Instruction

Bring-up/test HW development

Document No.:	INS14283
Version:	6
Description:	This document describes how to use Silicon Labs development tools to bringup and validate HW based on the Z-Wave 700 devices, ZGM130S and EFR32ZG14.
Written By:	OPP;AYURTTAS;JFR
Date:	2019-12-05
Reviewed By:	NTJ;LTHOMSEN;SCBROWNI
Restrictions:	

Approved by:

Date	CET	Initials	Name	Justification
2019-12-05	10:14:04	NTJ	Niels Johansen	

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
8	20191125	OPP	Figure 2 Figure 3 Figure 4 Figure 5 Figure 6	Functionality of PF0 and PF1 swapped in drawings to match the datasheet and pin-mapping.
7	20191105	OPP	All Added section 4.5.10	Fixed broken references in the text. Added description of how to read UUID from Z-Wave device.
6	20190312	OPP	Figure 1 Section 4.4 Section 4.5.2 Section 4.5.1 Section 4.5.7.1 Section 4.5.7.2	Added a product with matching circuit to Figure 1 Added section 4.4 about how to get help in RailTest Added frequency table reference and elaborated on frequency selection Moved Section 4.5.1 to create a more logical reading flow. Changed wording about how to set RF attenuator (must be set to 0dB) and how to build setup. Added weather conditions to possible sources of RF performance variations
5	20190212	OPP	Section 4, Added 4.5.9 Table 5	Added description of the RailTest Scripting feature. Corrected LBT parameters.
4	20181212	OPP	Section 3.1.4, 4.5.8	Added section about RSSI threshold adjustment
3	20181205	OPP	Section 4.5.3, 4.5.7.1, 4.6	Corrected Z-Wave device name, added scripting languages like Perl and Python to section and changed the wordings.
2	20181130	OPP	All, sections 4.5.6.1, 4.5.6.2 and 4.6 added	Updates to document due to review comments. All pages affected. Added new sections. Updated all drawings with 0 Ohms resistor and a GND symbol for landing pad #3. Changed description of Z-Wave device.
1	20181122	OPP	ALL	Initial version of document

Table of Contents

Table of Figures

Table of Tables

1 ABBREVIATIONS

Abbreviation	Explanation
BOM	Bill Of Materials
CLI	Command Line Interface
CW	Carrier Wave (RF single tone)
DUT	Device Under Test
LBT	Listen Before Talk
PA	Power amplifier
PCB	Printed Circuit Board
ppm	Parts Per Million, for GHz frequencies app. 1 kHz
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
UART	Universal Asynchrony Receiver and Transmitter (RS232)
UUID	Unique Unit IDentification
WSTK	Wireless Starter Kit

2 INTRODUCTION

2.1 Purpose

The purpose of this document is to describe how to use the RailTest software to perform RF bring-up tests and validate a design based on the Z-Wave 700 devices, the ZGM130S for end-devices and the EFR32ZG14 for gateway devices.

2.2 Audience and Prerequisites

The audience for this document is Z-Wave hardware developers and test engineers working with the development, bring-up test, and validation test of Z-Wave hardware using Z-Wave 700 devices.

It is assumed that the reader has a basic knowledge of RF metrics and Z-Wave as such.

The prerequisites are:

1. Access to Simplicity Studio
2. Access to RailTest binary for the correct Z-Wave 700 device
3. A PC with a UART terminal program
4. A Z-Wave development kit or hardware containing a Z-Wave 700 device with UART access
5. Access to RF measurement equipment, such as an RF Spectrum Analyzer and an RF generator

3 BRINGING UP Z-WAVE 700 HARDWARE IN GENERAL

When bringing up and validating a new Z-Wave 700 hardware platform, Silicon Labs provides a special software tool called RailTest.

RailTest is part of the Z-Wave software offering, and there are two versions:

1. A version for the ZGM130S End device
2. A version for the EFR32ZG14 Gateway device.

RailTest eliminates the need for creating a special “Hardware bring-up” Z-Wave application, because RailTest offers all the functionality needed to exercise the RF parts of the new Z-Wave 700 product under development.

Using RailTest, parameters such as:

- RF output power
- RF frequency
- Crystal fine tuning
- Z-Wave sensitivity

can easily be measured on the new Z-Wave 700 hardware which is being brought up / validated. Settings for output power, etc., can be fine-tuned and afterwards incorporated in the final Z-Wave application code, thus ensuring that the performance of the new Z-Wave 700 product will live up to all customer *and* RF regulatory demands.

3.1 Hardware Design Consideration for Easy Bring-Up and Validation

As a part of the general hardware design procedure for a new Z-Wave 700 product, one should take the following items into consideration:

1. How can the RF performance of the new product be tested?
2. How to interface the new product during the bring-up phase of the project?
3. How to provide means for the RF regulatory authorities to approve the new product?
4. How to fine-tune RF parameters to obtain the best possible performance of the new product?

If a solution for each of the above-mentioned items are implemented in the product at design-time, the effort of bringing up the new product will be significantly reduced.

3.1.1 How to Test the RF Performance of the New Product

All Z-Wave products contains an RF antenna, and many products incorporate the RF antenna as an integrated PCB antenna.

If the product has a PCB antenna, it is not easy to measure the output power of a Z-Wave 700 device. Such a measurement usually requires access to an anechoic chamber. However, even though it might be difficult to measure the output, this must be done to pass regulatory RF tests *and* to ensure that the settings of the output power matches the needs for the specific product.

At design time, the decision of adding an RF connection point for such a conducted measurement can be taken.

An RF connection point can be a non-populated footprint for an RF connector, which can be switched into the RF path for conducted measurements, or it can be a PCB landing pattern for a soldered attachment of an RF coax pig-tail. If the antenna structure requires an RF matching structure, the placement of the RF matching components can serve as an RF connection point as well. See the examples below:

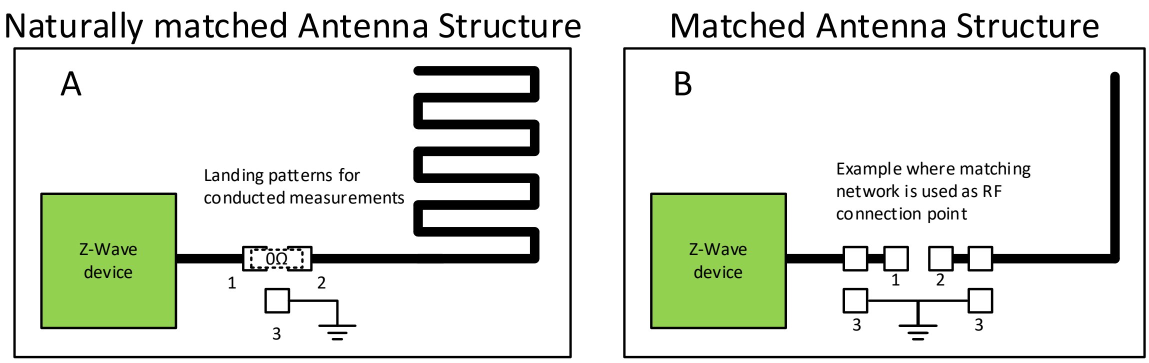


Figure 1. Landing Patterns for Conducted RF Measurements

Figure 1 shows two examples, a product with a naturally matched antenna, A, and a product requiring a match, B. The three landing patterns shown on Product A can be used to:

1. Measure the output power of the Z-Wave device
2. Measure the RF properties of the antenna structure of the device

For normal usage, the landing patterns marked “1” and “2” on Figure 1 A are shorted by a 0 ohms resistor. Landing pattern “3” is connected to the RF ground plane of the PCB.

For product A: During bring-up of the product or when the RF regulatory authorities are to measure conducted RF power, the 0 ohms resistor is removed, and an RF pig-tail can be soldered to the landing patterns “1” and “3”. The inner-core of the RF pig-tail must be connected to landing pattern “1” and the grounding shield of the RF pig-tail must be connected to landing pattern “3”. The RF pig-tail can now be connected to an RF spectrum analyzer, and the output power transmitted by the Z-Wave device can be measured with a minimal insertion loss and without the need for any radiated RF power measurements.

When the RF properties of the antenna structure are to be measured, this too can be done with the usage of the landing patterns and an RF pig-tail. In this case, the 0 ohms resistor connecting landing pattern “1” and “2” must be removed, and the inner-core of the RF pig-tail must be connected to landing pattern “2” and the grounding shield of the RF pig-tail must be connected to the landing pattern “3”.

Note: When measuring the parameters of the antenna structure using a network analyzer, please be aware of the effect of the cable used to conduct the measurement.

For product A, the cost for this landing-pattern connection method will be the PCB real-estate for the three landing patterns and the BOM cost of a 0 ohms resistor.

If the antenna requires an RF matching structure, such as a T or Pi configuration, please see product B in Figure 1. Here, the landing patterns for the matching components may serve as connections points as well. In this case, the landing pattern “1” can be used to measure the direct output power from the Z-Wave device, where as landing pattern “2” can be used to measure the RF characteristics of the antenna structure. Again, landing patterns 3 are RF ground connections.

The benefit of this hardware design strategy, where the possibility for an RF connection point is contained in the final product design is:

1. Shorter bring-up time
    - Effortless connection of the RF pig-tail without destroying or cutting in the product
2. More reliable and predictable measurements
    - With a firmly attached RF pig-tail, measurements can be repeated and are not depending on reflections or environmental changes as with radiated measurements
        - Be aware of the cable effect for antenna measurements
3. Better possibilities to adjust the Z-Wave device for best possible performance, which creates a better product.
    - With a well-tuned RF product, most of the RF energy transmitted by the Z-Wave device will be radiated by the antenna-structure, hence the customers will experience the best possible RF range between Z-Wave products.

Note: If a non-populated RF connector is used instead of the landing pattern strategy, an even more robust and easy-to-use RF connection point can be created during the bring-up and validation phase. Then an RF connector is soldered onto the product when needed, and once finished with the prototypes, the RF connector is removed from the BOM.

3.1.2 How to Interface the New Product During the Bring-Up Phase of the Project

Since Silicon Labs strongly recommends using the software RailTest during the bring-up and validation phase of a new product, interfacing to the Z-Wave 700 device is as important as being able to perform RF measurements on the new product.

RailTest is a stand-alone software package, available as a binary, and is *not* a part of any Z-Wave application. To use RailTest, one must be able to flash the Z-Wave 700 device with the software during the development phase.

