Bahria University,

Karachi Campus



COURSE: CEL-439 EMBEDDED SYSTEMS

TERM: FALL 2022, CLASS: BSE- 5(B)

Submitted By:

Abdul Quddos (02-131202-033)

(Name) (Reg. No.)

Submitted To:

**Engr. Syed Zia Uddin/Engr. Rizwan Fazal**

Signed Remarks: Score:

INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNO | DATE | LAB NO | LAB OBJECTIVE | SIGN |
| 1 |  | 1 |  |  |
| 2 |  | 2 |  |  |
| 3 |  | 3 |  |  |
| 4 |  | 4 |  |  |
| 5 |  | 5 |  |  |
| 6 |  | 6 |  |  |
| 7 |  | 7 |  |  |
| 8 |  | 8 |  |  |
| 9 |  | 9 |  |  |
| 10 |  | 10 |  |  |
| 11 |  | 11 |  |  |
| 12 |  | 12 |  |  |
| 13 |  | 13 |  |  |
| 14 |  | 14 |  |  |

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

1

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | Write a sketch to interface Arduino with LED, LED should blink with a delay of 1 second. |
| 2 | Write a sketch to interface Arduino with SPDT switch & LED. |

Submitted On:

13/10/2022

(Date: DD/MM/YY)

**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);

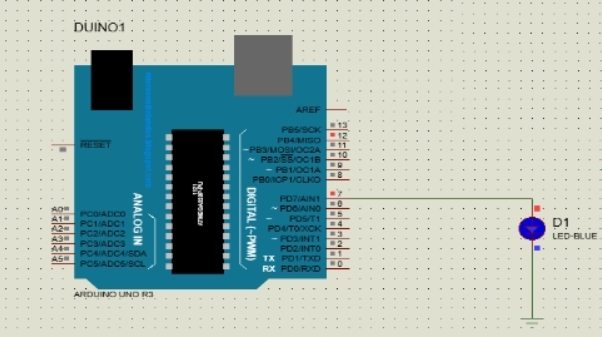
delay(1000);

digitalWrite(LED, LOW);

delay(1000);

}

**Output:**



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

**Solution:**

void setup() {

pinMode(12,INPUT);

pinMode(7,OUTPUT);

}

void loop() {

int sw = digitalRead(12);

if(sw==1)

{

digitalWrite(7,HIGH);

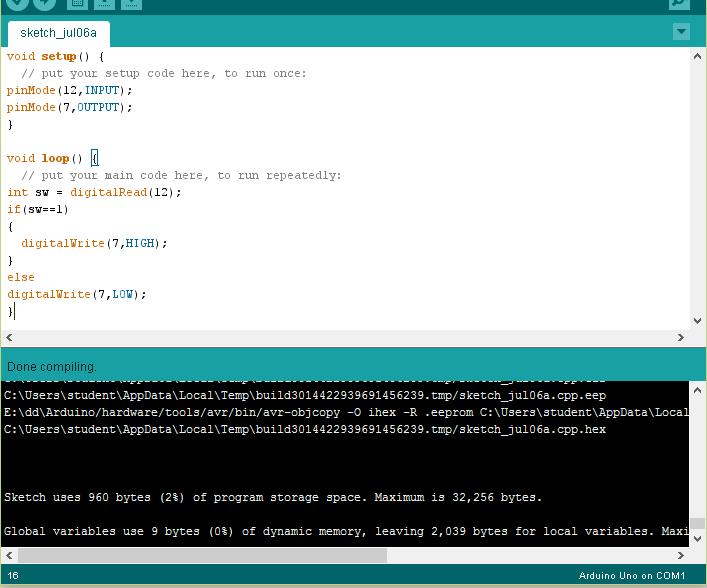
}

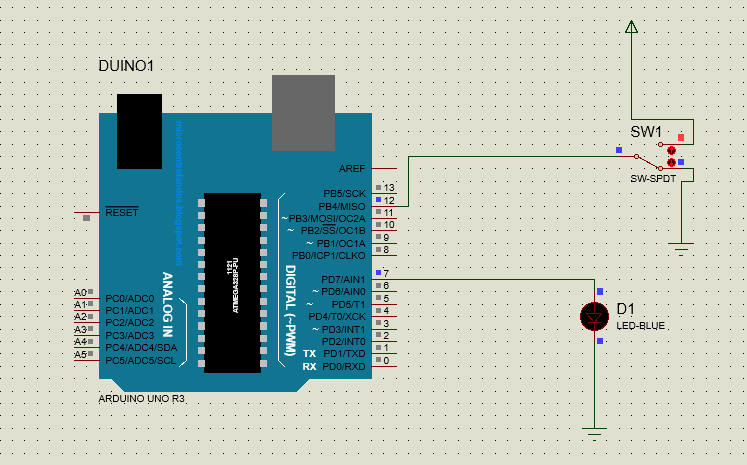
else

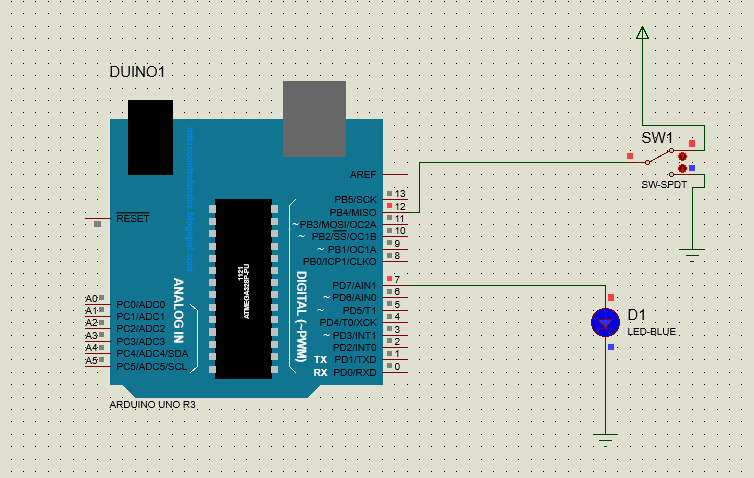
digitalWrite(7,LOW);

}

**Output:**







Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

2

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Write a sketch to interface Arduino with the Relay. The Relay should be controlled by a SPDT Switch. |

Submitted On:

13/10/2022

(Date: DD/MM/YY)

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

**Solution:**

void setup(){

pinMode(12,INPUT);

pinMode(7,OUTPUT);

}

void loop() {

int sw=digitalRead(12);

If(sw==1)

{ digitalWrite(7,HIGH); }

else

digitalWrite(7,LOW);

}

void setup() {

pinMode(12,INPUT);

pinMode(7,OUTPUT);

}

void loop() {

int sw = digitalRead(12);

if(sw==1)

{

digitalWrite(7,HIGH);

}

else

digitalWrite(7,LOW);

}

**Output:**

Diagram, schematic

Description automatically generated

Diagram

Description automatically generated

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

3

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **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 “BEE - 8A/8B” in the 2nd Line of LCD. This Text should blink with a delay of 0.5 seconds. |
| 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. |

Submitted On:

20/10/2022

(Date: DD/MM/YY)

**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 “BEE- 8A/8B” 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("BEE - 8A/8B");

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); // Setting Up the LCD No. of Rows & Columns }

void loop(){

lcd.setCursor(0, 0);

lcd.print("Muhammad Abdullah");

lcd.setCursor(0, 1);

