Circuit design for smoke detector

By

Uday Chauhan

Roll no: 210108057

&

Rohan Kanduri

Roll no: 210108020

Under the Guidance of

Dr. Rishikesh Dilip Kulkarni

&

Dr. Mahima Arrawatia

INDIAN INSTITUTE OF TECHNOLOGY

GUWAHATI

Certificate

This is to certify that the work contained in this project "**Circuit design for smoke detector**"*is a bonafide work of* **Uday Chauhan (Roll No. 210108057)** and **Rohan Kanduri (Roll No. 210108020)** *,which has been carried out in the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree*

Dr. Rishikesh Dilip Kulkarni
Dept. of Electronics and Electrical Engineering
IIT Guwahati, India., Assam-781039
May 2024

Dr. Mahima Arrawatia
Dept. of Electronics and Electrical Engineering
IIT Guwahati, India., Assam-781039
May 2024

Acknowledgement

Firstly, We owe our heartfelt gratitude and deepest regard to our supervisors, Dr. Risikesh Dilip Kulkarni and Dr. Mahima Arrawatia, Department of Electronics and Electrical Engineering, IIT Guwahati, India, for their relentless guidance, constant support and full cooperation throughout the project.

We wish to express our deep appreciation to Kethiri Narsimha Reddy for his valuable guidance,assistance, patience and encouragement while developing this project.

We also want to express our sincere gratitude to the Electronics Club, IIT Guwahati for constantly supporting us with their valuable resources and giving us a working environment.

Contents

Chapter 1

Introduction

In our pursuit of designing a smoke detector with better accuracy than most of the smoke detectors available on the market, our main objective was to address the issue of false alarms while operating on lower power. Recognizing that the existing motherboards for smoke detectors lacked the desired functionality, we set out to enhance the device's performance.

After careful consideration, we turned our attention to the Arduino Mega. The Arduino Mega contained numerous components that were unnecessary for our specific task. Using EasyEDA, we gained access to the sketch of the Arduino Mega. We removed unnecessary pins and components, ensuring that only essential elements remained. Simultaneously, we incorporated additional components, such as the laser driver circuit and the IR circuit, important for augmenting the functionality of the smoke detector.

Once we were satisfied with the modifications made to the Arduino Mega sketch, we proceeded to place an order for the manufacturing of our final motherboard. The resulting motherboard was delivered to us.We needed to source and integrate certain through-hole components onto the board. We carefully selected and ordered the necessary components.

With the arrival of the ordered through-hole components, we soldered them onto the motherboard. By implementing these modifications and incorporating the necessary components, we have completed the smoke detector motherboard with improved accuracy.

