North South University
Department of Electrical Engineering and Computer Science

Group - 09 Project Report

Project Title: **Implementation of an Encryption Table Using Microcontroller**

Faculty Name: Dr. Dihan Md Nuruddin Hasan (DMH)
Course Code: CSE 331, **Section-**02, **Group-**09

Course Title: Microprocessor interfacing and embedded system

Submitted by:

Name	ID
Monjurul Aziz Fahim	1913080642
Al-Jubayer Pial	2012972042
MD Ariful Hasan Suvo	1913087042
Humayra Akhter	1831429642

Table of Contents

CHAPTER 01 6

Abstract 6

General description 6

CHAPTER 02 11

Equipment 11

2.1 For Hardware 11

2.2 For Software 11

CHAPTER 03 13

3.1 Experimental Planning: 13

3.1.1 Experimental Circuit Setup: 13

3.1.2 Method of Derivation and Results 14

3.1.3 Theoretical observation 14

3.1.4 Truth Table 14

3.1.5 K-Mapping 15

3.1.6 Derived Results 17

3.1.7 Expressions 17

CHAPTER 04 18

Logisim Circuit 18

Circuit Diagram with Values of Electrical Components 19

Circuit Operation Principles 23

CHAPTER 05 25

Program Flow Chart 25

Arduino Program with No Compiling Error 26

Hex Code 27

CHAPTER 06 30

Hardware Implementation 30

Hardware Implementation with Values of Electrical Components 31

CHAPTER 07 34

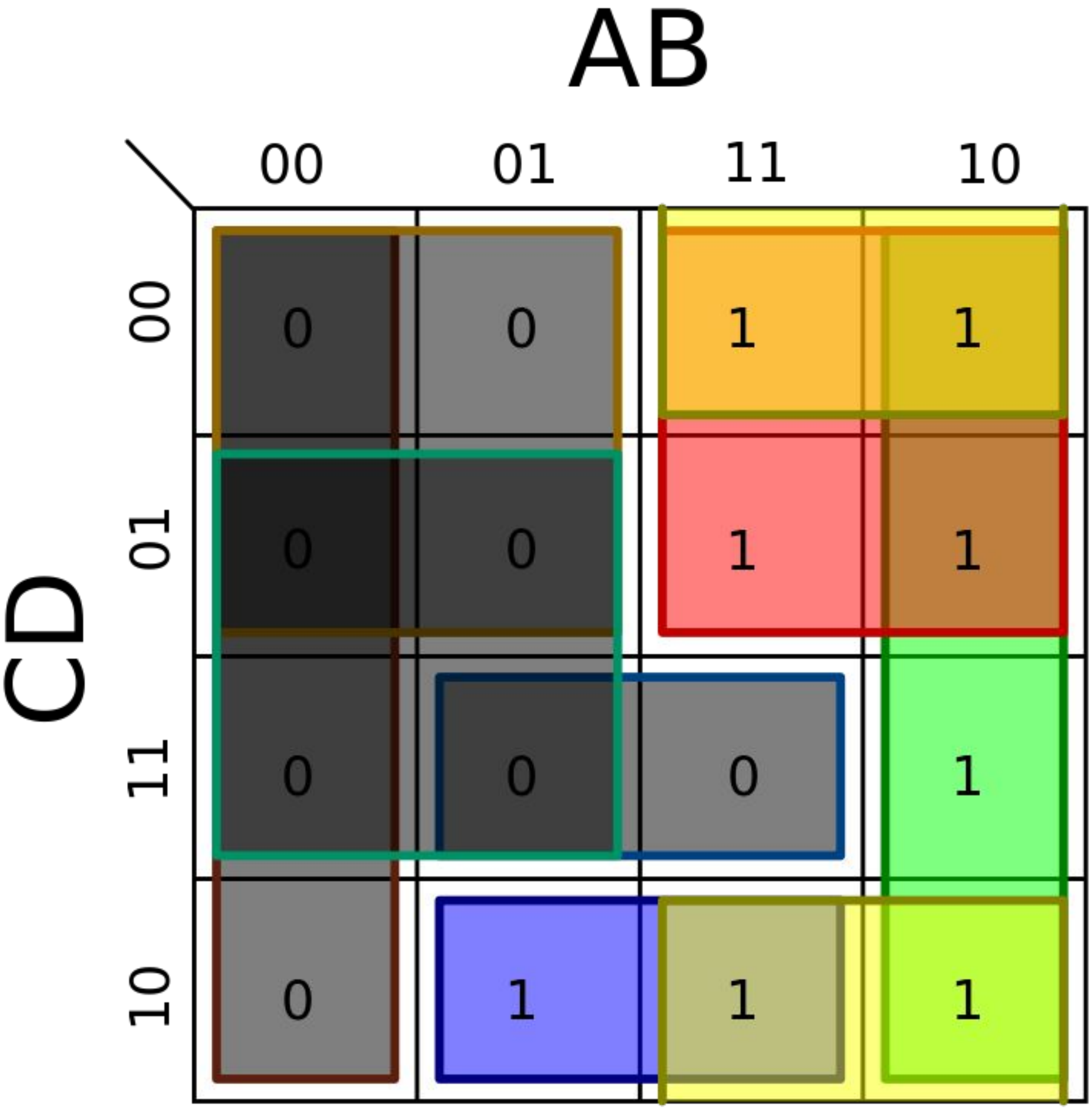
References 34

Question

Implement the given encryption table using 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	0	1	0	1
1	0	0	0	1	1	1	1
0	1	0	0	0	0	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	1	0	0
1	1	1	0	0	0	0	0
0	0	0	1	0	1	0	1
1	0	0	1	1	0	0	1
0	1	0	1	0	0	1	0
1	1	0	1	1	1	1	1
0	0	1	1	0	1	0	0
1	0	1	1	1	0	1	0
0	1	1	1	1	1	1	0
1	1	1	1	0	0	0	0

Hints for deriving the logic expression:



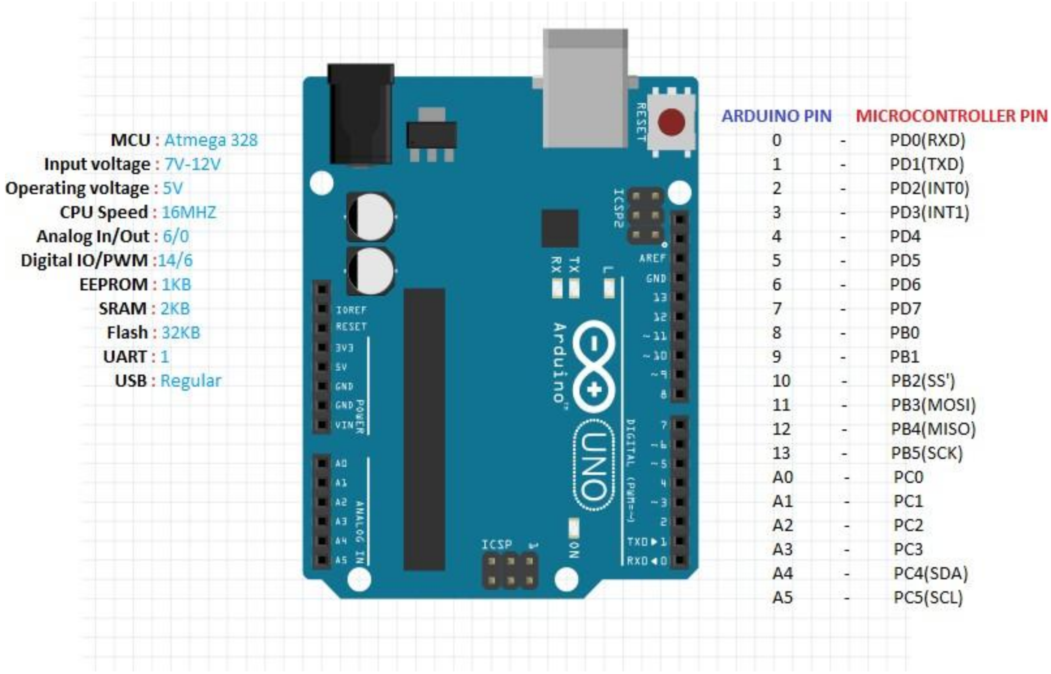
CD \ AB	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	0	1
10	0	1	1	1

$f(A,B,C,D) = E(6,8,9,10,11,12,13,14)$

$F=AC'+AB'+BCD'+AD'$

$F=(A+B)(A+C)(B'+C'+D')(A+D')$

Example microcontroller: (Arduino UNO)



MCU :	Atmega 328
Input voltage :	7V-12V
Operating voltage :	5V
CPU Speed :	16MHZ
Analog In/Out :	6/0
Digital IO/PWM :	14/6
EEPROM :	1KB
SRAM :	2KB
Flash :	32KB
UART :	1
USB :	Regular

ARDUINO PIN		MICROCONTROLLER PIN
0	-	PD0(RXD)
1	-	PD1(TXD)
2	-	PD2(INT0)
3	-	PD3(INT1)
4	-	PD4
5	-	PD5
6	-	PD6
7	-	PD7
8	-	PB0
9	-	PB1
10	-	PB2(SS')
11	-	PB3(MOSI)
12	-	PB4(MISO)
13	-	PB5(SCK)
A0	-	PC0
A1	-	PC1
A2	-	PC2
A3	-	PC3
A4	-	PC4(SDA)
A5	-	PC5(SCL)

Project requirement:

-Theoretical derivation + Hardware (optional for bonus)

-Theoretical: K map + circuit diagram (proteus to be uploaded)

- **Report should contain**
- **General description**
- **Method of derivation and derived results**
- **Circuit diagram with values of electrical components**
- **Circuit operation principles**
- **Program flow chart**
- **Arduino program with no compiling error**

CHAPTER 01

Abstract

A microcontroller is used in the development board known as Arduino. In comparison to other microcontroller systems, Arduino boards are less costly, and the programming interface is easy to use. Using Arduino, the encryption table was constructed. For the purpose of adjusting the inputs for high and low situations, we employed a Single Pole Double Throw (SPDT) switch. To evaluate the different output states, LEDs were also used.

