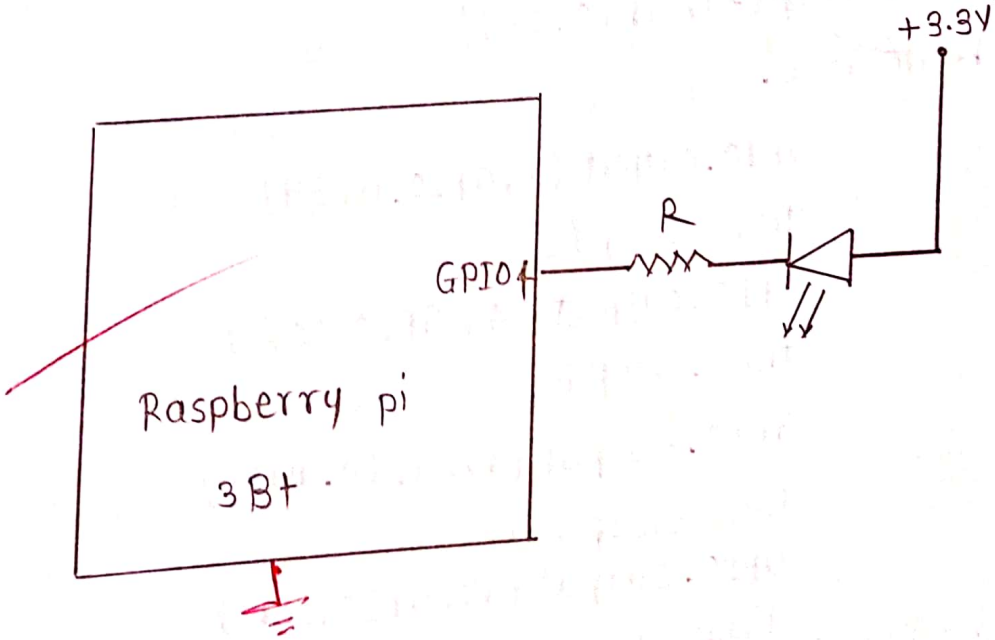
Expt. (1) Title : Programming of Raspberry pi to control LED's attached to the GPIO pins.

AIM : To study and understand interfacing LED array Raspberry pi

OBJECTIVES : ① To understand the control of LED interfacing.
② To study interfacing of LED array to Raspberry pi.

CIRCUIT DIAGRAM :



(•) Program =

(1) For 1 LED =

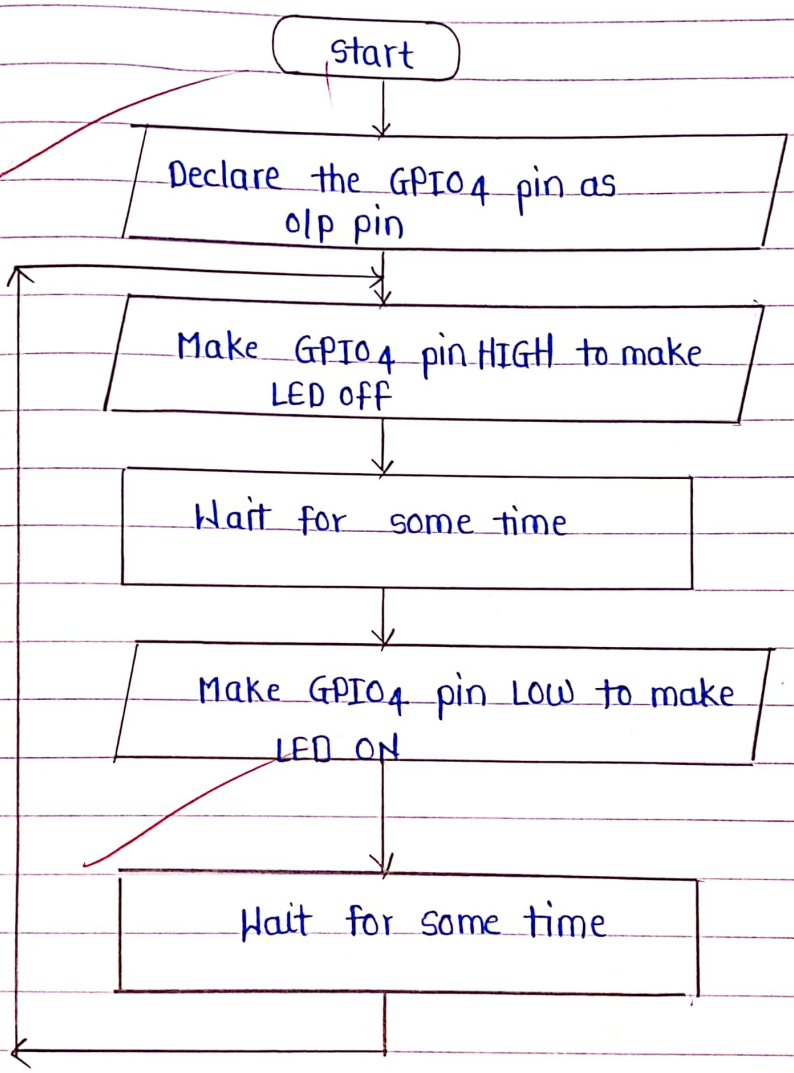
```
import Rpi.GPIO as GPIO
import time
GPIO.setwarning(false)
GPIO.setmode(GPIO.BCM)
GPIO.setup(4,GPIO.OUT)
while True:
        GPIO.output(4,GPIO.HIGH)
        time.sleep(3)
        GPIO.output(4,GPIO.LOW)
        time.sleep(3)
```

(2) For 4 LED Blinking =

```
import Rpi.GPIO as GPIO
import time
GPIO.setwarnings(false)
GPIO.setmode(GPIO.BCM)
GPIO.setup(4,GPIO.OUT)
GPIO.setup(17,GPIO.OUT)
GPIO.setup(18,GPIO.OUT)
GPIO.setup(27,GPIO.OUT)
While True:
        GPIO.output(4,GPIO.HIGH)
        time.sleep(3)
        GPIO.output(4,GPIO.LOW)
        time.sleep(3)
        GPIO.output(17,GPIO.HIGH)
        time.sleep(3)
        GPIO.output(17,GPIO.LOW)
        time.sleep(3)
        GPIO.output(18,GPIO.HIGH)
        time.sleep(3)
```

```
GPIO.output(18, GPIO.LOW)
time.sleep(3)
GPIO.output(27, GPIO.HIGH)
time.sleep(3)
GPIO.output(27, GPIO.LOW)
time.sleep(3)
```

(•) Flow chart=



(•) Result=

Interfacing ang programming of Raspberry pi to control LED is succesfully studied, tested and observe the o/p.

