

URBI Doc
for Aibo ERS2xx ERS7
Devices documentation

v 1.3

©Jean-Christophe Baillie

October 2005

Contents

| | | |
|----------|-----------------------------|-----------|
| 1 | ERS-7 | 3 |
| 1.1 | Motors | 3 |
| 1.2 | Leds, switches, | 4 |
| 1.3 | Sensors | 4 |
| 1.4 | Camera | 5 |
| 1.5 | Speaker | 5 |
| 1.6 | Micro | 6 |
| 2 | ERS-210 | 7 |
| 2.1 | Motors | 7 |
| 2.2 | Leds, switches, | 8 |
| 2.3 | Sensors | 8 |
| 2.4 | Camera | 8 |
| 2.5 | Speaker | 9 |
| 2.6 | Micro | 9 |
| 3 | ERS-220 | 10 |
| 3.1 | Motors | 10 |
| 3.2 | Leds, switches, | 11 |
| 3.3 | Sensors | 11 |
| 3.4 | Camera | 11 |
| 3.5 | Speaker | 12 |
| 3.6 | Micro | 12 |
| 4 | URBLINI | 13 |
| 5 | CLIENT.INI | 16 |

Introduction

The "URBI Doc" file contains anything you need to know about your robot. This is the URBI Doc file for the Aibo ERS7, ERS210 and ERS220. You will find here a list and a description of all available devices and robot specific functions.

See <http://www.urbiforge.com> for news & updates.

This is not a tutorial. Please, check the URBI Tutorial for an introduction to URBI.

Chapter 1

ERS-7

1.1 Motors

The following devices are joints, with the corresponding range indicated:

| | | |
|----------|---------------------------------|-----------------------------|
| legRF1 | range= [-134.000000,120.000000] | unit=deg : Right fore legJ1 |
| legRF2 | range= [-9.000000,91.000000] | unit=deg : Right fore legJ2 |
| legRF3 | range= [-29.000000,119.000000] | unit=deg : Right fore legJ3 |
| legRH1 | range= [-134.000000,120.000000] | unit=deg : Right hind legJ1 |
| legRH2 | range= [-9.000000,91.000000] | unit=deg : Right hind legJ2 |
| legRH3 | range= [-29.000000,119.000000] | unit=deg : Right hind legJ3 |
| legLF1 | range= [-120.000000,134.000000] | unit=deg : Left fore legJ1 |
| legLF2 | range= [-9.000000,91.000000] | unit=deg : Left fore legJ2 |
| legLF3 | range= [-29.000000,119.000000] | unit=deg : Left fore legJ3 |
| legLH1 | range= [-120.000000,134.000000] | unit=deg : Left hind legJ1 |
| legLH2 | range= [-9.000000,91.000000] | unit=deg : Left hind legJ2 |
| legLH3 | range= [-29.000000,119.000000] | unit=deg : Left hind legJ3 |
| neck | range= [-79.000000,2.000000] | unit=deg : Neck tilt1 |
| headTilt | range= [-16.000000,44.000000] | unit=deg : Neck tilt2 |
| headPan | range= [-91.000000,91.000000] | unit=deg : Head pan |
| tailPan | range= [-59.000000,59.000000] | unit=deg : Tail pan |
| tailTilt | range= [2.000000,63.000000] | unit=deg : Tail tilt |
| mouth | range= [-58.000000,-3.000000] | unit=deg : Mouth |

For all joints, you have the following fields:

- `val` : the value of the joint.
- `force` : the torque measured on the joint
- `load` : the load of the joint. 0 means “loose”, and 1 means “blocked”. Values in between give intermediary results.
- `PGain` : the P gain of the joint
- `IGain` : the I gain of the joint
- `DGain` : the D gain of the joint
- `PShift` : the P shift of the joint

