**K L UNIVERSITY**

**FRESHMAN ENGINEERING DEPARTMENT**

**A Project Based Lab Report**

**On**

**ARDUINO TRAFFIC LIGHT CONTROLLER**

**SUBMITTED BY:**

170040320 **J.V.N.D.S.R.PRASAD**

170040351 **K.V.SAI TEJA**

**UNDER THE ESTEEMED GUIDANCE OF**

**Mr. SRIPATH ROY KOGANTI**

**Professor**



**KL UNIVERSITY**

Green fields, Vaddeswaram – 522 502

Guntur Dt., AP, India.

**DEPARTMENT OF BASIC ENGINEERING SCIENCES**



**CERTIFICATE**

This is to certify that the project based laboratory report entitled “**ARDUINO TRAFFIC LIGHT CONTROLLER**” submitted by **Mr. J.V.N.D.S.R.PRASAD, Mr. K.V.SAI TEJA** bearing Regd. No.170040320, 170040351 to the **Department of Basic Engineering Sciences, KL University** in partial of the requirements for the completion of a project based Laboratory in INTRODUCTION TO ELECTRONIC ENGINEERINGcourse in I B.Tech I Semester, is a bonafide record of the work carried out by him/her under my supervision during the academic year 2017 – 2 018

PROJECT SUPERVISOR HEAD OF THE DEPARTMENT

Mr. SRIPATH ROY KOGANTI Mr. A.C.S. SASTHRY

**ACKNOWLEDGEMENTS**

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project based laboratory report.

I express the sincere gratitude to our principal **Dr. A. Anand Kumar** for his administration towards our academic growth.

I express sincere gratitude to our Coordinator and HOD-BES **Mr. A.C.S. SASTHRY** for her leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor Mr. SRIPATH ROY KOGANTI for his/her novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

By

**170040320 J.V.N.D.S.RPRASAD**

**170040351 K.V.SAI TEJA**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.no** | **TITLE** | **PAGE NO.** |
| **1** | **COVER PAGE** | **1** |
| **2** | **CERTIFICATE** | **2** |
| **3** | **DECLARATION** | **3** |
| **4** | **CONTENTS INDEX** | **4** |
| **5** | **ABSTRACT** | **5** |
| **6** | **INTRODUCTION** | **6-10** |
| **7** | **BLOCK DIAGRAM and**  **CIRCUIT DIAGRAM** | **11-12** |
| **8** | **HARWARE MODULES EXPLANATION** | **13** |
| **9** | **SOFTWARE** | **14-17** |
| **10** | **RESULTS** | **18-19** |
| **11** | **CONCLUSIONS** | **20** |
| **12** | **SOURCE CODE** | **21-22** |
| **13** | **REFERENCES** | **23** |

**ABSTRACT**

Aim of the project is to build a TRAFFIC LIGHT CONTROLLER model using ARDUINO. In this project we are going to construct a junction having two traffic lights and shows the working of the model. By this model we came to know how traffic lights are blinking accordingly for a certain period of time. This model provides us the idea about how traffic lights are controlled by using electronics and takes us to know how we are going to use it and connection to people requirements. By using Arduino, LEDs and some wires for connections we are going to build a Traffic light controller and know how it works.

**INTRODUCTION**

**ARDUINO:-**

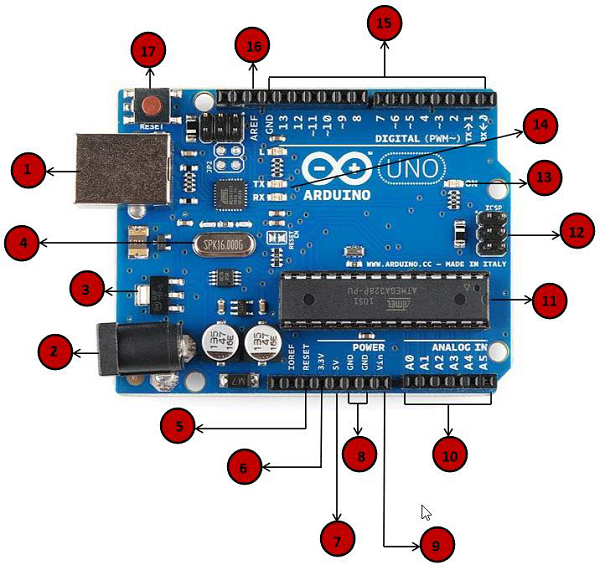
Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/)), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/).