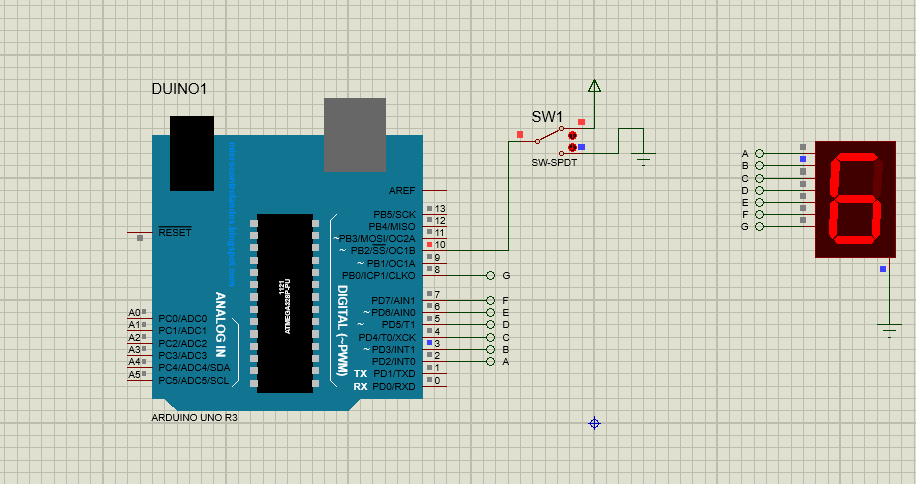
lcd.print("Reg# 70000");

delay(500);

lcd.clear();

delay(5000); }

**Output:**



Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

4

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **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. |
| 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. |

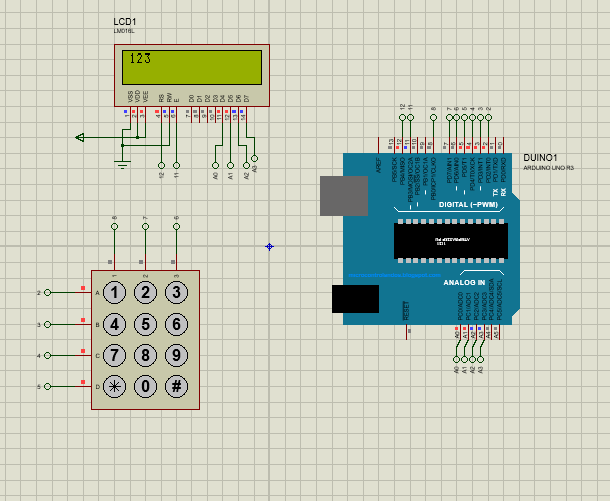
Submitted On:

27/10/2022

(Date: DD/MM/YY)

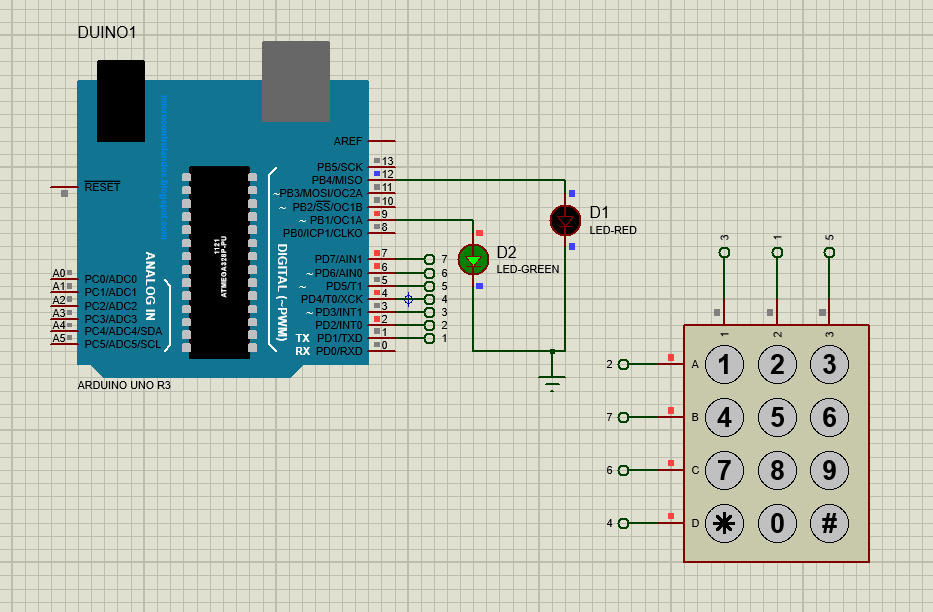
**Task No. 1:** Write a sketch to interface Arduino with a 3 x 4 Matrix Keypad. The display of the pressed key should be displayed on the LCD.

**Output:**



**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.

**Output:**



Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

5

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **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). |
| 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. |

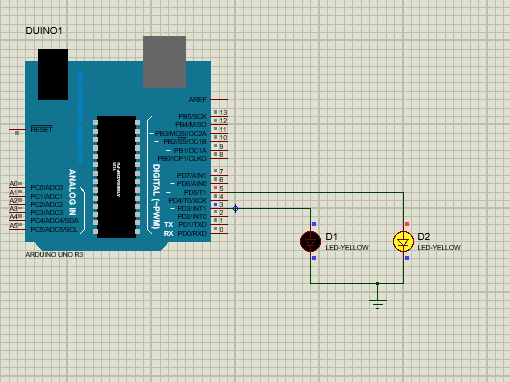
Submitted On:

1/11/2022

(Date: DD/MM/YY)

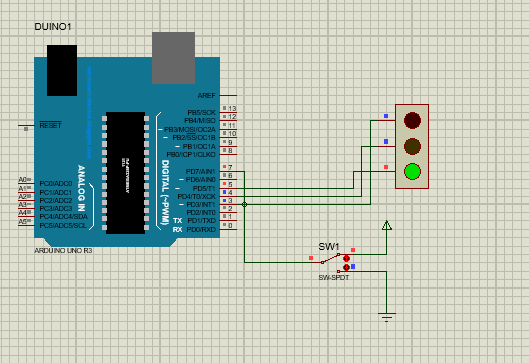
**Task No. 1:** Write a sketch to blink the 2 LEDs interfaced with Arduino at a different rate simultaneously. (i.e. “delay” function limits the designer to perform multitasking from the controller, so this sketch is implement without utilizing this function).

**Output:**



**Task No. 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.

**Output:**



Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

6

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | Write a program to interface potentiometer with analog pin of Arduino Uno to read analog values and display it on LCD |
| 2 | Write a sketch to interface Arduino with the Temperature Sensor (LM35). The value of the Temperature should be displayed on the LCD |

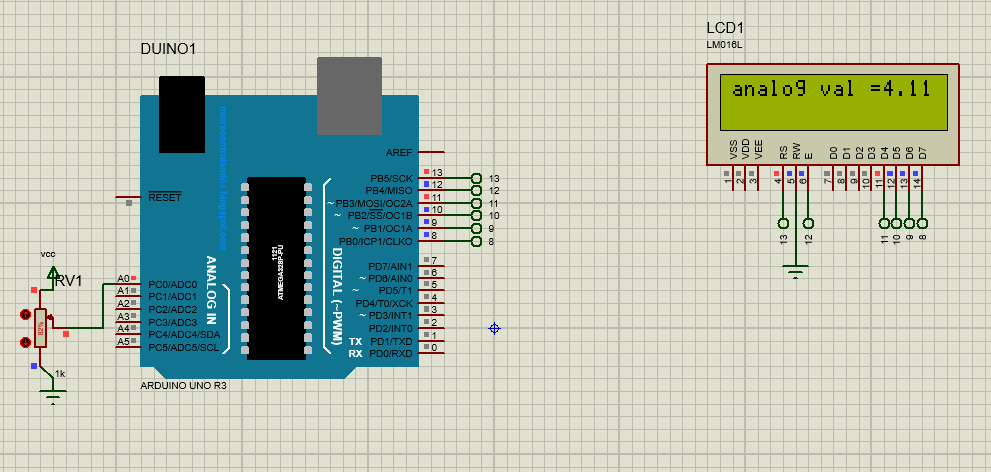
Submitted On:

10/11/2022

(Date: DD/MM/YY)

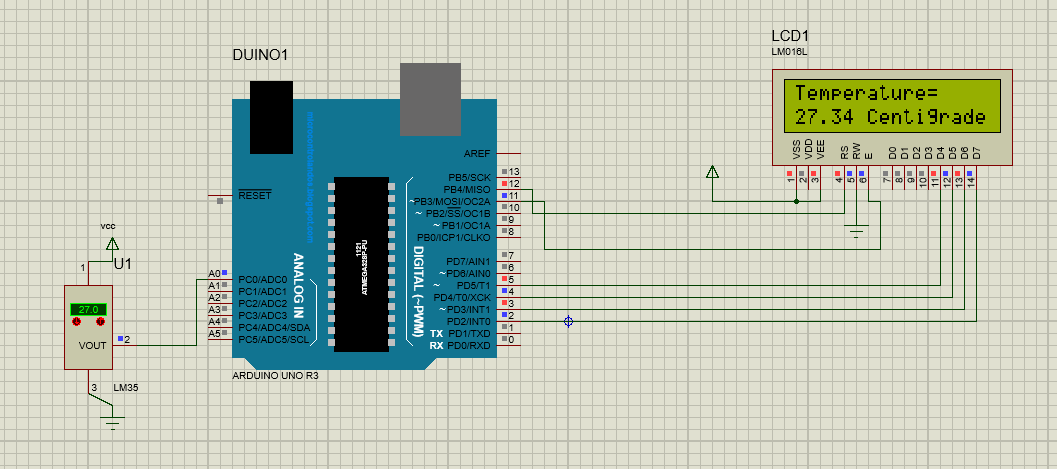
**Task No. 1:** Write a program to interface potentiometer with analog pin of Arduino Uno to read analog values and display it on LCD.

**Output:**



**Task No. 2:** Write a sketch to interface Arduino with the Temperature Sensor (LM35). The value of the Temperature should be displayed on the LCD.

**Output:**



Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

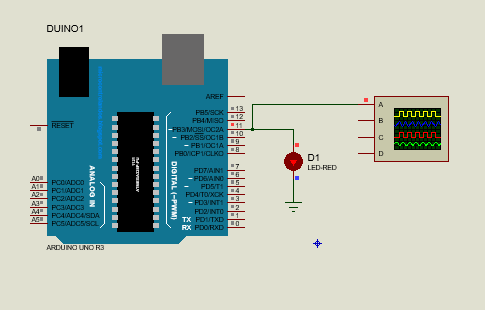
7

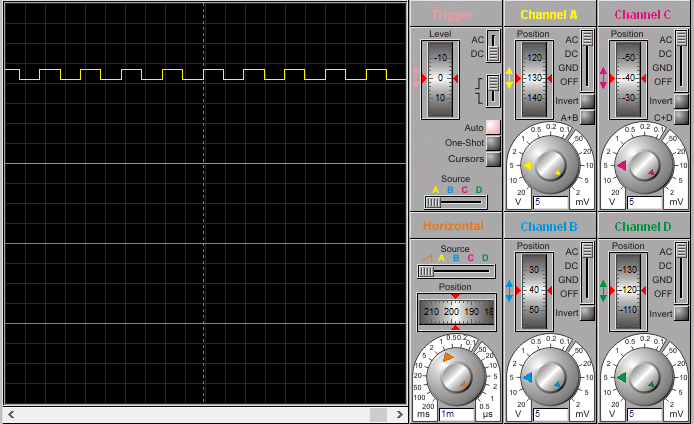
LIST OF TASKS

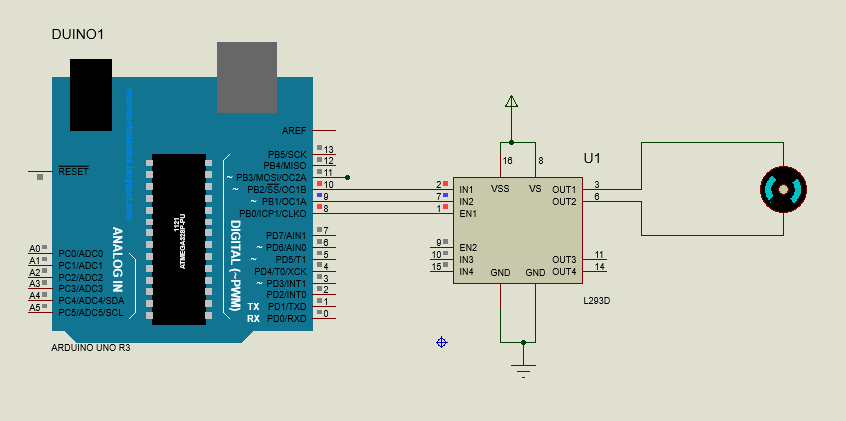
|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** |  |
| 2 |
| 3 |  |
|  |  |
|  |  |

Submitted On:

(Date: DD/MM/YY)







Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

8

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | **In this lab, we will use external interrupt in Arduino UNO.** |
| 2 | **Write a program to use timer interrupt in Arduino Uno.** |
| 3 | **Using the concept of interrupts you have learnt, develop a program that uses interrupt.** |
|  |  |
|  |  |

Submitted On:

10/11/2022

(Date: DD/MM/YY)

**Task # 01: In this lab, we will use external interrupt in Arduino UNO.**

**Solution:**

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);

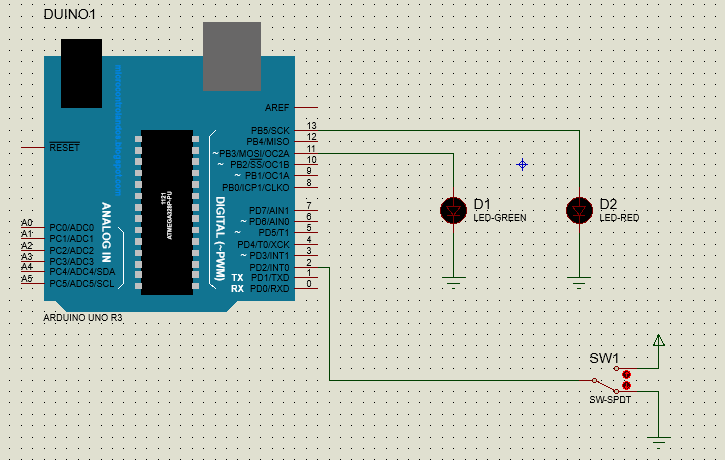
digitalWrite(11,LOW);

