HANcoder STM32 Target Getting Started

Version 0.4
9/28/2016

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 Olimexino STM32.

HANcoder STM32 Target Getting Started

GETTING STARTED GUIDE

Contents

VERSIONING

Nr	Date	Person	Changes	Status
0.0.1	19-12-2013	F. Voorburg	First version	Concept
0.0.2	13-08-2014	J van Kolfschoten	Added information about the supported MATLAB versions Added a chapter for flashing with the CAN dongle	Concept
0.0.3	29-08-2014	J van Kolfschoten	Rewritten the MATLAB installation Added Troubleshooting chapter	Draft
0.0.4	18-09-2014	J van Kolfschoten	Added description for programming with the ST-Link	Draft
0.2.0	02-12-2014	F. Voorburg	Updated for USB bootloader	Draft
0.3	07-09-2015	J van Kolfschoten	Moved some parts to the appendices and made the manual more readable	Ready for review
0.4	28-09-2016	G Jansma J van Kolfschoten	Added and Improved figures improved text for better understanding	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 Olimexino STM32 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 Olimexino.

Furthermore, it will explain how you can view signals and edit parameters while the software runs on the Olimexino. 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 would 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 Olimexino Target:

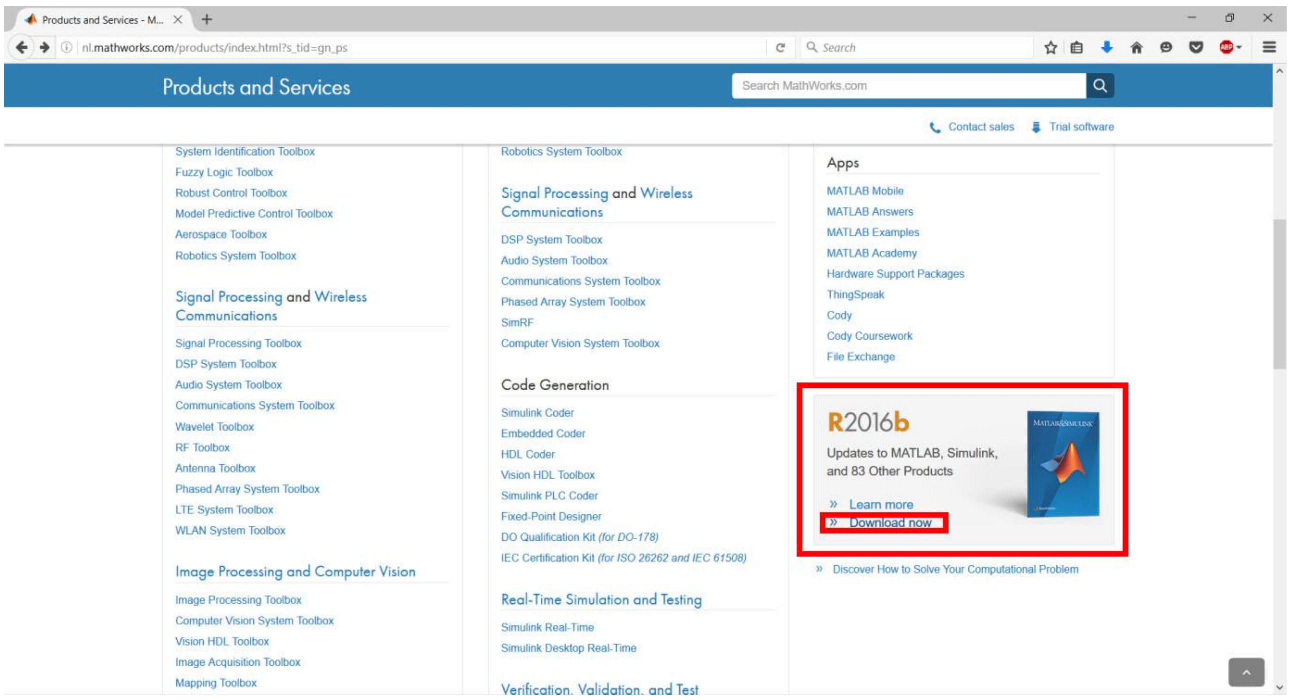
MATLAB 32-bit, 2009a until 2014a

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.



Products and Services - M...

nl.mathworks.com/products/index.html?s_tid=gn_ps

Products and Services

Search MathWorks.com

Contact sales   Trial software

System Identification Toolbox
Fuzzy Logic Toolbox
Robust Control Toolbox
Model Predictive Control Toolbox
Aerospace Toolbox
Robotics System Toolbox

Signal Processing and Wireless Communications

Signal Processing Toolbox
DSP System Toolbox
Audio System Toolbox
Communications System Toolbox
Wavelet Toolbox
RF Toolbox
Antenna Toolbox
Phased Array System Toolbox
LTE System Toolbox
WLAN System Toolbox

Image Processing and Computer Vision

Image Processing Toolbox
Computer Vision System Toolbox
Vision HDL Toolbox
Image Acquisition Toolbox
Mapping Toolbox

Robotics System Toolbox

Signal Processing and Wireless Communications

DSP System Toolbox
Audio System Toolbox
Communications System Toolbox
Phased Array System Toolbox
SimRF
Computer Vision System Toolbox

Code Generation

Simulink Coder
Embedded Coder
HDL Coder
Vision HDL Toolbox
Simulink PLC Coder
Fixed-Point Designer
DO Qualification Kit *(for DO-178)*
IEC Certification Kit *(for ISO 26262 and IEC 61508)*

Real-Time Simulation and Testing

Simulink Real-Time
Simulink Desktop Real-Time

Verification, Validation, and Test

Apps

MATLAB Mobile
MATLAB Answers
MATLAB Examples
MATLAB Academy
Hardware Support Packages
ThingSpeak
Cody
Cody Coursework
File Exchange

R2016b

Updates to MATLAB, Simulink, and 83 Other Products

» Learn more
» Download now

» Discover How to Solve Your Computational Problem

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

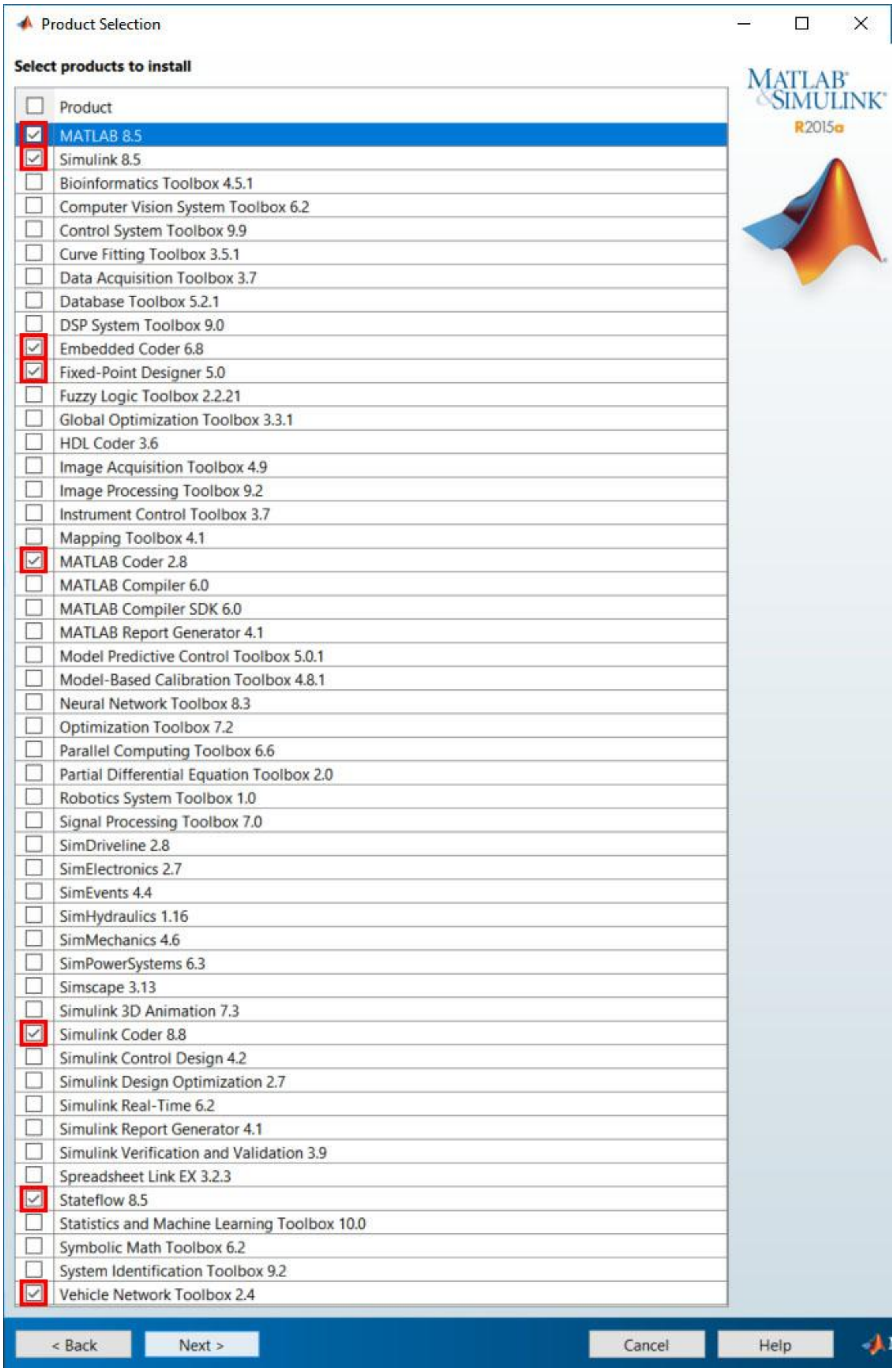
While installing Matlab you're being asked to select the products to install. As previous said the required toolboxes are:

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

The following options 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:



Product Selection

Select products to install

	Product
✓	MATLAB 8.5
✓	Simulink 8.5
☐	Bioinformatics Toolbox 4.5.1
☐	Computer Vision System Toolbox 6.2
☐	Control System Toolbox 9.9
☐	Curve Fitting Toolbox 3.5.1
☐	Data Acquisition Toolbox 3.7
☐	Database Toolbox 5.2.1
☐	DSP System Toolbox 9.0
✓	Embedded Coder 6.8
✓	Fixed-Point Designer 5.0
☐	Fuzzy Logic Toolbox 2.2.21
☐	Global Optimization Toolbox 3.3.1
☐	HDL Coder 3.6
☐	Image Acquisition Toolbox 4.9
☐	Image Processing Toolbox 9.2
☐	Instrument Control Toolbox 3.7
☐	Mapping Toolbox 4.1
✓	MATLAB Coder 2.8
☐	MATLAB Compiler 6.0
☐	MATLAB Compiler SDK 6.0
☐	MATLAB Report Generator 4.1
☐	Model Predictive Control Toolbox 5.0.1
☐	Model-Based Calibration Toolbox 4.8.1
☐	Neural Network Toolbox 8.3
☐	Optimization Toolbox 7.2
☐	Parallel Computing Toolbox 6.6
☐	Partial Differential Equation Toolbox 2.0
☐	Robotics System Toolbox 1.0
☐	Signal Processing Toolbox 7.0
☐	SimDriveline 2.8
☐	SimElectronics 2.7
☐	SimEvents 4.4
☐	SimHydraulics 1.16
☐	SimMechanics 4.6
☐	SimPowerSystems 6.3
☐	Simscape 3.13
☐	Simulink 3D Animation 7.3
✓	Simulink Coder 8.8
☐	Simulink Control Design 4.2
☐	Simulink Design Optimization 2.7
☐	Simulink Real-Time 6.2
☐	Simulink Report Generator 4.1
☐	Simulink Verification and Validation 3.9
☐	Spreadsheet Link EX 3.2.3
✓	Stateflow 8.5
☐	Statistics and Machine Learning Toolbox 10.0
☐	Symbolic Math Toolbox 6.2
☐	System Identification Toolbox 9.2
✓	Vehicle Network Toolbox 2.4

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**

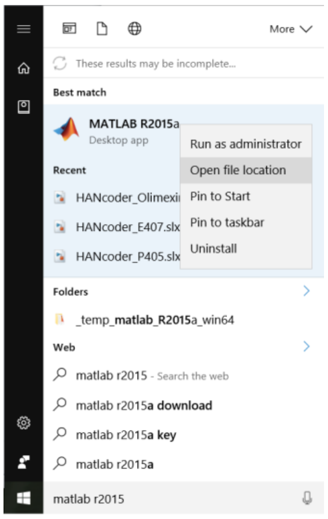
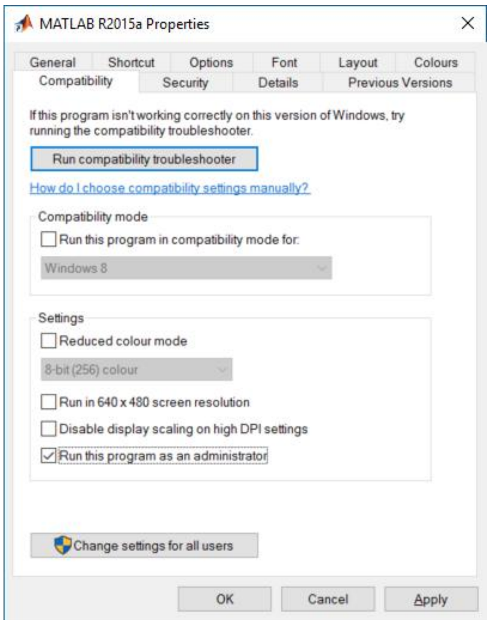


Figure 1-3 Windows 10 Start menu, searh section

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**.


MATLAB R2015a Properties

General | Shortcut | Options | Font | Layout | Colours | Compatibility | Security | Details | Previous Versions

If this program isn't working correctly on this version of Windows, try running the compatibility troubleshooter.

Run compatibility troubleshooter

How do I choose compatibility settings manually?

