* AIM :

Fire detection System Using ESP8266 using temperature and smoke sensors,ensuring rapid response and enhanced safety to detect flames

* PROBLEM STATEMENT:

Design and develop an fire detection system utilizing an ESP8266, bread board,lighter,power supply,Flame sensor and connecting wires components to create a seamless checkout process. The system should accurately detect the fire flames and makes an alarm to protect every one

**SOLUTION:**

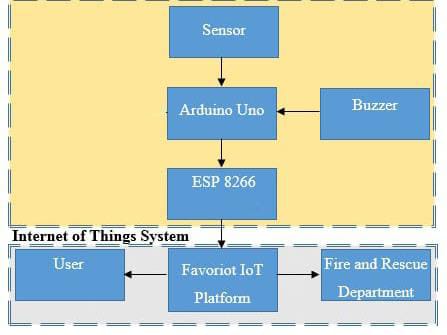
Certainly! Building an automated billing system using an Arduino Uno involves integrating components like a barcode/QR code scanner, a display module, and structuring the software to manage the scanning and billing processes. Here's a general solution outline.

**Hardware Setup:**

* **ESP8266:** Acts as the central control unit.
* **Flame sensor:** Connected to the ESP8266 for detect the flames

**Software Implementation:**

* **Arduino IDE:**
* This ide used to write the programs and upload them into ESP8266
* The ide is also has a message area,text console and a toolbar
* **Blynk App:**
* The fire detector is powered by the ESP32 nodeMCU which can be controlled remotely via an application using Blynk.
* The system uses a flame sensor, which senses the light of a flame and sends signals to an ESP8266 board.
* **User Interface:**
* Use an LCD or LED display to show the scanned items and the total bill amount.
* Implement a user-friendly menu or prompts for better interaction.
* **Integration with ESP8266:**
* Write program code that coordinates between the fire sensor and ESP8266
* Ensure proper communication and data processing between these components.
* **Testing and Debugging:**
* Test each component individually to ensure functionality.
* Combine components and run tests to check the system's overall operation.
* Debug any issues encountered during testing.
* **BASIC BLOCK DIAGRAM FOR THE SOLUTION:**



* **Program code:**

#define FLAME\_SENSOR A0 // Pin connected to the flame sensor

#define LED\_PIN 2 // Pin connected to an LED for indicating fire detection

#define THRESHOLD 400 // Threshold value for flame detection

void setup() {

pinMode(FLAME\_SENSOR, INPUT);

pinMode(LED\_PIN, OUTPUT);

Serial.begin(9600);

}

void loop() {

int sensorValue = analogRead(FLAME\_SENSOR);

Serial.print("Sensor value: ");

Serial.println(sensorValue);

if (sensorValue > THRESHOLD) {

digitalWrite(LED\_PIN, HIGH); // Turn on the LED to indicate fire detection

Serial.println("Fire detected!");

} else {

digitalWrite(LED\_PIN, LOW); // Turn off the LED

}

delay(1000); // Delay for stability

}

* **OUTCOMES FOR THE SOLUTION:**

The outcomes for the solution of Fire detection system using ESP8266 with temperature ,smoking censors

* **Improved Accuracy and Reduced Errors:**
* Minimize human error in manual fire detector
* Accurance of this system is high compared to other devices
* **Testing and Validation**: Thoroughly test the system to ensure reliable detection of fires and proper functioning of alert mechanisms.
* **Documentation and Understanding:**
* Creation of detailed documentation outlining system architecture, components used, code implementation, and usage instructions, facilitating easy understanding and potential future development.
* **Fire sensor:**

A photosensor that can detect flames and light wavelengths within 760nm-1100nm. It can also detect lighter flames from a distance of 80 cm. The sensor has a 60 degree detect angle and is very sensitive to the flame spectrum.

* **Buzzer alarm:**

When a fire is detected, the NodeMCU turns on the buzzer.

Alert email

When a fire is detected, the NodeMCU automatically sends an alert email to the person using SMTP2GO.

* **Alarm board:**

The Node MCU ESP8266 can be programmed to turn on an alarm board and a water pump with a location estimator when a fire occurs.

Thinger.Io Web

If the sensor detects a fire hazard, the microcontroller sends a danger status to Thinger.Io Web. Thinger.Io Web then sends a notification of a detected fire to the user's mobile device via the

* **Learning and Skill Development:**
* Opportunity for individuals to learn about integrating hardware components, handling fire sensor, and coding for ESP8266 using Arduino.
* **Practical Application:**
* The project serves as a practical demonstration of integrating hardware and software components for real-world applications.

**Conclusion:**

Smart home in the form of an early detection system for potential home fires can be designed and built based on the IoT concept. The system design is an integration of hardware and software assembly consisting of Arduino Mega 2560, ESP8266, datalogger, fire sensor, gas sensor, Arduino IDE and the Blynk application on an Android smartphone. The results of the system design are able to work and show the system's response to the presence of fire and the presence of gas leaks in the kitchen. The system design results are displayed via the Blynk application on a smartphone. The system performance towards the presence of fire is indicated by changing the detection of fire in the form of changing the Virtual Button from the initial color to red. The display of a gas detection system in the form of a Virtual Level moves when a gas leak is detected. Fire sensor activity recording data from the datalogger is displayed in the form of a graph of the relationship of the working voltage of the fire sensor versus time. While the data recording of gas sensor activity from the datalogger is displayed in the form of a graph of the relationship of gas concentration versus time. All system response results and graphic data are displayed in realtime through the Blynk application on a smartphone. In addition, record system performance data stored on the micro SD card for further analysis. From the results of the design of this system is expected to be used as a reference technology for home monitoring of potential fire equipped with a data storage system.