- IShift : the I shift of the joint
- DShift : the D shift of the joint

1.2 Leds, switches, ...

```

ledF1      range=[0.000000,1.000000] : Face light1
ledF2      range=[0.000000,1.000000] : Face light2
ledF3      range=[0.000000,1.000000] : Face light3
ledF4      range=[0.000000,1.000000] : Face light4
ledF5      range=[0.000000,1.000000] : Face light5
ledF6      range=[0.000000,1.000000] : Face light6
ledF7      range=[0.000000,1.000000] : Face light7
ledF8      range=[0.000000,1.000000] : Face light8
ledF9      range=[0.000000,1.000000] : Face light9
ledF10     range=[0.000000,1.000000] : Face light10
ledF11     range=[0.000000,1.000000] : Face light11
ledF12     range=[0.000000,1.000000] : Face light12
ledF13     range=[0.000000,1.000000] : Face light13
ledF14     range=[0.000000,1.000000] : Face light14
ledBFC     range=[0.000000,1.000000] : Back light (front,color)
ledBFW     range=[0.000000,1.000000] : Back light (front,white)
ledBMC     range=[0.000000,1.000000] : Back light (middle,color)
ledBMW     range=[0.000000,1.000000] : Back light (middle,white)
ledBRC     range=[0.000000,1.000000] : Back light (rear,color)
ledBRW     range=[0.000000,1.000000] : Back light (rear,white)
ledHC      range=[0.000000,1.000000] : Head light (color)
ledHW      range=[0.000000,1.000000] : Head light (white)
modeB      range=[0.000000,1.000000] : Mode Indicator(blue)
modeG      range=[0.000000,1.000000] : Mode Indicator(green)
modeR      range=[0.000000,1.000000] : Mode Indicator(red)
earL       range=[0.000000,1.000000] : Left ear
earR       range=[0.000000,1.000000] : Right ear
WIFIswitch range=[0.000000,1.000000] : Wireless LAN switch
ledWIFI    range=[0.000000,1.000000] : Wireless light

```

For all these devices, you have the following fields:

- val : the value of the device

There is a global “led mode” available on ERS7. You can change it via the global variable `global.ledMode` (default value is 0). The lights that will be switched on (on the face), and the colors of those lights depends on the mode. Because of this, there is no real one-to-one mapping between a device name and a physical led on the ERS-7 robot.

1.3 Sensors

```

pawLF      range=[0.000000,1.000000] : Left fore leg,paw sensor
pawLH      range=[0.000000,1.000000] : Left hind leg,paw sensor
pawRF      range=[0.000000,1.000000] : Right fore leg,paw sensor

```

| | | |
|---------------|-------------------------------|-----------------------------------|
| pawRH | range= [0.000000,1.000000] | : Right hind leg,paw sensor |
| accelX | range= [-19.613300,19.613300] | : Acceleration sensor(front-back) |
| accelY | range= [-19.613300,19.613300] | : Acceleration sensor(right-left) |
| accelZ | range= [-19.613300,19.613300] | : Acceleration sensor(up-down) |
| chinSensor | range= [0.000000,1.000000] | : Chin sensor |
| backSensorF | range= [0.000000,60.000000] | : Back sensor(front) |
| backSensorM | range= [0.000000,60.000000] | : Back sensor(middle) |
| backSensorR | range= [0.000000,60.000000] | : Back sensor(rear) |
| headSensor | range= [0.000000,35.000000] | : Head sensor |
| distanceChest | range= [19.000000,90.000000] | : Chest distance sensor |
| distanceNear | range= [5.700000,50.000000] | : Head distance sensor(near) |
| distance | range= [20.000000,150.000000] | : Head distance sensor |

For all these devices, you have the following fields:

- val : the value of the device

1.4 Camera

The camera device on Aibo is called `camera`. The available fields are:

- val : the image (binary)
- shutter : the camera shutter speed: 1=SLOW (default), 2=MID, 3=FAST
- gain : the camera gain: 1=LOW, 2=MID, 3=HIGH (default)
- wb : the camera white balance: 1=INDOOR (default), 2=OUTDOOR, 3=FLUO
- format : the camera image format: 0=YCbCr 1=jpeg (default)
- jpegfactor : the jpeg compression factor (0 to 100). Default=80
- resolution : the image resolution: 0:208x160 (default) 1:104x80 2:52x40
- reconstruct : reconstruction of the high resolution image(slow): 0:no (default) 1:yes
- width : image width
- height : image height
- xfov : camera x FOV (degrees)
- yfov : camera y FOV (degrees)

1.5 Speaker

The speaker device on Aibo is called `speaker`. The available fields are:

- val : the sound to play (binary)
- playing : equal 1 when there is a sound playing, 0 otherwise
- remain : number of milliseconds of sound to play, 0 when the buffer is empty. The delta of this variable is 32. Useful with the `=` operator.

There is also a method available:

- play (file) : plays the wav file *file* which is stored on the root of the memorystick.

1.6 Micro

The micro device on Aibo is called `micro`. The available fields are:

- `val` : contains always a buffer of 2048 bytes of the latest sound heard by the robot.
- `volume` : measures the volume of sound.
- `volumeR` : measures the volume of sound in the Right ear.
- `volumeL` : measures the volume of sound in the Left ear.

Chapter 2

ERS-210

2.1 Motors

The following devices are joints, with the corresponding range indicated:

| | | |
|----------|---------------------------------|-----------------------------|
| legRF1 | range= [-116.000000,116.000000] | unit=deg : Right fore legJ1 |
| legRF2 | range= [-9.000000,92.000000] | unit=deg : Right fore legJ2 |
| legRF3 | range= [-26.000000,146.000000] | unit=deg : Right fore legJ3 |
| legRH1 | range= [-116.000000,116.000000] | unit=deg : Right hind legJ1 |
| legRH2 | range= [-9.000000,92.000000] | unit=deg : Right hind legJ2 |
| legRH3 | range= [-26.000000,146.000000] | unit=deg : Right hind legJ3 |
| legLF1 | range= [-116.000000,116.000000] | unit=deg : Left fore legJ1 |
| legLF2 | range= [-9.000000,92.000000] | unit=deg : Left fore legJ2 |
| legLF3 | range= [-26.000000,146.000000] | unit=deg : Left fore legJ3 |
| legLH1 | range= [-116.000000,116.000000] | unit=deg : Left hind legJ1 |
| legLH2 | range= [-9.000000,92.000000] | unit=deg : Left hind legJ2 |
| legLH3 | range= [-26.000000,146.000000] | unit=deg : Left hind legJ3 |
| tailPan | range= [-21.000000,21.000000] | unit=deg : Tail tilt |
| tailTilt | range= [-21.000000,21.000000] | unit=deg : Tail pan |
| headTilt | range= [-87.000000,41.000000] | unit=deg : Head tilt |
| headPan | range= [-89.000000,89.000000] | unit=deg : Head pan |
| headRoll | range= [-27.000000,27.000000] | unit=deg : Head roll |
| mouth | range= [-46.000000,-3.000000] | unit=deg : Mouth |

For all joints, you have the following fields:

- `val` : the value of the joint
- `force` : the torque measured on the joint
- `load` : the load of the joint. 0 means “loose”, and 1 means “blocked”. Values in between give intermediary results.
- `PGain` : the P gain of the joint
- `IGain` : the I gain of the joint
- `DGain` : the D gain of the joint
- `PShift` : the P shift of the joint

- IShift : the I shift of the joint
- DShift : the D shift of the joint

2.2 Leds, switches, ...

ERS210

| | | |
|--------|---------------------------|-------------------------------------|
| ledEUL | range=[0.000000,1.000000] | unit=bool : Eye light(Upper left) |
| mode | range=[0.000000,1.000000] | unit=bool : Mode indicator |
| ledEUR | range=[0.000000,1.000000] | unit=bool : Eye light(Upper right) |
| earL | range=[0.000000,1.000000] | unit=bool : Left ear |
| earR | range=[0.000000,1.000000] | unit=bool : Right ear |
| ledTB | range=[0.000000,1.000000] | unit=bool : Tail light(Blue) |
| ledTO | range=[0.000000,1.000000] | unit=bool : Tail light(Orange) |
| ledELL | range=[0.000000,1.000000] | unit=bool : Eye light(Lower left) |
| ledEML | range=[0.000000,1.000000] | unit=bool : Eye light(Middle left) |
| ledELR | range=[0.000000,1.000000] | unit=bool : Eye light(Lower right) |
| ledEMR | range=[0.000000,1.000000] | unit=bool : Eye light(Middle right) |

For all these devices, you have the following fields:

- val : the value of the device

2.3 Sensors

| | | |
|--------------|------------------------------|---|
| pawLF | range=[0.000000,1.000000] | unit=bool : Left fore leg,paw sensor |
| pawLH | range=[0.000000,1.000000] | unit=bool : Left hind leg,paw sensor |
| pawRF | range=[0.000000,1.000000] | unit=bool : Right fore leg,paw sensor |
| pawRH | range=[0.000000,1.000000] | unit=bool : Right hind leg,paw sensor |
| backSensor | range=[0.000000,1.000000] | unit=bool : Back sensor |
| chinSensor | range=[0.000000,1.000000] | unit=bool : Chin sensor |
| headSensorB | range=[0.000000,98.000000] | unit=uPa : Head sensor(back) |
| headSensorF | range=[0.000000,98.000000] | unit=uPa : Head sensor(front) |
| distance | range=[10.000000,90.000000] | unit=m : Distance device |
| thermoSensor | range=[0.000000,60.000000] | unit=C : Thermo sensor |
| accelX | range=[-19.613300,19.613300] | unit=m/s2 : Acceleration sensor(right-left) |
| accelY | range=[-19.613300,19.613300] | unit=m/s2 : Acceleration sensor(front-back) |
| accelZ | range=[-19.613300,19.613300] | unit=m/s2 : Acceleration sensor(up-down) |

For all these devices, you have the following fields:

- val : the value of the device

2.4 Camera

The camera device on Aibo is called camera. The available fields are:

- val : the image (binary)
- shutter : the camera shutter speed: 1=SLOW (default), 2=MID, 3=FAST
- gain : the camera gain: 1=LOW, 2=MID, 3=HIGH (default)

- `wb` : the camera white balance: 1=INDOOR (default), 2=OUTDOOR, 3=FLUO
- `format` : the camera image format: 0=YCbCr 1=jpeg (default)
- `jpegfactor` : the jpeg compression factor (0 to 100). Default=80
- `resolution` : the image resolution: 0:208x160 (default) 1:104x80 2:52x40
- `reconstruct` : reconstruction of the high resolution image(slow): 0:no (default) 1:yes
- `width` : image width
- `height` : image height
- `xfov` : camera x FOV (degrees)
- `yfov` : camera y FOV (degrees)

2.5 Speaker

The speaker device on Aibo is called `speaker`. The available fields are:

- `val` : the sound to play (binary)
- `playing` : equal 1 when there is a sound playing, 0 otherwise
- `remain` : number of milliseconds of sound to play, 0 when the buffer is empty. The delta of this variable is 32. Useful with the `=` operator.

There is also a method available:

- `play (file)` : plays the wav file *file* which is stored on the root of the memorystick.

2.6 Micro

The micro device on Aibo is called `micro`. The available fields are:

- `val` : contains always a buffer of 2048 bytes of the latest sound heard by the robot.
- `volume` : measures the volume of sound.
- `volumeR` : measures the volume of sound in the Right ear.
- `volumeL` : measures the volume of sound in the Left ear.

Chapter 3

ERS-220

3.1 Motors

The following devices are joints, with the corresponding range indicated:

| | | |
|----------|---------------------------------|-----------------------------|
| legRF1 | range= [-116.000000,116.000000] | unit=deg : Right fore legJ1 |
| legRF2 | range= [-10.000000,92.000000] | unit=deg : Right fore legJ2 |
| legRF3 | range= [-26.000000,146.000000] | unit=deg : Right fore legJ3 |
| legRH1 | range= [-116.000000,116.000000] | unit=deg : Right hind legJ1 |
| legRH2 | range= [-10.000000,92.000000] | unit=deg : Right hind legJ2 |
| legRH3 | range= [-26.000000,146.000000] | unit=deg : Right hind legJ3 |
| legLF1 | range= [-116.000000,116.000000] | unit=deg : Left fore legJ1 |
| legLF2 | range= [-10.000000,92.000000] | unit=deg : Left fore legJ2 |
| legLF3 | range= [-26.000000,146.000000] | unit=deg : Left fore legJ3 |
| legLH1 | range= [-116.000000,116.000000] | unit=deg : Left hind legJ1 |
| legLH2 | range= [-10.000000,92.000000] | unit=deg : Left hind legJ2 |
| legLH3 | range= [-26.000000,146.000000] | unit=deg : Left hind legJ3 |
| headPan | range= [-89.000000,89.000000] | unit=deg : Head pan |
| headRoll | range= [-27.000000,27.000000] | unit=deg : Head roll |
| headTilt | range= [-87.000000,41.000000] | unit=deg : Head tilt |

For all joints, you have the following fields:

- `val` : the value of the joint
- `force` : the torque measured on the joint
- `load` : the load of the joint. 0 means “loose”, and 1 means “blocked”. Values in between give intermediary results.
- `PGain` : the P gain of the joint
- `IGain` : the I gain of the joint
- `DGain` : the D gain of the joint
- `PShift` : the P shift of the joint
- `IShift` : the I shift of the joint
- `DShift` : the D shift of the joint

3.2 Leds, switches, ...

| | | | |
|----------------|---------------------------|-----------|--------------------------------------|
| ledTailC | range=[0.000000,1.000000] | unit=bool | : Tail light (Center) |
| ledBL | range=[0.000000,1.000000] | unit=bool | : Head Face side light(Back left) |
| ledTailL | range=[0.000000,1.000000] | unit=bool | : Tail light (Left) |
| ledCL | range=[0.000000,1.000000] | unit=bool | : Head Face side light(Center left) |
| ledBR | range=[0.000000,1.000000] | unit=bool | : Head Face side light(Back right) |
| ledTailR | range=[0.000000,1.000000] | unit=bool | : Tail light (Right) |
| ledCR | range=[0.000000,1.000000] | unit=bool | : Head Face side light(Center right) |
| ledFL | range=[0.000000,1.000000] | unit=bool | : Head Face side light(Front left) |
| ledL1 | range=[0.000000,1.000000] | unit=bool | : Back multi indic (1st from left) |
| ledL2 | range=[0.000000,1.000000] | unit=bool | : Back multi indic (2nd from left) |
| ledRetractHead | range=[0.000000,1.000000] | unit=bool | : Retractable head light |
| ledL3 | range=[0.000000,1.000000] | unit=bool | : Back multi indic (3rd from left) |
| ledFR | range=[0.000000,1.000000] | unit=bool | : Head Face side light(Front right) |
| ledHead | range=[0.000000,1.000000] | unit=bool | : Head indicator |
| ledR1 | range=[0.000000,1.000000] | unit=bool | : Back multi indic (1st from right) |
| ledR2 | range=[0.000000,1.000000] | unit=bool | : Back multi indic (2nd from right) |
| ledR3 | range=[0.000000,1.000000] | unit=bool | : Back multi indic (3rd from right) |
| ledA | range=[0.000000,1.000000] | unit=bool | : Face front light A |
| ledB | range=[0.000000,1.000000] | unit=bool | : Face front light B |
| ledC | range=[0.000000,1.000000] | unit=bool | : Face front light C |

For all these devices, you have the following fields:

- val : the value of the device

3.3 Sensors

| | | | |
|--------------|------------------------------|-----------------------|------------------------------------|
| pawLF | range=[0.000000,1.000000] | unit=bool | : Left fore leg,paw sensor |
| pawLH | range=[0.000000,1.000000] | unit=bool | : Left hind leg,paw sensor |
| pawRF | range=[0.000000,1.000000] | unit=bool | : Right fore leg,paw sensor |
| pawRH | range=[0.000000,1.000000] | unit=bool | : Right hind leg,paw sensor |
| thermoSensor | range=[0.000000,60.000000] | unit=uPa | : Thermo sensor |
| tailSensorC | range=[0.000000,1.000000] | unit=uPa | : Tail sensor (Center from behind) |
| backSensor | range=[0.000000,99.000000] | unit=uPa | : Back sensor |
| tailSensorL | range=[0.000000,1.000000] | unit=uPa | : Tail sensor (Left from behind) |
| tailSensorR | range=[0.000000,1.000000] | unit=uPa | : Tail sensor (Right from behind) |
| headSensorB | range=[0.000000,35.000000] | unit=uPa | : Head sensor(back) |
| headSensorF | range=[0.000000,60.000000] | unit=uPa | : Head sensor(front) |
| faceSensor | range=[0.000000,60.000000] | unit=uPa | : Face sensor |
| distance | range=[10.000000,90.000000] | unit=cm | : Distance device |
| accelX | range=[-19.613300,19.613300] | unit=m/s ² | : Acceleration sensor(right-left) |
| accelY | range=[-19.613300,19.613300] | unit=m/s ² | : Acceleration sensor(front-back) |
| accelZ | range=[-19.613300,19.613300] | unit=m/s ² | : Acceleration sensor(up-down) |

