TEMPERATURE AND SMOKE DETECTOR BASED LOAD SWITCHING PROJECT AND DATA SHARING AND IMPLEMENTATION DETAILS

Submitted By
Md. Hasemi Rafsan Jani Shohan
Embedded System Developer
BSc in EEE
hasemi33-849@diu.edu.bd
01784032597

Submitted To
SARBS Communication Ltd
Embedded System Section

INTRODUCTION:

This project is designed to monitor the environmental conditions of a space using an Arduino microcontroller and three different sensor modules - a smoke sensor, a temperature sensor, and a DC voltage sensor. In addition to this, it also includes four channel relay control functionality.

The project includes an external set/reset function that allows the user to select specific threshold values for each of the sensors. This is done through a value selecting process that is controlled by the Arduino. The user can input specific values for each sensor using a set of buttons or switches.

The four channel relay control function allows the user to control various devices or equipment based on the sensor readings. For example, if the smoke sensor detects smoke, the Arduino can activate a relay that turns on a fan or an alarm system. Similarly, if the temperature sensor detects a high temperature, the Arduino can activate a relay that turns on an air conditioner or a heater.

The project also includes two power LED indicators that provide visual feedback on the status of the sensors. The LEDs will light up when the sensors detect smoke or a high temperature, or when the voltage of the power source is outside of the selected range.

Overall, this project provides a flexible and customizable solution for monitoring environmental conditions using an Arduino microcontroller, a set of sensors, and four channel relay control functionality.

EQUIPMENT:

1. Arduino UNO
2. Microcontroller
3. I^{2}CDriver
4. LCD Display
5. Voltage Sensor
6. Relay
7. LED Bulb
8. Push Button Switch
9. Connecting wires
10. Breadboard
11. Temperature Sensor
12. Smoke Sensor

Picture1: Arduino UNO

Picture2: Microcontroller

Picture3: LCD Display & Driver

Picture4: DC Voltage Sensor

Picture5: Relay

Picture6: LED Bulb

Picture7: Push Button Switch

Picture8: Connecting wires

Picture9: Breadboard

Picture10: Smoke or Gas sensor

Picture11: Temperature and Humidity sensor

Component Details

WORKING PRINCIPLE

The Arduino specific value limit range-based project with smoke sensor, temperature sensor, DC voltage sensor, external set reset, and value selecting process with two power LED indicators and four channel relay control works based on the following principles:

1. Smoke Sensor: The smoke sensor is used to detect the presence of smoke in the environment. It works on the principle of light scattering. When smoke enters the chamber of the sensor, it scatters the light beam, which is detected by a photodiode. The Arduino reads the sensor data and triggers the relay if the smoke level exceeds the set threshold value.

2. Temperature Sensor: The temperature sensor is used to measure the ambient temperature. It works on the principle of thermistor. The thermistor changes its resistance value based on the temperature. The Arduino reads the resistance value and converts it to the temperature value. The relay is triggered if the temperature exceeds the set threshold value.

3. DC Voltage Sensor: The DC voltage sensor is used to measure the voltage of a power source. It works on the principle of voltage divider. The voltage is divided between two resistors, and the voltage across one resistor is measured by the Arduino. The Arduino then converts the voltage value to the actual voltage level. The relay is triggered if the voltage is outside of the set range.

4. External Set/Reset and Value Selecting Process: The external set/reset function allows the user to select specific threshold values for each of the sensors. This is done through a value selecting process that is controlled by the Arduino. The user can input specific values for each sensor using a set of buttons or switches.

5. Two Power LED Indicators: The project also includes two power LED indicators that provide visual feedback on the status of the sensors. The LEDs will light up when the sensors detect smoke or a high temperature, or when the voltage of the power source is outside of the selected range.

6. Four Channel Relay Control: The four channel relay control function allows the user to control various devices or equipment based on the sensor readings. For example, if the smoke sensor detects smoke, the Arduino can activate a relay that turns on a fan or an alarm system. Similarly, if the temperature sensor detects a high temperature, the Arduino can activate a relay that turns on an air conditioner or a heater.

