# HANcoder Target STM32-E407 Getting Started

Version 0.4
8/26/2016

## Getting Started Guide

A guide explaining all necessary steps to be able to use the code generation blockset, in MATLAB, and to use HANtune for the STM32-E407.

# HANcoder Target STM32-E407 Getting Started

GETTING STARTED GUIDE

## Contents

VERSIONING ..... 3

INTRODUCTION ..... 4

1. MATLAB INSTALLATION ..... 5
   1.1. Requirements ..... 5
   1.2. Install ..... 5
   1.3. Toolboxes to install ..... 6
   1.4. MATLAB as administrator ..... 8

2. GNU ARM TOOLCHAIN INSTALLATION ..... 9
   2.1. Installation ..... 9

3. TEST THE BUILD PROCESS ..... 10
   3.1. Explanation of the template model ..... 10
   The inputs subsystem ..... 12
   The algorithm subsystem ..... 12
   The outputs subsystem ..... 13
   The configuration blocks ..... 13
   The System Information Subsystem ..... 13
   3.2. Build your own model ..... 14
   3.3. Working with m-scripts ..... 16
   3.4. Code generation ..... 18

4. FLASHING THE CONTROLLER WITH THE BOOTLOADER ..... 19
   4.1. Flashing the controller via USB ..... 19
   4.1.1. Install bootloader driver ..... 19
   4.2. Flashing the controller via TCP/IP ..... 22
   4.3. Flashing the controller via CAN ..... 24

5. FLASHING THE CONTROLLER WITH THE ST-LINK ..... 26
   5.1. Software program reconfiguration ..... 26
   5.2. Uploading the software with the ST-Link ..... 26
   5.3. Flashing the bootloader ..... 28

6 HANTUNE ..... 29
   6.1 Installing the virtual COM port driver ..... 29
   6.2 Using HANtune ..... 29

APPENDIX 1. ST-LINK INSTALLATION ..... 32
APPENDIX 2. SEGGER J-LINK INSTALLATION ..... 34
Installation ..... 34
APPENDIX 3. FLASHING THE BOOTLOADER WITH THE J-LINK ..... 36
APPENDIX 4. CONFIGURING PC TO FLASH THE E407 OVER ETHERNET ..... 39
APPENDIX 5. PEAK PCAN INSTALLATION ..... 41
APPENDIX 6. REBUILDING THE BOOTLOADER ..... 42

# VERSIONING

| Nr | Date | Person | Changes | Status |
|---|---|---|---|---|
| 0.1 | 18-03-2014 | F. Voorburg | First version | Concept |
| 0.2 | 02-12-2014 | F. Voorburg | Updated for USB bootloader | Concept |
| 0.3 | 18-09-2015 | J. van Kolfschoten | Moved some parts to the appendices and made the manual more readable | Ready for review |
| 0.33 | 26-08-2016 | F. Voorburg | Updated USB bootloader description. | Ready for review |

# INTRODUCTION

This document is for the user who wants to make a software program using Simulink and the HANcoder blockset for the Olimex STM32 E407 development board. It contains installation tutorials for the programs necessary to work with the blockset. It will also explain the template model, building a model in Simulink and flashing the software to the E407.

Furthermore, it will explain how you can view signals and edit parameters while the software that runs on the E407. This is done with a program called HANtune.
So HANcoder is used to build the software program from a Simulink model and HANtune is used to view and edit (tune) the program once the program is running on the hardware.

If you encounter problems with the blockset or with HANtune, please contact the project developer Jason van Kolfschoten: Jason.vanKolfschoten@han.nl

# 1. MATLAB INSTALLATION

First of all, MATLAB has to be installed. If this is already done, please check if the required toolboxes are installed.

## 1.1. Requirements

MATLAB can be installed with Windows XP, Windows 7, Windows 8 and Windows 10. Somehow, using MATLAB with Windows Vista seems to cause problems with the code generation.

The following MATLAB versions are compatible with the HANcoder STM32 E407 Target:

MATLAB 32-bit, 2009a until 2014a

MATLAB 64-bit, 2012a until 2016b

For Matlab versions from 2009b until 2013a a special set of template models is available.

## 1.2. Install

Matlab can be downloaded from the MathWorks site. As can seen in the figure below, you can find the download on the **products** page. You'll need a Mathworks account to download products of Mathworks.

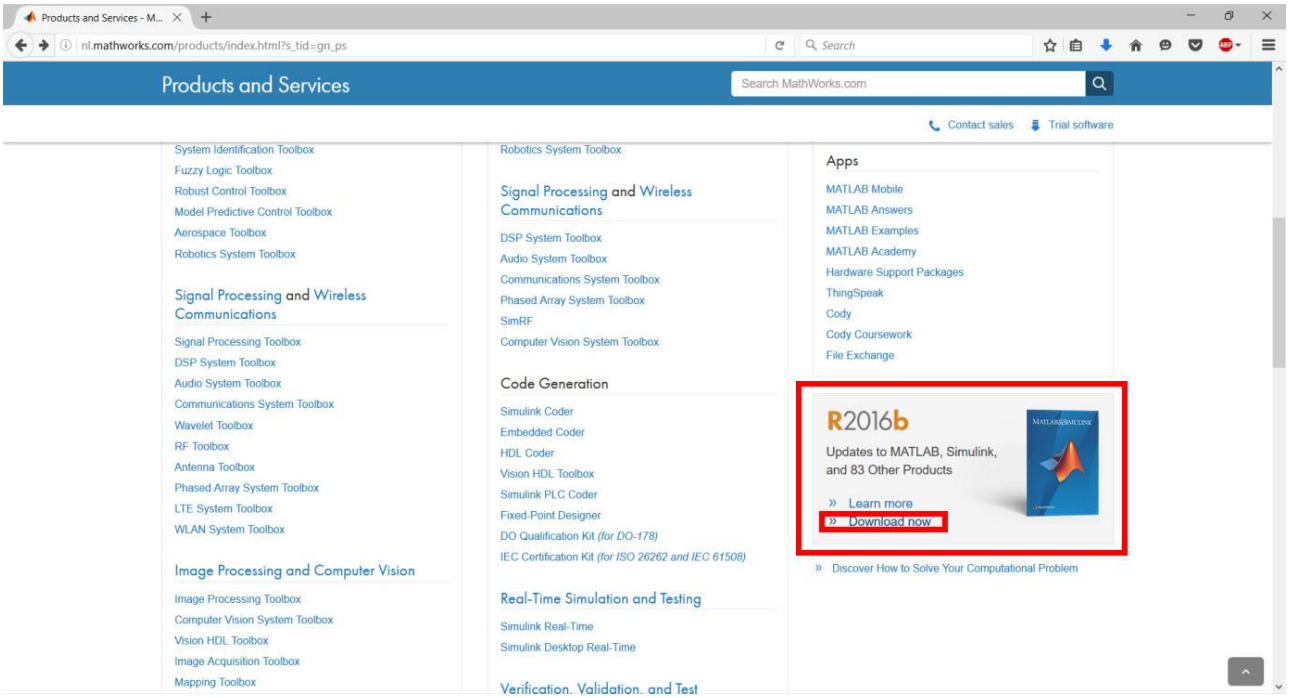



**Figure 1-1 Download section MathWorks website**

The installation steps for MATLAB are documented on the MathWorks website. Please follow the steps as indicated on the website, the steps for older MATLAB versions are similar. Be aware of the **additional installing info**, located below, that is important for the use of HANcoder.

## 1.3. Toolboxes to install

While installing Matlab you're being asked to select the products to install. As previous said the required toolboxes are:

- MATLAB
- Simulink
- Simulink Coder (formerly Real-time Workshop)
- Embedded Coder (formerly Real-time Workshop Embedded Coder)

The following toolboxes are optional, but highly recommended

- MATLAB Coder
- Stateflow
- Stateflow Coder (nowadays integrated in Simulink Coder)
- Fixed-point Designer (formerly Fixed-Point Toolbox & Simulink Fixed Point)
- Vehicle Network Toolbox

The tools can be selected in the window that is shown on the next page:

**Figure 1-2 Tool selection window installing MATLAB R2015a**

## 1.4. MATLAB as administrator

To give Matlab the permission to remove and overwrite files, it has to be run as administrator. It can be set as default by following these steps:

Step 1: Go to the location where the shortcut to Matlab is located.

Start → search for Matlab R2015a → right click: **Open file location**

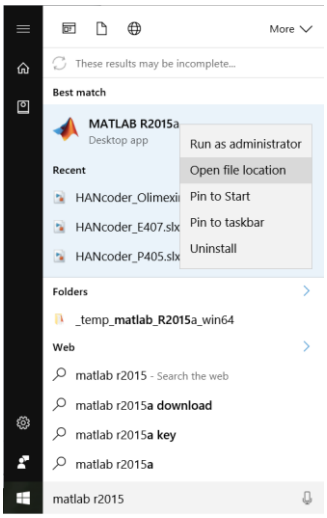



**Figure 1-3 Windows 10 Start menu, search section**

Step 2: Right click on the MATLAB R2015a shortcut and select properties.

Go to the Compatibility tab and select **Run this program as an administrator**. Click **apply**.

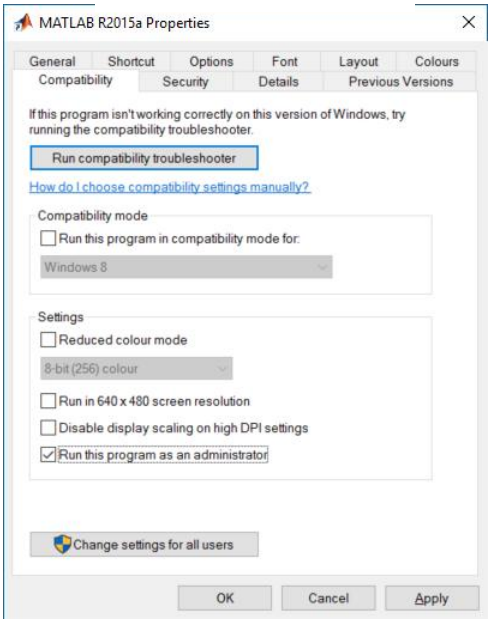



**Figure 1-4 MATLAB R2015a Properties, Compatibility tab**

# 2. GNU ARM TOOLCHAIN INSTALLATION

The E407 STM32 uses a Cortex-M microprocessor from ST. To compile for this target, the “GNU ARM Embedded Toolchain” has to be installed. It can be found on this website:

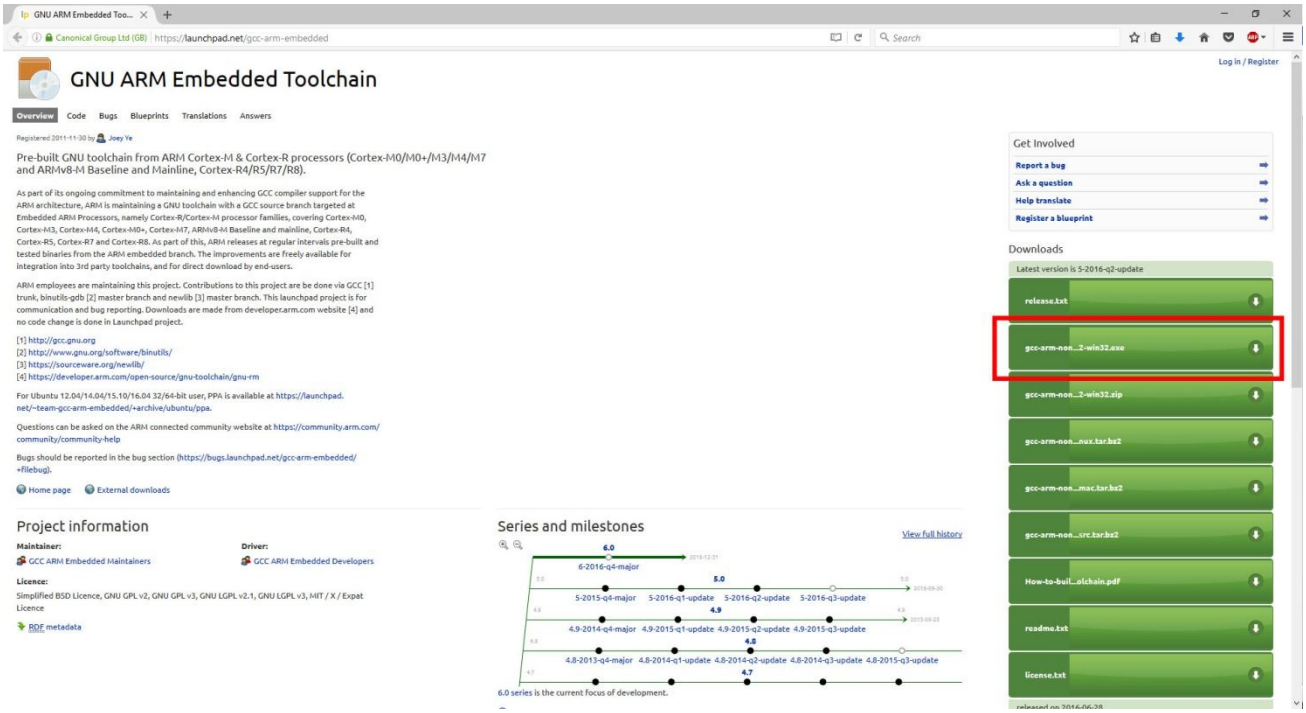



**Figure 2-1 Frontpage Launchpad.net website link**

Choose the following file: ‘gcc-arm-none-eabi-5_4-2016q2-20160622-win32.exe’

## 2.1. Installation

Start the installation by double-clicking the .exe file and follow the installation steps until you arrive to the following screen.

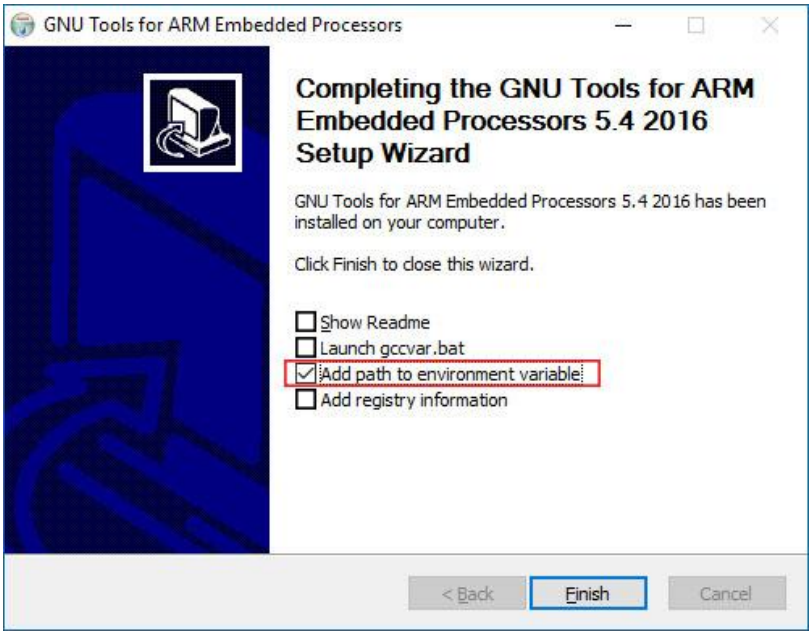


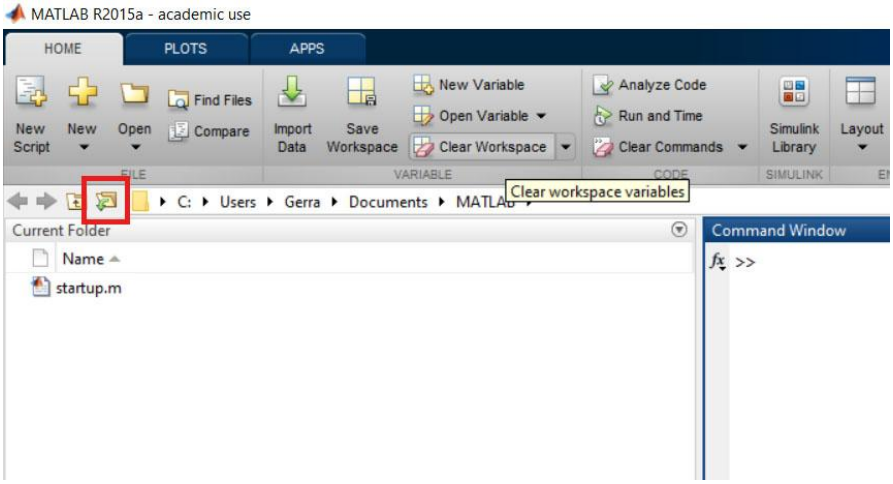

**Figure 2-2 Setup wizard properties**

NOTE!: Make sure that **add path to environment variable** is turned on after the installation.

# 3. TEST THE BUILD PROCESS

## 3.1. Explanation of the template model

Open the Simulink model by starting MATLAB and navigating to the HANcoder directory where the Simulink models are located, the 'Target' directory. By double-clicking the provided model file, i.e. 'HANcoder_E407.slx', from within MATLAB the model is started.




The right directory can easily be found with the **browse for folder** button.

**Figure 3-1 Browse for folder button Matlab**

Make sure that the **Target folder** is selected. Otherwise Matlab will have trouble finding the necessary files to build HANcoder models.

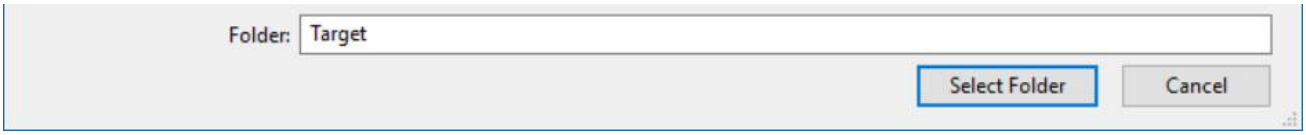



**Figure 3-2 select target folder**

NOTE: MATLAB versions prior to the 2013b release cannot use HANcoder's .slx files and instead the .mdl files for these earlier versions can be found in the zip file: **Template models for older versions.zip**.

If the default project is used, the following screen will appear:

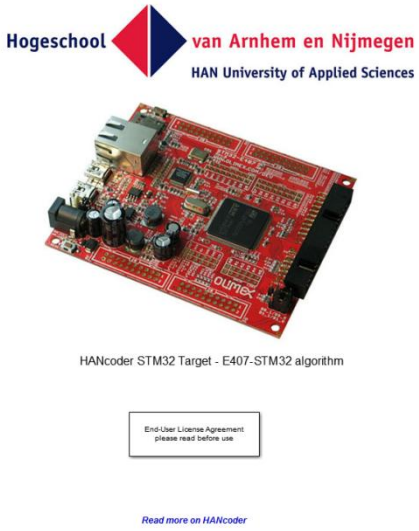



**Figure 3-3 View when opening the model**

Nothing in this part of the model can be changed, otherwise the code generation will no longer work!

When double-clicked on the picture of the controller, the content of this block is shown:

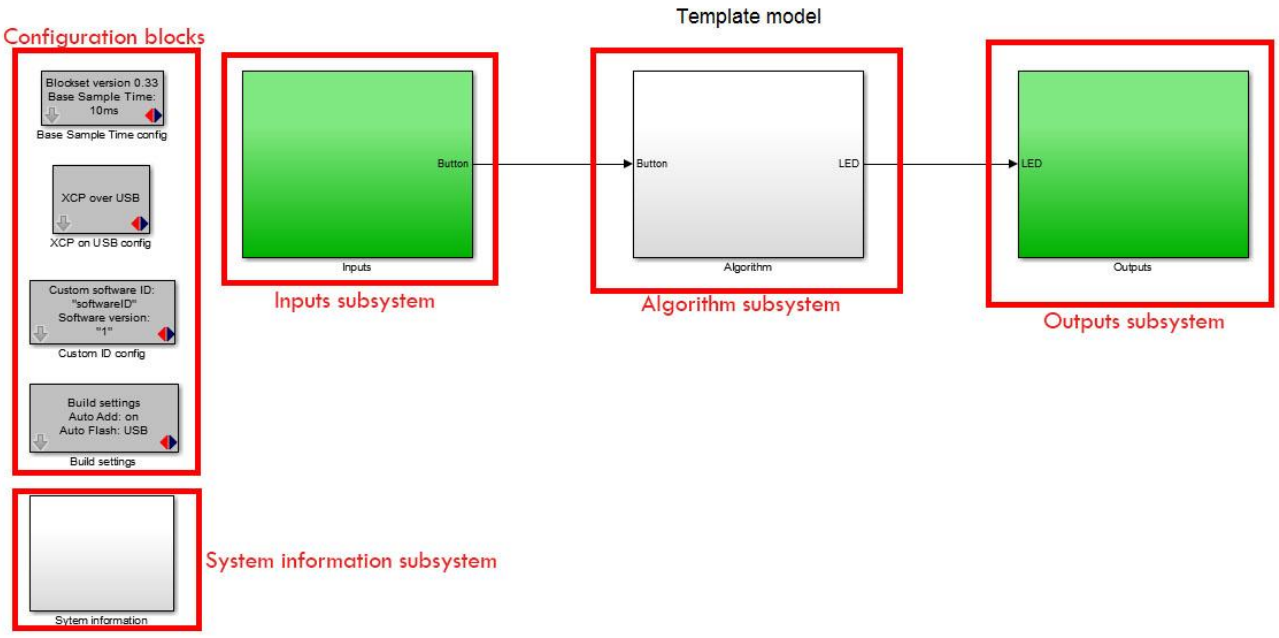



**Figure 3-4 Basic model delivered with blockset**

**Figure 3-4** shows the template model. This is the base of every new project. This simple project makes the green LED blink at 5Hz when the button isn't pressed and at 10Hz when it is pressed. It is advised to keep this functionality in your project so you can always check if the software is still responsive.

The template model consists of:

- The inputs subsystem
- The algorithm subsystem
- The outputs subsystem
- The configuration blocks
- The system Information subsystem

## The inputs subsystem

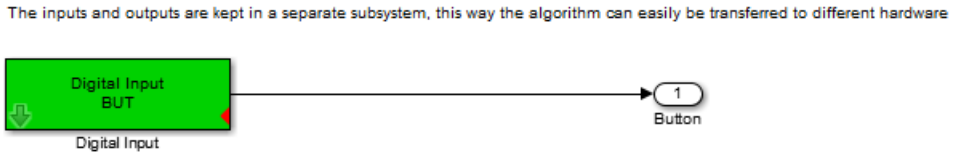



**Figure 3-5 insite the Inputs subsystem**

The inputs of the model are placed inside the Inputs subsystem. The inputs are connected with the Algorithm through Outport blocks (the block with 'Button' below it in the figure above).

## The algorithm subsystem

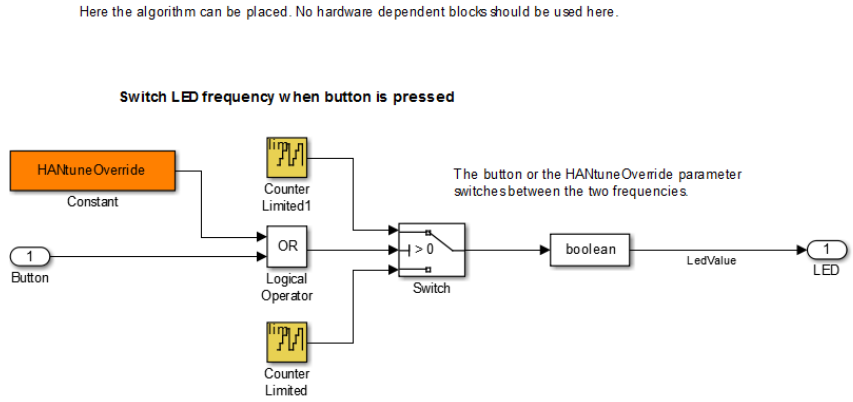



**Figure 3-6 The algorithm subsystem template model**

In the algorithm subsystem the functionality is placed. The subsystem is connected with the inputs and outputs through In- and Outport blocks. The functionality can easily be transferred to another hardware platform because there are no in- or outputs in this part, it is recommended to work the same way in your own projects.

## The outputs subsystem

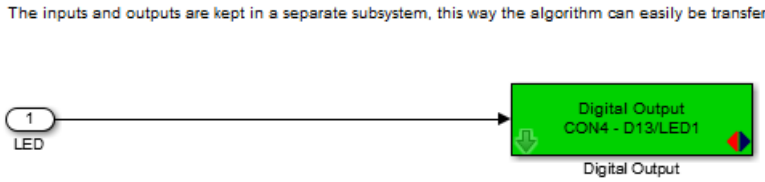



**Figure 3-7 The outputs subsystem template model**

In here the outputs of your model are located.

## The configuration blocks

These blocks are the settings of the system.
The Base Sample Time determines the interval at which the model is run. You can add blocks to the model which run at lower rates but not faster. In the template model the base sample time is 10ms so the model can run at a 100Hz.

The XCP on USB config block configures the E407 to communicate to HANtune over USB. (More on HANtune in chapter 6)

The custom software ID block lets you choose a name and version number for your model, this will be used when connecting with HANtune.

With the Build settings you can let HANcoder automatically add all signals and parameters in the workspace of the project. This is important to communicate with HANtune.
The Auto Flash function flashes the software automatically after a successful build. explains more about flashing the program.

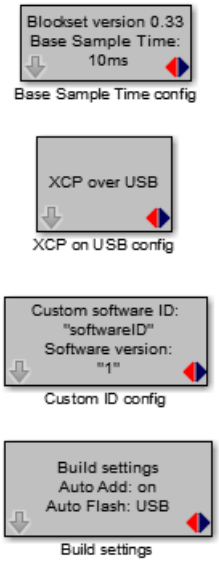



**Figure 3-8 Configuration blocks template model**

## The System Information Subsystem

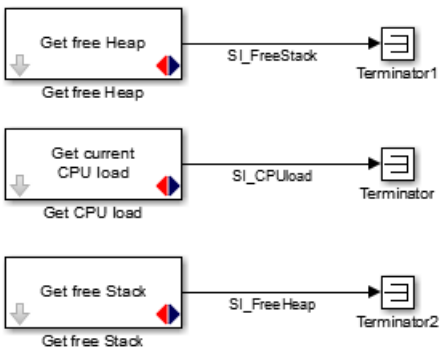



**Figure 3-9 The System Information subsystem template model**

The System Information Subsystem gives you information about the Load, Heap and Stack of the system once the model is running. This way you can monitor how many resources your software program uses. The signals are already defined by HANcoder and will be visible in HANtune.

## 3.2. Build your own model

To build your own model, you have to open the Simulink Library browser from within MATLAB:

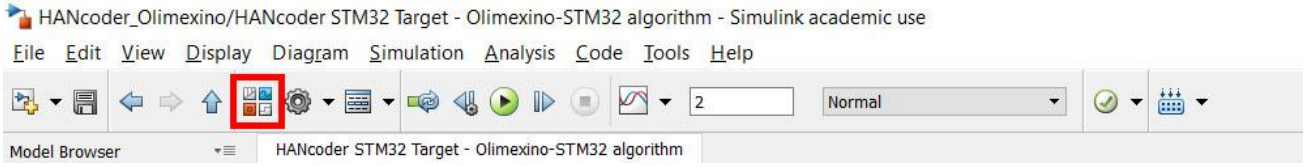



**Figure 3-10 Simulink Library button**

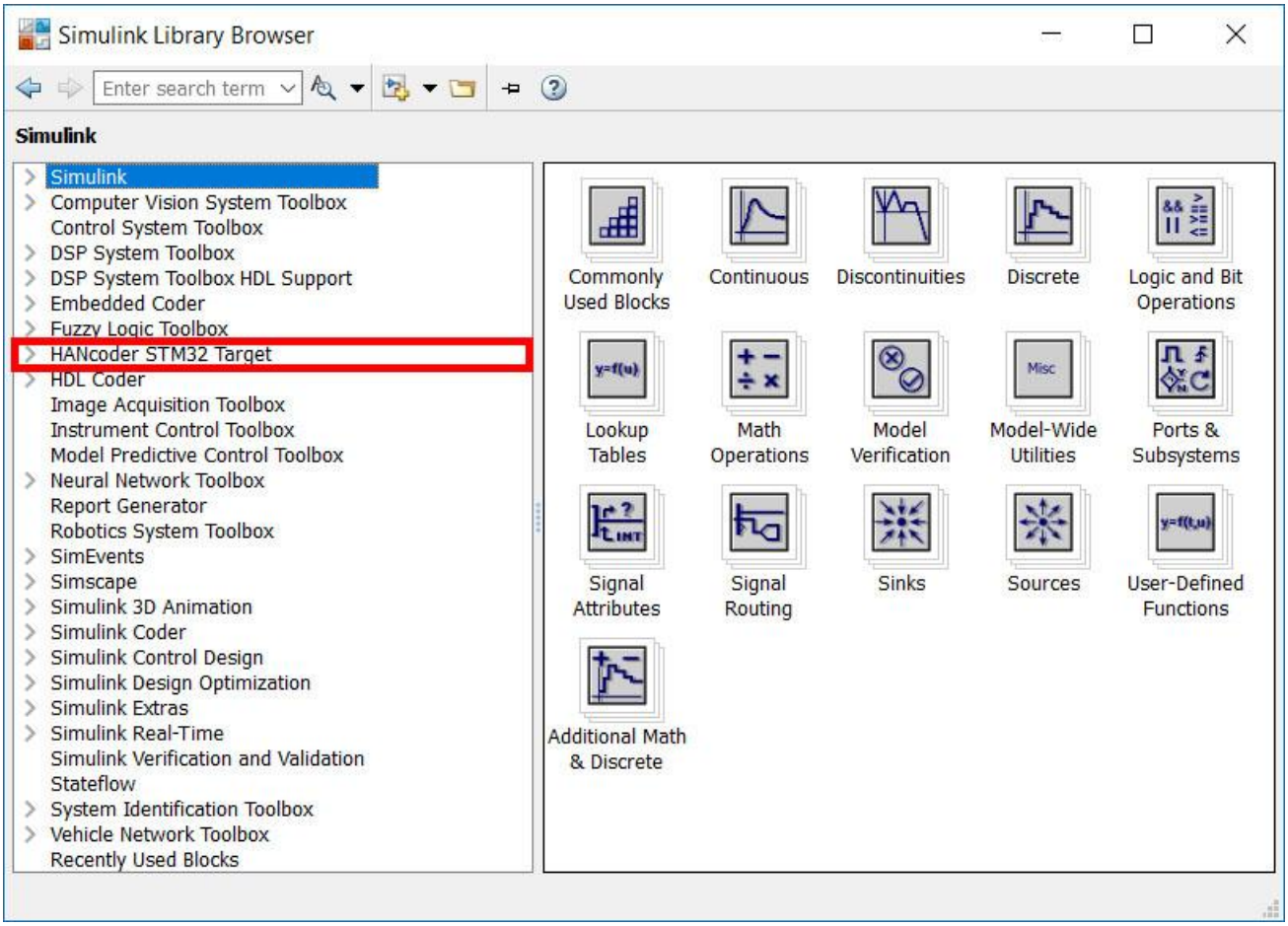



**Figure 3-11 Simulink library**

Open the 'HANcoder STM32 Target' toolbox by clicking on it.

If the library isn't visible try refreshing the library tree view by pressing F5. If there is a Pop up bar that says 'Some libraries are missing repository information. Fix' Click on Fix and select 'Generate repositories in memory' in the pop up.