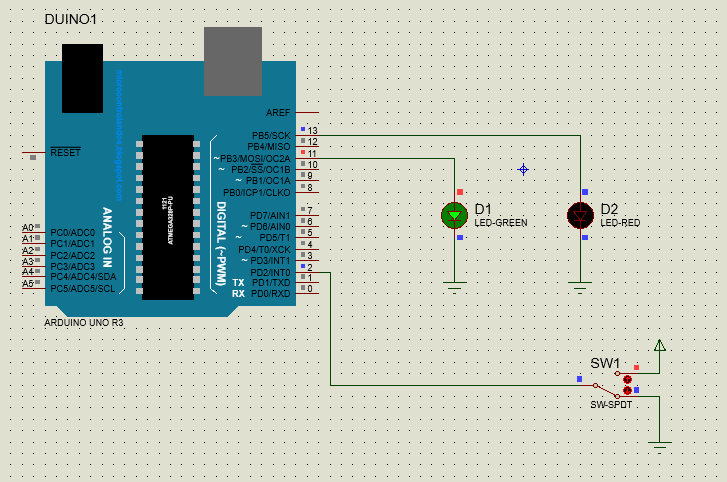
delay(1000);}

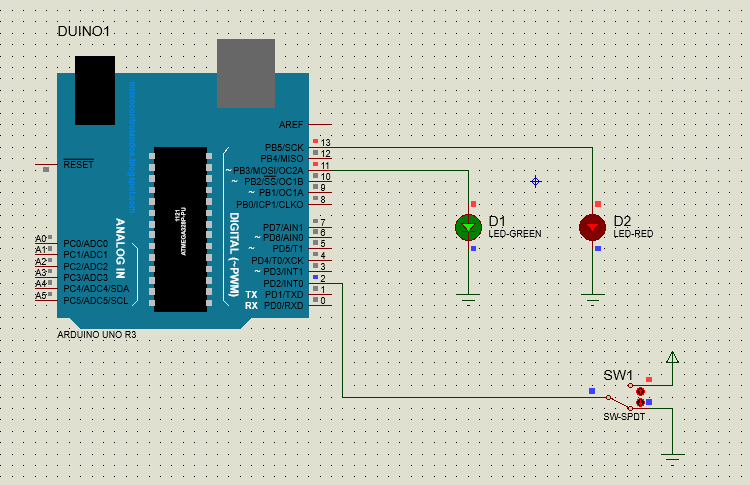
void routine()

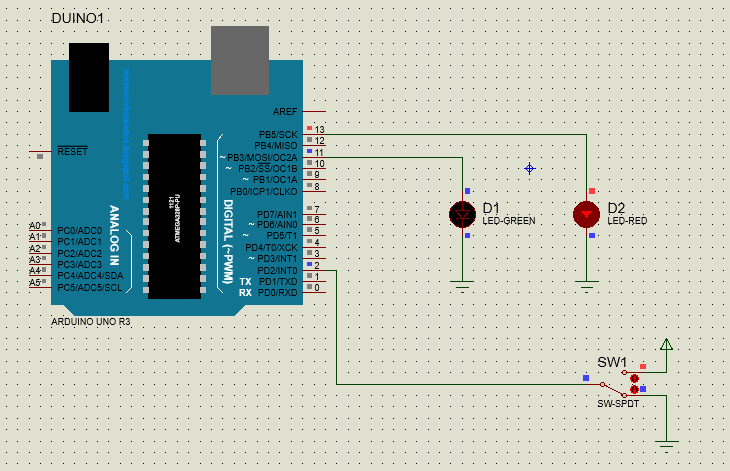
{ digitalWrite(13,HIGH);}

**Output:**









**Task # 02: Write a program to use timer interrupt in Arduino Uno.**

**Solution:**

#include <LiquidCrystal.h>

#include <TimerOne.h>

int led = 2; // led connect to pin 2 of arduino

int led\_i = 3; // led\_i connect to pin 3 of arduino

int x =0;

LiquidCrystal lcd(12, 11, 10, 9, 8, 7); // arduino pins for lcd connections

void setup() {

pinMode(led,OUTPUT);

pinMode(led\_i,OUTPUT);

pinMode(4,INPUT);

lcd.begin(16,2);

Timer1.initialize(4000000); /// 4 sec timer

Timer1.attachInterrupt(interrupt);

}

void interrupt()

{

lcd.setCursor(0, 0);

lcd.print("Timer1 interrupt"); // ISR

if (x==LOW)

{ digitalWrite(3,HIGH);

x = HIGH; }

else

{digitalWrite(3,LOW);

x =LOW;

}}

void loop() {

delay(500);

lcd.clear();

digitalWrite(2 ,HIGH);

delay(1000);

digitalWrite(2,LOW);

delay(1000);

**Output:**

Diagram

Description automatically generated A picture containing diagram

Description automatically generated

Diagram

Description automatically generated with medium confidence Diagram

Description automatically generated

**Task # 03: Using the concept of interrupts you have learnt, develop a program that uses interrupt.**

**Solution:**

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);}

**Output:**

A picture containing chart

Description automatically generated Chart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generated

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

9

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** |  |
| 2 |
| 3 |  |
|  |  |
|  |  |

Submitted On:

(Date: DD/MM/YY)

**Task 1:**

**Solution:**

Master:

#include <SPI.h>

void setup (void)

{

digitalWrite(SS, HIGH); // disable Slave Select

SPI.begin ();

SPI.setClockDivider(SPI\_CLOCK\_DIV2);//divide the clock by 8

}int state;

int oldstate=0;

void loop (void)

{ int c ;

state = !oldstate;

digitalWrite(SS, LOW); // enable Slave Select

// send test string

c = state;

SPI.transfer(c);

digitalWrite(SS, HIGH);

oldstate = state;// disable Slave Select

delay(500);

}

Slave:

#include <SPI.h>

int buff;

volatile boolean process;

void setup (void)

{ pinMode(MISO, OUTPUT); // have to send on master in so it set as output

SPCR |= \_BV(SPE); // turn on SPI in slave mode

process = false;

SPI.attachInterrupt(); // turn on interrupt

pinMode(7,OUTPUT);

}ISR (SPI\_STC\_vect) // SPI interrupt routine {

{ int c = SPDR;

buff = c;

process = true;// read byte from SPI Data Register

}void loop (void) {

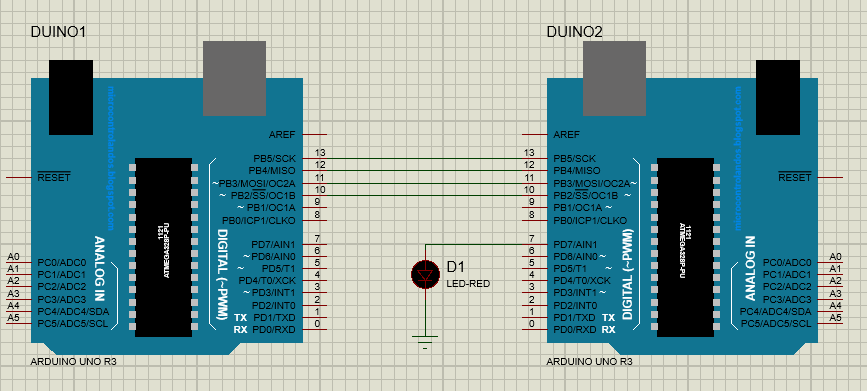
if (process) {

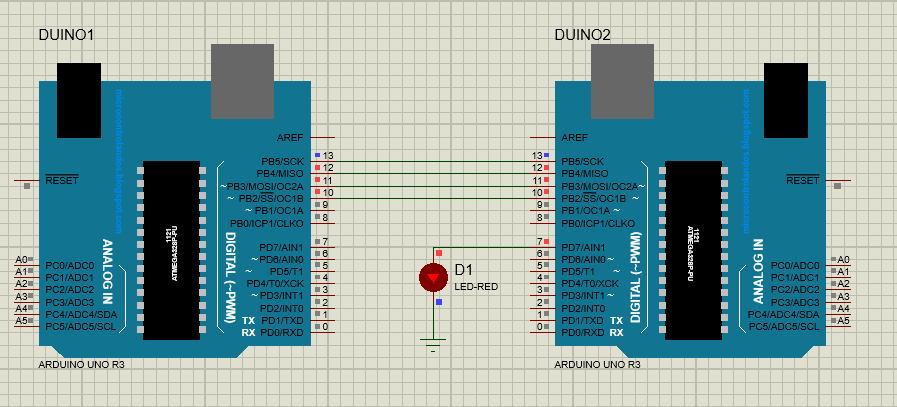
process = false; //reset the process

digitalWrite(7,buff);