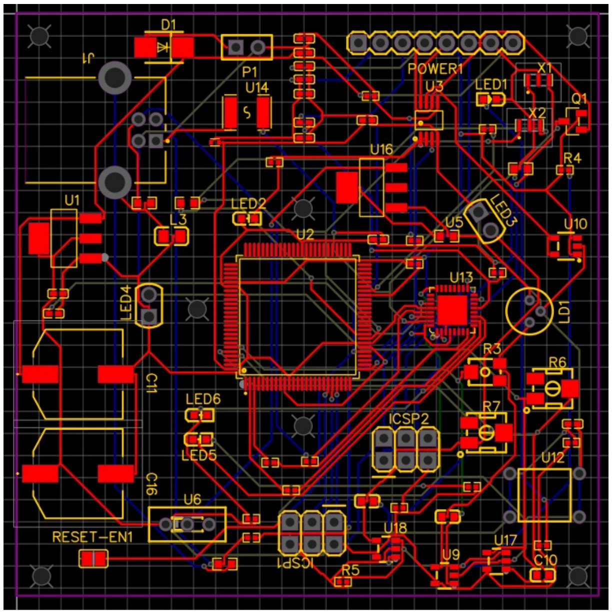


Figure 1.1: Schematic Diagram of PCB

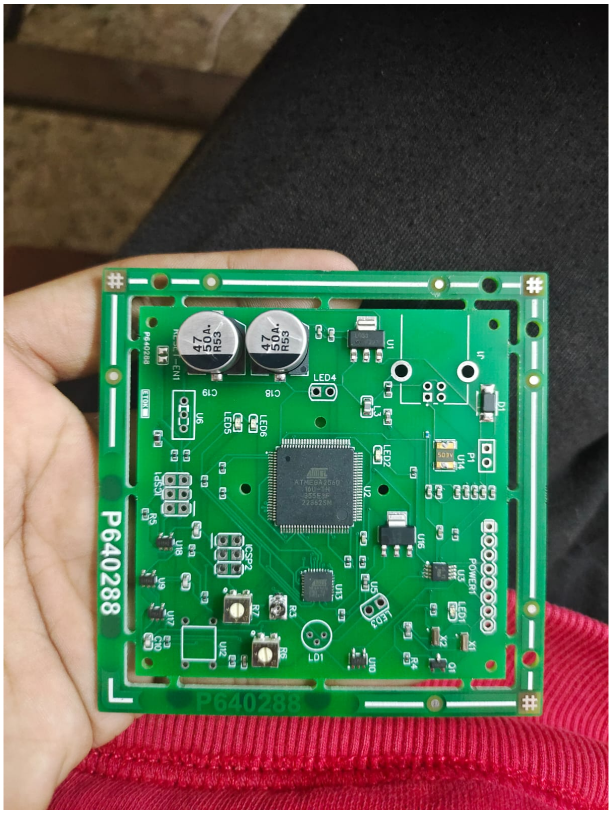


Figure 1.2: Schematic Diagram of PCB

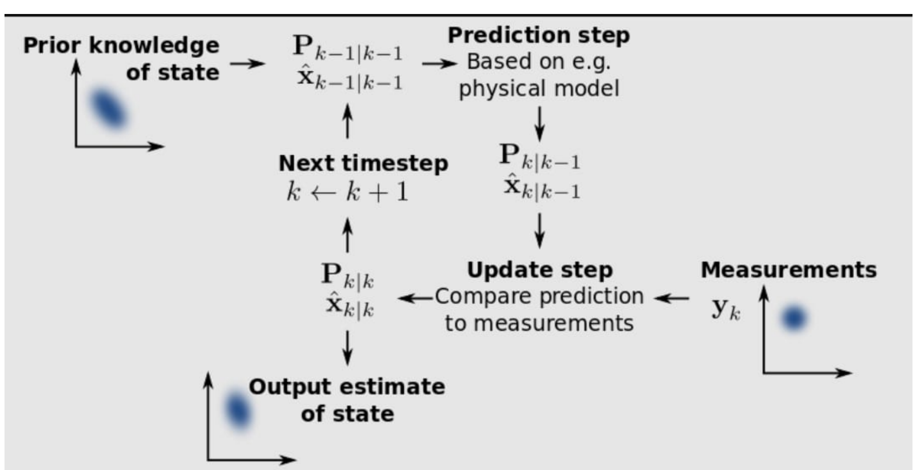
Chapter 2

Methodology

To make a good smoke detector fire sensor its sensibility should be good enough and for good sensibility we are using customoized mother board other than that board we purchased some other components

After connecting all the components on the board we dumped the arduino code on our mother board to make our code efficient we made a python code which reads data from arduino to our serial monitor.we took a lot of samples and to check the threshold of fire/smoke we will use halmon filtering.

Halmon filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed. Time,including statistical noise and other inaccuracies, produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone by estimating a joint probability distribution over the variables for each time frame.



Components Used:

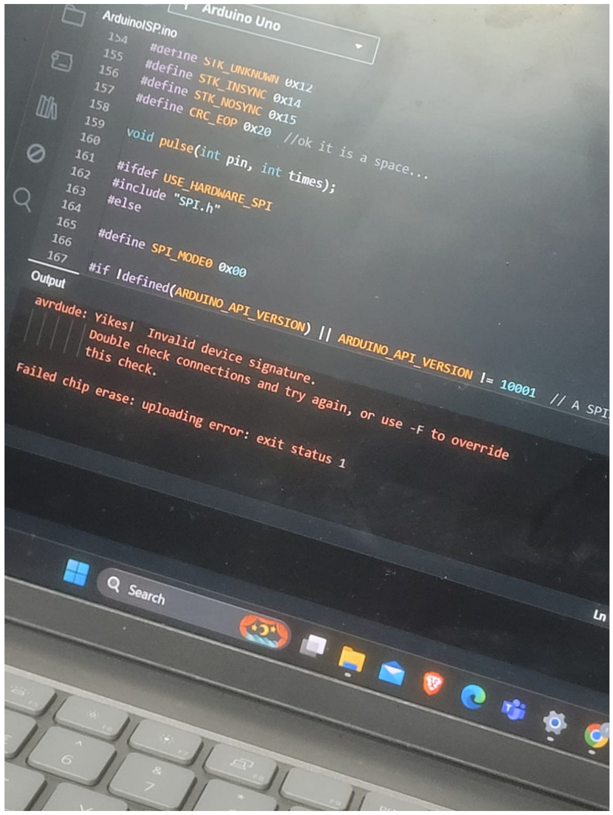
1. Customized board
2. 2A-switch
3. IR sensor
4. Laser diode
5. USB connector

2A switch is used to control the laser diode when laser diode is on there is no smoke then IR sensor will deliver a certain voltage data to our board hence no alarm will be there when there will be smoke inside the chamber then laser is on ,the IR sensor will give a different voltage data to our board and hence alarm will be there.

Chapter 3

Problems Faced

Our major problem was bootloading the microcontroller, atmega2560. When we ordered a customized motherboard, they implemented a new microcontroller or microprocessor, which is in factory reset mode. If we dump some random or arbitrarily generated Arduino code into that chip, then it shows an invalid signature. Also, the COM port cannot be recognized until we burn the bootloader in those microcontrollers.



Now, after bootloading, if we dump some arbitrarily code on our board, it cannot be changed with a USB cable. If we want to change the code on our chip, then we have to bootload it again, and then it will be ready to accept the new code.

Our reset function was not working properly, which is the main reason bootloading was not able to be completed. Then we took the original Arduino board and connected that Arduino board to the oscilloscope while uploading some code to it. We saw a circular pattern of potential variation on the oscilloscope, and we repeated the same procedure on our customized board, but the potential pattern was not there on the oscilloscope.

Then we thought there might be some problem with the clocking of the customized board, so we searched about it on the internet and found that there is another clock crystal inside the microcontroller that can be made to run without an external clock. Then we removed two clock crystals from that customized board and observed the potential variation pattern of the reset pin on the oscilloscope, but still we were not able to figure it out.

Bootloading of the microcontroller (Atmega 2560) is done by ICSP1 pins, and for other microcontrollers, it is done by ICSP2 pins. After bootloading, we have to code our Arduino very carefully; otherwise, we will have to start from the above step.

For writing the code in Arduino, we require some tests on different smokes, and we have to distinguish between safe and hazardous situations. Then we have to take a lot of samples, and we will have to find out the threshold manually.

On our board, the sensors are connected through variable resistances. To get the correct threshold on the alarm, we have to adjust it accordingly. This part is the toughest, as it can be damaged if we rotate it again and again.

