#include <TinyGPS++.h>

#include <HardwareSerial.h>

// Function prototypes

void sendCommand(String command, const int timeout, boolean debug);

void readSMS(int index);

void sendSMS(String number, String text);

void sendGPSLocation(String number);

// Define the hardware serial port for SIM800L

#define SIM800LSerial Serial2

// Define the baud rate for SIM800L

#define SIM800L\_BAUD 9600

// Define the serial port connected to the GPS module

#define GPS\_SERIAL\_PORT Serial1

// Define the baud rate of the GPS module

#define GPS\_BAUD 9600

// Define the GPS data parser object

TinyGPSPlus gps;

// Define the phone number to send the response to

#define RESPONSE\_NUMBER "+923350305078"

// Define the response message

#define RESPONSE\_MESSAGE "Hello"

// Define the interval for sending location (5 minutes in milliseconds)

#define LOCATION\_SEND\_INTERVAL 300000

unsigned long lastLocationSendTime = 0;

void setup() {

Serial.begin(9600); // Initialize Serial Monitor

SIM800LSerial.begin(SIM800L\_BAUD, SERIAL\_8N1, 4, 2); // Initialize SIM800L hardware serial

GPS\_SERIAL\_PORT.begin(GPS\_BAUD, SERIAL\_8N1, 16, 17); // Initialize GPS serial port

delay(30000); // Allow time for the SIM800L module to initialize

Serial.println("Initializing SIM800L...");

// Send initialization commands to SIM800L

sendCommand("AT", 1000, true); // Check if the module is ready

sendCommand("AT+CMGF=1", 1000, true); // Set SMS mode to text mode

sendCommand("AT+CNMI=1,2,0,0,0", 1000, true); // Configure the module to notify about new SMS

Serial.println("Setup complete.");

}

void loop() {

// Check if there is any data from the SIM800L module

if (SIM800LSerial.available()) {

String message = SIM800LSerial.readString();

Serial.println("Received from SIM800L: " + message);

// Check if the message contains "+CMTI" (new SMS indicator)

if (message.indexOf("+CMT:") != -1) {

// Extract the index of the new message

int index = message.substring(message.lastIndexOf(",") + 1).toInt();

readSMS(index); // Read and display the new message

// Print out the content of the received message for debugging

Serial.println("Received message content: " + message);

// Check if the received message is "Hello" from the specified number

if (message.indexOf(RESPONSE\_NUMBER) != -1 && message.indexOf(RESPONSE\_MESSAGE) != -1) {

Serial.println("Conditions met for response.");

// Attempt to send the GPS location as a response to the specified number

if (gps.location.isValid()) {

sendGPSLocation(RESPONSE\_NUMBER);

Serial.println("GPS location sent!");

} else {

Serial.println("No valid GPS location available.");

}

} else {

Serial.println("Message does not meet conditions for response.");

}

}

}

// Keep reading data from the GPS module

while (GPS\_SERIAL\_PORT.available() > 0) {

gps.encode(GPS\_SERIAL\_PORT.read()); // Feed GPS data to the parser

}

// Check if valid GPS data is available and send it immediately if it is the first time

if (gps.location.isValid()) {

if (lastLocationSendTime == 0 || millis() - lastLocationSendTime >= LOCATION\_SEND\_INTERVAL) {

sendGPSLocation(RESPONSE\_NUMBER);

lastLocationSendTime = millis(); // Update the last send time

Serial.println("GPS location sent!");

}

} else {

Serial.println("No valid GPS data.");

}

// Check if there is any data from the Serial Monitor

if (Serial.available()) {

String command = Serial.readStringUntil('\n');

command.trim(); // Remove any leading or trailing whitespace

// Check if the command is to send an SMS

if (command.startsWith("sendSMS")) {

// Extract the phone number and message

int firstComma = command.indexOf(',');

int firstQuote = command.indexOf('\"');

int secondQuote = command.indexOf('\"', firstQuote + 1);

int thirdQuote = command.indexOf('\"', secondQuote + 1);

int fourthQuote = command.indexOf('\"', thirdQuote + 1);

String number = command.substring(firstQuote + 1, secondQuote);

String text = command.substring(thirdQuote + 1, fourthQuote);

sendSMS(number, text);

} else {

// Send the entered command to SIM800L

sendCommand(command, 1000, true);

}

}

}

// Function definitions

void sendCommand(String command, const int timeout, boolean debug) {

SIM800LSerial.println(command);

delay(timeout);

if (debug) {

while (SIM800LSerial.available()) {

Serial.write(SIM800LSerial.read());

}

Serial.println();

}

}

void readSMS(int index) {

// Send command to read the SMS at the given index

sendCommand("AT+CMGR=" + String(index), 1000, true);

}

void sendSMS(String number, String text) {

// Send the command to set the recipient's phone number

sendCommand("AT+CMGS=\"" + number + "\"", 1000, true);

delay(1000);

// Send the SMS text and Ctrl+Z to indicate the end of the message

SIM800LSerial.print(text);

delay(1000);

SIM800LSerial.write(26); // Ctrl+Z ASCII code

delay(1000);

while (SIM800LSerial.available()) {

Serial.write(SIM800LSerial.read());

}

Serial.println();

}

// Function to send the GPS location via SMS

void sendGPSLocation(String number) {

if (gps.location.isValid()) {

// Construct Google Maps URL

String googleMapsURL = "http://maps.google.com/maps?q=" + String(gps.location.lat(), 6) + "," + String(gps.location.lng(), 6);

sendSMS(number, googleMapsURL);

} else {

Serial.println("No valid GPS location available.");

}

}

// Code written by Syed Muhammad Khizer Haider BSCS 23 (NBC)