AIBO Remote Framework (No Programming)
Story 4. Using Recognition Functions with RemoteTest

You can enjoy AIBO Remote Framework by using the sample programs.
A development environment (VC++) is not needed.
What’s Needed

- PC with Windows Xp or 2000
- Memory Stick Reader/Writer
  Or a PC that has a Memory Stick reader/writer (VAIO PC)
- AIBO Programming Memory Stick
- AIBO ERS-7
  or
  AIBO ERS-7M2
  Wireless LAN is built-in
- AIBO MIND2 Memory stick
  or
  An Upgraded Memory Stick to AIBO MIND2 from
  AIBO MIND
- Access Point (supports IEEE802.11b)
  It is more stable to use a Wireless LAN Access point even if your PC has a wireless LAN adapter.
Preparation

• See Story 1 to setup the Wireless LAN and AIBO Remote Framework.
Open Remote Test

1. Open RemoteTest.exe. It should be placed on the desktop as a shortcut.

2. Input AIBO’s IP address and click the Connect button.
3. The “AIBO Output” window pops up. This window shows the information coming from AIBO.
4. Check that AIBO’s mode lamps (above the ears) turn Blue, then click the Remote button.
5. You will hear a sound when AIBO switches to Remote mode. AIBO’s mode lamps will blink blue.
Pop-up Dialog Windows to Control AIBO

6. Click “Posture”, ”Walk”, and ”Motion” buttons to pop-up each dialog.
Pop-up Dialog Windows to Start Recognition Functions

7. Click the “Autonomy services” button to pop-up the dialog window.
Start All Recognition Functions

8. When AIBO is controlled in remote mode, all recognition functions start disabled, except for cliff recognition. Click ON to start each recognition function.
Watch the Results

9. The result of each recognition is sent to the PC as “Input semantics”. In the RemoteTest sample, the result can be watched in the “AIBO Output” window. Show your AIBO the AIBONE. Can you see the letters “BONE”? AIBONE is also pink, so “PINK” is also recognized.
Watch the Results

10. The AIBONE may be moved by you, so AIBO recognizes it as a moving object. “MOVE_OBJ” will be shown in the window. When AIBO hears a voice, “VOICE_BEGIN” may be detected. If the voice is loud, “LOUD” may also be detected.
Details of the Results

11. The detailed information is reported in 4 additional variables; SubInfo[0..3]. In the case of “BONE”:

```
SEMID_BONE (0x010a) // AIBONE
// SubInfo[0] = horizontal position (degrees; left is positive)
// SubInfo[1] = vertical position (degrees; up is positive)
// SubInfo[2] = estimated distance(cm) to the AIBONE (calculated by size and shape)
// SubInfo[3] = angle/state of AIBONE
//   = (int) (((int)state)&0xffff)<<16) + (((int)angle)&0xffff);
//   angle : AIBONE angle, relative to AIBO's view.
//   A horizontal angle (perpendicular to AIBO's field of view) is 0;
//   other values range between -90 and 90, where positive values are clockwise.
// state : Mask 0x0001 1 : AIBONE is standing up 0 : AIBONE is on its side
//       Mask 0x0002 1 : AIBO is looking at AIBONE's shaft 0 : side
```

You can get some information about the AIBONE from SubInfo[].
Ex) AIBO sees the AIBONE’s shaft or side (triangle shape)?
The shaft is standing or is it on its side? Distance? Location?

“Input semantics” is described in the document
¥docE¥HTML¥API Spec¥Semantics.txt
Voice Recognition

12. Say “Dance” to AIBO. You will see some semantics in the AIBO Output window. Please scroll and search for the “VOICE” line. (It is not “VOICE_BEGIN”)

SubInfo[0] specifies what voice is recognized in the lower word. In the left window.
SubInfo[0] = 0x00010017
So ID=0x0017, AIBO recognizes the voice command “dance”.
The word list is in Semantics.txt

Upper word 0x0001 means the word is a favorite. 0x0001=favorite 0x0002=Not favorite.
If AIBO recognizes “Don't do that”, then SubInfo[0] = 0x0002002B and the lower word is ID=0x2B, so the word is not a favorite one.

