



GRUP DE RECERCA EN ENGINYERIA
DEL CONEIXEMENT (GREC)

Aibo Quickstart Manual

Internal Report, v 1.2

Ricardo A. Téllez

Agents Research Lab
Universitat de Girona

August 31, 2004

Contents

1 Introduction	2
2 Installing the OPEN-R SDK on the PC	3
3 Installing the Memory Stick Reader/Writer	3
4 Installing the OPEN-R base system on a memory stick	5
5 Setting the Wireless Network	5
6 Brief OPEN-R SDK Description	7
7 Other Aibo Programming Environments	8
8 Tips and Tricks	8
9 Interesting Links	9

1. Introduction

Development with Aibo is possible using the OPEN-R SDE. The OPEN-R SDE is a set of tools released by Sony for the development of programs for Aibo. It is composed by the OPEN-R SDK, the R-Code SDK, the Aibo Remote Framework and the Aibo Motion Editor. This document describes how to set up a research environment with an Aibo robot for the use of those tools, but it does not includes instructions about how to use the tools or how to program the robot. For those other purposes you can consult the official documentation or the other documents released by the same author of this document¹.

Aibo can be programmed by using the scripting language R-Code or the C++ environment provided by OPEN-R, but when using Aibo for complex control only the second option is used. Because of that, this document will focus on the creation of a development environment for the OPEN-R. OPEN-R is composed of libraries and sample programs that allow a complete control over the Aibo hardware using a C++ environment. Even that this development kit is mainly targeted to a Linux platform, it is also possible to use it on Windows platforms by installing the Cygwin application. The usual procedure to work with Aibo is the following: a program is created on the computer using OPEN-R and C++. This program is then cross-compiled to target the Aibo platform. Then the resulting binaries are transferred from the PC to a memory stick. The memory stick is then inserted onto Aibo.

To set up all the required environment, it will be necessary to install first the OPEN-R SDK on the PC. Then a installation on the PC of a memory stick reader/writer will also be necessary in order to transfer programs from the computer to the memory stick. Once the PC is able to use the memory stick reader/writer, it will be necessary to prepare a memory stick with the OPEN-R base system. The last step will be to configure the wireless network for communication between the PC and Aibo.

The used environment for the development of experiments with Aibo was composed of an Aibo ERS-220a equipped with a wireless LAN card, a computer running Windows XP and a wireless access point connected to the computer. A second environment was also proved to work perfectly, consisting of the new Aibo ERS-7 (which has a built in wireless card) and a Linux computer equipped with a wireless LAN card, working in Ad-hoc mode.



Figure 1: The Aibo ERS-220a and ERS-7 models

¹ For a description of the R-Code language read the *R-Code SDK Tutorial*. For a description of OPEN-R read *OPEN-R Essentials* both available at <http://www.ouroboros.org/~rt71592>

2. Installing the OPEN-R SDK on the PC

When starting to work with Aibo, the first thing to do is to visit the OPEN-R web page (<http://openr.aibo.com>) and download all the information and programs required. When accessing the page for the first time a registration will be required (a free one, though). After having the access code, the download section will give access to all the programs and documentation related to the SDE.

Download the following files for any operating system:

- OPEN-R SDK Documents English
- OPEN-R SDK for ERS-7/ERS-200 series
- Sample programs

If you are working with a Windows machine also download:

- Cygwin binaries
- MIPS cross-development tools for Cygwin

Download the following files if you are using a Linux machine:

- gcc source files
- binutils source files
- newlib source files
- Shell script for building cross development tools

At this point, you should first decompress the OPEN-R SDK Documentation package and start reading the *Installation Guide* document. This document will guide you on how to install the OPEN-R SDK under Windows or Linux, and it is very straight forward. The installation of the OPEN-R SDK onto your PC will allow you to use the OPEN-R libraries to create your Aibo programs, and also to cross-compile those programs generating Aibo binaries.

