

Project Title: Real Time Occupancy Detection

1. Problem Statement

Develop a simple prototype that will mimic a Real-time occupancy detection system which ultimately can offer an energy efficiency.

2. Scope Of Solution

Smart lighting control: Automatically adjust lighting levels in rooms or offices based on occupancy to save energy.

Intrusion detection: Detect and alert homeowners about unexpected entry into the premises.

Waiting areas: Manage patient waiting areas efficiently by tracking occupancy and wait times.

SafeCount

Real time occupancy monitoring



Supermarkets



Retail Stores



Buildings



3. Required Components To Develop Solution

Hardware:



Arduino



Ultrasonic Sensor



Breadboard



LED



Resistor

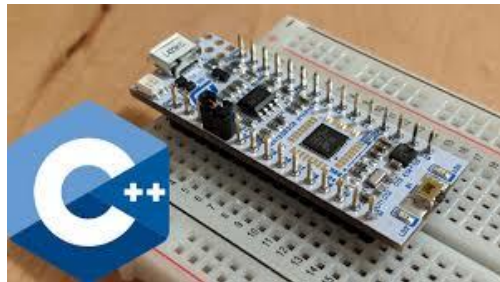


Jumper Wires

Software:



Arduino IDE

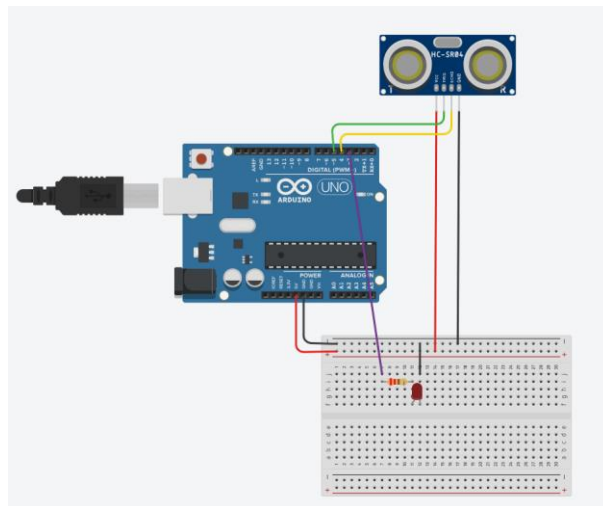


Embedded C++

4. Simulated Circuit(TinkerCad)

Circuit diagram illustrating the connections

Simulated using TinkerCad



5. Code For the Solution

```
#define trigPin 9 // Arduino pin connected to trigger pin on the ultrasonic sensor
#define echoPin 10 // Arduino pin connected to echo pin on the ultrasonic sensor
#define maxDistance 100 // Maximum distance (in centimeters) to detect occupancy
#define ledPin 13 // Arduino pin connected to the LED
```

```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
```

```
Serial.begin(9600);  
}  
  
void loop() {  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  
  unsigned long pulseDuration = pulseIn(echoPin, HIGH); // Measure pulse width in microseconds  
  
  unsigned int distance = pulseDuration / 58; // Convert pulse width to distance in centimeters  
  
  if (distance < maxDistance) {  
    digitalWrite(ledPin, HIGH); // Turn on the LED if an object is detected  
    Serial.println("Occupied");  
  } else {  
    digitalWrite(ledPin, LOW); // Turn off the LED if no object is detected  
    Serial.println("Vacant");  
  }  
  
  delay(100); // Delay for stability and to reduce serial output  
}
```