General description

Microprocessor:

A microprocessor, which is built on a small chip and can carry out ALU (Arithmetic Logical Unit) tasks as well as communicate with other connected devices, is the controlling element of a microcomputer. A microprocessor consists of a control unit, an ALU, and a register array.

ALU: An ALU is a sort of digital circuit that performs arithmetic and logical operations on data coming from a memory or input source.

Register Array: The register array is made up of the accumulator, B, C, D, E, H, and L-letter registers.

Control Unit: The computer's control unit is in charge of controlling the flow of data and commands.

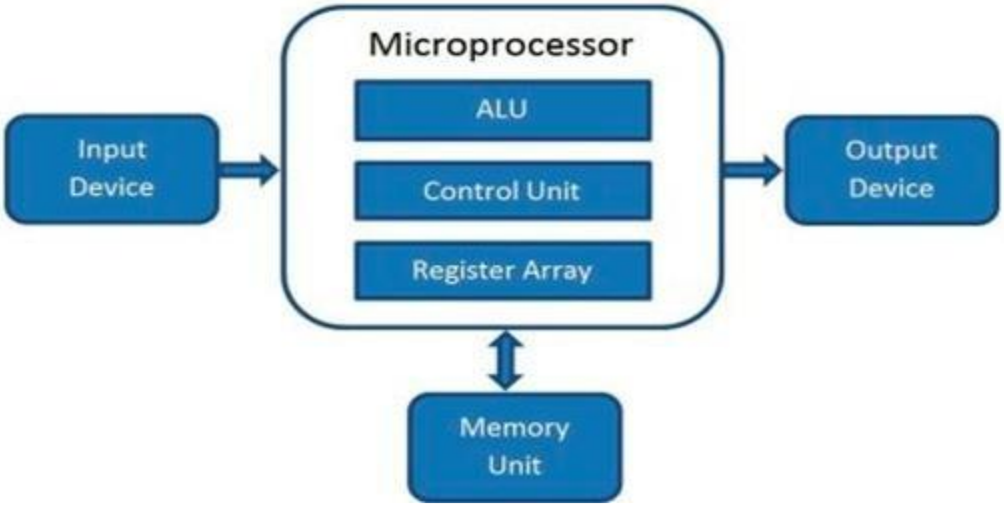


Figure: Block Diagram of basic microcomputer

Each microprocessor has a set of instructions provided by the manufacturer. A microprocessor has two versions of its instruction set.

- Binary Machine Code
- Mnemonics.

Binary numbers 0 and 1 are used by microprocessors for both communication and computation. A computer language is a set of instructions presented as binary patterns that are challenging for humans to understand. The assembly language is created as a consequence, and the binary patterns are given short names like mnemonics. A software called Assembler transforms assembly-level language into machine-level binary code.

8085 Microprocessor: In March 1976, Intel unveiled the Intel 8085, an 8-bit microprocessor. It is a software binary compatible with the more well-known Intel 8080, with the addition of two tinier instructions added to handle the interrupt and serial input/output functionalities.

8086 Microprocessor: A more potent variation of the Intel 8085 Microprocessor, which was originally offered in 1976, is the Intel 8086 Microprocessor. It has a 1MB memory capacity and a 16-bit CPU with 20 address lines and 16 data lines. Operations like multiplication and division are made simpler because to its adaptable instruction set. It operates in two different ways: maximum and minimum. While Minimum mode is excellent for single-processor systems, Maximum mode is ideal for multi-processor machines.

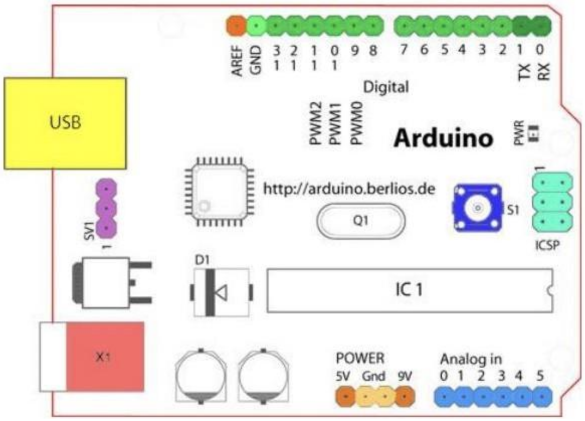
Microcontroller:

A microcontroller is a compact, reasonably priced microprocessor that is made to carry out the particular functions of embedded systems, such as showing microwave data, receiving distant signals, etc. The CPU, memory (RAM, ROM, EPROM), Serial ports, peripherals (timers, counters, etc.), and other components make up a generic microcontroller.

Arduino:

The Arduino system consists of a physical programmable circuit board, or microcontroller, as well as programming software, or IDE (Integrated Development Environment), which can be run on a PC and is used to create and transmit PC code to the circuit board. The Arduino Software (IDE), based on Processing, and the Arduino Programming Language (based on Wiring), may be used to accomplish this. The Arduino does not need special hardware (referred to as a software engineer) to upload code to the circuit board; instead, one may effectively use a USB link. Additionally, the Arduino IDE uses a modified version of C++ that makes learning to program easier. **Typical Elements on Arduino Boards:**

For various uses, there are several varieties of Arduino boards. But the bulk of the following elements are present on all the boards.



Starting clockwise from the top center:

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)

- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (digitalRead and digitalWrite) if serial communication is also being used (e.g. Serial. begin).
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) -SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

Digital Pins:

The digital pins on an Arduino board can be used for general-purpose input and output via the pinMode(), digitalRead(), and digitalWrite() commands. Each pin has an internal pull- up resistor which can be turned on and off using digitalWrite() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input.

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
- External Interrupts 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On boards with an ATmega8, PWM output is available only on pins 9, 10,
  and 11.[1]

Analog Pins:

The analog input pins support 10-bit analog-to-digital conversion (ADC) using the analogRead() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19.

Power Pins:

- 9V: The input voltage to the Arduino board when it’s using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). Different boards accept different input voltages ranges.
- 5V: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3: (Decimal-only) A 3.3-volt supply generated by the on-board FTDI chip.
- GND: Ground pins.

CHAPTER 02

Equipment

As we are implementing hardware and software both in our project, we would need the following equipment:

1. For Hardware
   - Arduino UNO R3
   - BreadBoard
   - Resistors (10k Ohm 1/4W Resistor,330 Ohm 1/4W Resistor)
   - Wires
   - Red LED (5mm) (5 mm Red Color LED)
   - Slide Switch (3 pin slide switch)

2. For Software
   - **Proteus 8 Professional Software**
   - **Logisim**
   - **Arduino IDE**

Arduino IDE:

It is simple to create code and upload it to the board using the free and open-source Arduino Software (IDE). The environment is created using Processing and other open-source technologies and is written in Java. Any Arduino board may be used with this software.

Proteus 8 Professional:

The Proteus Design Suite is a proprietary software tool suite used primarily for electric design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing painted circuit boards.

The Proteus Design Suite is a Windows application for schematic capture, simulation, and PCB (Printed Circuit Board) layout design. It can be purchased in many configurations, depending on the size of designs being produced and the requirements for microcontroller simulation. All PCB Design products include an autorouter and basic mixed mode SPICE simulation capabilities.

Logisim:

Logisim is an educational tool for designing and simulating digital logic circuits. With its simple toolbar interface and simulation of circuits as you build them, it is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to build larger circuits from smaller subcircuits, and to draw bundles of wires with a single mouse drag, Logisim can be used (and is used) to design and simulate entire CPUs for educational purposes.

We have designed our circuit using Logic Gates using the software Logism.

CHAPTER 03

1. Experimental Planning:

We have to implement an encryption table using a microcontroller. Microcontroller is a small computer based on a single metal-oxide-semiconductor, in sort MOS, integrated circuit chip. It contains one or more CPUs with memory and programmable input/outputs. In this project, we are using Arduino Uno.

To get the desired output, here we have written a code based on derived logical expressions. To get these expressions, we have derived K-map according to the given input table.

