No web data submission

#include <Arduino.h>

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_BMP280.h>

#define ledPin 4 // LED pin

String incomingNum = ""; // Variable to store the sender's number

Adafruit\_BMP280 bmp; // I2C

void setup() {

pinMode(ledPin, OUTPUT);

Serial2.begin(9600, SERIAL\_8N1, 16, 17); // Using Serial2 for GSM module on GPIO 16 (RX) and 17 (TX)

Serial.begin(115200); // USB Serial Monitor

Wire.begin(21, 22); // SDA on GPIO 21, SCL on GPIO 22

if (!bmp.begin(0x76)) { // Address of the BMP280 sensor

Serial.println("Could not find a valid BMP280 sensor, check wiring!");

while (1);

}

// Read and print initial sensor values

printSensorValues();

delay(1000);

sendATCommand("AT+CMGF=1");

delay(1000);

sendATCommand("AT+CNMI=1,2,0,0,0"); // Configure module to send SMS data directly to the serial

delay(1000);

}

void loop() {

if (Serial2.available()) {

String smsText = Serial2.readString();

Serial.println("Received SMS: " + smsText); // Debug: print the whole SMS data

int indexNumStart = smsText.indexOf("+CMT: \"") + 7;

int indexNumEnd = smsText.indexOf('\"', indexNumStart);

incomingNum = smsText.substring(indexNumStart, indexNumEnd);

Serial.println("Sender Number: " + incomingNum); // Debug: print the sender's number

if (smsText.indexOf("#1") != -1) {

String sensorValues = readSensorValues();

sendSMS(sensorValues, incomingNum);

} else if (smsText.indexOf("#4") != -1) {

toggleLED();

sendSMS("LED toggled.", incomingNum);

} else if (smsText.indexOf("#5") != -1) {

pingAndRespond();

}

}

}

void sendSMS(String message, String num) {

Serial.println("Sending SMS to: " + num);

Serial2.print("AT+CMGS=\"");

Serial2.print(num);

Serial2.println("\"");

delay(1000);

Serial2.print(message);

Serial2.write(26); // ASCII code for Ctrl-Z

delay(5000);

printSerial2Response();

}

void toggleLED() {

digitalWrite(ledPin, HIGH);

delay(5000);

digitalWrite(ledPin, LOW);

}

String readSensorValues() {

float temperature = bmp.readTemperature();

float pressure = bmp.readPressure();

float altitude = bmp.readAltitude(1013.25); // Adjust the sea level pressure as necessary

String message = String(temperature) + ", ";

message += String(pressure) + ", ";

message += String(altitude);

return message;

}

void printSensorValues() {

float temperature = bmp.readTemperature();

float pressure = bmp.readPressure();

float altitude = bmp.readAltitude(1013.25); // Adjust the sea level pressure as necessary

Serial.print("Initial Sensor Readings -> ");

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.print(" °C, Pressure: ");

Serial.print(pressure);

Serial.print(" Pa, Altitude: ");

Serial.print(altitude);

Serial.println(" m");

}

void setupGPRS() {

sendATCommand("AT+SAPBR=3,1,\"Contype\",\"GPRS\"");

delay(2000);

sendATCommand("AT+SAPBR=3,1,\"APN\",\"live.vodafone.com\"");

delay(2000);

sendATCommand("AT+SAPBR=1,1");

delay(5000);

sendATCommand("AT+SAPBR=2,1");

delay(2000);

sendATCommand("AT+CSTT=\"live.vodafone.com\",\"\",\"\"");

delay(2000);

sendATCommand("AT+CIICR");

delay(5000);

sendATCommand("AT+CIFSR");

delay(3000);

}

void sendATCommand(String command) {

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

Serial2.println(command);

delay(1000);

printSerial2Response();

}

void printSerial2Response() {

// Read and print the response from Serial2

long timeout = millis() + 5000; // Timeout of 5 seconds

while (millis() < timeout) {

while (Serial2.available()) {

String response = Serial2.readStringUntil('\n');

response.trim(); // Remove any leading/trailing whitespace

if (response.length() > 0) { // Only print non-empty responses

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

}

}

}

}

void pingAndRespond() {

setupGPRS(); // Setup GPRS before pinging

String pingResult = pingIP("1.1.1.1");

sendSMS(pingResult, incomingNum);

}

String pingIP(const char\* ip) {

sendATCommand("AT+CIPPING=\"" + String(ip) + "\""); // Ping command

delay(5000); // Wait for ping response

String response = "";

long timeout = millis() + 5000; // Timeout of 5 seconds

while (millis() < timeout) {

while (Serial2.available()) {

response += Serial2.readStringUntil('\n');

}

}

if (response.indexOf("+CIPPING: 1") != -1) {

return "Successful ping to " + String(ip);

} else {

return "Failed ping attempt to " + String(ip);

}

}