Toll Tax System using Arduino: Ultrasonic Sensor with Servo Motor

By- Mayank Patel

The **Toll Tax System using Arduino** is an automated mechanism designed to streamline toll operations using an **ultrasonic sensor** and a **servo motor**. At its core, an **Arduino Uno** is the central microcontroller, coordinating all system functions. The **ultrasonic sensor** detects approaching vehicles by emitting sound waves and measuring the time it takes for the echoes to return. Once a vehicle is detected within a predefined range, the Arduino triggers the **servo motor** to lift the toll barrier, allowing the vehicle to pass. After the vehicle clears the sensor's range, the servo motor automatically lowers the barrier to its default position. The setup includes a breadboard for electrical connections and is powered by a battery. The road and barrier are represented using a simple prototype to simulate a toll plaza. This system eliminates the need for manual toll operations, improving efficiency and reducing traffic congestion. It also lays the foundation for further enhancements, such as integrating RFID or payment modules, to create a fully automated toll collection system.
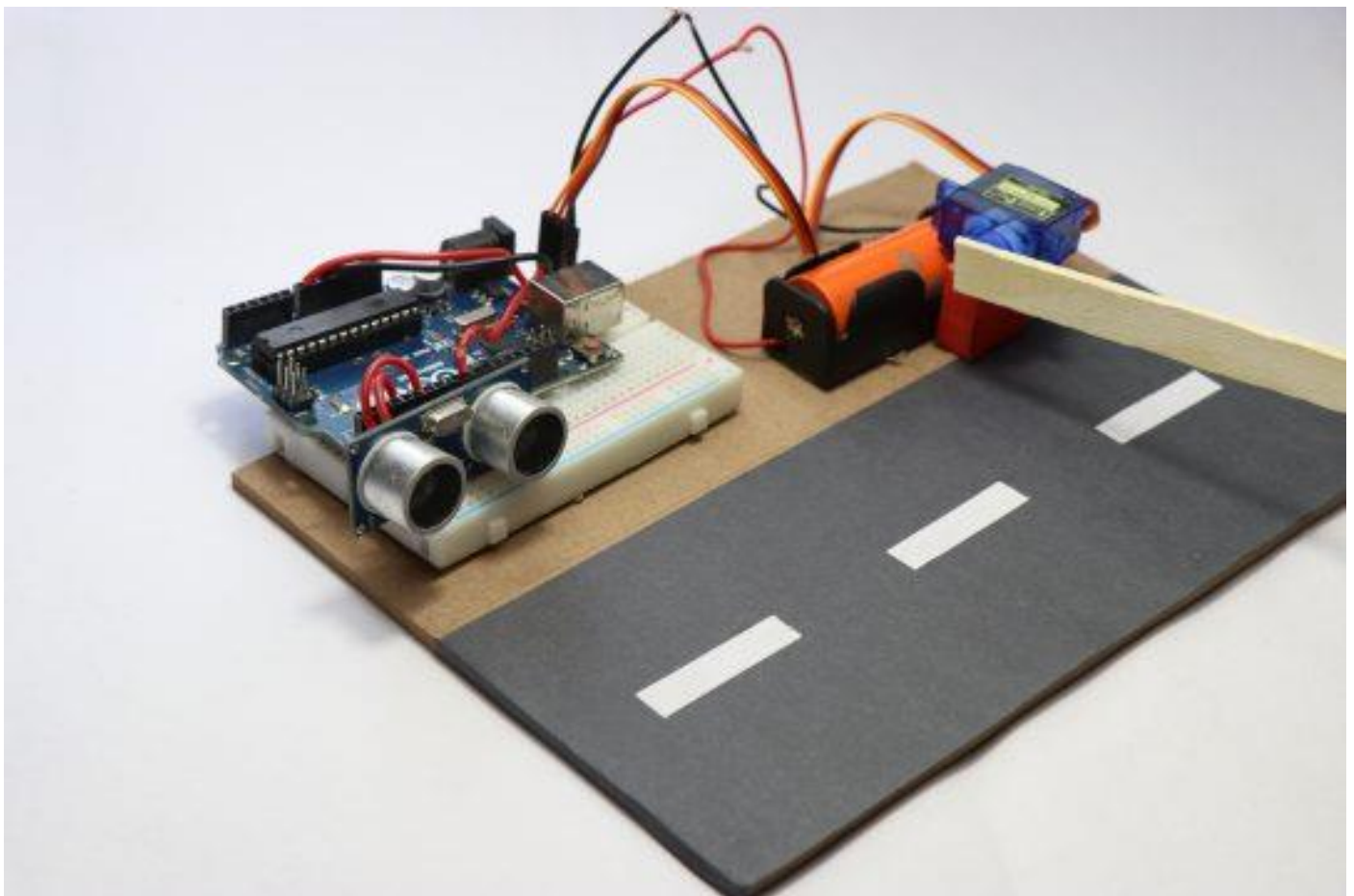
Material Required

1. Arduino Uno

Arduino UNO is a microcontroller that offers 14 digital input/output pins. It uses ATmega328P ic for performing various given tasks. We can write our code in the Arduino IDE software and then upload it to the Arduino board. There is a USB port in the Arduino through which we can power it or program it. You can also connect a DC adapter of 5 to 9 volts to power the board
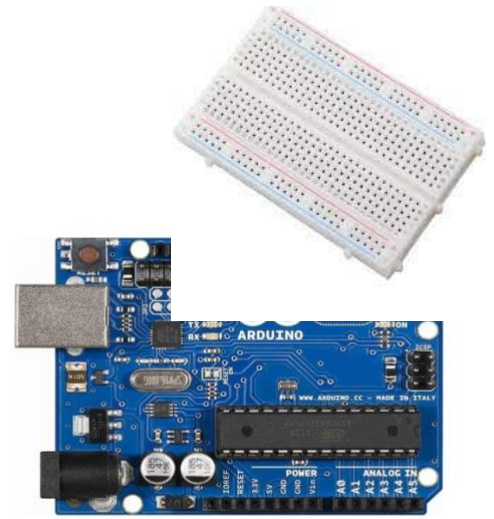
2. Ultrasonic sensor

This is an ultrasonic sensor that releases ultrasonic waves from one end and receives it from the other end. We can calculate the distance between two marks by calculating the time and converting it. This is like the principle of a SONAR mechanism.

3. Plastic-geared Micro servo

A Servo motor is a special type of motor available in both AC/DC but in this project, we are using a DC servo motor. We can control the range of shafts in a servo motor.

4. Single-strand wire instead of jumper wires

Jumper wires are used for making connections between two pins, and ports. These are of three types, female to female jumper wires, male to male jumper wires, and male-to-female jumper wires.

5. Mini Breadboard

A breadboard is a mounting board that has so many slots for making the connections.
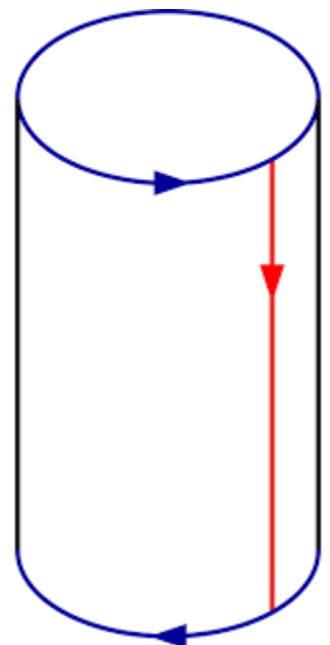
6. Arduino programming cable

Connect this cable to the USB port in your PC or laptop and then plug it into the Arduino such that the power LED on the Arduino goes on.
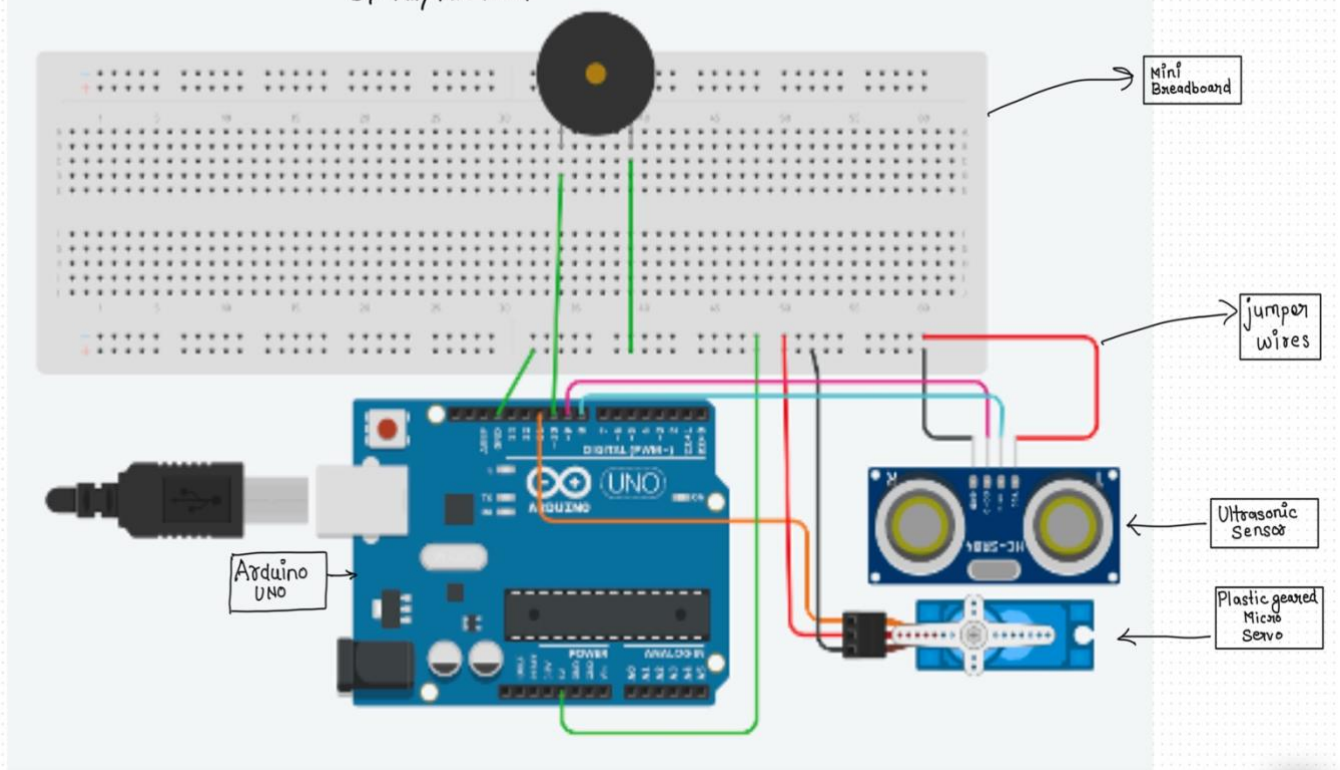
7. Arduino IDE

8. A Piece of paper stick

CIRCUIT DIAGRAM

Circuit Diagram of Toll System:-
By- Mayank Patel

- Above is the circuit diagram for the automatic toll plaza project.
- You need to make all the connections according to it.
- First of all, take a servo motor and connect its positive wire with the 5-volt pin of the Arduino.
- Then join the negative wire of the servo motor with the GND pin of the Arduino.
- Make sure that all the connections are correct and tight.
- Attach the signal wire of the servo motor with the digital- pin of the Arduino.
- After that take the ultrasonic sensor, you can see the names of the pins mentioned on the surface of the sensor. So read them carefully and connect the VCC pin of the sensor with the 5-volt pin of the Arduino.
- Then connect the GND pin of the ultrasonic sensor with the GND pin of the Arduino.
- Notice that these two pins are for the power and for sending the data we have another two pins left in the sensor. Join the TRIG pin of the ultrasonic sensor with the digital- pin of the Arduino.
- Attach the ECHO pin of the ultrasonic sensor with the digital- pin of the Arduino.
- All the connections are complete now and you can proceed with the next steps.

CODE Uses

```cpp
#include<Servo.h>
int trig=8;
int echo=9;
int buzz=10;
int dt=10;
Servo servo;

//int distance,duration;
void setup() {
  // put your setup code here, to run once:
pinMode(trig,OUTPUT);
pinMode(buzz,OUTPUT);
pinMode(echo,INPUT);
Serial.begin(9600);
servo.attach(11);
}

void loop() {
  // put your main code here, to run repeatedly:
if (calc_dis()<15)
{
  digitalWrite(buzz,HIGH);
}
else
{
  digitalWrite(buzz,LOW);
}

if (calc_dis()<10)
{
 for (int i=0;i<=540;i++)
 {
   servo.write(i);
   delay(1);
 }
 delay(100);
 for (int i=540;i>=0;i--)
 {
   servo.write(i);
   delay(1);
 }
   delay(100);
}
```

```
}

int calc_dis()
{
  int duration,distance;
  digitalWrite(trig,HIGH);
  delay(dt);
  digitalWrite(trig,LOW);
  duration=pulseIn(echo,HIGH);
  distance = (duration/2) / 29.1;
  return distance;
}
```