Compatibility mode
- ☐ Run this program in compatibility mode for:
- Windows 8

Settings
- ☐ Reduced colour mode
- 8-bit (256) colour
- ☐ Run in 640 x 480 screen resolution
- ☐ Disable display scaling on high DPI settings
- ☑ Run this program as an administrator

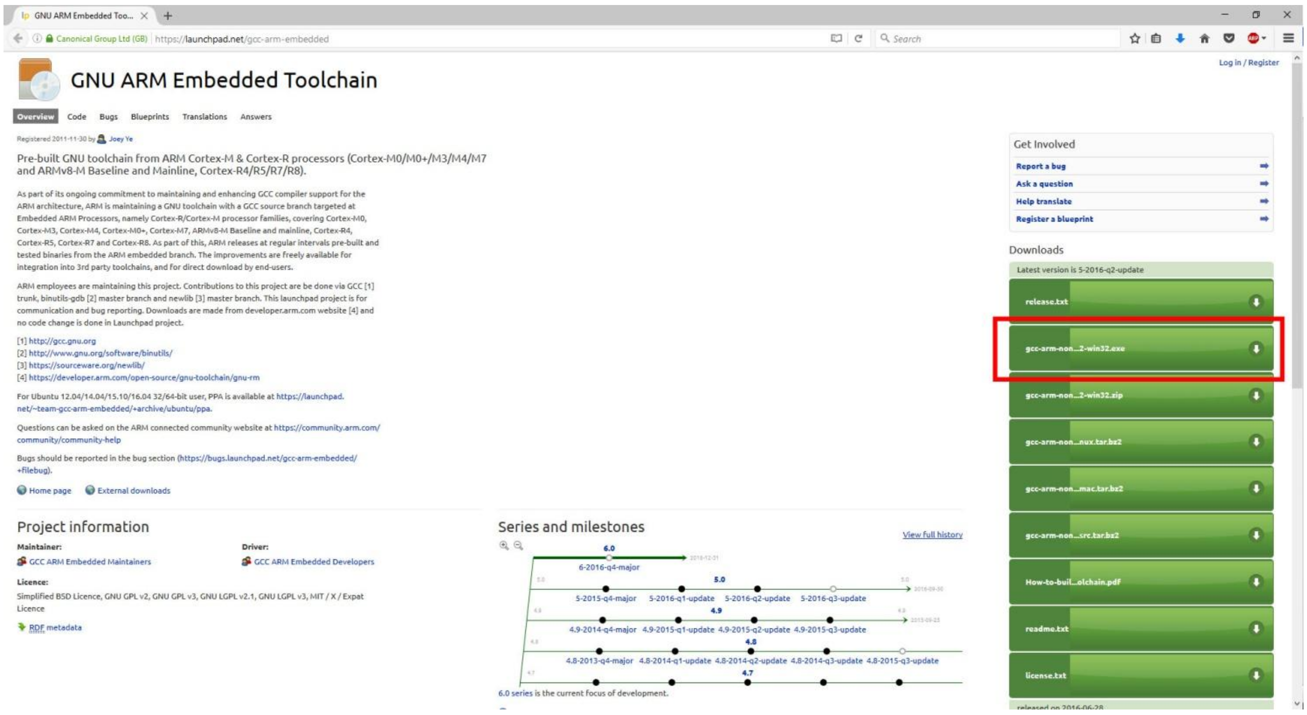
Change settings for all users

OK | Cancel | Apply

Figure 1-4 MATLAB R2015a Properties, Compatibility tab

2 GNU ARM TOOLCHAIN INSTALLATION

The Olimexino 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:


GNU ARM Embedded Too...
Canonical Group Ltd (GB) https://launchpad.net/gcc-arm-embedded

GNU ARM Embedded Toolchain

Overview Code Bugs Blueprints Translations Answers

Registered 2011-11-30 by Joey Ye

Pre-built GNU toolchain from ARM Cortex-M & Cortex-R processors (Cortex-M0/M0+/M3/M4/M7 and ARMv8-M Baseline and Mainline, Cortex-R4/R5/R7/R8).

As part of its ongoing commitment to maintaining and enhancing GCC compiler support for the ARM architecture, ARM is maintaining a GNU toolchain with a GCC source branch targeted at Embedded ARM Processors, namely Cortex-R/Cortex-M processor families, covering Cortex-M0, Cortex-M3, Cortex-M4, Cortex-M0+, Cortex-M7, ARMv8-M Baseline and mainline, Cortex-R4, Cortex-R5, Cortex-R7 and Cortex-R8. As part of this, ARM releases at regular intervals pre-built and tested binaries from the ARM embedded branch. The improvements are freely available for integration into 3rd party toolchains, and for direct download by end-users.

ARM employees are maintaining this project. Contributions to this project are be done via GCC [1] trunk, binutils-gdb [2] master branch and newlib [3] master branch. This launchpad project is for communication and bug reporting. Downloads are made from developer.arm.com website [4] and no code change is done in Launchpad project.

[1] http://gcc.gnu.org
[2] http://www.gnu.org/software/binutils/
[3] https://sourceware.org/newlib/
[4] https://developer.arm.com/open-source/gnu-toolchain/gnu-rm

For Ubuntu 12.04/14.04/15.10/16.04 32/64-bit user, PPA is available at https://launchpad.net/~team-gcc-arm-embedded/+archive/ubuntu/ppa.

Questions can be asked on the ARM connected community website at https://community.arm.com/community/community-help

Bugs should be reported in the bug section (https://bugs.launchpad.net/gcc-arm-embedded/+filebug).

Home page External downloads

Project information

Maintainer: GCC ARM Embedded Maintainers

Driver: GCC ARM Embedded Developers

Licence:
Simplified BSD Licence, GNU GPL v2, GNU GPL v3, GNU LGPL v2.1, GNU LGPL v3, MIT / X / Expat Licence

RDF metadata

Series and milestones

View full history

6.0 series is the current focus of development.

Log in / Register

Get Involved
- Report a bug
- Ask a question
- Help translate
- Register a blueprint

Downloads

Latest version is 5-2016-q2-update
- release.txt
- gcc-arm-non...2-win32.exe
- gcc-arm-non...2-win32.zip
- gcc-arm-non...nux.tar.bz2
- gcc-arm-non...mac.tar.bz2
- gcc-arm-non...src.tar.bz2
- How-to-buil...olchain.pdf
- readme.txt
- license.txt

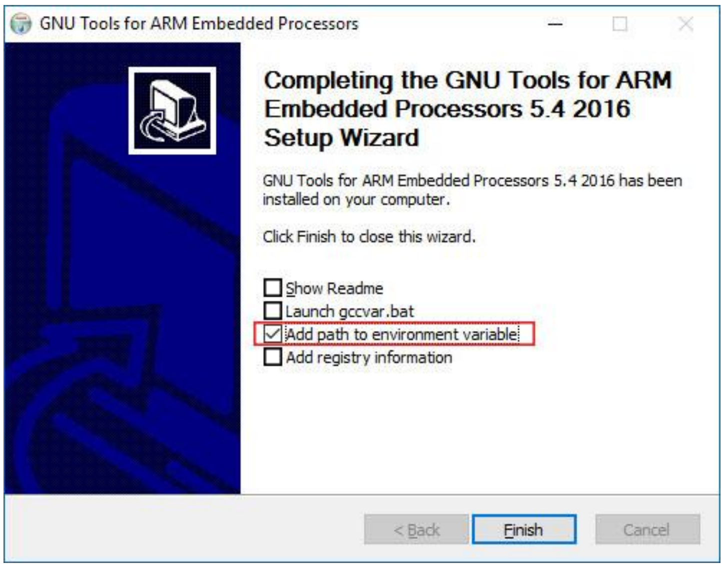
released on 2016-06-28

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.


GNU Tools for ARM Embedded Processors

Completing the GNU Tools for ARM Embedded Processors 5.4 2016 Setup Wizard

GNU Tools for ARM Embedded Processors 5.4 2016 has been installed on your computer.

Click Finish to close this wizard.

- [ ] Show Readme
- [ ] Launch gccvar.bat
- [✓] Add path to environment variable
- [ ] Add registry information

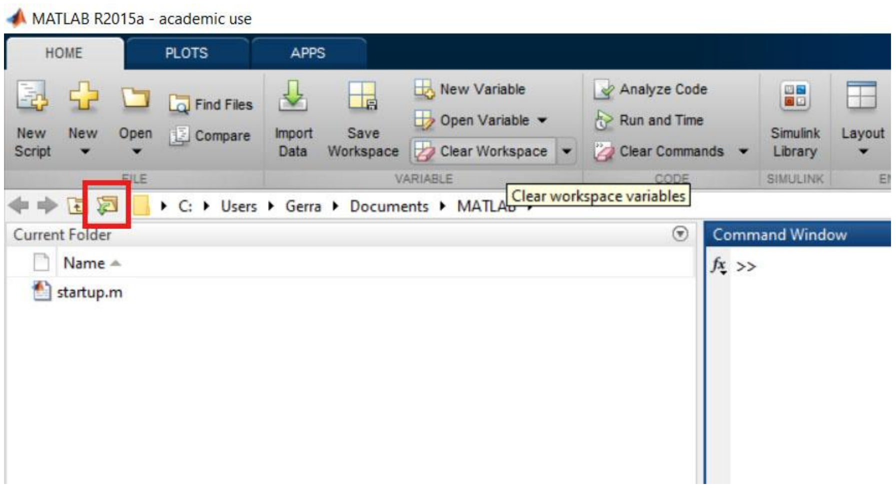
Figure 2-2 Setup wizard properties

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_Olimexino.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.

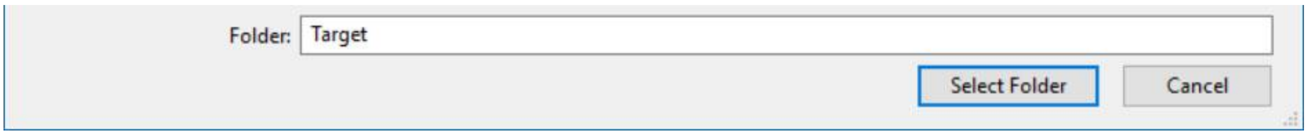


Figure 3-2 select target folder

NOTE: MATLAB versions prior to the 2013b release cannot open the .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:

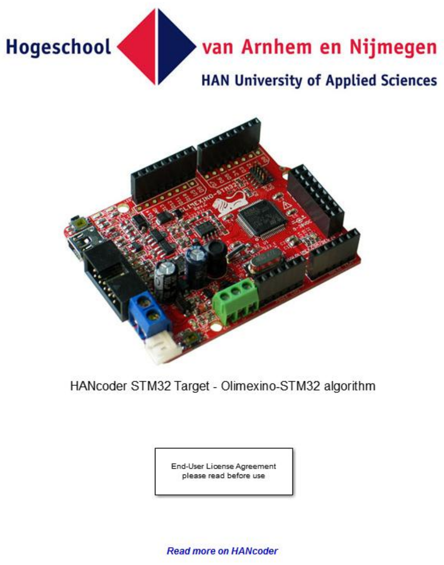

Hogeschool van Arnhem en Nijmegen
HAN University of Applied Sciences
HANcoder STM32 Target - Olimexino-STM32 algorithm
End-User License Agreement please read before use
Read more on HANcoder

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:

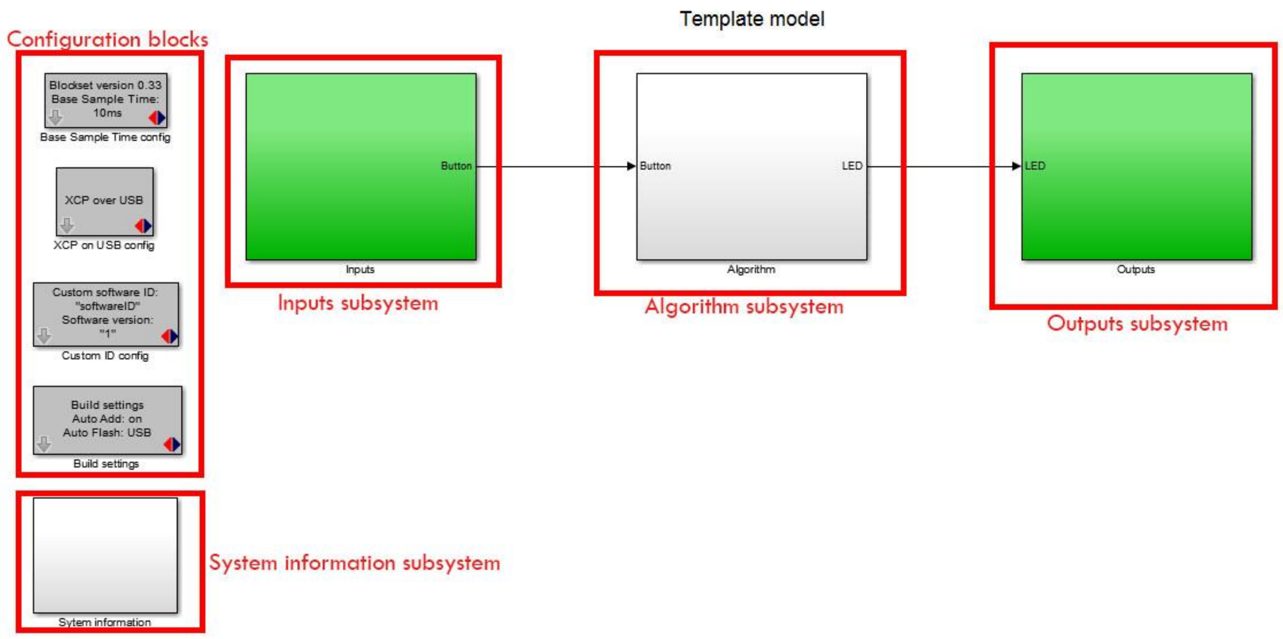


Figure 3-4 Template model delivered with blockset

Figure 3-2 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. 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

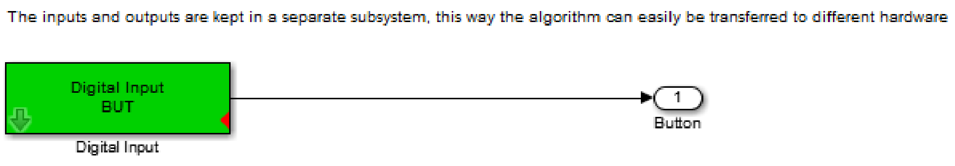

The inputs and outputs are kept in a separate subsystem, this way the algorithm can easily be transferred to different hardware
Digital Input
BUT
Digital Input
1
Button

Figure 3-5 Inside 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

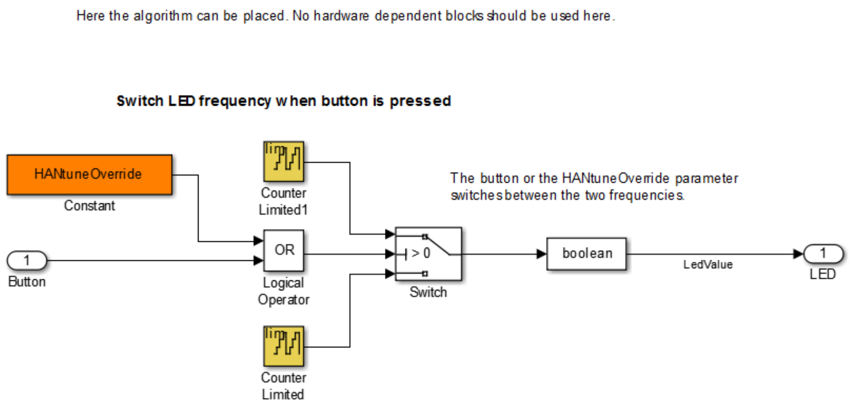

Here the algorithm can be placed. No hardware dependent blocks should be used here.
Switch LED frequency when button is pressed
HANtuneOverride
Constant
Counter
Limited1
The button or the HANtuneOverride parameter switches between the two frequencies
1
Button
OR
Logical
Operator
> 0
Switch
boolean
LedValue
1
LED
Counter
Limited

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

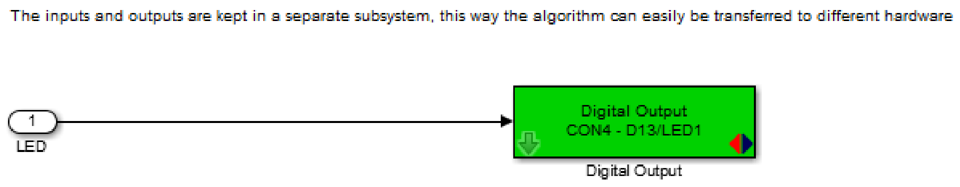

The inputs and outputs are kept in a separate subsystem, this way the algorithm can easily be transferred to different hardware
1
LED
Digital Output
CON4 - D13/LED1
Digital Output

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 model.
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 Olimexino to communicate to HANtune over USB. (More on HANtune in chapter: **Error! Reference source not found.** Error! ference source not found.)

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.

Chapter 4 explains more about flashing the program.

