Topic

Intruder alert system using Passive Infrared Sensor (PIR) and Arduino.

Abstract

- Here we making an intruder alarm using Arduino and PIR sensor. PIR sensor is a special type sensor which is usually used for security purposes.
- It detects the objects by reading the Infrared radiations emitted by the objects.
- Any object whose temperature is above absolute zero, emits radiation. This radiation is not visible to human eyes. The PIR sensor is designed to detect this Infrared radiation.
- When the PIR sensor detects an intruder, it will send a signal to Arduino and the Arduino will sound an alarm.

Motivation

This project aims at providing a security system for our house by detecting any intruders trying to break into our house and alert the owner through a message or a call.

Main objective of our Project is to provide a working model and a basic security system for anyone, which can detect motion and alerts the user immediately via SMS or a Call.

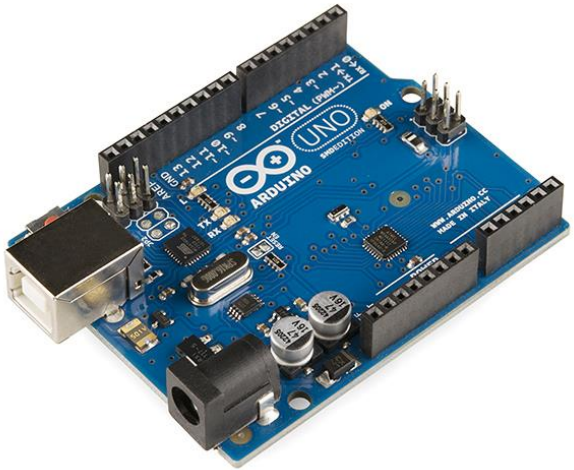
Agenda

1. Introduction about all the sensors
2. TinkerCad Circuit
3. Arduino Code
4. Implementation
5. Output and Results
6. Further Extensions

01

Introduction about all the sensors

Arduino



Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on wiring), and the Arduino Software (IDE), based on Processing.

Buzzer

The buzzer is a sounding device that can convert electric signals into sound signals. It is usually powered by DC voltage. It is widely used in alarms, computers, printers and other electronic products as sound devices. It is mainly divided into piezoelectric buzzer and electromagnetic buzzer.

According to different designs and uses, the buzzer can emit various sounds such as music, siren, buzzer, alarm, and electric bell.

Working Principle of a Buzzer:

The working principle of a buzzer depends on the theory that, once the voltage is given across a piezoelectric material, then a pressure difference is produced. A xi piezo type includes piezo crystals among two conductors.

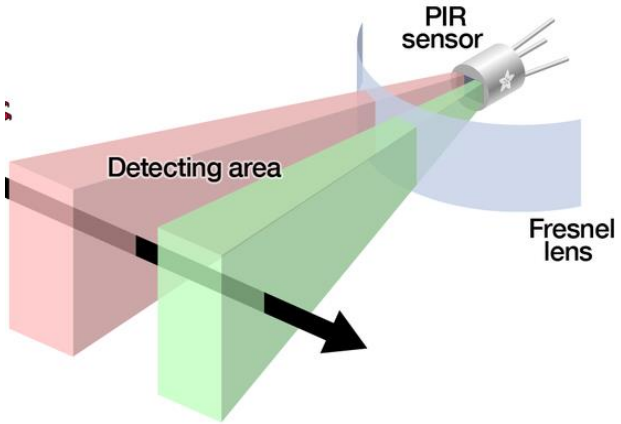
Once a potential disparity is given across these crystals, then they thrust one conductor & drag the additional conductor through their internal property. So, this continuous action will produce a sharp sound signal.

Passive IR

A passive infrared sensor is an electronic sensor that measures infrared light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications.

The IR sensor itself is housed in a hermetically sealed metal can to improve noise/temperature/humidity immunity. There is a window made of IR-transmissive material (typically coated silicon since that is very easy to come by) that protects the sensing element. Behind the window are the two balanced sensors.

PIR sensors have ranges of up to 10 meters (30 feet), a single detector placed near the entrance is typically all that is necessary for rooms with only a single entrance.


