TRAFFIC LIGHT CONTROL USING ARDUINO

A PROJECT REPORT SUBMITTED

for external evaluation of

ELECTRICAL & ELECTRONICS WORKSHOP

Bachelor of Technology

In

Electronics & Communication Engineering

By

SUMIT KUMAR PUROHIT 2141016376

1ST Semester, ECE (2141032)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

Institute of Technical Education and Research

SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY

Bhubaneswar, Odisha, India

(MARCH, 2022)

ACKNOWLEDGEMENT

I would like to take this opportunity to express our gratitude to our respected mentors Prof. Priyabrata Pattanaik, Dr. Biswaranjan Swain, Dr. Hara Prasad Tripathy for the inspired guidance, insight, continuous encouragement, timely suggestions that they have provided throughout the duration of this work. The present work, being successfully completed due to their sincere monitoring and vital inputs.

We are grateful to Prof. Bibhu Prasad Mohanty, Head of the Department of Electronics and communication Engineering permitted us to make use of the available facilities in the department to carry out the project successfully.

We would also thank all our staff members especially Mr. Pradeep Kumar Bebarta, Mr. Uma Sankar Sahu, Mrs. Manjulata, friends and faculty of Department of Electronics & Communication Engineering for their support and all kinds of help to accomplish this work.

Signature of the Student

SUMIT KUMAR PUROHIT 2141016376

(Name and Registration No.)

ABSTRACT

The simple traffic light controller design project was introduced to alleviate this shortcoming and gain experience in solving implementation and interfacing problems of a modern digital system. we implement a fully functional traffic signal controller for a four-way intersection. The function of a traffic light controller requires sophisticated control and coordination to ensure that traffic moves as smoothly and safely as possible. The project was designed in DSCH by using a 4-bit counter. The software implementation of the circuit was simple and reliable

DECLARATION

I declared that

a. The work contained in this report is original and has been done by me.

b. I have followed the guidelines provided by the Department in preparing the report.

c. I have followed the professional and ethical responsibility provided by the university.

Name of the Student	*Registration Number*	*Signature*
SUMIT KUMAR PUROHIT	**2141016376**	

TABLE OF CONTENTS

Project Tittle

Traffic light control using Arduino

Problem Definition

Controlling LED through Arduino

Project Objective

***Getting information from different Sources**

***Implementation of Block diagram**

***Implementation of Flow chart**

***Simulation of Required Circuit Through Online Simulator (TINKER CAD)**

*** LEDS control through Arduino**

*** Breadboard Implementation**

*** LEDS control through Arduino**

***Preparation of Physical Prototype**

Theoretical Background

*** Getting information from different Source: -**

Information can come from virtually anywhere — media, blogs, personal experiences, books, journal and magazine articles, expert opinions, encyclopedias, and web pages, MAKE ELOCTRONICS BOOK — and the when looking for background information on a topic.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards 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 (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments.

Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50

- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

- **Open source and extensible software** - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

- Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

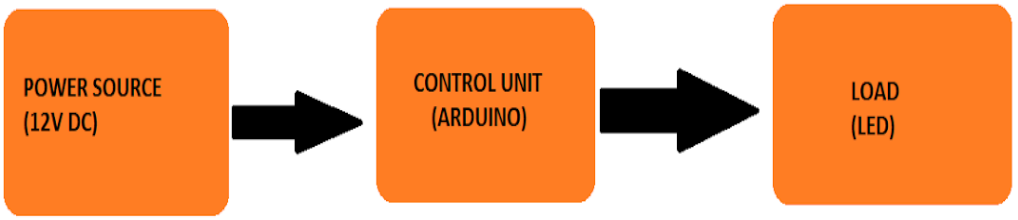
- Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name *Arduino* to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in *-duino*.
- MOST Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, or ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed *shields*. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lilypad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.
- Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino Uno is the Opti boot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using

USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

● Implementation of Block Diagram

POWER SOURCE AS (12V DC) CONTROL UNIT (ARDUINO)

LOAD(LED)



***SOURCE:**

To start the circuit, we need a source in the form to start the simulation of the circuit and to get the required output.

There are various ways to supply power to the circuit, from which the most commonly used source is the 12V DC adapter and the 9V Alkaline battery. The 12V DC adapter is of moderate cost and we can say it is a permanent source of battery and it is used for a way long period of time. So, there is a chance that our circuit will work efficiently.

The 9V Alkaline battery is of low cost and we can say that it is compact in size and also portable everywhere but the disadvantage is that it is temporary as there is chance of running out of charge.

Although both the sources work efficiently on the circuit.

The 12 V battery and the 9V Alkaline battery are the most commonly used sources.

***CONTROL UNIT:**

A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio (older) and Atmel Studio (newer)

The Arduino integrated development environment (IDE) is a cross-platform application (for Microsoft Windows, macOS, and Linux) that is written in the Java programming language. It originated from the IDE for the languages *Processing* and *Wiring*. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

Fig1

Fig2

Arduino and Arduino-compatible boards use printed circuit expansion boards called *shields*, which plug into the normally supplied Arduino pin headers. Shields can provide motor controls for 3D printing and other applications, GNSS (satellite navigation), Ethernet, liquid crystal display (LCD), or breadboarding

LOAD(LED)

A **light-emitting diode** (**LED**) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The colour of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device.

Appearing as practical electronic components in 1962, the earliest LEDs emitted low-intensity infrared (IR) light.[1] Infrared LEDs are used in remote-control circuits, such as those used with a wide variety of consumer electronics. The first visible-light LEDs were of low intensity and limited to red. Early LEDs were often used as indicator lamps, replacing small incandescent bulbs, and in seven-segment displays. Recent developments have produced LEDs available in visible, ultraviolet (UV), and

infrared wavelengths, with high, low, or intermediate light output, for instance white LEDs suitable for room and outdoor area lighting. LEDs have also given rise to new types of displays and sensors, while their high switching rates are useful in advanced communications technology with applications as diverse as aviation lighting, fairy lights, automotive headlamps, advertising, general lighting, traffic signals, camera flashes, lighted wallpaper, horticultural grow lights, and medical devices.

LEDs have many advantages over incandescent light sources, including lower power consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. In exchange for these generally favourable attributes, disadvantages of LEDs include electrical limitations to low voltage and generally to DC (not AC) power, inability to provide steady illumination from a pulsing DC or an AC electrical supply source, and lesser maximum operating temperature and storage temperature. In contrast to LEDs, incandescent lamps can be made to intrinsically run at virtually any supply voltage, can utilize either AC or DC current interchangeably, and will provide steady illumination when powered by AC or pulsing DC even at a frequency as low as 50 Hz. LEDs usually need electronic support components to function, while an incandescent bulb can and usually does operate directly from an unregulated DC or AC power source.

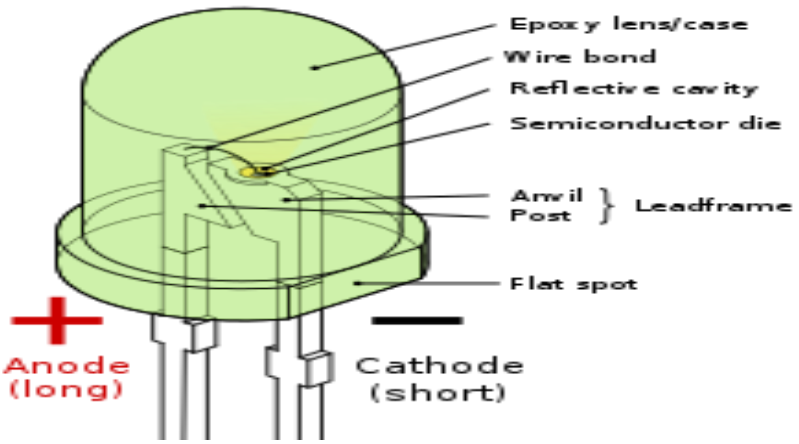
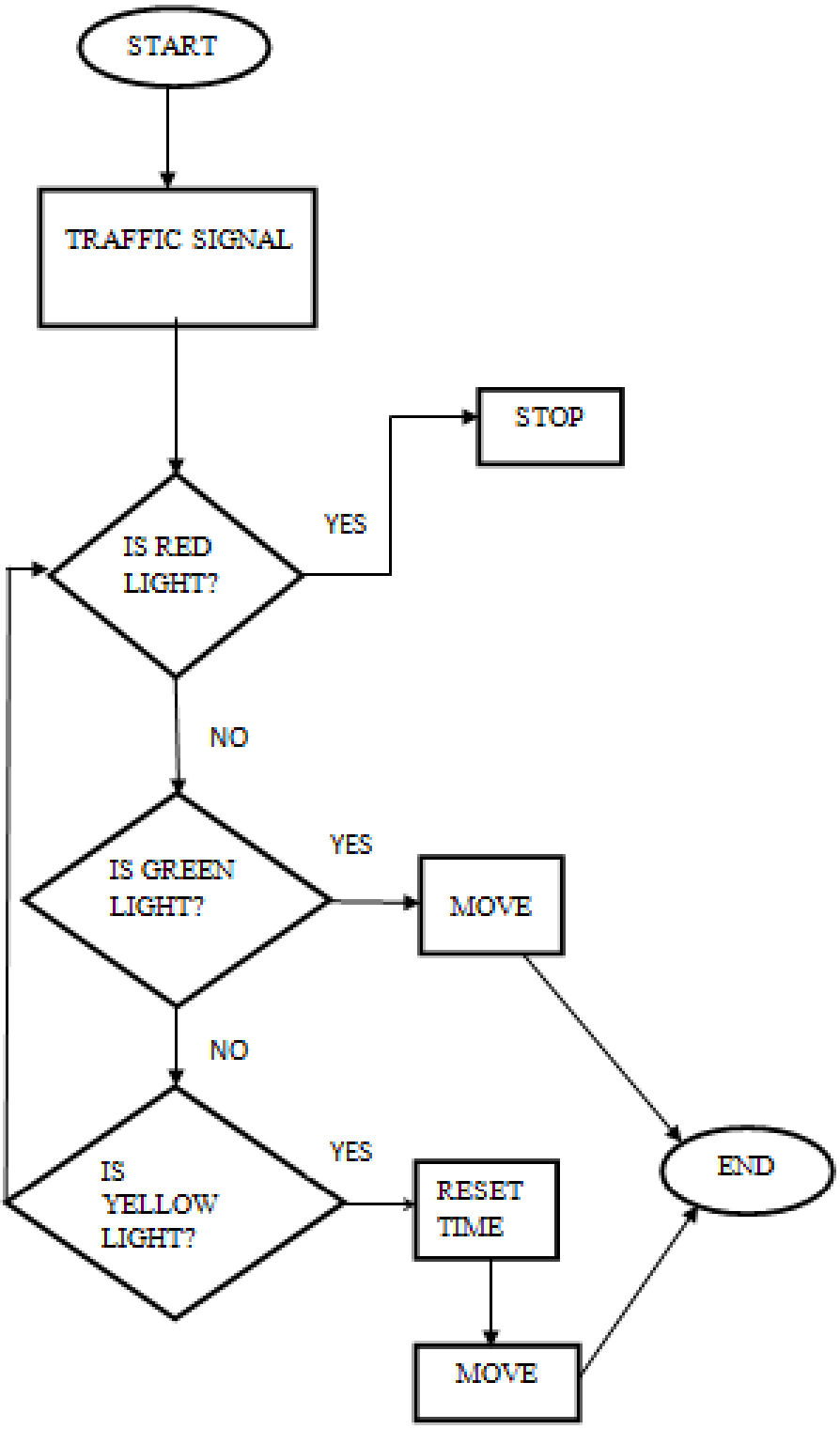


