



# Robot documentation

---

CIMA

TI5 - 2015 / 2016

**Written by**

ABDELKADER Malik

DA SILVA Frédéric

**Reviewed and corrected by**

DA SILVA Frédéric

February 5, 2016

## Table of contents

1. [Robot code documentation](#)
2. [How to run server on EV3](#)
3. [Robot test unit](#)

# 1. Robot code documentation

**Packages robot server** (This package has not been modified)

- net.sourceforge.kitteh
- net.sourceforge.kitteh.impl
- net.sourceforge.kitteh.impl.ws
- net.sourceforge.kitteh.producers
- net.sourceforge.kitteh.utils

---

**Package main :**

- robot

We will focus on the robot package :

**Interface interface\_robot**

It's a generic interface who call the **getAllcapacite()** function which retrieve the robot capabilities.

**Class Capacite.java**

It's a generic class who stock the robot capabilities

**String URI** : Uri of the capability.

**String shorturi** : Short Uri of the capability.

**DocumentProducer dp** : Function which is call when the Uri of the capability is call.

**String sensormotor** : Variable which show if the capability is for a motor or a sensor.

**String id** : Id of the capability.

**String hasresult** : Variable which inform if the capability return a value or not.

**String type** : Variable which give the type of the return value.

**String protocolName** : The protocol use to reach the capability.

**String port** : The port use to reach the capability.

**String method** : The method use on the capability.

**String body** : The body of the capability.

**Map<String,String> idp\_type** : The parameters of the capability and their types

### **Class robot.java :**

This is the main class, it has two function

- **main**

This function create an interface of the actual robot type (for example : new EV3() for an EV3). It create all the different URI (rules) from the capability retrieves by the function **getAllcapacite()**.

- **produceDocument**

This function create the XML document which regroup all the robot capabilities which are exposed on the “/infos” path.

### **Classe EV3.java**

This class is only for EV3 (you have to make one for every type of robot you want to use). It contains all the code for detecting motors and sensors plugged and create all the capabilities of the robot in the function **getAllcapacite()**. It also detect all the sensor plugged modes and create a capability for each one.

### **Classe Forward.java / Backward.java / Stop.java / Rotate.java**

These class are for EV3 only, they call some functions of the EV3 motors to make an action when the correspondent URI is called.

### **Classe Color.java / IR.java / Gyro.java / Touch.java / US.java**

These class are for EV3 only, they call some functions of the EV3 sensors to get a value when the corresponding URI is called. They also retrieve value from the different modes of the sensors when the corresponding URI is called.

### **Classe Exit.java**

Stop the server when the URI /exit is call.

---

## **2.How to run server on EV3**

Once the robot EV3 code recovered (file CIMA.jar), there are several ways to copy it into the EV3 robot:

- **SSH connection:**

(1) first open a new terminal (under linux).

(2) verify that you are connected to the same router as the robot EV3.

(3) retrieve the IP address of the EV3 robot that appears on robot screen.

(4) enter the command:

"scp Path\_to\_the\_file/CIMA.jar root@robot\_addressIP:lejos / samples"  
in the terminal, then press "enter" when asked for the password.

- (5) on the robot EV3, press twice on the right for you arrow place on "Samples" and press the middle button to validate.
- (6) select the CIMA.jar file dropdown with down/up arrow and press the middle button to validate.
- (7) start the execution of the program, a blank screen appears on the robot EV3, meaning the server is properly started.
- (8) in order to leave the server just enter "robotAdressIP:8080 / exit" in your browser bar.

**- Copy on the MicroSD card:**

- (1) connect the MicroSD card to your PC/MAC.
- (2) place you in the "lejos/samples" from the MicroSD card folder.
- (3) copy the "CIMA.jar" file to this location.
- (4) plug the MicroSD card on the robot EV3.
- (5) repeat the procedure "SSH Connection" from step 5)

### **3. Robot test unit**

**Unit test 1 :** Display the XML with EV3 capabilities on "/infos" → **OK**

**Unit test 1.1 :** Validation of the generate XML → **OK**

**Unit test 2 :** Testing capabilities forwarding of motor port A on "/A/forward" → **OK**

**Unit test 3 :** Testing capabilities backwarding of motor port A on "/A/backward" → **OK**

**Unit test 5 :** Testing capabilities rotate of motor port A on "/A/rotate" → **OK**

**Unit test 6 :** Testing capabilities stop of motor port A on "/A/stop" → **OK**

**Unit test 7 :** Testing capabilities forwarding of motor port D on "/D/forward" → **OK**

**Unit test 8 :** Testing capabilities backwarding of motor port D on "/D/backward" → **OK**

**Unit test 9 :** Testing capabilities rotate of motor port D on "/D/rotate" → **OK**

**Unit test 10 :** Testing capabilities stop of motor port D on "/D/stop" → **OK**

**Unit test 11 :** Testing capabilities sensor EV3UltrasonicSensor port S1 on

"/S1/EV3UltrasonicSensor" → **OK**

**Unit test 12 :** Testing capabilities mode "Distance" on sensor EV3UltrasonicSensor port S1 on "/S1/EV3UltrasonicSensor/Distance" → **OK**

**Unit test 13 :** Testing capabilities mode "Listen" on sensor EV3UltrasonicSensor port S1 on "/S1/EV3UltrasonicSensor/Listen" → **OK**

**Unit test 14 :** Testing capabilities sensor EV3GyroSensor port S4 on "/S4/EV3GyroSensor" → **OK**

**Unit test 15 :** Testing capabilities mode "Rate" on sensor EV3GyroSensor port S4 on "/S4/EV3GyroSensor/Rate" → **OK**

**Unit test 16 :** Testing capabilities mode "Angle" on sensor EV3GyroSensor port S4 on "/S4/EV3GyroSensor/Angle" → **OK**

**Unit test 17 :** Testing capabilities mode “/Angle and Rate” on sensor EV3GyroSensor port S4 on “/S4/EV3GyroSensor/Angle and Rate” → **OK**