

Instrucciones de instalación de un entorno de  
desarrollo en LINUX para el robot AIBO  
(version 1.0)

Nacho Herrero

Marzo, 2004

## Resumen

Este documento indica los pasos a seguir para la instalación de un entorno de programación para el robot *AIBO*. Dicho entorno está basado en el lenguaje *OPEN-R*, que es una subclase de C++, y el entorno *Tekkotsu*, que funciona por encima del anterior, y que incorpora una biblioteca de funciones basada en OPEN-R. También se indicará como instalar un entorno de desarrollo JAVA de SUN, necesario para algunos programas de *Tekkotsu*.

# Capítulo 1

## Instalación del S.O. Linux

Se utilizará la distribución **Mandrake 9.2**, disponible en RedIris (*ftp://ftp.-rediris.es/sites/carroll.cac.psu.edu/mandrake/9.2/*). El uso de cualquier otra distribución puede modificar algo los pasos y resultados de la instalación. La distribución RedHat parece haber dado algunos problemas a otros usuarios. Se supone que una distribución superior - Mandrake 10.0 - debería dar los mismos resultados.

La instalación se ha realizado en un portátil ASUS con tarjeta WIFI; se puede hacer sobre un PC fijo, pero será necesario dotarle de este tipo de tarjetas, ya que va a ser el principal medio de comunicación con el robot AIBO.

Para la instalación se puso en la BIOS el modo de arranque primario en CD-ROM, y se insertó el CD1 de la distribución Mandrake. Antes se reservó un espacio de 3Gb en el disco duro, mediante el Partition Magic.

La instalación de Linux es sencilla; hay que crear una partición SWAP de unos 256 a 512 Mb, y utilizar el resto del espacio libre (hasta los 3Gb) para la partición Linux. Es importante acordarse de instalar los paquetes de desarrollo en C y C++ (trae la versión 3.3.1), aunque esto se podría realizar también después.

Las instalaciones de SW las he realizado desde el usuario *root*. Se recomienda crear otro usuario para trabajar de forma "normal".

## Capítulo 2

# Instalación de OPEN-R

Se supone que se ha instalado el SW de desarrollo en C y C++ (compiladores, linkadores, makes,...).

Para la instalación de OPEN-R se necesita lo siguiente:

- *binutils-2.14.tar.gz*
- *gcc-3.3.2.tar.gz*
- *newlib-1.10.0.tar.gz*
- *build-devtools-3.3.2-r1.sh*
- *OPEN\_R\_SDK-1.1.5-r1.tar.gz*
- *OPEN\_R\_SDK-sample-1.1.5-r1.tar.gz*
- *OPEN\_R\_SDK-docE-1.1.5-r1.tar.gz* (contiene documentación)

La mayoría se puede obtener desde:

<https://openr.aibo.com/openr/eng/perm/openrsdk/download/download.php4>.

### 2.1. Proceso de Instalación

Arrancar como *root*, y llevar todos los ficheros necesarios al directorio */usr/local*. Abrir una ventana de comandos, dirigirse al directorio anterior, y ejecutar el fichero *build-devtools-3.3.2-r1.sh*, que crea un directorio */usr/local/OPEN\_R\_SDK* y compila las herramientas a utilizar para el desarrollo de programas (básicamente un compilador de c, una serie de librerías, y utilidades como *make* y similares)<sup>1</sup>.

Descomprimir el fichero de OPEN-R mediante *tar xvf OPEN\_R\_SDK-1.1.5-r1.tar.gz*. Se obtiene una carpeta */usr/local/OPEN\_R\_SDK*, con varias subcarpetas y ejecutables.

---

<sup>1</sup>Este proceso de compilación da muchos *warnings*, pero parece que no son importantes

## 2.2. Creación de los programas de proceso remoto (RP\_OPEN\_R)

Estos programas permiten que la ejecución de código se realice tanto en el perrito, como en un PC remoto que se comunica con el robot mediante WIFI (por tanto es imprescindible para nuestras necesidades).

Esta fase dio bastantes problemas, ya que al realizar la construcción (*make*) de los programas, se producían fallos y el proceso no llegaba a su fin. El problema es causado por la versión del compilador de C empleado en el sistema; la versión disponible (3.3.1) es demasiado moderna, por lo que necesitamos compilar con otra versión anterior. Una posibilidad sería modificar el fichero *make* para obligar a usar la versión 2.96, que también viene incorporada en Mandrake 9.2. Yo he preferido simplemente desinstalar los paquetes de la versión 3.3.1; realizar el *make*; y volver a instalar los paquetes anteriores.

1. Desinstala todos los paquetes de *gcc3.3.1*. Yo lo he hecho con el Centro de Control de Mandrake.
2. Si haces *gcc -version*, te debe decir que se trata de la versión 2.96.
3. Vete al subdirectorio *RP\_OPEN\_R*, debajo de *OPEN\_R\_SDK*, en una ventana de comandos. Allí, haz *setup-rp-openr*; te deberá instalar todo correctamente.
4. Reinstala las aplicaciones de desarrollo en C, C++, versión 3.3.1, mediante el Centro de Control de Mandrake (te pedirá los CDs de instalación).

## 2.3. Prueba de Funcionamiento

Para probar que todo va correctamente, descomprime *OPEN\_R\_SDK-sample-1.1.5-r1.tar.gz* en tu directorio de trabajo (el mío es */home/nacho/AIBO*; se creará una carpeta */sample*. Si nos metemos en un subdirectorio de esa carpeta (por ejemplo, */ERS7/BlinkingLED*) y hacemos *make install*, crearemos los programas BIN adecuados para cargar en el memory stick de AIBO. Estos programas quedan en */ERS7/BlinkingLED/MS/OPEN-R/MW/OBJS*.