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various expansion boards (*shields*) and other circuits. The boards feature serial communications interfaces, including [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus)(USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). In addition to using traditional compiler toolchains, the Arduino project provides an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) based on the [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) language project.

Using this ARDUINO as main hardware we are going to do this project.

**ARDUINO UNO BOARD**



**ARDUINO BOARD DESCRIPTION**

|  |  |
| --- | --- |
| Power USB | **Power USB**  Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1). |
| Barrel Jack | **Power (Barrel Jack)**  Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2). |
| Voltage Regulator | **Voltage Regulator**  The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements. |
| Crystal Oscillator | **Crystal Oscillator**  The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz. |
| Arduino Reset | **Arduino Reset**  You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5). |
| Pins | **Pins (3.3, 5, GND, Vin)**   * 3.3V (6) − Supply 3.3 output volt * 5V (7) − Supply 5 output volt * Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. * GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. * Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply. |
| Analog pins | **Analog pins**  The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor. |
| Main microcontroller | **Main microcontroller**  Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet. |
| ICSP pin | **ICSP pin**  Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus. |
| Power LED indicator | **Power LED indicator**  This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection. |
| TX and RX LEDs | **TX and RX LEDs**  On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process. |
| Digital I/O | **Digital I/O**  The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM. |
| AREF | **AREF**  AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins. |

**BLOCK DIAGRAM**

RED

LED

YELLOW

LED

GREEN

LED

**ARDUINO**

RED

LED

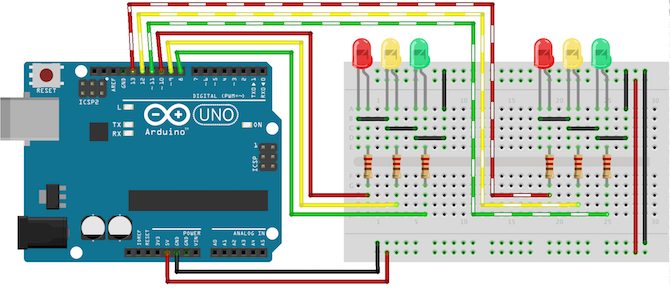
YELLOW

LED

GREEN

LED

**CIRCUIT DIAGRAM**



**HARDWARE MODULES**

**EXPLANATION**

**HARDWARE USED:-**

1. ARDUINO
2. Red, Yellow and Green LEDs
3. A BREAD BOARD
4. 6 x 220Ω RESISTORS
5. CONNECTING WIRES

ARDUINO:-

Our main hardware is Arduino. Using this we can control traffic light system. Arduino is used as interconnection between user and hardware. It makes user to do what he wants to do with hardware. User writes code and dump to the hardware using Arduino as a mediating device.

LEDs:-

In this experiments we used red, yellow and green colored LEDs for showing the working of Traffic lights.

BREAD BOARD:-

A **breadboard** is a construction base for [prototyping](https://en.wikipedia.org/wiki/Prototype) of [electronics](https://en.wikipedia.org/wiki/Electronic_circuit). Our all hardware connections are done on this Bread board.

RESISTORS:-

Resistors are used to obstruct the flow of current to ground. To obstruct flow of current.

CONNECTING WIRES:-

Connecting Wires are used to connect wires on the bread board.