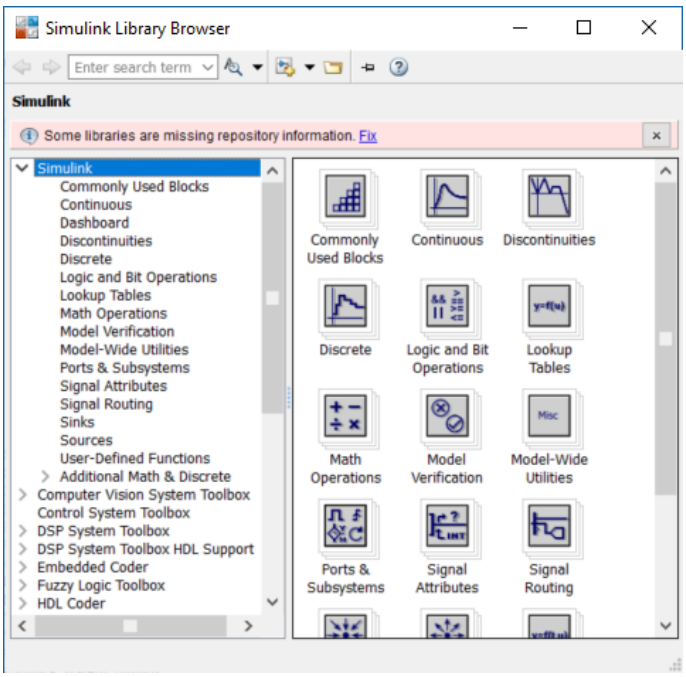



Figure 3-13 Library pop up

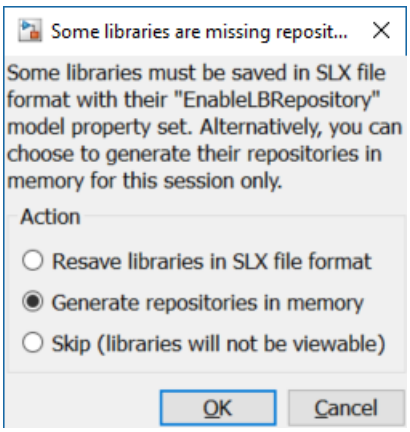



Figure 3-12 library Fix pop up

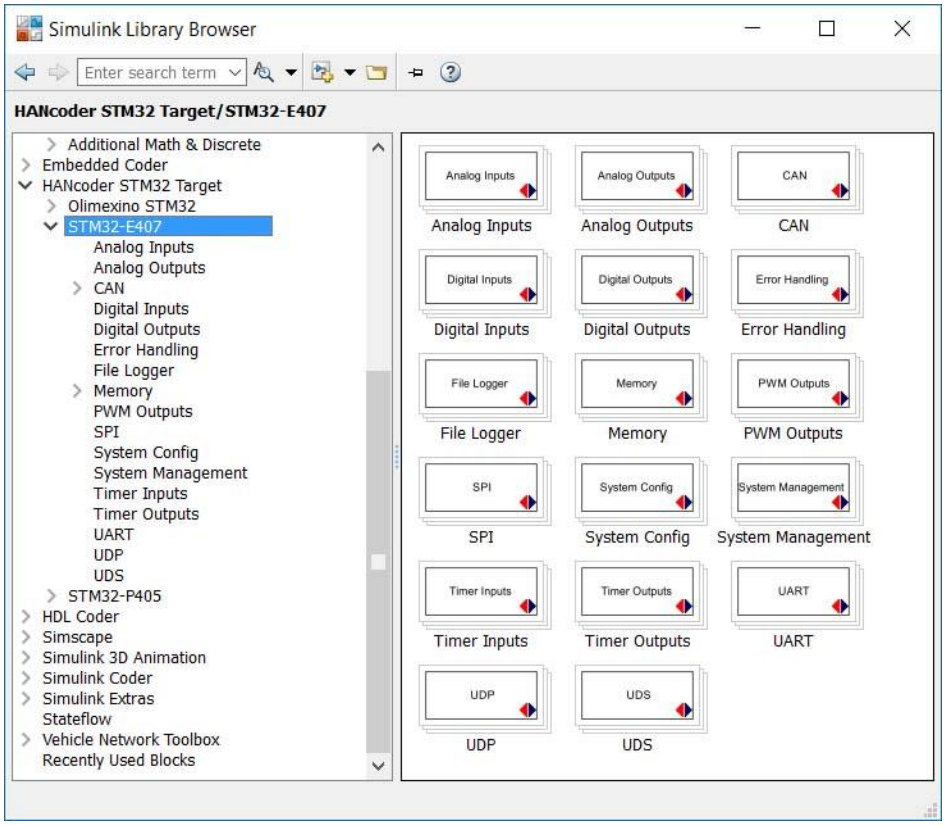



Figure 3-14 HANcoder library STM-E407

To the right, the content of the toolbox is placed. This content is sorted in a tree like view for easy access. Use the help files to determine the proper use of a block. These are accessible by right clicking on a block and then selecting “Help”.

## 3.3. Working with m-scripts

The best option to define the parameters is by using m-scripts. An m-script is useful for a series of Matlab commands. In HANcoder an m-script is used for defining both signals and parameters. The template model automatically calls the **startup** m-script when it opens. This is done by adding it to the model properties: callbacks. Go to **Simulink model→ File → Model properties → Callbacks.** The Model pre-load function runs every command that is written in the section before the model starts.

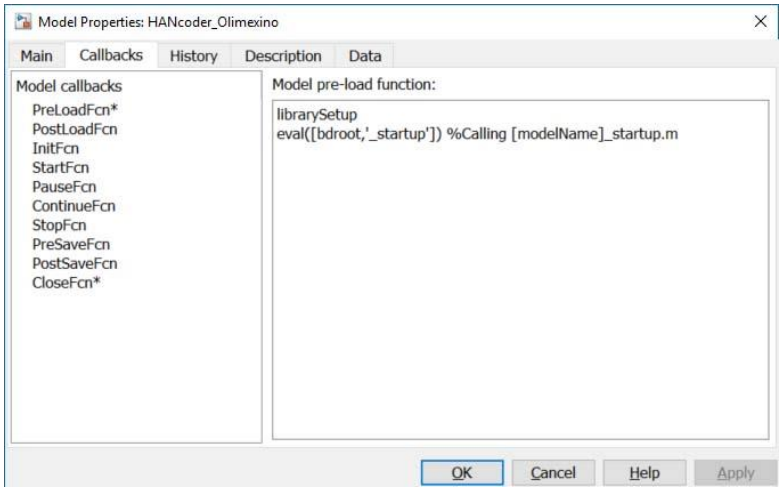



**Figure 3-15 Model Properties - Callbacks**

As can be seen in Figure 3-15 Model Properties - Callbacks, the call for the startup file is made with the following command:

eval([bdroot,'_startup']) (This calls the HANcoder_E407_startup.m script)

It is advisable to define all parameters in the startup script. If this is done, the parameters will be added to the workspace every time the model is opened. When a parameter is added in the m-script, the m-script can be run manually by clicking run. The parameter will be added to the workspace.

The HANcoder_E407_startup.m is shown below:

```
% This m-file will be run automatically when starting model 'HANcoder_E407'
% The name of the m-file must consist of the model name with the addition:
% '_startup'!
% This m-file will be run when loading the model because it is added to the
% model callbacks: see File->Model Properties->Model Properties->Callbacks

% In this m-file the signals and parameters for HANtune can be defined. As
% an example the signal LedValue and parameter HANtuneOverride are already
% defined.
% You can add your own signals and parameters to this m-file below

%% Signals
% Defining signals for viewing in HANtune
LedValue = Simulink.Signal; % Define as signal
LedValue.StorageClass = 'ExportedGlobal'; % Only Exported Global will be visible
in HANtune

% Defining System Information Signals
SI_FreeStack = Simulink.Signal;
SI_FreeStack.StorageClass = 'ExportedGlobal';
SI_CPUload = Simulink.Signal;
SI_CPUload.StorageClass = 'ExportedGlobal';
```

```
SI_FreeHeap = Simulink.Signal;
SI_FreeHeap.StorageClass = 'ExportedGlobal';

%% Parameters
% Defining a parameter for editing in HANtune
HANtuneOverride = Simulink.Parameter; % Define as parameter
HANtuneOverride.StorageClass = 'ExportedGlobal'; % Only Exported Global will be
visible in HANtune
HANtuneOverride.Value = 0; % Initial value is set to zero, no override
```

Signals and parameters can be added in this file. If the storage class is set to 'ExportedGlobal' the value is set as a global variable. In this case the signal or parameter can be seen by HANtune.

## 3.4. Code generation

To generate C code from the Simulink model use CLTR-B on the keyboard when the model is opened. The model will 'grey' out during the build procedure.

In MATLAB versions earlier than 2014a the progress can be observed in the MATLAB command window. From 2014a this can only be viewed by clicking View diagnostics on the bottom of the model.

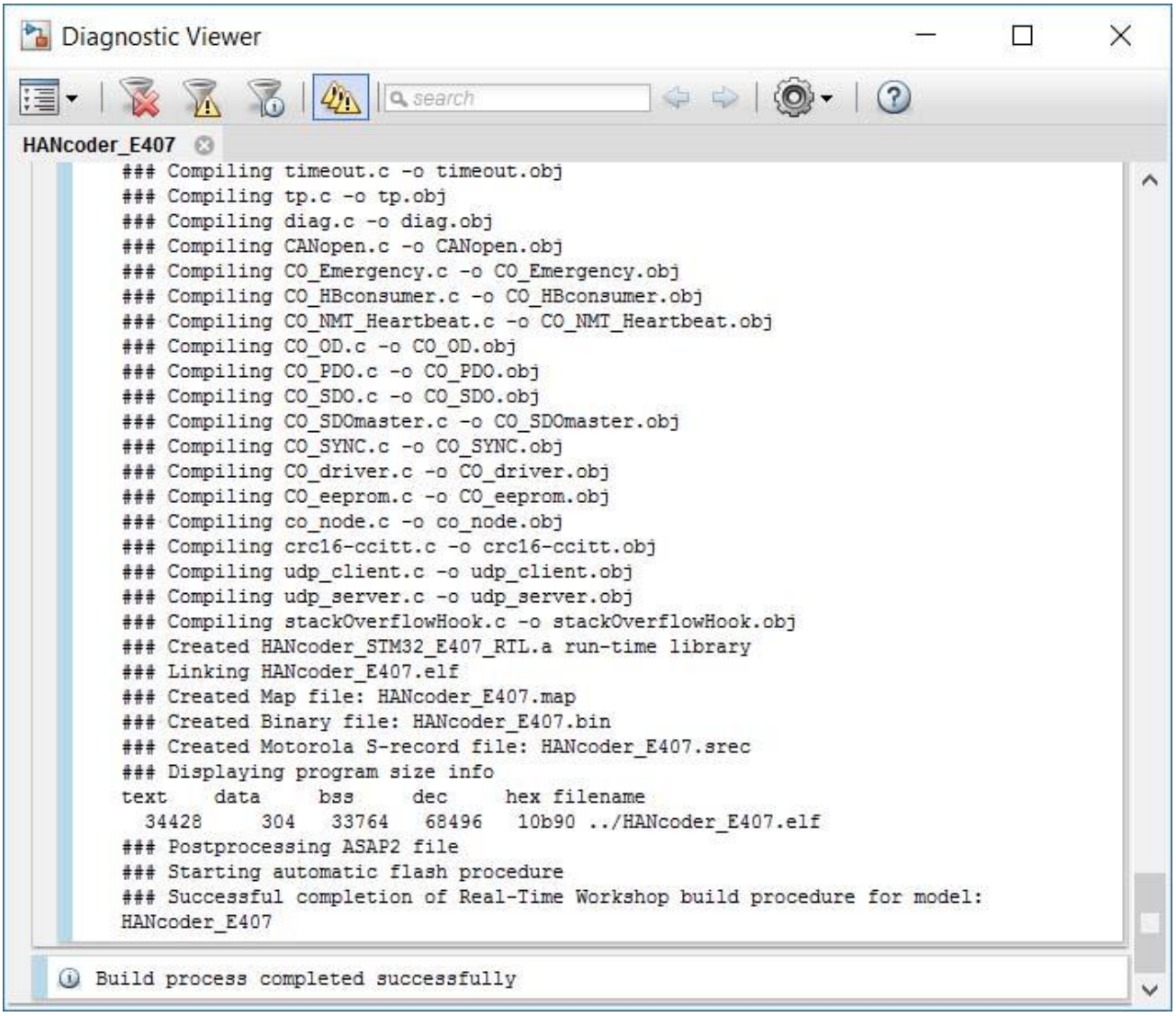



**Figure 3-16 Code generation and build progress**

Once the model indicates ready in the lower left corner, the software program is built and flashed (if Auto Flash is on). In the target directory two new files appear.

Please consult The Troubleshooting Guide if there are any errors.

# 4. FLASHING THE CONTROLLER WITH THE BOOTLOADER

This chapter describes the steps to be taken to flash the controller by CAN, Ethernet or USB. This is only possible if you possess an E407 with the correct bootloader. If you wish to flash the bootloader or just flash the software directly onto the E407 please follow the steps in chapter 5 Flashing the controller with the ST-Link

## 4.1. Flashing the controller via USB

It is possible to flash the E407 via the CANbus or via an USB connection. This paragraph describes how to flash the E407 using USB.

Downloading the software program to the flash memory of the microcontroller can be done with the MicroBoot utlity. Before continuing, make sure that:

- The board is connected to the PC's USB port.
- Power is supplied to the E407 board, change jumper to OTG1 to power the board with USB
- The E407 has the correct bootloader on it

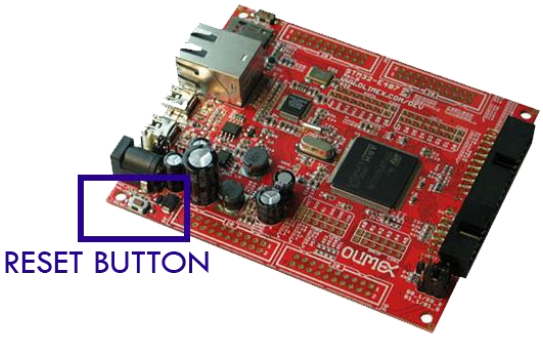



**Figure 4-1 RESET button E407**

### 4.1.1. Install bootloader driver

To install the driver the program should be kept in bootloader mode. This is achieved by keeping the WKUP button pressed while resetting the controller using the RESET-button. The first time the USB is activated on a different PC, Window's will ask you to install the USB driver. The driver can be found here.

Quoting OpenBLT:

Before installing the USB driver, make sure the USB device that runs the OpenBLT USB bootloader is not connected to your PC. If a prior version of the USB driver was installed, it is recommended to first uninstall this driver.

The installation of the USB driver is performed using the Zadig tool. You can download the necessary files here: OpenBLT USB Driver Package. After downloading the OpenBLT USB Driver Package, unzip the archive to a location of your liking and start the program **zadig_2.2.exe**.

