Assignment 1

Aim:- Design and implement IoT system using Arduino Uno/ Raspberry Pi using 'Ultrasonic sensor and Servo motor' such as 'Door opener in home automation'

Theory:-

Ultrasonic sensor-

An **ultrasonic sensor** is a device that measures the distance to an object using sound waves. It emits high-frequency sound waves (typically above the human hearing range, hence "ultrasonic") and then listens for the echo that bounces back after hitting an object. By calculating the time it takes for the sound wave to return, the sensor can determine the distance to the object.

Key Components:

- **Transmitter**: Sends out ultrasonic sound waves.
- **Receiver**: Detects the reflected waves.
- **Controller**: Calculates the time difference and determines the distance.

Applications:

- **Distance measurement**: Used in robots for obstacle avoidance.
- **Object detection**: In parking sensors for cars.
- **Level measurement**: For liquid levels in tanks.

Servo motor-

A **servo motor** is a rotary or linear actuator designed to precisely control angular or linear position, velocity, and acceleration. It uses a feedback system to achieve accurate control of its movement.

Key Components:

- **Motor**: Typically a DC or AC motor that drives the movement.
- **Control circuit**: Processes input signals and commands the motor.
- **Feedback mechanism**: Usually a sensor (like a potentiometer or encoder) that provides position feedback to ensure accurate movement.

Applications:

- **Robotics**: Used for precise joint movement.
- **RC vehicles**: Controls steering, flaps, etc.
- **Automation**: In manufacturing for precise positioning tasks.

Steps to Design the IoT-Based Door Opener System

1. **Ultrasonic Sensor (HC-SR04) Pin Connections**:
    - **VCC**: Connect to 5V on Arduino.
    - **GND**: Connect to GND on Arduino.
    - **Trig**: Connect to a digital pin (e.g., Pin 9) on Arduino.
    - **Echo**: Connect to a digital pin (e.g., Pin 8) on Arduino.
2. **Servo Motor Pin Connections**:
    - **Red wire (VCC)**: Connect to 5V on Arduino.
    - **Brown wire (GND)**: Connect to GND on Arduino.
    - **Yellow/Orange wire (PWM Signal)**: Connect to a PWM digital pin (e.g., Pin 6) on Arduino.

Ultrasonic sensor Code:-

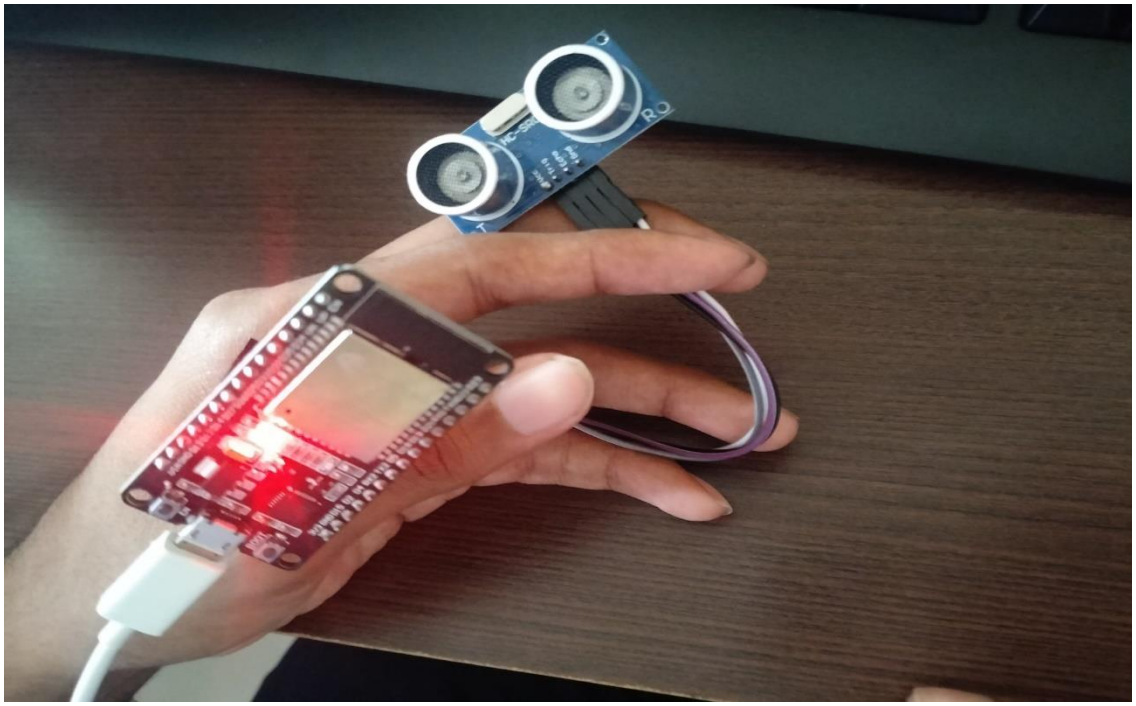
```
int TRIG_PIN=5;
int ECHO_PIN=18;
int SOUND_SPEED=0.034;  // Speed of sound in cm/us

int duration;
int distanceCm;

void setup() {
  Serial.begin(115200);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

void loop() {
  // Trigger the ultrasonic pulse
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  // Read the echo time
  duration = pulseIn(ECHO_PIN, HIGH);

  // Calculate the distance (duration * speed of sound / 2)
  distanceCm=(duration*0.034)/2;

  // Print the distance to Serial Monitor
```