}}

**Output:**





**Task 2:**

**Solution:**

Master:

#include <SPI.h>

void setup (void) {

digitalWrite(7,INPUT);

digitalWrite(SS, HIGH); // disable Slave Select

SPI.begin ();

SPI.setClockDivider(SPI\_CLOCK\_DIV2);//divide the clock by 8

}int state;

int oldstate=0;

void loop (void) {

int c =digitalRead(7);

digitalWrite(SS, LOW); // enable Slave Select

// send test string

SPI.transfer(c);

digitalWrite(SS, HIGH);

}

Slave:

#include <SPI.h>

int buff;

volatile boolean process;

void setup (void) {

pinMode(MISO, OUTPUT); // have to send on master in so it set as output

SPCR |= \_BV(SPE); // turn on SPI in slave mode

process = false;

SPI.attachInterrupt(); // turn on interrupt

pinMode(7,OUTPUT);

}ISR (SPI\_STC\_vect) // SPI interrupt routine {

{ int c = SPDR;

buff = c;

process = true;// read byte from SPI Data Register

}void loop (void) {

if (process) {

process = false; //reset the process

digitalWrite(7,buff);

}}

**Output:**

Chart

Description automatically generated

Chart, bar chart

Description automatically generated

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

10

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** |  |
| 2 |
| 3 |  |
|  |  |
|  |  |

Submitted On:

(Date: DD/MM/YY)

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

**Solution:**

**Master:**

#include <Wire.h>

int x = 0;

void setup() {

// Start the I2C Bus as Master

Wire.begin();

pinMode(10,INPUT);

}

void loop() {

x = digitalRead(10);

Wire.beginTransmission(2); // transmit to device #2

Wire.write(x); // sends x

Wire.endTransmission(); // stop transmitting}

**Slave:**

#include <Wire.h>

int LED = 13;

int x = 0;

void setup() {

// Define the LED pin as Output

pinMode (LED, OUTPUT);

// Start the I2C Bus as Slave on address 2

Wire.begin(2);

// Attach a function to trigger when something is received.

Wire.onReceive(receiveEvent);

}void receiveEvent(int bytes) {

x = Wire.read(); // read one character from the I2C

}void loop() {

//If value received is 0 blink LED for 200 ms

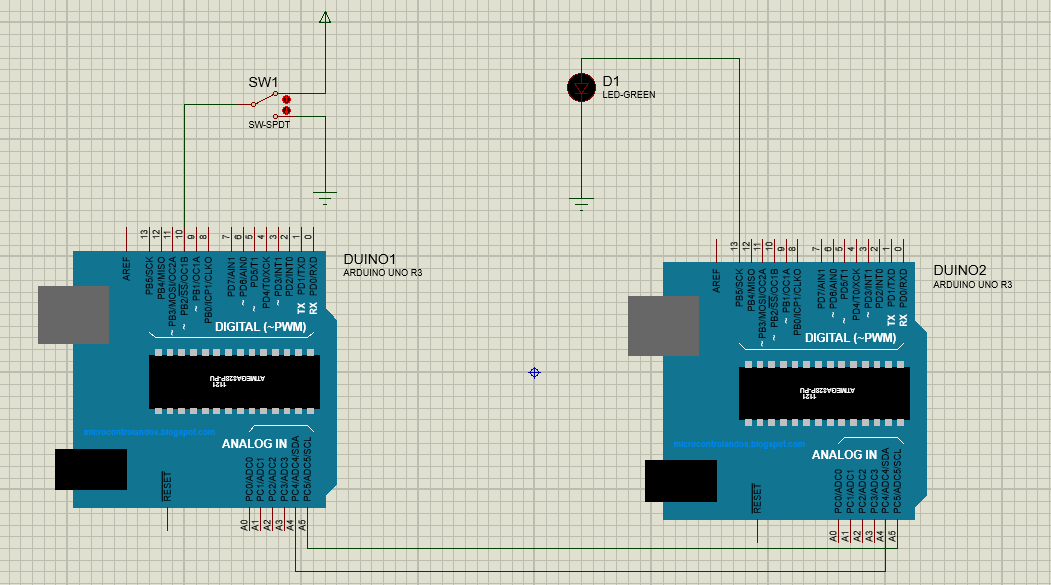
if (x == 1) {

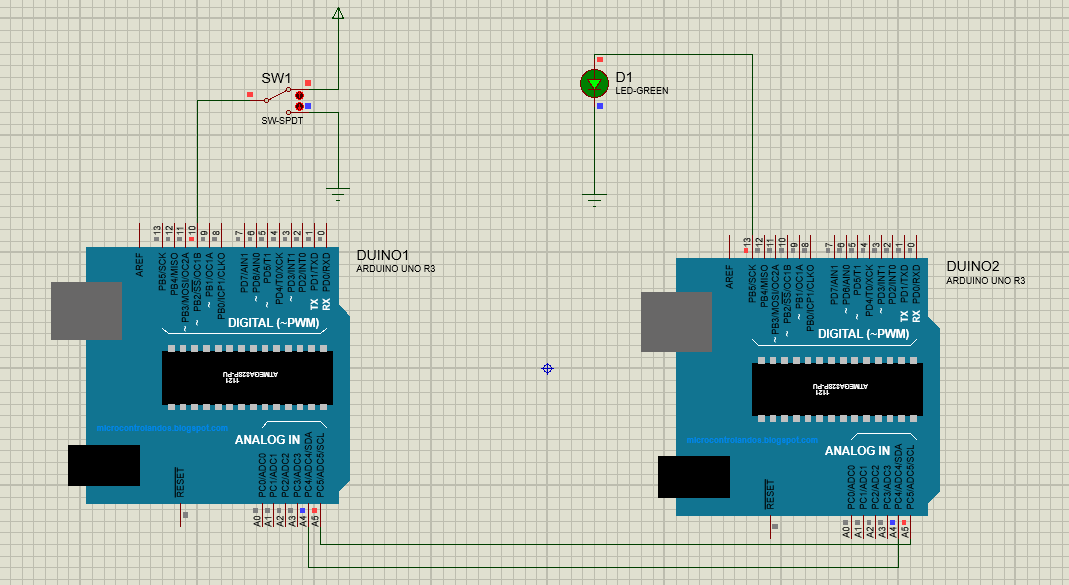
digitalWrite(LED, HIGH); }

else

{ digitalWrite(LED,LOW);}}

**Output:**





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

**Solution:**

**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

**Output:**

Chart, box and whisker chart

Description automatically generated Chart, box and whisker chart

Description automatically generated

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

11

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** |  |
| 2 |
| 3 |  |
|  |  |
|  |  |

Submitted On:

(Date: DD/MM/YY)

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

**Solution:**

void setup() {

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

Serial.begin(9600); // begin serial communication , define baud rate.

}

void loop() {

char string[10] = "Embedded";

int index=0;

for(index=0 ; index <9 ; index++)

{

Serial.print(string[index]);

}

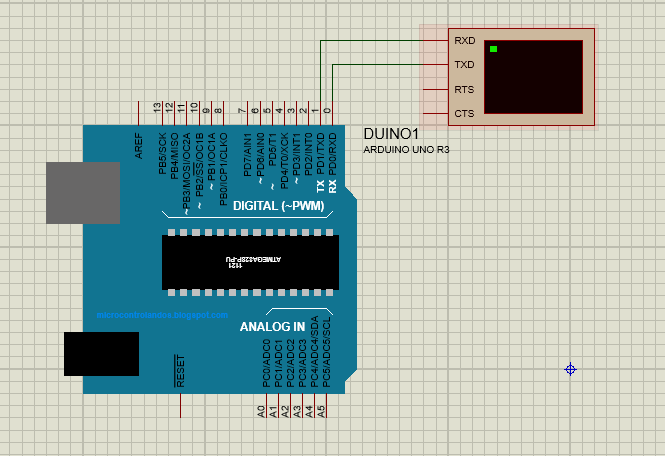
// put your main code here, to run repeatedly:

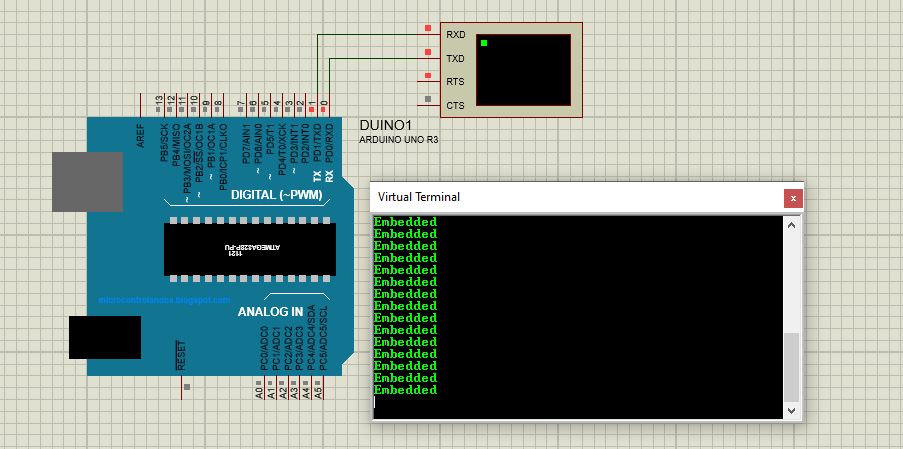
Serial.println(" ");

delay(500);

}

**Output:**





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

**Solution:**

**Arduino 1:**

int SPDT =7;

void setup() {

pinMode(SPDT,INPUT);

Serial.begin(9600);

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

}

void loop() {

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

int button\_status = digitalRead(7);

Serial.write(button\_status);

//delay(500);}

**Arduino 2:**

int LED = 4;

void setup() {

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

pinMode(LED,OUTPUT);

Serial.begin(9600);

}

int x =0;

void loop() {

if(Serial.available() > 0)

{

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

x = Serial.read();

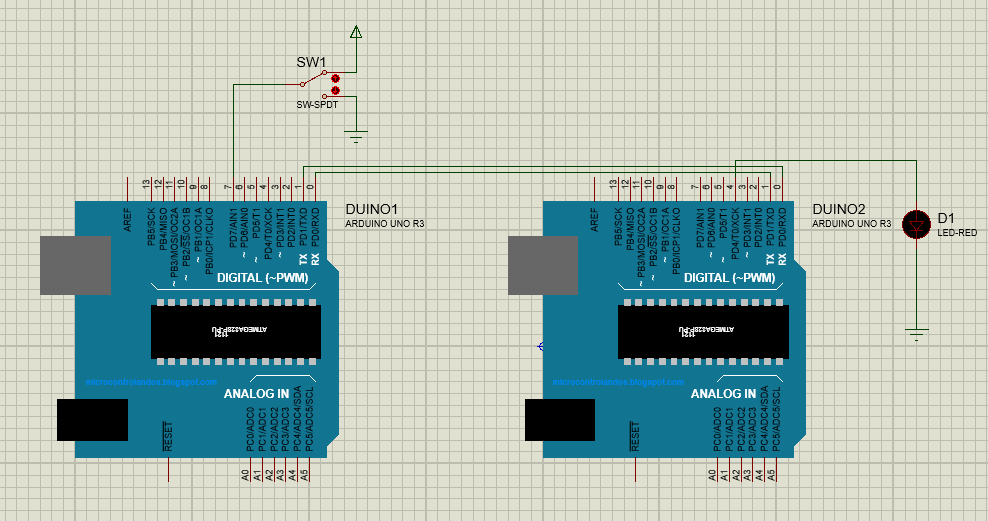
if (x == 1)

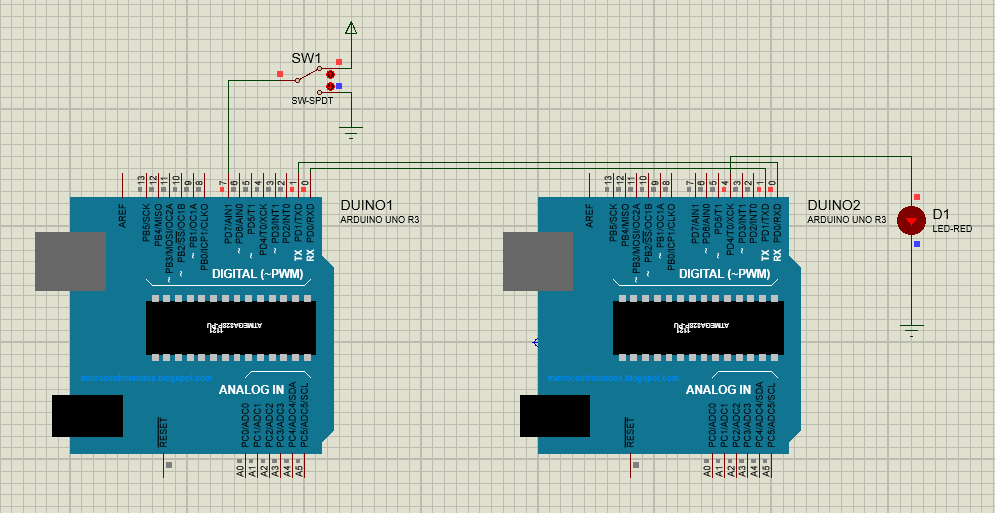
{digitalWrite(LED, HIGH);

