

### **3 Instalación y puesta a punto**

#### **3.1 Elección del Sistema Operativo**

##### **3.1.1 Windows**

En un primer momento se instalaron sistemas operativos basados en Windows en el disco duro con el objetivo de familiarizarnos con la placa Hercules, básicamente para tratarla como lo que es: un pc embebido que funciona igual que un pc una vez conectados los diversos periféricos.

Se instalaron Win98, Win2000 y Win XP.

Entre las ventajas encontramos los siguientes campos:

- Facilidad de instalación
- Compatibilidad con el hardware
- Software conocido

Inconvenientes:

- necesita mucho espacio en disco
- poca configurabilidad
- escasas prestaciones en tiempo real

Debido a los inconvenientes no se continuo por el camino de windows y no se hizo ningún trabajo de programación, desechándose el uso de estos sistemas operativos.

##### **3.1.2 QNX**

En esta segunda fase se probó con el sistema operativo QNX, destinado a los sistemas en tiempo real. Tras una primera instalación en disco duro se desestimo su uso debido a los problemas que se encontraban con la incompatibilidad de hardware y la poca información existente, decantándonos por Linux debido a la gran cantidad de usuarios en todo el mundo así como por su distribución gratuita.

## 3.2 Linux

En esta tercera fase se trabajo en la instalación del sistema Linux.

Este cuenta con numerosas ventajas siendo estas:

- Distribución gratuita.
- Poco espacio en disco (tras configurarlo).
- Alta configurabilidad.
- Buena respuesta en tiempo real.
- Miles de usuarios en todo el mundo.
- Gran cantidad de información en la red.

Para la instalación de Linux distinguimos entre las diferentes fases.

### 3.2.1 Instalación de Red Hat Linux 9

La instalación de este sistema operativo en disco duro es relativamente sencilla debido al uso de un programa de instalación que nos guía a través de todo el proceso.

El sistema de archivos utilizado es el siguiente;

una partición ext3 donde guardamos los archivos de sistema (8Gb).

Una partición de intercambio Swap del doble de memoria que la Ram del sistema (500Mb).

Estas particiones las podremos hacer con partition magic, programa que podemos utilizar desde Windows si lo tenemos instalado previamente o desde disquete si no lo tenemos.

El gestor de arranque utilizado es GRUB, instalado en el MBR (master boot record).

Al intentar pasar este sistema operativo a la flash disk se encontraron numerosos problemas derivados del poco espacio que disponemos en ella (128Mb) y de lo abultado de la instalación en disco duro (2Gb).

Por lo que este Sistema Operativo fue desestimado en busca de una distribución de Linux mas antigua y que ocupase menos en el disco duro. La distribución elegida fue Debian Woody.

## 3.2.2 Instalación de Debian

Instalando la versión 3.0r4 de Debian (woody) en su forma mínima (kernel 2.2.20-adece) en el disco duro ocupa tan solo 90Mb.

El sistema operativo se consiguió desde la pagina de Debian:

<http://www.debian.org>

la instalación resulta sencilla y guiada.

El paso siguiente consiste en pasar el sistema de archivos a la flash disk, cambiando los datos de Lilo (el gestor de arranque de Debian) para que bote desde la flash.

El sistema de archivos utilizado es una partición en ext2 con / y bootable

### 3.2.2.1 Detalles instalación

#### 3.2.2.1.1 *Formateo flash*

Formateamos la flash disk antes de empezar a trabajar con ella. Utilizamos la orden:

```
fdisk /dev/hda
```

#### 3.2.2.1.2 *Instalación / desinstalación de módulo*

Para que el sistema operativo ocupe el mínimo espacio posible es necesario limpiar todos los módulos que no se utilizan. Como ayuda a instalar y desinstalar módulos se utiliza el modulo aptitude, el cual nos indica cuales están instalados, para que sirven y cuales son sus dependencias

Eliminamos;

Exim (correo electrónico)

Lpr (impresión)

Después de una instalación mínima es necesario instalar una serie de módulos como;

Gcc (compilar)

Jed (editor)

Aptitude

Less (editor)

Entorno grafico para el disco duro

Para la instalación de módulos podemos conectar con la pagina web de Debian y desde allí hacer la bajada de los archivos necesarios.

