Voyantic

Voyantic Tagsurance 2

Manual

Sep-2024

Compatible with:

Software version 2.7.0 ->

HF Firmware version 2.8.0 ->

UHF Firmware version 1.9.3 ->

1 Important Information

PLEASE READ THE COMPLETE USER GUIDE CAREFULLY BEFORE USING THE VOYANTIC TAGSURANCE™ 2 TESTER SYSTEM.

Voyantic Ltd. operates a policy of ongoing development. Voyantic Ltd. reserves the right to make changes and improvements to any of the products described in this user guide without prior notice.

THE CONTENTS OF THIS USER GUIDE ARE PROVIDED "AS IS". EXCEPT AS REQUIRED BY APPLICAPLE LAW, NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, ARE MADE IN RELATION TO THE ACCURACY, RELIABILITY OR CONTENTS OF THIS USER GUIDE. VOYANTIC LTD. RESERVES THE RIGHT TO REVISE THIS USER GUIDE OR WITHDRAW IT AT ANY TIME WITHOUT PRIOR NOTICE.

The General Terms and Conditions of Voyantic Ltd. shall apply. These can be downloaded at: http://www.voyantic.com/termsandconditions.pdf

THE MEASUREMENT UNIT ENCLOSURE DOES NOT CONTAIN ANY USER SERVICEABLE PARTS AND SHALL NOT BE OPENED. Voyantic Ltd. will inspect the seals of the equipment in case of warranty and maintenance procedures. If the seals are broken and the device therefore suspected to be tampered with, the regular warranty and the extended warranty under the maintenance program will be void.

The Voyantic Tagsurance 2 Tester makes use of radio frequencies. If the device RF-port is connected to a radiating element, such as an antenna outside a shielded environment, local permissions or approvals may be needed. VOYANTIC LTD. DOES NOT WARRANT ANY TYPE OF APPROVAL FOR VOYANTIC TAGSURANCE 2 TESTER. VOYANTIC LTD. SHALL UNDER NO CIRCUMSTANCES BE LIABLE OF ANY USE OF VOYANTIC TAGSURANCE 2 TESTER.

Use of any additional software requires a valid license. Any copyrights, patents and other intellectual property rights (including the right to change and further develop) in and to the Voyantic Tagsurance 2 Tester (including any related documentation and other materials delivered by Voyantic Ltd.) shall belong to Voyantic Ltd.

2 Table of Contents

1 IMPORTANT INFORMATION 2

2 TABLE OF CONTENTS 3

3 PRODUCT OVERVIEW 5

3.1 OPERATING PRINCIPLE 5
3.2 FUNCTIONS AND LICENSE OPTIONS 6
3.3 TECHNICAL SPECIFICATIONS 8

4 QUICK-START AND MANUAL REFERENCE GUIDE 10

4.1 HOW TO INSTALL TEST SYSTEM AND BUILD THE BASIC TEST MEASUREMENT SETUP 10
4.2 HOW TO MAKE THE FIRST TEST MEASUREMENT 10
4.3 TROUBLESHOOTING 11

5 SOFTWARE INSTALLATION AND MEASUREMENT SETUP 12

5.1 PC REQUIREMENTS 12
5.2 INSTALLING TAGSURANCE 2 GRAPHICAL USER INTERFACE 12
5.3 UNINSTALLING TAGSURANCE 2 GRAPHICAL USER INTERFACE 14
5.4 SETTING UP A MEASUREMENT SETUP 15

6 OPERATION USING THE TAGSURANCE 2 GRAPHICAL USER INTERFACE 17

6.1 DEVICE MANAGER 18
6.2 TEST MANAGER 21
6.3 INTERFACE MANAGER 31
6.4 OPERATOR INTERFACE 32

7 LANGUAGE PACKS 37

7.1 OVERVIEW 37
7.2 GUI 37
7.3 CONFIGURATION 37
7.4 UPDATING EXISTING INSTALLATIONS 38

TROUBLESHOOTING 39

EXTIO SIGNALING 41

7.5 TAGSURANCE HF VERSION 3.X 41
7.6 TAGSURANCE HF VERSION 1.X & 2.X 42
7.7 TAGSURANCE UHF 45

8 FIRMWARE UPDATE PROCEDURE 48

8.1 TAGSURANCE HF 48
8.2 TAGSURANCE UHF 49

9 OPERATION USING THE TAGSURANCE HF TCP PROTOCOL 50

9.1 DEVICE CONNECTION ESTABLISHMENT 50
9.2 COMMANDS DESCRIPTION 51

10 OPERATION USING THE UHF SERIAL COMMAND INTERFACE 73

10.1 COMMUNICATION PROTOCOL 73

10.2 SYSTEM CONFIGURATION COMMANDS 75
10.3 INLINE MEASUREMENT COMMANDS 77
10.4 OTHER MEASUREMENT COMMANDS 83
10.5 TAG ENCODING, LOCKING, AND KILLING COMMANDS 85
10.6 COMBINING ENCODING WITH PERFORMANCE TESTING 96
10.7 ERROR HANDLING 96

APPENDIX A TCP REMOTE ACCESS INTERFACE FOR TAGSURANCE 2 97

A.1 OVERVIEW 97
A.2 REMOTE ACCESS CONNECTION ESTABLISHMENT 98
A.3 COMMANDS DESCRIPTION 100

APPENDIX B TIMING CALCULATIONS 122

B.1 TAGSURANCE HF 122
B.2 TAGSURANCE UHF 124

APPENDIX C SUMMARY OF TAGSURANCE UHF SERIAL INTERFACE COMMANDS 125

3 Product Overview

3.1 Operating Principle

The testing of the tag samples is always carried in the active state using the selected carrier frequency and transmitted power. The tag is given a command according to the communication protocol and it is detected if a valid response is sent back by the tag. Active mode is always preferred as it tests both the performance and the functionality by challenging the tag in a similar way as an RFID reader would, providing applicable results.

HF tags test

The measurement unit is optimized to operate over the frequency range of 10 to 30MHz for HF tags. The reason for the large range is to match and exceed the operating range of the tags. HF inlay tags operate at 13,56MHz, but are typically tuned up to 14-16MHz when operating in an unloaded state. This way the tag can adapt to detuning caused by the environment, such as a hand holding a public transport ticket. As the center frequency and bandwidth are important for the predictable behavior of the tag, it is also important to verify the tuning for every tag in production. For this reason, the wide 10 to 30MHz operating range is needed.

UHF tags test

The measurement unit is optimized to operate over the extended frequency range of 800 MHz to 1100 MHz. The reason for the large range is to match and exceed the operating range of the tags. RFID inlay tags typically operate on some bandwidth on a range from 850 MHz to around 1050 MHz. This way the tag is more tolerant to material induced detuning and has an optimally stable performance over the entire 865 MHz to 960 MHz range even when attached on various materials. As the broad operating range is important for the tag, it is also important to verify the bandwidth for every tag in production. For this reason, the 800 MHz to 1100 MHz operating range is needed.

3.2 *Functions and License Options*

Tagsurance HF

Test type	Supported protocols	License type
Point test	ISO 15693, ISO 14443-A, ISO 14443-B (Standard,ST25TB), FeliCa, ISO 18000-3M3, TTO PR1101, TTO PR1102, TTO NFC Barcode	Full or temporary
UID read test	ISO 15693, ISO 14443-A, ISO 18000-3M3, ISO 14443-B (ST25TB), TTO PR1101, TTO PR1102, TTO NFC Barcode	Full or temporary
Sensitivity test	ISO 15693, ISO 14443-A, ISO 14443-B (Standard,ST25TB), FeliCa, ISO 18000-3M3, TTO PR1101, TTO PR1102, TTO NFC Barcode	Full or temporary
*Sweep test**	ISO 15693, ISO 14443-A, ISO 14443-B (Standard,ST25TB), FeliCa, ISO 18000-3M3, TTO PR1101, TTO PR1102, TTO NFC Barcode	Full or temporary
Freq Range	12-16 MHz or 10-30 MHz	Full

*Reference Curve measurement doesn’t require any license

Tagsurance UHF

Test type	Standard features	Optional features	License type
Freq. range	860 – 960 MHz	800 – 1100 MHz	Full or temporary
Point test	X		Full or temporary
Sweep test	Reference Curve	X	Full or temporary
Read test		X	Full or temporary
Write test		X	Full or temporary
Sensitivity test		X	Full or temporary
Encode, Kill, lock		X	Full or temporary
6B protocol support		X*	Full or temporary
GB protocol support		X**	Full or temporary

*6B protocol support includes: Point Test, Sweep Test, Sensitivity Test, Read Test
**GB protocol support includes: Point Test, Sweep Test, Sensitivity Test

GUI License

License options	Standard features	Optional features	License type
GUI UHF	X		Full or temporary
GUI HF	X		Full or temporary
Multi Device Option		X	Full or temporary
*Plug-Ins**		X	Full or temporary

*This option might not enable all the plug-ins installed on the machine as some specific plug-ins need to be purchased separately.

3.3 Technical Specifications

HF Specifications

Parameter	**Value**
Frequency	12 to 16MHz (standard range), 10 to 30MHz (extended range)
Output Power Range	-10 to +25dBm
Output power accuracy	+/-1dB @-10 to +25dBm
Power resolution	0,1dB
Supported protocols	ISO 15693, ISO 14443-A, ISO 14443-B, FeliCa, ISO 18000-3M3

HF Supported protocols and commands

Protocol	**Point**	**Sensitivity**	**Sweep**	**UID Read**
ISO 15693	Inventory	Inventory	Inventory	Inventory
ISO 14443-A	REQA	REQA	REQA	Anticollision sequence
ISO 14443-B	REQB	REQB	REQB	
ISO 14443-B ST25TB	Initiate	Initiate	Initiate	Get_UID
FeliCa	Polling	Polling	Polling	
ISO 18000-3M3	Begin Round	Begin Round	Begin Round	Read
*TTO PR1101**	*n/c*	*n/c*	*n/c*	*n/c*
*TTO PR1102**	*n/c*	*n/c*	*n/c*	*n/c*
*NFC Barcode**	*n/c*	*n/c*	*n/c*	*n/c*

*During tests of Tag Talks Only (TTO) tags Tagsurance outputs unmodulated carrier signal to provide power to the tag. No commands (n/c) are sent out from the Tagsurance unit.

UHF Specifications

Parameter	**Value**
RF input power	+30 dBm
External Power Voltage	17-20 VDC
Output Power Range	-10 to +25dBm
Output power accuracy	+/-1dB @-10 to +25dBm
Power resolution	0,25dB
Carrier freq. accuracy	10 ppm
Carrier freq. resolution	100 kHz
Supported protocols	ISO 18000-6C, ISO 18000-6B, ISO 18000-6B-D (6B-Double), GB/T-29768

4 Quick-start and Manual Reference Guide

A basic test setup that can be used to test the system and learn how to use it includes a Tagsurance (HF or/and UHF) test unit, Snoop Pro (HF or/and UHF) coupling element, and the PC software. The installation steps and guide for making the first test measurements are briefly listed below. Detailed instructions can be found from referred chapters.

4.1 How to install test system and build the basic test measurement setup

1. Install Tagsurance 2 software (see Chapter 5.1 for more information)
2. Place the GUI license file to the correct location (…/Tagsurance 2/Data/license.lic)
3. Unpack and build the basic measurement setup (see Chapter 5.4 for more information)
4. Wait for the device to initialize, and run Tagsurance 2 Launcher software
5. Associate the test device(s) with the Tagsurance 2 software (see Chapter 6.1 for more information)
6. Ready to start testing

4.2 How to make the first test measurement

1. Run Tagsurance 2 Launcher software
2. Create a tag test including the test device (see Chapter 6.2 for more information)
3. Run the test (see Chapter 6.4 for more information)

4.3 Troubleshooting

1. Software installation fails:

   a. Ensure that user privileges do not restrict installation process

2. Connection issues with the test device

   a. Check setup and connections, and reset device (see Chapter 5.4 for more information)

   b. Check device associations (see Chapter 6.1 for more information)

   c. Check that the correct device is associated with test (see Chapter 6.2 for more information)

3. No test results are received, or test cannot be started

   a. Check Ethernet cable (Tagsurance HF device)

   b. Check Serial Cable (Tagsurance UHF device)

   c. Check device settings related to the test (see Chapter 6.2 for more information)

   d. Check that the appropriate license is in use (see Chapter 3.2 for more information)

4. If these steps do not solve the issue, please contact support@voyantic.com

5 Software installation and measurement setup

5.1 PC Requirements

Windows 10

20 GB free disk space (installation does not need that much but we recommend for not getting in trouble with storing results etc.)

Processor that has a CPU benchmark of 4000 points or higher (see cpubenchmark.net)

8 GB of memory (4 GB would do it for us but for nice operation with Windows, get 8 GB or more)

5.2 Installing Tagsurance 2 Graphical User Interface

NOTE

It is recommended to turn off the Windows fast startup during the installation if requested by installer.

INSTALLATION

To install the Tagsurance 2 Graphical User Interface:

1. To prepare for the installation process, if updating an existing copy of the software, please make a backup copy of the …Tagsurance 2\Data\ folder before running the installer to make sure all data lost during the installation process can be restored from a safe location.

2. Browse to the installation package location on the USB stick included with the delivery or the software package downloaded from Voyantic download site and run: .../Tagsurance 2 Installer v.../setup.exe.

3. Select the destination folder paths for the installation files

   - Note that when used, the program creates and updates files under its own installation folder. The program and the user must have the privileges for the location for the installation to proceed correctly. Specific Windows system folders have restricted access for programs to write to. This includes the Program files folder in Windows 7/8/10.

4. Read and accept the license agreements and follow the instructions on the screen to install the product

5. The summary of the contents to-be-installed/modified will be presented for acceptance before starting the installation

6. After successful installation, the installer prompts the user to restart their computer

7. If making an update to an existing program, check …Tagsurance 2\Data\ folder for the following files: Devices.txt, test cases, and output files exist. Also ensure that Devices.txt contents match the original copy. If not, these files can be restored from the backup folder made in step 1.
8. Place the GUI license file in the correct location (…/Tagsurance 2/Data/license.lic)
9. The program is ready for use

5.3 Uninstalling Tagsurance 2 Graphical User Interface

1. If there are any measurement results, setup files, etc. stored under C:\Tagsurance 2, please take backups before continuing
2. Login to the computer with Administrator privileges
3. Go to Control Panel -> Programs and Features
4. Find Tagsurance 2 and uninstall it
5. Find National Instruments Software from the list and double click it
6. Select ‘NI System configuration runtime 17.5.0’ and click remove selected
7. In case there are products which are dependent on NI System Configuration Runtime, the uninstaller will display a list of these products. If there are products on the list still that are still required do not remove these NI components.
8. Allow the uninstaller to finish and reboot the computer when prompted to do so
9. Login to the computer with Administrator privileges
10. Delete C:\Tagsurance 2 folder

5.4 Setting Up a Measurement Setup

5.4.1 Tagsurance HF

Setting up the basic test setup requires:

- Tagsurance HF Unit with a power supply
- Snoop Pro HF coupling element with RF cable
- Ethernet cable
- A computer with Tagsurance 2 installed
- I/O beeper connector, pedal trigger, and DB25->DB15 adapter (optional for manual testing)

Assembly instructions:

1. Connect Snoop Pro HF to the TX Port on the device front panel (RF 50Ω)

2. Connect an Ethernet cable from the computer to the Ethernet port in the device back panel (LAN is required!)

3. Connect the power supply to the power in port on the device back panel (18-24V)

4. Wait for the device to initialize. A beep tone will sound and power led brightens when the device is ready for operation.

5. Start the Tagsurance 2 application to perform measurements.

5.4.2 Tagsurance UHF

Setting up the basic test setup requires:

- Tagsurance UHF with a power supply
- Snoop Pro UHF coupling element with RF cable
- RS-232 serial data cable
- A computer with Tagsurance 2 installed
- Serial adapter (if a native serial port is not available)
- I/O beeper connector and pedal trigger (optional, used for manual testing)

Assembly instructions:

1. Connect Snoop Pro UHF to the TX port in the device front panel (RF 50Ω)

2. Connect the serial cable from the computer to the serial port in the device back panel (RS-232)

3. Connect the power supply to the Power in port in the device back panel (18VDC)

4. Connect the I/O beeper connector to the device I/O port and pedal trigger to the plug in the beeper component (only applicable for the Manual Test Station)

5. Start the Tagsurance 2 application to perform measurements

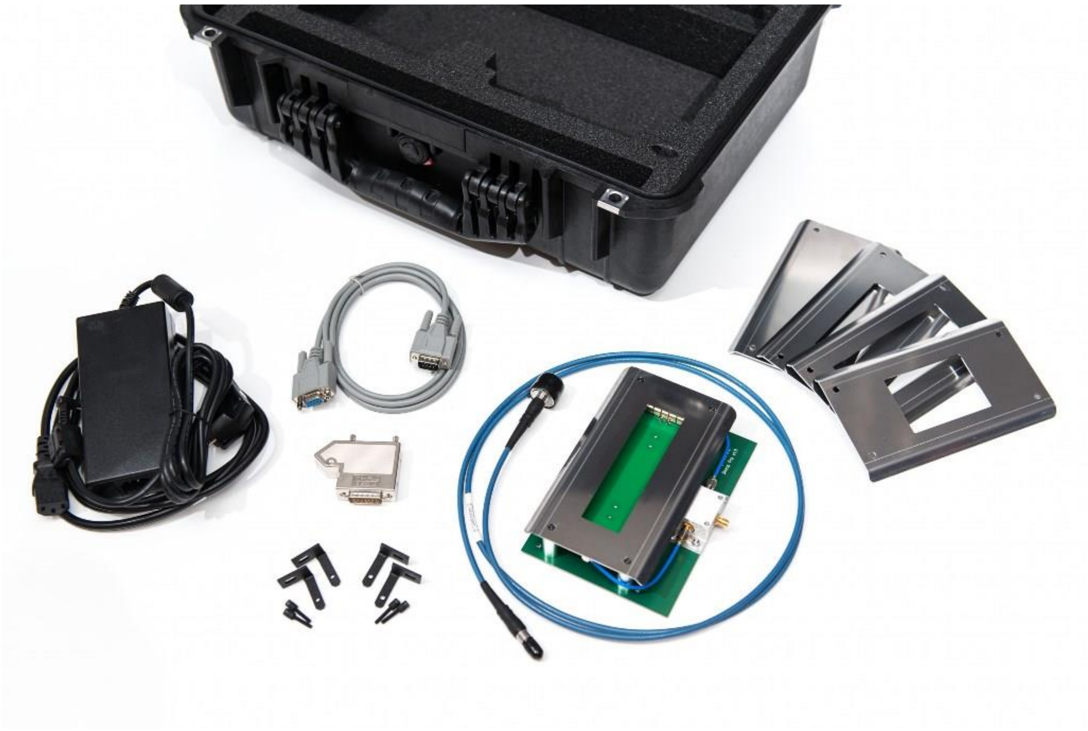


Figure 1: Basic UHF Starter Kit

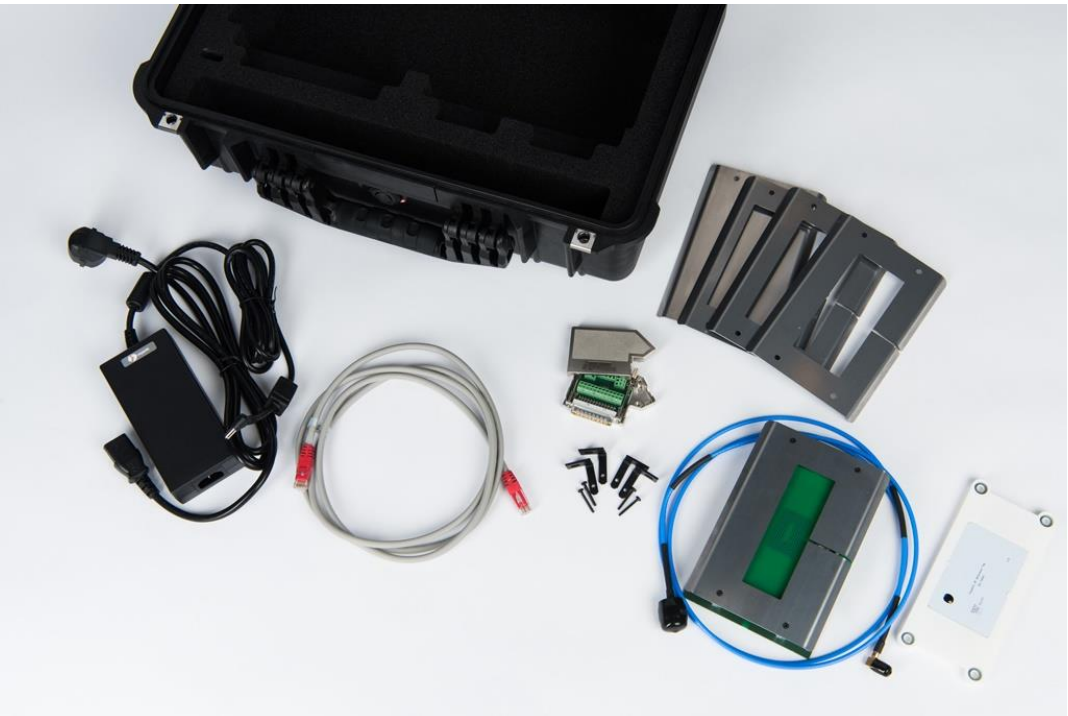


Figure 2: Basic HF Starter Kit

6 Operation Using the Tagsurance 2 Graphical User Interface

Overview

Tagsurance 2 is a dedicated PC software used to manage test operations with Tagsurance UHF and HF test devices. It provides simple and efficient tools for device management (Device Manager), test definition (Test Manager), Interface settings (Interface Manager) test process (Operator Interface). These functions are introduced in the following Chapters 6.1, 6.2, Interface Manager and Operator Interface.

To start the application, run either 'Tagsurance 2 Launcher.exe' or 'Tagsurance 2.exe'. 'Tagsurance 2 Launcher.exe' launches a menu that can be used to navigate to all of the available software, while "Tagsurance 2.exe" launches Tagsurance 2 Operator Interface.

Important note!

Particularly when using "Tagsurance 2.exe", please make sure that the device association file (Devices.txt) is properly defined and available in the Tagsurance 2 Data folder ('<system directory>\Tagsurance 2\Data\Devices.txt') before launching the executable. Devices.txt includes all device association data and connection parameters and is utilized by all of the Tagsurance 2 software tools. Tagsurance 2 operator interface and Test Manager will not run properly if the file is corrupted or missing. The file can be managed by using Tagsurance 2 Device Manager (recommended), or manually edited by using any standard text editor. If run specific plug-ins are ran (eg. Multi Lane Viewer), make sure that the "Plug-In Configuration.ini" file has been edited correctly by using the "Interface Manager" tool accessible the Tagsurance 2 Launcher.

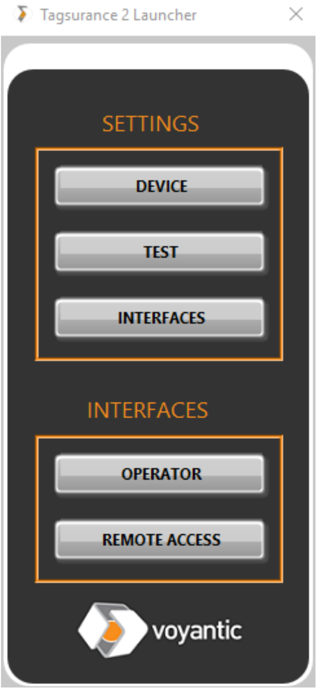


Figure 3: Tagsurance 2 Launcher menu.
All functions except Device manager and Interface will be disabled, if "Devices.txt" is not properly defined.

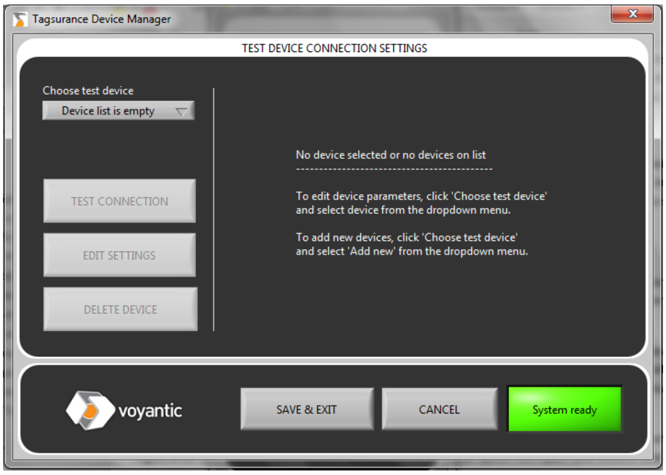
6.1 Device Manager

Overview

Tagsurance 2 Device manager is an interactive program used to manage test device associations and device communication settings. Tagsurance 2 supports controlling multiple test devices simultaneously (both UHF and HF). It keeps track of the devices associated with the software in an initialization file named 'Devices.txt', located in Tagsurance 2 Data folder. Device Manager can read or modify the contents of this file. The file is a human readable text file and can be modified by using any standard text editor. However, to avoid syntax errors and human errors, it is recommended to use the Device Manager tool.

Adding new test devices

The first time the program is run, the initialization file 'Devices.txt' is empty, and no devices are present on the device list. To add a new device, click "Choose test device" and select: "Add new device" from the drop-down menu. This will launch a pop-up window where the user can define the device name, type and relevant connection parameters.



Tagsurance Device Manager

TEST DEVICE CONNECTION SETTINGS

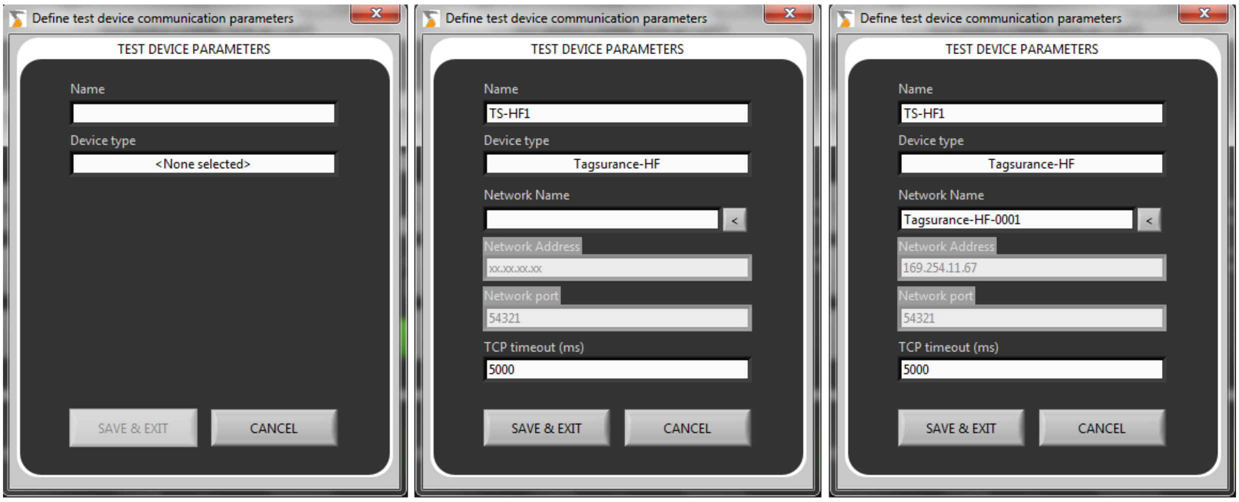
Choose test device: Device list is empty

No device selected or no devices on list

To edit device parameters, click 'Choose test device' and select device from the dropdown menu.

To add new devices, click 'Choose test device' and select 'Add new' from the dropdown menu.

Figure 4: Tagsurance 2 Device Manager tool



Define test device communication parameters

TEST DEVICE PARAMETERS

Name:

Device type: <None selected>

Define test device communication parameters

TEST DEVICE PARAMETERS

Name: TS-HF1

Device type: Tagsurance-HF

Network Name:

Network Address: xx.xx.xx.xx

Network port: 54321

TCP timeout (ms): 5000

Define test device communication parameters

TEST DEVICE PARAMETERS

Name: TS-HF1

Device type: Tagsurance-HF

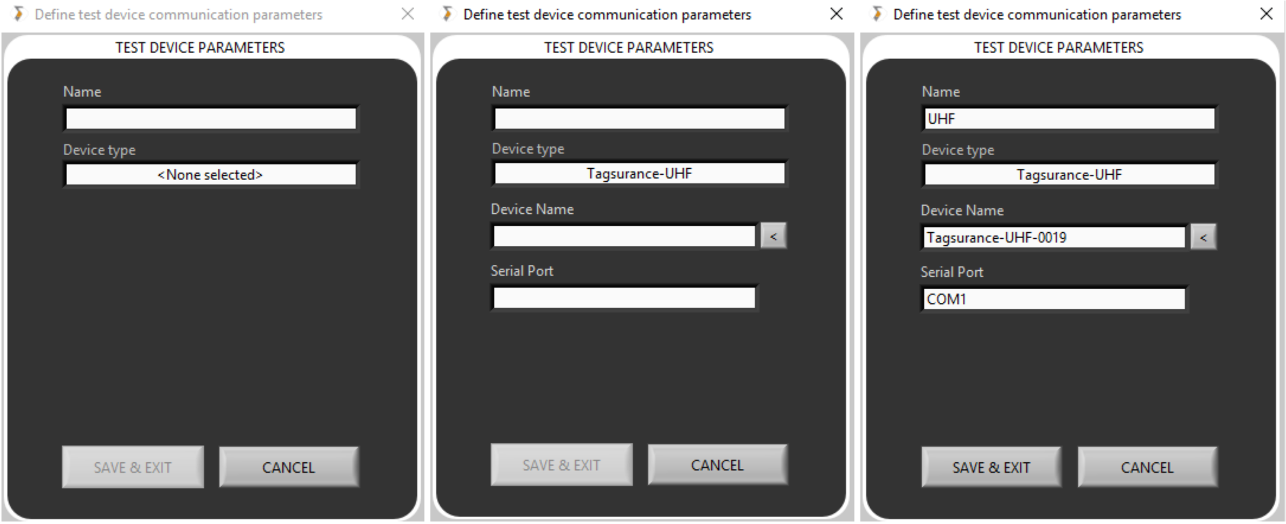
Network Name: Tagsurance-HF-0001

Network Address: 169.254.11.67

Network port: 54321

TCP timeout (ms): 5000

Figure 5: Adding a new Tagsurance HF test device



Define test device communication parameters

TEST DEVICE PARAMETERS

Name:

Device type: <None selected>

Define test device communication parameters

TEST DEVICE PARAMETERS

Name:

Device type: Tagsurance-UHF

Device Name:

Serial Port:

Define test device communication parameters

TEST DEVICE PARAMETERS

Name: UHF

Device type: Tagsurance-UHF

Device Name: Tagsurance-UHF-0019

Serial Port: COM1

Figure 6: Adding a new Tagsurance UHF test device

Name is an identifier for the specific device used to link test cases and setups to that specific device. The name can be freely defined, but should be unique, contain no special characters or spaces, and be contain only capital letters.

Device type indicates the type of the device. Tagsurance 2 supports two device types: Tagsurance HF and Tagsurance UHF. Tagsurance HF refers to Tagsurance HF test device connected over Ethernet. Choosing that option opens new controls where the user can search the device by clicking on the "<" button and define *TCP timeout (ms)*. The rest of the network settings are defined automatically after selecting a device.

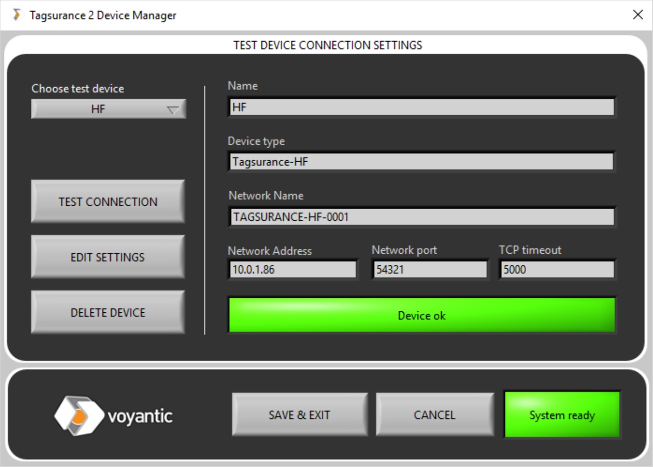
Tagsurance UHF refers to Tagsurance UHF test device connected over Serial RS-232. Choosing that option opens new controls where the user can search the device by clicking on the "<" button. The Serial Port in use is defined automatically after selecting a valid device.

To finish the setup and to save the new test device(s) to the list, click on "SAVE & EXIT". The program will then test the connection to the test device and indicate whether the connection attempt was successful. If the connection attempt fails, the user is asked to retry, or to cancel. Clicking retry will initiate another connection test. Clicking cancel will ignore the failed connection attempt and return to Device Manager's main window (test device will still be added to the list).

The "Device ok" LED will show the firmware version of the configured device when the mouse cursor is moved on top of the button itself.

Editing test device setting and test device list

Once all devices have been added the user may take further actions. The connection parameters can be edited or the connection tested by choosing the device from the choose test device drop-down menu and clicking "Edit settings" or "Test connection" respectively. The connection test may take some tens of seconds. The software will inform user when the test has been completed. In addition, devices can be removed from the list by clicking: "Delete device".



Choose test device: HF

Name: HF

Device type: Tagsurance-HF

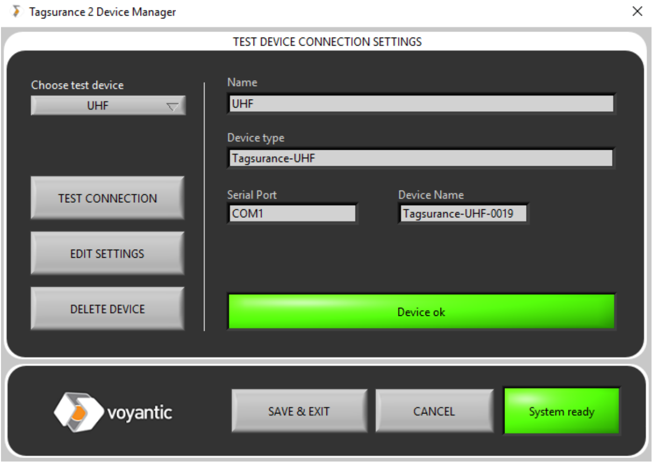
Network Name: TAGSURANCE-HF-0001

Network Address: 10.0.1.86

Network port: 54321

TCP timeout: 5000

Figure 7: HF Device has been added



Choose test device: UHF

Name: UHF

Device type: Tagsurance-UHF

Serial Port: COM1

Device Name: Tagsurance-UHF-0019

Figure 8: UHF Device has been added

Saving changes to Devices.txt

Once all device settings have been defined, close the Device Manager and return to the Launcher menu by clicking on “Save & exit” or “Cancel”. Clicking “Save & exit” will save any changes to the Devices.txt (or create the file), while clicking: “Cancel” will reject them and leave “Devices.txt” unchanged.

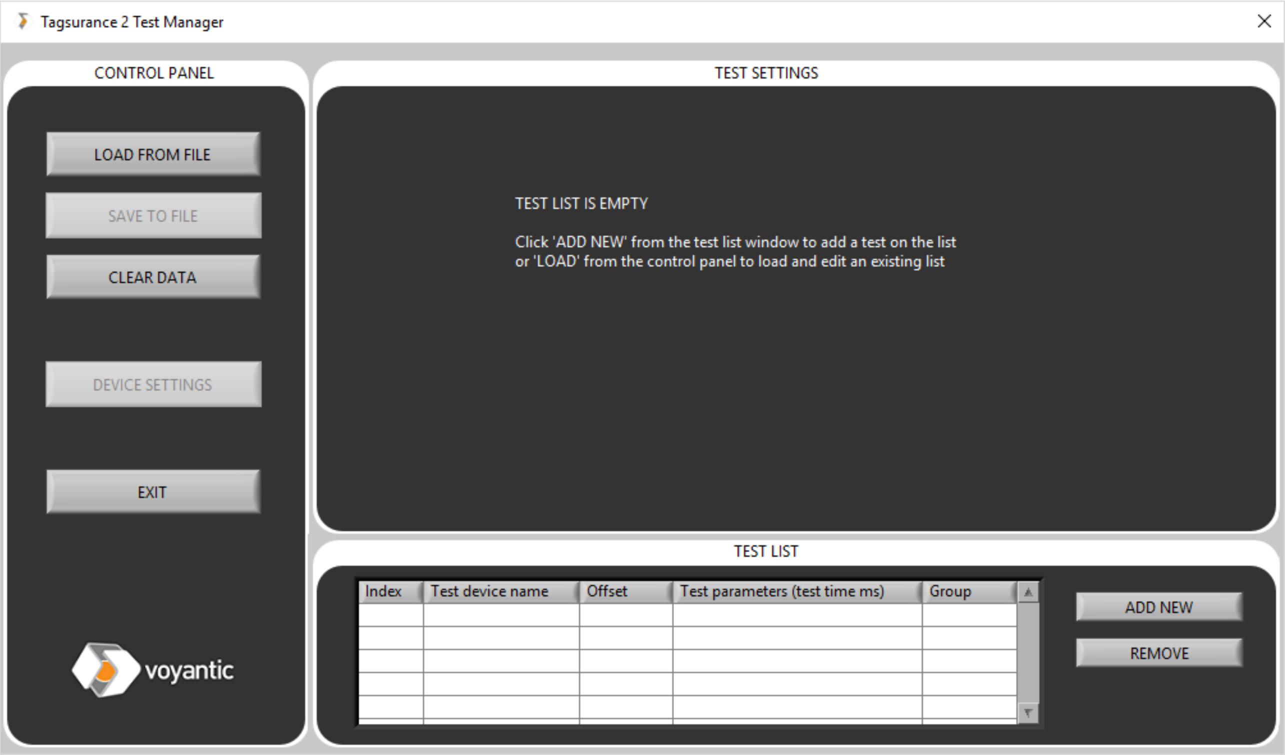
6.2 *Test Manager*

Overview

Tagsurance 2 Test Manager is a dedicated tool for creating and editing test files. The tool supports multi-device and multilane testing and is compatible with Tagsurance HF and Tagsurance UHF, thus enabling the testing of dual frequency tags as well.

A typical test definition process includes the following steps:

1. Define test instances, one per each test position. For example, if there are four lanes and 1 Snoop Pro coupling element per lane, there should be four test instances added to the list.
2. Define test(s) to-be-performed for each instance. The definition of the tests depends on the test device type associated with the instance.
3. Define and check test device settings. Test device settings, such as trigger source and type, are associated with the test file and can be modified in the Test Manager tool. Test device settings management is described in this chapter.



TEST LIST IS EMPTY

Click 'ADD NEW' from the test list window to add a test on the list or 'LOAD' from the control panel to load and edit an existing list

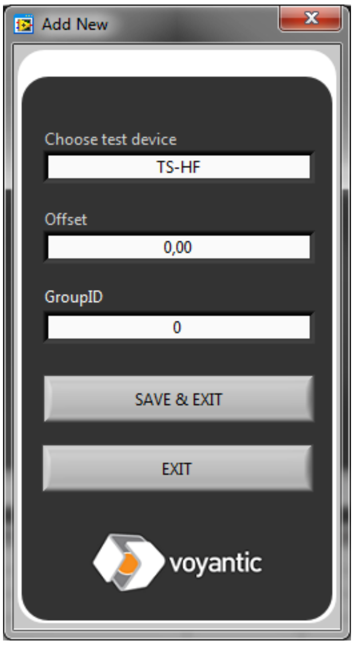
Index	Test device name	Offset	Test parameters (test time ms)	Group

Figure 9: Test Manager tool startup view. The tool allows creating and managing test files for Tagsurance 2 GUI

Test Instance List

A test instance is a location on a production line where tests are performed. In a single-lane production lane, the tags are typically tested at a single location, while a 4-lane production lane might have 4 test positions. Thus, in those cases there would be 1 and 4 test instances, respectively. The Test instance list is a list of test instances and the associated parameters (test device, group, and trigger offset).

To add an instance on the list, click on “Add new”. This will open a pop-up dialog where the user must define the test device, GroupID and trigger offset.



Add New

Choose test device
TS-HF

Offset
0,00

GroupID
0

Figure 10: Add new/edit test instance dialog

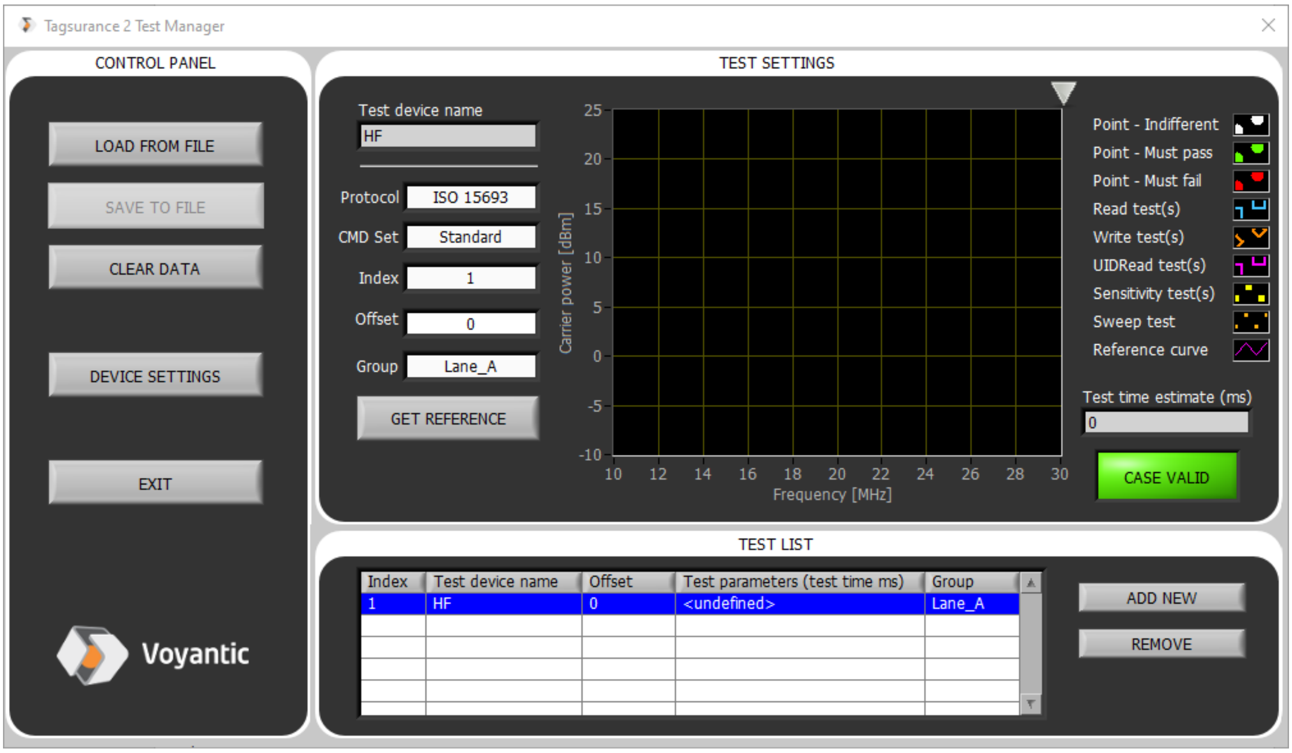
Choose test device selects the test device used to conduct the test. To become available in the drop-down menu, the device(s) must be defined in “Devices.txt”.

Offset is the number of trigger intervals between test instances on the production line. In practice, it controls the number of test results that are ignored before starting to save data. In a typical single-device test system, offset is 0, but if there are multiple test instances on the same lane, it is used to bundle test data from different sources. For example, in a single lane production lane where there are two test instances, and 3 tags in between, the offset would be 0 for the first and 3 for the latter one. For the instance #1, the data is valid from the beginning. For the instance #2, first 3 results are ignored, since it takes 3 trigger intervals for the tag first tested on instance #1 to travel to instance #2. This way the first saved result for both instances represent the data for the same tag.

GroupID is used to tie results from different test instances together. The results from instances that have same GroupID are bundled and saved together as a single log file, whereas results from instances with different names are collected in separate log files. In practice, if there are, for example, three test lanes, there should also be three test groups. In a test setup with multiple test instances on a single lane, there should only be one GroupID which ties the results from all instances together. In a single-line, single-device setup, this parameter can be ignored.

In a typical Multilane setup, the number of lanes and unique Groups are equal. In this case the Index number of a group is the same as the channel number used to trigger a measurement in Tagsurance (e.g. Index 4 is triggered by channel 4). The channel pulse widths can be found on page 57 for HF, and on page 75 for UHF. The order of the Test List may be edited after creation by dragging and dropping the items in the list.

Once the instance settings are defined, clicking on "SAVE & EXIT" saves the parameters and adds the new instance to the list. "EXIT" rejects all changes. Once saved to the list, the instance parameters can be modified by clicking the item on the list and modifying the parameters in the upper window. Instances can be removed from the list by clicking the item on the list and "REMOVE" button in the list window.



Test device name: HF

Protocol: ISO 15693

CMD Set: Standard

Index: 1

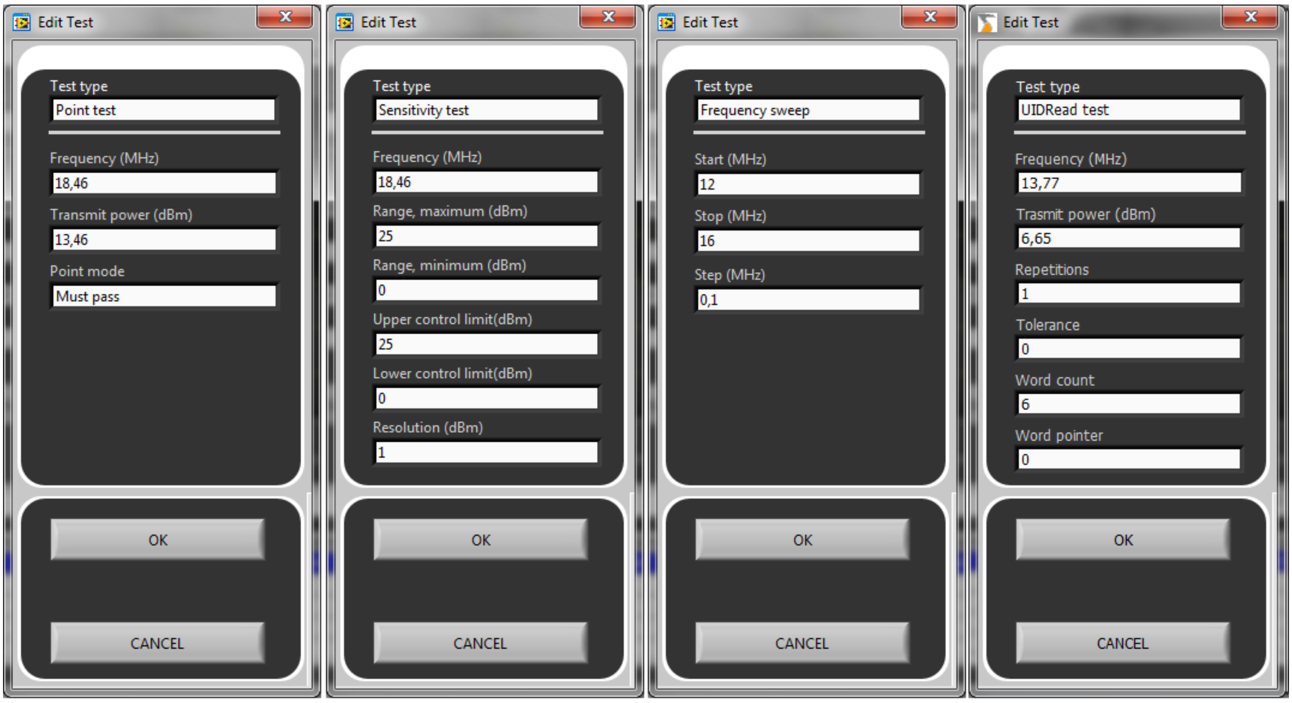
Offset: 0

Group: Lane_A

Test time estimate (ms): 0

Index	Test device name	Offset	Test parameters (test time ms)	Group
1	HF	0	<undefined>	Lane_A

Figure 11: The first test instance has been added to the list. Parameters can be modified in the upper window and the instance be removed by clicking: "REMOVE". New instances can be added by clicking: "ADD NEW"



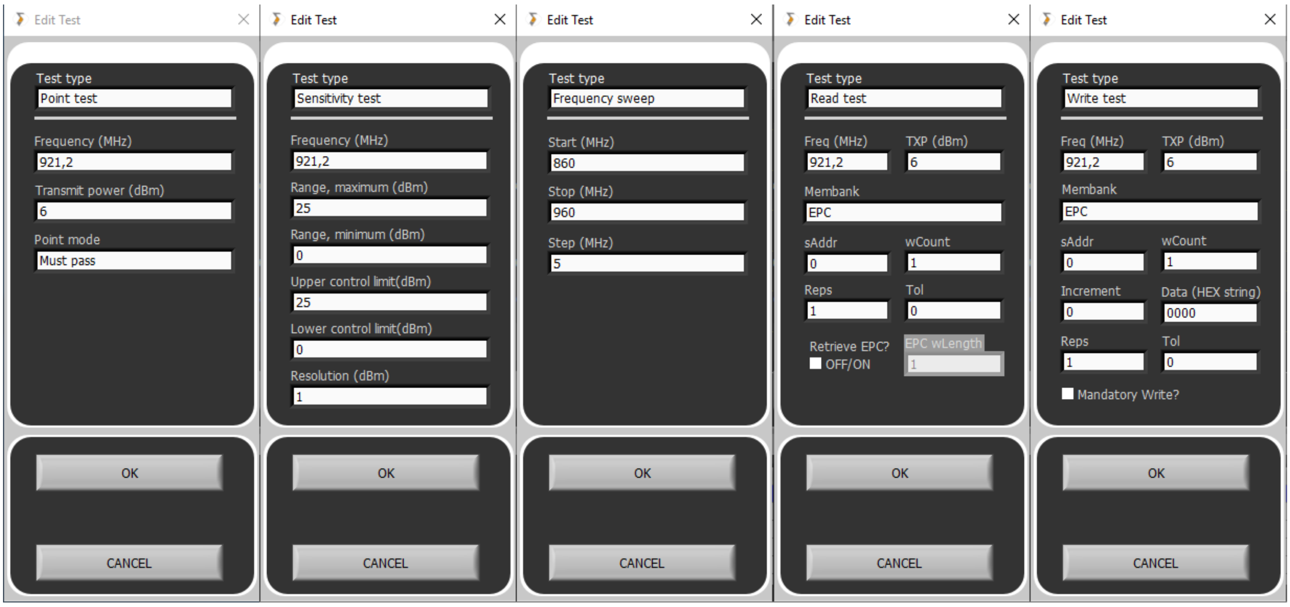
Edit Test
Test type: Point test
Frequency (MHz): 18,46
Transmit power (dBm): 13,46
Point mode: Must pass
OK
CANCEL

Edit Test
Test type: Sensitivity test
Frequency (MHz): 18,46
Range, maximum (dBm): 25
Range, minimum (dBm): 0
Upper control limit(dBm): 25
Lower control limit(dBm): 0
Resolution (dBm): 1
OK
CANCEL

Edit Test
Test type: Frequency sweep
Start (MHz): 12
Stop (MHz): 16
Step (MHz): 0,1
OK
CANCEL

Edit Test
Test type: UIDRead test
Frequency (MHz): 13,77
Trasmit power (dBm): 6,65
Repetitions: 1
Tolerance: 0
Word count: 6
Word pointer: 0
OK
CANCEL

Figure 12: Add test dialog for Tagsurance HF target. When editing a test, the delete option will also appear



Edit Test
Test type: Point test
Frequency (MHz): 921,2
Transmit power (dBm): 6
Point mode: Must pass
OK
CANCEL

Edit Test
Test type: Sensitivity test
Frequency (MHz): 921,2
Range, maximum (dBm): 25
Range, minimum (dBm): 0
Upper control limit(dBm): 25
Lower control limit(dBm): 0
Resolution (dBm): 1
OK
CANCEL

Edit Test
Test type: Frequency sweep
Start (MHz): 860
Stop (MHz): 960
Step (MHz): 5
OK
CANCEL

Edit Test
Test type: Read test
Freq (MHz): 921,2
TXP (dBm): 6
Membank: EPC
sAddr: 0
wCount: 1
Reps: 1
Tol: 0
Retrieve EPC? □ OFF/ON
EPC wLength: 1
OK
CANCEL

Edit Test
Test type: Write test
Freq (MHz): 921,2
TXP (dBm): 6
Membank: EPC
sAddr: 0
wCount: 1
Increment: 0
Data (HEX string): 0000
Reps: 1
Tol: 0
□ Mandatory Write?
OK
CANCEL

Figure 13: Add test dialog for Tagsurance UHF target. When editing a test, the delete option also will appear

Test definitions

Once there are one or multiple test instances saved on the list, the test(s) associated with the instances are defined in the window which appears above the test list. The procedure depends on the associated test device type.

Adding tests for Tagsurance HF

If the test device type associated with the test instance is Tagsurance HF, a test can be added by clicking the graph and defining test parameters from the pop-up dialog (Figure 12). The test type needs to be chosen first, and the rest of the parameters will appear based on which test was selected. Test types and the associated parameters are explained in the table below. When ready, the test can be saved by clicking: "OK" or rejected by clicking: "Cancel". Once a test has been saved, the test may be edited by double- or right clicking the test on the graph. This opens the task edit dialog, allowing modification and deletion of the test. Drag and drop is also supported. The Test protocol is chosen from the drop-down menu, named: 'Protocol'. The selected protocol is used for tasks of that specific instance.

Test type	Test description	Test parameters
Point test	A single command[1] is transmitted and it is detected whether tag replies or not	Carrier transmit frequency Carrier transmit power Point mode - Interpretation of the test[3]
UID read test	UID memory read[1] is performed and it is detected if read is successful or not	Carrier transmit frequency Carrier transmit power Repetitions - Maximum number of test repetitions Tolerance - Maximum number of failed test repetitions Word pointer - address of the first word to be read[4] Word counter - word count to be read[4]
Sensitivity test	Threshold power at a single frequency is searched[1] and it is detected if it is within control limits	Carrier transmit frequency (MHz) Range, maximum – Highest test power Range, minimum – Lowest test power Upper control limit – Highest accepted threshold power Lower control limit – Lowest accepted threshold power Resolution – Allowed uncertainty of the search result
Frequency sweep	Threshold power across a frequency range is searched[1]	Start – Start frequency of the sweep Stop – Stop frequency of the sweep Step – Frequency step of the sweep

[1] See HF Supported protocols and commands at page 8 for more information about supported commands

[3] Must respond, must not respond, indifferent (tag may or may not respond)

[4] Valid for ISO 18000-3M3 only

Adding tests for Tagsurance UHF

If the test device type associated with the test instance is Tagsurance UHF, the test can be added by clicking the graph and defining test parameters the pop-up dialog (Figure 13). The test type needs to be chosen first, and the rest of the parameters will appear based on which test was selected. Test types and the associated parameters are explained in the table below. Once ready, the test can be saved by clicking: "OK" or rejected by clicking: "Cancel". Once a test has been saved, the test may be edited by double- or right clicking the test on the graph. This opens the task edit dialog, allowing modification and deletion of the test. Drag and drop is also supported. The Test protocol is chosen from the drop-down menu, named: 'Protocol'. The selected protocol is used for tasks of that specific instance.

Test type	Test description	Test parameters
Point test	A single command[1] is transmitted and it is detected whether tag replies or not	Carrier transmit frequency Carrier transmit power Point mode - Interpretation of the test[3]
Read test	memory bank read[2] is performed and it is detected if read is successful or not	Carrier transmit frequency Carrier transmit power Repetitions - Maximum number of test repetitions Mem bank – Reserved, EPC, TID and User Tolerance - Maximum number of failed test repetitions Word pointer - address of the first word to be read Retrieve EPC[4] – Retrieve the EPC data from the Inventory sequence EPC wLength – Select the EPC length (words) to be retrieve
Sensitivity test	Threshold power at a single frequency is searched[1] and it is detected if it is within control limits	Carrier transmit frequency (MHz) Range, maximum – Highest test power Range, minimum – Lowest test power Upper control limit – Highest accepted threshold power Lower control limit – Lowest accepted threshold power Resolution – Allowed uncertainty of the search result
Frequency sweep	Threshold power across a frequency range is searched[1]	Start – Start frequency of the sweep Stop – Stop frequency of the sweep Step – Frequency step of the sweep
Write test	memory bank write[2] is performed and it is detected if read is successful or not	Carrier transmit frequency Carrier transmit power Repetitions - Maximum number of test repetitions Data (HEX string) – Data value to write Mem bank – Reserved, EPC, TID and User Tolerance - Maximum number of failed test repetitions Word pointer - address of the first word to be read Word counter - word count to be read Mandatory Write? – Use this checkbox the Write test should be performed with Mandatory Write instead of BlockWrite

[1] Supported Protocols: ISO 18000-6C, ISO 18000-6B, ISO 18000-6B-D (6B-Double), GB/T-29768

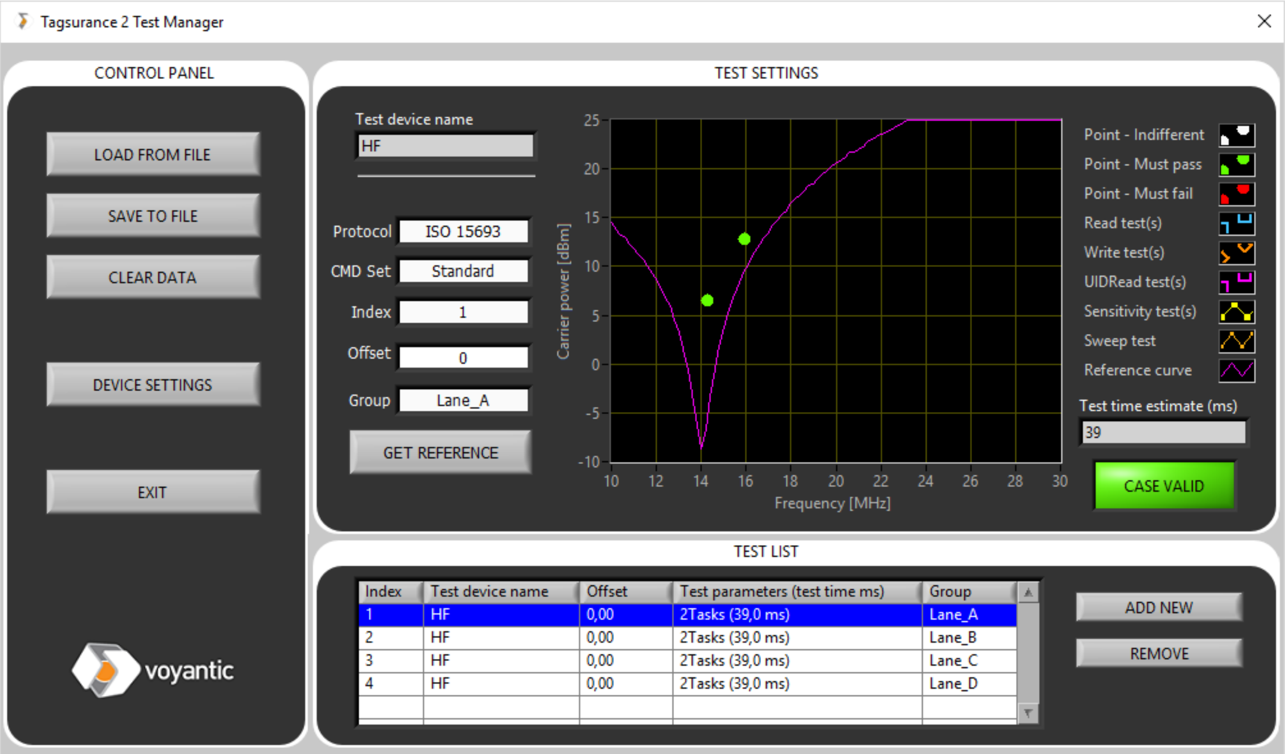
[2] Supported Protocols: ISO 18000-6C, ISO 18000-6B (Read only), ISO 18000-6B-D (Read only)

[3] Must respond, must not respond, indifferent (tag may or may not respond)

[4] Supported Protocols: ISO 18000-6C

Multiplexed tests

Tagsurance 2 supports multiplexing in a way that a single device performs tests on multiple lanes. In order to generate this kind of multiplexed test, choose the same test device for multiple test instances and add tests normally. Tests performed in each multiplexed case must be the same and therefore the changes made to the tests in one test instance are also automatically applied to other instances involving the same test device.



Test device name: HF

Protocol: ISO 15693

CMD Set: Standard

Index: 1

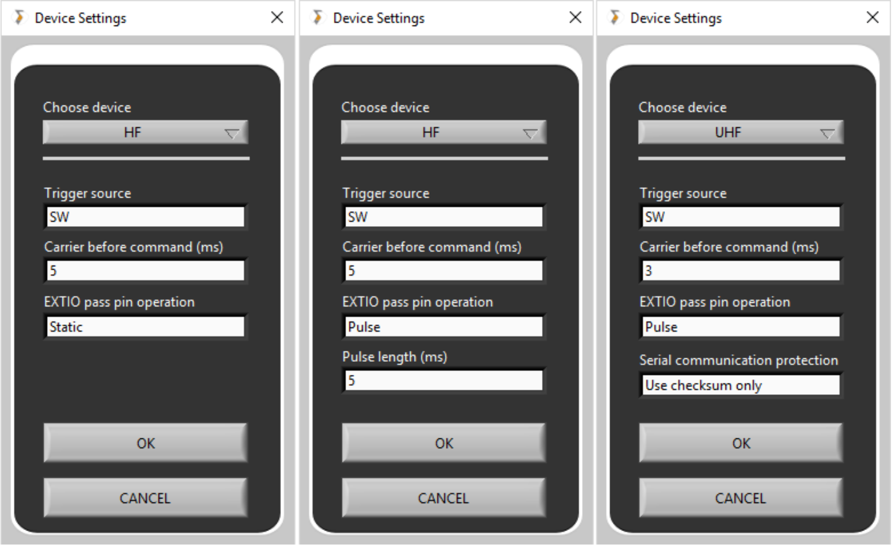
Offset: 0

Group: Lane_A

Test time estimate (ms): 39

Index	Test device name	Offset	Test parameters (test time ms)	Group
1	HF	0,00	2Tasks (39,0 ms)	Lane_A
2	HF	0,00	2Tasks (39,0 ms)	Lane_B
3	HF	0,00	2Tasks (39,0 ms)	Lane_C
4	HF	0,00	2Tasks (39,0 ms)	Lane_D

Figure 14: Multiplexed test is generated by including the same test device to multiple test instances in the test list. The test performed in each multiplexed instance must be identical



Device Settings

Choose device: HF
Trigger source: SW
Carrier before command (ms): 5
EXTIO pass pin operation: Static

Device Settings

Choose device: HF
Trigger source: SW
Carrier before command (ms): 5
EXTIO pass pin operation: Pulse
Pulse length (ms): 5

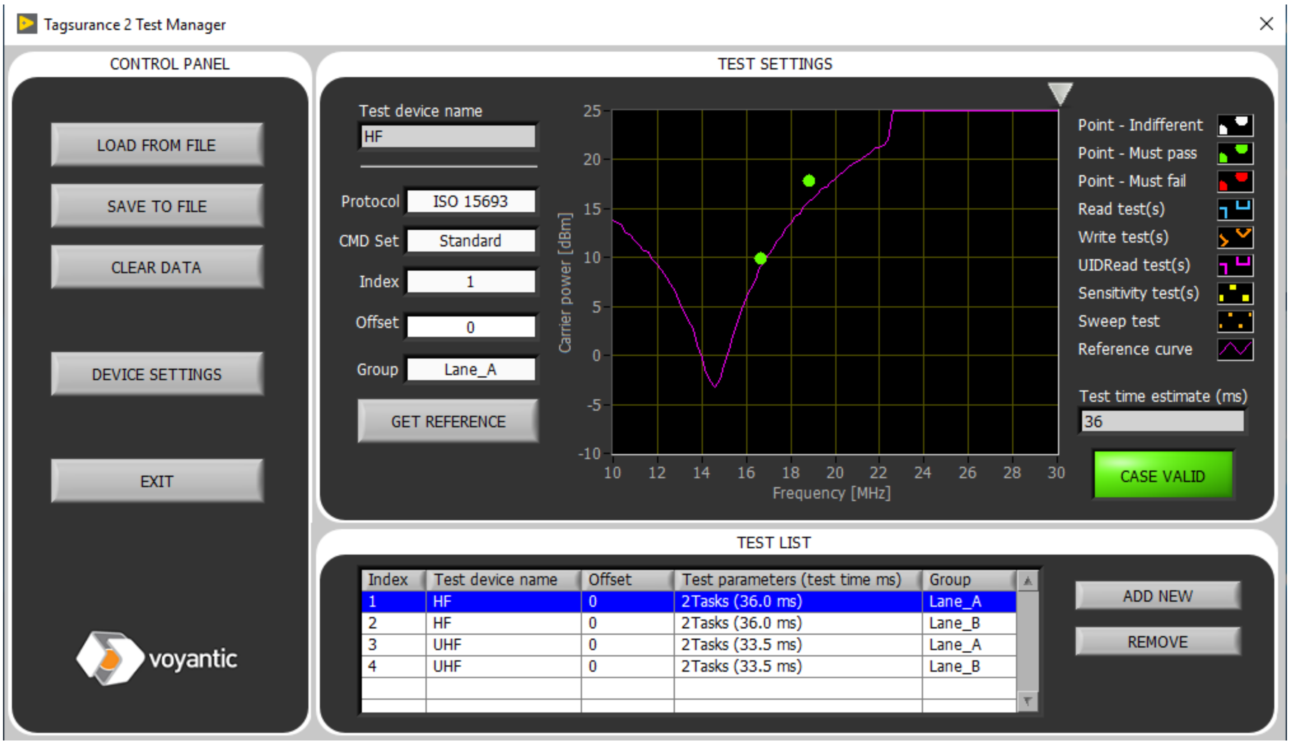
Device Settings

Choose device: UHF
Trigger source: SW
Carrier before command (ms): 3
EXTIO pass pin operation: Pulse
Serial communication protection: Use checksum only

Figure 15: Device settings dialog for different test device types and modes

Multi Device and Multi Lane support

Tagsurance 2 supports a multi device (both multiple UHFs as well as HFs) option within a single test recipe (a recipe may contain several test cases). It is also possible to create a Multi Lane system in which multiple devices act on the same lane. In order to associate two or more devices to the same lane, the same *GroupID* has to be assigned for each test system added to the instance list. It is important to note, that a Multi Device setup and a Multi Lane setup are mutually exclusive.



Tagsurance 2 Test Manager

Test device name: HF
Protocol: ISO 15693
CMD Set: Standard
Index: 1
Offset: 0
Group: Lane_A

Test time estimate (ms): 36

Index	Test device name	Offset	Test parameters (test time ms)	Group
1	HF	0	2Tasks (36.0 ms)	Lane_A
2	HF	0	2Tasks (36.0 ms)	Lane_B
3	UHF	0	2Tasks (33.5 ms)	Lane_A
4	UHF	0	2Tasks (33.5 ms)	Lane_B

Figure 16: Multi Device (1 UHF and 1 HF) and Multi Lane option

Test device settings

The final part of test definition is defining the test device settings. After selecting a device from the drop-down menu, definable parameters appear below depending on the type of device.

HF settings: Trigger source (SW, HW (MASTER) rising and HW (MASTER) falling) where SW trigger is a button on the Operator interface and HW triggers are external trigger signals provided to the I/O connector pins; carrier before command is a numeric value in milliseconds, default value is 5ms; EXTIO pass pin operation (Static, Pulse, Static Inverted, Pulse inverted) defines the Busy/Ready pin behavior (see Chapter 7.5).

UHF settings: Trigger source (SW, HW (MASTER) rising and HW (MASTER) falling) where SW trigger is a button on the Operator interface and HW triggers are external trigger signals provided to the I/O connector pins; carrier before command is a numeric value in milliseconds, default value is 3ms; EXTIO pass pin operation (Static, Pulse, Static Inverted, Pulse inverted) defines the Busy/Ready pin behavior (see Chapter 7.7); Serial communication protection (Use checksum only, Use framesync only, Use checksum and framesync, No protection).

Saving test

Once the test definitions have been completed, the test can be saved to a file, by clicking: ‘Save to file’. Once the file is saved it can be applied in the Tagsurance 2 Operator interface or re-opened for modification and checking in the Test Manager tool.

Important note!

Test case files can be saved practically anywhere, but if the test is saved to a folder in Tagsurance 2 test case folder (<system directory>\Tagsurance 2\Data\Test case files\), it will become available in a drop-down menu in the operator GUI. The tests can, for example, be saved in folders named after products. This will simplify test case application from the operator’s point-of-view. For more information, see Chapter 6.4.

Other features

Reference curve measurement

In order to simplify test task definition, it is possible to acquire frequency response curve for a tag. When clicking on “Get reference”, the software will try to connect to the associated device and perform a frequency sweep, displaying the results on the graph. The protocol depends on the protocol defined for the instance.

Test execution time estimation

Test Manager tool can estimate the execution time for the tasks associated for an instance. The execution time refers to the minimum time the tag must spend in the field of view of the coupling element. The test time depends on the number of tasks, type of tests, carrier frequency, and well as chip type (e.g. UID memory size). The estimate represents the worst-case scenario for a successfully tested tag. If accurate timing info is needed, it is recommended to test run the case and measure busy signal high time from the EXTIO connector of the Tagsurance HF or UHF. Equations used to estimate the test case execution time are presented in Appendix B.

Case Valid indicator (UHF devices only)

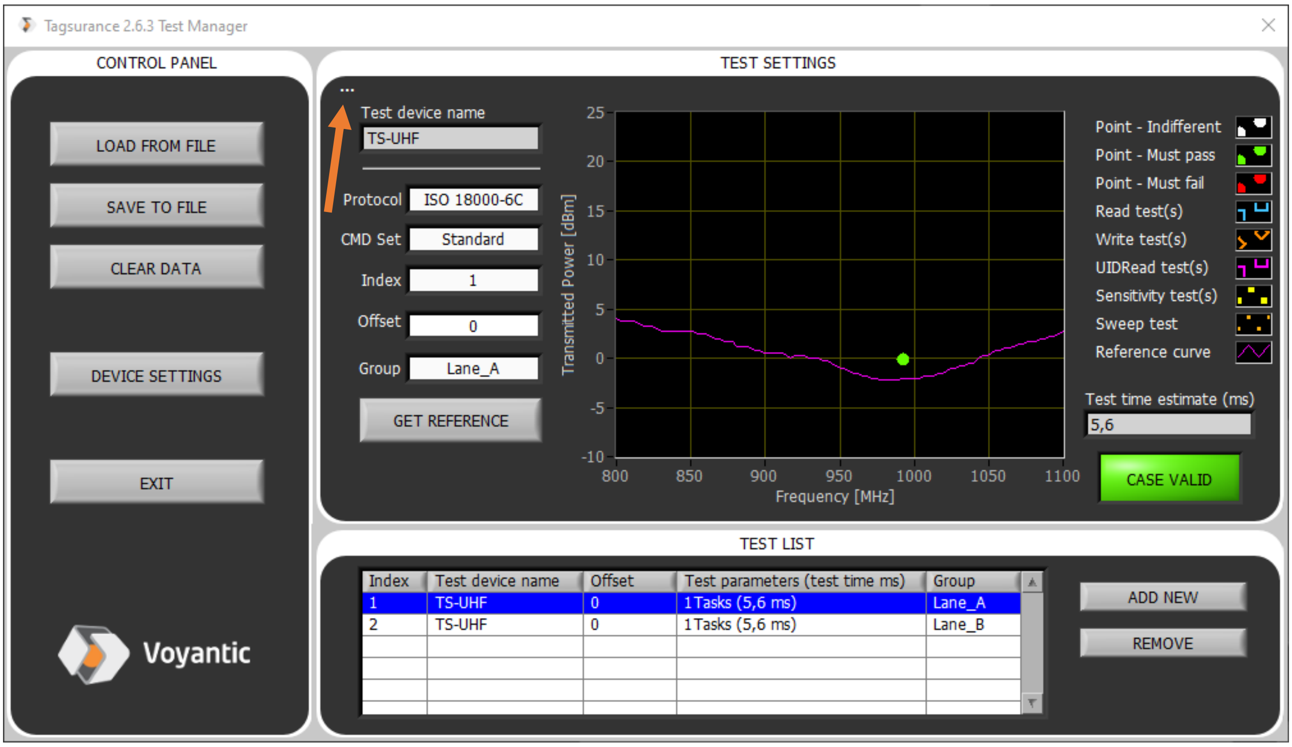
In order to be feasible, the case size must meet the following requirements:

- Test case definition shall not take more than 510 bytes, i.e.:
  $(3 \times N_{test\_points} + 3) + 7 \times N_{read\_tasks} + \sum(8 + 2 \times N_{words,j}) + 7 \times N_{sweep\_tasks} <= 510$,
  where $j=0…N_{write\_tasks}$
- Test result data size shall not exceed 100 bytes, i.e.:
  $ceil(N_{test\_points} / 8) + \sum(1 + 2 \times N_{words,l}) + N_{write\_tasks} +\sum((f_{end,k} / f_{start,k}) / f_{step,k} + 1) <= 100$,
  where $l = 0…N_{read\_tasks}$, $j = 0…N_{read\_tasks}$, and $k = N_{sweep\_tasks}$

The Case Valid indicator will turn red if the requirements above are not complied to.

Import legacy test case files (UHF devices only)

Test Manager tool allows the import of legacy test case files (previously generated with Tagsurance GUI Case Builder) by using the Import button on the right corner of the Test Manager front panel. Once a compatible test case has been selected (UHF device) from the Test List, press the Import button and select the case file to be loaded. If the file is correct, the selected test case will be automatically filled with all the tests available within the file.



Test device name: TS-UHF

Protocol: ISO 18000-6C

CMD Set: Standard

Index: 1

Offset: 0

Group: Lane_A

Test time estimate (ms): 5,6

Index	Test device name	Offset	Test parameters (test time ms)	Group
1	TS-UHF	0	1 Tasks (5,6 ms)	Lane_A
2	TS-UHF	0	1 Tasks (5,6 ms)	Lane_B

Figure 17: Legacy UHF test case files Import button

6.3 *Interface Manager*

Tagsurance 2 Interface Manager is a dedicated tool for setting the Operator Interface run-time position as well as the run-time position and size of all the plug-ins installed in the machine. Moreover, it allows the user to enable and disable specific plug-ins by pressing the arrow keys between the Enable and Disable tables. Interface Manager can be opened from Launcher Menu, by pressing the “Interfaces” button from the Settings category.

Once the interface has been defined and set, by pressing the “Save & Exit” button, the software will create a “Plug-In configuration.ini” file (Path: …Tagsurance 2/Data/Plug-Ins/Plug-Ins configuration.ini) in which all the position and size information will be stored.

In order to install a new plug-in, download it from the Voyantic Download page. Move the files to the Plug-Ins folder (…/Tagsurance 2/Data/Plug-Ins/).

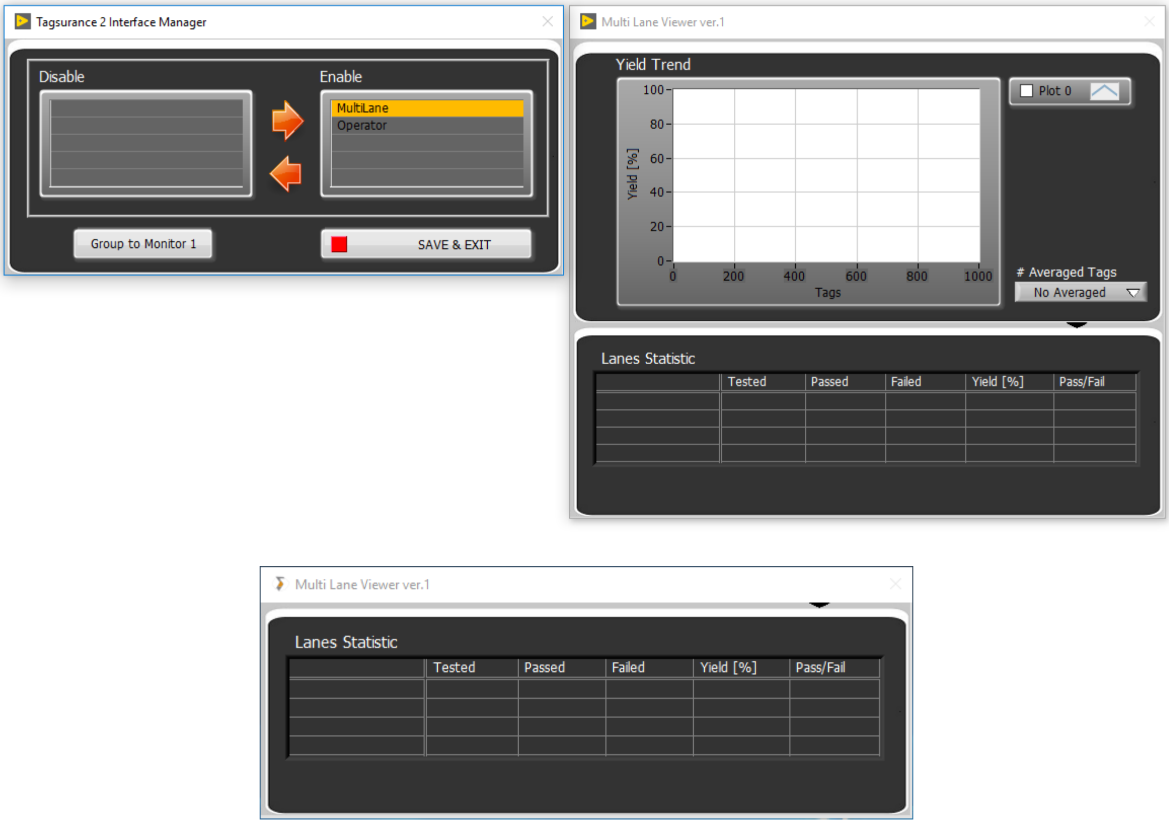


Figure 18: Interface Manager (left) and Multi Lane Viewer plug-in view (Full -Right & Statistic - Below)

Plug-Ins List

Plug-In	Plug-In description	Licensed
Multi Lane Viewer	Allows the user to monitor multilane statistics and Yield trend for each lane shown in the running test case.	Yes

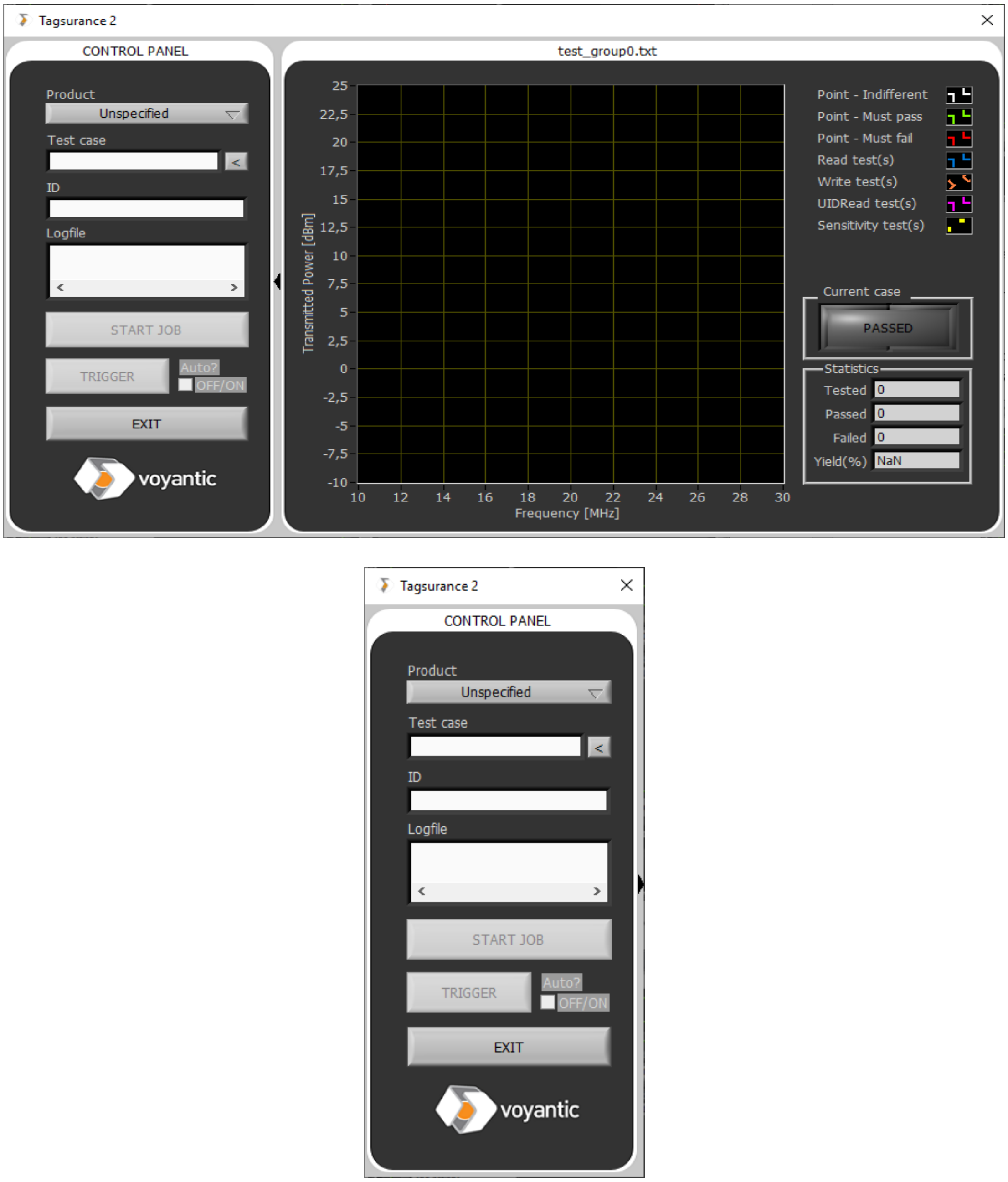
6.4 Operator Interface

Overview

Tagsurance 2 operator interface is a dedicated tool for testing tags with Tagsurance test devices (Tagsurance HF and Tagsurance UHF). Operator Interface can be opened from Launcher menu, or by running: ‘Tagsurance 2.exe’. The tool is intended for production purposes and it provides the comprehensive tools for performance test execution, test process monitoring, and test data logging.

Application of the Operator interface follows a simple procedure:

1. Select test to be performed (see Chapter 6.1 for details)
2. Apply job identifier to the ID field (optional)
3. Execute test (see Chapter 6.1 for details)
4. Select Graph/Full View (Graph View button)
5. Stop test and check data



Product: Unspecified

Test case

ID

Logfile

Tested 0

Passed 0

Failed 0

Yield(%) NaN

Figure 19: Tagsurance 2 GUI operator interface (Full View and Control View)

Important note!

When the tool initiates, it connects to all devices in the initialization file (Devices.txt) and returns to idle state if successful. If some devices cannot be found, the operator is prompted to open Device Manager tool, close the software, or to continue and ignore the connection issues. If the operator chooses to continue, the program will open in idle mode, and if “Exit” is chosen the program closes. If the operator chooses to open Device Manager, the Operator Interface will retry connection after Device Manager is closed.

Test selection

The first step in the testing process is to choose a test file.

Method 1

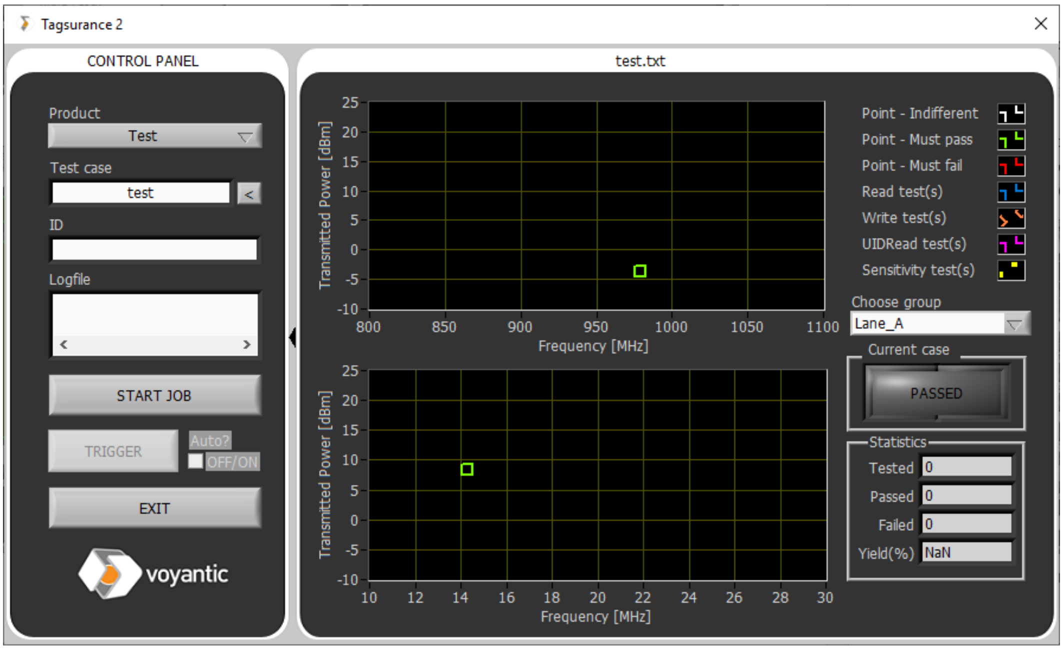
The operator can browse test files by clicking on the “<” button in the control panel. This will open a dialog where the user is prompted to select a test file. The file is tested, and an error indication is shown if the file is invalid. At this point errors may be caused by e.g. erroneous file syntax, or incorrect test device associations (i.e. associated device(s) cannot be found in Devices.txt).

Method 2 (recommended)

If the test files are arranged to folders according to e.g. product name, and the folder is located in Tagsurance 2 test case directory (<system directory>\Tagsurance 2\Data\Test case files\), the software locates them automatically and lists the products (folder names) to the Product dropdown menu. The related test files (located in product folder) populate the “Test case” drop down menu. In this case, the operator may first choose a product from the “Product” drop-down menu and the appropriate test file from the “Test case” drop-down menu. This is a recommended strategy since it simplifies the process from the operator point-of-view and reduces possibilities for human errors.

Job identifier

The ID field is intended for user input. The text from this field will be added with the test logfile name and can be used to help associating the test results e.g. with a certain LOT number. Data in this field can be either typed, or e.g. added by using a HID barcode reader.



Tagsurance 2

CONTROL PANEL

Product: Test

Test case: test

ID:

Logfile:

test.txt

Choose group: Lane_A

Current case: PASSED

Statistics

Tested 0

Passed 0

Failed 0

Yield(%) NaN

Figure 20: Test has been successful loaded and the system is ready for measurements

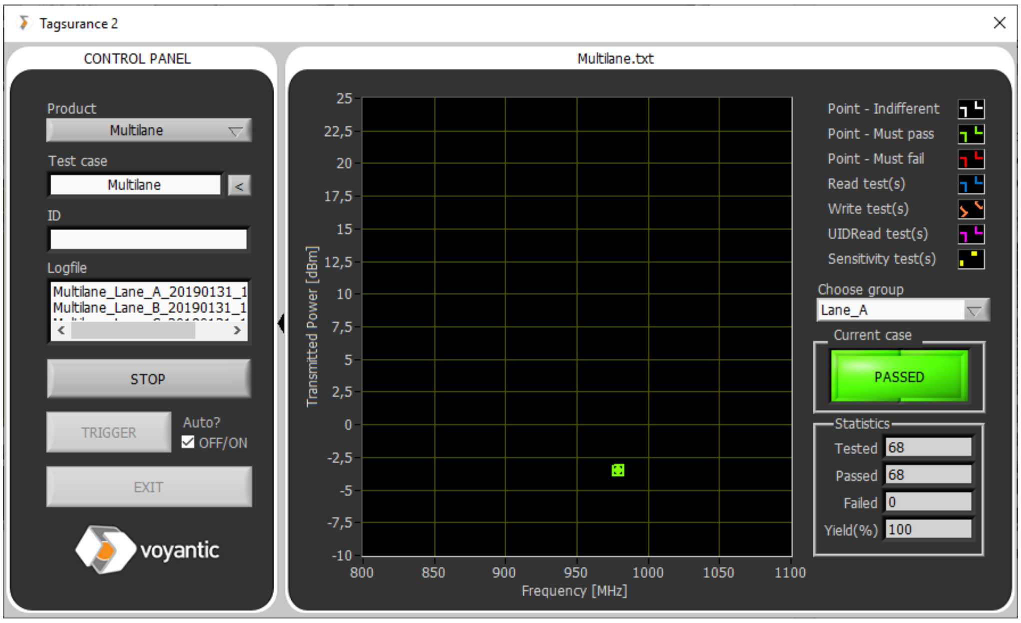
Test execution

Once the test case has been chosen successfully, the “START JOB” button is enabled, and the test can be performed by clicking it. The system will inform the operator if there are issues when uploading test specifications to the device(s) (e.g. license issues), or if the associated test device(s) cannot be connected.

After the test completion, the system is ready for use, and test data should be presented on the graphs. If the test is running with a software trigger, the operator must click: “TRIGGER” to trigger a test measurement or enable the “Auto?” checkbox in order to enable automatic SW trigger generation. The results are updated on the screen after the test has been completed. Triggers are provided for each instance one by one. If EXT IO trigger was chosen, the results will appear on the screen whenever group result data is ready.

Test data is presented in two different ways.

1. On the graph the latest test data received is shown from the test devices. There may be multiple graphs available depending on the test type.
    a. In a single group, single modality test, there is only one graph available.
    b. In a multimodal test there will be a separate graph for both modalities.
    c. For multi group (typically multilane) a control appears on the right side of the graph where the operator may choose which lane is monitored (by installing the “Multi Lane Viewer” plug-in, it will possible to monitor the statistics of all the lanes as shown in Figure 22)
2. Test statistics are shown on the lower-right corner which indicates the total number of tags tests, total yield, and the number of failed and passed tags separately.



Product: Multilane

Test case: Multilane

ID:

Logfile:
Multilane_Lane_A_20190131_1
Multilane_Lane_B_20190131_1

Auto? ☑ OFF/ON

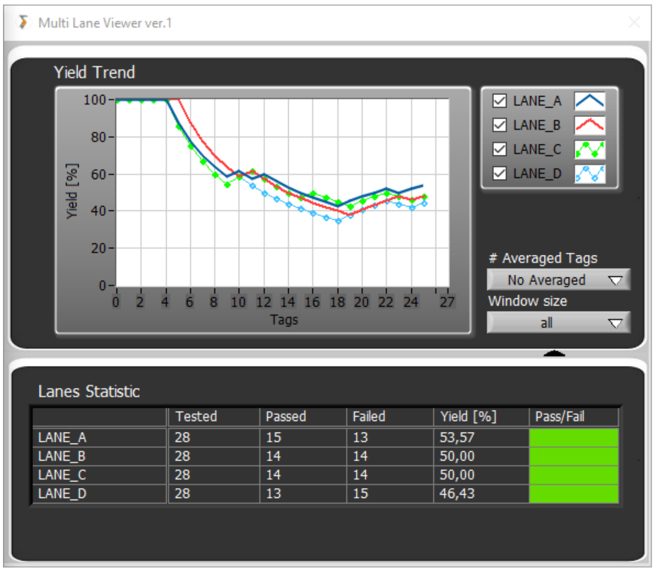
Choose group: Lane_A

Current case: PASSED

Statistics

Tested	68
Passed	68
Failed	0
Yield(%)	100

Figure 21: Test is on and the software is acquiring data from the test device (UHF)



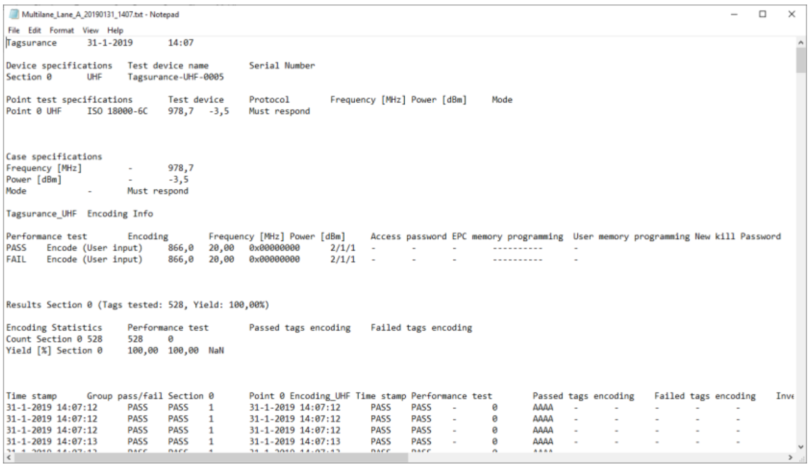
	Tested	Passed	Failed	Yield [%]	Pass/Fail
LANE_A	28	15	13	53,57	🟩
LANE_B	28	14	14	50,00	🟩
LANE_C	28	14	14	50,00	🟩
LANE_D	28	13	15	46,43	🟩

Figure 22: Multi Lane Viewer Plug-In

Measurement log files

During the measurement, the measurement data is automatically streamed to a log file, which can be used for achieving and performing analysis. The file contains: date, time, task specifications, encoding settings, test devices performing the tests, results from each test performed and the final Passed/Failed statistics (Performance Test and Encoding statistics are shown separately).

The log file is automatically named and saved to Tagsurance 2 output data folder (<system directory>\Tagsurance 2\Data\Output files\). If the operator has browsed the test file without product association, the log file is found in a folder named: ‘Undefined’. If the operator has chosen the test file through the Product menu, the log file will be located in a folder after the selected Product. The test file name indicates product name, test group, date, and time. If there were multiple test groups, the GUI generates one log file for each group.


Multilane_Lane_A_20190131_1407.txt - Notepad
File Edit Format View Help

```
Tagsurance      31-1-2019       14:07

Device specifications   Test device name        Serial Number
Section 0       UHF     Tagsurance-UHF-0005

Point test specifications       Test device     Protocol        Frequency [MHz] Power [dBm]     Mode
Point 0 UHF     ISO 18000-6C    978,7   -3,5    Must respond



Case specifications
Frequency [MHz]         -       978,7
Power [dBm]             -       -3,5
Mode            -       Must respond

Tagsurance_UHF  Encoding Info

Performance test        Encoding        Frequency [MHz] Power [dBm]     Access password EPC memory programming  User memory programming New kill Password
PASS    Encode (User input)     866,0   20,00   0x00000000      2/1/1   -       -       -       ----------      -
FAIL    Encode (User input)     866,0   20,00   0x00000000      2/1/1   -       -       -       ----------      -



Results Section 0 (Tags tested: 528, Yield: 100,00%)

Encoding Statistics     Performance test        Passed tags encoding    Failed tags encoding
Count Section 0 528     528     0
Yield [%] Section 0     100,00  100,00  NaN



Time stamp      Group pass/fail Section 0       Point 0 Encoding_UHF Time stamp Performance test        Passed tags encoding    Failed tags encoding    Inve
31-1-2019 14:07:12      PASS    PASS    1       31-1-2019 14:07:12      PASS    PASS    -       0       AAAA    -       -       -       -       -
31-1-2019 14:07:12      PASS    PASS    1       31-1-2019 14:07:12      PASS    PASS    -       0       AAAA    -       -       -       -       -
31-1-2019 14:07:12      PASS    PASS    1       31-1-2019 14:07:12      PASS    PASS    -       0       AAAA    -       -       -       -       -
31-1-2019 14:07:13      PASS    PASS    1       31-1-2019 14:07:13      PASS    PASS    -       0       AAAA    -       -       -       -       -
```

Figure 23: Example of test log data output file

7 Language packs

7.1 Overview

Language packs enable the use of Tagsurance 2 with other languages. The following sections detail how to utilize this feature.

7.2 GUI

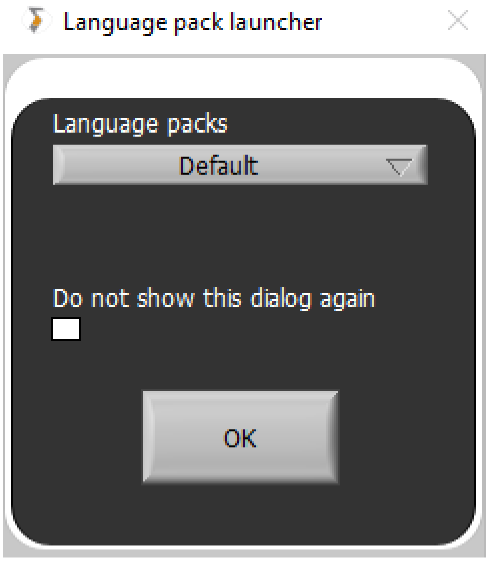
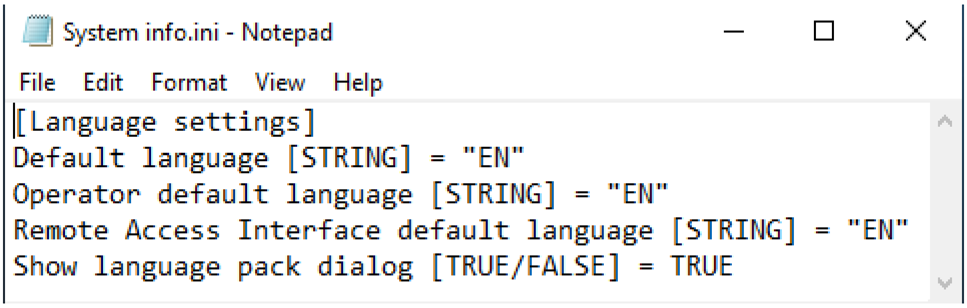


Figure 24 Language dialog

When 'Tagsurance 2 Launcher' is ran, the Language dialog (Figure 24) is shown. The drop-down menu shows all language pack files found in the language pack directory (<system directory>\Tagsurance 2\Data\Language packs). The 'Default' option will start the software in English. If it is expected that the software will be used with a single language, select that language, and tick the 'Do not show this dialog again' -box. From that point onwards the software will start in that language without showing the Language dialog. If the selected language file cannot be loaded the user is notified of the error and the software is loaded in English.

7.3 Configuration



```
[Language settings]
Default language [STRING] = "EN"
Operator default language [STRING] = "EN"
Remote Access Interface default language [STRING] = "EN"
Show language pack dialog [TRUE/FALSE] = TRUE
```

Figure 25 System info.ini default contents

System info.ini (Figure 25) is used for configuring the language pack selections. The file is located in (<system directory>\Tagsurance 2\Data).

Tagsurance 2 Launcher

If the language selection dialog should be shown when 'Tagsurance 2 Launcher' is ran, change the "Show language pack dialog [TRUE/FALSE]" to TRUE. Selecting a language value for "Default language" will make it the default selection in the language dialog (Figure 24).

Tagsurance 2

Changing the value for "Operator default language" will make the software start with the selected language.

Tagsurance 2 RAI

Changing the value for "Remote Access Interface default language" will make the software start with the selected language.

Configuration

Use these values to select a language pack:

- English – en
- Chinese – cn
- German – de
- Spanish – es
- Finnish – fi
- Other – see the documentation included with the language download

If editing System info.ini does not yield the correct results, try deleting it and then launching any Tagsurance 2 software. This will re-create this file with the default value as seen in Figure 25. Try making the changes again to see if the issue persists.

7.4 Updating existing installations

Each Tagsurance software package contains the latest language packs. If new languages or revisions to existing languages are introduced between releases, these can be downloaded from the Voyantic download site. Place the downloaded files to the language pack directory.

Troubleshooting

The software is built to recover from many errors that may occur between and during the measurement. However, if something unexpected happens, the software will display an error dialog, which shows the possible error code, explains it and prompts the user for further actions.

Error codes from Tagsurance 2 are described in Chapter 9.2 (from Tagsurance HF) and Chapter 10.7 (from Tagsurance UHF) and the typical error situations are described below:

Some of the most common error situations are described below

1. Software installation fails:

    a. Ensure that user privileges do not restrict installation process

2. Connection issues with the test device

    a. Check setup and connections, and reset device (see Chapter 5.4 for more information)

    b. Check device associations (see Chapter 6.1 for more information)

    c. Check that the right device is associated with test (see Chapter 6.2 for more information)

3. No test results are received, or test cannot be started

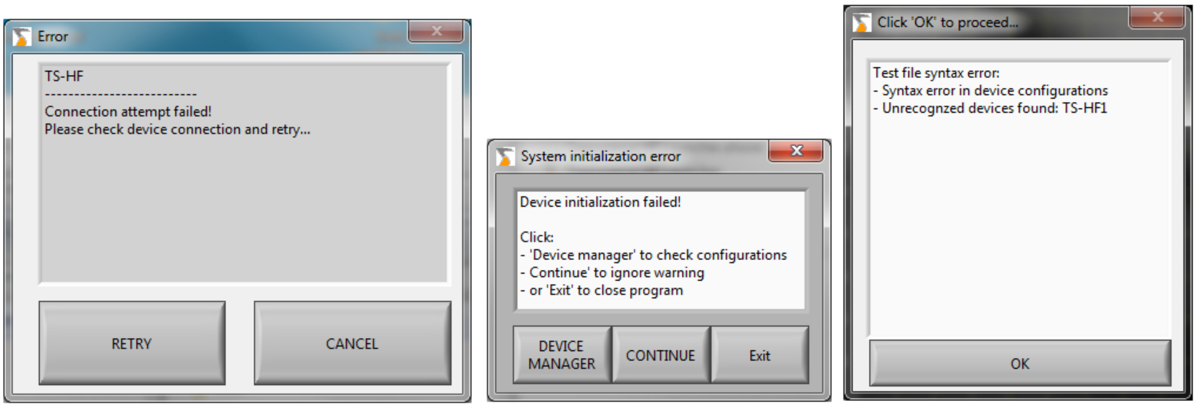
    a. Check Ethernet cable (see Chapter 5.4 for more information)

    b. Check serial cable (see Chapter 5.4 for more information)

    c. Check device settings related to the test (see Chapter 6.2 for more information)

    d. Check that you have appropriate license (see Chapter 3.2 for more information)

4. If the issue does not get solved, please contact: support@voyantic.com



Error

TS-HF

Connection attempt failed!
Please check device connection and retry...

System initialization error

Device initialization failed!

Click:
- 'Device manager' to check configurations
- Continue' to ignore warning
- or 'Exit' to close program

Click 'OK' to proceed...

Test file syntax error:
- Syntax error in device configurations
- Unrecognzed devices found: TS-HF1

Figure 26: Error indications related to device connection issues and erroneous device definitions

5. Wrong Channel Detection signal (Multilane configuration) will be shown on the Tagsurance 2 GUI and It may happen in certain circumstances:

a. One of the devices is not able to calculate the pulse width correctly from the IOBox

b. Hardware failure -> Check the connection between the IOBox and Tagsurance device

c. Wrong Multilane IOBox settings -> Check the IOBox controller configuration

d. The trigger signal to the IOBox is not “clean” (possible glitches or voltage drops) -> Check the trigger source

EXTIO Signaling

7.5 Tagsurance HF version 3.x

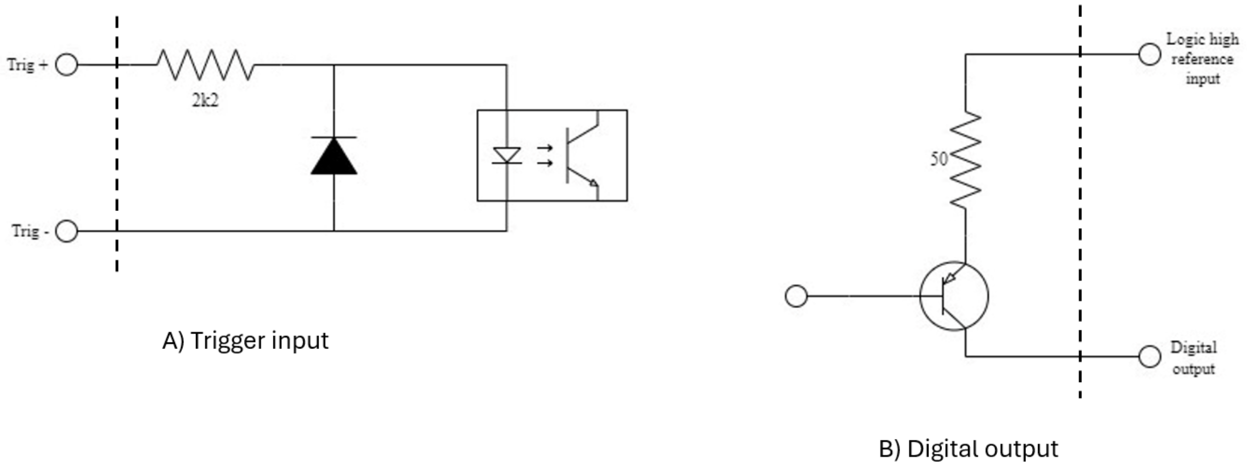
Tagsurance HF version 3.x has DA-15 female connector where EXTIO signals can be connected. The connector pinout and electrical connection interfaces are described below.

Pin	Function	Pin	
1	N.C.	**9**	RFU
2	RFU	**10**	RFU
3	GND	**11**	+5 VDC[1]
4	GND	**12**	Logic 0-level reference
5	TRIG -	**13**	Logic 1-level reference
6	TRIG +	**14**	RFU
7	Busy/Ready	**15**	RFU
8	Pass/Fail	**Shield**	GND

1) The maximum output current is 20 mA

The high and low levels of the output pins 7 and 8 are set by the pins 12 and 13. The voltage level of pin 13 must be higher than the voltage of pin 12.

The trigger signal is a 5-24 V differential signal between pins 5 and 6. If a unipolar signal is preferred, pin 5 can be grounded to either pin 3 or pin 4.



A) Trigger input

B) Digital output

Figure 27: EXTIO signal electrical connection interfaces in Tagsurance HF version 3. x

7.6 Tagsurance HF version 1.x & 2.x

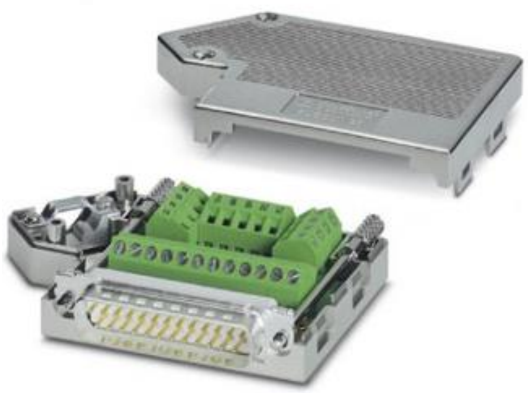


Figure 28: DB25 EXTIO connector that can be used to connect signals to the Tagsurance HF EXTIO port

Pin	Function	Pin	
1	Power input (18-24VDC)	14	+5V out
2	Power input (18-24VDC)	15	Logic high reference input
3	Power GND	16	Trigger threshold level
4	Power GND	17	Trigger input
5	Ext Power enable (active low)	18	Busy/ready output
6	Digital output (RFU)	19	Pass/fail output
7	Digital output (RFU)	20	Digital output (RFU)
8	Digital output (RFU)	21	Digital output (RFU)
9	Digital output (RFU)	22	Digital output (RFU)
10	Digital input (RFU)	23	Digital output (RFU)
11	Signal ground	24	Digital input (RFU)
12	Analog input positive (RFU)	25	Digital input (RFU)
13	Analog input negative (RFU)	Shield	GND

Pins 1-5 are related to application of external power source. It can be connected to pins 1-4. Pin 5 is external power enable pin and needs to be grounded when external power source is applied.

Pins 6-10 are 3V3 compatible logic pins (reserved for future use).

Pin 11 is signal ground.

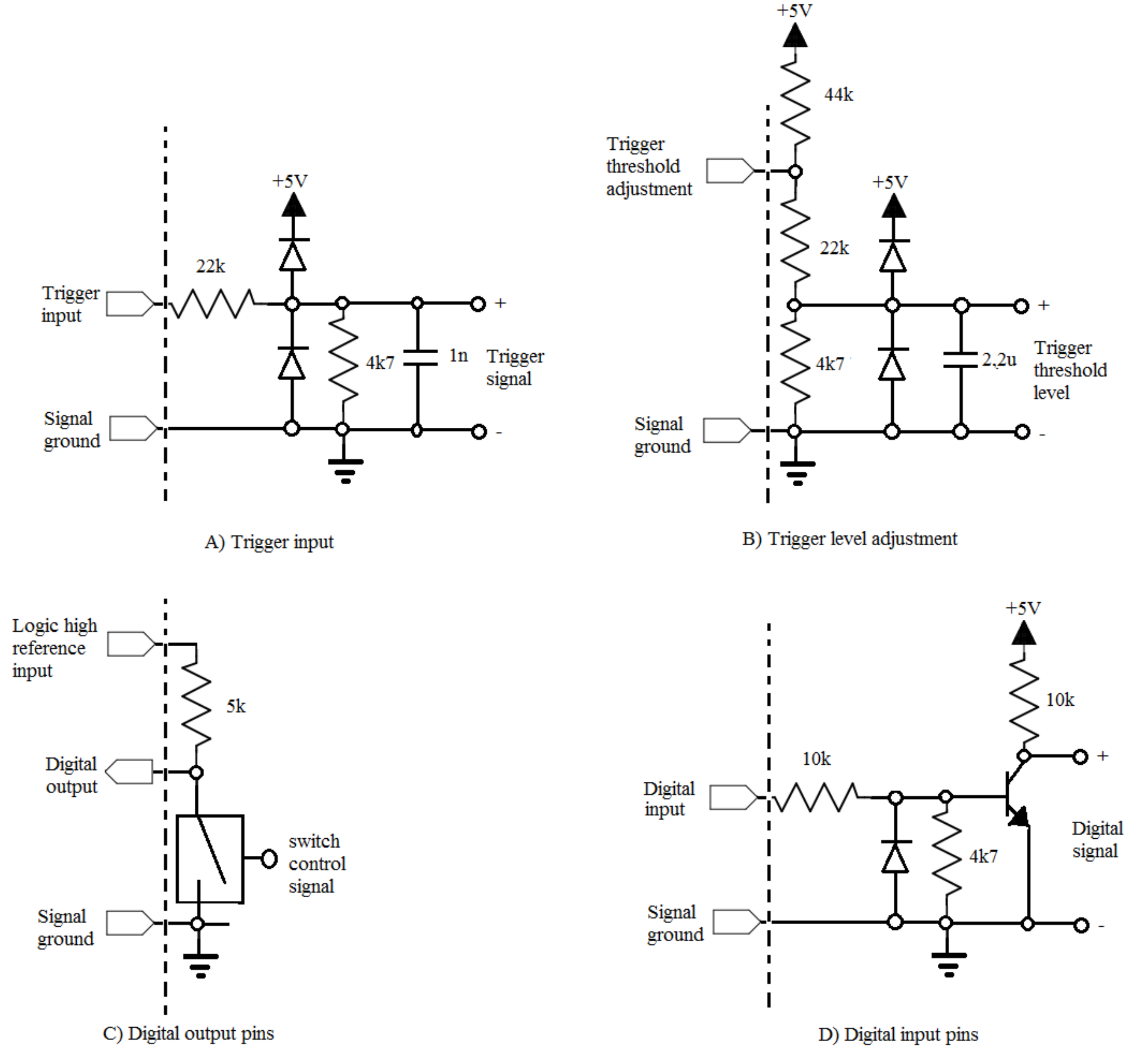
Pins 12-13 are differential analog input pins relative to signal ground at Pin 11 (Reserved for future use).

Pin 14 is an isolated +5Vdc, 100mA voltage output.

Pin 15 is logic high level input pin relative to pin 11. It sets the high voltage level at output pins 18-23. When these pins are used, pin 11 should be connected to ground and DC level at pin 15 must be 0V or higher.

Pin 17 is unipolar trigger input and compatible with voltages from 3,3V up to 24V. Minimum trigger pulse length is 5µs.

Pin 16 is trigger threshold adjustment pin. Default trigger threshold level is at 38% (pin 16 is floating). It can be adjusted by adding a pull-up resistor to +5V output (increases threshold), or pull-down resistor to signal ground (lowers threshold level). For details, please see Figure 29.



A) Trigger input

B) Trigger level adjustment

C) Digital output pins

D) Digital input pins

Figure 29: EXTIO signal electrical connection interfaces in Tagsurance HF version 2.x

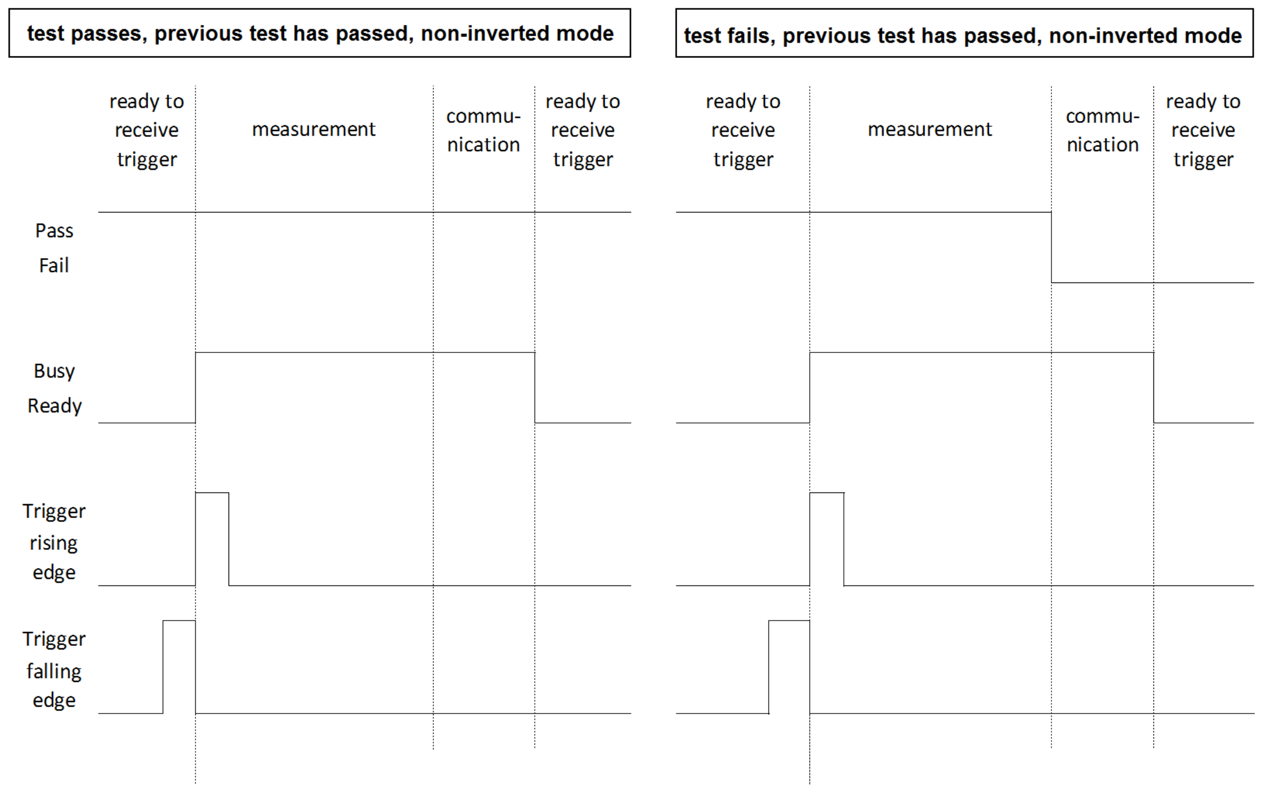


Figure 30: Tagsurance HF (all versions) signal timing diagrams, extio pass signal in static mode

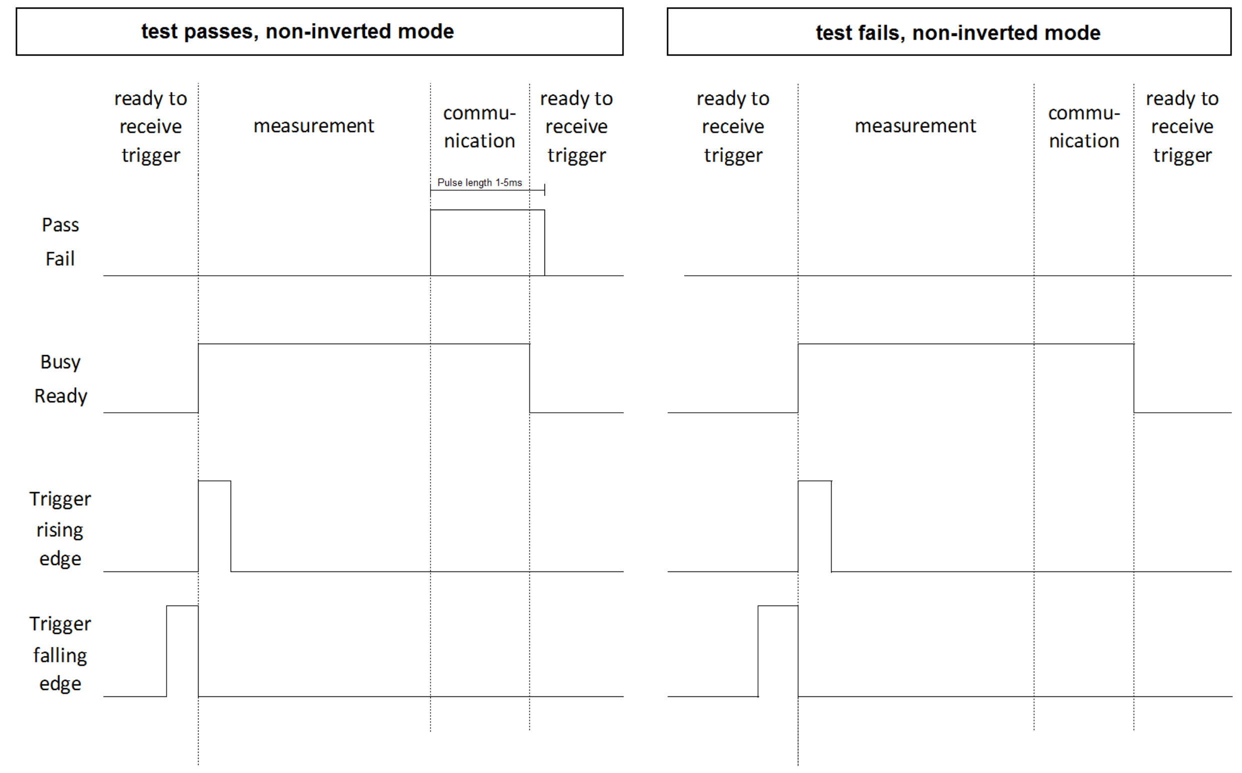


Figure 31: Tagsurance HF (all versions) signal timing diagrams, extio pass signal in pulse mode

7.7 Tagsurance UHF

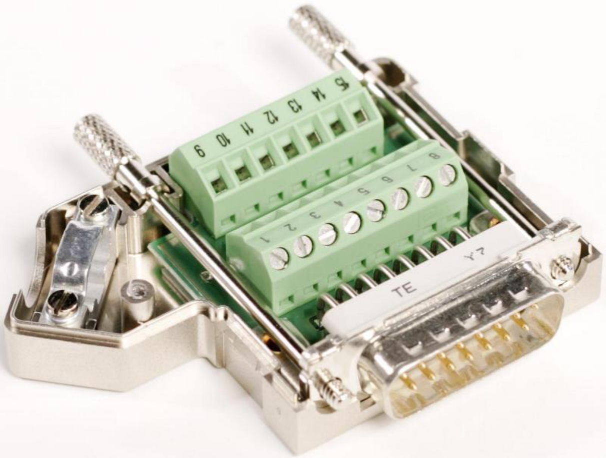


Figure 32: External I/O connector

Pin	Function	Pin	
1	Power input (+18VDC)	**9**	External power supply enable input (active low)
2	Load and run case 1 input (active low)	**10**	Firmware update enable input (active low)
3	Power GND	**11**	+5VDC output (maximum current: 100mA)
4	Power GND	**12**	Logic 0-level reference input
5	External trigger input -	**13**	Logic 1-level reference input (max voltage: 24V)
6	External trigger input +	**14**	Output: (reserved for future use)
7	Busy/Ready output (busy=HIGH)	**15**	Output: (reserved for future use)
8	Pass/Fail output (default: pass=HIGH)		

The high/low levels of the output pins 7, 8, 14 and 15 are set by the pins 12 and 13. The voltage level of the pin 13 must be higher than the voltage of pin 12.

The trigger signal is a 5-24V differential signal between pins 5 and 6. If a unipolar signal is preferred, pin 5 can be grounded to either pin 3 or pin 4.

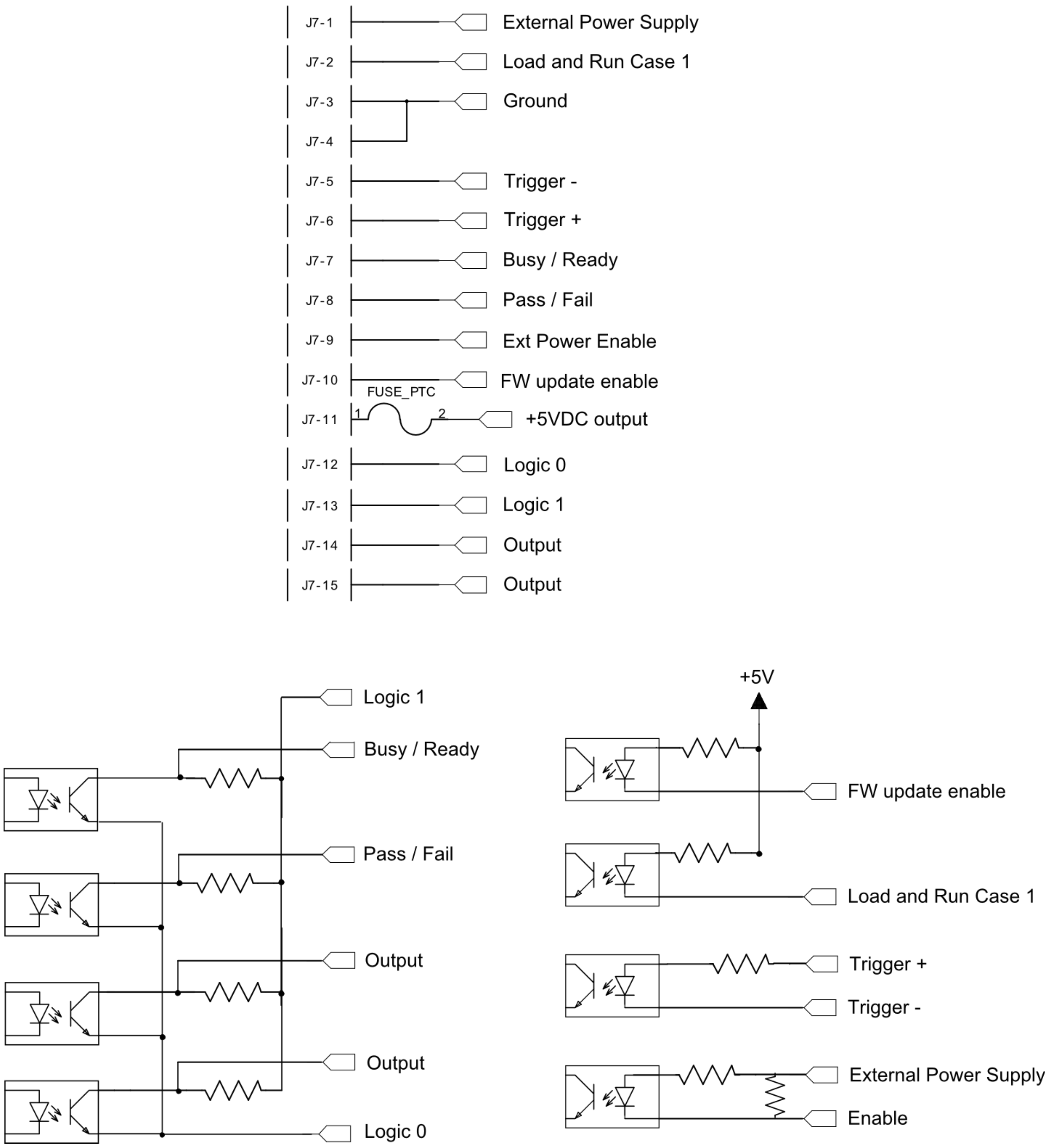


Figure 33: I/O schematic drawing

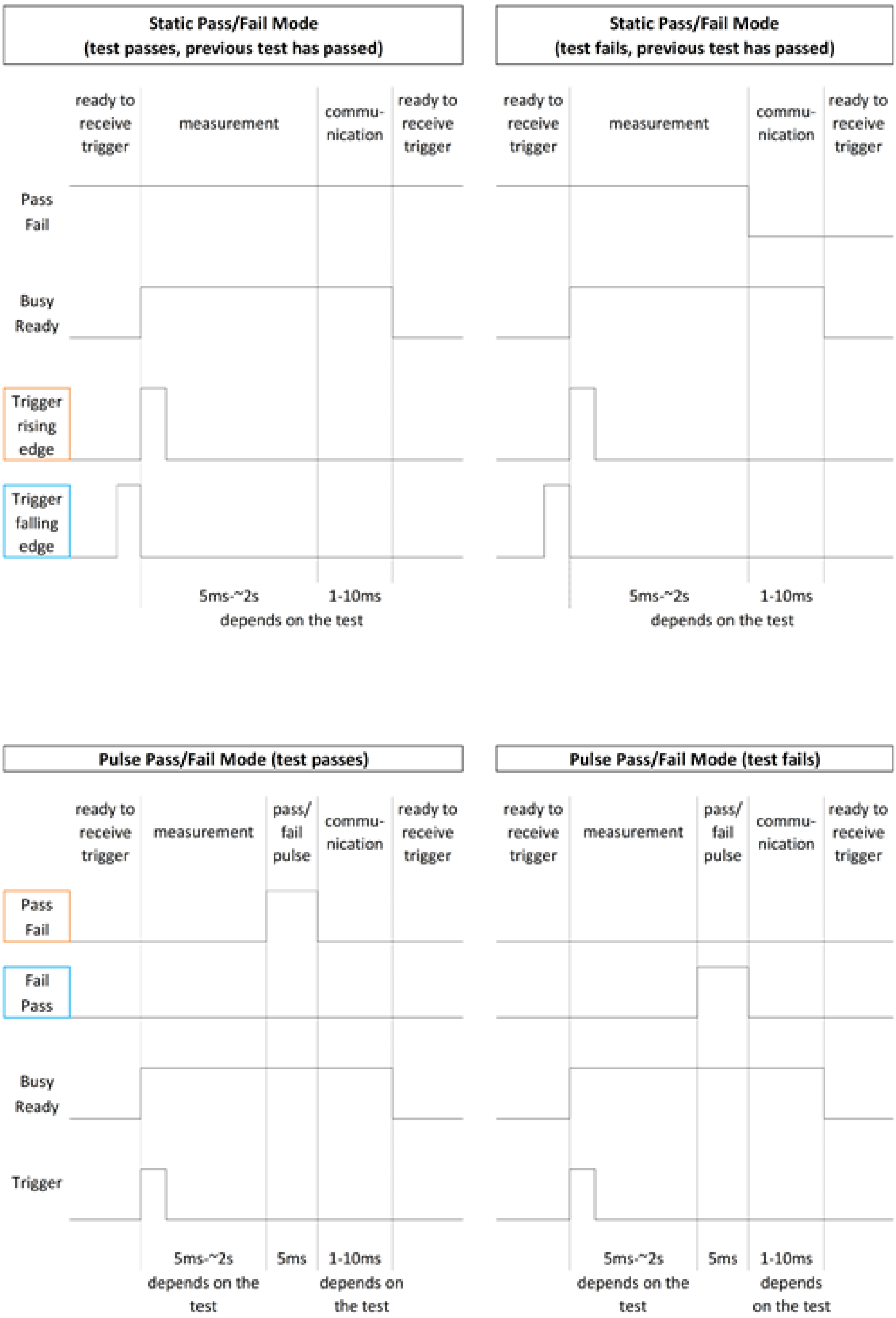


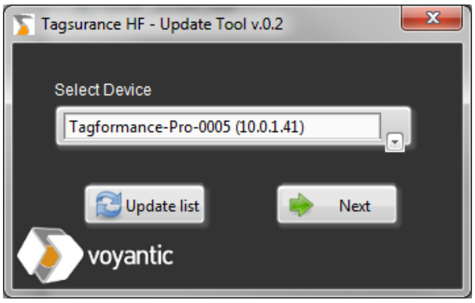
Figure 34:Tagsurance UHF signal timing diagrams

8 Firmware update procedure

8.1 *Tagsurance HF*

Tagsurance HF Update Tool is a dedicated software for updating device firmware, license and calibration data, or IP settings. To update the device firmware or license, follow the instructions below:

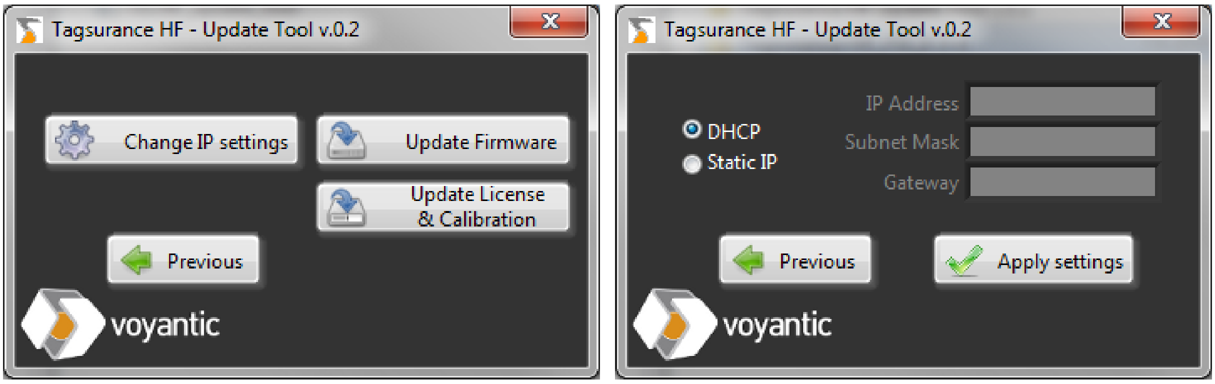
1. Power on the Tagsurance unit and connect it to the same network as the PC
2. Start the update software on the USB stick by executing: Tagsurance Update Tool\Update_Tool.exe
3. From the startup screen, choose device and click “Next” button to continue. If device does not appear on the drop-down list, check cabling and click “Update list” to refresh the device list.



Select Device: Tagformance-Pro-0005 (10.0.1.41)

Figure 35: Tagsurance HF update tool startup screen

4. In the next screen (Figure 36) it is possible to choose the update action to be done.
   a. To update device firmware, click: “Update Firmware” and browse file (Tagsurance-HF_FW_XX-XX-XX.zip).
   b. To update the license, click: “Update License & Calibration” and browse the correct license file (.lic). License file contains both calibration and license data.
   c. To change IP settings, click “Change IP settings”, and choose mode (DHCS or static IP). Then click: “Apply settings” to save changes, or “Previous” to cancel operation.
5. Once the action is chosen, wait until the software indicates that the operation has been completed.



● DHCP
○ Static IP

Figure 36: Tagsurance HF update options (left) and IP settings options screen (right)

8.2 Tagsurance UHF

The USB memory stick delivered with the device contains an Update Tool, which is used to update either the device firmware or the license. The Update Tool is located in the “Tagsurance Update Tool” folder. To update the device firmware or license, follow the instructions below:

1. Power on the Tagsurance unit and connect to the PC with serial cable
2. Start the update software by executing: Tagsurance Update Tool\Update_Tool.exe The following interface will appear:

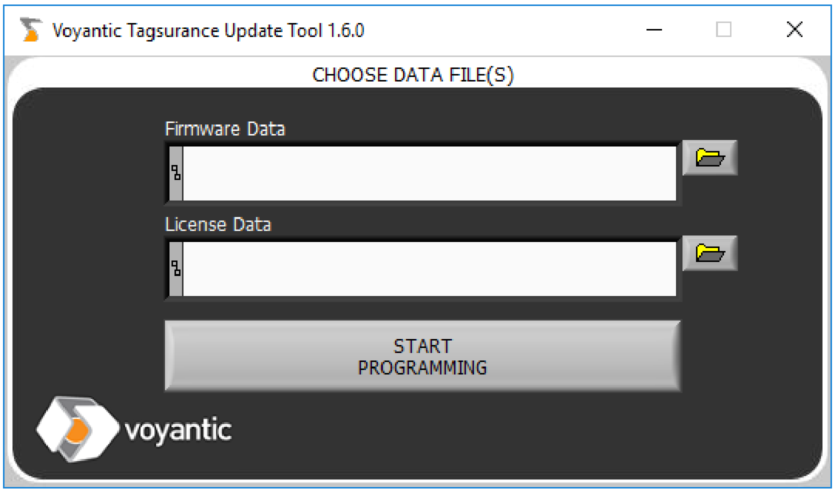


Figure 37: Tagsurance Update Tool Interface

3. To update the firmware, insert the file path in the “Firmware Data” field or browse and define the file path by clicking the folder icon on the right of the “Firmware Data” field

   To update the license, insert the file path in the “License Data” field or browse and define the file path by clicking the folder icon on the right of the “License Data” field

4. Press the “START PROGRAMMING” button in the interface
5. After receiving an instruction window pop-up, press down the programming button in the back panel and recycle device power while keeping the button pressed.
6. Press ‘ok’ button in the instruction window
7. The program will run the update procedure and inform when the update is ready

9 Operation Using the Tagsurance HF TCP Protocol

Tagsurance HF is controlled over an Ethernet connection. Tagsurance HF acts as a TCP-server, and the client can control the measurement activities by using a specific command set.

The connection establishment procedure is described in chapter 9.1 and the TCP-commands in chapter 9.2.

9.1 Device Connection Establishment

To access Tagsurance HF, follow these instructions:

1. Connect an Ethernet cable to the device and power the Tagsurance HF test unit up. Device initialization will take some 20-30s. Once completed, the unit will emit a beep sound and the power LED turns brighter. The device is now ready for connecting.

2. Establish connection
    - Client opens TCP connection (test unit IP address is needed, port number is 54321)
    - Tagsurance HF recognizes the connection attempt, which is indicated by light flash.
      The device completes the initialization and moves to the idle state to wait for commands.
    - Client can test the connection by transmitting 'TCP Test' (=00 00 00 04 00 F0 00 00)

   For a valid command, the Tagsurance HF responds with 'TCP ready' (=00 00 00 02 00 F1).
   → Tagsurance HF is activated and ready to use.
   For an invalid command, the Tagsurance GUI responds with 'ERR' (= 00 00 00 03 00 FF xx).
   Else (i.e. if timeout occurs)
   → Connection attempt is terminated, and the Tagformance-HF is not activated.

3. After the connection establishment has been successfully completed, the Tagformance-HF is ready to be used. This is indicated in the Tagsurance HF front panel by the 'Ready' indicator led.

9.2 *Commands Description*

Control over the network is made by sending specific commands which initiates certain activities.

The commands consist of three parts: 32-bit length data, 16-bit command data, and the possible command parameters (see table below). The length bytes indicate the length of the data (i.e. the amount of parameter bytes + 2 command bytes). Data is in hexadecimal format and numbers are sent high byte first.

Length Byte 1 (high)	Length Byte 2	Length Byte 3	Length Byte 4 (low)	Command Byte 1 (high)	Command Byte 2 (low)	Parameters (N data bytes)
0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	...

List of available commands is given below. Detailed description is provided in the next chapters.

CODE	COMMAND	DESCRIPTION	INIT STATE	NEXT
0x0010	LTC	Load Test Case	idle	no change
0x0011	TCL	Test Case Loaded	-	-
0x0012	STC	Start Loaded Test Case	idle	inline/idle[1]
0x0013	TCS	Test Case Started	-	-
0x0014	STOP	Stop running case	inline	idle
0x0015	STOPPED	Case stopped	-	-
0x0021	MCHD	Enable Multilane Channel Detection	idle	no change
0x0022	MCHE	Disable Multilane Channel Detection	-	-
0x0023	MCHR	Multi Channel Ready Response	-	-
0x0030	POINT	Run a single Point Test	idle	no change
0x0031	SWEEP	Measure tag's response threshold curve	idle	no change
0x0033	UIDREAD	Read tag's ID[5]	idle	no change
0x004A	CARRIER	Turn the Carrier On/Off	idle	no change
0x001A	TRIG	Software trigger to Tagsurance HF Tester Unit	inline	inline
0x001B	TRIGGERED	Software triggering successful (and device is ready)	-	-
0x001F	TR	Test Results	inline/idle[2]	
0x00F0	TCP Test	TCP Connection Test	idle	idle[3]
0x00F1	TCP Ready	State of TCP connection[4]	-	-
0x00FF	ERR	Error from Tagsurance	-	-

[1] If start command fails, Tagsurance HF will return an error and stay in the idle state.

[2] Inline test results are sent after the test is completed (TR frame).

[3] Command does not change device state

[4] Tagsurance HF accepts this command in inline state but does not answer to it.

[5] ISO 15693: UID, ISO 14443-A: UID, ISO 18000-3M3: TID, ISO 14443-B-ST25TB: UID

LTC (Load Test Case)

INIT STATE: idle
NEXT STATE: idle
FUNCTION: loads test case to device and validates its contents
TAGSURANCE RESPONSE: TCL, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0xXX	Length byte 1 (MSB)
1	Length byte 2	0xXX	Length byte 2
2	Length byte 3	0xXX	Length byte 3
3	Length byte 4	0xXX	Length byte 4 (LSB)
4	Command	0x00	Load Test Case (LTC), high byte
5	Command	0x10	Load Test Case (LTC), low byte
6	Number of tasks	0xXX	Number of tasks include to the test
7…	Configuration task list[1]	…	Task definitions list (for details, see table below) [TaskID][Ndata,high][Ndata,low][Data byte 1]….[Data byte N]
…	Test task list[2]	…	Task definitions list (for details, see table below) [TaskID][Ndata,high][Ndata,low][Data byte 1]….[Data byte N]

[1] Configuration tasks are optional. Default values will apply if no configuration tasks are defined.

[2] List should start with wait for trigger, end with send results, and number of tests should be at least 1.

Configuration tasks:

Configuration task are optional commands that can be used to configure general testing options.
If no configuration tasks are defined, the default values indicated in the table below will apply.

Configuration Task	TaskID	Length	Data
Set Carrier before command (default: 5ms)	0xA0	0x04	[Time[1]]
Set EXT-IO pass-fail indication mode (default: pass=high, static)	0xB0	0x03	[Inversion[2]] [Pulse mode[3]] [Pulse time[4]]
Set trigger signal parameters (default: 5µs)	0xB1	0x01	[Minimum length[5]]

[1] Time format: microseconds (µs), 32-bits (MSB first)

[2] Inversion byte: pass-signal is high (0x00) / low (0x01)

[3] Pulse mode: pass-signal is static (0x00) / pulsed (0x01)

[4] Pulse time format: milliseconds, 8 bits

[5] Minimum length of trigger pulse: microseconds, 8 bits

Test tasks:

Test tasks are activities that are executed in each testing cycle. First task should be ‘Wait for trigger’ and ‘Send results’ the last one.

Test Task	TaskID	Length	Data
Wait for trigger	0x20	0x06	[Source[1]] [0x00] [0x00] [0x00] [0x00] [0x00]
Point test	0x30	0x0B	[Protocol[2]] [Command set[12]] [Transmit power[3]] [Frequency[4]] [Point mode[5]]
UID read test	0x31	0x0C	[Protocol[2]] [Command set[12]] [Transmit power[3]] [Frequency[4]] [Repetitions[6]] [Tolerance[7]] [Word pointer[9,10]] [Word count[9,11]]
Sensitivity test	0x32	0x1A	[Protocol[2]] [Command set[12]] [Frequency[4]] [Range max[3]] [Range min[3]] [Resolution[3]] [Upper control limit[3]] [Lower control limit[3]]
Sweep test	0x33	0x0E	[Protocol[2]] [Command set[12]] [Start Frequency[4]] [Stop Frequency[4]] [Frequency Step[4]]
Send results	0x21	0x01	[TCP disable/enable byte]

[1] Source options: Software (0x00), EXTIO17 rising edge (0x02), EXTIO17 falling edge (0x03).

[2] See HF Supported protocols and commands at page 8 for more information about supported commands.

[3] Transmit power format: (dBm x 1000) + 2^31, 4 bytes, MSB first.

[4] Carried frequency format: Hz, 4 bytes, MSB first.

[5] Point mode: indifferent (0x00), must respond (0x01), must not respond (0x02)

[6] Repetitions: number of times the test is repeated, minimum value: 1

[7] Tolerance: number of repetitions that are allowed to fail

[8] Send results options: Disable (0x00) / Enable (0x01) sending result data to TCP

[9] Optional parameters for ISO18000-3M3 (must be provided together or not at all)

[10] Word pointer: 4 bytes, MSB first, default value: 0x00000000 (if not defined)

[11] Word count: 1 byte, default value: 0x06 (if not defined)

[12] Command set: 1 byte, Protocol standard commands (0x00). See “HF Protocol and Command sets” at page 73 for more information.

Example Transmit:

Length	Command	Number of Tasks	Task 1 ID	Task data length	Task data	Task 2 ID	Task data length	Task data	Task 3 ID	Task data length
18 bytes	0x0010	0x06	Wait for Trigger	0x06	Source + RFU bytes	Point test	0x0B	Data bytes	Send Results	0x00

Cmd: 00 00 00 28 00 10 03 20 06 02 00 00 00 00 00 30 0B 00 00 80 00 23 28 00 CE E8 C0 01 21 00

Load test case of 3 tasks; ‘Wait for Trigger’(source=EXTIO17); ‘Point Test’(15693, 9dBm, 13,56MHz, must respond); ‘Send Results’

TCL (Test Case Loaded)

The Tagsurance HF response to a successful test case initialization initiated by “LTC” from client. Tagsurance HF will first evaluate test case data and verify validity. The response is provided after the case has been loaded successfully. If test definitions are not valid, error frame is sent instead. Error response includes general error byte and separate error indication for each task.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	Test Case Loaded (TCL), high byte
5	Command	0x11	Test Case Loaded (TCL), low byte

Example Transmit:

Length	Command	Data
0x00000002	0x0011	(none)

Cmd: 00 00 00 02 00 11

STC (Start loaded Test Case)

INIT STATE: idle (test case successfully loaded)
NEXT STATE: inline
FUNCTION: start the loaded test case
TAGSURANCE HF RESPONSE: TCS, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	Start loaded Test Case (STC), high byte
5	Command	0x12	Start loaded Test Case (STC), low byte

Example Transmit:

Length	Command	Data
2 bytes	0x0012	(none)

Cmd: 00 00 00 02 00 12

TCS (Test Case Started)

The Tagsurance HF response to a successful test case initialization initiated by “STC” from client. If there is no test case loaded, error frame is sent instead.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	Test Case Started (TCS), high byte
5	Command	0x13	Test Case Started (TCS), low byte

Example Transmit:

Length	Command	Data
2 bytes	0x0013	(none)

Cmd: 00 00 00 02 00 13

STOP (Stop Running Test Case)

INIT STATE: inline
NEXT STATE: idle
FUNCTION: stop the initiated test case
TAGSURANCE HF RESPONSE: STOPPED, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	Stop running Test Case (STOP), high byte
5	Command	0x14	Stop running Test Case (STOP), low byte

Example Transmit:

Length	Command	Data
2 bytes	0x0014	(none)

Cmd: 00 00 00 02 00 14

STOPPED (Test Case Stopped)

The Tagsurance HF response to ‘STOP’ from client. Tagsurance will stop the inline case that has been initiated. Response is provided after the case has been stopped and the system returned to idle state.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	Test Case Stopped (STOPPED), high byte
5	Command	0x15	Test Case Stopped (STOPPED), low byte

Example Transmit:

Length	Command	Data
2 bytes	0x0015	(none)

Cmd: 00 00 00 02 00 15

MCHD (Multi Channel Detection Disable)

INIT STATE: idle
NEXT STATE: idle
FUNCTION: Disable Multi Channel Detection in Multilane
TAGSURANCE HF RESPONSE: MCHR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	Multi Channel Detection Disable, high byte
5	Command	0x21	Multi Channel Detection Disable, low byte

Example Transmit:

Length	Command	Data
2 bytes	0x0021	(none)

Cmd: 00 00 00 02 00 21

MCHE (Multi Channel Detection Enable)

INIT STATE: idle
NEXT STATE: idle
FUNCTION: Enable Multi Channel Detection in Multilane
TAGSURANCE HF RESPONSE: MCHR, or ERR

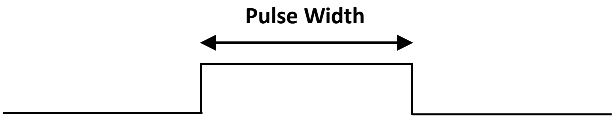
Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	Multi Channel Detection Enable, high byte
5	Command	0x22	Multi Channel Detection Enable, low byte

Example Transmit:

Length	Command	Data
2 bytes	0x0022	(none)

Cmd: 00 00 00 02 00 22

By enabling the channel detection in Multilane system, the HW trigger signal used to enable Tagsurance HF acquisition must respect specific rules (Pulse Width depth and signal shape as shown below) in order to allow Tagsurance HF to detect the correct channel within the Multilane system. The HW trigger source must be selected to the “HW (MASTER), rising” option.



Channel 1 -> 50 us < PW < 100 us
Channel 2 -> 100 us < PW < 150 us
Channel 3 -> 150 us < PW < 200 us
Channel 4 -> 200 us < PW < 250 us
Channel 5 -> 250 us < PW < 300 us
Channel 6 -> 300 us < PW < 350 us

MCHR (Multi Channel Ready)

The Tagsurance HF response to ‘MCHE’ and ‘MCHD’ from client.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	Multi Channel Ready, high byte
5	Command	0x23	Multi Channel Ready, low byte

Example Transmit:

Length	Command	Data
2 bytes	0x0023	(none)

Cmd: 00 00 00 02 00 23

TRIG (TCP trigger to Tagsurance HF Tester Unit)

INIT STATE: inline
NEXT STATE: inline
FUNCTION: trigger a new test measurement specified in the initiated test case
TAGSURANCE HF RESPONSE: TRIGGERED, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)

4	Command	0x00	TCP Trigger to Tagsurance HF Tester Unit (TRIG), high byte
5	Command	0x1A	TCP Trigger to Tagsurance HF Tester Unit (TRIG), low byte

Example Transmit:

Length	Command	Data
2 bytes	0x001A	(none)

Cmd: 00 00 00 02 00 1A

TRIGGERED (TCP Triggering Successful)

The Tagsurance HF response to successful ‘TRIG’ from client. The Tagsurance HF will trigger a new measurement from TCP port and wait for the triggered measurement to be performed. The response is sent after the test has been completed.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	TCP Trigger Successful (TRIGGERED), high byte
5	Command	0x1B	TCP Trigger Successful (TRIGGERED), low byte

Example Transmit:

Length	Command	Data
2 bytes	0x001B	(none)

Cmd: 00 00 00 02 00 1B

POINT (Point Test ISO 15693 only)

INIT STATE: idle
NEXT STATE: no change
FUNCTION: perform a single Point Test (ISO 15693 only)
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x0F	Length byte 4 (LSB)
4	Command	0x00	Point Test (POINT), high byte
5	Command	0x30	Point Test (POINT), low byte
6	Transmit Power 1	0xXX	Transmit Power 1 ((dBm x 1000) + 2^31, MSB)
7	Transmit Power 2	0xXX	Transmit Power 2 ((dBm x 1000) + 2^31)
8	Transmit Power 3	0xXX	Transmit Power 3 ((dBm x 1000) + 2^31)
9	Transmit Power 4	0xXX	Transmit Power 4 ((dBm x 1000) + 2^31, LSB)
10	Frequency 1	0xXX	Frequency 1 (Hz, MSB)
11	Frequency 2	0xXX	Frequency 2 (Hz)
12	Frequency 3	0xXX	Frequency 3 (Hz)
13	Frequency 4	0xXX	Frequency 4 (Hz, LSB)
14	Carrier Before Command 1	0xXX	Carrier Before Command 1 (us, MSB)
15	Carrier Before Command 2	0xXX	Carrier Before Command 2 (us)
16	Carrier Before Command 3	0xXX	Carrier Before Command 3 (us)
17	Carrier Before Command 4	0xXX	Carrier Before Command 4 (us, LSB)
18	Modulation Index	0x0X	Modulation index (10%: 0x00, 100%: 0x01)

Example Transmit:

Length	Command	Data
15 bytes	0x0030	TX Power 10dBm, Frequency: 13.56MHz, Carrier Before Command: 5000us, Mod.Index = 10%

Cmd: 00 00 00 0F 00 30 80 00 27 10 00 CE E8 C0 00 00 13 88 00

9.2.1 Response to “POINT” from client

In case of POINT test, result data includes Pass/fail indicator and error code.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x04	Length byte 4 (LSB)
4	Command	0x00	Test Results (TR), high byte
5	Command	0x1F	Test Results (TR), low byte
6	Pass/fail	0x0X	0x00 (failed) / 0x01 (passed)
7	Error Code	0xXX	0x00 (No Error)

Example Transmit:

Length	Command	Pass/fail	Error Code
4 bytes	0x001F	passed	No Error

Cmd: 00 00 00 04 00 1F 01 00

SWEEP (Simple Sweep)

INIT STATE: idle
NEXT STATE: no change
FUNCTION: perform a single response threshold sweep for a tag
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x12	Length byte 4 (LSB)
4	Command	0x00	Simple sweep test (SWEEP), high byte
5	Command	0x31	Simple sweep test (SWEEP), low byte
6	<RFU>	0x00	0x00
7	<RFU>	0x00	0x00
8	Protocol	0xXX	ISO 15693 (0x00), ISO 14443-A (0x01), ISO 14443-B (0x02), FeliCa (0x03), ISO 18000-3M3 (0x04), TTO (0x05)
9	Command set	0x00	Standard command set (0x00)[1]
10	Start frequency 1	0xXX	Start frequency 1 (Hz, MSB)
11	Start frequency 2	0xXX	Start frequency 2 (Hz)
12	Start frequency 3	0xXX	Start frequency 3 (Hz)
13	Start frequency 4	0xXX	Start frequency 4 (Hz, LSB)
14	Stop frequency 1	0xXX	Stop frequency 1 (Hz, MSB)
15	Stop frequency 2	0xXX	Stop frequency 2 (Hz)
16	Stop frequency 3	0xXX	Stop frequency 3 (Hz)
17	Stop frequency 4	0xXX	Stop frequency 4 (Hz, LSB)
18	Frequency step 1	0xXX	Frequency step 1 (Hz, MSB)
19	Frequency step 2	0xXX	Frequency step 2 (Hz)
20	Frequency step 3	0xXX	Frequency step 3 (Hz)
21	Frequency step 4	0xXX	Frequency step 4 (Hz, LSB)

[1] See “HF Protocol and Command sets” at page 73 for more information.

Example Transmit:

Length	Command	<RFU>	<RFU>	Protocol	<RFU>	Data
18 bytes	0x0031	0x00	0x00	ISO 14443-A	0x00	13MHz, Stop: 14MHz, Step: 0,1MHz

Cmd: 00 00 00 12 00 31 00 00 01 00 00 C6 5D 40 00 D5 9F 80 00 01 86 A0

9.2.2 Response to “SWEEP” from client

In case of SWEEP, result data includes threshold values at test frequencies

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0xXX	Length byte 1 (MSB)
1	Length byte 2	0xXX	Length byte 2
2	Length byte 3	0xXX	Length byte 3
3	Length byte 4	0xXX	Length byte 4 (LSB)
4	Command	0x00	Test Results (TR), high byte
5	Command	0x1F	Test Results (TR), low byte
6	Pass/fail	0x0X	0x00 (failed) / 0x01 (passed)
7	TaskID	0x33	Sweep test ID (in test case)
8	Result length byte 1	0xXX	Result length byte 1 (MSB)
9	Result length byte 2	0xXX	Result length byte 2 (LSB)
10	Task pass/fail	0x01	0x00 (failed) / 0x01 (passed)
11…	Threshold values	0xXX….	[Threshold power[1] @freq1]…[[Threshold power @freqN]

[1] Threshold power format: (dBm x 1000) + 2^31, 32-bits, MSB first

Example Transmit:

Length	Command	Pass/fail	Data
27 bytes	0x001F	passed	SWEEP: passed, 5 frequency points, Thresholds: 0, 1, 2, 3, 4 dBm

Cmd: 00 00 00 1B 00 1F 01 33 00 15 01 80 00 00 00 80 00 00 01 80 00 00 02 80 00 00 03 80 00 00 04

9.2.1 UID Read (Simple Read Tag's ID)

INIT STATE: idle
NEXT STATE: no change
FUNCTION: Read tag's ID
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x0D	Length byte 4 (LSB)
4	Command	0x00	Simple UID read test (UIDRead), high byte
5	Command	0x33	Simple UID read test (UIDRead), low byte
6	<RFU>	0x00	0x00
7	<RFU>	0x00	0x00
8	Protocol	0xXX	ISO 15693 (0x00), ISO 14443-A (0x01), ISO 14443-B (0x02) [2] , ISO 18000-3M3 (0x04) , TTO (0x05)
9	Command set	0xXX	Standard command set (0x00) [3]
10	Transmit Power 1	0xXX	Transmit Power 1 ((dBm x 1000) + 2^31, MSB)
11	Transmit Power 2	0xXX	Transmit Power 2 ((dBm x 1000) + 2^31)
12	Transmit Power 3	0xXX	Transmit Power 3 ((dBm x 1000) + 2^31)
13	Transmit Power 4	0xXX	Transmit Power 4 ((dBm x 1000) + 2^31, LSB)
14	Frequency 1	0xXX	Start frequency 1 (Hz, MSB)
15	Frequency 2	0xXX	Start frequency 2 (Hz)
16	Frequency 3	0xXX	Start frequency 3 (Hz)
17	Frequency 4	0xXX	Start frequency 4 (Hz, LSB)
18	Word pointer 1	0xXX	Word pointer 1 (MSB)[1]
19	Word pointer 2	0xXX	Word pointer 2[1]
20	Word pointer 3	0xXX	Word pointer 3[1]
21	Word pointer 4	0xXX	Word pointer 4 (LSB) [1]
22	Word Count	0xXX	Word count to be read[1]

[1] Required parameter, but only affects with ISO 180000-3M3

[2] Only supported in combination with Command set 0x01 (ST25TB)

[3] See "HF Protocol and Command sets" at page 73 for more information. Note: Not all Protocol and Command set combinations are available for UID Read.

Example Transmit:

Length	Command	<RFU>	<RFU>	Protocol	Command set	Data
18 bytes	0x0033	0x00	0x00	ISO 14443-A	Standard	TX power: 10dBm, Frequency: 13,56MHz, Word pointer: 0, Word Count: 0

Cmd: 00 00 00 12 00 33 00 00 01 00 80 00 27 10 00 CE E8 C0 00 00 00 00 00

9.2.1 Response to “UIDREAD” from client

In case of successful UIDREAD, result data includes UID number of the tag tested according to the definition in the table below.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0xXX	Length byte 1 (MSB)
1	Length byte 2	0xXX	Length byte 2
2	Length byte 3	0xXX	Length byte 3
3	Length byte 4	0xXX	Length byte 4 (LSB)
4	Command	0x00	Test Results (TR), high byte
5	Command	0x1F	Test Results (TR), low byte
6	Pass/fail	0x0X	0x00 (failed) / 0x01 (passed)
7	TaskID	0x31	UID Read test ID (in test case)
8	Result length byte 1	0xXX	Result length byte 1 (MSB)
9	Result length byte 2	0xXX	Result length byte 2 (LSB)
10	Task pass/fail	0x01	0x00 (failed) / 0x01 (passed)
11	Error code	0xXX…	Task error code (0x00 = no error)
12..	Data	0xXX…	[Tag’s ID[1]]

[1] Tag’s ID returned by a successful Simple UIDRead command:

ISO 15693	UID
ISO 14443A	UID
ISO 14443B (ST25TB)	UID
ISO 18000-3M3	TID data depending on word pointer and word count defined in the command
TTO PR1101	UID
TTO PR1102	UID
TTO NFC Barcode	Barcode

Example Transmit:

Length	Command	Pass/fail	Data
12 bytes	0x001F	passed	UIDRead: passed, no error, UID number: 0x01020304

Cmd: 00 00 00 0C 00 1F 01 31 00 06 01 00 01 02 03 04

CARRIER (Carrier ON/OFF)

INIT STATE: idle
NEXT STATE: no change
FUNCTION: Turn the Carrier On/Off
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x0B	Length byte 4 (LSB)
4	Command	0x00	Carrier On/Off (CARRIER), high byte
5	Command	0x4A	Carrier On/Off (CARRIER), low byte
6	Transmit Power 1	0xXX	Transmit Power 1 ((dBm x 1000) + 2^31, MSB)
7	Transmit Power 2	0xXX	Transmit Power 2 ((dBm x 1000) + 2^31)
8	Transmit Power 3	0xXX	Transmit Power 3 ((dBm x 1000) + 2^31)
9	Transmit Power 4	0xXX	Transmit Power 4 ((dBm x 1000) + 2^31, LSB)
10	Frequency 1	0xXX	Frequency 1 (Hz, MSB)
11	Frequency 2	0xXX	Frequency 2 (Hz)
12	Frequency 3	0xXX	Frequency 3 (Hz)
13	Frequency 4	0xXX	Frequency 4 (Hz, LSB)
14	Carrier On/Off	0x0X	Carrier Off: 0x00, Carrier On: 0x01

Example Transmit:

Length	Command	Data
11 bytes	0x004A	TX Power 10dBm, Frequency: 13.56MHz, Carrier ON

Cmd: 00 00 00 0B 00 4A 80 00 27 10 00 CE E8 C0 01

9.2.1 Response to "CARRIER" from client

In case of CARRIER command, result data includes an error code.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x03	Length byte 4 (LSB)
4	Command	0x00	Test Results (TR), high byte
5	Command	0x1F	Test Results (TR), low byte
7	Error Code	0xXX	0x00 (No Error)

Example Transmit:

Length	Command	Error Code
3 bytes	0x001F	No Error

Cmd: 00 00 00 03 00 1F 00

TR (Test Result)

TR header is common response but result data format depends on the calling command.

Performance test results frame format sent at the end of case (”Send Test Results” test task)

Unless TCP communication is disabled, Tagsurance HF will send test results to TCP when performing “Send Test Results” test task. That test is typically the last test in a test case and the frame data section thus includes test results.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0xXX	Length byte 1 (MSB)
1	Length byte 2	0xXX	Length byte 2
2	Length byte 3	0xXX	Length byte 3
3	Length byte 4	0xXX	Length byte 4 (LSB)
4	Command	0x00	Test Results (TR), high byte
5	Command	0x1F	Test Results (TR), low byte
6	Pass/fail	0x0X	0x00 (failed) / 0x01 (passed)
7…	Result data	…	See table below

Test result data format:

Test	TaskID	Length	Data
Point test	0x30	0x0001	[passed/failed[1]]
UID read test	0x31	0xXXXX	[passed/failed[1]] [errorCode[2]] [ID[3], byte0]….[ID, byteN]
Sensitivity test	0x32	0x0002	[error code[4]] [threshold power[5]]
Frequency sweep test	0x33	0xXXXX	[passed/failed[1]] [threshold$_{fstart}$[5]]…. [threshold$_{fstop}$[5]]

[1] Passed/failed byte: failed (0x00), passed (0x01).

[2] Error code: 0x00 = no error.

[3] ID length depends on the chip memory type, protocol, and command parameters (ISO 18000-3M3).

[4] 0x00 out of limits, 0x01 within control limits, 0x02 sensitivity equals lower control limit.

[5] mdBm +2^{31}, 32-bits, MSB first

Example Transmit:

Length	Command	Pass/fail	Data
7 bytes	0x001F	passed	tests: 1 point test (passed)

Cmd: 00 00 00 07 00 1F 01 30 00 01 01

TCP Test

INIT STATE: handshake (within 10s from opening TCP connection)
NEXT STATE: no change
FUNCTION: perform handshake with the Tagsurance GUI
TAGSURANCE RESPONSE: TCP Ready, or ERR

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02/0x04	Length byte 4 (LSB)
4	Command	0x00	TCP Connection Test (TCP Test), high byte
5	Command	0xF0	TCP Connection Test (TCP Test), low byte
6	Optional byte 1	0xXX	Heartbeat interval[1] (ms, MSB)
7	Optional byte 2	0xXX	Heartbeat interval[1] (ms, LSB)

[1] Heartbeat interval is optional. If parameter is defined as 0x0000, or left undefined, heartbeat is disabled.

Example Transmit:

Length	Command	Data
4 bytes	0x00F0	Disable heartbeat

Cmd: 00 00 00 04 00 F0 00 00

TCP Ready

Tagsurance answer to ‘TCP Test’ from client.

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x02	Length byte 4 (LSB)
4	Command	0x00	TCP Connection Test (TCP Test), high byte
5	Command	0xF1	TCP Connection Test (TCP Test), low byte

Example Transmit:

Length	Command	Data
2 bytes	0x00F1	(none)

Cmd: 00 00 00 04 00 F0 00 00

ERR (Error from Tagsurance HF)

Tagsurance HF can send ERR frame as a response to an invalid command.
Data format depends on the command causing the error response

Common use

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0x03	Length byte 4 (LSB)
4	Command	0x00	Error (ERR), high byte
5	Command	0xFF	Error (ERR), low byte
6	Generic Error Code	0xXX	Error code (see table below)

Generic Error codes:

Code	Command	Description
0x00	No error	No error
0x01	Invalid command	The command provided was not valid
0xFF	Unspecified Error	Unidentified error

Example Transmit:

Length	Command	Data
3 bytes	0x00FF	invalid command

Cmd: 00 00 00 03 00 FF 01

Invalid inline case loading attempt

Byte No.	Function	Value (HEX)	Description
0	Length byte 1	0x00	Length byte 1 (MSB)
1	Length byte 2	0x00	Length byte 2
2	Length byte 3	0x00	Length byte 3
3	Length byte 4	0xXX	Length byte 4 (LSB)
4	Command	0x00	Error (ERR), high byte
5	Command	0xFF	Error (ERR), low byte
6	Generic Case Error	0xXX	Generic Case Error code (see table below)
7…	Task error codes	…	Error code for each task included into load command (same order)

Generic case error code:

Bit	Command	Description
0	Task definition error	Task definition error(s) exists
1	Trigger task missing	No "Wait for trigger" task is defined in test case
2	<RFU>	
3	<RFU>	
4	<RFU>	
5	<RFU>	
6	<RFU>	
7	<RFU>	

Task error code:

Bit	Command	Description
0	Invalid data length	The number of parameter bytes don't match with the specification.
1	Invalid parameter	Parameter has an invalid value.
2	Invalid power	The power exceeds the valid range, or the start and stop values mismatch.
3	Invalid frequency	The frequency exceeds the valid range, or the start and stop values mismatch.
4	<RFU>	
5	<RFU>	
6	Invalid task ID	
7	License error	The license does not cover the asked function, protocol, or frequency.

Example Transmit:

Length	Command	Data
7 bytes	0x00FF	4 tasks, 3rd is invalid with respect to frequency

Cmd: 00 00 00 07 00 FF 01 00 00 08 00

HF Protocol and Command sets

	Command Set 0x00	Command Set 0x01	Command Set 0x02	Command Set 0x03
Protocol 0x00	ISO 15693	-	-	-
Protocol 0x01	ISO 14443-A	-	-	-
Protocol 0x02	ISO 14443-B	ST25TB	-	-
Protocol 0x03	FeliCa	-	-	-
Protocol 0x04	ISO 18000-3M3	-	-	-
Protocol 0x05	-	PR1101	PR1102	NFC Barcode

This table is a summary of all possible combinations of Protocols and Command Sets. Note that all combinations are not available for all test types. See table HF Supported protocols and commands at page 8.

“-“ in the table means that there’s no defined combination and it is reserved for future use.

10 Operation Using the UHF Serial Command Interface

10.1 Communication Protocol

When the Tagsurance UHF Tester is turned on, it first performs initialization and then moves to the IDLE state. The process takes less than one second and the device is then ready to accept commands from the serial port. IDLE state is indicated by setting the ready LED in the device from panel on and the BUSY pin in the external connector low. When interfacing the device directly from the serial port, the READY pin can be used to detect when the initialization is complete.

The serial command interface provides commands that can be used to perform measurements (inline testing, response threshold measurements, and self-checking), or to modify system settings. The different functions can be initiated when the device is in the IDLE mode, and it is also where it returns when the requested function is successfully performed.

RS-232 serial port settings:
Baud rate: 38400kbps, Data: 8 bits, no parity, LSB first, Interpretation: ASCII format

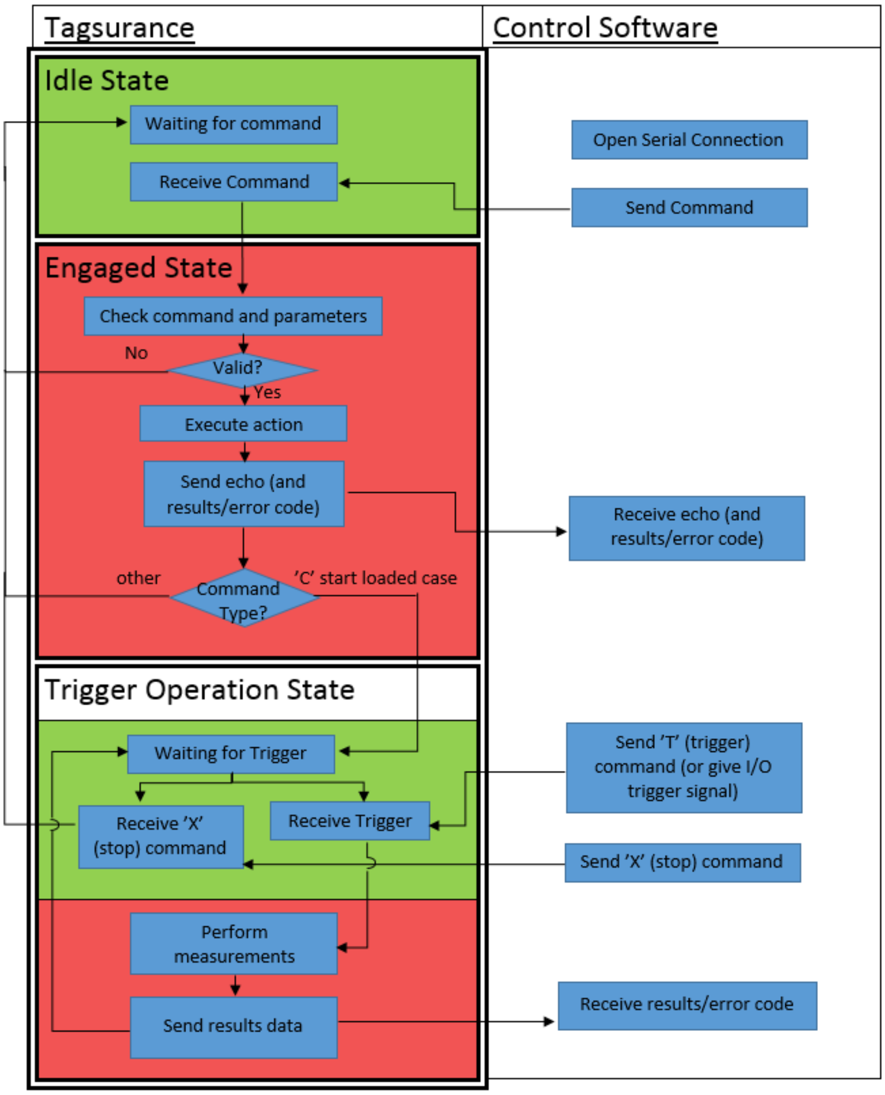


Figure 38: Basic operation cycle of the Voyantic Tagsurance UHF Tester

10.2 System Configuration Commands

System settings can be modified through the serial interface.

SYSTEM CONFIGURATION			
Function	**Command**	**Response**	**Next state**
Enable external trigger (default)	'PTE'	(none)	IDLE
Disable external trigger	'PTI'	(none)	IDLE
Set trigger (ext) to rising edge (default)	'PTH'	(none)	IDLE
Set trigger (ext) to falling edge	'PTL'	(none)	IDLE
Enable internal gen2 chip	'POC'	(none)	IDLE
Disable internal gen2 chip (default)	'POA'	(none)	IDLE
Enable Multi Channel Detection* **	'PQE'	(none)	IDLE
Disable Multi Channel Detection	'PQD'	(none)	IDLE
Set carrier time (default: 480=[1][224]=2,5ms)	'PC' [scaler+time_high_byte] [time_low_byte] [1]	(none)	IDLE
Set pass(/fail) signal to static mode (default)	'PIPS'	(none)	IDLE
Set pass(/fail) signal to pulsed mode	'PIPP'	(none)	IDLE
Enable "failed = HIGH" option	'PIPM'	(none)	IDLE
Disable "failed = HIGH" option (default)	'PIPN'	(none)	IDLE
Save settings	'PFS'	(none)	IDLE
Load saved settings	'PFL'	[fbyte$_L$][fbyte$_H$] [time$_H$][time$_L$]	IDLE
Reset factory default settings	'PFR'	(none)	IDLE

[1] [SSTT TTTT] [TTTT TTTT] i.e. S=scaler (2bits), T=time (14 bits)

INTERPRETATION AND DEFINITION OF CARRIER TIME

calculation formula

Carrier_time[ms] = (time/160 – 0,5ms) * scaler

Where:

- scaler (2 bits): 1x (00_2), 10x (01_2), 100x (10_2), 1000x (11_2)
- time (14 bits): (carrier_time[ms]/scaler + 0,5ms) * 160

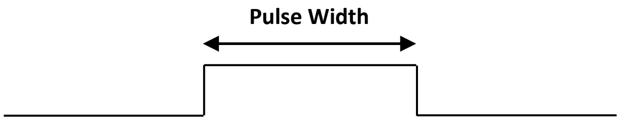
valid ranges

Time: 112 – 16080 = 0,2ms – 100ms

Scaler: 00_2, 01_2, 10_2, 11_2 = 1x, 10x, 100x, 1000x

INTERPRETATION OF THE FBYTE DATA		
bit	**FBYTE$_L$**	**FBYTE$_H$**
0	trigger mode (0->ext, 1->int, default: ext)	(RFU)
1	trigger edge (0->falling, 1->rising, default: rising)	(RFU)
2	pass signal mode (0->static, 1->pulsed, default: static)	(RFU)
3	failed = HIGH option (0->disabled, 1->enabled, default: disabled)	(RFU)
4	(RFU)	(RFU)
5	(RFU)	(RFU)
6	(RFU)	(RFU)
7	(RFU)	(RFU)

***By enabling the channel detection in Multilane system, the HW trigger signal used to enable Tagsurance UHF acquisition must respect specific rules (Pulse Width depth and signal shape as shown below) in order to allow Tagsurance UHF to detect the correct channel within the Multilane system. The HW trigger source must be selected to “HW (MASTER), rising” option.



Channel 1 -> 50 us < PW < 100 us
Channel 2 -> 100 us < PW < 150 us
Channel 3 -> 150 us < PW < 200 us
Channel 4 -> 200 us < PW < 250 us
Channel 5 -> 250 us < PW < 300 us
Channel 6 -> 300 us < PW < 350 us

10.3 *Inline Measurement Commands*

The Tagsurance Tester can be instructed to conduct a series of tests (i.e. inline case) for a tag. Tests may include point tests (i.e. response tests in specific frequency-power points), and/or read tests (i.e. reading the tag memories). Inline case data are uploaded through the serial interface and ran by using specific serial commands. Alternatively, they can also be stored and applied from the device memory (up to 5 cases).

INLINE MEASUREMENT COMMANDS			
Function	**Command**	**Response**	**Next state**
Upload case data		error code[4]	IDLE
- header	['L'][Data Length H][Data Length L]		
- data for point tests	['P'][Tolerance][Npoints][Mode + f_H][f_L][Power]		
- data for read tasks	['R'][Bank + f_H][f_L][Power][Word Pointer] [Word Count][Repetitions + Tolerance]		
- data for write tasks	['W'][6] [Bank + f_H][f_L][Power][Word Pointer] [Repetitions + Tolerance][Increment] [Word Count][Data]		
- data for sweep task	['S'][Start f_H][Start f_L] [Stop f_H][Stop f_L] [Step f_H][Step f_L]		
- data for sensitivity test	['C'][f_H][f_L][P_L][P_H][P_{LCL}][P_{UCL}][Uncertainty]		
- data for fast EPC[7]	['M'][wCount]	[Err][Data][8]	IDLE
Start (loaded) case	'C'	error code[4]	WAIT FOR TRIG[2]
Trigger	'T'	pass/fail &data[5]	WAIT FOR TRIG
Stop (running) case[1]	'X'	(none)	IDLE
Save (active) case	['F'][case number (0-4)]	error code[4]	IDLE
Load saved case	['O'][case number (0-4)]	error code[4]	IDLE
Get active case data	'PD'	case data[3]	IDLE

[1] Any char other than 'T' also stops the running case and causes the device to return to IDLE state
However, for the sake of simplicity, 'X' is recommended

[2] If no valid case is loaded, the device will return error and return to IDLE state

[3] If no valid case is loaded, the device will return {0x00, 0x00} (i.e. number of data bytes is zero)

[4] For description of the error codes, see Section 10.7

[5] For description of the data, see section 10.3.2

[6] Use 'V' for MandatoryWrite

[7] Only for ISO 18000-6C

[8] err (0x00 = no error) followed by the data read (repeated string of 0x00's if read was failed)

10.3.1 Uploading Inline Case Data

Test case data can be uploaded by using a specific command sequence which consists of a header and a data sequence. Header initiates the activity in the IDLE state and defines the amount of data included in the case data. Data sequence, on the other hand, defines the tests to be made when a case is performed. Possible measurements are point-vise (frequency, power) response tests and reading of tag memories.

While designing a case, the user must take care of the case data size. Maximum length of the data sequence is 510 bytes, and the maximum length of the result data vector is 100 bytes.

Case Data Header

['L'][Data Length H][Data Length L]

Where:

['L'] is the control char that initiates the activity from the IDLE state
[Data length] (16bits) indicates the number of bytes included in the data sequence that follows

Definition of the Point Tests

['P'][Tolerance][Npoints][Mode$_1$ + f$_{H1}$][f$_{L1}$][Power$_1$]…[Mode$_N$ + f$_{HN}$][f$_{LN}$][Power$_N$]

Where:

['P'] is the identifier of the point test definitions
[Tolerance] indicates the amount of point tests that may fail while the tag still passes the test
[Npoints] is the amount of test points
[Mode] (2 MSB bits) defines if the tag: must respond (01), must not respond (10), response is indifferent (00)
[f] (14 bits) defines the carrier frequency presented as [MHz]x10)
[Power] defines the carrier power (presented as: [dBm]x4+128)

Definition of the Read Tests

['R'][Bank + f$_H$][f$_L$][Power][Word Pointer][Word Count][Repetitions + Tolerance]

Where:

['R'] is the identifier of the read test definitions
[Bank] (memory bank, 2 MSB bits) is the identifier of the tag's memory bank (00, 01, 10, or 11)
[f] (14 bits) defines the carrier frequency (presented as [MHz]x10)
[Power] defines the carrier power (presented as [dBm]x4+128)
[Word pointer] identifies the start address for the read command (0-128, EBV format)
[Word count] defines the number of words to be read from the tag
[Repetitions] number of repetitions (4 MSB bits) is the number of times the read-test is performed
[Tolerance] is the number of times the read test may fail while it still is considered to be successful

Definition of the Write Tests

[‘W’][Bank + f_H][f_L][Power][Word Pointer][Repetitions + Tolerance][Increment][Word Count][Data]

Where:

[‘W’] is the identifier of the write test definitions
[Bank] (memory bank, 2 MSB bits) is the identifier of the tag’s memory bank (00, 01, 10, or 11)
[f] (14 bits) defines the carrier frequency (presented as [MHz]x10)
[Power] defines the carrier power (presented as [dBm]x4+128)
[Word pointer] identifies the start address for the read command (0-128, EBV format)
[Repetitions] number of repetitions (4 MSB bits) is the number of times the read-test is performed
[Tolerance] is the number of times the read test may fail while it still is considered to be successful
[Increment] defines the increment for the data per cycle (0 for static data, >0 for incremental data)
[Word Count] defines the number of words to be written to the memory bank (max. word count is 8)
[Data] is the data that is written to the tag (number of bytes = 2 x word count)

Definition of the Sweep Test

[‘S’][Start f_H][Start f_L][Stop f_H][Stop f_L][Step f_H][Step f_L]

Where:

[Start f] defines the start frequency
[Stop f] defines the stop frequency
[Step f] defines the frequency step size

Definition of the Sensitivity Test

[‘C’][f_H][f_L][P_L][P_H]P_{LCL}][P_{UCL}][Uncertainty]

Where:

[f] defines the frequency
[P_L] defines the lowest tested power
[P_H] defines the highest tested power
[P_{LCL}] defines the lower control limit
[P_{UCL}] defines the upper control limit
[Uncertainty] defines the uncertainty criterion (dB x 4)

Definition of fast EPC

Command: ['M'][wCount]

Response: [err][Data.....]

Where:

Err: 0x00 no error, 0x01 not read, 0x03 interpret error (CRC), 0x04 EPC overflow error, 0x05 EPC underflow error. Please refer to Figure 39 for an example. If wCount is greater than the tag's EPC codelength, the entire EPC is returned padded with 0s for the missing number of words.

The rest retrieves the EPC code from the inventory sequence. Due to this, the function requires a 'Read' or 'Write' test to be included in the loaded test case in which fast EPC is used. In addition, fast EPC must be performed after the 'Read' or 'Write'.

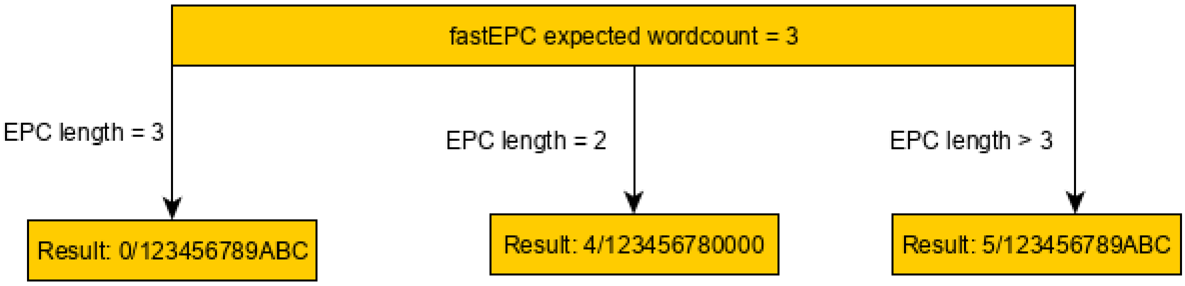


Figure 39 fast EPC error codes explained

10.3.2 Running an Inline Case

Starting the Measurement

The test case can be started by sending a ‘Start case’ command from the serial port (‘C’). Having received the command, the device checks that a valid case has been loaded and goes to the WAITING FOR TRIGGER state. If a valid case is not available, the device returns to the IDLE state. The action taken can be interpreted from the error flag sent as a response to the command.

In some case, it may be useful to download the active case data back from the device. This can be done by using command: ‘PD’, which returns the case length and data, if one has been loaded.

Triggering

In the WAIT FOR TRIGGER state a measurement can be initiated by providing a trigger from the serial port by sending: ‘T’, or by applying trigger signal to the external trigger pin. In response, the device performs the measurement, sends the results to the serial port, and returns to the WAIT FOR TRIGGER state.

Interpretation of Results

The results are provided to the serial port in the following format:

Byte 1:	Pass/fail byte, 0x00 (failed), 0x01 (passed)
Byte 2…N:	Test details

- Point Test results are provided as single bits starting from the first empty byte in the result vector. LOW represents a failed test, HIGH represents a passed test.
- Read Test results are provided with an error byte (see table below) followed by the data read. If the task was failed, the data is replaced with 0x00’s.
- Write Test results are provided with an error byte (see table below).
- Sweep Test results are provided as bytes representing the power thresholds at tested frequencies. Powers are indicated as: ([dBM] x 4) + 128.
- Sensitivity Test results are provided with and error byte indicating that LCL/UCL criterion was passed (0x00), failed (0x01) or in a case where the LCL/UCL criterion is not defined inside the measurement range and the result is out of measurement range (0x02). Next byte is the sensitivity threshold measured. Powers are indicated as: ([dBM] x 4) + 128.

INTERPRETATION OF READ/WRITE TASK ERROR BYTE			
0x00	0_{10}	No errors	Tolerance condition was met
0x01	1_{10}	Connection failed	Tag did not respond to query
0x02	2_{10}	Command failed	Tag did not respond to the command
0x03	3_{10}	Inventory failed	Tag responds to query, but inventory failed
0x04	4_{10}	Tag error message 1	Other error (not covered by the other tag error codes)
0x34	52_{10}	Tag error message 2	Memory overrun (too high memory address)
0x44	68_{10}	Tag error message 3	Memory locked (memory cannot be read/written)
0xB4	180_{10}	Tag error message 4	Insufficient power (carrier level too low)
0xF4	244_{10}	Tag error message 5	Non-specified (error-specific codes not supported by the tag)

INTERPRETATION OF SENSITIVITY TEST TASK ERROR BYTE			
0x00	0_{10}	No errors	Tolerance condition was met
0x01	1_{10}	Out of tolerances	Tolerance condition was not met, the threshold power exceeded the upper control limit or was below the lower control limit
0x02	2_{10}	Out of range	UCL and LCL are not defined inside the measurement range and the response power is out of measurement range.

The WAIT FOR TRIGGER mode is indicated by the ready led in the device front panel being lit and the ready/busy pin in the external connector low. During the case execution, the BUSY pin is high and the BUSY led in the device front-panel lit. Furthermore, the outcome of the test (PASS/FAIL) is indicated by the PASS and FAIL LEDs in the device front panel, and the PASS/FAIL pin in the external connector.

Stopping the Measurement

To stop the case and to return to the IDLE state, the user must send 'X' to the serial port. This will cause the device to stop performing the case and to go to the IDLE state. Also any other character will have the same effect, but for the sake of simplicity, 'X' is recommended.

10.3.3 Test Case Storage and Handling

After a case has been successfully uploaded, it can be stored in the device memory. The device will respond to the command by returning an error flag, which indicates if the task could be successfully performed. It then returns to the IDLE state. Load function works the same way. Furthermore, once loaded, the active case data can be downloaded through the serial port by using command: 'PD'.

The following commands can be used to perform these actions:

['F'][Memory Location]

save active case to a memory location (0x0 to 0x4)

['O'][Memory Location]

load case from a memory location (0x0 to 0x4)

['PD']

Transmit active case data to the serial port

To avoid the need to build code for uploading the inline test cases, they can be loaded and restored by using the application software. This allows simplification of the control code.

10.4 Other Measurement Commands

In addition to inline cases, the device can perform separate tests that can be executed in the IDLE state by using specific commands. These tests include: response threshold test and system self-test.

OTHER MEASUREMENT FUNCTIONS			
Function	**Command**	**Response**	**Next state**
Threshold sweep	[‘S’][Start f_H][Start f_L][Stop f_H][Stop f_L][Step f_H][Step f_L]	[Err][N Bytes][Data][1]	IDLE
Read	[‘R’][Bank + f_H][f_L][Power][Word Pointer] [Word Count][Repetitions + Tolerance]	[Err][ErrByte][Data][2]	IDLE
BlockWrite	[‘W’][Bank + f_H][f_L][Power][Word Pointer][Word Count][Data]	[Err][ErrByte][3]	IDLE

[1] Byte count and data will follow if the command is valid (user must check the error flag, 0x00=no error)
[2] errByte (0x00 = no error) followed by the data read (repeated string of 0x00’s if read was failed)
[3] errByte (0x00 = no error)

10.4.1 Threshold Sweep

The Threshold Sweep test performs a frequency sweep and measures the power threshold needed to wake up the tag through a certain frequency band. The sweep start frequency, stop frequency and the size of the frequency step are defined by the user. The device responds to the command by sending an error byte. The byte count and measurement data will follow if the command is valid.

Command: [‘S’][Start f_H][Start f_L][Stop f_H][Stop f_L][Step f_H][Step f_L]

Response: [err][N bytes][Threshold Power$_1$][Threshold Power$_2$]… [Threshold Power$_N$]

Where

[Err]	identifies if the command was valid (0x00=no error)
[ErrByte]	indicates the error while performing the task
[N bytes]	byte count of the following data
[Threshold Power$_N$]	the threshold power at N^{th} measurement frequency

Frequencies are given in format: [MHz]x10
Powers are given in format: [dBm]x4+128

10.4.2 Read

The Read Test performs a single read operation for tag.

Command: [‘R’][Bank + f_H][f_L][Power][Word Pointer][Word Count][Repetitions + Tolerance]

Response: [Err][ErrByte][Data Byte$_1$][Data Byte$_2$]…[Data Byte$_N$]

Where

[Err]	identifies if the command was valid (0x00=no error)
[ErrByte]	indicates the error while performing the task
[Data Byte]	the data read from the tag is divided to N data bytes

10.4.3 Block Write

The Block Write performs a single write operation for tag using blockWrite command.

Command: [‘W’][Bank + f_H][f_L][Power][Word Pointer][Word Count][Data]

Response: [err][errByte]

Where

[Err] identifies if the command was valid (0x00=no error)
[ErrByte] indicates the error while performing the task

10.5 Tag Encoding, Locking, and Killing Commands

Tagsurance 2 provides custom encoding session commands. Principles on how to make use of these commands and their syntax are described in the following chapters.

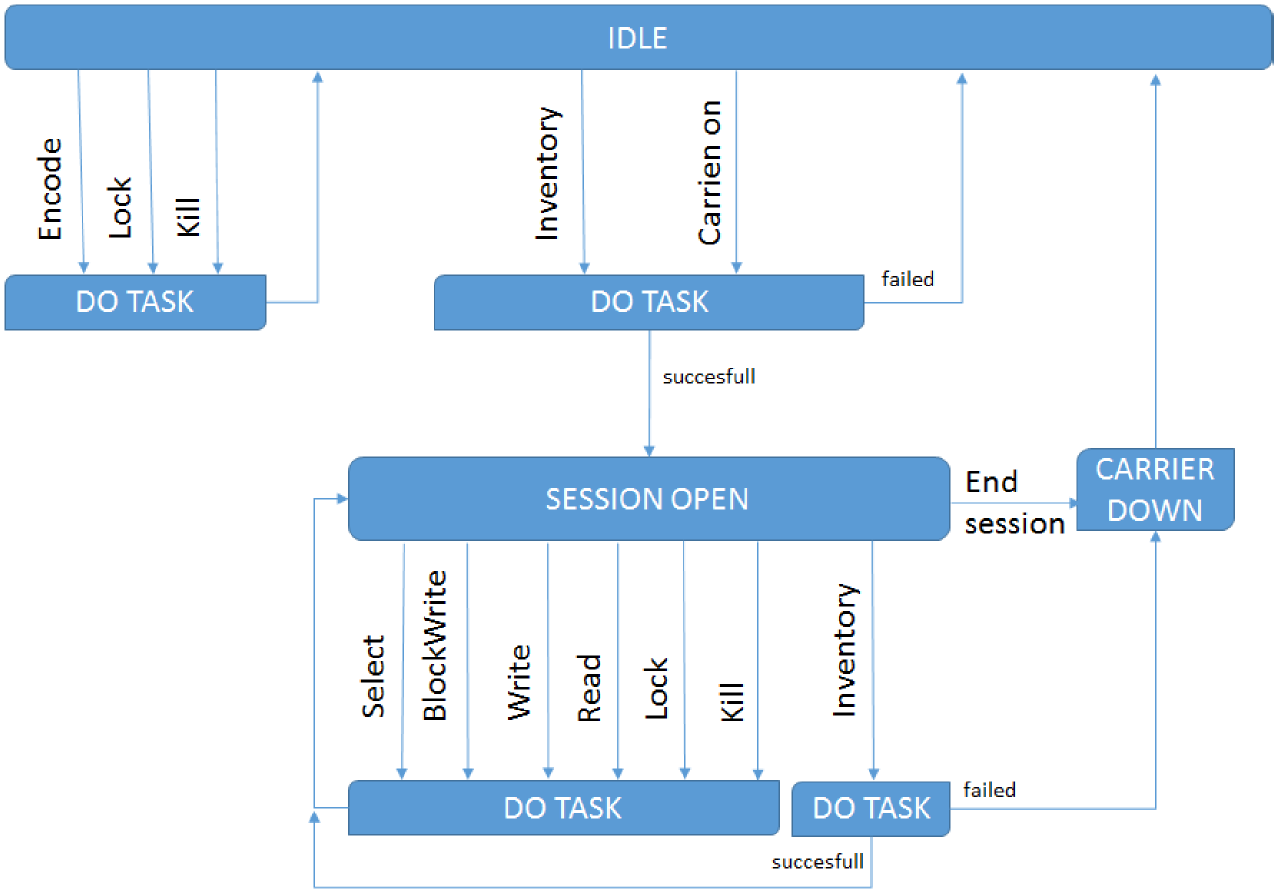


Figure 40: Operation logic and state diagram of Tagsurance encoding commands

Stand-alone commands are operation macros which will first switch carrier on and perform inventory, then perform the specific encoding task(s), and finally end the session by switching carrier off and returning the Tagsurance to the idle state. The supported stand-alone functions are: encode, lock, and kill. “Encode” can be used to program and lock EPC memory, user memory, access password, and kill password. “Lock” can be used to set tag lock bits. “Kill” is used to perform tag killing procedure with optional kill password encoding.

STANDALONE ENCODING COMMANDS

Function	Command	Response	Next state
Encode	[‘H’][Data Length H][Data Length L] [f$_H$][f$_L$][Power][Acc-pwd$_{1,MSB}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$] [EPC Word count][EPC Word pointer][EPC Data] [User Word count][User Word pointer][User Data] [Enab kill-pwd prog][Kill-pwd$_{1,MSB}$][Kill-pwd$_2$][Kill-pwd$_3$][Kill-pwd$_4$] [Enab acc-pwd prog][Acc-pwd$_{1,MSB}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$] [Enab lock prog][Payload$_{1,MSB}$][Payload$_2$][Payload$_3$][5]	[Err][1][Task][ErrByte][2-4]	IDLE
Lock	[‘D’][f$_H$][f$_L$][Power][Repetitions + Tolerance] [Payload$_{1,MSB}$][Payload$_2$][Payload$_3$]	[Err][1][ErrByte][2,4]	IDLE
Kill	[‘K’][f$_H$][f$_L$][Power] [Acc-pwd$_{1,MSB}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$] [Kill-pwd$_{1,MSB}$][Kill-pwd$_2$][Kill-pwd$_3$][Kill-pwd$_4$]	[Err][1][Task][ErrByte][2-4]	IDLE

[1] Err represents system error code (0x00=no error). For description of the codes, see Section 10.7: “Error Handling”

[2] ErrByte (and task) will follow Err if the command is valid (Err=0x00)

[3] Task indicates the last task performed before end session (See task indices table below)

[4] ErrByte represents error code from the last task (0x00 = no error)

[5] Payload byte syntax.

INTERPRETATION OF TASK IDENTIFIER BYTE

0x00	**0_{10}**	Inventory (and access) sequence
0x01	**1_{10}**	EPC memory programming
0x02	**2_{10}**	User memory programming
0x03	**3_{10}**	Kill password programming
0x04	**4_{10}**	Access password programming
0x05	**5_{10}**	Lock bit programming
0x06	**6_{10}**	Kill procedure

Custom encoding session commands can be used to perform a user-defined encoding session. The session is initiated by sending “Inventory” command. Upon successful inventory, the device will go to encoding mode where the user is allowed to perform other custom encoding commands (“blockWrite”, “write”, “read”, “lock”, “kill”). The session is finished by issuing “End Session” command, which will switch carrier off and return the device to the idle state.

CUSTOM ENCODING SESSION INITIALIZATION COMMANDS

Function	Command	Response	Next state
Inventory	[‘I’] [f$_H$][f$_L$][Power][Acc-pwd$_{1,MSB}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$]	[Err][1][ErrByte][2,4]	IDLE/ENCODING
Carrier on	[‘J’] [f$_H$][f$_L$][Power]	[Err][1]	IDLE/ENCODING

[1] Err represents system error code (0x00=no error). For description of the codes, see Section 10.7: “Error Handling”

[2] ErrByte (and task) will follow Err if the command is valid (Err=0x00)

[3] Task indicates the last task performed before end session (See task indices table below)

[4] ErrByte represents error code from the last task (0x00 = no error)

[5] Payload byte syntax

CUSTOM ENCODING SESSION COMMANDS

Function	Command	Response	Next state
Select	[‘E’][‘S’] [bit count[6]][byte count[7]] [cmd bits$_1$]… [cmd bits$_N$]	[Err][1][ErrByte][2,4]	ENCODING
Inventory (insession)	[‘E’][‘I’] [queryPar][Acc-pwd$_{1,MSB}$][Acc-pwd$_2$] [Acc-pwd$_3$][Acc-pwd$_4$]	[Err][1][ErrByte][2,4]	IDLE/ENCODING
BlockWrite	[‘E’][‘W’][Bank][Word pointer][Word count][Data]	[Err][1][ErrByte][2,4]	ENCODING
Mandatory Write	[‘E’][‘V’][Bank][Word pointer][Word count][Data]	[Err][1][ErrByte][2,4]	ENCODING
Read	[‘E’][‘R’][Bank][Word pointer][Word count]	[Err][1][ErrByte][2,4]	ENCODING
Lock	[‘E’][‘L’][Payload$_{1,MSB}$][Payload$_2$][Payload$_3$]	[Err][1][ErrByte][2,4]	ENCODING
Kill	[‘E’][‘K’] [Kill-pwd$_{1,MSB}$][Kill-pwd$_2$][Kill-pwd$_3$][Kill-pwd$_4$]	[Err][1][ErrByte][2,4]	ENCODING
End session	[‘X’]	-	IDLE

[1] Err represents system error code (0x00=no error). For description of the codes, see Section 10.7: “Error Handling”

[2] ErrByte (and task) will follow Err if the command is valid (Err=0x00)

[3] Task indicates the last task performed before end session (See task indices table below)

[4] ErrByte represents error code from the last task (0x00 = no error)

[5] Payload byte syntax

[6] Maximum bit count is 171

[7] Byte count in select command represents the number of command bytes

10.5.1 Stand-alone command set: Encode

The encode command performs programming and locking of EPC memory, user memory, access password, and kill password according to user provided parameters.

Command:

['H'] Data Length H][Data Length L]…

…[fH][fL][Power][Acc-pwd1,MSB][Init acc-pwd$_{1,\mathrm{MSB}}$][Init acc-pwd$_2$][Init acc-pwd$_3$][Init acc-pwd$_4$]…

…[EPC Word count][EPC Word pointer][EPC Data]…

…[User Word count][User Word pointer][User Data]…

…[Enable kill password programming][Kill-pwd$_{1,\mathrm{MSB}}$][Kill-pwd$_2$][Kill-pwd$_3$][Kill-pwd$_4$]…

…[Enable access password programming][Acc-pwd$_{1,\mathrm{MSB}}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$]…

…[Enable lockbits programming][Payload$_{1,\mathrm{MSB}}$][Payload$_2$][Payload$_3$]

Where

['H'] is the identifier of the encode parameters
[Data length] (16bits) is the number of bytes included in the data sequence that follows
[f] (14 LSB bits) defines the carrier frequency (presented as [MHz]x10)
[Power] defines the carrier power (presented as [dBm]x4+128)
[Init acc-pwd$_{1\text{-}4}$] are access password bytes used to perform access after inventory (MSB first)
[Word Count] defines the number of words to be written to the memory bank
[Word Pointer] identifies the start address for the write command (0-128, EBV format)
[Data] is the data that is written to the tag (number of bytes = 2 x word count)
[Enable … programming] enables/disables the programming tasks (0x00=disabled, 0x01=enabled)
[Acc-pwd$_{1\text{-}4}$] are access password bytes (MSB first) to be programmed
[Kill-pwd$_{1\text{-}4}$] are kill password bytes (MSB first) to be programmed
[Payload$_{1\text{-}3}$] are lock command payload bits (20 bits, MSB first) + 4 dummy bits (LSB)

Notes:

- inventory is made with access if initial access password is non-zero
- the amount of bytes after data length has to match with the data length
- the amount of bytes sent has to match with the word count (i.e. data and word pointer not sent, if word count=0)
- password bytes are provided only if password programming is enabled (Enable … programming = 0x01)
- lock bits are provided only if lockbits programming is enabled (enable lockbits programming = 0x01)

Response:

[Err][Task][errByte]

Where

[Err] identifies if the command was valid (0x00=no error)
[Task] indicates the last task performed
[ErrByte] indicates the error while performing the task (0x00=no error)

10.5.2 Stand-alone command set: Lock

Lock performs locking procedure for a tag at predefined frequency and power.

Command:

[‘D’][fH][fL][Power][Repetitions + Tolerance][Payload$_{1,MSB}$][Payload$_2$][Payload$_3$]

Where

[‘D’]	is the identifier of the lock parameters
[f]	(14 LSB bits) defines the carrier frequency (presented as [MHz]x10)
[Power]	defines the carrier power (presented as [dBm]x4+128)
[Repetitions]	(4 MSB bits) is the maximum number of repetitions the task is performed
[Tolerance]	(4 LSB bits) is the maximum number of repetitions allowed to fail
[Payload$_{1-3}$]	are lock command payload bits (20 bits, MSB first) + 4 dummy bits (LSB) (see table below)

Response:

[Err][errByte]

Where

[Err]	identifies if the command was valid (0x00=no error)
[ErrByte]	indicates the error while performing the task (0x00=no error)

	KILL PASSWORD		ACCESS PASSWORD		EPC MEMORY		TID MEMORY		USER MEMORY	
	23 (MSB)	22	21	20	19	18	17	16	15	14
MASK BITS	0:skip 1:write	0:skip 1:write	0:skip 1:write	0:skip 1:write	0:skip 1:write	0:skip 1:write	0:skip 1:write	0:skip 1:write	0:skip 1:write	0:skip 1:write
	13	12	11	10	9	8	7	6	5	4
ACTION BITS	pwd write/ read	perma lock	pwd write/ read	perma lock	pwd write	perma lock	pwd write	perma lock	pwd write	perma lock
	3	2	1	0						
DUMMY BITS	-	-	-	-						

10.5.3 Stand-alone command set: Kill

Kill performs killing procedure for a tag at predefined frequency and power.

Command:

[‘K’][fH][fL][Power]...
…[Acc-pwd$_{1,MSB}$][Acc-pwd$_{2}$][Acc-pwd$_{3}$][Acc-pwd$_{4}$]…
…[Kill-pwd$_{1,MSB}$][Kill-pwd$_{2}$][Kill-pwd$_{3}$][Kill-pwd$_{4}$]

Where

[‘K’] is the identifier of the kill parameters
[f] (14 LSB bits) defines the carrier frequency (presented as [MHz]x10)
[Power] defines the carrier power (presented as [dBm]x4+128)
[Acc-pwd$_{1\text{-}4}$] are access password bytes (MSB first)
[Kill-pwd$_{1\text{-}4}$] are kill password bytes (MSB first)

Response:

[Err][Task][errByte]

Where

[Err] identifies if the command was valid (0x00=no error)
[Task] indicates the last task performed
[ErrByte] indicates the error while performing the task (0x00=no error)

Notes:

- A tag will not accept kill command if kill password is zero. If the kill password provided with the command is zero, the Tagsurance will try to program kill password before attempting to kill the tag.
- If kill password is locked, reprogramming can only be performed if the tag is in secured state. If the access password provided with the command is non-zero, Tagsurance will perform inventory with access.

10.5.4 Custom Command Set: Inventory with Optional Access

Inventory with optional access command turns carrier on and performs inventory with the tag. Then, it continues with the optional access procedure, if a non-zero access password is provided with the command. If successful, leaves device in encoding mode.

Command:

[‘I’] [fH][fL][Power][Acc-pwd$_{1,MSB}$][Acc-pwd$_{2}$][Acc-pwd$_{3}$][Acc-pwd$_{4}$]

Where

[‘I’] is the identifier of the command
[f] (14 bits) defines the carrier frequency (presented as [MHz]x10)
[Power] defines the carrier power (presented as [dBm]x4+128)
[Acc-pwd$_{1-4}$] are access password bytes used to perform access after inventory (MSB first)

Response:

[Err][errByte]

Where

[Err] identifies if the command was valid (0x00=no error).
[ErrByte] indicates the error while performing the task (0x00=no error)

10.5.5 Custom Command Set: Carrier on

Switches carrier on and, if successful, leaves the device to encoding mode.

Command:

[‘J’] [fH][fL][Power]

Where

[‘J’] is the identifier of the command
[f] (14 bits) defines the carrier frequency (presented as [MHz]x10)
[Power] defines the carrier power (presented as [dBm]x4+128)

Response:

[Err]

Where

[Err] identifies if the command was valid (0x00=no error).

10.5.6 Custom Command Set: Select

Select performs select with user-defined parameters. The user defines number of command bits in and the number of bytes carrying that information. All the select command bits have to be provided with the command, but CRC-16 is automatically handled by Tagsurance.

Command:

[‘E’][‘S’] [bit count][byte count][cmd bits1]… [cmd bitsN]

Where

[‘E’]	is the identifier of the encode command set
[‘S’]	is the identifier of the select command
[bit count]	is the number of command bits to be sent (maximum value: 171)
[byte count]	is the number of command bytes which carry the command bits (i.e. N)
[cmd bits]	are select command bits divided to bytes with the last byte filled to the end (LSB with 0’s)

Response:

[Err][errByte]

Where

[Err]	identifies if the command was valid (0x00=no error).
[ErrByte]	indicates the error while performing the task (0x00=no error)

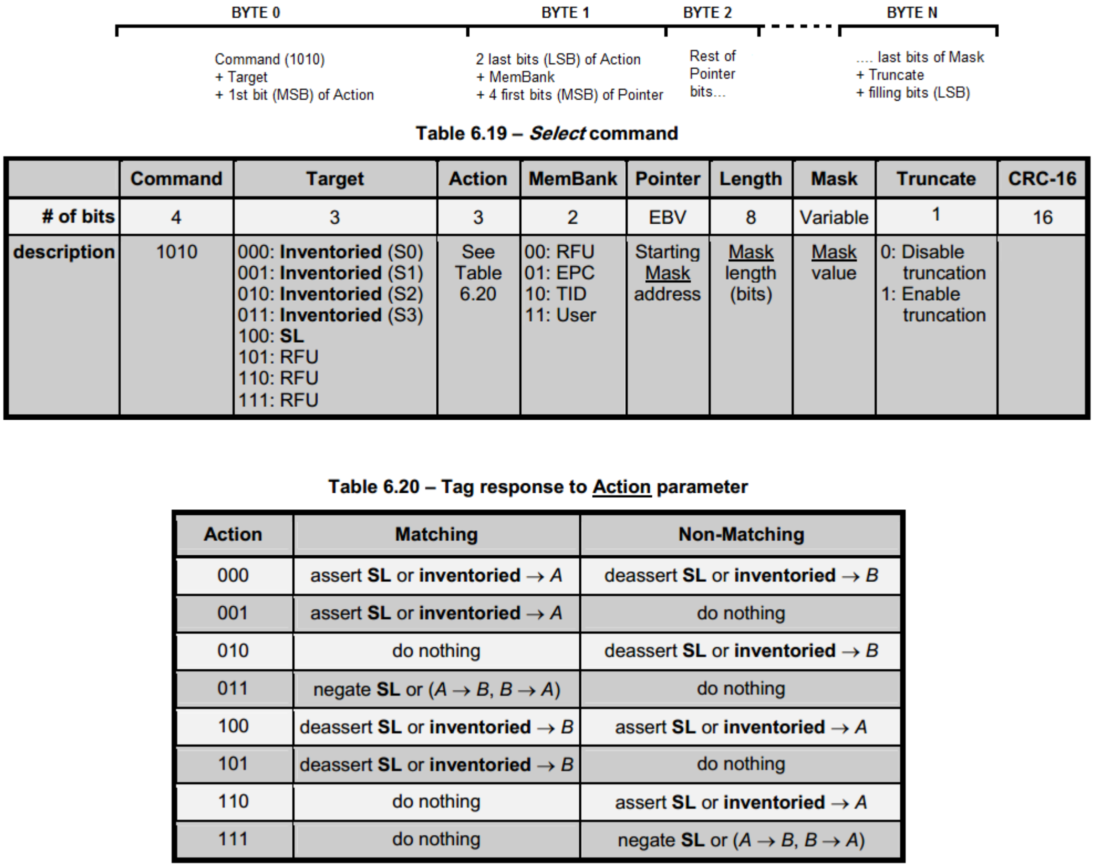


Table 6.19 – *Select* command

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC-16
# of bits	4	3	3	2	EBV	8	Variable	1	16
description	1010	000: **Inventoried** (S0) 001: **Inventoried** (S1) 010: **Inventoried** (S2) 011: **Inventoried** (S3) 100: **SL** 101: RFU 110: RFU 111: RFU	See Table 6.20	00: RFU 01: EPC 10: TID 11: User	Starting Mask address	Mask length (bits)	Mask value	0: Disable truncation 1: Enable truncation	

Table 6.20 – Tag response to Action parameter

Action	Matching	Non-Matching
000	assert **SL** or **inventoried** → *A*	deassert **SL** or **inventoried** → *B*
001	assert **SL** or **inventoried** → *A*	do nothing
010	do nothing	deassert **SL** or **inventoried** → *B*
011	negate **SL** or (*A* → *B*, *B* → *A*)	do nothing
100	deassert **SL** or **inventoried** → *B*	assert **SL** or **inventoried** → *A*
101	deassert **SL** or **inventoried** → *B*	do nothing
110	do nothing	assert **SL** or **inventoried** → *A*
111	do nothing	negate **SL** or (*A* → *B*, *B* → *A*)

Figure 41: Select command parameters and definition of the command bytes for Tagsurance

10.5.7 Custom Command Set: Insession Inventory with Optional Access

Insession inventory with optional access performs inventory with the tag with user-defined query parameters. Then, it continues with the optional access procedure, if a non-zero access password is provided with the command. Only available in encoding mode.

Command:

[‘E’][‘I’][QueryPar][Acc-pwd$_{1,\mathrm{MSB}}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$]

Where

[‘E’] is the identifier of the encode command set
[‘I’] is the identifier of the insession inventory command
[QueryPar] represents SEL (SS, msb), SESSION (ss), TARGET (T), and three dummy bits (000, lsb)
[Acc-pwd$_{1\text{-}4}$] are access password bytes used to perform access after inventory (MSB first)

Response:

[Err][errByte]

Where

[Err] identifies if the command was valid (0x00=no error).
[ErrByte] indicates the error while performing the task (0x00=no error)

10.5.8 Custom Command Set: Block Write

Block Write Performs a single write operation for tag using blockWrite command. It can be performed after a successful inventory.

Command:

[‘E’][‘W’][Bank][Word Pointer][Word Count][Data]

Where

[‘E’] is the identifier of the encode command set
[‘W’] is the identifier of the blockWrite command
[Bank] (memory bank, 2 MSB bits) is the identifier of the tags memory bank (00, 01, 10, or 11)
[Word Pointer] identifies the start address for the write command (0-128, EBV format)
[Word Count] defines the number of words to be written to the memory bank
[Data] is the data that is written to the tag (number of bytes = 2 x word count)

Response:

[Err][errByte]

Where

[Err] identifies if the command was valid (0x00=no error)
[ErrByte] indicates the error while performing the task (0x00=no error)

10.5.9 Custom Command Set: Mandatory Write

Write Performs a single write operation for tag using mandatory write command. It can be performed after a successful inventory.

Command:

[‘E’][‘V’][Bank][Word Pointer][Word Count][Data]

Where

[‘E’] is the identifier of the encode command set
[‘V’] is the identifier of the write command
[Bank] (memory bank, 2 MSB bits) is the identifier of the tags memory bank (00, 01, 10, or 11)
[Word Pointer] identifies the start address for the write command (0-128, EBV format)
[Word Count] defines the number of words to be written to the memory bank
[Data] is the data that is written to the tag (number of bytes = 2 x word count)

Response:

[Err][errByte]

Where

[Err] identifies if the command was valid (0x00=no error)
[ErrByte] indicates the error while performing the task (0x00=no error)

10.5.10 Custom Command Set: Read

Read performs a single read operation for a tag. It can be performed after a successful inventory.

Command:

[‘E’][‘R’][Bank][Word Pointer][Word Count]

Where

[‘E’] is the identifier of the encode command set
[‘R’] is the identifier of the read command
[Bank] (memory bank, 2 MSB bits) is the identifier of the tags memory bank (00, 01, 10, or 11)
[Word Pointer] identifies the start address for the read command (0-128, EBV format)
[Word Count] defines the number of words to be written to the memory bank

Response:

[Err][errByte]

Where

[Err] identifies if the command was valid (0x00=no error)
[ErrByte] indicates the error while performing the task (0x00=no error)

10.5.11 Custom Command Set: Lock

Lock performs a locking procedure for a tag. It can be performed after a successful inventory.

Command:

[‘E’][‘L’][Payload$_{1,\text{MSB}}$][Payload$_2$][Payload$_3$]

Where

[‘E’] is the identifier of the encode command set
[‘L’] is the identifier of the lock command
[Payload$_{1\text{-}3}$] are lock payload bits (20 bits, MSB first) + 4 dummy bits (LSB) (see table below)

Response:

[Err][errByte]

Where

[Err] identifies if the command was valid (0x00=no error)
[ErrByte] indicates the error while performing the task (0x00=no error)

10.5.12 Custom Command Set: Kill

Kill performs a killing procedure for a tag. It can be performed after a successful inventory.

Command:

[‘E’][‘K’] [Kill-pwd$_{1,\text{MSB}}$][Kill-pwd$_2$][Kill-pwd$_3$][Kill-pwd$_4$]

Where

[‘E’] is the identifier of the encode command set
[‘K’] is the identifier of the kill command
[Kill-pwd$_{1\text{-}4}$] are kill password bytes (MSB first)

Response:

[Err][errByte]

Where

[Err] identifies if the command was valid (0x00=no error)
[ErrByte] indicates the error while performing the task (0x00=no error)

10.5.13 Custom Command Set: End Session

End session turns carrier down to power-off the tag

Command:

[‘X’]

Response:

None

10.6 Combining Encoding with Performance Testing

In order to allow encoding while running a performance test case, the user must activate "encode after performance testing" option. This is done by using a setup command while the device is in IDLE state. With this option activated, the Tagsurance will automatically go to IDLE state after completing performance test case. In this state, encoding can be performed by using the encode commands. After completing the encoding procedure, performance testing is reinitiated from serial port.

In normal testing mode, the Tagsurance front-panel led lights and back panel signals are updated after the performance testing is done. With "encode after performance testing" option activated, the indicators are updated with the reinitialization of the test case. Pass/fail signals are affected by the encode functions performed and the test result is failed if any of the encoding activities (encode, kill, encode command set functions) fail.

Once activated ('PME'), the option will stay on until it is deactivated from the serial port ('PMD'), or the device power is recycled.

COMBINING ENCODING WTIH PERFORMANCE TESTING			
Function	**Command**	**Response**	**Next state**
Activate encode after performance test option	'PME'	(none)	IDLE
Deactivate encode after performance test option	'PMD'	(none)	IDLE
Reinitialize inline case (after encoding)	'C'	Err[1]	WAIT FOR TRIG

[1] Err represents system error code (0x00=no error). For description of the codes, see Section 10.7: "Error Handling"

10.7 Error Handling

If errors occur during the data transmission or if invalid commands are given, it will be identified by the error code returned. No error is indicated 0x00. Any other value indicates that an error occurred.

INTERPRETATION OF SYSTEM ERROR CODE(S) (0 -> no error, 1 --> error)		
Bit 0	invalid input data sequence	valid range: 1 to 510 bytes
Bit 1	timeout during data reception	maximum delay between data bytes: 1000ms
Bit 2	invalid memory bank	valid range: 0 to 4
Bit 3	invalid word count	valid range: >0 (read test), 1-8 (write test)
Bit 4	invalid command characters	valid values: 'P' , 'R', 'W', and 'S'
Bit 5	output data size required too large	valid range: 0 to 100 bytes
Bit 6	invalid frequency	valid range: 860 to 960MHz (extended: 800 to 1100MHz)
Bit 7	Invalid power	valid range: -10 to +25dBm
0xFF	license error	requested task not covered by current license options

Appendix A
TCP Remote Access Interface for Tagsurance 2

A.1 Overview

The TCP/IP interface allows access and control to the Tagsurance test system remotely over a network connection. It can, for example, be used to provide control of measurement events to the main computer unit while the computer with the Tagsurance 2 application is kept close to the production line (Figure 42). This allows efficient use of the graphical indicators in the Tagsurance 2 Operator Interface while the data is collected and stored elsewhere.

In this mode, Tagsurance 2 will act as a TCP server, and the client can control the measurement activities by using a specific command set. Overtaking the control requires a simple handshaking procedure.

Setting up the remote access interface and establishing a connection from client software is described in chapter A.2, and the remote access commands in chapter A.3.

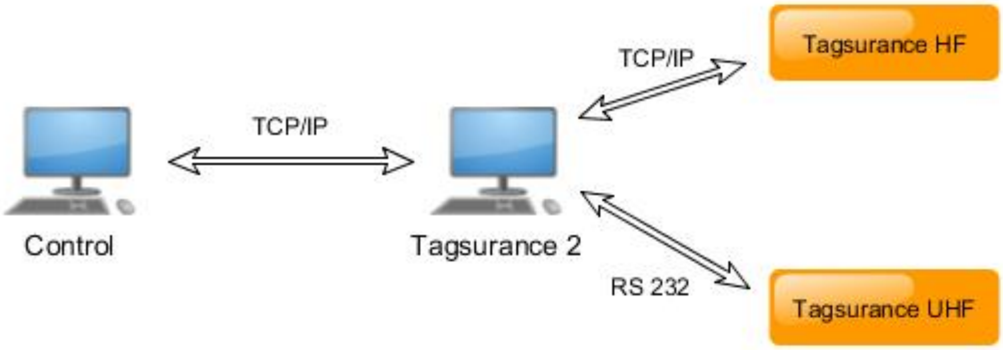


Figure 42: Example of a TCP/IP controlled setup

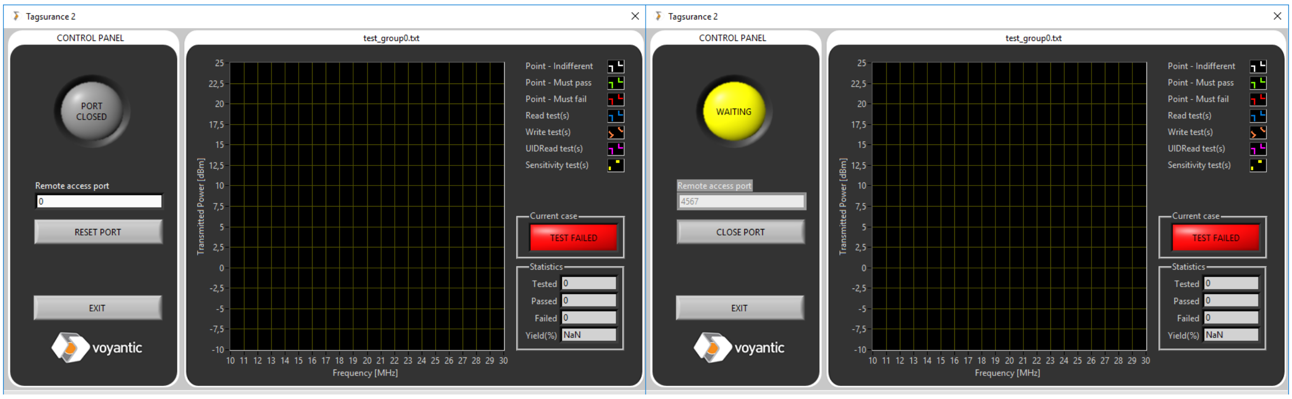
A.2 Remote Access Connection Establishment

To enable access to the Tagsurance 2 remote access interface, the user must follow these steps:

1. Run Tagsurance 2 Launcher.exe, choose remote access interface from the menu and define the network port that will be used for the remote access.

   When ran in remote access mode, the normal operator functions are disabled, and the user is only allowed to define the remote access port number or to close the program. To change port number, the user must first close the port, enter a new value to ‘Remote access port’, and finally re-enable remote access by clicking ‘RESET PORT’.

   When the remote access interface runs for the first time, the port data has not been saved anywhere and the user needs to enter the port number and click ‘RESET PORT’ to enable remote access. This value will be stored to ‘Devices.txt’ in the data folder and retrieved automatically when the remote access interface is opened again. The value can be changed manually from the file and it will also be updated if user changes it again from the Operator Interface.



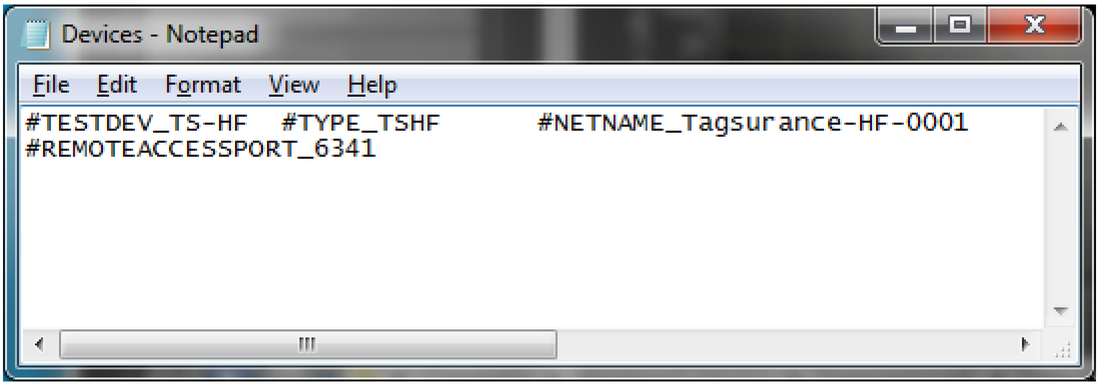
Remote access port 0

Tested 0
Passed 0
Failed 0
Yield(%) NaN

Remote access port 4567

Tested 0
Passed 0
Failed 0
Yield(%) NaN

Figure 43: Tagsurance 2 in remote access mode when port is predefined (right), and when it is not (left). Indicator led in the left-upper corner indicates remote access connection state (gray=OFF, yellow=WAITING, green=CONNECTED)



```
#TESTDEV_TS-HF   #TYPE_TSHF        #NETNAME_Tagsurance-HF-0001
#REMOTEACCESSPORT_6341
```

Figure 44: Remote access connection port is stored in ‘Devices.txt’ with keyword #REMOTEACCESSPORT_<port_number>. This value can be modified either from the GUI or manually from the text file. If the value is defined, the remote access interface will automatically start listening to that port once the program is running. If not, the remote access interface opens in closed mode and the user needs to enter port number and reset port to enable client connection

2. Establish connection

Once a valid port number is entered and the port reset, Tagsurance 2 Operator Interface will start polling for connection attempts at the defined port. This is indicated by the indicator LED turning to blinking yellow and the text 'WAITING'.

To establish connection:

- Client opens TCP connection (computer IP address, and port number needed)
- Tagsurance 2 recognizes connection attempt and waits for 'TCP Test' (timeout: 10s)
- Client transmits 'TCP Test' (=F0 00 00) within 10s from the connection attempt

For a valid command, the Tagsurance 2 GUI responds with 'TCP ready' (=F1 00 00)

→ Tagsurance 2 remote access interface is activated and ready to use

For an invalid command, the Tagsurance 2 GUI responds with 'ERR' (= FF 01 00 xx)

→ Connection attempt is terminated, and remote access interface is not activated

Else (i.e. if timeout occurs)

→ Connection attempt is terminated, and remote access interface is not activated

3. After the handshake procedure has been successfully completed, the remote access interface is ready for use. This is indicated by the indicator LED turning to GREEN and the text 'CONNECTED'.

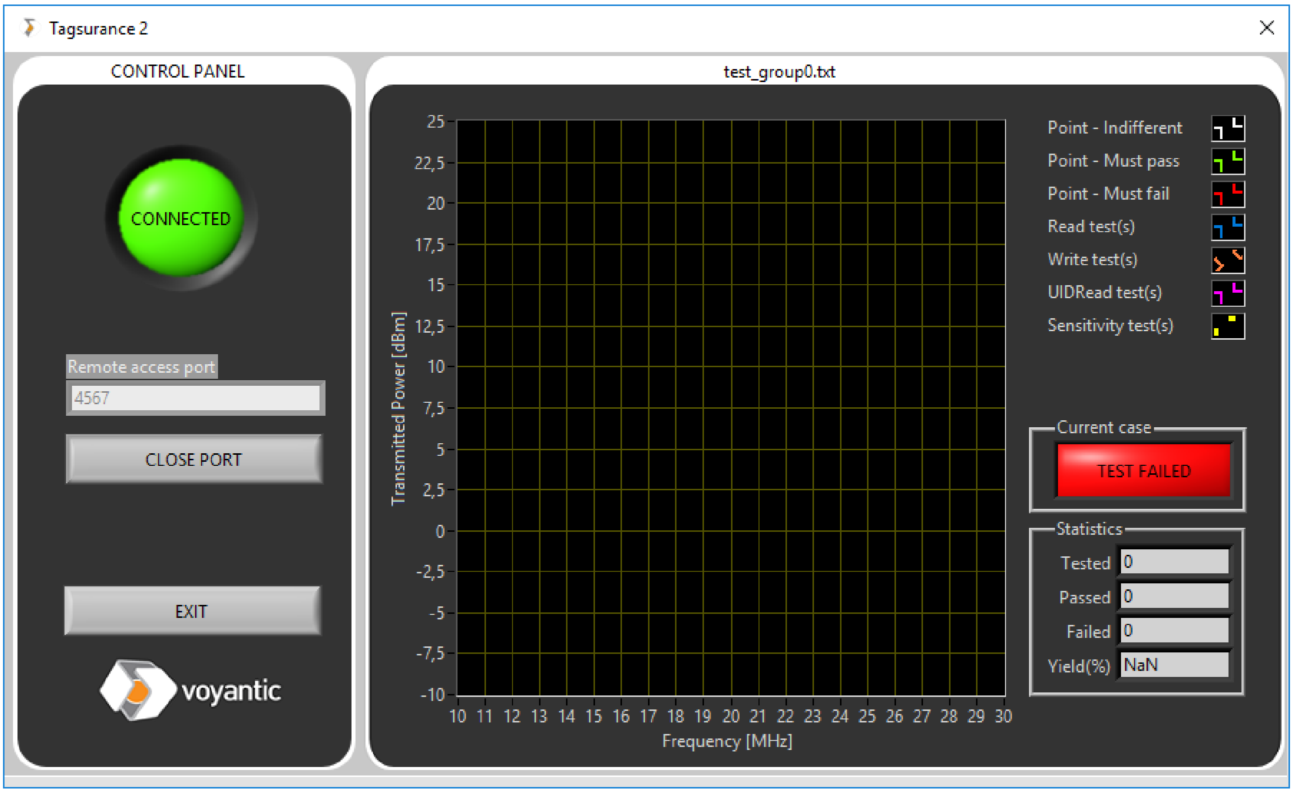


Figure 45: Remote access connection has been established successfully and the interface is ready to use

A.3 Commands Description

Control over the network is made by sending specific commands that initiate certain activities.

A command consists of three parts: command byte, two length bytes, and the possible command parameters (see table below). The length bytes indicate the length of the last part of the command (i.e. the amount of parameter bytes). The data is in hexadecimal format and numbers are sent low byte first.

Command	Length Low Byte (Nlow)	Length High Byte Nhigh	Parameters (N bytes of data)
0x00	0x00	0x00	...

List of available commands is given below. Detailed description is provided in the next chapters.

CODE	COMMAND	DESCRIPTION	INIT STATE	NEXT STATE
0x01	Connect	Connect to Tagsurance Tester Unit	idle/inline	no change
0x02	Connection State	State of Connection to Tagsurance Tester Unit	-	-
0x03	GCL	Get Case List	idle/inline	no change
0x04	CL	Case List (semicolon “;” separated)	-	-
0x05	LSC	Load and Start Specific Case	idle/inline	inline
0x06	SCL	Specific Case Loaded and Started	-	-
0x07	TRIG	Serial trigger to Tagsurance Tester Unit	inline	inline
0x08	READY	Serial triggering successful (and device is ready)	-	-
0x09	STOP	Stop running case	inline	idle
0x0A	STOPPED	Case stopped	-	-
0x10	GTR	Get Test Result	idle/inline *	no change
0x11	TR	Test Results	-	-
0x12	GBS	Get Buffer Status	idle/inline *	no change
0x13	BS	Buffer State	-	-
0x30	READ***	Read data from tag memory	idle/inline	no change
0x31	WRITE***	Write data to tag memory	idle/inline	no change
0x32	SWEEP	Measure tag’s response threshold curve	idle/inline	no change
0x33	ENCODE***	Program and lock tag memories	idle/inline	no change
0x34	KILL***	Kill tag	idle/inline	no change
0x35	UID Read****	Read Tag’s ID	idle/inline	no change
0xF0	TCP Test	TCP Connection Test	handshake	idle**
0xF1	TCP Ready	State of TCP connection	-	-
0xFF	ERR	Error from Tagsurance	-	-

*Buffer state is initialized and the contents flushed when a new case is started.

** This command is used during handshake procedure. Command has no effect in other states (idle, inline).

*** Compatible with UHF devices only

**** Compatible with HF devices only

A.3.1 Connect

INIT STATE: idle, inline
NEXT STATE: no change
FUNCTION: test the state of connection between the Tagsurance 2 GUI and the tester units
TAGSURANCE RESPONSE: Connection State, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x01	Connect
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

Connect	Length Low Byte	Length High Byte
0x01	0x00	0x00

Cmd: 01 00 00

A.3.2 Connection State

The Tagsurance response to a successful connection test initiated by “Connect” from client.

The Tagsurance 2 GUI will test the connection with the tester units and respond with connection status and device name.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x02	Connection State
1	Length Low	0x03	Length Low
2	Length High	0x00	Length High
3	Count of Devices	0x00 … 0xFF	Number of available tester units
4	String 1st Device status	…	Status/device name (0->no connection, 1->ok)
…	Semicolon for Separation	0x3B (;)	Semicolon to separate different names for case
…	String 2nd… Device status	…	Status/device name (0->no connection, 1->ok)
…	Semicolon for Separation	0x3B (;)	Semicolon to separate different names for case
…	String Last Device status	…	Status/device name (0->no connection, 1->ok)

Example Transmit:

Con State	Length Low Byte	Length High Byte	Device Count	1st Device Status “1/TS-UHF”	Separator	2nd Device Status “1/TS-HF”
0x02	0x11	0x00	0x02	31 2F 54 53 2D 55 48 46	0x3B	31 2F 54 53 2D 48 46

Cmd: 02 11 00 02 31 2F 54 53 2D 55 48 46 3B 31 2F 54 53 2D 48 46

A.3.3 GCL (Get Case List)

INIT STATE: idle, inline
NEXT STATE: no change
FUNCTION: ask for a list of available test cases in the data folder
TAGSURANCE RESPONSE: CL, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x03	Get Case List (GCL)
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

GCL	Length Low Byte	Length High Byte
0x03	0x00	0x00

Cmd: 03 00 00

A.3.4 CL (Case List)

The Tagsurance response to "GCL" from client.

The Tagsurance 2 GUI will search for valid test files from the data folder ("...\Tagsurance 2\Data\Test cases\") and return a list of them separated by ';'. Included in the list are only the cases that can be run in the current system. To be included, the file names should include keyword: "Test". Validation is licensing sensitive.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x04	Case List (CL)
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High
3	Count of Case	0x00 … 0xFF	Number of available cases
4	String First Case Name	…	First case name in hex
…	Semicolon for Separation	0x3B (;)	Semicolon to separate different names for case
…	Case Name	…	… in hex
…	Semicolon for Separation	0x3B (;)	Semicolon to separate different names for case
…	String Last Case Name	…	Name for last case in hex

Example Transmit:

CL	Length Low Byte	Length High Byte	Count of Case	Case 1	Separator	Case2	Separator	Case 3
0x04	0x12	0x00	0x03	"Test1"	0x3B	"Test2"	0x3B	"Test3"

Cmd: 04 12 00 03 54 65 73 74 31 3B 54 65 73 74 32 3B 54 65 73 74 33

A.3.5 LSC (Load Specific Case)

INIT STATE: idle, inline
NEXT STATE: inline
FUNCTION: initialize and start a new inline case specified in the indicated case file
TAGSURANCE RESPONSE: SCL, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x05	Load and Start Specific Case (LSC)
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High
3	Case Name	…	Name of the Case

Example Transmit:

LSC	Length Low Byte	Length High Byte	Case Name
0x05	0x05	0x00	"Test1"

Cmd: 05 05 00 54 65 73 74 31

A.3.6 SCL (Specific Case Loaded and Started)

The Tagsurance response to a successful test case initialization initiated by "LSC" from client.

The Tagsurance will first search for the file specified from the data folder, validate the contents, and then start it. The response is provided after the case has been loaded and started successfully.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x06	Specific Case Loaded and Started (SCL)
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High
3	Case Name	…	Name of the Case

Example Transmit:

SCL	Length Low Byte	Length High Byte	Case Name
0x06	0x05	0x00	"Test1"

Cmd: 06 05 00 54 65 73 74 31

A.3.7 TRIG (Serial trigger to Tagsurance Tester Unit)

INIT STATE: inline
NEXT STATE: inline
FUNCTION: trigger a new test measurement specified in the initiated test case
TAGSURANCE RESPONSE: READY, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x07	Serial trigger to Tagsurance Tester Unit
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High
3	Device Name	…	Device Name

Example Transmit:

TCP Test	Length Low Byte	Length High Byte	Device Name
0x07	0x04	0x00	“Dev1”

Cmd : 07 04 00 44 65 76 31

A.3.8 READY (Serial Triggering Successful)

The Tagsurance response to successful ‘TRIG’ from client.

The Tagsurance will trigger a new measurement and wait for the triggered measurement to be performed. If there are multiple devices connected to the system, each of the subsystems one by one depending on the arrangement in the test case. If there are e.g. 2 subsystems, 2 trig commands are required to trigger both measurements.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x08	Serial triggering successful
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

TCP ready	Length Low Byte	Length High Byte
0x08	0x00	0x00

Cmd: 08 00 00

A.3.9 STOP (Stop Running Test Case)

INIT STATE: inline
NEXT STATE: idle
FUNCTION: stop the initiated test case
TAGSURANCE RESPONSE: STOPPED, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x09	Stop running case
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

STOP	Length Low Byte	Length High Byte
0x09	0x00	0x00

Cmd: 09 00 00

A.3.10 STOPPED (Test Case Stopped)

The Tagsurance response to 'STOP' from client.

The Tagsurance will stop the inline case that has been initiated. Response is provided after the case has been stopped and the system returned to idle state.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x0A	Test case stopped
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

STOPPED	Length Low Byte	Length High Byte
0x0A	0x00	0x00

Cmd: 0A 00 00

A.3.11 GTR (Get Test Result)

INIT STATE: inline, idle
NEXT STATE: no change
FUNCTION: acquire measurement data from the results buffer
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x10	Get Test Result
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

GTR	Length Low Byte	Length High Byte
0x10	0x00	0x00

Cmd: 10 00 00

A.3.12 TR (Test Result)

TR header is common but result data format depends on the calling command.

The Tagsurance response to: "GTR" from client
The Tagsurance will check the status of the inline case results buffer and return the first data that has not been read. For example, if "GTR" is provided after three triggered measurements, included in the response are the data from the measurement that was triggered first. Second "GTR" will return second dataset, etc.

Tagsurance response to: "READ/WRITE/SWEEP/KILL/UIDREAD" from client
In case of WRITE, result data includes an error code.
In case of READ, result data includes error code and the acquired data, separated by '/'.
In case of SWEEP, result data includes threshold values at test frequencies, separated by '\tab'.
In case of ENCODE/KILL, result data includes latest task indicator and error code, separated by: '/'.
In case of UIDREAD, result data includes error code and the acquired data, separated by '/'.

INTERPRETATION OF TASK IDENTIFIER BYTE		
0x00	0_{10}	Inventory (and access) sequence
0x01	1_{10}	EPC memory programming
0x02	2_{10}	User memory programming
0x03	3_{10}	Kill password programming
0x04	4_{10}	Access password programming
0x05	5_{10}	Lock bit programming
0x06	6_{10}	Kill procedure

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x11	Test Result
1	Length Low	0xXX	Length Low
2	Length High	0xXX	Length High
3	Failed/Passed	0x00 || 0x01	0x00 → Failed 0x01 → Passed
4	Group index	0xXX	Group index in multilane and multidevice test setup
5	Index$_{1,LSB}$	0xXX	Test Result Index (LSB)
6	Index$_{2}$	0xXX	Test Result Index
7	Index$_{3}$	0xXX	Test Result Index
8	Index$_{4,LSB}$	0xXX	Test Result Index (MSB)
9…	Result data	…	Other Test Results…
…	End Sequence	0x0D0A	End sequence

Example Transmit:

TR	Length Low Byte	Length High Byte	Failed	Group Index (0)	Result Index 1	Result Index 2	Result Index 3	Result Index 4	Result	End Sequence
0x11	0xXX	0xXX	0x00	0x00	0x02	0x00	0x00	0x00	“all information”	0x0D0A

Cmd: 11 1D 00 01 00 02 00 00 00 0A 31 33 3A 34 38 3A 33 39 09 50 41 53 53 09 50 41 53 53 09 31 0D 0A

The syntax of result data is similar to result lines in the Tagsurance 2 GUI log file. The first provided test result, after loading a new case with the “LSC” command, includes the log file header lines. Following test results consist of a single result line.

The example result data of the first “TR” after the case loading

```
Tagsurance        16.9.2016         16:04

Device specifications               Test device name
Section 0         TS-HF

Point test specifications           Test device       Protocol          Frequency [MHz]   Power [dBm]       Mode
Point 0           TS-HF             ISO 14443-A       13,029            15,693            Must respond

UID read specifications             Test device       Protocol          Frequency [MHz]   Power [dBm]       Repetitions       Tolerance
UID read 0        TS-HF             ISO 14443-A       14,743            15,300            1                 0



Case spesifications
Frequency [MHz}                     -                 13,029            14,743
Power [dBm]                         -                 15,693            15,300
Mode                                -                 Must respond      UID read 0


Results

Time stamp        Group pass/fail   Section 0         Point 0           UID read 0 data (err/data)
16:04:11          PASS              PASS              1                 0/00000000CBFAA1EC
```

The example result data of other (second, third and so on) “TR” after the case loading

```
16:04:14          PASS              PASS              1                 0/00000000CBFAA1EC
```

The example result data of simple encode command (successful EPC write + lock)

```
5/0
```

A.3.13 GBS (Get Buffer Status)

INIT STATE: inline, idle
NEXT STATE: no change
FUNCTION: test inline test results data buffer state
TAGSURANCE RESPONSE: BS, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x12	Get Buffer Status
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

GBS	Length Low Byte	Length High Byte
0x12	0x00	0x00

Cmd: 12 00 00

A.3.14 BS (Buffer Status)

The Tagsurance response to result buffer state query initiated by “GBS” from client.

The Tagsurance will count the amount of unread data in the results buffer and check if the data bank has overflowed. The maximum amount of unread data in the buffer is 65535 results.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x13	Get Test Result
1	Length Low	0x03	Length Low
2	Length High	0x00	Length High
3	Nlow	0xXX	Number of results in buffer (low byte)
4	Nhigh	0xXX	Number of results in buffer (high byte)
5	Buffer OVF	0x00 | 0x01	buffer OVF (0x01), no error (0x00)

Example Transmit:

BS	Length Low Byte	Length High Byte	Nlow	Nhigh	Buffer overflow
0x13	0x03	0x00	0x01	0x00	0x00

Cmd: 13 03 00 01 00 00

A.3.15 READ (Read Tag Memory)

INIT STATE: inline, idle
NEXT STATE: no change
FUNCTION: perform a single tag memory read
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x30	Simple Read Tag Memory
1	Length Low	0xXX	Length Low
2	Length High	0xXX	Length High
3	$F_{1,LSB}$	0xXX	Frequency [kHz] (low byte)
4	F_2	0xXX	Frequency [kHz]
5	F_3	0xXX	Frequency [kHz]
6	$F_{4,MSB}$	0xXX	Frequency [kHz] (high byte)
7	$P_{1,LSB}$	0xXX	Power Level (dB*1000+2^15) (low byte)
8	P_2	0xXX	Power Level (dB*1000+2^15)
9	P_3	0xXX	Power Level (dB*1000+2^15)
10	$P_{4,MSB}$	0xXX	Power Level (dB*1000+2^15) (high byte)
11	Memory Bank	0xXX	Tag Memory to be read
12	Word Address	0xXX	First word address to be read (0 to 127)
13	Word Count	0xXX	Number of words to be read (1 to 127)
14	Protocol[1]	0xXX	Optional, default: ISO18000-63 (if left undefined)
15…	Test device name	…	Optional, default: latest triggered or 1st in devices.txt

[1] 0x00 ISO18000-63, 0x01 ISO18000-6B, 0x02 ISO-18000-6B-D (UHF only)

Example Transmit:

READ	Length Low Byte	Length High Byte	$F_{1,LSB}$ (866MHz)	F_2	F_3	$F_{4,MSB}$	$P_{1,LSB}$ (2dBm)	P_2	P_3	$P_{4,MSB}$
0x30	0x12	0x00	0xD0	0x36	0x0D	0x00	0xD0	0x87	0x00	0x00

Bank (TID)	Addr	WCount (1 pcs)	Protocol (ISO18000-63)	Device name ("TS-UHF")
0x10	0x00	0x01	0x00	0x54532D554846

Cmd: 30 12 00 D0 36 0D 00 D0 87 00 00 10 00 01 00 54 53 2D 55 48 46

A.3.16 WRITE (Simple Write)

INIT STATE: inline, idle
NEXT STATE: no change
FUNCTION: perform a single tag memory write
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x31	Simple Read Tag Memory
1	Length Low	0xXX	Length Low
2	Length High	0xXX	Length High
3	$F_{1,LSB}$	0xXX	Frequency [kHz] (low byte)
4	F_2	0xXX	Frequency [kHz]
5	F_3	0xXX	Frequency [kHz]
6	$F_{4,MSB}$	0xXX	Frequency [kHz] (high byte)
7	$P_{1,LSB}$	0xXX	Power Level (dB*1000+2^15) (low byte)
8	P_2	0xXX	Power Level (dB*1000+2^15)
9	P_3	0xXX	Power Level (dB*1000+2^15)
10	$P_{4,MSB}$	0xXX	Power Level (dB*1000+2^15) (high byte)
11	Memory Bank	0xXX	Tag Memory to be written
12	Word Address	0xXX	First word address to be written (0 to 127)
13	Word Count	0xXX	Number of words to be written (1 to 127)
14…	Data	0x…	Data to be written
…	Mandatory Write?	0x0X	0x00: Use BlockWrite; 0x01: Use Mandatory Write[1]
…	Protocol[1]	0xXX	Optional, default: ISO18000-63 (if left undefined)
…	Test device name	0x…	Optional, default: latest triggered or 1st in devices.txt

[1] 0x00 ISO18000-63, 0x01 Mandatory Write (UHF only)

Example Transmit:

WRITE	Length	Length	$F_{1,LSB}$	F_2	F_3	$F_{4,MSB}$	$P_{1,LSB}$	P_2	P_3	$P_{4,MSB}$
	Low	High	(866MHz)				(2dBm)			
	Byte	Byte								
0x31	0x14	0x00	0xD0	0x36	0x0D	0x00	0xD0	0x87	0x00	0x00

Bank	Addr	WCount	Data	Mandatory	Protocol	Device name
(EPC)		(1 pcs)		Write?	(ISO18000-	("TS-UHF")
					63)	
0x01	0x02	0x01	0xABCD	0x00	0x00	0x54532D554846

Cmd: 31 14 00 D0 36 0D 00 D0 87 00 00 01 02 01 AB CD 00 00 54 53 2D 55 48 46

A.3.17 SWEEP (Simple Sweep)

INIT STATE: inline, idle
NEXT STATE: no change
FUNCTION: perform a single response threshold sweep for a tag
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x31	Simple Read Tag Memory
1	Length Low	0xXX	Length Low
2	Length High	0xXX	Length High
3	$F_{1,LSB}$	0xXX	Start Frequency [kHz] (low byte)
4	F_2	0xXX	Start Frequency [kHz]
5	F_3	0xXX	Start Frequency [kHz]
6	$F_{4,MSB}$	0xXX	Start Frequency [kHz] (high byte)
7	$F_{1,LSB}$	0xXX	Stop Frequency [kHz] (low byte)
8	F_2	0xXX	Stop Frequency [kHz]
9	F_3	0xXX	Stop Frequency [kHz]
10	$F_{4,MSB}$	0xXX	Stop Frequency [kHz] (high byte)
11	$F_{1,LSB}$	0xXX	Frequency Step [kHz] (low byte)
12	F_2	0xXX	Frequency Step [kHz]
13	F_3	0xXX	Frequency Step [kHz]
14	$F_{4,MSB}$	0xXX	Frequency Step [kHz] (high byte)
15…	Protocol[1]	0xXX	Optional, default: ISO18000-63 (if left undefined)
…	Test device name	0x…	Optional, default: latest triggered or 1st in devices.txt

[1] 0x00 ISO18000-63, 0x01 ISO18000-6B, 0x02 ISO-18000-6B-D, 0x03 GB/T 29768, 0x10 ISO15693, 0x11 ISO14443A, 0x12 ISO14443B, 0x13 FeliCa, 0x14 ISO18000-3M3, 0x15 ISO14443B–ST25TB, 0x16 TTO PR1101, 0x17 TTO PR1102, 0x18 TTO NFC Barcode

Example Transmit:

SWEEP	Length Low Byte	Length High Byte	Start $F_{1,LSB}$ (12MHz)	Start F_2	Start F_3	Start $F_{4,MSB}$	Stop $F_{1,LSB}$ (16MHz)	Stop F_2	Stop F_3	Stop $F_{4,MSB}$
0x32	0x12	0x00	0xE0	0x2E	0x00	0x00	0x80	0x3E	0x00	0x00

Step $F_{1,LSB}$ (100kHz)	Step F_2	Step F_3	Step $F_{4,MSB}$	Protocol (ISO15693)	Device name (“TS-HF”)
0x64	0x00	0x00	0x00	0x10	0x54532D4846

Cmd: 32 12 00 E0 2E 00 00 80 3E 00 00 64 00 00 00 10 54 53 2D 48 46

A.3.18 ENCODE (All-In-One)

INIT STATE: inline, idle
NEXT STATE: no change
FUNCTION: perform encoding and locking operations for tag memories
TAGSURANCE RESPONSE: TR, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x33	Encode tag
1	Length Low	0xXX	Length Low
2	Length High	0xXX	Length High
3	$F_{1,LSB}$	0xXX	Frequency [kHz] (low byte)
4	F_2	0xXX	Frequency [kHz]
5	F_3	0xXX	Frequency [kHz]
6	$F_{4,MSB}$	0xXX	Frequency [kHz] (high byte)
7	$P_{1,LSB}$	0xXX	Power Level (dB*1000+2^15) (low byte)
8	P_2	0xXX	Power Level (dB*1000+2^15)
9	P_3	0xXX	Power Level (dB*1000+2^15)
10	$P_{4,MSB}$	0xXX	Power Level (dB*1000+2^15) (high byte)
11…14	Access password[2]	0xXXXXXXXX	All 0's: no access, else: access after inventory
.	EPC memory action byte	0xXX	Operations to perform for EPC memory[3]
.	Word Address	0xXX	First word address to be written (0 to 127)[1]
.	Word Count	0xXX	Number of words to be written (1 to 127) [1]
.	EPC Data	0xXX	Data to be written[1]
.	USER memory action byte	0xXX	Operations to perform for USER memory[3]
.	Word Address	0xXX	First word address to be written (0 to 127)[1]
.	Word Count	0xXX	Number of words to be written (1 to 127) [1]
.	USER Data	0xXX	Data to be written[1]
.	KILL-PWD action byte	0xXX	Operations to perform for kill password[3]
.	KILL-PWD data	0xXX	Kill password (4 bytes, MSB first) [1]
.	ACCESS-PWD action byte	0xXX	Operations to perform for access password[3]
.	ACCESS-PWD data	0xXX	Access password (4 bytes, MSB first) [1]
.	Protocol[4]	0xXX	Optional, default: ISO18000-63 (if left undefined)
.	Test device name	0x…	Optional, default: latest triggered or 1st in devices.txt

[1] Defined only if data is programmed with data (action byte value is: 0x01…0x05).

[2] If access password provided is 0x00000000, access is not performed.

[3] Memory programming operations for different action byte values.

[4] 0x00 ISO18000-63 (UHF only)

Action byte	Encoding action	Write operation	Lock operation
0x00	no action	-	-
0x01	write	program data to memory	-
0x02	write + unlock	program data to memory	unlock memory
0x03	write + lock	program data to memory	lock memory[1]
0x04	write + permalock	program data to memory	lock memory permanently[1]
0x05	write + permaunlock	program data to memory	unlock memory permanently
0x06	unlock	-	unlock memory
0x07	lock	-	lock memory[1]
0x08	permalock	-	lock memory permanently[1]
0x09	permaunlock	-	unlock memory permanently

[1] EPC, USER: write protection, RESERVED (password memory): read and write protection.

Example Transmit 1:

Frequency 866MHz, Power 15dBm, no access after inventory
Write 4 bytes words to EPC memory, starting form word address 0x02, and set write lock on
Write access password (0x01020304)

ENCODE	**Length**	**Length**	**$F_{1,LSB}$**	**F_2**	**F_3**	**$F_{4,MSB}$**	**$P_{1,LSB}$**	**P_2**	**P_3**	**$P_{4,MSB}$**
	Low Byte	**High Byte**	**(866MHz)**				**(15dBm)**			
0x33	0x21	0x00	0xD0	0x36	0x0D	0x00	0x98	0xBA	0x00	0x00

ACC-PWD	**ACC-PWD**	**ACC-PWD**	**ACC-PWD**
(MSB)	**…**	**…**	**(LSB)**
0x00	0x00	0x00	0x00

EPC memory	**Word**	**Word**	**Data**	**USER memory**	**KILL-PWD**
action byte	**address**	**count**	**(4 bytes)**	**action byte**	**action byte**
(write + lock)	**(0x02)**	**(4 bytes)**		**(no action)**	**(no action)**
0x03	0x02	0x02	0x11223344	0x00	0x00

ACC-PWD	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**
action byte	**(MSB)**			**(LSB)**
(write)				
0x01	0x01	0x02	0x03	0x04

Protocol	**Device name**
(ISO 18000-63)	**(“TS-UHF”)**
0x10	0x54532D554846

Cmd: 33 21 00 D0 36 0D 00 98 BA 00 00 00 00 00 00 03 02 02 11 22 33 44 00 00 01 01 02 03 04 00 54 53 2D 55 48 46

<u>Example Transmit 2:</u>

Frequency 866MHz, Power 15dBm, no access

Lock EPC memory, and write and lock access password (0x01020304)

ENCODE	**Length**	**Length**	**$F_{1,LSB}$**	**F_2**	**F_3**	**$F_{4,MSB}$**	**$P_{1,LSB}$**	**P_2**	**P_3**	**$P_{4,MSB}$**
	Low Byte	**High Byte**	**(866MHz)**				**(15dBm)**			
0x33	0x1B	0x00	0xD0	0x36	0x0D	0x00	0x98	0xBA	0x00	0x00

ACC-PWD	**ACC-PWD**	**ACC-PWD**	**ACC-PWD**
(MSB)	**…**	**…**	**(LSB)**
0x00	0x00	0x00	0x00

EPC memory	**USER memory**	**KILL-PWD**
action byte	**action byte**	**action byte**
(no action)	**(no action)**	**(no action)**
0x00	0x00	0x00

ACC-PWD	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**
action byte	**(MSB)**			**(LSB)**
(write + lock)				
0x03	0x01	0x02	0x03	0x04

Protocol	**Device name**
(ISO 18000-63)	**(“TS-UHF”)**
0x10	0x54532D554846

Cmd: 33 1B 00 D0 36 0D 00 98 BA 00 00 00 00 00 00 00 00 00 03 01 02 03 04 00 54 53 2D 55 48 46

A.3.19 KILL

INIT STATE: inline, idle
NEXT STATE: no change
FUNCTION: perform a single kill operation for a tag
TAGSURANCE RESPONSE: TR, or ERR

Notice!

A tag will not accept kill command if kill password is zero. If the kill password provided with the command is zero, the Tagsurance will try to program kill password before attempting to kill the tag.

If kill password is locked, reprogramming can only be performed if the tag is in secured state. If the access password provided with the command is non-zero, the Tagsurance will perform inventory with access.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x34	Kill tag
1	Length Low	0x0B	Length Low
2	Length High	0x00	Length High
3	$F_{1,LSB}$	0xXX	Frequency [kHz] (low byte)
4	F_2	0xXX	Frequency [kHz]
5	F_3	0xXX	Frequency [kHz]
6	$F_{4,MSB}$	0xXX	Frequency [kHz] (high byte)
7	$P_{1,LSB}$	0xXX	Power Level (dB*1000+2^15) (low byte)
8	P_2	0xXX	Power Level (dB*1000+2^15)
9	P_3	0xXX	Power Level (dB*1000+2^15)
10	$P_{4,MSB}$	0xXX	Power Level (dB*1000+2^15) (high byte)
11…14	Access password[1]	0xXXXXXXXX	0x00: no access, else: access after inventory
15…18	Kill password[2]	0xXXXXXXXX	0x00: written before kill
19	Protocol[3]	0xXX	Optional, default: ISO18000-63 (if left undefined)
20	Test device name	0x…	Optional, default: latest triggered or 1st in devices.txt

[1] If access password provided is 0x00000000, access is not performed.

[2] If kill password provided is 0x00000000, password memory is programmed before kill operation. This will require access, if reserved memory is locked, and access password is non-zero.

[3] 0x00 ISO18000-63 (UHF only)

Example Transmit: Frequency 866MHz, Power 15dBm, no access, no pre-programmed kill password.

KILL	Length	Length High	$F_{1,LSB}$	F_2	F_3	$F_{4,MSB}$	$P_{1,LSB}$	P_2	P_3	$P_{4,MSB}$
	Low Byte	Byte	(866MHz)				(15dBm)			
0x34	0x17	0x00	0xD0	0x36	0x0D	0x00	0x98	0xBA	0x00	0x00

ACC-PWD	ACC-PWD	ACC-PWD	ACC-PWD	KILL-PWD	KILL-PWD	KILL-PWD	KILL-PWD
(MSB)	…	…	(LSB)	(MSB)	…	…	(LSB)
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Protocol	Device name
(ISO 18000-63)	(“TS-UHF”)
0x10	0x54532D554846

Cmd: 34 17 00 D0 36 0D 00 98 BA 00 00 00 00 00 00 00 00 00 00 00 54 53 2D 55 48 46

A.3.20 UID READ (Read Tag's ID)

INIT STATE: inline, idle
NEXT STATE: no change
FUNCTION: perform a single tag memory read
TAGSURANCE RESPONSE: TR[1], or ERR

[1] The data returned by a successful UIDRead depends on the protocol:

ISO 15693	UID
ISO 14443A	UID
ISO 14443B-ST25TB	UID
ISO 18000-3M3	TID data depending on word pointer and word count defined in the command
TTO PR1101	UID
TTO PR1102	UID
TTO NFC Barcode	Barcode

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0x35	Simple Read Tag'd ID
1	Length Low	0xXX	Length Low
2	Length High	0xXX	Length High
3	$F_{1,LSB}$	0xXX	Frequency [kHz] (low byte)
4	F_2	0xXX	Frequency [kHz]
5	F_3	0xXX	Frequency [kHz]
6	$F_{4,MSB}$	0xXX	Frequency [kHz] (high byte)
7	$P_{1,LSB}$	0xXX	Power Level (dB*1000+2^15) (low byte)
8	P_2	0xXX	Power Level (dB*1000+2^15)
9	P_3	0xXX	Power Level (dB*1000+2^15)
10	$P_{4,MSB}$	0xXX	Power Level (dB*1000+2^15) (high byte)
11	$wPtr_{1,LSB}$	0xXX	Word pointer (low byte)[2]
12	$wPtr_2$	0xXX	Word pointer[2]
13	$wPtr_3$	0xXX	Word pointer[2]
14	$wPtr_{4,MSB}$	0xXX	Word pointer (high byte)[2]
15	wCount	0xXX	Word count to be read[2]
16	Protocol[1]	0xXX	Optional, default: ISO15693 (if left undefined)
17…	Test device name	…	Optional, default: latest triggered or 1st in devices.txt

[1] 0x10 ISO15693, 0x11 ISO14443A, 0x14 ISO18000-3M3 (HF only), 0x15 ISO14443B-ST25TB, 0x16 TTO PR1101, 0x17 TTO PR1102, 0x18 TTO NFC Barcode

[2] Required parameter, but only takes effect with ISO 18000-3M3

Example Transmit:

READ	**Length LowByte**	**Length HighByte**	**$F_{1,LSB}$ (13,5MHz)**	**F_2**	**F_3**	**$F_{4,MSB}$**	**$P_{1,LSB}$ (10dBm)**	**P_2**	**P_3**	**$P_{4,MSB}$**
0x35	0x0E	0x00	0xBC	0x34	0x00	0x00	0x10	0xA7	0x00	0x00

$wPtr_{1,LSB}$ (0)	**$wPtr_2$**	**$wPtr_3$**	**$wPtr_{4,MSB}$**	**wCount (0)**	**Protocol (ISO15693)**	**Device name ("TS-HF")**
0x00	0x00	0x00	0x00	0x00	0x10	0x54532D4846

Cmd: 35 0E 00 BC 34 00 00 10 A7 00 00 00 00 00 00 00 10 54 53 2D 48 46

A.3.21 TCP Test

INIT STATE: handshake (within 10s from opening TCP connection)
NEXT STATE: no change
FUNCTION: perform handshake with the Tagsurance 2 GUI
TAGSURANCE RESPONSE: TCP Ready, or ERR

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0xF0	Test TCP connection
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

TCP Test	Length Low Byte	Length High Byte
0xF0	0x00	0x00

Cmd: F0 00 00

A.3.22 TCP Ready

Tagsurance answer to ‘TCP Test’ from client.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0xF1	Connect
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High

Example Transmit:

TCP Ready	Length Low Byte	Length High Byte
0xF1	0x00	0x00

Cmd: F1 00 00

A.3.23 ERR (Error from Tagsurance)

The Tagsurance response in case of an error.

Byte No.	Function	Value (HEX)	Description
0	Operation code, command	0xFF	Error from Tagsurance
1	Length Low	0x00	Length Low
2	Length High	0x00	Length High
3	Parameter Byte	0x00 ... 0xFF	Error code

Example:

ERR	Length Low Byte	Length High Byte	Error Code
0xFF	0x01	0x00	0x00

Cmd: FF 01 00 01

Error codes:

Code	Command	Description
0x00	Invalid command	The command provided was not valid
0x01	Handshaking failed	‘TCP test’ not received during handshake
0x02	Device busy	Device is busy and command cannot be performed
0x03	RS232 connection lost	No connection to Tagsurance Tester unit
0x04	Bad request	The requested action cannot be performed in current operation mode
0x05	Reception timeout	Timeout occurred during command reception
0x06	Framesync error	Unexpected response from subsystem
0x10	Case Init Error 0	Case file not found
0x11	Case Init Error 1	Case file is corrupted
0x12	Case Init Error 2	Case parameters are invalid
0x13	Case Init Error 3	Licence options exceeded
0x21	Invalid parameters	Command parameters are invalid or insufficient
0x22	Licence error	Current licence does not allow the action requested
0x30	Buffer Empty	No data available in the result data buffer
0x31	Buffer OVF	Result data buffer has overflowed
0x32	Buffer Full	Result buffer is full (size: 2^16-1)

Appendix B
Timing calculations

B.1 Tagsurance HF

Calculation of the test case execution time

Case execution times depend on type of tests as well as protocol to be used for the tests. To calculate an estimate of the whole test sequence, individual test times for each test have to be estimated first and then they are summed up by applying the following equation:

$$T_{case} = T_{trigger} + T_{test1} + T_{test2} + \cdots + T_{testX}$$

where:

$T_{trigger}$ is the time for the device to start the first test in the case. The value is

$$T_{trigger} = 1{,}04$$

T_{test} is the time it takes to execute each individual test defined in the case

Equations for estimating execution time for individual tests are provided in the following chapters and the protocol dependent time constants to be used with the equations are listed in the table below.

Protocol and test type	k	m	$T_{interpret}$
14443A Single UID Read Test	2,84	16	2,8
14443A Double UID Read Test	4,63	24,56	2,8
15693 UID Read Test	11,25	4,89	9,6
18000-3M3 UID Read Test	12,13	47	2 x wCount
15693 Point Test	11,13	4,95	2,76
14443A Point Test	0,53	4,98	1,6
14443B Point Test	3,65	5	9
FeliCa Point Test	6,43	5	8,77
18000-3M3 Test	1,33	5	1,81
14443B-ST25TB Point Test	2,25	4,8	5,38
14443B-ST25TB UID Read Test	10,22	43,44	30

Point and UID Read Test time estimation

Point test and UID Read test execution times can be calculated by applying the following equation:

$$T_{test} = T_{initPoint} + T_{COM} + T_{interpret}$$

where:

$T_{initPoint}$ is the time it takes for each test to setup the device. For point tests this time is

$T_{initPoint} = 0{,}76$

T_{COM} is the time spent communicating with a tag and is depending on the carrier frequency and protocol and can be calculated by applying the following equation

$$T_{COM} = \frac{13{,}56}{f} * k + m$$

where:

f is the carrier frequency in MHz and $k\ m$ are protocol dependent constants.

Sensitivity test time estimation

Sensitivity test execution time can be calculated by applying the following equation:

$$T_{test} = \ T_{initSens} + \left(T_{subInit} + T_{COM} + T_{interpret}\right) * N$$

where:

$T_{initSens}$ is the time it takes for each sensitivity test to setup the device. The time is $T_{initSens} = 1{,}15$

$T_{subInit}$ is the time it takes to setup the device in each iteration. The time is

$T_{subInit} = 3{,}51$

N is the number of iterations that will be in the sensitivity test. It can be calculated according to:

$$N = ceil\left(\log_2 \frac{P_{high} - P_{low}}{Uncertainty}\right) + 1$$

where:

P_{high} is the highest allowed power

P_{low} is the lowest allowed power and $Uncertainty$ is the allowed uncertainty of the test.

B.2 Tagsurance UHF

Case execution time depends on the amount and type of tasks included. This can be estimated by using the equations below. Results are expressed in milliseconds (ms).

- Point test takes about: 3,6 + carrier_time
- Read test takes about: (16,7 + carrier_time + 0,5 x N_{word_count}) x $N_{repetitions}$.
- BlockWrite test takes about: (35,4 + carrier_time + 0,6 x N_{word_count}) x $N_{repetitions}$.
- Mandatory Write test takes about: (35,4 + carrier_time + 0,6) x N_{word_count} x $N_{repetitions}$.
- Sweep test time consumption depends on the tag and the sweep parameters. A rough estimate is given by: (($f_{end,k}$ / $f_{start,k}$) / $f_{step,k}$ + 1) x 5 x 5ms.
- Sensitivity test takes about: 2,7 + carrier_time + ($N_{iterations}$ - 1) x (3,8 + carrier_time), where $N_{iterations}$=ceil($\log_2$((highest test level – lowest test level) / uncertainty) ϵ[1,9]

Appendix C
Summary of Tagsurance UHF Serial Interface Commands

DEVICE SETTINGS

SYSTEM CONFIGURATION			
Function	**Command**	**Response**	**Next state**
Enable external trigger (default)	’PTE’	(none)	IDLE
Disable external trigger	’PTI’	(none)	IDLE
Set trigger (ext) to rising edge (default)	’PTH’	(none)	IDLE
Set trigger (ext) to falling edge	’PTL’	(none)	IDLE
Enable internal gen2 chip	’POC’	(none)	IDLE
Disable internal gen2 chip (default)	’POA’	(none)	IDLE
Set carrier time (default: 480=[1][224]=2,5ms)	'PC'[time_high][time_low]	(none)	IDLE
Set pass signal to static mode (default)	'PIPS'	(none)	IDLE
Set pass signal to pulsed mode	'PIPP'	(none)	IDLE
Enable “failed = HIGH” option	’PIPM’	(none)	IDLE
Disable “failed = HIGH” option (default)	’PIPN’	(none)	IDLE
Save settings	'PFS'	(none)	IDLE
Load saved settings	'PFL'	[fbyte$_L$][fbyte$_H$] [time$_H$][time$_L$]	IDLE
Reset factory default settings	'PFR'	(none)	IDLE

INTERPRETATION AND DEFINITION OF CARRIER TIME
calculation formula
carrier_time[ms] = time/160 – 0,5ms
time = (carrier_time[ms] + 0,5ms) * 160
valid range
112 - 560 = 0,2ms - 3ms

INTERPRETATION OF THE FBYTE DATA		
bit	**FBYTE$_L$**	**FBYTE$_H$**
0	trigger mode (0->ext, 1->int, default: ext)	(RFU)
1	trigger edge (0->falling, 1->rising, default: rising)	(RFU)
2	pass signal mode (0->static, 1->pulsed, default: static)	(RFU)
3	failed = HIGH option (0->disabled, 1->enabled, default: disabled)	(RFU)
4	(RFU)	(RFU)
5	(RFU)	(RFU)
6	(RFU)	(RFU)
7	(RFU)	(RFU)

PERFORMANCE TESTING

INLINE MEASUREMENT COMMANDS			
Function	**Command**	**Response**	**Next state**
Upload case data		error code[4]	IDLE
• header	[‘L’][Data Length H][Data Length L]		
• data for point tests	[‘P’][Tolerance][Npoints][Mode + f_H][f_L][Power]		
• data for read tasks	[‘R’][Bank + f_H][f_L][Power][Word Pointer] [Word Count][Repetitions + Tolerance]		
• data for write tasks	[‘W’][6] [Bank + f_H][f_L][Power][Word Pointer] [Repetitions + Tolerance][Increment] [Word Count][Data]		
• data for sweep task	[‘S’][Start f_H][Start f_L] [Stop f_H][Stop f_L] [Step f_H][Step f_L]		
• data for sensitivity test	[‘C’][f_H][f_L][P_L][P_H][P_{LCL}][P_{UCL}][Uncertainty]		
• data for fast EPC[7]	[‘M’][wCount]	[Err][Data][8]	IDLE
Start (loaded) case	‘C’	error code[4]	WAIT FOR TRIG[2]
Trigger	‘T’	pass/fail &data[5]	WAIT FOR TRIG
Stop (running) case[1]	‘X’	(none)	IDLE
Save (active) case	[‘F’][case number (0-4)]	error code[4]	IDLE
Load saved case	[‘O’][case number (0-4)]	error code[4]	IDLE
Get active case data	‘PD’	case data[3]	IDLE

[1] Any char other than ‘T’ also stops the running case and causes the device to return to IDLE state
However, for the sake of simplicity, ‘X’ is recommended

[2] If no valid case is loaded, the device will return error and return to IDLE state

[3] If no valid case is loaded, the device will return {0x00, 0x00} (i.e. number of data bytes is zero)

[4] For description of the error codes, see 10.7: “Error Handling”

[5] For description of the data, see section 10.3.2

[6] Use ‘V’ for MandatoryWrite

[7] Only for ISO 18000-6C

[8] err (0x00 = no error) followed by the data read (repeated string of 0x00’s if read was failed)

OTHER MEASUREMENT FUNCTIONS			
Function	**Command**	**Response**	**Next state**
Threshold sweep	[‘S’][Start f_H][Start f_L][Stop f_H][Stop f_L][Step f_H][Step f_L]	[Err][N Bytes][Data][1]	IDLE
Read	[‘R’][Bank + f_H][f_L][Power][Word Pointer] [Word Count][Repetitions + Tolerance]	[Err][ErrByte][Data][2]	IDLE
BlockWrite	[‘W’][Bank + f_H][f_L][Power][Word Pointer][Word Count][Data]	[Err][ErrByte][3]	IDLE

[1] Byte count and data will follow if the command is valid (user must check the error flag, 0x00=no error)

[2] errByte (0x00 = no error) followed by the data read (repeated string of 0x00’s if read was failed)

[3] errByte (0x00 = no error)

ENCODING, LOCKING, AND KILLING

TAG ENCODING, LOCKING, AND KILLING FUNCTIONS			
Function	**Command**	**Response**	**Next state**
BASIC COMMANDS			
Encode	[‘H’][Data Length H][Data Length L] [f$_H$][f$_L$][Power][Acc-pwd$_{1,MSB}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$] [EPC Word count][EPC Word pointer][EPC Data] [User Word count][User Word pointer][User Data] [Enab kill-pwd prog][Kill-pwd$_{1,MSB}$][Kill-pwd$_2$][Kill-pwd$_3$][Kill-pwd$_4$] [Enab acc-pwd prog][Acc-pwd$_{1,MSB}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$] [Enab lock prog][Payload$_{1,MSB}$][Payload$_2$][Payload$_3$]	[Err][1][Task][ErrByte][2-4]	IDLE
Lock	[‘D’][f$_H$][f$_L$][Power][Repetitions + Tolerance] [Payload$_{1,MSB}$][Payload$_2$][Payload$_3$]	[Err][1][ErrByte][2,4]	IDLE
Kill	[‘K’][f$_H$][f$_L$][Power] [Acc-pwd$_{1,MSB}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$] [Kill-pwd$_{1,MSB}$][Kill-pwd$_2$][Kill-pwd$_3$][Kill-pwd$_4$]	[Err][1][Task][ErrByte][2-4]	IDLE
CUSTOM COMMAND SET			
Inventory	[‘I’] [f$_H$][f$_L$][Power][Acc-pwd$_{1,MSB}$][Acc-pwd$_2$][Acc-pwd$_3$][Acc-pwd$_4$]	[Err][1][ErrByte][2,4]	IDLE/ENCODING
BlockWrite	[‘E’][‘W’][Bank][Word pointer][Word count][Data]	[Err][1][ErrByte][2,4]	ENCODING
Read	[‘E’][‘R’][Bank][Word pointer][Word count]	[Err][1][ErrByte][2,4]	ENCODING
Lock	[‘E’][‘L’][Payload$_{1,MSB}$][Payload$_2$][Payload$_3$]	[Err][1][ErrByte][2,4]	ENCODING
Kill	[‘E’][‘K’] [Kill-pwd$_{1,MSB}$][Kill-pwd$_2$][Kill-pwd$_3$][Kill-pwd$_4$]	[Err][1][ErrByte][2,4]	ENCODING
End session	[‘X’]	-	IDLE

[1] Err represents system error code (0x00=no error). For description of the codes, see Section 10.7: “Error Handling”

[2] ErrByte (and task) will follow Err if the command is valid (Err=0x00)

[3] Task indicates the last task performed before end session (See task indices table below)

[4] ErrByte represents error code from the last task (0x00 = no error)

INTERPRETATION OF TASK IDENTIFIER BYTE		
0x00	**0$_{10}$**	Inventory
0x01	**1$_{10}$**	EPC memory programming
0x02	**2$_{10}$**	User memory programming
0x03	**3$_{10}$**	Kill password programming
0x04	**4$_{10}$**	Access password programming
0x05	**5$_{10}$**	Lock bit programming
0x06	**6$_{10}$**	Kill procedure

COMBINING ENCODING WTIH PERFORMANCE TESTING			
Function	**Command**	**Response**	**Next state**
Activate encode after performance test option	‘PME’	(none)	IDLE
Deactivate encode after performance test option	‘PMD’	(none)	IDLE
Reinitialize inline case	‘C’	Err[1]	WAIT FOR TRIG

[1] Err represents system error code (0x00=no error). For description of the codes, see Section 10.7: “Error Handling”