CSE331: Microprocessor Interfacing & Embedded System

Summer 2022

Project Report

Section: 4

Faculty: Dr. Dihan Md. Nurudddin Hasan (DMH)

Group - 7

Aporbo Ghosh	1931458042
Ayesha Rahman	1931209042
Tahira Kabir	1911269642
Sumaia Rahman	1430058042
Md.Amrin Ibna Hasnath	1922057042
Siaam Ibn Ali	1931863642
Dewan Fatin Hasnat	1921227642

Department of Electrical and Computer Engineering

North South University

Contents

Question:

Implement the given encryption table using a microcontroller. Use single pole, double throw switch to configure the inputs for high and low conditions. Use LEDs to represent the corresponding output statuses

Input				Output			
I3	I2	I1	I0	O3	O2	O2	O1
0	0	0	0	1	1	0	0
1	0	0	0	1	1	0	1
0	1	0	0	0	1	1	0
1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	1
1	0	1	0	1	0	1	0
0	1	1	0	1	0	1	1
1	1	1	0	0	0	0	0
0	0	0	1	0	0	0	1
1	0	0	1	0	0	1	1
0	1	0	1	1	0	1	0
1	1	0	1	1	1	1	0
0	0	1	1	0	1	0	0
1	0	1	1	1	1	0	0
0	1	1	1	1	1	1	0
1	1	1	1	1	1	0	0

1. General Description:

1.1 Arduino UNO

The **Arduino Uno** is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc . The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable.

The Uno board is the first in a series of USB-based Arduino boards;it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases.[4] The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter

1.2 Arduino IDE

The software used for writing, compiling & uploading code to Arduino boards is called Arduino IDE (Integrated Development Environment). It is a cross-platform software that is available for every Operating System like Windows, Linux, macOS. This software can be used with any Arduino board. We write our code in this compiler and then connect the microcontroller via USB to upload the code in it. Arduino IDE consists of different sections such as WindowBar, MenuBar, ShortcutButtons, Text Editor, Output Panel.

1.3 Proteus 8 pro

Proteus 8 Professional is software that can be used to draw schematics, PCB layout, code, and even simulate the schematic. The Proteus Design Suite is a software suite containing schematic, simulation as well as PCB designing. Schematic Capture in the Proteus Design Suite is used for both the simulation of designs and as the design phase of a PCB layout project. It is therefore a core component and is included with all product configurations.

We simulate our circuit in Proteus 8 Pro using Arduino Uno, resistors, single pole double throw switches, LED lights.

2. Equipment

We used variety for software and hardware to finish the projects, those are

1. Arduino UNO
2. Proteus 8 Pro
3. Button switches
4. 4 x LED light
5. Jumper Wire
6. Breadboard
7. 4 x 330-ohm resistor
8. 4 x 10k ohm resistor

3. Method of Derivation:

We used the given table of Group-7 and we built it into a truth table for inputs and outputs. Then we used the concept of SOP to produce the KMAPs below and then found the logical expression for each of the outputs using KMAP.

3.1 Truth Table:

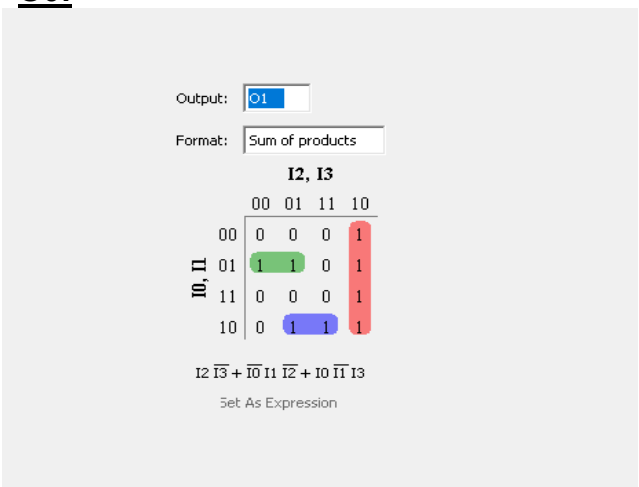
I3	I2	I1	I0	O3	O2	O2	O1
0	0	0	0	1	1	0	0
1	0	0	0	1	1	0	1
0	1	0	0	0	1	1	0
1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	1
1	0	1	0	1	0	1	0
0	1	1	0	1	0	1	1
1	1	1	0	0	0	0	0
0	0	0	1	0	0	0	1
1	0	0	1	0	0	1	1
0	1	0	1	1	0	1	0
1	1	0	1	1	1	1	0
0	0	1	1	0	1	0	0
1	0	1	1	1	1	0	0
0	1	1	1	1	1	1	0
1	1	1	1	1	1	0	0

3.2 Derived Result:

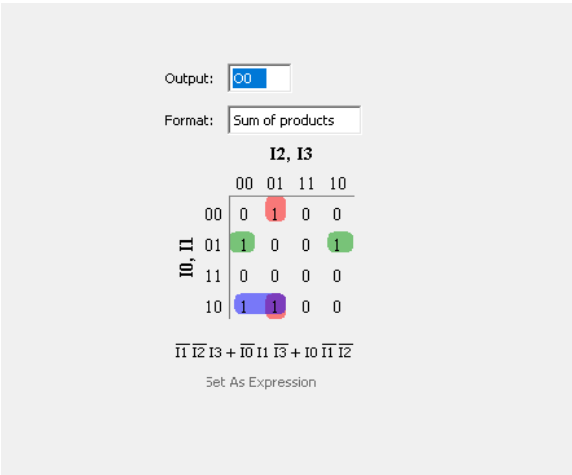
We use the K-Map to derive the expression for the output.

