| **EX .NO :01** | **Arduino Program with LED Blink** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the LED Blink.

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the LED.

**Step4:** Enter the code to blink the LED.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

void setup() {

pinMode(13, OUTPUT);

}

void loop() {

digitalWrite(13,HIGH);

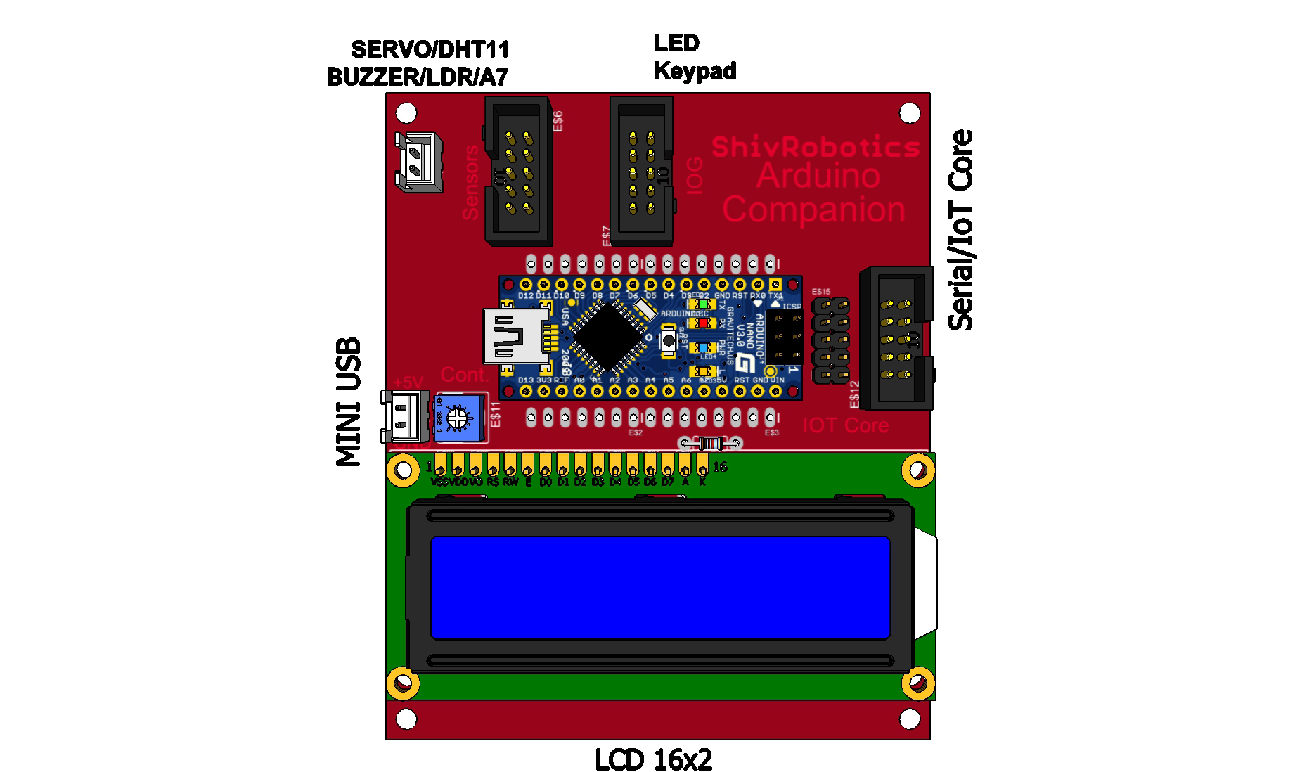
delay (5000);

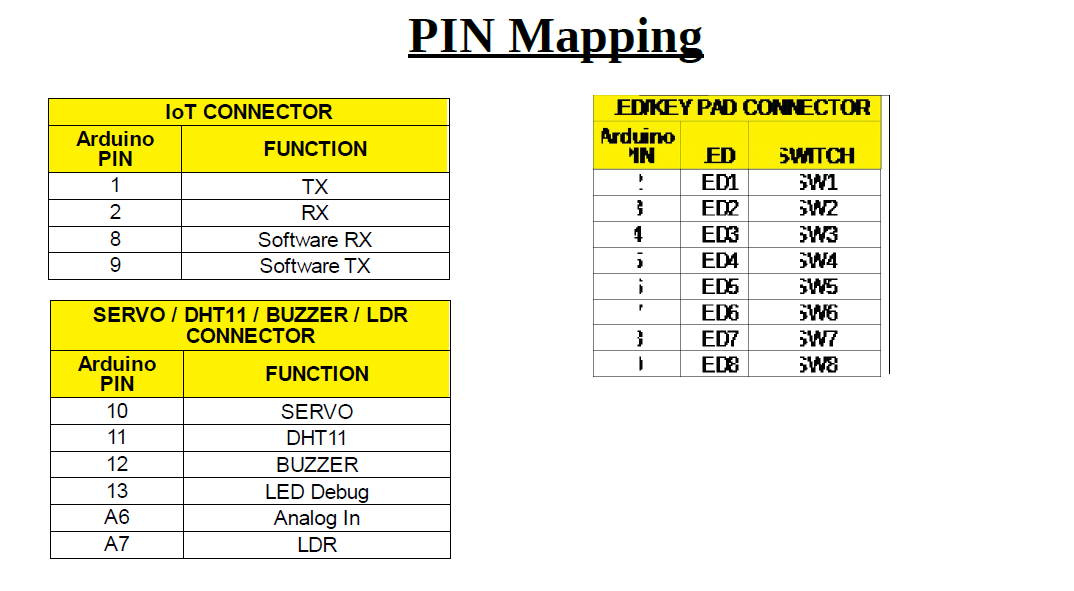
digitalWrite(13,LOW);

delay(5000);

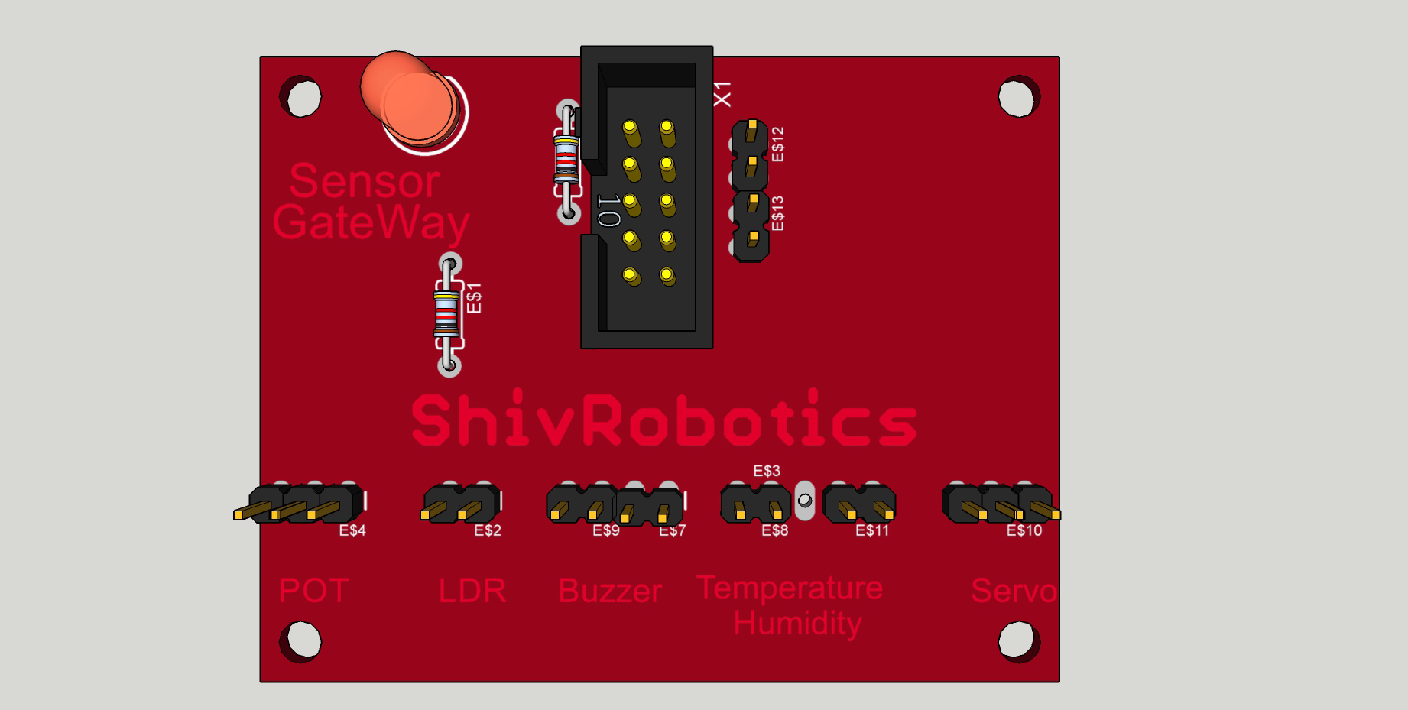
}

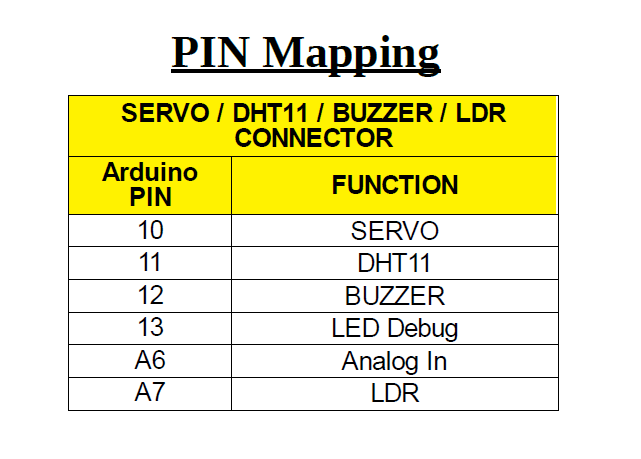
**Arduino Companion**





**Sensor Gateway**





**Output:**

| **Input** | **Output** |
| --- | --- |
| digitalWrite HIGH | LED ON |
| digitalWrite LOW | LED OFF |

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program to blink an LED is written and output was verified successfully.

| **EX .NO :02** | **Arduino Program to control intensity light** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the control intensity light.

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the LED.

**Step4:** Enter the code to blink the LED.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

const int LED\_PIN = 5;

const int WAITTIME = 50;

const int STEP = 5;

void setup(){

pinMode( LED\_PIN, OUTPUT );

}

void loop(){

int i;

i = 0;

while ( i<= 255 ){

analogWrite( LED\_PIN, i );

delay( WAITTIME );

i = i + STEP;

}

i = 255;

while ( i>= 0 ){

analogWrite( LED\_PIN, i );

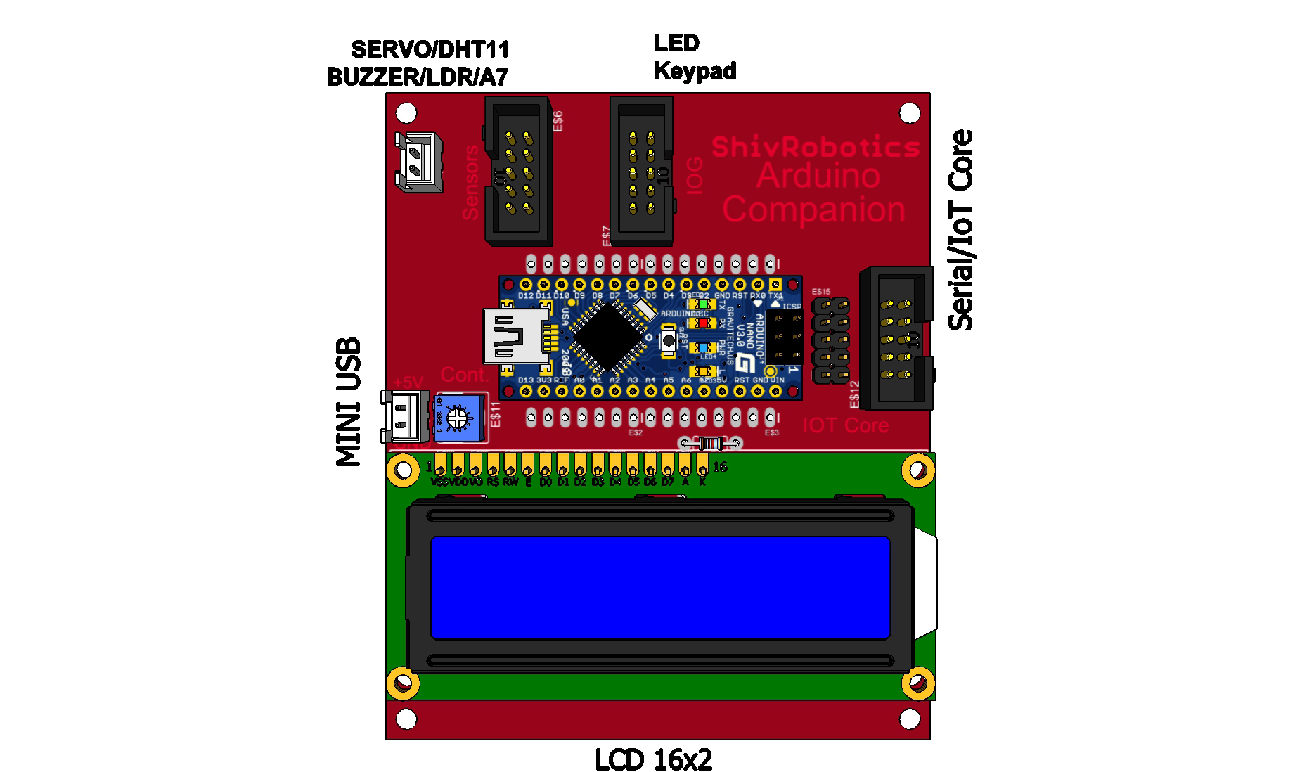
delay( WAITTIME );

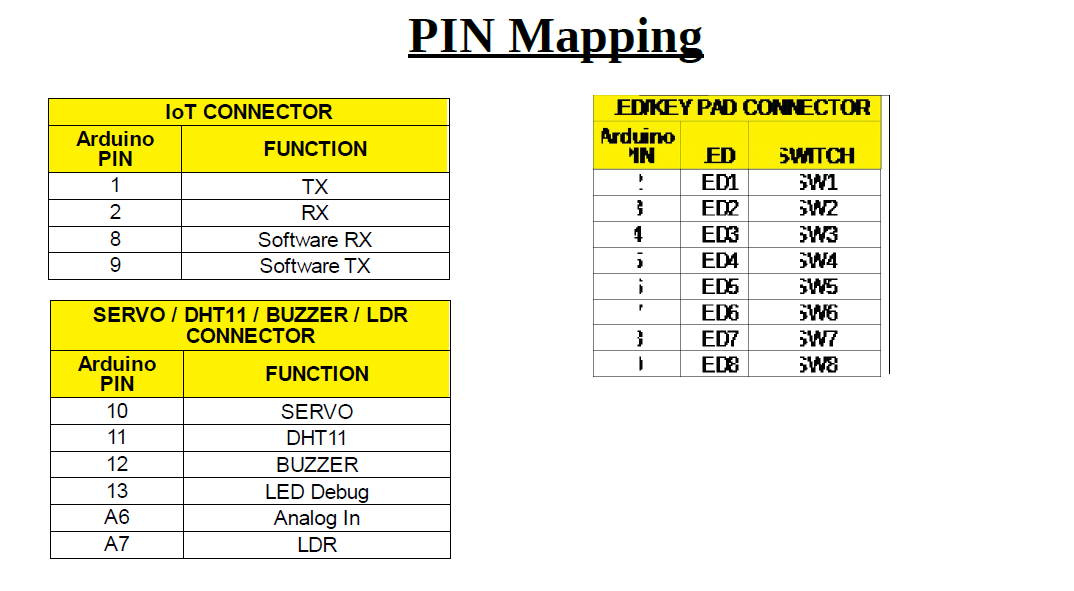
i = i - STEP;

}

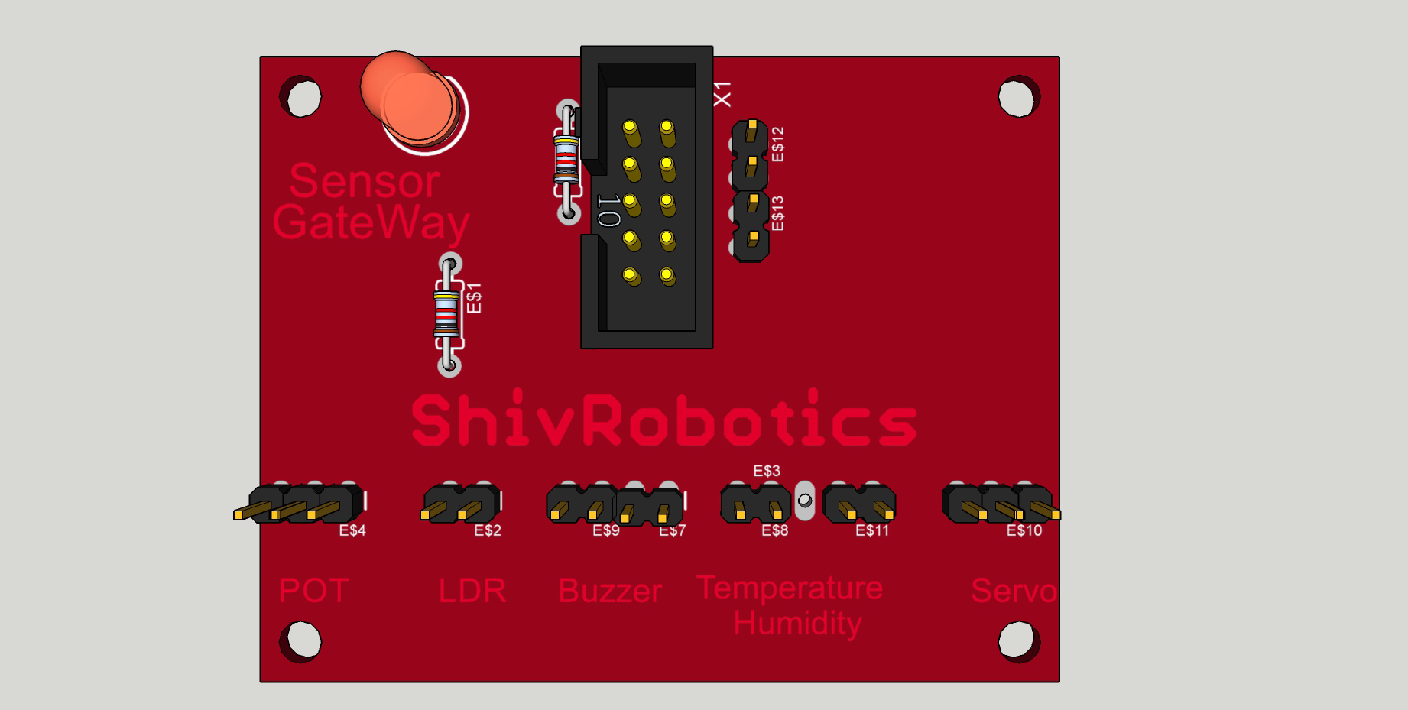
}

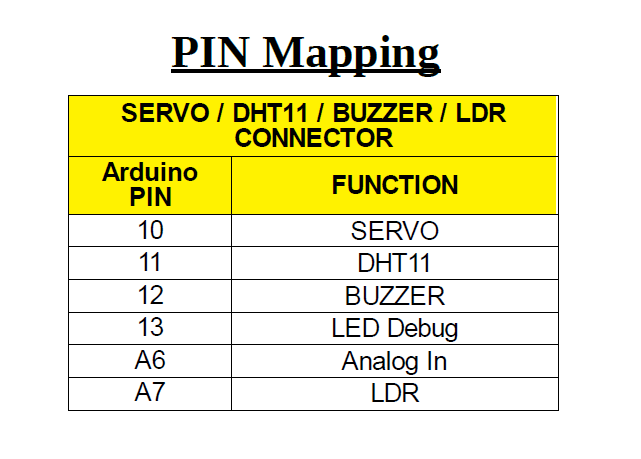
**Arduino Companion**





**Sensor Gateway**





**Output:**

LED light blinks

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program to control intensity light was written and executed successfully.

| **EX .NO : 3** | **Arduino Program with LCD control** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the LCD control**.**

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the LCD.

**Step4:** Enter the code to activate the LCD.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

/\*

The circuit:

\* LCD RS pin to digital pin D2

\* LCD Enable pin to digital pin D3

\* LCD D4 pin to digital pin D4

\* LCD D5 pin to digital pin D5

\* LCD D6 pin to digital pin D6

\* LCD D7 pin to digital pin D7

\* LCD R/W pin to ground

\* LCD VSS pin to ground

\* LCD VCC pin to 5V

\* TrimPot to LCD VO pin (pin 3)

\*/

#include <LiquidCrystal.h>

LiquidCrystal lcd (A0,A1,A2,A3,A4,A5);

int i;

void setup() {

lcd.begin(16, 2);

lcd.print("Hello World");

}

void loop() {

if(i>=10)

{

i=0;

lcd.clear();

lcd.print("Hello World");

}

lcd.setCursor(0,1);

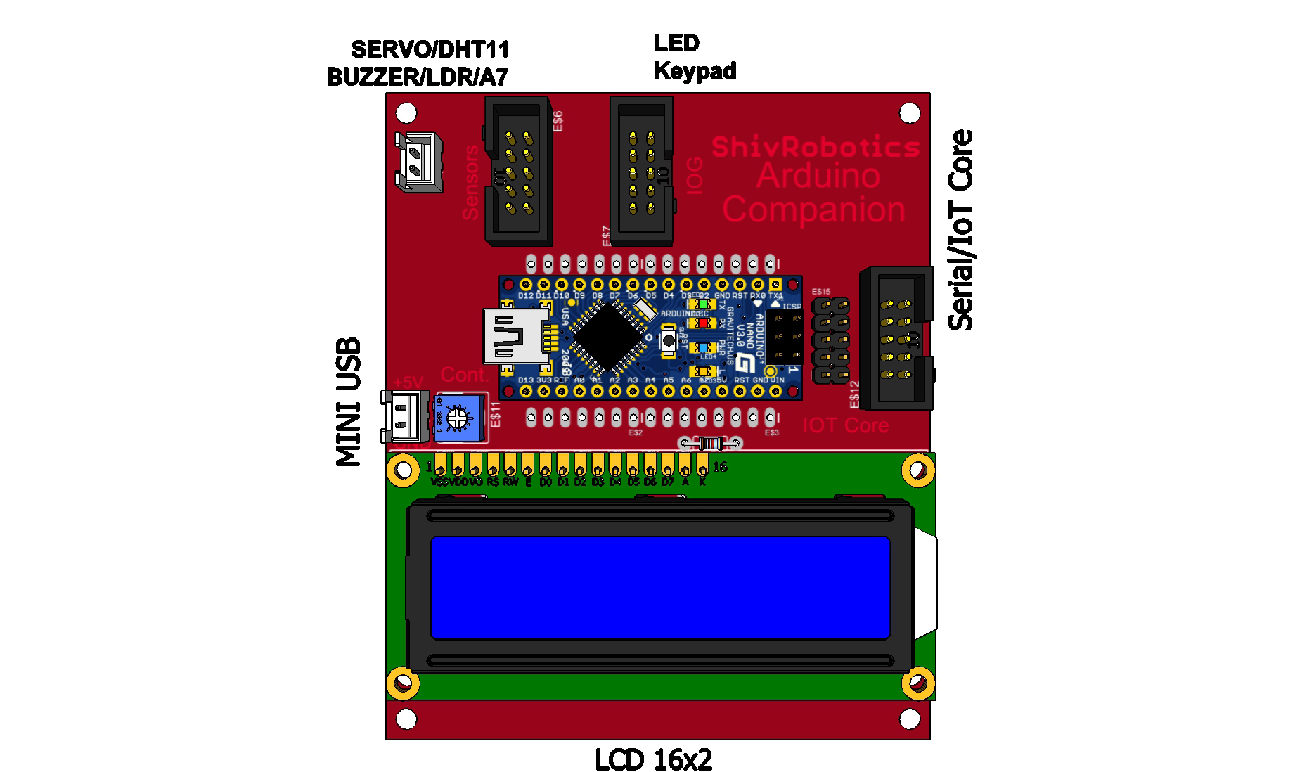
lcd.print(i);

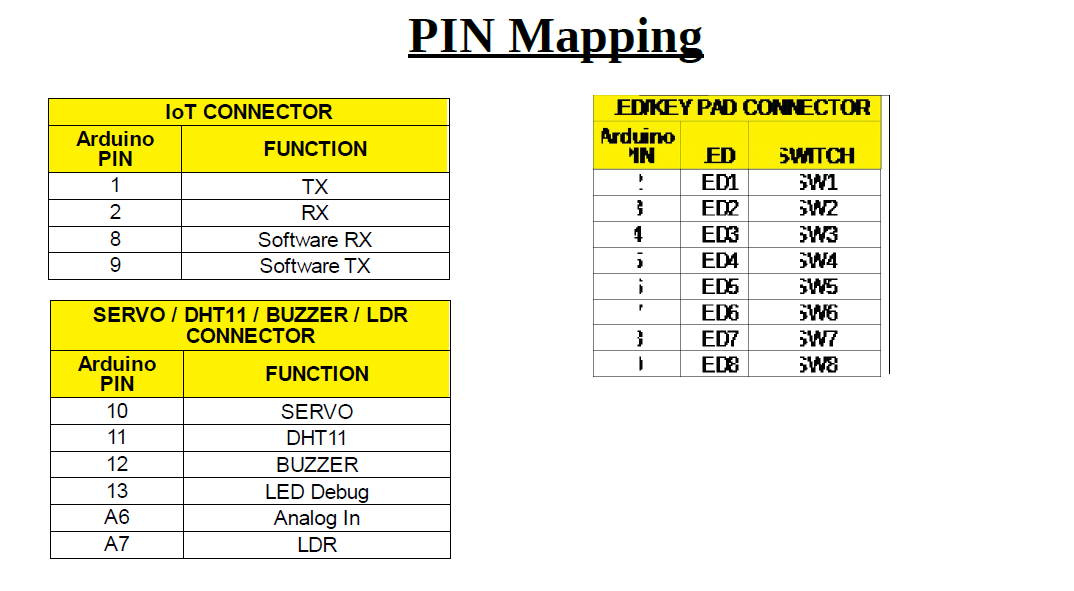
delay(1000);

i++;

