# HANcoder Target STM32-P405 Getting Started



Version 0.4 10/4/2016

## 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-P405.

## HANcoder Target STM32-P405 Getting Started

GETTING STARTED GUIDE

#### Contents

VERSIONING	3
INTRODUCTION	4
1. MATLAB INSTALLATION         1.1. Requirements         1.2. Install         1.3. Toolboxes to install         1.4. MATLAB as administrator	<b>5</b> 5 5 6 8
2. GNU ARM TOOLCHAIN INSTALLATION	9 9
3. TEST THE BUILD PROCESS       1         3.1. Explanation of the template model       1         The inputs subsystem       1         The algorithm subsystem       1         The outputs subsystem       1         The configuration blocks       1         The System Information Subsystem       1         3.2. Build your own model       1         3.3. Working with m-scripts       1         3.4. Code generation       1         4.1. Flashing the controller via USB       1	0 2 2 3 3 4 6 8 9 9
4.1.1.       Install bootloader driver	9 2
5. FLASHING THE CONTROLLER WITH THE ST-LINK.       2         5.1. Software program reconfiguration       2         5.2. Uploading the software with the ST-Link       2         5.3. Flashing the bootloader       2	<b>4</b> 4 4 6
6       HANTUNE       2         6.1       Installing the virtual COM port driver       2         6.2       Using HANtune       2	7 7 7
APPENDIX 1. ST-LINK INSTALLATION	0

APPENDIX 2. Installation	SEGGER J-LINK INSTALLATION	<b>32</b> 32
APPENDIX 3.	FLASHING THE BOOTLOADER WITH THE J-LINK	34
APPENDIX 4.	PEAK CAN INSTALLATION	37
APPENDIX 5.	REBUILDING THE BOOTLOADER	37

## 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
0.4	04-10-2016	G. Jansma	Updated the installation and pictures	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 <u>Olimex STM32 P405</u> 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 P405.

Furthermore, it will explain how you can view signals and edit parameters while the software that runs on the P405. 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 <u>toolboxes</u> 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 P405 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 <u>MathWorks site</u>. 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 <u>MathWorks website</u>. 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

intervention       Introduct         Introduct       Introduct         Simuline & S       Simuline & S         Bioinformatics Toolbox 45.1       Introduct & S         Control System Toolbox 39       Introduct & S         Control System Toolbox 3.7       Introduct & S         Data Acquisition Toolbox 3.7       Introduct & S         Image Acquisition Toolbox 3.1       Introduct & S         Image Acquisition Toolbox 3.1       Introduct & S         Image Acquisition Toolbox 3.1       Introduct & S         Image Acquisition Toolbox 4.1       Introduct & S         MatTLAB Compiler 5.0       Introduct & S         Introduct Control Toolbox 9.2       Introduct & S         Introduct Generator 4.1       MatTLAB Compiler 5.0         MatTLAB Compiler 5.0       Introduct & S         MatTLAB Compiler 5.0       Introl & S         MatTLAB Contro	Product Selection		- 0	×
Product       MATLAB 63         Simulic & S         Bioinformatics Toolbox 45.1         Control System Toolbox 62         Control System Toolbox 3.1         Data Acquisition Toolbox 3.7         Data Acquisition Toolbox 3.7         Data Acquisition Toolbox 3.7         Bioinformatics Toolbox 9.0         Embedded Coder 6.3         Fixed-Point Designer 5.0         Fixed-Point Designer 5.0         Fixed-Point Designer 5.0         Image Acquisition Toolbox 3.1         HOL Coder 3.6         Image Acquisition Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5.0         Instrumer Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5.0 5.0         MATLAB Compiler 5.0 5.1         Model Pased Calibration Toolbox 5.0.1         Model Pased Calibration Toolbox 5.0.1         Model Pased Calibration Toolbox 5.0.1         Model Based Calibration Toolbox 2.0         Robotics System Toolbox 7.0         SimDirectine 2.8         SimDirectine 2.8         SimDirectine 2.4         SimDirectine 2.4         SimDirectine 2.4         SimDirection 2.7         SimDirectine 2.8         SimDirection 2.4	lect products to install		MATLA	B
MATLB B3       E2054         Simulink 8.5       Bioinformatics Toolbox 45.1	Product		SIMU	LINK
Simulark & 5         Bioinformatics Toolbox 45.1         Computer Vision System Toolbox 62         Control System Toolbox 9.9         Carve Fitting Toolbox 3.7         Data Acquisition Toolbox 3.7         Database Toolbox 5.0         Dire Fitting Toolbox 3.7         Database Toolbox 9.0         Fibed-Point Designer 5.0         Fitzey Logic Toolbox 2.2.1         Global Optimization Toolbox 3.3.1         HDL Coder 3.6         Image Processing Toolbox 9.2         Instamer Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5DK 6.0         MATLAB Report Generator 4.1         Model Pased Calibration Toolbox 5.0.1         Model Pased Calibration Toolbox 4.8.1         Neural Network Toolbox 7.2         Paratial Computing Toolbox 1.0         SimDirveline 2.8         SimDirveline 2.8 <tr< td=""><td>MATLAB 8.5</td><td></td><td>R2015</td><td>a</td></tr<>	MATLAB 8.5		R2015	a
Bioinformatics Toolbox 45.1         Computer Vision System Toolbox 6.2         Control System Toolbox 9.9         Curve Fitting Toolbox 3.7.1         Data Acquisition Toolbox 3.7         Databace Toolbox 9.0         Embedded Coder 6.8         Fixed-Point Designer 5.0         Image Processing Toolbox 4.9         Image Processing Toolbox 4.9         Image Processing Toolbox 4.9         Image Processing Toolbox 4.9         Image Processing Toolbox 4.1         MATLAB Compiler 5.0         MatLaB Computing Toolbox 4.1         Neural Network Toolbox 8.3         Optimization Toolbox 7.2         Paratlel Computing Toolbox 6.6         Paratlel Computing Toolbox 7.0         SimDireverster Toolbox 7.1 <td>Simulink 8.5</td> <td></td> <td></td> <td>•</td>	Simulink 8.5			•
Computer Vision System Toolbox 6.2         Control System Toolbox 3.5.1         Database Toolbox 5.2.1         DSP System Toolbox 2.2.1         Global Optimization Toolbox 3.3.1         HDL Coder 3.6         Image Processing Toolbox 4.9         Image Processing Toolbox 4.9         Instrument Control Toolbox 3.7.7         Mapping Toolbox 4.1         MATLAB Compiler 50.6         MATLAB Compiler 50.6         MATLAB Compiler 50.6.1         MATLAB Compiler 50.6.1         Model Pased Calibration Toolbox 4.0.1         Model Pased Calibration Toolbox 5.0.1         Model Pased Calibration Toolbox 7.2         Paralel Computing Toolbox 7.0         SimCivenier 2.8         SimCivenier 2.8         SimMic Contol Design 4.2         SimMic Contol Design 4.2	Bioinformatics Toolbox 4.5.1			
Control System Toolbox 9.9         Curve Fitting Toolbox 3.7.         Data Acquisition Toolbox 3.7.         Database Toolbox 52.1         DSP System Toolbox 9.0         Embedded Coder 6.8         Fixed-Point Designer 5.0         Fuzzy Logic Toolbox 2.2.11         Global Optimization Toolbox 4.9         Image Acquisition Toolbox 9.2         Instrument Control Toolbox 3.7         Mapping Toolbox 4.9         Image Processing Toolbox 9.2         Instrument Control Toolbox 3.7         MATLAB Compiler 5.0         Mattrue Control Toolbox 5.0.1         Model-Base Calibration Toolbox 4.8.1         Neural Network Toolbox 7.2         Parallel Computing Toolbox 5.6         Partial Differential Equation Toolbox 2.0         Robotics System Toolbox 7.2         SimDrink 4.	Computer Vision System Toolbox 6.2			
Curve Fitting Toolbox 35.1         Data Acquisition Toolbox 3.7         Database Toolbox 5.1         DsP System Toolbox 9.0         Embedded Coder 6.8         Fixed-Point Designer 5.0         Image Processing Toolbox 4.9         Image Processing Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5.0         MATLAB Compiler 5.0         MATLAB Compiler 5.0 K 6.0         MATLAB Compiler 5.0 K 6.0         MATLAB Complet for Dotox 5.0.1         Model Predictive Control Toolbox 5.0.1         Model-Based Calibration Toolbox 4.8.1         Neural Network Toolbox 7.2         Parallel Computing Toolbox 7.6         Parallel Computing Toolbox 7.0         Signal Processing Toolbox 7.0         SimUriveline 2.8         SimUriveline 2.8         SimUriveline 2.4         SimUriveline 2.4         Simulink Control Design 4.2         Simulink Control Design 4.2         Simulink Real-Time 6.2	Control System Toolbox 9.9			
Data Acquisition Toolbox 3.7         Data base Toolbox 5.2.1         DSF 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         HOL Coder 3.6         Image Acquisition Toolbox 4.9         Image Acquisition Toolbox 4.7         Mapping Toolbox 4.1         MATLAB Conpiler 5.0.         MATLAB Compiler 5.0.         MatLaB Report Generator 4.1         Model-Based Calibration Toolbox 4.8.1         Neural Network Toolbox 8.3         Optimization Toolbox 7.2         Parallel Computing Toolbox 7.0         SimDriveline 2.8         SimEvent 5.4         SimDriveline 2.4         SimDriveline 2.4         SimDriveline 2.4	Curve Fitting Toolbox 3.5.1			
Database Toolbox 52.1         DSP System Toolbox 9.0         Embedded Coder 6.8         Fixed-Point Designer 5.0         Fuzzy Logic Toolbox 2.2.21         Global Optimization Toolbox 3.1         HDL Coder 3.6         Image Acquisition Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5.0         MATLAB Compiler 5.0.         MATLAB Compiler 5.0.         MATLAB Compiler 5.0.         MATLAB Compiler 5.0.         MATLAB Report Generator 4.1         Model Predictive Control Toolbox 5.0.1         Model Predictive Control Toolbox 7.2         Paratial Differential Equation Toolbox 2.0         Robotics System Toolbox 7.0         SimDirevine 2.8         SimDirevine 2.8         SimDirevine 2.8         SimUnik Report Generator 4.1         Simulink	Data Acquisition Toolbox 3.7			
DSP System Toolbox 9.0         Embedded Coder 6.8         Fixed-Point Designer 5.0         Fuzzy Logic Toolbox 2.2.1         Global Optimization Toolbox 3.3.1         HDL Coder 3.6         Image Acquisition Toolbox 4.9         Instrument Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5.0         MATLAB Compiler 5.0.         Mattab Compiler 5.0.         Mattab Computing Toolbox 5.0.1         Model Predictive Control Toolbox 5.0.1         Model Predictive Control Toolbox 5.0.1         Model Predictive Control Toolbox 5.0.1         Signal Processing Toolbox 6.6         Parallel Computing Toolbox 6.6         SimDirection 2.7         SimElectronics 2.7         SimElectronics 2.7 <tr< td=""><td>Database Toolbox 5.2.1</td><td></td><td></td><td></td></tr<>	Database Toolbox 5.2.1			
