

Door Security System

Problem Statement

There is difficulty in knowing the security of that place from intruders or any caretakers to find their people etc.

Objective

The objective of this project is to send a push notification to the owner or the required person's mobile immediately after the door of that place (Room, Vault, House, etc) becomes open.

Components Required

1. Arduino Uno
2. Esp8266 wifi module (Any module containing esp8266 like node MCU)
3. Breadboard
4. Ir sensors (three of which I used)
5. Ultrasonic sensor
6. Power Supply (Batteries if possible)
7. Voltage regulators (if required)
8. Connecting / Jumper wires

Components With Their Usage

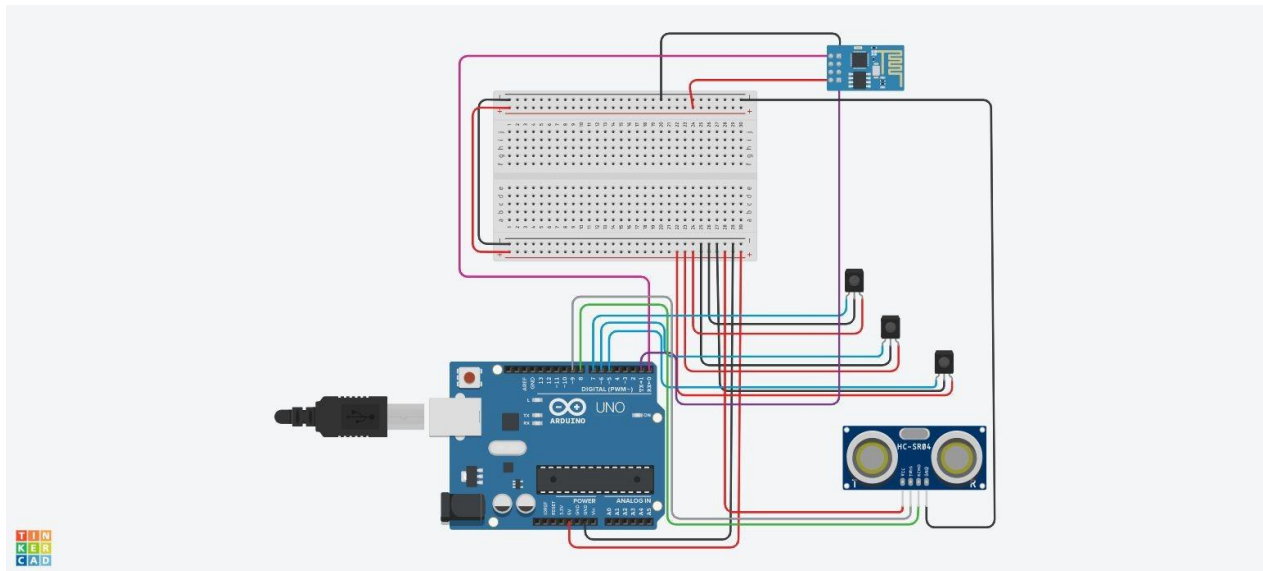
Arduino Uno - MicroController Which interacts with Sensors and Esp8266 wifi module

Esp8266 wifi module - Connects with the Internet to send push Notification

Ultrasonic Sensor - Measures distance - Here especially between door and sensor

Ir Sensor - Emits Ir rays to detect objects - Here to detect whether the door is near(at a very small distance)

Circuit Diagram



Circuit Explanation

The above circuit diagram is just a prototype. It may not look the same as the real components.

The **Arduino Uno** is the main part that senses the data from the sensors and communicates with ESP8266. **ESP8266** is the module that is connected to the internet and sends a command to send a notification.

The **Red and Black** wires are the Power supply and Ground of the whole system.

A separate **voltage regulator** must be connected only for the esp8266 module(not mentioned in the above circuit) whereas it works only up to 3.6v

The **Green and Grey** wire are the connection for data transfer between the Arduino Uno and the ultrasonic sensor.

The **blue wires** are the connection for data transfer between the Ir sensor and the Arduino Uno

The **purple wire** is the connection between the Tx of Esp8266 and the Rx of Arduino UNO where data is transferred from Esp8266 to Arduino UNO.

The **violet wire** is the connection between the Tx of Arduino UNO and the Rx of Esp8266 wheredata is transferred from Arduino UNO to Esp8266.

Explanation

The system ensures whether the door is open or not by the distance between the side wall or the place where the door touches when it opens and the door

The system is placed at the wall or any plane closer to the door (near to hinges of the door) so that if the door is open the distance between the ultrasonic sensor and the wall is less.

There is a signal transfer between the Arduino Uno and the esp8266 wifi module through the Tx and Rx GPIO pins. Where the Tx(Transfer) pin sends the signal Rx(Receiving) pin receives the signal.

The code processed in the system is done separately for Arduino Uno and Esp8266 wifi module

Here the ultrasonic sensor checks the distance between the door and the sensor. If the distance is decreasing then the Arduino Uno sends the signal for the **opening of the door** through the Tx pin to esp8266.

Else If the door is very near to the IR sensor and ultrasonic sensor (The Door is open) the Arduino Uno sends the signal that ***the door is Open*** through the Tx pin to esp8266.

If both the above conditions are not satisfied (Indicates door is not open) Arduino doesn't send any signal or it sends that door is not open through Tx pin to esp8266.

