7/11/2018

Rick Jansen

Gibas automation b.v.

FestoConnect URcap

Manual 1.0

Contents

[1. Installation 2](#_Toc522543277)

[2. Explanation usable program Node's 3](#_Toc522543278)

[3. Example program node structure: 5](#_Toc522543279)

[4. FOR INTERNAL USE: Explanation URCAP 7](#_Toc522543280)

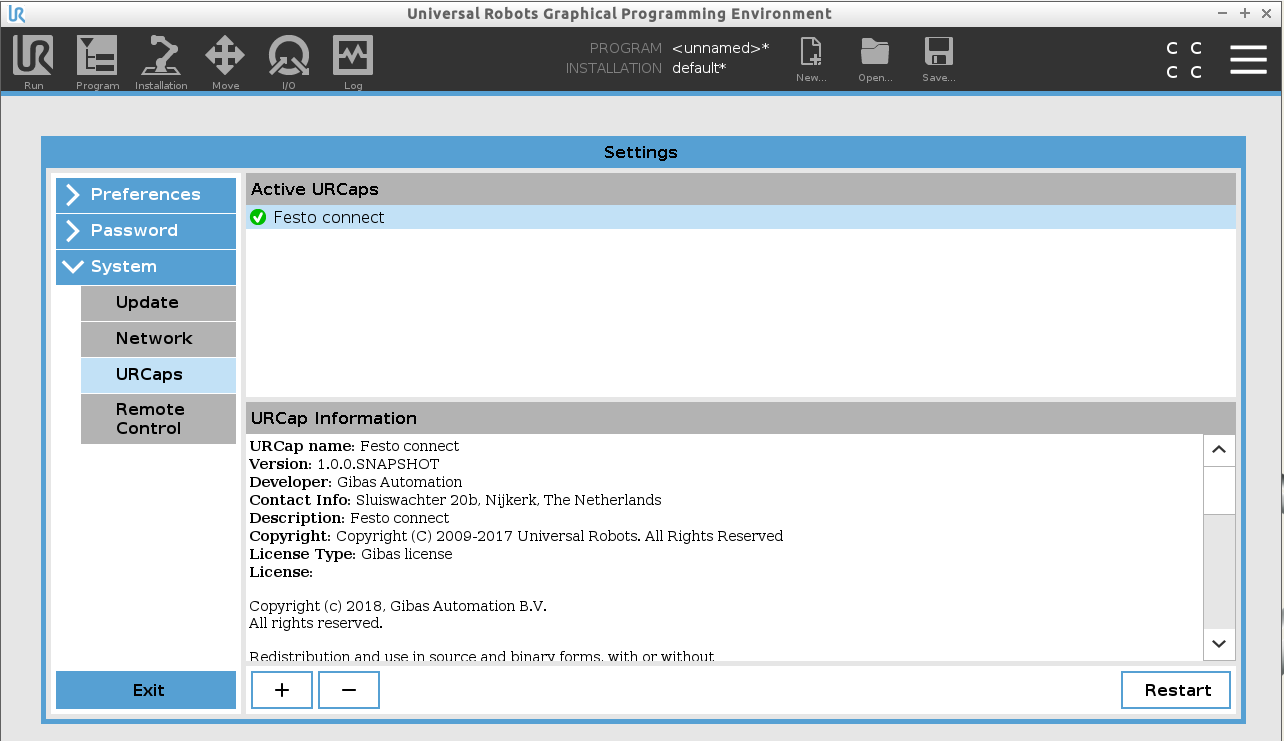
[4.1 Explanation structure 7](#_Toc522543281)

[4.2 Software packages 7](#_Toc522543282)

[4.3 installation software 8](#_Toc522543283)