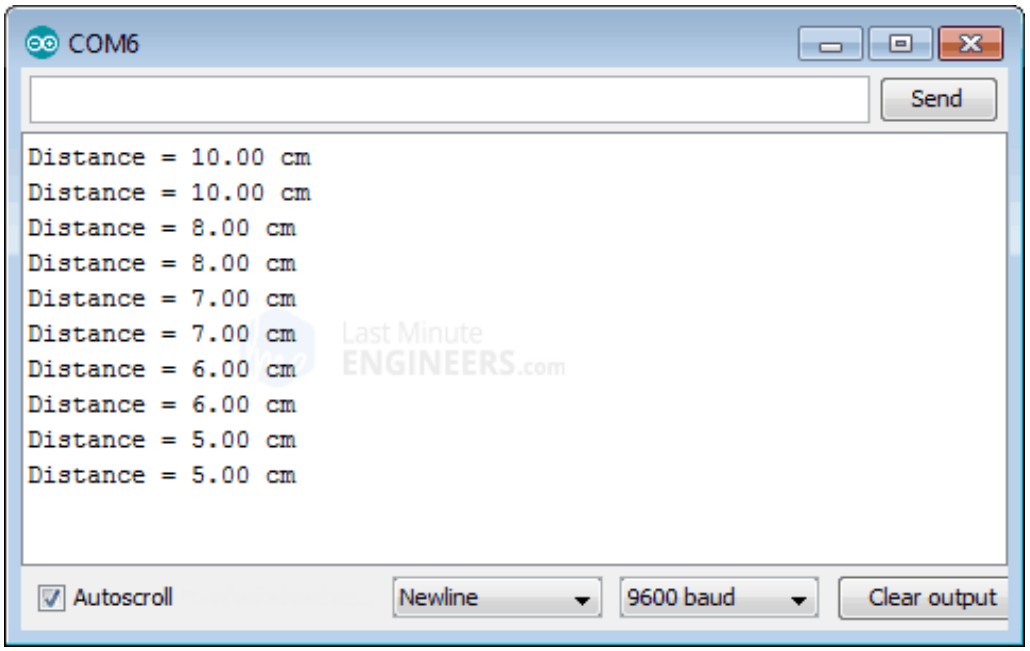
```
    Serial.print("Distance: ");
    Serial.print(distanceCm);
    Serial.println(" cm");
    delay(500);
  }
```

