This page describes how to build and run the *Forklift* simulation - as the real Forklift robots are built from custom hardware and are not available for purchase.

(However, the code for running the application on real hardware is included. It is currently run on x86 tablets connected via CAN to a custom DSP board for hardware access and control. It should be possible to obtain plans in case you are interested. Apart from that, there's a certain level of interest to create a low-cost variant for e.g. home use among former group members.)

Check out and build the project

Prerequisites

You have prepared your system and a working copy of the **development branch** as described in Getting started.

Additional preparation

Third-party projects typically need some additional preparation steps:

Mercurial configuration

You need to add authentication information for RRLab to your `~/.hgrc` :

```
[auth]
rrlab.prefix = https://agrosy.informa
rrlab.username = anonymous
rrlab.password = your@email.address
```

Finroc repository

You also have to add RRLab's Finroc repository to your `$FINROC_HOME/etc/sources.list` :

```
~/finroc$ echo http://agrosy.informat
```

Ubuntu repository

When working on Ubuntu, you might want to add RRLab's Ubuntu repositories, too. It provides meta-packages for easier installation and i.e. ready-to-use packages for libraries that are not included in the official Ubuntu repositories.

```
~/finroc$ sudo su -
~# cat > /etc/apt/sources.list.d/rrlal
> deb     http://agrosy.informatik.un
> deb-src http://agrosy.informatik.un
> EOF
~# apt-keys adv --keyserver keyserver
~# apt-get update
~# logout
```

Required libraries

If you added the Ubuntu repository in the last step, the required libraries can easily be installed using RRLab's meta package:

```
~/finroc$ sudo apt-get install finroc
```

Otherwise install the following libraries on your system (with development support):

- libcoin (6.x if available)
- libdc1394
- libboost_filesystem
- libboost_iostreams
- libopencv
- libsimage
- libnewton (physics engine; not available in Ubuntu repositories; we use version 2.35)

Graphics drivers

Offscreen buffers in *libcoin* used to work pretty well with open source drivers (3D simulation and visualization in the forklift project are currently based on *libcoin*). Unfortunately, this seems to have changed in newer Ubuntu releases. The proprietary AMD and nVidia drivers typically work a lot better.

(On Ubuntu 13.10, AMD cards require proprietary drivers from the AMD homepage from 2014. On Ubuntu 14.04, the driver *fglrx-updates* from the Ubuntu repositories works.)

Getting the code

Now check out the project (and a GUI plugin for 3D visualization).

```
~/finroc$ finroc_get finroc_projects_
```

Build the project

Make the installed libraries known to Finroc's build system.

```
~/finroc$ updatelibdb
```

Then you can build the project with e.g.

```
~/finroc$ make -j4
```

Run the project

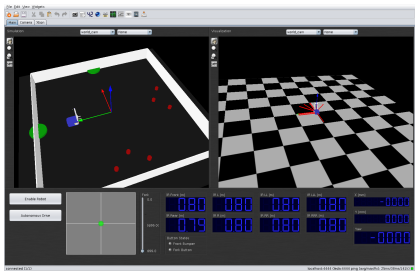
Plain simulation

After building succeeded, you can start the plain simulation (only remote control via the GUI) with

```
~/finroc$ forklift --simulation-standa
```

A standard GUI is also included.

```
~/finroc$ fingui sources/cpp/projects/
```



Higher-level Control System

The forklift executable can be run in various configurations. `--help` gives an overview.

```
~/finroc$ forklift --help
```

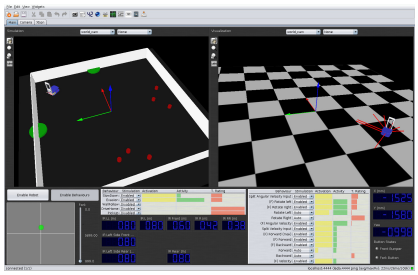
To run the simulation together with a higher-level robot control system from another project, use the `--control` command line parameter - e.g.:

```
~/finroc$ forklift --simulation-standa
```

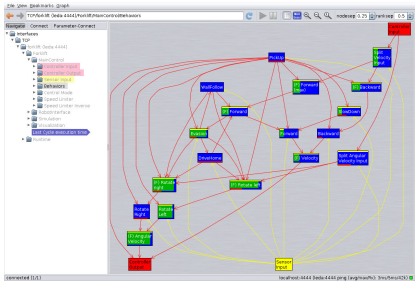
The *forklift_control_edu project* contains some convenient scripts (`start` , `stop` and `gui`) to start the Forklift simulation with the *edu* control system and the GUI. They are located in `sources/cpp/projects/forklift_control_edu/scripts` . `$FINROC_PROJECT_HOME/scripts` is automatically included in the path after calling `source scripts/setenv` .

```
~/finroc$ source scripts/setenv -p fo
~/finroc$ start
```

Notably, this *edu* control contains a simple iB2C network. The behaviors can be activated and deactivated in the GUI. You can watch how their meta signals change when you move the robot.



In finstruct, you can view the behavior network using the *ib2cView*. Again, watch how their meta signals change when you move the robot.



Create your own higher-level forklift control project

In case you are interested, there is a convenient template for creating your own higher-level forklift control projects:

```
~/finroc$ source scripts/setenv -p fo
~/finroc$ forklift_create_control_pro
```

This will create a new project folder with an empty group *gMainControl*. Its base class *gMainControlBase* contains the interface for your application (set of input and output ports). Create your components inside the *gMainControl* (graphically and/or in the code) and connect them to the group's inputs and outputs. After building your control you can test it with

```
~/finroc$ forklift --simulation-standa
```

Last modified on 07.05.2014 21:36:57

Attachments (4)