

# Quick guide to software for the AIBO

October 10, 2006

This is meant to give a brief overview of each software package, and highlight some of the differences. It should help you choose which environment to use.

## 1 R-CODE

### 1.1 Description

R-CODE is a relatively simple interpreted scripting language which runs on the AIBO robot. The code is simply a text file, which can be created with any text editor.

### 1.2 Pros

- Simple to learn
- No compilation
- Good for those without much C++ experience
- Much less complexity

### 1.3 Cons

- Not suitable for processing data
- Cannot access many of the low level functions, such as individual LEDs, wireless LAN, some sensors.

## 2 OPEN-R SDK

### 2.1 Description

A C++ like API for low level access to the robot and its hardware. Intended for use with gcc under linux, but can be used under cygwin in windows.

Tutorial/Introduction: <https://openr.aibo.com/openr/eng/perm/openrsdk/university2.php4#SDKSCHOOL>  
[http://www.ensta.fr/%7Ebaillie/tutorial\\_OPENR\\_ENSTA-1.0.pdf](http://www.ensta.fr/%7Ebaillie/tutorial_OPENR_ENSTA-1.0.pdf)  
<http://www.cc.gatech.edu/tucker/courses/amrs/aibo/AIBOProgrammingTutorial.pdf>

## 2.2 Pros

- Most access to the robot's hardware
- Least overhead
- Users wanting to design their own control architecture will like this

## 2.3 Cons

- Complex to learn compared to rcode
- Must use low level APIs (may have to do your own kinematics if your project involves certain things)

# 3 Remote Processing Framework

## 3.1 Description

A set of Visual C++ libraries to write applications that can remotely control the AIBOs. An example is the entertainment player software demoed in class.

## 3.2 Pros

- Can design a flashy GUI
- Users who really like Visual C++ can use it
- Uses the PC's memory and processing power
- Can get audio/video/sensors back to PC for processing
- Can use built in robot motions

## 3.3 Cons

- Not truly "robotics" in some minds
- Less access to the hardware than with OPEN-R SDK
- Restricted to Windows

# 4 Layered Environments (built on OPEN-R SDK)

## 4.1 URBI

Control architecture over wireless LAN. URBI is a scripting language that can be used both to write programs on the memory stick and to send commands to the AIBO via the wireless network.

Info: <http://uei.ensta.fr/baillie/eng/urbi.html>

## 4.2 Pyro

A Python version of the remote control architecture, layered on Tekkotsu

## 4.3 CMPack

### 4.3.1 Description

A framework developed at CMU mainly for robot soccer, but used in their class with AIBOs. This seems to be very full featured, including routines that do the needed kinematics for the user.

Overview:

<http://www-2.cs.cmu.edu/robosoccer/cmrobotbits/lectures/system-overview.pdf>

### 4.3.2 Pros

- Kinematics done for you
- Nice control structures
- Hides some complexity of the hardware

### 4.3.3 Cons

- Introduces new layers of complexity
- Multiple software layers (slower?)
- Not well documented

## 4.4 Tekkotsu

### 4.4.1 Description

This is similar to CMPack, also developed at CMU (apparently they are quite separate projects however). This is more general than CMPack.

Tutorial/Introduction: <http://www-2.cs.cmu.edu/%7Edst/Tekkotsu/Tutorial/intro.shtml>

### 4.4.2 Pros

- Very well documented