Embedded Coder 6.8         Fixed-Point Designer 5.0         Fizzy Logic Toolbox 2.21         Global Optimization Toolbox 3.1         HDL Coder 3.6         Image Processing Toolbox 9.2         Instrument Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5.0         MATLAB Compiler 5.0         MATLAB Compiler 5.0.1         Model Predictive Control Toolbox 5.0.1         Model-Based Calibration Toolbox 4.8.1         Neural Network Toolbox 7.2         Parallel Computing Toolbox 6.6         Parallel Computing Toolbox 7.0         SimDreferential Equation Toolbox 2.0         Robotics System Toolbox 7.0         SimDreferential Equation Toolbox 2.0         SimDreferential Equation Toolbox 7.0         SimProferential Equation Toolbox 7.0	DSP System Toolbox 9.0			
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         Instrument Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5.0         MATLAB Compiler 5.0         MATLAB Compiler 5.0         MATLAB Compiler 5.0.1         Model-Based Calibration Toolbox 4.1         MATLAB Compiler 5.0.1         Model Predictive Control Toolbox 5.0.1         Model-Based Calibration Toolbox 4.3.1         Neural Network Toolbox 4.3         Optimization Toolbox 7.2         Parallel Computing Toolbox 7.0         Simpler toolbox 1.0         Signal Processing Toolbox 7.0         SimDriveline 2.8         SimElectronics 2.7         SimWechanics 4.6         SimVerine 4.4         SimVerine 5.3         Simulink XD Animation 7.3         Simulink SD Animation 7.3         Simulink Coder 8.8         Simulink Report Generator 4.1         Simulink Report Generator 4.1         Simulink Design Optimization 2.7         Simulink Design Optimization 2.7         Simulink Report Generator 4.1         Simulink Report Generator 4.1	Embedded Coder 6.8		1	
Fuzzy Logic Toolbox 2.2.21         Global Optimization Toolbox 3.3.1         HDL Coder 3.6         Image Acquisition Toolbox 4.9         Instrument Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 50.0         Model-Based Calibration Toolbox 50.1         Model-Based Calibration Toolbox 7.2         Parallel Computing Toolbox 7.3         Signal Processing Toolbox 7.4         Parallel Computing Toolbox 6.6         Parallel Computing Toolbox 7.0         Signal Processing Toolbox 7.0         Simbriveline 2.8         SimElectronics 2.7         SimElectronics 2.7         SimMurk Add         Simverts 4.4         Simverts 4.4         Simular Koder 8.3         Simular Koder 8.4         Simular Koder 8.8         Simular Koder 8.8         Simular Koder 4.1         Simular Koder 4.1         Simular Koder 4.1         Simular Koder 4.1	Fixed-Point Designer 5.0			
Global Optimization Toolbox 3.3.1         HDL Coder 3.6         Image Acquisition Toolbox 4.9         Image Acquisition Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 6.0         MATLAB Compiler 5DK 6.0         MATLAB Compiler 5DK 6.0         MATLAB Compiler 5DK 6.0         MATLAB Compiler 5DK 6.0         MATLAB Report Generator 4.1         Model Predictive Control Toolbox 5.0.1         Model Predictive Control Toolbox 4.8.1         Neural Network Toolbox 8.3         Optimization Toolbox 7.2         Parallel Computing Toolbox 4.6         Robotics System Toolbox 7.0         Signal Processing Toolbox 7.0         SimDriveline 2.8         SimDriveline 2.8         SimDriveline 2.4         SimPowersystems 6.3         SimEvents 4.4         SimPowersystems 6.3         Simulink 20 Animation 7.3         Simulink 20 Animation 7.3         Simulink Control Design 4.2         Simulink Report Generator 4.1	Fuzzy Logic Toolbox 2.2.21		1	
HDL Coder 3.6         Image Acquisition Toolbox 4.9         Image Processing Toolbox 9.2         Intarumet Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 6.0         MATLAB Compiler 6.0         MATLAB Compiler 5.0         Matter Control Toolbox 5.0.1         Model Predictive Control Toolbox 5.0.1         SimDirition Toolbox 7.0         SimDirition Toolbox 7.0         SimDiriveline 2.8         SimUlink So Animation 7.3         SimUlink Coder 8.8         Simulink Coder 8.8         Simu	Global Optimization Toolbox 3.3.1			
Image Acquisition Toolbox 4.9         Image Processing Toolbox 9.2         Instrument Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 5.0         MATLAB Compiler 5.0         MATLAB Control Toolbox 5.0.1         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 7.0         SimDirveline 2.8         SimDirveline 2.8         SimDirveline 2.8         SimPderbards         SimPderbards         SimSectories 2.7         SimSectories 2.7         SimVente 3.4         SimPowerSystems 6.3         Simvents 4.4         SimPowerSystems 6.3         Simulink 20 Animation 7.3         Simulink Control Design 4.2         Simulink Report Generator 4.1         Simulink Report Gen	HDI Coder 3.6		-	
Image Processing Toolbox 9.2         Instrument Control Toolbox 3.7         Mapping Toolbox 4.1         MATLAB Compiler 6.0         MATLAB Compiler 6.0         MATLAB Compiler 5DK 6.0         MATLAB Compiler 5DK 6.0         MATLAB Compiler 5DK 6.0         MATLAB Compiler 5DK 6.0         MATLAB Report Generator 4.1         Model Pased Calibration Toolbox 4.8.1         Neural Network Toolbox 7.2         Parallel Computing Toolbox 7.2         Parallel Computing Toolbox 7.0         SimDirveline 2.8         SimEctronics 2.7         SimEvents 4.4         SimEvents 4.4         SimPowerSystems 6.3         Simulink Coder 8.8         Simulink Coder 8.8         Simulink Coder 8.8         Simulink Coder 8.8         Simulink Real-Time 6.2         Simulink Real-Time 6	Image Acquisition Toolbox 4.9		1	
Instrument Control Toolbox 3.7           Mapping Toolbox 4.1           MATLAB Coder 2.8           MATLAB Compiler 6.0           MATLAB Compiler 50K 6.0           Model Predictive Control Toolbox 5.0.1           Model Predictive Control Toolbox 4.8.1           Neural Network Toolbox 8.3           Optimization Toolbox 7.2           Parallel Computing Toolbox 6.6           Parallel Computing Toolbox 1.0           Signal Processing Toolbox 7.0           SimDriveline 2.8           SimElectronics 2.7           SimEvents 4.4           SimProvertis 2.7           SimMechanics 4.6           SimProvertis 4.4           SimProvertis 4.4           SimUnink Coder 8.8           Simulink Coder 8.8           Simulink Coder 8.8           Simulink Coder 8.8           Simulink Design Optimization 2.7           Simulink Real-Time 6.2           Simulink Real-Time 6.2           Simulink Real-Time 6.2           System didentification and Validation 3.9	Image Processing Toolbox 92			
Instruct Control           Mapping Toolbox 4.1           MATLAB Compiler 5.0           MATLAB Compiler 5.0K 6.0           MATLAB Compiler 5.0K 6.0           MATLAB Control Toolbox 5.0.1           Model-Based Calibration Toolbox 4.8.1           Neural Network Toolbox 8.3           Optimization Toolbox 7.2           Parallel Computing Toolbox 7.2           Parallel Computing Toolbox 7.2           Robotics System Toolbox 7.0           Signal Processing Toolbox 7.0           SimElectronics 2.7           SimElectronics 2.7           SimElectronics 4.4           SimPowerSystem 6.3           SimSup 7.3           Simulink Coder 8.8           Simulink Coder 8.8           Simulink Coder 0.2           Simulink Coder 0.2           Simulink Real-Time 6.2           System 16 dubing 0.3.9           System 16 dubing 0.3.9           System 16 dubing 0.3.9           System 16 dubing 0.2.2           Vehicle Network Toolbox 2.4	Instrument Control Toolbox 3.7			
MATLAB Coder 2.8         MATLAB Compiler SDK 6.0         MATLAB Compiler SDK 6.0         MATLAB Compiler SDK 6.0         MATLAB Report Generator 4.1         Model Pased Calibration Toolbox 5.0.1         Model Based Calibration Toolbox 4.8.1         Neural Network Toolbox 6.6         Parallel Computing Toolbox 6.6         Parallel Computing Toolbox 7.0         Simplertonics 2.7         SimElectronics 2.7         SimPkyraulics 1.16         SimPkyraulics 1.6         SimPkores 6.3         Simcape 3.13         Simulink Coder 8.8         Simulink Coder 8.8         Simulink Real-Time 6.2         Simulink Real-Time 6.2         Simulink Real-Time 6.2         Simulink Real-Time 6.2         Simulink Coder 8.3         System function 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 2.4	Mapping Toolbox 4.1		-	
MATLAB Compiler 50         MATLAB Compiler 50K 6.0         MATLAB Report Generator 4.1         Model Predictive Control Toolbox 5.0.1         Model Predictive Control 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 7.2         Parallel Computing Toolbox 7.0         Signal Processing Toolbox 7.0         SimDriveline 2.8         SimElectronics 2.7         SimEvents 4.4         SimEvents 4.4         SimEvents 4.4         SimSincape 3.13         Simulink 3D Animation 7.3         Simulink Control Design 4.2         Simulink Control Design 4.2         Simulink Control Design 4.2         Simulink Control Autoriation 3.9         Spreadsheet Link EX 3.2.3         Stateflow 8.5         Statistics and Machine Learning Toolbox 10.0         Symbolic Math Toolbox 9.2         Vehicle Network Toolbox 9.2	MATLAB Coder 28		-	
