```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x3f, 16, 2);

#include <TinyGPS++.h>
TinyGPSPlus gps;

int tempPin=35;
float latt;
float longi;

int sensor_value,bp1=0;
int tempdata;
int cel;
int celcius;
int sw1=2;
int sw2=14;
int val;

////////////////////////////////////////////
#include "ThingSpeak.h"
#include "secrets.h"
#include <WiFi.h>

char ssid[] = "Ramya's iPhone";   // your network SSID (name)
char pass[] = "dan03123";   // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)
WiFiClient  client;

unsigned long myChannelNumber = 1069052;
const char * myWriteAPIKey = "JRYCNL54FNNAPRTE";

// Initialize our values

String myStatus = "";



void setup(void)
{
  Serial.begin(9600);
  Serial2.begin(9600);
  lcd.begin();
  pinMode(sw1,INPUT);
```

```
  pinMode(sw2,INPUT);
  // Start up the library

  Serial.println(F("DeviceExample.ino"));
  Serial.println(F("A simple demonstration of TinyGPS++ with an attached GPS module"));
  Serial.print(F("Testing TinyGPS++ library v. "));
  Serial.println(TinyGPSPlus::libraryVersion());
  Serial.println(F("by Mikal Hart"));
  Serial.println();

   WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);  // Initialize ThingSpeak


}


void loop(void)
{
  if(WiFi.status() != WL_CONNECTED){
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(SECRET_SSID);
    while(WiFi.status() != WL_CONNECTED){
      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using
open or WEP network
      Serial.print(".");
      lcd.setCursor(0, 0);
   lcd.print("connecting..");
      delay(5000);
    }
    Serial.println("\nConnected.");
  }

  for(int z=0;z<100;z++)
{
tempdata = analogRead(tempPin);
Serial.print("adc=");
Serial.println(tempdata);
float mv = (tempdata/4096.0)*3700;
Serial.print("MV=");
Serial.println(mv);
cel = mv/10;
celcius=celcius+cel;
}
celcius=celcius/100;
```

```
  for(int i=0;i<100;i++)
  {

   sensor_value = analogRead(34);
 float  bp = (sensor_value/120)*3.3;
       bp1=bp1+bp;
  }
  bp1=bp1/100;
  Serial.print("temp=");
    Serial.println(celcius);
    Serial.print("BPM=");
    Serial.println(bp1);
     lcd.setCursor(0, 0);
   lcd.print("T:");
  lcd.print(celcius);
  lcd.print(" ");
delay(2000);
     lcd.print("BPM:");
  lcd.print(bp1);
 lcd.print("     ");
loc();
if(digitalRead(sw1)==LOW)
{
send_sms();
}

  ThingSpeak.setField(1, latt);
  ThingSpeak.setField(2, longi);
  ThingSpeak.setField(3, bp1);
  ThingSpeak.setField(4, celcius);



  // write to the ThingSpeak channel
  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  if(x == 200){
    Serial.println("Channel update successful.");
  }
  else{
    Serial.println("Problem updating channel. HTTP error code " + String(x));
  }

  // change the values
```

```
  ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  //delay(20000); // Wait 20 seconds to update the channel again
 // accident="";




 }


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

//////////////////////////////////////////////
void send_sms()
{

 lcd.setCursor(0,0);
  //  lcd.print("sending sms..");
  // put your main code here, to run repeatedly:
  Serial.print("AT=1\r");
  delay(100);
  Serial.print("AT+CMGF=1\r");
  delay(100);
    Serial.print("AT+CMGS=\"+919844977066\"\r");
  delay(100);
  Serial.println("CHILD IS IN PROBLEM  AT THE LOCATION=");
  Serial.println("https://earth.google.com/web/search/");
  Serial.print(latt);
  Serial.print(",");
  Serial.print(longi);
  Serial.println("/");
    delay(100);
  Serial.write(0x1A);
  delay(2000);
   Serial.write(0x1A);
  delay(2000);
  Serial.write(0x1A);
  delay(2000);
// accident="accident occured";

 //////////////////////////////////////////////
Serial2.print("AT=1\r");
  delay(100);
  Serial2.print("AT+CMGF=1\r");
  delay(100);
    Serial2.print("AT+CMGS=\"+919844977066\"\r");
```

```
  delay(100);
  Serial2.print("CHILD IS IN PROBLEM AT THE LOCATION=");
  Serial2.println("https://earth.google.com/web/search/");
  Serial2.print(latt);
  Serial2.print(",");
  Serial2.print(longi);
  Serial2.println("/");
    delay(100);
  Serial2.write(0x1A);
  delay(2000);
   Serial2.write(0x1A);
  delay(2000);
  Serial2.write(0x1A);
  delay(2000);

   lcd.setCursor(0,0);
    lcd.print("msg sent....");
 ////////////////////////////////

}

////////////////////////////////////////////////
void loc()
{
  // This sketch displays information every time a new sentence is correctly encoded.
  while (Serial2.available() > 0)
    if (gps.encode(Serial2.read()))
      displayInfo();

  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
    while(true);
  }
}

void displayInfo()
{
  Serial.print(F("Location: "));
  if (gps.location.isValid())
  {
    Serial.print(gps.location.lat(), 7);
    Serial.print(F(","));
    Serial.print(gps.location.lng(), 7);
    latt=(gps.location.lat(), 7);
    longi=(gps.location.lng(), 7);
```

```
   lcd.setCursor(0,1);
  lcd.print(gps.location.lat(),4);
  lcd.setCursor(6,1);
  lcd.print("  ");
  lcd.setCursor(8,1);
  lcd.print(gps.location.lng(),4);
}
else
{
  Serial.print(F("INVALID"));
}

Serial.print(F("  Date/Time: "));
if (gps.date.isValid())
{
  Serial.print(gps.date.month());
  Serial.print(F("/"));
  Serial.print(gps.date.day());
  Serial.print(F("/"));
  Serial.print(gps.date.year());
}
else
{
  Serial.print(F("INVALID"));
}

Serial.print(F(" "));
if (gps.time.isValid())
{
  if (gps.time.hour() < 10) Serial.print(F("0"));
  Serial.print(gps.time.hour());
  Serial.print(F(":"));
  if (gps.time.minute() < 10) Serial.print(F("0"));
  Serial.print(gps.time.minute());
  Serial.print(F(":"));
  if (gps.time.second() < 10) Serial.print(F("0"));
  Serial.print(gps.time.second());
  Serial.print(F("."));
  if (gps.time.centisecond() < 10) Serial.print(F("0"));
  Serial.print(gps.time.centisecond());
}
else
{
  Serial.print(F("INVALID"));
}
```

```
  Serial.println();
}
```