KMAP:

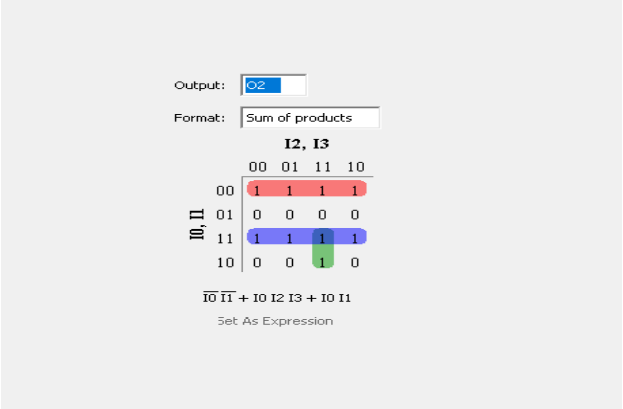
O0:



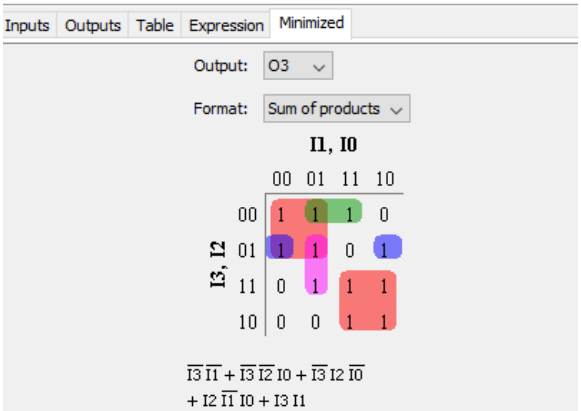
O1:



O2

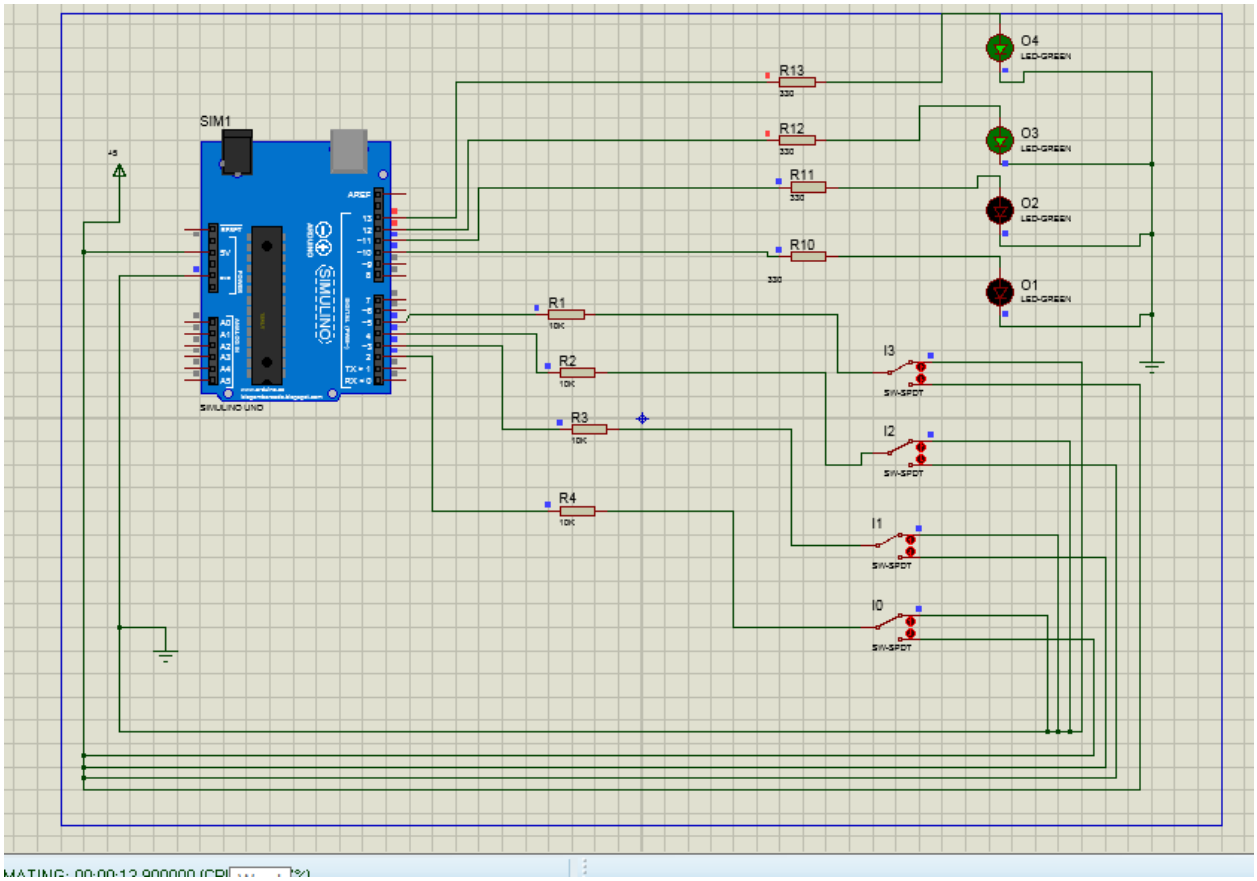


O3



4. Circuit Diagram with Values of Electrical Components

4.1 Figure 1:

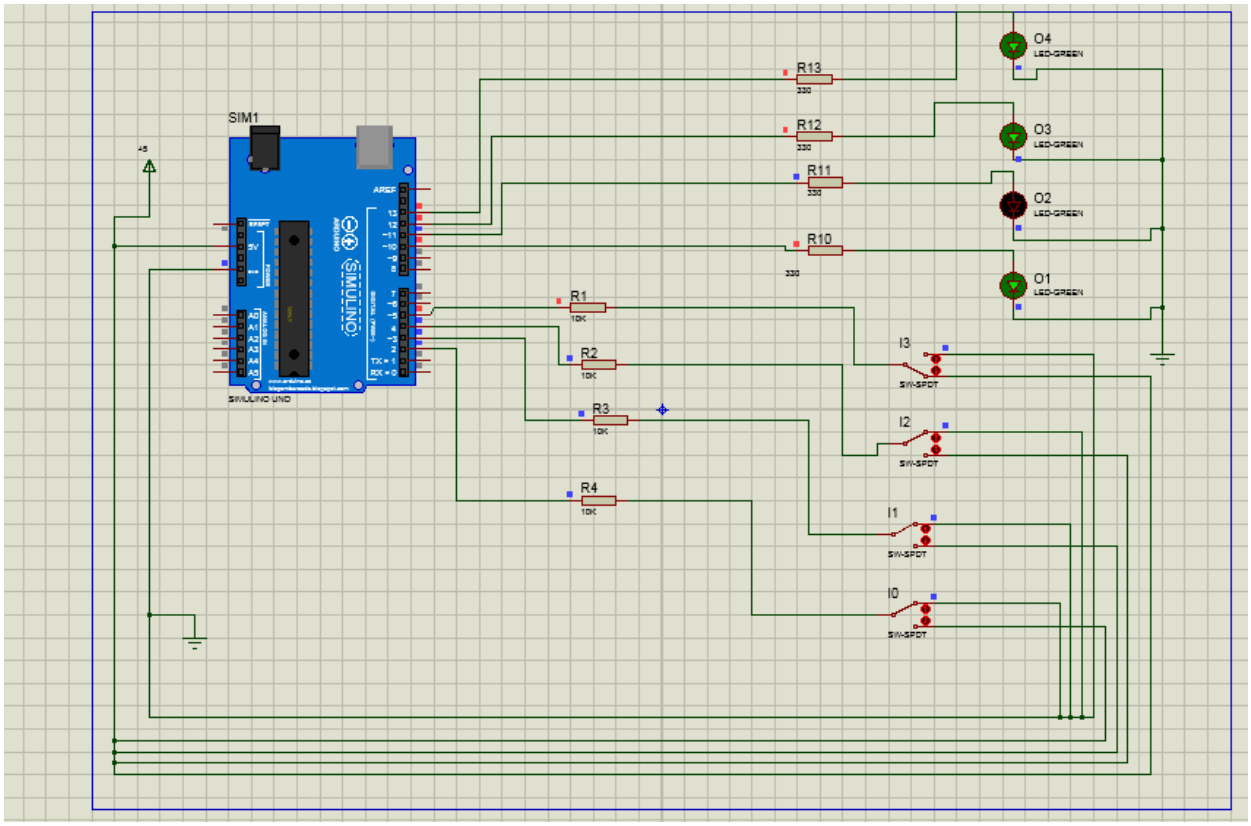


Here we can see the circuit in Proteus 8,

for the inputs are I0=1, I2=0, I1=0, I0=0 the LED outputs are O4=1, O3=1, O2=0, O1=0.

