Adhiparasakthi College of Engineering

G.B.Nagar , Kalavai – 632 506

LAB MANUAL

Regulation	:*2021*
Branch	: *B.Tech. –IT*
Year&Semester	: III Year / V Semester

CS3691 – EMBEDDED SYSTEMS AND IOT LAB

ANNA UNIVERSITY CHENNAI

Regulation 2021

CS3691 - EMBEDDED SYSTEMS AND IOT

SYLLABUS

LIST OF EXPERIMENTS

8051 Programs using EDSIM51

1. Write 8051 Assembly Language experiments using simulator.
2. Test data transfer between registers and memory.
3. Perform ALU operations.

8051grams using Embedded C

1. Write Basic and arithmetic Programs Using Embedded

IOT using Arduino and Raspberry Pi

1. Introduction to Arduino platform and programming
2. Explore different communication methods with IoT devices (Zigbee, GSM, Bluetooth)
3. Introduction to Raspberry PI platform and python programming
4. Interfacing sensors with Raspberry PI
5. Communicate between Arduino and Raspberry PI using any wireless medium
6. Setup a cloud platform to log the data
7. Log Data using Raspberry PI and upload to the cloud platform
8. Design an IOT based system

1. Write 8051 Assembly Language experiments using simulator. 2. Test data transfer between registers and memory. 3. Perform ALU operations. 4. Write Basic and arithmetic Programs Using Embedded C. 5. Introduction to Arduino platform and programming 6. Explore different communication methods with IoT devices (Zigbee, GSM, Bluetooth) 7. Introduction to Raspberry PI platform and python programming 8. Interfacing sensors with Raspberry PI 9. Communicate between Arduino and Raspberry PI using any wireless medium 10. Setup a cloud platform to log the data 11. Log Data using Raspberry PI and upload to the cloud platform 12. Design an IOT based system

TOTAL: 30 PERIODS

INDEX

EX. NO : 01

DATE :

8 BIT ADDITION USING ARITHMETIC OPERATION 8051 MICROCONTROLLER (USING SIMULATOR)

AIM:

To write an ALP program to add, Subtract, multiply and divide two 8-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 theprogram.

Subtraction program ALGORITHM:

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

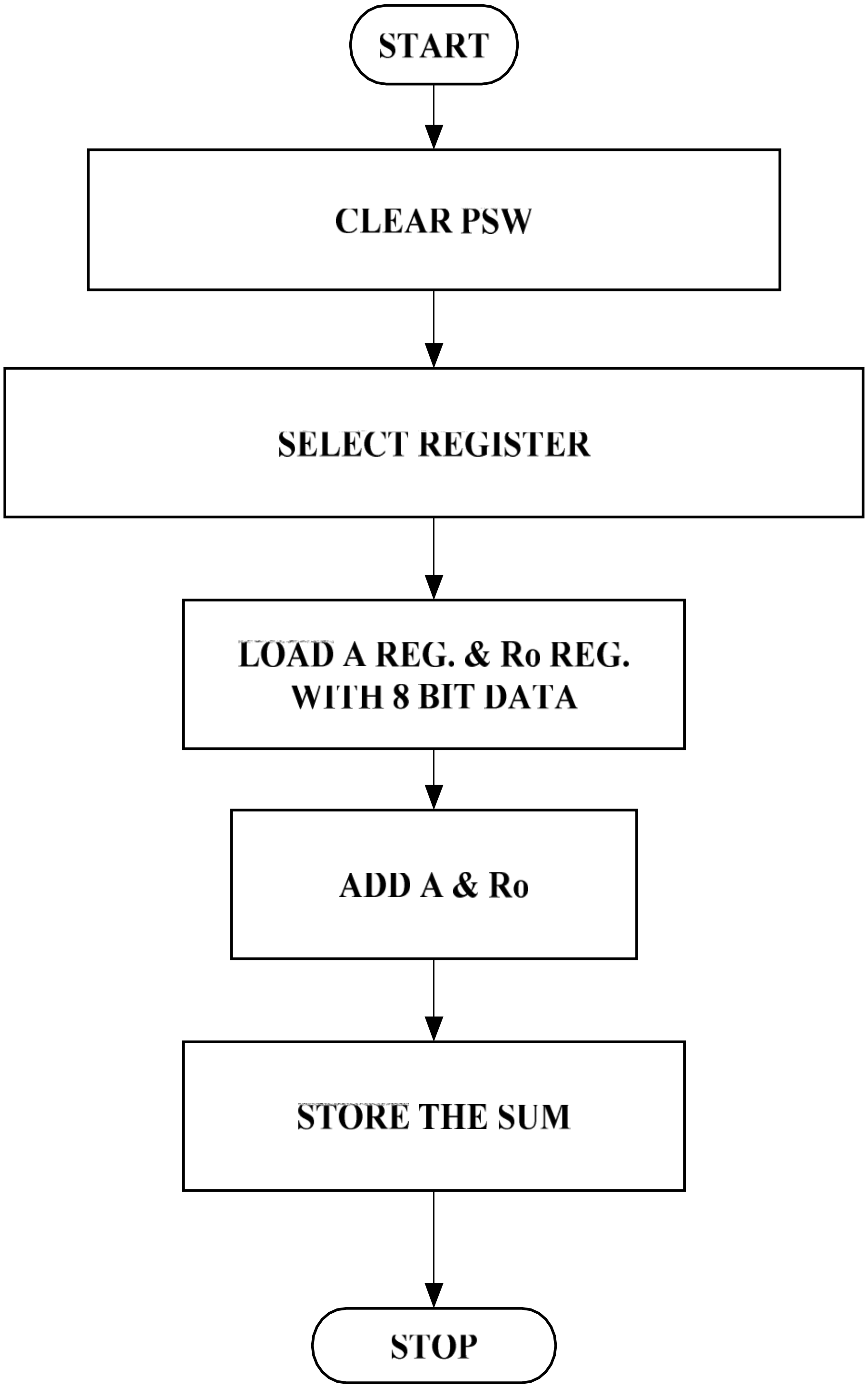
Multiplication program ALGORITHM:

➢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 theprogram.

Division program ALGORITHM:

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

FLOW CHART


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

MEMORY LOCATION	OPCODES	LABEL	PROGRAM	COMMENDS
Addition program				
0000	74, 07		***MOV A, #07***	Move data to Acc
0002	75 ,F0, 03		***MOV B, #03***	Move data to B Reg
0005	25, F0		***ADD A,B***	Adding Acc with B Reg.
0007	F8		***MOV R0,A***	Move result to R0 Reg.
0008	80, FE	***HERE:***	***SJMP HERE***	Short jump here
Subtraction program				
0000	74, 07		***MOV A, #07***	Move data to Acc
0002	75, F0, 03		***MOV B, #03***	Move data to B Reg
0005	95, F0		***SUBB A,B***	Subtract Acc with B Reg.
0007	F9		***MOV R1,A***	Move result to R1 Reg.
0008	80, FE	***HERE:***	***SJMP HERE***	Short jump here
Multiplication program				
0000	74, FF		***MOV A,#0FFH***	Move data to Acc
0002	75, F0, 03		***MOV B,#03***	Move data to B Reg
0005	A4		***MUL AB***	Multiplying Acc wit B Reg.
0006	FB		***MOV R3,A***	Move Result to R3 Reg.
0007	AC, F0		***MOV R4,B***	Move Carry to R4 Reg.
0009	80, FE	***HERE:***	***SJMP HERE***	Short jump here

MEMORY LOCATION	OPCODES	LABEL	PROGRAM	COMMENDS
Division program				
0000	74, 0D		**MOV A,#13**	Move data to Acc
0002	75, F0, 02		**MOV B,#03**	Move data to B Reg.
0005	84		**DIV AB**	Divide Acc with B Reg.
0006	FB		**MOV R3,A**	DPTR = 4500
0007	AC, F0		**MOV R4,B**	Store result in 4500 memory location
0009	80, FE	**HERE:**	**SJMP HERE**	DPTR = DPTR+1

OUTPUT:

INPUT		OUTPUT	
ADDITION			
ACC		R0	
B Reg			
SUBTRACTION			
ACC		R1	
B Reg			
MULTIPLICATION			
ACC		R3	
B Reg		R4	
DIVISION			
ACC		R3	
B Reg		R4	

RESULT:

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

EX. NO : 2 DATE :

LOGICAL OPERATIONS and 2’s Complement

USING8051 MICROCONTROLLER

AIM:

To perform logical operation using 8051 microcontroller AND, OR & EX-OR.

