



## American International University- Bangladesh

Department of Electrical and Electronic Engineering

EEE 4103: Microprocessor and Embedded Systems Laboratory

---

### Lab Report Cover Sheet

Course Name	Microprocessor and Embedded Systems
Lab Report No.	02
Lecturer Name	NADIA ANAM
Semester	Spring, 2021-22
Submission Date	16/02/2022
Section	F
Uploaded Date	15/02/2022

**Experiment 02:** Familiarization with microcontroller, study of blink test using and implementation of a traffic control system using microcontrollers.

### Submitted by

Nabil, MD. Jobair Ahmad  
18-38837-3

### Group Members

	Name	ID	Department
01.	AKTER, ALAM MORIUM	19-39957-1	CSE
02.	MOBARAK, BHUIYAN SAADBIN	19-41059-2	EEE
03.	NABIL, MD. JOBAIR AHMAD	18-38837-1	CSE
04.	SAKIB, A.B.M. NAZMUS	19-41582-3	EEE
05.	BHUIYAN, FAHIM MAHMUD	20-42970-1	CSE

**Title:** Familiarization with microcontroller, study of blink test using and implementation of a traffic control system using microcontrollers.

### Introduction:

The goal of this project is to become acquainted with the Microcontroller.

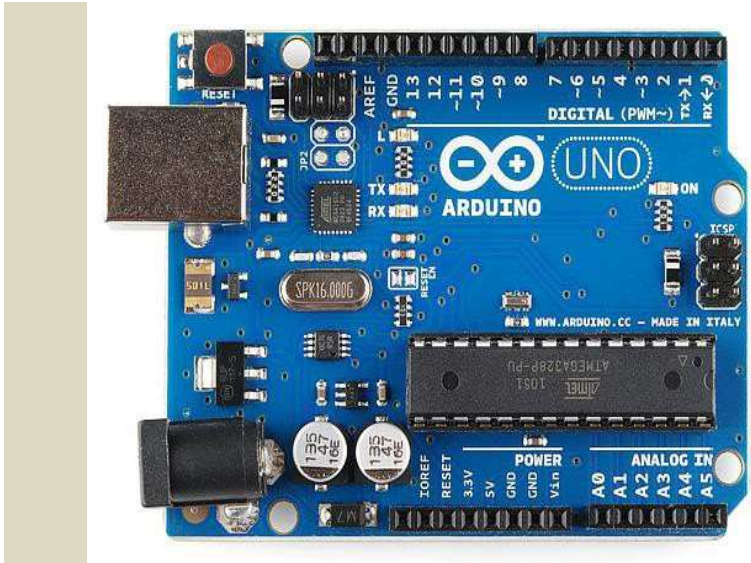
- Learning how to use Arduino and the delay functions to make an LED blink
- Arduino-based traffic control system implementation.

