**LAB 01**

**INTRODUCTION TO SINGLE BOARD MICROCONTROLLER ,ARDUINO PROGRAMMING AND DIGITAL HARDWARE INTERFACING**

**Task No. 1: Write a sketch to interface Arduino with LED, LED should blink with a delay of 1 second.**

**Solution:**

int LED = 7;

void setup() {

pinMode(LED, OUTPUT);

}

void loop() {

digitalWrite(LED, HIGH);

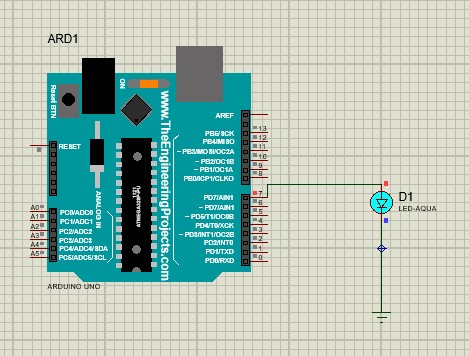
delay(1000);

digitalWrite(LED, LOW);

delay(1000);

}

**Output:**



**Task No. 2: Write a sketch to interface Arduino with SPDT switch & LED.**

**Solution:**

**Output:**

int LED = 7;

int SW = 12;

void setup() {

pinMode(LED, OUTPUT);

pinMode(SW, INPUT);

}

void loop() {

int buttonState = digitalRead(SW);

if (buttonState == HIGH) {

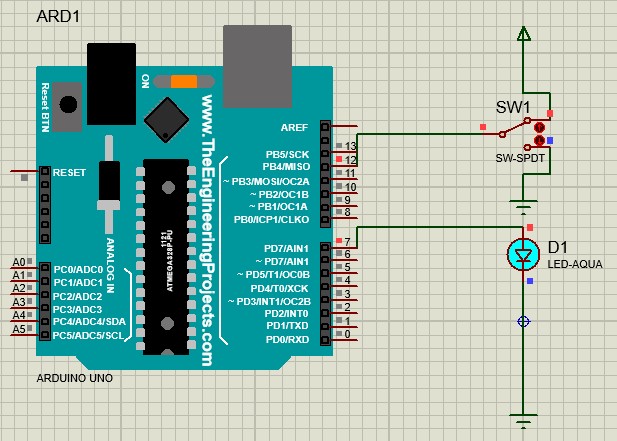
digitalWrite(LED, HIGH);

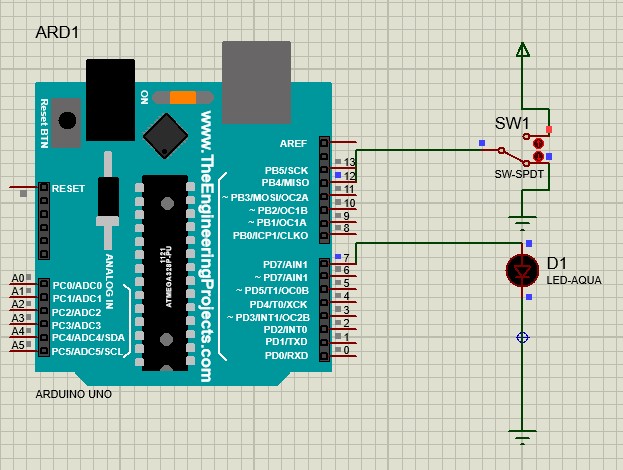
} else {

digitalWrite(LED, LOW);

}

}





**LAB 02**

**ARDUINO INTERFACING WITH RELAY AND SEVEN SEGMENT DISPLAY**

**Task No. 1: Write a sketch to interface Arduino with the Relay. The Relay should be controlled by a SPDT Switch.**

**Solution:**

int buttonPin = 13; int relayPin = 3; void setup() {

pinMode(relayPin, OUTPUT);

pinMode(buttonPin, INPUT);

}

void loop() {

int buttonState = digitalRead(buttonPin); if (buttonState == HIGH)

{

digitalWrite(relayPin, HIGH);

}

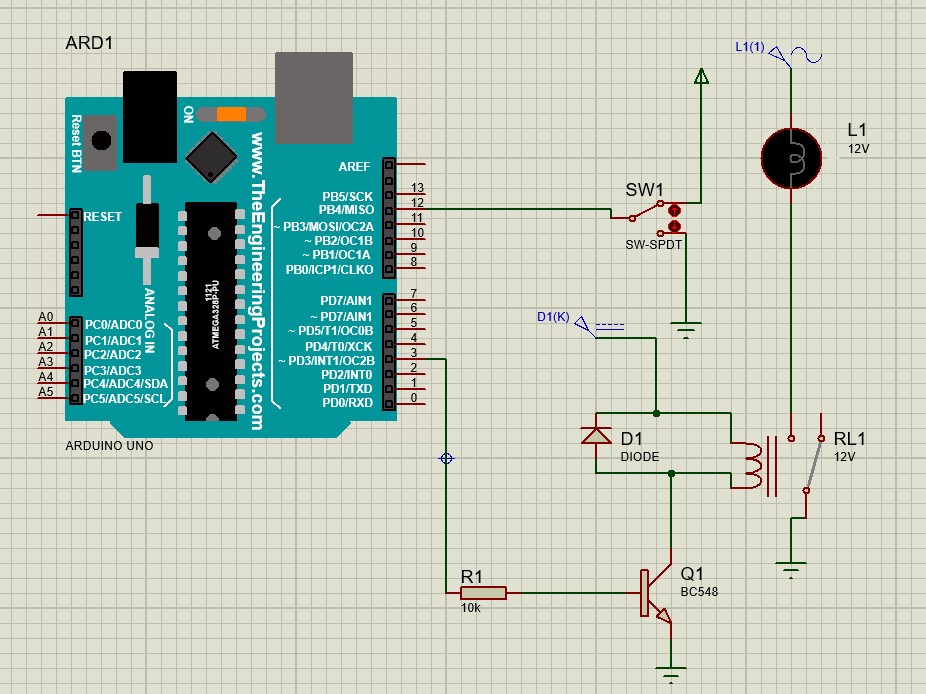
else {

digitalWrite(relayPin, LOW);

}

}

**Output:**



**Task No. 2: Write a sketch to interface Arduino with Seven Segment Display. It should work as a decade counter. The Start / Stop of counting should be controlled through a SPDT Switch. Solution:**

**Output:**

#define segA 2

#define segB 3

#define segC 4

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, LOW);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, LOW);

digitalWrite(segG, HIGH);

break;

case 3:

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, HIGH);

break;

case 4:

// Display 4

digitalWrite(segA, LOW);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 5:

// Display 5

digitalWrite(segA, HIGH);

digitalWrite(segB, LOW);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, LOW);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 6:

// Display 6

digitalWrite(segA, HIGH);

digitalWrite(segB, LOW);

#define segD 5

#define segE 6

#define segF 7

#define segG 8

#define button 10

int COUNT = 0;

int ButtonState;

void setup() {

for (int i = 2; i <= 8; i++) {

pinMode(i, OUTPUT);

}

pinMode(button, INPUT);

}

void loop() {

ButtonState = digitalRead(button);

if (ButtonState == HIGH) {

switch (COUNT) {

case 0:

// Display 0

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, LOW);

break;

case 1:

// Display 1

digitalWrite(segA, LOW);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

case 2:

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 7:

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

break;

case 8:

digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, HIGH);

digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

case 9:

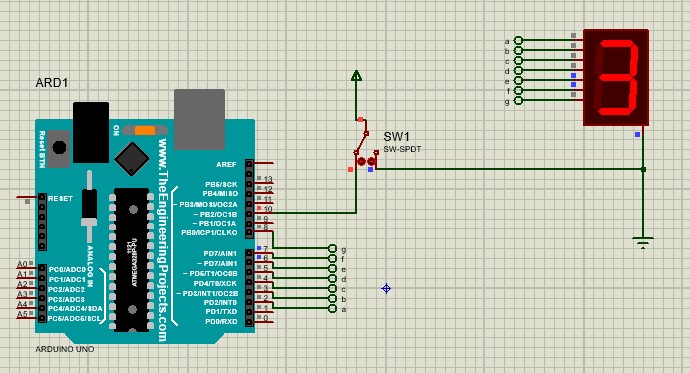
digitalWrite(segA, HIGH);

digitalWrite(segB, HIGH);

digitalWrite(segC, HIGH);

digitalWrite(segD, HIGH);

digitalWrite(segE, LOW);



digitalWrite(segF, HIGH);

digitalWrite(segG, HIGH);

break;

