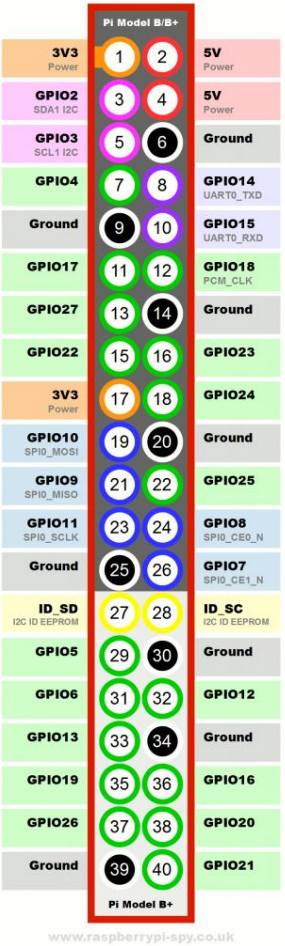
The Raspberry Pi is a mini computer we are going to use today.

The operating system is Raspian which is a flavor of Debian Linux that was optimized for the Raspberry Pi.

We will be using Python IDLE to run the code to read data from the sensors.

Raspberry Pi-Pinout Pins

Power – These pins provide power to the breadboard either 3.3 volts or 5 volts. We are going to use 3.3 volts today.
Ground – These pins complete the circuits we will be creating.
GPIO - General-purpose input/output pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output).

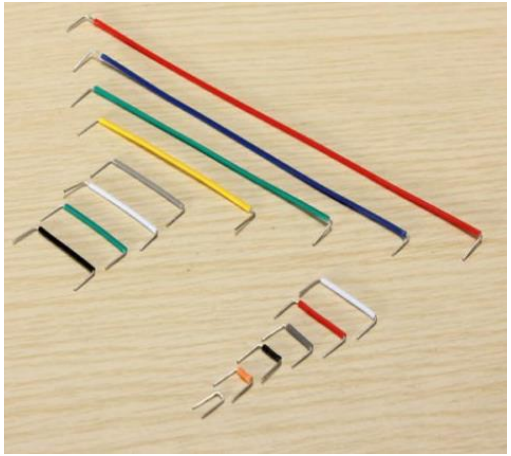


DuPont Wires

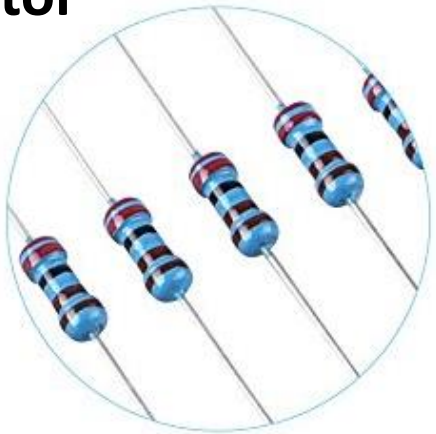
These wires will be used to connect the Raspberry Pi to the breadboard. They come in different colors to make it easier to distinguish its use. One end will connect to the Raspberry Pi and other to the breadboard.

Jumper

These little wires of various sizes are used to connect parts on the breadboard. We will use our jumper to connect the LDR sensor to power.



Resistor



A resistor represents a given amount of resistance in a circuit. In our case we use the resistor to stabilize the current flowing to the LED thus protecting it.

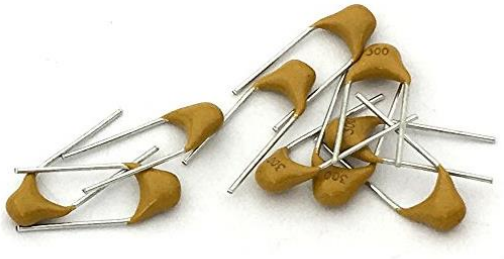
Resistance is a measure of how the flow of electric current is opposed or "resisted." It is defined by Ohm's law which says the resistance equals the voltage divided by the current.

Resistance = voltage/current or R = V/I

Resistance is measured in Ohms. The Ohm is often represented by the omega symbol: Ω.

Capacitor

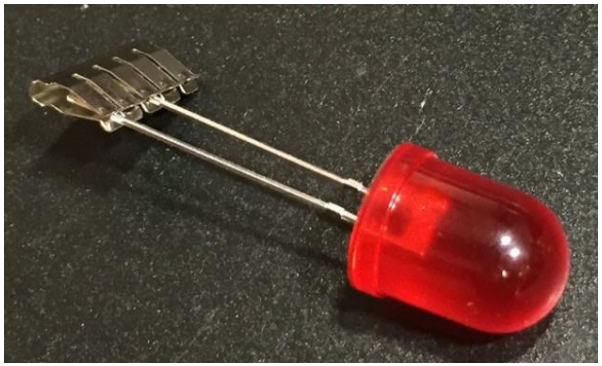
A **capacitor** is a device that stores electric charge. It consists of two conductive plates separated by an insulator or dielectric.



The **capacitor's capacitance** (C) is a measure of how much voltage (V) appears across the plates for a given charge (Q) stored in it: C = Q / V. Capacitance is measured in Farads.

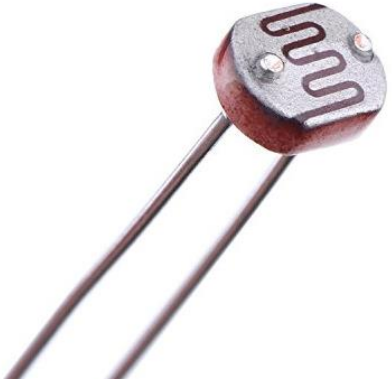
Since the Raspberry Pi does not have Analog pins we measure the time it takes for the capacitor to charge and send the GPIO pin high.

LED Diode



Light-emitting diode (LED) is a small, low-power light source used widely in electronics. We will turn on our LED when the LDR sensor detects the light in the room is low.

LDR Sensor



The light dependent resistor (LDR) sensor is the most important piece of equipment in our circuit. Without it we wouldn't able to detect whether it is dark or light. When the light is bright, the sensor will have a resistance of only a few hundred ohms while in the dark it can have a resistance of several megohms.

We need the capacitor in our circuit to measure the resistance of the LDR sensor. A capacitor essentially acts like a battery charging up while receiving power and then discharging when no longer receiving power. Using this in series with the LDR we can determine how much resistance the LDR has; turning on the LED based on light or darkness.

The Python Code

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)

pin_to_circuit = 7
pin_to_led = 16
LedTime = 200

try:
    # Main loop
    print('Starting')
    rcTime = 0

    while True:
        rcTime=rc_time(pin_to_circuit)
        print('Time = {}'.format(rcTime))
        if rcTime > LedTime:
            ToggleLed(pin_to_led,True)
        else:
            ToggleLed(pin_to_led,False)
        time.sleep(1)

except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```

This library is how our code will be able to access the pins
This library is for time functions

We are telling the code with this how we will reference the pins

This defines the pin locations on the Raspberry Pi for the photo sensor (7) and the LED (16)
LedTime is used to indicate when we should turn on and off the LED. The larger this number the darker the room will be before the LED turns on.

This is the main section of our code that calls the functions to check the photo sensor and turn on the LED light.

The rc_time function returns the count of tries it takes for the capacitor (yellow) to fill and the GPIO pin to be set to High. The larger this number is, the less light there is in the room.

We are then checking to see if the rcTime is greater than LedTime if it is, we are indicating this the code the light in the room has dimmed and we need to turn on our LED.

While True: - this code will run until we send the keyboard command to kill it. CTRL + C.

The Python Code

```
def rc_time (pin_to_circuit):
    count = 0

    #Output on the pin for
    GPIO.setup(pin_to_circuit, GPIO.OUT)
    GPIO.output(pin_to_circuit, GPIO.LOW)
    time.sleep(0.1)

    #Change the pin back to input
    GPIO.setup(pin_to_circuit, GPIO.IN)

    #Count until the pin goes high
    while (GPIO.input(pin_to_circuit) == GPIO.LOW):
        count += 1

    return count
```

This function first sets the pin for the photo sensor to low. The pin is then set to receive input from the sensor and the while loop counts the time it takes for the pin to change from low to high, When the pin changes to high the count is returned. The higher the count the less light there is.

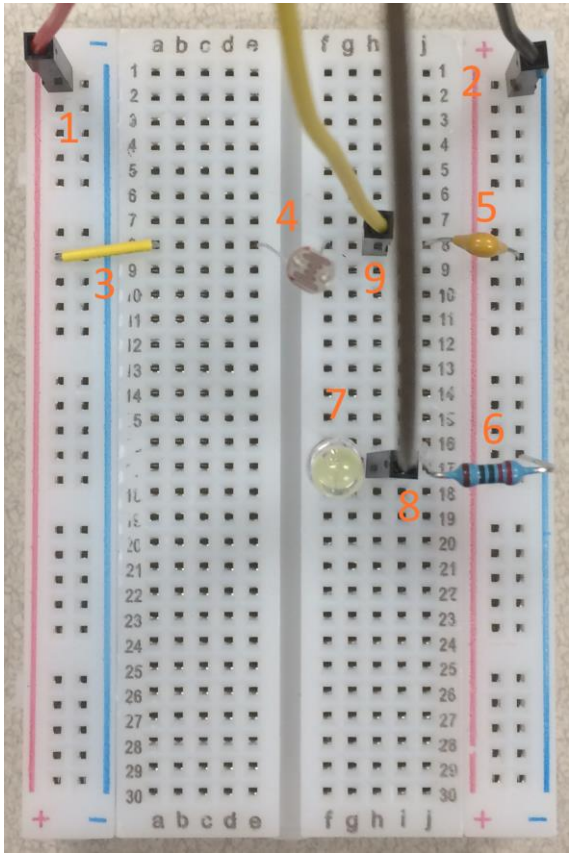
```
def ToggleLed (pin_to_led,toggle):

    GPIO.setup(pin_to_led,GPIO.OUT)
    if (toggle):
      print("LED on")
      GPIO.output(pin_to_led,GPIO.HIGH)
      time.sleep(1)
    else:
      print("LED off")
      GPIO.output(pin_to_led,GPIO.LOW)
```

This function turns the LED light on and off depending on the Boolean value (true or false) sent from the main section of code. We turn the LED on by setting the pin to high and off by setting the pin to low.

The code is set to turn on the pin when the count is LedTime You can play with this number in the main function to see how it affects when your light turns on and off.

Start Building – Time to try to build the circuit!



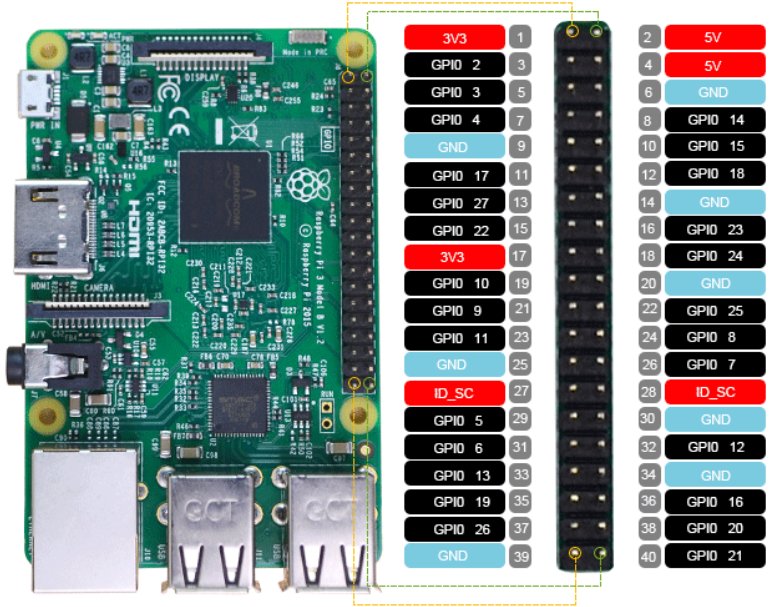
Very important Rule! Your Raspberry Pi MUST be powered off and the power not plugged in. If not you risk "bricking" your Raspberry Pi by causing electrical current to flow not as expected. Please let one of the instructors look at your finished circuit before you turn on your Raspberry Pi.

Steps.

1. Connect the Red (or Purple) DuPont wire to the top + (positive) line of your breadboard.
2. Connect the Black (or Blue) DuPont wire to the top opposite side – (negative) line of your breadboard.
3. Connect the small jumper to the + (positive) rail and to the "a" slot of line number 8.
4. Connect the LDR sensor across the center divider in row 8 slots "e" and "f".
5. Connect the 1uF capacitor to row 8 slot "j" and to the – (negative) rail.
6. Connect the resistor to the – (negative) rail and to row 16 slot "j".
7. Connect the LED in slot "f". Connect the negative wire to 16 and the positive wire to 17.
    1. Note: The LED has polarity meaning it has a positive and negative side like a battery does. You can tell which side is which by looking at the wires. The wire on the positive side is longer than that on the negative. The longer side should go in row 17 the shorter in row 16.
8. Connect the brown (or Gray) DuPont wire to slot "h" of in row 17.
9. Connect the Yellow (or White) DuPont wire to slot "h" of row 8.

Start Building – Connect to the Pi!

Very important! Your Raspberry Pi MUST STILL be powered off. The pins on the Raspberry Pi are ordered as represented in the chart. The instructions are telling you which pin (in gray) to connect to.



Steps.

1. Count the pins from the top of the Pi as seen in the picture to the left.
2. Connect the Red DuPont wire to pin 1 (Power). Top left pin
3. Connect the Yellow DuPont wire to pin 7 (GIPO 4). Fourth pin down on the left side.
4. Connect the Black DuPont wire to pin 6 (Ground). Third pin down on the right side.
5. Connect the Brown DuPont wire to pin 16 (GPIO 18). Eighth pin down on the right side.
6. Please wait for one of the instructors look at your finished circuit before you turn on your Raspberry Pi.
7. After the OK, turn on your Raspberry Pi.
8. Run the Python Code.