PIR
sensor
Detecting area
Fresnel
lens

GSM Module

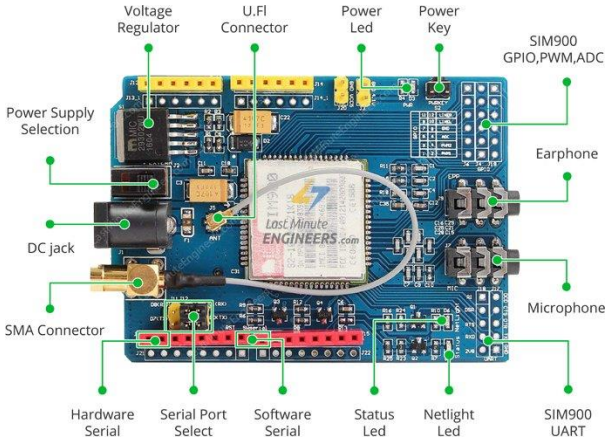
- GSM is called Global System for Mobile communication.
- A GSM modem or GSM module is a device that uses GSM mobile telephone technology to provide a wireless data link to a network.
- The Arduino GSM shield allows an Arduino board to connect to the internet, send and receive SMS, and make voice calls using the GSM library.
- The GSM module is a high quality, portable solution for any indoor and outdoor applications.

Network operator requirements:

To access a network, you must have a subscription with a mobile phone operator a GSM compliant device like the GSM shield or mobile phone, and SIM card. The network operator provides the SIM card, which contains information like the mobile number, and can store limited amounts of contacts and SMS messages.

SIM card:

In addition to the GSM shield and an Arduino, you need a SIM card. The SIM represents a contract with a communications provider. The communications provider selling you the SIM has to either provide GSM coverage where you are, or have a roaming agreement with a company providing GSM coverage in your location.