Special mention is required for the installation of Remote Processing OPEN-R. This is an additional package of OPEN-R that allows a computer to run some of the Aibo *processes* when the whole processing load is very high to be executed on Aibo itself. It allows the split of programs between Aibo and a computer host. To install this extra package just go to the `/usr/local/OPEN_R_SDK/RP_OPEN_R/bin` directory and execute the `setup-rp-openr` script. Do not install the package if you do not plan to use it.

3. Installing the Memory Stick Reader/Writer

Installing the memory stick under Windows is very straight forward and has no problem at all. Just plug the USB into the computer and play. Windows XP comes already with the required drivers. For other versions of Windows, the driver may be required. Just use the CD provided with the reader/writer.

Installation under Linux is a little more complicated. You will need to have your kernel compiled with support for USB. Also you should include *SCSI emulation* and the *usb-storage* module. Once this has been done you should follow the instructions given at the OPEN-R BBS (posted bellow)



Figure 2: The memory stick reader/writer

How to access Sony AIBO Memorystick using Dazzle USB Memorystick Reader under Linux.

Yu Wang (wangy22@mcmaster.ca)

INTRODUCTION

Dazzle Memorystick Reader allows people access information of a memorystick media, which is mainly produced by Sony, via USB port. This documentation shows simple steps of accessing the Sony AIBO Memorystick using this read under Linux operation environment.

It is tested in Redhat 9 on Acer Travelmate 332t laptop computer. The kernel version is 2.4.20-8.

MOUNTING THE MEMORYSTICK

It is recommended to plug the usb reader into the USB port before turn on the power of the computer, although most of the linux distributions support hotplug technology for USB devices. Make sure the Sony AIBO Memorystick is correctly plugged into the usb reader.

After log into the bash shell, make sure that the usb reader is recognized by the system. You can check it by typing:

```
$ cat /proc/bus/usb/devices | grep S:
```

and you will probably see something like this:

```
S: Product=USB OHCI Root Hub
S: SerialNumber=ccacb000
S: Manufacturer=SCM Microsystems Inc.
S: Product=eUSB MemoryStick Reader
S: SerialNumber=0000000011CF
```

Now we can mount the memorystick into the system by the following steps:

1. Create a empty directory to which be mounted, ex. /mnt/usb:

```
$ mkdir /mnt/usb
```

2. And then mount the memorystick (You must have supervisor privilege first)

```
$ mount /dev/sda1 /mnt/usb
```

USING THE MEMORYSTICK

After mount the memorystick to the specified directory(/mnt/usb), you can use it just like other storage device, such as hard drive. Most of the file system

command, such as `ls`, `mkdir`, `cat`, etc., can be applied.

When you "ls" the memorystick for the first time, you will find a file names `memstick.ind` already existing in the memorystick. DO NOT delete it, since this memorystick is to be used by the Sony AIBO robotic dog

4. Installing the OPEN-R base system on a memory stick



Figure 3: A memory stick

In order to be able to use a memory stick into Aibo and to accommodate your program, a basic system (Aibo's operating system) must be loaded into an empty one. Once the stick is prepared with that basic system, you will be able to set up the wireless network and transfer your program to it.

The preparation of the stick is a simple process: just copy some files provided by the OPEN-R SDK into an empty stick. Those files are the operating system of the robot. Three possible options of the operating system are possible: *Basic*, is a version of the operating system with all its capabilities but without wireless LAN environment; *Wlan*, with a wireless environment but no wireless console; *Wconsole*, with wireless environment and console (more about the console on section 8). On top of that, you must choose between an operating system with memory protection (*memprot*) where the operating system implements some control on memory accessing to prevent errors, or without it (*nomemprot*). You can find more information about it on chapter 4 of the *Programmer's Guide*.