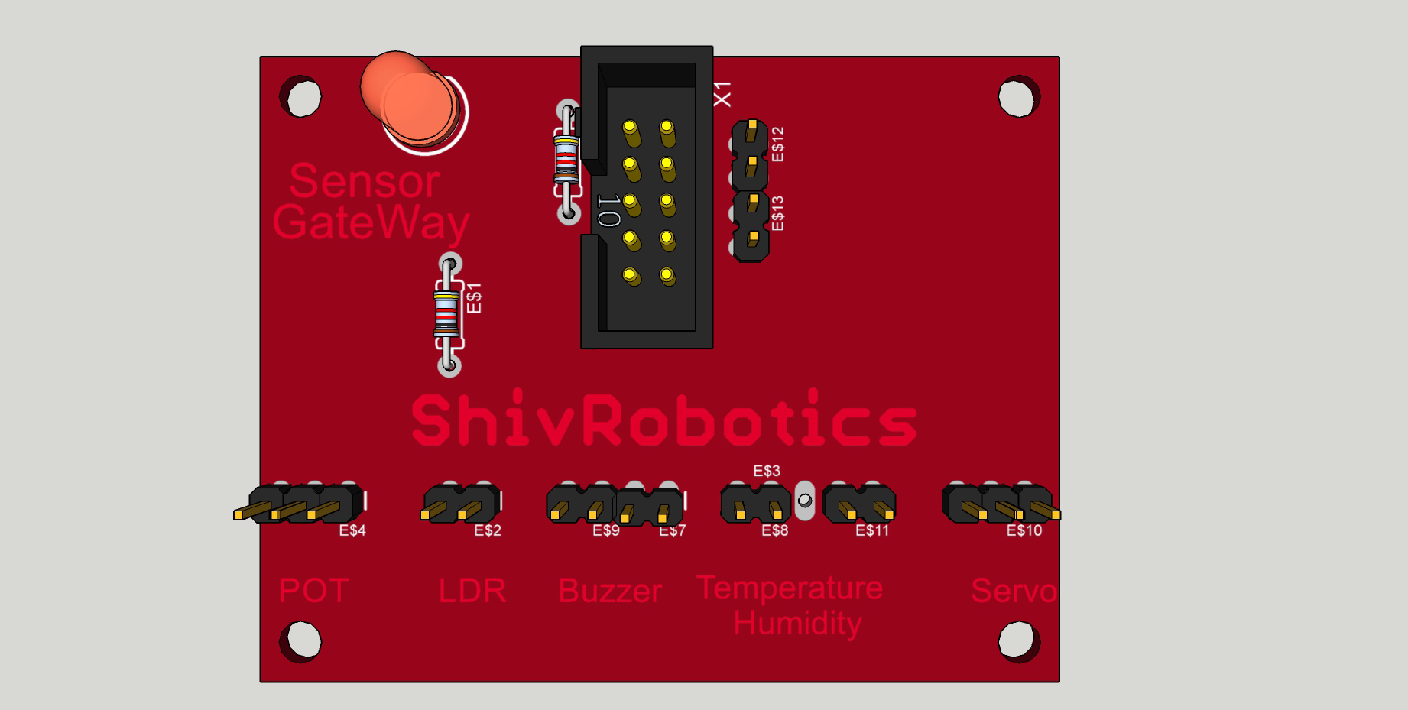
}

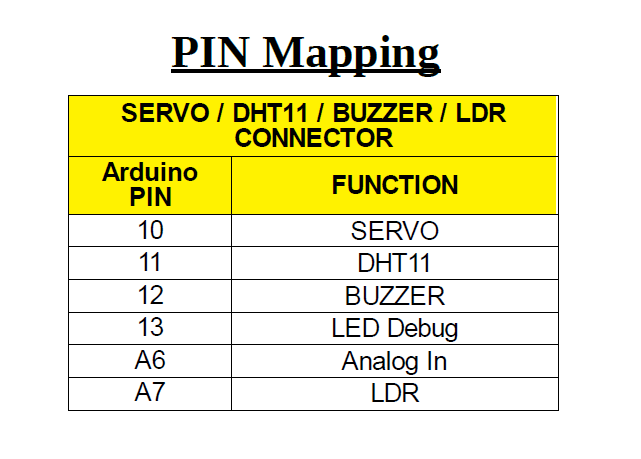
**Arduino Companion**





**Sensor Gateway**





**Output:**

LCD display:

Hello World

1

2

…..

10

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program with LCD control is written and output was verified successfully.

| **EX .NO : 04** | **Arduino Program with Buzzer Control** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the Buzzer Control.

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the Buzzer.

**Step4:** Enter the code to activate the Buzzer.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

void setup() {

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

pinMode(12, OUTPUT);

}

void loop() {

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

digitalWrite(12,HIGH);

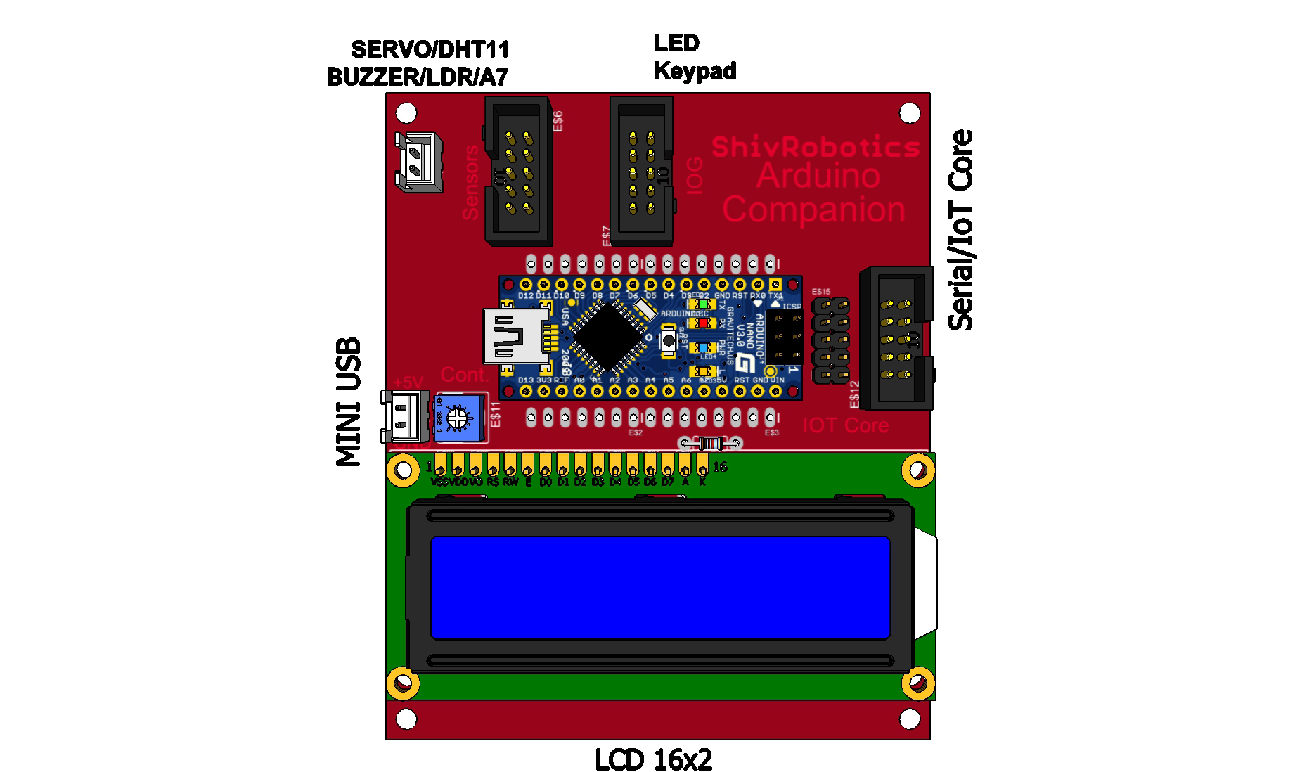
delay (5000);

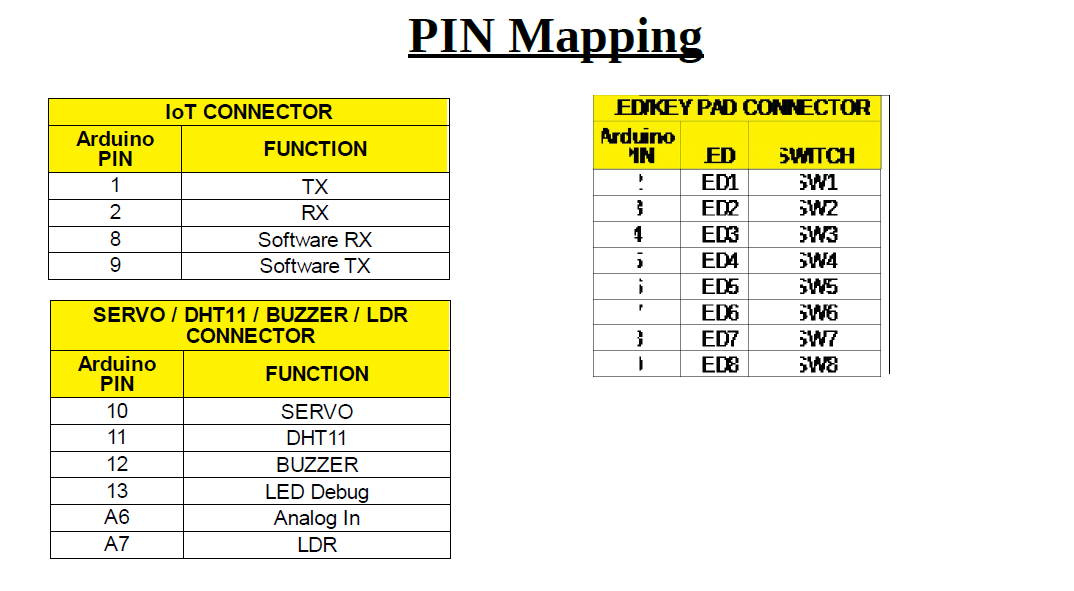
digitalWrite(12,LOW);

delay (5000);

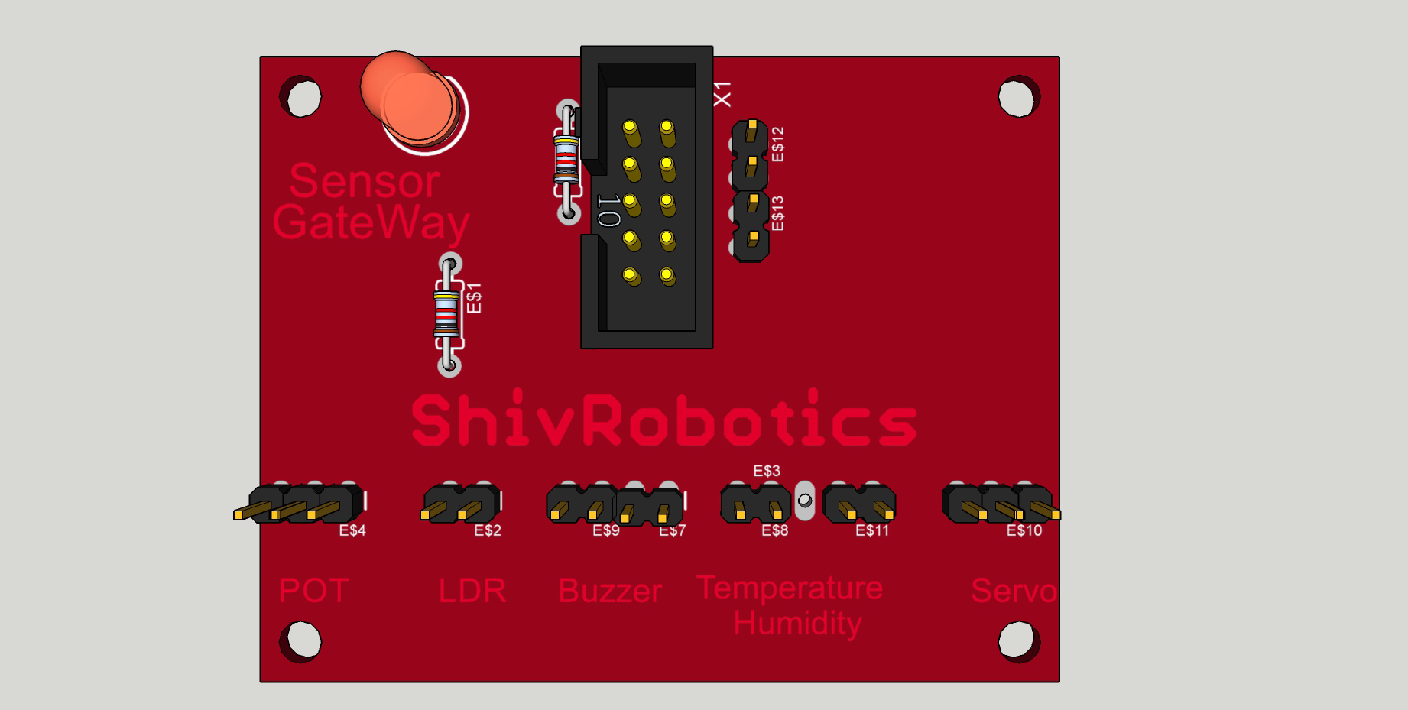
}

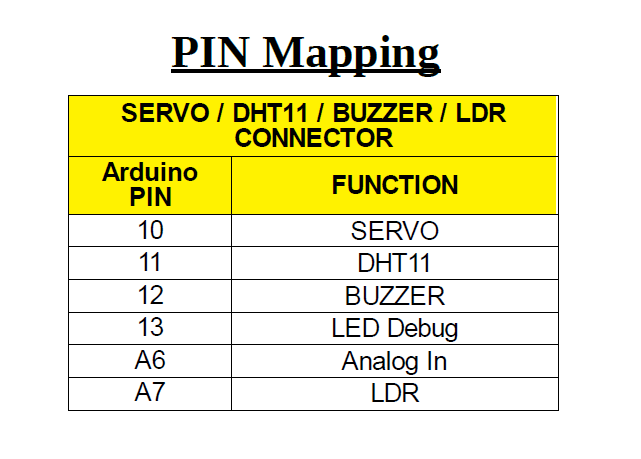
**Arduino Companion**





**Sensor Gateway**





**Output:**

| **Input** | **Output** |
| --- | --- |
| digitalWrite HIGH | BUZZER 1 |
| digitalWrite LOW | BUZZER 0 |

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program with Buzzer Control is written and output was verified successfully.

| **EX .NO : 05** | **Arduino Program with LDR control** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the LDR control**.**

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the LDR.