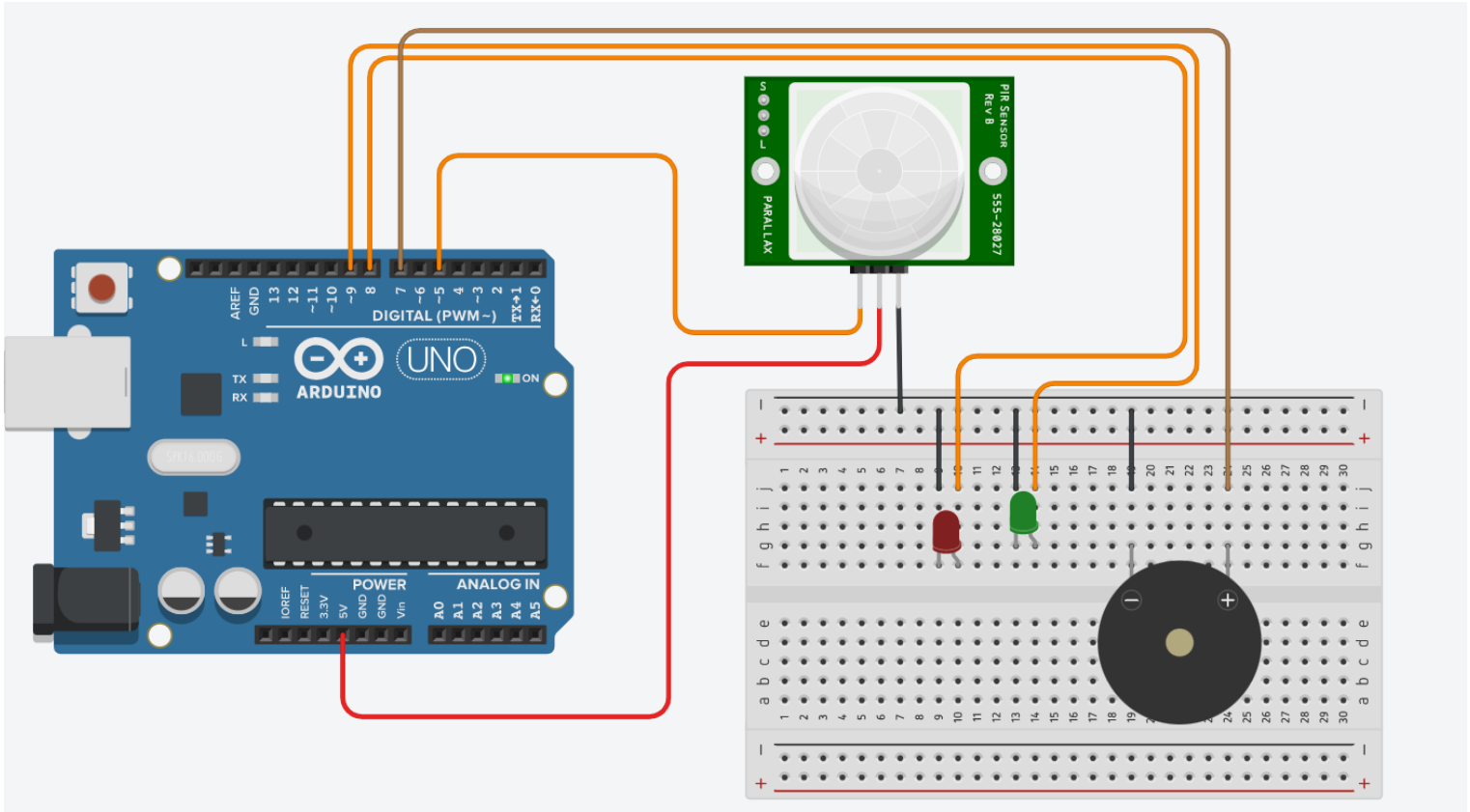


3	DTR	Data terminal Ready [Serial port]
5	DCD	Data carry detect [Serial port]
6	DSR	Data Set Ready [Serial port]
7	CTS	Clear to send [Serial port]
8	RTS	Request to send [Serial port]
9	TXD	Transmit data [Serial port]
10	RXD	Receive data [Serial port]

02

TinkerCAD Circuit

TinkerCAD circuit – without GSM



03

Arduino Code

Software Part - Arduino Code

```
#include <SoftwareSerial.h>

SoftwareSerial SIM900(2, 3);
String textForSMS;

int Buzzer= 8;  // the pin that the Buzzer is atteched to
int RedLED= 13;    // the pin that the Red LED is atteched to
int GreenLED = 12; // the pin that the Green LED is atteched to
int PIR= 5;     // the pin that the PIR sensor is atteched to
int val= 0;

void setup() {
  randomSeed(analogRead(0));

pinMode(Buzzer, OUTPUT);
pinMode(RedLED, OUTPUT);
pinMode(GreenLED, OUTPUT);
pinMode(PIR, INPUT);

digitalWrite(Buzzer, LOW);
digitalWrite(RedLED, LOW);
digitalWrite(GreenLED, LOW);
delay(100);

Serial.begin(9600);
SIM900.begin(9600);
```

- Import SoftwareSerial.h library to Arduino library folder.
- Give all pin connections
- In void setup put your setup code here, to run once.
- Her we give pinMode to all sensors which is input and which is output.

```
}

void loop() {
val = digitalRead(PIR);

if(val == HIGH){
  Serial.println("Motion detected!");
textForSMS =  "\n Motion detected!!! \n Any Person in your Room  Plz Check It ";


  digitalWrite(RedLED, HIGH);
  delay(10);
  digitalWrite(Buzzer, HIGH);
  delay(10);
  digitalWrite(GreenLED, LOW);

sendSMS(textForSMS);
//Serial.println(textForSMS);
Serial.println("message sent.");
delay(100);

MakeCall();
delay(100);
}
```

- In void loop, put your main code here, to run repeatedly.
- Here we are created two if conditions, one condition runs when PIR sensor holds value high.
- Another runs when PIR is low.
- Inside the loop we give respective value for other sensors.
- And in PIR high case, we write code that calls the GSM main function.
- Which send messages and calls to user.

```
52  if(val == LOW){
53
54    digitalWrite(RedLED, LOW);
55    digitalWrite(Buzzer, LOW);
56    digitalWrite(GreenLED, HIGH);
57    delay(1000);
58    Serial.println("Movement not Detected");
59
60  }
61  }
62
63  void sendSMS(String message){
64  SIM900.print("AT+CMGF=1\r");                    // AT command to send SMS message
65  delay(1000);
66  SIM900.println("AT + CMGS = \"+917569183893\""); // recipient's mobile number, in international format
67  delay(1000);
68  SIM900.println(message);                        // message to send
69  SIM900.println((char)26);                       // End AT command with a ^Z, ASCII code 26
70  delay(1000);
71  SIM900.println();
72  // give module time to send SMS
73  }
```

```
75  void MakeCall()
76  {
77    SIM900.println("ATD+917569183891;"); // ATDxxxxxxxxxx; -- watch out here for semicolon at the end!!
78    SIM900.println("Calling  "); // print response over serial port
79    delay(1000);
80  }
81
```

- We create to functions in the code, one is to Send messages and another function is to make call.
- There are AT commands to make this function works.
- To send sms we need to use this command: mySerial.println("AT+CMGF=1");
- To make a call, the command is mySerial.println("ATD+60XXXXXXXXX;"); replace x with number you want to call, and change +60 to your country code.

