



Robotica ed Animazione Digitale



Prof. Alberto Borghese
Dott. Gilberto Decaro
Dipartimento di Scienze dell'Informazione
borgnese@dsi.unimi.it

Università degli Studi di Milano



Sommario dell'esercitazione



- **Preparazione dell'SDK – OpenR.**
- Oggetti OpenR e comunicazione inter-object.
- La libreria Cmotion di Carnegie Mellon
- Esempi di generazione di movimenti e debugging.
- Sperimentazione a gruppi.



Open-R SDK



- <http://openr.aibo.com/>
- E' necessario registrarsi per poter:
 - ◆ Scaricare l'ambiente di sviluppo (SDK)
 - ◆ Leggere la BBS di Open-R
- Nell'area Download sono presenti:
 - ◆ OPEN-R SDK
 - ◆ R-CODE SDK
 - ◆ AIBO Remote Framework
 - ◆ AIBO Motion Editor



Da cosa è composto?



The OPEN-R SDK is a cross-development environment based on gcc (C++) with which you can make software that works on AIBO (ERS-7, ERS-210, ERS-220, ERS-210A, and ERS-220A).

- Open-R è basato sui tool di sviluppo base forniti con Linux (da questo il nome Open):

–**gcc**: compilatore C/C++ GNU

–**binutils**: utility per la gestione dei file (principalmente librerie).

–**newlib**: librerie standard C sviluppate da *RedHat* per sistemi Embedded

- Scopo dell'installazione è creare un ambiente di *cross-developing* per processori Mips in ambiente Linux.



OS supportati



- Essendo basati su tool GNU distribuiti con GPL, è possibile creare un ambiente di sviluppo per i seguenti sistemi:
 - ◆ **Linux**: standard
 - ◆ **Windows**: con emulatore
 - ◆ **MacOs**: con emulatore



Utilizzo in ambiente Windows



- E' necessario creare un ambiente "*Simil-Linux*".
- Viene adottato Cygwin (www.cygwin.com) un ambiente Linux per windows:
 - ◆ Layer di emulazione della API di Linux in ambiente Windows (cygwin1.dll)
 - ◆ Collezione di tool standard presenti in una distribuzione Linux.

Per l'installazione dell'SDK sotto Windows occorre scaricare:

- E' necessario scaricare i seguenti pacchetti (tutti disponibili in openr.aibo.com):
 - ◆ cygwin-packages-1.5.5-bin.exe
 - ◆ mipsel-devtools-3.3.2-bin-r1.tar.gz
 - ◆ OPEN_R_SDK-1.1.5-r5.tar.g
 - ◆ OPEN_R_SDK-docE-1.1.5-r1.tar.gz
 - ◆ OPEN_R_SDK-sample-1.1.5-r2.tar.gz



Installazione in ambiente Windows



- Passi da seguire:
 - (1) Decomprimere `cygwin-packages-1.5.5-bin.exe`
 - (2) Eseguire `setup.exe` nella directory decompressa
 - (3) Copiare i vari `OPEN_R_SDK-*` e `mipsel-devtools-3.3.2-bin-r1.tar.gz` nella directory in cui installato Cygwin
 - (4) Spostarsi in `/usr/local` e decomprimere i vari `OPEN_R_SDK` e `mipsel-devtools-3.3.2-bin-r1.tar.gz`



Installazione in ambiente Linux - I



- Sono necessari i seguenti pacchetti (tutti disponibili in openr.aibo.com):
 - ◆ `binutils-2.14.tar.gz`
 - ◆ `gcc-3.3.2.tar.gz`
 - ◆ `newlib-1.10.0.tar.gz`
 - ◆ Lo script `build-devtools-3.3.2-r1.sh`
 - ◆ `OPEN_R_SDK-1.1.5-r5.tar.g`
 - ◆ `OPEN_R_SDK-docE-1.1.5-r1.tar.gz`
 - ◆ `OPEN_R_SDK-sample-1.1.5-r2.tar.gz`



Installazione in ambiente Linux- II



- La prima riga di build-devtools specifica la directory di installazione

```
build-devtools-3.3.2-r1.sh
File Edit Options Buffers Tools Insert Help

# SDK.
#
# binutils-2.14
# gcc-3.3.2
# newlib-1.10.0
#
# Run the script in the directory where the source packages (.tar.gz
# files) of the above tools are placed.
#####
# Constants
# Installation directory
PREFIX=/usr/local/OPEN_R_SDK
# Target name
# Note: The target name 'mipsel-linux' does not mean that Linux is
# running on AIBO. 'linux' is specified just for enabling
```



