IoT PROJECT REPORT

ANTI-THEFT SECURITY SYSTEM

Guided By

Prof Anurag Lakhlani

REPRESENTATIVES

Mudra Suthar (1741022)

Hardi Trivedi (1741092)

Index

1 Introduction and Motivation

We plan to design a project which works as an anti-theft system and provides a sense of security and safety to the common people. We plan to build this system by using the concepts of IoT. It will provide security by providing a smoke alert system, fire alert system, automatic lock alert system.
We aim to build a system that can be used not only in our homes but in an office or corporate buildings, banks extra that will act as an anti-theft system and provide security to the valuables and belongings.
We will design a system which will take care of financial cost as well as which will be easy for the people to use.

Everyone wants safety but in the present scenario, nothing is safe not even in their own houses/offices. Home/Office is a place where we keep our assets and our capital. But we can never be sure about the security of that asset behind us and the possibilities of intrusion are increasing day by day. We generally lock houses/offices when going out of the house/office. But just locking the home/office is not enough, there must be a system which takes safety into our home/office, belongings and income from theft are the necessary requirements for an Anti-theft security system and keep track of the activities and report to the owner accordingly and work according to the response of the owner.

Our learning Goals from this project are concepts of the Internet Of Things. We would learn how to form a physical network of things or objects—devices, other items—embedded with electronics, software, sensors, and network connectivity that enables these things or objects to collect and exchange data.

Whenever the thief enters the house/bank or any corporate office, and steps on the floor immediately it is sensed by the sensor which passes on the signal to the Arduino controller. It will provide a smoke alert, fire alert system along with an automatic lock system.
This system is suitable for small personal area surveillance. i.e. personal office cabin, bank locker room, parking entrance. Whenever the motion is detected through.

The main Advantage of the project is Easy to implement, Low cost with High quality.

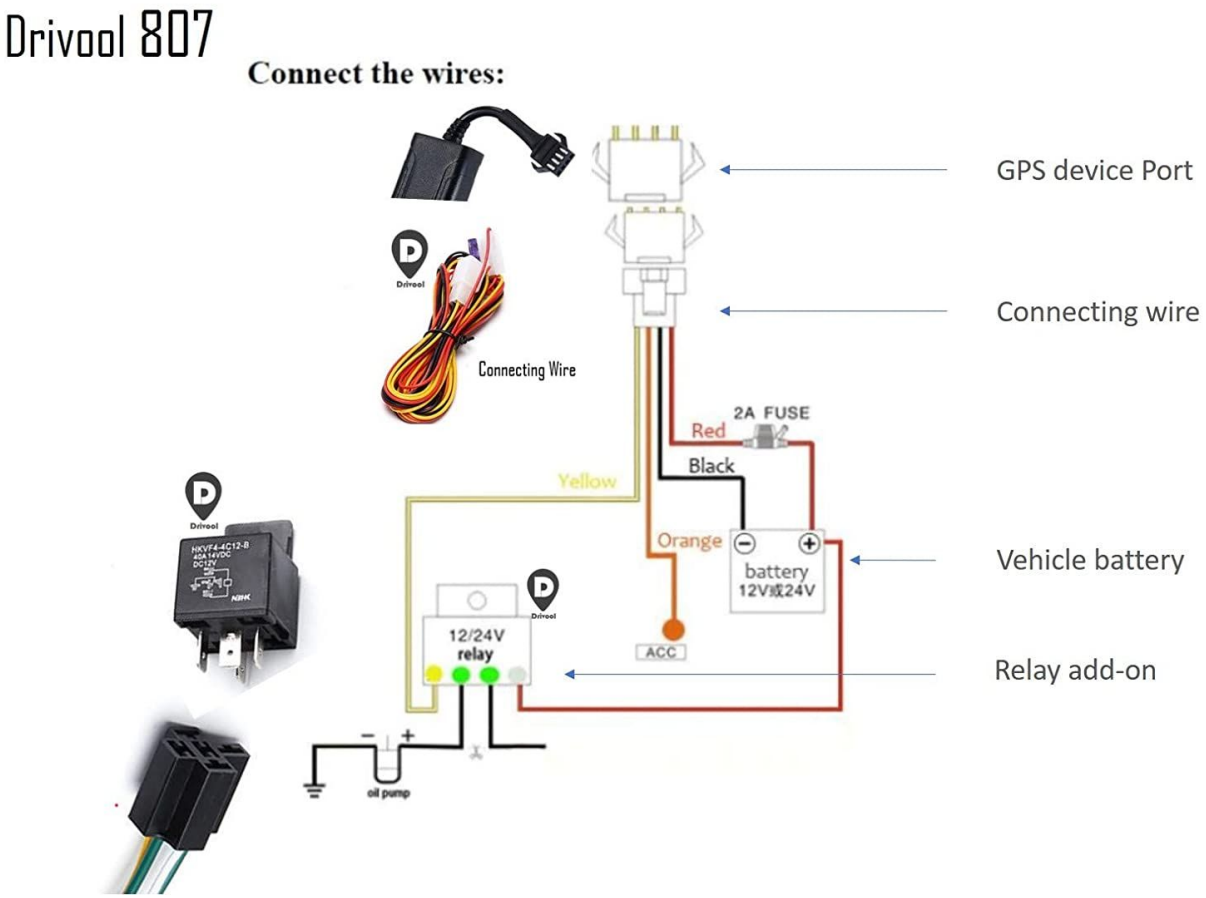
2 Market Survey

There have been some products in the market which work on the concept of IoT based anti-theft security system. None of the exact products in the market are available for sale. Given below are some examples::1. Wireless Anti-theft Security / IoT Based Anti-theft Safety and Security / Intrusion Detection / Door Window Sensor / Motion Sensor

The device contains one Gateway, one door/window sensor, and one motion sensor. The gateway is a link between your mobile phone application (IOS/Android) and other sensors.

Through the mobile application, you will get immediate notification if there is any intrusion in your house. Motion Detection Distance is around 9 -12 Meters and Door/Window Sensors detects the opening and closing of the door/window.

Wireless System: Every sensor is wireless and they all are connected through WiFi Network/GPRS. Gateway has a GPRS slot in it and it has eight hours of battery backup in case of a power cut off.
The device also has an Inbuilt Alarm, The Gateway has an inbuilt alarm in it which automatically gets generated if any sensor detects an intrusion.
2.This device behaves similarly to the anti-theft security system but it operates on vehicles, its application is found in vehicles.



Drivool 807 is a Mini Smallest GPS Vehicles Tracker Device Locator, a robust and versatile GPS device.

It is very easy to hide and install in the vehicles. Having an open wire connection for power source you can use in Bike, Motorcycle, Auto, Truck, Bus, Mini Van, or any kind of Vehicle.

ST-901(M) is another version of the same device and supported by the Drivool Platform.

It works based on existing GSM/GPRS network and GPS satellites, this product can locate and monitor any remote targets by SMS or GPRS. Real-time tracking location by SMS/GPRS, GSM quad-band frequency, With ACC to detect ignition, Set authorized number, Over-speed alarm, and many versatile features with DRIVOOL ‘uMove’ App**.**

3 . The Anti-theft Security System has been an interesting topic for research for years. Many papers are published in the view to find out a better and a robust system, Few research articles, and their screenshots are attached below.

-

ANTITHEFT VEHICLE TRACKING AND CONTROL SYSTEM BASED IOT

Ahmed Sabeeh Ali[1*], Ali Hussein Hasan[2], Hussein Atiyah Lafta[3]

[1*]College of Science for Woman – University of Babylon, Babil, Iraq. ahmedsabeeh84@gmail.com
[2]College of Computer Science and Information Technology – University of Sumer, Thi-Qar, Iraq.
[3]College of Science for Woman – University of Babylon, Babil, Iraq.

Received: 16.03.2020 *Revised: 18.04.2020* *Accepted: 20.05.2020*

Abstract

This paper introduces a low-cost and reliable Anti-Theft Vehicle tracking System based on IoT services with microcontrollers for tracking vehicles in real-time, controls it in case of theft and notify the nearby police station in a short period. Low-cost, effective and reliable modules and technologies are used in the implementation of the proposed system. The system consists of two parts, the embedded system, and web application. The embedded system includes GPS, GSM, GPRS, relay, and microcontroller, it placed inside the vehicle in a hidden place so that thief cannot know its location. The web application with Google map is designed for tracking vehicles in real-time and the host server transmit data between the embedded system and web application. Haver sine Formula is implemented to determine the nearest police station by comparing the last coordinate received of the vehicle with coordinates of police stations. The microcontroller manages the operation of embedded system, GPS get current location of the vehicle; GSM/GPRS Module used to send data to the host server to monitor the vehicle location by Google map embedded with web application and can control the embedded system remotely by sending SMS to stop vehicle fuel line by relay. The system is equipped with rechargeable batteries to ensure continuous operation when disconnecting the vehicle battery.

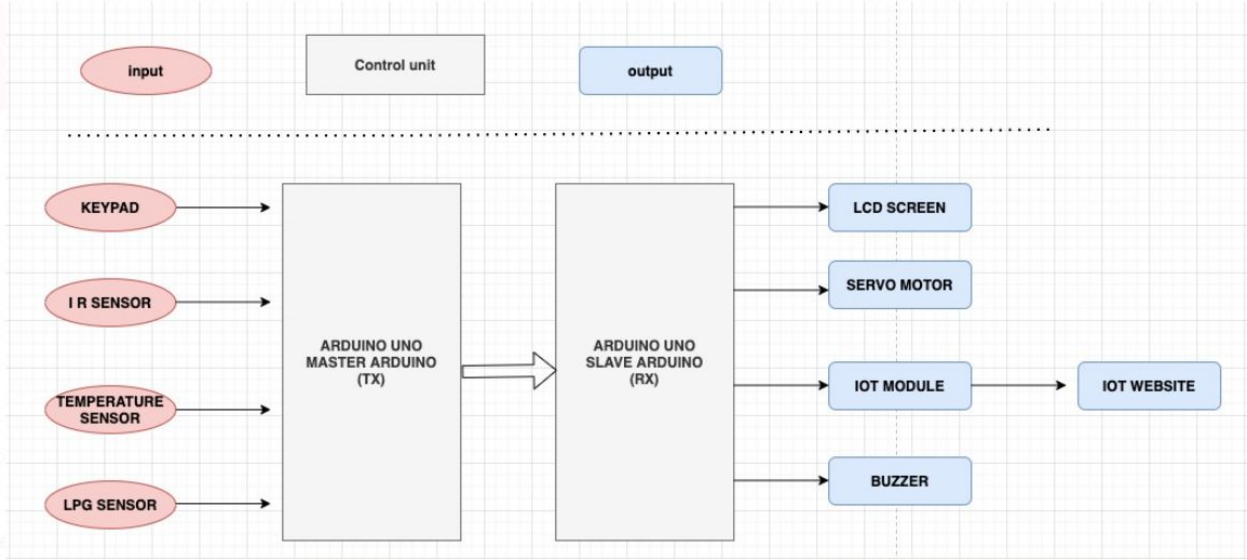
Keywords: Internet of Things. GPS, GSM, Tracking System, Microcontroller, Haversine Formula.

- **An Advanced IOT based Antitheft Security System with Video Monitoring Facility**

ABSTRACT

Security is a too much important thing to be concerned in our day-to-day life. Everyone wants to be secured as much as possible. Knowing our home or shop is secure provides us peace of mind. We know now a day's theft has become a major issue. In this project we design an advanced electronic security system by using small PIR and IR sensors built around the Node MCU controller. PIR sensor sense the presence of intruder & Controller reads the signal from sensors and if intruder is detected, it compares the detected image with predefined images in the database then it turns on the buzzer as well as making a notification to predefined number. At the same time the video of intruder can also be monitored and make them anesthetic.

3 Block Diagram



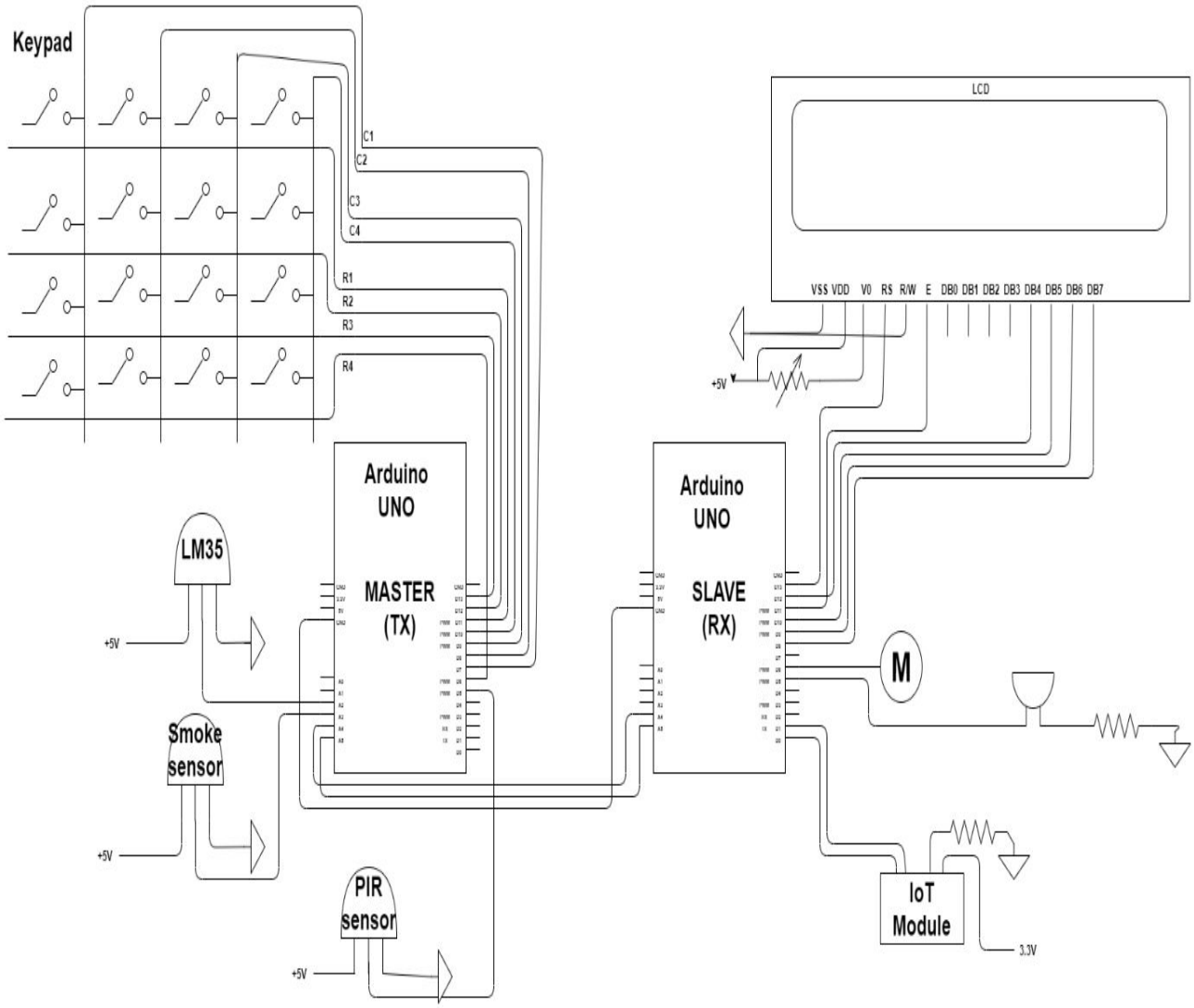
EXPLANATION OF BLOCK DIAGRAM

Our main aim of the project is to provide a sense of safety and security to the users with the help of automated systems and IoT. We plan to design a system that not only will alert the user on fire and smoke leakage but also provides an automated lock system to avoid any kind of theft.

All the devices which will be used to create this project are as follows:

- Input Devices
  - ➔ Keypad: For user input
  - ➔ Temp sensor: For Fire alert
  - ➔ Smoke sensor: to detect smoke leakage
  - ➔ PIR sensor: for anti-theft automated lock system
- Control Units: Arduino Uno (we are using two Arduino's for our project)
- Output Devices:
  - ➔ LCD screen: for displaying password
  - ➔ DC Motor: will function as the door
  - ➔ Buzzer: will ring when fire alert, theft, or smoke is detected
  - ➔ IoT module: will send the data to the IoT cloud

4 Circuit Diagram

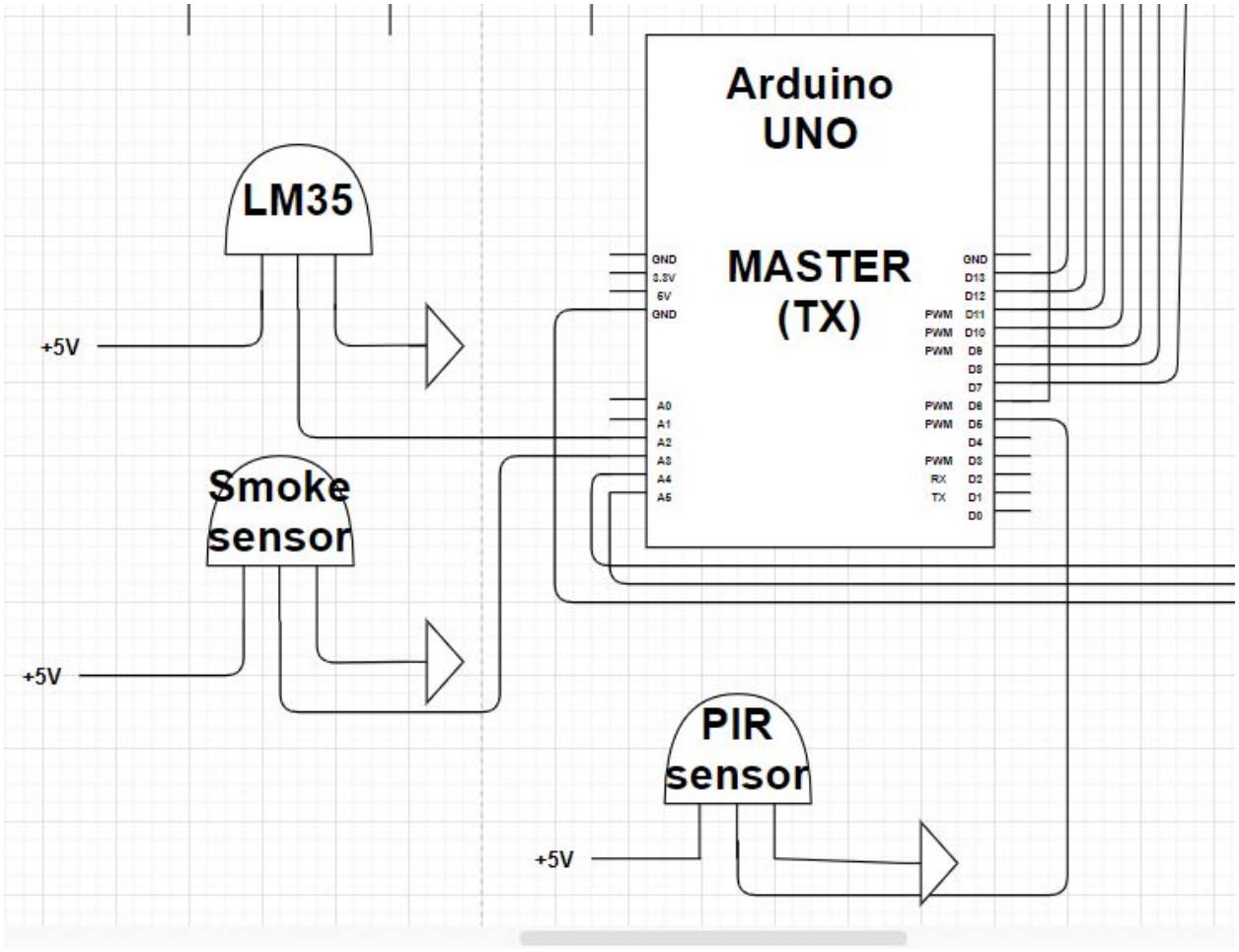


PIN CONNECTION OF MASTER ARDUINO:

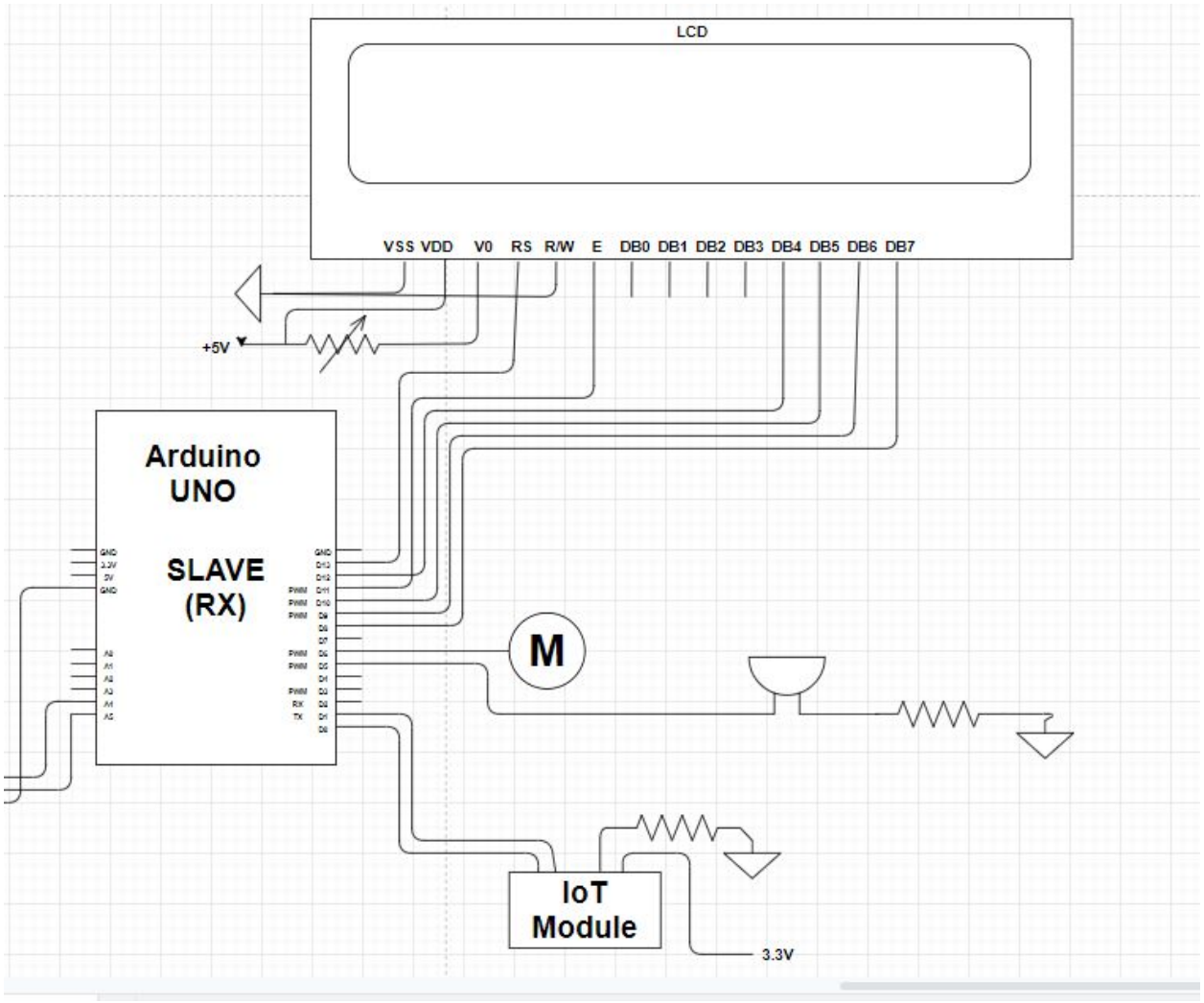
All the input devices are connected to the master Arduino and their connection are given as follows:

- **Keypad**: Keypad has eight pins (4-row pins and 4 column pins). These pins are connected to digital pins 6 to 13 of the Arduino.

- **Temp sensor:** The temperature sensor has three pins. One is connected to the power source, one to the A2 of the analog pin, and the last to the ground.
- **Smoke sensor:** The Smoke sensor has three pins. One is connected to the power source, one to the A3 of the analog pin, and the last to the ground.
- **PIR sensor:** The motion sensor has three pins. One is connected to the power source, one to the 5 of the digital pin, and the last to the ground



PIN CONNECTION OF THE SLAVE ARDUINO:



All the output devices are connected to the slave Arduino, the pin connection of it as follows:

- **LCD screen:** The connection of the LCD screen is as follows:
  - ➔ VSS: connected to the ground
  - ➔ VDD: connected to a power source
  - ➔ V0: connected to the potentiometer to control contrast
  - ➔ RS: digital pin 13
  - ➔ R/W: connected ground
  - ➔ E: digital pin 11
  - ➔ DB4 to DB7: digital pin 12,10,9,8.

- **IoT module:** This is connected to the transmission and receiver pins of Arduino. And the occ pin is connected to a 3.3-volt power source and the last pin is connected to the ground
- **DC Motor:** The DC Motor will be connected to the digital pin 6
- **Buzzer:** The buzzer will be connected to the digital pin 5.

INPUT-OUTPUT AND SELECTION CRITERIA

All the devices which will be used to build this project are as follows:

- **Input devices:**
  - **Keypad:** We are using a 16 button keypad with 0-9 digits and special characters '*' and '#' and letters A to D. User will type the selected password in this keypad. This keypad will take this password input and send it to Arduino in the form of electric signals.
  - **PIR sensor:** Pir sensor or passive infrared motion sensor senses motion in front of it. It will be connected with the door and will sense whether anyone is not trying to break the look.
  - **Temp sensor:** we will be using TMP36 as a temp sensor. It is also available on the tinkercad which is the online platform on which we will simulate. It will send different voltages on the basis of the temperature of the surrounding. It is the best temperature sensor available on the tinkercad.
  - **Gas Sensor:** We will be using a Wilson gas sensor, as it is the only gas sensor that can detect carbon monoxide, alcohol, and methane. Hence it is ideal for our project

- **Control units:** We will be using two Arduino. Ideally, our work can be done using Arduino mega, but Arduino mega is not supported by tinkercad. So two Arduino will be used.
  - **Master Arduino**: This Arduino also called the transmitter Arduino, will be connected to all the input devices. This will collect the input data and transmit it to the slave Arduino or the receiver Arduino.
  - **Slave Arduino**: this Arduino will receive the data from the master Arduino and it will generate the output on the output devices as per the data.

- **Output devices:**
  - **LCD Screen:** We will be using a 16*2 LCD display. So we will be capable of using two lines of 16 characters. This is enough to display a

password and show the message of whether the password which was entered by the user is correct or wrong.

- ➔ **DC Motor:** Here the DC motor will act as the automated door. The DC motor will be connected to the Arduino via H-Bridge which will control the direction of the rotation of the DC Motor, and hence the will automatically open or close the door. This is the most effective and easiest implementation to apply the principles of the door.
- ➔ **IoT Module:** IoT module will send all the data to the IoT cloud and hence the user can know whether his/her house is safe from anywhere. By just checking on the cloud.
- ➔ **Buzzer:** Piezo buzzer will alert when a fire, smoke, or theft is detected.

5. Arduino Uno Features

The Arduino Uno is one kind of microcontroller board based on ATmega328, and Uno is an Italian term which means one. Arduino Uno is named for marking the upcoming release of the microcontroller board namely Arduino Uno Board 1.0. This board includes digital I/O pins-14, a power jack, analog i/ps-6, ceramic resonator-A16 MHz, a USB connection, an RST button, and an ICSP header. All these can support the microcontroller for further operation by connecting this board to the computer. The power supply of this board can be done with the help of an AC to DC adapter, a USB cable, otherwise a battery

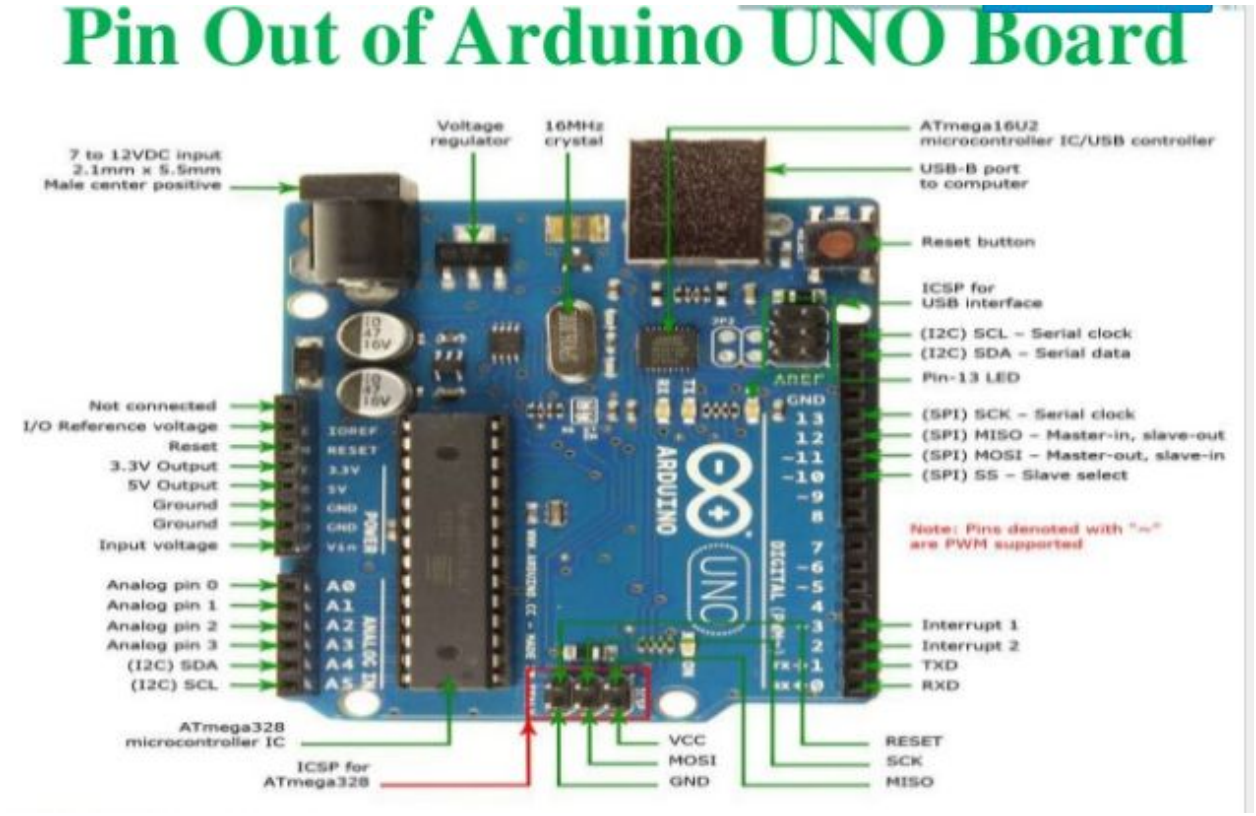
Features of Arduino Uno Board

The **features of Arduino Uno ATmega328** include the following.

- The operating voltage is 5V
- The recommended input voltage will range from 7v to 12V
- The input voltage ranges from 6v to 20V
- Digital input/output pins are 14
- Analog i/p pins are 6
- DC Current for each input/output pin is 40 mA
- DC Current for 3.3V Pin is 50 mA
- Flash Memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK Speed is 16 MHz

Physical Characteristics

The physical characteristics of an Arduino board mainly include length and width. The printed circuit board of the Arduino Uno length and width are 2.7 X 2.1 inches, but the power jack and the USB connector will extend beyond the previous measurement. The board can be attached on the surface otherwise case with the screw holes.



Power Supply

The **Arduino Uno power supply** can be done with the help of a USB cable or an external power supply. The external power supplies mainly include AC to DC adapter otherwise a battery. The adapter can be connected to the Arduino Uno by plugging into the power jack of the Arduino board. Similarly, the battery leads can be connected to the Vin pin and the GND pin of the POWER connector. The suggested voltage range will be 7 volts to 12 volts.

Input & Output

The 14 digital pins on the Arduino Uno can be used as input & output with the help of the functions like pinMode(), digitalWrite(), & Digital Read().

Pin1 (TX) & Pin0 (RX) (Serial): This pin is used to transmit & receive TTL serial data, and these are connected to the ATmega8U2 USB to TTL Serial chip equivalent pins.

Pin 2 & Pin 3 (External Interrupts): External pins can be connected to activate an interrupt over a low value, change in value.

Pins 3, 5, 6, 9, 10, & 11 (PWM): This pin gives 8-bit PWM o/p by the function of analogWrite().

SPI Pins (Pin-10 (SS), Pin-11 (MOSI), Pin-12 (MISO), Pin-13 (SCK): These pins maintain SPI-communication, even though offered by the fundamental hardware, is not presently included within the Arduino language.

Pin-13(LED): The inbuilt LED can be connected to pin-13 (digital pin). As the HIGH-value pin, the light-emitting diode is activated, whenever the pin is LOW.

Pin-4 (SDA) & Pin-5 (SCL) (I2C): It supports TWI-communication with the help of the Wire library.

AREF (Reference Voltage): The reference voltage is for the analog i/ps with analogReference().

Reset Pin: This pin is used for reset (RST) the microcontroller.

Memory

The memory of this Atmega328 Arduino microcontroller includes flash memory-32 KB for storing code, SRAM-2 KB EEPROM-1 KB.

Communication

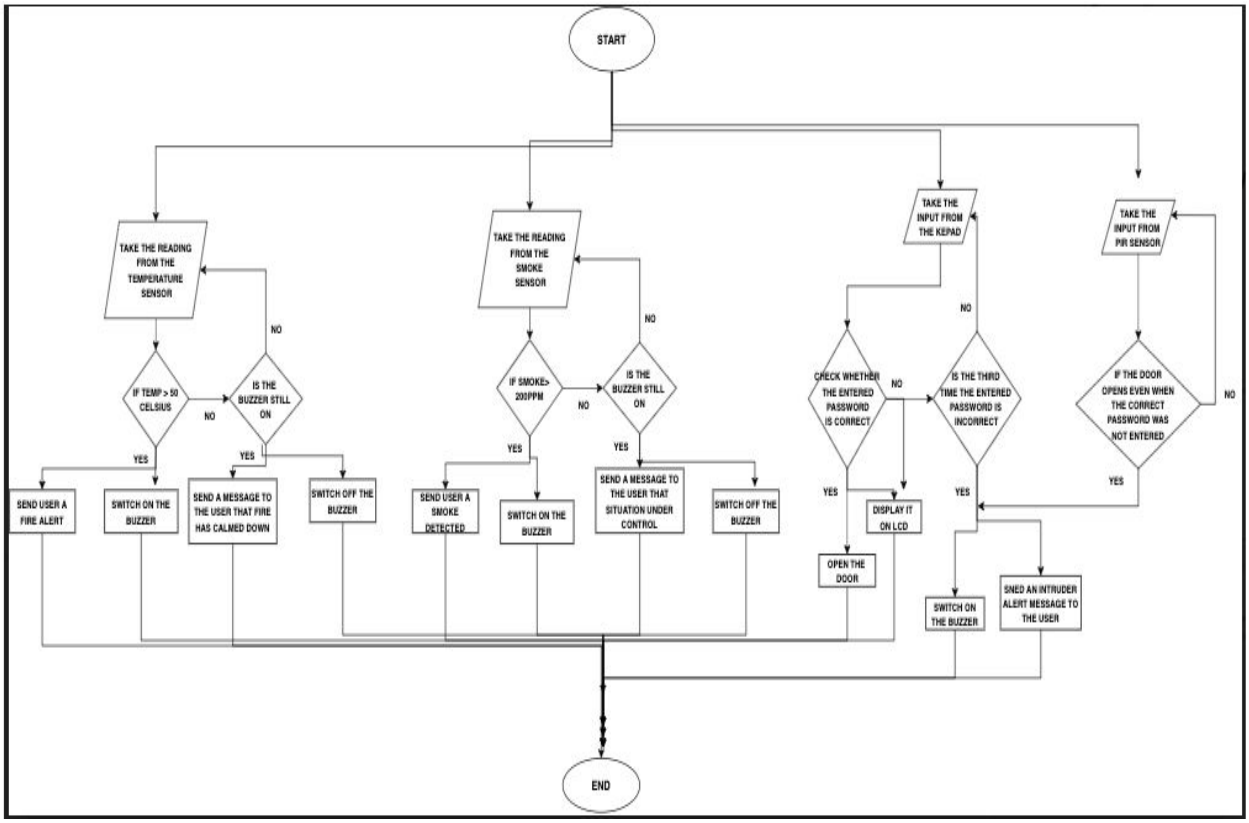
The Arduino Uno ATmega328 offers UART TTL-**serial communication**, and it is accessible on digital pins like TX (1) and RX (0). The software of an Arduino has a serial monitor that permits easy data. There are two LEDs on the board like RX & TX which will blink whenever data is being broadcasted through the USB.