ALGORITHM:

- ➢ Get the input value and store data in theaccumulator.
- ➢ Get the second values and store the Bregister.
- ➢ Logical operation to perform the givennumber
- ➢ Store the output value inmemory.

MEMORY LOCATION	OPCODES	LABEL	PROGRAM	COMMENDS
AND Operation program				
0000	C3		*CLR C*	Clear the carry
0001	74,07		*MOV A, #07*	Move data to Acc
0003	54,03		*ANL A, #03*	AND Acc with immediate
0005	F8		*MOV R0, A*	Move result to R0 Reg
0006	80,FE	*HERE:*	*SJMP HERE*	Short jump here
OR Operation program				
0000	C3		*CLR C*	Clear the carry
0001	74,07		*MOV A, #07*	Move data to Acc
0003	44,03		*ORL A, #03*	OR Acc with immediate
0005	F4		*MOV R1, A*	Move result to R1 Reg
0006	80,FE	*HERE:*	*SJMP HERE*	Short jump here
XOR OPERATION PROGRAM				
0000	C3		*CLR C*	Clear the carry
0001	74,07		*MOV A, #07*	Move data to Acc
0003	64,03		*XRL A, #03*	XOR Acc with immediate data
0005	FA		*MOV R2, A*	Move result to R2 Reg
0006	80,FE	*HERE:*	*SJMP HERE*	Short jump here
2's COMPLEMENT PROGRAM				
0000	C3		*CLR C*	Clear the carry
0001	74,07		*MOV A, #07*	Move data to Acc
0003	F4		*CPL A*	Complement Accumulator
0004	04		*INC A*	A = A+1
0005	FB		*MOV R3,A*	Move result to R3 Reg
0006	80,FE	*HERE:*	*SJMP HERE*	Short jump here

INPUT		OUTPUT	
ADD			
DATA1		R0	
DATA 2			
OR			
DATA1		R1	
DATA 2			
XOR			
DATA1		R2	
DATA 2			
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 : 03

PDATE :

8 BIT ARITHMETIC OPERATION USING 8051 MICROCONTROLLER C PROGRAMMING (Using Keil uVision5)

Write Basic and arithmetic Programs 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-bitdata to a variable x.
➢Assign another desired 8-bitdata to another variable y.
➢Add two 8-bitnumbers and store in another variable z.
➢Store the result in Port 0

Subtraction program ALGORITHM:

➢Assign any desired 8-bitdata to a variable a.
➢Assign another desired 8-bitdata to another variable b.
➢Subtract two 8-bitnumbers and store in another variable c.
➢Store the result in Port 1

Multiplication program ALGORITHM:

➢Assign any desired 8-bitdata to a variable d.
➢Assign another desired 8-bitdata to another variable e.
➢Multiply two 8-bitnumbers and store in another variable f.
➢Store the result in Port 2

Division program ALGORITHM:

➢Assign any desired 8-bitdata to a variable p.
➢Assign another desired 8-bitdata to another variable q.
➢Divide two 8-bitnumbers and store in another variable r.
➢Store the result in Port 3
➢Stop theprogram.

```
# 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; //second 8-bit number

P0=0x00; //declare port 0 as output port

z=x+y; // perform addition

P0=z; //display result on port 0

 //subtraction
a=0x03; //first 8-bit number

b=0x04; //second 8-bit number

P1=0x00; //declare port 1 as output port

c=b-a; // perform subtraction

P1=c; //display result on port 1

 //multiplication
d=0x03; //first 8-bit number

e=0x04; //second 8-bit number

P2=0x00; //declare port 2 as output port

f=e*d; // perform multiplication

P2=f; //display result on port 2

 //division
p=0x03; //first 8-bit number

q=0x04; //second 8-bit number

P3=0x00; //declare port 3 as output port

r=q/p; // perform division

P3=r; //display result on port 3

 while(1);

}
```

