#include <DFRobot\_sim808.h>

#include <SoftwareSerial.h>

//Mobile phone number, need to change

#define PHONE\_NUMBER "+15195721264"

#define MESSAGE\_LENGTH 160

char message[MESSAGE\_LENGTH];

int messageIndex = 0;

char phone[16];

char datetime[24];

//The content of messages sent

//#define MESSAGE "http://maps.google.com/maps?q=%s,%s"

DFRobot\_SIM808 sim808(&Serial);

char POSITION\_MESSAGE[300];

char lat[12];

char lon[12];

boolean getGPS();

boolean readSMS();

boolean readLocation = false;

void setup() {

//mySerial.begin(9600);

Serial.begin(9600);

//\*\*\*\*\*\*\*\* Initialize sim808 module \*\*\*\*\*\*\*\*\*\*\*\*\*

while(!sim808.init()) {

delay(1000);

Serial.print("Sim808 init error\r\n");

}

Serial.println("Sim808 init success");

//Serial.println("Start to send message ...");

//\*\*\*\*\*\*\*\* define phone number and text \*\*\*\*\*\*\*\*\*\*

//sim808.sendSMS(PHONE\_NUMBER,MESSAGE);

}

void loop()

{

boolean smsRead = readSMS();

//boolean isLocationAvailable = false;

if(smsRead)

{

while(!getGPS())

{

}

delay(000);

sim808.sendSMS(PHONE\_NUMBER,POSITION\_MESSAGE);

delay(10000);

readLocation = false;

}

sim808.detachGPS();

}

boolean readSMS()

{

boolean smsAvailable = false;

messageIndex = sim808.isSMSunread();

Serial.print("messageIndex: ");

Serial.println(messageIndex);

//\*\*\*\*\*\*\*\*\*\*\* At least, there is one UNREAD SMS \*\*\*\*\*\*\*\*\*\*\*

if (messageIndex > 0) {

sim808.readSMS(messageIndex, message, MESSAGE\_LENGTH, phone, datetime);

//\*\*\*\*\*\*\*\*\*\*\*In order not to full SIM Memory, is better to delete it\*\*\*\*\*\*\*\*\*\*

sim808.deleteSMS(messageIndex);

Serial.print("From number: ");

Serial.println(phone);

Serial.print("Datetime: ");

Serial.println(datetime);

Serial.print("Recieved Message: ");

Serial.println(message);

smsAvailable = true;

}

return smsAvailable;

}

boolean getGPS()

{

while(!sim808.attachGPS())

{

Serial.println("Open the GPS power failure");

delay(1000);

}

delay(3000);

Serial.println("Open the GPS power success");

while(!sim808.getGPS())

{

}

//delay(3000);

Serial.print(sim808.GPSdata.year);

Serial.print("/");

Serial.print(sim808.GPSdata.month);

Serial.print("/");

Serial.print(sim808.GPSdata.day);

Serial.print(" ");

Serial.print(sim808.GPSdata.hour);

Serial.print(":");

Serial.print(sim808.GPSdata.minute);

Serial.print(":");

Serial.print(sim808.GPSdata.second);

Serial.print(":");

Serial.println(sim808.GPSdata.centisecond);

Serial.print("latitude :");

Serial.println(sim808.GPSdata.lat);

Serial.print("longitude :");

Serial.println(sim808.GPSdata.lon);

Serial.print("speed\_kph :");

Serial.println(sim808.GPSdata.speed\_kph);

Serial.print("heading :");

Serial.println(sim808.GPSdata.heading);

Serial.println();

float la = sim808.GPSdata.lat;

float lo = sim808.GPSdata.lon;

float ws = sim808.GPSdata.speed\_kph;

dtostrf(la, 4, 6, lat); //put float value of la into char array of lat. 4 = number of digits before decimal sign. 6 = number of digits after the decimal sign.

dtostrf(lo, 4, 6, lon); //put float value of lo into char array of lon

sprintf(POSITION\_MESSAGE, "http://maps.google.com/maps?q=%s,%s", lat, lon);

Serial.println(POSITION\_MESSAGE);

delay(3000);

readLocation = true;

//sim808.sendSMS(PHONE\_NUMBER,POSITION\_MESSAGE);

return readLocation;

}

**package** com.project.packettracer.Call;  
  
**import** android.Manifest;  
**import** android.annotation.SuppressLint;  
  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.net.Uri;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.support.v4.content.ContextCompat;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.text.TextUtils;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
  
