CHILD TRACKER WITH EMERGENCY NOTIFIER

SUBMITTED BY

M.MADHAN MOHAN

B.INDRA KIRAN REDDY

P.JAGADEESH

Abstract

This project presents a child safety system with a emergency tracker using GPS and GSM modules.The system can be interconnected with the notification and alert the parents. This detection and messaging system is composed of a GPS receiver, Microcontroller and a GSM Modem. GPS Receiver gets the location information from satellites in the form of latitude and longitude.

The Microcontroller processes this information and this processed information is sent to the user using GSM modem A GSM modem is interfaced to the MCU. The GSM modem sends an SMS to the predefined mobile number. When a child is in danger and in need of self-defense if the child crosses the GEOFENCE the entire system will be activated then immediately a sms will be sent to concern person with location using GSM and GPS.

TECHNOLOGIES USED

* GSM TECHNOLOGY

DEFINITION OF GSM

Global system for mobile communication (GSM) is a globally accepted standard for digital cellular communication. GSM is the name of a standardization gro

up established in 1982 to create a common European mobile telephone standard that would formulate specifications for a pan-European mobile cellular radio system operating at 900 MHz. Figure3.1 GSM module 3.1.2 HISTORY OF GSM Global system for mobile communication is a globally accepted standard for digital cellular communication. GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard that would formulate specifications for a pan-European mobile cellular radio system operating at 900 MHz It is estimated that many countries outside of Europe will join the GSM partnership.GSM, the Global System for Mobile communications, is a digital cellular communications system, which has rapidly gained acceptance and market share worldwide, although it was initially developed in a European context. In addition to digital transmission, GSM incorporates many advanced services and features,

including ISDN compatibility and worldwide roaming in other GSM networks. The advanced services and architecture of GSM have made it a model for future third generation cellular systems, such as UMTS.

GPS RECEIVER: GPS, in full Global Positioning System, space-based radio-navigation system that broadcasts highly accurate navigation pulses to users on or near the Earth. In the United States’ Navstar GPS, 24 main satellites in 6 orbits circle the Earth every 12 hours. In addition, Russia maintains a constellation called GLONASS (Global Navigation Satellite)

Create and monitor geofences

Geofencing combines awareness of the user's current location with awareness of the user's proximity to locations that may be of interest. To mark a location of interest, you specify its latitude and longitude. To adjust the proximity for the location, you add a radius. The latitude, longitude, and radius define a geofence, creating a circular area, or fence, around the location of interest.

You can have multiple active geofences, with a limit of 100 per device user across all apps. For each geofence, you can ask Location Services to send you entrance and exit events, or you can specify a duration within the geofence area to wait, or *dwell*, before triggering an event. You can limit the duration of any geofence by specifying an expiration duration in milliseconds. After the geofence expires, Location Services automatically removes it.



* This project aim is to protect the children when they are alone in the house or an area
* Hardware Required :
* GSM
* GPS
* AURDINO

Block Digram

|  |
| --- |
| 5V power supply |

|  |
| --- |
| Arduino Board |

|  |
| --- |
| GPS Module |

|  |
| --- |
| Cloud Account |

|  |
| --- |
| Geofence |

|  |
| --- |
| GSM Module |

|  |
| --- |
| Notification |

CODE:

#include <TinyGPS++.h>

#include <SoftwareSerial.h>

SoftwareSerial myGPRS(10,11);

TinyGPSPlus gps; // Create an Instance of the TinyGPS++ object called gps

SoftwareSerial ss(7,8);

String host= "api.thingspeak.com";

String GET = "GET /update?key=B76VLGP6385RMYG5";

String latt,lon;

char d;

void setup()

{

// Open serial communications and wait for port to open:

Serial.begin(9600);

myGPRS.begin(9600);

ss.begin(9600);

delay(1000);

sendmyGPRS("AT");

delay(5000);

if(myGPRS.find("OK"))

{

d = myGPRS.read();

//Serial.println(d);

}

}

void loop() // run over and over

{

ss.listen();

Serial.print("Latitude : ");

Serial.println(gps.location.lat(), 5);

Serial.print("Longitude : ");

Serial.println(gps.location.lng(), 4);

smartDelay(1000);

uploadtoThingspeak();

delay(15000);

}

void uploadtoThingspeak(){

myGPRS.listen();

sendmyGPRS("AT+CIPSHUT");

delay(2000);

if( myGPRS.find( "SHUT OK" ) )

{

d = myGPRS.read();

Serial.write(d);

//Serial.println(d);

//return;

}

sendmyGPRS("AT+CIPMUX=1");

delay(2000);

if( myGPRS.find( "OK" ) )

{

d = myGPRS.read();

Serial.write(d);

//Serial.println();

//return;

}

sendmyGPRS("AT+CGATT=1");

delay(2000);

if( myGPRS.find( "OK" ) )

{

d = myGPRS.read();

Serial.write(d);

//Serial.println(d);

//return;

}

String cmd = "AT+CSTT=\"bsnlnet\",\"\",\"\"";// Setup TCP connection

sendmyGPRS(cmd);

delay(2000);

if( myGPRS.find( "OK" ) )

{

d= myGPRS.read();

Serial.write(d);

// Serial.println();

//return;

}

sendmyGPRS("AT+CIICR");

delay(2000);

if( myGPRS.find( "OK" ) )

{

d = myGPRS.read();

Serial.write(d);

//Serial.println();

//return;

}

sendmyGPRS("AT+CIFSR");

delay(2000);

if( myGPRS.find( "Error" ) )

{

myGPRS.println( "ERROR Received" );

//return;

}

else {

d = myGPRS.read();

Serial.write(d);

// Serial.println();

}

cmd = "AT+CIPSTART=3,\"TCP\",\"";// Setup TCP connection

cmd += host;

cmd += "\",80";

sendmyGPRS(cmd);

delay(4000);

if( myGPRS.find("CONNECTED") )

{

d = myGPRS.read();

//Serial.write(d);

Serial.println("Connected to Thingspeak");

//return;

}

String cmd1 = "AT+CIPSEND=3,100";// Setup TCP connection

sendmyGPRS(cmd1);

delay(2000);

cmd = GET+ "&field1="+String(gps.location.lat(),DEC)+"&field2="+String(gps.location.lng(),DEC)+"\r\n";

if( myGPRS.find( ">" ) )

{

myGPRS.println(cmd);

myGPRS.println();

Serial.println(cmd);

}

else

{

sendmyGPRS( "AT+CIPCLOSE" );//close TCP connection

}

sendmyGPRS("AT+CIPCLOSE");

}

void sendmyGPRS(String cmd)

{

myGPRS.println(cmd);

//Serial.println(cmd);

}

static void smartDelay(unsigned long ms) // This custom version of delay() ensures that the gps object is being "fed".

{

unsigned long start = millis();

do

{

while (ss.available())

gps.encode(ss.read());

} while (millis() - start < ms);

}