# Installation

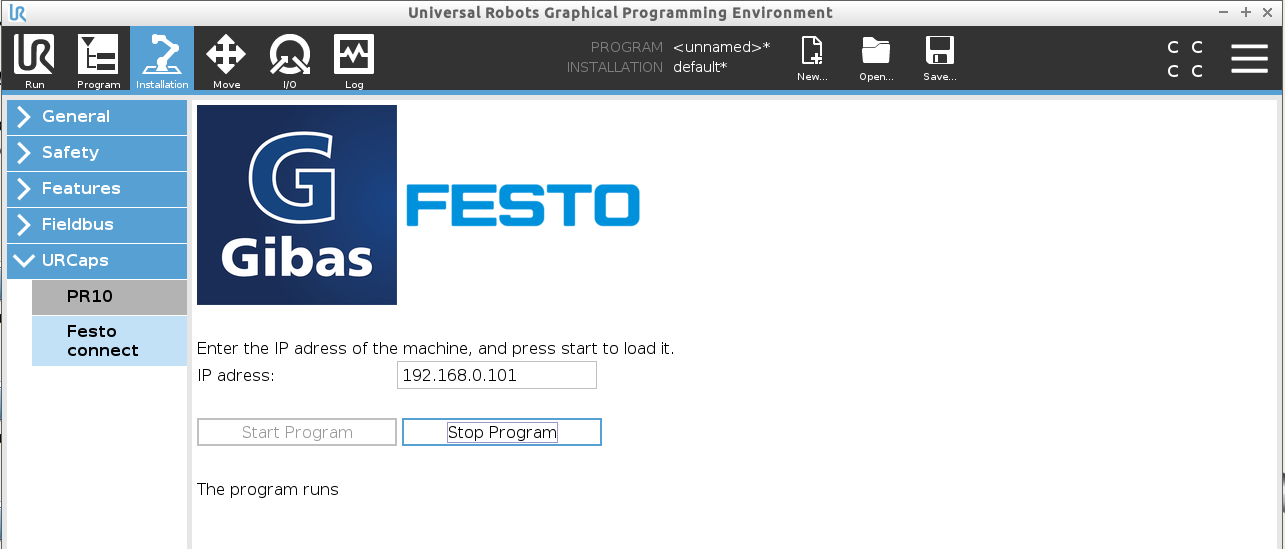
Obtain the URCAP Festo Connect on a USB stick, and plug it into a universal robot machine.

The following illustration shows the installation screen.

Restart the UR, and the URCAP is installed. No further software is required.

# Explanation usable program Node's

*Installation node:*



On this page the connection with the Festo is set up. This is done by stopping the program, adjusting the IP address to that of the local Festo machine and starting the program again.

With the current URCAP version it is only possible to connect with a single Festo drive.

*Homing node:*



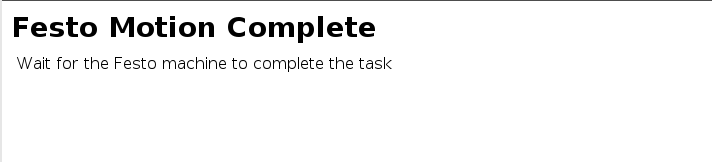
This will “home” the Festo (this position is set in the Festo Connect Tool), which is best to be placed in the Before start of the UR program. Doing this on startup will calibrate the Festo drive’s position to ensure accurate movement during the program.

*Motion node:*



This is the program node that sends the move command to the Festo, where a variable with the speed and position can be used. These must be created before this node, to be able to select them from the drop-down menu. It is possible to select every variable created by the program, it does not require a specific name for it to work.

*Motion complete node:*



This comes after the Motion node in the program structure. Like the name implies, this node will halt the Universal robot program until the Festo reaches the desired position. With this node, more complex movements can be made. Example: place a movement of the UR itself in-between the motion & motion complete node to make a joint motion. This also ensures that both have finished before continuing the program.

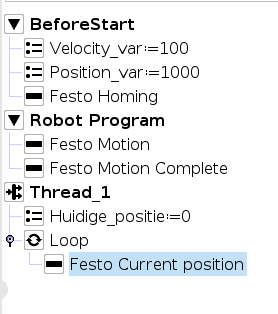
*Current position node:*



This obtains the current position of the Festo and writes this position (in millimeters) to a variable chosen by the operator.

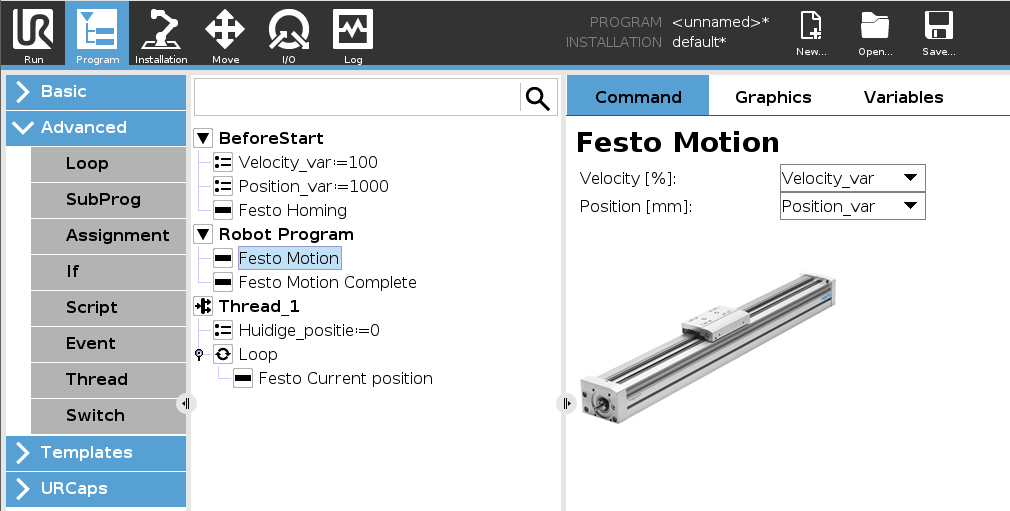
# Example program node structure:

To explain the necessary structure of this URCAP, here is a small example program to make a movement with the Festo drive.



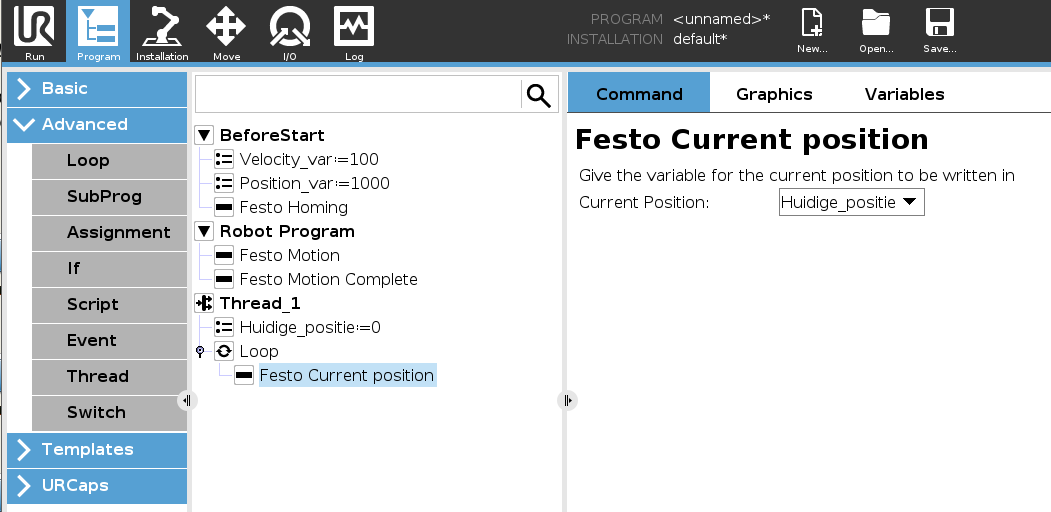
The position and speed variables are created at first, although this can also be done just before the Festo motion in the robot program itself. Homing of the Festo is recommended to place in the Before start, as this only needs to be done once.

the motion node is shown below.



Now that the position and speed are sent, the motion complete node halts the program until the Festo drive is finished positioning. TIP: It is therefore also possible to put a movement of the UR in between to move at the same time.

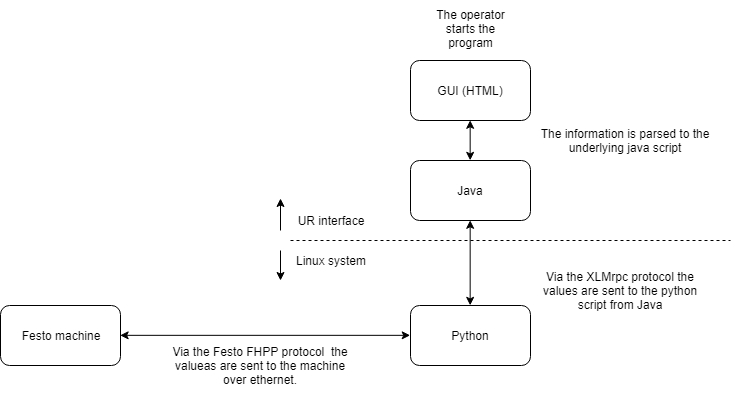
Finally, there is the position feedback, which must be put in a thread to be able to update in real time. First the variable is created, and the loop afterwards ensures that the program continues to update the position.



This is all that is needed to send a Festo drive to a specified position.

# FOR INTERNAL USE: Explanation URCAP

## Explanation structure

This is a brief overview of the structure from this URCAP.

## Software packages

Since the (URCAP) software already comes with a manual from UR itself (very handy to learn the structure of the software needed) and my code has description with every function, it will not be explained here. See the source code or the URCAP manual for more information.

Also in the software package (the starter kit provided by UR) There are several examples (Hello World, Mydeamon, etc.) in which it is easy to recognize the structure and functioning of it. It is advisable to start with these as the structure of the FestoConnect project is a lot more complex.

