AN14125

Manufacturing Software Development Kit API Specification

Rev. 2.0 — 28 February 2025 Application note

Document information

Information	Content
Keywords	Labtool, Software Development Kit (SDK), API, annexes, calibration sequences, RSSI, power, crystal, TX/RX, storage, Bluetooth/Bluetooth Low Energy (LE), Wi-Fi, 802.15.4, manufacturing
Abstract	Details how to use the manufacturing SDK API to automate Wi-Fi, Bluetooth, and 802.15.4 radio calibration.

1 About this document

1.1 Purpose

This document explains the Labtool Software Development Kit (SDK) used to automate manufacturing test programs. The SDK contains Application Programming Interfaces (APIs) and a list of data elements (annexes). The APIS are called in sequence for Wi-Fi, Bluetooth/Bluetooth LE, 802.15.4 calibrations. The annexes are used to store calibration data.

1.2 Supported products

- AW611 [15]
- IW610 [16]
- IW611 [17]
- IW612 [18]
- RW610 [19]
- RW612 [20]

1.3 System requirements

The manufacturing SDK requires:

- A x86 based Windows computer with Ethernet connection and a local storage device (hard driver or flash driver).
- Windows 10 or Windows 11 operating system (OS)
- A software development IDE such as Microsoft Visual Studio 2022 or newer versions.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

2 Labtool SDK

2.1 Download the SDK

To download the Labtool SDK:

- Go to the web page of the supported device. See [15], [16], [17], [18], [19], and [20].
- Click **Design Resources** (Figure 1).

Figure 1. Design resources tab on the product page

- Sign in to access secure files
- Click **Software** in the Design resource section (Figure 2).

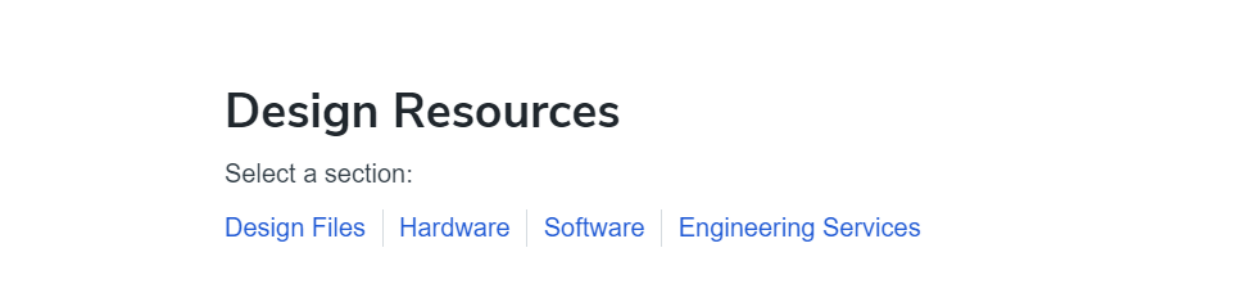

Figure 2. Design resources section on the product page

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- Select **Test, Debug, Analyzer Software** (Figure 3).

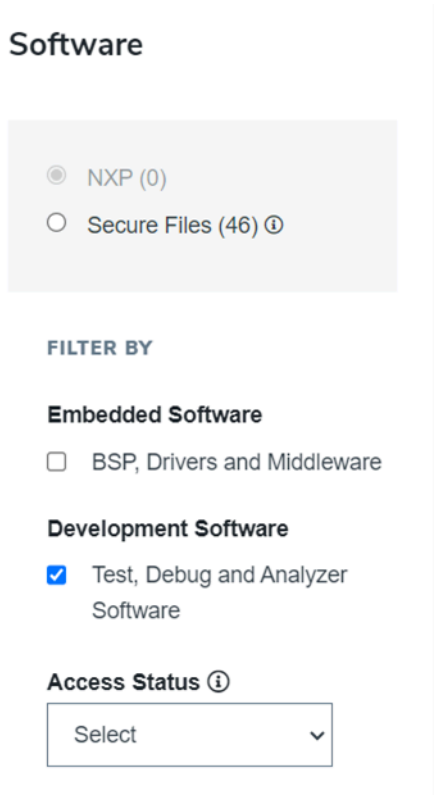

Figure 3. Software items

- Look for the latest Labtool SDK and click **Download** (Figure 4).

Figure 4. Labtool SDK

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

3 Calibration data

Calibration data includes Wi-Fi, Bluetooth, and IEEE 802.15.4 (IW612 and RW612 only) configuration and calibration parameters.

Calibration data is design specific and includes the following:

Wi-Fi

- Wi-Fi MAC address
- Crystal frequency calibration
- Wi-Fi RSSI calibration
- Wi-Fi transmit power calibration
- Thermal compensation for Wi-Fi transmit power
- Thermal compensation for crystal frequency
- RF front-end control configuration
- Wi-Fi system configuration

Bluetooth/Bluetooth Low Energy (LE)

- Initial Bluetooth transmit power
- Bluetooth TX Power Class
- Bluetooth RF front end (FE) Loss
- Bluetooth device (BD) address
- Bluetooth frequency calibration
- Bluetooth UART baud rate

802.15.4 (IW612 and RW612 only)

- 802.15.4 maximum power limit
- EUI 64-bit MAC address
- SPI clock frequency
- 802.15.4 front-end (FE) loss

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

4 Flowchart

Figure 5 illustrates the back-end of Labtool application. On Labtool applications, users input commands which call the APIs in sequence. If the values are within range, the API action is performed. The API data can be stored in the device storage or annex. Else, the user is prompted again for a valid command.

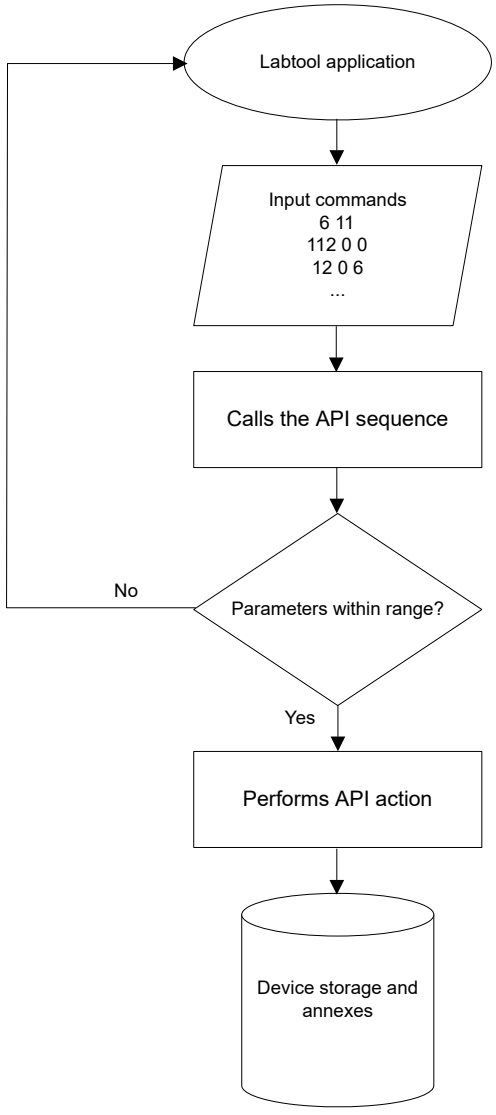

Figure 5. Flowchart of Labtool back-end

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5 Wi-Fi sequences

This section describes the API sequence to calibrate a Wi-Fi radio with example parameters. Read more about the Wi-Fi subsystem API in [1].

5.1 Initialize the Wi-Fi

Before calibrating, the Wi-Fi DUT must be initialized. Other radios are turned off to ensure there is no interference with Wi-Fi operations.

- To connect to the Wi-Fi DUT:

```
DutIf_InitConnection(NULL)
```

- To connect to Bluetooth DUT:

```
Dut_Bt_OpenDevice (NULL)
```

- To connect to 15.4 DUT:

```
Dut_15_4_OpenDevice (NULL)
```

- To disconnect the Bluetooth DUT:

```
Dut_Bt_CloseDevice(NULL)
```

- To disconnect 802.15.4 DUT:

```
Dut_15_4_CloseDevice(NULL)
```

- To disconnect the Wi-Fi DUT:

```
DutIf_Disconnection(NULL)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5.2 Configure the calibration data storage

Calibration data can be stored in the EEPROM, the configuration file, or the on-chip OTP memory.

- Set to EEPROM

```
Dut_Shared_SetStorageType(0, 0)
```

- Set to calibration *.conf* file

```
Dut_Shared_SetStorageType(1, 0)
```

- Set to OTP

```
Dut_Shared_SetStorageType(2, 0)
```

5.3 Load Wi-Fi calibration data

Calibration data can be stored in a configuration file.

- Load the default Wi-Fi hardware configuration file

```
DutIf_SetCustomizedSettings(NULL)
```

5.4 Set the thermal compensation for the crystal frequency

The Wi-Fi thermal crystal frequency compensation parameter is used to maintain the crystal frequency accuracy across the operating temperature range.

A third order polynomial equation is used for the crystal frequency parameter:

$$y(t) = A \times X^3 + B \times X^2 + C \times X + D$$

Where:

- Y is the crystal frequency offset
- X is the thermal sensor (Tsen) reading
- A is the third order coefficient
- B is the second order coefficient
- C is the first order coefficient
- D is the constant coefficient

To calculate the average frequency offset delta, refer to the section *Wi-Fi thermal crystal frequency compensation* in [2] and [3].

Store the values in annex 83.

```
Dut_Shared_SetAnnexType83Data(ThirdOrderCoefficient, SecondOrderCoefficient,
 FirstOrderCoefficient, ConstantCoefficient)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5.5 Set the thermal TX power compensation

The Wi-Fi thermal power compensation parameter is used to maintain the transmit power accuracy across the operating temperature range.

The power compensation parameter is expressed as a slope of a linear equation for power correction. As the temperature changes, the slope compensation is applied to maintain the transmit power accuracy.

The formula for TX power thermal slope is shown below:

$$Thermal\ \ slope = \frac{[(\text{Output Pwr @Hot})-(\text{Output Pwr @Cold})]\times 1000}{[(\text{Tsens @Hot})-(\text{Tsens @Cold})]}$$

Where:

• Tsen is the temperature sensor reading.

• Output Pwr is the Tx power measured at the RF connector of the DUT.

To calculate the thermal slope, refer to the section *Wi-Fi thermal Tx power compensation* in [2] and [3].

Store the values in annex 75.

```
Dut_Shared_SetAnnexType75Data_Rev2(int DUTIndex, int PathId, int Rev, int NumOfEnterirs,
 Thermal_POWER_RSSI_COMPENSSTION_VE *ThermalPowerRSSIData)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5.6 Crystal frequency calibration

The frequency calibration is used to calibrate the frequency accuracy when an external crystal is used as a reference clock source.

The frequency calibration process involves tuning the load capacitance value integrated in the wireless SoC using the crystal compensation parameter `RFXTAL`.

To measure the frequency change with the crystal calibration-offset change:

- Start the calibration mode.

```
DutIf_SetRfControlMode(0x10, 0)
```

- Set the radio band and antenna mode.

```
DutIf_SetRadioMode_Ext_W909X(mode0, mode1, 0)
```

- Set the channel bandwidth to 20 MHz.

```
DutIf_SetChannelBw(0, 0)
```

- Set the radio channel.

```
DutIf_SetRfChannel_new(channel, 0, 0)
```

- Set the crystal calibration offset. Refer to DutIf_InitConnection .

```
DutIf_SetRfXTal(extension, cal, 0)
```

- Start TX for power index 4.

```
DutIf_AdjustPcktSifs11ax_W909X(1, dataRate, pattern, length,
Short_Preamble, Bssid, 0, 0, 0, 0, 0, 0, NULL, 4, 0, 0,
0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
0xffffffff, 0xffffffff, 0xffffffff, 0)
```

- Measure and record the frequency using the VSA.
- Increase the crystal offset value by one step. Refer to DutIf_InitConnection .

```
DutIf_SetRfXTal(extension, cal, 0)
```

- Measure and record the frequency using the VSA.
- Repeat the transmission and increment the crystal offset for at least 10 frequencies.

To measure the frequency error with the change of temperature:

- Start the calibration mode.

```
DutIf_SetRfControlMode(0x10, 0)
```

- Set the radio band and antenna mode.

```
DutIf_SetRadioMode_Ext_W909X(mode0, mode1, 0)
```

- Set the channel bandwidth to 20 MHz.

```
DutIf_SetChannelBw(0, 0)
```

- Set the radio channel.

```
DutIf_SetRfChannel_new(channel, 0, 0)
```

- Set the crystal calibration offset. Refer to DutIf_InitConnection .

```
DutIf_SetRfXTal(extension, cal, 0)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- Start TX for power index 4.

```
DutIf_AdjustPcktSifs11ax_W909X(1, dataRate, pattern, length,
Short_Preamble, Bssid, 0, 0, 0, 0, 0, 0, NULL, 4, 0, 0,
0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
0xffffffff, 0xffffffff, 0xffffffff, 0)
```

- Measure and record the frequency using the VSA.
- Get the thermal sensor reading.

```
DutIf_GetThermalSensorReading_W909X(SensorIndex, *pReadBack)
```

- Stop TX.

```
DutIf_AdjustPcktSifs11ax_W909X(0, dataRate, pattern, length,
Short_Preamble, Bssid, 0, 0, 0, 0, 0, 0, NULL, 4, 0, 0,
0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
0xffffffff, 0xffffffff, 0xffffffff, 0)
```

- Repeat the above process over the operating temperature range.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5.7 Calibrate RSSI

The Wi-Fi RSSI calibration is used to tune the reported signal strength referenced at the antenna connector. The delta between the DUT read-back value and a tester signal level is stored as an RSSI offset. The offset is used to calibrate the reported signal strength.

Each subband has four RSSI offset entries for each LNA gain state:

- **ELNA_LO_ILNA_HI_Mode** is for external-low and internal-high LNA gain state.
- **ELNA_HI_ILNA_HI_Mode** is for external-high and internal-high LNA gain state.
- **ELNA_LO_ILNA_LO_Mode** is for external-low and internal-low LNA gain state.
- **ELNA_HI_ILNA_LO_Mode** is for external-high and internal-low LNA gain state.

RSSI is calibrated on one channel per subband.

Based on the reference design, an external LNA is not used. The internal LNA high and low gain states are linear. As a result, one value is used for every gain state.

To calibrate RSSI:

- Set the radio band and antenna mode.

```
DutIf_SetRadioMode_Ext_W909X(mode0, mode1, 0)
```

- Set the channel bandwidth.

```
DutIf_SetChannelBw(0, 0)
```

- Set the radio channel.

```
DutIf_SetRfChannel_new(channel, 0, 0)
```

- Start RX test

```
DutIf_WlanRSSI_Cal_Start(0)
```

- Transmit Wi-Fi packets from a tester with -50 dBm power level OFDM-6 signal
- Stop RX and get report

```
DutIf_WlanRSSI_Cal_Stop(RSSI_Val, RSSI_Nf, Rssi_packet_count, LNAGainMode, 0)
```

- Calculate RSSI offset

```
RSSI Offset = INT(2*( Tester Tx power value - RSSI readback power value ))
```

  Where:
  `Tester Tx power value` is the RF tester signal level
  `RSSI readback power value` is the RSSI value read from `DutIf_WlanRSSI_Cal_Stop`
  RF tester repeats the steps four times for each gain state: ELNA_LO_ILNA_HI_Mode, ELNA_HI_ILNA_HI_Mode, ELNA_LO_ILNA_LO_Mode and ELNA_HI_ILNA_LO_Mode RX modes
- Store RSSI offset in annex 98

```
Dut_Shared_ReadAnnex98DataFromFile_W909X(char *FileName, int DUTIndex, int PathId, int
 *NumOfEnterirs, RSSI_CAL_SE *RSSIData)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5.8 Calibrate TX power

The Wi-Fi transmit power calibration is used to tune the Wi-Fi transmit power for accurate output power at the antenna connector.

The transmit power calibration process involves measuring the output power at the antenna connector for eight power indexes. Each power index corresponds to a specific output power.

The transmit power is measured for each subband. A maximum of two 20 MHz channels per subband are allowed.

Table 1 lists the recommended calibration channels.

Table 1. Recommended RSSI calibration channels

Band	Subband	RSSI calibration channel
2.4 GHz	0	2437 MHz
5 GHz	1	5180 MHz
	2	5600 MHz
	3	5825 MHz

To calibrate TX power:

- Start the calibration mode.

```
DutIf_SetRfControlMode(0x10, 0)
```

- Set the radio band and antenna mode.

```
DutIf_SetRadioMode_Ext_W909X(mode0, mode1, 0)
```

- Set the channel bandwidth to 20 MHz.

```
DutIf_SetChannelBw(0, 0)
```

- Set the radio channel.

```
DutIf_SetRfChannel_new(channel, 0, 0)
```

- Start TX for power index 0.

```
DutIf_AdjustPcktSifs11ax_W909X(1, dataRate, pattern, length,
Short_Preamble, Bssid, 0, 0, 0, 0, 0, 0, NULL, 0, 0, 0,
0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
0xffffffff, 0xffffffff, 0xffffffff, 0)
```

- Measure and record the output power.
- Stop TX.

```
DutIf_AdjustPcktSifs11ax_W909X(0, dataRate, pattern, length,
Short_Preamble, Bssid, 0, 0, 0, 0, 0, 0, NULL, 0, 0, 0,
0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
0xffffffff, 0xffffffff, 0xffffffff, 0)
```

- Repeat start/stop TX, measure, and record output power process for power indexes 1 – 5.

```
DutIf_AdjustPcktSifs11ax_W909X(1, dataRate, pattern, length,
Short_Preamble, Bssid, 0, 0, 0, 0, 0, 0, NULL, powerIndex, 0, 0,
0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff, 0xffffffff,
0xffffffff, 0xffffffff, 0xffffffff, 0)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

- At power index 5, collect the power detect offset and temperature for annex 97.

```
DutIf_GetPwrDetOffset(*PwrReadingIndBm, *pPwrDetOffsetIndB, *pOffsetCode, PathId, 0)
DutIf_GetThermalSensorReading_W909X(SensorIndex, *pReadBack)
```

- Repeat start/stop TX, measure, and record the output power process for power indexes 6 and 7.
- Write to Annex 97

```
Dut_Shared_SetAnnexType97Data_W909X(DUTIndex, PathId, *CalOption, NumOfEnterirs,
 *TX_PowerData, *BM_Data, TraTempThr)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

5.9 Store the data

During the Wi-Fi TX/RX calibration process, the RF test App collects the values of the crystal, TX power, power detector reading offset and RSSI offset. Next, the RF test App stores the calibration data to annex 97 (Wi-Fi TX power table) and annex 98 (Wi-Fi RSSI calibration data).

The RF test App keeps the data values for the main structure and the other annexes in a configuration database.

- Set the main structure annex.

```
Dut_Shared_SetCalMainDataRevF_V3()
```

- Set the annex data for each annex.

```
Dut_Shared_SetAnnexType<NUM>Data_W909X()
```

- Program the annex data to storage.

```
DutIf_SetCalDataRevF()
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6 Wi-Fi APIs

6.1 List of Wi-Fi APIs

Table 2. List of Wi-Fi APIs

Command
Connection
DutIf_InitConnection
DutIf_Disconnection
Calibration file
DutIf_SetCustomizedSettings
DutIf_SetCalDataRevF
Channel bandwidth
DutIf_SetChannelBw
RF channel
DutIf_SetRfChannel_new
Crystal adjustment
DutIf_SetRfXTal
Power control
DutIf_SetRfControlMode
Power detect reading offset
DutIf_GetPwrDetOffset
Thermal reading
DutIf_GetThermalSensorReading_W909X
Duty-cycle Transmit
DutIf_AdjustPcktSifs11ax_W909X
DutIf_AdjustPcktSifs_W909X
DutIf_SetTxContMode_W90XX
Receive
DutIf_EnableBssidFilter
DutIf_WlanRSSI_Cal_Start
DutIf_WlanRSSI_Cal_Stops

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.2 Data rates

Refer to the section *Data rates* in [10], [11], [12], [13], and [14].

6.3 RF channels

Refer to the section *Wi-Fi RF channels* in [4],[5], [6], [7], [8], and [9].

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.4 DutIf_InitConnection

This API opens a Wi-Fi connection.

```
DUT_WLAN_API int STDCALL DutIf_InitConnection(void **theObj, int PortNum)
```

Table 3. Command parameters

Parameter	Type	Description
theObj	void	Pointer to object
PortNum	int	Port number to establish connection

Return parameters

0 = InitConnection Success

Nonzero = InitConnection Failed

6.5 DutIf_Disconnection

This API closes Wi-Fi connection.

```
DUT_WLAN_API int STDCALL DutIf_Disconnection(void ** theObj)
```

Table 4. Command parameters

Parameter	Type	Description
theObj	void	Pointer to object

Return parameters

0 = Connection terminated

Nonzero = Connection not terminated

6.6 DutIf_SetChannelBw

This API sets the channel bandwidth.

```
DUT_WLAN_API int STDCALL DutIf_SetChannelBw(int ChannelBw,
DWORD DeviceId)
```

Table 5. Command parameters

Parameter	Type	Description
ChannelBW	int	Channel bandwidth of Wi-Fi
DeviceId	Double word	Device ID 0 = Default

Return parameters

0 = SetChannelBw success

Nonzero = SetChannelBw failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.7 DutIf_SetRfChannel_new

This API sets the Wi-Fi RF channel.

```
DUT_WLAN_API int STDCALL DutIf_SetRfChannel_new(int channel, int chanOffset, DWORD
 DeviceId)
```

Table 6. Command parameters

Parameter	Type	Description
`channel`	int	Channel bandwidth of Wi-Fi
`chanOffset`	int	Channel offset
`DeviceId`	Double word	Device ID 0 = Default

Return parameters

0 = SetRfChannel_new success

Nonzero = SetRfChannel_new failed

6.8 DutIf_SetRfXTal

This API sets RF crystal cap code.

```
DUT_WLAN_API int STDCALL DutIf_SetRfXTal(int Extension,int Cal,
DWORD DeviceId)
```

Table 7. Command parameters

Parameter	Type	Description
`Extension`	int	Extension value
`cal`	int	Calibration data value
`DeviceId`	Double word	Device ID 0 = Default

Return parameters

0 = SetRfXTal success

Nonzero = SetRfXTal failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.9 DutIf_SetRfControlMode

This API sets Wi-Fi DUT to regular or power calibration mode. Regular mode takes power specified in the Wi-Fi packet TX API to set RF power. Calibration mode takes the power index specified in the Wi-Fi packet TX API to set RF power.

```
DUT_WLAN_API int STDCALL DutIf_SetRfControlMode(DWORD Mode, DWORD DeviceId)
```

Table 8. Command parameters

Parameter	Type	Description
`Mode`	Double word	0 = Regular mode 0x10 = Calibration mode
`DeviceId`	Double word	Device ID 0 = Default

Return parameters

0 = SetRfControlMode success

Nonzero = SetRfControlMode failed

6.10 DutIf_GetPwrDetOffset

This API returns power detection offset.

```
DUT_WLAN_API int STDCALL DutIf_GetPwrDetOffset(float PwrReadingIndBm, float
 *pPwrDetOffsetIndB, signed char *pOffsetCode, DWORD PathId, DWORD DeviceId)
```

Table 9. Command parameters

Parameter	Type	Description
`PwrReadingIndBm`	float	Power reading in dBm
`pPwrDetOffsetIndB`	float	Power detection offset in db
`pOffsetCode`	Signed char	Pointer to offset code
`PathID`	Double word	ID of the RF transmission path 0 = Path A (SISO chipsets) 1 = Path B (MIMO chipsets)
`DeviceId`	Double word	Device ID 0 = Default

Return parameters

0 = GetPwrDetOffset return success

Nonzero = GetPwrDetOffset return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.11 DutIf_GetThermalSensorReading_W909X

This API returns the device thermal sensor reading.

```
DUT_WLAN_API int STDCALL DutIf_GetThermalSensorReading_W909X(int SensorIndex, int
 *pReadBack)
```

Table 10. Command parameters

Parameter	Type	Description
`SensorIndex`	Int	Sensor index 0 = CAU 1 = Radio0_PathA 2 = Radio0_PathB
`pReadBack`	Int	Pointer to readback sensor reading value

Return parameters

0 = GetThermalSensorReading_W909X return success

Nonzero = GetThermalSensorReading_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.12 DutIf_AdjustPcktSifs11ax_W909X

This API adjusts the packet for short interframe spacing (SIFS).

```
DUT_WLAN_API int STDCALL DutIf_AdjustPcktSifs11ax_W909X(DWORD Enable,
DWORD dataRate, DWORD pattern, DWORD length, int ShortPreamble, char *Bssid, DWORD
 ShortGI, DWORD AdvCoding, DWORD TxBfOn, DWORD GFMode, DWORD STBC, DWORD DPDEnable, Float
 *PowerLevel, DWORD SignalBw, DWORD AdjustTxBurstGap, DWORD BurstSifsInUs, DWORD NumPkt,
 DWORD MaxPktExt, DWORD BeamChange, DWORD Dcm, DWORD Doppler,DWORD MidamblePeriod, DWORD
 QNum, DWORD DeviceId)
```

Table 11. Command parameters

Parameter	Type	Description
`Enable`	Double word	Enable/Disable 0 = disable 1 = enable
`dataRate`	Double word	Data rate (Section 6.2)
`Pattern`	Double word	4-bytes payload data
`Length`	Double word	Payload length
`ShortPreamble`	Int	Short preamble value -1 = default, unless otherwise stated For legacy 802.11b: 0 = long 1 = short For legacy 802.11g: Not valid Set to -1 For 802.11n: Greenfield PPDU indicator 0 = HT-mix 1 = HT-GF For 11ac: Not valid Set to -1 For 11ax: PPDU type 0 = HE-SU 1 = HE-EXT-SU 2 = HE-MU 3 = HE-Triggered-based
`Bssid`	char	Pointer to BSSID
`ShortGI`	Double word	0 = long GI 1 = short GI
`AdvCoding`	Double word	-1 = default 0 = disable advanced coding 1 = enable advanced coding
`TxBfOn`	Double word	0 = disable TX beamforming 1 = enable TX beamforming
`GFMode`	Double word	0 = disable Greenfield 1 = enable Greenfield

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 11. Command parameters*...continued*

Parameter	Type	Description
STBC	Double word	Set 2 bits of Space-time block code data
DPDEnable	Double word	Digital predistortion (DPD) 0 = disable DPD training 1 = enable DPD training
PowerLevel	float	Power level in dBm If in calibration mode, power step index
SignalBw	Double word	Signal bandwidth value -1 = default 0 = 20 MHz 1 = 40 MHz 3 = 5 MHz 4 = 80 MHz
AdjustTxBurstGap	Double word	Adjust TX burst gap value 0 = default SIF gap 1 = set SIF gap
BurstSifsInUs	Double word	SIF gap in microseconds
NumPkt	Double word	Number of packets to send 0xffffffff = duty cycle TX
MaxPktExt	Double word	Maximum Packet -1 = use current value 0 = 0 us 8 = 8 us 16 = 16 us
BeamChange	Double word	Beam exchange 0 = disable 1 = enable
Dcm	Double word	DCM value 0 = disable DCM 1 = enable DCM
Doppler	Double word	Doppler value 0 = disable 1 = enable
MidamblePeriod	Double word	Midamble period 10 or 20 = value -1 = use current value in firmware
QNum	Double word	Queue number (0-12)(17-20) -1 = to use FW selected queue
DeviceId	Double word	Device ID 0 = Default

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Return parameters

0 = AdjustPcktSifs11ax_W909X return success

Nonzero = AdjustPcktSifs11ax_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.13 DutIf_AdjustPcktSifs_W909X

This API adjusts the packet for short interframe spacing (SIFS) for legacy 802.11ax.

```
DUT_WLAN_API int STDCALL DutIf_AdjustPcktSifs_W909X(DWORD Enable, DWORD dataRate, DWORD
 pattern, DWORD length, int ShortPreamble, char *Bssid, DWORD ShortGI, DWORD AdvCoding,
 DWORD TxBfOn, DWORD GFMode, DWORD STBC, DWORD DPDEnable, float *PowerLevel, DWORD
 SignalBw, DWORD AdjustTxBurstGap, DWORD BurstSifsInUs, DWORD DeviceId)
```

Table 12. Command parameters

Parameter	Type	Description
`Enable`	Double word	Enable/Disable 0 = disable 1 = enable
`dataRate`	Double word	Data rate (Section 6.2)
`Pattern`	Double word	4-bytes payload data
`Length`	Double word	Payload length
`ShortPreamble`	Int	Short preamble value -1 = default, unless otherwise stated For legacy 802.11b: 0 = long 1 = short For legacy 802.11g: Not valid Set to -1 For 802.11n: Greenfield PPDU indicator 0 = HT-mix 1 = HT-GF For 802.11ac: Not valid Set to -1 For 11ax: PPDU type 0 = HE-SU 1 = HE-EXT-SU 2 = HE-MU 3 = HE-Triggered-based
`Bssid`	char	Pointer to BSSID
`ShortGI`	Double word	0 = long GI 1 = short GI
`AdvCoding`	Double word	-1 = default 0 = disable advanced coding 1 = enable advanced coding
`TxBfOn`	Double word	0 = disable TX beamforming 1 = enable TX beamforming
`GFMode`	Double word	0 = disable Greenfield 1 = enable Greenfield
`STBC`	Double word	Set 2 bits of space-time block code data

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 12. Command parameters*...continued*

Parameter	Type	Description
`DPDEnable`	Double word	Digital predistortion (DPD) 0 = disable DPD training 1 = enable DPD training
`PowerLevel`	float	Power level in dBm If in calibration mode, power step index
`SignalBw`	Double word	Signal bandwidth value -1 = default 0 = 20 MHz 1 = 40 MHz 3 = 5 MHz 4 = 80 MHz
`AdjustTxBurstGap`	Double word	Adjust TX burst gap value 0 = default SIF gap 1 = set SIF gap
`BurstSifsInUs`	Double word	SIF gap in microseconds
`DeviceId`	Double word	Device ID 0 = Default

Return parameters

0 = AdjustPcktSifs_W909X return success

Nonzero = AdjustPcktSifs_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.14 DutIf_SetTxContMode_W90XX

This API sets the TX continuous mode.

```
DUT_WLAN_API int STDCALL DutIf_SetTxContMode_W90XX(BOOL enable, BOOL CWmode, DWORD
 dataRate, DWORD *PowerID, DWORD payloadPattern, DWORD CSmode, DWORD, ActSubCh, DWORD
 DeviceId)
```

Table 13. Command parameters

Parameter	Type	Description
`Enable`	Bool	Enable/disable 0 = disable 1 = enable
`CWmode`	Bool	CW mode Set to 1
`dataRate`	Double word	Data rate (Section 6.2)
`PowerID`	Double word	Pointer to PowerID Set to NULL
`PayloadPattern`	Double word	Payload pattern value Set to 0
`CSmode`	Double word	Carrier sense (CS) mode Set to 0
`ActSubch`	Double word	Act subchannel value Set to -1
`DeviceId`	Double word	Device ID 0 = Default

Return parameters

0 = SetTxContMode return success

Nonzero = SetTxContMode return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.15 DutIf_EnableBssidFilter

This API enables BSSID filter.

```
DUT_WLAN_API int STDCALL DutIf_EnableBssidFilter(int mode, BYTE *Bssid, char *SSID, DWORD
 DeviceId)
```

Table 14. Command parameters

Parameter	Type	Description
Mode	int	BSS mode Set to 0
BSSID	Byte	Pointer to BSSID
SSID	char	Pointer to SSID Set to NULL
DeviceId	Double word	Device ID 0 = Default

Return parameters

0 = EnableBssidFilter success

Nonzero = EnableBssidFilter failed

6.16 DutIf_WlanRSSI_Cal_Start

This API starts Wi-Fi RSSI calibration.

```
DUT_WLAN_API int STDCALL DutIf_WlanRSSI_Cal_Start(DWORD DeviceId)
```

Table 15. Command parameters

Parameter	Type	Description
DeviceId	Double word	Device ID 0 = Default

Return parameters

0 = WlanRSSI_Cal_Start success

Nonzero = WlanRSSI_Cal_Start failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

6.17 DutIf_WlanRSSI_Cal_Stops

This API stops Wi-Fi RSSI calibration.

```
DUT_WLAN_API int STDCALL DutIf_WlanRSSI_Cal_Stop(WORD RSSI_Val[MAXNUM_RXPATH], WORD
 RSSI_Nf[MAXNUM_RXPATH], DWORD Gain_Region[MAXNUM_RXPATH], DWORD *pRssi_packet_count,
 DWORD DeviceId);
```

Table 16. Command parameters

Parameter	Type	Description
RSSI_Val	word	RSSI value
RSSI_Nf	Word	Noise floor reading
Gain_Region	Double word	LNA gain mode
pRssi_packet_count	Double word	Pointer to RSSI packet count
DeviceId	Double word	Device ID 0 = Default

Return parameters

0 = WlanRSSI_Cal_Stop return success

Nonzero = WlanRSSI_Cal_Stop return failed

6.18 DutIf_SetCustomizedSettings

This API loads the default calibration data filename.

```
DUT_WLAN_API int STDCALL DutIf_SetCustomizedSettings(char *FileName)
```

Table 17. Command parameters

Parameter	Type	Description
FileName	char	Wi-Fi calibration data filename Set to NULL for the default filename

Return parameters

0 = DutIf_SetCustomizedSettings return success

Nonzero = DutIf_SetCustomizedSettings return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

7 Bluetooth/Bluetooth LE sequences

This section describes the API sequence to calibrate a Bluetooth/Bluetooth LE radio. Refer to Bluetooth/ Bluetooth LE APIs for API details.

7.1 Initialize the DUT

Before calibrating, the Bluetooth DUT must be initialized. Other radios are turned off to ensure there is no interference with Bluetooth operations.

- To connect to Bluetooth/Bluetooth LE DUT.

```
Dut_Bt_OpenDevice(NULL)
```

- To disconnect the 802.15.4 DUT.

```
Dut_15_4_CloseDevice(NULL)
```

- To disconnect the Wi-Fi DUT.

```
DutIf_Disconnection(NULL)
```

- To disconnect the Bluetooth DUT.

```
Dut_Bt_CloseDevice(NULL)
```

7.2 Reset the DUT

To reset the Bluetooth/Bluetooth LE DUT:

```
DutIf_SetBtBrfReset_W909X(0)
```

7.3 Load Bluetooth calibration data

Calibration data can be stored in a configuration file.

- Load the default Bluetooth hardware configuration file.

```
Dut_Bt_ReloadCalData_W909X(0)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

7.4 Calibrate TX power

Bluetooth front-end loss (FE loss) is used to tune the Bluetooth/Bluetooth LE transmit power (with step size of 0.5 dB) at the antenna connector.

Increasing FE Loss results in increasing the output power from the radio. Similarly, decreasing FE Loss decreases the output power from the radio.

To calibrate TX power and calculate FE loss:

- Prepare a Bluetooth Calibration data file with FE Loss = 0.
- Clear FE loss by loading the Bluetooth Calibration data file.

```
Dut_Bt_ReloadCalData_W909X(0)
```

- Disable power control.

```
Dut_Bt_SetBtDisableBtuPwrCtl_W909X9(1)
```

- Set the Bluetooth class.

```
Dut_Bt_SetBtPwrControlClass_W909X(option)
```

- Set the power level.

```
Dut_Bt_SetBtPwrLvlValue_W909X(powerLevelDB, 0)
```

- Transmit.

```
Dut_Bt_TxBtBTUPwrDutyCycleHop(True, TxPwrLvldBm, PacketType, payloadPattern,
 payloadLenghtInByte, HopMode, interval, Whitening, *pChannelAccess, 0)
```

- Calculate the FE loss.

$$FE\_Loss = int(2 \times (set\_power - meas\_power))$$

- Store FE loss as 8-bit number in annex 55.

```
Dut_Shared_SetAnnexType55Data_Rev3(Version, RFXtal, InitPwr, FELoss, ForceClass2Op,
 Class1OpSupport, DisablePwrControl, MiscFlag, UsedInternalSleepClock,
 AOALocaltionSupport, NumberOfAntennas, Rssi_Golden_Lo,  Rssi_Golden_Hi,  BTBAUDRate,
 BDAddr[6], Encr_Key_Len_Max,  Encr_Key_Len_Min,  RegionCode)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

7.5 Calibrate the crystal

Similar to Wi-Fi thermal crystal frequency compensation, Bluetooth/Bluetooth LE can calibrate the crystal frequency accuracy.

To measure the frequency change with the crystal calibration-offset change:

- Set a test RF channel.

```
Dut_Bt_SetBtChannel_W909X(channel, 0, 0)
```

- Set the initial RF crystal cap code.

```
Dut_Bt_SetBtXtal_W909X(External mode, XtalValue, 0)
```

- Start TX.

```
Dut_Bt_TxBtBTUPwrDutyCycleHop(True,  TxPwrLvldBm, PacketType, payloadPattern,
 payloadLenghtInByte, HopMode, interval, Whitening, *pChannelAccess, 0)
```

- Check the center frequency offset on the RF tester. If the center frequency is within range, skip this step. Else, adjust the RF crystal cap code until the center frequency is within range.
- Stop the TX and note down the crystal cap code for annex 55.

```
Dut_Bt_TxBtBTUPwrDutyCycleHop(False,  TxPwrLvldBm, PacketType, payloadPattern,
 payloadLenghtInByte, HopMode, interval, Whitening, *pChannelAccess, 0)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

7.6 Verify Bluetooth TX

To verify Bluetooth TX:

- Set the class mode.

```
Dut_Bt_GetBtPwrControlClass_W909X(option, 0)
```

- Set the channel.

```
Dut_Bt_SetBtChannel_W909X(channel, 0, 0)
```

- Start TX.

```
Dut_Bt_TxBtBTUPwrDutyCycleHop(True,  TxPwrLvldBm, PacketType, payloadPattern,
 payloadLenghtInByte, HopMode, interval, Whitening, char *pChannelAccess, 0)
```

- Stop TX.

```
Dut_Bt_TxBtBTUPwrDutyCycleHop(False,  TxPwrLvldBm, PacketType, payloadPattern,
 payloadLenghtInByte, HopMode, interval, Whitening, char *pChannelAccess, 0)
```

- Repeat the above steps for different band, channel bandwidth, channel, and TX data rates. and check TX performance.

7.7 Verify Bluetooth RX

To verify Bluetooth RX:

- Connect the DUT to Bluetooth tester.
- Set the test for RF channel.

```
Dut_Bt_SetBtChannel_W909X(channel, 0, 0)
```

- Start RX test.

```
Dut_Bt_MrvlRxTest_W909X(RxChannel, PacketNumber, PacketType, payloadPattern,
 PayloadLenghtInByte, TxBdAddress, 0)
```

- Send out Bluetooth waveform to the DUT.
- Stop RX test and check RX report.

```
Dut_Bt_MrvlRxTestResult_Ext_W909X()
```

- Repeat the above steps for different channels and/or data rates, and check RX performance.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

7.8 Verify Bluetooth LE TX

To verify Bluetooth LE TX:

- Set the class mode.

```
Dut_Bt_GetBtPwrControlClass_W909X(option, 0)
```

- Set TX power.

```
Dut_Bt_BleWriteTxPower_W909X(Tx power, 0)
```

- Specify the frequency index (channel) and physical data rate, and start TX.

```
Dut_Bt_BleEnhancedTxTest_W909X(Freq index, Phy, 0, 0)
```

- Check Bluetooth LE power, spectrum, and the center frequency offset.
- Stop TX

```
Dut_Bt_BleTestEnd_W909X(pRxedPckCnt, 0)
```

- Repeat the above steps for different channels, power, TX data rates, and check TX performance.

7.9 Verify Bluetooth LE RX

To verify Bluetooth LE RX:

- Specify the frequency index (channel), the physical data rate, and start RX.

```
Dut_Bt_BleEnhancedRxTest_W909X(Freq index, Phy, 0, 0)
```

- Send out Bluetooth LE waveform to the DUT.
- Stop RX

```
Dut_Bt_BleTestEnd_W909X(pRxedPckCnt, 0)
```

- Check RX report.

```
Dut_Bt_BleGetRxTestBer_W909X(pCrcErrCnt, pNoCorrelationCnt, 0)
```

- Repeat the above steps for different channels and/or data rates, and check RX performance.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

7.10 Store the data

The application stores the cap code value for the DUT in the main structure and annex 55.

The RF test application keeps the data values for the main structure annex and other annexes in a calibration database.

- Set the main structure annex.

```
Dut_Shared_SetCalMainDataRevF_V3()
```

- Set the annex data for each annex.

```
Dut_Shared_SetAnnexType<NUM>Data_W909X()
```

- Program the annex data to storage.

```
DutIf_SetCalDataRevF()
DutIf_SetCalDataRevFBTOnly()
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8 Bluetooth/Bluetooth LE APIs

8.1 List of Bluetooth APIs

Table 18. List of Bluetooth APIs

Commands
Connection
Dut_Bt_OpenDevice
Dut_Bt_CloseDevice
Reset
Dut_Bt_SetBtBrfReset_W909X
Power
Dut_Bt_SetBtPwrControlClass_W909X
Dut_Bt_SetBtPwrLvlValue_W909X
DutIf_SetRfControlMode
Channel
Dut_Bt_SetBtChannel_W909X
Crystal adjustment
Dut_Bt_SetBtXtal_W909X
Dut_Bt_WriteBrfRegister
Calibration file
Dut_Bt_ReloadCalData_W909X
DutIf_SetCalDataRevFBTOnly
Duty-cycle transmit
Dut_Bt_TxBtBTUPwrDutyCycleHop
Dut_Bt_BleWriteTxPower_W909X
Dut_Bt_BleEnhancedTxTest_W909X
Dut_Bt_TxBtCw_W909X
Receive
Dut_Bt_MrvlRxTest_W909X
Bt_MrvlRxTestResult_Ext_W909X
Dut_Bt_BleEnhancedRxTest_W909X
Dut_Bt_BleGetRxTestBer_W909X
Dut_Bt_BleTestEnd_W909X

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.2 Dut_Bt_OpenDevice

This API opens the Bluetooth connection.

```
DUT_BT_API int STDCALL Dut_Bt_OpenDevice(void **theObj, int PortNum)
```

Table 19. Command parameters

Parameter	Type	Description
`theObj`	void	Pointer to object Set to NULL
`PortNum`	int	Port number to establish connection

Return parameters

0 = InitConnection Success

Nonzero = InitConnection Failed

8.3 Dut_Bt_CloseDevice

This API closes Bluetooth connection.

```
DUT_BT_API int STDCALL Dut_Bt_CloseDevice(void **theObj)
```

Table 20. Command parameters

Parameter	Type	Description
`theObj`	void	Pointer to object Set to NULL

Return parameters

0 = Dut_Bt_CloseDevice Connection terminated

Nonzero = Dut_Bt_CloseDevice Connection not terminated

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.4 Dut_Bt_SetBtBrfReset_W909X

This API resets Bluetooth.

```
DUT_WLAN_API int STDCALL DutIf_SetBtBrfReset_W909X(DWORD DeviceId)
```

Table 21. Command parameters

Parameter	Type	Description
DeviceId	Double word	Device ID 0 = Default

Return parameters

0 = SetBtBrfReset_W909X return success

Nonzero = SetBtBrfReset_W909X return failed

8.5 Dut_Bt_SetBtXtal_W909X

This API sets the Bluetooth crystal cap code.

```
DUT_BT_API int STDCALL Dut_Bt_SetBtXtal_W909X(BYTE ExtMode,BYTE XtalValue, DWORD
 DeviceId)
```

Table 22. Command parameters

Parameter	Type	Description
ExtMode	byte	External mode
XtalValue	byte	Crystal cap code value
DeviceID	double word	Device ID 0 = Default

Return parameters

0 = SetBtXtal return success

Nonzero = SetBtXtal return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.6 Dut_Bt_SetBtChannel_W909X

This API sets the Bluetooth channel.

```
DUT_BT_API int STDCALL Dut_Bt_SetBtChannel_W909X(int channelNum,bool BT2, DWORD DeviceId)
```

Table 23. Command parameters

Parameter	Type	Description
channelNum	int	Channel number
BT2	Bool	Bluetooth 2 True/False Set to 0
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = SetBtChannel_W909X return success

Nonzero = SetBtChannel_W909X return failed

8.7 Dut_Bt_SetBtPwrControlClass_W909X

This API sets the Bluetooth power control.

```
DUT_BT_API int STDCALL Dut_Bt_SetBtPwrControlClass_W909X(DWORD option, DWORD DeviceId)
```

Table 24. Command parameters

Parameter	Type	Description
option	Double word	Power level option 0 = class 2 1 = class 1.5 3 = class 1
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = SetBtPwrControlClass_W909X return success

Nonzero = SetBtPwrControlClass_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.8 Dut_Bt_TxBtBTUPwrDutyCycleHop

This API sets the Bluetooth transmit power duty cycle.

```
DUT_BT_API int STDCALL Dut_Bt_TxBtBTUPwrDutyCycleHop(bool enable,  int TxPwrLvldBm, int
 PacketType, int payloadPattern, int payloadLenghtInByte, bool HopMode, int interval, int
 Whitening, unsigned char *pChannelAccess, DWORD DeviceId)
```

Table 25. Command parameters

Parameter	Type	Description
enable	Bool	Start/Stop TX True = start TX False = stop TX
TxPwrLvldBm	Int	Transmit Power level in dBm
PacketType	Int	Rate and time slot GFSK, 1M FEC 0x01 = DM1 0x03 = DM3 0x05 = DM5 GFSK, 1M 0x11 = DH1 0x13 = DH3 0x15 = DH5 DQPSK, 2M 0x21 = 2-DH1 0x23 = 2-DH3 0x25 = 2-DH5 8PSK, 3M 0x31 = 2-DH1 0x33 = 3-DH3 0x35 = 3-DH5
payloadPattern	Int	Payload pattern 0 = all 1 = all 2 = PN9 3 = 0xAA 4 = 0xF0
PayloadLenghtInByte	Int	Payload length in bytes Set to -1 for maximum payload length
HopMode	Bool	Enable/disable channel hopping 0 = disable channel hopping 1 = enable channel hopping
Interval	Int	Interval
Whitening	Int	Enable/disable whitening 0 = disable whitening 1 = enable whitening
pChannelAccess	Unsigned char	10-byte array to specify the channel mask

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 25. Command parameters*...continued*

Parameter	Type	Description
`DeviceID`	Double word	Device ID 0 = Default

Return parameters

0 = TxBtBTUPwrDutyCycleHop return success

Nonzero = TxBtBTUPwrDutyCycleHop return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.9 Dut_Bt_TxBtCw_W909X

This API sets continuous transmit.

```
DUT_BT_API int STDCALL Dut_Bt_TxBtCw_W909X(bool enable, DWORD DeviceId)
```

Table 26. Command parameters

Parameter	Type	Description
enable	Bool	Start/Stop CW TX True = start CW TX False = stop CW TX
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = Dut_Bt_TxBtCw_W909X return success

Nonzero = Dut_Bt_TxBtCw_W909X return failed

8.10 Dut_Bt_BleWriteTxPower_W909X

This API writes the transmit power value.

```
DUT_BT_API int STDCALL Dut_Bt_BleWriteTxPower_W909X(int TxPwr,DWORD DeviceId)
```

Table 27. Command parameters

Parameter	Type	Description
TxPower	int	TX power level in dBm
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = Dut_Bt_TxBtCw_W909X return success

Nonzero = Dut_Bt_TxBtCw_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.11 Dut_Bt_BleEnhancedTxTest_W909X

This API returns the Bluetooth LE enhanced transmit test.

```
DUT_BT_API int STDCALL Dut_Bt_BleEnhancedTxTest_W909X(int FreqIndex, int LenOfTxData, int
 PayloadPattern, int Phy, DWORD DeviceId)
```

Table 28. Command parameters

Parameter	Type	Description
`FreqIndex`	int	Bluetooth LE channel index Range 0 – 39 for channels 0-39
`LenOfTxData`	Int	Payload size 0-37 bytes
`PayloadPattern`	Int	Payload pattern 0 = PN9 1 = 0xF0 2 = 0xAA 3 = PN15 4 = all 1 5 = all 0 6 = 0x0F 7 = 0x55
`Phy`	Int	Physical data rate 1 = 1 Mbit/s 2 = 2 Mbit/s 3 = LR8 4 = LR2
`DeviceID`	Double word	Device ID 0 = Default

Return parameters

0 = BleEnhancedTxTest_W909X return success

Nonzero = BleEnhancedTxTest_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.12 Dut_Bt_MrvlRxTest_W909X

This API tests RX.

```
DUT_BT_API int STDCALL Dut_Bt_MrvlRxTest_W909X(int RxChannel, int PacketNumber, int
 PacketType, int payloadPattern, int PayloadLenghtInByte, char TxBdAddress, DWORD
 DeviceId)
```

Table 29. Command parameters

Parameter	Type	Description
RxChannel	int	RX channel
PacketNumber	Int	Packet number
PacketType	Int	Rate and time slot GFSK, 1M FEC 0x01 = DM1 0x03 = DM3 0x05 = DM5 GFSK, 1M 0x11 = DH1 0x13 = DH3 0x15 = DH5 DQPSK, 2M 0x21 = 2-DH1 0x23 = 2-DH3 0x25 = 2-DH5 8PSK, 3M 0x31 = 2-DH1 0x33 = 3-DH3 0x35 = 3-DH5
payloadPattern	Int	Payload pattern 0 = all 1 = all 2 = PN9 3 = 0xAA 4 = 0xF0
payloadLenghtInByte	Int	Payload length in bytes Set to -1 for maximum payload length
TxBdaddress	Char	BD address of the transmit device
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = MrvlRxTest_W909X return success

Nonzero = MrvlRxTest_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.13 Bt_MrvlRxTestResult_Ext_W909X

This API returns the Bluetooth extended NXP RX test result.

```
DUT_BT_API int STDCALL Dut_Bt_MrvlRxTestResult_Ext_W909X(BYTE* Report_Status, DWORD*
 Report_TotalPcktCount, DWORD* Report_NoRxCount, DWORD* Report_Correlation, DWORD*
 Report_HecMatchCount, DWORD* Report_HecMatchCrcPckts, DWORD* Report_HdrErrCount,
 DWORD* Report_CrcErrCount, DWORD* Report_TotPktRx, DWORD* Report_PktNoError,
 DWORD* Report_DropPkt, FLOAT* Report_PER, DWORD* Report_TotalBitsExp, DWORD*
 Report_TotalBitsError, FLOAT* Report_BER, DWORD* Report_TotalByteCount, DWORD*
 Report_BitErrorRx, DWORD* Report_AvgRssi, DWORD* Report_Max, DWORD DeviceId)
```

Table 30. Command parameters

Parameter	Type	Description
Report_Status	Byte*	Return RX session status 0 = completed 1 = aborted
Report_TotalPcktCount	Double word*	Number of packets expected to receive
Report_NoRxCount	Double word*	Return number of packets missed during RX session
Report_Correlation	Double word*	Return successful full correlation counts
Report_HecMatchCount	Double word*	Return successful RX HEC match counts
Report_HecMatchCrcPckts	Double word*	Return successful HEC match counts
Report_HdrErrCount	Double word*	Return successful RX header error counts
Report_CrcErrCount	Double word*	Return RX CRC error counts
Report_TotPktRx	Double word*	Return the number of packets received during an RX session
Report_PktNoError	Double word*	Return the number of packets received during an RX session with no error
Report_DropPkt	Double word*	Return the number of packets received during an RX session
Report_PER	Double word*	Return PER rate
Report_TotalBitsExp	Double word*	Number of bits expected to receive
Report_TotalBitsError	Double word*	Return number of bit errors
Report_BER	FLOAT*	Return BER rate
Report_TotalByteCount	Double word*	Return the total bytes received during an RX session
Report_BitErrorRx	Double word*	Return but error counts

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 30. Command parameters*...continued*

Parameter	Type	Description
`Report_AvgRssi`	Double word*	Return average RSSI in dBm
`Report_Max`	Double word*	Reserved
`DeviceID`	Double word	Device ID 0 = Default

Return parameters

0 = MrvlRxTestResult_Ext_W909X return success

Nonzero = MrvlRxTestResult_Ext_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.14 Dut_Bt_BleEnhancedRxTest_W909X

This API returns the Bluetooth LE RX test.

```
DUT_BT_API int STDCALL Dut_Bt_BleEnhancedRxTest_W909X(int FreqIndex, int Phy, int
 ModIndex, DWORD DeviceId)
```

Command parameters

Table 31. Command parameters

Parameter	Type	Description
FreqIndex	Int	Bluetooth LE channel index
Phy	Int	Physical data rate 1 = 1 Mbit/s 2 = 2 Mbit/s 3 = LR8 4 = LR2
ModIndex	int	Mode index value Set to 0
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = BleEnhancedRxTest_W909X return success

Nonzero = BleEnhancedRxTest_W909X return failed

8.15 Dut_Bt_BleTestEnd_W909X

This API returns the Bluetooth LE test end.

```
DUT_BT_API int STDCALL Dut_Bt_BleTestEnd_W909X(int *pRxedPckCnt, DWORD DeviceId)
```

Table 32. Command parameters

Parameter	Type	Description
pRxedPckCnt	Int	return number of packet received
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = BleTestEnd_W909X return success

Nonzero = BleTestEnd_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.16 Dut_Bt_BleGetRxTestBer_W909X

This API returns the Bluetooth LE TX test.

```
DUT_BT_API int STDCALL Dut_Bt_BleGetRxTestBer_W909X(int *pCrcErrCnt, int
 *pNoCorrelationCnt, DWORD DeviceId)
```

Table 33. Command parameters

Parameter	Type	Description
pCrcErrCnt	Int	Return CRC error count
pNoCorrelationCnt	Int	Return missing correlation count
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = BleGetRxTestBer_W909X return success

Nonzero = BleGetRxTestBer_W909X return failed

8.17 Dut_Bt_ReloadCalData_W909X

This API loads the Bluetooth calibration data.

```
DUT_BT_API int STDCALL Dut_Bt_ReloadCalData_W909X(DWORD DeviceId = 0)
```

Table 34. Command parameters

Parameter	Type	Description
DeviceID	Double word	Device ID 0 = Default

Return parameters

0 = Dut_Bt_ReloadCalData_W909X return success

Nonzero = Dut_Bt_ReloadCalData_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

8.18 Dut_Bt_WriteBrfRegister

This API writes the BRF register.

```
DUT_BT_API int STDCALL Dut_Bt_WriteBrfRegister(DWORD address,WORD *data,DWORD DeviceID )
```

Table 35. Command parameters

Parameter	Type	Description
Address	Double word	Register address to write
Data	word	Data to write in address
DeviceID	Double word	Device ID 0 = default

Return parameters

0 = Dut_Bt_WriteBrfRegister return success

Nonzero = Dut_Bt_WriteBrfRegister return failed

8.19 Dut_Bt_SetBtPwrLvlValue_W909X

This API sets the Bluetooth power level value.

```
DUT_BT_API int STDCALL Dut_Bt_SetBtPwrLvlValue_W909X(double PwrLvlValueDB,DWORD IsEDR,
 DWORD DeviceId)
```

Table 36. Command parameters

Parameter	Type	Description
PwrLvlValueDB	double	Power level in dB
isEDR	Double word	EDR data
DeviceID	Double word	Bluetooth/Bluetooth LE device ID 0 = default

Return parameters

0 = Dut_Bt_WriteBrfRegister return success

Nonzero = Dut_Bt_WriteBrfRegister return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

9 802.15.4 radio sequences

This section describes the API sequence to calibrate 802.15.4 radio. Refer to 802.15.4 APIs for API details.

9.1 Initialize the DUT

Before calibrating, the 802.15.4 DUT must be initialized.

- To connect to 802.15.4 DUT:

```
Dut_15_4_OpenDevice(NULL)
```

- To disconnect 802.15.4 DUT:

```
Dut_15_4_CloseDevice(NULL)
```

9.2 Reset the DUT

To reset 802.15.4 DUT:

```
Dut_15_4_ResetMCU(0)
```

9.3 Configure the data calibration storage

The calibration data can be stored in the EEPROM, the configuration file, or the on-chip OTP memory.

- Set to EEPROM.

```
Dut_Shared_SetStorageType(0, 0)
```

- Set to the calibration configuration file.

```
Dut_Shared_SetStorageType(1, 0)
```

- Set to the OTP.

```
Dut_Shared_SetStorageType(2, 0)
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

9.4 Verify 802.15.4 TX

To verify 802.15.4 TX:

- Set the RF channel.

```
Dut_15_4_SetChannel(channel, 0)
```

- Set TX power level.

```
Dut_15_4_SetPwrLvl(PwrLevel, 0)
```

- Start TX.

```
Dut_15_4_TxDutycycle(0, 0)
```

- Check 802.15.4 TX power, spectrum, and center frequency offset.
- Run 802.15.4 RX test, and check RX performance and RSSI accuracy.
- Disable TX.

```
Dut_15_4_TxDutycycle(1, 0)
```

- Repeat the above steps for different channels and TX data rates, and check TX performance.

9.5 Verify 802.15.4 RX

To verify 802.15.4 RX:

- Set the RF channel.

```
Dut_15_4_SetChannel(channel, 0)
```

- Start RX test

```
Dut_15_4_StartRxTest(0)
```

- Send 802.15.4 waveform to the DUT.
- Check 802.15.4 RX test result.

```
Dut_15_4_RxTestResult(BYTE * Status,DWORD * RxTotalPcktCount,DWORD * RxPcktCount,int *
 RSSI,0)
```

- Repeat the above steps for different channels and check RX performance.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

9.6 Set EUI-64 MAC address

To set the EUI-64 MAC address:

```
Dut_15_4_SetEUI64MacAddress(EUI64MacAddress,0)
```

To get the EUI-64 MAC address:

```
Dut_15_4_SetEUI64MacAddress(EUI64MacAddress, 0)
```

9.7 Store the data

RF test App should maintain a configuration database with the data values for the main structure annex and other annexes, so the data can be programmed to the calibration storage.

- Set the main structure annex.

  ```
  Dut_Shared_SetCalMainDataRevF_V3()
  ```

- Set the annex data for each annex.

  ```
  Dut_Shared_SetAnnexType<NUM>Data_W909X()
  ```

- Program the annex data to the calibration storage.

  ```
  DutIf_SetCalDataRevF15p4Only()
  ```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

10 802.15.4 radio APIs

10.1 List of 802.15.4 APIs

Table 37. List of 802.15.4 radio APIs

Commands
Connection
Dut_15_4_OpenDevice
Dut_15_4_CloseDevice
Reset
Dut_15_4_ResetMCU
Power level
Dut_15_4_SetPwrLvl
Channel
Dut_15_4_SetChannel
Calibration file
DutIf_SetCalDataRevF15p4Only
Duty-cycle transmit
Dut_15_4_TxDutycycle
Dut_15_4_TxBurst
Receive
Dut_15_4_StartRxTest
Dut_15_4_RxTestResult
EUI-64 address
Dut_15_4_GetEUI64MacAddress
Dut_15_4_SetEUI64MacAddress

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

10.2 Dut_15_4_OpenDevice

This API opens 802.15.4 connection.

```
DUT_15_4_API int STDCALL Dut_15_4_OpenDevice(void ** theObj, int PortNum )
```

Table 38. Command parameters

Parameter	Type	Description
`theObj`	void	Pointer to the object
`PortNum`	Int	802.15.4 port number

Return parameters

0 = Dut15_4_OpenDevice return success

Nonzero = Dut15_4_OpenDevice return failed

10.3 Dut_15_4_CloseDevice

This API closes 802.15.4 connection.

```
DUT_15_4_API int STDCALL Dut_15_4_CloseDevice(void ** theObj)
```

Table 39. Command parameters

Parameter	Type	Description
`theObj`	void	Pointer to the object

Return parameters

0 = Dut15_4_CloseDevice return success

Nonzero = Dut15_4_CloseDevice return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

10.4 Dut_15_4_ResetMCU

This API reset the 802.15.4 radio.

```
DUT_ResetMCUDUT_15_4_API int STDCALL Dut_15_4_ResetMCU(DWORD DeviceId)
```

Table 40. Command parameters

Parameter	Type	Description
Device ID	Double word	Device ID NULL = default

Return parameters

0 = Dut_15_4_ResetMCU return success

Nonzero = Dut_15_4_ResetMCU return failed

10.5 Dut_15_4_SetChannel

This API sets the channel number.

```
DUT_SetChannel DUT_15_4_API int STDCALL Dut_15_4_SetChannel(int ChannelNum, DWORD
 DeviceId )
```

Table 41. Command parameters

Parameter	Type	Description
ChannelNum	int	Channel number
DeviceId	Double word	Device ID Default = 0

Return parameters

0 = Dut_15_4_SetChannel return success

Nonzero = Dut_15_4_SetChannel return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

10.6 Dut_15_4_SetPwrLvl

This API sets the 802.15.4 power level.

```
DUT_15_4_API int STDCALL Dut_15_4_SetPwrLvl(int PwrLvl,DWORD DeviceId )
```

Table 42. Command parameters

Parameter	Type	Description
PwrLvl	int	Power level value
DeviceId	Double word	Device ID Default = 0

Return parameters

0 = Dut_15_4_SetPwrLvl return success

Nonzero = Dut_15_4_SetPwrLvl return failed

10.7 Dut_15_4_TxDutycycle

This API sets the 802.15.4 TX duty cycle.

```
DUT_15_4_API int STDCALL Dut_15_4_TxDutycycle(bool Enable, DWORD DeviceId)
```

Table 43. Command parameters

Parameter	Type	Description
Enable	bool	Enable/Disable transmit 0 = start duty cycle TX 1 = stop TX
DeviceId	Double word	Device ID Default = 0

Return parameters

0 = Dut_15_4_TxDutycycle return success

Nonzero = Dut_15_4_TxDutycycle return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

10.8 Dut_15_4_TxBurst

This API performs TX test in burst mode.

```
DUT_15_4_API int STDCALL Dut_15_4_TxBurst ( int PacketOption, int PktGap, DWORD
 DeviceId )
```

Table 44. Command parameters

Parameter	Type	Description
PacketOption	Int	Packet Option 0 = 1 packet 1 = 25 packets 2 = 100 packets 3 = 500 packets 4 = 1000 packets 5 = 2000 packets 6 = 5000 packets 7 = 10000 packets
PktGap	Int	Value of packet gap Must be greater than 6 ms
DeviceId	Double word	Device ID Default = 0

Return parameters

0 = Zigbee15_4_TxBurst return success

Nonzero = Zigbee15_4_TxBurst return failed

10.9 Dut_15_4_SetChannel

This API sets 802.15.4 RX test RF channel.

```
DUT_15_4_API int STDCALL Dut_15_4_SetChannel(int ChannelNum,DWORD DeviceId)
```

Table 45. Command parameters

Parameter	Type	Description
ChannelNum	int	Channel number
DeviceId	Double word	Device ID Default = 0

Return parameters

0 = Zigbee15_4_SetChannel return success

Nonzero = Zigbee15_4_SetChannel return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

10.10 Dut_15_4_StartRxTest

This API starts RX test.

```
DUT_15_4_API int STDCALL Dut_15_4_StartRxTest(DWORD DeviceId)
```

Table 46. Command parameters

Parameter	Type	Description
DeviceId	Double word	Device ID Default = 0

Return parameters

0 = Zigbee15_4_StartRxTest return success

Nonzero = Zigbee15_4_StartRxTest return failed

10.11 Dut_15_4_RxTestResult

This API returns RX test results.

```
DUT_15_4_API int STDCALL Dut_15_4_RxTestResult(BYTE * Status,DWORD *
 RxTotalPcktCount,DWORD * RxPcktCount,int * RSSI,DWORD DeviceId)
```

Table 47. Command parameters

Parameter	Type	Description
Status	Byte	Status of RX test 0 = RX test completed 1 = RX test aborted
RxTotalPcktCount	Double word	Total number of packets specifies in packet waveform
RxPcktCount	Double word	Packets received during RX test session
RSSI	Int	Average RSSI value in dBm
DeviceId	Double word	Device ID Default = 0

Return parameters

0 = Zigbee15_4_RxTestResult return success

Nonzero = Zigbee15_4_RxTestResult return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

10.12 Dut_15_4_GetEUI64MacAddress

This API gets the EUI-64 MAC address.

```
DUT_15_4_API int STDCALL Dut_15_4_GetEUI64MacAddress(unsigned char * EUI64MacAddress,
            DWORD DeviceId)
```

Table 48. Command parameters

Parameter	Type	Description
`EUI64MacAddress`	Unsigned char	EUI-64 Mac Address
`DeviceId`	Double word	Device ID Default = 0

Return parameters

0 = GetEUI64MacAddress return success

Nonzero = GetEUI64MacAddress return failed

10.13 Dut_15_4_SetEUI64MacAddress

This API sets the EUI-64 MAC address.

```
DUT_15_4_API int STDCALL Dut_15_4_SetEUI64MacAddress(unsigned char * EUI64MacAddress,
 DWORD DeviceId)
```

Table 49. Command parameters

Parameter	Type	Description
`EUI64MacAddress`	Unsigned char	EUI-64 MAC address
`DeviceId`	Double word	Device ID Default = 0

Return parameters

0 = SetEUI64MacAddress return success

Nonzero = SetEUI64MacAddress return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

11 Storage API

This section describes the storage APIs.

11.1 List of storage APIs

Table 50. List of storage APIs

Commands
Dut_Shared_SetStorageType
DutIf_ForceE2PromType
DutIf_SetCalDataRevF
DutIf_SetCalDataRevFBTOnly
DutIf_SetCalDataRevF15p4Only

11.2 Dut_Shared_SetStorageType

This API configures the storage type.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetStorageType(int StorageType, DWORD DeviceId)
```

Table 51. Command parameters

Parameter	Type	Description
`Storage Type`	Byte	Storage type 0 = EEPROM 1 = Configuration file 2 = OTP
`DeviceId`	Double word	Device ID Default = 0

Return parameters

0 = Dut_Shared_SetStorageType return success

Nonzero = Dut_Shared_SetStorageType return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

11.3 DutIf_ForceE2PromType

This API specifies the calibration data storage type.

```
DUT_WLAN_API int STDCALL DutIf_ForceE2PromType(DWORD IfType, DWORD AddrWidth, DWORD
 DeviceType);
```

Table 52. Command parameters

Parameter	Type	Description
IfType	Double word	Serial communication 1 = SPI 2 = I2C
AddrWidth	Double word	Address width in byte
DeviceType	Double word	Storage type 1 = EEPROM

Return parameters

0 = DutIf_ForceE2PromType return success

Nonzero = DutIf_ForceE2PromType return failed

11.4 DutIf_SetCalDataRevF

This API stores combo (Wi-Fi, Bluetooth, and 802.15.4) data.

```
DUT_WLAN_API int STDCALL DutIf_SetCalDataRevF( int PurgeAfter, char
 *FlexFileNameNoE2prom)
```

Table 53. Command parameters

Parameter	Type	Description
PurgeAfter	int	1 = calibration will be cleared after data is programmed to storage device/ file
FlexFileNameNoE2prom	char	Output Filename to store combo configuration data NULL = default calibration Filename

Return parameters

0 = DutIf_SetCalDataRevF return success

Nonzero = DutIf_SetCalDataRevF return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

11.5 DutIf_SetCalDataRevFBTOnly

This API sets the Bluetooth calibration data.

```
DUT_WLAN_API int STDCALL DutIf_SetCalDataRevFBTOnly(char *FlexFileNameNoE2prom);
```

Table 54. Command parameters

Parameter	Type	Description
FlexFileNameNoE2prom	char	Output Filename to store Bluetooth configuration data NULL = default calibration Filename

Return parameters

0 = DutIf_SetCalDataRevFBTOnly return success

Nonzero = DutIf_SetCalDataRevFBTOnly return failed

11.6 DutIf_SetCalDataRevF15p4Only

This API sets the 802.15.4 calibration data.

```
DUT_WLAN_API int STDCALL DutIf_SetCalDataRevF15p4Only (char *FlexFileNameNoE2prom);
```

Table 55. Command parameters

Parameter	Type	Description
FlexFileNameNoE2prom	char	Output Filename to store 802.15.4 configuration data NULL = default calibration Filename

Return parameters

0 = DutIf_SetCalDataRevF15p4Only return success

Nonzero = DutIf_SetCalDataRevF15p4Only return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12 Annexes

Hardware main structure is the first and main structure of the calibration structure chain. From the main structure, there is a pointer to the next (annex) structure.

- The annex structure is formed as a link-list.
- The pointer to the next annex structure:
  - The last annex has a pointer for the next annex equal to 0xFFFFFFFF.
  - Length-In-Byte is the length of each annex structure itself. Aligned to 32 bits.
- The type of the annex is an indicator of the annex structure:
  - Annex types 0 and 255 are un-defined.
  - Annex types 201 to 254 are assigned to customers.
- Contact your NXP representative for the document with the annex type definitions.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.1 Hardware main structure

Table 56. Hardware main structure

Offset	Byte3	Byte2	Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)		Checksum	Rev (0x0F)
0x04	Pointer to the first annex structure			
0x08	TX Path Configuration	RX Path Configuration	SPI Size	Device ID (upper 4 bits)
0x0C	RF crystal	Temperature at Cal time		SOC O.R.
0x10	Foundry code	Design Type	Misc_Flag	Region code
0x14	Cal data Version			
0x18	MFG FW Version			
0x1C	DLL Version			

Table 57. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Rev`	Revision of structure Set to 0x07
`Pointer for first annex structure`	Pointer for first annex structure
`TX Path Configuration`	Maximum TX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 57. Command parameters*...continued*

Parameter	Description
`RX Path Configuration`	Maximum RX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
`SPI Size`	Size of SPI EEPROM SPI[6:0] = size of the device SPI[7] = size is in number in kB SPI[7] = size in number in MB
`Device ID`	4-bit device ID Bit [7:4]
`RF Xtal`	External crystal calibration value Applicable only if MiscFlag[0] = 1, meaning that an external crystal is used.
`Temperature at Cal time`	Temperature reading at calibration time
`Foundry code`	Foundry code
`SOC O.R.`	O.R revision for SoC O.R[7:6] = customer ID O.R[5:4] = major O.R revision number O.R[3:0] = minor O.R revision number

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 57. Command parameters*...continued*

Parameter	Description
`Design Type`	Code for the reference design type 0x00 = Reserved
`Misc_Flag`	Flag[0] = Crystal oscillator 0 = use internal crystal oscillator 1 = use external crystal oscillator Flag[1] = Sleep clock 0 = use internal sleep clock 1 = use external sleep clock Flag[2] = Wi-Fi wake up 0 = feature not available 1 = feature available Flag[3] = flip-chip indication, FW checks this bit when there is a difference in setting 0 = nonflip chip 1 = flip chip indication Flag[4] = Calibration data handling options 0 = program all annexes on storage device 1 = only program-specific data on storage device and generate conf file to store module-specific annexes Flag[5] = external calibration data handling in FW 0 = legacy handling, discard all annexes in FW and use new one 1 = retain old annexes from FW and override new one
`Region code`	1 byte region code according to 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China
`Cal data Version`	Calibration data version
`MFG FW Version`	Manufacturing Task version (FW)
`DLL Version`	Manufacturing DLL version (Host)

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.2 Bluetooth configuration – Annex 55

Table 58. Annex 55 structure

Offset	Byte3	Byte2	Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)		Checksum	Annex Type (0x37)(0x5 F for second device)
0x04	Pointer to the next annex structure			
0x08	FE Loss	Initial Power in dBm	RFXTAL	Version
0x0C	Reserved	Reserved	Number of Antennas	Bluetooth Option
0x10	Bluetooth Baud Rate			
0x14	BD Address[3]	BD Address[2]	BD Address[1]	BD Address[0]
0x18	Region code	Reserved	BD Address[5]	BD Address[4]

Table 59. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x37 0x57 for second device
`Pointer for first annex structure`	Pointer for first annex structure
`FE Loss`	Signed byte to keep the front-end loss value in 0.5 dB step
`Initial Power`	Initial Bluetooth TX power in dBm 0.5 dB step
`RFXTAL`	External crystal calibration value used of standalone Bluetooth device Set to 0 for Wi-Fi and Bluetooth device
`Version`	Info filed definition version 0x01 = include AOA support and number of antennas (version 1) 0x02 = include Bluetooth Class 1 support (version 2) 0x03 = region information support (version 3)
`Number of Antennas`	Number of antennas available on the hardware

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 59. Command parameters*...continued*

Parameter	Description
`Bluetooth Option`	Bluetooth option[0] = Force Class 2 operation Bluetooth option[1] = Disable power control for class 2 Bluetooth option[2] = Ext crystal used Bluetooth option[3] = Use internal sleep clock (0 – External, 1- Internal) Bluetooth option[4] = Bluetooth AOA location support (0 –Disabled, 1 – Enabled) Bluetooth option[5] = Force Class 1 mode Bluetooth option[7:6] = Reserved Default = Force Class 1.5
`Bluetooth Baud Rate`	Bluetooth baud rate
`BD Address`	Reverse of Wi-Fi MAC address stored on SPI header BD_Addr[5] = most significant byte BD_Addr[0] = least significant byte
`Region code`	1 byte region code according to the 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.3 Thermal power and RSSI compensation – Annex 75

Table 60. Annex 75 structure

Offset	Byte3	Byte2	Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)		Checksum	Annex type (0x4B)
0x04	Pointer to the next annex structure			
0x08	Reserved	Reserved	Number of thermal compensation entries	Device ID/Path ID
0x0C	Thermal compensation data entry			
0x10				
0x14	Thermal data for next band/subband			

Table 61. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x4B
`Pointer for next annex structure`	Pointer for next annex structure
`Number of RSSI Cal entries`	Number of RSSI calibration data entries stored on this annex
`Device ID/ Path ID`	Bit[7:4] = 4-bit Device ID, set to 0x0 Bit[3:0] = path number. Range 0x0 – 0xf

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 61. Command parameters*...continued*

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Thermal Compensation Data Entry</td><td>Data structure for RSSI calibration data entry

Table 62. RCCI calibration entries
<table>
<tr><th>Byte 3</th><th>Byte 2</th><th>Byte 1</th><th>Byte 0</th></tr>
<tr><td>Denominator</td><td>Numerator cold</td><td>Numerator Hot</td><td>Band/Subband</td></tr>
<tr><td>Reserved</td><td>Back-off value</td><td>RSSI Temp slope numerator bypass</td><td>RSSI temp Slope numerator normal</td></tr>
</table>
Denominator = common denominator
Numerator cold = Slope numerator for cold temperature range
Numerator hot = Slope numerator for hot temperature range
Band/Subband[7:6] = Band index
0b00 = 2.4 GHz band
0b01 = 5 GHz band
0b10 = Reserved
0b11 = Reserved
Band/Subband[5:2] = Subband index
Back-off value = a static back-off 8-bit signed integer. The scale is 1/16. High temp static back-off is used only for the ePA design for temperatures above 85°C ambient or 95°C read on TSEN.
RSSI temp slope numerator bypass = When external LNA is used, this field keeps the RSSI offset temperature slope numerator for external LNA bypass mode.
RSSI temp slope numerator normal = RSSI offset temperature slope numerator for normal mode.</td></tr>
</table>

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.4 Thermal crystal – Annex 83

Table 63. Annex 83 structure

Offset	Byte3	Byte2	Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)		Checksum	Annex type (0x53)
0x04	Pointer to the next annex structure			
0x08	Third Order Coefficient (a)			
0x0C	Second Order Coefficient (b)			
0x10	First Order Coefficient (c)			
0x14	Constant Coefficient (d)			

Table 64. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x53
`Pointer for next annex structure`	Pointer for next annex structure
`Third order efficient (a)`	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$ Unit of float32
`Second order efficient (b)`	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$ Unit of float32
`First order efficient (c)`	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$ Unit of float32
`Constant coefficient (d)`	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$ Unit of float32

The crystal thermal behavior can be calculated using the equation:

$$y(t) = A \times X^3 + B \times X^2 + C \times X + D$$

Where:

y = Frequency offset for a given temperature x in relative crystal cap code values. That is, how many relative crystal cap codes are the frequency error off by.

x = Temperature from device thermal sensor reading. That is, device thermal sensor reading at a given temperature in degrees Celsius.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.5 FEM configuration – Annex 90

Table 65. Annex 90 structure

Offset	Byte3	Byte2	Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)		Checksum	Annex Type (0x5A)
0x04	Pointer to the first annex structure			
0x08	Concurrency Info	FEM Capabilities	Number of FEM Data Entries	Revision
0x0C	2.4 GHz TXVR_A bit mask (16 bits)		2.4 GHz TXVR_B bit mask (16 bits)	
0x10	5 GHz TXVR_A bit mask (16 bits)		5 GHz TXVR_B bit mask (16 bits)	
0x14	5 GHz Antenna Diversity mask (16 bits)		2.4 GHz Antenna Diversity mask (16 bits)	
0x18 . .	FEM Data Entry			
	More Entries……			

Table 66. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x5A
`Pointer for first annex structure`	Pointer for first annex structure
`Concurrency info`	2.4 GHz 2x2 + Bluetooth concurrent capabilities [3:0] [0] = Wi-Fi 2x2 TX + Bluetooth TX and RX concurrent allowed [1] = Wi-Fi 1x2 or 2x2 RX + Bluetooth TX concurrent allowed [2] = Wi-Fi 1x2 or 2x2 RX + Bluetooth RX concurrent allowed [3] = Wi-Fi 1x1 or 1x2 TX + Bluetooth TX and RX concurrent allowed 2.4 GHz 1x1 + Bluetooth concurrent capabilities [7:4] [4] = Wi-Fi TX + Bluetooth TX concurrent allowed [5] = Wi-Fi TX + Bluetooth RX concurrent allowed [6] = Wi-Fi RX + Bluetooth TX concurrent allowed [7] = Wi-Fi RX + Bluetooth RX concurrent allowed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 66. Command parameters*...continued*

Parameter	Description
`FEM capabilities`	FEM capability information 0 = 2.4 GHz capable 1 = 5 GHz capable 2 = Use internal Bluetooth 3 = Bluetooth location feature • 0 = disabled • 1 = enabled 4 = Use external 2.4 GHz LNA 5 = Use external 5 GHz LNA 6 = Reserved 7 = Use external 5 GHz PA
`Number of FEM Data Entries`	Number of FEM data entries
`Rev`	Revision of structure
`2.4 GHz TXVR_A bit mask`	16-bit RF control line bit mask for 2.4 GHz TVXR
`2.4 GHz TXVR_B bit mask`	16-bit RF control line bit mask for 2.4 GHz TVXR
`5 GHz TXVR_A bit mask`	16-bit RF control line bit mask for 5 GHz TVXR
`5 GHz TXVR_B bit mask`	16-bit RF control line bit mask for 5 GHz TVXR
`5 G Antenna Diversity mask`	16-bit antenna diversity RF control line flip mask
`2.4 G Antenna Diversity Mask`	16-bit antenna diversity RF control line flip mask
`FEM Data Entry`	Data entry. See Table 67.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 66. Command parameters*...continued*

Parameter	Description
`Operation Mode Mask`	FEM Data Entry Mode bit position 0 = 2.4 GHz 2x2 802.11ax/802.11ac 1 = 2.4 GHz 1x1 A 802.11ax/802.11ac 2 = 2.4 GHz 1x1 B 802.11ax/802.11ac 3 = 2.4 GHz 1x2 4 = 5 GHz 2x2 802.11ax/802.11ac 5 = 5 GHz 1x1 A 802.11ax/802.11ac 6 = 5 GHz 1x1 B 802.11ax/802.11ac 7 = 5 GHz 1x2 8 = 5 GHz 1x1A 0-DFS 9 = 5 GHz 80+80 10 = 5 GHz 2x2 802.11p 11 = 5 GHz 1x1 B 802.11p ***Note:*** *In 2.4 GHz, 0x000F means MODE_1x1A, MODE_1x1B, MODE_1x2, and MODE_2x2 are supported.*
`Front-end #X RF Ctrl Info`	16-bit RF Control Info Each bit is mapping to the corresponding external RF control pin

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 67. FEM data entry

Offset	Byte3	Byte2	Byte1	Byte0
0x00	Reserved	Reserved	Operation Mode Mask	
0x04	Front-end#1 RF Ctrl Info (16 bits)		Front-end#0 RF Ctrl Info (16 bits)	
0x08	Front-end#3 RF Ctrl Info (16 bits)		Front-end#2 RF Ctrl Info (16 bits)	
0x0C	Front-end#5 RF Ctrl Info (16 bits)		Front-end#4 RF Ctrl Info (16 bits)	
0x10	Front-end#7 RF Ctrl Info (16 bits)		Front-end#6 RF Ctrl Info (16 bits)	
0x14	Front-end#9 RF Ctrl Info (16 bits)		Front-end#8 RF Ctrl Info (16 bits)	
0x18	Front-end#11 RF Ctrl Info (16 bits)		Front-end#10 RF Ctrl Info (16 bits)	
0x1C	Front-end#13 RF Ctrl Info (16 bits)		Front-end#12 RF Ctrl Info (16 bits)	
0x20	Front-end#15 RF Ctrl Info (16 bits)		Front-end#14 RF Ctrl Info (16 bits)	
0x24	Front-end#17 RF Ctrl Info (16 bits)		Front-end#16 RF Ctrl Info (16 bits)	
0x28	Front-end#19 RF Ctrl Info (16 bits)		Front-end#18 RF Ctrl Info (16 bits)	

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.6 TX power table – Annex 97

Table 68. Annex 97 structure

Offset	Byte3	Byte2	Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)		Checksum	Annex Type (0x61)
0x04	Pointer to the next annex structure			
0x08	Temp Threshold	Cal Option	Number of Power Table Entries	Device ID/Path ID
0x0C	Power Table Entry			
0x10				
0x14	More power table entries			
.....				

Table 69. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x61
`Pointer for next annex structure`	Pointer for next annex structure
`Temp threshold`	Temperature threshold for IQ calibration and DPD retrigger
`Cal Option`	Bit[0-1] = power control method 0x0 = close loop power control 0x01 = open loop power control Bit[2] = external PA power detector polarization 0 = positive 1 = negative Bit[3] = Use annex 78 power polynomial data 0 = not in use 1 = to use
`Number of power table entries`	Number of power table entries stored on this annex
`Device ID/Path ID`	Bit[7:4] = 4-bit Device ID, set to 0x0 Bit[3:0] = path number. Range 0x0 – 0xf
`Power table entry`	See Table 70.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 70. Power table entries

Byte3-2		Byte 1	Byte 0
Band/Subband channel		Total number of BM indexes in Cal	Power detector offset
Reserved (2 bits)	Power level MB0/RF1 (10 bits)	Power level BM0/RF0 (10 bits)	Temperature (10 bits)
Reserved (2 bits)	Power level BM0/RF4 (10 bits)	Power level BM0/RF3 (10 bits)	Power level MB0/RF2 (10 bits)
Reserved (2 bits)	Power level BM0/RF7 (10 bits)	Power level BM0/RF6 (10 bits)	Power level BM0/RF5 (10 bits)
Reserved (2 bits)	Power level BM1/RF1 (10 bits)	Power level BM1/RF0 (10 bits)	Temperature (10 bits)
Reserved (2 bits)	Power level BM1/RF4 (10 bits)	Power level BM1/RF3 (10 bits)	Power level BM1/RF2 (10 bits)
Reserved (2 bits)	Power level BM1/RF7 (10 bits)	Power level BM1/RF6 (10 bits)	Power level BM1/RF5 (10 bits)
Power info for BM2			
Power info for BMX			

Table 71. Command parameters

Parameter	Description
`Band/Subband/ Channel`	Bit [15:14] = band index 0x00 = 2.4 GHz 0x01 = 5 GHz 0x10 = 6 GHz Bit[13:10] = subband index Bit[9:0] = channel number
`Number of BM indexes in calibration`	Stored at byte 1 of the power table entry
`Power detector offset`	8-bit signed integer to indicate power detector offset If not used, set to 0x00.d
`Power level BMX/RFX`	10-bit power level info to keep the power level measured during power calibration for the specified BM index/RF step Bit[9] = sign bit Bit[8:3] = integer power level Bit[2:0] = fraction power level

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.7 RSSI offset – Annex 98

Table 72. Annex 98 structure

Offset	Byte3		Byte2	Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)			Checksum	Annex Type (0x62)
0x04	Pointer to the next annex structure				
0x08	Reserved	Reserved	Number of RSSI Cal Entries		Device ID/Path ID
0x0C	RSSI Cal Data Entry				
0x10 …	More entries…				

Table 73. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x63
`Pointer for first annex structure`	Pointer for first annex structure
`Number of RSSI Cal Entries`	Number of RSSI Cal Entries
`RSSI Cal data entry`	See Table 74. Band/Subband Bit [7:6] = Band Index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband Bit [5:2] = Subband Index RSSI Cal E-LNA HIGH, I-LNA HIGH Mode = RSSI calibration data for high external LNA gain, high internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA LOW, I-LNA HIGH Mode = RSSI calibration data for low external LNA gain, high internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA HIGH, I-LNA LOW mode = RSSI calibration data for high external LNA gain, low internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA LOW, I-LNA LOW mode = RSSI calibration data for low external LNA gain, low internal LNA gain mode. The value is in 0.5dB step. RSSI Cal Common offset = if any of the above cal values are outside the [-8, 7.5] dB range, write the average here and the deviations from average into the other RSSI cal entries. The resolution is 1dB.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 74. RSSI calibration data entries

[31:26]	[25:20]	[19:15]	[14:10]	[9:5]	[4:0]
Band/Subband	RSSI cal common offset	RSSI Cal E-LNA HIGH, I-LNA LOW mode	RSSI Cal E-LNA LOW, I-LNA LOW mode	RSSI Cal E-LNA HIGH, I-LNA HIGH Mode	RSSI Cal E-LNA LOW, I-LNA HIGH Mode

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.8 Antenna isolation – Annex 99

Table 75. Annex 99 structure

Offset	Byte3		Byte2		Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)				Checksum	Annex Type (0x63)
0x04	Pointer to the next annex structure					
0x08	Reserved	Zigbee Antennas	Bluetooth/ Bluetooth LE Antennas	Wi-Fi Antennas	Number of Data Entries	Revision
0x0C	Ant sharing Info (32 bits)					
0x10	Isolation Data Entry					
0x14	More Data entries…					
……						

Table 76. Isolation data entries

[31:24]	[23:16]	[15:12]	[11:8]	[7:4]	[3:0]
Reserved	Isolation Value	ANT B	Tech B	ANT A	Tech A

Table 77. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x63
`Pointer for first annex structure`	Pointer for first annex structure
`Zigbee Antennas`	4-bit value Maximum 16 antennas per radio
`Bluetooth/Bluetooth LE Antennas`	4-bit value Maximum 16 antennas per radio
`Wi-Fi antennas`	4-bit value Maximum 16 antennas per radio
`Number of data entries`	Number of Isolation data entries in this annex
`Rev`	Revision of structure
`Antenna sharing info`	See Table 78.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 77. Command parameters*...continued*

Parameter	Description
`Isolation data entry`	Default = 255 dB Range of 255 dB [31:24] = reserved [23:16] = isolation value, range 0 – 255 dB, default = 255 dB [15 :12] = ANT B ID for Tech B [11:8] = Tech B • 0 = Wi-Fi • 1 = Bluetooth/Bluetooth LE • 2 = 802.15.4 [7 :4] = ANT A ID for Tech A [3 :0] = ANT B ID for Tech B

Antenna sharing information:

- If Bit [7:0] are zero, the antennas are shared.
- If Bit [15:8] are zero, all antennas are separate.

Table 78. Antenna sharing information

Byte	Configuration	Bit	Setting	Description	Value
Byte 0	SLNA	LSB-0	Shared LNA Wi-Fi + Bluetooth	If set, Wi-Fi and Bluetooth share LNA	0
		1	Reserved	If set, Wi-Fi and Bluetooth 2 share LNA	0
		2	Shared LNA Wi-Fi + 802.15.4	If set, Wi-Fi and 802.15.4 share LNA	0
		3	Shared LNA Bluetooth + 802.15.4	If set, Bluetooth and 802.15.4 share LNA	0
		4	Reserved	Reserved	0
		5	Reserved	Reserved	0
		6	Reserved	Reserved	0
		7	Reserved	Reserved	0
Byte 1	Antenna sharing	8	Antenna sharing for Wi-Fi and Bluetooth	If set, Wi-Fi and Bluetooth share the antenna	0
		9	Reserved	Reserved	0
		10	Antenna sharing for Wi-Fi and 802.15.4	If set, Wi-Fi and 802.15.4 share the antenna	0
		11	Antenna sharing for Bluetooth and 802.15.4	If set, Bluetooth and 802.15.4 share the antenna	0
		12	Reserved	Reserved	0
		13	Reserved	Reserved	0
		14	Reserved	Reserved	0
		15	Reserved	Reserved	0

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 78. Antenna sharing information*...continued*

Byte	Configuration	Bit	Setting	Description	Value
Byte 2	Shared path	16:17	Wi-Fi path shared with Bluetooth	If bit [8] is set, shared path = A (0) / B (1) / C (2) / D (3)	0
		18:19	Reserved	Reserved	0
		20:21	Wi-Fi path shared with 802.15.4	If bit [10] is set, shared path = A (0) / B (1) / C (2) / D (3)	0
		22	Bluetooth path shared with 802.15.4	If bit [11] is set, shared path = A (0) / B (1)	0
		23	Reserved	Reserved	0
Byte 3	—	24	Reserved	Reserved	0
		25:31	Reserved	Reserved	0
32-bit mask					0xXXXX

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.9 Bluetooth hardware – Annex 100

Table 79. Annex 100 structure

Offset	Byte3	Byte2	Byte1		Byte0	
0x00	Length-In-Byte (aligned to 4 bytes)		Checksum		Annex Type (0x64/ 0x65)	
0x04	Pointer to the next annex structure					
0x08	Bluetooth/Bluetooth LE ext PA FEM CTRL BITMASK		Ext ANT Gain value	Ext ANT Gain Present	Ext_PA Gain	Ext_PA Present
0x0C	Bluetooth/Bluetooth LE ext LNA FEM CTRL BITMASK		Reserved		Ext_LNA Gain	Ext_LNA Present

Table 80. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`Checksum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x37 0x57 for the second device
`Pointer for next annex structure`	Pointer for next annex structure
`Bluetooth/Bluetooth LE ext PA FEM CTRL BITMASK`	Defines the state of RF control line for TX through PA Bit[15:0]
`Ext_PA Gain`	External PA gain in dB Bit[7:1]
`Ext_PA Present`	0 = External PA gain is not present 1 = External PA gain is present
`Bluetooth/Bluetooth LE ext LNA FEM CTRL BITMASK`	Defines the state of RF control line for RX through LNA Bit[15:0]
`Ext_LNA Gain`	External LNA gain in dB Bit[7:1]
`Ext_LNA Present`	0 = External LNA gain is not present 1 = External LNA gain is present
`Ext ANT Gain value`	Specifies the external antenna gain in 0.5 dB step Bit[4:1]
`Ext ANT Gain Present`	0 = External antenna gain is not present 1 = External antenna gain is present

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

12.10 Configuration – Annex 104

Table 81. Annex 104 structure

Offset	Byte3	Byte2	Byte1	Byte0
0x00	Length-In-Byte (aligned to 4 bytes)		CheckSum	Annex Type (0x68)
0x04	Pointer to the next annex structure			
0x08	Reserved	SpiMaxFrequency	TxPowerLimit	Rev
0x0C	EUI64[3]	EUI64[2]	EUI64[1]	EUI64[0]
0x10	EUI64[7]	EUI64[6]	EUI64[5]	EUI64[4]
0x14	Reserved			

Table 82. Command parameters

Parameter	Description
`Length-In-Byte`	32 bytes as indicated by 0x48[15:0] 0x4C = starting address 0xFFFFF000 = recommended for main and annex structure followed with its pointer indication
`CheckSum`	1 byte sum of all the bytes in the structure, including reserved bytes, excluding checksum bytes
`Annex type`	0x68
`Pointer for next annex structure`	Pointer for next annex structure
`Rev`	Revision of structure
`SpiMaxFrequency`	Max SPI bus spped supported 0x00 = default clock speed supports SPI range up to 4 MHz, internal AHB2 bus frequency is set to 16 MHz 0x01 = higher SPI clock speed supports SPI range frequency up to 10 MHz, internal AHB2 bus frequency is set to 64 HHz
`TxPowerLimit`	Tx power clamp applied to IEEE 802.15.4 0x00 = no Tx power clamp is applied 0x14 = limit Tx power to 10 dBm (for regulatory purposes) Set to max Tx power with 0.5 dBm granularity
`EUI164`	802.15.4 EUI 64 MAC address Most significant byte = EUI64[7] Least significant byte = EUI64[0]

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13 Annex APIs

13.1 List of annex APIs

Table 83. List of annex APIs

Command
Hardware main structure
Dut_Shared_ReadMainDataFromFile_RevF_V3
Dut_Shared_SetCalMainDataRevF_V3
Dut_Shared_GetCalMainDataRevF_V3
Dut_Shared_WriteMainDataToFile_RevF_V3
Bluetooth configuration – Annex 55
Dut_Shared_ReadAnnex55DataFromFile_Rev3
Dut_Shared_WriteAnnex55DataToFile_Rev3
Dut_Shared_GetAnnexType55Data_Rev3
Dut_Shared_SetAnnexType55Data_Rev3
Thermal power and RSSI compensation – Annex 75
Dut_Shared_ReadAnnex75DataFromFile_Rev2
Dut_Shared_SetAnnexType75Data_Rev2
Dut_Shared_GetAnnexType75Data_Rev2
Dut_Shared_WriteAnnex75DataToFile_Rev2
Thermal crystal – Annex 83
Dut_Shared_ReadAnnex83DataFromFile
Dut_Shared_SetAnnexType83Data
Dut_Shared_GetAnnexType83Data
Dut_Shared_WriteAnnex83DataToFile
FEM configuration – Annex 90
Dut_Shared_ReadAnnex90DataFromFile_W909X
Dut_Shared_SetAnnexType90Data_W909X
Dut_Shared_GetAnnexType90Data_W909X
TX power table – Annex 97
Dut_Shared_ReadAnnex97DataFromFile_W909X
Dut_Shared_SetAnnexType97Data_W909X
Dut_Shared_GetAnnexType97Data_W909X
Dut_Shared_WriteAnnex97DataToFile_W909X

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 83. List of annex APIs*...continued*

Command
RSSI offset – Annex 98
Dut_Shared_ReadAnnex98DataFromFile_W909X
Dut_Shared_SetAnnexType98Data_W909X
Dut_Shared_GetAnnexType98Data_W909X
Dut_Shared_WriteAnnex98DataToFile_W909X
Antenna isolation – Annex 99
Dut_Shared_ReadAnnex99DataFromFile
Dut_Shared_SetAnnexType99Data
Dut_Shared_GetAnnexType99Data
Dut_Shared_WriteAnnex99DataToFile
Bluetooth hardware – Annex 100
Dut_Shared_ReadAnnex100DataFromFile_Rev2
Dut_Shared_SetAnnexType100Data_Rev2
Dut_Shared_GetAnnexType100Data_Rev2
Dut_Shared_WriteAnnex100DataToFile_Rev2
802.15.4 configuration – Annex 104
Dut_Shared_ReadAnnex104DataFromFile
Dut_Shared_SetAnnexType104Data
Dut_Shared_GetAnnexType104Data
Dut_Shared_WriteAnnex104DataToFile

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.2 Hardware main structure

13.2.1 Dut_Shared_ReadMainDataFromFile_RevF_V3

This API reads the main structure value from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadMainDataFromFile_RevF_V3(char *FileName,
 BYTE *StructRev, BYTE *DesignType, BYTE *SpiSizeInfo, BYTE *DeviceID, BYTE *TXAntConf,
 BYTE *RXAntConf, BYTE *SocOrRev, WORD *TemperatureAtCalTime, BYTE *RFXTAL, BYTE
 *RegionCode, bool *ExtXtalSource, bool *ExtSleepClk, bool *WlanWakeUp, bool *FlipChip,
 bool *Caldatahandling, bool *CalDataLoad, DWORD *TestToolVer, DWORD *MfgTaskVersion,
 DWORD *DllVersion);
```

Table 84. Command parameters

Parameter	Type	Description
`FileName`	char	Filename
`StructRev`	byte	Revision of structure Set to 0x07
`DesignType`	byte	Code for the reference design type 0x00 = Reserved
`SpiSizeInfo`	byte	Size of SPI EEPROM SPI[6:0] = size of the device SPI[7] = size is in number in kB SPI[7] = size in number in MB
`DeviceID`	byte	Device ID Default = 0
`TXAntConf`	byte	Maximum TX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
`RXAntConf`	byte	Maximum TX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
`SocOrRev`	byte	O.R revision for SoC O.R[7:6] = customer ID O.R[5:4] = major O.R revision number O.R[3:0] = minor O.R revision number

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 84. Command parameters*...continued*

Parameter	Type	Description
TemperatureAtCalTime	word	Temperature reading at calibration time
RFXTAL	byte	External crystal Calibration Value
RegionCode	byte	1 byte region code according to the 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China
ExtXtalSource	Bool	Crystal oscillator False = use internal crystal oscillator True = use external crystal oscillator
ExtSleepClk	Bool	Sleep clock False = use internal sleep clock True = use external sleep clock
WlanWakeUp	Bool	Wi-Fi wake-up False = feature not available True = feature available
FlipChip	Bool	Flip-chip indication, FW checks this bit when there is a difference in setting False = nonflip chip True = flip chip indication
Caldatahandling	Bool	Calibration data handling options False = program all annexes on storage device True = only program-specific data on storage device and generate conf file to store module-specific annexes
CalDataLoad	Bool	External calibration data handling in FW False = legacy handling, discard all annexes in FW and use new one True = retain old annexes from FW and override new one
TestToolVer	Double word	Test algorithm code version
MfgTaskVersion	Double word	Manufacturing Task version (FW)
DllVersion	Double word	Manufacturing DLL version (Host)

Return parameters

0 = Dut_Shared_ReadMainDataFromFile_RevF_V3 return success

Nonzero = Dut_Shared_ReadMainDataFromFile_RevF_V3 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.2.2 Dut_Shared_SetCalMainDataRevF_V3

This API sets the main structure values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetCalMainDataRevF_V3(BYTE StructRev, BYTE
 DesignType, BYTE SpiSizeInfo, BYTE DeviceID, BYTE TXAntConf, BYTE RXAntConf, BYTE
 SocOrRev, WORD TemperatureAtCalTime, BYTE RFXTAL, BYTE RegionCode, bool ExtXtalSource,
 bool ExtSleepClk, bool WlanWakeUp, bool FlipChip, bool Caldatahandling, bool
 CalDataLoad, DWORD TestToolVer, DWORD MfgTaskVersion, DWORD DllVersion);
```

Table 85. Command parameters

Parameter	Type	Description
StructRev	byte	Revision of structure Set to 0x07
DesignType	byte	Code for the reference design type 0x00 = Reserved
SpiSizeInfo	byte	Size of SPI EEPROM SPI[6:0] = size of the device SPI[7] = size is in number in kB SPI[7] = size in number in MB
DeviceID	byte	Device ID Default = 0
TXAntConf	byte	Maximum possible TX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
RXAntConf	byte	Maximum TX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
SocOrRev	byte	O.R revision for SoC O.R[7:6] = customer ID O.R[5:4] = major O.R revision number O.R[3:0] = minor O.R revision number
TemperatureAtCalTime	word	Temperature reading at calibration time
RFXTAL	byte	External crystal calibration value

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 85. Command parameters*...continued*

Parameter	Type	Description
RegionCode	byte	1 byte region code according to the 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China
ExtXtalSource	Bool	Crystal oscillator False = use internal crystal oscillator True = use external crystal oscillator
ExtSleepClk	Bool	Sleep clock False = use internal sleep clock True = use external sleep clock
WlanWakeUp	Bool	Wi-Fi wake-up False = feature not available True = feature available
FlipChip	Bool	Flip-chip indication, FW checks this bit when there is a difference in setting False = nonflip chip True = flip chip indication
Caldatahandling	Bool	Calibration data handling options False = program all annexes on storage device True = only program-specific data on storage device and generate conf file to store module-specific annexes
CalDataLoad	Bool	External calibration data handling in FW False = legacy handling, discard all annexes in FW and use new one True = retain old annexes from FW and override new one
TestToolVer	Double word	Test algorithm code version
MfgTaskVersion	Double word	Manufacturing Task version (FW)
DllVersion	Double word	Manufacturing DLL version (Host)

Return parameters

0 = Dut_Shared_SetCalMainDataRevF_V3 return success

Nonzero = Dut_Shared_SetCalMainDataRevF_V3 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.2.3 Dut_Shared_GetCalMainDataRevF_V3

This API gets the main structure data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetCalMainDataRevF_V3(BYTE *StructRev, BYTE
 *DesignType, BYTE *SpiSizeInfo, BYTE *DeviceID, BYTE *TXAntConf, BYTE *RXAntConf,
 BYTE *SocOrRev, WORD *TemperatureAtCalTime, BYTE *RFXTAL, BYTE *RegionCode,
 bool *ExtXtalSource, bool *ExtSleepClk, bool *WlanWakeUp, bool *FlipChip, bool
 *Caldatahandling, bool *CalDataLoad, DWORD *TestToolVer, DWORD *MfgTaskVersion, DWORD
 *DllVersion);
```

Table 86. Command parameters

Parameter	Type	Description
StructRev	byte	Revision of structure Set to 0x07
DesignType	byte	Code for the reference design type
SpiSizeInfo	byte	Size of SPI EEPROM SPI[6:0] = size of the device SPI[7] = size is in number in kB SPI[7] = size in number in MB
DeviceID	byte	Device ID Default = 0
TXAntConf	byte	Maximum TX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
RXAntConf	byte	Maximum TX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
SocOrRev	byte	O.R revision for SoC O.R[7:6] = Customer ID O.R[5:4] = major O.R revision number O.R[3:0] = minor O.R revision number
TemperatureAtCalTime	word	Temperature reading at calibration time
RFXTAL	byte	External crystal calibration value

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 86. Command parameters*...continued*

Parameter	Type	Description
`RegionCode`	byte	1 byte region code according to the 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China
`ExtXtalSource`	Bool	Crystal oscillator False = use internal crystal oscillator True = use external crystal oscillator
`ExtSleepClk`	Bool	Sleep clock False = use internal sleep clock True = use external sleep clock
`WlanWakeUp`	Bool	Wi-Fi wake-up False = feature not available True = feature available
`FlipChip`	Bool	Flip-chip indication, FW checks this bit when there is a difference in setting False = nonflip chip True = flip chip indication
`Caldatahandling`	Bool	Calibration data handling options False = program all annexes on storage device True = only program-specific data on storage device and generate conf file to store module-specific annexes
`CalDataLoad`	Bool	External calibration data handling in FW False = legacy handling, discard all annexes in FW and use new one True = retain old annexes from FW and override new one
`TestToolVer`	Double word	Test algorithm code version
`MfgTaskVersion`	Double word	Manufacturing Task version (FW)
`DllVersion`	Double word	Manufacturing DLL version (Host)

Return parameters

0 = Dut_Shared_GetCalMainDataRevF_V3 return success

Nonzero = Dut_Shared_GetCalMainDataRevF_V3 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.2.4 Dut_Shared_WriteMainDataToFile_RevF_V3

This API write the main structure data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteMainDataToFile_RevF_V3(char *FileName, BYTE
 StructRev, BYTE DesignType, BYTE SpiSizeInfo, BYTE DeviceID, BYTE TXAntConf, BYTE
 RXAntConf, BYTE SocOrRev, WORD TemperatureAtCalTime, BYTE RFXTAL, BYTE RegionCode, bool
 ExtXtalSource, bool ExtSleepClk, bool WlanWakeUp,  bool FlipChip, bool Caldatahandling,
 bool CalDataLoad, DWORD TestToolVer, DWORD MfgTaskVersion, DWORD DllVersion);
```

Table 87. Command parameters

Parameter	Type	Description
FileName	char	Filename
StructRev	byte	Revision of structure Set to 0x07
DesignType	byte	Code for the reference design type 0x00 = Reserved
SpiSizeInfo	byte	Size of SPI EEPROM SPI[6:0] = size of the device SPI[7] = size is in number in kB SPI[7] = size in number in MB
DeviceID	byte	Device ID Default = 0
TXAntConf	byte	Maximum TX path configuration available for the device PathConf[7:0] where bit 0 to 7 mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
RXAntConf	byte	Maximum TX path configuration available for the device PathConf[7:0] where bit 0 to 7 are mapped to Path A to H 0x00 = No path available 0x01 = Only Path A available 0x02 = Only Path B available 0x03 = Path A and Path B available 0x04 = Only Path C available 0x05 = Path A and Path C available 0x06 = Path B and Path C available 0x07 = Path A, B, and C all available 0x0F = Path A, B, C, D all available ***Note:*** *If set to 0x00, FW does not perform antenna diversity.*
SocOrRev	byte	O.R revision for SoC O.R[7:6] = customer ID O.R[5:4] = major O.R revision number O.R[3:0] = minor O.R revision number
TemperatureAtCalTime	word	Temperature reading at calibration time
RFXTAL	byte	External crystal calibration value

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 87. Command parameters*...continued*

Parameter	Type	Description
`RegionCode`	byte	1 byte region code according to the 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China
`ExtXtalSource`	Bool	Crystal oscillator False = use internal crystal oscillator True = use external crystal oscillator
`ExtSleepClk`	Bool	Sleep clock False = use internal sleep clock True = use external sleep clock
`WlanWakeUp`	Bool	Wi-Fi wake-up False = feature not available True = feature available
`FlipChip`	Bool	Flip-chip indication, FW checks this bit when there is a difference in setting False = nonflip chip True = flip chip indication
`Caldatahandling`	Bool	Calibration data handling options False = program all annexes on storage device True = only program-specific data on storage device and generate conf file to store module-specific annexes
`CalDataLoad`	Bool	External calibration data handling in FW False = legacy handling, discard all annexes in FW and use new one True = retain old annexes from FW and override new one
`TestToolVer`	Double word	Test algorithm code version
`MfgTaskVersion`	Double word	Manufacturing Task version (FW)
`DllVersion`	Double word	Manufacturing DLL version (Host)

Return parameters

0 = Dut_Shared_WriteMainDataToFile_RevF_V3 return success

Nonzero = Dut_Shared_WriteMainDataToFile_RevF_V3 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.3 Bluetooth configuration – Annex 55

13.3.1 Dut_Shared_ReadAnnex55DataFromFile_Rev3

This API reads annex 55 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex55DataFromFile_Rev3(char *FileName,
 BYTE *Version,BYTE *RFXtal, BYTE *InitPwr,BYTE *FELoss,bool *ForceClass2Op, bool
 *Class1OpSupport, bool *DisablePwrControl, bool *MiscFlag, bool *UsedInternalSleepClock,
 bool *AOALocaltionSupport, BYTE *NumberOfAntennas, BYTE *Rssi_Golden_Lo,BYTE
 *Rssi_Golden_Hi, DWORD *BTBAUDRate, BYTE BDAddr[6],BYTE *Encr_Key_Len_Max, BYTE
 *Encr_Key_Len_Min, BYTE *RegionCode);
```

Table 88. Command parameters

Parameter	Type	Description
`FileName`	char	Filename
`Version`	byte	Info filed definition version 0x01 = include AOA support and number of antennas (version 1) 0x02 = include Bluetooth Class 1 support (version 2) 0x03 = region information support (version 3)
`RFXtal`	byte	External crystal calibration value used of standalone Bluetooth device Set to 0 for Wi-Fi and Bluetooth device
`InitPwr`	byte	Initial power
`FELoss`	byte	Signed byte to keep the front-end loss value in 0.5 dB step
`ForceClass20p`	Bool	Class 2 support False = class 2 not supported True = class 2 supported
`Class10pSupport`	Bool	Force class 1 support False = class 1 not supported True = class 1 supported
`DisablePwrControl`	Bool	Power control False = power control disabled True = power control enabled
`MiscFlag`	Bool	Flag[0] = Crystal oscillator 0 = use internal crystal oscillator 1 = use external crystal oscillator Flag[1] = Sleep clock 0 = use internal sleep clock 1 = use external sleep clock Flag[2] = Wi-Fi wake up 0 = feature not available 1 = feature available Flag[3] = flip-chip indication, FW checks this bit when there is a difference in setting 0 = nonflip chip 1 = flip chip indication Flag[4] = Calibration data handling options 0 = program all annexes on storage device 1 = only program-specific data on storage device and generate conf file to store module-specific annexes Flag[5] = external calibration data handling in FW 0 = legacy handling, discard all annexes in FW and use new one 1 = retain old annexes from FW and override new one

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 88. Command parameters*...continued*

Parameter	Type	Description
UsedInternalSleepClock	Bool	Sleep clock False = External sleep clock True = internal sleep clock
AOALocaltionSupport	Bool	AOA location support False = not supported True = supported
NumberOfAntennas	byte	Number of antennas available on hardware
Rssi_Golden_Lo	byte	Golden RSSI for internal-low LNA gain state
Rssi_Golden_Hi	byte	Golden RSSI for internal-high LNA gain state
BTBAUDRate	byte	Bluetooth baud rate
BDAddr	byte	Reverse of Wi-Fi MAC address stored on SPI header BD_Addr[5] = most significant byte BD_Addr[0] = least significant byte
Encr_Key_Len_Max	byte	Encryption key length maximum
Encr_Key_Len_Min	byte	Encryption key length minimum
RegionCode	byte	1 byte region code according to 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China

Return parameters

0 = Dut_Shared_ReadAnnex55DataFromFile_Rev3 return success

Nonzero = Dut_Shared_ReadAnnex55DataFromFile_Rev3 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.3.2 Dut_Shared_SetAnnexType55Data_Rev3

This API sets the annex 55 values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType55Data_Rev3( BYTE Version, BYTE
 RFXtal, BYTE InitPwr, BYTE FELoss, bool ForceClass2Op, bool Class1OpSupport, bool
 DisablePwrControl, bool MiscFlag, bool UsedInternalSleepClock, bool AOALocaltionSupport,
 BYTE NumberOfAntennas, BYTE Rssi_Golden_Lo, BYTE Rssi_Golden_Hi, DWORD BTBAUDRate, BYTE
 BDAddr[6], BYTE Encr_Key_Len_Max, BYTE Encr_Key_Len_Min, BYTE RegionCode);
```

Table 89. Command parameters

Parameter	Type	Description
Version	byte	Info filed definition version 0x01 = include AOA support and number of antennas (version 1) 0x02 = include Bluetooth Class 1 support (version 2) 0x03 = region information support (version 3)
RFXtal	byte	External crystal calibration value used of standalone Bluetooth device Set to 0 for Wi-Fi and Bluetooth device
InitPwr	byte	Initial power
FELoss	byte	Signed byte to keep the front-end loss value in 0.5 dB step
ForceClass20p	Bool	Class 2 support False = class 2 not supported True = class 2 supported
Class10pSupport	Bool	Force class 1 support False = class 1 not supported True = class 1 supported
DisablePwrControl	Bool	Power control False = power control disabled True = power control enabled
MiscFlag	Bool	Flag[0] = Crystal oscillator 0 = use internal crystal oscillator 1 = use external crystal oscillator Flag[1] = Sleep clock 0 = use internal sleep clock 1 = use external sleep clock Flag[2] = Wi-Fi wake up 0 = feature not available 1 = feature available Flag[3] = flip-chip indication, FW checks this bit when there is a difference in setting 0 = nonflip chip 1 = flip chip indication Flag[4] = Calibration data handling options 0 = program all annexes on storage device 1 = only program-specific data on storage device and generate conf file to store module-specific annexes Flag[5] = external calibration data handling in FW 0 = legacy handling, discard all annexes in FW and use new one 1 = retain old annexes from FW and override new one
UsedInternalSleepClock	Bool	Sleep clock False = External sleep clock True = internal sleep clock

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 89. Command parameters*...continued*

Parameter	Type	Description
AOALocaltionSupport	Bool	AOA location support False = not supported True = supported
NumberOfAntennas	byte	Number of antennas available on hardware
Rssi_Golden_Lo	byte	Golden RSSI for internal-low LNA gain state
Rssi_Golden_Hi	byte	Golden RSSI for internal-high LNA gain state
BTBAUDRate	byte	Bluetooth Baud Rate
BDAddr	byte	Reverse of Wi-Fi MAC address stored on SPI header BD_Addr[5] = most significant byte BD_Addr[0] = least significant byte
Encr_Key_Len_Max	byte	Encryption key length maximum
Encr_Key_Len_Min	byte	Encryption key length minimum
RegionCode	byte	1 byte region code according to the 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China

Return parameters

0 = Dut_Shared_SetAnnexType55Data_Rev3 return success

Nonzero = Dut_Shared_SetAnnexType55Data_Rev3 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.3.3 Dut_Shared_GetAnnexType55Data_Rev3

This API gets the annex 55 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType55Data_Rev3( BYTE *Version, BYTE
 *RFXtal, BYTE *InitPwr, BYTE *FELoss, bool *ForceClass2Op, bool *Class1OpSupport,
 bool *DisablePwrControl, bool *MiscFlag, bool *UsedInternalSleepClock, bool
 *AOALocaltionSupport, BYTE *NumberOfAntennas, BYTE *Rssi_Golden_Lo, BYTE
 *Rssi_Golden_Hi, DWORD *BTBAUDRate, BYTE BDAddr[6], BYTE *Encr_Key_Len_Max, BYTE
 *Encr_Key_Len_Min, BYTE *RegionCode);
```

Table 90. Command parameters

Parameter	Type	Description
`Version`	byte	Info filed definition version 0x01 = include AOA support and number of antennas (version 1) 0x02 = include Bluetooth Class 1 support (version 2) 0x03 = region information support (version 3)
`RFXtal`	byte	External crystal calibration value of the standalone Bluetooth device Set to 0 for Wi-Fi and Bluetooth devices.
`InitPwr`	byte	Initial power
`FELoss`	byte	Signed byte to keep the front-end loss value in 0.5 dB step
`ForceClass20p`	Bool	Class 2 support False = class 2 not supported True = class 2 supported
`Class10pSupport`	Bool	Force class 1 support False = class 1 not supported True = class 1 supported
`DisablePwrControl`	Bool	Power control False = power control disabled True = power control enabled
`MiscFlag`	Bool	Flag[0] = Crystal oscillator 0 = use internal crystal oscillator 1 = use external crystal oscillator Flag[1] = Sleep clock 0 = use internal sleep clock 1 = use external sleep clock Flag[2] = Wi-Fi wake up 0 = feature not available 1 = feature available Flag[3] = flip-chip indication, FW checks this bit when there is a difference in setting 0 = nonflip chip 1 = flip chip indication Flag[4] = Calibration data handling options 0 = program all annexes on storage device 1 = only program-specific data on storage device and generate conf file to store module-specific annexes Flag[5] = external calibration data handling in FW 0 = legacy handling, discard all annexes in FW and use new one 1 = retain old annexes from FW and override new one
`UsedInternalSleepClock`	Bool	Sleep clock False = External sleep clock True = internal sleep clock

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 90. Command parameters*...continued*

Parameter	Type	Description
AOALocationSupport	Bool	AOA location support False = not supported True = supported
NumberOfAntennas	byte	Number of antennas available on hardware
Rssi_Golden_Lo	byte	Golden RSSI for internal-low LNA gain state
Rssi_Golden_Hi	byte	Golden RSSI for internal-high LNA gain state
BTBAUDRate	byte	Bluetooth Baud Rate
BDAddr	byte	Reverse of Wi-Fi MAC address stored on SPI header BD_Addr[5] = most significant byte BD_Addr[0] = least significant byte
Encr_Key_Len_Max	byte	Encryption key length maximum
Encr_Key_Len_Min	byte	Encryption key length minimum
RegionCode	byte	1 byte region code according to the 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China

Return parameters

0 = Dut_Shared_GetAnnexType55Data_Rev3 return success

Nonzero = Dut_Shared_GetAnnexType55Data_Rev3 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.3.4 Dut_Shared_WriteAnnex55DataToFile_Rev3

This API writes the annex 55 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex55DataToFile_Rev3(char *FileName,
 BYTE Version, BYTE RFXtal,BYTE InitPwr, BYTE FELoss, bool ForceClass2Op,bool
 Class1OpSupport, bool DisablePwrControl, bool MiscFlag, bool UsedInternalSleepClock,
 bool AOALocaltionSupport, BYTE NumberOfAntennas, BYTE Rssi_Golden_Lo, BYTE
 Rssi_Golden_Hi, DWORD BTBAUDRate, BYTE BDAddr[6], BYTE Encr_Key_Len_Max, BYTE
 Encr_Key_Len_Min, BYTE RegionCode);
```

Table 91. Command parameters

Parameter	Type	Description
`FileName`	char	Filename
`Version`	byte	Info filed definition version 0x01 = include AOA support and number of antennas (version 1) 0x02 = include Bluetooth Class 1 support (version 2) 0x03 = region information support (version 3)
`RFXtal`	byte	External crystal calibration value used of standalone Bluetooth device Set to 0 for Wi-Fi and Bluetooth device
`InitPwr`	byte	Initial power
`FELoss`	byte	Signed byte to keep the front-end loss value in 0.5 dB step
`ForceClass20p`	Bool	Class 2 support False = class 2 not supported True = class 2 supported
`Class10pSupport`	Bool	Force class 1 support False = class 1 not supported True = class 1 supported
`DisablePwrControl`	Bool	Power control False = power control disabled True = power control enabled
`MiscFlag`	Bool	Flag[0] = Crystal oscillator 0 = use internal crystal oscillator 1 = use external crystal oscillator Flag[1] = Sleep clock 0 = use internal sleep clock 1 = use external sleep clock Flag[2] = Wi-Fi wake up 0 = feature not available 1 = feature available Flag[3] = flip-chip indication, FW checks this bit when there is a difference in setting 0 = nonflip chip 1 = flip chip indication Flag[4] = Calibration data handling options 0 = program all annexes on storage device 1 = only program-specific data on storage device and generate conf file to store module-specific annexes Flag[5] = external calibration data handling in FW 0 = legacy handling, discard all annexes in FW and use new one 1 = retain old annexes from FW and override new one
`UsedInternalSleepClock`	Bool	Sleep clock False = External sleep clock True = internal sleep clock

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Table 91. Command parameters*...continued*

Parameter	Type	Description
AOALocaltionSupport	Bool	AOA location support False = not supported True = supported
NumberOfAntennas	byte	Number of antennas available on hardware
Rssi_Golden_Lo	byte	Golden RSSI for internal-low LNA gain state
Rssi_Golden_Hi	byte	Golden RSSI for internal-high LNA gain state
BTBAUDRate	byte	Bluetooth Baud Rate
BDAddr	byte	Reverse of Wi-Fi MAC address stored on SPI header BD_Addr[5] = most significant byte BD_Addr[0] = least significant byte
Encr_Key_Len_Max	byte	
Encr_Key_Len_Min	byte	
RegionCode	byte	1 byte region code according to the 802.11 standard 0x10 = FCC (USA) default 0x20 = IC 0x30 = ETSI (European) 0x31 = Spain 0x32 = France 0x40 = MKK (Japan) 0x41 = MKK (Japan) 0x50 = China

Return parameters

0 = Dut_Shared_WriteAnnex55DataToFile_Rev3 return success

Nonzero = Dut_Shared_WriteAnnex55DataToFile_Rev3 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.4 Thermal power and RSSI compensation – Annex 75

Thermal power and RSSI compensation is stored in annex 75.

13.4.1 Structures

The following structures are stored in Annex 75.

```
typedef struct Annex75_V2
{
   BYTE backoff;
   BYTE reserved;
} Annex75_V2;
typedef struct Thermal_POWER_RSSI_COMPENSSTION_VE
{
   BANDSUBBAND_verE BANDSUBBNAD_Info;
   union
   {
       BYTE  CCK_SLOPE;
       BYTE  Numerator_Hot;
   };
   union
   {
      BYTE  OFDM_SLOPE;
      BYTE  Numerator_Cold;
   };
   BYTE  Denominator;
   BYTE  RSSITempSlopeNumeratorNormal;
   BYTE  RSSITempSlopeNumeratorBypass;
   union
   {
      WORD SwitchPointTemperature;
Annex75_V2 V2_Data;
};
}Thermal_POWER_RSSI_COMPENSSTION_VE;
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.4.2 Dut_Shared_ReadAnnex75DataFromFile_Rev2

This API reads annex 75 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex75DataFromFile_Rev2(char *FileName, int
 DUTIndex, int PathId, int *Rev, int *NumOfEnterirs, Thermal_POWER_RSSI_COMPENSSTION_VE
 *ThermalPowerRSSIData);
```

Table 92. Command parameters

Parameter	Type	Description
FileName	char	Filename
DUTIndex	Int	4-bit Device ID set to 0x0
PathID	Int	Path number range 0x0 – 0xf
Rev	Int	Revision
NumOfEnterirs	int	Number of RSSI Calibration data entries stored on this annex
ThermalPowerRSSIData	Thermal_POWER_RSSI_COMPENSATION_VE	Denominator = common denominator Numerator cold = Slope numerator for cold temperature range Numerator hot = Slope numerator for hot temperature range Band/Subband[7:6] = Band index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband[5:2] = Subband index Back-off value = a static back-off 8-bit signed integer. The scale is 1/16. High temp static back-off is used only for the ePA design for temperatures above 85°C ambient or 95°C read on TSEN. RSSI temp slope numerator bypass = When external LNA is used, this field keeps the RSSI offset temperature slope numerator for external LNA bypass mode. RSSI temp slope numerator normal = RSSI offset temperature slope numerator for normal mode.

Return parameters

0 = Dut_Shared_ReadAnnex75DataFromFile_Rev2 return success

Nonzero = Dut_Shared_ReadAnnex75DataFromFile_Rev2 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.4.3 Dut_Shared_SetAnnexType75Data_Rev2

This API sets the annex 75 values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType75Data_Rev2(int DUTIndex, int PathId,
 int Rev, int NumOfEnterirs, Thermal_POWER_RSSI_COMPENSSTION_VE *ThermalPowerRSSIData);
```

Table 93. Command parameters

Parameter	Type	Description
DUTIndex	Int	4-bit Device ID set to 0x0
PathID	Int	Path number range 0x0 – 0xf
Rev	Int	Revision
NumOfEnterirs	int	Number of RSSI Calibration data entries stored on this annex
ThermalPowerRSSIData	Thermal_POWER_RSSI_COMPENSATION_VE	Denominator = common denominator Numerator cold = Slope numerator for cold temperature range Numerator hot = Slope numerator for hot temperature range Band/Subband[7:6] = Band index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband[5:2] = Subband index Back-off value = a static back-off 8-bit signed integer. The scale is 1/16. High temp static back-off is used only for the ePA design for temperatures above 85°C ambient or 95°C read on TSEN. RSSI temp slope numerator bypass = When external LNA is used, this field keeps the RSSI offset temperature slope numerator for external LNA bypass mode. RSSI temp slope numerator normal = RSSI offset temperature slope numerator for normal mode.

Return parameters

0 = Dut_Shared_ReadAnnex75DataFromFile_Rev2 return success

Nonzero = Dut_Shared_ReadAnnex75DataFromFile_Rev2 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.4.4 Dut_Shared_GetAnnexType75Data_Rev2

This API gets the annex 75 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType75Data_Rev2(int DUTIndex, int PathId,
 int *Rev, int *NumOfEnterirs, Thermal_POWER_RSSI_COMPENSSTION_VE *ThermalPowerRSSIData);
```

Table 94. Command parameters

Parameter	Type	Description
DUTIndex	Int	4-bit Device ID set to 0x0
PathID	Int	Path number range 0x0 – 0xf
Rev	Int	Revision
NumOfEnterirs	int	Number of RSSI Calibration data entries stored on this annex
ThermalPowerRSSIData	Thermal_POWER_RSSI_COMPENSATION_VE	Denominator = common denominator Numerator cold = Slope numerator for cold temperature range Numerator hot = Slope numerator for hot temperature range Band/Subband[7:6] = Band index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband[5:2] = Subband index Back-off value = a static back-off 8-bit signed integer. The scale is 1/16. High temp static back-off is used only for the ePA design for temperatures above 85°C ambient or 95°C read on TSEN. RSSI temp slope numerator bypass = When external LNA is used, this field keeps the RSSI offset temperature slope numerator for external LNA bypass mode. RSSI temp slope numerator normal = RSSI offset temperature slope numerator for normal mode.

Return parameters

0 = Dut_Shared_GetAnnexType75Data_Rev2 return success

Nonzero = Dut_Shared_GetAnnexType75Data_Rev2 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.4.5 Dut_Shared_WriteAnnex75DataToFile_Rev2

This API writes the annex 75 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex75DataToFile_Rev2(char *FileName, int
 DUTIndex, int PathId, int Rev, int NumOfEnterirs, Thermal_POWER_RSSI_COMPENSSTION_VE
 *ThermalPowerRSSIData);
```

Table 95. Command parameters

Parameter	Type	Description
FileName	char	Filename
DUTIndex	Int	4-bit Device ID set to 0x0
PathID	Int	Path number range 0x0 – 0xf
Rev	Int	Revision
NumOfEnterirs	int	Number of RSSI Calibration data entries stored on this annex
ThermalPowerRSSIData	Thermal_POWER_RSSI_COMPENSATION_VE	Denominator = common denominator Numerator cold = Slope numerator for cold temperature range Numerator hot = Slope numerator for hot temperature range Band/Subband[7:6] = Band index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband[5:2] = Subband index Back-off value = a static back-off 8-bit signed integer. The scale is 1/16. High temp static back-off is used only for the ePA design for temperatures above 85°C ambient or 95°C read on TSEN. RSSI temp slope numerator bypass = When external LNA is used, this field keeps the RSSI offset temperature slope numerator for external LNA bypass mode. RSSI temp slope numerator normal = RSSI offset temperature slope numerator for normal mode.

Return parameters

0 = Dut_Shared_WriteAnnex75DataToFile_Rev2 return success

Nonzero = Dut_Shared_WriteAnnex75DataToFile_Rev2 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.5 Thermal crystal – Annex 83

13.5.1 Dut_Shared_ReadAnnex83DataFromFile

This API reads annex 83 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex83DataFromFile(char *FileName, float
 *ThirdOrderCoefficient, float *SecondOrderCoefficient, float *FirstOrderCoefficient,
 float * ConstantCoefficient);
```

Table 96. Command parameters

Parameter	Type	Description
FileName	char	Filename
ThirdOrderCoefficient	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
SecondOrderCoefficient	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
FirstOrderCoefficient	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
ConstantCoefficient	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$

Return parameters

0 = Dut_Shared_ReadAnnex83DataFromFile return success

Nonzero = Dut_Shared_ReadAnnex83DataFromFile return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.5.2 Dut_Shared_SetAnnexType83Data

This API sets the annex 83 values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType83Data(float ThirdOrderCoefficient,
 float SecondOrderCoefficient, float FirstOrderCoefficient, float ConstantCoefficient);
```

Table 97. Command parameters

Parameter	Type	Description
`ThirdOrderCoefficient`	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
`SecondOrderCoefficient`	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
`FirstOrderCoefficient`	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
`ConstantCoefficient`	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$

Return parameters

0 = Dut_Shared_SetAnnexType83Data return success

Nonzero = Dut_Shared_SetAnnexType83Data return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.5.3 Dut_Shared_GetAnnexType83Data

This API gets the annex 83 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType83Data(float *ThirdOrderCoefficient,
 float *SecondOrderCoefficient, float *FirstOrderCoefficient, float
 *ConstantCoefficient);
```

Table 98. Command parameters

Parameter	Type	Description
`ThirdOrderCoefficient`	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
`SecondOrderCoefficient`	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
`FirstOrderCoefficient`	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
`ConstantCoefficient`	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$

Return parameters

0 = `Dut_Shared_GetAnnexType83Data` return success

Nonzero = `Dut_Shared_GetAnnexType83Data` return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.5.4 Dut_Shared_WriteAnnex83DataToFile

This API writes the annex 83 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex83DataToFile(char *FileName,float
 ThirdOrderCoefficient, float SecondOrderCoefficient, float FirstOrderCoefficient, float
 ConstantCoefficient);
```

Table 99. Command parameters

Parameter	Type	Description
FileName	char	Filename
ThirdOrderCoefficient	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
SecondOrderCoefficient	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
FirstOrderCoefficient	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$
ConstantCoefficient	Float	Taken from $y(t) = A \times X^3 + B \times X^2 + C \times X + D$

Return parameters

0 = Dut_Shared_WriteAnnex83DataToFile return success

Nonzero = Dut_Shared_WriteAnnex83DataToFile return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.6 FEM configuration – Annex 90

Front-end modules (FEM) are configured with Annex 90.

13.6.1 Structures

The following structures are stored in Annex 90.

```
typedef struct FEM_FrontEnd
{
  WORD FrontEnd;
}FEM_FrontEnd;


#define FEMFrontEndNum 20
typedef struct FEM_DATA_R4_VF
{
   WORD    AntOperationMode;
   WORD    Reserved;
   FEM_FrontEnd  FrontEnd[FEMFrontEndNum];
}FEM_DATA_R4_VF;
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.6.2 Dut_Shared_ReadAnnex90DataFromFile_W909X

This API reads annex 90 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex90DataFromFile_W909X(char *FileName,
 int *pNumOfEnterirs, int *pFEMRev, int *pFEMCap, int *pFEMConcurrency, int
 *pFEM2G_TVXR_A_Mask, int *pFEM2G_TVXR_B_Mask, int *pFEM5G_TVXR_A_Mask, int
 *pFEM5G_TVXR_B_Mask, int *pFEM2G_AntDivMask, int *pFEM5G_AntDivMask, FEM_DATA_R4_VF
 *pData);
```

Table 100. Command parameters

Parameter	Type	Description
FileName	char	Filename
pNumOfEnterirs	Int	Number of FEM data entries
pFEMRev	Int	Revision
pFEMCap	Int	FEM capability information 0 = 2G capable 1 = 5 GHz capable 2 = Use internal Bluetooth 3 = Bluetooth location feature • 0 = disabled • 1 = enabled 4 = Use external 2.4 GHz LNA 5 = Use external 5 GHz LNA 6 = Reserved 7 = Use external 5 GHz PA
pFEMConcurrency	Int	2.4 GHz 2x2 + Bluetooth Concurrent Capabilities [3:0] [0] = Wi-Fi 2x2 TX + Bluetooth TX and RX concurrent allowed [1] = Wi-Fi 1x2 or 2x2 RX + Bluetooth TX concurrent allowed [2] = Wi-Fi 1x2 or 2x2 RX + Bluetooth RX concurrent allowed [3] = Wi-Fi 1x1 or 1x2 TX + Bluetooth TX and RX concurrent allowed 2.4 GHz 1x1 + Bluetooth concurrent capabilities [7:4] [4] = Wi-Fi TX + Bluetooth TX concurrent allowed [5] = Wi-Fi TX + Bluetooth RX concurrent allowed [6] = Wi-Fi RX + Bluetooth TX concurrent allowed [7] = Wi-Fi RX + Bluetooth RX concurrent allowed
pFEM2G_TVXR_A_Mask	Int	16-bit RF control line bit mask for 2.4 GHz TVXR
pFEM2G_TVXR_B_Mask	Int	16-bit RF control line bit mask for 2.4 GHz TVXR
pFEM5G_TVXR_A_Mask	Int	16-bit RF control line bit mask for 5 GHz TVXR
pFEM5G_TVXR_B_Mask	Int	16-bit RF control line bit mask for 5 GHz TVXR
pFEM2G_AntDivMask	Int	16-bit antenna diversity RF control line flip mask
pFEM5G_AntDivMask	Int	16-bit antenna diversity RF control line flip mask
pData	FEM_DATA_R4_VF	FEM configuration data contain antenna operation mode and front end

Return parameters

0 = Dut_Shared_ReadAnnex90DataFromFile_W909X return success

Nonzero = Dut_Shared_ReadAnnex90DataFromFile_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.6.3 Dut_Shared_SetAnnexType90Data_W909X

This API sets Annex 90 data values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType90Data_W909X(int NumOfEnterirs, int
 FEMRev, int FEMCap, int FEMConcurrency, int FEM2G_TVXR_A_Mask, int FEM2G_TVXR_B_Mask,
 int FEM5G_TVXR_A_Mask, int FEM5G_TVXR_B_Mask, int FEM2G_AntDivMask, int
 FEM5G_AntDivMask, FEM_DATA_R4_VF *pData);
```

Table 101. Command parameters

Parameter	Type	Description
pNumOfEnterirs	Int	Number of FEM data entries
pFEMRev	Int	Revision
pFEMCap	Int	FEM capability information 0 = 2.4 GHz capable 1 = 5 GHz capable 2 = Use internal Bluetooth 3 = Bluetooth location feature • 0 = disabled • 1 = enabled 4 = Use external 2.4 GHz LNA 5 = Use external 5 GHz LNA 6 = Reserved 7 = Use external 5 GHz PA
pFEMConcurrency	Int	2.4 GHz 2x2 + Bluetooth concurrent capabilities [3:0] [0] = Wi-Fi 2x2 TX + Bluetooth TX and RX concurrent allowed [1] = Wi-Fi 1x2 or 2x2 RX + Bluetooth TX concurrent allowed [2] = Wi-Fi 1x2 or 2x2 RX + Bluetooth RX concurrent allowed [3] = Wi-Fi 1x1 or 1x2 TX + Bluetooth TX and RX concurrent allowed 2.4 GHz 1x1+Bluetooth concurrent capabilities [7:4] [4] = Wi-Fi TX + Bluetooth TX concurrent allowed [5] = Wi-Fi TX + Bluetooth RX concurrent allowed [6] = Wi-Fi RX + Bluetooth TX concurrent allowed [7] = Wi-Fi RX + Bluetooth RX concurrent allowed
pFEM2G_TVXR_A_Mask	Int	16-bit RF control line bit mask for 2.4 GHz TVXR
pFEM2G_TVXR_B_Mask	Int	16-bit RF control line bit mask for 2.4 GHz TVXR
pFEM5G_TVXR_A_Mask	Int	16-bit RF control line bit mask for 5 GHz TVXR
pFEM5G_TVXR_B_Mask	Int	16-bit RF control line bit mask for 5 GHz TVXR
pFEM2G_AntDivMask	Int	16-bit antenna diversity RF control line flip mask
pFEM5G_AntDivMask	Int	16-bit antenna diversity RF control line flip mask
pData	FEM_DATA_R4_VF	FEM configuration data contain antenna operation mode and front end

Return parameters

0 = Dut_Shared_SetAnnexType90Data_W909X return success

Nonzero = Dut_Shared_SetAnnexType90Data_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.6.4 Dut_Shared_GetAnnexType90Data_W909X

This API gets Annex 90 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType90Data_W909X(int *pNumOfEnterirs,
 int *pFEMRev, int *pFEMCap, int *pFEMConcurrency, int *pFEM2G_TVXR_A_Mask,
 int *pFEM2G_TVXR_B_Mask, int *pFEM5G_TVXR_A_Mask, int *pFEM5G_TVXR_B_Mask, int
 *pFEM2G_AntDivMask, int *pFEM5G_AntDivMask, FEM_DATA_R4_VF *pData);
```

Table 102. Command parameters

Parameter	Type	Description
pNumOfEnterirs	Int	Number of FEM data entries
pFEMRev	Int	Revision
pFEMCap	Int	FEM capability information 0 = 2.4 GHz capable 1 = 5 GHz capable 2 = Use internal Bluetooth 3 = Bluetooth location feature • 0 = disabled • 1 = enabled 4 = Use external 2.4 GHz LNA 5 = Use external 5 GHz LNA 6 = Reserved 7 = Use external 5 GHz PA
pFEMConcurrency	Int	2.4 GHz 2x2 + Bluetooth concurrent capabilities [3:0] [0] = Wi-Fi 2x2 TX + Bluetooth TX and RX concurrent allowed [1] = Wi-Fi 1x2 or 2x2 RX + Bluetooth TX concurrent allowed [2] = Wi-Fi 1x2 or 2x2 RX + Bluetooth RX concurrent allowed [3] = Wi-Fi 1x1 or 1x2 TX + Bluetooth TX and RX concurrent allowed 2.4 GHz 1x1 + Bluetooth concurrent capabilities [7:4] [4] = Wi-Fi TX + Bluetooth TX concurrent allowed [5] = Wi-Fi TX + Bluetooth RX concurrent allowed [6] = Wi-Fi RX + Bluetooth TX concurrent allowed [7] = Wi-Fi RX + Bluetooth RX concurrent allowed
pFEM2G_TVXR_A_Mask	Int	16 bits RF control line bit mask for 2.4 GHz TVXR
pFEM2G_TVXR_B_Mask	Int	16 bits RF control line bit mask for 2.4 GHz TVXR
pFEM5G_TVXR_A_Mask	Int	16 bits RF control line bit mask for 5 GHz TVXR
pFEM5G_TVXR_B_Mask	Int	16 bits RF control line bit mask for 5 GHz TVXR
pFEM2G_AntDivMask	Int	16 bits antenna diversity RF control line flip mask
pFEM5G_AntDivMask	Int	16 bits antenna diversity RF control line flip mask
pData	FEM_DATA_R4_VF	FEM configuration data contain antenna operation mode and front end

Return parameters

0 = Dut_Shared_GetAnnexType90Data_W909X return success

Nonzero = Dut_Shared_GetAnnexType90Data_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.6.5 Dut_Shared_WriteAnnex90DataToFile_W909X

This API writes the annex 90 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex90DataToFile_W909X(char *FileName, int
 NumOfEnterirs, int FEMRev, int FEMCap, int FEMConcurrency, int FEM2G_TVXR_A_Mask, int
 FEM2G_TVXR_B_Mask, int FEM5G_TVXR_A_Mask, int FEM5G_TVXR_B_Mask, int FEM2G_AntDivMask,
 int FEM5G_AntDivMask, FEM_DATA_R4_VF *pData);
```

Table 103. Command parameters

Parameter	Type	Description
pNumOfEnterirs	Int	Number of FEM data entries
pFEMRev	Int	Revision
pFEMCap	Int	FEM capability information 0 = 2.4 GHz capable 1 = 5 GHz capable 2 = Use internal Bluetooth 3 = Bluetooth location feature • 0 = disabled • 1 = enabled 4 = Use external 2.4 GHz LNA 5 = Use external 5 GHz LNA 6 = Reserved 7 = Use external 5 GHz PA
pFEMConcurrency	Int	2.4 GHz 2x2 + Bluetooth concurrent capabilities [3:0] [0] = Wi-Fi 2x2 TX + Bluetooth TX and RX concurrent allowed [1] = Wi-Fi 1x2 or 2x2 RX + Bluetooth TX concurrent allowed [2] = Wi-Fi 1x2 or 2x2 RX + Bluetooth RX concurrent allowed [3] = Wi-Fi 1x1 or 1x2 TX + Bluetooth TX and RX concurrent allowed 2.4 GHz 1x1 + Bluetooth concurrent capabilities [7:4] [4] = Wi-Fi TX + Bluetooth TX concurrent allowed [5] = Wi-Fi TX + Bluetooth RX concurrent allowed [6] = Wi-Fi RX + Bluetooth TX concurrent allowed [7] = Wi-Fi RX + Bluetooth RX concurrent allowed
pFEM2G_TVXR_A_Mask	Int	16-bit RF control line bit mask for 2.4 GHz TVXR
pFEM2G_TVXR_B_Mask	Int	16-bit RF control line bit mask for 2.4 GHz TVXR
pFEM5G_TVXR_A_Mask	Int	16-bit RF control line bit mask for 5 GHz TVXR
pFEM5G_TVXR_B_Mask	Int	16-bit RF control line bit mask for 5 GHz TVXR
pFEM2G_AntDivMask	Int	16-bit antenna diversity RF control line flip mask
pFEM5G_AntDivMask	Int	16-bit antenna diversity RF control line flip mask
pData	FEM_DATA_R4_VF	FEM configuration data contain antenna operation mode and front end

Return parameters

0 = Dut_Shared_WriteAnnex90DataToFile_W909X return success

Nonzero = Dut_Shared_WriteAnnex90DataToFile_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.7 TX power table – Annex 97

TX power table information is stored in annex 97.

13.7.1 Structures

The following structures are stored in Annex 97.

```
typedef struct TX_POWER_CAL_OPTION_VF
{
   BYTE  PowerControlMethod:2;    //[1:0]
   BYTE     ExternalPAPowerDetectPolar:1;  //[2]
   BYTE  UsePolynomialData:1;    //[3]
   BYTE  Reserved:4;       //[7:4]
}TX_POWER_CAL_OPTION_VF;

typedef struct TX_POWER_SE_SUB
{
   DWORD  Temperature:10;     //Bit[9:0];
   DWORD  PWRLevelBMRF0:10;  //Bit[19:10];
   DWORD  PWRLevelBMRF1:10;  //Bit[29:20];
   DWORD  Reserve1:2;      //Bit[31:30];
   DWORD  PWRLevelBMRF2:10;   //Bit[9:0];
   DWORD  PWRLevelBMRF3:10;  //Bit[19:10];
   DWORD  PWRLevelBMRF4:10;  //Bit[29:20];
   DWORD  Reserve2:2;      //Bit[31:30];
   DWORD  PWRLevelBMRF5:10;     //Bit[9:0];
   DWORD  PWRLevelBMRF6:10;  //Bit[19:10];
   DWORD  PWRLevelBMRF7:10;  //Bit[29:20];
   DWORD  Reserve3:2;     //Bit[31:30];
}TX_POWER_SE_SUB;

typedef struct TX_POWER_SE
{
   BYTE  PwrDetOffset; //Byte 0
   BYTE  NumerOfBM;
   WORD  ChannelNum:10;  //Byte 3-2 ,[9:0]
   WORD  SubBandIndex:4;  //Byte 3-2 ,[13:10]
   WORD  BandId:2;   //Byte 3-2 ,[15:14]
}TX_POWER_SE;
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.7.2 Dut_Shared_ReadAnnex97DataFromFile_W909X

This API reads annex 97 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex97DataFromFile_W909X(char *FileName, int
 DUTIndex, int PathId, TX_POWER_CAL_OPTION_VF *CalOption, int *NumOfEnterirs, TX_POWER_SE
 *TX_PowerData, int *pTraTempThr);
```

Table 104. Command parameters

Parameter	Type	Description
FileName	char	Filename
DUTIndex	Int	4-bit Device ID set to 0x0
PathID	Int	path number range 0x0 – 0xf
CalOption	TX_POWER_CAL_OPTION_VF	Bit[0-1] = power control method 0x0 = close loop power control 0x01 = open loop power control Bit[2] = external PA power detector polarization 0 = positive 1 = negative Bit[3] = Use annex 78 power polynomial data 0 = not in use 1 = to use
NumOfEnterirs	int	Number of power table entries stored on this annex
Tx_PowerData	TX_POWER_SE	TX power data containing power detector offset, number of BM, Channel number Subband index, Band ID
pTraTempThr	Int	Temperature threshold

Return parameters

0 = Dut_Shared_ReadAnnex97DataFromFile_W909X return success

Nonzero = Dut_Shared_ReadAnnex97DataFromFile_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.7.3 Dut_Shared_SetAnnexType97Data_W909X

This API sets the annex 97 values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType97Data_W909X(int DUTIndex, int PathId,
 TX_POWER_CAL_OPTION_VF *CalOption, int NumOfEnterirs, TX_POWER_SE *TX_PowerData,
 TX_POWER_SE_SUB *BM_Data, int TraTempThr);
```

Table 105. Command parameters

Parameter	Type	Description
DUTIndex	Int	4-bit device ID set to 0x0
PathID	Int	path number range 0x0 – 0xf
CalOption	TX_POWER_CAL_OPTION_VF	Bit[0-1] = power control method 0x0 = close loop power control 0x01 = open loop power control Bit[2] = external PA power detector polarization 0 = positive 1 = negative Bit[3] = Use annex 78 power polynomial data 0 = not in use 1 = to use
NumOfEnterirs	int	Number of power table entries stored on this annex
Tx_PowerData	TX_POWER_SE	TX power data containing power detector offset, number of BM, Channel number Subband index, Band ID
BM_Data	TX_POWER_SE_SUB	Defined Power levels 0 - 7
TraTempThr	Int	Temperature threshold

Return parameters

0 = Dut_Shared_SetAnnexType97Data_W909X return success

Nonzero = Dut_Shared_SetAnnexType97Data_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.7.4 Dut_Shared_GetAnnexType97Data_W909X

This API gets the annex 97 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType97Data_W909X(int DUTIndex, int PathId,
 TX_POWER_CAL_OPTION_VF *CalOption, int *NumOfEnterirs, TX_POWER_SE *TX_PowerData,
 TX_POWER_SE_SUB *BM_Data, int *pTraTempThr);
```

Table 106. Command parameters

Parameter	Type	Description
DUTIndex	Int	4-bit device ID set to 0x0
PathID	Int	path number range 0x0 – 0xf
CalOption	TX_POWER_CAL_OPTION_VF	Bit[0-1] = power control method 0x0 = close loop power control 0x01 = open loop power control Bit[2] = external PA power detector polarization 0 = positive 1 = negative Bit[3] = Use annex 78 power polynomial data 0 = not in use 1 = to use
NumOfEnterirs	int	Number of power table entries stored on this annex
Tx_PowerData	TX_POWER_SE	TX power data containing power detector offset, number of BM, Channel number Subband index, Band ID
BM_Data	TX_POWER_SE_SUB	Defined Power levels 0 - 7
TraTempThr	Int	Temperature threshold

Return parameters

0 = Dut_Shared_GetAnnexType97Data_W909X return success

Nonzero = Dut_Shared_GetAnnexType97Data_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.7.5 Dut_Shared_WriteAnnex97DataToFile_W909X

This API writes the annex 97 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex97DataToFile_W909X(char *FileName, int
 DUTIndex, int PathId, TX_POWER_CAL_OPTION_VF *CalOption, int NumOfEnterirs, TX_POWER_SE
 *TX_PowerData, int TraTempThr);
```

Table 107. Command parameters

Parameter	Type	Description
FileName	char	Filename
DUTIndex	Int	4-bit Device ID set to 0x0
PathID	Int	path number range 0x0 – 0xf
CalOption	TX_POWER_CAL_OPTION_VF	Bit[0-1] = power control method 0x0 = close loop power control 0x01 = open loop power control Bit[2] = external PA power detector polarization 0 = positive 1 = negative Bit[3] = Use annex 78 power polynomial data 0 = not in use 1 = to use
NumOfEnterirs	int	Number of power table entries stored on this annex
Tx_PowerData	TX_POWER_SE	TX power data containing power detector offset, number of BM, Channel number Subband index, Band ID
BM_Data	TX_POWER_SE_SUB	Defined Power levels 0 - 7
TraTempThr	Int	Temperature threshold

Return parameters

0 = Dut_Shared_WriteAnnex97DataToFile_W909X return success

Nonzero = Dut_Shared_WriteAnnex97DataToFile_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.8 RSSI offset – Annex 98

13.8.1 Structures

This structure is stored in Annex 75.

```
typedef struct RSSI_CAL_SE
{
 DWORD  ELNA_LO_ILNA_HI_Mode:5; //[4:0]
 DWORD  ELNA_HI_ILNA_HI_Mode:5; //[5:9]
 DWORD  ELNA_LO_ILNA_LO_Mode:5; //[10:14]
 DWORD  ELNA_HI_ILNA_LO_Mode:5; //[15:19]
 DWORD  RSSICalOffSet:6;        //[20:25]
 DWORD  CAL_SubBandID:4;;  //[26:29]
 DWORD  CAL_BandID:2;  //[30:31]
} RSSI_CAL_SE;
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.8.2 Dut_Shared_ReadAnnex98DataFromFile_W909X

This API reads annex 98 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex98DataFromFile_W909X(char *FileName, int
 DUTIndex, int PathId, int *NumOfEnterirs, RSSI_CAL_SE *RSSIData);
```

Table 108. Command parameters

Parameter	Type	Description
FileName	char	Filename
DUTIndex	Int	4-bit device ID set to 0x0
PathID	Int	Path number range 0x0 – 0xf
NumOfEnterirs	Int	Number of RSSI Cal Entries
RSSIData	RSSI_CAL_SE	See Table 74 "RSSI calibration data entries". Band/Subband Bit [7:6] = Band Index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband Bit [5:2] = Subband Index RSSI Cal E-LNA HIGH, I-LNA HIGH mode = RSSI calibration data for high external LNA gain, high internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA LOW, I-LNA HIGH mode = RSSI calibration data for low external LNA gain, high internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA HIGH, I-LNA LOW mode = RSSI calibration data for high external LNA gain, low internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA LOW, I-LNA LOW mode = RSSI calibration data for low external LNA gain, low internal LNA gain mode. The value is in 0.5dB step. RSSI Cal Common offset = if any of the above cal values are outside the [-8, 7.5] dB range, write the average here and the deviations from average into the other RSSI cal entries. The resolution is 1dB.

Return parameters

0 = Dut_Shared_ReadAnnex98DataFromFile_W909X return success

Nonzero = Dut_Shared_ReadAnnex98DataFromFile_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.8.3 Dut_Shared_SetAnnexType98Data_W909X

This API sets the annex 98 values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType98Data_W909X(int DUTIndex, int PathId,
 int NumOfEnterirs, RSSI_CAL_SE *RSSIData);
```

Table 109. Command parameters

Parameter	Type	Description
DUTIndex	Int	4-bit device ID set to 0x0
PathID	Int	Path number range 0x0 – 0xf
NumOfEnterirs	Int	Number of RSSI Cal Entries
RSSIData	RSSI_CAL_SE	See Table 74 "RSSI calibration data entries". Band/Subband Bit [7:6] = Band Index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband Bit [5:2] = Subband Index RSSI Cal E-LNA HIGH, I-LNA HIGH mode = RSSI calibration data for high external LNA gain, high internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA LOW, I-LNA HIGH mode = RSSI calibration data for low external LNA gain, high internal LNA gain mode. The value is in 0.5 dB step. RSSI Cal E-LNA HIGH, I-LNA LOW mode = RSSI calibration data for high external LNA gain, low internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA LOW, I-LNA LOW mode = RSSI calibration data for low external LNA gain, low internal LNA gain mode. The value is in 0.5 dB step. RSSI Cal Common offset = if any of the above cal values are outside the [-8, 7.5] dB range: write the average here and the deviations from average into the other RSSI cal entries. The resolution is 1 dB.

Return parameters

0 = Dut_Shared_SetAnnexType98Data_W909X return success

Nonzero = Dut_Shared_SetAnnexType98Data_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.8.4 Dut_Shared_GetAnnexType98Data_W909X

This API gets the annex 98 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType98Data_W909X(int DUTIndex, int PathId,
 int *NumOfEnterirs, RSSI_CAL_SE *RSSIData);
```

Table 110. Command parameters

Parameter	Type	Description
DUTIndex	Int	4-bit device ID set to 0x0
PathID	Int	Path number range 0x0 – 0xf
NumOfEnterirs	Int	Number of RSSI Cal Entries
RSSIData	RSSI_CAL_SE	See Table 74 "RSSI calibration data entries". Band/Subband Bit [7:6] = Band Index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband Bit [5:2] = Subband Index RSSI Cal E-LNA HIGH, I-LNA HIGH mode = RSSI calibration data for high external LNA gain, high internal LNA gain mode. The value is in 0.5 dB step. RSSI Cal E-LNA LOW, I-LNA HIGH mode = RSSI calibration data for low external LNA gain, high internal LNA gain mode. The value is in 0.5dB step. RSSI Cal E-LNA HIGH, I-LNA LOW mode = RSSI calibration data for high external LNA gain, low internal LNA gain mode. The value is in 0.5 dB step. RSSI Cal E-LNA LOW, I-LNA LOW mode = RSSI calibration data for low external LNA gain, low internal LNA gain mode. The value is in 0.5dB step. RSSI Cal Common offset = if any of the above cal values are outside the [-8, 7.5] dB range, write the average here and the deviations from average into the other RSSI cal entries. The resolution is 1 dB.

Return parameters

0 = Dut_Shared_GetAnnexType98Data_W909X return success

Nonzero = Dut_Shared_GetAnnexType98Data_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.8.5 Dut_Shared_WriteAnnex98DataToFile_W909X

This API writes the annex 98 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex98DataToFile_W909X(char *FileName, int
 DUTIndex, int PathId, int NumOfEnterirs, RSSI_CAL_SE *RSSIData);
```

Table 111. Command parameters

Parameter	Type	Description
FileName	char	Filename
DUTIndex s	Int	4-bit device ID set to 0x0
PathID	Int	Path number range 0x0 – 0xf
NumOfEnterirs	Int	Number of RSSI Cal Entries
RSSIData	RSSI_CAL_SE	See Table 74 "RSSI calibration data entries". Band/Subband Bit [7:6] = Band Index 0b00 = 2.4 GHz band 0b01 = 5 GHz band 0b10 = Reserved 0b11 = Reserved Band/Subband Bit [5:2] = Subband Index RSSI Cal E-LNA HIGH, I-LNA HIGH mode = RSSI calibration data for high external LNA gain, high internal LNA gain mode. The value is in 0.5 dB step. RSSI Cal E-LNA LOW, I-LNA HIGH mode = RSSI calibration data for low external LNA gain, high internal LNA gain mode. The value is in 0.5 dB step. RSSI Cal E-LNA HIGH, I-LNA LOW mode = RSSI calibration data for high external LNA gain, low internal LNA gain mode. The value is in 0.5 dB step. RSSI Cal E-LNA LOW, I-LNA LOW mode = RSSI calibration data for low external LNA gain, low internal LNA gain mode. The value is in 0.5 dB step. RSSI Cal Common offset = if any of the above cal values are outside the [-8, 7.5] dB range, write the average here and the deviations from average into the other RSSI cal entries. The resolution is 1 dB.

Return parameters

0 = Dut_Shared_WriteAnnex98DataToFile_W909X return success

Nonzero = Dut_Shared_WriteAnnex98DataToFile_W909X return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.9 Antenna isolation – Annex 99

Antenna isolation information is stored in annex 99.

13.9.1 Structures

The following structures are stored in Annex 99.

```
typedef struct ANT_ISOLATION_DATA_VF
{
   WORD  TechA:4; //[3:0]
   WORD  AntA:4;  //[7:4]
   WORD  TechB:4; //[11:8]
   WORD  AntB:4;  //[15:12]
   BYTE  Isolation;
   BYTE  Reserved;
} ANT_ISOLATION_DATA_VF;

//Annex 99 Antenna Isolation Annex Rev 1
typedef struct ANNEX_TYPE_99_ANT_ISOLATION_DATA
{
   ANNEX_TYPE_HEADER_COMMON typeHdr;
   BYTE      Revision;
   BYTE      NumOfEnterirs;
   WORD      Num_WLAN_Ant:4; //[3:0]
   WORD      Num_BT_Ant:4; //[7:4]
   WORD      Num_ZB_Ant:4; //[11:8]
   WORD      Reserved:4;  //[15:12]
   DWORD     Ant_Sharing_Info;
   ANT_ISOLATION_DATA_VF  Data[0];
}ANNEX_TYPE_99_ANT_ISOLATION_DATA;
```


All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.9.2 Dut_Shared_ReadAnnex99DataFromFile

This API reads annex 99 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex99DataFromFile(char *FileName, int
 *pNumOfEnterirs, int *pRevision, int *pNumWLANAnt, int *pNumBTAnt, int *pNumZBAnt, DWORD
 *pAnt_Sharing_Info, ANT_ISOLATION_DATA_VF *pIsolationData);
```

Table 112. Command parameters

Parameter	Type	Description
FileName	char	Filename
pNumOfEnterirs	Int	Number of Isolation data entries in this annex
pRevision	Int	Revision of structure
pNumWLANAnt	Int	4-bit value Maximum 16 antennas per radio
pNumBTAnt	Int	4-bit value Maximum 16 antennas per radio
pNumZBAnt	Int	4-bit value Maximum 16 antennas per radio
pAnt_Sharing_Info	Double word	If Bit 7:0 are zero, then no shared LNA. If Bit 15:8 are zero, all antennas are separate.
pIsolationData	ANT_ISOLATION_DATA_VF	Default = 255 dB Range of 255 dB [31:24] = reserved [23:16] = isolation value, range 0 – 255 dB, default = 255 dB [15 :12] = ANT B ID for Tech B [11 :8] = Tech B • 0 = Wi-Fi • 1 = Bluetooth/Bluetooth LE • 2 = 802.15.4 [7 :4] = ANT A ID for Tech A

Return parameters

0 = Dut_Shared_ReadAnnex99DataFromFile return success

Nonzero = Dut_Shared_ReadAnnex99DataFromFile return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.9.3 Dut_Shared_SetAnnexType99Data

This API sets the annex 99 values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType99Data(int NumOfEnterirs, int
 Revision, int NumWLANAnt, int NumBTAnt, int NumZBAnt, DWORD Ant_Sharing_Info,
 ANT_ISOLATION_DATA_VF *pIsolationData);
```

Table 113. Command parameters

Parameter	Type	Description
pNumOfEnterirs	Int	Number of Isolation data entries in this annex
pRevision	Int	Revision of structure
pNumWLANAnt	Int	4-bit value Maximum 16 antennas per radio
pNumBTAnt	Int	4-bit value Maximum 16 antennas per radio
pNumZBAnt	Int	4-bit value Maximum 16 antennas per radio
pAnt_Sharing_Info	Double word	If Bit 7:0 are zero, then no shared LNA. If Bit 15:8 are zero, all antennas are separate.
pIsolationData	ANT_ISOLATION_DATA_VF	Default = 255 dB Range of 255 dB [31:24] = reserved [23:16] = isolation value, range 0 – 255 dB, default = 255 dB [15:12] = ANT B ID for Tech B [11:8] = Tech B • 0 = Wi-Fi • 1 = Bluetooth/Bluetooth LE • 2 = 802.15.4 [7:4] = ANT A ID for Tech A

Return parameters

0 = Dut_Shared_SetAnnexType99Data return success

Nonzero = Dut_Shared_SetAnnexType99Data return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.9.4 Dut_Shared_GetAnnexType99Data

This API gets the annex 99 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType99Data(int *pNumOfEnterirs, int
 *pRevision, int *pNumWLANAnt, int *pNumBTAnt, int *pNumZBAnt, DWORD *pAnt_Sharing_Info,
 ANT_ISOLATION_DATA_VF *pIsolationData);
```

Table 114. Command parameters

Parameter	Type	Description
pNumOfEnterirs	Int	Number of Isolation data entries in this annex
pRevision	Int	Revision of structure
pNumWLANAnt	Int	4-bit value Maximum 16 antennas per radio
pNumBTAnt	Int	4-bit value Maximum 16 antennas per radio
pNumZBAnt	Int	4-bit value Maximum 16 antennas per radio
pAnt_Sharing_Info	Double word	If Bit 7:0 are zero, then no shared LNA. If Bit 15:8 are zero, all antennas are separate.
pIsolationData	ANT_ISOLATION_DATA_VF	Default = 255 dB Range of 255 dB [31:24] = reserved [23:16] = isolation value, range 0 – 255 dB, default = 255 dB [15 :12] = ANT B ID for Tech B [11 :8] = Tech B • 0 = Wi-Fi • 1 = Bluetooth/Bluetooth LE • 2 = 802.15.4 [7 :4] = ANT A ID for Tech A

Return parameters

0 = Dut_Shared_GetAnnexType99Data return success

Nonzero = Dut_Shared_GetAnnexType99Data return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.9.5 Dut_Shared_WriteAnnex99DataToFile

This API writes the annex 55 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex99DataToFile(char *FileName, int
 NumOfEnterirs, int Revision, int NumWLANAnt, int NumBTAnt, int NumZBAnt, DWORD
 Ant_Sharing_Info, ANT_ISOLATION_DATA_VF *pIsolationData);
```

Table 115. Command parameters

Parameter	Type	Description
pNumOfEnterirs	Int	Number of Isolation data entries in this annex
pRevision	Int	Revision of structure
pNumWLANAnt	Int	4-bit value Maximum 16 antennas per radio
pNumBTAnt	Int	4-bit value Maximum 16 antennas per radio
pNumZBAnt	Int	4-bit value Maximum 16 antennas per radio
pAnt_Sharing_Info	Double word	If Bit 7:0 are zero, then no shared LNA. If Bit 15:8 are zero, all antennas are separate.
pIsolationData	ANT_ISOLATION_DATA_VF	Default = 255 dB Range of 255 dB [31:24] = reserved [23:16] = isolation value, range 0 – 255 dB, default = 255 dB [15 :12] = ANT B ID for Tech B [11 :8] = Tech B • 0 = Wi-Fi • 1 = Bluetooth/Bluetooth LE • 2 = 802.15.4 [7 :4] = ANT A ID for Tech A
IsolationData	ANT_ISOLATION_DATA_VF	

Return parameters

0 = Dut_Shared_WriteAnnex99DataToFile return success

Nonzero = Dut_Shared_WriteAnnex99DataToFile return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.10 Bluetooth hardware – Annex 100

13.10.1 Dut_Shared_ReadAnnex100DataFromFile_Rev2

This API reads annex 100 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex100DataFromFile_Rev2(char *FileName,
 BYTE *pEPA_Present, BYTE *pEPA_Gain, WORD *pEPA_FEM_Mask, BYTE *pELNA_Present, BYTE
 *pELNA_Gain, WORD *pELNA_FEM_Mask, BYTE *pEANT_Present, BYTE *pANT_Gain);
```

Table 116. Command parameters

Parameter	Type	Description
FileName	byte	Filename
pEPA_Present	byte	0 = External PA gain is not present 1 = External PA gain is present
pEPA_Gain	byte	External PA gain in dB Bit[7:1]
pEPA_FEM_Mask	word	Defines the state of RF control line for TX through PA Bit[15:0]
pELNA_Present	byte	0 = External LNA gain is not present 1 = External LNA gain is present
pELNA_Gain	byte	External LNA gain in dB Bit[7:1]
pELNA_FEM_Mask	word	Defines the state of RF control line for RX through LNA Bit[15:0]
pEANT_Present	byte	0 = External antenna gain is not present 1 = External antenna gain is present
pANT_Gain	byte	Specifies the external antenna gain in 0.5 dB step Bit[4:1]

Return parameters

0 = Dut_Shared_ReadAnnex100DataFromFile_Rev2 return success

Nonzero = Dut_Shared_ReadAnnex100DataFromFile_Rev2 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.10.2 Dut_Shared_SetAnnexType100Data_Rev2

This API sets the annex 100 values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType100Data_Rev2(BYTE EPA_Present, BYTE
 EPA_Gain, WORD EPA_FEM_Mask, BYTE ELNA_Present, BYTE ELNA_Gain, WORD ELNA_FEM_Mask, BYTE
 EANT_Present, BYTE EANT_Gain);
```

Table 117. Command parameters

Parameter	Type	Description
FileName	byte	Filename
pEPA_Present	byte	0 = External PA gain is not present 1 = External PA gain is present
pEPA_Gain	byte	External PA gain in dB Bit[7:1]
pEPA_FEM_Mask	word	Defines the state of RF control line for TX through PA Bit[15:0]
pELNA_Present	byte	0 = External LNA gain is not present 1 = External LNA gain is present
pELNA_Gain	byte	External LNA gain in dB Bit[7:1]
pELNA_FEM_Mask	word	Defines the state of RF control line for RX through LNA Bit[15:0]
pEANT_Present	byte	0 = External antenna gain is not present 1 = External antenna gain is present
pANT_Gain	byte	Specifies the external antenna gain in 0.5 dB step Bit[4:1]

Return parameters

0 = `Dut_Shared_ReadAnnex100DataFromFile_Rev2` return success

Nonzero = `Dut_Shared_ReadAnnex100DataFromFile_Rev2` return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.10.3 Dut_Shared_GetAnnexType100Data_Rev2

This API gets the annex 100 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType100Data_Rev2(BYTE *pEPA_Present,
 BYTE *pEPA_Gain, WORD *pEPA_FEM_Mask, BYTE *pELNA_Present, BYTE *pELNA_Gain, WORD
 *pELNA_FEM_Mask, BYTE *pEANT_Present, BYTE *pEANT_Gain);
```

Table 118. Command parameters

Parameter	Type	Description
pEPA_Present	byte	0 = External PA gain is not present 1 = External PA gain is present
pEPA_Gain	byte	External PA gain in dB Bit[7:1]
pEPA_FEM_Mask	word	Defines the state of RF control line for TX through PA Bit[15:0]
pELNA_Present	byte	0 = External LNA gain is not present 1 = External LNA gain is present
pELNA_Gain	byte	External LNA gain in dB Bit[7:1]
pELNA_FEM_Mask	word	Defines the state of RF control line for RX through LNA Bit[15:0]
pEANT_Present	byte	0 = External antenna gain is not present 1 = External antenna gain is present
pANT_Gain	byte	Specifies the external antenna gain in 0.5 dB step Bit[4:1]

Return parameters

0 = Dut_Shared_GetAnnexType100Data_Rev2 return success

Nonzero = Dut_Shared_GetAnnexType100Data_Rev2 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.10.4 Dut_Shared_WriteAnnex100DataToFile_Rev2

This API writes the annex 100 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex100DataToFile_Rev2(char *FileName, BYTE
 EPA_Present, BYTE EPA_Gain, WORD EPA_FEM_Mask, BYTE ELNA_Present, BYTE ELNA_Gain, WORD
 ELNA_FEM_Mask, BYTE EANT_Present, BYTE EANT_Gain);
```

Table 119. Command parameters

Parameter	Type	Description
FileName	char	Filename
pEPA_Present	byte	0 = External PA gain is not present 1 = External PA gain is present
pEPA_Gain	byte	External PA gain in dB Bit[7:1]
pEPA_FEM_Mask	word	Defines the state of RF control line for TX through PA Bit[15:0]
pELNA_Present	byte	0 = External LNA gain is not present 1 = External LNA gain is present
pELNA_Gain	byte	External LNA gain in dB Bit[7:1]
pELNA_FEM_Mask	word	Defines the state of RF control line for RX through LNA Bit[15:0]
pEANT_Present	byte	0 = External antenna gain is not present 1 = External antenna gain is present
pANT_Gain	byte	Specifies the external antenna gain in 0.5 dB step Bit[4:1]

Return parameters

0 = Dut_Shared_WriteAnnex100DataToFile_Rev2 return success

Nonzero = Dut_Shared_WriteAnnex100DataToFile_Rev2 return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.11 configuration – Annex 104

13.11.1 Dut_Shared_ReadAnnex104DataFromFile

This API reads annex 104 values from a file.

```
DUT_SHARED_API int STDCALL Dut_Shared_ReadAnnex104DataFromFile(char *FileName, BYTE
 *Version, BYTE *TxPowerLimit, BYTE _15_4_Addr[SIZE_15_4_ADDRESS_INBYTE]);
```

Table 120. Command parameters

Parameter	Type	Description
FileName	char	Filename
Version	byte	Version
TxPowerLimit	byte	Tx power clamp applied to IEEE 802.15.4 0x00 = no Tx power clamp is applied 0x14 = limit Tx power to 10 dBm (for regulatory purposes) Set to max Tx power with 0.5 dBm granularity
_15_4_Addr	byte	802.15.4 EUI 64 MAC address Most significant byte = EUI64[7] Least significant byte = EUI64[0]

Return parameters

0 = Dut_Shared_ReadAnnex104DataFromFile return success

Nonzero = Dut_Shared_ReadAnnex104DataFromFile return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.11.2 Dut_Shared_SetAnnexType104Data

This API sets the annex 104 values.

```
DUT_SHARED_API int STDCALL Dut_Shared_SetAnnexType104Data(BYTE Version, BYTE
 TxPowerLimit, BYTE _15_4_Addr[SIZE_15_4_ADDRESS_INBYTE]);
```

Table 121. Command parameters

Parameter	Type	Description
`Version`	byte	Version
`TxPowerLimit`	byte	Tx power clamp applied to IEEE 802.15.4 0x00 = no Tx power clamp is applied 0x14 = limit Tx power to 10 dBm (for regulatory purposes) Set to max Tx power with 0.5 dBm granularity
`_15_4_Addr`	byte	802.15.4 EUI 64 MAC address Most significant byte = EUI64[7] Least significant byte = EUI64[0]

Return parameters

0 = Dut_Shared_SetAnnexType104Data return success

Nonzero = Dut_Shared_SetAnnexType104Data return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.11.3 Dut_Shared_GetAnnexType104Data

This API gets the annex 104 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_GetAnnexType104Data(BYTE *Version, BYTE
 *TxPowerLimit, BYTE _15_4_Addr[SIZE_15_4_ADDRESS_INBYTE]);
```

Table 122. Command parameters

Parameter	Type	Description
`Version`	byte	Version
`_15_4_Addr`	byte	802.15.4 EUI 64 MAC address Most significant byte = EUI64[7] Least significant byte = EUI64[0]

Return parameters

0 = Dut_Shared_GetAnnexType104Data return success

Nonzero = Dut_Shared_GetAnnexType104Data return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

13.11.4 Dut_Shared_WriteAnnex104DataToFile

This API writes the annex 104 data.

```
DUT_SHARED_API int STDCALL Dut_Shared_WriteAnnex104DataToFile(char *FileName, BYTE
 Version, BYTE TxPowerLimit, BYTE _15_4_Addr[SIZE_15_4_ADDRESS_INBYTE]);
```

Table 123. Command parameters

Parameter	Type	Description
`FileName`	char	Filename
`Version`	byte	Version
`TxPowerLimit`	byte	Tx power clamp applied to IEEE 802.15.4 0x00 = no Tx power clamp is applied 0x14 = limit Tx power to 10 dBm (for regulatory purposes) Set to max Tx power with 0.5 dBm granularity
`_15_4_Addr`	byte	802.15.4 EUI 64 MAC address Most significant byte = EUI64[7] Least significant byte = EUI64[0]

Return parameters

0 = Dut_Shared_WriteAnnex104DataToFile return success

Nonzero = Dut_Shared_WriteAnnex104DataToFile return failed

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

14 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2023, 2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

15 References

[1] Application note – AN13538: Wi-Fi Embedded Subsystem API Specification V18 (link)
[2] Application note – AN13983: Calibration Structure for AW611, IW611, and IW612 (link)
[3] Application note – AN13639: Calibration Structure for RW61x
[4] Data sheet – AW611: 1x1 Wi-Fi 6 and Bluetooth Combo Solution (link)
[5] Data sheet – IW610: 1x1 Wi-Fi 6 and Bluetooth Low Energy/802.15.4 Solution Family (link)
[6] Data sheet – IW611: 2.4/5 GHz Dual-band 1x1 Wi-Fi 6 and Bluetooth Solution (link)
[7] Data sheet – IW612: 1x1 Dual-band Wi-Fi 6, Bluetooth and 802.15.4 Tri-radio Solution (link)
[8] Data sheet – RW610: Wireless MCU with Integrated Wi-Fi 6 and Bluetooth Low Energy (link)
[9] Data sheet – RW612: Wireless MCU with Integrated Tri-radio Wi-Fi 6 + Bluetooth Low Energy / 802.15.4 (link)
[10] User manual – UM11694: Manufacturing Software User Manual for AW611 (link)
[11] User manual – UM12062: Manufacturing Software User Manual for IW610 (link)
[12] User manual – UM11749: Manufacturing Software User Manual for IW611 (link)
[13] User manual – UM11717: Manufacturing Software User Manual for IW612 (link)
[14] User manual – UM11801: Manufacturing Software User Manual for RW61x (link)
[15] Webpage – AW611: 2.4/5 GHz Dual-band 1x1 Wi-Fi® 6 (802.11ax) + Bluetooth® Automotive Solution (link)
[16] Webpage – IW610: 2.4/5 GHz Dual-band 1x1 Wi-Fi® 6 + Bluetooth Low Energy + 802.15.4 Tri-Radio Solution (link)
[17] Webpage – IW611: 2.4/5 GHz Dual-band 1x1 Wi-Fi® 6 (802.11ax) + Bluetooth® Solution (link)
[18] Webpage – IW612: 2.4/5 GHz Dual-band 1x1 Wi-Fi® 6 (802.11ax) + Bluetooth® + 802.15.4 Tri-radio Solution (link)
[19] Webpage – RW610: Wireless MCU with Integrated Radio: 1x1 Wi-Fi® 6 + Bluetooth® Low Energy Radios (link)
[20] Webpage – RW612: Wireless MCU with Integrated Tri-radio: 1x1 Wi-Fi® 6 + Bluetooth® Low Energy / 802.15.4 (link)

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

16 Revision history

Table 124. Revision history

Document ID	Release date	Description
AN14125 v.2.0	28 February 2025	• Document access changed to public. • Section 1.2 "Supported products": updated. • Section 3 "Calibration data": updated. • Section 6.2 "Data rates": updated. • Section 6.3 "RF channels": updated. • Section *Bluetooth channels*: removed. • Section 12.8 "Antenna isolation – Annex 99": updated. • Section 15 "References": updated.
AN14125 v.1.0	4 December 2023	• Initial version

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors 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. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Suitability for use in automotive applications — This NXP product has been qualified for use in automotive applications. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Contents

All information provided in this document is subject to legal disclaimers.

© 2025 NXP B.V. All rights reserved.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2025 NXP B.V. All rights reserved.