SRI BALAJI CHOCKALINGAM ENGINEERING COLLEGE

A.C.S Nagar(Irumbedu), Arni,

T.V.Malai Dt.-632 317.

Department

Of

Information Technology

CS3691-Embedded Systems And IOT Laboratory

SRI BALAJI CHOCKALINGAM ENGINEERING COLLEGE

A.C.S Nagar(Irumbedu), Arni, T.V.Malai Dt.-632 317.

Department

of

Information Technology

BONAFIDE CERTIFICATE

Certified that this is a bonafide record of work done by

Of Third Year / V Semester ***B.Tech Information Technology*** *in the Anna University Practical Examination during the year* ***20 - 20*** *in* ***CS3691 Embedded Systems And IOT Laboratory.***

Register No. :

StaffIn-Charge

Head of theDepartment

Submitted for Practical Examination held on

Internal Examiner

External Examiner

INDEX

S.NO	Date	Title	Page Number	Signature
1		Write 8051 Assembly Language experiments using simulator.		
2		Test data transfer between registers and memory.		
3		Perform Logical operations.		
4		Write Basic and arithmetic Programs Using Embedded C.		
5a		Arduino Programming for LED Blinking		
5b		Arduino Programming for ANALOG Read		
6		Communication with IOT devices		
7		Introduction to Raspberry pi and python programming. (LED interfacing with Raspberry pi)		
8		Setup a cloud platform and upload the Temperature and Humidity using DHT11()		
9		Sensor and LDR sensor to Thingspeak cloud using Raspberry pi 4 controller		

EX.NO: 1 DATE:	**Write 8051 Assembly Language experiments using simulator.**

AIM:

To write an ALP program to add, Subtract, multiply and divide two8-bit numbers using 8051 microcontroller.

Addition Program ALGORITHM:

- Clear carry.
- Load accumulator A with any desired 8-bitdata.
- Add accumulator with 8-bitnumbers.
- Store the result using DPTR.
- Stop the program.

Subtraction program ALGORITHM:

- Clear carry.
- Load accumulator A with any desired 8-bitdata.
- Subtract accumulator with 8-bitnumbers.
- Store the result using DPTR.
- Stop the program.

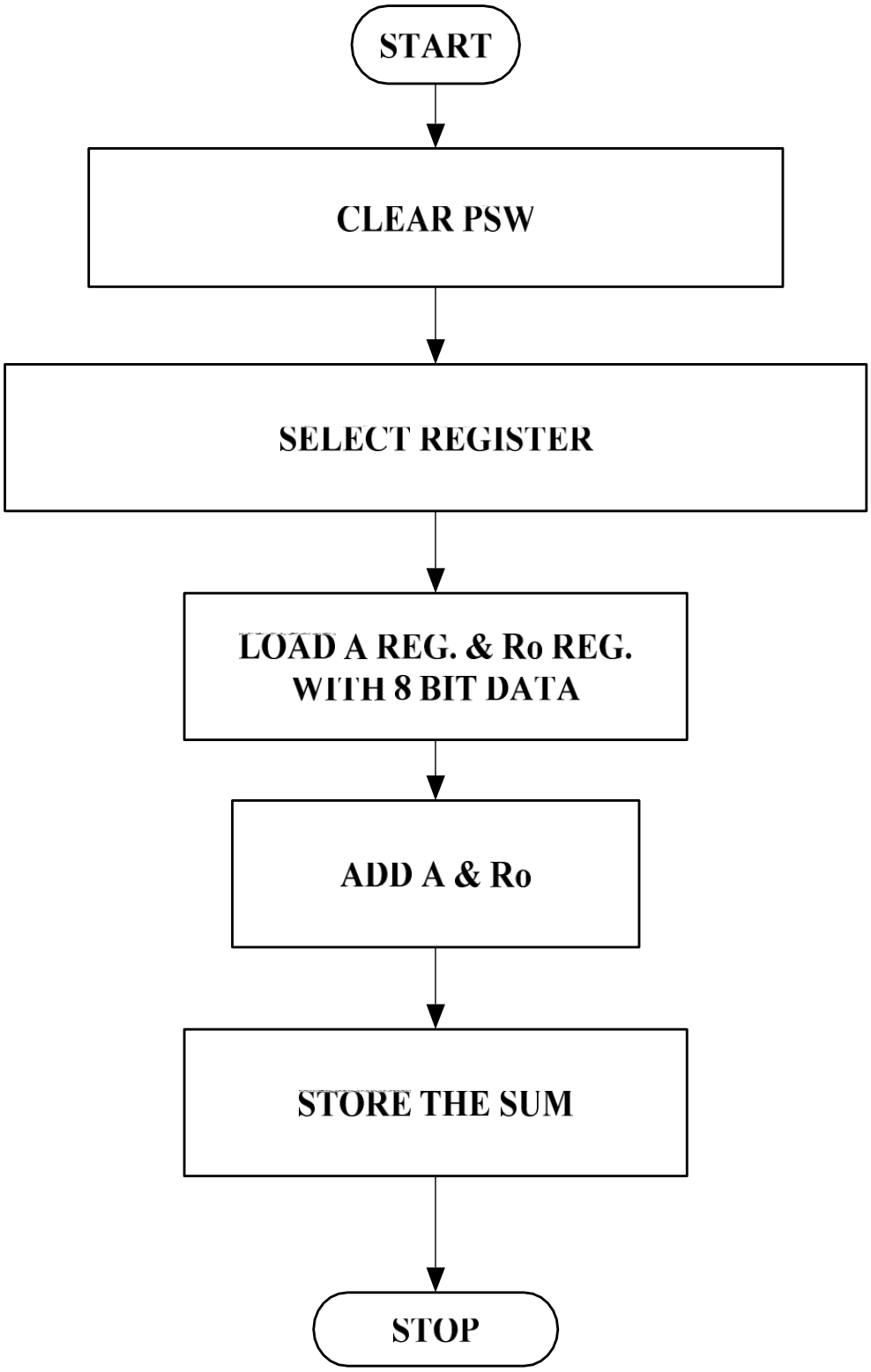
Multiplication program ALGORITHM:

- Load accumulator A with any desired 8-bitdata.
- Load B Register with any desired 8-bitdata.
- Multiply Accumulator with B register.
- Store the result Present in Accumulator and B register using DPTR.
- Stop the program.