From the program menu, select **Device** → **Load Preset Device** and select the file **openblt.cfg**, which is found in the same directory as where **zadig_2.2.exe** is located. Next, click the **Install Driver** button to install the USB driver for the OpenBLT bootloader:

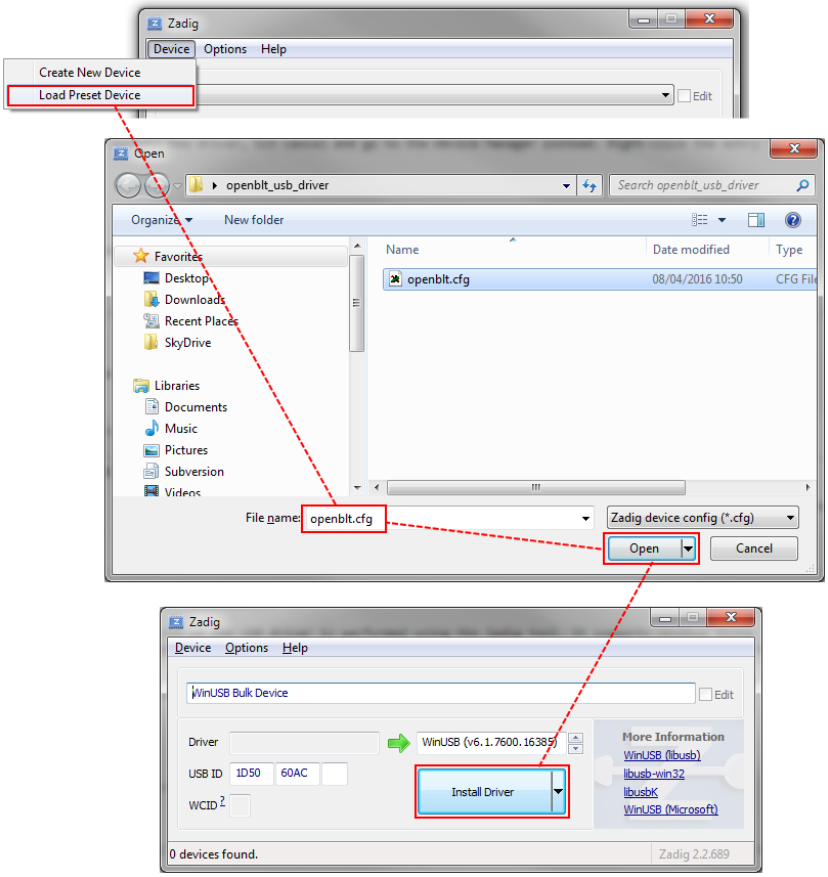



Figure 4-2 Feaser Open BLT explanation figure 1

After successfully completion of the USB driver installation, you can plug in the USB device that runs the OpenBLT USB bootloader. Windows will automatically detect the USB device and map it to the newly installed USB driver. You can verify the correct USB driver installation and USB device detection using the Device Manager in Windows. You should see an entry for **WinUSB Bulk Device**, without a yellow exclamation mark in the icon:

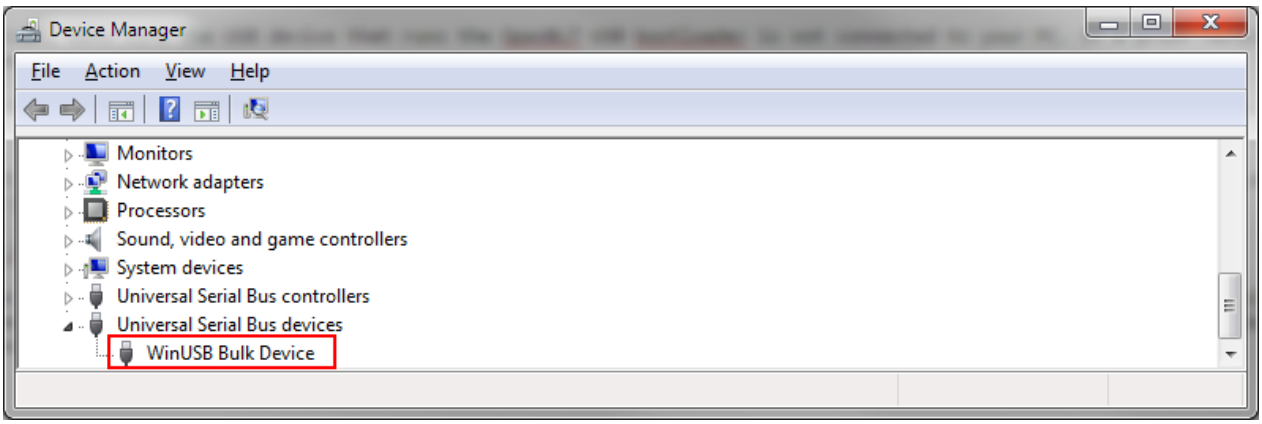



Figure 4-3 Feaser Open BLT explanation figure 2

To verify that the bootloader is active, look for the entry 'WinUSB Bulk Device' in Window's device manager:

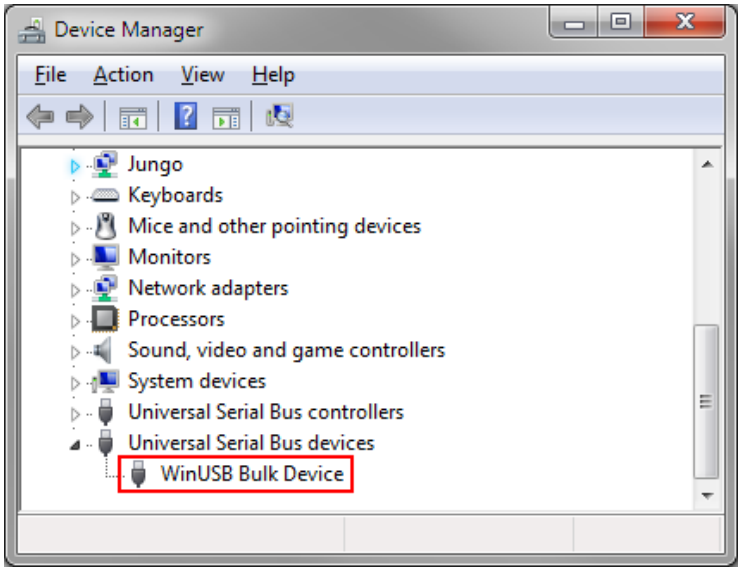



**Figure 4-4 USB bootloader in Device Manager**

Start the MicroBoot utility. It can be found in '\Host\MicroBoot\MicroBoot.exe'. Click the *Settings*-button and select 'OpenBLT using USB', and click the *OK*-button to save the settings.

Next, click the *Browse*-button to select the .srec file (Motorola S-Record file) of the software program. The one from the demo model is located at '\Target\HANcoder_E407.srec'.

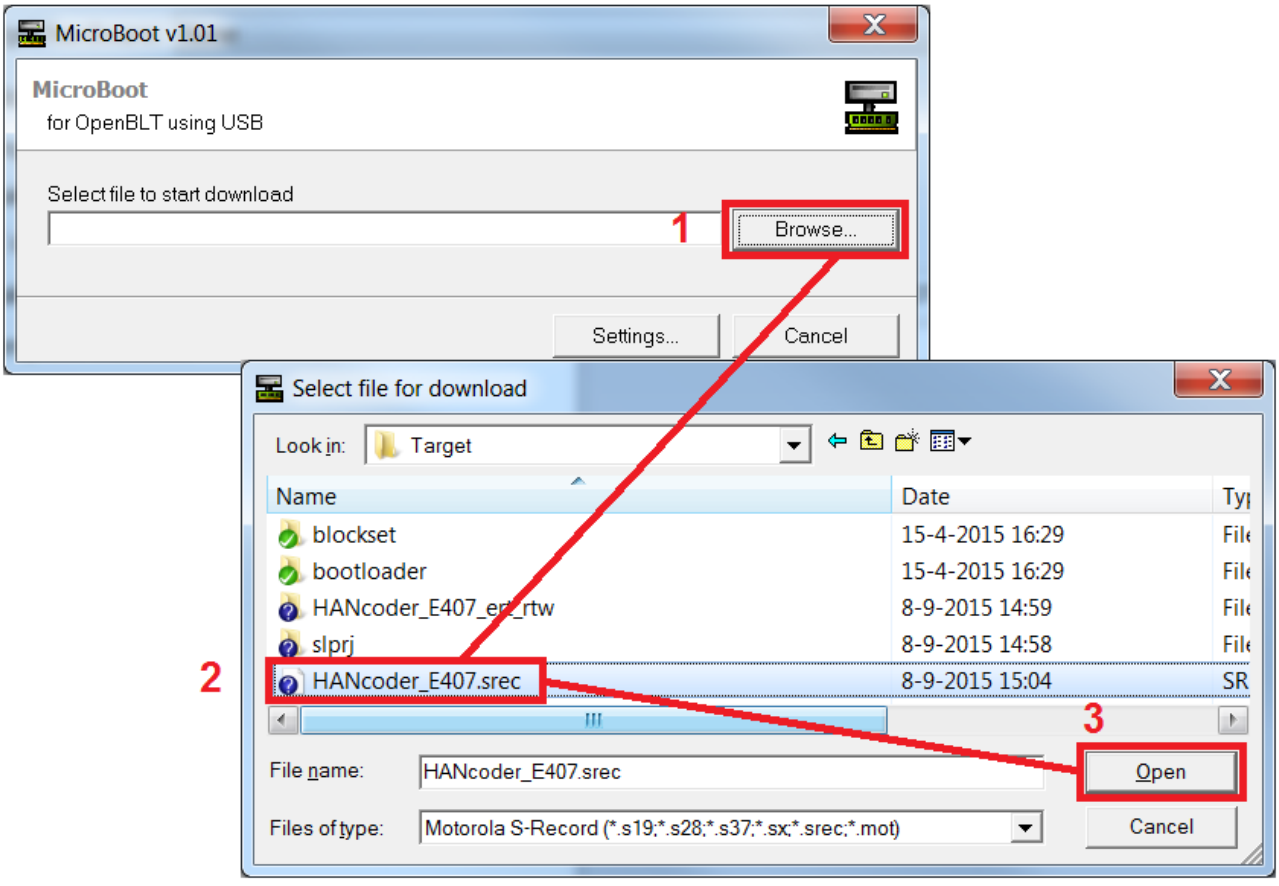



**Figure 4-5 Select Motorola S-Record in MicroBoot**

Once selected, the firmware update with the bootloader is started. Its progress is shown in the main screen. Upon completion, the newly flashed software program is automatically started.

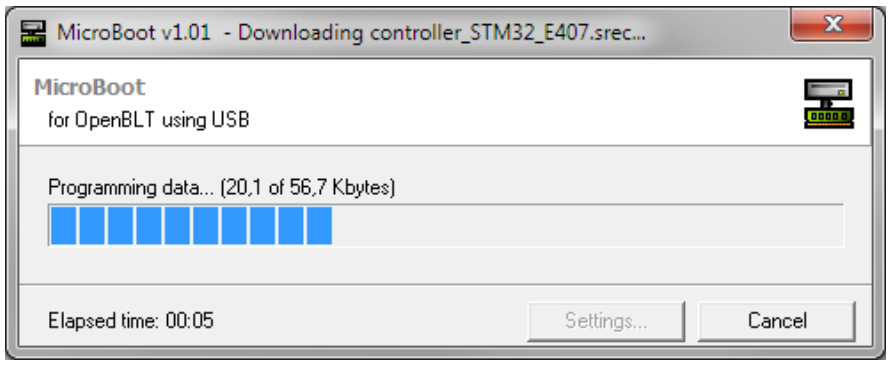



**Figure 4-6 Bootloader progress**

## 4.2. Flashing the controller via TCP/IP

Downloading the software program to the flash memory of the microcontroller can be done with the MicroBoot utlity. Before continuing, make sure that:

- The board is connected to the PC's Ethernet port with an Ethernet cable
- Power is supplied to the E407 board
- The PC's/router's Ethernet settings are correct, see Appendix 4 Configuring PC to flash the E407 over Ethernet
- The E407 has the correct bootloader on it

To double-check that the connection is okay, run the command 'ping 169.254.19.63' from the command prompt(run as admin) and make sure you get a response:

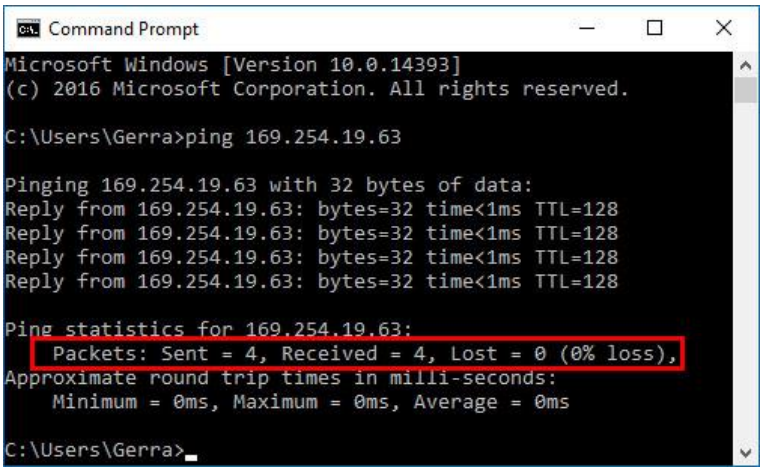



**Figure 4-1 – Checking ping response**

Start the MicroBoot utility. It can be found in '\Host\MicroBoot\MicroBoot.exe'. Click the *Settings*-button and select 'OpenBLT using TCP/IP' if this one was not already active, and click the *OK*-button to save the settings.

Next, click the *Browse*-button to select the Motorola S-Record of your software program. The one from the demo model is located at '\Target\HANcoder_E407.srec'.

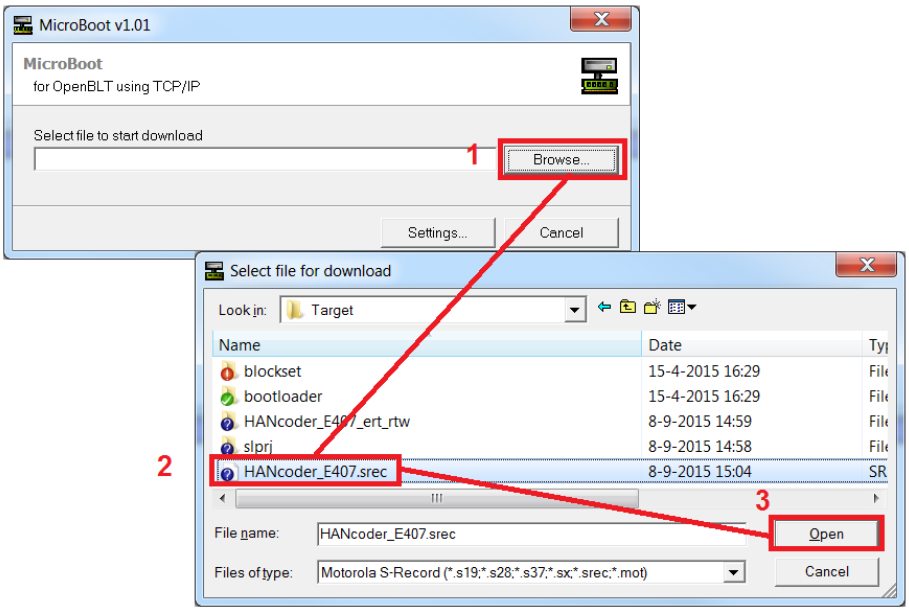



**Figure 4-2 – Select Motorola S-Record in Microboot**

The firmware update with the bootloader is now started. If a user program is currently running, Microboot will not get a response from the target. In this case Microboot will display the following info:

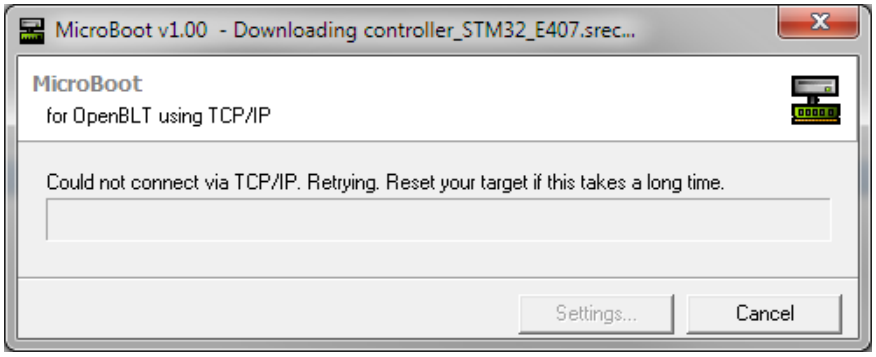



**figure 4-3 - Request for reset**

In this case press and release the reset button on the board. This will activate the bootloader. After a few seconds the actual firmware update will start. Its progress is shown in the main screen. Once completed, your newly flashed software program is automatically started.

