

**Lab 7**  
**CSC 2053 - Platform Based Computing**  
**Grading: 0 points**  
**Due Date: Oct 27th, 2017**

---

**Description:**

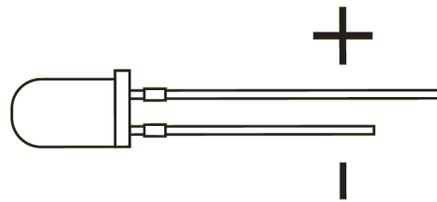
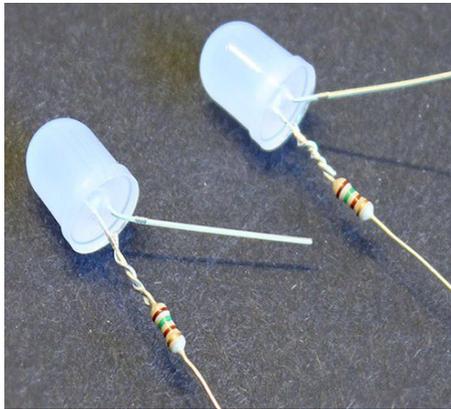
In this lab we will explore the capabilities of your Raspberry Pi. One of the very interesting functions of the Raspberry Pi is the GPIO pins. General-purpose input/output (GPIO) is a generic pin on an integrated circuit or computer board whose behavior-including whether it is an input or output pin-is controllable by the user at run time. GPIO pins have no predefined purpose, and go unused by default.

**Part 1** - Connect to your Raspberry Pi using SSH. You can use the command,

```
ssh -l pi youripaddress
```

to open up a remote shell on your pi.

Obtain an LED and resistor. Twist the resistor around the longer leg of the LED. This is the positive terminal.

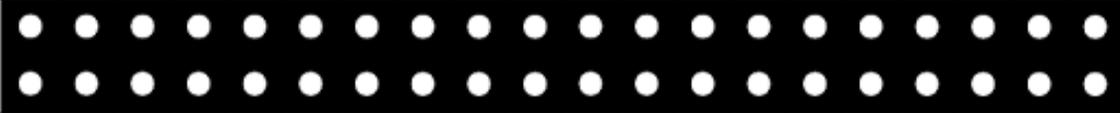


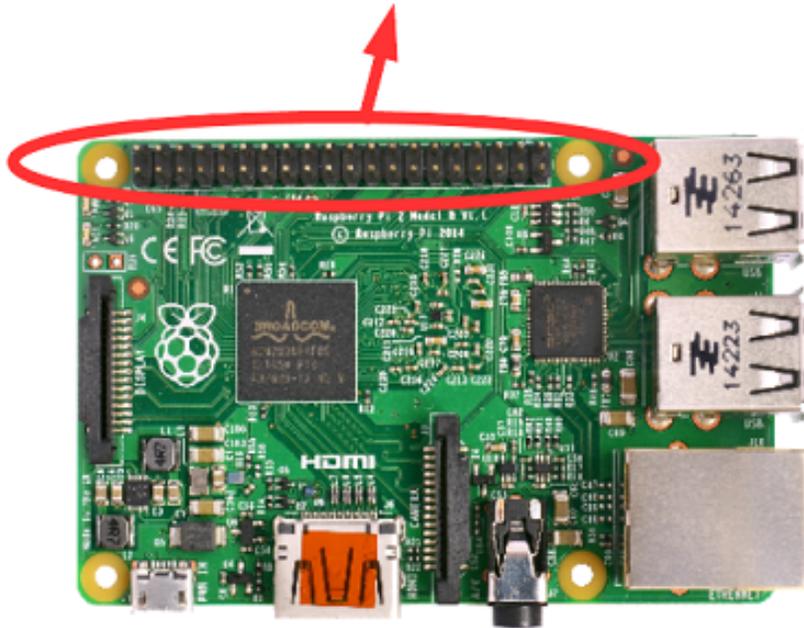
Create a file on your PI named blinker.py. The content of this file should look like the following.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(2, GPIO.OUT)
GPIO.setup(18, GPIO.IN, pull_up_down=GPIO.PUD_UP)
try:
    while 1:
        GPIO.output(2,not GPIO.input(18))
except KeyboardInterrupt:
    GPIO.cleanup()
```

Use the jumper cables provided to connect the LED to the pi. You will connect the positive side of the LEDs to pin 2, and the negative to a ground, pin 6. Notice in the above code, we set

BCM	5V	5V	GND	14	15	18	GND	23	24	GND	25	8	7	ID SC	GND	12	GND	16	20	21
BOARD	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40
																				
BOARD	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
BCM	3,3V	2	3	4	GND	17	27	22	3,3V	10	9	11	GND	ID SD	5	6	13	19	26	GND



the GPIO pin 2 to be an output pin. If we set the pin to output GPIO.HIGH, it will send 3.3V through pin 2 to the ground. If we set the pin to GPIO.LOW it will the volts to 0. In the above code, we are setting the output of pin 2 to be the opposite of input pin 18. We can set pin 18 to be a pull up resistor such that it registers HIGH when there is no complete circuit and LOW upon closing the circuit. I will demonstrate how this can be used as a switch to turn the light on and off in class.

You can run your program using the command,

```
python blinker.py
```

and exit your program using Ctrl-C

**Deliverables:** none.