```cpp
#include <SoftwareSerial.h>
#include <TinyGPS.h>

#include <ESP8266WiFi.h>

#include "DHT.h"

#define DHTPIN 2   // what digital pin we're connected to
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
#define RXPIN 3
#define TXPIN 1
#define GPSBAUD 9600

// Create an instance of the TinyGPS object
TinyGPS gps;
// Initialize the NewSoftSerial library to the pins you defined above
SoftwareSerial uart_gps(RXPIN, TXPIN);
void getgps(TinyGPS &gps);

WiFiClient client;

String MakerIFTTT_Key ;
;String MakerIFTTT_Event;
char *append_str(char *here, String s) {  int i=0; while (*here++ = s[i]){i++;};return here-1;}
char *append_ul(char *here, unsigned long u) { char buf[20]; return append_str(here,
ultoa(u, buf, 10));}
char post_rqst[256];char *p;char *content_length_here;char *json_start;int compi;

void setup()
{
  Serial.begin(9600);
  while (!Serial);
   uart_gps.begin(GPSBAUD);
   delay(1500);
    Serial.println("");
  Serial.println("GPS Shield QuickStart Example Sketch v12");
  Serial.println("      ...waiting for lock...          ");
  Serial.println("");
  WiFi.disconnect();
  delay(500);
  Serial.println("DHTxx test!");

  dht.begin();
  delay(1000);
  //collecting the variables
  float h = dht.readHumidity();
```

```
   float t = dht.readTemperature();
  delay(500);
   String buf1,buf2;
   //converting them into strings
  buf1+= F("Car temprature :");
  buf1 += String(t, 6);
   buf2 += F("Car humidity :");
  buf2 += String(h, 6);
  buf2 += F("\n");
        Serial.print(buf1);
        Serial.print("  ");
        Serial.print(buf2);
        Serial.print("\n");
        //this is the code for the gps moduel
  while(uart_gps.available())    // While there is data on the RX pin...
  {
    Serial.print("Humidity:");
        Serial.print(h);
        Serial.print(" %\t");
        Serial.print("Temperature: ");
        Serial.print(t);
        Serial.print(" *C \n");

     int c = uart_gps.read();   // load the data into a variable...
     if(gps.encode(c))     // if there is a new valid sentence...
     {
      getgps(gps);        // then grab the data.
     }
  }


  Serial.println("START");
   WiFi.begin("notconnected","Ekminute");
  while ((!(WiFi.status() == WL_CONNECTED))){
   delay(300);
   Serial.print("..");

  }
 Serial.println("Connected");
  Serial.println("Your IP is");
  Serial.println((WiFi.localIP().toString()));
  if (client.connect("maker.ifttt.com",80)) {
   MakerIFTTT_Key ="cAGnF95BNtQfk4ZGpnv34KQmEwka2pMWNoHylXrSTHV";
   MakerIFTTT_Event ="sms";
   p = post_rqst;
   p = append_str(p, "POST /trigger/");
   p = append_str(p, MakerIFTTT_Event);
```

```
    p = append_str(p, "/with/key/");
    p = append_str(p, MakerIFTTT_Key);
    p = append_str(p, " HTTP/1.1\r\n");
    p = append_str(p, "Host: maker.ifttt.com\r\n");
    p = append_str(p, "Content-Type: application/json\r\n");
    p = append_str(p, "Content-Length: ");
    content_length_here = p;
    p = append_str(p, "NN\r\n");
    p = append_str(p, "\r\n");
    json_start = p;
    p = append_str(p, "{\"value1\":\"");
    p = append_str(p,buf1);
    p = append_str(p, "\",\"value2\":\"");
    p = append_str(p,buf2);
    p = append_str(p, "\",\"value3\":\"");
    p = append_str(p,"");
    p = append_str(p, "\"}");

    compi= strlen(json_start);
    content_length_here[0] = '0' + (compi/10);
    content_length_here[1] = '0' + (compi%10);
    client.print(post_rqst);

  }

}


void loop()
{


}

// The getgps function will get and print the values we want.
void getgps(TinyGPS &gps)
{
  // To get all of the data into varialbes that you can use in your code,
  // all you need to do is define variables and query the object for the
  // data. To see the complete list of functions see keywords.txt file in
  // the TinyGPS and NewSoftSerial libs.

  // Define the variables that will be used
  float latitude, longitude;
  // Then call this function
  gps.f_get_position(&latitude, &longitude);
  // You can now print variables latitude and longitude
```

```
    Serial.print("Lat/Long: ");
    Serial.print(latitude,5);
    Serial.print(", ");
    Serial.println(longitude,5);

    // Same goes for date and time
    int year;
    byte month, day, hour, minute, second, hundredths;
    gps.crack_datetime(&year,&month,&day,&hour,&minute,&second,&hundredths);
    // Print data and time
    Serial.print("Date: "); Serial.print(month, DEC); Serial.print("/");
    Serial.print(day, DEC); Serial.print("/"); Serial.print(year);
    Serial.print("  Time: "); Serial.print(hour, DEC); Serial.print(":");
    Serial.print(minute, DEC); Serial.print(":"); Serial.print(second, DEC);
    Serial.print("."); Serial.println(hundredths, DEC);
    //Since month, day, hour, minute, second, and hundr

    // Here you can print the altitude and course values directly since
    // there is only one value for the function
    Serial.print("Altitude (meters): "); Serial.println(gps.f_altitude());
    // Same goes for course
    Serial.print("Course (degrees): "); Serial.println(gps.f_course());
    // And same goes for speed
    Serial.print("Speed(kmph): "); Serial.println(gps.f_speed_kmph());
    Serial.println();

    // Here you can print statistics on the sentences.
    unsigned long chars;
    unsigned short sentences, failed_checksum;
    gps.stats(&chars, &sentences, &failed_checksum);
    //Serial.print("Failed Checksums: ");Serial.print(failed_checksum);
    //Serial.println(); Serial.println();
}
```