The ***ESP8266 WIFI module*** which is connected to wifi of that specific place receives the signal from the Arduino Uno through the Rx pin of Esp8266. This ESP8266 Module will be connected to any of the ***Cloud services*** like Blynk(which I used), IFTTT, Push Bullet, etc to send push notifications. We have to log in and set everything according to the command received from the esp8266 WIFI module.

Whereas the Esp8266 module responds to the signal received from the Arduino Uno by sending the command about the status of the door accordingly - if the door is opening then the notification Door is opening is sent, if the door is open then the notification door is open is sent.

Code:

Node MCU:

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "Zm5xXEIhbL_UiJusy6PgvL-Vle2TUcJz";
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "LAPTOP-FHOGLOPQ 0315";
char pass[] = "B5&s*NJI";
String myString;
char rdata;
String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
```

```

int maxIndex = data.length() - 1;
for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
        found++;
        strIndex[0] = strIndex[1] + 1;
        strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
}
return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

void sendSensor(String Deside){

    // and PUSH Notification
    if( Deside == "alert"){
        Blynk.notify("Alert The Door is open");
    }

    else if( Deside == "just"){
        Blynk.notify("The Door is opening");
    }
}

void setup(){
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass);
    while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
    }
}

void loop(){

    if (Serial.available() > 0 )
    {
        rdata = Serial.read();
        myString = myString+ rdata;
        // Serial.print(rdata);
        if( rdata == '\n')
        {    // new code
            String l = getValue(myString, ',', 0);
            myString = "";

```

```

Serial.println(l);
Blynk.run();//communicate to mobile
sendSensor(l);
}
}}

```

Arduino:

```

#include <SoftwareSerial.h>
SoftwareSerial nod(0,1);
int IRSensor1 = 4;// connect ir sensor to arduino pin 2
int IRSensor2 = 5;// connect ir sensor to arduino pin 2
int IRSensor3 = 6;// connect ir sensor to arduino pin 2

#define trigPin 9 //declaring trigger pin
#define echoPin 10 //declaring echo pin
long duration; // declaring variable for storing time
int distance; // declaring variable for storing distance

String cdata; // complete data, consisting of sensors values

void Ultrasonic(){

digitalWrite(trigPin, LOW);//Intialize trigpin to low
delayMicroseconds(2);//wait for 2 micro seconds

digitalWrite(trigPin, HIGH); // Sets the trigPin on HIGH state for 10 micro seconds
delayMicroseconds(10);
digitalWrite(trigPin, LOW);// After 10 micro seconds make it low


duration = pulseIn(echoPin, HIGH);// Reads the echoPin, returns the sound wave
travel time in microseconds

distance= duration*0.034/2; // Calculating the distance
}
String DoorValue(int a,int b, int c,int d){
  if(a == 0 && b == 0 && c == 0 && d < 15 ){
    return "alert,";
  }
}

```

```

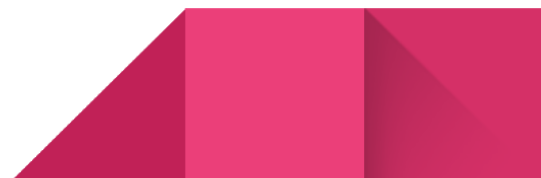
}
else if( distance < 25 ){
  return "just,";
}
else
{
  return "ok,";
}
}
void setup() {
pinMode (IRSensor1, INPUT); // sensor pin 1 INPUT
pinMode (IRSensor2, INPUT); // sensor pin 2 INPUT
pinMode (IRSensor3, INPUT); // sensor pin 3 INPUT
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(115200);
nod.begin(115200);
}
void loop() {
Ultrasonic();
int statusSensor1 = digitalRead (IRSensor1);
int statusSensor2 = digitalRead (IRSensor2);
int statusSensor3 = digitalRead (IRSensor3);
cdata= DoorValue(statusSensor1,statusSensor2,statusSensor3,distance);
nod.println(cdata);
cdata="";
// Prints the distance on the Serial Monitor
delay(100);
}

```

Alternative Method:

The alternative (not mentioned in the above circuit and components) which can be done is that we can use some other sensors. Here the Ir sensors can be replaced by a **Magnetic door sensor**. This sensor where is attached to the corner of the door and door frame.

Whereas here if the two magnets are separated(door is opening), the "**Door is opening**" signal is sent to the Esp8266. If the above action and the ultrasonic value are less(Door is open) then the "**the door is open**" signal is sent to esp8266.

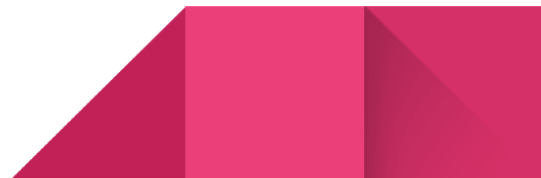


Then the work of Esp8266 will be the same as above mentioned

Real-Time Usage

This system is very useful for security vaults if the password for the lock is cracked.

This system will be useful for people who must be tracked for any lockdown etc. For example, if we wanna track Elderly people from their rooms etc can be done through this.



Conclusion

Ensuring security is a must for everyone. Here the information of Entry or Exit is intimated to the Owner or specific people is done. This can also be altered by checking the number of people who entered and exited, recording them for future purposes, etc. These all can be done through IoT.