REPORT: OTTO ROBOT

Project Features:

1. **Modular Design**: OTTO robots are designed with modular components, allowing for easy assembly, customization, and upgrades. This design makes them highly adaptable for various educational and hobbyist projects.

2. **Open-Source Platform**: OTTO robots are built on an open-source platform, providing access to extensive documentation, software, and community support. This openness encourages innovation and collaboration among users.

3. **Interactive Capabilities**: Equipped with sensors and actuators, OTTO robots can interact with their environment. They can detect obstacles, follow lines, respond to sounds, and perform a variety of movements, making them highly interactive and engaging.

4. **Programmable with Multiple Languages** : Users can program OTTO robots using various programming languages, including Blockly, Scratch, and Python. This versatility makes them suitable for learners of different skill levels, from beginners to advanced programmers.

5. **Educational Focus:** OTTO robots are designed with an educational focus, providing a hands-on learning experience in STEM (Science, Technology, Engineering, and Mathematics). They help users develop skills in electronics, coding, and robotics through practical application.

Hardware Requirements:

1. Feviquick Adhesive
2. Screwdriver set
3. Mini USB Cable (A to B)
4. Power Adapter (9V, 1A)

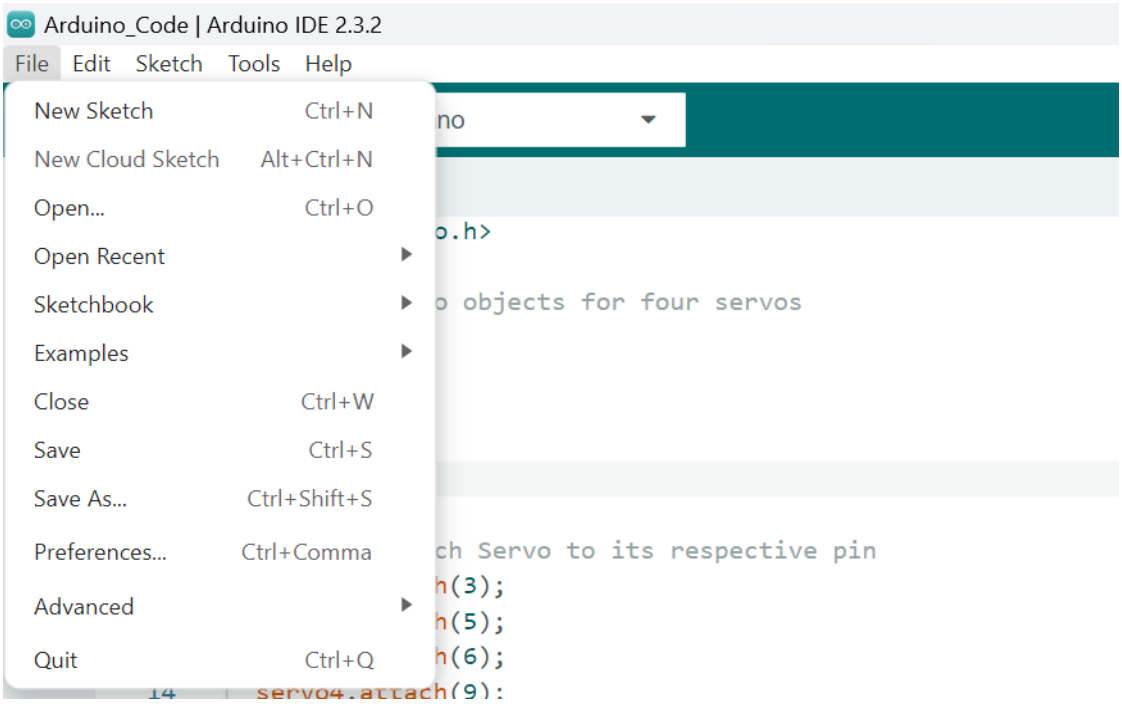
Software Requirements:

1. Arduino IDE

Assembly

1) Calibrate all servo motors at 90 Degree

a. Open **Arduino IDE**



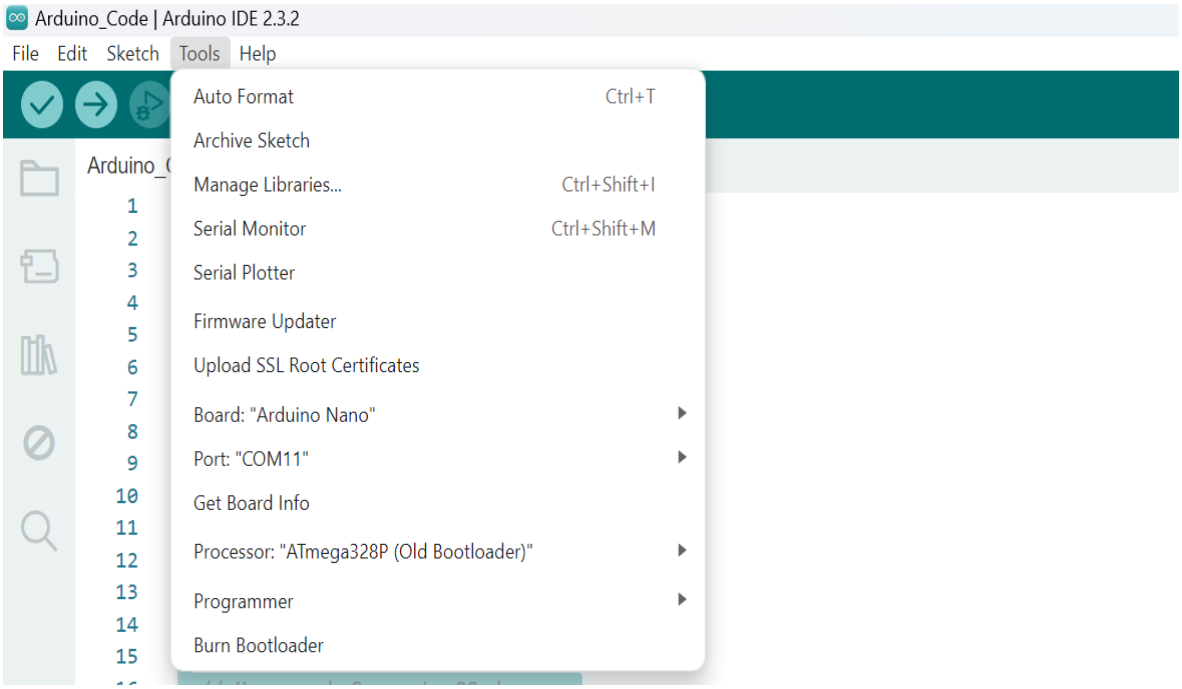
b. Now Click on >> File>>New Sketch

c. Now Paste the Servo Caliper Code

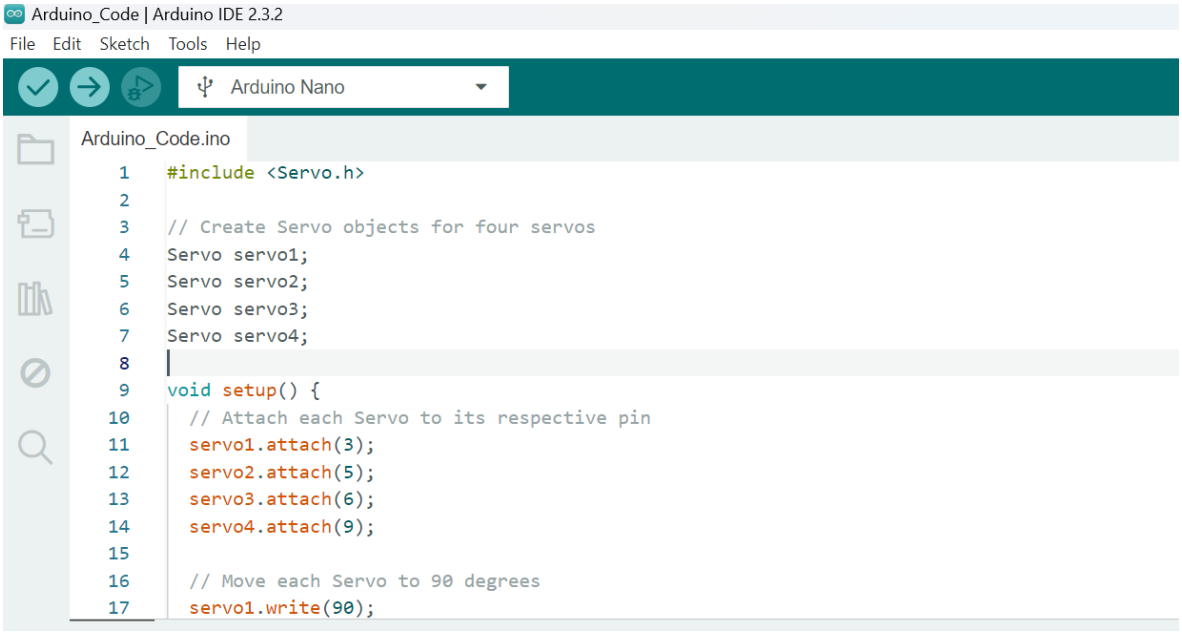
```
#include <Servo.h>

// Create Servo objects for four servos
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;

void setup() {
  // Attach each Servo to its respective pin
```

```
  servo1.attach(3);
  servo2.attach(5);
  servo3.attach(6);
  servo4.attach(9);

  // Move each Servo to 90 degrees
  servo1.write(90);
  servo2.write(90);
  servo3.write(90);
  servo4.write(90);
}

void loop() {
  // Nothing to do here
}
```

Now click on Tools>>Board>>Arduino Nano



d. Tools>>Port>> (Select Port)
e. Tools>> Processor>>ATmega328P
f. Now it's time to compile the code for compiling click on 



g. Connect Arduino Nano with nano cable and push the code, for push the code click on 

h. Once you push the code remove the cable and give the 9V 1A current to board by using adapter

i. Output>> All Servo motors are at 90 Degree.

2) Now Take the Otto Robot Head

3) Now fix the servo motor to the lower body of the otto robot

4) Attach the 2 servos motors to the otto robot below body and screw it.

5) Take the servo motor cap and cut according to the size of the leg.

   a. Servo motor Cap

   b. Mount like below

6) Do cutting (according to the the 3d part)

7) Mount servo motor like this, fix the screw and fix the foot to the leg.

8) attach the leg to the lower body.

9) Mount Ultrasonic sensor to the upper body of the otto robot.

10) Then Place the Expansion board with Expansion Board.

11) Make the connections as follows.

Servo Pins:

Left Leg: Pin 3

Right Leg: Pin 5

Left Foot: Pin 6

Right Foot: Pin 9

Buzzer: Pin 12

Ultrasonic Sensor:

VCC- VCC

Triger- 8

Echo Pin –7

GND – GND

12) Now The Final Otto Look like

Testing Steps:

1) Open the Arduino IDE and Create a New Sketch.
    a. Paste the below code

```
#include <Otto.h>
Otto Otto;

#define LeftLeg 3 // left leg pin, servo[0]
#define RightLeg 5 // right leg pin, servo[1]
#define LeftFoot 6// left foot pin, servo[2]
#define RightFoot 9 // right foot pin, servo[3]
#define Buzzer 12 //buzzer pin

long ultrasound_distance_simple() {
   long duration, distance;
   digitalWrite(8,LOW);
   delayMicroseconds(2);
   digitalWrite(8, HIGH);
   delayMicroseconds(10);
   digitalWrite(8, LOW);
   duration = pulseIn(7, HIGH);
   distance = duration/58;
   return distance;
}



void setup() {
  Serial.begin(9600); // Initialize serial communication
  randomSeed(analogRead(0)); // Seed the random number generator with an analog
pin's value
  Otto.init(LeftLeg, RightLeg, LeftFoot, RightFoot, true, Buzzer);
Otto.home();

  pinMode(8, OUTPUT); //trig
  pinMode(7, INPUT);//echo pin

}

void loop() {
  int randomNumber = random(0, 6); // Generate a random number between 1 and 10

  Serial.print("Random Number: ");
```

```
  Serial.println(randomNumber);

  switch (randomNumber) {
    case 1:
      // Do something for case 1
      Serial.println("Case 1: Action A");
       if (ultrasound_distance_simple() < 15 && ultrasound_distance_simple() !=0)
{
      Otto.moonwalker(4, 1500, 40, 1);
      Otto.walk(5,750,1); // FORWARD
      Otto.walk(5,750,-1);
      Otto.swing(2, 1000, 25);
      //Otto.jitter(2, 1000, 25);
      Otto.sing(S_superHappy);
      Otto.home();
    }
      break;

    case 2:
      // Do something for case 2
      Serial.println("Case 2: Action B");
       if (ultrasound_distance_simple() < 15 && ultrasound_distance_simple() !=0)
{
      Otto.turn(3,1000,1); //
      Otto.walk(5,750,1); // FORWARD
      Otto.walk(5,750,-2);
     // Otto.jitter(2, 1000, 25);
      Otto.ascendingTurn(5, 1000, 45);
      Otto.turn(3,750,-1); // LEFT
      Otto.sing(S_sleeping);
      Otto.home();
    }
    delay(500);
      break;

    case 3:
      // Do something for case 3
      Serial.println("Case 3: Action C");
       if (ultrasound_distance_simple() < 15 && ultrasound_distance_simple() !=0)
{
      Otto.walk(5,1000,1); // FORWARD

      Otto.sing(S_OhOoh2);
```

```
      Otto.walk(5,1000,-1); // BACKWARD
     // Otto.bend(1,1250,1);
      Otto.sing(S_OhOoh);
     // Otto.bend(1,1250,-1);

      Otto.home();
    }
      break;

    case 4:
      // Do something for case 4
      Serial.println("Case 4: Action D");
       if (ultrasound_distance_simple() < 15 && ultrasound_distance_simple() !=0)
{
      Otto.flapping(1, 1000, 25, 1);
      // Otto.flapping(1, 1000, 25, -1);
      Otto.sing(S_connection);
      Otto.swing(3, 1000, 25);

      Otto.updown(3, 1000, 25);
      Otto.tiptoeSwing(3, 1000, 25);
      Otto.sing(S_fart2);
      Otto.home();
    }
      break;

    case 5:
      // Do something for case 5
      Serial.println("Case 5: Action E");
       if (ultrasound_distance_simple() < 15 && ultrasound_distance_simple() !=0)
{
      Otto.walk(5,750,1); // FORWARD
      Otto.walk(5,750,-1); // BACKWARD
      Otto.sing(S_buttonPushed);
      Otto.moonwalker(2, 1500, 25, -1);
      delay(1000);
      Otto.swing(1, 1000, 25);
      Otto.sing(S_superHappy);
      //Otto.sing(PIRATES);
      Otto.home();
    }
  Otto.home();
  }
```

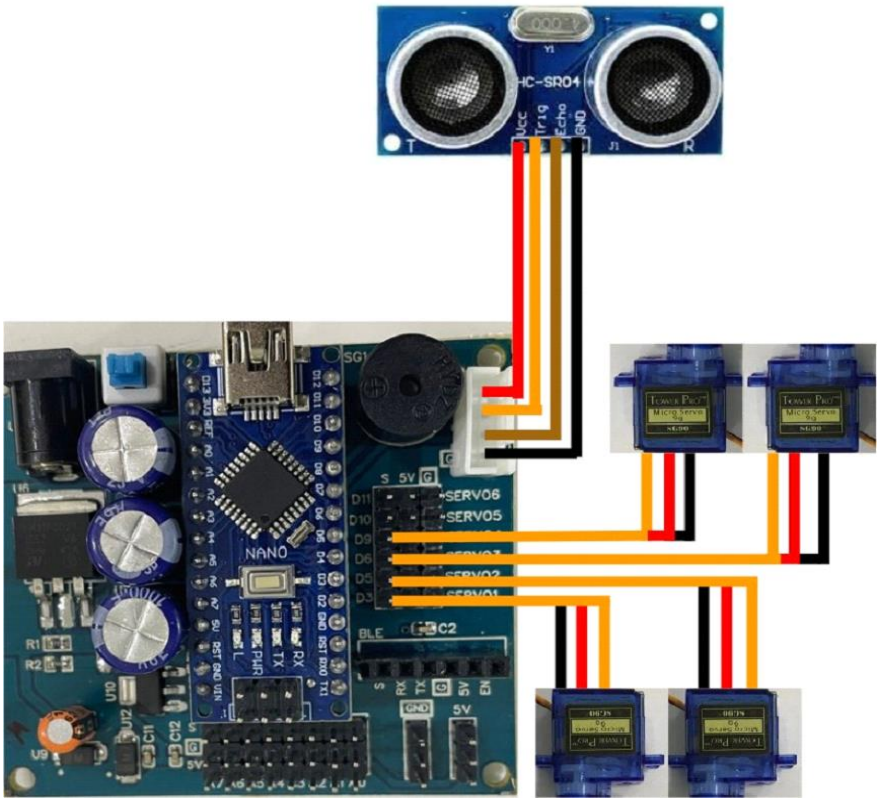
```
  delay(500); // Delay for 2 seconds before generating another random number
}
```

2) Now Compile the code by clicking on  this button.
3) Now Push the code by Clicking on 
4) Conclusion of Testing: The Otto robot is ready (doing the different moves).

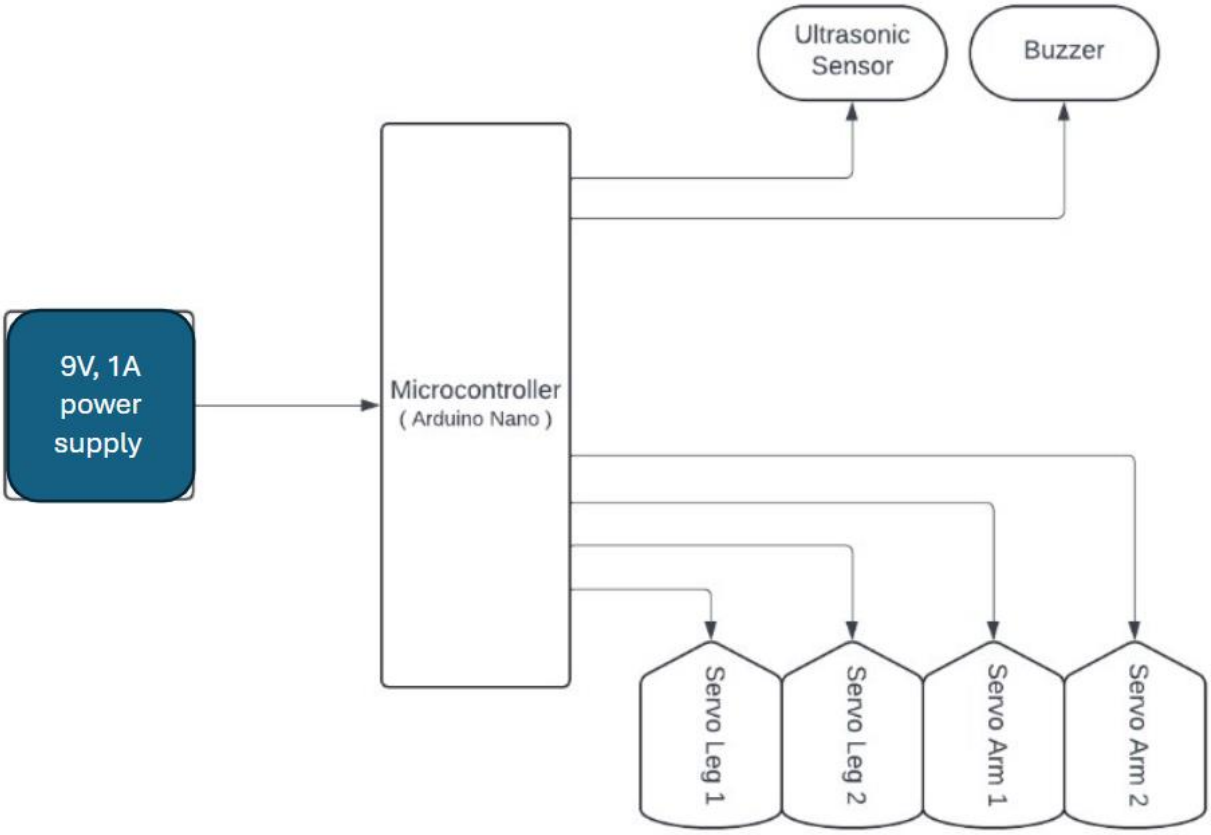
Test Video:

Click Here

Interfacing Diagram:



Block Diagram:

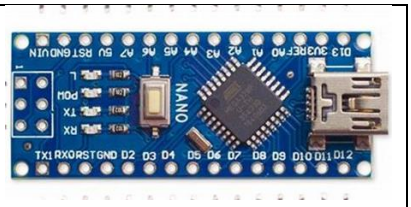
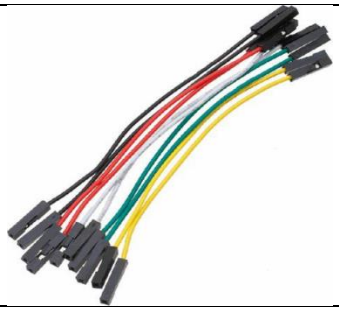
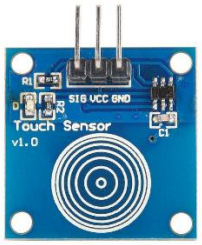


Bill Of Material (BOM)

SR No	Components	Specifications	Reference Links	Total QTY
1	3D Printed Body Parts		link	1
2	Arduino Nano	Operating Voltage (logic level): 5V **With Soldered Connector** 8 analog inputs ports: A0 ~ A7 14 Digital input / output ports: TX, RX, D2 ~ D13 Using Atmel Atmega328P-AU MCU	Link	1
3	Arduino Nano Expansion Board Customized	PHN Customized	Link	1
4	Ultrasonic Sensor - HC SR04	Operating Voltage: 5V DC Operating Current: 15mA	Link	1

		Measure Angle: 15° Ranging Distance: 2cm - 4m		
5	Servo Motors(MG 90), 180 Degree	Model: MG90S **Rotation 180 Degree** Operating voltage: 4.8V~ 6.6V Servo Plug: JR Stall torque @4.8V : 1.8kg-cm Stall torque @6.6V : 2.2kg-cm"	Link	4
6	Double End 4 Pin XH JST Female to Female Wire Connector	XH series double end female JST connector Number of holes : 4	Link	1
7	M2*8 Self Tapping Screw	M 2/8 Self Tapping Screw	Link	2
8	Fewiquick(5Rs)	Brand Pidilite	Link	1
9	Bluetooth Module - HC-05	Bases at CSR BC04 Bluetooth technology. with build-in 2.4GHz PCB antenna It's at the Bluetooth class 2 power level. Range test: 10 meters Operating voltage: 3.3V to 6V DC Operating current in pairing is in the range of 30~40mA.	Link	1
10	Jumper cables(F-F)			6
11	Touch Sensor - TTP223B module	Power supply voltage(VCC): 2.0, 3, 5.5 V. Output high VOH: 0.8VCC V Output low VOL: 0.3VCC V Response time (touch mode) : 60 mS Response time (low power mode) : 220 mS	Link	1
12	Jumper cables(M-F)		Link	3

Image Directory

		
HC-SR04 Ultrasonic Sensor * 1	**MG90 Servo Motor*4**	**Arduino Nano*1**
		
PHN Customized Nano Expansion Board*1	**Female To Female Jumper Wires*6**	**OTTO Robot 3D Printed Parts*6**
		
Bluetooth Module HC-05*1	**Touch Sensor*1**	**4 Pin JST Cable (Both Side) *1**
		
M2/8mm Self Taping *2 Screws	**Male To Female Wires*3**	**Fevikwik Adhesive*1**

FAQ / Troubleshooting Instructions:

1. Calibrate All Servo Motors To 90 Degrees Before Installing.
2. Connect The Arduino Nano Using the USB Mini Cable.
3. Install All Necessary Libraries Like Servo.h From Library Manager.
4. Double Check Connections before plug in the power, to Avoid Short Circuit.