User Manual

SmartBond Production Line Tool

UM-B-041

Abstract

This document describes the SmartBond Production Line Tool (PLT) for DA1470x wireless SoC family of products. The various software applications, as well as the PLT hardware are explained in detail. The purpose of this document is to guide users on how to use the various PLT components.

Contents

Figures

Tables

1 Terms and Definitions

API	Application Programming Interface
BD	Bluetooth Device
.bin	Firmware files in binary format
Bluetooth® LE	Bluetooth® Low Energy
CFG	Configuration
CLI	Command Line Interface
COM	Communication Port
CPLD	Complex Programmable Logic Device
CRC	Cyclic Redundancy Check
CS	Configuration Script
CSV	Comma Separated Values
DLL	Dynamic Link Library
DMA	Direct Memory Access
DMM	Digital Multi-Meter
DTM	Direct Test Mode (as specified by the Bluetooth® LE Core standard)
DUT	Device Under Test
DVM	Digital Voltage Meter
EEPROM	Electrically Erasable Programmable Read-Only Memory
.exe	Executable file
FTDI	Future Technology Devices International Ltd.
GPIO	General Purpose Input-Output
GU	Golden Unit
GUI	Graphical User Interface
Hex	Firmware file in ASCII format
HW	Hardware
IC	Integrated Circuit
IDE	Integrated Development Environment
I2C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
OQSPI	Octal/Quad SPI Flash
OS	Operating System
OTP	One-Time Programmable (memory)
PC	Personal Computer
PCB	Printed Circuit Board
PER	Packet Error Rate
PLT	Production Line Tool
PLTD	Production Line Tool DLL
POR	Power-On Reset
RAM	Random-Access Memory
RCX	Resistor Crystal Oscillator
RF	Radio Frequency
RX	Receive
SCPI	Standard Commands for Programmable Instruments
SoC	System-on-Chip

SDK	Software Development Kit
SPI	Serial Peripheral Interface
SW	Software
TCS	Trim and Calibration Section
TX	Transmit
UART	Universal Asynchronous Receiver/Transmitter
UI	User Interface
USB	Universal Serial Bus
VISA	Virtual Instrument Software Architecture
VPP	Programming Supply Voltage (pin)
XML	Extensible Markup Language
XTAL	Crystal
XSD	XML Schema Definition

2 References

[1] UM-B-040, DA1458x/DA1468x Production Line Tool Libraries v4.2, User manual, Renesas

[2] UM-B-014, DA1458x Bluetooth Smart Development Kit - Expert, User Manual, Renesas

[3] FT4232H – Hi-Speed Quad USB UART IC, FTDI Chip

[4] FT232 – USB UART IC, FTDI Chip

[5] Anritsu MT8852B, https://www.anritsu.com/en-US/test-measurement/products/mt8852b

[6] Keysight 34401A, http://www.keysight.com/en/pd-1000001295%3Aepsg%3Apro-pn-34401A/digital-multimeter-6-digit?cc=US&lc=eng

[7] Keithley 2000, http://www.tek.com/tektronix-and-keithley-digital-multimeter/keithley-2000-series-6%C2%BD-digit-multimeter-scanning

[8] Papouch TMU USB thermometer, https://www.papouch.com/en/shop/product/tmu-usb-thermometer/

[9] NI USB TC-01, http://sine.ni.com/nips/cds/view/p/lang/en/nid/208177

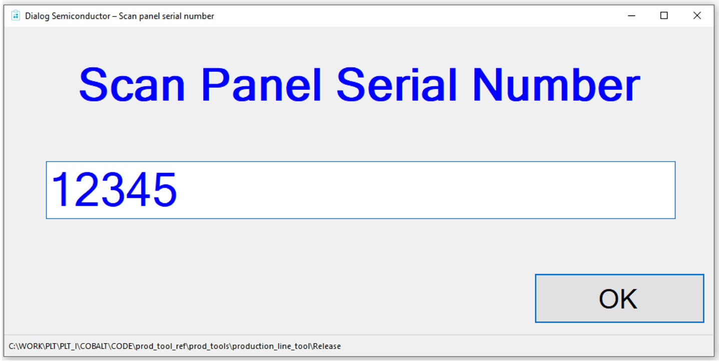
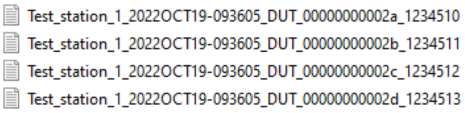
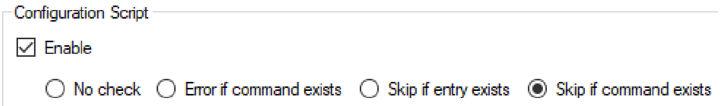
[10] LitePoint IQxel-M, http://www.litepoint.com/test-solutions-for-manufacturing/iqxel-m/

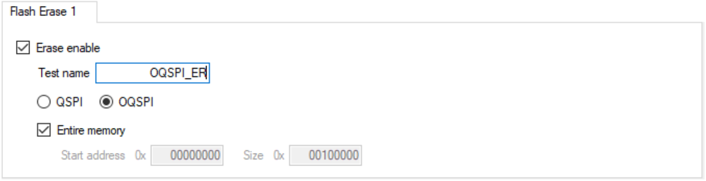
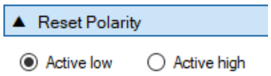
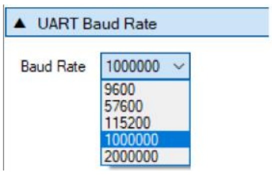
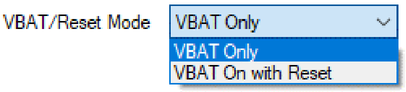
[11] Keysight 34461A, http://www.keysight.com/en/pd-2270273-pn-34461A/digital-multimeter-6-digit-34401a-replacement-truevolt-dmm?cc=GR&lc=eng

3 New Version Features

This manual explains the usage of the 16 channels SmartBond^{TM} Production Line Tool (PLT). It refers to the SmartBond_PLT_v5.0 software release, which compared to SmartBond_PLT_v_4.5 has the differences shown in Table 1.

Table 1: SmartBond_PLT_v5.0 Added Features

#	Features	Description
1	DA1470x family support	Supports only DA1470x family. The older SmartBond family of products are supported in previous PLT versions.
2	PCB panel serial number	Before each test a screen prompts user to enter the panel serial number. PLT will add number 1-16 to the end of the serial number and append it to each DUT log as shown below for DUTs 10, 11, 12 and 13.  Dialog Semiconductor – Scan panel serial number Scan Panel Serial Number 12345 C:\WORK\PLT\PLT_I\COBALT\CODE\prod_tool_ref\prod_tools\production_line_tool\Release  Test_station_1_2022OCT19-093605_DUT_00000000002a_1234510 Test_station_1_2022OCT19-093605_DUT_00000000002b_1234511 Test_station_1_2022OCT19-093605_DUT_00000000002c_1234512 Test_station_1_2022OCT19-093605_DUT_00000000002d_1234513
3	OTP configuration script	The OTP CS programming parameters are specific to the DA1470x product family.
4	OTP configuration script check empty	Updated operation of OTP CS check empty algorithm.  Configuration Script ☑ Enable ○ No check ○ Error if command exists ○ Skip if entry exists ● Skip if command exists ● **No check:** Check disabled. ● **Error if command exists:** Returns error if the command is already written in the DUT, even if the data are the same. ● **Skip if entry exist:** Skip writing an entry without error if the command and the data are already written in the DUT. If same command is found with different data an error will be returned. ● **Skip if command exists:** Skip writing without returning error if the command in the DUT OTP CS is already written, no matter what the data are (same or different).
5	XTAL 32 MHz settle time calibration.	XTAL 32 MHz. calibration is extended to support settle time calibration, with values found programed in OTP CS and applied in XTAL32M_TRIM_REG (0x50050408) DA1470x register. Operation adds less than 1sec extra delay but has great benefits in overall product sleep and thus power performance.

#	Features	Description
6	Added support for Octal SPI flash interface.	DA1470x supports Octal SPI flash. User has the option to choose which interface to use for flash erase, programming or read. An example of flash erase choosing the OQSPI interface is shown next.  Flash Erase 1 ☑ Erase enable Test name OQSPI_ER ○ QSPI ● OQSPI ☑ Entire memory Start address 0x 00000000 Size 0x 00100000
6	Reset polarity selection	DUT reset polarity is now configurable.  ▲ Reset Polarity ● Active low ○ Active high
7	2Mbaud UART baud rate	Support of 2Mbaud UART baud rate.  ▲ UART Baud Rate Baud Rate 1000000 9600 57600 115200 1000000 2000000
8	Removed barcode scanner support	The feature of scanning BD addresses using a barcode scanner was removed.
9	Removed feature **VBAT as Reset**	Power cycle and DUT reset can only be done using **VBAT Only** and **VBAT On with Reset**.  VBAT/Reset Mode VBAT Only VBAT Only VBAT On with Reset
10	Removed support for using DA1468x DK as current measurement instrument.	Using DA1468x DK as current measurement instrument was removed because it requires extra calibration steps that makes it difficult to safely be used in production.
11	Low level debug log files flush	Low level debug logs are flushed continuously. No need to close the application executable anymore for the log files to be updated.

4 Introduction

By using the PLT, it is possible to test, calibrate and load firmware for 16 different devices under test (DUTs) in parallel.

The following are a list of parts delivered with the tool:

- Hardware
  - Main board (Figure 1) together with a DA14580-QFN48 Golden Unit (GU)
  - Electrical schematics
  - Gerber files
  - Bill of Materials
- Software
  - Source code files organized in a Microsoft® Visual Studio Express 2017 solution
  - Application executables and required DLLs
- Documents

An example of a sequence of actions the tool performs is given below. All actions are performed in parallel for up to 16 devices.

1. Download the production test firmware.
2. Perform automatic crystal (XTAL) trimming.
3. Perform RF RSSI test.
4. Download and burn the customer firmware (into OTP, QSPI flash).
5. Burn the OTP header.
6. Perform Scan test. Reset the DUTs and set the GU to scan for the DUT Bluetooth® LE advertisements.

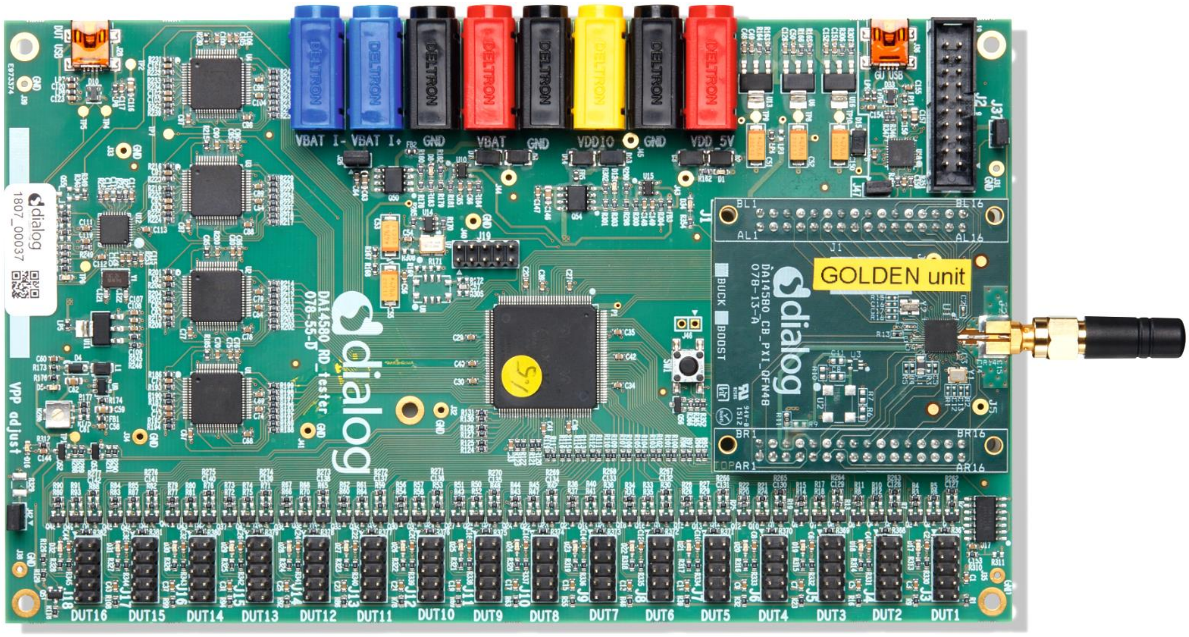


Figure 1: Production Line Tool Hardware

5 Hardware

5.1 Hardware Block Diagram

The Production Line Tool hardware consists of various blocks, as illustrated in Figure 2. These blocks are explained below.

- **Blue blocks:** USB-to-UART interfaces
  - Four **FT4232 FTDI** QUAD USB-to-UART interfaces are used for a 16-channel USB-to-UART conversion
  - The GU is connected to the PC via an **FT232 FTDI** USB-to-UART interface
- **Red block:** A **CPLD** that has the following purpose
  - Switch UART signals between the PC USB-UART and DUTs
  - Switch DUTs VBAT signal
  - Switch DUTs VPP signal (only when VBAT is enabled)
  - Produce Reset signal to the DUTs
  - Produce very accurate 300 ms or 500 ms XTAL calibration pulse
- **Orange block:** A **Golden Unit** (GU) is mounted, which has the following functionality:
  - CPLD control using custom commands
  - Transceiver for Bluetooth RF signals to and from the DUTs
  - Produce an audio tone using PWM, used for audio testing
  - Scan for device Bluetooth® LE advertisements, after the customer firmware has been programmed
- **Purple blocks:** Sixteen (16) device connectors

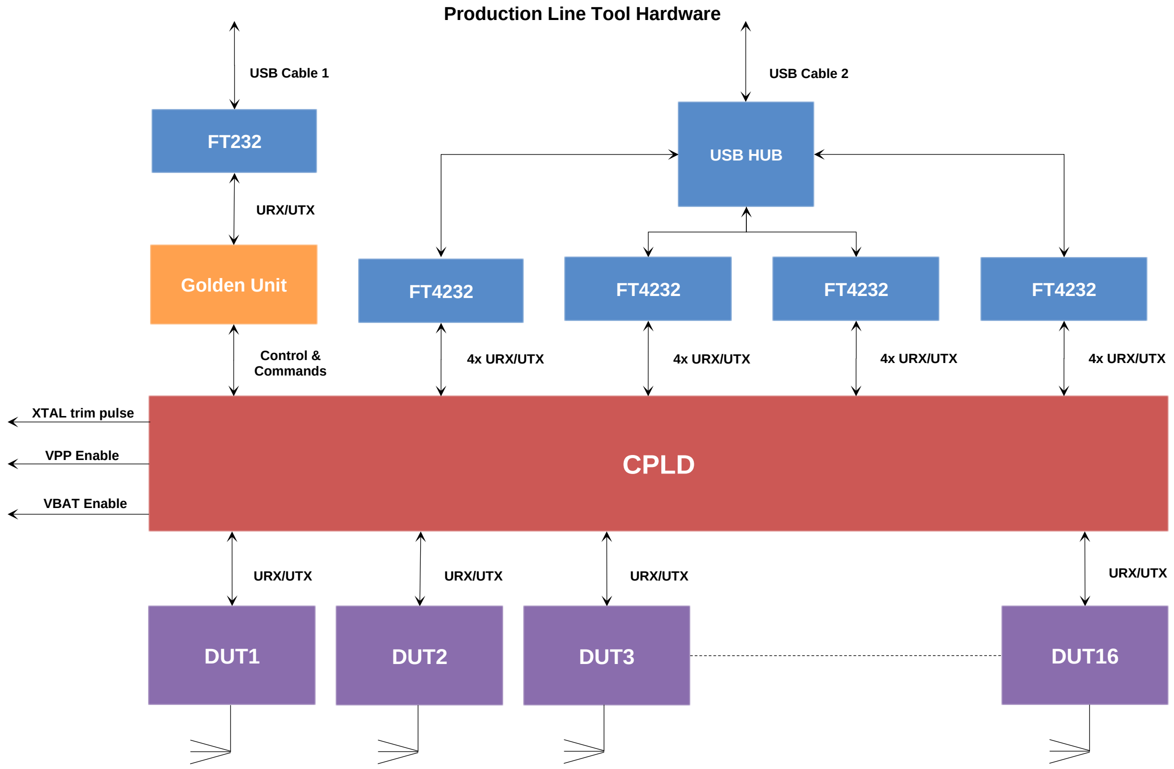


Figure 2: Production Line Tool Hardware Board Block Diagram

5.2 Printed Circuit Board Layout

In Figure 3, the top view of the PLT board is illustrated. The important parts are pointed by the orange boxes. The *VPP jumper* and the *Current jumper* are colored in blue.

The Golden Unit has a DA14580 QFN48-die soldered. Most of the 48 pins are used to connect to the CPLD. The CPLD is programmed during the production of the PLT board via the CPLD socket. No need for the users to use the CPLD socket.

The black banana sockets are all connected to the same ground (GND) plane.

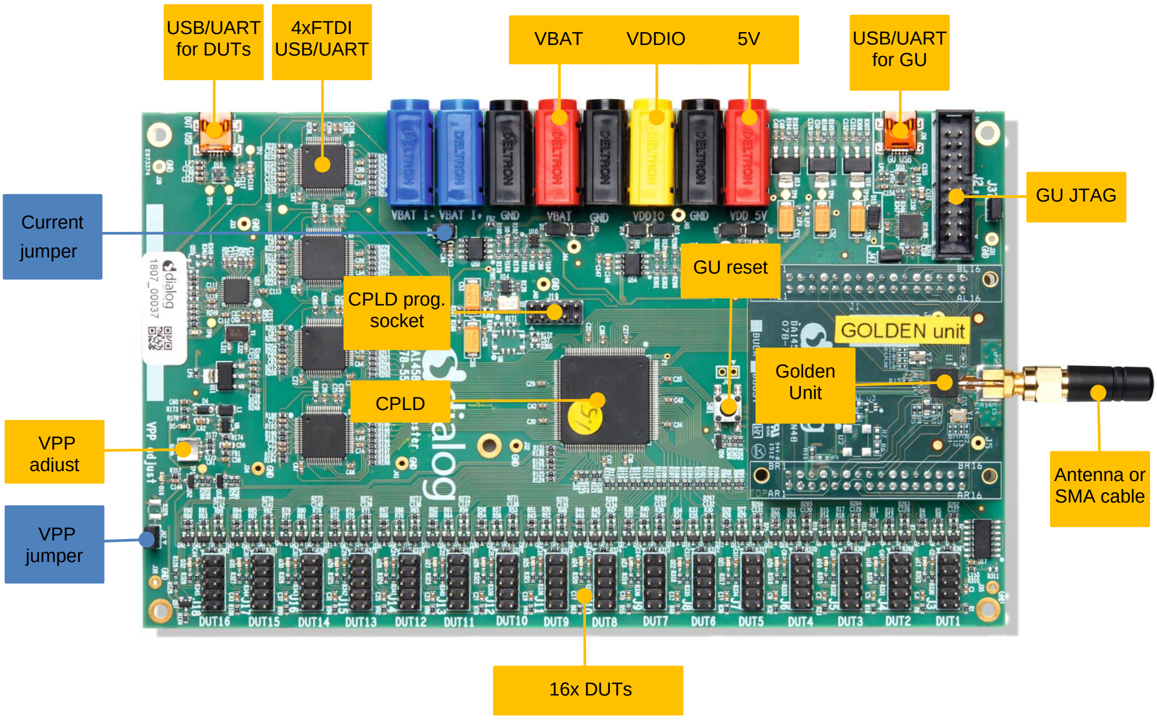


Figure 3: Top View of the PLT Hardware Board

5.3 PLT Power Supply

External power supply is needed for the PLT to run. This should be connected to the banana sockets shown in Figure 4.

Table 2 shows the voltage and current requirements for each power supply. The blue banana sockets can be used for device current measurements.

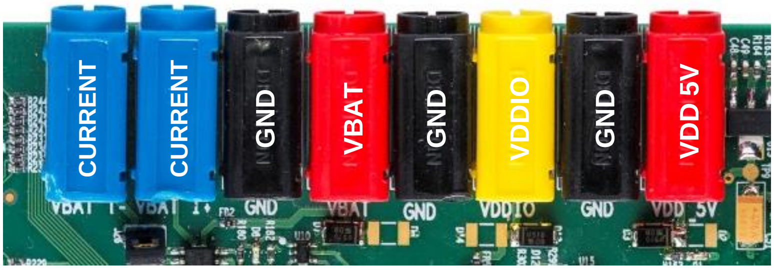


Figure 4: PLT Hardware Power Connections

Table 2: Power Supply Requirements

Power Supply	Voltage (V)	Current (mA)	
		Buck Mode	Boost Mode
VBAT (Buck mode)	2.4 … 3.3	16 x 20	
VBAT (Boost mode)	1.5 … 3.3		16 x 20
VDDIO	2.4 … 3.3	70	70
VDD 5V	4.75 … 5.25	~335	~335
VPP	6.6 … 6.8	16 x 2	16 x 2

5.4 DUT Connector

The Bluetooth® LE devices are connected to the PLT using the DUT1-16 connectors at the edge of the PLT board. Figure 5 shows the pin-header connections from the Production Line Tool hardware board to the DUTs. Table 3 describes the purpose of each pin.

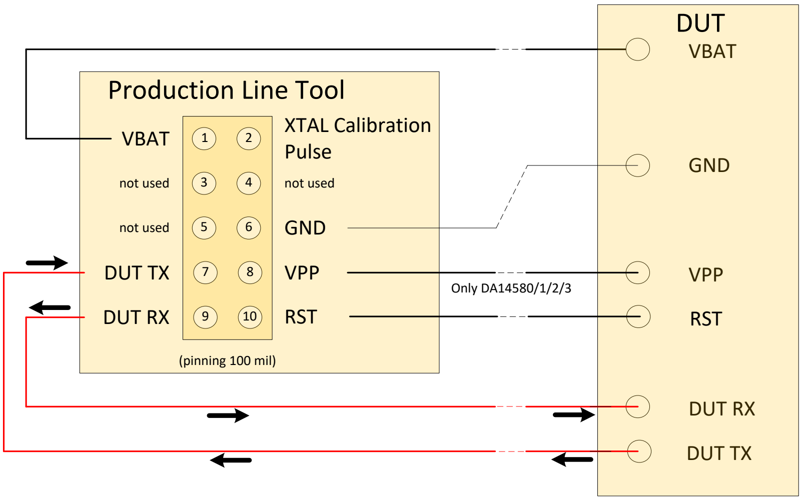


Figure 5: Production Line Tool DUT Connections

Table 3: PLT Connections to Applications

Header Pin	Name	Description
1	VBAT	Depending on the VBAT/Reset Signals Operation mode (Section 6.9)this can be used as Voltage supply for the DUT. Due to this connection, no external power supply is needed for the DUTs. This pin must be connected if there is no other power supply (e.g., battery).
2	XTAL Calibration Pulse	This pin can be used as a reference pulse during the automatic crystal calibration. More details are given in Section 7.2.6.3. The crystal trim pulse can also be supplied in the UART RX device pin. This is the recommended scenario. However, there may be hardware limitations where the UART RX pin cannot be used. In such cases, the particular PLT header pin should be used.
6	GND	Ground pin. This pin must be connected.
7	DUT TX	This is connected to the device UART TX pin. This pin must be connected.
8	VPP	Not used for DA1470x
9	DUT RX	This is connected to the device UART RX pin. This pin can also provide the crystal calibration reference pulse for the automatic crystal (XTAL) trim procedure, as described in Section 7.2.6.3. This pin must be connected.
10	RST	The reset signal must be connected mostly if battery powered devices are used. A power cycle of VBAT will produce a Power on Reset (POR) in which case the RST-wire is not needed. However, using the RST signal is faster, whereas VBAT POR needs time to discharge.

5.5 Data Streaming

Figure 6, Figure 7 and Figure 8 illustrate the three possible data streams through the CPLD. The CPLD switches S1, S2, S3 and S4 are controlled by the software via the Golden Unit.

5.5.1 Normal Operation

- **UART-RxD data is transported via the RED arrows (AA):**
  PC ➔ USB ➔ USB HUB ➔ Quad UART ➔ CPLD signal ‘AA’ ➔ DUT RxD (programmed as RxD).
- **UART-TxD data is transported via the BLUE arrows (BB):**
  PC 🡐 USB 🡐 USB HUB 🡐 Quad UART 🡐 CPLD signal ‘BB’ 🡐 DUT TxD.

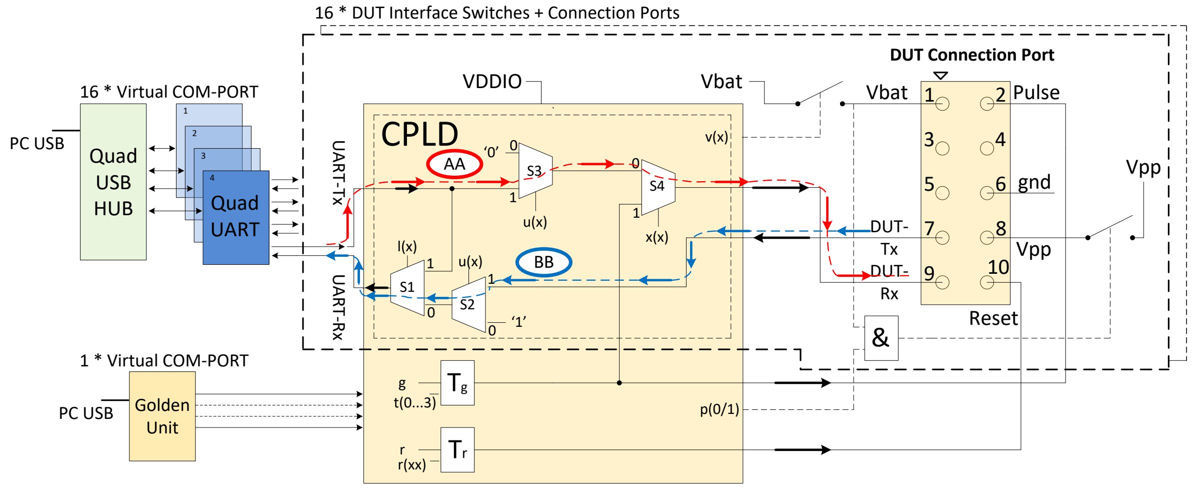


Figure 6: CPLD UART Data Streams

5.5.2 Crystal Trimming

- **The XTAL calibration pulse (300 ms or 500 ms) is transported via the PURPLE arrows (CC):**
  CPLD TIMER Tg ➔ CPLD S4 ➔ DUT RxD (programmed as GPIO).
- **UART-TxD data is transported via the BLUE arrows (BB):**
  PC 🡐 USB 🡐 USB HUB 🡐 Quad UART 🡐 CPLD signal ‘BB’ 🡐 DUT TxD.

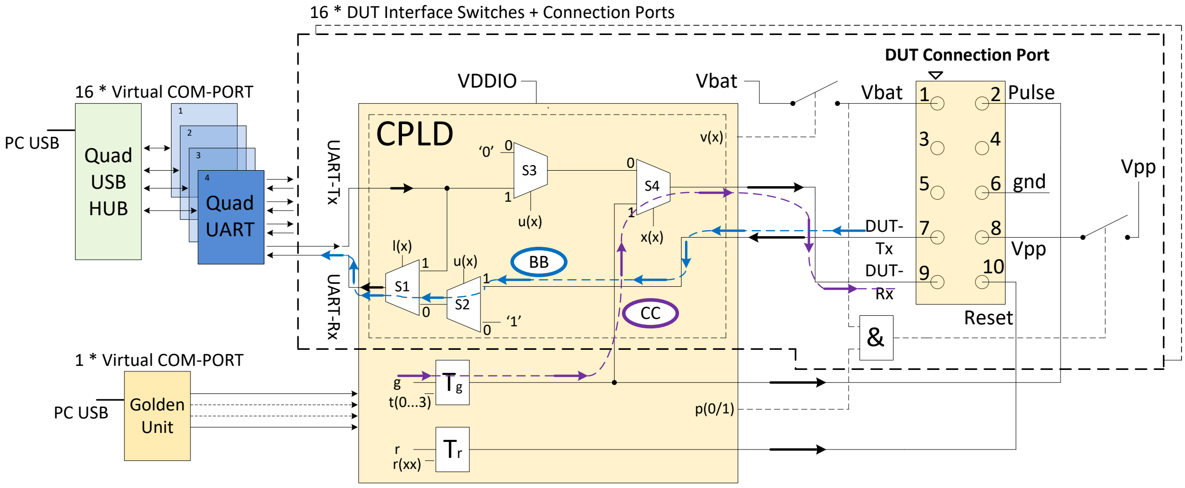


Figure 7: CPLD XTAL Trim Pulse Data Stream

5.5.3 Loopback Operation

Loopback operation is used during the start of the tests. The PLT software uses this feature to automatically find the numbers of the Virtual COM ports in the Windows PC.

- **The UART loopback data is transported via the GREEN arrows (DD):**
  PC ➔ USB ➔ USB HUB ➔ Quad UART ➔ CPLD signal ‘DD’ SW1 ➔ Quad UART ➔ USB HUB ➔ USB ➔ PC.

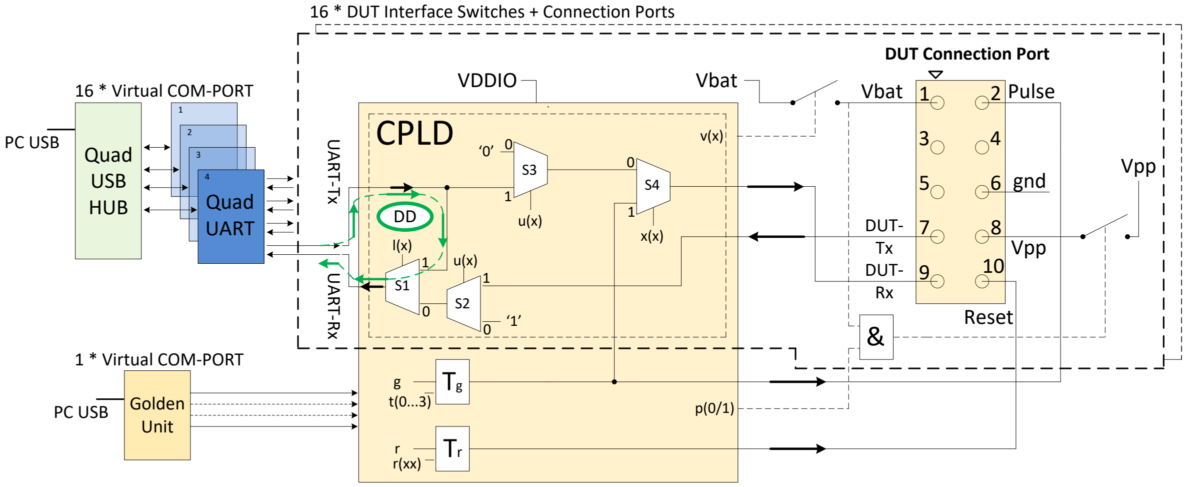


Figure 8: CPLD UART Loopback Data Stream

Note: The CPLD is also used to switch the UART signals between the QUAD FTDIs and the DUTs. When the VBAT is switched off and the UART wires are not disconnected, a ‘rest voltage’ may be present on the product. This could cause problems with the power-on reset (POR) and the product might not boot correctly. The CPLD will switch off the UART signals when the VBAT is not present.

5.6 Golden Unit

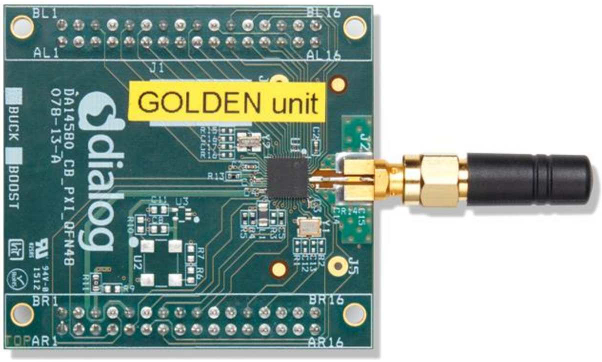


Figure 9: Golden Unit

The Golden Unit (GU) is a 'daughter' board mainly used in the DA1458x Expert Development Kit Ref [2]. In the PLT, the GU is used for various purposes:

- RF transmitter for the RF RSSI DUT test
- RF receiver for the device Bluetooth® LE advertisement scan test
- Audio tone generator for the audio test (not supported in DA1470x family of products)
- Controlling the CPLD

The GU uses an SPI Flash memory mounted on the PLT board. The SPI Flash is pre-programmed with a specific production test firmware. If required, there are several ways to upgrade the GU firmware, either via the PLT's GU JTAG connector, via the UART or using the GUI application executable (`GU_fw_upgrade.exe`) as explained in Section 7.5. The latest GU firmware can be found inside the latest PLT software release, under the `executables\binaries\GU` folder.
Note: PLT v4.3 and onwards requires the latest firmware version of the Golden Unit. If the Golden Unit firmware is not updated, then the PLT applications will not run.

Note: The Golden Unit is calibrated during PLT production. It is delivered with a calibration characterization document.

5.6.1 GU Reset

The Golden Unit includes a hardware reset circuit. The GU reset signal is connected to an FTDI FT232 GPIO pin.

Figure 10 illustrates the electrical schematics of the GU reset circuit. Section 5.8.2 illustrates the jumper positions on the PLT PCB.

The red line is the connection between the FTDI IC GPIO pin (DTR) and the GU reset signal on the PLT GU connector header. The PLT software controls this pin via the FTDI DLL driver `ftd2xx.dll`. Making pin DTR low for a short period of time will reset the GU. Every time the PLT tests start, a hardware reset is issued to the Golden Unit. Jumper J47 should be ON and J46 OFF for this reset method to operate.

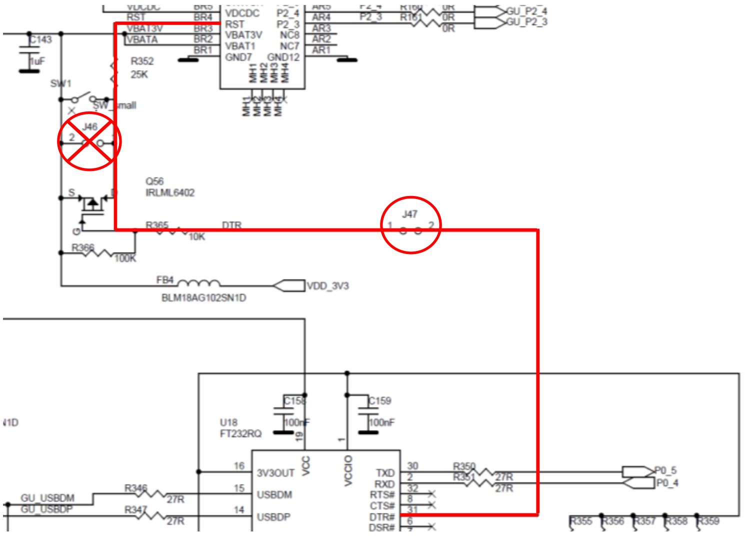


Figure 10: GU Reset Circuit

5.7 Current Measurements

The PLT board provides connections to perform DUT current measurements (Figure 11). By connecting a current meter to the blue banana sockets, the combined VBAT current of all DUTs can be measured. Jumper J26 should be removed when a current meter is connected. If no current meter is used, jumper J26 should be mounted. See also Section 5.8.

The connection shown in Figure 11 can only be used with the VBAT Only (Section 6.9.1) and VBAT On with Reset (Section 6.9.2)(when the VBAT lines are used to power the DUTs) modes. If the DUTs are powered using a single external power supply, then the multi-meter should be connected on that power supply in a similar way as described before with the PLT. If the DUTs are powered independently (e.g., each one with its own battery) the current measurement procedure cannot be used.

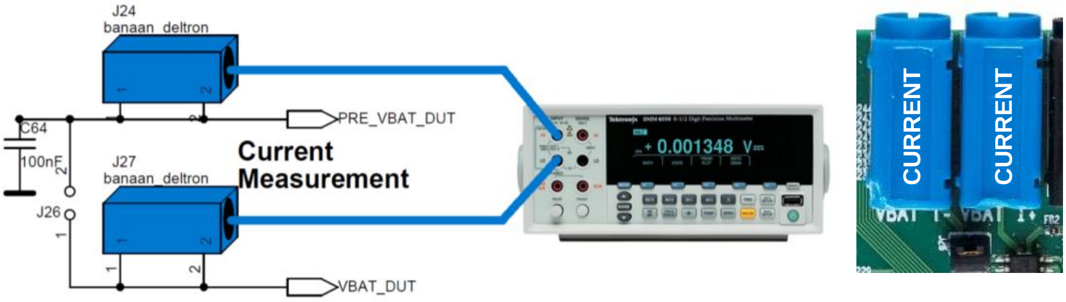


Figure 11: VBAT DUT Current Measurement Setup

5.8 Jumper Settings

This section describes the PLT hardware jumper settings.

Table 4: Jumpers

Jumper	PLT HW version	Description
J26	A, B, C, D	Connects the VBAT line from the PLT power supply to the DUTs. This jumper can be used when there is no multi-meter instrument connected for current measurement.
J37	B, C, D	This jumper sets the Golden Unit's SPI Flash chip select (CS) pin to high. This jumper is placed when the Golden Unit should **NOT** boot from the SPI flash.
J42	B, C, D	Feeds the VPP lines of the DUT connectors with VPP voltage used for OTP burning in DA14580/1/2/3 DUTs.
J46	C, D	This jumper can be used to reset the Golden Unit. The two pins on the jumper are the same as the ones in the GU reset switch next to the jumper.
J47	D	This jumper connects the Golden Unit's FTDI DTR line to the Golden Unit's reset pin. With this jumper on the PLT, software can reset the Golden Unit on-demand.

5.8.1 J26 - Current Measurements

As shown in Figure 12, jumper J26 should be mounted when no external current meter is attached. Otherwise, when a current meter is connected via the blue banana sockets to measure the device current, the J26 jumper should be removed.

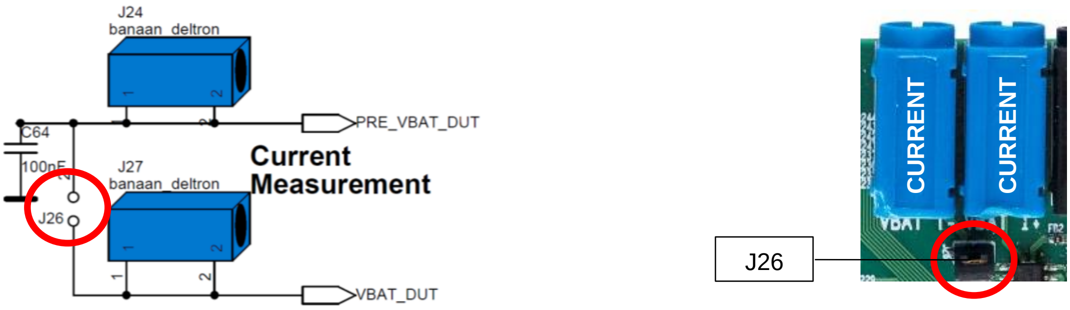


Figure 12: Connections for 'Floating Current' Measurements

5.8.2 J47, J46 - GU Reset

For a GU hardware reset, jumper J47 should be mounted and jumper J46 should be removed. These two jumpers are involved in the circuit illustrated in Figure 10. In this way, the PLT software will control the GU hardware reset. Figure 13 shows the jumper placement on the actual PCB.

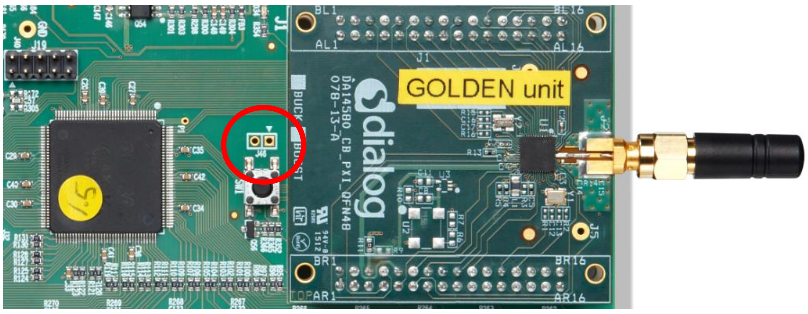


Figure 13: Location of J46 Jumper

2

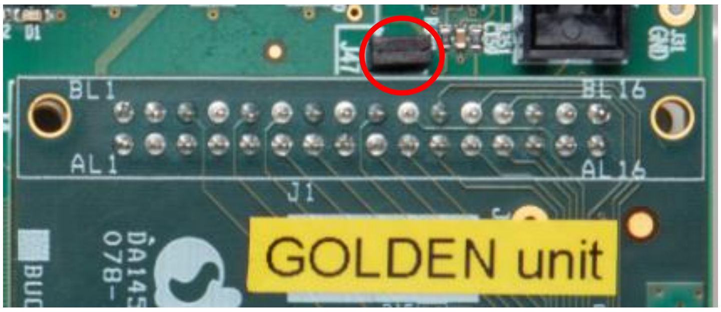


Figure 14: Location of J47 Jumper

5.8.3 J37 - GU Programming

Jumper J37 connects the Chip Select of the GU SPI Flash to a logic high level. This causes the GU not to boot from the already programmed SPI Flash, allowing the GU to load different code into its System-RAM via the JTAG connector or via UART. Figure 15 shows the circuit schematic and Figure 16 shows the location of jumper J37 on the PLT PCB.

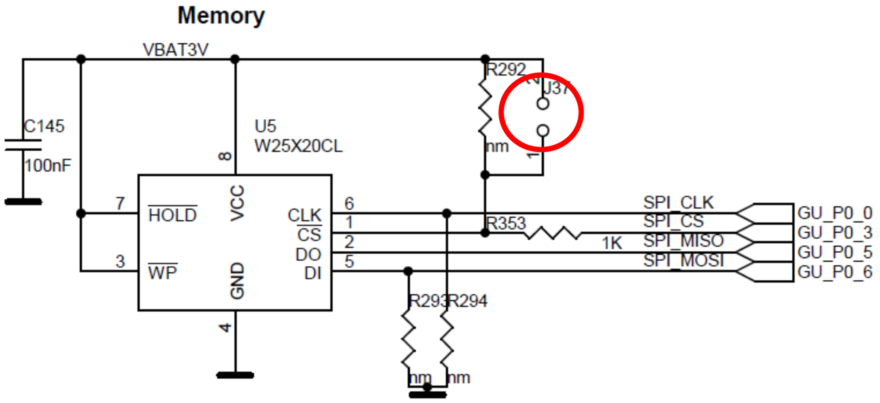


Figure 15: J37 - GU Programming Jumper Schematics

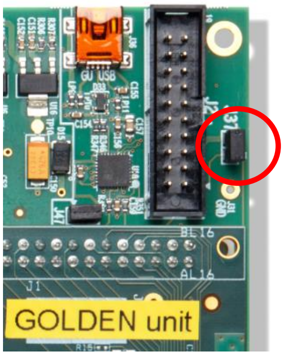


Figure 16: Location of J37 Jumper

5.9 PLT Functional Blocks

Figure 17 shows an overview of the PLT hardware functions. For detailed electrical schematics, see Appendix B.

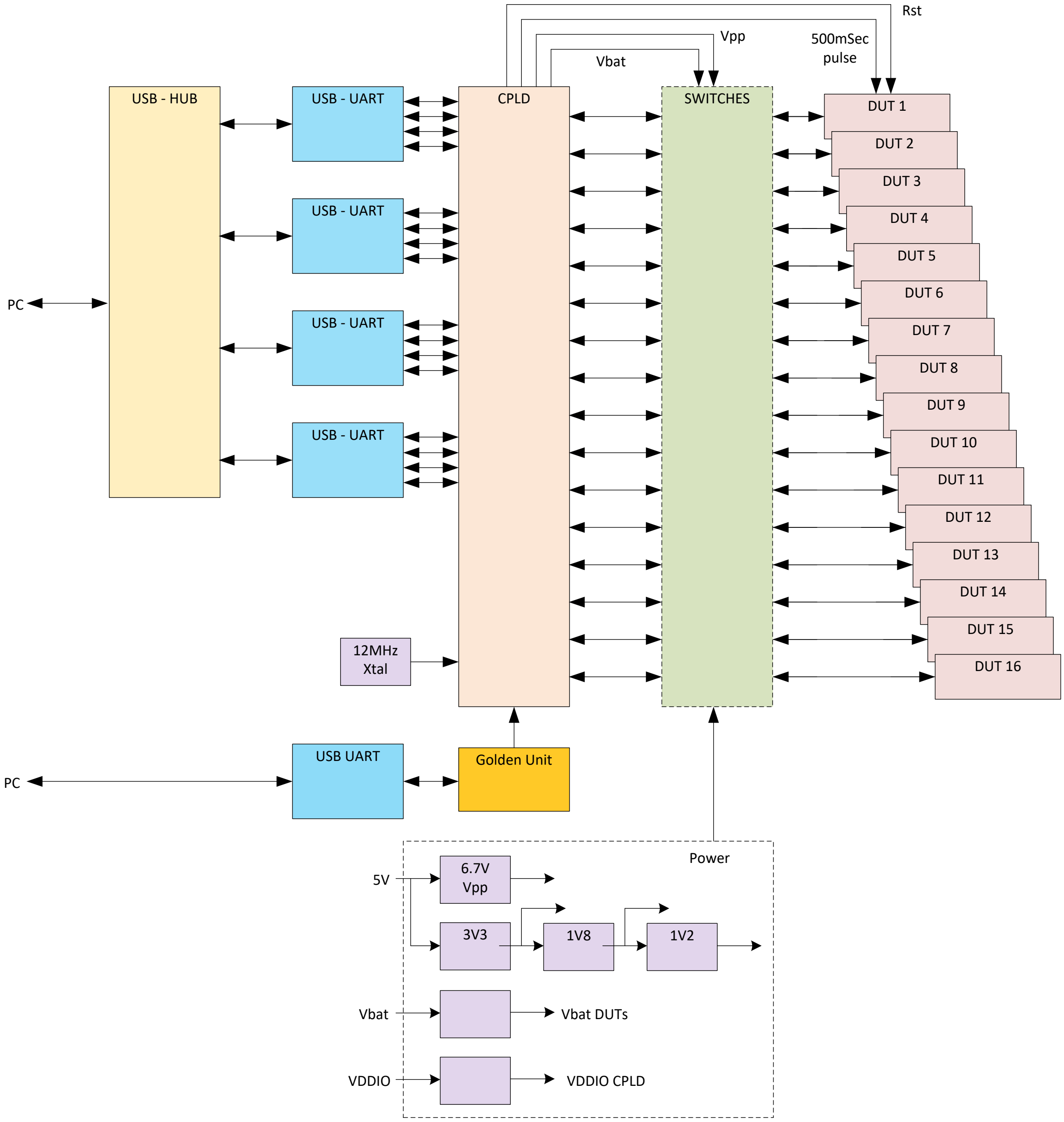


Figure 17: PLT Functional Blocks

6 Software

6.1 Introduction

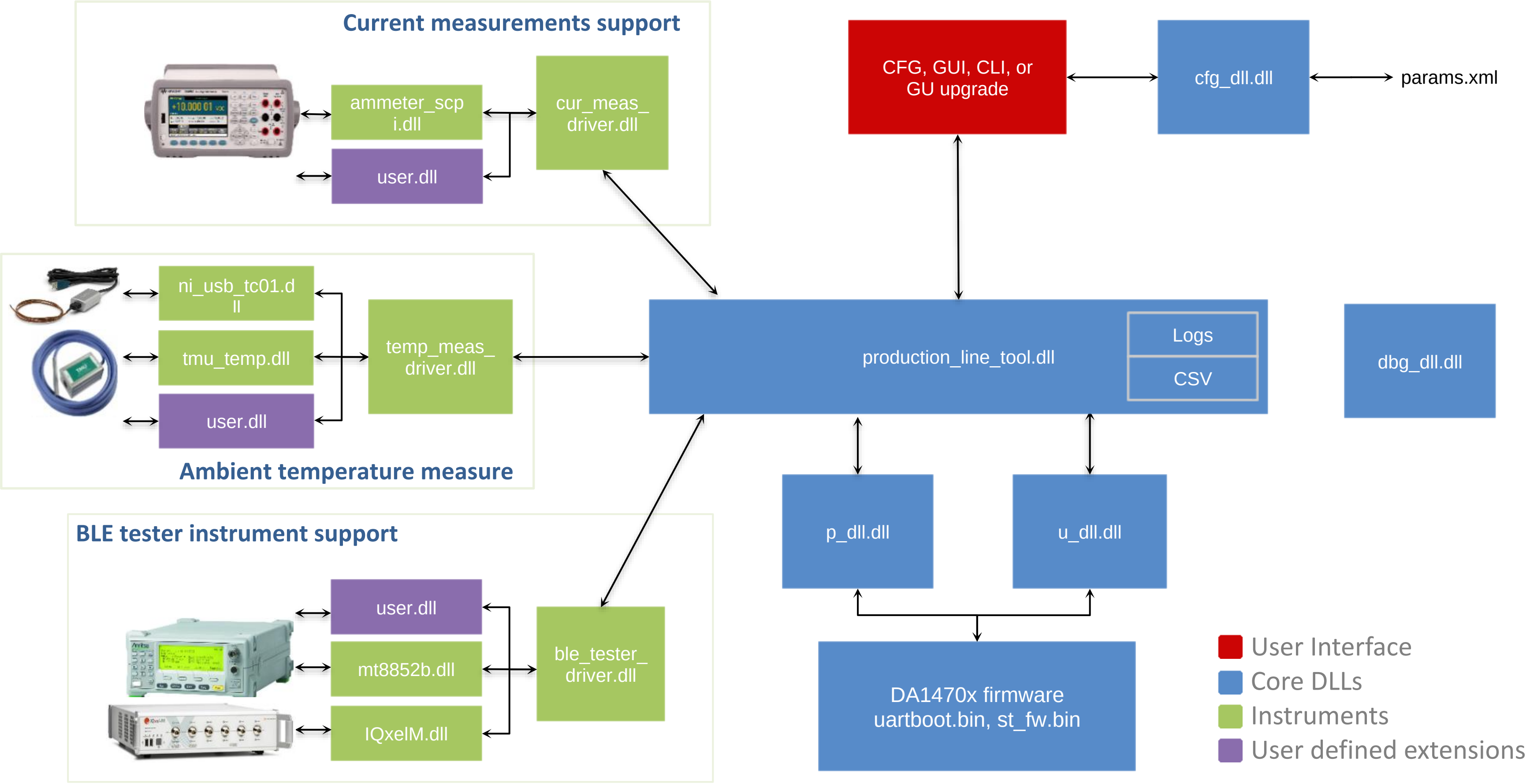


Figure 18: Production Line Tool Software Block Diagram

The Production Line Tool software is a collection of software blocks that interact with each other, as shown in Figure 18. Its main purpose is to communicate with the PLT hardware and the DUTs to be able to run the production tests and perform memory operations. The software blocks can be arranged in four main groups:

- **Red blocks:** User Interface (UI) applications
- **Blue blocks:** Core libraries
- **Green blocks:** Instrument interface libraries
- **Purple blocks:** User defined extensions

Core libraries, instrument interface libraries and user-defined extension APIs can be found in the HTML help inside the source PLT directory. The User Interface applications block consists of four application executables. For details, see Section7.

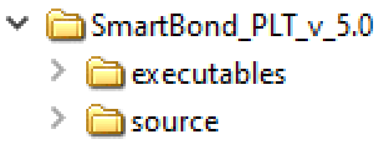
Table 5: PLT User Interface Application Executables

Short Name	File Name	Description
CFG PLT	`SmartBond_PLT_CFG.exe`	Configuration application. Load, edit and save the test parameters and the memory actions to be performed during device testing.
GUI PLT	`SmartBond_PLT_GUI.exe`	Graphical User Interface (GUI) application. Performs the actual device validation and memory programming. Provides a visual indication of the test results and access to the result logs.
CLI PLT	`SmartBond_PLT_CLI.exe`	The same as the GUI PLT but console based.
GU upgrade	`GU_fw_upgrade.exe`	A Graphical User Interface (GUI) application, which is used to easily upgrade the firmware of the Golden Unit.

6.2 Software Package Contents

The PLT software release package comes in a compressed folder `SmartBond_PLT_v_X.X.zip`, where '`X`' represents the version number of the current PLT release.

Figure 19 illustrates the main folders of the PLT software package. Folder `executables` holds all the executables and libraries needed for the PLT to run on a Windows 7/8/8.1/10 machine. Folder `source` contains the entire source code of the PLT, organized in a Visual Studio Express 2017 solution.



SmartBond_PLT_v_5.0
executables
source

Figure 19: SmartBond PLT Software Package Contents

Table 6 gives a short description of the files and folders contained in the `executables` directories.

Table 6: Executables Folder Description

File or Folder	Description
`ammeter_instr_plugins/`	Contains the current measurement instrument DLLs, used during the current measurement tests.
`ammeter_instr_plugins/ammeter_scpi.dll`	This is the DLL for taking current measurements using a DMM that supports the standard SCPI commands. NI-VISA is also used for this purpose. Example DMM

File or Folder	Description
	instruments are the Keysight 34401A Ref. [6], the Keithley 2000 Ref. [7] or the Keysight 34461A Ref. [11]. The PLT has been tested with all three instruments.
binaries/	Contains the necessary firmware binaries used during DUT testing.
binaries/GU/prod_test_GU.bin	Contains the GU latest firmware binary. Users should better upgrade their PLT hardware with the GU firmware contained in this folder.
binaries/GU/flash_programmer_GU.bin	This firmware is used to upgrade the GU firmware by the GU_fw_upgrade.exe application.
ble_tester_instr_plugins/	Contains the Bluetooth® LE tester instrument DLLs.
ble_tester_instr_plugins/mt8852b.dll	This is the DLL that performs the Direct Test Mode RF tests using the Anritsu MT8852B instrument Ref. [5]. **Note:** There is an issue in Anritsu MT8852B firmware version 4.20.000 and should be upgraded to the latest one. Latest MT8852B instrument firmware can be downloaded from the following link: https://www.anritsu.com/en-US/test-measurement/support/downloads?model=MT8852B
ble_tester_instr_plugins/IQxelM.dll	This is the DLL that performs the Direct Test Mode RF tests using the Litepoint IQxel-M instrument Ref. [10].
icons/	Contains pictures used by the PLT applications.
IQmeasure_3.1.2/	Contains specific Litepoint IQxel-M DLLs as released by Litepoint.
params/	Contains the configuration params.xml file, the XML schema params.xsd and a sample of BD address file named bd_address.ini.
params/custom_mem_data.csv	This is a sample CSV file to be used in the custom memory burn action. Users could edit this file and add their own specific memory data to be burned by the PLT. The PLT will match the entries in the CSV file using the BD addresses. The format of the file is explained later.
scripts/	Contains sample batch script files. User can select batch script files to be executed by the PLT before and after each test.
scripts/run_before_tests.cmd	An example script that copies and renames binaries from a directory to a folder required by the PLT. This folder is accessed by the PLT to read and burn different binaries per DUT.
scripts/run_after_tests.cmd	An example script that moves all logs files, except the ones with the current date, to a specific folder.
temp_meas_instr_plugins/	Contains the temperature measurement instrument DLLs.
temp_meas_instr_plugins/ni_usb_tc01.dll	The ni_usb_tc01.dll is the DLL used to interface a NI USB TC01 Ref. [9] temperature sensor for temperature measurements.
temp_meas_instr_plugins/tmu_temp_sens.dll	The tmu_temp_sens.dll is the DLL used to interface a Papouch TMU sensor Ref. [7] for temperature measurements.

File or Folder	Description
SmartBond_PLT_CFG.exe	This is the configuration application. It is a graphical user interface application used to edit the PLT test configuration parameters, saved in an XML file, params.xml.
SmartBond_PLT_CLI.exe	This is the command line interface tool. It performs the production tests and memory programming through a console.
SmartBond_PLT_GUI.exe	This is the graphical user interface tool. It performs the production tests and memory programming through a graphical user interface.
GU_fw_upgrade.exe	This is the Golden Unit firmware upgrade application.
ammeter_driver.dll/.lib	This DLL loads and accesses all DMM instrument DLLs from inside the ammeter_instr_plugins. It acts as an intermediate layer between the prod_line_tool_dll.dll and the instrument DLLs.
barcode_scanner.dll/.lib	This DLL receives BD addresses from a barcode scanner with USB to serial interface. Has been tested with Honeywell Xenon 1900 and the Motorola LS2208 barcode scan readers Ref. [1].
ble_tester_driver.dll/.lib	This DLL loads and accesses all Bluetooth® LE tester instrument DLLs from inside ble_tester_instr_plugins folder.
cfg_dll.dll/.lib	This is the configuration parameter handling DLL. It can validate, load, and save parameters from a given XML file.
dbg_dll.dll/.lib	The dbg_dll.dll file is a DLL used to print debug messages to a file or to a debug console.
ftd2xx.dll	This is the FTDI DLL. Used to hard reset the Golden Unit from the application whenever needed through an FTDI GPIO pin.
p_dll.dll/.lib	This is the production test DLL that performs device functional tests.
prod_line_tool_dll.dll/.lib	This is the core DLL. The heart of the system that performs the state machines for all tests and memory actions to be executed. It is responsible to log the results and notify the user interfaces about the current device test status.
temp_meas_driver.dll/.lib	This is the temperature measurement driver DLL. It loads and accesses all temperature measurement DLLs from inside the temp_meas_instr_plugins folder.
u_dll.dll/.lib	This is the DLL that performs the memory actions, like the memory programming, erasing, etc.
vc_redist.x86.exe/vc_redist.x64.exe	These are the Visual Studio 2017 Express redistributable packages for 32 and 64-bit machines. For installing these, users should agree to the license requirements described during the installation of any of these packages. It is also found here: https://www.visualstudio.com/license-terms/mt171551/.

6.3 Prerequisites

Before executing the code, the packages indicated in Table 7 should be installed on the PC. Some are required only if users would like to build the PLT code. Others are optional depending on the tests or actions needed.

Table 7: Production Line Tool Prerequisites

Item	Optional	Description
Visual Studio 2017 Express	Yes	The IDE used to edit and debug the Production Line Tool. This is only required if users would like to edit the software.
vc_redist.x86.exe	No	Already described in Table 6. Users should agree to the license requirements described during the installation of any of these packages. The license file can also be found here: https://www.visualstudio.com/license-terms/mt171551/.
MSXML6	No	Installed by default in Win 10/11.
.NET framework 4.5	No	Needed for the graphical user interface applications.
Latest FTDI drivers	No	Tested with FTDI v2.12.24, v2.12.26, v2.12.28 and v2.12.36.4 drivers.
NI-VISA 15.5	Yes	Used for optional instrument control, like Bluetooth® LE tester and voltage meter. NI-VISA 15.5 can be downloaded from http://www.ni.com/download/ni-visa-15.5/5846/en/
NI-488.2 15.5	Yes	Used for instrument control, like Bluetooth® LE tester and DMM. NI-488.2 15.5 can be downloaded from http://www.ni.com/download/ni-488.2-15.5/5859/en/
NI-DAQmx	Yes	Used for optional instrument control like temperature measurements using the NI USB TC01 sensor.

6.4 System Requirements

Table 8 contains the minimum system requirements for the PLT to operate.

Table 8: Minimum System Requirements

Item	Minimum Requirements	
Operating system	Windows 10	
CPU	Quad Core CPU	
Memory	4 GB RAM or larger. Each device log can reach up to 40 kB.	
Hard drive	For 100000 devices, at least **4 GB** of available hard disk is required.	
Monitor resolution	1280 x 768 or higher	
Monitor DPI	Smaller – 100 % = 96 DPI	Supported
	Medium – 125 % = 120 DPI	Supported
	Larger – 150 % = 144 DPI	Not supported

6.5 Limitations

Parallel control of multiple PLT hardware boards on the same PC is **not supported**.

However, by correctly setting up the system, two or more PLT hardware boards could be connected and controlled by multiple GUI PLT application instances on the same PC, but the tests should only be executed **sequentially**. The main reasons for this limitation are indicated below:

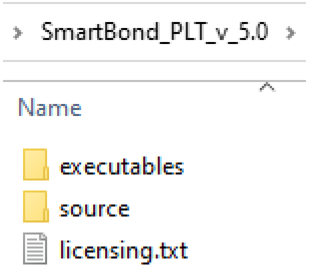
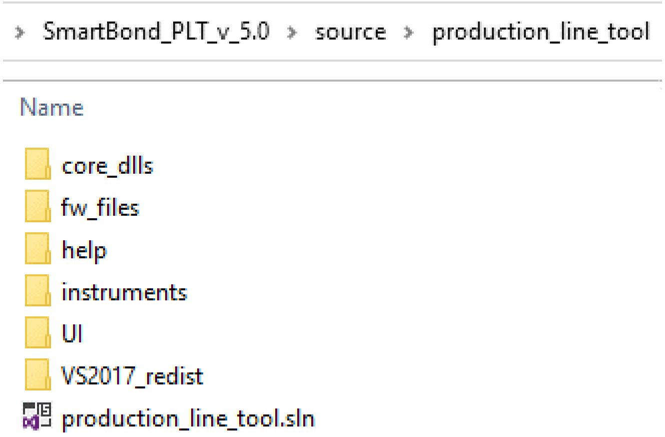
- The GU FT232 FTDI IC is programmed with a special serial string, "`DialogSemi`" (see Table 68). This is used in the '`GU COM port find`' PLT operation. This operation searches all PC connected FTDIs to find the serial string "`DialogSemi`". When found, it saves it as the GU COM port number to be used by the PLT. The '`GU COM port find`' operation will open and lock, for a short period of time, all Windows COM ports, one by one, even the ones used by the other PLT hardware. If the second GUI PLT application instance is performing test operations at the same time and wants to open its DUT COM ports, the operation may fail.
- When the GUI PLT application starts the test operations, it performs a DUT COM port enumeration. During this process, the GU sets the CPLD in UART loopback mode. It opens all PC COM ports one by one and sends a specific word, while trying to see if it receives it back. During this process, other PLTs may need to work with 'their' DUT COM ports, which may happen to be currently used by the 'DUT COM port enumeration' process of the first PLT.
- GU hardware reset. In every PLT test run a GU HW reset is issued from the PLT software using a specific GU FTDI GPIO pin. To access the GU FTDI, the FTDI API is used from `ftd2xx.dll`. To access the FTDI hardware and read the serial number through the FTDI `ftd2xx.dll` the `FT_Open` API is used on all PC COM ports, one by one. Since `FT_Open` is used in all PC COM ports, conflicts could arise if other PLTs would also like to use these COM ports.
- BD addresses handling. Usually, the PLT automatically sets the DUT BD addresses by increasing them one by one. Special care should be taken to work with multiple PLT hardware and software. Most probably, two different BD address files should be used for each PLT hardware.

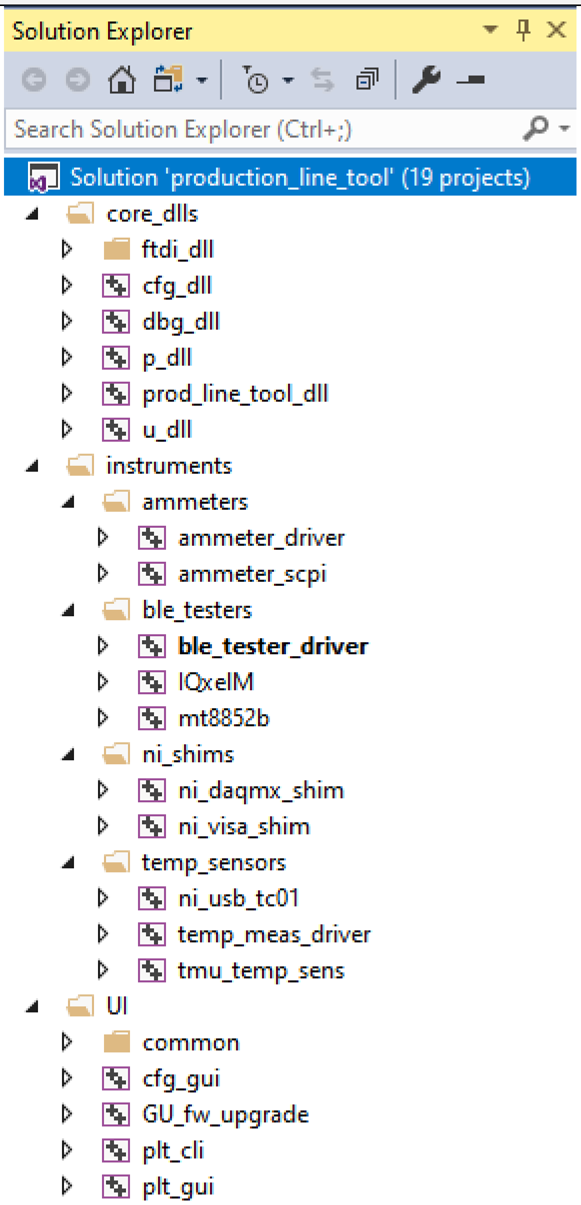
6.6 Building the Code

The PLT software release package contains not only application executables for directly performing the tests out of the box, but also the entire source code of the tools. This is organized in a Visual Studio 2017 Express solution.

To open the Visual Studio 2017 Express PLT source code solution the following steps should be executed (see Table 9).

Table 9: Opening the PLT Visual Studio 2017 Express Source Code Solution

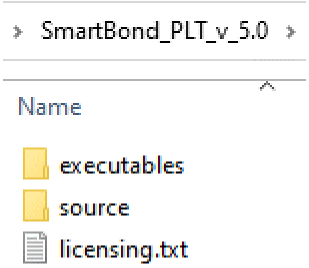
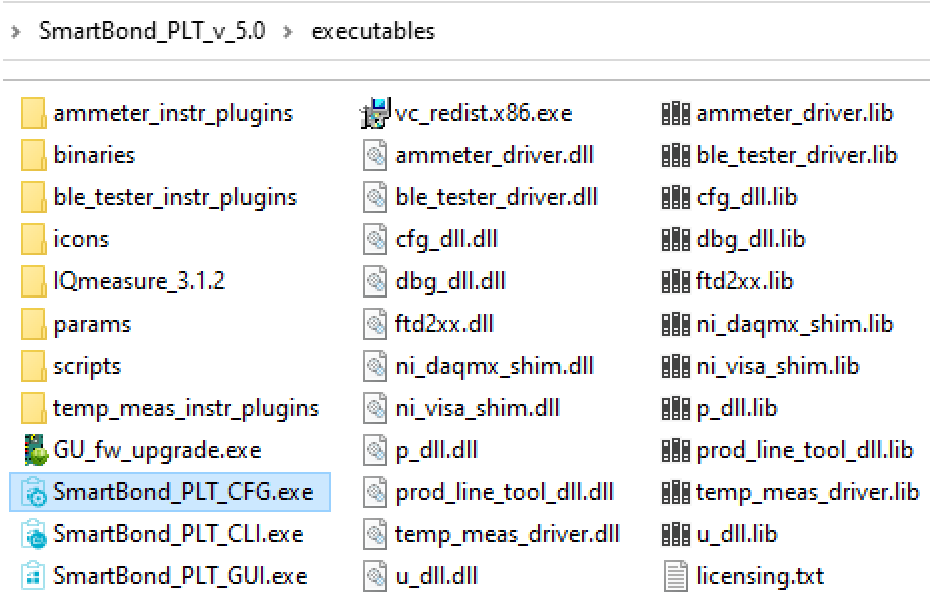
Step	Description
1	Download the latest PLT software package (e.g., `SmartBond_PLT_v_4.x.zip`)
2	Extract the software package. The following two folders should exist.  SmartBond_PLT_v_5.0 Name executables source licensing.txt
3	Go to folder '`source\production_line_tool`'. The following files and folders should exist.  SmartBond_PLT_v_5.0 › source › production_line_tool Name core_dlls fw_files help instruments UI VS2017_redist production_line_tool.sln
4	Double click the `production_line_tool.sln` Visual Studio 2017 Express solution file. The Visual Studio 2017 Express application will start and the PLT Solution Explorer should be shown.

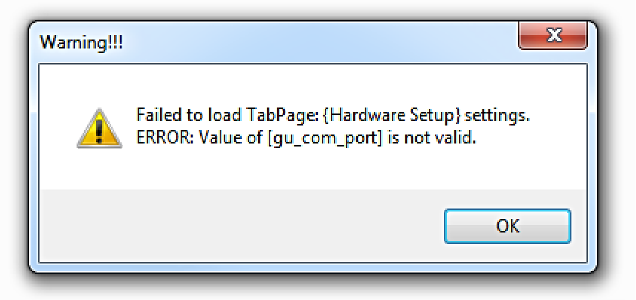
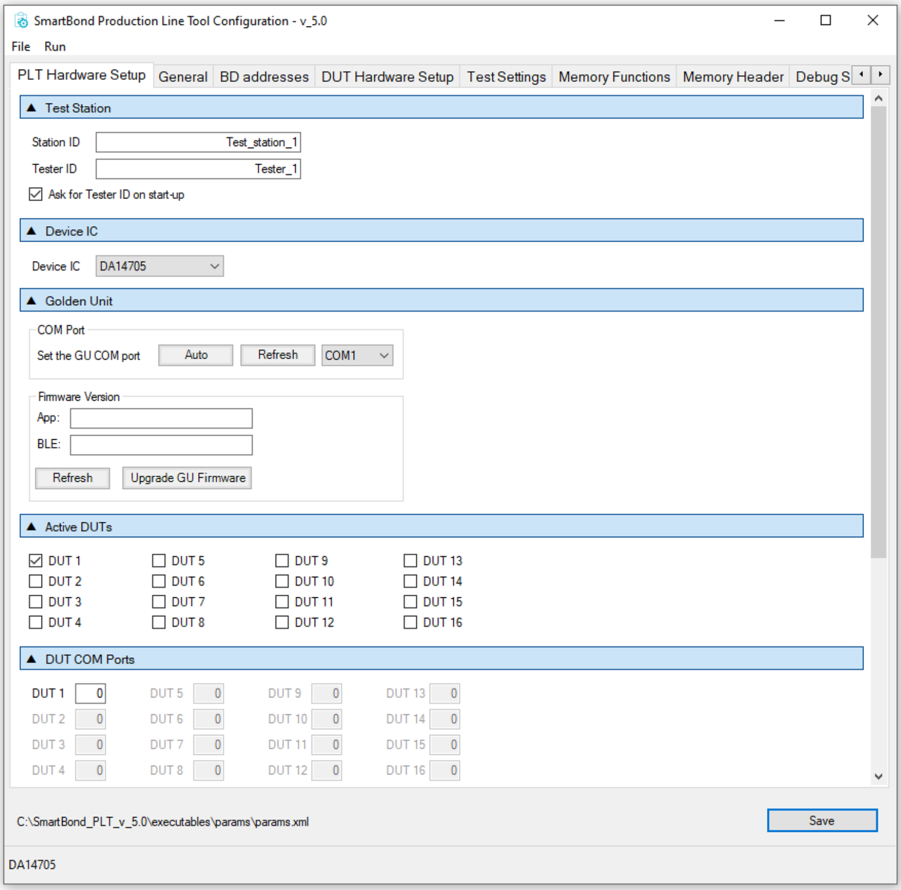
Step	Description
	 Solution Explorer Search Solution Explorer (Ctrl+;) Solution 'production_line_tool' (19 projects) core_dlls ftdi_dll cfg_dll dbg_dll p_dll prod_line_tool_dll u_dll instruments ammeters ammeter_driver ammeter_scpi ble_testers **ble_tester_driver** IQxelM mt8852b ni_shims ni_daqmx_shim ni_visa_shim temp_sensors ni_usb_tc01 temp_meas_driver tmu_temp_sens UI common cfg_gui GU_fw_upgrade plt_cli plt_gui
5	To build the code follow the process described in the *UM-B-040* document Ref. [1]. **Note:** There is no need to unload any projects for the solution to be built. This was fixed in PLT v4.4 and onwards by dynamically linking DLLs.

6.7 Executing the Applications

To execute the Production Line Tool applications, the process described in the following tables should be followed.

Table 10: SmartBond_PLT_CFG.exe Application Execution

Step	Description
1	Download the latest PLT software package (e.g., `SmartBond_PLT_v_x.x.zip`).
2	Extract the software package. The following two folders should exist.  › SmartBond_PLT_v_5.0 › Name executables source licensing.txt
3	Go to folder '`executables`'. This folder should contain the following files and sub-folders.  › SmartBond_PLT_v_5.0 › executables ammeter_instr_plugins binaries ble_tester_instr_plugins icons IQmeasure_3.1.2 params scripts temp_meas_instr_plugins GU_fw_upgrade.exe SmartBond_PLT_CFG.exe SmartBond_PLT_CLI.exe SmartBond_PLT_GUI.exe vc_redist.x86.exe ammeter_driver.dll ble_tester_driver.dll cfg_dll.dll dbg_dll.dll ftd2xx.dll ni_daqmx_shim.dll ni_visa_shim.dll p_dll.dll prod_line_tool_dll.dll temp_meas_driver.dll u_dll.dll ammeter_driver.lib ble_tester_driver.lib cfg_dll.lib dbg_dll.lib ftd2xx.lib ni_daqmx_shim.lib ni_visa_shim.lib p_dll.lib prod_line_tool_dll.lib temp_meas_driver.lib u_dll.lib licensing.txt

Step	Description
4	Double click the `SmartBond_PLT_CFG.exe` application executable. Most probably, the following warning will be shown.  Warning!!! Failed to load TabPage: {Hardware Setup} settings. ERROR: Value of [gu_com_port] is not valid. OK During start-up, the `SmartBond_PLT_CFG.exe` application loads the `Hardware` configuration parameters from the `params.xml` file. These parameters also contain the GU COM port. The default `params.xml` file has the GU COM port set to 4. If this COM port number does not exist in the PC, then this warning message will be shown. Therefore, this warning message indicates that the default GU COM port set in the `params.xml` file is not valid or that the GU USB cable is not connected to the PC. Press 'OK' if the warning message appears.
5	The application will start, and the initial *Hardware Setup* screen will be shown.  SmartBond Production Line Tool Configuration - v_5.0 File Run PLT Hardware Setup | General | BD addresses | DUT Hardware Setup | Test Settings | Memory Functions | Memory Header | Debug S ▲ Test Station Station ID: Test_station_1 Tester ID: Tester_1 ☑ Ask for Tester ID on start-up ▲ Device IC Device IC: DA14705 ▲ Golden Unit COM Port Set the GU COM port: Auto | Refresh | COM1 Firmware Version App: BLE: Refresh | Upgrade GU Firmware ▲ Active DUTs ☑ DUT 1 ☐ DUT 5 ☐ DUT 9 ☐ DUT 13 ☐ DUT 2 ☐ DUT 6 ☐ DUT 10 ☐ DUT 14 ☐ DUT 3 ☐ DUT 7 ☐ DUT 11 ☐ DUT 15 ☐ DUT 4 ☐ DUT 8 ☐ DUT 12 ☐ DUT 16 ▲ DUT COM Ports DUT 1: 0 DUT 5: 0 DUT 9: 0 DUT 13: 0 DUT 2: 0 DUT 6: 0 DUT 10: 0 DUT 14: 0 DUT 3: 0 DUT 7: 0 DUT 11: 0 DUT 15: 0 DUT 4: 0 DUT 8: 0 DUT 12: 0 DUT 16: 0 C:\SmartBond_PLT_v_5.0\executables\params\params.xml Save DA14705

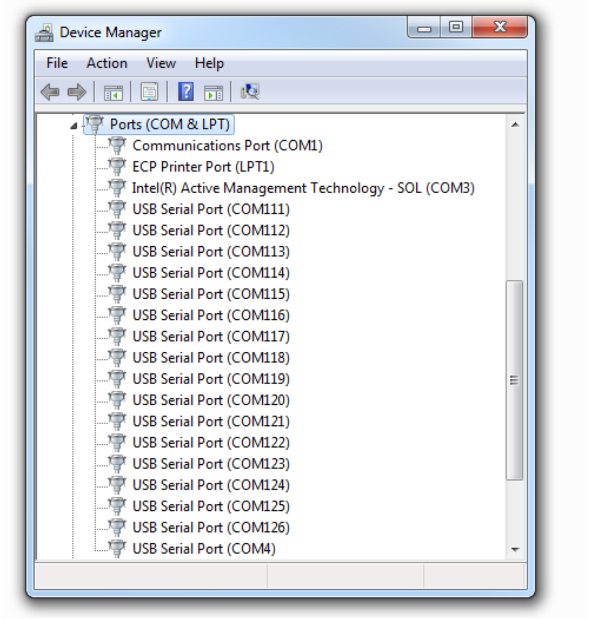
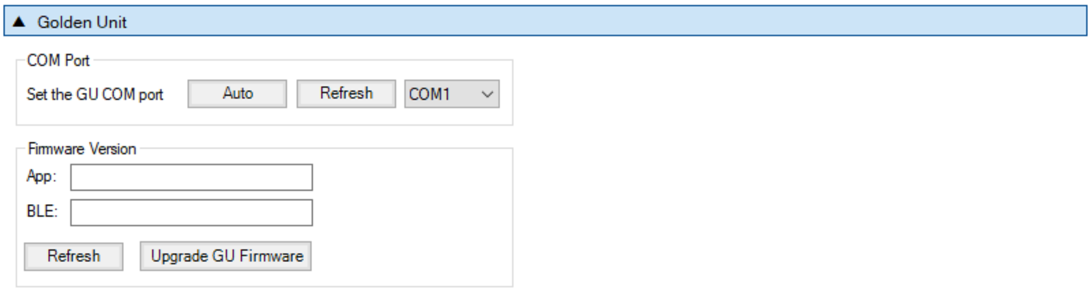
Step	Description
6	Connect the PLT HW to the PC. Connect the GU and the DUT USB cables to the PC. Check the Windows Device Manager that 17 new COM ports were found, 16 for the DUTs and 1 for the GU. The following screenshot is an example of a Device Manager COM ports for a PC that has the PLT connected.  Device Manager Ports (COM & LPT) Communications Port (COM1) ECP Printer Port (LPT1) Intel(R) Active Management Technology - SOL (COM3) USB Serial Port (COM111) USB Serial Port (COM112) USB Serial Port (COM113) USB Serial Port (COM114) USB Serial Port (COM115) USB Serial Port (COM116) USB Serial Port (COM117) USB Serial Port (COM118) USB Serial Port (COM119) USB Serial Port (COM120) USB Serial Port (COM121) USB Serial Port (COM122) USB Serial Port (COM123) USB Serial Port (COM124) USB Serial Port (COM125) USB Serial Port (COM126) USB Serial Port (COM4)
7	On the SmartBond_PLT_CFG.exe Hardware Setup initial screen click **Auto** to automatically find the GU COM port among the 17 Windows enumerated COM ports. The **Auto** button will turn green if successful. Press **Save*** to save the new GU COM port in the params.xml file.  Golden Unit COM Port Set the GU COM port: COM1 Firmware Version App: BLE:

Table 11: SmartBond_PLT_GUI.exe Application Execution

Step	Description
1	To successfully start the SmartBond_PLT_GUI.exe application, the SmartBond_PLT_CFG.exe should be executed first in order to set up the system and perform the required tests. See Table 10.
2	Go to folder 'executables'. This folder should contain the following files and sub-folders. Double click on SmartBond_PLT_GUI.exe.

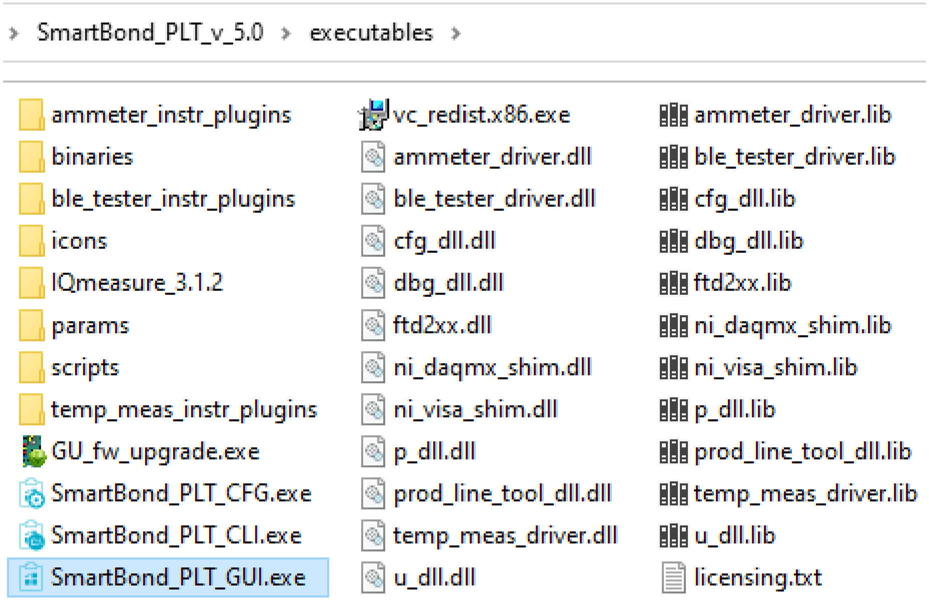
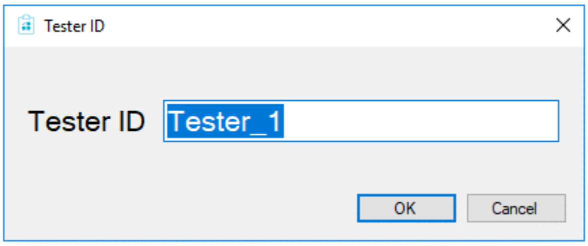
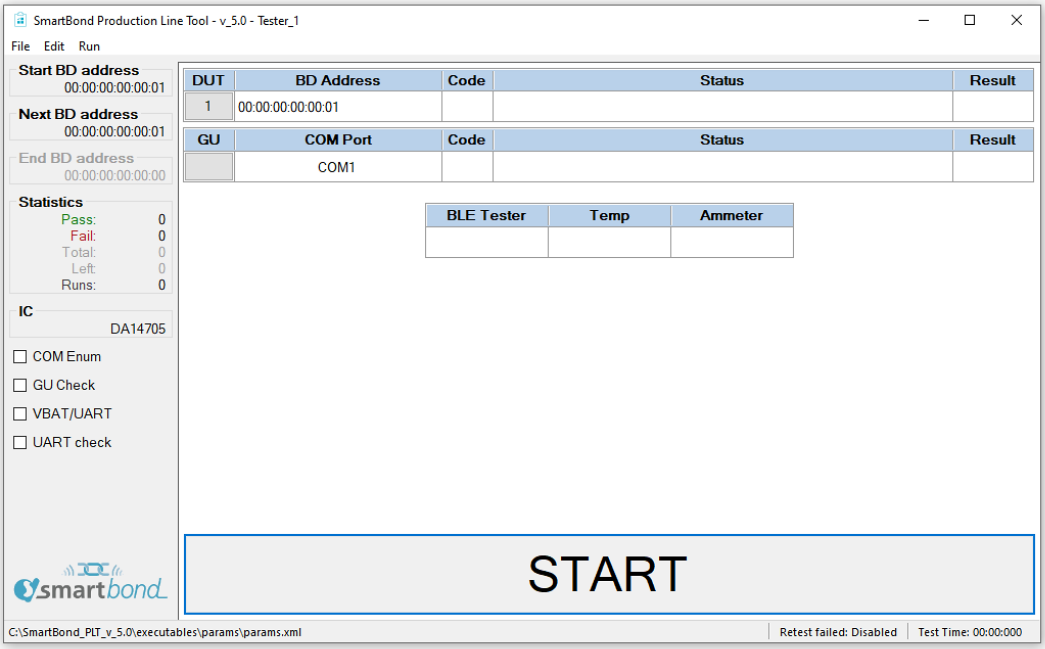
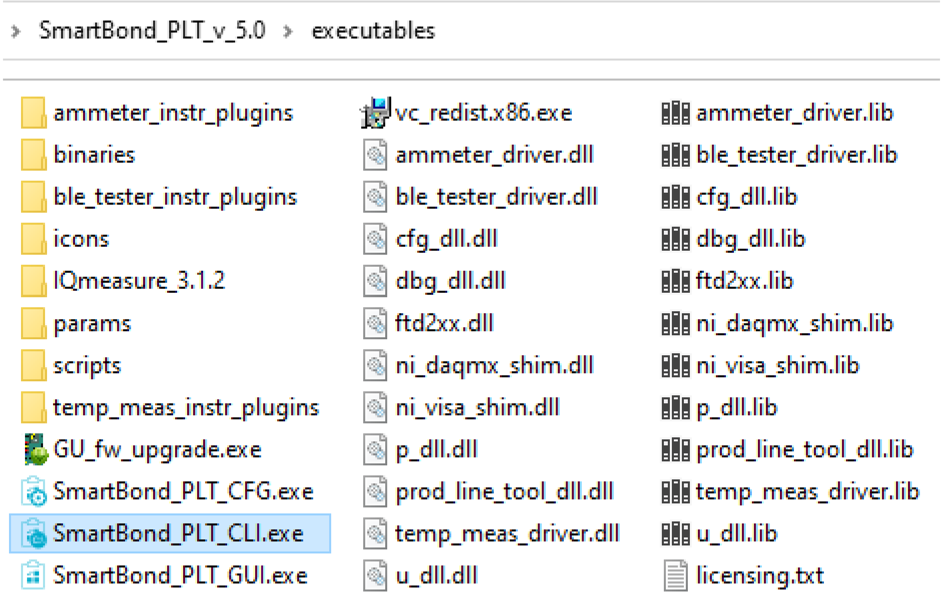
Step	Description
	 SmartBond_PLT_v_5.0 > executables > ammeter_instr_plugins binaries ble_tester_instr_plugins icons IQmeasure_3.1.2 params scripts temp_meas_instr_plugins GU_fw_upgrade.exe SmartBond_PLT_CFG.exe SmartBond_PLT_CLI.exe SmartBond_PLT_GUI.exe vc_redist.x86.exe ammeter_driver.dll ble_tester_driver.dll cfg_dll.dll dbg_dll.dll ftd2xx.dll ni_daqmx_shim.dll ni_visa_shim.dll p_dll.dll prod_line_tool_dll.dll temp_meas_driver.dll u_dll.dll ammeter_driver.lib ble_tester_driver.lib cfg_dll.lib dbg_dll.lib ftd2xx.lib ni_daqmx_shim.lib ni_visa_shim.lib p_dll.lib prod_line_tool_dll.lib temp_meas_driver.lib u_dll.lib licensing.txt
3	A new window will ask for the Tested ID name. Enter the name and click OK.  Tester ID Tester ID: Tester_1 OK Cancel
3	The following initial screen will appear.  SmartBond Production Line Tool - v_5.0 - Tester_1 File Edit Run Start BD address: 00:00:00:00:00:01 Next BD address: 00:00:00:00:00:01 End BD address: 00:00:00:00:00:00 Statistics Pass: 0 Fail: 0 Total: 0 Left: 0 Runs: 0 IC: DA14705 ☐ COM Enum ☐ GU Check ☐ VBAT/UART ☐ UART check DUT | BD Address | Code | Status | Result 1 | 00:00:00:00:00:01 | | | GU | COM Port | Code | Status | Result | COM1 | | | BLE Tester | Temp | Ammeter START C:\SmartBond_PLT_v_5.0\executables\params\params.xml Retest failed: Disabled Test Time: 00:00:000
4	By clicking the keyboard spacebar, the **START** button will be pressed and the preconfigured tests and memory actions will start to be executed.

Table 12: SmartBond_PLT_CLI.exe Application Execution

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>1</td><td>To successfully start the SmartBond_PLT_CLI.exe application, the SmartBond_PLT_CFG.exe should be executed first, in order to set up the system and perform the required tests. See Table 10.</td></tr>
<tr><td>2</td><td>Go to folder 'executables'. This folder should contain the following files and sub-folders.

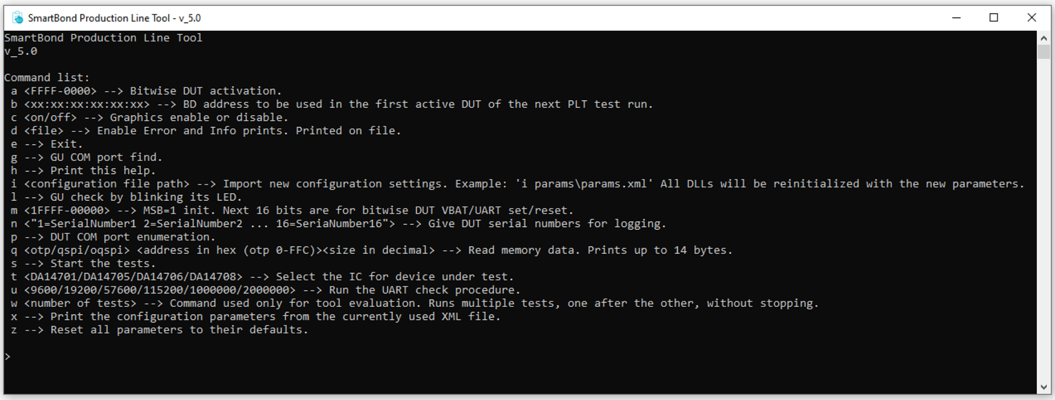


› SmartBond_PLT_v_5.0 › executables

ammeter_instr_plugins
binaries
ble_tester_instr_plugins
icons
IQmeasure_3.1.2
params
scripts
temp_meas_instr_plugins
GU_fw_upgrade.exe
SmartBond_PLT_CFG.exe
SmartBond_PLT_CLI.exe
SmartBond_PLT_GUI.exe

vc_redist.x86.exe
ammeter_driver.dll
ble_tester_driver.dll
cfg_dll.dll
dbg_dll.dll
ftd2xx.dll
ni_daqmx_shim.dll
ni_visa_shim.dll
p_dll.dll
prod_line_tool_dll.dll
temp_meas_driver.dll
u_dll.dll

ammeter_driver.lib
ble_tester_driver.lib
cfg_dll.lib
dbg_dll.lib
ftd2xx.lib
ni_daqmx_shim.lib
ni_visa_shim.lib
p_dll.lib
prod_line_tool_dll.lib
temp_meas_driver.lib
u_dll.lib
licensing.txt</td></tr>
<tr><td>3</td><td>Double click the SmartBond_PLT_CLI.exe application executable. The following initial screen will appear.



SmartBond Production Line Tool - v_5.0

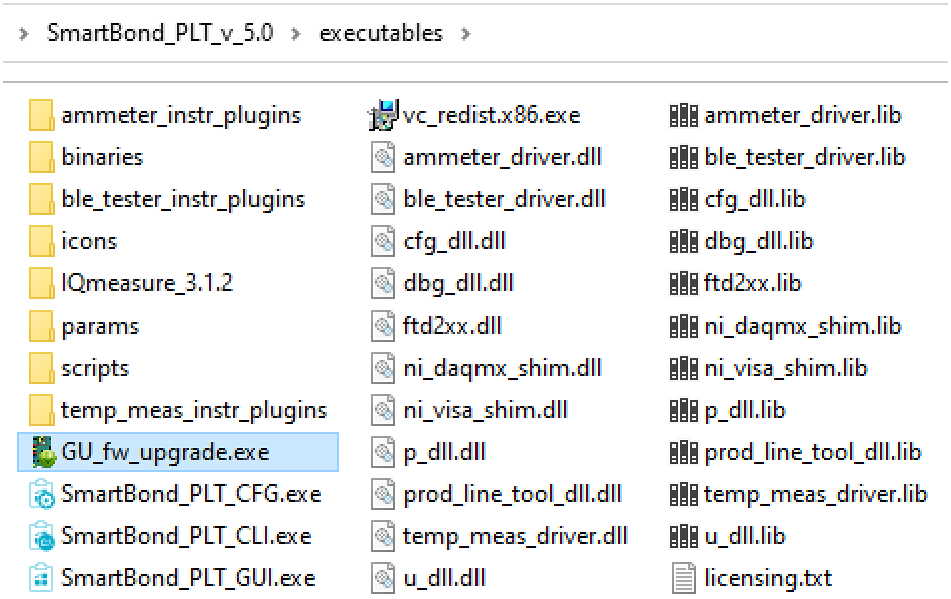
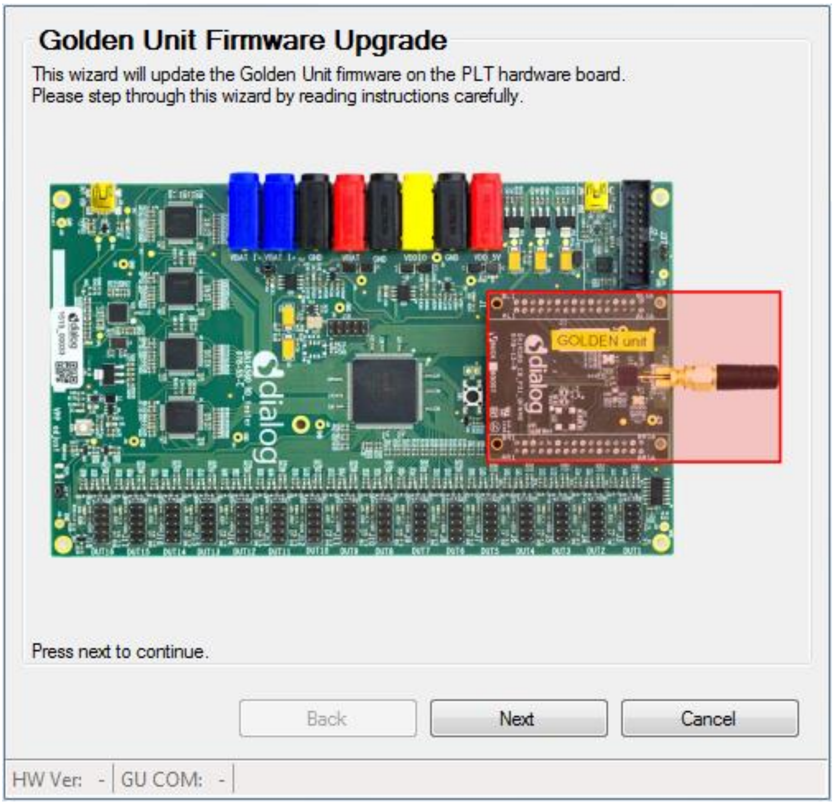
```
SmartBond Production Line Tool
v_5.0

Command list:
 a <FFFF-0000> --> Bitwise DUT activation.
 b <xx:xx:xx:xx:xx:xx> --> BD address to be used in the first active DUT of the next PLT test run.
 c <on/off> --> Graphics enable or disable.
 d <file> --> Enable Error and Info prints. Printed on file.
 e --> Exit.
 g --> GU COM port find.
 h --> Print this help.
 i <configuration file path> --> Import new configuration settings. Example: 'i params\params.xml' All DLLs will be reinitialized with the new parameters.
 l --> GU check by blinking its LED.
 m <1FFFF-00000> --> MSB=1 init. Next 16 bits are for bitwise DUT VBAT/UART set/reset.
 n <"1=SerialNumber1 2=SerialNumber2 ... 16=SeriaNumber16"> --> Give DUT serial numbers for logging.
 p --> DUT COM port enumeration.
 q <otp/qspi/oqspi> <address in hex (otp 0-FFC)><size in decimal> --> Read memory data. Prints up to 14 bytes.
 s --> Start the tests.
 t <DA14701/DA14705/DA14706/DA14708> --> Select the IC for device under test.
 u <9600/19200/57600/115200/1000000/2000000> --> Run the UART check procedure.
 w <number of tests> --> Command used only for tool evaluation. Runs multiple tests, one after the other, without stopping.
 x --> Print the configuration parameters from the currently used XML file.
 z --> Reset all parameters to their defaults.

>
```
</td></tr>
<tr><td>4</td><td>Type 's' then click Enter. The preconfigured tests and memory actions will start to be executed.</td></tr>
</table>

Table 13: GU_fw_upgrade.exe Application Execution

Step	Description
1	GU_fw_upgrade.exe can be started by either opening the application from the executables folder or by pressing the 'Upgrade GU Firmware' button in the PLT Hardware Setup tab in SmartBond_PLT_CFG.exe.

Step	Description
2	Go to folder 'executables'. This folder should contain the following files and sub-folders.  › SmartBond_PLT_v_5.0 › executables › ammeter_instr_plugins binaries ble_tester_instr_plugins icons IQmeasure_3.1.2 params scripts temp_meas_instr_plugins GU_fw_upgrade.exe SmartBond_PLT_CFG.exe SmartBond_PLT_CLI.exe SmartBond_PLT_GUI.exe vc_redist.x86.exe ammeter_driver.dll ble_tester_driver.dll cfg_dll.dll dbg_dll.dll ftd2xx.dll ni_daqmx_shim.dll ni_visa_shim.dll p_dll.dll prod_line_tool_dll.dll temp_meas_driver.dll u_dll.dll ammeter_driver.lib ble_tester_driver.lib cfg_dll.lib dbg_dll.lib ftd2xx.lib ni_daqmx_shim.lib ni_visa_shim.lib p_dll.lib prod_line_tool_dll.lib temp_meas_driver.lib u_dll.lib licensing.txt
3	Double click the `GU_fw_upgrade.exe` application executable. The following initial screen will appear.  Golden Unit Firmware Upgrade This wizard will update the Golden Unit firmware on the PLT hardware board. Please step through this wizard by reading instructions carefully. Press next to continue. GOLDEN unit dialog Back Next Cancel HW Ver: - GU COM: -
4	Follow the instructions to guide you to successfully configure and select the Golden Unit. A detailed procedure is explained in Section 7.5.

6.8 Test Sequence

This section describes the sequence of steps involved for the DA1470x device testing. It outlines all the steps the PLT follows to successfully test a device.

6.8.1 DA1470x Test Sequence

Table 14 describes each step that the PLT undertakes to validate and program DA1470x-based devices. Some of the steps are optional and will only be executed if the equivalent actions are enabled in the configuration parameters.

The entire test sequence for the DA1470x DUTs is shown in Figure 20.

Table 14: DA1470x Test Sequence

Step	Action	Opt.	Description
1	Statistics update	No	Update the total tests executed.
2	BD addresses	No	Update the BD addresses for all DUTs.
3	Configuration parameters	No	Configuration parameters are passed from the CLI or GUI to the `prod_line_tool_dll.dll`. If any of the parameters is not valid, an error will occur.
4	Reset GU	No	GU hardware reset by controlling an FT232 pin.
5	Initialize PLT hardware (CPLD)	No	The GU will set the CPLD to an initial known state.
6	Check temperature sensor instrument	Yes	Check whether the temperature measurement instrument is online, only if the temperature measurement test is active.
7	Check Bluetooth® LE tester instrument	Yes	Check whether the Bluetooth® LE tester instrument is online, only if any of the Bluetooth® LE tester test operations is active.
8	Check ammeter instrument	Yes	Check whether the ammeter is online, only if any of the current measurement tests is active.
9	Toggle GU LED	No	Toggle the GU red LED on the PLT hardware to indicate that the GU is alive.
10	Check DUT COM ports	No	Check whether the PLT has identified the DUT COM ports and if not run the automatic DUT COM port identification.
11	Temperature measure	Yes	If the temperature measurement test is active, take a measurement and log it to all DUT logs and in the CSV file.
12	Power-up DUTs	Yes	Provide power to the DUTs so the boot process can start.
13	Download `uartboot_da1470x.bin`	Yes	If any of the production tests is active the `uartboot_da1470x.bin` will be downloaded so `st_da1470x.bin` can be downloaded to the system RAM. In addition, if the GPIO watchdog option is enabled, it will start toggling after the `uartboot_da1470x.bin` is loaded and right before the production test download.
14	Get `uartboot_da1470x.bin` version.	Yes	After `uartboot_da1470x.bin` has been downloaded, commands can be sent to it. A command to get the `uartboot_da1470x.bin` firmware version is sent to the devices.

Step	Action	Opt.	Description
15	GPIO watchdog	Yes	If the GPIO watchdog option is enabled, then firmware will start the toggling after the `uartboot_da1470x.bin` is loaded and right before the production test download.
16	Download `st_da1470x.bin`	Yes	If any of the production tests is active (e.g., RF tests, XTAL trim, etc.) download the `st_da1470x.bin` to the devices.
17	Open the devices COM ports and get the `st_da1470x.bin` firmware version	Yes	After `st_da1470x.bin` has been downloaded, commands can be sent to it. First, the Windows DUTs COM ports are opened. Then, a command to get the `st_da1470x.bin` firmware version is sent to the devices. If there is a problem in the firmware or in the device, then this is the first failure to happen. The FW version get action will fail.
18	GPIO watchdog	Yes	If the GPIO watchdog option is enabled, the firmware will begin a periodic GPIO toggling during the whole production test procedure.
19	OTP timestamp	Yes	PLT will read the IC production timestamp and log it.
20	VBAT level measure	Yes	PLT will send a command to each DUT to measure VBAT for each one, using the internal ADC. VBAT level will be logged for debugging purposes.
21	External 32kHz	Yes	Check whether the external 32kHz crystal operates correctly.
22	XTAL trim	Yes	Perform the XTAL trim procedure if this is active.
23	UART resync	Yes	If the XTAL trim procedure was performed in the UART RX pin, then a special UART resync procedure takes place to resynchronize the device's UART RX path as it may have entered in a baud rate error state due to the 500ms received XTAL trim pulse.
24	Bluetooth® LE scan (TX test)	Yes	If the Scan DUT Advertise test is active, then perform a Bluetooth® LE scan test using HCI triggered advertisements. DUTs will start to advertise with a well-known BD address, one by one. GU will scan and return the RSSI for each DUT.
25	Bluetooth® LE tester TX power	Yes	If the Bluetooth® LE tester TX Power test is active, then perform the test using the external Bluetooth® LE tester instrument.
26	Bluetooth® LE tester TX carrier offset	Yes	If the Bluetooth® LE tester TX carrier offset test is active, then perform the test using the external Bluetooth® LE tester instrument.
27	Bluetooth® LE tester TX modulation index	Yes	If the Bluetooth® LE tester TX modulation index test is active, then perform the test using the external Bluetooth® LE tester instrument.
28	Bluetooth® LE tester RSSI	Yes	If the Bluetooth® LE tester RSSI test is active, then perform the test using the external Bluetooth® LE tester instrument.
29	GU RSSI test	Yes	If the RSSI test using the GU as transmitter is active, then perform the test.
30	GPIO/LED	Yes	Perform the GPIO/LED test, if the test is active.
31	GPIO connection test	Yes	Perform a GPIO continuity or voltage level test if the test is active.

Step	Action	Opt.	Description
32	Sensor test	Yes	Perform the sensor tests only if these are enabled.
33	Custom test	Yes	Perform any active custom test.
34	Current measure peripheral	Yes	Perform any active current measurement test for peripherals.
35	Current measure sleep	Yes	Perform the sleep current measurement.
36	Power cycle or reset DUTs	Yes	Power cycle with VBAT or Reset DUTs t enter boot procedure.
37	Open COM port and download `uartboot_da1470x.bin`	Yes	If any of the memory actions is active (e.g., QSPI burn, QSPI erase, etc.) download the `uartboot_da1470x.bin` to the devices.
38	Get `uartboot_da1470x.bin` version.	Yes	After `uartboot_da1470x.bin` has been downloaded, commands can be sent to it. A command to get the `uartboot_da1470x.bin` firmware version is sent to the devices.
39	GPIO watchdog	Yes	If the GPIO watchdog option is enabled, the firmware will begin a periodic GPIO toggling during the whole memory programming procedure.
40	QSPI/OSPI memory initialization	Yes	If any QSPI or OSPI operation is enabled, initialize the memory connected to the appropriate interface.
41	QSPI/OSPI erase	Yes	Erase the QSPI or OSPI, either the entire or part of it depending on the configuration.
42	QSPI/OSPI image write	Yes	If enabled, write the QSPI or OSPI with the customer image. If verify is enabled, the contents of the QSPI or OSPI will be read back and compared to the original image downloaded.
43	Custom memory data	Yes	Write custom memory data, taken from a barcode scanner, entered manually or through a CVS file.
44	OTP binary write	Yes	Write the OTP binary with the customer application area. If verify is enabled, the contents of the OTP memory will be read back and compared to the original image downloaded.
45	OTP CS write	Yes	If enabled, PLT will program the OTP configuration script area.
46	Memory read	Yes	Up to 10 memory read tests can be performed with up to 256 bytes in length.
47	Power cycle or reset DUTs	Yes	Power cycle with VBAT or Reset DUTs t enter boot procedure.
48	Scan test	Yes	If enabled, the GU will scan for device Bluetooth® LE advertisements. For the DUTs to be scanned a valid firmware must be burned into the OTP or QSPI flash that sends Bluetooth® LE advertisements after power up. Additionally, the BD address should be burned into the OTP or the QSPI by the PLT. The PLT expects to find devices in the air with the BD addresses programmed by the same tool, so it can match the BD addresses returned by the GU.

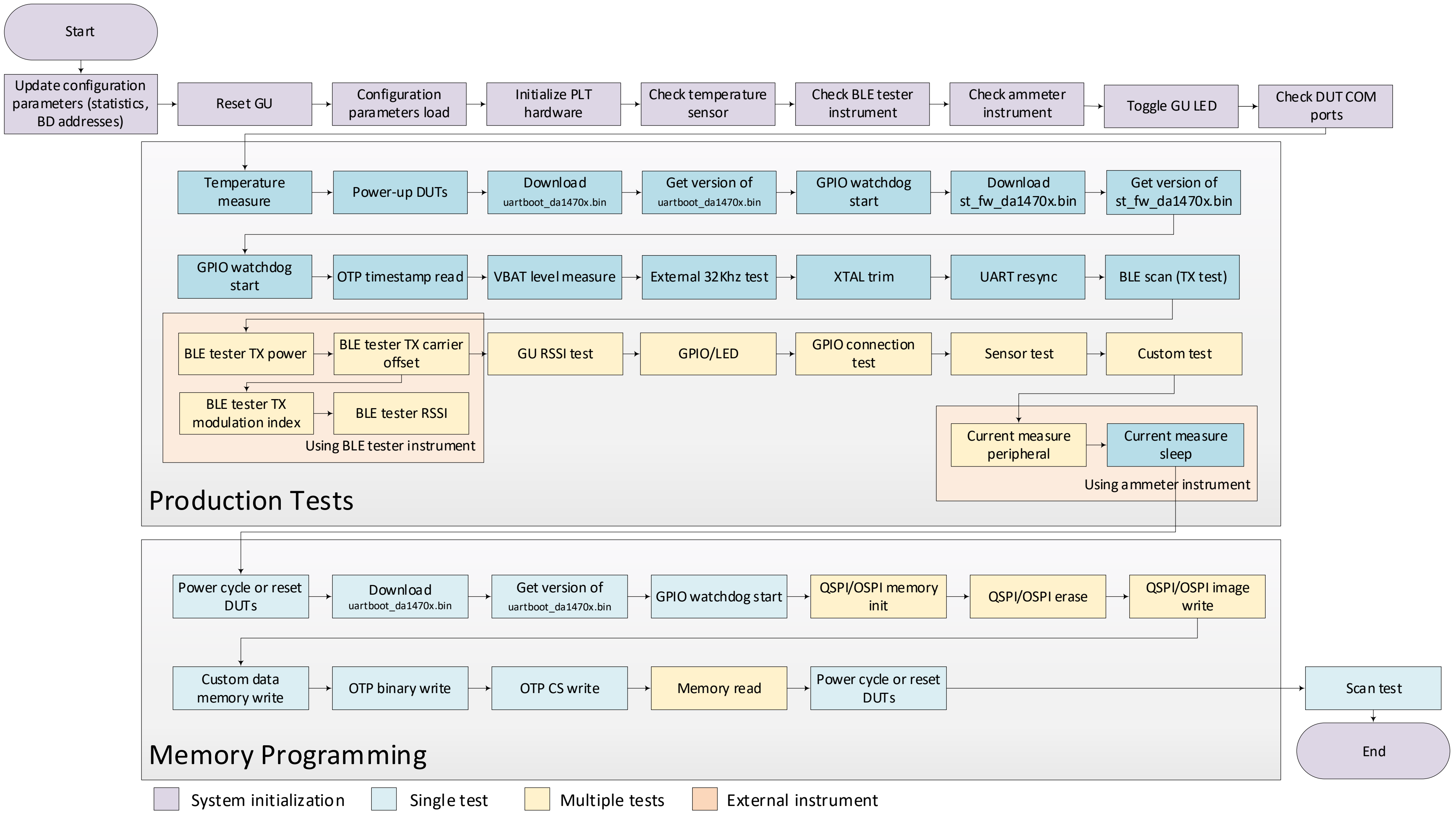


Figure 20: DA1470x Test Sequence

6.9 VBAT/Reset Signals Operation

The following section describes the PLT hardware VBAT and Reset signal operation during the DUT test sequence in Section 6.8.

There are two different modes available to power and reset the DUTs using a combination of the PLT VBAT and Reset lines. These are described next.

6.9.1 VBAT Only

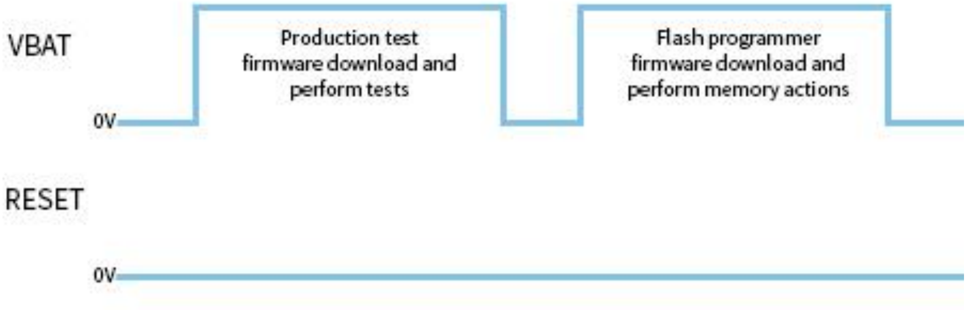

VBAT
Production test firmware download and perform tests
Flash programmer firmware download and perform memory actions
0V
RESET
0V

Figure 21: VBAT Only

In this mode only the VBAT line is used, as shown in Figure 21. Only the VBAT signal from the PLT hardware board to the DUT should be connected. The Reset signal is not driven. The DUTs are powered independently from their VBAT lines connected to the PLT HW and when reset is needed, the PLT software toggles the VBAT line low to perform a POR to each device.

Battery powered DUTs or DUTs with an external power supply are not supported in this mode. PLT to DUT VBAT line connection is mandatory. PLT Reset line connection is not required.

6.9.1.1 Firmware Download

When the firmware download procedure begins, the PLT VBAT line will power the DUTs and the UART connections will open. This will result to a POR for all active devices. The POR will activate the DUTs UART booting procedure and the PLT software will be able to download the test firmware.

If there are devices that failed the test firmware download procedure, PLT will perform a VBAT POR to retry the firmware download procedure only for those that failed. During the extra attempts to download firmware to the failed devices, the VBAT lines of the devices that succeeded will remain active. After maximum of three retry attempts, the PLT VBAT lines will remain active only for the devices that have succeeded. The retry operation and the amount of retries can be configured by the user. See Section 7.2.3.2 for more details.

When the production testing has finished the above procedure will be repeated for the memory programming, as a different firmware needs to be downloaded to the DUTs.

6.9.1.2 Current Measurement

Since the DUTs will be powered through the PLT HW using the VBAT line, the

Current Measurement Test (Table 46) for the DA1470x are supported as described in Section 5.7.

6.9.2 VBAT On with Reset

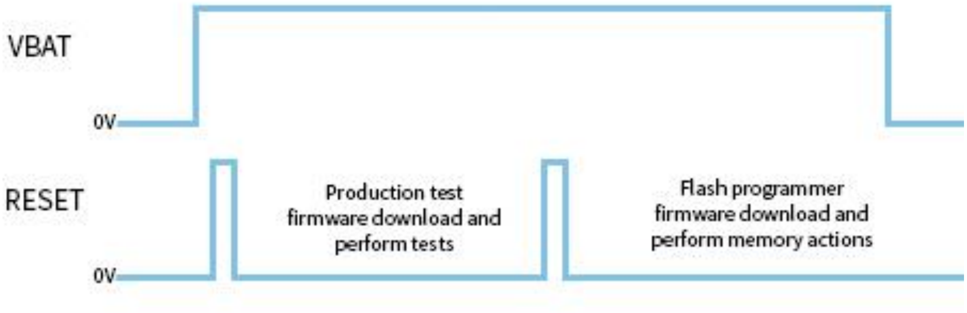

VBAT
0V
RESET
0V
Production test firmware download and perform tests
Flash programmer firmware download and perform memory actions

Figure 22: VBAT On with Reset

In this mode the reset of the DUTs will be performed by the PLT Reset line as shown in Figure 22. During this mode, the PLT VBAT line continuously provides power to the DUTs and the DUTs are reset using the PLT Reset line.

Power supply can be provided to the DUTs if the PLT VBAT line is connected to the DUTs. However, for battery powered DUTs or for DUTs with an external power supply VBAT should not be connected. For such devices, only the connection to the PLT Reset line is mandatory.

6.9.2.1 Firmware Download

When the firmware download procedure begins, PLT will reset the DUTs using the PLT Reset line. The VBAT line is already active and remains active for the entire PLT test and memory programming procedure. If there are devices that failed to download firmware, the PLT will reset all the DUTs again and retry to download firmware to all of them even if these have succeeded. This is different approach from the VBAT Only procedure (Section 6.9.1), since the Reset line is a single hardware line that cannot be differently controlled for each DUT, as opposed to the VBAT lines. The retry operation and the amount of retries can be configured by the user. Check 7.2.3.2 for more details.

When the production testing has finished, the above procedure will be repeated for the memory programming, as a different firmware needs to be downloaded to the DUTs.

6.9.2.2 Current Measurement

If the DUTs are powered through the PLT HW using the VBAT line, or if they are powered using a single common line from an external power supply, the Current Measurement Test (Table 46) for the DA1470x are supported as described in the Section 5.7. If the DUTs are powered independently or have their own power supply (e.g., have a battery) then the current measurement tests are not supported.

6.10 Custom Memory Data

The following section describes the PLT '`Custom Memory Data`' configuration and programming procedure.

The PLT supports programming custom user data of any size up to 256 bytes, to any memory and from any start address. Custom data can be entered to the PLT by the three input methods described in Table 15.

Table 15: Custom Memory Data Input Modes

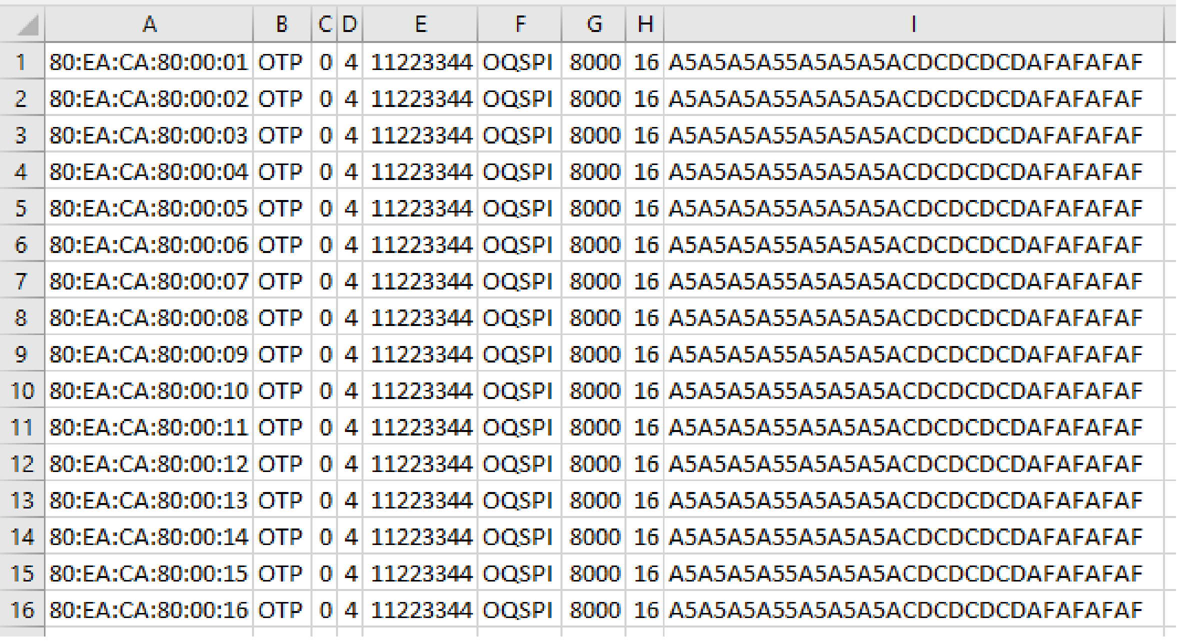
Input Modes	Description
CSV file	Users can provide a path to a CSV file that will contain the custom memory data for each DUT. The format of the CSV file is specific and is provided in Section 6.10.1.

Input Modes	Description
Manual	Users can manually edit the custom memory data prior of each PLT test run. The edit can be done in the PLT GUI or in the params.xml file using an external application or script. If different data per DUT is required, then the update of the custom memory data should be done before every PLT test run.

Section 7.2.8.1 explains in detail the various configuration parameters of the '`Custom Memory Data`' programming PLT feature.

6.10.1 Custom Data CSV File Format

This section describes the format of the CSV file used in *CSV file input mode* of the Custom Memory Data test (Section 6.10).



	A	B	C	D	E	F	G	H	I
1	80:EA:CA:80:00:01	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
2	80:EA:CA:80:00:02	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
3	80:EA:CA:80:00:03	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
4	80:EA:CA:80:00:04	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
5	80:EA:CA:80:00:05	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
6	80:EA:CA:80:00:06	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
7	80:EA:CA:80:00:07	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
8	80:EA:CA:80:00:08	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
9	80:EA:CA:80:00:09	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
10	80:EA:CA:80:00:10	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
11	80:EA:CA:80:00:11	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
12	80:EA:CA:80:00:12	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
13	80:EA:CA:80:00:13	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
14	80:EA:CA:80:00:14	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
15	80:EA:CA:80:00:15	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF
16	80:EA:CA:80:00:16	OTP	0	4	11223344	OQSPI	8000	16	A5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF

Figure 23: Custom Memory Data CSV File Example

Each line in the CSV file corresponds to a specific DUT, which is bound to a BD address. The BD address is written in the first column of the CSV file. After the DUT BD address, up to five memory operations can exist.

Each of these operations must have the following columns in the correct order as described below:

- Memory type (DA1470x can have **OTP** and **QSPI** or **OQSPI**).
- Start address.
- Size of data in bytes.
- Data to be written.

Figure 23 shows an example of a CSV file targeted DA1470X DUTs. In this example the CSV file contains information for DUTs with BD addresses `80:EA:CA:80:00:01` to `80:EA:CA:80:00:16`.

- The first operation is to write the OTP memory of the DA1470x DUTs four bytes, in OTP address 0x0000 (OTP user application area).
- The second operation is configured to write into the OSPI flash address 0x8000 sixteen bytes (0xA5A5A5A55A5A5A5ACDCDCDCDAFAFAFAF).

6.11 Golden Unit Scan Test

This section describes the PLT scan test procedure using the Golden Unit as scanner device.

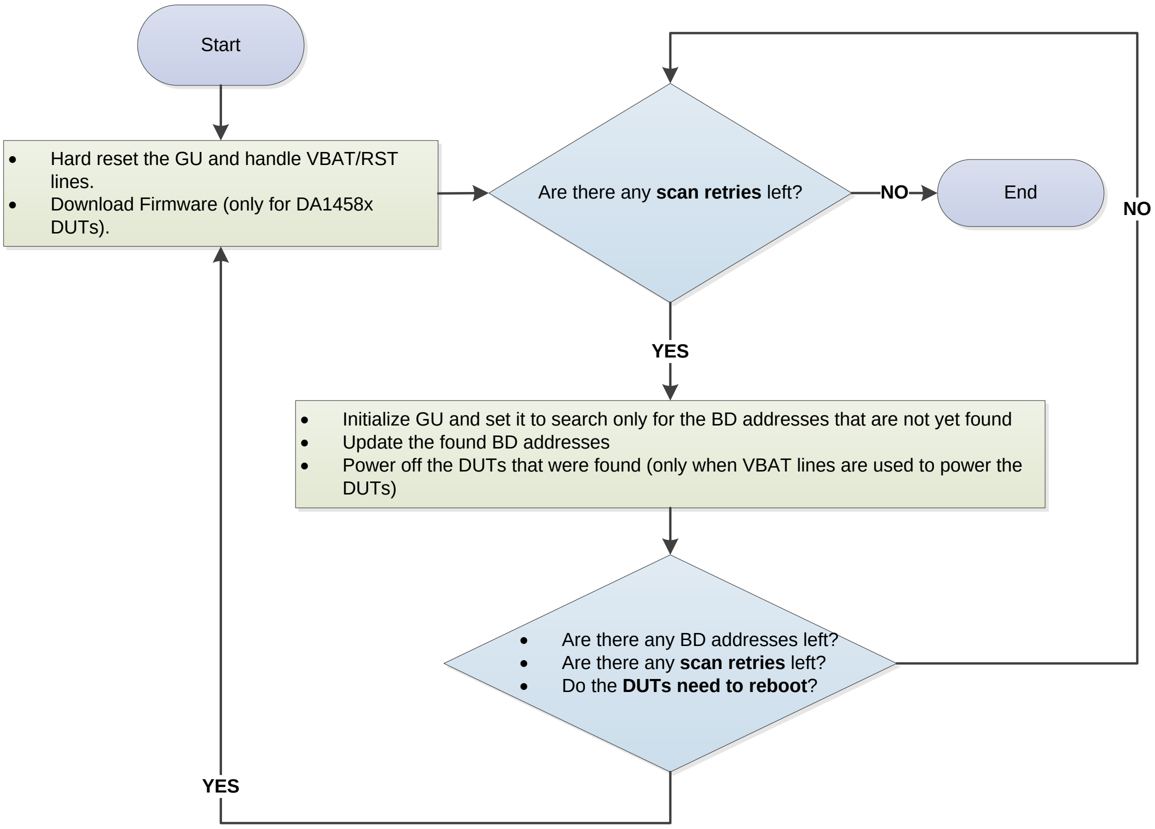
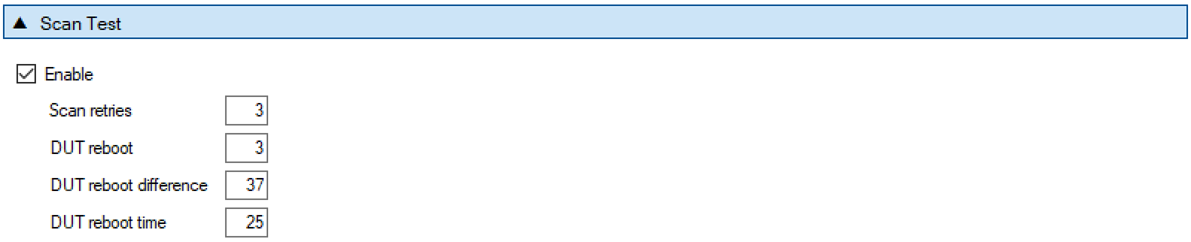

Start
Hard reset the GU and handle VBAT/RST lines.
Download Firmware (only for DA1458x DUTs).
Are there any scan retries left?
NO
End
NO
YES
Initialize GU and set it to search only for the BD addresses that are not yet found
Update the found BD addresses
Power off the DUTs that were found (only when VBAT lines are used to power the DUTs)
Are there any BD addresses left?
Are there any scan retries left?
Do the DUTs need to reboot?
YES

Figure 24: Golden Unit Scan Test

User can set various scan properties to adjust the Scan test procedure. The available properties are described in Table 50.

Figure 24 shows the scan sequence. First, the GU and the DUTs are reset. The DUTs must be burned with a valid application image that sends Bluetooth Low Energy advertisements after boot. Additionally, the PLT selected BD address should have been programmed to the OTP CS of the DUTs, so it is used in advertisements.

The GU will begin scanning for the BD addresses of all active DUTs. After each scan cycle, the already found BD addresses are removed from the search list of the GU and the appropriate DUTs will be powered off. This procedure will continue until the retries have reached the *Scan retries* set by the user. The PLT will reset the GU after a specific number or retries, given in *DUT reboot* option. Finally, the parameters *DUT reboot time* and *DUT reboot difference* set the DUT time needed to perform a POR with a small delay between the DUTs if needed.



▲ Scan Test

☑ Enable

Scan retries	3
DUT reboot	3
DUT reboot difference	37
DUT reboot time	25

Figure 25: Golden Unit Scan Test Example Parameters

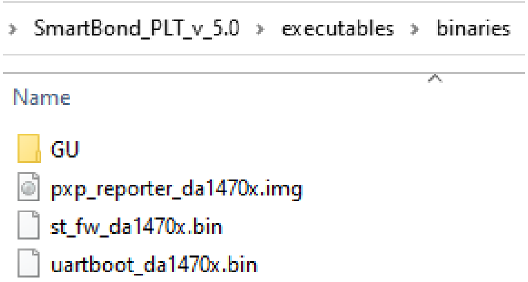
Figure 25 shows an example of DA1470x DUT scan test settings. For this example, the following steps will be executed:

- DUTs should have been programmed with a valid application image in the flash memory. The BD address selected by PLT should have been programmed in the OTP CS memory.
- Reset the GU in order to be in a clear state, power-off the DUTs and wait for 2000ms (DUT reboot time). Power-on DUTs.
- DUTs should start advertising with the already programmed BD address selected by PLT.
- Execute three GU scans. When each scan procedure is finished, power-off the found DUTs.
- Again, power-off the DUTs and wait for 2000ms (DUT reboot time) until power-on again.
- Continue with another two GU scans and after each scan procedure power off the found DUTs

6.12 Creating PLT Firmware Files

For the PLT to successfully operate, various firmware files are used based on the device type (GU or DUT), and the purpose of the firmware (different firmware for tests and memory programming).

All these firmware files are kept under the `binaries` folder in the PLT software package, as shown in Figure 26. To create these firmware files, the SDK packages should be downloaded from the customer portal and the source code patches located under the `fw_files` folder in the PLT software package shown in Figure 27, should be applied.



› SmartBond_PLT_v_5.0 › executables › binaries

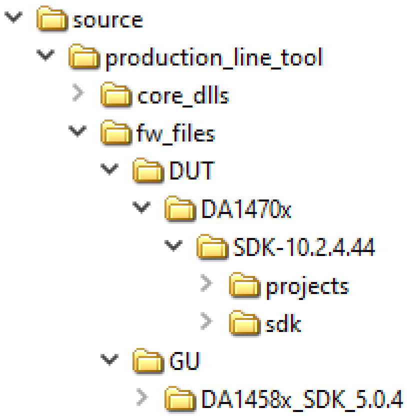
Name

GU
pxp_reporter_da1470x.img
st_fw_da1470x.bin
uartboot_da1470x.bin

Figure 26: Firmware Files

The source code patches maintain the folder structure of the SDK they are targeting, in order to apply the source code patch using a simple copy and replace. After patching, the projects contain all the necessary changes and the same firmware files can be built as those in the `binaries` folder of the PLT software package.

The '`fw_files`' folder has two main categories. Firmware targeted for the GU and for the DUTs. Under each category there is a folder indicating the IC target and the SDK used.



```
source
  production_line_tool
    core_dlls
    fw_files
      DUT
        DA1470x
          SDK-10.2.4.44
            projects
            sdk
      GU
        DA1458x_SDK_5.0.4
```

Figure 27: Folder Contents of 'fw_files'

Applying a source code patch for each one of the binaries is described below.

6.12.1 Golden Unit Firmware

The Golden Unit is a DA14580 device. A modified version of the `prod_test_580.bin` firmware is used.

This patch contains all the changes needed to re-create the following firmware.

- `prod_test_GU.bin`

To re-create the exact source code of the `prod_test_GU.bin` firmware:

1. Use a clean copy of the `DA1458x_SDK_5.0.4` SDK from the customer portal.
2. Copy the contents of the '…`\fw_files\GU\DA1458x_SDK_5.0.4\DA1458x_SDK\5.0.4\`' folder to the default SDK.
3. The Keil v5 project file of the `prod_test_GU.bin` is the '`prod_test.uvprojx`' under the folder '`\5.0.4\projects\target_apps\prod_test\prod_test\Keil_5\`'.

6.12.2 DA1470x Firmware

This patch contains all the changes needed to re-create the following firmware:

- `uartboot_da1470x.bin`
- `st_fw_da1470x.bin`

To re-create the exact source code of the above firmware:

1. Use a clean copy of the `SDK_10.2.4.44` from the customer portal.
2. Copy the contents from '`fw_files\DUT\DA1470x\SDK_10.2.4.44`' folder to the `SDK_10.2.4.44` SDK.
3. The SmartSnippets™ Studio project file of the `st_fw_da1470x.bin` is the '`st_fw`' project under '`SDK_10.2.4.44\projects\dk_apps\reference_designs\st_fw`' folder. To create each binary, select from the drop-down menu the "`Release RAM`" option for each chip.
4. The SmartSnippets™ Studio project file of the `uartboot_da1470x.bin` is the '`uartboot`' project under the folder '`SDK_10.2.4.44\sdk\bsp\system\loaders\uartboot`'. To create the binary, select from the drop-down menu the "`Release`" option.
5. The Smart Snippets Studio project file of the `pxp_reporter_da1470x.img` is the '`pxp_reporter`' under the folder '`SDK_10.2.4.44\projects\dk_apps\demos\pxp_reporter`'. To create each binary, select from the drop-down menu the '`QSPI_Release`' option for each chip.

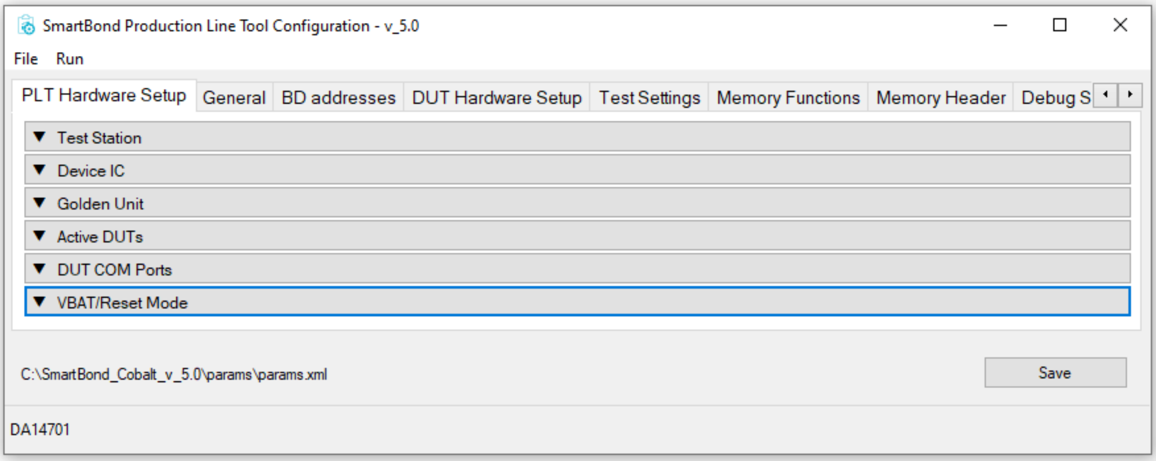
Each binary will be created under the project folder in a folder having the same name as the selected option.

7 Applications

7.1 Introduction

The PLT software includes four different applications (Table 5). The CFG PLT is used to setup the system according to the device hardware options and select the required tests and memory actions to be performed. The GU Upgrade is used to update the Golden Unit firmware. The GUI and the CLI PLT applications are used to perform the tests, monitor their progress in real-time and view the test results.

7.2 CFG PLT Application


SmartBond Production Line Tool Configuration - v_5.0

C:\SmartBond_Cobalt_v_5.0\params\params.xml

DA14701

Note 1 Each field can be minimized by clicking on it, but it will not be disabled. The tests will run if they are enabled, even when the test field is minimized and not shown.

