CSE3009 – Internet of Things

PROJECT COMPONENT

Slot: F1+TF1

IoT smoke alarm

project submitted by Abhinav.S
17BCE0554

Submitted to:

Parveen Sultana H
Associate Professor
School of Computer Science and Engineering

ABSTRACT:

Aim:

To make an IoT smoke detection system in which the MQ-2 smoke sensor will sense presence of smoke. If smoke is nearby, then the buzzer will start beeping and the red LED will light up and a warning notification will be sent using the NodeMCU module. Live feed of the values with a minor delay will be updated on a ThinkSpeak channel.

Materials required:

1 NodeMCU

1 MQ-2 gas sensor

1 Green, red LED

1 Buzzer

2 220-ohm resistor

Connecting wires

1 Breadboard

1 Standard Power bank of Output 5V/1A or 5V/2.1A

Working:

Hardware:

The MQ-2 smoke sensor will give the output in the form of analog voltage. We have set a condition in our code that if the output value of the sensor is greater than 400 ppm, then the buzzer will start to beep and the red LED will light up; and if the output value of the sensor is less than 400 ppm, then the buzzer will remain quiet and the green LED will light up.

The NodeMCU will send notification to the user's smartphone that will warn the user to check the ThinkView app . After uploading the code, this IP address can be seen in the serial monitor.

The MQ-2 gas sensor is connected with the NodeMCU. Connect the VCC and the GND on the gas sensor to the 3.3V and GND pins on the NodeMCU. Then connect the A0 pin on the MQ-2 gas sensor to the A0 on the NodeMCU.

After that, connect the Buzzer and the LEDs to the NodeMCU. Connect the positive on the buzzer with pin D2 on the NodeMCU and the negative on the buzzer with GND on the NodeMCU. Then connect the negative side of the LEDs to the GND through the 220-ohm resistor and the positive side to pins D0 and D1 on the NodeMCU.

Software:

In the setup function, first, we set the baud rate for the NodeMCU at 9600. Then we declared the led pins and the buzzer pin as the output pins because we will give the output from these pins to the LEDs and the buzzer.

Last, we declared the smoke sensor pin as the input because the smoke sensor pin will take the input from the sensor and will give it to the NodeMCU. The function is called and it will set up a server at an address provided by the NodeMCU. The NodeMCU will then send the data at this address to print in the ThinkSpeak channel. Concurrently, we send a notification through PushingBox cloud using PushBullet when smoke detected by MQ-2 gas sensor crosses a particular threshold.

In the loop function, we read the values from the smoke sensor and then applied a condition that if the output value is greater than 400, then the red LED will light and the buzzer will start to beep. If the output value is less than 400, then the green LED will light up and the buzzer will remain quiet.

Proposed system:

The IoT smoke alarm will have the following functionalities:

1)Get the smoke level(in ppm) using MQ-2 gas sensor and makes the buzzer beep and switches off an active green LED and instead switches on a red LED.

2)Update the values in Thinkspeak channel to allow cloud access of data from anywhere to PC or smartphone(via app)

3)Using a Pushbullet service in PushingBox cloud to send warning notification to user if smoke levels are above a certain limit. ThinkSpeak and Pushbullet services are used independently for better reliability.