RESULT:

INPUT		OUTPUT	
ADDITION			
DATA1		PORT 0	
DATA 2			
SUBTRACTION			
DATA1		PORT 1	
DATA 2			
MULTIPLICATION			
DATA1		PORT 2	
DATA 2			
DIVISION			
DATA1		PORT 3	
DATA 2			

RESULT:

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

EX. NO : 04

DATE :

ARDUINO PROGRAMMING (LED BLINKING AND ANALOG READD)

EXP 4a

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.

```
const int led = 2;
void setup()
{
 pinMode(led, OUTPUT);
}
void loop()
 {
digitalWrite(led, HIGH);
delay(1000);
 digitalWrite(led, LOW);
delay(1000);
}
```

RESULT: LED is successfully controlled by Arduino microcontroller Board.

EXP 4b

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	12V Adaptor		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:

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:

```
#define LED 11

#define LD 12

#define POT A0

void setup()

{

pinMode(LED, OUTPUT);

pinMode(LD,OUTPUT);

pinMode(POT, INPUT);

}

void loop()

{

int x = analogRead(POT);

if(x >= 512)

{

 digitalWrite(LED,HIGH);

digitalWrite(LD,LOW);

}

else

{

digitalWrite(LED,LOW);

digitalWrite(LD,HIGH);

}
}
```

RESULT: LED is successfully controlled by Arduino microcontroller Board.

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

EXP 5a

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

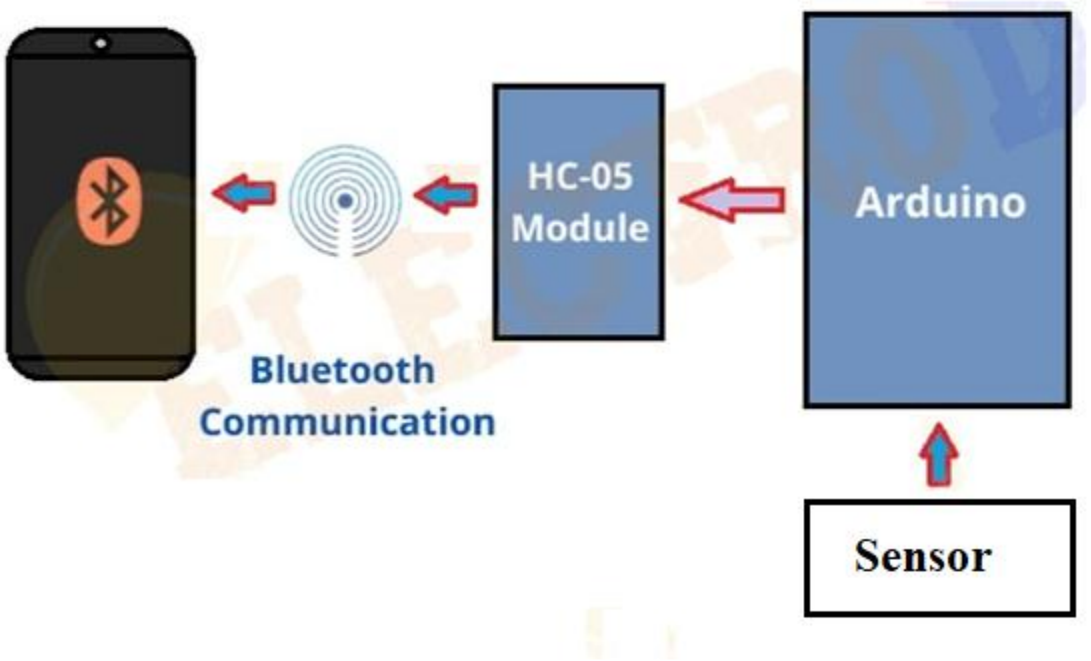
Hardware Procedure:

- 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:

8. Click on Arduino IDE
9. Click on file
10. Click on New
11. Write a Program as per circuit Pin connections
12. Click on Save
13. Click on Verify
14. Click on Upload the code into Arduino Uno by using USB cable.