}else

{ digitalWrite(LED,LOW);}}}

**Output:**





**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())

{

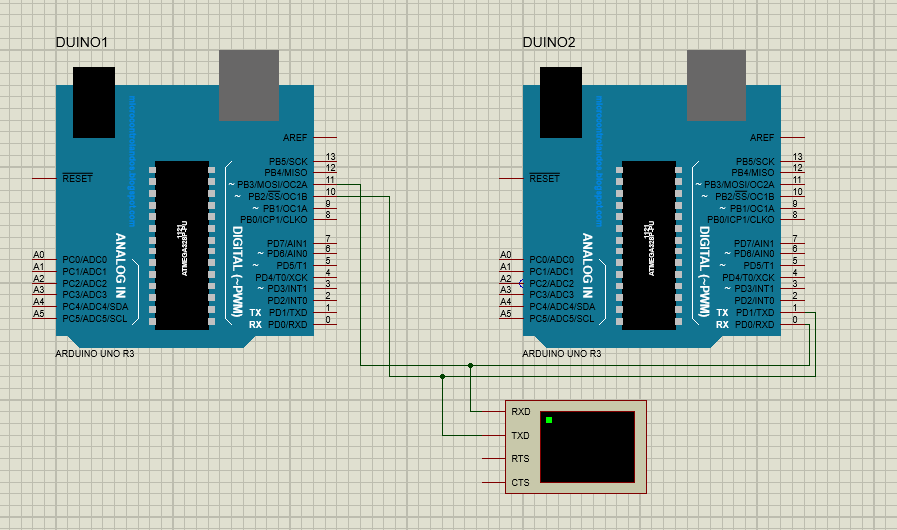
ip=softSerial.read();

softSerial.print(ip);

}

}

**Output:**



Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

12

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | In this task we will write a simple Python script in IDLE to calculate the Percentage by taking the Marks of the courses as input |
| 2 |
| 3 | Write Python Script for Calculator which can perform simple operations of Addition, Subtraction, Multiplication and Division. |
|  |  |
|  |  |

Submitted On:

(Date: DD/MM/YY)

1. In this task we will write a simple Python script in IDLE to calculate the Percentage by taking the Marks of the courses as input.

*print “Welcome to Bahria University (Karachi Campus)”*

* 1. *= input (“Enter Marks Obtained in the Course of E.N.A: ”)*
  2. *= input (“Enter Marks Obtained in the Course of SIGNALS AND SYSTEMS: ”)*
  3. *= input (“Enter Marks Obtained in the Course of COMPUTER ARCHITECTURE: ”) total\_marks = a+b+c*

*print “Total Marks Obtained are: ”, total\_marks percentage = float(total\_marks) \* (100 / 300) print “Percentage is: ”, percentage*

1. In this task we will write a simple Python script to check that the input number is either even or odd.
   1. *= input (“Enter Number: ”)*

*if (a == 0):*

*print “Entered Number is Zero.”*

*elif (a % 2 == 0):*

*print “Entered Number is Even.”*

*else:*

*Print “Entered Number is Odd.”*

1. Write Python Script for Calculator which can perform simple operations of Addition, Subtraction, Multiplication and Division.

**Solution:**

def calculator():

while True:

# Get the user's input

num1 = input("Enter a number (or 'q' to quit): ")

if num1 == 'q':

break

num2 = input("Enter another number: ")

if num2 == 'q':

break

operator = input("Enter an operator (+, -, \*, /): ")

# Convert the numbers to floats

try:

num1 = float(num1)

num2 = float(num2)

except ValueError:

print("Invalid input. Please try again.")

continue

# Perform the calculation

if operator == '+':

result = num1 + num2

elif operator == '-':

result = num1 - num2

elif operator == '\*':

result = num1 \* num2

elif operator == '/':

result = num1 / num2

else:

print("Invalid operator. Please try again.")

continue

# Print the result

print(result)

calculator()

**Output:**

Enter a number (or 'q' to quit): 2

Enter another number: 4

Enter an operator (+, -, \*, /): +

6.0

Enter a number (or 'q' to quit): 4

Enter another number: 2

Enter an operator (+, -, \*, /): -

2.0

Enter a number (or 'q' to quit): 6

Enter another number: 2

Enter an operator (+, -, \*, /): \*

12.0

Enter a number (or 'q' to quit): 8

Enter another number: 2

Enter an operator (+, -, \*, /): /

4.0

Enter a number (or 'q' to quit): q

1. Write ten main Raspberry-Pi application projects. Briefly discuss their details.

1. [Add a Button to a Raspberry Pi](https://www.makeuseof.com/tag/add-button-raspberry-pi-project/):

Adding a button is best done using a breadboard and additional wiring, resistor, and the RPi.GPIO library. If you’re also new to electronics, this project is ideal; it also has an element of programming in Python, which will help any beginner to understand the importance of the Raspberry Pi's GPIO.

2. [Control LED Lights With Raspberry Pi](https://www.makeuseof.com/tag/raspberry-pi-control-led/):

This involves building a simple LED circuit using a breadboard, two LEDs, two resistors, and suitable wires. Once set up, you'll develop knowledge of Python, breadboards, and learn how to safely connect LEDs and other components.

3. [Pi-Powered Motion Sensor and Alarm](http://pimylifeup.com/raspberry-pi-motion-sensor/):

Creating a simple motion sensor and piezo buzzer alarm will teach you the basics of working with external hardware with the Pi, and this project uses Python, so you'll start learning a bit of that, too.

4. [Add a Power Button to Your Raspberry Pi](https://www.makeuseof.com/tag/add-power-button-raspberry-pi/)

One of the cost-saving measures of the Raspberry Pi is that it doesn't include a power switch. Instead, you safely shut down using the operating system and power off when this has completed. Alternatively, you could add your own. Adding a power button is one of the most basic Raspberry Pi projects for beginners, suitable for Raspberry Pi Zero and Raspberry Pi 4 as well as standard models.

1. [Make Your Printer Wireless](https://www.makeuseof.com/tag/make-wireless-printer-raspberry-pi/)

With one of the easiest Raspberry Pi projects, you can make an old printer wireless with a few simple commands. If your Pi model doesn't have built-in Wi-Fi, you can buy [a $10 Wi-Fi dongle](https://www.amazon.com/Edimax-EW-7811Un-150Mbps-Raspberry-Supports/dp/B003MTTJOY?linkCode=ll1&tag=makeusw-20&linkId=cd7029f0324834cd3f07468a4d68520a&language=en_US&ref_=as_li_ss_tl&ascsubtag=UUmuoUeUpU37000). Simply insert the Wi-Fi dongle, plug in your printer, enter some commands, and remote printing will be enabled in minutes.

6. [Make Music with Sonic Pi](http://sonic-pi.net/)

Sonic Pi turns your Raspberry Pi into a music-coding machine, letting you use basic programming skills to make music. This can be anything from sound effects and short samples to full-length songs. Sonic Pi comes preinstalled with Raspberry Pi OS, making it a perfect Raspberry Pi beginner project for anyone with some musical ability.

7. [Build a Network Game Server](https://www.makeuseof.com/tag/raspberry-pi-game-servers/)

Particularly suited to more recent Raspberry Pi models, game server projects are remarkably straightforward. All you need to do is make sure the Pi is connected to your network, then install the server software for your preferred game from the command line.Various games have servers that will run on the Pi, including Minecraft and Quake. If you have any compatible games running on another device, you can set up network play sessions on your Pi.

8. [Connect Dual 4K Monitors to a Raspberry Pi 4 or 400](https://www.makeuseof.com/4k-dual-monitor-setup-with-raspberry-pi/)

If you have two suitable HDMI monitors, you’ll also need a pair of micro-HDMI to HDMI cables. Support for dual displays is built into Raspberry Pi OS, which makes set up particularly straightforward. If you’re comfortable connecting a computer to a monitor, this is the perfect Raspberry Pi beginner project.