Basic AT commands

1. To change sms sending mode : **AT+CMGF=1**

```
mySerial.println("AT+CMGF=1");
```

2. To read SMS in text mode : **AT+CNMI=2,2,0,0,0**

```
mySerial.println("AT+CNMI=2,2,0,0,0");
```

3. To make a call : **ATD+60XXXXXXXXX;** //replace X with number you want to call, change +60 to your country code

```
mySerial.println("ATD+60XXXXXXXXX;");
```

4. To disconnect / hangup call : **ATH**

```
mySerial.println("ATH");
```

5. To redial : **ATDL**

```
mySerial.println("ATDL");
```

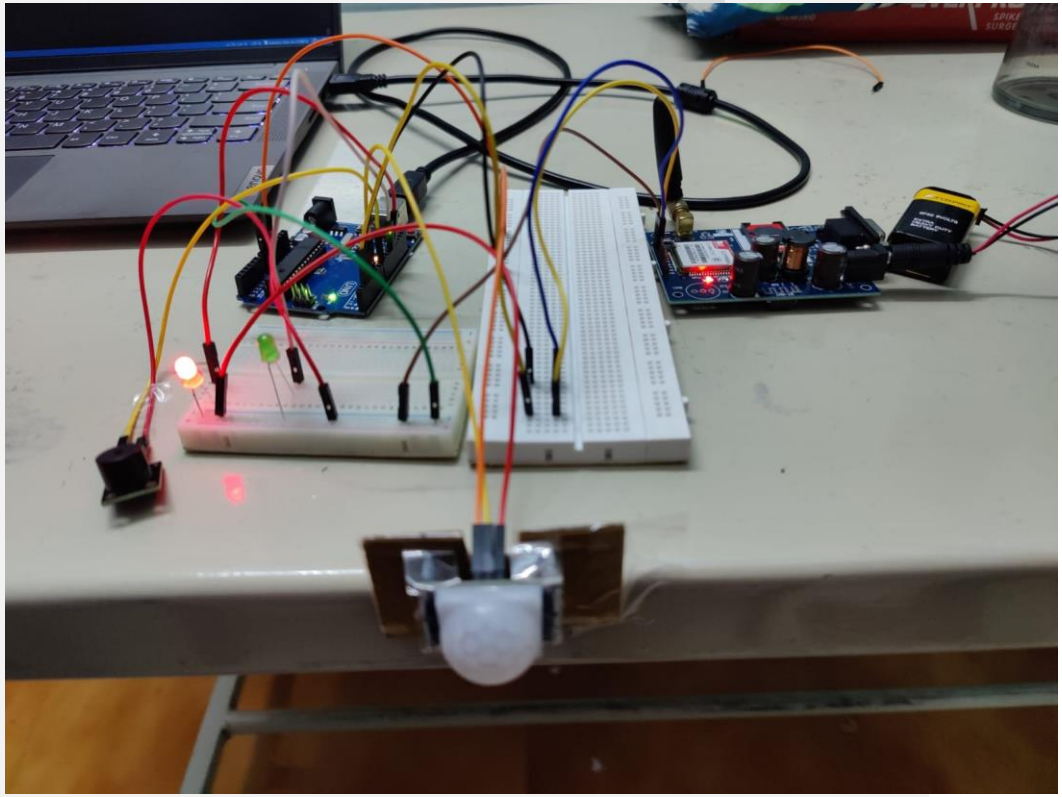
6. To receive a phone call : **ATA**

```
mySerial.println("ATA");
```

04

Implementation of Circuit

Circuit



Implementation Process

1) PIR to Arduino

- Connect the Vcc of PIR to 5V on Arduino
- Connect the GND of PIR to GND on Arduino
- Connect the OUTPUT pin of PIR to Digital pin D3 on Arduino

2) Buzzer to Arduino

- Connect one pin of buzzer to digital pin D8 on Arduino
- Connect the other pin of buzzer to GND on Arduino

3) LED to Arduino

- Connect the LED positive to Digital pin D13 on Arduino through a resistor.
- Connect the LED negative to GND on Arduino.

