Title

1. Experiments to switch on/off LED's with the user's choice.

Hardware Requirements / components:

(i) Arduino UNO board.
(ii) LED
(iii) Jumper wires F-M
(iv) Bread board and USB cable

Software component: Arduino IDE

Setup connection

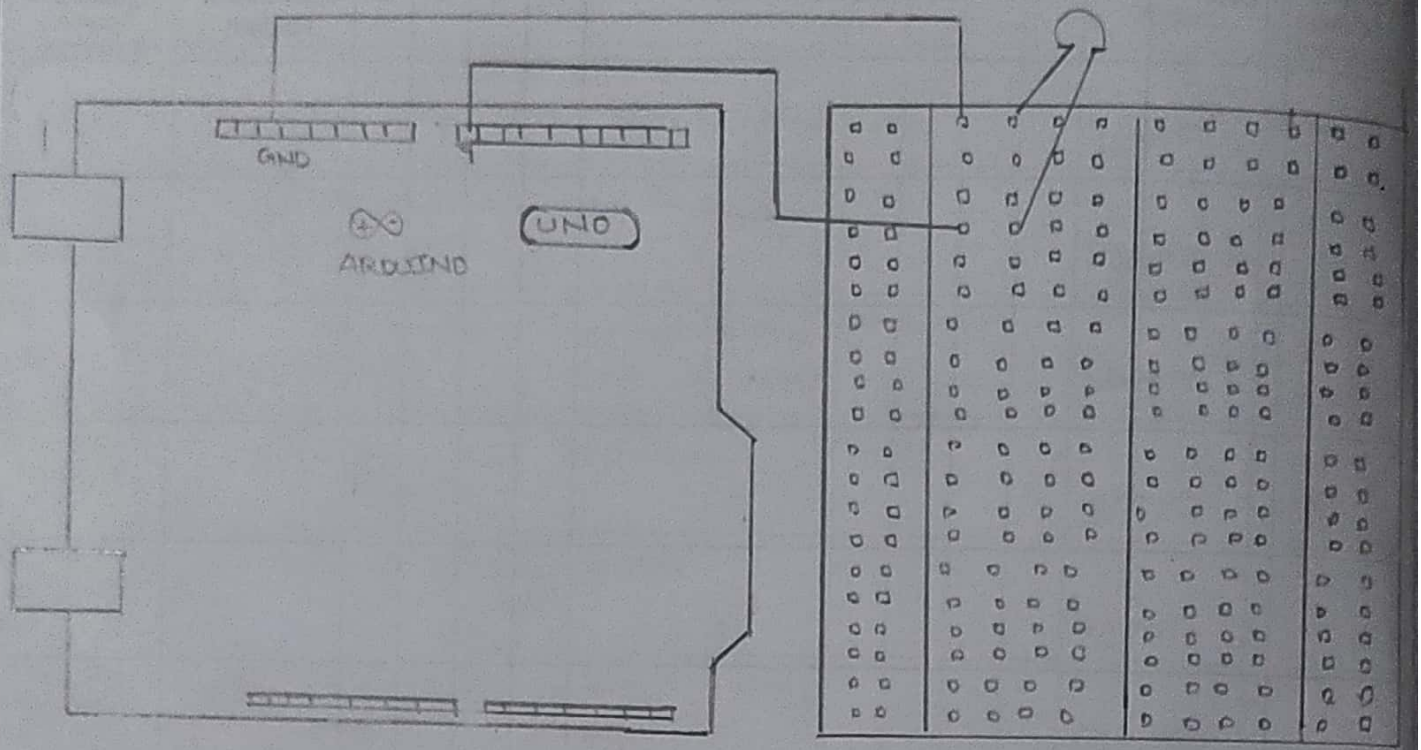
(i) Connect the short leg of LED (cathode) to ground (GND) pin of Arduino.
(ii) Connect the long leg of LED (anode) to the pin of Arduino.
(iii) Connect the USB cable both, Arduino and to your computer → write and upload the code to Arduino to Run the program.

CODE

```
void setup() {
    pinMode (7, OUTPUT);
}
void loop() {
    digitalWrite (7, HIGH);
    delay (500);
    digitalWrite (7, LOW);
    delay (500);
}
```

Date | Signature of the faculty

OUTPUT



Title : Experiment with IR Sensor and LED.