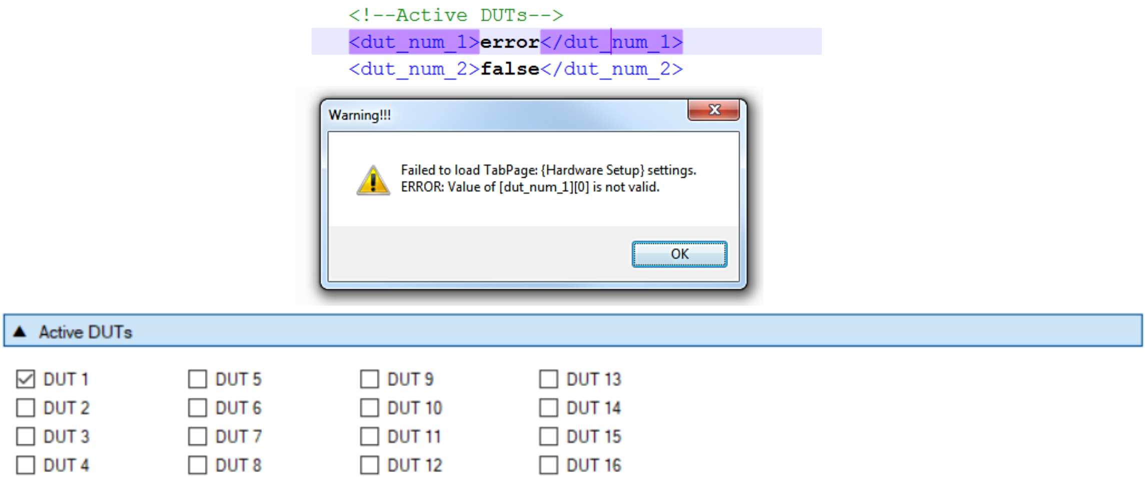
Figure 28: CFG PLT Startup Screen

The CFG PLT application (`SmartBond_PLT_CFG.exe`) is a GUI application tool, which is mainly used to appropriately configure the tests and memory operations the tool will perform. Depending on the selected device chipset and the enabled actions, only appropriate options are enabled and shown. Any change made by the user is validated before being saved to the XML file, with the use of a schema XSD file. This prevents erroneous values to be stored in the XML file that would harm the production procedure.

Figure 28 shows the initial CFG PLT screen. The `Main Menu` options are described in Table 16 and the bottom strip information is described in Table 17. The application begins with the `Hardware Setup` tab (see Section 7.2.2). Users can navigate to the other PLT configurable options by selecting the different tabs.

When a tab is selected, the settings of this tab are reloaded from the XML file. If there is an error in the configuration XML file, a warning message will be shown indicating which of the parameters has error. Additionally, the related graphic entry in the CFG application for the erroneous configuration parameter will be highlighted in red.

An example is given in Figure 29. Configuration parameter `dut_num_1` has wrong value (`error` instead of either `false` or `true`) in the `params.xml` file. When the `Hardware Setup` CFG tab is selected the warning message will be displayed. If `OK` is pressed, the `Hardware Setup` tab will be loaded with the `DUT 1` checkbox in red. The displayed value will be the default value taken from the XML schema document (`params.xsd`).



```
<!--Active DUTs-->
<dut_num_1>error</dut_num_1>
<dut_num_2>false</dut_num_2>
```

Warning!!!

Failed to load TabPage: {Hardware Setup} settings.
ERROR: Value of [dut_num_1][0] is not valid.

▲ Active DUTs

☑ DUT 1 □ DUT 5 □ DUT 9 □ DUT 13
□ DUT 2 □ DUT 6 □ DUT 10 □ DUT 14
□ DUT 3 □ DUT 7 □ DUT 11 □ DUT 15
□ DUT 4 □ DUT 8 □ DUT 12 □ DUT 16

Figure 29: CFG PLT with Erroneous Configuration Parameter

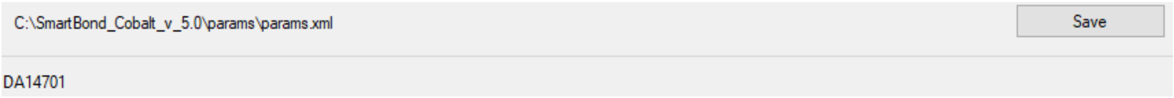
When the user makes a change, the `Save` button will become `Save*` to indicate that a save is required.

In case of a configuration parameter error, pressing `Save` will save the default parameter value, overwriting the erroneous value.

Table 16: CFG PLT Main Menu Options

Region	Option	Description
File	Open XML file	Opens a new XML file and loads the settings. The full path of the new XML file is shown at the bottom end of the screen.
	View XML file	Opens the XML file in notepad.
	Save as…	Exports all settings to a new XML file. The full path of the new XML file is shown at the bottom end of the screen.
	Reset to defaults	Overwrites all parameters options in the XML file with their default values taken from the XSD file.
	Exit	Exits the CFG PLT application.
Run	Run GUI PLT	Opens the GUI PLT application.
	Run CLI PLT	Opens the CLI PLT application.

Table 17: CFG PLT Bottom Strip Options

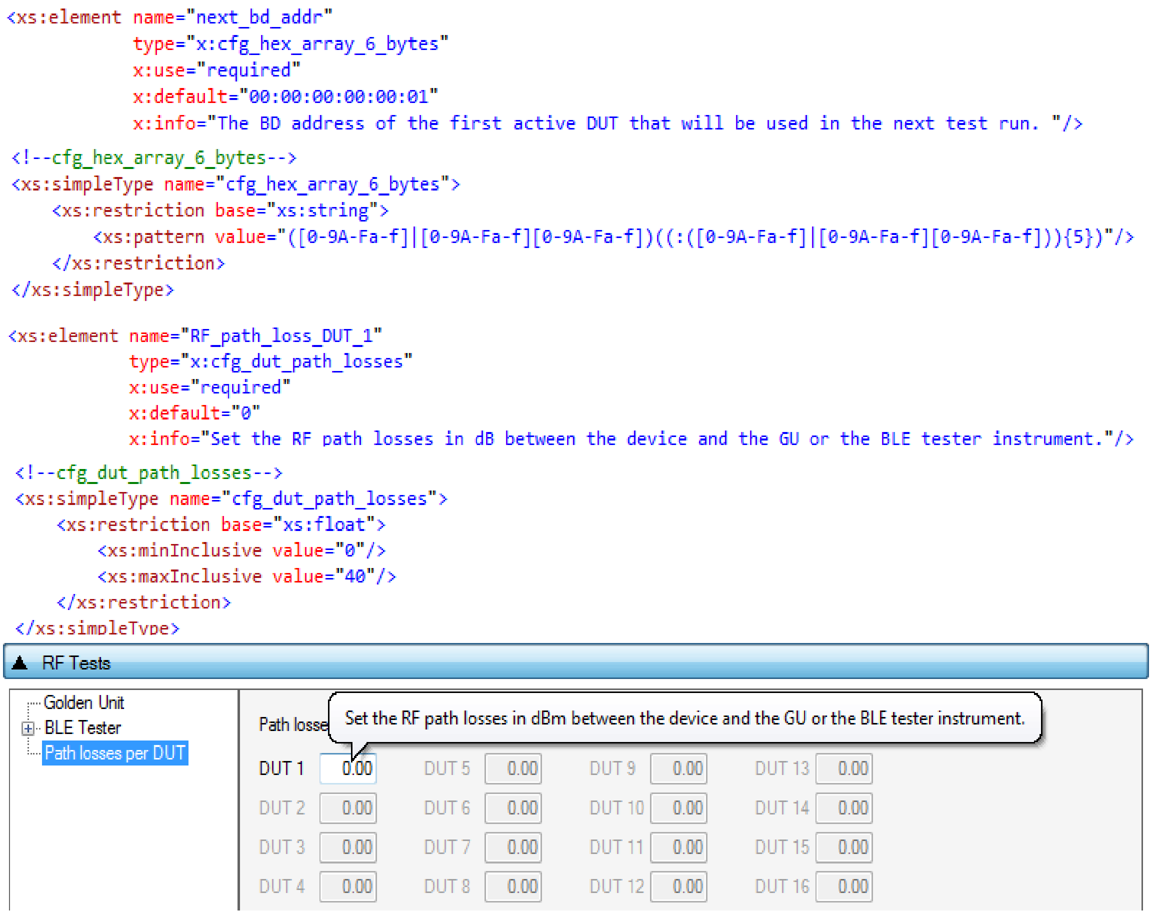
Option	Description
 C:\SmartBond_Cobalt_v_5.0\params\params.xml Save DA14701	
C:\SmartBond_Cobalt_v_5.0\params\params.xml	Shows the full path of the XML file currently used.
DA14701	Shows the selected device IC.
Save	Saves the options of the selected tab. E.g. If General settings tab is selected, then only the settings for this tab will be saved. **Note:** A shortcut for this button is the `Ctrl+S` key combination.

7.2.1 XML and XSD Files

The CFG PLT application is a front-end user interface for the cfg_dll.dll library (Figure 18). The cfg_dll.dll library, explained in detail in Ref. [1], is an XML parser, editor, and parameter validator. It has an easy-to-use API for reading and manipulating the params.xml file. File params.xsd is the XML schema used for parameter validation.

In the CFG PLT application, all user selectable options are loaded and saved inside the XML file, by effectively using the cfg_dll.dll library API. The XSD schema file params.xsd is not edited in any way but only read by the cfg_dll.dll library API, whenever a parameter validation is needed.

The XSD schema file, params.xsd, holds information about the overall structure of the params.xml file, the default, and valid values a parameter can take and useful help information about the purpose of each parameter. An example part of the XSD file is given in Figure 30.



```
<xs:element name="next_bd_addr"
            type="x:cfg_hex_array_6_bytes"
            x:use="required"
            x:default="00:00:00:00:00:01"
            x:info="The BD address of the first active DUT that will be used in the next test run. "/>
 <!--cfg_hex_array_6_bytes-->
 <xs:simpleType name="cfg_hex_array_6_bytes">
     <xs:restriction base="xs:string">
         <xs:pattern value="([0-9A-Fa-f]|[0-9A-Fa-f][0-9A-Fa-f])((:([0-9A-Fa-f]|[0-9A-Fa-f][0-9A-Fa-f])){5})"/>
     </xs:restriction>
 </xs:simpleType>

<xs:element name="RF_path_loss_DUT_1"
            type="x:cfg_dut_path_losses"
            x:use="required"
            x:default="0"
            x:info="Set the RF path losses in dB between the device and the GU or the BLE tester instrument."/>
 <!--cfg_dut_path_losses-->
 <xs:simpleType name="cfg_dut_path_losses">
     <xs:restriction base="xs:float">
         <xs:minInclusive value="0"/>
         <xs:maxInclusive value="40"/>
     </xs:restriction>
 </xs:simpleType>
```

Set the RF path losses in dBm between the device and the GU or the BLE tester instrument.

DUT 1	0.00	DUT 5	0.00	DUT 9	0.00	DUT 13	0.00
DUT 2	0.00	DUT 6	0.00	DUT 10	0.00	DUT 14	0.00
DUT 3	0.00	DUT 7	0.00	DUT 11	0.00	DUT 15	0.00
DUT 4	0.00	DUT 8	0.00	DUT 12	0.00	DUT 16	0.00

Figure 30: XSD Schema File Example

Element next_bd_addr holds the Next BD address, as described in Section 7.2.4.1 and Table 27. It has a default value of x:default="00:00:00:00:00:01". This default value will be returned by the cfg_dll.dll API if the XML file has an error entry in the equivalent next_bd_addr element, since the validation of the parameter will fail.

The x:info="The BD address …" value will be loaded by the cfg_dll.dll API and be used in the CFG PLT tooltips. The type="x:cfg:hex_array_6_bytes" defines the parameter type. This is the actual XSD entry that is used for the parameter validation. The cfg:hex_array_6_bytes type is defined later in the file and has a rather complicated pattern defined with <xs:pattern value

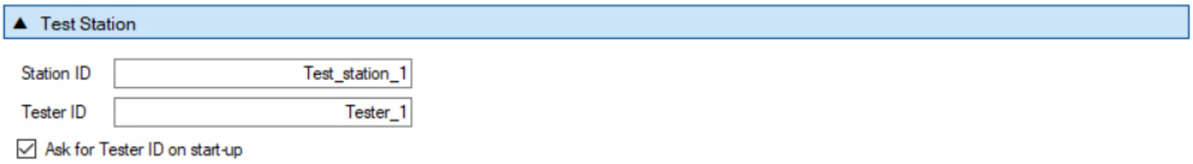
`="([0-9A-Fa-f]…"/>`. If the `next_bd_addr` element in the XML file has a value that does not match this pattern, the validation of the parameter will fail and the `cfg_dll.dll` API will return the default value (00:00:00:00:01). In the CFG PLT, the default value will be shown in red, indicating that an error exists in the `params.xml` file for this parameter. It will not change the erroneous value in the `params.xml` file until the user presses the **Save** button, in which case the default value will overwrite the erroneous value.

In the second example of Figure 30, the `RF_path_loss_DUT_1` XSD element is shown. This element is used in the Path Losses per DUT as shown in Figure 55. This element has a default value of 0 and the allowed values are floats, between `<xs:minInclusive value="0"/>` and `<xs:maxInclusive value="40"/>`, as shown in the `cfg_dut_path_losses` type description. The `x:info="Set the RF path .."/>` will be loaded by the `cfg_dll.dll` API and used in the CFG PLT tooltips as shown in the bottom part of Figure 30.

7.2.2 Hardware Setup

This section describes the Hardware Setup settings available for the PLT hardware board, shown in Figure 28.

7.2.2.1 Test Station



▲ Test Station

Station ID: Test_station_1

Tester ID: Tester_1

☑ Ask for Tester ID on start-up

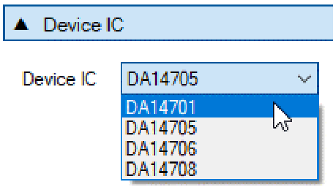
Figure 31: Station Identification

These fields hold the station ID and tester ID names to distinguish between different test stations and users. The values of these fields are written into the DUT logs and CSV files. Table 18 describes the available options for the *Station Identification*.

Table 18: Station Identification

Option	Description
Station ID	The name of the PLT test station.
Tester ID	The PLT tester ID name.
Ask for Tester ID on start-up	When `SmartBond_PLT_GUI.exe` starts it will ask for the tester ID name.

7.2.2.2 Device IC



▲ Device IC

Device IC: DA14705

DA14701
DA14705
DA14706
DA14708

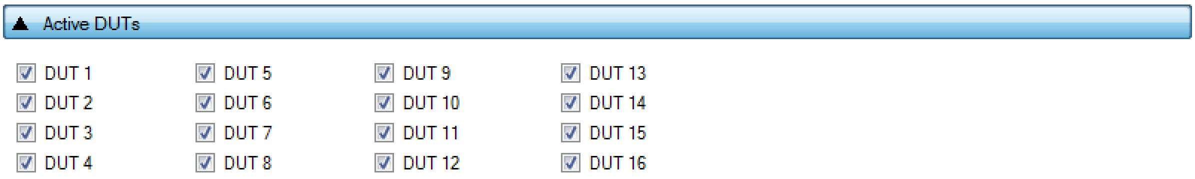
Figure 32: Device IC

Users can select the device IC type. This option may change any IC related configuration parameters and their equivalent graphics, such as selectable tabs and tests. Table 19 describes the available options for the *Device IC*.

Table 19: Device IC

Option	Description
Device IC	The Renesas Bluetooth® LE chipset used in the device under test.

7.2.2.3 Active DUTs



▲ Active DUTs

☑ DUT 1
☑ DUT 2
☑ DUT 3
☑ DUT 4
☑ DUT 5
☑ DUT 6
☑ DUT 7
☑ DUT 8
☑ DUT 9
☑ DUT 10
☑ DUT 11
☑ DUT 12
☑ DUT 13
☑ DUT 14
☑ DUT 15
☑ DUT 16

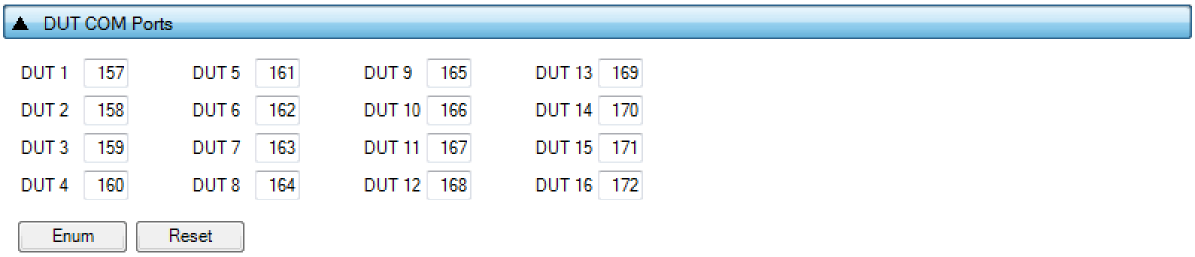
Figure 33: Active DUTs

Enables or disables the testing for each DUT. Table 20 describes the available options for the *Active DUT*.

Table 20: Active DUTs

Option	Description
DUT1-16	Enables the specific DUT device placed on connector DUT1-DUT16.

7.2.2.4 DUT COM Ports



▲ DUT COM Ports

DUT 1 157
DUT 2 158
DUT 3 159
DUT 4 160
DUT 5 161
DUT 6 162
DUT 7 163
DUT 8 164
DUT 9 165
DUT 10 166
DUT 11 167
DUT 12 168
DUT 13 169
DUT 14 170
DUT 15 171
DUT 16 172

Figure 34: DUT COM Ports

This field shows the Windows COM port assigned to each DUT. The table is filled only when the '`COM Enum`' action has been performed by the CFG or the GUI PLT applications, or when non-zero entries exist in the `com_port_x` params.xml options. When the '`COM Enum`' action is performed, the tools will automatically find the DUT COM ports and save them in the `params.xml` file. These values will be read by the CFG PLT application and be displayed here. When a '`COM Enum`' action has not been performed, the GUI PLT will automatically run it once in every first test execution.

Note: Great care must be taken when the `params.xml` file is shared across different test stations, where different DUT COM Ports will probably exist. The '`COM Enum`' action should then be performed again, so the new COM ports of the new PC system are identified and updated in the XML file.

Table 21 describes the available options for the *DUT COM Ports*.

Table 21: DUT COM Ports

Option	Description
DUT1-16	Shows the Widows COM port assigned to a specific DUT.
Reset	Sets all values to zero.

Option	Description
Enum	Executes the COM port enumeration procedure. The found COM ports are shown before being saved.

7.2.2.5 Golden Unit

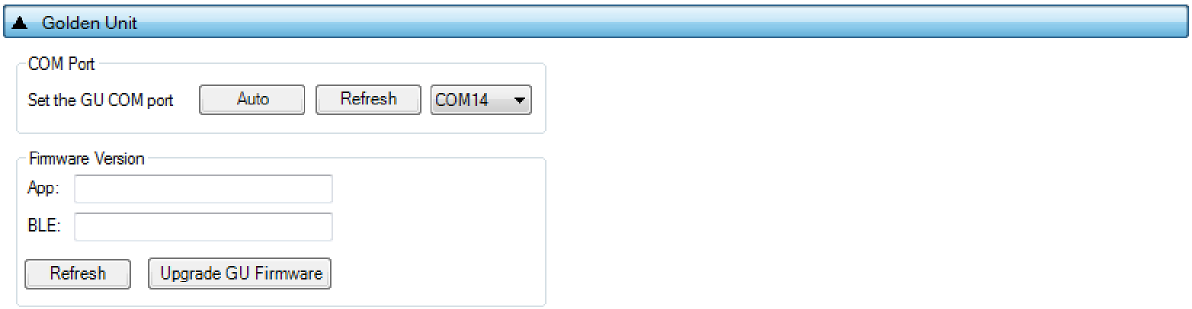


Figure 35: Golden Unit COM Port

This field holds the Golden Unit COM port. Manual or automatic COM port find can be selected.

The Golden Unit COM port can be manually selected from the list with all the available COM ports existing in the system. Additionally, it can automatically be found by pressing the `Auto` button. The automatic procedure searches the serial number of all system COM ports to find the “`DialogSemi`” string. Details on how to program the serial number in the GU FTDI can be found in Appendix H.

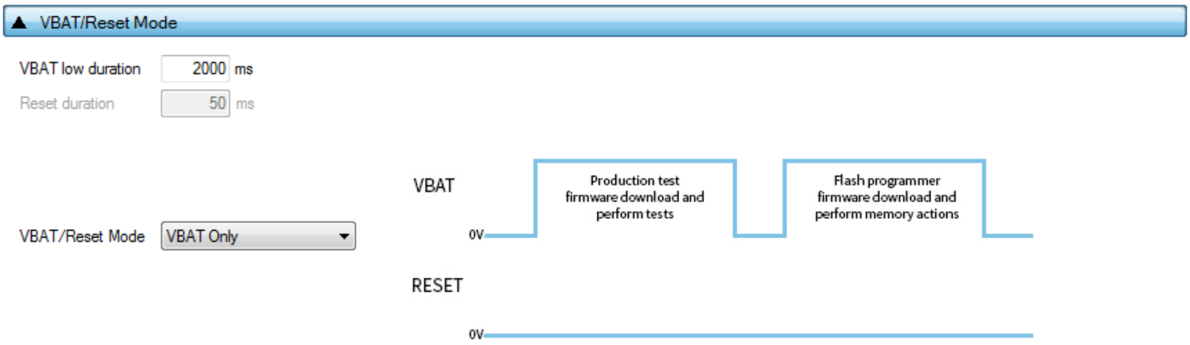
Table 22: Set the GU COM Port

Option	Description
Auto	Initiates the automatic Golden Unit COM port find procedure.
Refresh	Refreshes the dropdown menu with all the available system COM ports.
Dropdown Menu	Manually select the Golden Unit COM port from all the available system COM ports.

Table 23: Golden Unit Firmware Version Upgrade

Option	Description
Refresh	Retrieves the current Bluetooth® LE and application versions of the connected Golden Unit.
Upgrade GU Firmware	Opens the Section 7.5 application, which is used to update the GU firmware.

7.2.2.6 VBAT/Reset Mode



▲ VBAT/Reset Mode
VBAT
Production test firmware download and perform tests
Flash programmer firmware download and perform memory actions
0V
RESET
0V

VBAT low duration 2000 ms

Reset duration 50 ms

VBAT/Reset Mode VBAT Only

Figure 36: VBAT/Reset Mode Selection

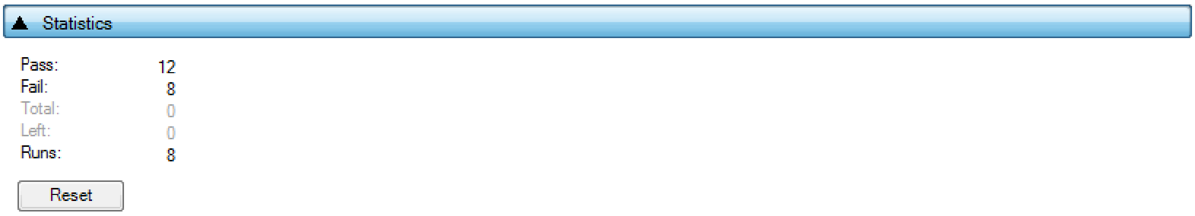
This field holds the VBAT/Reset mode selections. This option sets the PLT VBAT and PLT Reset line modes for the DUT power supply and reset during the PLT test sequence. Table 24 describes the available selections.

Table 24: VBAT/Reset Mode

Option	Description
VBAT/Reset Mode	Select the operation for VBAT/Reset signals. Available options are: ● VBAT Only ● VBAT On with Reset Section 6.9 describes each mode in detail. Default setting is *VBAT only*.

7.2.3 General

7.2.3.1 Statistics



Pass:	12
Fail:	8
Total:	0
Left:	0
Runs:	8

Figure 37: Statistics

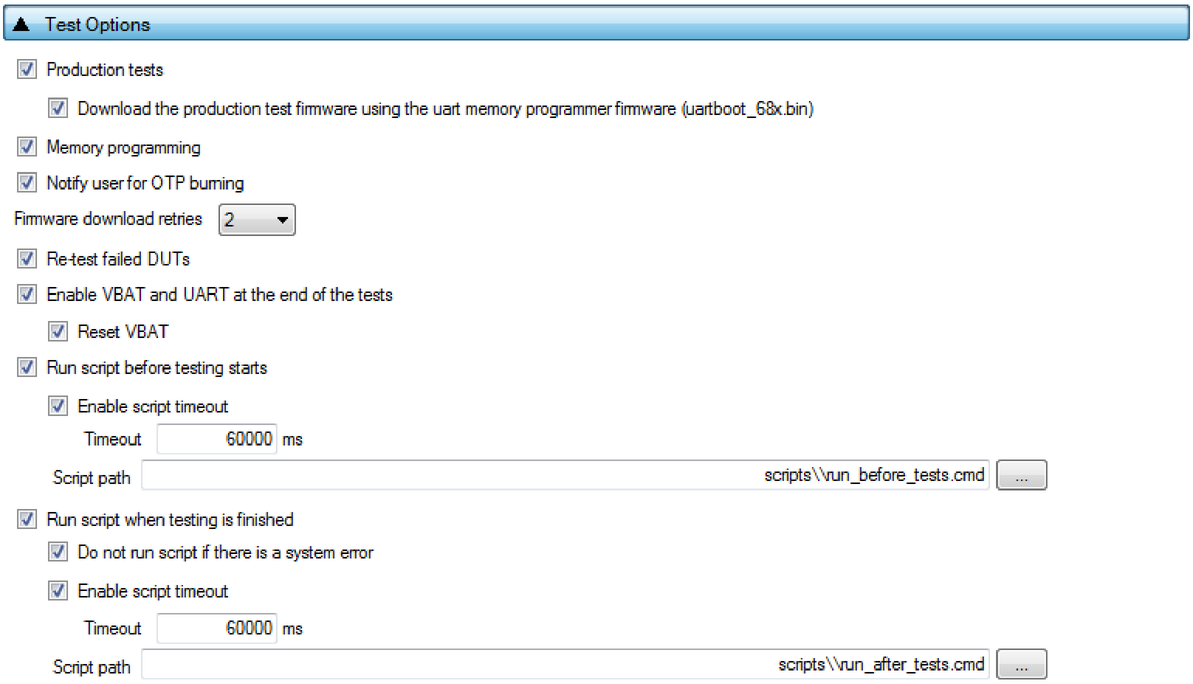
This field holds the test result statistics. Table 25 describes the *Statistics* field.

Table 25: Statistics

Option	Description
Pass	Shows the number of DUTs that have successfully passed all the tests.
Fail	Shows the number of DUTs that have failed the tests.
Total	Shows the number of DUTs that will be tested. This option is available only when *Range* mode is enabled in Section 7.2.4.1.
Left	Shows how many DUTs are still to be tested. This option is available only when *Range* mode is enabled in Section 7.2.4.1.
Runs	Shows the number of test-runs the PLT has performed.

Option	Description
Reset	Pressing the `Reset` button clears all statistics values to their defaults. Values *Pass*, *Fail* and *Runs* will be set to zero. If *Range* mode is enabled in Section 7.2.4.1, the *Total* and *Left* values will be set as the difference of *Next* and *End BD address*, otherwise will be set to zero.

7.2.3.2 Test Options



▲ Test Options

☑ Production tests

☑ Download the production test firmware using the uart memory programmer firmware (uartboot_68x.bin)

☑ Memory programming

☑ Notify user for OTP burning

Firmware download retries 2

☑ Re-test failed DUTs

☑ Enable VBAT and UART at the end of the tests

☑ Reset VBAT

☑ Run script before testing starts

☑ Enable script timeout

Timeout 60000 ms

Script path scripts\\run_before_tests.cmd ...

☑ Run script when testing is finished

☑ Do not run script if there is a system error

☑ Enable script timeout

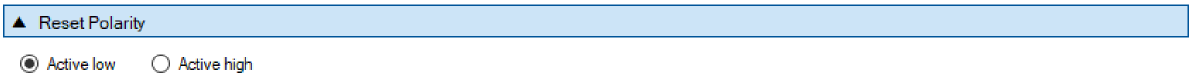
Timeout 60000 ms

Script path scripts\\run_after_tests.cmd ...

Figure 38: Test Options

This field holds generic PLT test procedure options. The PLT procedure is split into two main parts: *Production tests* and *Memory programming*.

Production tests include all the tests under in Section 7.2.5.2.



▲ Reset Polarity

● Active low ○ Active high

Figure 39: Reset Polarity

The Reset Polarity selected here is used in combination to the reset method. If the reset method is set to VBAT only then this setting is ignored. Depending on the implementation, active high or active low reset signal might be required. This setting controls the reset polarity as active high or active low.

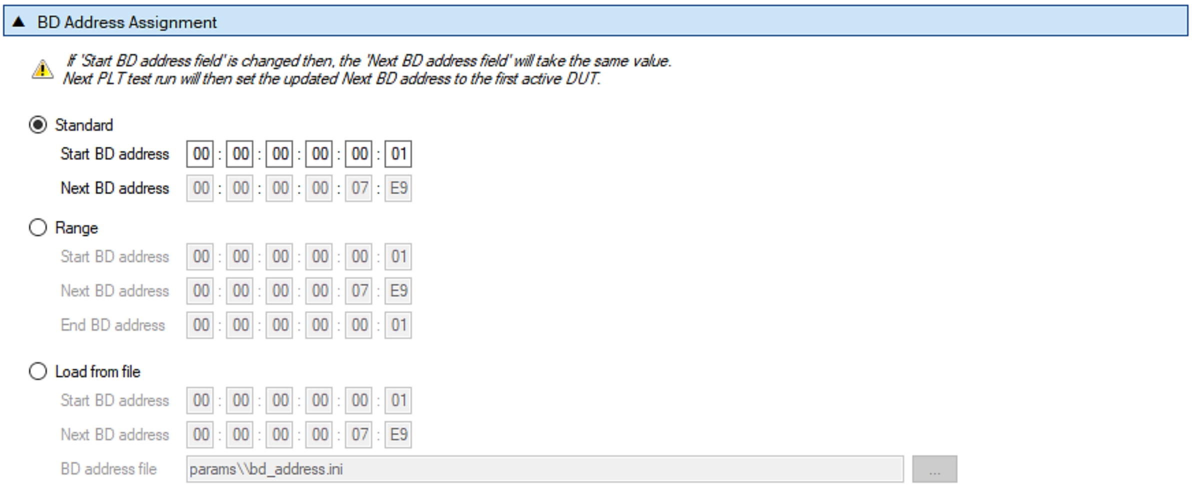
Test Settings. Memory Programming includes all the tests in Sections 7.2.7and 7.2.8.

Table 26: Test Options

Option	Description
Production tests	This option enables the production test operations.
Memory programming	This option enables the memory programming operations.
Notify user for OTP burning	When this option is enabled, PLT informs the user with all the OTP burning tests that are enabled. A pop-up message appears, prompting the user whether to proceed with the tests.
Firmware download retries	Configures the firmware download retries in case of an error during firmware download.
Re-test failed DUTs	When this option is enabled, any DUT that failed will immediately be retested with the exact same options, including the *BD address*. This option is the same to the *Retest failed DUTs - Enable* in Section 7.3.1.
Enable VBAT and UART at the end of the tests	Enables the VBAT lines and UART communication between the PC and the devices after all the tests have finished. If enabled, DUTs will remain powered after the end of the tests.
Reset VBAT	If this option is enabled the VBAT line will be toggled. If not selected, the DUTs will not be rebooted, thus their system RAM will have the last test firmware downloaded by the PLT (either the `st_fw_da1470x.bin` or the `uartboot_da1470x.bin`).
Run script before testing starts	This option enables the execution of a batch or an executable before the device testing procedure starts. As described in Section 7.3.2, the success return code should be a value between 0 and 100 for the tool not to report an error. Any other value will be taken as error and prevent the tool from running the tests.
Enable script timeout	Enables a wait timeout for the script to finish. The time to wait is set in the `timeout` field below. If this option is disabled PLT will wait until the script ends.
Timeout	The time to wait for the script to finish.
Script path	The path of file to execute when the *Run script before testing starts* option is enabled.
Run script when testing is finished	This option enables the execution of a batch or an executable after the device testing procedure has finished. The success return code should be 0 for the tool not to report an error.
Enable script timeout	Enables a wait timeout for the script to finish. The time to wait is set in the `timeout` field below. If this option is disabled, PLT will wait until the script ends.
Timeout	The time to wait for the script to finish.
Script path	The path of file to execute when the *Run script when testing is finished* option is enabled.

7.2.4 BD Addresses

7.2.4.1 BD Address Assignment



▲ BD Address Assignment

⚠ If 'Start BD address field' is changed then, the 'Next BD address field' will take the same value.
Next PLT test run will then set the updated Next BD address to the first active DUT.

● Standard

Start BD address 00 : 00 : 00 : 00 : 00 : 01

Next BD address 00 : 00 : 00 : 00 : 07 : E9

○ Range

Start BD address 00 : 00 : 00 : 00 : 00 : 01

Next BD address 00 : 00 : 00 : 00 : 07 : E9

End BD address 00 : 00 : 00 : 00 : 00 : 01

○ Load from file

Start BD address 00 : 00 : 00 : 00 : 00 : 01

Next BD address 00 : 00 : 00 : 00 : 07 : E9

BD address file params\\bd_address.ini ...

Figure 40: BD Address Assignment

The *BD Address Assignment* field defines different ways the PLT can handle the device BD address. The available modes are *Standard*, *Range*, *Load from file* and *Scan* mode.

The *Standard*, *Range*, and *Load from file* modes are similar. These have a *Start BD address,* which is the initial address that the PLT session begins. The *Next BD address* field holds the BD address that will be used on the next PLT run, so the BD address assignment can be continued even after the GUI PLT is closed. For that reason, the user cannot alter the *Next BD address*. The *Next BD address* initial value is the same as the *Start BD address* when the PLT session begins.

Note: In CFG PLT only the Start BD address is given. The assignment of the actual device BD addresses occurs in the GUI PLT at the beginning of each test run.

Note: The only invalid BD address is `00:00:00:00:00:00`.

Standard Mode

Table 27 describes the available options for the *Standard* mode. In this mode, the first active DUT takes the *Next BD address*. This BD address is incremented by one and assigned to the next active DUT until all active DUTs have a BD address assigned to them.

This assignment mode never runs out of BD addresses and it will continue assigning addresses until the *Next BD address* reaches `FF:FF:FF:FF:FF:FF`.

Table 27: BD Address Assignment - Standard Mode

Option	Description
Start BD address	The BD address that the PLT session has started with.
Next BD address	The BD address that will be used in the first active DUT of the next PLT run.

Range Mode

Table 28 describes the available options for the *Range* mode. This mode is the same as *Standard* mode except for the additional *End BD address*.

Sine a ‘Start BD address’ and an ‘End BD address’ exist, the total amount of devices to be tested can be calculated. Therefore, this mode enables the *Total* and *Left* fields in Section 7.2.3.1, where *Total* is the number of the BD addresses to be used from *Start BD address* to *End BD address* and *Left* is the number of BD addresses remaining.

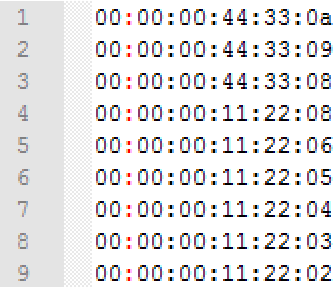
Note: The *End BD address* must always be greater than the *Start BD address*. In addition, when *Left* BD addresses are not enough for the remaining active DUTs, the PLT will not run.

Table 28: BD Address Assignment Options - Range Mode

Option	Description
Start BD address	The BD address that the PLT session has started with.
Next BD address	The BD address that will be used in the first active DUT of the next PLT run.
End BD address	The BD address that the PLT session will end with.

Load from File Mode

Table 29 describes the available options for the *Load from file* mode. In this mode, the *Start BD address* and the *Next BD address* have the same roles as before. The difference in this mode is that the BD addresses are loaded from a file in the order as they are written in that file, not using the automatic incremental method of the previous modes. In every test run, the PLT will search for the first occurrence of the *Next BD address* in the file and will load it along with the BD addresses that follow, until all active DUTs have a BD address.



```
00:00:00:44:33:0a
00:00:00:44:33:09
00:00:00:44:33:08
00:00:00:11:22:08
00:00:00:11:22:06
00:00:00:11:22:05
00:00:00:11:22:04
00:00:00:11:22:03
00:00:00:11:22:02
```

Figure 41: Example for Load from File Mode

For example, consider three active DUTs: DUT3, DUT6, and DUT 9 and the *Next BD address* to be `00:00:00:11:22:08`. Figure 41 shows the beginning of the BD address file used in this example. The PLT will search for the *Next BD address* in the file and load it to the first active DUT, DUT3. It will then continue with `00:00:00:11:22:06` for DUT6 and `00:00:00:11:22:05` for DUT9. It will also return `00:00:00:11:22:04` as the *Next BD address* to be used at the next PLT test run.

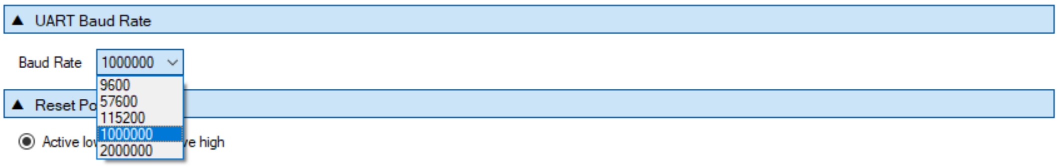
Note: The BD address file should always end with a zero BD address (`00:00:00:00:00:00`) and a new line at the end.

Table 29: BD Address Assignment Options - Load from File Mode

Option	Description
Start BD address	The BD address that the PLT session has started with.
Next BD address	The BD address from file that will be used in the first active DUT of the next PLT run.
BD address file	Path to the file that contains the BD addresses. Use button [...] on the right to navigate and select a file.

7.2.5 DUT Hardware Setup

7.2.5.1 UART Baud Rate



Baud Rate 1000000

Figure 42: UART Baud Rate

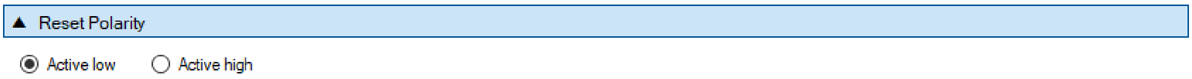
Table 30 shows the available options for the *UART Baud Rate* used during memory programming.

The *Baud Rate* selected here is used after the firmware (`uartboot_da1470x.bin`) has been downloaded to the DUT. The software will send a command to the DUT to change the UART baud rate to the one selected. All following UART communications with the DUT will be performed using the new baud rate. Note that this is happening only during memory programming where `uartboot_da1470x.bin` is used. During tests (`RF tests, XTAL trimming`, etc.), where the production test firmware is used, the baud rate is fixed to 115200 bit/s.

Table 30: UART Baud Rate

Option	Description
Baud Rate	● 9600 (bit/s) ● 57600 (bit/s) ● 115200 (bit/s) ● 1000000 (bit/s) ● 2000000 (bit/s) **Note:** 1 Mbit/s and 2 Mbit/s are the fastest and safest with 0% baud rate error.

7.2.5.2 Reset Polarity



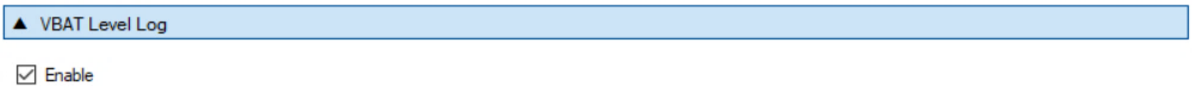
● Active low ○ Active high

Figure 43: Reset Polarity

The *Reset Polarity* selected here is used in combination to the reset method. If the reset method is set to VBAT only then this setting is ignored. Depending on the implementation, active high or active low reset signal might be required. This setting controls the reset polarity as active high or active low.

7.2.6 Test Settings

7.2.6.1 VBAT Level Log

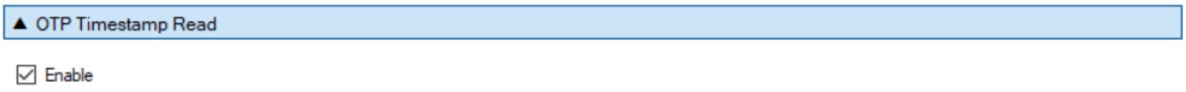


☑ Enable

Figure 44: VBAT Level Log

When this feature is enabled, PLT will send a command to the device to measure VBAT using its internal ADC. The VBAT level will then be logged. No, pass or fail limits exist for this test. It is only used for logging purposes.

7.2.6.2 OTP Timestamp Read



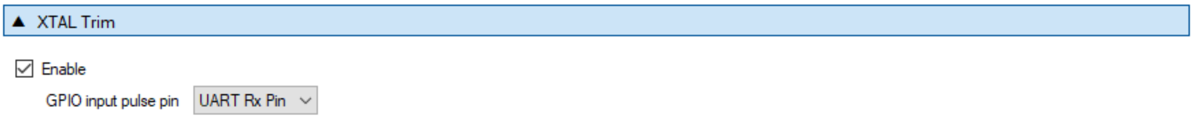
▲ OTP Timestamp Read

☑ Enable

Figure 45: OTP Timestamp Read

If this option is enabled, PLT will read the device timestamp from the OTP memory and log it. This operation is mainly used for logging purposes.

7.2.6.3 XTAL Trim



▲ XTAL Trim

☑ Enable

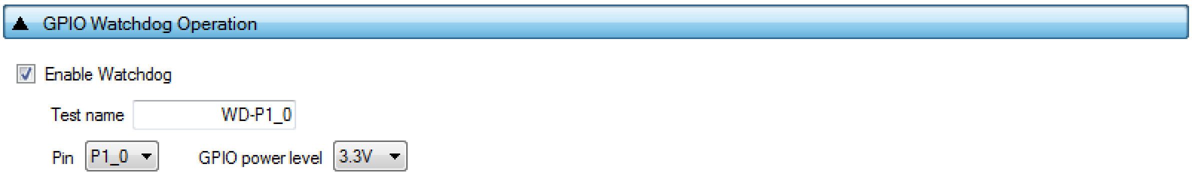
GPIO input pulse pin: UART Rx Pin

Figure 46: XTAL Trim

Table 31 describes the available options for the *XTAL Trim* operation.

Table 31: XTAL Trim

Option	Description
Enable	This option enables the automatic crystal oscillator frequency calibration procedure.
GPIO input pulse pin	The DUT GPIO to receive the reference pulse during calibration. UART RX pin can be used without any additional connection from the PLT hardware to the DUT.



▲ GPIO Watchdog Operation

☑ Enable Watchdog

Test name: WD-P1_0

Pin: P1_0 GPIO power level: 3.3V

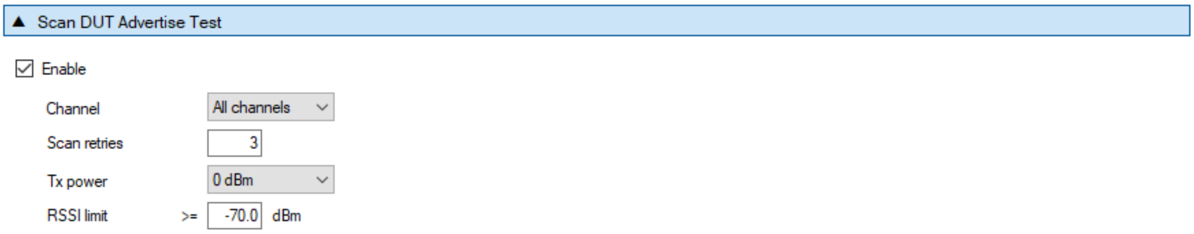
Figure 47: GPIO Watchdog Operation

Table 32 describes the available options for the *GPIO Watchdog operation*.

Table 32: GPIO Watchdog Operation

Option	Description
Enable Watchdog	This option enables the continuous toggling of a GPIO during the whole production testing and memory programming procedure, except during firmware download. The pulse on the GPIO has approximately 0.75 % duty cycle and 0.5 Hz frequency. **Note:** Production test firmware is downloaded through `uartboot_da1470x.bin`. After the uartboot firmware is downloaded, the watchdog pin will be pulsed.
Test name	The name assigned for this test.
Pin	Select the GPIO that will be toggled.
GPIO power level	Sets the power level of the GPIOs.

7.2.6.4 Scan DUT Advertise Test


▲ Scan DUT Advertise Test

☑ Enable

Channel: All channels

Scan retries: 3

Tx power: 0 dBm

RSSI limit >= -70.0 dBm

Figure 48: Scan DUT Advertise Test

Table 33 describes the available options for the *Scan DUT Advertise Test* operation.

Table 33: Scan DUT Advertise Test

Option	Description
Enable	This option enables the `Scan DUT Advertise Test` operation.
Channel	The Bluetooth® LE channel frequency used in the RF RX test using the Golden Unit.
Scan retries	The number of retries to perform the test.
Tx Power	The Bluetooth® LE Tx Power to be used by the device for the test.
RSSI limit	The RSSI limit for pass/fail criteria in the RF RX test using the Golden Unit. If the average RSSI of the device, after it has received the packets transmitted from the Golden Unit is less than that the test will be considered as failed.

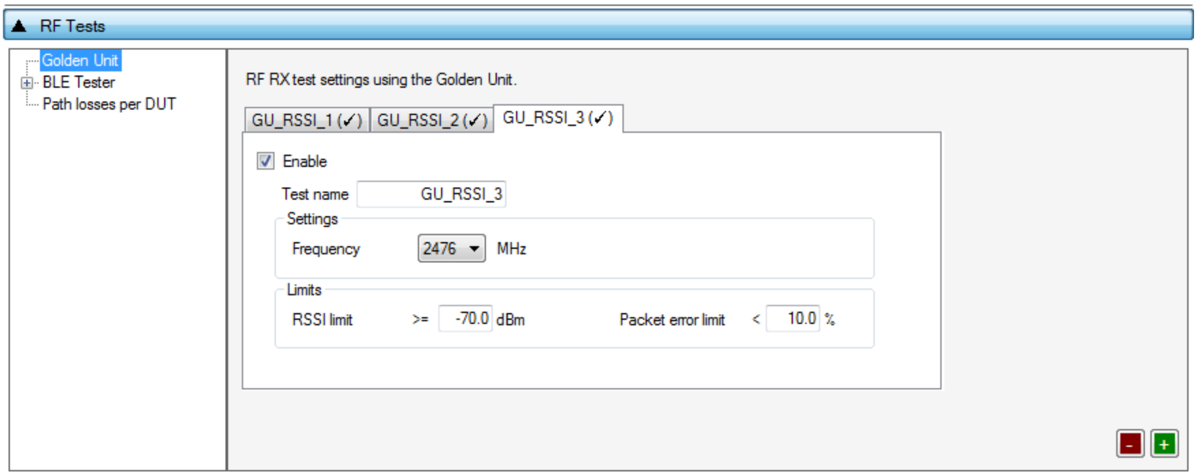
7.2.6.5 RF Tests

This section refers to various RF tests conducted between the DUTs and the Golden Unit or an external Bluetooth® LE tester.

The following tests can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g., - and + in Figure 49) at the bottom right side of each panel.

Note: When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Golden Unit



RF RX test settings using the Golden Unit.

☑ Enable

Test name: GU_RSSI_3

Frequency: 2476 MHz

RSSI limit >= -70.0 dBm

Packet error limit < 10.0 %

Figure 49: Golden Unit RF Tests

Table 34 describes the available options for the *RF RX* test using the *Golden Unit* as a transmitter.

In the RF RX test, the Golden Unit sends 500 packets. The DUTs are set in receive mode and the RSSI is measured. If the RSSI measured by the DUT reception is less than the specified *RSSI limit*, the device will fail, and the tests will stop for that particular device.

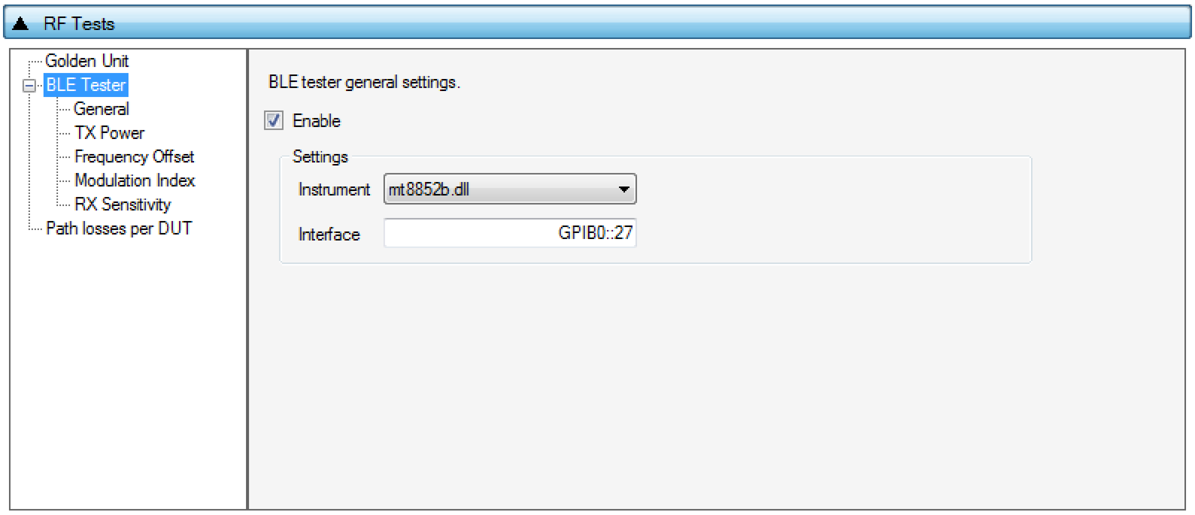
Table 34: Golden Unit RF Tests

Option	Description
Enable	This option enables the specific RF RX test using the Golden Unit as a transmitter.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Frequency	The Bluetooth® LE channel frequency used in the RF RX test using the Golden Unit.
RSSI limit	The RSSI limit for pass/fail criteria in the RF RX test using the Golden Unit. If the average RSSI of the device after it has received the packets transmitted from the Golden Unit is less than this value, the test will be considered as failed.
Packet error limit	This configures the PER limit for pass/fail criteria. If the percentage of the correct packets received is less than the value entered here, the test will fail.

BLE Tester

In the *BLE Tester* panels, several tests can be enabled that require an external Bluetooth® LE tester instrument. More detailed information about the Bluetooth® LE tester can be found in Ref. [1].

BLE Tester - General



BLE tester general settings.

☑ Enable

Settings

Instrument: mt8852b.dll

Interface: GPIB0::27

Figure 50: BLE Tester General Settings

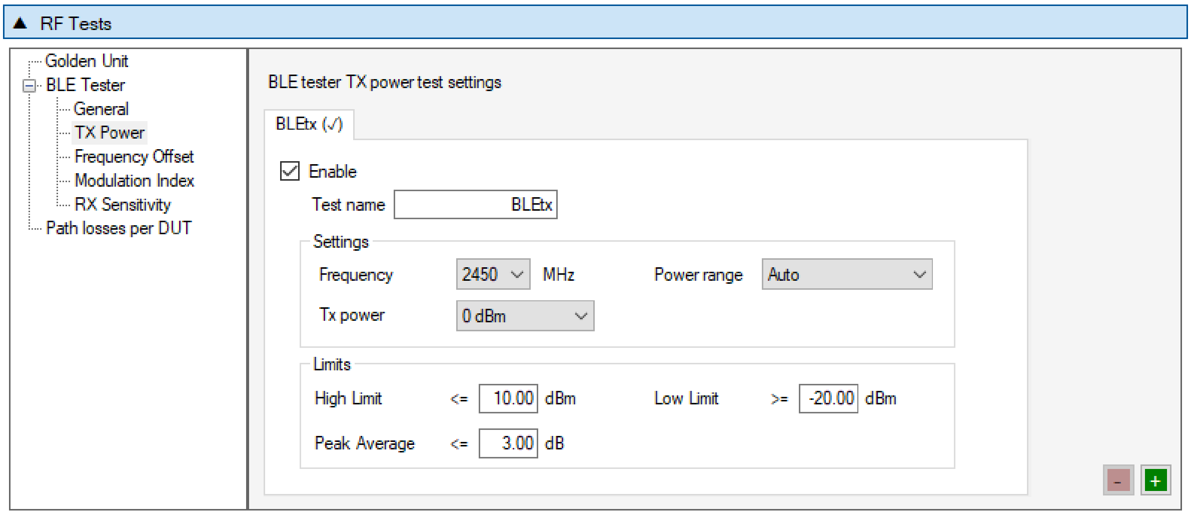
Table 35 describes the *General* settings for the *BLE Tester* supported tests. Any available external instrument found by the `ble_tester_driver` DLL and their interfaces can be selected.

Table 35: BLE Tester General Settings

Option	Description
Enable	This option enables the *BLE Tester* tests, which include: ● Bluetooth® LE Tester TX Power ● Frequency Offset

Option	Description
	- Modulation Index - RX Sensitivity
Instrument	Selects the Bluetooth® LE tester DLL. Names are shown only if a Bluetooth® LE tester instrument DLL exists in the project folder `ble_tester_instr_plugins`.
Interface	The interface of the instrument to be used by the driver.

BLE Tester - TX Power



▲ RF Tests

- Golden Unit
- BLE Tester
  - General
  - TX Power
  - Frequency Offset
  - Modulation Index
  - RX Sensitivity
- Path losses per DUT

BLE tester TX power test settings

BLEtx (✓)

☑ Enable

Test name: BLEtx

Settings

Frequency: 2450 MHz　Power range: Auto

Tx power: 0 dBm

Limits

High Limit <= 10.00 dBm　Low Limit >= -20.00 dBm

Peak Average <= 3.00 dB

Figure 51: BLE Tester TX Power

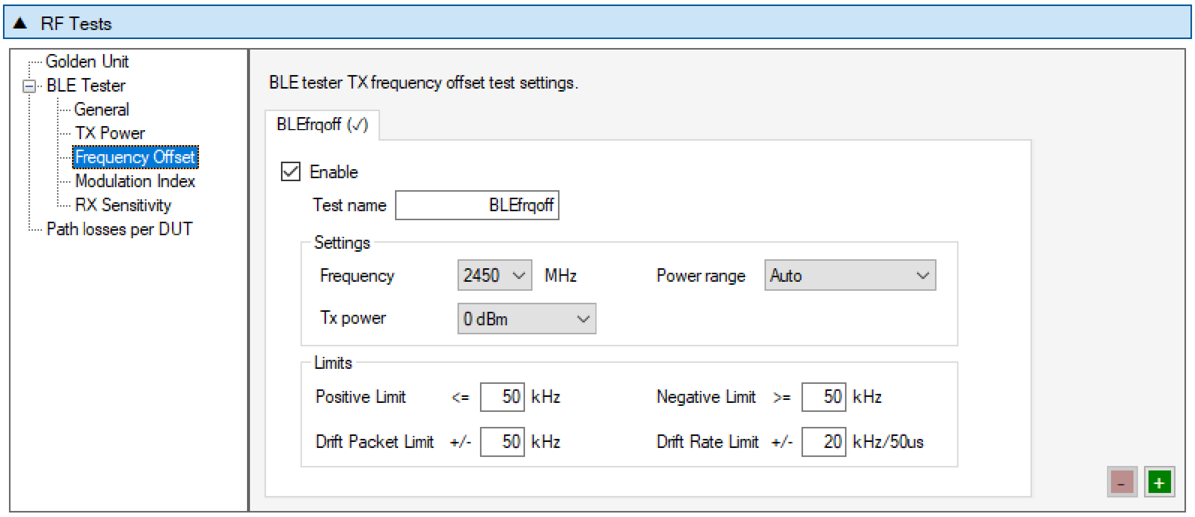
Table 36 describes the available options for the *TX Power* test using a Bluetooth® LE Tester instrument.

Table 36: BLE Tester TX Power

Option	Description
Enable	This option enables the specific TX power test using a Bluetooth® LE tester instrument.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Frequency	The Bluetooth® LE channel frequency used in the Bluetooth® LE TX power test.
Power range	Set the device TX output power range. Available options are: - Auto (No auto option for Litepoint IQxel-M. Sets the instrument to trigger at -25 dBm) - +22 dBm to +7 dBm - +9 dBm to -3 dBm - +5 dBm to -7 dBm - -4 dBm to -16 dBm - -12 dBm to -26 dBm - -24 dBm to -35 dBm Default value is *Auto.*
Tx Power	The DUT transmit output power. -50 dBm to +6 dBm is supported.

Option	Description
High limit	Set the average high-power limit for the Bluetooth® LE TX output power pass/fail test criteria.
Low limit	Set the average low-power limit for the Bluetooth® LE TX output power pass/fail test criteria.
Peak average	Set the peak-to-average power limit for the Bluetooth® LE TX output power pass/fail test criteria.

BLE Tester - Frequency Offset



BLE tester TX frequency offset test settings.

☑ Enable

Test name: BLEfrqoff

Frequency: 2450 MHz

Power range: Auto

Tx power: 0 dBm

Positive Limit <= 50 kHz

Negative Limit >= 50 kHz

Drift Packet Limit +/- 50 kHz

Drift Rate Limit +/- 20 kHz/50us

Figure 52: BLE Tester Frequency Offset

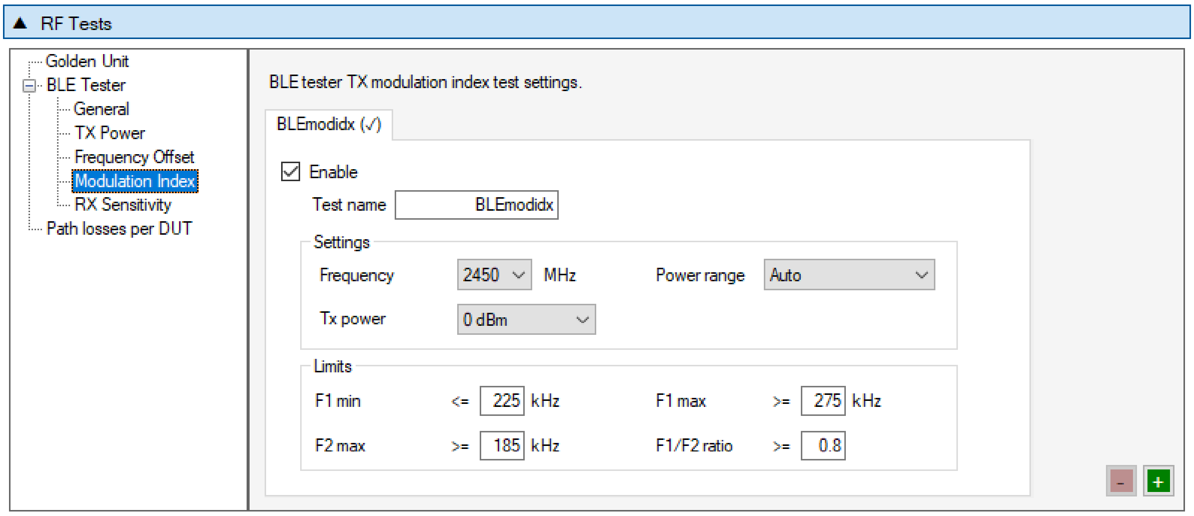
Table 37 describes the available options for the *Frequency Offset* test using a Bluetooth® LE Tester instrument.

Table 37: BLE Tester Frequency Offset

Option	Description
Enable	This option enables the specific TX frequency offset test using a Bluetooth® LE tester instrument.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Frequency	The Bluetooth® LE channel frequency used in the Bluetooth® LE TX frequency offset test.
Power range	Set the device TX output power range. Available options are: ● Auto ● +22 dBm to +7 dBm ● +9 dBm to -3 dBm ● +5 dBm to -7 dBm ● -4 dBm to -16 dBm ● -12 dBm to -26 dBm ● -24 dBm to -35 dBm Default value is *Auto.*
Tx Power	The DUT transmit output power. -50 dBm to +6 dBm is supported.

Option	Description
Positive limit	Set the maximum positive offset limit in kHz for the TX carrier frequency offset pass/fail test criteria.
Negative limit	Set the maximum negative offset limit in kHz for the TX carrier frequency offset pass/fail test criteria.
Drift packet limit	Set the overall packet drift in kHz for the TX drift pass/fail test criteria.
Drift rate limit	Set the drift rate limit in kHz/50 µs for the TX drift pass/fail test criteria.

BLE Tester - Modulation Index



BLE tester TX modulation index test settings.

BLEmodidx (✓)

☑ Enable

Test name: BLEmodidx

Settings

Frequency: 2450 MHz

Power range: Auto

Tx power: 0 dBm

Limits

F1 min <= 225 kHz

F1 max >= 275 kHz

F2 max >= 185 kHz

F1/F2 ratio >= 0.8

Figure 53: BLE Tester Modulation Index

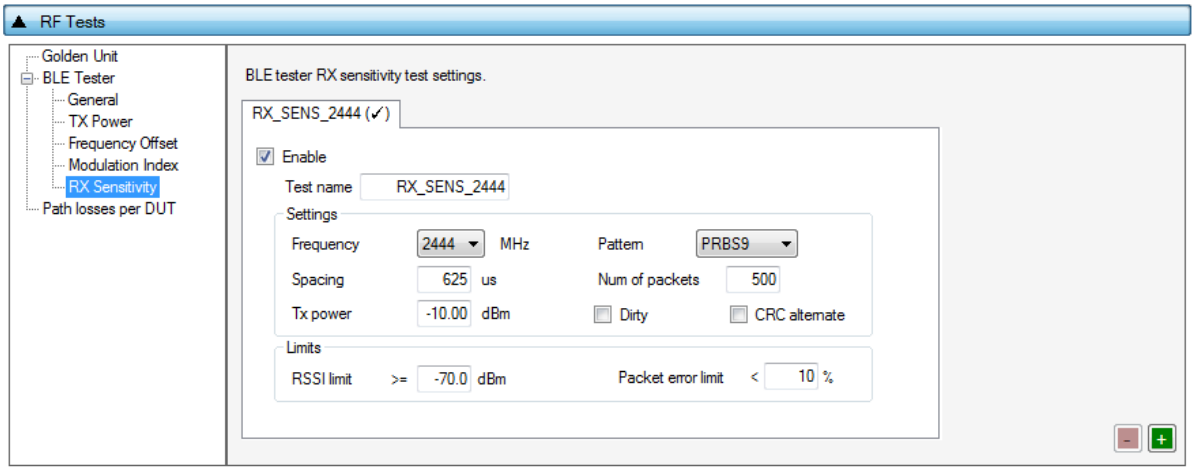
Table 38 describes the available options for the *Modulation Index* test using a Bluetooth® LE Tester instrument.

Table 38: BLE Tester Modulation Index

Option	Description
Enable	This option enables the specific TX modulation index test using a Bluetooth® LE tester instrument.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Frequency	The Bluetooth® LE channel frequency used in the Bluetooth® LE TX modulation index offset test.
Power range	Set the device TX output power range. Available options are: ● Auto ● +22 dBm to +7 dBm ● +9 dBm to -3 dBm ● +5 dBm to -7 dBm ● -4 dBm to -16 dBm ● -12 dBm to -26 dBm ● -24 dBm to -35 dBm Default value is *Auto.*

Option	Description
Tx Power	The DUT transmit output power. -50 dBm to +6 dBm is supported.
F1 min	Set the F1 minimum average limit in kHz for the TX modulation index pass/fail test criteria.
F1 max	Set the F1 maximum average limit in kHz for the TX modulation index pass/fail test criteria.
F2 max	Set the F2 maximum limit in kHz for the TX modulation index pass/fail test criteria.
F1/F2 ratio	Set the F1/F2 maximum average ratio limit for the TX modulation index pass/fail test criteria.

BLE Tester - RX Sensitivity



BLE tester RX sensitivity test settings.

RX_SENS_2444 (✓)

☑ Enable

Test name RX_SENS_2444

Settings

Frequency 2444 MHz

Pattern PRBS9

Spacing 625 us

Num of packets 500

Tx power -10.00 dBm

□ Dirty

□ CRC alternate

Limits

RSSI limit >= -70.0 dBm

Packet error limit < 10 %

Figure 54: BLE Tester RX Sensitivity

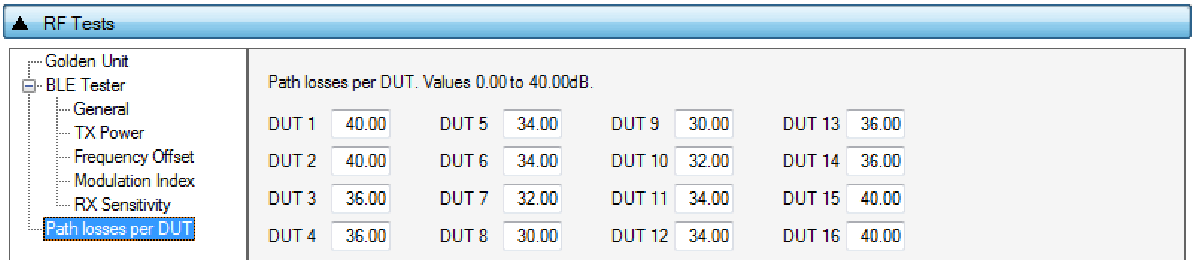
Table 39 describes the available options for the *RX Sensitivity* test using a Bluetooth® LE Tester instrument.

Table 39: BLE Tester RX Sensitivity

Option	Description
Enable	This option enables the specific RX sensitivity test using a Bluetooth® LE tester instrument.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Frequency	The Bluetooth® LE channel frequency used in the Bluetooth® LE RX sensitivity test.
Pattern	The bit pattern of the TX data. Available options are: ● PRBS9 ● 10101010 ● 11110000
Spacing	The packet spacing in µs.
Num of packets	The number of packets the Bluetooth® LE tester instrument to transmit.
Tx power	The TX output power of the Bluetooth® LE tester instrument. Suggested values are 0 to -10 dBm.

Option	Description
Dirty	When enabled, the Bluetooth® LE tester packet generator can use a dirty table to transmit.
CRC alternate	When enabled, the Bluetooth® LE tester will alternatingly send packets with CRC correct and CRC incorrect.
RSSI limit	The RSSI limit for pass/fail criteria in the RF RX sensitivity test. If the average RSSI of the device after it has received the transmitted packets is less than this value, the test will be considered as failed.
Packet error limit	This configures the PER limit for pass/fail criteria. If the percentage of the correct packets received is less than the value entered here, the test will fail.

Path Losses per DUT



Path losses per DUT. Values 0.00 to 40.00dB.

DUT 1	40.00	DUT 5	34.00	DUT 9	30.00	DUT 13	36.00
DUT 2	40.00	DUT 6	34.00	DUT 10	32.00	DUT 14	36.00
DUT 3	36.00	DUT 7	32.00	DUT 11	34.00	DUT 15	40.00
DUT 4	36.00	DUT 8	30.00	DUT 12	34.00	DUT 16	40.00

Figure 55: Path Losses per DUT

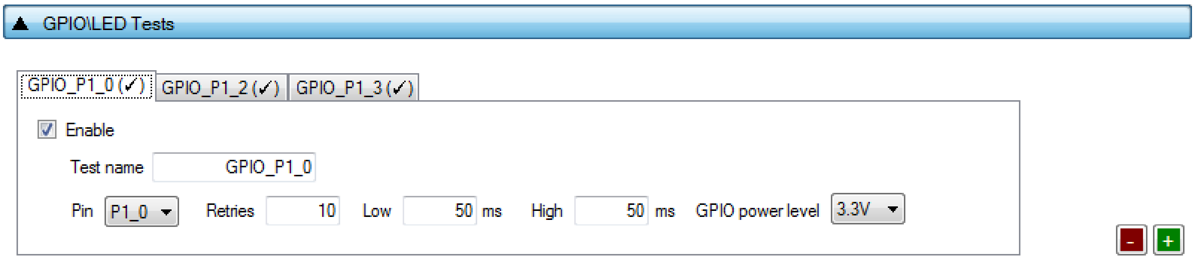
Table 40 describes the available options for the *Path losses per DUT.*

Based on the relative position of each DUT during RF tests and since the RF tests are performed over the air, values can be used to correct for any path losses. These values are added to the limits of the TX Power and RF RX RSSI tests. Additional information can be found in Appendix D and Appendix F.

Table 40: Path Losses per DUT from RF Tests Options

Option	Description
DUT1-16	Set the path loss value for each DUT. These will be added as corrections to the limits of the TX Power and RF RX RSSI tests

7.2.6.6 GPIO/LED Test



☑ Enable

Test name: GPIO_P1_0

Pin: P1_0 ▾ Retries: 10 Low: 50 ms High: 50 ms GPIO power level: 3.3V ▾

Figure 56: GPIO/LED Tests

GPIO/LED Tests can have multiple instances with different settings. Tests can be added or removed using the two buttons (e.g., - and + in Figure 56) at the bottom right side of each panel.

Note: When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

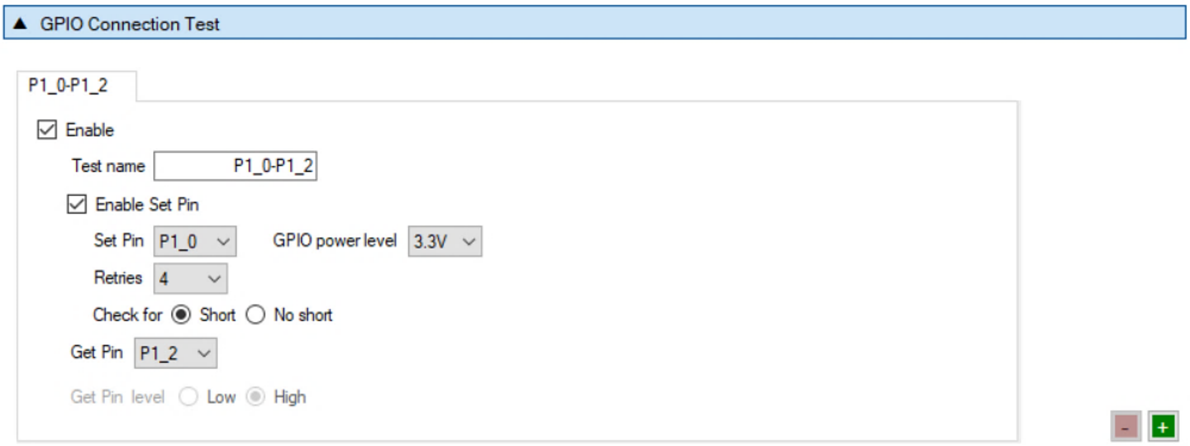
Table 41 describes the available options for the *GPIO/LED Tests* Options.

In these tests, a specific pulse can be given to a GPIO, and any LED connected to it can be visually tested. The *Pin* option sets the GPIO to be used, *Low* and *High* define the duty cycle and the *Retries* the number of pulses.

Table 41: GPIO/LED Tests

Option	Description
Enable	This option enables the GPIO/LED toggling. Can be used for visual LED testing.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Pin	The GPIO that will be used for the specific test.
Retries	Number of pulses to be generated for the specific test.
Low	Sets the amount of the OFF time of the pulse in ms for the specific test.
High	Sets the amount of the ON time of the pulse in ms for the specific test.
GPIO power level	Sets the power level of the GPIOs.

7.2.6.7 GPIO Connection Test



▲ GPIO Connection Test

P1_0-P1_2

☑ Enable

Test name: P1_0-P1_2

☑ Enable Set Pin

Set Pin: P1_0 GPIO power level: 3.3V

Retries: 4

Check for ● Short ○ No short

Get Pin: P1_2

Get Pin level ○ Low ● High

Figure 57: GPIO Connection Test

GPIO Connection Tests can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g., - and + in Figure 57) at the bottom right side of each panel.

Note: When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 42 describes the available options for the *GPIO Connection Test*.

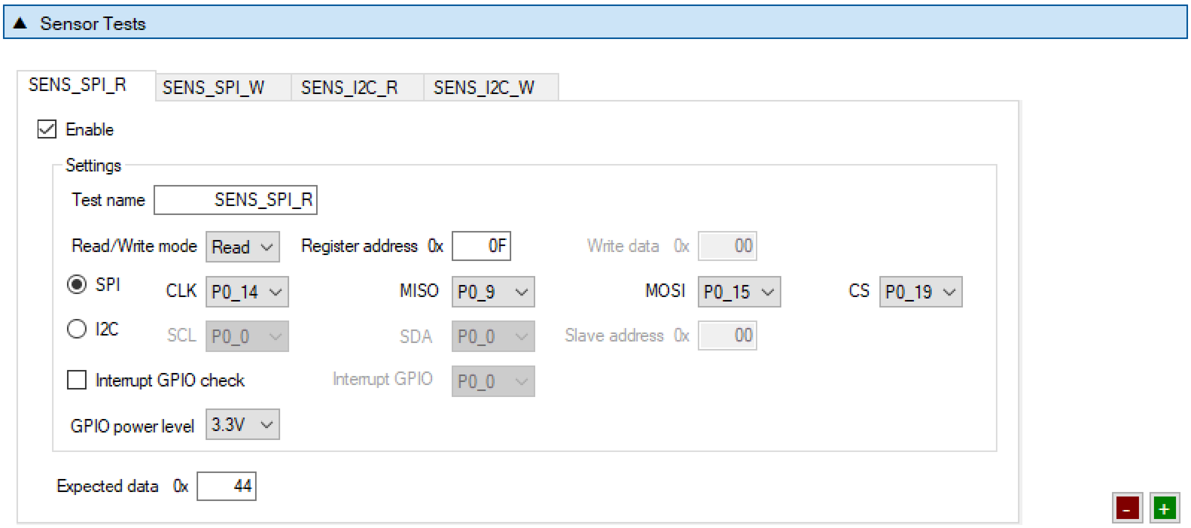
When enabled, the PLT software will check the connection of the specified GPIO (`Get Pin`) by either checking its state or the connection with another pin (`Set Pin`). In the latter case, the user gives the `Set Pin` and the state to check. It will also check for shorts between given GPIOs.

Table 42: GPIO Connection Test

Option	Description
Enable	This option enables the specific custom test.

Option	Description
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Enable Set Pin	Enables the use of the secondary GPIO to drive the GPIO under test. When this option is set, the `Get Pin level` option will be disabled.
Set Pin	Select the GPIO to be tested
GPIO power level	The output GPIO power level, 3.3 V or 1.8 V.
Retries	How many times the software will check for GPIO connection or short. In every retry it will change the `Set Pin` level and check the `Get Pin` level.
Check for Short/No short	If Short is selected, PLT will check whether the `Set Pin` has the same level with the `Get Pin` for all `Retries` tested. If `No short` is selected, PLT will check whether `Get Pin` is always low no matter what the `Set Pin` level is.
Get Pin	Select the GPIO to be tested.
Get Pin level	Sets the GPIO state the test awaits to see in the `Get Pin`. This option is disabled if the Set Pin mode is enabled.

7.2.6.8 Sensor Test


▲ Sensor Tests

SENS_SPI_R | SENS_SPI_W | SENS_I2C_R | SENS_I2C_W

☑ Enable

Settings

Test name: SENS_SPI_R

Read/Write mode: Read | Register address 0x: 0F | Write data 0x: 00

● SPI | CLK: P0_14 | MISO: P0_9 | MOSI: P0_15 | CS: P0_19

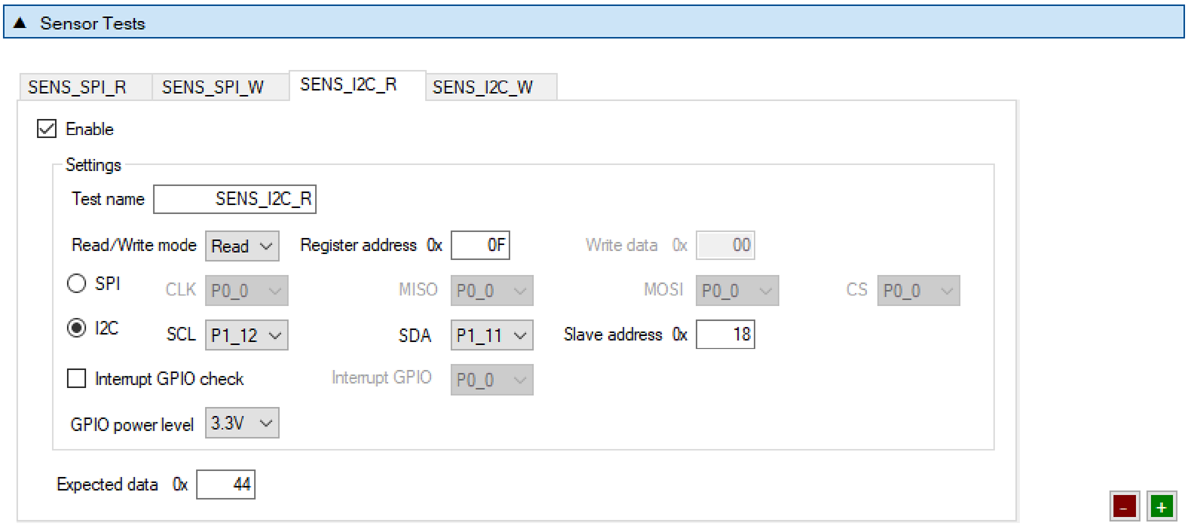
○ I2C | SCL: P0_0 | SDA: P0_0 | Slave address 0x: 00

☐ Interrupt GPIO check | Interrupt GPIO: P0_0

GPIO power level: 3.3V

Expected data 0x: 44

Figure 58: Sensor Test (SPI)



▲ Sensor Tests

SENS_SPI_R | SENS_SPI_W | SENS_I2C_R | SENS_I2C_W

☑ Enable

Settings

Test name: SENS_I2C_R

Read/Write mode: Read | Register address 0x: 0F | Write data 0x: 00

○ SPI | CLK: P0_0 | MISO: P0_0 | MOSI: P0_0 | CS: P0_0

● I2C | SCL: P1_12 | SDA: P1_11 | Slave address 0x: 18

□ Interrupt GPIO check | Interrupt GPIO: P0_0

GPIO power level: 3.3V

Expected data 0x: 44

Figure 59: Sensor Test (I2C)

Sensor Tests can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g., − and + in Figure 58) at the bottom right side of each panel.

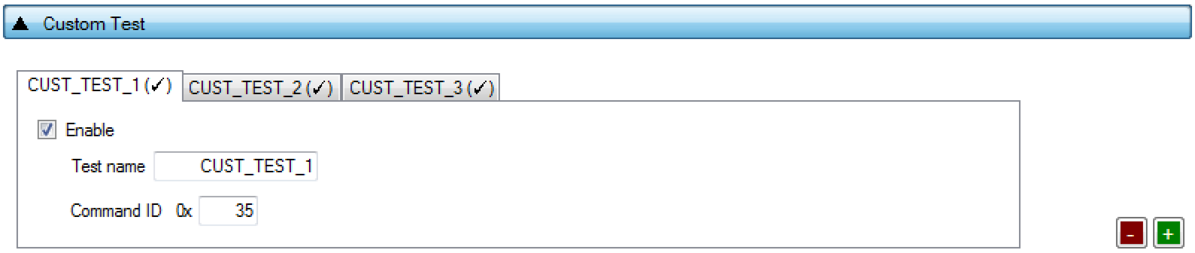
Note: When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 43 describes the available options for the *Sensor Tests* Options.

Table 43: Sensor Tests

Option	Description
Enable	This option enables the specific sensor test.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Read / Write mode	Select the sensor test procedure, to read or write.
Register address	The sensors register address to read or write data.
Write data	The byte to be written at the sensor register.
SPI / I2C	Select the interface that the sensor is connected to.
SPI - CLK	Select the GPIO for the sensor SPI CLK.
SPI - MISO	Select the GPIO for the sensor SPI MISO.
SPI - MOSI	Select the GPIO for the sensor SPI MOSI.
SPI - CS	Select the GPIO for the sensor SPI CS.
I2C - SCL	Select the GPIO for the sensor I2C SCL.
I2C - SDA	Select the GPIO for the sensor I2C SDA.
Slave address	The sensor I2C bus slave address.
Interrupt GPIO check	Enables the sensor interrupt signal test via GPIO.
Interrupt GPIO	Select the GPIO to be used as a sensor interrupt.
GPIO power level	Sets the power level of the GPIOs.
Expected data	The received sensor byte that will be expected on a successful operation.

7.2.6.9 Custom Test



☑ Enable

Test name: CUST_TEST_1

Command ID: 0x 35

Figure 60: Custom Test

Custom tests can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g., - and + in Figure 60) at the bottom right side of each panel.

Note: When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

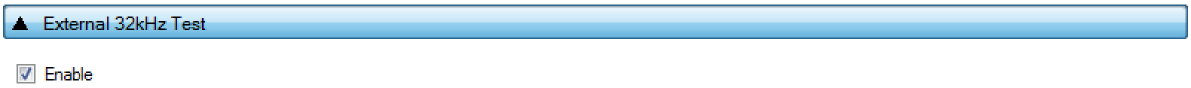
Table 44 describes the available options for the *Custom Tests*.

When enabled, the PLT software will send an HCI command through UART to activate a customer-defined test that will run on the DUTs. The HCI custom test command will contain a single byte as data (the *Command ID* byte), to be used mainly as identification for a specific test in the firmware. Default functionality of the production test firmware is to respond with the same `Command ID`. Otherwise, the test will be considered as failed.

Table 44: Custom Tests Options

Option	Description
Enable	This option enables the specific custom test.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Command ID	The byte that will be sent to the device running the production test firmware.

7.2.6.10 External 32 kHz Test



☑ Enable

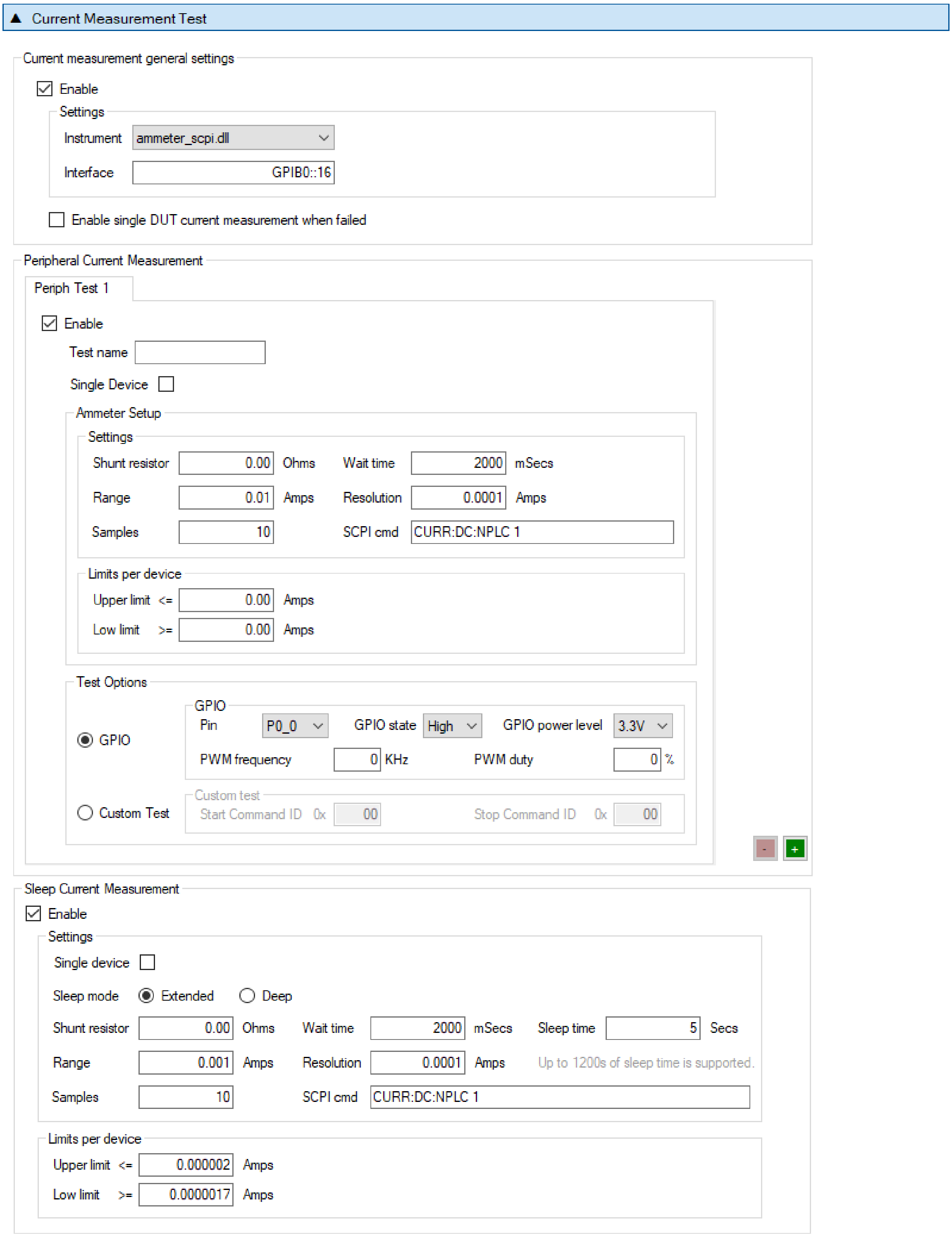
Figure 61: External 32 kHz Test

Table 45 describes the available options for the External 32 kHz *Tests*.

Table 45: External 32 kHz Tests Options

Option	Description
Enable	This option enables the external 32 kHz low power clock test.

7.2.6.11 Current Measurement Test



▲ Current Measurement Test

Current measurement general settings

☑ Enable

Settings

Instrument: ammeter_scpi.dll

Interface: GPIB0::16

□ Enable single DUT current measurement when failed

Peripheral Current Measurement

Periph Test 1

☑ Enable

Test name:

Single Device □

Ammeter Setup

Settings

Shunt resistor: 0.00 Ohms

Wait time: 2000 mSecs

Range: 0.01 Amps

Resolution: 0.0001 Amps

Samples: 10

SCPI cmd: CURR:DC:NPLC 1

Limits per device

Upper limit <= 0.00 Amps

Low limit >= 0.00 Amps

Test Options

● GPIO

GPIO

Pin: P0_0

GPIO state: High

GPIO power level: 3.3V

PWM frequency: 0 KHz

PWM duty: 0 %

○ Custom Test

Custom test

Start Command ID 0x 00

Stop Command ID 0x 00

Sleep Current Measurement

☑ Enable

Settings

Single device □

Sleep mode: ● Extended ○ Deep

Shunt resistor: 0.00 Ohms

Wait time: 2000 mSecs

Sleep time: 5 Secs

Range: 0.001 Amps

Resolution: 0.0001 Amps

Up to 1200s of sleep time is supported.

Samples: 10

SCPI cmd: CURR:DC:NPLC 1

Limits per device

Upper limit <= 0.000002 Amps

Low limit >= 0.0000017 Amps

Figure 62: Current Measurement Tests

In this test, an external ammeter can be used to measure the total current consumption of all active DUTs. The ammeter can be connected in the blue banana plugs as described in Section 5.7 or to an external power supply (if present) depending the selected VBAT/Reset Mode (Section 7.2.2.6).

During measurement, PLT will control the instrument using the `ammeter_driver` DLL Ref. [1]. Table 46 describes the instrument selection settings found by the `ammeter_driver` DLL, Table 47 describes the settings used for each of the peripheral current measurement tests and Table 48 describes the current measurement options for each sleep state.

Note: Modifications in the production test firmware are mandatory in order to achieve the correct current consumption of a specific hardware design (IC and peripherals) for each sleep state. Running the default firmware without any modifications specific for the hardware design, may cause increased current consumption.

Peripheral Current Measurement Tests can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g., - and + in Figure 62) at the bottom right side of each panel.

Note: When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 46: Current Measurement Tests

Option	Description
Enable	This option enables all Current Measurement tests, which include: ● Peripheral current measurements ● Extended sleep current measurement ● Deep sleep current measurement Only one of the Extended/Deep sleep current measurements can be selected.
Instrument	Select the Ammeter instrument DLL name. Names are shown only if an ammeter DLL exists in the project `ammeter_instr_plugins` folder.
Interface	The interface of the instrument to be used by the driver.
Enable single DUT current measurement when failed	If this option is enabled and if the measurement taken is outside of the limits, PTL will reset all devices and begin a firmware download and measure the current to each device separately, in order to identify which exact device failed.

Table 47: Current Measurement Test – Peripheral Current Measurement

Option	Description
Enable	This option enables the specific peripheral current measurement test.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Single Device	If this option is enabled, PLT will measure the current consumption one device at the time. Initially, it will power off all DUTs. It will then power on one by one, download firmware and measure the current individually. This procedure takes a lot of time. It should only be used for production setup purposes, to identify the correct limits by measuring multiple DUTs and taking the average.
Shunt resistor	The value of the shunt resistor used for peripheral measurement. Applicable only if a voltmeter or a NI-DAQ is used to measure current instead of a DMM. Values from 0 to 9999 are supported.
Wait time	The time in ms the PLT will wait before taking a current measurement, after it has sent an instruction to the DUTs to go into a specific sleep state. Supported values are 1 to 500000 ms.
Range	The range value in Ampere units that the ammeter will operate. Supported values are 0 to 9999 and default value is 0.001 A. When the `ammeter_scpi.dll` is used, if this value is set to zero, the instrument will use the automatic range functionality.

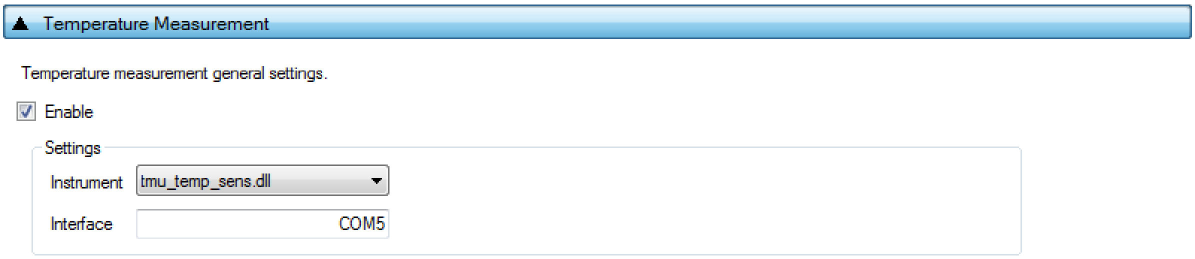
Option	Description	
Resolution	The ammeter resolution value in Ampere units.	
Samples	The number of samples that the ammeter will read and average. 1 to 1000 is supported.	
SCPI cmd	A SCPI command to be passed to the ammeter just before the measurement is taken. Supports multiple commands separated with a column. Up to 256 characters are supported.	
Upper limit	The upper limit value of the peripheral current measurement test procedure, for a single DUT. Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing. **Note:** During testing, some DUTs may fail until the current measurement test takes place. In that case, the PLT will automatically re-calculate the total upper limit by using the value given for a single DUT multiplied with the number of the active DUTs at the time that the test will run.	
Low limit	The low limit value of the peripheral current measurement test, for a single DUT. Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing. **Note:** During testing, some DUTs may fail until the current measurement test takes place. In that case, the PLT will automatically re-calculate the total low limit by using the value given for a single DUT multiplied with the number of the active DUTs at the time that the test will run.	
Test Options	Select between a PWM GPIO test and custom test. **Note:** For the custom tests to work, a modified production test firmware must be created with tests that set the DUTs to specific states before the current measurement test. Each test must be assigned to a specific opcode. The custom tests are the exact same as in Section 7.2.6.9.	
Test Options - GPIO	Pin	Sets the GPIO to toggle with the PWM pulse.
	GPIO state	Sets the active state of the GPIO.
	GPIO power level	Sets the power level of the GPIOs.
	PWM frequency	Sets the PWM frequency.
	PWM duty	Sets the PWM duty cycle.
Test Options – Custom Test	Start Command ID	The opcode of the custom test that sets the state of the DUT.
	Stop Command ID	The opcode of the custom test that restores the DUT to its original state.

Table 48: Current Measurement Test - Sleep Current Measurement

Option	Description
Enable	This option enables the sleep current measurement using the ammeter provided in the *Instrument* section.
Single Device	If this option is enabled, PLT will measure the current consumption, one device at the time. Initially, it will power off all DUTs. It will then power on one by one, download firmware and measure the current individually. This procedure takes a lot of time. It should only be used for production setup purposes, to identify the correct limits by measuring multiple DUTs and taking the average.
Sleep mode	User can select either the Extended or the Deep sleep mode.
Shunt resistor	The value of the shunt resistor used for sleep current measurement. Applicable only if a voltmeter or a NI-DAQ is used to measure current instead of a DMM. Values from 0 to 9999 are supported.

Option	Description
Wait time	The time in ms the PLT will wait before taking a current measurement, after it has sent an instruction to the DUTs to go into a specific sleep state. Supported values are 1 to 500000 ms.
Sleep time (only for extended sleep)	The time in seconds that the DUTs will remain in extended sleep mode. A timer in the production test firmware will wake up the devices and set them to idle mode. Supported values are 1 to 9 sec for DA14580/1/2/3 and up to 1200 sec for the rest.
Range	The range value in Ampere units that the ammeter will operate. Supported values are 0 to 9999 and default value is 0.001 A. When the `ammeter_scpi.dll` is used, if this value is set to zero, the instrument will use the automatic range functionality.
Resolution	The ammeter resolution value in Ampere units.
Samples	The number of samples that the ammeter will read and average. 1 to 1000 is supported.
SCPI cmd	A SCPI command to be passed to the ammeter just before the measurement is taken. Supports multiple commands separated with a column. Up to 256 characters are supported.
Upper limit	The upper limit value for the sleep current measurement test procedure, for a single DUT. Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing. **Note:** During testing, some DUTs may fail until the current measurement test takes place; in that case, the PLT will automatically re-calculate the total upper limit by using the value given for a single DUT multiplied with the number of the active DUTs at the time that the test will run.
Low limit	The low limit value for the sleep current measurement test procedure, for a single DUT. Next to this input field the total upper limit current consumption for all the enabled DUTs is shown, which is the value to be used during testing. **Note:** During testing, some DUTs may fail until the current measurement test takes place; in that case, the PLT will automatically re-calculate the total low limit by using the value given for a single DUT multiplied with the number of the active DUTs at the time that the test will run.

7.2.6.12 Temperature Measurement Test



▲ Temperature Measurement

Temperature measurement general settings.

☑ Enable

Settings

Instrument: tmu_temp_sens.dll

Interface: COM5

Figure 63: Temperature Measurement Test

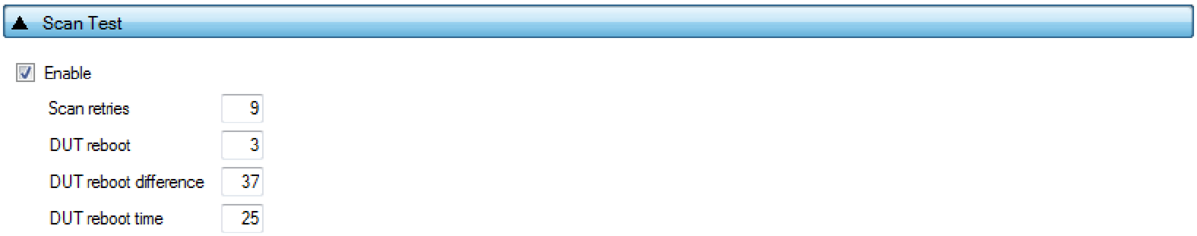
Table 49 describes the available options for the *Temperature Measurement Test*.

Table 49: Temperature Measurement Test

Option	Description
Enable	This option enables the Temperature measurement test.

Option	Description
Instrument	Select the Temperature measurement DLL. Names are shown only if a Temperature measurement instrument DLL exists in the project folder `temp_meas_instr_plugins`.
Interface	The interface of the instrument to be used by the driver.

7.2.6.13 Scan Test



☑ Enable

Scan retries 9

DUT reboot 3

DUT reboot difference 37

DUT reboot time 25

Figure 64: Scan Test

Table 50 describes the available options for the *Scan Test*.

By enabling this test, the Golden Unit will scan for the DUT’s BD addresses advertised after the customer firmware has been programmed to the DUT. For this test to work, a bootable firmware with the ability to advertise with the BD address given by the PLT must be burned into each DUT. Additionally, the BD addresses provided by the PLT should be burned into OTP CS memory such that the devices advertise with the BD addresses the tool uses.

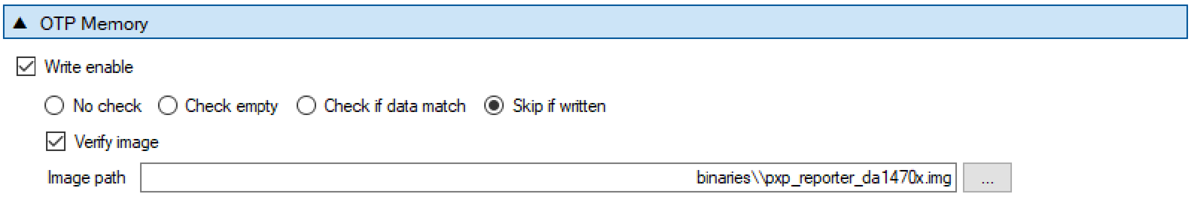
Table 50: Scan Test Options

Option	Description
Enable	This option enables the Scan test.
Scan retries	The total number of Bluetooth® LE advertising scans the Golden Unit will perform.
DUT reboot	Define after how many retries the PLT will reboot the DUTs.
DUT reboot difference	Set the time difference between each DUT when the PLT reboots the devices, in order to avoid air collisions.
DUT reboot time	The time the VBAT will remain low during the device reboot. This value is time in ms*100 (e.g., 15 is 1500 ms).

7.2.7 Memory Functions

This section describes the *Memory Functions* settings available when using devices. Memory functions include OTP, QSPI or OQSPI flash memory programming.

7.2.7.1 OTP Memory



▲ OTP Memory

☑ Write enable

○ No check ○ Check empty ○ Check if data match ● Skip if written

☑ Verify image

Image path binaries\\pxp_reporter_da1470x.img ...

Figure 65: OTP Memory

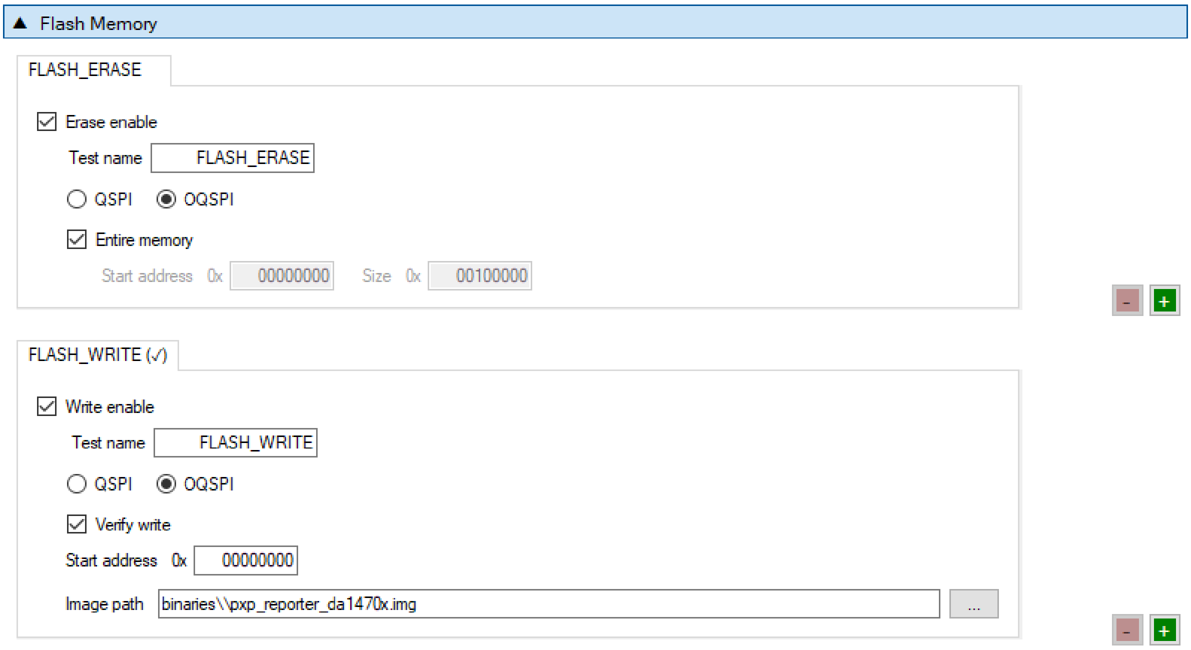
This test enables the OTP memory programming. Table 51 describes the available options for the *OTP Memory* image write operation.

Note: If the binary is larger than the available OTP image area (OTP memory excluding the header area), the PLT software will split the binary into two parts. The first part will contain only the OTP image area. The second part will contain the OTP header fields, split in OTP words. PLT will burn the non-zero words one by one, as single OTP entries in the OTP header area. The check empty feature will handle the first part as an OTP image binary. The second part will be checked word by word.

Table 51: OTP Memory

Option	Description
Write enable	This option enables the OTP image write operation.
• No check • Check empty • Check if data match • Skip if written	Memory protection options: **No check:** No protection is enabled. PLT will attempt to burn the OTP memory without running any check. **Check empty:** PLT will first check if the OTP memory to be burned is empty. If it is empty will burn it. **Check if data match:** PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory and continue without error. **Skip if written:** PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error.
Verify image	If this option is enabled, PLT will read back the contents of the OTP memory and compare them to the original image file. If these do not match it will fail.
Image path	Via this field, the user specifies the binary file to be burned into the OTP. A `.bin` binary file of any name can be selected.

7.2.7.2 Flash Memory



▲ Flash Memory

FLASH_ERASE

☑ Erase enable

Test name FLASH_ERASE

○ QSPI ● OQSPI

☑ Entire memory

Start address 0x 00000000 Size 0x 00100000

FLASH_WRITE (✓)

☑ Write enable

Test name FLASH_WRITE

○ QSPI ● OQSPI

☑ Verify write

Start address 0x 00000000

Image path binaries\\pxp_reporter_da1470x.img

Figure 66: Flash Memory

This section describes how the QSPI Flash memory can be erased and programmed.

Both erase and write operations can have multiple instances with different settings. Tests can be added and removed using the two buttons (e.g., - and + in Figure 66) at the bottom right side of each panel.

Note: When adding or removing a test all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

The QSPI flash memory should be erased before any image is written to it. Table 52 describes the available options for the *QSPI Flash Erase* operation.

Table 52: QSPI Flash Erase

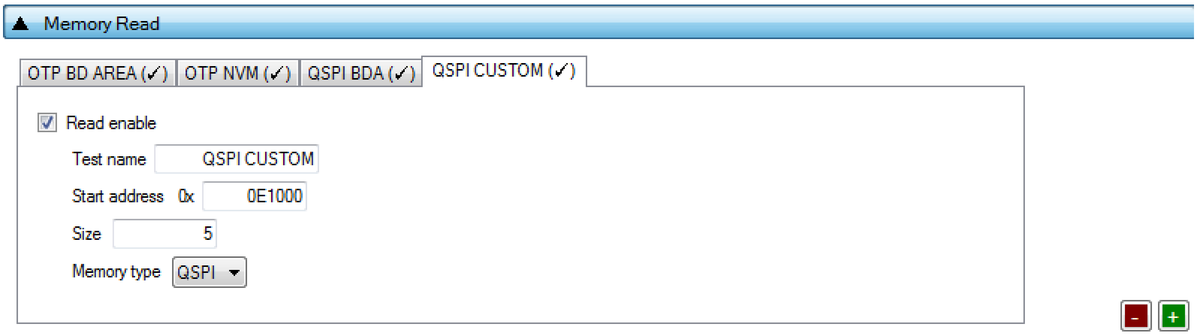
Option	Description
Erase enable	This will enable the specific QSPI flash erase test.
QSPI/OQSPI	QSPI refers to the QSPI interface using a QSPI compatible flash OQSPI refers to the OSPI interface using a QSPI or OQSPI compatible flash
Entire memory	This option is only available for the *Erase enable* option. When this checkbox is selected, the entire memory can be erased. Otherwise, the user can give a start address and a specific number of bytes to be erased.
Start address	The user can enter a specific start address for the QSPI erasure to start.
Size	The size in bytes to erase, starting from the *Start address* as explained above.

After every *QSPI/OQSPI Flash erase* test has finished, the *QSPI/OQSPI image write* tests will begin. Table 53 describes the available options for the *QSPI/OQSPI Flash Image Write* operation.

Table 53: Flash Image Write

Option	Description
Write enable	This will enable the specific QSPI or OQSPI flash image programming test.
QSPI/OQSPI	QSPI refers to the QSPI interface using a QSPI compatible flash OQSPI refers to the OSPI interface using a QSPI or OQSPI compatible flash
Verify image	By selecting this option, the PLT software will read back the contents of the flash memory and compare them to the original image file. If these do not match, the memory programming will fail.
Start address	The user can configure the flash start address where the image will be written.
Image path	Via this field, the user specifies the image file to be burned into the flash memory. A `.bin` binary file of any name can be selected.

7.2.7.3 Memory Read



☑ Read enable

Test name: QSPI CUSTOM

Start address 0x: 0E1000

Size: 5

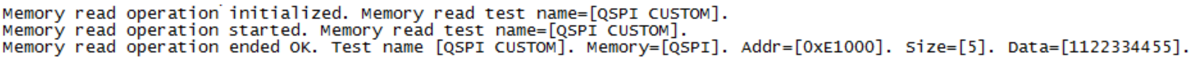
Memory type: QSPI ▾

Figure 67: Memory Read Test

Memory Read Tests can have multiple instances with different settings. Tests can be added or removed using the two buttons (e.g., − and + in Figure 67) at the bottom right side of each panel.

Note: When adding or removing a test, all settings are refreshed with the values written to the XML file, meaning that any unsaved settings will be lost.

Table 54 describes the memory read test options. With this test, the user can read up to 64MBytes of data from any address from any available memory for the devices, such as OTP and QSPI. An example of how the data appears on the log file is shown in Figure 68.



```
Memory read operation initialized. Memory read test name=[QSPI CUSTOM].
Memory read operation started. Memory read test name=[QSPI CUSTOM].
Memory read operation ended OK. Test name [QSPI CUSTOM]. Memory=[QSPI]. Addr=[0xE1000]. Size=[5]. Data=[1122334455].
```

Figure 68: Memory Read Test Example Log File

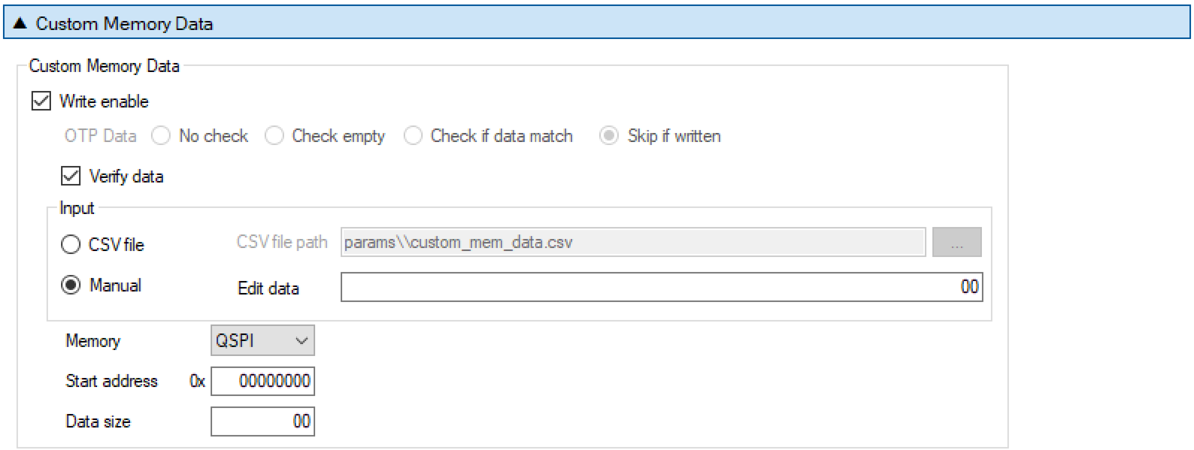
Table 54: Memory Read Test

Option	Description
Read enable	This will enable the specific memory reading test.
Test name	The name assigned to each test. The assigned name will be shown on the tab and next to it an indication showing whether the specific test is enabled or not.
Start address	Configures the start address. Valid addresses for OTP are 0x0000 – 0x0FFC.
Size	Number of bytes to read, up to 64MBytes. If data to be read are greater than 256 bytes, then a file will be created to store the data under folder `mem_read_test` in the PLT execution path.
Memory type	The type of memory to read the data from. Available options are OTP, QSPI FLASH and OQSPI FLASH.

7.2.8 Memory Header

This section describes the Memory Header programming settings (OTP, QSPI or OQSPI), available in devices.

7.2.8.1 Custom Memory Data



▲ Custom Memory Data

Custom Memory Data

☑ Write enable

OTP Data ○ No check ○ Check empty ○ Check if data match ● Skip if written

☑ Verify data

Input

○ CSV file CSV file path params\\custom_mem_data.csv ...

● Manual Edit data 00

Memory QSPI

Start address 0x 00000000

Data size 00

Figure 69: Custom Memory Data

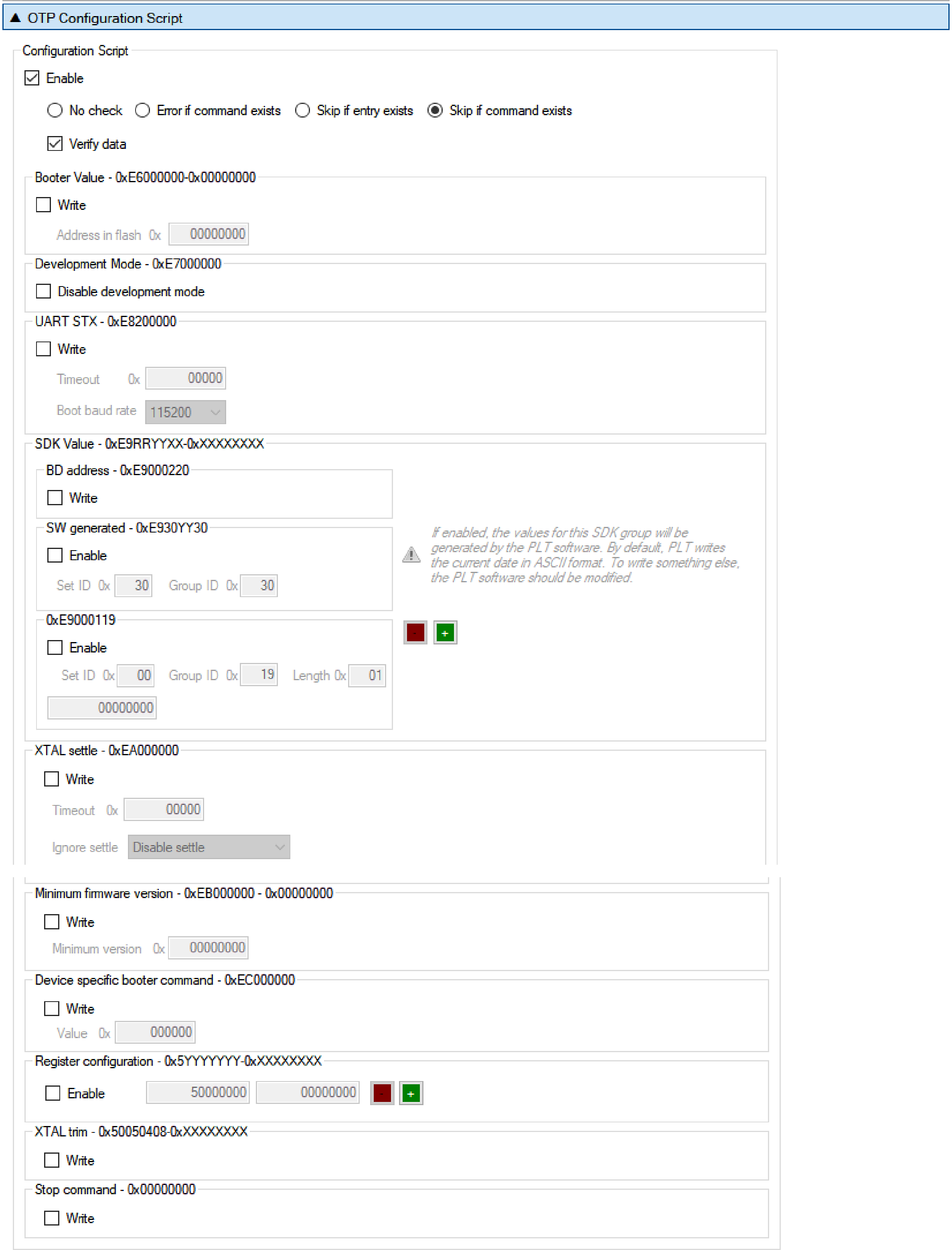
Table 55 describes the *Custom Memory* data test options. With this test, the user can write any data to any address to any available memory for the devices, such as OTP, register initialization at OTP-CS section and QSPI or OQSPI. Data input modes can be a CSV file or manually entered data.

Table 55: Custom Memory Data

<table>
<tr><th>Option</th><th colspan="2">Description</th></tr>
<tr><td>Write enable</td><td colspan="2">This option enables the Custom data programming.</td></tr>
<tr><td>Verify data</td><td colspan="2">When selected, the data written will be read back from the memory and will be compared to the original.</td></tr>
<tr><td>● No check
● Check empty
● Check if data match
● Skip if empty
(Only available when OTP memory, OTP TCS field or CSV file as input is selected)</td><td colspan="2">Memory protection options:
<b>No check:</b> No protection is enabled. PLT will attempt to burn the OTP memory without running any check.
<b>Check empty:</b> PLT will first check if the OTP memory to be burned is empty. If it is empty will burn it.
<b>Check if data match:</b> PLT will first check if the memory to be burned is empty. If it is not, it will compare the contents with the data to be burned. If the data are not the same the test will fail without making any changes to the memory. If the data are the same, PLT will not burn the memory and continue without error.
<b>Skip if written:</b> PLT will read the contents of OTP memory. If it contains any data, it will skip writing without producing any error.</td></tr>
<tr><td rowspan="2">● CSV file
● Manual data</td><td>CSV file path (CSV file)</td><td>Path to the CSV file containing data for each device discriminated using BD addresses. The CSV file format is described in Section 6.10.1.</td></tr>
<tr><td>Edit data (Manual data)</td><td>Hexadecimal data input of up to 256 bytes to burn. These data will be burned to all active DUTs.</td></tr>
<tr><td>Memory (Manual data mode)</td><td colspan="2">Memory type selection to burn the data. Available options are OTP, QSPI and OQSPI (only with <code>Manual data</code>).</td></tr>
<tr><td>Start address (available with Manual data mode)</td><td colspan="2">Memory address offset to begin burning the data. OTP user application valid address is 0x000 – 0xBFC.</td></tr>
</table>

Option	Description
Data size (available with Manual data mode)	The size of the memory data to burn. The size is number of bytes.

7.2.8.2 OTP Configuration Script



▲ OTP Configuration Script

Configuration Script

☑ Enable

○ No check ○ Error if command exists ○ Skip if entry exists ● Skip if command exists

☑ Verify data

Booter Value - 0xE6000000-0x00000000

□ Write

Address in flash 0x 00000000

Development Mode - 0xE7000000

□ Disable development mode

UART STX - 0xE8200000

□ Write

Timeout 0x 00000

Boot baud rate 115200

SDK Value - 0xE9RRYYXX-0xXXXXXXXX

BD address - 0xE9000220

□ Write

SW generated - 0xE930YY30

□ Enable

Set ID 0x 30 Group ID 0x 30

If enabled, the values for this SDK group will be generated by the PLT software. By default, PLT writes the current date in ASCII format. To write something else, the PLT software should be modified.

0xE9000119

□ Enable

Set ID 0x 00 Group ID 0x 19 Length 0x 01

00000000

XTAL settle - 0xEA000000

□ Write

Timeout 0x 00000

Ignore settle Disable settle

Minimum firmware version - 0xEB000000 - 0x00000000

□ Write

Minimum version 0x 00000000

Device specific booter command - 0xEC000000

□ Write

Value 0x 000000

Register configuration - 0x5YYYYYYY-0xXXXXXXXX

□ Enable 50000000 00000000

XTAL trim - 0x50050408-0xXXXXXXXX

□ Write

Stop command - 0x00000000

□ Write

Figure 70: OTP Configuration Script

The OTP CS is an area used for programming system registers with values that are defined during production testing, store a trim value for the application software or define UART time out time during booting. It will be executed by the booter to prepare and initialize the system prior to the CPU start running application code from the non-volatile memory (QSPI Flash, OSPI Flash, etc.).

It is a table, of 32-bit entries, 256 spaces deep making a total size of 1024 bytes.

The OTP Configuration Script comprises ten different **commands**. The CS begins with the **Start** command followed by a set of commands that refer to certain things, like register configurations, trim values, etc. The CS is ended with the **Stop** command. Empty entries between commands are not allowed. If an empty entry is found, then all the following commands should be discarded. The commands are depicted in Table 56.

Table 56: OTP Configuration Script Commands

#	Command	Command Header	# of Words following
1	Start of CS	0xA5A5A5A5	0
2	Register Configuration	Register Address	1
3	Booter Value	0xE6000000	1
4	Development Mode Disable	0xE7000000	0
5	Boot UART Configuration	0xE8YXXXXX	0
6	SDK Trim Value	0xE9RRYYXX	1 – 0xFF
7	XTAL configuration	0xEAXXXXXX	0
8	Minimum FW version	0xEB000000	1
9	Device specific booter command	0xECXXXXXX	0
10	Stop of CS	0x00000000	0

In Table 57 the PLT available parameters for programming the OTP CS are explained.

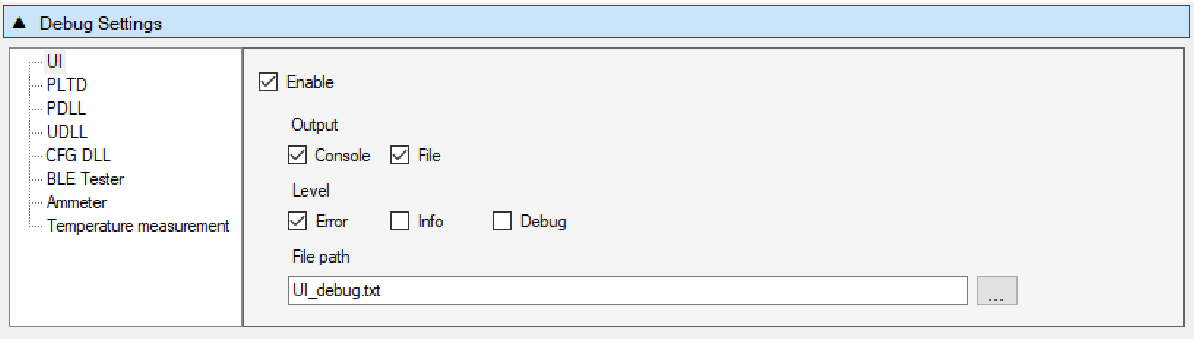
Table 57: OTP Configuration Script

Option	Command	Description
Enable	-	When selected, OTP Configuration Script options will be enabled.
● No check ● Error if command exists ● Skip if entry exists ● Skip if command exists	-	OTP CS memory protection options: ● **No check**: Check disabled. ● **Error if command exists**: Returns error if the command of an entry is already written in the DUT, no matter if the data are the same. ● **Skip if entry exists**: Skip writing an entry without error if the command and the data are already written in the DUT. If same command is found with different data an error will be returned. ● **Skip if command exists**: Skip writing without returning error if the command in the DUT OTP CS is already written, no matter what the data are (same or different).
Verify	-	
Booter value	0xE6000000	The Booter Value command consists of two 32-bit words. The first word is equal to 0xE6000000, indicating that an address pointing to the Flash product header in flash follows. The second 32-bits word is the address at 0xYYYYYYYY. If the Booter Value command exists multiple times within the CS, the booter will execute the command as many times as the command exists resulting in using only the last encountered value.
Development mode	0xE7000000	The Development Mode command consists of one 32-bit word which is equal to 0xE7000000, instructing the booter to disable the development mode. There is no limitation for the number of entries the Development

		Mode command exists in the CS, thus the first entry should only be considered by the booter.
UART STX	0xE8YXXXXX	The UART STX command consists of one 32-bit word which is equal to 0xE8YXXXXX. The XXXXX is used to program the selected STX timeout in multiples of 100 us. So i.e., 0xE8000028 is 40x100 μs = 4 ms. The Y is used to select one of the predefined UART baud rates.
SDK value	0xE9RRYYXX	The Trim Value command consists of one 32-bit word which is equal to 0x**E9RRYYXX** indicating that the next word is a value stored during production testing, followed by one (or more) 32-bit word(s) which represent the value(s). More specifically: ● **E9**: indicates that the following word(s) are not to be stored to registers but will be used by the SDK SW ● **RR:** is an index (Set ID) and can be used for different sets of parameters by the SW application. One Set ID comprises 256 Group IDs. ○ RR = 0xFF is reserved for customer use only. ○ RR = 0x00 will be used for Renesas parameters ○ RR = 0x01 – 0xEF are reserved for future use of Renesas parameters ● **YY**: indicates that 0xYY number of words will follow ● **XX**: is an index (Group ID) and can be used for different parameters by the SW application. The maximum number of Group IDs per Set ID is 256. The Group IDs will be used for production or other data, e.g., BD address, preferred value, or function coefficients.
BD address	0xE9000220	If this option is selected PLT will burn the BD address to the OTP CS with command 0xE9000220, which is the appropriate Group ID for SDK BD address.
SW generated	0xE930YY30	If enabled, PLT software will burn the current date in ASCII format in PLT OTP CS. The default Set ID is set to 0x30, and default Group ID is set to 0x30. These can be changed. Additionally, PLT software can be modified to write something different than then current date.
XTAL settle	0xEA000000	The XTAL settle command consists of one 32-bit word, which is equal to 0xEAYXXXXX. It is used by the booter to configure the XTAL settling times. Settling times are in multiples of 100us.
Minimum firmware version	0xEB000000	The Minimum FW version command consists of one 32-bit word which is equal to 0xEB000000, followed by another 32bit word specifying the minimum FW version. The minimum FW version is evaluated by the booter and sets the minimum acceptable firmware version that the device will boot.
Device specific booter command	0xECXXXXXX	The Device Specific Booter Command consists of one 32-bit word, which is equal to 0xECXXXXXX. It is used by the booter to configure device specific settings. Each bit or a number of bits of XXXXXX part of the command can be freely defined and used by the booter.
Register Configuration		If this option is enabled, user can program in OTP CS a value that will be set to a specific register during boot. It contains: • A 32-bit word containing an address of an existing register • A 32-bit word containing the data value of the register These are always in pairs with the address sitting in even memory addresses.
Register configuration	0x5YYYYYYY	The Register Configuration command consist of one 32-bit word containing an address of an existing register within the available memory map range,

		followed by a second 32-bit word containing the data value of the register. These words are always in pairs. If the Register configuration command for a specific register exists multiple times within the CS, the booter will execute the command as many times as the command exists resulting with the specific register having the value the last command instructed. The SDK should also make sure that the last value is considered as valid.
XTAL trim	0x50050408	The XTAL trim calibration value found in Section 7.2.6.3 will be programmed in OTP CS. The value found in the XTAL trim calibration process will be applied to register 0x50050408.
Stop command	0x00000000	The Stop of CS command consists of one 32-bit word (0x00000000), designating that the configuration script has reached the end and execution should be terminated. It should be noted that there is no limitation for the number of entries the Stop command exists in the CS, thus the first entry is only considered by the booter. Apart from the minimum firmware version entries that is evaluated by the booter at all times, all other commands after the first entry of the Stop command are discarded.

7.2.9 Debug Settings



▲ Debug Settings

- UI
- PLTD
- PDLL
- UDLL
- CFG DLL
- BLE Tester
- Ammeter
- Temperature measurement

☑ Enable

Output

☑ Console ☑ File

Level

☑ Error □ Info □ Debug

File path

UI_debug.txt

Figure 71: Debug Settings

Table 58 describes the available options for the *Debug Settings*. Debug messages are available in all PLT software blocks shown in Figure 18.

Note: Printing debug information may introduce system delay and thus some tests may fail due to time out expirations. We suggest having debug information disabled in all software blocks and only partially enable when there is a real need for it. From PLT v4.0 and onwards, this system delay has been almost eliminated as debug print messages are printed from a lower priority queue. It is safer, but it is still suggested to have the debug prints disabled.

Table 58: Debug Settings

Option	Description
Enable	Enable debug message prints for the selected library or UI.
Output - Console	Sends the debug messages to the `stdio` output. The PLT CLI does not support this option. If enabled, debug messages will be redirected to the equivalent files.
Output - File	Save the debug messages to a file.
Level - Error	Enable error debug level messages. All debug print messages marked as error will be printed.
Level - Info	Enable info debug level messages. All debug print messages marked as info will be printed.

Option	Description
Level - Debug	Enable low level debug level messages. All low-level debug print messages will be printed.
File path	Select the file that the debug messages will be saved. The file should exist; otherwise, it should be created manually. Used only when the option Output - File is selected.

7.2.10 Security

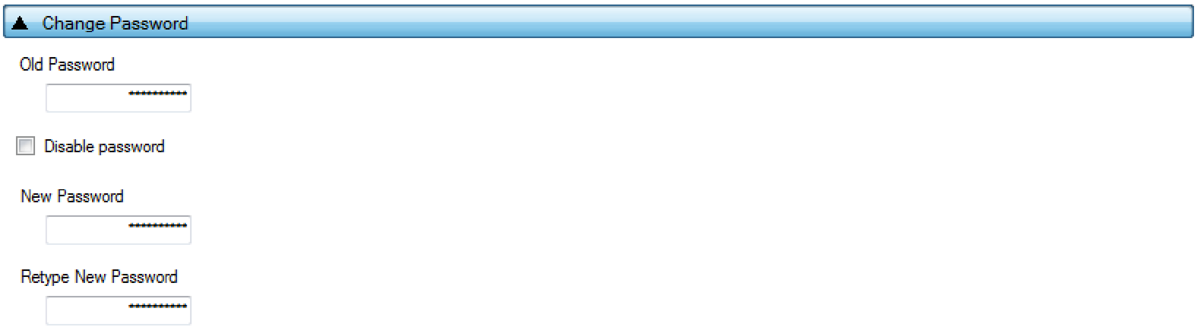


Figure 72: Security

In this field, a password can be set to protect specific tool actions, such as:

- Opening the CFG PLT or the GUI PLT application.
- Closing the CFG PLT or the GUI PLT application.
- Opening or refreshing configuration settings in the GUI PLT application.
- Opening the settings menu in the GUI PLT application.

Table 59 describes the available options for the *Security* Options.

Table 59: Security Options

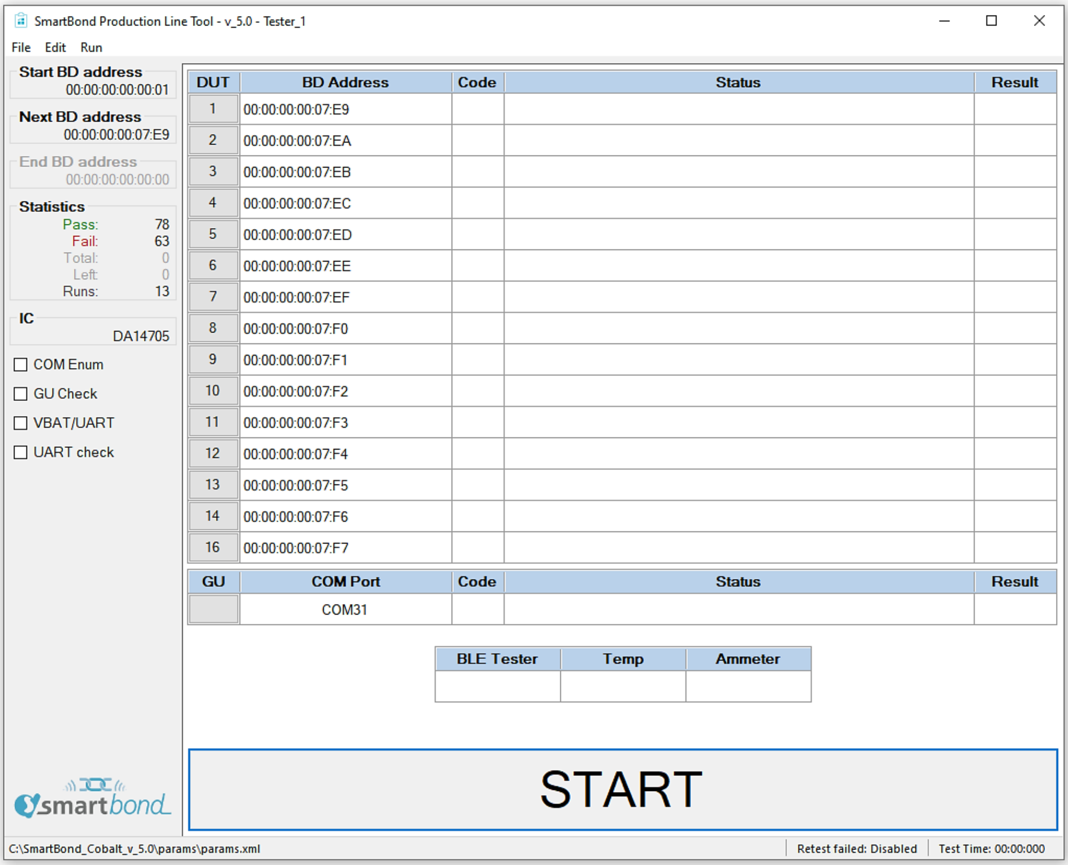
Option	Description
Old Password	Type the current password to enable changing of the following fields.
Disable Password	This option will disable the password usage.
New Password	Type a new password.
Retype New Password	Verify the new password.

7.3 GUI PLT Application

The GUI PLT (`SmartBond_PLT_GUI.exe`) is a Graphical User Interface application that performs the device validation and programming process. At the same time, it allows the users to monitor the entire procedure in detail. The GUI PLT uses the same XML file configured from CFG PLT as described in Section 7.2.

Note: If a change is made to the XML file from the CFG PLT, then the GUI PLT settings should be refreshed as described in Table 60.

Figure 73 shows the initial screen of the GUI PLT, which is described in Table 60.



Start BD address: 00:00:00:00:00:01

Next BD address: 00:00:00:00:07:E9

End BD address: 00:00:00:00:00:00

Statistics
Pass: 78
Fail: 63
Total: 0
Left: 0
Runs: 13

IC: DA14705

□ COM Enum

□ GU Check

□ VBAT/UART

□ UART check

DUT	BD Address	Code	Status	Result
1	00:00:00:00:07:E9			
2	00:00:00:00:07:EA			
3	00:00:00:00:07:EB			
4	00:00:00:00:07:EC			
5	00:00:00:00:07:ED			
6	00:00:00:00:07:EE			
7	00:00:00:00:07:EF			
8	00:00:00:00:07:F0			
9	00:00:00:00:07:F1			
10	00:00:00:00:07:F2			
11	00:00:00:00:07:F3			
12	00:00:00:00:07:F4			
13	00:00:00:00:07:F5			
14	00:00:00:00:07:F6			
16	00:00:00:00:07:F7			

GU	COM Port	Code	Status	Result
	COM31			

BLE Tester	Temp	Ammeter

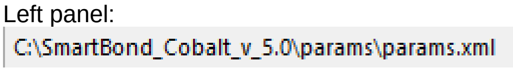
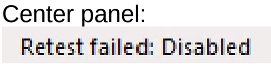
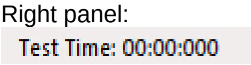
C:\SmartBond_Cobalt_v_5.0\params\params.xml | Retest failed: Disabled | Test Time: 00:00:000

Figure 73: GUI PLT Main Screen

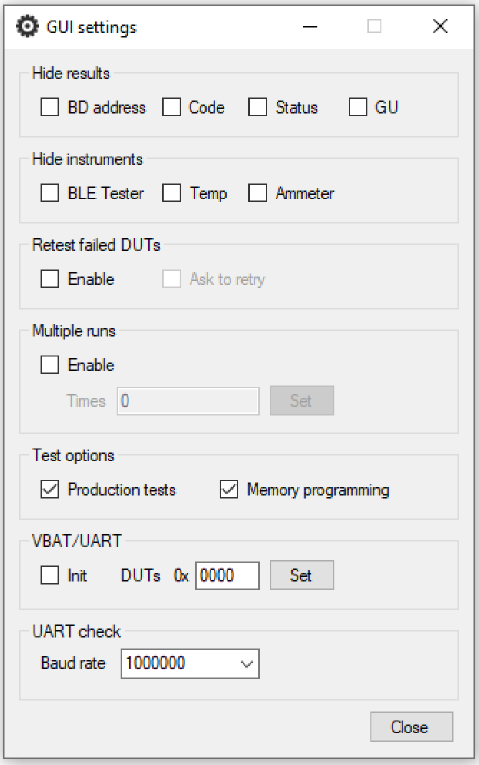
Table 60: GUI PLT Main Screen Description

Options	Description
File options	
File > Open XML file	Opens a new XML file and loads its settings. The full path of the new XML file is shown at the bottom end of the screen.
File > Refresh XML file	Reloads the settings from the XML file and initializes itself with the new settings.
File > Open CSV file	Contains a list with all the available CSV files to open.
File > Exit	Exits the GUI PLT application.

Options	Description
Edit options	
Edit > Settings	Opens the GUI PLT Settings window (Section 7.3.1).
Run options	
Run > Run Configuration PLT	Opens the CFG PLT application.
Left Column options	
Start BD Address	The BD address the PLT session started with, as described in Section 7.2.4.
Next BD Address	The BD address that will be used on the BD address assignment for the next run as described in Section 7.2.4.
End BD Address	The BD address the PLT session ends with as described in Section 7.2.4. This option is available only when *Range mode* is enabled.
Statistics	This field holds statistics for each PLT session. Table 25 describes the *Statistics* field.
IC	The selected IC of the PLT.
COM Enum	If this checkbox is enabled, then the START button initiates the automatic Window COM port enumeration for the DUT.
GU Check	If this checkbox is enabled, then the START button initiates the automatic Window COM port enumeration for the Golden Unit.
VBAT/UART	If this checkbox is enabled, then the START button will enable the VBAT and UART for the DUTs selected under *VBAT/UART* in Table 61.
UART check	If this checkbox is enabled, then the START button initiates the UART check procedure for the DUTs with a specified Baud rate set from the user through the GUI PLT Settings (Section 7.3.1). During this test, 1000 packets will be sent, received back, and checked for errors. The packets contain 252 bytes. **Note:** Before any UART transfer begins, PLT ill download the production test firmware to the active DUTs.
Center screen options	
DUT panel	Shows the following fields for each DUT: ● **DUT:** DUT connector number on the PLT hardware. This field is also a button that opens the Log file for the specific DUT. ● **BD Address:** BD address assigned to the DUT. ● **Code:** Real-time status as a PLTD DLL special code described in Ref. [1]. ● **Description:** A brief description of the status code. ● **Result:** Simplified color-coded status showing the progress per DUT.

Options	Description
GU panel	Shows the following fields for the Golden Unit: ● **GU:** A button that opens the Golden Unit Log file. ● **COM Port:** The COM port assigned to the Golden Unit. ● **Code:** Real-time status as a PLTD DLL special code described in Ref. [1]. ● **Status:** A brief description of the status code. ● **Result:** Simplified color-coded status showing the progress of the GU.
Instrument panel	This field shows a simplified color-coded status is shown for each of the instruments (**BLE Tester**, **Temp, Ammeter** and **Voltmeter**), if they are enabled.
START button	If one of the options *COM Enum*, *GU Check, VBAT/UART* or *UART check* is enabled, then selecting the *START* button will initiate the chosen test. If no option is selected, selecting the *START* button initiates the production procedure. **Note:** To select and press the `Start Button` press the `space-bar` key. The `Start Button` can only be pressed with the mouse (or use the `'f'` key as a shortcut), after the selected procedure is finished, in order to return to main screen. This is to avoid pressing the `Start Button` and starting a new test procedure, by mistake.
Bottom of the main screen	
Left panel:  C:\SmartBond_Cobalt_v_5.0\params\params.xml	Shows the full path of the XML file that is currently used.
Center panel:  Retest failed: Disabled	Shows if the re-test option in Section 7.3.1 is enabled.
Right panel:  Test Time: 00:00:000	This timer starts counting when the START button is pressed and runs until the PLT returns to its idle state, showing the approximate duration of the tests.

7.3.1 GUI PLT Settings



GUI settings

Hide results

□ BD address □ Code □ Status □ GU

Hide instruments

□ BLE Tester □ Temp □ Ammeter

Retest failed DUTs

□ Enable □ Ask to retry

Multiple runs

□ Enable

Times 0 Set

Test options

☑ Production tests ☑ Memory programming

VBAT/UART

□ Init DUTs 0x 0000 Set

UART check

Baud rate 1000000

Close

Figure 74: GUI PLT Settings

Figure 74 shows the *GUI PLT settings* window. In this window, various graphic options and features can be set as described in Table 61.

Table 61: GUI PLT Settings

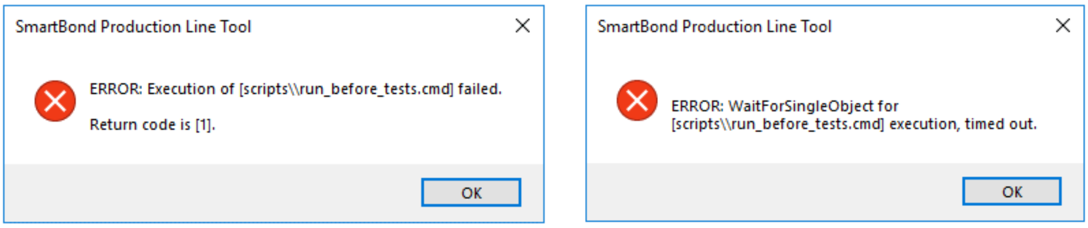
Field	Option	Description
Hide results	BD address	This option will hide the *BD address* column in the DUT panel of the GUI PLT.
	Code	This option will hide the *Code* column in the DUT panel of the GUI PLT.
	Status	This option will hide the *Status* column in the DUT panel of the GUI PLT.
	GU	This option will hide the *GU* column in the DUT panel of the GUI PLT.
Hide instruments	BLE Tester	This option will hide the *BLE Tester* column in the GU panel of the GUI PLT.
	Temp	This option will hide the *Temp* column in the GU panel of the GUI PLT.
	Ammeter	This option will hide the *Ammeter* column in the GU panel of the GUI PLT.
Retest failed DUTs	Enable	If this option is enabled, any DUT that failed during the main procedure will immediately re-run the tests having the exact same options including the *BD address* assigned to it. This option is the exact same option as *Re-test failed DUTs* in Section 7.2.3.2.
	Ask to retry	This option will show a message asking to do a re-test in case any DUT failed. If this option is disabled, the re-testing will be done automatically.
Multiple Runs	Enable	By enabling this option, the GUI PLT will perform multiple procedures without any delay between them. This is used for only for evaluation.
	Times	The number of times to run.

Field	Option	Description
Test Options	Production tests	Enables /Disables the production test procedure. This is the same option as *Production tests* in Section 7.2.3.2.
	Memory programming	Enables /Disables the production test procedure. This is the same option as *Memory programming* in Section 7.2.3.2.
VBAT/UART	Init	If this option is enabled, the PLT hardware will be reset before enabling the DUTs. This option is enabled only when *VBAT/UART* in the main screen is enabled.
	DUTs	Bitwise DUT set/reset for each of the 16 DUTs using a 16-bit hexadecimal value. 089 **Example:** To enable only DUTs 1, 2, 15 and 16 use "C003" (1100 0000 0000 0011 = 0xC003).
UART check	Baud rate	Sets the Baud rate for the UART check test.

7.3.2 Running the GUI PLT and Executing Tests

The GUI PLT starts the test procedure when users click the *START* button. Before initiating the test procedure, the GUI PLT will assign BD addresses to the active DUTs and check for any wrong configuration parameters.

If `Run scripts before testing starts` is enabled, PLT will execute the selected script/executable, and wait until it finishes or times out, depending on the selections made in Section 7.2.3.2. If the script/executable has returned on time, PLT will check the return code. Values from 0 to 100 indicate a successful completion. Negative values or values larger than 100 indicate an error. In the case of an error (either time out or error returned result), a pop-up message will appear indicating the return code and the test procedure will not start.


SmartBond Production Line Tool

ERROR: Execution of [scripts\\run_before_tests.cmd] failed.

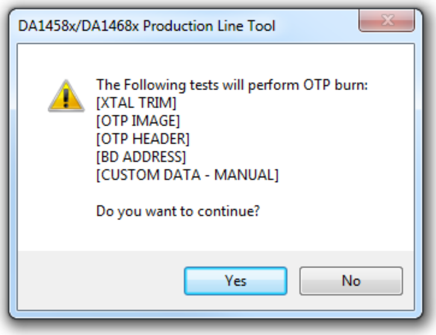
Return code is [1].

SmartBond Production Line Tool

ERROR: WaitForSingleObject for [scripts\\run_before_tests.cmd] execution, timed out.

Figure 75: GUI PLT - Erroneous Messages in "Run Scripts Before Testing Starts"

If any OTP burning test is scheduled, a pop-up message will inform the user and prompt to continue (Figure 76).

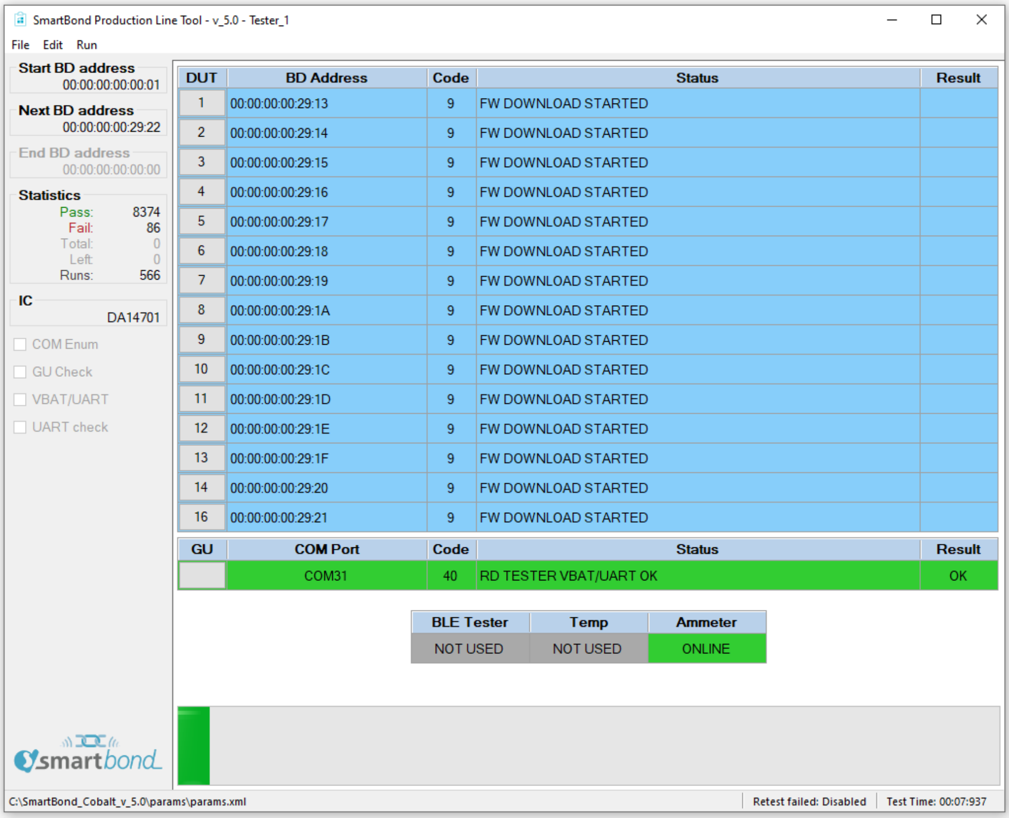

DA1458x/DA1468x Production Line Tool

The Following tests will perform OTP burn:
[XTAL TRIM]
[OTP IMAGE]
[OTP HEADER]
[BD ADDRESS]
[CUSTOM DATA - MANUAL]

Do you want to continue?

Figure 76: GUI PLT OTP Burn Warning Message

Click **Yes** to start the testing procedure. PLT updates the status of the procedure for each DUT and the Golden Unit (Figure 77). The *START* button is replaced by a progress bar indicating the progress of the tests.



SmartBond Production Line Tool - v_5.0 - Tester_1

Start BD address 00:00:00:00:00:01

Next BD address 00:00:00:00:29:22

End BD address 00:00:00:00:00:00

Statistics

Pass:	8374
Fail:	86
Total:	0
Left:	0
Runs:	566

IC DA14701

□ COM Enum

□ GU Check

□ VBAT/UART

□ UART check

DUT	BD Address	Code	Status	Result
1	00:00:00:00:29:13	9	FW DOWNLOAD STARTED	
2	00:00:00:00:29:14	9	FW DOWNLOAD STARTED	
3	00:00:00:00:29:15	9	FW DOWNLOAD STARTED	
4	00:00:00:00:29:16	9	FW DOWNLOAD STARTED	
5	00:00:00:00:29:17	9	FW DOWNLOAD STARTED	
6	00:00:00:00:29:18	9	FW DOWNLOAD STARTED	
7	00:00:00:00:29:19	9	FW DOWNLOAD STARTED	
8	00:00:00:00:29:1A	9	FW DOWNLOAD STARTED	
9	00:00:00:00:29:1B	9	FW DOWNLOAD STARTED	
10	00:00:00:00:29:1C	9	FW DOWNLOAD STARTED	
11	00:00:00:00:29:1D	9	FW DOWNLOAD STARTED	
12	00:00:00:00:29:1E	9	FW DOWNLOAD STARTED	
13	00:00:00:00:29:1F	9	FW DOWNLOAD STARTED	
14	00:00:00:00:29:20	9	FW DOWNLOAD STARTED	
16	00:00:00:00:29:21	9	FW DOWNLOAD STARTED	

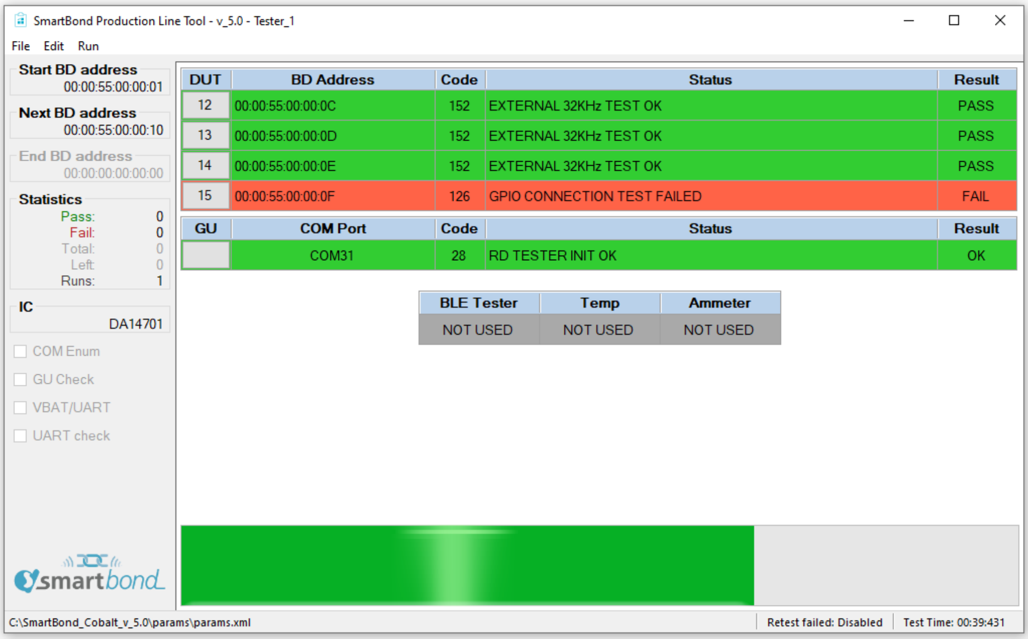
GU	COM Port	Code	Status	Result
	COM31	40	RD TESTER VBAT/UART OK	OK

BLE Tester	Temp	Ammeter
NOT USED	NOT USED	ONLINE

C:\SmartBond_Cobalt_v_5.0\params\params.xml | Retest failed: Disabled | Test Time: 00:07:937

Figure 77: GUI PLT Testing (1 of 2)

If an error in a DUT is found (Figure 78), PLT will show the status code, a brief description of the error and the color of the DUT’s status line will turn red. The DUT number button can be pressed anytime to access the DUT Log File (Section 7.3.4)to get more details about the parameters used, calculated values and the reason for failure in the case of an error.



SmartBond Production Line Tool - v_5.0 - Tester_1

File Edit Run

Start BD address 00:00:55:00:00:01

Next BD address 00:00:55:00:00:10

End BD address 00:00:00:00:00:00

Statistics	
Pass:	0
Fail:	0
Total:	0
Left:	0
Runs:	1

IC DA14701

□ COM Enum
□ GU Check
□ VBAT/UART
□ UART check

DUT	BD Address	Code	Status	Result
12	00:00:55:00:00:0C	152	EXTERNAL 32KHz TEST OK	PASS
13	00:00:55:00:00:0D	152	EXTERNAL 32KHz TEST OK	PASS
14	00:00:55:00:00:0E	152	EXTERNAL 32KHz TEST OK	PASS
15	00:00:55:00:00:0F	126	GPIO CONNECTION TEST FAILED	FAIL

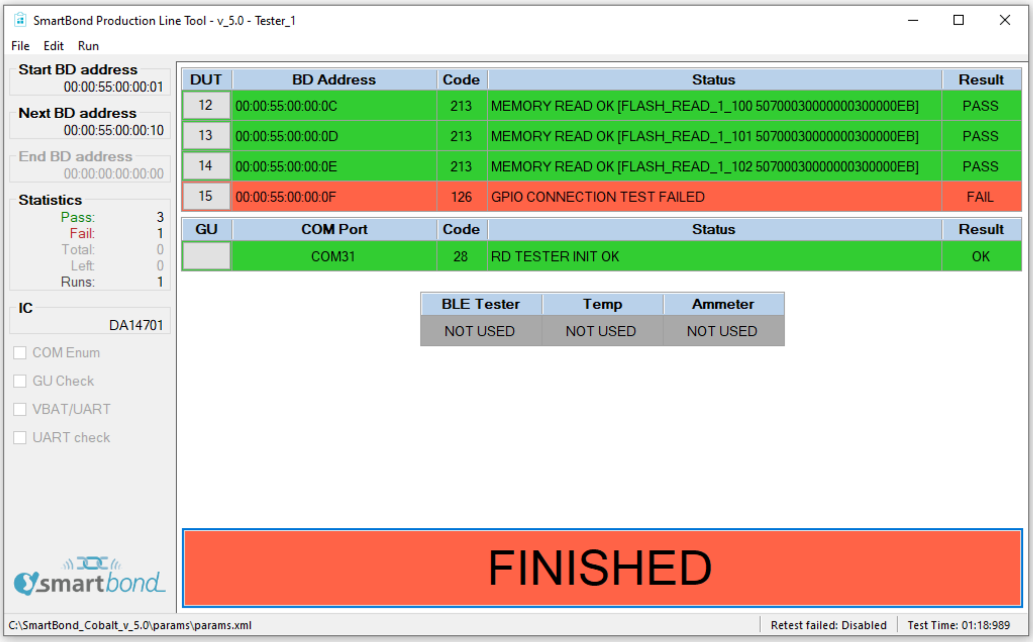
GU	COM Port	Code	Status	Result
	COM31	28	RD TESTER INIT OK	OK

BLE Tester	Temp	Ammeter
NOT USED	NOT USED	NOT USED

C:\SmartBond_Cobalt_v_5.0\params\params.xml | Retest failed: Disabled | Test Time: 00:39:431

Figure 78: GUI PLT Testing (2 of 2)

After the testing procedure is completed (Figure 79), the progress bar shows *FINISHED* and the color turns red if any DUT has failed, otherwise it is green. If there is an error and the *Retest failed DUTs* and *Ask to retry* options are enabled, a message will appear asking if the user would like to retest the failed DUTs, as shown in Figure 80. When the GUI PLT performs a retest run, all options (including the BD addresses) remain the same and only tests that failed are retested. At this time, the CSV File (Section 7.3.5) and all the DUT Log Files (Section 7.3.4) are updated.



SmartBond Production Line Tool - v_5.0 - Tester_1

File Edit Run

Start BD address 00:00:55:00:00:01

Next BD address 00:00:55:00:00:10

End BD address 00:00:00:00:00:00

Statistics	
Pass:	3
Fail:	1
Total:	0
Left:	0
Runs:	1

IC DA14701

□ COM Enum
□ GU Check
□ VBAT/UART
□ UART check

DUT	BD Address	Code	Status	Result
12	00:00:55:00:00:0C	213	MEMORY READ OK [FLASH_READ_1_100 50700030000000300000EB]	PASS
13	00:00:55:00:00:0D	213	MEMORY READ OK [FLASH_READ_1_101 50700030000000300000EB]	PASS
14	00:00:55:00:00:0E	213	MEMORY READ OK [FLASH_READ_1_102 50700030000000300000EB]	PASS
15	00:00:55:00:00:0F	126	GPIO CONNECTION TEST FAILED	FAIL

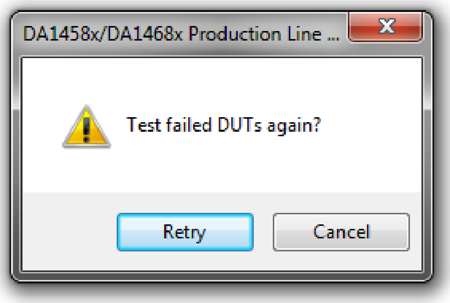
GU	COM Port	Code	Status	Result
	COM31	28	RD TESTER INIT OK	OK

BLE Tester	Temp	Ammeter
NOT USED	NOT USED	NOT USED

FINISHED

C:\SmartBond_Cobalt_v_5.0\params\params.xml | Retest failed: Disabled | Test Time: 01:18:989

Figure 79: GUI PLT Tests Finished



Test failed DUTs again?

Figure 80: GUI PLT Retry Failed DUTs Message

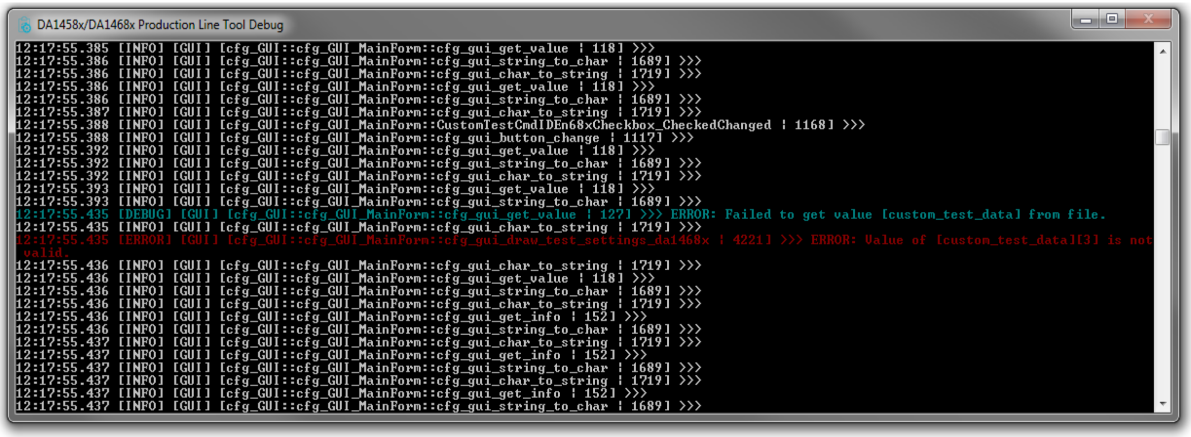
If the DUT fails again, after the retest has finished the GUI PLT will remain in the *FINISHED* screen (Figure 79) with the *FINISHED* button shown in red.

If `Run scripts when testing is finished` is enabled, pressing the finished button will execute the selected script/executable. As with `Run scripts before testing starts`, PLT will wait until it finishes or times out, depending on the selections made in Section 7.2.3.2. If the script/executable has returned on time, PLT will check the return code. Zero value indicates a successful completion. Any other value is considered an error. In the case of an error (either time out or error result), a pop-up message will appear indicating the return code.

7.3.3 Debug Console

Section 7.2.8.1 shows the debug settings for all PLT applications including the GUI PLT. If at least one debug session is enabled with the output set to *Console,* the GUI PLT will open a new console window showing the desired debug information.

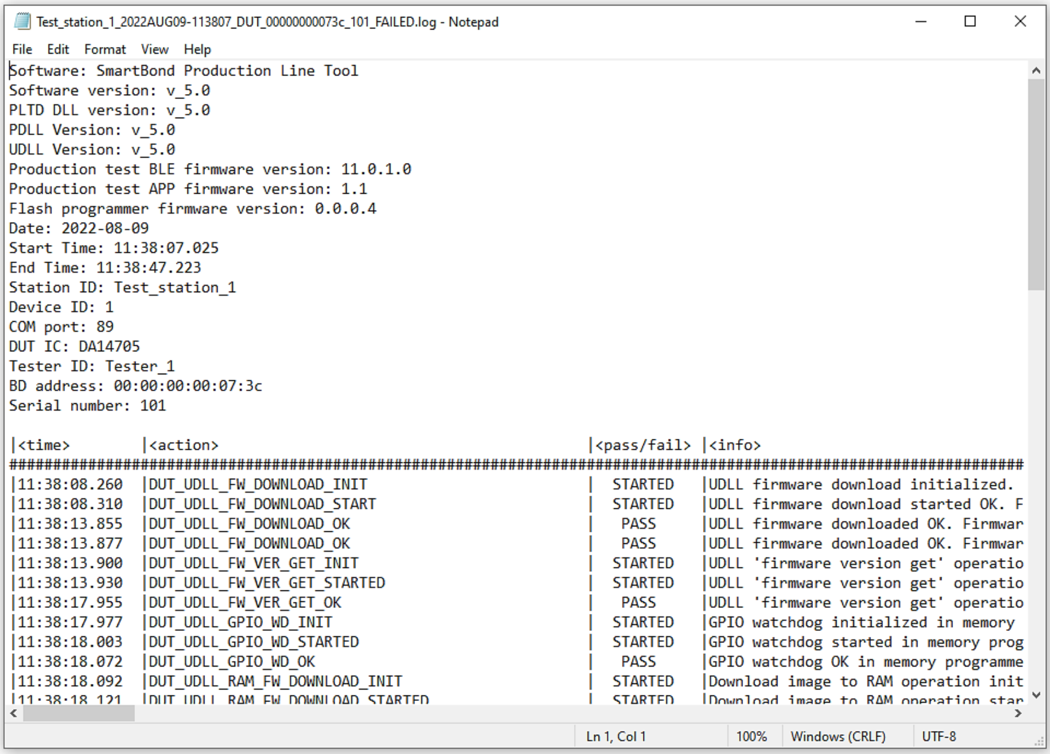
Figure 81 shows an example of the *Debug Console*. Depending on the type of the message, a different color is used: *DEBUG* messages are light blue, *INFO* messages are white and *ERROR* messages are red.



```
12:17:55.385 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_value ¦ 118] >>>
12:17:55.386 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_string_to_char ¦ 1689] >>>
12:17:55.386 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_char_to_string ¦ 1719] >>>
12:17:55.386 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_value ¦ 118] >>>
12:17:55.386 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_string_to_char ¦ 1689] >>>
12:17:55.387 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_char_to_string ¦ 1719] >>>
12:17:55.388 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::CustomTestCmdIDEn68xCheckbox_CheckedChanged ¦ 1168] >>>
12:17:55.388 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_button_change ¦ 1117] >>>
12:17:55.392 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_value ¦ 118] >>>
12:17:55.392 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_string_to_char ¦ 1689] >>>
12:17:55.392 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_char_to_string ¦ 1719] >>>
12:17:55.393 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_value ¦ 118] >>>
12:17:55.393 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_string_to_char ¦ 1689] >>>
12:17:55.435 [DEBUG] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_value ¦ 127] >>> ERROR: Failed to get value [custom_test_data] from file.
12:17:55.435 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_char_to_string ¦ 1719] >>>
12:17:55.435 [ERROR] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_draw_test_settings_da1468x ¦ 4221] >>> ERROR: Value of [custom_test_data][3] is not
 valid.
12:17:55.436 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_char_to_string ¦ 1719] >>>
12:17:55.436 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_value ¦ 118] >>>
12:17:55.436 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_string_to_char ¦ 1689] >>>
12:17:55.436 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_char_to_string ¦ 1719] >>>
12:17:55.436 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_info ¦ 152] >>>
12:17:55.436 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_string_to_char ¦ 1689] >>>
12:17:55.437 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_char_to_string ¦ 1719] >>>
12:17:55.437 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_info ¦ 152] >>>
12:17:55.437 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_string_to_char ¦ 1689] >>>
12:17:55.437 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_char_to_string ¦ 1719] >>>
12:17:55.437 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_get_info ¦ 152] >>>
12:17:55.437 [INFO] [GUI] [cfg_GUI::cfg_GUI_MainForm::cfg_gui_string_to_char ¦ 1689] >>>
```

Figure 81: Debug Console

7.3.4 DUT Log File


Test_station_1_2022AUG09-113807_DUT_00000000073c_101_FAILED.log - Notepad
File
Edit
Format
View
Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

```
Software: SmartBond Production Line Tool
Software version: v_5.0
PLTD DLL version: v_5.0
PDLL Version: v_5.0
UDLL Version: v_5.0
Production test BLE firmware version: 11.0.1.0
Production test APP firmware version: 1.1
Flash programmer firmware version: 0.0.0.4
Date: 2022-08-09
Start Time: 11:38:07.025
End Time: 11:38:47.223
Station ID: Test_station_1
Device ID: 1
COM port: 89
DUT IC: DA14705
Tester ID: Tester_1
BD address: 00:00:00:00:07:3c
Serial number: 101

|<time>        |<action>                                         |<pass/fail> |<info>
###################################################################################################################
|11:38:08.260  |DUT_UDLL_FW_DOWNLOAD_INIT                        |  STARTED   |UDLL firmware download initialized.
|11:38:08.310  |DUT_UDLL_FW_DOWNLOAD_START                       |  STARTED   |UDLL firmware download started OK. F
|11:38:13.855  |DUT_UDLL_FW_DOWNLOAD_OK                          |   PASS     |UDLL firmware downloaded OK. Firmwar
|11:38:13.877  |DUT_UDLL_FW_DOWNLOAD_OK                          |   PASS     |UDLL firmware downloaded OK. Firmwar
|11:38:13.900  |DUT_UDLL_FW_VER_GET_INIT                         |  STARTED   |UDLL 'firmware version get' operatio
|11:38:13.930  |DUT_UDLL_FW_VER_GET_STARTED                      |  STARTED   |UDLL 'firmware version get' operatio
|11:38:17.955  |DUT_UDLL_FW_VER_GET_OK                           |   PASS     |UDLL 'firmware version get' operatio
|11:38:17.977  |DUT_UDLL_GPIO_WD_INIT                            |  STARTED   |GPIO watchdog initialized in memory
|11:38:18.003  |DUT_UDLL_GPIO_WD_STARTED                         |  STARTED   |GPIO watchdog started in memory prog
|11:38:18.072  |DUT_UDLL_GPIO_WD_OK                              |   PASS     |GPIO watchdog OK in memory programme
|11:38:18.092  |DUT_UDLL_RAM_FW_DOWNLOAD_INIT                    |  STARTED   |Download image to RAM operation init
|11:38:18.121  |DUT_UDLL_RAM_FW_DOWNLOAD_STARTED                 |  STARTED   |Download image to RAM operation star
```

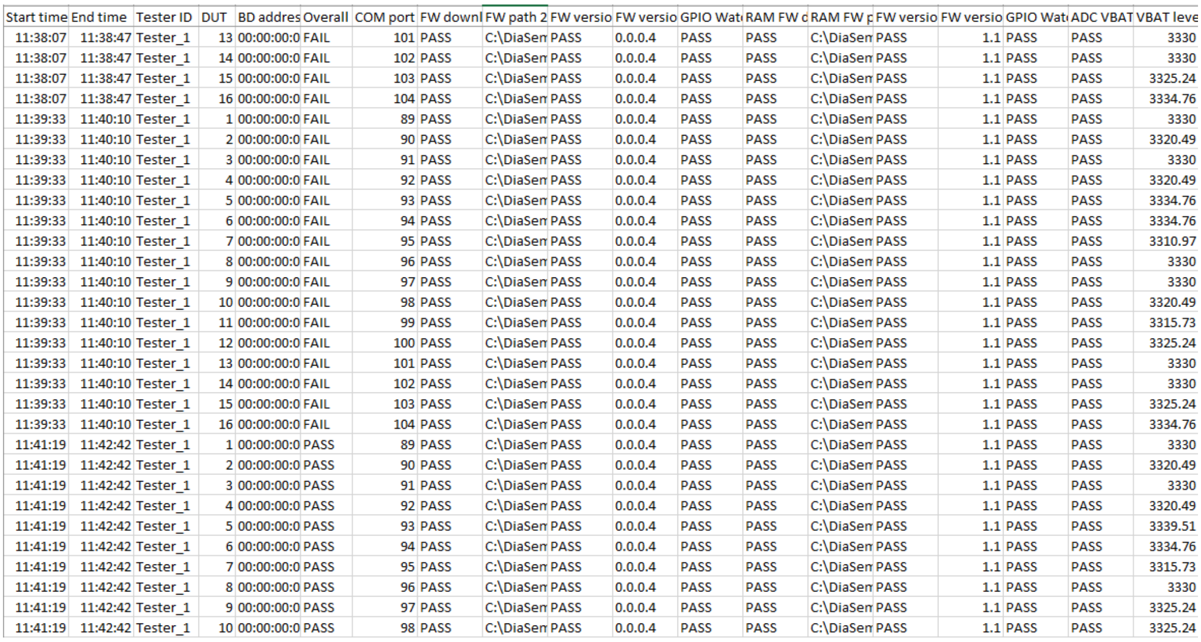
Figure 82: DUT Log File

Figure 82 shows a Log file generated for DUT1 during testing.

In the first few lines of the log, a header is created giving vital information about the PLT hardware and the software. It includes the firmware and software version, the station name and test dates and times. It also holds information about the DUT, such as the connector number in the PLT hardware, the BD address assigned to it and the Windows COM port. For the DUTs that have failed, the log file is renamed with the word "_FAILED" at the end for easier retrieval.

The Log file is created at the beginning of each test, containing only the header and all information available at the time of creation. As the device testing progresses, the status of each test is written at the end of the log file, including information about the DUT and a timestamp of the event. After the tests finish the header is updated with the end time of the test and the firmware versions, which were retrieved during testing.

7.3.5 CSV File



Start time	End time	Tester ID	DUT	BD addres	Overall	COM port	FW downl	FW path 2	FW versio	FW versio	GPIO Wat	RAM FW c	RAM FW p	FW versio	FW versio	GPIO Wat	ADC VBAT	VBAT leve
11:38:07	11:38:47	Tester_1	13	00:00:00:0	FAIL	101	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:38:07	11:38:47	Tester_1	14	00:00:00:0	FAIL	102	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:38:07	11:38:47	Tester_1	15	00:00:00:0	FAIL	103	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3325.24
11:38:07	11:38:47	Tester_1	16	00:00:00:0	FAIL	104	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3334.76
11:39:33	11:40:10	Tester_1	1	00:00:00:0	FAIL	89	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:39:33	11:40:10	Tester_1	2	00:00:00:0	FAIL	90	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3320.49
11:39:33	11:40:10	Tester_1	3	00:00:00:0	FAIL	91	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:39:33	11:40:10	Tester_1	4	00:00:00:0	FAIL	92	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3320.49
11:39:33	11:40:10	Tester_1	5	00:00:00:0	FAIL	93	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3334.76
11:39:33	11:40:10	Tester_1	6	00:00:00:0	FAIL	94	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3334.76
11:39:33	11:40:10	Tester_1	7	00:00:00:0	FAIL	95	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3310.97
11:39:33	11:40:10	Tester_1	8	00:00:00:0	FAIL	96	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:39:33	11:40:10	Tester_1	9	00:00:00:0	FAIL	97	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:39:33	11:40:10	Tester_1	10	00:00:00:0	FAIL	98	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3320.49
11:39:33	11:40:10	Tester_1	11	00:00:00:0	FAIL	99	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3315.73
11:39:33	11:40:10	Tester_1	12	00:00:00:0	FAIL	100	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3325.24
11:39:33	11:40:10	Tester_1	13	00:00:00:0	FAIL	101	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:39:33	11:40:10	Tester_1	14	00:00:00:0	FAIL	102	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:39:33	11:40:10	Tester_1	15	00:00:00:0	FAIL	103	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3325.24
11:39:33	11:40:10	Tester_1	16	00:00:00:0	FAIL	104	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3334.76
11:41:19	11:42:42	Tester_1	1	00:00:00:0	PASS	89	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:41:19	11:42:42	Tester_1	2	00:00:00:0	PASS	90	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3320.49
11:41:19	11:42:42	Tester_1	3	00:00:00:0	PASS	91	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:41:19	11:42:42	Tester_1	4	00:00:00:0	PASS	92	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3320.49
11:41:19	11:42:42	Tester_1	5	00:00:00:0	PASS	93	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3339.51
11:41:19	11:42:42	Tester_1	6	00:00:00:0	PASS	94	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3334.76
11:41:19	11:42:42	Tester_1	7	00:00:00:0	PASS	95	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3315.73
11:41:19	11:42:42	Tester_1	8	00:00:00:0	PASS	96	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3330
11:41:19	11:42:42	Tester_1	9	00:00:00:0	PASS	97	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3325.24
11:41:19	11:42:42	Tester_1	10	00:00:00:0	PASS	98	PASS	C:\DiaSem	PASS	0.0.0.4	PASS	PASS	C:\DiaSem	PASS	1.1	PASS	PASS	3325.24

Figure 83: CSV File

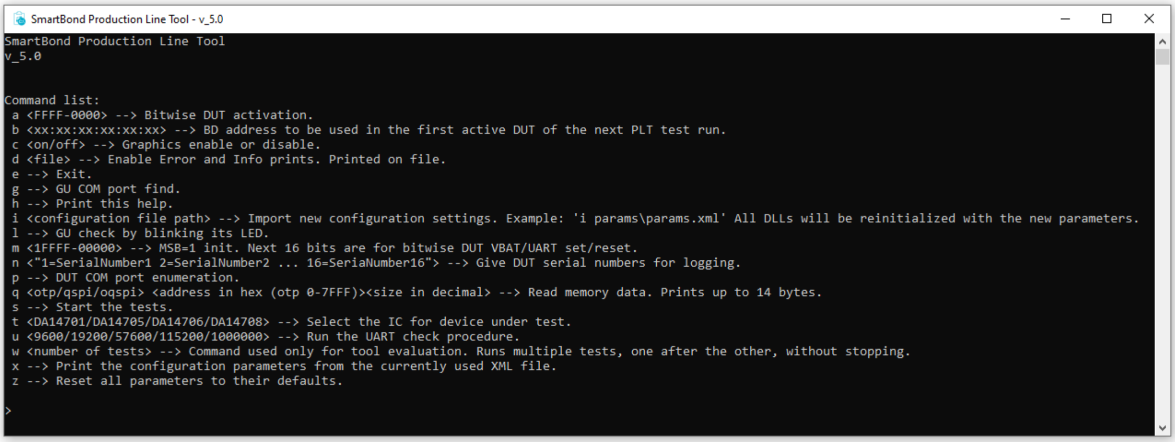
Figure 83 shows an example of a generated CSV file. As with the DUT Log File (Section 7.3.4), the PLT software and hardware information are shown along with valuable DUT information. The CSV file keeps information about all the production tests of a single day. A new CSV file is created every day.

7.4 CLI PLT Application

The CLI PLT (`SmartBond_PLT_CLI.exe`) is a Command Line Interface application with similar functionality and features as the PLT GUI. It performs the device testing and memory programming. At the same time, it allows users to monitor the test procedure in detail. It supports the same configuration file created from the CFG PLT and can run the same tests as the GUI PLT.

Figure 84 shows the initial screen of the CLI PLT software. The CLI PLT can be executed from a command line prompt, passing arguments externally and initiating the tests immediately. This is useful for scripting/batch files as shown in Section 7.4.3.

Parameters are automatically loaded from the `params/params.xml` file when the CLI PLT starts. If there is a parameter error, a warning will be shown. It is recommended to run the 'x' command or start the CFG PLT before running the tests and check the configuration parameters. If a change is made to the `params.xml` configuration file while CLI is open, the file should be reloaded using the 'i' command. If any OTP burning test is scheduled, a message will inform the user and prompt for continuing.



```
SmartBond Production Line Tool
v_5.0


Command list:
 a <FFFF-0000> --> Bitwise DUT activation.
 b <xx:xx:xx:xx:xx:xx> --> BD address to be used in the first active DUT of the next PLT test run.
 c <on/off> --> Graphics enable or disable.
 d <file> --> Enable Error and Info prints. Printed on file.
 e --> Exit.
 g --> GU COM port find.
 h --> Print this help.
 i <configuration file path> --> Import new configuration settings. Example: 'i params\params.xml' All DLLs will be reinitialized with the new parameters.
 l --> GU check by blinking its LED.
 m <1FFFF-00000> --> MSB=1 init. Next 16 bits are for bitwise DUT VBAT/UART set/reset.
 n <"1=SerialNumber1 2=SerialNumber2 ... 16=SeriaNumber16"> --> Give DUT serial numbers for logging.
 p --> DUT COM port enumeration.
 q <otp/qspi/oqspi> <address in hex (otp 0-7FFF)><size in decimal> --> Read memory data. Prints up to 14 bytes.
 s --> Start the tests.
 t <DA14701/DA14705/DA14706/DA14708> --> Select the IC for device under test.
 u <9600/19200/57600/115200/1000000> --> Run the UART check procedure.
 w <number of tests> --> Command used only for tool evaluation. Runs multiple tests, one after the other, without stopping.
 x --> Print the configuration parameters from the currently used XML file.
 z --> Reset all parameters to their defaults.

>
```

Figure 84: CLI Software Start Screen

7.4.1 CLI Commands

Table 62 lists all available CLI commands. A list with brief description of these commands can be printed using the 'h' command.

Table 62: CLI Commands

Cmd	Arguments	Description
a	Hex values from "FFFF" to "0000".	Bitwise DUT activation. Sets the active DUTs to be tested. **Examples:** ● "a 1": Only DUT 1 will be activated and tested. ● "a 9": DUTs 1 and DUT 4 will be activated and tested.
b	xx:xx:xx:xx:xx:xx	Sets the start BD address of the first active DUT. **Example:** "b FE:00:11:22:33:44"
c	on/off	Enables or disables the graphics debug output of the CLI. Useful in the read BD address command *r*, in order to see only the DUT BD address returned and not the entire process. **Example:** "c on"
d	● console ● file	Use this option to enable error and info prints. Choose file output or console output. Only the *file* option is currently supported. **Example:** "d file"

Cmd	Arguments	Description
`e`	*none*	Exits the application.
`g`	*none*	Execute the automatic GU Window COM port enumeration.
`h`	*none*	Help print out. Prints the list of the CLI commands that are available.
`i`	Path to XML configuration file.	Initializes the PLT with the parameters found in the `params.xml` file. **Example:** "`i params\params.xml`".
`l`	*none*	Run the GU sanity check. The Golden Unit will start blinking its red LED.
`m`	First character: "`1`" or "`0`". Then hex value from "`1FFFF`" to "`0000`".	MSB character should be '`1`' the first time this command is executed. Consecutive '`m`' commands should have the MSB character set to '`0`'. The next 16 bits are used for bitwise DUT VBAT/UART enable/disable. **Example:** "`m 1FFFF`"
`n`	1=SerialNumber1 2=SerialNumber2 .. 16=SeriaNumber 16	User can provide a serial number for each DUT. The serial number will be saved in the DUT log file and be added in the DUT log file name.
`p`	*none*	Execute the automatic DUT Window COM port enumeration.
`q`	First argument ● otp ● qspi ● oqspi Second argument ● The address in hex Third argument ● The size in bytes	This option can read from any memory, for any address offset up to 256 bytes of data.
`s`	*none*	Starts the tests.
`t`	● DA14701 ● DA14705 ● DA14706 ● DA14708	Selects the type of IC that the DUT uses. This option will change the DUT IC setting in the configuration file and reload all the settings **Example:** "`DA14701`".
`u`	● 9600 ● 19200 ● 57600 ● 115200 ● 1000000 ● 2000000	This command performs a UART check connection for all the active DUTs for a specific baud rate.
`w`	Number of tests to run.	This command is used only for PLT evaluation. It starts multiple tests. These are executed one after the other without user intervention.
`x`	*none*	Print the configuration parameters from the currently used XML file.
`z`	*none*	Resets all the XML parameters to their defaults.

7.4.2 Running the CLI and Executing Tests

There are several options to be called in order to make sure that the CLI PLT is set up correctly. Each of following commands is explained in Table 62.

- Using the help command (‘h’) the entire CLI command list will be shown.

  Example: >h

Set Console Options

- To redirect the debugging messages to the file use command ‘d’. This option is going to replace the UI debug values in the configuration file.

  Example: >d file

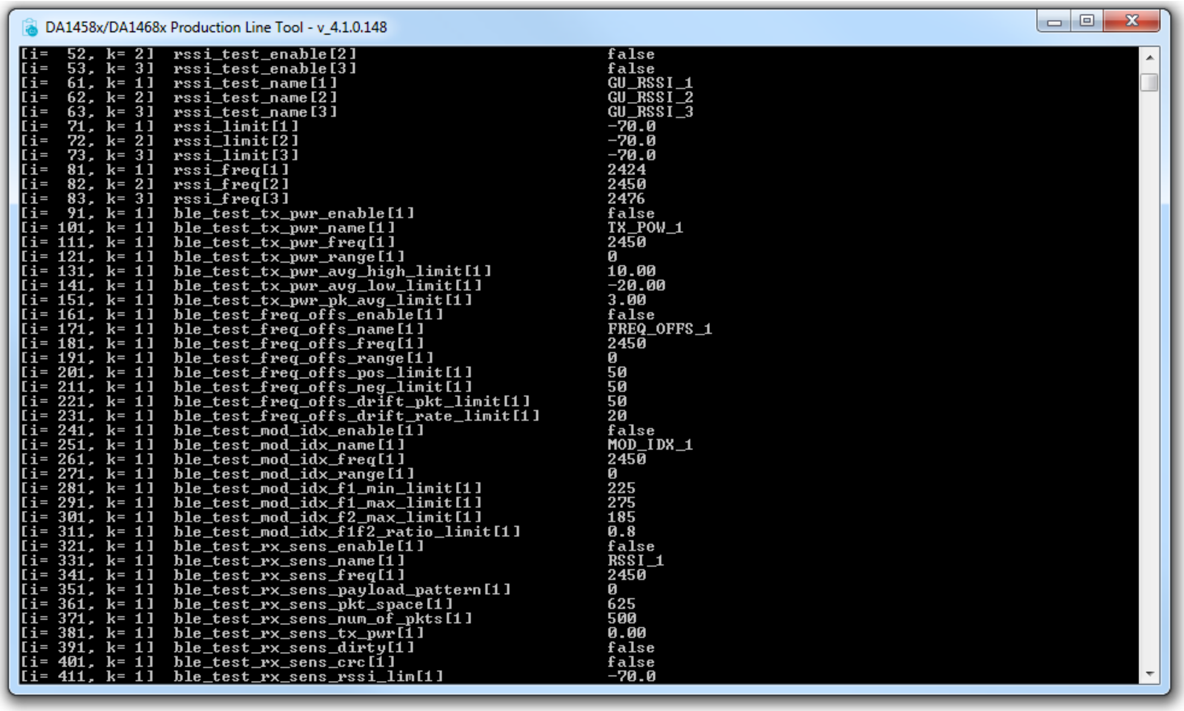
- To show or hide any prints in the Console window use command ‘c’ with on/off argument.

  Example: >c on

Check, Reset, Reload, and Change Settings

- Because the configuration file is automatically loaded, use the ‘x’ command (Figure 85) to see the loaded settings. Errors will be shown in red.

  Example: >x



```
[i=  52, k= 2]  rssi_test_enable[2]                          false
[i=  53, k= 3]  rssi_test_enable[3]                          false
[i=  61, k= 1]  rssi_test_name[1]                            GU_RSSI_1
[i=  62, k= 2]  rssi_test_name[2]                            GU_RSSI_2
[i=  63, k= 3]  rssi_test_name[3]                            GU_RSSI_3
[i=  71, k= 1]  rssi_limit[1]                                -70.0
[i=  72, k= 2]  rssi_limit[2]                                -70.0
[i=  73, k= 3]  rssi_limit[3]                                -70.0
[i=  81, k= 1]  rssi_freq[1]                                 2424
[i=  82, k= 2]  rssi_freq[2]                                 2450
[i=  83, k= 3]  rssi_freq[3]                                 2476
[i=  91, k= 1]  ble_test_tx_pwr_enable[1]                    false
[i= 101, k= 1]  ble_test_tx_pwr_name[1]                      TX_POW_1
[i= 111, k= 1]  ble_test_tx_pwr_freq[1]                      2450
[i= 121, k= 1]  ble_test_tx_pwr_range[1]                     0
[i= 131, k= 1]  ble_test_tx_pwr_avg_high_limit[1]            10.00
[i= 141, k= 1]  ble_test_tx_pwr_avg_low_limit[1]             -20.00
[i= 151, k= 1]  ble_test_tx_pwr_pk_avg_limit[1]              3.00
[i= 161, k= 1]  ble_test_freq_offs_enable[1]                 false
[i= 171, k= 1]  ble_test_freq_offs_name[1]                   FREQ_OFFS_1
[i= 181, k= 1]  ble_test_freq_offs_freq[1]                   2450
[i= 191, k= 1]  ble_test_freq_offs_range[1]                  0
[i= 201, k= 1]  ble_test_freq_offs_pos_limit[1]              50
[i= 211, k= 1]  ble_test_freq_offs_neg_limit[1]              50
[i= 221, k= 1]  ble_test_freq_offs_drift_pkt_limit[1]        50
[i= 231, k= 1]  ble_test_freq_offs_drift_rate_limit[1]       20
[i= 241, k= 1]  ble_test_mod_idx_enable[1]                   false
[i= 251, k= 1]  ble_test_mod_idx_name[1]                     MOD_IDX_1
[i= 261, k= 1]  ble_test_mod_idx_freq[1]                     2450
[i= 271, k= 1]  ble_test_mod_idx_range[1]                    0
[i= 281, k= 1]  ble_test_mod_idx_f1_min_limit[1]             225
[i= 291, k= 1]  ble_test_mod_idx_f1_max_limit[1]             275
[i= 301, k= 1]  ble_test_mod_idx_f2_max_limit[1]             185
[i= 311, k= 1]  ble_test_mod_idx_f1f2_ratio_limit[1]         0.8
[i= 321, k= 1]  ble_test_rx_sens_enable[1]                   false
[i= 331, k= 1]  ble_test_rx_sens_name[1]                     RSSI_1
[i= 341, k= 1]  ble_test_rx_sens_freq[1]                     2450
[i= 351, k= 1]  ble_test_rx_sens_payload_pattern[1]          0
[i= 361, k= 1]  ble_test_rx_sens_pkt_space[1]                625
[i= 371, k= 1]  ble_test_rx_sens_num_of_pkts[1]              500
[i= 381, k= 1]  ble_test_rx_sens_tx_pwr[1]                   0.00
[i= 391, k= 1]  ble_test_rx_sens_dirty[1]                    false
[i= 401, k= 1]  ble_test_rx_sens_crc[1]                      false
[i= 411, k= 1]  ble_test_rx_sens_rssi_lim[1]                 -70.0
```

Figure 85: CLI PLT Print Settings (x Command)

- To reset the configuration parameters to their defaults values the ‘z’ command should be used.
- To reload the configuration file or to load another one, ‘i’ command can be used.

  Example: >i params/params.xml

- To change only the selected device IC, use the ‘t xxx’ command, where ‘xxx’ is the desired IC selection. If a change is made, all the settings will be reloaded.

  Example: >t DA14705

- To change the active DUTs use the 'a' command. As an argument a 16-bit hexadecimal value is used, which is the bitwise representation of the active DUTs with DUT 1 being the LSB. This command will replace the dut_num_x values in the configuration file.

  The following example will enable only DUT1, DUT2, DUT15 and DUT 16.

Example: >a C003

- To change the BD address that will be used in the next run use the 'b' command.

  This option is going to replace the *BD address* and *Statistics* values in the configuration file.

  Example: >b 00:00:00:00:00:01

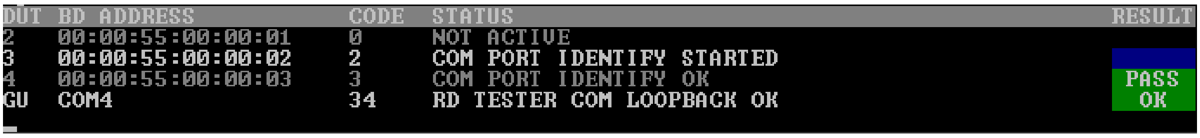
Hardware Specific Tests

- To automatically find the Windows COM port assigned to the Golden Unit, use the 'g' command. This command will replace the gu_com_port value in the configuration file.

  Example: >g
- To verify that the Golden Unit COM port is found correctly and to check if the Golden Unit is ready run the 'l' command.

  Example: >l
- To automatically find the Windows COM Port assigned for each DUT, use the 'p' command. This command will replace the com_port_dut_x values in the configuration file.

  Example: >p



```
DUT BD ADDRESS                CODE  STATUS                         RESULT
2   00:00:55:00:00:01         0     NOT ACTIVE
3   00:00:55:00:00:02         2     COM PORT IDENTIFY STARTED
4   00:00:55:00:00:03         3     COM PORT IDENTIFY OK            PASS
GU  COM4                      34    RD TESTER COM LOOPBACK OK       OK
```

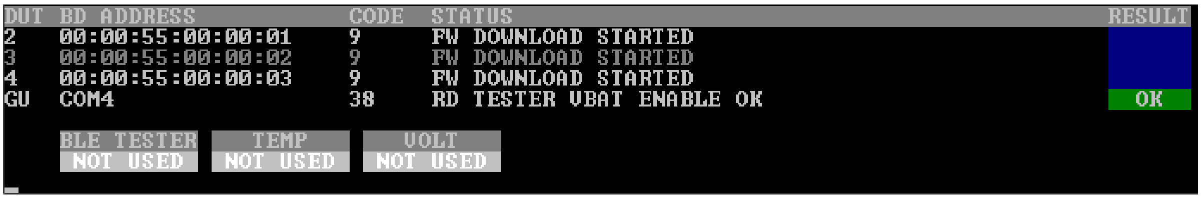
Figure 86: CLI PLT DUT COM Port Enumeration ('p' Command)

- To run a UART error check use the 'u' command followed with a specific Baud rate.

  Example: >u 1000000

PLT Production Tests

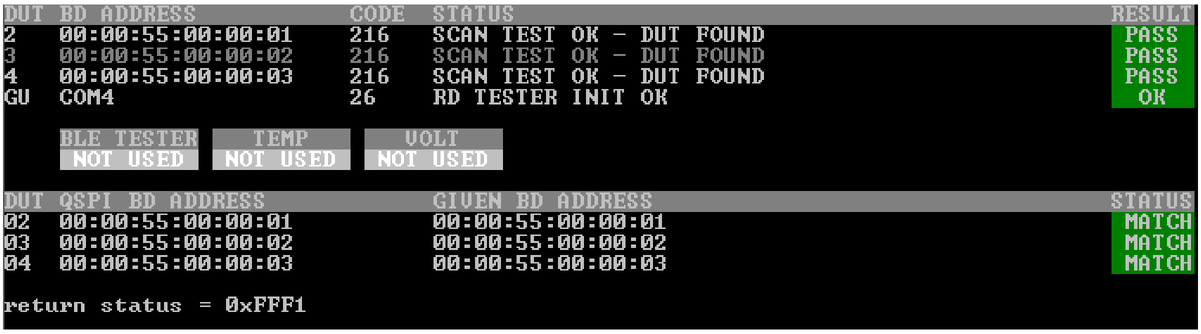
- Use the 's' command to begin testing with the current configuration. Figure 87 shows the CLI during the testing. After all the tests have finished, the result remains on the screen as shown in Figure 88.



```
DUT BD ADDRESS                CODE  STATUS                         RESULT
2   00:00:55:00:00:01         9     FW DOWNLOAD STARTED
3   00:00:55:00:00:02         9     FW DOWNLOAD STARTED
4   00:00:55:00:00:03         9     FW DOWNLOAD STARTED
GU  COM4                      38    RD TESTER VBAT ENABLE OK        OK

    BLE TESTER  TEMP       VOLT
    NOT USED    NOT USED   NOT USED
```

Figure 87: CLI PLT Testing



```
DUT BD ADDRESS                CODE  STATUS                         RESULT
2   00:00:55:00:00:01         216   SCAN TEST OK - DUT FOUND        PASS
3   00:00:55:00:00:02         216   SCAN TEST OK - DUT FOUND        PASS
4   00:00:55:00:00:03         216   SCAN TEST OK - DUT FOUND        PASS
GU  COM4                      26    RD TESTER INIT OK               OK

    BLE TESTER  TEMP       VOLT
    NOT USED    NOT USED   NOT USED

DUT QSPI BD ADDRESS                 GIVEN BD ADDRESS               STATUS
02  00:00:55:00:00:01               00:00:55:00:00:01               MATCH
03  00:00:55:00:00:02               00:00:55:00:00:02               MATCH
04  00:00:55:00:00:03               00:00:55:00:00:03               MATCH

return status = 0xFFF1
```

Figure 88: CLI PLT Testing Finished

- Use the 'q' command to read from any memory up to 256 bytes of data. The following example will read the first 10 bytes from the flash memory of the DUT.
  Example: `>q oqspi 0 10`

Other Test Commands

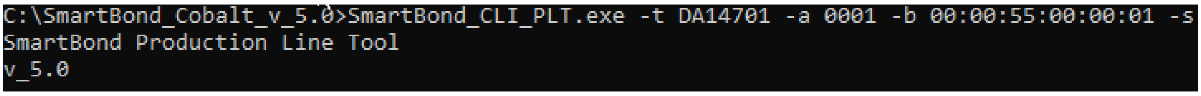
- Use the 'm' command to power on and access the DUTs to perform further testing. This will open the VBAT and the COM ports from the PLT hardware to the DUTs. As an argument, a ‘0’ or ‘1’ character is used to reset the PLT hardware. This is followed by a 16-bit hexadecimal number, which is the bitwise representation of the DUTs to use, with DUT 1 being the LSB.

  In the following example, the PLT hardware will be reset and DUTs 1, 2, 15 and 16 will be used.

  Example: `>m 1C003`

7.4.3 Using CLI Commands with Arguments

It is possible to start the CLI program with the commands described in Section 7.4.2 as arguments. This is useful for scripting/batch files.



```
C:\SmartBond_Cobalt_v_5.0>SmartBond_CLI_PLT.exe -t DA14701 -a 0001 -b 00:00:55:00:00:01 -s
SmartBond Production Line Tool
v_5.0
```

Figure 89: CLI with Commands as Arguments

The example shown in Figure 89 will perform the following commands:

1. '`-t DA14701`': Set the DUT as DA14701
2. '`-a 0001`': Set the DUT1 as the only active DUT.
3. '`-b 00:00:55:00:00:01`'. Assign as the first BD address to be assigned the `00:00:55:00:00:01`. This BD address will be used in DUT1, as it is the only active DUT.
4. '`-s`': Perform the tests. BD address write in QSPI header should be enabled for the following test to pass.

7.5 GU Upgrade Application

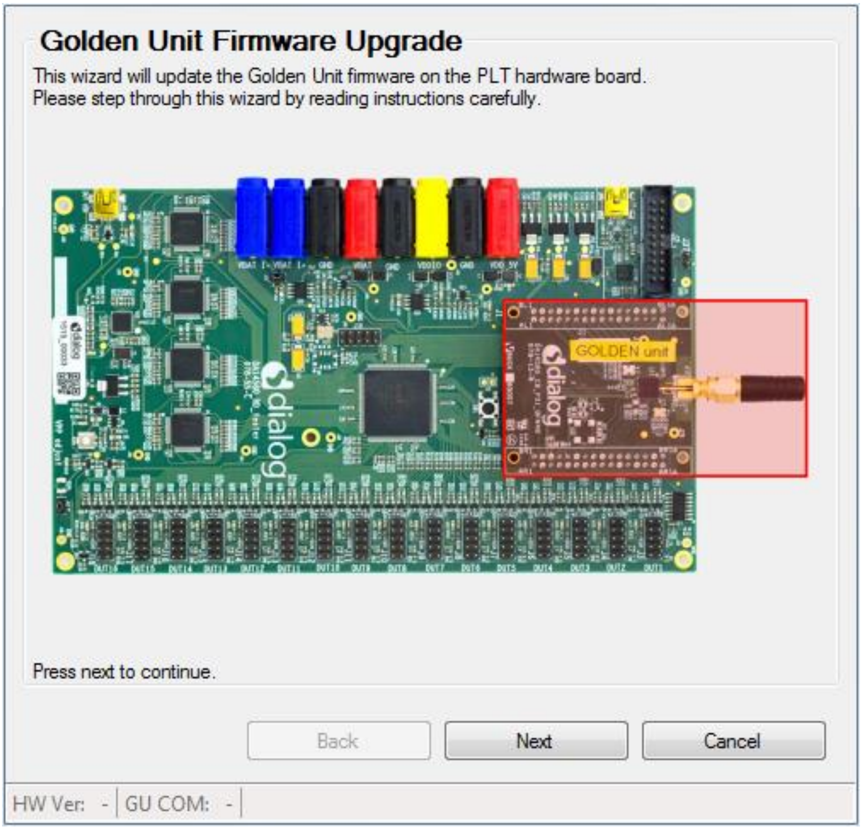
The GU Upgrade (GU_fw_upgrade.exe) is a Graphical User Interface application, which can be used to upgrade the firmware of the Golden Unit automatically, in contrast with that describes a manual way to upgrade the firmware of the Golden Unit. It guides the user to configure, detect the PLT hardware and finally reprogram the SPI Flash memory onboard the PLT hardware with the Golden Unit firmware.

Note: Quick access to GU upgrade tool is provided by pressing the Upgrade GU firmware button, under Firmware Version – Golden Unit section 7.2.2.5 in CFG PLT. Current version of the GU firmware can be seen using the Refresh button on the same section.

Note: This tool cannot upgrade PLT hardware version A.

7.5.1 Introduction Screen

The first page of the tool is an introduction screen (Figure 90) showing the purpose of the tool. User can exit the tool at any step by pressing the **Cancel** button or close the application using the **X** button from the windows bar at the top.

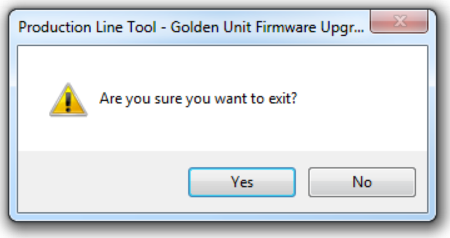


Golden Unit Firmware Upgrade

This wizard will update the Golden Unit firmware on the PLT hardware board.
Please step through this wizard by reading instructions carefully.

Press next to continue.

Figure 90: GU Upgrade - Introduction Page



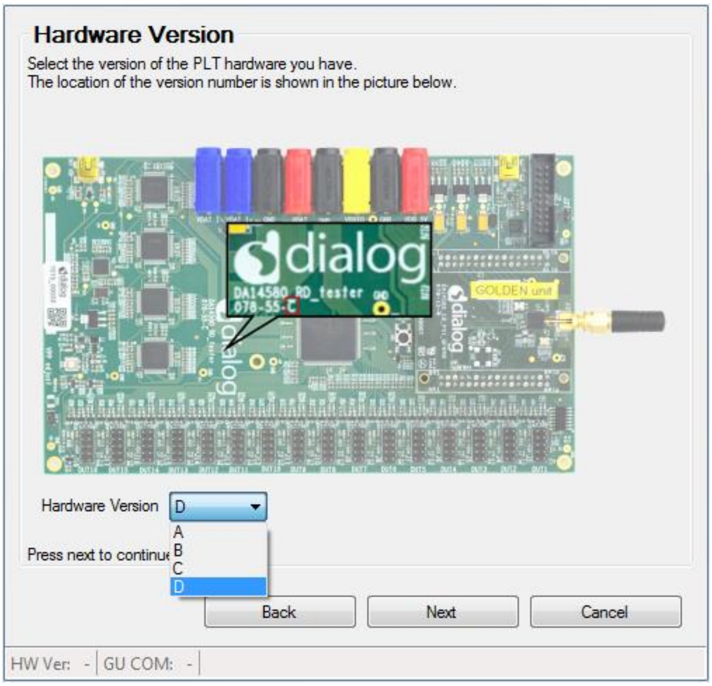
Production Line Tool - Golden Unit Firmware Upgr...

Are you sure you want to exit?

Figure 91: GU FW Upgrade - Exit Message

7.5.2 Hardware Version Screen

In this screen the PLT hardware version can be selected. Depending on the version, some options may be missing, or the tool may not support it. Selected hardware version will be shown on the left bottom corner of the tool at any of the following steps. As noted, the PLT hardware `version A` is not supported by this tool.



Hardware Version

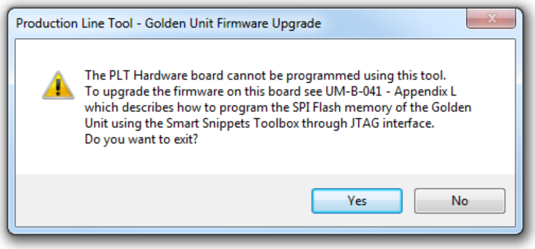
Select the version of the PLT hardware you have.
The location of the version number is shown in the picture below.

Hardware Version: D

Press next to continue

HW Ver: - | GU COM: -

Figure 92: GU FW Upgrade - Hardware Version



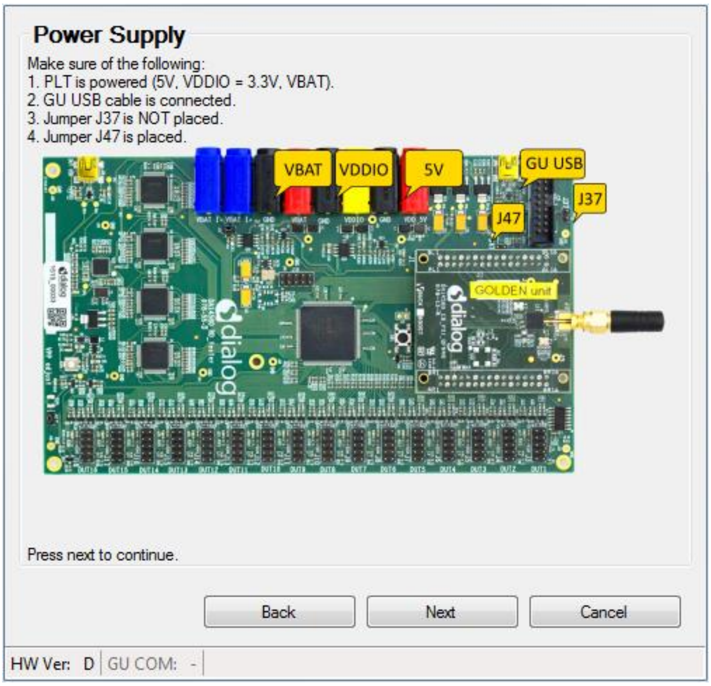
Production Line Tool - Golden Unit Firmware Upgrade

The PLT Hardware board cannot be programmed using this tool.
To upgrade the firmware on this board see UM-B-041 - Appendix L
which describes how to program the SPI Flash memory of the Golden
Unit using the Smart Snippets Toolbox through JTAG interface.
Do you want to exit?

Figure 93: GU FW Upgrade - Hardware Version Compatibility

7.5.3 Power Supply Screen

Connect the power supply of the PLT, as described in Section 5.3 and connect the jumpers as shown in Figure 95 where applicable. Adjust the jumpers as proposed by the tool.

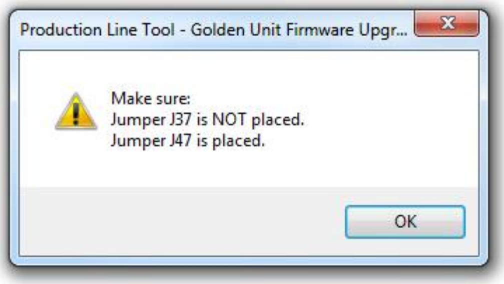


Power Supply

Make sure of the following:
1. PLT is powered (5V, VDDIO = 3.3V, VBAT).
2. GU USB cable is connected.
3. Jumper J37 is NOT placed.
4. Jumper J47 is placed.

Press next to continue.

Figure 94: GU FW Upgrade - Power Supply



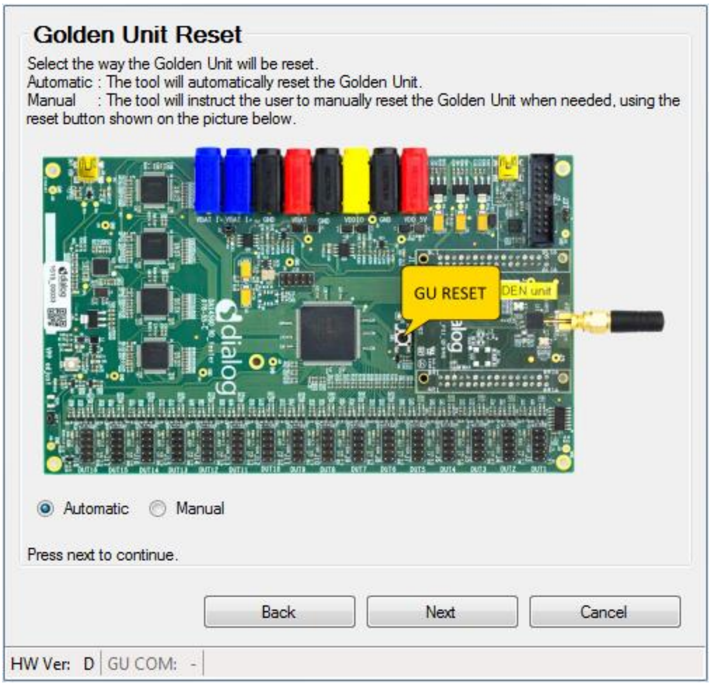
Production Line Tool - Golden Unit Firmware Upgr...

Make sure:
Jumper J37 is NOT placed.
Jumper J47 is placed.

Figure 95: GU FW Upgrade - Power Supply Pop-Up

7.5.4 Golden Unit Reset Screen

In this screen the way the GU will be reset can be selected. User can manually reset the GU, but PLT can do it automatically, which is the default selection. If the manual mode is selected, the user will be prompt any time the Golden Unit must be reset, to click the reset button located next to the Golden Unit on top of the PLT hardware.



Golden Unit Reset

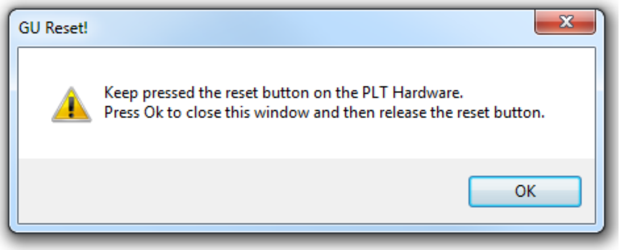
Select the way the Golden Unit will be reset.
Automatic : The tool will automatically reset the Golden Unit.
Manual : The tool will instruct the user to manually reset the Golden Unit when needed, using the reset button shown on the picture below.

● Automatic ○ Manual

Press next to continue.

HW Ver: D | GU COM: -

Figure 96: GU FW Upgrade - Golden Unit Reset



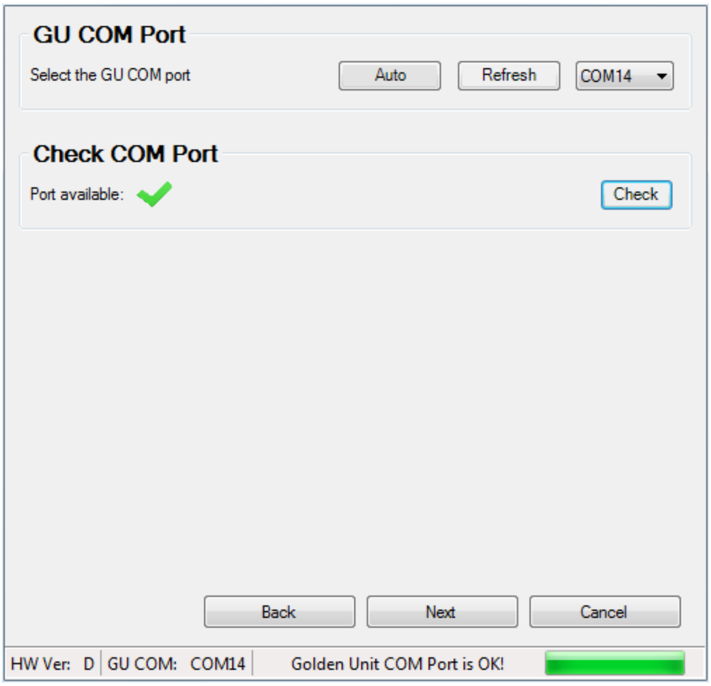
GU Reset!

Keep pressed the reset button on the PLT Hardware.
Press Ok to close this window and then release the reset button.

Figure 97: GU FW Upgrade - Golden Unit Reset Message for Manual Mode

7.5.5 GU COM Port Screen

The Windows assigned GU COM port can be selected in this screen. User can either select it from the dropdown list or use the *Auto* button to find it using the serial number as described in Appendix H. GU COM port can be also verified using the **Check** button. Selected GU COM port will be shown on the left bottom corner of the tool.



GU COM Port

Select the GU COM port

Check COM Port

Port available: ✓

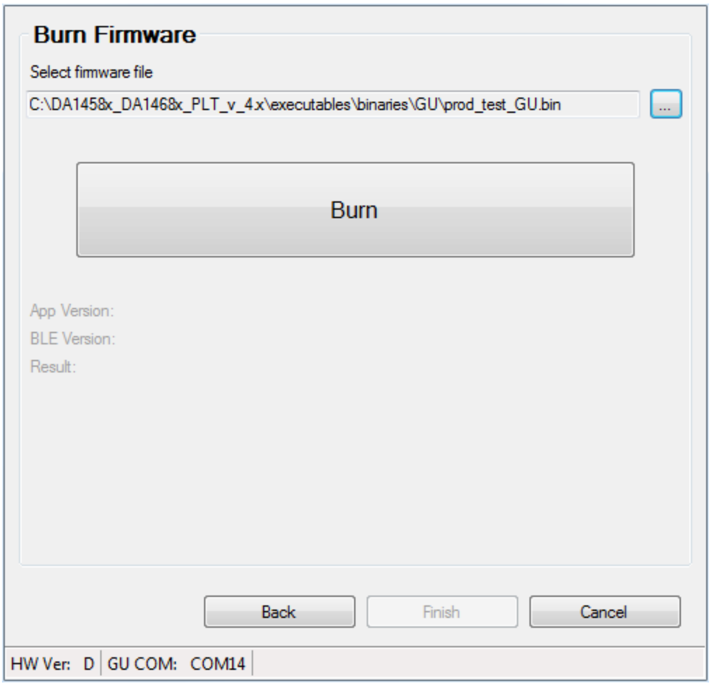
HW Ver: D | GU COM: COM14 | Golden Unit COM Port is OK!

Figure 98: GU FW Upgrade - GU COM Port

7.5.6 Burn Firmware Screen

This is the final step. The binary to burn should be selected.

Pressing the **Burn** button will erase the SPI Flash on the PLT hardware, program it with the new firmware selected, and read it back to verify that the contents written are the same as those in the binary.



Burn Firmware

Select firmware file

C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\GU\prod_test_GU.bin

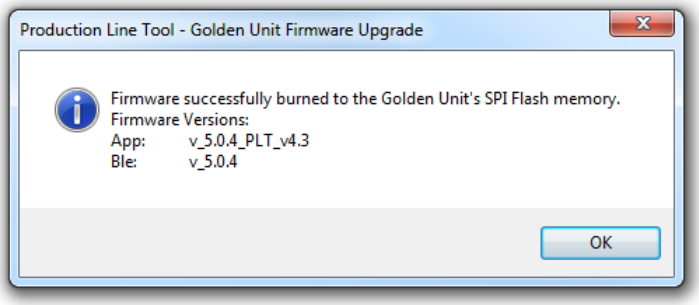
App Version:

BLE Version:

Result:

HW Ver: D | GU COM: COM14

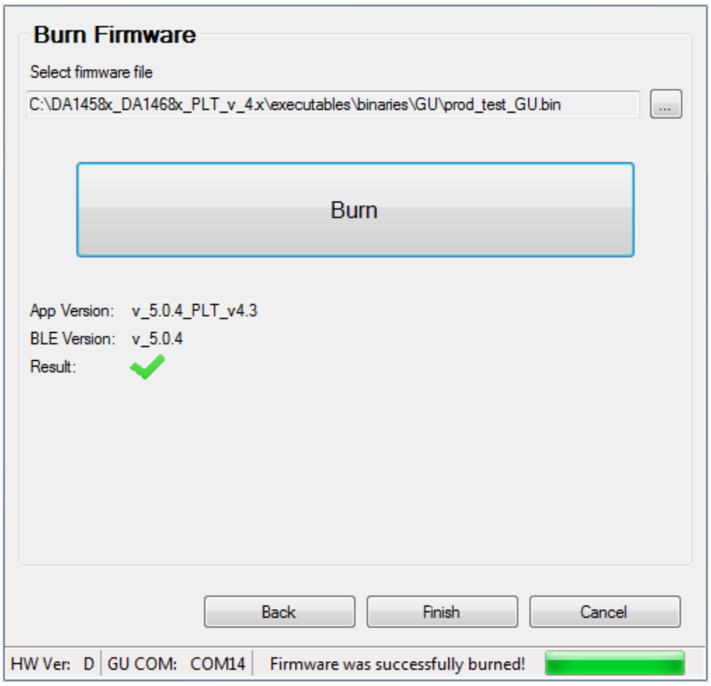
Figure 99: GU FW Upgrade - Burn Firmware



Production Line Tool - Golden Unit Firmware Upgrade

Firmware successfully burned to the Golden Unit's SPI Flash memory.
Firmware Versions:
App: v_5.0.4_PLT_v4.3
Ble: v_5.0.4

Figure 100: GU Upgrade - Burn Firmware Pop-Up Message



Burn Firmware

Select firmware file: C:\DA1458x_DA1468x_PLT_v_4.x\executables\binaries\GU\prod_test_GU.bin

App Version: v_5.0.4_PLT_v4.3
BLE Version: v_5.0.4
Result: ✓

HW Ver: D | GU COM: COM14 | Firmware was successfully burned!

Figure 101: GU FW Upgrade - Burn Firmware Success

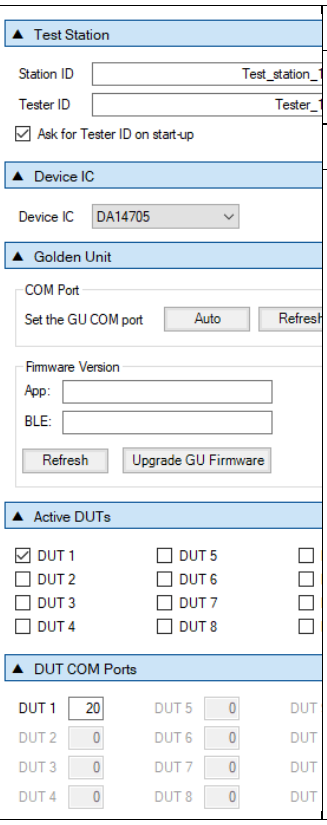
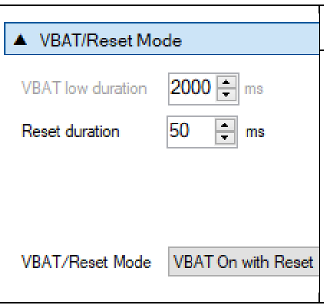
After the SPI flash program procedure is finished, a pop-up message appears with the result of the programming procedure. If the SPI flash was programmed successfully, the pop-up message will also show the version of the new Golden Unit firmware (Figure 100).

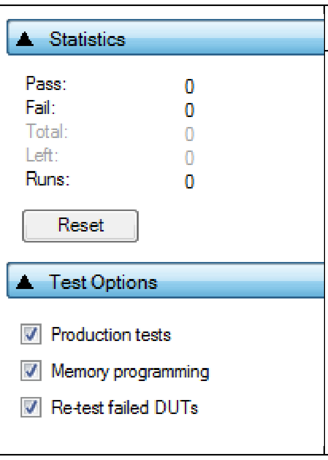
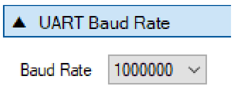
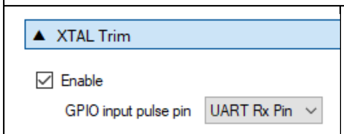
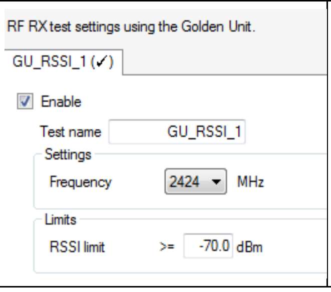
8 Example Usage

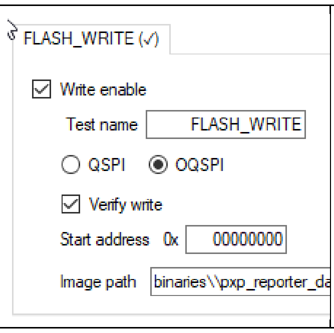
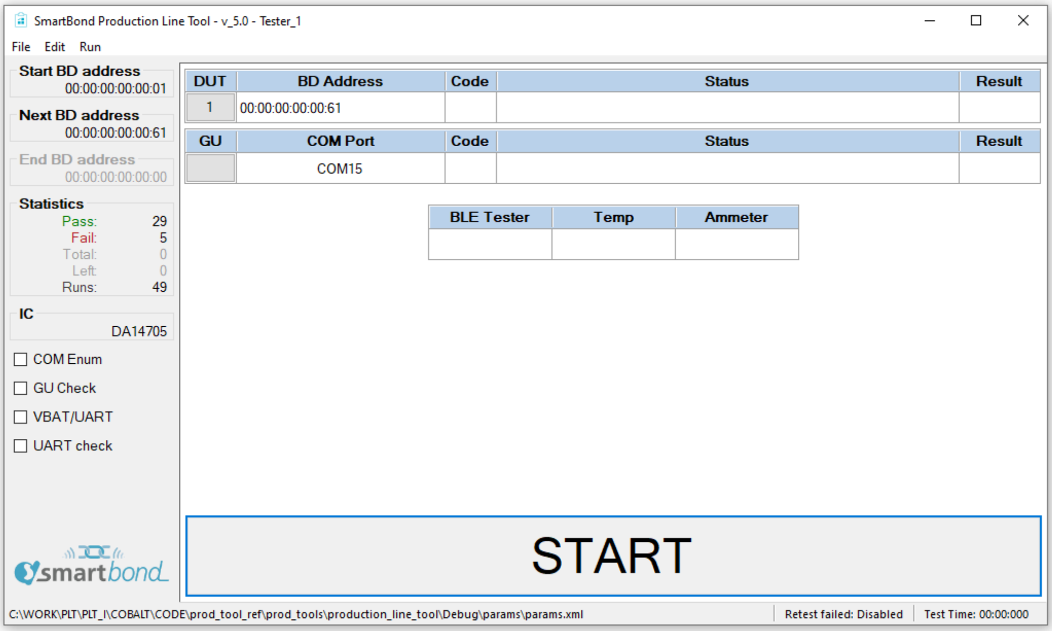
In this section, a simple example of the PLT will be described using one DA14701 device. The example test procedure is explained step-by-step in Table 63.

The tests to run in this example are the XTAL trim, RF RSSI test, OQSPI programming. One DUT will be used at PLT DUT connector position DUT1.

Table 63: PLT Example Usage

Step	Description		
Hardware Connections			
1	Connect both DUTs to the PLT hardware as shown in Appendix K. Four cables are needed. To use the SPI Flash memory, a custom triple-jumper must be used.		
CFG GUI Settings			
2	Open CFG PLT, make the following selections.		
3	Hardware Setup		
	 Test Station Station ID: Test_station_1 Tester ID: Tester_1 ☑ Ask for Tester ID on start-up Device IC Device IC: DA14705 Golden Unit COM Port Set the GU COM port [Auto] [Refresh] Firmware Version App: BLE: [Refresh] [Upgrade GU Firmware] Active DUTs ☑ DUT 1 ☐ DUT 5 ☐ ☐ DUT 2 ☐ DUT 6 ☐ ☐ DUT 3 ☐ DUT 7 ☐ ☐ DUT 4 ☐ DUT 8 ☐ DUT COM Ports DUT 1: 20 DUT 5: 0 DUT DUT 2: 0 DUT 6: 0 DUT DUT 3: 0 DUT 7: 0 DUT DUT 4: 0 DUT 8: 0 DUT	Device IC	Select DA14701, then Save.
		Golden Unit COM Port	Click on Auto, then Save.
		Active DUTs	Enable DUT1 only.
		DUT COM Ports	Click on **Enum**, then Save.
4	VBAT/Reset Mode		
	 VBAT/Reset Mode VBAT low duration: 2000 ms Reset duration: 50 ms VBAT/Reset Mode: VBAT On with Reset	Reset duration	Set a time for the Reset to stay active.
		VBAT/Reset Mode	Use **VBAT On with Reset** since it is faster to reset DUTs rather that performing a POR with the VBAT signal.

Step	Description		
5	General		
	 Statistics Pass: 0 Fail: 0 Total: 0 Left: 0 Runs: 0 Reset Test Options ☑ Production tests ☑ Memory programming ☑ Re-test failed DUTs	Statistics	Click Reset, then Save.
		Test Options	Enable all options then Save. Production tests should be enabled for the XTAL Trim and the RF tests to run. Memory programming is required for the QSPI erase and check empty functions. Disable the rest of the options.
6	DUT Hardware Setup		
	 UART Baud Rate Baud Rate 1000000	UART Baud Rate	1000000
7	Test Settings		
	 XTAL Trim ☑ Enable GPIO input pulse pin UART Rx Pin	XTAL Trim	Check *Enable* in XTAL Trim. Select the same pin as the *UART Rx*,
	 RF RX test settings using the Golden Unit. GU_RSSI_1 (✓) ☑ Enable Test name GU_RSSI_1 Settings Frequency 2424 MHz Limits RSSI limit >= -70.0 dBm	RF Tests - Golden Unit	Only one test is enabled for this example. In Golden Unit: ● Check *Enable* ● Select 2424 MHz as *frequency* ● Set the *RSSI limit* to -70 dBm ● Set *Packet error limit* to 10%

Step	Description		
8	Memory Functions		
	 FLASH_WRITE (✓) ☑ Write enable Test name FLASH_WRITE ○ QSPI ● OQSPI ☑ Verify write Start address 0x 00000000 Image path binaries\\pxp_reporter_da	Write 1 test	● Check Write enable ● Check OQSPI ● Check *Verify Write* ● Make sure `pxp_reporter_da1470x.img` is selected.
GUI PLT			
10	Open GUI PLT		
			

SmartBond Production Line Tool - v_5.0 - Tester_1

Start BD address 00:00:00:00:00:01

Next BD address 00:00:00:00:00:61

End BD address 00:00:00:00:00:00

Statistics

Pass:	29
Fail:	5
Total:	0
Left:	0
Runs:	49

IC DA14705

□ COM Enum

□ GU Check

□ VBAT/UART

□ UART check

DUT	BD Address	Code	Status	Result
1	00:00:00:00:00:61			

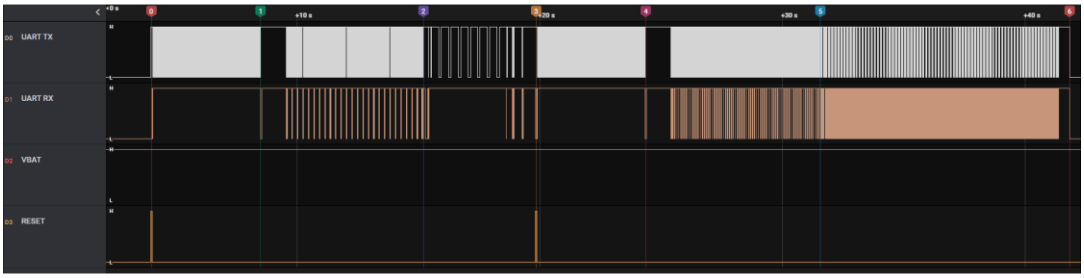
GU	COM Port	Code	Status	Result
	COM15			

BLE Tester	Temp	Ammeter

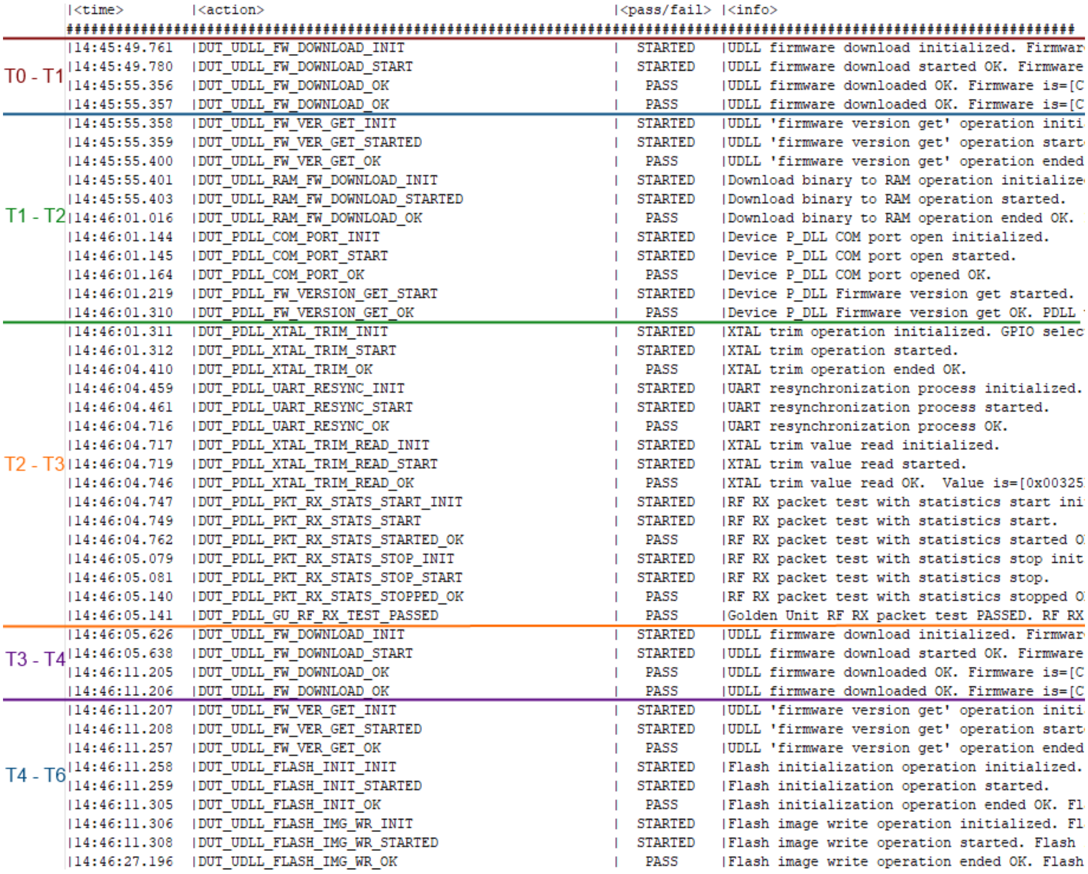
C:\WORK\PLT\PLT_I\COBALT\CODE\prod_tool_ref\prod_tools\production_line_tool\Debug\params\params.xml | Retest failed: Disabled | Test Time: 00:00:000

Step	Description
11	Press the Space Bar to begin testing.

The following screenshot shows the UART channels for both DUTs for the entire PLT run. Top markers show the timings between each of the active tests.

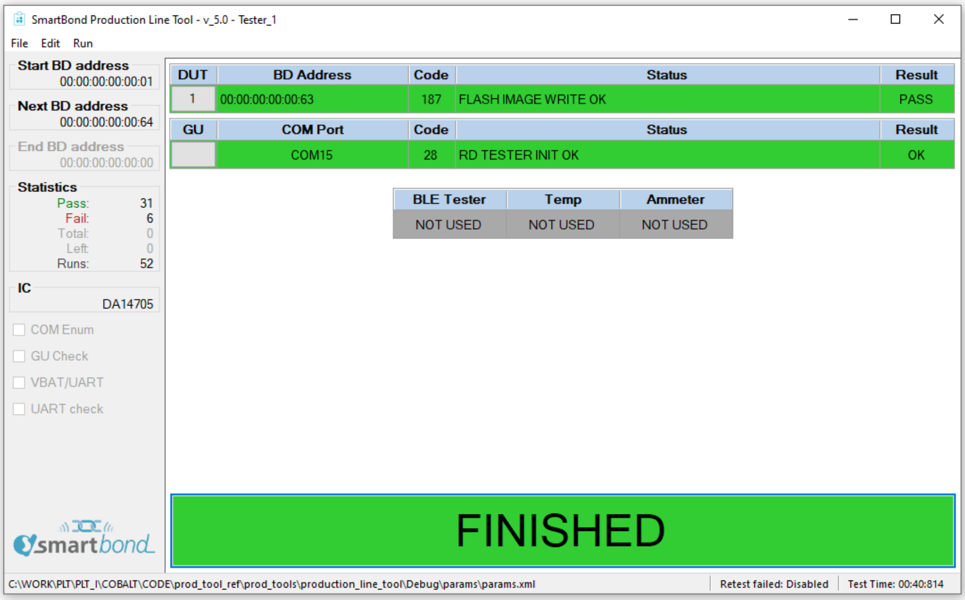


A log file following screenshot shows the test steps from the log file of DUT1 with the timing marks from the logic analyzer capture. These marks are described below. From the log file, the total time of each test can be calculated.



```
        |<time>        |<action>                                           |<pass/fail> |<info>
        ###########################################################################################################################
        |14:45:49.761  |DUT_UDLL_FW_DOWNLOAD_INIT                          |  STARTED   |UDLL firmware download initialized. Firmwar
T0 - T1 |14:45:49.780  |DUT_UDLL_FW_DOWNLOAD_START                         |  STARTED   |UDLL firmware download started OK. Firmware
        |14:45:55.356  |DUT_UDLL_FW_DOWNLOAD_OK                            |   PASS     |UDLL firmware downloaded OK. Firmware is=[C
        |14:45:55.357  |DUT_UDLL_FW_DOWNLOAD_OK                            |   PASS     |UDLL firmware downloaded OK. Firmware is=[C
        |14:45:55.358  |DUT_UDLL_FW_VER_GET_INIT                           |  STARTED   |UDLL 'firmware version get' operation initi
        |14:45:55.359  |DUT_UDLL_FW_VER_GET_STARTED                        |  STARTED   |UDLL 'firmware version get' operation start
        |14:45:55.400  |DUT_UDLL_FW_VER_GET_OK                             |   PASS     |UDLL 'firmware version get' operation ended
        |14:45:55.401  |DUT_UDLL_RAM_FW_DOWNLOAD_INIT                      |  STARTED   |Download binary to RAM operation initialize
        |14:45:55.403  |DUT_UDLL_RAM_FW_DOWNLOAD_STARTED                   |  STARTED   |Download binary to RAM operation started.
T1 - T2 |14:46:01.016  |DUT_UDLL_RAM_FW_DOWNLOAD_OK                        |   PASS     |Download binary to RAM operation ended OK.
        |14:46:01.144  |DUT_PDLL_COM_PORT_INIT                             |  STARTED   |Device P_DLL COM port open initialized.
        |14:46:01.145  |DUT_PDLL_COM_PORT_START                            |  STARTED   |Device P_DLL COM port open started.
        |14:46:01.164  |DUT_PDLL_COM_PORT_OK                               |   PASS     |Device P_DLL COM port opened OK.
        |14:46:01.219  |DUT_PDLL_FW_VERSION_GET_START                      |  STARTED   |Device P_DLL Firmware version get started.
        |14:46:01.310  |DUT_PDLL_FW_VERSION_GET_OK                         |   PASS     |Device P_DLL Firmware version get OK. PDLL
        |14:46:01.311  |DUT_PDLL_XTAL_TRIM_INIT                            |  STARTED   |XTAL trim operation initialized. GPIO selec
        |14:46:01.312  |DUT_PDLL_XTAL_TRIM_START                           |  STARTED   |XTAL trim operation started.
        |14:46:04.410  |DUT_PDLL_XTAL_TRIM_OK                              |   PASS     |XTAL trim operation ended OK.
        |14:46:04.459  |DUT_PDLL_UART_RESYNC_INIT                          |  STARTED   |UART resynchronization process initialized.
        |14:46:04.461  |DUT_PDLL_UART_RESYNC_START                         |  STARTED   |UART resynchronization process started.
        |14:46:04.716  |DUT_PDLL_UART_RESYNC_OK                            |   PASS     |UART resynchronization process OK.
        |14:46:04.717  |DUT_PDLL_XTAL_TRIM_READ_INIT                       |  STARTED   |XTAL trim value read initialized.
T2 - T3 |14:46:04.719  |DUT_PDLL_XTAL_TRIM_READ_START                      |  STARTED   |XTAL trim value read started.
        |14:46:04.746  |DUT_PDLL_XTAL_TRIM_READ_OK                         |   PASS     |XTAL trim value read OK.  Value is=[0x00325
        |14:46:04.747  |DUT_PDLL_PKT_RX_STATS_START_INIT                   |  STARTED   |RF RX packet test with statistics start ini
        |14:46:04.749  |DUT_PDLL_PKT_RX_STATS_START                        |  STARTED   |RF RX packet test with statistics start.
        |14:46:04.762  |DUT_PDLL_PKT_RX_STATS_STARTED_OK                   |   PASS     |RF RX packet test with statistics started O
        |14:46:05.079  |DUT_PDLL_PKT_RX_STATS_STOP_INIT                    |  STARTED   |RF RX packet test with statistics stop init
        |14:46:05.081  |DUT_PDLL_PKT_RX_STATS_STOP_START                   |  STARTED   |RF RX packet test with statistics stop.
        |14:46:05.140  |DUT_PDLL_PKT_RX_STATS_STOPPED_OK                   |   PASS     |RF RX packet test with statistics stopped O
        |14:46:05.141  |DUT_PDLL_GU_RF_RX_TEST_PASSED                      |   PASS     |Golden Unit RF RX packet test PASSED. RF RX
        |14:46:05.626  |DUT_UDLL_FW_DOWNLOAD_INIT                          |  STARTED   |UDLL firmware download initialized. Firmwar
T3 - T4 |14:46:05.638  |DUT_UDLL_FW_DOWNLOAD_START                         |  STARTED   |UDLL firmware download started OK. Firmware
        |14:46:11.205  |DUT_UDLL_FW_DOWNLOAD_OK                            |   PASS     |UDLL firmware downloaded OK. Firmware is=[C
        |14:46:11.206  |DUT_UDLL_FW_DOWNLOAD_OK                            |   PASS     |UDLL firmware downloaded OK. Firmware is=[C
        |14:46:11.207  |DUT_UDLL_FW_VER_GET_INIT                           |  STARTED   |UDLL 'firmware version get' operation initi
        |14:46:11.208  |DUT_UDLL_FW_VER_GET_STARTED                        |  STARTED   |UDLL 'firmware version get' operation start
        |14:46:11.257  |DUT_UDLL_FW_VER_GET_OK                             |   PASS     |UDLL 'firmware version get' operation ended
T4 - T6 |14:46:11.258  |DUT_UDLL_FLASH_INIT_INIT                           |  STARTED   |Flash initialization operation initialized.
        |14:46:11.259  |DUT_UDLL_FLASH_INIT_STARTED                        |  STARTED   |Flash initialization operation started.
        |14:46:11.305  |DUT_UDLL_FLASH_INIT_OK                             |   PASS     |Flash initialization operation ended OK. Fl
        |14:46:11.306  |DUT_UDLL_FLASH_IMG_WR_INIT                         |  STARTED   |Flash image write operation initialized. Fl
        |14:46:11.308  |DUT_UDLL_FLASH_IMG_WR_STARTED                      |  STARTED   |Flash image write operation started. Flash
        |14:46:27.196  |DUT_UDLL_FLASH_IMG_WR_OK                           |   PASS     |Flash image write operation ended OK. Flash
```

T0-T1	The UART boot firmware download (uartboot_da1470x.bin) begins. From the log file, the test lasted for about 5.6 seconds, which can be verified from the logic analyzer capture. This is followed by the operation to get the version of the uartboot_da1470x.bin firmware.
T1-T2	This time includes getting the version of the uartboot_da1470x.bin firmware, downloading st_fw_da1470x.bin in RAM and getting the version of the st_fw_da1470x.bin. This time is about 6 seconds.
T2-T3	This time includes the production tests. The XTAL trim and the GU RF RSSI test. Time is about 4.3 seconds.

Step		Description
	T3-T4	This time is for downloading `uartboot_da1470x.bin` to be used for memory programming. Time is about 5.6 seconds.
	T4-T5	This is the time for OQSPI Flash memory programming of `pxp_reporter_da1470x.img` (310 KB) with readback verify operation. Total time is about 16 seconds. Time can be reduced significantly if readback verification is disabled.
		The PLT run has finished. 

Start BD address 00:00:00:00:00:01

Next BD address 00:00:00:00:00:64

End BD address 00:00:00:00:00:00

Statistics

Pass:	31
Fail:	6
Total:	0
Left:	0
Runs:	52

IC DA14705

- □ COM Enum
- □ GU Check
- □ VBAT/UART
- □ UART check

DUT	BD Address	Code	Status	Result
1	00:00:00:00:00:63	187	FLASH IMAGE WRITE OK	PASS

GU	COM Port	Code	Status	Result
	COM15	28	RD TESTER INIT OK	OK

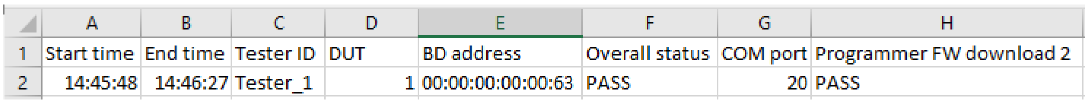
BLE Tester	Temp	Ammeter
NOT USED	NOT USED	NOT USED

FINISHED

C:\WORK\PLT\PLT_I\COBALT\CODE\prod_tool_ref\prod_tools\production_line_tool\Debug\params\params.xml | Retest failed: Disabled | Test Time: 00:40:814

After the tests have finished, the log and CSV files are fully updated. The GUI PLT test counter on the bottom right of the screen indicates that the test took approximately 40 seconds. This is verified from both the DUT log file and the CSV log file. Both files are shown below.

```
Software: SmartBond Production Line Tool
Software version: v_5.0
PLTD DLL version: v_5.0
PDLL Version: v_5.0
UDLL Version: v_5.0
Production test BLE firmware version: 11.0.1.0
Production test APP firmware version: 1.1
Flash programmer firmware version: 0.0.0.4
Date: 2022-10-25
Start Time: 14:45:48.487
End Time: 14:46:27.818
Station ID: Test_station_1
Device ID: 1
COM port: 20
DUT IC: DA14705
Tester ID: Tester_1
BD address: 00:00:00:00:00:63
Serial number: sd
```



	A	B	C	D	E	F	G	H
1	Start time	End time	Tester ID	DUT	BD address	Overall status	COM port	Programmer FW download 2
2	14:45:48	14:46:27	Tester_1	1	00:00:00:00:00:63	PASS	20	PASS

Appendix A Top View of PLT PCB Version D

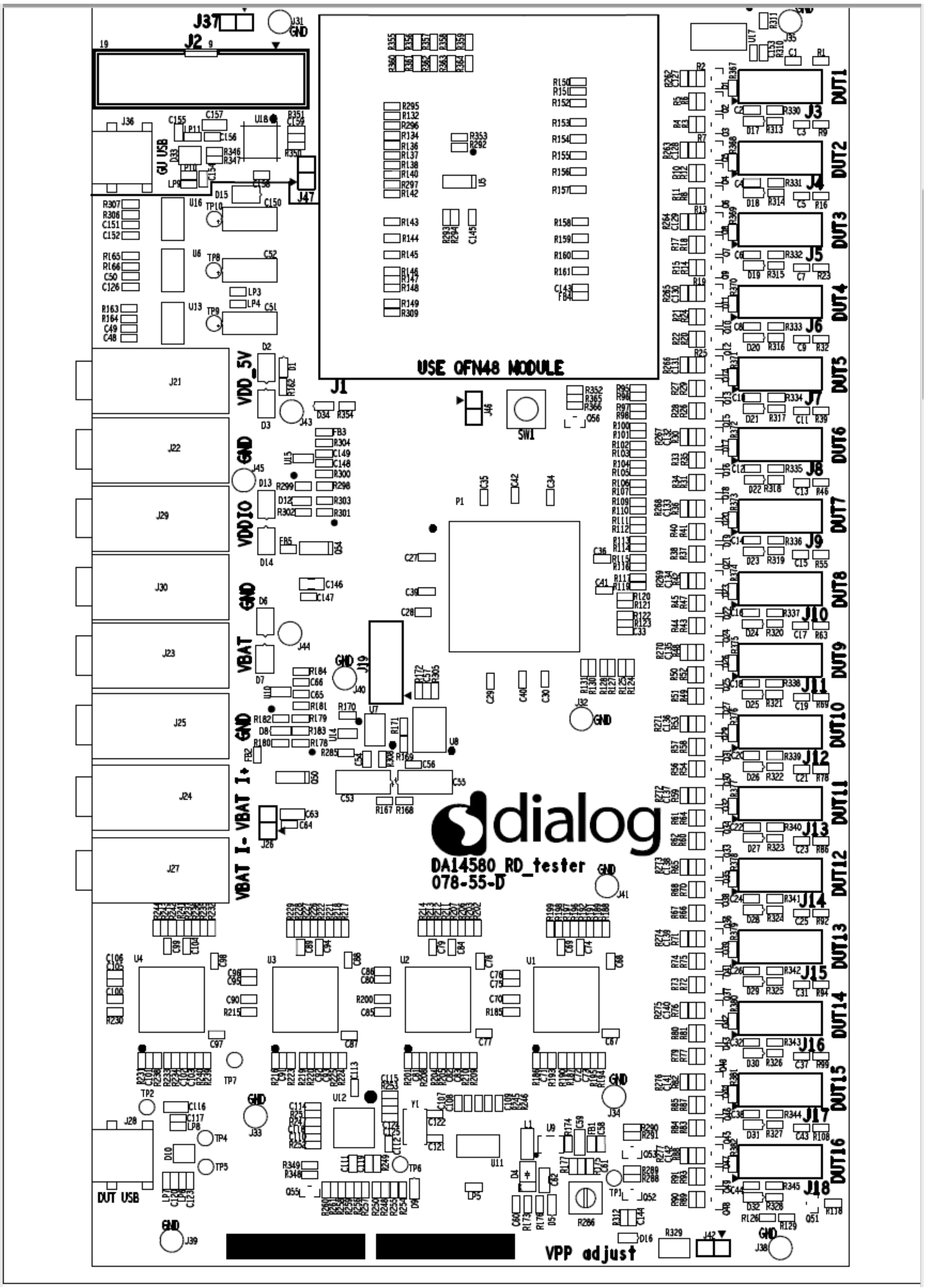


Figure 102: Top View of PLT PCB Version D

Appendix B Electrical Schematics

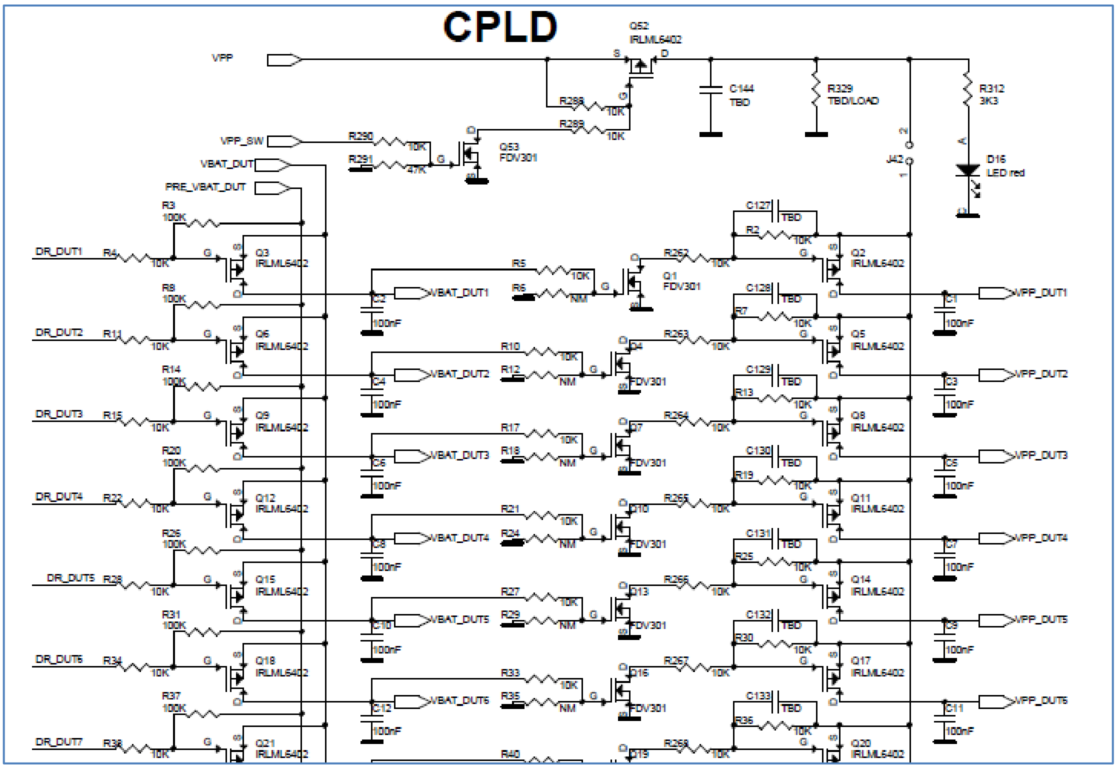


Figure 103: VBAT and VPP Control from CPLD

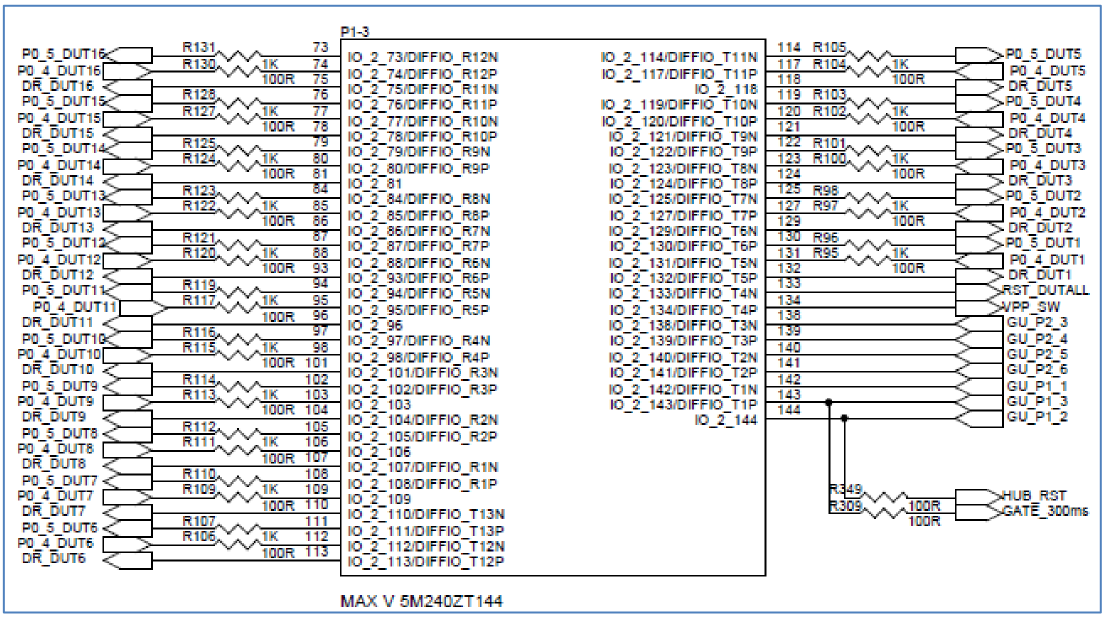


Figure 104: CPLD DUT UART Connections

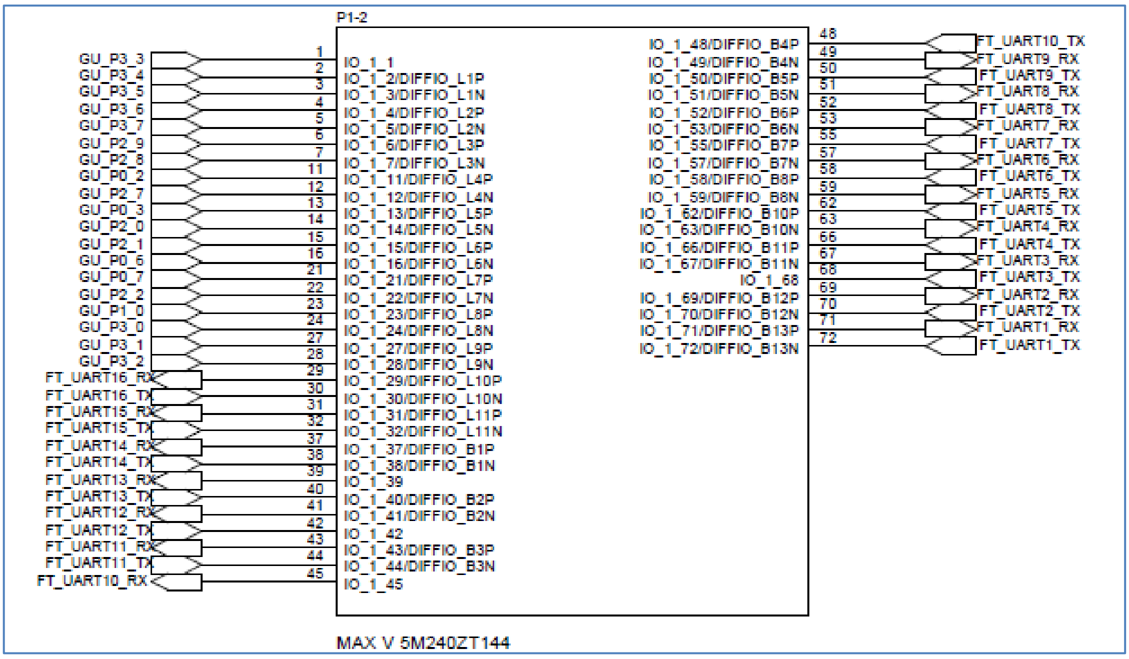


Figure 105: CPLD FTDI and GU Control Connections

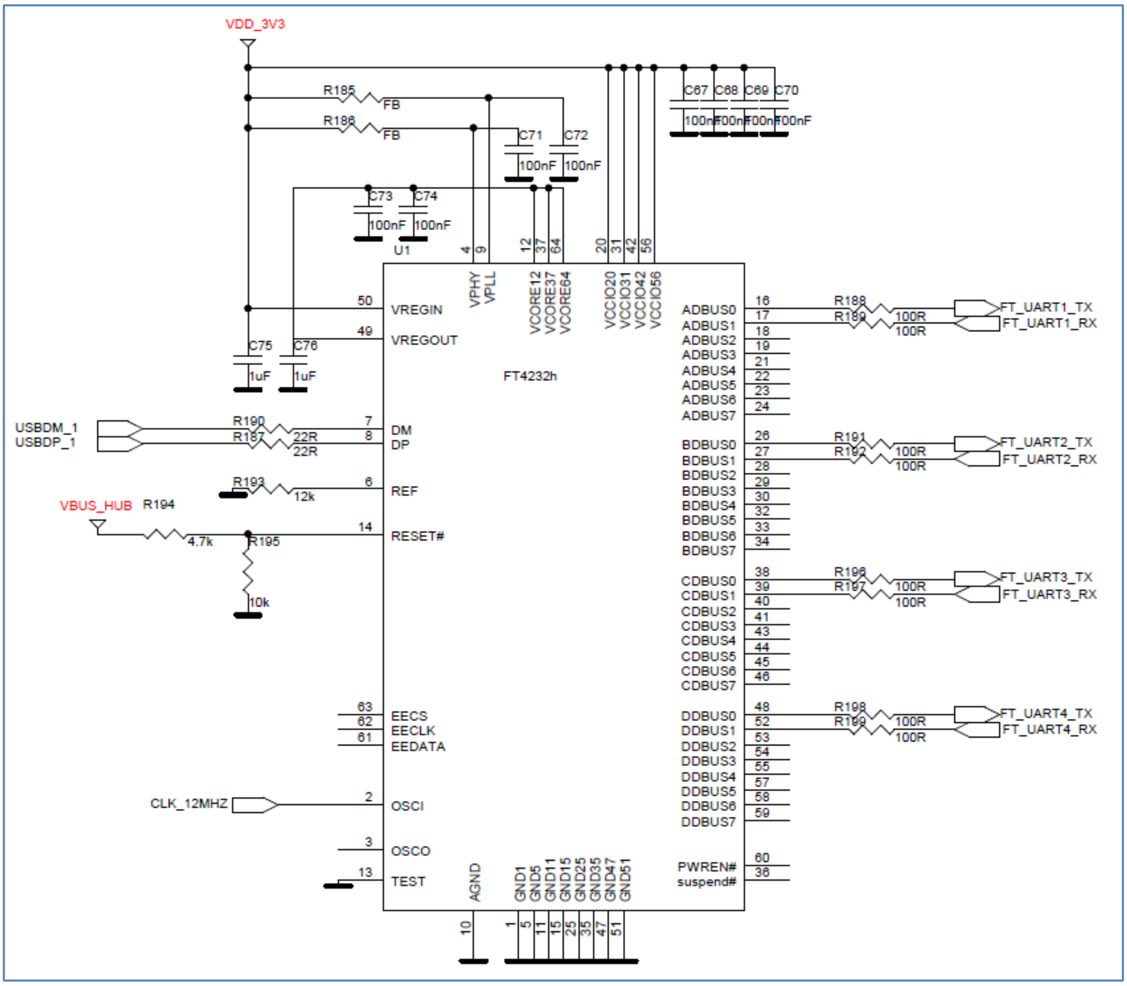


Figure 106: FTDI Chip for USB UART to DUTs 1, 2, 3 and 4

The USB HUB provides 5 V input for the 3.3 V LDO and USB input-signals to the four Quad FTDI chips.

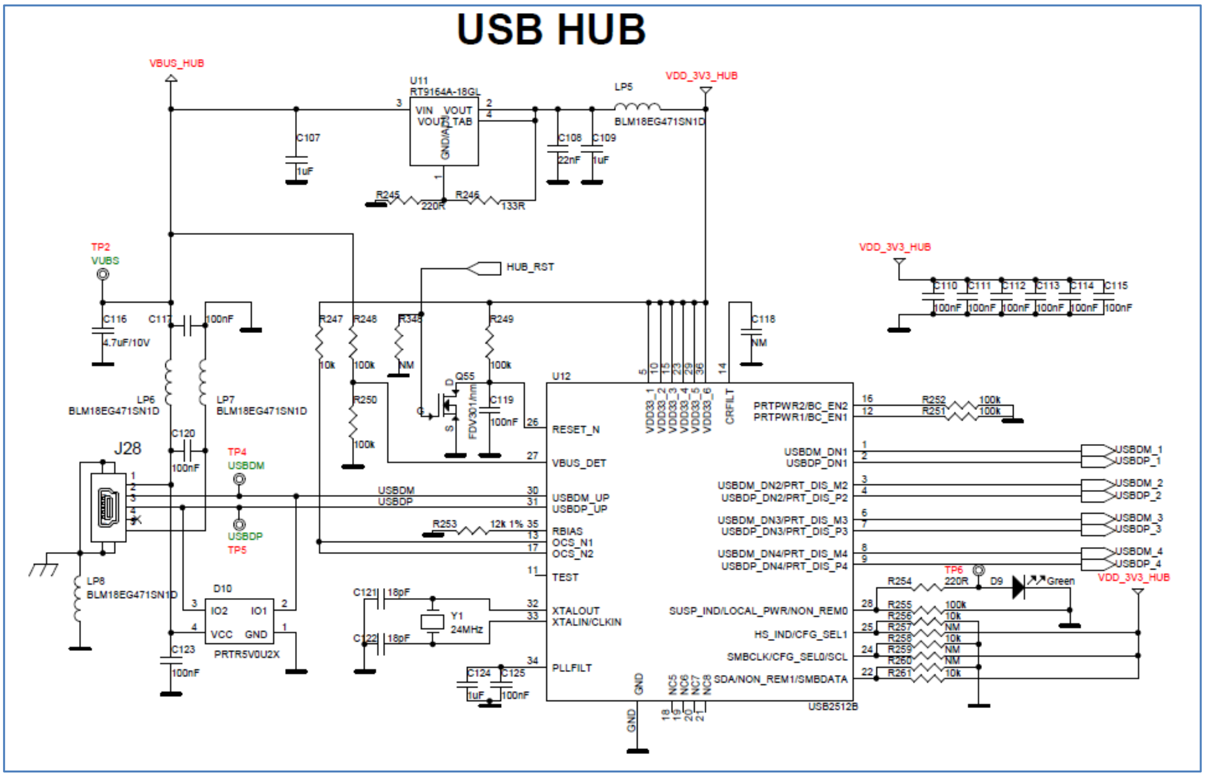


Figure 107: Quad USB HUB

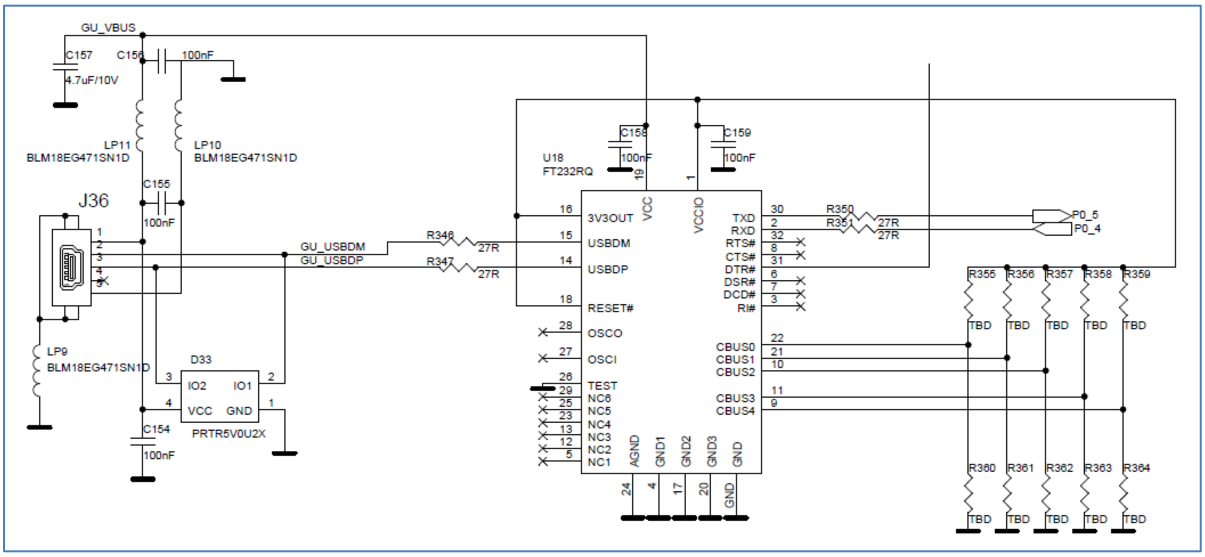


Figure 108: Golden Unit - Dedicated USB Port and FTDI Chip

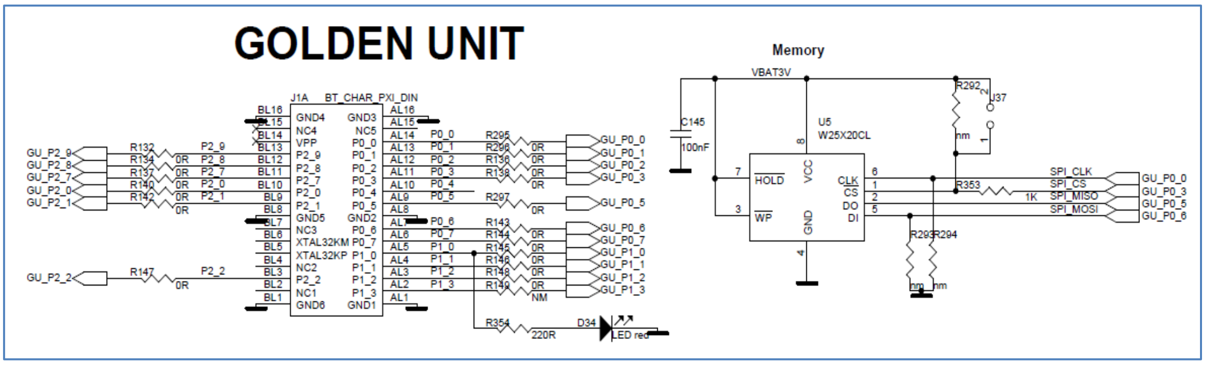


Figure 109: Golden Unit - GU LED and SPI Flash Memory

The Golden Unit SW (`prod_test_GU.bin`) is programmed into the SPI Flash memory mounted on the PLT hardware and is loaded into the GU's system RAM when powered on.

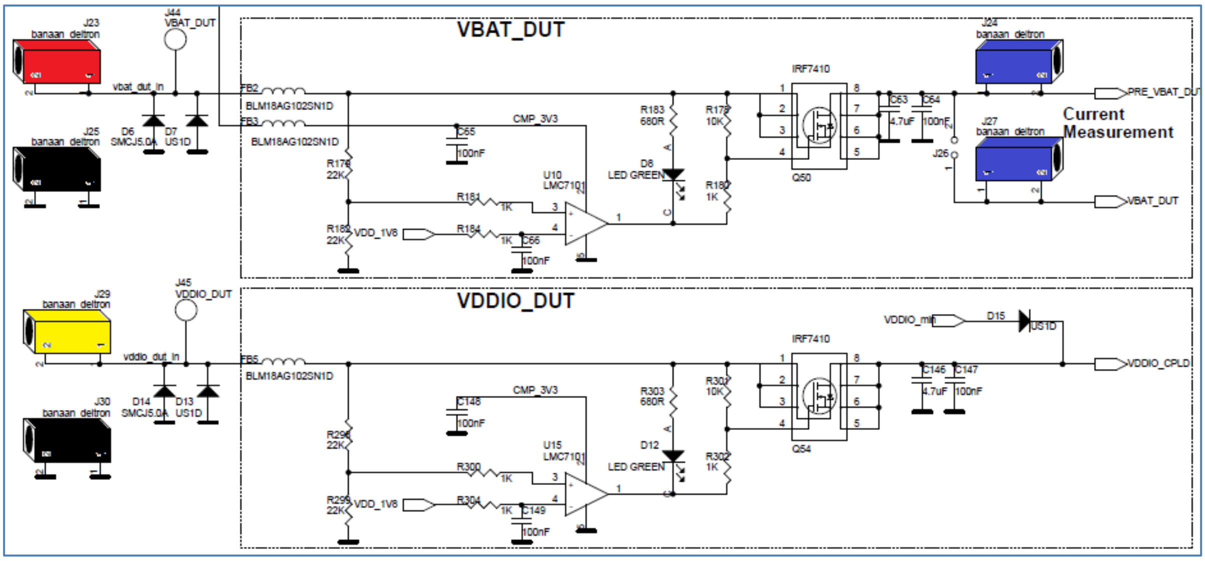


Figure 110: VBAT_DUT and VDDIO Supplies

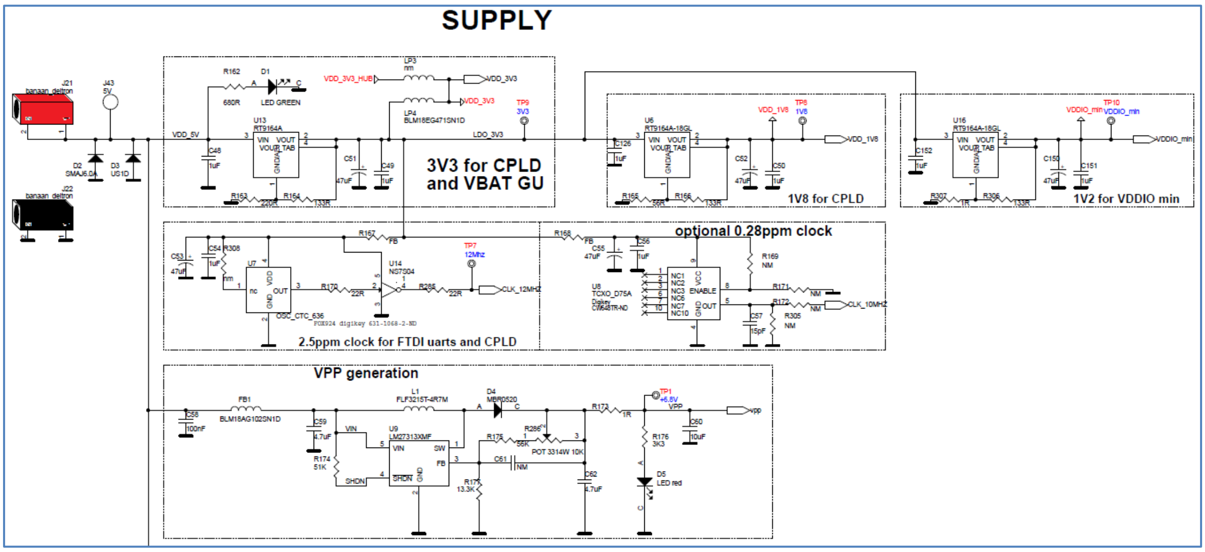


Figure 111: GU Supply and VPP Generation

Appendix C Hardware Modifications PLT Version D

In the PLT hardware version D, there is a small modification. Resistor R365 (10 kΩ) and jumper J47 are added in series to the GU reset circuit.

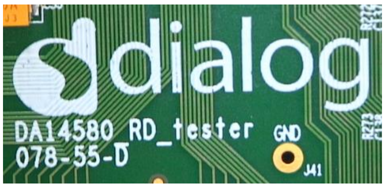


Figure 112: DA14580_RD_tester Version D

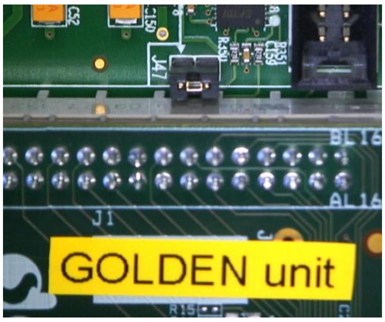


Figure 113: Jumper J47 Added Next to Golden Unit Socket

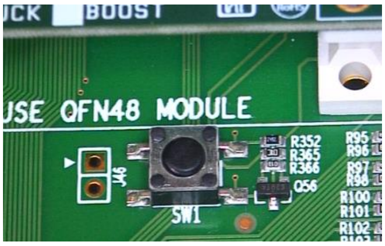


Figure 114: R365 (10 kΩ) Added Next to Reset Button

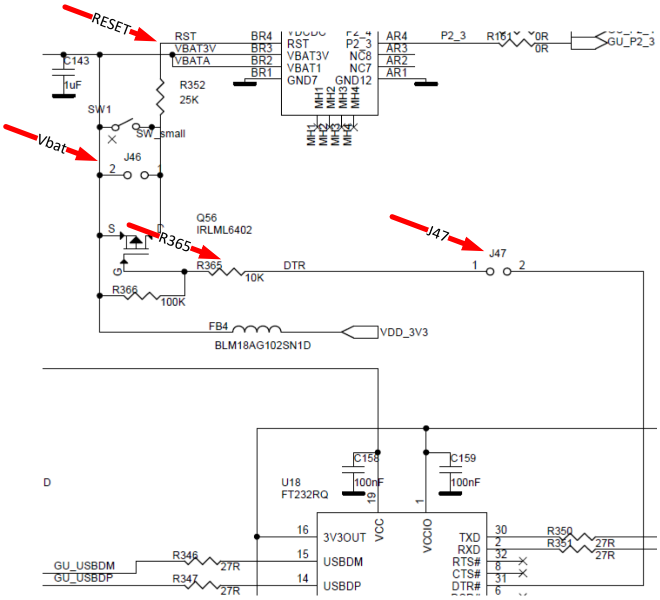


Figure 115: R365, J47 and RESET Shown in Electrical Schematic

Appendix D Application Hardware Design Considerations

When the Production Line Tool (PLT) is used, one should be aware of the following items:

- One could consider adding additional pads to the design for future debugging, not related to PLT, like pins for SWD
- Pads are, in most cases, placed on the rear side of the circuit board. They should be gold plated. Dimensions of these pads are crucial and have to do with the stability and accuracy of the pogo-pins that connect to the PLT HW. They should not be designed too critical. Long pogo pins might bend during production testing
- Optionally, holes can be added for guiding-pins that fit on the test jig used for the PCB or panel
- Orientation of the antenna used on the application board will impact the RSSI-value.
  When panels are used, this RSSI will vary, dependent on the distance to the GU antenna on the PLT. In the PLT software an RSSI-offset can be added for each DUT location to compensate for these differences

More reference documentation is available on Renesas website:

AN-B-087: DA1470x HW Guidelines https://www.renesas.com/eu/en/document/apn/b-087-da1470x-hw-guidelines?language=en&r=1614971

Appendix E Suggestions about Hardware and Cabling

When connecting the PLT to the DUTs, special care should be taken regarding cabling.

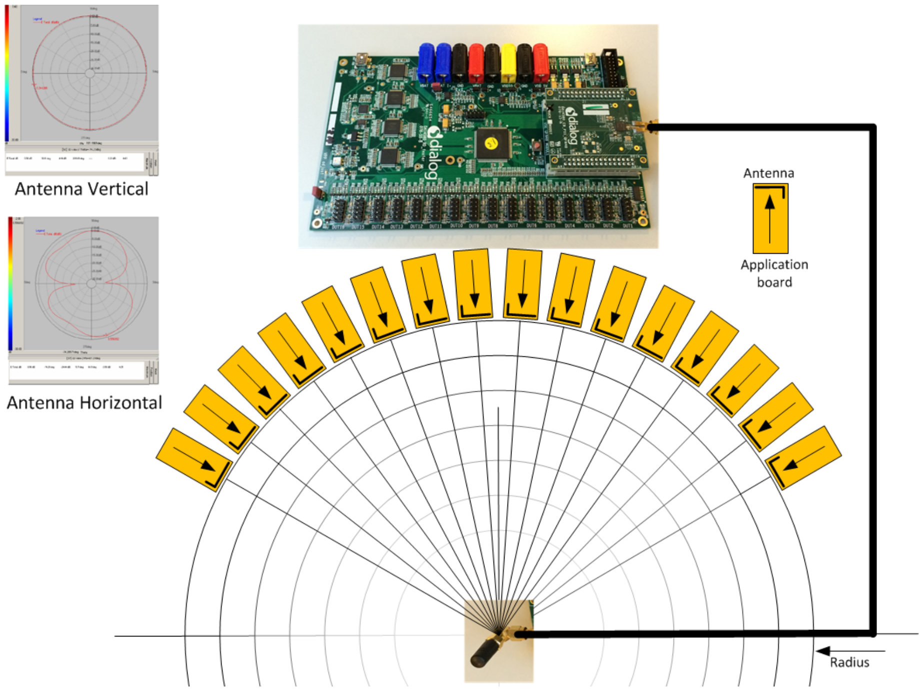


Figure 116: Possible Solution of Antenna on Cable and Fixed Radius of DUTs to Antenna

The user should realize that the PLT system is equipped with RF transmitters and receivers. These parts may induce noise on hardware and cables. Take note of the following:

- The direction of the GU antenna to the DUT antenna will influence the RSSI value
- The distance of the DUT antenna to the GU antenna (radius) will influence the RSSI value
- The control lines from the PLT to the DUTs must be kept as short as possible
- A vertical GU antenna has different characteristics from a horizontal one, see Figure 116

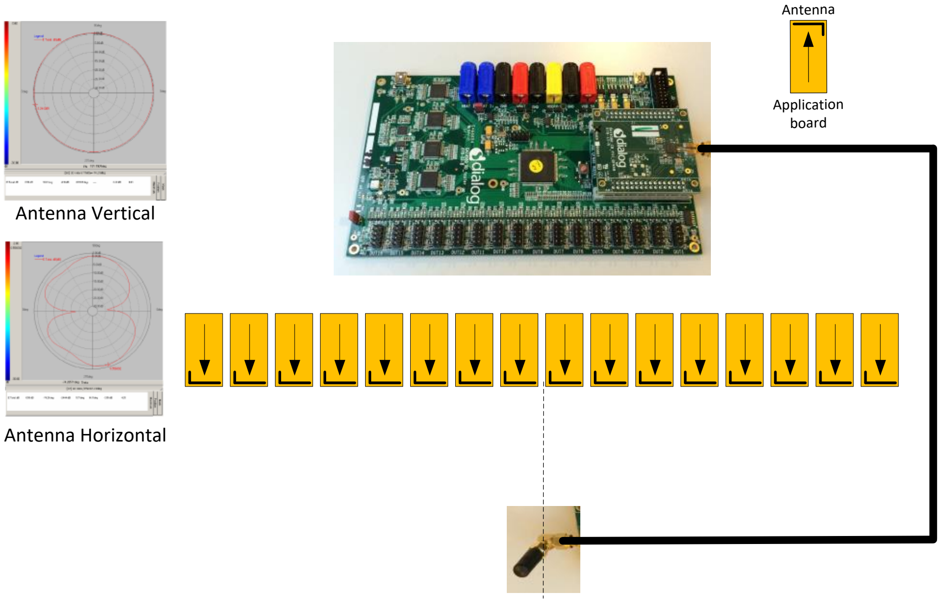


Figure 117: Possible Solution of Antenna on Cable and DUTs Put in Line

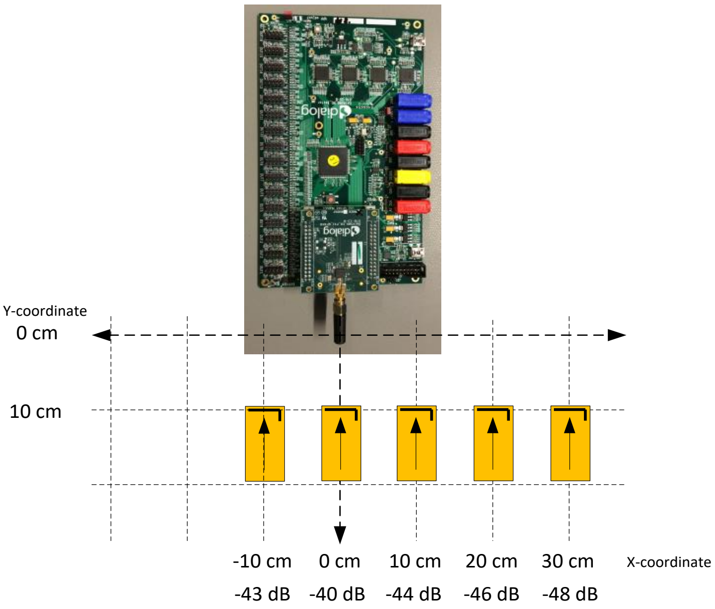


Figure 118: Example Locations and RSSI Readouts of Horizontal Antenna

Figure 118 shows the measured values from Table 64.

Table 64: RF Test RSSI Results

Test	Distance (cm)	Offset (cm)	RSSI (dBm)	Description
1	10	0	-40	DUT and GU boards are inline.
2	10	-10	-43	DUT moved 10 cm to the left relative to the GU.
3	10	10	-44	DUT moved 10 cm to the right relative to the GU.
4	10	20	-46	DUT moved 20 cm to the right relative to the GU.

Test	Distance (cm)	Offset (cm)	RSSI (dBm)	Description
5	10	30	-48	DUT moved 30 cm to the right relative to the GU.
6	10	normal	-40	DUT and GU boards are inline, functioning normally.
7	10	defect 1	-60 ~ -70	Coupling capacitor not soldered well, missing, or damaged.
8	10	defect 2	~ -60	Short-circuited shunt matching inductor (e.g., solder bridge)
9	10	defect 3	< -100	32 MHz crystal oscillator not working well. Received packets ~ 0.
Golden Unit output power = 0 dBm				

For more details on the RF setup, see Ref. [1] and Appendix F.

Appendix F RF Path Losses Calibration

To accurately perform radiated tests for 16 DUTs using the Golden Unit or a Bluetooth® LE tester, one should calibrate the setup to know what RSSI value can be expected for non-problematic devices. Because the distance and the position of each of the 16 DUTs to the GU RF antenna are different, the calibration process calculates the different path losses to compensate for these differences. The calculated values are applied to the Production Line Tool configuration as RF path losses (DA1470x - Path Losses per DUT Figure 55), which are added in the RSSI result.

This section describes the process to calculate the RF path losses for each different DUT position.

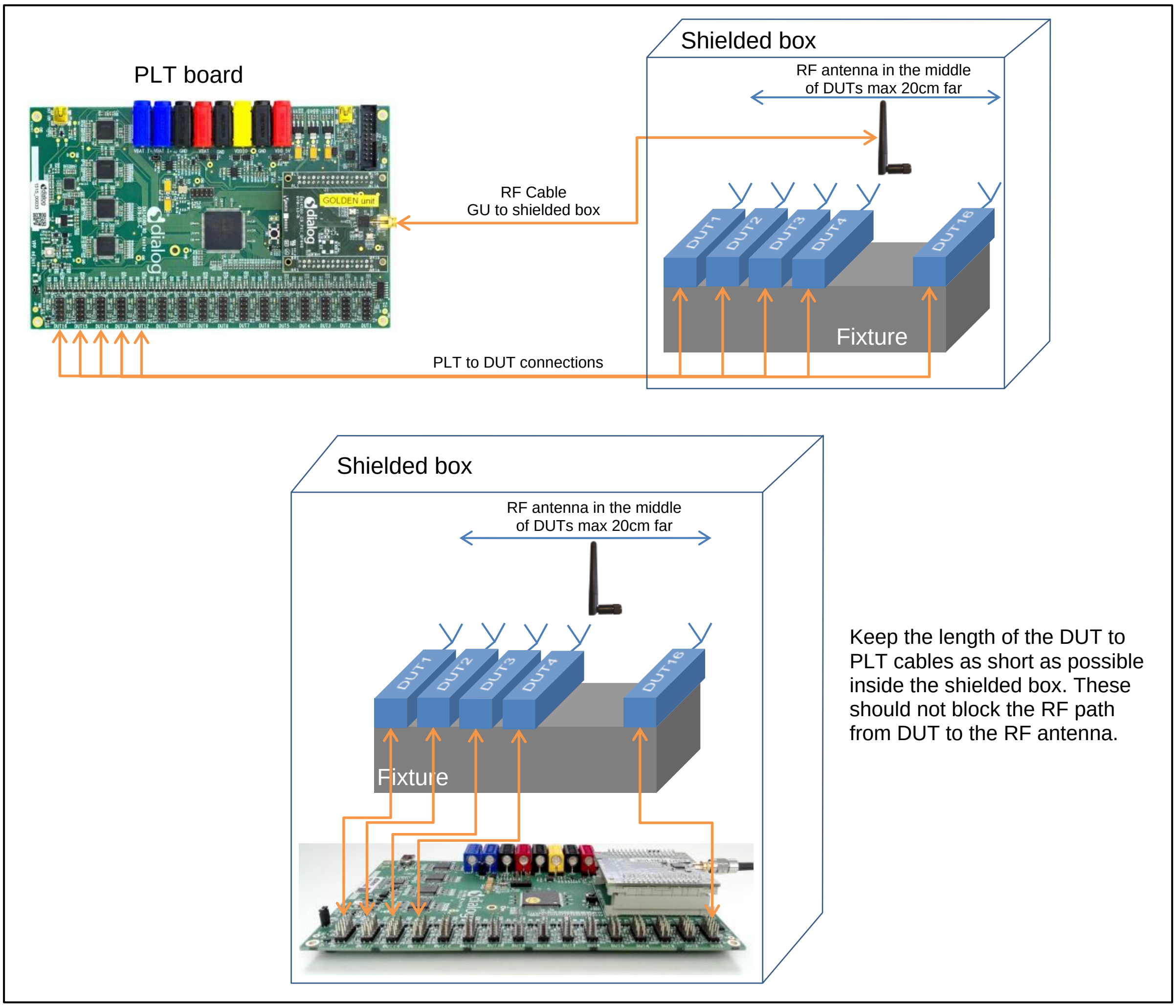
F.1 Prerequisites

Table 65 illustrates the prerequisites needed for performing the RF path loss calibration procedure.

Table 65: Prerequisites

#	Requirements	Description
1	1 PLT board	1 PLT board with Golden Unit. Power supply for the PLT. USB cables for the PLT to the PC.
2	<10 PCBA with 16 DUTs each	At least 10 PCBAs. The more PCBAs used the better. The PCBAs selected should all work as good as possible. If a fault device is identified on a PCBA that PCBA should be replaced.
3	1 shielded box	It must be big enough to fit the PCBA and the fixture. It should have one SMA female to female connector and a small hole to pass the DUT to PLT cable connections.
4	2 RF cables (1 optional)	One cable to be used from the PLT GU to the shielded box. One more cable to be used from the shielded box to the RF antenna. (This is optional since the RF antenna can be directly mounted into the shielded box RF SMA connector). The cables should have low attenuation at 2.5 GHz range (<2 dB) and high shielding effectiveness (>60 dB). Proposed cables are from Radiall. Cables datasheet: https://www.radiall.com/media/files/RFCableAssemblies%20D1C004XEe.pdf ● Flexible cable 2.6/50 D (RD316) P/N: C291 185 067 ● Flexible cable 2/50 D (124416 type) P/N: C291 146 087 ● Flexible cable 2.6/50 D (ECO316D: alternative to RD316) P/N: C291 999 905 ● Flexible cable 5/50 D (ECO142: alternative to RG142) P/N: C291 325 290 ● Flexible cable 5/50 D (Power 142: alternative to RG142) P/N: C291 325 270 ● Flexible cable 6/50 D (ECO230) P/N: C291 326 490
5	DUT fixture	A fixture to be placed inside the shielded box to easily connect the PCBAs to the PLT.

F.2 Setup


PLT board
Shielded box
RF antenna in the middle of DUTs max 20cm far
RF Cable GU to shielded box
DUT1
DUT2
DUT3
DUT4
DUT16
Fixture
PLT to DUT connections
Shielded box
RF antenna in the middle of DUTs max 20cm far
DUT1
DUT2
DUT3
DUT4
DUT16
Fixture

Keep the length of the DUT to PLT cables as short as possible inside the shielded box. These should not block the RF path from DUT to the RF antenna.

Figure 119: Setup Diagram

Table 66: RF Path Losses Calibration Setup

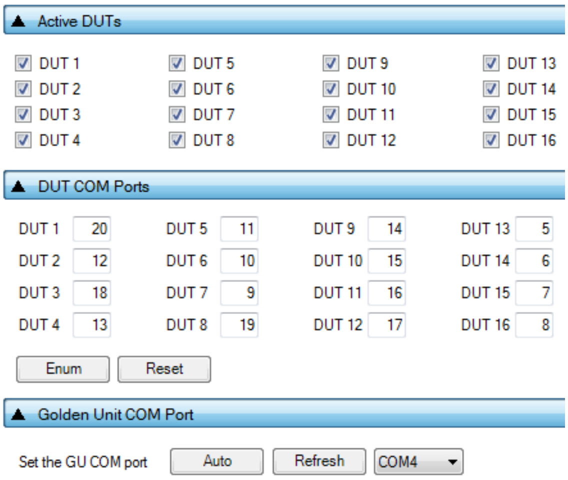
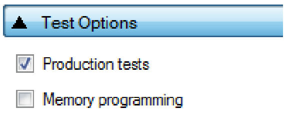
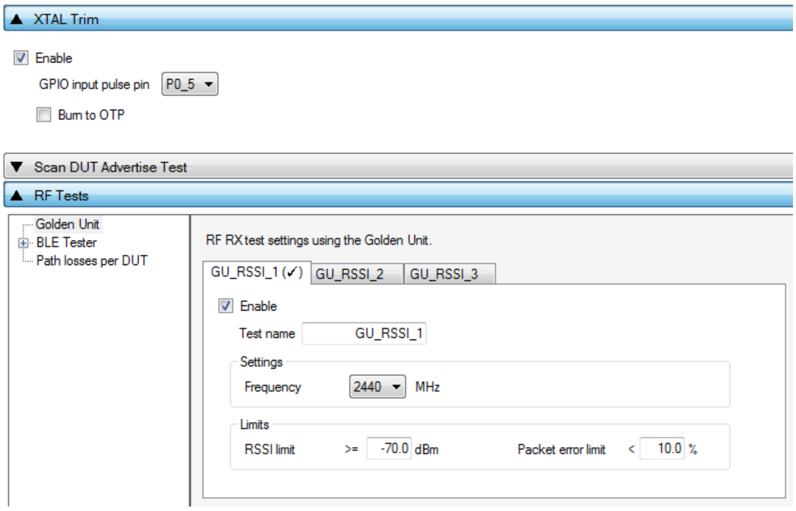
Item	Description
PLT board	The PLT board could either be outside or inside the shielded box, as shown in Figure 119, depending on the fixture setup.
DUTs	The DUT antennas should point the RF antenna.
PLT to DUT connections	The PLT to DUT cable connections should be as short as possible. Also, the cables should not block the RF path from DUT to the RF antenna.
Shielded box	The shielded box must be big enough to fit the PCBA and the fixture and the PLT if it is inside.
RF cable - PLT to shielded box.	This type of cable must have good shielding and low attenuation. Effective shielding must be >60 dB. Check Table 65 for the best proposals. If multiple production lines are used, RF cable shielding is important to avoid disturbance from other close by PLTs. However, if other PLTs are far away (>3 m) other cables can be used. ● FLEXIBLE CABLE 2.6/50 S (RG316 - KX22A) P/N: C291 170 007

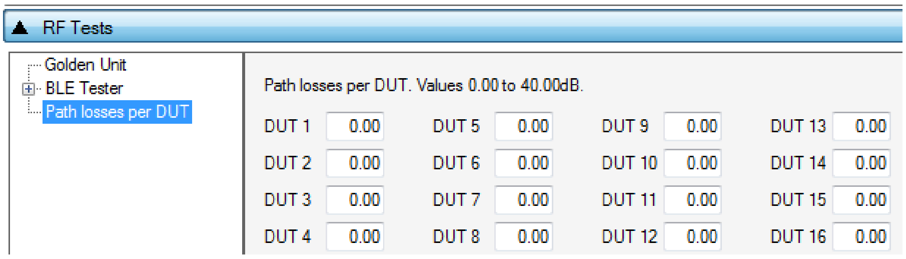
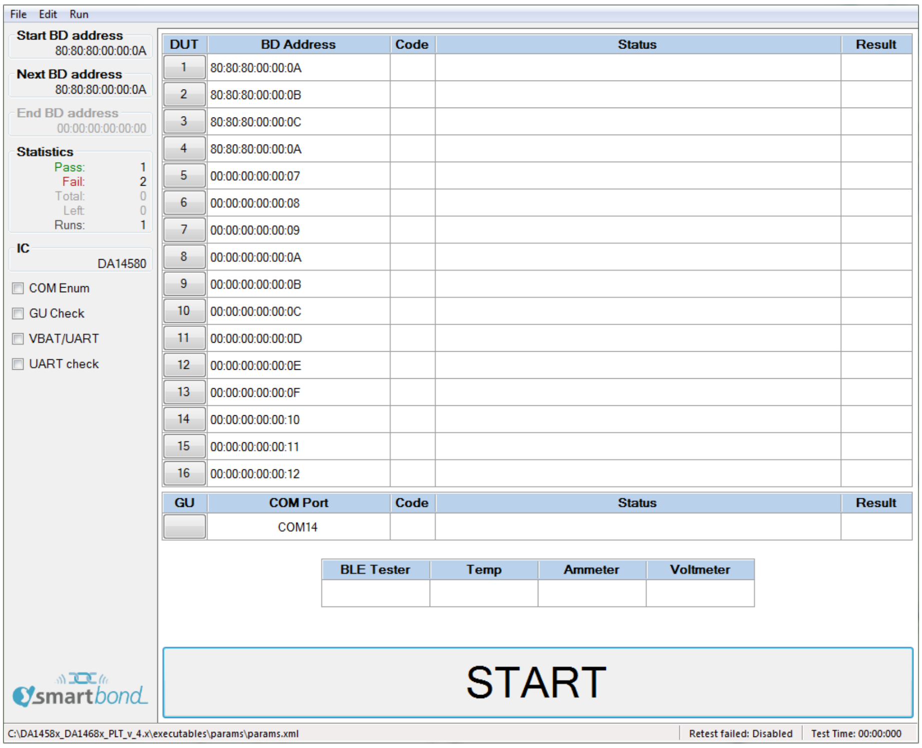
Item	Description
RF cable – Shielded box to RF antenna (optional).	Due to the close distance between the DUT antennas and the RF antenna cable, cable shielding must be good, at least 60 dB. Check Table 65 for proposals. This cable can be optional. The RF antenna may be mounted directly onto the SMA RF connector inside the shielded box. In that case, the fixture and the DUTs should be placed appropriately (less than 20 cm).
RF antenna	The RF antenna can be any good Wi-Fi antenna that operates at 2.5 GHz. It should be placed in a vertical position as shown in Figure 119. The distance to the DUTs should not be larger than 20 cm. It should be placed in the middle of the DUTs (in front of DUT 8). Bear in mind that the Anritsu MT8852B Bluetooth® LE tester cannot perform TX measurements if the signal received at its antenna is less than -50 dBm. For a good measurement, the signal reaching its antenna should be greater than -40 dBm. Therefore, the distance and the placement between the DUTs and the RF antenna are very important. Trial-and-error test should be carried out until the optimal antenna position is found. The RF antenna placement should be very stable. After the optimal position is found, it should be fixed into position and not able to move again.
DUT fixture	The fixture position should be fixed compared to the RF antenna. The fixture should not move in any way to keep the distance between the DUTs, and the RF antenna fixed.

F.3 Procedure

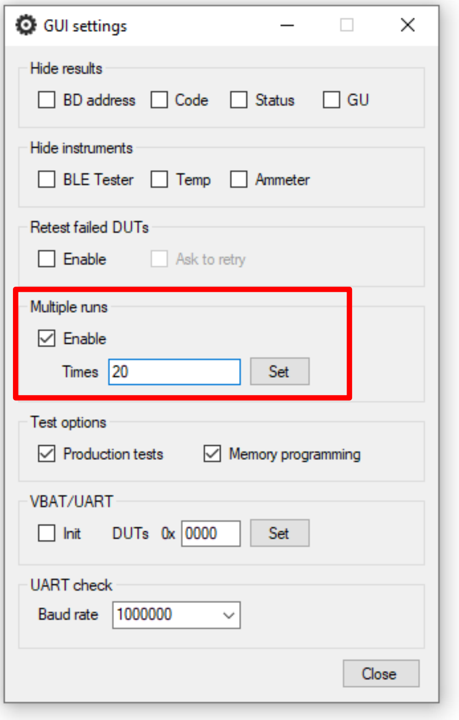
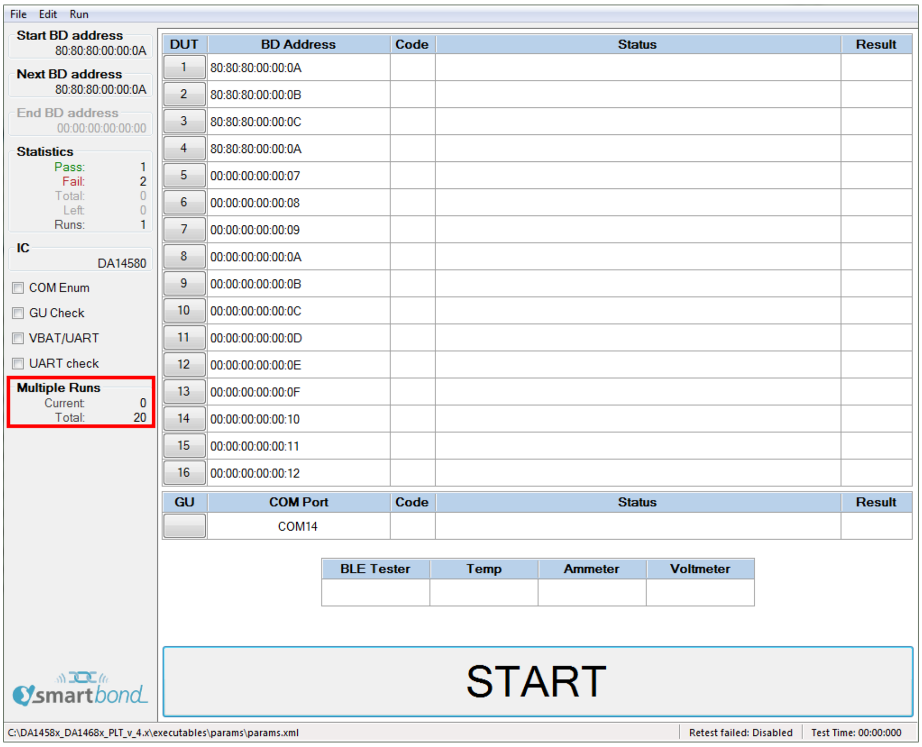
Table 67 describes the steps to follow to calculate the RF path losses for each different DUT position.

Table 67: Procedure Steps

#	Step	Description
1	Open the `SmartBond_PLT_CFG.exe` PLT configuration executable.	User should configure the PLT so only XTAL trim (without OTP burn) and one Golden Unit RF test is enabled.
2	Enable all 16 DUTs	Select the GU COM port and Enumerate the DUT COM port numbers.  Active DUTs ☑ DUT 1 ☑ DUT 2 ☑ DUT 3 ☑ DUT 4 ☑ DUT 5 ☑ DUT 6 ☑ DUT 7 ☑ DUT 8 ☑ DUT 9 ☑ DUT 10 ☑ DUT 11 ☑ DUT 12 ☑ DUT 13 ☑ DUT 14 ☑ DUT 15 ☑ DUT 16 DUT COM Ports DUT 1 20, DUT 2 12, DUT 3 18, DUT 4 13 DUT 5 11, DUT 6 10, DUT 7 9, DUT 8 19 DUT 9 14, DUT 10 15, DUT 11 16, DUT 12 17 DUT 13 5, DUT 14 6, DUT 15 7, DUT 16 8 Enum Reset Golden Unit COM Port Set the GU COM port Auto Refresh COM4
3	Enable only the Production tests	Under the General tab, enable only the Production tests and disable the Memory programming.  Test Options ☑ Production tests □ Memory programming
4	Enable XTAL trim (no OTP burn) and one Golden Unit RF test at middle band at 2440 MHz.	RSSI and PER limits do not matter. These should be set to a small value (less than -70 dBm) as shown below. Ideally, we want the limits to be set to a value that all RF tests PASS.  XTAL Trim ☑ Enable GPIO input pulse pin P0_5 □ Burn to OTP Scan DUT Advertise Test RF Tests Golden Unit BLE Tester Path losses per DUT RF RX test settings using the Golden Unit. GU_RSSI_1 (✓) GU_RSSI_2 GU_RSSI_3 ☑ Enable Test name GU_RSSI_1 Settings Frequency 2440 MHz Limits RSSI limit >= -70.0 dBm Packet error limit < 10.0 %

<table>
<tr><th>#</th><th>Step</th><th>Description</th></tr>
<tr><td>5</td><td>Set all DUT path losses to 0 dB.</td><td>
RF Tests
Golden Unit
BLE Tester
Path losses per DUT
Path losses per DUT. Values 0.00 to 40.00dB.
<table>
<tr><td>DUT 1</td><td>0.00</td><td>DUT 5</td><td>0.00</td><td>DUT 9</td><td>0.00</td><td>DUT 13</td><td>0.00</td></tr>
<tr><td>DUT 2</td><td>0.00</td><td>DUT 6</td><td>0.00</td><td>DUT 10</td><td>0.00</td><td>DUT 14</td><td>0.00</td></tr>
<tr><td>DUT 3</td><td>0.00</td><td>DUT 7</td><td>0.00</td><td>DUT 11</td><td>0.00</td><td>DUT 15</td><td>0.00</td></tr>
<tr><td>DUT 4</td><td>0.00</td><td>DUT 8</td><td>0.00</td><td>DUT 12</td><td>0.00</td><td>DUT 16</td><td>0.00</td></tr>
</table></td></tr>
<tr><td>6</td><td>Backup CSV log file.</td><td>Go to <code>SmartBond_PLT_v_x.x\executables\logs</code> and back up the today’s CSV file. For example, if today’s CSV file is <code>Test_station_1_20180306_csv_results.csv</code>
rename it to:
<code>Test_station_1_20180306_csv_results_BackUp.csv.</code>
Doing so ensures a new CSV file is created at the next PLT test run.</td></tr>
<tr><td>7</td><td>Place the 1st PCBA</td><td>Place the 1st PCBA into the fixture inside the shielded box.</td></tr>
<tr><td>8</td><td>Open <code>SmartBond_PLT_GUI.exe</code></td><td>

Start BD address 80:80:80:00:00:0A
Next BD address 80:80:80:00:00:0A
End BD address 00:00:00:00:00:00
Statistics
Pass: 1
Fail: 2
Total: 0
Left: 0
Runs: 1
IC DA14580
□ COM Enum
□ GU Check
□ VBAT/UART
□ UART check
<table>
<tr><th>DUT</th><th>BD Address</th><th>Code</th><th>Status</th><th>Result</th></tr>
<tr><td>1</td><td>80:80:80:00:00:0A</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>80:80:80:00:00:0B</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>80:80:80:00:00:0C</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>80:80:80:00:00:0A</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>00:00:00:00:00:07</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>00:00:00:00:00:08</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td>00:00:00:00:00:09</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td>00:00:00:00:00:0A</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td>00:00:00:00:00:0B</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>00:00:00:00:00:0C</td><td></td><td></td><td></td></tr>
<tr><td>11</td><td>00:00:00:00:00:0D</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td>00:00:00:00:00:0E</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td>00:00:00:00:00:0F</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td>00:00:00:00:00:10</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td>00:00:00:00:00:11</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td>00:00:00:00:00:12</td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><th>GU</th><th>COM Port</th><th>Code</th><th>Status</th><th>Result</th></tr>
<tr><td></td><td>COM14</td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><th>BLE Tester</th><th>Temp</th><th>Ammeter</th><th>Voltmeter</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
C:\DA1458x_DA1468x_PLT_v_4.x\executables\params\params.xml
Retest failed: Disabled
Test Time: 00:00:000</td></tr>
</table>

#	Step	Description
9	Go to Edit->Settings	Select the Multiple runs. Set the Times to 20. Click Set and Close.  GUI settings Hide results: □ BD address □ Code □ Status □ GU Hide instruments: □ BLE Tester □ Temp □ Ammeter Retest failed DUTs: □ Enable □ Ask to retry Multiple runs: ☑ Enable; Times 20 [Set] Test options: ☑ Production tests ☑ Memory programming VBAT/UART: □ Init DUTs 0x 0000 [Set] UART check: Baud rate 1000000 [Close]
10	In the `SmartBond_PLT_GUI.exe` the Multiple Runs should be shown on the left panel	
11	Press START	Click START and wait for the 20 tests to be performed.
12	Backup CSV log file	Go to SmartBond_PLT_v_x.x\executables\logs and back up the CSV file for the 1st PCBA. For example, if today's CSV file is Test_station_1_20180306_csv_results.csv rename it to Test_station_1_20180306_csv_results_PCBA_1.csv.

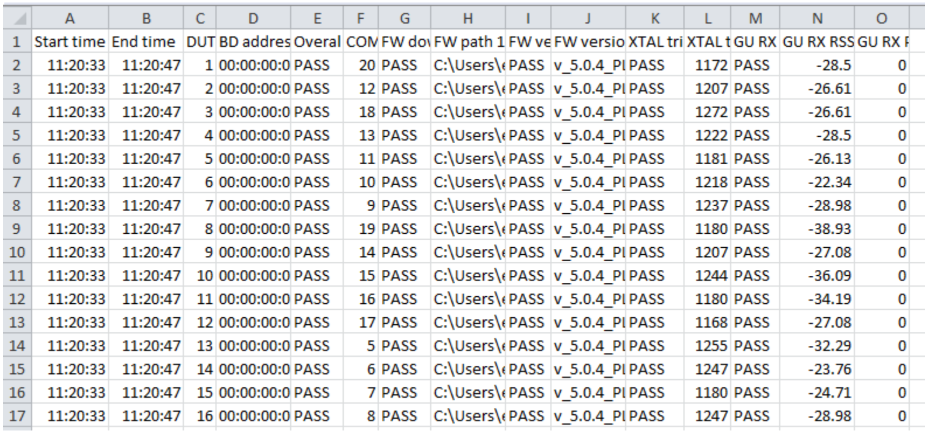
Start BD address 80:80:80:00:00:0A
Next BD address 80:80:80:00:00:0A
End BD address 00:00:00:00:00:00
Statistics
Pass: 1
Fail: 2
Total: 0
Left: 0
Runs: 1
IC DA14580
□ COM Enum
□ GU Check
□ VBAT/UART
□ UART check
Multiple Runs
Current: 0
Total: 20

DUT	BD Address	Code	Status	Result
1	80:80:80:00:00:0A			
2	80:80:80:00:00:0B			
3	80:80:80:00:00:0C			
4	80:80:80:00:00:0A			
5	00:00:00:00:00:07			
6	00:00:00:00:00:08			
7	00:00:00:00:00:09			
8	00:00:00:00:00:0A			
9	00:00:00:00:00:0B			
10	00:00:00:00:00:0C			
11	00:00:00:00:00:0D			
12	00:00:00:00:00:0E			
13	00:00:00:00:00:0F			
14	00:00:00:00:00:10			
15	00:00:00:00:00:11			
16	00:00:00:00:00:12			

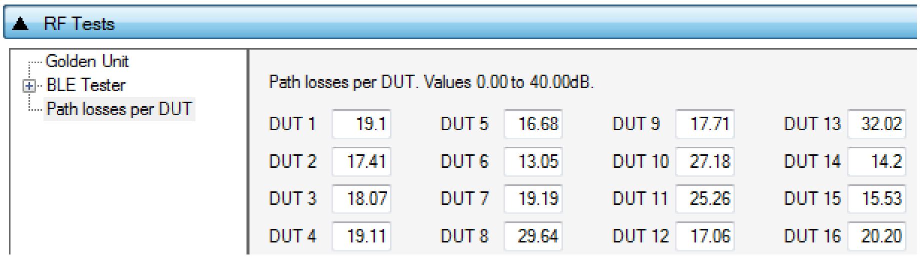
GU	COM Port	Code	Status	Result
	COM14			

BLE Tester	Temp	Ammeter	Voltmeter

C:\DA1458x_DA1468x_PLT_v_4.x\executables\params\params.xml | Retest failed: Disabled | Test Time: 00:00:000

<table>
<tr><th>#</th><th>Step</th><th>Description</th></tr>
<tr><td>13</td><td>Place the 2nd PCBA</td><td>Place the 2nd PCBA into the fixture inside the shielded box.</td></tr>
<tr><td>14</td><td>Repeat steps</td><td>Repeat the procedure from step 9 to step 13 for all 10 PCBAs.</td></tr>
<tr><td>15</td><td>Check CSV results</td><td>At the end, 10 CSV logs files should exist. For example,
Test_station_1_20180306_csv_results_PCBA_1.csv.
Test_station_1_20180306_csv_results_PCBA_2.csv.
…
Test_station_1_20180306_csv_results_PCBA_10.csv.
Each CSV file should have 16 DUTs * 20 Tests = 320 lines + 1 CSV header = 321 lines. Example:

<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th><th>G</th><th>H</th><th>I</th><th>J</th><th>K</th><th>L</th><th>M</th><th>N</th><th>O</th></tr>
<tr><td>1</td><td>Start time</td><td>End time</td><td>DUT</td><td>BD addres</td><td>Overal</td><td>COM</td><td>FW dov</td><td>FW path 1</td><td>FW ve</td><td>FW versio</td><td>XTAL tri</td><td>XTAL t</td><td>GU RX</td><td>GU RX RSS</td><td>GU RX F</td></tr>
<tr><td>2</td><td>11:20:33</td><td>11:20:47</td><td>1</td><td>00:00:00:0</td><td>PASS</td><td>20</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1172</td><td>PASS</td><td>-28.5</td><td>0</td></tr>
<tr><td>3</td><td>11:20:33</td><td>11:20:47</td><td>2</td><td>00:00:00:0</td><td>PASS</td><td>12</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1207</td><td>PASS</td><td>-26.61</td><td>0</td></tr>
<tr><td>4</td><td>11:20:33</td><td>11:20:47</td><td>3</td><td>00:00:00:0</td><td>PASS</td><td>18</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1272</td><td>PASS</td><td>-26.61</td><td>0</td></tr>
<tr><td>5</td><td>11:20:33</td><td>11:20:47</td><td>4</td><td>00:00:00:0</td><td>PASS</td><td>13</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1222</td><td>PASS</td><td>-28.5</td><td>0</td></tr>
<tr><td>6</td><td>11:20:33</td><td>11:20:47</td><td>5</td><td>00:00:00:0</td><td>PASS</td><td>11</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1181</td><td>PASS</td><td>-26.13</td><td>0</td></tr>
<tr><td>7</td><td>11:20:33</td><td>11:20:47</td><td>6</td><td>00:00:00:0</td><td>PASS</td><td>10</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1218</td><td>PASS</td><td>-22.34</td><td>0</td></tr>
<tr><td>8</td><td>11:20:33</td><td>11:20:47</td><td>7</td><td>00:00:00:0</td><td>PASS</td><td>9</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1237</td><td>PASS</td><td>-28.98</td><td>0</td></tr>
<tr><td>9</td><td>11:20:33</td><td>11:20:47</td><td>8</td><td>00:00:00:0</td><td>PASS</td><td>19</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1180</td><td>PASS</td><td>-38.93</td><td>0</td></tr>
<tr><td>10</td><td>11:20:33</td><td>11:20:47</td><td>9</td><td>00:00:00:0</td><td>PASS</td><td>14</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1207</td><td>PASS</td><td>-27.08</td><td>0</td></tr>
<tr><td>11</td><td>11:20:33</td><td>11:20:47</td><td>10</td><td>00:00:00:0</td><td>PASS</td><td>15</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1244</td><td>PASS</td><td>-36.09</td><td>0</td></tr>
<tr><td>12</td><td>11:20:33</td><td>11:20:47</td><td>11</td><td>00:00:00:0</td><td>PASS</td><td>16</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1180</td><td>PASS</td><td>-34.19</td><td>0</td></tr>
<tr><td>13</td><td>11:20:33</td><td>11:20:47</td><td>12</td><td>00:00:00:0</td><td>PASS</td><td>17</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1168</td><td>PASS</td><td>-27.08</td><td>0</td></tr>
<tr><td>14</td><td>11:20:33</td><td>11:20:47</td><td>13</td><td>00:00:00:0</td><td>PASS</td><td>5</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1255</td><td>PASS</td><td>-32.29</td><td>0</td></tr>
<tr><td>15</td><td>11:20:33</td><td>11:20:47</td><td>14</td><td>00:00:00:0</td><td>PASS</td><td>6</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1247</td><td>PASS</td><td>-23.76</td><td>0</td></tr>
<tr><td>16</td><td>11:20:33</td><td>11:20:47</td><td>15</td><td>00:00:00:0</td><td>PASS</td><td>7</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1180</td><td>PASS</td><td>-24.71</td><td>0</td></tr>
<tr><td>17</td><td>11:20:33</td><td>11:20:47</td><td>16</td><td>00:00:00:0</td><td>PASS</td><td>8</td><td>PASS</td><td>C:\Users\</td><td>PASS</td><td>v_5.0.4_Pl</td><td>PASS</td><td>1247</td><td>PASS</td><td>-28.98</td><td>0</td></tr>
</table></td></tr>
<tr><td>16</td><td>Get the average for each DUT in each CSV file</td><td>In each CSV file, get the average GU RX RSSI value for each DUT. Apply the following formulas to each of the 10 CSV files.
<table>
<tr><th>DUT #</th><th>Formula</th><th>Example Result</th></tr>
<tr><td>DUT 1</td><td>=SUMIF($C$2:$C$321,<b>1</b>, $N$2:$N$321)/20</td><td>-28.5015</td></tr>
<tr><td>DUT 2</td><td>=SUMIF($C$2:$C$321,<b>2</b>, $N$2:$N$321)/20</td><td>-26.609</td></tr>
<tr><td>DUT 3</td><td>=SUMIF($C$2:$C$321,<b>3</b>, $N$2:$N$321)/20</td><td>-26.8685</td></tr>
<tr><td>DUT 4</td><td>=SUMIF($C$2:$C$321,<b>4</b>, $N$2:$N$321)/20</td><td>-28.406</td></tr>
<tr><td>DUT 5</td><td>=SUMIF($C$2:$C$321,<b>5</b>, $N$2:$N$321)/20</td><td>-26.346</td></tr>
<tr><td>DUT 6</td><td>=SUMIF($C$2:$C$321,<b>6</b>, $N$2:$N$321)/20</td><td>-22.5515</td></tr>
<tr><td>DUT 7</td><td>=SUMIF($C$2:$C$321,<b>7</b>, $N$2:$N$321)/20</td><td>-28.98</td></tr>
<tr><td>DUT 8</td><td>=SUMIF($C$2:$C$321,<b>8</b>, $N$2:$N$321)/20</td><td>-39.283</td></tr>
<tr><td>DUT 9</td><td>=SUMIF($C$2:$C$321,<b>9</b>, $N$2:$N$321)/20</td><td>-27.08</td></tr>
<tr><td>DUT 10</td><td>=SUMIF($C$2:$C$321,<b>10</b>,$N$2:$N$321)/20</td><td>-36.607</td></tr>
<tr><td>DUT 11</td><td>=SUMIF($C$2:$C$321,<b>11</b>,$N$2:$N$321)/20</td><td>-34.19</td></tr>
<tr><td>DUT 12</td><td>=SUMIF($C$2:$C$321,<b>12</b>,$N$2:$N$321)/20</td><td>-26.7745</td></tr>
<tr><td>DUT 13</td><td>=SUMIF($C$2:$C$321,<b>13</b>,$N$2:$N$321)/20</td><td>-32.0095</td></tr>
<tr><td>DUT 14</td><td>=SUMIF($C$2:$C$321,<b>14</b>,$N$2:$N$321)/20</td><td>-23.784</td></tr>
<tr><td>DUT 15</td><td>=SUMIF($C$2:$C$321,<b>15</b>,$N$2:$N$321)/20</td><td>-24.71</td></tr>
<tr><td>DUT 16</td><td>=SUMIF($C$2:$C$321,<b>16</b>,$N$2:$N$321)/20</td><td>-29.168</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>#</th><th>Step</th><th>Description</th></tr>
<tr><td>17</td><td>Get all values to a new excel sheet.</td><td>Create a new excel sheet. Copy all values created at step 16 from the 10 CSV files to this new excel sheet. An example of all DUT average values from all 10 CSV files is given below. Only 2 decimal digits are shown.
<table>
<tr><th>DUT</th><th>PCBA 1</th><th>PCBA 2</th><th>PCBA 3</th><th>PCBA 4</th><th>PCBA 5</th><th>PCBA 6</th><th>PCBA 7</th><th>PCBA 8</th><th>PCBA 9</th><th>PCBA 10</th></tr>
<tr><td>1</td><td>-28.50</td><td>-28.30</td><td>-28.51</td><td>-27.56</td><td>-29.48</td><td>-28.59</td><td>-30.09</td><td>-29.33</td><td>-30.50</td><td>-30.13</td></tr>
<tr><td>2</td><td>-26.61</td><td>-26.07</td><td>-27.09</td><td>-26.56</td><td>-27.99</td><td>-27.38</td><td>-28.31</td><td>-27.64</td><td>-28.36</td><td>-28.11</td></tr>
<tr><td>3</td><td>-26.87</td><td>-26.71</td><td>-27.65</td><td>-26.71</td><td>-28.64</td><td>-27.84</td><td>-29.14</td><td>-28.43</td><td>-29.54</td><td>-29.14</td></tr>
<tr><td>4</td><td>-28.41</td><td>-27.98</td><td>-28.82</td><td>-27.87</td><td>-29.49</td><td>-29.24</td><td>-30.10</td><td>-29.61</td><td>-30.22</td><td>-29.36</td></tr>
<tr><td>5</td><td>-26.35</td><td>-25.69</td><td>-26.72</td><td>-26.58</td><td>-26.74</td><td>-26.02</td><td>-27.37</td><td>-26.75</td><td>-27.42</td><td>-27.20</td></tr>
<tr><td>6</td><td>-22.55</td><td>-21.60</td><td>-22.68</td><td>-21.81</td><td>-23.32</td><td>-22.33</td><td>-23.82</td><td>-23.77</td><td>-24.45</td><td>-24.19</td></tr>
<tr><td>7</td><td>-28.98</td><td>-28.59</td><td>-29.31</td><td>-28.49</td><td>-29.42</td><td>-28.63</td><td>-29.84</td><td>-28.95</td><td>-29.98</td><td>-29.73</td></tr>
<tr><td>8</td><td>-39.28</td><td>-39.06</td><td>-39.50</td><td>-38.95</td><td>-39.53</td><td>-39.38</td><td>-40.32</td><td>-40.11</td><td>-40.39</td><td>-39.90</td></tr>
<tr><td>9</td><td>-27.08</td><td>-26.64</td><td>-27.68</td><td>-27.07</td><td>-27.76</td><td>-27.28</td><td>-28.40</td><td>-27.51</td><td>-28.89</td><td>-28.75</td></tr>
<tr><td>10</td><td>-36.61</td><td>-35.74</td><td>-36.85</td><td>-36.05</td><td>-37.07</td><td>-37.04</td><td>-37.96</td><td>-37.59</td><td>-38.84</td><td>-38.03</td></tr>
<tr><td>11</td><td>-34.19</td><td>-33.57</td><td>-35.15</td><td>-34.94</td><td>-36.09</td><td>-35.10</td><td>-36.11</td><td>-35.23</td><td>-36.21</td><td>-35.97</td></tr>
<tr><td>12</td><td>-26.77</td><td>-26.51</td><td>-26.90</td><td>-26.65</td><td>-27.33</td><td>-27.11</td><td>-27.40</td><td>-26.67</td><td>-28.03</td><td>-27.26</td></tr>
<tr><td>13</td><td>-32.01</td><td>-31.14</td><td>-32.47</td><td>-31.73</td><td>-33.29</td><td>-32.96</td><td>-34.01</td><td>-33.61</td><td>-34.85</td><td>-34.17</td></tr>
<tr><td>14</td><td>-23.78</td><td>-23.31</td><td>-24.50</td><td>-23.56</td><td>-24.51</td><td>-24.46</td><td>-24.55</td><td>-23.60</td><td>-25.03</td><td>-24.67</td></tr>
<tr><td>15</td><td>-24.71</td><td>-23.89</td><td>-25.39</td><td>-24.82</td><td>-25.46</td><td>-25.37</td><td>-26.24</td><td>-25.42</td><td>-27.09</td><td>-26.94</td></tr>
<tr><td>16</td><td>-29.17</td><td>-28.37</td><td>-30.09</td><td>-30.07</td><td>-30.78</td><td>-29.83</td><td>-31.27</td><td>-30.51</td><td>-31.31</td><td>-30.56</td></tr>
</table>
</td></tr>
<tr><td>18</td><td>Get the average of each DUT for all PCBAs</td><td>Average each DUTs results.
<table>
<tr><th>DUT</th><th>=AVERAGE(B1:K1)</th></tr>
<tr><td>DUT 1</td><td>-29.10</td></tr>
<tr><td>DUT 2</td><td>-27.41</td></tr>
<tr><td>DUT 3</td><td>-28.07</td></tr>
<tr><td>DUT 4</td><td>-29.11</td></tr>
<tr><td>DUT 5</td><td>-26.68</td></tr>
<tr><td>DUT 6</td><td>-23.05</td></tr>
<tr><td>DUT 7</td><td>-29.19</td></tr>
<tr><td>DUT 8</td><td>-39.64</td></tr>
<tr><td>DUT 9</td><td>-27.71</td></tr>
<tr><td>DUT 10</td><td>-37.18</td></tr>
<tr><td>DUT 11</td><td>-35.26</td></tr>
<tr><td>DUT 12</td><td>-27.06</td></tr>
<tr><td>DUT 13</td><td>-33.02</td></tr>
<tr><td>DUT 14</td><td>-24.20</td></tr>
<tr><td>DUT 15</td><td>-25.53</td></tr>
<tr><td>DUT 16</td><td>-30.20</td></tr>
</table>
</td></tr>
</table>

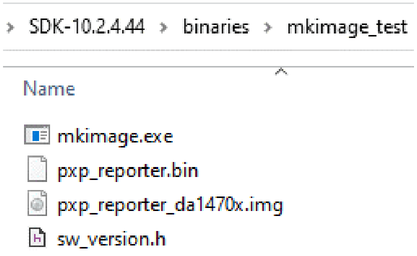
<table>
<tr><th>#</th><th>Step</th><th>Description</th></tr>
<tr><td>19</td><td>Calculate the RF path loss</td><td>To calibrate the RF result to -10 dBm we should apply the formula shown in the third column below.
<table>
<tr><th>DUT</th><th>=AVERAGE(B1:K1)
Row L</th><th>Path Loss=-10-L1</th></tr>
<tr><td>DUT 1</td><td>-29.10</td><td>19.10</td></tr>
<tr><td>DUT 2</td><td>-27.41</td><td>17.41</td></tr>
<tr><td>DUT 3</td><td>-28.07</td><td>18.07</td></tr>
<tr><td>DUT 4</td><td>-29.11</td><td>19.11</td></tr>
<tr><td>DUT 5</td><td>-26.68</td><td>16.68</td></tr>
<tr><td>DUT 6</td><td>-23.05</td><td>13.05</td></tr>
<tr><td>DUT 7</td><td>-29.19</td><td>19.19</td></tr>
<tr><td>DUT 8</td><td>-39.64</td><td>29.64</td></tr>
<tr><td>DUT 9</td><td>-27.71</td><td>17.71</td></tr>
<tr><td>DUT 10</td><td>-37.18</td><td>27.18</td></tr>
<tr><td>DUT 11</td><td>-35.26</td><td>25.26</td></tr>
<tr><td>DUT 12</td><td>-27.06</td><td>17.06</td></tr>
<tr><td>DUT 13</td><td>-33.02</td><td>23.02</td></tr>
<tr><td>DUT 14</td><td>-24.20</td><td>14.20</td></tr>
<tr><td>DUT 15</td><td>-25.53</td><td>15.53</td></tr>
<tr><td>DUT 16</td><td>-30.20</td><td>20.20</td></tr>
</table></td></tr>
<tr><td>20</td><td>Apply the calculated path losses to the SmartBond_CFG_PLT.exe PLT configuration executable.</td><td>

Path losses per DUT. Values 0.00 to 40.00dB.
<table>
<tr><td>DUT 1</td><td>19.1</td><td>DUT 5</td><td>16.68</td><td>DUT 9</td><td>17.71</td><td>DUT 13</td><td>32.02</td></tr>
<tr><td>DUT 2</td><td>17.41</td><td>DUT 6</td><td>13.05</td><td>DUT 10</td><td>27.18</td><td>DUT 14</td><td>14.2</td></tr>
<tr><td>DUT 3</td><td>18.07</td><td>DUT 7</td><td>19.19</td><td>DUT 11</td><td>25.26</td><td>DUT 15</td><td>15.53</td></tr>
<tr><td>DUT 4</td><td>19.11</td><td>DUT 8</td><td>29.64</td><td>DUT 12</td><td>17.06</td><td>DUT 16</td><td>20.20</td></tr>
</table></td></tr>
<tr><td>21</td><td>Verify</td><td>Repeat steps 10 to 13 with the 10 PCBAs. Check the GU RX RSSI results in the 10 CSV files. The results should be very close to -10 dBm.</td></tr>
</table>

Appendix G Mkimage.exe

Figure 121 shows an example of using the `mkimage.exe` utility, which creates a bootable image, appropriate to be burned to a QSPI or OSPI Flash memory.

The file `mkimage.exe` must be put in the same directory as the file to be converted. Figure 120 shows the directory and the files used in this example.

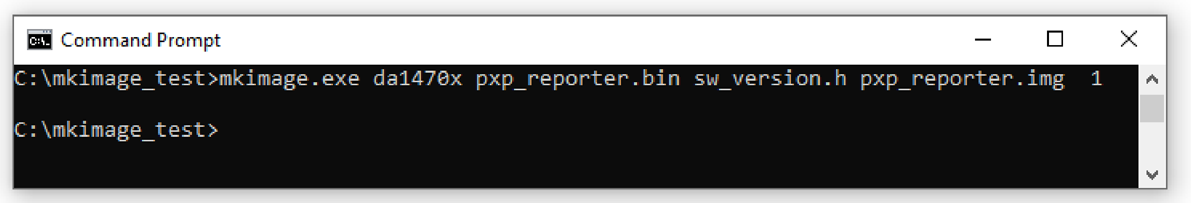


SDK-10.2.4.44 › binaries › mkimage_test

Name

- mkimage.exe
- pxp_reporter.bin
- pxp_reporter_da1470x.img
- sw_version.h

Figure 120: mkimage.exe Example Directory with Files

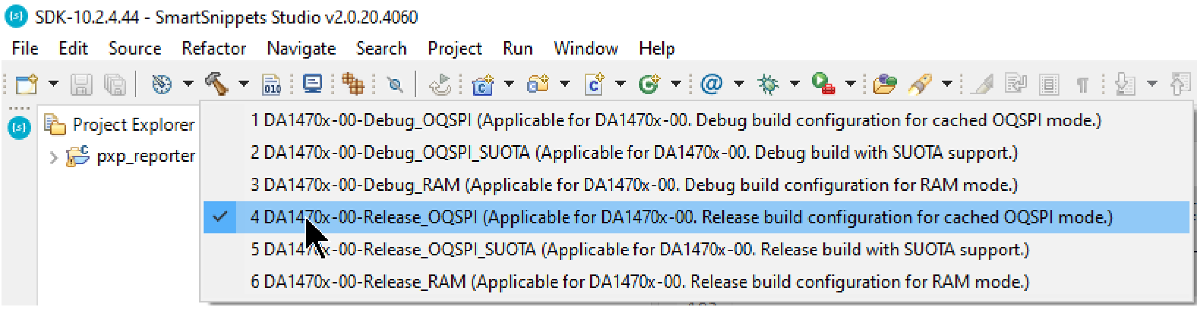


```
C:\mkimage_test>mkimage.exe da1470x pxp_reporter.bin sw_version.h pxp_reporter.img  1

C:\mkimage_test>
```

Figure 121: mkimage.exe Example Usage

In this example the `pxp_reporter` project was build using SmartSnippets^{TM} Studio. Configuration DA1470x-00-Realease_OQSPI was used as shown in Figure 122. The output binary `pxp_reporter.bin` found inside `DA1470x-00-Release_OQSPI` folder, can then be converted to a bootable image using `mkimage.exe`, as explained above.



SDK-10.2.4.44 - SmartSnippets Studio v2.0.20.4060
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
pxp_reporter

1 DA1470x-00-Debug_OQSPI (Applicable for DA1470x-00. Debug build configuration for cached OQSPI mode.)
2 DA1470x-00-Debug_OQSPI_SUOTA (Applicable for DA1470x-00. Debug build with SUOTA support.)
3 DA1470x-00-Debug_RAM (Applicable for DA1470x-00. Debug build configuration for RAM mode.)
✓ 4 DA1470x-00-Release_OQSPI (Applicable for DA1470x-00. Release build configuration for cached OQSPI mode.)
5 DA1470x-00-Release_OQSPI_SUOTA (Applicable for DA1470x-00. Release build with SUOTA support.)
6 DA1470x-00-Release_RAM (Applicable for DA1470x-00. Release build configuration for RAM mode.)

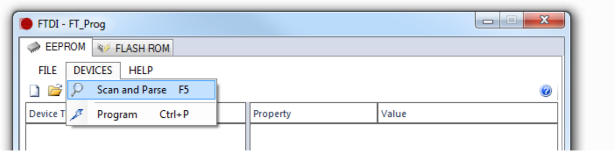
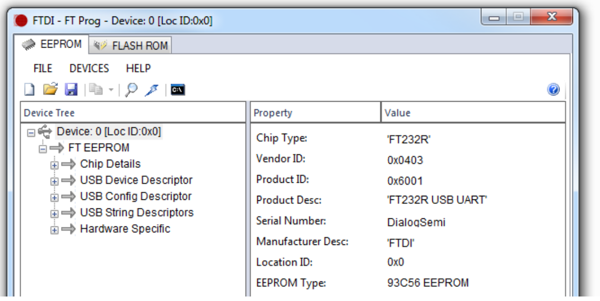
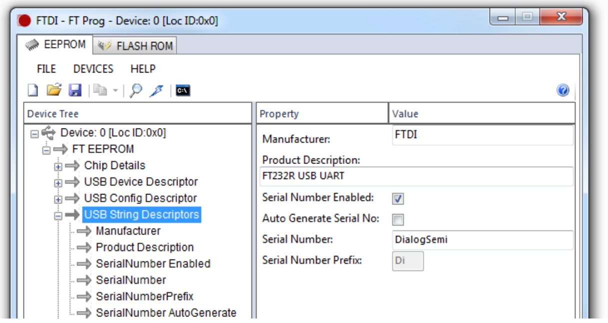
Figure 122: Build pxp_reporter Using SmartSnippets^{TM} Studio

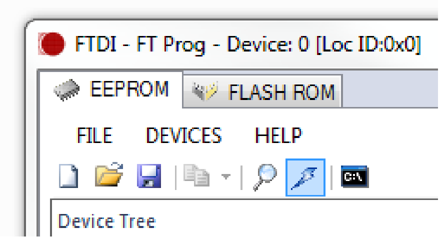
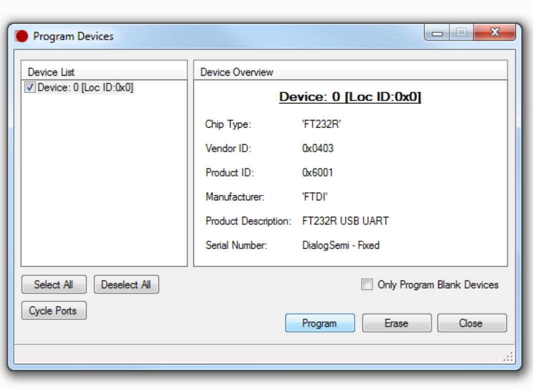
Appendix H Automatic GU COM Port Find

For the GU COM port automatic recognition to operate, a special serial number should exist in the GU FTDI IC. Usually, this serial number is programmed during PLT PCB manufacturing, but it may not exist in some older versions.

If the 'GU COM port find' operation does not work, then the steps described in should be followed.

Table 68: FTDI "DialogSemi" Serial Number

Step	Description
1	Download the FTDI `FT_Prog` tool from http://www.ftdichip.com/Support/Utilities.htm#FT_PROG.
2	Put power on the PLT board.
3	Remove any other USB FTDI connection to the PC.
4	Connect the USB cable of the GU to the PC.
5	Check the Device Manager that the GU COM port has been found.
6	Run the `FT_Prog.exe`. FT_Prog.exe
7	Select 'Devices > Scan and Parse.  FTDI - FT_Prog EEPROM FLASH ROM FILE DEVICES HELP Scan and Parse F5 Program Ctrl+P Device T Property Value
8	A single 'FT232' device should be found.  FTDI - FT Prog - Device: 0 [Loc ID:0x0] EEPROM FLASH ROM FILE DEVICES HELP Device Tree Device: 0 [Loc ID:0x0] FT EEPROM Chip Details USB Device Descriptor USB Config Descriptor USB String Descriptors Hardware Specific Property: Value Chip Type: 'FT232R' Vendor ID: 0x0403 Product ID: 0x6001 Product Desc: 'FT232R USB UART' Serial Number: DialogSemi Manufacturer Desc: 'FTDI' Location ID: 0x0 EEPROM Type: 93C56 EEPROM
9	Select 'USB String Descriptors'.
10	Uncheck the 'Auto Generate Serial No:'.
11	Edit ‘Serial Number’ to "`DialogSemi`" as shown below.  FTDI - FT Prog - Device: 0 [Loc ID:0x0] EEPROM FLASH ROM FILE DEVICES HELP Device Tree Device: 0 [Loc ID:0x0] FT EEPROM Chip Details USB Device Descriptor USB Config Descriptor USB String Descriptors Manufacturer Product Description SerialNumber Enabled SerialNumber SerialNumberPrefix SerialNumber AutoGenerate Property: Value Manufacturer: FTDI Product Description: FT232R USB UART Serial Number Enabled: ☑ Auto Generate Serial No: □ Serial Number: DialogSemi Serial Number Prefix: Di

Step	Description
12	Click the ‘Flash’ button to program the change to the FTDI IC.  FTDI - FT Prog - Device: 0 [Loc ID:0x0] EEPROM FLASH ROM FILE DEVICES HELP C:\ Device Tree
13	Click 'Program' in the new window.  Program Devices Device List Device Overview Select All Deselect All Cycle Ports Program Erase Close ☑ Device: 0 [Loc ID:0x0] Device: 0 [Loc ID:0x0] Chip Type: 'FT232R' Vendor ID: 0x0403 Product ID: 0x6001 Manufacturer: 'FTDI' Product Description: FT232R USB UART Serial Number: DialogSemi - Fixed □ Only Program Blank Devices
14	Click ‘Close’.
15	Unplug and reconnect the GU USB cable to the PC.
16	Verify the Serial Number change by running `FT_Prog.exe` again and reading the Serial Number value.

Appendix I Improving Cabling Between PLT and DUTs

The following recommendations can be used to improve the connections between PLT and DUTs:

- Keep the lengths of the cables as short as possible
- When possible, use twisted pair cables instead of separate cables for:
  - GND/VBAT
  - GND/TxD
  - GND/RxD
- Use ferrite beads for noise reduction in cables

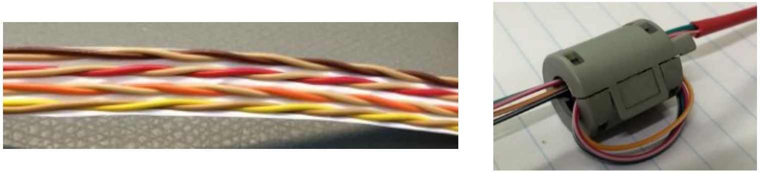


Figure 123: Example of Twisted Pair Cable with 4 Pairs and Ferrite

- Connect pull-down resistors at the end of the PLT TX signal lines. Use a 4.7 kΩ resistor at PLT DUT Connector Pin 7 (DUT TX) with the other end connected to ground. In total 16 resistors must be mounted, one for each PLT DUT connector
- Connect a pull-down resistor as close as possible to the UART RX signal connector on the DUT. The value should be approximately 4.7 kΩ. Connect the other end of the resistor to ground
- Use gold plated contacts in the connections between the PLT and the DUTs
- Use extra drivers in the UART lines
- Use series resistors of approximately 100 Ω in the UART lines, one mounted at the beginning and one at the end of the signal lines

Note: Start with the simple solutions first by testing them one-by-one for stability.

Figure 124 and Figure 125 show examples of some of the above proposals.

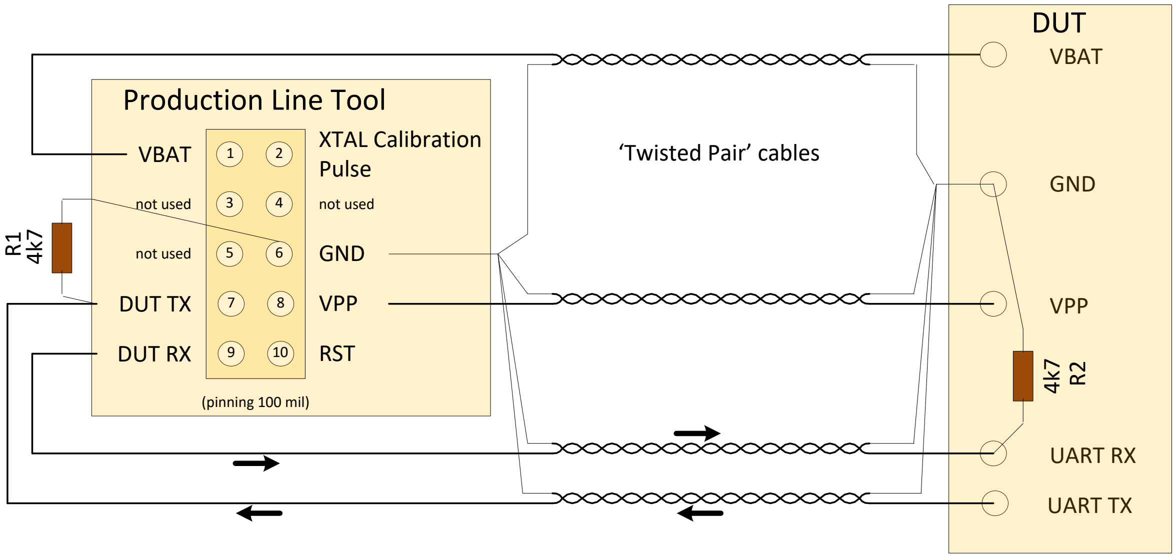


Figure 124: Location of Pull-Down Resistors

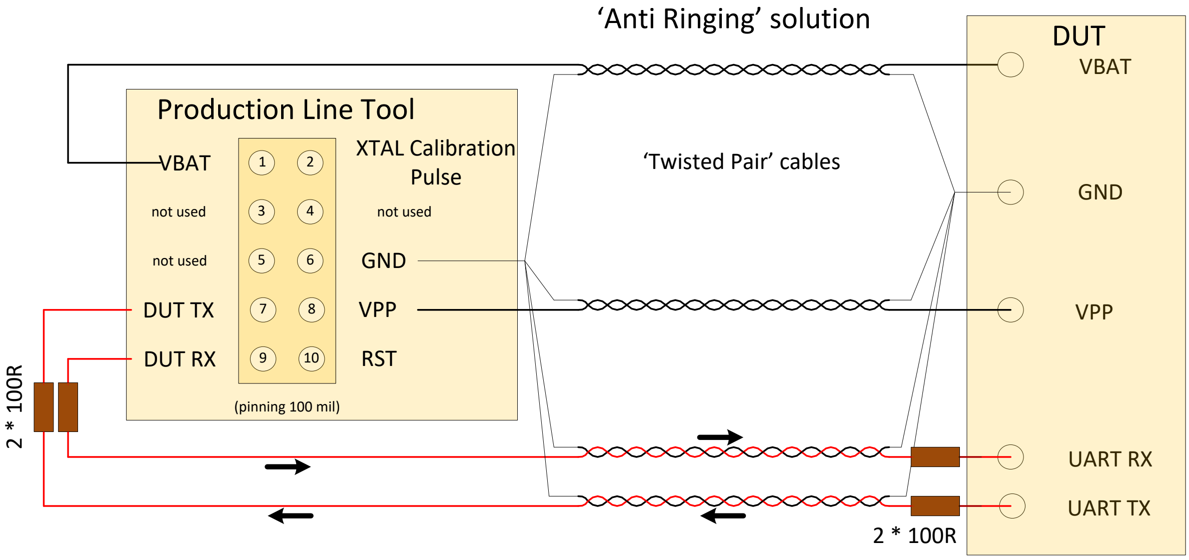


Figure 125: Anti-Ringing Solution

Appendix J FTDI Driver Removal and Installation

To re-install the latest FTDI drivers, the previous should be uninstalled.

FTDI driver removal:

5. Download CDM uninstaller from http://www.ftdichip.com/Support/Utilities.htm#CDMUninstaller.
6. Run `CDMuninstallerGUI.exe`
7. The VID/PID of the PLT FTDIs are VID=0403/PID=6011 for the DUTs and VID=0403/PID=6001 for the GU.
   Enter these VIDs and PIDs in the CDM Uninstaller and click Add for each one.
8. Then click on `Remove Devices` to uninstall the FTDI drivers.
9. Un-plug both USB cables.

More information can be found in the following link:

http://www.ftdichip.com/Support/Utilities/CDM_Uninst_GUI_Readme.html

FTDI driver installation:

1. Download the latest drivers from http://www.ftdichip.com/Drivers/VCP.htm and install them using the executable.
2. After uninstalling the drivers, plug in both USB cables. Windows will automatically assign the new drivers. Do not remove the cables during driver installation. A driver installation error may occur, and the removal-installation will have to be repeated.
3. Check in the Windows Device manager that the driver versions of the 17 PLT COM Ports->USB Serial Ports are the latest.

FTDI driver versions v2.12.24, v2.12.26, v2.12.28 and v2.12.36.4 have been tested.

Appendix K DA1470x DK Pro Motherboard Connection

Figure 126 shows the wiring to a Pro DK motherboard for the DA1470x DUTs. As described in Section 5.4 the following connections are needed to connect a DUT to the PLT:

Table 69: DA1470x Pro DK to PLT connections

Connect Function	PLT DUT connector pin number	Pro DK pin number
Ground	DUT connector pin6	Pro DK J3 pin2 (or any available ground pin)
VBAT	DUT connector pin1	Pro DK J9 pin4
UART TX	DUT connector pin7	Pro DK J3 P008
UART RX	DUT connector pin9	Pro DK J5 P201
Reset	DUT connector pin10	Not required

Note 1 The 2 indicated jumpers must be removed. Power supply will be provided from the PLT HW (VBAT line).

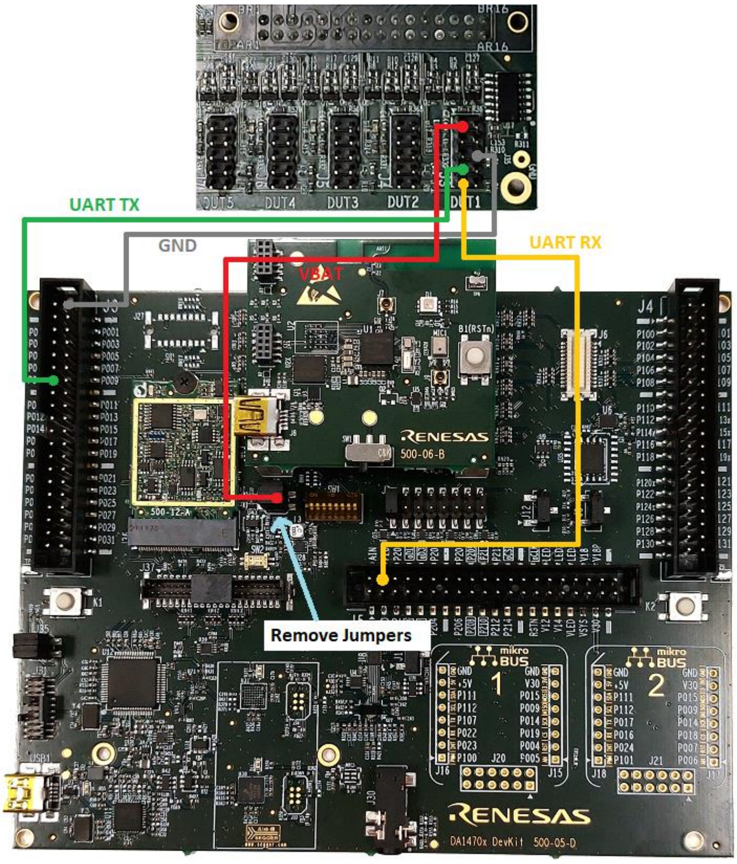


Figure 126: DA1470x Pro Motherboard DK Wiring

Appendix L Connecting DUT with Battery Supply

Wiring connections to a battery powered DUT is described in Section 5.4. Example connections can be found in Appendix K.

1. Four wires are mandatory for the connection:
   - Common Ground
   - UART Tx
   - UART Rx
   - Reset line
2. `'VBAT On with Reset'` mode is the only mode supporting battery powered DUTs since POR cannot be performed. For the PLT to perform a reset on the DUTs, the VBAT line of each DUT connector must be connected to the reset line of the DUT.
3. Current measurement is not supported since there is no way to measure the current of the DUTs.

To have the least possible wiring connections, UART Rx line can be used as input GPIO for the pulse used during the XTAL Trim procedure.

Appendix M User Interfaces Shortcut Keys

Table 70: User Interface Shortcut Keys

Application	Shortcut	Description
CSV File CFG PLT Application (Section 7.2)	`Ctrl + S`	This shortcut is equivalent to clicking the save button at the bottom of the screen.
GUI PLT Application (Section 7.3)	`Spacebar`	It is used to select the ‘Start’ button to start testing.
	`F`	It is equivalent to clicking the ‘Finished’ button with the left mouse-click.

Appendix N DA1470x Supported QSPI Flash Memories

Table 71 describes the supported QSPI flash memories for DA1470x devices. To support a QSPI or OSPI flash memory that is not in the list, follow the instructions in the `README OQSPI.md` file located under `SDK_xx.x.x.xxx\sdk\bsp\memory\` folder to manually add it inside the `uartboot_da1470x.bin` firmware.

Table 71: DA1470x Supported QSPI/OQSPI Flash Memories

Memory vendor	Flash Type	Product number
Adesto	QSPI	AT25SL128
Macronix	QSPI	MX25U6432
Windbond	QSPI	W25U6432
Windbond	QSPI	W25Q256JW

Appendix O Bluetooth® LE Tester Measurement Results

When using an MT8852B as an external Bluetooth® LE tester instrument for the RF Tests (Section 7.2.6.5), PLT will instruct the MT8852B to perform specific tests and then wait for its reply. MT8852B replies with a string, containing the command code of the test performed, followed by the results of the test or an Error Response string (Appendix O.4).

Table 72 shows the result command codes.

The PDF document (located by clicking) the following link, describes the format of the result string for each command code, under section 15 – 7.

https://dl.cdn-anritsu.com/en-au/test-measurement/files/Manuals/Programming-Manual/MT8852B/MT8852B-Bluetooth%20tester-Programming%20Manual%20Rev%20X.pdf

Table 72: MT8852B Supported Command Codes

Code	Test
LEOP0	TX power
LEICD0	Carrier frequency and Drift
LEMI	Modulation index
ERRLST	Error response

As an example, the test results of DUT1 in the example of CSV Log File Contents (Appendix P) will be used.

O.1 Tx Power

The result of DUT1 for the TX-Power test is:

```
TRUE;-14.25;-14.25;-14.25;0.10;0;2;PASS
```

Table 73: MT8852B – Bluetooth® LE TX Output Power Test Results

Description	Format	Example
Results valid	TRUE | FALSE	TRUE
Packet average power in dBm	floating point	-14.25
Test avg max in dBm	floating point	-14.25
Test avg min in dBm	floating point	-14.25
Test peak to average power in dBm	floating point	0.10
Number of failed packets	integer	0
Number of tested packets	integer	2
Pass/fail result	PASS | FAIL	PASS

O.2 Frequency Offset

The result of DUT1 for the Frequency Offset test is:

```
TRUE;-2.500e+003;2.800e+003;-7.800e+003;-7646;-9.0e+003;-9.0e+003;0;2;PASS;-7646
```

Table 74: MT8852B – Bluetooth® LE Carrier Frequency Offset and Drift Test Results

Description	Format	Example
Drift rate valid	TRUE | FALSE	TRUE
Average Fn	Integer	-2.500e+003
Maximum Positive Fn	Integer	2.800e+003
Minimum Negative Fn	integer	-7.800e+003
Drift rate	integer	-7646
Average drift	integer	-9.0e+003
Maximum drift	integer	-9.0e+003
Packets Failed	integer	0
Packets Tested	integer	2
Pass/fail result	PASS | FAIL	PASS
Initial drift rate	integer	-7646

O.3 Modulation Index

The result of DUT1 for the Modulation Index test is:

```
TRUE;282100.00;249100.00;200700.00;248700.00;1.00;0;576;1;1;FAIL;100.00%
```

The tester responded with FAIL because two tests with different patterns were needed. The overall result of the Modulation Index test will be concluded in a second step, after the second payload is tested.

Table 75: MT8852B – Bluetooth® LE Modulation Characteristics Test Results

Description	Format	Example
Results valid	TRUE | FALSE	TRUE
Delta f1 max in Hz	floating point	282100.00
Delta f1 average in Hz	floating point	249100.00
Delta f2 max in Hz (Delta f1 max lowest for BLR8)	floating point	200700.00
Delta f2 average in Hz (Omitted for BLR8)	floating point	248700.00
Delta f2 avg / delta f1 avg (Omitted for BLR8)	floating point	1.00
Delta f2 max Failed limit (Delta f1 max Failed limit for BLR8)	integer	0
Delta f2 max count(Delta f1 max count for BLR8)	integer	576
Packets failed	integer	1
Packets tested	integer	1
Pass/fail result	PASS | FAIL	FAIL
Delta f2 max % pass rate (Delta f1max % pass rate for BLR8)	floating point	100.00%

O.4 Error Response

If the Bluetooth® LE Tester fails to perform the tests it will respond with an error. Table 76 describes the parts of the error message. Click the following link to find more details in Section 4.3 of the relative PDF document:

https://dl.cdn-anritsu.com/en-au/test-measurement/files/Manuals/Programming-Manual/MT8852B/MT8852B-Bluetooth%20tester-Programming%20Manual%20Rev%20X.pdf

The format of the response message is

```
ABCCDDEFGHIIJKK!LLLLLLL!MMMMMMM!NNNNNNN!OOOOOOO!
```

A common error response, which is not an actual error, is the message below:

```
000000000000000!NO ERRORS!NO ERRORS!!!
```

This is due to the output received signal may be less than -50 dB.

Table 76: MT8852B – Error List

Alias	Error	Status	Description
A	CONNECTION ALREADY EXISTS	0	No previous connection
		1	Connection already exists
B	EUT TEST MODE STATE	0	EUT Test Mode enabled
		1	EUT Test Mode not enabled
CC	EUT HCI ERROR	00	OK
		XX	2-digit hex error code (EUT controlled via RS232)
DD	INTERNAL HCI ERROR	00	OK
		XX	2-digit hexadecimal error code
E	INTERNAL SYNC ERROR	0	OK
		1	Internal HCI synchronization error
F	EUT SYNC ERROR	0	OK
		1	EUT HCI synchronization error (control via RS232)
G	EUT HARDWARE ERROR	0	OK
		1	EUT Reported HCI Hardware error message
H	REQUEST FAILED	0	OK
		1	Request failed (system busy)
II	DSP STATUS **Note:** Setting of the DSP status code will not set the DDE bit of the event register	00	OK
		01	Searching channel
		02	Searching sync word
		03	Incorrect packet length
		04	No payload
		05	Auto ranging
		06	Incorrect packet
		07	Incorrect packet type
		08	Over range
		09	Under range

Alias	Error	Status	Description
		10	Invalid payload
		11	Error finding start of packet using power profile
		12	Error locating P0/GFSK sync word
		13	Location of P0/GFSK sync word exceeds allowed limits
		14	Error locating EDR sync word
		15	Location of EDR sync word exceeds allowed limits
		16	Error decoding the packet type field
		17	Modulation mode of PI/4-DQPSK or 8DPSK not specified
		18	pi/4-DQPSK modulation does not match with detected packet type
		19	8DPSK modulation does not match with packet type
		20	Invalid packet type decoded
		21	Unknown packet type decoded
		22	Expected and measured packet lengths do not match
		23	Insufficient blocks in packet for measurement
J	EUT BT ADDRESS	0	OK
		1	EUT Bluetooth Address set (in Manual mode)
KK	HCI COMM STATUS	00	OK
		01	Unknown HCI command
		02	No connection
		03	Hardware failure
		04	Paging timeout
		05	Connection timeout
		06	Unsupported feature parameter
		07	Connection ended by user
		08	Low resource connection ended
		09	Power Off connection ended
		10	Local host connection ended
		11	Unsupported remote feature
		12	Role change not allowed
		13	LMP response timeout
		14	IQ modem DAC saturation
LLLLLLL			Internal core error text (variable length)
MMMMMMM			EUT core error text (variable length)
NNNNNNN			Last GPIB command that caused a Command error (variable length)
OOOOOOO			Last GPIB command that caused an Execution error (variable length)

Appendix P CSV Log File Contents

Table 77 describes the CSV File columns (Section 7.3.5) generated during PLT testing. In general, not all CSV file columns shown in Table 77 will be printed, but only those that relate to the enabled tests and memory operations user has selected. An example is given in Appendix P.1.

Table 77: CSV File Contents

Header	Value	Description
Start time	hh:mm:ss	Shows the time the test procedure has started
End time	hh:mm:ss	Shows the time the test procedure has ended
Tester ID	ID	The tester ID name.
DUT	1-16	The PLT connector for the DUT. Values 1-16.
BD address	XX:XX:XX:XX:XX:XX	The BD address assigned for this device.
Overall status	PASS\FAIL	The overall final test result for this DUT.
COM Port	XX	Windows assigned COM Port
Temperature test	PASS\FAIL	Shows the temperature measured during temperature test. The first column shows the result of the test. The second column shows the temperature measured.
Temperature	xx.xx	
Programmer FW download 1	PASS\FAIL	`uartboot_da1470x.bin` FW download. The first column shows the result of the programmer firmware download procedure. The second column shows the path to the firmware.
Programmer FW path 1	C:\folder\to\bin	
Programmer FW version get 1	PASS\FAIL	The firmware version of the programmer firmware (`uartboot_da1470x.bin`). The first column shows the result of the firmware version acquisition. The second column shows the actual firmware version.
Programmer FW version 1	x.x.x.x	
GPIO Watchdog mem 1 [‘Test Name’]	PASS\FAIL	The GPIO watchdog operation for the programmer firmware (`uartboot_da1470x.bin`).
RAM FW download	PASS\FAIL	Production test FW download through memory programming FW. The first column shows the result of the production test firmware download procedure. The second column shows the path to the firmware.
RAM FW path	C:\path\to\bin	

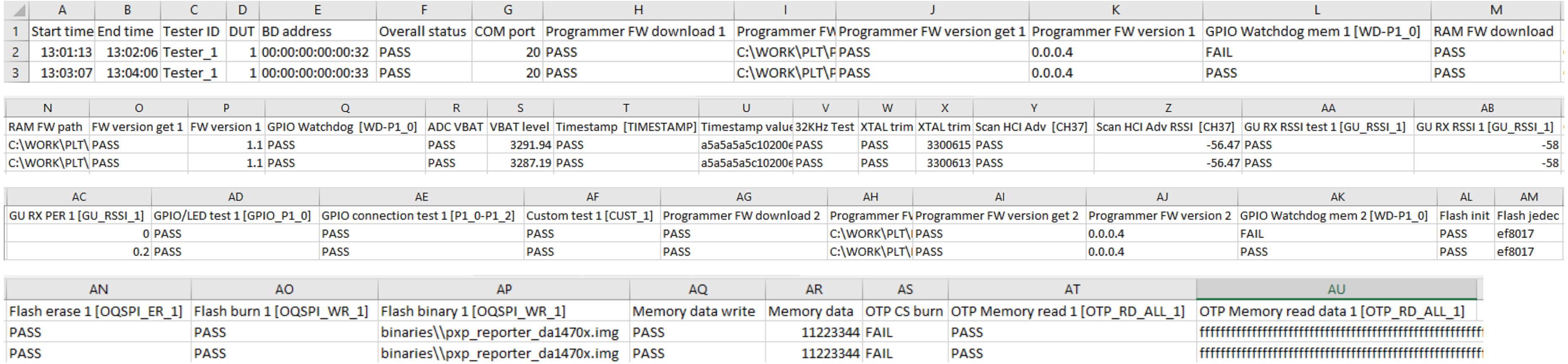
Header	Value	Description
FW version get 1	PASS\FAIL	Production test FW version. The first column shows the result of the test. The second column shows the production test FW version read back from each device.
FW version 1	e.g., “v_5.0.4_PLT_v4.3”	
GPIO Watchdog [‘Test Name’]	PASS\FAIL	GPIO watchdog toggling test for the production test firmware (`st_fw_da1470x.bin`)
ADC VBAT	PASS\FAIL	The first column contains the result whether the VBAT was successfully measured. The second column has the level of the VBAT as measured by the internal DUT ADC.
VBAT level	VBAT level	
Timestamp [‘Test Name’]	PASS\FAIL	The first column contains the result whether the OTP timestamp read succeeded. The second column has the actual timestamp of the DUT IC.
Timestamp value [‘Test Name’]	Timestamp	
32 Khz Test	PASS\FAIL	Pass or fail result of the external 32 Khz test.
XTAL trim test	PASS\FAIL	Automated XTAL Trim value calculation. The first column shows the result of the test. The second column shows the calculated value in decimal.
XTAL trim	e.g., “2775311”	
Scan HCI Adv [CH37-9\All]	PASS\FAIL	Scan test using Advertising through HCI. The first column shows the result of the test. The second column shows the calculated value in decimal.
Scan HCI Adv RSSI [CH37-9\All]	The RSSI value measured	
BLE TX power test ‘X’ [Test Name]	PASS\FAIL	Tx Power tests using external Bluetooth® LE Tester
BLE TX power ‘X’ [Test Name]	TRUE;-13.16;-13.16;-13.16;0.08;0;2;PASS	
BLE TX offset test ‘X’ [‘Test Name’]	PASS\FAIL	Tx Frequency offset tests using external Bluetooth® LE Tester
BLE TX offset ‘X’ [Test Name]	TRUE;-1.100e+003;1.400e+003;-4.000e+003;2902;-3.0e+003;-3.0e+003;0;2;PASS;2503	
BLE TX modulation test ‘X’ [‘Test Name’]	PASS\FAIL	Tx Modulation Index tests using external Bluetooth® LE Tester. The first column shows the result of the test. The second column shows the calculated value.
BLE TX modulation ‘X’ [Test Name]	TRUE;260600.00;253300.00;216800.00;248000.00;0.98;0;576;1;1;FAIL;100.00%	
RX RSSI test ‘X’ [‘Test Name’]	PASS\FAIL	

Header	Value	Description
BLE RX RSSI [‘Test Name’]	The RSSI value measured for this device.	Rx sensitivity tests using external Bluetooth® LE Tester. The first column shows the result of the test. The second column shows the RSSI value measured for this device. The third column shows the Packet Error Rate measured for this device.
BLE RX PER [‘Test Name’]	The Packet Error Rate measured for this device.	
GU RX RSSI test ‘X’ [‘Test Name’]	PASS\FAIL	Rx sensitivity tests using Golden Unit as Tester. The first column shows the result of the test. The second column shows the RSSI value measured for this device. The third column shows the Packet Error Rate measured for this device.
GU RX RSSI ‘X’ [‘Test Name’]	The RSSI value measured for this device.	
GU RX PER ‘X’ [‘Test Name’]	The PER measured for this device.	
GPIO/LED test ‘X’ [‘Test Name’]	PASS\FAIL	GPIO\LED tests
GPIO connection test ‘X’ [Test Name]	PASS\FAIL	GPIO connection tests
Sensor test ‘X’ [‘Test Name’]	PASS\FAIL	Sensor tests.
Custom test ‘X’ [‘Test Name’]	PASS\FAIL	Custom tests
Peripheral test ‘X’ [‘Test Name’]	PASS\FAIL	Current measurement tests for peripherals. The first column shows the result of the test. The second column shows the calculated value, and the high and low limits used for this test.
Peripheral test current	RES=[xxxxA]. LL=[xxxxA]. HL=[xxxxA].	
Extended\Deep sleep current test	PASS\FAIL	Current measurement test for DUT sleep. The first column shows the result of the test. The second column shows the calculated value, and the high and low limits used for this test.
Extended\Deep sleep current	RES=[xxxxA]. LL=[xxxxA]. HL=[xxxxA].	
Programmer FW download 2	PASS\FAIL	Memory programming FW download. The first column shows the result of the production test firmware download procedure. The second column shows the path to the firmware.
Programmer FW path 2	C:\path\to\bin	
Programmer FW version get 2	PASS\FAIL	Memory programming FW version. The first column shows the result of the test. The second column shows the memory programming FW version read back from each device.
Programmer FW version 2	e.g., “v_5.0.4_PLT_v4.3”	
GPIO Watchdog mem 2 [‘Test Name’]	PASS\FAIL	GPIO watchdog toggling test for the production test firmware (st_fw_da1470x.bin)
Flash init [“OQSPI”]	PASS\FAIL	

Header	Value	Description
Flash jedec [“OQSPI”]	xxxxxx	Initialize OQSPI Flash memory. The first column shows the result of the test. The second column shows the JEDEC ID read back from the device.
Flash erase ‘X’ [‘Test Name’]	PASS\FAIL	Erase the OQSPI or QSPI Flash memory test.
Flash burn ‘X’ [‘Test Name’]	PASS\FAIL	OQSPI or QSPI image write test. The first column shows the result of the test. The second column shows the path to the firmware burnt.
Flash binary ‘X’ [‘Test Name’]	C:\path\to\bin	
Memory data write	PASS\FAIL	Write any available custom memory test. The first column shows the result of the test. The second columns shows whether the data are given from a CSV file, or the contents written.
Memory data	“Data from CSV file” or “xx…”	
OTP burn	PASS\FAIL	OTP binary write test. The first column shows the result of the test. The second column shows the path to the firmware burnt.
OTP image	C:\path\to\bin	
OTP CS burn	PASS\FAIL	OTP Configuration Script area burn.
‘SPI\OTP\EEPROM\QSPI’ Memory read ‘X’ [‘Test Name’]	PASS\FAIL	Read any part of any available memory. The first column shows the result of the test. The second column shows the contents read back.
Memory read data	xx…	
Scan	PASS\FAIL	Scan test with the DUTs booting and advertising.

P.1 CSV Log File Example

Figure 127 shows a CSV log example for two DUTs. The contents are explained in Table 78.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Start time	End time	Tester ID	DUT	BD address	Overall status	COM port	Programmer FW download 1	Programmer FW	Programmer FW version get 1	Programmer FW version 1	GPIO Watchdog mem 1 [WD-P1_0]	RAM FW download
2	13:01:13	13:02:06	Tester_1	1	00:00:00:00:00:32	PASS	20	PASS	C:\WORK\PLT\F	PASS	0.0.0.4	FAIL	PASS
3	13:03:07	13:04:00	Tester_1	1	00:00:00:00:00:33	PASS	20	PASS	C:\WORK\PLT\F	PASS	0.0.0.4	PASS	PASS

N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
RAM FW path	FW version get 1	FW version 1	GPIO Watchdog [WD-P1_0]	ADC VBAT	VBAT level	Timestamp [TIMESTAMP]	Timestamp value	32KHz Test	XTAL trim	XTAL trim	Scan HCI Adv [CH37]	Scan HCI Adv RSSI [CH37]	GU RX RSSI test 1 [GU_RSSI_1]	GU RX RSSI 1 [GU_RSSI_1]
C:\WORK\PLT\	PASS	1.1	PASS	PASS	3291.94	PASS	a5a5a5a5c10200e	PASS	PASS	3300615	PASS	-56.47	PASS	-58
C:\WORK\PLT\	PASS	1.1	PASS	PASS	3287.19	PASS	a5a5a5a5c10200e	PASS	PASS	3300613	PASS	-56.47	PASS	-58

AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
GU RX PER 1 [GU_RSSI_1]	GPIO/LED test 1 [GPIO_P1_0]	GPIO connection test 1 [P1_0-P1_2]	Custom test 1 [CUST_1]	Programmer FW download 2	Programmer FV	Programmer FW version get 2	Programmer FW version 2	GPIO Watchdog mem 2 [WD-P1_0]	Flash init	Flash jedec
0	PASS	PASS	PASS	PASS	C:\WORK\PLT\	PASS	0.0.0.4	FAIL	PASS	ef8017
0.2	PASS	PASS	PASS	PASS	C:\WORK\PLT\	PASS	0.0.0.4	PASS	PASS	ef8017

AN	AO	AP	AQ	AR	AS	AT	AU
Flash erase 1 [OQSPI_ER_1]	Flash burn 1 [OQSPI_WR_1]	Flash binary 1 [OQSPI_WR_1]	Memory data write	Memory data	OTP CS burn	OTP Memory read 1 [OTP_RD_ALL_1]	OTP Memory read data 1 [OTP_RD_ALL_1]
PASS	PASS	binaries\\pxp_reporter_da1470x.img	PASS	11223344	FAIL	PASS	ffffffffffffffffffffffffffffffffffffffffffffffffffffff
PASS	PASS	binaries\\pxp_reporter_da1470x.img	PASS	11223344	FAIL	PASS	ffffffffffffffffffffffffffffffffffffffffffffffffffffff

Figure 127: Example CSV Log File

Table 78: Example CSV Log File Content Analysis

#	Header	Value	Description
A	Start time	4:53:30 PM	The time the tests started.
B	End time	4:55:03 PM	The time the tests finished.
C	Tester ID	Tester_1	Tester ID Entered in config.
D	DUT	1, 2	The PLT position each device is connected.
E	BD address	00:00:00:00:00:32, 00:00:00:00:00:33	The BD address assigned to each device.
F	Overall status	PASS, PASS	The overall result for each device.
G	COM port	20, 20	The Windows assigned COM port to each device.
H	Programmer FW download 1	PASS, PASS	Booter firmware (`uartboot_da1470x.bin`) loaded successfully.

#	Header	Value	Description
I	Programmer FW path 1	C:\WORK\PLT\SmartBond_PLT_v_5.0\executables\binaries\uartboot_da1470x.bin	The path to the booter firmware, `uartboot_da1470x.bin`.
J	Programmer FW version get 1	PASS	All devices responded to firmware version request. Version is `0.0.0.4`.
K	Programmer FW version 1	0.0.0.4	
L	GPIO Watchdog mem 1 [WD-P1_0]	PASS	GPIO toggling for watchdog has started. In addition, header has the name assigned to the test.
M	RAM FW download	PASS	Production test firmware (`st_fw_da1470x.bin`) downloaded to RAM.
N	RAM FW Path	C:\SmartBond_Cobalt_v_5.0_RC2\binaries\st_fw_da1470x.bin	
O	FW version get 1	PASS	All devices responded to firmware version request. All of them have version 1.1.
P	FW version 1	1.1	
Q	GPIO Watchdog [WD-P1_0]	PASS	
R	ADC VBAT	PASS	ADC VBAT test success.
S	VBAT level	3291.94, 3287.19	ADC VBAT test result.
T	Timestamp [TIMESTAMP]	PASS	Timestamp read test success.
U	Timestamp value [TIMESTAMP]	a5a5a5a5c10200e940881e1902b …	Timestamp read value.
V	XTAL Trim test	PASS	XTAL Trim test finished successfully. The calculated value (in decimal) is shown for the first device
W	XTAL Trim	3300615, 3300613	
X	Scan HCI Adv [CH37]	PASS	Scan test using advertising through HCI commands test finished successfully. Channel 37 is selected, which is also shown on the header. RSSI values, for each device are also shown.
Y	Scan HCI Adv RSSI [CH37]	-37.51, -32.77	
AA	GU RX RSSI test 1 [GU_RSSI_1]	PASS	Reception test 1 using the Golden Unit finished successfully.

#	Header	Value	Description
AB	GU RX RSSI 1 [GU_RSSI_1]	-58, -58	In addition, the header has the name assigned to the test and the RSSI and packet error rate for each device are shown.
AC	GU RX PER 1 [GU_RSSI_1]	0, 0.2	
AD	GPIO/LED test 1 [GPIO_P1_0]	PASS, PASS	GPIO\LED toggling test finished successfully.
AE	GPIO connection test 1 [P1_0-P1_2]	PASS, PASS	GPIO connection test for GPIOs P1.0 and P1.2. Test finished successfully.
AF	Custom test 1 [CUST_1]	PASS, PASS	Custom test finished successfully.
AG	Programmer FW download 2	PASS, PASS	Booter firmware (uartboot_da1470x.bin) loaded successfully.
AH	Programmer FW path 2	C:\WORK\PLT\SmartBond_PLT_v_5.0\executables\binaries\uartboot_da1470x.bin	The path to the booter firmware, uartboot_da1470x.bin.
AI	Programmer FW version get 2	PASS, PASS	All devices responded to firmware version request. Version is 0.0.0.4.
AJ	Programmer FW version 2	0.0.0.4	
AK	GPIO Watchdog mem 1 [WD-P1_0]	PASS, PASS	GPIO toggling for watchdog has started. In addition, header has the name assigned to the test.
AL	Flash init	PASS, PASS	Initialization of Flash memory succeeded. The JEDEC ID for both Flash memories is 0xEF8017.
AM	Flash jedec	ef8017, ef8017	
AN	Flash erase 1 [OQSPI_ER_1]	PASS, PASS	The flash was erased successfully.
AO	Flash burn 1 [OQSPI_WR_1]	PASS, PASS	Flash was programmed successfully. The path to the binary programmed is binaries\\pxp_reporter_da1470x.img.
AP	Flash binary 1 [OQSPI_WR_1]	binaries\\pxp_reporter_da1470x.img	
AQ	Memory data write	PASS, PASS	

#	Header	Value	Description
AR	Memory data	11223344, 11223344	Custom memory data were programmed successfully. Data to program was 0x11223344.
AS	OTP CS burn	PASS, PASS	The OTP CS was successfully programmed.
AT	OTP Memory read 1 [OTP_RD_ALL_1]	PASS, PASS	OTP memory read was successful. Data read from DUTs are 0xFFFF…
AU	OTP Memory read data 1 [OTP_RD_ALL_1]	ffffffffffffffffffffffff,.. ffffffffffffffffffffffff…	

Appendix Q DUT Status Codes

Table 79 contains all the possible status codes a DUT can have, followed by a brief description. The table categorizes the status based on the various states of the DUT during testing and programming.

Table 79: DUT Status Codes

No.	Status	Description
	Generic	
	DUT_NOT_ACTIVE	Device is not active.
1.	DUT_INTERNAL_SYSTEM_ERROR	Internal system error.
2.	DUT_COM_PORT_IDENTIFY_STARTED	COM port identification started.
3.	DUT_COM_PORT_IDENTIFY_OK	COM port identified successfully.
4.	DUT_COM_PORT_IDENTIFY_FAILED	COM port identification failed.
	Temperature measurement	
5.	DUT_TEMPERATURE_MEASUREMENT_INIT	Temperature measurement initialized.
6.	DUT_TEMPERATURE_MEASUREMENT_OK	Temperature measurement finished successfully.
7.	DUT_TEMPERATURE_MEASUREMENT_ERROR	Temperature measurement error.
	Firmware download	
8.	DUT_UDLL_FW_DOWNLOAD_INIT	UDLL firmware download initialized.
9.	DUT_UDLL_FW_DOWNLOAD_STARTED	UDLL firmware download started successfully.
10.	DUT_UDLL_FW_DOWNLOAD_RETRY	UDLL firmware download retry.
11.	DUT_UDLL_FW_DOWNLOAD_OK	UDLL firmware downloaded successfully.
12.	DUT_UDLL_FW_DOWNLOAD_FAILED	UDLL firmware download failed.
	Memory programming - Firmware version	
13.	DUT_UDLL_FW_VER_GET_INIT	UDLL device firmware version acquisition initialized
14.	DUT_UDLL_FW_VER_GET_STARTED	UDLL device firmware version acquisition started.
15.	DUT_UDLL_FW_VER_GET_OK	UDLL device firmware version acquisition completed successfully.
16.	DUT_UDLL_FW_VER_GET_FAILED	UDLL device firmware version acquisition failed.
	Memory programming – GPIO watchdog	
17.	DUT_UDLL_GPIO_WD_INIT	UDLL GPIO watchdog operation initialized.
18.	DUT_UDLL_GPIO_WD_STARTED	UDLL GPIO watchdog operation started.
19.	DUT_UDLL_GPIO_WD_OK	UDLL GPIO watchdog operation ended successfully.
20.	DUT_UDLL_GPIO_WD_FAILED	UDLL GPIO watchdog operation failed.
	Memory programming – RAM firmware download	
21.	DUT_UDLL_RAM_FW_DOWNLOAD_INIT	UDLL firmware download to RAM initialized
22.	DUT_UDLL_RAM_FW_DOWNLOAD_STARTED	UDLL firmware download to RAM started.

No.	Status	Description
23.	DUT_UDLL_RAM_FW_DOWNLOAD_OK	UDLL firmware download to RAM completed successfully.
24.	DUT_UDLL_RAM_FW_DOWNLOAD_FAILED	UDLL firmware download to RAM failed.
	Production test - Generic errors	
25.	DUT_PDLL_NO_ERROR	PDLL returned success.
26.	DUT_PDLL_PARAMS_ERROR	PDLL Device parameters contain errors.
27.	DUT_PDLL_RX_TIMEOUT	Device did not reply on a PDLL message request.
28.	DUT_PDLL_TX_TIMEOUT	Sending a message to the device failed due to Tx timeout.
29.	DUT_PDLL_UNEXPECTED_EVENT	Received an unexpected message from the device.
30.	DUT_PDLL_CANNOT_ALLOCATE_MEMORY	PDLL cannot allocate memory.
31.	DUT_PDLL_INTERNAL_ERROR	PDLL internal system error.
32.	DUT_PDLL_THREAD_CREATION_ERROR	PDLL thread creation error.
33.	DUT_PDLL_INVALID_DBG_PARAMS	PDLL debug library (dbg_dll.dll) access error.
34.	DUT_PDLL_DBG_DLL_ERROR	PDLL invalid debug library (dbg_dll.dll) parameters.
	Production test - COM port	
35.	DUT_PDLL_COM_PORT_INIT	PDLL device COM port open initialized.
36.	DUT_PDLL_COM_PORT_START	PDLL device COM port open started.
37.	DUT_PDLL_COM_PORT_OK	PDLL device COM port opened successfully.
38.	DUT_PDLL_COM_PORT_FAILED	PDLL device COM port failed.
	Production test - Firmware version	
39.	DUT_PDLL_FW_VERSION_GET_START	PDLL device Firmware version acquisition started.
40.	DUT_PDLL_FW_VERSION_GET_OK	PDLL device Firmware version acquisition completed successfully.
41.	DUT_PDLL_FW_VERSION_GET_FAILED	PDLL device Firmware version acquisition failed.
	Production test – GPIO watchdog	
42.	DUT_PDLL_GPIO_WD_INIT	GPIO watchdog operation initialized.
43.	DUT_PDLL_GPIO_WD_START	GPIO watchdog operation started.
44.	DUT_PDLL_GPIO_WD_OK	GPIO watchdog operation ended successfully.
45.	DUT_PDLL_GPIO_WD_FAILED	GPIO watchdog operation failed.
	Production test – OTP timestamp read	
46.	DUT_PDLL_TIMESTAMP_RD_INIT	Initialize timestamp read from the OTP.
47.	DUT_PDLL_TIMESTAMP_RD_START	OTP timestamp read operation started.
48.	DUT_PDLL_TIMESTAMP_RD_OK	OTP timestamp read operation ended successfully.
49.	DUT_PDLL_TIMESTAMP_RD_FAILED	OTP timestamp read operation failed.
	Production test – VBAT level read	
50.	DUT_PDLL_ADC_VBAT_INIT	Initialize VBAT level read using internal ADC.

No.	Status	Description
51.	DUT_PDLL_ADC_VBAT_START	VBAT level read operation started.
52.	DUT_PDLL_ADC_VBAT_OK	VBAT level read operation ended successfully.
53.	DUT_PDLL_ADC_VBAT_FAILED	VBAT level read operation failed.
	Production test – TX power level set	
54.	DUT_PDLL_SET_TX_PWR_INIT	Initialize TX power set operation.
55.	DUT_PDLL_SET_TX_PWR_START	TX power set operation started.
56.	DUT_PDLL_SET_TX_PWR_OK	TX power set operation ended successfully.
57.	DUT_PDLL_SET_TX_PWR_FAILED	TX power set operation failed.
	Production test - External 32 Khz crystal	
58.	DUT_PDLL_EXT32KHz_TEST_INIT	External 32kHz test operation initialized.
59.	DUT_PDLL_EXT32KHz_TEST_START	External 32kHz test operation started.
60.	DUT_PDLL_EXT32KHz_TEST_OK	External 32kHz test operation ended successfully.
61.	DUT_PDLL_EXT32KHz_TEST_FAILED	External 32kHz test operation failed.
	Production test - XTAL trim	
62.	DUT_PDLL_XTAL_TRIM_INIT	XTAL trim operation initialized.
63.	DUT_PDLL_XTAL_TRIM_START	XTAL trim operation started.
64.	DUT_PDLL_XTAL_TRIM_OK	XTAL trim operation ended successfully.
65.	DUT_PDLL_XTAL_TRIM_OUT_OF_RANGE	XTAL trim failed. Input frequency is out of range.
66.	DUT_PDLL_XTAL_TRIM_FREQ_CAL_NOT_CONNECTED	XTAL trim could not be performed. Could not detect external input frequency.
67.	DUT_PDLL_XTAL_TRIM_WRITE_FAILED	XTAL trim failed. Could not write the calculated value to the OTP CS.
68.	DUT_PDLL_XTAL_TRIM_FAILED	XTAL trim failed.
	Production test – Read value written in OTP	
69.	DUT_PDLL_XTAL_TRIM_READ_INIT	XTAL trim register read operation initialized.
70.	DUT_PDLL_XTAL_TRIM_READ_START	XTAL trim register read operation started.
71.	DUT_PDLL_XTAL_TRIM_READ_OK	XTAL trim register operation ended successfully.
72.	DUT_PDLL_XTAL_TRIM_READ_FAILED	XTAL trim register operation ended successfully.
	Production test - UART resync	
73.	DUT_PDLL_UART_RESYNC_INIT	UART resync process initialized.
74.	DUT_PDLL_UART_RESYNC_START	UART resync process started.
75.	DUT_PDLL_UART_RESYNC_OK	UART resync process completed successfully.
76.	DUT_PDLL_UART_RESYNC_FAILED	UART resync process failed.
	Production test - Scan with HCI Bluetooth® LE advertisements test	
77.	DUT_PDLL_BLE_HCI_ADV_START_INIT	Bluetooth® LE HCI advertise start initialized.
78.	DUT_PDLL_BLE_HCI_ADV_START	Bluetooth® LE HCI advertise start started.

No.	Status	Description
79.	DUT_PDLL_BLE_HCI_ADV_START_OK	Bluetooth® LE HCI advertise start success.
80.	DUT_PDLL_BLE_HCI_ADV_START_FAILED	Bluetooth® LE HCI advertise start failed.
81.	DUT_PDLL_BLE_HCI_ADV_STOP_INIT	Bluetooth® LE HCI advertise stop initialized.
82.	DUT_PDLL_BLE_HCI_ADV_STOP_START	Bluetooth® LE HCI advertise stop started.
83.	DUT_PDLL_BLE_HCI_ADV_STOPPED_OK	Bluetooth® LE HCI advertise stop success.
84.	DUT_PDLL_BLE_HCI_ADV_STOP_FAILED	Bluetooth® LE HCI advertise stop failed.
85.	DUT_PDLL_BLE_HCI_ADV_SCAN_START	Bluetooth® LE HCI advertise scan started.
86.	DUT_PDLL_BLE_HCI_ADV_NOT_YET_FOUND	Bluetooth® LE HCI advertise not yet found.
87.	DUT_PDLL_BLE_HCI_ADV_FOUND	Bluetooth® LE HCI advertise found.
88.	DUT_PDLL_BLE_HCI_ADV_RSSI_FAILED	Bluetooth® LE HCI advertise RSSI failed.
89.	DUT_PDLL_BLE_HCI_ADV_FAILED	Bluetooth® LE HCI advertise failed.
	Production test – DUT packet transaction	
90.	DUT_PDLL_PKT_TX_START_INIT	RF packet TX initialized.
91.	DUT_PDLL_PKT_TX_START	RF packet TX start.
92.	DUT_PDLL_PKT_TX_STARTED_OK	RF packet TX started successfully.
93.	DUT_PDLL_PKT_TX_STARTED_FAILED	RF packet TX failed to start.
94.	DUT_PDLL_PKT_TX_ENDED_START	RF packet TX ended successfully.
95.	DUT_PDLL_PKT_TX_ENDED_OK	RF packet TX end initiated.
96.	DUT_PDLL_PKT_TX_ENDED_FAILED	RF packet TX failed to end.
	Production test – TX power measurement	
97.	DUT_BLE_TESTER_TX_PWR_PASSED	Bluetooth® LE tester TX power test passed.
98.	DUT_BLE_TESTER_TX_PWR_FAILED	Bluetooth® LE tester TX power test failed.
	Production test – TX carrier offset measure	
99.	DUT_BLE_TESTER_TX_OFFS_PASSED	Bluetooth® LE tester TX frequency offset test passed.
100.	DUT_BLE_TESTER_TX_OFFS_FAILED	Bluetooth® LE tester TX frequency offset test failed.
	Production test – TX modulation index measure	
101.	DUT_BLE_TESTER_TX_MOD_IDX_PASSED	Bluetooth® LE tester TX modulation index test passed.
102.	DUT_BLE_TESTER_TX_MOD_IDX_FAILED	Bluetooth® LE tester TX modulation index test failed.
	Production test – DUT start packet RX	
103.	DUT_PDLL_PKT_RX_STATS_START_INIT	RF RX packet test with statistics start initialized.
104.	DUT_PDLL_PKT_RX_STATS_START	RF RX packet test with statistics start.
105.	DUT_PDLL_PKT_RX_STATS_STARTED_OK	RF RX packet test with statistics started successfully.
106.	DUT_PDLL_PKT_RX_STATS_START_FAILED	RF RX packet test with statistics started failed.
	Production test – DUT stop packet RX	
107.	DUT_PDLL_PKT_RX_STATS_STOP_INIT	RF RX packet test with statistics stop initialized.

No.	Status	Description
108.	DUT_PDLL_PKT_RX_STATS_STOP_START	RF RX packet test with statistics stop.
109.	DUT_PDLL_PKT_RX_STATS_STOPPED_OK	RF RX packet test with statistics stopped successfully.
110.	DUT_PDLL_PKT_RX_STATS_STOP_FAILED	RF RX packet test with statistics stop failed.
	Production test – RX sensitivity test	
111.	DUT_BLE_TESTER_RX_TEST_PASSED	Bluetooth® LE tester RX sensitivity test passed.
112.	DUT_BLE_TESTER_RX_TEST_FAILED	Bluetooth® LE tester RX sensitivity test failed.
	Production test – HCI error	
113.	DUT_PDLL_HCI_STANDARD_ERROR	An error occurred in HCI commands.
	Production test - Golden Unit RSSI	
114.	DUT_PDLL_GU_RF_RX_TEST_PASSED	Golden Unit RF RX packet test passed.
115.	DUT_PDLL_GU_RF_RX_TEST_FAILED	Golden Unit RF RX packet test failed.
	Production test - GPIO/LED test	
116.	DUT_PDLL_GPIO_TOGGLE_INIT	GPIO-LED test operation initialized.
117.	DUT_PDLL_GPIO_TOGGLE_START	GPIO-LED test operation start.
118.	DUT_PDLL_GPIO_TOGGLE_FINISHED_OK	GPIO-LED test operation completed successfully.
119.	DUT_PDLL_GPIO_TOGGLE_ERROR	GPIO-LED test operation error.
120.	DUT_PDLL_GPIO_TOGGLE_FAILED	GPIO-LED test operation failed.
121.	DUT_PDLL_GPIO_TOGGLE_PASSED	GPIO-LED test operation passed.
	Production test – GPIO connection test	
122.	DUT_PDLL_GPIO_CONNECTION_INIT	GPIO connection test operation initialized.
123.	DUT_PDLL_GPIO_SET_START	GPIO connection test set operation start.
124.	DUT_PDLL_GPIO_SET_ERROR	GPIO connection test set operation error.
125.	DUT_PDLL_GPIO_SET_FINISHED_OK	GPIO connection test set operation success.
126.	DUT_PDLL_GPIO_GET_START	GPIO connection test get operation start.
127.	DUT_PDLL_GPIO_GET_ERROR	GPIO connection test get operation passed.
128.	DUT_PDLL_GPIO_GET_FINISHED_OK	GPIO connection test get operation success.
129.	DUT_PDLL_GPIO_CONNECTION_ERROR	GPIO connection test operation error.
130.	DUT_PDLL_GPIO_CONNECTION_FAILED	GPIO connection test operation failed.
131.	DUT_PDLL_GPIO_CONNECTION_PASSED	GPIO connection test operation completed successfully.
	Production test - Sensor test	
132.	DUT_PDLL_SENSOR_TEST_INIT	Sensor test action initialized.
133.	DUT_PDLL_SENSOR_TEST_START	Sensor test action start.
134.	DUT_PDLL_SENSOR_TEST_OK	Sensor test action ended successfully.
135.	DUT_PDLL_SENSOR_TEST_FAILED	Sensor test action failed.
136.	DUT_PDLL_SENSOR_TEST_DATA_MATCH_OK	Sensor test action data matched.

No.	Status	Description
137.	DUT_PDLL_SENSOR_TEST_DATA_MATCH_FAILED	Sensor test action data match failure.
	Production test - Custom action test	
138.	DUT_PDLL_CUSTOM_ACTION_INIT	Custom test action initialized.
139.	DUT_PDLL_CUSTOM_ACTION_START	Custom test action start.
140.	DUT_PDLL_CUSTOM_ACTION_OK	Custom test action ended successfully.
141.	DUT_PDLL_CUSTOM_ACTION_FAILED	Custom test action failed.
142.	DUT_PDLL_CUSTOM_ACTION_DATA_MATCH_OK	Custom test action data matched.
143.	DUT_PDLL_CUSTOM_ACTION_DATA_MATCH_FAILED	Custom test action data match failure.
	Production test – UART loop test	
144.	DUT_PDLL_UART_LOOP_INIT	UART loop test initialized.
145.	DUT_PDLL_UART_LOOP_START	UART loop test start.
146.	DUT_PDLL_UART_LOOP_OK	UART loop test ended successfully.
147.	DUT_PDLL_UART_LOOP_FAILED	UART loop test failed.
	Production test – Sleep current measurement	
148.	DUT_SLEEP_CURRENT_MEASURE_INIT	Sleep current measurement test initialized.
149.	DUT_SLEEP_CURRENT_MEASURE_START	Sleep current measurement test started.
150.	DUT_SLEEP_DEVICE_SLEPT_OK	Sleep current measurement test, device went to sleep successfully.
151.	DUT_SLEEP_CURRENT_MEASURE_ERROR	Sleep current measurement test error.
152.	DUT_SLEEP_CURRENT_MEASURE_PASSED	Sleep current measurement test passed.
153.	DUT_SLEEP_CURRENT_MEASURE_FAILED	Sleep current measurement test failed.
	Production test – Peripheral current measurement	
154.	DUT_PDLL_PERIPH_AMMETER_TEST_INIT	Peripheral current measurement test initialized.
155.	DUT_PDLL_PERIPH_AMMETER_TEST_START	Peripheral current measurement test started.
156.	DUT_PDLL_PERIPH_AMMETER_TEST_ERROR	Peripheral current measurement test error.
157.	DUT_PDLL_PERIPH_AMMETER_TEST_PASSED	Peripheral current measurement test passed.
158.	DUT_PDLL_PERIPH_AMMETER_TEST_FAILED	Peripheral current measurement test failed.
	Memory programming - Generic errors	
159.	DUT_UDLL_SUCCESS	UDLL returned success.
160.	DUT_UDLL_ACTION_RESPONSE_ERROR	UDLL device responded with error.
161.	DUT_UDLL_UART_RX_TIMEOUT_ERROR	UDLL UART RX timeout. Cannot communicate with the DUT or DUT is not present.
162.	DUT_UDLL_NO_CRC_MATCH_ERROR	UDLL CRC match error.
163.	DUT_UDLL_PROG_PARAMS_ERROR	UDLL programming parameter error.
164.	DUT_UDLL_DEVICE_PARAMS_ERROR	UDLL device parameter error.

No.	Status	Description
165.	DUT_UDLL_UART_WRITE_ERROR	UDLL UART write returned error.
166.	DUT_UDLL_UART_READ_ERROR	UDLL UART read returned error.
167.	DUT_UDLL_INTERNAL_ERROR	UDLL internal error.
168.	DUT_UDLL_COM_PORT_INIT_ERROR	UDLL COM port initialization error.
169.	DUT_UDLL_COM_PORT_ERROR	UDLL COM port error.
170.	DUT_UDLL_CANNOT_ALLOCATE_MEMORY	UDLL cannot allocate memory.
171.	DUT_UDLL_READ_FILE_SIZE_ERROR	UDLL read file size error.
172.	DUT_UDLL_CANNOT_OPEN_FW_FILE	UDLL cannot open firmware file.
173.	DUT_UDLL_CANNOT_OPEN_IMAGE_FILE	UDLL cannot open image file.
174.	DUT_UDLL_UART_PINS_PATCH_ERROR	UDLL cannot patch the UART pins into the firmware file.
175.	DUT_UDLL_INVALID_DBG_PARAMS	UDLL invalid debug library (dbg_dll.dll) parameters.
176.	DUT_UDLL_DBG_DLL_ERROR	UDLL debug library (dbg_dll.dll) access error.
	Memory programming – QSPI/OQSPI memory initialization	
177.	DUT_UDLL_FLASH_INIT_INIT	QSPI/OQSPI initialization operation initialized.
178.	DUT_UDLL_FLASH_INIT_STARTED	QSPI/OQSPI initialization operation started.
179.	DUT_UDLL_FLASH_INIT_OK	QSPI/OQSPI initialization operation ended successfully.
180.	DUT_UDLL_FLASH_INIT_FAILED	QSPI/OQSPI initialization operation failed.
	Memory programming – QSPI/OQSPI memory erase	
181.	DUT_UDLL_FLASH_ERASE_INIT	QSPI/OQSPI erase operation initialized.
182.	DUT_UDLL_FLASH_ERASE_STARTED	QSPI/OQSPI erase operation started.
183.	DUT_UDLL_FLASH_ERASE_OK	QSPI /OQSPI erase operation ended successfully.
184.	DUT_UDLL_FLASH_ERASE_FAILED	QSPI/OQSPI erase operation failed.
	Memory programming - QSPI/OQSPI image write	
185.	DUT_UDLL_FLASH_IMG_WR_INIT	QSPI/OQSPI image write operation initialized.
186.	DUT_UDLL_FLASH_IMG_WR_STARTED	QSPI/OQSPI image write operation started.
187.	DUT_UDLL_FLASH_IMG_WR_OK	QSPI/OQSPI image write operation ended successfully.
188.	DUT_UDLL_FLASH_IMG_WR_FAILED	QSPI/OQSPI image write operation failed.
	Memory programming – OTP check empty	
189.	DUT_UDLL_OTP_CHECK_EMPTY_INIT	OTP check empty initialize.
190.	DUT_UDLL_OTP_CHECK_EMPTY_STARTED	OTP check empty operation started.
191.	DUT_UDLL_OTP_CHECK_EMPTY_STATUS	OTP check empty update status to UI.
192.	DUT_UDLL_OTP_CHECK_EMPTY_OK	OTP check empty finished successfully.
193.	DUT_UDLL_OTP_CHECK_INPUT_DATA_EMPTY_OK	OTP check empty input data are empty.
194.	DUT_UDLL_OTP_CHECK_SAME_DATA_OK	OTP check empty user data and device data are the same.

No.	Status	Description
195.	DUT_UDLL_OTP_CHECK_SKIP_IF_WRITTEN_OK	OTP check empty device already written with data.
196.	DUT_UDLL_OTP_CHECK_DIFFERENT_DATA_FAILED	OTP check empty device contains different data.
197.	DUT_UDLL_OTP_CHECK_EMPTY_FAILED	OTP check empty failed.
	Memory programming - OTP binary write	
198.	DUT_UDLL_OTP_BIN_WR_INIT	OTP binary write operation initialized.
199.	DUT_UDLL_OTP_BIN_WR_STARTED	OTP binary write operation started.
200.	DUT_UDLL_OTP_BIN_WR_OK	OTP binary write operation ended successfully.
201.	DUT_UDLL_OTP_BIN_WR_FAILED	OTP binary write operation failed.
	Memory programming - OTP configuration script write	
202.	DUT_UDLL_OTP_CS_WRITE_INIT	OTP CS field write operation initialized.
203.	DUT_UDLL_OTP_CS_WRITE_STARTED	OTP CS field write operation started.
204.	DUT_UDLL_OTP_CS_WRITE_NUM_OF_ENTRIES	OTP CS update the number of fields to be burned.
205.	DUT_UDLL_OTP_CS_WRITE_STATUS	OTP configuration script field write progress status update
206.	DUT_UDLL_OTP_CS_WRITE_RDBK_STARTED	OTP configuration script readback process started.
207.	DUT_UDLL_OTP_CS_WRITE_RDBK_STATUS	OTP configuration script readback progress status update
208.	DUT_UDLL_OTP_CS_WRITE_OK	OTP configuration script write ended successfully.
209.	DUT_UDLL_OTP_CS_WRITE_FAILED	OTP configuration script write failed.
	Memory programming - Memory write operation	
210.	DUT_UDLL_MEM_DATA_WR_INIT	Custom memory data burn operation started.
211.	DUT_UDLL_MEM_DATA_WR_INIT	Custom memory data burn operation started.
212.	DUT_UDLL_MEM_DATA_WR_STARTED	Custom memory data burn update interim status.
213.	DUT_UDLL_MEM_DATA_WR_OK	Custom memory data burn operation ended successfully.
214.	DUT_UDLL_MEM_DATA_WR_FAILED	Custom memory data burn operation failed.
	Memory programming - Memory read operation	
215.	DUT_UDLL_MEM_RD_INIT	Memory read operation initialized.
216.	DUT_UDLL_MEM_RD_STARTED	Memory read operation started.
217.	DUT_UDLL_MEM_RD_OK	Memory read operation ended successfully.
218.	DUT_UDLL_MEM_RD_FAILED	Memory read operation failed.
	Production tests – Scan test	
219.	DUT_PDLL_BLE_SCAN_INIT	Scan test initialize.
220.	DUT_PDLL_BLE_SCAN_START	Scan test started.
221.	DUT_PDLL_BLE_SCAN_NOT_YET_FOUND	Scan test, DUT not yet found.
222.	DUT_PDLL_BLE_SCAN_FOUND	Scan test, DUT found.
223.	DUT_PDLL_BLE_SCAN_RSSI_FAILED	Scan test, DUT found but RSSI failed.

No.	Status	Description
224.	DUT_PDLL_BLE_SCAN_FAILED	Scan test failed. DUT not found.
	Generic	
225.	DUT_GU_ERROR	GU has error
226.	INVALID_DUT_RES	Invalid DUT status result.

Appendix R Golden Unit Status Codes

Table 80 contains all the possible status codes the Golden Unit can have, followed by a brief description. The table categorizes the status based on the various states the Golden Unit may be during testing and programming the DUTs.

Table 80: Golden Unit Status Codes

No.	Status	Description
	Generic	
	GU_NOT_ACTIVE	GU is not active.
1.	GU_INTERNAL_SYSTEM_ERROR	GU internal system error.
2.	GU_COM_OPEN_OK	GU COM port opened successfully.
3.	GU_COM_OPEN_FAILED	GU COM port failed to open.
4.	GU_RESET_START	GU reset started.
5.	GU_RESET_OK	GU reset successfully.
6.	GU_RESET_FAILED	GU reset failed.
7.	GU_PDLL_NO_ERROR	PDLL returned success.
8.	GU_PDLL_PARAMS_ERROR	GU PDLL parameters have errors.
9.	GU_PDLL_RX_TIMEOUT	GU did not reply on a PDLL message request.
10.	GU_PDLL_TX_TIMEOUT	GU TX timeout when sending a message to CPLD.
11.	GU_PDLL_UNEXPECTED_EVENT	Received an unexpected message from the GU.
12.	GU_PDLL_CANNOT_ALLOCATE_MEMORY	PDLL cannot allocate memory.
13.	GU_PDLL_INTERNAL_ERROR	PDLL internal system error.
14.	GU_PDLL_THREAD_CREATION_ERROR	PDLL thread creation error.
15.	GU_PDLL_DBG_DLL_ERROR	PDLL debug library (dbg_dll.dll) access error.
16.	GU_PDLL_INVALID_DBG_PARAMS	PDLL invalid debug library (dbg_dll.dll) parameters.
	GU COM port handling	
17.	GU_PDLL_COM_PORT_INIT	GU COM port open initialized.
18.	GU_PDLL_COM_PORT_START	GU COM port open started.
19.	GU_PDLL_COM_PORT_OK	GU COM port opened OK.
20.	GU_PDLL_COM_PORT_FAILED	GU COM port FAILED.
	GU firmware version	
21.	GU_PDLL_FW_VERSION_GET_START	GU PDLL firmware version acquisition started.
22.	GU_PDLL_FW_VERSION_GET_OK	GU PDLL firmware version acquisition OK.
23.	GU_PDLL_FW_VERSION_GET_FAILED	GU PDLL firmware version acquisition FAILED.
24.	GU_PDLL_FW_VERSION_VALID	GU firmware version is valid.
25.	GU_PDLL_FW_VERSION_NOT_VALID	GU firmware version is not valid. Upgrade is needed.
	GU CPLD control	
26.	GU_PDLL_RDTESTER_INIT	PLT HW tester initializing.
27.	GU_PDLL_RDTESTER_INIT_START	PLT HW tester initialize started.

28.	GU_PDLL_RDTESTER_INIT_OK	PLT HW tester initialized successful.
29.	GU_PDLL_RDTESTER_INIT_FAILED	PLT HW tester initialization failed.
30.	GU_PDLL_RDTESTER_UART_CONNECT_INIT	PLT HW tester UART connection initialized.
31.	GU_PDLL_RDTESTER_UART_CONNECT_START	PLT HW tester UART connection started.
32.	GU_PDLL_RDTESTER_UART_CONNECT_OK	PLT HW tester UART connected successfully.
33.	GU_PDLL_RDTESTER_UART_CONNECT_FAILED	PLT HW tester UART connection failed.
34.	GU_PDLL_RDTESTER_UART_LOOPBACK_INIT	PLT HW tester UART loopback process initialized.
35.	GU_PDLL_RDTESTER_UART_LOOPBACK_START	PLT HW tester UART loopback process started.
36.	GU_PDLL_RDTESTER_UART_LOOPBACK_OK	PLT HW tester UART loopback process success.
37.	GU_PDLL_RDTESTER_UART_LOOPBACK_FAILED	PLT HW tester UART loopback process failed.
38.	GU_PDLL_RDTESTER_VBAT_UART_CNTRL_INIT	PLT HW tester VBAT/UART control initialized.
39.	GU_PDLL_RDTESTER_VBAT_UART_CNTRL_START	PLT HW tester VBAT/UART control started.
40.	GU_PDLL_RDTESTER_VBAT_UART_CNTRL_OK	PLT HW tester VBAT/UART control success.
41.	GU_PDLL_RDTESTER_VBAT_UART_CNTRL_FAILED	PLT HW tester VBAT/UART control failed.
42.	GU_PDLL_RDTESTER_VBAT_UART_RST_CNTRL_INIT	PLT HW tester VBAT/UART/Reset control initialized.
43.	GU_PDLL_RDTESTER_VBAT_UART_RST_CNTRL_START	PLT HW tester VBAT/UART/Reset control started.
44.	GU_PDLL_RDTESTER_VBAT_UART_RST_CNTRL_OK	PLT HW tester VBAT/UART/Reset control success.
45.	GU_PDLL_RDTESTER_VBAT_UART_RST_CNTRL_FAILED	PLT HW tester VBAT/UART/Reset control failed.
46.	GU_PDLL_RDTESTER_VPP_CNTRL_INIT	PLT HW tester VPP control initialized.
47.	GU_PDLL_RDTESTER_VPP_CNTRL_START	PLT HW tester VPP control started.
48.	GU_PDLL_RDTESTER_VPP_CNTRL_OK	PLT HW tester VPP control success.
49.	GU_PDLL_RDTESTER_VPP_CNTRL_FAILED	PLT HW tester VPP control failed.
50.	GU_PDLL_RDTESTER_RST_PULSE_INIT	PLT HW tester Reset pulse control initialized.
51.	GU_PDLL_RDTESTER_RST_PULSE_START	PLT HW tester Reset pulse control started.
52.	GU_PDLL_RDTESTER_RST_PULSE_OK	PLT HW tester Reset pulse control success.
53.	GU_PDLL_RDTESTER_RST_PULSE_FAILED	PLT HW tester Reset pulse control failed.
54.	GU_PDLL_RDTESTER_UART_PULSE_INIT	PLT HW tester XTAL trim pulse in UART TX pin initialized.
55.	GU_PDLL_RDTESTER_UART_PULSE_START	PLT HW tester XTAL trim pulse in UART TX pin started.
56.	GU_PDLL_RDTESTER_UART_PULSE_OK	PLT HW tester XTAL trim pulse in UART TX pin success.
57.	GU_PDLL_RDTESTER_UART_PULSE_FAILED	PLT HW tester XTAL trim pulse in UART TX pin failed.
58.	GU_PDLL_RDTESTER_XTAL_PULSE_INIT	PLT HW tester XTAL trim pulse in GATE pin initialized.
59.	GU_PDLL_RDTESTER_XTAL_PULSE_START	PLT HW tester XTAL trim pulse in GATE pin started.
60.	GU_PDLL_RDTESTER_XTAL_PULSE_OK	PLT HW tester XTAL trim pulse in GATE pin success.

61.	GU_PDLL_RDTESTER_XTAL_PULSE_FAILED	PLT HW tester XTAL trim pulse in GATE pin failed.
62.	GU_PDLL_RDTESTER_PULSE_WIDTH_INIT	PLT HW tester pulse width initialized.
63.	GU_PDLL_RDTESTER_PULSE_WIDTH_START	PLT HW tester pulse width started.
64.	GU_PDLL_RDTESTER_PULSE_WIDTH_OK	PLT HW tester pulse width success.
65.	GU_PDLL_RDTESTER_PULSE_WIDTH_FAILED	PLT HW tester pulse width failed.
66.	GU_PDLL_RDTESTER_VBAT_CNTRL_INIT	PLT HW tester VBAT control initialized.
67.	GU_PDLL_RDTESTER_VBAT_CNTRL_START	PLT HW tester VBAT control started.
68.	GU_PDLL_RDTESTER_VBAT_CNTRL_OK	PLT HW tester VBAT control success.
69.	GU_PDLL_RDTESTER_VBAT_CNTRL_FAILED	PLT HW tester VBAT control failed.
70.	GU_PDLL_RDTESTER_INVALID_COMMAND	PLT HW tester unknown command.
	GU RF packet TX for DUT RSSI RF test	
71.	GU_PDLL_PKT_TX_START_INIT	GU RF packet TX initialized.
72.	GU_PDLL_PKT_TX_START	GU RF packet TX started.
73.	GU_PDLL_PKT_TX_STARTED_OK	GU RF packet TX success.
74.	GU_PDLL_PKT_TX_STARTED_FAILED	GU RF packet TX failed.
75.	GU_PDLL_PKT_TX_ENDED_OK	GU RF packet TX ended successfully.
76.	GU_PDLL_PKT_TX_ENDED_FAILED	GU RF packet TX ended failed.
	GU GPIO toggling for sanity test	
77.	GU_PDLL_GPIO_TOGGLE_INIT	GU GPIO toggle operation initialized.
78.	GU_PDLL_GPIO_TOGGLE_START	GU GPIO toggle operation start.
79.	GU_PDLL_GPIO_TOGGLE_FINISHED_OK	GU GPIO toggle operation completed successfully.
80.	GU_PDLL_GPIO_TOGGLE_FAILED	GU GPIO toggle operation failed.
	GU Bluetooth® LE advertising scan test	
81.	GU_PDLL_BLE_SCAN_INIT	GU scan operation initialized.
82.	GU_PDLL_BLE_SCAN_START	GU scan operation started.
83.	GU_PDLL_BLE_SCAN_RESET	GU scan operation reset.
84.	GU_PDLL_BLE_SCAN_RETRY	GU scan operation retry.
85.	GU_PDLL_BLE_SCAN_OK	GU scan operation completed successfully.
86.	GU_PDLL_BLE_SCAN_FAILED	GU scan operation failed.
	General	
87.	GU_PDLL_HCI_STANDARD_ERROR	HCI communication error.
88.	GU_ALL_DUT_FAILED	All DUTs failed.
89.	INVALID_GU_RES	Invalid GU status result.

Revision History

Revision	Date	Description
4.8	02-Nov-2022	Updated for SmartBond_PLT_v_5.0 for DA1470x product family.
4.7	14-Feb-2022	Updated Appendix S - Connecting DA1468x DK Pro Motherboard for Current Measurements.
4.6	24-Jan-2022	Updated logo, disclaimer, copyright.
4.5	01-Oct-2020	Editorial, document debugging
4.4	08-May-2020	Updated for SmartBond_PLT_v4.4 software release
4.3	17-Jul-2018	Updated for DA1458x_DA1468x_PLT_v4.3 software release
4.2	10-Oct-2017	Updated for DA1458x_DA1468x_PLT_v4.2 software release
4.1	06-Oct-2017	Updated for DA1458x_DA1468x_PLT_v4.1 software release
4.0	22-Dec-2016	Updated for DA1458x_DA1468x_PLT_v4.0 software release
3.0	22-May-2016	Adding changes for the DA1468x
2.0	04-May-2016	Adding text and drawings
1.1	18-Jan-2016	CLI part is added and Rev D. PLT Hardware
1.0	08-Jul-2015	Initial version.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

RoHS Compliance

Renesas'-ref suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.