4) GSM to Arduino

- Connect vcc pin to 5V on Arduino
- Connect the GND of GSM to GND on Arduino
- Connect tx pin to Digital pin D02 on Arduino
- Connect rx pin to Digital pin D03 on Arduino

Requirements

- Components required :

1. PIR
2. Buzzer
3. 10k-ohm resistor
4. Breadboard
5. Connecting wires
6. Arduino
7. Power supply
8. LED
9. Wi-Fi/GSM module

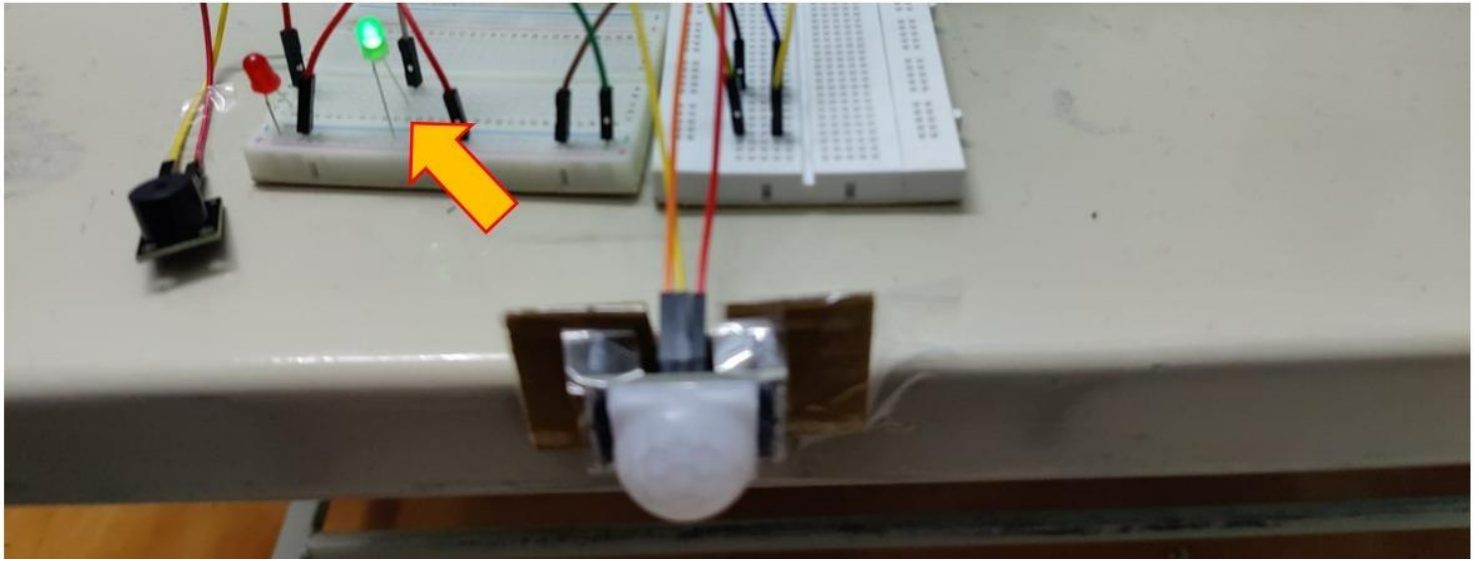
- Now we just assemble all the components according to the code in Arduino and we connect the connecting wires according to the Code.

