```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <SD.h>

//GPS variables
static const int RXPin = 4, TXPin =
3;
static const int GPSBaud = 9600;
double latitude=0,longitude=0;
int minutes=0, seconds=0, hours=0;

//basic input variables
static const int switchPin = 2,
button1 = 9, button2 = 8;

//SD card variables
File myFile;

//basic output variables
static const int buzzer = 7, LED = 6;
```

```
// The TinyGPS++ object
TinyGPSPlus gps;

// The serial connection to the GPS
device
SoftwareSerial ss(RXPin, TXPin);

/************************************
*************************************
****************************/

void setup(){
  pinMode(switchPin,INPUT_PULLUP);
  pinMode(button1,INPUT_PULLUP);
  pinMode(button2,INPUT_PULLUP);
  pinMode(buzzer,OUTPUT);
  pinMode(LED,OUTPUT);
  pinMode(10, OUTPUT);

  Serial.begin(9600);
  ss.begin(GPSBaud);   //Establishes
a software serial connection with
the GPS with a 9600 Baud, which the
GPS can only function at



//Initializes the SD card and makes
sure an SD card is present
  if (!SD.begin(10)) {
    Serial.println("initialization
failed!");
    while(1){}
  }
  Serial.println("initialization
done.");

SDWRITE();
       //Gives an indication that
the device is started on the SD card
}

/*************************************
```

```
*************************************
****************************/
/************************************
*************************************
****************************/

//This section is the main loop of
the code, where the switch input is
continuously tested for and then the
button inputs are continually tested
for, then starts up seperate
functions depending on what was
pressed

void loop(){

while (digitalRead(switchPin) ==
LOW){               //when switch is
flipped
  analogWrite(LED,
25);
//turn LED on to a lower intensity

GPSDATA();
          //begin recording GPS data

    if (digitalRead(button1) ==
LOW){            //if emergency
button is pressed, begin Panic
function
      PANIC();
    }

    if (digitalRead(button2) ==
LOW){            //if discreet
button is pressed, begin QUIET
function
      QUIET();
    }

}
```

```
noTone(buzzer);
            //if the switch is
turned off, then deactivate buzzer
digitalWrite(LED,
LOW);                           //if
the switch is turned off, then
deactivate the LED
}

/************************************
*************************************
****************************/
/************************************
*************************************
****************************/

//This section receives and stores
the GPS data as several variables
for use in the SDWRITE function

void GPSDATA(){
 if (ss.available() >
0){                        //if
there is a serial connection with
the GPS (if there is not something
is wrong with the wiring)

  gps.encode(ss.
read());
//decode the data from the GPS by
parsing through the strings returned

    if (gps.location.
isUpdated()){              //if two
satellites are located to accurately
tell the position of the user

      //Latitude
      latitude=gps.location.lat(),
6;            //save the latitude as
a variable
      Serial.print("Latitude= ");
      Serial.print(gps.location.
lat(), 6);

      //Longitude
      longitude=gps.location.lng(),
6;           //save the longitude as
a variable
      Serial.print(" Longitude= ");
      Serial.println(gps.location.
lng(), 6);

      // Hour
(0-23)                   //save the
hour as a variable
      hours=gps.time.hour()-5;
//conversion from UTC to EST
      if (hours<0){
//hour conversion overflow
      hours=hours+24;
      }
      // Minute
(0-59)                //save the
minutes as a variable
      minutes=gps.time.minute();

      // Second (0-59)
//save the seconds as a variable
      seconds=gps.time.second();
  }
}
}
/************************************
*************************************
****************************/

//This section is the discreet
functionality, that only records the
GPS data to the SD card and turns on
the LED to a higher intensity

void QUIET(){
  for(int a=0;a<=400;a++){   //This
loop is important, as the GPS data
needs to be tested a certain amount
of times in a row for the GPS to
output all the data strings before
the location can be recorded
  GPSDATA();
  }

  digitalWrite(LED,HIGH);
//turn LED on HIGH
  SDWRITE();
//Write GPS data to SD card

  if (digitalRead(switchPin) ==
HIGH){          //if the device is
switched off, return to the main
loop so everything shuts off
      return loop();
    }
  else if (digitalRead(button1) ==
LOW){        //if the emergency
button is pressed, switch to the
PANIC mode
    PANIC();
  }
    else
{
  //if nothing is pressed, this
function will loop indefinitely
      return QUIET();
    }
}


/************************************
*************************************
****************************/

//This section is the emergency
functionality, that both records the
```

```
GPS data to the SD card, turns on
the buzzer, and flashes the LED

void PANIC(){
  tone(buzzer, 50);     //activate
buzzer

  for(int a=0;a<=5;a++){   //this
loop flashes the LED and receives
the GPS data

    digitalWrite(LED,HIGH);

    for(int a=0;a<=400;a++){
    GPSDATA();
    }

    digitalWrite(LED,LOW);
    delay(50);              //delay
to make LED turning on and off
visible
```

```
  }

  SDWRITE();
//Record GPS data to SD card

    if (digitalRead(switchPin) ==
HIGH){      //If the switch is
turned off, then everything turns off
      return loop();
    }
    else
{                            //if
the switch is not pressed, then this
function will repeat indefinitely
      return PANIC();
    }
}

/************************************
*************************************
****************************/
```

```
//This section writes to the SD card
the variables that were stored in
the GPS function

void SDWRITE() {



  myFile = SD.open("data.txt",
FILE_WRITE);           //create data
file

  // if the file opened okay, write
to it:

  if (myFile) {

    //printing the longitude
    myFile.print("Longtitude: ");
    myFile.println(longitude);
    //printing the latitude
    myFile.print("Latitude: ");
    myFile.println(latitude);


    //printing the time
    myFile.print("Time: ");
    myFile.print(hours);
    myFile.print(":");
    myFile.print(minutes);
    myFile.print(":");
    myFile.println(seconds);

    myFile.println(" ");

  // close the file:
    myFile.close();

  } else {
    // if the file didn't open,
print an error:
    Serial.println("error opening
test.txt");
  }

}
```