**Step4:** Enter the code to activate the LDR.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

int lightPin = A6;

void setup()

{

Serial.begin(9600);

}

void loop()

{

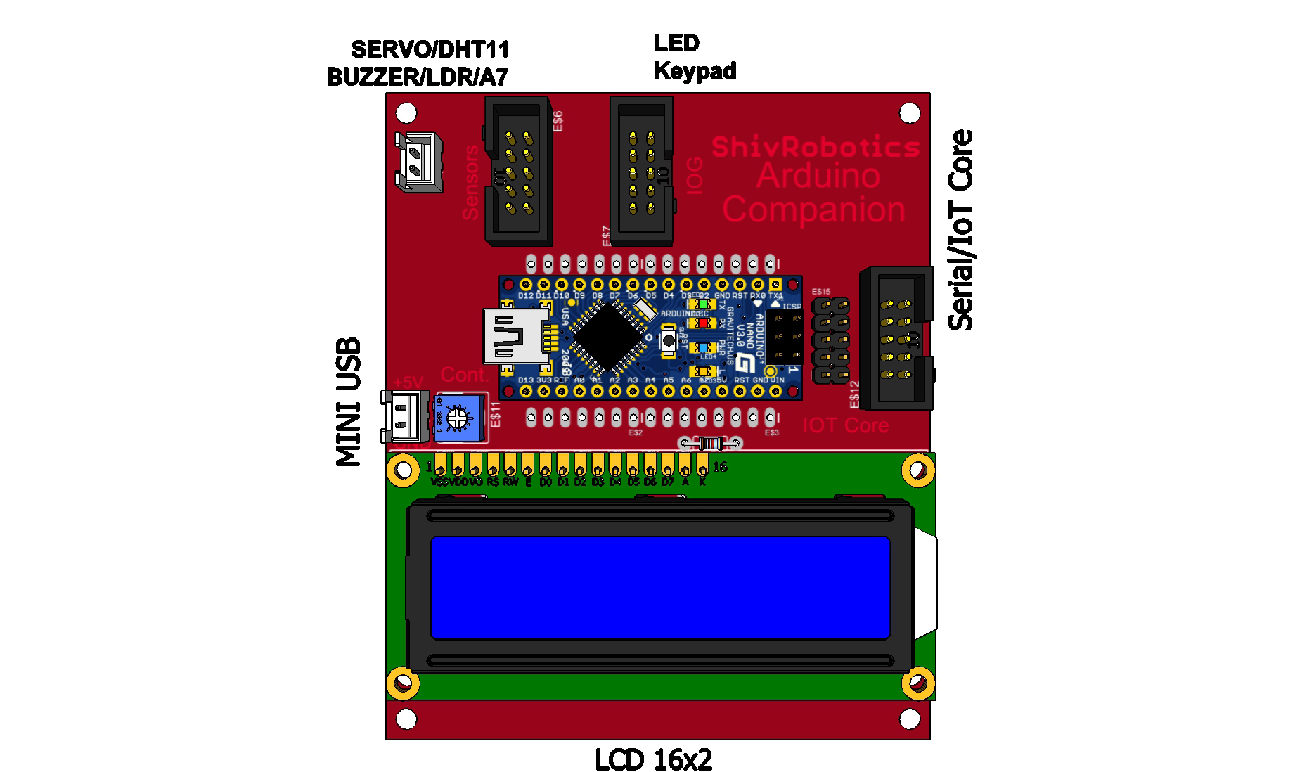
Serial.print("Light Intensity : ");

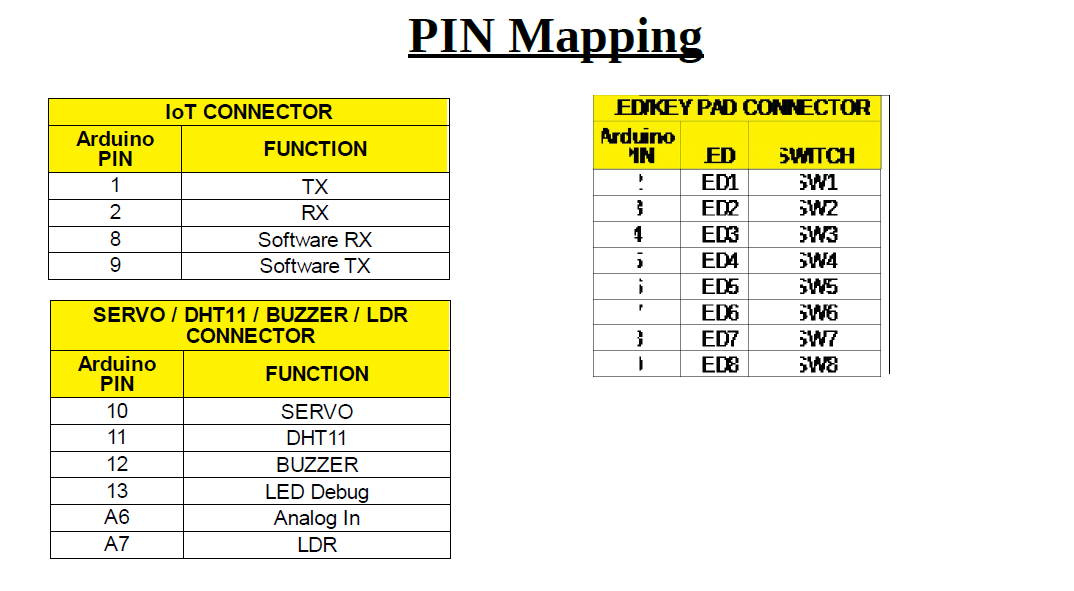
Serial.println(analogRead(lightPin));

delay(1000);

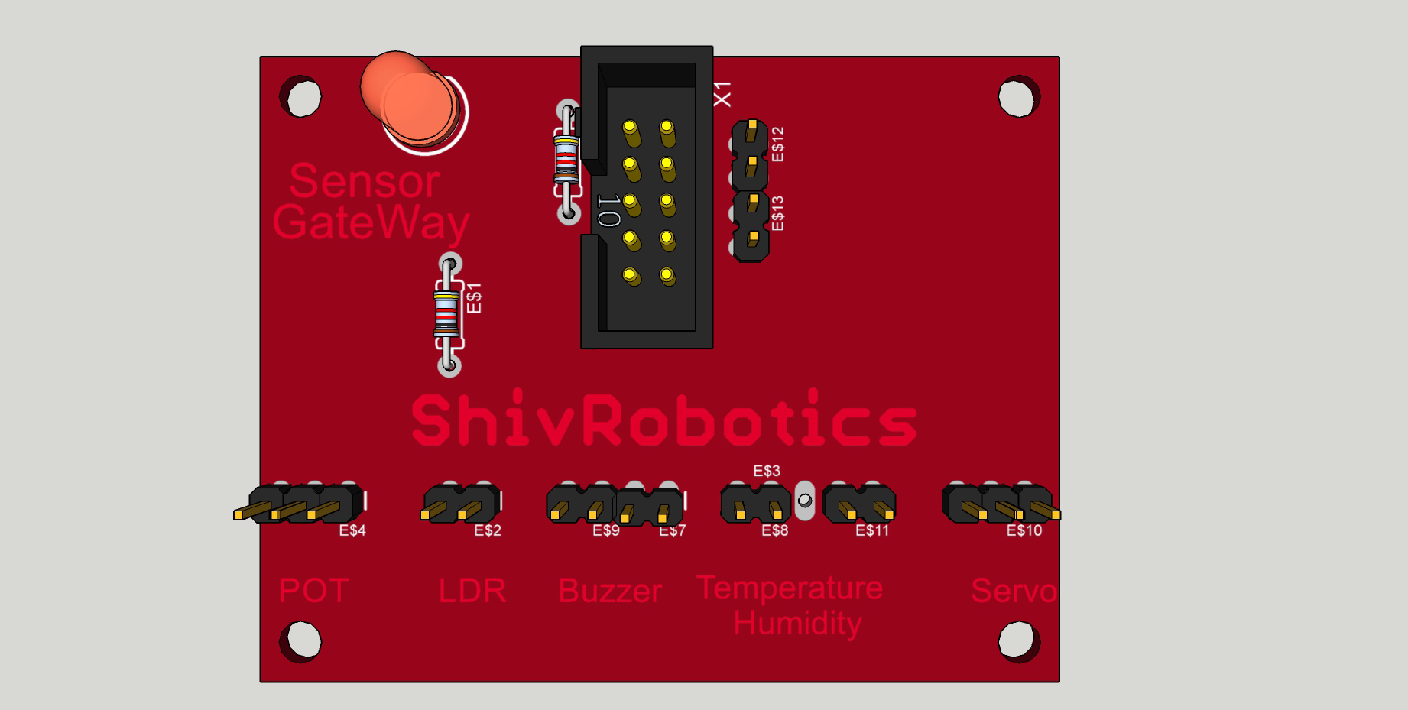
}

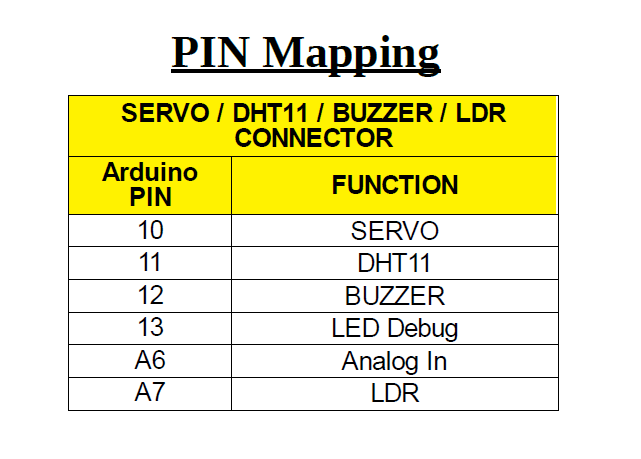
**Arduino Companion**





**Sensor Gateway**





**Output:**

Light Intensity : 234

Light Intensity : 237

Light Intensity : 242

Light Intensity : 245

Light Intensity : 236

Light Intensity : 289

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program with LDR control is written and output was verified successfully

| **EX .NO : 06** | **Arduino Program with LM35 Sensor** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the LM35 sensor.

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the LM35 sensor.

**Step4:** Enter the code to activate the LM35 sensor.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

int val;

int tempPin = 6;

void setup()

{

Serial.begin(9600);

}

void loop()

{

val = analogRead(tempPin);

float mv = ( val/1024.0)\*5000;

float cel = mv/10;

float farh = (cel\*9)/5 + 32;

Serial.print("TEMPERATURE = ");

Serial.print(cel);

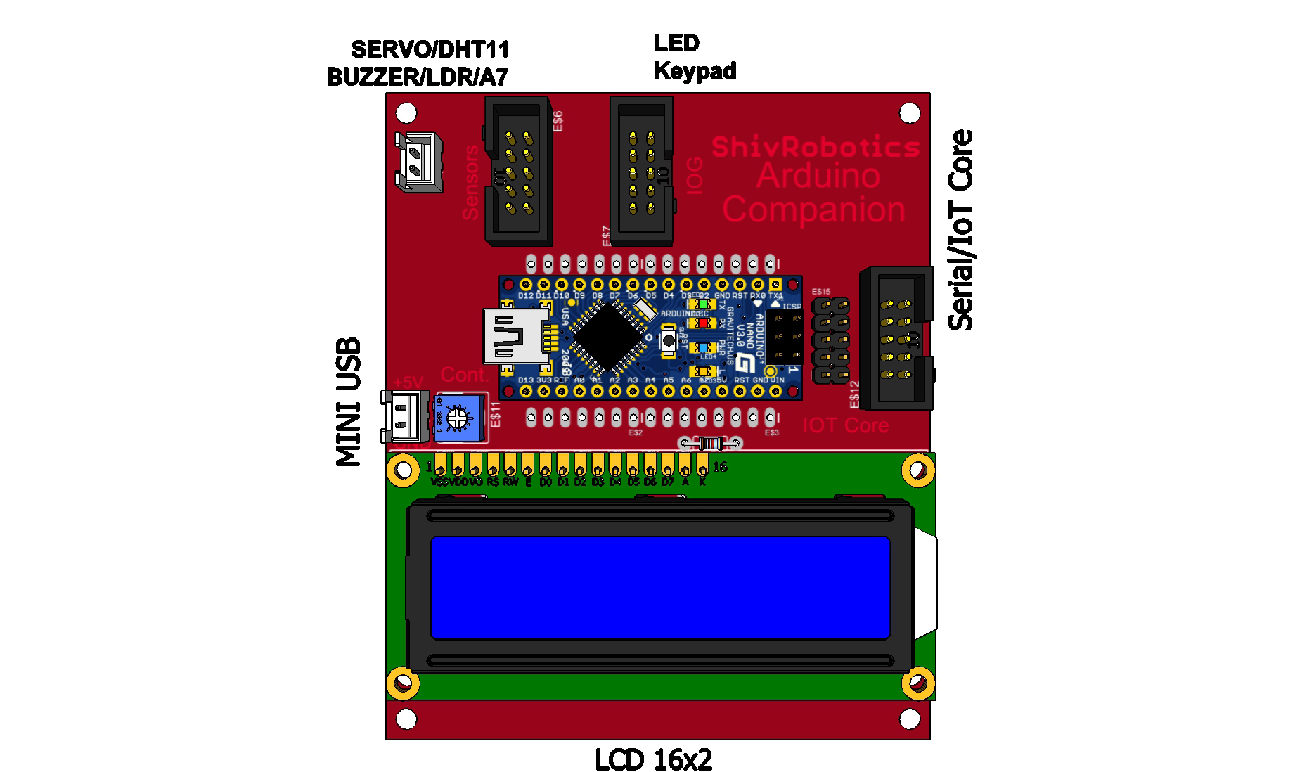
Serial.print("\*C");

