WALCHAND COLLEGE OF ENGINEERING, SANGLI

Department of Electronics Engineering

PROJECT REPORT

Embedded System and Design Mini Project

T. Y. B. Tech Electronics

FAULT DETECTION SYSTEM

Submitted By	
Name	**PRN**
Viraj Gotmare	22410068
Atharava Pandey	22410069
Suprit Bashetti	22410073
Pushkar Nashikkar	22410074

Under Guidance of **Mr. S.R. Khedkar**

Table of Contents

Abstract

The Fault Detection System using the LPC2138 microcontroller automatically monitors and controls temperature and pressure in safety-critical settings. With two LPC2138 microcontrollers, it continuously checks sensor data, and if limits are exceeded, it automatically sends a warning via UART to a mobile device, shuts down the CNG nozzle, and activates a buzzer. Simulated in Proteus, this fully automated design ensures reliable fault detection and response to prevent hazards.

Introduction

In various industrial and research settings, maintaining controlled temperature and pressure levels is crucial for safety and operational efficiency. Any deviation from these specified parameters can pose risks, such as equipment damage, compromised safety, or system failure. Automated fault detection systems play a vital role in preventing such situations by continuously monitoring environmental parameters and initiating preventive actions.

The *Fault Detection System using LPC2138 Microcontroller* is a compact, embedded solution designed to identify temperature and pressure abnormalities and respond immediately. Utilizing two ARM7-based LPC2138 microcontrollers, this system performs real-time monitoring using temperature and pressure sensors and facilitates communication through UART for remote alerts. The main functions of this project include:

- **Temperature Monitoring and Response:**
    - Primary microcontroller continuously reads temperature sensor data from the CNG cylinder's surface.
    - If temperature exceeds the pre-set threshold:
        - Sends warning message to connected mobile device via UART.
        - Shuts down DC motor to prevent overheating.
        - Activates buzzer to alert on-site personnel.
    - Monitoring cylinder surface temperature is crucial as it affects internal gas pressure.
- **Pressure Monitoring and Control:**
    - Secondary microcontroller monitors pressure levels inside the gas cylinder via pressure sensor.
    - If pressure exceeds safe levels:
        - Closes CNG nozzle to stop gas flow.
        - Prevents over-pressurization for safety.
        - Activates buzzer to alert on-site personnel

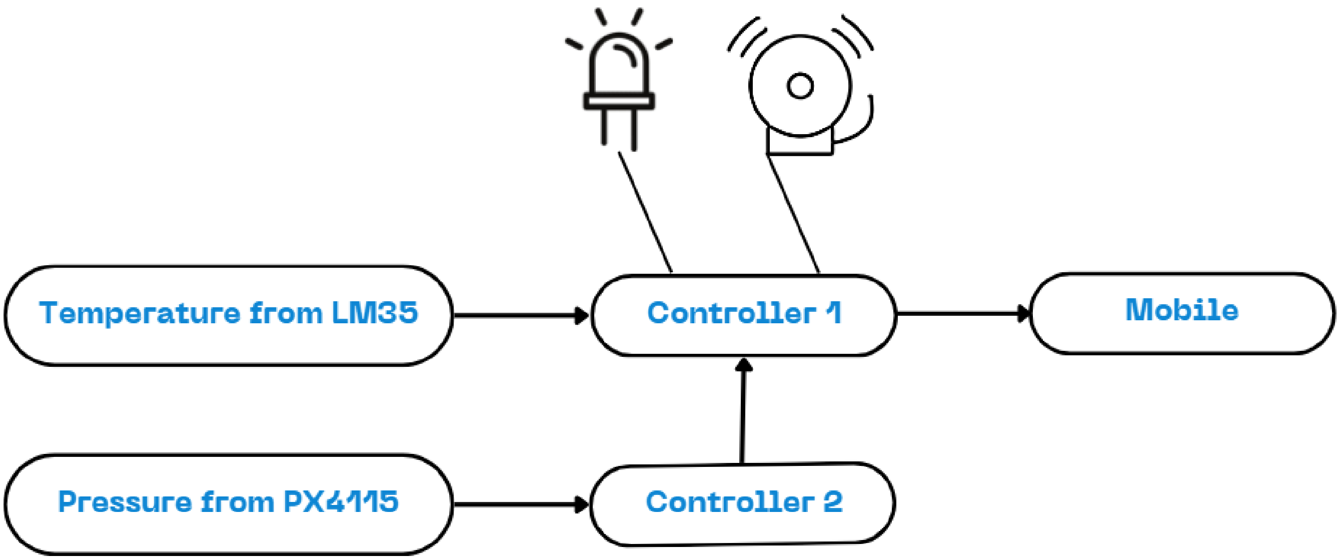
This system's design is implemented and tested in the Proteus simulator, enabling a full digital prototype for debugging, testing, and validation before real-world deployment. Proteus provides a virtual environment for evaluating the functionality and response time of the system, ensuring that the fault detection mechanisms are reliable and accurate.

The project incorporates critical features such as serial communication for alert transmission, real-time analog-to-digital conversion for precise sensor readings, and the flexibility of ARM7 microcontrollers. This fault detection system can be applied to various industrial environments, such as gas pipelines, HVAC systems, and chemical plants, where maintaining specific temperature and pressure thresholds is vital for operational safety. Through efficient design and practical implementation, this project exemplifies an embedded solution to fault detection, ensuring both preventive and protective measures in real-time.

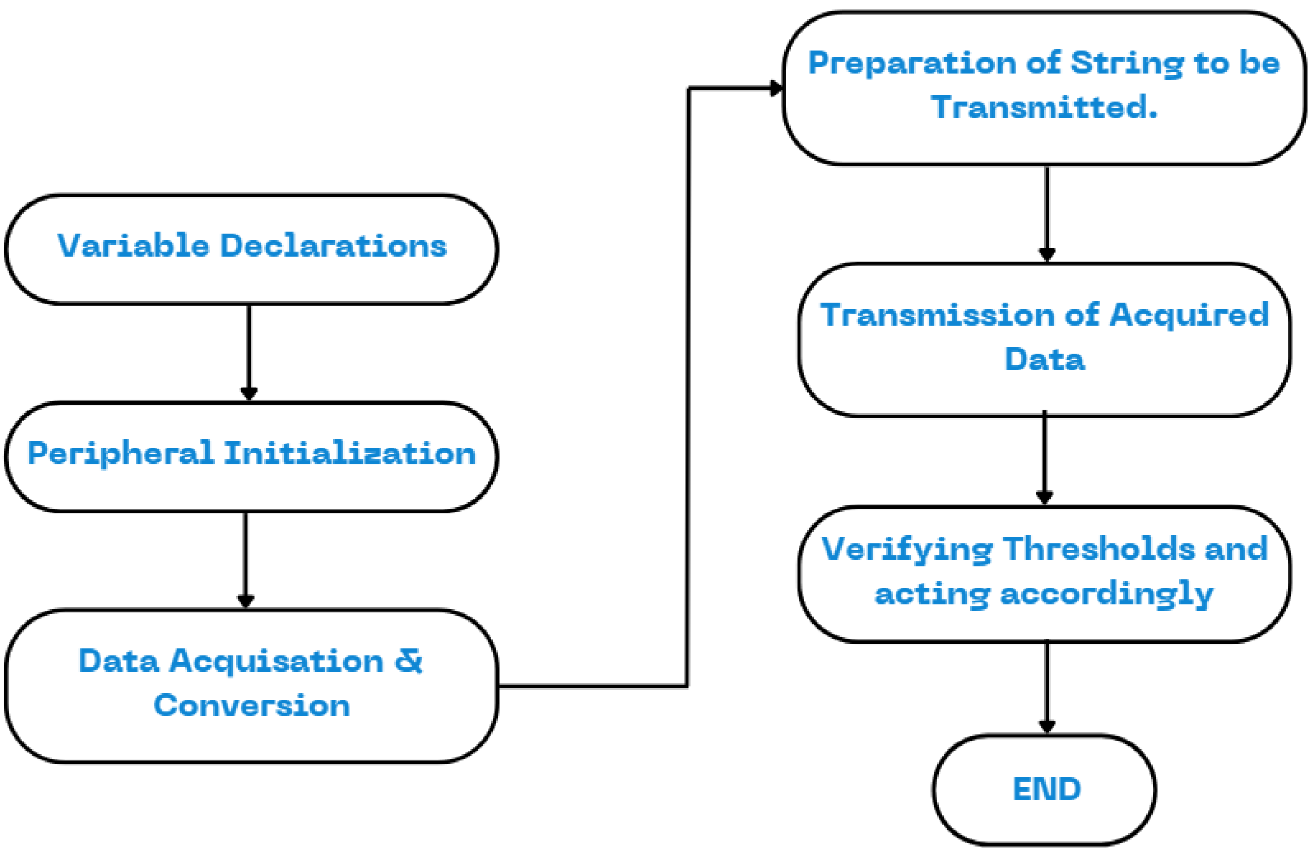
System Overview:

1. **Components used** –
    - **ARM7 controller** – The **LPC2138** is a 32-bit ARM7-based microcontroller with specifications, Flash memory – 512KB, SRAM – 32KB, Clock speed- 60MHz.
    - **HC-05 Bluetooth module** – A Bluetooth module used for wireless serial communication between devices.
    - **LM-35 -** A temperature sensor that provides a linear analog output proportional to the temperature in °C.
    - **MPX4115 -** A pressure sensor that converts atmospheric pressure readings into an analog output.
    - **BC-547** - A general-purpose NPN transistor commonly used for switching and amplification.
    - **Buzzer** - An audio signaling device that produces sound alerts for notifications or warnings.
    - **Diode -** A semiconductor device that allows current to flow in one direction, used for rectification and protection in circuits.
    - **Relay** - An electromechanical switch that allows low-power control of high-power devices, enabling isolation and control of separate electrical circuits.
2. **System Workflow** –
    - The primary microcontroller continuously reads the temperature sensor's data. If the temperature exceeds the set threshold, it closes the CNG nozzle, activates the buzzer, and sends a warning message via UART.
    - The secondary microcontroller monitors pressure through the pressure sensor. If the pressure surpasses the threshold, it immediately closes the CNG nozzle to prevent further gas flow. Both microcontrollers operate independently, ensuring timely response for each fault condition.

Block Diagram :



Flowchart:



Hardware Design

- **LPC2138 Microcontroller (Primary):**
  - Acquires temperature data from the LM35 sensor.
  - Transmits warning notifications to mobile via UART upon exceeding temperature / pressure threshold.
  - Regulates the CNG nozzle state & Buzzer Alarm based on the data acquired.
- **LPC2138 Microcontroller (Secondary):**
  - Monitors pressure data from the MPX4115 sensor.
  - Gives to flag to controller 1 upon exceeding pressure threshold.
- **LM35 Temperature Sensor:**
  - Monitors the surrounding temperature and provides an analog output to the primary microcontroller.
- **MPX4115 Pressure Sensor:**
  - Monitors atmospheric pressure, providing an analog signal to the secondary microcontroller.
- **HC-05 Bluetooth Module:**
  - Facilitates wireless transmission of fault warning notifications to a connected device.
- **Relay:**
  - Functions as a switch to safely control high-power components, such as the DC motor or CNG nozzle.
- **BC547 Transistor:**
  - Functions as a switch to activate the relay for controlling high-power components.
- **Diode:**
  - Protects the circuit from inductive kickback when the relay is deactivated.

Software Design

Software Design for Primary Microcontroller (Temperature Monitoring)

1. **Initialization:**
    - Configure UART0 for serial communication (baud rate: 9600).
    - Initialize ADC for the temperature sensor (LM35).
    - Set I/O pins for controlling the buzzer and CNG nozzle.
2. **Main Loop:**
    - **Read Temperature:**
        - Continuously read the temperature of the cylinder surface from the LM35 sensor using the ADC.
        - Convert the ADC value to temperature in degrees Celsius (°C).
    - **Transmit Temperature Data:**
        - Format the temperature data into a string.
        - Send the temperature data via UART to the mobile device.
    - **Check Temperature Threshold:**
        - If the temperature exceeds the defined threshold (e.g., 45°C):
            - Activate the buzzer to sound an alert.
            - Close the CNG nozzle to prevent overheating.
    - **Receive Command:**
        - Continuously check for any commands received via UART from the mobile device and process them as needed.

Software Design for Secondary Microcontroller (Pressure Monitoring)

1. **Initialization:**
    - Configure UART0 for serial communication (baud rate: 9600).
    - Initialize ADC for the pressure sensor (MPX4115).
    - Set I/O pins for controlling the CNG nozzle.
2. **Main Loop:**
    - **Read Pressure:**
        - Continuously read the pressure from the MPX4115 sensor using the ADC.
        - Convert the ADC value to pressure in kilopascals (kPa).
    - **Transmit Pressure Data:**
        - Format the pressure data into a string.
        - Send the pressure data via UART to the mobile device.
    - **Check Pressure Threshold:**
        - If the pressure exceeds the defined threshold (e.g., 90 kPa):
            - Close the CNG nozzle to prevent further gas flow.

3. Source Code:

Microcontroller 1:

```
#include <LPC213X.H>

void delay(unsigned int t) {
    unsigned int i, j;
for (j = 0; j < t; j++)
for (i = 0; i < 1275; i++);
}

void UART0_Init(void) {
int divisor = 3000000 / (16 * 9600);
PINSEL0 |= (1 << 0) | (1 << 2);
U0LCR = 0x83;
U0DLL = divisor & 0xFF;
U0DLM = (divisor >> 8) & 0xFF;
U0LCR = 0x03;
U0FCR = 0x07;
}

Void UART0_SendChar(char ch) {
while (!(U0LSR & (1 << 5)));
U0THR = ch;
}

Void UART0_SendString(char *str)
{
while (*str)
        UART0_SendChar(*str++);
}

Char UART0_ReceiveChar() {
while (!(U0LSR & 0x01));
return U0RBR;
}

void UART0_ReceiveString(char *buffer) {
char ch;float result = 0.0;
float decimalPlace = 0.1;
int isDecimal = 0;
int i = 0;
int j = 0;

while (1) {
        ch = UART0_ReceiveChar();
if (ch == '\n' || ch == '\r')
{
buffer[i] = '\0';
break;
}
else
buffer[i++] = ch;
}
}

void itstr(int num, char *str)
{
    int i = 0, temp, sign = 0, j;
    if (num == 0)
{
str[i++] = '0';
str[i] = '\0';
return;
}
if (num < 0)
{
sign = 1;
num = -num;
}
while (num > 0)
{
temp = num % 10;
str[i++] = temp + '0';
```

```
num /= 10;
}
if (sign)
str[i++] = '-';
str[i] = '\0';
for (j = 0; j < i / 2; j++)
{
char temp = str[j];
str[j] = str[i - j - 1];
str[i - j - 1] = temp;
}
}
void ADC0_Init()
{
PINSEL1 = (1 << 24);
}
unsigned int ADC0_Read() {
unsigned int data;
AD0CR = (1 << 21) | (3 << 9) | (1 << 1);
AD0CR |= (1 << 24);
while ((AD0GDR & (1 << 31)) == 0);
data = AD0GDR;
data = (data >> 6) & 0x03FF;
return data;
}
void ADC1_init()
    {
        PINSEL0 = 0x00030000;
    }
unsigned int ADC1_Read() {
unsigned int data;
AD1CR = (1 << 21) | (3 << 9) | (1 << 1);
AD1CR |= (1 << 24);
while ((AD1GDR & (1 << 31)) == 0);
data = AD1GDR;
data = (data >> 6) & 0x03FF;
return data;
}
int main()
{
char temp_str[10], buffer[100], buff[100];
unsigned int val;
int temp;
char prefix[100] = "uC1 - Temperature: ";
int i, j;
IODIR0 = (1 << 4)|(1<<15);
IODIR1 |= (1 << 16);
UART0_Init();
ADC0_Init();
while (1)
{
val = ADC0_Read();
temp = (val * 3.3 / 1023) * 100;
itstr(temp, temp_str);
i = j = 0;
while (prefix[j] != '\0') {
buffer[i++] = prefix[j++];
}
j = 0;
while (temp_str[j] != '\0') {
buffer[i++] = temp_str[j++];
}
buffer[i++] = '\r';
buffer[i++] = '\n';
buffer[i] = '\0';
UART0_SendString(buffer);
UART0_ReceiveString(buff);
UART0_SendString(buff);
UART0_SendChar('\n');
if (temp > 45  || (IOPIN0>>3 & 0x01))
IOSET0 = (1 << 4)|(1<<15);
else
IOCLR0 = (1 << 4)|(1<<15);
delay(100);
}
}
```

Microcontroller 2:

```
#include <LPC213X.H>

void delay(unsigned int t) {
unsigned int i, j;
for (j = 0; j < t; j++) {
        for (i = 0; i < 1275; i++);
}
}

void UART0_Init(void) {
int divisor = 3000000 / (16 * 9600);
PINSEL0 |= (1 << 0) | (1 << 2);
U0LCR = 0x83;
U0DLL = divisor & 0xFF;
U0DLM = (divisor >> 8) & 0xFF;
U0LCR = 0x03;
U0FCR = 0x07;
}

void UART0_SendChar(char ch) {
while (!(U0LSR & (1 << 5)));
U0THR = ch;
}

    void UART0_SendString(char *str) {
    while (*str) {
            UART0_SendChar(*str++);
    }
    }

    void ftostr(float num, char *str) {
    int integer_part = (int)num;
    int decimal_part = (int)((num - integer_part) * 100);
    int i = 0, j;
    if (integer_part == 0) {
            str[i++] = '0';
    } else {
int temp = integer_part, digits = 0;
char pre_str[10];
while (temp) {
pre_str[digits++] = (temp % 10) + '0';
temp /= 10;
}
for (j = digits - 1; j >= 0; j--) {
str[i++] = pre_str[j];
}
}
str[i++] = '.';
str[i++] = (decimal_part / 10) + '0';
str[i++] = (decimal_part % 10) + '0';
str[i] = '\0';
}

void ADC0_Init() {
PINSEL1 = (1 << 24);
}

unsigned int ADC0_Read() {
unsigned int data;
AD0CR = (1 << 21) | (3 << 9) | (1 << 1);
AD0CR |= (1 << 24);
while ((AD0GDR & (1 << 31)) == 0);
data = AD0GDR;
data = (data >> 6) & 0x03FF;
return data;
}

int main() {
    float pressure, tmp;
    char pre_str[10];
    char buffer[100];
    int i, j;
```

```
char prefix[] = "uC2 - Pressure: ";
UART0_Init();
ADC0_Init();
IODIR0 = (1<<4);
while (1) {
unsigned int adc_val = ADC0_Read();
pressure = ((float)adc_val / 1023.0 + 0.095) / 0.009;
pressure -= 1.46;
ftostr(pressure, pre_str);
if(pressure > 90)
IOSET0 = (1<<4);
else
IOCLR0 = (1<<4);
i = j = 0;
while (prefix[j] != '\0') {
buffer[i++] = prefix[j++];
}
j = 0;
while (pre_str[j] != '\0') {
buffer[i++] = pre_str[j++];
}
buffer[i++] = 'k';
buffer[i++] = 'P';
buffer[i++] = 'a';
buffer[i++] = '\n';
buffer[i] = '\0';
UART0_SendString(buffer);
delay(200);
}
}
```

Security Features:

Threshold-Based Fault Detection:

The system continuously monitors temperature and pressure and activates alarms or triggers actions (buzzer, motor, nozzle shut-off) whenever any sensor reading exceeds the pre-set threshold. This provides early detection of hazardous conditions, minimizing the risk of accidents or explosions.

Redundant Fault Detection System:

The use of two independent sensors (temperature and pressure) ensures redundancy, meaning the system can still function if one sensor fails. Both sensors work in tandem to monitor the cylinder's environment, and either one can trigger a shutdown in case of a fault.

Automatic Shutdown of CNG Nozzle:

The system is programmed to automatically shut down the CNG nozzle when critical thresholds are crossed, helping to control the release of gas and reduce pressure buildup. This feature helps contain any possible hazard without relying on human intervention.

Audible Alarm (Buzzer Activation):

The buzzer immediately alerts nearby personnel to dangerous conditions, providing a secondary layer of security through real-time alerts. This allows nearby operators to take additional safety actions or evacuate if necessary.

Communication with External System:

Through serial communication, warning messages can be transmitted to a connected mobile or monitoring device. This real-time data transmission enables remote monitoring, potentially allowing for centralized control or early alerting systems.

