

ARTIFICIAL INTELLIGENCE PROJECT

TRAFFIC CONTROL SYSTEM USING ARDUINO AND SENSORS

TEAM MEMBERS:

LAKSHYA TYAGI (17SCSE101490) – CODING AND WIRING

RISHAV SHARMA (17SCSE101062) – STRUCTURE AND WIRING

FAISAL ANWAR (17SCSE101496) – DOCUMENTATION

VIKAS KUMAR (17SCSE101429) – DOCUMENTATION AND STRUCTURE

AIM

To reduce the traffic police efforts using an artificial intelligence technology with the help of Microcontroller, Sensors and coding. That's all make it a fully automatic traffic controlled system.

TRAFFIC CONTROL SYSTEM

COMPONENTS USED

1. ARDUINO UNO MICROCONTROLLER
2. ULTRASONIC SENSOR (4 PINS)
3. SERVO MOTOR 9G (3 PINS)
4. JUMPER WIRE MALE TO MALE
5. JUMPER WIRE MALE TO FEMALE
6. BREADBOARD
7. CARDBOARD FOR DUMMY

TRAFFIC CONTROL SYSTEM

BRIEF DESCRIPTION ABOUT PARTS

1. ARDUINO UNO REVISION 3

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz



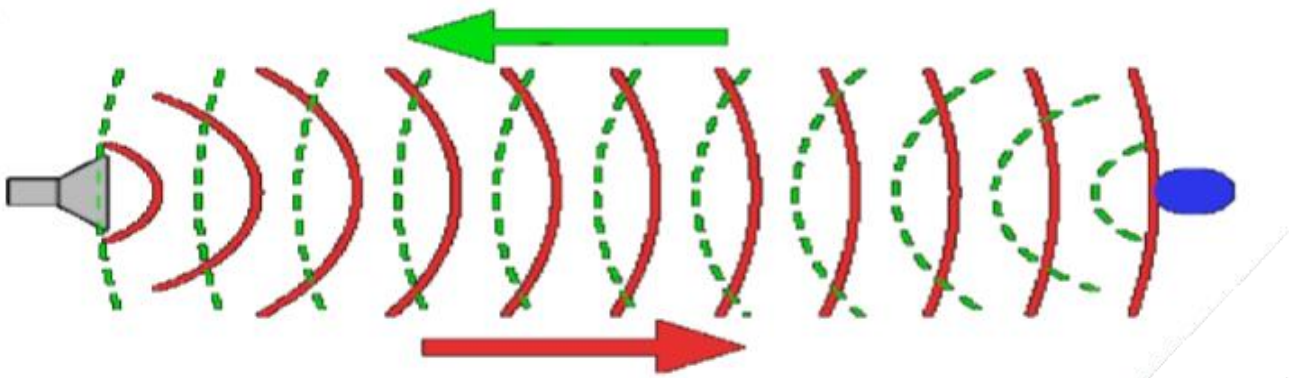
2. Ultrasonic Sensor (4 pins)

The Ultrasonic Sensor sends out a high-frequency sound pulse and then times how long it takes for the echo of the sound to reflect back. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, (like a tiny speaker), the other receives them, (like a tiny microphone).



The speed of sound is approximately 341 meters (1100 feet) per second in air. The ultrasonic sensor uses this information along with the time difference between sending and receiving the sound pulse to determine the distance to an object. It uses the following mathematical equation:

$$\text{Distance} = \text{Time} \times \text{Speed of Sound} / 2$$



The ultrasonic sensor can measure the distances in centimetres and inches. It can measure from 0 to 2.5 meters with a precision of 3 cm. It has more difficulties reading reflections from soft, curved thin or small objects like cotton, curved surface, etc.

3. Servo motor 9g (3 pins)

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback.



4. Jumper Wire (MALE TO MALE AND MALE TO FEMALE)

Used to make connections quickly and efficiently.