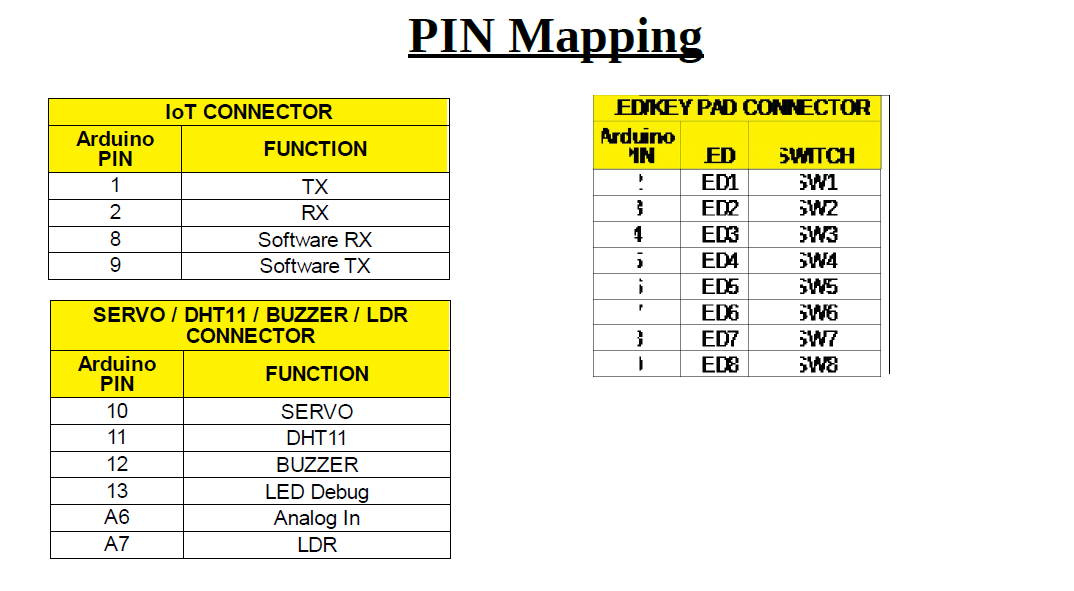
Serial.println();

delay(1000);

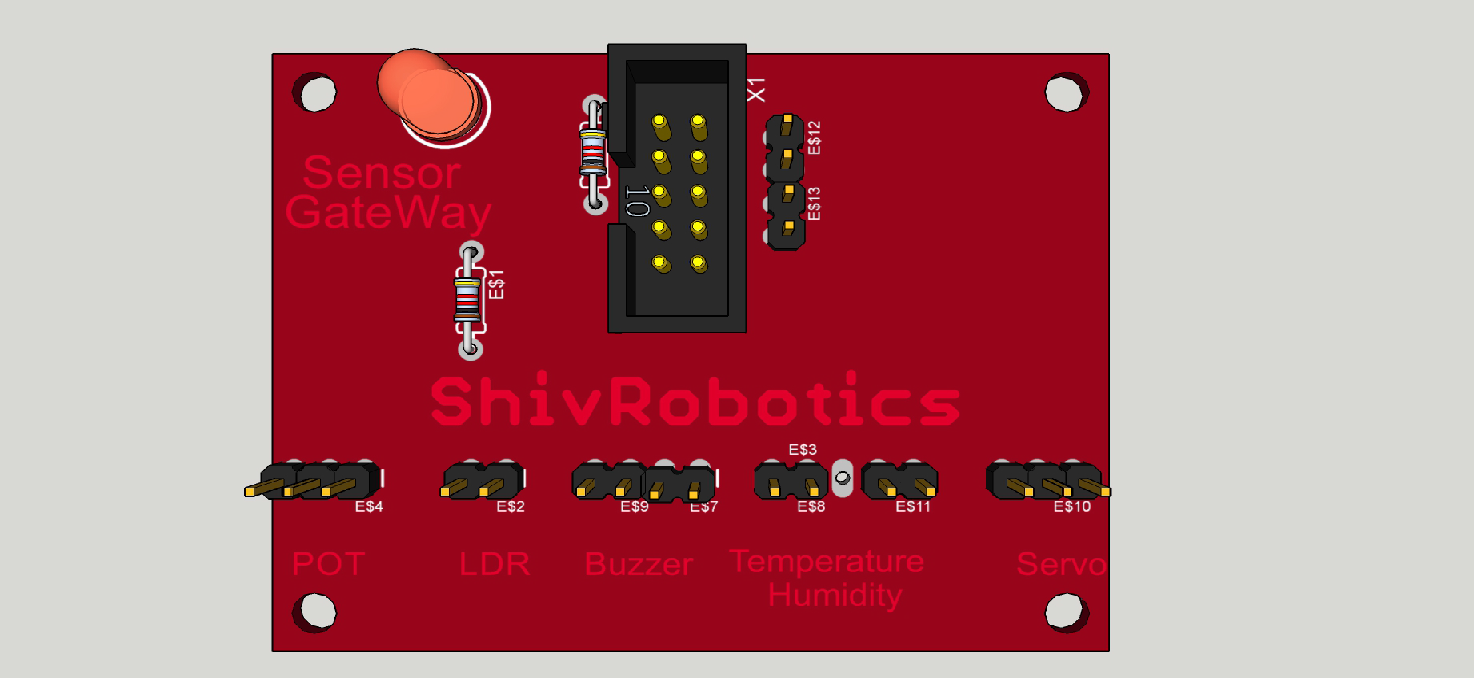
}

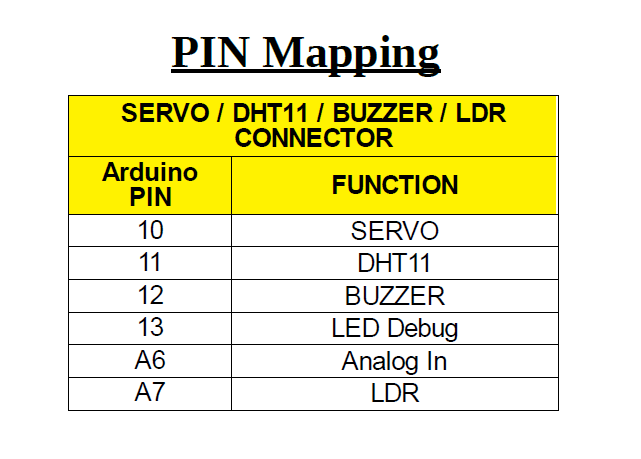
**Arduino Companion**





**Sensor Gateway**





**Output:**

TEMPERATURE = 30.60\*C

TEMPERATURE = 30.51\*C

TEMPERATURE = 30.72\*C

TEMPERATURE = 31.68\*C

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program with LM35 sensor is written and output was verified successfully

| **EX .NO : 07** | **Arduino Program for key input with LED** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the key input with LED

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the key input with LED.

**Step4:** Enter the code to activate the LED.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

int i;

void setup()

{

pinMode(9, INPUT);

digitalWrite(9, HIGH);

for (i =2;i<=7;i++)

{

pinMode(i, OUTPUT);

digitalWrite(i, HIGH);

}

}

void loop()

{

boolean var=digitalRead(9);

if(!var)

{

for (i =2;i<=7;i++)

{

digitalWrite(i, LOW);

delay(2000);

}

for (i =2;i<=7;i++)

{

digitalWrite(i, HIGH);

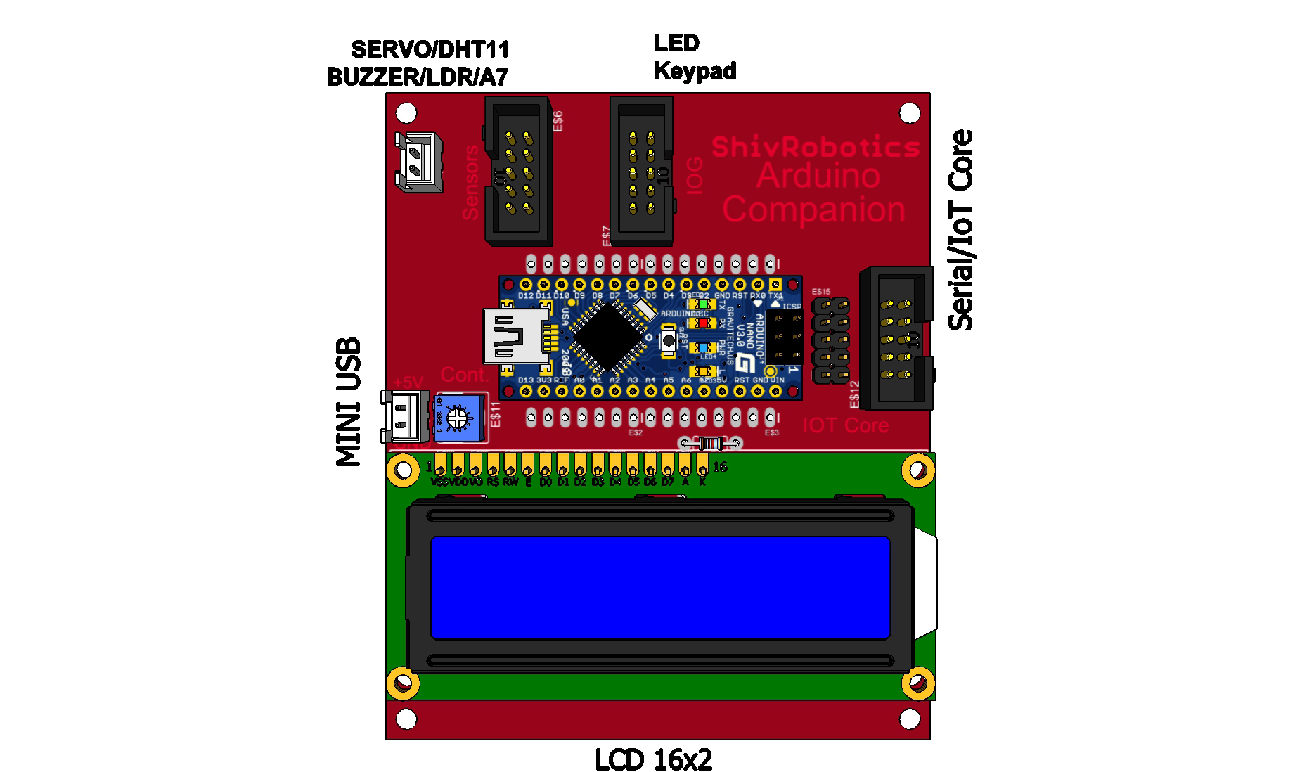
delay(2000);