Division program ALGORITHM:

- Load accumulator A with any desired 8-bitdata.
- Load B Register with any desired 8-bitdata.
- Divide Accumulator with Bregister.
- Store the result Present in Accumulator and B register using DPTR.
- Stop the program.

FLOWCHART


START
CLEAR PSW
SELECT REGISTER
LOAD A REG. & Ro REG.
WITH 8 BIT DATA
ADD A & Ro
STORE THE SUM
STOP

Program :

```
org 0000h
mov a, #20h
add a, #03h
Mov r0, a
clr a
clr c
Mov a, #05h
Subb a, #02h
Mov r1, a
Mov a, #03h
Mov b, #04h
Mul ab
Mov r2, a
Mov r3, b
clr a
Mov a, #95h
Mov b, #10h
div ab
Mov r4, a
Mov r5, b
END
```

OUTPUT:

INPUT		OUTPUT	
ADDITION			
ACC		R0	
Breg			
SUBTRACTION			
ACC		R1	
Breg			
MULTIPLICATION			
ACC		R3	
Breg		R4	
DIVISION			
ACC		R3	
Breg		R4	

RESULT:

Thus the 8051 ALP for Addition, Subtraction, Multiplication and Division of two 8 bit numbers is executed.

EX.NO:2 DATE:	**Test data transfer between registers and memory.**

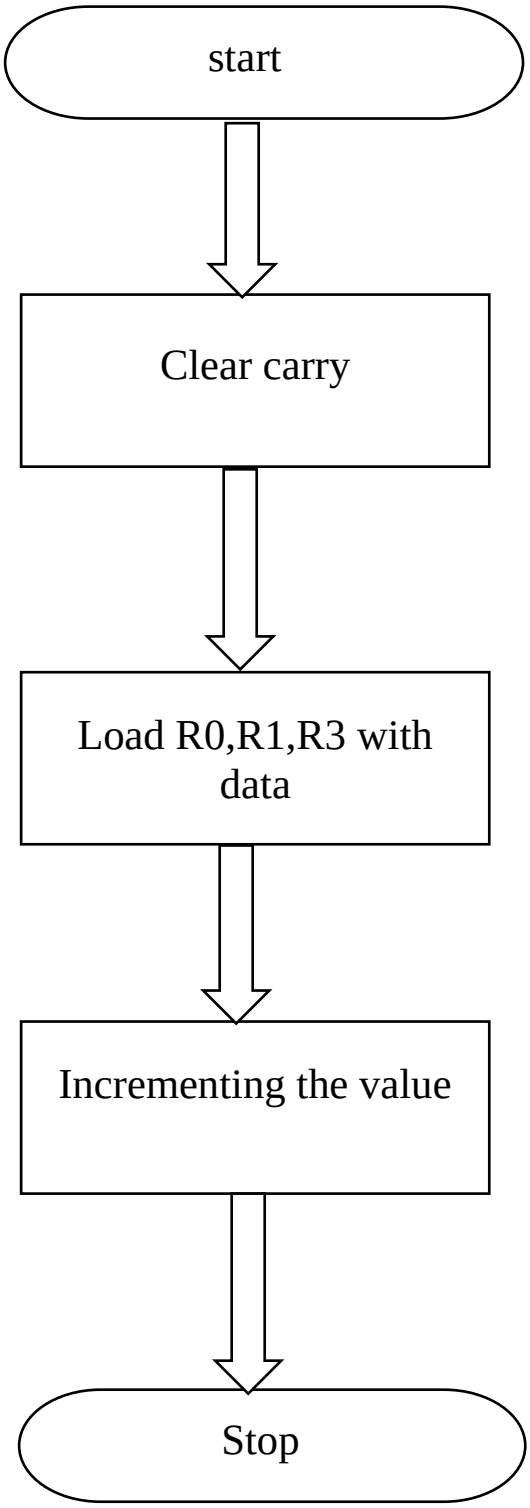
AIM:

To write an assembly language program to transfer 5 data bytes.

ALGORITHM:

- Clear carry.
- Load R0 with any desired data.
- Load R1 with any desired data.
- Load R3 with the value of 5.
- Observe the incrementing values.
- Stop the program.

FLOWCHART:


start
Clear carry
Load R0,R1,R3 with data
Incrementing the value
Stop

Program :

```
Org 00h
Mov R0, #30H
Mov R1, #40H
Mov R3, #05
Mov A2@R0
Up: Mov@R1, A
INC R0
INC R1
DJNF R1,Up
END
```

RESULT:

Thus the 8051ALP for data transfer is executed.

EX.NO:3 DATE:	**Logical operations.**

AIM:

To perform logical operation using8051microcontroller AND,OR&EX-OR.