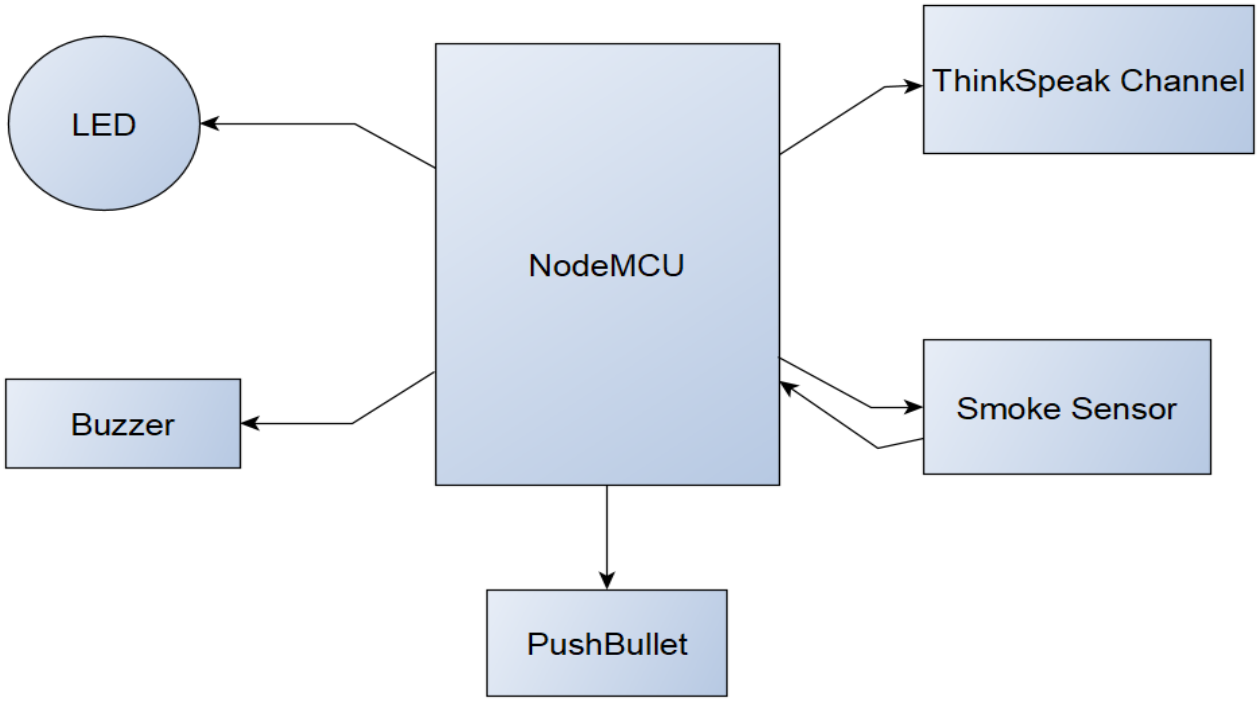
4)An LED to show to people near the device that the alarm is functional and online.

Structure:

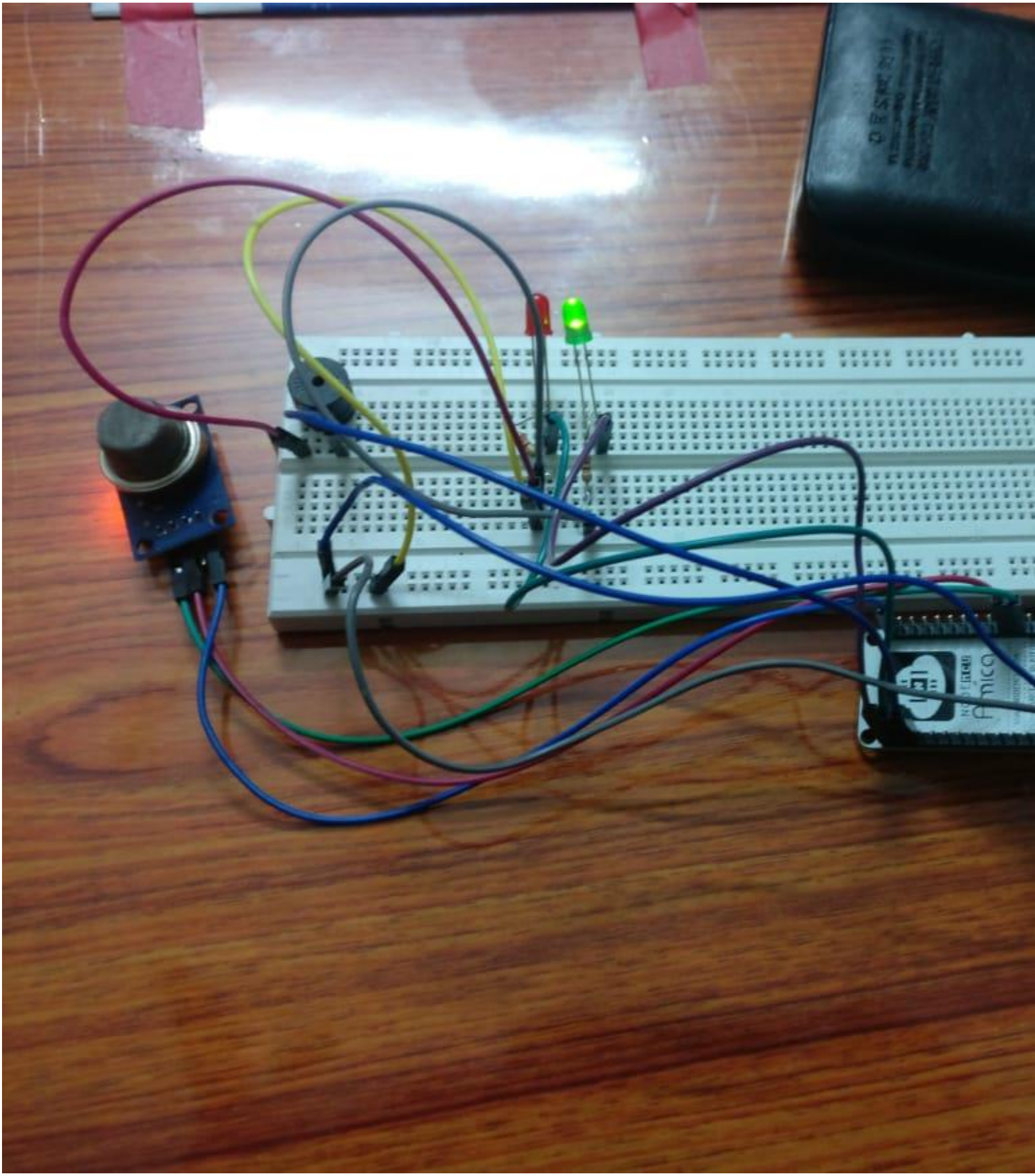
It consists of a microcontroller(NodeMCU), which is connected to the MQ-2 Smoke alarm, two LEDs and a buzzer as follows:
Components:

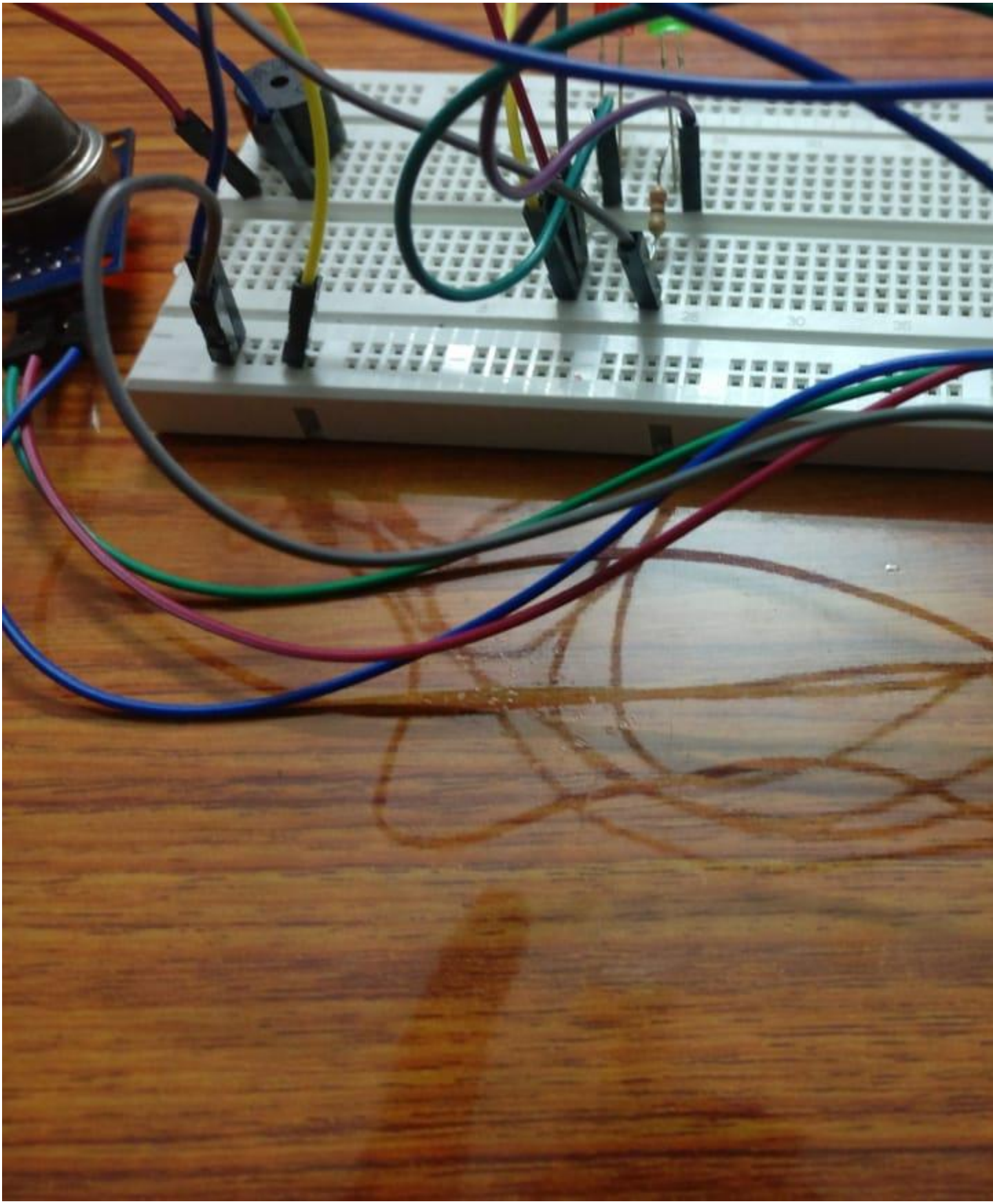
Connections:

Block Diagram:

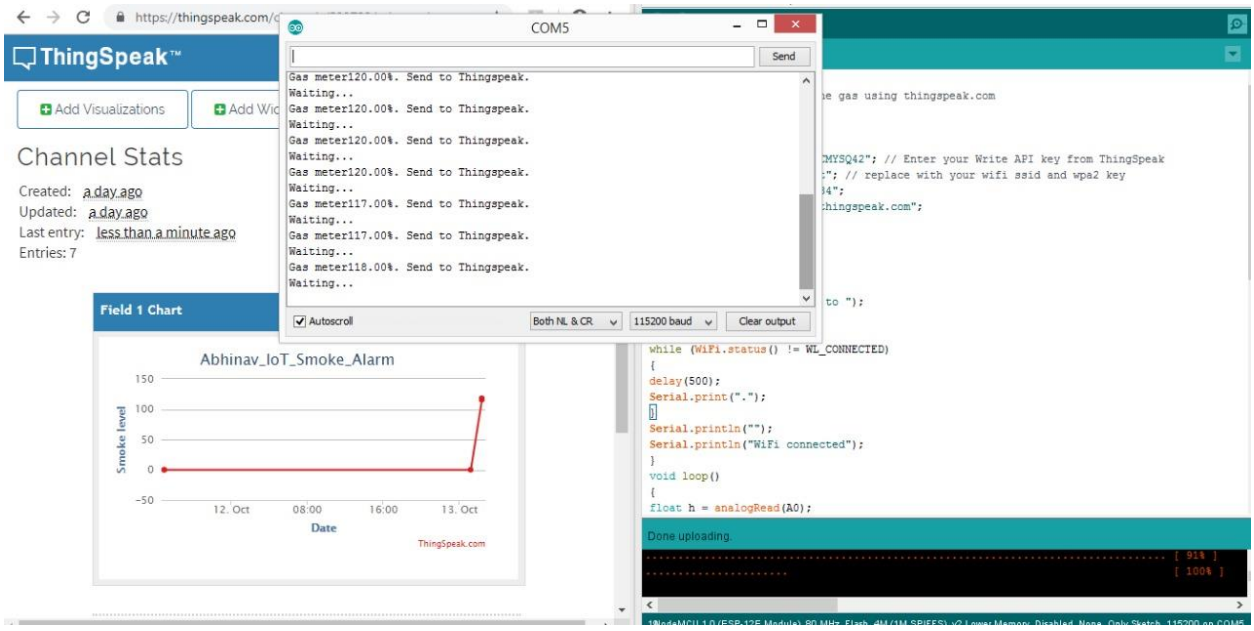


Setup:

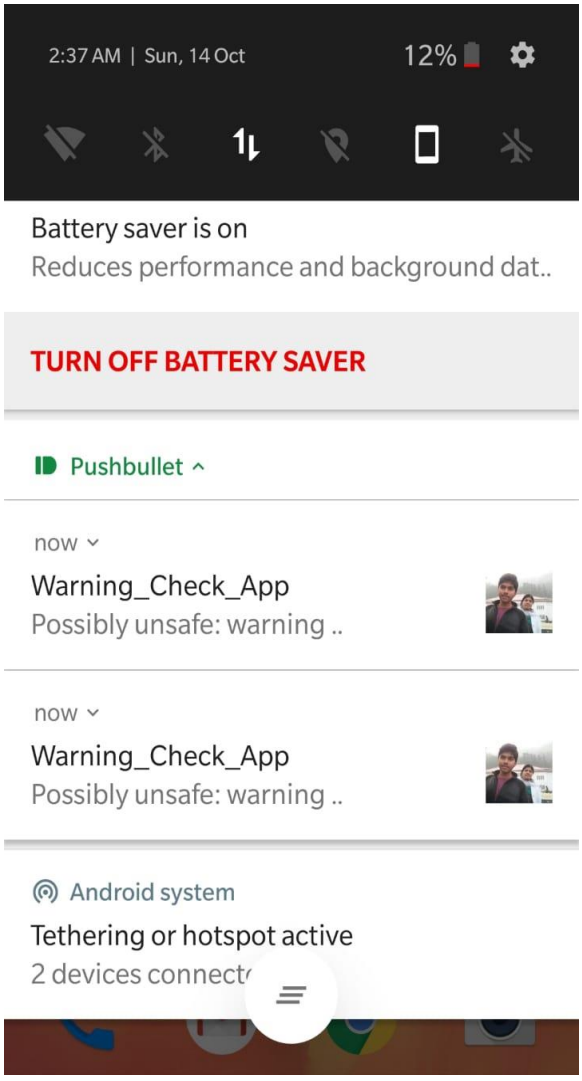


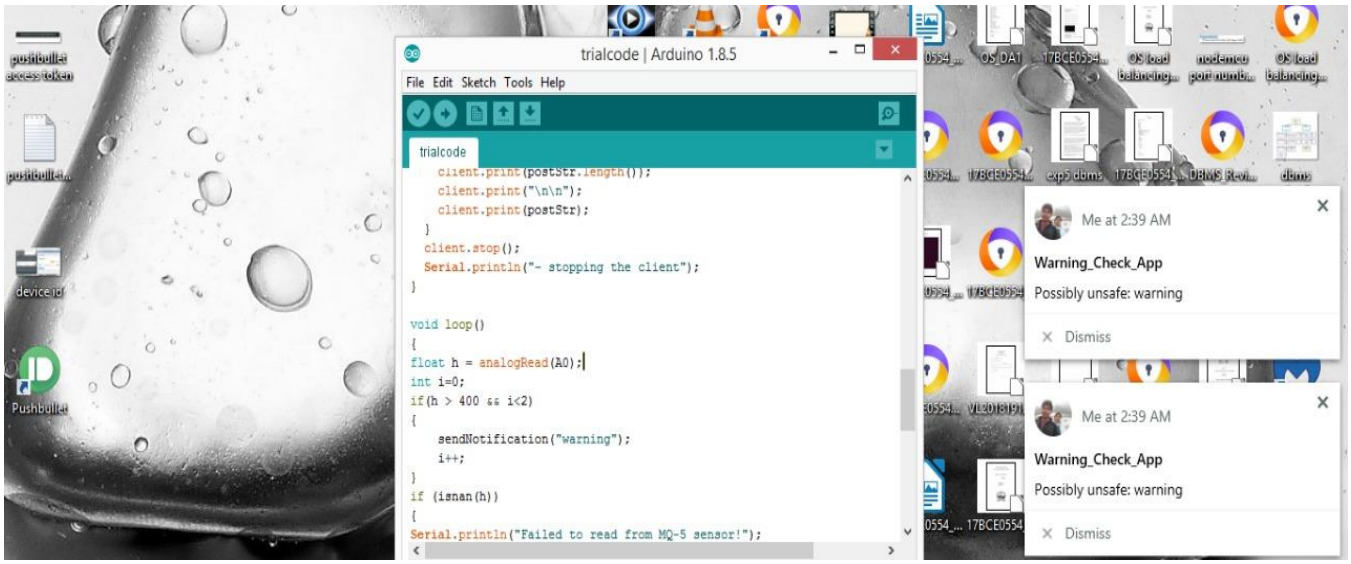


Results:


ThingSpeak™
Add Visualizations
Channel Stats
Created: a day ago
Updated: a day ago
Last entry: less than a minute ago
Entries: 7
Field 1 Chart
Abhinav_IoT_Smoke_Alarm
Smoke level
Date
12. Oct
08:00
16:00
13. Oct
ThingSpeak.com
COM5
Send
Gas meter120.00%. Send to Thingspeak.
Waiting...
Gas meter117.00%. Send to Thingspeak.
Gas meter118.00%. Send to Thingspeak.
Autoscroll
Both NL & CR
115200 baud
Clear output
while (WiFi.status() != WL_CONNECTED)
delay(500);
Serial.print(".");
Serial.println("");
Serial.println("WiFi connected");
void loop()
float h = analogRead(A0);
Done uploading.