4.2 Figure 2:

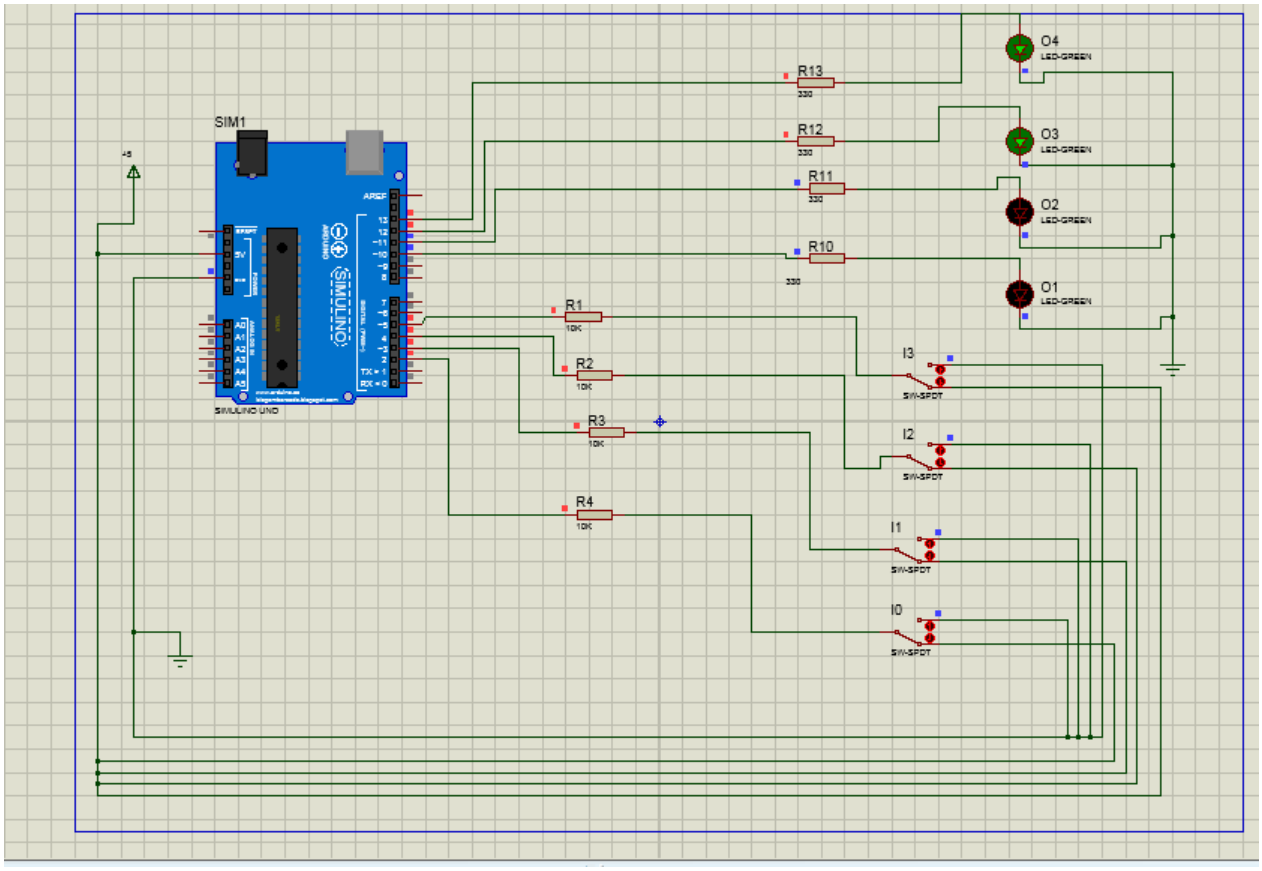
Here we can see the circuit in Proteus 8,



for the inputs are I3=1, I2=0, I1=0, I0=0 the LED outputs are O4=1, O3=1, O2=0, O1=1

4.3 Figure 3:

Here we can see the circuit in Proteus 8,



for the inputs are I3=0, I2=1, I1=0, I0=1 the LED outputs are O4=0, O3=1, O2=0, O1=1.

4.4 Proteus Simulation Video:

https://drive.google.com/drive/folders/1CYxIodUF6GvVyZXixKX6aw0uoQsR8VVB?fbclid=IwAR2CkEUx2XNmA7jY9iSTSbbXtaq3x4943RhL8XI6BVX2C3j4OFDgXkuiVg

5. Circuit Operation Principles:

Arduino Uno:

We have used Proteus 8 pro to build the hardware circuit diagram, the components used in proteus are **Arduino UNO,** this is the microcontroller we used in the simulation.

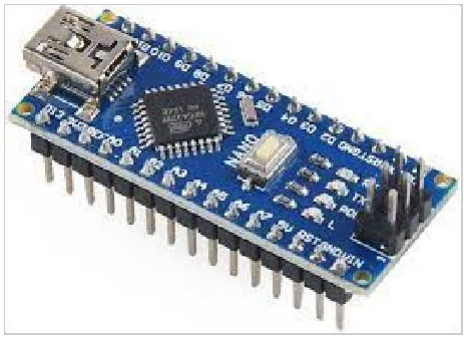


Fig. An Arduino Uno

Button Switch: We have used button switches to connect with Arduino pins. Then we connected the wire with the circuit.

Fig. A button Switch

Resistors: We used 4 10K -ohm resistors to connect with the switches and 4 330 -ohm resistors to connect with the LED outputs.

Fig. A 10K Ω Resistor

Fig. A 330 Ω Resistor

LEDs: We connected 4 LED outputs O0, O1, O2, O3 to Arduino Nano pin no D12, D11, D10, D9 respectively.