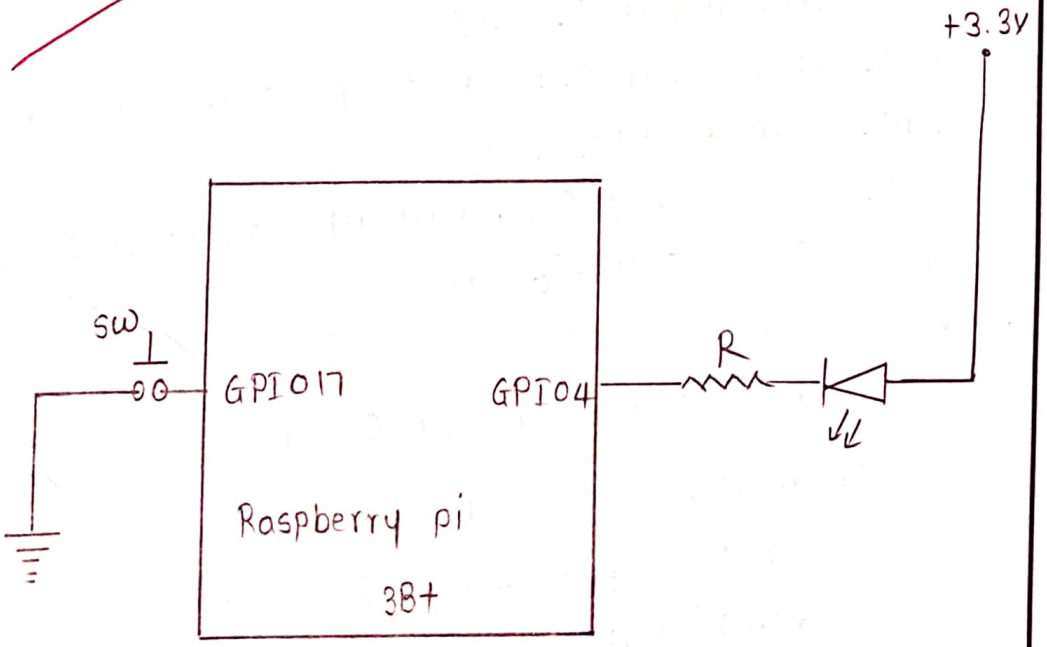
Expt. (2)

Tile : Programming of Raspberry pi to get feedback from a switch connected to file GPIO pins.

AIM : To study and understand interfacing switch connected to the GPIO pins of Raspberry pi.

OBJECTIVES : ① To understand the concept of switch interfacing
② To study interfacing of switch to Raspberry pi.

CIRCUIT DIAGRAM :



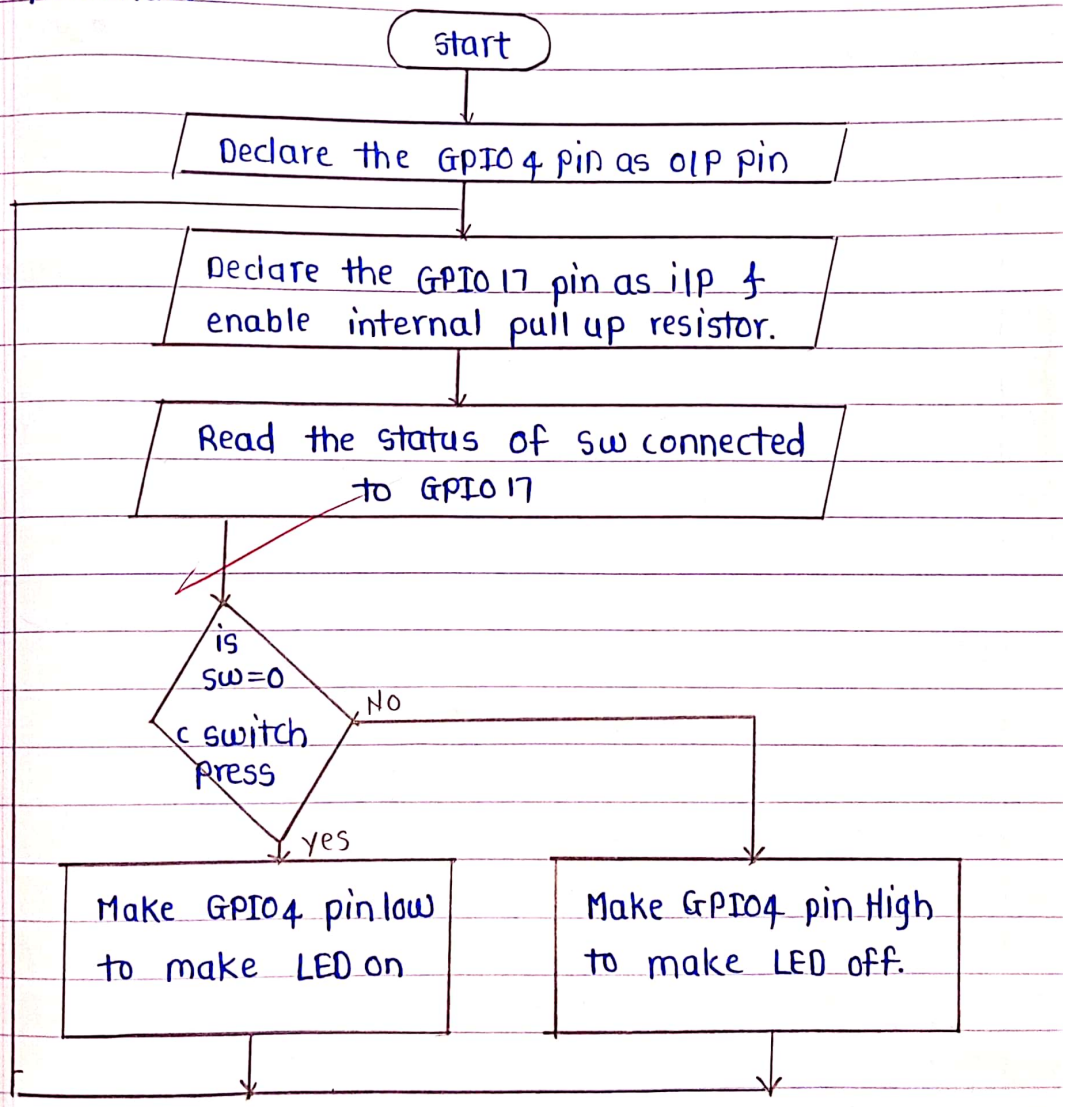
(•) Program =

(1)
```
import Rpi.GPIO as GPIO
import time
GPIO.set warnings (false)
GPIO.setmode (GPIO.BCM)
GPIO.setup (17, GPIO.In, pull-up-down = GPIO.PUD_up)
GPIO.setup (4, GPIO.OUT)
while True:
    button_state = GPIO.input(17)
    if button_state == False:
        GPIO.output (4, false)
        print ('Button pressed ....')
        while GPIO.input (17) == false
            time.sleep(0.2)
    else
        GPIO.output (4, True)
```

(2)
```
import Rpi GPIO as GPIO
import time
GPIO.setwarnings (false)
GPIO.setmode (GPIO.BCM)
GPIO.setup(17, GPIO.In, pull-up-down = GPIO.PUD_UP)
GPIO.setup(4, GPIO.OUT)
GPIO.setup(27, GPIO.In, Pull-up_down = GPIO.PUD_up)
GPIO.set up(18, GPIO.OUT)
While True:
    buttonon.state = GPIO.input(n)
If button_state == False:
    GPIO.output(4, False)
    Print('Button pressed...')
while GPIO.input (17) == false:
    time.sleep (0.2)
```

```
else
        GPIO.output(4,True)
        button_state = GPIO.input(27)
    if button_state == false:
            GPIO.output(18,false)
            print('Button pressed')
            while GPIO.input(27) == false:
             time.sleep(0.2)
      else
            GPIO.output(18,True)
```

(•) Flow chart =



Expt. (9)