}

if (COUNT < 9) {

COUNT++;

delay(1000);

} else {

COUNT = 0;

delay(1000);

}

}

}

**Task No. 1: Write a sketch to interface Arduino with 16x2 Liquid Crystal Display (LCD). Write the name of your course “Embedded Systems” in the 1st Line and your Section “BSE - 5B” in the 2nd Line of LCD. This Text should blink with a delay of 0.5 seconds.**

**Solution:**

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

lcd.begin(16, 2); // Setting Up the LCD No. of Rows & Columns

}

void loop() {

lcd.setCursor(0, 0);

lcd.print("Embedded Systems");

lcd.setCursor(0, 1);

lcd.print("BSE - 5B");

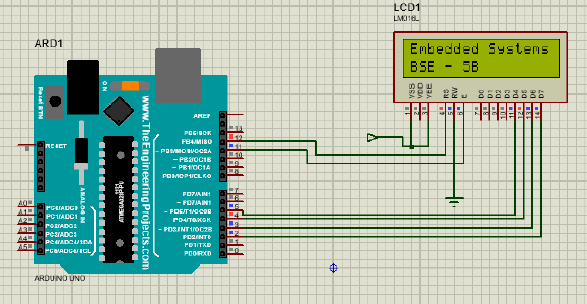
delay(500);

lcd.clear();

delay(5000);

}

**Output:**



**Task No. 2: Write a sketch to interface Arduino with 16x2 Liquid Crystal Display (LCD). First line of LCD should display your name, second line of LCD should display your registration number, and text in both line should keep moving from left to right.**

**Solution:**

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

lcd.begin(16, 2);

}

void loop() {

lcd.setCursor(0, 0);

lcd.print("Embedded Systems");

lcd.setCursor(0, 1);

lcd.print("BSE - 5B");

delay(500);

for (int i = 0; i < 29; i++) {

lcd.scrollDisplayRight();

delay(100);

}

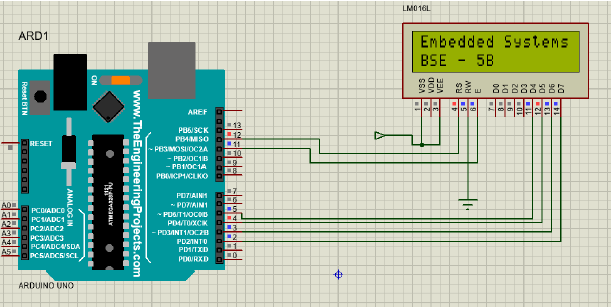
delay(1000);

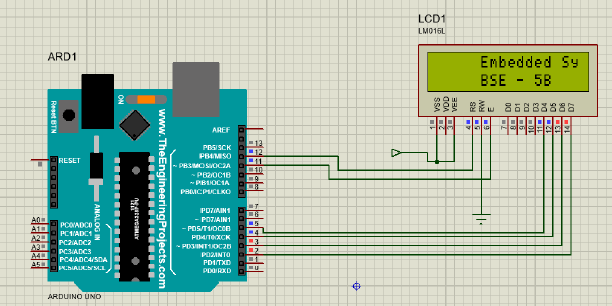
lcd.clear();

delay(5000);

}

**Output:**

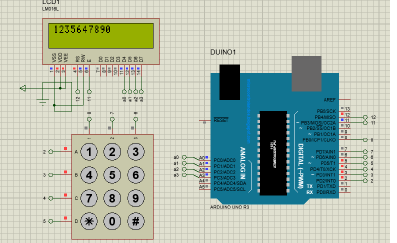




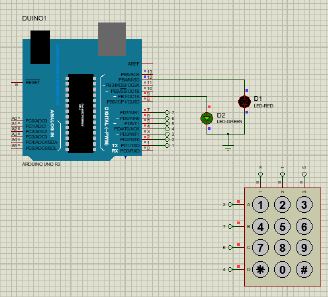
**LAB 04**

**Task No. 1: Write a sketch to interface Arduino with a 3 x 4 Matrix Keypad. The display of the pressed key should be displayed on the LCD.**