After building the circuit we uploaded the Arduino code to the Nano and simulated the project. We uploaded a video on Drive to demonstrate the Proteus simulation of the circuit.

Fig. LED

6. Arduino Program Code

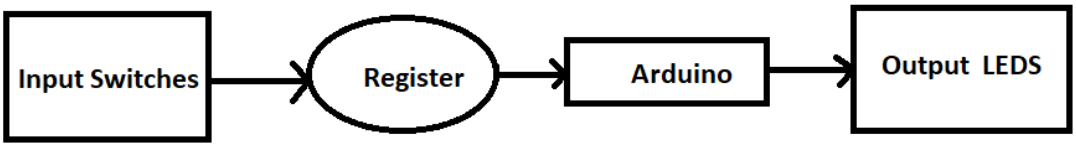
```
const int I0 = 2;
const int I1 = 3;
const int I2 = 4;
const int I3 = 5;

const int LED1 = 10;
const int LED2 = 11;
const int LED3 = 12;
const int LED4 = 13;

int BUTTONstate0 = 0;
int BUTTONstate1 = 0;
int BUTTONstate2 = 0;
int BUTTONstate3 = 0;

void setup()
{
  pinMode(I0, INPUT);
  pinMode(I1, INPUT);
  pinMode(I2, INPUT);
  pinMode(I3, INPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
}

void loop()
{
  BUTTONstate0 = digitalRead(I0);
  BUTTONstate1 = digitalRead(I1);
  BUTTONstate2 = digitalRead(I2);
  BUTTONstate3 = digitalRead(I3);

  if (BUTTONstate0 == LOW && BUTTONstate1 == LOW && BUTTONstate2 == LOW
&& BUTTONstate3 == LOW) { //1
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
  }
  else if(BUTTONstate0 == LOW && BUTTONstate1 == LOW && BUTTONstate2 ==
LOW && BUTTONstate3 == HIGH) {//2
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
```

```
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
  }
  else if(BUTTONstate0 == LOW && BUTTONstate1 == LOW && BUTTONstate2 ==
HIGH && BUTTONstate3 == LOW) {//3
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW);
  }
  else if(BUTTONstate0 == LOW && BUTTONstate1 == LOW && BUTTONstate2 ==
HIGH && BUTTONstate3 == HIGH) {//4
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
  }
  else if (BUTTONstate0 == LOW && BUTTONstate1 == HIGH && BUTTONstate2 ==
LOW && BUTTONstate3 == LOW) {//5
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, HIGH);
  }
  else if (BUTTONstate0 == LOW && BUTTONstate1 == HIGH && BUTTONstate2 ==
LOW && BUTTONstate3 == HIGH) {//6
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, HIGH);
  }
  else if (BUTTONstate0 == LOW && BUTTONstate1 == HIGH && BUTTONstate2 ==
HIGH && BUTTONstate3 == LOW) {//7
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, HIGH);
  }
  else if (BUTTONstate0 == LOW && BUTTONstate1 == HIGH && BUTTONstate2 ==
HIGH && BUTTONstate3 == HIGH) {//8
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
  }
  else if (BUTTONstate0 == HIGH && BUTTONstate1 == LOW && BUTTONstate2 ==
LOW && BUTTONstate3 == LOW) {//9
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
  }
  else if (BUTTONstate0 == HIGH && BUTTONstate1 == LOW && BUTTONstate2 ==
```

```
LOW && BUTTONstate3 == HIGH) {//10
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
  }
  else if (BUTTONstate0 == HIGH && BUTTONstate1 == LOW && BUTTONstate2 ==
HIGH && BUTTONstate3 == LOW) {//11
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, HIGH);
  }
  else if (BUTTONstate0 == HIGH && BUTTONstate1 == LOW && BUTTONstate2 ==
HIGH && BUTTONstate3 == HIGH) {//12
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
  }
  else if (BUTTONstate0 == HIGH && BUTTONstate1 == HIGH && BUTTONstate2
== LOW && BUTTONstate3 == LOW) {//13
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW);
  }
  else if (BUTTONstate0 == HIGH && BUTTONstate1 == HIGH && BUTTONstate2
== LOW && BUTTONstate3 == HIGH) {//14
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
  }
  else if (BUTTONstate0 == HIGH && BUTTONstate1 == HIGH && BUTTONstate2
== HIGH && BUTTONstate3 == LOW) {//15
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
  }
  else if (BUTTONstate0 == HIGH && BUTTONstate1 == HIGH && BUTTONstate2
== HIGH && BUTTONstate3 == HIGH) {//16
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
  }
  else {
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
```