Fig3

IMPLEMENTATION OF FLOW CHAT



*CODE

```
int signal1[] = {1, 2, 3};

int signal2[] = {4, 5, 6};

int signal3[] = {7, 8, 9};

int signal4[] = {10, 11, 12};

int redDelay = 5000;

int yellowDelay = 2000;

void setup() {

 // Declaring all the LEDs as output

 for (int i = 0; i < 3; i++) {

   pinMode(signal1[i], OUTPUT);

   pinMode(signal2[i], OUTPUT);

   pinMode(signal3[i], OUTPUT);

   pinMode(signal4[i], OUTPUT);

 }

}

void loop() {
```

```
  // Making Green LED at signal 1 and red LEDs at other signal HIGH
  digitalWrite(signal1[2], HIGH);
  digitalWrite(signal1[0], LOW);
  digitalWrite(signal2[0], HIGH);
  digitalWrite(signal3[0], HIGH);
  digitalWrite(signal4[0], HIGH);
  delay(redDelay);
// Making Green LED at signal 1 LOW and making yellow LED at signal 1 HIGH for
2 seconds
  digitalWrite(signal1[1], HIGH);
  digitalWrite(signal1[2], LOW);
  delay(yellowDelay);
  digitalWrite(signal1[1], LOW);
 // Making Green LED at signal 2 and red LEDs at other signal HIGH
  digitalWrite(signal1[0], HIGH);
  digitalWrite(signal2[2], HIGH);
  digitalWrite(signal2[0], LOW);
  digitalWrite(signal3[0], HIGH);
  digitalWrite(signal4[0], HIGH);
```

```
  delay(redDelay);


  // Making Green LED at signal 2 LOW and making yellow LED at signal 2 HIGH for
2 seconds
  digitalWrite(signal2[1], HIGH);
  digitalWrite(signal2[2], LOW);
  delay(yellowDelay);
  digitalWrite(signal2[1], LOW);


  // Making Green LED at signal 3 and red LEDs at other signal HIGH
  digitalWrite(signal1[0], HIGH);
  digitalWrite(signal2[0], HIGH);
  digitalWrite(signal3[2], HIGH);
  digitalWrite(signal3[0], LOW);
  digitalWrite(signal4[0], HIGH);
  delay(redDelay);


  // Making Green LED at signal 3 LOW and making yellow LED at signal 3 HIGH for
2 seconds
```

```
  digitalWrite(signal3[1], HIGH);
  digitalWrite(signal3[2], LOW);
  delay(yellowDelay);
  digitalWrite(signal3[1], LOW);

  // Making Green LED at signal 4 and red LEDs at other signal HIGH
  digitalWrite(signal1[0], HIGH);
  digitalWrite(signal2[0], HIGH);
  digitalWrite(signal3[0], HIGH);
  digitalWrite(signal4[2], HIGH);
  digitalWrite(signal4[0], LOW);
  delay(redDelay);

  // Making Green LED at signal 4 LOW and making yellow LED at signal 4 HIGH for
2 seconds
  digitalWrite(signal4[1], HIGH);
  digitalWrite(signal4[2], LOW);
  delay(yellowDelay);
  digitalWrite(signal4[1], LOW);
```

}

*Simulation of the required circuit through online simulator (TINKERCAD)

Electronic Circuits with Tinkercad An electronic circuit consist of a variety of electronic components such as LED's, switches, resistors and a power source. Tinkercad is a free online service for creating 3D shapes and developing digital prototypes of electronic components. The digital prototype includes basic circuits with LED lights, battery, switches, buzzers, etc. The Prototyping process used in Tinkercad is a process where we can develop electronic circuits using the components in a flexible manner that can be quickly updated, modified and tested within the browser, before building it in real life. To start creating circuits, perform the following steps.

Step 1: Log on to www.tinkercad.com.

Step 2: Sign in to your account and then click Circuits.

Step 3: Click Create new Circuit button.

Workspace It is an area where the electronic components and 3D objects are placed. Components of a circuit Tinkercad displays the electronic components used to create circuits in the panel usually at the right side of the page. Some of the electronic components used to create circuits are.

- Basic components – It consists of a battery, push button, resistor, LED, capacitor, etc.

- Circuit Assembly - Circuit assemblies are simple, pre-made circuits that can be incorporated into the 3D designs.

Some of the circuit assemblies available are glow circuit, move circuit and spin circuit. Rotate It is used to rotate by 360 degrees any component placed onto the workspace.

Using Breadboard Perform the following steps to use the breadboard in Tinkercad.

Step 1: Locate the breadboard from the right panel and click it once. The breadboard will be selected and temporarily attached to the mouse pointer.

Step 2: Move the mouse pointer onto the workspace and click the mouse button to place the breadboard onto the workspace.

Step 3: Click on the zoom to fit button, so the breadboard is centered and magnified.

Step 4: Move the mouse pointer over one of the holes in the main area of the board. The hole will be identified with a red square and a black border. The other holes will be identified with green circles.

Breadboard

It is a flat piece of wood that is provided with holes that allow users to build a circuit including the components and connections without the need for soldering due to which it is reusable and finds its best usage for the students who are new to build circuits. The holes in any horizontal line will be in series whereas the vertical holes remain at an equipotential state. There are different sizes of breadboards available in the market which include "full size", "half-size" and, "mini size" breadboards.

Resistor:

Resistor as the name signifies, opposes the flow of electrons. The Resistor is a passive element and has two terminals. They are basically used to monitor the current

flow and also as voltage dividers. The resistor that we would use in this circuit is a 10kilohm resistor and the color code for this resistor is orange, *black, Brown & Gold.*

LED

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron hole, releasing energy in the form of photons.