Servo motor Code:-

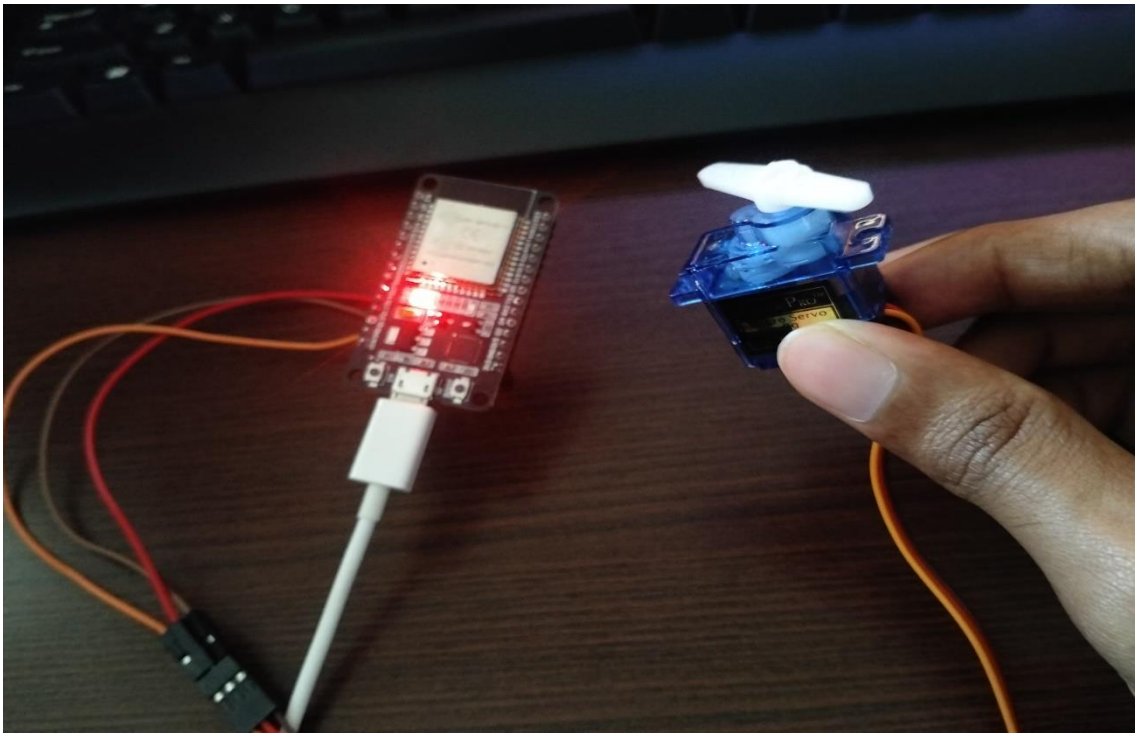
```
#include <ESP32Servo.h>
Servo myservo;  // Create servo object to control a servo
int servoPin = 27;  // Pin connected to the servo signal
int pos = 0;       // Variable to store the servo position

void setup() {
  myservo.attach(servoPin);  // Attach the servo on GPIO 27 to the servo object
  Serial.begin(115200);
}
void loop() {
  // Sweep from 0 to 180 degrees
  for (pos = 0; pos <= 180; pos += 1) {
    myservo.write(pos);
    Serial.println(pos);            // Set the servo position
    delay(15);                    // Wait for the servo to reach the position
  }
  // Sweep back from 180 to 0 degrees
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos);
    Serial.println(pos);          // Set the servo position
    delay(15);                    // Wait for the servo to reach the position
  }
}
```

Ultrasonic sensor-





Servo motor-



Assignment 2

Aim:- Design and implement parameter monitoring IoT system keeping records on Cloud such as 'environment humidity and temperature monitoring'.

Theory:-

Humidity sensor-

A **humidity sensor** is a device used to measure the amount of moisture or water vapor present in the air. It provides readings of **relative humidity**, which is the ratio of the current moisture in the air to the maximum amount of moisture the air can hold at a given temperature, expressed as a percentage.

There are different types of humidity sensors, such as:

1. **Capacitive**: Measures humidity based on changes in electrical capacitance caused by varying moisture levels.
2. **Resistive**: Uses the change in electrical resistance of a material when exposed to different humidity levels.
3. **Thermal**: Measures the difference in thermal conductivity between dry and moist air.

Humidity sensors have a wide range of applications across various industries. Here are some key examples:

1. **HVAC Systems**: Used in heating, ventilation, and air conditioning to maintain comfortable indoor environments by regulating humidity levels.
2. **Weather Stations**: Measure humidity as part of atmospheric data collection for weather forecasting and climate monitoring.
3. **Agriculture**: Help monitor and control humidity levels in greenhouses and crop storage areas, ensuring optimal conditions for plant growth and preventing mold.
4. **Home Appliances**: Found in dehumidifiers, air purifiers, and smart home systems to control moisture levels for better air quality.
5. **Medical Devices**: Used in respiratory equipment, incubators, and other medical instruments to maintain precise humidity for patient care.

Temperature sensor-

A **temperature sensor** is a device that detects and measures temperature, converting it into a readable signal, typically in degrees Celsius or Fahrenheit. It operates by sensing changes in physical properties like electrical resistance or voltage in response to temperature variations.

