



Programmazione Aibo

Gilberto Decaro



Configurazione Aibo



- Systema Operativo: Aperios, Real Time sorgenti non disponibili.
- Supporto per dati e programmi: Aibo Programmable Memory Stick, rosa, da 8 o 16MB



Configurazione scheda WiFi



WLANCONF.TXT

(in /MS/OPEN-R/SYSTEM/CONF)

```
HOSTNAME = AIBO
ETHER_IP = 172.16.1.192
ETHER_NETMASK = 255.255.255.0
IP_GATEWAY = 172.16.1.180
ESSID = Aibonet
WEPENABLE = 1
WEPKEY = AIBO2
APMODE = 2
CHANNEL = 3
#DNS_SERVER_1=10.0.1.1
#DNS_DEFDNAME=example.net
#USEDHCP = 1
```



WLANCONF.TXT



- **HOSTNAME:** il nome dell'Aibo in rete (massimo 8 caratteri).
- **ETHER_IP:** indirizzo IP dell'Aibo (da impostare nel caso non venga utilizzato il DHCP).
- **ETHER_NETMASK:** specifica la subnetmask dell'Aibo.
- **IP_GATEWAY:** l'indirizzo IP del Gateway.
- **ESSID:** il nome della rete Wireless a cui accedere (massimo 32 caratteri).
- **WEPENABLE:** 0 disattiva la cifratura dei dati WEP, 1 la attiva.
- **WEPKEY:** chiave di cifratura da utilizzare nel caso WEP sia attivato. E' supportato il WEP64 (chiave a 40 bit o 5 caratteri) e WEP128 (chiave a 104bit o 13 caratteri).
- **APMODE:** specifica la modalità di rete Wireless da utilizzare: 0 per la modalità AdHoc, 1 per la modalità Infrastructured, 2 per il rilevamento automatico.
- **CHANNEL:** canale di comunicazione Wireless da utilizzare nella modalità Ad Hoc (sono permessi i canali compresi tra 1 e 11); per usare più Aibo in modalità Ad Hoc è necessario configurare entrambi sullo stesso canale.
- **DNS_SERVER:** specifica l'IP del server DNS da utilizzare (da specificare se non viene utilizzato il DHCP).
- **DNS_DEFDNAME:** specifica il nome di dominio di default.
- **USEDHCP:** 0 disabilita il DHCP, 1 lo abilita.



Tool di Sviluppo (della Sony)



- **R-CODE** : linguaggio di Script, poco versatile, facile da usare.
- **Remote Framework**: codice di controllo su PC (visual C++), comanda l'Aibo via Wireless, poco utile per lo studio di sistemi Realtime.
- **Open-R**: SDK in C++, programmazione concorrente di oggetti eseguiti in parallelo, molto potente, difficile da usare.



Altri Tool di Sviluppo



- **Tekkotsu** (www.tekkotsu.org): framework basato su Open-R, scrittura di *behaviors*, molto potente.
- **URBI** (Universal Robotics Body Interface) (urbi.sourceforge.net): linguaggio basato su reti client/server via Wireless; comandi via Telnet o con programmi in C++ o Java.



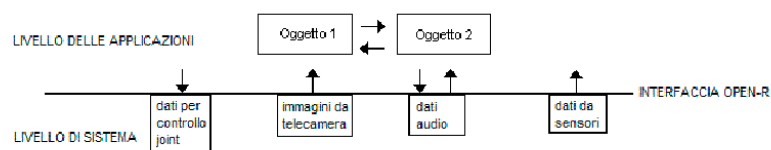
Open-R



- Basato sul C++ di Linux.
- Esecuzione concorrente di oggetti.
- Oggetti simili a processi Unix.
- Scambio di messaggi tra oggetti Open-R.



