**PROJECT- PROXIMITY SENSOR**

OBJECTIVE:

*A proximity sensor is an application that uses ultra-sonic distance sensor in a circuit to detect any movement in the surrounding region. Alarms can be used to detect.*

APPLICATIONS:

Proximity sensor are usually used in museums and in hospitals for quarantined patients.

**COMPONENTS USED TO BUILD THE CIRCUIT:-**

**1. ULTRA-SONIC DISTANCE SENSOR**:

Ultra-sonic distance sensor are electronic devices that are used to detect any movement at given distance.

**2. RESISTOR:**

Resistor helps in limiting the current to a bare minimum level, but also allows the panel to ensure that the current is flowing through all wires.

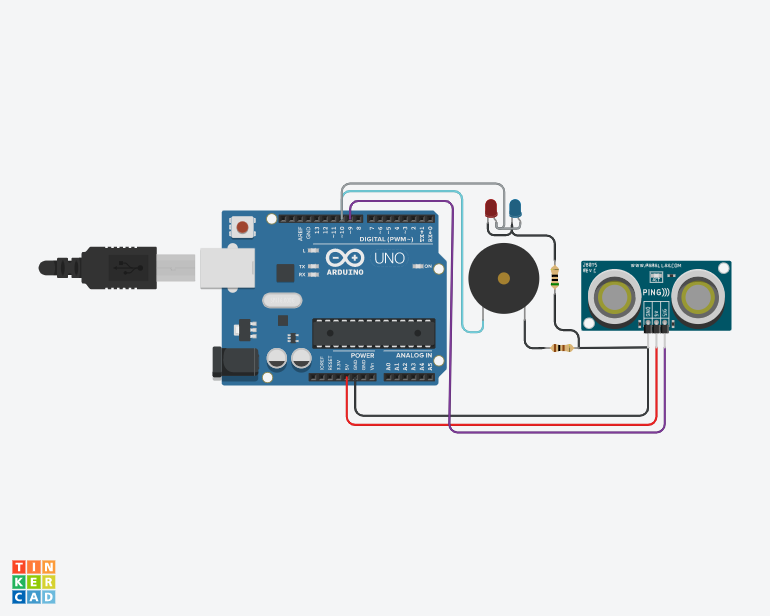
**3. LED:**

The led in this circuit is used as an indicator for movement.

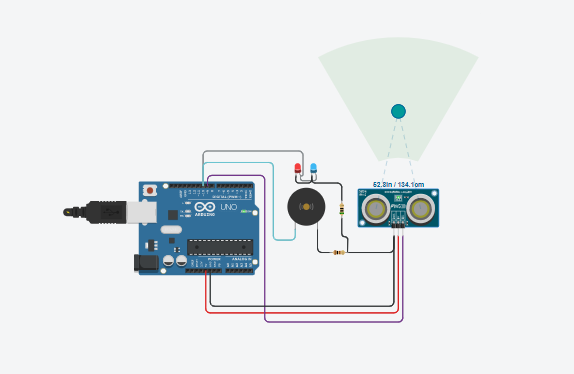
**4. BUZZER:**

A buzzer is used in a proximity sensor circuit. The circuit triggers the buzzer whenever it detects movement.

TINKERCAD CIRCUIT CONNECTION:



OUTPUT:



**WORKING:-**

Whenever the ultra-sonic distance sensor senses motion, it reduces it resistance and due to this decrease in resistance, voltage across the panel increases. Now when the voltage across the panel increases above a particular level, it triggers the led to glow and the buzzer also starts beeping. And when there is no motion, both the indication components turns off as the voltage across the panel goes below the desired level.

TINKERCAD ARDUINO CODE:

const int pingPin = 9;

const int buzzer = 10;

void setup() {

Serial.begin(9600);

pinMode(buzzer, OUTPUT);

}

void loop() {

long duration, cm;

pinMode(pingPin, OUTPUT);

digitalWrite(pingPin, LOW);

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);

delayMicroseconds(5);

digitalWrite(pingPin, LOW);

pinMode(pingPin, INPUT);

duration = pulseIn(pingPin, HIGH);

cm = microsecondsToCentimeters(duration);

Serial.print("Distance: ");

Serial.print(cm);

Serial.print("cm");

Serial.println();

if(cm < 150) {

digitalWrite(buzzer, HIGH);

}

else {

digitalWrite(buzzer, LOW);

}

delay(100);

}

long microsecondsToCentimeters(long microseconds) {

return microseconds / 29 / 2;

}