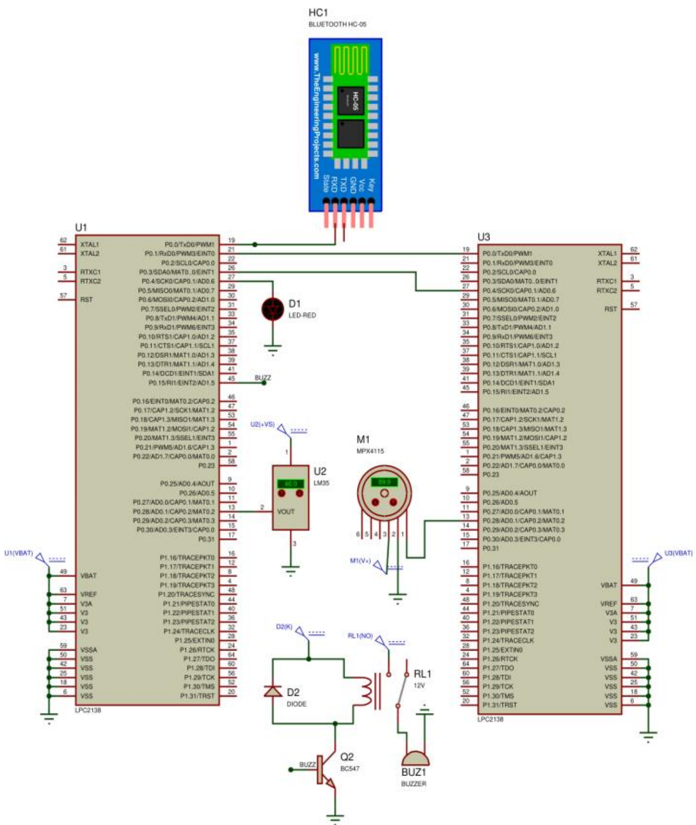
Power Isolation Using Relays:

The relay module isolates high-power components, protecting the microcontroller and other low-voltage circuits from voltage surges or short circuits, thus enhancing overall system stability and safety.

Prevention of Manual Override:

The automatic operation design limits manual intervention, preventing human error and unauthorized access. Only under safe conditions can the CNG nozzle be reopened or the system reset.

Proteus Design :



Applications:

1. **Automotive CNG Safety Systems**:

   Use Case: Embedded in CNG-powered vehicles, this system can monitor in-cylinder pressure and surrounding temperatures to provide warnings or automatic shutdowns if pressure or temperature reaches unsafe levels.

   Application: Real-time alerts in vehicles during refueling or operation, protecting against potential cylinder malfunctions or leaks.

2. **Gas Refueling Station Safety System**:

   Use Case: In CNG and LPG refueling stations, the system could monitor cylinders and the surrounding environment for excessive pressure or high ambient temperatures, alerting personnel or automatically stopping the refueling process if a critical threshold is crossed.

   Application: Prevents over-pressurization or overheating incidents that could lead to explosions, particularly in busy, high-turnover areas.

3. **Household LPG Cylinder Safety Monitor**:

   Use Case: For households, the system can be adapted to detect dangerous temperature increases around the LPG cylinder, predicting potential leaks or pressure build-up and alerting residents via alarms or mobile notifications.

Application: Predictive alerts for households to prevent gas leak incidents, enabling timely evacuation or corrective action if dangerous conditions are detected.

4. **Industrial Gas Storage Facilities**:

   Use Case: In large industrial settings where CNG or LPG is stored in bulk, this system can monitor individual storage tanks and warn operators of any high-pressure situations or environmental temperature rises that could lead to hazardous incidents.

   Application: Allows centralized monitoring and alerting across numerous tanks, enabling preventive maintenance and reducing the likelihood of accidents in industrial storage facilities.

5. **Transport and Logistics for Gas Cylinders**:

   Use Case: During transport, particularly for hazardous materials like CNG or LPG, the system can be used in trucks to monitor for any temperature or pressure fluctuations within or around the gas cylinders.

   Application: Enhances safety during transit by providing real-time monitoring of the cylinders, alerting drivers to potential issues like pressure surges or ambient temperature increases, thereby reducing transportation risks.

Advantages:

1. **Direct Cylinder Surface Monitoring**:

   By measuring temperature directly on the cylinder's surface, the system captures critical internal temperature changes more accurately, offering a precise early warning of potential issues before they escalate.

2. **Automated Safety Measures:**

   The system autonomously shuts off the CNG nozzle and triggers an audible alarm if any sensor threshold is crossed, reducing reliance on human response in dangerous situations and increasing operational safety.

