**ETEC 306**

**Sangaran Subramaniyam ID: 301251046**

**Proposal for Arduino’s use in fire alarm temperature sensor.**

On behalf of my team, my contribution was to do research on the Arduino R3 UNO and its compatibility to our temperature sensing Fire alarm.

**Introduction.**

Arduino is a versatile microcontroller platform that can be used to create a wide range of projects, including temperature sensor-based fire alarms. Arduino's compatibility with a temperature sensor for a fire alarm depends on the type of temperature sensor you use and how you interface it with the Arduino.

**Compatibility.**

Here's a general outline of how you can make Arduino compatible with a temperature sensor for a fire alarm:

1. **Select a Temperature Sensor**: There are various temperature sensors available, such as the DHT11, DHT22, DS18B20, LM35, and more. Choose a sensor that is suitable for your specific application in terms of accuracy, range, and interface (digital or analog).

2. **Wire the Sensor:** Connect the temperature sensor to the Arduino according to its datasheet or documentation. Depending on the sensor, you might need to connect it to specific pins (analog or digital) on the Arduino board.

3. **Code the Arduino:**Write the Arduino code to read the temperature data from the sensor. You will need to use the appropriate libraries and functions for the chosen sensor. You may also need to define temperature thresholds that trigger the fire alarm.

4. **Fire Alarm Logic:** Implement the logic for the fire alarm. Typically, this would involve checking the temperature readings and comparing them to predefined thresholds. If the temperature exceeds a certain threshold, trigger the alarm. The alarm can be in the form of a buzzer, LED, or a notification sent through various means (e.g., Wi-Fi, GSM module, etc.).

5. **Power Supply:** Ensure that you have a reliable power supply for your Arduino and sensor, especially in the event of a fire. You might consider using a backup power source or battery to ensure continuous operation during power outages.

6. **Safety Considerations**: When designing a fire alarm system, safety is paramount. Make sure to follow safety guidelines and use appropriate enclosures, fire-resistant materials, and wiring practices to minimize the risk of fire-related accidents.

7. **Testing and Calibration:** Test your fire alarm system extensively under controlled conditions to ensure it works as expected. You may also need to calibrate the temperature sensor to ensure accurate readings.

8. **Compliance and Regulations**: Depending on your location, there may be regulations and standards governing fire alarms and safety systems. Ensure that your project complies with these regulations.

**Summary.**

Arduino can be made compatible with a temperature sensor for a fire alarm by selecting the right sensor, wiring it correctly, writing the appropriate code, and implementing safety measures. Keep in mind that while Arduino can be used for various DIY projects, for critical applications like fire alarms, it's essential to follow best practices and consider professional solutions if necessary to ensure reliability and safety.