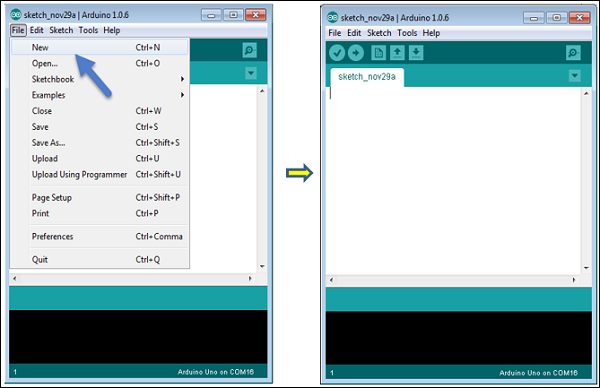
**SOFTWARE**

How to use ARDUINO SOFTWARE:-

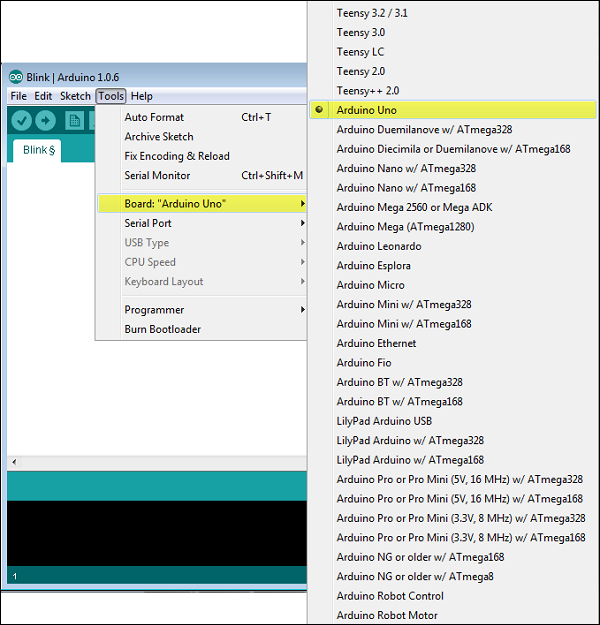
1. First we must have our ARDUINO board and a cable.
2. Download ARDUINO IDE SOFTWARE.
3. Power up the Arduino board.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

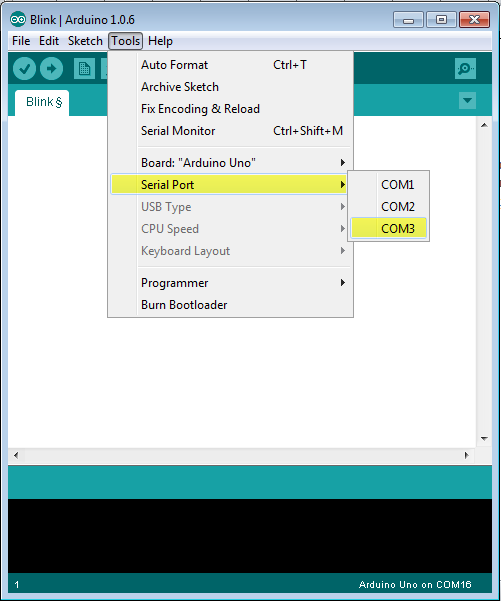
1. Launch ARDUINO IDE
2. Open our project.



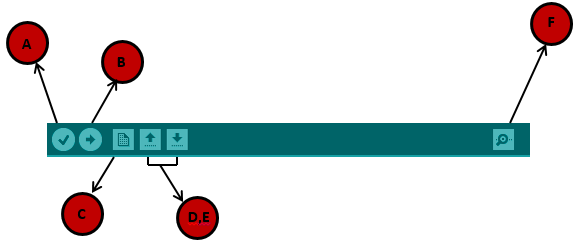
1. Select our Arduino board.



1. Select our Serial port.



1. Upload the program to our Arduino board.



**A** − Used to check if there is any compilation error.

**B** − Used to upload a program to the Arduino board.

**C** − Shortcut used to create a new sketch.

**D** − Used to directly open one of the example sketch.

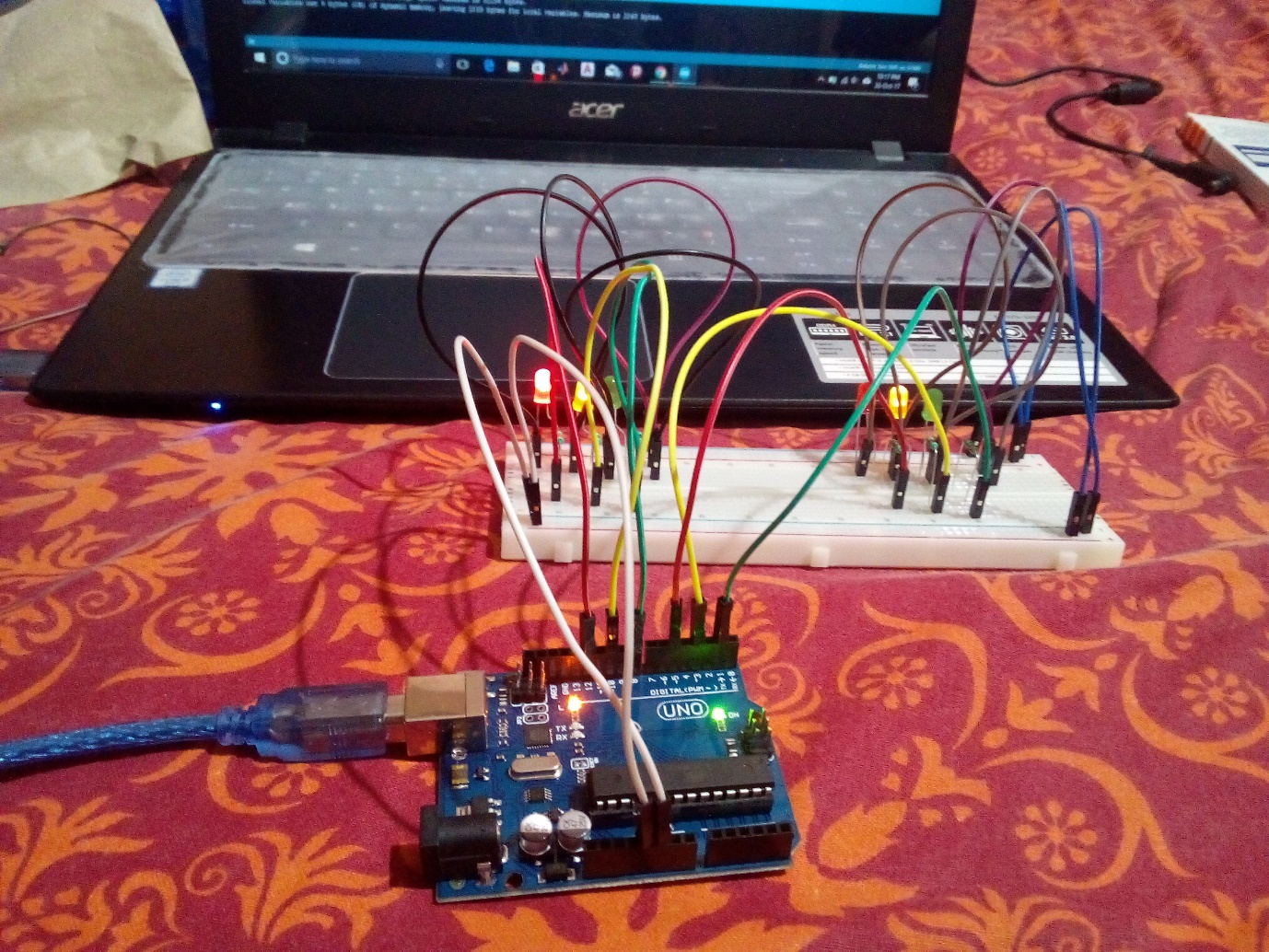
**E** − Used to save your sketch.

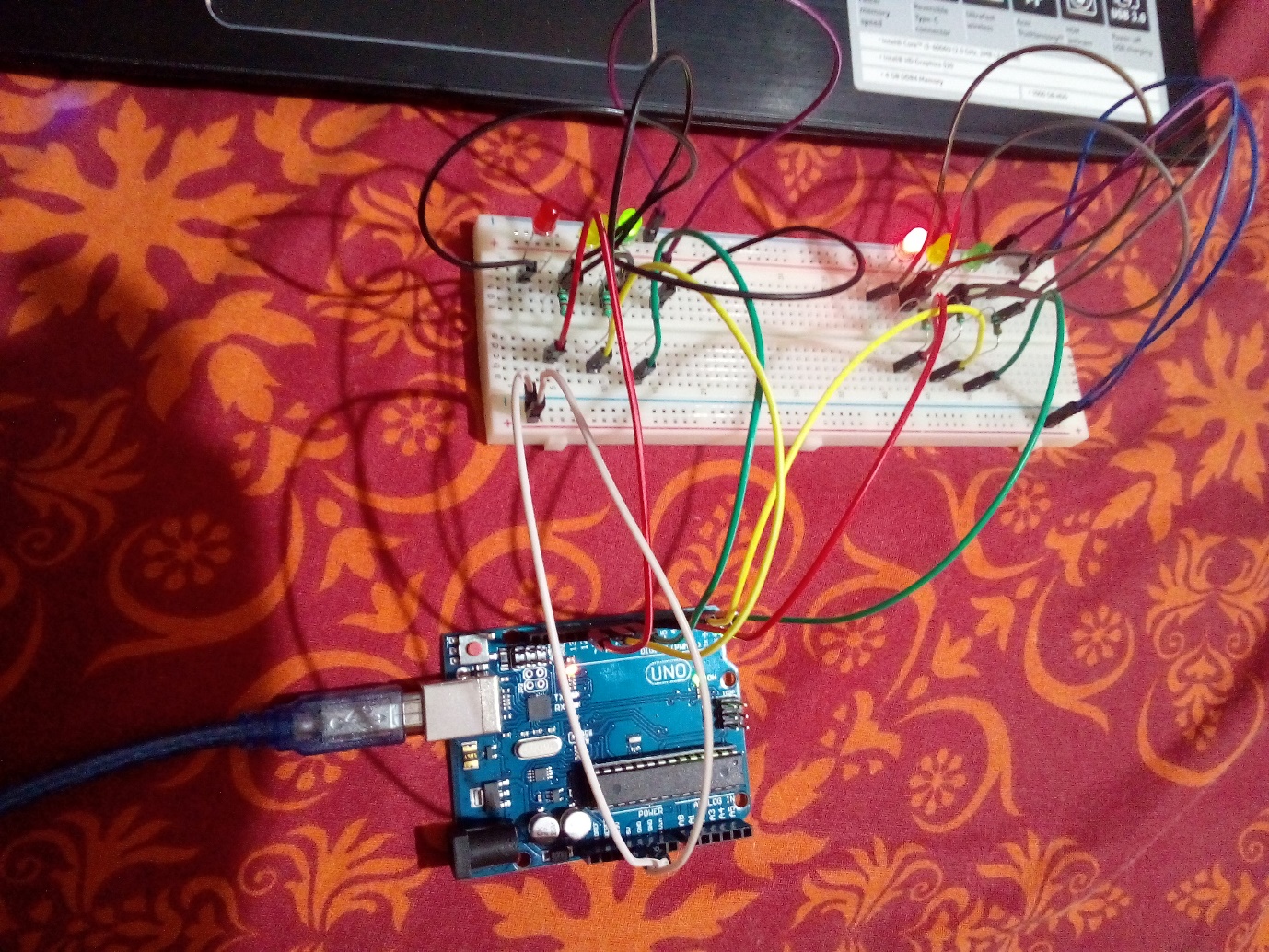
**F** − Serial monitor used to receive serial data from the board and send the serial data to the board.

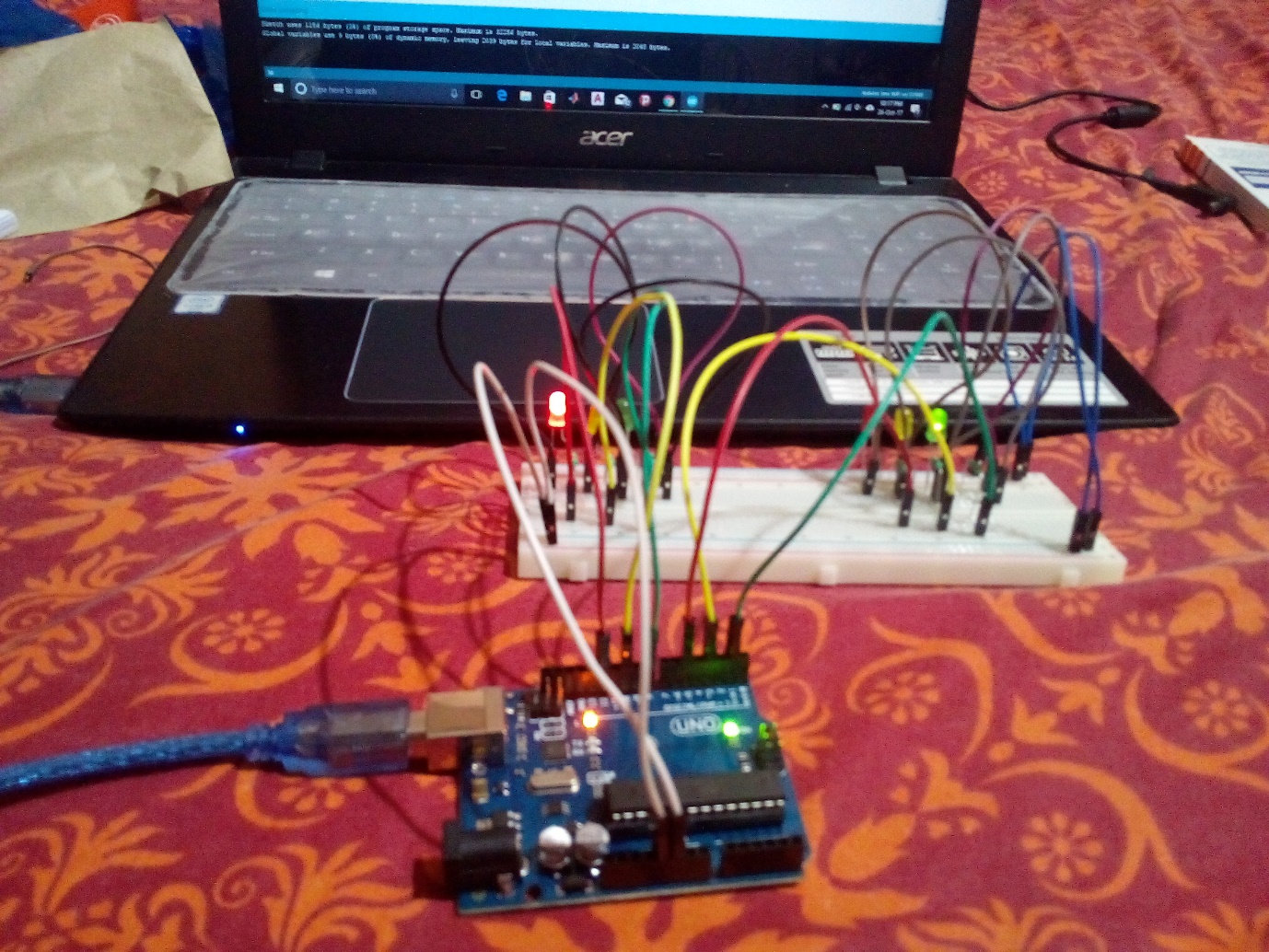
Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

**RESULTS**

By using ARDUINO we had done TRAFFIC LIGHT CONTROLLING model.







**CONCLUSIONS**

By using Arduino, LEDs, Resistors, Bread board and connecting wires we made Traffic Light controller model.

By making some modifications like using four traffic lights we can build a traffic junction model.

By using other sensors like Proximity sensors, Gas sensors, LDR, we can design a better model which is used in future.

**SOURCE CODE**

void setup() {

// put your setup code here, to run once:

//light 1

pinMode(12,OUTPUT); //Red 1

pinMode(10,OUTPUT); //Yellow 1

pinMode(8,OUTPUT); //Green 1

//light 2

pinMode(5,OUTPUT); //Red 2

pinMode(3,OUTPUT); //Yellow 2

pinMode(1,OUTPUT); //Green 2

}

void loop() {

// put your main code here, to run repeatedly:

changeLights();

delay(15000);

}

void changeLights(){

digitalWrite(8,LOW);

digitalWrite(10,HIGH);

digitalWrite(3,HIGH);

delay(5000);

digitalWrite(10,LOW);

digitalWrite(12,HIGH);

digitalWrite(3,LOW);

digitalWrite(5,LOW);

digitalWrite(1,HIGH);

delay(5000);

digitalWrite(10,HIGH);

digitalWrite(3,HIGH);

digitalWrite(1,LOW);

delay(3000);

digitalWrite(8,HIGH);

digitalWrite(10,LOW);

digitalWrite(12,LOW);

digitalWrite(3,LOW);

digitalWrite(5,HIGH);

delay(5000);

}

**REFERENCES**

[www.tutorialspoint.com](http://www.tutorialspoint.com) ARDUINO

[www.makeuseof.com](http://www.makeuseof.com) [Arduino Programming For Beginners: The Traffic Light Controller](http://www.makeuseof.com/tag/arduino-traffic-light-controller/)

[www.arduino.cc](http://www.arduino.cc)