3. **Enhanced Reliability with Dual-Sensor Setup:**

   Using both temperature and pressure sensors ensures redundancy. If either sensor detects an unsafe level, the system activates, ensuring reliability in identifying risks due to overheating or pressure buildup.

Disadvantages:

1. **Calibration and Maintenance Needs:**

Regular calibration of the temperature and pressure sensors is essential for maintaining accuracy over time. Poor maintenance could lead to inaccurate readings, which may result in missed or false alerts.

2. **Potential for False Positives:**

   Surface temperature readings could be influenced by environmental factors (like ambient heat) that do not necessarily indicate a fault within the cylinder, potentially leading to unnecessary shutdowns or alarms.

3. **Single-Point Measurement Limitations:**

   Monitoring only the cylinder's surface temperature may limit the system's responsiveness to external factors impacting the entire storage area, like a nearby fire. For broader safety coverage, additional environmental monitoring may still be needed.

- P0.0 is configured in its 2^{nd} functionality which is TxD(transmission of data). This is connected to the RxD pin of HC-05.
- P0.1 is configured in its 2^{nd} functionality which is RxD(reception of data). This is connected to the TxD(P0.1) pin of secondary microcontroller. This pin receives the serial data of the pressure sensor from the secondary microcontroller.
- P0.3 is shorted to P0.4 of secondary controller.
- P0.4 is connected to an led which can be replaced by a servo motor to operate the CNG nozzle.
- P0.15 is connected to a relay through a wire label which is further connected to the buzzer.
- P0.28 is configured in its 2^{nd} functionality which is ADC0.1(ADC 0 capture 1) to read the analog data of temperature sensor. ADC will convert the analog data to its equivalent digital format.

- P0.0 of secondary controller is connected to the P0.1 of primary controller for transmitting serial data of pressure sensor.
- P0.4 is shorted to P0.3 of primary controller. This is flag bit which will be set if the pressure sensed by pressure sensor crosses the threshold (90kPa for CNG cylinders). This flag bit will be an indication to regulate the state of nozzle(open/close).
- P0.28 is configured in its 2^{nd} functionality which is ADC0.1(ADC 0 capture 1) to read the analog data from pressure sensor. ADC will convert the analog data to its equivalent digital format.

Conclusion

The Fault Detection System is a comprehensive safety solution for CNG storage and handling environments, designed to monitor and respond to dangerous temperature and pressure levels. By integrating both temperature and pressure sensors, this system provides dual-layered safety, where any detected anomaly—such as excessive heat on the cylinder surface or unsafe pressure within the tank—triggers an immediate safety response. This response includes shutting off the CNG nozzle and activating an alarm, effectively reducing the risk of accidents by allowing early intervention.

A notable strength of the system is its ability to detect issues directly from the cylinder's surface, ensuring that internal risks like overheating or pressure increases are identified quickly and accurately. This approach enhances safety, especially in environments where compressed gases are stored and handled regularly. Additionally, the system’s automatic responses lessen reliance on manual monitoring, making it an ideal safety mechanism for busy refueling stations, CNG-powered vehicles, and industrial storage facilities.

Key System Elements:

- **Direct Cylinder Surface Monitoring**: Temperature is measured directly on the cylinder surface for accurate internal risk assessment.
- **Dual-Sensor Approach**: Combines temperature and pressure sensors, offering redundancy and ensuring alerts for any threshold breach.
- **Automated Response**: Includes automatic shutdown of the CNG nozzle and activation of an alarm, removing reliance on manual intervention in emergencies.
- **Microcontroller Control**: Each sensor operates with a dedicated microcontroller to streamline processing and control for rapid response.

Key Outcomes:

- **Enhanced Safety**: Provides early detection of potential hazards, allowing proactive intervention to prevent incidents.
- **High Reliability**: The redundancy of dual sensors improves the system’s reliability in identifying overheating or pressure build-up.
- **Versatile Applications**: Suitable for a variety of CNG storage and handling settings—such as refueling stations, automotive safety, and industrial gas storage.
- **Operational Efficiency**: Automated actions help reduce human error, simplify monitoring, and offer peace of mind in high-risk environments.

Considerations for Effective Use:

- **Regular Calibration and Maintenance**: Essential to maintain sensor accuracy and prevent false alerts.
- **Potential for Expansion**: Additional sensors could broaden safety monitoring to include surrounding environmental risks, such as fires nearby.