



# Ganpat University

॥ विद्यया समाजोत्कर्षः ॥

Institute of  
Computer  
Technology

**Name: Tushar Panchal**

**En.No: 21162101014**

**Sub: IOT (Internet OF Things)**

**Branch: CBA**

**Batch:71**

## -----PRACTICAL 07-----

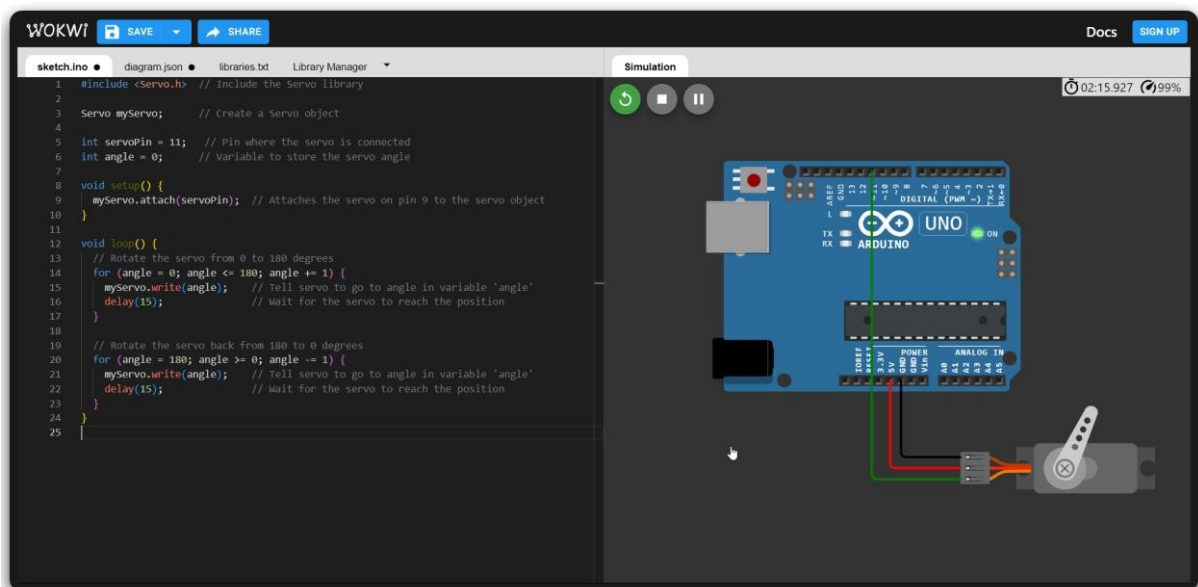
**Interface Servo Motor with Arduino & test it with ultrasonic sensor.**

**Parts needed :**

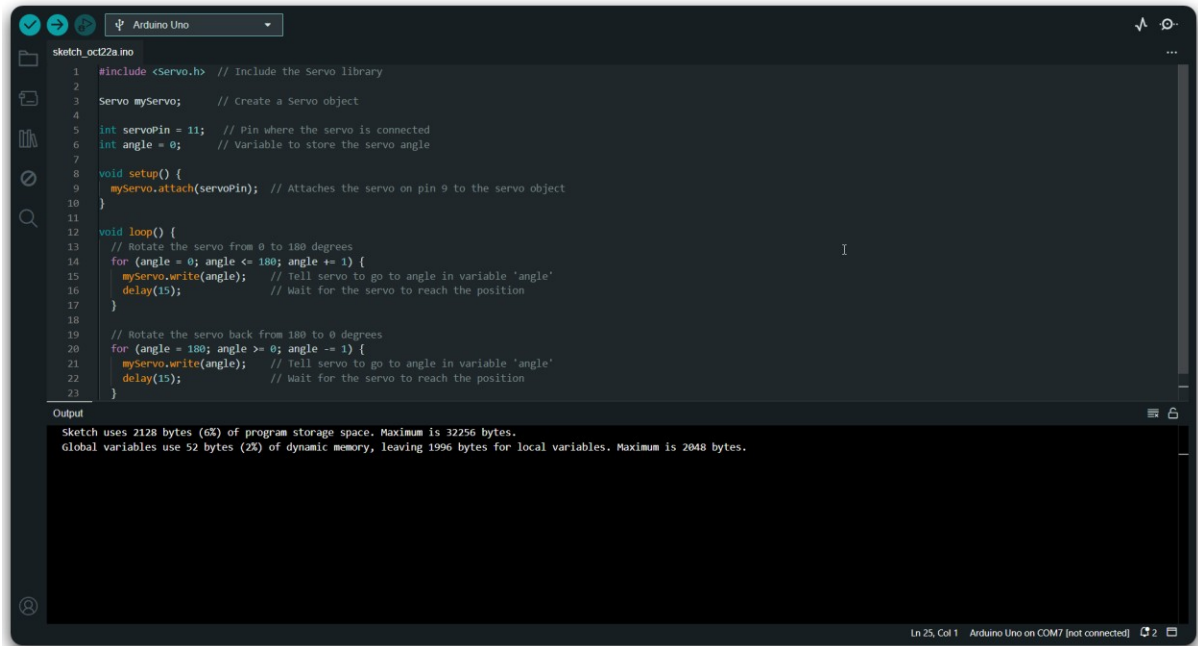
- 1) Arduino
- 2) Led
- 3) Servo Motor
- 4) Ultrasonic Sensor
- 5) Jumper wires

### Servo Motor With Arduino:

**Wokwi:**



# Arduino IDE:



The screenshot shows the Arduino IDE interface with a sketch named 'sketch\_oct22a.ino'. The code is as follows:

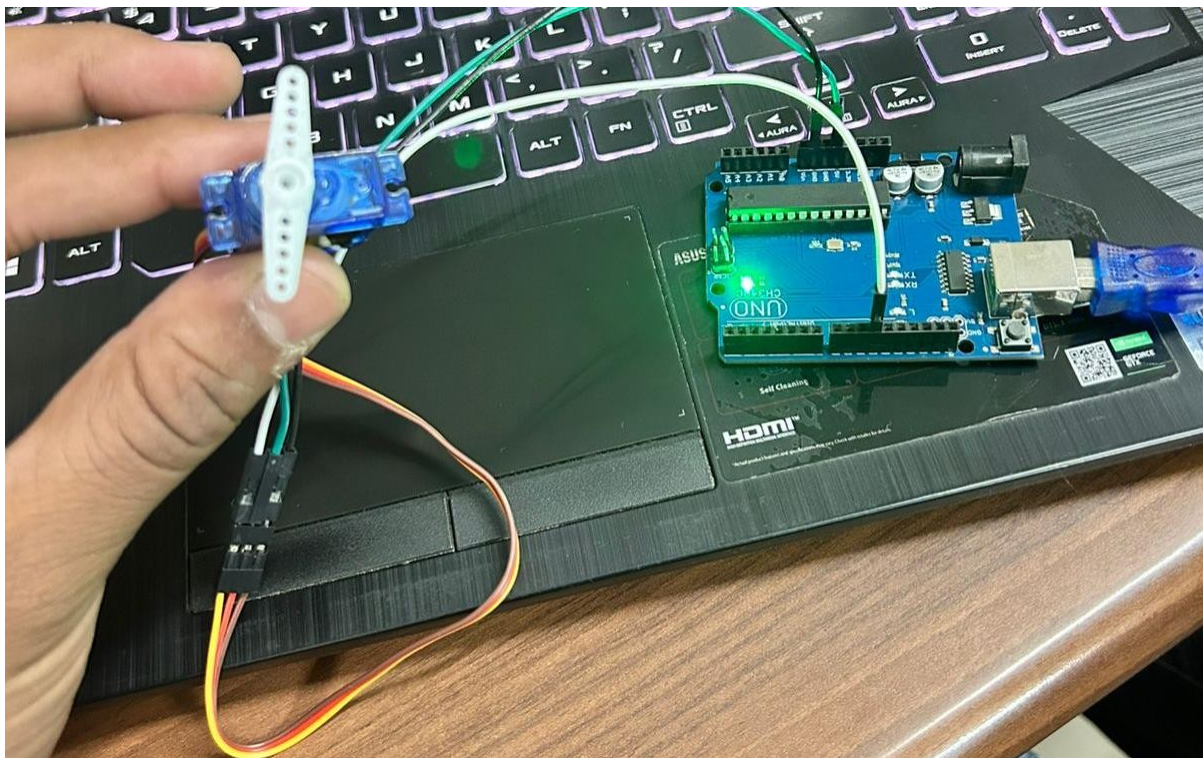
```
1 #include <Servo.h> // Include the Servo library
2
3 Servo myServo;      // Create a Servo object
4
5 int servoPin = 11;   // Pin where the servo is connected
6 int angle = 0;       // Variable to store the servo angle
7
8 void setup() {
9   myServo.attach(servoPin); // Attaches the servo on pin 9 to the servo object
10 }
11
12 void loop() {
13   // Rotate the servo from 0 to 180 degrees
14   for (angle = 0; angle <= 180; angle += 1) {
15     myServo.write(angle); // Tell servo to go to angle in variable 'angle'
16     delay(15);            // Wait for the servo to reach the position
17   }
18
19   // Rotate the servo back from 180 to 0 degrees
20   for (angle = 180; angle >= 0; angle -= 1) {
21     myServo.write(angle); // Tell servo to go to angle in variable 'angle'
22     delay(15);            // Wait for the servo to reach the position
23   }
24 }
```

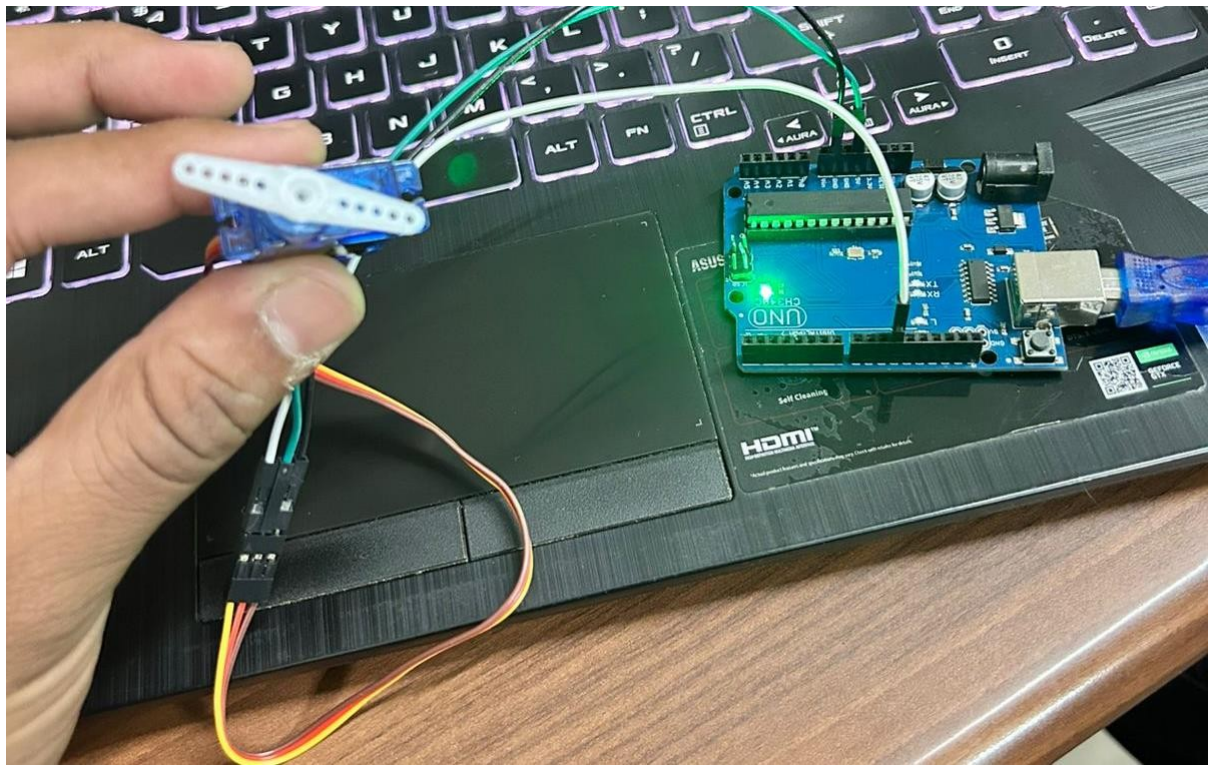
The Output window at the bottom shows the following message:

```
Sketch uses 2128 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 52 bytes (2%) of dynamic memory, leaving 1996 bytes for local variables. Maximum is 2048 bytes.
```

The status bar at the bottom right indicates 'Ln 25, Col 1 Arduino Uno on COM7 [not connected]'.

# OUTPUT:





# Ultrasonic sensor with Arduino:

## Wokwi:

The Wokwi simulation interface displays the following code in the sketch editor:

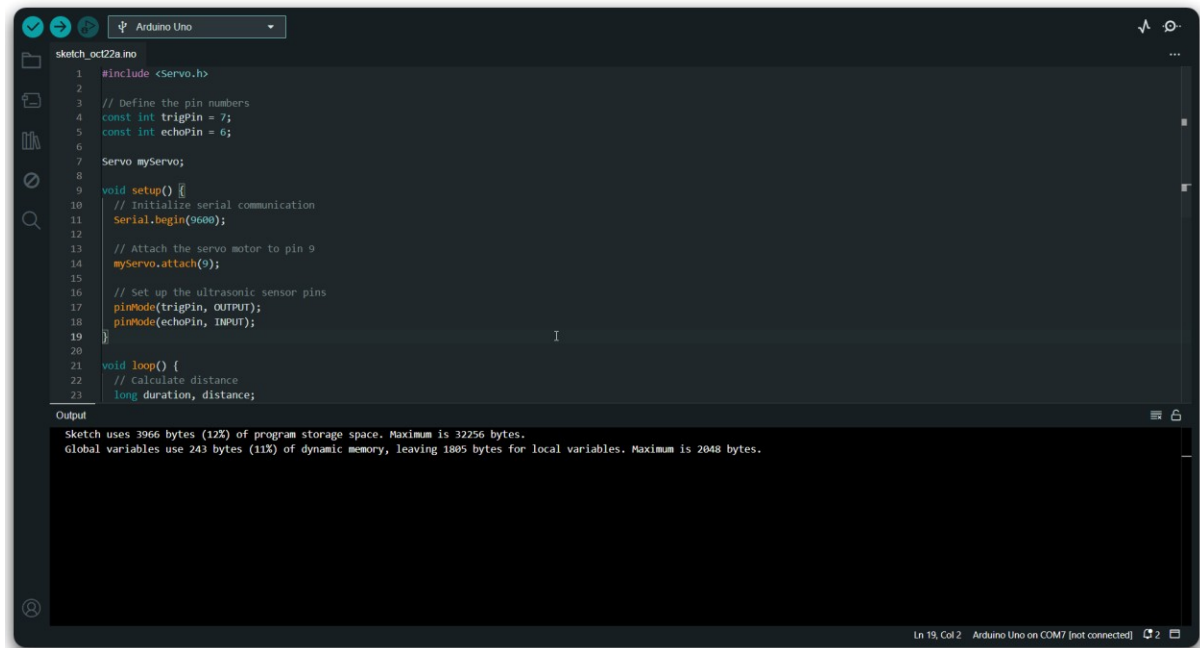
```
1 include <Servo.h> // Include the Servo library
2
3 / Define pins for ultrasonic sensor
4 const int trigPin = 7;
5 const int echoPin = 6;
6
7 servo myServo; // Create a Servo object
8
9 long duration;
10 int distance;
11 int servoPin = 11;
12
13 void setup() {
14   // Initialize serial communication for debugging
15   Serial.begin(9600);
16
17   // Set pin modes for ultrasonic sensor
18   pinMode(trigPin, OUTPUT);
19   pinMode(echoPin, INPUT);
20
21   // Attach the servo motor to the servo object
22   myServo.attach(servoPin);
23
24
25 void loop() {
26   // Generate a trigger pulse for 10 microseconds
27   digitalWrite(trigPin, LOW);
28   delayMicroseconds(2);
29   digitalWrite(trigPin, HIGH);
30   delayMicroseconds(10);
31   digitalWrite(trigPin, LOW);
32
33   // Measure the duration of the pulse on the echo pin
34   duration = pulseIn(echoPin, HIGH);
35 }
```

The simulation window shows the ultrasonic sensor reading a distance of 82cm. The sensor is connected to the Arduino Uno via a breadboard. The servo motor is connected to the Arduino Uno via a breadboard.

The Wokwi simulation interface displays the same code as the first image. The simulation window shows the ultrasonic sensor reading a distance of 235cm. The sensor is connected to the Arduino Uno via a breadboard. The servo motor is connected to the Arduino Uno via a breadboard.



## Arduino IDE:



The screenshot shows the Arduino IDE interface with a sketch named `sketch_oct22a.ino`. The code is as follows:

```
1 #include <Servo.h>
2
3 // Define the pin numbers
4 const int trigPin = 7;
5 const int echoPin = 6;
6
7 Servo myServo;
8
9 void setup() {
10   // Initialize serial communication
11   Serial.begin(9600);
12
13   // Attach the servo motor to pin 9
14   myServo.attach(9);
15
16   // Set up the ultrasonic sensor pins
17   pinMode(trigPin, OUTPUT);
18   pinMode(echoPin, INPUT);
19 }
20
21 void loop() {
22   // Calculate distance
23   long duration, distance;
```

The Output window at the bottom shows the following message:

```
Sketch uses 3966 bytes (12%) of program storage space. Maximum is 32256 bytes.
Global variables use 243 bytes (11%) of dynamic memory, leaving 1805 bytes for local variables. Maximum is 2048 bytes.
```

The status bar at the bottom right indicates: `Ln 19, Col 2 Arduino Uno on COM7 [not connected]`.

## Output:

