City, University of London

Department of Electrical and Electronic Engineering

LAB - MANUAL

EE3600 / EE3700 Design - III

Blinking LED Design

T1 LAB - 1

1.0 Introduction

LEDs are small and powerful lights that are used in many different applications. To start with the design labs, we will work on a **Blinking LED Design**. It is as simple as turning a light "**ON**" and "**OFF**". This basic concept will help you to understand more complex design principles later in this module.

2.0 Required Components

- 1 x Arduino UNO
- 1 × Breadboard
- 2 × LEDs
- 2 x 330 Ohms Resistors
- Connecting wires

3.0 Components Description

3.1 Arduino UNO

In the introduction to Arduino lecture, we have focused on the properties of the Arduino software and hardware. We saw that Arduino boards are programmed using a language derived from C and C++ in Arduino's Integrated Development Environment (IDE). Arduino Uno is based on the *ATmega328* by Atmel. The Arduino Uno pinout consists of 14 digital pins, 6 analog inputs, a power jack, USB connection and ICSP header. The versatility of the pinout provides many different options such as driving motors, LEDs, reading sensors and others.

The Arduino UNO Pinout diagram in Fig 1.1 will hopefully help you to understand the board and its pin configurations:

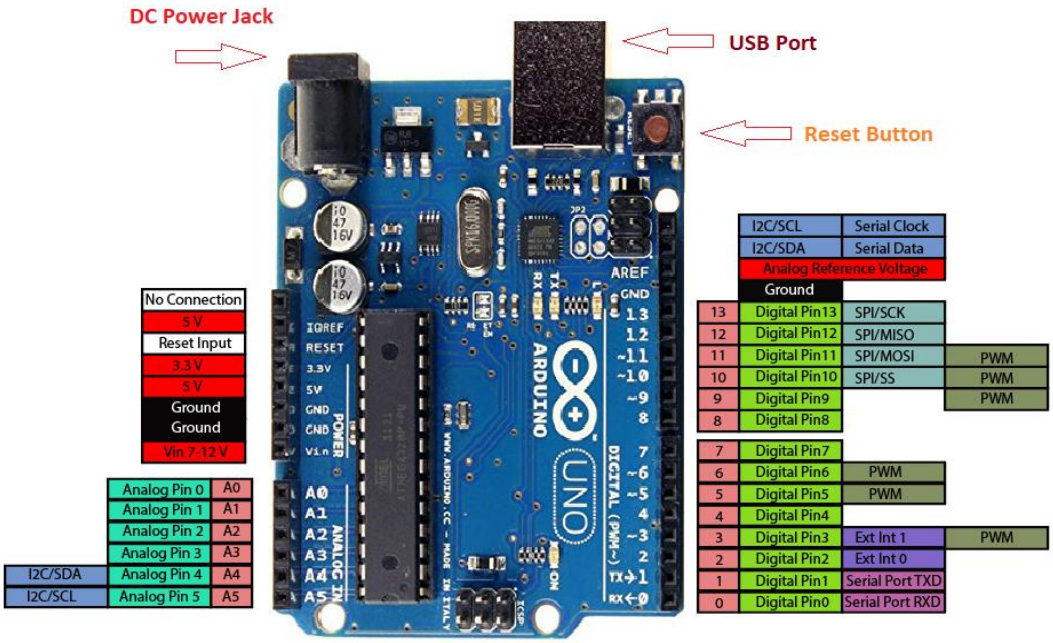

Fig 1.1 Arduino UNO Pinout diagram

3.2 LED

The LED is a specialised form of PN junction that uses a compound junction. The semiconductor material used for the junction must be a compound semiconductor. The commonly used semiconductor materials including silicon and germanium are simple elements and the junction made from these materials do not emit light. However, semiconductor materials such as gallium arsenide, gallium phosphide and indium phosphide are compound semiconductors and junctions made from these materials and do emit light.

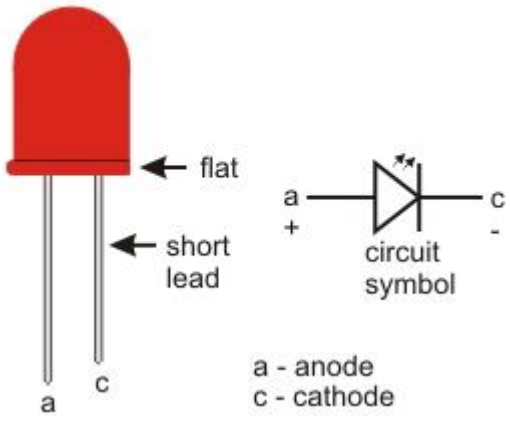

Fig 1.2 Schematic of an LED

The longer leg of LED represents the positive electrode or anode.

LEDs with more than 2 legs are also available such as 3, 4 and 6 pin configurations to obtain multiple colours in the same LED package. Surface mounted LED displays are available that can be mounted on the PCBs.

The current rating of LED is of few tens of milli-amps. Hence it is necessary to connect a high resistance in series to it. The forward voltage drop of an LED is much larger than an ordinary diode and is around 1.5 to 3.5 volts.

3.1.1 Applications of LEDs

- Electronic displays such as OLEDs, micro-LEDs, quantum dots etc.
- As an LED indicator.
- In remote controls.
- Lightings.
- Opto-isolators.

4.0 Circuit Design

First, connect pin 7 on your Arduino to any location on your breadboard and then onto the resistor. On the other side of the resistor, insert your LED as shown in Figure 1.3 below.
Connect the negative terminal of the LED to GND terminal of the Arduino board.