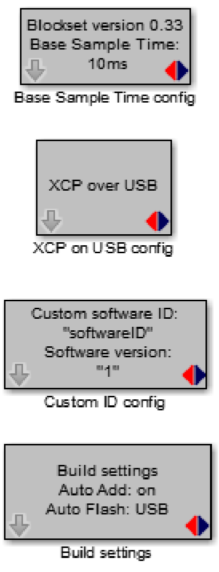


Figure 3-8 Configuration blocks template model

The System Information Subsystem

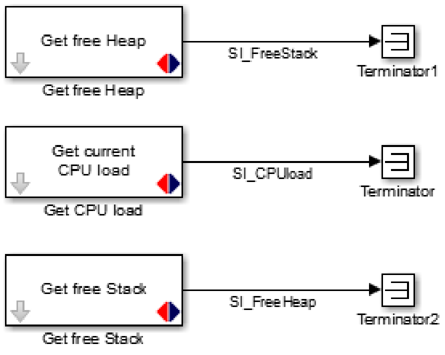


Figure 3-9 The System Information subsystem template model

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:

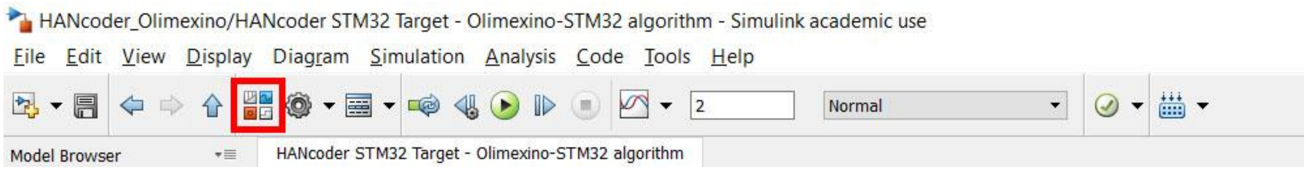

HANcoder_Olimexino/HANcoder STM32 Target - Olimexino-STM32 algorithm - Simulink academic use
File Edit View Display Diagram Simulation Analysis Code Tools Help
2
Normal
Model Browser
HANcoder STM32 Target - Olimexino-STM32 algorithm

Figure 3-10 Simulink Library button

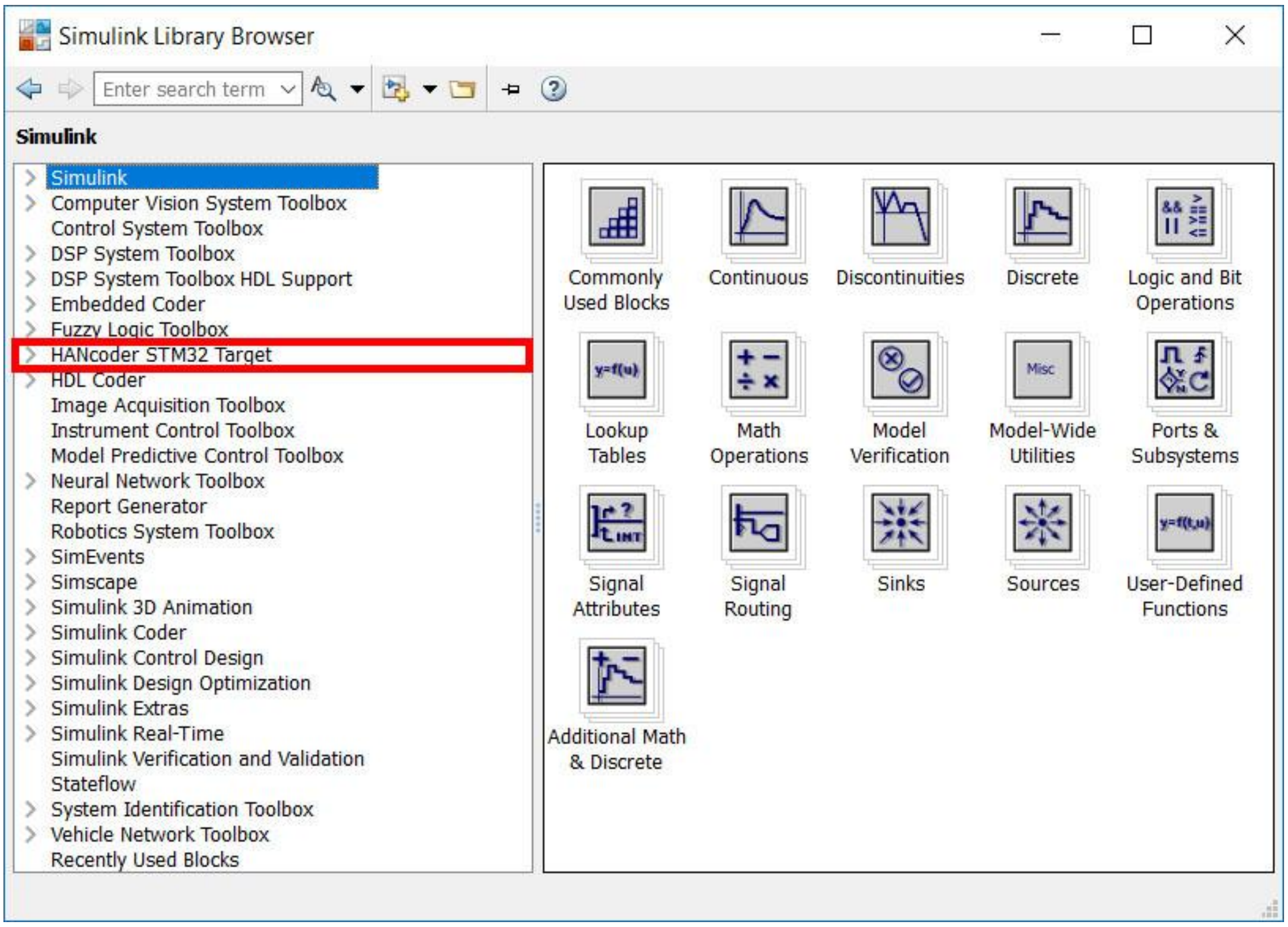


Figure 3-11 Simulink library

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. Turn on ‘Generate repositories in memory’ in the pop up.

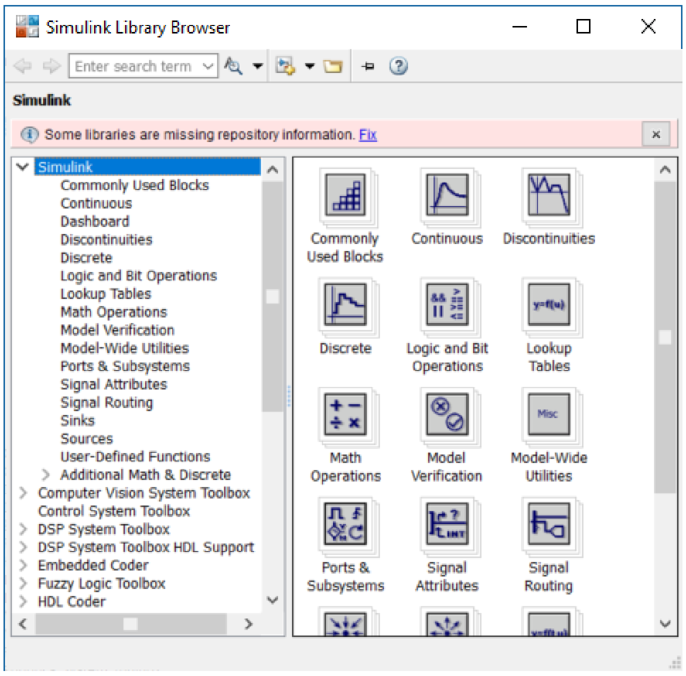
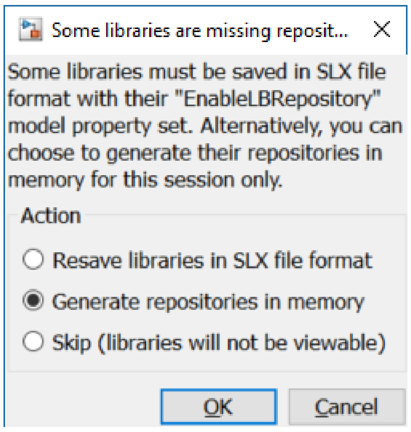


Figure 3-12 Library pop up


Some libraries are missing reposit...

Some libraries must be saved in SLX file format with their "EnableLBRepository" model property set. Alternatively, you can choose to generate their repositories in memory for this session only.

Action

- ○ Resave libraries in SLX file format
- [✓ Generate repositories in memory]
- ○ Skip (libraries will not be viewable)

Figure 3-13 library Fix pop up

When the HANcoder STM32 Target expands the HANcoder blocks for the compatible STM32 development boards are visible.

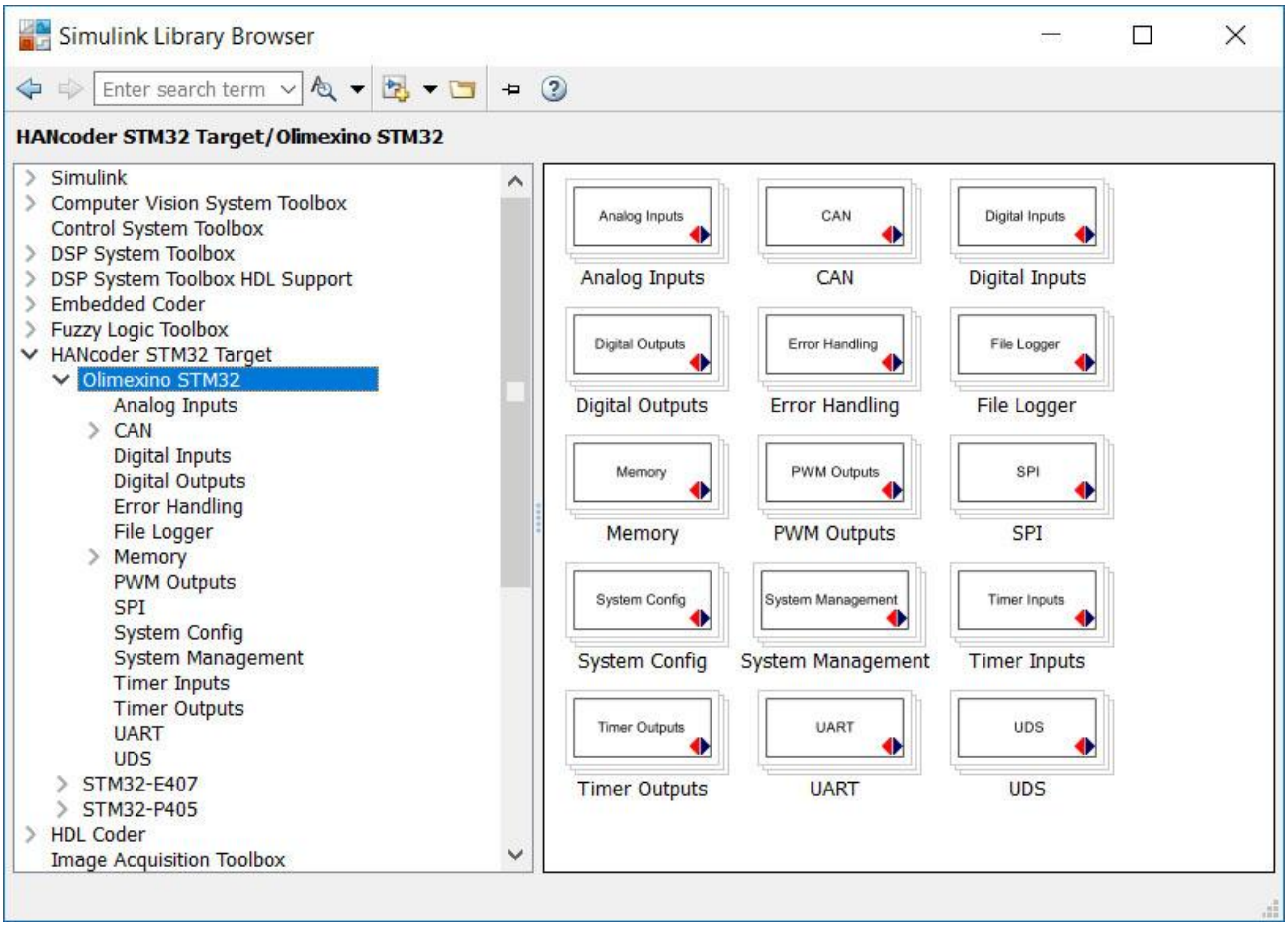
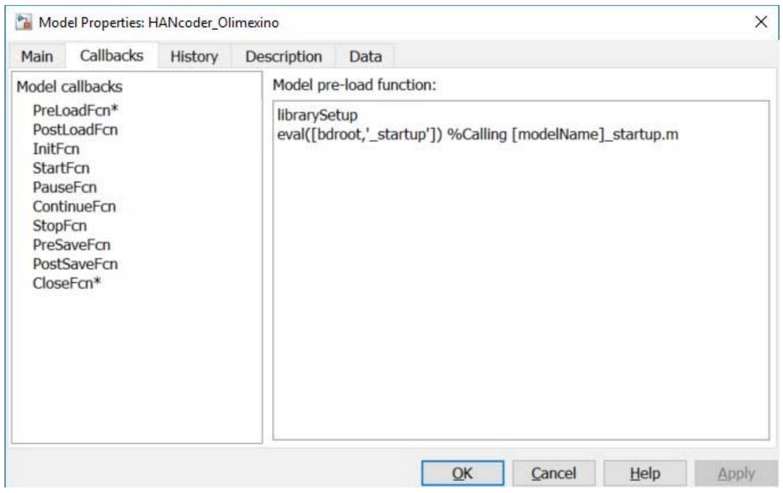


Figure 3-14 Simulink HANcoder STM32 Target Library

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".

3.3Working 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.



```
librarySetup
eval([bdroot,'_startup']) %Calling [modelName]_startup.m
```

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_Olimexino_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_Olimexino_startup.m is shown below:

```
% This m-file will be run automatically when starting model 'HANcoder_Olimexino'
% 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
```

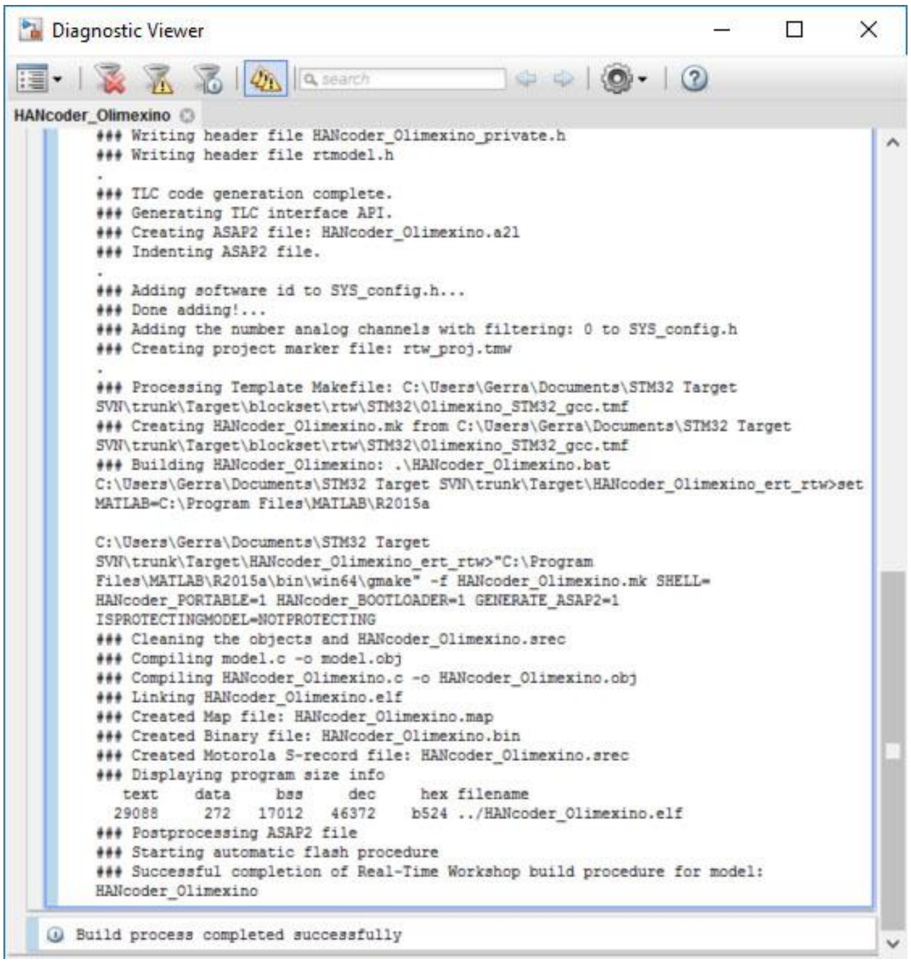
```
SI_FreeStack = Simulink.Signal;
SI_FreeStack.StorageClass = 'ExportedGlobal';
SI_CPUload = Simulink.Signal;
SI_CPUload.StorageClass = 'ExportedGlobal';
SI_FreeHeap = Simulink.Signal;
SI_FreeHeap.StorageClass = 'ExportedGlobal';

%% 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.



