

**North South University**

Department of Electrical & Computer Engineering

Summer 2023

**CSE331**

Microprocessor Interfacing & Embedded System

**Project Report**

***‘Implementation of an Encryption Table Using***

***Micro controller’***

**Faculty: Dr. Dihan Md. Nuruddin Hasan (DMH)**

**Section: 1**

**Group: 8**

**Project members:**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **RDS Section** |
| **Ferdous Reza Niloy** | **2021281042** | **1** |
| **Rafiduzzaman Niloy** | **1921723642** | **1** |
| **Md. Afjal Hossain Dali** | **2013652042** | **1** |
| **Sanjana Islam** | **2013337642** | **1** |
| **Afsara Waziha** | **2021235642** | **1** |
| **Faizul Islam Bhuiyan** | **2012878042** | **1** |

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[1.](#page4)** | **[Main Objective](#page4)** [....................................................................................................................................](#page4) | | | **3** |
| **[2.](#page5)** | **[General Description](#page5)** [............................................................................................................................](#page5) | | | **4** |
| [2.1](#page5) | |  | [Arduino UNO R3 ..........................................................................................................................](#page5) | **4** |
| [2.2](#page6) | |  | [Arduino Mega 2560 ......................................................................................................................](#page6) | **[5](#page6)** |
| [2.3](#page7) | |  | [Arduino IDE ..................................................................................................................................](#page7) | **5** |
| [2.4](#page7) | |  | [Proteus 8 Pro .................................................................................................................................](#page7) | **[6](#page7)** |
| [2.5](#page8) | |  | [Logisim .........................................................................................................................................](#page8) | **6** |
| **[3.](#page8)** | **[Equipment Used](#page8)** [..................................................................................................................................](#page8) | | | **[7](#page8)** |
| **[4.](#page9)** | **[Method of Derivation](#page9)** [..........................................................................................................................](#page9) | | | **[8](#page9)** |
| [4.1](#page9) | |  | [Truth Table ....................................................................................................................................](#page9) | **[8](#page9)** |
| [4.2](#page10) | |  | [Karnaugh Map ..............................................................................................................................](#page10) | **[9](#page10)** |
| [4.3](#page12) | |  | [Derived Boolean Output Expression from Karnaugh Map .........................................................](#page12) | **[11](#page12)** |
| **[5.](#page12)** | **[Circuit Diagram with Values of Electrical Components](#page12)** [................................................................](#page12) | | | **[11](#page12)** |
| [5.1](#page12) | |  | [Logisim Circuit ...........................................................................................................................](#page12) | **[11](#page12)** |
| [5.2](#page13) | |  | [Proteus Circuit ............................................................................................................................](#page13) | **[12](#page13)** |
| **[6.](#page14)** | **[Circuit Operation Principles](#page14)** [............................................................................................................](#page14) | | | **[13](#page14)** |
| **[7.](#page15)** | **[Program Flow Chart](#page15)** [.........................................................................................................................](#page15) | | | **15** |
| **[8.](#page16)** | **[Arduino Program](#page16)** [..............................................................................................................................](#page16) | | | **16** |
| [8.1 Arduino Program for both Arduino UNO R3 and Mega using Arduino Libraries: ...........................](#page16) | | | | **16** |
| 8.2 | |  | [Hex Code ....................................................................................................................................](#page19) | **[18](#page19)** |
|  | [8.2.1](#page19) | | [Hex Codes generated for use on Arduino UNO R3 ............................................................](#page19) | **[18](#page19)** |
|  | [8.2.2 Hex Code generated for use on Arduino Mega 2560 .................................................................](#page29) | | | **21** |
| **[9](#page42)** | **[Result Analysis and Comparison Among The Boards Used](#page42)** [..........................................................](#page42) | | | **25** |
| [9.1](#page42) | |  | [Hardware Comparison Among Boards Used ..............................................................................](#page42) | **25** |
| [9.2](#page43) | |  | [Power Consumption Comparison ...............................................................................................](#page43) | **26** |
| [9.3 Memory Size and Execution Time Comparison ...............................................................................](#page44) | | | | **27** |
| **[10](#page45)** |  | **[Conclusion](#page45)** [.....................................................................................................................................](#page45) | | **28** |
| **[11](#page45)** | **[Questions and Answers](#page45)** [.....................................................................................................................](#page45) | | | **29** |
| [11.1](#page45) | |  | [Arduino UNO ..............................................................................................................................](#page45) | **29** |
| [11.2](#page46) | |  | [Arduino Mega 2560 ....................................................................................................................](#page46) | **30** |
| **12**.**[References](#page47)** [...................................................................................................................................](#page47)………. | | | | **32** |

**13 Resources** ……………………………………………………………………………………………………………………**32**

**1. Main Objective**

The objective of this project was to create a 4 -bit encryption system that masks that the output using Boolean Expression. The encryption algorithm was derived from a Truth Table of inputs and its subsequent output. The 4-bit input would be fed into a micro controller which would then encrypt the output according to the Truth Table using Boolean logic expression and display it via 4 LED's to signify 4- bits. A total of 16 possible encryption combinations are possible using this 4-bit circuit: from a value of 0 to 15(ranging from 0000 to 1111); for each of these combinations, the circuit's current is measured.

.

The goal of this experiment is to measure how much current the circuit draws for every possible logic that can be applied to the input and the entire system This is carried out for two different micro controller boards with various specifications and compare their differences of program size, memory occupied, and time taken to execute the program.

The HEX file of the programmed Arduino IDE code was then imported to a circuit schematic designed in Proteus for simulation of the complete system. A virtual terminal was also used monitor the execution. The encryption system was implemented using two different boards in Proteus:

Arduino UNO which uses the ATmega328p micro controller and Arduino Mega 2560 which uses the ATmega2650 micro controller. Both boards implemented the encryption program in each type of programming to compare the differences of performance and memory to ensure the results are consistent and reliable

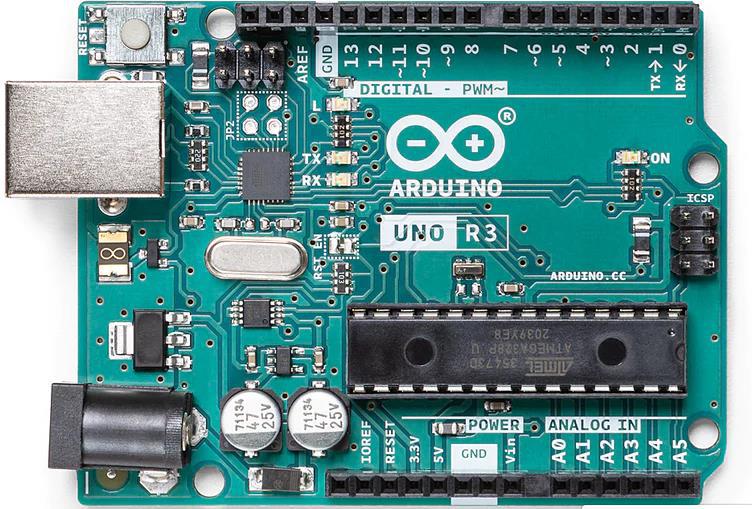
**2.General Description**

**2.1 Arduino UNO R3**

Arduino UNO R3 is a micro controller board based on the ATmega328P micro controller. It has a 16 MHz ceramic resonator (CSTCE16M0V53-R0), 6 analog inputs, 14 digital input/output pins (of which 6 can be used as PWM outputs), a USB port, a power jack, an ICSP header, and a reset button. It has a clock frequency of 16 MHz, 32 KB of flash memory, and 2 KB of SRAM. The Arduino UNO R3 supports serial communication via USB and interfaces such as UART, SPI, and I2C

It can run programs coded with standard C/C++ libraries and headers created for Arduino on the Arduino IDE. It’s simplicity to use, small form factor makes it a popular prototyping board.

