

## Capítulo 2: Estructura del sistema

### 2.1 Introducción

En este capítulo se pretende describir globalmente el funcionamiento del sistema así como sus partes. En el capítulo 5, se describirá como se ha desarrollado cada una de éstas y se explicará el código asociado.

### 2.2 Modularización del sistema

De la figura 1.1 se observa que en este proyecto aparecen varias partes bien diferenciadas entre sí. Por un lado, se encuentran aquellos aspectos relacionados con el manejo de la IMU. A destacar:

- Configuración del puerto serie para el establecimiento de la comunicación entre cada sistema móvil y su IMU.
- Conversión de datos. Ya que según la documentación de Kearfott, la IMU utiliza una estructura de bits determinada para enviar y recibir datos.
- Envío y recepción de datos por el puerto serie de forma síncrona. La IMU posee dos modos de funcionamiento bien claros. El primero consiste en, tras el envío de un paquete por parte de la CPU, la respuesta a esa petición. El segundo trata del envío periódico de datos por parte de la IMU.

Por otro lado, al aparecer una comunicación Ethernet, se ha diseñado un protocolo para sincronizar ambas viper entre sí y con el monitor central. Lo que supone el desarrollo de una máquina de estados capaz de gestionar todo el sistema, así como la estructura de mensajes intercambiados entre los equipos.

Por último, aquellos aspectos relacionados con el almacenamiento de la configuración del sistema en la memoria no volátil del computador, así como de los experimentos realizados para su posterior análisis.

## 2.3 Configuración del sistema

Debido a la necesidad de realizar un mismo software para dos computadores empotrados diferentes, nos vemos obligados a la necesidad de almacenar ciertos datos de configuración. Algunos de estos datos son:

- IP local de la viper.
- IP remota de la otra viper.
- IP remota del PC servidor.
- Puerto serie utilizado en el sistema móvil para conectarnos a la IMU, ya que estos sistemas viper poseen más de un puerto COM.
- Puertos UDP y TCP para las conexiones Ethernet.
- Modo de trabajo. Nos indica si estamos en modo depuración o no del sistema.

El sistema viper que se ha utilizado, dispone de cierta memoria no volátil que permite almacenar la configuración. Ésta se almacena en una carpeta que posee el sistema, denominada "Flashdisk" y que se encuentra en el directorio raíz. El contenido de esta carpeta no se pierde al reiniciar la viper. En el capítulo 4 se explica detalladamente como se ha llevado a cabo este cometido.

No obstante, no sólo es necesario leer el contenido de un fichero para poder leer la configuración del sistema. También necesitamos poder escribir tras finalizar un experimento, ya que los datos de los experimentos necesitan ser salvados para su envío al monitor central y posterior procesamiento.

## 2.4 Comandos y mensajes enviados y recibidos por la IMU

Una gran parte del proyecto, se ha invertido en la compactación correcta de los datos por parte del sistema móvil, para facilitar el envío y recepción de paquetes hacia la IMU. Los paquetes enviados desde la CPU hacia la IMU poseen las siguientes características:

- La CPU debe iniciar la comunicación enviando un paquete de petición. La IMU no enviará ningún mensaje sin una petición previa.
- Existe una opción de envío periódico de datos que se explicará más adelante.
- Los comandos y mensajes están formados por estructuras de bytes.

Por lo tanto, la comunicación con la IMU se traduce a una serie de bytes, ordenados según una estructura determinada. La definición de esta estructura se muestra a continuación:

**Tabla 1 Mensajes enviados hacia la IMU desde la Viper**

Byte	Nombre	Descripción
1	Número de bytes (N)	Este octeto nos informa del número bytes del mensaje, incluido el primer byte
2	Número de comando y de secuencia (S/C)	En este byte nos encontramos con el código del mensaje y el número de secuencia. Los 3 bits más significativos se corresponden con el número de secuencia y los 5 menos significativos con el código del mensaje
3	Datos (D)	De aquí en adelante tendremos tantos bytes de datos como los que se hayan especificado en el primer byte (N-2)