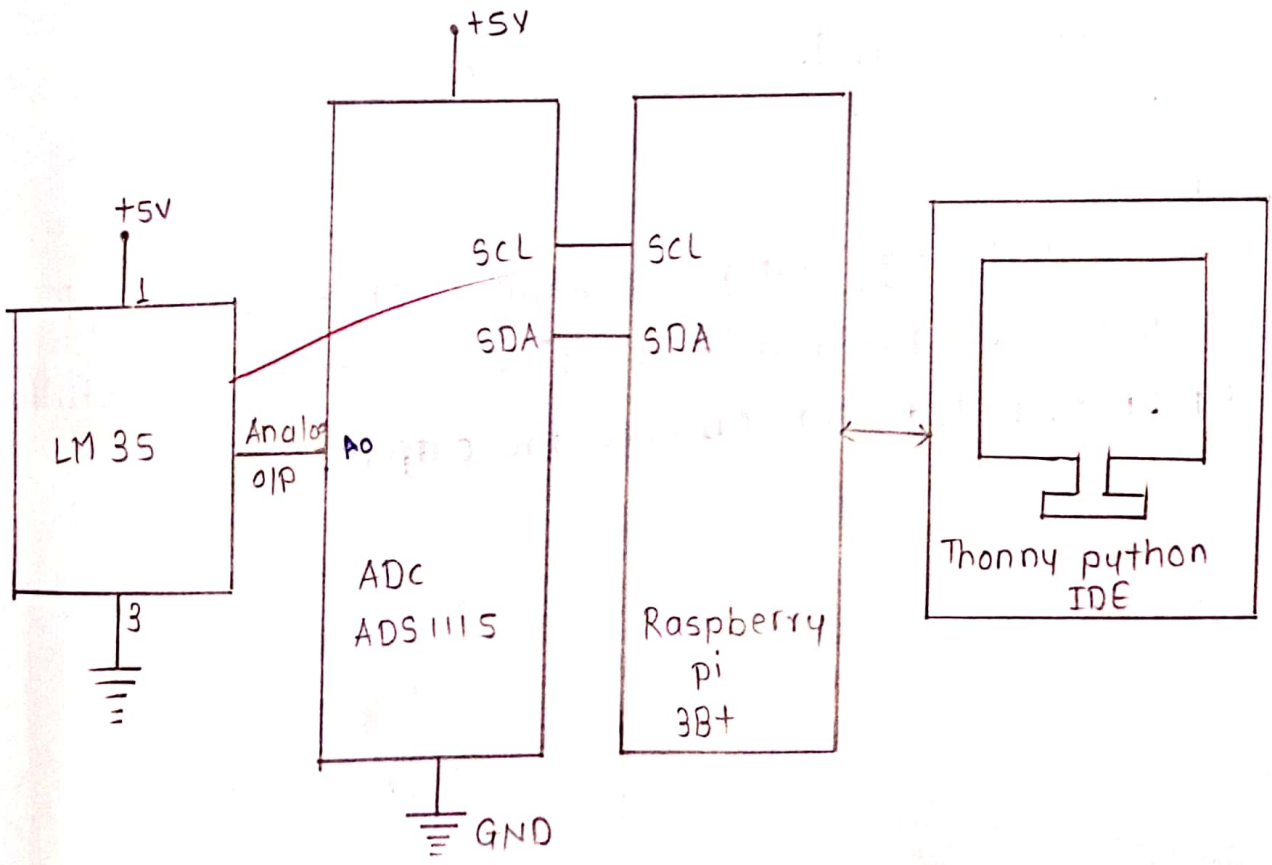
Title : Programming of Raspberry Pi to detect using temp sensor

AIM : To study and understanding the working principle of LM35 sensor to Raspberry pi.

OBJECTIVES :

1. To understand the working principle of LM-35 sensor.
2. To study interfacing of temp sensor (LM35) to Raspberry pi.

CIRCUIT DIAGRAM :



(•) Program =

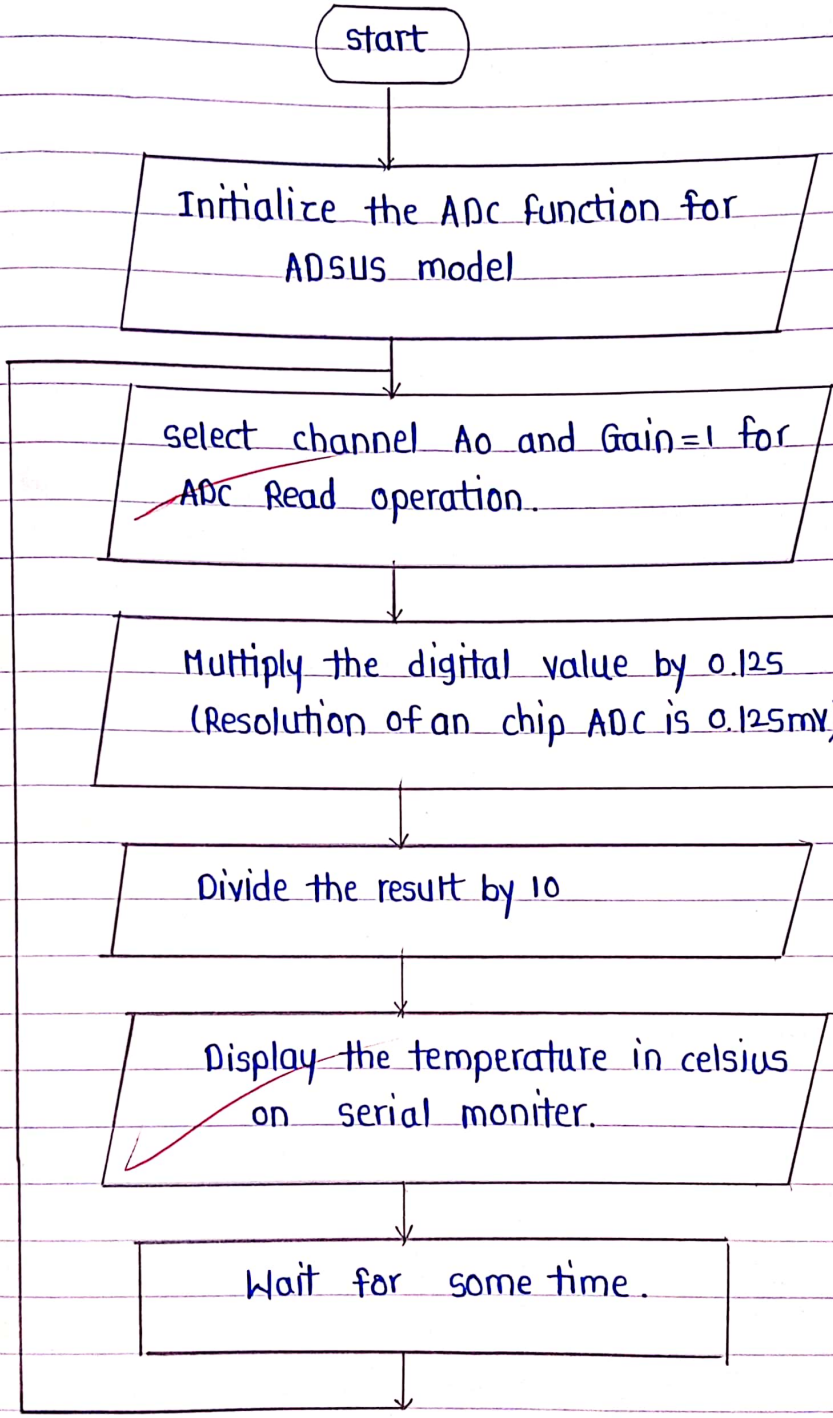
```
import time
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1
While True:
        values = [0]*1
        for i in range (1):
        values[i] = adc read.adc (i, gain = GAIN)
        temp = values[i]* 0.125
        temp = temp/10
        print ('Analog o/p = {0:>6}', format (*values)
        print ('Temp in c = {0:>6}' format (temp))
        time.sleep (2)
```

- Application:
  (1) Medical
  (2) Motorsport
  (3) HV Ac
  (4) Agriculture
  (5) Industrial.

