URBI Doc for urbiengine-irobot

Devices documentation

(book compiled from Revision 425M)

Guillaume Deslandes
Gégoire Le Blan

URBI Doc for urbiengine-irobot: Devices documentation: (book compiled from Revision 425M)

by Guillaume Deslandes and Gégoire Le Blan

Published
Copyright © 2006-2007 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

List of Tables

Chapter 1. Introduction

This documentation contains informations about the iRobot Create URBI engine (iRobot URBI engine, or iRobot engine for short).

You may be interested in reading information about the iRobot Create at *irobot.com/create* [http://www.irobot.com/create] if you are not familiar with it yet.

The first chapters provide quicklaunch instructions and a short tutorial for using URBI on the iRobot Create.

The remainder of the documentation contains an exhaustive reference for the iRobot URBI engine and more advanced URBI script examples.

Chapter 2. Quickstart

Fisrt, make sure your iRobot Create is operational. You may have to have a look at irobot.com/create [http://www.irobot.com/create] if you need help with this step...

Once you have a working robot you can install the iRobot engine. The installation procedure may vary depending on your system, but it should not be unfamiliar to you. Once it is completed, you will be provided with a `bin` directory in which you will find the `urbi-server-irobot` binary. Other important files are located in the `data` directory.

Before launching the engine, you may need to change the `system.comport` value in the `data/config.u` file. It should have a value like `COM1` on MS Windows, `/dev/cu.SLAB_USBtoUART` on Mac OS X or `/dev/ttyUSB0` on Linux. The engine will abort if the port specified is incorect or doesn't have the proper permissions.

Go to your iRobot install directory, change to the `bin` directory and launch `urbi-server-irobot`. The following help message should appear :

```
usage: ./urbi-server-irobot [options] period [path1 path2 ...]
  period : base URBI interval in milliseconds
  path items are absolute or relative path elements searched in order
    for files when 'load' is called.
  options:
    -p port : specify the tcp port URBI will listen to.
    -b address: bind to a specific ip address.
    -n      : disable networking.
    -r      : enable reporting of time taken by URBI loop to execute
    -s <period>: shell-mode (no network) with given period
    -f      : enable fast mode: the server will run as fast
              as possible and emulate the period specified
```

To quickly launch the iRobot engine, just type :

```
./urbi-engine-irobot 50 ../data
```

The engine will display a header and say it is ready. If all is right, nothing more should happen. If your robot is switched off, or unreachable, the engine should say *Waiting for robot initialisation.*

At this time, your engine should be running. To test the connection, use either a tool specialy provided by Gostai, or a simple telnet client (please note that all URBI engine ports default to 54000).

```
telnet localhost 54000
```

If no error occurs, your telnet client should receive the same kind of header you saw in the engine window. The last line should give your connection id and be of the form :

```
[65000000:ident] *** ID: U135766920
[65000011:notag] *** Connected to robot
```

If this is not the case, please make sure you correctly installed the engine and followed the previous steps. Please refer to Troubleshooting and FAQ if you still have problems launching the server afterwards.

You are now ready to send commands to your robot through the iRobot URBI engine. For example try :

```
vars;
...
[00004004:notag] *** ledPlay = OBJ
...
```

The first line asks for all variables known by the server and the following lines, which you don't have to take care of for now, list symbols and their respective values.

Chapter 3. First commands

Forewords

This chapter will introduce basic concepts for the use of the iRobot Create URBI engine. It may be skipped if you are already familiar with URBI engines.

For further instructions and informations on URBI commands, please read the first chapters of the URBI tutorial [http://www.gostai.com/doc/en/urbi-tutorial-1.0].

Before anything else, make sure the steps presented in the Quickstart chapter succeeded.

Device quick list

Here is a quick list of the devices available on the iRobot Create which you can control through the engine.

If you have a closer look at the `groups.u` and `aliases.u` file distributed with the URBI engine, you will see how theses devices are grouped and aliased for direct and simpler access.

Table 3.1. Devices' characteristics

Name	Type	Description
battery	Sensor	Battery status (charge, current...)
bumperL	Sensor	Left bumper
bumperR	Sensor	Right bumper
buttonA	Sensor	Advance button
buttonP	Sensor	Play button
cargobay	Sensor/Actor	Extra I/O
cliffL	Sensor	Left cliff sensor
cliffFL	Sensor	Front left cliff sensor
cliffFR	Sensor	Front right cliff sensor
cliffR	Sensor	Right cliff sensor
command	Actor	Command interface
ir	Sensor	Infra red receiver
ledAdvance	Actor	Advance button led
ledPlay	Actor	Play buttun led
ledPower	Actor	Power button led
lsd0	Actor	Low Side Driver 0
lsd1	Actor	Low Side Driver 1
lsd2	Actor	Low Side Driver 2
odometry	Sensor	Odometry wrapper
speaker	Actor	Robot speaker
virtualwall	Sensor	Virtual wall sensor
wall	Sensor	Wall sensor
wheelF	Actor	Front wheel
wheelL	Actor	Left wheel
wheelR	Actor	Right wheel

Simple commands

Try this first command in your telnet session :

```
ledPlay;
[00000476:notag] 0.000000
```

The first line asks for the value of the play button led. The second is the answer from the iRobot engine. The first integer value is a timestamp for the answer (in miliseconds from the engine startup). It is followed by a `tag`. Please refer to the URBI tutorial [http://www.gostai.com/doc/en/urbi-tutorial-1.0] for mode details. What is important here is the floating point value ending the line. It is the value stored in the led object. Here *0.000000* indicates the led is switched off. Setting a value of *1* will switch the led on.

Do not forget the `';'` at the end of the commands. The URBI server will wait indefinitely for the command to wait if omitted.

The next commands will introduce the `+end` feature. Enclosing you commands in `+end:{...};`. Will make the engine warn you when a command comes to an end.

Try the following :

```
+end:{wheels = 100 time:2s | sleep (2s) | wheels = 0;};
[00002622:notag] *** end
```

`wheels` is a device groupe for the left and right wheels of the robot. Your robot should speed up to `100 mm/s` in 2 seconds, and then go forward maintaining this speed for 2 mode seconds. The robot then stops.

You may now try to make it come back to its initial position with a similar command.

Next, try this :

```
+end:{wheelL = -150 & wheelR = 150 | sleep (2s) | wheels = 0;};
[00005095:notag] *** end
```

Your robot should turn on itself for 2 seconds. This command tells the robot to set both its wheels speed at the same time, and then to wait for 2 seconds before full stop.

The last simple command will be :

```
bumpRDetect:at (bumperR != 0) echo "Collision on right side";
```

Now try to put press the front right of your iRobot. You should get a message like :

```
[00008215:notag] *** Collision on right side
```

The `at` keyword allows you to detect events and make your iRobot Create act accordingly. The `bumpRDetect:` is called a `tag`. It is used to give a name, or an identifier to a command. You can use special commands like `stop, freeze` and `unfreeze` to manipulate running commands. Try `freeze bumpRDetect;`. The previous message should not be printed anymore when bumper is pressed. The same command with `unfreeze` will bring it back on.

You should now be able to play with the iRobot URBI engine. The next chapters will give you complete details about the engine, but what we saw here is enough to get used with the iRobot and URBI.

Once again, you can get complete URBI tutorial there tutorial [http://www.gostai.com/doc/en/urbi-tutorial-1.0].

Chapter 4. iRobot Create Devices

Forewords

This chapter will list all the objects (UObjects) defined by the iRobot Create URBI engine. It is a full list giving only main attributes or methods of these objects or devices.

In any case, typing the device name in a connection to the URBI server should display the whole attribute list (you may need to type `unalias dev` if the particular device has an alias set on).

Every devices and attributes described in iRobot Create Open Interface Specification [http://www.irobot.com/sp.cfm?pageid=294] should be accessible from the URBI engine.

Please note that `config.u` file has a `system.aliasmode = 1;` line. When value is not `0`, more aliases are defined in `aliases.u`. In the following we will considered it is the case.

Motors

Two motors are defined, `wheelL, wheelR`. They are grouped in `wheels` for faster access when the given speed is the same for both wheels.

Wheels speed go from `-500mm/s` to `500mm/s`.

Each motor is aliased to its `.val` attribute. This atribute gives the current wheel speed in mm/s:

```
wheelL = 10;
```

is equivalent to

```
wheelL.val = 10;
```

The `drop` attribute holds the current state of the wheel. It is `0` if wheel is on the floor, `1` if the wheel has no ground contact.

Sensors

By default, all sensors are aliased to their attribute `.val`.

Sensors are grouped by type in `buttons, walls, cliffs` and `bumpers` groups. They are also all grouped in the `sensors` group.

Please keep in mind that groups and aliases are there to give simpler access to most useful information on devices. You may want to tweak `aliases.u` and `groups.u` files to adapt the robot device layout to your personal use.

Bumpers

Bumpers `bumperL` and `bumperR` are input devices set on the front and side of the iRobot Create.

They have a value of `1` if pressed, `0` otherwise.

Wall Detector

The wall detector is refered by the `wall` object.

Its value depends on how well the wall is detected and does not refer to any distance. A black wall will have a lower signal than a white wall even if detected at the same distance.

Virtual Wall Detector

The wall detector is refered by the `virtualwall` object.

Its value is set to `1` if a virtual wall is detected, `0` otherwise.

Cliff Detectors

The cliff detectors are respectively called `cliffL`, `cliffFL`, `cliffFR`, `cliffR` depending on their position on the robot (left, front left, front right, right).

Their value detects surfaces under the robot. `0` is for ground, `1` is in case of a possible fall.

The cliff detectors are grouped under `cliffs`.

The `signal` attribute gives a value helpful to get the distance from the sensor to the ground.

Battery

The battery device is called `battery`.

The battery object gives the following informations about the iRobot battery. The `charge` attribute in mAh, the estimated battery and the `capacity` attribute in mAh. The battery internal `temperature` attribute and its power `source` attribute (none, power station or adapter).

It also indicate the battery current `voltage` in mV and its current (`val` or `current`) in mA.

The `state` indicates if the battery is charging.

Buttons

The button devices are called `buttonP` for the Play button and `buttonA` for the advance button.

Button values are `1` when pressed, `0` otherwise.

Speaker

The device `speaker` is used to produce music notes.

Once music is stored into the `speaker`, a call to its `play()` method will produce notes.

The following commands :

```
speaker = [69,60,73,32];
speaker.play();
```

Will play note 69 : A (440Htz) with a length of 1s (60/60s) and the note 73 : C# (554.4Htz) for half about a second (32/60s).

Full note code list is given in iRobot Create Open Interface Specification [http://www.irobot.com/sp.cfm?pageid=294]

The list given as a value of `speaker` or `speaker.val` is a succession of `note / duration` couples composing a song. Note durations are given in 60th of a second.

Songs can only be 16 note long.

The command `speaker.play()` returns immediatly. If you launch another song while the song is still playing, the command will be ignored.

Command

The `command` UObject is not a real device. It is a software interface used to send direct commands to an iRobot Create.

This object offers to change the robot `mode` attribute. While `Full` mode is used most of the time, you may need to switch to `Safe` mode or any other defined mode in iRobot Create Open Interface Specification [http://www.irobot.com/sp.cfm?pageid=294]

`command` can also be used to handle builtin demos with `command.launchDemo(id)` (starts demo with given id) and `command.stopDemo()` (stop current demo, same as starting demo with id `-1`).

This object also has methods to send and receive data directly to the iRobot Create : `command.send(byte_list)` and `command.request(byte_list, answer_size)`

For example :

```
command.send([136, 1]);
```

will launch the Create demo 1.

And :

```
command.request([142,8], 1);
```

will return information on the wall detector sensor.

These functions can be used to access any device or feature specified in iRobot Create Open Interface Specification [http://www.irobot.com/sp.cfm?pageid=294], and thus features you may add to you robot.

Please note that

- Integers in list are in [|0, 255|]
- Expected answer size must be exact
- `[]` (empty list) is returned when request fails.

These commands are designed for expert users. If you are new with URBI or iRobot Create, standard devices and methods should provide access to everything needed.

Please refer to iRobot Create Open Interface Specification [http://www.irobot.com/sp.cfm?pageid=294] in order to get the command codes and expected return values.

Odometry

The `odometry` Uobject is a special device that grabs the postion of the iRobot Create in its environnement.

In the `odometry` variable are stored :

- the distance (`val` in mm) that the Create has traveled since URBI server was launched
- the `angle` rotated since server launch
- the last requested `radius`, `angle` and `velocity`

You can use the `odometry.move(velocity, radius)` function to control your robot. It is a direct access on command 137 in iRobot protocol.

```
odometry.move(40, -200);
```

This example will make your robot move at a global velocity of 40 mm/s and turn on the right following the arc of a circle of a radius of 200;

Chapter 5. URBI function Examples

Forewords

This chapter describes basic functions bundled with the iRobot engine.

You can access their source code in the `data` directory provided with the engine. You are free to use them to build advanced behaviours for your iRobot Create.

Movement functions

These first functions are used to make your robot move. Odometry support has still to be improved, so, if used in proper conditions, time based functions are more precise at the moment.

Odometry based

The functions `func.rotateAngleOdo (angleNeeded, velocity)` and `func.moveStraightOdo (distance, velocity)` make the robot move based on the odometry informations sent by the robot.

The `func.rotateAngleOdo` function makes the robot rotate on itself of a given angle, at a given speed.

The `func.moveStraightOdo` function makes the robot follow a straight line at a given speed on a given distance.

If speed is not 0, the robot moves. If not, it only waits for the odometry value to be the one expected.

As said previously, these functions may not be precise depending on how you use your URBI server. A server period of 100ms seems to give not so bad results.

Time based

The following functions `func.rotateAngleTime (angle, radius, velocity, go)` and `function func.moveStraighTime (distance, velocity, go)` make the robot move based on the theoretical time needed to complete a move.

Note that in this implementation, `func.rotateAngleTime` allows to set the rotation radius.

The `go` argument activate move. If 0 is given, the function return the time to wait in ms for the movement to complete. If not 0 is given, the robot effectively moves before the time beeing returned.

This implementation has proved to be more precise than the previous one. But please note that it is so only in good conditions, meaning external interaction with the wheels will not be taken into account.

Demo functions

Builtin demos

The function `func.demo(demoId, duration)` gives access to builtin demos.

This implementation allows to set a demo timeout. If `duration` is not 0, the demo is stoped after given time of execution. If `duration` is 0, the demo will run indefinitely.

Please note that launching a demo set the robot in `Passive` mode, you may want to set it back to `Full` on demo completion.

Simple demos

The extra function `func.blinkingLeds (factor, loops)` plays a short demo using robot leds.

The `factor` change the demo cycle duration into `1/factor second`, and the demo runs for `loops` cycles.

Chapter 6. Troubleshooting and FAQ

Troubleshooting

Frequently Asked Questions

The FAQ for URBI iRobot Create engine can be found at : www.gostai.com/irobot_faq.html [http://www.gostai.com/irobot_faq.html].

You may also be interested in having a look at our forum at : forum.gostai.com/ [http://forum.gostai.com/]

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, MERCHANTIBILITY, 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.