### Theory and Methodology:

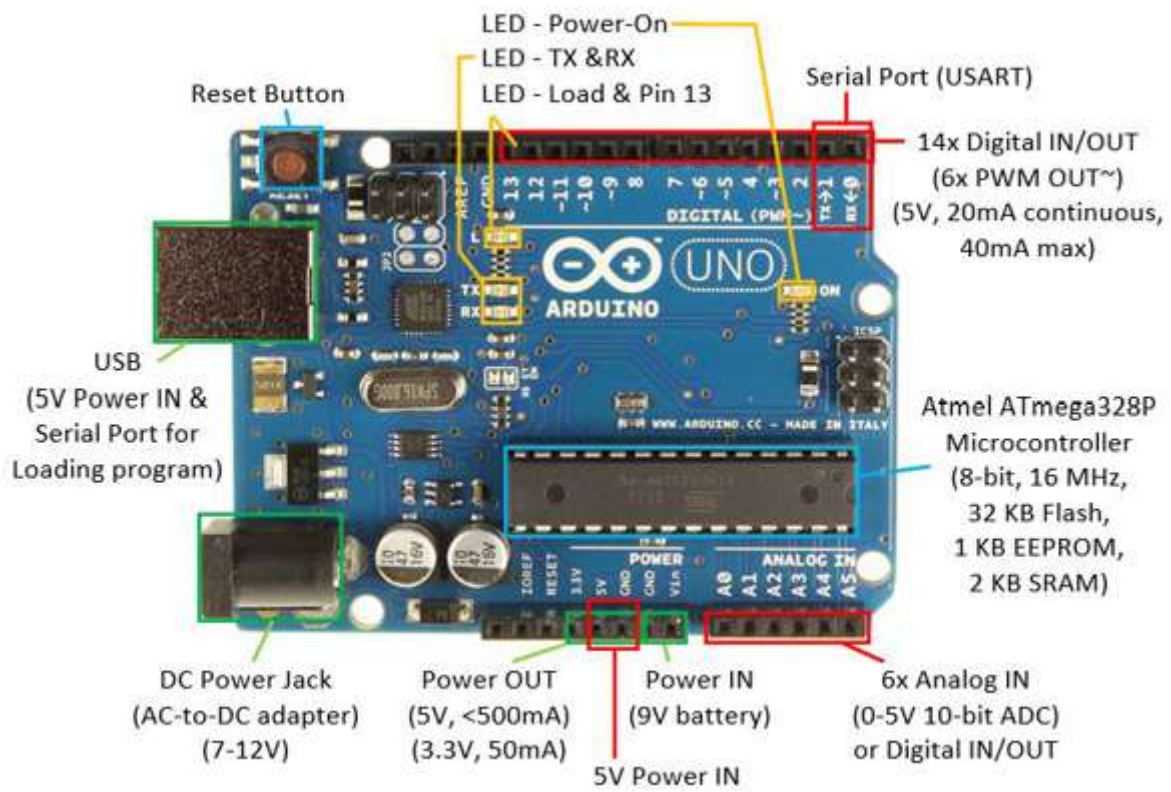
Arduino is a free and open-source platform for building interactive electronics projects. Arduino is made up of a programmable microcontroller as well as software, known as an IDE (Integrated Development Environment), that runs on your computer and is used to create and upload computer code to the microcontroller board. A hardware circuit (programmer/burner) is also not required to load fresh code into the Arduino Uno. We can simply load code into the board using a USB connection and the Arduino IDE (which utilizes a simplified version of C++ to write code).

### Arduino Family:

Arduino makes several different boards, each with different capabilities. In addition, part of being open-source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality.

Figures of different types of Arduino boards	Configuration & Features
	<p><b><u>Arduino Uno (R3)</u></b></p> <p>The Uno is an excellent choice for your first Arduino project.</p> <ul style="list-style-type: none"><li>• It has everything you need to get started as well as nothing you don't.</li><li>• It contains 14 digital I/O pins (six of which may be used as PWM outputs), 6 analog inputs, a USB connection, a power connector, a reset button, and other features.</li><li>• It comes with everything you need to support the microcontroller; simply connect it to a computer through USB or power it using an AC-to-DC converter or battery to get started.</li></ul>

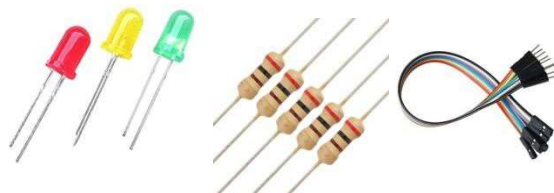
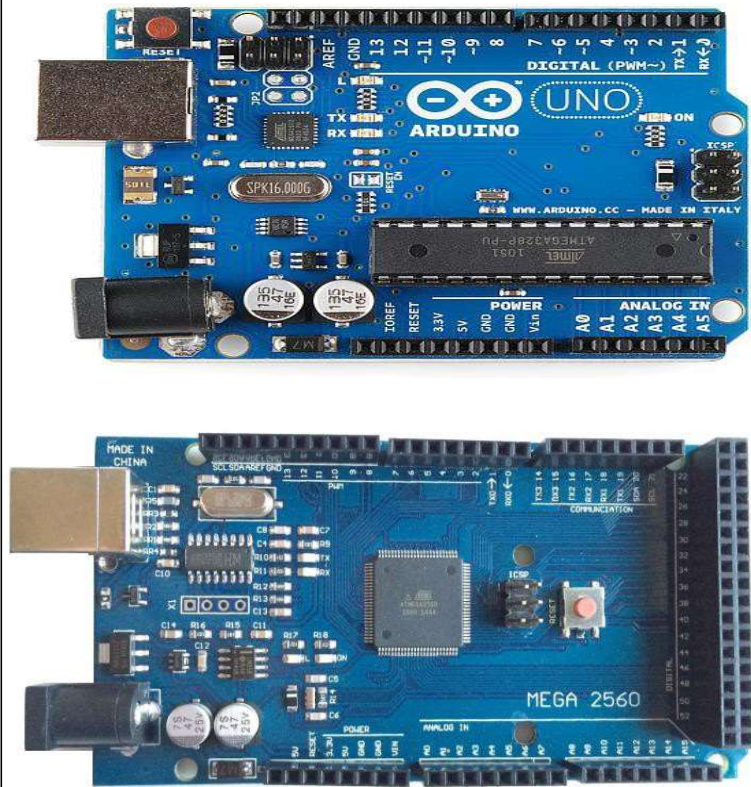
## Overview of the board (Arduino Uno R3):