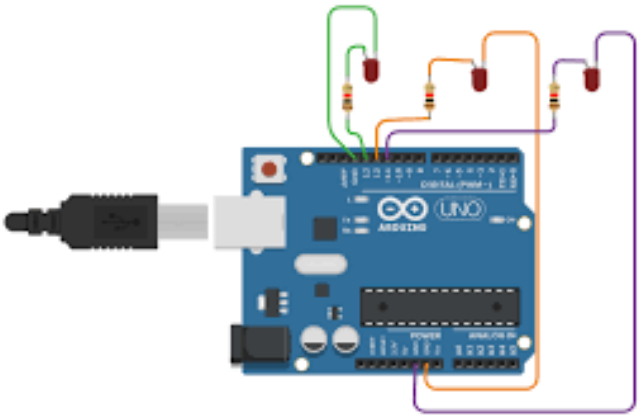


Fig 4

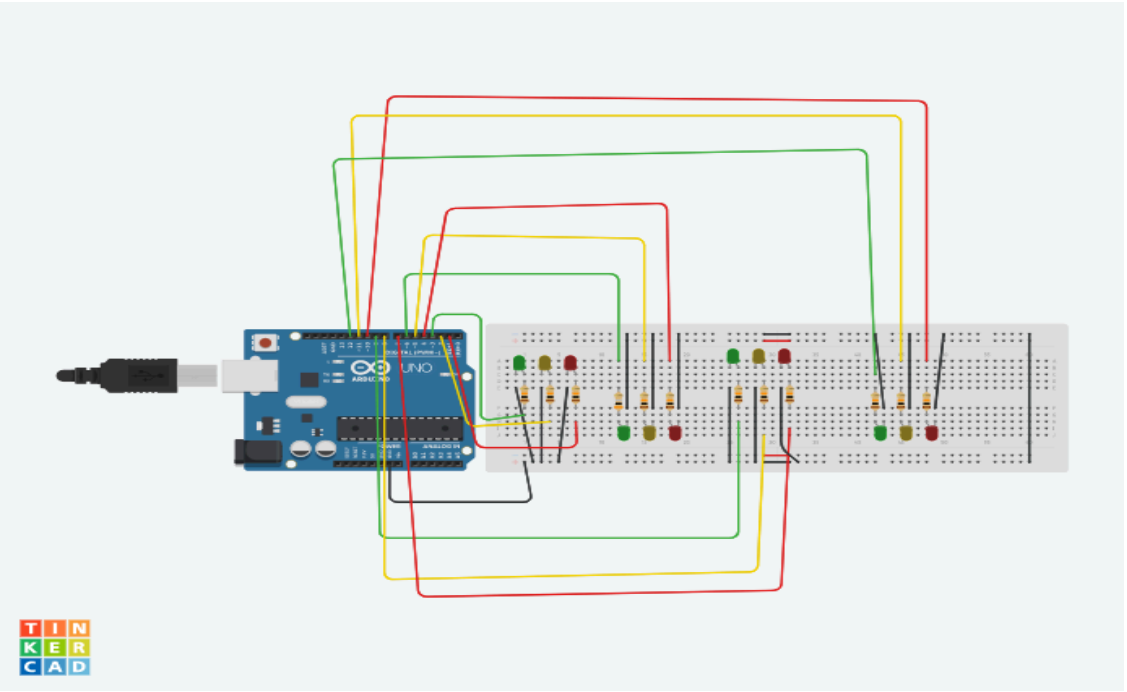


Fig 5

During the online classes We know about the TINKERCAD. How to implement circuit using bread board by dragging the components, by connected the wires, also for better showing changing the colour of wires. Also know about How to change the value of resistor, value of potentiometer, value of capacitor etc.

For this experiment, I use Arduino, 10kohm resistance, 12 led, and wires for connection.

1st I implement the Arduino as shown in TINKERCAD. I made four lanes circuit for a four-way traffic. In lane one, the anode legs of red led is connected to resistor and resistor is connected to D1 of Arduino. The cathode Red LED is connected with ground.

In lane one, the anode legs of yellow led is connected to resistor and resistor is connected to D2 of Arduino. The cathode Yellow LED is connected with ground.

In lane one, the anode legs of green led is connected to resistor and resistor is connected to D3 of Arduino. The cathode Green LED is connected with ground.

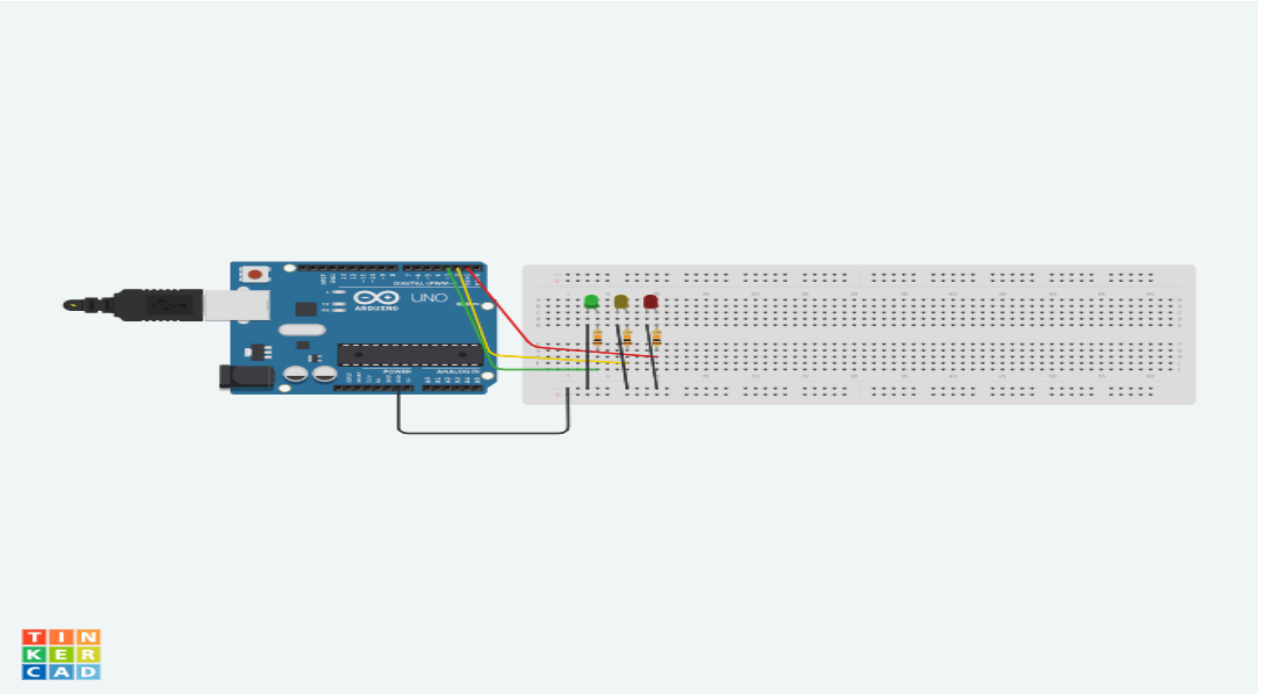


Fig6

In lane two, the anode legs of red led is connected to resistor and resistor is connected to D4 of Arduino. The cathode Red LED is connected with ground.

In lane two, the anode legs of yellow led is connected to resistor and resistor is connected to D5 of Arduino. The cathode Yellow LED is connected with ground.

In lane two, the anode legs of green led is connected to resistor and resistor is connected to D6 of Arduino. The cathode Green LED is connected with ground.

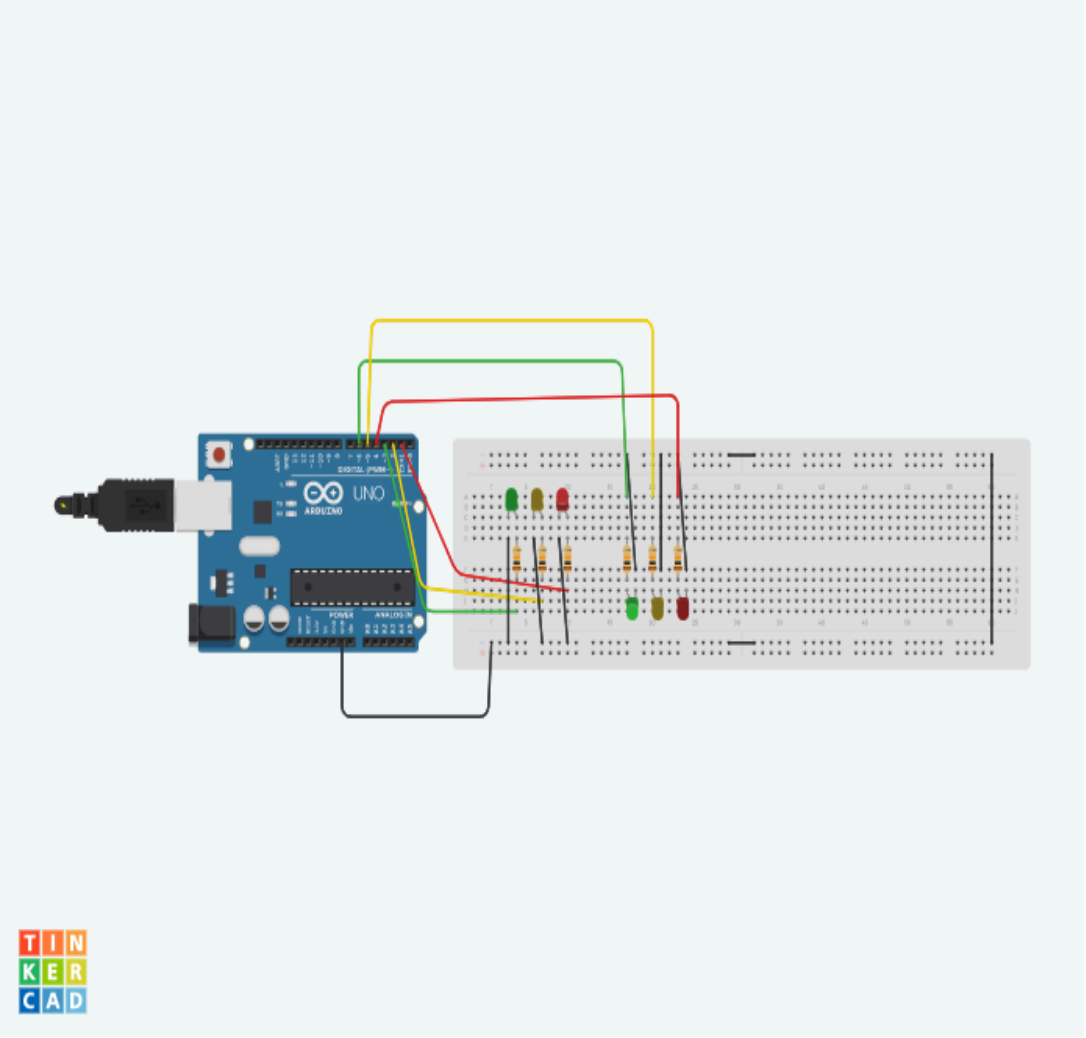


Fig7

In lane three, the anode legs of red led is connected to resistor and resistor is connected to D7 of Arduino. The cathode Red LED is connected with ground.