**public class** MainActivity **extends** AppCompatActivity **implements** ActivityCompat.OnRequestPermissionsResultCallback {  
  
 **static boolean** *PermissionGranted*;  
 **private static final int *REQUEST\_CALL*** = 10;  
 **private static** String[] *PERMISSIONS\_CALL* = { Manifest.permission.***READ\_PHONE\_STATE***,  
 Manifest.permission.***CALL\_PHONE***, Manifest.permission.***SEND\_SMS***, Manifest.permission.***READ\_SMS*** };  
  
 Button **btnCall**, **btnExit**;  
 EditText **etCallNumber**;  
 EditText **etPassword**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState)  
 {  
 *PermissionGranted* = **false**;  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 SmsReceiver sr = **new** SmsReceiver();  
  
  
  
 sr.*bindListener*(**new** SmsListener() {  
 @Override  
 **public void** messageReceived(String messageText) {  
 Log.*e*(**"Message"**,messageText);  
 *//if(SmsReceiver.senderPhoneNo.equalsIgnoreCase("519-781-1075"))* Toast.*makeText*(MainActivity.**this**,messageText.toString(), Toast.***LENGTH\_SHORT***).show();  
  
 Uri gmmIntentUri = Uri.*parse*(**"geo:0,0?q=37.7749,-122.4192(Your Packet)"**);  
 Intent mapIntent = **new** Intent(Intent.***ACTION\_VIEW***, gmmIntentUri);  
 mapIntent.setPackage(**"com.google.android.apps.maps"**);  
 startActivity(mapIntent);  
 }  
 });  
  
 **btnCall** = findViewById(R.id.*btn\_dial*);  
 **btnExit** = findViewById(R.id.*btn\_exit*);  
 **etCallNumber** = (EditText) findViewById(R.id.*et\_phone\_no*);  
 **final** EditText etPassword = (EditText) findViewById(R.id.*et\_password*);  
  
 **btnCall**.setOnClickListener(**new** View.OnClickListener()  
 {  
 @SuppressLint(**"MissingPermission"**)  
 @Override  
 **public void** onClick(View v)  
 {  
 String phoneNo = **etCallNumber**.getText().toString();  
 **if** (*PermissionGranted*)  
 {  
 **if** (!TextUtils.*isEmpty*(phoneNo))  
 {  
 String dial = **"tel:"** + phoneNo;  
 startActivity(**new** Intent(Intent.***ACTION\_CALL***, Uri.*parse*(dial)));  
 }  
 **else** {  
 Toast.*makeText*(MainActivity.**this**, **"Enter a phone number"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 **else** {  
 Toast.*makeText*(MainActivity.**this**, **"Go to Settings to set your Permissions !"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });  
  
 **btnExit**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 finish();  
 System.*exit*(0);  
 }  
 });  
  
 }  
  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 requestPermissions(*PERMISSIONS\_CALL*,R.string.*msg\_permission*,***REQUEST\_CALL***);  
 }  
  
 **private void** requestPermissions(String[] permissionsCall, **int** msg\_permission, **int** requestCode)  
 {  
*// mErrorString.put(requestCode, msg\_permission);* **int** permissionCheck = PackageManager.***PERMISSION\_GRANTED***;  
 **boolean** showRequestPermissions = **false**;  
  
  
 **for**(String permission: permissionsCall) {  
 permissionCheck = permissionCheck + ContextCompat.*checkSelfPermission*(**this**, permission);  
 showRequestPermissions = showRequestPermissions || ActivityCompat.*shouldShowRequestPermissionRationale*(**this**, permission);  
 }  
  
 **if** (permissionCheck!=PackageManager.***PERMISSION\_GRANTED***)  
 {  
 **if**(showRequestPermissions) {  
 ActivityCompat.*requestPermissions*(MainActivity.**this**, permissionsCall, requestCode);  
 } **else** {  
 ActivityCompat.*requestPermissions*(**this**, permissionsCall, requestCode);  
 }  
 }  
 **else** {  
 onPermissionsGranted(requestCode);  
 }  
 }  
  
 **public void** onPermissionsGranted(**int** requestCode) {  
 *//Do anything when permisson granted* **switch**(requestCode)  
 {  
 **case *REQUEST\_CALL***:  
 *PermissionGranted* = **true**;  
 Toast.*makeText*(getApplicationContext(), **"Permission granted"**, Toast.***LENGTH\_SHORT***).show();  
 **break**;  
 }  
 }  
  
}