ALGORITHM:

- ➢ Get the input value and store data in the accumulator.
- ➢ Get the second values and store the B register.
- ➢ Logical operation to perform the given number
- ➢ Store the output value in memory.

Program:

```
clr c
Mov A, #07
ANL A, #03
Mov R0, A
clr  c
Mov A, #07
ORL A, #03
Mov R1, A
clr  c
Mov A, #07
XRL A, #03
Mov R2, A
clr c
Mov A, #07
CPL A
INC A
Mov R3, A
END
```

OUTPUT:

INPUT		OUTPUT	
ADD			
DATA1		R0	
DATA2			
OR			
DATA1		R1	
DATA2			
XOR			
DATA1		R2	
DATA2			
2's COMPLEMENT			
DATA1		R3	

RESULT:

Thus the assembly language program to perform logical operations AND,OR & EX-OR and 2's Complement using 8051 Performed and the result is stored.

EX.NO:4 DATE:	**Write Basic and arithmetic Program Using Embedded C.**

AIM:

To write an Arithmetic program to add, Subtract, multiply and divide two 8-bit numbers using C Programming for 8051 microcontroller.

Addition Program ALGORITHM:

- Assign any desired 8-bit data to a variable x.
- Assign another desired 8-bit data to another variable y.
- Add two 8-bit numbers and store in another variable z.
- Store the result in Port 0

Subtraction program ALGORITHM:

- Assign any desired 8-bit data to a variable a.
- Assign another desired 8-bit data to another variable b.
- Subtract two8-bit numbers and store in another variable c.
- Store the result in Port 1

Multiplication program ALGORITHM:

- Assign any desired 8-bit data to a variable d.
- Assign another desired 8-bit data to another variable e.
- Multiplytwo8-bit numbers and store in another variable f.
- Store the result in Port 2

Division program ALGORITHM:

- Assign any desired 8-bit data to a variable p.
- Assign another desired 8-bit data to another variable q.
- Dividetwo8-bit numbers and store in another variable r.
- Store the result in Port 3
- Stop the program.

Program:

```
#include<reg51.h>
 void main(void)
{
  Unsigned char x, y, z, a, b, c, d, e, f, p, q, r;//define variables
   //addition
x=0x03; //first 8-bit number
 y=0x04;//second8-bitnumber
P0=0x00;//declare port0 as output port
z=x+y; // perform addition
P0=z;//display result on port0

//subtraction
a=0x03; //first 8-bit number
 b=0x04;//second8-bitnumber
P1=0x00;//declare port1as output port
 c=b-a; // perform subtraction
P1=c;//display result on port1

//multiplication
d=0x03; //first 8-bit number
e=0x04;//second8-bitnumber
P2=0x00;//declareport2asoutputport
 f=e*d; // perform multiplication
P2=f;//display result on port 2

//division
p=0x03; //first 8-bit number
q=0x04;//second8-bitnumber
P3=0x00;//declareport3asoutputport
r=q/p; // perform division
P3=r;//displayresultonport3
 while(1);
  }
```

Output:

INPUT		OUTPUT	
ADDITION			
DATA1		PORT0	
DATA2			
SUBTRACTION			
DATA1		PORT1	
DATA2			
MULTIPLICATION			
DATA1		PORT2	
DATA2			
DIVISION			
DATA1		PORT3	
DATA2			

RESULT:

Thus the 8051C–Programming for Addition, Subtraction, Multiplication and Division of two 8 bit numbers is executed in Keil.

EX.NO:5a DATE:	**Arduino Programming for LED Blinking**

Aim:

To control LED Using Arduino Uno board.

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Arduino board		1
3	Led		1
4	12V Adaptor		1
5	Power jack		1
6	USB Cable		1
7	Jumper Wires		Required

Hardware Procedure:

- LED pin is Connected to Arduino Uno pin of 2.
- Power jack is connected to the Arduino Uno.
- USB connector is connected to Arduino Uno to monitor.
- Connect the 12V power supply to development board.
- Check the output from the development board.

Software Procedure:

1. Click on Arduino IDE
2. Click on file
3. Click on New
4. Write a Program as per circuit Pin connections

5. Click on Save
6. Click on Verify
7. Click on Upload the code into Arduino Uno by using USB cable.

Program:

```
constintl
ed=2;
void
setup()
  {
   pinMode(led,OUTPUT);
  }
voidloop()
   {
digitalWrite(le
d,HIGH);
delay(1000);
digitalWrite(le
d, LOW);
delay(1000);
  }
```

Result:

LED is successfully controlled by Arduino microcontroller Board.

EX.NO:5b DATE:	**Arduino Programming for ANALOG Read**

Aim:

To Interface Potentiometer and IR Sensor Using Arduino Uno board.

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Arduino board		1
3	POT sensor		1
4	IR Sensor		
5	12VAdaptor		1
6	Power jack		1
7	USB Cable		1
8	Jumper Wires		Required

Hardware Procedure:

- LED pin is Connected to Arduino Uno pin of 11&12.
- POT pin is connected to the Arduino pin A1.
- Power jack is connected to the Arduino.
- USB connector is connected to Arduino Uno to monitor.
- Connect the 12V power supply to development board.
- Check the output from the development board.

Software Procedure:

- Click on Arduino IDE
- Click on file
- Click on New
- Write a Program as per circuit Pin connections
- Click on Save
- Click on Verify
- Click on Upload the code into Arduino Uno by using USB cable

Program:

```
#defineLED11

#define LD 12
#definePOTA0
void setup()
{

pinMode(LED,OUTPUT);
pinMode(LD,OUTPUT);
pinMode(POT, INPUT);
}

voidloop()

{

intx=analogRead(POT); if(x >= 512)
{

digitalWrite(LED,HIGH);
digitalWrite(LD,LOW);
}

else
{
digitalWrite(LED,LOW);
digitalWrite(LD,HIGH);
}
}
```

RESULT:

Analog POT Value (Sensors data) are successfully measured by Arduino.