1. Experimental Circuit Setup:

- First, power and ground were connected to the breadboard through the Arduino.
- Then placed the Arduino Board
- Connected the wires with input and output pins
- Placed the LED’s on breadboard
- Connected the power supply to the Arduino

2. Method of Derivation and Results

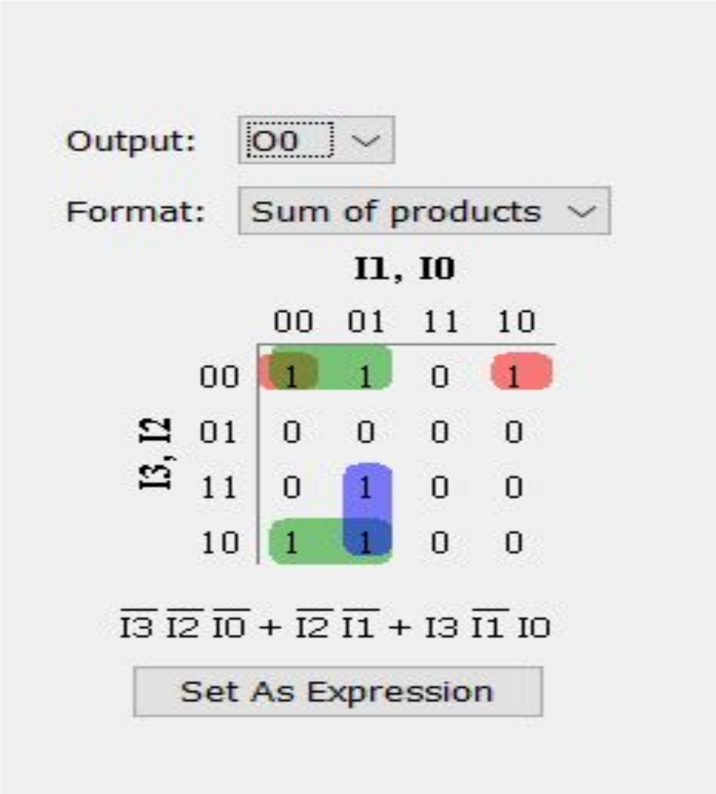
3. Theoretical observation

4. Truth Table

I3	I2	I1	I0	O3	O2	O1	O0
0	0	0	0	0	1	0	1
0	0	0	1	0	1	0	1
0	0	1	0	1	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	0
0	1	0	1	0	0	1	0
0	1	1	0	1	1	0	0
0	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	0
1	1	0	0	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0

3.1.5 K-Mapping

K-maps are used to simplify the logic requirements and provide a visual way to produce a much simpler formula for expressing the same logic derived from the truth table. Here our task is to minimize Boolean expressions of four variables from the given encryption table.

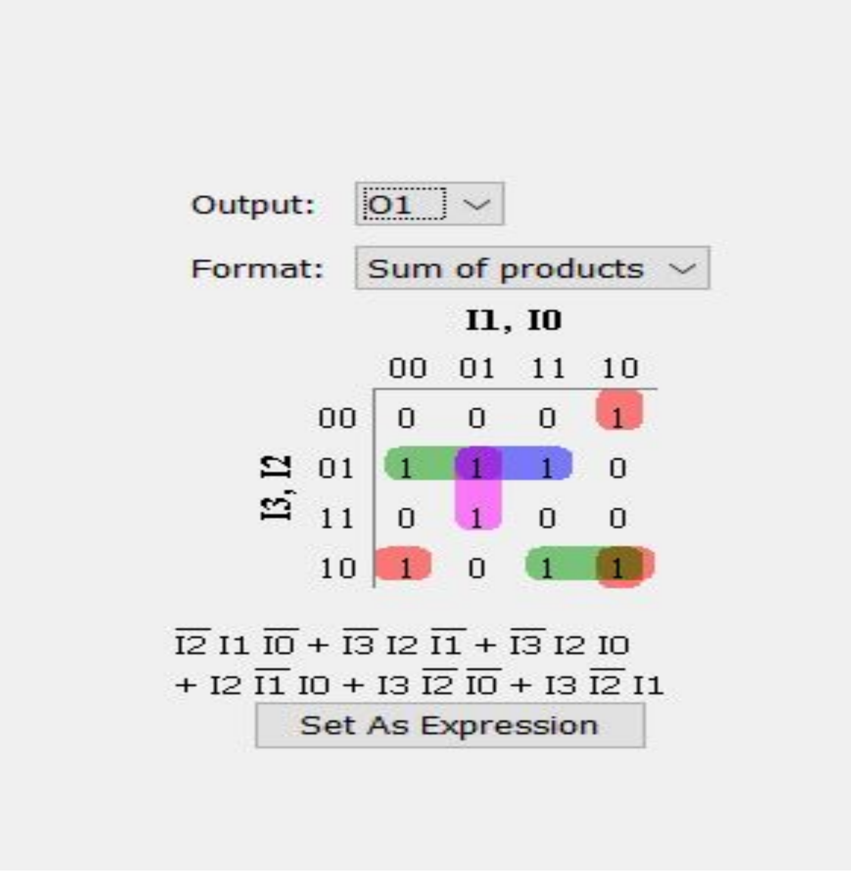


Output: O0

Format: Sum of products

I3, I2 \ I1, I0	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	0	1	0	0
10	1	1	0	0

$\overline{I3}\ \overline{I2}\ \overline{I0} + \overline{I2}\ \overline{I1} + I3\ \overline{I1}\ I0$

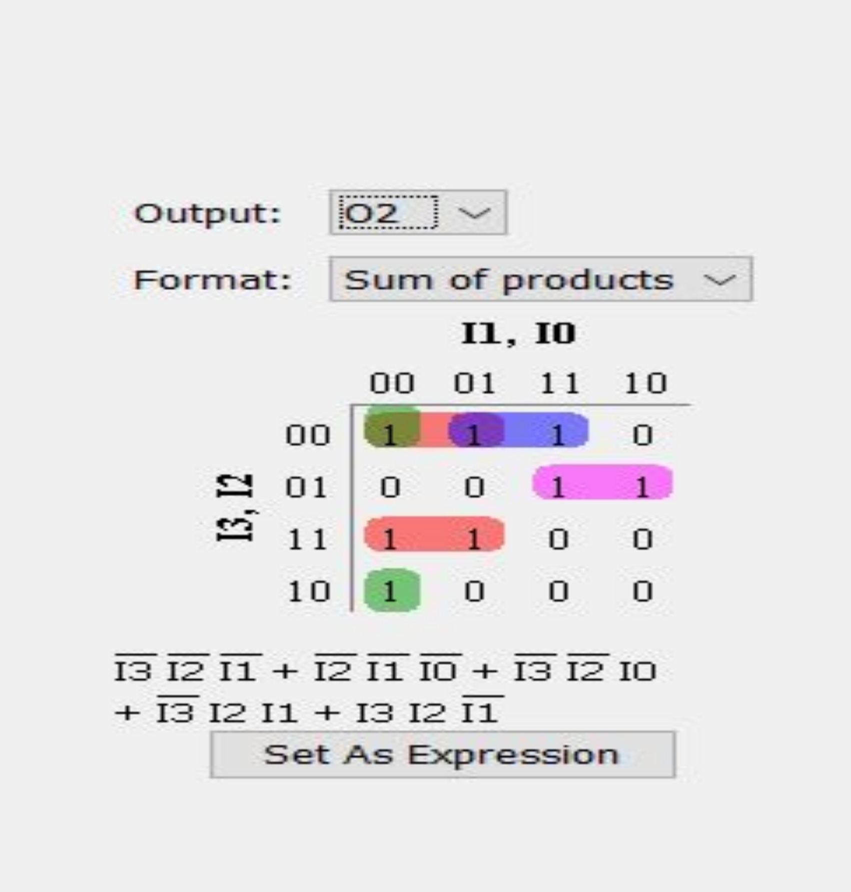


Output: O1

Format: Sum of products

I3, I2 \ I1, I0	00	01	11	10
00	0	0	0	1
01	1	1	1	0
11	0	1	0	0
10	1	0	1	1

$\overline{I2}\; I1\; \overline{I0} + \overline{I3}\; I2\; \overline{I1} + \overline{I3}\; I2\; I0$
$+ I2\; \overline{I1}\; I0 + I3\; \overline{I2}\; \overline{I0} + I3\; \overline{I2}\; I1$

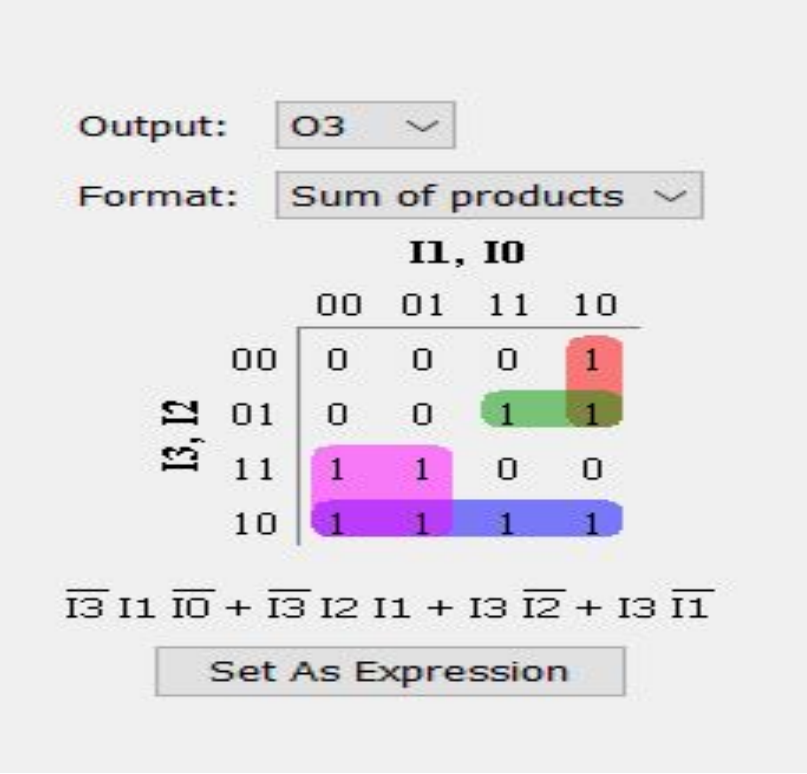


Output: O2

Format: Sum of products

I3, I2 \ I1, I0	00	01	11	10
00	1	1	1	0
01	0	0	1	1
11	1	1	0	0
10	1	0	0	0

$\overline{I3}\; \overline{I2}\; \overline{I1} + \overline{I2}\; \overline{I1}\; \overline{I0} + \overline{I3}\; \overline{I2}\; I0$
$+ \overline{I3}\; I2\; I1 + I3\; I2\; \overline{I1}$



Output: O3

Format: Sum of products

I1, I0

I3, I2	00	01	11	10
00	0	0	0	1
01	0	0	1	1
11	1	1	0	0
10	1	1	1	1

$\overline{I3}\ I1\ \overline{I0} + \overline{I3}\ I2\ I1 + I3\ \overline{I2} + I3\ \overline{I1}$

3.1.6 Derived Results

After inserting the truth table in the Logisim we came up with these k-maps and expressions.

