BUZZ and ARGOS

Argos:  
  
**Installation:**

Github: <https://github.com/ilpincy/argos3>

Clone the repository.

* Compiling Argos Simulator for PC

$ cd argos3

$ mkdir build\_simulator

$ cd build\_simulator

$ cmake ../src

$ make

* Compiling Argos on Real Robot (not for PC)

$ cd argos3

$ mkdir build\_myrobot

$ cd build\_myrobot

$ cmake -DARGOS\_BUILD\_FOR=myrobot ../src

$ make

* Compiling Documentation

$ cd argos3

$ cd build\_simulator # or 'cd build\_myrobot'

$ make doc

**Examples:**

You can download Argos3 examples from the following link:

https://github.com/ilpincy/argos3-examples

**Problems:**

There may be a problem in running argos3 simulation examples as it may not be able to find the ‘build’ directory in usr/local/lib/argos3 due to which it wont be able to initialise the controllers.

To resolve this:

create a link directory “build” from:

usr/local/lib/argos3/build\*

to:

/home/<your\_pc\_name>/argos3-examples/build/

\* indicates linking directory

Buzz:

**Installation:**

Github: <https://github.com/MISTLab/Buzz>

* Compilation

$ cd buzz/build

$ sudo make install

* Installation

$ cd buzz/build

$ sudo make install

$ sudo ldconfig <- The last command is only for Linux OS

Sample Run:

1. Download the buzz and argos scripts
2. Compile the buzz scripts using bzzc command
3. Add the .bo and .bdb files generated after compilation to the <params> tag in the .argos file
4. Run the argos file using the command:

argos3 -c <your\_file\_name>.argos

**Note:**

If you get the .argos file from any other source than the one we have uploaded alongwith, make sure to check for the media tag as it is necessary for swarming. Media refers to the communication channel and an incomplete media tag can lead to errors during execution

**Additional Points I found useful to inform:**

1. /usr/local/include/buzz/argos/ (Location of buzz\_controller\_footbot.h to refer to functions)
2. There is no setleds function as used in examples. It is set\_leds.. refer to the .cpp file in above path for better details
3. There is no debug function as used in examples. Debug is a library and print is the function. So use debug.print(<value\_you\_want\_to\_check>
4. To include libraries, type at the top of the buzz file,

include “<file\_with\_complete\_path>”

1. Location of the library files:

/usr/local/share/buzz/include/

1. Location of buzz\_controller\_footbot.cpp:

/usr/local/lib/argos3/build/controllers/buzz\_controller\_footbot