

HiL-Simulation set-up

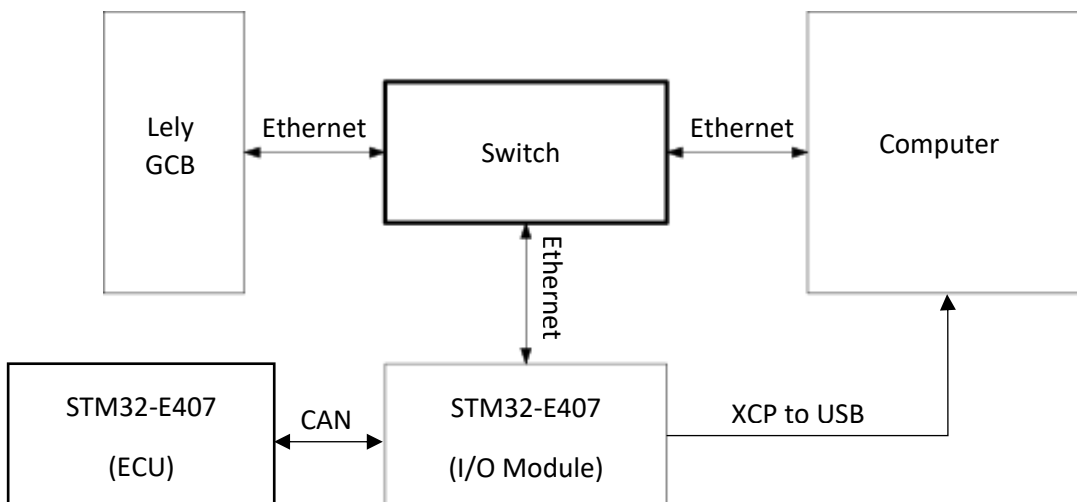
Guidelines for setting up a HiL Simulator

Introduction

This document describes the steps to be taken when setting up a HiL-simulator. The tutorial is written for a computer running on Windows 10.

The simulator described here uses the following set-up:

- Lely GCB (Simulation engine)
- STM32-E407 (I/O Module)
- STM32-E407 (ECU)
- Switch (To connect all components)
- Computer (programming simulation and displaying results)
- Set-Up:



All components are connected through ethernet. This is how the components communicate with each other.

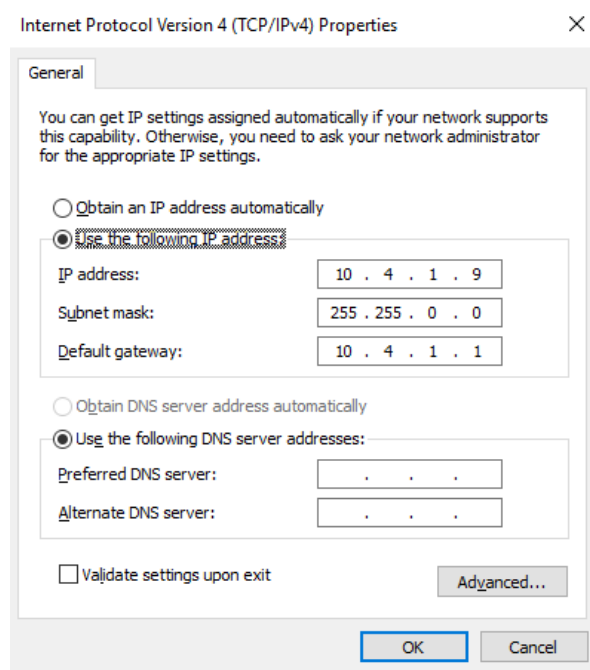
Necessary programmes:

- FileZilla: Used to transfer files to the Lely GCB
- Putty: Used to control the Lely GCB from the computer
- Linaro Binary Toolchain (arm-linux-gnueabi): Compiler to create models compatible with Lely GCB
 - o (<https://releases.linaro.org/components/toolchain/binaries/7.2-2017.11/arm-linux-gnueabi/hf/>)
 - o File: gcc-linaro-7.2.1-2017.11-i686-mingw32_arm-linux-gnueabi.tar
 - o *Note: WinRAR required to unpack .tar files*
- HANcoder: Used to generate code for flashing to STM32 and compiler for HANcoder. Install Guide: http://openmbd.com/wiki/HANcoder/STM32/Installation_Manual
- HANtune: Used to view results and progress of simulations. Install Guide: <http://openmbd.com/wiki/HANtune>

Steps for Installation

The first time installation of the HiL set-up contains several steps. Some of these steps require administrator permissions.

1. Connect the components as seen in the schematic above.
2. Set a fixed IP-address for the connection:
 - a. Select the network adapter options for the ethernet port at which the switch is connected.
 - b. Right-Click -> 'Properties' -> Select 'Internet Protocol Version 4 (TCP/IPv4)' -> Click 'Properties'
 - c. Fill in the following data to the window:
 - d. Now the PC knows where to find the switch and Lely GCB.



3. Add compiler path for Linux:
 - a. Download Linaro Binary Toolchain
 - b. Extract the .tar file to a path without spaces in the name
 - c. Copy the path to the bin folder. Example: C:\gcc-linaro-7.2.1-2017.11-i686-mingw32_arm-linux-gnueabi\bin

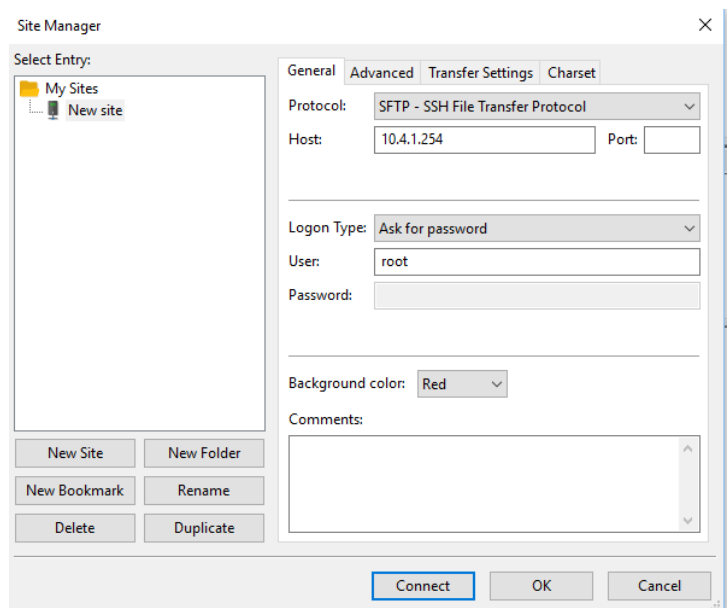
NOTE: Make sure there are no spaces in the name of the path!! So, save the file in the highest possible directory.

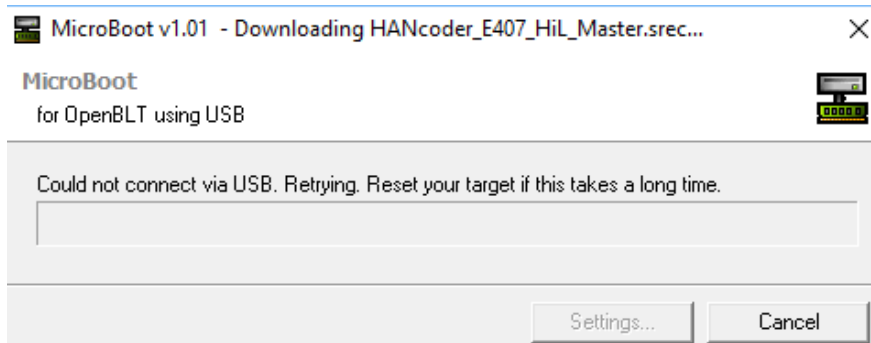
- d. Press the Windows Key and type 'System'. Click on the icon seen below.



- e. Click 'Advanced System Settings' -> Select tab 'Advanced' -> Click 'Environment Variables...'
- f. Select variable 'Path' and click 'Edit'. Click 'New' and paste the path of the Linaro bin folder.
- g. Click 'OK' -> 'OK' -> 'OK'
- h. The compiler path has now been added, but the computer must be reset for this to take effect. So, reset the computer and then continue.

4. The files to flash to Lely GCB and STM32
 - a. For this example, the files to flash to the separate components are:
 - i. Lely GCB: 'Linux_Autodocking_Case_HiL.elf'
 - ii. STM32 (I/O Module):
'HANcoder_E407_HiL_IOModule_Autodocking_Case.a2l'
 - iii. STM32 (ECU): 'HANcoder_E407_HiL_ECU_Autodocking_Case.a2l'
5. Build the Linux Model.
 - a. Open the 'Linux_Autodocking_Case_HiL.slx' file. Make sure the working MATLAB path is the same as the location of this simulation model.
 - b. Build the model by clicking the build icon or using 'ctrl + b'.
 - c. A model 'Linux_Autodocking_Case_HiL.elf', is created. This model will be transferred to the Linux GCB.
6. Open FileZilla to connect to the Lely GCB
 - a. Add a 'New Site' with the following details:
 - b. The host ID is the ID of the Lely GCB. Use 10.4.1.254 when using the ETH1 port or 10.4.1.253 when using the ETH0 port.
 - c. The default log in details for the Lely GCB are:
 - i. User: root
 - ii. Password: root
 - d. Click connect. The connection is successful if the right screen in FileZilla shows the details of the connected device.
 - e. Navigate to the location of the .elf model ('Linux_Autodocking_Case_HiL.elf') in the left browser and drag and drop the file to the right side (Lely).
 - f. Right click on the file when it has been transferred to the Lely and select 'Properties'. Check the box that says 'Execute' to give permission to execute the file. Click OK.
 - g. Now the file has been loaded to the Lely GCB and can be executed.
7. Next, flash the model to the STM32. Use MATLAB/Simulink with the 'HANcoder_E407_HiL_IOModule_Autodocking_Case.a2l' for this step.
 - a. In Simulink, navigate to the 'HANcoder_E407_HiL_IOModule_Autodocking_Case.slx' file. Make sure the MATLAB path is in the location of the Simulink file along with all necessary HANcoder files.
 - b. Check the IP-address in the 'HiL smart IO module' block. The HiL Linux IP must be the same as the IP host given in FileZilla. The further parameters can be left as default.
 - c. Build the model by clicking the build icon or using 'ctrl + b'. When the MicroBoot window appears in Simulink (below), reset the target by pressing the reset button. Then the model will be loaded to the STM32.





8. Repeat step 7 for the model 'HANcoder_E407_HiL_ECU_Autodocking_Case.slx'.
9. Now the models have been loaded and are ready for execution. Before execution, HANTune will be set up to follow the progress of the simulation.
 - a. Open HANTune.
 - b. Open the project that has been made available. All dials and dashboard elements will appear on screen.
 - c. Connect to the STM32 by XCP on USB/UART connection. For a tutorial on how to, see http://www.openmbd.com/wiki/HANTune/Communication#Connection_Dialog.
10. Now the models can be executed.
 - a. Open Putty and connect to the Lely GCB by entering the IP address used before.
 - b. Login in the Putty command prompt using the same details as used in Filezilla.
 - c. Enter the command to execute the .elf file on the Lely:
 - i. `./*path to file*/*filename*.elf`
 - ii. *Important, this execution line must always contain the './' to execute.*
 - d. Putty will return the information as seen below.

```

login as: root
root@10.4.1.254's password:
Linux GCB 3.10.33-R09-rt33 #1 SMP PREEMPT RT Fri May 2 17:30:16 CEST 2014 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan  1 01:01:51 1970 from 10.4.1.9
root@GCB:~# ./HANcoder_Linux_GCB_Hil_Willard.elf
  
```

- e. The simulation is running. Check HANTune to see what the results are. *Note: There is some delay between executing the model in Putty and seeing the 'signalOutput' results in HANTune.*
11. Stopping the simulation.

- a. To stop the simulation, a hard kill is executed. Press the ‘Pause/Break’ key (on laptops Fn + b) on the keyboard while in the Putty command prompt to kill the simulation.
- b. This command does not close the ports used by the simulation, so that must be done manually.
- c. Use the command: ‘netstat -ap | grep :5’. This shows if port 5 (which is the port selected in the STM32 model ‘HiL smart IO model’ block. This is seen below. The ports with HANcoder_Linux (in this case) are the ports which need to be shut down. For the example this is 2050.

```

10.4.1.254 - PuTTY
root@GCB:~# netstat -ap | grep :5
getnameinfo failed
tcp        0      0  *:57832             *:.*                LISTEN     1674/rpc.statd
tcp        0      0  *:50002             *:.*                LISTEN     2050/HANcoder_Linux
tcp6       0      0  [::]:51167         [::]:.*            LISTEN     1674/rpc.statd
udp        164079  0      *:5                 *:.*                2050/HANcoder_Linux
udp        0      0  *:54815            *:.*                1674/rpc.statd
udp6       0      0  [::]:50623        [::]:.*            1674/rpc.statd
root@GCB:~#

```

- d. Use the command: ‘kill -9 xxxx’. In which the ‘xxxx’ represents the process ID (kill -9 2050’. This command closes the port and the HiL Simulator is ready for another simulation.
- e. These steps must be done after each simulation is stopped. If not, the ports are occupied by the old simulation while the new simulation tries to connect.
- f. The STM32 boards must be reset (using the reset button) after the Linux Target process has been killed. If this is not done, the next simulation will fail due to no connection between the Lely GCB and the STM32 boards.

Note: Before logging the results in HANtune, wait for lag indicator to indicate no/little lag.