A SoftwareSerial library permits for serial communication on Arduino Uno digital pins and the ATmega328P supports TWI (I2C) as well as **SPI-communication**. The Arduino software contains a wired library for simplifying the utilization of the I2C bus

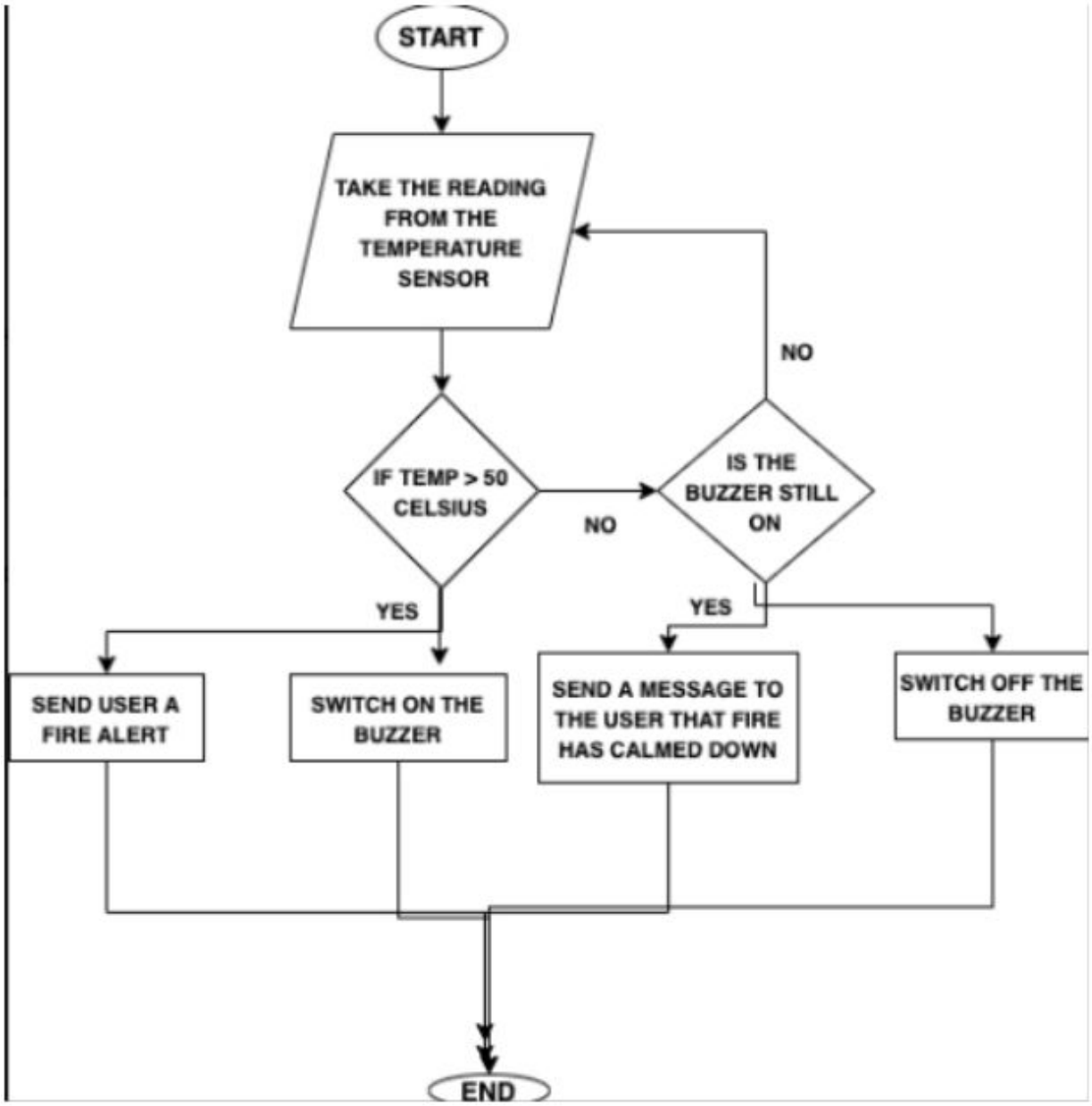
Thus, this is all about the Arduino **Uno.** From the above information finally, we can conclude that this is an 8-bit ATmega328P microcontroller. It has different components like serial communication, **crystal oscillator**, the voltage regulator for supporting the microcontroller. This board includes a USB connection, digital I/O pins-14, analog i/p pins-6, a power-barrel jack, a reset button, and an ICSP header.

6.PROGRAM FLOWCHART

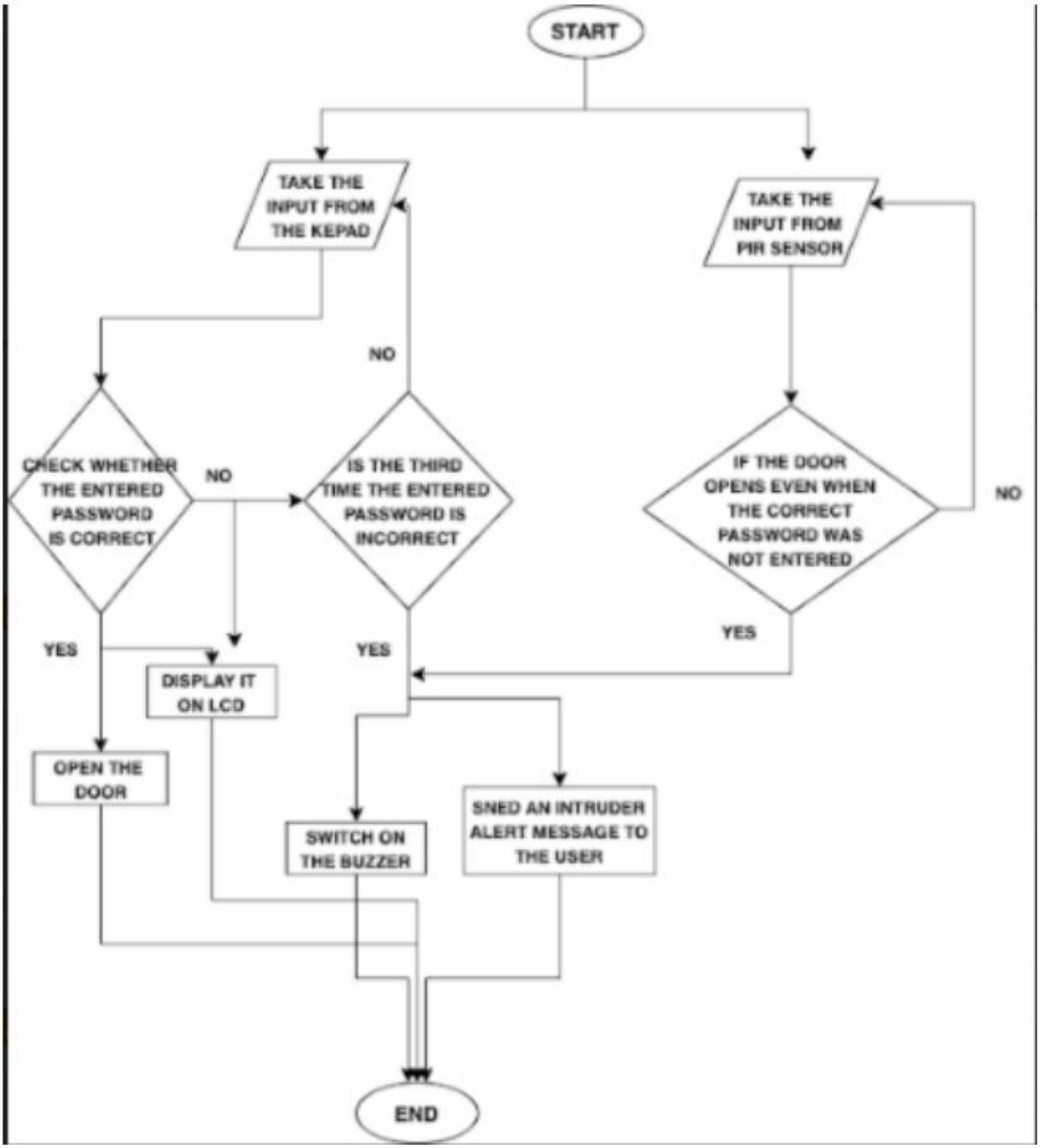
Whole System ::



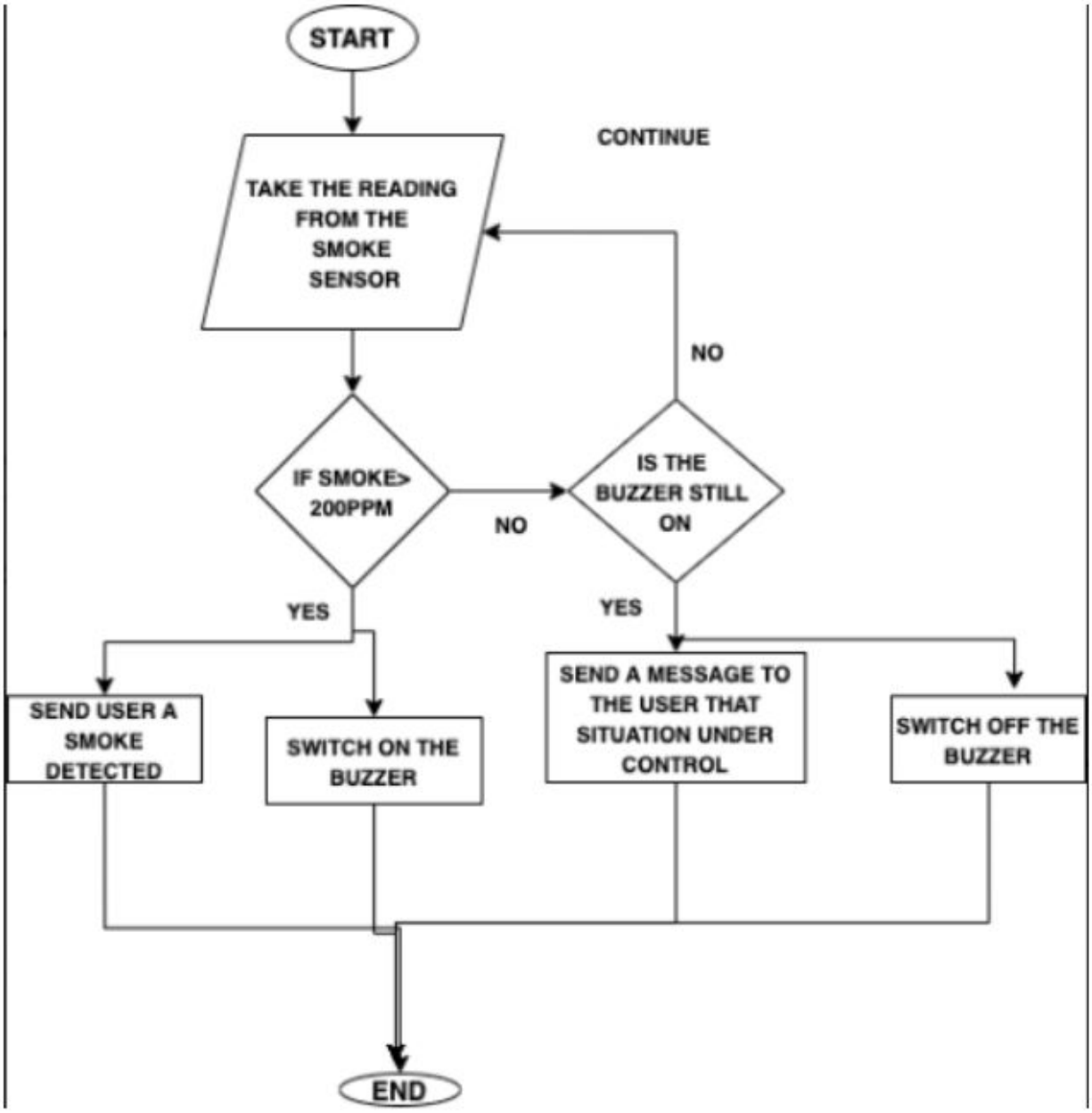
Fire Alert::



Intruder Alert::

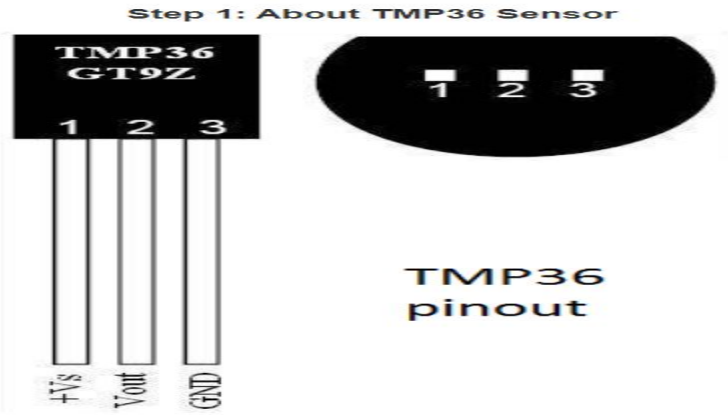

START
TAKE THE INPUT FROM THE KEPAD
TAKE THE INPUT FROM PIR SENSOR
NO
CHECK WHETHER THE ENTERED PASSWORD IS CORRECT
NO
IS THE THIRD TIME THE ENTERED PASSWORD IS INCORRECT
IF THE DOOR OPENS EVEN WHEN THE CORRECT PASSWORD WAS NOT ENTERED
NO
YES
YES
YES
DISPLAY IT ON LCD
OPEN THE DOOR
SWITCH ON THE BUZZER
SNED AN INTRUDER ALERT MESSAGE TO THE USER
END

Smoke Alert::


START
CONTINUE
TAKE THE READING FROM THE SMOKE SENSOR
NO
IF SMOKE> 200PPM
NO
IS THE BUZZER STILL ON
YES
YES
SEND USER A SMOKE DETECTED
SWITCH ON THE BUZZER
SEND A MESSAGE TO THE USER THAT SITUATION UNDER CONTROL
SWITCH OFF THE BUZZER
END

7.Sensors

1. TMP36 Temperature Sensor:



The TMP35/TMP36/TMP37 are low voltage, precision centigrade temperature sensors. They provide a voltage output that is linearly proportional to the Celsius (centigrade) temperature. The TMP35/ TMP36/TMP37 do not require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the −40°C to +125°C temperature range.

FEATURES::

- Low Voltage Operation (+2.7 V to+5.5 V)
- Calibrated Directly in °C
- 10 mV/8°C Scale Factor (20 mV/8°C on TMP37)
- ±2°C Accuracy OverTemperature (typ)
- ±0.5°C Linearity (typ)

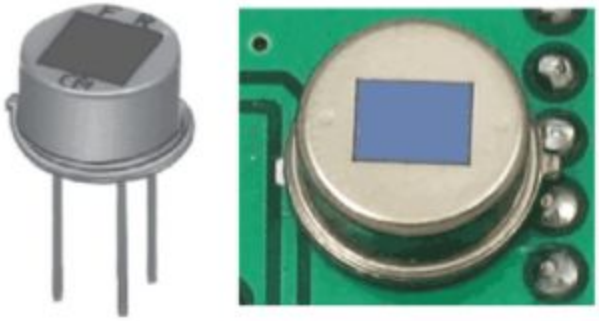
- Stable with Large Capacitive Loads
- Specified -40 °C to +125 °C, Operation to +150 °C
- Less than 50 µA Quiescent Current
- Shutdown Current 0.5 µA max

2.PIR MOTION SENSOR

PIR sensors allow you to sense motion. They are used to detect whether a human has moved in or out of the sensor's range. They are commonly found in appliances and gadgets used at home or for businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

Following are the advantages of PIR Sensors −

- Small in size
- Wide lens range
- Easy to interface
- Inexpensive
- Low-power
- Easy to use
- Do not wear out



Functional Description

If the active elements of the PIR sensor are exposed to a change in the surrounding temperature field, electrical charges are separated within the sensor elements. The voltage across the sensors controls a J-FET source follower impedance converter and thus modulates the output current of the PIR detector. The spectral sensitivity of the sensor is controlled by the optical transfer characteristics of the window in the case and has been optimized to pick up radiation of the human body.

Applications

♦ alarm systems

♦ consumer electronics

♦ human body detection

♦ automatic switches

3.Smoke Sensor MQ2

The **MQ-2** is a flammable gas and the smoke sensor detects the concentrations of combustible gas in the air and outputs its reading as an analog voltage. The sensor can measure concentrations of flammable gas of 300 to 10,000 ppm. The MQ-2 gas sensor is sensitive to LPG, i-butane, propane, methane, alcohol, Hydrogen, and smoke. They are used in gas leakage detecting types of equipment in family and industry and in the portable gas detectors.

Specifications

- Supply Voltage:5V
- Sensitive to H2, LPG, CH4, CO, Alcohol, Smoke or Propane
- Analog and Digital Output
- Digital Out is High or Low based on an adjustable preset threshold.

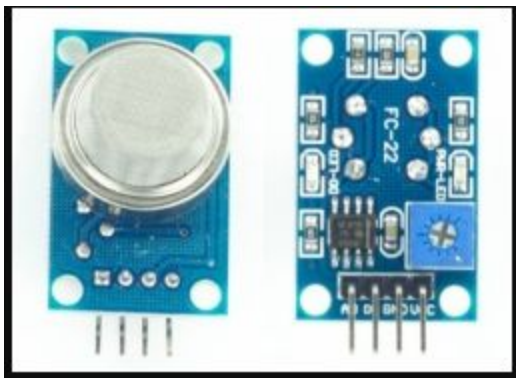
How To connect a smoke sensor with Arduino?

- MQ2 Smoke sensor Module
- Arduino Uno
- Arduino IDE(1.0.6 Version)

Hardware Connections

The connections are made as follows:

- Vcc to 5V
- GND to GND
- A0 to Analog0


FC-22

8.ACTUATORS AND DISPLAYS

2.SERVO motor

Servo motor is a type of motors whose output shaft can be moved to a specific angular position by sending it a coded signal. The servo motor will maintain the position of the shaft as long as you keep applying the coded signal. When you change the coded signal, the angular position of the shaft will change.

A common type of servo provides position control. Servos are commonly electrical or partially electronic in nature, using an electric motor as the primary means of creating mechanical force. Other types of servos use hydraulics, pneumatics, or magnetic principles.

Specifications::
The servo motors have their own characteristics like other types of motors, which are voltage, current, operating speed, torque, control pulse, resolution, and pulse, and weight.

Power Supply Voltage and Current :

The power supply voltage and current values are specified for each type of servo motors and depend on the application. The common RC servo motors powered from supplies in the range of 4 – 6 V and 100 mA - 2 A.

Operating Speed :

The operating speed of a servo motor is defined as the time required for the shaft to reach a specified position. Common servos have operating speeds in the range of 0,05 to 0,2 s/60 degrees.

Torque :

Typical values of torques of servo motors are in the range of 0,5 to 10 kg/cm.

Control Pulse :

The control pulse is referred to as the type of pulse used to position the shaft. Two main types of control pulses used in RC applications: center position in 1-2 ms and 1,25-1,75 ms.

Resolution :

It defines the precision with which the shaft is positioned when it receives an external command signal. Typical servo motors have resolutions in the range from 1 degree to 10 degrees.

Size and Weight :

This characteristic is important in the mechanic design of projects. Typical RC servo motors have a weight range between 15 and 200g.

3.LCD Display

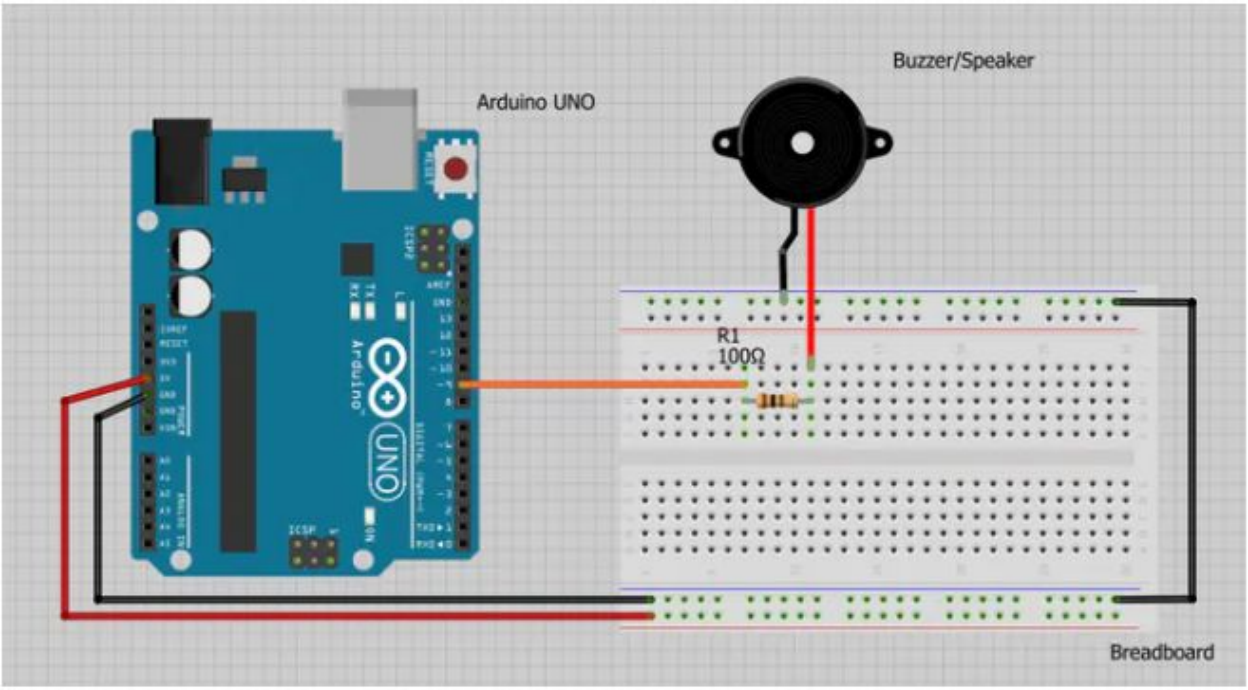
Liquid crystal display (LCD) is an electronic display device. We have used the 16*2 LCD display.

No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	V0	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

4.Buzzer / piezo speaker

Specifications and details

TYPE (UNIT)	MSPS42A29W12	MSPS42A31W24
Min. Sound Output at 10cm (dB)	100	105
Rated Voltage VDC	12	24
Operating Voltage VDC	3~20	18~30
Resonant Frequency (Hz)	2900+/-500	3100+/-500
Max. Current Consumption (mA)	15	20
Tone Nature	Continuous	Continuous
Operating Temperature (C)	-20~+60	-20~+60
Storage Temperature (C)	-30~+70	-30~+70
Weight (g)	13.4	13.4



5.Keypad

WHAT IS KEYPAD?

A keypad is a set of buttons arranged in a block or “pad” which bear digits, symbols, or alphabetical letters. Pads mostly containing numbers are called a numeric keypad.

The 4 x 4 matrix keypad usually is used as input in a project. It has 16 keys in total, which means the same input values. It is ultra-thin, easy to interface with any microcontroller, and has an adhesive backing for easy mounting for a variety of applications.

FEATURES

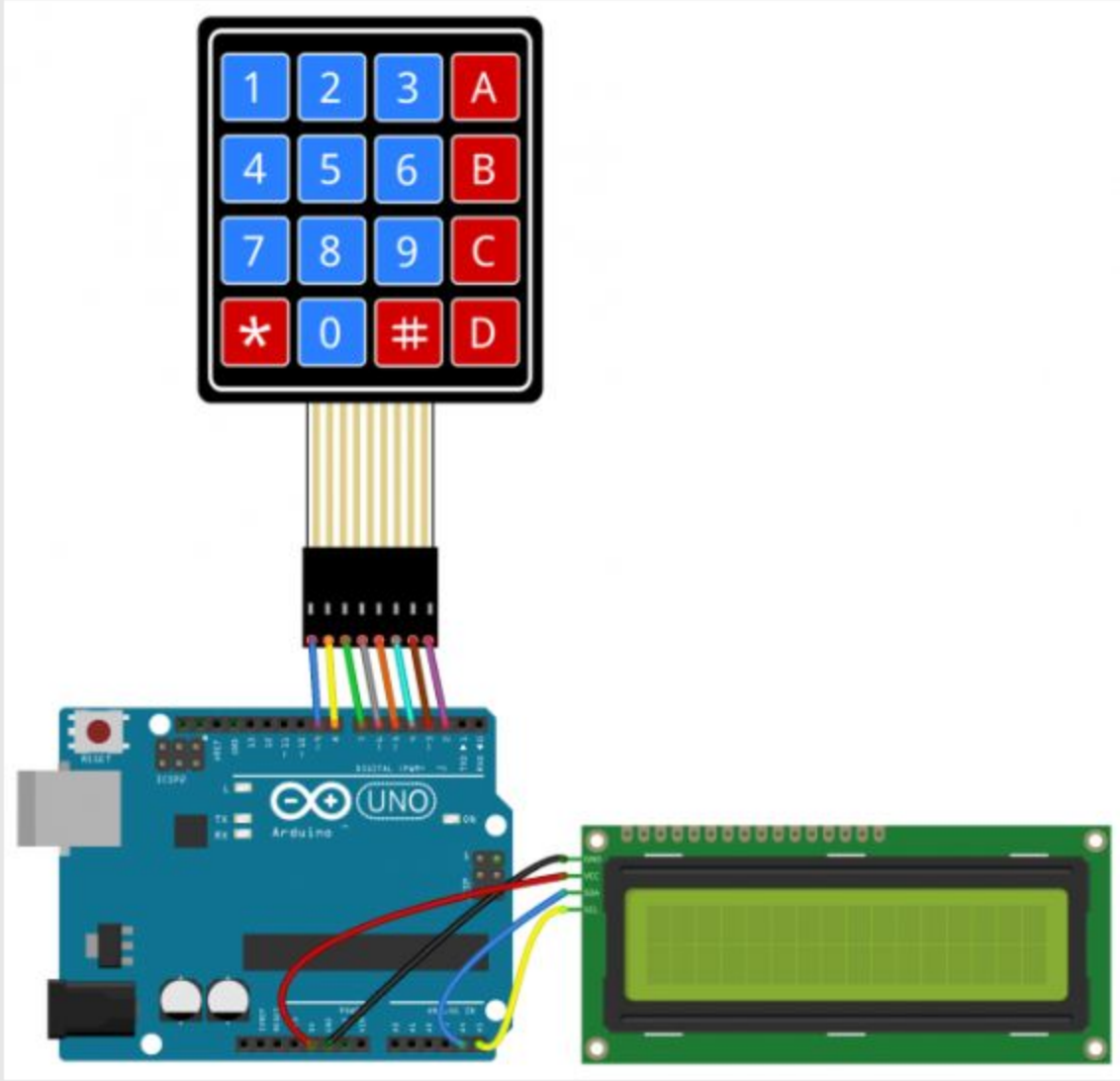
- Ultra-thin design
- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller
- Example programs provided for the BASIC Stamp 2 and Propeller P8X32A microcontrollers

SPECIFICATIONS

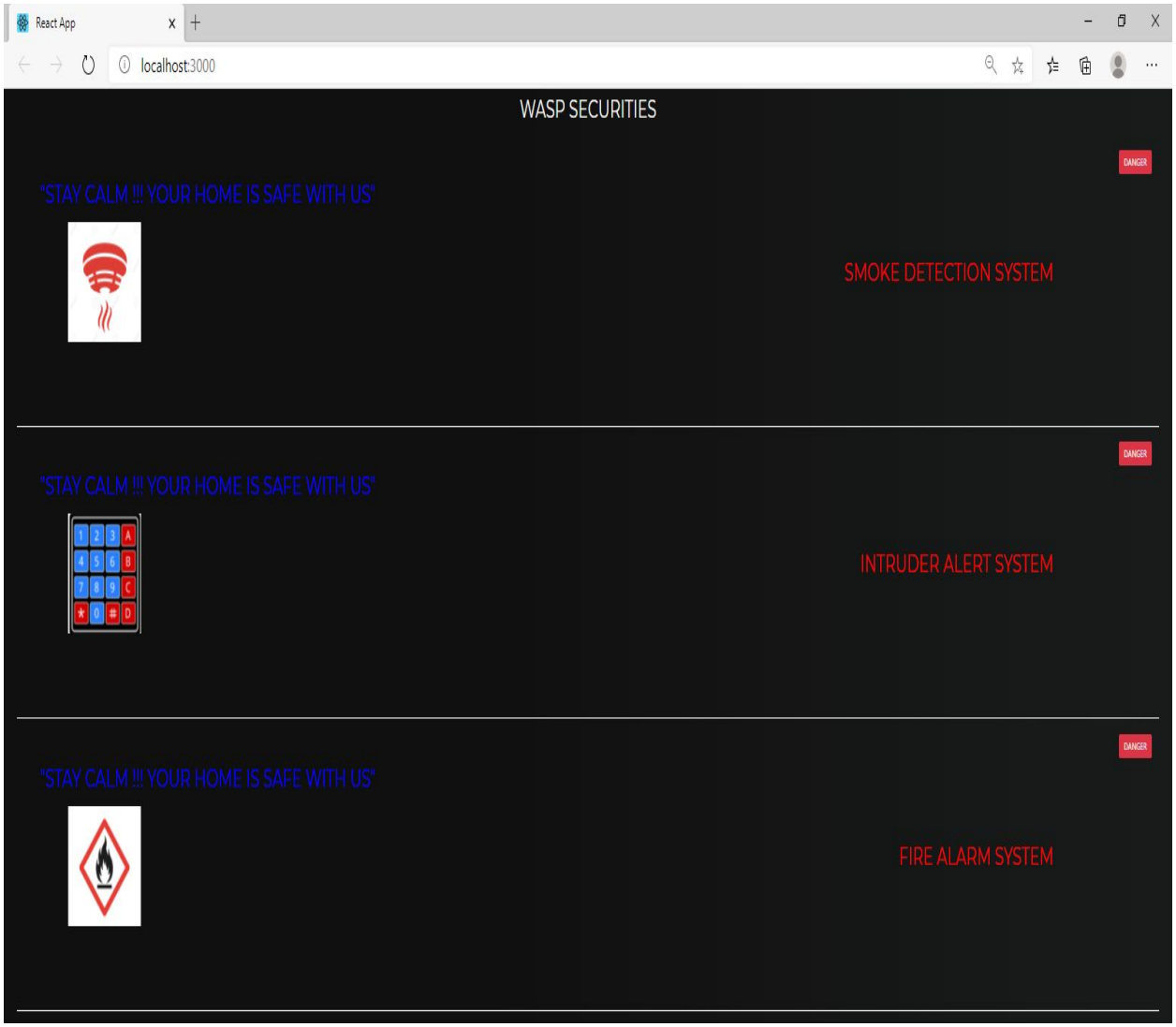
- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4×4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions:Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm) Cable: 0.78 x 3.5 in (2.0 x 8.8 cm)

Connection

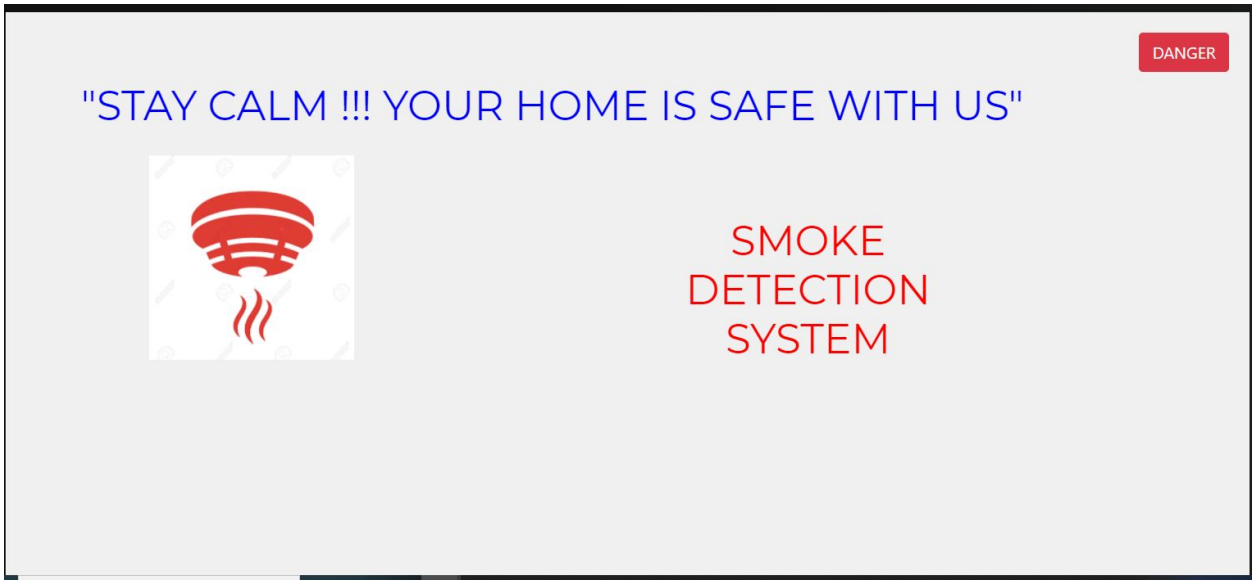
Build the circuit as below:

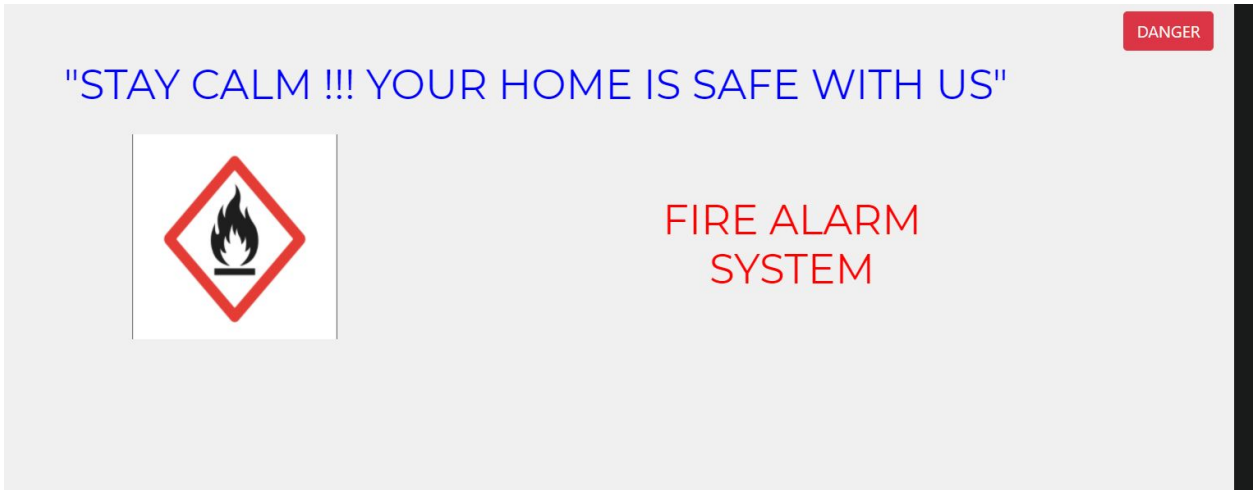


9. Details of a Website

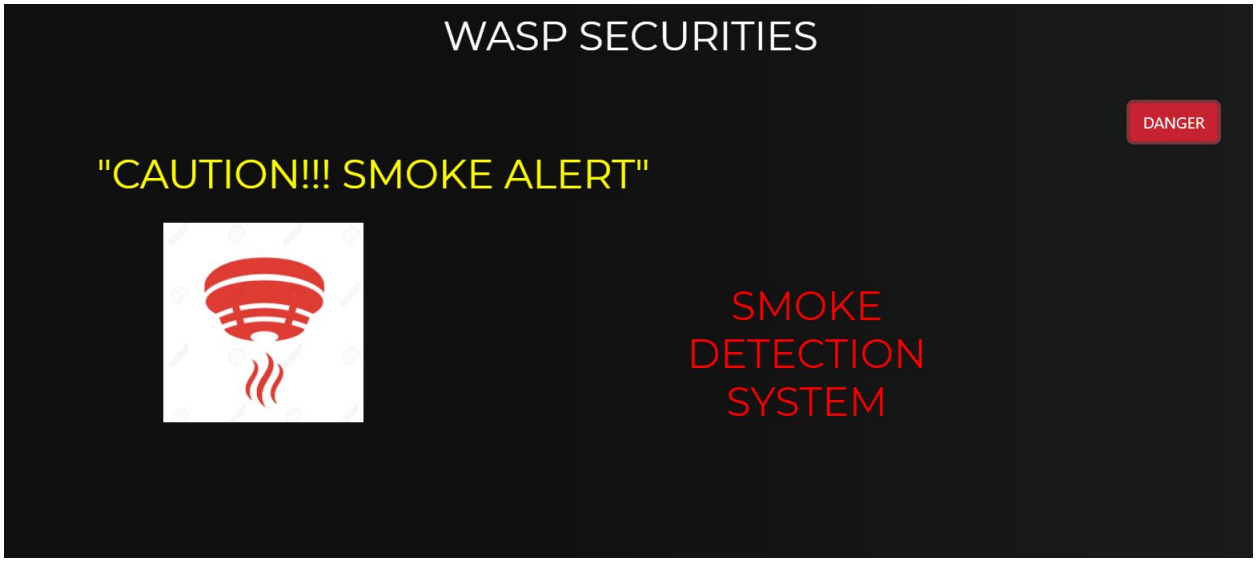


The default webpage would have the functionality for allowing users to enter the password of the lock of the door, if the user will enter the correct message then it will display “Password correct”, if the user enters an incorrect message then it would give an error message “Password incorrect”

