JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

INTELLIGENT SECUIRTY SYSTEM BASED ON DISTANCE RECOGNITION AND INTERNET OF THINGS

[1]Sonali Prasad 1st Author, [2]Rajesh Rajaan 2nd Author

[1] M.Tech 4th Sem Student, [2]Assistant Professor

[1]Computer Science and Engineering,

[1]Global Institute of Technology, Jaipur, India

Abstract: Security systems are being used exponentially in our daily lives. For example, the security of a business, organization, or bank safe deposit box is important to every individual today. These days, surveillance cameras are used to create safe spaces within organizations. IoT Security is a technology segment focused on protecting devices and networks connected to the Internet of Things. IoT security refers to the protection methods used to protect devices connected to the Internet or network-based devices. The term IoT is very broad, and as technology has advanced, the term has become more and more widespread. Almost all tech devices, from watches to thermostats to video game consoles, can interact with the Internet and other devices in some way. IoT Security is a family of technologies, strategies, and tools used to prevent these devices from being compromised. Ironically, it is the unique connectivity of the IoT that makes these devices increasingly vulnerable to cyberattacks. The IoT is so wide that IoT security is even wider. This has created a variety of ways to become part of IoT security. Application Program Interface (API) security, Public Key Infrastructure (PKI) authentication , and network security are methods that IT leaders can use to combat the growing threat of cybercrime and cyberterrorism with roots in vulnerable IoT devices. It's just a part.

Nowadays, security is the most critical concern, whether it is data security, home security, human life security, or asset protection. Threats are becoming more prevalent as security technology advances. As a result, the growing use of IoT and biometric authentication for security has gained traction. Distance recognition is regarded as a collectable and approachable biometric system. The same identifier that humans use to distinguish one person from another is used in biometric distance recognition.

Distance recognition was considered in this work to produce a highly reliable door lock mechanism. When an object or a person is within a particular distance of the door detector, the system performs real-time distance detection and recognition. A prototype has been created and tested.

I. Introduction

To avoid any security-related issues in your home or the organization we need to develop a system that can intimate or alert you before losing your valuables. The Internet of Things (IoT) is a network of connected devices, each with a unique identifier that automatically collects and exchanges data over the network. IoT umbrellas do not always include internet-based devices. Devices that use Bluetooth technology are also IoT devices, so IoT security is required. These mistakes have contributed to the recent surge in IoT-related data breaches.

IoT security refers to the protection methods used to protect devices connected to the Internet or network-based devices. The term IoT is very broad, and as technology has advanced, the term has become more and more widespread. Almost all tech devices, from watches to thermostats to video game consoles, can interact with the Internet and other devices in some way.

IoT Security is a family of technologies, strategies, and tools used to prevent these devices from being compromised. Ironically, it is the unique connectivity of the IoT that makes these devices increasingly vulnerable to cyberattacks.

IoT devices are being used in a variety of sectors and industries, including:

Consumer Applications - Smartphones, wearables, and smart houses that control everything from air conditioning to door locks from a single device are examples of consumer IoT devices.
Business Applications – Enterprises utilise IoT devices such as smart security cameras, automobiles, ships, product trackers, and sensors that collect data on industrial machines.
Government Applications – Devices for tracking wildlife, monitoring traffic bottlenecks, and giving natural catastrophe alerts are examples of government IoT uses. There are presently billions of IoT devices in use around the world. Their increasing prominence in our daily lives has increased examination of their specific security vulnerabilities. This will be taken into account here.

1.1 How to manage Internet of Things devices

IoT devices must be managed both internally (such as software maintenance) and externally in order to perform properly (communication with other devices).
This is accomplished by connecting each Internet of Things (IoT) device to a management entity known as the Command and Control (C & C) Center. The centre is in charge of deploying and authenticating operations such software maintenance, configuration, firmware upgrades for bug and vulnerability patches, and device registration. The application programme interface allows communication between devices (API). Once a device manufacturer's API is disclosed, other devices or applications can use it to gather and communicate data. Some APIs allow you to control the gadget as well. A building administrator, for example, can use the API to remotely lock a specific office door.

IOT Vulnerabilities and Security Issues

Day-to-day IoT activities are successfully managed through the C & C Center and API. However, because it is centralised, it offers a slew of exploitable flaws, including:

Unpatched vulnerabilities - Devices frequently run outdated software and remain open due to connectivity challenges or the requirement for end users to manually download updates from the C & C Center. I have. Concerning recently identified flaws.

Weak Authentication – Manufacturers frequently deploy IoT devices (such as home routers) with readily cracked passwords that providers and end users may possess. Allowing remote access to these devices exposes them to attackers using automated scripts for widespread exploitation.

Vulnerable APIs – APIs are commonly targeted by a number of threats, including man-in-the-middle (MITM), code injection (such as SQLI), and distributed denial of service. Attack (DDoS). Find out more about the consequences of API targeted attacks.
There are two types of hazards posed by exploitable devices: threats to users and threats to other users.

User Threats - Users are put at risk by compromised IoT devices in a variety of ways, including purchase data, credit card information, and personal health information. This data is vulnerable to theft when devices are not properly safeguarded. Furthermore, vulnerable devices can be utilised as gateways to other sections of the network where they are deployed, enabling for the extraction of more sensitive data.

Physical Harm - IoT devices are examples of pacemakers, cardiac monitors, defibrillators, etc., and are now common in the medical industry. While these devices are useful (for example, doctors can remotely fine-tune a patient's pacemaker), they are also vulnerable to security threats.
The improperly fixed device can be abused to interfere with the patient's medical care. This is a very rare event, but it should be considered when developing strategies to protect IoT devices.

Remote Exposure - Because of their internet-based connectivity, IoT devices offer a far larger attack surface than traditional technologies. While this accessibility is incredibly helpful, it also allows hackers to interact with the device remotely. This is why hacking tactics like phishing are so powerful. To secure its assets, IoT security, like cloud security, must cope with multiple entry points.

The present work is about developing a system like this. We used an HC-SR04 ultrasonic sensor to determine the distance to an object in our work. We used Arduino IDE to program for this purpose. SMS gateway we used to send a notification after distance detection.
It is very helpful for society, farmers, and corporate as it is easily affordable, and installation is very easy to do. Due to its low-cost farmers and corporate save their big loss in terms of money and time. It works as an antitheft system which will be primary source for any major loss related to security.

Chapter 2 talks about the literature survey based on IOT security and related things to IOT.

Chapter 3 is about the details of Arduino IDE (Integrated Development Environment), HC-SR04 Ultrasonic Sensor with the ESP8266 Node MCU (Node Microcontroller Unit) board, and SMS gateway API (Application Programming Interface).

Chapter 4 gives information about the methodologies adopted for this work.

Chapters 5 and 6 are Results Discussion, and Conclusions related to performed work.

II. INTRODUCTION ABOUT SOFTWARE

2.1 About Arduino IDE

The Arduino IDE is open-source software developed by Arduino. cc and is primarily used to write, compile, and upload code for almost all Arduino modules. This is the official Arduino software, which makes compiling code so easy that even non-technical people can sweat in the learning process. Available on all operating systems. H. It runs on MAC, Windows, Linux, and the Java platform with built-in functions and commands that play important roles in debugging, editing, and compiling code. A variety of Arduino modules are available, including the Arduino Uno, Arduino Mega, Arduino Leonardo, and Arduino Micro. Each contains an onboard microcontroller that is programmed and retrieves information in the form of code. The main code, also known as a sketch, created on the IDE platform will eventually generate a hexadecimal file that will be transferred and uploaded to the controller on board. The IDE environment contains two main parts. Editor and compiler. The former writes the necessary code, which is then compiled and uploaded to a specific Arduino module. This environment supports C and C ++ languages.

To begin, go to the Arduino website and download the Arduino IDE. Make certain that you select the suitable version for your operating system (OS). Please visit the Arduino guide for a detailed getting started guide for each OS. Unzip the arduino.zip file to a folder on your PC after downloading it. Nothing needs to be installed; simply open the folder and double-click the.exe.

Connecting an Arduino board to your PC is simple. When working with Microsoft Windows:

1. Connect the USB cable, one end to the other, to the PC and the Arduino board.
2. When prompted, select "Browse my computer for driver," and then locate to the folder where you saved your original Arduino IDE download.
3. Select "Install anyhow" if an error message occurs indicating that the board is not a Microsoft-certified device.
4. At this point, your board should be ready for programming.

2.2 About HC-SR04 Ultrasonic Sensor

It is an ultrasonic sensor, also known as an ultrasonic transducer, that is made up of a transmitter and a receiver and is used to determine the distance between the target object and the sensor.
The distance between the object and the sensor is determined by the time it takes to emit and receive waves. It relies heavily on sound waves and "non-contact" technology. The required distance to the target object is measured without any harm, giving you accurate and precise information.
With a range of 2cm to 400cm, this sensor is employed in a variety of applications such as speed and direction measurement, wireless charging, humidifiers, medical ultrasonography, sonar, burglar alarms, and non-destructive testing.

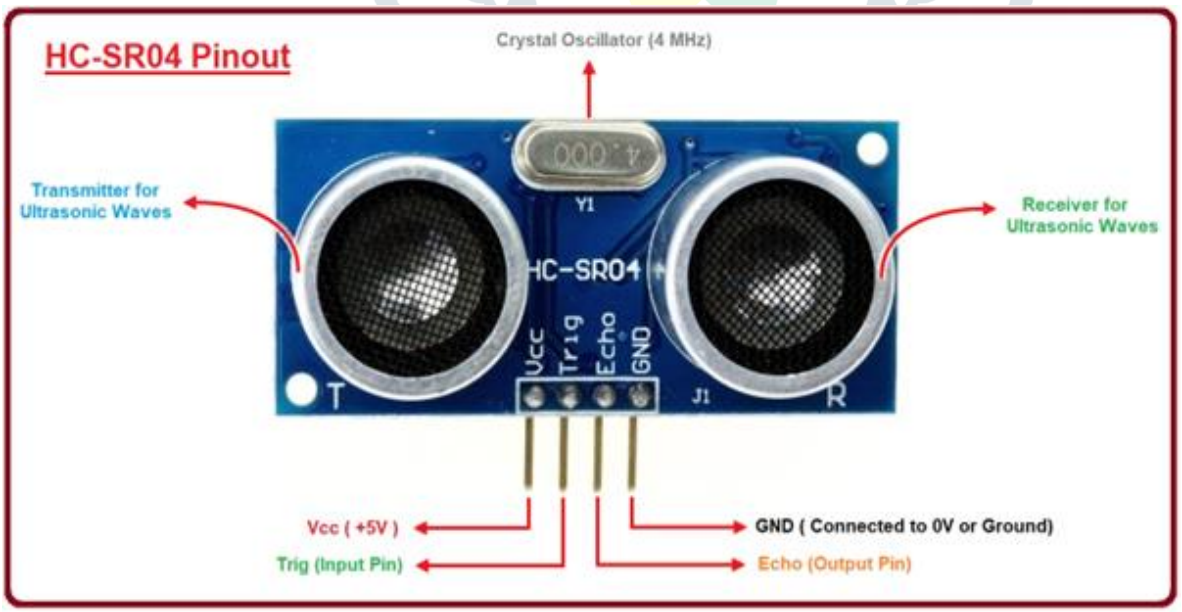


Fig. 2.2.1
Ultrasonic Sensor Schematic

Ultrasonic transducers having 4-pin connectors labelled Vcc, Trigger, Echo, and Ground are HC-SR04 (US) Ultrasonic Sensors. It is highly useful for correctly estimating the distance between two objects and primarily analyses sound waves. The sound wave transmission begins when the module is connected to 5V and the input pins are initialised, and the sound wave travels through the air and reaches the target item. These waves strike the item, bounce off, and are picked up by the module's receiver. The time it takes for these waves to return to the receiver determines the distance. The more the time, the greater the distance. When the Trig

pin is held high for 10 seconds, a wave is produced. These waves travel at the speed of sound and form an eight-cycle acoustic explosion, which the echo pen collects. The echo pins stay turned on for the duration of the time it takes for those waves to travel and bounce back to the receiver. This sensor measures the required distance and is mostly integrated with Arduino. To calculate the distance between two objects, use the formula below.

Formula :

$$S = (V \times t) / 2$$

where S is the required distance, V is the speed of sound, and t is the time it takes for the sound wave to hit an object and return. If the wave moves and bounces off the starting point, the time doubles, so you need to divide the value by two. To get the correct distance to the target item, divide this by two. HC-SR04 module with Arduino The HC-SR04 is mostly used in conjunction with various Arduino modules such as the Arduino Uno and Arduino Mega for accurate distance measurement.

You can connect your Arduino to this sensor using one of the following methods:

• To begin, power on the sensor using a stabilised 5VDC input to the sensor. Connect the power ground to the ground pin. The current drawn by the sensor is less than 15mA and has no effect on the rated current of the Arduino module, therefore the sensor module can also be powered by the Arduino 5V port. Connect the Trig and Echo pins to the Arduino board's I / O pins after you've determined the initial placement. To begin the measurement process, as previously stated, the Trig pin must be held high for 10us. The sensor module starts emitting sound waves from the transmitter at a rate of about 40,000 hertz per second.

As a result, when the wave bounces, the echo pen turns on until the receiver receives a sound wave. This time is calculated by the Arduino module.

2.3 API

API, which stands for Application Programming Interface, is a software middleman that enables two applications to communicate with one another. APIs are used every time you use an app like Facebook, send an instant message, or check the weather on your phone.

What is an API example?

When you use the app on your smartphone, it connects to the Internet and sends data to the server. The server then analyses the data, executes the appropriate operations, and delivers it back to your mobile phone. The application then analyses this information and displays it in a human-readable fashion. This is an application programming interface. This is all done using the API.

Let us use a well-known example to demonstrate this.

Consider sitting at a restaurant table with a menu from which to choose. The kitchen is a component of the "system" that processes the order. An crucial connection for sending orders to the kitchen and returning meals to the table is lacking. This is where the waiter or API comes in. The waiter is a messenger (or API) that takes your requests and orders and tells the kitchen (system) what to do. The waiter then returns the answer. In this case, it's food.

Here is an example of a real API. You may be accustomed to the process of searching for flights online.

You can select from a variety of alternatives, such as different cities, departure and return dates, and so on. Assume you book a flight on the airline's website. Choose the departure and return cities and dates, as well as the room class and other characteristics. To book a flight, go to the airline's website to access the database and see if seats are available on those dates and how much they cost.

But what if you don't use his airline's website (which provides immediate access to information)? What if you use an online travel service like Kayak or Expedia, which gathers data from a wide number of airline databases?

The travel service in this scenario connects with the airline's API. The API is an interface that allows this online travel service to request information from airline databases and reserve seats, luggage options, and other features, much like your friendly waiter. After that, the API receives the airline's response to your request and provides it to the online travel service accurately. This will show the most recent relevant information.

Modern API

Over the years, "API" has often been used to describe any kind of generic connectivity interface to an application. But these days, modern APIs have acquired a lot of traits with very valuable and useful features. understood.

Modern APIs are developer-friendly, easily accessible, and generally standards-compliant (usually HTTP and REST). understood. Treated like a product, not code. They are designed to be used by a specific audience (e.g. mobile developers), are documented, and are versioned to allow users to have specific expectations for maintenance and lifecycle.

Because they are much more standardized, there are stronger disciplines around security and governance and design, testing, building, maintaining, version performance, and scope monitoring and management lifecycles (SDLCs). Also, modern APIs for consumption and versioning are well documented.

Reasons to use API gateways

Most enterprise APIs are exposed through API gateways. It is common for API gateways to handle common tasks used in API service systems. User authentication, rate limits, and statistics.

The most basic API service accepts a remote request and responds with a response. But life isn't that straightforward. When hosting huge APIs, keep a variety of problems in mind.

1. To safeguard the API from abuse and misuse, utilise authentication services and rate limits.
2. I included analysis and monitoring tools because I want to know how people are interacting with the API.
3. If you intend to monetize the API, you must link to a billing system.
4. A microservices architecture may have been implemented. A single request in this situation may necessitate dozens of distinct application calls.
5. We've added some new API services and deprecated others over time, but consumers still want to discover them all in one spot.

In the face of all this complexity, your challenge is to provide your clients with a straightforward and dependable experience. API gateways allow you to decouple client interfaces from your backend implementation. When a client submits a request, API Gateway divides it into numerous requests, routes them to the relevant destinations, generates responses, and tracks everything.

The API gateway's role in API management

The API administration system includes the API gateway. All incoming requests are intercepted by the API gateway and routed through an API management system, which performs a number of needed duties.

2.4 About SMS Notification

We are using API of BhashSMS gateway API for sending alert SMS
API URL:
http://bhashsms.com/api/sendmsg.php?user=9971930997&pass=95c48ab&sender=SYSINT&phone=9971930997&text=your door opened&priority=ndnd&stype=normal

In this fast-paced world, staying connected is important to keep up with what's happening around you. Messaging or SMS is one of the most accepted methods today. SMS Gateway India is an important link between the sender and recipient of a message to complete the messaging outbound network. The SMS Gateway API regulates many factors such as delivery time gaps, and privacy protection and is an important factor to consider when choosing an SMS gateway provider in India.

Bulk SMS Gateway

Bhashsms.com is in control of the list of online SMS gateway providers. This bulk SMS gateway always provides excellent service. Bulk SMS gateway provider Bhashsms offers instant message delivery at a very affordable price. Use this bulk SMS gateway in India to send multiple messages online to any number of mobile users.

III. RESEARCH METHODOLOGY

3.1 Introduction

The protection and protection of cloud-connected devices such as home automation, SCADA equipment, surveillance cameras, and other technologies that are directly connected to the cloud is referred to as Internet of Things (IoT) security. IoT technology varies from mobile device technology (smartphones, tablets, etc.) in that gadgets automatically connect to the cloud. IoT security is the privacy and cybersecurity protection of previously poorly built gadgets. Recent data breaches demonstrate that IoT security is a top issue for the majority of manufacturers and developers.
Various strategies are used to confirm human security in any setting, including the home, office, or elsewhere. Alarms, whether wired or wireless, RFID keys, and biometric recognition are just a few examples. The first two techniques, on the other hand, have significant downsides, such as key loss and password loss. Although RFID-based password protection between the door and the control panel is effective, it is easily bypassed by advanced technologies such as intercepting data, interpreting commands, and replaying them to the control panel.

These signals can also be stuck to prevent an alert from being activated by sending radio noise to keep the signal from the sensors from reaching the control panel. Outstanding breakthroughs in the field of digital or virtual sciences have boosted the application of control engineering systems in daily life in recent years. As a result, systems capable of handling massive amounts of data, such as images and videos, are now available. Aside from fingerprint characteristics and facial recognition, distance recognition is a mechanism used in biometric recognition systems. The idea is to receive an SMS message and notify building security staff when the door unlocks in the honours' absence. In order for them to take the appropriate actions. This security can be particularly beneficial in device locking systems as well as other systems that require extra protection, such as banks and other private locations. To develop a low-cost and energy-efficient home security solution, we propose using IoT and distance recognition. We can calculate the distance to an item by taking the velocity of sound in the air and the transit time (the time between signal transmission and reception). Here's how it works:

Formula 1

$$\text{distance to an object} = ((\text{speed of sound in the air}) * \text{time})/2$$

We discovered that many technologies, including the Raspberry Pi Camera Board, Arduino, and Arduino UNO, and Android-based smartphones, as well as the ESP32 cam module, are available on the market due to the growing demand for an automatic security system that eliminates the need for manual intervention. The HC-SR04 Ultrasonic Sensor is utilised in this project in conjunction with the ESP8266 NodeMCU board and the Arduino core. The ultrasonic sensor uses sonar to detect the distance to an item.

3.2 Proposed System

The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an item. With a range of 2cm to 400cm (0.8inch to 157inch) and an accuracy of 0.3cm (0.1inch), this sensor is adequate for most amateur projects. Ultrasonic transmitter and receiver modules are also included in this module.

Fig. 4.2.1 shows the HC-SR04 ultrasonic sensor.

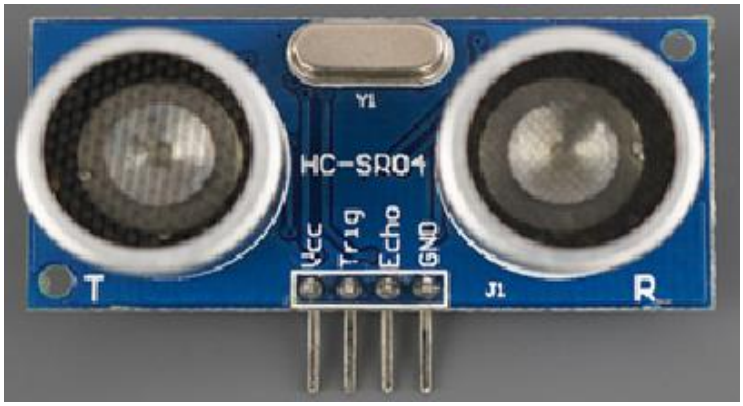


Fig. 4.2.1

Fig. 4.2.2 shows the other side of the sensor.

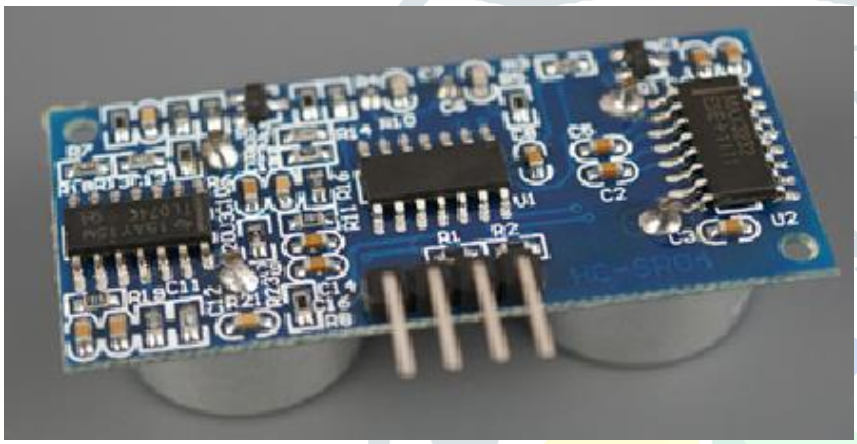


Fig. 4.2.2

Table 4.1 shows the key features and specs of the HC-SR04 ultrasonic sensor. For more information, you should consult the sensor's datasheet.

Table 4.1

HC-SR04 Ultrasonic Sensor Technical Data

5V DC	Power Supply
15 mA	Working Current
40 kHz	Working Frequency
4 meters	Maximum Range
2 cm	Minimum Range
15º	Measuring Angle
0.3 cm	Resolution

10uS TTL pulse	Trigger Input Signal
TTL pulse proportional to the distance range	Echo Output Signal
45(mm) x 20(mm) x 15(mm)	Dimensions

Table 4.2 Ultrasonic Sensor HC-SR04 Pinout

Powers the sensor (5V)	VCC
Trigger Input Pin	Trig
Echo Output Pin	Echo
Common GND	GND

3.3 What Is the Function of the HC-SR04 Ultrasonic Sensor?

Sonar is used by the ultrasonic sensor to determine the distance to an object. This is how it works:
A high-frequency sound is emitted by the ultrasound transmitter (trig pin) (40 kHz).
The sound is carried through the air. If it finds an object, it returns to the module.
The reflected sound is received by the ultrasound receiver (echo pin) (echo).

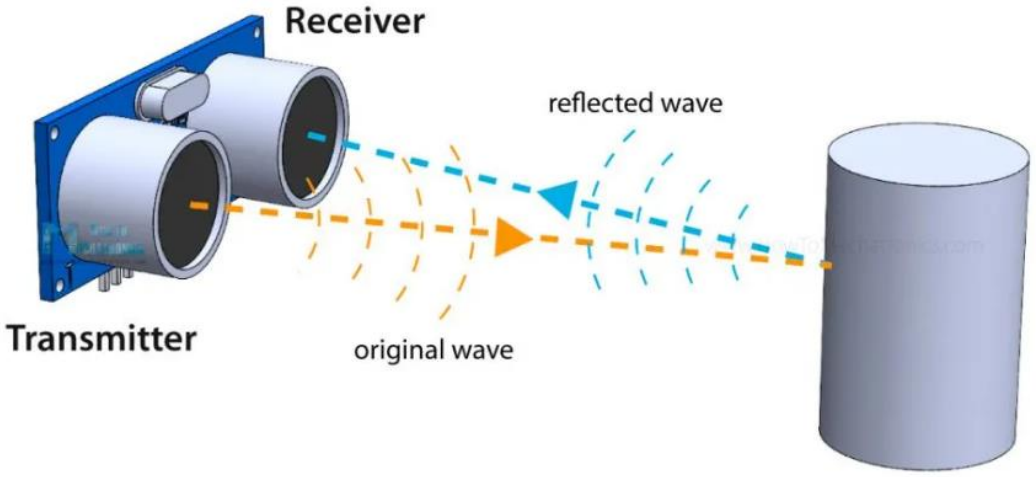


Fig. 4.3.1

Taking the velocity of sound in the air and the travel time, we may compute the distance to an object (the time between signal transmission and reception).

The formula is as follows:

distance to an object = ((speed of sound in the air) * time)/2

• Sound speed in air at 20oC (68oF) = 343m/s

3.4 Parts Required

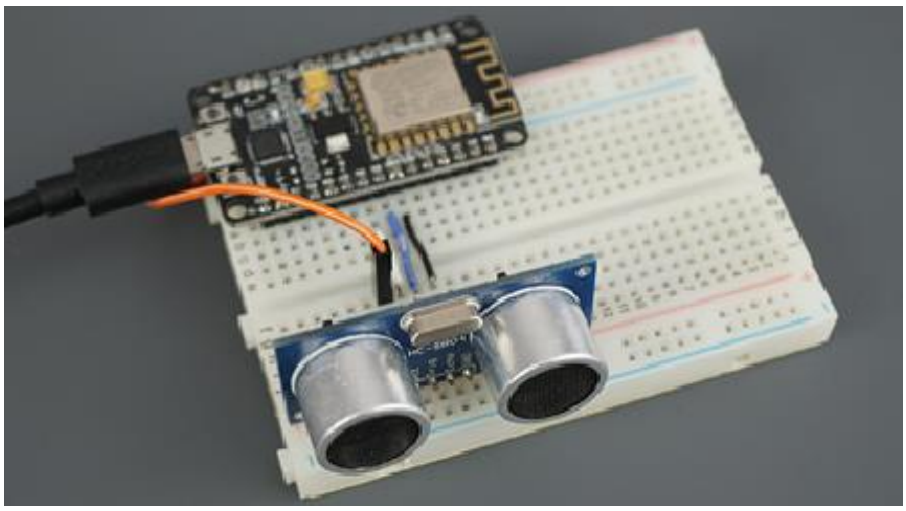
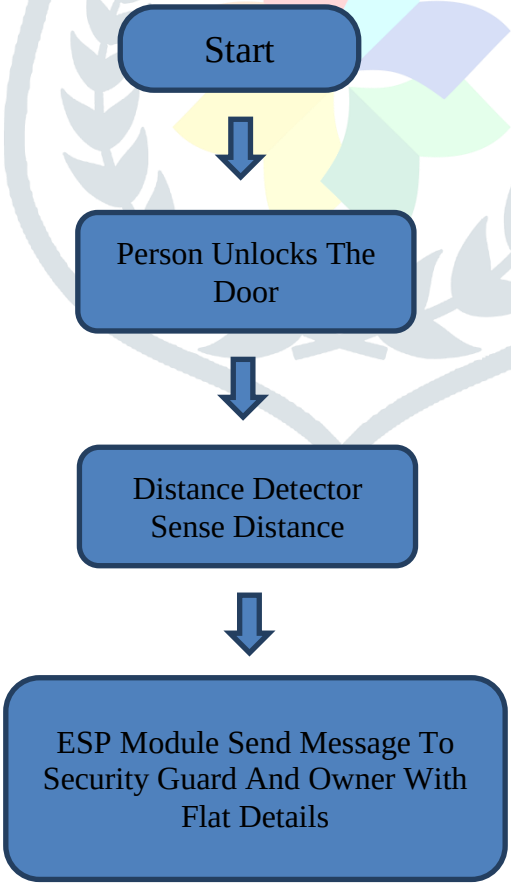


Fig. 4.4.1

To complete the setup, we need the below-mentioned parts:
Jumper wires
HC-SR04 Ultrasonic Sensor
ESP8266 (read Best ESP8266 development boards)
Breadboard
API gateway
SMS notification enablement

To understand the flow of the system below is the flow chart:



As shown in Fig. 4.4.2, connect the HC-SR04 ultrasonic sensor to the ESP8266. The Trig pin is connected to GPIO 5 and the Echo pin is connected to GPIO 18, but you can use any other suitable pins.

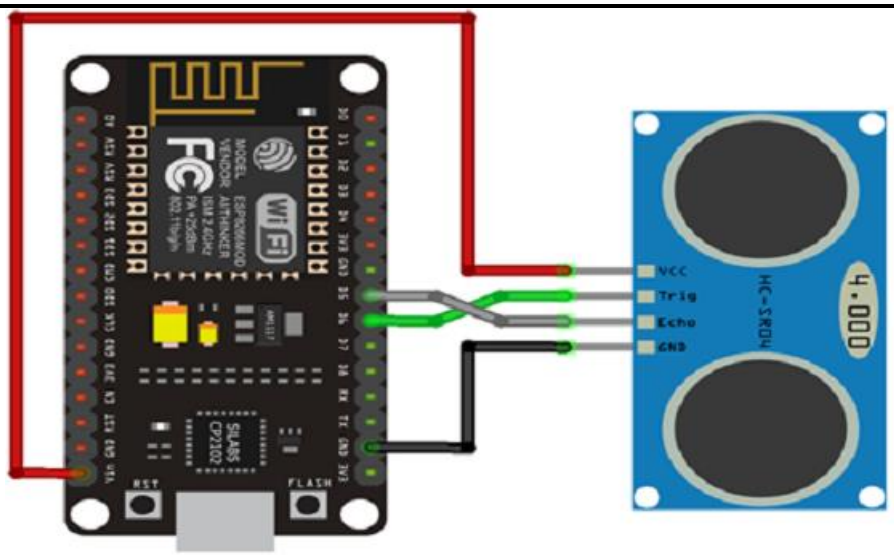


Fig. 4.4.2

Fig.4.4.3 shows the Arduino Ultrasonic Sensor and LCD Display

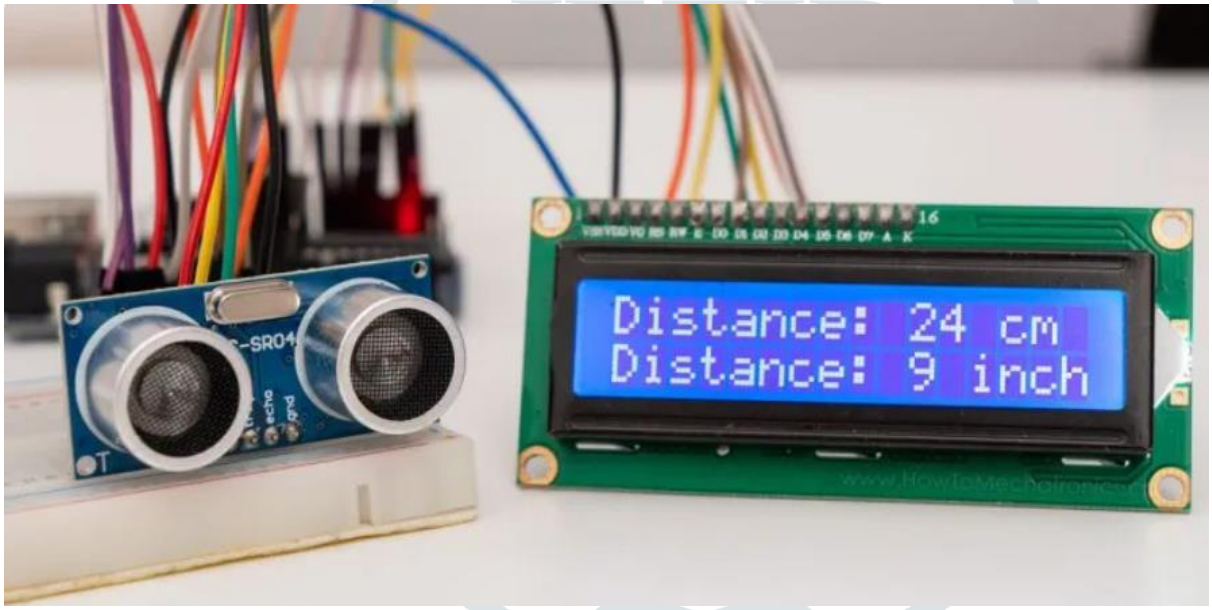


Fig. 4.4.3

Schematic – NodeMCU ESP8266 with HC-SR04 Ultrasonic Sensor

ESP8266	**Ultrasonic Sensor**
VIN	VCC
GPIO 12 (D6)	Trig
GPIO 14 (D5)	Echo
GND	GND

3.5 Getting the Arduino IDE Ready

We programmed the Arduino IDE to work properly by installing the ESP8266 add-on. Below is the code for estimating the distance to an item using the HC-SR04 Ultrasonic Sensor and the ESP8266.

setup() - The setup() function is called once at the beginning of the application. When the programme starts, it defines initial environment properties such as screen size and is used to load media such as graphics and fonts. Each programme can only have one setup() function, which cannot be invoked again after the first execution.

trig Pin - A high-frequency sound (40 kHz) is generated from the ultrasonic transmitter (trigger pin). Sound propagates through air. If the object is found, jump to the module and return.

echo Pin - An ultrasonic receiver (echo pin) picks up the reflected sound (echo).

digitalWrite() - Writes a HIGH or LOW value to a digital pin. If the pin is configured as OUTPUT in pinMode(), its voltage is set to the appropriate value: 5V for HIGH (or 3.3V for 3.3V boards) and 0V (ground) for LOW. is.

pinMode() - Sets the given pin to be an input or output.

delaymicroseconds() - The programme is paused for the time (in microseconds) indicated by the parameter.

pulseIn() - Read the pin's pulse (HIGH or LOW).

loop() - After you've used the setup() function to initialise and establish the initial values, the loop() function does exactly what the name implies: it loops one thing at a time, allowing the programme to modify and respond. Actively controls the Arduino board.

serial.println() and serial.print() - Serial.print() prints only numbers or strings, and Serial.println() prints with a newline character.

setup() - At the start of the sketch, the setup() function is invoked. Use it to initialise variables, change modes, access libraries, and so on. When the Arduino board is powered on or reset, the setup() code is only called once.

Program Code

```
void setup() {
  // put your setup code here, to run once:

}
const int trigPin = 12;
const int echoPin = 14;

//define sound velocity in cm/uS
#define SOUND_VELOCITY 0.034
#define CM_TO_INCH 0.393701

long duration;
float distanceCm;
float distanceInch;

void setup() {
  Serial.begin(115200); // Starts the serial communication
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
}

void loop() {

  // Clears the trigPin

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 microseconds

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```
  // Reads the echoPin, returns the sound wave travel time in
microseconds

  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance

  distanceCm = duration * SOUND_VELOCITY/2;

  // Convert to inches

  distanceInch = distanceCm * CM_TO_INCH;

  // Prints the distance on the Serial Monitor

  Serial.print("Distance (cm): ");
  Serial.println(distanceCm);
  Serial.print("Distance (inch): ");
  Serial.println(distanceInch);

  delay(1000);
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

To calculate the sound wave travel time, we use the pulseIn() function:

Formula 2:

duration = pulseIn(echoPin, HIGH);

The pulseIn() method reads the HIGH or LOW pulse from a pin. It takes as arguments the pin and the pulse state (either HIGH or LOW). The pulse length is returned in microseconds. The pulse length is the time it took to reach the object plus the time it took to return.
Using Tools > Boards, upload the code to the board. Remember to choose the relevant COM port under Tools > Ports.

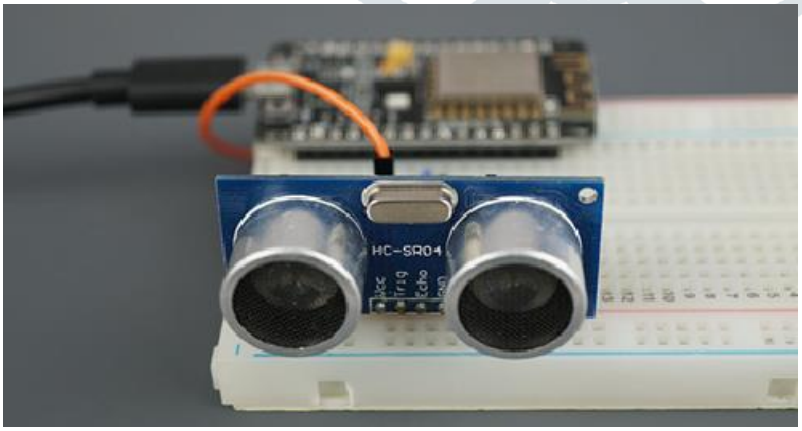


Fig. 4.5.1
ESP8266 with HC-SR04 and OLED Display

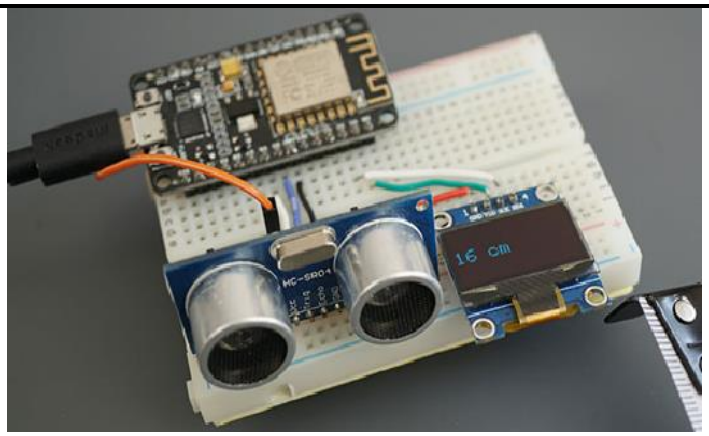


Fig. 4.5.2

This section will demonstrate a basic example using the ESP8266 to display distance on an I2C OLED display.

3.6 SMS Notification

To get the alert on your mobile we have set up this SMS notification functionality. Nowadays mobile phone is available to everyone. So, for alerts, mobile phones are the better approach.

The distance sensor continuously gets the distance of the nearest object. When no object is present it returns higher values. When a door opens and at a point when the distance between sensor and door remains less than 5 inches, the ESP8266 node MCU module is notified and it invokes an SMS gateway API which in turn sends an SMS to the user.

Below is the code for this

Program Code

Input for distance = less than 6

Input for Mobile Number = Dummy number (8888888888 we can use the mobile number on which we want to send notification.)

Timer = set to 5 seconds.

Sets the trigPin as an Output - The ultrasonic transmitter (trigger pin) produces a high-frequency sound (40kHz). Sound propagates through the air. When the object is found, it jumps back to the module. An ultrasonic receiver (echo pen) picks up the reflected sound (echo).

Sets the echoPin as an Input - When an ultrasonic burst is broadcast (in response to Trig), the Echo pin goes high and stays high until the sensor detects an echo of the burst, at which point it drops low. The distance can be measured by keeping track of how long the echo pen is up. Gnd is the ground pin. Gnd is the ground pin.

SSID - A service set identifier (SSID) is a string that gives a wireless local area network a unique name (WLAN). The SSID is also known as the "network name" in some cases. When operating many separate networks in the same physical area, this name allows stations to connect to the desired network.

ESP8266WiFi.h - The ESP8266WiFi library is included in the first line of the sketch, #include ESP8266WiFi.h>. This library contains Wi-Fi routines for the ESP8266 that can be used to connect to a network.

ESP8266HTTPClient.h - Allow ESP8266HTTPClient.h to make HTTP requests.

WiFiClient.h - With the Arduino WiFi Shield, enable network connectivity (both local and internet). This software lets you create a server and a client for sending and receiving UDP packets over WiFi. The shield can connect to both open and encrypted networks (WEP, WPA).

Serial.begin - The baud rate for serial data transfer is established by serial.begin(). The data rate in bits per second is indicated by the baud rate. The default baud rate for Arduino is 9600 bps (bits per second).

```
# include < ESP8266WiFi.h >
# include < ESP8266HTTPClient.h >
# include < WiFiClient.h >
const char * ssid = "REPLACE_WITH_YOUR_SSID";
const char * password = "REPLACE_WITH_YOUR_PASSWORD";
const int trigPin = 12;
const int echoPin = 14;

//define sound velocity in cm/uS

 # define SOUND_VELOCITY 0.034
 # define CM_TO_INCH 0.393701
long duration;
float distanceCm;
float distanceInch;

//Your Domain name with URL path or IP address with path

String serverName = "http://192.168.1.106:1880/update-sensor";

// the following variables are unsigned longs because the time
, measured in
// milliseconds will quickly become a bigger number than can b
e stored in an int.

unsigned long lastTime = 0;

// Timer set to 10 minutes (100000)
//unsigned long timerDelay = 600000;
// Set timer to 5 seconds (5000)
```

```
unsigned long timerDelay = 5000;
void setup() {
Serial.begin(115200);
WiFi.begin(ssid, password);
Serial.println("Connecting");
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.print("Connected to WiFi network with IP Address: ");
Serial.println(WiFi.localIP());
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
}
int calculateDistance() {

    // Clears the trigPin

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    // Sets the trigPin on the HIGH state for 10 microseconds

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time i
n microseconds

    duration = pulseIn(echoPin, HIGH);

    // Calculate the distance

    distanceCm = duration * SOUND_VELOCITY / 2;

    // Convert to inches

    distanceInch = distanceCm * CM_TO_INCH;
    return distanceInch;
}
void loop() {

// Send an HTTP POST request depending on timerDelay

if ((millis() - lastTime) > timerDelay) {
```

```
//Check WiFi connection status

if (WiFi.status() == WL_CONNECTED) {
WiFiClient client;
HTTPClient http;
int distanceInch = calculateDistance();

//when the door will open distance between the module and the
door will be less than 6 inches. Send  SMS to the owner

if (distance < 6) {
String serverPath = http: //bhashsms.com/api/sendmsg.php?user=
8888888888(no. on which notification we want to configure)
&pass=95c48ab&sender=SYSINT&phone=8888888888&text=your door op
ened&priority=ndnd&stype=normal

// Your Domain name with URL path or IP address with path

http.begin(client, serverPath.c_str());

// Send HTTP GET request

int httpResponseCode = http.GET();
if (httpResponseCode > 0) {
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
String payload = http.getString();
Serial.println(payload);
}
else {
Serial.print("Error code: ");
Serial.println(httpResponseCode);
}
}

// Free Resources

http.end();
}
else {
Serial.println("WiFi Disconnected");
}
lastTime = millis();
}
}
```

Program Code with different input:

Input for distance = less than 8

Input for Mobile Number = Dummy number (9999999999 we can use the mobile number on which we want to send notification.)

Timer = set to 8 seconds.

Sets the trigPin as an Output - A high-frequency sound is emitted by the ultrasonic transmitter (trigger pin) (40kHz). Sound travels through the air. When the item is located, the programme returns to the module. The reflected sound is picked up by an ultrasonic receiver (echo pen) (echo).

Sets the echoPin as an Input - When an ultrasonic burst is broadcast (in response to Trig), the Echo pin goes high and stays high until the sensor detects an echo of the burst, at which point it drops low. The distance can be measured by keeping track of how long the echo pen is up. Gnd is the ground pin. Gnd is the ground pin.

SSID - A service set identifier (SSID) is a string that gives a wireless local area network a unique name (WLAN). The SSID is also known as the "network name" in some cases. When operating many separate networks in the same physical area, this name allows stations to connect to the desired network.
ESP8266WiFi.h - The ESP8266WiFi library is included in the first line of the sketch, #include ESP8266WiFi.h>. This library contains Wi-Fi routines for the ESP8266 that can be used to connect to a network.

ESP8266HTTPClient.h - Allow ESP8266HTTPClient.h to make HTTP requests.

WiFiClient.h - With the Arduino WiFi Shield, enable network connectivity (both local and internet). This software lets you create a server and a client for sending and receiving UDP packets over WiFi. The shield can connect to both open and encrypted networks (WEP, WPA).

Serial.begin - serial.begin() sets the baud rate for serial data communication. Baud rate indicates the data rate in bits per second. The default Arduino baud rate is 9600 bps (bits per second).

```
# include < ESP8266WiFi.h >
# include < ESP8266HTTPClient.h >
# include < WiFiClient.h >
const char * ssid = "REPLACE_WITH_YOUR_SSID";
const char * password = "REPLACE_WITH_YOUR_PASSWORD";
const int trigPin = 12;
const int echoPin = 14;

//define sound velocity in cm/uS

 # define SOUND_VELOCITY 0.034
 # define CM_TO_INCH 0.393701
long duration;
float distanceCm;
float distanceInch;

//Your Domain name with URL path or IP address with path

String serverName = "http://192.168.1.106:1880/update-sensor";

// the following variables are unsigned longs because the time
, measured in
// milliseconds will quickly become a bigger number than can b
e stored in an int.

unsigned long lastTime = 0;

// Timer set to 10 minutes (100000)
//unasigned long timerDelay = 600000;
// Set timer to 5 seconds (5000)

unsigned long timerDelay = 8000;
void setup() {
Serial.begin(115200);
WiFi.begin(ssid, password);
Serial.println("Connecting");
while (WiFi.status() != WL_CONNECTED) {
delay(800);
Serial.print(".");
}
```

```
Serial.println("");
Serial.print("Connected to WiFi network with IP Address: ");
Serial.println(WiFi.localIP());
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
}
int calculateDistance() {

    // Clears the trigPin

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin on the HIGH state for 10 microseconds

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Reads the echoPin, returns the sound wave travel time i
n microseconds

    duration = pulseIn(echoPin, HIGH);
    // Calculate the distance

    distanceCm = duration * SOUND_VELOCITY / 2;

    // Convert to inches
    distanceInch = distanceCm * CM_TO_INCH;
    return distanceInch;
}
void loop() {

// Send an HTTP POST request depending on timerDelay
if ((millis() - lastTime) > timerDelay) {

//Check WiFi connection status

if (WiFi.status() == WL_CONNECTED) {
WiFiClient client;
HTTPClient http;
int distanceInch = calculateDistance();

//when the door will open distance between the module and the
door will be less than 6 inches. Send  SMS to the owner

if (distance < 8) {
String serverPath = http: //bhashsms.com/api/sendmsg.php?user=
9999999999 (no. on which notification we want to configure)
```

```
&pass=95c48ab&sender=SYSINT&phone=9999999999&text=your door op
ened&priority=ndnd&stype=normal

// Domain name with URL path or IP address with path

http.begin(client, serverPath.c_str());

// Send HTTP GET request
int httpResponseCode = http.GET();
if (httpResponseCode > 0) {
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
String payload = http.getString();
Serial.println(payload);
}

else {
Serial.print("Error code: ");
Serial.println(httpResponseCode);
}
}

// Free resources
http.end();
}
else {
Serial.println("WiFi Disconnected");
}
lastTime = millis();
}
}
```

IV. RESULTS AND DISCUSSION

The proposed method can be deployed in near-real time to IoT-based smart home monitoring systems. Investigate unlawful access in smart homes built on an integrated architecture of sensors, cameras, and unique hardware. The system functions on two levels: the hardware interface and the software interface.

This system was designed to prevent theft from the apartments when owners are away from the home. Once the owner got the SMS he could alert the security guard for a security check at the flat. Being a not very much known object like a security camera, thieves may not be aware of this system and will continue to do the ransacking and hence more chances of getting caught.

Possible Enhancements

1. Application may need a stop action possibly from an android app where a user should be able to disable the notifications. Once notifications are off system will not send SMS.
2. There could be an android app for security also where he gets to know the flat number where he needs to check, and he can mark the action taken so that the user gets notified easily.
3. We can connect the system for notification on email also with house details.
4. We can configure a multi-theft system like distance with face capture.

IV. CONCLUSION

1. This system was very cheap with component costs of just 300 rupees, it has a scope to be industrialized with some changes and offer enormous benefits for the company and peace of mind for users of this system. There is a scope to use such systems in other applications also –
2. Heat control Alert systems (When the temperature of a chamber should not go beyond some range. If it happens system should be able to shut the system down)

3. Water wastage management systems: In Housing societies where there is so much water flow a central system using a Water sensor, node MCU modules, and central apps can be created where administrators can see the overflow sitting in one room and upon clicking a button in app motor will stop.
4. Automatic Irrigation: This concept can also be used when farmers need to do irrigation at a point in time. They can use a similar system with Node MCU, Motors, Relay Switches, and an app. Farmer can tap at his app when he wants to start irrigation and stop it using his mobile.

V. ACKNOWLEDGMENT

I'd want to take this time to thank my guide, Mr. Rajesh Rajaan, for giving wonderful direction, motivation, and inspiration during the project work. This effort would not have been a success without his helpful guidance. I'd also like to thank all of my classmates for their helpful comments and discussions.

VI. REFERENCES

[1] P. Timse, P. Aggarwal, P. Sinha, N. Vora, Face recognition based door lock system using opencv and C# with remote access and security features, Int. J. Eng. Res. Appl. 4 (2014) 52–57.
[2] L. Zhang, An improved approach to security and privacy of RFID application system, in: Wireless Communications, Networking and Mobile Computing International Conference, 2005, pp. 1195–1198.
[3] Bagchi, T., Mahapatra, A., Yadav, D., Mishra, D., Pandey, A., Chandrasekhar, P., & Kumar, A. (2022). Intelligent security system based on face recognition and IoT. Materials Today: Proceedings.
[4] T. Gunawan, M.H.H. Gani, F. Rahman, M. Kartiwi, Development of face recognition on raspberry pi for security enhancement of smart home system, Indon. J. Electr. Eng. Inform. (2017).
[5] Mocrii, D., Chen, Y., & Musilek, P. (2018). IoT-based smart homes: A review of system architecture, software, communications, privacy, and security. Internet of Things, 1, 81-98.
[6] V. Kushnir, B. Koman, V. Yuzevych, IoT image recognition system implementation for blind peoples using esp32, mobile phone and convolutional neural network, in: 2019 XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT), 2019, pp. 183–187.
[7] Keshta, I. (2022). AI-driven IoT for smart health care: Security and privacy issues. Informatics in Medicine Unlocked, 30, 100903.
[8] Sisavath, C., & Yu, L. (2021). Design and implementation of security system for smart home based on IOT technology. Procedia Computer Science, 183, 4-13.
[9] Aldawira, C. R., Putra, H. W., Hanafiah, N., Surjarwo, S., & Wibisurya, A. (2019). Door security system for home monitoring based on ESp32. Procedia Computer Science, 157, 673-682.
[10] Saxena, N., & Varshney, D. (2021). Smart Home Security Solutions using Facial Authentication and Speaker Recognition through Artificial Neural Networks. International Journal of Cognitive Computing in Engineering, 2, 154-164.
[11] Mohan, J., & Rajesh, R. (2021). Enhancing home security through visual cryptography. Microprocessors and Microsystems, 80, 103355.
[12] Zhang, L., Zhou, H., Kong, R., & Yang, F. (2005, September). An improved approach to security and privacy of RFID application system. In Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005. (Vol. 2, pp. 1195-1198). IEEE.
[13] Zhu, Z., & Cheng, Y. (2020). Application of attitude tracking algorithm for face recognition based on OpenCV in the intelligent door lock. Computer Communications, 154, 390-397.