SubInfo[1] gives the angle the sound is coming from. There is owner information in SubInfo[2]. SubInfo[3] is for internal use.

AIBO has many recognition engines and the results are sent to the PC as “Input Semantics”.

See Semantics.txt to know what “Input Semantics” comes from AIBO.
Recognize an AIBO Card

13. Show AIBO an AIBO card. Search for “ECARD1” in the output window. (“MOVE_OBJ” will be reported because the card was moved. The upper word of SubInfo[0] is the CARD ID. In the figure, SubInfo[0]=0x00090007, so the upper word is 0x0009, so the card is number 9 (ECARD_I). See semantics.txt to know what CARDIDs can be recognized.

14. Show AIBO the station pole. “EPOLE” semantics will be shown.
Search and Track a Pink Ball

15. Make AIBO search and track a pink ball. Click the “Kick/Header...” button in the main dialog to pop-up the “Search, Track, Kick, Heading” dialog.
16. Target “PINK” is easy for AIBO, so set Target to “PINK”. Click “Search & Track Target (Normal).” Move the Pink ball in font of AIBO. AIBO will find the ball and follow it with its head. (tracking)
17. If AIBO looses the pink ball, tracking is done. Click the “Search & Track Target (Normal)” button if you want AIBO to track the pink ball again.
18. Using the visual pattern recognition (VPR) function. Click the “VPR” button to pop-up the VPR dialog. To see what AIBO is seeing, open the ImageTP dialog. To control AIBO’s head, open the “Posture/Neck angle” dialog.

19. Move AIBO’s head to let AIBO see the picture that you want to teach AIBO. VPR remembers the image pattern, so show something with a detailed pattern.
Visual Pattern Recognition

20. Click the “Train model” button in the VPR dialog. “Complete” will be shown in the “AIBO Output” window. After that, AIBO recognizes the pattern and “EPTN” will be shown just after complete. (If “EPTN” is not shown, the pattern is too plane to recognize. Change the object to one with a detailed pattern.)

21. Hide the object from AIBO’s camera. (“EPTN” will be shown so that you know it is gone) Let AIBO see it again, if the AIBO recognize it, “EPTN” will be shown in the “AIBO Output” window.
Visual Pattern Recognition

22. Teach AIBO some visual patterns. Model sub ID can be used to categorize the same object for different views (font and back side of a book). But you can use a new Model ID to teach a new pattern. The model ID that AIBO recognizes is in SubInfo[0] of the “EPTN” Input Semantics.
Summary

How about AIBO Remote Framework? In AIBO MIND2, these recognitions work all the time and control AIBO itself. Did you notice that AIBO’s emotion information is reported with recognition and sensor information?
AIBO’s emotion is changes with the environment that AIBO recognizes.

+ from Information Converter (Extraction)
  SEMID_BACK_M    ( 0x0821 ) // Back sensor middle
  SEMID_BACK_R    ( 0x0822 ) // Back sensor rear
  SEMID_SLIDE_FMR ( 0x0827 ) // Back sensors are stroked to the rear from the front
  SEMID_SLIDE_RMF ( 0x0828 ) // Back sensors are stroked from the rear to the front

+ from Instinct and Emotion Generator (Extraction)
  SEMID_APPETITE  ( 0x0a07 ) // Charging desire (Instinct level for autonomy) Value = SubInfo[1] 0-100
  SEMID_EXERCISE  ( 0x0a08 ) // Movement desire (Instinct level for autonomy) Value = SubInfo[1] 0-100
  SEMID_AFFECTION ( 0x0a09 ) // Affection desire (Instinct level for autonomy) Value = SubInfo[1] 0-100

In the RemoteTest sample, “Input Semantics” is only shown in the “AIBO Output” window.
You can make an interesting PC application using the result of each recognition.

Next step, do you want to try making your own program?