

```

1  #include <stdio.h>
2  #include <time.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #include <ctype.h>
6  #include "types.h"
7  #include "config.h"
8  #include "errors.h"
9  #include "date.h"
10 #include "gps.h"
11
12
13 /*****Prototipos*****/
14 status_t _parse_field_time(char [], struct tm *);
15 /*****/
16
17 /*****
18 Función para el proceso de los datos cargados.
19 Lee una línea del flujo de entrada, si esta empieza con el
20 encabezado de protocolo que contiene los datos de geolocalización
21 devuelve FOUND, extrae la hora y lo guarda en la estructura que recibe.
22 Sino devuelve NOT_FOUND.
23 *****/
24
25 status_t parse_line(struct tm *time_struct){
26     status_t st;
27     char line[MAX_LINE];
28     char field_time[FIELD_SPAN_TIME + 1];
29     size_t i;
30
31     if(time_struct == NULL)
32         return ERROR_NULL_POINTER;
33     if(fgets(line, MAX_LINE - 2, stdin) == NULL)
34         return END_OF_FILE;
35     if(!strcmp(line, ID_MSG, FIELD_SPAN_ID_MSG)){
36         for(i = 0; i < FIELD_SPAN_TIME; i++){
37             if(!isdigit(line[i + FIELD_POS_TIME]))
38                 return ERROR_READ_LINE;
39             field_time[i] = line[i + FIELD_POS_TIME];
40         }
41         field_time[FIELD_SPAN_TIME + 1] = '\0';
42         if((st = _parse_field_time(field_time, time_struct)) != OK)
43             return ERROR_INVALID_DATA;
44         return FOUND;
45     }
46     return NOT_FOUND;
47 }
48
49 /*****
50 Recibe una cadena de caracteres que contiene dígitos,
51 verifica que corresponda al formato de horas hhhmmss,
52 y si es así lo guarda en la estructura que recibe.
53 *****/
54
55 status_t _parse_field_time(char field[], struct tm *time_struct){
56     int aux;
57     int value;
58
59     value = atoi (field);
60     aux = (value / 10000);
61     if(aux > 23)

```

```
62         return ERROR;
63     (*time_struct).tm_hour = aux;
64     aux = (value%10000)/100;
65     if(aux > 59)
66         return ERROR;
67     (*time_struct).tm_min = aux;
68     aux = (value%100);
69     if(aux > 59)
70         return ERROR;
71     (*time_struct).tm_sec = aux;
72
73     return OK;
74 }
```