

URBI Doc for urbiengine-ePuck

Devices documentation

(book compiled from Revision 418M)

Benjamin Renoust

URBI Doc for urbiengine-ePuck: Devices documentation: (book compiled from Revision 418M)

by Benjamin Renoust

Published

Copyright © 2008 Gostai

This document is released under the Attribution-NonCommercial-NoDerivs 2.0 Creative Commons licence (<http://creativecommons.org/licenses/by-nc-nd/2.0/deed.en>).

Table of Contents

1. Introduction	1
2. Sample	2
3. Available UObject Devices	3
Wheel	3
Led	3
Micro	4
Selector	4
IR	4
Accelerometer	5
Camera	5
Colormap	5
Speaker	6
Command	7
Instances	8
Groups	8
4. How to use the current layout (not yet completed)	9
Introduction	9
Motors	9
Sensors	9
5. Tutorials (not yet completed)	10
How to make basic movement functions.	10
Tip :	10
How to make a random move	12
A. Copyright	13

List of Tables

3.1. Wheel's attributes:	3
3.2. Led's attributes:	3
3.3. Sound sensor's attributes:	4
3.4. Selector's attributes:	4
3.5. IR's attributes:	4
3.6. Accelerometer's attributes:	5
3.7. Camera's attributes:	5
3.8. colormap's attributes:	6
3.9. Speaker's methods:	7
3.10. Command's methods:	7
3.11. Servo's characteristics	8
3.12. Group's list	8

Chapter 1. Introduction

This documentation contains informations about the ePuck's URBI engine. For instance, the available devices are summarized here. You will find also many tutorials on how to use the current layout, how to make your own layout and more.

Chapter 2. Sample

Here is a simple example that explains how to use the URBI server. In the `bin/` directory, write:

```
./urbi.bat  
(or double-click on urbi.bat icon to launch it)
```

Linux users have a `urbi.sh` instead. This script launch e-Puck server with parameters. You can edit this launcher file to see parameters send.

Description of parameters:

- `-i` : to have interactive mode, you can enter commands directly in the server windows, no need of external client.
- `-P 10` : refresh rate of variables in the server : 10ms
- `-I ../gostai/ePuck.data` : (upper-case "i") include this folder to path to find Urbi files.

Like any Urbi server, the default port is 54000 (can be changed with `-P <PORT>`).

Don't forget to defines the connection port in `ePuck.data/config.u`, it can be any COM port where the robot is plugged (or any `/dev/ttyXX` or `/dev/rfcommXX`).

To configure bluetooth connection between your computer and your e-Puck, see e-Puck documentation, and documentations like http://www.edinburghrobotics.com/docs/howto_epuck_bluetooth.html.

Chapter 3. Available UObject Devices

The following devices documentation is written for the ePuck URBI engine.

Wheel

The Wheel UObject describes a wheel device and contains the following attributes:

Table 3.1. Wheel's attributes:

Name	Description
id	Wheel's identification, it can be "Left" or "Right"
val	Wheel's speed, from -1000 to 1000
position	Wheel's position, from -32000 to 32000.
init(id)	The UObject constructor. id is an identification. (See previous attribute id). Example: <pre>wheel = new Wheel("Left");</pre>
reset()	Reset the wheel current position to 0. Example: <pre>wheel.reset();</pre>

Led

The Led UObject describes a led device and contains the following attributes:

Table 3.2. Led's attributes:

Name	Description
id	An identification, values can be 0 to 9. Each value refers to a different led.
val	The status of the led. It can be 0 (off), 1 (on) or 2 (switch).
init(id)	The UObject constructor. id is an identification number. (See previous attribute id). Example: <pre>ledA = new Led(1);</pre>

Micro

The Micro UObject describes a microphone device and contains the following attributes:

Table 3.3. Sound sensor's attributes:

Name	Description
id	An identification value, it can be "Left", "Right" or "Back".
val	The sound level measured.
init(id)	The UObject constructor. id is an identification name. (See previous attribute id). Example: <code>mic = new Micro("Back");</code>

Selector

The Selector UObject describes the little selector switch on the top of the robot and contains the following attributes:

Table 3.4. Selector's attributes:

Name	Description
val	The current selector position (can be 0 to 15).
init()	The UObject constructor. Example: <code>sel = new Selector();</code>

IR

The IR UObject describes the 8 infrared devices and contains the following attributes:

Table 3.5. IR's attributes:

Name	Description
val	The IR proximity values. It's a list of 8 values corresponding to 8 proximity distances.
light	The IR light values. It's a list of 8 values corresponding to 8 ambient light intensity.
init()	The UObject constructor. Example: <code>infraRed = new IR();</code>

Accelerometer

The IR UObject describes the 3 accelerometers devices and contains the following attributes:

Table 3.6. Acceleromter's attributes:

Name	Description
val	The vertical acceleration value.
inclination	The inclination value.
orientation	The orientation value.
init()	The UObject constructor Example <pre>acc = new Acceleromter();</pre>

Camera

The Camera UObject describes the camera device and contains the following attributes:

Table 3.7. Camera's attributes:

Name	Description
val	The current camera image.
zoom	The zoom value. By default it's set to 8, but it can also be 4, 2 or 1.
mode	The camera mode value. By default it's set to 1 (which means color), otherwise it can be 0 (black and white).
height	The camera height value. By default it's set to 40, it's not advised to change this value which causes a high instability in the robot.
width	The camera width value. By default it's set to 40, it's not advised to change this value which causes a high instability in the robot.
init()	The UObject constructor Example <pre>c = new Camera();</pre>

Colormap

The Colomap UObject when binded to the camera device provides a basic blob detector and contains the following attributes:

Table 3.8. colormap's attributes:

Name	Description
x	The x position of the blob's gravity center.
y	The y position of the blob's gravity center.
visible	This property is set to 1 when the blob is visible.
ratio	The blob's ratio.
threshold	The blob's threshold.
orientation	The blob's orientation.
elongation	The blob's elongation.
ymin	The ymin value of the desired color. (The blob's color Y component in YCbCr).
ymax	The ymax value of the desired color. (The blob's color Y component in YCbCr).
cbmin	The cbmin value of the desired color. (The blob's color Cb component in YCbCr).
cbmax	The cbmax value of the desired color. (The blob's color Cb component in YCbCr).
crmin	The cbmin value of the desired color. (The blob's color Cr component in YCbCr).
crmax	The cbmax value of the desired color. (The blob's color Cr component in YCbCr).
init(source, ymin, ymax, cbmin, cbmax, crmin, crmax, threshold)	<p>The UObject constructor</p> <p><code>source</code> is the name of the image source (usually the camera).</p> <p><code>ymin</code>, <code>ymax</code>, <code>cbmin</code>, <code>cbmax</code>, <code>crmin</code>, <code>crmax</code> are parameters of the blob's color.</p> <p><code>threshold</code> is the minimum ratio of the chosen color to consider the blob visible.</p> <p>Example</p> <pre>c = new Colormap("camera.val", 0, 255, 120, 190, 190, 190, 0.5)</pre> <p>which detects a pink ball.</p>

Speaker

The Speaker UObject describes the speaker device, allowing you to use:

Table 3.9. Speaker's methodes:

Name	Description
play(sound)	Plays a custom sound. sound can be "wow", "waa", "ouch", "hit", "uh" or "off". Example: <code>Speaker.play("wow");</code>
stop()	Calls <code>play("off")</code> : it shuts off the speaker and cuts off the speaker's permanent noise. Example <code>Speaker.stop();</code>

Command

The Command UObject is designed to allow you special commands to use:

Table 3.10. Command's methodes:

Name	Description
reset()	Reset completely the robot. Example: <code>Command.reset();</code> .
stop()	Stop the robot and light off the leds. Example: <code>Command.stop();</code> .
version()	Prints out the current firmware version. Example: <code>Command.version();</code> .
calibrate()	Calibrate the IR sensors. Example: <code>Command.calibrate();</code> .

Instances

Here are the different instances we have chosen for the current layout.

Table 3.11. Servo's characteristics

Instance	UObject	Description
wheelL	Wheel	Left wheel
wheelR	Wheel	Right wheel
selector	Selector	button selector on the top
led[0]	Led	led on front
led[...]	Led	led around the robot
led[9]	Led	led on right
ir	IR	infraRed sensor table
acc	Accelerometer	Accelerometer

Groups

Here are the different groups provided in the current layout.

Table 3.12. Group's list

Group Name	Description
wheels	The two wheels.
motors	All the motors.
sensors	All the sensors.
hardware	All the devices.

Chapter 4. How to use the current layout (not yet completed)

Introduction

This documentation presents how to use the engine with the current layout loaded. You might find useful information about URBI on <http://gostai.com/doc/urbi-tutorial.htm>.

Motors

Two motors are available, `wheelL` and `wheelR`. This two motors are grouped in `wheels`.

The attribute `.torque` allows you to set the current speed of the motor. This value is set between -1000 and 1000. So you can control the speed of wheels just by typing :

```
wheels.torque = 500;
```

A negative value inverses the motor rotation.

You can read the motor position (in degrees) using the `.position` attribute. This attribute is set to 0 when you turn the robot on.

Sensors

Request `ir[0].val` to get its value. This works also with the battery object.

Sensors are grouped in the `sensors` group and all hardware devices (`sensors` + `motors` + `battery` + `beeper`) are grouped in the `hardware` group.

Chapter 5. Tutorials (not yet completed)

How to make basic movement functions.

A good thing that makes life easier with URBI is to code high-level simple functions. This allows you to move your robot easily just calling these functions. Let's start with the `forward()` function which will make your robot move... forward!

```
function global.forward()  
{  
  wheels = 300;  
},
```

Here we declare a function `forward()` in the `global` scope. Declaring the function in a scope (you can call it as you wish), allows it to make it visible from any connection where you execute it.

`wheels = 300;` sets the wheels speed to 300.

Tip :

In order not to always call `global.forward()`, you can create a `global` class and store inside it all your methods and attributes:

```
class global  
{  
  var defaultSpeed ;  
  var x ;  
  var y ;  
  function forward() ;  
  function randomMove() ;  
} ;  
function global.forward()  
{  
  //function definition ;  
} ;  
function global.randomMove()  
{  
  //function definition ;  
  forward() ; //called without global  
} ;
```

Now we can improve this code allowing a custom speed. Here is a good way to do it:

```
global.defaultSpeed = 300;  
  
function global.forward()  
{  
  wheels = global.defaultSpeed;
```

```
},
```

Here we set a variable in the `global` scope which defines the custom speed. That allows you to set the speed whenever you want (very useful while testing your own program).

Finally we can add a timer to this function, useful to move only during a fixed time:

```
global.defaultSpeed = 300;

function global.forward(timer)
{
    wheels = global.defaultSpeed |
    sleep(timer)|
    wheels = 0;
},
```

Here we added a parameter `timer` which is the time you wish to run your wheels. The syntax `wheels = global.defaultSpeed | sleep(timer)` execute sequentially `wheels = global.defaultSpeed` THEN `sleep(timer)`. When you separate a group to another one (e.g. : `wheels = 0` - which stops the wheels) using the separator `|`, that makes the instruction `sleep` for the previous instruction. So `wheels = 0` will be executed AFTER the timer is done. So this call makes the robot move forward for one second and then stop.

```
global.forward(1000),
```

Following is a sample from the `std.u` file which defines basic movement functions.

```
class global ;

global.defaultSpeed = 300;

function global.forward()
{
    wheels = global.defaultSpeed;
},

function global.backward()
{
    wheels = -global.defaultSpeed;
},

function global.turnleft()
{
    wheelL = - global.defaultSpeed &
    wheelR = global.defaultSpeed;
},

function global.turnright()
{
    wheelR = - global.defaultSpeed &
    wheelL = global.defaultSpeed;
```

```
},
```

How to make a random move

Here we will see how to create a function which will generate a random move on the robot. This function will be useful for the next part of our tutorial. This function is designed to make the robot react to an event. So the idea is to make a backward move and then to randomly change the direction. We also require 2 parameters, a constant time, which will have the robot turn for this amount of time (that avoid not to turn enough); and random time base which is the maximum turning time that will randomly generated. Here is the code:

```
function global.randomMoves(constantTime, randomTime)
{
  {backward() & sleep(2s)}
  |
  {
    {if (random(2) == 1) left()
      else right()}
    }
    &
    sleep( (constantTime+random(randomTime))*100)
  }
  |
  forward()
},
```

```
{backward() & sleep(2s)} So firstly
```

Appendix A. Copyright

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

1. "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License. 2. "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License. 3. "Licensor" means the individual or entity that offers the Work under the terms of this License. 4. "Original Author" means the individual or entity who created the Work. 5. "Work" means the copyrightable work of authorship offered under the terms of this License. 6. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive,

perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

1. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works; 2. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Derivative Works. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(d) and 4(e).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

1. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested.
2. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.
3. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied;

and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit. 4.

For the avoidance of doubt, where the Work is a musical composition:

1. Performance Royalties Under Blanket Licenses. Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.
2. Mechanical Rights and Statutory Royalties. Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.
5. Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions), if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR

EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

1. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. 2. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

1. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License. 2. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable. 3. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent. 4. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.