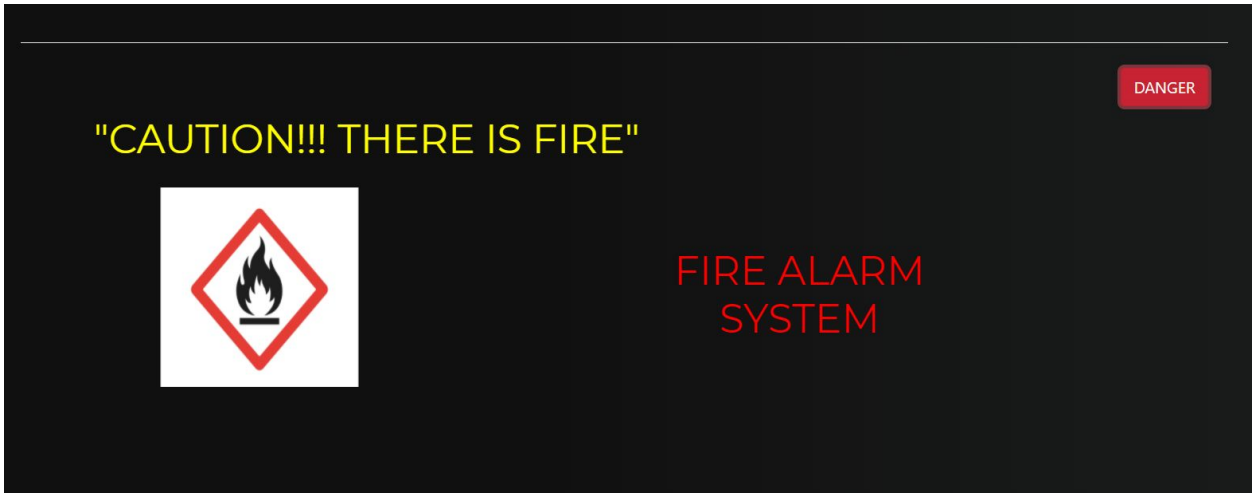




It would give users an alert if we will click on the smoke symbol[As here the proportion of smoke is high], it would give users an alert message “SMOKE ALERT ”,

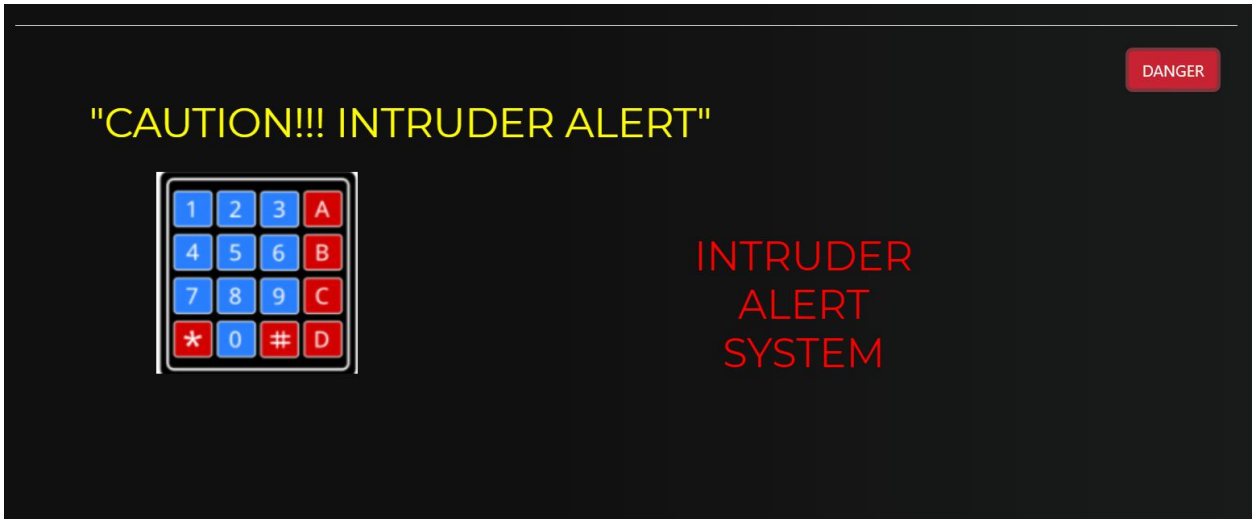


if we will click on the fire symbol [As here there are chances of risk of fire], so it would give users an alert message that “Caution !! There is fire”.



Intruder alert system

If a user enters a wrong password then an intruder alert message is shown.



CODE

Views.py

```
from django.shortcuts import render
from django.http import JsonResponse

from rest_framework.decorators import api_view
from rest_framework.response import Response
from rest_framework import viewsets
```

```
from .serializers import  ImageCustomSerializer
from .models import  Tag, Image



@api_view(['GET'])
def imageList(request):
    images = Image.objects.all().order_by('image_id')
    serializer = ImageCustomSerializer(images, context={"request": request},
many=True)
    return Response(serializer.data)
```

Models.py

```
from django.db import models

# Create your models here.


class Tag(models.Model):
    tag_name = models.CharField(max_length=200)

    def __str__(self):
        return str(self.tag_name)


class Image(models.Model):
    image_id = models.AutoField(primary_key=True)
    description = models.TextField()

    img = models.ImageField(upload_to="images/")
```

```
    x = models.IntegerField()
    y = models.IntegerField()
    height = models.IntegerField()
    width = models.IntegerField()

    tags = models.ManyToManyField(Tag)

    def __str__(self):
        return str(self.image_id)
```

Urls.py

```
"""facetag URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include
from django.views.generic import TemplateView
from django.conf import settings
from django.conf.urls.static import static
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('api.urls')),
    path('', TemplateView.as_view(template_name='index.html')),
]

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

App.js

```
import React from 'react';
import './App.css';


class App extends React.Component {

  constructor(props) {
    super(props);
    this.state = {

      images: [],
      tag_names: {},
      show_tag_detail_of: 1,
      text:
      <header style={{color: 'blue' }}>
      <h1>"STAY CALM !!! YOUR HOME IS SAFE WITH US"</h1>
      </header>

    }

    this.fetchImages = this.fetchImages.bind(this)          // get image list
    this.deleteTag = this.deleteTag.bind(this)              // delete tag with tag id
```

```
    this.addTag = this.addTag.bind(this)                    // add tag name with image id
    this.handleTagChange = this.handleTagChange.bind(this)     // tag_names
    this.changeTagDetailId = this.changeTagDetailId.bind(this) // show_tag_detail_of
   // this.handleClick = this.handleClick.bind(this);
};

state = {
    text:
    <header style={{color: 'Blue' }}>
    <h1>"STAY CALM !!! YOUR HOME IS SAFE WITH US"</h1>
    </header>
  }

  onClickButton1 = () => {
    this.setState({
        text:
       <header style={{color: 'Red'}}>
           <h1>"THERE IS FIRE MAKE SURE YOU ARE OKAY" </h1>
      </header>
    });
  }


componentWillMount() {
    this.fetchImages()
}

fetchImages() {
    console.log('Fetching...')

    fetch('http://127.0.0.1:8000/api/image-list/')
        .then(response => response.json())
        .then(data =>
```

```
            this.setState({
                images: data

            })
        )
}

getCookie(name) {
    var cookieValue = null;
    if (document.cookie && document.cookie !== '') {
        var cookies = document.cookie.split(';');
        for (var i = 0; i < cookies.length; i++) {
            var cookie = cookies[i].trim();
            // Does this cookie string begin with the name we want?
            if (cookie.substring(0, name.length + 1) === (name + '=')) {
                cookieValue = decodeURIComponent(cookie.substring(name.length + 1));
                break;
            }
        }
    }
    return cookieValue;
}

deleteTag(tag) {
    var csrftoken = this.getCookie('csrftoken')

    fetch(`http://127.0.0.1:8000/api/tag-delete/${tag.id}/`, {
        method: 'DELETE',
        headers: {
            'Content-type': 'application/json',
            'X-CSRFToken': csrftoken,
        },
    }).then((response) => {
```

```
        this.fetchImages()
    })
}

handleTagChange(e, image_obj) {
    e.preventDefault()

    var image_id = image_obj.image_id;
    var new_tag_name = e.target.value;

    this.state.tag_names[image_id] = new_tag_name;
}

changeTagDetailId(e, image_obj) {
    console.log("Clicked");
    var image_id = image_obj.image_id;
    this.state.show_tag_detail_of = image_id;
    this.fetchImages()
}




addTag(e, image_obj) {
    e.preventDefault();

    var image_id = image_obj.image_id;
    var csrftoken = this.getCookie('csrftoken')
    var url = `http://127.0.0.1:8000/api/image-tag-add/${image_id}/`;

    var data = {
        tag_name : this.state.tag_names[image_id]
```

```
        }

        fetch(url, {
            method: 'POST',
            headers: {
                'Content-type': 'application/json',
                'X-CSRFToken': csrftoken,
            },
            body: JSON.stringify(data)
        }).then((response) => {
            this.fetchImages()
            var tag_inputs = document.getElementsByClassName("tag_input_elements");
            for (var i=0; i<tag_inputs.length; i++) {
                tag_inputs[i].value = "";
            }
        }).catch(function(error) {
            console.log('ERROR:', error)
        })

    }




// show_tag_detail_of - it will decide which image-container will have tag-container


// mai-div
//     image-container
//         image-div
//             img
//         tag-container
//             tag-add-div
```

```
//                form
//                   form-div
//                      input-div
//                      submit-button-div
//            tag-list-div
//               div
//                   tag-name-div
//                   tag-delete-div
//    image-container
//       image-div
//          img

//    image-container
//       image-div
//          img

onClickButton1 = () => {
   this.setState({
      text:
      <header style={{color: 'yellow'}}>
         <h1>"CAUTION!!! INTRUDER ALERT" </h1>
      </header>
   });
 }
   render() {
      var images = this.state.images;

      var self = this;

      return (


         <div className="container">
```

```
          <header style={{color: 'white' ,textAlign: 'center' , padding: '10px'}}>
      <h1>WASP SECURITIES </h1>
          </header>
            {
            images.map(function(image_obj, index){
               return(
                    <div key={index} className="task-wrapper flex-wrapper" onClick={(e) =>
self.changeTagDetailId(e, image_obj)} >

                        <div className="ImageDiv" style={{flex: 6}}>
                       <img src={image_obj.img}  style={{width: '200px' , height: '200px'  ,
margin: '120px'   }} />

                    </div>
                        <header style={{color: 'Red' ,textAlign: 'center' ,margin: '180px'}}>
                          <h1>{image_obj.description}</h1>

                        </header>


                    <div style={{}}>

                       <button className="btn btn-danger"  onClick={self.onClickButton1
}>DANGER</button>

                       <header style={{ margin: '10px' ,  overflow: 'hidden',whiteSpace: 'nowrap'
,textOverflow: 'ellipsis' , left: '90px',position: 'absolute'}}>
                       <h1>{self.state.text }</h1>
                       </header>

                      </div>
```

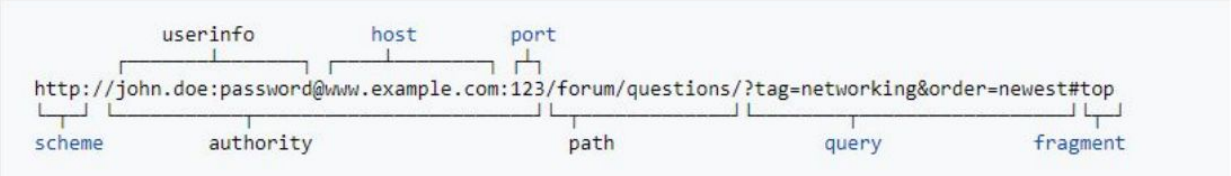
```
                </div>
              )
            })}