```
### Writing header file HANcoder_Olimexino_private.h
### Writing header file rtmodel.h
.
### TLC code generation complete.
### Generating TLC interface API.
### Creating ASAP2 file: HANcoder_Olimexino.a2l
### Indenting ASAP2 file.
.
### Adding software id to SYS_config.h...
### Done adding!...
### Adding the number analog channels with filtering: 0 to SYS_config.h
### Creating project marker file: rtw_proj.tmw
.
### Processing Template Makefile: C:\Users\Gerra\Documents\STM32 Target
SVN\trunk\Target\blockset\rtw\STM32\Olimexino_STM32_gcc.tmf
### Creating HANcoder_Olimexino.mk from C:\Users\Gerra\Documents\STM32 Target
SVN\trunk\Target\blockset\rtw\STM32\Olimexino_STM32_gcc.tmf
### Building HANcoder_Olimexino: .\HANcoder_Olimexino.bat
C:\Users\Gerra\Documents\STM32 Target SVN\trunk\Target\HANcoder_Olimexino_ert_rtw>set
MATLAB=C:\Program Files\MATLAB\R2015a

C:\Users\Gerra\Documents\STM32 Target
SVN\trunk\Target\HANcoder_Olimexino_ert_rtw>"C:\Program
Files\MATLAB\R2015a\bin\win64\gmake" -f HANcoder_Olimexino.mk SHELL=
HANcoder_PORTABLE=1 HANcoder_BOOTLOADER=1 GENERATE_ASAP2=1
ISPROTECTINGMODEL=NOTPROTECTING
### Cleaning the objects and HANcoder_Olimexino.srec
### Compiling model.c -o model.obj
### Compiling HANcoder_Olimexino.c -o HANcoder_Olimexino.obj
### Linking HANcoder_Olimexino.elf
### Created Map file: HANcoder_Olimexino.map
### Created Binary file: HANcoder_Olimexino.bin
### Created Motorola S-record file: HANcoder_Olimexino.srec
### Displaying program size info
   text    data     bss     dec     hex filename
  29088     272   17012   46372    b524 ../HANcoder_Olimexino.elf
### Postprocessing ASAP2 file
### Starting automatic flash procedure
### Successful completion of Real-Time Workshop build procedure for model:
HANcoder_Olimexino
```