Hardware requirement / Component

(i) Arduino UNO board
(ii) ID LED - Photodiode Module x1
(iii) LED [1]
(iv) Breadboard
(v) Jumper wires

Software component : Arduino IDE

Setup connection

(i) Connect the VCC of IR sensor to 5V of Arduino UNO board.
(ii) Connect the GND of ID Sensor to GND of Arduino UNO board.
(iii) Connect the output of the sensor module to pin 2 of the Arduino
(iv) Connect the anode of a LED at pin 7 connect the cathode to GND of Arduino

CODE

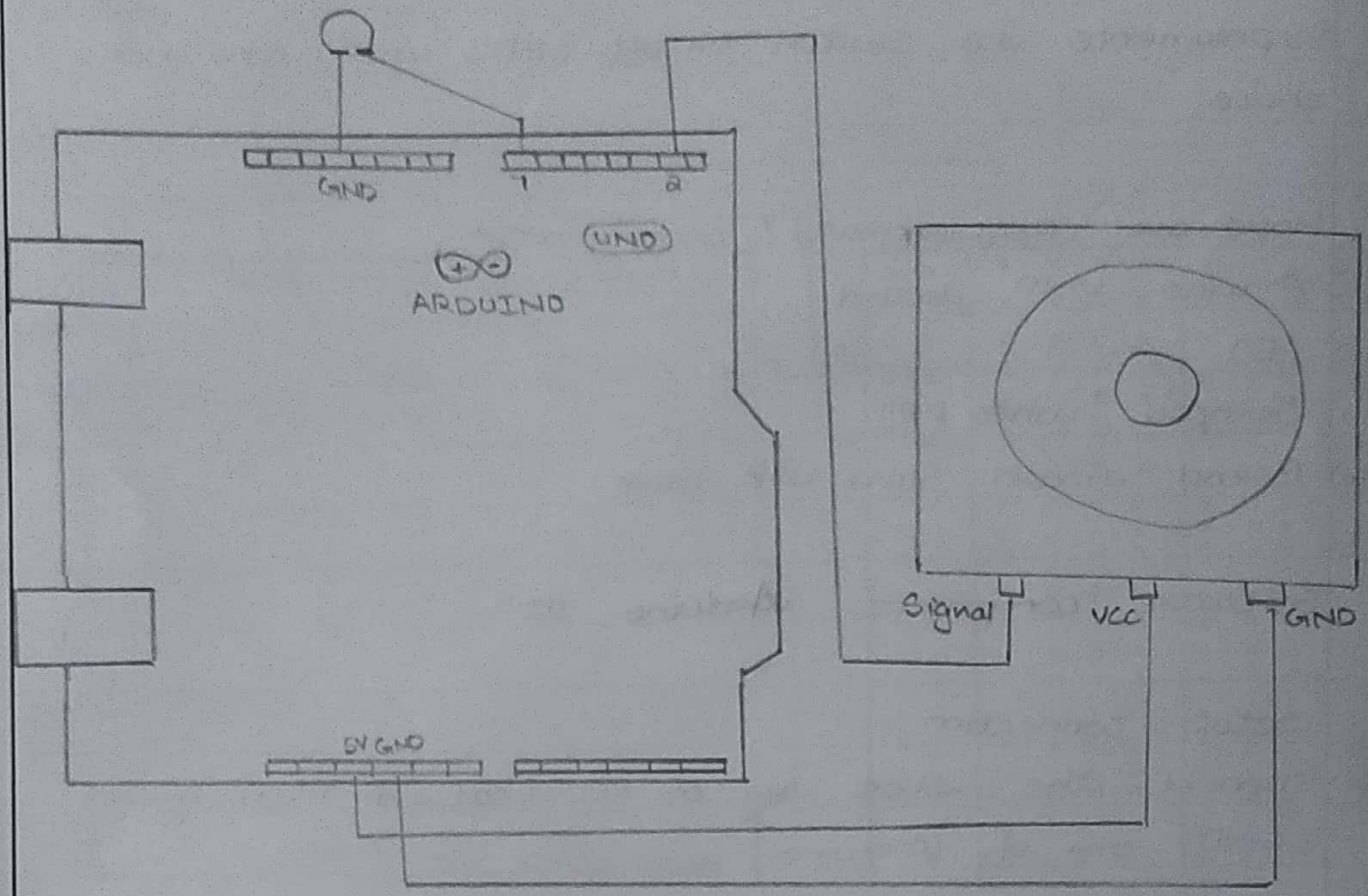
```
#define PIR_Sensor 2
#define Led 4
int pirState = LOW;
int lastState = LOW;
void setup () {
  pinMode (PIR_Sensor, INPUT);
  pinMode (7, OUTPUT);
  Serial.begin (9600);
}
```

Date | Signature of the faculty

```
void loop ()
{
  pirState = digitalRead (PIR_Sensor)
  if( pirState == HIGH && lastPirState == LOW)
  {
    Serial.println("MOTION DETECTED");
    digitalwrite (LED, HIGH);
    delay (1000);
  }
  else if (pirState == LOW && lastPirstate == HIGH)
  {
    Serial.println ("MOTION STOPPED");
    digitalwrite (LED, LOW);
    delay (1000);
  }
  lastPirState = pirState;
}
```

Date	Signature of the faculty

OUTPUT



le 3. Experiment to measure the distance using ultrasonic sensors.