EX.NO:6 DATE:	Communication with IOT devices

Aim:

To communication with IOT devices Using Arduino Uno board via GSM and Bluetooth .

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Arduino board		1
3	Bluetooth		1
4	Zigbee		
5	GSM board		
6	12V Adaptor		1
7	Power jack		1
8	USB Cable		1
9	Jumper Wires		Required

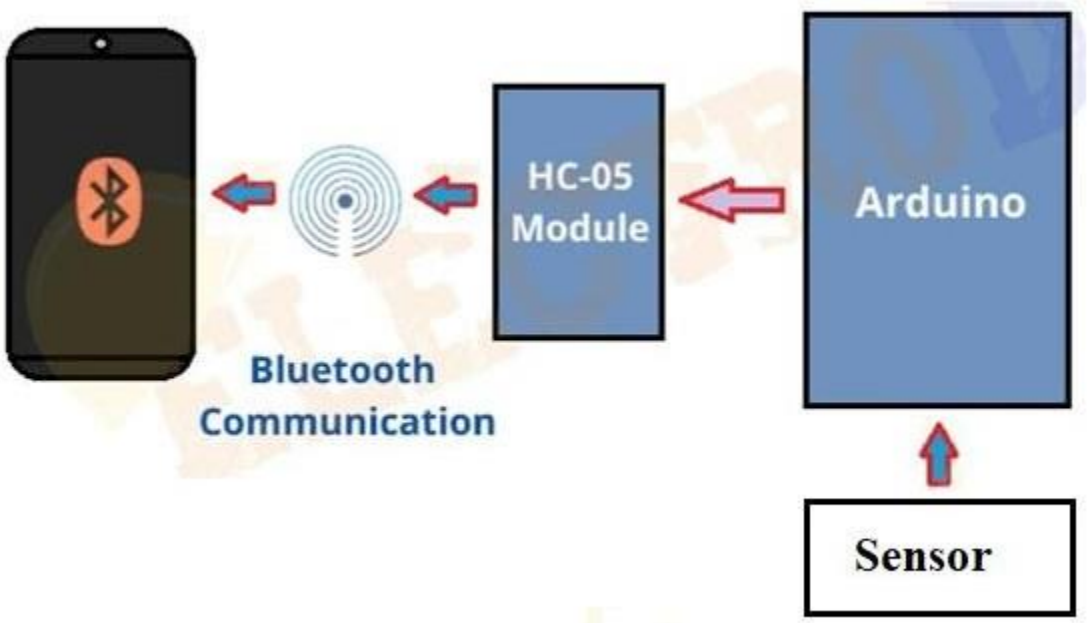
Hardware Procedure:

- Connect LM35 or LDR to Arduino Uno pin of A0.
- Read the sensor value from the Arduino pin A0.
- Power jack is connected to the Arduino.
- USB connector is connected to Arduino Uno to monitor.
- Connect the Bluetooth or Zigbee or GSM board with Arduino Uno.
- Check the output from the development board.

Software Procedure:

- Click on Arduino IDE
- Click on file
- Click on New
- Write a Program as per circuit Pin connections
- Click on Save
- Click on Verify
- Click on Upload the code into Arduino Uno by using USB cable.

BLOCK DIAGRAM BLUETOOTH INTERFACING



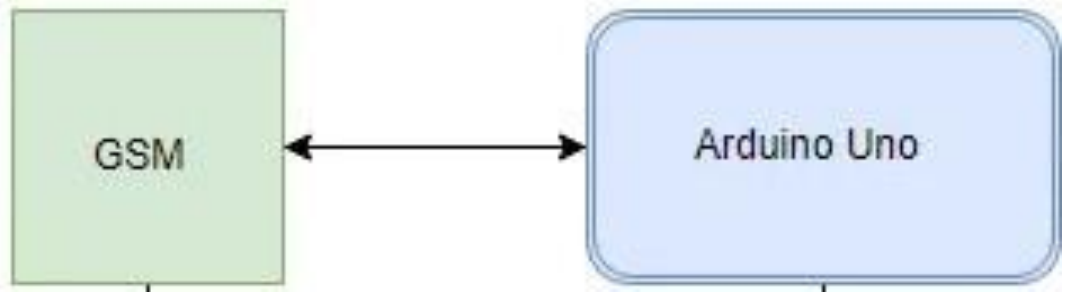
Program:

Communication using Bluetooth HC05 – Arduino Uno with Mobile App (IoTDevice)

```
int  val;
void setup()
{
Serial.begin(9600);
pinMode(A0,INPUT);
}
```

```
void loop()
{
val=analogRead(A0);
 Serial.print("Value =");
Serial.println(val);
delay(500);
}
```

BLOCK DIAGRAM GSM INTERFACING



Program:

```
#define sw1 11
int swstate1;
void setup()
{
Serial.begin(9600);
 pinMode(sw1,INPUT);
}

void loop()
{
swstate1 = digitalRead(sw1);

delay(500);
 if(swstate1 == 1)
{
Serial.println("sending SMS");
SendMessage();
delay(1000);
}
```

```
  else
  {
    Serial.println("Waiting for Emergency switch");
  }
  delay(500);

  }
 void SendMessage()
{
  Serial.println("AT");    //Sets the GSM
  Module in Text Modedelay(100);
  Serial.println((char)13);//
   ASCII code of enter
   delay(1000);
  Serial.println("AT+CMGF=1");    //Sets the GSM
  Module in Text Modedelay(100);
  Serial.println((char)13);// ASCII code
  of enter delay(1000);  // Delay of 1000
  milli seconds or 1 second
  Serial.println("ATE=0");    //Sets the GSM
  Module in Text Modedelay(100);
 Serial.println((char)13);//
   ASCII code of enter
   delay(1000);

  Serial.println("AT+CMGS=\"+919994085790\"\r"); // Replace x with mobile
  numberdelay(1000);
  Serial.println("CS 3691 – EMBEDDED SYSTEMS AND IOT
LAB");// The SMS textyou want to send
  delay(100);

    //mySerial.println("ATD+60X
  XXXXXXXX;");
  Serial.println((char)26);//
  ASCII code of CTRL+Z
  delay(5000);
  Serial.println("ATD+919994085790;"); // Replace x
    with mobile numberdelay(1000);
}
```

```
void RecieveMessage()
{
Serial.println("AT+CNMI=2,2,0,0,0"); // AT Command to receive a
 live SMSdelay(1000);
 }
```

RESULT:

Thus communication with IOT devices Using Arduino Uno board via GSM and Bluetooth is completed.

EX.NO:7 DATE:	**Introduction to Raspberry pi and python programming. (LED interfacing with Raspberry pi)**

Aim :

To Interface LED with Raspberry pi RP2040 and LM35 (or) LDR interface withRaspberry pi RP2040.

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	RP2040		1
6	Micro B Type cable		1
7	Power jack		1
8	USB Cable		1
9	Jumper Wires		Required

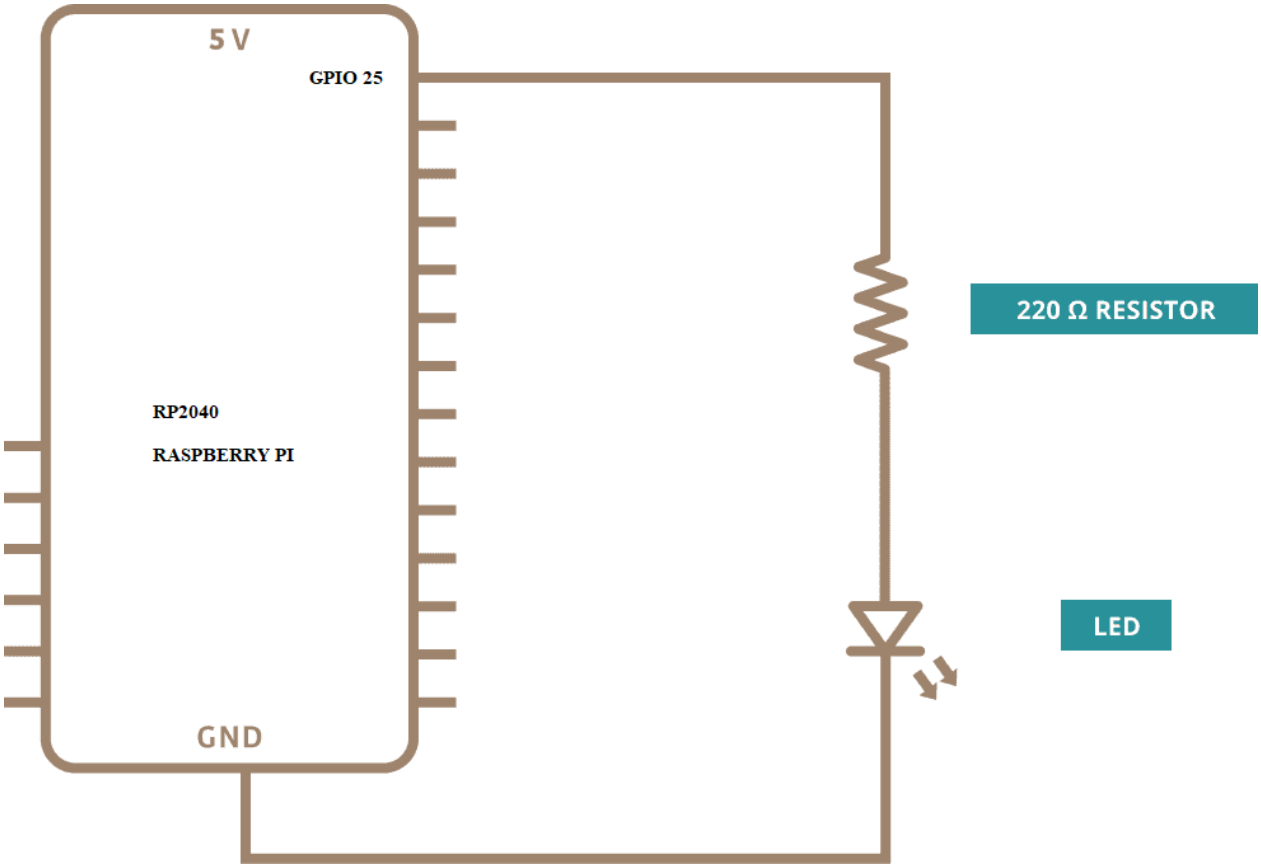
Hardware Procedure:

- Connect LED to GPIO 25
- Connect LM35 or LDR to RP2040 of A0.
- Read the sensor value from the Arduino pin A0.
- Power jack is connected to the Arduino.
- USB connector is connected to RP2040 to monitor.