Circuit Diagram:

Download link

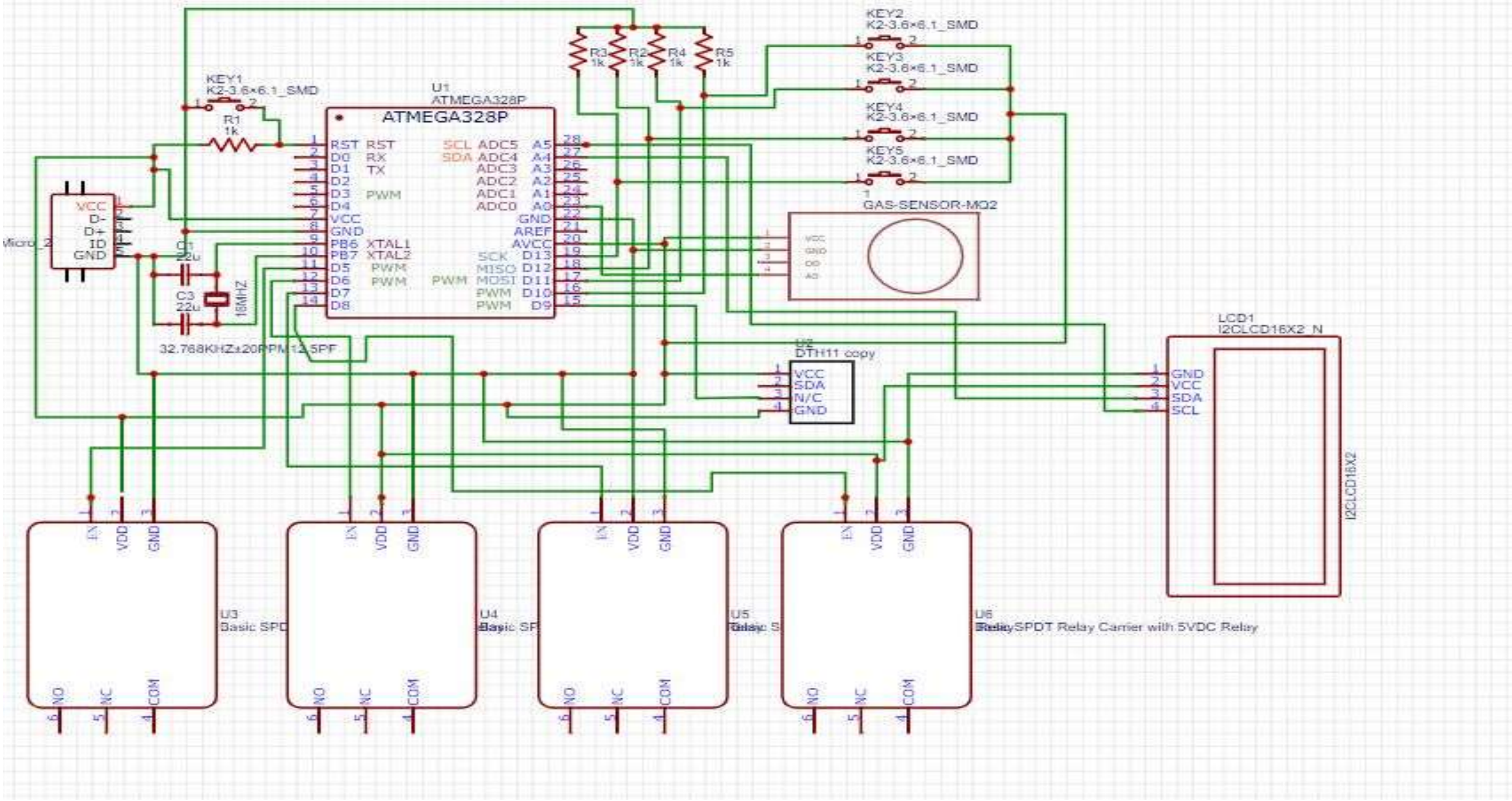


Fig1: Circuit Diagram Of This Project

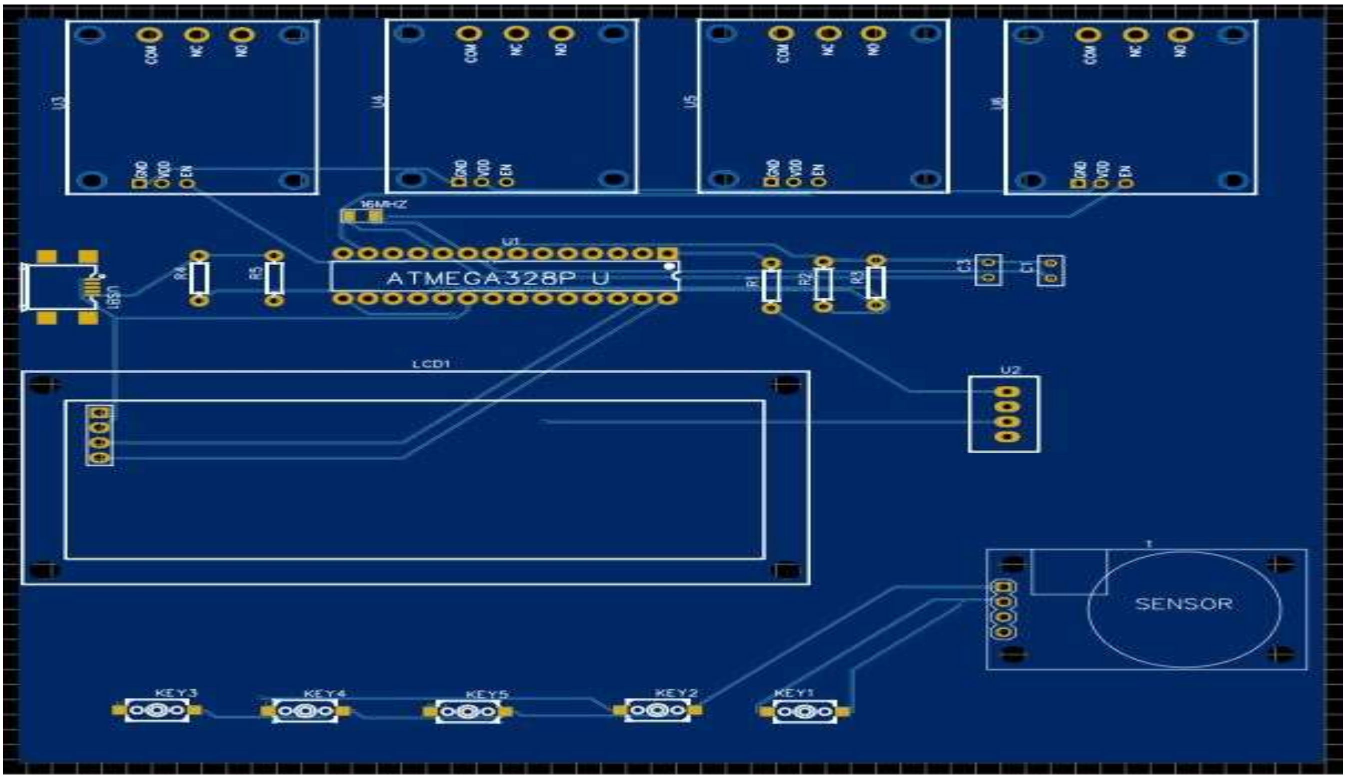