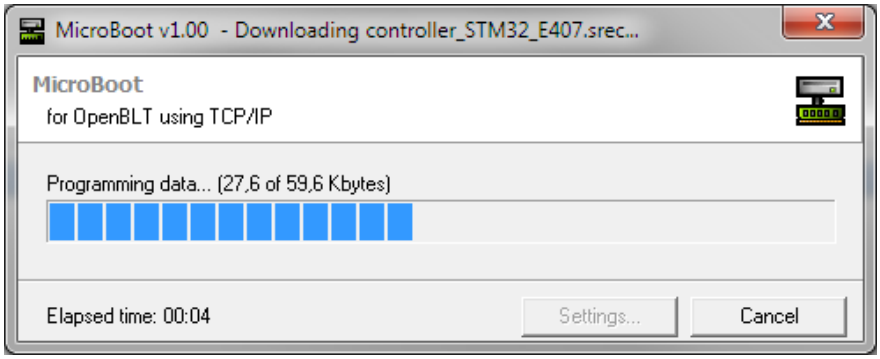



**Figure 4-4 – Bootloader progress**

Please note that there is a delay of several seconds before the newly programmed application is started. This is because after reset, the bootloader must stay active long enough for the TCP/IP stack to initialize and establish a link so it can receive a firmware update request from Microboot.

## 4.3. Flashing the controller via CAN

This paragraph describes how to flash the E407 using the CAN bus.

Downloading the software program to the flash memory of the microcontroller can be done with the MicroBoot utlity. Before continuing, make sure that:

- The board is connected to CAN bus 1 (this requires a separate CAN controller interface connected to pins: 3 & 4 (RX&TX) on connector PD)
- The Peak PCAN interface is connected to the PC's USB port
- Power is supplied to the E407 board.
- The E407 has the correct bootloader on it

The installation of the Peak PCAN drivers can be found in 1Appendix 5

Start the MicroBoot utility. It can be found in '\Host\MicroBoot\MicroBoot.exe'. Next, click the *Browse*-button to select the Motorola S-Record of your software program. The one from the demo model is located at '\Target\HANcoder_E407.srec'.

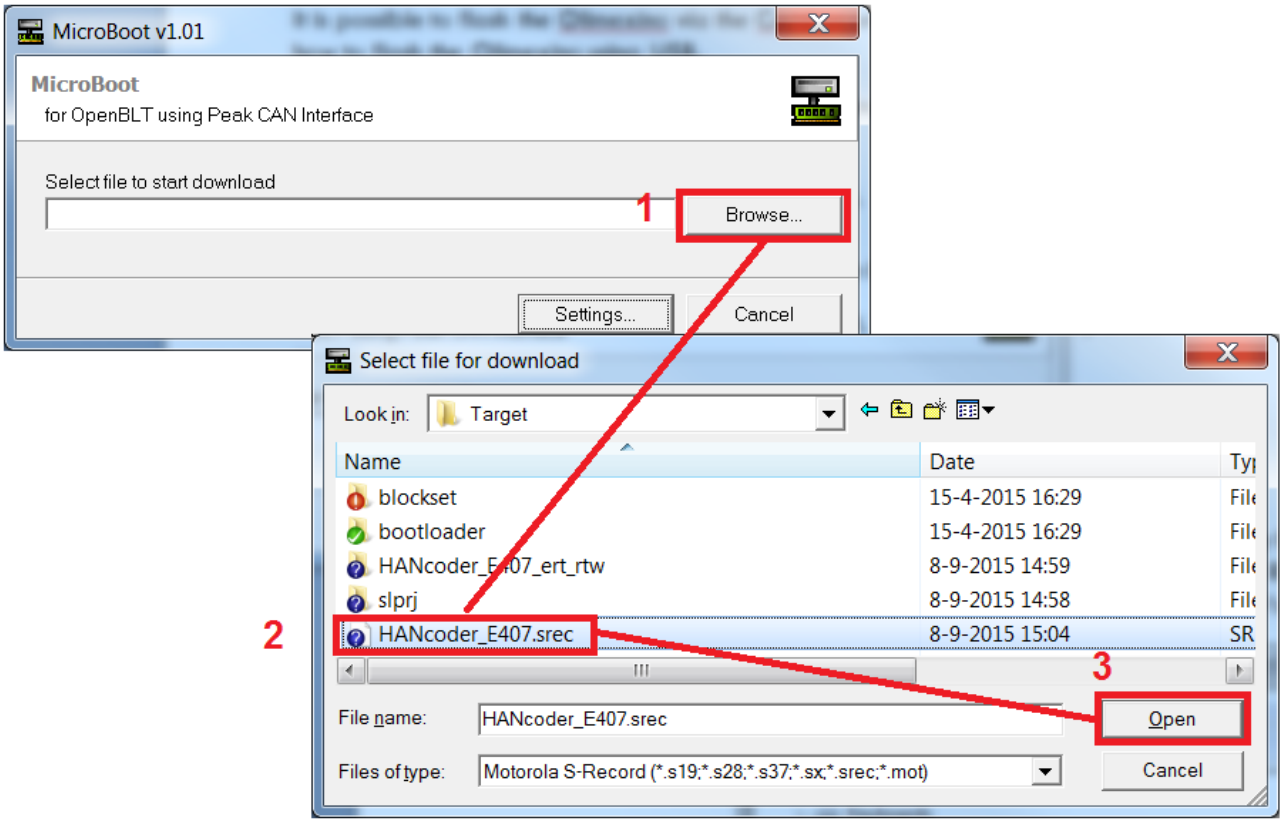



**Figure 4-5 - Select Motorola S-Record in MicroBoot**

Once selected, the firmware update with the bootloader is started. Its progress is shown in the main screen. Once completed, your newly flashed software program is automatically started.

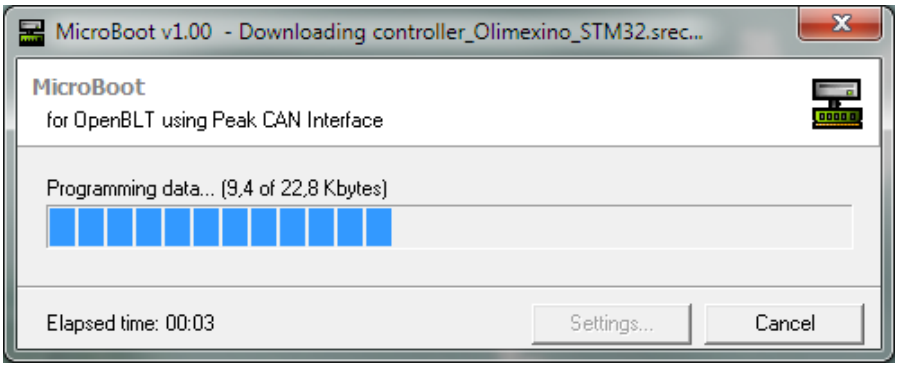



**Figure 4-6 - Bootloader progress**

# 5. FLASHING THE CONTROLLER WITH THE ST-LINK

Downloading the software program to the flash memory of the microcontroller can also be done with the help of the ST-Link. This chapter assumes the ST-Link utility is already installed. The installation of the ST-Link can be found in chapter 1Appendix 1 ST-link installation.

## 5.1. Software program reconfiguration

Before the software program can be flashed via the ST-Link a minor reconfiguration has to be done. This reconfiguration step can be performed entirely in the Simulink model. Open the Simulink model as described in chapter 3. Select *Simulation -> Configuration Parameters…* from the menu.

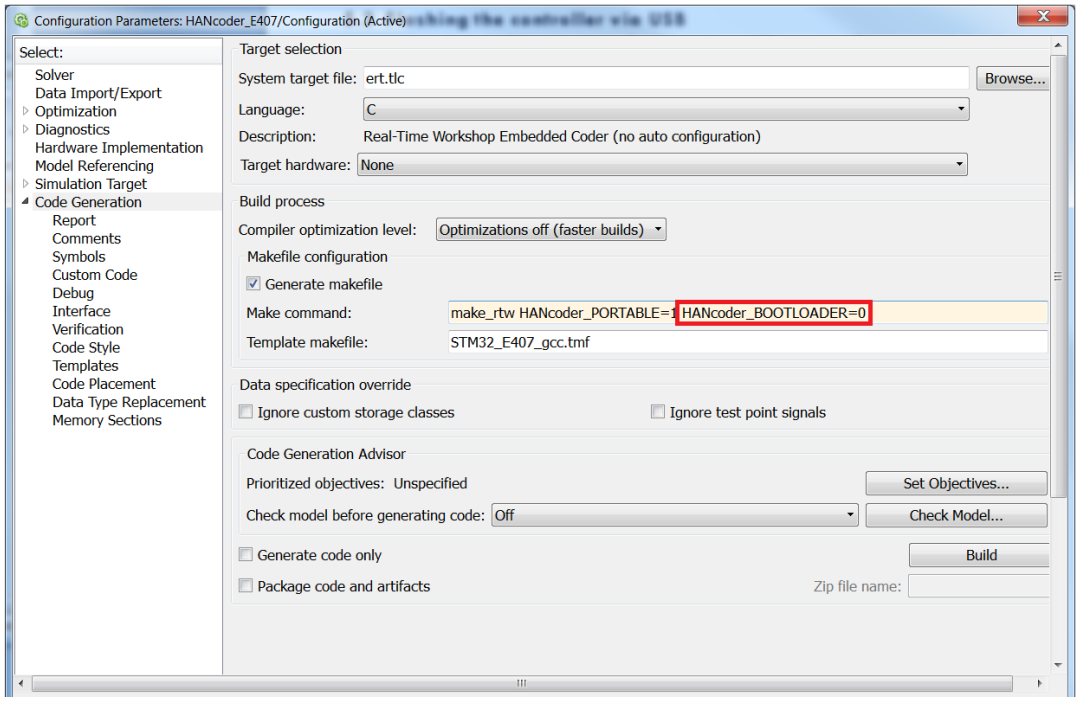



**figure 5-1 - Configuring the model for use without the bootloader**

In the Real-Time Workshop configuration screen, check “HANcoder_BOOTLOADER=0” in the *Make command* and click OK.

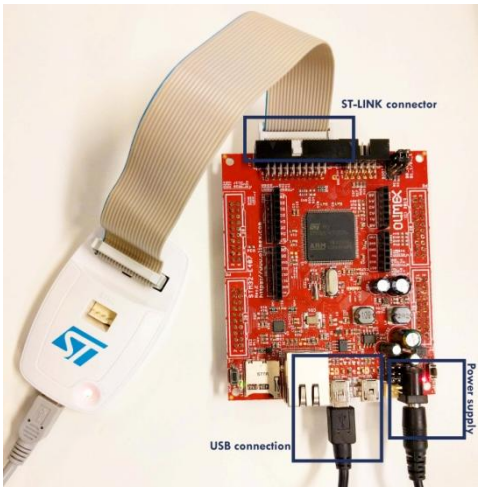
If you now generate the code and build the model, your software program is properly configured for use without the bootloader.

## 5.2. Uploading the software with the ST-Link

For programming the microcontroller with the ST-Link, make sure that:

- The ST-Link is connected to the E407 board through the J-tag connector.
- The ST-Link is connected to the PC’s USB port.
- Power is supplied to the E407 board.
- The ST-Link utility is installed

The installation of the ST-Link utility is described in chapter 1Appendix 1 ST-link installation

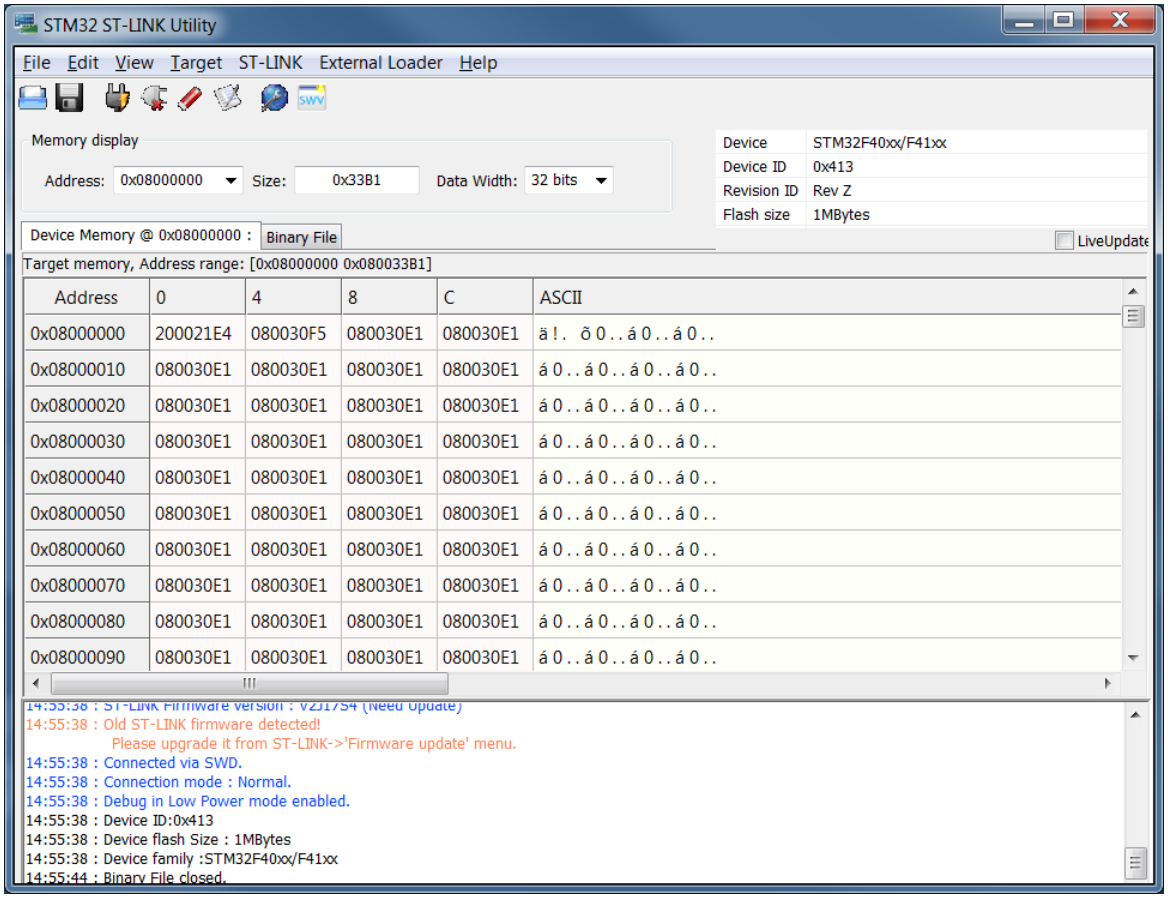



**5-2 ST_Link setup**

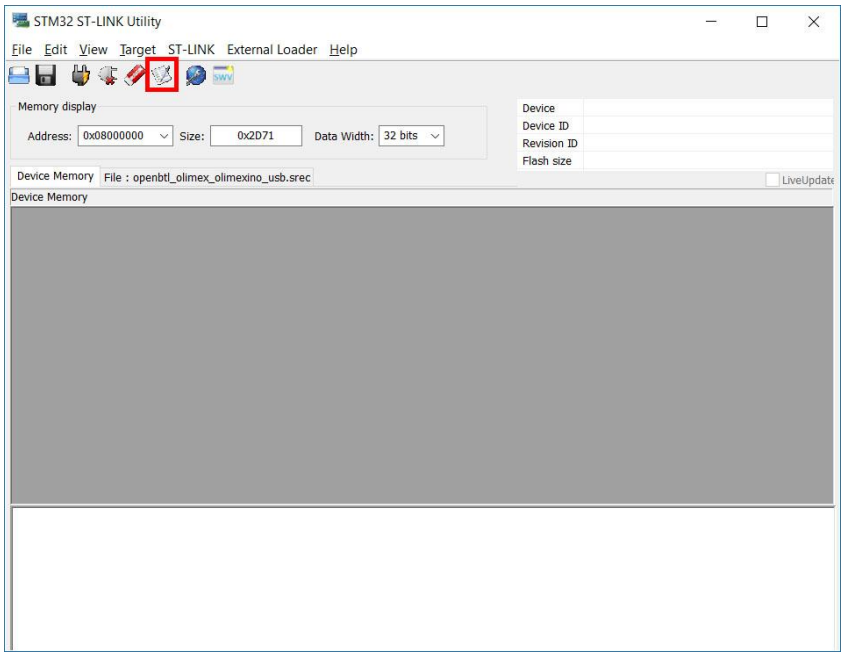
Start the STM32 ST-Link utility, located in the Start Menu under All Programs\STMicroelectronics\STM32 ST-Link Utility.