1. DC Power Jack and USB connector
2. Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)
3. Reset Button
4. Power LED indicator
5. TX RX LEDs
6. Main IC
7. Voltage Regulator

## Apparatus:

1) Arduino IDE (any version)	Software
2) Arduino Uno (R3) board or Arduino mega 2560 board	
3) LED lights (RED, GREEN and YELLOW) and three 200 ohms resistors and jumper wires	



## Familiarization with the Arduino Commands

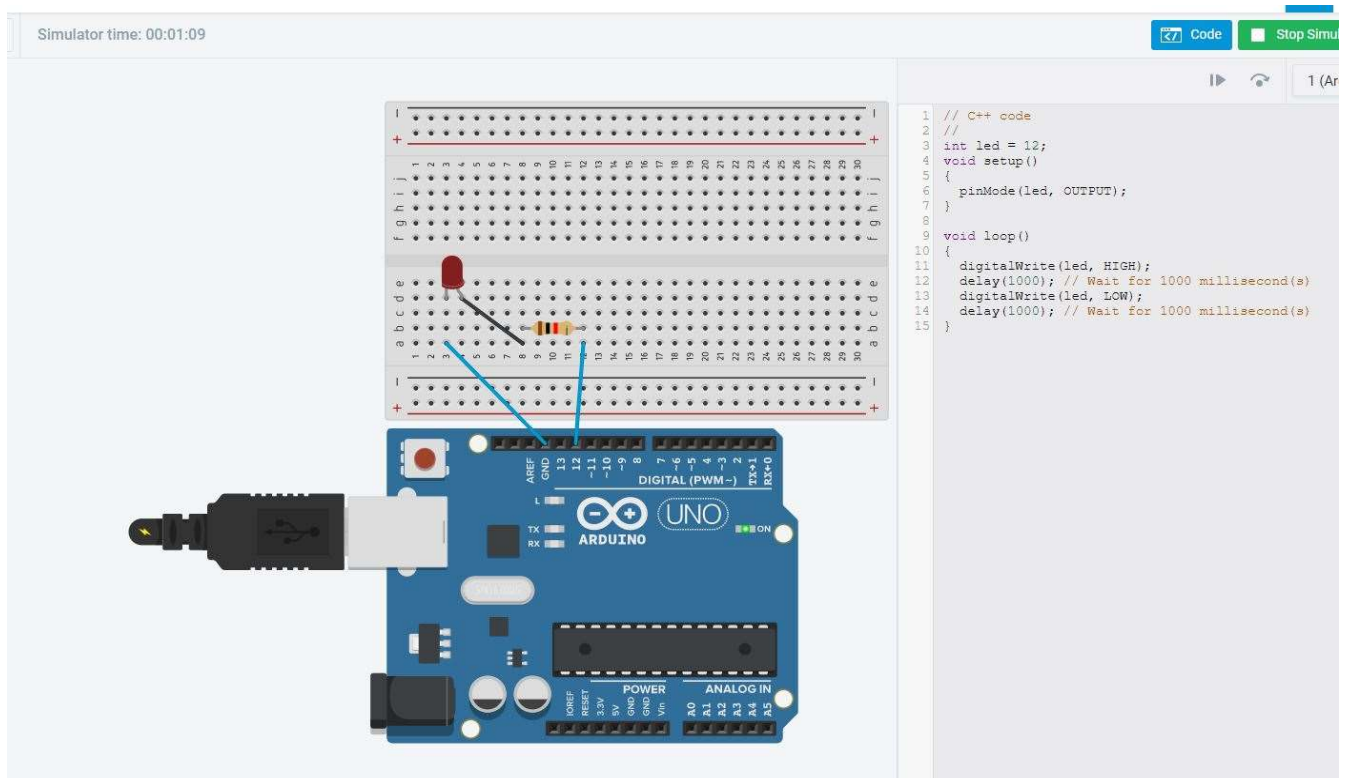
In this section,

1. We will learn about some common Arduino commands that will help write code.