## installation software

The software needed to create your own URCAP is already present, in the variant of a VirtualBox system. Here is the following software:

The root password of Ubuntu is admin.

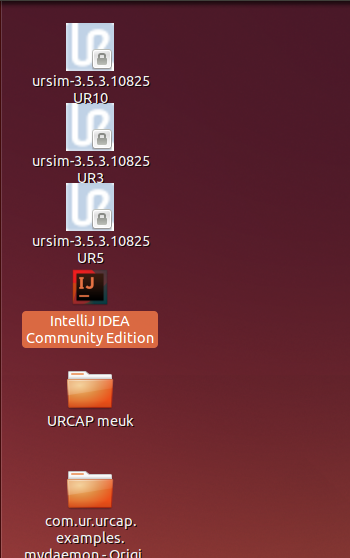
* Intellij IDEA (Here the URCAP software is written in.)
* Ursim 3.5.3 (To simulate & test the URCAP)
* UR SDK 1.2.65 (The compilation tool needed to build the project)

Location VirtualBox Image:

Z:\Automation\Data\Personeelsmappen\Rick\Virtualbox URCAP\_Dev Image.

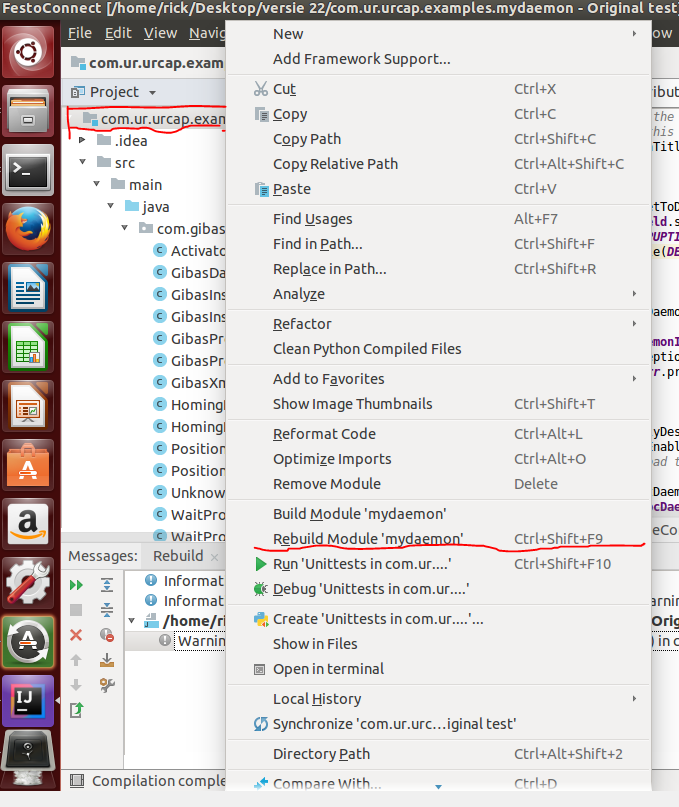
This VirtualBox OS is built up from the ground up to meet all the functionalities to create a URCAP. The VirtualBox Ursim provided by UR is a 32-bit system, but the software that compiles the URCAP (in the SDK) only works on 64-bit systems. Therefore, it was necessary to create this ubuntu operating system.

**How to make a URCAP:**

1. Open Intellij Idea

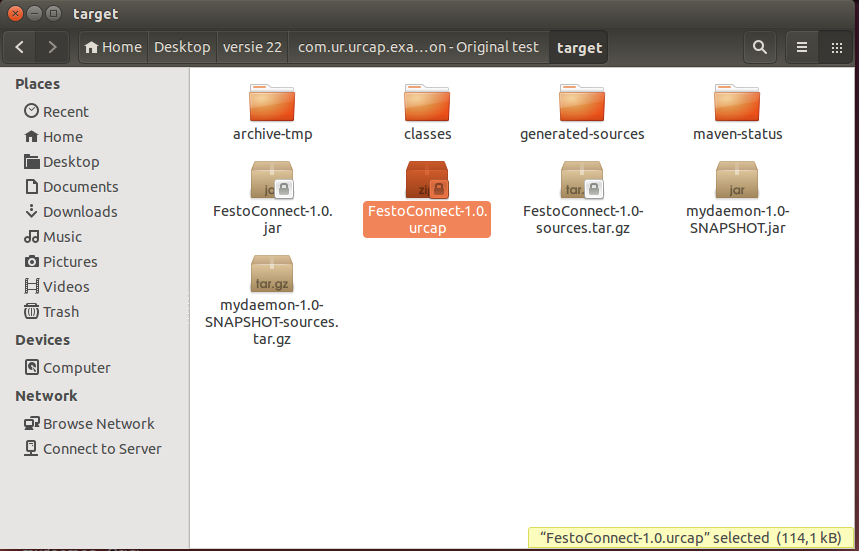
This will load the FestoConnect project.