Connect to the target by clicking the “connect” button or by simply pressing “Enter”.

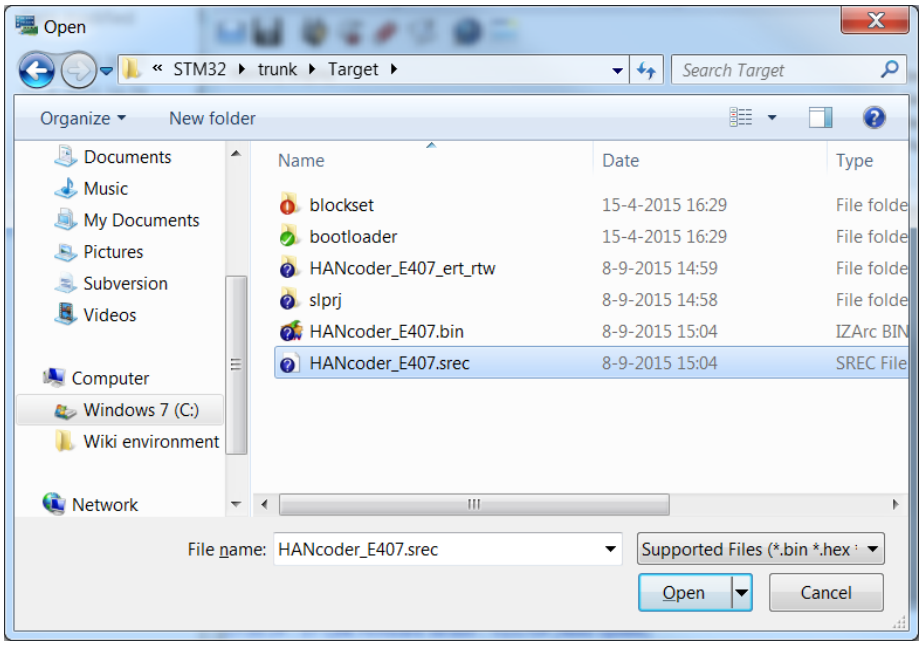
The following screen should appear:



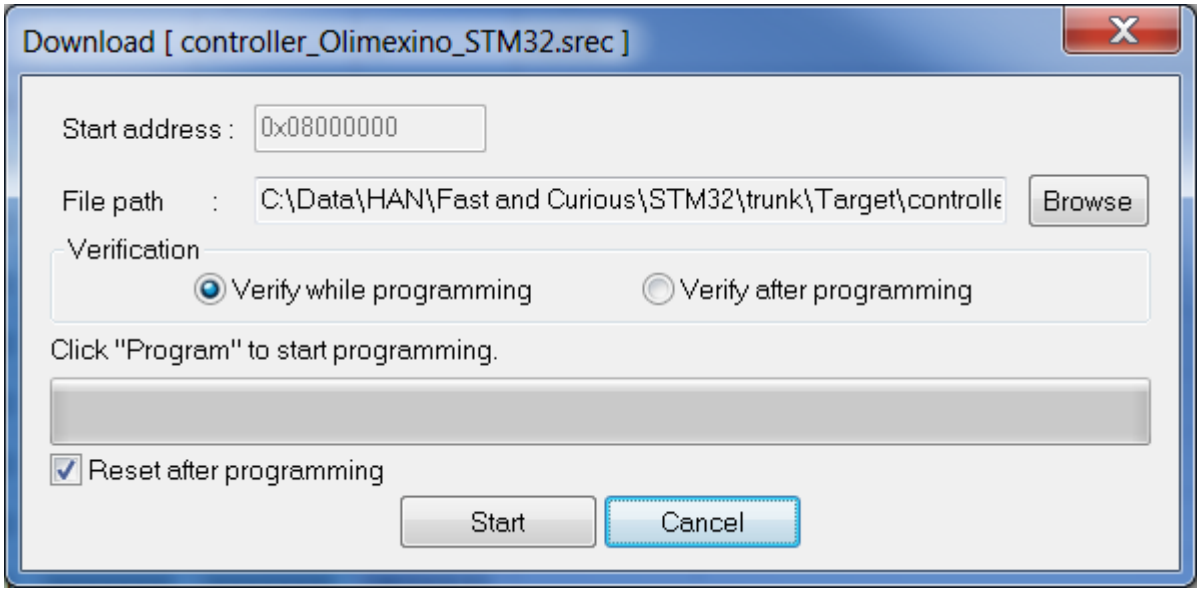

Next click “Program and Verify” in the Target menu or simply press CTRL+P



Choose the .srec file you want to flash to the controller, located in the same folder as the Simulink model:




The following screen will appear:




Press start and the flash procedure will start. The program is now flashed on the microcontroller and is ready to use. The following message will appear in the console of the STM32 ST-Link Utility:
Flash memory programmed in 4s and 509ms
Verification…OK

## 5.3. Flashing the bootloader

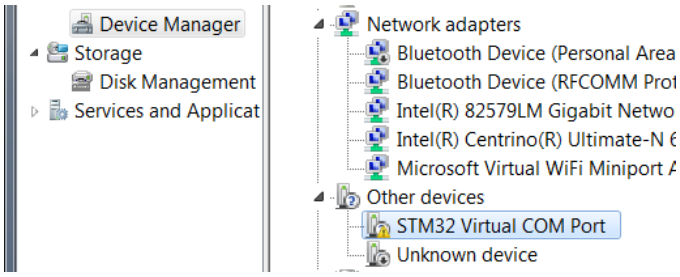
The bootloader can be flashed onto the E407 in exactly the same way as a program from Simulink. The bootloader files can be found in:
'Target\bootloader\Demo\ARMCM4_STM32_Olimex_STM32E407_GCC\Boot\bin' use the
openbtl_olimex_stm32e407_net.srec for the Ethernet bootloader, the
openbtl_olimex_stm32e407_can.srec for the CAN bootloader and the
openbtl_olimex_stm32e407_usb.srec for the USB bootloader.

# 6 HANtune

To read the variables and change the parameters that are in the model, you can use HANtune. HANtune is a Java program that reads signals (variables) and writes parameters (constants) from/to the E407. By reading these signals, you can see if the software is working correctly and the parameters can be changed to tune the algorithm.

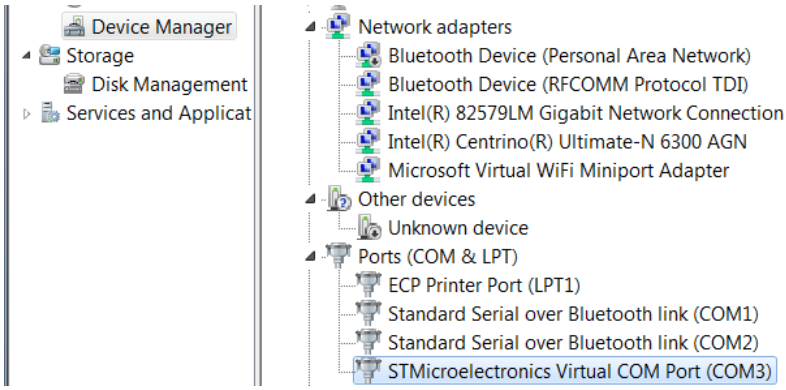
## 6.1 Installing the virtual COM port driver

To use HANtune over USB a virtual COM port is used to connect to the STM32 E407. When the E407 is connected the device manager will show it as STM32 Virtual COM port. This is not the right driver.




Install the correct driver by running VCP_V1.3.1_Setup.exe or VCP_V1.3.1_Setup_x64.exe located in the directory: /Installs/Drivers/VirtualCOMport/

Follow the instructions given by the installer. After the installation Windows should now recognize the E407 as STMicroelectronics Virtual COM Port, see below:




Please remember the COM port which has been assigned to it.

Note: When the E407 has just started up and is still in bootload mode it will show up in the device manager as WinUSB Bulk Device. After 2 seconds the program will be started and **only** if the 'XCP over USB' block is present in the model it will show up as STMicroelectronics Virtual COM Port.

## 6.2 Using HANtune

MATLAB generates an ASAP2(.a2l) file which contains the necessary information for HANtune to connect to the controller. The ASAP2 file can be loaded into HANtune by right clicking 'ASAP2 files' and choosing 'Add ASAP2 file' after it has been opened you have to load it by right clicking on the .a2l file en press 'Load File'.

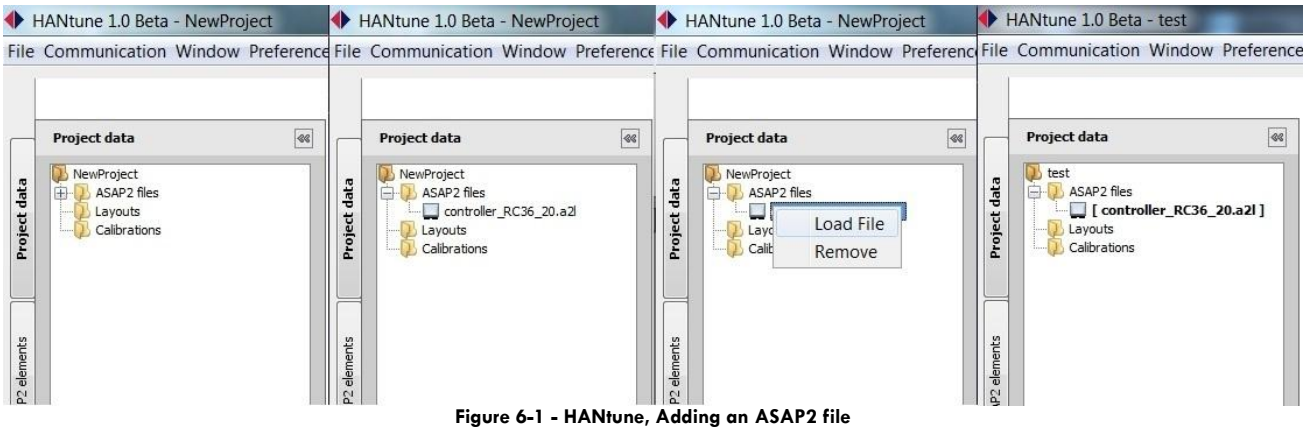



**Figure 6-1 - HANtune, Adding an ASAP2 file**

Before you can see any data, you need to create a new layout. This can be done by right clicking the layout map, and then selecting "New Layout". After you give the layout a name you can open it by right clicking on it and selecting 'Load File'.

In the layout you can change or show a value by right clicking on the variable in the ASAP2 list and selecting an editor for a parameter, or a viewer for a signal.

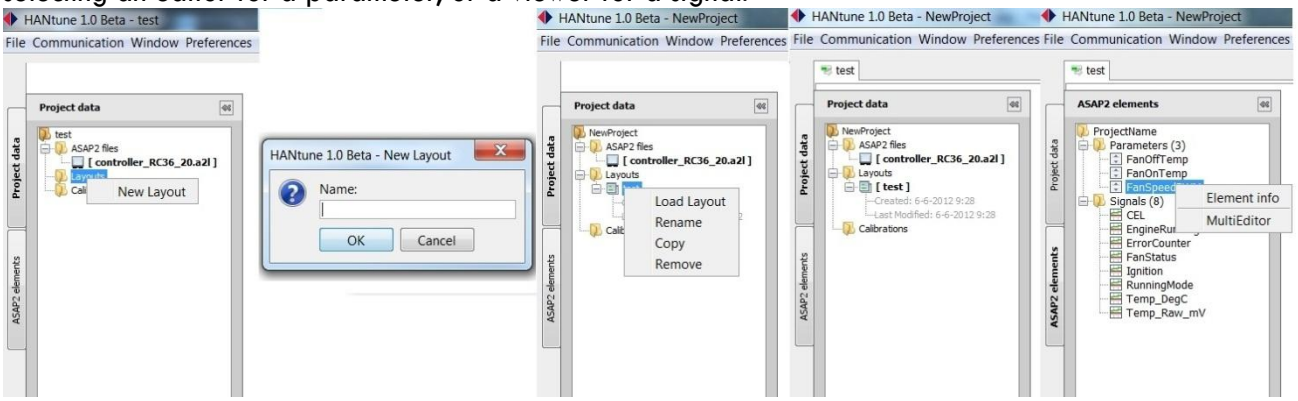



**Figure 6-2 - HANtune, Adding a layout and an item to the layout**

The maximum number of signals depends on the model running time of the model and HANtune. On a high frequency (1 kHz) you can only use a few signals without overloading the bus. The lower the frequency, the more signals you can properly use. Always check the busload and lag indictor, shown in figure 7.4, to know if you are using too many signals or not.

There are different ways to connect to the E407, using either the CANbus, USB or Ethernet. In order to use one of these communication options it has to be enabled in the model with the appropriate config block:

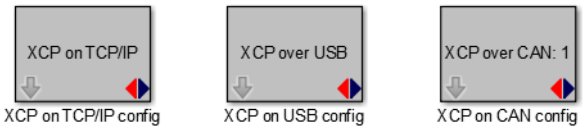



**Figure 6-3 – E407 XCP config blocks**

To select the protocol in HANtune go to the 'communication settings' located in the communication menu. In the tab 'general' the protocol can be chosen. In the other tabs the settings for each protocol can be adapted.

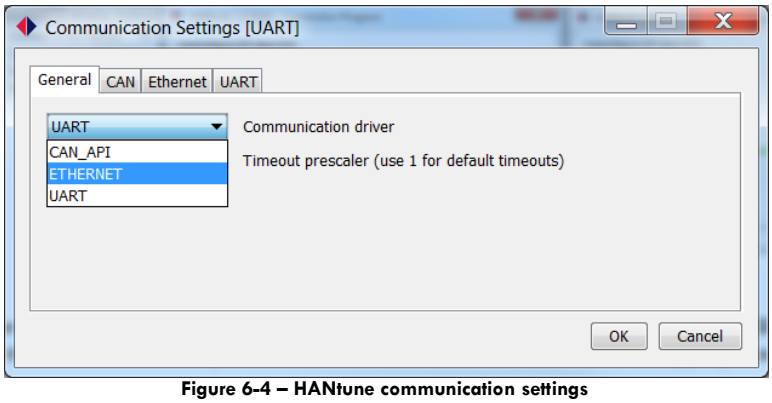



**Figure 6-4 – HANtune communication settings**

Press 'connect to XCP device' in the communication menu or press 'F5' to connect. Before HANtune sets up a connection to the controller you need to define if you want to 'Connect & Request' or 'Connect & Calibrate'. If you choose the 'Connect & Request' the parameters of the controller will be shown at your screen. By pressing 'Connect & Calibrate' the parameters that have been set in HANtune will be send to the controller when connecting.