```
Build process completed successfully
```

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 see The Troubleshooting Guide if there are any errors.

4 FLASHING THE CONTROLLER WITH THE BOOTLOADER

This chapter describes the steps that have to be taken to flash the controller by CAN or USB. This is only possible in combination with an Olimexino that has the correct bootloader. To flash the bootloader onto the Olimexino please follow the steps in chapter 5.3 Flashing the controller with the ST-Link

4.1 Flashing the controller via USB

It is possible to flash the Olimexino via the CANbus or via an USB connection. This paragraph describes how to flash the Olimexino 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 Olimexino board.
- The Olimexino has the correct bootloader on it

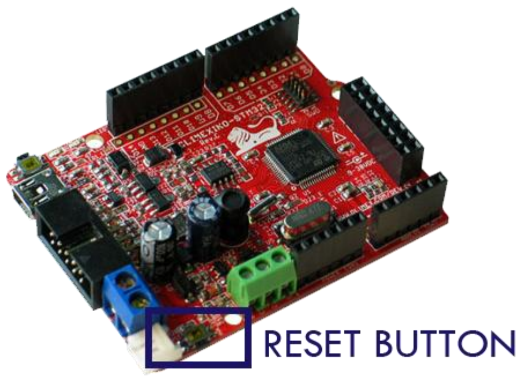


Figure 4-1 RESET button Olimexino

4.1.1 Install bootloader driver

To install the driver the program should be kept in bootloader mode. This is achieved by having D2 pulled down to ground with at least a 150 Ohm resistor(max 820 Ohm), while resetting the microcontroller 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:

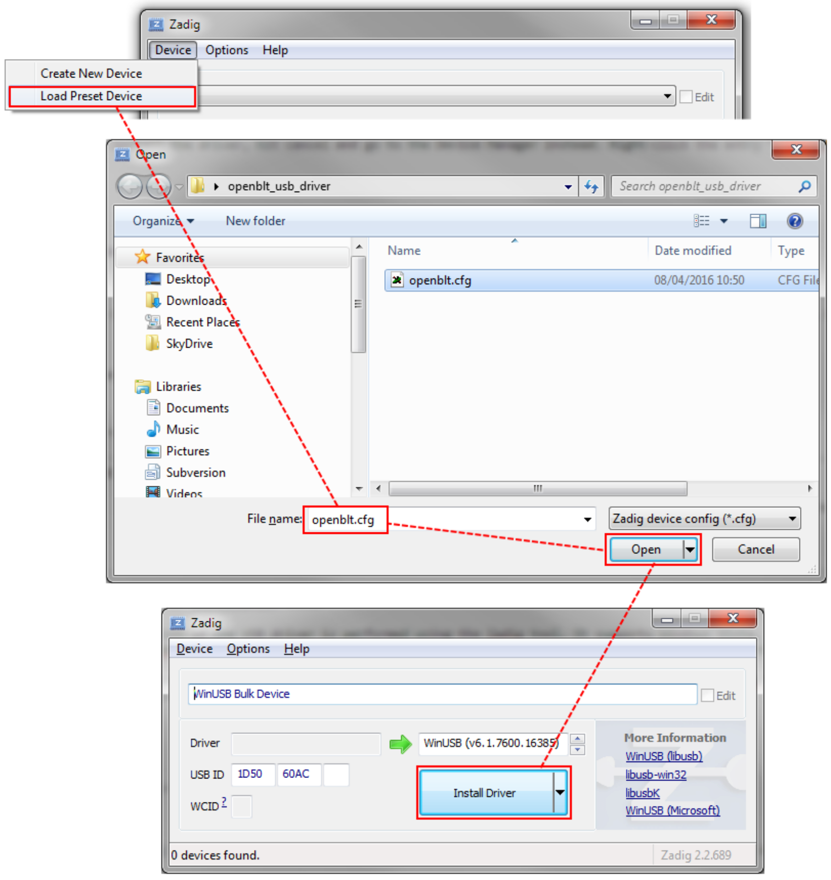


Figure 4-2 Feaser Open BLT explanation figure 1

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:

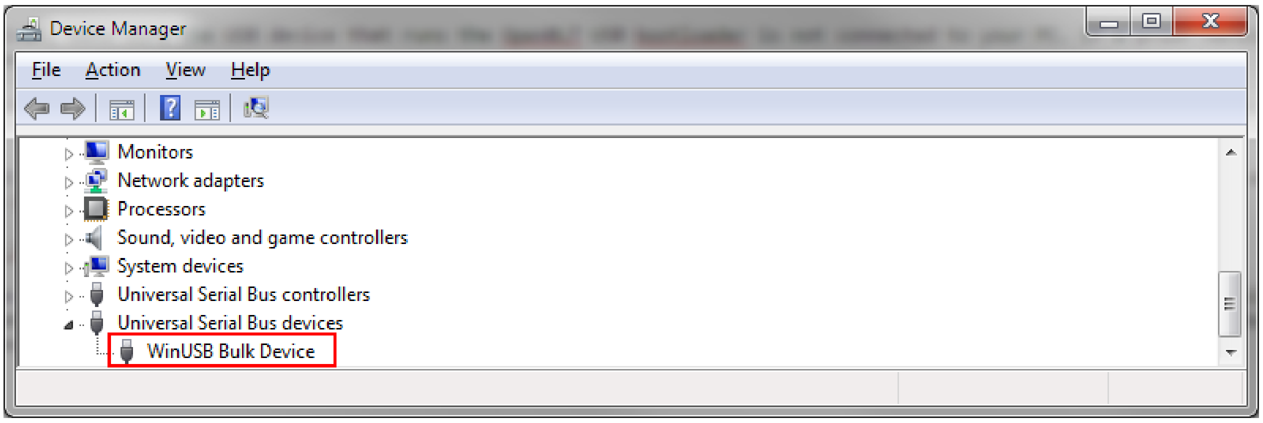


Figure 4-3 Feaser Open BLT explanation figure 2

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

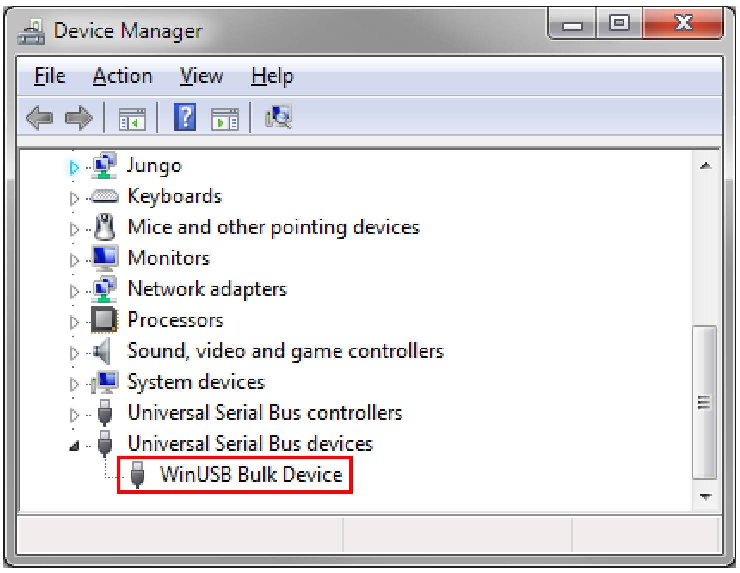

Device Manager
File Action View Help
Jungo
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Processors
Sound, video and game controllers
System devices
Universal Serial Bus controllers
Universal Serial Bus devices
WinUSB Bulk Device

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_Olimexino.srec'.

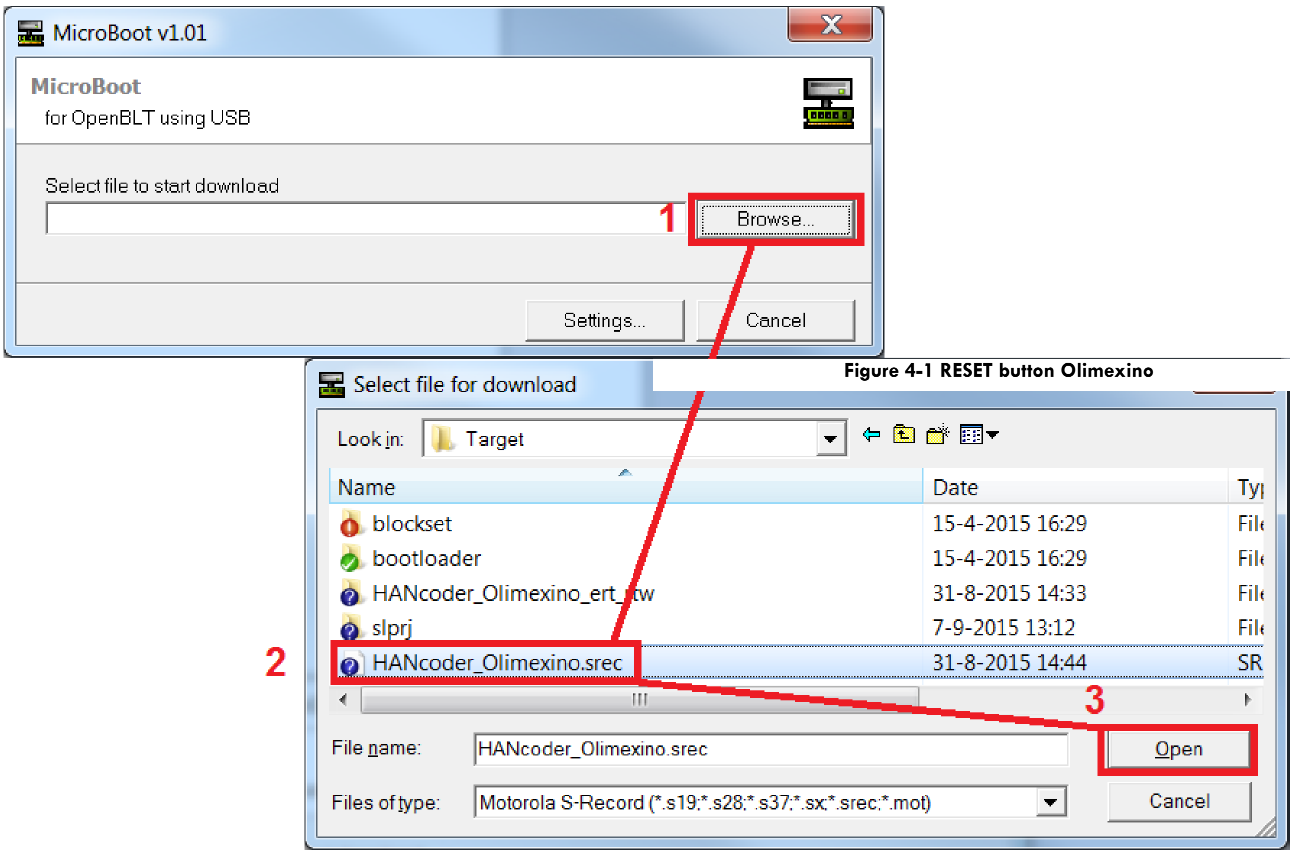


Figure 4-1 RESET button Olimexino

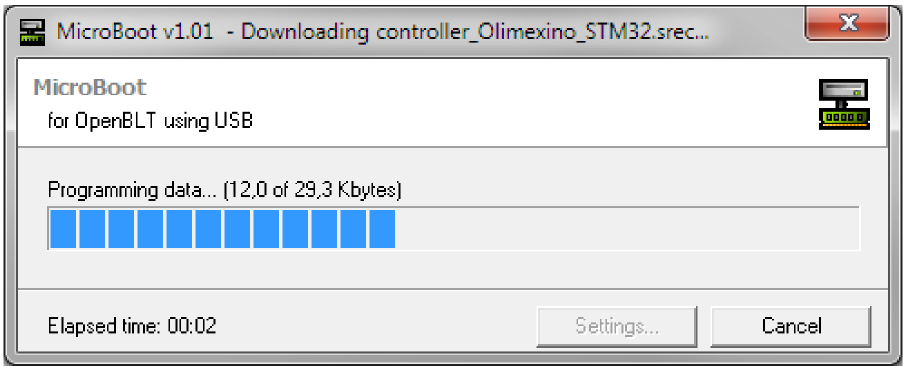
Name	Date	Ty
blockset	15-4-2015 16:29	File
bootloader	15-4-2015 16:29	File
HANcoder_Olimexino_ert_rtw	31-8-2015 14:33	File
slprj	7-9-2015 13:12	File
HANcoder_Olimexino.srec	31-8-2015 14:44	SR

File name: HANcoder_Olimexino.srec

Files of type: Motorola S-Record (*.s19;*.s28;*.s37;*.sx;*.srec;*.mot)

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.


MicroBoot v1.01 - Downloading controller_Olimexino_STM32.srec...

MicroBoot
for OpenBLT using USB

Programming data... (12.0 of 29.3 Kbytes)

Elapsed time: 00:02

Figure 4-6 Bootloader progress

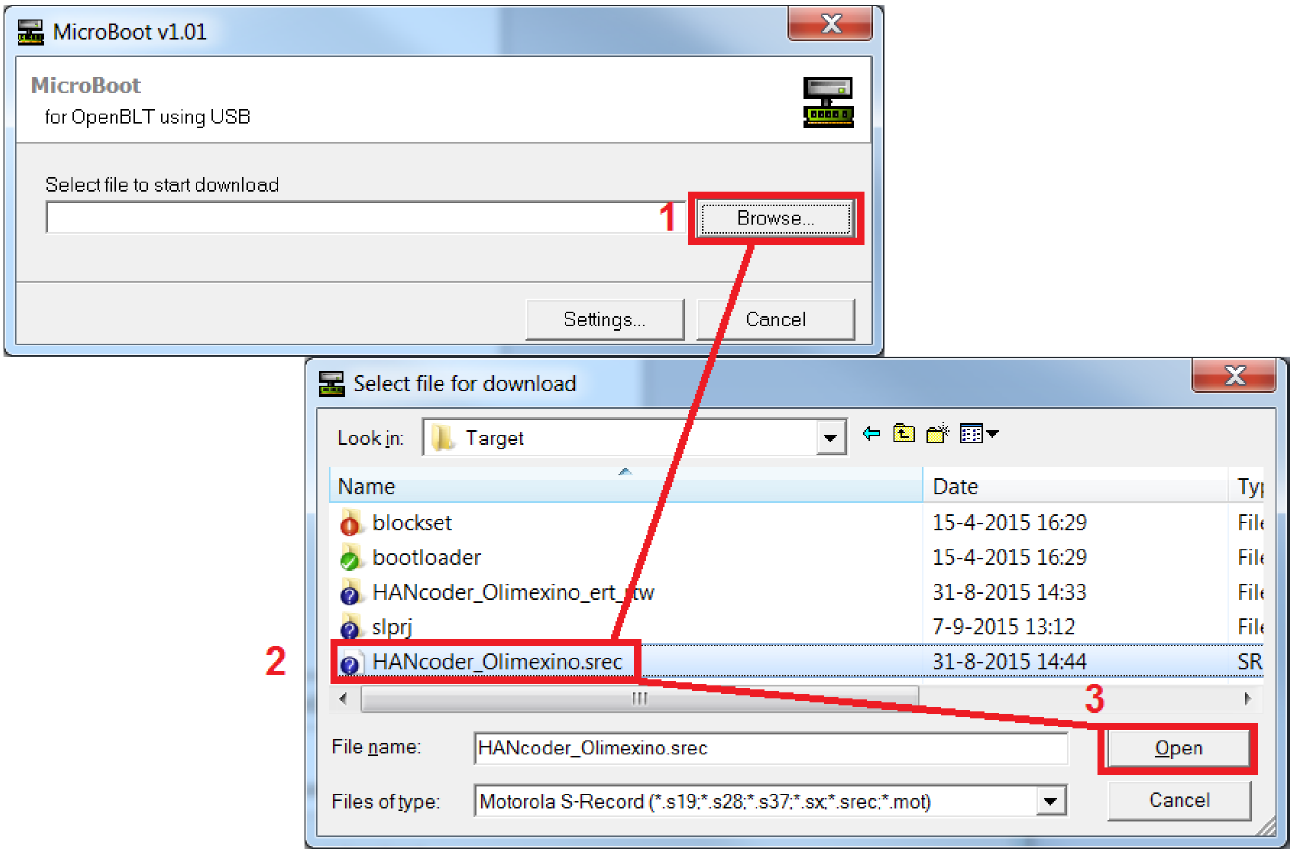
4.2Flashing the controller via CAN

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

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 CAN bus.
- The Peak PCAN interface (installation in 1Appendix 4) is connected to the PC's USB port
- Power is supplied to the Olimexino board.
- The Olimexino has the correct bootloader on it

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



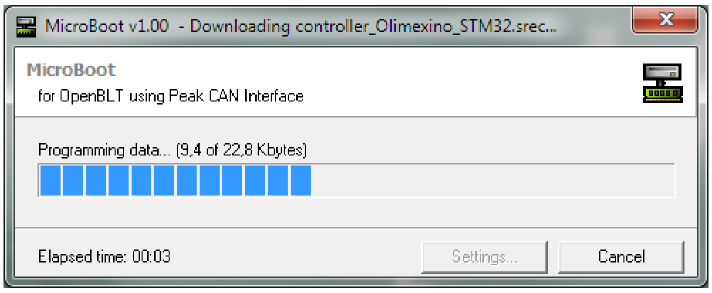
Name	Date	Typ
blockset	15-4-2015 16:29	File
bootloader	15-4-2015 16:29	File
HANcoder_Olimexino_ert_rtw	31-8-2015 14:33	File
slprj	7-9-2015 13:12	File
HANcoder_Olimexino.srec	31-8-2015 14:44	SR

File name: HANcoder_Olimexino.srec

Files of type: Motorola S-Record (*.s19;*.s28;*.s37;*.sx;*.srec;*.mot)

Figure 4-7 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, the newly flashed software program is automatically started.


MicroBoot v1.00 - Downloading controller_Olimexino_STM32.srec...
MicroBoot
for OpenBLT using Peak CAN Interface
Settings...
Cancel

Programming data... (9,4 of 22,8 Kbytes)

Elapsed time: 00:03

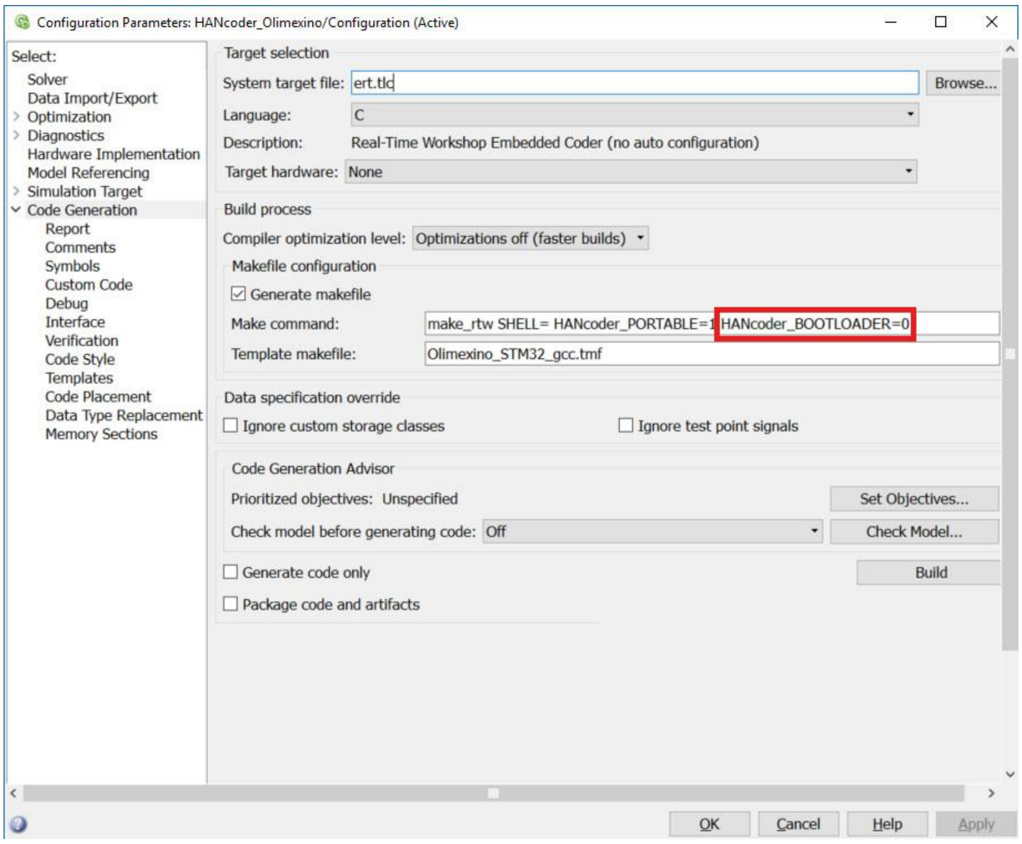
Figure 4-8 Microboot flashing progress

5 FLASHING THE CONTROLLER WITH THE ST-LINK

Downloading the software program to the flash memory of the microcontroller can be done with the help of the ST-Link. This chapter assumes the ST-Link utility is already installed. If this is not the case it can be downloaded here.

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 5. Select *Simulation -> Configuration Parameters…* from the menu.



Configuration Parameters: HANcoder_Olimexino/Configuration (Active)

Target selection
System target file: ert.tlc
Language: C
Description: Real-Time Workshop Embedded Coder (no auto configuration)
Target hardware: None

Build process
Compiler optimization level: Optimizations off (faster builds)
Makefile configuration
☑ Generate makefile
Make command: make_rtw SHELL= HANcoder_PORTABLE=1 HANcoder_BOOTLOADER=0
Template makefile: Olimexino_STM32_gcc.tmf

Data specification override
☐ Ignore custom storage classes
☐ Ignore test point signals

Code Generation Advisor
Prioritized objectives: Unspecified
Check model before generating code: Off

☐ Generate code only
☐ Package code and artifacts

Figure 5-1 Configuring the model for use without the bootloader

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

Now the software program is properly configured for the use without the bootloader.

5.2Uploading the software with the ST-Link

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

- The ST-Link is connected to the Olimexino STM32 board through an ARM-JTAG-20-10 adapter from Olimex.
- The ST-Link is connected to the PC's USB port.
- Power is supplied to the Olimexino board (in figure 5-2 with usb).

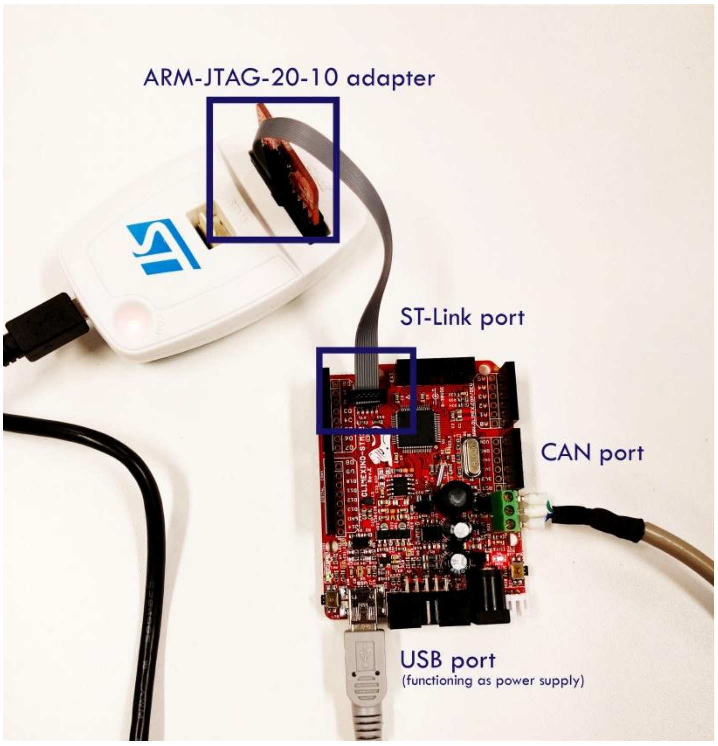
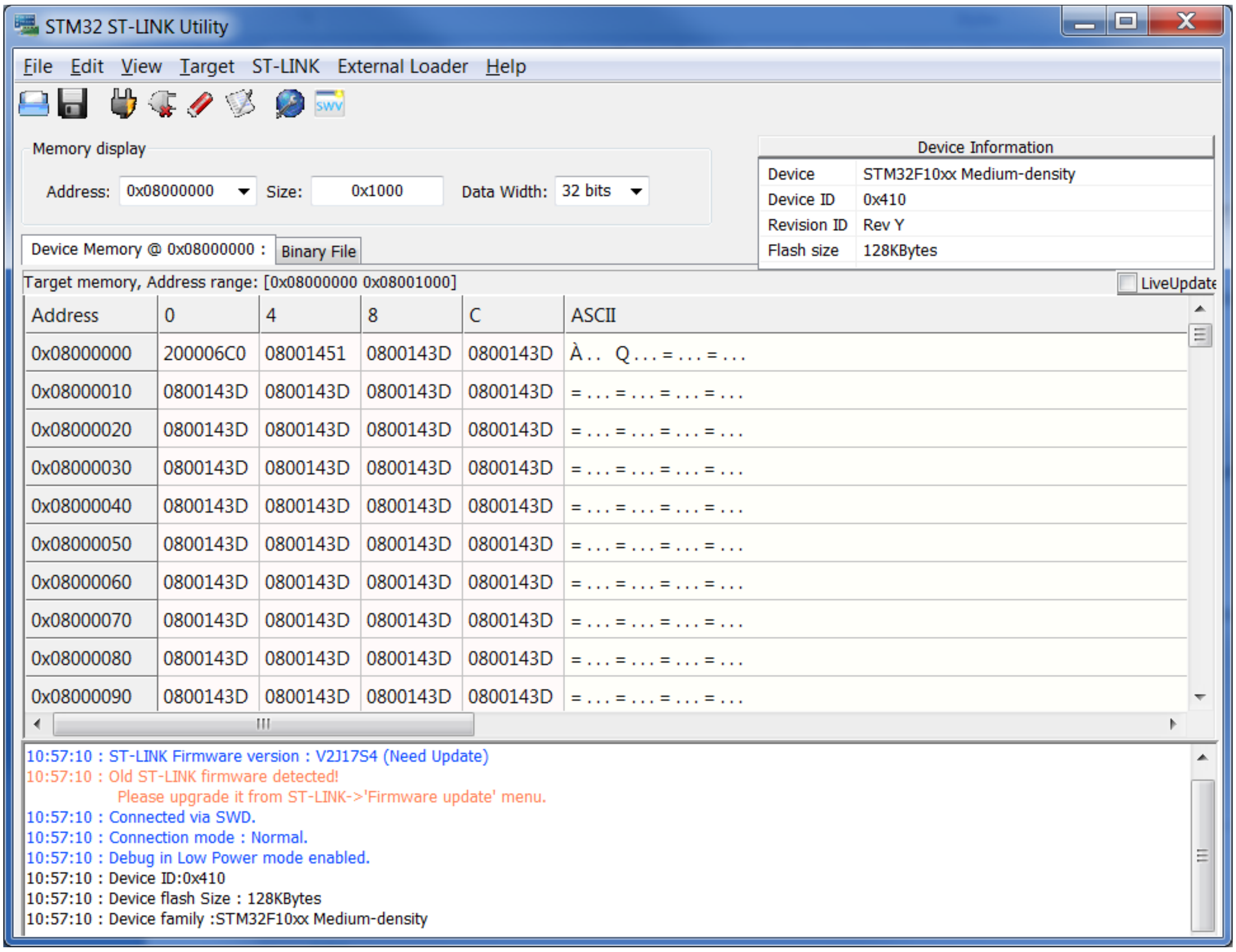


Figure 5-2 Olimexino ST-LINK setup

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:


STM32 ST-LINK Utility
File Edit View Target ST-LINK External Loader Help
Memory display
Device Memory @ 0x08000000 :
Binary File
LiveUpdate

Address: 0x08000000 Size: 0x1000 Data Width: 32 bits

Device Information	
Device	STM32F10xx Medium-density
Device ID	0x410
Revision ID	Rev Y
Flash size	128KBytes

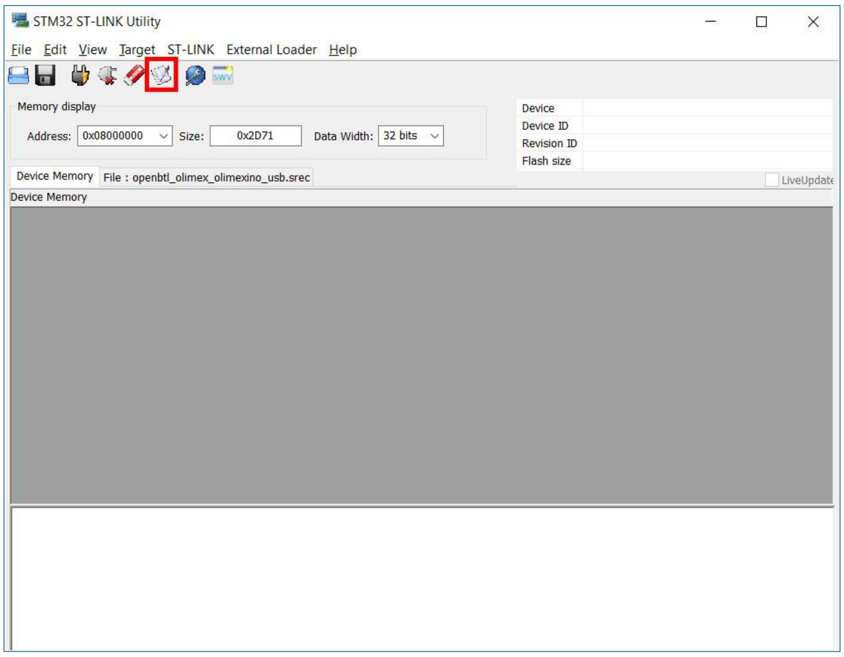
Target memory, Address range: [0x08000000 0x08001000]

Address	0	4	8	C	ASCII
0x08000000	200006C0	08001451	0800143D	0800143D	À . . Q . . . = . . . = . . .
0x08000010	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .
0x08000020	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .
0x08000030	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .
0x08000040	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .
0x08000050	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .
0x08000060	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .
0x08000070	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .
0x08000080	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .
0x08000090	0800143D	0800143D	0800143D	0800143D	= . . . = . . . = . . . = . . .