05

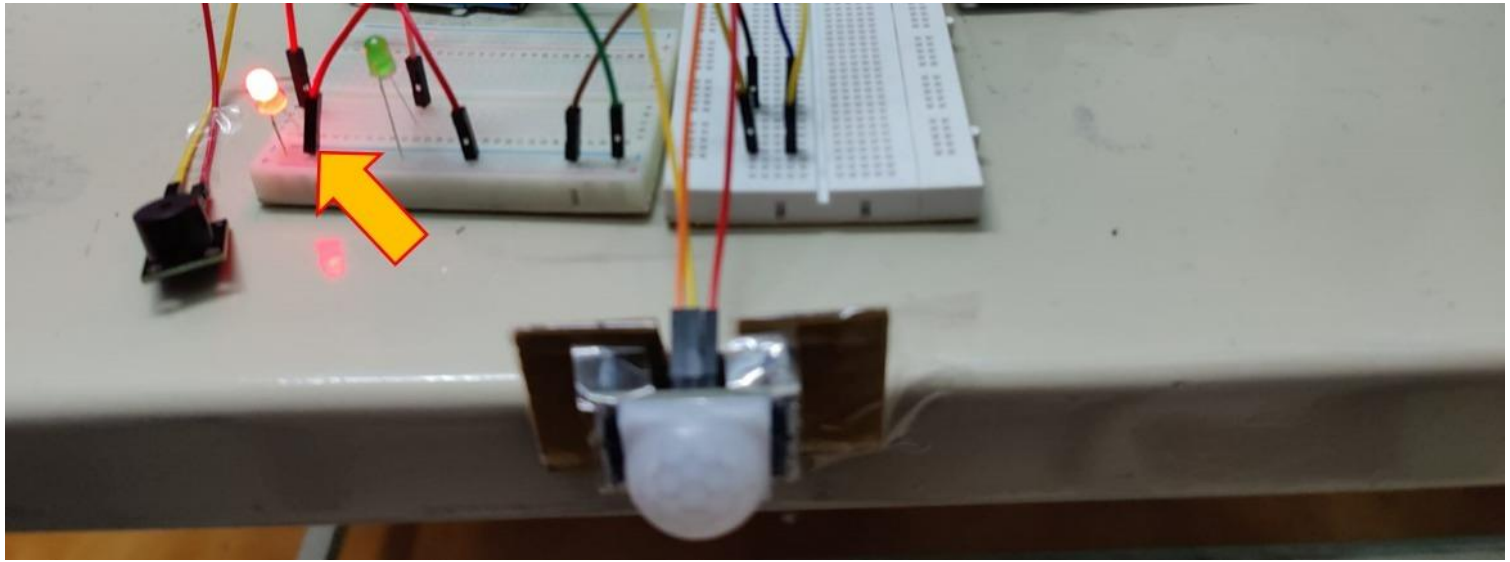
Output & Results

Output & Results

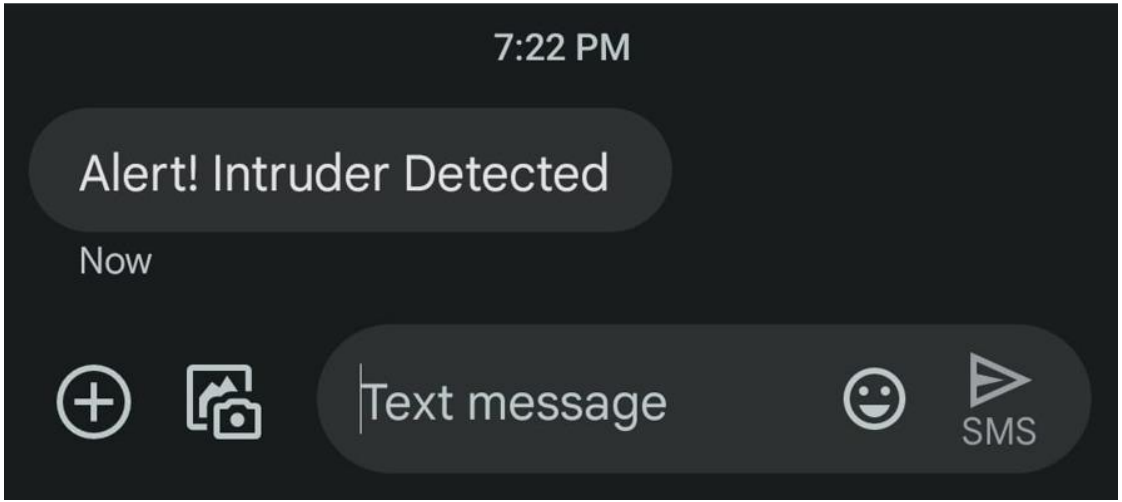
Before Detection (Green LED):

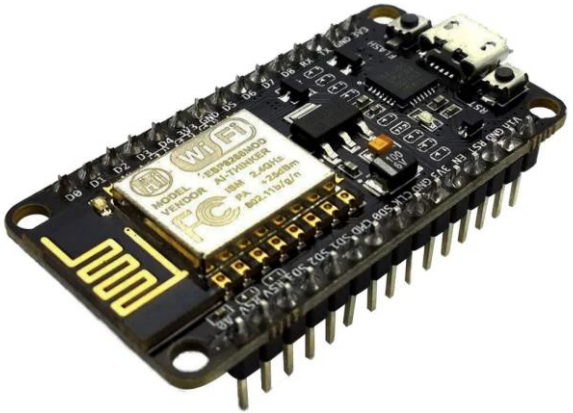


After Detection (Red LED):