**Tabla 2 Mensajes recibidos desde la IMU**

Byte	Nombre	Descripción
1	Número de bytes (N)	Este octeto nos informa del número bytes del mensaje, incluido el primer byte
2	Identificador del mensaje y número de secuencia (S/I)	En este byte nos encontramos con el identificador del mensaje y el número de secuencia. Los 3 bits más significativos se corresponden con el número de secuencia y los 5 menos significativos con el identificador del mensaje
3 y 4	Bytes de estatus (S1 y S2)	La IMU nos envía información de su estado en estos dos bytes. Cada bit nos indica la presencia o ausencia de un particular modo o condición
5	Datos	De aquí en adelante tendremos tantos bytes de datos como los que se hayan especificado en el primer byte (N-4)

Por lo tanto, se observa como tanto en los mensajes enviados a la IMU, como los que se reciben de ella, el primer byte es un indicador de la longitud del paquete total. Gracias a esto, y a que cada mensaje que devuelve la IMU va precedido de una petición, se ha diseñado nuestro sistema.

El segundo byte es una combinación del código de mensaje y del número de secuencia. Cada tipo de mensaje que se envía posee un código identificativo. El número de secuencia, aunque no se ha tenido en cuenta al diseñar el propio protocolo, se ha contemplado a la hora de desarrollar el sistema. De esta forma, se ha creado una serie de funciones que automatizan el proceso de colocar el número de secuencia en los 3 bits más significativos y el código de mensaje en los 5 menos significativos.

Los siguientes bytes son los de estatus. En estos se recoge la información que la IMU devuelve cada vez que se hace una petición. Existe la posibilidad de que la IMU envíe un tercer byte S3 cuando se le envía el mensaje "EXPANDED STATUS DATA".

Por último, los restantes bytes son los relativos a los datos que hayan sido solicitados. En el capítulo 5, se describe la solución adoptada para almacenar todos estos datos y facilitar el envío por el puerto serie.

## **2.5 Comunicación Ethernet**

Además de la conexión por el puerto serie con la IMU, cada viper está conectada mediante Ethernet con la otra Viper y con el PC monitor.

### **2.5.1 Conexión UDP**

La conexión con la otra CPU se utiliza para poder sincronizar ambas IMUs en las etapas de alineación, inicio del trayecto y fin del trayecto. Además, los experimentos que se pueden realizar con la IMU y que son presentados al operario en la interfaz gráfica, son enviados desde el PC servidor en formato de texto plano. De esta forma, la viper al arrancar, además de leer los datos de configuración también lee los posibles experimentos.

Uno de los mayores problemas a la hora de realizar el proyecto, ha sido la elaboración de la comunicación UDP. Esta tarea, que a priori parecía sencilla mediante el uso de las clases CSocket proporcionadas por la API de Windows, supuso un grave retraso, debido a que en el sistema Windows CE 5.0 bajo el que funciona la Viper, no funcionaba correctamente el mapeo

de mensajes entre el sistema operativo y el software. Aunque el envío de datos por la red si se realizaba correctamente, a la hora de recibir un paquete por Ethernet, el software no recibía notificación alguna por parte del sistema operativo. En el capítulo 5 se profundiza más acerca de este inconveniente así como la solución adoptada.

## 2.5.2 Conexión TCP

Para poder comunicar la CPU con el monitor central se ha hecho uso de una conexión TCP, ya que necesitábamos un protocolo orientado a conexión para realizar el envío de ficheros.

La idea consiste en poder enviar a la CPU desde el monitor central, tanto el fichero de configuración como la lista de experimentos. De esta forma, cuando la CPU arranque, podrá leer estos ficheros y mostrar al operario los experimentos disponibles.

La interfaz que se le presenta al usuario en el monitor central, puede verse a continuación.

The image shows a software window titled "Servidor PC". It contains two main panels. The left panel is labeled "Móvil" and includes an "IP:" text box, a "Mandar configuración" button, a "Mandar experimentos" button, and a "Descargar resultados" button. The right panel is labeled "Referencia" and includes an "IP:" text box and a "Mandar configuración" button. At the bottom right of the window is an "Aceptar" button.