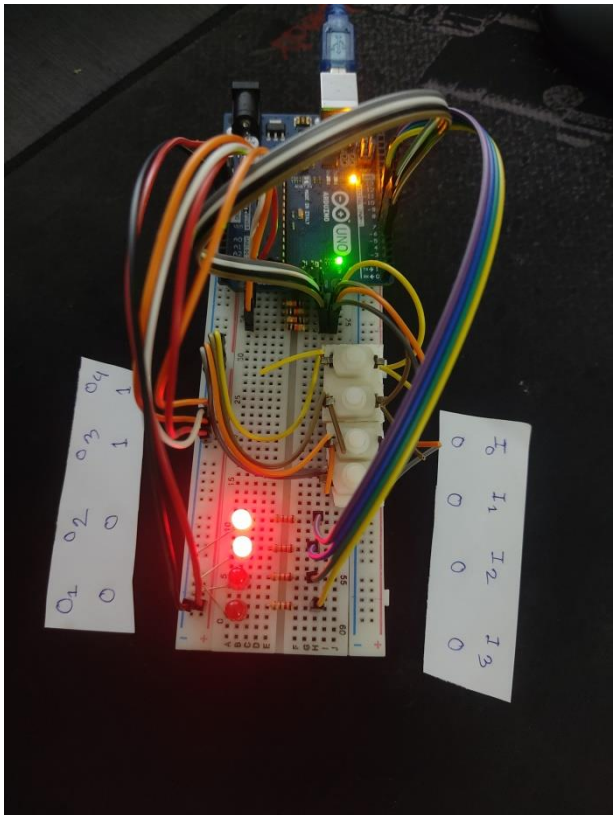
```
}

}
```

7. Proteus flow chart



8. Hardware Implementation

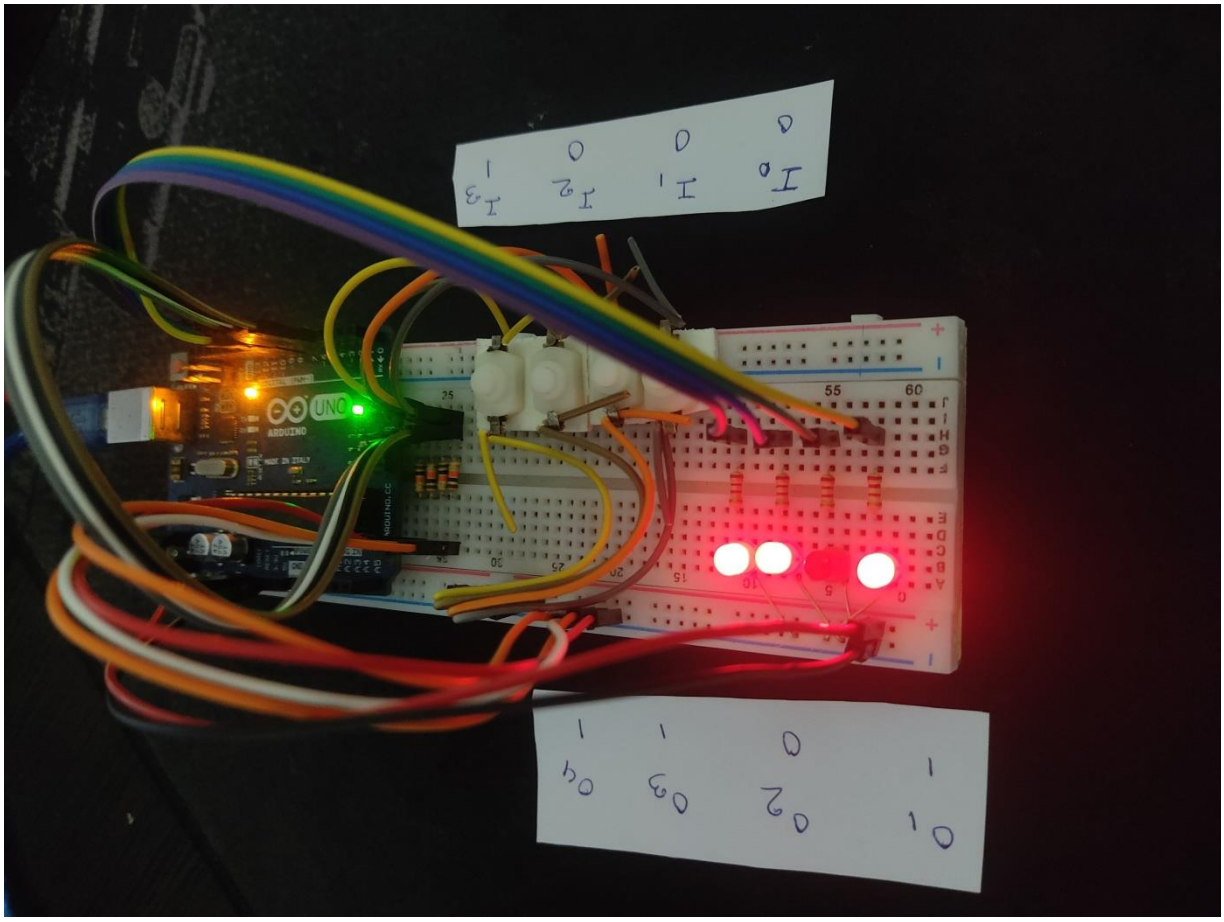
8.1 Figure 1:



Here we can see the circuit,

for the inputs are I3=0, I2=0, I1=0, I0=0 the LED outputs are O4=1, O3=1, O2=0, O1=0.