In lane three, the anode legs of yellow led is connected to resistor and resistor is connected to D8 of Arduino. The cathode Yellow LED is connected with ground.

In lane three, the anode legs of green led is connected to resistor and resistor is connected to D9 of Arduino. The cathode Green LED is connected with ground.

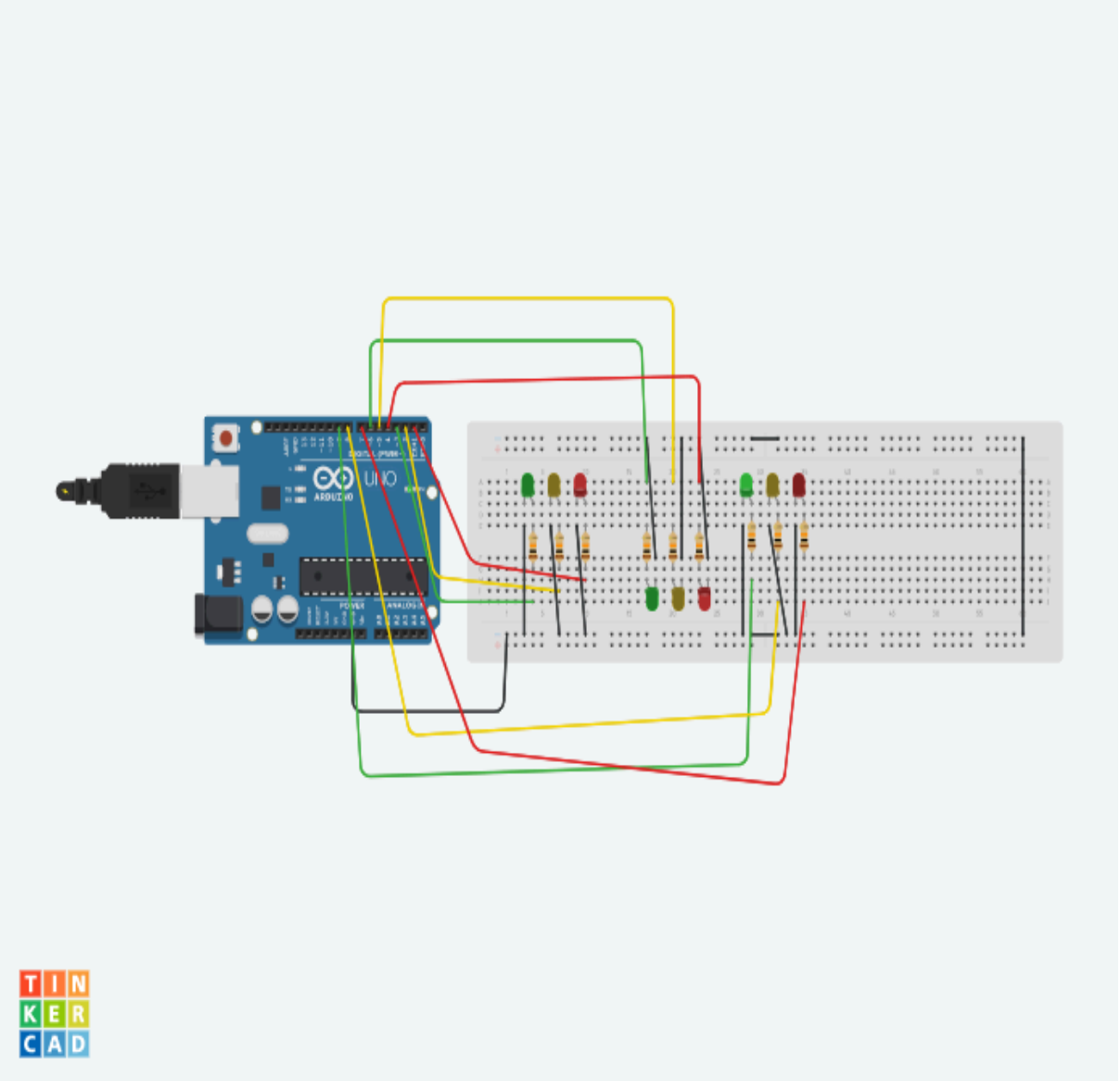


Fig8

In lane four, the anode legs of red led is connected to resistor and resistor is connected to D10 of Arduino. The cathode Red LED is connected with ground.

In lane four, the anode legs of yellow led is connected to resistor and resistor is connected to D11 of Arduino. The cathode Yellow LED is connected with ground.

In lane four, the anode legs of green led is connected to resistor and resistor is connected to D12 of Arduino. The cathode Green LED is connected with ground

