Department of Electrical and Computer Engineering
North South University

Md Sadikur Rahman Sadeed
ID 2112251642

CSE 323
Section : 03

Faculty Advisor: Dr Mosabber Uddin Ahmed

ECE Department , Summer, 2024

Abstract

In this report, the construction and implementation of a fire detector consisting of an Arduino Uno, a flame sensor module, and ESP8266 ESP-01 Wi-Fi are described. What is more, it is possible to receive information about deviations inside the system via both local alarm and e-mail. As a part of a science fest project, this work shows how hardware sensors can work in conjunction with wireless protocol and a back end system for real time monitoring with alert functionalities.

1. Introduction

Objective

The aim of this project is to design a low cost fire detection system which is a combination of hardware and software to inform the users that there is fire danger. Using Arduino Uno to control the sensors, ESP8266 to implement the wireless capability, this system allows local and remote notifications.

Rationale

Fire circumstances remains one of the major causes of life and property loss throughout the world. Conventional fire alarms are costly and do not have modern compatibility options. This project closes this gap by offering a cheaper solution with an email notification feature to guarantee timely intervention when there is an emergency.

Background Studies

There is sufficient evidence that Arduino-based systems have been applied in IoT applications. Flame sensors detect fire by sensing infrared light from flame and Wi-Fi module such as ESP8266 for data transmission. Prior research discusses how such devices can enhance early fire alert detection systems and incorporate it into smart home systems【1】【2】.

.

2. Technology and Methodology

2.1 Hardware Components

The following components were used in the project:

Arduino Uno: Responsible for controlling the processing of sensor signals which are acquired through specialized sensors.
Flame Sensor Module: Is sensitive to the infrared radiation from flames.
ESP8266 ESP-01 Wi-Fi Module: Supports wireless operations.
Breadboard and Jumper Wires: Enable circuit networks improvements.
Miscellaneous: It has an alarm buzzer and a local computer for the backend.

2.2 Software Tools

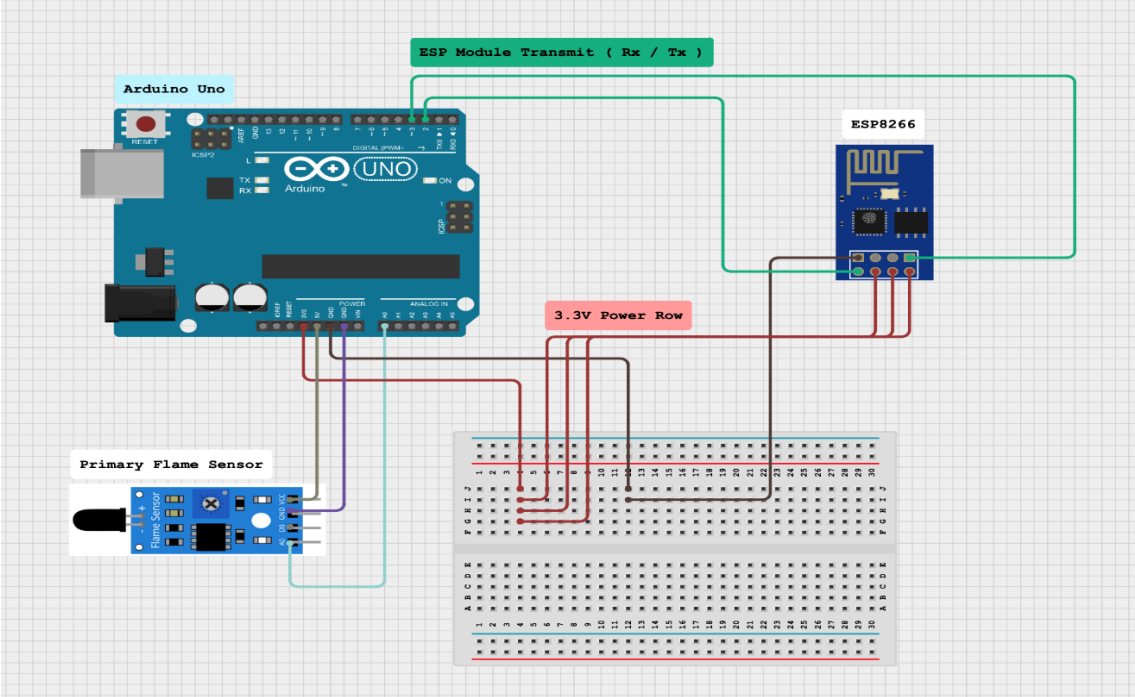
PlatformIO: Used for writing and transferring the codes for Arduino.
Python 3.8: System language used for processing the alerts or the emails.
Visual Studio Code: Exclusive editing platform for codes using the pro tools such as PlatformIO and Python extensions.
Cirkit Designer: Employed to design and draw circuit configuration.

2.3 System Architecture

Data Acquisition: Flame sensor module finds variations in thermal radiation.
Data Transmission: The Arduino analyzes data and transmits them to the ESP8266 module.
Backend Processing: Python scripts take data over TCP, can sound alarms and send email messages.
User Alerts: Local alarms are sounded instantly while emails are sent out for remote notification.

2.4 Circuit Design

Connecting the flame sensor with Arduino and ESP8266 is done with ease through the mentioned circuit. To the Arduino board jumper wires is used to connect the sensor at analog input pins while for Wi-Fi the ESP8266 connects to the digital pins.



3. Results and Observations

The system's effectiveness for identifying flames and conveying messages was assessed in simulated conditions. Key results include:

Detection Accuracy: The results indicated that the flame sensor was effective in a detecting fire at a distance of 3 meters.

Response Time: The local alarm was provided within 2 seconds whilst emails where generated within 5 seconds of an alert being raised.

Network Stability: The ESP8266 provided stable connection with the local network during the tests.

Ease of Use: It would also be possible to simply adjust all the e-mail connected settings and receive notices devoid of professional IT knowledge.

4. Discussion and Future Work

The project thus proves the possibility of using Arduino with IoT in firefighting processes. While the current system performs reliably in ideal conditions, there are areas for improvement:Multi-Sensor Integration: Bolstering the smoke or temperature sensors as a way of increasing its reliability.

Mobile Application: Designing a Smartphone application for notification and controlling.

Battery Backup: The system must operate during power failures.

Subsequent works could consider more sophisticated communication paradigms such as MQTT for smart homes in terms of growth.

5. Conclusion

Thanks to the integration of both hardware and software parts, this fire detection system can be considered as affordable and efficient one. It points to the possibility of enhancing safety by using IoT and at the same time illustrates that Arduino based systems are cogent applicative tools.

References

1. R. B. El-Basioni, H. F. Alam, and M. M. Salama, "IoT-based fire detection and notification system," *IEEE Access*, vol. 6, pp. 88380–88391, Dec. 2018.
2. A. Gupta, M. Vashisth, and P. Sharma, "Smart fire alarm system using Arduino," *International Journal of Scientific Research in Engineering and Management*, vol. 4, no. 4, pp. 12–17, Apr. 2021.