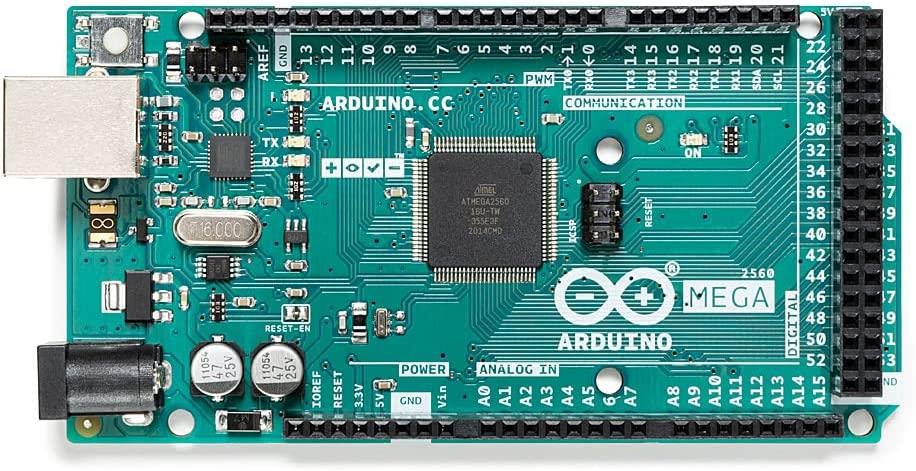




*Figure : Arduino UNO Rev3*

**2.2 Arduino Mega 2560**

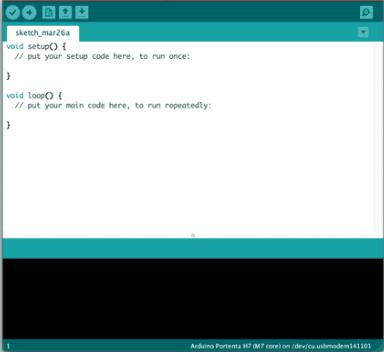




*Figure : Arduino Mega 2560*

The Arduino Mega 2560 is based on the Atmega2560 micro controller and is used for large projects that require high number of input and output ports. As such the Arduino Mega 2560 contains 54 digital IO pins and 16 analog pins. It also has 15 pins among the digital IO pins that supports PWM. Hence, it can support multiple components at a time to read and output values and data. The Arduino Mega 2560 supports serial communication via USB and interfaces such as UART, SPI, and I2C.To support this, it also contains 256 KB of memory to store large volumes of codes, and 8 KB of SRAM and 4 KB of EEPROM. This also owes to the boards large size. Like the Arduino UNO, the Arduino Mega 2560 has a clock frequency of 16 Mhz

**2.3 Arduino IDE**

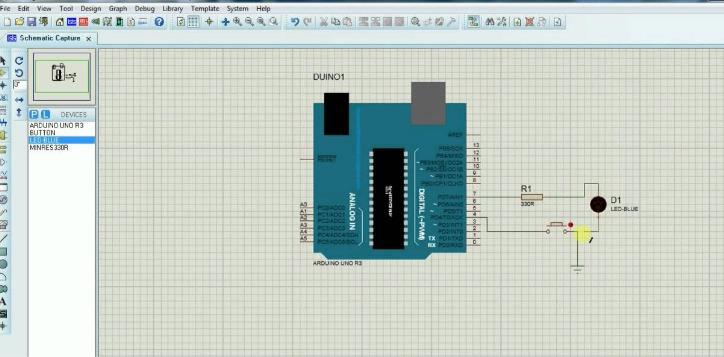


*Figure : Arduino IDE*

Arduino IDE is an integrated development environment that contains a text editor for writing code on C/C++ to program and develop projects on Arduino boards. The Arduino IDE is designed to be beginner-friendly and provides a simple way to write, compile, and upload code to Arduino boards. It contains a message area, a text console, a toolbar with buttons for common functions and a series of menus. Overall, the IDE provides a user-friendly interface for writing, compiling, and uploading codes to Arduino boards. It has set of libraries and functions that makes using the IDE and connecting code with the boards simple and easy.

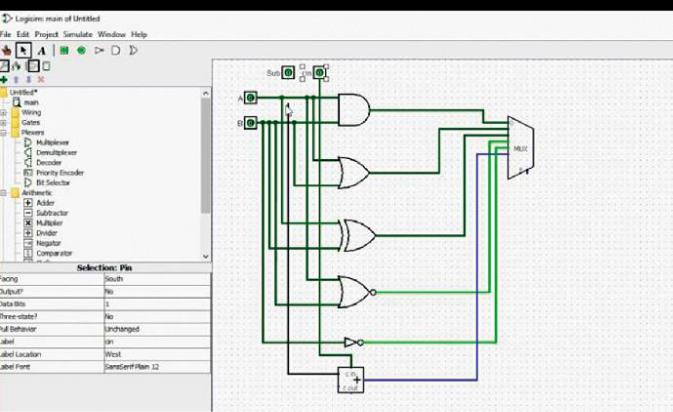
Programs written on the Arduino IDE are called sketched and they are uploaded to the board via its upload feature with just one button click

**2.4 Proteus 8 Pro**



*Figure : Proteus 8 Pro Interface*

Proteus 8 Pro is a comprehensive and versatile tool for electronic circuit design, simulation, and PCB layout. Its intuitive interface, powerful simulation engine, and extensive component library make it a popular choice among electronics enthusiasts and professionals alike.

**2.5 Logisim**



*Figure : Logisim Interface*

Logisim is an open-source program that assists in the design and simulation of logic circuits. The tool can be used to build both straightforward and intricate educational circuits and is intended for use in classrooms. The software contains a large collection of digital components that are widely used circuit design and simulation. It also has a user-friendly interface where components can be dragged and dropped and simulated in real time to see outputs and change inputs.

**3.Equipment Used**

**•Single pole double throw switch:**

It is an electrical switch that has one input terminal (pole) and can connect that terminal to one of two output terminals (throws).

**•10 kΩ Resistor:**

It is an electronic component that provides a resistance of 10,000 ohms to the flow of electric current.

**•Red LED:**

It is a semiconductor device that emits red light when an electric current is passed through it.

**•Arduino UNO REV3:**

It is a popular micro controller board that is widely used in electronics projects and prototyping. It is based on the ATmega328P micro controller and is part of the Arduino family of boards.

**•Arduino Mega 2560:**

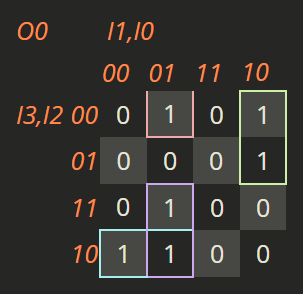
It is a large and versatile micro controller board that has a larger flash memory and EEPROM than other Arduino boards. It is based on the ATmega2560 micro controller and provides similar functionality but in a larger form factor and capacity

**4.Method of Derivation**

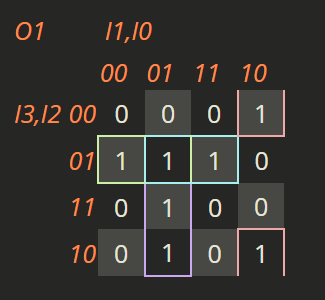
**4.1 Truth Table**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | | **Output** | | | | |
| **I3** | **I2** | **I1** | **I0** | **O3** | **O2** | **O1** | **O0** |
| **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **1** | **0** | **0** | **0** | **1** | **1** | **0** | **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** | **0** | **0** | **1** |
| **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **0** | **1** | **0** | **1** | **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** | **1** |
| **0** | **0** | **1** | **1** | **0** | **1** | **0** | **0** |
| **1** | **0** | **1** | **1** | **1** | **0** | **0** | **0** |
| **0** | **1** | **1** | **1** | **1** | **1** | **1** | **0** |
| **1** | **1** | **1** | **1** | **0** | **1** | **0** | **0** |