Field 1 Chart
Abhinav_IoT_Smoke_Alarm
160
140
120
100
Smoke level
02:30
02:35
02:40
02:45
Date
ThingSpeak.com


2:37 AM | Sun, 14 Oct
12%
Battery saver is on
Reduces performance and background dat..
TURN OFF BATTERY SAVER
Pushbullet
now
Warning_Check_App
Possibly unsafe: warning ..
now
Warning_Check_App
Possibly unsafe: warning ..
Android system
Tethering or hotspot active
2 devices connect



Need for the project:

Fires usually occur in homes because of carelessness and other environmental causalities. They cause threats to the residential community and may result in human death and property damage. Consequently, house fires must be detected early to prevent these types of threats. The immediate notification of a fire is the most critical issue in domestic fire detection systems. Fire detection systems using wireless sensor networks sometimes do not detect a fire as a consequence of sensor failure.

Fire alarms consist of different devices working together that have the ability to detect fire and alert people through visual and audio appliances. The detection devices (i.e., heat, smoke, and gas detectors) detect events and activate the alarm automatically, or sometimes the alarms are activated manually. The alarm may consist of bells, mountable sounders, or horns. Most of the fire alarm systems use the technology of a wireless sensor network (WSN). Today, smart houses and smart cities are equipped with different type of WSNs.

In WSNs, more energy may be consumed because of communication overhead. Thus, most of the time, a sensor's battery is exhausted very fast and it may cause the failure of the sensor or the breakdown of whole network, as houses have different sub-portions and each portion is equipped with one sensor with a single function, which in case of failure causes a system flaw. In this scenario, if an event occurs in a certain portion and the sensor fails to detect the accident, then there is no other way to detect the incident at its initial stage. Spikes in smoke or temperature levels due to various reasons sometimes trigger the alarm. This results in a lot of false alarm incidents. For

example, in the US, the cost of a false alarm is estimated between $30,000 and $50,000 per incident.

Today, sensors are very cheap and very small in size. Thus, to address the above-mentioned challenges, an efficient, IoT-based intelligent home fire prevention system using multiple sensors is required with each of the sensors uses its own mechanism for detection.

Basic diagram of initial smoke alarm set-up:

The microcontroller used is the NodeMCU. Some reasons for choosing this particular microcontroller is:

1)Low power requirements
2)Project requires low computation only
3)Quick setup of sensors and uploading of code
4)Useful if only one program needs to be run
5)cost efficient

The sensor module used to detect smoke is the MQ2 gas sensor. The reasons for choosing it are:

1)Wide detecting range and scope

2)Quick response and high sensitivity

3)Simple drive circuit

4)Reliable and long life

Future scope:

1)Low battery and faulty sensor notifier

2)Multiple sensor system including temperature, olfactory, other MQ series gas sensors etc. to analyse data using a basic ML classifier, which accepts data and predicts accurately in case of fire.

3)Connecting various IoT smoke alarm systems using Zigbee or bluetooth.

4)Hardwiring the system with batteries included as backup.

References:

1)https://maker.pro/NodeMCU/projects/iot-smoke-alarm-NodeMCU-NodeMCU8266-gas-sensor

2)https://www.instructables.com/id/How-To-Smell-Pollutants/

3)http://eie.uonbi.ac.ke/sites/default/files/cae/engineering/eie/SMOKE%20ALARM-project.pdf

4)https://www.dezyre.com/article/top-10-machine-learning-projects-for-beginners/397 (for data set analysis and improving accuracy of alarm system)

5)https://www.instructables.com/id/IoT-Water-Alarm/ (for low battery notification)

6)https://www.instructables.com/id/IOT-Smoke-Detector-Update-Existing-Smoke-Detector-/

7)International Journal of Innovations & Advancement in Computer Science IJIACS ISSN 2347 – 8616 Volume 6, Issue 9 September 2017.Saumya Tiwari1 , Shuvabrata Bandopadhaya2 School of Engineering & Technology, BML Munjal University, Gurgaon, Haryana, India

8) https://www.nfpa.org/News-and-Research/Data-research-and-tools/US-Fire-Problem

9) https://www.mdpi.com/2224-2708/7/1/11/htm