#include <LiquidCrystal.h>

#include <WiFi.h>

#include <ThingSpeak.h>

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

const char \*ssid = "Shanid12"; // Change to your WiFi network SSID

const char \*password = "efdn6262"; // Change to your WiFi network password

const unsigned long channelID = 2350289; // Change to your ThingSpeak channel ID

const char \*writeAPIKey = "KP9MF0ACBAKNF8MN"; // Change to your ThingSpeak channel write API key

WiFiClient client;

void setup() {

Serial.begin(9600);

lcd.begin(16, 2);

pinMode(15, INPUT); // Setup for leads off detection LO +

pinMode(21, INPUT); // Setup for leads off detection LO -

WiFi.begin(ssid, password);

ThingSpeak.begin(client);

}

void loop() {

int analogValue = analogRead(A0);

lcd.clear();

// Check for leads off condition

if ((digitalRead(15) == 1) || (digitalRead(21) == 1)) {

// Handle leads off condition if needed

} else {

// Display analog value and health condition on LCD

lcd.setCursor(0, 0);

lcd.print("Analog: ");

lcd.print(analogValue);

lcd.setCursor(0, 1);

lcd.print("Health: ");

if (analogValue >= 500 && analogValue <= 2000) {

lcd.print("Good");

} else {

lcd.print("Bad ");

}

// Print only analog value to Serial Monitor

Serial.println(analogValue);

// Send data to ThingSpeak

ThingSpeak.setField(1, analogValue);

ThingSpeak.setField(2, static\_cast<long>(millis())); // Cast to long to resolve ambiguity

int response = ThingSpeak.writeFields(channelID, writeAPIKey);

if (response == 200) {

Serial.println("Data sent to ThingSpeak successfully");

} else {

Serial.println("Failed to send data to ThingSpeak");

}

}

// Wait for a bit to keep LCD and Serial data from saturating

delay(500);

}