

Problem Statement :

IoT-Based Gas Leakage Monitoring and
Alerting System in Industries

Domain :

Internet of Things

Assignment 1 :

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

By,

Karthick G- 720819106050

Kathiravan V- 720819106051

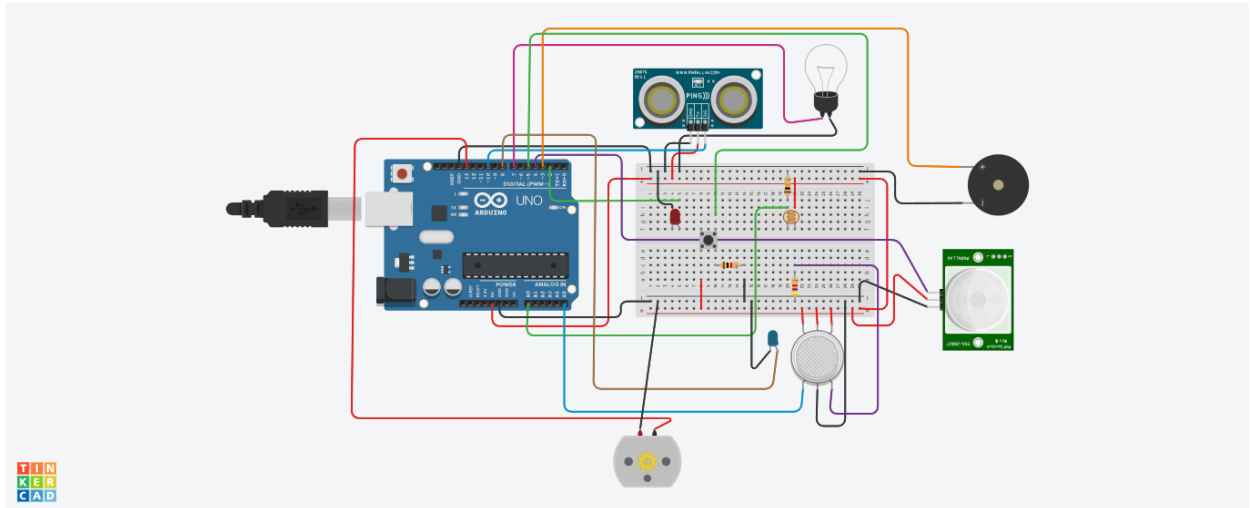
Kavin raj M- 720819106052

Krishnananth R– 720819106053

Link :

https://www.tinkercad.com/things/6k2FfP5I8fW-brave-migelo-amberis/editel?sharecode=KzPJ_TiBU8pThZploEd_nCUQhJjzx_X8DEbwvb1-ZLM

Circuit diagram :



Arduino Uno Code :

```
const int pingPin = 10;

const int ledUS = 2;

const int light = 7;

const int pir = 4;

#define photoSensor A0

#define buzzer 3

int const PINO_SGAS = A5;

int const ledGas = 8;

int const button = 5;

int const motor = 13;

void setup()

{

  pinMode(ledUS, OUTPUT);

  pinMode(light, OUTPUT);
```

```

pinMode(buzzer, OUTPUT);
pinMode(ledGas, OUTPUT);
pinMode(motor, OUTPUT);
pinMode(pir, INPUT);
pinMode(button, INPUT);
pinMode(photoSensor, INPUT);
Serial.begin(9600);
}
void loop()
{
    long duration, cm;
    int valLight = analogRead(photoSensor);
    int valPIR= digitalRead(pir);
    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(light, HIGH);
}else{
    digitalWrite(light, LOW);
}

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

```