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 7.2         Parallel Computing Toolbox 7.2         Parallel Computing Toolbox 7.2         Parallel Computing Toolbox 7.0         Signal Processing Toolbox 7.0         SimDriveline 2.8         SimEvents 4.4         SimPydraulics 1.16         SimRevents 4.4         SimPydraulics 1.16         SimRevents 4.4         SimPowerSystems 6.3         Simulink 2D Animation 7.3         Simulink Control Design 4.2         Simulink Control Design 4.2         Simulink Control Design 4.2         Simulink Control Generator 4.1         Simulink Report Generator 4.1         Simulink Verification and Validation 3.9         Spreadsheet Link EX 3.2.3         Stateflow 8.5         Stateflow 8.5         Stateflow 8.5         System Identification Toolbox 9.2         Vehicle Network Toolbox 2.4	MATLAB Compiler 60			
MATLAB Report Generator 4.1         Model Predictive Control Toolbox 5.0.1         Model Pased Calibration Toolbox 8.3         Optimization Toolbox 7.2         Parallel Computing Toolbox 6.6         Partial Differential Equation Toolbox 2.0         Robotics System Toolbox 7.0         Signal Processing Toolbox 7.0         SimDriveline 2.8         SimElectronics 2.7         SimEvents 4.4         SimPowerSystems 6.3         SimSurents 4.6         SimUlink Coder 8.8         Simulink Coder 8.8         Simulink Report Generator 4.1         Simulink Report Generator 4.1         Simulink Report Generator 4.1         Simulink Report Generator 4.1         Simulink Verification and Validation 3.9         Spreadsheet Link EX 3.23         Stateflow 8.5         Stateflow 8.5         Stateflow 8.5         Stateflow 8.5         Stateflow 8.5         Stateflow 8.2         Vehicle Network Toolbox 9.2         Vehicle Network Toolbox 2.4	MATLAB Compiler SDK 6.0		-	
Invariant         Model Predictive Control Toolbox 5.0.1         Partial Differential Equation Toolbox 2.0         Robotics System Toolbox 1.0         Signal Processing Toolbox 7.0         SimElectronics 2.7         SimElectronics 2.7         SimElectronics 2.7         SimEvents 4.4         SimPowerSystems 6.3         Simscape 3.13         Simulink D Animation 7.3         Simulink Control Design 4.2         Simulink Control Design 4.2         Simulink Real-Time 6.2         Simulink Verification and Validation 3.9         Spreadsheet Link EX 3.2.3         Stateflow 8.5         Stateflow 8.5         Statistics and Machine Learning Toolbox 10.0         Symbolic Math Toolbox 6.2         System Identification Toolbox 2.4	MATLAB Complet Spic 0.0		-	
Model-Based Calibration Toolbox 3.8.1         Model-Based Calibration 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 7.2         SimDriveline 2.8         SimElectronics 2.7         SimEvents 4.4         SimHydraulics 1.16         SimMechanics 4.6         SimPowerSystems 6.3         Simulink 20 Animation 7.3         Simulink Coder 8.8         Simulink Coder 8.8         Simulink Real-Time 6.2         Sitatistics and Machine Learning Toolbox 10.0         System Identification	Model Predictive Control Toolbox 5.0.1			
Induct-based calobation roolbox 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           SimPowerSystems 6.3           Simscape 3.13           Simulink 20 Animation 7.3           Simulink Coder 8.8           Simulink Control Design 4.2           Simulink Real-Time 6.2           Simulink Real-Time 6.2           Simulink Real-Time 6.2           Simulink Ket K 3.2.3           Stateflow 8.5           Stateflow 8.5           Stateflow 8.5           Stateflow 8.5           Stateflow 8.5           System Identification Toolbox 9.2           Vehicle Network Toolbox 2.4	Model Predictive Control Toolbox 5.0.1			
Network roomous as         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         SimElectronics 2.7         SimMechanics 4.4         SimMechanics 4.6         SimMoverSystems 6.3         Simscape 3.13         Simulink 20 Animation 7.3         Simulink Control Design 4.2         Simulink Control Design 4.2         Simulink Report Generator 4.1         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	Neural Network Teelbox 9.2		-	
Optimization Toolbox 7.2         Parallel Computing 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         SimMexins 6.3         Simulink 3D Animation 7.3         Simulink Coder 8.8         Simulink Control Design 4.2         Simulink Report Generator 4.1         Simulink Report Generator 4.1         Simulink Kest 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 9.2	Ontimization Toolbox 7.2			
Partial Differential Equation Toolbox 2.0         Robotics System Toolbox 1.0         Signal Processing Toolbox 7.0         SimDriveline 2.8         SimElectronics 2.7         SimMechanics 1.16         SimMechanics 4.6         SimMoverSystems 6.3         Simulink 20 Animation 7.3         Simulink Coder 8.8         Simulink Control Design 4.2         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 9.2	Parallel Computing Toolbox 6.6			
Partial Differential Equation Footbox 2:0         Robotics System Toolbox 1.0         Signal Processing Toolbox 7.0         SimDriveline 2.8         SimElectronics 2.7         SimEvents 4.4         SimMechanics 4.6         SimPowerSystems 6.3         Simulink 3D Animation 7.3         Simulink Coder 8.8         Simulink Control Design 4.2         Simulink Real-Time 6.2         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	Partial Differential Equation Taolhow 2.0		-	
Nobolics System Toolbox 1.0         Signal Processing Toolbox 7.0         SimDriveline 2.8         SimElectronics 2.7         SimElectronics 2.7         SimElectronics 1.16         SimMechanics 4.6         SimScape 3.13         Simulink 3D Animation 7.3         Simulink Coder 8.8         Simulink Control Design 4.2         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	Partial Differential Equation Toolbox 2.0		-	
Signal Processing rootbox 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 Real-Time 6.2         Simulink Report Generator 4.1         Simulink Verification and Validation 3.9         Spreadsheet Link EX 3.2.3         Statifics and Machine Learning Toolbox 10.0         System Identification Toolbox 9.2         Vehicle Network Toolbox 2.4	Cignal Processing Tablhay 70			
SimDriverine 2.8         SimElectronics 2.7         SimEvents 4.4         SimHydraulics 1.16         SimMechanics 4.6         SimScape 3.13         Simulink 3D Animation 7.3         Simulink Coder 8.8         Simulink Control Design 4.2         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         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	Signal Processing Toolbox 7.0		-	
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 Real-Time 6.2         Simulink Real-Time 6.2         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	Simpliveline 2.8		-	
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         Stateflow 8.5         Stateflow 8.5         Symbolic Math Toolbox 6.2         System Identification Toolbox 9.2         Vehicle Network Toolbox 2.4	SimElectronics 2.7			
SimPrydraulics 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 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	SimEvents 4.4			
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				
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	Simmechanics 4.6			
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	SimPowerSystems 6.3			
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	Simscape 3.13		-	
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	Simulink 3D Animation 7.3		-	
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	Simulink Coder 8.8		-	
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	Simulink Control Design 4.2		-	
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	Simulink Design Optimization 2.7		-	
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	Simulink Real-Time 6.2		-	
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	Simulink Report Generator 4.1		-	
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	Simulink Verification and Validation 3.9			
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	Spreadsheet Link EX 3.2.3		-	
Statistics and Machine Learning Toolbox 10.0         Symbolic Math Toolbox 6.2         System Identification Toolbox 9.2         Vehicle Network Toolbox 2.4	Stateflow 8.5		-	
Symbolic Math Toolbox 6.2         System Identification Toolbox 9.2         Vehicle Network Toolbox 2.4	Statistics and Machine Learning Toolbox 10.0			
System Identification Toolbox 9.2         Vehicle Network Toolbox 2.4	Symbolic Math Toolbox 6.2		-	
Vehicle Network Toolbox 2.4	System Identification Toolbox 9.2			
	Vehicle Network Toolbox 2.4			
Deale Next	Alext.	Court I	Unit	