There are various types of temperature sensors:

1. **Thermocouples**: Use two different metals to generate a voltage that changes with temperature.
2. **Resistance Temperature Detectors (RTDs)**: Measure temperature based on the change in electrical resistance of a material, usually platinum.
3. **Thermistors**: A type of resistor whose resistance changes significantly with temperature.
4. **Infrared Sensors**: Measure temperature by detecting the infrared radiation emitted by an object.

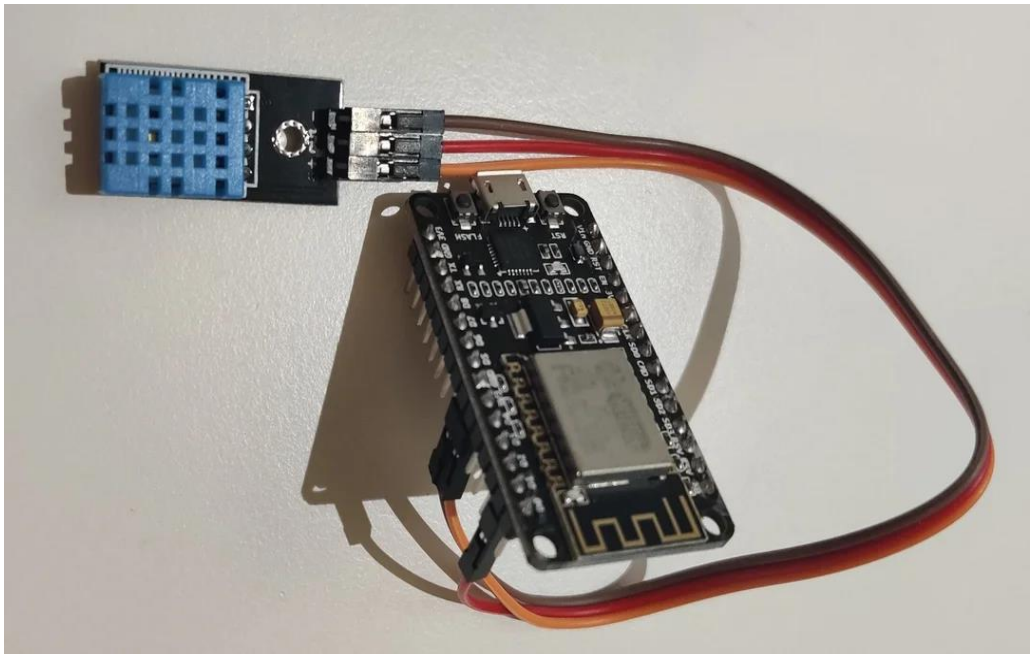
Temperature sensors are used in a wide range of applications across different industries. Here are some key examples:

1. **HVAC Systems**: Temperature sensors control heating, ventilation, and air conditioning systems to maintain comfortable indoor climates.
2. **Automotive Industry**: Used in engine management, climate control systems, battery monitoring in electric vehicles, and tire pressure monitoring to ensure safe operation.
3. **Industrial Processes**: Monitor and control temperature in manufacturing, chemical processing, and food production to ensure product quality and safety.
4. **Medical Devices**: Used in devices like thermometers, incubators, and patient monitoring systems to maintain and monitor body temperature for health and safety.

Code:-

```
#define BLYNK_TEMPLATE_ID "TMPL33uYbnMjM"
#define BLYNK_TEMPLATE_NAME "DHT11"
#define BLYNK_AUTH_TOKEN "m6z2u6JaQjjHhqQEepwqQG950otJKNib"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "DHT.h"
#define DHTTYPE DHT11
#define DHTPIN 4
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Green";
char pass[] = "1234567890";

float h;
float t;
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
WidgetLED led1(V3);

void sendSensor() {
  h = dht.readHumidity();
  t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
 Serial.println(t);
 Serial.println(h);
  Blynk.virtualWrite(V1, t);
  Blynk.virtualWrite(V2, h);
```

```
}
void setup() {
  // put your setup code here, to run once:

WiFi.begin(ssid, pass);
Blynk.begin(auth,ssid,pass);
Serial.begin(115200);
dht.begin();

timer.setInterval(1000L, sendSensor);
}

void loop() {
  // put your main code here, to run repeatedly:
   Blynk.run();
   timer.run();

}
```




humidity temperature
TEMPERATURE
29
0
100
HUMIDITY
45
0
100