Rapid Rescue

Arjun M[1], Alwyn Newman L J[2], Jayasurya V[3], Mobel Robinson J[4]
[1, 2, 3, 4] Dept of Computer Science And Engineering
[1, 2, 3, 4] SNS College Of Engineering, Coimbatore – 641 107

I. EMPATHY

FIGURE 1.9

EMPATHY:

PATIENT 01:

When the accident occurred, the rescue team's arrival was delayed, prolonging the distressing situation. Despite the urgency of the situation, their response time fell short, leaving us waiting anxiously for assistance. With each passing moment, concern escalated as we grappled with the aftermath of the accident. The delay compounded the severity of the situation, heightening feelings of vulnerability and uncertainty. Eventually, when the rescue team did arrive, their presence provided a much-needed sense of relief, though the experience underscored the critical importance of timely emergency response.

PATIENT02:

It appears that there may have been a misunderstanding or miscommunication regarding the details of your accident when informing your relatives. It's important to ensure that accurate information is relayed to them to avoid confusion or unnecessary concern. You may want to consider reaching out to them directly to provide them with the correct information and address any questions or concerns they may have. Additionally, it could be helpful to clarify any discrepancies or misunderstandings that may have occurred during the initial communication. Open and clear communication is key in situations like these to ensure that everyone involved is properly informed and supported.

PATIENT03:

Losing a loved one due to a delay in ambulance service is a devastating experience that no one should have to endure. In such situations, every minute counts, and any delay can have tragic consequences. The pain of losing family members in such circumstances is compounded by the knowledge that prompt medical attention might have made a difference. It's essential for emergency response systems to be efficient and responsive, ensuring that help reaches those in need as quickly as possible. My heart goes out to you for your loss, and I hope that steps are taken to prevent similar tragedies from occurring in the future.

Losing my family due to a delay in ambulance service has left me shattered and filled with unimaginable grief. It was a normal day when suddenly an accident occurred, and I frantically called for help. However, the information about the accident somehow reached the ambulance service too late, and by the time they arrived, it was too late to save my loved ones. The agonizing wait for help that never came will haunt me forever. The precious moments lost in the delay could have made all the difference, but instead, I am left to mourn the irreplaceable loss of my family. The pain is indescribable, and I can only hope that steps are taken to ensure that no one else has to suffer a similar fate due to such delays in emergency response.

DEFINE:

During the critical moments following an accident, timely rescue operations are paramount to saving lives. Any delay in the initiation of rescue efforts can significantly increase the risk of fatalities. This delay may result from various factors such as limited access to the accident site, inclement weather conditions, lack of resources, or communication breakdowns. Regardless of the cause, each passing moment without intervention can exacerbate injuries, worsen medical conditions, or expose victims to additional hazards. In such situations, every second counts, and swift action by trained rescue personnel is essential to providing immediate medical care, extricating individuals from danger, and transporting them to appropriate medical facilities for further treatment. Therefore, minimizing delays in rescue operations is crucial to maximizing the chances of survival and reducing the potential loss of life in the aftermath of an accident.

FIGURE 2.1

IDEATE:

SOLUTION

Certainly! Integrating an IoT-based WIFI module with a piezoelectric sensor in vehicles can offer several advantages, such as vehicle health monitoring, security, and real-time tracking. Here's a detailed breakdown of how you can implement this:

Components Needed:

1.IoT-based WIFI Module:

Choose a reliable and compatible WIFI module with IoT capabilities. Popular options include ESP8266 or ESP32 for DIY projects or specialized modules designed for automotive applications.

2.Piezoelectric Sensor:

Select a piezoelectric sensor suitable for detecting vibrations or impacts. Ensure it's durable and sensitive enough to detect relevant events in a vehicle.

3.Microcontroller:
You'll need a microcontroller to interface between the WIFI module, piezoelectric sensor, and vehicle systems. Arduino or Raspberry Pi are common choices.

4. Power Supply:

Provide a stable power source for the IoT module and sensors. Depending on the setup, you might use the vehicle's battery or an external power supply with proper voltage regulation.

5. Enclosure:

Encase the components securely to protect them from environmental factors and vibrations within the vehicle.

Implementation Steps:

1. Sensor Placement:

Install the piezoelectric sensor in a strategic location within the vehicle, such as near the engine or on critical structural components prone to impacts or vibrations.

2. Hardware Setup:

o Connect the piezoelectric sensor to the analog or digital input pins of the microcontroller.
o Connect the IoT WIFI module to the microcontroller. Ensure proper wiring and voltage levels compatibility.
o Power up the system and test the sensor readings to ensure it detects vibrations accurately.

3. Software Development:

o Write firmware for the microcontroller to read data from the piezoelectric sensor.
o Implement algorithms to analyze sensor data and detect significant events like collisions, sudden braking, or engine vibrations.
o Configure the WIFI module to establish a connection with your IoT platform or server for data transmission.
o Implement protocols for secure data transfer over WIFI, considering the sensitive nature of vehicle-related data.

4. Integration with IoT Platform:

o Set up an IoT platform or server to receive data from the vehicle.
o Develop backend software to process incoming data, perform analytics, and trigger alerts or actions based on predefined criteria.
o Implement features such as real-time monitoring, historical data analysis, and notifications for vehicle owners or fleet managers.

5. Testing and Calibration:

- o Conduct thorough testing of the entire system to ensure reliability and accuracy.
- o Fine-tune sensor thresholds and algorithms to minimize false positives and negatives.

6. Deployment and Maintenance:

- o Install the finalized system in vehicles, ensuring proper mounting and connections.
- o Regularly monitor system performance and address any issues promptly through remote updates or maintenance visits.

Considerations:

- o Data Security: Implement robust security measures to protect vehicle data from unauthorized access or tampering.
- o Power Efficiency: Optimize the system for power efficiency to minimize drain on the vehicle's battery.
- o Regulatory Compliance: Ensure compliance with relevant regulations and standards, especially concerning vehicle safety and data privacy.
- o Scalability: Design the system to scale easily for deployment across a fleet of vehicles if needed.

PROTOTYPE:

- o **WIFI module**
- o **Piezoelectric sensor**
- o **Cables**
- o **Power supply cable**
- o **Vibrative sensor**
- o **Transformer(230v converter 12v)**
- o **AC to DC Converter**
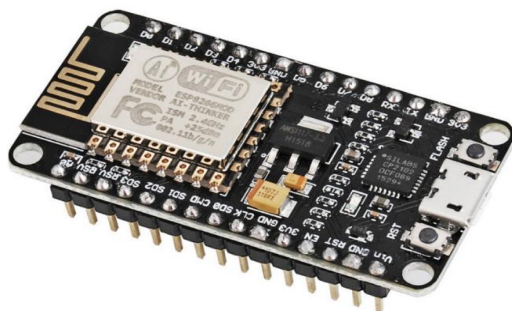- o **Relay module**

WIFI Module:

FIGURE 2.2

An IoT-based WiFi module is a device that enables connectivity to the internet for various IoT (Internet of Things) devices through a WiFi network. These modules typically incorporate a microcontroller unit (MCU), a WiFi transceiver, and often additional features such as onboard memory, security protocols, and GPIO (General Purpose Input/Output) pins for interfacing with sensors and actuators.

Here are some key features and functionalities commonly found in IoT-based WiFi modules:

1. WiFi Connectivity: The module allows devices to connect to WiFi networks, enabling them to communicate with other devices and access cloud services over the internet.

2. Microcontroller:Most WiFi modules integrate a microcontroller unit (MCU) to handle tasks such as data processing, protocol implementation, and interfacing with peripherals.

3.Security: Security features such as encryption protocols (e.g., WPA2, WPA3), secure boot, and data encryption are essential for protecting sensitive information transmitted over the network.

4. APIs and Protocols: Support for various communication protocols and APIs (Application Programming Interfaces) simplifies integration with IoT platforms and cloud services. Common protocols include MQTT, HTTP, CoAP, and WebSocket.

5. Power Management:Efficient power management mechanisms, such as low-power modes and sleep states, help prolong battery life in battery-operated IoT devices.

6. OTA (Over-the-Air) Updates: OTA update capabilities allow firmware and software updates to be deployed remotely, enhancing device functionality and security without requiring manual intervention.

7. Sensor Interfaces: GPIO pins and interfaces for connecting sensors and actuators enable the module to interact with the physical world, gathering sensor data and controlling actuators based on predefined logic.

8. Cloud Integration: Integration with cloud platforms (e.g., AWS IoT, Google Cloud IoT, Microsoft Azure IoT) facilitates data storage, analytics, and device management in the cloud.

9. Scalability:IoT-based WiFi modules should support scalability, allowing seamless integration with a growing

number of devices and accommodating increased data traffic as the IoT ecosystem expands.

10. Development Tools and SDKs: Comprehensive development tools, SDKs (Software Development Kits), and documentation help developers create applications and firmware for IoT devices using the WiFi module.

11. Certifications: Compliance with regulatory standards and certifications (e.g., FCC, CE) ensures the module meets quality, safety, and interoperability requirements for deployment in various regions.

Overall, an IoT-based WiFi module serves as a crucial component in building connected IoT solutions, offering reliable wireless connectivity, data processing capabilities, and integration with cloud services for a wide range of applications across industries

LED (Light Emitting Diode):

FIGURE 2.4

Light Emitting Diodes, commonly known as LEDs, are semiconductor devices that emit light when an electric current passes through them. They have become integral components in various electronic applications, owing to their energy efficiency, durability, and versatility. The fundamental principle behind an LED's operation involves the recombination of electrons and electron holes within the semiconductor material, resulting in the release of photons. This process is known as electroluminescence. LEDs come in various colours, including red, green, blue, and white, each corresponding to specific semiconductor materials and energy levels.

PIEZEOELECTRIC SENSOR:

FIGURE 2.5

A piezoelectric sensor is a device that utilizes the piezoelectric effect to measure changes in pressure, acceleration, force, or strain by converting them into an electrical charge. Here's a detailed breakdown of how piezoelectric sensors work and their applications:

How Piezoelectric Sensors Work:

o **Piezoelectric Material**: These sensors are typically made of certain crystals (e.g., quartz), ceramics (e.g., lead zirconatetitanate), or polymers that exhibit the piezoelectric effect. When mechanical stress is applied to these materials, they generate an electric charge proportional to the applied stress.
o **Sensing Mechanism**: When pressure or force is applied to the piezoelectric material, it deforms slightly, causing a displacement of positive and negative charges within the material.
o **Electric Signal Generation**: This displacement creates a potential difference across the material, resulting in the generation of an electric signal. The magnitude of the signal is directly proportional to the applied force or pressure.
o **Signal Processing**: The generated electrical signal can be amplified, conditioned, and processed using electronic circuitry to obtain meaningful data regarding the sensed parameter.

Applications of Piezoelectric Sensors:

o **Industrial Monitoring and Control**: Piezoelectric sensors are widely used in industrial applications for monitoring and controlling parameters such as pressure, force, and acceleration in machinery and equipment.

o **Medical Devices**: They are employed in medical devices for applications like pulse sensing, blood pressure monitoring, and ultrasound imaging.
o **Structural Health Monitoring**: Piezoelectric sensors are utilized for structural health monitoring in buildings, bridges, and aerospace structures to detect and monitor vibrations, stress, and strain.
o **Acoustic Sensors**: These sensors are used in acoustic applications such as microphones, accelerometers for detecting sound vibrations, and even in musical instruments like guitars for pickup systems.
o **Touch Sensors**: Piezoelectric sensors are used in touch-sensitive devices, such as touch screens and touch-sensitive buttons.
o **Energy Harvesting**: They can be employed to harvest energy from mechanical vibrations, such as those found in machinery or vehicles, and convert it into electrical energy for powering small devices or sensors.
o **Gas Sensors**: Piezoelectric sensors can be used to detect certain gases by measuring the change in resonance frequency of a piezoelectric crystal when exposed to gas molecules.

VIBRATION SENSOR:

A vibration sensor, also known as an accelerometer, is a device used to measure vibrations or changes in acceleration. It detects motion or vibration and converts it into an electrical signal that can be analyzed or monitored. Vibration sensors come in various types and configurations, including:

Vibration sensors find applications in various fields, including:

o Structural health monitoring of buildings, bridges, and machinery
o Condition monitoring of rotating equipment such as motors, pumps, and turbines
o Automotive industry for vehicle health monitoring and safety systems
o Aerospace industry for aircraft health monitoring and flight testing

o Consumer electronics for image stabilization in cameras and smartphones
o Seismic monitoring for earthquake detection and early warning systems

These sensors are crucial for detecting abnormal vibrations that could indicate faults or potential failures in mechanical systems, allowing for preventive maintenance and ensuring operational safety and efficiency.

RELAY MODULE:

A relay module is an electronic device that allows you to control high-power electrical devices using low-power signals from a microcontroller, Arduino, or other control circuits. It works by using an electromagnet to mechanically switch circuits on or off.

When selecting a relay module, consider the following details:

1. Voltage and Current Rating: Ensure that the relay module can handle the voltage and current requirements of your application. Exceeding these ratings can damage the relay or your circuit.

2.Type of Relay: There are different types of relays such as electromechanical relays, solid-state relays, and reed relays. Choose the type that best suits your application based on factors like switching speed, isolation, and reliability.

3. Number of Channels: Determine how many devices you need to control simultaneously. Relay modules come in various configurations ranging from single-channel to multi-channel.
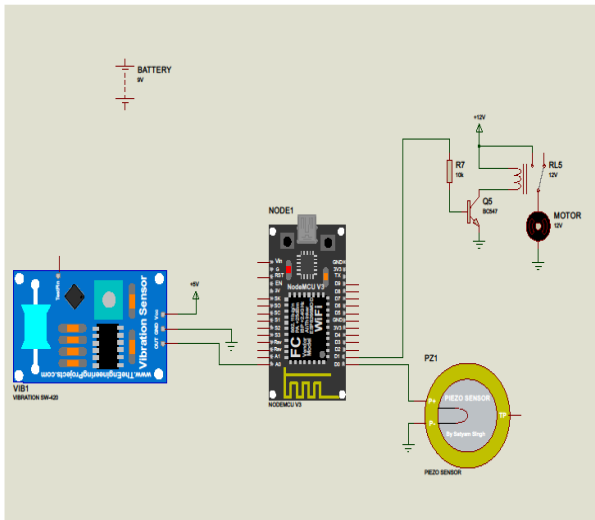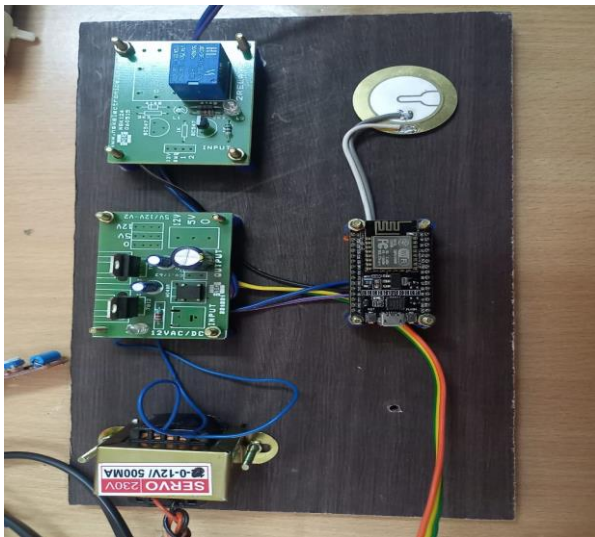
PROTOTYPE:

FIGURE 2.6

TESTING:

```
#define BLYNK_TEMPLATE_ID "TMPL37hk_f89X"
#define BLYNK_TEMPLATE_NAME "Vibrate Sensor Based
Air Pump Control"

#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
#include <ESP8266WiFi.h>

char auth[] = "3GS11uQ1w89rfwpz3AH3wVg51x6a2P09";
const char* ssid = "Esp8266";        // Your Wi-Fi network
SSID
const char* password = "123123123"; // Your Wi-Fi network
password

intvibrate_sensor = A0;
int motor = D1;
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(vibrate_sensor, INPUT);
  pinMode(motor, OUTPUT);

  Serial.println("Connecting to Wi-Fi...");
  WiFi.begin(ssid, password);

  // Wait for the Wi-Fi connection
  while (WiFi.status() != WL_CONNECTED) {
   delay(1000);
    Serial.print(".");
  }
  Blynk.begin(auth, ssid, password);
  digitalWrite(motor, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  int sensor = analogRead(vibrate_sensor);
  Serial.print("sensor");
  Serial.println(sensor);

  if (sensor > 1000) {
    digitalWrite(motor, HIGH);
    Blynk.virtualWrite(V1, "DETECTED");  // Send "Detected"
to V1 in the Blynk app
     Blynk.virtualWrite(V0, "ON");         // Send "Detected" to
V1 in the Blynk app
    delay(5000);
    digitalWrite(motor, LOW);
     Blynk.virtualWrite(V0, "OFF");  // Send "Detected" to V1
in the Blynk app

  } else {
    digitalWrite(motor, LOW);
    Blynk.virtualWrite(V1, "NOT DETECTED");  // Send "Not
Detected" to V1 in the Blynk app
    Blynk.virtualWrite(V0, "OFF");         // Send "Detected" to
V1 in the Blynk app
  }
  delay(500);
  Blynk.run();
  delay(500);
}
```

In summary, your Arduino sketch continuously reads the analogue sensor value from pin A0, checks if it's above a predefined threshold, and turns on the LED for one second if noise is detected. If the environment is quiet, the LED stays off. The serial monitor prints the analogue sensor value for

monitoring and debugging purposes. Adjust the threshold value based on your specific requirements and the characteristics of your sound sensor.

REFRENCES

[1] https://circuitdigest.com/microcontroller-projects/arduino-based-accident-alert-system-using-gps-gsm-acceleromete

[2] https://link.springer.com/chapter/10.1007/978-981-10-8681-6_52

[3] https://www.wikihow.com/Help-a-Victim-of-a-Car-Accident