Software Procedure:

- Click on Thonny
- Click on file
- Click on New
- Write a Program as per circuit Pin connections
- Click on Save
- Click on Verify
- Click on Upload the code into RP2040by using USB cable.

BLOCK DIAGRAM LED INTERFACING WITH RP2040

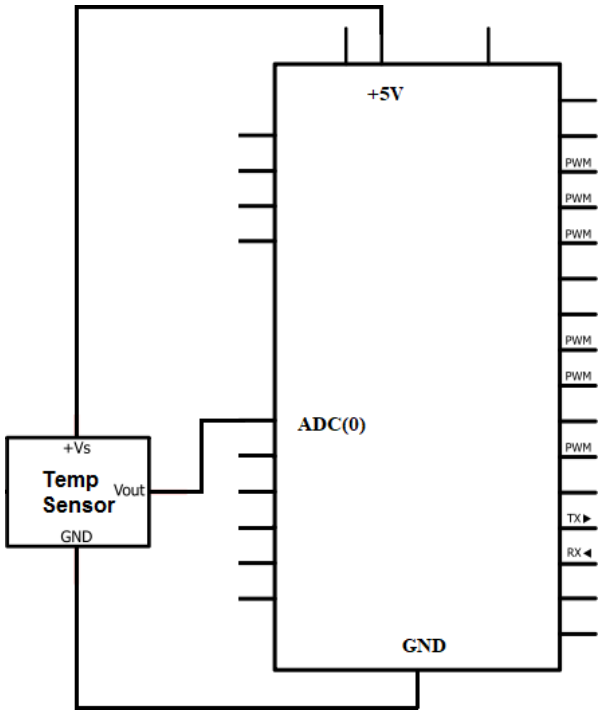


Program:

```
import time
from machine import Pin
led=Pin(25,Pin.OUT)       #create LED object from pin13,Set Pin13 to output

while True:
  led.value(1)          #Set led turn on
  time.sleep(1)
  led.value(0)          #Set led turn off
  time.sleep(1)         #delay(1 sec)
```

BLOCK DIAGRAM LM35 INTERFACING WITH RP2040



Program:

```
import machine
import utime

sensor_temp = machine.ADC(0)
 conversion_factor = 3.3 / (65535)

  while True:
     reading = sensor_temp.read_u16() *
     conversion_factortemperature = 27 -
     (reading - 0.706)/0.001721
     print("Temperature:
     {}".format(temperature))
     utime.sleep(2)
```

RESULT:

LED is successfully controlled by RP2040 and Analog LM35 Value (Sensorsdata) are successfully measured by RP2040.

EX.NO:8 DATE:	**Setup a cloud platform and upload the Temperature and Humidity using DHT11()**

Aim:

To Interface DHT11 and LDR interface with Node MCU and upload data to Thingspeak cloud and Firebase Console.

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Node MCU		1
6	Micro B Type cable		1
7	Power jack		1
8	USB Cable		1
9	Jumper Wires		Required
10	DHT11		1
11	LDR		1

Hardware Procedure:

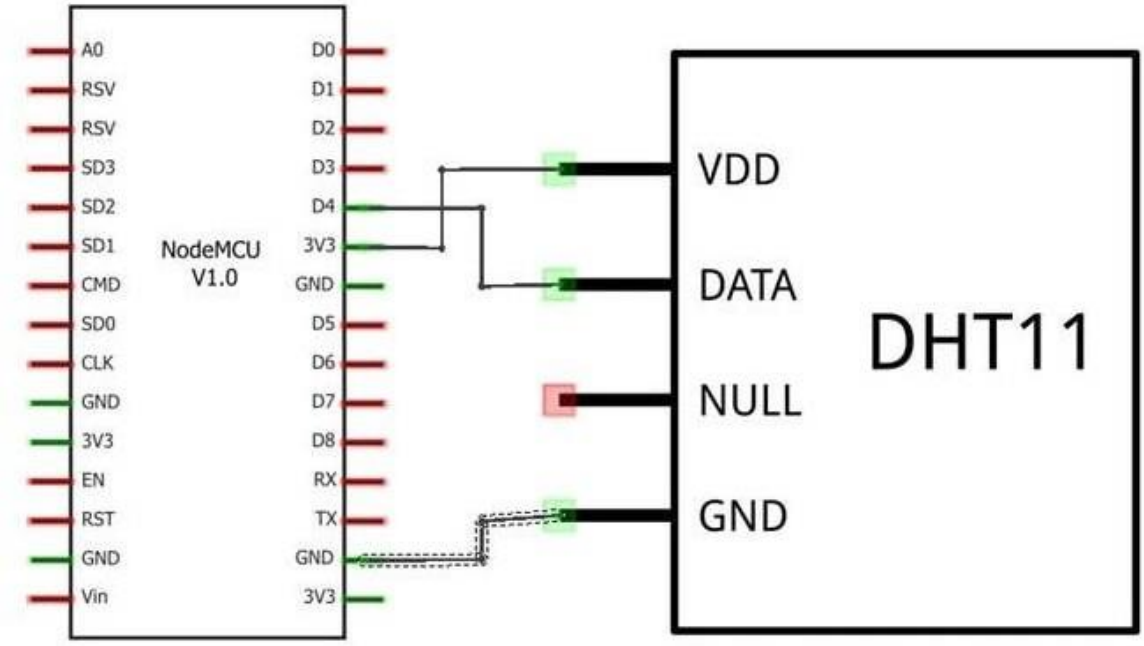
- The Dht 11 Has 4 Pins. Pin 1 Is Vcc, Pins 2 Is Data, Pin 3 Is Not Used, Pin 4 Is Ground.
- Connect Dht 11 Pin 1 To 3.3v
- Connect Dht 11 Pin 2 To Raspberry Pi Pin 16/Gpio 23 And Connect A 4.7 Or 10k Resistor From Dht 11 Pin 2 To Dht Pin 1
- Connect Dht 11 Pin 4 To Ground
- The Photo Resistor Has 2 Pins
- Connect One Pin To 3.3.V
- Connect The Other Pin To Raspberry Pi Pin 18/Gpio 24
- Connect A 1uf Capacitor To The Same Pin That The Photo Resistor Is Connected To On Gpio24. The Ground (White Stripe) Side Of The Capacitor Should Go To Ground.

