

Nearlab – NES

Technical

Reports

|  |  |
| --- | --- |
| **Title:** | Simulink Toolchain for Soft-Realtime performances |
| **Report Number:** | 190626 |
| **Author:** | Stefano Dalla Gasperina |
| **Author’s Email:** | stefano.dallagasperina@polimi.it |
| **Date:** | 2019.06 |
| **Brief Description** | This guide explains how to install Embedded Real-Time Linux Target to be used with Simulink blocks (Ubuntu 18.04 + Debian 9.5 on BeagleBoneBlack) |
| **Revision:** | 1.0 |

# Introduction

This repository holds the basic structure and, some utility scripts of the Soft-Realtime Simulink Toolbox (SRT).

This toolbox allows you to compile a Simulink diagram into a Linux executable which uses nano-sleep to ensure better real-time behavior.

This is done by the external **ERT\_Linux** target for the Simulink Embedded Coder.

ERT\_linux and the actual Simulink blocks, which use the Soft-Realtime provided by ERT\_Linux, are developed in separate Git repositories and included as Git submodules for convenience.

This guide guides you in the installation of ERT\_linux.tlc in MATLAB R2017b on top of Ubuntu 18.04. Executables can be both compiled by the Soft Real-time Toolbox to work in **Linux amd64** and cross-compiled for **Debian armhf** architecture embedded system (such as BeagleBoneBlack).

Development Computer Requirements

## Install Ubuntu Desktop

Download and install Ubuntu Desktop 64bit 18.04

<http://www.ubuntu.com/download/desktop>

Installation guidelines

<http://www.ubuntu.com/download/desktop/install-ubuntu-desktop>

## Add user with sudo rights

Add a dedicated user for development with sudo rights and add the user to the dialout group

(Important for Rehamove devices)

**TODO code**

## Install Git and Subversion

Open a terminal using CTRL+ALT+T

sudo apt-get install git

sudo apt-get install subversion

## Install MATLAB & Simulink

The current version requires a Linux 64 bit operating system (e.g. Ubuntu 18.04).

Besides MATLAB and Simulink, the following toolboxes are required:

\* [Simulink](https://www.mathworks.com/products/simulink.html) + Legacy Code Toolbox,

\* [Simulink Coder](https://www.mathworks.com/products/simulink-coder.html),

\* [Embedded Coder](https://www.mathworks.com/products/embedded-coder.html), and

\* (MATLAB Coder)

This toolchain is tested with MATLAB 2014b and MATLAB 2016b but should run with other versions too.

**Attention**: From MATLAB R2017b on, some changes have been done, thus a patch is needed to make the toolchain work.

# Development Computer Installation

## Retrieve source code

1. **Original source code**

Original code can be retrieved here (not updated - has been moved to GitHub):

git clone git://rtime.felk.cvut.cz/ert\_linux.git

1. **Complete Simulink toolchain (not suggested):**

git clone --recursive <https://github.com/worldwidemv/SimulinkToolchain.git>

ERT\_linux and the actual Simulink blocks, which use the Soft-Realtime provided by ERT\_Linux, are developed in separate Git repositories and included as Git submodules for convenience.

Otherwise the submodules will not be included and must be manually added by you.

Inside the main directory is the install script `srt\_InstallSRT.m`, which must be run with Matlab.

This script should generate a working setup with the new entry \_Soft-Realtime Simulink Toolbox - Available Blocks\_ in the Simulink library browser.

You will also have the new option `ert\_linux.tlc` for the \_System Target file\_ under \_Setup > Code Generation\_.

Please have a look at the [documentation](https://github.com/worldwidemv/SimulinkToolchain/wiki) for a more detailed description.

1. **Berlin Simulink Blocks:**

Retrieve the current ert\_linux target using the following svn command in a dedicated terminal

svn co <https://files.control.tu-berlin.de/svn/retrainer/simulink>

**username: public**

**password: 6dR.&7Qv3**

1. **Berlin Simulink Blocks updated from SVN**

Every time you want to download a new version of the blocks from the svn you should do it directly from matlab. You should go to the current folder panel in matlab. Go to the folder that contains the folder you want to update. Once the folder you want to update is selected, you should right click and select

source control-> update from svn

Normally the passwork is used automatically if you are connected to internet.

Once you have updated the block you should run the m.file having the same name of the block and then you should compile the testing Simulink model inside the block folder to see if it works.

Useful links:

<https://rtime.felk.cvut.cz/gitweb/ert_linux.git>

<http://lintarget.sourceforge.net>

## R2017b Patch

As mentioned before, starting MATLAB R2017b a patch is needed.  
A temporary copy is provided to work with Matlab R2017a/b at **TODO LINK**

Note that this copy will be removed as soon as a better solution is available!

Pavel Pisa created the patch (ert\_linux-matlab-2017a.diff) and sent it to Markus Valtin by e-mail.

## Install ert\_linux

Open MATLAB with super-user rights.   
Add ert\_linux\_R2017 folder to path.  
In MATLAB prompt execute:

cd <root>/ert\_linux/ert\_linux\_R2017

ert\_linux\_setup

From now on, Code Generation pane of Simulink Model Configuration Parameters should offer ert\_linux.tlc as System Target File (after pressing "Browse...." button)

## Check Toolchain

In MATLAB prompt execute:

mex -setup

Should give the following:

MEX configured to use 'gcc' for C language compilation.

Warning: The MATLAB C and Fortran API has changed to support MATLAB

variables with more than 2^32-1 elements. In the near future

you will be required to update your code to utilize the

new API. You can find more information about this at:

http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit-api.html.

To choose a different language, select one from the following:

mex -setup C++

mex -setup FORTRAN

Embedded System Requirements and Installation

## Install Debian 9.5

Format a microSD card of at least 32Gb. (SD Formatter Link)

Navigate to <https://beagleboard.org/latest-images> and install Debian 9.5 LXQT.

Use Etcher to flash the SD card. (Etcher Link)

## Update

Install cross-compiler on development computer

sudo apt-get install gcc-arm-linux-gnueabihf

sudo apt-get install g++-arm-linux-gnueabihf

arm-linux-gnueabihf-g++ -v

You may want to create alternatives for the cross-compiler so as to compile with the same version of gcc and g++

# References

Soft Real-Time Paper: <http://rtime.felk.cvut.cz/publications/public/ert_linux.pdf>

Linux Target: <http://lintarget.sourceforge.net/>