|  |  |
| --- | --- |
| 199.jpg | HANTUNE |
|  | BitsChipsRC30Target_HANtune.jpg |
| Version 0.1  9/13/2012 | Release document |
|  | A guide for testing and releasing new versions of HANtune. |

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

Release document

Contents

[1 Introduction 2](#_Toc335300478)

[2 Test protocol hantune 3](#_Toc335300479)

[2.1 Developer test 3](#_Toc335300480)

[2.2 Alpha test 4](#_Toc335300481)

[2.3 Beta test 4](#_Toc335300482)

[3 release checklist hantune 5](#_Toc335300483)

[4 release guide 6](#_Toc335300484)

[4.1 Launch4j 6](#_Toc335300485)

[4.2 Release file list 7](#_Toc335300486)

# Introduction

This document is a developers guide for testing and releasing new versions of HANtune. It is created to ensure a smooth as possible release of new versions.

It contains a test protocol, checklists and a guide for releasing new versions of HANtune. Make sure all the steps are finished before releasing a new version.

# Test protocol hantune

The steps described here are used to test new code, or bug fixes. Do this at least for every release, even when changes seem minor, or the code that deals with a part of the functionality hasn’t changed. This protocol is lengthy, but should save time when making an alpha or beta release. Obviously, every new function or bug fix should be tested before they are committed and its ticket closed, but testing just the function itself is sufficient for this.

## Developer test

The developer test is done by the developer before a release. This should catch most of the bugs. If possible, do all the steps before fixing the found bugs.

|  |  |  |
| --- | --- | --- |
| Nr. | Description of the test | Passed/  failed |

|  |  |  |
| --- | --- | --- |
| 1 | Start HANtune in debug mode. See if all the menu buttons work correctly. |  |

|  |  |  |
| --- | --- | --- |
| 2 | Open the respective project (to be found in the Developer docs directory) and load (and unload) the asap2 and layout. If this works, connect (with the correct program flashed in the correct controller). |  |

|  |  |  |
| --- | --- | --- |
| 3 | See if the signals update. |  |

|  |  |  |
| --- | --- | --- |
| 4 | Change all the parameters in the MultiEditor a few times, see if the corresponding signals update correctly. Take a few near and over the limits of the data type and the specific parameter (if it has any) and see if this works correctly. |  |

|  |  |  |
| --- | --- | --- |
| 5 | Try to make a few tabs. |  |

|  |  |  |
| --- | --- | --- |
| 6 | Try all possible windows, one for each data type. See if they work correctly. |  |

|  |  |  |
| --- | --- | --- |
| 7 | Try saving the project under a different name. Close, re-open and connect. |  |

|  |  |  |
| --- | --- | --- |
| 8 | Try all options of every window. Try, for options that require user input, both inputs that are correct and incorrect. |  |

|  |  |  |
| --- | --- | --- |
| 9 | Try all the prescaler settings. See if lag and missing packets stay within reason (and if the indicators work). |  |

|  |  |  |
| --- | --- | --- |
| 10 | Load the calibrations. |  |

|  |  |  |
| --- | --- | --- |
| 11 | See if data logging works correctly. |  |

|  |  |  |
| --- | --- | --- |
| 12 | Repeat steps 2, 3, 4, 6, 9 and 10 with all the controllers and their respective projects. Take extra care to check if the order of tables in ertha is correct. |  |

|  |  |  |
| --- | --- | --- |
| 13 | Repeat steps 1 true 12 in 64 bit JRE (or 32 if you started with 64). |  |

## Alpha test

When the developer test is done, an alpha version can be released. Do this by following the release checklist, ignoring the documents and the tests. Don’t tell the alpha testers specifically what to test, rather give them a broad set of things (for instance, a controller). Be sure that your testers have a way to report bugs, for instance a Trac useracount that is only allowed to create new tickets.

Tell your testers to do what they would normally do, but also to test extreme cases and obviously wrong inputs. Try things you think would cause errors. When in the alpha testing period, only report bugs, don’t fix them.

## Beta test

The beta test is similar to the alpha test. If you have testers that were also part of the alpha test group, let them pay special attention to the bugs they found. If bugs are found after a beta test, they are to be fixed in a new release, not in the current one.

# release checklist hantune

Check this list before releasing a new HANtune version.

|  |  |  |
| --- | --- | --- |
| Nr. | Action | Check |

Check if the following documents are up-to-date:

|  |  |  |
| --- | --- | --- |
| 1 | Help file/videos |  |
| 2 | Copyright notices |  |
| 3 | Change list |  |
| 4 | Programmers guide |  |
| 5 | Release document |  |
| 6 | Getting started |  |

Check if the following is done/updated:

|  |  |  |
| --- | --- | --- |
| 1 | All functions javadoced |  |
| 2 | Version number (in hantune.java, float version) |  |
| 3 | About |  |

Check if the following tests are done:

|  |  |  |
| --- | --- | --- |
| 1 | Tested with the test protocol |  |
| 2 | Alfa-test |  |
| 3 | Beta-test |  |

Check if the following tasks are done:

|  |  |  |
| --- | --- | --- |
| 1 | Tag created in SVN |  |
| 2 | All necessary files included |  |
| 3 | Download link made |  |
| 4 | Merge branch back to trunk |  |

# release guide

If you use the “clean and build” option in Netbeans a working executable is made in the “dist” map. If for some reason this does not work correctly you can use the following chapter to do it manually.

This chapter is a guide for releasing new versions of HANtune. It contains some easy steps for making an executable from the JAR file and a file list with all the files needed for the release.

## Launch4j

Launch4j is used to make a wrapper .exe file from the JAR file created by Netbeans. Follow the next steps to do so:

1. Make sure you have the correct version, compile and test it in Netbeans.
2. Start launch4j and open the existing configuration file (launch4j\_cfg.xml) found in the main folder.
3. On the first tab fill in the next fields:

Output file -change this to the location where you want to store the .exe.

Jar -change this to the location of the Jar file, located in the folder “dist”.

Icon -Change this to the path where the icon (HANtune.ico) can be found, is should be in the main folder.

1. Check if the Java download URL is correct (<http://java.com/download>).
2. Now go to the JRE tab and check the Min JRE version field. This should be 1.7.0.
3. Check the JVM options field, it should contain “-Xmx50m”. This is used to limit the RAM usage as Java is rather generous with assigning this.
4. Now build the wrapper by clicking on the gear icon.
5. Test if it works by clicking on the arrow.

If HANtune fails to load check if all the paths are correct and perform a “clean and build” in Netbeans, witch updates the HANtune.jar file.

## Release file list

In this chapter a list of all the files needed for a HANtune release.

For compatibility and stability issues only the basic versions of the 32 and 64 bit PCAN drivers need to be included. PCAN View is not officially supported.

C:\HANTUNE\_V\_1\_00 (Change to the current version)

| HANtune.exe

| HANtune.jar

| HANtune\_manual\_0.4.pdf

| README.TXT

| SeedKeyNative.dll

|

+---Basic32

| PCANBasic.dll

| PCANBasic\_JNI.dll

|

+---Basic64

| PCANBasic.dll

| PCANBasic\_JNI.dll

|

+---lib

| AbsoluteLayout.jar

| jcommon-1.0.16.jar

| jfreechart-1.0.13.jar

| jsr173\_1.0\_api.jar

| xbean.jar

|

+---images

|

\---log