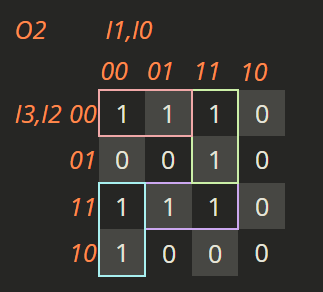
**4.2 Karnaugh Map**



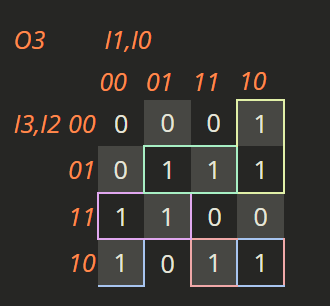
*Figure : K map Derivation for O0*



*Figure : K map Derivation for O1*



*Figure : K map Derivation for O2*



*Figure : K map Derivation for O3*

**4.3 Derived Boolean Output Expression from Karnaugh**

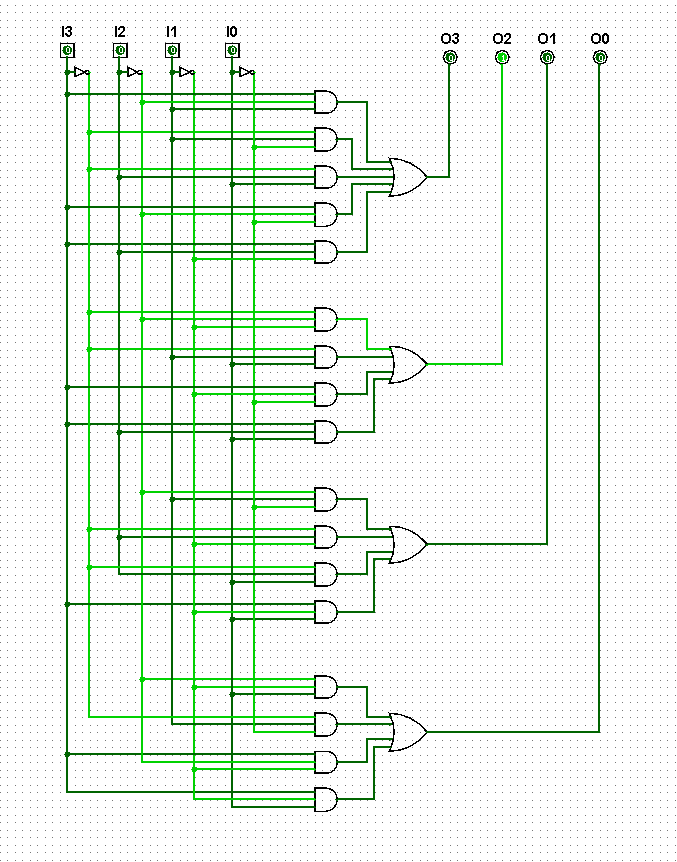
**Map**

The Final Derived Expression for each output from the K maps is:

1. **O0(I3, I2, I1, I0) = I2' I1' I0 + I3' I1 I0' + I3 I2' I1' + I3 I1' I0**
2. **O1(I3, I2, I1, I0) = I2' I1 I0' + I3' I2 I1' + I3' I2 I0 + I3 I1' I0**
3. **O2(I3, I2, I1, I0) = I3' I2' I1' + I3' I1 I0 + I3 I1' I0' + I3 I2 I0**
4. **O3(I3, I2, I1, I0) = I3 I2' I1 + I3' I1 I0' + I3' I2 I0 + I3 I2' I0' + I3 I2 I1'**

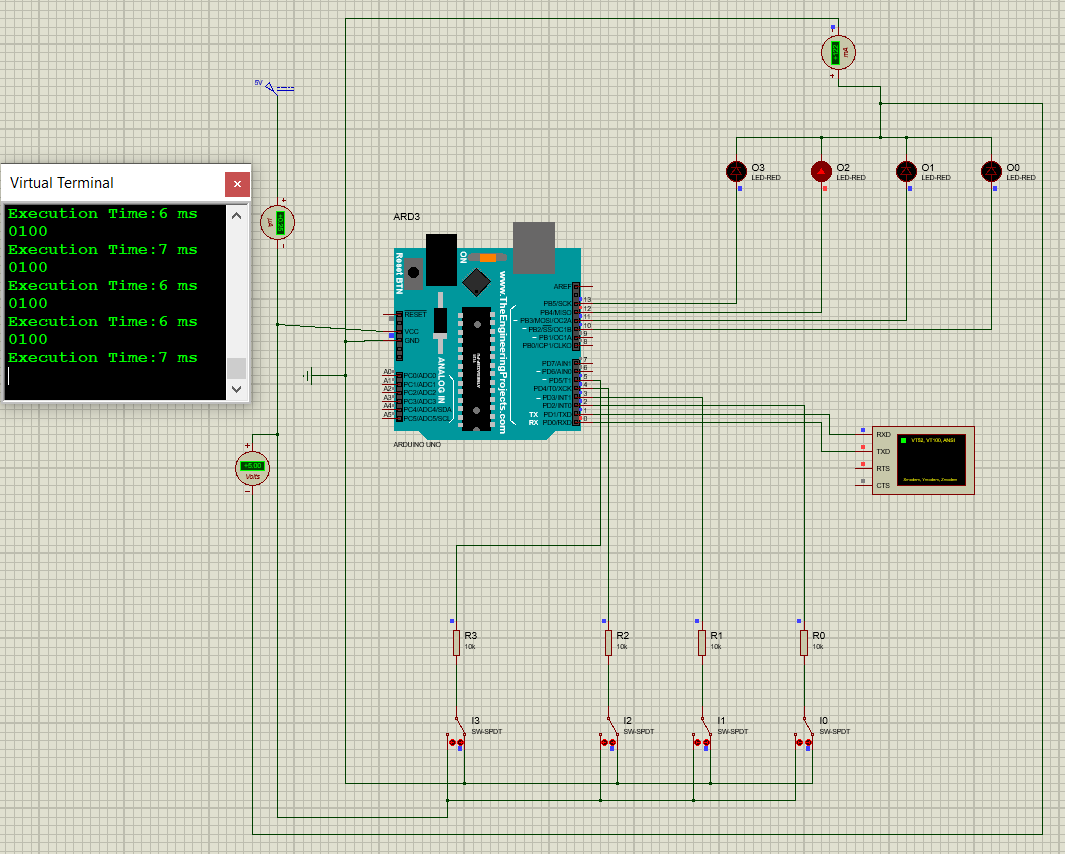
**5.Circuit Diagram with Values of Electrical Components**

**5.1 Logisim Circuit**

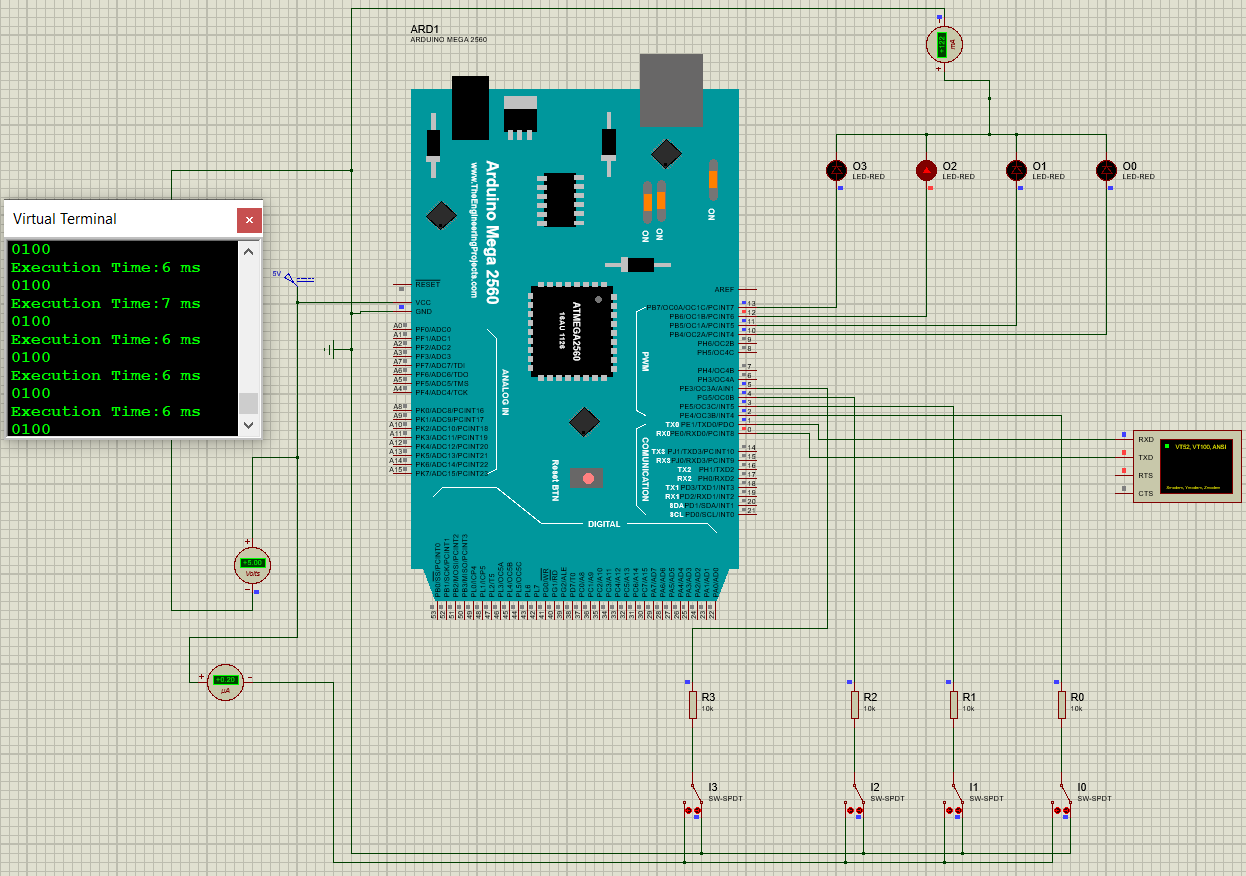
****

*Figure : Logisim Simulation*

**5.2 Proteus Circuit**

**

*Figure : Proteus Simulation Demo for Arduino Uno*

**

*Figure : Proteus Simulation Demo for Arduino Mega 2560*

**6.Circuit Operation Principles**

For our circuit operation the output equation from the Kmap is as follows:

**O0(I3, I2, I1, I0) = I2' I1' I0 + I3' I1 I0' + I3 I2' I1' + I3 I1' I0**

**O1(I3, I2, I1, I0) = I2' I1 I0' + I3' I2 I1' + I3' I2 I0 + I3 I1' I0**

**O2(I3, I2, I1, I0) = I3' I2' I1' + I3' I1 I0 + I3 I1' I0' + I3 I2 I0**

**O3(I3, I2, I1, I0) = I3 I2' I1 + I3' I1 I0' + I3' I2 I0 + I3 I2' I0' + I3 I2 I1'**

As a demo to test whether these expression is correct for our input bits, we designed a logic circuit on Logisim and simulated using four input and output bits and using the help AND, OR, and NOT gates to achieve the output equations.