        </div>
      )
    }
}

export default App;
```

10.Details of Communication Protocols::

HTTP Protocol

HTTP means HyperText Transfer Protocol. HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page. The other main standard that controls how the World Wide Web works are HTML, which covers how Web pages are formatted and displayed. HTTP is a stateless protocol because each command is executed independently, without any knowledge of the commands that came before it. This is the main reason that it is difficult to implement Web sites that react intelligently to user input. This shortcoming of HTTP is being addressed in a number of new technologies, including ActiveX, Java, JavaScript, and cookies.

A similar abbreviation, HTTPS means HyperText Transfer Protocol Secure. Basically, it is the secure version of HTTP. Communications between the browser 23 and website are encrypted by Transport Layer Security (TLS), or its predecessor, Secure Sockets Layer (SSL).



The request message consists of the following: ● A request line (e.g., GET /images/logo.png HTTP/1.1, which requests a resource from the server.) ● Request header fields (e.g., Accept-Language: en) ● An empty line ● An optional message body The request line and other header fields must each end with (that is, a carriage return character followed by a line feed character). The empty line must consist of only and no other whitespace. HTTP methods are case sensitive. HTTP contains GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, etc. methods that can be used for communication. The response message consists of the following:
● A status line that includes the status code and reason message (e.g., HTTP/1.1 200 OK, which indicates that the client's request succeeded.)
● Response header fields (e.g., Content-Type: text/HTML)
● An empty line
● An optional message body The status line and other header fields must all end with. The empty line must consist of only and no other whitespace. This strict requirement is relaxed somewhat within message bodies for consistent use of other system linebreaks such as or alone.

To communicate, we can use REST APIs and the Django framework. Django helps in building web application and it handles all of the things which are required to communicate with the server and fetch or display data to the user. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations

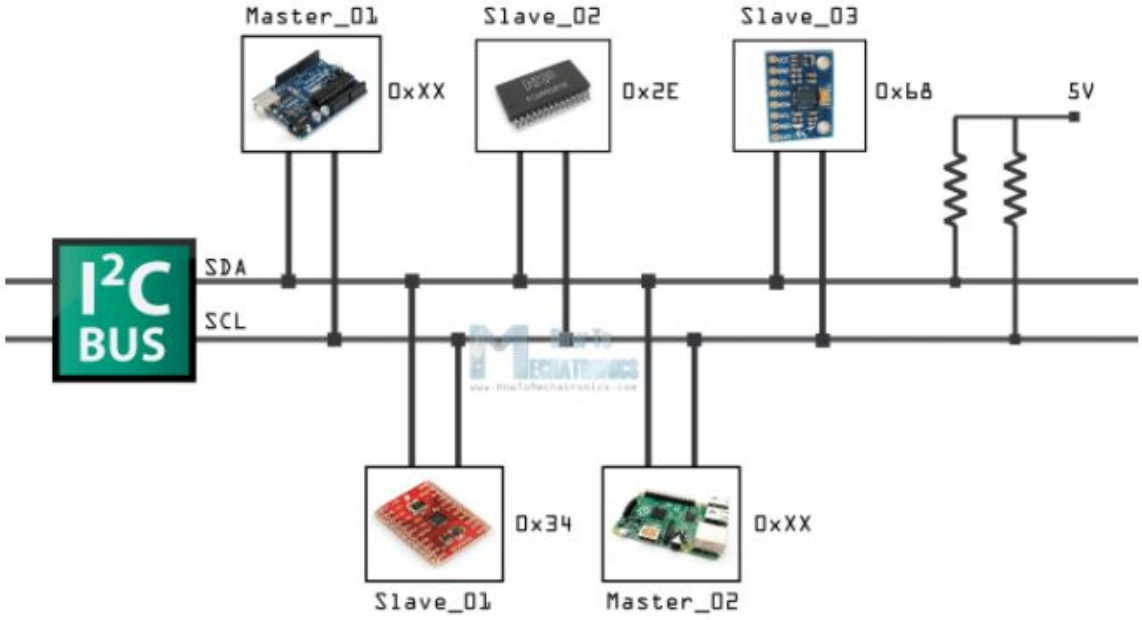
I2C protocol

Overview

The I2C communication bus is very popular and broadly used by many electronic devices because it can be easily implemented in many electronic designs which require communication between a master and multiple slave devices or even multiple master devices

How does It work?

How is it possible, communicate between so many devices with just to wires? Well, each device has a preset ID or a unique device address so the master can choose with which devices will be communicating.



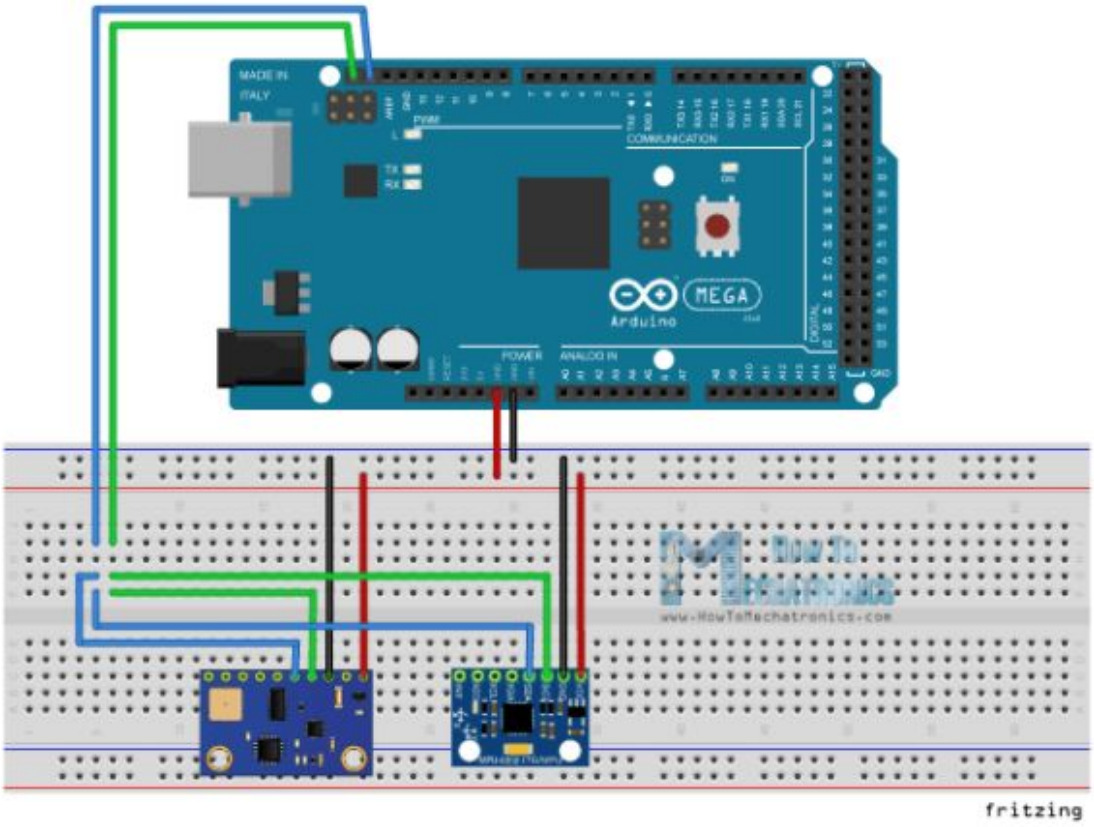
The data signal is transferred in sequences of 8 bits. So after a special start condition occurs comes the first 8 bits sequence which indicates the address of the slave to which the data is being sent. After each 8 bits sequence follows a bit called Acknowledge. After the first Acknowledge bit in most cases comes another addressing sequence but this time for the internal registers of the slave device. Right after the addressing sequences follows the data sequences as many until the data is completely sent and it ends with a special stop condition.

Let’s take an even closer look at these events. The start condition occurs when the data line drops low while the clock line is still high. After this, the clock starts, and each data bit is transferred during each clock pulse.

The device addressing sequence stars with the most significant bit (MSB) first and ends with the least significant bit (LSB) and it’s actually composed of 7 bits because the 8th bit is used for indicating whether the master will write to the slave (logic low) or read from it (logic high).

Next, is the internal registers addressing. The internal registers are locations in the slave's memory containing various information or data. For example, the ADX345 Accelerometer has a unique device address and addition internal registers addresses for the X, Y, and Z-axis. So if we want to read the data of the X-axis, first we need to send the device address and then the particular internal register address for the X-axis

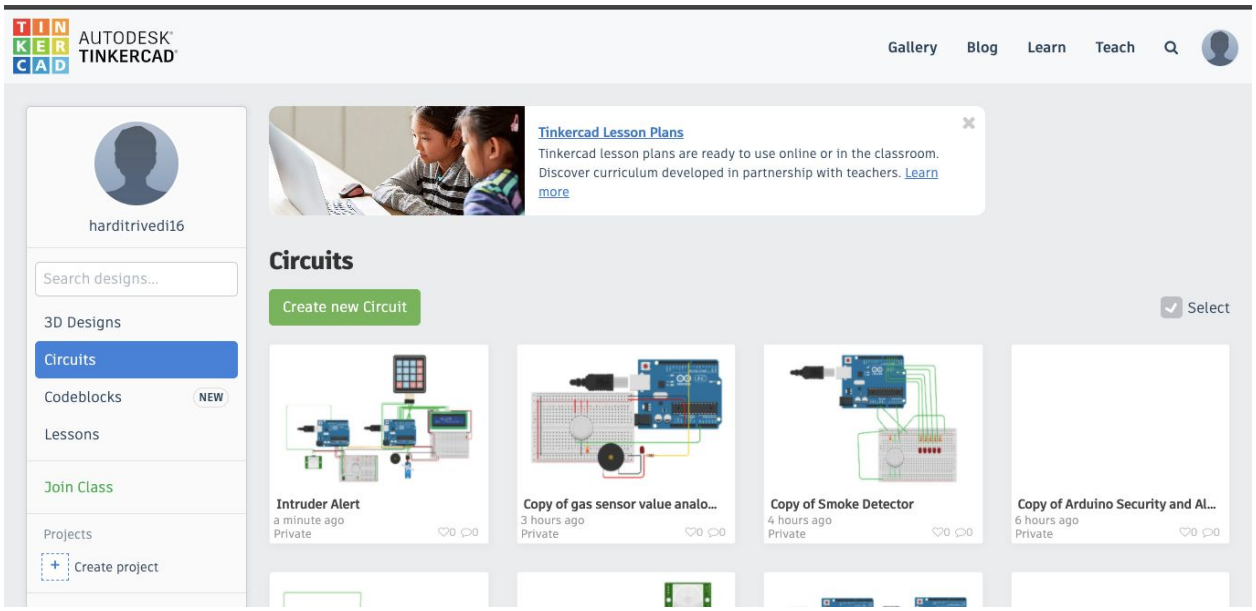
After the addressing, the data transfer sequences begin either from the master or the slave depending on the selected model at the R/W bit. After the data is completely sent, the transfer will end with a stop condition which occurs when the SDA line goes from low to high while the SCL line is high.

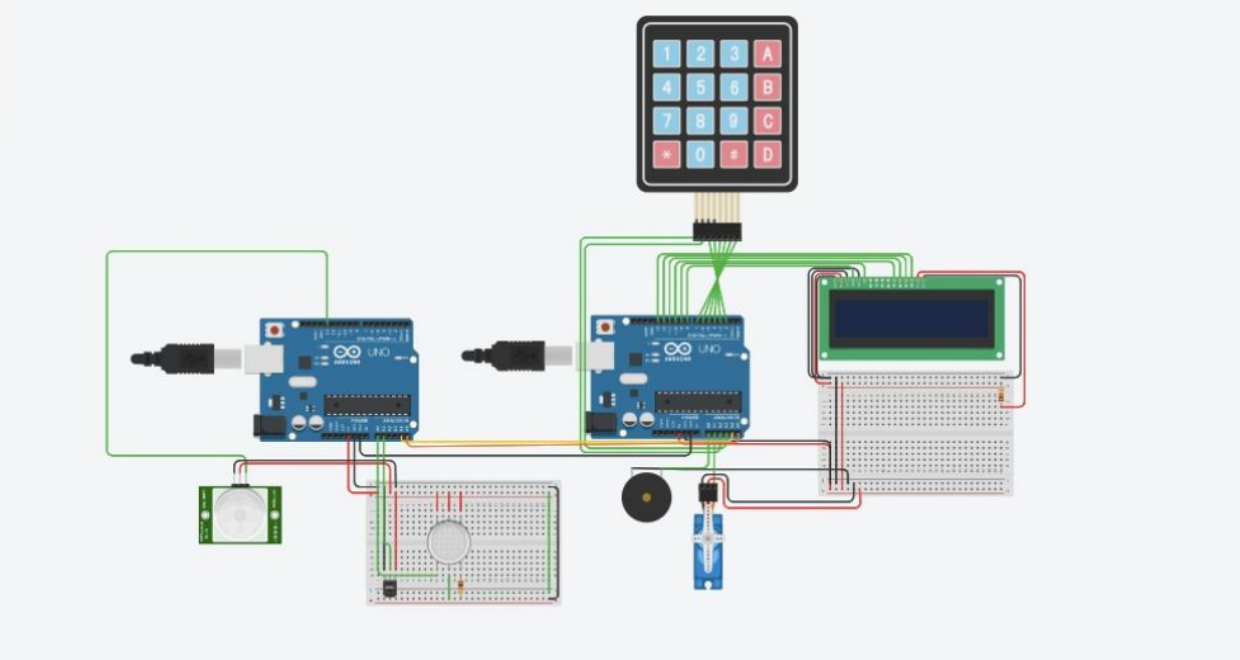


11.Details of Supporting tools

Tinkercad: It is a 3-D free online application that makes it easier to make schematic circuits and simulate them. In this case of a pandemic, where all hardware availability was next to impossible, tinkercad has been very helpful to simulate the project.

Here is a screenshot of the Tinkercad dashboard




1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D

12. Complete program

Code for Master Arduino

```
//IoT Final project
//Prepared by: Mudra Suthar(Au1741022) and Hardi Trivedi (AU1741092)
//Guided by: Prof.Anurag Lakhlani
//Code for Master Arduino


#include <Wire.h>
#include<SoftwareSerial.h>
SoftwareSerial client(2,3); //RX, TX

String webpage="";
int i=0,k=0;
String readString;
int x=0;

boolean No_IP=false;
String IP="";
//char temp0='0';

String name="<p>Wasp Securities</p>";   //22
String dat="<p>Data Received Successfully.....</p>";     //21

int temp_pin = A0;//declaring temperature sensor pin
int tmp_sensorvalue;    //value from the sensor
double temp;

int smoke_pin = A1;//declaring smokesensor pin
int smoke_sensorvalue;    //value from the sensor


int pir_sensorpin = 13;
int pir_sensorvalue = 0;

int temp1,temp2,temp3;
```

```
void check4IP(int t1)
{
  int t2=millis();
  while(t2+t1>millis())
  {
    while(client.available()>0)
    {
      if(client.find("WIFI GOT IP"))
      {
        No_IP=true;
      }
    }
  }
}

void get_ip()
{
  IP="";
  char ch=0;
  while(1)
  {
    client.println("AT+CIFSR");
    while(client.available()>0)
    {
      if(client.find("STAIP,"))
      {
        delay(1000);
        Serial.print("IP Address:");
        while(client.available()>0)
        {
          ch=client.read();
          if(ch=='+')
          break;
          IP+=ch;
        }
      }
```

```
      if(ch=='+')
      break;
    }
    if(ch=='+')
    break;
    delay(1000);
  }
  Serial.print(IP);
  Serial.print("Port:");
  Serial.println(80);
}

void connect_wifi(String cmd, int t)
{
  int temp=0,i=0;
  while(1)
  {
    Serial.println(cmd);
    client.println(cmd);
    while(client.available())
    {
      if(client.find("OK"))
      i=8;
    }
    delay(t);
    if(i>5)
    break;
    i++;
  }
  if(i==8)
  Serial.println("OK");
  else
  Serial.println("Error");
}

void wifi_init()
{
```

```
      connect_wifi("AT",100);
      connect_wifi("AT+CWMODE=3",100);
      connect_wifi("AT+CWQAP",100);
      connect_wifi("AT+RST",5000);
      check4IP(5000);
      if(!No_IP)
      {
        Serial.println("Connecting Wifi....");
        connect_wifi("AT+CWJAP=\"Hogwarts\",\"wasp1603\"",7000);
//provide your WiFi username and password here
     // connect_wifi("AT+CWJAP=\"vpn address\",\"wireless
network\"",7000);
      }
      else
        {
        }
      Serial.println("Wifi Connected");
      get_ip();
      connect_wifi("AT+CIPMUX=1",100);
      connect_wifi("AT+CIPSERVER=1,80",100);
}
void sendwebdata(String webPage)
{
    int ii=0;
     while(1)
     {
      unsigned int l=webPage.length();
      Serial.print("AT+CIPSEND=0,");
      client.print("AT+CIPSEND=0,");
      Serial.println(l+2);
      client.println(l+2);
      delay(100);
      Serial.println(webPage);
      client.println(webPage);
      while(client.available())
      {
        //Serial.print(Serial.read());
```

```
        if(client.find("OK"))
        {
          ii=11;
          break;
        }
      }
      if(ii==11)
      break;
      delay(100);
     }
}


void setup()
{
  Serial.begin(9600);
  Wire.begin(); // join i2c bus (address optional for master)
   client.begin(9600);
   wifi_init();
  pinMode(pir_sensorpin, INPUT);
}



void loop()
{
  tmp_sensorvalue = analogRead(A0);
  smoke_sensorvalue = analogRead(A1);
  pir_sensorvalue = digitalRead(pir_sensorpin);
  temp = calculateTemp(tmp_sensorvalue);
  if(temp >= 50){
    Serial.println("There is Fire!! Make sure you are okay");
    temp1 = 1;
    }
    else{
      Serial.println("Your home is safe and cool just like you;)");
      temp2 = 0;
```

```
      }
      byte flagt_buzzer = temp1;
      double smoke = map (smoke_sensorvalue,300,750,0,100);
      if(smoke >= 40){
    Serial.println("We are sensing something weird! Harmful smoke is
detected");
    temp2 = 1;
    }
    else{
      Serial.println("Be calm! Everything is fine");
      temp2 = 0;
      }
      byte smoke_buzzer = temp2;
      if (pir_sensorvalue == HIGH){
        temp3 = 1;
        }
        else{
          temp3 = 0;
          }
          byte flag_pirsensor = temp3;

  Wire.beginTransmission(4); // transmit to device #4
  Wire.write(flagt_buzzer); // sends one byte
  Wire.write(smoke_buzzer);
  Wire.write(flag_pirsensor);
  Wire.endTransmission();    // stop transmitting


  delay(2000);

  k=0;
  Serial.println("Please Refresh your Page");
  while(k<1000)
  {
    k++;
   while(client.available())
   {
```

```
    if(client.find("0,CONNECT"))
    {
      Serial.println("Start Printing");
      Send();
      Serial.println("Done Printing");
      delay(1000);
    }
  }
  delay(1);
 }
}
int calculateTemp(int x){
  float voltage = x * 5.0;
 voltage /= 1024.0;
 float temperatureC = (voltage - 0.5) * 100 ;
  }

  void Send()
{
      webpage = "<h1>Welcome to MiltonSecurities</h1><body bgcolor=ffffff
color=000080>";
      sendwebdata(webpage);
      webpage=name;
      webpage+=dat;
      sendwebdata(webpage);
      delay(1000);

    webpage = "<h2>Be Relaxed! Your home is safe with
us</h2><color=000080>";
      sendwebdata(webpage);

      webpage = "<h2>FIRE ALARAM SYSTEM</h2><color=000080 >";
      if(flagt_buzzer == 1){
        webpage+="<h3>There is a fire!!Make sure you are
Okay</h3><color=ff0000 ";
        }
        if(flagt_buzzer == 0){
```

```
        webpage+="<h3>Your home is safe and cool just like
you</h3><color=000080 >";
        }
      sendwebdata(webpage);