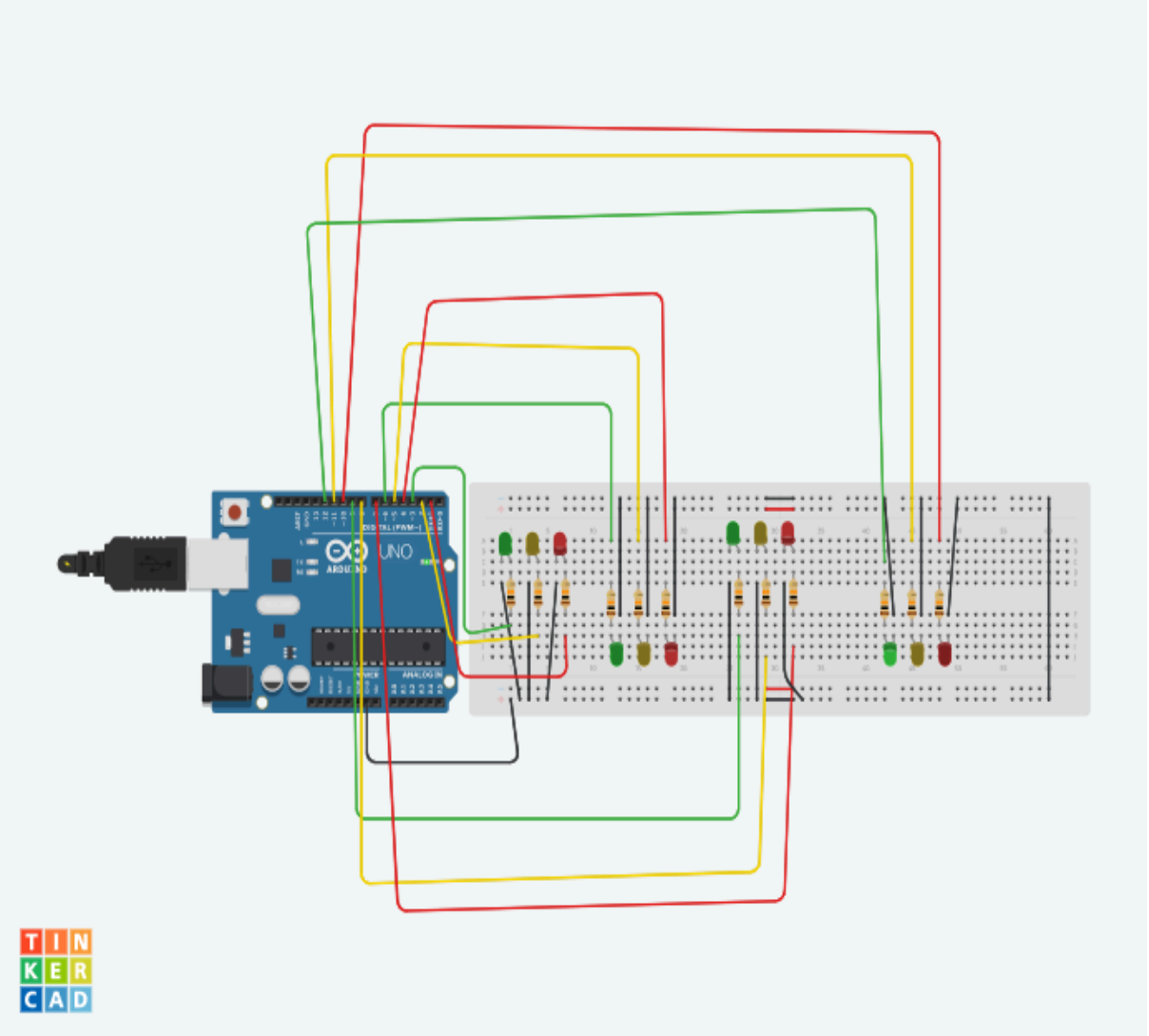


Fig9

*__RESULT: -__

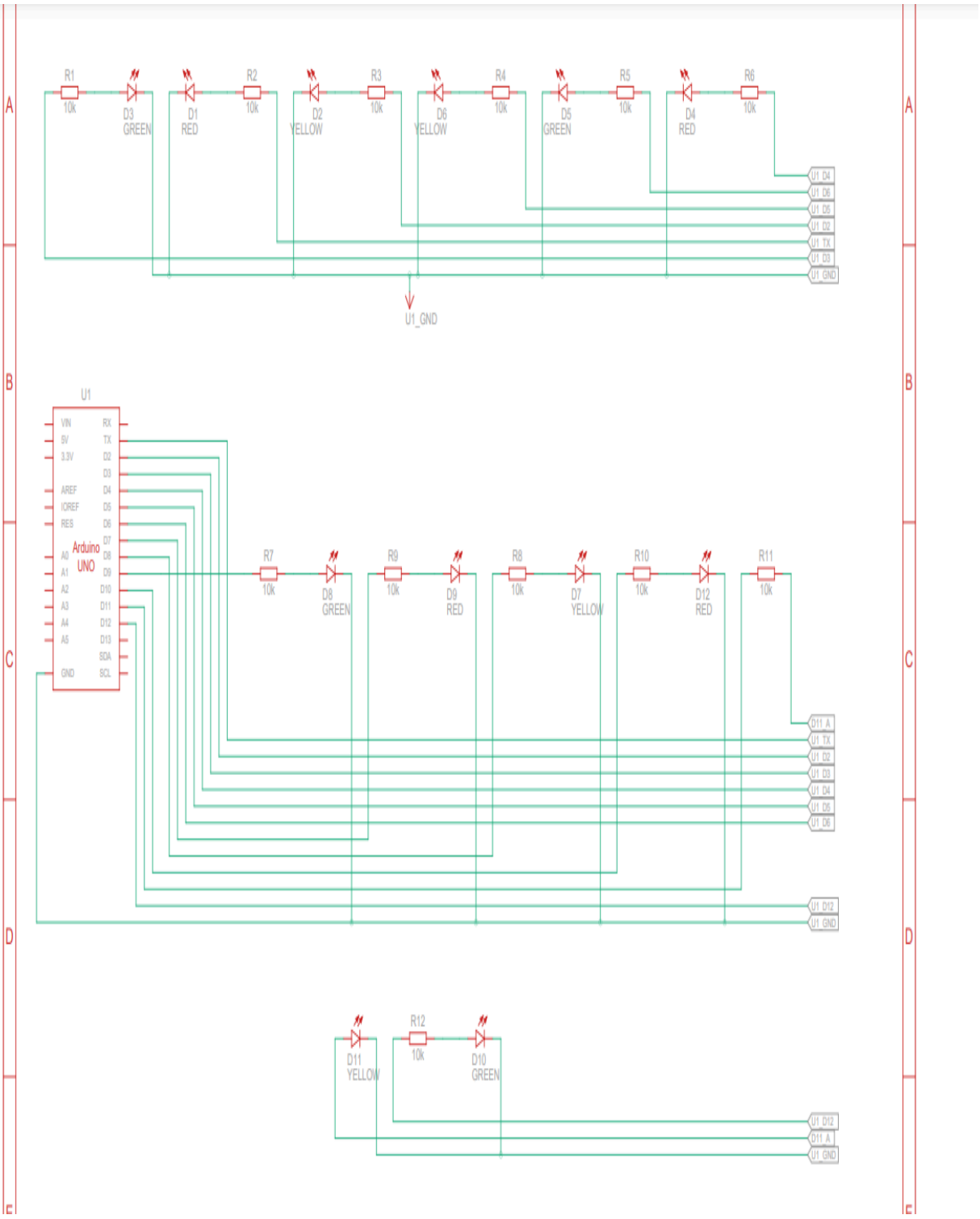


Fig10

-:(SCHEMATIC VIEW OF THIS CIRICUIT)

***<u>BREADBOARD IMPLEMENTAION: -</u>**

A breadboard, or protoboard, is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used when slicing bread. In the 1970s the solderless breadboard (plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property.

A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units.

In a more robust variant, one or more breadboard strips are mounted on a sheet of metal. Typically, that backing sheet also holds a number of binding posts. These posts provide a clean way to connect an external power supply. This type of breadboard may be slightly easier to handle. Several images in this article show such solderless breadboards.

Compared to more permanent circuit connection methods, modern breadboards have high parasitic capacitance, relatively high resistance, and less reliable connections, which are subject to jostle and physical degradation.

Signaling is limited to about 10 MHz, and not everything works properly even well below that frequency.

For this experiment, I use Arduino, 10kohm resistance, 12 led, and wires for connection.

1st I implement the Arduino as shown in TINKERCAD. I made four lanes circuit for a four-way traffic. In lane one, the anode legs of red led is connected to resistor and resistor is connected to D1 of Arduino. The cathode Red LED is connected with ground.

In lane one, the anode legs of yellow led is connected to resistor and resistor is connected to D2 of Arduino. The cathode Yellow LED is connected with ground.

In lane one, the anode legs of green led is connected to resistor and resistor is connected to D3 of Arduino. The cathode Green LED is connected with ground.

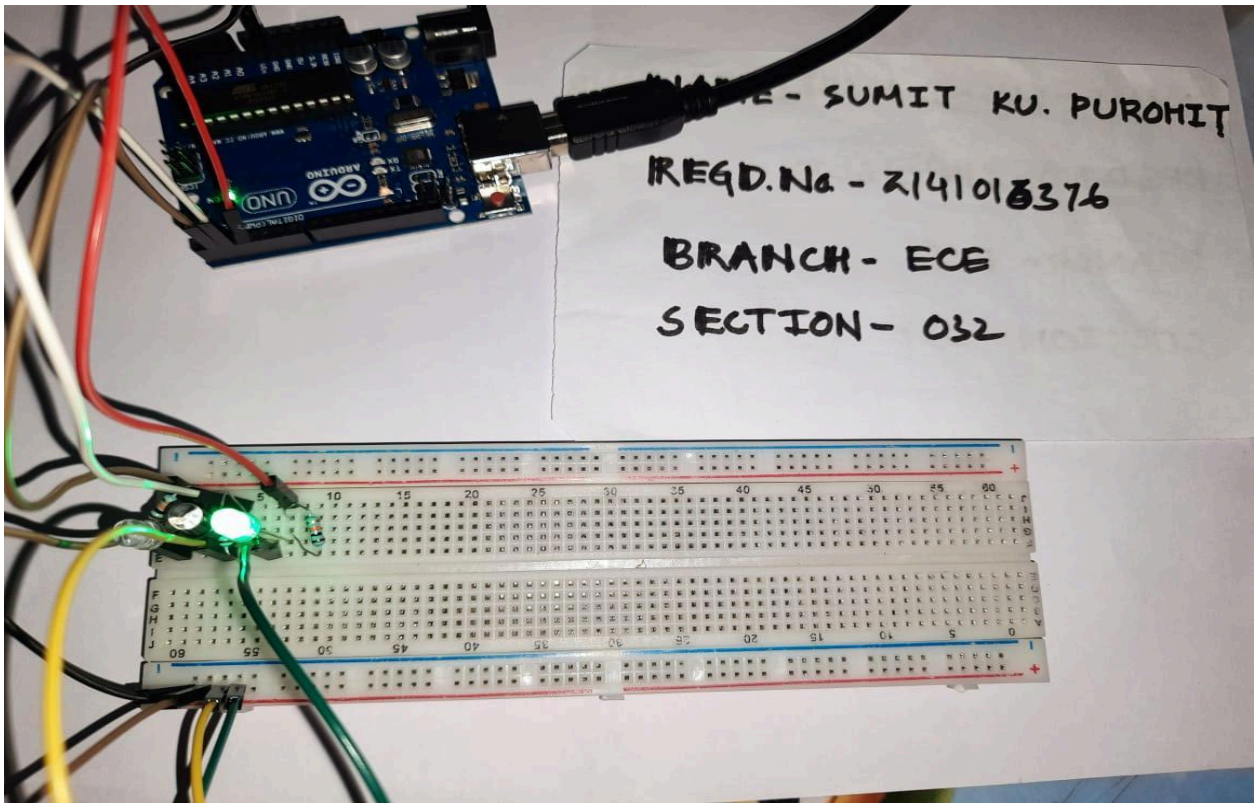


Fig11

In lane two, the anode legs of red led is connected to resistor and resistor is connected to D4 of Arduino. The cathode Red LED is connected with ground.

In lane two, the anode legs of yellow led is connected to resistor and resistor is connected to D5 of Arduino. The cathode Yellow LED is connected with ground.

In lane two, the anode legs of green led is connected to resistor and resistor is connected to D6 of Arduino. The cathode Green LED is connected with ground.