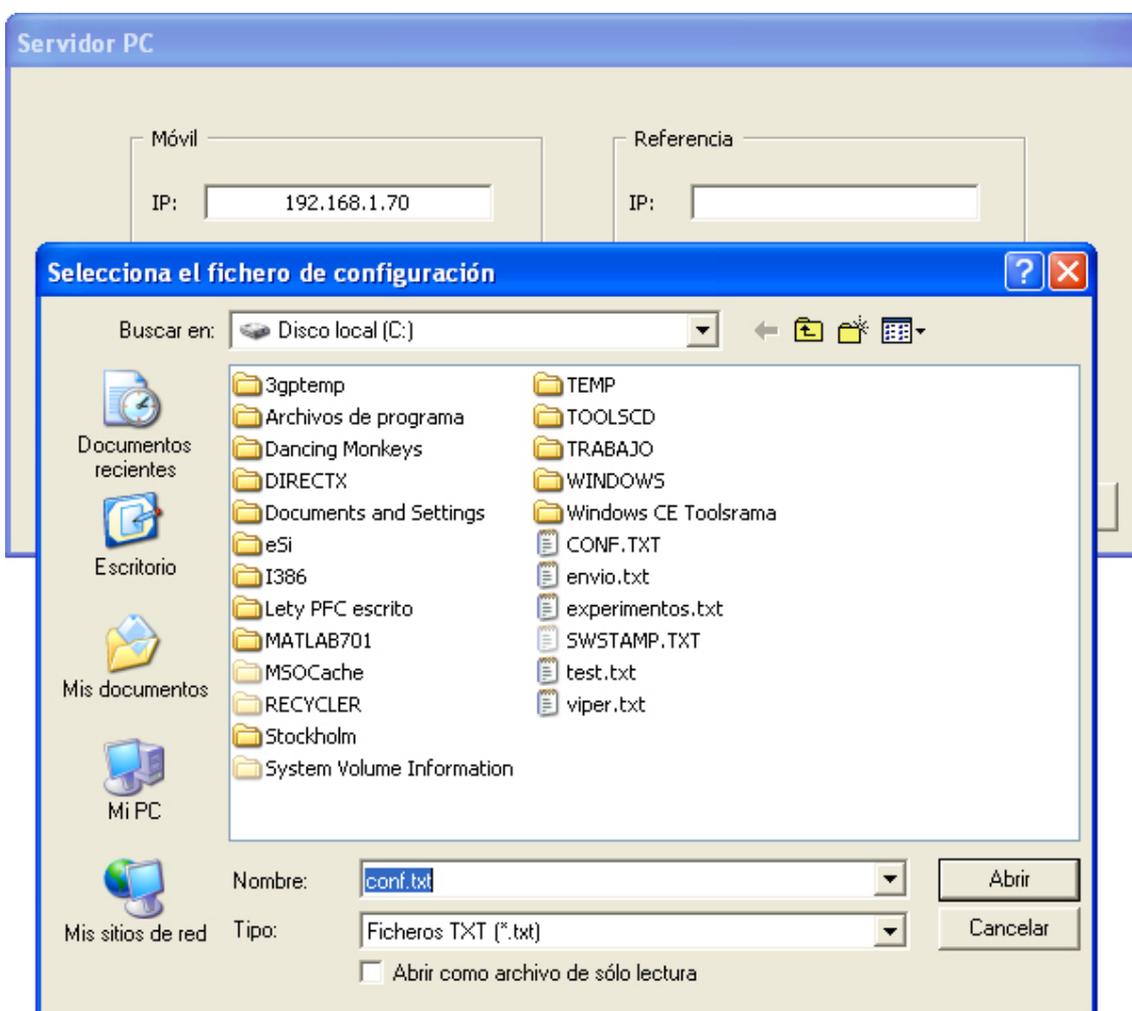
**Figura 2.1 Interfaz del monitor central**

Mediante un cuadro de diálogo se escoge el fichero que se desee enviar a cada Viper. Existen tres opciones posibles:

- Enviar a la Viper el fichero de configuración con la información necesaria para el correcto funcionamiento del sistema
- Enviar el fichero con la lista de experimentos disponibles

- Descargar los resultados obtenidos por la viper

En el capítulo 5 se explica con mayor detalle esta interfaz y su funcionamiento. En la Figura 2.2 puede observar el cuadro de diálogo que aparece al pulsar sobre el botón “Mandar configuración”.



**Figura 2.2 Selección del fichero de configuración en el monitor central**