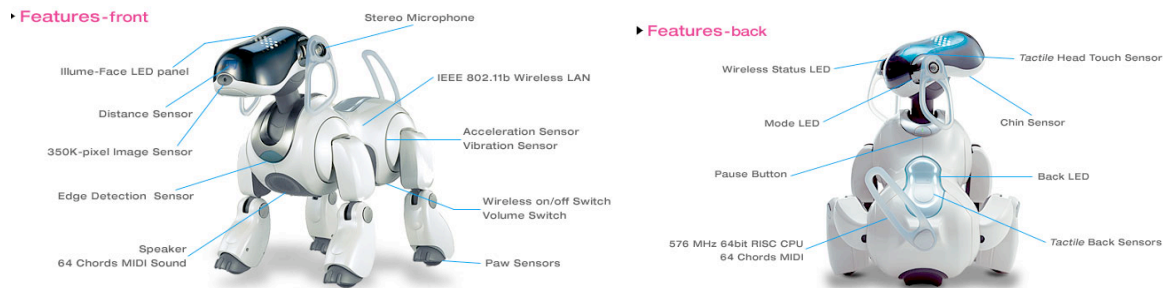


CS 1567  
Aibo/Tekkotsu – Lab 1

The purpose of this lab is to get you started with Aibo programming and the Tekkotsu framework. Before you start you should have some idea about the lay-of-the-land as far as Aibo programming is concerned. Here is a quick tour.

**Aibo Hardware** – From the programmer view, the aibo consists of a 64-bit RISC processor with a single 64MB memory space. All process threads (called behaviors) share the same memory and a common set of objects that encode the current state of the robot. The robot itself has an impressive array of sensors, actuators and output media for the price. The sensors and I/O devices locations are shown in the images below. They can be grouped as follows.



Sensors -

- 320 line CCD camera with color segmentation hardware
- IR distance sensors (2 on ERS-7), one on ERS 210
- Position encoders for all joints actuator motors (see below)
- Stereo Microphone
- Touch sensors (switches) head (1), back (3/ers7, 1/ers210), chin
- Accelerometer for detecting falls and pick-ups
- Switch sensors on each paw pad

Actuation motors and Outputs

Joint actuation motore (one per parenthesis)

Tail: (left/right)

Neck: (up/down)

Head: (left/right) (up/down)

High shoulder – per leg: (in/out) (forward backward)

Knee, (bend position)

Ankle (bend position)

Audio channel

Networking and I/O

803/11g – wireless network channel

removeable memory stick slot.

**Aibo Software** – The aibo ships with a fairly sophisticated native software application for “pet emulation”. This application boots each time the dog is turned on with no memory stick in the slot. When a memory stick is inserted, the aibo will boot into a user application. Sony provides two layers of software support for these applications. At the lowest level is the APERIOS operating system. Access to APERIOS is supported through the OPEN-R software developers kit (SDK) layer. OPEN-R is free for registering at the Sony AIBO developers site <http://openr.aibo.com/>. Please note that there are different versions of OPEN-R for the ERS-7 and ERS2xx models. Please make sure that you download the correct version.

In this class we will be using a programming framework called Tekkotsu (Japanese for ‘iron bones’) developed by CMU under a contract from Sony. In addition to hiding some of the uglier aspects of OPEN-R programming, Tekkotsu provides two advantages over directly programming OPEN-R. First, it provides a hierarchical model in which behaviours are decomposed into motions with the lowest level details of motion execution handled within Tekkotsu. Second, it provides a convenient remote host interface that allows the user to control and manage multiple behaviors in both foreground and background execution modes. A large part of this first lab, will be running through a few tutorials on the Tekkotsu web site <http://www.tekkotsu.org/>. Bookmark this web site, you will use it quite a bit in the next few weeks.

Getting Started –

We have pre-loaded a copy of Tekkotsu on OPEN-R on the server and each of the laptops in the lab. You may choose to download your own copies if you want to use a personal laptop.

**TASK 1 – Starting the default Tekkotsu environment**

- 1) In your home directory, make a local copy of the default project directory located at `$TEKKOTSUHOME/project`.
- 2) Depending on which dog you are using, rename the makefile (or symbolic link) for the `Makefile.ERS7` or the `Makefile.ERS210` to `Makefile`.
- 3) Read the first section of the Makefile w.r.t. the environmental variables that you need to set. Set them appropriately for your installation using whatever commands are appropriate to the your unix shell. The defaults settings for the lab laptops and server will be on the whiteboard in the labs.
- 4) Also add `$TEKKOTSUHOME/tools/mon` to your `PATH` variable, and set the `CLASSPATH` variable for java to `$TEKKOTSUHOME/tools/mon`.
- 5) Type `make install` to build your own memory-stick image of the default Tekkotsu environment.
- 6) Copy the memory stick image from the directory you specified in the makefile to the memory stick writer. Make sure that you use the `-Rfp` flags for the copy.
- 7) Unmount the memory stick, remove it from the reader, and insert it into the aibo. Power up the aibo. If things work successfully the aibo will growl after about 30 seconds.

- 8) After the aibo growls, type ContollerGUI <dogname> on your host machine. After a few seconds, the controller interface will appear on your screen, connected over the wireless interface to your dog. Explore the menus and behaviors provided as demos within Tekkotsu. These will be important programming examples for your own work.

#### TASK 2 – Run the Tekkotsu “first behavior” tutorial

- 1) Within the project directory that you just created, build the sample behavior shown in the “Writing your first behavior” tutorial on the Tekkotsu web page.
- 2) The only difference will be that you will add your behavior to the “mode” submenu rather than the main menu. (see Startup\_SetupMode.cc)
- 3) Run the makefile and memory stick installation the same way as in the previous task.
- 4) Your new behavior will appear in the Mode submenu of the contoller when you restart the aibo with the updated memory stick.

#### TASK 3 – Writing an original behavior

- 1) Add the cs1567BaseBehavior.cc and .h files your project directory.
- 2) Analyze this code, add it to your behaviour menu and run it if you like. It is a template behavior that starts the dog in a sitting position when any button is pressed.
- 3) Add additional the following additional motions to this behavior from the built-in motions provided with Tekkotsu. These motions are coded in C++ and can be found in the \$TEKKOTSUHOME/Motion. Motion command files are of the form <motionname>MC.cc At a minimum you should add tail wagging and head movement to this behavior.
- 4) Use the sound manager to add barking.
- 5) Using the information returned from the event manager, modify your behavior to respond with different sounds and/or motions based on particular buttons. For example, bark if the back is touched, whimper for chin, etc.

#### TASK 4 – design your own

- 1) In the next lab you will implement a complex set of behaviors. So it is important that you take this opportunity to learn all you can about Tekkotsu and programming motions and behaviors.
  - Try adding other motions to your behavior.
  - Explore the motion-sequence and poster motion commands.
  - Make sure that you understand the event manager and how to turn on/off listeners and how to distinguish between events.