For all these devices, you have the following fields:

- val : the value of the device

3.4 Camera

The camera device on Aibo is called camera. The available fields are:

- val : the image (binary)

- `shutter` : the camera shutter speed: 1=SLOW (default), 2=MID, 3=FAST
- `gain` : the camera gain: 1=LOW, 2=MID, 3=HIGH (default)
- `wb` : the camera white balance: 1=INDOOR (default), 2=OUTDOOR, 3=FLUO
- `format` : the camera image format: 0=YCbCr 1=jpeg (default)
- `jpegfactor` : the jpeg compression factor (0 to 100). Default=80
- `resolution` : the image resolution: 0:208x160 (default) 1:104x80 2:52x40
- `reconstruct` : reconstruction of the high resolution image(slow): 0:no (default) 1:yes
- `width` : image width
- `height` : image height
- `xfov` : camera x FOV (degrees)
- `yfov` : camera y FOV (degrees)

3.5 Speaker

The speaker device on Aibo is called `speaker`. The available fields are:

- `val` : the sound to play (binary)
- `playing` : equal 1 when there is a sound playing, 0 otherwise
- `remain` : number of milliseconds of sound to play, 0 when the buffer is empty. The delta of this variable is 32. Useful with the `=` operator.

There is also a method available:

- `play (file)` : plays the wav file *file* which is stored on the root of the memorystick.

3.6 Micro

The micro device on Aibo is called `micro`. The available fields are:

- `val` : contains always a buffer of 2048 bytes of the latest sound heard by the robot.
- `volume` : measures the volume of sound.
- `volumeR` : measures the volume of sound in the Right ear.
- `volumeL` : measures the volume of sound in the Left ear.

Chapter 4

URBI.INI

Here is a suggested default URBI.INI file which call the "std.u" file. "std.u" sets the standard grouping hierarchy for aibo. This URBI.INI example also starts a nice animation (tagged with anim, so that you can stop it with stop anim). Walk and sit/stand/poses are made available with the motion.u file.

The actual URBI.INI released with your URBI version might differ slightly from this one, but general ideas are the same.

```
#####  
### Default URBI.INI: You are advised to modify this file to ###  
### suit your needs, and you should not overwrite it with new ###  
### releases. ###  
#####  
  
load("std.u"); // Standard URBI file for Aibo  
load("motion.u"); // Walk functions  
  
speaker.play("start.wav");  
  
//ball tracking head example  
robot.bt = 0;  
balltracking :  
  whenever ( (robot.bt == 1) && (ball.x != -1) ) {  
    headPan.val = headPan.val + camera.xfov * ( 0.5 - ball.x ) &  
    headTilt.val = headTilt.val + camera.yfov * ( 0.5 - ball.y )  
  },  
  
//animations  
// ERS7  
if (global.name == "ERS-7") {  
  anim:ledF12.val = 1,  
  anim:ledBFW.val = 0.2 sin:4000 ampli:0.5 &  
  anim:ledBMW.val = 0.2 sin:4000 ampli:0.5 phase:(pi/3) &  
  anim:ledBRW.val = 0.2 sin:4000 ampli:0.5 phase:(2*pi/3)  
},  
  
// ERS210  
if (global.name == "ERS-210") {  
  anim: ledEML.val = 1,  
  anim: ledEMR.val = 1,
```

```

    anim: ledTB.val = 0.5 sin:4000 ampli:0.6
  },

  // ERS220
  if (global.name == "ERS-220") {
    anim:ledA.val = 1;
    anim:ledB.val = 1;
    anim:ledC.val = 1;
    anim:ledTailC.val = 0.5 sin:4000 ampli:0.6
  },