3.1.7 Expressions

O0 = **I3’ I1 I0’ + I3’ I2 I1 + I3 I2’ + I3 I1’**

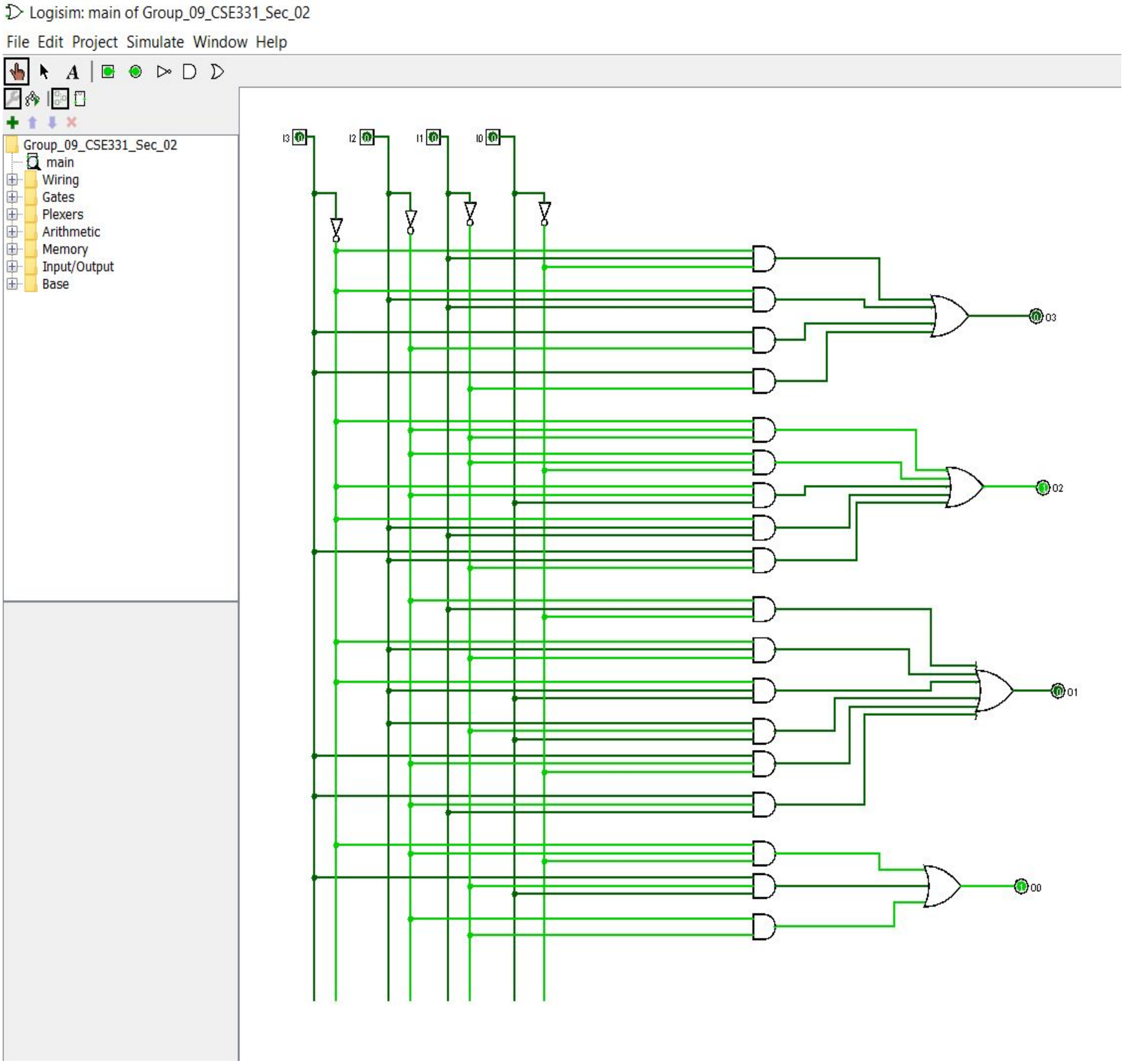
O1 = **I2’ I1 I0’ +I3’ I2 I1’ + I3’ I2 I0 + I2 I1’ I0 + I3 I2’ I0’ + I3 I2’ I1**

O2 = **I3’ I2’ I1+ I2’ I1’ I0’ + I3’ I2 I0 + I3’ I2 I1 + I3 I2 I1’**

O3 = **I3’ I1 I0’ + I3’ I2 I1+ I3 I2’ + I3 I1’**

CHAPTER 04

Logisim Circuit



Circuit Diagram with Values of Electrical Components

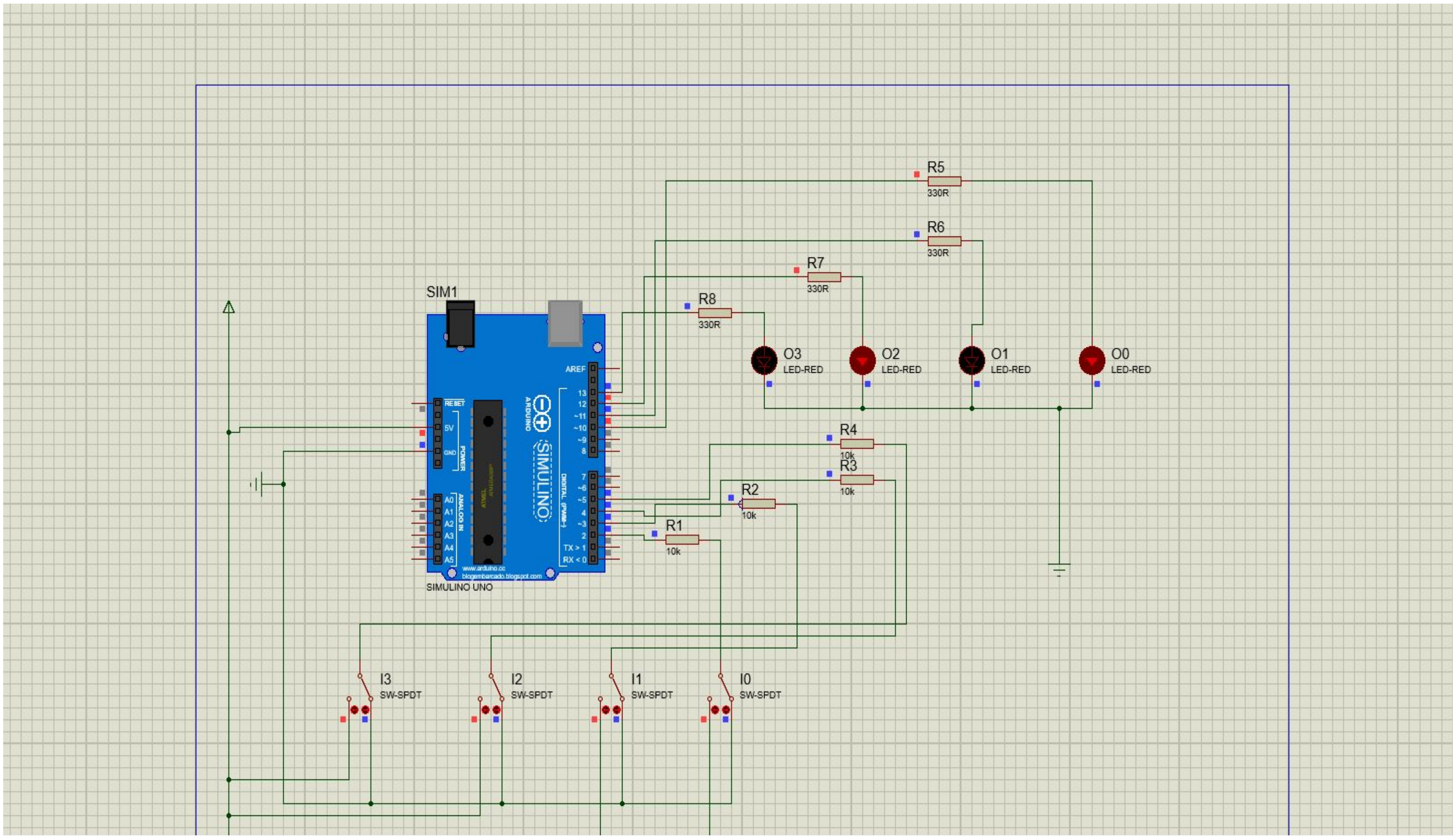
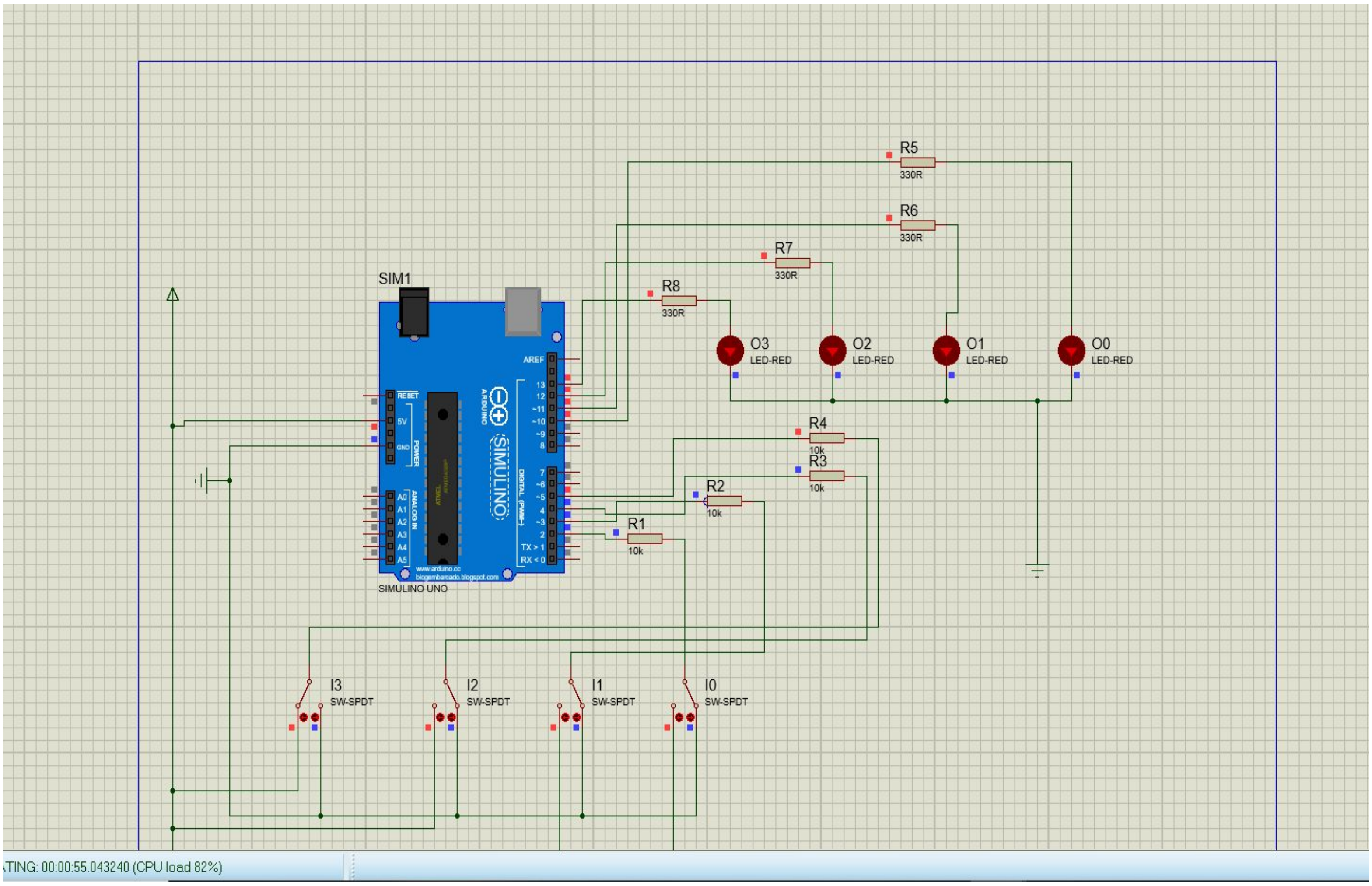