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

**Arduino Uno**: We have used an Arduino Uno Model as our first board to test.

We will connect the Vcc pin of Arduino to a power supply of 5V. And we will connect the ground pin of the Arduino to a Ground component.

**Arduino Mega**: Used as the second board to test and compare. Connection of Vcc is same as with Arduino Uno.

**Resistors:** We have used a total of 4 resistors of 10k Ohms. These are connected with the SPDT switches.

**Single Pole Double** **Throw Switches**: We used four of these Switches with our Arduino Digital pins. They are arranged as follows:

**Arduino UNO Digital Pin No. 2 --------- SPDT Switch I0**

**Arduino UNO Digital Pin No. 3 --------- SPDT Switch I1**

**Arduino UNO Digital Pin No. 4 --------- SPDT Switch I2**

**Arduino UNO Digital Pin No. 5 --------- SPDT Switch I3**

The Resistors are connected in series with the pins and SPDT switches while the other pole of the switches is connected to the GND. Hence, one side of the switch acts as open while the other acts as closed and provides current flow and power to the pins.

**LED's**: We have connected Four LED's to the Digital Pins of Arduino to receive Output through those Pins. They are arranged in such way:

**Arduino UNO Digital Pin No. 10 --------- LED O0**

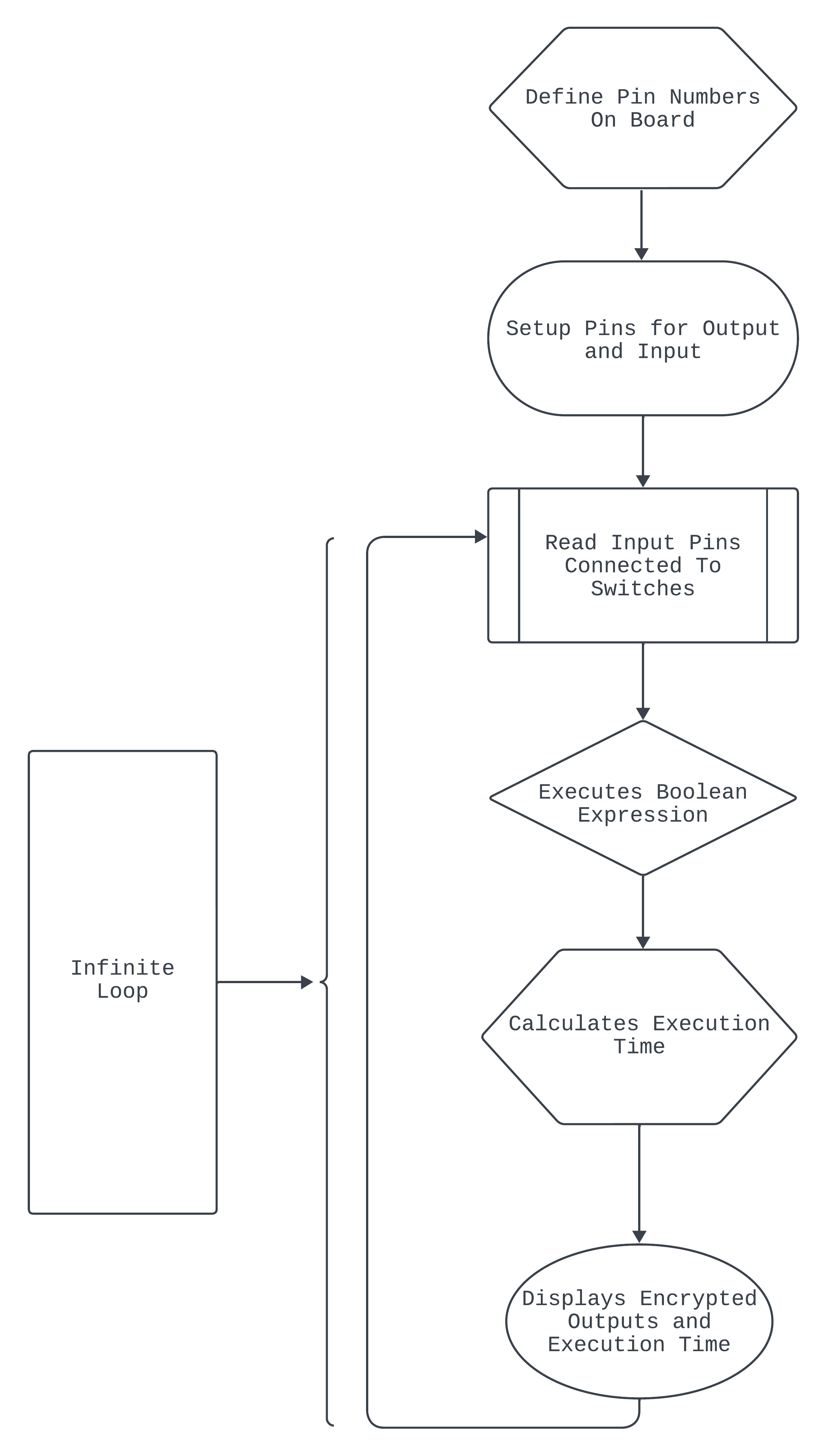
**Arduino UNO Digital Pin No. 11 --------- LED O1**

**Arduino UNO Digital Pin No. 12 --------- LED O2**

**Arduino UNO Digital Pin No. 13 --------- LED O3**

Finally, a Virtual Terminal is used to monitor the execution time to compare performance between Register Level Programming and C-programming. This execution time is analyzed in the Results Analysis section. The Hex file exported from Arduino IDE for both types of programming is imported on the board and simulated.

**7.Program Flow Chart**



*Figure : Program Loop Flowchart*

**8. Arduino Program**

**8.1 Arduino Program for both Arduino UNO R3 and Mega using Arduino Libraries**

**Arduino Code:**

/// output LED variable declaration

int O0LED = 10; /// variable for O0 LED connected to pin 10

int O1LED = 11; /// variable for O1 LED connected to pin 11

int O2LED = 12; // variable for O2 LED connected to pin 12

int O3LED = 13; //variable for O3 LED connected to pin 13

/// switch input variable declaration

int I0Switch = 2; /// variable for switch O0 connected to pin 2

int I1Switch = 3; /// variable for switch O1 connected to pin 3

int I2Switch = 4; /// variable for switch O2 connected to pin 4

int I3Switch = 5; /// variable for switch O3 connected to pin 5

unsigned long startTime;

unsigned long endTime;

void setup()

{

// sets Switches as input

pinMode(I0Switch,INPUT);

pinMode(I1Switch,INPUT);

pinMode(I2Switch,INPUT);

pinMode(I3Switch,INPUT);

// sets LEDs as output

pinMode(O0LED,OUTPUT);

pinMode(O1LED,OUTPUT);

pinMode(O2LED,OUTPUT);

pinMode(O3LED,OUTPUT);

Serial.begin(9600);

}

void loop()

{

startTime = millis();

boolean I0State = digitalRead(I0Switch);

boolean I1State = digitalRead(I1Switch);

boolean I2State = digitalRead(I2Switch);

boolean I3State = digitalRead(I3Switch);

boolean O0State;

boolean O1State;

boolean O2State;

boolean O3State;

// Kmap Equation Logic

O0State= (!I2State & !I1State & I0State)|(!I3State & I1State & !I0State)|(I3State & !I2State & !I1State)|(I3State & !I1State & I0State);

O1State= (!I2State & I1State & !I0State)|(!I3State & I2State & !I1State)|(!I3State & I2State & I0State)|(I3State & !I1State & I0State);