```
10:57:10 : ST-LINK Firmware version : V2J17S4 (Need Update)
10:57:10 : Old ST-LINK firmware detected!
           Please upgrade it from ST-LINK->'Firmware update' menu.
10:57:10 : Connected via SWD.
10:57:10 : Connection mode : Normal.
10:57:10 : Debug in Low Power mode enabled.
10:57:10 : Device ID:0x410
10:57:10 : Device flash Size : 128KBytes
10:57:10 : Device family :STM32F10xx Medium-density
```

Figure 5-3 ST-Link Utility

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

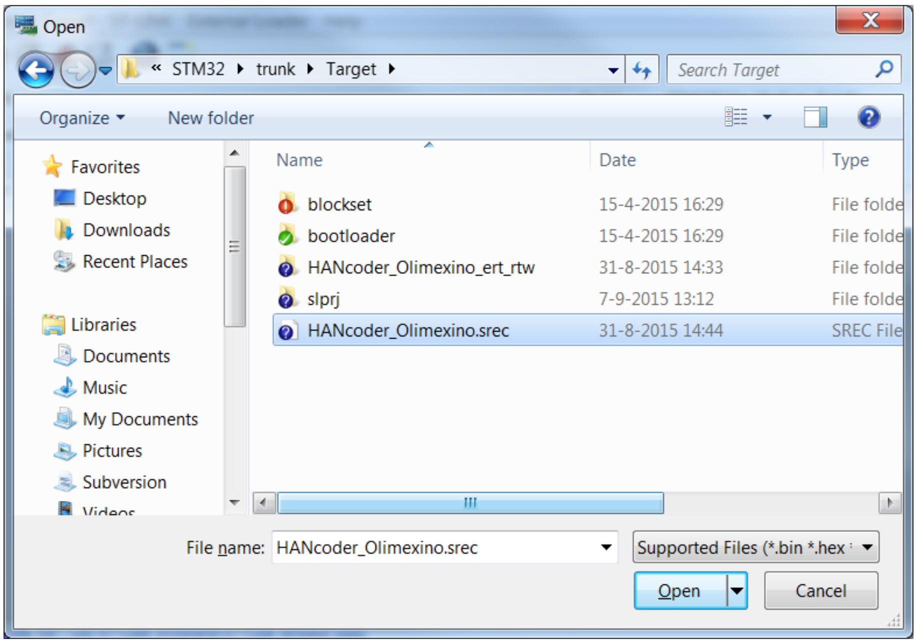


Address: 0x08000000 Size: 0x2D71 Data Width: 32 bits

File : openbtl_olimex_olimexino_usb.srec

Figure 5-4 Program and Verify

Click on Browse and choose the .srec file you want to flash to the controller, located in the same folder as the Simulink model:



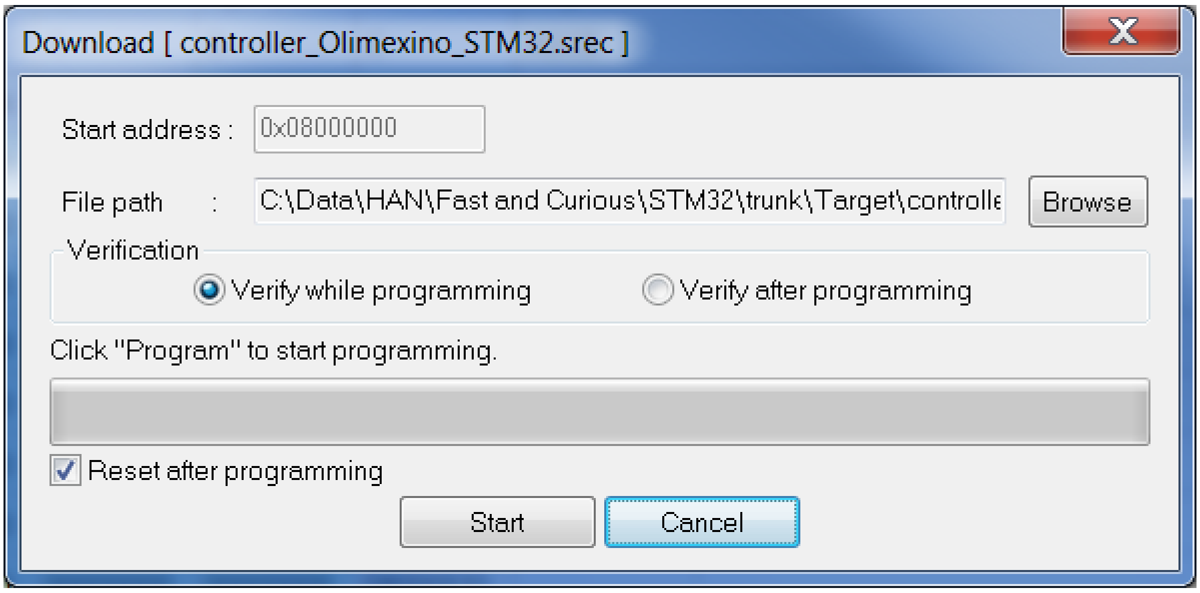
Name	Date	Type
blockset	15-4-2015 16:29	File folde
bootloader	15-4-2015 16:29	File folde
HANcoder_Olimexino_ert_rtw	31-8-2015 14:33	File folde
slprj	7-9-2015 13:12	File folde
HANcoder_Olimexino.srec	31-8-2015 14:44	SREC File

File name: HANcoder_Olimexino.srec

Supported Files (*.bin *.hex

Figure 5-5 Select srec

The following screen will appear:


Download [controller_Olimexino_STM32.srec]

Start address : 0x08000000

File path : C:\Data\HAN\Fast and Curious\STM32\trunk\Target\controlle

Verification: [✓ Verify while programming] Verify after programming

Click "Program" to start programming.

[✓] Reset after programming

Figure 5-6 MircoBoot

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
```

5.3Flashing the bootloader

The bootloader can be flashed onto the Olimexino in exactly the same way as a program from Simulink. The bootloader files can be found in: '\Target\bootloader\Demo\ARMCM3_STM32_Olimex_Olimexino_GCC\Boot\bin' use the openbtl_olimex_olimexino_can.srec for the CAN bootloader and the openbtl_olimex_olimexino_usb.srec for the USB bootloader.

NOTE!: Make sure that after the bootloader is flashed on the Olimexino, the ST-LINK is disconnected from the development board! Otherwise the Olimexino will expect a bootloader every time other programs want to flash on the dev board.

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 Olimexino. 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 Olimexino. When the Olimexino is connected the device manager will show it as STM32 Virtual COM port. This is not the right driver.

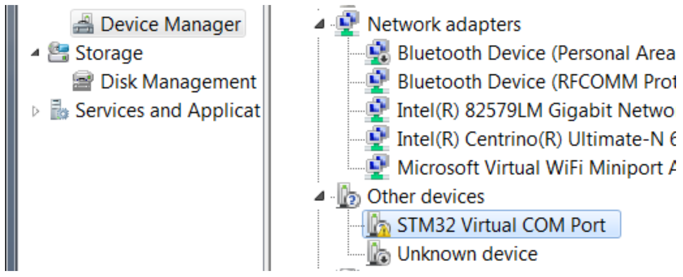

Device Manager
Storage
Disk Management
Services and Applicat
Network adapters
Bluetooth Device (Personal Area
Bluetooth Device (RFCOMM Prot
Intel(R) 82579LM Gigabit Netwo
Intel(R) Centrino(R) Ultimate-N 6
Microsoft Virtual WiFi Miniport A
Other devices
STM32 Virtual COM Port
Unknown device

Figure 6-1 Device Manager COM port

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 Olimexino as STMicroelectronics Virtual COM Port, see below:

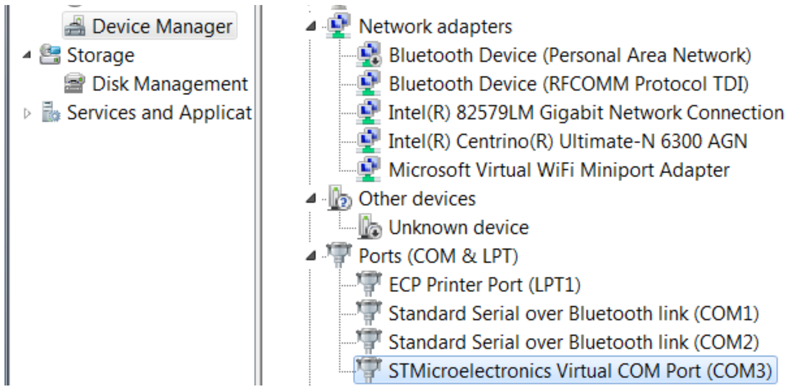

Device Manager
Storage
Disk Management
Services and Applicat
Network adapters
Bluetooth Device (Personal Area Network)
Bluetooth Device (RFCOMM Protocol TDI)
Intel(R) 82579LM Gigabit Network Connection
Intel(R) Centrino(R) Ultimate-N 6300 AGN
Microsoft Virtual WiFi Miniport Adapter
Other devices
Unknown device
Ports (COM & LPT)
ECP Printer Port (LPT1)
Standard Serial over Bluetooth link (COM1)
Standard Serial over Bluetooth link (COM2)
STMicroelectronics Virtual COM Port (COM3)

Figure 6-2 Device Manager virtual COM port installed

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

Note: When the Olimexino 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 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'. The .a2l file is located in the same directory as the model.

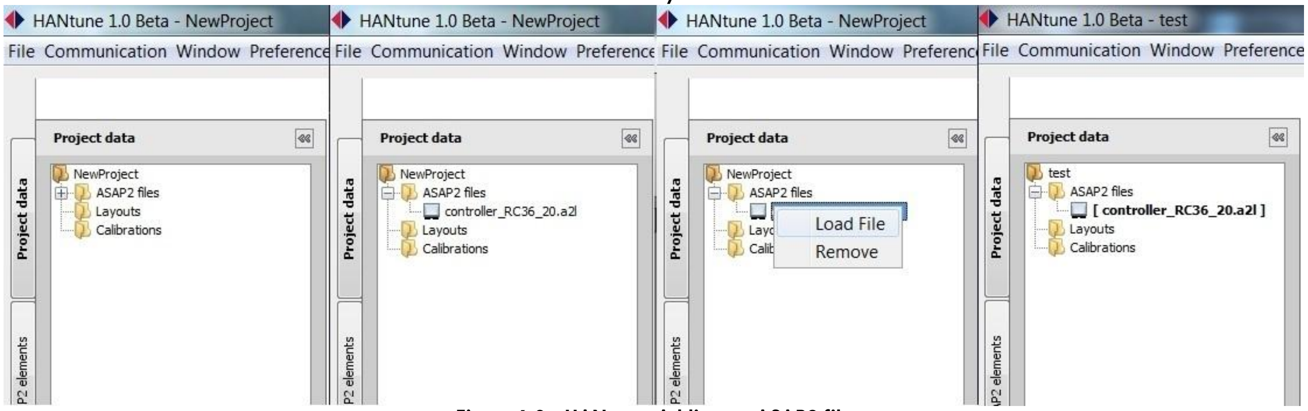


Figure 6-3 - HANtune, Adding an ASAP2 file

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.

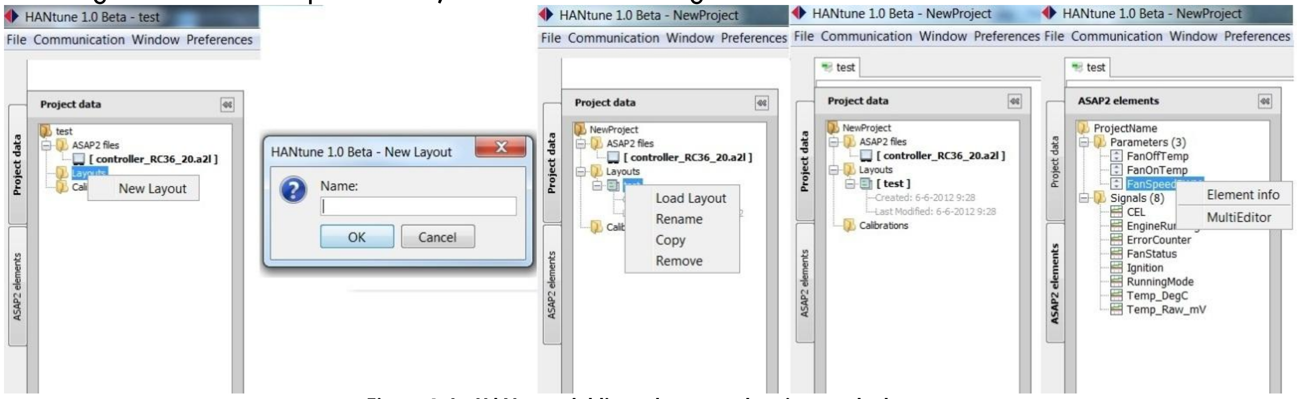


Figure 6-4 - HANtune, Adding a layout and an item to the layout

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 6-8 - HANtune Communication status, to know if you are using too many signals or not.

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

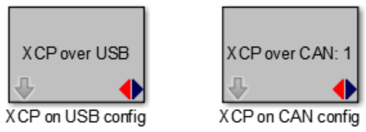


Figure 6-5 – Olimexino XCP config blocks

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.

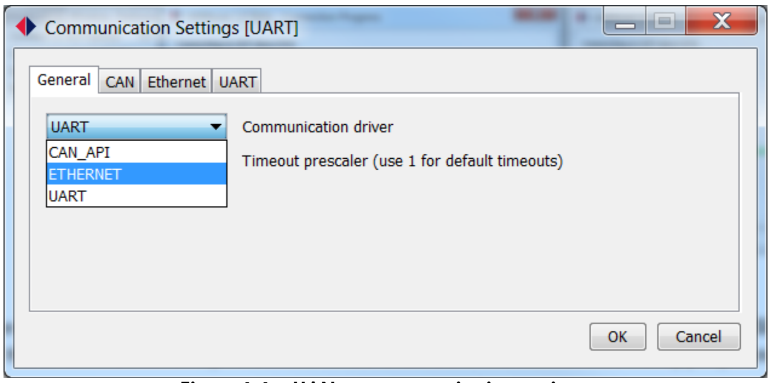

Communication Settings [UART]
General
CAN
Ethernet
UART
UART
CAN_API
ETHERNET
UART
Communication driver
Timeout prescaler (use 1 for default timeouts)
OK
Cancel

Figure 6-6 – 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.

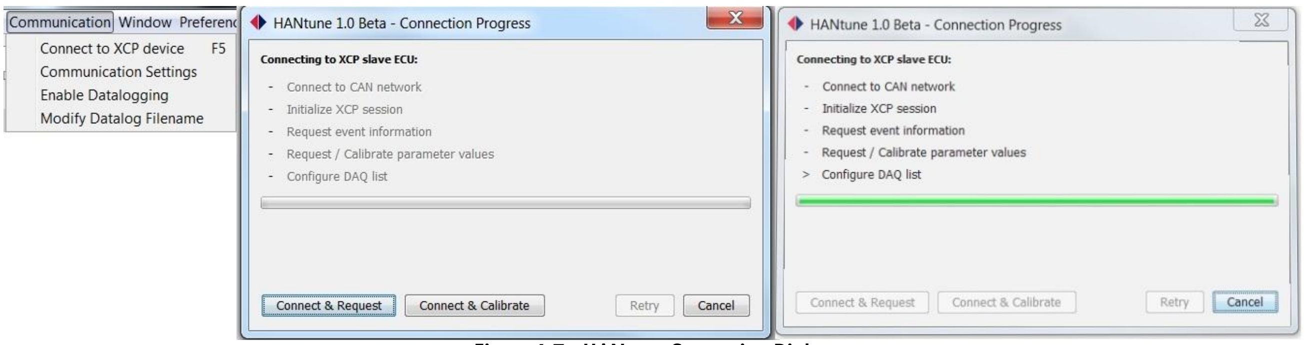


Figure 6-7 - HANtune Connection Dialog

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.

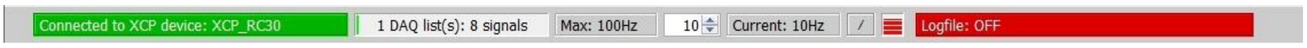


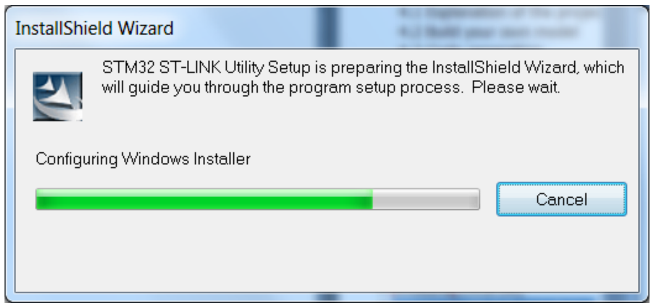
Figure 6-8 - HANtune Communication status

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.



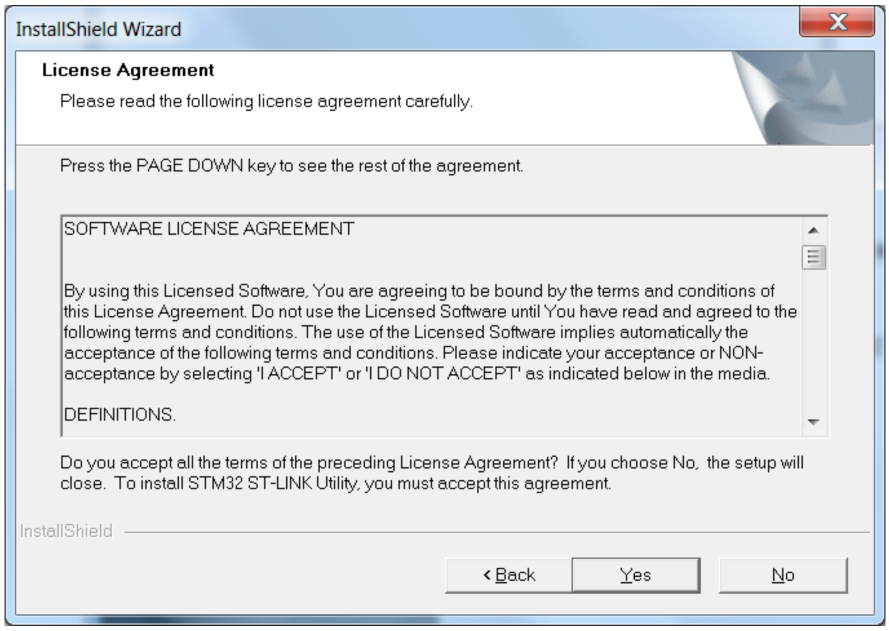
InstallShield Wizard

STM32 ST-LINK Utility Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.

Configuring Windows Installer

Figure 1-1 Installshield

Click next until the license agreement screen appears:



InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

Press the PAGE DOWN key to see the rest of the agreement.

SOFTWARE LICENSE AGREEMENT

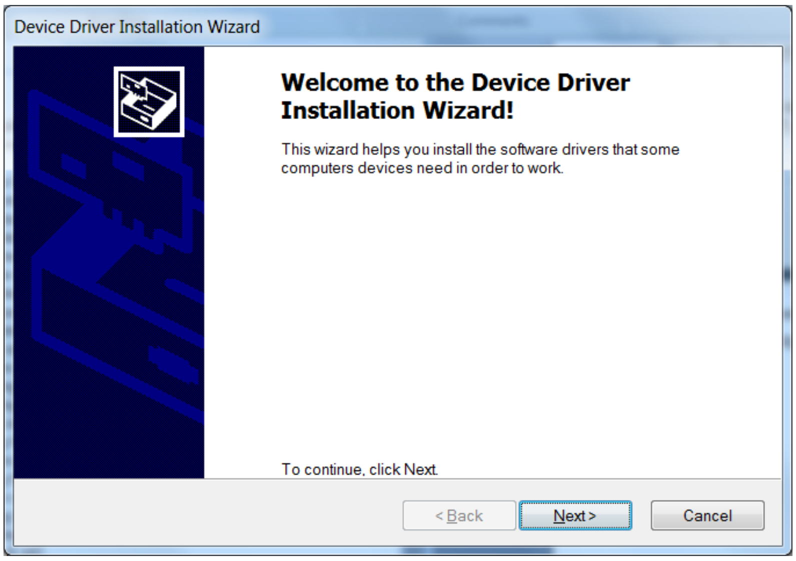
By using this Licensed Software, You are agreeing to be bound by the terms and conditions of this License Agreement. Do not use the Licensed Software until You have read and agreed to the following terms and conditions. The use of the Licensed Software implies automatically the acceptance of the following terms and conditions. Please indicate your acceptance or NON-acceptance by selecting 'I ACCEPT' or 'I DO NOT ACCEPT' as indicated below in the media.

DEFINITIONS.

Do you accept all the terms of the preceding License Agreement? If you choose No, the setup will close. To install STM32 ST-LINK Utility, you must accept this agreement.

Figure 1-2 License agreement

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.



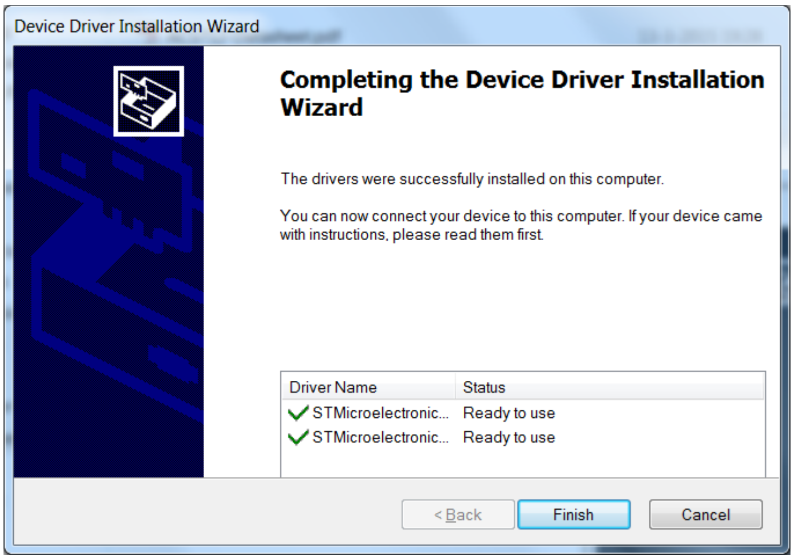
Welcome to the Device Driver Installation Wizard!

This wizard helps you install the software drivers that some computers devices need in order to work.

To continue, click Next.

Figure 1-3 Installation wizard

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



Completing the Device Driver Installation Wizard

The drivers were successfully installed on this computer.

You can now connect your device to this computer. If your device came with instructions, please read them first.

Driver Name	Status
✓ STMicroelectronic...	Ready to use
✓ STMicroelectronic...	Ready to use

Figure 1-4 Completed installation

APPENDIX 2. SEGGER J-LINK INSTALLATION

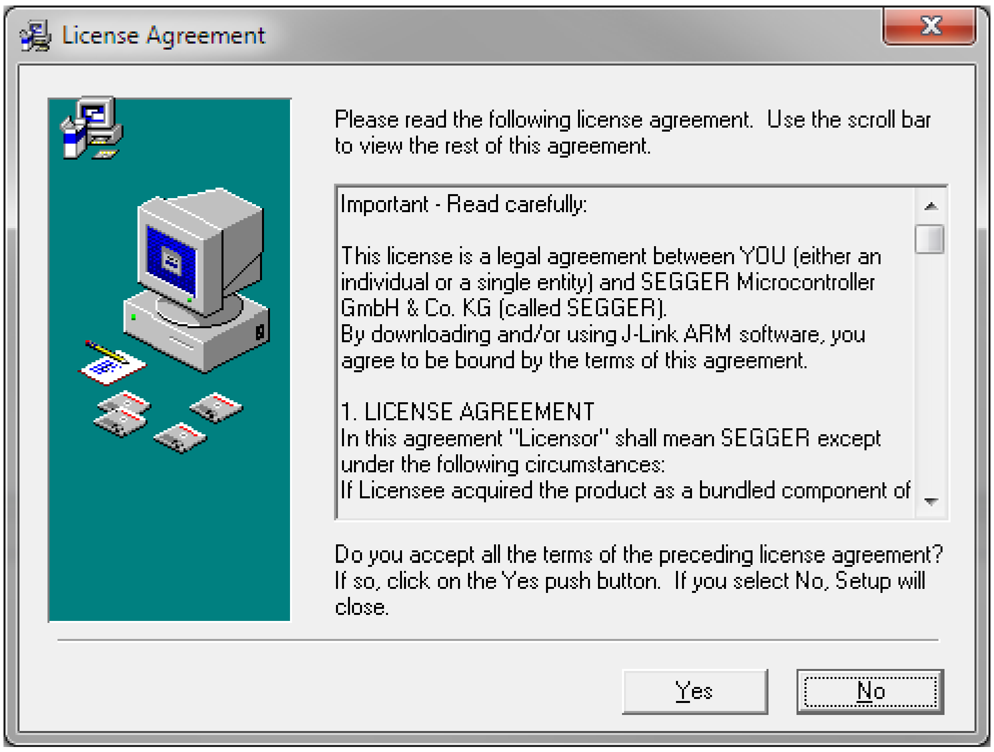
This part is only necessary if you do not possess an Olimexino with the correct bootloader or if you don't want to flash a program via CAN. For downloading a program to the flash memory of the microcontroller, the Segger J-Link interface is required:

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'.


License Agreement

Please read the following license agreement. Use the scroll bar to view the rest of this agreement.

Important - Read carefully:

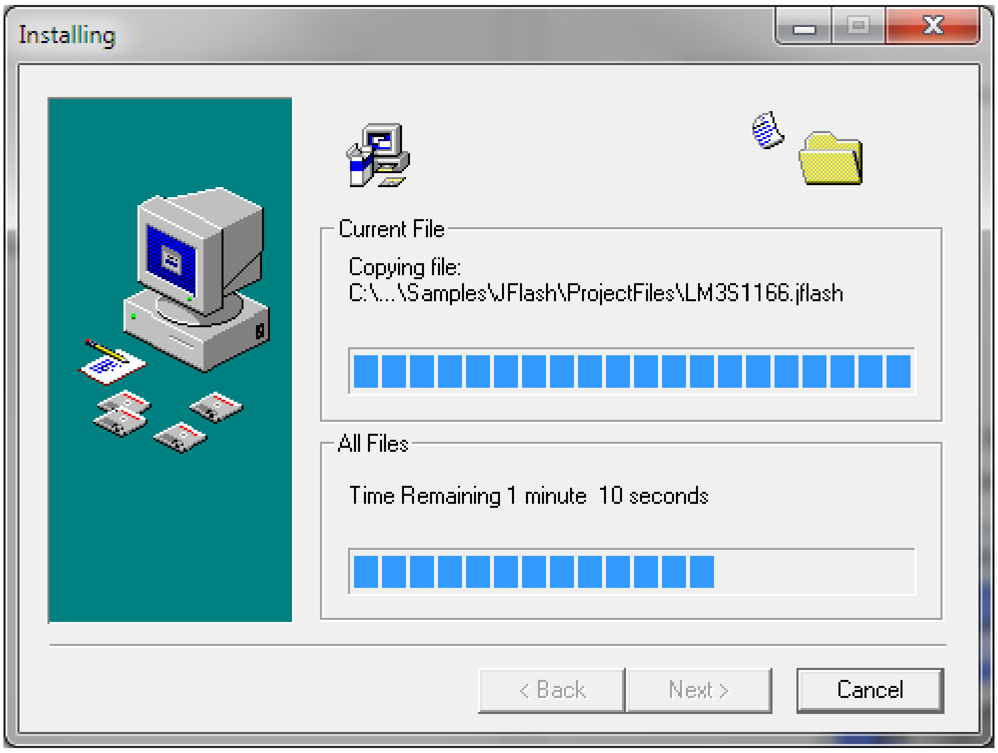
This license is a legal agreement between YOU (either an individual or a single entity) and SEGGER Microcontroller GmbH & Co. KG (called SEGGER).
By downloading and/or using J-Link ARM software, you agree to be bound by the terms of this agreement.

1. LICENSE AGREEMENT
In this agreement "Licensor" shall mean SEGGER except under the following circumstances:
If Licensee acquired the product as a bundled component of

Do you accept all the terms of the preceding license agreement? If so, click on the Yes push button. If you select No, Setup will close.

Figure 2-2 Accept license agreement

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

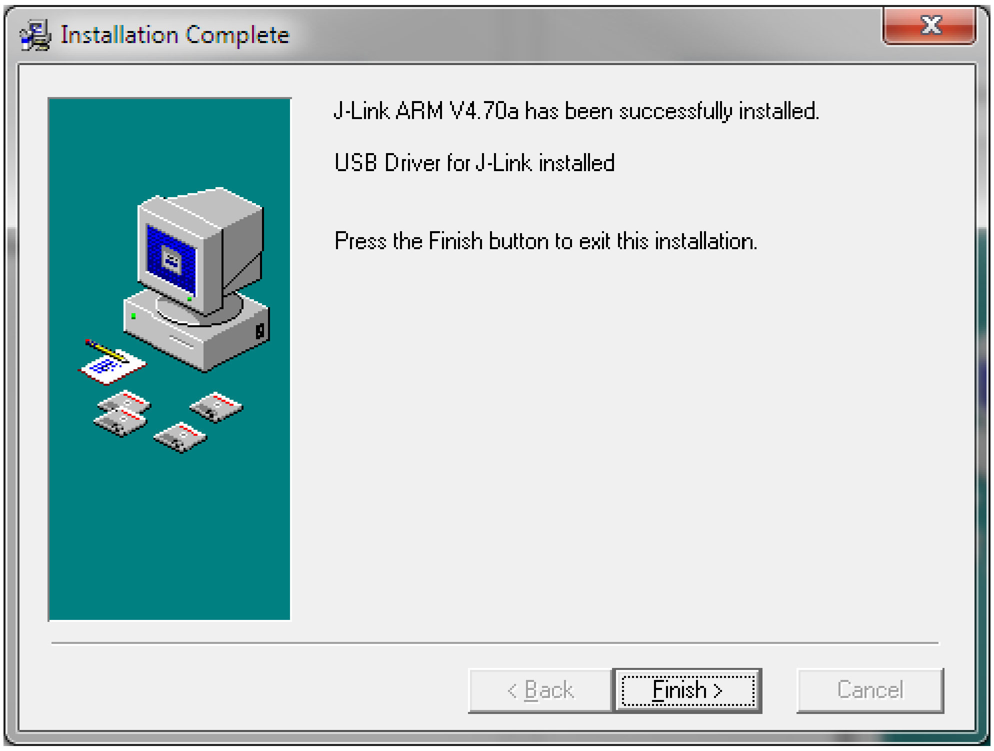


Copying file:
C:\...\Samples\JFlash\ProjectFiles\LM3S1166.jflash

Time Remaining 1 minute 10 seconds

Figure 2-3 Installation process

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



J-Link ARM V4.70a has been successfully installed.

USB Driver for J-Link installed

Press the Finish button to exit this 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 Olimexino STM32 board through an ARM-JTAG-20-10 adapter from Olimex.
- The Segger J-Link is connected to the PC's USB port.
- Power is supplied to the Olimexino board.

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

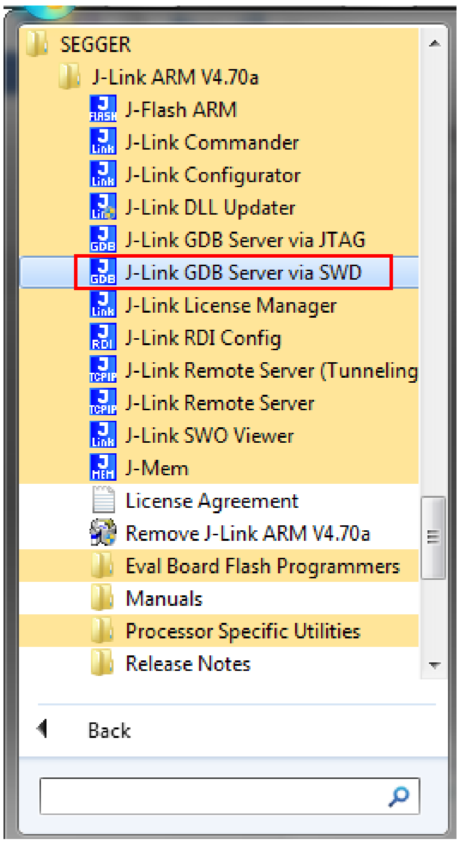
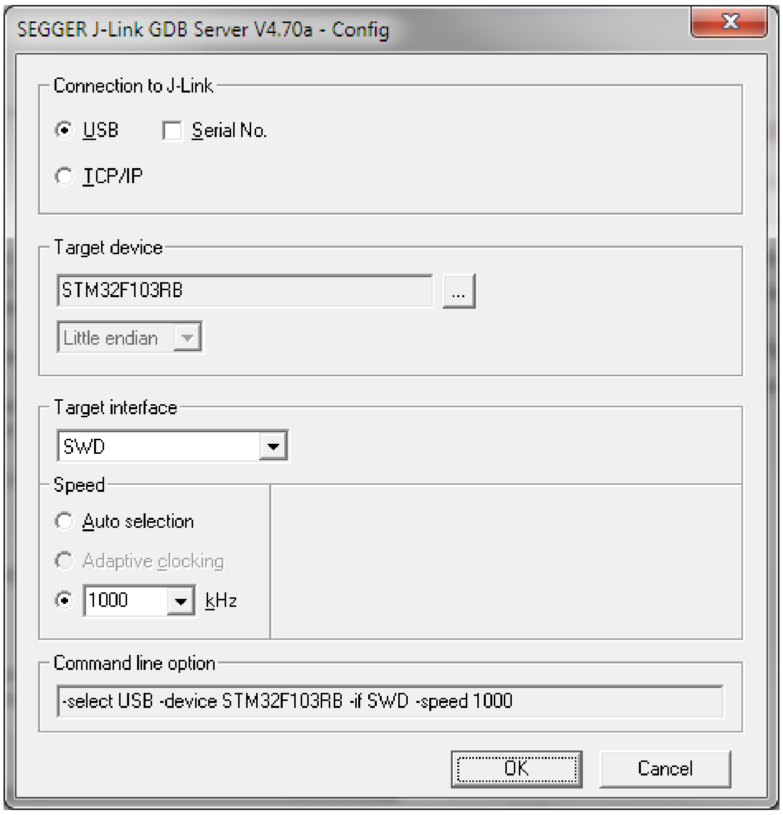


Figure 3-1 Start the J-Link GDB Server

Configure the following settings for the Olimexino STM32:


SEGGER J-Link GDB Server V4.70a - Config

Connection to J-Link
- (●) USB ☐ Serial No.
- ( ) TCP/IP

Target device
- STM32F103RB
- Little endian

Target interface
- SWD

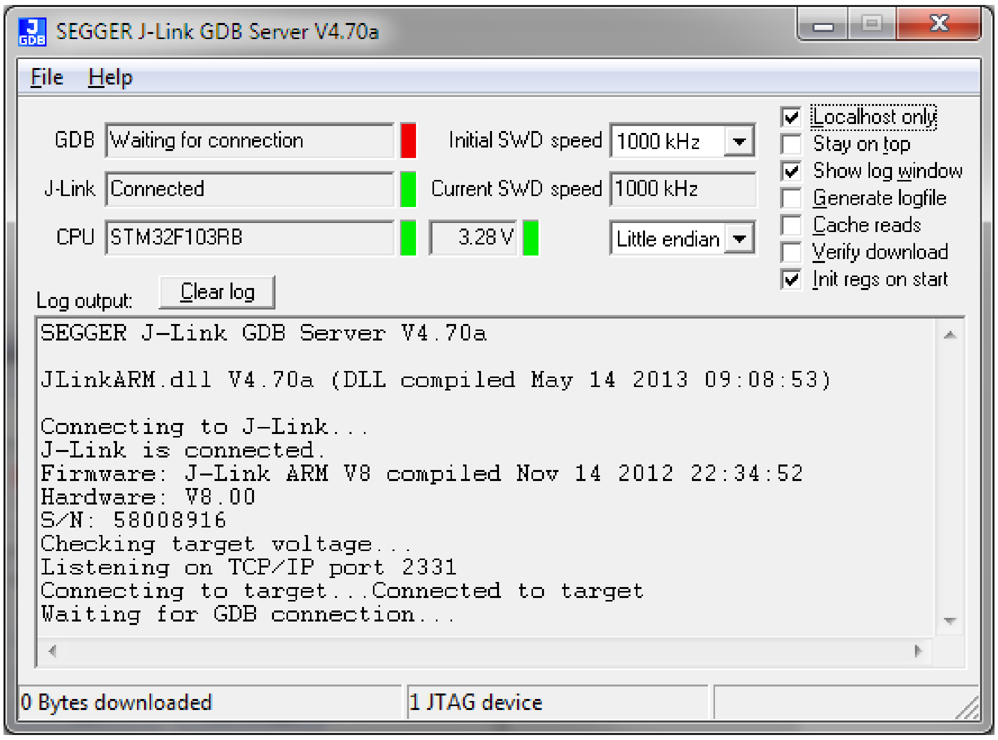
Speed
- ( ) Auto selection
- ( ) Adaptive clocking
- (●) 1000 kHz

Command line option
- -select USB -device STM32F103RB -if SWD -speed 1000

OK Cancel

Figure 3-2 J-Link GDB Server settings

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


SEGGER J-Link GDB Server V4.70a

File Help

GDB	Waiting for connection	🟥	Initial SWD speed	1000 kHz	☑ Localhost only
J-Link	Connected	🟩	Current SWD speed	1000 kHz	☐ Stay on top
CPU	STM32F103RB	🟩	3.28 V 🟩	Little endian	☑ Show log window
					☐ Generate logfile
					☐ Cache reads
					☐ Verify download
					☑ Init regs on start

Log output: Clear log

```
SEGGER J-Link GDB Server V4.70a

JLinkARM.dll V4.70a (DLL compiled May 14 2013 09:08:53)

Connecting to J-Link...
J-Link is connected.
Firmware: J-Link ARM V8 compiled Nov 14 2012 22:34:52
Hardware: V8.00
S/N: 58008916
Checking target voltage...
Listening on TCP/IP port 2331
Connecting to target...Connected to target
Waiting for GDB connection...
```

0 Bytes downloaded | 1 JTAG device

Figure 3-3 J-Link GDB server running

Flash programming a bootloader is done through a batch-file located in directory: ‘\Target\bootloader\Demo\ ARMCM3_STM32_Olimex_Olimexino_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'.

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:

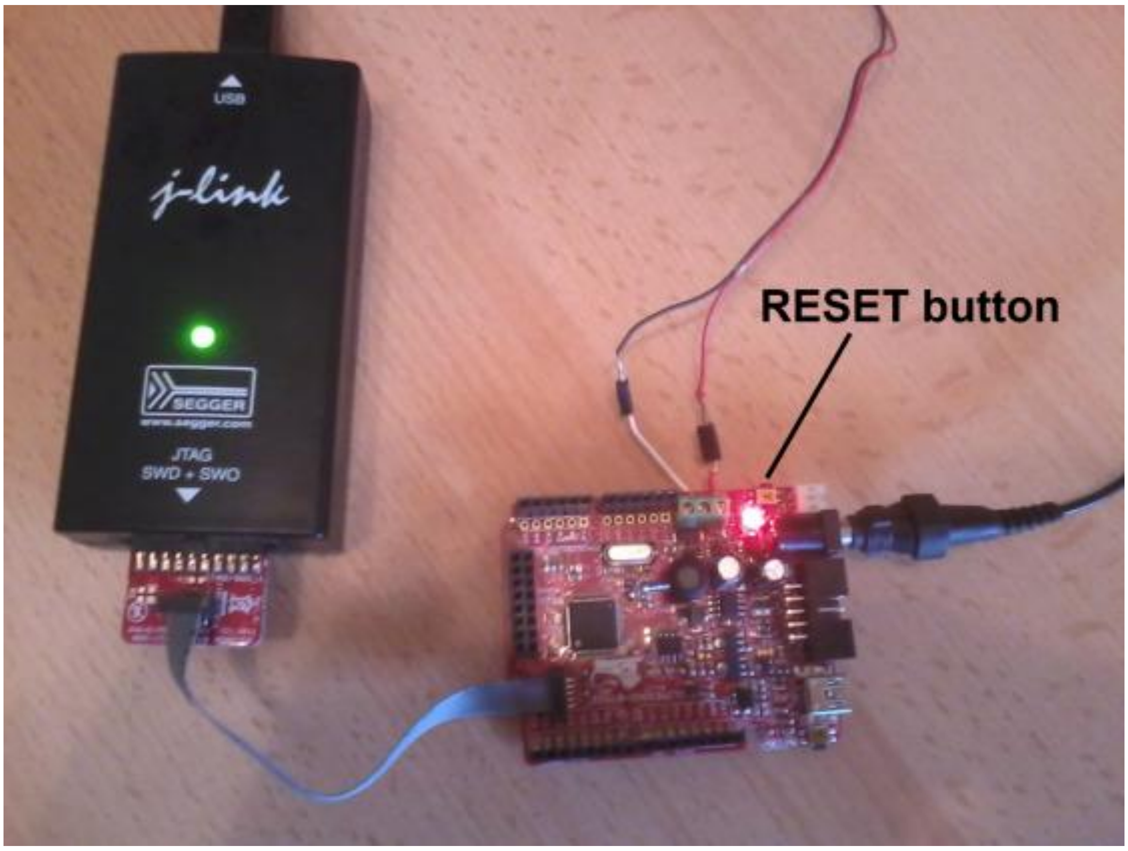
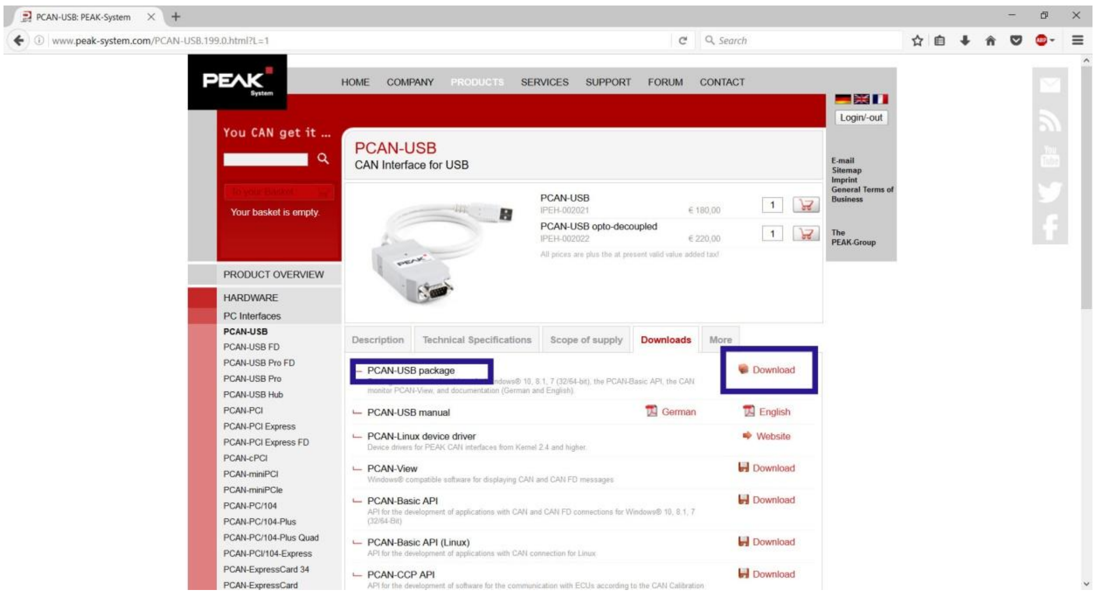


Figure 3-4 Olimexino STM32 connections and reset button

APPENDIX 4. PEAK PCAN INSTALLATION

To download the PEAK PCAN package click here.



PCAN-USB
CAN Interface for USB

PCAN-USB IPEH-002021	€ 180,00	1
PCAN-USB opto-decoupled IPEH-002022	€ 220,00	1

All prices are plus the at present valid value added tax!

PCAN-USB package ...ndows® 10, 8.1, 7 (32/64-bit), the PCAN-Basic API, the CAN monitor PCAN-View, and documentation (German and English)		Download
PCAN-USB manual	German	English
PCAN-Linux device driver Device drivers for PEAK CAN interfaces from Kernel 2.4 and higher		Website
PCAN-View Windows® compatible software for displaying CAN and CAN FD messages		Download
PCAN-Basic API API for the development of applications with CAN and CAN FD connections for Windows® 10, 8.1, 7 (32/64-Bit)		Download
PCAN-Basic API (Linux) API for the development of applications with CAN connection for Linux		Download
PCAN-CCP API API for the development of software for the communication with ECUs according to the CAN Calibration		Download

Figure 4-1 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 5. 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/.

After modifying the bootloader, it needs to be rebuild and then programmed again into the internal flash memory of the microcontroller. Refer to chapter Appendix 3 for instructions for this last part.

Rebuilding the bootloader is achieved by the following step:

Double-click the batch-file called 'build.bat' in directory:
'\Target\bootloader\Demo\ARMCM3_STM32_Olimex_Olimexino_GCC\Boot\cmd'.