Following your previous selection, copy the following directory into the root of the memory stick:

- For Basic environment with memory protection:
`/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS72/BASIC/memprot/OPEN-R`
- For Basic environment without memory protection:
`/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/BASIC/nomemprot/OPEN-R`
- For Wlan environment with memory protection:
`/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WLAN/memprot/OPEN-R`
- For Wlan environment without memory protection:
`/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WLAN/nomemprot/OPEN-R`
- For Wconsole environment with memory protection:
`/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WCONSOLE/memprot/OPEN-R`
- For Wconsole environment without memory protection:
`/usr/local/OPEN_R_SDK/OPEN_R/MS_ERS7/WCONSOLE/nomemprot/OPEN-R`

5. Setting up the Wireless Network

The wireless LAN network is not necessary in all cases, but it definitely will improve design speed by allowing direct transfer of new programs from the PC to the Aibo memory stick without having to take the stick out of Aibo. The wireless network can also allow more specialized experiments of remote monitoring and control of the Aibo pet.

Depending on the parts you have purchased, you can establish a wireless LAN by using an access point or by using just a simple wireless card on a PC.

Note: In all the configuration cases, it has been taken the 192.168.10.x network by default. This can be easily changed by specifying another network in the configuration files explained. Just be

² This path presumes the use of a ERS-7 model. For other Aibo models, change the ERS7 for ERS200

consistent with all the configurations to use the same network.

5.1.Communication through an Access Point (AP)

In this case the PC is connected to an AP through an ethernet cable. The AP connects via wireless with Aibo. This is called *managed* mode. In this case, the PC communicates with Aibo via the AP which is acting like a gateway. Things to set up are the PC, the AP and the Aibo wireless card.

5.1.1.Configuration of the Access Point

You must set the following parameters on the AP. Configuration of the AP depends on the model purchased, but most of them allow web based configuration. Check the AP instructions for that.

```
ESSID: AIBONET
WEP key: AIBO2
Wireless channel: any between 1 and 11
IP address: 192.168.10.1
Netmask: 255.255.255.0
Operation mode: router
```

5.1.2.Configuration of the PC

You can configure the PC using the typical *Control Panel* under Windows or the `ifconfig` command under Linux.

```
IP address: 192.168.10.2 (or any available in the same range)
Netmask: 255.255.255.0
Gateway: 192.168.10.1
```

5.1.3.Configuration of the Aibo wireless card

Wireless configuration of Aibo requires to modify a special file included in one directory of the memory stick. You should skip this step until you understand how the memory stick works. For this, you should read the *Programmer's Guide* document and section 4 of this document.

The wireless configuration file is `OPEN-R\SYSTEM\CONF\WLANDFLT.TXT` found in the directory structure of the memory stick. If you edit this file you will see easily how to configure the robot for wireless communication. Please, use the following data to configure accordingly with the AP and the PC:

```
HOSTNAME=AIBO
ETHER_IP=192.168.10.100
ETHER_NETMASK=255.255.255.0
IP_GATEWAY=192.168.10.1
ESSID=AIBONET
WEPENABLE=1
WEPKEY=AIBO2
APMODE=2 # this mode indicates auto-mode
CHANNEL=3
#DNS_SERVER_1=10.0.1.1
#DNS_SERVER_2=10.0.1.2
#DNS_SERVER_3=10.0.1.3
#DNS_DEFDNAME=example.net
```

Hint: the *auto-mode* of Aibo allows it to boot the wireless network on *managed* or *ad-hoc* mode, depending on the detection or not of a nearby AP. This means that, when Aibo is booting, if it detects an AP, it will boot in *managed* mode. Otherwise, it will boot in *ad-hoc* mode.

5.2.Communication without Acces Point

In this case, the PC communicates with Aibo directly by using a wireless LAN card. This is called *ad-hoc* mode. You should configure the PC wireless card and the Aibo wireless card.