       webpage = "<h2>SMOKE DETECTION SYSTEM</h2><color=000080 >";
      if(flagt_buzzer == 1){
        webpage+="<h3>We are sensing something weird!!Harmful smoke
detected</h3><color=ff0000 ";
        }
        if(flagt_buzzer == 0){
        webpage+="<h3>Be calm! Everything is fine</h3><color=000080 >";
        }
      sendwebdata(webpage);
     client.println("AT+CIPCLOSE=0");
}
```

Code for Slave Arduino

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <Servo.h>
#include <Wire.h>
#include<SoftwareSerial.h>
SoftwareSerial client(0,1); //RX, TX

String webpage="";
int i=0,k=0;
String readString;
int x=0;

boolean No_IP=false;
String IP="";
char temp0='0';

String name="<p>Milton Securities</p>";   //22
```

```
String dat="<p>Data Received Successfully.....</p>";     //21

// initialize the library with the numbers of the interface pins
LiquidCrystal LCD( 8, 9, 10, 11, 12, 13 ) ;   //RS,EN,D4,D5,D6,D7

Servo servo ;

const int buzzerPin = 14 ;   // pin-14 = pin-A0
const int servoPin = 15 ;   // pin-15 = pin A-1


const byte ROWS = 4, COLS = 4 ;

// Define the Keymap
char keys[ROWS][COLS] = {
   {'1','2','3','A'},
   {'4','5','6','B'},
   {'7','8','9','C'},
   {'*','0','#','D'}
} ;

// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
byte rowPins[ROWS] = { 16, 17, 2, 3 } ;
// Connect keypad COL0, COL1 and COL2 to these Arduino pins.
byte colPins[COLS] = { 4, 5, 6, 7 } ;

//  Create the Keypad
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS ) ;

char password[16], string[16] ;
int flag_h_setpassword = 1, flag_inputpassword = 0, flag_inputstring = 0,
flag_opendoor = 1;
int count = 0, pos = 0,  temp_count=0, count1 = 0 ;

void check4IP(int t1)
{
  int t2=millis();
```

```
  while(t2+t1>millis())
  {
    while(client.available()>0)
    {
      if(client.find("WIFI GOT IP"))
      {
        No_IP=true;
      }
    }
  }
}

void get_ip()
{
  IP="";
  char ch=0;
  while(1)
  {
    client.println("AT+CIFSR");
    while(client.available()>0)
    {
      if(client.find("STAIP,"))
      {
        delay(1000);
        Serial.print("IP Address:");
        while(client.available()>0)
        {
          ch=client.read();
          if(ch=='+')
          break;
          IP+=ch;
        }
      }
      if(ch=='+')
      break;
    }
    if(ch=='+')
```

```
    break;
    delay(1000);
  }
  Serial.print(IP);
  Serial.print("Port:");
  Serial.println(80);
}

void connect_wifi(String cmd, int t)
{
  int temp=0,i=0;
  while(1)
  {
    Serial.println(cmd);
    client.println(cmd);
    while(client.available())
    {
      if(client.find("OK"))
      i=8;
    }
    delay(t);
    if(i>5)
    break;
    i++;
  }
  if(i==8)
  Serial.println("OK");
  else
  Serial.println("Error");
}

void wifi_init()
{
      connect_wifi("AT",100);
      connect_wifi("AT+CWMODE=3",100);
      connect_wifi("AT+CWQAP",100);
      connect_wifi("AT+RST",5000);
```

```
      check4IP(5000);
      if(!No_IP)
      {
        Serial.println("Connecting Wifi....");
        connect_wifi("AT+CWJAP=\"Hogwarts\",\"wasp1603\"",7000);
//provide your WiFi username and password here
     // connect_wifi("AT+CWJAP=\"vpn address\",\"wireless
network\"",7000);
      }
      else
        {
        }
      Serial.println("Wifi Connected");
      get_ip();
      connect_wifi("AT+CIPMUX=1",100);
      connect_wifi("AT+CIPSERVER=1,80",100);
}

void sendwebdata(String webPage)
{
    int ii=0;
     while(1)
     {
      unsigned int l=webPage.length();
      Serial.print("AT+CIPSEND=0,");
      client.print("AT+CIPSEND=0,");
      Serial.println(l+2);
      client.println(l+2);
      delay(100);
      Serial.println(webPage);
      client.println(webPage);
      while(client.available())
      {
        //Serial.print(Serial.read());
        if(client.find("OK"))
        {
          ii=11;
```

```
          break;
        }
      }
      if(ii==11)
      break;
      delay(100);
     }
}


void setup() {
   for(int k=8 ; k<14 ; k++) {
      pinMode(k,OUTPUT) ;   //pins 8-14 are enabled as output
   }

   LCD.begin(16, 2) ;
   pinMode(buzzerPin, OUTPUT) ;
   servo.attach(servoPin) ;
   Wire.begin(4);                // join i2c bus with address #4
   Wire.onReceive(receiveEvent); // register event
   Serial.begin(9600) ;
    client.begin(9600);
   wifi_init();
   LCD.setCursor(0,0) ;
   LCD.print("   WELCOME !!") ;
   LCD.setCursor(0,1) ;
   LCD.print("Set a Password :") ;

   InitializePassword(), InitializeString() ;
   CloseDoor() ;


}



void loop() {
```

```
//First checking the door for the breakin

   // Now, the Welcome message...


   char key = kpd.getKey() ;   //storing pressed key value in a char

   if( key != NO_KEY ) {

      if( flag_h_setpassword == 1 ) {
         H_SetPassword() ;

      }
      if( key == '*' ) {
         if( flag_inputpassword == 1 ) {
            InitializePassword(), H_SetPassword() ;
         }
         else if( flag_inputstring = 1 ) {
            InitializeString(), H_EnterPassword() ;
         }
      }
      else if( key == '#' ) {
         if( flag_inputpassword == 1  &&  count > 0 ) {
            flag_inputpassword = 0 ;
            password[count] = '\0' ;

            H_EnterPassword() ;
         }
         else if( flag_inputstring == 1  &&  count > 0 ) {
            flag_inputstring = 0 ;
            string[count] = '\0' ;
            //Comparing the password and if correct opening the door
            if( Compare_Password_and_String() == 1 ) {
               LCD.clear() ;
```

```
                LCD.print("***VERIFIED!!***") ;
                Serial.println("Sleep tight! your home is safe from
intruders") ;
                count1 = count1 + 1;

                OpenDoor() ;
              }
              else {
               //In case when wrong password is entered, the the door
wont open and alert message will be shown
                LCD.clear() ;
                LCD.print("Wrong Password !") ;
                Serial.println("Someone unsuccessfully attempted to open
the lock !") ;

                tone(buzzerPin, 100, 250) ;
                //delay(100) ;

                H_EnterPassword() ;
                temp_count=temp_count +1;
                //In case wrong password us entered three consecutive
times theb the whole system will freeze
                if (temp_count>=3){
                   LCD.setCursor(0,0) ;
                   LCD.print("INTRUDER ALERT") ;
                   LCD.setCursor(0,1) ;
                    LCD.print("SYSTEM IS LOCKED") ;
             Serial.println("Intruder Alert!! Call Police!!") ;

                   tone(buzzerPin, 450, 150) ;
                   //delay(400) ;
                   tone(buzzerPin, 150, 250) ;
                   exit(0);
                  }

              }
            }
```

```
         }
        else if( flag_inputpassword == 1  ||  flag_inputstring == 1 ) {
            LCD.print(key) ;
            delay(100) ;
            LCD.setCursor(count,1) ;
            LCD.print('*') ;
            if( flag_inputpassword == 1 )   password[count] = key ;
            else if( flag_inputstring == 1 )   string[count] = key ;
            ++count ;
         }
      }
 k=0;
  Serial.println("Please Refresh your Page");
  while(k<1000)
  {
    k++;
   while(client.available())
   {
    if(client.find("0,CONNECT"))
    {
      Serial.println("Start Printing");
      Send();
      Serial.println("Done Printing");
      delay(1000);
    }
  }
  delay(1);
 }



}


//User Defined functions used in the program
void InitializePassword() {
   for( int i=0 ; i<16 ; ++i )
```

```
      password[i] = 0 ;
}

void InitializeString() {
   for( int i=0 ; i<16 ; ++i )
      string[i] = 1 ;
}

void H_SetPassword() {
   LCD.clear() ;
   LCD.setCursor(0,0) ;
   LCD.print("Set a Password :") ;
   LCD.setCursor(0,1) ;
   flag_h_setpassword = 0 ;
   flag_inputpassword = 1, count = 0 ;
}

void H_EnterPassword() {
   CloseDoor() ;
   LCD.clear() ;
   LCD.setCursor(0,0) ;
   LCD.print("Enter Password :") ;
   LCD.setCursor(0,1) ;
   flag_inputstring = 1, count = 0 ;
}

int Compare_Password_and_String() {
   int i ;
   for( i=0  ;  password[i]!='\0' && string[i]!='\0'  ;  ++i ) {
      if( password[i] != string[i] )
         return 0 ;
   }
   if( password[i] == '\0'  &&  string[i] == '\0' )
      return 1 ;
   else   return 0 ;
}
```

```
void OpenDoor() {
   if( flag_opendoor == 1 )
      return;
   for( pos=15 ; pos<=100 ; ++pos ) {
      servo.write(pos) ;
      delay(15) ;
   }
   flag_opendoor = 1 ;
}

void CloseDoor()  {
   if( flag_opendoor == 0 )
      return;
   for( pos=100 ; pos>=15; --pos ) {
      servo.write(pos) ;
      delay(15) ;
   }
   flag_opendoor = 0 ;
}



    //flag_opendoor = flag_opendoor + 0;
    //count1 = count1 + 0;

    void receiveEvent(int howMany)
{

  int flagt_buzzer = Wire.read();
  //delay(10);
  int smoke_buzzer = Wire.read();
  //delay(10);
  int flag_pirsensor= Wire.read();
  //delay(10);

  if(flagt_buzzer==1){
   tone(buzzerPin, 450, 150) ;
```

```
   //delay(400) ;
   tone(buzzerPin, 150, 250) ;
    }
    if(smoke_buzzer==1){
   tone(buzzerPin, 450, 150) ;
   //delay(400) ;
   tone(buzzerPin, 150, 250) ;
    }
    if(flag_pirsensor ==1 && flag_opendoor == 1 && count1==0){
      LCD.setCursor(0,0) ;
       LCD.print("INTRUDER ALERT") ;
       tone(buzzerPin, 450, 150) ;
      // delay(400) ;
       tone(buzzerPin, 150, 250) ;
       exit(0);
      }
}
    void Send()
{
      webpage = "<h1>Welcome to MiltonSecurities</h1><body bgcolor=ffffff
color=000080>";
      sendwebdata(webpage);
      webpage=name;
      webpage+=dat;
      sendwebdata(webpage);
      delay(1000);

    webpage = "<h2>Be Relaxed! Your home is safe with
us</h2><color=000080>";
      sendwebdata(webpage);

      webpage = "<h2>INTRUDER ALERT SYSTEM</h2><color=000080 >";
      if(temp_count >=3){
        webpage+="<h3>Sleep tight your home is safe from
Intruders</h3><color=ff0000 ";
        }
        else{
```

```
        webpage+="<h3>Intruder Alert!! Call Police</h3><color=000080 >";
        }
      sendwebdata(webpage);

      =
     client.println("AT+CIPCLOSE=0");
}
```

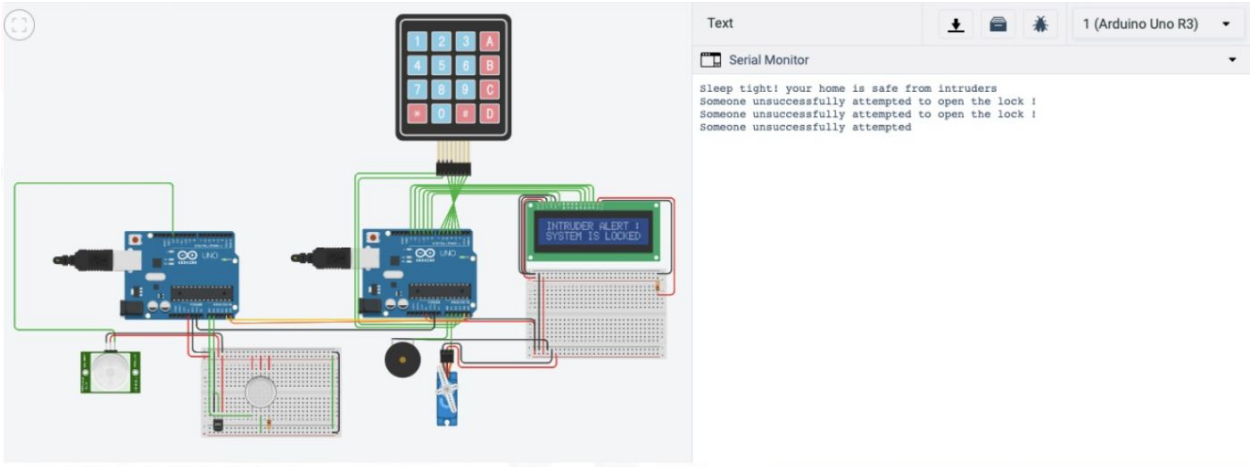
13 Summary

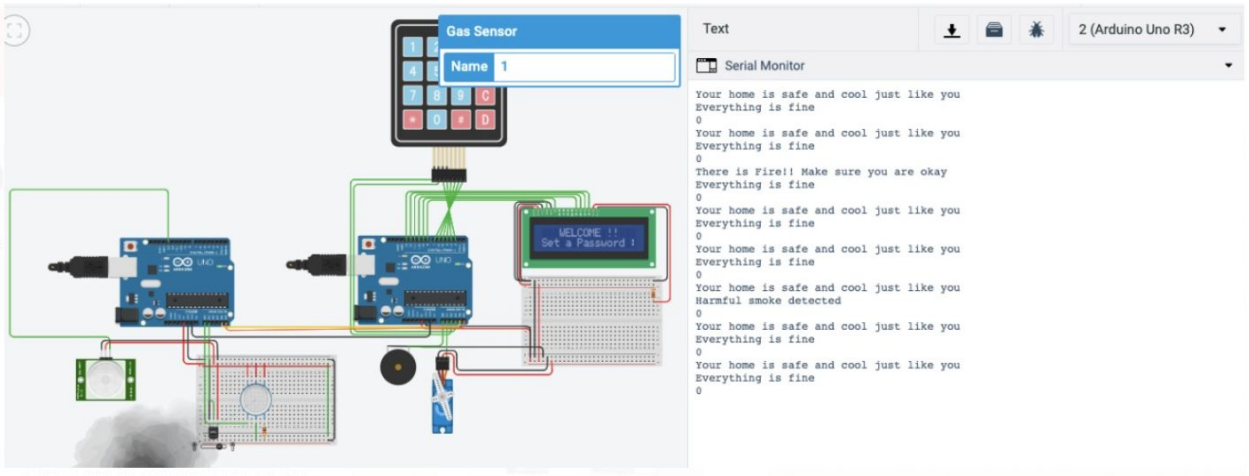
We have designed a project which works as an anti-theft system and provides a sense of security and safety to the common people. We have built this system by using the concepts of IoT. It will

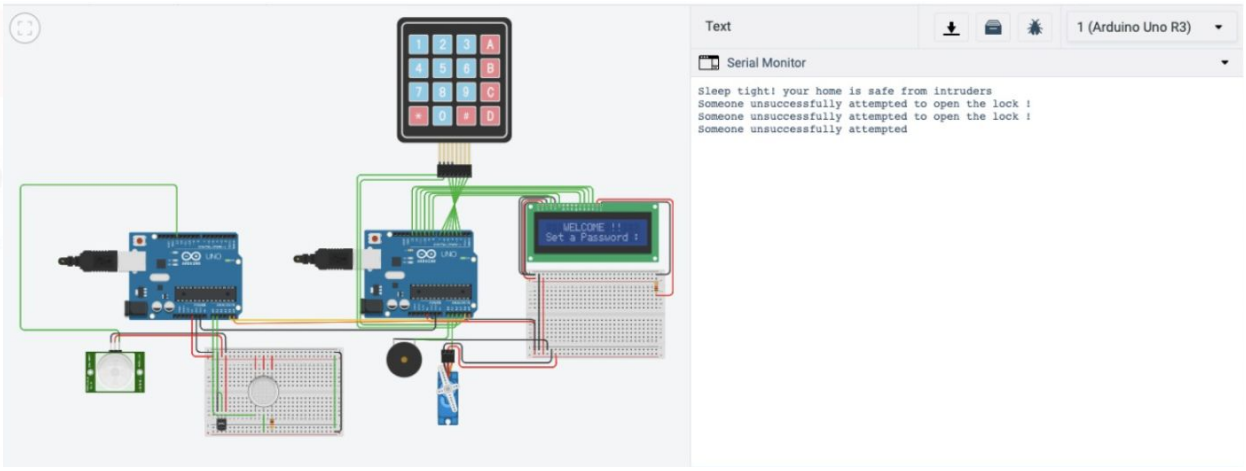
provide security by providing a smoke alert system, fire alert system, automatic lock alert system.
This system cannot only be used in our homes but an office or corporate buildings, banks extra that will act as an anti-theft system and provide security to the valuables and belongings.

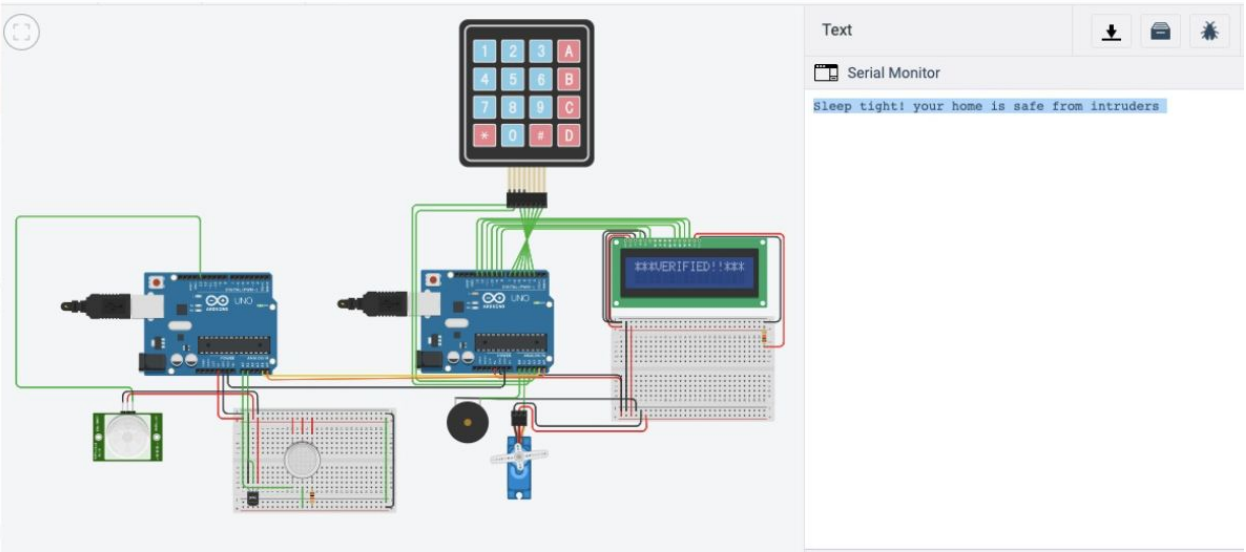
Our learning Goals from this project are concepts of the Internet Of Things. We have learned how to form a physical network of things or objects—devices, other items—embedded with electronics, software, sensors, and network connectivity that enables these things or objects to collect and exchange data along with learning various new components such as Tmp36 Sensor, Smoke Sensor MQ2, Pir Motion Sensor 2) Keypad 3)Servo Motor 4)Django Framework(for website) 5) React JS (for website)

Whenever the thief enters the house/bank or any corporate office, and steps on the floor immediately it is sensed by the sensor which passes on the signal to the Arduino controller. It will provide a smoke alert, fire alert system along with an automatic lock system.
This system is suitable for small personal area surveillance. i.e. personal office cabin, bank locker room, parking entrance. Whenever the motion is detected through




Gas Sensor
Name 1
Text
2 (Arduino Uno R3)
Serial Monitor
Your home is safe and cool just like you
Everything is fine
0
Your home is safe and cool just like you
Everything is fine
0
There is Fire!! Make sure you are okay
Everything is fine
0
Your home is safe and cool just like you
Everything is fine
0
Your home is safe and cool just like you
Everything is fine
0
Your home is safe and cool just like you
Harmful smoke detected
0
Your home is safe and cool just like you
Everything is fine
0
Your home is safe and cool just like you
Everything is fine
0
WELCOME !!
Set a Password :

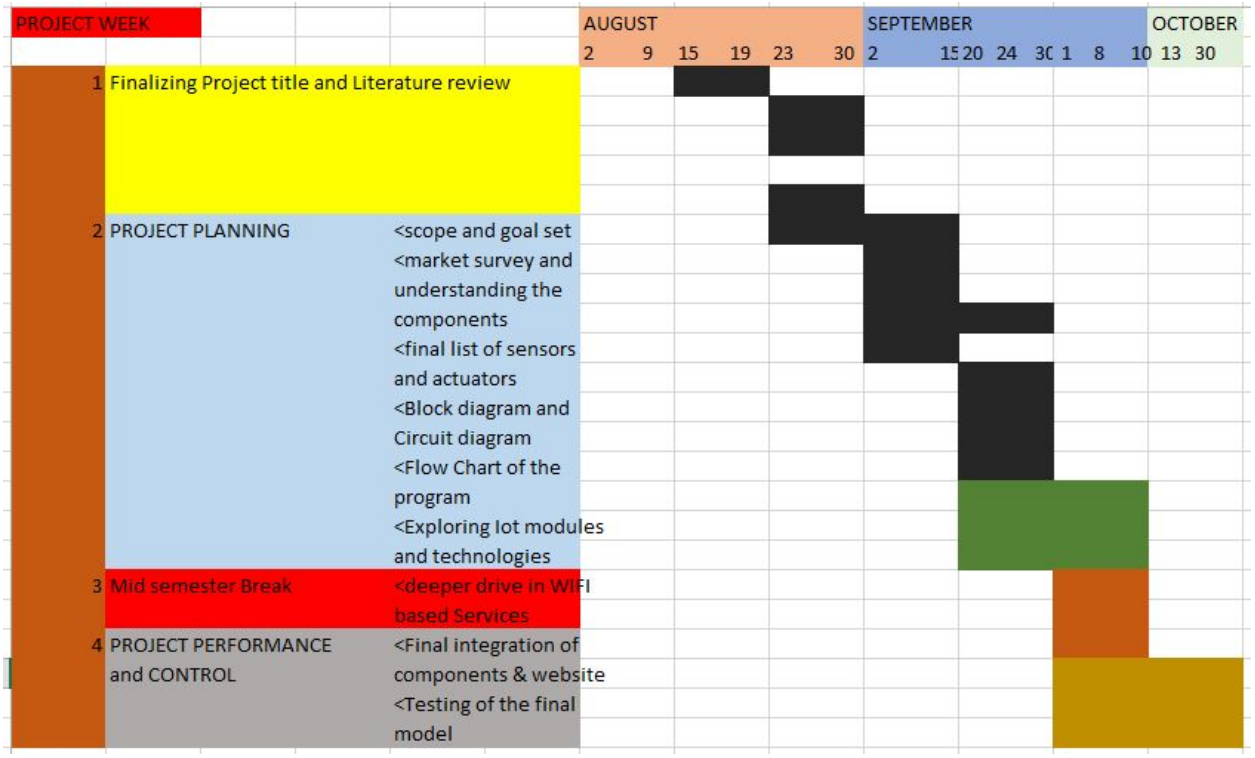

Text
1 (Arduino Uno R3)
Serial Monitor
Sleep tight! your home is safe from intruders
Someone unsuccessfully attempted to open the lock !
Someone unsuccessfully attempted to open the lock !
Someone unsuccessfully attempted
WELCOME !!
Set a Password :


Text
Serial Monitor
Sleep tight! your home is safe from intruders
VERIFIED!!