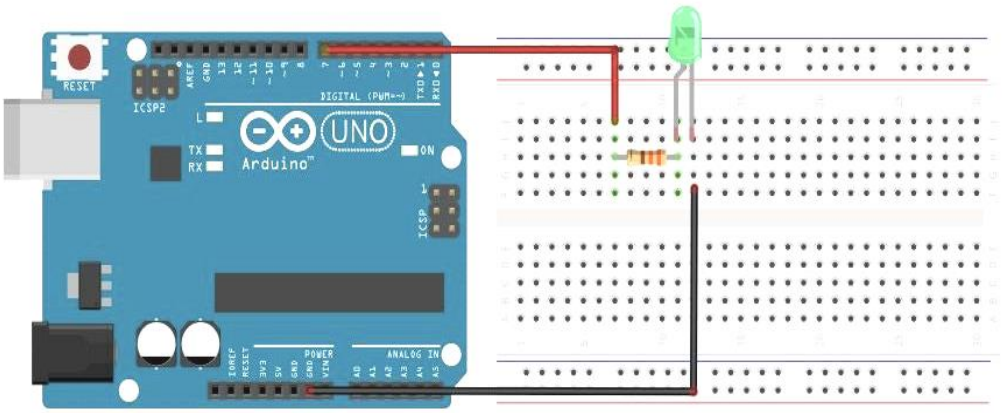

Figure1.3: Circuit diagram for blinking LED design

5.0 Arduino Code

In order to **blink the LED,** we have to programme the Arduino. This is where the real magic of Arduino happens. This little blue board can be programmed to do almost anything you can think-off. The code to achieve this functionality is quite simple!

To begin, you must set up your IDE for writing your code. Every Arduino program needs these two things for it to work. Start by writing the below:

```
void setup() {
}
void loop() {
}
```

When you reset your Arduino or boot it up, the code within the '**void setup**' section is executed. Once the execution is complete, '**void loop**' gets executed over and over until the power is removed from the Arduino. Before the '**void setup**', we must assign a name for **pin seven** for this design. So, we know what we are controlling on it. Therefore, we have to write before '**void setup'**:

```
int led = 7;
```

This assigns the name '**led**' to the integer **7**. Now every time we write '**led**' in our code, Arduino will interpret that as **7**. Within the '**void setup'**, we must write a line of code that will let the Arduino know that we want pin 7, or led, to act as an output. An output is a pin that is either **HIGH** or **LOW**, meaning it is either **ON** or **OFF**. But we don't need to worry about that yet. Just remember that an **OUTPUT** gives out electricity, and **INPUT** collects information from pins.

After void setup() {,
write:

```
pinMode(led, OUTPUT);
```

Important to remember that each line ends with a semicolon (i.e. ";").

Now for the actual controlling of the LED. After ***void loop()*** {,

write the following:

digitalWrite(led, HIGH);

This will set pin 7 *HIGH*, or *ON*, meaning that while it is *HIGH* it is outputting voltage. If you ran your code now, the LED would light up and wouldn't blink. However, the aim of this design was to make the LED blink.

On the next line, write:

delay(1000);

This will make the Arduino pause for one second. If you wanted it to be half a second, write 500 instead of 1000. You can choose this number according to the time delay required. But it still does not blink. On the line after that, write:

digitalWrite(led, LOW);

This turns your *LED OFF* after the period you specified. Run your code now. It still does not blink! Give it some thought and see if you can work it out yourself, keeping in mind what we learn about the *void loop function*.

Anyway, the reason is because it is looping as soon as it reaches the line that turns the LED OFF, going right back to where it turns ON. We do not have enough time to observe the LED being OFF. The solution for this problem is add another delay line after you turn the LED OFF.

Your final code should look like as follows:

```
int led = 7;
void setup() {
pinMode(led, OUTPUT);
}void loop() {
digitalWrite(led, HIGH);
delay(1000);
digitalWrite(led, LOW);
ddelay(1000);
}
```

6.0 Tasks to complete

1. Understand the overall concept of this design.
2. Familiarise yourself with the hardware setup.
3. Understand the given sketch for this design (Section 5.0) and test this sketch with your designed hardware.
4. Add another LED to the existing circuit and modify the connections to Arduino for these LEDs. Also modify the sketch to blink both LEDs in a sequence.
5. If you have 2 or more LED's connected to one circuit and set these LED's to blink in a sequence, do you need the second time delay that we have introduced earlier in the program?
6. Suggestions for where we can implement this design / idea.
7. Be innovative in your design to score additional marks in the assessment.

7.0 Marking Scheme

Description	Marks
Total marks for this project	**10**
Hardware Design	**04**
Oral (Individual assessment!)	**06**

8.0 Sample MCQs for quiz

8.1) How many **Arduino libraries** we used in our LAB-1 LED blinking design?

1) One 2) Three 3) None 4) Four

8.2) What is the main structure of the Arduino program (IDE)?

1) F
1) *or* and *while*
2) *if* and *void setup*
3) *void setup* and *void loop*
4) *elseif* and void loop

8.3) How many Pulse With Modulation (PWM) pins are in the Arduino UNO board?

1) Twenty-four
2) Nine
3) Six
4) Four

8.4) What is defined by ***digitalWrite (led, HIGH);***

1) Pin ON
2) Pin OFF
3) Pin position not defined
4) Pin position ON / OFF

8.5) What is the recommended input voltage for **Arduino UNO**?

1) 7V- 12V
2) 3.5V – 12V
3) 10V – 12V
4) None of the above

End of Lab - 1