# Description:

Program to send a sms to the number with the google link to the GEO location.

# Source Code:

// include the library code:

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(11,12,14,15,16,17);

char PhoneNo[]="+91xxxxxxxxxx";

int sendStatus=0;

int Device\_Status=0;

int current\_status=0;

char Input[200];

char Longitude[100];

char Latitude[100];

String serialResponse = "";

String arr[22];

int cnt=0;

void setup()

{

//Set pin number 4 as digital out where relay 1 is connected

pinMode(8,OUTPUT);

//SIM808 wakeup connected on pin 13 in IomaTic board

pinMode(13,OUTPUT);

//Initialize the SIM808 Module

digitalWrite(13, HIGH);

delay(1000);

//Sending wake up signal to SIM808 Module

digitalWrite(13, LOW);

delay(1000);

//Keeping SIM808 in active/wakeup state

digitalWrite(13, HIGH);

delay(1000);

//Initialize the LCD in 16x2 mode

lcd.begin(16, 2);

delay(1000);

//Set cursor at first character/coloumn of first line/row

lcd.setCursor(0,0);

//Print the message as metioned cursor location

lcd.print(" IomaTic ");

//Set cursor at first character/coloumn of first line/row

lcd.setCursor(0,1);

//Print the message as metioned cursor location

lcd.print("Dial to Track........");

//Initialize a serial communication with baud rate 9600

Serial.begin(9600);

delay(1000);

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

delay(2000);

Serial.print("AT+CGNSSEQ=\"RMC\"\r\n");

delay(2000);

GetLocation();

}

void loop()

{

//Phone activity status: 0= ready, 2= unknown, 3= ringing, 4= in call

Serial.println("AT+CPAS");

delay(100);

//Decode reply

if (Serial.find("+CPAS: "))

{

// gives ascii code for status number

char c = Serial.read();

// return integer value of ascii code

current\_status = c - 48;

if (current\_status == 0)

{

//Set cursor at first character/coloumn of first line/row

lcd.setCursor(0,1);

//Print the message as metioned cursor location

lcd.print("Waiting For Call...");

}

if (current\_status == 3)

{

//Set cursor at first character/coloumn of first line/row

lcd.setCursor(0,1);

//Print the message as metioned cursor location

lcd.print("Ringing............");

delay(4000);

//Automatically answer call after 1 ring

Serial.println("ATH");

delay(300);

//Set cursor at first character/coloumn of first line/row

lcd.setCursor(0,1);

//Print the message as metioned cursor location

lcd.print("Getting Location....");

GetLocation();

//Set cursor at first character/coloumn of first line/row

lcd.setCursor(0,1);

//Print the message as metioned cursor location

lcd.print("Triggering SMS....");

//Initialize a serial communication with baud rate 9600

Serial.begin(9600);

delay(1000);

//Initialize the GSM modem

Serial.println("AT+CMGF=1");

delay(2000);

//Send dial a phone AT command

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

//Send SMS receiver's phone number

Serial.print(PhoneNo);

//Hex code equivalent to "

Serial.write(0x22);

//Hex code equivalent to carraige return i.e. \r

Serial.write(0x0D);

//Hex code equivalent to new line char i.e. \n

Serial.write(0x0A);

delay(2000);

Serial.print("Tracked Location is: https://www.google.com/maps/search/?api=1&query="); //Test SMS Message Body to Send

Serial.print(arr[3]);

Serial.print(",");

Serial.print(arr[4]);

delay(500);

Serial.println(char(26));

}

}

delay(500);

}

void GetLocation()

{

// OPERATOR

Serial.print("AT+CGNSINF\r\n");

// find operator name between two double quotes

if (Serial.find("+CGNSINF: "))

serialResponse = Serial.readStringUntil('\r\n');

cnt=0;

char buf[sizeof(Input)];

{

serialResponse.toCharArray(buf, sizeof(buf));

char \*p = buf;

char \*str;

while ((str = strtok\_r(p, ",", &p)) != NULL) // delimiter is the semicolon

{

Serial.println(str);

arr[cnt]=str;

cnt++;

}

lcd.clear();

//Set cursor at first character/coloumn of first line/row

lcd.setCursor(0,0);

lcd.print("Lon:");

//Print the message as metioned cursor location

lcd.print(arr[3]);

//Set cursor at first character/coloumn of first line/row

lcd.setCursor(0,1);

lcd.print("Lat:");

//Print the message as metioned cursor location

lcd.print(arr[4]);

}

}

# Libraries:

No additional libraries required.

# Functions:

*Function Name*

Function Description