Oggetti Open-R



- Oggetti Open-R simili ai processi di Unix, eseguiti in concorrenza e in parallelo.
- Oggetti di sistema (OVirtualRobotComm, OVirtualRobotAudioComm, IPStack)
- Oggetti creati dall'Utente



Comunicazione inter-object



- Scambio di messaggi (tipi predefiniti, struct, array)
- Memoria condivisa (classe *RCRegion*)
- Ogni oggetto offre dei servizi (*gate*):
- Servizi subject: spediscono il messaggio
- Servizi observer: ricevono un messaggio
- Sintassi dei servizi:

```
<nome oggetto Open-R>.<nome del servizio>.<tipo di  
dato scambiato>.<O se observer; S se subject>
```



Il file stub.cfg



- Dichiarare i servizi offerti dall'Oggetto (in compilazione).

```
ObjectName : WakeUp  
NumOfOSubject : 1  
NumOfOObserver : 1  
Service : "WakeUp.ReceiveCommand.char.O", null,  
ReceiveCommand()  
Service : "WakeUp.Move.OCommandVectorData.S", null,  
Ready()
```



Classi Open-R per la comunicazione inter-object



- Subject
 - Classe OSubject
 - Classe OReadyEvent
- Observer
 - Classe OObserver
 - Classe ONotifyEvent



Entry Point



- Tipi di Entry Point di un oggetto Open-R:
 - Standard: *DoInit()*, *DoStart()*, *DoStop()*, *DoDestroy()*.
 - Dei servizi: *ReceiveCommand()*.
 - Extra Entry Point.



Il file CONNECT.CFG



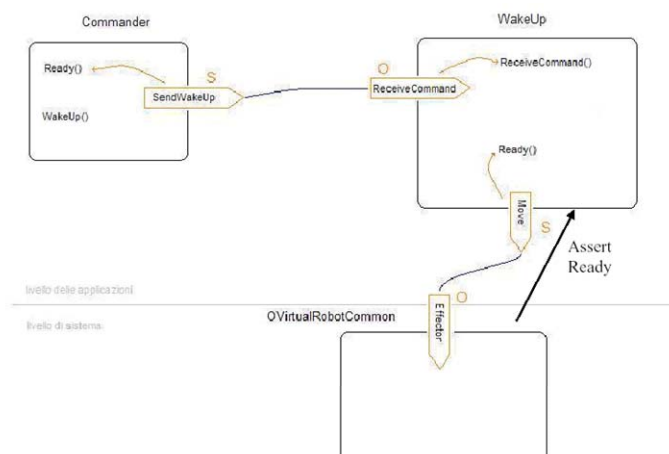
- Definisce i canali di comunicazione tra gli Oggetti eseguiti mettendo in relazione i vari servizi.
- Caricato all'avvio.

```
Commander.SendWakeUp.char.S WakeUp.ReceiveCommand.char.O
```

```
WakeUp.Move.OCommandVectorData.S  
OVirtualRobotComm.Effector.OCommandVectorData.O
```



Entry Point e scambio di Messaggi





Debugging



- Macro: OSYSDEBUG();
 - telnet <ip aibo> 59000
 - -DOPENR_DEBUG
- Eccezioni: emon.log
 - Analisi del dump dei registri/memoria
 - .../sample/common/Crush/util/emonlogparser
emon.log <oggetto.nosnap.elf>



Installazione Open-R



- E' disponibile per ambiente Linux, Window e MacOS.
- In Linux (installazione locale):
 - Creazione del Crosscompiler MIPS.
 - Installazione di Open-R
 - Copia dei Sample e Documentazione.



Utilizzo di Open-R



- **Compilazione:**
 - `make OPENRSDK_ROOT=/home/usr/local/OPEN_R_SDK/`
- **Installazione in MS:**
 - `make OPENRSDK_ROOT=/home/usr/local/OPEN_R_SDK/ install`
- **Cancellazione:**
 - `make OPENRSDK_ROOT=/home/usr/local/OPEN_R_SDK/ clean`



Rete in aula Delta



- IP degli Aibo: 172.16.1.192 /3 /4 /5
- IP Access Point: 172.16.1.180
- MemoryStick: mount point definito in pcd, pc02, pc03



Copia della MemoryStick



- Copia di MS dal proprio PC:
 - cd MS/
 - scp -r * user@pc02:gruppo<x>
- In pc02 o pc03
 - cd gruppo<x>
 - mount /mnt/stick
 - cp -R * /mnt/stick
 - umount /mnt/stick