Engineering Report

Vapoterm Board Test Application

(VapothermTest.exe)

Prepared for

Vapotherm

By

Enercon Technologies

update: 11-06-2014

Prepared by:

Jim Peterson

Table of Contents

[2. Overview 2](#_Toc403065669)

[Comments 3](#_Toc403065670)

[test <test name> 3](#_Toc403065671)

[type <string> 3](#_Toc403065672)

[desc <string> 4](#_Toc403065673)

[sendline\_a <string> 4](#_Toc403065674)

[sendline\_b <string> 4](#_Toc403065675)

[readline\_a 4](#_Toc403065676)

[readline\_b 4](#_Toc403065677)

[flush\_a 4](#_Toc403065678)

[flush\_b 4](#_Toc403065679)

[expect <field> <min> <max> 4](#_Toc403065680)

[expect\_char <field> <char number> <char> 4](#_Toc403065681)

[expect\_str <field> <string> 4](#_Toc403065682)

[sleep <int - milliseconds> 4](#_Toc403065683)

[prompt <string - question> 4](#_Toc403065684)

[pause <string - comment> 4](#_Toc403065685)

[waitfor <a|b> <int-ms> <string> 4](#_Toc403065686)

[end\_script 4](#_Toc403065687)

[end\_on\_error 4](#_Toc403065688)

# Overview

The Vapotherm board test program (Vapotherm.exe) is a script driven test program designed to test boards mounted in the custom test fixture. The program communicates to both the Vapotherm board as well as a processor built into the test fixture. The communication takes place via serial connections over USB. A single USB connection from the host PC connects to a hub in the fixture that intern provides communication to both devices.

The communication with both devices uses the following serial communication settings:

* Baud Rate: 38400
* Data: 8-bit
* Parity: none
* Stop: 1 bit
* Flow Control: none

Basic program settings are stored in the initialization file VapothermTest.ini which is located in the same directory as the executable. See the section below for details. The tests are driven by a script file. Details of the possible commands in the script file are detailed below.

Initialization File (VapothermTest.ini)

Basic settings are stored from run to run in the initialization file VapothermTest.ini which is located in the same directory as the executable. Some of the settings in this file can be modified via the application’s user interface, while others can only be modified by editing this file. If this file is missing when the application starts, default values will be used and on termination of the application a new file will be created.

The settings controlled by this file are:

* PortA – Name of the port connected to the board under test.
* PortB – Name of the port connected to the fixture.
* Operator – optional name of the operator. This will be written to the reports.
* Script – Name of the script file to run
* GenerateReport – If “true” a report will be generated after the test is run.
* TerminateOnError – If “true” the tests will be terminated on the first error.
* ReportDir – Directory in which reports will be written, when enabled.
* CheckSerialNumber – when “true” the serial number field is required.
* OutputDelayMS – delay (milliseconds) between characters written to the board under test.
* TimeoutMS\_A – Timeout (milliseconds) on reads from the board under test.
* TimeoutMS\_B – Timeout (milliseconds) on reads from the fixture processor.

Of these, the first six can be set via an interface provided by the application. The following is an example of this file:

PortA=COM10

PortB=COM7

Operator=Jim

Script=C:/Scripts/VapothermTestScript.txt

GenerateReport=true

TerminateOnError=true

ReportDir=c:/Vapotherm\_Reports/

CheckSerialNumber=false

OutputDelayMS=120

TimeoutMS\_A=100

TimeoutMS\_B=100

Script Format

The script file consists of a list of commands that are sequentially executed. The commands are grouped into “Tests”. The application allows the tests to be independently turned on and off for a given run.

The command parsing is very simplistic. Each line is considered a command. White space at the beginning and ending of the line is ignored. All commands are in lower case. The commands are parsed when the script file is loaded. A warning is generated for any syntactically incorrect or incomplete commands. While you can still run the script file, the flagged commands will be ignored.

## Comments

Any line beginning with “#” is treated as a blank line. Any text following “//” is treated as white space.

## test <test name>

This command signals the beginning of a new test named test. Each test consists of a list of commands. If any one of the commands fails then the test is flagged has having failed. The test name specified will appear in the test list

There are two special named tests that are used for cleanup and shutdown. A test with the name “OnAbort” is run when the operator aborts the script. Similarly, a test with the name “OnExit” will be run on the termination of the script. Neither of these tests will show up in the list of tests.

## type <string>

This command allows the test to be tagged with a descriptive type. This will be displayed in the feedback window as well as the generated reports. This pseudo-command will always succeed.

## desc <string>

Similar to the “type” command this pseudo-command allows a description of the test to be added to the feedback window as well as the generated reports. This command will not generate errors.

## sendline\_a <string>

This command sends the specified string to the serial port associated with the device under test. The transmission delay specified in the .ini file will be applied between characters. A line termination of carriage-return, line-feed (“\r\n”) is added to the end of the string.

## sendline\_b <string>

Similar to the “sendline\_a” command, this command sends the specified string to the processor in the fixture. Unlike the “sendline\_a” command transmission delays are added.

## readline\_a

## readline\_b

## flush\_a

## flush\_b

## expect <field> <min> <max>

## expect\_char <field> <char number> <char>

## expect\_str <field> <string>

## sleep <int - milliseconds>

## prompt <string - question>

## pause <string - comment>

## waitfor <a|b> <int-ms> <string>

## end\_script

## end\_on\_error