**FDAS PROJECT**

**Summary**

This circuit is a multi-sensor and communication system built around the STM32F103C8T6 microcontroller. It integrates various sensors, a display, and communication modules to monitor environmental conditions and transmit data. The system includes temperature and humidity sensors, a gas sensor, a flame sensor, a GPS module, and a GSM module for communication. The data is displayed on an LCD and can be transmitted via the SIM800L module.

**Component List**

1. **STM32F103C8T6 Microcontroller**

* A 32-bit ARM Cortex-M3 microcontroller used as the central processing unit for the circuit.

1. **DHT22 Sensor**

* A digital sensor for measuring temperature and humidity.

1. **LCD 20x4 I2C Display**

* A 20x4 character LCD with I2C interface for displaying data.

1. **GPS NEO 6M Module**

* A GPS module for obtaining location data.

1. **LM35DT Temperature Sensor**

* An analog temperature sensor providing a voltage output proportional to the temperature.

1. **SIM800L GSM Module**

* A GSM module for sending and receiving SMS and data.

1. **MQ-2 Gas Sensor**

* A sensor for detecting gas levels, particularly methane and LPG.

1. **INA219 Current Sensor**

* A sensor for measuring current and voltage.

1. **Flame Sensor**

* A sensor for detecting the presence of flame or fire.

1. **Li-ion Battery 18650-2000mAh 3.7V**

* A rechargeable battery providing power to the circuit.

1. **Buzzer**

* An audio signaling device used for alerts.

**Wiring Details**

**STM32F103C8T6 Microcontroller**

* **3.3V**: Connected to the Vs pin of the LM35DT, VCC pin of the GPS NEO 6M, VCC pin of the INA219, and Vcc pin of the Flame Sensor.
* **5V**: Connected to the 5v pin of the LCD 20x4 I2C and VCC pin of the MQ-2.
* **GND**: Common ground connected to multiple components including the LCD, LM35DT, GPS NEO 6M, SIM800L, INA219, Flame Sensor, DHT22, and Buzzer.
* **A0**: Connected to the Out pin of the DHT22.
* **A1**: Connected to the OUT pin of the LM35DT.
* **A2**: Connected to the RX pin of the GPS NEO 6M.
* **A3**: Connected to the TX pin of the GPS NEO 6M.
* **A4**: Connected to the D0 pin of the Flame Sensor.
* **A5**: Connected to the VCC pin of the Buzzer.
* **A6**: Connected to the Digital pin of the MQ-2.
* **A9**: Connected to the RXD pin of the SIM800L.
* **A10**: Connected to the TXD pin of the SIM800L.
* **B6**: Connected to the SCL pin of the LCD 20x4 I2C.
* **B7**: Connected to the SCA pin of the LCD 20x4 I2C.
* **B10**: Connected to the SCL pin of the INA219.
* **B11**: Connected to the SDA pin of the INA219.

**DHT22 Sensor**

* **+**: Connected to the 3.3V pin of the STM32F103C8T6.
* **Out**: Connected to the A0 pin of the STM32F103C8T6.
* **-**: Connected to the common ground.

**LCD 20x4 I2C Display**

* **5v**: Connected to the 5V pin of the STM32F103C8T6.
* **GND**: Connected to the common ground.
* **SCA**: Connected to the B7 pin of the STM32F103C8T6.
* **SCL**: Connected to the B6 pin of the STM32F103C8T6.

**GPS NEO 6M Module**

* **VCC**: Connected to the 3.3V pin of the STM32F103C8T6.
* **RX**: Connected to the A2 pin of the STM32F103C8T6.
* **TX**: Connected to the A3 pin of the STM32F103C8T6.
* **GND**: Connected to the common ground.

**LM35DT Temperature Sensor**

* **Vs**: Connected to the 3.3V pin of the STM32F103C8T6.
* **OUT**: Connected to the A1 pin of the STM32F103C8T6.
* **GND**: Connected to the common ground.

**SIM800L GSM Module**

* **VCC**: Connected to the + pin of the Li-ion battery.
* **RXD**: Connected to the A9 pin of the STM32F103C8T6.
* **TXD**: Connected to the A10 pin of the STM32F103C8T6.
* **GND**: Connected to the common ground.

**MQ-2 Gas Sensor**

* **VCC**: Connected to the 5V pin of the STM32F103C8T6.
* **Digital**: Connected to the A6 pin of the STM32F103C8T6.
* **GND**: Connected to the common ground.

**INA219 Current Sensor**

* **VCC**: Connected to the 3.3V pin of the STM32F103C8T6.
* **GND**: Connected to the common ground.
* **SCL**: Connected to the B10 pin of the STM32F103C8T6.
* **SDA**: Connected to the B11 pin of the STM32F103C8T6.

**Flame Sensor**

* **Vcc**: Connected to the 3.3V pin of the STM32F103C8T6.
* **D0**: Connected to the A4 pin of the STM32F103C8T6.
* **G**: Connected to the common ground.

**Li-ion Battery 18650-2000mAh 3.7V**

* **+**: Connected to the VCC pin of the SIM800L.
* **-**: Connected to the common ground.

**Buzzer**

* **VCC**: Connected to the A5 pin of the STM32F103C8T6.
* **GND**: Connected to the common ground.