Software Procedure:

- Click on Thonny
- Click on file
- Click on New
- Write a Program as per circuit Pin connections
- Click on Save
- Click on Verify
- Click on Upload the code into RP 4 by using USB cable.

- Create Channel in Thingspeak.com
- And Monitor the data uploaded in cloud

BLOCK DIAGRAM DHT11 INTERFACING WITH NodeMCU



PROGRAM TO UPLOAD TEMPERATURE AND HUMIDITY TO FIREBASE CONSOLE:

```
#include <DHT.h>
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#define FIREBASE_HOST "esiotlabpro-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "F6sgxiyuFaFkVWY9imfB1IhVO2m2HYCQq9FX49xQ"
```

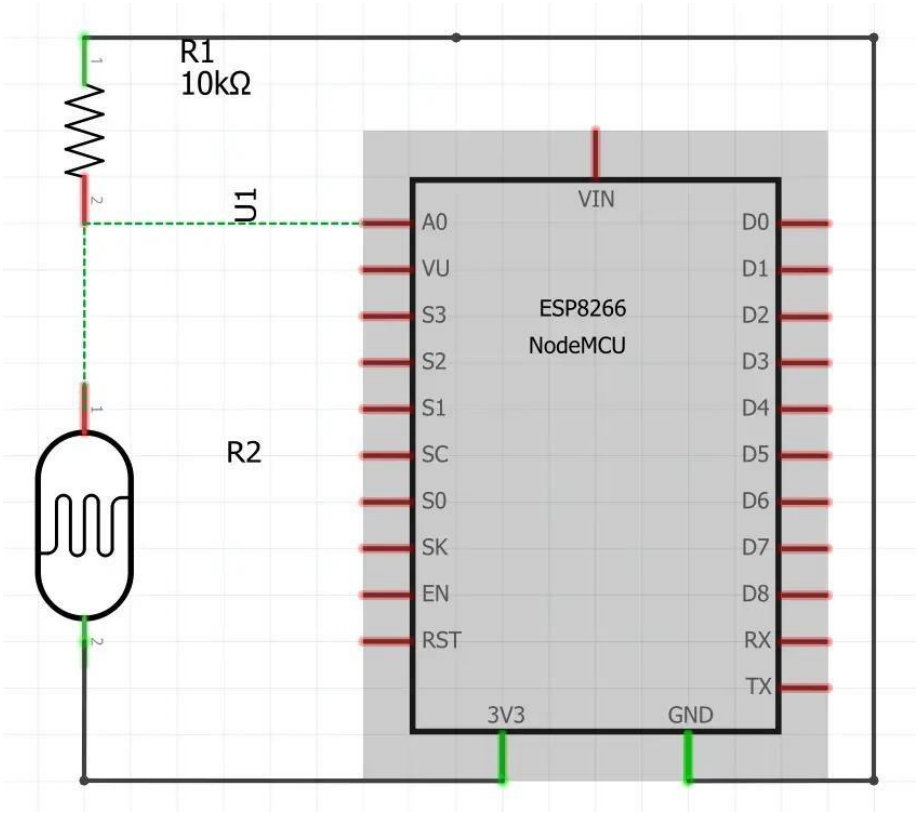
```
#define WIFI_SSID "GJC"
#define WIFI_PASSWORD "iforgott"#define DHTPIN 5
#define DHTTYPE     DHT11 DHT dht(DHTPIN, DHTTYPE);
String n;
 String m;
String o;
String p;
void setup()
{
Wire.begin(2,0);
delay(5000);
dht.begin();
 pinMode(D2,INPUT);
pinMode(D3,INPUT);
Serial.begin(115200);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("connecting");
while (WiFi.status() != WL_CONNECTED)
{
Serial.print(".");
delay(500);
}
Serial.println();
Serial.print("connected: ");
Serial.println(WiFi.localIP());
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
delay(2000);
}

void sensorUpdate()
{
float t = dht.readTemperature();
Firebase.set("TEMP",t);
 Serial.println(t);
float h = dht.readHumidity();
Firebase.set("HUMD",h);
Serial.println(h);
    if ( isnan(t))
    {
      Serial.println(F("Failed to read from DHT sensor!"));
      return;
    }
   }


  void loop()
  {
    sensorUpdate();
   if ((digitalRead(D2)==HIGH))
   {
```

```
  Firebase.set("LDR1","OFF");
 }
else
{
  Firebase.set("LDR1","ON");
}
}
```

BLOCK DIAGRAM LDR INTERFACING WITH NodeMCU



PROGRAM TO UPLOAD LDR DATA TO THINGSPEAK.COM

```
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>;
#include <WiFiClient.h>;
const char* ssid = "GJC";
const char* password = "iforgott";
WiFiClient client;
unsigned long myChannelNumber = 1013594;
const char * myWriteAPIKey = "UNDAT6YLR7NAMHTB";
void setup()
{
```

```
Serial.begin(115200);
delay(10);
WiFi.begin(ssid, password);
ThingSpeak.begin(client);
}
void loop()
{
int Value=analogRead(A0);
Serial.println(Value);
delay(100);
ThingSpeak.writeField(myChannelNumber,1,Value, myWriteAPIKey);
delay(100);
}
```

RESULT:

Sensor Data are successfully upload to Firebase and Thingspeak cloud .