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  $\rightarrow$  search for Matlab R2015a  $\rightarrow$  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.

eneral	Shortcut	Options	Font	Layout	Colours
Compatib	bility	Security	Details	Previous	s Versions
his progr	ram isn't wor compatibili	king correctly o ty troubleshoote	n this version er.	of Windows, t	ry
Runco	ompatibility t	roubleshooter			
wdolc	hoose com	atibility settings	manually?		
Compatib	oility mode				
Runti	his program	in compatibility	mode for:		
Window	rs 8			×.	
ettings					
Redu	ced colour	node			
8-bit (25	6) colour	~			
Runi	n 640 x 480 s	creen resolutio	n		
Disat	ole display s	caling on high [	OPI settings		
Runt	his program	as an administr	ator		
Ch	ange setting	s for all users			

Figure 1-4 MATLAB R2015a Properties, Compatibility tab

## 2. GNU ARM TOOLCHAIN INSTALLATION

The P405 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 <u>website</u>:

Ip GNU ARM Embedded Too × +					- 0	×
🔶 (i) 🖨 Canonical Group Ltd (GB)   https://launchpad.ne	et/goc-arm-embedded	E.	a c Search	合 自	🔸 n 🛡 🐠	- =
GNU ARM Embed	ded Toolchain				Log in / Regi	ilster (
Overview Code Bugs Blueprints Translations	Answers					
Registered 2011-11-30 by 🤱 Joey Ye				Get Involved		
Pre-built GNU toolchain from ARM Cortex-M 8	& Cortex-R processors (Cortex-M0/M0+/M3/M4/M7			Report a hug		
and ARMv8-M Baseline and Mainline, Cortex-H	R4/R5/R7/R8).			Ask a question	-	4
As part of its ongoing commitment to maintaining and enhan	ncing GCC compiler support for the			Help translate	-	
Embedded ARM Processors, namely Cortex-R/Cortex-H/ Cortex-M3, Cortex-H4, Cortex-M0+, Cortex-H7, ARM/8-M Bac Cortex-M3, Cortex-H4, Cortex-M0+, Cortex-H7, ARM/8-M Bac	essor families, covering Cortex-M0, seline and mainline, Cortex-R4, escars at require interpole			Register a blueprint	-	•
tested binaries from the ARM embedded branch. The improve	enses at regular intervals pre-built and ements are freely available for			Downloads		
integration into 3rd party toolchains, and for direct download	d by end-users.			Latest version is 5-2016-q2-update		
ARM employees are maintaining this project. Contributions to trunk, binutls-gdb [2] master branch and newlib [3] master br communication and bug reporting. Downloads are made from no code change is done in Launchpad project.	o this project are be done via GCC [1] ranch. This launchpad project is for n developer.arm.com website [4] and			release.txt	- 0	
[1] http://gcc.gnu.org [2] http://www.gnu.org/software/binutils/ [3] https://sourceware.org/newlib/ [4] https://developer.am.com/open-source/gnu-toolchain/gr	nu-rm			gcc-arm-non_2-win32.exe	•	
For Ubuntu 12.04/14.04/15.10/16.04 32/64-bit user, PPA is av net/~team-gcc-arm-embedded/+archive/ubuntu/ppa.	vailable at https://launchpad.			gcc-arm-non_2-win32.zip	0	
Questions can be asked on the ARM connected community w community/community-help	vebsite at https://community.arm.com/			gcc-arm-non_nux.tar.bz2	0	
Bugs should be reported in the bug section (https://bugs.laur +filebug).	nchpad.net/gcc-arm-embedded/					
Home page External downloads				gcc-arm-non_mac.tar.bz2	0	
Project information		Series and milestones	View full history	scc-arm-non_src.tar.bz2	-0	Ī.
Maintainer:	Driver:	Q, Q, 6.0				
GCC ARM Embedded Maintainers	GCC ARM Embedded Developers	6-2016-q4-major				
Licence:		5.0	5.0 > 2016-09-30	How-to-builolchain.pdf		
Licence	v2.1, GNU LGPL v3, MIT / X / Expec	5-2015-q4-major 5-2016-q1-update 5-2016-q2-update 5-2016-q3	-update (1)			5 1
PDF metadata		4.9-2014-q4-major 4.9-2015-q1-update 4.9-2015-q2-update 4.9-2015-q3	-update	readme.txt	0	
		4.8-2013-q4-major 4.8-2014-q1-update 4.8-2014-q2-update 4.8-2014-q3- 4.7	update 4.8-2015-q3-update	license.txt		Ĩ.
		6.0 series is the current focus of development.		relevand 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.



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

Figure 2-2 Setup wizard properties

## 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\_P405.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.

Folder:	Target		
		Select Folder	Cancel
			d

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

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

Digital Input	
UT BUT	Button
Digital Input	

#### 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

Here the algorithm can be placed. No hardware dependent blocks should be used here.
Switch LED frequency when button is pressed



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



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 P405 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.

