

# Humanoids

Laia Freixas

May 2017

## 1 Introduction

This document explains the basic tools in order to work with the humanoid bioloid robot by using the new IRI code. The catkin package is `bioloid_controller.cm510`. This document assumes all the code is correctly installed with the IRI dependencies met and having ROS indigo. You can find the installation instructions for this in the following documents :

1. Ros Indigo installation
2. Labrobotica Software installation
3. Bioloid robot github repository

The developing strategy for the bioloid with this software consists of having a C file which contains the state transitions defined for the main loop of the robot. When compiling the code with the `bioloid.cm510_controller` package we must specify in the `CMakeLists.txt` where that file is located.

## 2 Folder content

### 2.1 communication

### 2.2 controllers

### 2.3 dyn\_devices

### 2.4 motion

### 2.5 examples

Here there are several examples in order to understand the usage of the robot.

## 3 Using the Gazebo simulator

### 3.1 Editing CMakeLists.txt

In order to edit the CMakeLists.txt file we have to run the following commands in terminal.

```
roscd bioloid_controller_cm510
gedit Cmakelists.txt
```

Then we have to edit the following line and set ProjectPath to our folder.

```
SET(ProjectPath ~/path/to/myfolder)
```

### 3.2 Compiling and launching

The compilation is done through the catkin environment, so in order to compile correctly we must do

```
roscd
cd ..
catkin_make --only-pkg-with-deps bioloid_controller_cm510
```

Once the code is compiled, the following instruction will execute the launch file that runs the simulator with its basic components

```
roslaunch bioloid_apps ceabot_base.launch
```

### 3.3 Adding new environments and changing speed

The folder in which we can find the new environments is

```
~/iri-lab/iri_ws/src/bioloid_robot/bioloid_description/
  urdf/ceabot
```

The environments represent the different tests for the CEABOT competition, which are stairs, obstacles or vision. In order to load them, we must specify so at runtime:

```
roslaunch bioloid_apps ceabot_obstacles.launch
```

If we want to change the speed - faster or slower, we have to run the following commands

```
roscd bioloid_gazebo
cd worlds
gedit bioloid.world
```

```
<real_time_factor>0.1</real_time_factor>
<real_time_update_rate>100</real_time_update_rate>
```

### 3.4 Changing the robot model

The folder in which we can find the robot models is

```
~/iri-lab/iri_ws/src/bioloid_robot/bioloid_description/  
urdf
```

The .xacro files represent different robot models, in which the sensor disposition -among other things- can be modified. To do so, simply edit the .xacro file, and specify the position, orientation and type of sensors you want to add to your robot model. Finally, we can load the model in the simulator as follows :

```
roslaunch bioloid_apps ceabot_base.launch  
robot:=myrobot
```

## 4 Working with split pages

There are several actions that must be done in order to work with split pages. Firstly, in the program, inside the `user_init()` function, we must add a call to `mtn_lib_init()` . In the compilation step, we must edit the `CMakeLists.txt` in order to compile it with the split pages, adding to the files the `mtn_library.c` . When launching the simulator, we must specify that, so we will write

```
roslaunch bioloid_apps ceabot_base.launch mtn_file:=ceabot_motions
```

Finally, in order to upload the split pages onto the robot, we must use the following command from the appropriate folder :

```
mtn_downloader -d <serial device> -m <mtn_file>
```

which could be, for example :

```
mtn_downloader -d /dev/ttyUSB0 -m ceabot_motions_real.mtn
```

To download the rest of the program to the robot, we must do it from our program's directory and execute

```
make download
```

## 5 Expansion board

### 5.1 Updating the expansion board

Sometimes, the software in the expansion board changed on disk and must be updated on the robot. The steps in order to complete it are the following :

1. Download or update the repository `bioloid_exp_board_fw`.
2. Compile the code (`make`)
3. Connect an USB-MiniUSB from the computer to the expansion board

4. Make sure the device is recognized as `ttyUSB0`
5. Turn on the robot, and before one second, run the command "make download".

## **6 Useful information**

### **6.1 minicom**

Minicom is the serial interface program used to establish a communication between robot and computer when the program is running. Minicom can't be running when we download the program to the real robot because the serial interface doesn't allow this.

## **7 Conclusion**

## **References**