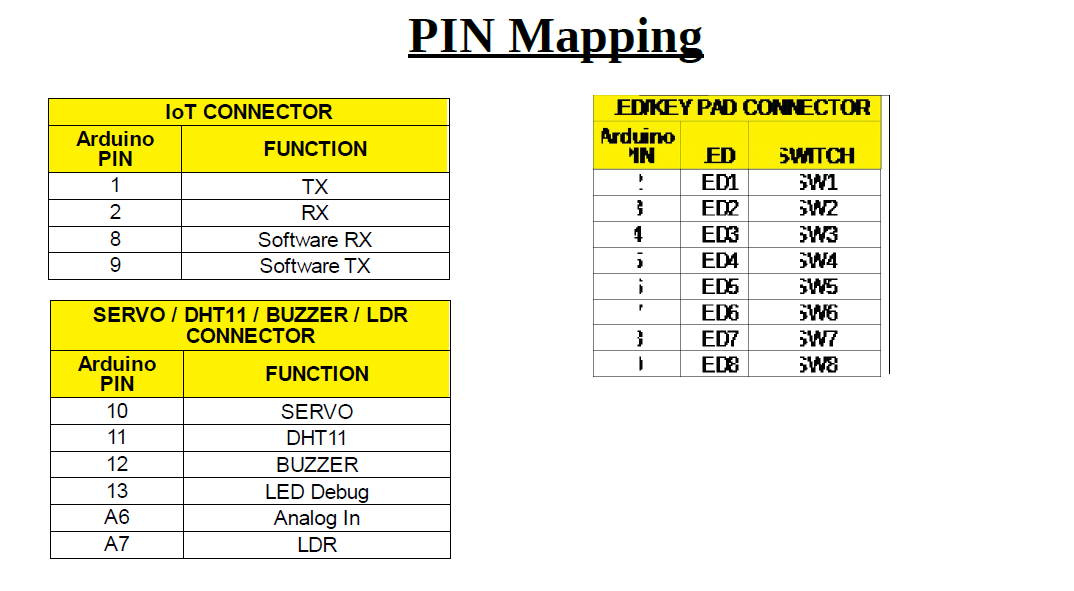
}

}

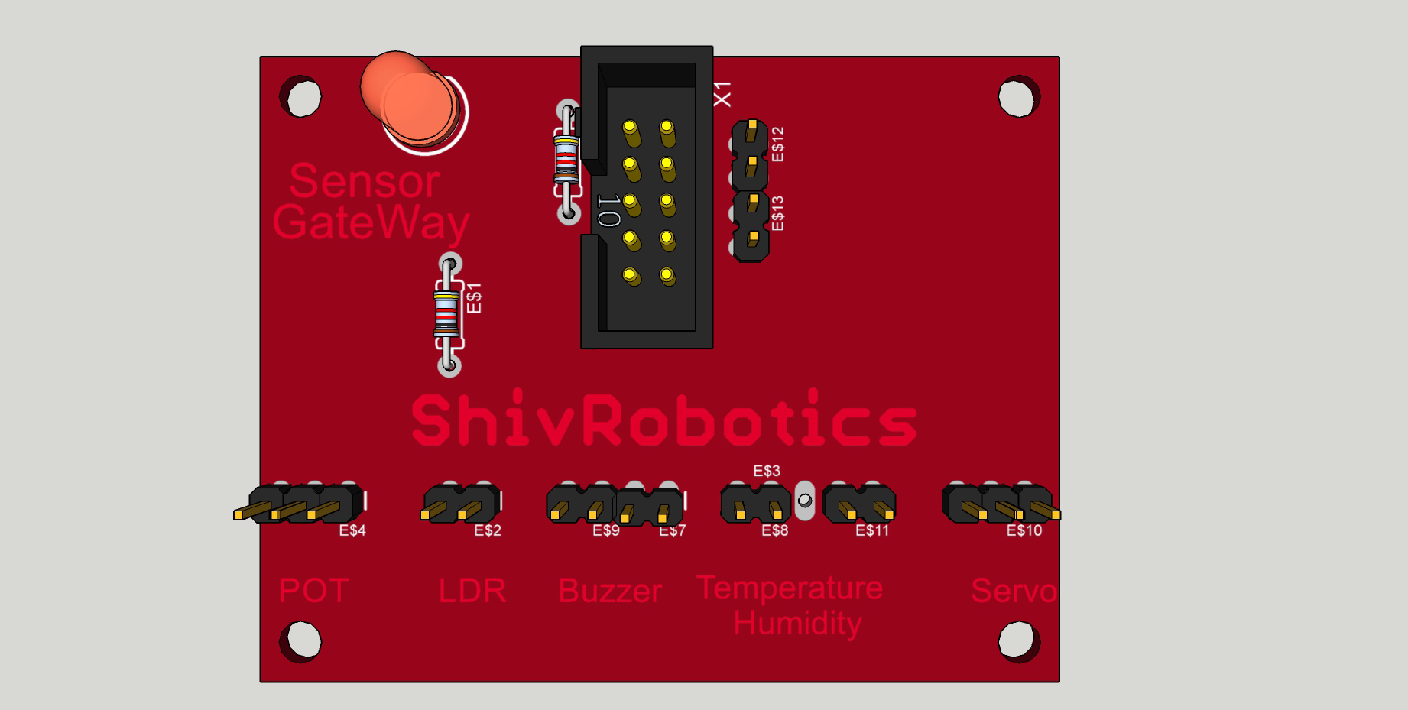
}

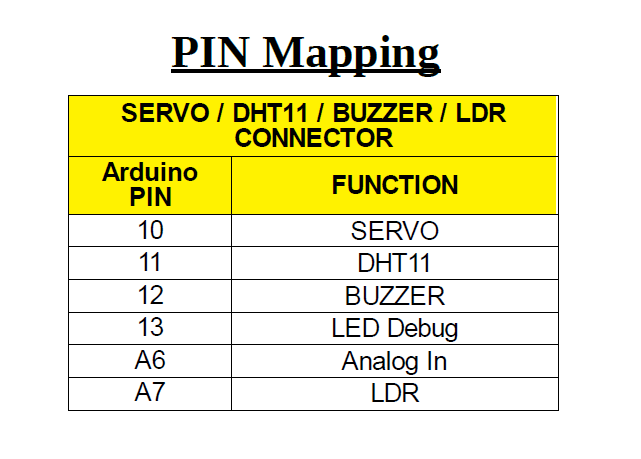
**Arduino Companion**





**Sensor Gateway**





**Output:**

| **Input** | **Output** |
| --- | --- |
| digitalWrite HIGH | LED ON |
| digitalWrite LOW | LED OFF |

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program with LED key input is written and output was verified successfully.

| **EX .NO : 08** | **Arduino Program for servo motor control** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the Servo Motor control**.**

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the Servo Motor.

**Step4:** Enter the code to activate the Servo Motor.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

#include <Servo.h>

Servo myservo;

int pos = 0;

void setup()

{ myservo.attach(10); }

void loop()

{ for(pos = 0; pos<= 180; pos += 10)

{ myservo.write(pos);

delay(1000);

}

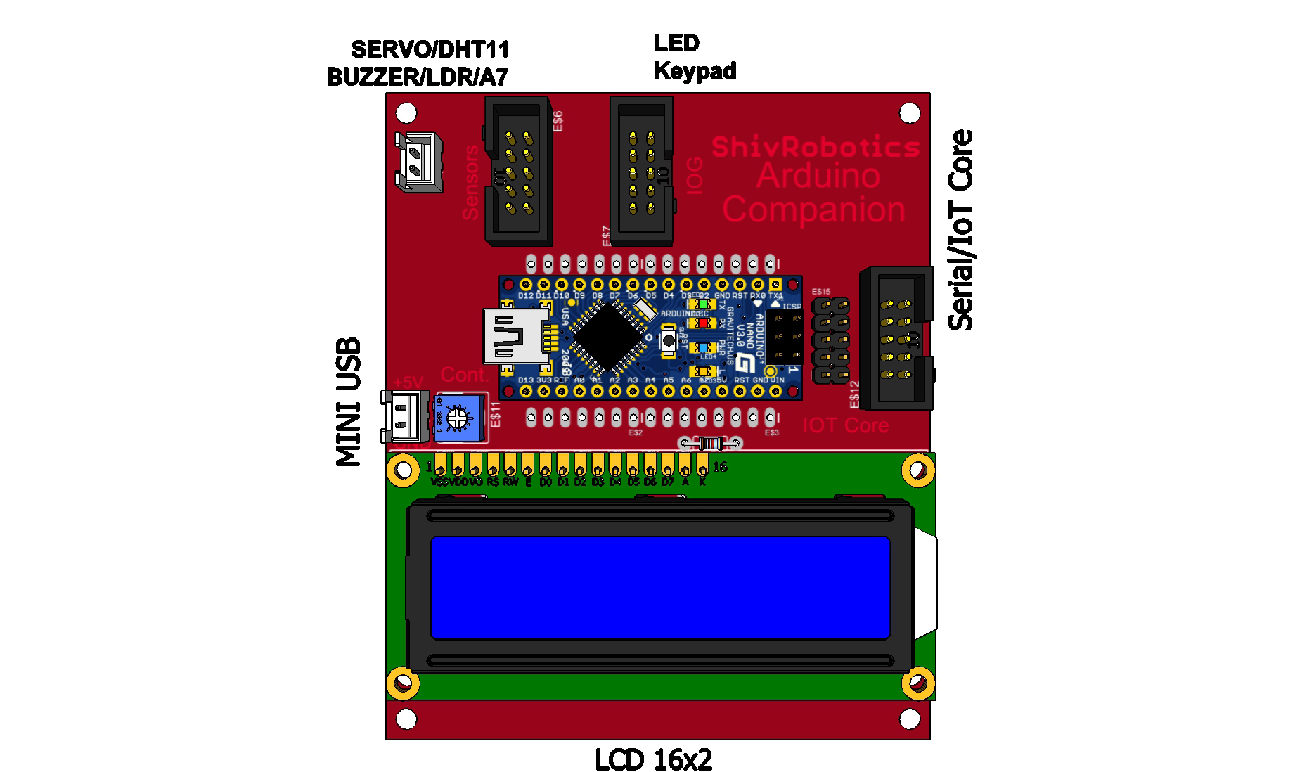
for(pos = 180; pos>=0; pos-=10)

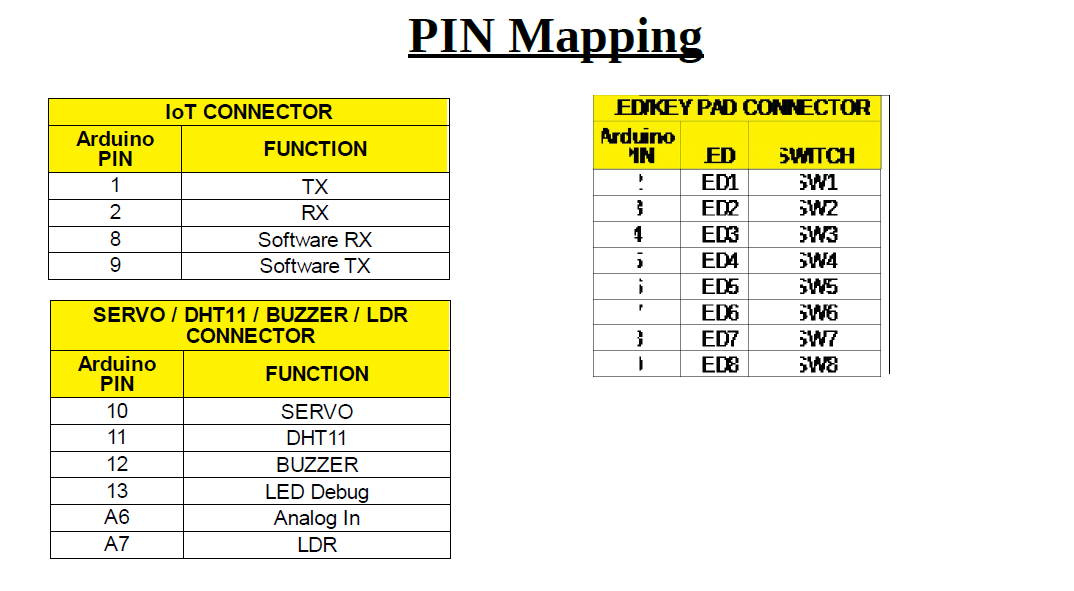
{

myservo.write(pos);

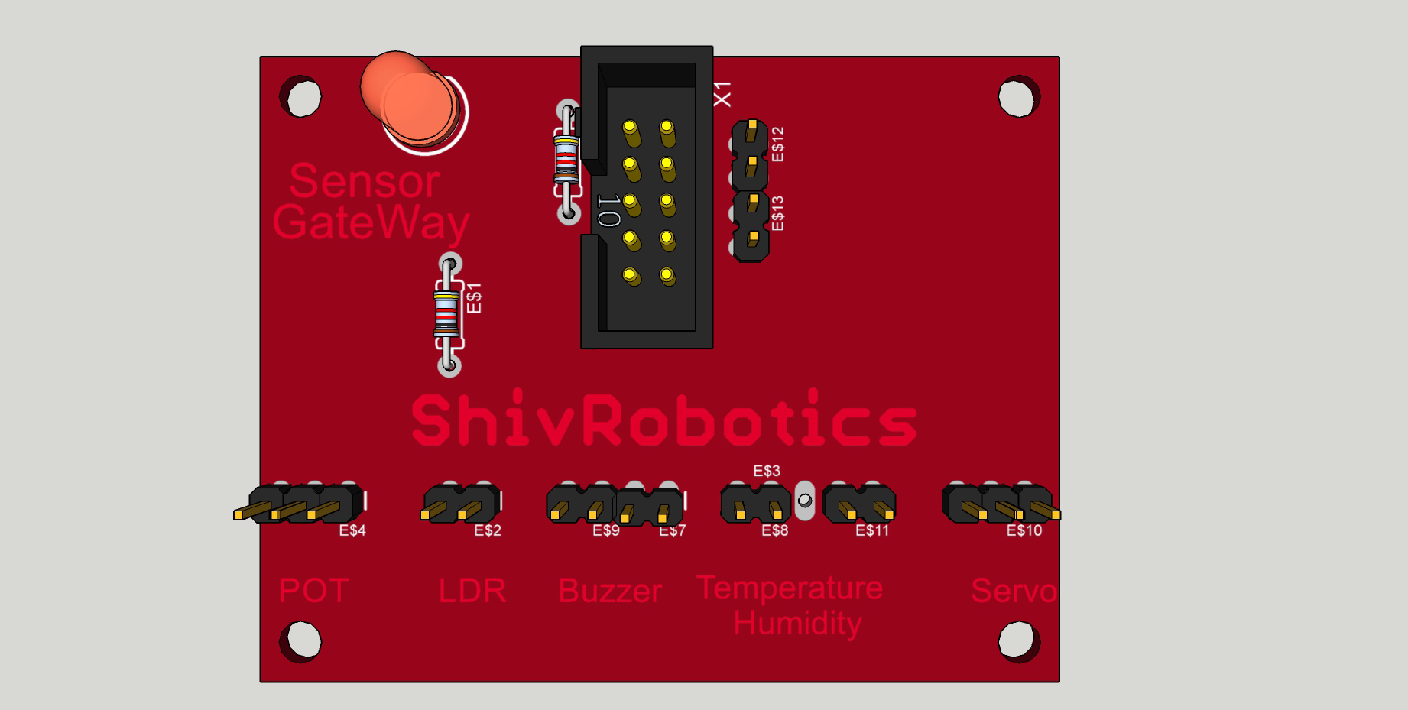
delay(1000); } }

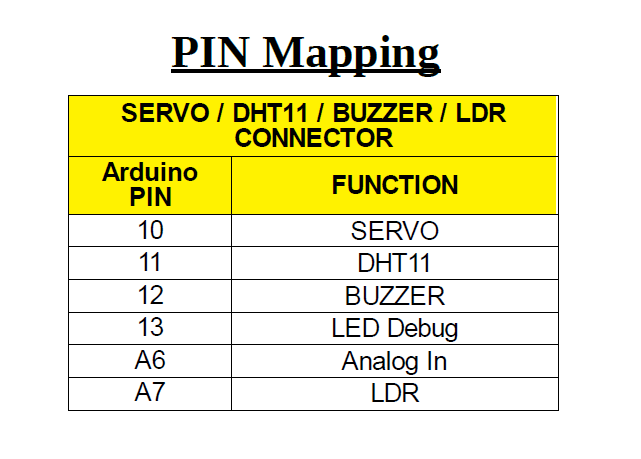
**Arduino Companion**

****



**Sensor Gateway**





**Output:**

Servo motor Rotates from position 0 to 180 and reverse back from position 180 to 0

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program with Servo Motor control is written and output was verified successfully.

| **EX .NO : 09** | **Arduino Program for blinking LED using NODEMCU with Blynk** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the LED blink using NODEMCU with Blynk**.**

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Set the pin mode to control the LED.