Installazione in ambiente Linux - III



- Spostarsi nella directory in cui sono stati scaricati i file e lanciare lo script build-devtools-3.3.2-r1.sh (assicurarsi che si abbiano i permessi di esecuzione).
- Lo script:
 - ◆ Ricompilerà i vari pacchetti per architettura Mips
 - ◆ Installerà gli oggetti nella directory specificata



Installazione in ambiente Linux - IV



- Spostarsi nella directory di installazione quindi decomprimere i vari OPEN_R_SDK con il comando:
 - ◆ `tar -xvzf OPEN_R_SDK-... .tar.gz ./`



Sommario dell'esercitazione



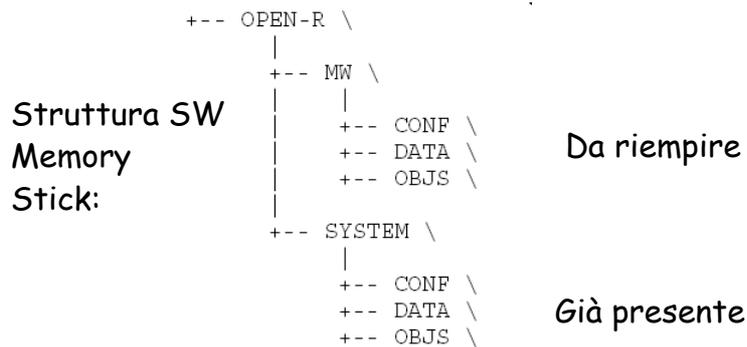
- Preparazione dell'SDK – OpenR.
- **Oggetti OpenR e comunicazione inter-object.**
- La libreria Cmotion di Carnegie Mellon
- Esempi di generazione di movimenti e debugging.
- Sperimentazione a gruppi.



Configurazione Aibo



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



A.A. 2005-2006

13/43

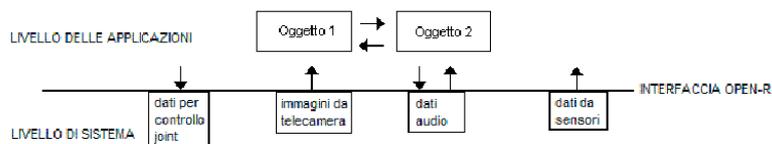
<http://homes.dsi.unimi.it/~borghese>



Open-R



In open-R vengono definiti degli oggetti.



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

A.A. 2005-2006

14/43

<http://homes.dsi.unimi.it/~borghese>



Gli oggetti



Ogni oggetto è di fatto una classe (figlia della classe `OObject`) che contiene le funzioni della classe.

Gli oggetti comunicano tra loro (e con gli oggetti di sistema) mediante:

- Memoria condivisa (classe `RRegion`)
- Scambio di messaggi (tipi predefiniti, struct, array)



Comunicazione mediante scambio di messaggi



Ogni oggetto offre dei servizi di spedizione e ricezione dei messaggi (*gate*):

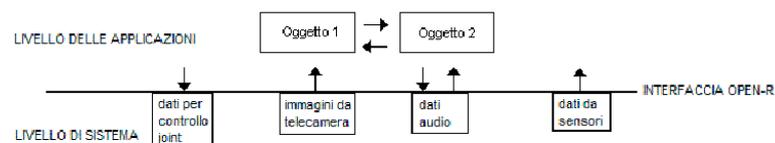
Servizi subject: spediscono il messaggio

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>`

Esempio: `GeneraMove.MotionControl.MotionCommand.S`





Classi Open-R per la comunicazione inter-object



Classi specifiche che si occupano di gestire la comunicazione.

- Classi per i Subject
 - ◆ Classe *OSubject*: classe per la gestione a runtime dei servizi subject dell'oggetto; offre metodi utili alla loro gestione.
 - ◆ Classe *OReadyEvent*: oggetto passato come parametro alle entry point invocate al ricevimento di un *Assert Ready*.
- Classi per gli Observer
 - ◆ Classe *OObserver*: classe per la gestione runtime dei servizi observer dell'oggetto.
 - ◆ Classe *ONotifyEvent*: oggetto passato come parametro all'entry point invocata al ricevimento di un messaggio; contiene tra i vari campi il messaggio stesso.

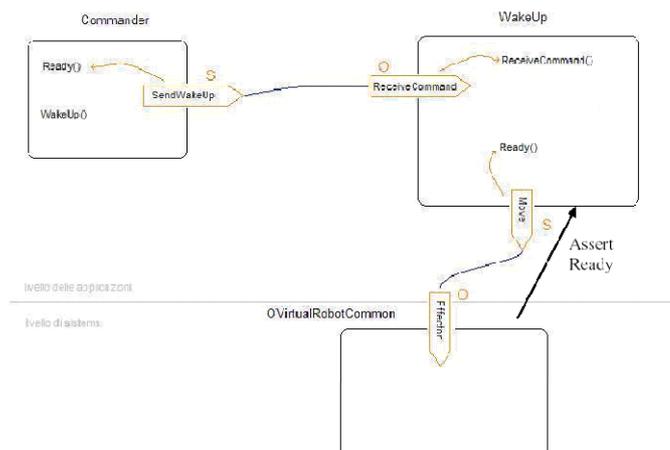
A.A. 2005-2006

17/43

<http://homes.dsi.unimi.it/~borgnese>



Entry Point e scambio di Messaggi



A.A. 2005-2006

18/43

<http://homes.dsi.unimi.it/~borgnese>



Il file stub.cfg



- **Dichiara i servizi offerti dall'Oggetto**

```

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

```

```
Extra: TimerEnd()
```

.O sta per Observer.

.S sta per Subject.

Esiste sempre almeno un observer ed un subject. In alternativa si utilizzano dummy:

```
Service: "WakeUp.DummyObserver.INT.O", null, null
```



Il file CONNECT.CFG



- **Definisce i canali di comunicazione tra gli Oggetti eseguiti mettendo in relazione i vari servizi.**
- **Inserito nella directory OPEN-R\MW\CONF.**
- **Caricato all'avvio.**

```

#
# GeneraMove <--> Motion
#
GeneraMove.MotionControl.MotionCommand.S
MotionObject.Control.MotionCommand.O

+-- OPEN-R \
|
+-- MW \
|
+-- CONF \
+-- DATA \
+-- OBJS \
|
+-- SYSTEM \
|
+-- CONF \
+-- DATA \
+-- OBJS \

```



Entry Point



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



Configurazioni di Sistema



- 3 tipi di sistema utilizzabili
 - ◆ WBASIC: senza accesso alla rete Wireless
 - ◆ WLAN: con accesso alla rete Wireless
 - ◆ WCONSOLE: con rete Wireless e console Telnet per il debug (porta 59000)

Vengono scritte la prima volta sulla memory stick USB.



Configurazione scheda WiFi



WLANCONF.TXT

Sulla memory stick nella directory: OPEN-R/SYSTEM/CONF

```
HOSTNAME = AIBO
ETHER_IP = 159.149.136.190 - 159.149.136.250
ETHER_NETMASK = 255.255.254.0
IP_GATEWAY = 0.0.0.0
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.



Sommario dell'esercitazione



- Preparazione dell'SDK – OpenR.
- Oggetti OpenR e comunicazione inter-object.
- **La libreria Cmotion di Carnegie Mellon**
- Esempi di generazione di movimenti e debugging.
- Sperimentazione a gruppi.



I movimenti principali



- Useremo la libreria di gestione dei movimenti della Carnegie Mellon
 - ◆ Sviluppata per la loro squadra di Robocup
 - ◆ Basata su cinematica inversa
 - ◆ Ottimizzata con algoritmi genetici
 - ◆ Gestisce tutti i servo motori dell'Aibo, il sottosistema audio e i LED
 - ◆ Definisce i movimenti in modo gerarchico.

E' presente un tutorial in italiano sul sito dell'Ais-Lab

:http://homes.dsi.unimi.it/~borgnese/Research/AIBO/Documenti/Aibo_MotionObject_ManualeUso_V1.1.pdf

Il sorgente è disponibile in:

http://homes.dsi.unimi.it/~borgnese/Research/AIBO/Software/MotionObject_1.1.zip



Cmotion



- Offre il servizio observer **MotionObject.Control.MotionCommand.O** per l'esecuzione di comandi di movimento.
- I comandi di movimento sono specificati nella struct **MotionCommand** (descritta nel file `MotionInterface.h` nella directory dell'oggetto Motion). Permette di specificare:
 - ◆ Direzione e velocità di spostamento dell'Aibo.
 - ◆ Movimento della testa.
 - ◆ Accensione dei LED (utili per Debug o per fornire maggiori informazioni di esecuzione).
 - ◆ Suono di file audio predefiniti.
- L'oggetto Motion non fornisce feed-back:
 - ◆ *al ricevimento di un comando, manda un Assert Ready, quindi esegue il comando fino al ricevimento di un altro comando.*
 - ◆ *Evita la cinematica inversa...*



Esempio di utilizzo di Cmotion: il cammino



I parametri del movimento:

Leg Parameters (46)

- Neutral Kinematic Position (3x4)
- Lift Velocity (3x4)
- Lift Time (1x4)
- Down Velocity (3x4)
- Down Time (1x4)
- Front Leg Height Limit (1)
- Back Leg Height Limit (1)

Body Parameters (8)

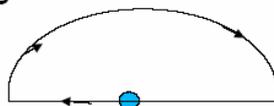
- Height of Body (1)
- Angle of Body (1)
- Hop Amplitude (1)
- Sway Amplitude (1)
- Walk Period (1)
- Max Velocities (3)



Neutral kinematics position



- **Neutral Kinematic Position** (3D vector relative to the motion coordinate frame) - Position of the leg on the ground at *some point* during the walk cycle
- Think of it as the position the legs would be in if the dog was pacing in place using your walk parameters



Path of the leg during 1 cycle



15-491 CMRoboBits

A.A. 2005-2006

29/43

<http://homes.dsi.unimi.it/~borgnese>



Parametri del movimento - I



- **Lift Velocity** (3D vector) – Velocity (mm/sec) with which the leg is lifted off the ground
- **Down Velocity** (3D vector) – Velocity (mm/sec) with which the leg is placed on the ground
- **Lift Time and Down Time** – This controls the order of the legs by specifying a percentage of the time through the time cycle that each leg is moved



15-491 CMRoboBits

A.A. 2005-2006

30/43

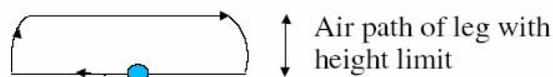
<http://homes.dsi.unimi.it/~borgnese>



Parametri del movimento - II



- **Front and Back Leg Height Limit** (mm) – Upper bound on the height of the airpath of the front and back legs.



15-491 CMRoboBits

A.A. 2005-2006

31/43

<http://homes.dsi.unimi.it/~borgese>



Parametri del movimento - III



- **Body Angle** (radians) – Angle of the body relative to the ground measured at the origin of the motion coordinate frame
- **Body Height** (mm) – Height of the body relative to the ground measured at the origin of the motion coordinate frame
- **Walk Period** (ms) – Time of one walk cycle



15-491 CMRoboBits

A.A. 2005-2006

32/43

<http://homes.dsi.unimi.it/~borgese>



Parametri del movimento - IV



- **Hop and Sway Amplitudes** (mm) –
Amplitude of vertical and horizontal oscillations
(Value usually set to 0)
- **Maximum Velocity Limits** (mm/sec)-
Velocity bounds for forward (dx), side (dy),
and angular (da) velocities. MaxDX, MaxDY,
MaxDA are equal to these parameters.



15-491 CMRoboBits

A.A. 2005-2006

33/43

<http://homes.dsi.unimi.it/~borgese>



L'insieme dei parametri - I



Values are in
millimeters,
milliseconds or
radians!

Motion::WalkParam wp; ← Parameter Struct

```
wp.leg[0].neutral.set( 125, 82,0);
wp.leg[1].neutral.set( 125,-82,0);
wp.leg[2].neutral.set( -100, 75,0);
wp.leg[3].neutral.set( -100,-75,0);
```

} Neutral Leg Position

wp.period = 640; ← Motion Cycle Period (ms)

```
wp.leg[0].lift_vel.set(0,0, 100);
wp.leg[1].lift_vel.set(0,0, 100);
wp.leg[2].lift_vel.set(0,0, 150);
wp.leg[3].lift_vel.set(0,0, 150);
```

} Lift Velocity

```
wp.leg[0].down_vel.set(0,0,-100);
wp.leg[1].down_vel.set(0,0,-100);
wp.leg[2].down_vel.set(0,0,-100);
wp.leg[3].down_vel.set(0,0,-100);
```

} Down Velocity

15-491 CMRoboBits



A.A. 2005-2006

34/43

<http://homes.dsi.unimi.it/~borgese>



L'insieme dei parametri - II



```
wp.leg[0].lift_time=0.0000; wp.leg[0].down_time=0.5000;
wp.leg[1].lift_time=0.5000; wp.leg[1].down_time=1.0000;
wp.leg[2].lift_time=0.5000; wp.leg[2].down_time=1.0000;
wp.leg[3].lift_time=0.0000; wp.leg[3].down_time=0.5000;
```

} Leg Order

```
wp.body_angle = RAD(16.0);
wp.body_height = 98;
wp.hop = 0;
wp.sway = 0;
wp.front_height = 9.0;
wp.back_height = 9.0;
```

} Limit how high the legs are lifted

```
wp.max_vel.max_dx = 240;
wp.max_vel.max_dy = 210;
wp.max_vel.max_da = 2.2;
```

} Maximum velocities
(forward, side, and angular)



15-491 CMRoboBits



Utilizzo dei parametri nel

WALK



Motion Type	Parameter File	Command
Forward Walk	walk_xy.prm	MOTION_WALK_TROT
Turning	walk_a.prm	MOTION_WALK_TROT



15-491 CMRoboBits



Sommario dell'esercitazione



- Preparazione dell'SDK – OpenR.
- Oggetti OpenR e comunicazione inter-object.
- La libreria Cmotion di Carnegie Mellon
- **Esempi di generazione di movimenti e debugging.**
- Sperimentazione a gruppi.



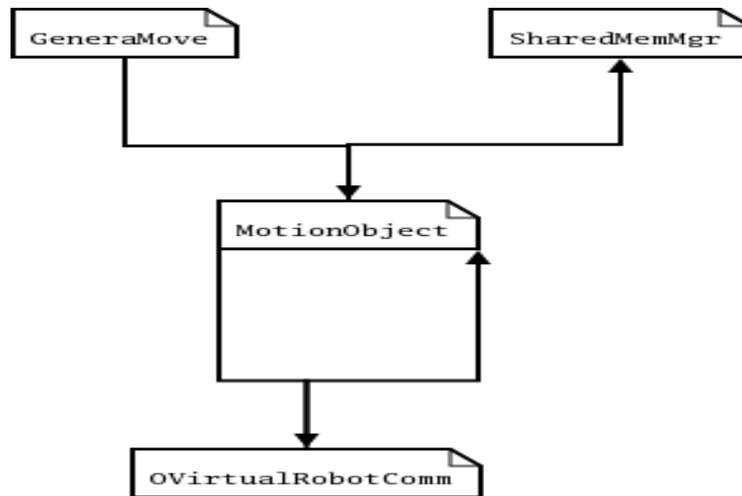
Esempio - GeneraMove



- Oggetto GeneraMove
 - ◆ Utilizza gli oggetti:
 - ☞ Motion: della Carnegie Mellon per i movimenti
 - ☞ PowerMonitor: per la gestione dell'alimentazione del cane
 - ☞ ShMemMgr: utilizzato internamente da Motion per la gestione della memoria condivisa.
 - ◆ Alla pressione del sensore posteriore dell'Aibo effettua un tiro di testa.
 - ◆ Struttura della directory:
 - ☞ **GeneraMove**: oggetto principale
 - ☞ **Headers**: header condivisi
 - ☞ **Makefile**: makefile generale
 - ☞ **Makefile.sdk.common**: variabili condivisi di compilazione
 - ☞ **Motion**: oggetto per la gestione movimenti
 - ☞ **MS**: directory da copiare in memystick
 - ☞ **shared_code**: codice condiviso
 - ☞ **SharedMemMgr**: oggetto per la gestione memoria condivisa



Esempio - GeneraMove



A.A. 2005-2006

39/43

<http://homes.dsi.unimi.it/~borghese>



Compilazione del file GeneraMove.cc



- Nella root del progetto digitare:
 1. Make all
 - Compila tutti gli oggetti del progetto
 2. Make install
 - Installa tutti gli oggetti nella directory MS

A.A. 2005-2006

40/43

<http://homes.dsi.unimi.it/~borghese>



Riempimento della memory stick



La directory MS contiene l'albero delle directory da copiare nella memystick.

Una volta compilato e installato il progetto (make install) è sufficiente copiare il contenuto di MS nella memystick

All'accensione gli oggetti vengono caricati ed eseguiti...



Debugging



- Macro: `OSYSDEBUG()`; // equivalente ad una printf
- ◆ `telnet <ip aibo> 59000` // consente di utilizzare il PC
- ◆ `telnet <ip aibo> 59000` // come console output
- ◆ `telnet <ip aibo> 59000` // dell'AIBO con IP <ip
- ◆ `-DOPENR_DEBUG` // da definire in compilazione

Se si hanno crash del processore (e.g. memory violation), viene salvato lo stato del processore nel file `emon.log`:

- Eccezioni: `emon.log`
 - ◆ Analisi del dump dei registri/memoria
 - ◆ `.../sample/common/Crush/util/emonlogparser emon.log <oggetto.nosnap.elf>`

La AIBOnet è: Aibonet

Gateway è:

Gli IP sono: 159.149.136.190 - 159.149.136.250

Subnet mask: 255.255.254. 0.



Sommario dell'esercitazione



- Preparazione dell'SDK – OpenR.
- Oggetti OpenR e comunicazione inter-object.
- La libreria Cmotion di Carnegie Mellon
- Esempi di generazione di movimenti e debugging.
- **Sperimentazione a gruppi.**