Fig2: PCB 2D View Of This Project

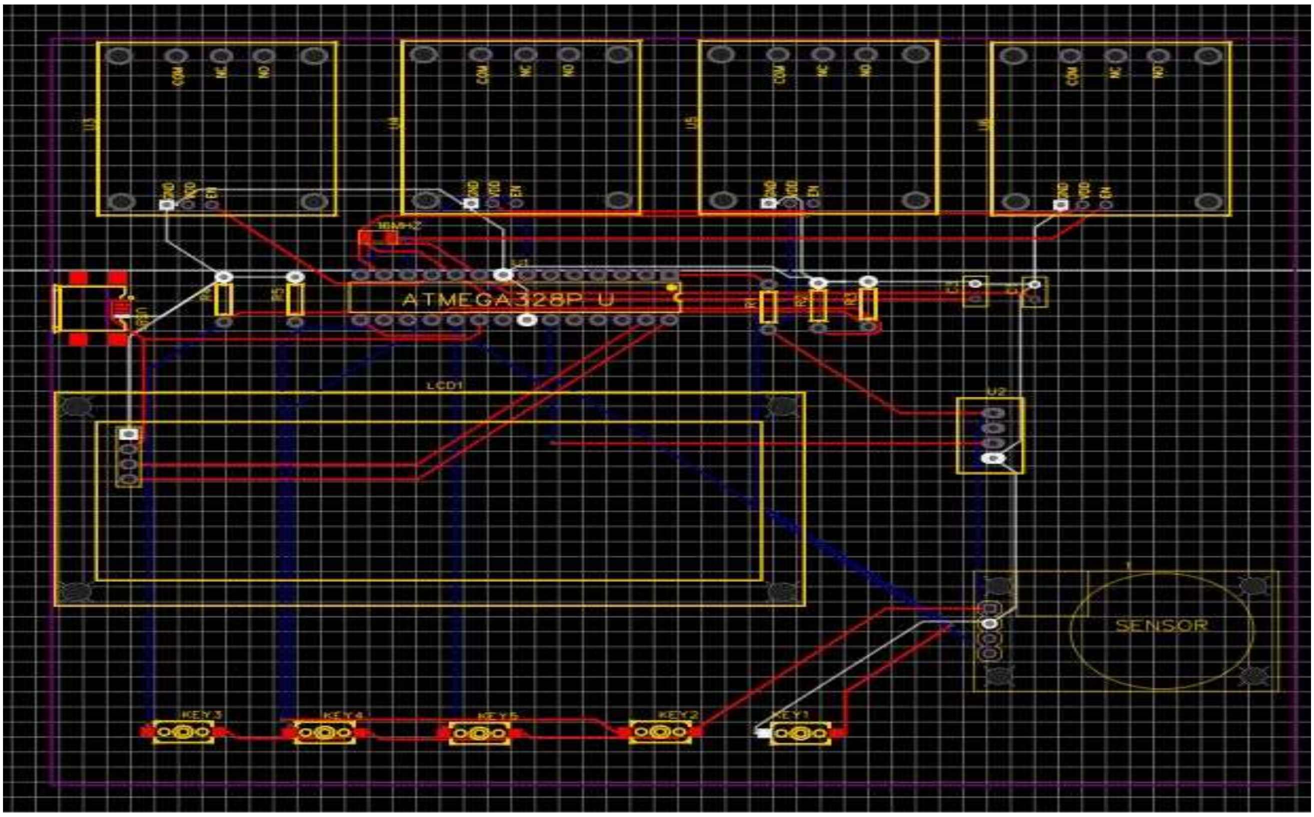


Fig3: PCB Schematic Diagram Of This Project

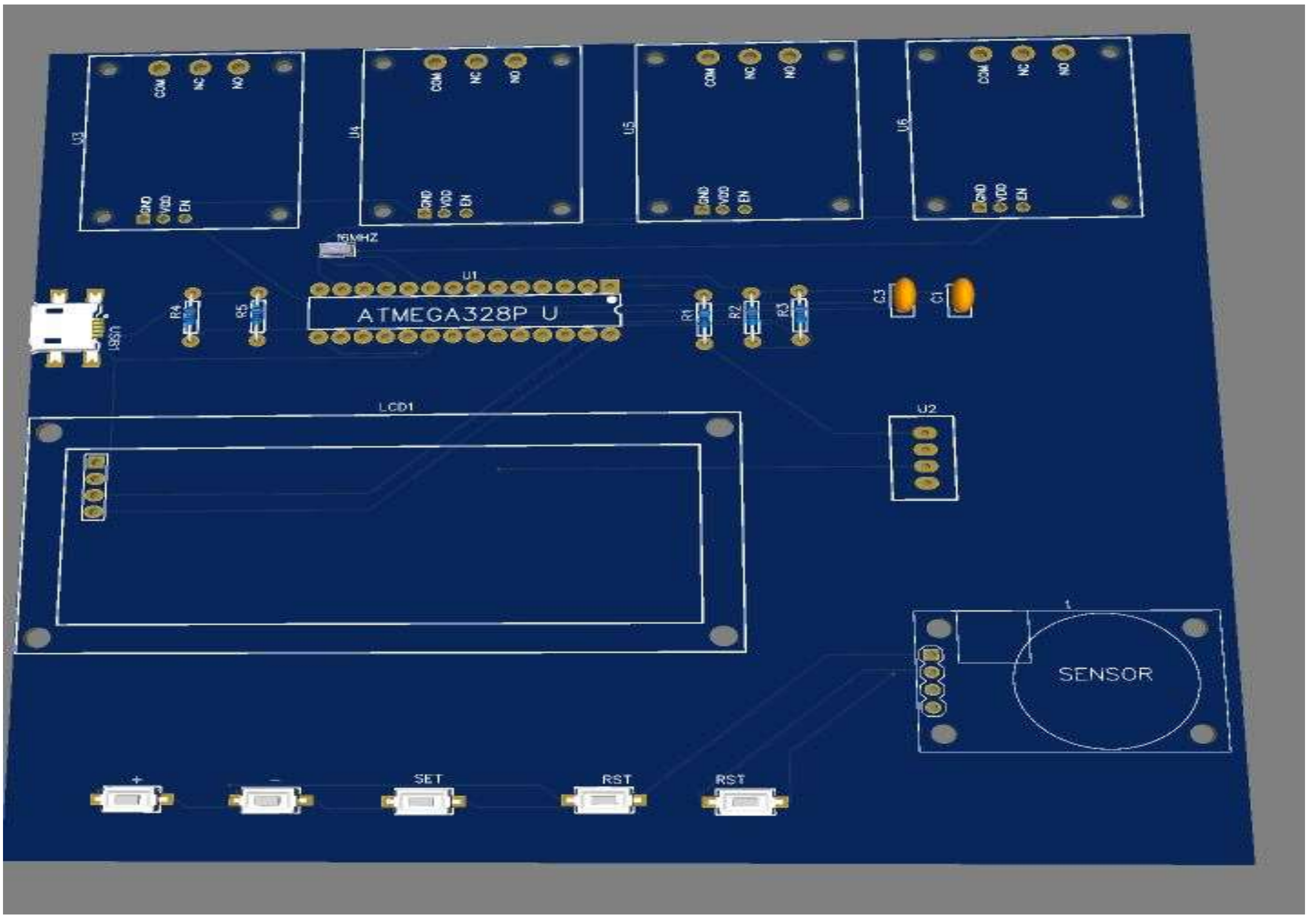


Fig3: PCB 3D view Of This Project

[Download the PCB File link]

Display:

Picture1: Pickup current is set.

Picture2: The system has been reset.

Picture3: Showing Pickup value and instant current value.

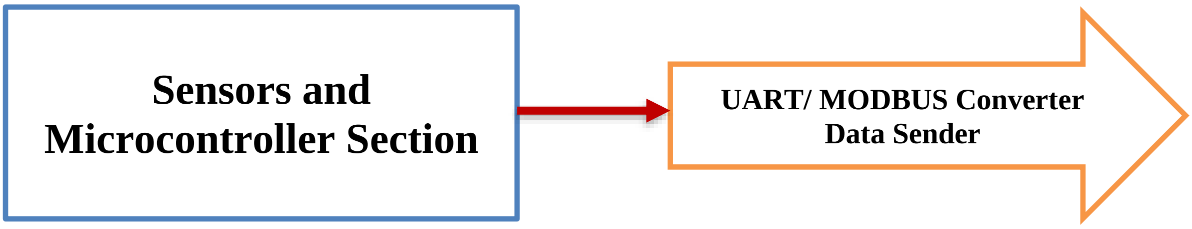
Picture4: Fault detected and waiting for 3 seconds.