EX.NO:9 DATE:	**Sensor and LDR sensor to Thingspeak cloud using Raspberry pi 4 controller**

Aim:

To Interface DHT11 with Raspberry pi and LDR interface with Raspberry pi 4 andupload data to Thingspeak cloud.

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	Raspberry pi 4		1
6	Micro B Type cable		1
7	Power jack		1
8	USB Cable		1
9	Jumper Wires		Required
10	DHT11		1
11	LDR		1

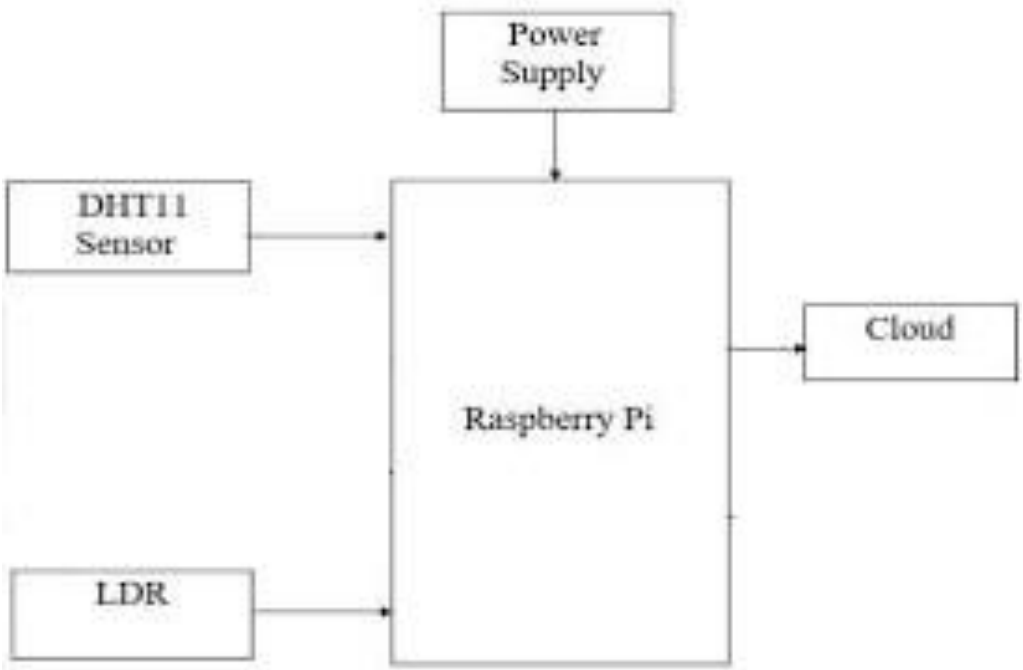
Hardware Procedure:

- **The DHT 11 has 4 Pins. Pin 1 is VCC, Pins 2 is Data, Pin 3 is NOT USED, Pin 4 isGround.**
- **Connect DHT 11 Pin 1 to 3.3v**
- **Connect DHT 11 Pin 2 to Raspberry PI Pin 16/GPIO 23 and connect a 4.7 or 10k resistorfrom DHT 11 Pin 2 to DHT Pin 1**
- **Connect DHT 11 Pin 4 to Ground**
- **The photo resistor has 2 pins**
- **Connect one pin to 3.3.v**
- **Connect the Other Pin to Raspberry Pi Pin 18/GPIO 24**
- **Connect a 1uF Capacitor to the same pin that the photo resistor is connected to onGPIO24. The Ground (White Stripe) side of the capacitor should go to Ground.**

Software Procedure:

- Click on Thonny
- Click on file
- Click on New
- Write a Program as per circuit Pin connections
- Click on Save
- Click on Verify
- Click on Upload the code into RP 4 by using USB cable.
- Create Channel in Thingspeak.com
- And Monitor the data uploaded in cloud

BLOCK DIAGRAM DHT11 and LDR INTERFACING WITH Raspberry pi - 4



Program Code:

```
import sys
import RPi.GPIO as GPIO
import os
from time import sleep
import Adafruit_DHT
import urllib2



 DEBUG = 1
 # Setup
 the pins
 we are
 connect to
 RCpin =
 24
 DHTpin = 23

 #Setup our API and delay
 myAPI = "***Insert Your API
 CODE HERE***" myDelay =
 15 #how many seconds
 between posting data

 GPIO.setmode(GPIO.BCM)
 GPIO.setup(RCpin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)



 def getSensorData():
    RHW, TW = Adafruit_DHT.read_retry(Adafruit_DHT.DHT11, DHTpin)

    #Convert
    from
    Celius to
    Farenheit
    TWF =
    9/5*TW+
    32
```

```
    # return dict
    return (str(RHW), str(TW),str(TWF))

 def RCtime(RCpin):
    LT = 0

    if (GPIO.input(RCpin) == True):
       LT += 1
    return (str(LT))

# main() functiondef main():

print 'starting...'baseURL """
```

RESULT:

DHT11 Sensor Data is successfully uploaded to Thingspeak cloud .