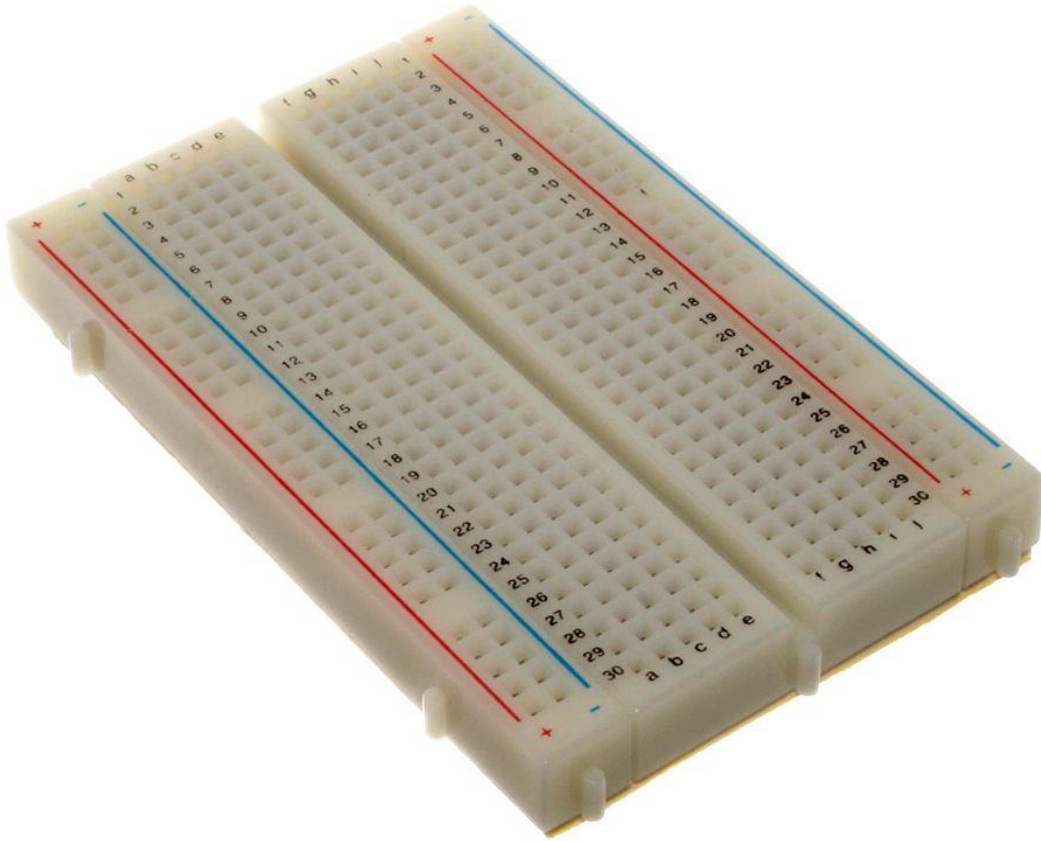
Male to Male Jumper wire



Male to Female Jumper wire

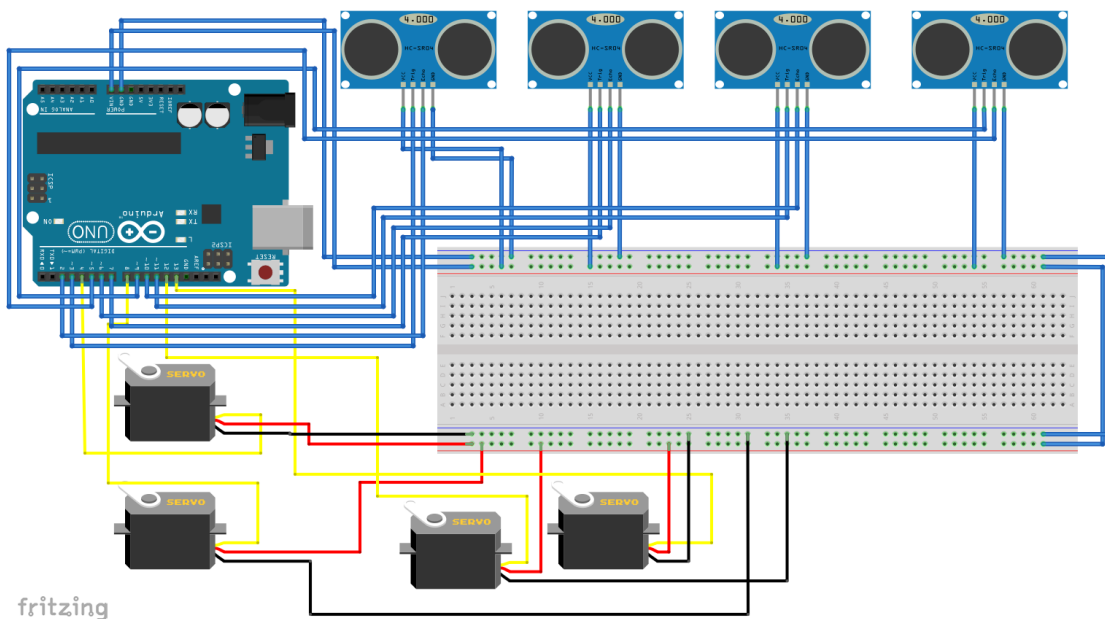
5. Bread Board

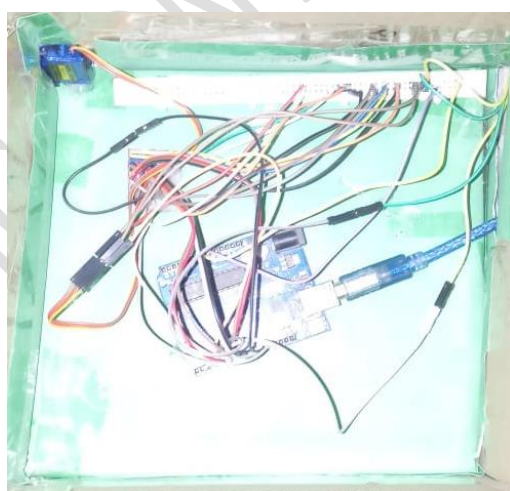
Used for easy and multiple connections



TRAFFIC

CIRCUIT DIAGRAM and Project Images





Code use in c++

```
#include <Servo.h>

Servo ServoMotor0;
Servo ServoMotor1;
Servo ServoMotor2;
Servo ServoMotor3;

int flag = 0, distance0, distance1, distance2, distance3, time0, time1, time2, time3,
duration0, duration1, duration2, duration3;

void setup() {
    Serial.begin(9600);
    pinMode(2, INPUT); //echo
    pinMode(3, OUTPUT); //trig
    ServoMotor0.attach(4);
    pinMode(6, INPUT); //echo
    pinMode(7, OUTPUT); //trig
    ServoMotor1.attach(8);
    pinMode(10, INPUT); //echo
    pinMode(11, OUTPUT); //trig
    ServoMotor2.attach(12);
    pinMode(5, INPUT); //echo
    pinMode(9, OUTPUT); //trig
    ServoMotor3.attach(13);
}

void loop() {
    if(flag >= 0 && flag <= 3)
    {
        digitalWrite(3, LOW);
        delayMicroseconds(2);
        digitalWrite(3, HIGH);
```

```
delayMicroseconds(10);
digitalWrite(2, LOW);
duration0=pulseIn(2, HIGH);
distance0=(duration0/2)/29.1;
Serial.println(distance0);
delay(500);
if(flag==0)
{
    ServoMotor0.write(0);
    ServoMotor1.write(90);
    ServoMotor2.write(90);
    ServoMotor3.write(90);
    flag=flag+1;
    Serial.println("Motor 0");
    Serial.println("Time Duration: ");
    if(distance0>(19-0) && distance0<=(19-5))
    {
        for(int i=0; i<15; i++)
        {
            Serial.println(i+1);
            delay(1000);
            digitalWrite(3, LOW);
            digitalWrite(3, HIGH);
            digitalWrite(3, LOW);
        }
    }
    else if(distance0>(19-5) && distance0<=(19-10))
    {
        for(int i=0; i<20; i++)
```

```
{
    Serial.println(i+1);
    delay(1000);
    digitalWrite(3, LOW);
    digitalWrite(3, HIGH);
    digitalWrite(3, LOW);
}
}
else if(distance0>(19-10) && distance0<=(19-15)
{
    for(int i=0; i<25; i++)
    {
        Serial.println(i+1);
        delay(1000);
        digitalWrite(3, LOW);
        digitalWrite(3, HIGH);
        digitalWrite(3, LOW);
    }
}
else
{
    for(int i=0; i<5; i++)
    {
        Serial.println(i+1);
        delay(1000);
        digitalWrite(3, LOW);
        digitalWrite(3, HIGH);
        digitalWrite(3, LOW);
    }
}
```

```

}

}

else if(flag==1)
{
    ServoMotor0.write(90);
    ServoMotor1.write(0);
    ServoMotor2.write(90);
    ServoMotor3.write(90);
    digitalWrite(8, LOW);
    delayMicroseconds(2);
    digitalWrite(8, HIGH);
    delayMicroseconds(10);
    digitalWrite(6, LOW);
    duration1=pulseIn(6, HIGH);
    distance1=(duration1/2)/29.1;
    Serial.println(distance1);
    delay(500);
    flag=flag+1;
    Serial.println("Motor 1");
    if(distance1>(19-0) && distance1<=(19-5))
    {
        for(int i=0; i<15; i++)
        {
            Serial.println(i+1);
            delay(1000);
            digitalWrite(7, LOW);
            digitalWrite(7, HIGH);
            digitalWrite(7, LOW);
        }
    }
}

```

```
}  
  
}  
else if(distance1>(19-5) && distance1<=(19-10))  
{  
    for(int i=0; i<20; i++)  
    {  
        Serial.println(i+1);  
        delay(1000);  
        digitalWrite(7, LOW);  
        digitalWrite(7, HIGH);  
        digitalWrite(7, LOW);  
    }  
}  
else if(distance1>(19-10) && distance1<=(19-15))  
{  
    for(int i=0; i<25; i++)  
    {  
        Serial.println(i+1);  
        delay(1000);  
        digitalWrite(7, LOW);  
        digitalWrite(7, HIGH);  
        digitalWrite(7, LOW);  
    }  
}  
else  
{  
    for(int i=0; i<5; i++)  
    {  
        Serial.println(i+1);
```

```
    delay(1000);
    digitalWrite(7, LOW);
    digitalWrite(7, HIGH);
    digitalWrite(7, LOW);
  }
}
}
else if(flag==2)
{
    ServoMotor0.write(90);
    ServoMotor1.write(90);
    ServoMotor2.write(0);
    ServoMotor3.write(90);
    digitalWrite(11, LOW);
    delayMicroseconds(2);
    digitalWrite(11, HIGH);
    delayMicroseconds(10);
    digitalWrite(10, LOW);
    duration2=pulseIn(10, HIGH);
    distance2=(duration2/2)/29.1;
    Serial.println(distance2);
    delay(500);
    flag=flag+1;
    Serial.println("Motor 2");
    if(distance2>(19-0) && distance2<=(19-5))
    {
        for(int i=0; i<15; i++)
        {
            Serial.println(i+1);
```



```
    delay(1000);
    digitalWrite(11, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(11, LOW);
}
}
else if(distance2>(19-5) && distance2<=(19-10))
{
    for(int i=0; i<20; i++)
    {
        Serial.println(i+1);
        delay(1000);
        digitalWrite(11, LOW);
        digitalWrite(11, HIGH);
        digitalWrite(11, LOW);
    }
}
else if(distance2>(19-10) && distance2<=(19-15))
{
    for(int i=0; i<25; i++)
    {
        Serial.println(i+1);
        delay(1000);
        digitalWrite(11, LOW);
        digitalWrite(11, HIGH);
        digitalWrite(11, LOW);
    }
}
else
```

```

{
  for(int i=0; i<5; i++)
  {
    Serial.println(i+1);
    delay(1000);
    digitalWrite(11, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(11, LOW);
  }
}
}
else if(flag==3)
{
  ServoMotor0.write(90);
  ServoMotor1.write(90);
  ServoMotor2.write(90);
  ServoMotor3.write(0);
  digitalWrite(9, LOW);
  delayMicroseconds(2);
  digitalWrite(9, HIGH);
  delayMicroseconds(10);
  digitalWrite(5, LOW);
  duration3=pulseIn(5, HIGH);
  distance3=(duration3/2)/29.1;
  Serial.println(distance3);
  delay(500);
  flag=flag+1;
  Serial.println("Motor 3");
  if(distance3>(19-0) && distance3<=(19-5))

```

```
{  
  for(int i=0; i<15; i++)  
  {  
    Serial.println(i+1);  
    delay(1000);  
    digitalWrite(9, LOW);  
    digitalWrite(9, HIGH);  
    digitalWrite(9, LOW);  
  }  
}  
else if(distance3>(19-5) && distance3<=(19-10))  
{  
  for(int i=0; i<20; i++)  
  {  
    Serial.println(i+1);  
    delay(1000);  
    digitalWrite(9, LOW);  
    digitalWrite(9, HIGH);  
    digitalWrite(9, LOW);  
  }  
}  
else if(distance3>(19-10) && distance3<=(19-15))  
{  
  for(int i=0; i<25; i++)  
  {  
    Serial.println(i+1);  
    delay(1000);  
    digitalWrite(9, LOW);  
    digitalWrite(9, HIGH);
```

```
    digitalWrite(9, LOW);  
    }  
    }  
    else  
    {  
        for(int i=0; i<5; i++)  
        {  
            Serial.println(i+1);  
            delay(1000);  
            digitalWrite(9, LOW);  
            digitalWrite(9, HIGH);  
            digitalWrite(9, LOW);  
        }  
    }  
    }  
    }  
    else  
    {  
        flag=0;  
    }  
}
```

Working of Project

It is a fully automatic traffic prediction system. First of all the we will name the Road by Road 1, Road 2, Road 3 & Road 4. If the Road 1 is on then at that time the other 3 Roads (Road 2, Road 3 & Road 4) will be barricaded. Now the Ultrasonic sensor will calculate the traffic of Road 2 and according to the traffic it will allot the time to Road 2 and when Road 1 time is done then automatically it will barricade the Road 1 and Road 2 barricade will be on and then it will calculate the traffic of Road 3 and accordingly the process will go on. According to this process we can save time.

TRAFFIC CONTROL SYSTEM