Code:

```
#include <LiquidCrystal_I2C.h>
#include "ACS712.h"
#include <IRremote.h>
#include<EEPROM.h>
ACS712 sensor(ACS712_05B, A0);
//ACS712_05B
//ACS712_20A for 20 Amp type
//ACS712_30A for 30 Amp type
float I = 0;
float cc = 0;
float data = 0;
int pp = 0;
int pin = 7;
int pin2 = 8;
int pin3 = 9;
int pin4 = 4;
int ccc = 0;
int bbb = 0;
int hh = 0;
int self = 0;
float picup_current = 0; // intial cosidering current for controlling
LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x3F for a 16 chars and 2 line display
void setup() {
digitalWrite(10, HIGH);
lcd.init(); // lcd display initialize
lcd.clear();// display clearing
lcd.backlight();    // Make sure backlight is o
pinMode(pin, INPUT); //  "
pinMode(pin2, INPUT); // "
pinMode(pin3, INPUT); // "
pinMode(pin4, INPUT); // "
pinMode(2, OUTPUT);// green indecator
pinMode(3, OUTPUT); // blue indecator
pinMode(6, OUTPUT); // red indecator
pinMode(5, OUTPUT); // relay coil
pinMode(10, OUTPUT); //rst
digitalWrite(2, LOW);
digitalWrite(3, HIGH);
digitalWrite(6, LOW);
digitalWrite(5, HIGH);
sensor.calibrate();
Serial.begin(9600);
delay(500);

}

void loop() {

unsigned long sec = millis();
sec = sec / 1000;
int input = digitalRead(pin); //++
int input2 = digitalRead(pin2); // --
int input3 = digitalRead(pin3); // reset
int input4 = digitalRead(pin4); // set
I = sensor.getCurrentAC();
//ignoring the value below 0.09
if (I < 0.09) {
I = 0;
}
lcd.setCursor(0, 1);
lcd.print("Current >");
lcd.print(I);
lcd.print("A");
//Serial.println(I);
lcd.setCursor(0, 0);
lcd.print("Pickup C>");
lcd.print(cc);
lcd.print("A");

// "Increasing" section
if (input == 1 && ccc == 0 )
{
cc = cc + 0.25;
cc = cc;
ccc = !ccc;

}

else if (input == 0 && ccc == 1 )
{
ccc = 0;
}

// "Decreasing" section
else if (input2 == 1 && bbb == 0 && cc > 0)
{
cc = cc - 0.25;
bbb = !bbb;
}

else if (input2 == 0 && bbb == 1 )
{
bbb = 0;
}

// "Reset" section
else if (input3 == 1 &&  hh > 0)
```

```
{
digitalWrite(5, HIGH); // relay off
digitalWrite(6, HIGH); // red led on
digitalWrite(2, LOW);
digitalWrite(3, LOW);
picup_current = cc;
lcd.clear();
lcd.setCursor(4, 0);
lcd.print("<<RESET>>");
lcd.setCursor(4, 1);
lcd.print("<<RESET>>");
delay(1000);
digitalWrite(10, LOW);
hh = 0;
cc = 0;
}

// auto reseting and 3 sec time delay section
else if (I > picup_current && hh > 0)
{
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Fault Detect");
lcd.setCursor(0, 1);
lcd.print("Wait 3 SEC");
//Serial.print(I);
delay(3000);
check();

}

// "SET" section

else if (input4 == 1 && cc > 0 )
{
picup_current = cc;
float fg = map(picup_current * 100, 0, 500, 0, 255);
EEPROM.write('L1', fg);
digitalWrite(5, LOW); // relay on
digitalWrite(3, LOW);
digitalWrite(6, LOW);
digitalWrite(2, HIGH ); // green led on
lcd.clear();
lcd.setCursor(5, 0);
lcd.print("<<SET>>");
lcd.setCursor(5, 1);
lcd.print("<<SET>>");
delay(1000);
hh = 5;
}

// Self Decission section for set
else if ( sec >= 20 && sec <= 30 && hh == 0 && self == 0 )
{
data = EEPROM.read('L1');
Serial.println(data);
data = map(data, 0, 255, 0, 500);
cc = data / 100;
picup_current = cc + 0.20;
digitalWrite(5, LOW); // relay on
digitalWrite(3, LOW);
digitalWrite(6, LOW);
digitalWrite(2, HIGH ); // green led on
lcd.clear();
lcd.setCursor(5, 0);
lcd.print("<<SET>>");
lcd.setCursor(5, 1);
lcd.print("<<SET>>");
delay(1000);
self = 5;
hh = 5;
}
else
{
}
}
// Stability checking section after 3 sec
void check()
{
I = sensor.getCurrentAC();
if (I > picup_current)
{
//Serial.print("YES");
digitalWrite(5, HIGH); // relay off
digitalWrite(6, HIGH); // red led on
digitalWrite(2, LOW);
digitalWrite(3, LOW);
picup_current = cc;
lcd.clear();
lcd.setCursor(4, 0);
lcd.print("<<RESET>>");
lcd.setCursor(4, 1);
lcd.print("<<RESET>>");
delay(1000);
digitalWrite(10, LOW);
hh = 0;
cc = 0;
}
else
{
Serial.print("NO");
}
}
```

Block Diagram Of This Project


Sensors and Microcontroller Section
UART/ MODBUS Converter Data Sender

Implementation Schedule :

- **Week 1: Gather necessary components such as Arduino board, smoke sensor, temperature sensor, DC voltage sensor, relay module, LEDs, resistors, buttons, and wires.**
- **Week 2: Start by setting up the basic circuit using the Arduino board, smoke sensor, temperature sensor, DC voltage sensor, and LEDs. Write the code to read sensor values and display them on the LEDs.**
- **Week 3: Add external set/reset functionality using buttons or switches. Write the code to allow the user to input threshold values for each sensor.**
- **Week 4: Add relay control functionality using the relay module. Write the code to activate relays based on sensor readings.**
- **Week 5: Test the project by simulating various scenarios such as smoke detection, high temperature, and voltage fluctuations. Verify that the relays are working as intended.**
- **Week 6: Finalize the project by adding finishing touches such as a case or enclosure, labels, and instructions. Document the project and its code for future reference.**

Thank You For Watching

Complete details of personal and project names are attached
in the link below: https://github.com/Rafsan12345/CV/blob/main/CV_Md_Hasemi.pdf