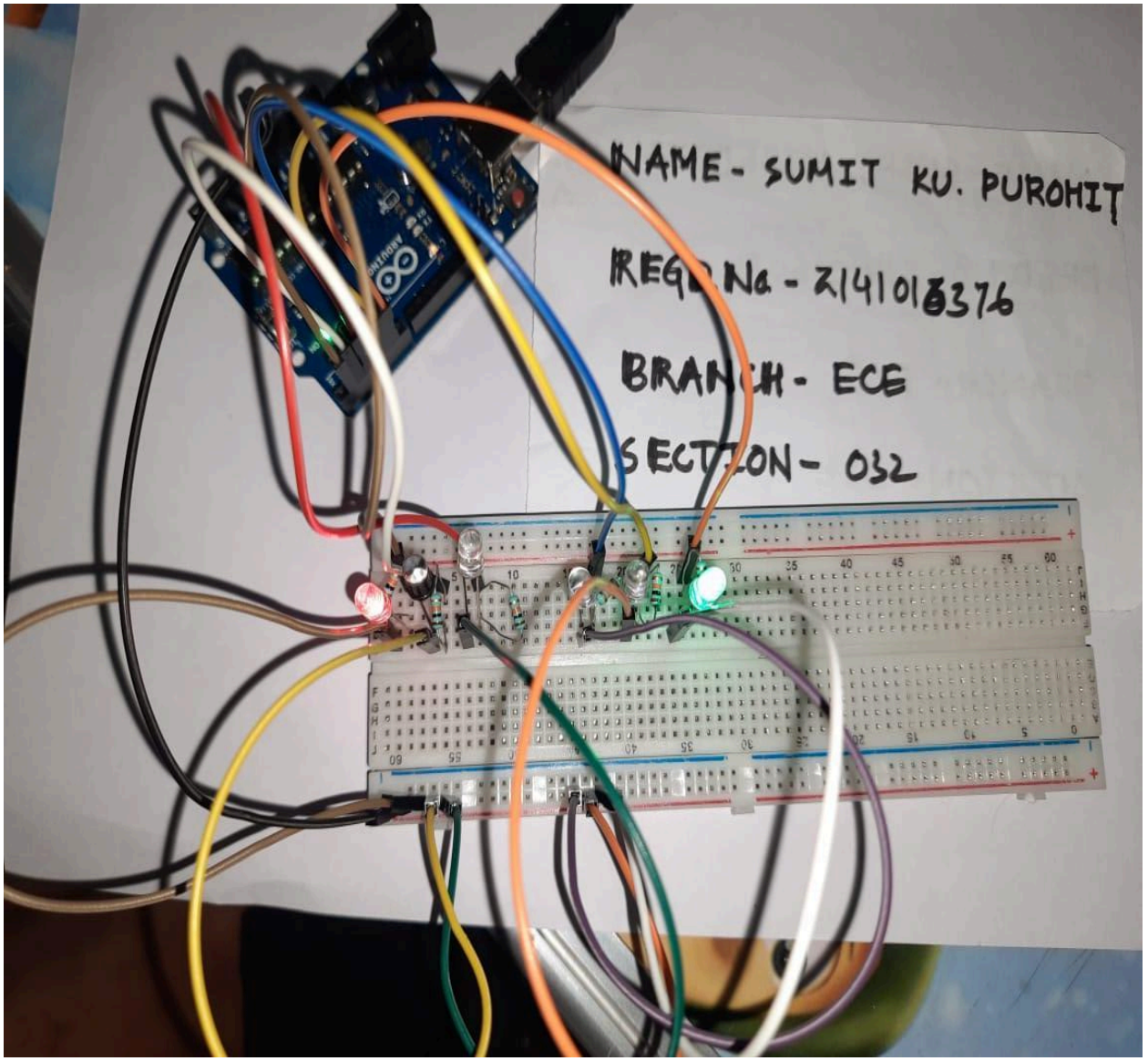


Fig12

In lane three, the anode legs of red led is connected to resistor and resistor is connected to D7 of Arduino. The cathode Red LED is connected with ground.

In lane three, the anode legs of yellow led is connected to resistor and resistor is connected to D8 of Arduino. The cathode Yellow LED is connected with ground.

In lane three, the anode legs of green led is connected to resistor and resistor is connected to D9 of Arduino. The cathode Green LED is connected with ground.

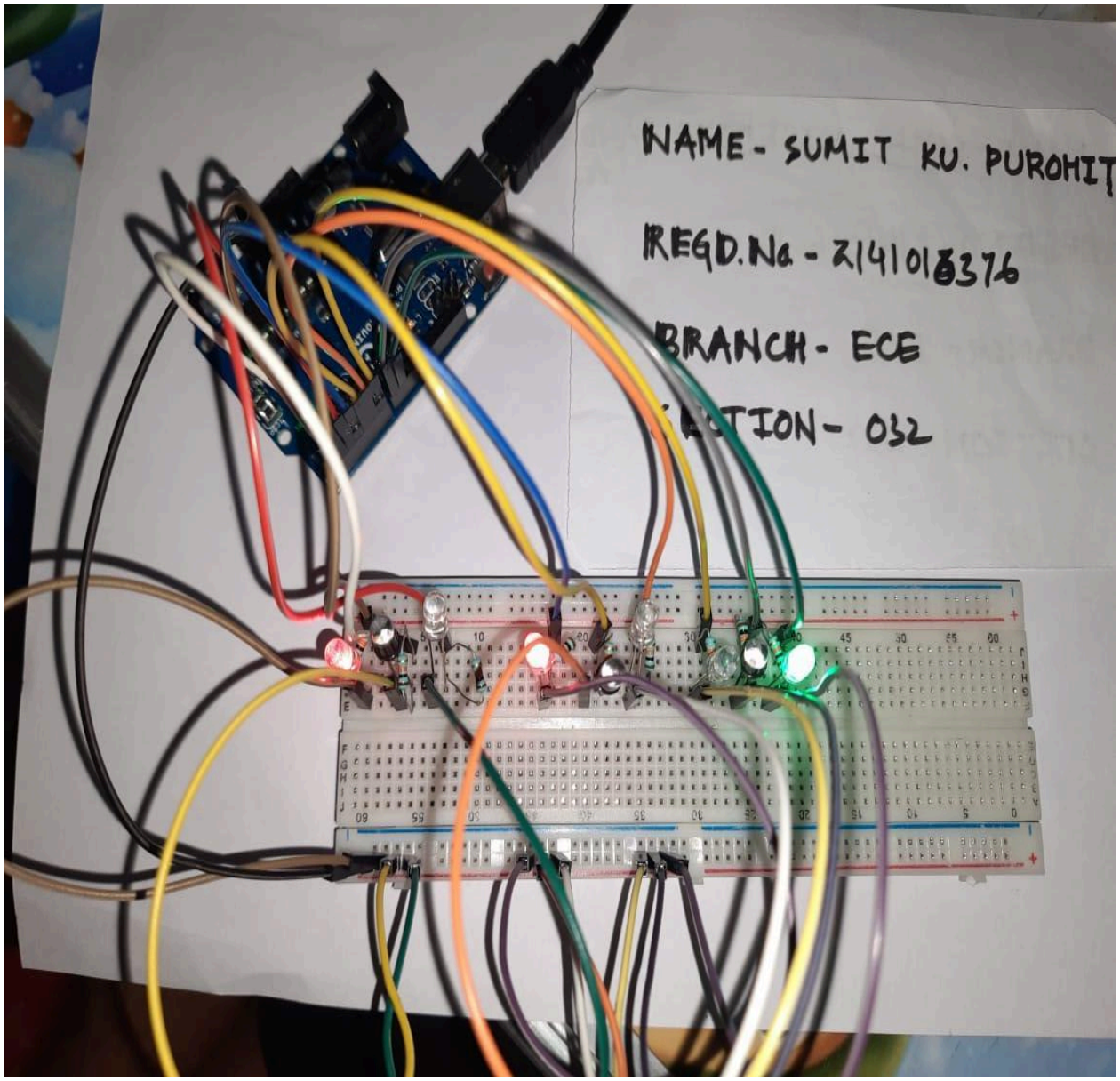


Fig13

In lane four, the anode legs of red led is connected to resistor and resistor is connected to D10 of Arduino. The cathode Red LED is connected with ground.

In lane four, the anode legs of yellow led is connected to resistor and resistor is connected to D11 of Arduino. The cathode Yellow LED is connected with ground.

In lane four, the anode legs of green led is connected to resistor and resistor is connected to D12 of Arduino. The cathode Green LED is connected with ground

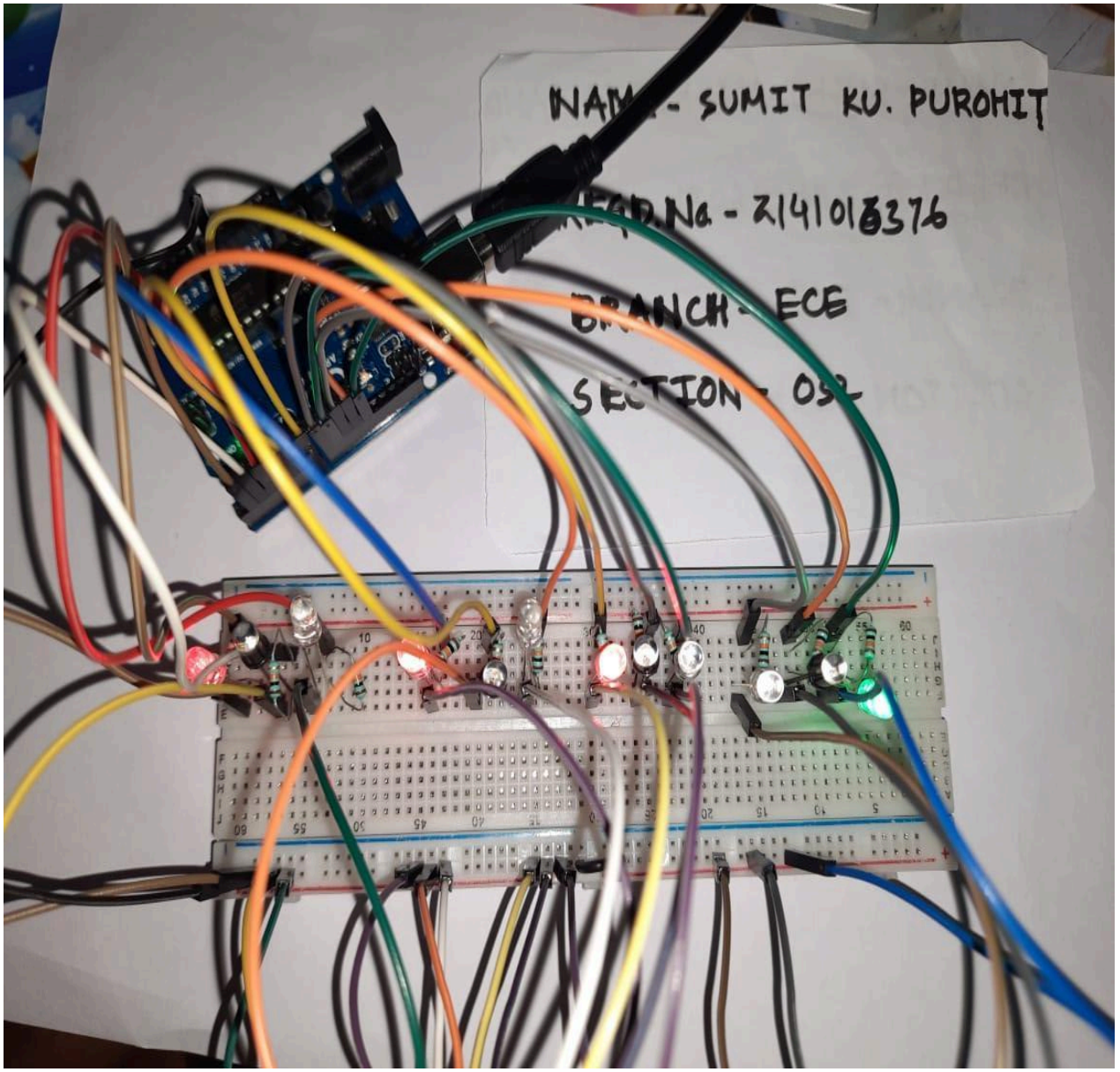


Fig14

*RESULT: -

We have exhibited traffic light for a four-way road. The LEDs will be glowing in a particular sequence to form an actual traffic light control system. At a time three red

LEDs will glow and one green LED will be ON. Yellow LEDs will glow on each transition between red and green LED. Red LED will glow for 5000 millisecond, yellow LED will glow for 2000 millisecond and Green LED will glow for 5000 milliseconds.

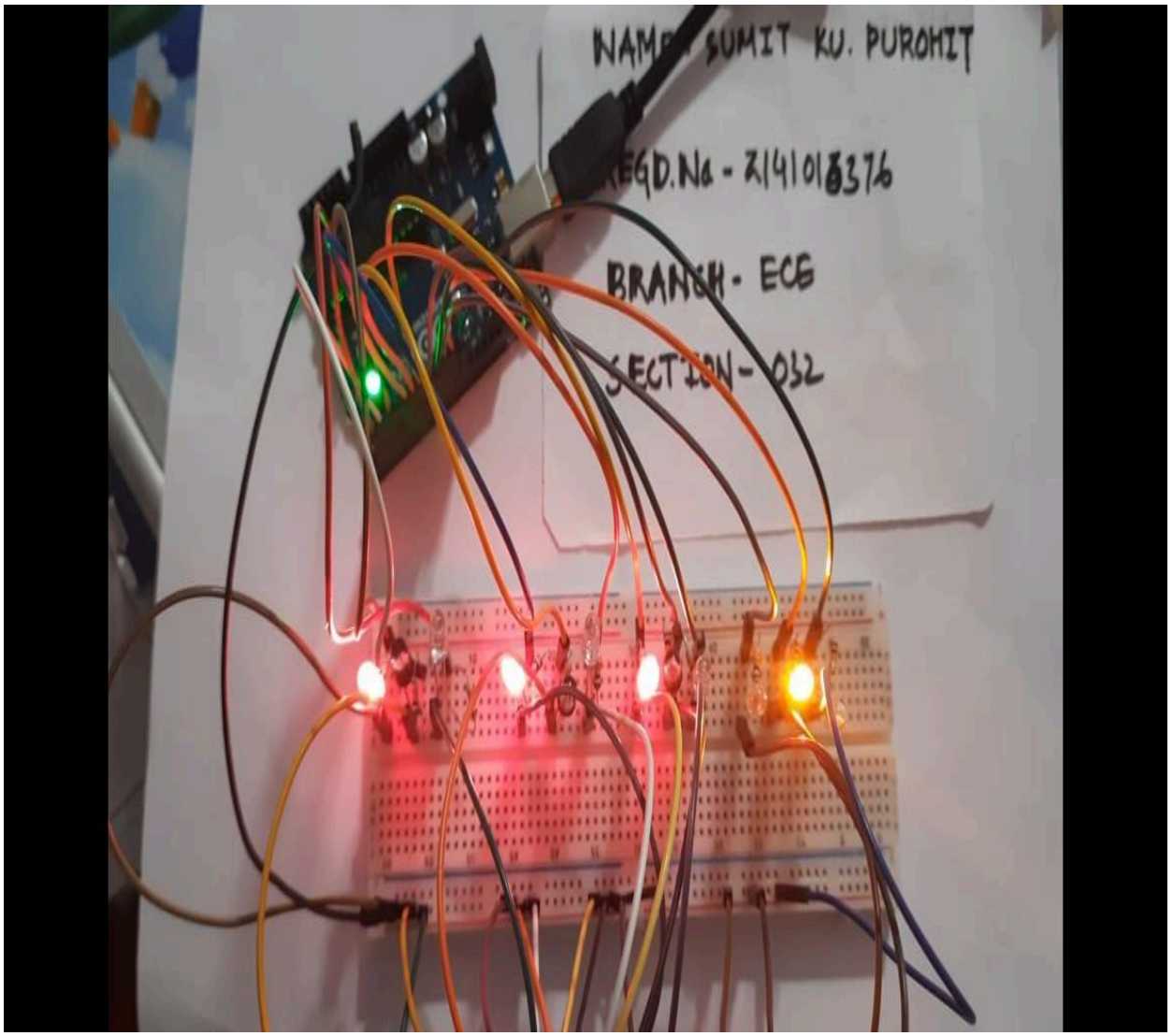


Fig15

TRANSITION PERIOD BETEWEN RED TO GREEN

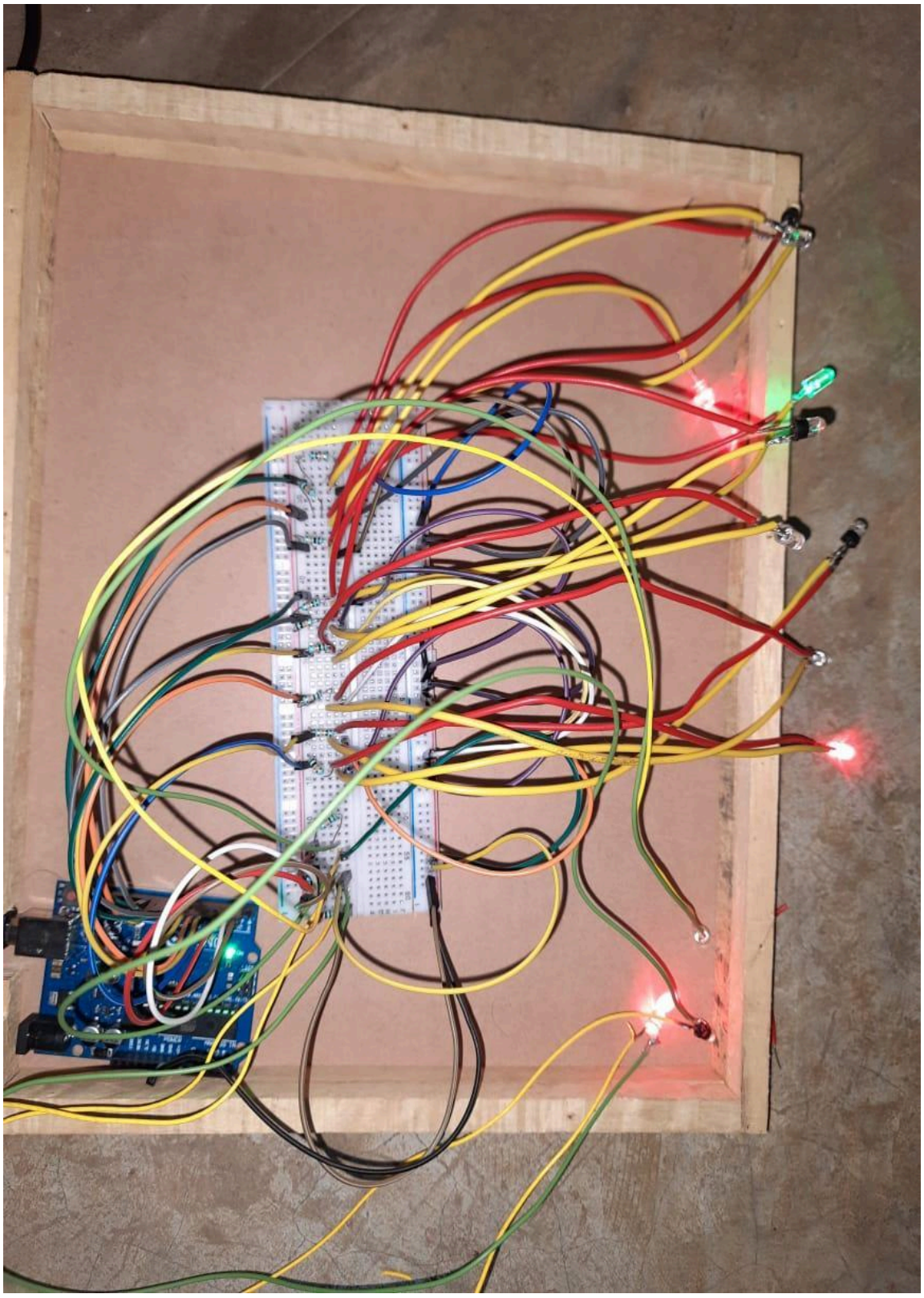


Fig16

LED GLOW UP FOR CLOSING THREE WAYS AND OPEN ONE WAY

*PREPARATION OF PHYSICALPROTOTYPE: -

(STEP BY STEP PROTOTYPE IMPLEMENTATION PHOTOGRAPHS)