2. This section also focuses on the standard Library functions associated with the IDE.
  - a) \*\*\*\*pinMode(X, INPUT) or pinMode(X,OUTPUT) \*\*\*\*  
this command will configure any pin at the Arduino board as either input/output.
  - b) \*\*\*\*digitalWrite(X, LOW) or digitalWrite(X, HIGH) \*\*\*\*  
this command will provide a HIGH/LOW value to any digital output pin at the Arduino board .

**Setting up the Circuit** The main task of our lab is to understand and implement a traffic control system after understanding to blink a LED light.

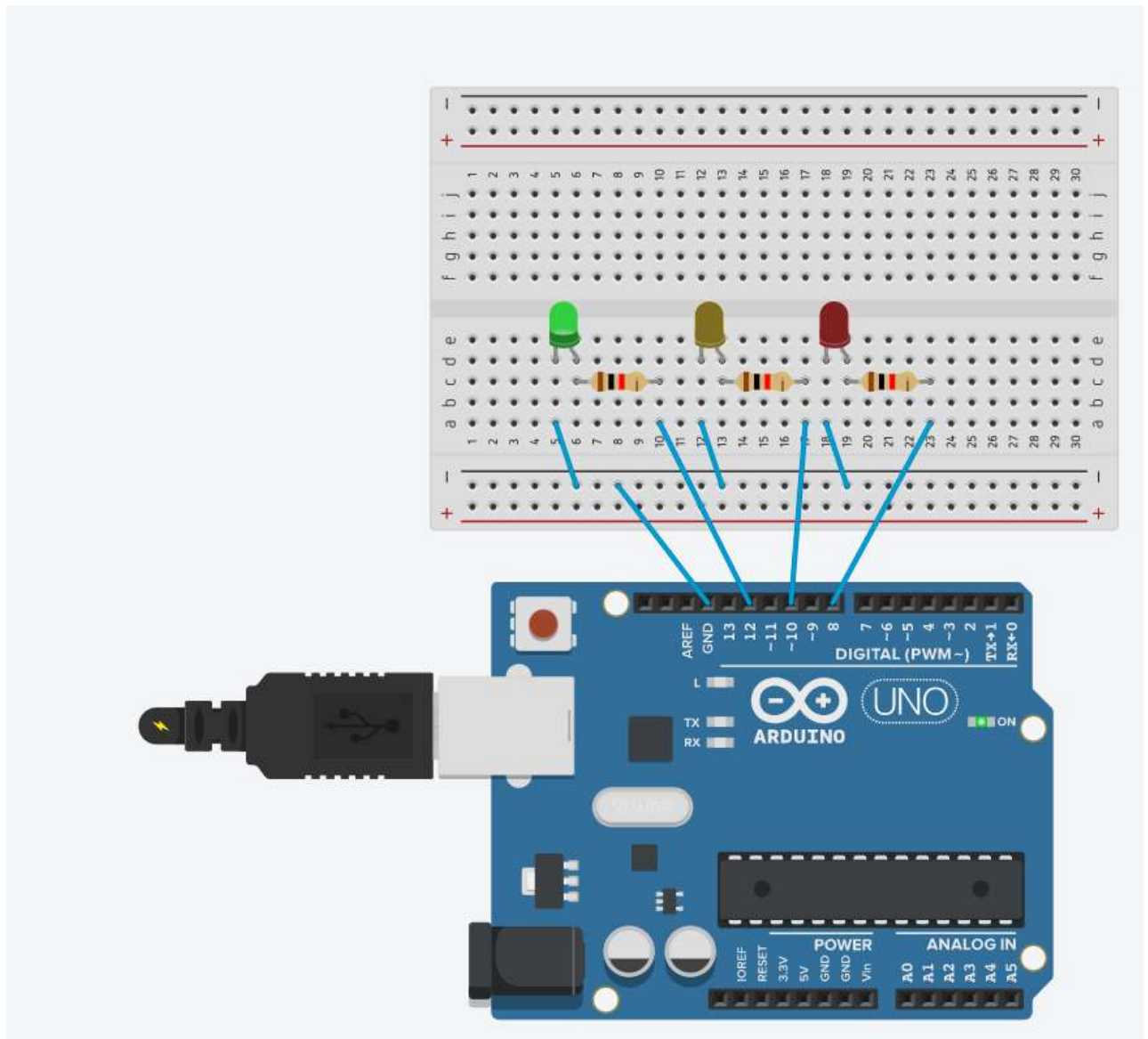
### Simulation & Code:

#### 1) Blink test:





## 2) Traffic Control System:



### Code:

```
#define RED_PIN 8
#define YELLOW_PIN 10
#define GREEN_PIN 12
int red_on = 3000;
int red_yellow_on = 1000;
int green_on = 3000;
int green_blink = 500;
int yellow_on = 1000;
void setup() {
  //ports for connecting LEDs
  pinMode(RED_PIN, OUTPUT);
  pinMode(YELLOW_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
```

```

}
void loop() {
//turning on voltage at output red LED
digitalWrite(RED_PIN, HIGH);
//to make red LED on
delay(red_on);
//to turn yellow LED on
digitalWrite(YELLOW_PIN, HIGH);
delay(red_yellow_on);
//turning off RED_PIN and YELLOW_PIN, and turning on greenLED
digitalWrite(RED_PIN, LOW);
digitalWrite(YELLOW_PIN, LOW);
digitalWrite(GREEN_PIN, HIGH);
delay(green_on);
digitalWrite(GREEN_PIN, LOW);
//for turning green Led on and off for 3 times
for(int i = 0; i < 3; i = i+1)
{
delay(green_blink);
digitalWrite(GREEN_PIN, HIGH);
delay(green_blink);
digitalWrite(GREEN_PIN, LOW);
}
//for turning on yellow LED
digitalWrite(YELLOW_PIN, HIGH);
delay(yellow_on);
digitalWrite(YELLOW_PIN, LOW);
}

```

### **How it works:**

In the traffic system, red light comes on after green light and yellow light comes off, thus when we need to halt the cars, red light comes on for 3 seconds and then yellow light comes on for 1 second. Both the red and yellow lights are turned off in order to turn on the green light for 3 seconds later. To turn on the red light again, the green light must blink three times for 0.5 seconds, then the yellow light is turned on for one second, then the red light is turned on again after the yellow light is turned off.

**Discussion:**

In this experiment, an Arduino Uno was used to construct a traffic management system. A website named tinkercad.com was utilized to achieve this. A circuit had been designed once an account had been established. Three LED lights, RED, YELLOW, and GREEN, were used with three resistors and a breadboard. The resistors have been removed from the circuit and placed on the ground. The coding was implemented after the circuit was designed. LED blink and delay routines were added to the scripts, which were developed in the Arduino IDE. The code ran correctly after being written and simulated, and the outcome was recorded.

**Reference(s):**

- 1) <https://www.arduino.cc/>.
- 2) <https://www.coursera.org/learn/arduino/lecture/ei4ni/1-10-first-glance-at-a-program>
- 3) Jeremy Blue; Exploring Arduino: Tools and Techniques for Engineering Wizardry