BLOCK DIAGRAM BLUETOOTH INTERFACING

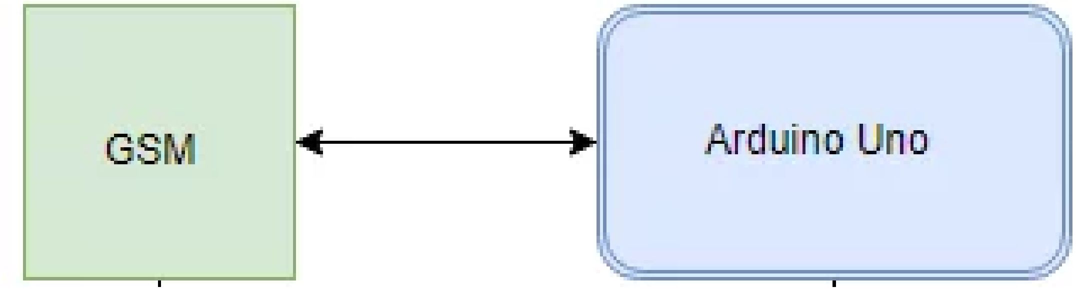


Program:

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

```
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 Mode
  delay(100);
  Serial.println((char)13);// ASCII code of enter
   delay(1000);
  Serial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(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 Mode
  delay(100);
 Serial.println((char)13);// ASCII code of enter
   delay(1000);
  Serial.println("AT+CMGS=\"+919994085790\"\r"); // Replace x with mobile number
  delay(1000);
  Serial.println("CS 3691 – EMBEDDED SYSTEMS AND IOT LAB");// The SMS text
you want to send
  delay(100);
    //mySerial.println("ATD+60XXXXXXXXX;");
  Serial.println((char)26);// ASCII code of CTRL+Z
  delay(5000);
  Serial.println("ATD+919994085790;"); // Replace x with mobile number
    delay(1000);
}

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

RESULT:

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

EXP 6

19/09/2023

Introduction to Raspberry PI platform and python programming and Interfacing LM35 Temperature sensors with Raspberry PI

Aim: To Interface LED with Raspberry pi RP2040 and LM35 (or) LDR interface with Raspberry 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

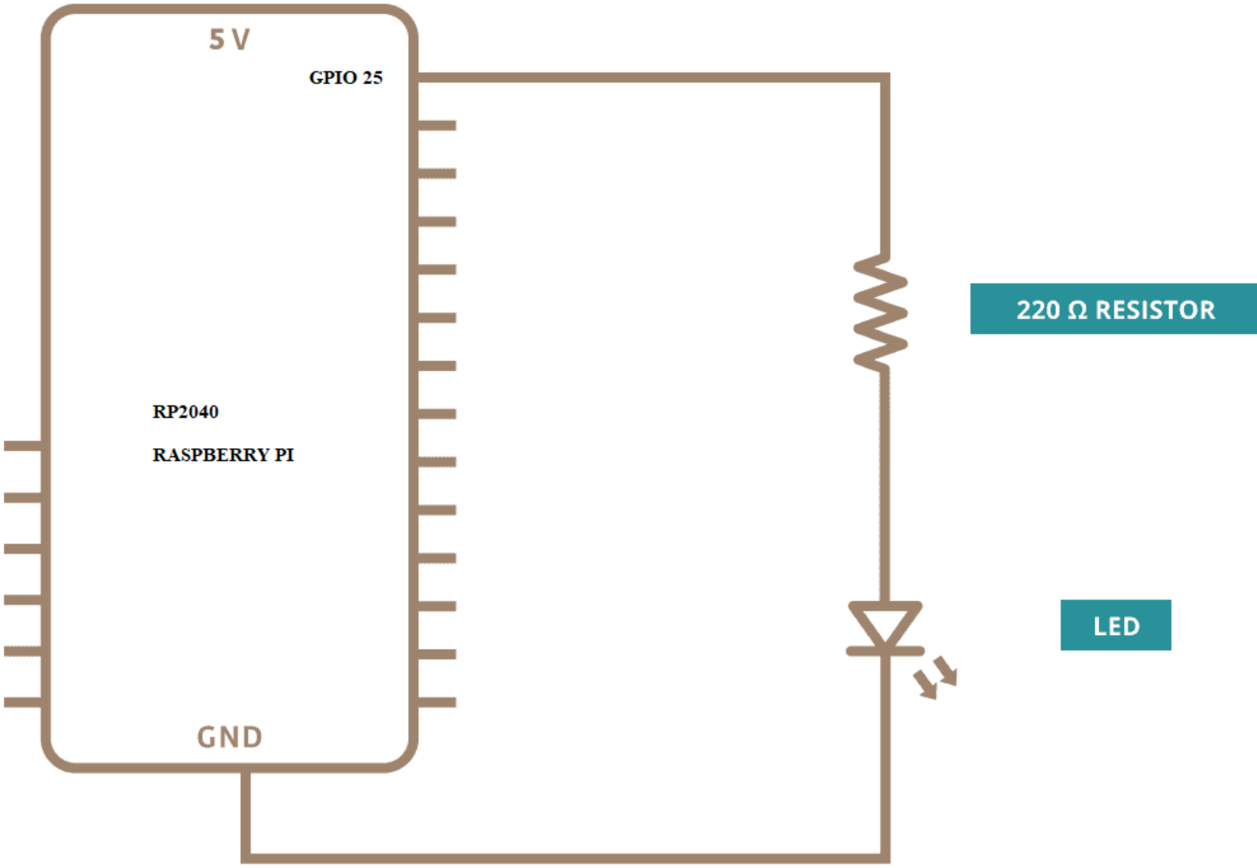
Hardware Procedure:

- 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

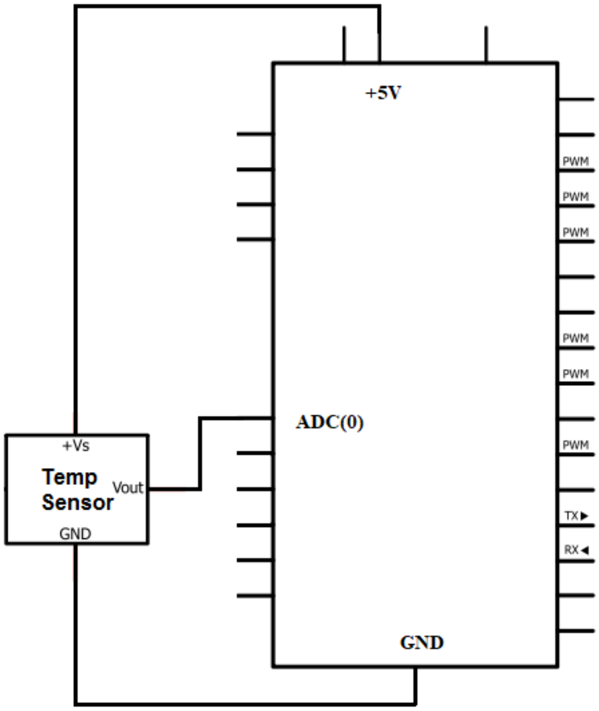


Program:

```
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_factor
   temperature = 27 - (reading - 0.706)/0.001721
   print("Temperature: {}".format(temperature))
   utime.sleep(2)
```

RESULT: LED is successfully controlled by RP2040 and Analog LM35 Value (Sensors data) are successfully measured by RP2040.

EXP 7

19/09/2023 Setup a cloud platform and upload the Temperature and Humidity using DHT11 Sensor and LDR Sensor to Thingspeak cloud and Firebase Cloud

Aim: To Interface DHT11 and LDR interface with NodeMCU and upload data to Thingspeak cloud and Firebase Console.

Apparatus:

S. No.	Apparatus	Range/Rating	Quantity
1	Universal Board		1
2	NodeMCU		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:

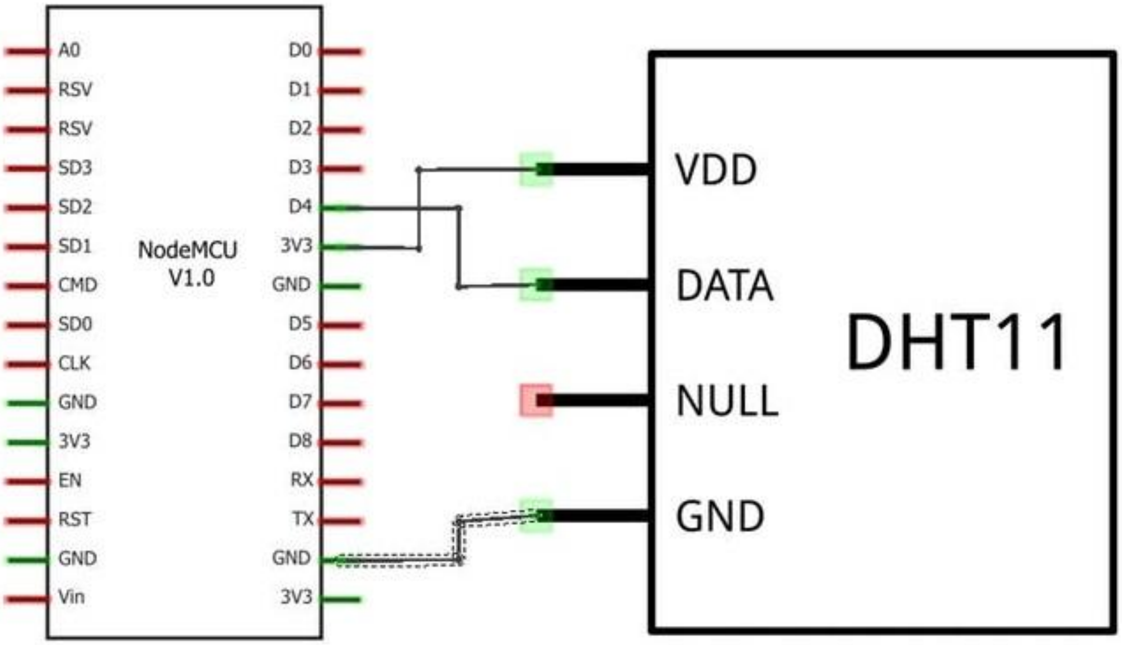
- **The DHT 11 has 4 Pins. Pin 1 is VCC, Pins 2 is Data, Pin 3 is NOT USED, Pin 4 is Ground.**
- **Connect DHT 11 Pin 1 to 3.3v**
- **Connect DHT 11 Pin 2 to Raspberry PI Pin 16/GPIO 23 and connect a 4.7 or 10k resistor from DHT 11 Pin 2 to DHT Pin 1**
- **Connect DHT 11 Pin 4 to Ground**
- **The photo 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



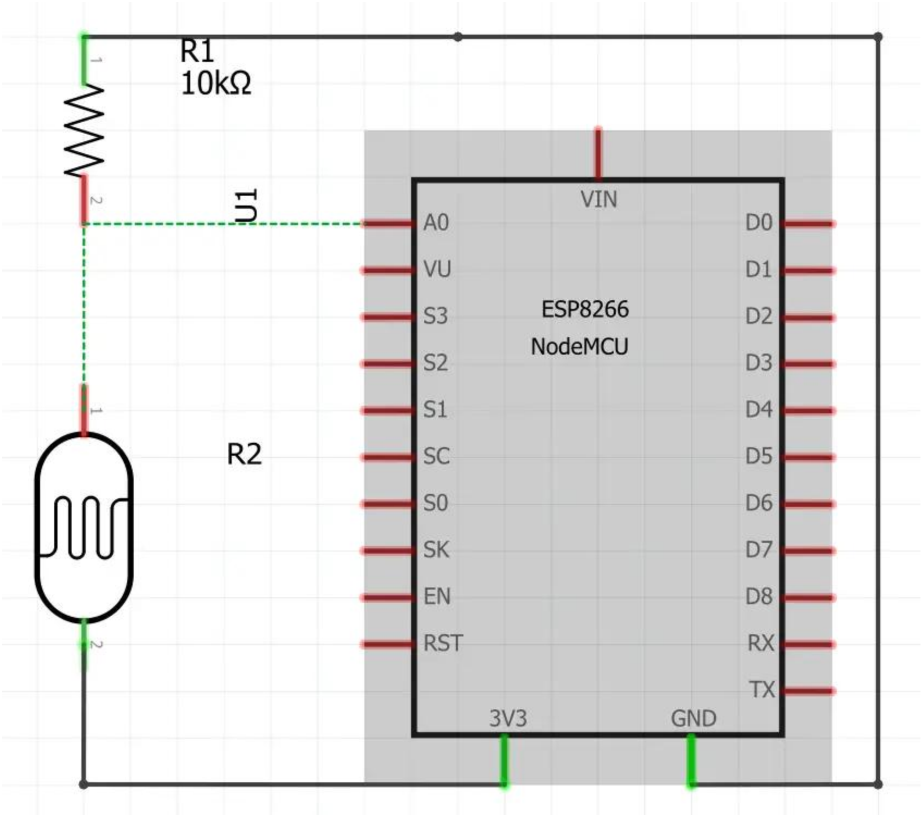
PROGRAM TO UPLOAD TEMPERATURE AND HUMIDITY TO FIREBASE CONSOLE:

```
#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

EXP 8

09/10/2023

Setup a cloud platform and upload the Temperature and Humidity using DHT11 Sensor and LDR Sensor to Thingspeak cloud using Raspberryi pi 4 controller

Aim: To Interface DHT11 with Raspberry pi and LDR interface with Raspberry pi 4 and upload 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:

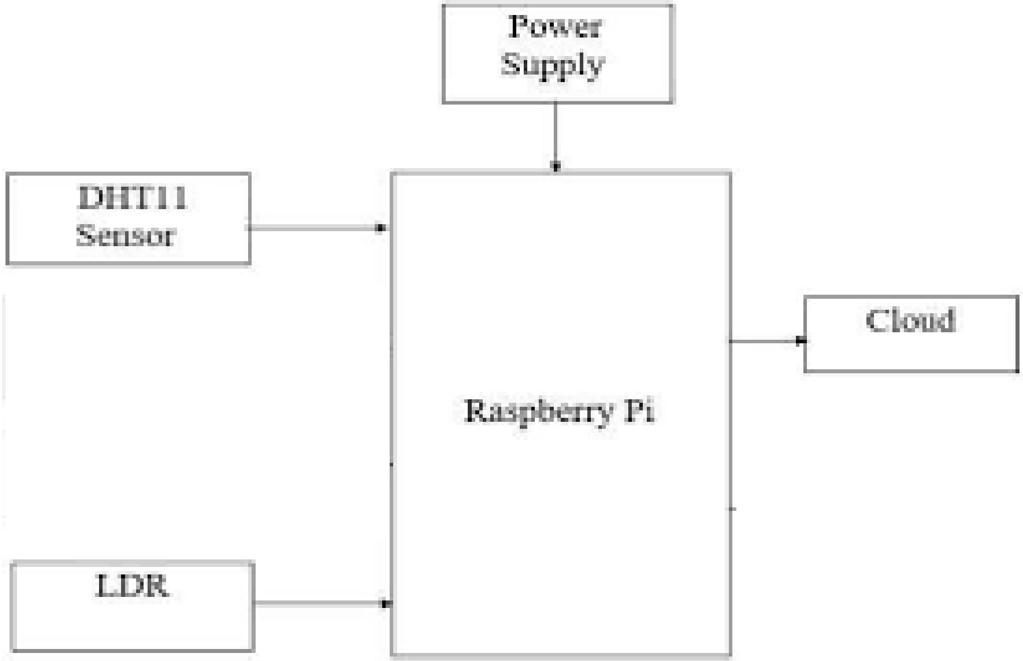
- **The DHT 11 has 4 Pins. Pin 1 is VCC, Pins 2 is Data, Pin 3 is NOT USED, Pin 4 is Ground.**
- **Connect DHT 11 Pin 1 to 3.3v**
- **Connect DHT 11 Pin 2 to Raspberry PI Pin 16/GPIO 23 and connect a 4.7 or 10k resistor from DHT 11 Pin 2 to DHT Pin 1**
- **Connect DHT 11 Pin 4 to Ground**
- **The photo 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 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() function
def main():

    print 'starting...'

    baseURL """
```

RESULT: DHT11 Sensor Data is successfully uploaded to Thingspeak cloud .

[Type here]