O2State= (!I3State & !I2State & !I1State)|(!I3State & I1State & I0State)|(I3State & !I1State & !I0State)|(I3State & I2State & I0State);

O3State= (I3State & !I2State & I1State )|(!I3State & I1State & !I0State)|(!I3State & I2State & I0State)|(I3State & !I2State& !I0State)|(I3State & I2State & !I1State);

digitalWrite(O0LED,O0State);

digitalWrite(O1LED,O1State);

digitalWrite(O2LED,O2State);

digitalWrite(O3LED,O3State);

Serial.print(O3State);

Serial.print(O2State);

Serial.print(O1State);

Serial.println(O0State);

endTime = millis();

Serial.print("Execution Time:");

Serial.print(endTime-startTime);

Serial.println(" ms");

}

**8.2.Hex Code**

**8.2.1 Hex Codes generated for use on Arduino UNO R3**

**Arduino Library and C programming Hex Code:**

:100000000C9462000C948A000C948A000C948A0070

:100010000C948A000C948A000C948A000C948A0038

:100020000C948A000C948A000C948A000C948A0028

:100030000C948A000C948A000C948A000C948A0018

:100040000C94C3020C948A000C9433030C940D039B

:100050000C948A000C948A000C948A000C948A00F8

:100060000C948A000C948A000000000024002700F1

:100070002A0000000000250028002B0000000000DE

:1000800023002600290004040404040404040202DA

:100090000202020203030303030301020408102007

:1000A0004080010204081020010204081020000012

:1000B0000008000201000003040700000000000027

:1000C0000000E50411241FBECFEFD8E0DEBFCDBF96

:1000D00011E0A0E0B1E0E8E7FAE002C005900D927F

:1000E000AA32B107D9F721E0AAE2B1E001C01D921E

:1000F000A83DB207E1F710E0C2E6D0E004C02197C6

:10010000FE010E943405C136D107C9F70E9465037C

:100110000C943A050C940000833081F028F481306F

:1001200099F08230A9F008958730A9F08830C9F09D

:100130008430B1F4809180008F7D03C08091800075

:100140008F7780938000089584B58F7784BD08955C

:1001500084B58F7DFBCF8091B0008F778093B00006

:1001600008958091B0008F7DF9CFCF93DF93282F32

:1001700030E0F901E255FF4F8491F901E656FF4F57

:10018000D491F901EA57FF4FC491CC23A1F081111A

:100190000E948C00EC2FF0E0EE0FFF1FE458FF4FA1

:1001A000A591B491EC91ED2381E090E009F480E019

:1001B000DF91CF91089580E090E0FACF1F93CF9325

:1001C000DF93282F30E0F901E255FF4F8491F901C8

:1001D000E656FF4FD491F901EA57FF4FC491CC2363

:1001E000A9F0162F81110E948C00EC2FF0E0EE0F89

:1001F000FF1FEE58FF4FA591B4918FB7F894EC9183

:10020000111108C0D095DE23DC938FBFDF91CF9111

:100210001F910895DE2BF8CFCF93DF9390E0FC0180

:10022000E656FF4F24918A579F4FFC018491882303

:10023000D1F090E0880F991FFC01E859FF4FA5917C

:10024000B491FC01EE58FF4FC591D49161110EC0DD

:100250009FB7F8948C91E22FE0958E238C932881A0

:10026000E223E8839FBFDF91CF9108958FB7F89481

:10027000EC91E22BEC938FBFF6CFAF92BF92CF926F

:10028000DF92EF92FF920F931F93CF93DF936C0156

:100290007B018B01040F151FEB015E01AE18BF0837

:1002A000C017D10759F06991D601ED91FC910190E9

:1002B000F081E02DC6010995892B79F7C501DF9101

:1002C000CF911F910F91FF90EF90DF90CF90BF9053

:1002D000AF900895FC01538D448D252F30E0842F7D

:1002E00090E0821B930B541710F0CF96089501975E

:1002F0000895FC01918D828D981761F0A28DAE0F4B

:10030000BF2FB11D5D968C91928D9F5F9F73928FD1

:1003100090E008958FEF9FEF0895FC01918D828DFD

:10032000981731F0828DE80FF11D858D90E00895CA

:100330008FEF9FEF0895FC01918D228D892F90E022

:10034000805C9F4F821B91098F73992708958BE3DF

:1003500091E00E949B0121E0892B09F420E0822F8B

:10036000089580E090E0892B29F00E94A701811177

:100370000C9400000895FC01A48DA80FB92FB11DA5

:10038000A35ABF4F2C91848D90E001968F739927CB

:10039000848FA689B7892C93A089B1898C918370A9

:1003A00080648C93938D848D981306C00288F389A2

:1003B000E02D80818F7D80830895EF92FF920F93CF

:1003C0001F93CF93DF93EC0181E0888F9B8D8C8D01

:1003D00098131AC0E889F989808185FF15C09FB7F5

:1003E000F894EE89FF896083E889F98980818370B8

:1003F000806480839FBF81E090E0DF91CF911F9167

:100400000F91FF90EF900895F62E0B8D10E00F5F87

:100410001F4F0F731127E02E8C8D8E110CC00FB65D

:1004200007FCFACFE889F989808185FFF5CFCE01F5

:100430000E94BB01F1CFEB8DEC0FFD2FF11DE35AB4

:10044000FF4FF0829FB7F8940B8FEA89FB89808178

:100450008062CFCFCF93DF93EC01888D8823B9F0F2

:10046000AA89BB89E889F9898C9185FD03C08081BF

:1004700086FD0DC00FB607FCF7CF8C9185FFF2CF3C

:10048000808185FFEDCFCE010E94BB01E9CFDF91D6

:10049000CF910895FC0101900020E9F73197AF0159

:1004A000481B590BBC018BE391E00C943D018F92EA

:1004B0009F92AF92BF920F931F93CF93DF93CDB7CD

:1004C000DEB7A1970FB6F894DEBF0FBECDBF19A25D

:1004D000423008F44AE08E010F5D1F4F842E912CAC

:1004E000B12CA12CA50194010E941205E62FB9019F

:1004F000CA01EA30F4F4E05DD801EE938D01232BBC

:10050000242B252B79F790E080E0109719F0CD018E

:100510000E944A02A1960FB6F894DEBF0FBECDBF6F

:10052000DF91CF911F910F91BF90AF909F908F90CF

:100530000895E95CE1CFCF92DF92EF92FF92CF93E3

:10054000DF936B017C016DE28BE391E00E94DD01A2

:10055000EC0166277727CB016C197D098E099F096D

:100560004AE00E9457028C0F9D1FDF91CF91FF90B0

:10057000EF90DF90CF900895BC01990F880B990BF5

:100580004AE00C9457021F920F920FB60F9211245B

:100590002F933F938F939F93AF93BF938091370196

:1005A00090913801A0913901B0913A013091360112

:1005B00023E0230F2D3758F50196A11DB11D20937F

:1005C00036018093370190933801A0933901B0939D

:1005D0003A018091320190913301A0913401B091A0

:1005E00035010196A11DB11D809332019093330115

:1005F000A0933401B0933501BF91AF919F918F913A

:100600003F912F910F900FBE0F901F90189526E8E5

:10061000230F0296A11DB11DD2CF1F920F920FB6CC

:100620000F9211242F933F934F935F936F937F9378

:100630008F939F93AF93BF93EF93FF938BE391E0DF

:100640000E94BB01FF91EF91BF91AF919F918F915C

:100650007F916F915F914F913F912F910F900FBEBE

:100660000F901F9018951F920F920FB60F921124A2

:100670002F938F939F93EF93FF93E0914B01F09112

:100680004C018081E0915101F091520182FD1BC02B

:100690009081809154018F5F8F7320915501821753

:1006A00041F0E0915401F0E0E55CFE4F958F8093BE

:1006B0005401FF91EF919F918F912F910F900FBE59

:1006C0000F901F9018958081F4CF789484B5826044

:1006D00084BD84B5816084BD85B5826085BD85B5E6

:1006E000816085BD80916E00816080936E00109264

:1006F00081008091810082608093810080918100DF

:10070000816080938100809180008160809380006F

:100710008091B10084608093B1008091B0008160CD

:100720008093B00080917A00846080937A008091F9

:100730007A00826080937A0080917A008160809351

:100740007A0080917A00806880937A001092C100CC

:1007500060E082E00E940C0160E083E00E940C01F6

:1007600060E084E00E940C0160E085E00E940C01E2

:1007700061E08AE00E940C0161E08BE00E940C01C4

:1007800061E08CE00E940C0161E08DE00E940C01B0

:10079000E0914B01F0914C0182E08083E0914701B0