Blockset version 0.33 Base Sample Time: 10ms Base Sample Time config
XCP over USB
Custom software ID: "softwareID" Software version: "1" Custom ID config
Build settings Auto Add: on Auto Flash: USB Build settings
Figure 3-8 Configuratior 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:

Pa H	ANcod	ler_Olim	exino/H	ANcoder ST	132 Target	- Olimexino-	STM32	2 algorit	nm - Simulin <mark>l</mark>	cacademic use				
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>D</u> ispla	y <mark>Di</mark> ag <u>r</u> am	<u>S</u> imulation	n <u>A</u> nalysis	Code	e <u>T</u> ools	<u>H</u> elp					
<b>•</b>	- 8	<p td="" ⇒<=""><td>☆</td><td>22 🕥 🗕 🗄</td><td>i - 📦</td><td>4 🕑 🕪</td><td></td><td>•</td><td>2</td><td>Normal</td><td>•</td><td>⊘ -</td><td><b>₩</b> •</td><td></td></p>	☆	22 🕥 🗕 🗄	i - 📦	4 🕑 🕪		•	2	Normal	•	⊘ -	<b>₩</b> •	
Mode	Browse	er	•=	HANcoder S	STM32 Targe	t - Olimexino-S	5TM32 a	algorithm						



Simulink Library Browser				0.00	
📮 🗇 Enter search term 🗸 🍖 🛨 📑 🗕	3				
imulink					
Simulink         Computer Vision System Toolbox         Control System Toolbox         DSP System Toolbox HDL Support         Embedded Coder         Fuzzy Logic Toolbox         HANcoder STM32 Target         HDL Coder         Image Acquisition Toolbox         Instrument Control Toolbox         Model Predictive Control Toolbox         Neural Network Toolbox         Neural Network Toolbox         SimEvents         Simscape         Simulink Coder         Simulink Coder         Simulink Design Optimization         Simulink Real-Time         Simulink Verification and Validation         Stateflow         System Identification Toolbox	Commonly Used Blocks	Continuous Continuous	Discontinuities Discontinuities Model Verification Sinks	Discrete Discrete Model-Wide Utilities Sources	Logic and Bit Operations Ports & Subsystems User-Defined Functions

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.

#### HANcoder Target STM32-P405 Getting Started, v0.4



Figure 3-13 Library pop up



Figure 3-14 HANcoder library STM-P405

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 \rightarrow File \rightarrow Model properties \rightarrow Callbacks. The Model pre-load function runs every command that is written in the section before the model starts.** 

🛅 Mode	l Properties: H	ANcoder_OI	limexino						×
Main	Callbacks	History	Description	Data					
Model ca	allbacks		Model p	re-load fu	unction:				
PreLo PostLu InitFc StartF Pause Contir StopF PreSa PostS Closel	adFon* adFon n ron Fon sueFon on veFon aveFon Fon*		libraryS eval([b	etup droot,'_si	tartup']) %	Calling	[modelName	]_startup.m	
					Q	K	Cancel	<u>H</u> elp	Apply

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\_P405\_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 mscript can be run manually by clicking run. The parameter will be added to the workspace.