Figure: Here, Input, I3=0 , I2=0 , I1=0 , I0=0 And the Output comes through LEDs , O3=0 , O2=1, O1=0 , O0=1



ATING: 00:00:55.043240 (CPU load 82%)

Figure: Here, Input, I3=1 , I2=0 , I1=0 , I0=0 And the Output comes through LEDs , O3=1 , O2=1, O1=1 , O0=1

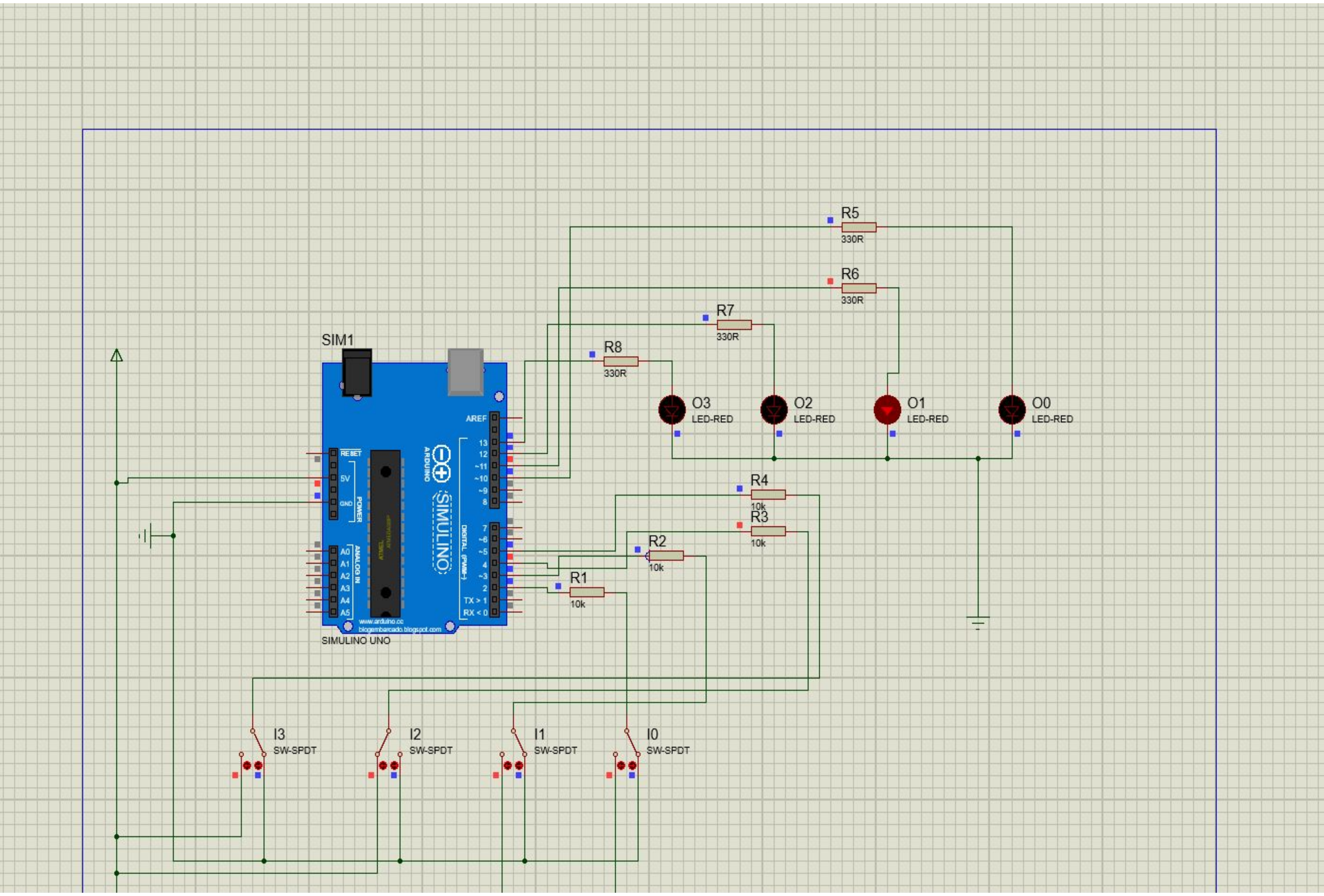


Figure: Here, Input, I3=0 , I2=1 , I1=0 , I0=1 And the Output comes through LEDs , O3=1 , O2=0, O1=0 , O0=0

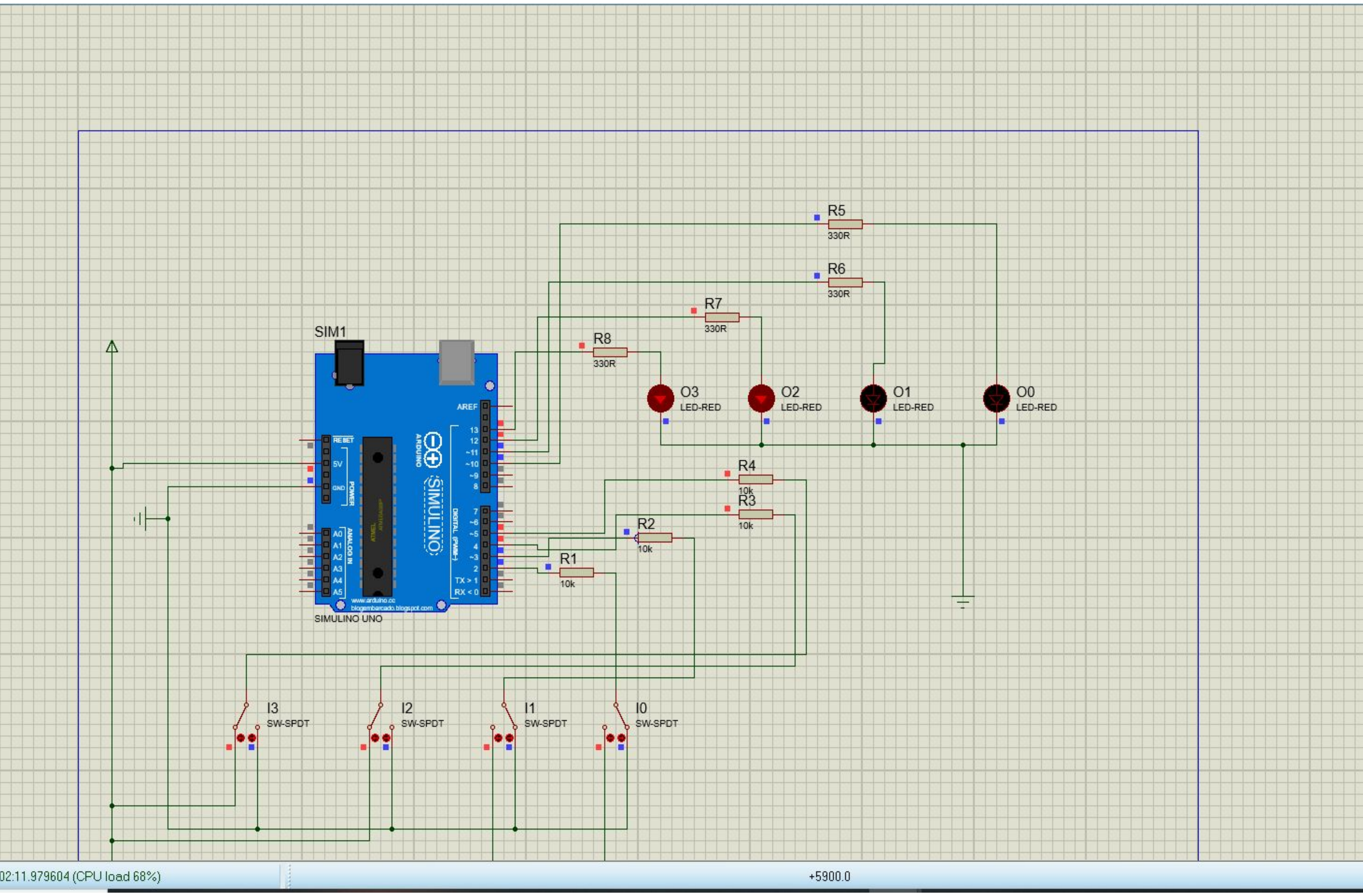


Figure: Here, Input, I3=1 , I2=1 , I1=0 , I0=0 And the Output comes through LEDs , O3=1 , O2=1, O1=0 , O0=0

Circuit Operation Principles

We have equated our Karnaugh Map and got 4 equations for our output O0, O1 , O2, O3. They are:

$O0 = I3' I1 I0' + I3' I2 I1 + I3 I2' + I3 I1'$
$O1 = I2' I1 I0' + I3' I2 I1' + I3' I2 I0 + I2 I1' I0 + I3 I2' I0' + I3 I2' I1$
$O2 = I3' I2' I1 + I2' I1' I0' + I3' I2 I0 + I3' I2 I1 + I3 I2 I1'$
$O3 = I3' I1 I0' + I3' I2 I1 + I3 I2' + I3 I1'$

With the help of these equations we have designed our Circuit Diagram using Logic Gates (AND , NOT , OR) . We have used the software Logism to build the Circuit Diagram.

Then, with the help of Proteus 8 Professional Software we have built our main Hardware Circuit.
We have used the following components:

1. **Arduino Uno :** We have used an Arduino Uno Model as our microcontroller which is the main of this project.
   We will connect the 5V pin of Arduino to a power supply of 5V. And we will connect the ground pin of the Arduino to a Ground component.
2. **Resistors :** We have used a total 8 resistors of two values. We have used 330 ohm resistors with the LEDs . And we have used 10k ohm resistors with the SPDT Switches.
3. **Single Pole Double Throw Switches :** We used four of these Switches with our Arduino Digital pins. They are arranged in such a way:

   Arduino UNO Digital Pin No. 2 --------- SPDT Switch 1 (I0)
   Arduino UNO Digital Pin No. 3 --------- SPDT Switch 2 (I1)
   Arduino UNO Digital Pin No. 4 --------- SPDT Switch 3 (I2)
   Arduino UNO Digital Pin No. 5 --------- SPDT Switch 4 (I3)

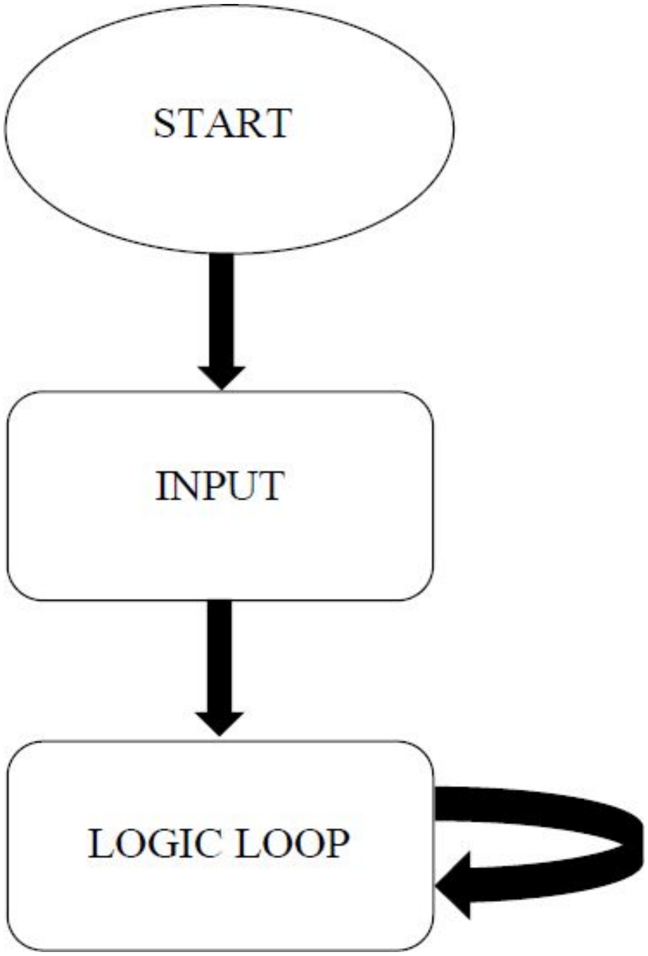
   Also we connected the 10k Resistor to one side of the Digital Pin that is connected to the Switch and another side to Ground . Like this orientation, we placed all the four Switches with all the four 10k resistors.
4. **LEDs :** We have connected Four LEDs to the Digital Pins of Arduino to receive Output through those Pins. They are arranged in such way:

Arduino UNO Digital Pin No. 10 --------- LED1 (O0)
Arduino UNO Digital Pin No. 11 --------- LED2 (O1)
Arduino UNO Digital Pin No. 12 --------- LED3 (O2)
Arduino UNO Digital Pin No. 13 --------- LED4 (O3)

After we connect all the wires in place. We will go for the Arduino Code and insert the logic gates or the logical conditionals that we got from kmap and logism.

CHAPTER 05

Program Flow Chart



Arduino Program with No Compiling Error

Code :

```
int O0 = 10;
int O1 = 11;
int O2 = 12;
int O3 = 13;

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


void setup()
{
  pinMode(I0,INPUT);
  pinMode(I1,INPUT);
  pinMode(I2,INPUT);
  pinMode(I3,INPUT);
  pinMode(O0,OUTPUT);
  pinMode(O1,OUTPUT);
  pinMode(O2,OUTPUT);
  pinMode(O3,OUTPUT);

}

void loop()
{

  boolean I0State = digitalRead(I0);
  boolean I1State = digitalRead(I1);
  boolean I2State = digitalRead(I2);
  boolean I3State = digitalRead(I3);
  boolean O0State;
  boolean O1State;
  boolean O2State;
  boolean O3State;
O0State= (!I3State & !I2State & !I0State)|(I3State & !I1State & I0State)|(!I2State & !I1State);
O1State= (!I2State & I1State & !I0State)|(!I3State & I2State & !I1State)|(I2State & !I1State & I0State)|(I3State & !I2State & !I0State)|(!I3State & I2State & I0State)|(I3State & !I2State & I1State);
O2State= (!I3State & !I2State & !I1State)|(!I3State & I2State & I1State)|(!I2State & !I1State & !I0State)|(!I3State & !I2State & I0State)|(I3State & I2State & !I1State);
O3State= (!I3State & I1State & !I0State)|(!I3State & I2State & I1State)|(I3State & !I2State)|(I3State & !I1State);


 digitalWrite(O0,O0State);
 digitalWrite(O1,O1State);
 digitalWrite(O2,O2State);
 digitalWrite(O3,O3State);

   }
```

Hex Code

```
:100000000C9461000C947E000C947E000C947E0095
:100010000C947E000C947E000C947E000C947E0068
:100020000C947E000C947E000C947E000C947E0058
:100030000C947E000C947E000C947E000C947E0048
:100040000C94F3010C947E000C947E000C947E00C2
:100050000C947E000C947E000C947E000C947E0028
:100060000C947E000C947E000000000080002010049
:100070000003040700000000000000000102040863
:100080001020408001020408102001020408102002
:10009000040404040404040402020202020203032E
:1000A000030303030000000002300260029000000D2
:1000B0000000250028002B00000000000240027007D
:1000C0002A0011241FBECFEFD8E0DEBFCDBF11E064
:1000D000A0E0B1E0E4EFF4E002C005900D92A031A1
:1000E000B107D9F721E0A0E1B1E001C01D92A9312B
:1000F000B207E1F70E94E4010C9478020C9400002E
:1001000060E0809106010E944B0160E08091040153
:100110000E944B0160E0809102010E944B0160E06F
:10012000809100010E944B0161E080910E010E94CC
:100130004B0161E080910C010E944B0161E08091D4
:100140000A010E944B0161E0809108010C944B016F
:10015000FF920F931F93CF93DF9380910601 0E942C
:10016000BA0101E0892B09F400E0809104010E94AA
```

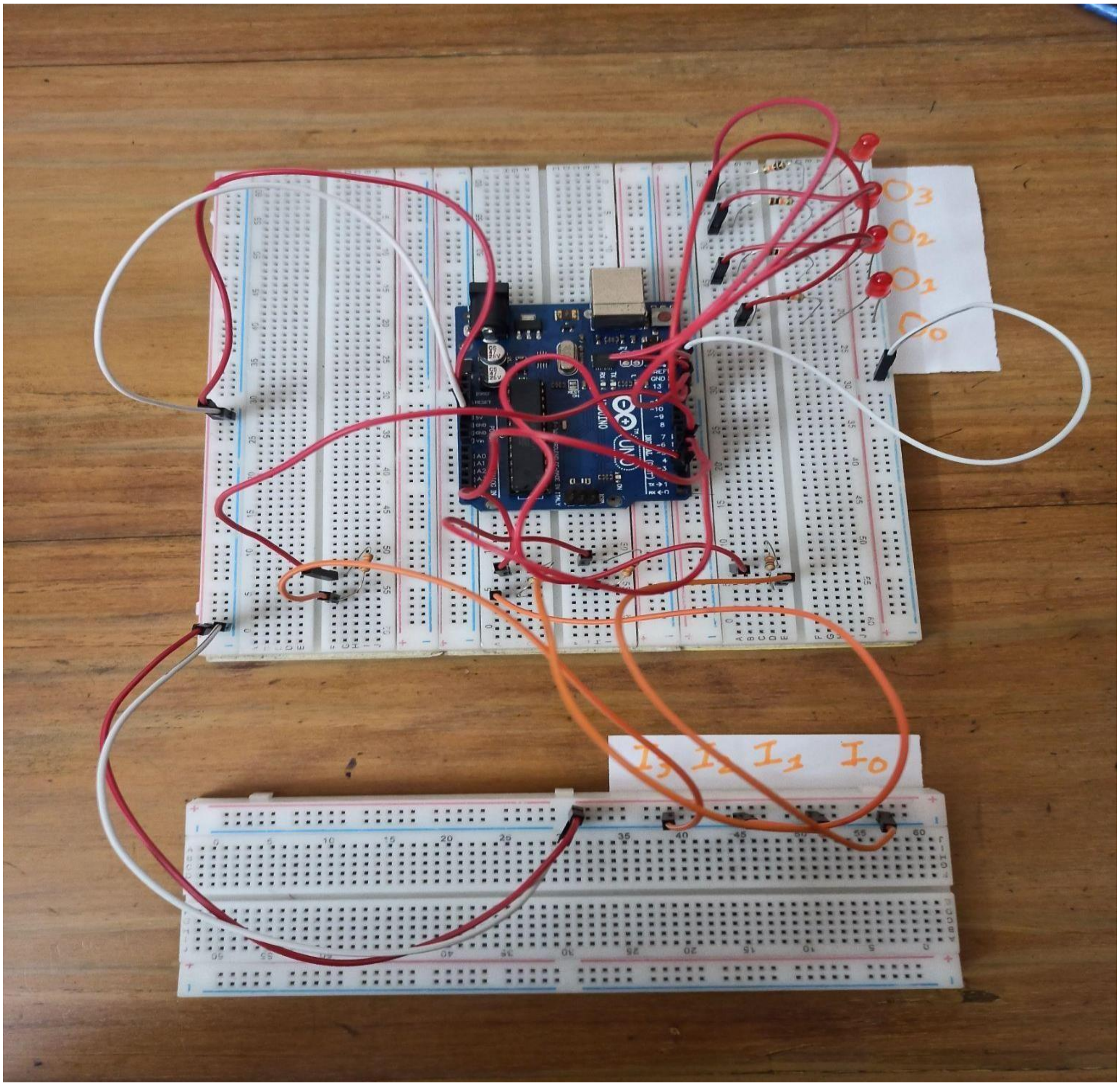
```
:10017000BA01FF24F394892B09F4F12C8091020138
:100180000E94BA0111E0892B09F410E0809100016E
:100190000E94BA0121E0892B09F420E081E0522F6E
:1001A0005827312F3827632F6523902F98278F25C5
:1001B000E82FE323452F4123F32FF223C32FC92335
:1001C000CF21742F7823C72B712F70237823C72B4F
:1001D0007F2F7923C72B742F7023C72BFF21CF2BA1
:1001E000D62FD823742F7F21D72B12231823D12B5E
:1001F0007E2F7923D72B762F7023D72B152F1923FA
:10020000142B1F21382B3223132B6923022308239D
:10021000602B6E2B80910E010E9484016C2F8091C7
:100220000C010E9484016D2F80910A010E948401BB
:10023000612F80910801DF91CF911F910F91FF9065
:100240000C948401833081F028F4813099F082305D
:10025000A1F008958730A9F08830B9F08430D1F446
:10026000809180008F7D03C0809180008F77809384
:10027000800008958480B58F7702C084B58F7D84BDDA
:100280000895809lB0008F7703C08091B0008F7D7A
:100290008093B0000895CF93DF9390E0FC01E45881
:1002A000FF4F2491FC01E057FF4F8491882349F1CF
:1002B00090E0880F991FFC01E854FF4FA591B4917D
:1002C00082559F4FFC01C591D4919FB7611108C021
:1002D000F8948C91209582238C93888182230AC084
:1002E000623051F4F8948C91322F309583238C93A3
:1002F0008881822B888304C0F8948C91822B8C9304
:100300009FBFDF91CF9108950F931F93CF93DF93FA
:100310001F92CDB7DEB7282F30E0F901E859FF4F23
:100320008491F901E458FF4F1491F901E057FF4F10
:1003300004910023C9F0882321F069830E942201DF
:100340006981E02FF0E0EE0FFF1FE255FF4FA5910E
:10035000B4919FB7F8948C91611103C010958123DB
:1003600001C0812B8C939FBF0F90DF91CF911F9184
:100370000F910895CF93DF93282F30E0F901E859CA
:10038000FF4F8491F901E458FF4FD491F901E057F0
:10039000FF4FC491CC2391F081110E942201EC2FD8
:1003A000F0E0EE0FFF1FEC55FF4FA591B4912C919B
:1003B0002D2381E090E021F480E002C080E090E015
:1003C000DF91CF91089508950E943D020E94E301BC
:1003D0000E948000C0E0D0E00E94A8002097E1F3D6
:1003E0000E940000F9CF1F920F920FB60F921124B6
:1003F0002F933F938F939F93AF93BF9380911101 5E
:1004000090911201A0911301B09114013091100148
:1004100023E0230F2D3720F40196A11DB11D05C047
:1004200026E8230F0296A11DB11D20931001809391
:10043000110190931201A0931301B09314018091C4
:100440001501909116 01A0911701B0911801019624
:10045000A11DB11D8093150190931601A093170162
```

```
:10046000B0931801BF91AF919F918F913F912F91C0
:100470000F900FBE0F901F901895789484B58260EE
:1004800084BD84B5816084BD85B5826085BD85B538
:10049000816085BDEEE6F0E0808181608083E1E8E7
:1004A000F0E01082808182608083808181608083 1F
:1004B000E0E8F0E0808181608083E1EBF0E0808122
:1004C00084608083E0EBF0E0808181608083EAE7F4
:1004D000F0E080818460808380818260808380817D
:1004E00081608083808180688083109 2C10008953C
:0404F000F894FFCFAE
:1004F40005000400030002000D000C000B000A00BC
:00000001FF
```

CHAPTER 06

Hardware Implementation

We have used our manually made one pole two throw switches because our purchased switches were not working. We made the switches by making one row connected with Vcc as input =1 and one row with GND as input = 0 and which work as the same.



Hardware Implementation with Values of Electrical Components

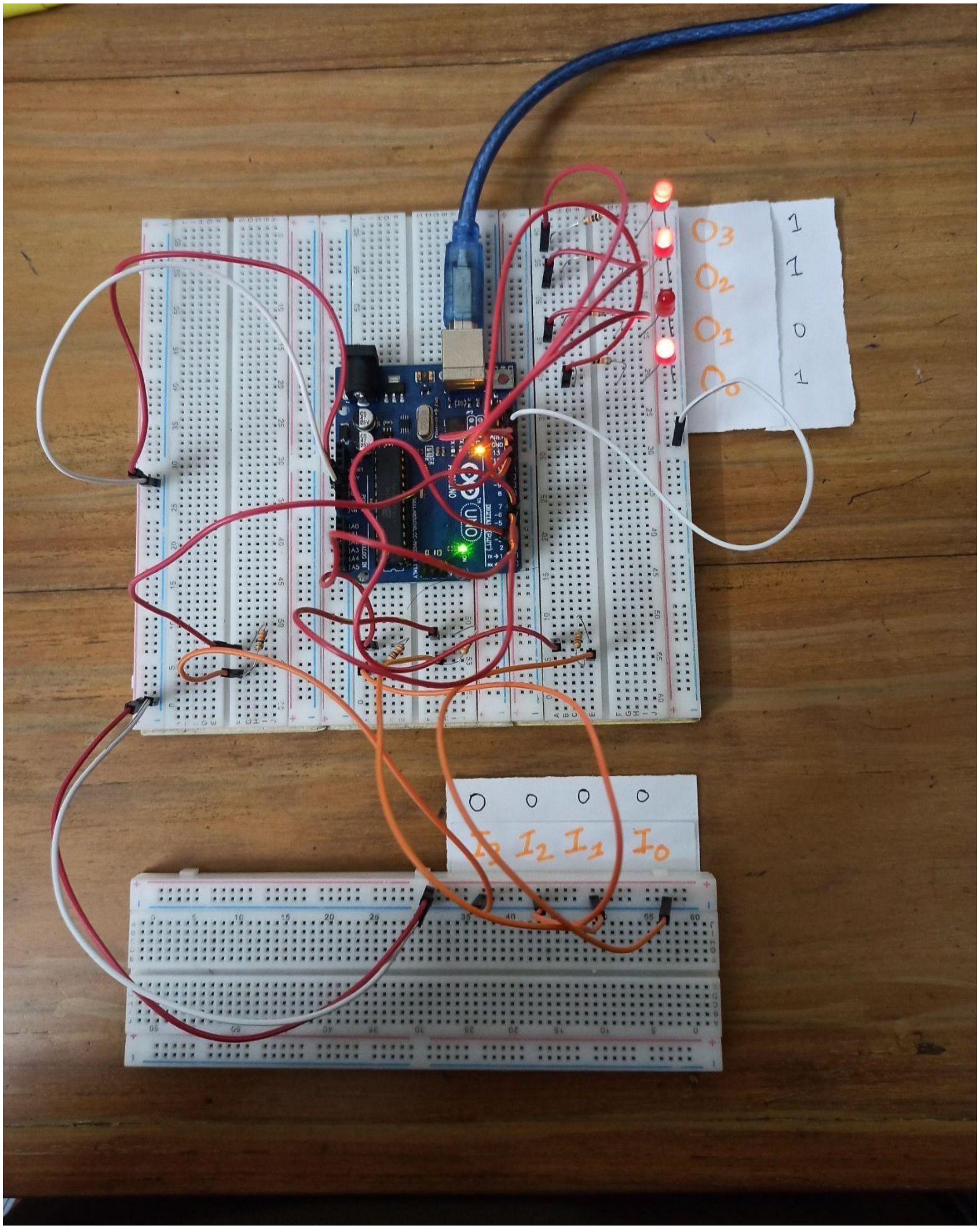


Figure: Here, Input, I3=0 , I2=0 , I1=0 , I0=0 And the Output comes through LEDs , O3=1 , O2=1, O1=0 , O0=1

