

nRF91x1 Production Test AT Commands

Command Reference Guide

v1.1

Contents

Revision history	iv
1 Introduction	5
2 AT command syntax	6
2.1 Set command <code><CMD> [= . . .]</code>	6
2.2 Read command <code><CMD>?</code>	6
2.3 Test command <code><CMD>=?</code>	7
2.4 Response	7
3 Modem firmware versions and hardware variants	8
4 Commands	9
4.1 Manufacturer identification <code>+CGMI</code>	9
4.2 Model identification <code>+CGMM</code>	9
4.3 Revision identification <code>+CGMR</code>	10
4.4 Product serial number identification <code>+CGSN</code>	11
4.5 Short software identification <code>%SHORTSWVER</code>	12
4.6 Hardware identification <code>%HWVERSION</code>	13
4.7 Modem build UUID <code>%XMODEMUUID</code>	14
4.8 SiP 2DID <code>%2DID</code>	14
4.9 Read device UUID <code>%DEVICEUUID</code>	15
4.10 Band lock <code>%XBANDLOCK</code>	15
4.11 Battery voltage <code>%XVBAT</code>	17
4.12 Modem trace activation <code>%XMODEMTRACE</code>	17
4.13 System mode <code>%XSYSTEMMODE</code>	19
4.14 Extra maximum TX power reduction <code>%XEMPR</code>	21
4.15 Write content to file <code>%XFILEWRITE</code>	22
4.16 COEX0 pin control configuration <code>%XCOEX0</code>	23
4.17 MAGPIO configuration <code>%XMAGPIO</code>	25
4.18 SiP-external MIPI RFFE device introduction <code>%XMIPIRFFEDEV</code>	27
4.19 SiP-external MIPI RFFE device control configuration <code>%XMIPIRFFECTRL</code>	29
4.20 Alternative configuration of SiP antenna switch <code>%XANTCFG</code>	32
4.21 COEX2 pin timing configuration <code>%XCOEX2</code>	33
4.22 Antenna detection test <code>%XANTDETMAGPIO</code>	35
4.23 RX test <code>%XRFTEST</code>	35
4.24 TX test <code>%XRFTEST</code>	37
4.25 GNSS SNR test <code>%XRFTEST</code>	43
4.26 RX SNR test <code>%XRFTEST</code>	44
4.27 AFC measurement and set <code>%XRFTEST</code>	46
4.28 RF sensor test <code>%XRFTEST</code>	49
4.29 Modem GPIO functionality test <code>%XGPIOTEST</code>	51
4.30 File system sync to flash <code>%XFSSYNC</code>	52
4.31 MIPI RFFE interface access <code>%XMIPIRFFEDBG</code>	54
4.32 SIM interface functionality test <code>%SIMTEST</code>	55
4.33 Low frequency jitter test <code>%LFCLKTEST</code>	56
4.34 TX self-calibration and test <code>%TXSELFAL</code>	57
4.35 Write IMEI <code>%IMEIWRITE</code>	58
Glossary	60

Legal notices. 63

Revision history

Date	Version	Description
January 2024	1.1	Updated: <ul style="list-style-type: none">• RX test %XRFTEST on page 35• TX test %XRFTEST on page 37• RX SNR test %XRFTEST on page 44• AFC measurement and set %XRFTEST on page 46
January 2024	1.0	First release

1 Introduction

This document describes the *AT commands* used to test and verify nRF9131 and nRF9161 (nRF91x1) in manufacturing. The *Production Test Image (PTI)* modem firmware supports a subset of AT commands for test purposes in non-signaling mode.

The production test AT commands are supported only by PTI modem firmware versions.

2 AT command syntax

The AT Commands have standardized syntax rules.

Words enclosed in <angle brackets> are references to syntactical elements. Words enclosed in [square brackets] represent optional items which may be left out from the command line at the specified point. The brackets are not used when the words appear in the command line.

<CR>, <LF>, and terminating NUL are allowed in an AT command sent by an application, but are not mandatory when using an interface where both the command string and length of command string are provided.

All standard AT commands for controlling a phone or a modem or managing the SMS feature begin with a plus sign (+), whereas Nordic-proprietary commands begin with a percent sign (%).

A string type parameter input should be enclosed between quotation marks ("").

For more information, see [3GPP 27.007 AT command set for User Equipment \(UE\)](#) and [3GPP 27.005 Use of Data Terminal Equipment - Data Circuit terminating Equipment \(DTE - DCE\) interface for Short Message Service \(SMS\) and Cell Broadcast Service \(CBS\)](#).

2.1 Set command <CMD> [= . . .]

Set commands set values or perform actions.

Example:

```
AT+CMD=1
```

where

- AT is the command line prefix
- + is the prefix for extended commands
- CMD is the body of a basic command
- 1 is a subparameter (multiple subparameters are separated by commas)

2.2 Read command <CMD>?

Read commands check the current values of subparameters.

Example:

```
AT+CMD?
```

where

- AT is the command line prefix
- + is the prefix for extended commands
- CMD is the body of a basic command
- ? represents a read command

2.3 Test command <CMD>=?

Test commands test the existence of the command and provide information about the type of its subparameters. Some test commands have also other functionality, which is described in the command-specific chapters.

Example:

```
AT+CMD=?
```

where

- AT is the command line prefix
- + is the prefix for extended commands
- CMD is the body of a basic command
- =? represents a test command for checking possible subparameter values

2.4 Response

AT responds to all commands with a final response.

The response is one of the following:

```
OK<CR><LF>  
ERROR<CR><LF>
```

"CMS ERROR:" is used as an error response for SMS related commands specified in *3GPP 27.005*.

Some commands may also produce a varying number of information response lines before the final response. An information response can be received only when a command-specific response syntax is specified. An information response line usually starts with a prefix, which is the command entered:

```
+CMD: [...]<CR><LF>
```

Some commands may also produce notifications, which do not start with the command prefix:

```
AT+CGSN  
490154203237518  
OK
```

3

Modem firmware versions and hardware variants

This document describes AT commands used in all versions of the nRF91x1 *PTI* modem firmware.

The modem firmware versions that support a command are marked in the command description with the following version tag: `vX.X.X`

If a parameter is not supported by all versions of the modem firmware that support the command, the modem firmware versions that support the parameter are marked after the parameter with the version tag.

The version tags are read as follows:

- If a command or parameter is marked `v2.2.x`, it is supported by modem firmware versions where the first two digits are 2 and 2.
- If a command or parameter is marked `v2.2.x≥3`, it is supported by modem firmware versions where the first two digits are 2 and 2 and the third digit is greater than or equal to 3.
- If a command or parameter is marked `v2.2.3`, it is supported only by modem firmware version 2.2.3.
- If a command or parameter is marked `nRF9131`, it is supported only by nRF9131.
- If a command or parameter is marked `nRF9161`, it is supported only by nRF9161.

If a hardware variant is not specified for a command or parameter, it is supported by both nRF9131 and nRF9161.

4 Commands

The commands can be used to read, write, and configure the modem parameters that are used in manufacturing testing.

4.1 Manufacturer identification +CGMI

The **+CGMI** command requests manufacturer identification. v2.2.x

For reference, see *3GPP 27.007 Ch. 5.1*.

4.1.1 Set command

The set command requests manufacturer identification.

Syntax:

```
+CGMI
```

Response syntax:

```
<manufacturer>
```

The <manufacturer> parameter returns a string of up to 2048 characters followed by <CR><LF>.

The following command example reads the manufacturer ID:

```
AT+CGMI
Nordic Semiconductor ASA
OK
```

4.1.2 Read command

The read command is not supported.

4.1.3 Test command

The test command is not supported.

4.2 Model identification +CGMM

For reference, see *3GPP 27.007 Ch. 5.2*. v2.2.x

4.2.1 Set command

The set command requests *System in Package (SiP)* model identification.

Syntax:

```
+CGMM
```

Response syntax:

```
<model>
```

The <model> parameter returns a string of up to 2048 characters followed by <CR><LF>OK.

The following command example reads the model ID:

```
AT+CGMM
nRF9131-LACA
OK
```

4.2.2 Read command

The read command is not supported.

4.2.3 Test command

The test command is not supported.

4.3 Revision identification +CGMR

The **+CGMR** command requests modem firmware revision identification. v2.2.x

For reference, see *3GPP 27.007 Ch. 5.3*.

4.3.1 Set command

The set command requests revision identification.

Syntax:

```
+CGMR
```

Response syntax:

```
<revision>
```

The <revision> parameter returns a string of up to 2048 characters followed by <CR><LF>OK.

The following command example reads the revision ID:

```
AT+CGMR
mfw-pti_nrf91x1_2.0.0
OK
```

4.3.2 Read command

The read command is not supported.

4.3.3 Test command

The test command is not supported.

4.4 Product serial number identification +CGSN

The **+CGSN** command requests product serial number identification. v2.2.x

For reference, see *3GPP 27.007 Ch. 5.4*.

4.4.1 Set command

The set command requests product serial number identification.

Syntax:

```
+CGSN[=<snt>]
```

The set command parameters and their defined values are the following:

<snt>

- 0 – Respond with <sn> (default)
- 1 – Respond with +CGSN: <imei>
- 2 – Respond with +CGSN: <imeisv>
- 3 – Respond with +CGSN: <svn>

<sn>

Information text determined by the manufacturer. Up to 2048 characters. *Electronic Serial Number (ESN)* returned if available. *International Mobile (Station) Equipment Identity (IMEI)* returned if ESN not available.

<imei>

String in decimal format indicating the IMEI. Composed of *Type Allocation Code (TAC)* (8 digits), *Serial Number (SNR)* (6 digits), and *Check Digit (CD)* (1 digit).

<imeisv>

String in decimal format indicating the *International Mobile (Station) Equipment Identity, Software Version (IMEISV)*. The 16 digits of IMEISV are composed of TAC (8 digits), SNR (6 digits), and *Software Version Number (SVN)* (2 digits).

<svn>

String in decimal format indicating the current SVN which is part of IMEISV.

Response syntax when <snt>=0 (or omitted):

```
<sn>
```

Response syntax for other <snt> values:

```
+CGSN: <string>
```

where <string> can be <imei>, <imeisv>, or <svn>.

The following command example reads the serial number:

```
AT+CGSN
352656100367872
OK
```

The following command example reads the IMEI:

```
AT+CGSN=1
+CGSN: "352656100367872"
OK
```

The following command example reads the IMEI when it is not written to the device:

```
AT+CGSN
FFFFFFFFFFFFFFFF
OK
```

4.4.2 Read command

The read command is not supported.

4.4.3 Test command

The test command returns a list of supported <sn> values.

Response syntax:

```
+CGSN: (list of supported <sn>s)
```

The test command parameter and its defined values are the following:

<sn>

- 0 – Respond with <sn> (default)
- 1 – Respond with +CGSN: <imei>
- 2 – Respond with +CGSN: <imeisv>
- 3 – Respond with +CGSN: <svn>

The following command example reads the supported types of product serial number identifications:

```
AT+CGSN=?
+CGSN: (0-3)
OK
```

4.5 Short software identification %SHORTSWVER

The proprietary %**SHORTSWVER** command requests short software identification. v2.2.x

4.5.1 Set command

The set command requests short software identification.

Syntax:

```
%SHORTSWVER
```

Response syntax:

```
%SHORTSWVER: <version_string>
```

The response parameter and its defined value are the following:

<version_string>

String without double quotes. Short software identification.

The following command example requests short software identification:

```
AT%SHORTSWVER
%SHORTSWVER: mfw-pti_nrf91x1_2.0.0
OK
```

4.5.2 Read command

The read command is not supported.

4.5.3 Test command

The test command is not supported.

4.6 Hardware identification %HWVERSION

The proprietary **%HWVERSION** command requests hardware identification. v2.2.x

4.6.1 Set command

The set command requests hardware identification.

Syntax:

```
%HWVERSION
```

Response syntax:

```
%HWVERSION: <version_string>
```

The response parameter and its defined value are the following:

<version_string>

String without double quotes. Hardware version.

The following command example requests hardware identification:

```
AT%HWVERSION
%HWVERSION: nRF9131 LACA A0A
OK
```

4.6.2 Read command

The read command is not supported.

4.6.3 Test command

The test command is not supported.

4.7 Modem build UUID %XMODEMUUID

The proprietary **%XMODEMUUID** command requests the *UUID* of a modem build. v2.2.x

4.7.1 Set command

The set command requests the *UUID* of a modem build.

Syntax:

```
%XMODEMUUID
```

Response syntax:

```
%XMODEMUUID: <UUID>
```

The response parameter and its defined value are the following:

<UUID>

String without double quotes. UUID of the modem build.

The following command example requests the UUID of a modem build:

```
AT%XMODEMUUID
%XMODEMUUID: 25c95751-efa4-40d4-8b4a-1dcaab81fac9
OK
```

4.7.2 Read command

The read command is not supported.

4.7.3 Test command

The test command is not supported.

4.8 SiP 2DID %2DID

The proprietary **%2DID** command reads the 2DID of the nRF91x1 *SiP*. The 2DID corresponds to the two-dimensional barcode etched on top of the module. v2.2.x

4.8.1 Set command

The set command reads the 2DID (*SERIAL_NUMBER_A*) of the nRF91x1 *SiP*. If the 2DID has not been written to the device, the response is an empty string.

Syntax:

```
%2DID
```

The following command example reads the 2DID when a 2DID exists on the device:

```
AT%2DID
%2DID: "F1413BQF072303"
OK
```

The following command example reads the 2DID when a 2DID does not exist on the device:

```
AT%2DID
%2DID: ""
OK
```

4.8.2 Read command

The read command is not supported.

4.8.3 Test command

The test command is not supported.

4.9 Read device UUID %DEVICEUUID

The proprietary **%DEVICEUUID** command reads the device-specific *UUID*. v2.2.x

4.9.1 Set command

The set command reads the 36-character hardware *UUID* from the device.

Syntax:

```
%DEVICEUUID
```

The following command example reads the device UUID:

```
AT%DEVICEUUID
%DEVICEUUID: 504d5632-3739-42fb-80c8-1411df972166
OK
```

4.9.2 Read command

The read command is not supported.

4.9.3 Test command

The test command is not supported.

4.10 Band lock %XBANDLOCK

The proprietary **%XBANDLOCK** command sets locked bands. v2.2.x

4.10.1 Set command

The set command sets locked bands and bitmasks to limit supported bands.

Supported bands are masked with permanent and runtime masks. A logical AND operation is performed to **%XBANDLOCK** commands. If a permanent or runtime mask has been given, it is required that the second mask has at least one band in common with the first mask. Otherwise, the command returns `ERROR`. The command returns `ERROR` at an attempt to disable all supported bands.

When permanently limiting the used bands with `%XBANDLOCK=3` and then performing TX self-calibration with `%TXSELFCAL=1`, execute `%XBANDLOCK=3` before `%TXSELFCAL=1`. nRF9131

Syntax:

```
%XBANDLOCK=<operation>[,<band_mask>]
```

The set command parameters and their defined values are the following:

<operation>

- 0 – Remove band locks.
- 1 – Set permanent band mask.
- 2 – Set runtime band mask.
- 3 – Set erase-protected band mask. Writing the mask is possible only in the *PTI* firmware. nRF9131

<band_mask>

String. Bit string where LSB is band 1. Leading zeros can be omitted. Maximum length 88 characters.

The following command example sets permanent band 4 lock:

```
AT%XBANDLOCK=1,"1000"
OK
```

The following command example sets runtime band 4 and 13 lock:

```
AT%XBANDLOCK=2,"1000000001000"
OK
```

4.10.2 Read command

The read command reads locked bands.

An empty string is returned if bandlock is not set.

Response syntax:

```
%XBANDLOCK: <permanent_lock>,<run_time_lock>,<erase_protected_lock>
```

The read response parameter and its defined value are the following:

<permanent_lock>

String. Bit string where LSB is band 1. Leading zeros can be omitted. Maximum length 88 characters.

<run_time_lock>

String. Bit string where LSB is band 1. Leading zeros can be omitted. Maximum length 88 characters.

<erase_protected_lock>

String. Bit string where LSB is band 1. Leading zeros can be omitted. Maximum length 88 characters.

4.12.1 Set command

The set command activates and deactivates modem trace.

Syntax:

```
%XMODEMTRACE=<oper>[,<set_id>[,<bitmap_id>,<bitmap>]]
```

Response syntax for Read trace bitmap:

```
%XMODEMTRACE: <bitmap>
```

The set command parameters and their defined values are the following:

<oper>

Operation

- 0 – Deactivate traces.
- 1 – Activate predefined trace set.
- 2 – Activate trace bitmap. To be used only on request by Nordic customer support.
- 3 – Read trace bitmap. To be used only on request by Nordic customer support.

<set_id>

Integer. Predefined trace set identifier.

- 1 – Coredump only.
- 2 – *Long-Term Evolution (LTE), Internet Protocol (IP), Global Navigation Satellite System (GNSS), and coredump.*
- 3 – Reserved for future use.
- 4 – IP only.
- 5 – LTE and IP.

<bitmap_id>

Integer. Trace bitmap identifier. Used only with the assistance of Nordic customer support.

<bitmap>

String in hexadecimal *International Reference Alphabet (IRA)* format. Used only with the assistance of Nordic customer support.

The following command example activates trace set 1 (Coredump only):

```
AT%XMODEMTRACE=1,1
OK
```

The following command example deactivates trace:

```
AT%XMODEMTRACE=0
OK
```

4.12.2 Read command

The read command is not supported.

4.12.3 Test command

The test command is not supported.

4.13 System mode %XSYSTEMMODE

The proprietary **%XSYSTEMMODE** command sets the modem system mode. v2.2.x

It is mandatory to set the command, but it has no effect on the *PTI* firmware.

4.13.1 Set command

The set command sets the supported system modes of the modem.

Note: Only one supported LTE mode is allowed at a time.

Syntax:

```
%XSYSTEMMODE=<LTE_M_support>,<NB_IoT_support>,<GNSS_support>,<LTE_preference>
```

+CME ERROR code

- 518 – Not allowed in active state.
- 522 – Band configuration not valid for selected mode.

The set command parameters and their defined values are the following:

<LTE_M_support>

- 0 – *LTE-M* not supported
- 1 – *LTE-M* supported

<NB_IoT_support>

- 0 – *Narrowband Internet of Things (NB-IoT)* not supported
- 1 – *NB-IoT* supported

<GNSS_support>

- 0 – *GNSS* not supported
- 1 – *GNSS* supported

<LTE_preference>

- 0 – No preference. Initial system selection is based on history data and *Universal Subscriber Identity Module (USIM)*. If history data or *USIM* configuration are not available, *LTE-M* is prioritized in the initial system selection.
- 1 – *LTE-M* preferred.
- 2 – *NB-IoT* preferred.
- 3 – Network selection priorities override system priority, but if the same network or equal priority networks are found, *LTE-M* is preferred.
- 4 – Network selection priorities override system priority, but if the same network or equal priority networks are found, *NB-IoT* is preferred.

Note: If <LTE_preference> is set to a non-zero value, <LTE_M_support> or <NB_IoT_support> or both must be set.

The following command example sets LTE-M and GNSS as the system modes. No preferred LTE mode set:

```
AT%XSYSTEMMODE=1,0,1,0
OK
```

4.13.2 Read command

The read command reads the supported modem system modes.

Response syntax:

```
%XSYSTEMMODE: <LTE_M_support>,<NB_IoT_support>,<GNSS_support>,<LTE_preference>
```

The read response parameters and their defined values are the following:

<LTE_M_support>

- 0 – LTE-M not supported
- 1 – LTE-M supported

<NB_IoT_support>

- 0 – NB-IoT not supported
- 1 – NB-IoT supported

<GNSS_support>

- 0 – GNSS not supported
- 1 – GNSS supported

<LTE_preference>

- 0 – No preference. Initial system selection is based on history data and *USIM*. If history data or *USIM* configuration are not available, LTE-M is prioritized in the initial system selection.
- 1 – LTE-M preferred.
- 2 – NB-IoT preferred.
- 3 – Network selection priorities override system priority, but if the same network or equal priority networks are found, LTE-M is preferred.
- 4 – Network selection priorities override system priority, but if the same network or equal priority networks are found, NB-IoT is preferred.

Note: If <LTE_preference> is set to a non-zero value, <LTE_M_support> or <NB_IoT_support> or both must be set.

The following command example reads the supported system mode:

```
AT%XSYSTEMMODE?
%XSYSTEMMODE: 1,0,0,0
OK
```

4.13.3 Test command

The test command is not supported.

4.14 Extra maximum TX power reduction %XEMPR

The proprietary **%XEMPR** command allows to configure an extra reduction of 0.5 or 1 dB to the maximum transmission power on all or selected supported 3GPP bands separately in the *NB-IoT* and *LTE-M* modes.

v2.2.x

Note:

- The use of this command in the nRF91x1 modem firmware image can be permanently prevented with the **%XPRODDONE** command.
- The **%XEMPR** TX power reduction feature can be used in the *PTI* modem firmware image even if the **%XEMPR** feature has been disabled with **%XPRODDONE**.

4.14.1 Set command

The set command sets the extra reduction to maximum TX power.

The command can be given separately to the *NB-IoT* and *LTE-M* modes. If a band is not mentioned in the command, the EMPR is 0 for that band. The command cannot be used to increase transmission power. **%XEMPR** should be given before the activation of the modem to be effective. If a valid EMPR configuration exists, the reduction is automatically applied to the applicable transmissions.

Syntax:

```
%XEMPR=<system_mode>,<k>,<band0>,<pr0>,<band1>,<pr1>,...,<bandk-1>,<prk-1>
```

or

```
%XEMPR=<system_mode>,0,<pr_for_all_bands>
```

The set command parameters and their defined values are the following:

<system_mode>

- 0 – NB-IoT
- 1 – LTE-M

<k>

The number of bands to which EMPR is set. If <k> is 0, the next parameter <pr_for_all_bands> is applied to all supported 3GPP bands. The **%XEMPR** command supports listing the power reduction for all the supported bands of nRF91x1 in one command for both NB-IoT and LTE-M.

<bandn>

The number of the 3GPP band to which the following <prn> is applied.

<prn>

- EMPR for <bandn>
- 0 – 0 dB
- 1 – Maximum power reduced 0.5 dB
- 2 – Maximum power reduced 1.0 dB
- > 2 is not allowed

The following command example reduces the maximum TX power on all bands in the NB-IoT mode by 1 dB:

```
AT%XEMPR=0,0,2
OK
```

The following command example reduces the maximum TX power on three (<k>=3) bands in the LTE-M mode (<system_mode>=1). The maximum TX power is reduced by 1 dB on bands 5 and 8 and by 0.5 dB on band 13:

```
AT%XEMPR=1,3,5,2,8,2,13,1
OK
```

The following command example deletes the existing configuration by sending the command without any parameters:

```
AT%XEMPR
OK
```

4.14.2 Read command

The read command reads the currently active configuration.

Syntax:

```
%XEMPR?
```

The following command example reads the currently active configuration after both examples of the set command have been given:

```
AT%XEMPR?
%XEMPR:
0,0,2
1,3,5,2,8,2,13,1
OK
```

4.14.3 Test command

The test command is not supported.

4.15 Write content to file %XFILEWRITE

The proprietary **%XFILEWRITE** command writes given content to a file. v2.2.x

4.15.1 Set command

The set command requests to write content to a file.

Syntax:

```
%XFILEWRITE=<file>,<content>[,<checksum>]
```

+CME ERROR code

518 – Not allowed in active state.

528 – Not allowed when power off warning is active.

The set command parameters and their defined values are the following:

<file>

1 – GNSS almanac.

Up-to-date content of the GNSS almanac file can be found in the GNSS sample applications of nRF Connect SDK.

<content>

Hexadecimal numbers containing two *IRA* characters per octet.

<checksum>

Checksum of the content calculated over the sha256 algorithm. Mandatory for GNSS almanac file.

The following command example requests to write the GNSS almanac to a file:

```
AT%XFILEWRITE=1,
  "f0ea0200312a080000000031...", "bf38c845eab79f459f7b3ef4393f1a2860d309952832a0073b990f12a7274e64"
OK
```

4.15.2 Read command

The read command is not supported.

4.15.3 Test command

The test command is not supported.

4.16 COEX0 pin control configuration %XCOEX0

The proprietary **%XCOEX0** command writes the **COEX0** pin configuration to device RAM. v2.2.x

The **COEX0** pin can be configured to switch its state based on the modem's RF frequency to enable external *Low-Noise Amplifier (LNA)* in *GNSS* mode. The behavior is similar to the **%XMAGPIO** command except this command controls only one pin.

The **%XCOEX0** command needs to be sent before any modem activity occurs. Based on the given configuration, the modem applies the **COEX0** state corresponding to the RF frequency range automatically during runtime.

4.16.1 Set command

The set command writes the **COEX0** pin configuration to device RAM.

When RF is turned off, the given **COEX0** state is inverted.

Syntax:

```
%XCOEX0=<count>,<state_0>,<freqlo_0>,<freqhi_0>,...
<state_count-1><freqlo_count-1><freqhi_count-1>
```

The set command parameters and their defined values are the following:

<count>

The number of frequency ranges. Valid values are 1, 2, 3, and 4.

<state_x>

The state of **COEX0** with the following frequency range. Valid values are 0 and 1.

<freqlo_x>

Low limit for the frequency range in MHz.

<freqhi_x>

High limit for the frequency range in MHz.

The following command example sets **COEX0** to 1 when *GNSS* is enabled (and 0 when *GNSS* is turned off). **COEX0** is not used with other frequencies (or *LTE*):

```
AT%XCOEX0=1,1,1570,1580
OK
```

The following command example sets **COEX0** to 1 when *GNSS* is enabled, or *LTE* frequency is 600 MHz to 800 MHz or 2000 MHz to 2180 MHz:

```
AT%XCOEX0=3,1,1570,1580,1,2000,2180,1,600,800
OK
```

If the command is given without any parameters, it deletes the previously written values:

```
AT%XCOEX0
OK
```

4.16.2 Read command

The read command returns the stored pin configuration.

Response syntax:

```
%XCOEX0: <count>,<state_0>,<freqlo_0>,<freqhi_0>,...
<state_count-1><freqlo_count-1><freqhi_count-1>
```

The read response parameters and their defined values are the following:

<count>

The number of frequency ranges. Valid values are 1, 2, 3, and 4.

<state_x>

The state of **COEX0** with the following frequency range. Valid values are 0 and 1.

<freqlo_x>

Low limit for the frequency range in MHz.

<freqhi_x>

High limit for the frequency range in MHz.

The following command example returns the stored configuration:

```
AT%XCOEX0?
AT%XCOEX0: 3,1,1570,1580,1,2000,2180,1,600,800
OK
```

4.16.3 Test command

The test command is not supported.

4.17 MAGPIO configuration %XMAGPIO

The proprietary %**XMAGPIO** command writes the MAGPIO configuration to device RAM. v2.2.x

The MAGPIO pins can be used, for example, to control an external antenna tuner, or any other *General-Purpose Input/Output (GPIO)*-controlled device, whose state depends on modem's RF frequency. The %**XMAGPIO** command needs to be sent before any modem activity occurs. Based on the given configuration, the modem applies the MAGPIO state corresponding to the RF frequency range automatically during runtime.

4.17.1 Set command

The set command writes the MAGPIO configuration to device RAM.

Syntax:

```
%XMAGPIO=<gpio_0>,<gpio_1>,<gpio_2>,<num_ranges>,<state_0>,<flo_0>,<fhi_0><state_1>,<flo_1>,<fhi_1>,...
```

A command without any parameters deletes the previously written values.

The set command parameters and their defined values are the following:

<gpio_x>

- 0 – MAGPIO_x is not used
- 1 – MAGPIO_x used

<num_ranges>

The number of frequency ranges, maximum value 12

<state_y>

Settings of the MAGPIO pins for the range x that follows

<flo_y>

Frequency range low value when the setting is active, in MHz

<fhi_y>

Frequency range high value when the setting is active, in MHz

The following table contains an example configuration for an antenna tuner:

	State	MAGPIO2	MAGPIO1	MAGPIO0	Low MHz	High MHz
Unused	0	0	0	0	-	-
LTE(746–803)	1	0	0	1	746	803
LTE(698–746)	2	0	1	0	698	746
LTE(1710–2200)	2	0	1	0	1710	2200
LTE(849–894)	3	0	1	1	849	894
LTE(894–960)	4	1	0	0	894	960
Unused	5	1	0	1	-	-
LTE(803–849)	6	1	1	0	803	849
GNSS	7	1	1	1	1574	1577

Table 1: Example configuration for an antenna tuner

The following command example writes seven ranges to device RAM:

```
AT%XMAGPIO=1,1,1,7,1,746,803,2,698,746,2,1710,2200,3,849,894,4,894,960,6,803,849,7,
1574,1577
OK
```

The following command example writes three ranges to device RAM:

```
AT%XMAGPIO=1,1,1,3,0,1574,1577,1,705,747,6,748,804
OK
```

The following command example deletes the previously written values:

```
AT%XMAGPIO
OK
```

4.17.2 Read command

The read command returns the stored MAGPIO configuration.

Response syntax:

```
%XMAGPIO:
<gpio_0>,<gpio_1>,<gpio_2>,<num_ranges>,<state_0>,<flo_0>,<fhi_0><state_1>,<flo_1>,
<fhi_1>,...
```

The read response parameters and their defined values are the following:

<gpio_x>

0 – MAGPIO_x is not used

1 – MAGPIO_x used

<num_ranges>

The number of frequency ranges, maximum value 12

<state_y>

Settings of the MAGPIO pins for the range x that follows

<flo_y>

Frequency range low value when the setting is active, in MHz

<fhi_y>

Frequency range high value when the setting is active, in MHz

The following command example returns the stored configuration:

```
AT%XMAGPIO?
AT%XMAGPIO: 1,1,1,3,0,1574,1577,1,705,747,6,748,804
OK
```

4.17.3 Test command

The test command is not supported.

4.18 SiP-external MIPI RFFE device introduction %XMIPIRFFEDEV

The proprietary **%XMIPIRFFEDEV** command introduces the device and its static parameters to the nRF91x1 SiP. v2.2.x

nRF91x1 can be configured to control a SiP-external, *MIPI RF Front-End Control Interface (RFFE)*-controlled device to a limited extent. Antenna tuner is the primary use case.

After introducing the MIPI RFFE device, the configuration for the various control phases can be given using the **%XMIPIRFFCTRL** command.

The **%XMIPIRFFEDEV** command needs to be sent before any modem activity occurs.

4.18.1 Set command

The set command writes the **%XMIPIRFFEDEV** configuration to nRF91x1 RAM memory.

Syntax:

```
%XMIPIRFFEDEV=<dev_id>,<def_usid>,<prod_id>,<man_id>,<pm_trig>
```

The set command parameters and their defined values are the following:

<dev_id>

Selectable identification number for the device. Non-zero. Valid range 1–255.
The given <dev_id> is used with the **%XMIPIRFFCTRL** and **%XMIPIRFFDBG** commands.

<def_usid>

A 4-bit default *Unique Slave Identifier (USID)* for the MIPI RFFE device. Typically, 7 for antenna tuners (as suggested by MIPI).

<prod_id>

An 8-bit PRODUCT_ID of the MIPI RFFE device. Only used if automatic reprogramming of the USID is needed. EXT_PRODUCT_ID is not supported.

<man_id>

A 10-bit MANUFACTURER_ID of the MIPI RFFE device. Only used if automatic reprogramming of the USID is needed.

<pm_trig>

An 8-bit content for PM_TRIG (address 0x1C = 28 dec) register. This is for setting the default power and triggering mode. The setting of PM_TRIG can be also changed in the ON phase. See [%XMIPIRFFCTRL](#).

All numbers should be given as decimals, that is, not as hexadecimals. Currently, nRF91x1 supports only one *SiP*-external MIPI RFFE-controlled device.

4.18.2 Read command

The read command returns the introductory information given for a device using the [%XMIPIRFFEDEV](#) command and the phase-specific configurations given in the [%XMIPIRFFCTRL](#) command. There is no dedicated read command for [%XMIPIRFFCTRL](#).

Response syntax:

```
%XMIPIRFFEDEV?
%XMIPIRFFEDEV: <dev_id>,<def_usid>,<prod_id>,<man_id>,<pm_trig>
INIT:
ON:
OFF:
PWROFF:
OK
```

The read response parameters and their descriptions for the “%XMIPIRFFEDEV” row are as defined in [Set command](#) on page 27 if a valid [%XMIPIRFFEDEV](#) command has been given earlier. Otherwise, the row is empty. The phase-specific rows that follow (INIT, ON, OFF, PWROFF) contain the parameters given for that phase or they are empty.

In the following command example, the following commands have been given:

```
AT%XMIPIRFFEDEV=1,7,171,331,184
OK
AT%XMIPIRFFCTRL= 1,1,1,28,56,6,1,2,2,3,750,3,8,850,18,9,1000,20,12,1700,35,19,1900,37,
25,2200
OK
```

In the following command example, the read command returns:

```
AT%XMIPIRFFEDEV?
%XMIPIRFFEDEV: 1,7,171,331,184
INIT:
ON: 1,1,1,28,56,6,1,2,2,3,750,3,8,850,18,9,1000,20,12,1700,35,19,1900,37,25,2200
OFF:
PWROFF:
OK
```

4.18.3 Delete configuration

A *MIPI RFFE* device configuration and control phase information can be deleted from the nRF91x1 memory using this command.

Syntax:

```
%XMIPIRFFEDEV=<dev_id>
```

The following command deletes the device whose <dev_id>=1 and all related phase controls that have been given using the **%XMIPIRFFCTRL** command:

```
AT%XMIPIRFFEDEV=1
OK
```

CAUTION: The combined load of *Printed Circuit Board (PCB)* routing, the input load of the MIPI RFFE-controlled device, and any parasitic load from application shall not exceed 15 pF at **SCLK** or **SDATA** pins. This load translates roughly to narrow transmission line length of less than 10 cm at the application board but it is dependent on the actual PCB design. A load higher than 15 pF at **SCLK** or **SDATA** pin increases the risk of unwanted behavior of the nRF91x1 *SiP* itself and of MIPI RFFE control.

4.19 SiP-external MIPI RFFE device control configuration %XMIPIRFFCTRL

The proprietary **%XMIPIRFFCTRL** command is used to configure the *SiP*-external *MIPI RFFE* device register accesses in different phases of modem operation. v2.2.x

The device must first be introduced using the **%XMIPIRFFEDEV** command.

MIPI RFFE devices contain an internal register map described in the datasheet of the device. To control the device, the registers in the device must be written with valid values. The **%XMIPIRFFCTRL** command allows configuring the nRF91x1 *SiP* to write the device's registers. The register addresses, the values, and timing (phase) can be configured as described below.

The external MIPI RFFE control in nRF91x1 supports configuring the RFFE device for four different phases of RF operation. The phases are initializing (INIT), start receiving or transmitting (ON), stop receiving or transmitting (OFF), and going to sleep (PWROFF).

The **%XMIPIRFFCTRL** command must be sent separately for each phase. It is not mandatory to configure all phases.

The **%XMIPIRFFCTRL** command must be sent before any modem activity occurs.

The **%XMIPIRFFEDBG** command can be used to debug MIPI RFFE configurations by reading or writing to device registers.

The phases are defined as follows:

INIT

Applied when RF is waking up. INIT is frequency-agnostic. Controls up to four MIPI RFFE device registers. The main purpose is to allow preparation or activation of the MIPI RFFE device if activation requires long settling.

ON

Applied when RF is starting for a specific frequency or when *LTE-M* frequency hopping is performed by the modem RF. Controls a maximum of two frequency-agnostic registers that can be used for device activation, for instance. This phase also controls a maximum of two registers whose value can be defined to depend on the RF frequency of the modem. The table for the frequency-dependent control can have a maximum of 64 frequencies.

OFF

Applied when RF is stopping. The configuration is frequency-agnostic. Controls up to four MIPI RFFE device registers.

PWROFF

Applied when RF is going to sleep. The configuration is frequency-agnostic. Controls up to four MIPI RFFE device registers. The main purpose is to deactivate the MIPI RFFE device.

4.19.1 Set command

The set command writes the `%XMIPIRFFCTRL` configuration to the nRF91x1 RAM memory.

The command is given separately for each phase. It is not necessary to send the command for each phase, which means it is possible to configure only one phase.

Syntax:

```
%XMIPIRFFCTRL=<dev_id>,<phase#>,<variable_number_of_phase_specific_parameters>
```

The set command parameters and their defined values are the following:

<dev_id>

The identification number of the *MIPI RFFE* device given when it was introduced using the `%XMIPIRFFEDEV` command.

<phase#>

Number of the phase INIT = 0, ON = 1, OFF = 2, PWR_OFF = 3. All numbers must be given as decimals (hexadecimals are not allowed).

The following figure illustrates the RFFE device control in different phases:

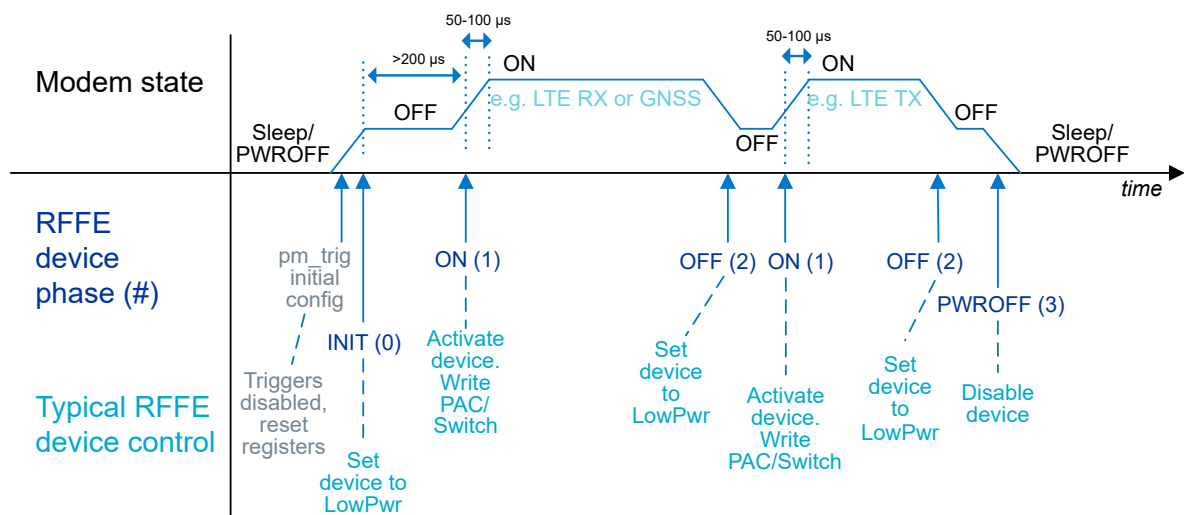


Figure 1: RFFE device control and timing in different phases

In the figure, PAC/Switch refers to a register in an example antenna tuner that controls the tunable capacitors and/or switches.

4.19.2 Phases INIT(0), OFF(2), and PWROFF(3)

Phases INIT(0), OFF(2), and PWROFF(3) are introduced here.

The syntax of each phase is the same except for the phase# parameter:

INIT (0)

```
%XMIPIRFFECTRL=<dev_id>,0,<n>,<address_0>,<data_0>,...,<address_n-1>,<data_n-1>
```

OFF (2)

```
%XMIPIRFFECTRL=<dev_id>,2,<n>,<address_0>,<data_0>,...,<address_n-1>,<data_n-1>
```

PWROFF (3)

```
%XMIPIRFFECTRL=<dev_id>,3,<n>,<address_0>,<data_0>,...,<address_n-1>,<data_n-1>
```

The parameters and their defined values are the following:

<n>

The number of address or data pairs. Valid values are 0, 1, 2, 3, 4. If the value is 0, all the following fields must be omitted.

<address_x>

The 8-bit address of the internal register in *MIPI RFFE* device. $x = 0, \dots, n-1$.

<data_x>

The 8-bit data to be written to <address_x>. $x = 0, \dots, n-1$.

4.19.3 Phase ON(1)

Phase ON(1) contains the most options for configuring the device.

Syntax:

```
%XMIPIRFFECTRL=<dev_id>,1,<n>,<act_addr_0>,<act_data_0>,<act_addr_n-1>,<act_data_n-1>,<k>,<addr_0>,<addr_1>,<data_0_0>,<data_1_0>,<freq_0>,...,<data_0_k-1>,<data_1_k-1>,<freq_k-1>
```

The parameters and their descriptions are the following:

<n>

The number of activation register address-data pairs. Valid values are 0, 1, 2. If $n = 0$, `act_addr_0/1` and `act_data_0/1` must be omitted.

<act_addr_x>

8-bit address of the first register whose value is set to activate device. This is written each time RF starts. Must be given if <n> is greater than 0.

<act_data_x>

Optional 8-bit data for the register in <act_addr_x>.
Must be given if <n> is greater than 0.

<k>

The number of frequencies in the configuration. Valid values are 0–64. If $k = 0$, all the following fields must be omitted.

<addr_0>

The 8-bit address of the first register, whose value is changed on the basis of RF frequency.

<addr_1>

The 8-bit address of the other register, whose value is changed on the basis of RF frequency. If `addr_1 == addr_0`, then only `<data_0_y>` is written.

<data_0_y>

The 8-bit data for the register in `<addr_0>`, if frequency is smaller than or equal to `<freq_y>`.

<data_1_y>

The 8-bit data for the register in `<addr_1>`, if frequency is smaller than or equal to `<freq_y>`. `data_1_y` must be given (for example, as 0) even if `addr_1 == addr_0`.

<freq_y>

Integer. The frequency in MHz to which the current RF frequency is compared. If current RF frequency is smaller than or equal to `<freq_y>`, then `<data_0_y>` is written to `<addr_0>` and `<data_1_y>` is written to `<addr_1>`. If the RF frequency is greater than `<freq_k-1>` (the last given frequency), then neither `<addr_0>` nor `<addr_1>` is written.

4.19.4 Delete configuration

The `AT%XMIPIRFFFEDEV= <dev_id >` command deletes all configurations for the *MIPI RFFE* device, including phase configurations. To delete the configuration of each phase individually, set `<n> = 0` or/and `<k> = 0` in the phase-specific command.

The following command example deletes only the ON phase configuration:

```
AT%XMIPIRFFFEDEV=<dev_id>,1,0,0
```

The following command example deletes the PWROFF phase configuration:

```
AT%XMIPIRFFFEDEV=<dev_id>,3,0
```

4.20 Alternative configuration of SiP antenna switch %XANTCFG

The proprietary `%XANTCFG` command configures the *SiP*-internal antenna switch to an alternative predefined position. v2.2.x

4.20.1 Set command

The set command configures the *SiP*-internal antenna switch to an alternative predefined position.

The `%XANTCFG` command supports one predefined setting. Therefore, in the *GNSS* mode the input signal to nRF91x1's **ANT** input is routed back out from the **AUX** output. After a valid configuration is set, nRF91x1 automatically controls the switch during *GNSS* reception. Adding further predefined settings requires a modification to modem firmware.

Syntax:

```
%XANTCFG=<cfg>
```

The set command parameter and its defined values are the following:

<cfg>

- 0 – Reserved, no action
- 1 – **ANT** input directed to **AUX** output in the GNSS mode
- 2 – Reserved, no action
- ...
- 7 – Reserved, no action

The following command example directs **ANT** input to **AUX** output:

```
AT%XANTCFG=1
OK
```

The following command example deletes the previous configuration:

```
AT%XANTCFG
OK
```

4.20.2 Read command

The read command reads the currently active configuration.

Syntax:

```
%XANTCFG?
```

The following command example returns the currently active configuration:

```
AT%XANTCFG?
%XANTCFG: 1
OK
```

4.20.3 Test command

The test command is not supported.

4.21 COEX2 pin timing configuration %XCOEX2

The proprietary **%XCOEX2** command determines the timing for enabling the **COEX2** pin. v2.2.x

By default, the **COEX2** pin is enabled automatically when LTE/GNSS RF is used and disabled when RF is powered down. The default timing of the **COEX2** pin has approximately 1 ms anticipation to the first sample in the antenna.

In addition to the default mode, the **%XCOEX2** command has two alternative anticipation times of approximately 400 μ s and 50 μ s to 100 μ s before the first sample in the antenna. The pin toggling can also be disabled.

The **COEX2** pin can also be set to toggle at only RF transmission or only RF reception.

The **%XCOEX2** command must be sent before any modem activity occurs. Based on the given configuration, the modem toggles the **COEX2** pin automatically during runtime.

4.21.1 Set command

The set command determines the timing for enabling the **COEX2** pin.

Syntax:

```
%XCOEX2[=<ctrl>]
```

The set command parameter and its defined values are the following:

<ctrl>

- Not set – Clears the current configuration and uses default values (1).
- 0 – **COEX2** pin toggling disabled, pin OFF.
- 1 – **COEX2** pin toggling enabled, default timing, 1 ms anticipation.
- 2 – **COEX2** pin toggling enabled, mid-timing, approximately 400 μ s anticipation.
- 3 – **COEX2** pin toggling enabled, as late as possible timing, approximately 50 μ s to 100 μ s anticipation.
- 4 – **COEX2** pin toggling enabled, TX indicator. Pin ON when RF is transmitting.
- 5 – **COEX2** pin toggling enabled, RX indicator. Pin ON when RF is receiving.

The following command example sets the mid-timing of the **COEX2** pin:

```
AT%XCOEX2=2
OK
```

4.21.2 Read command

The read command reads the current configuration of the **COEX2** pin.

Syntax:

```
%XCOEX2?
```

Response syntax:

```
%XCOEX2: <ctrl>
```

The read command parameter and its defined values are the following:

<ctrl>

- Not set – Clears the current configuration and uses default values (1).
- 0 – **COEX2** pin toggling disabled, pin OFF.
- 1 – **COEX2** pin toggling enabled, default timing, 1 ms anticipation.
- 2 – **COEX2** pin toggling enabled, mid-timing, approximately 400 μ s anticipation.
- 3 – **COEX2** pin toggling enabled, as late as possible timing, approximately 50 μ s to 100 μ s anticipation.
- 4 – **COEX2** pin toggling enabled, TX indicator. Pin ON when RF is transmitting.
- 5 – **COEX2** pin toggling enabled, RX indicator. Pin ON when RF is receiving.

The following command example reads the current configuration of the **COEX2** pin.

```
AT%XCOEX2?
%XCOEX2: 2
OK
```

4.21.3 Test command

The test command is not supported.

4.22 Antenna detection test %XANTDETMAGPIO

The proprietary **%XANTDETMAGPIO** command reads the MAGPIO pin status to detect if the antenna is connected. The antenna is detected when the pin is grounded to DC. v2.2.x

This command can be used, for example, in device production testing provided that the necessary circuitry between an MAGPIO pin and the antenna is in place. See [Antenna presence test using MAGPIO](#) in *Antenna and RF Interface Guidelines*.

4.22.1 Set command

The set command changes the pin specified in the command to input mode and sets internal pull-up for the corresponding pin.

After this, the pin state is read and reported in the command response.

Syntax:

```
%XANTDETMAGPIO=<magpio_pin>
```

The set command parameter and its defined values are the following:

<magpio_pin>

0, 1, 2 – The MAGPIO pin whose state is read.

The response values are the following:

1 – Antenna connected

0 – Antenna not connected

After the command, the pin state is set back to normal (high-Z).

The following command example sets MAGPIO pin 2 to input mode and sets internal pull-up for it. After this, the pin state is read and reported in the command response. After the command, the pin state is set back to normal (high impedance state):

```
AT%XANTDETMAGPIO=2
%XANTDETMAGPIO: 0
OK
```

4.22.2 Read command

The read command is not supported.

4.22.3 Test command

The test command is not supported.

4.23 RX test %XRFTTEST

The proprietary **%XRFTTEST** command can be used to test the nRF91x1 receiver. The command enables the RF receiver and measures signal power at the *SiP* antenna port with a time domain power meter and returns the measurement result and headroom. v2.2.x

The **%XRFTTEST** command allows starting RF to a desired frequency to test, debug, and verify configurations that depend on RF frequency, such as **%XMAGPIO** and **%XMIPIRFFCTRL**.

4.23.1 Set command

The set command controls the RF receiver.

Syntax:

```
%XRFTTEST=<test>,<operation>[,<param0>,<param1>,<param2>,<param3>]
```

The set command parameters and their defined values are the following:

<test>

0 – RX.

<operation>

1 – ON.

0 – OFF.

Note: Always send OFF <operation> before sending another ON <operation>.

<param0>

3GPP band number. Use 24 for GNSS reception.

DECT NR+ bands 1, 2, 9, and 22 are supported when <param3> is 10. See ETSI TS 103 636-2 V1.4.1 Table 5.2-1.

<param1>

Frequency 100 kHz.

Valid range 6000–22000 (corresponds to 600.0 MHz–2200.0 MHz).

In DECT NR+, the valid range is 18800–19300 (corresponds to 1880.0 MHz–1930.0 MHz) depending on the DECT NR+ band selected in <param0>.

Note: If *Carrier Wave (CW)* is used, an offset of about 45 kHz for NB-IoT and 300 kHz for LTE-M is recommended.

<param2>

RX signal power at the SiP antenna port.

Valid range from –127 dBm to –25 dBm.

<param3>

System mode.

0 – NB-IoT.

1 – LTE-M.

10 – DECT NR+.

Response syntax when <operation> is ON:

```
%XRFTTEST: <antenna_power>,<headroom>
```

The response parameter and its defined value are the following:

<antenna_power>

Measured power at the SiP antenna port in q8 dBm. q8 means that dividing the result by $2^8 = 256$ gives dBm.

<headroom>

Time domain signal headroom in dBFullScale.

Expected value is -15 dBFS or -16 dBFS if the signal level in generator corresponds to the level given in <param2>.

The following command example enables the RF receiver for Band 1, 2140.0 MHz, -65 dBm, NB-IoT mode:

```
AT%XRFTEST=0,1,1,21400,-65,0
%XRFTEST: -17002, -16
OK
```

Note: $-17002/256 = -66.4$ dBm

The following command example disables the RF receiver:

```
AT%XRFTEST=0,0
OK
```

4.23.2 Read command

The read command is not supported.

4.23.3 Test command

The test command is not supported.

4.24 TX test %XRFTEST

The proprietary **%XRFTEST** command can be used to enable the RF transmitter. The command also measures TX power with an internal measurement receiver in the time domain and returns the measurement result. v2.2.x

The **%XRFTEST** command allows starting RF to a desired frequency to test, debug, and verify configurations that depend on RF frequency, such as **%XMAGPIO** and **%XMIPIRFFCTRL**.

4.24.1 Set command

The set command enables the RF transmitter.

CAUTION: This command transmits power to the selected RF band and can violate the radio directives of the region or country. Make sure that the equipment is in an RF-shielded room or connected to an RF cable so that RF power does not leak.

Syntax:

```
%XRFTEST=<test>,<operation>[,<param0>,<param1>,...,<param10>]
```

The set command parameters and their defined values are the following:

<test>

1 – TX.

<operation>

1 – ON.

0 – OFF.

Note: Always send OFF <operation> before sending another ON <operation>.

<param0>

3GPP band number.

DECT NR+ band number. Bands 1, 2, 9, and 22 are supported. See *ETSI TS 103 636-2 V1.4.1 Table 5.2-1*. [v2.3.x](#)

<param1>

Frequency (100 kHz).

Valid range 6000–22000 (corresponds to 600.0 MHz–2200.0 MHz).

DECT NR+ carrier number. See *ETSI TS 103 636-2 v1.3.1 Table 5.4.2-1*. [v2.3.x](#)

<param2>

TX signal power at the *SiP* antenna port.

Valid range from +23 dBm to –50 dBm.

<param3>

System mode.

0 – *NB-IoT*.

1 – *LTE-M*.

10 – *DECT NR+*.

Note: Some *LTE* bands can be supported in only one of the system modes. To ensure that the band can be used, set the correct mode with the `%XSYSTEMMODE` command.

<param4>

Modulation.

0 – *Quadrature Phase-Shift Keying (QPSK)*.

1 – *16-state Quadrature Amplitude Modulation (16-QAM)*.

2 – Reserved.

3 – *Binary Phase-Shift Keying (BPSK)*.

4 – CW.

LTE-M: QPSK, 16-QAM, and CW.

NB-IoT: QPSK, BPSK, and CW.

Note:

- If <param4> is 4 (CW), in system modes *NB-IoT* and *LTE-M* the CW tone is offset by 255 kHz from the frequency given in <param1>.
- If <param4> is 4 (CW), the CW tone is offset from the frequency given in <param1> by 48.75 kHz in system mode *NB-IoT* and by 401.25 kHz in system mode *LTE-M*.
- If <param4> is 4 (CW), <param5>, <param6>, and <param7> are 0 (do not care).
- If <param4> is 4 (CW) in DECT NR+ system mode, the CW tone is offset by 150 kHz from the center frequency of the DECT NR+ carrier.

5 – DECT NR+ modulation. v2.3.x

DECT NR+ transmitted signal is burst-like. There is a short transmission gap of approximately 200 μ s in the signal after four DECT NR+ slots.

<param5>

Resource Block (RB)/Tone count.

Note: For allowed combinations for <param5>, <param6>, and <param7> in *NB-IoT* and *LTE-M* system modes (<param3>). See [Table 2: Allowed parameter combinations for <param5>, <param6>, and <param7> in system modes NB-IoT and LTE-M](#) on page 41.

Modulation and Coding Scheme (MCS) if <param4> is 5. v2.3.x

<param6>

RB/Tone start position.

If <param4> is 5, <param6> controls DECT NR+ Synchronization Training Field (STF) cover sequence. v2.3.1

0 – Disable cover sequence for STF.

1 – Enable cover sequence for STF.

<param7>

Subcarrier spacing.

0 – 15 k.

1 – 3.75 k.

Not applicable for DECT NR+. v2.3.x

<param8>

System bandwidth.

0 – NB-IoT.

1 – 1.4 M.

2 – Reserved.

3 – 5 MHz.

4 to 6 – Reserved.

Not applicable for DECT NR+. v2.3.x

<param9>

Narrowband index.

0 – 1.4 M.

0 to 3 – 5 M.

Defined in *3GPP TS 36.211 Ch. 6.2.7*.

Note: <param9> has effect only when <param8> is 3 (5 MHz).

Not applicable for DECT NR+. v2.3.x

<param10>

TX burst mode.

0 – Disable TX burst mode.

1 – Enable TX burst mode. Not applicable for DECT NR+. v2.3.x

For more information, see [TX burst mode](#) on page 42.

The allowed combinations for <param5>, <param6>, and <param7> for both system mode (<param3>) values are listed in the following table:

System mode <param3>	RB/Tone count <param5>	RB/Tone start position <param6>	Subcarrier spacing <param7>
NB-IoT (0)	1	0–11	0
	3	0, 3, 6, 9	0
	6	0, 6	0
	12	0	0
	1	0–47	1
LTE-M (1)	1	0–5	0
	2	0–4	0
	3	0–3	0
	4	0–2	0
	5	0–1	0
	6	0	0

Table 2: Allowed parameter combinations for <param5>, <param6>, and <param7> in system modes NB-IoT and LTE-M

Note: In system mode LTE-M, subcarrier spacing 3.75 kHz is not allowed.

Response syntax when <operation> is ON and TX burst mode is disabled:

```
%XRFTEST: <antenna_power>
```

The response parameter and its defined value are the following:

<antenna_power>

Internally measured TX power at the SiP antenna port in q4 dBm. q4 means that dividing the result by $2^4 = 16$ gives dBm.

In DECT NR+ system mode, the device does not support the self-measurement of antenna power. The power value in the response is always set to 0. [v2.3.x](#)

The following command example enables the RF transmitter for Band 5, 830.0 MHz, +17 dBm, NB-IoT, BPSK, 12 tones, tone start position 0, subcarrier spacing 15 kHz, system bandwidth NB1, NB index 0, TX burst mode disabled:

```
AT%XRFTEST=1,1,5,8300,17,0,3,12,0,0,0,0,0
%XRFTEST: 271 OK
```

Note: $271/16 = 16.9$ dBm

The following command example enables the RF transmitter for Band 5, 830 MHz, +17 dBm, LTE-M, 16-QAM, 6 RB, RB start position 0, subcarrier spacing 15 kHz, system bandwidth 5 MHz, NB index 3, TX burst mode disabled:

```
AT%XRFTEST=1,1,5,8300,17,1,1,6,0,0,3,3,0
%XRFTEST: 271
OK
```

The following command example enables the RF transmitter for DECT NR+ band 1 on carrier 1657 with +17 dBm power and MCS1 modulation: v2.3.x

```
AT%XRFTEST=1,1,1,1657,17,10,5,1,0,0,0,0
%XRFTEST: 0
OK
```

The following command example disables the RF transmitter:

```
AT%XRFTEST=1,0
OK
```

4.24.1.1 TX burst mode

TX burst mode is enabled when <param10> is set to 1. The mode allows non-continuous TX transmission. The ON and OFF periods are based on the reference measurement channel uplink subframe scheduling that is defined for LTE-M in *3GPP TS 36.521 Annex A, Section A.2.2* and for NB-IoT in *3GPP TS 36.521 Annex A, Section A.2.4*. The burst signal is transmitted until the TX OFF command is given.

The following table describes the different waveforms supported by <param3>, <param4>, <param5>, and <param7>:

System mode <param3>	Modulation <param4>	RB/Tone count <param5>	Subcarrier spacing <param7>
NB-IoT (0)	0	1	1
	0	1	0
	0	3	0
	0	6	0
	0	12	0
	3	1	0
	3	1	1
LTE-M (1)	0, 1	1–6	0

Table 3: Allowed parameter combinations for <param3>, <param4>, <param5>, and <param7> in system modes NB-IoT and LTE-M when using TX burst mode

Response syntax when <operation> is ON and TX burst mode is enabled:

```
%XRFTEST: OK
```

The following command example enables the RF transmitter for Band 5, 830 MHz, +17 dBm, LTE-M, 16-QAM, 6 RB, RB start position 0, subcarrier spacing 15 kHz, system bandwidth 5 MHz, NB index 3, TX burst mode enabled:

```
AT%XRFTEST=1,1,5,8300,17,1,1,6,0,0,3,3,1
OK
```

4.24.2 Read command

The read command is not supported.

4.24.3 Test command

The test command is not supported.

4.25 GNSS SNR test %XRFTEST

The proprietary %**XRFTEST** command can be used to execute a *GNSS Signal-to-Noise Ratio (SNR)* test.

v2.2.x

The %**XRFTEST** command allows starting RF to a desired frequency to test, debug, and verify configurations that depend on RF frequency, such as %**XMAGPIO** and %**XMIPIRFFCTRL**.

4.25.1 Set command

The set command executes a *GNSS SNR* test.

GNSS L1 frequency is 1575.42 MHz. This test expects the *CW* in the signal generator to be 1575.750 MHz, which means that the offset is 330 kHz. The measurement duration is 1 ms.

Syntax:

```
%XRFTEST=<test>,<operation>,<param0>,<param1>,<param2>
```

The set command parameters and their defined values are the following:

<test>

2 – GNSS SNR

<operation>

1 – ON

Note: OFF <operation> is not needed due to automatic stop.

<param0>

RX signal power at the *SiP* GNSS port

Valid range from –127 dBm to –25 dBm or 0 = default gain –105 dBm

<param1>

AFC correction

1 – Enable AFC correction in measurement

0 – Disable AFC correction in measurement

<param2>

Frequency. Manually give RF frequency for nRF91x1 in 100 kHz format. Accepted value is +/- 10 MHz from the default GNSS frequency of 1575.42 MHz. Set the signal generator to $f = \text{<param2>} + 330 \text{ kHz}$.

Note: If <param2> is not given, nRF91x1 uses the default GNSS L1 frequency 1575.42 MHz.

Note: It is recommended to set the signal level defined in the GNSS SNR ON parameters to the expected signal level at the GNSS antenna port of the nRF91x1 SiP.

Response syntax:

```
%XRFTEST: <snr>,<antenna_power>,<sb2hnbr>,<headroom>
```

The response parameters and their defined values are the following:

<snr>

The result of the SNR measurement in q4 dB. q4 means that dividing the result by $2^4 = 16$ gives dB.

<antenna_power>

Measured signal power at the SiP GNSS port in q8 dBm. q8 means that dividing the result by $2^8 = 256$ gives dBm.

<sb2hnbr>

Signal bin to highest noise bin ratio in q4 dB.

Computed by dividing the signal bin by the power of the highest noise bin.

Allows to find spurious signals possibly affecting reception. The expected value is the same or higher than <snr>.

<headroom>

Time domain signal headroom in dBFullScale.

Expected value is -15 dBFS or -16 dBFS if the signal level in generator corresponds to the level given in <param2>.

The following command example executes a GNSS SNR test at -80 dBm level with AFC disabled:

```
AT%XRFTEST=2,1,-80,0
%XRFTEST: 364,-20506,455,-16
OK
```

Note: $364/16=22.75\text{dB}$. $-20506/256=-80.1\text{dBm}$. $455/16 = 28.4\text{dB}$.

The following command example executes a GNSS SNR test for -80 dBm. AFC correction is enabled and uses the result of the previous AFC measurement or AFC set. Instead of the default, the receiver frequency is set to 1575.0 MHz:

```
AT%XRFTEST=2,1,-80,1,15750
%XRFTEST: 364,-20506,455,-16
OK
```

4.25.2 Read command

The read command is not supported.

4.25.3 Test command

The test command is not supported.

4.26 RX SNR test %XRFTEST

The proprietary **%XRFTEST** command can be used to measure SNR. v2.2.x

The **%XRFTEST** command allows starting RF to a desired frequency to test, debug, and verify configurations that depend on RF frequency, such as **%XMAGPIO** and **%XMIPIRFFCTRL**.

4.26.1 Set command

The set command measures *SNR*.

To measure SNR correctly, the *CW* offset must be +330 kHz for the *LTE-M* and *DECT NR+* modes and +45 kHz for the *NB-IoT* mode.

Syntax:

```
%XRFTEST=<test>,<operation>,<param0>,<param1>,...,<param4>
```

The set command parameters and their defined values are the following:

<test>

3 – RX SNR.

<operation>

1 – ON.

Note: OFF <operation> is not needed due to automatic stop.

<param0>

3GPP band number. Use 24 for *GNSS* reception.

DECT NR+ bands 1, 2, 9, and 22 are supported when <param3> is 10. See *ETSI TS 103 636-2 V1.4.1 Table 5.2-1*.

<param1>

Valid range 6000–22000 (corresponds to 600.0 MHz–2200.0 MHz).

In *DECT NR+*, the valid range is 18800–19300 (corresponds to 1880.0 MHz–1930.0 MHz) depending on the *DECT NR+* band selected in <param0>.

<param2>

RX signal power at the SiP antenna port.

Valid range from –127 dBm to –25 dBm.

<param3>

System mode.

0 – NB-IoT.

1 – LTE-M.

10 – *DECT NR+*.

<param4>

Enabling AFC correction during SNR test.

1 – Enable.

0 – Disable.

If <param4> is not given, AFC correction is disabled.

Response syntax:

```
%XRFTEST: <snr>,<antenna_power>,<sb2hnbr>,<headroom>
```

The response parameters and their defined values are the following:

<snr>

Result of the SNR measurement in q4 dB. q4 means that dividing the result by $2^4 = 16$ dB.

The SNR calculation is done by computing the power of the signal bins and dividing it by the power of all the other bins.

<antenna_power>

Measured signal power at the *SiP* antenna port in q8 dBm. q8 means dividing the result by $2^8 = 256$ gives dBm.

<sb2hnbr>

Signal bin to highest noise bin ratio in q4 dB.

Computed by dividing the signal bin by the power of the highest noise bin.

Allows to find spurious signals possibly affecting reception. The expected value is the same or higher than <snr>.

<headroom>

Time domain signal headroom in dBFullScale.

Expected value is -15 dBFS or -16 dBFS if the signal level in generator corresponds to the level given in <param2>.

The following command example enables the RX SNR measurement and RF receiver for Band 1, 2140.0 MHz, -65 dBm, LTE-M mode, AFC correction enabled:

```
AT%XRFTEST=3,1,1,21400,-65,1,1
%XRFTEST: 496,-17002,598,-16
OK
```

Note: $496/16 = 31$ dB. $-17002/256 = -66.4$ dBm. $598/16=37.3$ dB.

4.26.2 Read command

The read command is not supported.

4.26.3 Test command

The test command is not supported.

4.27 AFC measurement and set %XRFTEST

The proprietary **%XRFTEST** command can be used to measure or set the AFC error value for a given frequency. v2.2.x

The **%XRFTEST** command allows starting RF to a desired frequency to test, debug, and verify configurations that depend on RF frequency, such as **%XMAGPIO** and **%XMIPIRFFCTRL**.

Some frequency error can occur because the signal generator's and nRF91x1's reference clocks are different. This test allows to measure or set the frequency error and use the result value as compensation in subsequent tests. This improves the correctness of *SNR* tests and transmission frequency, but has no significant effect on the power measurement results in the RX or TX tests. The relative Q30 formatted response value is temporarily stored to RAM. In nRF9131, the value can be stored to NVM with the **%XFSSYNC** command and used in RX and *GNSS* SNR tests if the AFC correction enabling parameter, for example `<param4>` in the RX SNR test, is 1.

The measurement expects the *CW* in the signal generator to be 120 kHz lower than the given frequency. For example, if DUT frequency is 830 MHz, the signal generator must be set to 829.880 MHz.

Note: The same result value is used automatically in the TX and RX tests.

4.27.1 Set command

The set command measures or sets the AFC error value for a given frequency. The default error value is 0.

The AFC measurement and set functions are described in the following sections.

4.27.1.1 AFC measurement

The AFC measurement function measures the AFC error value for a given frequency. The default error value is 0.

Syntax:

```
%XRFTEST=<test>,<operation>,<param0>,<param1>,<param2>,<param3>
```

The AFC measurement command parameters and their defined values are the following:

<test>

4 – AFC

<operation>

1 – ON (measure).

Note: OFF `<operation>` is not needed due to automatic stop. Mode is always *LTE-M*. The stored value is reset when the device is booted.

4 – Calibration. Measures frequency error. The measured value is stored to *Non-volatile Memory (NVM)* with the **%XFSSYNC** command. [nRF9131](#)

<param0>

3GPP band number. Use 24 for *GNSS* reception.

DECT NR+ bands 1, 2, 9, and 22 are supported when `<param3>` is 10. See *ETSI TS 103 636-2 V1.4.1 Table 5.2-1*.

<param1>

Frequency 100 kHz.

Valid range 6000–22000 (corresponds to 600.0 MHz–2200.0 MHz).

In *DECT NR+*, the valid range is 18800–19300 (corresponds to 1880.0 MHz–1930.0 MHz) depending on the *DECT NR+* band selected in `<param0>`.

Note: If *CW* is used, an offset of about 45 kHz for *NB-IoT* and 300 kHz for *LTE-M* is recommended.

<param2>

RX signal power at the *SiP* antenna port in dBm.

<param3>

System mode.

0 – NB-IoT.

1 – LTE-M.

10 – DECT NR+.

Response syntax:

```
%XRFTEST: <AFC_error_value_Hz>,<AFC_error_value_Q30>,<antenna_power>,<result_status>
```

The response parameters and their defined values are the following:

<AFC_error_value_Hz>

Measured AFC error value in Hz format. 0 if <result_status> is 1 (fail).

<AFC_error_value_Q30>

Measured AFC error value in relative signed Q30 format. 0 if <result_status> is 1 (fail).

<antenna_power>

Measured signal power at the *SiP* antenna port in q8 dBm. q8 means dividing the result by $2^8 = 256$ gives dBm.

<result_status>

Status of the AFC measurement.

0 – OK.

1 – Fail.

Failure is caused if the measured <antenna_power> (converted to dB from dB Q8) is not within -3 ... +5 dB of the input <param2>. This is to ensure result quality.

The following command example successfully measures the AFC error value for band 1, 2140.0 MHz, -40 dBm in LTE-M mode:

```
AT%XRFTEST=4,1,1,21400,-40,1
%XRFTEST:349,175,-10240,0
OK
```

4.27.1.2 AFC set

The AFC set function allows to manually set the AFC error value in Hz or relative Q30 format, which can be used in RX and GNSS SNR tests if the AFC correction enabling parameter, for example <param4> in the RX SNR test, is 1. The value set with this command overwrites the value obtained with AFC measurement.

This value is used automatically in the RX and TX tests. To not use the AFC error value, for example in TX or RX tests, set it to 0.

Syntax:

```
%XRFTEST=<test>,<operation>,<param0>,<param1>
```

The AFC set command parameters and their defined values are the following:

<test>

4 – AFC

<operation>

3 – SET

<param0>

AFC error value (Hz), if <param1>=0 or not given

AFC error value (relative signed Q30), if <param1>=1

<param1>

AFC error format

0 – AFC error value interpreted as Hz format

1 – AFC error value interpreted as relative signed Q30 format

Response syntax:

```
%XRFTEST: OK
```

The following command example sets AFC error value to 349 Hz:

```
AT%XRFTEST=4,3,349,0
OK
```

Note: In the AFC correction feature, if the set error value is given in Hz format, the relative signed Q30 value is automatically calculated when used in the next test for a given frequency. The relative error is used automatically in subsequent tests. The suitable error value can be calculated for any frequency with the following formula: $\text{error_in_hz_f2} = \text{error_in_hz_f1} * \text{f2} / \text{f1}$. For example, if error is measured as 120 Hz at 2140 MHz, the error at 1950 MHz is $120 \text{ Hz} * 1950 \text{ MHz} / 2140 \text{ MHz} = 109 \text{ Hz}$.

4.27.2 Read command

The read command is not supported.

4.27.3 Test command

The test command is not supported.

4.28 RF sensor test %XRFTEST

The proprietary **%XRFTEST** command allows reading the temperature sensor next to the *Power Amplifier (PA)* inside the *SiP* and another temperature sensor inside the *nRF9120 System on Chip (SoC)*. v2.2.x

The **%XRFTEST** command allows starting RF to a desired frequency to test, debug, and verify configurations that depend on RF frequency, such as **%XMAGPIO** and **%XMIPIRFFCTRL**.

4.28.1 Set command

The set command reads the temperature sensor next to the *PA* inside the *SiP* and another temperature sensor inside the *nRF9120 SoC*.

When the modem is active (either LTE communication or GNSS receiver), the **%XRFTEST** command with `<param0>=0` returns the latest PA temperature sensor value measured automatically during modem wakeup or transmission. The temperature is not measured during LTE or GNSS reception because PA is not in use. During modem inactivity, the modem measures the PA temperature value when the command is received.

The command with `<param0>=1` returns the SoC-internal temperature sensor value with the same restrictions that apply to the PA temperature measurement regarding modem activity. When the GNSS receiver is active, the SoC-internal temperature sensor value is updated more regularly. During LTE, the value is updated only at wakeup.

Syntax:

```
%XRFTEST=<test>,<operation>,<param0>
```

The set command parameters and their defined values are the following:

<test>

10 – Sensor test

<operation>

1 – ON (measure)

<param0>

0 – PA temperature

1 – SoC-internal temperature sensor

2 – Reserved

3 – VBAT (same as **%XVBAT**)

4 – Reserved

Response syntax when `<param0>=0` or `<param0>=1`:

```
%XRFTEST: <temperature>
OK
```

When `<param0>=0` or `<param0>=1`, the response is given in millidegrees Celsius, with a resolution of approximately 50 millidegrees Celsius.

Response syntax when `<param0>=3`:

```
%XRFTEST: <vbat>
OK
```

When `<param0>=3`, the response is given in millivolts.

The following command example measures the PA temperature which is approximately 20.4 Celsius:

```
AT%XRFTEST=10,1,0
%XRFTEST: 20355
OK
```

4.28.2 Read command

The read command is not supported.

4.28.3 Test command

The test command is not supported.

4.29 Modem GPIO functionality test %XGPIOTEST

The proprietary %**XGPIOTEST** tests the functionality of the modem's *GPIO* pins. v2.2.x

4.29.1 Set command

The set command allows the input and output control of the following modem *GPIO* pins: **SIM_INT**, **SIM_RST**, **SIM_CLK**, **SIM_IO**, **COEX0**, **COEX1**, and **COEX2**.

Syntax:

```
%XGPIOTEST=<direction>,<mask>,<pin_ctrl>
```

+CME ERROR code

518 – Not allowed in active state.

The set command parameters and their defined values are the following:

<direction>

0 – Output

1 – Input

<mask>

Bitmask for pins to control.

1 bit per pin in ascending pin order starting from bit 0.

Control for some pins is possible only in input or output direction. The allowed pin control directions and bit positions are the following:

SIM_INT – Input, LSB

SIM_RST – Output

SIM_CLK – Output

SIM_IO – Input/Output

COEX0 – Input/Output

COEX1 – Output

COEX2 – Input/Output, MSB

<pin_ctrl>

Bits for controlling each masked pin. If <direction> is output, 1 sets the pin high and 0 sets the pin low. If <direction> is input, 1 must be set for each pin whose state is read.

Control for some pins is allowed as either input or output. The allowed pin control directions and bit positions are the following:

SIM_INT – Input, LSB

SIM_RST – Output

SIM_CLK – Output

SIM_IO – Input/Output

COEX0 – Input/Output

COEX1 – Output

COEX2 – Input/Output, MSB

The following command example configures all pins as output and sets their state low:

```
AT%XGPIOTEST=0,127,0
%XGPIOTEST:
OK
```

The following command example sets all pins as input and reads the pin state of each pin:

```
AT%XGPIOTEST=1,127,127
%XGPIOTEST: 0x9
OK
```

4.29.2 Read command

The read command is not supported.

4.29.3 Test command

The test command is not supported.

4.30 File system sync to flash %XFSSYNC

The proprietary **%XFSSYNC** command synchronizes and stores AT command configurations to the modem's non-volatile memory. v2.2.x

4.30.1 Set command

The set command writes user-modified configuration data to the modem's *NVM* so that the data persists through the power off cycles

The command stores to NVM configurations that are set with the following commands:

- **%XBANDLOCK**
- **%XEMPR**
- **%XCOEX0**
- **%XMAGPIO**
- **%XMIPIRFFDEV**

- **%XMIPIRFFCTRL**
- **%XANTCFG**
- **%XCOEX2**
- **%XRFTEST** (AFC)
- **%TXSELFAL**

Syntax:

```
%XFSSYNC [= <mode> [, <rat> ]]
```

+CME ERROR code

518 – Not allowed in active state.

The set command parameters and their defined values are the following:

<mode>

0 – Store configurations to NVM.

1 – Store configurations to NVM and set **%XBANDLOCK** operation 3, **%XRFTEST** (AFC measurement and set **%XRFTEST**) operation 4, and **%TXSELFAL** permanently as read only. [nRF9131](#)

%XFSSYNC=1 fails if AFC calibration (**%XRFTEST**) and TX self-calibration (**%TXSELFAL**) have not been executed.

CAUTION: This is a one-time programmable command for **%XBANDLOCK** operation 3, **%XRFTEST** (AFC measurement and set **%XRFTEST**) operation 4, and **%TXSELFAL** that cannot be re-executed or reversed.

If the parameter is not given, <mode> value 0 is used.

<rat>

Bit mask for selecting device RAT.

1 – *LTE-M* and *NB-IoT*.

2 – *DECT NR+*.

3 – *LTE-M*, *NB-IoT*, and *DECT NR+*.

Required when <mode> is 1. If a matching **%TXSELFAL** command has not been executed with the correct mode settings, **%XFSSYNC** returns `ERROR`.

The following command example synchronizes and stores AT command configurations to the modem's NVM:

```
AT%XSSYNC
OK
```

4.30.2 Read command

The read command is not supported.

4.30.3 Test command

The test command is not supported.

4.31 MIPI RFFE interface access %XMIPIRFFEDBG

The proprietary **%XMIPIRFFEDBG** command can be used to directly access registers on a *MIPI RFFE* device. v2.2.x

Note: The **%XMIPIRFFEDBG** command is for testing purposes only.

Note: The **%XRFTEST** command can be used to start nRF91x1 to a frequency configured with the **%XMIPIRFFECTRL** command. The **%XMIPIRFFEDBG** command can be used to read the value from the MIPI RFFE device register to verify that the configured value has been correctly written to the register.

4.31.1 Set command

The set command provides direct read and write access to the registers of a *MIPI RFFE* device configured with the **%XMIPIRFFECTRL** command.

Syntax:

```
%XMIPIRFFEDBG=<dev_id>,<direction>,<address>,[<value>]
```

Response syntax:

```
%XMIPIRFFEDBG=<status>,<readvalue>
```

The set command parameters and their defined values are the following:

<dev_id>

The identification number of the *MIPI RFFE* device given when it was introduced using the **%XMIPIRFFEDEV** command.

<direction>

0 – Write to register.

1 – Read register.

<address>

Address of the MIPI RFFE register as a decimal number.

<value>

Value to be written to the register in decimal format. Applies only to the write command.

<status>

Indicates the status of MIPI RFFE access.

0 – Success.

1 – No response event from the MIPI RFFE peripheral.

2 – Timeout during read.

3 – Undefined error.

<readvalue>

Value read from MIPI RFFE device register. Returns the set value on register write.

The following command example reads register 0 of a MIPI RFFE device with device ID 1:

```
AT%XMIPIRFFEDBG=1,1,0
%XMIPIRFFEDBG: 0,2
OK
```

The following command example writes value 5 to register 37 of a MIPI RFFE device with device ID 1:

```
AT%XMIPIRFFEDBG=1,0,37,5
%XMIPIRFFEDBG: 0,5
OK
```

4.31.2 Read command

The read command is not supported.

4.31.3 Test command

The test command is not supported.

4.32 SIM interface functionality test %SIMTEST

The proprietary **%SIMTEST** command tests the activation and deactivation of the *Subscriber Identity Module (SIM)* card. v2.2.x

The command is used to test products that have a SIM card present during production when the **%XGPIOTEST** command cannot be used. In signaling modem firmware, use the **%XSIM** command instead of **%SIMTEST**.

4.32.1 Set command

The set command tests the activation and deactivation of the *SIM* card.

Syntax:

```
%SIMTEST
```

The set command parameter and its defined values are the following:

<result>

0 – SIM test passed

1 – SIM test failed

The following command example shows a passed test:

```
AT%SIMTEST
%SIMTEST: 0
OK
```

The following command example shows a failed test:

```
AT%SIMTEST
%SIMTEST: 1
OK
```

4.32.2 Read command

The read command is not supported.

4.32.3 Test command

The test command is not supported.

4.33 Low frequency jitter test %LFCLKTEST

The proprietary **%LFCLKTEST** command executes a jitter test on the low-frequency clock. v2.2.x

The command executes multiple measurements of the clock period and returns the maximum value, minimum value, and average value in clock ticks.

Note: *Pulse Width Modulation (PWM)* is set to power off for the duration of the command.

4.33.1 Set command

The set command executes a jitter test on the low-frequency clock.

Syntax:

```
%LFCLKTEST=<testcount>
```

Response syntax:

```
%LFCLKTEST: <average>, <max>, <min>
```

+CME ERROR code

518 – Not allowed in active state.

The set command parameters and their defined values are the following:

<testcount>

Number of iterations during the test. Range 20–1000.

Note: One <testcount> corresponds to ten clock period measurements. For example, <testcount> 60 means that 600 clock period measurements are done.

<average>

Average clock period of all measurements in clock ticks.

<max>

Highest clock period of all measurements in clock ticks.

<min>

Lowest clock period of all measurements in clock ticks.

The following command example executes a jitter test on the low-frequency clock:

```
AT%LFCLKTEST=60
%LFCLKTEST: 49127,49222,49121
OK
```


4.33.2 Read command

The read command is not supported.

4.33.3 Test command

The test command is not supported.

4.34 TX self-calibration and test %TXSELFAL

The proprietary **%TXSELFAL** command performs TX calibration. nRF9131 v2.2.x

Device Under Test (DUT) RF output must be connected to a 50 Ω load or terminator when executing TX self-calibration.

Calibration command fills an internal gain compensation table on DUT. This table can be reset with the clear command. The test command executes the self-calibration routine but does not update the gain compensation table.

Gain compensation table must be updated to DUT flash memory with the **%XFSSYNC** command.

When permanently limiting the used bands with **%XBANDLOCK=3** and then performing TX self-calibration with **%TXSELFAL=1**, execute **%XBANDLOCK=3** before **%TXSELFAL=1**. nRF9131

4.34.1 Set command

The set command

Syntax:

```
%TXSELFAL=<cmd>
```

Response syntax:

```
%TXSELFAL:<err>,<pwr>,<freq>
```

The set command parameters and their defined values are the following:

<cmd>

- 0 – Clear calibration values.
- 1 – Execute TX self-calibration.
- 2 – Execute TX self-test.

<err>

Maximum gain error of a single measurement on calibration sweep in mdB.
dB error is multiplied by 1000. The value can be positive or negative.

<pwr>

Requested TX power of the measurement when TX power error is the biggest.
The value can be positive or negative.

<freq>

Frequency of the worst measurement in kHz.

The following command example clears calibration values:

```
AT%TXSELFCAL=0
OK
```

The following command example performs self-calibration:

```
AT%TXSELFCAL=1
%TXSELFCAL:1438,22,1950000
```

The following command example executes self-test:

```
AT%TXSELFCAL=2
%TXSELFCAL:-125,15,1950000
OK
```

4.34.2 Read command

The read command is not supported.

4.34.3 Test command

The test command is not supported.

4.35 Write IMEI %IMEIWRITE

The proprietary **%IMEIWRITE** command writes *IMEI* to the *One Time Programmable (OTP) memory* of a device. v2.2.x

CAUTION: The command is a one-time operation. The IMEI value cannot be rewritten. The device can become unusable if the wrong IMEI is written.

4.35.1 Set command

The set command writes *IMEI* to the *OTP memory* of a device.

The command responds with `OK` and the device's IMEI value if the command syntax is correct. Syntax error returns `ERROR`.

If IMEI has already been written earlier, the IMEI field is not updated. When writing IMEI, the IMEI in the return value can be used to check if the sent IMEI was accepted and cannot be used for other devices.

Syntax:

```
%IMEIWRITE=<operation_type>,<imei_string>
```

The set command parameters and their defined values are the following:

<operation type>

- Operation type.
- 0 – Write.

<imei_string>

- String in decimal format indicating the IMEI. Composed of *TAC* (8 digits), *SNR* (6 digits), and *CD* (1 digit).

The following command example writes IMEI to the device memory:

```
AT%IMEIWRITE=0,"123456789123458"  
%IMEIWRITE: 123456789123458  
OK
```

4.35.2 Read command

The read command is not supported.

4.35.3 Test command

The test command is not supported.

Glossary

16-state Quadrature Amplitude Modulation (16-QAM)

A digital modulation technique used for signals in which four bits are modulated at once by selecting one of 16 possible combinations of carrier phase shift and amplitude.

AT command

A command used to control the modem.

Binary Phase-Shift Keying (BPSK)

A digital modulation technique used for signals in which one bit is modulated by selecting one of two possible carrier phase shifts with a 180-degree difference.

Carrier Wave (CW)

A single-frequency electromagnetic wave that can be modulated in amplitude, frequency, or phase to convey information.

Cat-NB1

NB-IoT *User Equipment (UE)* category with 200 kHz UE bandwidth and a single RX antenna, specified in 3GPP Release 13.

Cat-NB2

An upgraded version of *Cat-NB1*, specified in 3GPP Release 14.

Check Digit (CD)

The last one-digit number of the *IMEI* code used for error detection.

DECT NR+

A non-cellular radio standard included as part of the 5G standards by the ITU.

Device Under Test (DUT)

A manufactured product undergoing testing.

Electronic Serial Number (ESN)

A unique number embedded on a microchip for identifying mobile devices.

Global Navigation Satellite System (GNSS)

A satellite navigation system with global coverage. The system provides signals from space transmitting positioning and timing data to GNSS receivers, which use this data to determine location.

General-Purpose Input/Output (GPIO)

A digital signal pin that can be used as input, output, or both. It is uncommitted and can be controlled by the user at runtime.

International Mobile (Station) Equipment Identity (IMEI)

A unique code consisting of 14 digits and a check digit for identifying 3GPP-based mobile devices.

International Mobile (Station) Equipment Identity, Software Version (IMEISV)

A unique code consisting of 16 decimal digits and two software version digits for identifying 3GPP-based mobile devices.

Internet Protocol (IP)

The network layer communications protocol in the Internet protocol suite for relaying datagrams across network boundaries. Its routing function enables internetworking, and essentially establishes the Internet.

International Reference Alphabet (IRA)

A seven-bit coded character set for information exchange.

Low-Noise Amplifier (LNA)

In a radio receiving system, an electronic amplifier that amplifies a very low-power signal without significantly degrading its signal-to-noise ratio.

Long-Term Evolution (LTE)

A wireless broadband communication standard for mobile devices and data terminals, based on the GSM/EDGE and UMTS/HSPA technologies.

LTE-M

An open standard that is most suitable for medium throughput applications requiring low power, low latency, and/or mobility, like asset tracking, wearables, medical, POS, and home security applications. Also known as Cat-M1.

MIPI RF Front-End Control Interface (RFFE)

A dedicated control interface for the RF front-end subsystem. [MIPI Alliance](#)

Narrowband Internet of Things (NB-IoT)

A narrowband technology standard with longer range, lower throughput, and better penetration in, for example, cellars and parking garages compared to LTE-M. NB-IoT is most suitable for static, low throughput applications like smart metering, smart agriculture, and smart city applications. Also known as *Cat-NB1* or *Cat-NB2*.

Non-volatile Memory (NVM)

Memory that can retrieve stored information even after having been power-cycled.

One Time Programmable (OTP) memory

A type of non-volatile memory that permits data to be written to memory only once.

Power Amplifier (PA)

A device used to increase the transmit power level of a radio signal.

Printed Circuit Board (PCB)

A board that connects electronic components.

Production Test Image (PTI)

A modem firmware version used in the device manufacturing phase.

Pulse Width Modulation (PWM)

A form of modulation used to represent an analog signal with a binary signal where the switching frequency is fixed, and all the pulses corresponding to one sample are contiguous in the digital signal.

Quadrature Phase-Shift Keying (QPSK)

A digital modulation technique used for signals in which two bits are modulated at once, selecting one of four possible carrier phase shifts.

Resource Block (RB)

The smallest unit of resources that can be allocated to a user.

Serial Number (SNR)

A unique six-digit number part of the *IMEI* code identifying each equipment within each *TAC*.

Signal-to-Noise Ratio (SNR)

The level of signal power compared to the level of noise power, often expressed in decibels (dB).

Software Version Number (SVN)

Part of the *IMEI* code identifying the revision of the software installed on a mobile device.

Subscriber Identity Module (SIM)

A card used in *UE* containing data for subscriber identification.

System in Package (SiP)

Several integrated circuits, often from different technologies, enclosed in a single module that performs as a system or subsystem.

System on Chip (SoC)

A microchip that integrates all the necessary electronic circuits and components of a computer or other electronic systems on a single integrated circuit.

Type Allocation Code (TAC)

The initial eight-digit part of an *IMEI* code used for identifying the model of a mobile phone.

User Equipment (UE)

Any device used by an end-user to communicate. The UE consists of the Mobile Equipment (ME) and the Universal Integrated Circuit Card (UICC).

Unique Slave Identifier (USID)

A unique address for identifying each slave device in an RFFE (RF Front-End) system.

Universal Subscriber Identity Module (USIM)

A card used in *UE* containing data for subscriber identification.

UUID

Universally Unique Identifier

Legal notices

By using this documentation you agree to our terms and conditions of use. Nordic Semiconductor may change these terms and conditions at any time without notice.

Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function, or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

Nordic Semiconductor ASA does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. If there are any discrepancies, ambiguities or conflicts in Nordic Semiconductor's documentation, the Product Specification prevails.

Nordic Semiconductor ASA reserves the right to make corrections, enhancements, and other changes to this document without notice.

Life support applications

Nordic Semiconductor products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury.

Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

RoHS and REACH statement

Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website www.nordicsemi.com.

Trademarks

All trademarks, service marks, trade names, product names, and logos appearing in this documentation are the property of their respective owners.

Copyright notice

© 2024 Nordic Semiconductor ASA. All rights are reserved. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