If the new product *does not* contain a host processor, a programming interface to the Z-Wave 700 device will be an integral part of the product, and it will not be a problem to flash RailTest to the Z-Wave 700 device using the main programming interface of the product:

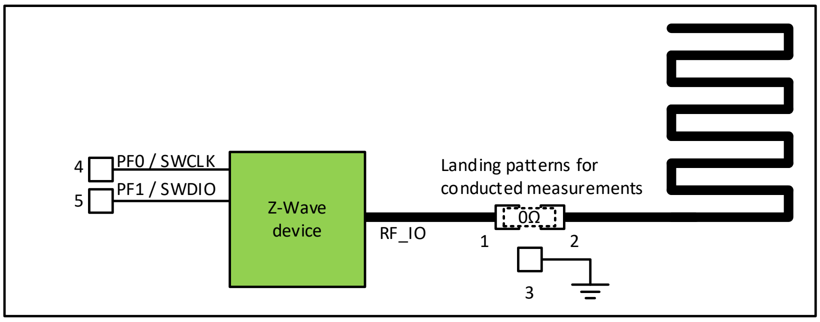
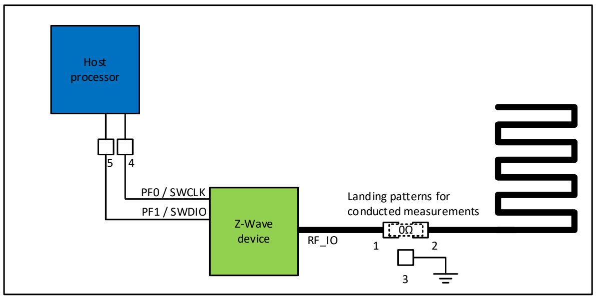


Figure 2. Programming Interface Landing Patterns for a Z-Wave 700 Device

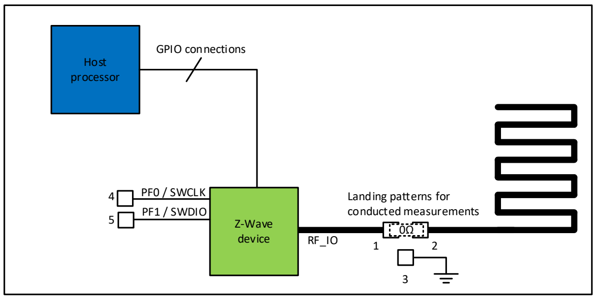
The programming pins of any Z-Wave 700 device are located at the ports PF0 and PF1, and the interface for these ports, along with supply and ground connections, etc., can be realized as a PCB connector, test points on the PCB, or landings patterns on the PCB. The method is selected depending on the requirements of the BOM cost and on how to assemble and program the Z-Wave product during the manufacturing flow. The main point is that there must be a physical programming interface available in the product.

If the new product *does* contain a host processor, there are two possibilities for programming the Z-Wave 700 device:

1. The host processor can download code to the Z-Wave 700 device
2. The host processor cannot download code to the Z-Wave 700 device



Host processor programming Z-Wave device



Host processor cannot program the Z-Wave device

Figure 3. Host Processor Application

For situation #1, where the host processor can program the Z-Wave device, one must ensure that the host processor is also able to handle and download the RailTest during the bring-up and validation phase. If this is not the case, implementing a direct programming interface to the Z-Wave device, e.g., at a set of landing patterns, is strongly advised.

For situation #2, where the host processor is not able to program the Z-Wave device, programming the Z-Wave device must be handled as if the product did not contain any host-processor at all. In this case, a connector, landing patterns, or test points must be designed in to the product for programming purposes.

Apart from the ability to download RailTest software to the Z-Wave 700 device, one must also be able to use and control the Command Line Interface, CLI, of RailTest, and this requires access to the UART pins of the Z-Wave 700 device.

For products *without* a host processor, the interface to the UART pins of the Z-Wave 700 device can be a connector, a set of landing patterns, or a set of test points on the product. See the example in Figure 4.

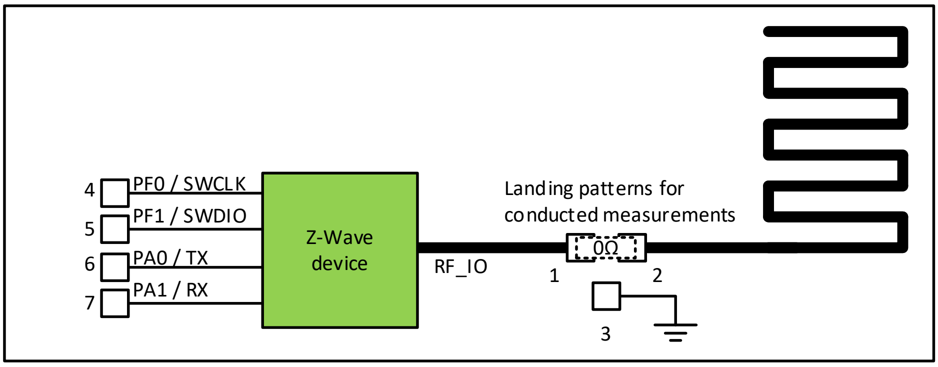
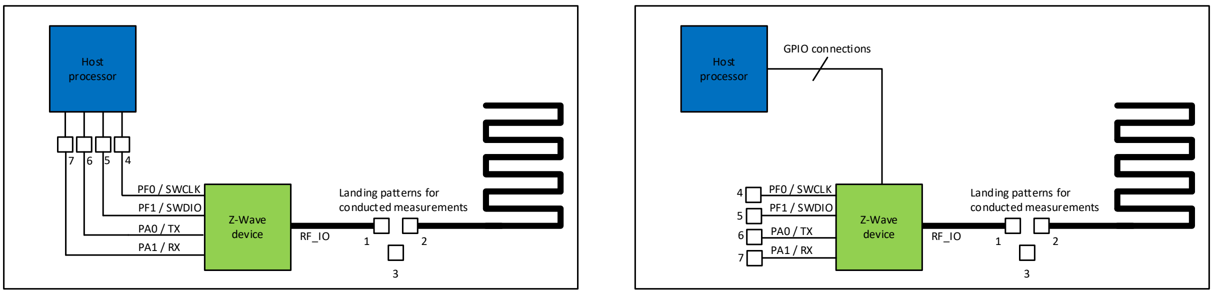


Figure 4. UART Connections for Z-Wave Product Without a Host Processor

During the bring-up and validation phase, wires can be temporarily connected to the UART interface and RailTest can be configured and controlled.

For products *with* a host processor, two possibilities exist for controlling the RailTest software through the UART interface of the Z-Wave device:

1. The host processor provides a terminal interface
2. The Z-Wave device can be controlled directly through its UART interface



Host processor communicates with Z-Wave device using UART

Host processor does not communicate with the Z-Wave device using the UART

Figure 5. UART Connection to Z-Wave Device in Host Processor Products

For situation #1, where the host processor uses the UART lines to control the Z-Wave device, the host processor should be able to directly relay any commands to and from the Z-Wave device. This will enable the developer to control the RailTest software using the connection through the host processor. If this is not possible, a connection in the form of a connector, a set of landing patterns or test points, like shown in Figure 5, should be available on the product. During bring-up and validation, temporary connections to the UART connection points can make this be used to control the RailTest software.

For situation #2, where there are no UART connections between the host processor and the Z-Wave device, it is strongly advised to provide connection possibilities to the two UART connections of the Z-Wave device, e.g., the landing patterns 6 and 7 shown in Figure 5.

The cost of implementing a programming- and control-interface will depend on the method selected. If a connector is selected, and the connector is going to be permanently mounted in the product for no other reason than ensuring the interface needed for the bring-up phase, the BOM cost will the high. But, if the connector is not mounted in the finished product, or if the interface connections are realized as landing patterns or test points, the cost will only be the added real-estate of the PCB. If the programming interface and control interface are implemented as part of a host processor system, there will be no cost added at all.

The benefit of this hardware design strategy, where the needed interface connections are contained in the final product design is:

1. Shorter bring-up time
    - Effortless reprogramming and control of the product without cuts and straps
2. More reliable and predictable measurements
    - With a firmly attached programming and control interface, changes to settings can be repeated and are not dependent on dangling wires which easily break.
3. Better possibility for third-parties to interact with the product
    - Regulatory authorities can more easily handle the products during the required RF regulatory measurements since the product is not fragile.

As an example of products with added interface structures for usage during both bring-up, validation and production, refer to the radio boards from the Z-Wave development kits BRD4200, BRD4201, and BRD4202.

3.1.3 How to Provide Means for the RF Regulatory Authorities to Approve a New Product

Any product containing a radio transmitting and receiving device must be approved by the appropriate RF regulatory authorities, and this is also valid for products based on the Z-Wave 700 devices.

The following RF properties are measured during an RF regulatory measurement session:

1. Conducted RF measurements
    a. Performed on a CW for each Z-Wave frequency applicable for the product.
2. Radiated RF measurements
    a. Performed on a CW for each Z-Wave frequency applicable for the product.
3. Performance measurements, such as blocking performance measurements
    a. Performed on a simple Z-Wave system.

To obtain as efficient a measurement session at the RF regulatory authorities as possible, it is strongly advised to be well prepared:

1. Prepare a test product fitted with RF pig-tail/wired RF connection, RailTest downloaded to the product, and UART interface
    a. This product will be for the Conducted and Radiated CW measurements
2. Prepare a test product with the Z-Wave application downloaded to the product
    a. This product will be for the performance measurements
3. Prepare a controller / companion product for the product, which can enable the product for normal Z-Wave operation
    a. This product will be for the performance measurements

The set of pre-configured modules, the controller, and how to use and setup the device for the various tests should be described in a Standard Operating Procedure (SOP), which should be sent with the modules when they are shipped to the RF regulatory authorities.

The controller to use can be a Silicon Labs Z-Wave PC controller or it can be the controller intended for controlling the product. The most important factor is that the controller and how it is to be used in relationship to the RF regulatory measurements is described in such a way that a third party, i.e., the test personnel at the RF regulatory test lab, will be able to setup the needed test sequence.

3.1.4 How to Adjust RF Parameters to Get the Best Possible Performance of the Product

Regarding adjustments and what to adjust to obtain the best possible RF performance of the product, most settings are taken care of by the Z-Wave protocol. There are, however, three settings which cannot be pre-set by the Z-Wave protocol because the settings are dependent of the layout for the product. These settings are as follows:

1. PA setting
    - The output level for the PA
2. Crystal calibration
    - Fine-tuning of the RF and system frequency
3. Listen Before Talk (LBT) RSSI threshold
    - The RF noise level threshold above which Z-Wave transmission is not allowed. Only valid for certain Z-Wave regions.

Regarding the PA setting: For each Z-Wave region, the level of RF power allowed to emit is regulated by the regional RF regulatory authorities. Both RailTest and the Z-Wave protocol allows for setting up the PA to be able to transmit as much RF power as allowed and fulfill the requirements for the RF fundamental as well as the RF harmonics.

Setting up the transmitted RF output power level can be adjusted in RailTest, and once the correct level is found, the setting must be transferred to the settings used in the Z-Wave protocol.

Instructions for using RailTest for setting the PA output power is described in section 4 of this document.

Regarding the crystal calibration: Calibration/Fine adjustment of the crystal is only applicable to products based on the Z-Wave 700 Gateway-device, the EFR32ZG14. For products based on the Z-Wave 700 End-device, the ZGM130S, no fine adjustment is needed since the ZGM130S is calibrated at production time.

What is the purpose of a system frequency calibration / fine adjustment? The tolerances of all crystals are divided into three components: An initial tolerance, a temperature tolerance, and an aging tolerance. The initial tolerance is what can be removed by a calibration / fine adjustment of the product.