## Capítulo 3

# Instalación de Tekkotsu

Tekkotsu es un entorno de desarrollo para AIBO, apoyado sobre OPEN-R, y que proporciona una biblioteca de primitivas y funciones de más alto nivel, permitiendo al programador concentrarse en las áreas específicas de interés del funcionamiento del robot (Para más información sobre sus posibilidades, mirar la documentación).

Para la instalación de Tekkotsu se necesitarán los siguientes ficheros:

- *j2re-1.4.2.04-linux-i586-rpm.bin*. Motor JAVA para Linux. (Al final no ha hecho falta)
- *j2sdk-1.4.2.04-linux-i586-rpm.bin*. Entorno de desarrollo JAVA para LINUX, edición estándar.
- *java3d-sdk-1.3.1-linux-i386.bin*. Biblioteca de funciones 3D del anterior.
- *Tekkotsu\_2.1.tar.gz*
- *Tekkotsu\_doc\_2.1.tar.gz*
- *Tekkotsu\_memstick\_ERS7\_2.1.tar.gz* (No lo he utilizado por ahora).
- *Tekkotsu\_mon\_2.1.tar.gz* (No hizo falta, se construyeron los ejecutables a partir de los fuentes).

Aunque Tekkotsu proporciona una biblioteca de funciones independiente de otros programas, también ofrece una serie de aplicaciones para el control, personalización, y monitorización del robot. Estos programas están desarrollados bajo JAVA o MATLAB. En nuestro caso, vamos a probar la opción soportada mediante JAVA.

### 3.1. Instalación bibliotecas y/o SDK de JAVA

La instalación de Mandrake 9.2 ofrecía la posibilidad de instalar un entorno de desarrollo JAVA (*gjc* y herramientas similares), pero los autores de Tekkotsu

recomiendan la instalación del paquete SDK *j2sdk-1.4.2\_04-linux-i586-rpm.bin* de SUN, que se puede obtener en <http://java.sun.com/><sup>1</sup>.

Para instalar el SDK (y motor) de JAVA, llevar el fichero *j2sdk-1.4.2\_04-linux-i586-rpm.bin* a cualquier sitio (por ejemplo */usr/local*), y ejecutarlo desde una ventana. Aparte de una información de licencia, se obtendrá un fichero *rpm*. Este fichero se puede instalar desde el Centro de Control Mandrake, y te lo instalará en la carpeta */usr/java/j2sdk-1.4.2\_04*. **Es necesario añadir la ruta de los ejecutables JAVA en la variable PATH.**

Para ello, editar el fichero */etc/bashrc*, y modificar la línea *export PATH=\$PATH* (o lo que tengas). Poner *export PATH=/usr/java/j2sdk-1.4.2\_04/bin:\$PATH*. Si tienes instalado el compilador de JAVA *gjc*, es importante que la nueva ruta se ponga antes del PATH antiguo; si no, compilará con el *gjc*, en vez de con el compilador de JAVA de SUN. Para probar si está todo bien, cerrar el terminal, abrir uno nuevo, y teclear *javac*. Si está todo bien, te saldrá un mensaje con información del uso del programa; si está mal te pondrá *gjc-3.3.1: no input files*.

Para probar que todo funciona, nos podemos ir a */usr/java/j2sdk-1.4.2\_04/demo/applets/ArcTest*, en una ventana de comandos; borramos los ficheros *cls*; y por último hacemos *javac ArcTest.java*. Se crearán de nuevo los ficheros *cls* que implementan la aplicación JAVA. Se puede ver que funcionan abriendo, en el mismo directorio, el fichero *example1.html* con el *mozilla*.

También nos interesa instalar la biblioteca de funciones 3D de JAVA, ya que algunas de las aplicaciones de Tekkotsu hacen uso de ellas.

Para ello, llevar el fichero *java3d-sdk-1.3.1-linux-i386.bin* al directorio */usr/java/j2sdk-1.4.2\_04*, y ejecutarlo allí en una ventana de comandos. Se puede comprobar que todo está correcto, compilando algún ejemplo de la carpeta */usr/java/j2sdk-1.4.2\_04/demo/java3D*, y ejecutando una página HTML que incluya el *applet*.

Ya tenemos el soporte básico para las herramientas de Tekkotsu; vamos ahora a su instalación.

## 3.2. Instalación de Tekkotsu

Descomprimir el fichero *Tekkotsu\_2.1.tar.gz* desde */usr/local*, con *tar -zxvf Tekkotsu\_2.1.tar.gz*. Te crea una carpeta *Tekkotsu\_2.1*. **Renombrar la carpeta anterior a */usr/local/Tekkotsu***. Dentro de esta carpeta hay un directorio *project*, que hay que copiar a tu lugar de trabajo (mi caso, */home/nacho/AIBO*). Luego, acceder a ese directorio en una ventana de comandos, y hacer *make*. El proceso da muchos *warnings*, pero acaba correctamente. Así generamos, tanto las herramientas y aplicaciones, como las bibliotecas y demás.

Ahora nos vamos a la carpeta */usr/local/Tekkotsu*, y modificamos el fichero *TARGET\_MODEL*, poniendo como única línea, *TGT\_ERS7*; esto permitirá que el código resultante se obtenga para el modelo ERS7 de AIBO. Volvemos a */home/nacho/AIBO/project* y repetimos el *make*.

---

<sup>1</sup>Es posible que solo hiciese falta instalar el motor JAVA (*j2re-1.4.2\_04-linux-i586-rpm.bin*) pero yo he decidido instalar el SDK que también incluye dicho motor

Dentro de ese directorio *project*, habrá que hacer posteriormente un *make install*, para generar los programas que se mandarán a la memory stick de AIBO.  
Con esto finaliza la instalación.