Hardware components

(i) Arduino UNO board

(ii) Ultrasonic sensor

(iii) Jumper wires (3)

Software component : Arduino IDE

Setup connection

(i) Connect the echo pin of the sensor to the D2 pin of the Arduino.

(ii) Connect the trig Pin of the sensor to the D3 pin of the Arduino

(iii) Connect VCC of sensor to 5V of Arduino.

(iv) Connect GND of sensor to GND of Arduino.

(v) Verify and compile the code, then upload the code of the Arduino UNO board.

(vi) Monitor the output in the Serial monitor (set the Tools Serial monitor or press Ctrl + Shift + M.

CODE

```
#define echoPin 2
#define trigPin 3
long duration;
int distance;
void Setup {
    pinMode (trigPin, OUTPUT);
    pinMode (echoPin, INPUT);
    Serial.begin (9600);
```

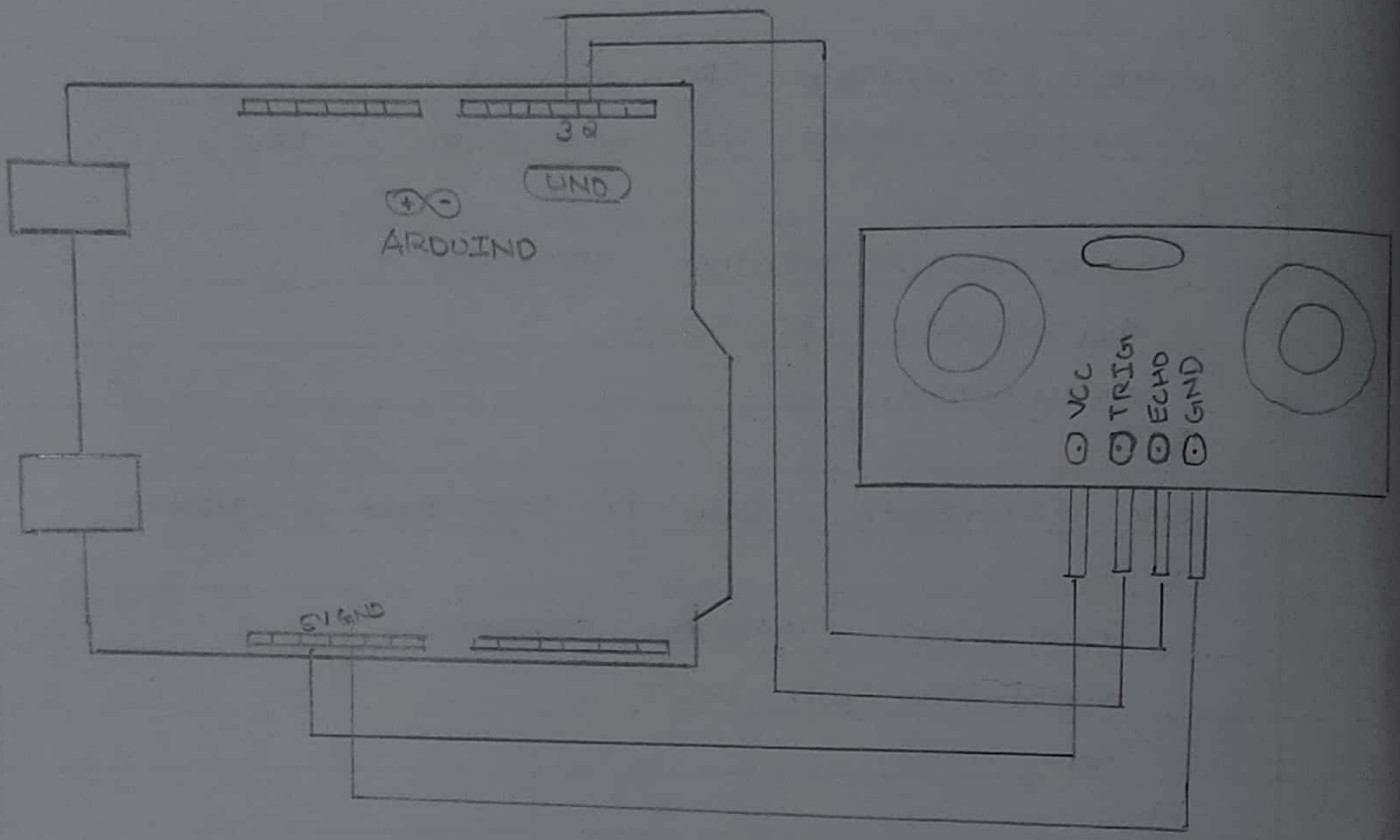
Date | Signature of the faculty

title

```
Serial.println("Distance measuring using Arduino");
delay(500);
}
void loop(){
digitalWrite(trigPin, LOW);
delayMicroSeconds(2);
digitalWrite(trigPin, HIGH);
delayMicroSeconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * 0.0344/2;
Serial.print("Distance:");
Serial.print(distance);
Serial.println("cm");
delay(100);
}
```

Date	Signature of the faculty

OUTPUT



Title 4: Experiment to check moisture.