9. [Build a Cryptocurrency Price Ticker](https://www.makeuseof.com/build-cryptocurrency-price-ticker-using-raspberry-pi/)

You’ll need a Raspberry Pi 3 or later with an attached display, preferably something pretty compact like a HyperPixel, rather than a larger wall-hanging TV. Once everything is set up, you’ll have a Raspberry Pi that displays the latest cryptocurrency prices, sourced from markets including CoinGecko, Binance, Coinbase Pro, Kraken, and many others.

10. [Set Up an Android Tablet as a Raspberry Pi Display](https://www.makeuseof.com/tag/android-tablet-screen-raspberry-pi/)

With this project, you can wirelessly connect an Android tablet to the Pi for use as a display. No cables are required, no wiring is necessary, and there is little software to install. But by the end you’ll have a handy portable display that lets you use your Raspberry Pi anywhere you can plug it in.

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

13

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | In this task we will write a simple Python script in LXTerminal to glow LED using GPIO 7 of Raspberry-Pi |
| 2 |
| 3 |  |
|  |  |
|  |  |

Submitted On:

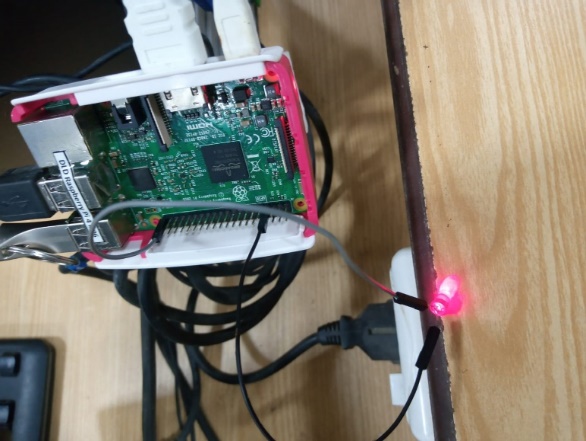
(Date: DD/MM/YY)

1. In this task we will write a simple Python script in LXTerminal to glow LED using GPIO 7 of Raspberry-Pi.

*import RPi.GPIO as GPIO GPIO.setmode (GPIO.BOARD)*

*GPIO.setup (7, GPIO.OUT)*

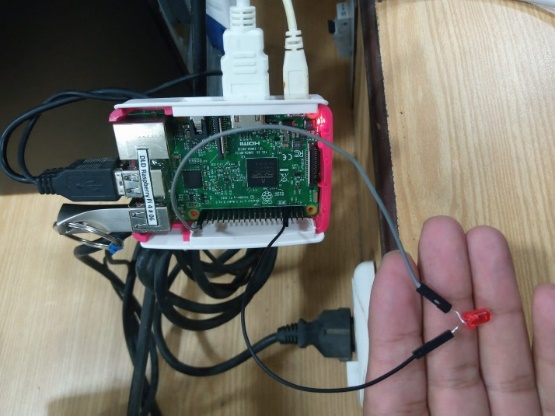
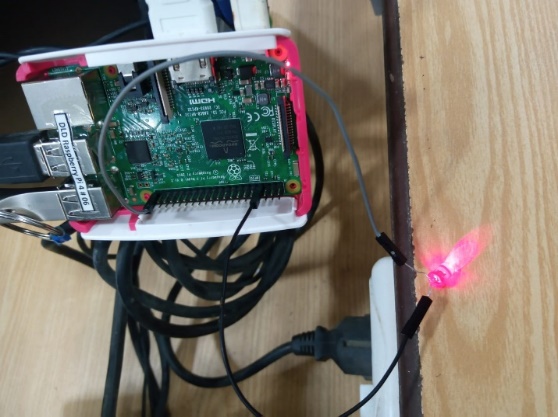
*GPIO.output (7, True)*



1. In this task we will write a simple Python script in LXTerminal to blink single LED using a GPIO 7 of Raspberry-Pi.

*import RPi.GPIO as GPIO import time*

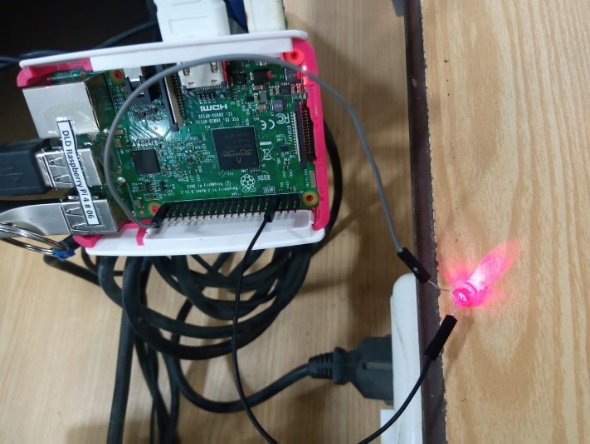
*GPIO.setmode (GPIO.BOARD) GPIO.setup (7, GPIO.OUT) While True:*

*GPIO.output (7, True) time.sleep (0.5) GPIO.output (7, False) time.sleep (0.5)*

1. In this task we will write a simple Python script to control LED using GPIOs as PWM application.

*import RPi.GPIO as GPIO GPIO.setmode(GPIO.BOARD) GPIO.setup(7, GPIO.OUT) pwm\_led = GPIO.PWM(7, 500)*

*pwm\_led.start(100) while True:*

*duty\_s = input("Enter Brightness (0 to 100):")*



1. In this task we will write a simple Python script for Traffic control signals using various GPIOs. 

*import RPi.GPIO as GPIO*

*import time GPIO.cleanup()*

*GPIO.setmode(GPIO.BOARD)*

*GPIO.setup(3,GPIO.OUT)*

*GPIO.setup(5,GPIO.OUT)*

*GPIO.setup(7,GPIO.OUT) GPIO.setwarnings(False) while True:*

*GPIO.output(7,GPIO.HIGH)* *time.sleep(5)*

*GPIO.output(5,GPIO.HIGH) time.sleep(2)*

*GPIO.output(7,GPIO.LOW)*

*GPIO.output(5,GPIO.LOW) GPIO.output(3,GPIO.HIGH) time.sleep(10)*

*GPIO.output(3,GPIO.LOW)*

*GPIO.output(5,GPIO.HIGH)*

4. Write a simple Python script to drive dc motors in both directions.

**Solution:**

import RPi.GPIO as GPIO

import time

# Set the GPIO mode to BCM numbering

GPIO.setmode(GPIO.BCM)

# Set the pin numbers for the motor control pins

motor\_pin\_1 = 18

motor\_pin\_2 = 17

# Set the motor control pins as output pins

GPIO.setup(motor\_pin\_1, GPIO.OUT)

GPIO.setup(motor\_pin\_2, GPIO.OUT)

# Function to drive the motor in a given direction

def drive\_motor(direction):

if direction == "forward":

# Set the motor pins to drive the motor forward

GPIO.output(motor\_pin\_1, GPIO.HIGH)

GPIO.output(motor\_pin\_2, GPIO.LOW)

elif direction == "reverse":

# Set the motor pins to drive the motor in reverse

GPIO.output(motor\_pin\_1, GPIO.LOW)

GPIO.output(motor\_pin\_2, GPIO.HIGH)

# Drive the motor forward for 2 seconds

drive\_motor("forward")

time.sleep(2)

# Drive the motor in reverse for 2 seconds

drive\_motor("reverse")

time.sleep(2)

# Turn off the motor

GPIO.output(motor\_pin\_1, GPIO.LOW)

GPIO.output(motor\_pin\_2, GPIO.LOW)

GPIO.cleanup()

**Output**:

