XWAY™ xRX300/WAVE300 Family

DUT Graphical User Interface

for
DUT Tool Package

User's Guide

Revision 1.1, 2013-05-06

Edition 2013-05-06

Published by
Lantiq Deutschland GmbH
Lilienthalstraße 15
85579 Neubiberg
Germany

Legal Disclaimer

The information given in this document is confidential.

SUCH INFORMATION SHALL IN NO EVENT BE REGARDED AS A GUARANTEE OF CONDITIONS OR CHARACTERISTICS. WITH RESPECT TO ANY VALUES STATED AS "TYPICAL" AS WELL AS EXAMPLES OR HINTS PROVIDED HEREIN, INCLUDING THOSE RELATED TO USE AND/OR IMPLEMENTATION OR APPLICATION OF COMPONENTS, LANTIQ DEUTSCHLAND GMBH ("LANTIQ") HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND, INCLUDING WITHOUT LIMITATION, WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY. IN PARTICULAR, THOSE VALUES, EXAMPLES OR HINTS ARE NO WARRANTIES OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE LANTIQ COMPONENT AND ALWAYS HAVE TO BE VERIFIED IN THE TARGET APPLICATION.

The information in this document is subject to regular updates and/or corrections via errata sheet(s) or new revisions. It is only valid as updated and/or corrected. Once a new revision or errata sheet is available, it can be downloaded from Lantiq's Competence Center:

www.lantiq.com/contact-lantiq

Any user of this document must ensure that they only use the latest update- and/or corrected revision.

LANTIQ HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND, INCLUDING WITHOUT LIMITATION, WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY, WITH RESPECT TO PREVIOUS REVISIONS OF THIS DOCUMENT ONCE AN UPDATE OR ERRATA SHEET IS AVAILABLE.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Lantiq office (**www.lantiq.com**).

Warning

Lantiq components must not be used in life-support devices or systems.

Life-support devices or systems are those systems intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Confidential

XWAY™ xRX300/WAVE300 Family, DUT Graphical User Interface for DUT Tool Package
Confidential
Revision History: Revision 1.1, 2013-05-06

Previous Revision: Revision 1.0, 2013-03-15

Page	Subjects (major changes since last revision)
8	Updated **Section 1.5**, **x86 PC Linux Host Requirements**.
47	Added **Chapter 10**, **Appendix D - Command Line Control**.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

This user guide explains how to use the DUT GUI application. The application is used for hardware development purposes and evaluations based on Lantiq's WiFi MIMO technology.

1.2 About the Test Driver and the DUT GUI Application

The DUT GUI is an application used to help hardware and RF designers test and verify the behavior of their design based on Lantiq's WiFi solution.

DUT GUI uses a specially designed test driver with the firmware that uses all HW and PHY level algorithms. The driver also allows transmission and receiving without the limitations of the IEEE 802.11n standard **[4]**.

The DUT GUI includes an API to control the WiFi functionality (mt_dut.dll). The API can be used to control the DUT (Device Under Test) from an external application.

Together with wireless LAN test equipment maker LitePoint, Lantiq have integrated this dll into LitePoint's IQFact Design Verification Test (DVT) software. IQFact is a comprehensive set of WiFi physical layer transmitter and receiver tests utilizing a large set of channels, data rates and channel bandwidths. The tool allows you to verify assembly in the production line with a vast test coverage in a short test time

1.3 Content of the Installation Kit

The installation package for the DUT GUI installs the following items onto your computer:

- DUT GUI Core
- ActivePerl application - Perl interpreter (used for TCP/IP mode)
- Release Notes **[5]**
- User's Guide (= this document)

Note: Contact Lantiq for the latest application version and installation instructions.

1.4 Supported Interfaces

The DUT GUI application allows you to control the WiFi chipset through the following interface:

- TCP/IP - Control the chipset when it is assembled on a Linux based host. In order for the DUT GUI to work, the host should contain operational driver firmware. The following Host and Linux versions are supported:
- Lantiq's Universal Gateway (UGW) software

1.5 x86 PC Linux Host Requirements

The DUT system provides following requirements to the x86 Linux host in order to operate properly:

1. Microsoft® Windows® XP SP 3.0 and .NET Framework 2.0 (min. requirement)
2. Support for Microsoft® Windows® 7, 32/64 bit
3. OS login/password required is root/wlan
4. This may be changed by editing the linux_hosts.txt file in the DUT's installation directory, with a new username/password to be specified in its [PC] section.
5. Tftp client package must be installed.

2 Installing DUT GUI

To install DUT GUI:

1. From the installation media, DUT_setupx.exe.
   The DUT Installation window appears.

Figure 1 DUT Installation Welcome Screen

Note: Components required for the installation/upgrade are automatically selected.

2. Ensure the correct elements are selected and click Install.
   If an older version is already installed on the computer, you will be prompted to remove the older version before proceeding.

In C:\Program Files\Lantiq\DUT\

- DUT_GUI.exe - GUI application
- dut_cfg.txt - Configuration file to determine the output power level desired during close loop transmission operation and basic platform configurations.
- \autoloader\linux_hosts.txt - configuration file that defines the login and password for the platform
- Regulatory Constraints Folder - Contains information for using regulatory constraints mode
- Developers Folder - Contains XXX.h files for integrating dut.dll in other applications
- DUT GUI User Guide.

Other:

- The log for the DUT GUI application, console.txt, is created in the My Documents folder
- A shortcut to DUT_GUI.exe is created on your desktop

3 Launching DUT GUI

To launch the DUT GUI application:

1. From the desktop, double click the DUT GUI shortcut.
2. Optionally, change the **“dut_cfg.txt Configuration Parameters” on Page 15**.
   The application tries to connect to the DUT.

- If the DUT was pre-configured for the proper parameters, the connection succeeds and a ready message is displayed.
- If the connection is not successful, an error message appears. In this case make sure that the proper parameters and information are correct. For more information on how to troubleshoot an unsuccessful launch, see **“Troubleshooting Launch Errors” on Page 11**.

Main screen of the DUT GUI following a successful launch:

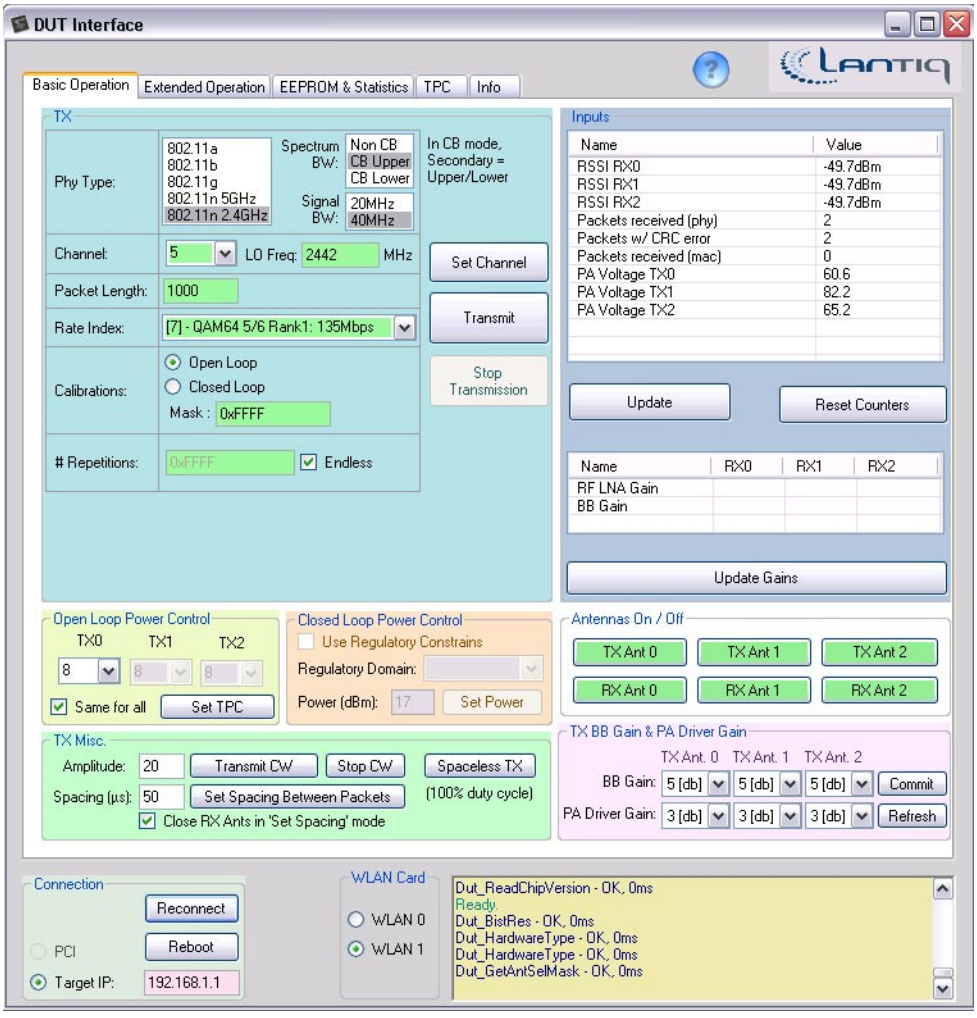


Figure 2 DUT GUI Main Screen (Successful Launch)

Note: Please pay attention to the “Ready” message in the messages pane.

3.1 Troubleshooting Launch Errors

Table 1 Typical Launch Errors and Solutions

Possible Cause of Failure	Solution
External power supply is not turned on.	Close the DUT GUI. Enter DUT into the host. Turn on the power supply. Run the application again.
For a Linux based host, the IP might not be correct.	Verify that the IP address is correct and the PC's Ethernet card is properly configured. Restart the host. After approximately 1 minute, press the reconnect button.
Another application (IQFact, DUT GUI) is using the test driver.	Close the DUT GUI. Reboot the DUT. After the DUT is up, restart the DUT GUI application.
Wrong dut_cfg.txt configuration file parameters are chosen.	Choose the correct HW type and version according to the platform type. Make sure that Wlan index in properly selected in case of dual band concurrent platform.
Wrong Band and channel are configured on initial usage	Insert appropriate Band/Channel and press the reconnect button

An example of the main screen of the DUT GUI following an unsuccessful connection:

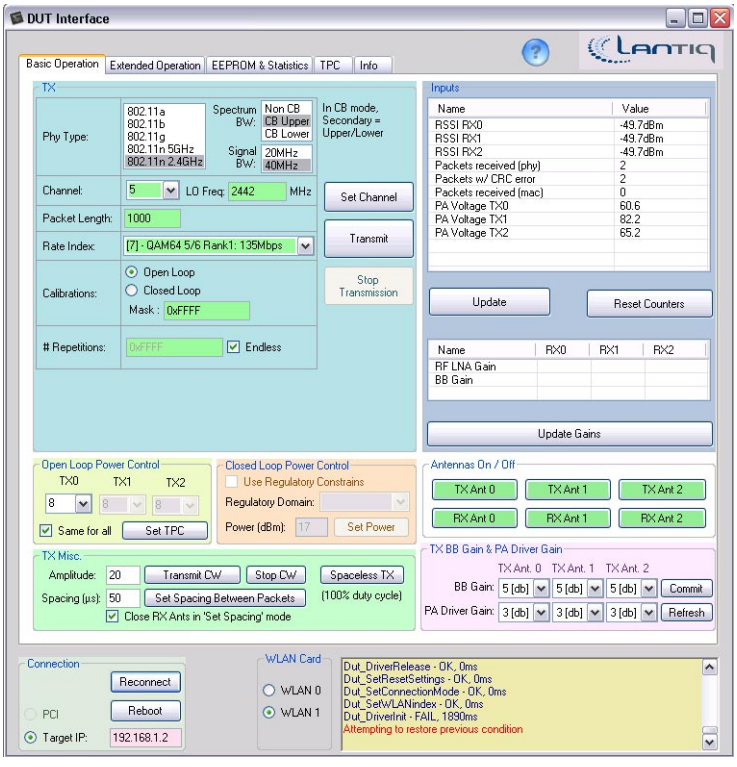


Figure 3 Main Screen of DUT GUI Application after an Unsuccessful Launch

Note: Please pay attention to the “There is no ping reply” message in the messages pane.

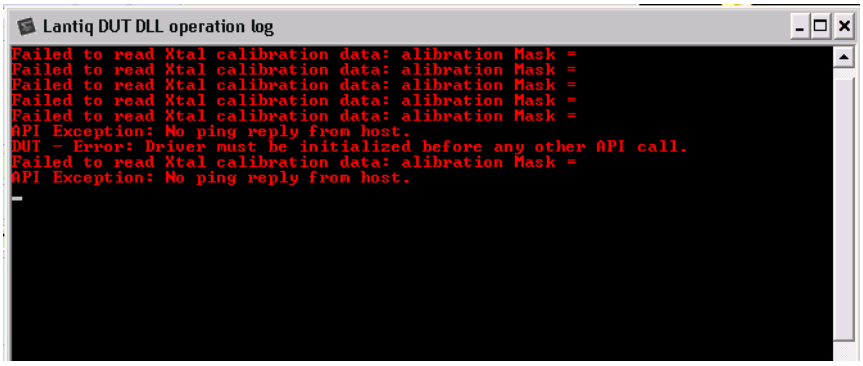


Figure 4 DUT DLL Log

Note: Please pay attention to the “No ping reply from host” message in the log pane.

4 Overview of the DUT GUI

The DUT GUI provides the following functions:

4.1 DUT GUI Features

- Change DUT operation to desired channel
- Turn on/off Rx/Tx antennas
- Receive 802.11a/b/g/n packets in nCB/CB Mode and at SISO/MIMO rates
- Transmit 802.11a/b/g/n packets in nCB/CB Mode and at SISO/MIMO rates
- Transmit in Open/Closed loop with Regulatory restrictions
- Read and burn the EEPROM/Flash
- Crystal frequency tuning tool

4.2 The Tab Bar

The DUT GUI includes five tabs:

- Basic Operation - Includes all functions needed to control the receiving and transmission for the DUT
- Extended operation - Includes calibration of the on-board Crystal and an option for future Antenna selection
- EEPROM & Statistics - Includes all functions needed to read and write to the EEPROM along with some advance options
- TPC - Includes specific calibration data information represented in the EEPROM/Flash
- INFO - Includes information about current system state, such as power-up self test results

Figure 5 DUT GUI Tabs

4.3 Log and Messages Pane

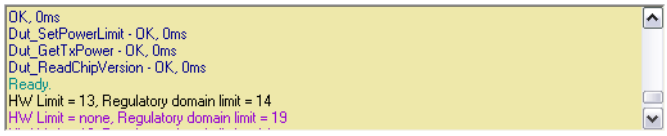


Figure 6 Log and Messages Pane

In the log pane, you can see the status of command execution. Problems are colored in red.

4.4 Calibration Data Screen

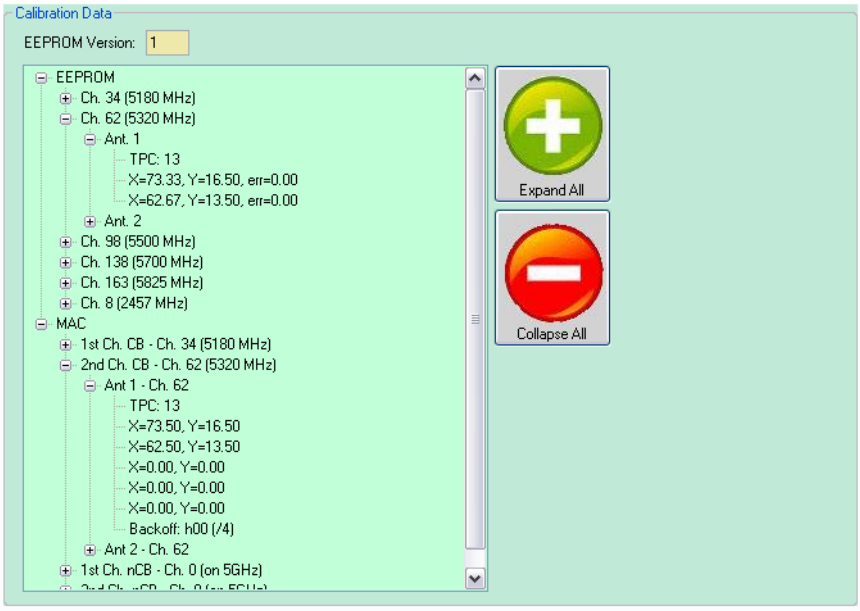


Figure 7 Calibration Data Screen

During production testing, TX power calibration is performed for enabling closed loop transmission.

The Calibration Data screen includes the information from the EEPROM which was collected during the calibration.

The information displayed is for debugging purposes only.

5 Using the DUT GUI

5.1 Connection Options

On the Basic Operation tab, the connection options are shown. Target IP should be provided. Reconnect and reboot buttons are available.

Wlan card 0 or 1 can be selected by pressing the desired radio button.

5.1.1 dut_cfg.txt Configuration Parameters

Before starting the connection, specific parameters should be checked in dut_cfg.txt file located in C:\Program Files\Lantiq\DUT

Reset Hardware Information

```
Reset_HW_Type0 = 64          //Wlan0 HW type (decimal value)
Reset_HW_Type1 = 49          //Wlan1 HW type (decimal value)
Reset_HW_Revision0 = D       //Wlan0 HW revision (decimal value)
Reset_HW_Revision1 = B       //Wlan1 HW revision (decimal value)
```

Set here the HW type and revision of your card. When the EEPROM/Flash is empty, DUT will use those values in order to load the correct progmodels. DUT supports two hardware types for supporting Dual band concurrent platforms.

Default WLAN index (0/1)

```
Default_WLAN_Index = 0       // Two Wlan units are supported 0 or 1
```

This value defines DEFAULT WLAN and may be overridden by Dut_SetWLANindex API call or Wlan card Wlan0 or Wlan1 radio buttons in DUT GUI.

Crystal Frequency Tuning

```
ENABLE_XTAL_CALIB = 1
```

1-Read XTAL tuning value from EEPROM/Flash and enable XTAL calibration

0-XTAL calibration is disabled

```
XTAL_CAL_BIAS = 0            // Siward is the default (76)
```

This is the value of the XTAL bias (7 msb of the register).

The following XTAL Bias values are supported:

Table 2 XTAL Bias Values

XTAL Vendor	XTAL Type	XTAL_CAL_BIAS decimal value
Siward	XTL541300A217002	76
Siward	XTL571100L137002	74
TXC	7M36090001	70
Kyocera	CX32255SBL0FLFA1	79

For example: `XTAL_CAL_BIAS = 70` should be used for TXC part.

Non-Volatile memory type

```
NV_MEMORY_TYPE0 = 1          // Wlan0 memory type
NV_MEMORY_TYPE1 = 1          // Wlan1 memory type
```

Select the memory type used for storing HW related data.

1 - eeprom, 2 - flash, 3 - efuse

Fast Operation

```
FAST_RESTART_METHOD = 1
0 - Regular Mode
1- Fast Restart mode
```

Note:

1. *Other parameters in dut_cfg.txt are used for debug and should not be modified.*
2. *In case improper hardware type was chosen, it may be required to reboot the DUT host (system to which the device is connected) before reconnection.*

5.2 Setting the DUT's Channel

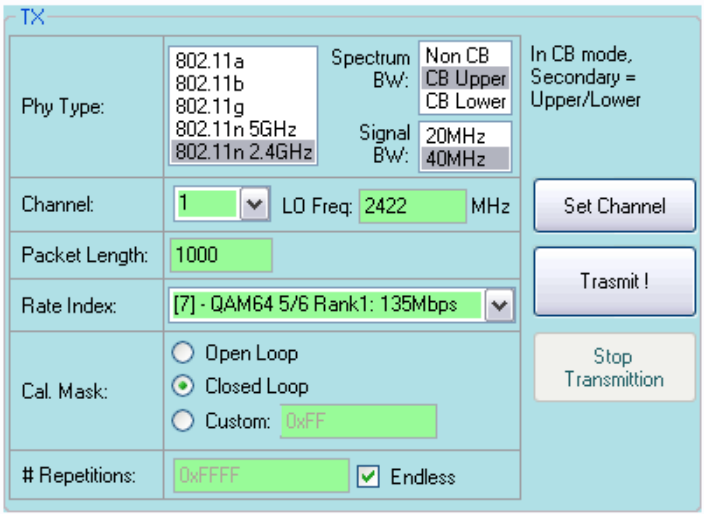


Figure 8 Channel Setting and Transmission Screen

When the DUT GUI is initialized, it sets the DUT to a channel according to the last run. You can select different parameters and set the DUT's channel using the following options:

- PHY Type - Used to select the operation band. PHY type also depends on the type of packets the DUT will transmit. However, if a PHY type of 2.4 Ghz is chosen (for example - 802.11b **[2]**) the DUT will also receive and analyze 802.11g and 802.11n packets.
- Spectrum BW (Band width) - Dictates the spectrum bandwidth of the transmission:
  - Non-channel Bonding (nCB) = 20 Mhz
  - Channel Bonding (CB) = 40 Mhz

If CB is used, the selected channel is the primary channel and you need to choose if the secondary channel will be on the lower or upper side of the primary channel. The DUT GUI automatically calculates the actual frequency of the DUT's LO according to the channel, the BW, and the location of the secondary channel

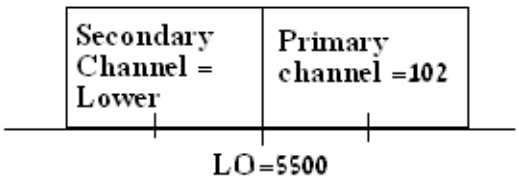


Figure 9 Example of Primary and Secondary Channel Selection in CB Mode

- Signal BW - When the 40Mhz spectrum BW is chosen, you can choose between CB (40 MHz) Tx or SSB (single side band) transmission.
  - In SSB mode, all Tx and Rx filters are on 40 MHz BW, LO is according to 40 MHz transmission but the actual transmitted signal is only in one side of the LO (according to the location of the primary channel)
  - The difference from the previous option of 20MHz is the location of the signal in comparison to the LO and the filters used.
- Channel - Select a channel according to the 802.11n standard definition of channel. LO Frequency will automatically be calculated according to BW definitions.
- Byte length - The packet length in bytes. The maximum length is 1570 bytes.
- Rate index - According to the MCS table at the 802.11a/b/g/n standards **[1]**/**[2]**/**[3]**/**[4]**. The list of available rate indexes will be according to the protocol type chosen.
- Calibration mask - This influences the initial calibration inside the RF chip after setting channel. The value should be set to 0xFFFF.
- Repetitions - How many times the packet should be transmitted.

Note: 0xFFFF (65535 in Decimal) = endless transmission

- Set Channel button - To instruct the DUT to change channel:
  - a.Set all the parameters.
  - b.Click Set Channel.
    When changing channel all transmissions will be stopped.
- Transmit button - To instruct the DUT to start transmitting:
  - a.Set all the parameters.
  - b.Click Transmit.
- Stop Transmission button - Use this to instruct the DUT to stop transmitting (only available when DUT is in Transmit mode).

5.3 Setting Output Power

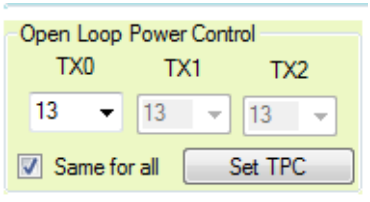


Figure 10 Open Loop Tx Setting

When transmission is used, you can select several standard and special configurations:

- Set TPC - This option is only available in open loop (see Calibration Mask details in **"Setting the DUT's Channel" on Page 16**.

TPC is one of the gain controls of the RF chip. Enter 1, to select transmission with the highest power, any number greater than 2 will lower the output power. The maximum value for TPC is 30 (lowest power). You can decide whether to apply the same TPC value to both transmitters or set independent values for each transmitter.

The output power in open loop mode can be changed before or during transmission.

5.4 Closed Loop Output Power Table

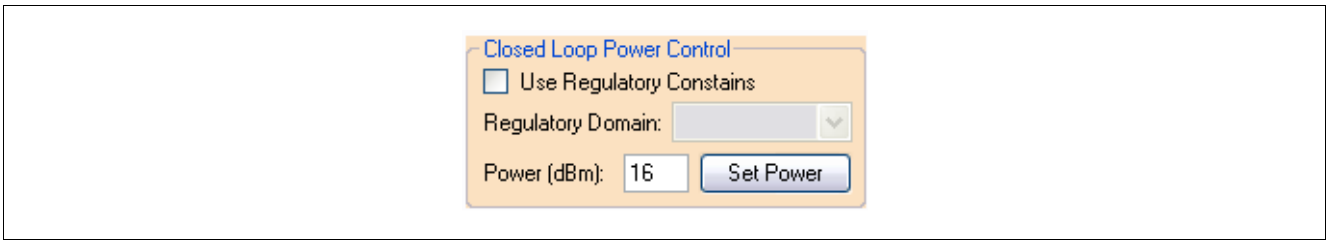


Figure 11 Closed Loop Tx Setting

- Set Tx power - In a closed loop transmission (see Calibration Mask details in **"Setting the DUT's Channel" on Page 16**, power is set automatically using pre-calibrated information stored in the EERPOM/flash. The information is stored in the EEPROM/flash during the TPC calibration method implemented by Litepoint's IQFact application.
- The Tx output power is configured automatically according to the rate (see tables below).
- Output power in closed loop can be changed before or during transmission.
- Close loop power can be set in 1/8 dB resolution example 16.125 dBm, 16.875 dBm etc.
- Power table according to the rate is found and can be adjusted in the file: dut_cfg.txt which is located in C:\Program Files\Lantiq\DUT.
- There are two options to set the default power per rate. Both are configured using the Use_TPC_Power parameter option in the dut_cfg.txt file: **Constant Power** and **"Back Off Power" on Page 19**.

5.4.1 Constant Power

Use_TPC_Power = 0

Power is determined according to the following table:

Table 3 Constant Power

MCS Rate	0	1	2	3	4	5	6	7
Const_Power_11a	17	17	16	15	15	15	15	15
Const_Power_11b	19	19	19	19	19	19	19	19
Const_Power_11g	19	19	18	18	17	17	17	17
Const_Power_11n5	17	16	16	15	15	15	15	15
Const_Power_11n24		18	18	17	17	17	17	17

*Note: For 802.11n **[4]**, rates 8-15 have the same power as rates 0-7.*

5.4.2 Back Off Power

Back off power from Max power in the EEPROM

Use_TPC_Power = 1

Power is determined according to the maximum power stored in the EEPROM minus the following back off. The back off can be modified differently for each rate:

Table 4 Back off Power

MCS Rate	0	1	2	3	4	5	6	7
TPC_Backoffs_11a	0	0	0	0	0	0	1	2
TPC_Backoffs_11b	0	0	0	0	0	0	0	0
TPC_Backoffs_11g	0	0	0	0	0	0	1	2
TPC_Backoffs_11n5	0	0	0	0	0	1	2	3
TPC_Backoffs_11n24	0	0	0	0	0	1	2	3

5.5 Using Regulatory Constraints in Closed Loop Tx Mode

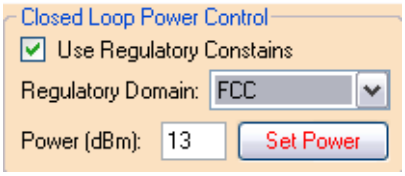


Figure 12 Regulatory Constraints in Closed Loop Tx Mode

For EMI testing, there is an option to limit the transmitted power according to the EMI standards.

To use this mode:

1. Consider the regulatory (see**"Regulatory Limitations" on Page 19**) and hardware (see **"Hardware Limitations" on Page 20**) limitations.
2. Select the Use regulatory constraints mode.

5.5.1 Regulatory Limitations

Regulatory limitations are imposed by the ETSI, FCC, and Telec standards for different channels and BW.

The reglimits.csv file is provided (in C:\Program Files\Lantiq\DUT\Regulatory_Constrains) with all the FCC, ETSI and Telec limitation for all the channels. The DUT GUI uses the information in this table to define the max output power allowed for each of the bands and frequencies. An example is shown in the following table.

Table 5 Reg Limits

Band	Domain	Class	Spacing	Channel	Power Limit	Sm Required
5	FCC	1	20	36	17	No
5	FCC	1	20	40	17	No
5	FCC	1	20	44	17	No

Table 5 Reg Limits (cont'd)

Band	Domain	Class	Spacing	Channel	Power Limit	Sm Required
5	FCC	1	20	48	17	No
5	FCC	2	20	52	24	Yes
5	FCC	2	20	56	24	Yes
5	FCC	2	20	60	24	Yes
5	FCC	2	20	64	24	Yes

5.5.2 Hardware Limitations

The hardware limitation are different for every type of board. During EMI certification, for example, an RF board might need to lower the output power even more than required in the regulation for the specific channel used because of band edge, 2nd harmonic and other limitations.

The DUT GUI uses information from the supplied limits.ini file according to the HW Type and HW Version information found in the EEPROM during startup. All known HW types are listed at the top of the ini file.

Example:

```
[HWTypes]
0x301a_0x8006_0xc4_0x45= GPB-208
0x301a_0x0007_0x30_0x43 = GPB-304
0x301a_0xC007_0x40_0x44= Easy388 family board (WRX RFIC)
```

If there is no HW limit for the HW used in the specific channel, only the regulatory limits will be taken into account.

DUT GUI will not allow users to change the desired Tx power value more than the limitation described above allow.

After the desired power is changed, click Set power for the change to take effect.

5.6 Changing BB and PA Driver Gains

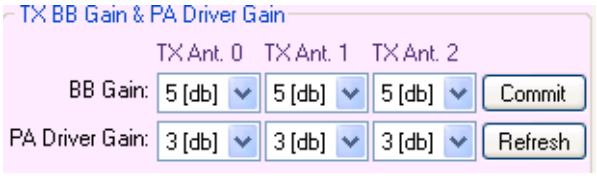


Figure 13 BB and PA Driver Gain Setting

The DUT GUI allows you to control the TPC attenuator in the RF chip and two additional RF chip internal amplifiers:

- Base band gain (BB gain) - In the BB side of the RF chip there is an option to boost dB gain by 0, 2.5, or 5
- PA driver gain - In the RF side of the RF chip there is an option to gain 0 or 3 dB using the PA driver gain control

The system automatically determines the initial value of both gains and this might differ between frequencies and bands.

Note: This Gain change option is applicable for XWAY™ WAVE300 devices only.

5.7 Transmitting in CW Mode and Changing IFS Section

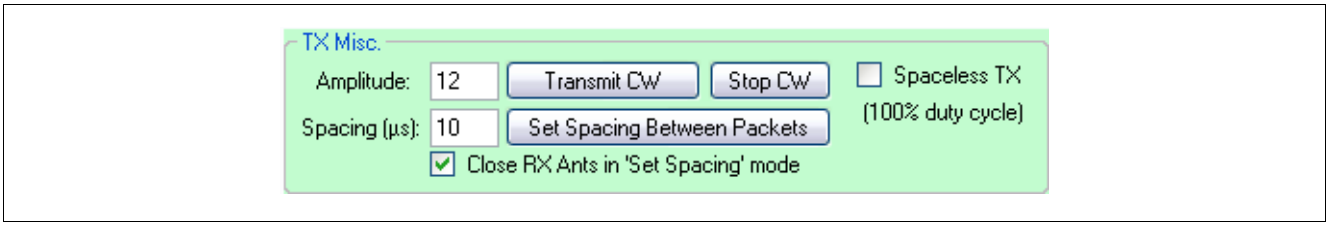


Figure 14 Transmitting in CW Mode and Changing IFS Section

- CW mode - To set the DUT to transmit in CW mode, define the CW amplitude and click Transmit CW. To stop the transmission, click Stop CW.

Note: The amplitude is just an index and is not in dBm. The higher the index used, the higher the CW's amplitude.

- Changing spacing - To change the IFS (inter frame spacing), set the spacing in µSec and click Set Spacing Between Packets. When changing the spacing, Rx Antennas are closed to verify that the IFS isn't determined also because of received packets for different 802.11 station in the area.
- Space-less transmission - When this mode is selected, the transmission coming out will be continual, meaning there will be no idle time - 100% Duty cycle. This is not a standard mode of operation and some time is required for certification testing.

5.8 Setting Antenna Configuration

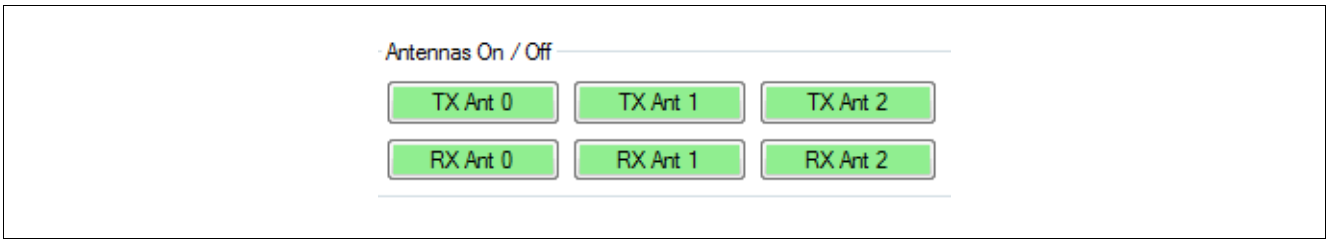


Figure 15 Antenna On/Off Section

Lantiq's WiFi solution includes three transmit and three receive antenna. You can turn off one or more of the antenna by clicking the relevant button as shown above.

5.9 Rx Information and PA Detector Information

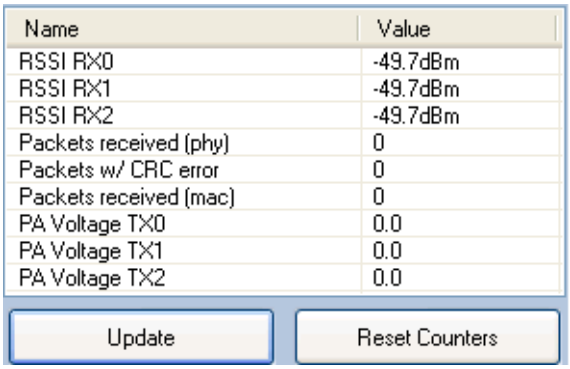


Figure 16 RSSI, Packets Counter and PA Detector Information

When you click Update, the following are read from the DUT:

- RSSI values (signal strength) for each of the three Rx antennas (in dBm)
- Packets received counters (PHY level) - There are two readings:
  - the upper is for the number of packet received in the modem
  - the lower is for the number of packets received in the modem with CRC error.
- Packets received counter (MAC) - This counter is not used.
- PA voltage - Returns the power indication from the PA detector during transmissions.

The PA detector information enters the BB chip through an 8 bit A/D. The reading in the DUT GUI is an average of several readings and the numbers are on a scale of 0-255. Usually, each step represents approximately 10 mV.

5.10 Chip Gain Information

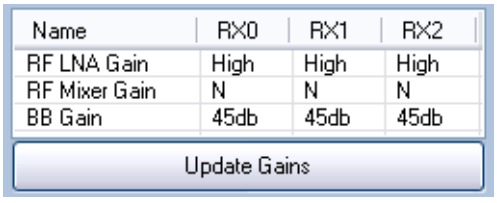


Figure 17 Chip Gain Information

When you click Update Gains, the following are read from the DUT:

- RF LNA Gain
- RF Mixer Gain
- BB Gain

Note: The gain information is applicable for XWAY™ WAVE300 devices only.

5.11 DUT and Chip Versions

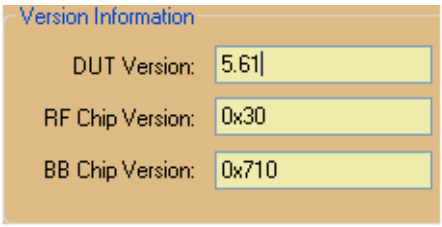


Figure 18 Version Information

During startup, the DUT GUI reads the RF and BB chip versions from the relevant HW registers in the chips and displays it together with the DUT GUI application version.

The RF chip version 0x30 indicates XWAY™ WAVE300 (PSB 8301, PSB 8304) RFIC

The BB chip version 0x710 indicates XWAY™ WAVE300 (PSB 8221, PSB 8231) BBIC

5.12 Reading and Writing the EEPROM/Flash

5.12.1 Configuring EEPROM/Flash Area Size

EEPROM/Flash area size may be configured via dut_cfg.txt file.

It is possible to specify EEPROM/Flash area size in bytes by the means of parameter EEPROM_Size_Bytes. The default value is 1024 bytes.

5.12.2 Burning a File to EEPROM/Flash

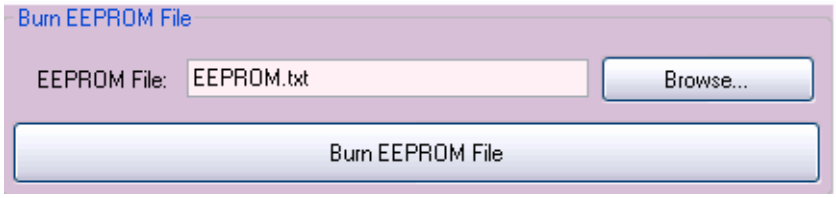


Figure 19 EEPROM/Flash Burning Section

The EEPROM/Flash contains important information for operation of the module. The information includes:

- PCI information (including Vendor ID and Device ID).
- HW Type and HW version information - To determine which firmware to load to the module.
- TPC calibration information.

Attention: A module will not work without correct values in the EEPROM/Flash for all three areas mentioned above.

The EEPROM/Flash is usually burned only once, during the production testing. Information burned into the EEPROM/Flash consists of a text file in the following form:

```
<Address><data>
```

Example:

```
0000    FC1B
0002    CD00
0004    0000
0006    0100
```

Attention: Burning the wrong information into the EEPROM/Flash can cause the module to stop working without the possibility to reverse the damage.

5.12.3 Working with Serial Numbers and MAC Addresses

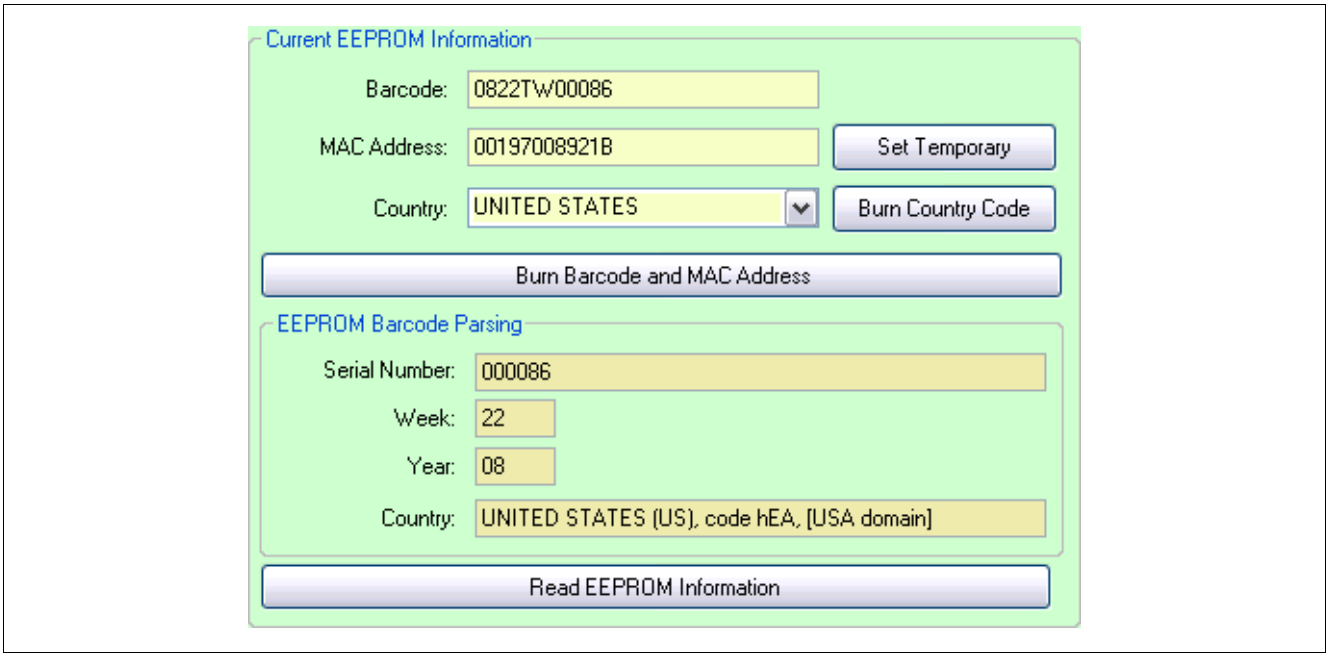


Figure 20 MAC Address and S/N Information R/W

The EEPROM/Flash includes MAC address, country code and serial number information burned in the production testing and assembly.

Using the DUT GUI, the following information can be read from the EEPROM/Flash:

- MAC address - is in 12 hexadecimal digits
- Serial number can be in 2 following formats

- YYWWMMNNNNNCC where
  - YY - Year of manufacture
  - WW - Week of manufacture
  - MM - Two letter initials of assembly house (this information is not stored in the EEPROM/Flash)
  - NNNNN - Five digits of sequential serial number
  - CC (optional) - Short country name (two-letters), as listed in the 'Country' drop list, used for regulatory domain limitations.
  - For full country code list see **Appendix C - Country Codes**
    Example: 0825ML12345US

- KKKKKYMNEXXXXXX
  - KKKKK - 5 digits of module name.

- Y - Year of manufacture
- M - Month of manufacture (A=October, B=November, C= December)
- N - board revision
- E - Regulatory domain (Example: E= ETSI)
- XXXXXX- Five digits of sequential serial number
  Example: AN66085NE123456

The following options are available in this section:

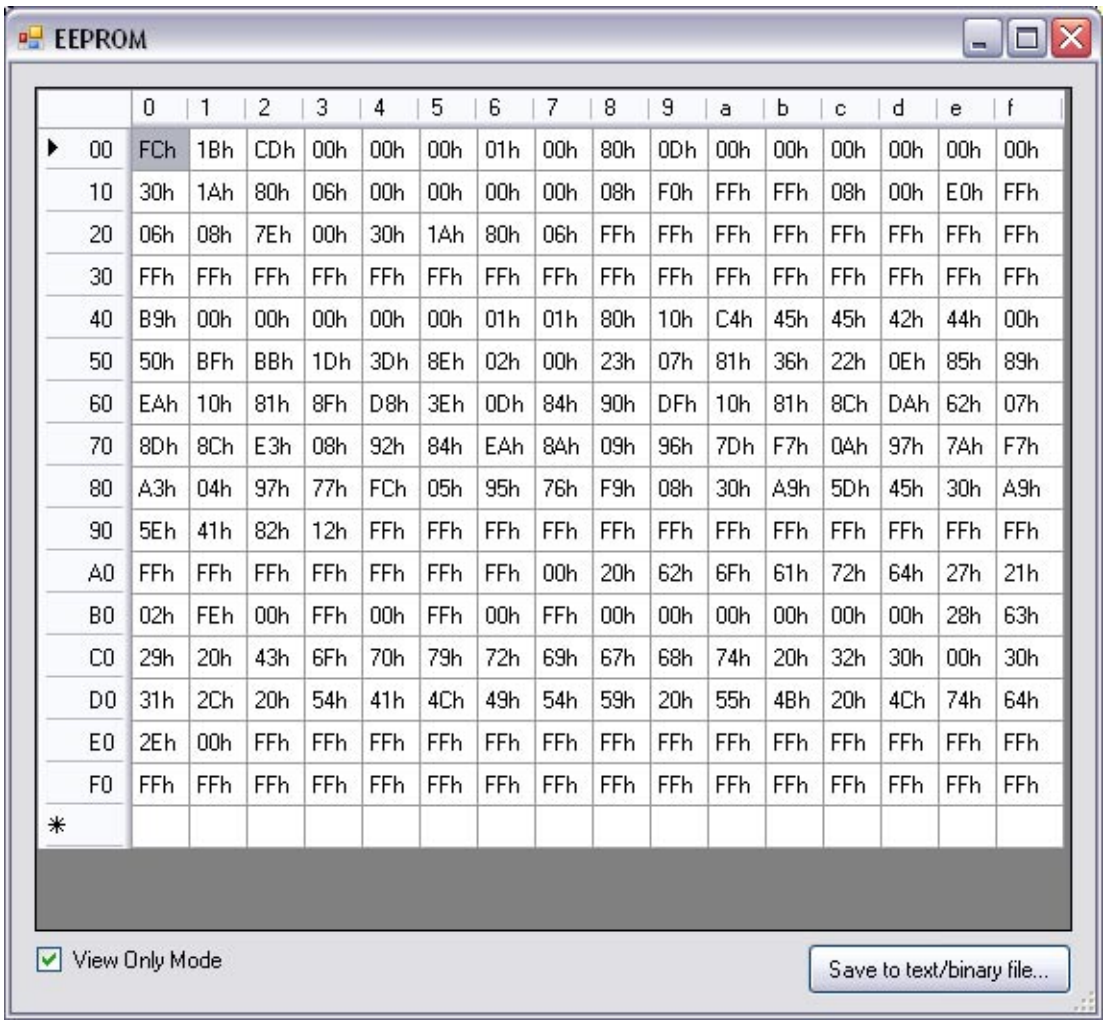
- **Read EEPROM/Flash Information** - Read current S/N, country code and MAC address information from the EEPROM/Flash
- **Burn Barcode and MAC address** - Write the S/N (barcode) and MAC address information provided by the user to the EEPROM/Flash
- **Set temporary** - Give the module a temporary MAC address. This MAC address will not be saved into the EEPROM/Flash but can be used to filter received packets without this MAC address.
- **Burn Country Code** - Write the chosen country code to the EEPROM/Flash.

The Rx packets counter for packets received with this MAC address can be found in the first tab of the application in the “Rx information” section.

5.12.4 Changing Raw Content

To view EEPROM/Flash raw data:

1. From the third tab, click View / Change/ Save EEPROM…
   The following screen appears.



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	FCh	1Bh	CDh	00h	00h	00h	01h	00h	80h	0Dh	00h	00h	00h	00h	00h	00h
10	30h	1Ah	80h	06h	00h	00h	00h	00h	08h	F0h	FFh	FFh	08h	00h	E0h	FFh
20	06h	08h	7Eh	00h	30h	1Ah	80h	06h	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh
30	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh
40	B9h	00h	00h	00h	00h	00h	01h	01h	80h	10h	C4h	45h	45h	42h	44h	00h
50	50h	BFh	BBh	1Dh	3Dh	8Eh	02h	00h	23h	07h	81h	36h	22h	0Eh	85h	89h
60	EAh	10h	81h	8Fh	D8h	3Eh	0Dh	84h	90h	DFh	10h	81h	8Ch	DAh	62h	07h
70	8Dh	8Ch	E3h	08h	92h	84h	EAh	8Ah	09h	96h	7Dh	F7h	0Ah	97h	7Ah	F7h
80	A3h	04h	97h	77h	FCh	05h	95h	76h	F9h	08h	30h	A9h	5Dh	45h	30h	A9h
90	5Eh	41h	82h	12h	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh
A0	FFh	FFh	FFh	FFh	FFh	FFh	FFh	00h	20h	62h	6Fh	61h	72h	64h	27h	21h
B0	02h	FEh	00h	FFh	00h	FFh	00h	FFh	00h	00h	00h	00h	00h	00h	28h	63h
C0	29h	20h	43h	6Fh	70h	79h	72h	69h	67h	68h	74h	20h	32h	30h	00h	30h
D0	31h	2Ch	20h	54h	41h	4Ch	49h	54h	59h	20h	55h	4Bh	20h	4Ch	74h	64h
E0	2Eh	00h	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh
F0	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh	FFh

Figure 21 EEPROM/Flash Raw Data

The information is shown as two hexadecimal digits for each address (hexadecimal marked by the 'h' after the two digits). For example, 1Ah (equivalent to 0x1A).

To edit the EEPROM/Flash raw data:

1. Clear the View Only Mode checkbox.
2. Edit the EEPROM/Flash content for each cell separately.
   Burning is performed when the cursor moves from the changed cell
3. To save all the EEPROM/Flash content to a file, click Save to text/binary file.
   The file can be loaded with the “burning EEPROM file” function in the second application tab

5.13 XTAL Calibration

This form can be found in the Extended operation tab.

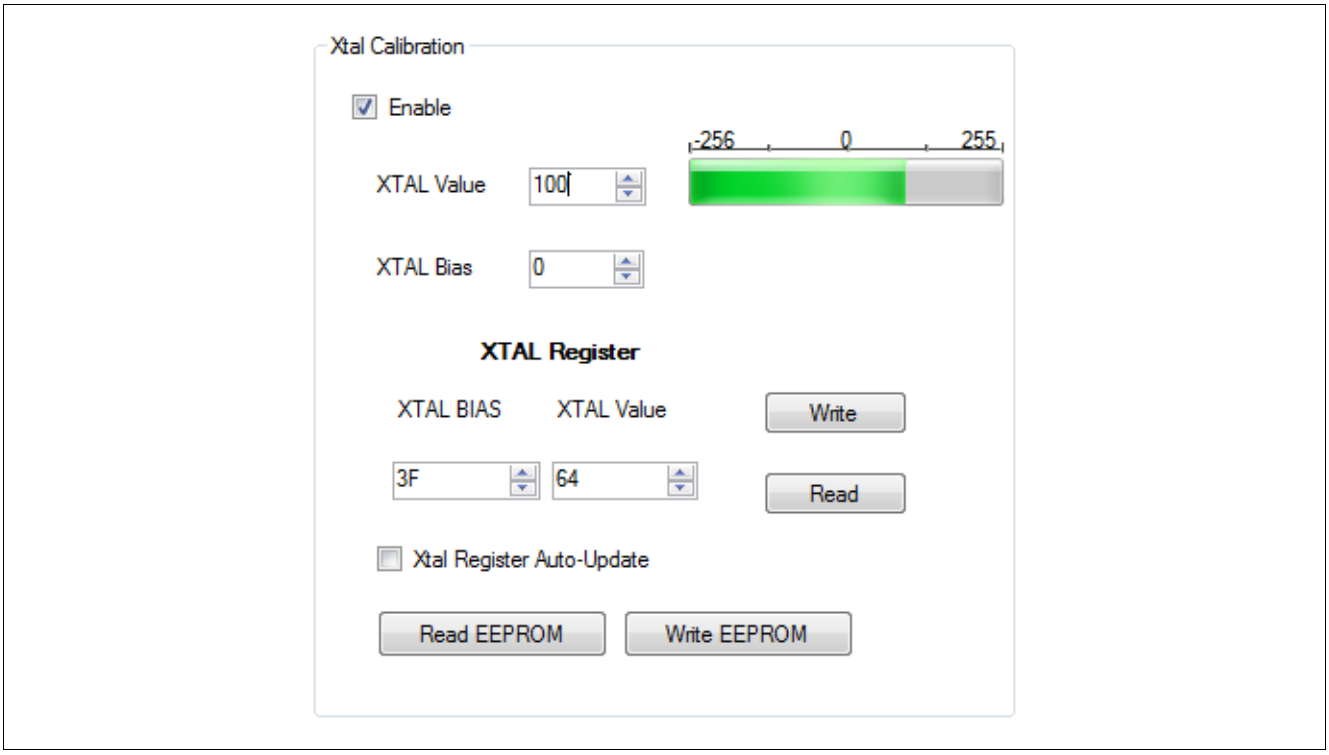


Figure 22 XTAL Calibration

Two XTAL configuration fields are available:

- XTAL Value - tuning value register
- XTAL Bias - drive strength value (the XTAL Bias field is supported in XWAY™ WRX312/313 RFIC only)

These fields can be read and written to the RFIC and EEPROM/Flash and used for proper XTAL operation.

The following XTAL Bias values are supported:

Table 6 XTAL Bias Values

XTAL Vendor	XTAL Type	XTAL_CAL_BIAS decimal value
Siward	XTL541300A217002	76
Siward	XTL571100L137002	74
TXC	7M36090001	70
Kyocera	CX32255SBL0FLFA1	79

6 Available Functions in mt_dut.dll

All Int functions will return a value:

- If a function is used to receive information (counter etc.) the value will be returned
- In all other cases, the function will return “1” for success or “0” for fail

Parameters for all functions have names as defined in the header file “DUT_API.h”. For example:

```
#define DUT_MODE_B 1
#define DUT_MODE_N 2
```

6.1 Init Process

```
Dut_SetConnectionMode(int connectionType, const char* optIPaddr);
// Set the connection mode to the device. 0=PCI, 1=UART,2=SPI,3=LINUX.
IP addr should be specified for linux connections, can be null otherwise This
function should be called BEFORE Dut_DriverInit.

Int Dut_DriverInit(void) // Return 1 after success

Int Dut_DriverVersion(double *version); // returns the version to the location of
the pointer

Int Dut_SetWLANindex(int wlanIndex);
// In case of concurrent dual-band board, use this function before connection or
reconnection in order to choose the needed WLAN card index - 0 or 1, where 0 is the
default. Call this function AFTER calling Dut_SetConnectionMode

Dut_BistRes(int* BistRes); //Reads results of the BIST
```

6.2 Calibration Process and Tx Tests

```
Int Dut_SetChannel(int Mode, int band, int CBMode, int channel, BYTE CalibrationMask)
// mode={a,b,g,n}, Band={2400,5000}, CBMode={20,40} channel_index (not the actual
frequency) CalibrationMask - in Hexadecimal

Int Dut_ChangeScramblerMode(int mode) // {0 ,1}

Int Dut_SetTPC (int TPC) // {25,..,2}

Int Dut_TxPackets (int Mode, int RateIndex, , int CBMode, int Length, int
repetitions) // mode={a,b,g,n}, rate={MSC table}, CBMode={20,40}, length in byte,
repetitions - number of packets or 0xFFFF for endless transmission.
Int Dut_EndTxPackets (Void)

Int Dut_ReadVoltageValue (double* voltages) // Returns the power values in Volts for
both Tx antennas to the pointer of the value array.
```

Int Dut_SetPower(int power); // Sets the Tx power for close loop only. The value is the desired Tx power per antenna in dBm. The function itself does not change the power: you must call this function before calling Dut_TxPackets.

Int Dut_SetPowerFloat(float power); // The float version can received more accurate values (in *1/8db)

6.3 Burning the EEPROM

Int Dut_BurnEEPROM (DWORD Address, DWORD length, const BYTE* value) // Burning information into the EEPROM.

- The address is in hexadecimal.
- The length is in bytes.
- The value is the string in hexadecimal.

Examples:

```
Dut_BurnEEPROM (C4, 2, FFAF)
Dut_BurnEEPROM (2D, 4, 01F499A3)
```

Int Dut_ReadEEPROM (DWORD address, DWORD length,BYTE* data); // Read data from the EEPROM, at a specified address. Length is the number of bytes.

Int Dut_Burn_MAC_Address (const BYTE *mac_address) // Used to burn the MAC address (length of 6 bytes).

Int Dut_Burn_Date (int year, int week) // Used to burn the date of the mPCI production.

Int Dut_Burn_SN (DWORD SN) // Used to burn the S/N (3 Bytes)

Int Dut_Burn_File (const char* filename) // Used to burn file with address and information to the EEPROM. The file should be in the form of:
<Address> <Data>

Example:

```
0040 AAFF
```

Dut_Burn_File can also be called instead of Dut_DriverInit, which in this the file will be burned before the init process first reads the EEPROM.

Int Dut_BurnInfo(const char* barcode, const BYTE* macAddr, const char* filename) // One function that burns the S/N from the barcode, the MAC address, and the EEPROM file.

Int Dut_Burn_Cal_Data (const Dut_CalDataStruct* pCarStr) // Used to burn the calibration data. The structure is in the form of:

	Freq 1	Freq 2	Freq 3	Freq 4	Freq 5	Freq 6
uint Channel {0-255}						
Bol Band (0=2.4, 1= 5)						
uint Ant0 TPC {0-31}						
uint Ant0 Max power{0-31}						
double a_0						
double b_0						
uint Ant1 TPC {0-31}						
uint Ant1 Max power{0-31}						
double a_1						
double b_1						

Figure 23 Burn Calibration Structure

6.4 EEPROM Burning New Calibration Method Function

```
extern MT_RET PASCAL Dut_EEPROM3_Burn_Cal_Data(const Dut_CalDataStruct3*
inCalArray, Dut_CalDataStruct3* outCalArray, int numStructures);

// Burn calibration data using the Dut_CalDataStruct3 for EEPROM Version 3 (defined
// above)
// int numStructurs - # of Dut_AntCalDataStruct3 structures in calArray (2.4 + 5 GHz)
// outCalArray is an optional parameter (can be NULL). If specified, the function
// fills the outCalArray with the linear correlation (LR) results.
// You can use the function Dut_EEPROM3_Test_Cal_Data to verify the LR results for
// part or all of the structures before burning.

extern MT_RET PASCAL Dut_EEPROM3_Test_Cal_Data(const Dut_CalDataStruct3*
inCalArray, Dut_CalDataStruct3* outCalArray, int numStructures);
// Same as Dut_EEPROM3_Burn_Cal_Data but without burning to the EEPROM.
```

6.5 Sensitivity Check

```
Int Dut_SetMACAddress(const BYTE* macAddr);
// Sets the MAC address for the testing procedure. Does NOT burn to EEPROM. macAddr
is a BYTE[6] array. This function can and should be called before the call to
Dut_DriverInit(). It sets the Rx filter in the lower Mac for packets only in this
MAC address.

Int Dut_SetRxAntenna(int rx_antenna_mask);
```

```
// options = {001, 010, 100, 011, 101, 110, 111}. Returns 1 following a success.

Int Dut_PHYPacketsCounter (unsigned int* packetsCounters)
// Returns the current value of the  received packet counter and CRC error counter.
Read before and after transmission. Values will be returned in decimal to the
location of the array. The array should be composed of two values of uint type. The
first is for the packet counter and the second is for the CRC error counter. This
function returns the counters from the PHY layer and does not filter packets
according to MAC address.

Int Dut_MACPacketsCounter (unsigned int* packetsCounters)
// As above, but returns the counter from the lower MAC level. Packets are filtered
according to the MAC address set in the MAC (by using the function Dut_SetMACAddress.

Int Dut_ResetPacketsCounter ()
// Resets the packets and CRC error counters.

Int Dut_RSSIVector (double* RSSI) // Returns the RSSI value of three Rx antennas at
the time of the call to the location of the array. The array should be composed of
3 values of double type. Values are in dBm.
```

6.6 Finalize the Test

```
Int Dut_DriverRelease(void);
// Used to terminate your driver control when you have allocated some resources, e.g.
memory. Can be just a dummy return if your driver has nothing to release.
```

6.7 Others

```
Int Dut_WriteReg(DWORD address, DWORD Data, DWORD mask); // For the BB chip

Int Dut_ReadReg(DWORD address, DWORD* pData, DWORD mask); // For the BB chip

Int Dut_ReadRFReg(DWORD address, DWORD* pData, DWORD mask); // For the RF chip

Int Dut_WriteRFReg(DWORD address, DWORD data, DWORD mask); // For the RF chip

Int Dut_ReadChipVersion(DWORF* RF_chip_version, DWORD* BB_chip_version)

Int Dut_ChangeRFChipMode(DWORD mode);
// Changes the mode of operation of the RF chip. In all modes which are not listen,
the gen risc is disabled. To re-enable, you must return to listen mode.
```

Possible options are:

```
#define DUT_RF_MODE_LISTEN 0xF77 // Tx turned off
#define DUT_RF_MODE_STANDBY 0x800 // Only LO is kept on
#define DUT_RF_MODE_POWERDOWN 0
```

```
Int Dut_ReadRFGain(BYTE* RF_Gain_Values); // Read RF Gain - RF_Gain_Values must be
an array of 6 bytes with:
0 - RX0_LNA_Gain
1 - RX0_MIXER_GAIN_6DB
2 - RX1_LNA_Gain
3 - RX1_MIXER_GAIN_6DB
4 - RX2_LNA_Gain
5 - RX2_MIXER_GAIN_6DB

Int Dut_ReadBBGain(int* BB_Gain_Values); // Read BB Gain - BB_Gain_Values must be an
array of three integers for each antenna - returns total of LPF1+LPF2+VGA in db

Int Dut_ReadDetector(BYTE* detector_values); // Read Detector - detector_values must
be an array of four bytes:
0 - PLL_Lock
1 - THD0 (Threshold detector RX0 output)
2 - THD1 (Threshold detector RX1 output)
3.      - THD2 (Threshold detector RX2 output)
        "Int Dut_ReadVoltageValueEx(double* pVoltage, int numSamples) // Same as
ReadVoltageValue, numSamples - # of samples for each antenna for averaging. Default
value for numSamples (used in Dut_ReadVoltageValue) function is 35
```

Tx Gains Set/Get for each TX antenna:

```
Int Dut_SetTxGains(int* bbGains, int* driverGains);
Int Dut_GetTxGains(int* bbGains, int* driverGains);

// Parameters:
// bbGains - array of 2 ints (for each TX ant). Each value can be 0 for 0[db], 1 for
2.5[db] and 2 for 5[db]
// driverGains - array of 2 ints (for each TX ant). Each value can be 0 (Low) or 1
(High)

Dut_StopRISC() // Stops the GenRISC processor (PHY)
Dut_StartRISC() // Starts the GenRISC processor (PHY)
```

7 Appendix A - PCI EEPROM Address Map

The eeprom structure includes four major sections:

1. Executive control Region (00-04h)
   The first two words are the eeprom signature. The signature is a known number which indicates valid eeprom information (e.g. not empty).
2. PCI spec RunTime registers (05-0Bh)
   These registers are the PCI configuration registers which must be in any PCI device. These registers with combination of the CIS (see section 4), define the HW installed at the PCI bus. This region is restricted for PCI HW definition native needs.
3. RunTime registers - (0C-3Fh)
   These registers are used as default configuration for the PCI
   Initiator and target HW. It includes Device and Vendor Id, shared RAM address and configuration space address.
4. Initialization section - (40-F8h)
   This section is used to save initialization parameters as Card type and version, MAC address, serial number, calibration data and other Metlalink's native information.

Table 7 EEPROM Address Map

Data	Address
Header	0x40-0x47
HW configuration Sub Section type	0x48
HW configuration Sub Section length	0x49
HW type	0x4A
HW revision	0x4B
Country code (for Regulation domain) See APPENDIX F for full country code list	0x4C
BB chip ID	0x4D
RF chip ID	0x4E
MAC address	0x4F-0x54
Serial number	0x55-0x57
Week and Year	0x58-0x59
TPC Calibration data Sub Section type	0x5A
TPC Calibration data Sub Section length	0x5B
TPC Calibration data	0x5C - 0xF8

Note: Modification of all sections off the EEPROM is forbidden and could result in initialization failure.

8 Appendix B - Example Code for mt_dut.dll Usage

Use the mt_dut.dll to develop automatic applications to control the DUT. The following files are available to you (in C:\Program Files\Lantiq\DUT\Developers):

- dut_api.h
- mt_types.h
- MT_DUT.lib

Example code:

```
This C source code defines the functions used by the supplied command line demo
application.
The Dut_XXX functions are declared in the supplied dut_api.h according to the
specifications, and you can also use the supplied MT_DUT.LIB for easier access to
the MT_DUT.DLL.

#include "../dut_api.h" // Set to your path of the dut_api.h file
#include <stdio.h>
#include <stdlib.h>

/// Prints the API error in case ret is MT_RET_FAIL
int TryRunning(int ret)
{
       if (ret == MT_RET_FAIL) printf("%s\n", Dut_GetLastError());
       return ret;
}

/// Prints the driver or API version
int DriverVersion()
{
       double version;
       Dut_DriverVersion(&version);
       printf("Driver version: %.2f\n", version);
       return MT_RET_OK;
}

// Initialize the driver (in our command line application, it is called
automatically)
int DriverInit()
{
       return TryRunning(Dut_DriverInit());
}

// Releases the driver, and exits the application
int DriverRelease()
{
       if (!TryRunning(Dut_DriverRelease())) return MT_RET_FAIL;
       exit(MT_RET_OK);
       return MT_RET_OK;
}
```

```
int SetChannel(int Mode, int band, int BW, int channel, BYTE calibrationMask)
{
	return TryRunning(Dut_SetChannel(Mode, band, BW, channel, calibrationMask));
}

int ChangeScramblerMode(BYTE mode)
{
	return TryRunning(Dut_ChangeScramblerMode(mode));
}

int SetTPC(int TPC)
{
	return TryRunning(Dut_SetTPC(TPC));
}

int TxPackets(int mode, int rateIndex, int BW, int length, int repetitions)
{
	return TryRunning(Dut_TxPackets(mode, rateIndex, BW, length, repetitions));
}

int EndTxPackets()
{
	return TryRunning(Dut_EndTxPackets());
}

int BurnEEPROM(DWORD address, BYTE value)
{
	return TryRunning(Dut_BurnEEPROM(address, 1, &value));
}

int ReadEEPROM(DWORD address)
{
	BYTE value;
	if (!TryRunning(Dut_ReadEEPROM(address, 1, &value))) return MT_RET_FAIL;
	printf("Got %02Xh from the EEPROM\n", (int)value);
	return MT_RET_OK;
}


int SetRxAntenna(int mask)
{
	return TryRunning(Dut_SetRxAntenna(mask));
}

int SetTxAntenna(int mask)
{
	return TryRunning(Dut_SetTxAntenna(mask));
}


int MacPacketsCounter()
```

```
{
   DWORD PacketsCounter;
    if(!TryRunning(Dut_MacPacketsCounter(&PacketsCounter))) return
MT_RET_FAIL;
    printf("MAC PacketCounter = %d\n",(int)PacketsCounter);
    return MT_RET_OK;
}

int PhyPacketsCounter()
{
       int i;
       double PacketsVector[2];
       if(!TryRunning(Dut_PacketsCounter(PacketsVector))) return MT_RET_FAIL;
       for (i=0; i<2; ++i)
              printf("PacketCounter %d=%f\n", i, PacketsVector[i]);
       return MT_RET_OK;
}


int ResetPacketsCounter()
{
       if(!TryRunning(Dut_ResetPacketsCounter())) return MT_RET_FAIL;
       return MT_RET_OK;
}

int RSSIVector()
{
       int i;
       double RSSIVector[3];
       if (!TryRunning(Dut_RSSIVector(RSSIVector))) return MT_RET_FAIL;
       for (i=0; i<3; ++i)
              printf("RSSI Ant %d=%f\n", i, RSSIVector[i]);
       return MT_RET_OK;
}

// Directly writes a masked 32-bit value to the firmware memory space.
int WriteReg(DWORD address, DWORD data, DWORD mask)
{
       return TryRunning(Dut_WriteData(address, data, mask));
}

int ReadReg(DWORD address, DWORD mask)
{
       DWORD data = 0;
       if (!TryRunning(Dut_ReadData(address, &data, mask))) return MT_RET_FAIL;
       printf("Received data: 0x%08X\n", data);
       return MT_RET_OK;
}

int ReadVoltageValue(int antenna)
{
```

```
	double voltage = 0.0;
	if (!TryRunning(Dut_ReadVoltageValue(antenna, &voltage))) return
MT_RET_FAIL;
	printf("Antenna voltage: %f\n", voltage);
	return MT_RET_OK;
}

int BURN_DATE(int year, int week)
{
	return (TryRunning(Dut_Burn_Date(year, week)));
}

int BURN_MAC_ADDRESS(BYTE b0, BYTE b1, BYTE b2, BYTE b3, BYTE b4, BYTE b5)
{
	BYTE macAddr[6] = {b0, b1, b2, b3, b4, b5};
	return (TryRunning(Dut_Burn_MAC_Address(macAddr)));
}

int BURN_CAL_DATA(int channel, int band, int TPC0, int maxPower0, double a0, double
b0)
{
	int i;
	Dut_CalDataStruct calData;
	calData.band = band;
	calData.channel = channel;
	for (i=0; i<2; ++i) // In this example, we use the same variables for ant0
and ant1
	{
		calData.ants[i].TPC = TPC0;
		calData.ants[i].a = a0;
		calData.ants[i].b = b0;
		calData.ants[i].maxPower = maxPower0;
	}
	return (TryRunning(Dut_Burn_Cal_Data(&calData)));
}

// Burn an example file to the EEPROM
int BURN_FILE()
{
	return (TryRunning(Dut_Burn_File("eeprom_file.txt")));
}

// Burns the serial number to the EEPROM
int BURN_SN(unsigned int SN)
{
	return (TryRunning(Dut_Burn_SN(SN)));
}
```

9 Appendix C - Country Codes

Table 8 Country Codes

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
1	1	AFGHANISTAN	AF	Europe
2	2	ÅLAND ISLANDS	AX	Europe
3	3	ALBANIA	AL	Europe
4	4	ALGERIA	DZ	Europe
5	5	AMERICAN SAMOA	AS	Europe
6	6	ANDORRA	AD	Europe
7	7	ANGOLA	AO	Europe
8	8	ANGUILLA	AI	USA
9	9	ANTARCTICA	AQ	Europe
0A	10	ANTIGUA AND BARBUDA	AG	USA
0B	11	ARGENTINA	AR	USA
0C	12	ARMENIA	AM	Europe
0D	13	ARUBA	AW	USA
0E	14	AUSTRALIA	AU	Japan
0F	15	AUSTRIA	AT	Europe
10	16	AZERBAIJAN	AZ	Europe
11	17	BAHAMAS	BS	USA
12	18	BAHRAIN	BH	Europe
13	19	BANGLADESH	BD	Japan
14	20	BARBADOS	BB	USA
15	21	BELARUS	BY	Europe
16	22	BELGIUM	BE	Europe
17	23	BELIZE	BZ	USA
18	24	BENIN	BJ	Europe
19	25	BERMUDA	BM	USA
1A	26	BHUTAN	BT	Japan
1B	27	BOLIVIA	BO	USA
1C	28	BOSNIA AND HERZEGOVINA	BA	Europe
1D	29	BOTSWANA	BW	Japan
1E	30	BOUVET ISLAND	BV	Japan
1F	31	BRAZIL	BR	USA

Table 8 Country Codes (cont'd)

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
20	32	BRITISH INDIAN OCEAN TERRITORY	IO	Japan
21	33	BRUNEI DARUSSALAM	BN	Japan
22	34	BULGARIA	BG	Europe
23	35	BURKINA FASO	BF	Japan
24	36	BURUNDI	BI	Europe
25	37	CAMBODIA	KH	Europe
26	38	CAMEROON	CM	Europe
27	39	CANADA	CA	USA
28	40	CAPE VERDE	CV	Japan
29	41	CAYMAN ISLANDS	KY	USA
2A	42	CENTRAL AFRICAN REPUBLIC	CF	Europe
2B	43	CHAD	TD	Europe
2C	44	CHILE	CL	USA
2D	45	CHINA	CN	Japan
2E	46	CHRISTMAS ISLAND	CX	Japan
2F	47	COCOS (KEELING) ISLANDS	CC	Japan
30	48	COLOMBIA	CO	USA
31	49	COMOROS	KM	Europe
32	50	CONGO	CG	Europe
33	51	CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD	Japan
34	52	COOK ISLANDS	CK	Japan
35	53	COSTA RICA	CR	USA
36	54	CÔTE D'IVOIRE	CI	Japan
37	55	CROATIA	HR	Europe
38	56	CUBA	CU	USA
39	57	CYPRUS	CY	Europe
3A	58	CZECH REPUBLIC	CZ	Europe
3B	59	DENMARK	DK	Europe
3C	60	DJIBOUTI	DJ	Japan
3D	61	DOMINICA	DM	USA

Table 8 Country Codes (cont'd)

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
3E	62	DOMINICAN REPUBLIC	DO	USA
3F	63	ECUADOR	EC	USA
40	64	EGYPT	EG	Europe
43	67	EL SALVADOR	SV	USA
44	68	EQUATORIAL GUINEA	GQ	Europe
45	69	ERITREA	ER	Europe
46	70	ESTONIA	EE	Europe
47	71	ETHIOPIA	ET	Europe
48	72	FALKLAND ISLANDS (MALVINAS)	FK	USA
49	73	FAROE ISLANDS	FO	Europe
4A	74	FIJI	FJ	Japan
4B	75	FINLAND	FI	Europe
4C	76	FRANCE	FR	Europe
4D	77	FRENCH GUIANA	GF	Europe
4E	78	FRENCH POLYNESIA	PF	Europe
4F	79	FRENCH SOUTHERN TERRITORIES	TF	Japan
50	80	GABON	GA	Europe
51	81	GAMBIA	GM	Europe
52	82	GEORGIA	GE	Europe
53	83	GERMANY	DE	Europe
54	84	GHANA	GH	Japan
55	85	GIBRALTAR	GI	Europe
56	86	GREECE	GR	Europe
57	87	GREENLAND	GL	USA
58	88	GRENADA	GD	USA
59	89	GUADELOUPE	GP	Europe
5A	90	GUAM	GU	Japan
5B	91	GUATEMALA	GT	USA
5C	92	GUERNSEY	GG	Europe
5D	93	GUINEA	GN	Europe
5E	94	GUINEA-BISSAU	GW	Europe
5F	95	GUYANA	GY	USA

Table 8 Country Codes (cont'd)

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
60	96	HAITI	HT	USA
61	97	HEARD ISLAND AND MCDONALD ISLANDS	HM	Japan
62	98	HOLY SEE (VATICAN CITY STATE)	VA	Europe
63	99	HONDURAS	HN	USA
64	100	HONG KONG	HK	Japan
65	101	HUNGARY	HU	Europe
66	102	ICELAND	IS	Europe
67	103	INDIA	IN	Japan
68	104	INDONESIA	ID	Japan
69	105	IRAN, ISLAMIC REPUBLIC OF	IR	Europe
6A	106	IRAQ	IQ	Europe
6B	107	IRELAND	IE	Europe
6C	108	ISLE OF MAN	IM	Europe
6D	109	ISRAEL	IL	Europe
6E	110	ITALY	IT	Europe
6F	111	JAMAICA	JM	USA
70	112	JAPAN	JP	Japan
71	113	JERSEY	JE	Europe
72	114	JORDAN	JO	Europe
73	115	KAZAKHSTAN	KZ	Europe
74	116	KENYA	KE	Europe
75	117	KIRIBATI	KI	Europe
76	118	KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP	Japan
77	119	KOREA, REPUBLIC OF	KR	Japan
78	120	KUWAIT	KW	Europe
79	121	KYRGYZSTAN	KG	Europe
7A	122	LAO PEOPLE'S DEMOCRATIC REPUBLIC	LA	Japan
7B	123	LATVIA	LV	Europe
7C	124	LEBANON	LB	Europe

Table 8 Country Codes (cont'd)

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
7D	125	LESOTHO	LS	Europe
7E	126	LIBERIA	LR	Europe
7F	127	LIBYAN ARAB JAMAHIRIYA	LY	Europe
80	128	LIECHTENSTEIN	LI	Europe
81	129	LITHUANIA	LT	Europe
82	130	LUXEMBOURG	LU	Europe
83	131	MACAO	MO	Europe
84	132	MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	MK	Europe
85	133	MADAGASCAR	MG	Europe
86	134	MALAWI	MW	Europe
87	135	MALAYSIA	MY	Japan
88	136	MALDIVES	MV	Japan
89	137	MALI	ML	Europe
8A	138	MALTA	MT	Europe
8B	139	MARSHALL ISLANDS	MH	Japan
8C	140	MARTINIQUE	MQ	Europe
8D	141	MAURITANIA	MR	Europe
8E	142	MAURITIUS	MU	Europe
8F	143	MAYOTTE	YT	Europe
90	144	MEXICO	MX	USA
91	145	MICRONESIA, FEDERATED STATES OF	FM	Japan
92	146	MOLDOVA, REPUBLIC OF	MD	Europe
93	147	MONACO	MC	Europe
94	148	MONGOLIA	MN	Europe
95	149	MONTENEGRO	ME	Europe
96	150	MONTSERRAT	MS	Europe
97	151	MOROCCO	MA	Europe
98	152	MOZAMBIQUE	MZ	Europe
99	153	MYANMAR	MM	Europe
9A	154	NAMIBIA	NA	Europe
9B	155	NAURU	NR	Japan

Table 8 Country Codes (cont'd)

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
9C	156	NEPAL	NP	Japan
9D	157	NETHERLANDS	NL	Europe
9E	158	NETHERLANDS ANTILLES	AN	USA
9F	159	NEW CALEDONIA	NC	Japan
A0	160	NEW ZEALAND	NZ	Japan
A1	161	NICARAGUA	NI	USA
A2	162	NIGER	NE	Europe
A3	163	NIGERIA	NG	Europe
A4	164	NIUE	NU	Japan
A5	165	NORFOLK ISLAND	NF	Japan
A6	166	NORTHERN MARIANA ISLANDS	MP	Japan
A7	167	NORWAY	NO	Europe
A8	168	OMAN	OM	Europe
A9	169	PAKISTAN	PK	Europe
AA	170	PALAU	PW	Japan
AB	171	PANAMA	PA	USA
AC	172	PAPUA NEW GUINEA	PG	Europe
AD	173	PARAGUAY	PY	USA
AE	174	PERU	PE	USA
AF	175	PHILIPPINES	PH	Japan
B0	176	PITCAIRN	PN	Japan
B1	177	POLAND	PL	Europe
B2	178	PORTUGAL	PT	Europe
B3	179	PUERTO RICO	PR	USA
B4	180	QATAR	QA	Europe
B5	181	RÉUNION	RE	Europe
B6	182	ROMANIA	RO	Europe
B7	183	RUSSIAN FEDERATION	RU	Europe
B8	184	RWANDA	RW	Europe
B9	185	SAINT BARTHÉLEMY	BL	USA
BA	186	SAINT HELENA	SH	Europe
BB	187	SAINT KITTS AND NEVIS	KN	USA
BC	188	SAINT LUCIA	LC	USA

Table 8 Country Codes (cont'd)

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
BD	189	SAINT MARTIN	MF	USA
BE	190	SAINT PIERRE AND MIQUELON	PM	USA
BF	191	SAINT VINCENT AND THE GRENADINES	VC	USA
C0	192	SAMOA	WS	Japan
C1	193	SAN MARINO	SM	Europe
C2	194	SAO TOME AND PRINCIPE	ST	Europe
C3	195	SAUDI ARABIA	SA	Europe
C4	196	SENEGAL	SN	Europe
C5	197	SERBIA	RS	Europe
C6	198	SEYCHELLES	SC	Japan
C7	199	SIERRA LEONE	SL	Europe
C8	200	SINGAPORE	SG	Japan
C9	201	SLOVAKIA	SK	Europe
CA	202	SLOVENIA	SI	Europe
CB	203	SOLOMON ISLANDS	SB	Japan
CC	204	SOMALIA	SO	Europe
CD	205	SOUTH AFRICA	ZA	Europe
CE	206	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS	Europe
CF	207	SPAIN	ES	Europe
D0	208	SRI LANKA	LK	Japan
D1	209	SUDAN	SD	Europe
D2	210	SURINAME	SR	USA
D3	211	SVALBARD AND JAN MAYEN	SJ	Europe
D4	212	SWAZILAND	SZ	Europe
D5	213	SWEDEN	SE	Europe
D6	214	SWITZERLAND	CH	Europe
D7	215	SYRIAN ARAB REPUBLIC	SY	Europe
D8	216	TAIWAN, PROVINCE OF CHINA	TW	Japan

Table 8 Country Codes (cont'd)

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
D9	217	TAJIKISTAN	TJ	Europe
DA	218	TANZANIA, UNITED REPUBLIC OF	TZ	Europe
DB	219	THAILAND	TH	Japan
DC	220	TIMOR-LESTE	TL	Japan
DD	221	TOGO	TG	Europe
DE	222	TOKELAU	TK	Japan
DF	223	TONGA	TO	Europe
E0	224	TRINIDAD AND TOBAGO	TT	Europe
E1	225	TUNISIA	TN	Europe
E2	226	TURKEY	TR	Europe
E3	227	TURKMENISTAN	TM	Europe
E4	228	TURKS AND CAICOS ISLANDS	TC	USA
E5	229	TUVALU	TV	Japan
E6	230	UGANDA	UG	Europe
E7	231	UKRAINE	UA	Europe
E8	232	UNITED ARAB EMIRATES	AE	Europe
E9	233	UNITED KINGDOM	GB	Europe
EA	234	UNITED STATES	US	USA
EB	235	UNITED STATES MINOR OUTLYING ISLANDS	UM	USA
EC	236	URUGUAY	UY	USA
ED	237	UZBEKISTAN	UZ	Europe
EE	238	VANUATU	VU	Japan
F0	240	VENEZUELA	VE	USA
F1	241	VIET NAM	VN	Japan
F2	242	VIRGIN ISLANDS, BRITISH	VG	USA
F3	243	VIRGIN ISLANDS, U.S.	VI	USA
F4	244	WALLIS AND FUTUNA	WF	Europe
F5	245	WESTERN SAHARA	EH	Europe
F6	246	YEMEN	YE	Europe

Table 8 Country Codes (cont'd)

EEPROM Value (Hex)	EEPROM Value (Decimal)	Country Name	Country Code	Regulatory Domain
F7	247	ZAMBIA	ZM	Europe
F8	248	ZIMBABWE	ZW	Europe

10 Appendix D - Command Line Control

During the DUT GUI installation, the DUT_BCL command line application is also installed.

The application enables 3 options for control of the Platform (all information can be seen with `-help` command):

- Opening a dedicated command window application
  Syntex: `DUT_BCL.exe -i <IP> -w<WLAN interface 0/1>`
- Providing direct commands (one command at a time)
  Syntax: `DUT_BCL.exe -i<IP> -w<Wlan interface 0/1> -s -c <DUT Command + correct syntax>`
- Running a script file with various command
  Syntax: `DUT_BCL.exe -i<IP> -w<Wlan interface 0/1> -s -f <file name>`

1. Run the tool from C:\Program Files\Lantiq\DUT\DUT_BCL.exe.
   The cmd window appears (**Figure 24**).

Figure 24 BCL CLI

2. Type help at the command prompt to see the complete list of available functions.
3. Type the command name to see the correct syntax for that command.
   For example:
   `> DUT_SET_CHANNEL`
   `Incorrect number of parameters.`
   See **Figure 25** for a successful command

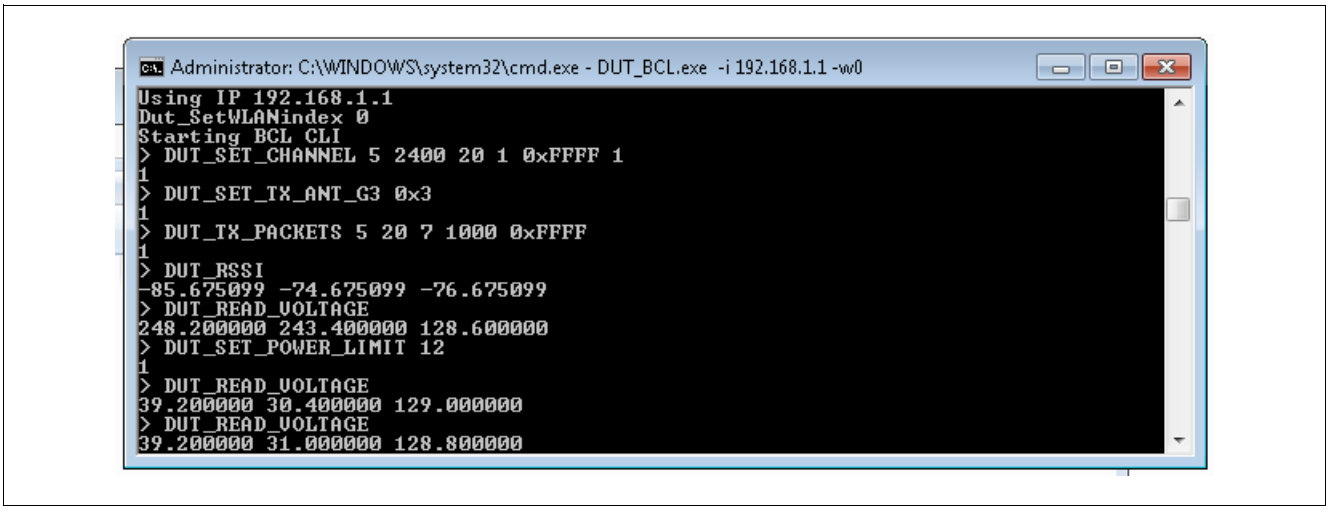


Figure 25 Successful CLI Inputs

10.1 Basic Configurations

Mode

- `0` for 802.11a
- `1` for 802.11b
- `2` for 802.11g
- `4` for 802.11n 5 GHz
- `5` for 802.11n 2.4 GHz

Band

`2400` or `5000`

BW

`20` or `40`

Channel

In accordance to channel index

calibrationMask

Always use `0xFFFF`

Closedloop

- `0` of open loop
- `1` for closed loop

Antenna Configuration

3 bits:

- Bit 0 for Ant1
- Bit 1 for Ant2
- Bit 2 for Ant3

References

[1] IEEE 802.11a-1999 High-speed Physical Layer in the 5 GHz band
http://standards.ieee.org/getieee802/download/802.11a-1999.pdf

[2] IEEE 802.11b-1999 Higher Speed Physical Layer Extension in the 2.4 GHz band
http://standards.ieee.org/getieee802/download/802.11b-1999.pdf

[3] IEEE 802.11g-2003 Further Higher Data Rate Extension in the 2.4 GHz Band
http://standards.ieee.org/getieee802/download/802.11g-2003.pdf

[4] IEEE 802.11n-2009—Amendment 5: Enhancements for Higher Throughput
http://standards.ieee.org/getieee802/download/802.11n-2009.pdf

[5] XWAY™ xRX300/WAVE300 DUT Tool Package Release 6.30 Release Note Rev. 2.4

Terminology

A	
ACL	Access Control List
AP	Access Point.
B	
BBIC	Baseband Integrated Circuit
BPF	Band Pass Filter
BSS	The Basic Service Set (BSS) is the basic building block of a wireless LAN. Coverage of one access point is called a BSS. An access point acts as a master to control the stations within that BSS. Each BSS is identified by an SSID.
C	
CB	Channel Bonding (40 MHz transmission)
CLI	Command Line Interface
D	
DUT	Device Under Test
E	
ERP	Extended Rate Policy
ESSID	Extended Service Set Identifier
I	
IOCTLS	Input/output controls - typically employed to allow userspace code to communicate with hardware devices
L	
LDPC	Low-Density Parity-Check code (LDPC code) - An error correcting code. A method of transmitting a message over a noisy transmission channel.
M	
MIMO	Multiple-Input and Multiple-Output - The use of multiple antennas at both the transmitter and receiver to improve communication performance. One of several forms of smart antenna technology.
mPCI	Mini PCI - A bus standard for attaching peripherals to a motherboard. Adapted from the Peripheral Component Interconnect (PCI) bus. Originally designed for laptops and other small-footprint computer systems.
Multicast	The delivery of information to a group of destinations simultaneously using the most efficient strategy to deliver the messages over each link of the network only once, creating copies only when the links to the destinations split.
N	
NCB	Non Channel bonding (20 MHz transmission)
nCB	Non-Channel Bonding
NFS	Linux's "Network File System" - A way to share files between machines on a network as if the files were located on the client's local hard drive.
NWID	Network Identification Designator.
O	

OCS	Optimal Channel Selection
P	
PBC	Push Button Configuration
R	
RFIC	RF Integrated Circuit
RSSI	Received Signal Strength Indication - a measurement of the power present in a received radio signal.
S	
SISO	Single Input Single Output
STA	Infrastructure Station
U	
UUID	Universally Unique Identifier
W	
WEP	Wired Equivalent Privacy - The original security protocol for Wi-Fi networks.
WPA	WiFi Protected Access - A security protocol for Wi-Fi networks which provides stronger security than WEP via enhanced encryption and user authentication.
WPS	WiFi Protected Setup - A protocol designed to make it easier to set up and configure security on Wi-Fi networks.

CE

CE RF Radiation Exposure Statement:
Caution This equipment complies with European RF radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with a minimum distance of 20 centimeters between the radiator and your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

FCC Information

This device complies with part 15 of the FCC Results. Operation is subject to the following two conditions :

(1) This device may not cause harmful interface, and

(2) This device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for CLASS B digital device, pursuant to Part 15 of FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try correct the interference by one or more of the following measures:

1.1. Reorient or relocate the receiving antenna.
1.2. Increase the separation between the equipment and receiver.
1.3. Connect the equipment into an outlet on a circuit different from that to which receiver is connected.
1.4. Consult the dealer or experienced radio/TV technician for help.

WARNING

Changes or modifications not expressly approved by the manufacturer could void the user's authority to operate the equipment.

Information for OEM Integrator

This device is intended only for OEM integrators under the following conditions:

1) The antenna must be installed such that 20 cm is maintained between the antenna and users, and

2) The transmitter module may not be co-located with any other transmitter or antenna.

End product labelling

The label for end product must include "Contains FCC ID: O6ZT9".

"CAUTION : Exposure to Radio Frequency Radiation.

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment must be installed and operated with minimum distance of 20cm between the radiator and your body. This transmitter module is authorized only for use in device where the antenna may be installed such that 20 cm may be maintained between the antenna and users."

www.lantiq.com

Published by Lantiq Deutschland GmbH