Hardware components :

(i) Soil moisture sensor.
(ii) Arduino UNO board.
(iii) Jumper wires [5]
(iv) LED [1]

Software Component : Arduino IDE

Setup Connection

* Moisture Sensor- to Arduino Board.
(i) Connect GND of Sensor to GND of Arduino
(ii) Connect 5V of Arduino to power VCC of sensor.
(iii) Connect signal pin of Sensor to A0 of Arduino
* LED to Arduino
(iv) Connect GND of Arduino to cathode (short leg) of LED.
(v) Connect 7th pin of Arduino to anode (long leg) of LED.

CODE

```
void setup(){
    Serial.begin(9600);
    pinMode(7, OUTPUT);
}
void loop(){
    int level;
    level = analogRead(0);
    Serial.println("Analog value:");
    Serial.println(level);
```

Date	Signature of the faculty

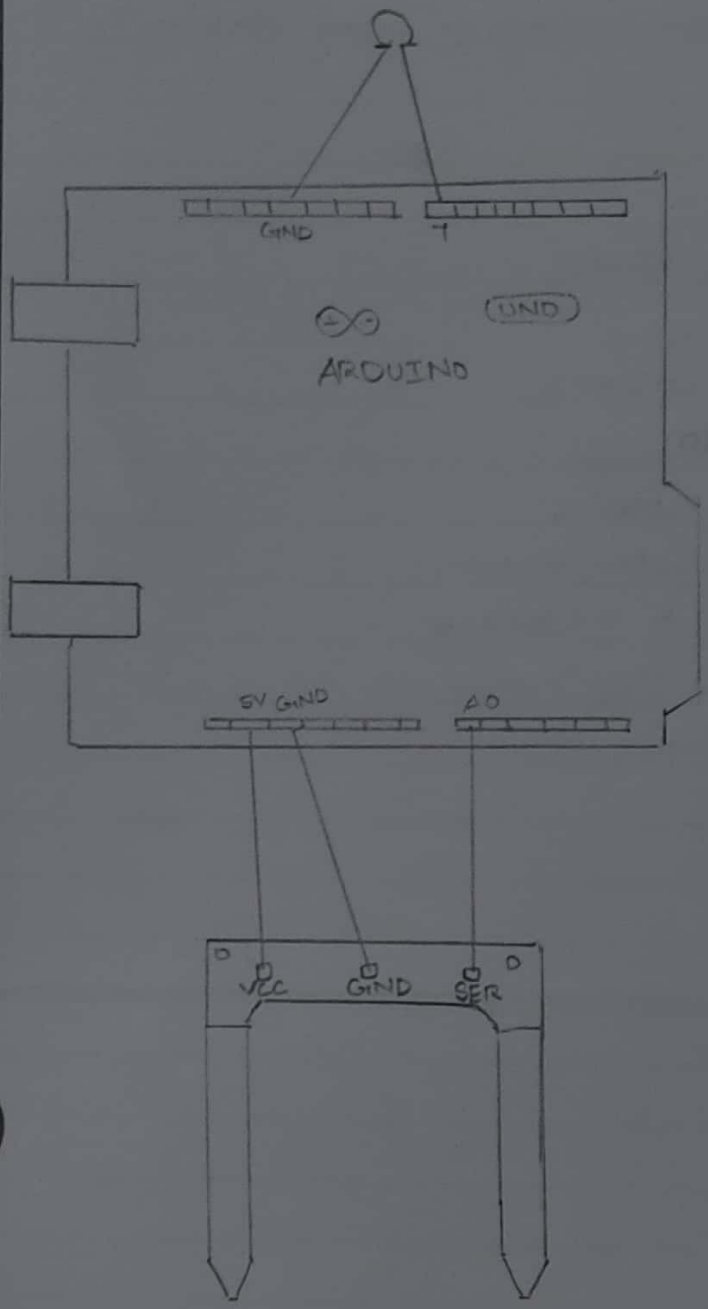
Title

```
if (level <175)
{
   digitalwrite (7, HIGH);
   Serial.println ("LED ON!");
}
else if (level >175)
{
   digitalwrite (7, LOW);
   Serial.println ("LED OFF:");
}
}
```

Date

Signature of the faculty

OUTPUT



5 Experiment with Servo Motor

Hardware Components:

(i) Servo Motor
(ii) Arduino UNO
(iii) Jumper wires [3]

Software Component: Arduino IDE

Setup Connection

1. Connect the VCC of servo motor to 5V pin.
2. Connect the GND of servo motor to GND.
3. Import the header files to Arduino code.
4. Compile and upload the code to Arduino board

CODE

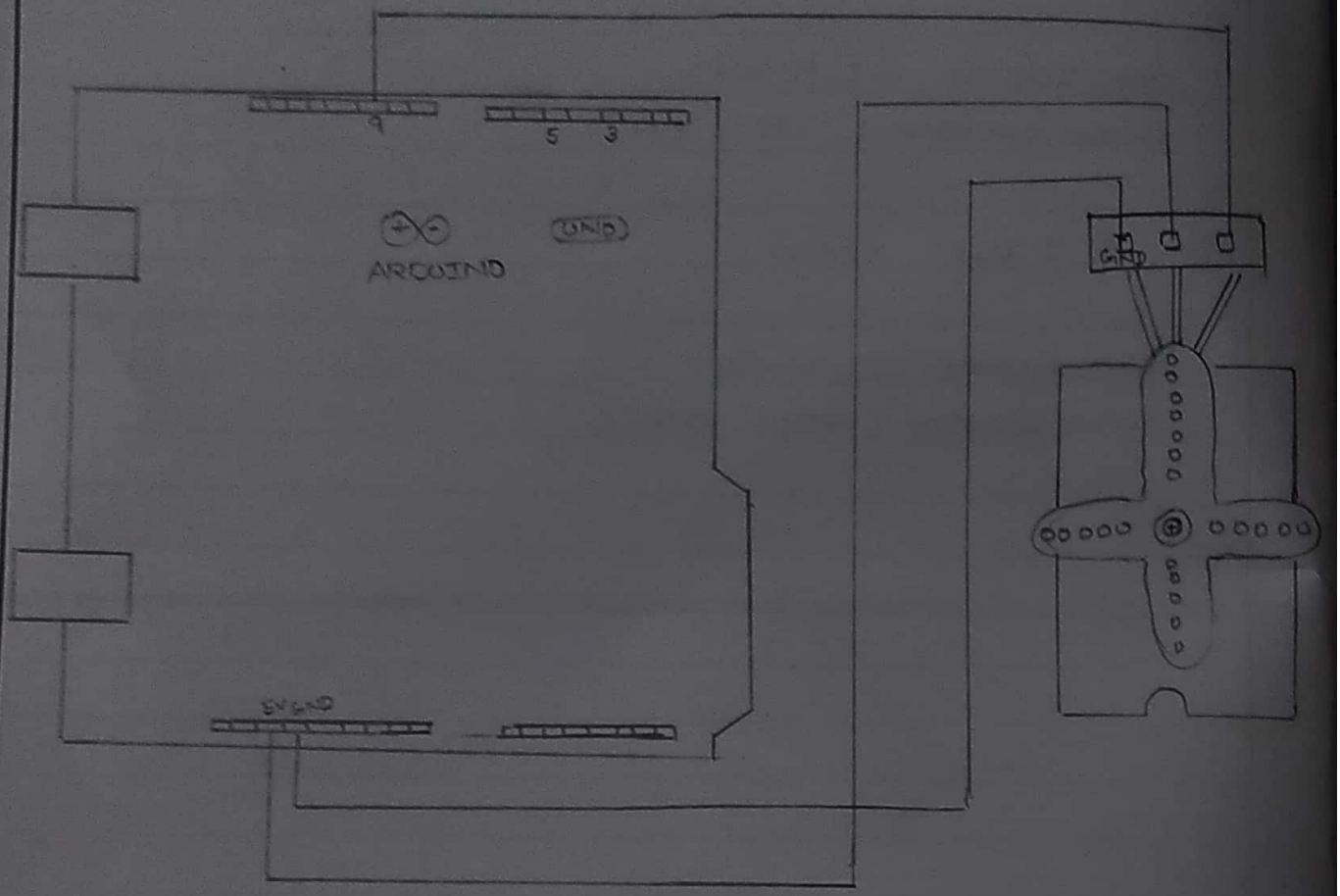
```
#include <Servo.h>
int pos=0;
Servo servo_9;
void setup(){
    servo_9.attach(9);
}
void loop(){
  for(pos=0; pos<=90; pos+=1){
    servo_9.write(pos);
    delay(10);
  }
```

Date	Signature of the faculty

```
delay(1000);
for (pos=90; pos<=0; pos--)
{
  servo_9.write(pos);
  delay(10);
}
delay(100);
}
```

Date | Signature of the facult

OUTPUT



Title

6. Experiment with ultrasonic sensor and servo motor

Hardware components:

(i) Servo Motor
(ii) Ultrasonic sensor (HC)
(iii) Jumper wires and bread board (optional)
(iv) Arduino UNO board

Setup Connection:

1. Connect ultrasonic Sensor (HC - SR04)
   - VCC Pin of the Sensor to 5V Pin on Arduino UNO
   - GND Pin of the Sensor to GND pin on Arduino UNO
   - TRIG Pin of the sensor of digital pin 3 on Arduino
   - ECHO Pin of the sensor of digital pin 2 on Arduino

2. Connect Servo Motor
   - Connect the signal (usually orange or yellow) of the servo motor to digital pin 9 on Arduino UNO.
   - Connect the VCC wire (usually red) of the servo motor to the 5V pin on Arduino UNO
   - Connect the GND wire (usually brown or black) of the GND pin on Arduino UNO.

3. Power
   - Connect the Arduino UNO board to your computer via USB cable for power.

Date	Signature of the faculty

Concept:

- If the distance is less than equal to 10cm, servo motor opens the gate by rotating to 90°
- If the distance increase (object moves away), the servo motor closes the gate by rotating to 0 degree
- Write code in Arduino IDE → Connect USB to maintain → Select board and port → upload → open serial monitor

CODE:

```
#include <Servo.h>
#define echoPin 2
#define trigPin 3
int pos = 0;
Servo servo_9;
long duration;
int distance;

void setup(){
    servo_9.attach(9, 500, 2500);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(9600);
    Serial.println("Distance measurement using Arduino");
    delay(500);
}
void loop(){
```

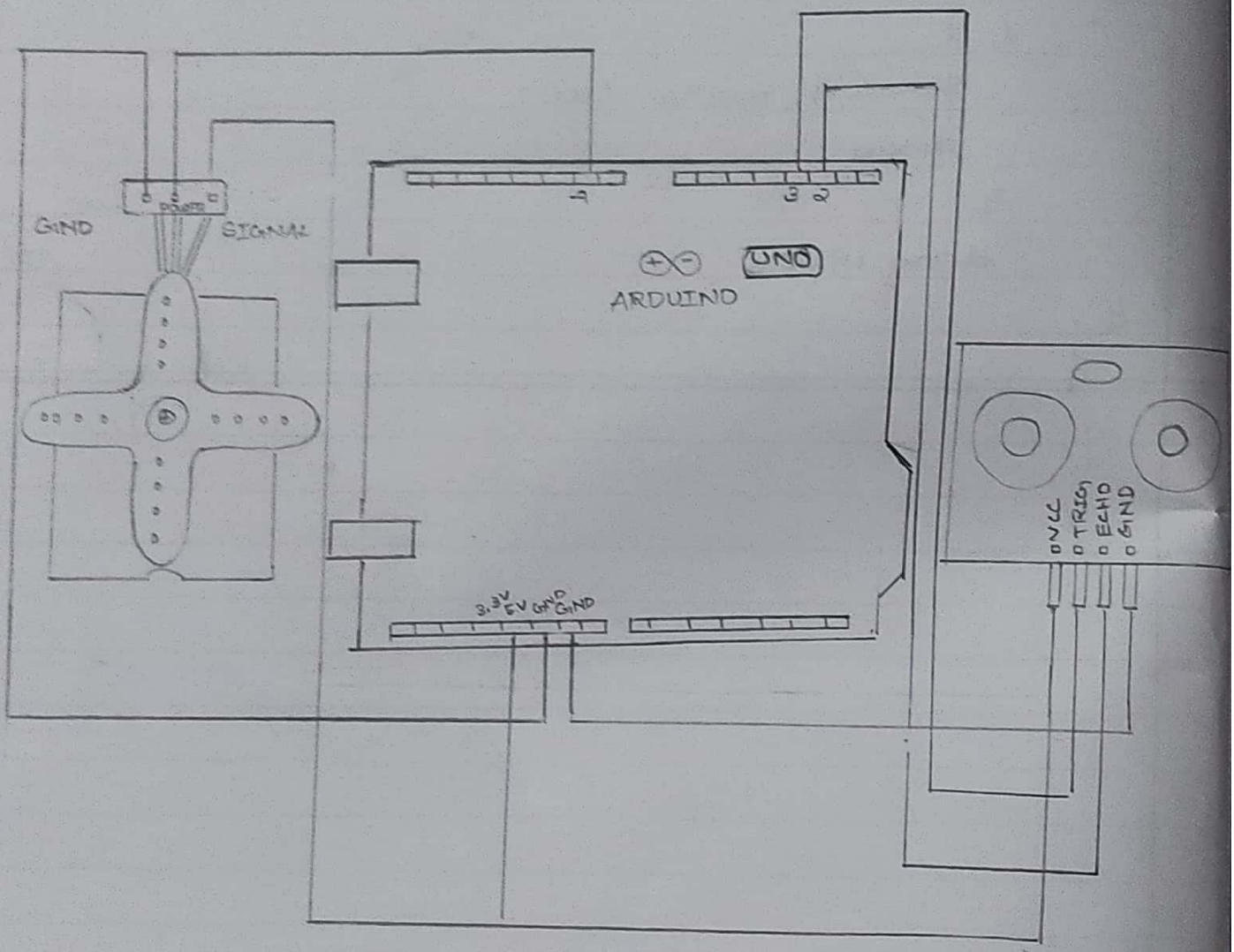
Date	Signature of the faculty

CMRU

```
digitalWrite (trigPin, LOW);
delay MicroSeconds (2);
digitalWrite (trigPin, HIGH);
delayMicroSeconds (10);
digitalWrite (trigPin , LOW);

duration = pulseIn ( echoPin, HIGH);
distance = duration * 0.0344/2;

Serial.print ("Distance:");
Serial.print (distance);

if ( distance <= 10)
{
  for ( pos = 0; pos <= 90; pos += 1)
  {
    servo_9.write (pos);
    delay(15);
  }
  delay (1000);
  for (pos = 90; pos <= 0; pos -= 1)
  {
    servo_9.write (pos)
    delay (15);
  }
}
Serial.println("cm");
delay(100);
}
}
```

	Date	Signature of the faculty

OUTPUT



Experiment to read the temperature and switch on the fan/LED

Hardware Components:

(i) DHT11 Sensor
(ii) Fan
(iii) Jumper wires (5)
(iv) USB Cable
(v) ESP32 WROOM 32 board

Software Components:

(i) Micropython
(ii) Thony IDE

Connections:

Fan => VCC -> VIN, SIGNA -> D2

DHT11 => VCC -> 3V3, DAI -> D24, GND -> GND

CODE:

```
import Adafruit_DHT

Sensor = Adafruit_DHT.DHT11
DHT11_pin = 23
fan = 21
humidity, temperature = Adafruit_DHT.read_retry(sensor,
DHT11_pin)
```

	Date	Signature of the faculty

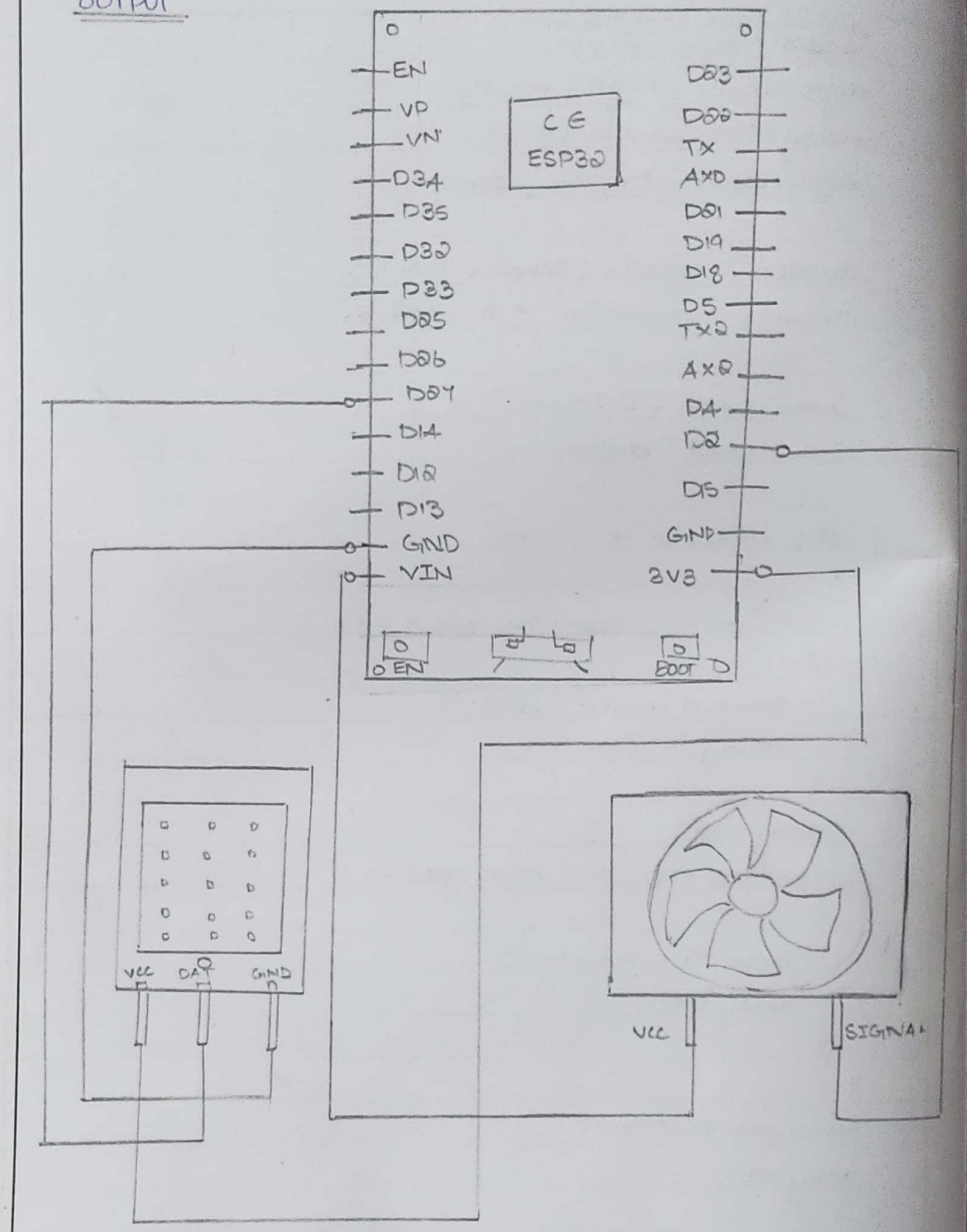
Title

```
if humidity is not None and temperature is not None:
    if temperature > 25:
        GPIO.output(fan, GPIO.HIGH);
    print("Temperature = {0:0.1f}*C Humidity = {1:0.1f}%", format(temperature, humidity))
else:
    print("Failed to get reading from the sensor. Try again!")
```