Para instalar un modulo en particular utilizaremos la orden;

Apt # instal=> instala el modulo #

### **3.2.2.1.3 Copiado de los archivos del disco duro a la flash**

Las carpetas que forman el sistema de archivos de Linux son copiadas del disco duro a la flash disk una a una con la orden;

Cp -a /bin /mnt/hda1=> copia todo lo que hay en bin al nuevo disco duro

Las carpetas que forman el sistema de archivos Linux son las siguientes;

bin

boot

dev

etc

home

initrd

lib

lost+found

mnt

opt

proc (esta carpeta no hay que copiarla!, debido a que contiene elementos utilizados por el sistema en el momento)

root

sbin

tmp

usr

var

vmlinuz

### **3.2.2.1.4 Archivos a modificar en el cambio de ubicación**

El archivo que hay que modificar al cambiar estos de ubicación es el archivo que controla el gestor de arranque: lilo

Para modificarlo usamos la orden;

```
jed lilo.conf
```

donde cambiamos los valores donde indica el lugar de arranque, pasando de hdc1 (el disco duro) a hda1 (la flash disk)

lilo.conf tendrá que quedar de la siguiente forma:

```
# Specifies the boot device. This is where Lilo installs its boot
# block. It can be either a partition, or the raw device, in which
# case it installs in the MBR, and will overwrite the current MBR.
#
```

```
boot=/dev/hda
```

```
# Specifies the device that should be mounted as root. (`/')
```

```
#
root=/dev/hda1
```

```
image=/vmlinuz
```

```
    root=/dev/hda1
```

```
    label=Linux-2.4.22
```

```
    read-only
```

```
image=/vmlinuz
```

```
    root=/dev/hdc1
```

```
    label=Linux-2.4.22
```

```
    read-only
```

después hay que ejecutar lilo para que los cambios tengan efecto: `./lilo`

así podremos seleccionar entre los dos sistemas operativos en las dos ubicaciones.

### 3.2.3 Instalación de RT-Linux

Para poder ejecutar una aplicación en tiempo real estricto necesitamos tener un Sistema Operativo que soporte el tiempo real. En Linux esto quiere decir que los programas se ejecutarán como módulos, en los que el modulo tomara el control del kernel y de las interrupciones y así estas no tendrán retrasos debidos a actividades “secundarias” o no de tiempo real.

Los detalles sobre la instalación se encuentran en el Apéndice B.

### 3.2.4 Conclusiones RT-Linux

Finalmente y con el Sistema Operativo instalado y los ejemplos de prueba funcionando se desestimo su uso debido a los problemas que surgían del uso de los programas de la Imu y de los programas del driver de la placa (DSCud).

Los programas de la Imu tienen que tener sus declaraciones en main, y en la forma de RT-Linux esto no se utiliza al ser módulos lo que se programa. Con tiempo y estudio estos programas podrían adaptarse para que funcionarán en tiempo real siendo el problema del tiempo el factor limitante para que finalmente fuera desestimado. Este campo seria uno de los posibles avances futuros para este proyecto.

### 3.3 Programas paso a paso

He creado una serie de programas “paso a paso” para ir familiarizándonos con el software de manera secuencial.

Estos se encuentran en el CD del proyecto en la carpeta / /programas\_paso a paso

Como primer programa de toma de contacto tenemos el programa P1.c que se compila con make P1 y se ejecuta con ./P1

este programa hará una lectura de los datos de la Imu y los escribirá por pantalla.

el segundo programa P2.c se compila con make P2 y se ejecuta con ./P2

este hace una lectura de los datos analógicos que le llegan a la Hercules y le hace una transformación analógica-digital, mostrándolos por pantalla

el tercer programa P3.c se compila con make P3 y se ejecuta con ./P3

en este programa se ejecuta una serie de interrupciones en las que se hace una lectura de los datos de la imu y de los canales analógicos de la Hercules, mostrando estos por pantalla.