:1007A000F09148011082E0914901F0914A018FECEB

:1007B000808310925301E0914F01F091500186E047

:1007C0008083E0914D01F0914E01808180618083B2

:1007D000E0914D01F0914E01808188608083E0912D

:1007E0004D01F0914E01808180688083E0914D0140

:1007F000F0914E0180818F7D808300E010E02FB763

:10080000F8948091370190913801A0913901B0910D

:100810003A012FBF80932E0190932F01A0933001B6

:10082000B093310182E00E94B5004C01FF24F394A3

:10083000892B09F4F12C83E00E94B5005C01C1E032

:10084000892B09F4C0E084E00E94B5006C01EE241D

:10085000E394892B09F4E12C85E00E94B500AC01FA

:1008600071E0892B09F470E0D1E0C114D10409F0E2

:10087000D0E031E0A114B10409F030E0AC28BD288B

:1008800081E04115510509F080E0AC2FA823E1E09B

:10089000892809F0E0E09A2F9E23F72FFD23B72F38

:1008A000B323CF2CCB2261E0A114B10409F060E0A6

:1008B000D62EDF20D92A632F6F23D62ADC288E215B

:1008C0006F2D6823DC23DE23DC298323D82BD62B52

:1008D000E722A42AB52A21E0AB2809F020E0AF21C5

:1008E0002A2BBE23B22BFE20FB2A962BCF23C92B0B

:1008F000EF23CE2B3E21C32B6D2D8AE00E94DE001C

:100900006D2F8BE00E94DE006F2D8CE00E94DE00D8

:100910006C2F8DE00E94DE008C2F90E00E94BC02C4

:100920008F2D90E00E94BC028D2F90E00E94BC02AF

:100930006D2D70E090E080E04AE00E94570282E175

:1009400091E00E944A022FB7F8948091370190916C

:100950003801A0913901B0913A012FBF80932A014B

:1009600090932B01A0932C01B0932D0185E191E090

:100970000E944A0260912A0170912B0180912C0102

:1009800090912D01C0902E01D0902F01E090300168

:10099000F09031016C197D098E099F094AE00E948F

:1009A000570285E291E00E944A0282E191E00E94B2

:1009B0004A020115110509F422CF0E94A7018823DC

:1009C00009F41DCF0E9400001ACFEBE3F1E013827F

:1009D000128288EE93E0A0E0B0E084839583A68342

:1009E000B78384E091E09183808385EC90E09587E4

:1009F000848784EC90E09787868780EC90E0918BE9

:100A0000808B81EC90E0938B828B82EC90E0958BD5

:100A1000848B86EC90E0978B868B118E128E138ED2

:100A2000148E0895A1E21A2EAA1BBB1BFD010DC056

:100A3000AA1FBB1FEE1FFF1FA217B307E407F5078E

:100A400020F0A21BB30BE40BF50B661F771F881F6A

:100A5000991F1A9469F760957095809590959B0100

:100A6000AC01BD01CF010895EE0FFF1F0590F49179

:080A7000E02D0994F894FFCF7A

:100A780000000000DD013D016A012A029B017901A5

:100A88008D010D0A00457865637574696F6E205491

:0A0A9800696D653A00206D730000DF

:00000001FF

**8.2.2 Hex Code generated for use on Arduino Mega 2560**

**Arduino Library and C programming Hex Code:**

:100000000C9435010C9466010C9466010C94660105

:100010000C9466010C9466010C9466010C946601C4

:100020000C9466010C9466010C9466010C946601B4

:100030000C9466010C9466010C9466010C946601A4

:100040000C9466010C9466010C9466010C94660194

:100050000C9466010C9466010C9466010C940404E3

:100060000C9466010C9478040C944E040C94660174

:100070000C9466010C9466010C9466010C94660164

:100080000C9466010C9466010C9466010C94660154

:100090000C9466010C9466010C9466010C94660144

:1000A0000C9466010C9466010C9466010C94660134

:1000B0000C9466010C9466010C9466010C94660124

:1000C0000C9466010C9466010C9466010C94660114

:1000D0000C9466010C9466010C9466010C94660104

:1000E0000C9466010C94A7020C94C6010C94AF0109

:1000F0000C94C1010C94AB010C94B5020C94ED026C

:100100000C94B3010C9450060C94A1010C94C701FB

:100110000C9493020C94B9010C943A030C9484024D

:100120000C94CB010C9494010C9490010C949B01C1

:100130000C94A5010C9498010C94BD010C948601BB

:100140000C9457020C948C0100002100240027001D

:100150002A002D00300033000101000004010701D6

:100160000A0100002200250028002B002E0031008B

:10017000340002010000050108010B01000020000D

:100180002300260029002C002F003200000100006F

:10019000030106010901050505050705080808080A

:1001A000020202020A0A080804040404010101010F

:1001B000010101010303030303030303040707070A

:1001C0000C0C0C0C0C0C0C0C0202020206060606AF

:1001D000060606060B0B0B0B0B0B0B0B010210207C

:1001E000200808102040102040800201020108046D

:1001F0000201010204081020408080402010080401

:100200000201800402018040201008040201080459

:10021000020101020408102040800102040810209D

:10022000408000000A0B02090C0D0E0807030401B0

:1002300000000000000000000000000000000000BE

:10024000000000000000000000000000000012118B

:10025000100000000000000000000000000000008E

:100260000000000000000000500611241FBECFEF68

:10027000D1E2DEBFCDBF00E00CBF12E0A0E0B2E0F3

:10028000E2E5FDE000E00BBF02C007900D92AA324C

:10029000B107D9F722E0AAE2B2E001C01D92A83D61

:1002A000B207E1F711E0C5E3D1E000E006C0219715

:1002B0000109802FFE010E949F06C433D10780E010

:1002C0000807A9F70E94AE040C94A7060C9400003E

:1002D0008150823108F05AC0E82FF0E08827EC58AE

:1002E000FE4F8F4F0C949F069401980186018C015C

:1002F0009001C6019B01A101A501AB01AF01B301B2

:10030000B901BD01C601C101C701CB0180918000C7

:100310008F77809380000895809180008F7DF9CF42

:1003200080918000877FF5CF84B58F7784BD089555

:1003300084B58F7DFBCF8091B0008F778093B00024

:1003400008958091B0008F7DF9CF809190008F77D4

:10035000809390000895809190008F7DF9CF8091D7

:100360009000877FF5CF8091A0008F778093A000C9

:1003700008958091A0008F7DF9CF8091A000877FA4

:10038000F5CF809120018F7780932001089580918F

:1003900020018F7DF9CF80912001877FF5CFCF930A

:1003A000DF93282F30E0F901EE5DFD4F8491F901D4

:1003B000E452FE4FD491F901EA56FE4FC491CC238A

:1003C000A1F081110E946801EC2FF0E0EE0FFF1FF9

:1003D000E458FE4FA591B491EC91ED2381E090E0BB

:1003E00009F480E0DF91CF91089580E090E0FACFAA

:1003F0001F93CF93DF93282F30E0F901EE5DFD4F7F

:100400008491F901E452FE4FD491F901EA56FE4F6E

:10041000C491CC23A9F0162F81110E946801EC2F02

:10042000F0E0EE0FFF1FEE59FE4FA591B4918FB78C

:10043000F894EC91111108C0D095DE23DC938FBFA6

:10044000DF91CF911F910895DE2BF8CFCF93DF93EB

:1004500090E0FC01E452FE4F24918A569E4FFC012D

:1004600084918823D1F090E0880F991FFC01E85B0C

:10047000FE4FA591B491FC01EE59FE4FC591D49168

:1004800061110EC09FB7F8948C91E22FE0958E23F6

:100490008C932881E223E8839FBFDF91CF91089559

:1004A0008FB7F894EC91E22BEC938FBFF6CFAF921D

:1004B000BF92CF92DF92EF92FF920F931F93CF9351

:1004C000DF936C017B018B01040F151FEB015E01B3

:1004D000AE18BF08C017D10759F06991D601ED9148

:1004E000FC910190F081E02DC6011995892B79F7D7

:1004F000C501DF91CF911F910F91FF90EF90DF9099

:10050000CF90BF90AF900895FC01538D448D252F5F

:1005100030E0842F90E0821B930B541710F0CF969D

:10052000089501970895FC01918D828D981761F0CF

:10053000A28DAE0FBF2FB11D5D968C91928D9F5FE6

:100540009F73928F90E008958FEF9FEF0895FC01C5

:10055000918D828D981731F0828DE80FF11D858D78

:1005600090E008958FEF9FEF0895FC01918D228D0B

:10057000892F90E0805C9F4F821B91098F73992790

:1005800008958BE392E00E94B50221E0892B09F4E3

