

HANTUNE

Tuning, calibration and visualization



Version 2.0
14-Feb-17

User Manual

Includes a Quick start for first time users and detailed information about the HANTune features and functionality.

Preface

This manual is intended to satisfy both the users who only want the absolute basics to get them started, as well as the users who want more in-depth knowledge about HANtune.

During the development of HANtune we continuously strive towards a robust, easy to use and versatile solution. To paraphrase Albert Einstein: it should be made as simple as possible, but not simpler. We are open to suggestions and feedback when it comes to chances we have missed to further improve HANtune!

Contents

PREFACE	1
1 INTRODUCTION	3
2 QUICK START	4
2.1 Using HANtune for Tuning and Calibration.....	4
2.2 Using HANtune as a CAN-dashboard	5
3 PROJECT DATA	6
3.1 ASAP2 files.....	6
3.2 DBC files.....	6
3.3 Layouts	7
3.4 Calibrations.....	7
4 ASAP2 ELEMENTS.....	8
4.1 Parameter and Signal list.....	8
4.2 DAQ List info.....	8
4.3 DAQ lists	8
5 CAN ELEMENTS	9
5.1 CAN message list	9
5.2 CAN Messages	10
5.3 CAN Signals	10
6 CREATING A LAYOUT	11
6.1 Editors.....	11
6.1.1 MultiEditor	11
6.1.2 SliderEditor	11
6.1.3 ButtonEditor	12
6.1.4 RadioButtonEditor	12
6.2 Viewers.....	12
6.2.1 MultiViewer.....	12
6.2.2 DigitalViewer.....	12
6.2.3 BooleanViewer	12
6.2.4 BarViewer	12
6.2.5 GaugeViewer.....	13
6.2.6 ScopeViewer.....	13
6.2.7 MultiLedViewer	13
6.3 Markup windows	13
6.3.1 TextMarkup.....	13
6.3.2 ImageMarkup.....	13
6.4 Arranging HANtune windows	14
7 COMMUNICATION	16
7.1 Communication Settings	16
7.2 XCP on CAN.....	16
7.3 XCP on Ethernet.....	17
7.4 XCP on USB/UART	17
7.5 CAN-only	17
7.6 The Connection Dialog	18
8 LOGGING	19
9 ERROR MONITORING	20
10 TROUBLESHOOTING	21
10.1 HANtune cannot start or hangs at the splash screen.....	21
10.2 HANtune has crashed or stalled	21
10.3 HANtune cannot connect.....	22
11 LIST OF REFERENCES	24

1 Introduction

This is the manual for the HANtune monitoring and tuning software. It provides basic information about the program and how to use it.

HANtune is a real-time software application for Windows, written in Java. It can be used as a generic CAN-dashboard, as well as a tuning and calibration tool using the XCP protocol. HANtune is capable of:

- ✓ Automated recognition of parameters from generated code using the ASAP2 file, and present these in a list for usage. These can be constants (parameters) or variables (signals).
- ✓ Automated recognition of CAN messages and signals from a dbc file, and present these in a list for usage. These can be used for both visualizing and sending CAN-messages.
- ✓ Visualizing parameters and CAN-messages by using “viewers” and adjusting parameters and sending CAN-Messages by using “editors”.
- ✓ Logging the received data to a log file. (.csv file)
- ✓ Displaying active errors (occurred at runtime) and stored errors (all errors) with an option to delete or log them.

Chapter 2: Quick start aims at the HANtune users who want to get up and running with the absolute minimum, without being bothered by any other details. It may also be very convenient to first go through this chapter first, before diving into the details provided by the other chapters.

Chapter 3 and further will provide in-depth details about most features of HANtune. To keep the size and complexity of this manual manageable, some less relevant details are not discussed.

2 Quick start

HANTune can be used for tuning and calibration, using the XCP protocol or as a generic CAN-dashboard, therefore this quick-start contains two sections: one for tuning and calibration and one for HANTune as CAN-dashboard. This section is intended for first time users. Follow these steps to create a basic project.

2.1 Using HANTune for Tuning and Calibration

For tuning and calibration, using the XCP protocol, the following two elements are needed:

An ASAP2 File (.a2l) which matches the program in the ECU. This could be a file which is automatically generated after compiling your Simulink model or one which is provided elsewhere. (following the ASAM standard) This describes all XCP related parameters and signals.

- In the side-tab “Project Data” there is a folder named “ASAP2 files”. Right click this folder and choose “Add ASAP2 file”
- Browse to the ASAP2 (*.a2l) file which you want to use, select it and press “Open”
- The file has now been added to your project but is not loaded. To load, double click on the ASAP2 file which you want to load or right click and choose “load file”

(For more info go to: Chapter 3.1 how to add an ASAP2 file)

A Layout which contains tabs with editors and/or viewers. A default Layout called “Unnamed” is already available and loaded. You can find the parameters and signals under the ASAP2 elements side-tab.

Editors: can be used to send parameters to the target

Viewers: can be used to visualize signals from the target

To create an editor (change parameters):

- Go to the tab “ASAP2 Elements”
- Search for the parameter which needs to be changed
- Drag-and-drop the parameter to your layout (empty white space)¹
- Choose the right type of editor

For more info on how to use editors see chapter **Error! Reference source not found.**

To create a viewer (visualize signals):

- Go to the tab “ASAP2 Elements”
- Search for the signal which needs to be showed
- Drag-and-drop the signal to your layout (empty white space)¹
- Choose the right type of viewer

For more info on how to use viewers see chapter **Error! Reference source not found.**

You are now ready to connect to the target. Check the communication settings by clicking the menu item “Communication” (top of screen), “Communication settings”. If these settings are correct and your target is powered on and connected to the PC, press F5 and click “Connect and request” to connect.

For more info on establishing a connection see: chapter 0 how to make a connection with a controller

To save your project: click “File”, “Save Project as...” in the menu. Choose a filename and the path where to save your project and click “Save”. Your project will be saved as a .html file.

¹⁾ Try to use multi-select! Some editors and viewers are capable of showing more than just one signal or parameter.

2.2 Using HANtune as a CAN-dashboard

HANtune can be used to present CAN-traffic in a human readable format. That means that the raw CAN-data is converted to physical values and presented along with the appropriate name and unit. To be able to do this, a descriptive file is needed which contains this information: the DBC file.

These files have the following convention:

CAN-Message: The entire message frame, belonging to one CAN-ID

CAN-Signal: One CAN-Message contains one or more signals. A signal has a unit and name, e.g.
oilTemp [°C]

To start using HANtune as a CAN-dashboard, the following two elements are needed:

A CAN database file (DBC) which matches the messages on the CAN-bus. This file with the .dbc file extension following the .dbc (quasi-) standard. This file defines the identifier, name, scaling, offset, etc. of CAN messages, enabling HANtune to present the raw CAN-data in a human readable format.

- In the side-tab “Project Data” there is a folder named “DBC files”. Right click this folder and choose “Add DBC file”
- Browse to the DBC (*.dbc) file which you want to use, select it and press “Open”
- The file has now been added to your project but is not loaded. To load, double click on the DBC file which you want to load or right click and choose “load file”

(For more info go to: Chapter xx how to add a DBC file)

A Layout which contains tabs with editors and/or viewers. A default Layout called “Unnamed” is already available and loaded. You can find the CAN messages in the “CAN elements” side tab.

Editors: can be used to send CAN-messages to the CAN-bus

Viewers: can be used to visualize CAN-messages from the CAN-bus

To create an editor (send CAN-messages):

- Go to the tab “CAN Elements”
- Search for the CAN-signal which needs to be sent
- Drag-and-drop the CAN-signal to your layout (empty white space)¹
- Choose the right type of editor

For more info on how to use editors see chapter **Error! Reference source not found.**

To create a viewer (visualize CAN-messages):

- Go to the tab “CAN elements”
- Search for the CAN-signal which needs to be showed
- Drag-and-drop the CAN-signal to your layout (empty white space)¹
- Choose the right type of viewer

For more info on how to use viewers see chapter **Error! Reference source not found.**

You are now ready to connect to the target. Check the communication settings by clicking the menu item “Communication” (top of screen), “Communication settings”. If these settings are correct and your target is powered on and connected to the PC, press F5 and click “Connect and request” to connect.

For more info on establishing a connection see: chapter 0 how to make a connection with a controller

To save your project: click “File”, “Save Project as...” in the menu. Choose a filename and the path where to save your project and click “Save”. Your project will be saved as a .html file.

¹) Try to use multi-select! Some editors and viewers are capable of showing more than just one signal or parameter.

3 Project data

The project data is located in the window belonging to the side-tab named “Project data”. It contains the descriptive files necessary for XCP or generic CAN communication, the DAQ lists, Layouts and Calibrations. Detailed information about these topics will be provided in this chapter.

Before HANTune can actively be used, at least one ASAP2 file or DBC file should be loaded, and at least one Layout should be loaded (the latter is automatically handled by HANTune). After saving the project, HANTune will remember the status and location of all the files which are added to the project, so it can be used right away upon next startup.

It is possible to actively use XCP on CAN together with DBC files. If this is desired, please choose “XCP on CAN in the communication settings dialog. HANTune supports both CAN2.0a (standard ID's) and CAN2.0b (extended ID's) simultaneously.

3.1 ASAP2 files

An ASAP2 file (.a2l) contains all the data which is needed by HANTune to properly use the parameters of a controller. It can be created when the code for a controller is generated. An ASAP2 file needs to “match” the controller code or else the parameters will not show up properly.

USAGE:

A new or existing project has to be opened. Right click on “ASAP2 files” and select “add ASAP2 file”. Navigate to the location where it is stored and open it. After this right click the file and select “load file”. If you wish to delete it select “remove file”. Multiple ASAP2 files may be added to your project but only one ASAP2 file can be loaded at the same time.

3.2 DBC files

A DBC file (.dbc) contains all the data which is needed by HANTune to properly parse the raw CAN-data to a physical value and vice-versa. It contains a list with which links the received messages to a name, scaling, offset, etc. A DBC file can be created using for example the KVASER Database Editor² or Vector's CANdb++³.

USAGE:

A new or existing project has to be opened. Right click on “DBC files” and select “add DBC file”. Navigate to the location where it is stored and open it. After this right click the file and select “load file”. If you wish to delete it select “remove file”. Multiple DBC files can be loaded and used at the same time.

² The KVASER Database editor is available for free and can be downloaded from: www.kvaser.com/downloads-kvaser

³ Vector's CANdb++ editor is available for free and can be downloaded from: https://vector.com/vi_downloadcenter_en.html

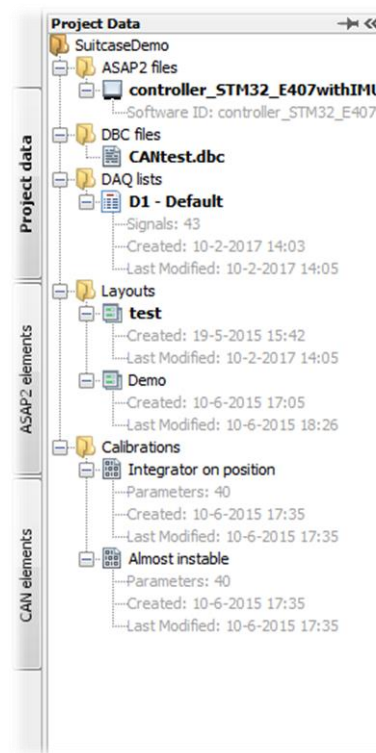


IMAGE 1: PROJECT DATA SIDE-TAB

3.3 Layouts

A layout contains all the viewers and editors used for visualizing and adjusting parameters. It can contain multiple tabs. Multiple Layouts may be added to your project but only one Layout can be loaded at the same time. A default layout titled “Unnamed” will be present and loaded upon starting a new project.

USAGE:

Right click “Layouts” and select “new layout” and give it a name. After this it needs to be loaded, right click it and select “load layout”. Other options are: Rename, Copy and Remove. If you wish to add more tabs, click “Window” in the top menu and select “new tab”. Tabs can also be renamed or deleted by right clicking the label of the tab.

3.4 Calibrations

Parameters which have been changed by HANTune are stored in the target’s RAM memory and will therefore be lost after a power-cycle of the target.

One option is to manually re-enter every parameter. A better option would be to save the current status of all active parameters into a calibration. HANTune checks which parameters are “active”, i.e. present in an editor which resides in a currently loaded layout, and stores the respective values in a calibration. After a power-cycle, or connecting another target with the same application software, this calibrated set of parameters can be uploaded with just two mouse clicks.

USAGE:

First a calibration should be created by right clicking the folder “Calibrations” and select “New calibration”. Provide a name for the calibration.

To save the current values of the parameters into a calibration:

- Right click the calibration where it needs to be saved
- Choose “Update Calibration” in the context menu
- Confirm the action

To load the current calibration into the target⁴:

- Right click the calibration which should be loaded into the target
- Choose “Load Calibration” in the context menu
- Confirm the action

Other options are: Rename, Copy, Remove and Export to MATLAB. Export to MATLAB creates an .m file which can be used to load the parameters into MATLAB.

⁴) HANTune can only write the parameter values to the target’s RAM memory, meaning these values will be reset to their default values after a power-cycle. Currently new features are under development which enables HANTune to create an export file which can be flashed into the target. This export file merges the calibration with the initial targets’ application, enabling users to create fully calibrated distributions of the targets’ application.

4 ASAP2 elements

THE ASAP2 ELEMENTS SIDE TAB CONTAINS THREE SECTIONS: THE LIST CONTAINING PARAMETERS AND SIGNALS, THE DAQ LIST INFO SECTION AND THE DAQ LIST SECTION.

4.1 Parameter and Signal list

This section shows the list of parameters and signals belonging to the currently loaded ASAP2 file (see “Project Data” tab).

Multi Select

Multiple Parameters or Signals can be selected to subsequently be dragged and dropped into the layout. To do so, hold the Ctrl key and select the Parameters or Signals needed. To select a range, hold Shift and click the first and last element needed.

Hint: drag and drop also works to add signals/parameters to an already existing viewer/editor.

Also multi-select and right clicking the selection is an option to quickly add a lot of parameters/signals to a viewer in the layout.

4.2 DAQ List info

This section shows a list of all DAQlists with their respective name, prescaler, the calculated frequency and a checkbox to (de)activate.

This is the place where the DAQlist frequency can be influenced by changing the prescaler. It adheres to the following rule:

$$DAQ_{frequency} = \frac{TargetApp_{freq}}{Prescaler}$$

Where $TargetApp_{freq}$ stands for the frequency with which the targets' application runs a complete application cycle.

4.3 DAQ lists

This section displays a full list of signals currently available in the loaded ASAP2 file. Note that the default DAQlist (D1) always contains the entire set of signals. Every other manually created DAQlist can be edited freely.

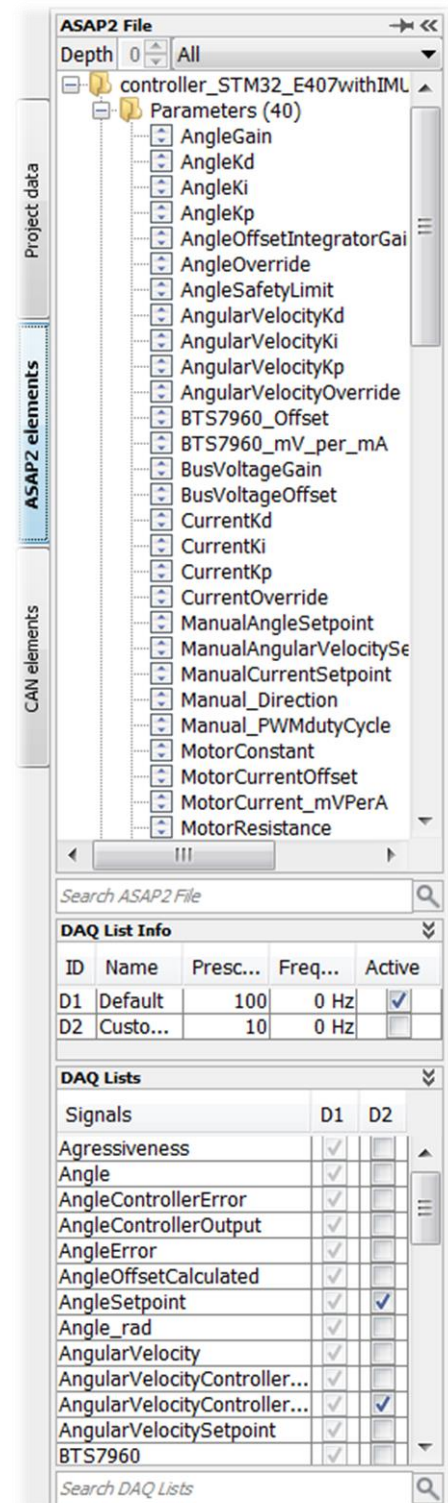


IMAGE 2: ASAP2 DATA SIDE-TAB

5 CAN elements

Much like the ASAP2 elements side tab, the CAN elements side tab also contains three sections. These include a CAN-Message list, a section “CAN Messages” and a section “CAN Signals”.

Like mentioned earlier, the convention here is that a CAN Message always has a CAN-ID and entails one entire message frame. The CAN-Message (or message frame) contains one or multiple CAN-Signals.

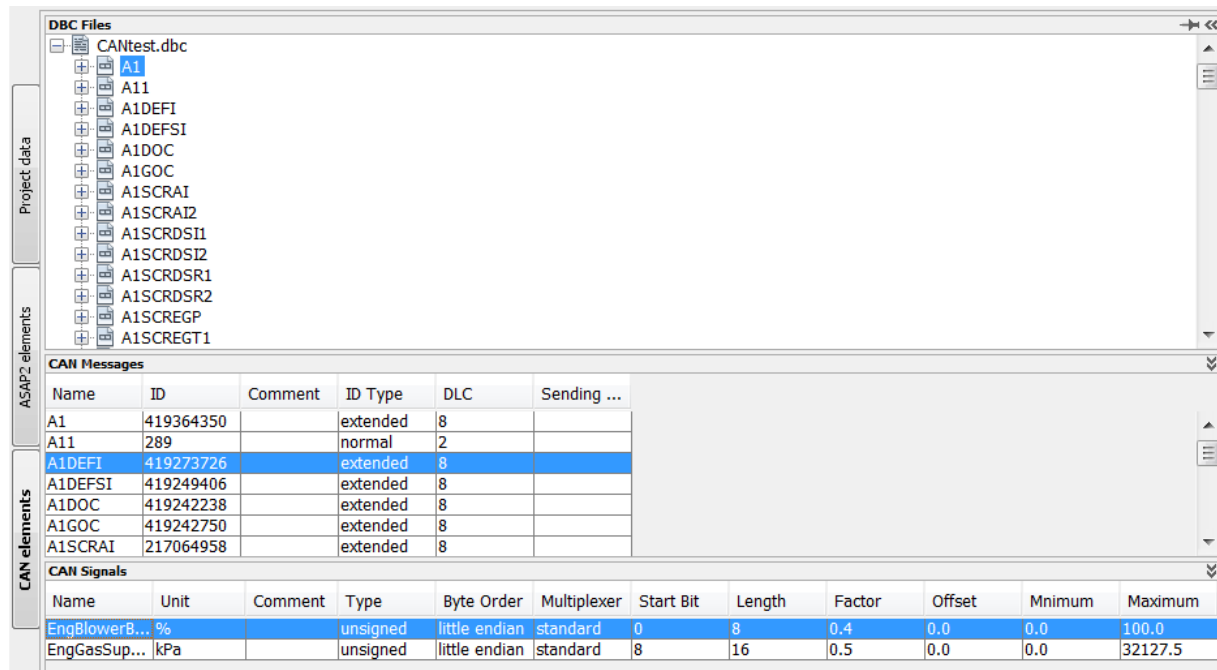


IMAGE 3: CAN ELEMENTS SIDE-TAB

5.1 CAN message list

The top-level node (CANtest.dbc in this example) shows the loaded DBC files. Multiple files may be loaded at the same time.

The next level contains the CAN Messages. If these nodes are expanded the CAN Signals become visible.

Also Multi-Select can be used to put CAN signals into viewers or editors.

Unlike in the XCP protocol, the data direction (sending or receiving) is not predefined in generic CAN messages. That means that the same signal can be used in both an editor and a viewer. To avoid mistakes, the signals' icon in the side panel will be updated to reflect the current use of the signal:



Neither added to a viewer or editor

Added to one or more editors

Added to one or more viewers

Added to one or more editors AND one or more viewers

TABLE 1: CAN ELEMENTS ICON INTERPRETATION

5.2 CAN Messages

The CAN-Messages section shows information about the messages in the currently loaded DBC file(s). Select a message in the message list and it will also be highlighted in the CAN Message section.

Please note that the ID is being shown decimally.

Name	ID	Comment	ID Type	DLC	Sending ...
A1	419364350		extended	8	
A11	289		normal	2	
A1DEFI	419273726		extended	8	
A1DEFSI	419249406		extended	8	
A1DOC	419242238		extended	8	
A1GOC	419242750		extended	8	
A1SCRAI	217064958		extended	8	

IMAGE 4: CAN MESSAGES SECTION

5.3 CAN Signals

The CAN Signals section contains a list of CAN signals belonging to the currently selected CAN Message. It can be used to get more information about the contents of CAN Messages and also shows all the details which are also available in the DBC file, however in a much more user-friendly way of presenting.

Name	Unit	Comment	Type	Byte Order	Multiplexer	Start Bit	Length	Factor	Offset	Minimum	Maximum
EngBlowerB...	%		unsigned	little endian	standard	0	8	0.4	0.0	0.0	100.0
EngGasSup...	kPa		unsigned	little endian	standard	8	16	0.5	0.0	0.0	32127.5

IMAGE 5: CAN SIGNALS SECTION

6 Creating a layout

Editors, Viewers and Markup Windows can all be grouped to be part of HANtune windows. All HANtune windows share some common features. All these window will contain an item “Window” in the right mouse button context menu, enabling the user to change the appearance of that particular Editor, Viewer or Markup Window.

For the ease of convention the HANtune window descendants are displayed in Image 6: HANtune Window Hierarchy.

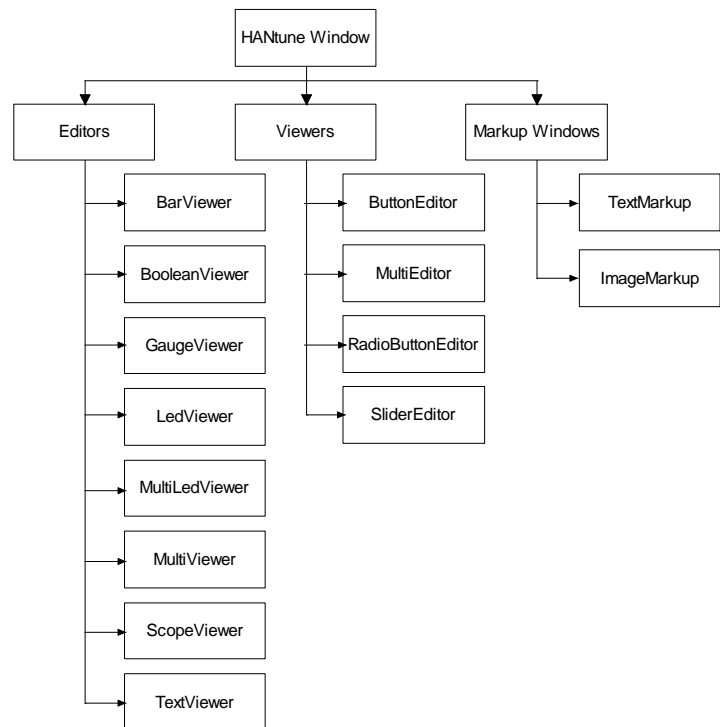


IMAGE 6: HANTUNE WINDOW HIERARCHY

6.1 Editors

Editors are HANtune windows which can be used to send data. It will display the data it has sent last. Unlike viewers, Editors will not update periodically but rather send their data after a user command (e.g. pressing the “enter” key).

Editors can be created by either a right-click or a drag-and-drop action on an XCP Parameter or a DBC CAN signal.

Multiple parameters can be added or removed from one editor by using the right-mouse-button context menu, or drag-and-drop other parameter(s) or CAN-signals(s) on the editor. Please note that not all editors are capable of handling multiple parameters or signals.

6.1.1 MultiEditor

A MultiEditor can be used to adjust parameter values. This is the most basic editor in HANtune. Values can be adjusted by using the two little arrow keys or by directly typing in the text box and pressing enter.



IMAGE 7: MULTIEDITOR

The MultiEditor is capable of handling multiple parameters or CAN-signals.

6.1.2 SliderEditor

A SliderEditor can be used to adjust parameter values. Values can be adjusted by using the slider with either the arrow keys or grabbing it with the mouse. Sliding the slider will cause the new values to be sent with at most 10Hz.



IMAGE 8: SLIDEREDITOR

The SliderEditor is capable of handling multiple parameters or CAN-signals.

6.1.3 ButtonEditor

The ButtonEditor needs to be configured first before it can be used. The “active” (button pressed) and “inactive” (button unpressed) values and texts need to be defined first. This editor will show the value which has been sent last to the left of the button.



IMAGE 9: BUTTONEDITOR

The ButtonEditor is capable of handling multiple Parameters or CAN-Signals.

6.1.4 RadioButtonEditor

The RadioButtonEditor needs to be configured first before it can be used. Each radiobutton needs to get a value assigned. Optionally the text can be changed as well. When connected, it will show the value which has been sent last to the left of the radiobutton.



IMAGE 10: RADIOBUTTONEDITOR

The ButtonEditor is capable of handling multiple Parameters or CAN-Signals.

6.2 Viewers

Under the header “Signals” are all the signals as described in the ASAP2 file. For each signal different viewers can be selected to visualize them. Right clicking a signal shows a list with available viewers. Some viewers have multiple options which can be accessed by right clicking on the viewer.

Multiple signals can be added or removed from one viewer by using the menu.

6.2.1 MultiViewer

A MultiViewer can display multiple signals. The visualization type depends on the type of parameter.

6.2.2 DigitalViewer

A DigitalViewer can display values as numbers. It can display Decimal, Hexadecimal or Binary values. The number of decimals to display can be adjusted.

6.2.3 BooleanViewer

A Boolean viewer can only display two values. The color can be adjusted and also the method of triggering: Turn on above or below a limit.

6.2.4 BarViewer

A BarViewer displays values by filling a bar, it also show the exact value below the bar. Options are: adjusting the upper limit, lower limit, the color and the use of sub ranges.

6.2.5 GaugeViewer

A GaugeViewer displays values by using a dial. Options are: adjusting the upper limit, lower limit, the color and the use of sub ranges.

6.2.6 ScopeViewer

A ScopeViewer displays a value at a certain time by using a graph. Multiple signals can be displayed in a single scope. Options are: adjusting the lower limit, the upper limit, the interval time, the use of auto scaling and a hold option which then can be used to save an image of the scope.

6.2.7 MultiLedViewer

A MultiLedViewer uses a led and a numeric representation to display a value. It has the option to adjust the colors and the limit of the led's.

6.3 Markup windows

6.3.1 TextMarkup

A TextMarkup is neither an editor nor a true viewer as it is not linked to an ASAP2 reference. It is basically a window which can be used to show additional markup information on a tab.

TextMarkup can be added by using the “Window menu” or by pressing Ctrl+L.

You can type text inside the window and by using the right mouse click menu you can use different options.

Options:

Modify font (the font of the entire window changes)

Modify font color (the font color of the entire window changes)

Toggle frame (the window is not resizable after removing the frame)

Toggle 1 px border

Toggle background (turns on a white background, default is transparent)

Send to back (useful when using multiple TextMarkup windows on top of each other)

Notes: When a TextMarkup overlaps a viewer or editor you cannot select it anymore.

6.3.2 ImageMarkup

ImageMarkup is a window in which an image can be displayed. It is always displayed behind other components.

Usage:

Right click inside the window and select “Choose an Image”. A file chooser will open for selecting an image. Only png, jpg, jpeg and bmp file types are allowed.

Images fill the entire window and can be resized by resizing the window.

There is an option for scaling the images which will keep the aspect ratio of the image and it can also be used for returning to the original dimensions.

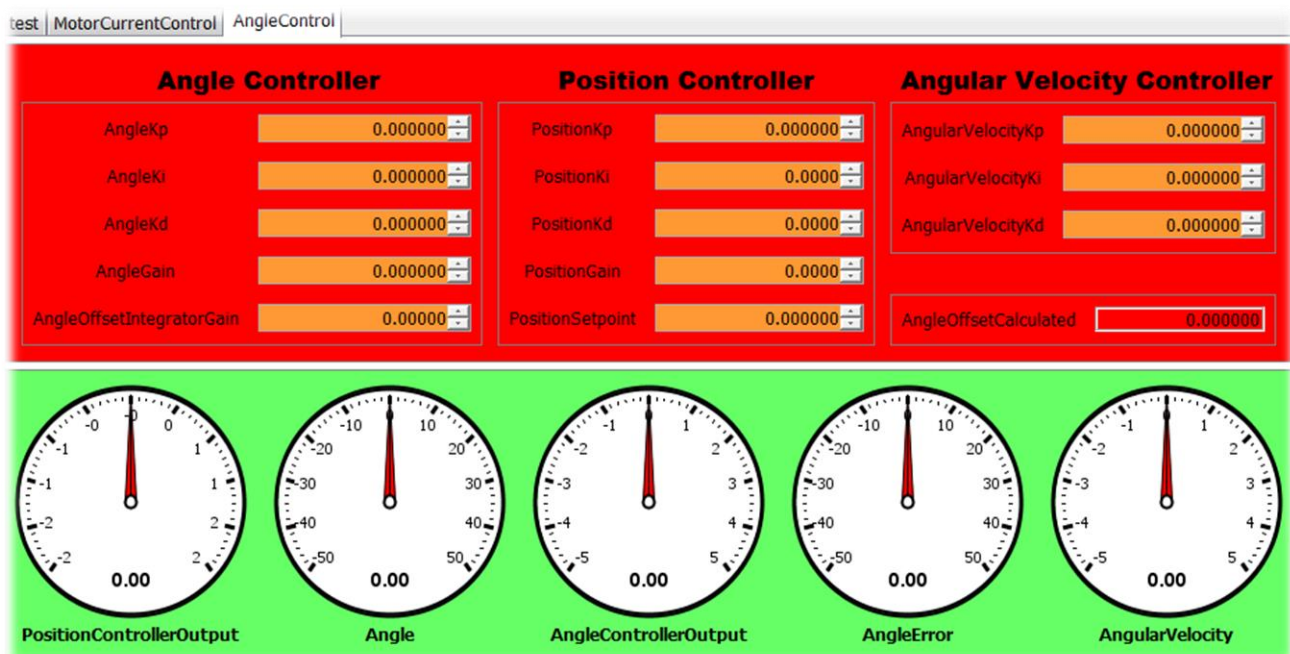
It is advisable to keep the images in a folder with the project file or in the images folder with HANTune to prevent loss of images on layouts.

6.4 Arranging HANtune windows

The main screen of HANtune consists of windows, these can be editors, viewers or markup windows. These windows have the same layering principle as the objects in Microsoft Office. In the right mouse button context menu of all HANtune windows a “Window” menu item can be found. This contains all generic HANtune window functionalities, like ordering the windows above or under each other. This is particularly useful when using the markup windows. HANtune has three main layers: background, default and foreground (“Move to Layer” option). Within each layer, the HANtune windows can be put forwards or backwards relative to each other, within the same layer (Move to front/back option).

1.1.1 Move to front/back

In the following example demonstrates how markup windows and layering could be used. A textMarkup window is used as a colored background, causing the visual effect of a “container”. Other textMarkup windows containing the bold texts are placed on top of it, indicating the groups of editors. Also the editors themselves are placed on top the same colored textMarkup windows. (similar to the gauges)



USAGE:

The above example is used to explain the usage of layers and markup windows.

Create the red background

In the top of HANtune, click “Window” → “New TextMarkup”. A new textMarkup window will be added to the active HANtune layout.

Press ctrl+R to enable the “Resize and Move” mode and position and size the window. Press ctrl+R again to return to normal mode.

Right click the window, choose “Window” → “Background” and check the item “Visible”

Right click the window, choose “Window” → “Background” → “Color” to set the background color.

Add texts to the red background

In the top of HANTune, click “Window” → “New TextMarkup”. A new textMarkup window will be added to the active HANTune layout.

Press ctrl+R to enable the “Resize and Move” mode and position and size the window to where the text should appear. Press ctrl+R again to return to normal mode.

Right click the window again, choose “Modify Text”. Enter the text to be shown. Note: “Enter” will add a new line. To stop editing, click anywhere else in HANTune. (Window loses focus)

If desired, the right click context menu can also be used to modify the text color and font.

Add the editors

Drag and drop a parameter or CAN-signal to the layout to create an editor.

Use ctrl+R (Resize and move mode) to position and size the editor

To show the background color through the editor, right click the editor, choose “Window” → “Background” and deselect “Visible”.

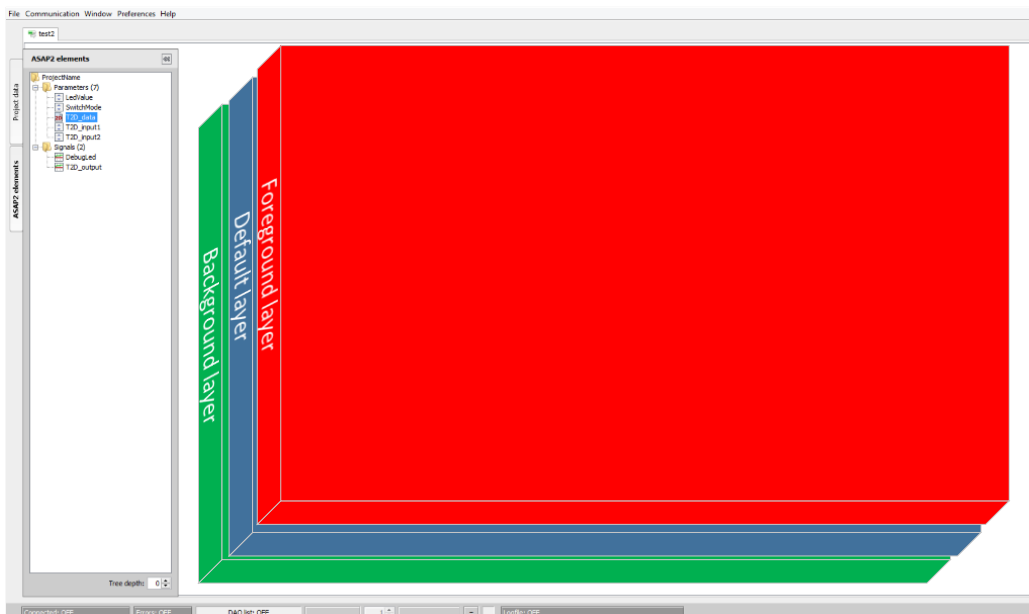
Arrange the windows in the right order

Right click the editor which should be moved to the front or back

Choose “Window” → “Move” and use “Move to front (current layer)” or “Move to back (current layer)” to arrange the windows in the right order.

1.1.2 Layers

Additional to the arrangement relative to other windows, HANTune also has a background and foreground layer. Components located in the foreground layer will always appear on top of windows residing in the default layer or the background layer. To move a window to a specific layer, right click that window, choose “Window” → “Move” → “Move to Layer”. The following image visualizes the principle:



7 Communication

HANTune supports various ways of communicating. With regards to hardware support there are three options: Ethernet, USB and CAN. Ethernet and USB/UART can only be used in combination with the XCP (universal Calibration Protocol). The CAN-bus can be used either with or without XCP functionalities. This chapter provides more information about the (combination of) these options, as well as the general communication settings in HANTune.

7.1 Communication Settings

Image 11: Communication Settings Dialog shows a screenshot of the communication settings in HANTune. It can be found under the menu item “Communication” → “Communication Settings” and in the connection dialog by clicking the button “Settings”.

In the tab “General” the general communication settings have to be set. A choice has to be made for a communication protocol.

The timeout prescaler is used to define the communication time-out. If HANTune doesn't receive any messages anymore, by default it cancels the connection after $2 * t_{prescaler}$. Please note that this is a temporary solution. It is planned that in a future version of HANTune the behavior of the timeout prescaler will be depending on the expected delay time of incoming messages.

The other tabs contain hardware-specific settings for CAN, Ethernet and USB, respectively.

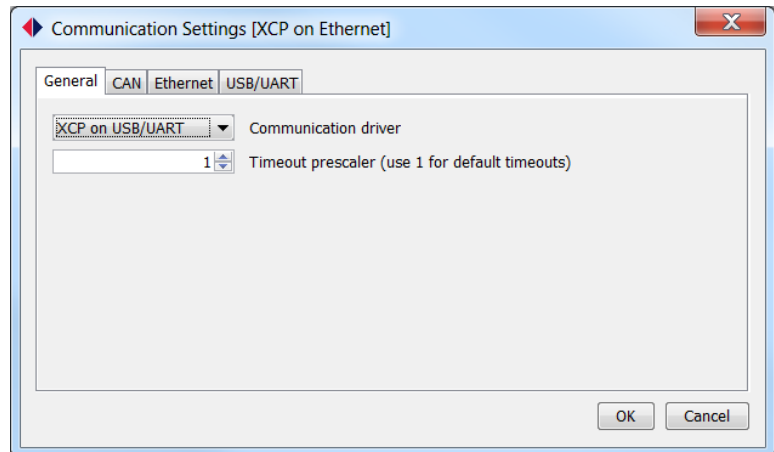


IMAGE 11: COMMUNICATION SETTINGS DIALOG

7.2 XCP on CAN

To use the CAN functionality of HANTune, a CAN-dongle is needed. HANTune contains drivers for the PCAN-USB dongle. More info about the USB to CAN dongle can be found on this website: www.peak-system.com/PCAN-USB.199.0.html

The XCP on CAN protocol uses two CAN-ID's, one for sending XCP messages and one for receiving them. By default HANTune uses 665 for transmission and 666 for reception. Additionally the baud-rate needs to be set in order to be able to establish a connection with the target.

If there is a DBC file available in HANTune, the XCP on CAN option also provides the option to include generic CAN messages (not part of the XCP traffic).

CAUTION: PLEASE VERIFY THAT THE DBC FILE DOES NOT INTERFERE WITH THE RECEPTION AND TRANSMISSION ID'S OF THE XCP TRAFFIC.

Please note that these default settings match the default settings of the HANcoder blockset.

For help, please refer to chapter 10: Troubleshooting.

7.3 XCP on Ethernet

To use XCP on Ethernet, a connection should be established between the PC and the target. While the options are not limited to a direct UTP connection between PC and target, this manual only covers the direct wired link.

Additional information about other connection options can be requested by sending an email to: hantune@han.nl.

HANcoder-built targets have a default IP address similar to the one in HANTune: 169.254.19.63. Please make sure that any wireless networks are set inactive when plugging in the Ethernet cable. The communication driver should be set to “XCP on Ethernet” in the tab “General”. Also the IP address should be set to the right address. Then a connection can be established.

Please note that these default settings match the default settings of the HANcoder blockset.

For help, please refer to chapter 10: Troubleshooting.

7.4 XCP on USB/UART

While HANTune is capable of communicating via any COM-Port, this scope of this chapter is limited to the use of a HANcoder-built target.

When using targets built with the HANcoder blockset, HANTune depends on a virtual COM port to be able to communicate to the target. It is being provided with the HANcoder blockset, which can be downloaded from: www.openmbd.com. While most PC's don't support a physical UART connection anymore, UART to USB converters are freely available. The Olimex boards supported by HANcoder even have the hardware on board, eliminating the use of converters.

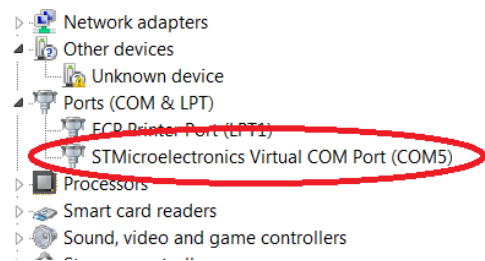


IMAGE 12: DEVICE MANAGER COM PORT

When the virtual com port device driver has been installed, Windows automatically assigns a com port to the plugged-in device. To find the assigned COM-port, go to the windows Start menu, type “Device Manager” (Dutch: “Apparaatbeheer”), and press enter. The device manager pops up. Under “Ports (COM & LPT)” a device should be visible with the name “STMicroelectronics Virtual COM Port (COMx)”. Remember the assigned COM-Port.

Go to the “Communication Settings” dialog, select the “XCP on USB/UART” options in the tab “General”. Now go to the tab “USB/UART” and set the right COM-Port. HANTune is now set to connect to the target.

Please note that these default settings match the default settings of the HANcoder blockset.

For help, please refer to chapter 10: Troubleshooting.

7.5 CAN-only

HANTune can also be used without the use of the XCP protocol. When a DBC file is available, it can be loaded via the “Project Data” side tab. When loaded, the CAN messages and signals which are contained by the DBC file will become visible in the CAN Elements side tab, ready to be used in editors and viewers.

To connect using only CAN, the “CAN” option should be chosen in the Communication Settings, under the tab “General”. Furthermore the baud rate should be set to match the bus's baud rate. The XCP on CAN settings can be neglected, as they aren't used.

The viewers will be updated upon receiving a new value. The editors will only send a value upon user input. This means that a time-triggered mode is (as of today) not available in HANTune.

For help, please refer to chapter 10: Troubleshooting.

7.6 The Connection Dialog

A connection can be established by pressing the hot-key F5, by clicking the menu item “Communication” → “Connect” or clicking the connect button at the lower left side of HANtune (with the text “Connected: off”). This will cause the connection dialog to pop up. (see Image 13: Connection Dialog)

Two types of connection are available when connecting via the XCP protocol: Request Parameters and Calibrate Parameters. This is what they do:

Request Parameters

During the establishment of the connection, HANtune will request the values of all parameters from the target to display them in the editors. No parameter will be changed on the target-side during the connection procedure.

Calibrate Parameters

This type of connection will first connect to the target and immediately thereafter sends the currently active values of the editors to the target to calibrate. It will overwrite the target’s current parameter values.

The option for error monitoring will be further explained in chapter 9: PrefaceError monitoring.

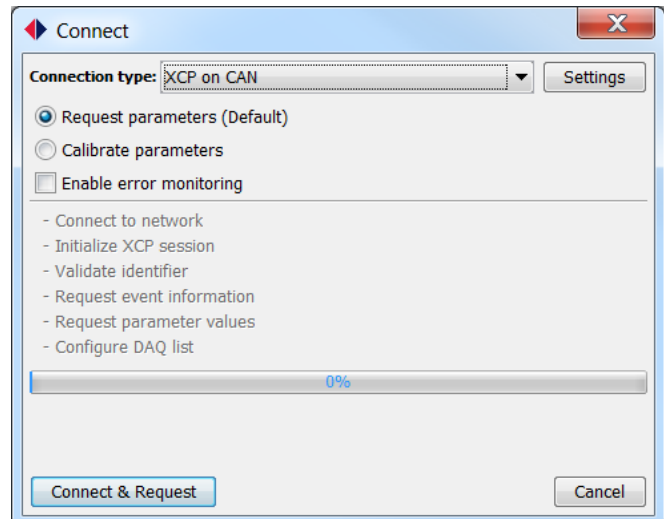


IMAGE 13: CONNECTION DIALOG

8 Logging

HANTune has the option to log parameter values to a .csv file.

To use logging enable it by going to the “Communication menu” and select “Enable Datalogging” or click on the log file button in the bottom status bar. The name of the log file can partially be changed by selecting “Modify Datalog Filename”.

The log file will be saved in the “log” folder in the HANTune folder.

LOGFILE:

A log file consists of two parts: a header and log values. The header contains additional information as:

- The name of the project
- The name of the ECU
- The sample frequency
- The time and date of the log file

9 ERROR MONITORING

HANTune has an option to monitor errors that occur during the current session or have occurred during an earlier session. Please note that error monitoring is a feature provided by the HANcoder library.

STATUS INDICATOR:

In the status bar there is an indicator named: Errors. This shows the current state of the error monitoring process and the numbers of errors which are stored in the controller memory. The indicator uses the following colors:

- Gray: Error monitoring is not running
- Green: Error monitoring is running and there are no errors in the controller
- Orange: There is an old error present in the memory
- Red: There is an error present which occurred during the current session

ERRORVIEWER:

The ErrorViewer can be used to display additional information about individual errors. The viewer consists of two tables: Active errors, which show the errors occurred during the current session and Stored, which are older errors. Each line represents one error and contains the following information:

- Code: shows the hex value of an error code
- Parameter: shows additional info about an error
- Occurrence: shows how many times an error occurred
- Timestamp: shows when the error occurred
- Info: shows the type of error

This information can be used to lookup additional information in the following recourses:

- | | |
|----------------------|-----------------------------------|
| Rexroth system error | => Bosch RC 30 series manual |
| RC30 Target error | => Error list.xlsx |
| Application error | => Possible project documentation |

10 Troubleshooting

This section provides the answers to the most commonly asked questions. When this section doesn't provide a solution, please contact the developer via hantune@han.nl, provide a description of what happened and make sure the errorLog.txt file is attached, which can be found in the HANTune folder. We will get back to your request as soon as possible. (Usually within one or two working days)

10.1 HANTune cannot start or hangs at the splash screen

This could be caused by an outdated version of the Java Runtime Environment (JRE) or not having one at all.

Please check which version you have installed currently, by following these steps:

- 1) Click the windows home button
- 2) Type "cmd" and press "Enter" (command prompt appears)
- 3) Type "java -version" (without the quotation marks) and press "Enter"

Make note of the message which appears in the command prompt.

In case the command prompt returns "Command not found": install the Java Runtime Environment by going to the following website and download and install the JRE: www.java.com/en/download/

In case the command prompt returns something like: `java version "1.8.0_101"`, please go to www.java.com/en/download/ and check whether you are running the most recent version of Java. If not, please download and install the latest version.

Issue not solved? Please contact us via hantune@han.nl

10.2 HANTune has crashed or stalled

Our policy during development and testing is to avoid all situations in which HANTune can get unresponsive or crashes. Unfortunately, not all situations can be tested and accounted for, so in case this has happened we would like to ask you to contact us via hantune@han.nl. Do not forget to add the errorLog.txt in your email attachment!

10.3 HANtune cannot connect

When HANtune cannot connect, usually it's wise to first check the power supply and correct wiring. Finally it's advised to use a debug LED (or blinking LED) to indicate that the target's software is running. If these three topics are okay, please proceed to the following hints:

XCP on CAN

When HANtune cannot connect via XCP on CAN, please check the following items:

- ✓ Are the communication settings correct?
Hint: Click the menu item "Communication" → "Communication Settings". The "XCP on CAN" option should be selected. Go to the tab "CAN" and check whether the baudrate and CAN ID's used for XCP communication are set according to the target's running application.
- ✓ Does the CAN-bus comply to the CAN-standard?
Hint: Check the resistance between CAN high and CAN low. It should be 60Ω. Not 600Ω? Check the termination resistors.
- ✓ Is the CAN-dongle fully functional?
Hint: Check if the dongle is working correctly using PCAN-View. (freeware)
- ✓ Does another application use the PCAN hardware driver?
Hint: Close all other application which might use the PCAN dongle, restart HANtune and reconnect.

XCP on Ethernet

When HANtune cannot connect via XCP on Ethernet, please check the following items:

- ✓ Are the communication settings correct?
Hint: Click the menu item "Communication" → "Communication Settings". The "XCP on Ethernet" option should be selected. Go to the tab "Ethernet" and check whether the IP address and destination port are set according to the target's running application.
- ✓ Is the target's IP address reachable?
Hint: Open a command prompt (windows home → type "cmd", press enter) and type "ping" followed by the target's IP address, for example: "ping 168.254.19.63". There should be a response. If not, please check if the IP address is correct, check if your application is running (blinking led?) and check whether the PC recognizes the Ethernet connection. (Ethernet connector icon in windows taskbar)
- ✓ Are the communication settings correct in HANtune?
Hint: Open the "Communication Settings" dialog, and check whether "XCP on Ethernet" is selected in the tab "General". Also check if the correct IP address is entered in the tab "Ethernet".
- ✓ Still no connection? Add HANtune to your firewall's exceptions.
Hint: Go to Windows Firewall (windows home, → type "windows firewall", press "Enter") and click "Allow a program or feature through Windows Firewall. Click "Allow another program" and browse to the HANtune executable.

Problem not solved? Please contact us via hantune@han.nl with a detailed description of the issue.

XCP on USB/UART

When HANTune cannot connect via XCP on USB, please check the following items:

- ✓ Are the communication settings correct?
Hint: Click the menu item "Communication" → "Communication Settings". The "XCP on USB/UART" option should be selected. Go to the tab "USB/UART" and check whether the baudrate is set according to the target's running application. Also the assigned COM port should be correct according to the device manager. See next item to check the COM port.
- ✓ Is the right COM port used?
Hint: Go to the Device Manager (windows home, → type "Device Manager" or "Apparaatbeheer", press "Enter") and click the item "Ports (COM & LPT)". A device called "STMicroElectronics Virtual COM Port" should be visible. The COM port is mentioned at the end of the line. Check whether the COM port corresponds to the one selected in HANTune. (Communication Settings, tab: USB/UART)
- ✓ Does the Virtual COM port driver recognize the device?
Hint: Go to the Device Manager (windows home, → type "Device Manager" or "Apparaatbeheer", press "Enter") and click the item "Ports (COM & LPT)". A device called "STMicroElectronics Virtual COM Port" should be visible. If not, check if the target's application is running (blinking LED?). If so, please shut down HANTune, unplug the target from the PC. Plug it in again and restart HANTune. Please refer to the Changelog_KnownIssues.txt file in the HANTune folder for more info on this topic.
- ✓ Using an Olimex E407?
Hint: Because there are two USB ports on the E407, and only one is used for XCP communication, it's an easy mistake to make. Please make sure to use the USB port closes to the Ethernet connector.

Problem not solved? Please contact us via hantune@han.nl with a detailed description of the issue.

CAN

When HANTune cannot connect via CAN, please check the following items:

- ✓ Are the communication settings correct?
Hint: Click the menu item "Communication" → "Communication Settings". The "CAN" option should be selected. Go to the tab "CAN" and check whether the baudrate is set according to the baudrate of the currently connected CAN-bus. Also, please note that HANTune currently only supports the PCAN-USB dongle to communicate via CAN. (The XCP on CAN settings can be ignored)
- ✓ Is the CAN-bus functioning correctly?
Hint: PCAN-View is available to download and use for free from www.peak-system.com. Use this application to check for CAN-bus related errors (e.g. bus light, bus heavy, etc.). Dedicated troubleshooting is available on the same website.
- ✓ Is another driver confiscating the PCAN-USB driver?
Hint: Some applications can confiscate the PCAN-USB driver. Please shut down all applications which might use the PCAN-USB driver and try to reconnect in HANTune.

Problem not solved? Please contact us via hantune@han.nl with a detailed description of the issue.

11 List of References

Images

Image 1: Project Data side-tab	6
Image 2: ASAP2 Data side-tab.....	8
Image 3: CAN Elements side-tab	9
Image 4: CAN Messages section	10
Image 5: CAN Signals section	10
Image 6: HANtune Window Hierarchy	11
Image 7: MultiEditor	11
Image 8: SliderEditor	11
Image 9: ButtonEditor	12
Image 10: RadioButtonEditor	12
Image 11: Communication Settings Dialog.....	16
Image 12: Device Manager COM port	17
Image 13: Connection Dialog.....	18

Tables

Table 1: CAN Elements Icon Interpretation	9
---	---