- Result =

Interfacing and programming of Raspberry pi to detect temperature from temp sensor is succesfull studied, tested and observe the output.

Flowchart =



Expt. (4)

Tile : Programming of Raspberry pi to detect high intensity using photo sensor.

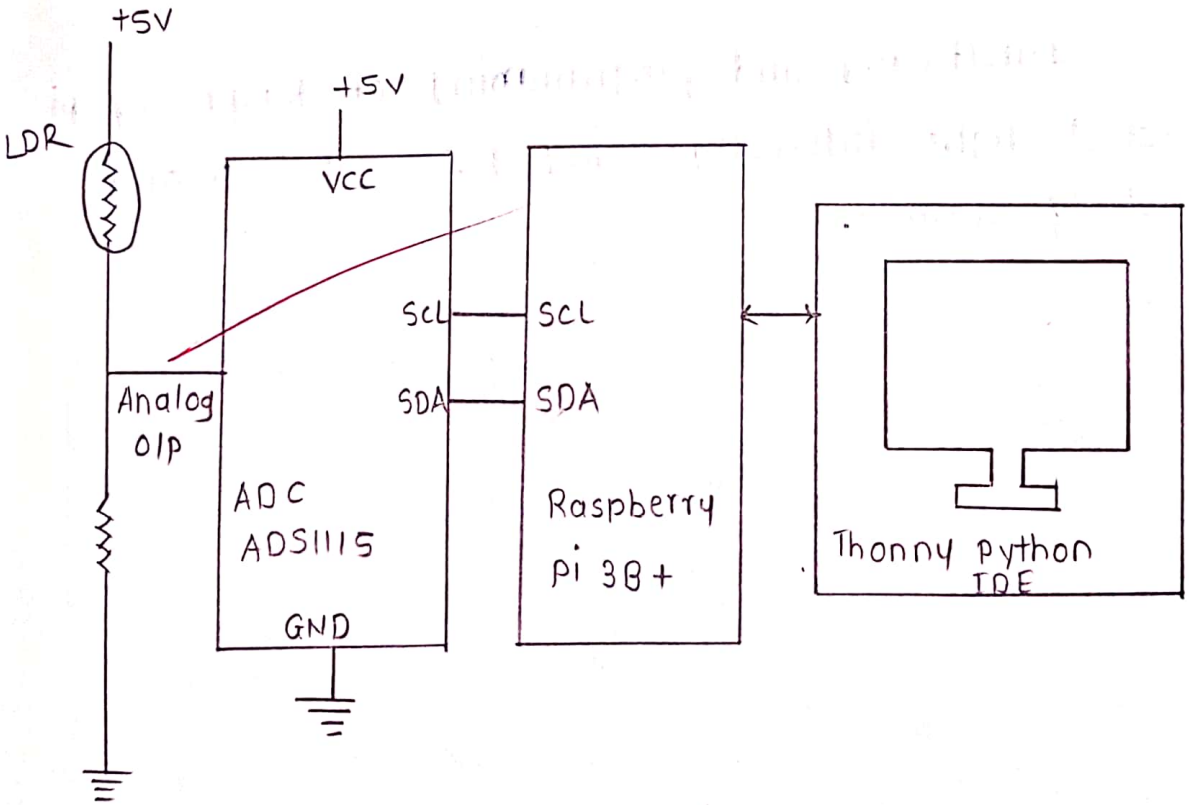
AIM : To study and understand interfacing photocell sensor to Raspberry pi.

OBJECTIVES :

① To understand the working principle of photocell sensor.

② To study interfacing of photocell sensor (LDR) to the Raspberry pi.

CIRCUIT DIAGRAM :



(•) Program =

```
import time
import Adafruit_ADS1x15
adc = Ada fruit_ADS1x15.ADS1115()
GAIN = 1
While True:
 values = [0]*1
 for i in range (1);
 Values[i] = adc.read_adc (i, gain = GAIN)
 if values[i] > 3000;
 print ('Dark condition')
 else
     print ('illuminated condition...')
 time.sleep(2)
```

- Applications =

① To count the vehicles on the road
② solar energy system such as solar panel
③ Medical applications.

- Result =

Interfacing and programming of Raspberry pi to detect light intensity using photocell sensor is succesfully studied.

Expt. (5)

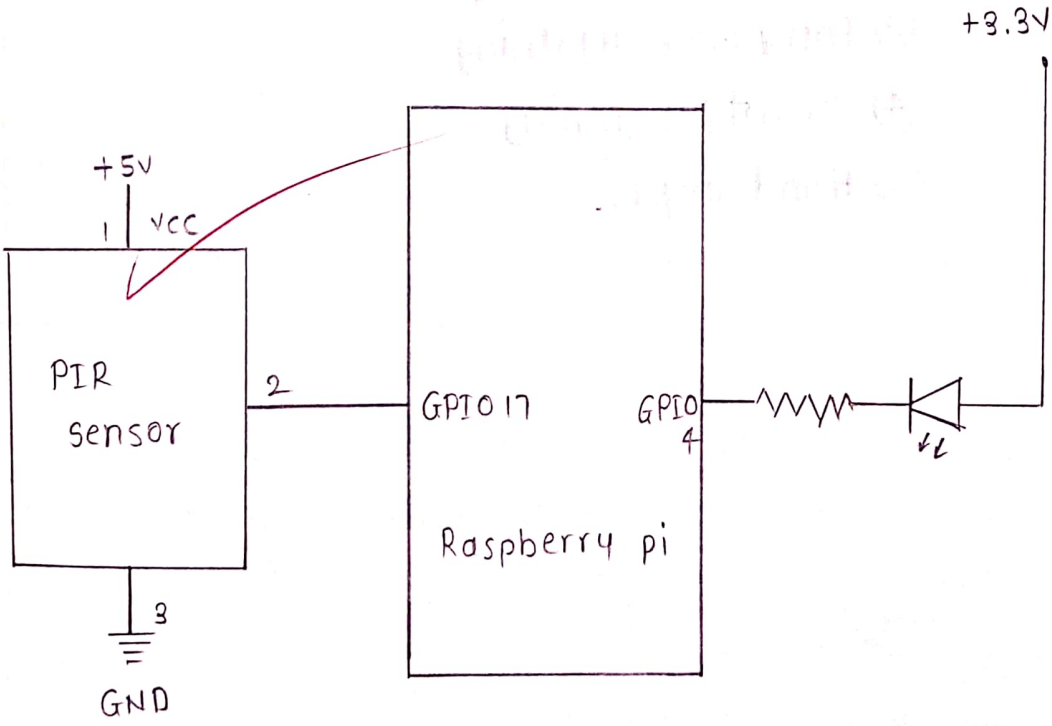
Tile : Programming of Raspberry Pi for motion detection.

AIM : Programming to study and understand of Raspberry pi for motion detection.

OBJECTIVES : ① To understand working principle of PIR sensor.

② To study interfacing of PIR sensor to the Raspberry pi.

CIRCUIT DIAGRAM :



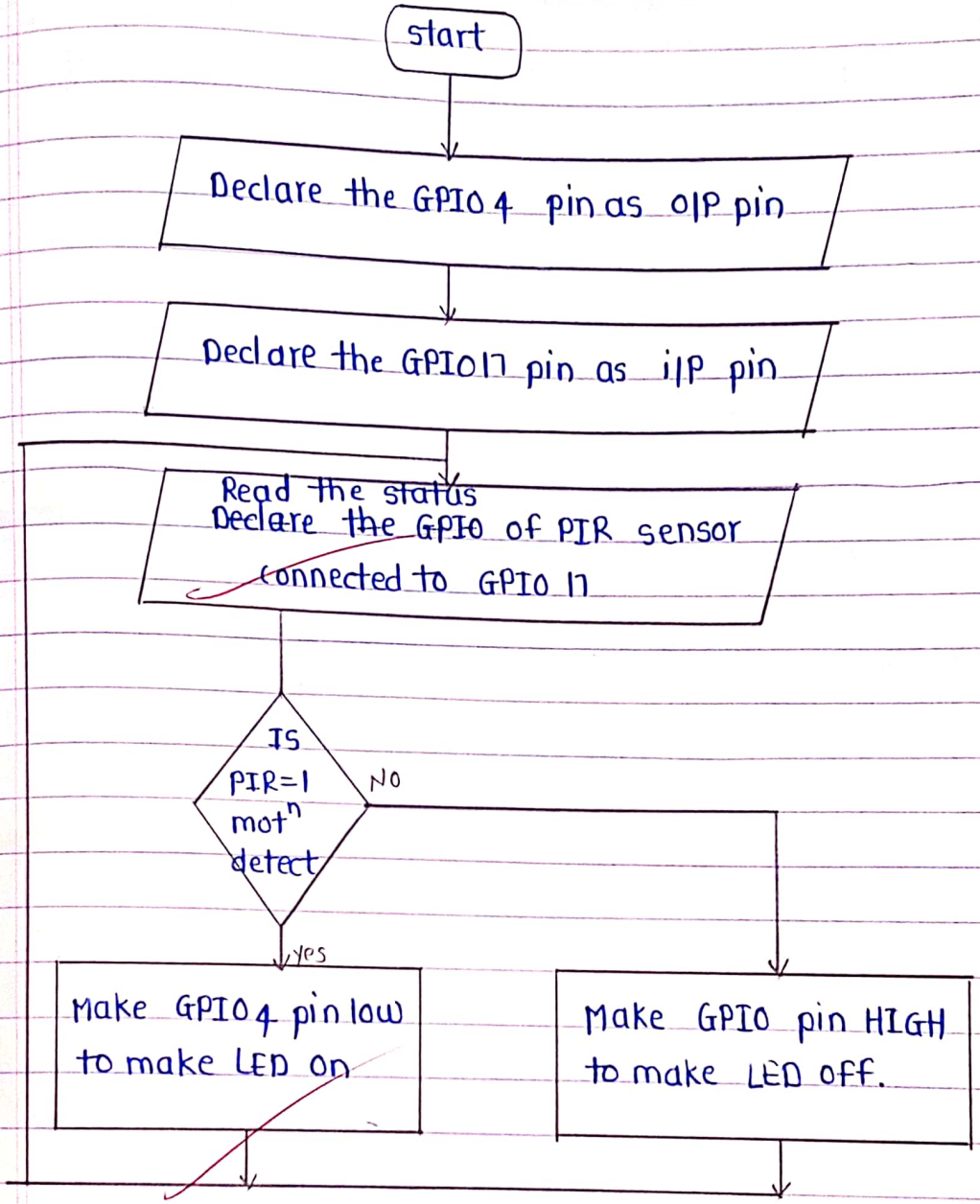
(•) Program =

```
import Rpi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN)
GPIO.setup(4, GPIO.OUT)
While True:
        button_state = GPIO.input(17)
        if button_state == True:
            GPIO output(4, False)
            print('Motion Detected...')
            While GPIO.input(17) == True:
            time.sleep(0.2)
        else
            GPIO.output(4, True)
```

- Application =

(1) Ithrader alarms
(2) Automatic ticket gates
(3) Entry way lightning
(4) security lightning
(5) Hand dryers.

Flowchart=



- Result =

Interfacing and programming of Raspberry for motion detection using PIR sensor is succesfully studied, tested and observe the o/p.

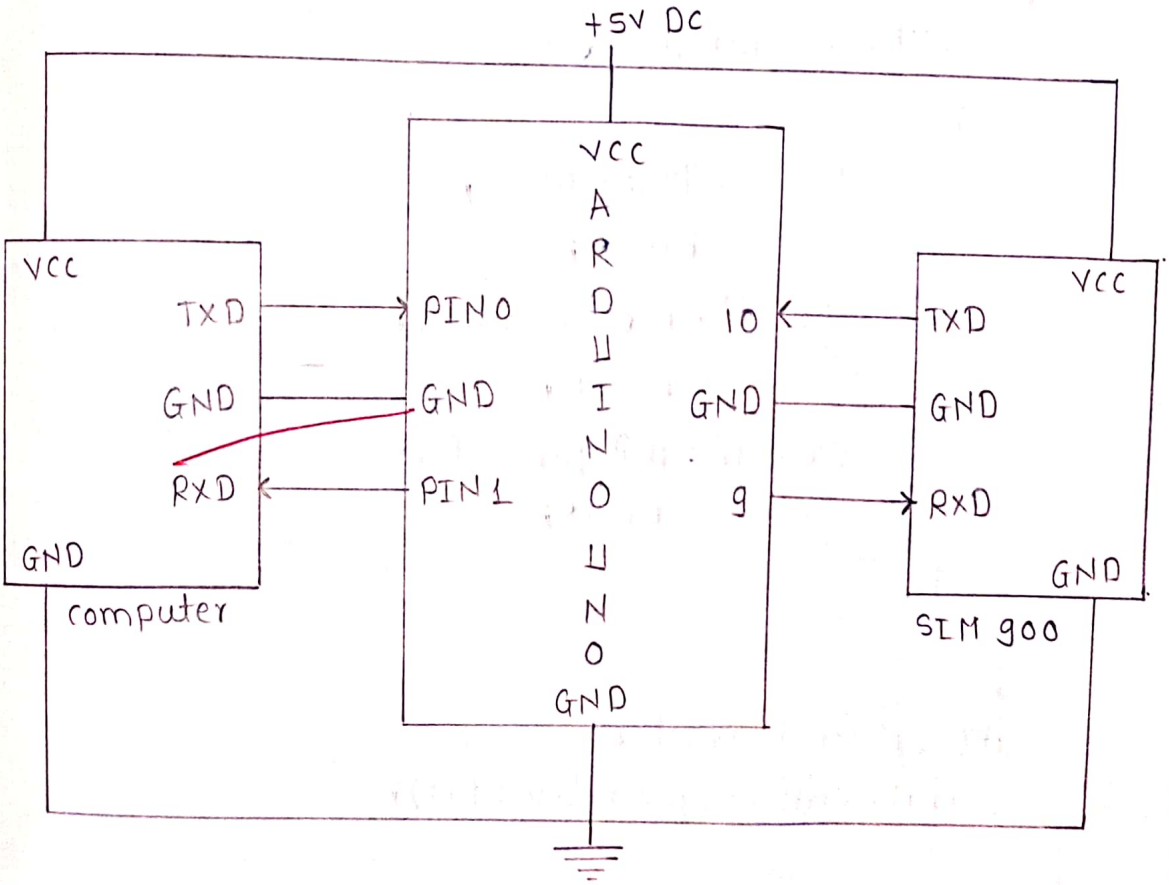
Expt. (1)

Title : Study of GSM System.

AIM : To study GSM system for call.

OBJECTIVES : ① To understand the working principle of SIM800 GSM module.
② To study interfacing of SIM800 GSM module to Arduino.
③ To study Arduino IDE software.

CIRCUIT DIAGRAM :



(•) Program =

```
#include <softwareserial.h>
Software Serial  myserial(10,9);
char call;
void setup()
{
    myserial.begin(9600);
    Serial.begin(9600);
    Serial.println("GSM SIM 900A BEGIN");
    Serial.println("Enter character for control option:
    Serial.println("c : to make a call");
    Serial.println("i: to recieve a call");
    Serial.println("h : to disconnect a call");
    Serial.println();
    delay(100);
}
Void loop()
{
     if(serial.available()>0))
     {
        switch(serial.read())
        {
            case 'c': MakeCall();
                      break;
            case 'i': RecieveCall();
                      break;
            case 'h': HangupCall();
                      break;
        }
     }
     if(myserial.available()>0)
     Serial.write(myserial.read());
}
```

```
void MakeCall()
{
    myserial.println("ATD +91xxxxxxxxxx");
        Serial.println("calling....");
        delay(1000);
}
void HangUpCall()
{
    myserial.println("ATH");
    Serial.println("Hang up call");
    delay(1000);
}
void ReceiveCall()
{
    myserial.println("ATA");
    delay(1000);
    {
        call=myserial.read();
            Serial.print(call);
    }
}
```

(•) Result=

Interfacing SIM800 GSM module with Aurduino is successfully studied, tested and observe the output.

(•) Application=

- Remote control and default reporting of stolen sets.
- Communicate with family, friends and business collegues.

Expt. (2)

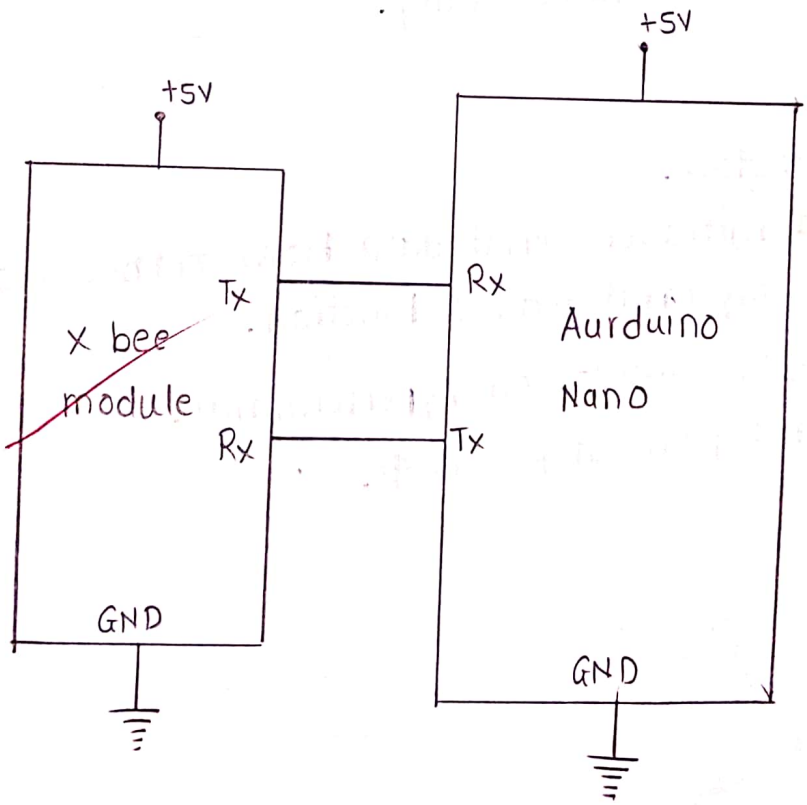
Tile : Study of zig-bee for one application.

AIM : To interface xbee module to Aurduino & establish zigbee communication betw^n Arduino Board and PC.

OBJECTIVES : ① To study concept of zigbee communication.
② To study xbee model interfacing and establish zigbee communication betw^n Aurduino and PC.
③ To study Aurduino IDE software.

(•) Software used = Arduino IDE software.

CIRCUIT DIAGRAM :



(•) Program =

```
void setup ()
{
   serial begin (9600);
}
void loop()
{
   serial.print ("Hello");
}
```

- Application = (1) Home Automation
  (2) wireless sensor Networks.
  (3) Industrial control system.
  (4) Embedded system.
  (5) Building automation.

- Result = Interfacing of ~~zigbee~~ xbee module to Aurduino and zigbee communication of Aurduino with PC is studied successfully.

- Algorithm =
  (1) Start.
  (2) Initialize serial data from zigbee module using myserial read () function.
  (3) See result on hyperterminal.
  (4) Repeat step 3 to 4.

	PAN ID	SH	SL
C	3332	13A200	41E56B32
E	3932	13A200	41E56C4A

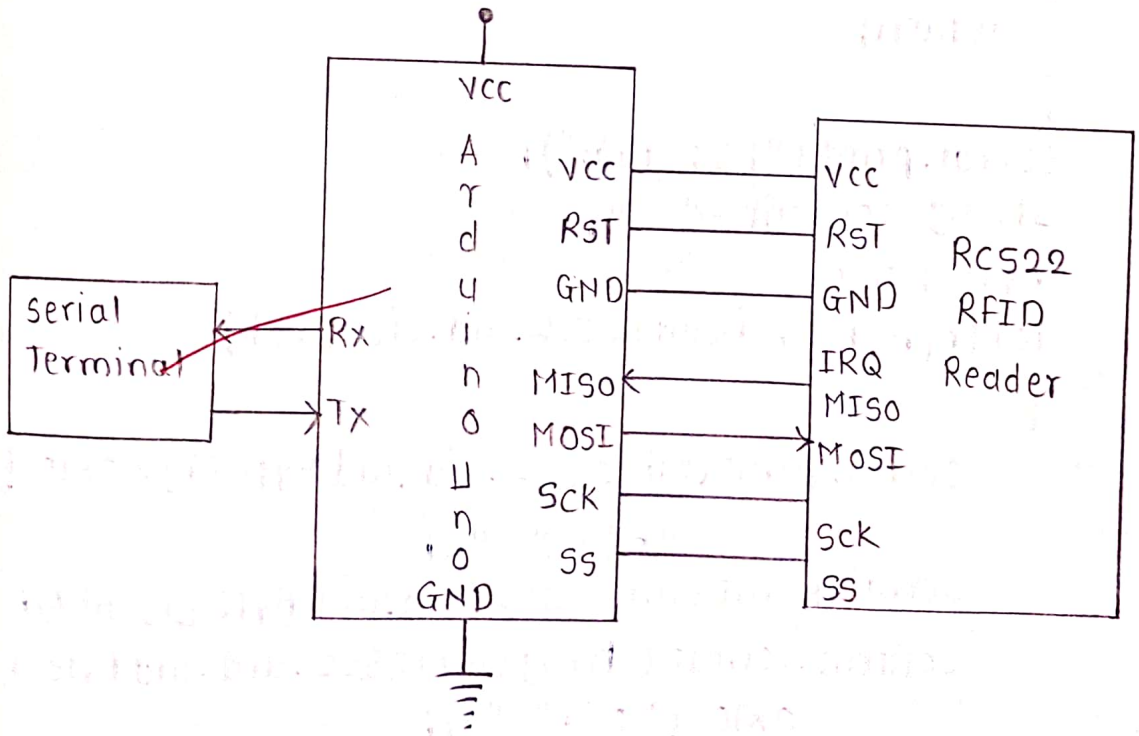
Expt. (3) Title : Study of RFID System.

AIM : To study Rc 522 RFID module interfacing to Arduino.

OBJECTIVES : ① To understand the working principle of RFID module.
② To study interfacing of Rc 522 RFID module to Arduino.
③ To study IDE software.

Software Used : Arduino IDE

CIRCUIT DIAGRAM :

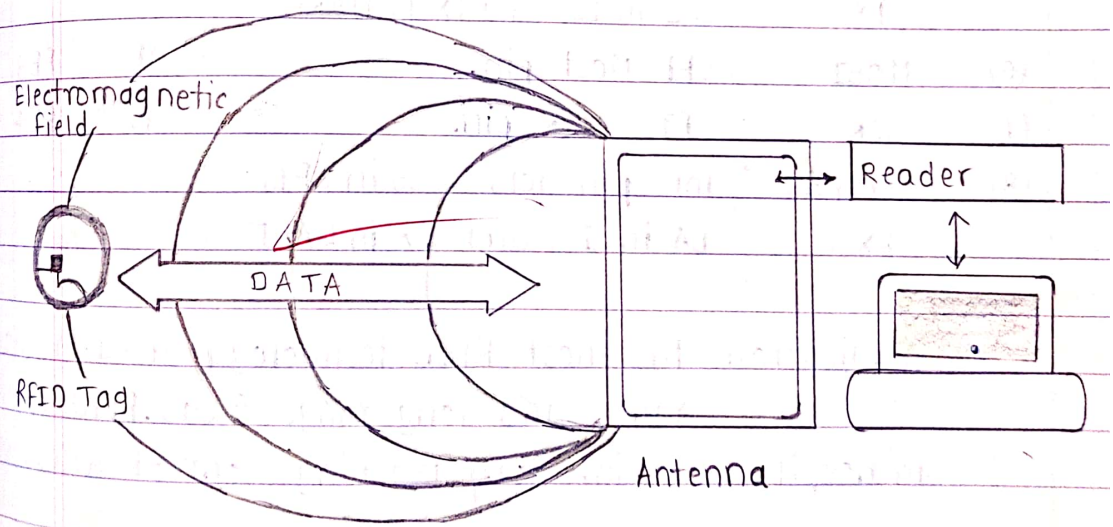


• Program =

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup()
{
    Serial.begin(9600);
    SPI.begin();
    mfrc522.PCD_Init();
    Serial.println("Approximate your card to the
                    reader...");
    Serial.println();
}
void loop()
{
  if(!mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  if(!mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  Serial.print("UID tag:");
  String content="  ";
  byte letter;
  for(byte i=0; i<mfrc522.uid.size; i++)
  {
    Serial.print(mfrc522.uid.uidByte[i]<0x10 ?
                 "0":" ");
    Serial.print(mfrc522.uid.uidByte[i],HEX);
    content.concat(String(mfrc522.uid.uidByte[i]<
        0x10 ?"0":" "));
    content.concat(String(mfrc522.uid.uidByte[i],
                          HEX));
```

```
}
Serial.println();
Serial.print("Message:");
content.toUpperCase();
if(content.substring(1)=="F9 21 E4 D5")
{
        Serial.println("Authorized access");
        Serial.println();
        delay(3000);
}
else
{
        Serial.println("Access denied");
        delay(3000);
}
}
```

(*) Diagram =



(•) Algorithm of RFID=

(1) - start.
(2) - Initialize SPI
(3) - Read data from RFID module using mySerial read()
(4) - Is data equal to store data then make LES ON.
(5) - Repeat step 3 to 4.

(•) Pinout of RC522=

		RC522 RFID	Arduino UNO	
Pin.No	Pin Name	Description	Pin NO	Pin Name
(1)	VCC	3.3V Supply Pin		3.3V
(2)	RST	Reset pin. When LOW, resets the MFRC522 Ic.	9	PB1
(3)	GND	Ground		GND
(4)	IRQ	Interrupt pin. To interrupt host device (Mc)	Nc	Nc
(5)	MISO/SCL/ TX	This pin acts as MISO in SPI, SCL in I2C and TX in UART.	12	PB4
(6)	MOSI	SPI Most pin.	11	PB3
(7)	SCK	SPI clock pin.	13	PB4
(8)	SS/SPA/ RX	This pin acts as ss in SPI, SDA in I2C and RX in UART		

(•) Application = The most RFID technology used in short range and long range areas. RFID applicat in hospitals are inventory tracking, control access, etc.

(•) Result = Interfacing of RFID module with Aurduino is succesfully studied, tested and observed output

Expt. (4)

Tile : Aurduino based LED switching using mobile.

AIM : To study Aurduino based LED switching using mobile.

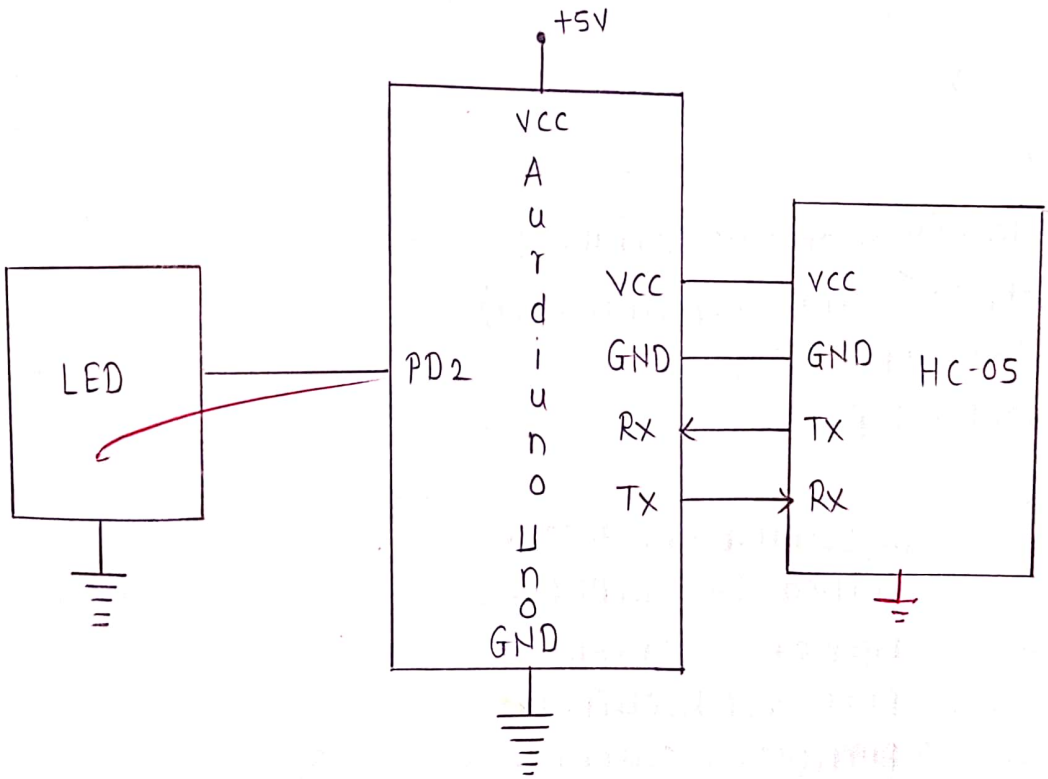
OBJECTIVES : ① To understand the working principle of Hcos Bluetooth model.

② To study interfacing of Bluetooth module to Arduino.

③ To study Aurduino IDE software.

Software Used = Arduino IDE.

CIRCUIT DIAGRAM :



⊙ Program:-

①
```
#include <SoftwareSerial.h>
SoftwareSerial myserial(8,9);
char inputByte;
void setup()
{
   myserial.begin(9600);
   pinmode(2,OUTPUT);
}
void loop()
{
   while(myserial.available()>0)
   {
      inputByte = myserial.read();
      if(inputByte == 'A');
      {
         digitalWrite(2,HIGH);
      }
      else if(inputByte == 'a')
      {
         digitalWrite(2,LOW);
      }
   }
}
```

②
```
#include <SoftwareSerial.h>
SoftwareSerial myserial(8,9)
char inputByte;
void setup()
{
      myserial.begin(9600);
      pinMode(2,OUTPUT);
      PinMode(3,OUTPUT);
      pinMode(4,OUTPUT);
      pinMode(5,OUTPUT);
}
```

