Project Number: **14011**	Document Type: **Software Requirement Specification**	
Date: **12-May-2025**	Product Name: **End-Of-Line (EOL) Test**	Revision: **A**

History:

Rev	Date	Comment	Author
A	03-Mar-2025	Initial	Thomas Wiedebusch

Copyright © 2025 by
SENSUS GmbH
CONFIDENTIAL

REVIEWS

Rev	Date	Reviewer	Department	Function

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

CONTENTS

1 General Part 7
1.1 Introduction 7
1.2 Document Content 7
1.3 System Overview 8
2 Prerequisites 9
2.1 Hardware Requirements 9
2.2 Software Requirements 9
2.3 Network Connection 9
2.4 Mini-MOXA Installation 9
2.5 Mini-MOXA Setup 10
3 End of Line Test 11
3.1 FinalTest.xaml 11
3.2 FinalProcessController 11
3.2.1 ProcessStateDef 11
3.2.2 State Machine 11
3.3 DetectCordonel 11
3.4 CheckNewMeter 11
3.5 ConnectCordonel 11
3.6 GetSerialNumberAndRadioAddress 11
3.7 CheckProductionState 11
3.7.1 Check EOL Progress 11
3.7.2 Check Requirements 12
3.7.3 Check ProductionOrderNo 12
3.7.4 Check PcbId 12
3.7.5 Check Region 12
3.7.6 Check LUT_CRC 12
3.7.7 Check Meter Size 12
3.7.8 Check FW and Apps 12
3.7.9 Check Remaining Battery Load 12
3.7.10 Check Remaining Lifetime 12
3.7.11 Check Storage Months 13
3.7.12 Check Preadjustment 13
3.7.13 Check Flow Test State 13
3.7.14 Check Calibration Factors 13
3.7.15 Check Complete Flow Production State 13
3.8 GetFinalParametrization 13

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

3.9 ExecPasswordFile 14
3.10 CheckOrderNumber 14
3.11 CheckSerialNumber 14
3.12 ExecFinalParametrization 14
3.12.1 Prepare Programming 14
3.12.2 Execute Final Parametrization 14
3.12.3 Reset Accumulators 15
3.12.4 Finalize Programming 15
3.13 ExecStoreConfiguration 15
3.14 ExecRebootCordonel 15
3.15 CheckFinalParametrization 16
3.15.1 Read Registers 16
3.15.2 Compare Registers 16
3.15.3 Check Parametrization Sequence 17
3.16 CheckRadio 17
3.16.1 Preparation 17
3.16.2 RSSI Test 17
3.16.3 Finalization 17
3.17 CheckAttachments 18
3.18 VisualInspection 18
3.19 PrepareShipping 18
4 Restrictions and Uncertainties 18
5 Contacts 18

TABLES

Table 1.1 Abbreviations 6
Table 3.1 Un-Reversed Programming Parameters 14
Table 3.2 Prepare Programming Parameters 14
Table 3.3 Reset Accumulators Parameters 15
Table 3.4 Finalize Programming Parameters 15
Table 3.5 Set Cordonel Time 15
Table 3.6 Clear Reset-Counters 16
Table 3.7 Excluded Registers from Reading 16
Table 3.8 Excluded Registers from Comparison 17
Table 3.9 Readout Cordonel Parameters 17
Table 3.10 Prepare RSSI Test 17
Table 3.11 Finalize Radio Check - Write 18

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

Table 3.12 Finalize Radio Check – Read and Compare 18
Table 5.1 Contacts 18

FIGURES

Figure 1.1 System Overview 8
Figure 2.1 Mini-MOXA Serial Ports setup 10

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

ABBREVIATIONS

CSD	Customer specific document, special requirement of the customer for setup of the individual Cordonel™
Common	Implementation of commonly used functions
Cordonel™	Water meter with ultra-sonic measurement, initial name GENESIS
CRC	Cyclic Redundancy Check – checksum to identify a correct code
DB	Data Base
EMEA	European, Middle East and Africa regions for the Cordonel™
EOL	End Of Line – used for preparing for shipping as last process in production
FW	Firmware as identifier for the installed applications to the Cordonel™
GENESIS	Legacy name for the Cordonel™
GTB	GENESIS Toolbox, selection of tools to access to the Cordonel™ internals
HW	Hardware
ID	Identification number
IrDA	Infrared Data Association, used as serial communication interface
LED	Light Emitting Diode, used as serial communication interface
LUT	Metrology Look-Up-Table for the Cordonel™
NA	North America region for the Cordonel™
NVM	Non-Volatile-Memory – permanent memory for parameters in the Cordonel™
PCB	Printed Circuit Board, the Cordonel™ electronics
SENTINEL	Program to crosscheck the EOL results and production status of the Cordonel
SRS	Software Requirement Specification
SAV	Sensus Order Administration - Sensusauftragsverwaltung (German)
SW	Software
tbd	to be defined
TBF	Test Bench Framework, the entire SW for the test benches and meters
UM	User Manual
VAKO	Variant configurator
VPN	Virtual Private Network

Table 1.1 Abbreviations

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

1 General Part

1.1 Introduction

The End-Of-Line test (EOL) is the final test which prepares the Cordonel™ for shipping. This final production process of the device takes all information collected during the preceding production processes and checks that those have been carried out successfully. It takes special approvals into account regarding the electronics age (production date until now), the required battery life time, the radio signal strength test, the required firmware (FW) and metrology Lock-Up-Table (LUT).

After the approval of all production processes, the final parametrization, based on the standard setup of the product, the special requirements covered by the VAKO and customer specific setups covered by the CSD, will take place.

To lock the device from unauthorized access, a password file will be generated and installed and the password will be updated to the data base enabling the login to the device.

After those configuration processes, the Cordonel will be rebooted and all parameters will be read out and checked for equality. This guarantees that all settings are going to survive an accidentally reset of the device in the field.

For EMEA regions the Cordonel radio will be tested for the correct installed radio key and the signal strength.

Finally, all alarms and the accumulator are going to be reset and all attachment will be checked.

1.2 Document Content

This document covers the required hardware and software for the EOL test. It states all requirements for this test as base for the unit test of the software.

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

1.3 System Overview

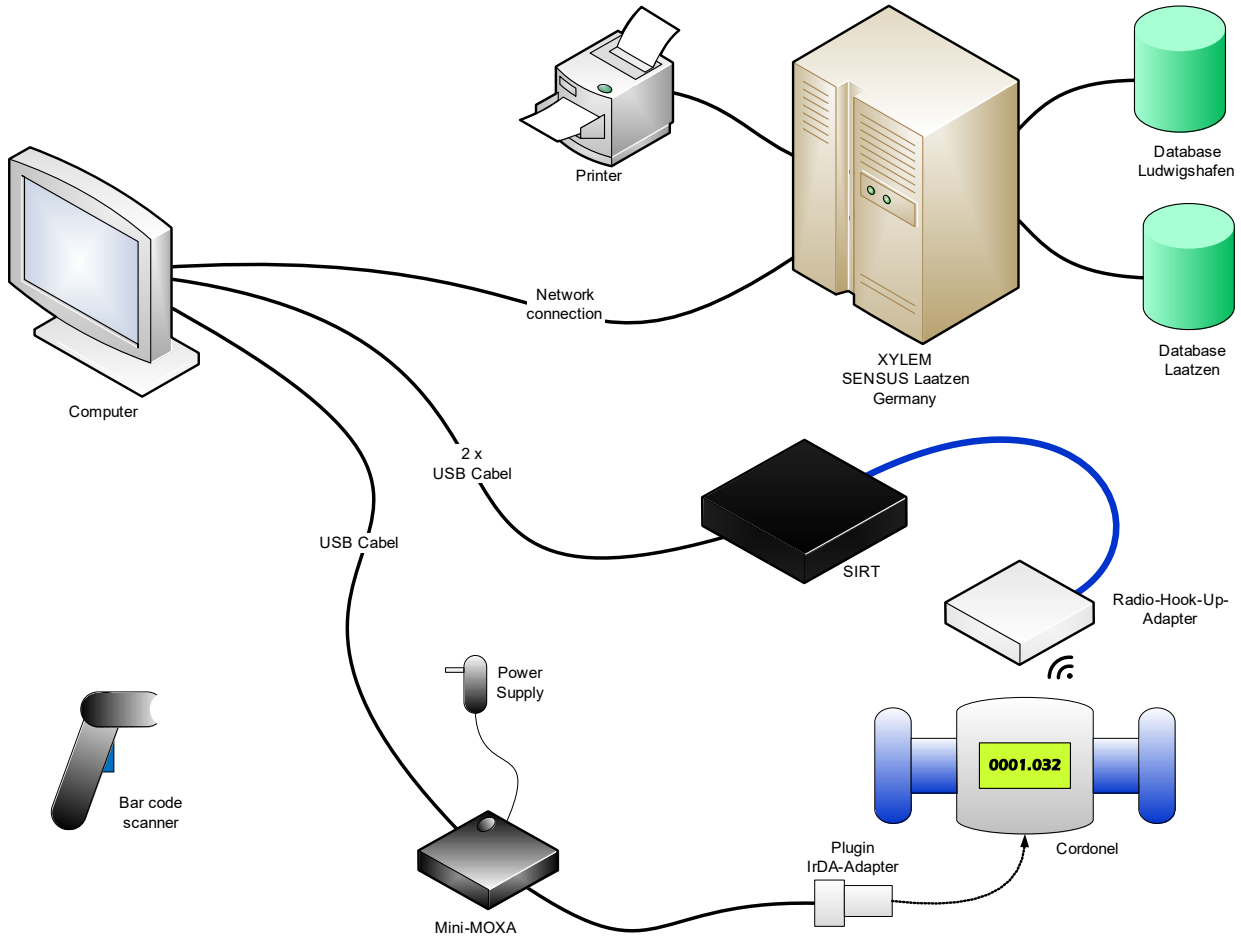


Figure 1.1 System Overview

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

2 Prerequisites

This chapter lists the SW and HW needed for the EOL test.

2.1 Hardware Requirements

- A PC with MS-Windows™ 10 or higher operating system.
- A Mini-MOXA or Micro-MOXA with IrDA head or direct USB to IrDA-Head,
- For the EMEA-Region a SIRT-Test-Box containing two SIRTs one for 433 MHz and one for 866 MHz and the radio-hook-up-adapter,
- A QR-code- / barcode-scanner.

2.2 Software Requirements

- Microsoft™ .NET framework 4.8.1 runtime or higher,
- Adobe™ Reader for software manuals,
- Webbrowser for inspection and/or change of production requirements.

2.3 Network Connection

A direct network connection has to be established to access the SENSUS Laatzen network. This connection is needed to get a valid licence for this software, to access the Cordonel password, to download the latest used interface-description-file (`configuration.json`) and all production relevant information.

2.4 Mini-MOXA Installation

To support the Mini-MOXA a driver has to be loaded from the Share-Point:

http://project.emea.sensus.net/PWA/EMEAAP_Genesis/Project Documents/300_R+D/340_Software+Firmware/342_Software_and_Firmware/343_Driver/driv_win_uport_v3.0_build_18080316_whql.exe

Download and install this before connecting the Mini-MOXA.

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

2.5 Mini-MOXA Setup

After the Mini-MOXA installation, the serial ports have to be setup to “RS-485 4W” as this is the IrDA-Plug-In HW-interface. Therefore, start the “Device Manager” with administrator rights, select “Multi-port serial adapters->MOXA UPort 1250 Series” and adjust both serial ports as shown:

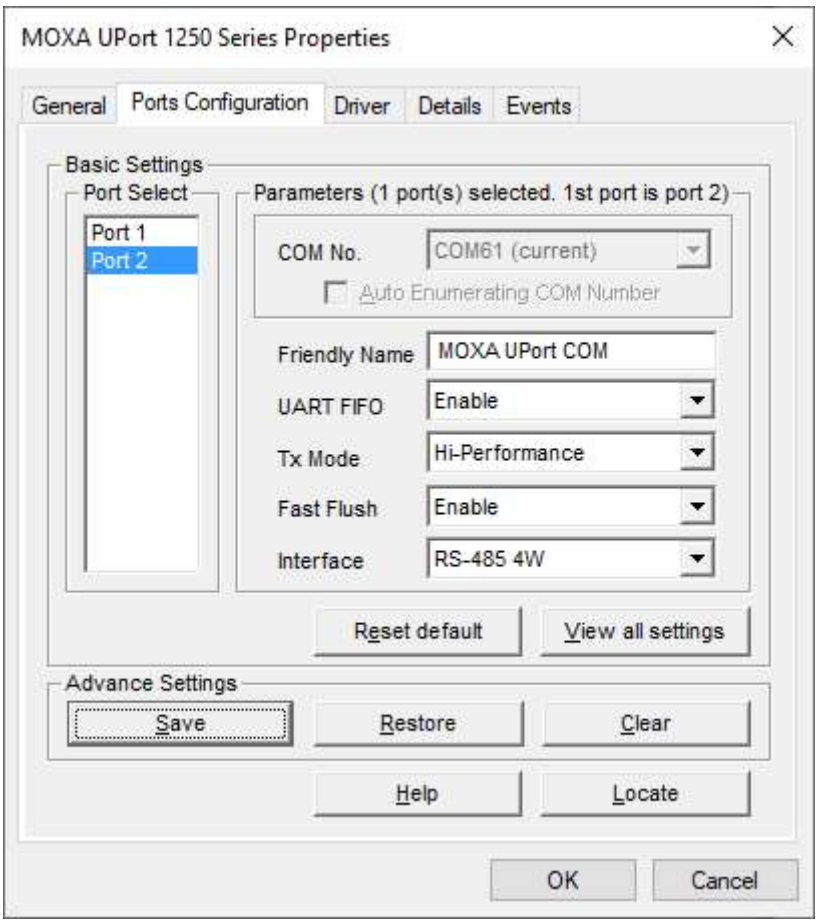


Figure 2.1 Mini-MOXA Serial Ports setup

The Mini-MOXA status LEDs underneath the small round window in the Mini-MOXA box will show the setup of the interface (red LEDs) and blink on an active communication (yellow LEDs). Check the 4W RS-485 LEDs being active.

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

3 End of Line Test

The end of line test should be split to several sub-processes. This is required to isolate functionalities to logically groups. With this design new tests could expand the EOL test easily. Also, this will support the reordering of processes in case of new requirements. The most valuable advantage is, that each process could be implemented and tested separately without influence to the other processes and that multiple processes can be run in parallel if those are not depending on data from any preceding process.

3.1 FinalTest.xaml

Shall implement the User interface for the EOL.

3.2 FinalProcessController

This controller shall implement the sequencing of the EOL process. It shall handle errors and user breaks on demand.

3.2.1 ProcessStateDef

All processes shall be listed in a definition file as single source. This should to combine the states for the state machine, the processes and the state information message for logging and user interaction.

3.2.2 State Machine

A state machine shall be used to sequence through the individual processes.

3.3 DetectCordonel

The detect Cordonel shall read the PcbId of the actual attached meter on bidirectional communication line.

3.4 CheckNewMeter

The check new meter shall detect if the meter has been changed. With this procedure a retry of the process can be speed up, as a reload of data base contents can be skipped as long as the meter stays identical.

Soon a new meter has been connected, all collected data for the old meter shall be dismissed and reloaded from the database to avoid wrong parametrization.

3.5 ConnectCordonel

The connect Cordonel shall log-in with the password acquired from the database and read out all installed applications, build a list of all supported registers and log the collected information.

3.6 GetSerialNumberAndRadioAddress

The get serial number and radio address process shall request that information from the database.

3.7 CheckProductionState

The check production state process shall require all needed information from the database to validate the executed production steps and the preconditions needed for finalizing the Cordonel. This includes the production requirements as base for the storage time and battery consumption, the remaining lifetime threshold and the requirement for the radio test.

3.7.1 Check EOL Progress

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_001]

Shall generate a new EOL progress data struct being able to fill it with the test results of the EOL processes.

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

3.7.2 Check Requirements

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_002]

Shall check if any requirement is defined being active for this Cordonel.

3.7.3 Check ProductionOrderNo

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_003]

Shall compare the Cordonel production-order with the production-order from the requirement to validate that the correct requirements going to be used for the check of the actual production state.

3.7.4 Check PcbId

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_004]

Shall compare the PcbId from Cordonel, EOL-progress and requirement.

3.7.5 Check Region

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_005]

Shall compare the Region from Cordonel and requirement.

3.7.6 Check LUT_CRC

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_006]

Shall check for the checksum (cyclic redundancy check - CRC) of the metrology lookup-table.

3.7.7 Check Meter Size

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_007]

Shall compare the Meter Size from Cordonel and requirement.

3.7.8 Check FW and Apps

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_008]

Compare Cordonel installed Firmware and Apps with required Firmware.

3.7.9 Check Remaining Battery Load

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_009]

Shall compare the remaining battery load from Cordonel and the requirement.

3.7.10 Check Remaining Lifetime

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_010]

Shall compare the remaining life time reported by the Cordonel and the requirement.

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

3.7.11 Check Storage Months

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_011]

Shall compare the production date reported by the Cordonel with the actual date and check the requirement.

3.7.12 Check Preadjustment

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_012]

Shall check the database content reporting a flow-test pre-adjustment.

3.7.13 Check Flow Test State

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_013]

Shall check the database content reporting the flow-test state.

3.7.14 Check Calibration Factors

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_014]

Shall compare the calibration values reported by the Cordonel with the flow-test calibration values reported by the database.

3.7.15 Check Complete Flow Production State

[SRS_TBF_EOL_PP_CHECK_PROD_STATE_015]

Shall check the entire production process sequence for the production state, except the database calls, with all underlaying sub-processes.

3.8 GetFinalParametrization

The programming parameters shall be required from a WEB service. This WEB-API has to collect all data based on the standard setup for this Cordonel, the VAKO and the CSD. The priority rule for those parameters is lowest for the standard, middle for VAKO and highest to CSD, does mean if any parameter is doubled in the list, the higher priority has to be taken.

Each parameter contains a name which is in the form “APPLICATION_Register” and states directly the register and a byte array with the parameters. As these values need a special handling they will be taken “as is” or in reversed byte order.

The standard is to reverse the order. Exceptions from these are of data type “string”, “uint48, “uint72”, “uint88”, “sint8”, “enum8”, “uint96” and “uint128” and some explicitly defined registers defined in the following table. The data format shall be taken from the actual `configuration.json` which is the interface description file between the FW and the SW.

The process getting the programming parameters from the database shall convert all byte values to the correct order being able to use the byte array directly to write it to the register.

Name	Data Type	Description
SENSUSRADIO_Authentification	UInt96	
SENSUSRADIO_EncryptionKey	UInt128	
SENSUSRADIO_TfxSecondaryChannelInfo_1	UInt48	
SENSUSRADIO_TfxSecondaryChannelInfo_2	UInt48	

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

Name	Data Type	Description
SENSUSRADIO_CustomerText	UInt72	
SENSUSRADIO_Tfx_Structure	UInt88	
GENESISFLOW_DisplayPow10	SInt8	
METROLOGYASST_FlowPoint	Enum8	
CUSTOMER_SerialNumber	String	
SYSTEM_CustomerSerialNumber0	String	
SYSTEM_CustomerSerialNumber1	String	
SYSTEM_CustomerSerialNumber2	String	
SYSTEM_CustomerSerialNumber3	String	
TESTMANAGER_OutputFile	String	

Table 3.1 Un-Reversed Programming Parameters

3.9 ExecPasswordFile

Shall create and install password file and check with SYSTEM_ExitReason for App 15 (NFC) for correct password level 3, which is the skeleton key, installation.

3.10 CheckOrderNumber

Shall crosscheck the production-order-number between the Cordonel and the production-papers to verify the correct assignment.

3.11 CheckSerialNumber

Shall crosscheck the serial-number lasered on the housing (EMEA region) or labelled with a sticker to the register (NA region) with the internal Cordonel information to verify the correct assignment.

3.12 ExecFinalParametrization

The final parameterization shall prepare the Cordonel with the previously collected programming parameters [Get]. Special sequences for the start of programming, reset of accumulators and completion of programming have to be executed.

3.12.1 Prepare Programming

Before programming the Cordonel some registers shall be prepared being able to access some other registers.

Name	Value	Description
GENESISFLOW_SealDisplay	0	The display needs to be unsealed for change of unit and resolution

Table 3.2 Prepare Programming Parameters

3.12.2 Execute Final Parametrization

The final parametrization shall write all parameters form “VAKO” or “CSD” to the Cordonel.

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

3.12.3 Reset Accumulators

After the programming for the EOL the accumulator shall be reset to clear the accumulated volume during the calibration tests for a fresh usage to the customer and to reset the internal statistics.

Name	Value	Description
GENESISFLOW_ResetAccumulators	1	Writing 1 to this register clears the volume accumulator
NFC_EraseRMA	1	Writing 1 to this register erases the RMA area
CUSTOMER_RebootCount	0	Set counter to 0 as this is a value how often the customer app detected a reset

Table 3.3 Reset Accumulators Parameters

3.12.4 Finalize Programming

To finalize the programming the software shall set all needed registers to customer operation.

Name	Value	Description	Region
LOGGER_TriggerLogFlush	1	Writing 1 will force a log flush	All
GENESISFLOW_TriggerIdle	0	Trigger idle 0 releases the display to normal operation	All
SENSUSRADIO_ResetCounter	0	Set counter to 0 as this is a value how often the radio detected a reset	EMEA
SENSUSRADIO_SystemState	0xFF	System state 0xFF will force a storage of all parameters and reset	EMEA
GENESISFLOW_SealDisplay	1	Seal the display to lock against changes	EMEA
PERIODICLOG_ResetCounter	0	Set counter to 0 as this is a value how often the radio detected a reset	EMEA

Table 3.4 Finalize Programming Parameters

3.13 ExecStoreConfiguration

Force applications to store all settings in non-volatile-memory (NVM)

3.14 ExecRebootCordonel

Shall execute a reboot including the reload of all parameters from NVM. After the reboot the internal clock and the manufacturing date shall be set. Finally, all reset counters shall be cleared.

Name	Description
SYSTEM_CalendarSeconds	Set the seconds since 01. January 2000 to correct the new UTC after reboot
IRDA_MfgDate	This is the manufacturing date of the Cordonel

Table 3.5 Set Cordonel Time

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

Reset all counters after reboot

Name	Value	Description	Region
CUSTOMER_RebootCount	0	Set counter to 0 as this is a value how often the customer app detected a reset	All
PERIODICLOG_ResetCounter	0	Set counter to 0 as this is a value how often the radio detected a reset	EMEA
SENSUSRADIO_ResetCounter	0	Set counter to 0 as this is a value how often the radio detected a reset	EMEA

Table 3.6 Clear Reset-Counters

3.15 CheckFinalParametrization

Shall compare the written with the read-back values.

3.15.1 Read Registers

Registers which cannot be read should be excluded from the trial to access them. Some registers which are identified causing problems to read those shall be filled to a list of unreadable registers to exclude those manually. This list should be kept as small as possible as it has to be modified manually.

Name	Description
CUSTOMER_InternalLoginTest	Does NOT respond
CUSTOMER_AlarmVisualAutoClearMask	Does NOT respond
SYSTEM_CRC	Needs application number written to it
SYSTEM_CRC32	Needs application number written to it
SYSTEM_ExitReason	Needs application number written to it
SYSTEM_DriveCapacity	Needs drive number written to it
GENESISFLOW_TriggerActive	Does NOT respond
SENSUSRADIO_CurrentLoopMax	Does NOT respond
SENSUSRADIO_CurrentLoopSource	Does NOT respond
SENSUSRADIO_CustomerText	Does NOT respond
SENSUSRADIO_TfxSecondaryChannelInfo_1	Does NOT respond
SENSUSRADIO_TfxSecondaryChannelInfo_2	Does NOT respond

Table 3.7 Excluded Registers from Reading

3.15.2 Compare Registers

All registers marked in the `configuration.json` with the “statictype = approximate” or “dynamic” or infrequentlyupdated” cannot be directly compared to the written value. Registers which are as “statictype = Approximate” shall be checked for a range with allowed deviation of +/- 3 digits from the originally written to the read back value. All others with those markers shall be excluded from comparison as this will fail.

Some registers which are identified causing problems to compare those shall be filled to a list of uncomparable registers to exclude those manually. This shall be kept as small as possible as those need to be programmed manually.

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

Name	Description
CUSTOMER_AlarmVisualMask	Fully hard coded, cannot be overwritten
CUSTOMER_AlarmEnableMask	Will be overwritten by SENSUSRADIO if set System state to 0xFF
CUSTOMER_AppArrangement	Is read only, makes no sense to compare anything
SENSUSRADIO_UpgFWVersion	After a FW-Update the FW version has changed

Table 3.8 Excluded Registers from Comparison

3.15.3 Check Parametrization Sequence

[SRS_TBF_EOL_PP_CHECK_FINAL_PARAM_001]

Shall check the sequence of final parametrization with writing all parameters, safe them to non-volatile memory, reboot the Cordonel and check if the parameters survived the reboot.

3.16 CheckRadio

Shall check the encryption key, address the correct device with the radio-address and measure the RSSI values for the EMEA region.

3.16.1 Preparation

All parameters have to be written to the Cordonel in advance as those are going to read out for this test.

Parameters to readout:

Name	Description
SENSUSRADIO_FrequencyIndicator	433 or 868 MHz
SENSUSRADIO_RadioAddress	Individual radio address
SENSUSRADIO_EncryptionKey	Individual encryption key

Table 3.9 Readout Cordonel Parameters

Name	Value	Description
SENSUSRADIO_IrDAModulePresent	1	Tell the radio that something is in the slot which will impact the radio transmission – The IrDA Head is attached

Table 3.10 Prepare RSSI Test

3.16.2 RSSI Test

The SIRT has to send a PAM to the Cordonel and receive the SEMI from the Cordonel and shall measure the signal strength (RSSI) of the Cordonel transmitter. The SEMI contains the measured signal strength of the SIRT.

3.16.3 Finalization

After successfully test or if the test was skipped, the radio has to be switched to a not transmitting and ready to wakeup mode being able to ship it via airplane and wake up the radio with flow detection or via a PAM sent by the SIRT.

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL

Name	Value	Description
SENSUSRADIO_SystemState	0xFF	Force radio to restart and initialize all to set system to shipping state with radio in standby

Table 3.11 Finalize Radio Check - Write

Name	Value	Description
SENSUSRADIO_SystemState	2	Radio is in customer mode, transmitter is switched off, radio is ready to wakeup on flow detection or via PAM from SIRT

Table 3.12 Finalize Radio Check – Read and Compare

3.17 CheckAttachments

Shall mark the manual check of production papers and attachments by the operator as executed.

3.18 VisualInspection

Shall mark the inspection of the Cordonel display by the operator as executed.

3.19 PrepareShipping

Shall check of all preceding production steps and ask the Sentinel to crosscheck the results.

4 Restrictions and Uncertainties

The EOL parametrization is relying on the database contents and the Web-APIs. Except the check of the valid data range defined in the interface-description-file (`configuration.json`), the EOL software cannot validate the correctness of the data source. The parameters will be passed as is from the database to the Cordonel.

5 Contacts

Name	Function	Mail
Stoyan Zlatev	SW Developer - Sentinel	stoyan.zlatev@xylem.com
Roland Drabesch	SW Developer	Roland.Drabesch@xylem.com
Thomas Wiedebusch	SW Developer - external	thomas.wiedebusch@t-wiedebusch.de
Nikolas Orelly	IT Department	nicolas.orelly@xylem.com

Table 5.1 Contacts

Copyright ©2025 by SENSUS GmbH
CONFIDENTIAL