Item required to make the prototype

1 Hard Board
2 An Arduino
3 LED bulbs (Red, Yellow and green)
4 Resistance
5 Breadboard
6 Male to Male jumper wire
7 Soldering Iron with solder and flux
8 Four pens as traffic post pillar
9 Metal cut piece for fixing the LEDs
10 Artificial Grass
11 Connecting wires (Red, Yellow and green)
12 House
13 Colour paper & Glue
14 One toy bus and car

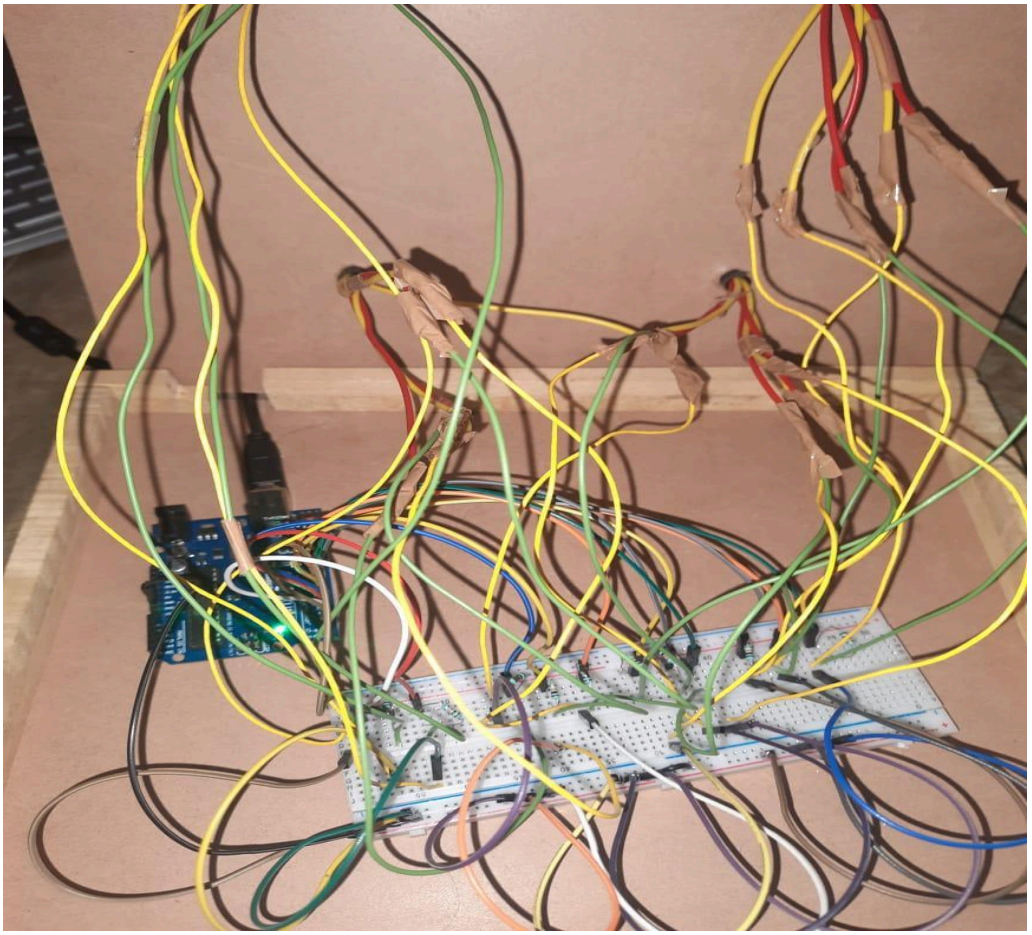


Fig17

I cut one side of the Hard board to fix the Arduino for connecting with power supply. I fix the breadboard in the base of the hard board. By cutting the legs of LED I make it shorter and soldered the connecting wires. The LEDs are fixed on a small metal piece with three holes. The connecting wires are routed through four black pens. I put the wire in the breadboard. I put 12 resistances in between the supply source of Arduino and individual LED.

Fig18

Before giving the power source

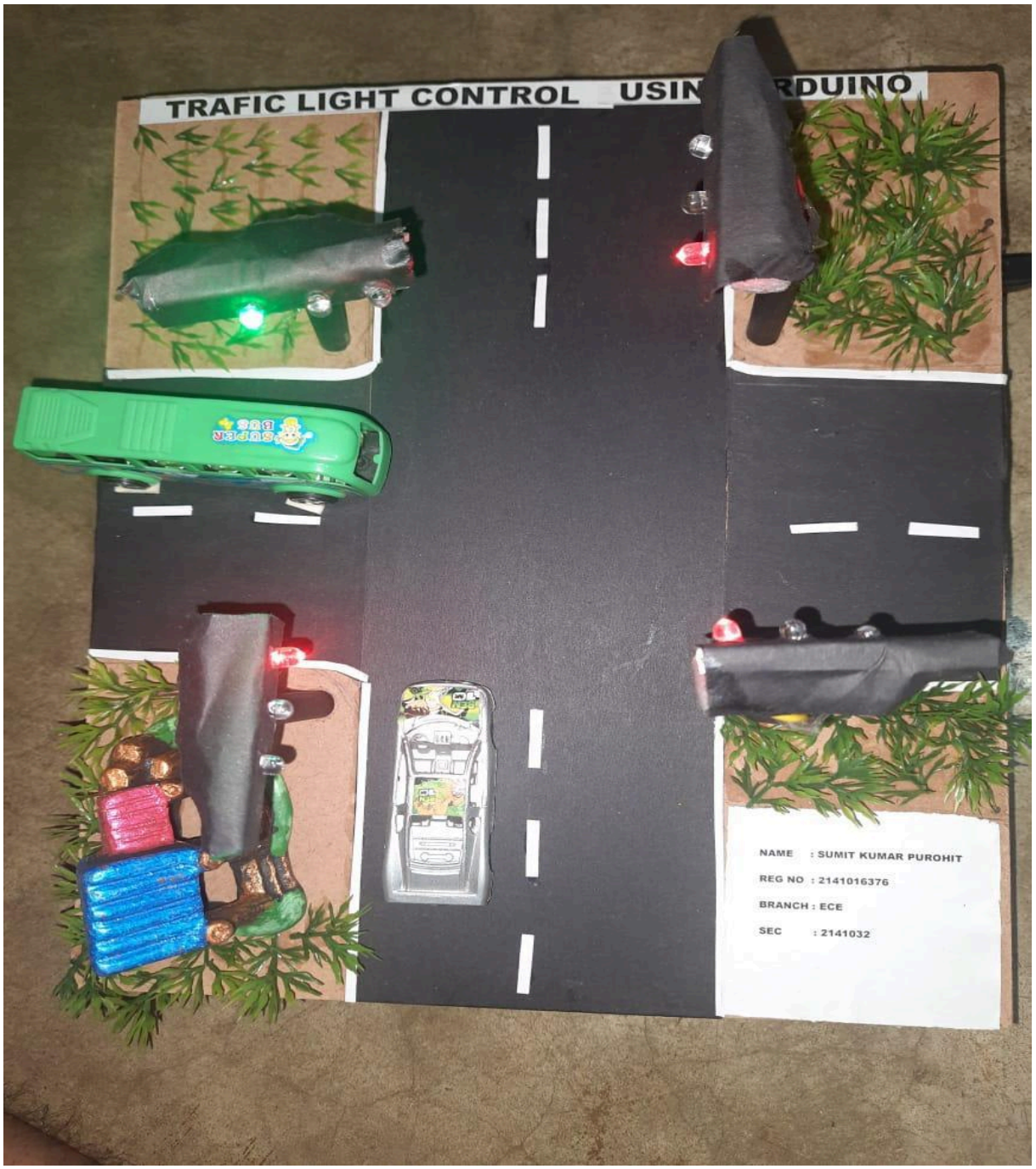


Fig19

After giving the power source

Physical prototype in four different angles: -

Fig20


NAME : SUMIT KUMAR PUROHIT
REG NO : 21410116376
BRANCH : ECE
SEC : 21410I32
HT CONTROL

Fig21


RAFIC LIGHT CONTROL USING ARDUINO

Fig22

Fig23


TRAFIC LIGHT CONTROL USING
NAME : SUMIT KUMAR PUROHIT
REG NO : 2141016376
BRANCH : ECE
SEC : 2141032

Fig24

It is the duty of every citizen to obey the traffic rule for the safety of self and others. I make the circuit so that at a time one lane is free to move by stopping other three lanes with red light. between red and green light, yellow light for waiting period. It is better to obey the traffic rules for accident-free driving .

CONCLUSION

We have designed full system of Traffic-Light Controller by implementing the logic of block diagram of the system and could simulate the result on LED using ARDUINO and switchover of LED during this counter interval. Instead of a manual traffic signal, we could go for an autonomous traffic light simulator, where the controller on each crossway will determine the duration of signal lights for the traffic of that particular road. The controller, on a time interval will trigger the control and ensure that the green signal is given for the required duration. A simulation-based traffic light controller system is good for analyzing and making necessary modifications. Further development would be done on the traffic control system to become a more suitable and less costly model.

REFERENCES

- Wikipedia
- Electronics.com
- Elprocus.com
- www.arduion.cc