:1005900020E0822F0895FC01A48DA80FB92FB11D72

:1005A000A35ABF4F2C91848D90E001968F739927A9

:1005B000848FA689B7892C93A089B1898C91837087

:1005C00080648C93938D848D981306C00288F38980

:1005D000E02D80818F7D80830895EF92FF920F93AD

:1005E0001F93CF93DF93EC0181E0888F9B8D8C8DDF

:1005F00098131AC0E889F989808185FF15C09FB7D3

:10060000F894EE89FF896083E889F9898081837095

:10061000806480839FBF81E090E0DF91CF911F9144

:100620000F91FF90EF900895F62E0B8D10E00F5F65

:100630001F4F0F731127E02E8C8D8E110CC00FB63B

:1006400007FCFACFE889F989808185FFF5CFCE01D3

:100650000E94CB02F1CFEB8DEC0FFD2FF11DE35A81

:10066000FF4FF0829FB7F8940B8FEA89FB89808156

:100670008062CFCFCF93DF93EC01888D8823B9F0D0

:10068000AA89BB89E889F9898C9185FD03C080819D

:1006900086FD0DC00FB607FCF7CF8C9185FFF2CF1A

:1006A000808185FFEDCFCE010E94CB02E9CFDF91A3

:1006B000CF91089580E090E0892B29F00E94C1023B

:1006C00081110E94000080E090E0892B49F080E0D9

:1006D00090E0892B29F00E94000081110E94000007

:1006E00080E090E0892B49F080E090E0892B29F0B0

:1006F0000E94000081110E94000080E090E0892BA0

:1007000049F080E090E0892B29F00E9400008111DF

:100710000C9400000895FC0101900020E9F7319746

:10072000AF01481B590BBC018BE392E00C945702BC

:100730008F929F92AF92BF920F931F93CF93DF93AD

:10074000CDB7DEB7A1970FB6F894DEBF0FBECDBF11

:1007500019A2423008F44AE08E010F5D1F4F842E2B

:10076000912CB12CA12CA50194010E947D06E62FAD

:10077000B901CA01EA30F4F4E05DD801EE938D01CD

:10078000232B242B252B79F790E080E0109719F08C

:10079000CD010E948B03A1960FB6F894DEBF0FBE69

:1007A000CDBFDF91CF911F910F91BF90AF909F90E0

:1007B0008F900895E95CE1CFCF92DF92EF92FF92A4

:1007C000CF93DF936B017C016DE28BE392E00E949B

:1007D000ED02EC0166277727CB016C197D098E09A4

:1007E0009F094AE00E9498038C0F9D1FDF91CF91D3

:1007F000FF90EF90DF90CF900895BC01990F880B88

:10080000990B4AE00C9498031F920F920FB60F9227

:1008100011242F933F938F939F93AF93BF93809116

:10082000370290913802A0913902B0913A0230918A

:10083000360223E0230F2D3758F50196A11DB11D77

:10084000209336028093370290933802A0933902A6

:10085000B0933A028091320290913302A091340217

:10086000B09135020196A11DB11D80933202909383

:100870003302A0933402B0933502BF91AF919F91A0

:100880008F913F912F910F900FBE0F901F90189551

:1008900026E8230F0296A11DB11DD2CF1F920F9201

:1008A0000FB60F9211240BB60F922F933F934F93D5

:1008B0005F936F937F938F939F93AF93BF93EF93C8

:1008C000FF938BE392E00E94CB02FF91EF91BF91E7

:1008D000AF919F918F917F916F915F914F913F91D8

:1008E0002F910F900BBE0F900FBE0F901F90189579

:1008F0001F920F920FB60F9211240BB60F922F93E7

:100900008F939F93EF93FF93E0914B02F0914C02F2

:100910008081E0915102F091520282FD1DC09081D0

:10092000809154028F5F8F7320915502821741F09E

:10093000E0915402F0E0E55CFD4F958F8093540206

:10094000FF91EF919F918F912F910F900BBE0F9080

:100950000FBE0F901F9018958081F2CF789484B5C8

:10096000826084BD84B5816084BD85B5826085BDAB

:1009700085B5816085BD80916E00816080936E0039

:10098000109281008091810082608093810080912B

:1009900081008160809381008091800081608093DC

:1009A00080008091B10084608093B1008091B0009C

:1009B00081608093B000809191008260809391006B

:1009C000809191008160809391008091900081607E

:1009D000809390008091A10082608093A10080911B

:1009E000A10081608093A1008091A000816080932C

:1009F000A0008091210182608093210180912101DA

:100A00008160809321018091200181608093200189

:100A100080917A00846080937A0080917A0082606D

:100A200080937A0080917A00816080937A0080912F

:100A30007A00806880937A001092C10060E082E0C2

:100A40000E94260260E083E00E94260260E084E0CB

:100A50000E94260260E085E00E94260261E08AE0B2

:100A60000E94260261E08BE00E94260261E08CE099

:100A70000E94260261E08DE00E942602E0914B0276

:100A8000F0914C0282E08083E0914702F0914802AD

:100A90001082E0914902F0914A028FEC808310921B

:100AA0005302E0914F02F091500286E08083E09182

:100AB0004D02F0914E02808180618083E0914D0271

:100AC000F0914E02808188608083E0914D02F09128

:100AD0004E02808180688083E0914D02F0914E0249

:100AE00080818F7D80832FB7F89480913702909119

:100AF0003802A0913902B0913A022FBF80932E02A2

:100B000090932F02A0933002B093310282E00E94B2

:100B1000CF015C0111E0892B09F410E083E00E9411

:100B2000CF016C0101E0892B09F400E084E00E9410

:100B3000CF017C0199249394892B09F4912C85E0B1

:100B40000E94CF01AC01F1E0892B09F4F0E0D1E083

:100B5000E114F10409F0D0E031E0C114D10409F04E

:100B600030E0CE28DF2881E04115510509F080E012

:100B7000A02FA82371E0AB2809F070E09A2F9723EB

:100B8000EF2FED23BF2FB323E12EEB2261E0C11441

:100B9000D10409F060E0F62EF122F92A632F6E23CA

:100BA000F62AFE288921C12FC823D023D723DE2986

:100BB0008323D82BDC2B9F22C42AD52A21E0CD28E1

:100BC00009F020E0A1232A2BB7232B2B1921122B6C

:100BD0009C2BC02FCE23C92B7E23C72B3921C32B9F

:100BE0006F2D8AE00E94F8016D2F8BE00E94F801C2

:100BF000612F8CE00E94F8016C2F8DE00E94F801BB

:100C00008C2F90E00E94FD03812F90E00E94FD0355

:100C10008D2F90E00E94FD036F2D70E090E080E04A

:100C20004AE00E94980382E192E00E948B032FB772

:100C3000F8948091370290913802A0913902B091D6

:100C40003A022FBF80932A0290932B02A0932C028A

:100C5000B0932D0285E192E00E948B0360912A02FD

:100C600070912B0280912C0290912D0200912E0206

:100C700010912F022091300230913102601B710BD4

:100C8000820B930B4AE00E94980385E292E00E9457

:100C90008B0382E192E00E948B030E945A0323CFD0

:100CA000EBE3F2E01382128288EE93E0A0E0B0E082

:100CB00084839583A683B78384E092E091838083C5

:100CC00085EC90E09587848784EC90E09787868711

:100CD00080EC90E0918B808B81EC90E0938B828B09

:100CE00082EC90E0958B848B86EC90E0978B868BE2

:100CF000118E128E138E148E0895A1E21A2EAA1B45

:100D0000BB1BFD010DC0AA1FBB1FEE1FFF1FA217BB

:100D1000B307E407F50720F0A21BB30BE40BF50BB8

:100D2000661F771F881F991F1A9469F76095709541

:100D3000809590959B01AC01BD01CF010895EE0F08

:100D4000FF1F881F8BBF0790F691E02D1994F89430

:020D5000FFCFD3

:100D520000000000ED02570284023A03B50293023A

:100D6200A7020D0A00457865637574696F6E205499

:0A0D7200696D653A00206D73000002

:00000001FF

**9.Result Analysis and Comparison Among The Boards Used**

**9.1 Hardware Comparison Among Boards Used**

|  |  |  |
| --- | --- | --- |
| **Hardware Specifications** | **Arduino Uno R3** | **Arduino Mega 2560** |
| **Micro controller Used** | ATmega328p | ATmega2560 |
| **Flash Memory** | 32 KB | 256KB |
| **EEPROM** | 1 KB | 4KB |
| **Clock Frequency** | 16 MHz | 16 MHz |
| **PWM pins** | 6 | 15 |
| **Analog Pins** | 6 | 16 |
| **Digital PIns** | 14 | 54 |
| **SRAM** | 2 KB | 8 KB |
| **Power** | 5V | 5V |