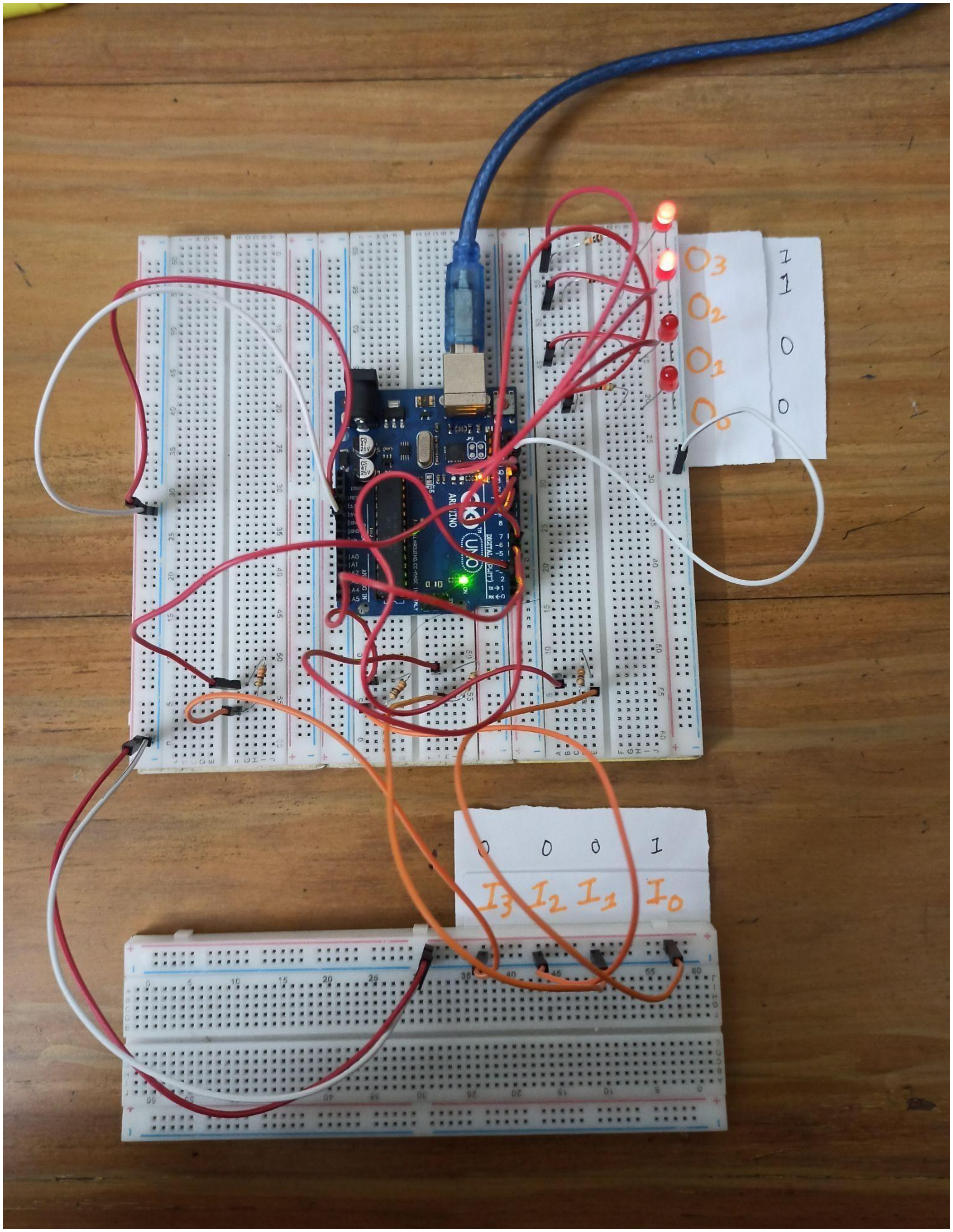


Figure: Here, Input, I3=0 , I2=0 , I1=0 , I0=1 And the Output comes through LEDs , O3=1 , O2=1, O1=0 , O0=0

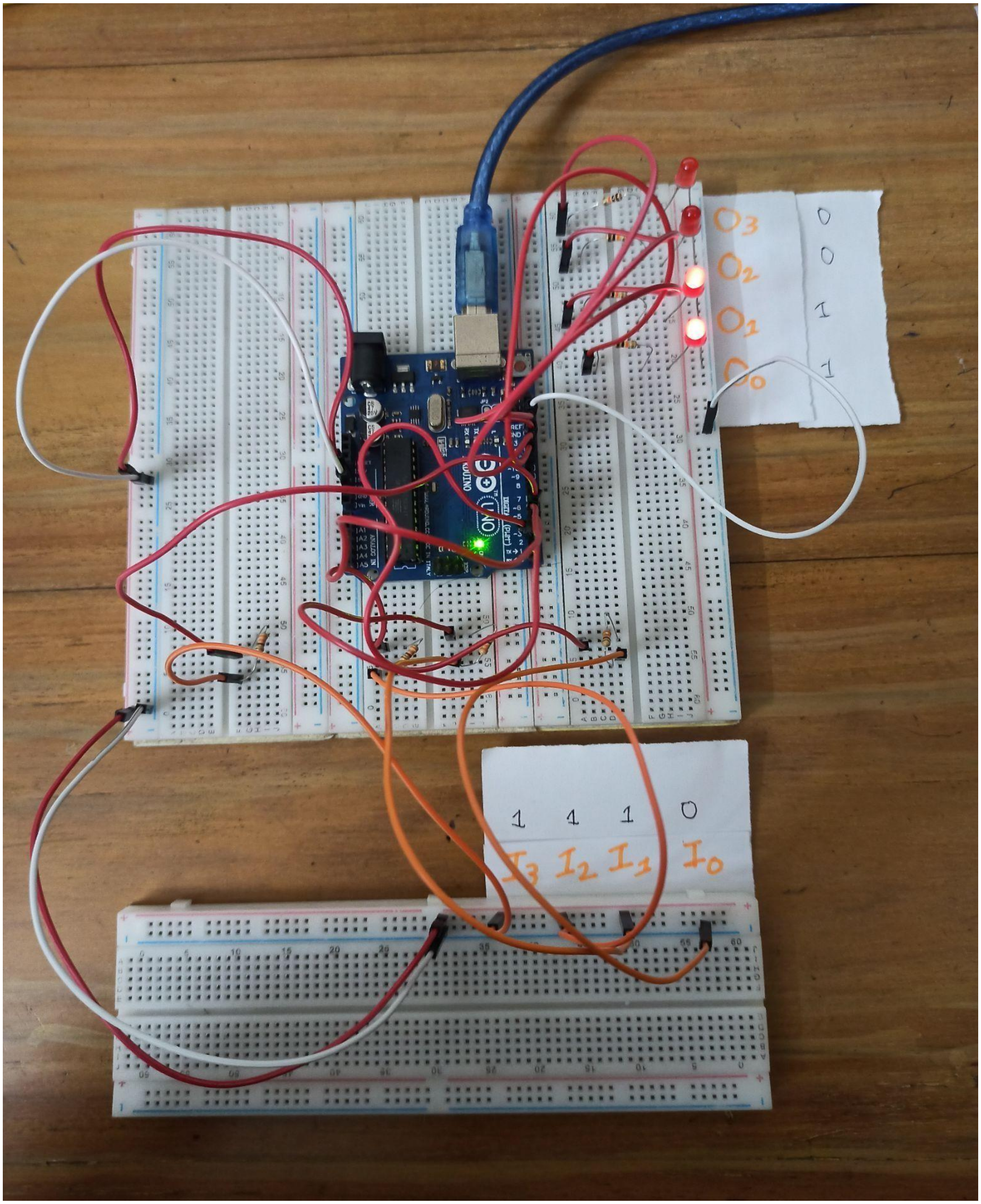


Figure: Here, Input, I3=1 , I2=1 , I1=1 , I0=0 And the Output comes through LEDs , O3=0 , O2=0, O1=1 , O0=1

CHAPTER 07

References

- http://www.multiwingspan.co.uk/arduino.php?page=led7

- https://forum.arduino.cc/index.php?topic=375806.0

- https://www.researchgate.net/publication/322159080_Digital_Logic_Gate_Simulation_using_Arduino_Microcontroller

- https://dronebotworkshop.com/basic-logic/
- https://arduino.stackexchange.com/questions/30119/and-gate-based-on-truth-table