#include <TinyGPS.h>

#include <SoftwareSerial.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

SoftwareSerial Gsm(7, 8);

char phone\_no[] = "+91XXXXXXXXX"; //replace with phone no. to get sms

TinyGPS gps; //Creates a new instance of the TinyGPS object

int state;

String textMessage;

const int sensor=A1;

const int ldrPin = A0;// Assigning analog pin A5 to variable 'sensor'

float tempc; //variable to store temperature in degree Celsius

float vout; //temporary variable to hold sensor reading

void setup()

{

Serial.begin(9600);

Gsm.begin(9600);

delay(2000);

// AT command to set SIM900 to SMS mode

Serial.print("AT+CMGF=1\r");

delay(100);

// Set module to send SMS data to serial out upon receipt

Serial.print("AT+CNMI=2,2,0,0,0\r");

delay(100);

pinMode(6,INPUT);

pinMode(sensor,INPUT);

pinMode(ldrPin, INPUT);

lcd.begin();

lcd.backlight();

lcd.clear();

lcd.print("Searching ");

lcd.setCursor(0,1);

lcd.print("Network....... ");

delay(3000);

}

void loop()

{

bool newData = false;

unsigned long chars;

unsigned short sentences, failed;

// For one second we parse GPS data and report some key values

for (unsigned long start = millis(); millis() - start < 1000;)

{

while (Serial.available())

{

char c = Serial.read();

Serial.print(c);

if (gps.encode(c))

newData = true;

}

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////

vout=analogRead(sensor); //Reading the value from sensor

vout=(vout\*500)/1023;

tempc=vout;

int ldrStatus = analogRead(ldrPin);// Storing value in Degree Celsius

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////

if(Gsm.available()>0){

textMessage = Gsm.readString();

textMessage.toUpperCase();

delay(10);

}

if(textMessage.indexOf("LOCAL ")>=0){

lcd.clear();

lcd.print("Message Recevived");

delay(3000);

lcd.clear();

lcd.print("Getting ");

lcd.setCursor(0,1);

lcd.print("Location ");

delay(3000);

float flat, flon;

unsigned long age;

gps.f\_get\_position(&flat, &flon, &age);

Gsm.print("AT+CMGF=1\r");

delay(400);

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

Gsm.print(phone\_no);

Gsm.println("\"");

lcd.clear();

lcd.print("Sending location");

lcd.setCursor(0,1);

lcd.print("To Base....");

delay(3000);

Gsm.println("Asheesh ,Battalion No. 1233456 ");

Gsm.print("Temperature is ");

Gsm.println(tempc);

Gsm.print("Heartbeat is ");

Gsm.println(ldrStatus/10);

Gsm.print("http://maps.google.com/maps?q=loc:");

// Gsm.print("Latitude = ");

Gsm.print(flat == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flat, 6);

//Gsm.print(" Longitude = ");

Serial.print(",");

Gsm.print(flon == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flon, 6);

delay(200);

Gsm.println((char)26); // End AT command with a ^Z, ASCII code 26

delay(200);

Gsm.println();

delay(10000);

}

state= digitalRead(6);

if(state==1) //If newData is true

{

float flat, flon;

unsigned long age;

gps.f\_get\_position(&flat, &flon, &age);

Gsm.print("AT+CMGF=1\r");

delay(400);

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

Gsm.print(phone\_no);

Gsm.println("\"");

lcd.clear();

lcd.print("Sending location");

lcd.setCursor(0,1);

lcd.print("To Base....");

delay(3000);

Gsm.println("Alert I need help....Asheesh ,Battalion No. 1233456 ");

Gsm.print("Temperature is ");

Gsm.println(tempc);

Gsm.print("Heartbeat is ");

Gsm.println(ldrStatus/10);

Gsm.print("http://maps.google.com/maps?q=loc:");

// Gsm.print("Latitude = ");

Gsm.print(flat == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flat, 6);

//Gsm.print(" Longitude = ");

Serial.print(",");

Gsm.print(flon == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flon, 6);

delay(200);

Gsm.println((char)26); // End AT command with a ^Z, ASCII code 26

delay(200);

Gsm.println();

delay(10000);

}

else{

lcd.clear();

lcd.print("Soldier Tracking");

lcd.setCursor(0,1);

lcd.print("System ");

delay(2000);

}

Serial.println(failed);

// if (chars == 0)

// Serial.println("\*\* No characters received from GPS: check wiring \*\*");

}