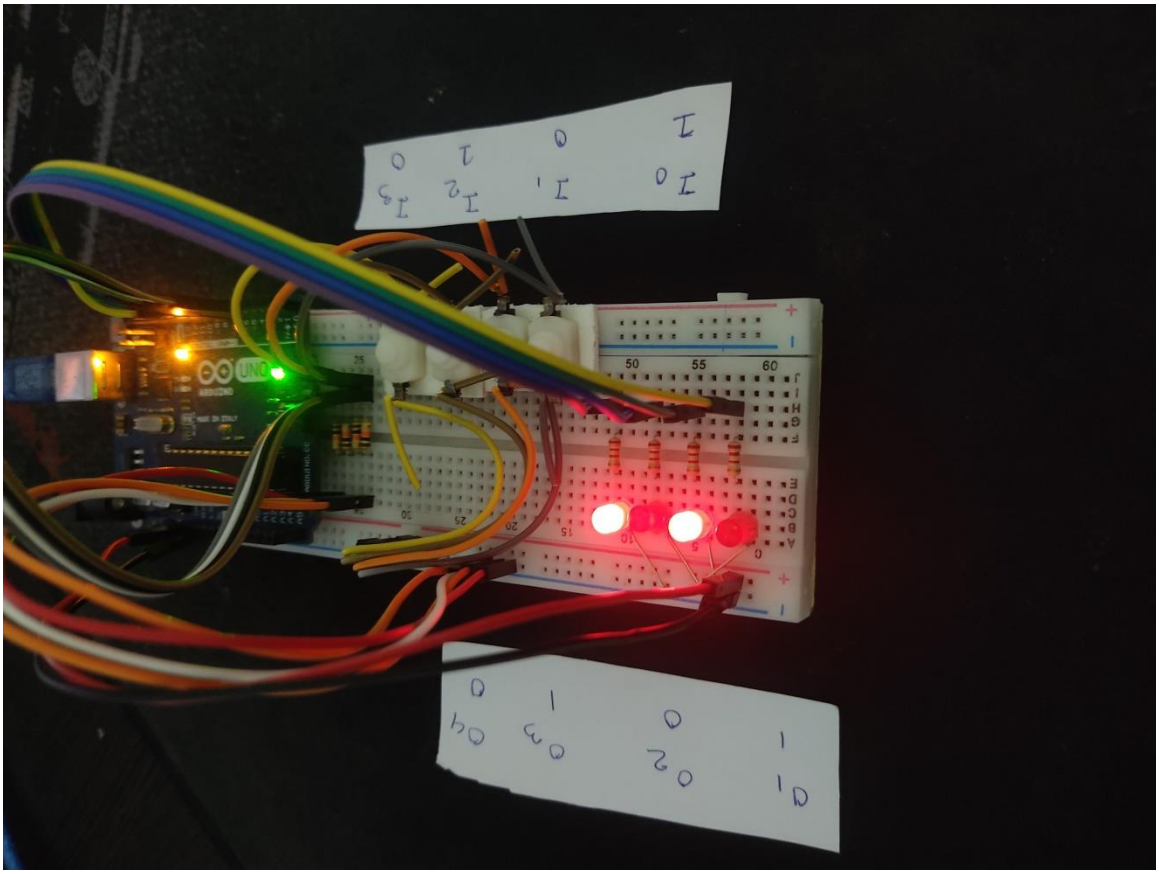
8.2 Figure 2:



Here we can see the circuit,

for the inputs are I3=1, I2=0, I1=0, I0=0 the LED outputs are O4=1, O3=1, O2=0, O1=1.

8.3 Figure 3:



Here we can see the circuit,

for the inputs are I3=0, I2=1, I1=0, I0=1 the LED outputs are O4=0, O3=1, O2=0, O1=1.

8.4 Hardware Implementation Video:

https://drive.google.com/drive/folders/1CYxIodUF6GvVyZXixKX6aw0uoQsR8VVB?fbclid=IwAR15AuDDMgMDwiBhhZeNmKmrSOzcONP4C5O809mdTr9umyqRqqMBgFeIthI

9. Questions and answers:

* What is the clock frequency of the microcontroller used?
Ans : The Clock frequency of the microcontroller used is 16 MHz

* What is the data bus width of the microcontroller used ?
Ans: The width of the microcontroller's data bus is 8 bit

* What is the size of your hex file generated ?
Ans: The generated hex file is 4 KB in size and has been used in the project.

The hexacode are given below:

:100000000C9461000C9473000C9473000C947300B6
:100010000C9473000C9473000C9473000C94730094
:100020000C9473000C9473000C9473000C94730084
:100030000C9473000C9473000C9473000C94730074
:100040000C9426010C9473000C9473000C947300B0
:100050000C9473000C9473000C9473000C94730054
:100060000C9473000C947300000000240027001F
:100070002A0000000000250028002B0000000000DE
:10008000230026002900040404040404040202DA
:10009000020202020303030303030301020408102007
:1000A000408001020408102001020408102000012
:1000B000000800020100000304070000000000027
:1000C000000011241FBECFEFD8E0DEBFCDBF21E07E
:1000D000A0E0B1E001C01D92AF30B207E1F70E948D
:1000E00070010C9484020C940000833081F028F499
:1000F000813099F08230A9F008958730A9F08830D6
:10010000C9F08430B1F4809180008F7D03C080916C
:1001100080008F7780938000089584B58F7784BDA9
:10012000089584B58F7DFBCF8091B0008F77809349
:10013000B00008958091B0008F7DF9CFCF93DF9309
:10014000282F30E0F901E255FF4F8491F901E6567E
:10015000FF4FD491F901EA57FF4FC491CC23A1F08E
:1001600081110E947500EC2FF0E0EE0FFF1FE458A4

:10017000FF4FA591B491EC91ED2381E090E009F45B
:1001800080E0DF91CF91089580E090E0FACF1F9357
:10019000CF93DF93282F30E0F901E255FF4F849190
:1001A000F901E656FF4FD491F901EA57FF4FC49188
:1001B000CC23A9F0162F81110E947500EC2FF0E0DE
:1001C000EE0FFF1FEE58FF4FA591B4918FB7F89433
:1001D000EC91111108C0D095DE23DC938FBFDF9125
:1001E000CF911F910895DE2BF8CFCF93DF9390E04E
:1001F000FC01E656FF4F24918A579F4FFC018491E2
:100200008823D1F090E0880F991FFC01E859FF4F37
:10021000A591B491FC01EE58FF4FC591D4916111A5
:100220000EC09FB7F8948C91E22FE0958E238C93AB
:100230002881E223E8839FBFDF91CF9108958FB794
:10024000F894EC91E22BEC938FBFF6CF1F920F92B4
:100250000FB60F9211242F933F938F939F93AF93D9
:10026000BF9380910B0190910C01A0910D01B09171
:100270000E0130910A0123E0230F2D3758F5019626
:10028000A11DB11D20930A0180930B0190930C01D5
:10029000A0930D01B0930E0180910601909107018A
:1002A000A0910801B09109010196A11DB11D809393
:1002B000060190930701A0930801B0930901BF9133
:1002C000AF919F918F913F912F910F900FBE0F9003
:1002D0001F90189526E8230F0296A11DB11DD2CFBD
:1002E000789484B5826084BD84B5816084BD85B511
:1002F000826085BD85B5816085BD80916E0081601D
:1003000080936E0010928100809181008260809 3C2
:10031000810080918100816080938 1008091800 0C4
:1003200081608093800080 91B10084608093B100EF
:100330008091B00081608093B00080917A008460E9
:1003400080937A0080917A00826080937A00809115
:100350007A00816080937A0080917A00806880932F
:100360007A001092C10060E082E00E94F50060E037
:1003700083E00E94F50060E084E00E94F50060E008
:1003800085E00E94F50061E08AE00E94F50061E0EE
:100390008BE00E94F50061E08CE00E94F50061E0D6
:1003A0008DE00E94F50080E0C82E80E0D82E82E02B
:1003B0000E949E007C0190930501809304018 3E0DC

```
:1003C0000E949E008C01909303018093020184E0BF
:1003D0000E949E00EC0190930101809300185E052
:1003E0000E949E00E114F10409F058C001151105A6
:1003F00009F5209781F4009751F460E08AE00E94AB
:10040000C70060E08BE00E94C70061E023C0019755
:1004100071F561E0F3CF219751F5009709F070C0B5
:1004200060E08AE00E94C70061E08BE00E94C700A4
:1004300061E026C001301105D1F42097A1F40097A6
:1004400071F461E08AE00E94C70061E08BE00E94E5
:10045000C70060E08CE00E94C70061E015C0019712
:1004600031F460E0EFCF219711F4892B51F360E074
:100470008AE00E94C70060E08BE00E94C70060E055
:100480008CE00E94C70060E08DE00E94C700C114AC
:10049000D10409F48CCF0E94000089CFEA94EF28A0
:1004A00031F701151105D1F4209761F4009711F48B
:1004B00061E0DECF0197D9F661E08AE00E94C700D3
:1004C00061E0DACF219799F6009759F2019779F612
:1004D00060E08AE00E94C70061E094CF013011051E
:1004E00031F6209741F4009759F460E08AE00E94C9
:1004F000C70060E09ACF219709F0B9CF009741F388
:0C050000019709F0B4CF79CFF894FFCF39
:00000001FF
```

* Can the project be implemented by using interrupt?

Ans: Yes, interrupt can be used to implement the project. It is mainly used by application systems to support hardware clocks, data transportation and communication devices.

* Is the main routine required to be an infinite loop? provide an explanation in favor of your answer.

Ans: Yes, the main routine had to run in an infinite loop, as the value was not updated when it is not looped simultaneously.

* Is there any difference between level triggered and edge triggered operation for the given project?

Ans: Yes, the main difference between edge and level triggering is that the yield of consecutive circuit changes between high and low voltage cycles in terms of edge activating.

* Is the project referring to encryption or decryption from input to output?

Ans: The project refers to encryption from input to output as the main motive of encryption is to convert a message by changing it into a code. The project accomplished this by encryption.

10. References:

- **Introduction to Arduino Uno**
- **Arduino IDE Tutorial**
- **Proteus 8 Introduction**