5.2.1.Configuration of the PC

Use *Control Panel* under Windows, or `ifconfig/iwconfig` under Linux, to configure the wireless card with the following parameters:

ESSID: AIBONET

WEP Key: AIBO2

Wireless Channel: any between 1 and 11

IP address: 192.168.10.2 (or any available in the same range)

Netmask: 255.255.255.0

Communication mode: ad-hoc (also called Peer-to-Peer)

5.2.2.Configuration of the Aibo wireless card

Do the same as explained in 5.1.3.

6. Brief OPEN-R SDK Description

As shown in section 2, the SDK is mainly composed of a set of C++ libraries, some documentation files and a set of sample programs. Apart from reading the *Programmer's Guide* and the *Level2 Reference Guide*, sample programs are the best way to learn about OPEN-R. Sony doesn't have any specific OPEN-R programming tutorial but you will find a very good one at ENSTA (http://www.ensta.fr/~baillie/openr_tutorial.html) created by Jean-Christophe Baillie and François Serra. You can also find a similar tutorial by the author of this document at <http://www.ouroboros.org/~rt71592>

The procedure to run a program on Aibo is as follows: having an empty memory stick, you first copy to it Aibo's operating system. Then you go to the directory in your PC that contains the program you want to run on Aibo and generate the binary files (cross-compilation). Next, you must modify the wireless LAN settings if necessary. Last, you copy the generated binaries into the memory stick, and plug it into Aibo.

The best way to start is to test this procedure with one of the sample programs. To compile and install any of the sample programs do always the same: go to the directory in your PC that contains the program (for example, I would like to compile the HelloWord sample program, and I uncompressed the sample programs under the `/usr/local/sample_programs/` directory) and type `make install`. This will create the binary files and place them into the `/usr/local/sample_programs/HelloWord/MS/OPEN-R/MW/OBJS/` directory. Then transfer the contents of `/usr/local/sample_programs/HelloWord/MS/OPEN-R` directory to the memory stick (this transfer process must not replace the existing directory, but to merge with the existing files. This means, do not replace one directory for the other, but overwrite common files). If you want to modify the wireless settings, do it in the `/OPEN-`

R/SYSTEM/CONF/wlanconf.txt file of the memory stick.

Any OPEN-R program on Aibo is composed of objects running concurrently. When you create a program for the robot, you must create all the required objects for your control program. Those objects will run independently one of the others, but will communicate between them through connections in order to behave coordinately. Most of the sample programs are already valid objects for your own programs, so it is a good idea to start creating your programs by modifying the existing sample ones.

Note: you can use Remote Processing OPEN-R to execute some of the objects on a PC and the rest of object on Aibo, sharing in this way the CPU load that has to support Aibo's processor (for more information on this see the *Programmer's Guide*).

7. Other Aibo Programming Environments

Apart from using OPEN-R to program the robot, you can use other higher level programming environment created by third parts. I would mainly comment the following two:

- **The Tekkotsu environment** (<http://www-2.cs.cmu.edu/~tekkotsu/index.html>)

Created at the Carnegie Mellon University, it provides a programming framework built on top of OPEN-R. It allows the programmer to easily access and control any of the sensors/actuators of the robot. It also provides powerful processing tools combined with other programs like *Matlab* for image processing. Everything is Open Source and you can download it for free.

This environment also provides similar tools to those of Sony commercial packages like *Navigator* for free.

Main environment is Linux but it can be run under Windows using *Cygwin* (some bugs reported).

- **Yart/RCodePlus** (<http://www.aibohack.com/rcode/index.html>)

This is an environment built on top of the Rcode of Sony, that allows the easy creation of behaviors on Aibo by using a scripting language that boosts the power of the original Rcode. It is easy to use and program but functionality is limited by its easiness. However, it contains some interesting tools added, like remote control of Aibo, wireless consoles, etc.