Date

Signature of the faculty

OUTPUT


EN
VP
VN
D34
D35
D32
D33
D25
D26
D27
D14
D12
D13
GND
VIN
D23
D22
TX
AXD
D21
D19
D18
D5
TX2
AX2
D4
D2
D15
GND
3V3
CE
ESP32
EN
BOOT
VCC
DAT
GND
VCC
SIGNAL

Title

8. Create a code to send email from the board

Sending Mail using MIME.

```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
fromadd = "Sender@gmail.com"
toadd = "receiver@gmail.com"
msg = MIMEMultipart()
msg['From'] = fromadd
msg['TO'] = toadd
msg['Subject'] = 'Python Email
Body = "PythonMail"
msg.attach (MIMEText ('body', 'plain'))

import smtplib
Sewer = smtplib.SMTP ('smtp.gmail.com', 587);
Sewer.ehlo_or_helo_if_needed()
Sewer.starttls()
Sewer.ehlo_or_helo_if_needed()
Sewer.login ("sender@gmail.com", "Password")
text = msg.as_string()
Sewer.sendmail (fromadd, toadd, text)
```

Date	Signature of the faculty

Title

9 IOT experiment to read the temperature and Humidity and push it to IOT cloud for visualization

Hardware components

(i) DHT11 Sensor

(ii) ESP32 WROOM 32 board

(iii) Jumper wires (3)

Software components:

(i) Micropython, Thony IDE

Setup connection:

- Connect GND of DHT11 sensor to GND of ESP32
- Connect GND of DHT11 sensor to 3V3 of ESP32

ThingSpeak Cloud:

- Go to the official website of ThingSpeak IOT cloud
- Create an account set password and continue
- Create a new channel and give the name, description and set the fields field1 as temperature and field2 as humidity
- Write the code and then upload.

CODE:

```
import sys
from utime import sleep
import urllib3
```

Date

Signature of the faculty

Title

```
def getSensorData():
    return (25, 40)

def main():
    print ('Starting')
    key = '1U06J70U98QA342P'
    baseURL = "https://api.thingspeak.com/update?api_key
=%s' %key
    print('* * * * * * * * * * * * * * * * * *')
    while True:
        try:
            RH, T = getSensorData()
            http = urllib3.PoolManager()
            r = http.request ('POST', url = baseURL + '&field1 =%s
&field2 = %s' % (T, RH))
            print (r.data())
            print ("Humidity" + str(RH) + "%")
            print ("Temperature" + str(T) + "C")
            sleep (15)
        except:
            print (sys.exc_info()[0])
            break
main()
```

Date	Signature of the faculty

CMRU

Contents

Expt. # / Page #	Date	Title of the Experiment	R	C	V	ST	Signature of Student	Signature Course Instructo
1. (1)		1. Experiments to switch on/ off LED's with user's choice					Bhavana	
2. (2-3)		2. Experiment with IR Sensor and LED					Bhavana	
3. (4-5)		3. Experiment to measure the distance using ultrasonic sensor					Bhavana	
4. (6-7)		4. Experiment to check moisture					Bhavana	
5. (8-9)		5. Experiment with Servo motor.					Bhavana	
6. (10-12)		6. Experiment with ultrasonic sensor and servo motor					Bhavana	
7. 13-14		7. Experiment to read the temperature & switch on/off LED					Bhavana	
8 15		8. Create a code to send email from the board.					Bhavana	
9. 16-17		9. Experiment to read the temperature & humidity					Bhavana	

R=Marks for record; C= Marks for conduct of experiment; V=Marks for Viva; ST=Sub total