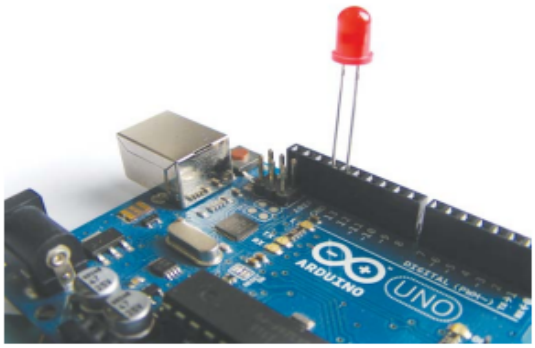
3.GPIO: Light the LED with Python with/without a button using either Uno/Raspberry Pi.

Remark: Signature:

Now that you've seen the hardware and software, let's begin our tour with the classic first Arduino project: blinking a light emitting diode (LED). Not only is this the simplest way to make sure that your Arduino is working correctly, but it will also introduce you to a simple sketch. As I mentioned earlier, a sketch is just a series of instructions that run on a computer. The Arduino can hold only one sketch at a time, so once you upload your sketch to your Arduino, that sketch will run every time the Arduino is switched on until you upload a new one. For this project we'll use the Blink example sketch that comes with the Arduino IDE. This program turns on an LED for 1 second and then off for 1 second, repeatedly. An LED emits light when a small current is passed through it. The LED will work only with current flowing in one direction, so the longer wire must connect to a positive power connection. LEDs also require a current limiting resistor; otherwise, they may burn out. There is a built-in resistor inline with pin 13 of the Arduino.

Follow these steps to set up your test:

1. Insert the long positive leg (also known as +5V or anode) of the LED into pin 13 on the Arduino, as shown in Figure 0-7. Connect the short negative leg (also known as cathode) to the GND pin next to pin 13.



2. Connect the Arduino to your computer with the USB cable.

3. Enter the following sketch into the IDE.

```
// Blinking LED Project

int led = 13;
void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

4. Click the **Verify** button (which looks like a check mark) to confirm that the sketch is working correctly.
5. Now click the **Upload** button to send the sketch to your Arduino.

Understanding the Sketch

Here's what's happening on each line of the sketch:

1. This is a comment. Any line in your program starting with // is meant to be read by the user only, and is ignored by the Arduino, so use this technique to enter notes and describe your code (called commenting your code). If a comment extends beyond one line, start the first line with /* and end the comment with */. Everything in between will be ignored by the Arduino

2. This gives pin 13 the name led. Every mention of led in the sketch refers to pin 13.

3. This means that the code between the curly brackets, { }, that follow this statement will run once when the program starts. The open curly bracket, {, begins the setup code.

4. This tells the Arduino that pin 13 is an output pin, indicating that we want to send power to the LED. The close curly bracket, }, ends the setup code.

5. This creates a loop. Everything between the curly brackets, { }, after the loop() statement will run once the Arduino is powered on and then repeat until it is powered off.

6. This tells the Arduino to set led (pin 13) to HIGH, which sends power to that pin. Think of it as switching the pin on. In this sketch, this turns on the LED.

7. This tells the Arduino to wait for 1 second. Time on the Arduino is measured in milliseconds, so 1 second = 1,000 milliseconds.

8. This tells the Arduino to set led (pin 13) to LOW, which removes power and switches off the pin. This turns off the LED.

9. Again the Arduino is told to wait for 1 second.

10. This closing curly bracket ends the loop. All code that comes after the initial setup must be enclosed within curly brackets. A common cause of errors in a sketch is missing open or close brackets, which will prevent your sketch from compiling correctly. After this curly bracket, the sketch goes back to the start of the loop at 3

Running this code should make your LED flash on and off. Now that you've tested your Arduino and understand how a sketch works and how to upload it, we'll take a look next at the components you'll need to carry out all of the projects in this book. Appendix A has more details about each component, what it looks like, and what it does.