8. Tips and Tricks

8.1. Printing to the wireless console

If you install in the memory stick the operating system version that has a wireless console (see *Programmer's Guide*), then you can telnet Aibo from a PC for monitoring and debugging purposes. Connection is accomplished by telnetting at the 59000 port of Aibo:

```
> telnet Aibo_IP 59000
```

You can include debugging messages in your Aibo code to appear in the telnet console. The allowed calls for messages are `OSYSPRINT()` and `OSYSDEBUG()`. The first one will print always the message indicated. The second one will only print on console the message if the debug flag was set during compilation.

8.2. Using the FTP server to transfer a program to AIBO without using the memory stick reader/writer

You can avoid the extraction and insertion process of the memory stick anytime you want to change

Aibo's program, by using an FTP server. The FTP server is an Aibo program provided by Sony on the samples directory, called TinyFTPD, that allows direct transfer of any new program from the PC to the Aibo memory stick, when the stick is in Aibo. It uses FTP protocol, so any typical FTP client can be used on the PC to transfer the files.

To use that program, you must first compile it and install it onto the memory stick. So, you must go to the TinyFTPD directory (`samples/TinyFTPD/`) and do `make install`. Then you will have to copy the generated object `samples/TinyFTPD/MS/OPEN-R/MW/OBJS/TINYFTPD.BIN` to the `/OPEN_R/MW/OBJS/` directory of the memory stick.

Next, you must add a line containing `/OPEN-R/MW/OBJS/TINYFTPD.BIN` to the file `/OPEN-R/MW/CONF/OBJECT.CFG` (of the memory stick).

Last, add the file `samples/TinyFTPD/MS/OPEN-R/MW/CONF/PASSWD` to the `/OPEN-R/MW/CONF/` directory of the memory stick.

When Aibo is running this new program, it will be possible to access Aibo's memory stick using any FTP client on the PC. Just type `ftp AIBO_IP`. The server will answer requesting a username and a password, where you can answer *anonymous* on both cases.

Once connected you can use the typical FTP commands to transfer, delete and copy new files to the Aibo memory stick. If you upload a new program to Aibo and what to reboot it, just type the following FTP command: `quote rebt`.

Notes:

- Remember to assign writing privileges to the configuration files `OBJECT.CFG` , `CONNECT.CFG`, etc, otherwise, the FTP commands will not be able to overwrite the files.
- **Be careful when rebooting Aibo, since if it is standing up, on rebooting it will fall down causing severe damages !!**

8.3.Hints

- To have a good overview of Aibo's capabilities, including physical limits, return values of sensors and output devices, consult the document *Model Information for ERS-xxx*
- Use `OSYSRINT` for debugging on a wireless console environment.
- Execute `make clean` after any modification in any of the `*.h` files, since it can clear some untraceable *segmentation fault* errors.
- It has to be taken into account that Aibo's MIPS processor is little-endian when transferring data to/from systems working in big-endian mode.

9. Interesting Links

Apart from the well known page of the OPEN-R SDK (<http://openr.aibo.com>) there are just a few good web pages related to the insights of Aibo. Nevertheless hundreds of pages can be found with no interest for research. Only the first type of pages are listed here:

- OPEN-R SDK Tutorial at ENSTA: a very good tutorial about OPEN-R available on the network http://www.ensta.fr/~baillie/openr_tutorial.html
- OPEN-R Essentials: another tutorial about OPEN-R based on the previous one <http://www.ouroboros.org/~rt71592>
- R-Code SDK Tutorial: the first tutorial on the net about R-Code programming

<http://www.ouroboros.org/~rt71592>

- Sony OPEN-R SDK school: 6 powerpoint tutorials describing some of the sample programs
<http://openr.aibo.com/> (OPEN-R SDK University section of the registered area)
- Aibo Hack: a web site with lots of hacks for Aibo
<http://www.aibohack.com>
- Aibo life bot house: an interesting forum where to learn and ask about Aibo, including technical details
<http://www.aibo-life.org/>
- Dogsboddy net: good extension programs for Aibo
<http://www.dogsboddy.net/>