```
void loop()
{
    while (mySerial.available()>0)
    {
        inputByte = mySerial.read();
        if (inputByte == 'A')
        {
            digitalWrite (2,HIGH);
        }
        else if(inputByte == 'a')
        {
            digitalWrite (2, LOW);
        }
        if (inputByte == 'B')
        {
            digitalWrite (3,HIGH)
        }
        else if (inputByte == 'b')
        {
            digitalWrite (3,LOW);
        }
        if (inputByte == 'c')
        {
            digitalWrite (4, HIGH);
        }
        else if (inputByte == 'c')
        {
            digitalWrite (4,'LOW');
        }
        if (inputByte == 'D')
        {
```

```
        digitalWrite(5,HIGH);
      }
      else if(inputByte == 'd')
      {
        digitalWrite(5,'low');
      }
    }
}
```

Application=

Bluetooth consume very small an of energy. Therefore it is used in smartphone con robot or car.

Result=

Interfacing of Bluetooth with Arduino succesfully studied, tested and observe the out

Flowchart=

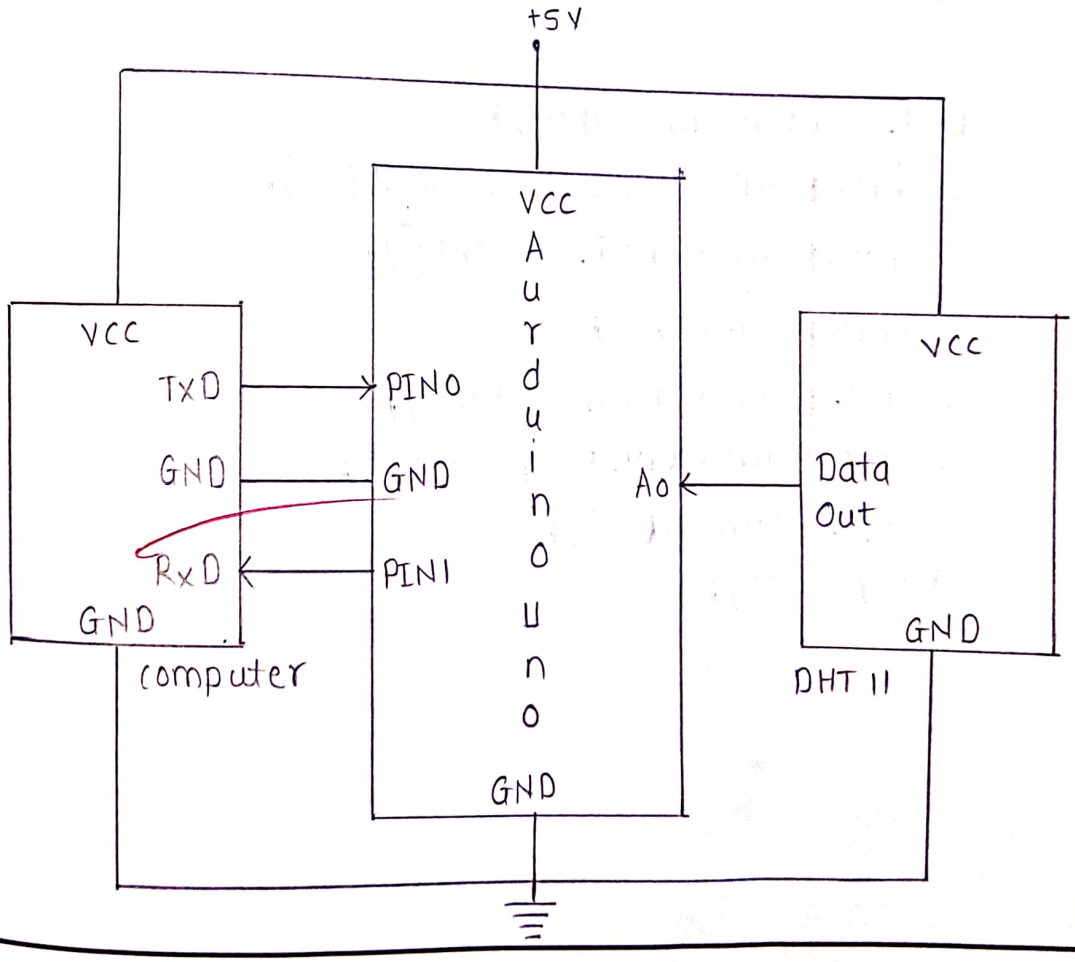
Expt. (5)

Title : Study of temperature and humidity sensing Aurduino

AIM : To study and understand interfacing temperature and humidity sensor (DHT11) to Arduino.

OBJECTIVES :
1. To understand the working principle of DHT 11 sensor.
2. To study interfacing of DHT 11 sensor to Aurduino
3. To study Aurduino IDE software.

CIRCUIT DIAGRAM :



(•) Algorithm =

(1) Start.
(2) Initialize DHT 11 sensor.
(3) Read data from DHT11 sensor.
(4) see result on hyperterminal.
(5) Repeat step 3 to 4.

(•) Program =

```
#include <dht.h>
# define dht_pin A0
dht DHT;
void setup()
{
      Serial.begin (9600);
      delay (500);
      Serial.println("DHT11 Humidity & temperature
                        sensor \n\n");
      delay (1000);
}
void loop ()
{
      DHT.read11(dht_apin);
      Serial.print ("current humidity =");
      Serial.print(" DHT.humidity");
      Serial.print("%");
      Serial.print("temperature");
      Serial.print (DHT.temperature);
      Serial.println("C");
      delay (5000);
}
```