*Table : Hardware Specification of Arduino UNO R3 and Arduino Mega 2560*

**9.2 Power Consumption Comparison:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **InPut Bits** | **OutPut Bits** | **Current ( mA)** | **Voltage (V)** | **Power (mW)** |
| 0000 | 0100 | 365 | 5 | 1825 |
| 1000 | 1101 | 365 | 5 | 1825 |
| 0100 | 0010 | 243 | 5 | 1215 |
| 1100 | 1100 | 243 | 5 | 1215 |
| 0010 | 1011 | 243 | 5 | 1215 |
| 1010 | 1011 | 122 | 5 | 610 |
| 0110 | 1001 | 243 | 5 | 1215 |
| 1110 | 0000 | 243 | 5 | 1215 |
| 0001 | 0101 | 243 | 5 | 1215 |
| 1001 | 0011 | 243 | 5 | 1215 |
| 0101 | 1010 | 243 | 5 | 1215 |
| 1101 | 1111 | 122 | 5 | 610 |
| 0011 | 0100 | 365 | 5 | 1825 |
| 1011 | 1000 | 243 | 5 | 1215 |
| 0111 | 1110 | 356 | 5 | 1825 |
| 1111 | 0100 | 122 | 5 | 610 |

Table:Current Voltage and Power use for **Arduino UNO R3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **InPut Bits** | **OutPut Bits** | **Current ( mA)** | **Voltage (V)** | **Power (mW)** |
| 0000 | 0100 | 365 | 5 | 1825 |
| 1000 | 1101 | 365 | 5 | 1825 |
| 0100 | 0010 | 243 | 5 | 1215 |
| 1100 | 1100 | 243 | 5 | 1215 |
| 0010 | 1011 | 243 | 5 | 1215 |
| 1010 | 1011 | 122 | 5 | 610 |
| 0110 | 1001 | 243 | 5 | 1215 |
| 1110 | 0000 | 243 | 5 | 1215 |
| 0001 | 0101 | 243 | 5 | 1215 |
| 1001 | 0011 | 243 | 5 | 1215 |
| 0101 | 1010 | 243 | 5 | 1215 |
| 1101 | 1111 | 122 | 5 | 610 |
| 0011 | 0100 | 365 | 5 | 1825 |
| 1011 | 1000 | 243 | 5 | 1215 |
| 0111 | 1110 | 356 | 5 | 1825 |
| 1111 | 0100 | 122 | 5 | 610 |

Table: Current Voltage and Power use for **Arduino Mega 2560**

Both Arduino UNO R3 and Mega show the same current draw, voltage, and Power consumption since both use the same number of components and resistors connected to it and both have same 5V power input via DC Generator. Increasing supplied power increases the current drawn.

|  |  |  |
| --- | --- | --- |
|  | **Arduino UNO R3** | **Arduino Mega 2560** |
| **C Programming** | **C Programming** |
| **Program Memory Size** | 2750 bytes | 3474 bytes |
| **Local Variable Size Left** | 1832 bytes | 7976 bytes |
| **Global Variable Size** | 216 bytes | 216 bytes |
| **Program Execution Time** | 6-7 milliseconds | 6-7 milliseconds |

**9.3 Memory Size and Execution Time Comparison**

Table: Program memory Size, Variable Size and Program Execution Time using

**Arduino Uno R3** and **Arduino Mega 2560**

The program when used with Arduino UNO R3 uses 2750 bytes in C programming .

Out of a maximum of 32256 bytes. Hence uses only 8.52% of memory , respectively. From a maximum of 2048 bytes for variable size, C programme use 216 bytes for Global Variables, allocating 1832 bytes for local variables when coded with C language,

When used with Arduino Mega 2560, the C programme takes up 3474 bytes out of 253952 bytes which is only 1.3% of the maximum program memory size.

It also uses 222 bytes of variable memory for Global variable out of a maximum of 8192 bytes, leaving 7970 bytes for local variables,.

Moreover, when C programming is used. The code executes in 6-7 milliseconds. Here program execution time means the time between switch reading to LED lighting output

**10 Conclusion;**

This project's primary goal was to examine the power consumption of various microcontroller boards (Arduino Uno R3 and Arduino Mega 2560 ) for a unique situation. Applying a 4-bit logic input in the microcontrollers is a particular case.

This experiment shows that C-programming is much easier to code for due to its massive libraries and functions. Both boards show that the program size occupies less memory and has a faster execution time.

It can be concluded that the data and results of this project are reliable.

.

**11.Questions and Answers**

**11.1 Arduino UNO**

**1. What is the clock frequency of the micro controller used?**

Answer: 16 MHz is the clock frequency of the micro controller.

**2. What is the data bus width of the micro controller used?**

Answer: The data bus width of the micro controller used is 8 bit.

**3. What is the size of your hex file generated?**

Answer: For default programming- 8KB, for Register Level Programming- 6KB

**4. Can the project be implemented by using interrupt?**

Answer: Yes, since Arduino UNO pins support interrupt use. External Interrupt are supported on digital pin 2 and 3 while Pin Change interrupt are supported on any of the 20 pins, A0-A5 and D0-D13.

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

Answer: Yes, the main routine is required to be an infinite loop. Since we need to continuously provide input using switches and monitor the output accordingly as well as monitor the execution time continuously, for each input and output. Without an infinite loop, the program would run once and then terminate.

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

Answer: Since we are using SPDT switch, one terminal connects to ground and the other connects to input pins. There is no difference between level trigger and edge trigger in this project. The toggling switch provides either an ON-1(ON/HIGH) signal or ON-2 (OFF/LOW) signal.

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

Answer: The project refers to an encryption circuit from input to output where an encryption algorithm is applied to a given piece of information. Here, input of 4 bits corresponds to a unique 4-bit output, encrypting the original input.

**11.2 Arduino Mega 2560**

**1. What is the clock frequency of the micro controller used?**

Answer: 16 MHz is the clock frequency of the micro controller.

**2. What is the data bus width of the micro controller used?**

Answer: The data bus width of the micro controller used is 8 bit.

**3. What is the size of your hex file generated?**

Answer: For default programming- 10KB, for Register Level Programming- 7KB

**4. Can the project be implemented by using interrupt?**

Answer: Yes, since Arduino Mega 2560 pins support interrupt use. It supports up to 6 external interrupt on digital pin 2,3,18,19,20,21. Like UNO, Pin Change Interrupt can be implemented on all pins.

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

Answer: Yes, the main routine is required to be an infinite loop. Since we need to continuously provide input using switches and monitor the output accordingly as well as monitor the execution time continuously, for each input and output. Without an infinite loop, the program would run once and then terminate.

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

Answer: Since we are using SPDT switch, one terminal connects to ground and the other connects to input pins. There is no difference between level trigger and edge trigger in this project. The toggling switch provides either an ON-1(ON/HIGH) signal or ON-2 (OFF/LOW) signal.

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

Answer: The project refers to an encryption circuit from input to output where an encryption algorithm is applied to a given piece of information. Here, input of 4 bits corresponds to a unique 4-bit output, encrypting the original input.

**12.References:**

1. Materials provided by Dr. Dihan Md. Nuruddin Hassan (DMH).
2. Differentiate between arduino uno and arduino mega - Semiconductor for You (semiconductorforu.com)
3. Tinkercad + Arduino Lesson 4: Blink an LED (DDR and PORT registers) - YouTube
4. Arduino Uno Vs Nano Vs Mega, Pinout, and technical Specifications (electroniclinic.com)

6. Arduino Comparison Guide - SparkFun Learn

**13.Resources:**

**1.Project Materials Drive link:**

<https://drive.google.com/drive/folders/1BPo7y-ts3qASD2YVE2BX_ll3AU37RCXs?usp=sharing>

**2.Project Demonstration Video Drive link:**

<https://drive.google.com/drive/folders/16KjSFGUHJxN5Bwj6nfHyR60_ToxIYVUd?usp=sharing>

**3.Project members and their contributions:**

|  |  |
| --- | --- |
| Ferdous Reza Niloy | Wrote the code for both Arduino Uno R3 and  mega2560 , hex code ,  designed Arduino mega2560 in proteus |
| Rafiduzzaman Niloy | Result Analysis and Comparison Among The Boards Used |
| Md. Afjal Hossain Dali | K map derivation and designed the logisim circuit |
| Sanjana Islam | Wrote the lab report and designed  Arduino Uno R3 in proteus |
| Afsara Waziha | Wrote the lab report , circuit operation  principles , designed the flowchart |
| Faizul Islam Bhuiyan | Conclusion and Question Answer |