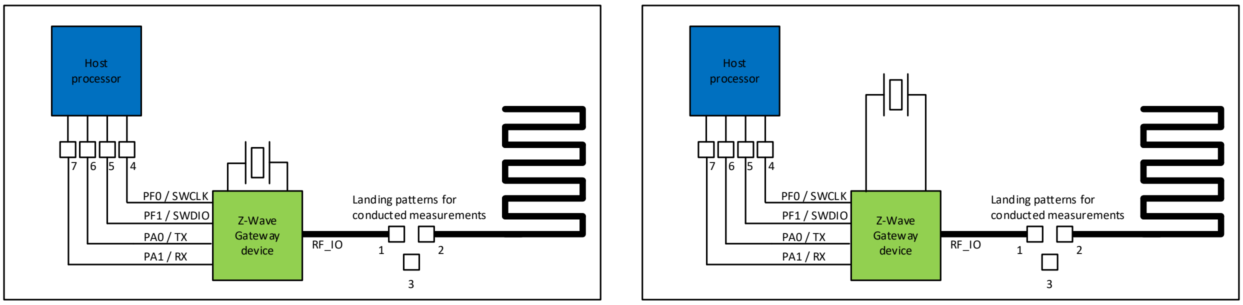
The general need for precise system frequency in a radio system arises from the fact that all receiver systems have filters in the receive path. To eliminate unwanted noise in the receiver path, the bandwidth of these receiver filters must be as narrow as possible. Therefore, a designer must make sure that the frequency variation of the transmitted signal matches the bandwidth of the receiver filters because if the transmitter varies too much, the receiver filters starts to attenuate the information carrying signal. So, if both the transmitter and the receiver are aligned on the same RF frequency / system frequency, an optimal receive chain can be realized where only the information from the transmitters signal passes through the receiver filters.

There are two methods which can be used to conduct a crystal fine-tuning / calibration:

1. Each module is individually frequency calibrated during the manufacturing of the product
2. The product is generally frequency adjusted

Approach #1 results in the best possible performance of the product because it will ensure that the individual crystals are adjusted to their nominal frequencies. The drawback is a costly process to perform an inline crystal calibration during the manufacturing of a product. Note that all ZGM130S devices are individually frequency calibrated during the device manufacturing process at Silicon Labs.

Approach #2 is a general frequency adjustment, which is highly recommended for the product when it is in the prototype phase. The general frequency adjustment counteracts for the parasitic load of the crystal component which varies from product layout to product layout. See the two examples below:



Short distance from system crystal to Z-Wave Gateway device

Long distance from system crystal to Z-Wave Gateway device

Figure 6. System Crystal Location on the Product

The product with the short distance between the system crystal and the Z-Wave Gateway device has a relative small parasitic load capacitance compared to the product with the longer distance between the crystal and the Z-Wave Gateway device.

Note: It is always recommended to keep the distance from the system crystal to the Z-Wave device as short as possible.

An internal load-capacitance can be adjusted through RailTest or through a Z-Wave API in the Z-Wave protocol. This load-capacitance adjustment should be performed when the general frequency trend of the new product can be measured:

1. At the prototype stage, a larger sample of products are configured through RailTest to transmit a CW at a known frequency.
2. The frequency for each product is measured and noted.
3. The average frequency offset is found.
4. RailTest is used to adjust the crystal-setting until the average frequency measured is at the target frequency.

The value of the adjustment is used in the product's Z-Wave application code, and hence the average system frequency of the product is adjusted according to the PCB of the product.

Refer to section 4 to read more about how to perform the crystal adjustment.

Regarding Listen Before Talk RSSI threshold: For certain Z-Wave regions, RF regulatory authorities require that a transmitting device performs a clear channel assessment before the device can transmit any RF energy. In Japan, the LBT limit is -80dBm and in Korea, the limit is -65dBm. This means that Z-Wave devices must be able to measure the amount of RF energy in the channel and, if the measured RF noise level is below -80dBm, the Z-Wave device may transmit a Z-Wave frame. The same applies for Z-Wave devices in Korea, only with the limit of RF noise at -65dBm.

To compensate for any RF loss in the RF path of a Z-Wave device, the LBT thresholds -80dBm or -65dBm cannot be used directly as threshold values in the Z-Wave protocol, since this would give a too pessimistic LBT threshold. For example, if the antenna gain of the product is -2dBi and the RF path-loss after the antenna is 2.5dB due to some matching components, the total loss in the RF path is 4.5dB. With a loss of 4.5dB, the LBT threshold should be set to (-80dBm + 4.5dB) = -75.5dBm for the Japanese LBT threshold or (-65dBm + 4.5dB) = -61.5dBm for the Korean LBT threshold.

Note: Since LBT is part of the RF regulatory tests performed in certain Z-Wave regions, it is important that the LBT thresholds are correctly setup for the Z-Wave product.

Information on using the RailTest for finding the LBT threshold is described in section 4.5.8 of this document.

4 USING RAILTEST

As mentioned in section 3, RailTest is a software tool distributed as part of the Z-Wave software offering, through which the user can setup various parameters relevant for both RF regulatory testing and for hardware bring-up and validation tests:

1. PA output power setting
2. Crystal fine-adjustment
3. Transmission of CW
4. Transmission of Z-Wave frames
5. Reception of Z-Wave frames
6. Finding the LBT threshold
7. Scripting of RailTest

When using RailTest, a special Z-Wave application is not required for the hardware developer to bring-up a new product and validate it.

The usage of RailTest thus decouples the software development phase from the hardware bring-up phase.

4.1 Downloading RailTest

RailTest is part of the Z-Wave software distribution; it is a pre-compiled binary file, ready to be downloaded to the product using a tool such as Simplicity Commander (refer to UG162: Simplicity Commander Reference Guide).

There are two versions of RailTest:

1. RailTest for ZGM130S End-devices
2. Railtest for EFR32ZG14 Gateway-devices

The correct RailTest version for the correct Z-Wave 700 device must be used in the new product.

4.2 Interfacing to RailTest

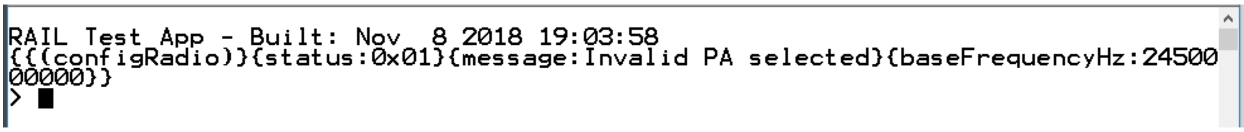
As described in section 3.1.2, RailTest is using UART lines for communication, which means that a computer with a terminal program is needed to be able to control and use RailTest.

The serial settings must be:

1. Baudrate: 115.2kBaud
2. No parity
3. 8 databits
4. 1 stopbit
5. No handshake

4.3 Starting RailTest

Once connected to the product, and after having downloaded RailTest and started the terminal program on the computer, RailTest outputs information on both the display of the BRD4001 WSTK board and on the UART :



```
RAIL Test App - Built: Nov  8 2018 19:03:58
{{(configRadio)}{status:0x01}{message:Invalid PA selected}{baseFrequencyHz:24500
00000}}
>
```

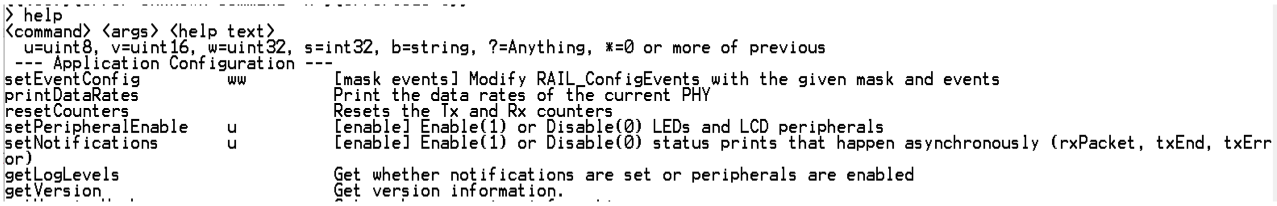
Figure 7, RailTest reset prompt

The program is now ready for input.

Initially, RailTest enables the device in receive mode at a non-Z-Wave frequency and setting. This setting needs to be changed.

4.4 Getting help in RailTest

RailTest has a build in Help Menu. The Help Menu is shown using the command ‘help’:



```
> help
<command> <args> <help text>
  u=uint8, v=uint16, w=uint32, s=int32, b=string, ?=Anything, *=0 or more of previous
 --- Application Configuration ---
setEventConfig        ww        [mask events] Modify RAIL_ConfigEvents with the given mask and events
printDataRates                  Print the data rates of the current PHY
resetCounters                   Resets the Tx and Rx counters
setPeripheralEnable   u         [enable] Enable(1) or Disable(0) LEDs and LCD peripherals
setNotifications      u         [enable] Enable(1) or Disable(0) status prints that happen asynchronously (rxPacket, txEnd, txErr
or)
getLogLevels                    Get whether notifications are set or peripherals are enabled
getVersion                      Get version information.
```

Figure 8, RailTest Help Menu

The Help Menu will show all the capabilities of RailTest. Many of these are not relevant for Z-Wave applications, and only the relevant ones are described in this document.

4.5 Setting up the Z-Wave environment for RailTest

Follow the steps below change to Z-Wave mode:

1. Disable the initial receiver mode
2. Setup Z-Wave mode
3. Setup Z-Wave packets
4. Setup Z-Wave region
5. Setup Z-Wave channel number

Once this is done, the user must decide if a transmission of Z-Wave frames, a CW transmission, or reception of Z-Wave frames is needed.

The following sections will show how to setup RailTest for various user-scenarios. All RailTest parameters will be written with the `Courier font` and enclosed with `''`. Comments are preceded with two semi-colons, such as `;; Comment`.

4.5.1 How to Adjust the PA Output Power Using RailTest

RailTest can be used to set the RF output level of the PA. The ability to adjust the PA output power is required to pass RF regulatory tests, where one of the key measurements is to ensure, that the level of RF power transmitted is at or below the required limits.

The RailTest command used to setup the output power is:

```
'SetPower yy raw'     ;; Set PA output power to setting number yy
```

Please refer to sections 4.5.2 and 4.5.3 for examples, where this command is used.

The number yy is a decimal number ranging from 0 to 155, where 0 is the lowest possible power setting and 155 is the highest possible power setting.

Note: Please notice, that the RF PA power cannot be adjusted unless the device is in idle mode, that is, the command 'rx 0' must have been issued and the device may not be in any kind of transmit mode.

4.5.2 How to Setup for CW Transmission, No SAW Filter/One SAW Filter

This section describes how to use RailTest for setting up a CW transmission and for adjusting the PA output power on products with one or no SAW filters:

```
'rx 0'                ;; Set device in idle state
'SetZwaveMode 1 3'    ;; Setup for Z-Wave operation
'SetTxLength 20'      ;; Setup TX frame length
'SetTxPayload 7 20'   ;; Setup payload type and length
'setzwaveregion x';; Setup Z-Wave region to number x, please see below
'SetPower yy raw'     ;; Set PA output power to setting number yy
'setchannel z'        ;; Set Z-Wave channel number to z
'settxtone 1'         ;; Enable the CW
```

The number x for Z-Wave regions can be a number ranging from 0 to 10 (refer to Table 1). The Z-Wave frequencies for each region can be found in [2].

Table 1. Z-Wave Region Numbers

Region Number	Region Location	Saw Filter Region
0	EU-European Union	E
1	US-United States	U
2	ANZ-Australia/New Zealand	H
3	HK-Hong Kong	H
4	MY-Malaysia	E
5	IN-India	E
6	JP-Japan	H
7	RU-Russia	E
8	IL-Israel	U
9	KR-Korea	H
10	CN-China	E

According to Table 1, the command ‘setzwaveregion 0’ would enable the device for usage in countries following the European RF legislation.

The output of the PA is regulated using the command ‘setPower yy raw’ and the value yy is a number ranging from 0 to 155. The higher a number, the more output power; the lower a number, the less output power.

The Z-Wave channel is setup using the command ‘setchannel z’, and the value z is a number ranging from 0 to 2. Channel and baud-rate are tied together as shown in Table 2:

Table 2. Channel Speed*

Channel Number	Channel Speed
0	100 kBit
1	40 kBit
2	9.6 kBit

*Channel speeds shown are not valid for JP and KR regions. In those regions, all channel speeds are 100 kBit.

The frequency of the CW is determined by 1) The Z-Wave region selected and 2) The Z-Wave channel selected within the Z-Wave region.

So, if transmission is to be tested on the European frequency 868.42 MHz, which is the 9.6 kBit channel, the channel number must be set to 2.

Below is an example where a device is initialized to transmit a CW at the Indian channel 2. The input commands to RailTest are marked with a blue line next to the command:

```
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:2014050208}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Enabled}}
> settxlength 20
{{(setTxLength)}{TxLength:20}}
> settxpayload 7 20
{{(setTxPayload)}{len:20}{payload: 0x0f 0x0e 0x11 0x22 0x33 0x44 0x55 0x14 0x77
0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee 0x10 0x11 0x12 0x13}}
> setpower 50 raw
{{(setPower)}{powerLevel:255}{power:200}}
> setzwaveregion 5
{{(setZWaveRegion)}{ZWaveRegion:IN-India}{ZWaveRegionIndex:5}}
> setchannel 2
{{(setchannel)}{channel:2}}
> settxtone 1
> {{(setTxTone)}{Tone:Enabled}{None:Disabled}{Time:2115683027}}
```

Figure 9. Example of setting up a CW for the Indian channel 2 using 50 as raw power

Note: The product will continue to transmit a CW until the command ‘settxtone 0’ is issued.

4.5.3 How to Setup for CW Transmission with a SAW Filter Bank

For products with multiple SAW filters, the correct SAW filter must be selected by setting the appropriate GPIO pins of the Z-Wave 700 device. Below is an example which can be used *if the product is configured like the BRD4200 or BRD4201 Z-Wave 700 development boards*. For these boards, PB14 and PB15 are used to select which SAW filter to use:

Table 3. SAW Filter Selection Settings for BRD4200 and BRD4201 Development Boards

SAW Filter Region	PB14	PB15
H	0	0
E	1	0
U	0	1

```
'rx 0'                  ;; Disable receiver mode
'SetZwaveMode 1 3'      ;; Setup for Z-Wave operation
'SetTxLength 20'        ;; Setup TX frame length
'SetTxPayload 7 20'     ;; Setup payload type and length
'setzwaveregion x'      ;; Setup Z-Wave region to number x
'SetPower yy raw'       ;; Set PA output power to setting number yy
'setchannel z'          ;; Set Z-Wave channel number to z
'setGPIOoutPin B 14 m';; Set state of PB14 GPIO, m =[0,1]
'setGPIOoutPin B 15 n';; set state of PB15 GPIO, n =[0,1]
'settxtone 1'           ;; Enable the CW
```

For a description of the number for the Z-Wave regions, refer to Table 1.

For a description of the Z-Wave channels, refer to Table 2

Below is an example where the product has a SAW filter bank like the one used on the BRD4200 / BRD4201 boards and the CW must be set to the US channel 0. The input commands are marked with glue line next to the commad:

```
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:3347594483}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Enabled}}
> setzwaveregion 1
{{(setZWaveRegion)}{ZWaveRegion:US-United States}{ZWaveRegionIndex:1}}
> setchannel 0
{{(setchannel)}{channel:0}}
> settxpayload 7 20
{{(setTxPayload)}{len:20}{payload: 0x0f 0x0e 0x11 0x22 0x33 0x44 0x55 0x14 0x77 0x88 0x99 0xaa 0xbb 0xcc 0
xdd 0xee 0x10 0x11 0x12 0x13}}
> settxlength 20
{{(setTxLength)}{TxLength:20}}
> setgpiooutpin b 14 0
{{(setGpioOutPin)}{Status:Success.}}

> setgpiooutpin b 15 1
{{(setGpioOutPin)}{Status:Success.}}
> settxtone 1
> {{(setTxTone)}{Tone:Enabled}{None:Disabled}{Time:3443572378}}
```

Figure 10. Example of Setting up a CW for the US Channel 0 and with U-SAW Filter Selected

If the PA power needs to be adjusted, this must be done prior to enabling the CW using the command 'setpower xx raw', where xx is a number ranging from 0 to 155.

Note: The product will continue to transmit a CW until the command 'settxtone 0' is issued.

Note: The selection of a SAW filter in a SAW filter bank depends on how the SAW filter bank is implemented in the product.

4.5.4 How to Setup for Transmitting Z-Wave Frames

RailTest can transmit Z-Wave frames formatted according to the Z-Wave specification. This can be very useful in connection with sensitivity measurements or when the RF regulator authorities measure the bandwidth of the Z-Wave radio channels.

Setting up for Z-Wave frame transmission is almost identical to setting up for CW transmission, except for the last command, which in this case is not 'settxtone 1' but instead 'tx x', where x is a number. If the number is 0, a continuous stream of Z-Wave frames will be transmitted until the command 'tx 0' is issued again. If the number is not 0, then the number reflects how many Z-Wave should be transmitted.

Below is an example where a product with only one or zero SAW filters is enabled for transmitting Z-Wave frames with a length of 20 bytes at the Z-Wave region 0 channel 0, which according to Table 1 and Table 2, is the EU region 100 kBit channel:

```
'rx 0'                 ;; disable RX mode
'setzwavemode 1 3'     ;; Setup Z-Wave mode
'setzwaveregion 0'     ;; Setup Z-Wave region 0, EU
'setchannel 0'         ;; Setup Z-Wave channel 0, 100 kBit channel
'settxpayload 7 20'    ;; Setup payload type and payload length
'settxlength 20'       ;; Setup payload length
'tx 0'                 ;; Start endless transmission of frames
```

For a description of the number for the Z-Wave regions, refer to Table 1.

For a description of the Z-Wave channels, refer to Table 2.

Below is an example of how RailTest responds to the commands described above. The input commands to RailTest are marked with a blue line to the right of the command:

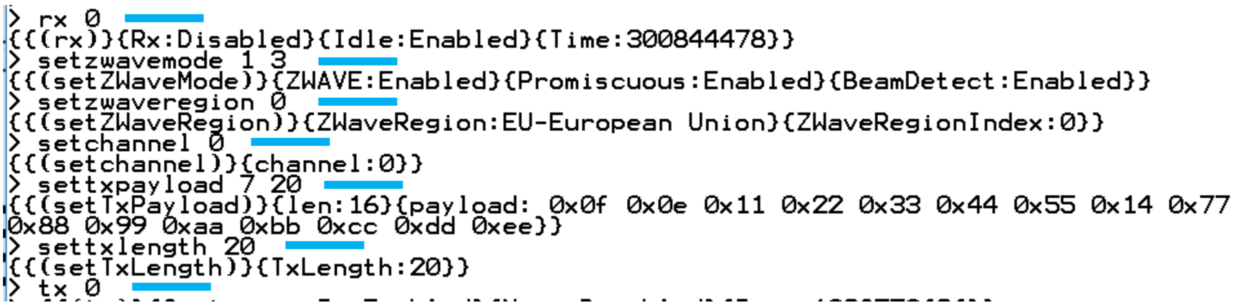


Figure 11. Setting up for Z-Wave frames transmission, region 0 channel 0

Note: To stop the transmission of Z-Wave frames, re-issue the command 'tx 0'.

If the product has multiple SAW filters, use the same method as described in section 4.5.3. Use the commands `'setGPIOoutPin B 14 m'` and `'setGPIOoutPin B 15 n'` according to the Table 3.

4.5.5 How to Setup for Reception of Z-Wave Frames

RailTest can receive and decode Z-Wave frames according to the Z-Wave specification. This can be very useful in connection with sensitivity measurements.

When setting up for Z-Wave reception, the setup is a bit more complicated compared to the transmission setup.

Below is an example where a product with only one or zero SAW filters is enabled for receiving Z-Wave frames with a length of 60 bytes at the Z-Wave region 0, which according to Table 1 is the EU region. Note that setting up a channel is not needed because the Z-Wave 700 device will scan all three Z-Wave channels for valid Z-Wave traffic:

```
'rx 0'                ;; Disable RX
'SetZwaveMode 1 3'    ;; Setup for Z-Wave mode
'SetTxLength 60'      ;; Setup Z-Wave frame length
'SetTxPayload 7 60'   ;; Setup Z-Wave frame length and type
'setzwaveregion 0'    ;; Setup Z-Wave region
'configRXChannelHopping x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12' ;; Ch setup
'enableRxChannelHopping 1' ;; Enable multi scanning mode
'rx 1'                ;; Enable reception of Z-Wave frames
```

The values for the demodulator settings x_1 to x_{12} must be setup according to Table 4:

Table 4. Multi-Channel Settings

Region	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
0	0	3	270	0	1	3	450	0	2	3	560	0
1	0	3	270	0	1	3	450	0	2	3	560	0
2	0	3	270	0	1	3	450	0	2	3	560	0
3	0	3	270	0	1	3	450	0	2	3	560	0
4	0	3	270	0	1	3	450	0	2	3	560	0
5	0	3	270	0	1	3	450	0	2	3	560	0
6	0	3	270	0	1	3	270	0	2	3	270	0
7	0	3	270	0	1	3	450	0	2	3	560	0
8	0	3	270	0	1	3	450	0	2	3	560	0
9	0	3	270	0	1	3	270	0	2	3	270	0
10	0	3	270	0	1	3	450	0	2	3	560	0

Below is an example of how RailTest responds to the commands described above. The RailTest commands are marked with a blue line:

```
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:2598745711}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Enabled}}
> settxlength 60
{{(setTxLength)}{TxLength:60}}
> settxpayload 7 60
{{(setTxPayload)}{len:60}{payload: 0x0f 0x0e 0x11 0x22 0x33 0x44 0x55 0x3c 0x77 0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee 0x1
0 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1a 0x1b 0x1c 0x1d 0x1e 0x1f 0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0
x28 0x29 0x2a 0x2b 0x2c 0x2d 0x2e 0x2f 0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3a 0x3b}}
> setzwaveregion 0
{{(setZWaveRegion)}{ZWaveRegion:EU-European Union}{ZWaveRegionIndex:0}}
> configrxchannelhopping 0 3 270 0 1 3 450 0 2 3 560 0
{{(configRxChannelHopping)}{numberOfChannels:3}{buffer:0x20005b14}{Success:True}}
> enablerxchannelhopping 1
{{(enableRxChannelHopping)}{Success:True}}
> rx 1
{{(rx)}{Rx:Enabled}{Idle:Disabled}{Time:2701630361}}
```

Figure 12. Setting Up RailTest for Z-Wave Reception in the Z-Wave Region 0, EU Region

The parameters for the channel hopping settings are found in Table 4. It is very important to use the correct settings for the selected Z-Wave region.

Once the receiver is setup and enabled, the product will start to receive any Z-Wave frame correctly formatted according to the Z-Wave specification. An example of received Z-Wave frames due to the setup described are shown in Figure 13:

```
> rx 1
{{(rx)}{Rx:Enabled}{Idle:Disabled}{Time:2701630361}}
> {{(rxPacket)}{len:11}{timeUs:2766887753}{crc:Pass}{rssi:-100}{lqi:92}{phy:0}{isAck:False}{syncWordId:0}{antenna:0}{c
hannelHopIdx:1}{payload: 0xd0 0x11 0x4e 0x34 0x02 0x41 0x06 0x0c 0x06 0x98 0x40}}
{{(rxPacket)}{len:9}{timeUs:2802766603}{crc:Pass}{rssi:-97}{lqi:96}{phy:0}{isAck:False}{syncWordId:0}{antenna:0}{chann
elHopIdx:1}{payload: 0xd0 0x11 0x4e 0x34 0x02 0x03 0x09 0x0a 0x06}}
{{(rxPacket)}{len:19}{timeUs:2814689603}{crc:Pass}{rssi:-100}{lqi:114}{phy:0}{isAck:False}{syncWordId:0}{antenna:0}{ch
annelHopIdx:1}{payload: 0xd0 0x11 0x4e 0x34 0x02 0x41 0x05 0x14 0x06 0x98 0x80 0x3d 0xbb 0x49 0xcd 0x2a 0xca 0xc8 0x17
}}
```

Figure 13. Example of Received Z-Wave Frames

Figure 13 shows how RailTest presents the reception of three different Z-Wave frames. Right after the enabling of the receiver, the command 'rx 1', a Z-Wave frame with the length of 11 bytes is received. A time stamp is given, the CRC value of the Z-Wave frame is validated, the incoming RSSI level is stated and then the payload of the frame is written. Next, a frame of 9 bytes is received, followed by a frame with the length of 19.

So even though a Z-Wave frame length of 60 was setup, all frames are received, validated, and decoded.

If the product has multiple SAW filters, and a SAW filter must be selected, use the same method as described in section 4.5.3 and use the commands `'setGPIOoutPin B 14 m'` and `'setGPIOoutPin B 15 n'` according to the Table 3.

4.5.6 How to Fine-Tune the System Crystal Using RailTest

4.5.6.1 For Z-Wave 700 Gateway devices

As mentioned in section 3.1.4, an average adjustment of the RF system frequency is strongly recommended for Z-Wave 700 Gateway devices to compensate for the parasitic load capacitance added by the products PCB to the system crystal.

The adjustment procedure is a two-step process:

1. Establish the average RF frequency offset
2. Adjust and remeasure the RF frequency offset

For item #1, generating an RF CW is needed. How to generate a CW using RailTest is described in either section 4.5.2 or 4.5.3.

Once the product is transmitting a CW, the frequency of the CW must be measured with a precise and newly calibrated spectrum analyzer. The frequency span of the spectrum analyzer should be set to no more than 100kHz with a low resolution bandwidth in order to be able to measure the small frequency changes introduced by the crystal frequency adjustments.

The frequency measured on the products must be recorded for later comparison purposes.

To get a general idea about the frequency offset, it is strongly recommended to perform this measurement on several boards.

Once the frequencies of the boards are recorded, the adjustment process can start. Assuming the TxTune being turned on, please use below RailTest commands:

```
'setTxTone 0'          ;; Disable the TxTone
'rx 0'                 ;; Set the system in idle
'getCtune'             ;; Read the current CTune value
'setCtune 0xYYY'       ;; Set a new CTune value
'setTxTone 1'          ;; Re-enable the TxTone again
```

The command 'getctune' is used to find the initial setting for the product. This number will be given as a hexadecimal number, hence the notation 0xYYY.

An iterative process now starts. It is strongly advised to only change the value of CTune in small increments, e.g. change the CTune value 2 digits up or down. With the new CTune setting 0xYYY, a new frequency measurement should be performed to verify if the CW frequency is adjusted to the target frequency.

An example of a frequency adjustment cycle is shown below. In the example, the target frequency is 908.42 MHz, since the Z-Wave region is selected to '1' and the channel number is set to 2 (for reference, please see Table 1 and Table 2). The product does not have a switchable SAW filter bank. The RailTest commands are marked with a blue line to the right of the command:

```
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:40695686}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Ena
bled}}
> setzwaveregion 1
{{(setZWaveRegion)}{ZWaveRegion:US-United States}{ZWaveRegionIndex:1
}}
> setchannel 2
{{(setchannel)}{channel:2}}
> settxtone 1
> {{(setTxTone)}{Tone:Enabled}{None:Disabled}{Time:72449781}}
settxtone 0
> {{(setTxTone)}{None:Enabled}{Tone:Disabled}{Time:76889486}}
getctune
{{(getCtune)}{CTUNE:0x155}}
> setctune 0x154
{{(setCtune)}{CTUNE:0x154}}
> settxtone 1
> {{(setTxTone)}{Tone:Enabled}{None:Disabled}{Time:317881568}}
```

Figure 14. System Frequency Adjustment Procedure

If the CW is adjusted to be within 1ppm, no further adjustments are needed.

Once a CTune value is found, this value should be tested on the other prototype products to ensure, that the average RF frequency offset is removed.

Note: Due to the inherent tolerances of crystals, only a "pr. Product" fine-adjustment/calibration will eliminate the frequency offset for each individual product.

4.5.6.2 For Z-Wave 700 End Devices

Since the ZGM130S devices are frequency calibrated during the manufacturing process, no frequency adjustment is needed to obtain a precise RF frequency. However, RailTest does not automatically use the CTune value found during the calibration. If there is a need to fine-tune a ZGM130S based product during the usage of RailTest, please follow the procedure shown below:

```
'rx 0'                ;; Set device in idle
'getmemw 0xFE081B8'   ;; Read crystal calibration value from ZGM130S
'setctune 0xZZZ'      ;; write the calibration value to RailTest
```

The value 0xZZZ depends on the value return by the memory read operation 'getmemw'.

The CTune value 0xZZZ is equal to a binary AND operation of the returned memory content :

CTune = Returned result of `'getmemw 0xFE081B8' & 0x1FF`

Example:

The 'getmemw' returns the hexadecimal value 0xffffff38. The CTune value to write is:

CTune = 0xffffff38 & 0x1ff => Ctune = 0x138

An example of how to read back the calibration value for the system crystal in RailTest is shown below. The RailTest commands are marked with a blue line to the right of the command:

```
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:14266570}}
> getmemw 0xFE081B8
#{{(getmemw)}{address}{value}}
{{0x0fe081b8}{0xffffff38}}
> setctune 0x138
{{(setCtune)}{CTUNE:0x138}}
>
```

Figure 15. Read Back of Crystal Calibration Value for ZGM130S Device

4.5.7 How to Perform a Sensitivity Measurement Using RailTest

A sensitivity measurement is the most vital measurement which can be performed on a wireless product. It shows the RF qualities of the product, and for wireless enabled products, the RF qualities will define the overall customer experience of the product:

1. The better sensitivity, the better a range between nodes in a Z-Wave network
    - Better coverage with fewer devices
2. Better range means fewer repeater devices in the Z-Wave network
    - Less cost for the customer
    - Lower network latency
3. With better sensitivity, frames do not need to be re-transmitted and hence less power is needed.
    - For sensor products, limiting the number of re-transmissions conserves the battery

There are two ways of conducting sensitivity measurements:

1. Wired
2. Wireless

4.5.7.1 Wired Sensitivity Measurements

For a wired sensitivity measurement, the DUT is connected to a Z-Wave frame transmitter through coax wires, and an adjustable RF attenuator is inserted in between the Z-Wave frame transmitter and the DUT. The Z-Wave frame generator could be a BRD4200, a BRD4201 or a BRD4202 Z-Wave 700 development board mounted on a BRD4001 WSTK and flashed with RailTest:

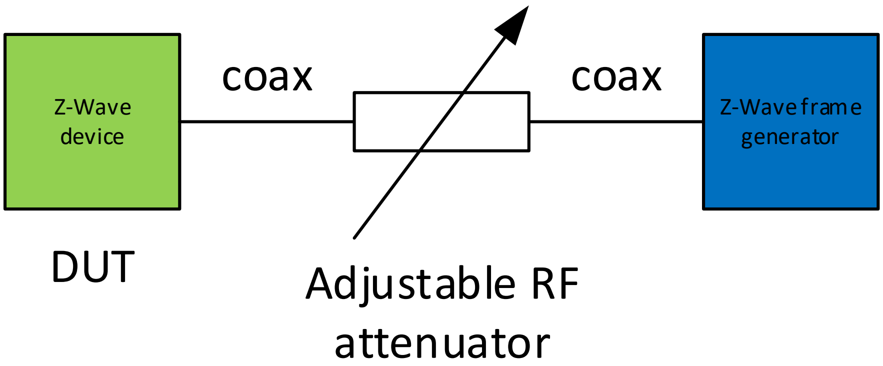


Figure 16. Simplified Wired Sensitivity Setup

For each part of the setup shown in Figure 16, a control unit is required, e.g. for the Z-Wave device, a computer must be connected to the product in order to configure RailTest; For the adjustable RF attenuator, either a test operator must adjust the RF attenuation or an automatic RF attenuator can be used, and for the Z-Wave Frame generator, control unit is needed as well.

Note: To avoid cross-hearing between the Z-Wave device and the Z-Wave frame generator it is recommended to lower the output power of the Z-Wave transmitters to e.g. -10dBm or even lower. An alternative is to place the Z-Wave device in a radio-shielded box.

Once the setup is created, one could perform the test as described below:

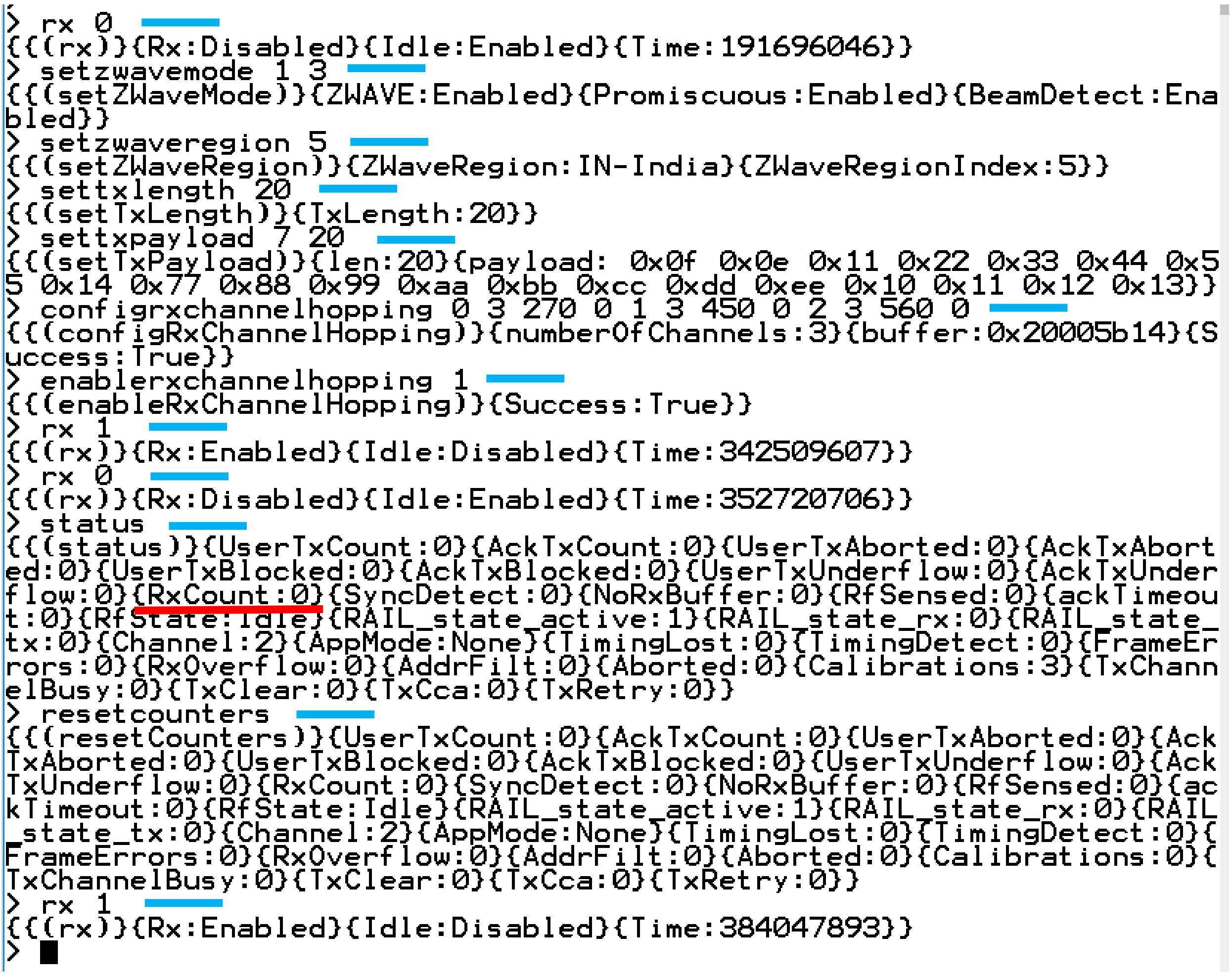
1. Setup both transmitter and receiver to the correct Z-Wave settings, as described in sections 4.5.4 and 4.5.5.
2. Set the RF attenuator to setting X, initially e.g. -50dB
3. Enable the DUT for reception
4. Transmit 1000 frames
5. Use RailTest to analyze how many of the 1000 frames were correctly received by the DUT
6. Reset the receive counters in the DUT
7. Decrease the RF attenuator value with 1 dB, to e.g. -51dB
8. Jump to #3 until less than 990 frames are received by the DUT

Using above described test method will test for a 1% packet error level.

The method to setup a device for Z-Wave frame reception is shown in section 4.5.5. Apart from this setup, RailTest's 'status' function can be used to evaluate the number of correctly received frames. The commands to use are:

```
'rx 0'              ;; Disable reception
'status'            ;; Get frame reception status
'resetCounters'     ;; Reset frame status counters
'rx 1'              ;; Enable reception for new attenuator setting
```

Below is an example, where the DUT is setup for reception of Z-Wave frames in the Indian region. The product is setup for reception, reception is enabled, status is asked for, the counters are cleared, and the device is re-enabled for another RF attenuator setting. Each RailTest command is marked with a blue line to the right of the command:



```
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:191696046}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Ena
bled}}
> setzwaveregion 5
{{(setZWaveRegion)}{ZWaveRegion:IN-India}{ZWaveRegionIndex:5}}
> settxlength 20
{{(setTxLength)}{TxLength:20}}
> settxpayload 7 20
{{(setTxPayload)}{len:20}{payload: 0x0f 0x0e 0x11 0x22 0x33 0x44 0x5
5 0x14 0x77 0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee 0x10 0x11 0x12 0x13}}
> configrxchannelhopping 0 3 270 0 1 3 450 0 2 3 560 0
{{(configRxChannelHopping)}{numberOfChannels:3}{buffer:0x20005b14}{S
uccess:True}}
> enablerxchannelhopping 1
{{(enableRxChannelHopping)}{Success:True}}
> rx 1
{{(rx)}{Rx:Enabled}{Idle:Disabled}{Time:342509607}}
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:352720706}}
> status
{{(status)}{UserTxCount:0}{AckTxCount:0}{UserTxAborted:0}{AckTxAbort
ed:0}{UserTxBlocked:0}{AckTxBlocked:0}{UserTxUnderflow:0}{AckTxUnder
flow:0}{RxCount:0}{SyncDetect:0}{NoRxBuffer:0}{RfSensed:0}{ackTimeou
t:0}{RfState:Idle}{RAIL_state_active:1}{RAIL_state_rx:0}{RAIL_state_
tx:0}{Channel:2}{AppMode:None}{TimingLost:0}{TimingDetect:0}{FrameEr
rors:0}{RxOverflow:0}{AddrFilt:0}{Aborted:0}{Calibrations:3}{TxChann
elBusy:0}{TxClear:0}{TxCca:0}{TxRetry:0}}
> resetcounters
{{(resetCounters)}{UserTxCount:0}{AckTxCount:0}{UserTxAborted:0}{Ack
TxAborted:0}{UserTxBlocked:0}{AckTxBlocked:0}{UserTxUnderflow:0}{Ack
TxUnderflow:0}{RxCount:0}{SyncDetect:0}{NoRxBuffer:0}{RfSensed:0}{ac
kTimeout:0}{RfState:Idle}{RAIL_state_active:1}{RAIL_state_rx:0}{RAIL
_state_tx:0}{Channel:2}{AppMode:None}{TimingLost:0}{TimingDetect:0}{
FrameErrors:0}{RxOverflow:0}{AddrFilt:0}{Aborted:0}{Calibrations:0}{
TxChannelBusy:0}{TxClear:0}{TxCca:0}{TxRetry:0}}
> rx 1
{{(rx)}{Rx:Enabled}{Idle:Disabled}{Time:384047893}}
>
```

Figure 17. RX Sensitivity Setup Method

Figure 17 shows the output of the status command, and there are many parameters being written by RailTest. The important number to notice is the {RxCount} number, underlined with red, in Figure 17.

If the transmitter has been started with the command 'tx 1000', 1000 frames are transmitted. Once the transmission is done, the receiver is disabled with the 'rx 0' command, the command 'status' is issued on the DUT, and if the reading of the RxCount value is 1000, like {RxCount : 1000}, all of the transmitted frames were received. If the reading is less, Z-Wave frames were lost due to a less optimal RF channel between the Z-Wave frame transmitter and the DUT.

Note: One does not need to setup any channel number in the receiver, since the receiver will scan all the RF channels for the Z-Wave region.

The method for setting up the transmitter for Z-Wave frame transmission is shown in section 4.5.4. To test all the Z-Wave channels within the Z-Wave region, it is important that each channel number, ranging from 0 to 2, is used in the transmitter setup and the sensitivity test. Below, all RailTest commands are marked with a blue line to the right of the command:

```
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:20544087}}
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:25205944}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Ena
bled}}
> setzwaveregion 5
{{(setZWaveRegion)}{ZWaveRegion:IN-India}{ZWaveRegionIndex:5}}
> settxpayload 7 20
{{(setTxPayload)}{len:16}{payload: 0x0f 0x0e 0x11 0x22 0x33 0x44 0x5
5 0x14 0x77 0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee}}
> settxlength 20
{{(setTxLength)}{TxLength:20}}
> setpower 24 raw
{{(setPower)}{powerLevel:24}{power:16}}
> setchannel 0
{{(setchannel)}{channel:0}}
> tx 100
> {{(tx)}{PacketTx:Enabled}{None:Disabled}{Time:113518115}}
{{(appMode)}{None:Enabled}{PacketTx:Disabled}{Time:138811980}}
{{(txEnd)}{txStatus:Complete}{transmitted:100}{lastTxTime:138781413}
{failed:0}{lastTxStatus:0x0}{isAck:False}}

> 
```

Figure 18. TX Sensitivity Setup, 100 Frames Transmitted for Region 5 Channel 0

When the command ‘tx xxx’ is issued, immediately the response from the RailTest will be what is shown in Figure 18 at the {{(tx)} *transmit information*}} information. Once the transmission of the 100 frames is done, the RailTest will show the line marked with the green line, starting with {{(txEnd)} *transmitted information*}} . When this line is observed on the UART output from the Z-Wave frame generator, the receiver is then ready for a status reading using the command ‘status’.

Note: It is most important to know the RF loss in the sensitivity measurement setup. This can be found by replacing the DUT shown in Figure 16 with a spectrum analyzer, setting the RF attenuator to 0 dB, and transmitting a CW from the Z-Wave frame transmitter. The RF power measured by the spectrum analyzer can be used to calculate the sensitivity as:

$$\text{Sensitivity} = P_{\text{measured at RF attenuator setting X}} - \text{Attenuator\_setting}_{\text{at 1\% packet error}}$$

Example:

> A spectrum analyzer is inserted instead of the DUT, the RF attenuator is adjusted to 0 dB attenuation, and the measured power at this RF attenuator setting is -20 dBm.
>
> The following sensitivity measurement shows that there is a 1% packet error level at an attenuator setting of 80 dB.

What is the sensitivity of the system?

Sensitivity = $-20\text{dBm}_{\text{measured at RF attenuator setting 0dB}} - 80\text{dB}_{\text{attenuator setting at 1\% packet error}}$

Sensitivity = -100dBm

If the 1% packet error level was measured at an RF attenuator setting = -82 dB, the sensitivity of the system is -102 dBm.

To avoid interference from the environment during the sensitivity measurements, it is strongly recommended to perform the measurements in a radio shielded environment such as an RF shielded room or by placing the receiver inside an RF shielded box. It is further advised to use a low PA setting for the Z-Wave frame transmitter to avoid any cross radiation directly from the PCB of the Z-Wave frame generator to the PCB of the DUT.

4.5.7.2 Radiated Sensitivity Measurements

A radiated sensitivity measurement is the most representative use-case measurement which can be performed because the DUT is used and tested just like the end-user of the product would experience it.

Reliable and trustworthy radiated measurements can only be conducted if:

1. The radiated measurement setup is performed under strictly controlled manners inside an anechoic chamber and a Total Isotropic Sensitivity test is conducted.
2. The radiated test is performed on a huge open field, high above the ground, and with no nearby constructions to reflect or transmit RF energy.

For normal urban areas in which a test would typically be conducted, the results from a radiated sensitivity measurement cannot be trusted or reproduced because:

1. RF channel will be strongly dependent on the environment
    - Buildings
    - Reflections from objects
    - Reflections from the ground
    - Weather conditions, if test is performed on an open field
2. RF channel will be strongly dependent on the exact positions of the DUT and the Z-Wave frame generator

Note: A radiated sensitivity measurement can give an *indication* of the line-of-sight range.

The RailTest setup method for the radiated sensitivity / range measurement is almost identical to the setup described in section 4.5.7.1. For the DUT, the method would be identical, but for the Z-Wave frame generator, decreasing the PA setting is not recommended. Instead, the planned PA setting must be used.

Figure 19 shows a simple radiated setup:

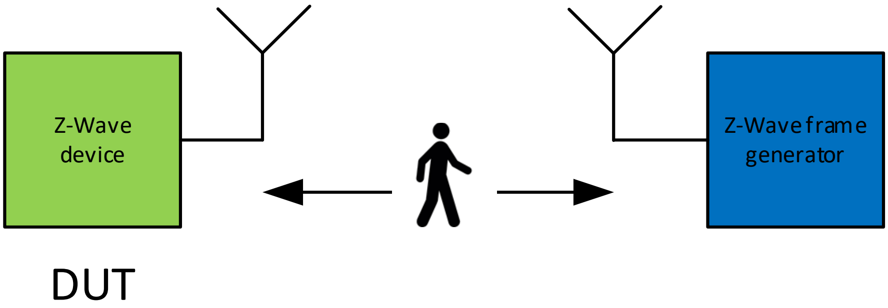


Figure 19. Radiated Sensitivity Measurement

Either the DUT or the Z-Wave frame generator is placed in a fixed location, and the distance between the DUT and the Z-Wave frame generator is gradually increased, with one set of frame transmission pr. distance-increment. Once the number of received frames has dropped by 1%, the pathloss due to the distance is equivalent to the sensitivity limit of the product.

The distance can also be increased until no frames are received at all. This would be the maximum line of sight distance that can be obtained with the DUT under the given conditions.

4.5.8 How to Find the LBT Threshold

This section describes how to use a RailTest for products with one or zero SAW filters when finding the LBT threshold for either the Japanese or the Korean Z-Wave region:

```
'rx 0'                    ;; Set device in idle state
'SetZwaveMode 1 3'        ;; Setup for Z-Wave operation
'SetTxLength 20'          ;; Setup TX frame length
'SetTxPayload 7 20'       ;; Setup payload type and length
'setzwaveregion x';; Setup Z-Wave region to number x, see Table 1
'SetPower yy raw'         ;; Set PA output power to setting number yy
'setchannel 0'            ;; Set Z-Wave channel number to 0
'setLbtParams L1 L2 L3 L4 L5 L6 L7' ;; Set LBT params and threshold
'setLbtMode csma'         ;; Set detection method
'configTxOptions 64'      ;; Setup transmitter options
'rx 1'                    ;; Enable reception and apply RF noise
'getRssi'                 ;; Read RSSI value for RF noise at threshold
'rx 0'                    ;; Set device in idle state
'setLbtParams L1 L2 L3 L4 L5 L6 L7' ;; Set LBT params and threshold
'rx 1'                    ;; Enable device in RX mode
'tx xx'                   ;; Transmit xx number of frames
```

The parameters for the command '`setLbtParams`' are shown in Table 5.

Table 5, Z-Wave LBT Settings

Region Number	L_1	L_2	L_3	L_4	L_5	L_6	L_7
6 (Japan)	2	2	15	$RSSI_{Japan}$	750	128	200000
9 (Korea)	2	2	15	$RSSI_{Korea}$	750	5000	200000

The purpose of this measurement is to find the number for the parameter L_4. This value will vary from Z-Wave region to Z-Wave region and from product design to product design.

Above, the device is initialized to transmit Z-Wave frames with a length of 20 bytes for a certain Z-Wave region. The PA output power is set to a value, yy, and the channel is selected to 0. For this measurement, it does not matter if the channel is set to 0, 1, or 2. The LBT parameters are then initialized, the collision avoidance is selected to type csma, and a TX option is set to 64.

Next the parameter L_4 must be found, and this is done by connecting the Z-Wave product to an RF generator using a wired connection, i.e., a coax cable, setting the output power of the RF generator to the RF regulation limit (for Japan -80 dBm and for Korea -65 dBm), and enabling an RF CW at the frequency of the channel. The Z-Wave device is now exposed to an amount of RF energy identical to the LBT noise level as required by the RF regulatory authorities.

With the CW applied to the Z-Wave product, the Z-Wave product is configured to be in receive mode, and an RSSI reading is performed using the RailTest command ‘getRssi’. The found RSSI value is identical to the LBT threshold L_4. For a conservative approach, it is advised to set the LBT threshold 1 dB below the found RSSI in order to have some margin in the threshold.

With L_4 found, the LBT parameters are re-initialized with the expected value for L_4 and a test can be performed as described below:

1. Turn off the RF output of the connected RF generator.
2. Transmit XX number of frames using the command ‘tx XX’.
3. Upon end of transmission, write the statistical values for the number of transmitted frames to the terminal. The number of frames transmitted must match the number of frames used in the command ‘tx XX’. For example, if the command ‘tx 2’ was used, the status must show that 2 frames were transmitted.
4. Enable the RF generator and set the RF output power to the level which matches the LBT threshold required.
5. Transmit a number of frames.
6. At the end of transmission, the number of frames transmitted should be 0 and the number of failed transmissions should be identical to the number of frames used in the command in item #5.
7. Adjust the output power of the RF generator up and down and check that the LBT threshold is set correct

Note that, for products with a SAW filter or a SAW filter bank, the procedure described is identical except for the fact that for SAW filter bank products, one must select the correct SAW as described in section 4.5.3.

Figure 20 shows an example where the LBT threshold is to be found for a device in the Z-Wave region 6, Japan. RailTest commands are underlined in blue and important results are underlined in orange. Information on enabling or disabling the RF generator used to create the noise in the channel are marked with a red box:

```
rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:149431364}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Enabled}}
> settxlength 20
{{(setTxLength)}{TxLength:20}}
> settxpayload 7 20
{{(setTxPayload)}{len:20}{payload: 0x0f 0x0e 0x11 0x22 0x33 0x44 0x55 0x14 0x77
0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee 0x10 0x11 0x12 0x13}}
> setzwaveregion 6
{{(setZWaveRegion)}{ZWaveRegion:JP-Japan}{ZWaveRegionIndex:6}}
> setpower 50 raw
{{(setPower)}{powerLevel:50}{power:80}}
> setchannel 0
{{(setchannel)}{channel:0}}
> setlbtparams 2 2 15 -63 750 128 200000
{{(setLbtParams)}{MinBo:2}{MaxBo:2}{Tries:15}{Threshold:-63}{Backoff:750}{Durati
on:128}{Timeout:200000}}
> setlbtmode csma
{{(setLbtMode)}{LbtMode:CSMA}}
> configtxoptions 64
{{(configTxOptions)}{waitForAck:False}{removeCrc:False}{syncWordId:0}{txAntenna:
Any}{altPreambleLen:False}{ccaPeakRssi:True}}
> rx 1
{{(rx)}{Rx:Enabled}{Idle:Disabled}{Time:247548184}}
> getrssi
{{(getRssi)}{rssi:-62.50}}
> rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:260241183}}
> setlbtparams 2 2 15 -63 750 128 200000
{{(setLbtParams)}{MinBo:2}{MaxBo:2}{Tries:15}{Threshold:-63}{Backoff:750}{Durati
on:128}{Timeout:200000}}
```

RF generator is enabled with RF noise equivalent to the LBT limit for Japan

```
> rx 1
{{(rx)}{Rx:Enabled}{Idle:Disabled}{Time:290778269}}
```

RF generator is disabled

```
tx 2
> {{(tx)}{PacketTx:Enabled}{None:Disabled}{Time:309234434}}
{{(appMode)}{None:Enabled}{PacketTx:Disabled}{Time:309502018}}
{{(txEnd)}{txStatus:Complete}{transmitted:2}{lastTxTime:309501970}{failed:0}{las
tTxStatus:0x0}{isAck:False}}
```

RF generator is enabled with RF noise equivalent to the LBT limit for Japan

```
> tx 2
> {{(tx)}{PacketTx:Enabled}{None:Disabled}{Time:351225
{{(txPacket)}{txStatus:Error}{errorReason:Tx channel busy}{errorCode:0xe0000000}
}
{{(appMode)}{None:Enabled}{PacketTx:Disabled}{Time:351504677}}
{{(txEnd)}{txStatus:Error}{transmitted:0}{lastTxTime:309501970}{failed:2}{lastTx
Status:0x20000000}{isAck:False}}
```

Figure 20. Example of Setting up a Device for Finding a LBT Threshold

The red boxes in Figure 20 indicate when to turn on and off the RF generator that creates the noise floor for the RSSI measurement and the following LBT measurement.

In the example, only two frames are transmitted and used to verify if the LBT threshold is correctly setup. It is strongly recommended to use a larger number of frames for this test to increase the statistical validity of the LBT threshold setting.

It may be an iterative process to find the exact LBT value. For good measurements and to add a bit of margin to the LBT threshold, it is advised to add a dB to the threshold value as a margin.

The value found for L_4 must be used in the Z-Wave protocol as described below:

- For Gateway devices, EFR32ZG14: LBT threshold value must be used in connection with FUNC_ID_ZW_SET_LISTEN_BEFORE_TALK_THRESHOLD
- For End devices, ZGM130S: LBT threshold value must be used in connection with the command EZWAVECOMMANDTYPE_ZW_SET_LBT_THRESHOLD

Note: It is very important to perform the LBT measurements with a calibrated setup, that is, all cable losses must be accounted for when setting the level of the RF generator to the LBT noise level.

4.5.9 Scripting RailTest

For some products, it may be impractical, or even impossible, to submit a product to the RF regulatory authorities if the only means of controlling the product is through a UART terminal interface, as described in Section 4.2:

1. The UART terminal interface may be impractical for the test house to handle.
2. The UART terminal interface may completely change the radiation pattern of the product and, thus, give a wrong impression of how the product performs.
3. The test house may not be trusted to correctly setup the device before the RF regulatory tests.

To accommodate for any of above situations, RailTest can be scripted in such a way that the Z-Wave 700 device stores the RailTest instructions in the internal flash memory and then retrieves *and* initializes the Z-Wave 700 device with the stored settings upon reset or re-power.

This means that it will be possible to setup, validate, and ship fully pre-configured products for RF regulatory tests without having the RF regulatory test house to do anything else but to apply power to the product under test.

This following section describes how to enable the script mode, enter a script, validate a script, and exit script mode.

Note: A permanent UART connection-means to the Z-Wave 700 device in the product is still strongly advised, please refer to 4.2, since the scripting of the Z-Wave device is done through the UART interface of the Z-Wave 700 device. But once setup, the UART connections used can be removed from the product to ensure, that the appearance of the product is as un-interrupted as possible.

4.5.9.1 Workflow for Script Mode

To be able to use RailTest in a scripted mode, one must:

1. Establish power, programming, and UART connections to the product as shown in section 3.1.2
2. Download RailTest to the product as described in section 4.1.
3. Interface to the UART of the Z-Wave 700 device as described in section 4.2.
4. Initialize scripting mode
5. Enter the script
6. Exit script mode
7. Re-power the product
8. Test the functionality of the script
9. Power off the product
10. Remove the UART and the programming connections to the Z-Wave 700 device

A script can at any time be changed or removed by re-connecting the Z-Wave 700 device of the product to its UART connections and then either delete or change the script.

4.5.9.2 How to Start, Exit and Erase a Script

RailTest can store commands in the flash memory and retrieve these commands upon a reset of a power-off/on event. This can be very useful in connection with pre-configuration of products which must be submitted to RF regulatory tests.

Below is an example where scripting is enabled and disabled:

```
'enterscript 1'        ;; Enable script mode, save in flash
'endscript'            ;; End script mode / clear script
```

The first command ‘enterscript 1’ enables scripting mode, and the ‘1’ means that the script is stored in the flash memory of the Z-Wave 700 device.

In the above example, the script is immediately terminated using the command ‘endscript’. When written as above, both script-mode is terminated *and* an eventual script stored will be erased from the flash memory of the Z-Wave 700 device.

Please notice that the first command written ‘enterscript 1’ must be followed by a “Return” character, where as the last command written ‘endscript’ activates once the letter ‘t’ is typed.

Figure 21 shows an example, where script mode is entered and immediately terminated again.

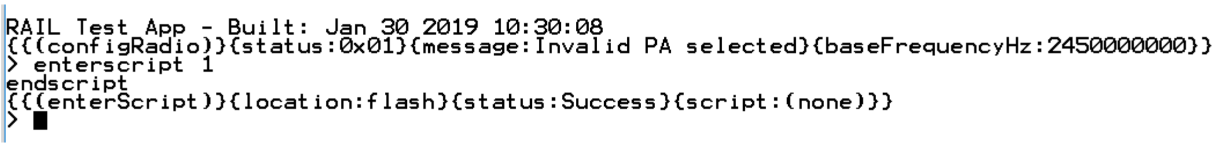


Figure 21. Enabling and Terminating Script Mode

Note: When the command ‘enterscript 1’ is directly followed by ‘endscript’, the content of the flash memory is erased and Script mode is disabled.

4.5.9.3 setTxTone Script

RailTest can handle the same commands in Scripting mode as it can through the interactive CLI previously described in the Sections 4.5 through 4.6. Finding and setting the LBT threshold is an interactive operation and is not well suited for Scripting mode.

Below is an example, where the Z-Wave 700 device is setup to transmit a CW tone:

```
‘enterscript 1’        ;; Enable script mode, store in flash
‘rx 0’                 ;; Disable RX
‘SetZwaveMode 1 3’     ;; Setup for Z-Wave operation
‘SetTxLength 20’       ;; Setup payload length
‘SetTxPayload 7 20’    ;; Setup payload type and length
‘setzwaveregion 6’     ;; Setup Z-Wave region, Japan
‘SetPower 50 raw’      ;; Setup output power
‘setchannel 0’         ;; Setup Z-Wave channel
‘settxtone 1’          ;; Enable CW
‘endscript’            ;; Store script in flash and end
```

When comparing the script above with the similar operation performed through the interactive CLI as described in section 4.5, please notice that the only difference between the commands are the ‘enterscript 1’ and the ‘endscript’.

Figure 22 shows an example, where the above script is typed into RailTest. RailTest commands are marked with blue.

```
> enterscript 1
rx 0
setzwavemode 1 3
settxlength 20
settxpayload 7 20
setzwaveregion 6
setpower 50 raw
setchannel 0
settxtone 1
endscript
{{(enterScript)}{location:flash}{status:Success}{script:rx 0
setzwavemode 1 3
settxlength 20
settxpayload 7 20
setzwaveregion 6
setpower 50 raw
setchannel 0
settxtone 1
}}
>
```

Figure 22. Scripting setTXTone

Please notice that when in Script mode, commands are *not* acknowledged by RailTest.

When the letter “t” in the command ‘endscript’ is typed, the status of the Script operation is shown as well as the content of the script stored. The Script is now stored in the flash memory.

When the Z-Wave 700 device is either reset or power-cycled, the content of the Script will be executed. Please refer to Figure 23 to see what RailTest outputs when the script from Figure 22 is executed by means of a reset pulse.

```
RAIL Test App - Built: Jan 30 2019 10:30:08
{{(configRadio)}{status:0x01}{message:Invalid PA selected}{baseFreque
ncyHz:2450000000}}
> {{(runScript)}{location:flash}{status:Running}}
rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:13033}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Enab
led}}
> settxlength 20
{{(setTxLength)}{TxLength:20}}
> settxpayload 7 20
{{(setTxPayload)}{len:20}{payload: 0x0f 0x0e 0x11 0x22 0x33 0x44 0x55
 0x14 0x77 0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee 0x10 0x11 0x12 0x13}}
> setzwaveregion 6
{{(setZWaveRegion)}{ZWaveRegion:JP-Japan}{ZWaveRegionIndex:6}}
> setpower 50 raw
{{(setPower)}{powerLevel:50}{power:80}}
> setchannel 0
{{(setchannel)}{channel:0}}
> settxtone 1
> {{(setTxTone)}{Tone:Enabled}{None:Disabled}{Time:435376}}
```

Figure 23. Execution of setTxTone script

Please notice that each of the command lines are acknowledged by RailTest as if they were directly typed on the CLI.

Note: It is only after a reset / power-cycle event that the validity of a Script can be proven. So all Scripts should be validated by means of a reset event, followed by a visual inspection of the RailTest outputs as well as a functional test by means of a measurement.

4.5.9.4 RX Mode Script

If the LO leakage is to be measured by the RF regulatory authorities, this can be done by entering a Script in the Z-Wave 700 device of the product, and all the authorities must do is to apply power to the product and measure the LO leakage. No interaction with the product will be needed.

Below is an example, where the Z-Wave 700 device is initialized for RX operation:

```
'enterscript 1'       ;; Enter Script mode
'rx 0'                ;; Disable RX mode
'SetZwaveMode 1 3'    ;; Setup for Z-Wave mode
'setzwaveregion 6'    ;; Setup Z-Wave region, Japan
'setchannel 0'        ;; Setup channel 0
'rx 1'                ;; Enable RX mode
'endscript'           ;; Exit script mode
```

Figure 24 shows an example where the above script is typed into RailTest. RailTest commands are marked with blue.

```
RAIL Test App - Built: Jan 30 2019 10:30:08
{{(configRadio)}{status:0x01}{message:Invalid PA selected}{baseFreque
ncyHz:2450000000}}
> enterscript 1
rx 0
setzwavemode 1 3
setzwaveregion 6
setchannel 0
rx 1
endscript
{{(enterScript)}{location:flash}{status:Success}{script:rx 0
setzwavemode 1 3
setzwaveregion 6
setchannel 0
rx 1
}}
> █
```

Figure 24. Scripting RX Mode

Please notice that when in Script mode, commands are *not* acknowledged by RailTest.

When the letter “t” in the command ‘endscript’ is typed, the status of the Script operation is shown as well as the content of the script stored. The Script is now stored in the flash memory.

When the Z-Wave 700 device is either reset or power-cycled, the content of the Script will be executed. Please refer to Figure 25 to see what RailTest outputs when the script from Figure 24 is executed by means of a reset pulse.

```
RAIL Test App - Built: Jan 30 2019 10:30:08
{{(configRadio)}{status:0x01}{message:Invalid PA selected}{baseFreque
ncyHz:2450000000}}
> {{(runScript)}{location:flash}{status:Running}}
rx 0
{{(rx)}{Rx:Disabled}{Idle:Enabled}{Time:13033}}
> setzwavemode 1 3
{{(setZWaveMode)}{ZWAVE:Enabled}{Promiscuous:Enabled}{BeamDetect:Enab
led}}
> setzwaveregion 6
{{(setZWaveRegion)}{ZWaveRegion:JP-Japan}{ZWaveRegionIndex:6}}
> setchannel 0
{{(setchannel)}{channel:0}}
> rx 1
{{(rx)}{Rx:Enabled}{Idle:Disabled}{Time:410894}}
> █
```

Figure 25. Execution of RX Mode Script

Please notice that each of the command lines are acknowledged by RailTest as if they were directly typed on the CLI.

Note: It is only after a reset / power-cycle event that the validity of a Script can be proven. So all Scripts should be validated by means of a reset event, followed by a visual inspection of the RailTest outputs as well as a functional test by means of a measurement.

4.5.10 How to read UUID

Each Z-Wave device is programmed with a unique UUID during the manufacturing of the device. This UUID can be used to track the device, correlate production test measurements with the individual devices or for other purposes.

RailTest can be used to read the UUID from the Z-Wave device.

Below is an example on how to read the UUID using RailTest:

```
'getmemw 0x0FE081F0'        ;; Read lower 32 bits of UUID
'getmemw 0x0FE081F4'        ;; Read upper 32 bits of UUID
```

The output of the "getmemw `0x0FE081F0`" will give you the lower 32 bits of the UUID, the output of the "getmemw `0x0FE081F4`" will give you the upper 32bits of the UUID, so the entire UUID will be:

UUID = ((getmemw `0x0FE081F4 << 32) || (`getmemw `0x0FE081F0))`

4.6 Tips and Tricks

Using RailTest requires a lot of command typing to setup the correct parameters. It can be an advantage to find a UART terminal program that can be scripted or using a scripting programming language like Perl or Python. Using scripted control of RailTest has several advantages:

1. Once created, setting up for repeated measurements is much faster since the typing of the individual commands can be omitted.
2. Once debugged, the scripts will ensure identical setup from measurement session to measurement session, eliminating the risk of a type-error to ruin the measurement results.
3. Scripts can be handed over to third parties, e.g., RF regulatory authorities, and this limits the amount of instructions needed for the third party.

If a product can only be tested by the RF regulatory authorities when disconnected from a UART connection, the scripting functionality of RailTest, as described in section 4.5.9, is the solution to use. With the scripting mode, all non-interactive operations are stored in the Z-Wave 700 device flash and executed by the Z-Wave 700 device after a reset / power-cycle.

REFERENCES

[1] UG162: Simplicity Commander Reference Guide
[2] https://www.silabs.com/products/wireless/mesh-networking/z-wave/benefits/technology/global-regions