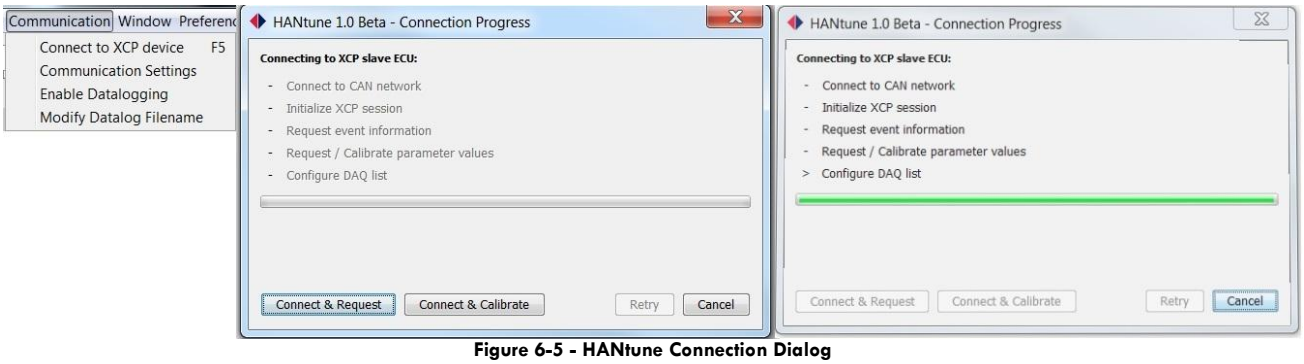



**Figure 6-5 - HANtune Connection Dialog**

The status bar on the bottom of the screen gives information to the user about the connection. It shows to which controller HANtune is connected, the amount of Parameters and Signals, the maximum frequency, a prescaler to change the frequency, the calculated real frequency, and a bar to enable data logging. The real frequency is the frequency used by HANtune to refresh the signals. At the right side of the status bar there is an indicator that turns clockwise to show if there are signals transmitted. This indicator will color red if there is data loss. The bars left from the log file bar are a lag indicator showing how much of the received data is actually shown on the screen. Although you do not see some of that data, it is still logged in the log file. The DAQ list bar shows the **estimated** busload by the number of signals and the used prescaler.

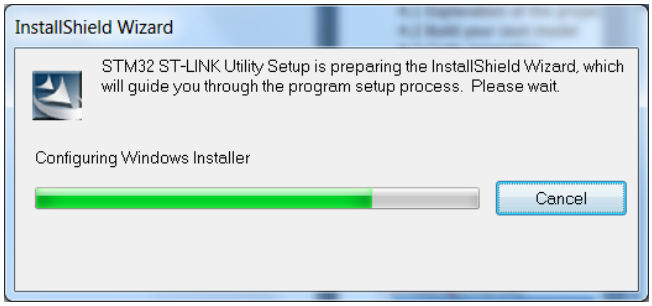
**Figure 6-6 - HANtune Communication status**

# APPENDIX 1. ST-LINK INSTALLATION

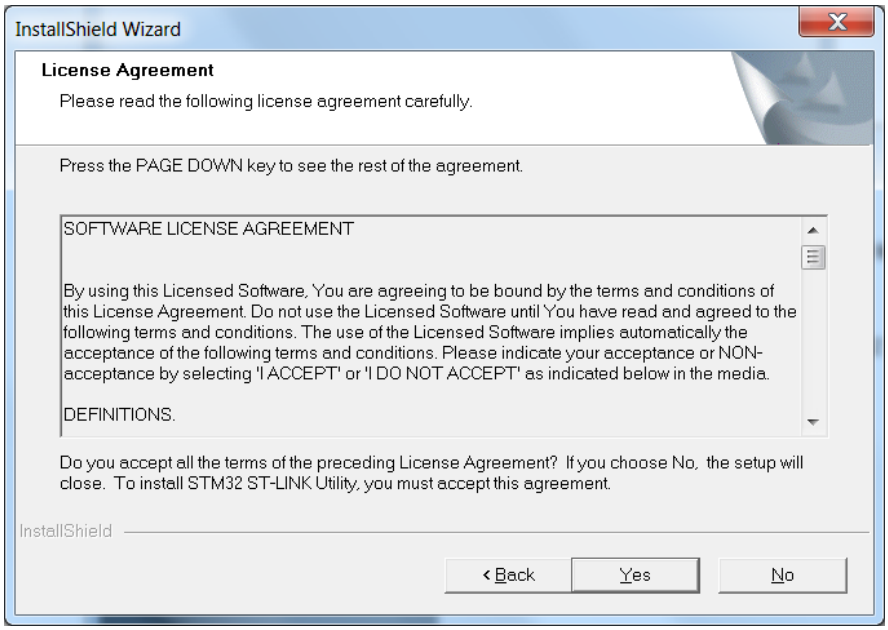
The STM32 ST-Link Utility can be downloaded for free from www.ST.com. To be specific:

http://www.st.com/web/en/catalog/tools/PF258168

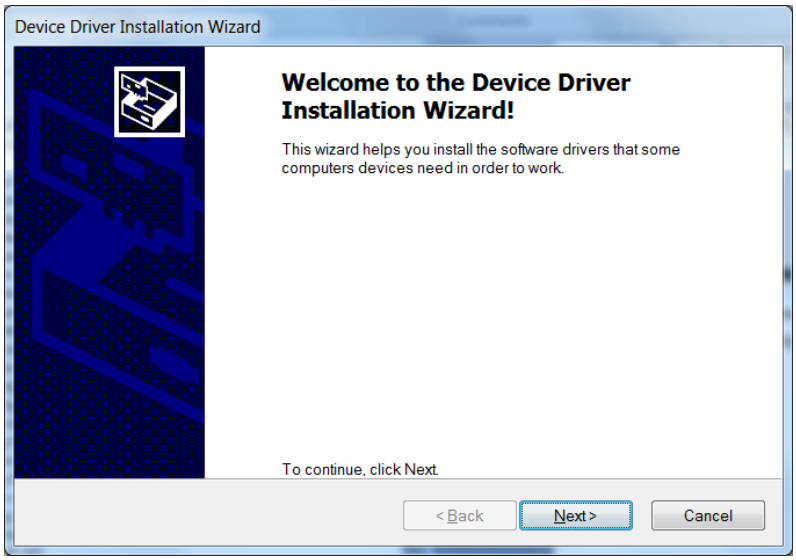
Once downloaded unzip the executable: 'STM32 ST-LINK Utility_v3.6.0.exe' and run it.



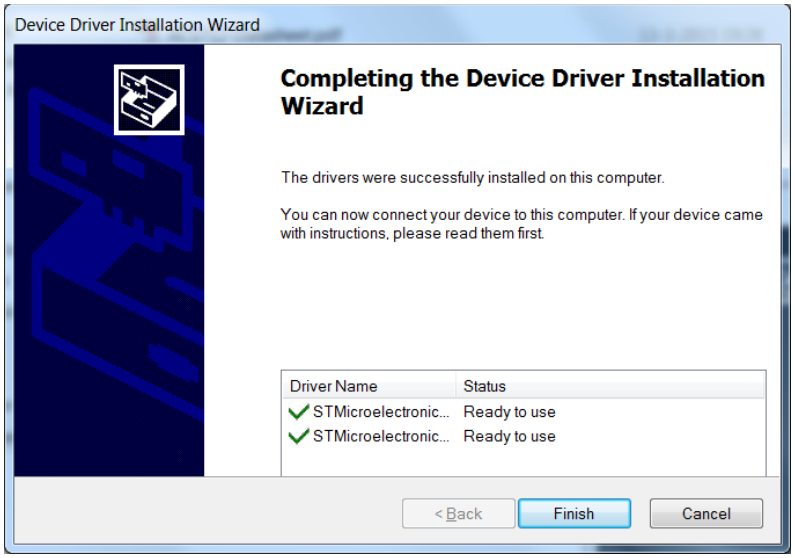

Click next until the license agreement screen appears:




After accepting the license agreement in the next screen you can choose the installation path. You are free to choose any installation path. When you click next the application is installed and a new screen appears which allows you to install the driver.




If the program will ask for permission, press allow to install the drivers.



# APPENDIX 2. SEGGER J-LINK INSTALLATION

This part is only necessary if you do not possess an E407 with the correct bootloader or if you don't want to flash a program via CAN. For downloading a program to the flash memory of the microcontroller, the Segger J-Link interface is required:

**Figure 2-1 - Segger J-Link interface**

## Installation

On the CD that was delivered with the Segger J-Link interface, you can find the J-Link tools installer. Start the installation by double-clicking the executable, for example 'Setup_JLinkARM_V470a.exe'.

Accept the license agreement by clicking 'Yes'.

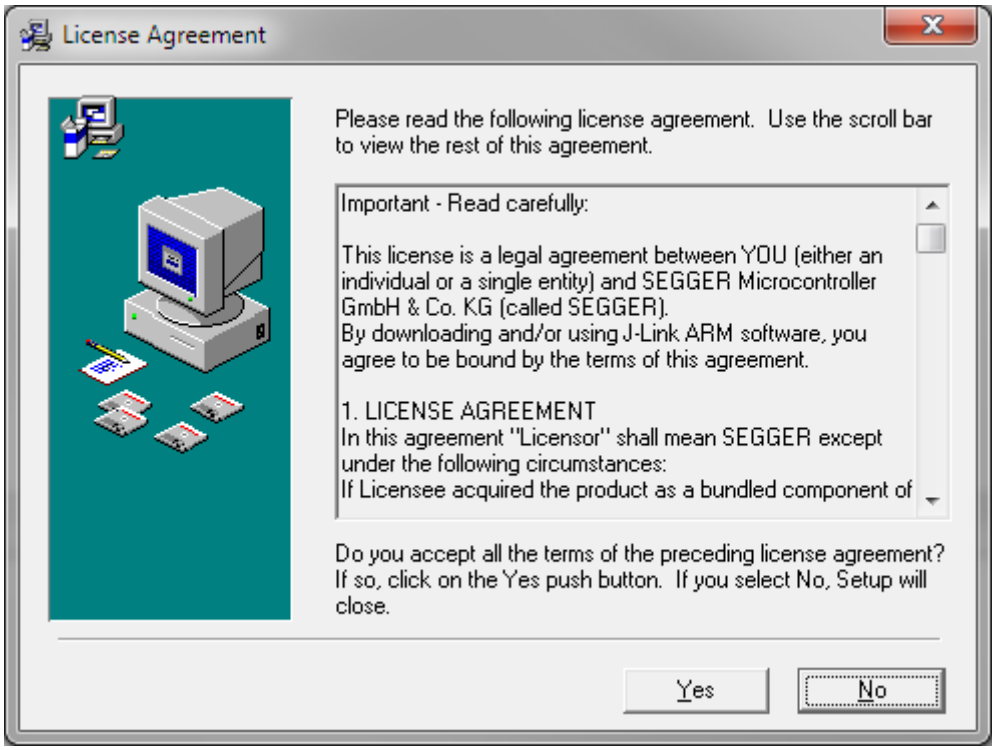



**Figure 2-2 - Accept license agreement**

All default installer settings are correct, so keep clicking next until the actual installation starts:

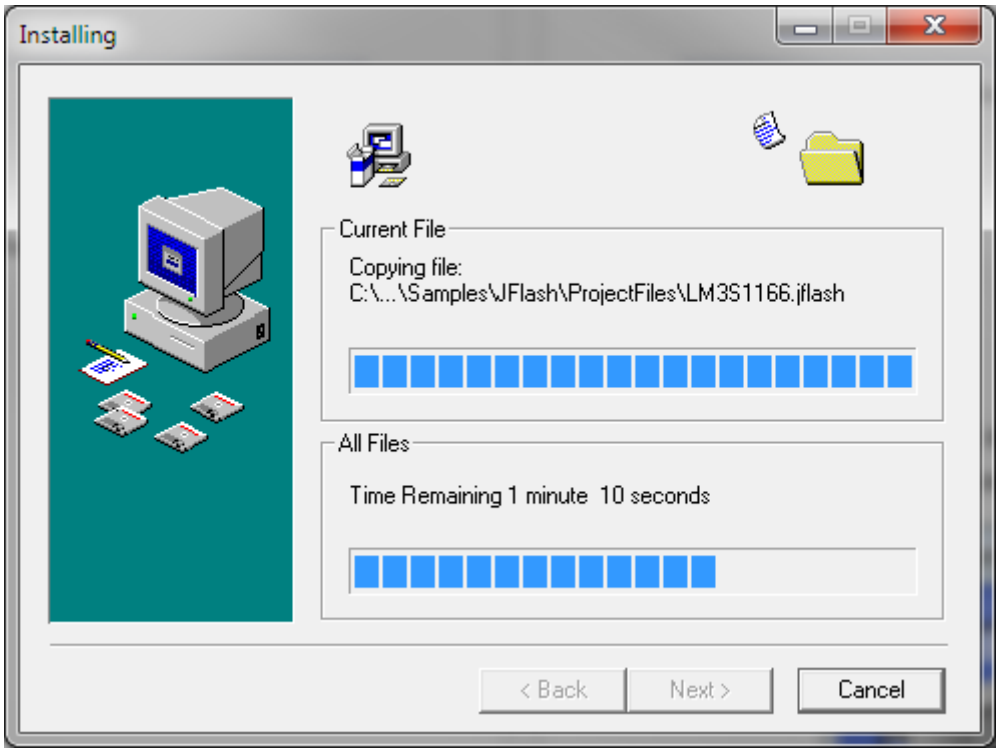



**Figure 2-3 - Installation process**

Once done, click 'Finish' to complete the installation:

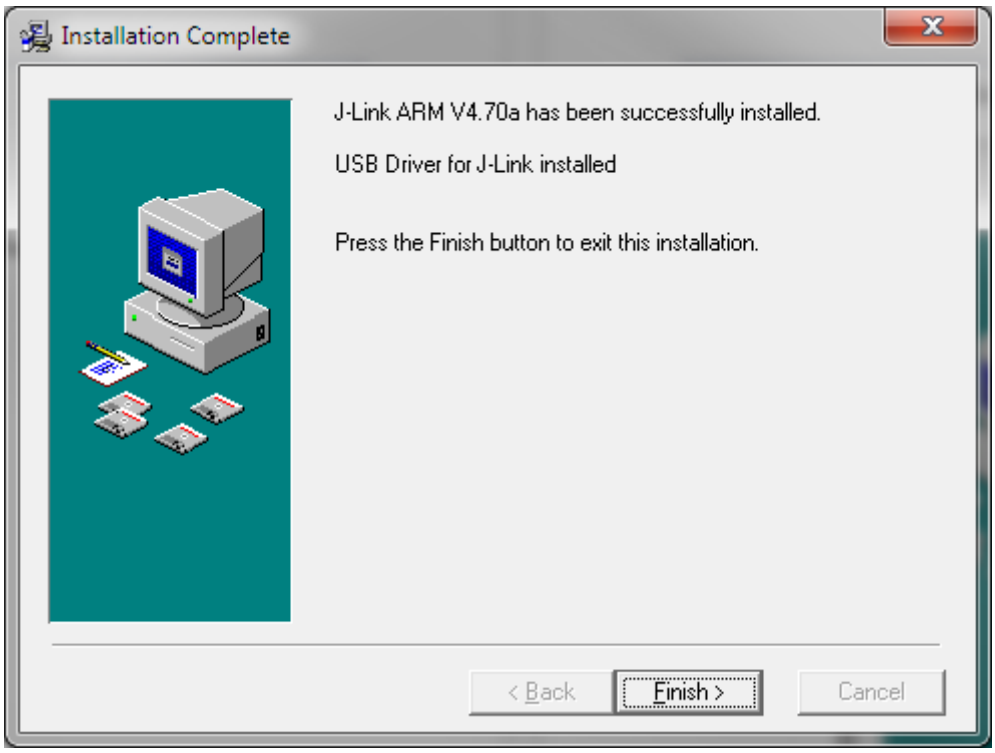



**Figure 2-4 - Installation completion**

# APPENDIX 3. FLASHING THE BOOTLOADER WITH THE J-LINK

For flashing the bootloader with the Segger J-Link it is necessary that all drivers and software tools for this debugger interface are already installed. If not, then follow the installation instructions in the previous paragraph.