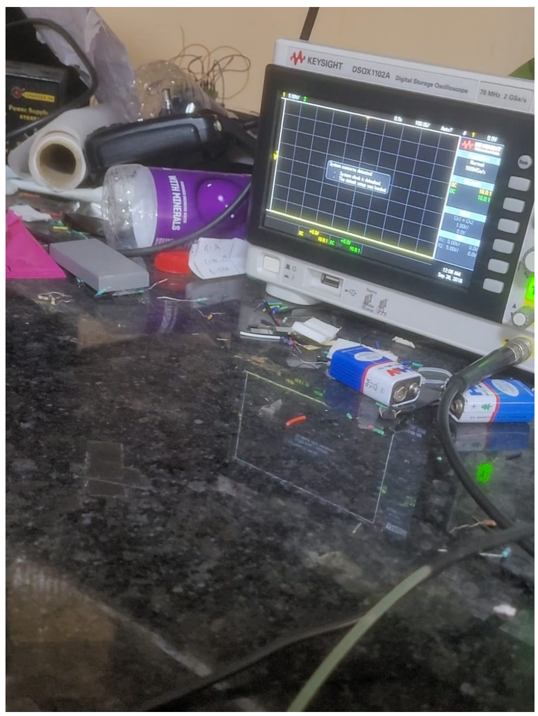


Figure 3.1: Potential of Reset Pins

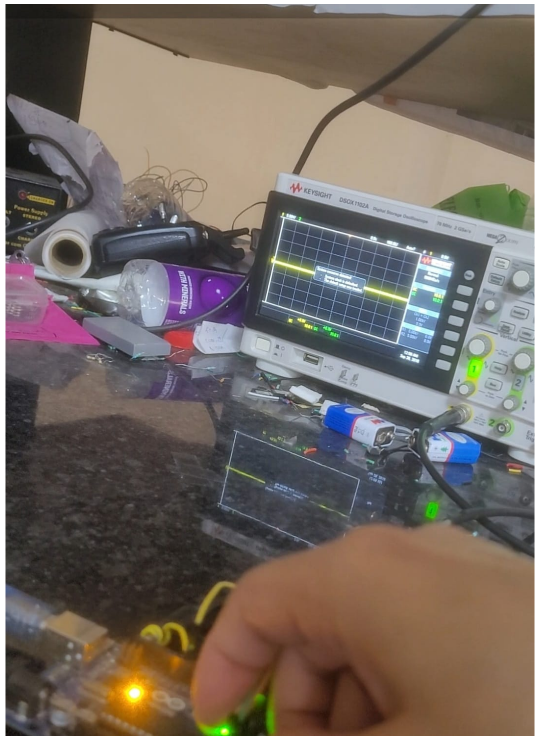
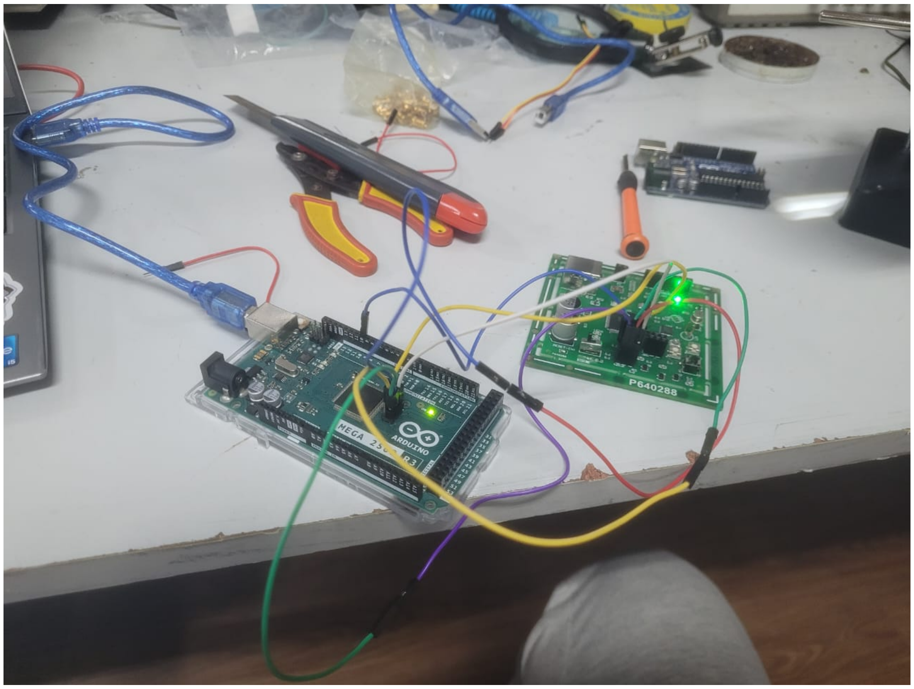


Figure 3.2: Potential of Reset Pins

How we solved:

So our main problem was bootloading our microcontroller. After removing quartz crystals from our board, we didn't find any significant change in our reset pin, so we took a new board and started to bootload it from another Arduino Mega through the ICSP1 pins.



And we were able to do it successfully and now the problem was of recognition of COM port then we did the same connections between arduino mega to ICSP2 pins and we were able to do it successfully now our board is ready to accept any arduino code in it

Now we had to read values from 2 sensors through arduino mega then we upload code on it

```
// Define the analog pin
const int analogPin1 = A0;
const int analogPin2 = A1;
void setup() {
  // Initialize serial communication at 9600 baud rate
  Serial.begin(9600);
}
void loop() {
  // Read the analog value from the defined pin
  int sensorValue1 = analogRead(analogPin1);
  int sensorValue2 = analogRead(analogPin2);
  // Print the analog value to the serial plotter
```

```
  String output = String(sensorValue1) + " " + String(sensorValue2);
  // Print the concatenated string
  Serial.println(output);
  // Add a short delay before taking the next reading
  delay(100);
}
```

And we used another python code on our computer which links arduino IDE to our python IDE. python code was used for recording the samples and storing them into csv files into computer

```python
import serial
import csv
from datetime import datetime
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import numpy as np
import threading
import time
import statistics

# Configure the serial port
port = "COM14"  # Change this to match your Arduino's port
baudrate = 9600

# Open the serial port
ser = serial.Serial(port, baudrate)

# Initialize data structures
plotting_data = [[] for _ in range(2)]  # Initialize with 2 lists
average_data = [0, 0]  # Initialize with 2 elements
current_time = datetime.now().strftime("%H%M%S")
```

```
# Function to collect data from serial port
def data_collection():
    # Open the CSV file for writing
    filename = "DATA_COLLECTION_" + str(current_time) + ".csv"
    with open(filename, "w+", newline='') as csvfile:
        csvwriter = csv.writer(csvfile)
        c = 0
        try:
            while True:
                if ser.in_waiting > 0:
                    line = ser.readline().decode().strip()
                    print(line)
                    if line:
                        data = line.split(" ")
                        print(data)
                        if len(data) == 2:
                            for i, value in enumerate(data):
                                value = float(value)
                                plotting_data[i].append(value - average_data[i])
                                plotting_data[i] = plotting_data[i][-150:]  # Keep last 150 data poin
                            c += 1
                            csvwriter.writerow(data)
                            csvfile.flush()
                            print(c)
                            if c > 149:
                                break
        except KeyboardInterrupt:
            print("Data collection interrupted")
        finally:
```

```
            # Close the CSV file and serial port
            csvfile.close()
            ser.close()

def calculate_avg():
    for i, row in enumerate(plotting_data):
        if row:  # Ensure the list is not empty to avoid division by zero
            average_data[i] = statistics.mean(row)

if _name_ == "_main_":
    data_collection()
    # Calculate initial averages
    calculate_avg()
    print("Initial averages:", average_data)
    print("Start the test")
```

That csv file can be accessed by matlab can be plotted now after plotting certain smokes we will get different graphs now we will see the different threshold and we will set the variable resistance on our board accordingly

Chapter 4

Observation

If true smoke goes inside the chamber of the smoke detector, then LED2 will glow up on the board.

But the LED2 is inside our chamber, so we will not be able to know, so there will be a USB connected from the smoke detector to our computer or buzzer system. When the smoke density is higher than that, we put on a threshold value, and our smoke detector will detect it.

Figure 4.1: With Smoke

Figure 4.2: Without Smoke

Chapter 5

Conclusion

Now it's time for a conclusion. Our smoke detector has now started to detect hazardous fires and smoke very efficiently. It is quite powerful and efficient compared to most of the smoke detectors available on the market. Generally, smoke detectors use very basic microcontrollers, but here in our microcontrollers, we can adjust sensitivity according to our needs and uses.

Then the microcontroller that we used, the Atmega 2560, is quite powerful, so we can drive our smoke detector very easily and efficiently.

Chapter 6

Future Works

We can do more signal processing with different types of smoke and optimize our code for our smoke detector. In our model, we are checking the results through our computer, but in the future, we can also implement a buzzer speaker inside it, or we can also make a buzzer system separately.

We can also add a bluetooth or wifi module to our customized board, so when fire or smoke gets in, our system will be connected wirelessly and a buzzer can be formed on it,so that we can take protective measures.