1. Perform customizations to the program and do a rebuild to process it.



1. Check which folder the project is in

(In this case /home/rick/Desktop/versie 22/COM. . . . )



1. Delete the current URCAP file (as it is sometimes not overwritten by the compiler)

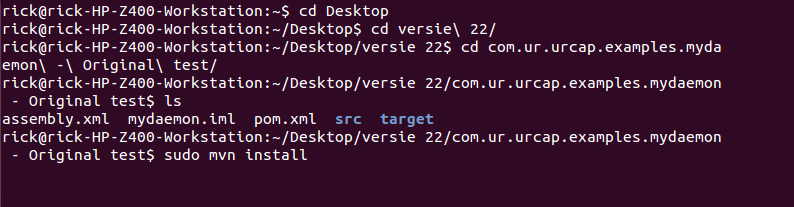
(in the folder “target”, file: FestoConnect-1.0. URCAP)

1. Compile the URCAP:

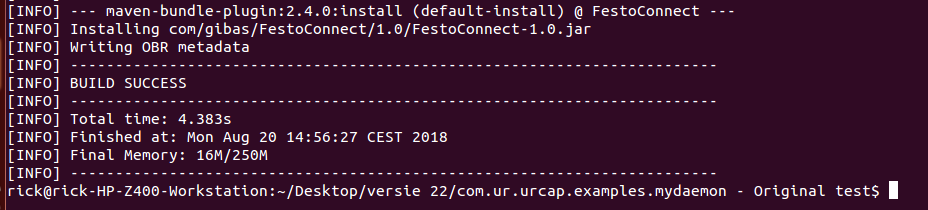
Open a terminal and go to the project folder:

(CD /Desktop/versie 22/COM....) (to the main folder of the project, not the subfolders)

Now type: **sudo mvn install**



(you can press TAB to autocomplete the folder names).



If the build info code (the pom.xml) is correct, it should show “BUILD SUCCES”. This does not check the program code itself.

This will create the new URCAP and put it back in the/target folder.

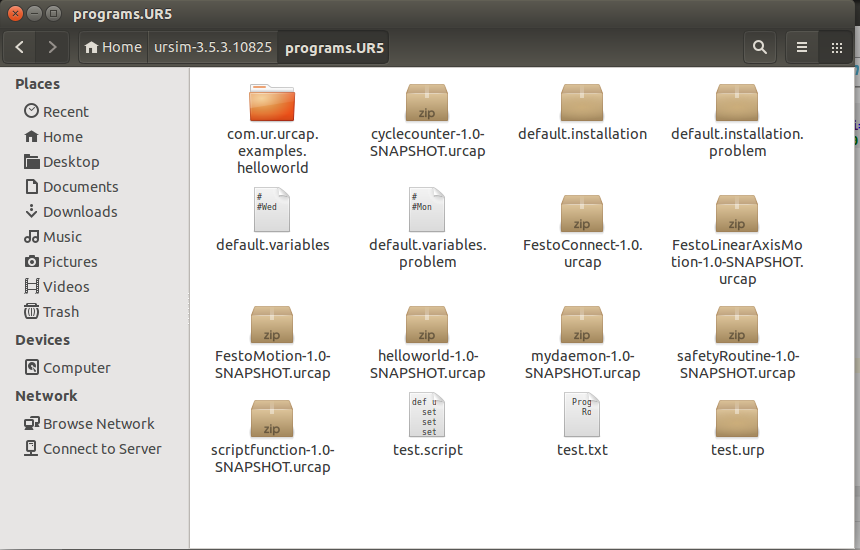
Now the URCAP is ready to be used in a Universal robot.

**Implement it in Ursim:**

To quickly test the urcap you just made. (although it is always better to test it on a physical robot because a lot of signals don’t convert well into a simulator).

1. Open both the target folder (where you just created the urcap) and the following folder:

/home/ursim-3.5.3.10825/programs.UR5



1. Copy the URcap from the developer folder to this folder. Also delete the previous URCAP in the folder because overwriting does not work well with .urcap files.
2. Go back to the desktop and open the simulator:



!! Make sure to open the UR5 version if you have placed the urcap in the programs.UR5 folder. !!

1. Now install the urcap into the universal robot as usual.