**Step4:** Enter the code to activate the LED.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "";

char pass[] = "";

int LED = D8; // Define LED as an Integer (whole numbers) and pin D8 on Wemos D1 Mini Pro

void setup()

{

// Debug console

Serial.begin(115200);

pinMode(LED, OUTPUT); //Set the LED (D8) as an output

Blynk.begin(auth, ssid, pass);

}

void loop()

{

Blynk.run();

}

// This function will be called every time button Widget

// in Blynk app writes values to the Virtual Pin V3

BLYNK\_WRITE(V3) {

int pinValue = param.asInt(); // Assigning incoming value from pin V3 to a variable

if (pinValue == 1) {

digitalWrite(LED, HIGH); // Turn LED on.

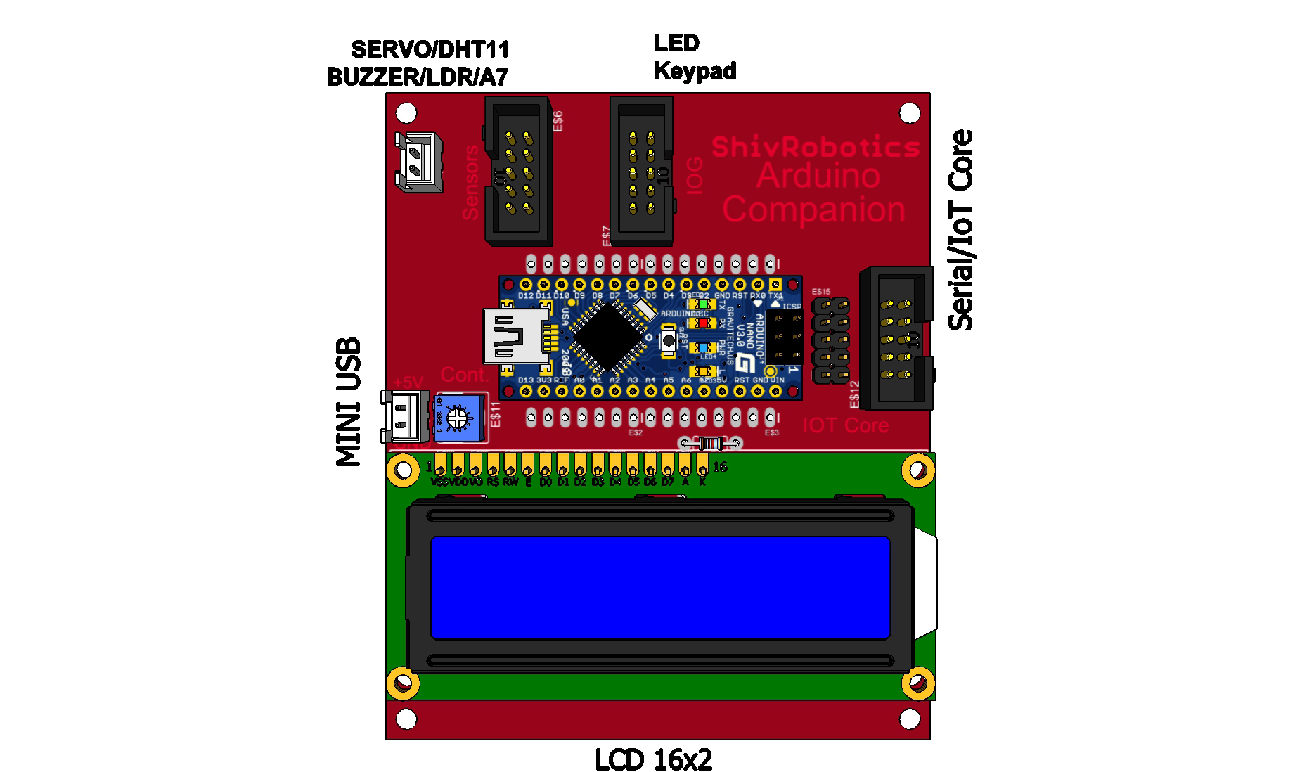
} else {

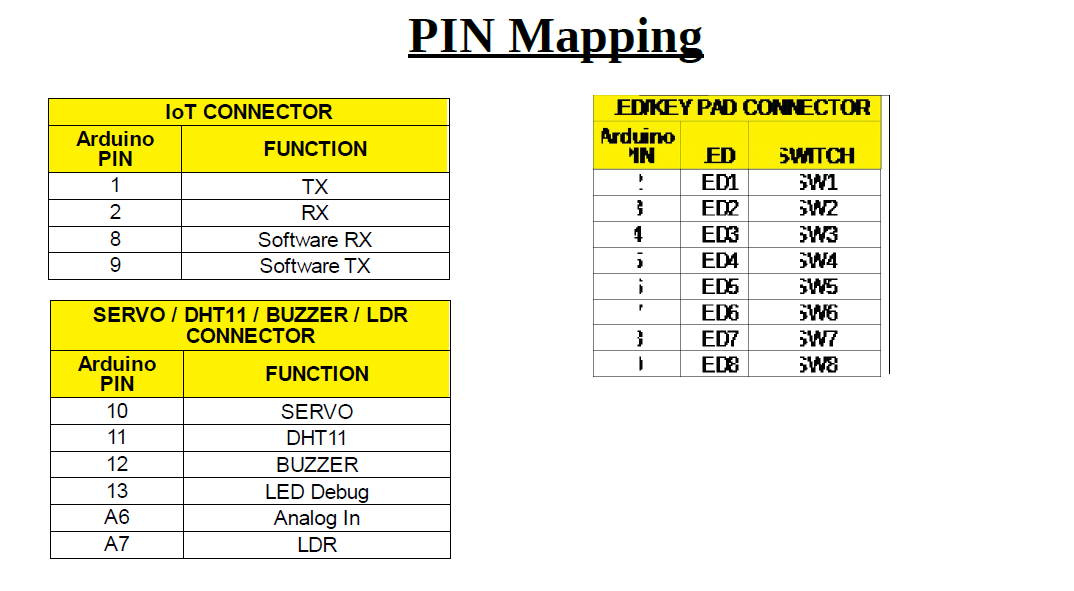
digitalWrite(LED, LOW); // Turn LED off.

}

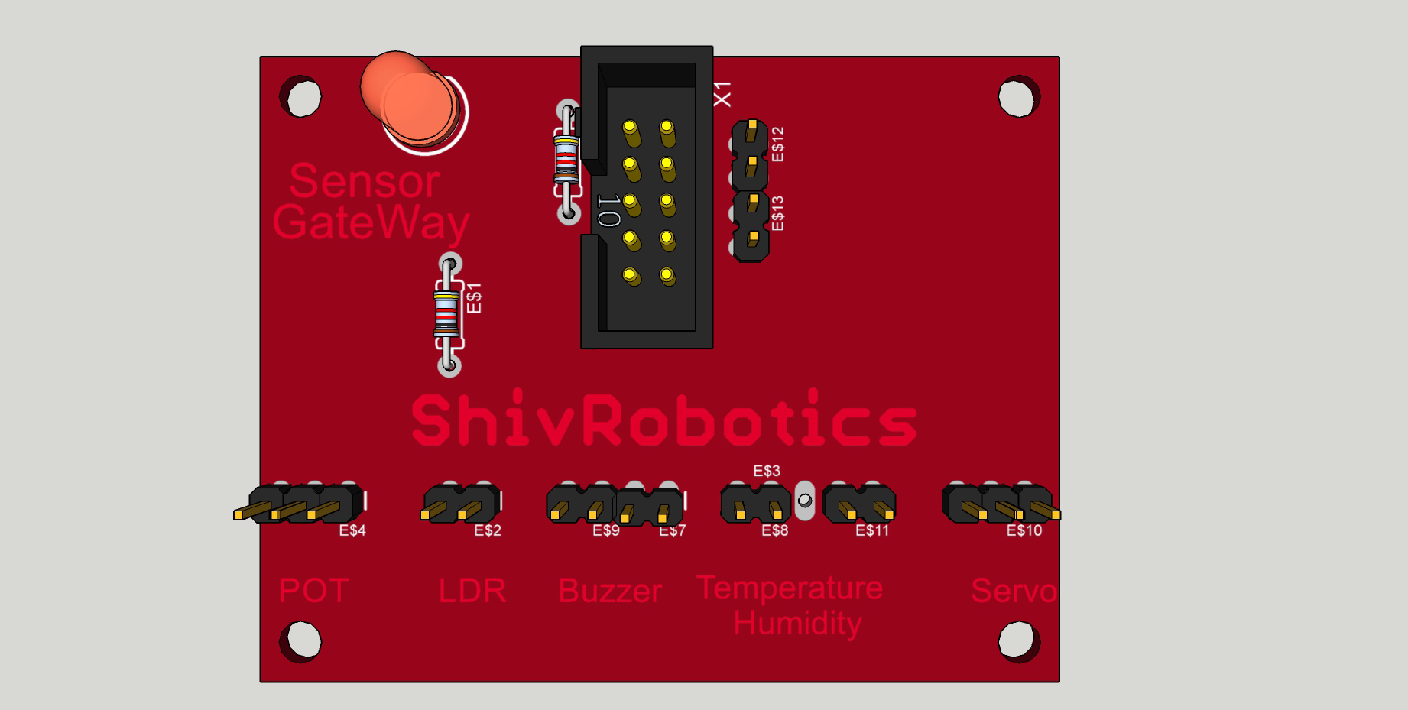
}

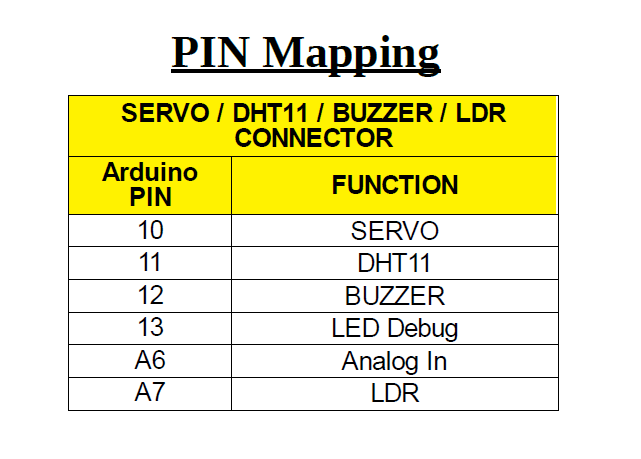
**Arduino Companion**

****



**Sensor Gateway**





**Output:**

| **Input** | **Output** |
| --- | --- |
| digitalWrite HIGH | LED ON |
| digitalWrite LOW | LED OFF |

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program to blink an LED using NODEMCU with Blynk is written and output was verified successfully.

| **EX .NO : 10** | **Sensor value logging in Cloud** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program to implement the sensor value logging in cloud.

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Enter the code to activate the sensor.

**Step4:** Check the code before executing.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

**#include<SoftwareSerial.h>**

**#include <LiquidCrystal.h>**

**#include "DHT.h"**

**SoftwareSerial iot(8,9);**

**String response;**

**int pos = 0;**

**float h;**

**float t;**

**float f;**

**LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);**

**int i;**

**int demo\_wait = 2000;**

**int lightPin = A6;**

**int anapin = A7;**

**#define DHTPIN 11**

**#define DHTTYPE DHT11**

**DHT dht(DHTPIN, DHTTYPE);**

**void setup() {**

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

**Serial.begin(115200);**

**pinMode(12,OUTPUT);**

**digitalWrite(12,LOW);**

**lcd.begin(16, 2);**

**// Print a message to the LCD.**

**lcd.print("Welcome To NEC");**

**lcd.setCursor(0,1);**

**lcd.print("IoT Experimenter");**

**delay(3000);**

**lcd.clear();**

**}**

**void loop() {**

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

**h = dht.readHumidity();**

**t = dht.readTemperature();**

**f = dht.readTemperature(true);**

**response += String(t,0);**

**response += String(",");**

**response += String(h,0);**

**response += String("#");**

**Serial.println(response);**

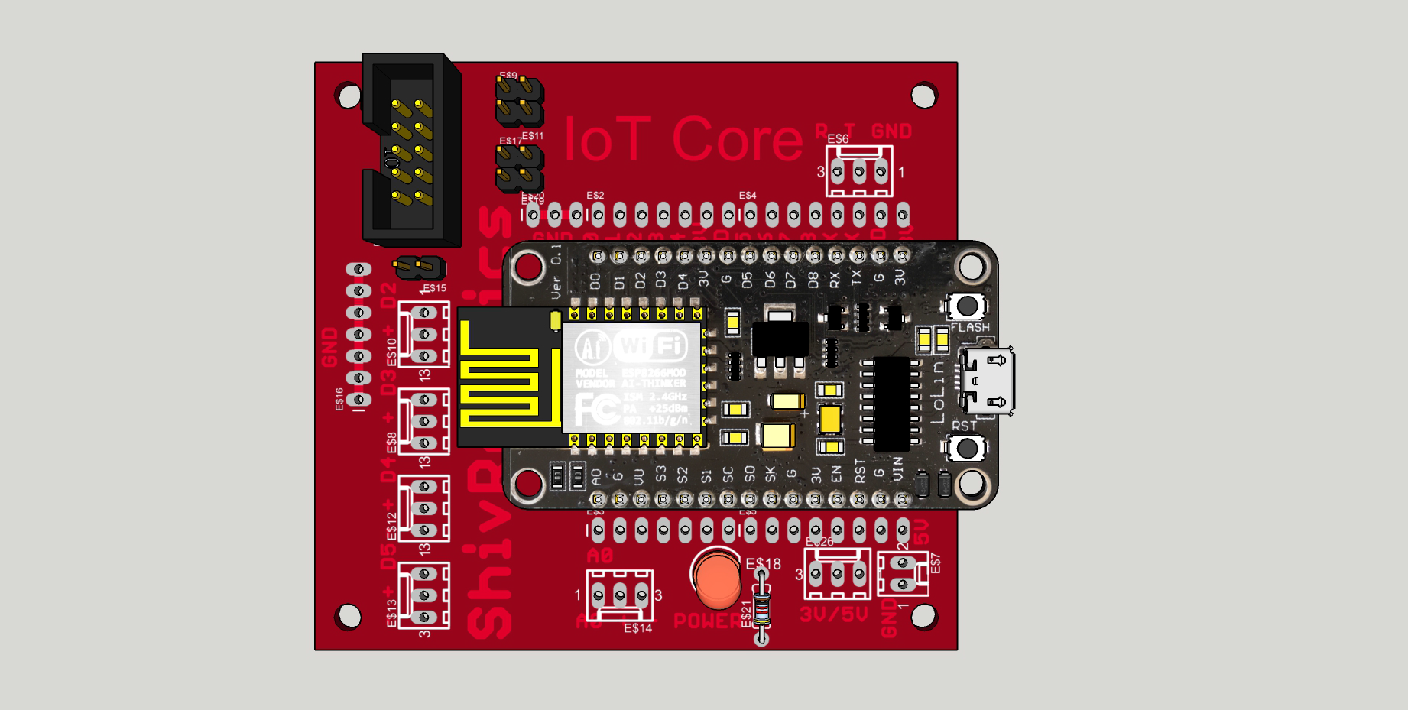
**lcd.clear();**

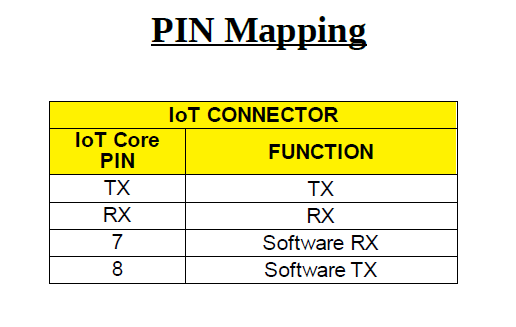
**lcd.print("Waiting for 2s");**

**delay(2000);**

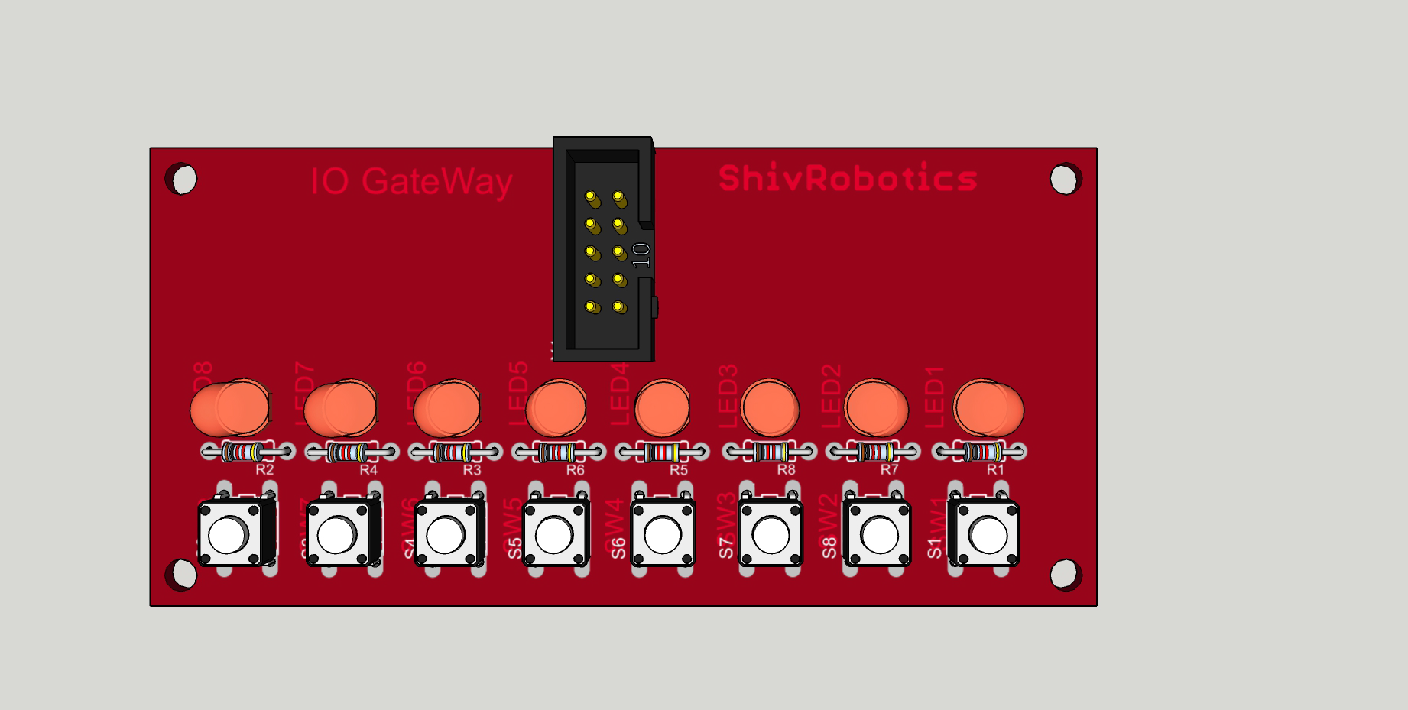
**}**

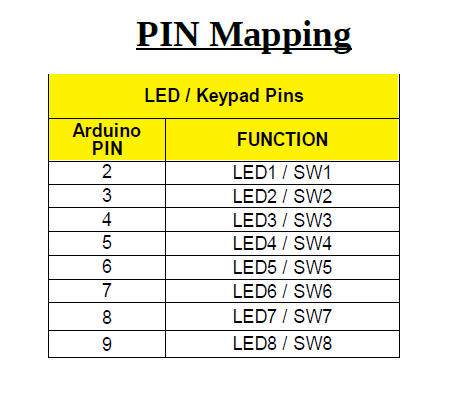
**IoT Core**

****

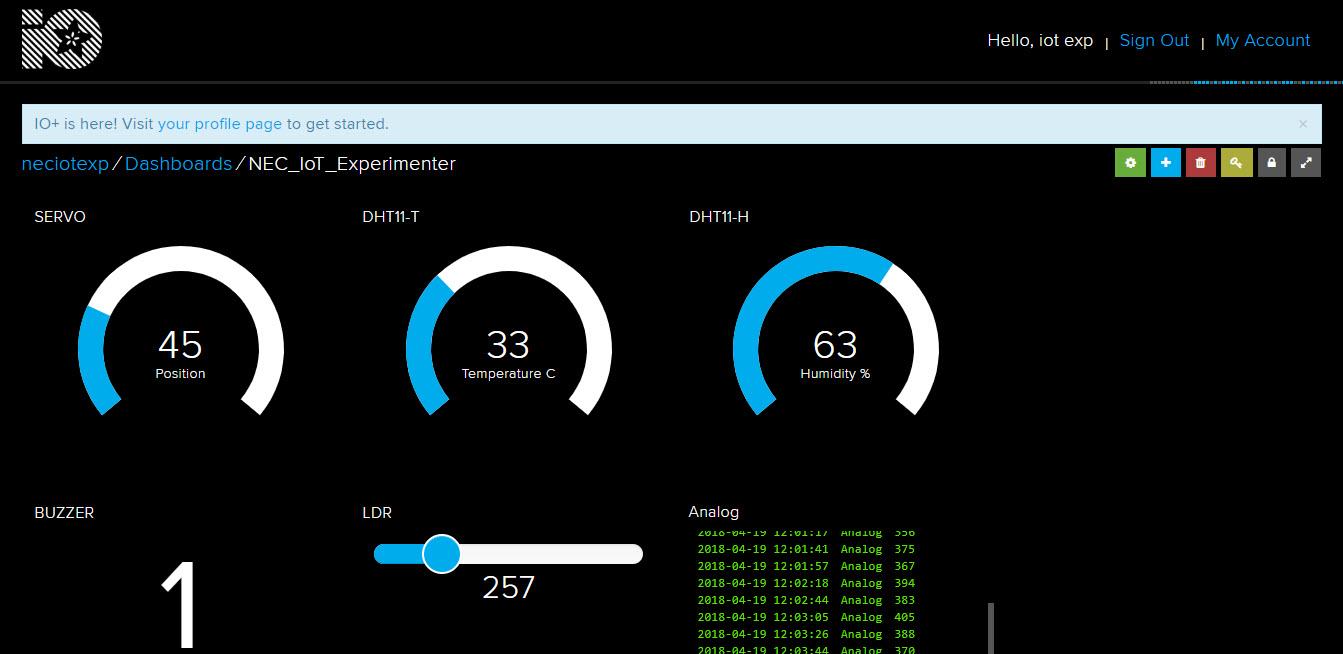


**IO Gateway**





**Output:**



| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, the Arduino program to implement the sensor value logging in cloud is written and output was verified successfully

| **EX .NO : 11** | **Deep sleep with timer wakeup using NODEMCU** |
| --- | --- |
| **DATE :** |

**AIM:**

To write the Arduino program for deep sleep with timer wakeup using NODEMCU

**Procedure:**

**Step1:** Connect the Arduino board with the computer.

**Step2:** Assign the ports to communicate with Arduino kit.

**Step3:** Enter the code to activate

**Step4:** Check the code before executing.

**Step5:** Upload and execute the program.

**Step6:** Observe the result.

**Program:**

void setup() {

Serial.begin(115200);

Serial.setTimeout(2000);

// Wait for serial to initialize.

while(!Serial) { }

// Deep sleep mode for 30 seconds, the ESP8266 wakes up by itself when GPIO 16 (D0 in NodeMCU board) is connected to the RESET pin

Serial.println("I'm awake, but I'm going into deep sleep mode for 30 seconds");

ESP.deepSleep(30e6);

// Deep sleep mode until RESET pin is connected to a LOW signal (for example pushbutton or magnetic reed switch)

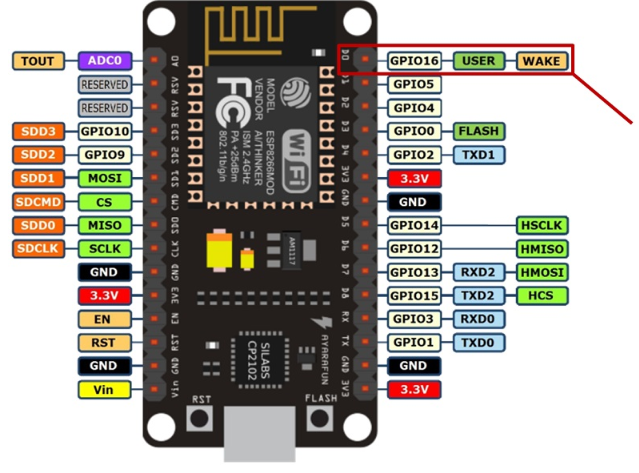
//Serial.println("I'm awake, but I'm going into deep sleep mode until RESET pin is connected to a LOW signal");

//ESP.deepSleep(0);

}

void loop() {

}

****

| **MARK ALLOCATION** | | |
| --- | --- | --- |
| Preparation and conduct of experiments | (50) |  |
| Observation & result | (30) |  |
| Record | (10) |  |
| Viva-voce | (10) |  |
| Total | (100) |  |

**Result:**

Thus, an Arduino program using NODEMCU is written and output was verified successfully