VIDEO LINKS::

The final project video links of both tinkercad and Website are attached below::
1)TINKERCAD
2)WEBSITE

TIMELINE OF THE PROJECT



Future Work:

This project can be expanded and many other functions can be added. For example, instead of having a keypad, a thumb recognition sensor, or retina scanner, or face recognition system can be added. A password can be misplaced but you can never change ur retina or face nor can anyone have the same retina or face as you. This way more protection can be obtained.

Also, an algorithm can be modified in such a way that, if a fire alarm goes on then it automatically calls the fire brigade or when an intruder alarm goes on then it can automatically call the police.

Apart from that, instead of PIR sensors, we can use a camera, so the owner can have a more proper view of what's going on in the house.

14. References

1.(2017). Anti theft detection system. Retrieved 2017, from http://www.internationaljournalssrg.org/uploads/specialissuepdf/ICRTETM-2019/2019/CSE/IJCSE-ICRTETM-P113.pdf

2 .(2020). Tincercad.com. Retrieved October 10, 2020, from http://www.tincercad.com/

3.(2020). Diagrams.net - free flowchart maker and diagrams online. Retrieved October 10, 2020, from https://app.diagrams.net/

4. Home. Retrieved October 10, 2020, from https://www.arduino.cc/

5. Retrieved - from http://www.jcreview.com/fulltext/197-1590153080.pdf/

6.ESP8266 Arduino tutorial – WiFi module complete review. (n.d.). Retrieved October 31, 2020, from https://www.geekstips.com/esp8266-arduino-tutorial-iot-code-example/

7. *How to Send Data from Arduino to Webpage using WiFi*. (2019, January 7). Circuit Digest. https://circuitdigest.com/microcontroller-projects/sending-arduino-data-to-webpag

Appendix A

1 . TMP36 sensor

SPECIFICATIONS

V_S = 2.7 V to 5.5 V, −40°C ≤ T_A ≤ +125°C, unless otherwise noted.

Table 1.

Parameter[1]	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
ACCURACY						
TMP35/TMP36/TMP37 (F Grade)		T_A = 25°C		±1	±2	°C
TMP35/TMP36/TMP37 (G Grade)		T_A = 25°C		±1	±3	°C
TMP35/TMP36/TMP37 (F Grade)		Over rated temperature		±2	±3	°C
TMP35/TMP36/TMP37 (G Grade)		Over rated temperature		±2	±4	°C
Scale Factor, TMP35		10°C ≤ T_A ≤ 125°C		10		mV/°C
Scale Factor, TMP36		−40°C ≤ T_A ≤ +125°C		10		mV/°C
Scale Factor, TMP37		5°C ≤ T_A ≤ 85°C		20		mV/°C
		5°C ≤ T_A ≤ 100°C		20		mV/°C
		3.0 V ≤ V_S ≤ 5.5 V				
Load Regulation		0 µA ≤ I_L ≤ 50 µA				
		−40°C ≤ T_A ≤ +105°C		6	20	m°C/µA
		−105°C ≤ T_A ≤ +125°C		25	60	m°C/µA
Power Supply Rejection Ratio	PSRR	T_A = 25°C		30	100	m°C/V
		3.0 V ≤ V_S ≤ 5.5 V		50		m°C/V
Linearity				0.5		°C
Long-Term Stability		T_A = 150°C for 1000 hours		0.4		°C
SHUTDOWN						
Logic High Input Voltage	V_{IH}	V_S = 2.7 V	1.8			V
Logic Low Input Voltage	V_{IL}	V_S = 5.5 V			400	mV
OUTPUT						
TMP35 Output Voltage		T_A = 25°C		250		mV
TMP36 Output Voltage		T_A = 25°C		750		mV
TMP37 Output Voltage		T_A = 25°C		500		mV
Output Voltage Range			100		2000	mV
Output Load Current	I_L		0		50	µA
Short-Circuit Current	I_{SC}	Note 2			250	µA
Capacitive Load Driving	C_L	No oscillations[2]	1000	10000		pF
Device Turn-On Time		Output within ±1°C, 100 kΩ‖100 pF load[2]		0.5	1	ms
POWER SUPPLY						
Supply Range	V_S		2.7		5.5	V
Supply Current	I_{SY} (ON)	Unloaded			50	µA
Supply Current (Shutdown)	I_{SY} (OFF)	Unloaded		0.01	0.5	µA

[1] Does not consider errors caused by self-heating.
[2] Guaranteed but not tested.

2 . PIR SENSOR

Pin description

Pin Number	Symbol	Description
1	**A**	Retriggerable & non-retriggerable mode select (A=1 : re-triggerable)
2	**VO**	Detector output pin (active high)
3	**RR1**	Output pulse width control (Tx) * See definition below
4	**RC1**	Output pulse width control (Tx) *
5	**RC2**	Trigger inhibit control (Ti) *
6	**RR2**	Trigger inhibit control (Ti) *
7	**Vss**	Ground
8	**VRF**	RESET & voltage reference input (Normally high. Low=reset)
9	**VC**	Trigger disable input (VC >0.2Vdd=enable; Vc<0.2Vdd =disabled)
10	**IB**	Op-amp input bias current setting
11	**Vdd**	Supply voltage
12	**2OUT**	2nd stage Op-amp output
13	**2IN-**	2nd stage Op-amp inverting input
14	**1IN+**	1 st stage Op-amp non-inverting input
15	**1IN-**	1 st stage Op-amp inverting input
16	**1OUT**	1 st stage Op-amp output

*

Tx = The time duration during which the output pin (Vo) remains high after triggering.

Ti = During this time period, triggering is inhibited. See timing charts for details.

3. Smoke Sensor

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, i-butane, propane, methane ,alcohol, Hydrogen, smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
V_H	Heating voltage	5V±0.1	ACOR DC
R_L	Load resistance	can adjust	
R_H	Heater resistance	33Ω ±5%	Room Tem
P_H	Heating consumption	less than 800mw	

B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-20℃-50℃	
Tas	Storage Tem	-20℃-70℃	
R_H	Related humidity	less than 95%Rh	
O_2	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	3KΩ-30KΩ (1000ppm iso-butane)	Detecting concentration scope: 200ppm-5000ppm LPG and propane 300ppm-5000ppm butane 5000ppm-20000ppm methane 300ppm-5000ppm H_2 100ppm-2000ppm Alcohol
α (3000/1000) isobutane	Concentration Slope rate	≤0.6	
Standard Detecting Condition	Temp: 20℃±2℃ Humidity: 65%±5%	Vc:5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

	Parts	Materials
1	Gas sensing layer	SnO_2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al_2O_3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

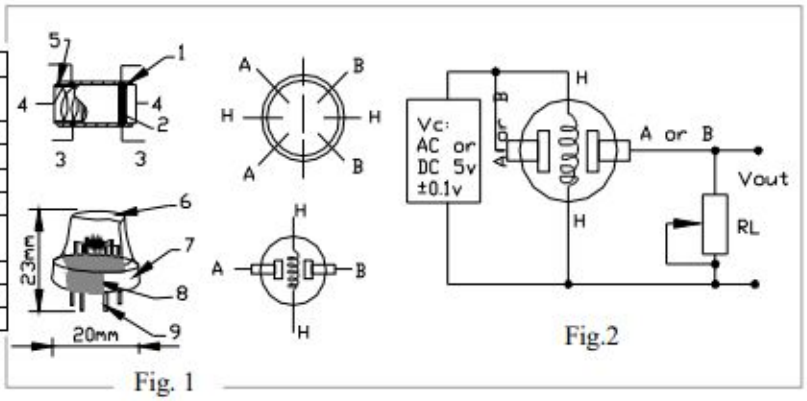
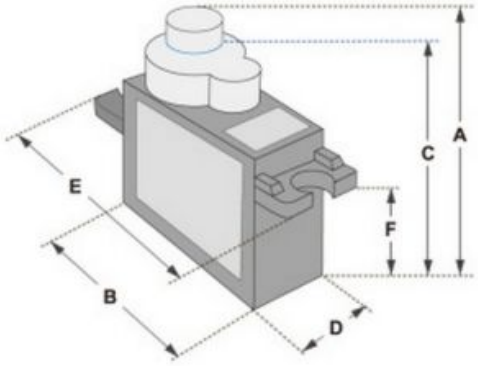


Fig. 1

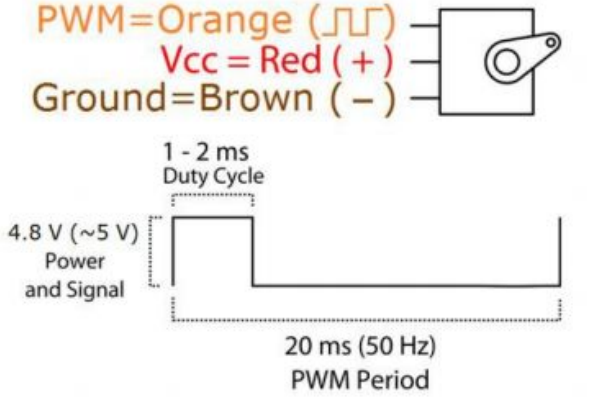
Fig.2

5. SERVO motor



Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.



There are two types of Servo motors, continuous and sweep. More about SERVO Motors

Title	Servo Motor
TI Item Name	STEMKT/AC/D
Description	360 degree, continuous rotation servo motor with gearing and feedback system; used in driving mechanism of robots.
Category	Motors
Hub Connection	4-Pin Cable to only this port: OUT 3
Assembly Instructions	Mount a gear to the top of the Servo Motor using one of the provided screws.
Precautions	Use an Auxiliary Power Source. Do not hold the Servo Motor's shaft while it is rotating. Also, do not rotate the Servo Motor by hand.
Specifications	Operating Speed: 110RPM (4.8V), 130RPM (6V) Stall Torque: 1.3kg.cm/18.09oz.in (4.8V), 1.5kg.cm/20.86oz.in(6V) Operating Voltage: 4.8V~6V

6. LCD Display

1. Features

1. 5x8 dots with cursor
2. 16characters *2lines display
3. 4-bit or 8-bit MPU interfaces
4. Built-in controller (ST7066 or equivalent)
5. Display Mode & Backlight Variations
6. ROHS Compliant

LCD type	☐TN			
	☐FSTN	☑FSTN Negative		
	☐STN Yellow Green	☐STN Gray	☐STN Blue Negative	
View direction	☑6 O'clock	☐12 O'clock		
Rear Polarizer	☐Reflective	☐Transflective	☑Transmissive	
Backlight Type	☑LED	☐EL	☐Internal Power	☑3.3V Input
		☐CCFL	☑External Power	☐5.0V Input
Backlight Color	☑White	☐ Blue	☐ Amber	☐Yellow-Green
Temperature Range	☑Normal	☐Wide		☐Super Wide
DC to DC circuit	☐Build-in		☑Not Build-in	
Touch screen	☐With		☑Without	
Font type	☑English-Japanese	☐English-Europen	☐English-Russian	☐other

2. MECHANICAL SPECIFICATIONS

Module size	80.0mm(L)*36.0mm(W)* Max13.5(H)mm
Viewing area	64.5mm(L)*16.4mm(W)
Character size	3.00mm(L)*5.23mm(W)
Character pitch	3.51mm(L)*5.75mm(W)
Weight	Approx.

7.Buzzer

FEATURES

- wire leads with feedback
- 12 Vdc rating
- low profile
- 4.5 kHz rated frequency

RoHS

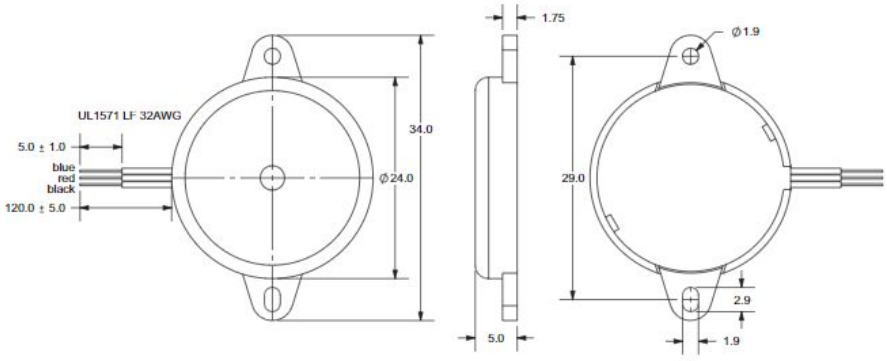
SPECIFICATIONS

parameter	conditions/description	min	typ	max	units
rated voltage			12		Vdc
operating voltage		3		28	Vdc
current consumption	at rated voltage			13	mA
rated frequency		4,000	4,500	5,000	Hz
sound pressure level	at 30 cm, rated voltage	83			dB
dimensions	Ø24.0 x 5.0				mm
weight				7.4	g
material	ABS UL94 1/16" HB High Heat (black)				
terminal	wire leads				
operating temperature		-30		85	°C
storage temperature		-40		95	°C
RoHS	2011/65/EU				

MECHANICAL DRAWING

units: mm
tolerance: ±0.5 mm

WIRE CONNECTIONS	
Color	Function
Red	+terminal (M)
Black	-terminal (G)
Blue	feedback (F)



8.Keypad

Design

The unique design of the housing allows the keypad to be used in the most demanding surroundings. It can be installed directly onto a metal surface without insulation.

Features

- Robust moulded plastic housing
- Indoor or outdoor mounting
- Request to Exit input
- Open collector output (door relay/strike)
- IP67 rating

Specifications	
Input power	10.8 to 30 VDC
Current consumption	130 m A max.
Material of design housing	Plastic
Dimensions (WxHxD)	4 x 140 x 20 mm
Operating temperature	-40 to +55 C
Storage temperature	-40 to +55 C
IP rating	IP67
Cable	2 m LiYY (included)
Addressing	Via keypad
EMC	C-Tick
Pry-off tamper	Optical sensor
Open collector output	10 mA max.

Appendix B

Programming Review:::

Comparison of C with JAVA and C++ ::

Aspects	C	C++	Java
Developed Year	1972	1979	1991
Developed By	Dennis Ritchie	Bjarne Stroustrup	James Gosling
Successor of	BCPL	C	C(Syntax) & C++ (Structure)
Paradigms	Procedural	Object Oriented	Object Oriented
Platform Dependency	Dependent	Dependent	Independent
Keywords	32	63	50 defined (goto, const unusable)
Datatypes : union, structure	Supported	Supported	Not Supported
Pre-processor directives	Supported (#include, #define)	Supported (#include, #define)	Not Supported
Header files	Supported	Supported	Use Packages (import)
Inheritance	No Inheritance	Supported	Multiple Inheritance not Supported
Overloading	No Overloading	Supported	Operator Overloading not Supported
Pointers	Supported	Supported	No Pointers
Code Translation	Compiled	Compiled	Interpreted
Storage Allocation	Uses malloc, calloc	Uses new , delete	uses garbage collector
Multi-threading and Interfaces	Not Supported	Not Supported	Supported

Comparison of Python with C , C++ , Java

	C	C++	Java	Python
Object oriented	No	Yes	Yes	Yes
Functional	No	Yes	No	Yes
Type safety	Unsafe	Unsafe	Safe	Safe
Type expression	Explicit	Explicit	Explicit	Implicit
Type checking	Static	Static	Static	Dynamic
Failsafe I/O	No	No	Yes	Yes
Readability	Difficult	Difficult	Difficult	Easy
Learning	Difficult	Difficult	Difficult	Easy
Language	Programming	Programming	Programming	Programming and scripting
Length of code	5-10 times greater than python	5-10 times greater than python	3-5 times greater than python	Small and manageable codes

Appendix C::

Troubleshooting

The following troubles were seen while performing the following project

- The first and foremost trouble was that due to the pandemic, we were not able to complete the project using the hardware. And had to depend on the online virtual tool tinkercad. So we left out the experience and the learning which we could have acquired by marking a working hardware model.
- The next troubleshoot was that tinkercad does not support any IoT or wifi modules. Due to which the simulation of the block related to the network of things could not be completed. We as a result made a website that could show us how it would have worked with a functioning IoT module.
- As our system required large numbers of sensors one single Arduino was not enough for it. We lacked digital and analog pins. And as the tinkercad does not support Arduino mega, we decided to use two Arduinos and connect them using I2C protocol.

Appendix D: Real Life Mounting of Our System.

In real life, this system can be placed at many buildings, homes, corporate offices, Shops, Banks, or anywhere security is important. In real life, for a big complex, we need to use more than just one sensor to implement it, and also instead of wires they can be connected via BlueTooth. The algorithms of this code are strong enough to work even at huge buildings, and we only require to make changes in terms of Hardware.

In real life, most of the sensors and equipment will be inside the building so it is safe from the weather and the environment. Except for the keypad and LCD display which can be saved from extreme weather conditions by keeping them inside see through the closet with glass doors, outside the main door of the building.

More Power will be used in the large case scenario, as the number of devices connected increases and also wifi access points so in order to solve this problem we can generate power from natural forms of energy like solar energy.

Appendix E: Real-Life Scenarios

When applying this system in real life many problems can be faced and many parameters should be taken into consideration.

- The range of all the sensors used in this system is very low, but in real life scenario we need sensors with large ranges and high power, and so cost may increase according to that.
- And for big places like malls and corporate buildings, more than one sensor should be connected to the system.
- This is the small-scale prototype but when applied in real life, it will use more power compared to this small-scale prototype.
- In the real case scenario, we need a wide range of wifi, for which we need to take extra access points into consideration.
- We might have to use a more efficient microcontroller than Arduino Uno.