**Solution:**

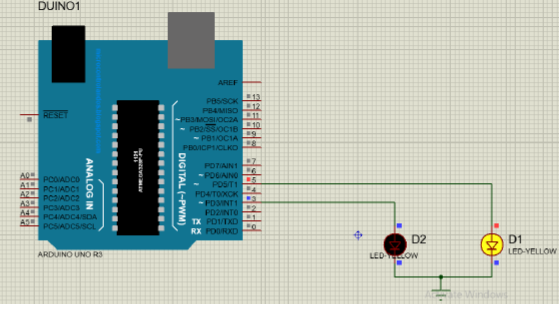


**Task No. 2: Write a sketch that works as security keypad lock. Set any password. If the input password matches with the set password, Green LED should glow, otherwise Red LED will glow.**

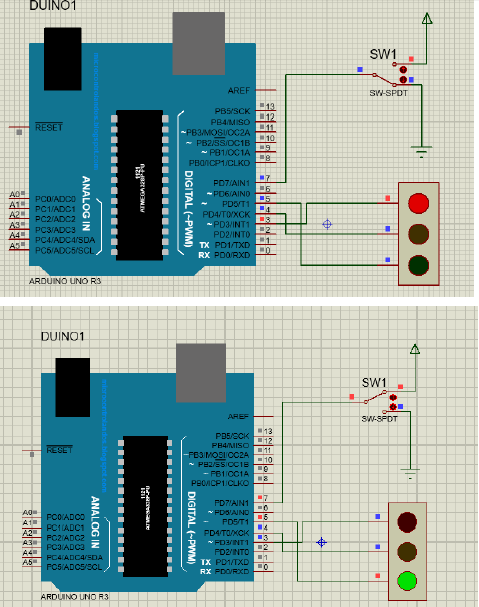


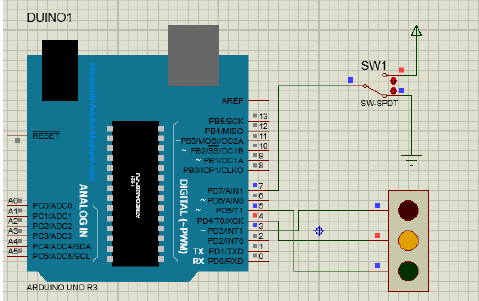
**Lab 05**

**Task 1: Write a sketch to blink the 2 LEDs interfaced with Arduino at a different rate simultaneously. (i.e. “delay” function limits the designer to perform multitasking from the controller, so this sketch is implement without utilizing this function).**



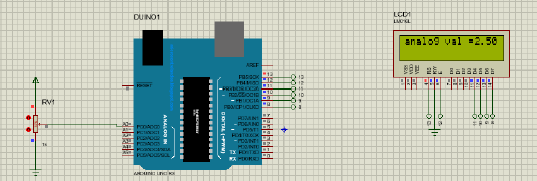
**Task 2: Write a sketch to implement the one-way traffic light controller using FSM concepts. The sensor will work to sense the traffic on the road whose output will be the stimulus for the state transition.**



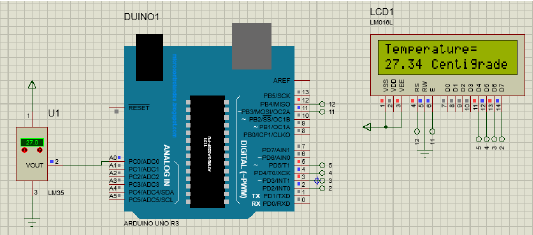


**LAB 06**

**Task 1: Write a program to interface potentiometer with analog pin of Arduino Uno to read analog values and display it on LCD.**

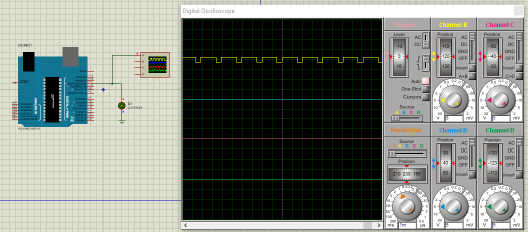


**Task 2: Write a sketch to interface Arduino with the Temperature Sensor (LM35). The value of the Temperature should be displayed on the LCD.**

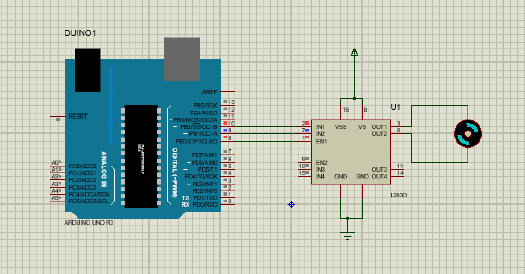


**LAB 07**

**Task 1: Write a program to generate a PWM signal with duty cycles (25%, 50%, 75% & 100%).**

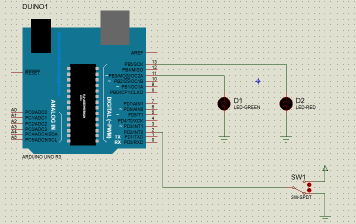
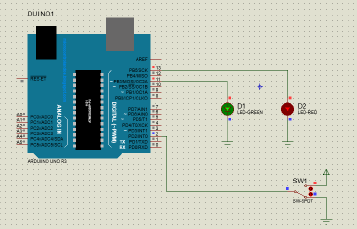
****

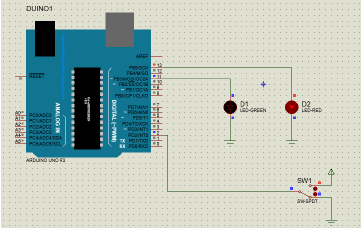
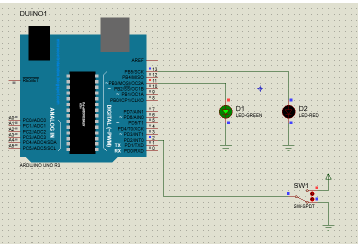
**Task 2: Write a program to interface DC motor with Arduino Uno.**

****

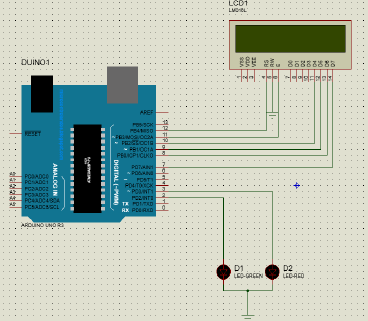
**LAB 08**

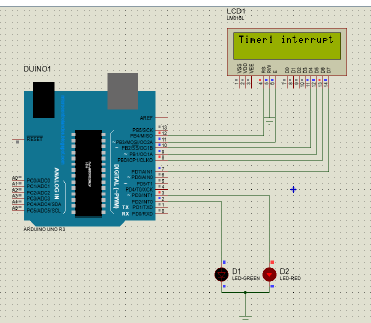
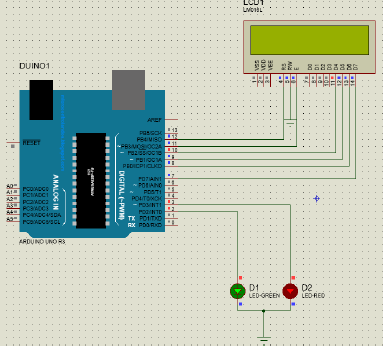
**Task 1: In this lab, we will use external interrupt in Arduino UNO.**

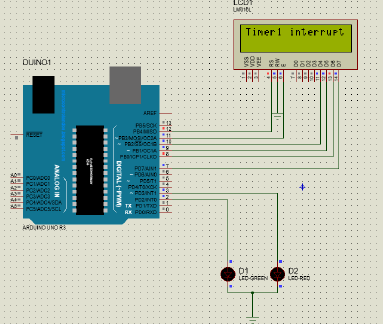
 

**Task 2: Write a program to use timer interrupt in Arduino Uno.**





**Task 3: Using the concept of interrupts you have learnt, develop a program that uses interrupt.**

int i=0;

int x=0;

void setup() {

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

pinMode(11,OUTPUT);

pinMode(13,OUTPUT);

pinMode(2,INPUT);

attachInterrupt(digitalPinToInterrupt(2),routine,CHANGE);

}

void loop() {

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

digitalWrite(11,HIGH);

delay(1000);

digitalWrite(11,LOW);

delay(1000);

}

void routine()

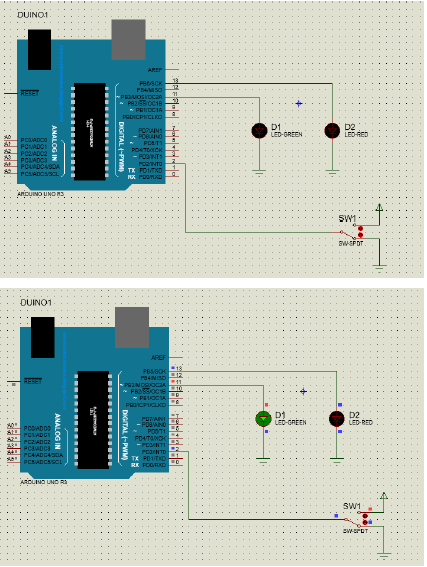
{

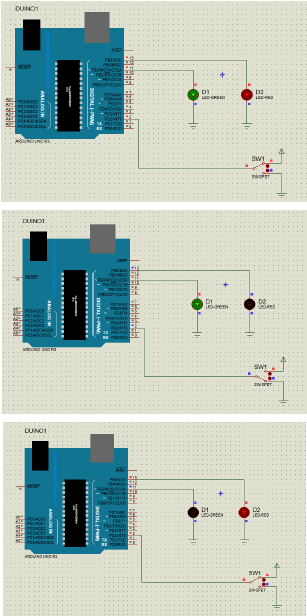
x= ~i;

i=x;

digitalWrite(13,i);

}





**LAB 09**

**Task 1: Using the Concept of SPI communication, write a program to control a LED on slave arduino by a SPDT switch on master arduino. Attach proteus simulation results and arduino code.**

**Code Master:**

#include <SPI.h>

int pin = 9;

void setup() {

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

digitalWrite(SS,HIGH);

SPI.begin();

SPI.setClockDivider(SPI\_CLOCK\_DIV2);

pinMode(pin,INPUT);

}

int state;

int oldstate = 0;

void loop() {

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

int sw= digitalRead(pin);

if(sw==HIGH)

{

int c;

state =! oldstate;

digitalWrite(SS,LOW);

c=state;

SPI.transfer(c);

digitalWrite(SS,HIGH);

oldstate = state;

delay(500);

}

}

**Code Slave:**

#include <SPI.h>

int buff;

volatile boolean process;

//SLAVE

void setup() {

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

pinMode(MISO,OUTPUT);

SPCR = \_BV(SPE);

process = false;

SPI.attachInterrupt();

pinMode(7,OUTPUT);

}

ISR(SPI\_STC\_vect)

if(process)

{

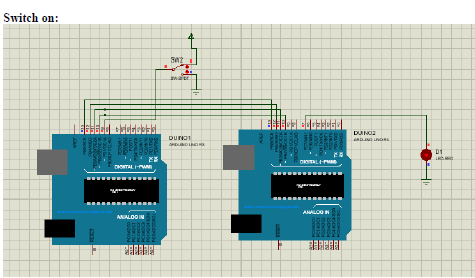
process=false;

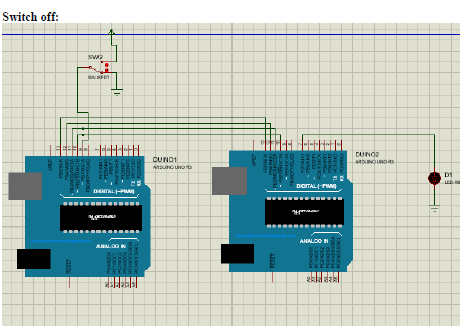
digitalWrite(7,buff);

}

}

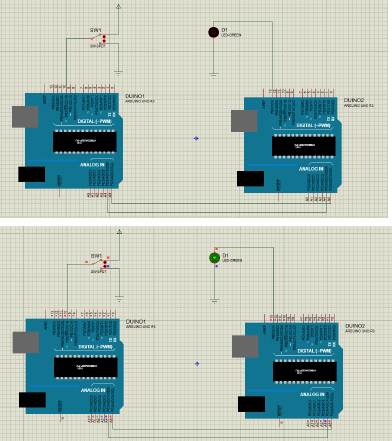
**OUTPUT:**





**LAB 10**

**Task 1: In this task we will control the LED on Slave Arduino using SPDT switch on Master Arduino through I2C communication.**

****

**Task 2: Develop a Program which provide the use of Inter integrated 12C Interface programming.**

**Master:**

#include <Wire.h>

#define ledPin 9

byte rcvData;

int potValue;

void setup()

{

Wire.begin();

rcvData = 255;

pinMode(ledPin, OUTPUT);

}

void loop()

{

potValue = analogRead(A0);

potValue = map(potValue, 0, 1023, 0, 255);

Wire.beginTransmission(0x14);

Wire.write(potValue);

Wire.endTransmission();

Wire.requestFrom(0x14, 1);

if(Wire.available())

{

rcvData = Wire.read();

}

analogWrite(ledPin, rcvData);

}

**Slave:**

#include <Wire.h>

#define ledPin 9

byte rcvData;

int potValue;

void setup()

{

Wire.begin(0x14);

/\*Event Handlers\*/

Wire.onReceive(DataReceive);

Wire.onRequest(DataRequest);

rcvData = 255;

pinMode(ledPin, OUTPUT);

}

void loop()

{

potValue = analogRead(A0);

potValue = map(potValue, 0, 1023, 0, 255);

analogWrite(ledPin, rcvData);

}

void DataReceive(int numBytes)

{

while(Wire.available())

{

rcvData = Wire.read();

}

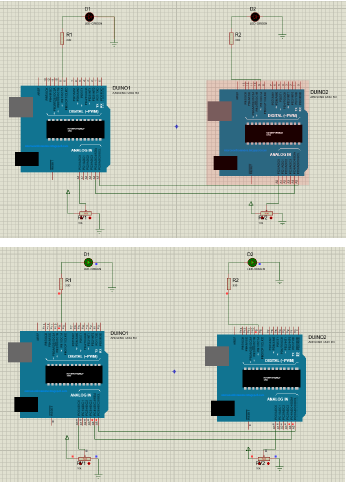
}

void DataRequest()

{

Wire.write(potValue);

}

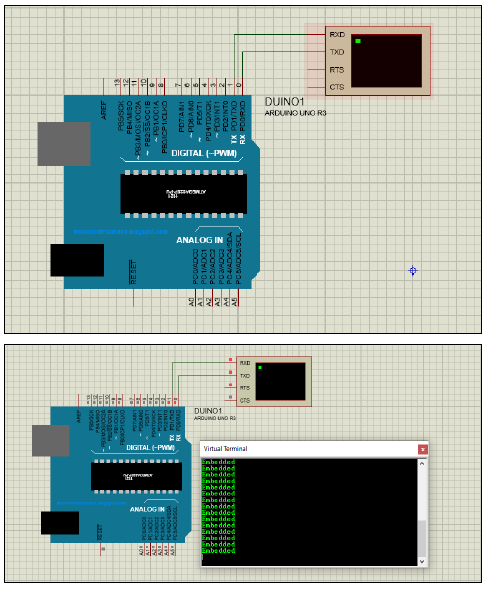


**LAB 11**

**Task1:**

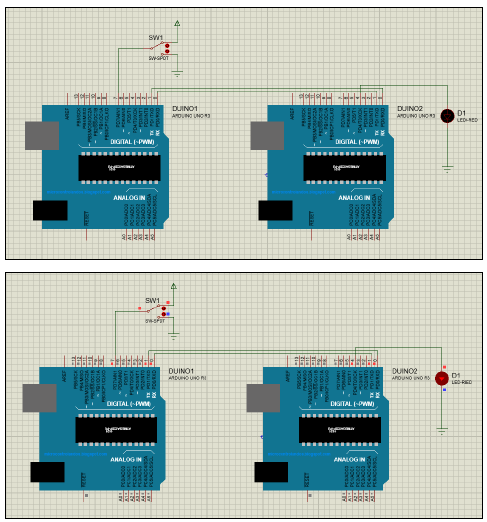
**In this task, we will write a program to print a string on serial monitor.**

**Solution:**

****

**Task 2:**

**In this task we will control the LED on “Arduino 2” using SPDT switch on “Arduino 1” through UART communication.**

****

**Task 3: Develop a Program which provide serial usart interfacing programming using Arduino UNO board.**

**Solution:**

**Master:**

#include <SoftwareSerial.h>

SoftwareSerial softSerial(10, 11);

void setup()

{

softSerial.begin(9600);

}

void loop()

{

softSerial.write("UART Communication");

delay (100);

}

**Slave:**

#include <SoftwareSerial.h>

SoftwareSerial softSerial(10, 11);

char ip;

void setup()

{

softSerial.begin(9600);

}

void loop()

{

if (softSerial.available())

{

ip=softSerial.read();

softSerial.print(ip);

}

} 