Before continuing, make sure that:

- The Segger J-Link is connected to the STM32-E407 board.
- The Segger J-Link is connected to the PC's USB port.
- Power is supplied to the STM32-E407 board.

Continue by starting the J-Link GDB Server. A link to this program can be found in the Windows Start Menu:

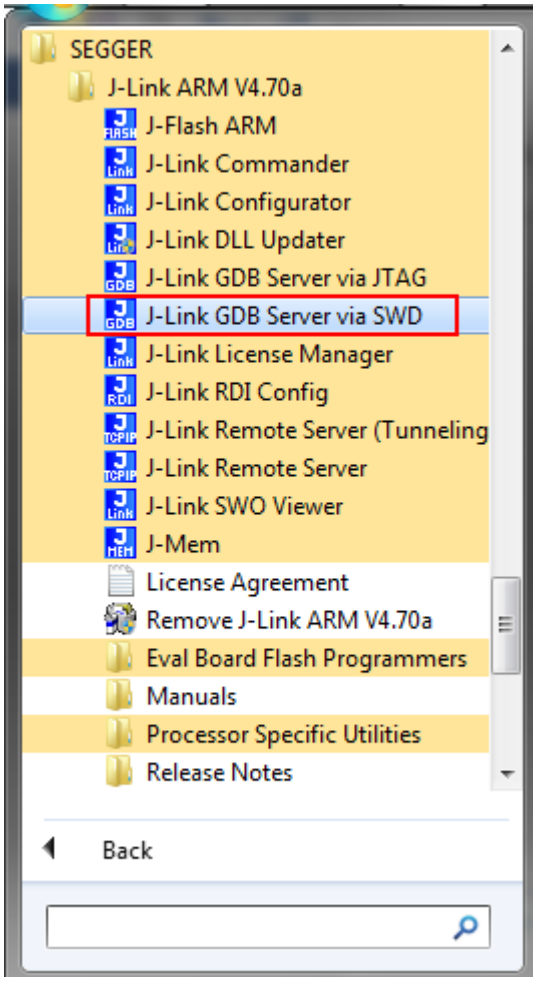



**Figure 3-1 - Start the J-Link GDB Server**

Configure the following settings for the STM32-E407:

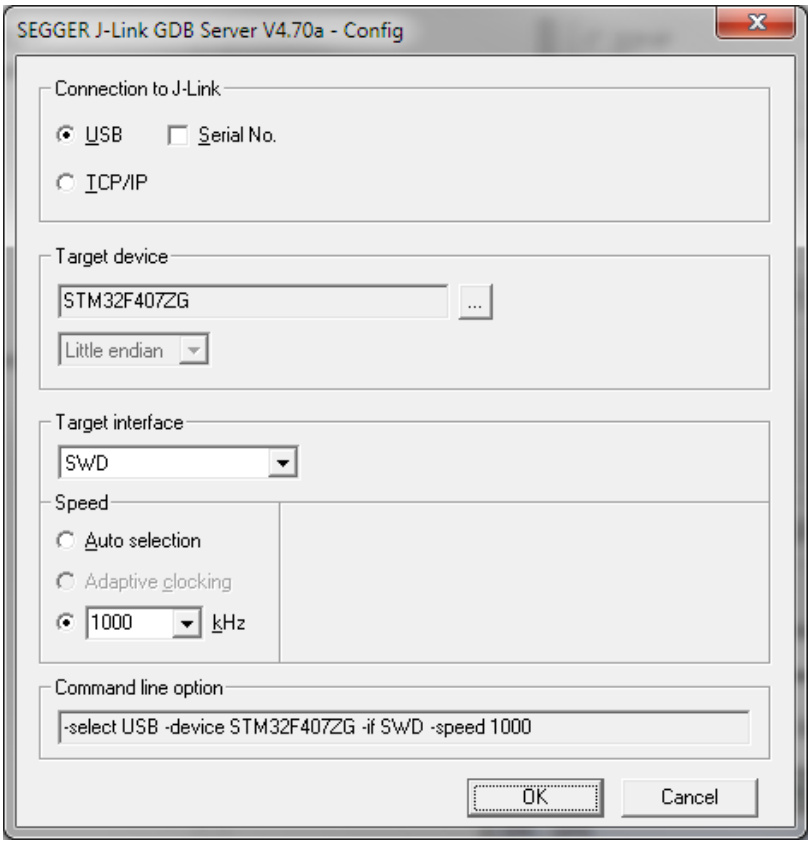



**Figure 3-2- J-Link GDB Server settings**

Finally, click 'Ok' to start the server. The following screen appears:

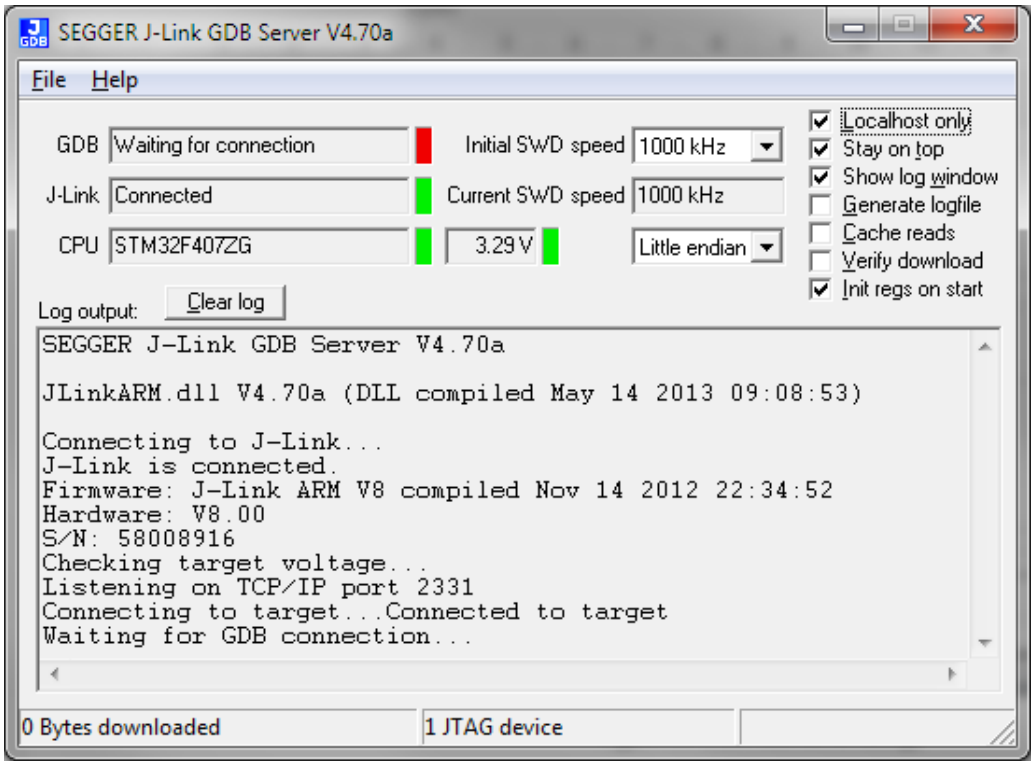



**Figure 3-3 - J-Link GDB server running**

Flash programming a bootloader is done through a batch-file located in directory: '\Target\bootloader\Demo\ARMCM4_STM32_Olimex_STM32E407_GCC\Boot\cmd'.

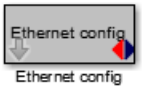
- To start flash programming the bootloader with support for firmware updates via **CAN,** simply double-click the batch-file called 'gdbflash_can.bat'.
- To start flash programming the bootloader with support for firmware updates via **USB,** simply double-click the batch-file called 'gdbflash_usb.bat'.
- To start flash programming the bootloader with support for firmware updates via **TCP/IP,** simply double-click the batch-file called 'gdbflash_net.bat'.

This will only take a few seconds. After flash programming, the software program is started after pressing and releasing the reset button on the board.

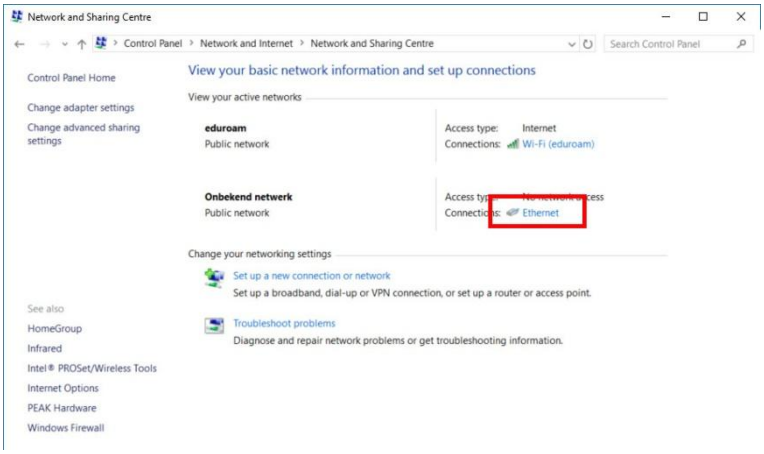
# APPENDIX 4. CONFIGURING PC TO FLASH THE E407 OVER ETHERNET

In this paragraph it is described how to directly connect your PC to the E407 board so HANtune and MicroBoot can connect to it. This manual is written for Windows **10**.

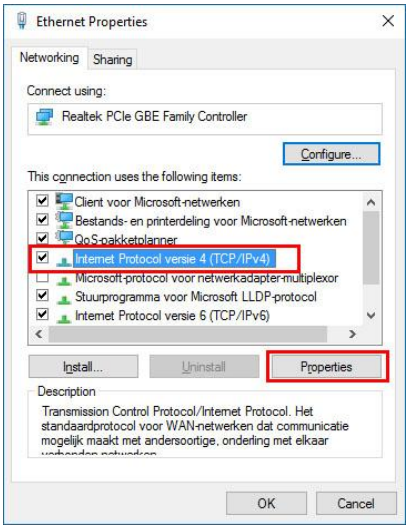
First the Ethernet address that the E407 wants to use has to be known. The standard Ethernet address for the bootloader is: 169.254.19.63 The IP address for HANtune can be chosen in the 'Ethernet config' block in Simulink. Its default setting is also 169.254.19.63



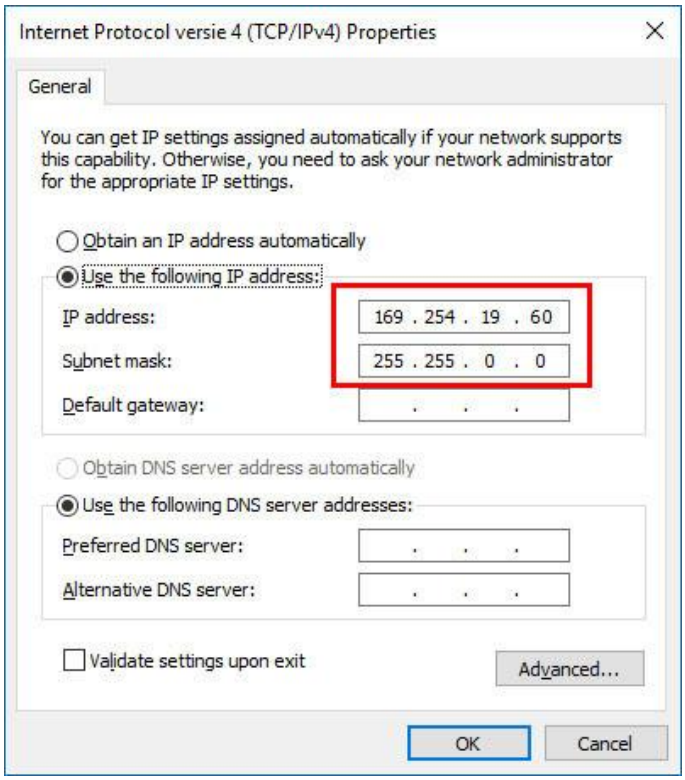

- Connect the Ethernet cable directly between the E407 and your computer.
- Open the 'Network and Sharing Center' by clicking the 'View network status and tasks' in the Control Panel.
- Click on **Ethernet**




- Go to Properties and select the 'Internet Protocol Version4(TCP/IPv4)' and click on properties once again



- Adjust the IP address to an address which is in the same range as the IP address of the E407 but not exactly the same address. In this case the pc IP address is set to 169.254.19.60. Also adjust the Subnet mask to match the subnet mask in the settings of the E407. Leave the Default gateway and DNS settings open. See following screen:




- Your connection should now work. You can test the connection by pinging the E407 as described in chapter 4.2 Flashing the controller via TCP/IP

# APPENDIX 5. PEAK PCAN INSTALLATION

To download the PEAK PCAN package click here.

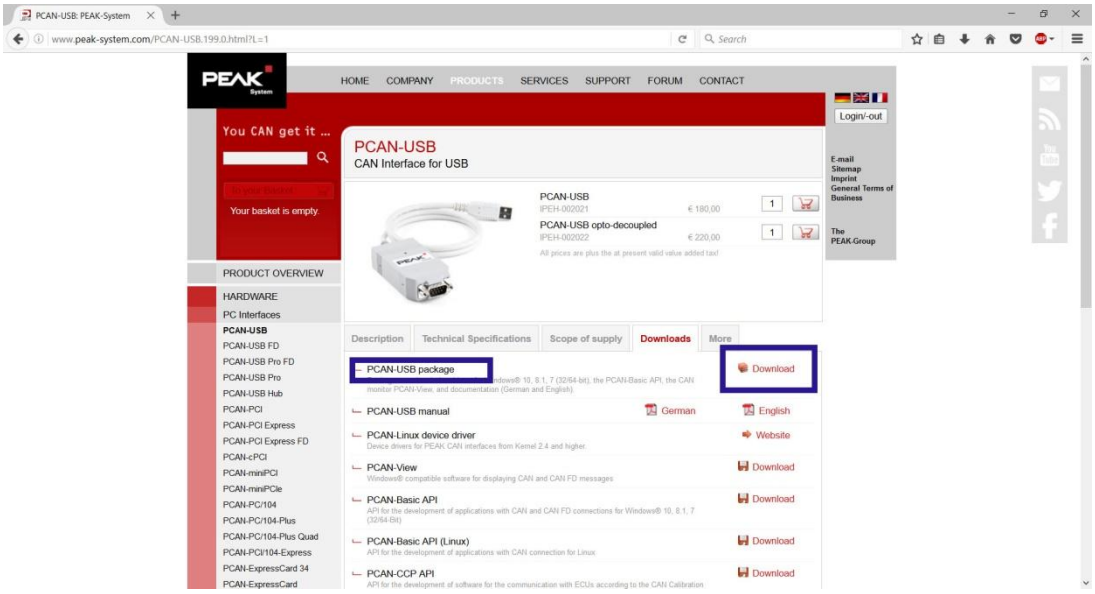



**Figure 5-1 PEAK Installation**

Extract the zip file at the desired folder and run the PeakOemDrv.exe to install the drivers. The zip file also contains PCAN-View, a program which will come out handy for CAN communication.

# APPENDIX 6. REBUILDING THE BOOTLOADER

The bootloader comes with full source code and it can therefore easily be modified to fit your personal needs. The bootloader documentation can be found at http://www.feaser.com/openblt/.

The most important settings of the bootloader are the IP address, network mask and gateway address. These settings can be changed in the blt_conf.h file located in 'Target\bootloader\Demo\ARMCM4_STM32_Olimex_STM32E407_GCC\Boot\'.

After modifying the settings of the bootloader, it needs to be rebuild.

Rebuilding the bootloader is achieved by the following step:

1. Double-click the batch-file called 'build.bat' in directory: '\Target\bootloader\Demo\ ARMCM4_STM32_Olimex_STM32E407_GCC\Boot\cmd'.

After rebuilding the bootloader, it needs to be programmed again into the internal flash memory of the microcontroller