The HANcoder\_P405\_startup.m is shown below:

```
% This m-file will be run automatically when starting model 'HANcoder P405'
% 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 <u>View diagnostics</u> on the bottom of the model.

🔁 Diagnostic Viewer	-	×
📰 -   📡 🛣 🐻   🌆   🔍 search 🗢 💠   🚳 -	2	
HANcoder P405 💿		
### compliing usba_desc.c -o usba_desc.opj		
### Compiling usbd_usr.c -o usbd_usr.obj		<u> </u>
### Compiling usb_bsp.c -o usb_bsp.obj		
### Compiling timeout.c -o timeout.obj		
### Compiling tp.c -o tp.obj		
### Compiling diag.c -o diag.obj		
### Compiling CANopen.c -o CANopen.obj		
### Compiling CO_Emergency.c -o CO_Emergency.obj		
### Compiling CO_HBconsumer.c -o CO_HBconsumer.obj		
### Compiling CO_NMT_Heartbeat.c -o CO_NMT_Heartbeat.obj		
### Compiling CO_OD.c -o CO_OD.obj		
### Compiling CO_PDO.c -o CO_PDO.obj		
### Compiling CO_SDO.c -o CO_SDO.obj		
### Compiling CO_SDOmaster.c -o CO_SDOmaster.obj		
### Compliing CO_SINC.c -0 CO_SINC.obj		
### complling CO_ariver.c =0 CO_ariver.obj		
### compiling co_eeprom.c -o co_eeprom.obj		
### Compiling co_node.c =0 co_node.obj		
### Complifing Croid-Cold.c -D Croid-Cold.obj		
### Created nameoder_SINS2_F405_KiL.a full-time fibrary		
### Dinking Mancoder_F403.en		
### Created Rinary file: HaNcoder P405 hin		
### Created Motorola S-record file: HANcoder P405.srec		
### Displaying program size info		
text data bss dec hex filename		
31208 304 28540 60052 ea94/HANcoder P405.elf		
### Postprocessing ASAP2 file		
### Starting automatic flash procedure		
### Successful completion of Real-Time Workshop build procedure f	or model:	
HANcoder_P405		in the second
		_
Build process completed successfully		
		<b>_</b>

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 a P405 with the correct bootloader. If you wish to flash the bootloader or just flash the software directly onto the P405 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 P405 via the CANbus or via an USB connection. This paragraph describes how to flash the P405 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 P405 board.
- The P405 has the correct bootloader on it



Figure 4-1 RESET button P405

#### 4.1.1. Install bootloader driver

To install the driver the program should be kept in bootloader mode. This is achieved by keeping the WAKE UP 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 <u>here</u>.

#### 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: <u>OpenBLT USB Driver Package</u>. 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**  $\rightarrow$  **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:

#### HANcoder Target STM32-P405 Getting Started, v0.4

Create New Device		
Copen		5 Search openblt_usb_driver
Organiza 🔻 New folder		
Favorites	Name	Date modified Type
Desktop	openblt.cfg	08/04/2016 10:50 CFG File
SkyDrive SkyDrive ⇒ Libraries ⇒ Documents → Music ⇒ Pictures ⇒ Subversion ⇒ Videns File <u>n</u> ame: openblt.cfg	< m	✓ Zadig device config (*.cfg)     ✓     Open      Cancel
Zadig     Device Options Help     WnUSB Bulk Device     Driver     USB ID 1D50 60AC     WCID 2	WinUSB (v6. 1.7600. 16389	Edt Hore Information WinUSB (Ibusb) Ibusb-win32 Ibusbx WinUSB (Microsoft)

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	_ 0	x
<u>File Action View H</u> elp		
👂 👰 Jungo		*
⊳ · C Keyboards		
Mice and other pointing devices		
Monitors		
Network adapters		
Processors		_
Sound, video and game controllers		
⊳ ₁∎ System devices		
🖡 🚽 Universal Serial Bus controllers		=
🖌 🖕 Universal Serial Bus devices		
🚽 🚽 WinUSB Bulk Device		
		Ψ.

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 Motorola S-Record of your software program. The one from the demo model is located at '\Target\controller\_STM32\_P405.srec'.

MicroBoot v1.01		22	
MicroBoot for OpenBLT using USB			
Select file to start downlo	ad1	Browse	
	E Select file for download		x
	Look in: 📗 Target	- 🗕 🖆 📰 -	
	Name	Date modified	Туре 🖍
	ontroller_STM32_P405_ert_rtw	2-12-2014 9:49	File folder
	👩 slprj	15-5-2014 11:03	File folder
	controller_Olimexino_STM82.srec	19-11-2014 8:50	SREC File
	controller_STM32_E407 srec	2-12-2014 8:06	SREC File
	2 O controller_STM32_P405.srec	2-12-2014 9:49	SREC File 👻
	•		Þ
	File name: controller_STM32_P405.srec		3 Open
l	Files of type: Motorola S-Record (*.s19;*.s28;*.s37;*	.sx;*.srec;*.mot)	Cancel

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_STM32_P405.srec				
MicroBoot				
Programming data (16,6 of 29,6 Kbytes)				
Elapsed time: 00:03	Settings	Cancel		

Figure 4-6 Bootloader progress

## 4.2. Flashing the controller via CAN

It is possible to flash the P405 via the CANbus or via an USB connection. This paragraph describes how to flash the P405 using the CANbus.

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

- The board is connected to the CAN bus.
- The Peak PCAN interface is connected to the PC's USB port
- Power is supplied to the P405 board.
- The P405 has the correct bootloader on it

Start the MicroBoot utility. It can be found in '\Host\MicroBoot\MicroBoot.exe'. Next, click the Browsebutton to select the Motorola S-Record of your software program. The one from the demo model is located at '\Target\HANcoder\_P405.srec'.

HicroBoot v1.01	to Specific 7 arrays (18)	X	
MicroBoot for OpenBLT using Peak CAN Interface			
Select file to start download	Browse	a	
	Settings Cance	el	Y
	Look in: Look in:	← È 📸	
	Name	Date	Ту
	🦻 blockset	15-4-2015 16:29	Fil€
	i bootloader	15-4-2015 16:29	Fil€
	ANcoder_P405_ert_rtw	17-9-2015 9:40	Fil€
	👩 slprj	17-9-2015 9:40	File
2	HANcoder_P405.srec	17-9-2015 9:41	SR
	■ III	3	4
	File <u>n</u> ame: HANcoder_P405.srec		<u>O</u> pen
	Files of type: Motorola S-Record (*.s19;*.s	28;*.s37;*.sx;*.srec;*.mot)	Cancel

Figure 4-1 - 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.

🔛 MicroBoot v1.00 - Downloading controller_Olin	nexino_STM32.srec	×
MicroBoot for OpenBLT using Peak CAN Interface		
Programming data (9,4 of 22,8 Kbytes)		
Elapsed time: 00:03	Settings	Cancel

Figure 4-2 - 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 1 Appendix 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.

Configuration Parameters: HANco	der_P405/Configuration (Active)	X	
Select:	Target selection	<b>A</b>	
Solver Data Import/Export Diagnostics Hardware Implementation Model Referencing Simulation Target Code Generation Report Comments Symbols Custom Code Debug Interface Verification Code Style	System target file: ert.tlc Language: C Description: Real-Time V Target hardware: None Build process Compiler optimization level: Makefile configuration I Generate makefile Make command: Template makefile:	Workshop Embedded Coder (no auto configuration)         Optimizations off (faster builds) •         make_rtw HANcoder_PORTABLE=1 HANcoder_BOOTLOADER=0         STM32_P405_gcc.tmf	
Templates       Code Placement         Data Type Replacement       Data specification override         Memory Sections       Ignore custom storage classes			
	Prioritized objectives: Unspec Check model before generatin	ified Set g code: Off • Che	
0		QK <u>Cancel H</u> elp <u>Apply</u>	

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 P405 STM32 board through the J-tag connector.
- The ST-Link is connected to the PC's USB port.
- Power is supplied to the P405 board.
- The ST-Link utility is installed

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

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 with or by simply pressing "Enter".

The following screen should appear:

1	STM32 ST-LIN	NK Utility							
E	ile <u>E</u> dit <u>V</u> iev	v <u>T</u> arget S	ST-LINK Ex	ternal Loade	er <u>H</u> elp				
E	3 🖥 🖕 🤅	🤹 🥒 🗭	iii 🔍						
l c	Memory display						Device	STM32F40xx/F41xx	
	Address: 0x08	8000000 -	Size: (	)x33B1	Data Width:	32 bits 🔻	Device ID	0x413	
	, addresser		0.201		bata matin		Revision ID	Rev Z	
	Device Memory @	0x0800000	: Binary File				Fidsh size	Impytes	LiveUpdate
T	arget memory, A	ddress range:	[0x0800000	0x080033B1]	_		_		
	Address	0	4	8	С	ASCII			
	0x08000000	200021E4	080030F5	080030E1	080030E1	ä!. õ0á0á0.			=
	0x08000010	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			
	0x08000020	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			
	0x08000030	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			
	0x08000040	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			
	0x08000050	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			
	0x08000060	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			
	0x08000070	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			
	0x0800080	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			
	0x08000090	080030E1	080030E1	080030E1	080030E1	á 0 á 0 á 0 á 0			-
	<			Sec. 1 (012) (11 ) (11)					4
1	4:55:38 : Old ST	-LINK firmwar	re detected!	54 (Need opd	latey				•
	Pleas 4:55:38 : Conne	e upgrade it fi cted via SWD.	rom ST-LINK->	>'Firmware up	idate' menu.				
1	4:55:38 : Conne	ction mode : N	Normal.						
1	4:55:38 : Debug 4:55:38 : Device	ID:0x413	mode enable	α.					
1	4:55:38 : Device	flash Size : 1	MBytes						
	4:55:44 : Binarv	File closed.	21 -1000						=

figure 5-2 ST-Link Utility connected to board

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

stm32 st-Link Utility		-	
Eile Edit View Target ST-LINK External Loader Help			
🖴 🖥 👙 🛠 🖉 💋 📾			
Memory display	Device		
Address 0x02000000 City 0x2071 Date Width 22 bits	Device ID		
Address: 0x0800000 V Size: 0x2071 Data Width: 32 bits V	Revision ID		
Device Memory Elia : apophti alimax alimavina uch crac	Flash size		
Device Memory			Liveupdat
o the memory			
figure 5-3 ST-Link Utility Prog	ram and Verify		

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

Solution Open	and Locality Marks		X
	▶ trunk ▶ Target ▶	✓ ✓ Search Target	٩
Organize   New for	der	□ □ □	
Documents	Name	Date	Туре
Music	👌 blockset	15-4-2015 16:29	File folder
My Documents     Dictures	🌛 bootloader	15-4-2015 16:29	File folder
Subversion		17-9-2015 9:40	File folder
La Videos	🧑 slprj	17-9-2015 9:40	File folder
	MANcoder_P405.bin	17-9-2015 9:41	IZArc BIN
scomputer	= 👔 HANcoder_P405.srec	17-9-2015 9:41	SREC File
😻 Windows 7 (C:)			
👢 online.han.nl			
👢 Wiki environment			
	▼ <		Þ
File <u>r</u>	ame: HANcoder_P405.srec	<ul> <li>✓ Supported Files (*.</li> <li>Open ▼</li> </ul>	bin *.hex : ▼ Cancel

figure 5-4 Select the .srec

The following screen will appear:

Download [ HANcoder_	P405.srec ]	X
Start address : 0x0800	04000	
File path : C:\Da	ta\HAN\Fast and Curious\STM32\trunk\Target\HANcoc	Browse
Verification		
Overify whether the second	ile programming 💫 🔘 Verify after programming	
Click "Start" to program	target.	
📝 Reset after program	ming	
	Start Cancel	

figure 5-5 Flashing target with ST-Link Utility

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 P405 in exactly the same way as a program from Simulink. The bootloader files can be found in:

'\Target\bootloader\Demo\ARMCM4\_STM32\_Olimex\_STM32P405\_GCC\Boot\bin' use the openbtl\_olimex\_stm32p405\_can.srec for the CAN bootloader and the openbtl\_olimex\_stm32p405\_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 P405. 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 P405. When the P405 is connected the device manager will show it as STM32 Virtual COM port. This is not the right driver.



figure 6-1 Device Manager virtual 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 P405 as STMicroelectronics Virtual COM Port, see below:



figure 6-2 STMicroelectronics in Device Manager

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

Note: When the P405 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'.

#### HANcoder Target STM32-P405 Getting Started, v0.4



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 7.4, to know if you are using too many signals or not.

There are different ways to connect to the P405, 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 6-5 – P405 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 6-6 – 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.

Communication Window Preference	HANtune 1.0 Beta - Connection Progress	HANtune 1.0 Beta - Connection Progress
Connect to XCP device F5 Communication Settings Enable Datalogging Modify Datalog Filename	Connecting to XCP slave ECU:  Connect to CAN network Initialize XCP session Request event information Request / Calibrate parameter values Configure DAQ list	Connecting to XCP slave ECU: Connect to CAN network Linitialize XCP session Request event information Request / Calibrate parameter values Configure DAQ list
	Connect & Request Connect & Calibrate Retry Cancel	Connect & Request Connect & Calibrate Retry Cancel

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.

Connected to XCP device: XCP_RC30	1 DAQ list(s): 8 signals	Max: 100Hz	10 🌲	Current: 10Hz	1		Logfile: OFF
				0	4300	1	

Figure 6-8 - HANtune Communication status

## APPENDIX 1. ST-LINK INSTALLATION

The STM32 ST-Link Utility can be downloaded for free from <u>www.ST.com</u>. 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.



figure 1-1 Installshield

Click next until the license agreement screen appears:

InstallShield Wizard	X				
License Agreement Please read the following license agreement carefully.	N				
Press the PAGE DOWN key to see the rest of the agreement.					
SOFTWARE LICENSE AGREEMENT	* 11				
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.	Ŧ				
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.					
< <u>Back</u> <u>Y</u> es <u>N</u> o					

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.

Device Driver Installation Wizard			
	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.		
	< <u>B</u> ack <u>Next</u> > Cancel		

figure 1-3 Installation wizard

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

Device Driver Installation Wizard	and and	10-1-10-10 M		
	Completing the Device Driver Installation Wizard			
25	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		
	V S I WICIOElectronic	Ready to use		
	< <u>B</u>	ack Finish Cancel		

figure 1-4 Completed installation

## APPENDIX 2. SEGGER J-LINK INSTALLATION

This part is only necessary if you do not possess an P405 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:

Installing	
	Current File Copying file: C:\\Samples\JFlash\ProjectFiles\LM3S1166.jflash All Files Time Remaining 1 minute 10 seconds
	< Back Next > Cancel

Figure 2-3 - Installation process

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

😼 Installation Complete		x
	J-Link ARM V4.70a has been successfully installed. USB Driver for J-Link installed Press the Finish button to exit this installation.	
	< <u>B</u> ack <b><u>Finish &gt;</u> Car</b>	ncel

#### 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- P405 board.
- The Segger J-Link is connected to the PC's USB port.
- Power is supplied to the STM32- P405 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- P405:

SEGGER J-Link GDB Server V4.70a - Config			
Connection to J-Link			
Image: Contract of the second sec			
O ICP/IP			
Target device			
STM32F103RB			
Little endian 💌			
☐ Target interface			
SWD			
Speed			
C Auto selection			
C Adaptive <u>clocking</u>			
Command line option			
-select USB -device STM32F103RB -if SWD -speed 1000			
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					
<u>F</u> ile <u>H</u> elp					
GDB Waiting for connection Ini J-Link Connected Curr CPU STM32F407ZG 3	ial SWD speed 1000 kHz ent SWD speed 1000 kHz 29 V	<ul> <li>✓ Localhost only</li> <li>✓ Stay on top</li> <li>✓ Show log window</li> <li>Generate logfile</li> <li>Cache reads</li> <li>✓ Verify download</li> </ul>			
Log output: <u>C</u> lear log		I✓ Init regs on start			
SEGGER J-Link GDB Server V4.70	a	*			
JLinkARM.dll V4.70a (DLL compi	led 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 targetConnected to target Waiting for GDB connection					
14		+			
0 Bytes downloaded 1 JTAC	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\ ARMCM4\_STM32\_Olimex\_P405\_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.

## APPENDIX 4. PEAK CAN INSTALLATION

To download the PEAK PCAN package click here.



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 <u>http://www.feaser.com/openblt/</u>.

After modifying the bootloader, it needs to be rebuild and then programmed again into the internal flash memory of the microcontroller. Refer to chapter **Error! Reference source not found.** for instructions for this last part.

Rebuilding the bootloader is achieved by the following step: