AN IOT PROJECT ON

BLIND SPOT DETECTION

ANEESH ANNE

PRAKASH CHANDRA NAYAK

PARDHA SARADHI RAMINENI

DURGA SAI RUCHITHA BATHULA

Abstract:

Now-a-days safety has become a major concern due to the increase in the number of road accidents. Most of the accidents occur due to the negligence of the drivers or may be due to the lack of detecting obstacles in their blind spot while changing the lanes and directions. To reduce these mishaps blind spot detection is the best solution till date. Even government has made it compulsory that any car that is launched in 2021 should be equipped with blind spot detection system. Companies like Mercedes, BMW, Tesla any many others are trying to build their own form of technology in this system. It has been one of the biggest evolutions with respect to safety in the automobile industry. Blind spot detection using IOT can be a great revolution for the wireless technologies in the car. As IOT is one of the fast-growing technologies in the recent times. So, collaborating IoT and the safety features of the car helps in boosting up the technological features of the product.

Introduction:

In our project we are detecting the blind spots with the help of ultrasonic sensors. These sensors detect the obstacles or any vehicles behind that are in the blind spot and sends the information to Arduino to which LED's are connected. So, whenever there is an obstacle the LED's glow by alarming the driver. This system helps in detecting any obstacle while changing the lanes and while any vehicle is overtaking our vehicle. LED's are placed on the side mirrors of the car so that their light reflects in the mirror showing at which side the obstacle or the vehicle is present.

Blind Spot Detection:

It is a collision avoidance system that helps in enhancing the safety of the automobile by reducing the probability of the road accidents. Blind spot of a vehicle is the area that cannot be seen through the rear or side mirrors. This system generally uses the ultra-sonic sensors. The sensor detects the obstacle and sends the data that is used to alert the driver.

Hardware and Software Requirements:

Arduino Uno

HC-SRO4 Ultrasonic Sensor

Resistors

LED

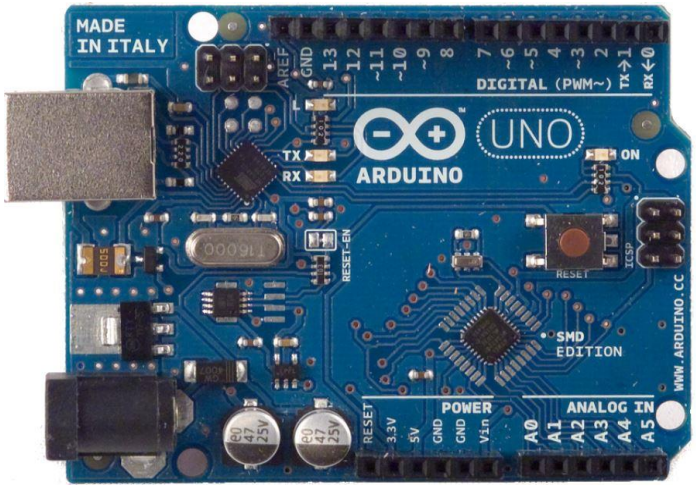
Connecting wires

Arduino IDE

Arduino Uno:

It is a microcontroller board developed by aurdino.cc that is based on Atmega328. In the recent times electronics are becoming cheaper, compact, and flexible thereby increasing their performance capabilities unlike the previous ones. Microcontrollers are introduced to make our work easy by consuming the minimum effort and producing the maximum output. There are many versions of Arduino boards like Arduino Uno, Arduino Mega, Arduino Nano, Arduino Leo, etc. Arduino uno and Arduino Mega are the most common microcontrollers used.

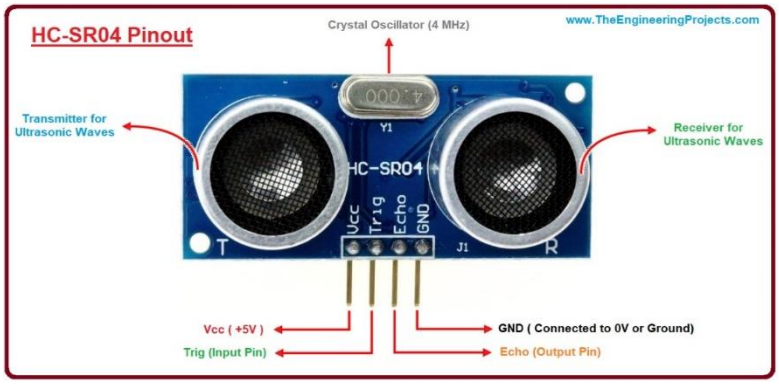
So, in our project we are using Arduino Uno as a microcontroller which collects the data from the sensor and send it to the output i.e., LED.



HC-SRO4 Ultrasonic Sensor:

It is a distance sensor which helps in detecting any obstacle within a range of 2cm to 400cm (usually up to 13 feet) thereby providing an accuracy of 3mm. This sensor generally operates on 5v. Since Arduino also operates on 5v we can directly connect the sensor to the Arduino or any other 5v microcontrollers.

Ultrasonic sensors consist of four pins namely trigger pin, echo pin, ground and vcc. Trigger pin is used to transmit the waves and when an obstacle is detected these waves are reflected. These reflected waves are received at the Echo pin.

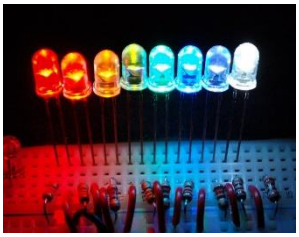


LED and Resistor:

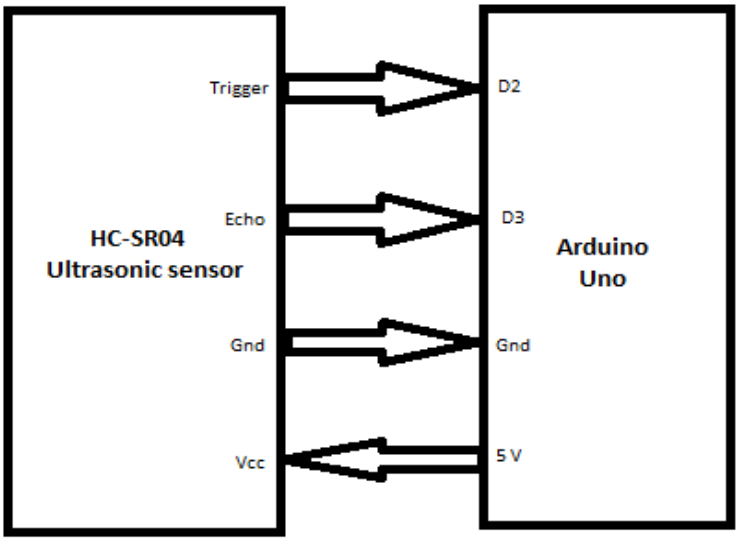
LED (light emitting diode) is a semiconductor device that emits light when current flows through it.

Resistor is an electrical component that is used to reduce the current flow.

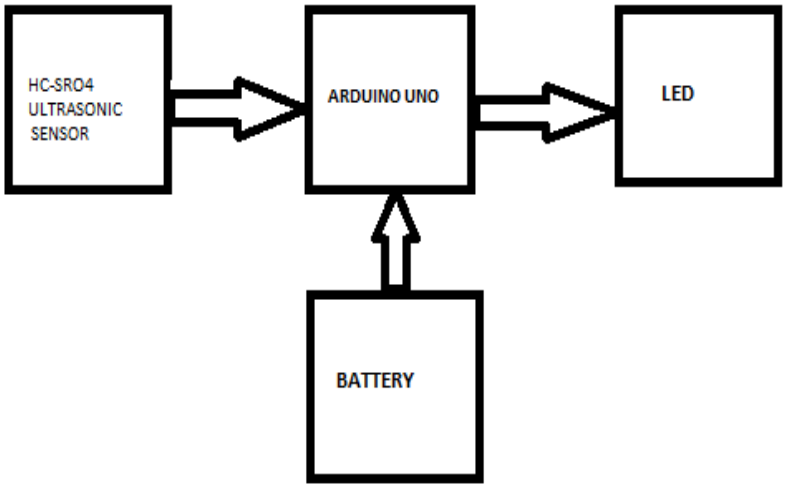
Here we are using LED as the output source that glows whenever the Arduino sends the data collected from the sensor about the obstacle. But the LED require a current of 3.3v to emit light but as we know that Arduino operates on 5v we are using a resistor of 220 ohms to reduce the current.



Pin Diagram:

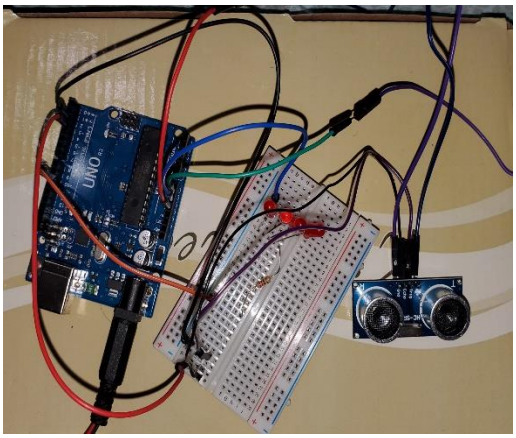


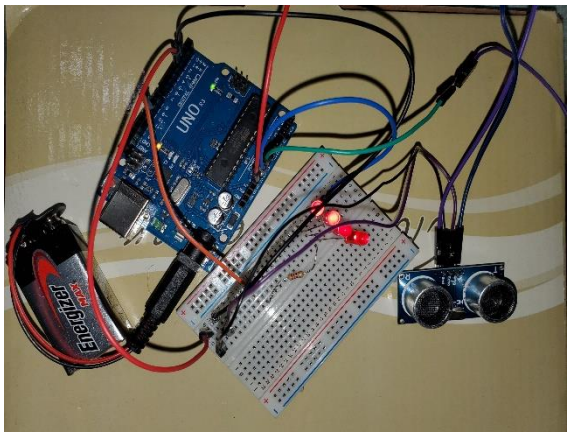
Block Diagram:



The input voltage supplied to ultrasonic sensor is 5v, so the VCC pin of the ultrasonic sensor is connected to the VCC pin of the Arduino. The ground terminal of ultrasonic sensor is connected to the ground terminal of Arduino UNO. The trigger pin is connected to the D2 pin and the echo pin is connected to the D3 pin. There is an additional pin called motor pin which acts as a connection to the output (LED) via D10 pin. The power is supplied to the equipment by using a 9v battery.

Working Model:





PROGRAM:

```
const int pingPin = 2; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 3; // Echo Pin of Ultrasonic Sensor
const int motorPin = 10;
```

We are assigning second pin, third pin of Arduino to trigger pin, echo of ultrasonic sensor. The tenth pin is assigned to LED.

```
Serial.begin(9600);
```

Initializing the baud rate to 9600

```
long duration, inches, cm;
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(10);
digitalWrite(pingPin, LOW);
pinMode(echoPin, INPUT);
pinMode(motorPin, OUTPUT);
```

We are assigning three variables duration, inches and cm. The trigger pin is set to low and high as it is used to transmit the signal. The echo is triggered when we receive high signal at the Echo terminal.

```
duration = pulseIn(echoPin, HIGH);
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);
```

Duration is set to high when the pulse received at echo pin is high. Microseconds are converted to inches and cm using the above two lines.

```
    if(inches>=23 && cm>=60)
    {
     digitalWrite(motorPin, LOW);
     Serial.println("Car not coming");
     }
else{
  digitalWrite(motorPin, HIGH);
     Serial.println("Car coming");
     }
```

If the value of inches and cm is greater the 23 and 60 then the output of the led is set to low and if the value detected is less than 23 inches and 60 cm the the led pin is set to high.

```
long microsecondsToInches(long microseconds) {
   return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
   return microseconds / 29 / 2;
}
```

Used to convert microseconds to inches and centimeters.

Final output: