

- **PROBLEM STATEMENT :**

IoT Based Smart Solution For Railways

- **DOMAIN :**

Internet of Things

- **ASSIGNMENT 1:**

Smart home with at least two sensors and led, buzzer in TinkerCad

**By,**

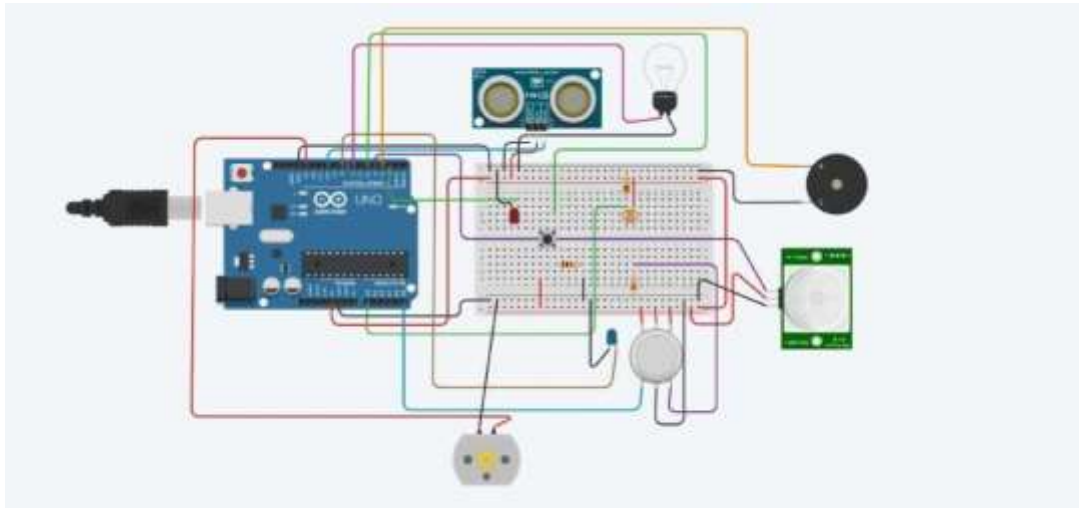
Raghul R (TEAM LEAD)

Krithicraj P

Ragul Dev P

Premnath G

## CIRCUIT DIAGRAM :



## ARDUINO UNO CODE:

```
const int pingPin =10;
const int ledUS = 2;
const int led = 7;
const int pirsensor =4;
#define photoSensor A0
#define sound 3
int const PINO_SGAS =A5;
int const ledGas = 8;
int const button = 5;
int const motor = 13;

void setup()
{
  pinMode(ledUS, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(sound, OUTPUT);
```

```
pinMode(ledGas, OUTPUT);
pinMode(motor, OUTPUT);
pinMode(pirsensor, INPUT);
pinMode(button, INPUT);
pinMode(photoSensor, INPUT);
Serial.begin(9600);
}
```

```
void loop()
{
    long duration, cm;
    int valLight = analogRead(photoSensor);
    int valPIR= digitalRead(pirsensor);
    int valGAS = analogRead(PINO_SGAS);
    valGAS = map(valGAS, 300, 750, 0, 100);
    int valBt = digitalRead(button);
    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);

    if(cm < 336){
        digitalWrite(ledUS, HIGH);
    }else{
        digitalWrite(ledUS, LOW);
    }
}
```

```
    }

    if(valLight < 890)
    {
        digitalWrite(led, HIGH);
    }
    else
    {
        digitalWrite(led, LOW);
    }

    if(valPIR == 1)
    {
        digitalWrite(sound, HIGH);
    }
    else
    {
        digitalWrite(sound, LOW);
    }

    if(valBt == 1)
    {
        digitalWrite(motor, HIGH);
    }
    else
    {
        digitalWrite(motor, LOW);
    }

    if(valGAS > 20)
    {
        digitalWrite(ledGas, HIGH);
```

```
    }  
else  
{  
    digitalWrite(ledGas, LOW);  
}
```

```
Serial.print(valPIR);  
Serial.println();  
}
```

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