Screenshot of SMS Alert to the Phone





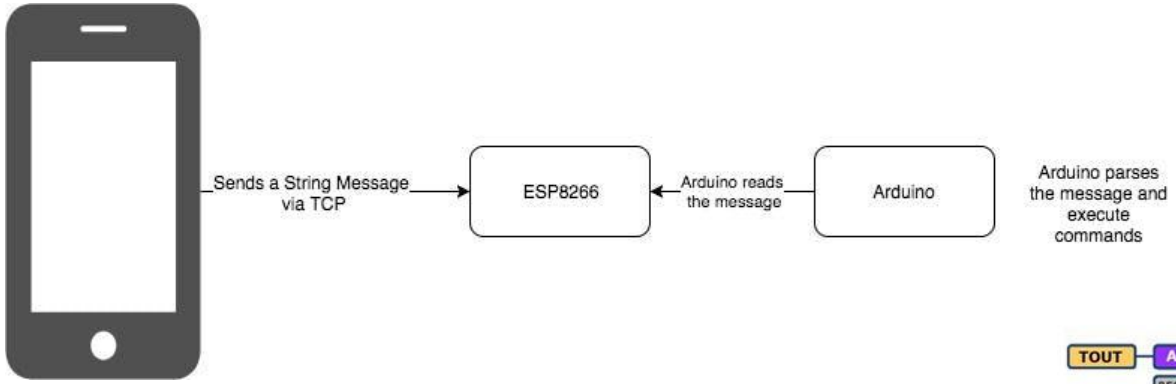
Implementation using

ESP 8266 Node MCU Wi-Fi Module

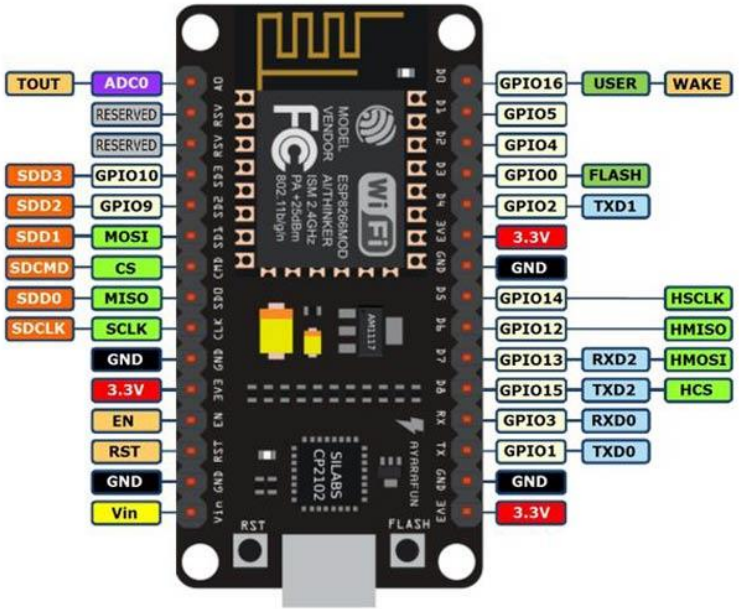
ESP8266 Wi-Fi Module:

- An ESP8266 Wi-Fi module is a SOC microchip mainly used for the development of end-point IoT (Internet of things) applications. It is referred to as a standalone wireless transceiver, available at a very low price. It is used to enable the internet connection to various applications of embedded systems.
- It supports both the TCP/IP capability and the microcontroller access to any Wi-Fi network. It provides the solutions to meet the requirements of industries of IoT such as cost, power, performance, and design.

Working of Module and it's Pins



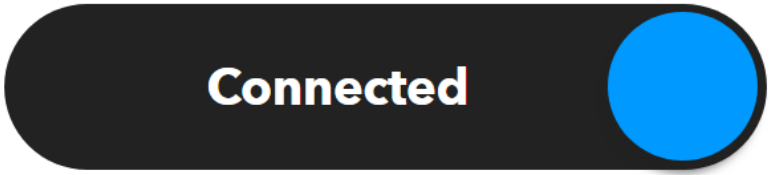
Schematic Diagram of Pins



Working Theory:

- We have connected the PIR motion sensor to one of the GPIO pins of the ESP 8266 – 01.
- Whenever the PIR detects motion then the output sates of the PIR sensor will change which will be detected by our ESP 8266 – 01 controller.
- We have programmed the ESP to make an HTTP GET request to our IFTTT webhooks applet.
- Whenever the HTTP GET request has been made then it will trigger a notification alert on our mobile phone.

Creating an Applet in IFTTT



Arduino Code for Intruder Alert System

sketch_jul19c

```
// Intruder Alert Detection
// Using ESP8266 NodeMCU Module we send notification or email via ifttt.com webhook service
//
// When the trigger GPIO input (sensorPin) goes high, an Notification/Email Alert is generated by using ESP8266
// to contact ifttt.com over a wifi connection and trigger an action to send an alert email


#include <ESP8266WiFi.h>

// wifi config
const char* ssid      = "Rahul's OnePlus";
const char* password  = "987654321";

// ifttt.com access details
const char* host      = "maker.ifttt.com";
const char* urlPt1    = "https://maker.ifttt.com/trigger/";
const char* urlPt2    = "/json/with/key/";
const char* eventName = "Door_Opened";                    // the event to trigger at ifttt.com
const char* apikey    = "4qE8WuyFLz5AL7gBfGMkp";                    // your ifttt.com api key
const byte  port      = 80;                                  // http standard port
```

Uploading and testing:

- Now upload the code to your ESP8266 – 01 using an Arduino as a USB to TTL converter and upload the code.
- Then connect the circuit and power it on.
- Wait for 30 seconds to calibrate the PIR sensor, when calibration is done the onboard LED will turn OFF.
- Now whenever motion is detected the onboard LED will blink and a push notification will arrive on your phone saying “Motion Detected”.

```
#define sensorPin  D1    // PIR Sensor input: low = closed  high = open
#define led     D4    // status LED on esp8266 module

WiFiClient client;    // wifi client object
void setup() {
  Serial.begin(9600);

  pinMode(sensorPin, INPUT_PULLUP);    //  switch input
  pinMode(led, OUTPUT);             // use on board LED for sensor state indication and wifi joining progress
  digitalWrite(led, HIGH);          // turn LED off

  // join wifi network
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi= ");
  Serial.print(ssid);
  Serial.println(" ...");

  while (WiFi.status() != WL_CONNECTED) {
    digitalWrite(led, !(digitalRead(led)));  // toggle LED
    delay(500);
    Serial.print('.');
  }

  digitalWrite(led, HIGH);          // turn LED off

  Serial.println();
  Serial.println("Welcome to the Working of Alert System.");
  Serial.print("Connecting to WiFi= ");
  Serial.println(ssid);
  Serial.println("Connected!");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Serial.println("Waiting for Event to Trigger.\n");
}
```

```
void loop() {
  if (digitalRead(sensorPin) == HIGH) {                // check for motion

    Serial.println("Event Occured ");
    Serial.print("Sending Alert Notification");
    digitalWrite(led, LOW);                            // turn LED on

    if (!client.connect(host, port)) {                 // connect to ifttt.com
      Serial.println("Connection failed.");
      return;
    }

    // request ifttt.com to trigger the Notification
    String url = String(urlPt1) + String(eventName) + String(urlPt2) + String(apikey);
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
                 "Host: " + host + "\r\n" +
                 "Connection: close\r\n\r\n");

    delay(100);  // without the delay, the request may not be completed


        while (client.available())
        {
          Serial.write(client.read()); // optionally print out the response from ifttt.com
          Serial.println(client.read());
        }


    Serial.println("\nDisconnecting from ifttt.com");
    client.stop();
```

```
    // wait 5 seconds when door is first opened,
    // then assuming front door is likely still going to be open (as people enter/exit),
    // wait until front door is closed before looking for next door open event
    delay(5000);
    while (digitalRead(sensorPin) == HIGH) {
      delay(100);
    }

    digitalWrite(led, HIGH);                          // turn LED off
    Serial.println("\nWaiting for Event to Occurred.\n");
  }
}
```

FUTURE EXTENSIONS

- This project can be further improvised by adding multiple Arduino Modules such as Bluetooth Module and Wi-Fi Module.
- With Bluetooth Module, we can connect to many other IOT devices at house to our Arduino Connection.
- Smart locking systems can be connected and whenever PIR detects an intruder, it sends signal to Arduino and Bluetooth; and to Smart Lock System which locks all our internal systems before intruder tries to enter inside.
- Wi-Fi module is also very useful if the Owner's Mobile device is not connected to Carrier Services and not able to receive calls or SMS.
- If by any chance the Owner is connected to Wi-Fi, using APIs of some Internet Messaging Services, we can automate our code to send a text message to that app, which doesn't require a Signal to be established

Thank You