**Code Documentation**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

#include <TinyGPS++.h>

// Pin Definitions

#define LM35\_PIN PA1 // Analog pin connected to LM35

#define DHTPIN PA0 // Digital pin connected to DHT22

#define DHTTYPE DHT22 // Define DHT22 sensor type

#define MQ2\_DIGITAL\_PIN PA6 // Digital pin connected to MQ-2 Smoke Sensor DO (Digital Output)

#define FLAME\_SENSOR\_PIN PA4 // Digital pin connected to Flame Sensor

#define BUZZER\_PIN PA5 // Digital pin connected to Buzzer

// Initialize objects

DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal\_I2C lcd(0x27, 20, 4); // LCD with I2C address 0x27 (20x4)

#define SIM800 Serial1 // SIM800 on Serial1 (PA9 = TX, PA10 = RX)

//#define GPS\_SERIAL Serial2 // GPS on Serial2 (PA2 = TX, PA3 = RX)

//#define FLAME\_SENSOR\_PIN PA4

HardwareSerial GPS\_SERIAL(PA3, PA2); // RX, TX for GPS (connect GPS TX to PA3)

LiquidCrystal\_I2C lcd(0x27, 20, 4); // LCD at I2C address 0x27

TinyGPSPlus gps;

void sendATCommand(String command);

void sendSMS(String message);

void setup() {

SIM800.begin(9600); // For SIM800

GPS\_SERIAL.begin(9600); // For GPS

Serial.begin(9600); // Start Serial Communication

dht.begin(); // Initialize DHT sensor

lcd.init(); // Initialize LCD

lcd.backlight(); // Turn on the backlight

sendATCommand("AT");

delay(2000);

sendATCommand("AT+CMGF=1"); // SMS text mode

delay(2000);

sendATCommand("AT+CSCS=\"GSM\""); // GSM charset

delay(2000);

Serial.println("Ready to detect");

// Entry Display

lcd.setCursor(4, 1);

lcd.print("FDAS SYSTEM");

delay(3000); // Display for 3 seconds

lcd.clear();

pinMode(MQ2\_DIGITAL\_PIN, INPUT); // Set the MQ-2 digital output pin as input

pinMode(FLAME\_SENSOR\_PIN, INPUT); // Set the flame sensor digital output pin as input

pinMode(BUZZER\_PIN, OUTPUT); // Set the buzzer pin as output

}

void loop() {

// Continuously read GPS data

while (GPS\_SERIAL.available()) {

gps.encode(GPS\_SERIAL.read());

}

// Read LM35 analog value

int adcLM35 = analogRead(LM35\_PIN);

// Convert ADC value to voltage for LM35

float voltageLM35 = ((adcLM35 \* 3.3) / 4095.0) \* 2;

// Convert voltage to temperature (LM35: 10mV/°C)

float lm35Temp = voltageLM35 \* 100.0;

// Read DHT22 temperature

float dhtTemp = dht.readTemperature(); // Celsius

// Check if DHT22 readings are valid

if (isnan(dhtTemp)) {

Serial.println("Failed to read from DHT sensor!");

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("DHT Read Error!");

delay(2000);

return;

}

// Read MQ-2 digital output value (HIGH or LOW)

int smokeDetected = digitalRead(MQ2\_DIGITAL\_PIN);

// Read Flame Sensor digital output value (HIGH or LOW)

int flameState = digitalRead(FLAME\_SENSOR\_PIN);

// Debugging Output

Serial.print("LM35 Temp: "); Serial.print(lm35Temp, 2);

Serial.print(" °C | DHT22 Temp: "); Serial.print(dhtTemp, 2);

Serial.print(" °C | Smoke Detected: ");

if (smokeDetected == HIGH) {

Serial.println("Yes");

} else {

Serial.println("No");

}

// Display on LCD

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("LM35 Temp: ");

lcd.print(lm35Temp, 2);

lcd.print((char)223); // Degree symbol

lcd.print("C");

lcd.setCursor(0, 1);

lcd.print("DHT22 Temp: ");

lcd.print(dhtTemp, 2);

lcd.print((char)223); // Degree symbol

lcd.print("C");

// Check if smoke is detected

lcd.setCursor(0, 2);

if (smokeDetected == HIGH) {

lcd.print("NO Smoke");

} else {

lcd.print("Smoke Detected!");

}

// Check for flame detection and trigger buzzer

if (flameState == LOW) { // Flame detected

digitalWrite(BUZZER\_PIN, HIGH);

delay(3000);

digitalWrite(BUZZER\_PIN, LOW);

String smsMessage = "FLAME DETECTED\n";

if (gps.location.isValid() && gps.location.age() < 10000) {

smsMessage += "Lat: ";

smsMessage += String(gps.location.lat(), 6);

smsMessage += "\nLng: ";

smsMessage += String(gps.location.lng(), 6);

} else {

smsMessage += "GPS ERROR";

}

sendSMS(smsMessage);

delay(10000); // Wait 10 seconds to avoid flooding

}

delay(500);

}

void sendATCommand(String command) {

SIM800.println(command);

delay(1000);

Serial.println("Command: " + command);

String response = "";

while (SIM800.available()) {

response += (char)SIM800.read();

}

Serial.println("Modem Response: " + response);

}

void sendSMS(String message) {

Serial.println("Sending SMS...");

SIM800.println("AT+CMGS=\"+919008714862\""); // Replace with your number

delay(2000);

SIM800.print(message);

delay(500);

SIM800.write(26); // CTRL+Z

delay(5000);

Serial.println("SMS Sent: " + message);

}

**Circuit Diagram:**