```

Here is the std.u file:

```

// std.u is a basic "standard" set of scripts for aibo
// normally, you want to include it at the beginning
// of your URBI.INI file

/*****
*
*   BASIC GROUPING
*
*****/

group legRF {legRF1,legRF2,legRF3},
group legLF {legLF1,legLF2,legLF3},
group legRH {legRH1,legRH2,legRH3},
group legLH {legLH1,legLH2,legLH3},
group legs  {legRF,legLF,legRH,legLH},
group leg1  {legRF1,legLF1,legRH1,legLH1},
group leg2  {legRF2,legLF2,legRH2,legLH2},
group leg3  {legRF3,legLF3,legRH3,legLH3},

group legF1 {legRF1,legLF1},
group legF2 {legRF2,legLF2},
group legF3 {legRF3,legLF3},
group legH1 {legRH1,legLH1},
group legH2 {legRH2,legLH2},
group legH3 {legRH3,legLH3},

legs.val->blend = mix;

#odd/even leg pair groups (using the 'exec' function)
for (i=1;i<4;i++) exec("group legE"+string(i)+" {legLH"+string(i)+",legRF"+string(i)+"
for (i=1;i<4;i++) exec("group legO"+string(i)+" {legLF"+string(i)+",legRH"+string(i)+"

// Useful aliases

alias global.leg[1][1][1] legLF1.val;
alias global.leg[1][1][2] legLF2.val;
alias global.leg[1][1][3] legLF3.val;
alias global.leg[1][2][1] legRF1.val;
alias global.leg[1][2][2] legRF2.val;

```

```

alias global.leg[1][2][3] legRF3.val;
alias global.leg[2][1][1] legLH1.val;
alias global.leg[2][1][2] legLH2.val;
alias global.leg[2][1][3] legLH3.val;
alias global.leg[2][2][1] legRH1.val;
alias global.leg[2][2][2] legRH2.val;
alias global.leg[2][2][3] legRH3.val;

// Robot-specific groups

if (global.name == "ERS-7") {

    group head {neck,headPan,headTilt,mouth},
    group tail {tailPan,tailTilt},
    group ears {earR,earL},
    group robot {legs,head,tail},

    group ledF {ledF1,ledF2,ledF3,ledF4,ledF5,ledF6,ledF7,
                ledF8,ledF9,ledF10,ledF11,ledF12,ledF13,ledF14},
    group ledHead {modeR,modeG,modeB,ledHC,ledHW},
    group ledBW {ledBFW,ledBMW,ledBRW},
    group ledBC {ledBFC,ledBMC,ledBRC},
    group leds {ledF,ledHead,ledBW,ledBC,ledWIFI},
},

if (global.name == "ERS-210") {

    group head {headRoll,headPan,headTilt,mouth},
    group tail {tailPan,tailTilt},
    group robot {legs,head,tail},
    group ears {earR,earL},

    group ledT {ledTB,ledTO},
    group ledE {ledELL,ledEML,ledEUL,ledELR,ledEMR,ledEUR},
    group leds {mode,ledT,ledE},
},

if (global.name == "ERS-220") {

    group head {headRoll,headPan,headTilt},
    group robot {legs,head},
    group ledTail {ledTailC,ledTailR,ledTailL},
    group ledFace {ledBL,ledCL,ledBR,ledCR,ledFL,ledFR},
    group ledBack {ledL1,ledL2,ledL3,ledR1,ledR2,ledR3},
    group leds {ledTail,ledFace,ledBack},
},

```


Chapter 5

CLIENT.INI

Here is the default URBI.INI file which plays a sound and start a battery monitor:

```
speaker.play("client.wav");

//Power Monitoring
current_power=inf;
power:at ( current_power - power() >= 0.01 ) {
  current_power=power();
  power:echo "Battery at "+string(current_power*100)+" %"
};
```