|  |  |
| --- | --- |
| 199.jpg | HANTUNE |
|  | BitsChipsRC30Target_HANtune.jpg |
| Version 1.1  3/20/2013 | User Manual |
|  | Includes a Quick start for first time users and general information about the different options and components. |

Copyright (c) 2020 [HAN University of Applied Sciences]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

HANTUNE

User Manual

Contents

[1 Introduction 2](#_Toc351551052)

[2 Quick start 3](#_Toc351551053)

[3 Project data 4](#_Toc351551054)

[3.1 ASAP2 files 4](#_Toc351551055)

[3.2 Layouts 4](#_Toc351551056)

[3.3 Calibrations 4](#_Toc351551057)

[4 Layers 5](#_Toc351551058)

[5 ASAP2 elements 6](#_Toc351551059)

[5.1 Parameters (editors) 6](#_Toc351551060)

[5.1.1 MultiEditor 6](#_Toc351551061)

[5.2.1 SliderEditor 6](#_Toc351551062)

[5.2 Signals (viewers) 6](#_Toc351551063)

[5.1.2 MultiViewer 6](#_Toc351551064)

[5.2.2 DigitalViewer 7](#_Toc351551065)

[5.3.2 BooleanViewer 7](#_Toc351551066)

[5.4.2 BarViewer 7](#_Toc351551067)

[5.5.2 GaugeViewer 8](#_Toc351551068)

[5.6.2 ScopeViewer 9](#_Toc351551069)

[5.7.2 MultiLedViewer 9](#_Toc351551070)

[5.3 TextMarkup 10](#_Toc351551071)

[5.4 ImageMarkup 11](#_Toc351551072)

[6 Communication 12](#_Toc351551073)

[7 Logging 13](#_Toc351551074)

[8 Error monitoring 14](#_Toc351551075)

# Introduction

This is the help file 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 uses the standardized XCP protocol for communication with an ECU. HANtune is capable of:

* Automated recognition of parameters from generated code, and present these in a list for usage. These can be constants or variables.
* Visualizing parameters by using “viewers” and adjusting parameters by using “editors”.
* Logging parameters to a log file.
* Displaying active errors (occurred at runtime) and stored errors (all errors) with an option to delete or log them.

# Quick start

This short guide is intended for first time users; follow it to create a basic project.

When you first open HANtune a new project is already waiting. To use it you need to add two components:

An ASAP2 File (.a2l) which matches the program in the ECU.

* See chapter 3.1how to add an ASAP2 file.

A Layout which contains tabs with multiple editors and/or viewers.

* See chapter 3.2 how to add a Layout.

To start using HANtune you will need editors and viewers. You can find them under the ASAP2 elements tab on the left side. There are two options: Parameters and Signals.

Right click a Parameter to add an editor to change parameter values.

* For more info on editors see chapter 4.1

Right click a Signal to add a viewer to visualize parameter values.

* For more info on viewers see chapter 4.2

After this you will need to connect to a supported controller.

* See chapter 5 how to make a connection with a controller.

Save the project.

* For additional information see the appropriate chapter in this manual.

# Project data

## 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”.

## Layouts

A layout contains all the viewers and editors used for visualizing and adjusting parameters. It can contain multiple tabs.

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.

## Calibrations

Calibrations are used to save certain parameter adjustments for later use.

Usage:

Right click “Calibrations” and select “New calibration”. After this it can be loaded by right clicking the calibration. 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.

# Layers

The main screen of HANtune consists of windows, these can be editors, viewers or markup windows. This screen is divided into three layers in which the different windows can be placed. When created the window is placed in the default layer, which is the middle of the three. There also is the foreground- and the background layer.



Figure - Layers in HANtune

Usage:

Right click the frame on one of the windows. A popup menu will become visible:

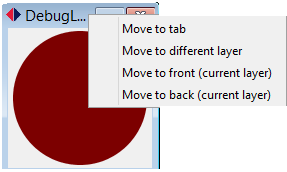


Figure 2 - Frame popup menu

Click on “move to different layer”. Then select the layer to which the window has to be moved. This can be the background layer, default layer or foreground layer.

In the popup menu it is also possible to click “Move to front (current layer)” and “Move to back (current layer)”. These will move the window to the front or the back of the layer it is in.

# ASAP2 elements

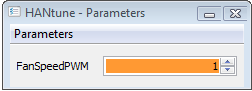
## Parameters (editors)

Under the header “Parameters” are all the parameters as described in the ASAP2 file. For each parameter different editors can be selected to adjust them. Right clicking a parameter show a list with available editors. Some editors have multiple options which can be accessed by right clicking on the editor.

Multiple parameters can be added or removed from one editor by using the menu.

### MultiEditor

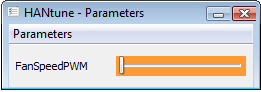
A MultiEditor can be used to adjust parameter values. Values can be adjusted by using the two little arrow keys or by directly typing in the text box and pressing enter.



**Figure 1 - MultiEditor**

### 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.



**Figure 2 - SliderEditor**

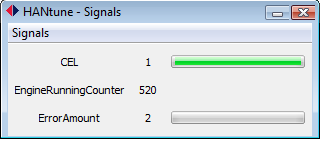
## Signals (viewers)

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.

### MultiViewer

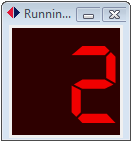
A MultiViewer can display multiple signals. The visualization type depends on the type of parameter.



**Figure 3 - MultiViewer**

### DigitalViewer

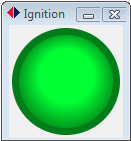
A DigitalViewer can display values as numbers. It can display Decimal, Hexadecimal or Binary values. The number of decimals to display can be adjusted.



**Figure 4 - DigitalViewer**

### BooleanViewer

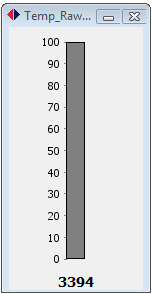
A Boolean viewer can only display two values. The color can be adjusted and also the method of triggering: Turn on above or below a limit.



**Figure 5 - BooleanViewer**

### BarViewer

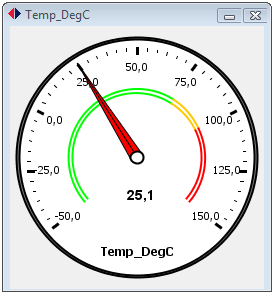
A BarViewer displays values by filling a bar, it also show the exact value below the bar. Options are: adjusting the upper limit, lower limit, the color and the use of sub ranges.



**Figure 6 - BarViewer**

### GaugeViewer

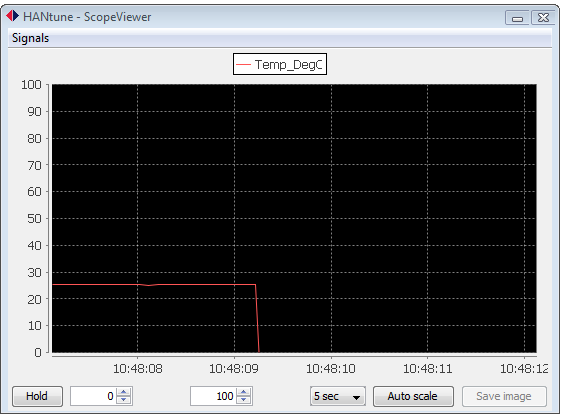
A GaugeViewer displays values by using a dial. Options are: adjusting the upper limit, lower limit, the color and the use of sub ranges.



**Figure 7 - GaugeViewer**

### ScopeViewer

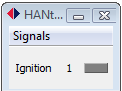
A ScopeViewer displays a value at a certain time by using a graph. Multiple signals can be displayed in a single scope. Options are: adjusting the lower limit, the upper limit, the interval time, the use of auto scaling and a hold option which then can be used to save an image of the scope.



**Figure 8 - ScopeViewer**

### 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.



**Figure 9 - MultiLedViewer**

## TextMarkup

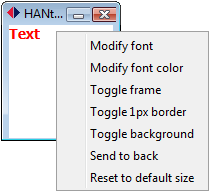
A TextMarkup is neither an editor nor a true viewer as it is not linked to an ASAP2 reference. It is basically a window which can be used to show additional markup information on a tab.

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)



**Figure 10 - TextMarkup**

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

## ImageMarkup

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

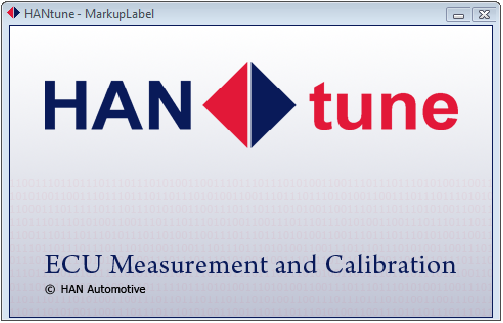
Usage:

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 sued 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.



**Figure 10 - TextMarkup**

# Communication

HANtune uses the XCP protocol to communicate with a controller. A communication with a controller can be established by either clicking on the connect button in the left bottom corner, by opening the “Communication menu” and selecting “Connect to XCP device” or by pressing F5.

After the connection dialog opened, connect by clicking “Connect & Request”. If you want to connect using calibrations, select “Connect & Calibrate”.

Options:

* The following values can be changed:
* Driver type (default PCAN BASIC)
* CAN identifier used for XCP transmission (default 665)
* CAN identifier used for XCP reception (default 666)
* Timeout prescaler (default 1)
* CAN baudrate (default 500)
* Start PCAN-View

Warning: Do not change these values if not needed.

# 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

# Error monitoring

HANtune has an option to monitor errors that occur during the current session or have occurred during an earlier session.

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 witch 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