# Connectivity Production Line Tool

This document describes the Renesas Connectivity Production Line Tool. The Connectivity PLT hardware, as well as various software applications, are explained in detail. The purpose of this document is to guide users in the setup and operation of the tool.

## Contents

# Figures

# Tables

# 1. Terms and Definitions

| | |
|---|---|
| BDA | Bluetooth® Device Address |
| Bluetooth® LE | Bluetooth® Low Energy |
| CS | Configuration Script (a Renesas DA14xxx devices memory area related to production testing) |
| DIP Switch | Dual in-line package switch |
| DMM | Digital Multimeter |
| DUT | Device Under Test |
| HCI | Host Controller Interface |
| PLT | Production Line Tool |
| RF | Radio Frequency |
| RFTU | Radio Frequency Test Unit |

# 2. References

[1] DA14531 Datasheet, Renesas Electronics.

[2] DA14535 USB kit - DA14535-00FXDEVKT-U.

[3] DA14695 USB kit - DA14695-00HQDEVKT-U.

[4] https://www.microchip.com/en-us/development-tool/sam-ba-in-system-programmer

[5] https://www.ni.com/en/support/downloads/drivers/download.ni-visa.html#565016

**Note 1** References are for the latest published version, unless otherwise indicated.

# 3. Introduction

The Renesas Connectivity PLT is designed to test, calibrate, and perform memory operations for multiple devices under test (DUTs) in parallel. Specifically, up to four Connectivity PLT boards with eight test sites each can be daisy-chained to achieve 32 test sites.

The following parts are delivered with the tool:

- Hardware:
  - Main board (Figure 1)
  - Power cable
- Software:
  - Installation package for Windows environment
- Documents:
  - User manual
  - Electrical schematics
  - Gerber files
  - Bill of Materials

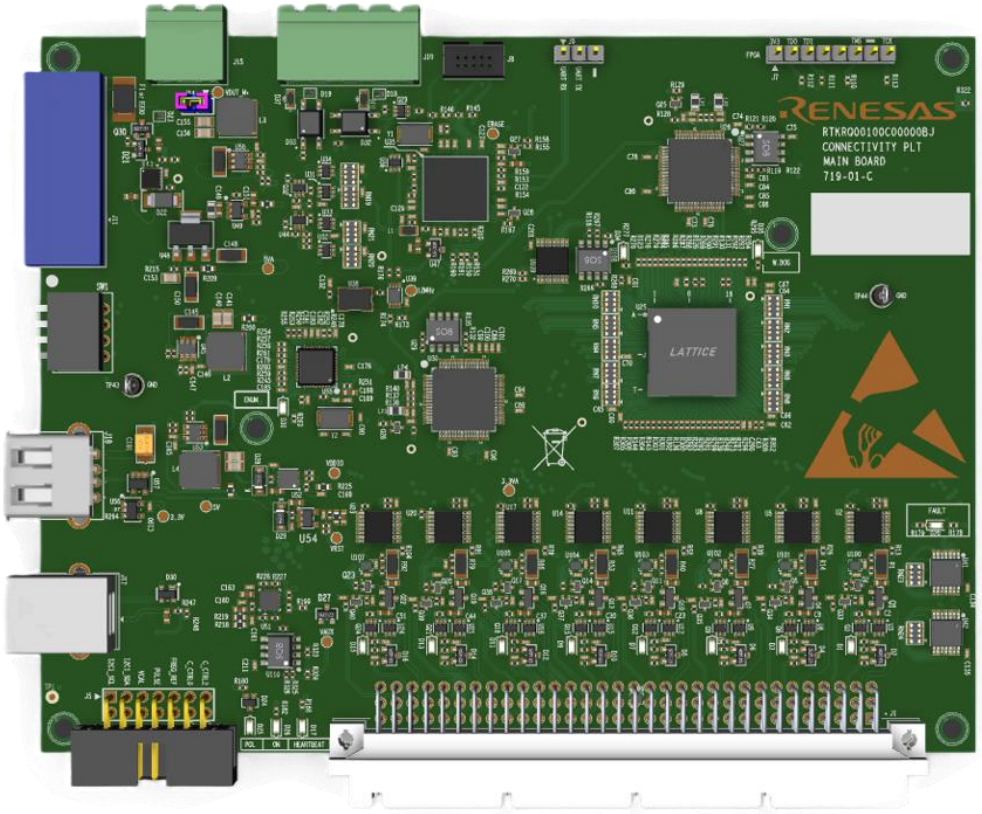



**Figure 1. Main board**

# 4. System Overview

## 4.1 Typical Setup

The Connectivity PLT is a configurable tool that can be adapted to the needs of the production line. Figure 2 shows a simple configuration that consists of a single Connectivity PLT board with eight connected DUTs powered by an external power supply. The host PC is connected to the Connectivity PLT board and to an RF test unit (in this case the SmartBond™ DA14695 Bluetooth® Low Energy 5.2 USB Development Kit). The host has a connected screen, so it can run the Connectivity PLT execution application.

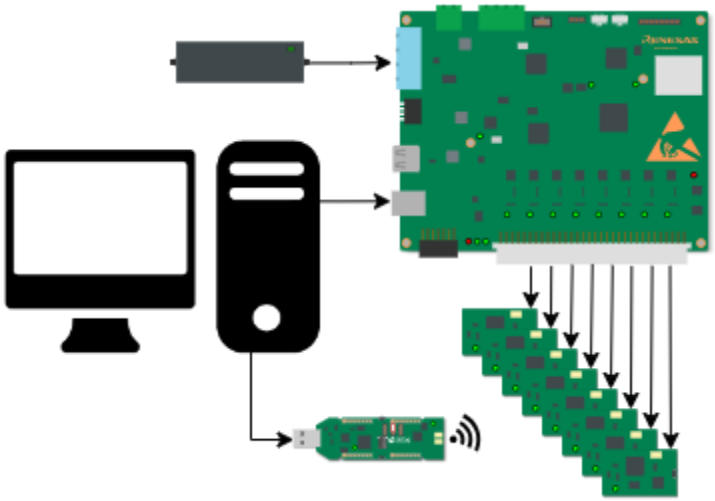


**Figure 2. Basic setup**

## 4.2 Typical Test Sequence

The tool performs the following actions:

1. The production test firmware is downloaded to the DUT.
2. Factory timestamp is read from OTP.
3. Automatic crystal (XTAL) trimming is performed.
4. RF RSSI RX tests are performed.
5. Advertise scan test is performed.
6. Flash programmer firmware is downloaded to RAM.
7. The (customer) application firmware is downloaded (into OTP, QSPI flash).
8. The OTP Configuration Script and Header are written to OTP.
9. OTP and Flash memory are read back for verification.
10. DUTs are powered down.

All actions are performed at the same time on all DUTs up to 32 devices.

# 5. Hardware Setup

## 5.1 Power Supply Connections

The Connectivity PLT is designed for use with an external power supply connected to J11. The type of connector is a 4-pin terminal block (TBP01R1W-508-04BE). Table 1 shows the requirements for this power supply, and Figure 3 shows the pinout of J11, indicating that the power connector allows for daisy chaining.

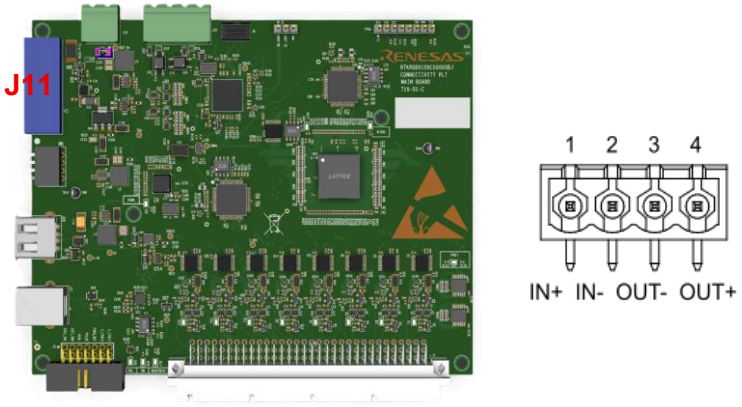



**Figure 3. Power connector J11**

**Table 1. Power supply characteristics**

| Parameter | Description | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{IN}$ | Input supply voltage | | 11 | 12 | 15 | V |
| $I_{IN}$ | Input supply current | Single Connectivity PLT board | 2 | | | A |
| $I_{IDLE}$ | IDLE current draw | Single Connectivity PLT board | | 200 | | mA |

## 5.2 DUT Connections

The DUT connector is an interface for up to eight application boards that can be connected. The type of connector is a 96-pin DIN 41612 male connector from Amphenol (86093967113745ELF). Mating parts can be found on the producer's website. The power and the signals to the application board go through this connector. The connector is divided into eight sections, one section per DUT (Figure 4). The pin layout of each section is the same and can be found in Figure 5.

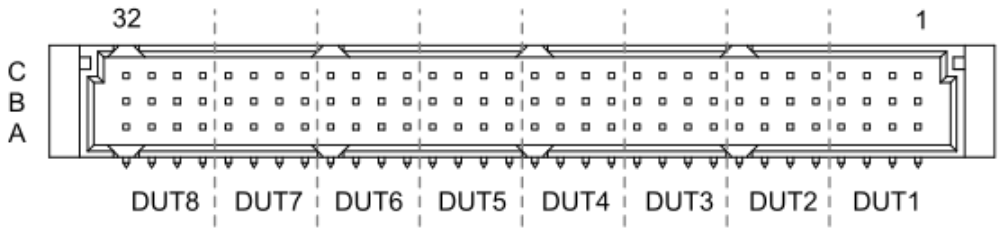



**Figure 4. DUT connector**

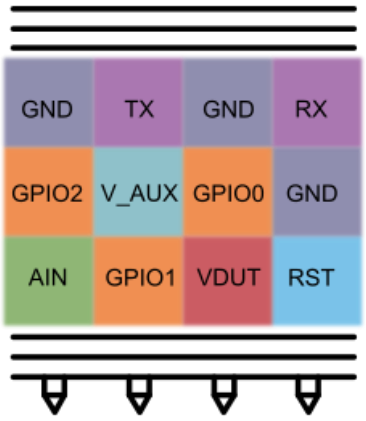



**Figure 5. DUT connector pin layout**

**Table 2. DUT connection characteristics**

| Parameter | Description | Conditions | Min | Typical | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DUT}$ | Configurable DUT supply voltage | | 1.1 | | 5.0 | V |
| $I_{DUT}$ | DUT current | Independent of $V_{DUT}$ | | | 300 | mA |
| $V_{AUX}$ | Configurable auxiliary voltage | | 0 | | 5.0 | V |
| $I_{AUX}$ | | 1 kΩ series resistor | | | 1 | mA |
| AIN | Analog input voltage | | 0 | | 5.0 | V |
| | Input impedance | | | 830 | | kΩ |
| | ADC resolution (LSB) | | | 1.25 | | mV |
| $V_{GPIO}$ | Configurable GPIO voltage | | 1.2 | | 3.3 | V |
| $V_{GPIO\_OUT}$ | GPIO out high voltage | Percentage of $V_{GPIO}$ | 70 | | | % |
| | GPIO out low voltage | Percentage of $V_{GPIO}$ | | | 30 | % |
| $I_{GPIO\_OUT}$ | GPIO output current | Depends on FPGA setting | | | | mA |
| $V_{GPIO\_IN}$ | GPIO in high threshold | Percentage of $V_{GPIO}$ | 70 | | | % |
| | GPIO in low threshold | Percentage of $V_{GPIO}$ | | | 30 | % |
| | GPIO in voltage range | | 0 | | 3.3 | V |
| $V_{TX}$ | TX high voltage | Percentage of $V_{GPIO}$ | 70 | | | % |
| | TX low voltage | Percentage of $V_{GPIO}$ | | | 30 | % |
| | Configurable reset voltage | Percentage of $V_{GPIO}$ | 70 | | | % |

## 5.3 GPIO Connections

The GPIO connector is an optional galvanically isolated interface to external test equipment such as the testing fixture. The connector consists of one general-purpose input and one general-purpose output. The output does not supply its own power; it must be externally powered. The type of connector is a 4-pin WR-TBL terminal block from Würth Elektronik (691313510004).

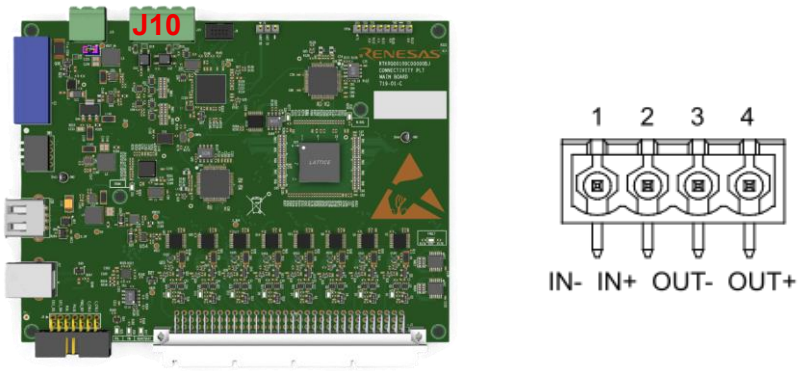



**Figure 6. GPIO connector J10**

**Table 3. GPIO characteristics**

| Parameter | Description | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| $V_{OUTPUT}$ | General-purpose output voltage | | 24 | 80 | V |
| $I_{OUTPUT}$ | General-purpose output current | | | 5 | mA |
| $V_{INPUT}$ | General-purpose input voltage | 2 | | 24 | V |

## 5.4 Daisy Chaining

The Connectivity PLT allows for up to four Connectivity PLT mainboards to be connected in a chain. Each Connectivity PLT board has a USB in (J17) and a USB out (J18). Each output connects to the next board's input, up to a maximum of four boards, see Figure 7. The USB must be chained from board to board and cannot be connected in any other way (such as through a hub). The power may however be connected as desired, if the requirements mentioned in Section 5.1 are met. Figure 8 shows an example of alternative power connections.

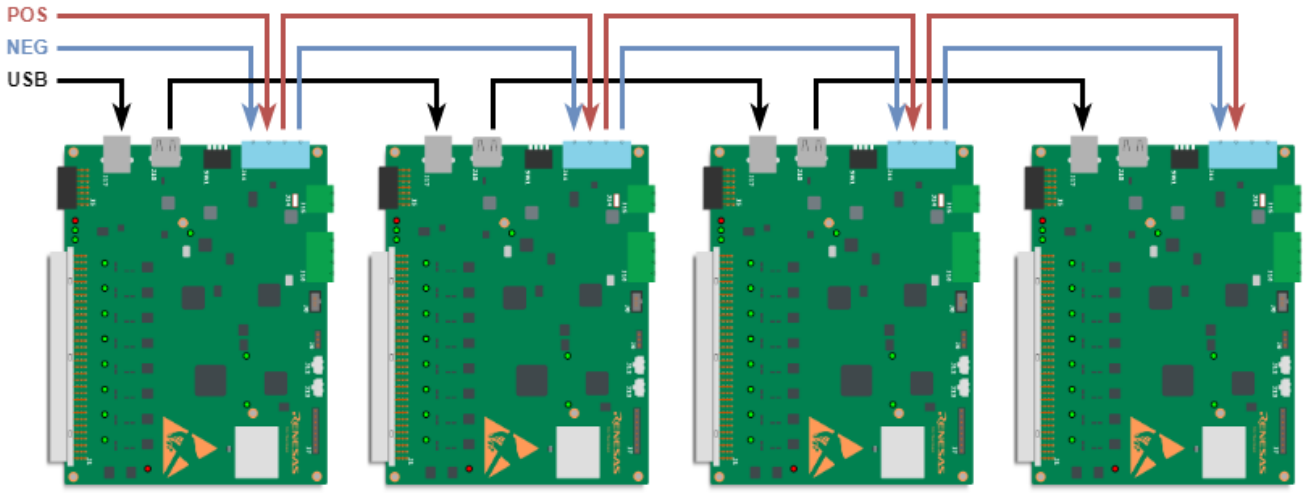



**Figure 7. Connectivity PLT chain**

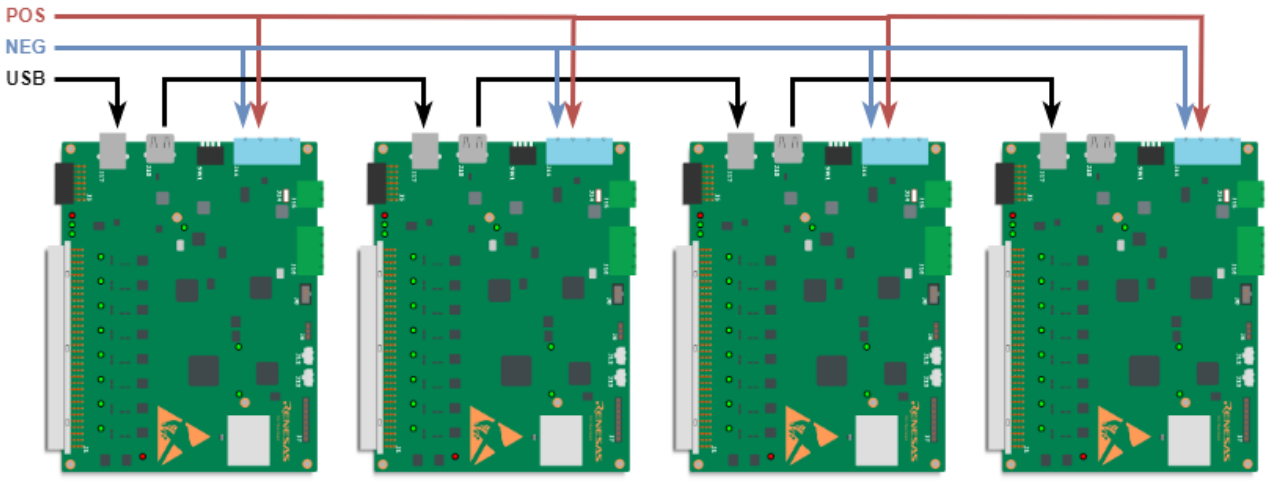



**Figure 8. Connectivity PLT chain alternative**

### 5.4.1 Addressing

When using multiple Connectivity PLT boards, the address on each of the boards needs to be set using DIP switches SW1. The switches are numbered, and the ON position is marked on the switch housing. The addressing is in binary format and switch number 1 is the least significant bit. For example, to set board number 3, switches 1 and 2 should be ON, and switches 3 and 4 should be OFF.

### 5.4.2 Sleep Current Measurement Using External DMM

Using externally connected current measurement equipment on J15, the Connectivity PLT can provide precise sleep current measurements for the DUTs. The current measurement is done on the VDUT rail so all DUTs are measured simultaneously. A jumper on J14 is used to bypass the external current measurement.

To set up the sleep current measurement with an external instrument, jumper J14 must be removed and a connection between J15 and the instrument needs to be made. Two wires from the positive and negative current measurement connections of the DMM must be connected to the terminal block for J15. The instrument must be

NI VISA compatible through a USB connection or through a GPIB interface. The implemented protocol is compatible with Agilent/Keysight 3440x common among many DMMs.

The NI VISA drivers and software also need to be installed, see Section 6.4.

### 5.4.3 RFTUs

Radio Frequency Test Unit can be connected directly to PLT board on USB port.

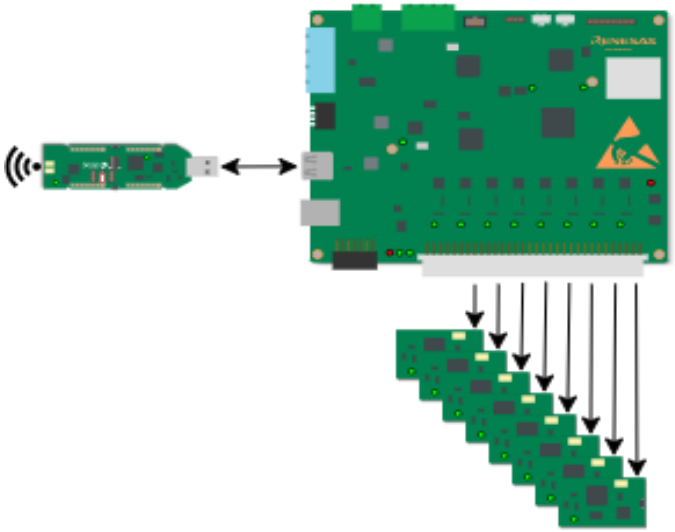


**Figure 9. RFTU connected on PLT Board**

If chaining is used, the RFTU can be connected to the last PLT board.

For combo devices, that support multiple protocols (for example, Wi-Fi and Bluetooth® LE) different RFTUs are needed for each protocol. See an example of RFTU connections in Section 5.5.

### 5.4.4 RFTU Auto-Detection

The Connectivity PLT service supports automatic enumeration of Radio Frequency Test Units (RFTUs), both Bluetooth® LE and Wi-Fi. When enabled, this feature allows the system to automatically detect and assign the correct serial COM port to each RFTU at startup, without requiring manual configuration.

Auto-enumeration is designed to handle environments where COM port assignments may change between sessions, for example, when devices are reconnected or the host PC is restarted. By powering the RFTUs on sequentially and monitoring which new COM port appears after each power-on event, the system can reliably map each RFTU to the PLT board it is physically connected to and verify the connection by communicating with the device.

This process runs transparently during system initialization and requires no user intervention. It applies independently to Bluetooth® LE and Wi-Fi RFTUs and also covers RFTUs connected directly to the host PC rather than through a PLT board.

## 5.5 Production Line Setup

To deploy the Connectivity PLT in a production environment, there are some considerations that must be made regarding the RF environment. If any RF tests are to be performed by the Connectivity PLT, there should be minimal background noise and interference. The recommended way to achieve this in any production environment is to place the Connectivity PLT main board(s), DUT's, and RF testing unit in a shielded box. Figure 10 shows a single board and single RFTU setup, and Figure 13. When multiple PLT main boards and multiple RFTUs are used, they can be set up in multiple shieled boxes (Figure 12) or a single shielded box (Figure 13).

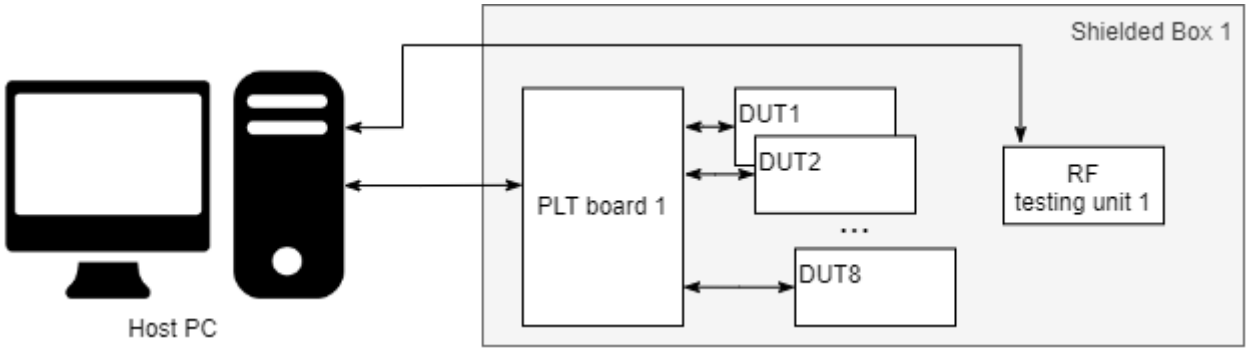



**Figure 10. Connectivity PLT in a shielded box**

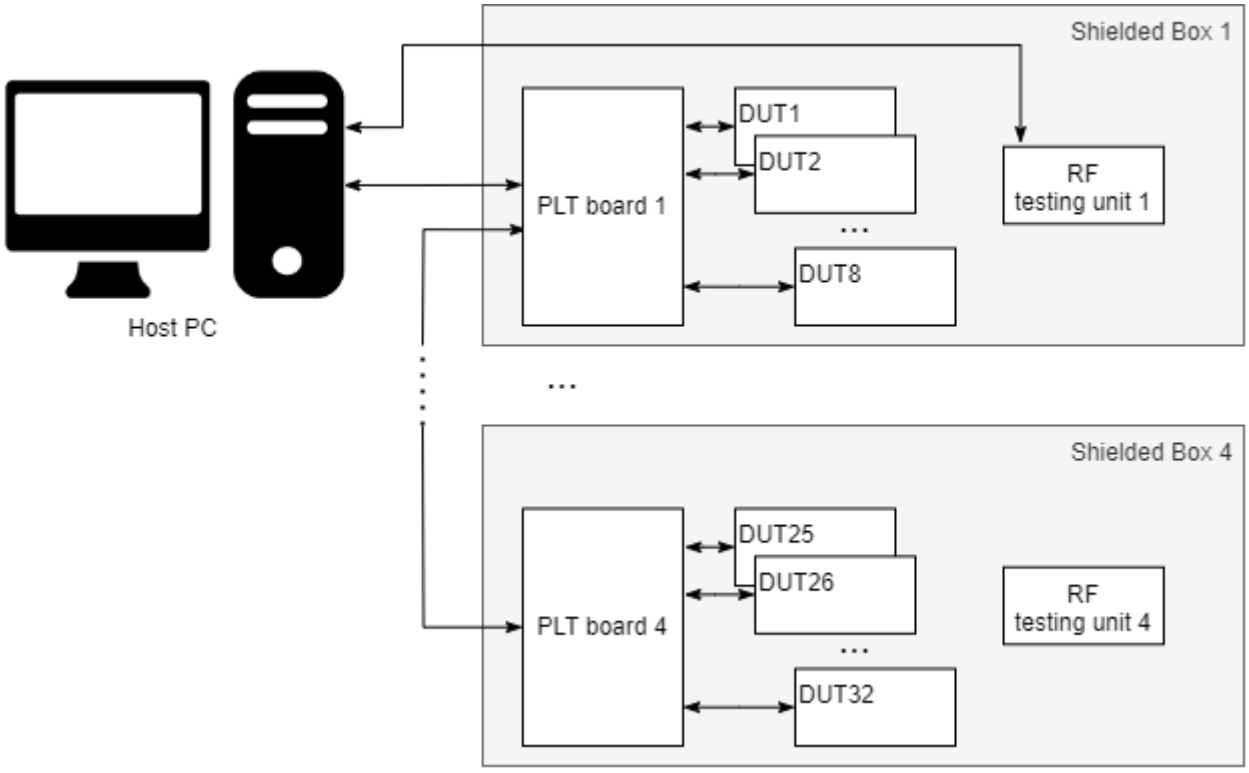



**Figure 11. Connectivity PLT chain in multiple shielded boxes**

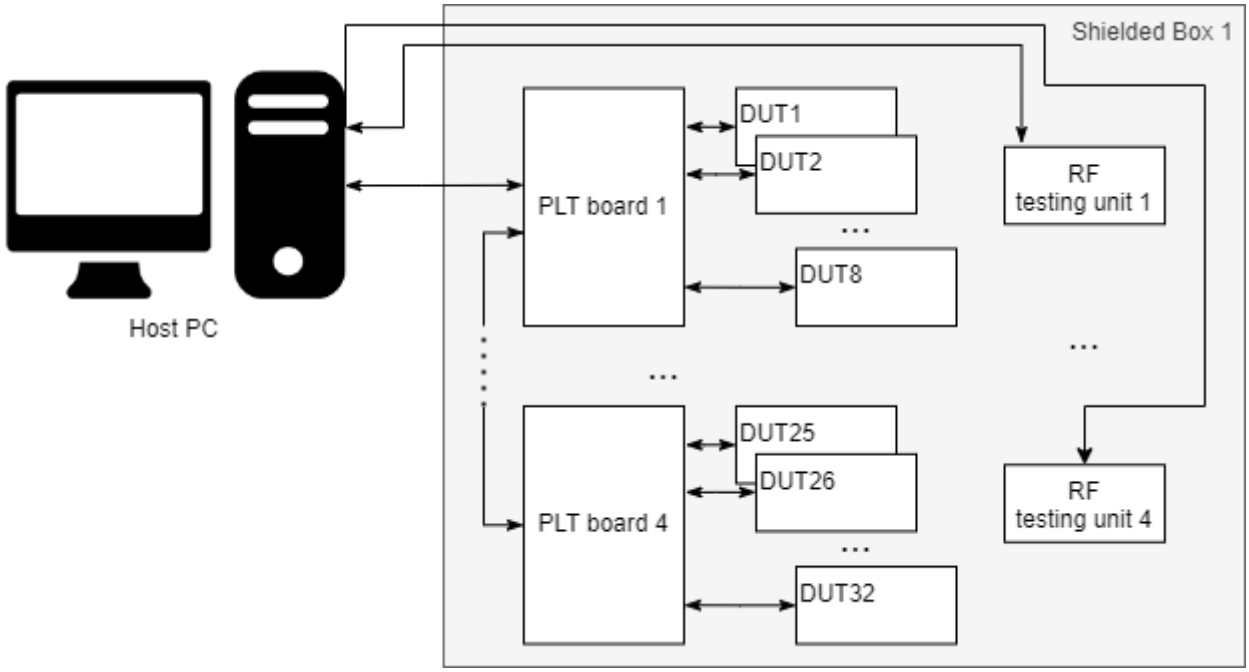



**Figure 12. Connectivity PLT chain in single shielded box with multiple RF testing units**

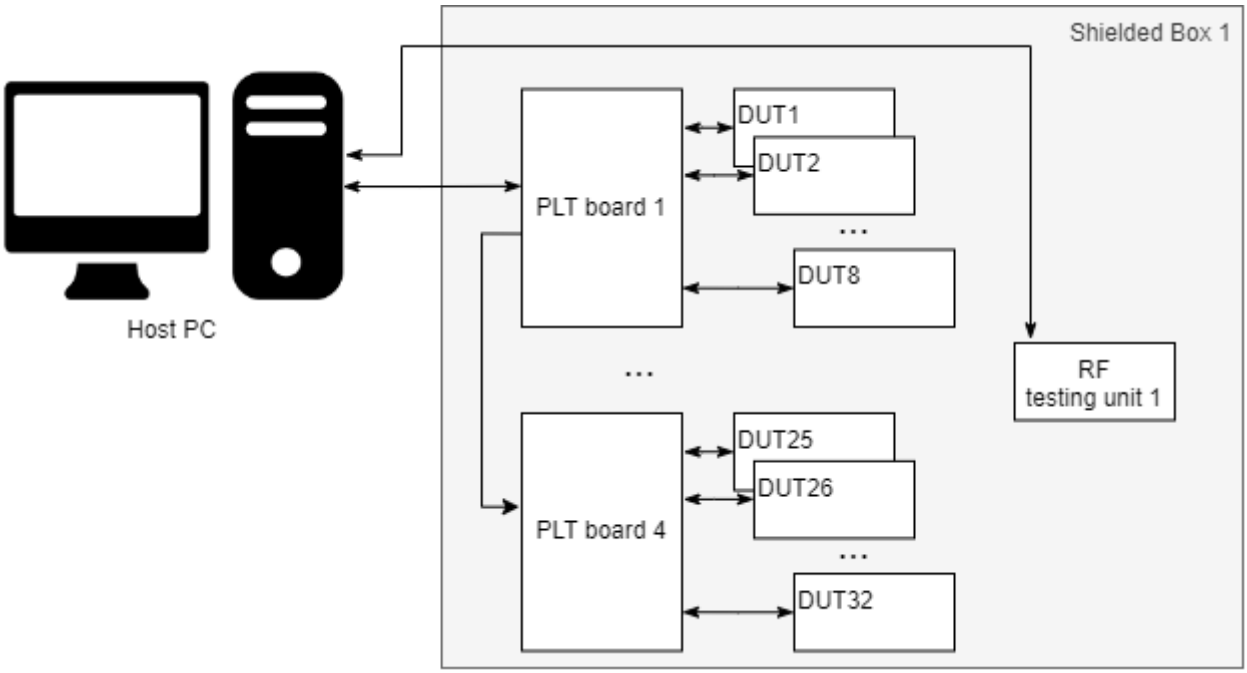



**Figure 13. Connectivity PLT chain in single shielded box**

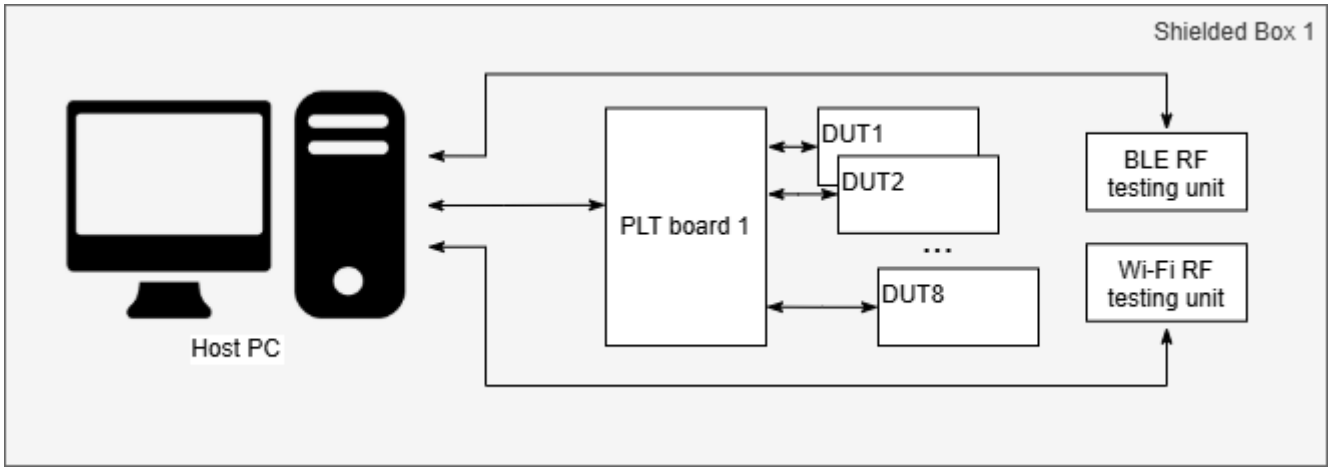



**Figure 14. Connectivity PLT with Bluetooth® LE and Wi-Fi RF testing units**

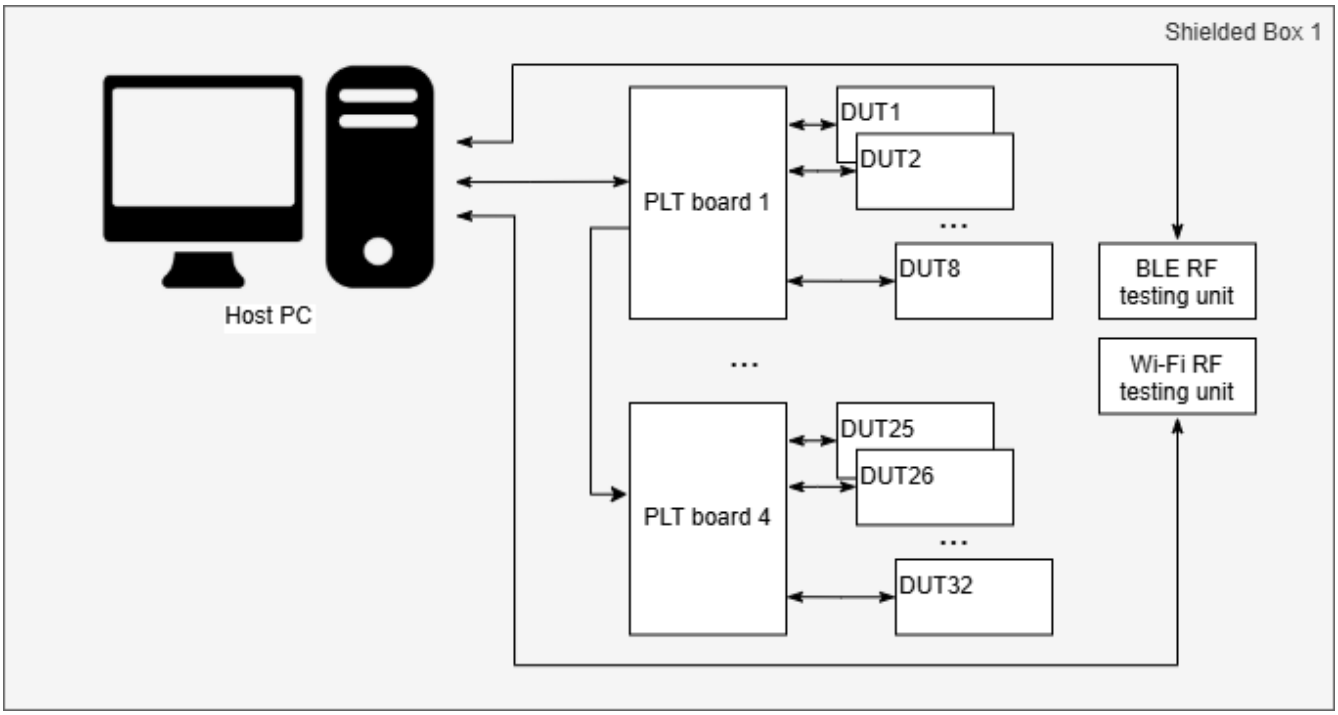



**Figure 15. Connectivity PLT chain with Bluetooth® LE and Wi-Fi RF testing units**

# 6. Installation

## 6.1 Introduction

This section provides information on how to install the Connectivity PLT software. The Connectivity PLT is not a single application but a group of services and applications that can be installed on a single host or multiple networked machines. The installation process differs depending on the individual setup of the production line.

A single installer is used for all types of installation.

## 6.2 Architecture and Different Setup Strategies

The Connectivity PLT is split into two services and a few user interface applications, see Figure 16.

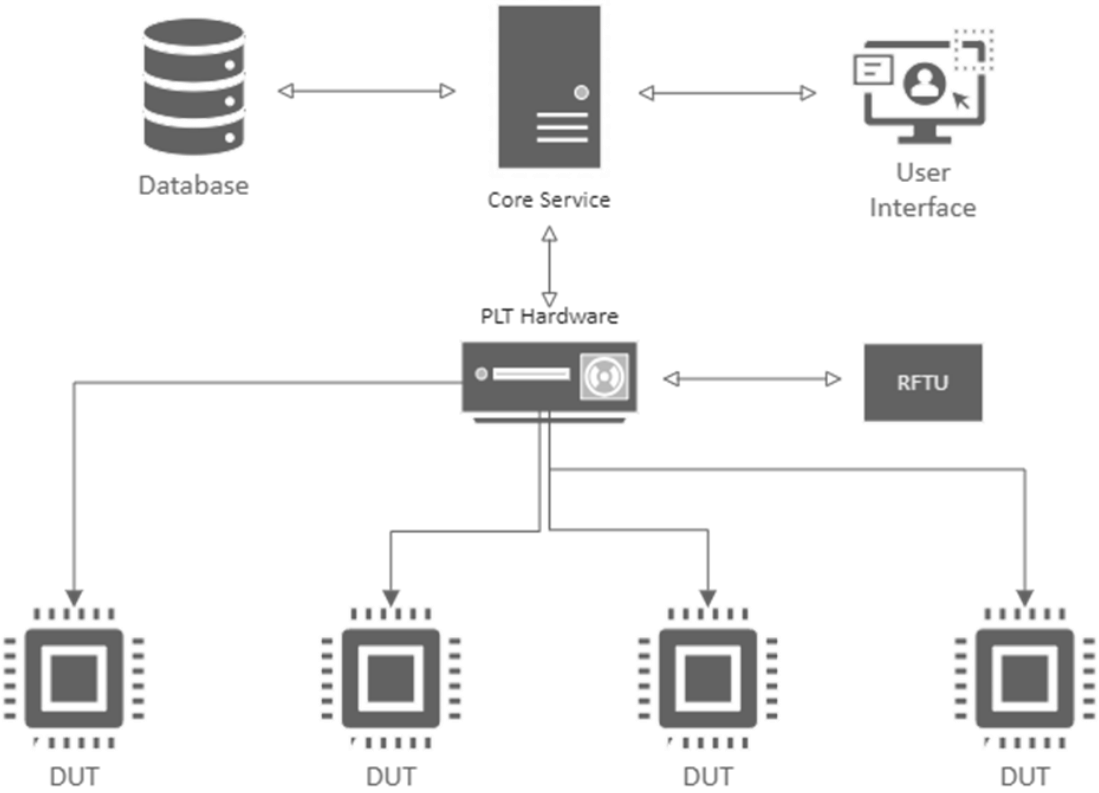



**Figure 16. Connectivity PLT architecture**

### 6.2.1 PLT Services

The Connectivity PLT installer can install two Windows services:

- **The Renesas Connectivity PLT Database Service** – A host service for the main database that holds all the production related data such as serial numbers, BD addresses, test logs, and results.
- **The Renesas Connectivity PLT Core Service** – The main service responsible for executing all operations at the host machine which has **a physical connection to the PLT hardware**.

These services can either be installed and run as Windows services or can be manually started to run in a shell window and can be installed on the same system or on separate networked machines.

### 6.2.2 PLT Applications

- **Connectivity PLT** – The main application which provides the user interface to the Connectivity PLT execution. This application can run on a separate but networked system to the core service or locally at the same system.
- **Connectivity PLT CLI** – A command line interface with similar functionality to the main application but without a graphical user interface. This application can run on a separate but networked system to the core service or locally at the same system.
- **Connectivity PLT Configuration** – An application providing a user interface to configure every aspect of the PLT operation, including setup, tests, and programming. This application needs to run on the same system as the core service.
- **Connectivity PLT Service Agent** – A tray application providing access to the Core Service log when it is running as a windows service as well as a simple way to start and stop the Windows service. This application needs to run on the same system as the core service.
- **Connectivity PLT Firmware Update** – An application providing an interface to update the Connectivity PLT firmware. This application needs to run on the same system that the hardware is connected to.

## 6.3 Installation

### 6.3.1 Standard Installation

To install all applications and services of the Connectivity PLT on a single host system, run the installer with default selections and with the **Install Connectivity PLT CoreService as a Windows Service** checkbox selected.

It is strongly recommended to keep the installation folder in the default location. However, if it is necessary to change it, select a folder that does not require special permissions, for example, user folders or application folders. The Windows services are executed from the SYSTEM account that does not have access to users' or some other special folders.

### 6.3.2 Remote System Installation

Installation of Core Service, Database Service, and Application can be installed on two or three separate remote systems.

- **CoreService System** – This should be installed in the system that is physically connected to the Connectivity PLT hardware. The service can be installed by enabling the Renesas Connectivity PLT Core Service Windows service checkbox in the installation wizard Figure 17. Take note of the IP address of the system, it is needed for the configuration of the Connectivity PLT application.

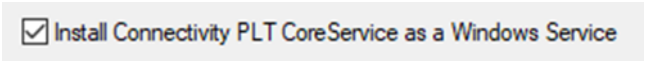



**Figure 17. CoreService Windows Service installation**

- **Database Service System** – To use another host only as the database service, complete the installation enabling the Renesas Connectivity PLT Database Service Windows service during the installation and disabling the CoreService installation (Figure 18). Remember the IP address of this host as well.
  For the CoreService to communicate with the database on a separate host, the configuration should be updated with the correct IP address or a host name as shown in Figure 19.

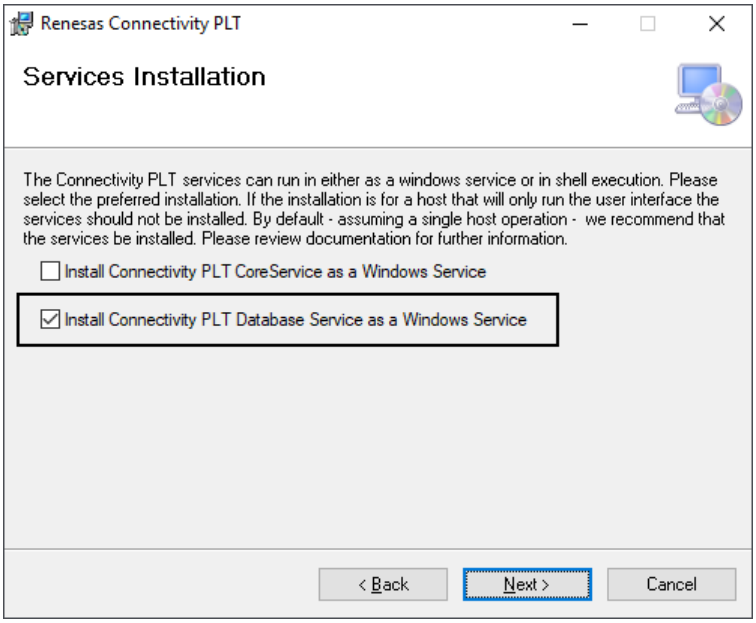



**Figure 18. Database Service installation**

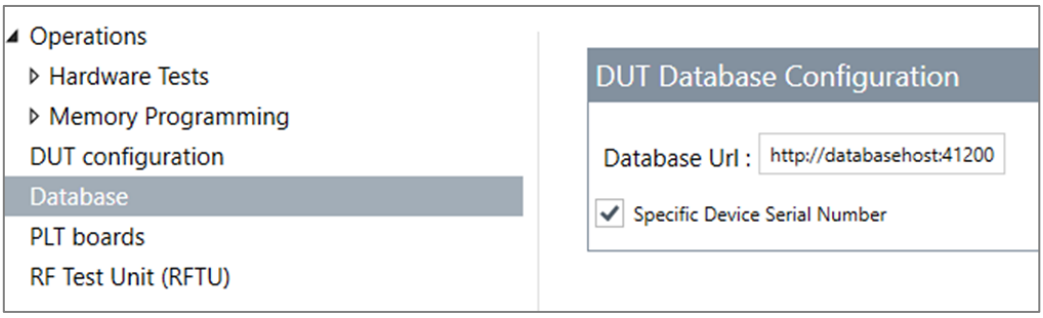



**Figure 19. Database Host configuration**

- **Connectivity PLT Application** (main application system) – The third system can be used as an operators' console which is, for example, located further away from the hardware. For this instance, during the installation, no services should be selected.

  To set up the system in a distributed architecture, the configuration needs to be updated of the main application to use the correct IP address of the CoreService. Go to the `C:\Renesas Electronics\Renesas Connectivity PLT\ Renesas_Connectivity_PLT.exe.config` file and update the `HostUrl` value to the correct IP address of the CoreService host as shown in Figure 20.

```
53   <appSettings>
54     <add key="HostUrl" value="127.0.0.1" />
55     <add key="Port" value="7000" />
56     <add key="RequestUri" value="/PltWebSocketService" />
57     <add key="ConnectionRetryLimit" value="3" />
58   </appSettings>
```

**Figure 20. Main Application CoreService Host configuration**

## 6.4 Install Necessary Drivers

Due to licensing restrictions, some drivers cannot be included in the installation package. You need to manually download and install these drivers.

- **Segger J-Link driver** – Can be installed as part of the Segger J-Link Application installation. You can download the driver from https://www.segger.com/downloads/jlink/.
  If the installer is launched from a restricted account, you need to run the driver installer manually as an administrator. The driver installer is in the installation folder of the JLink application. (Figure 21).

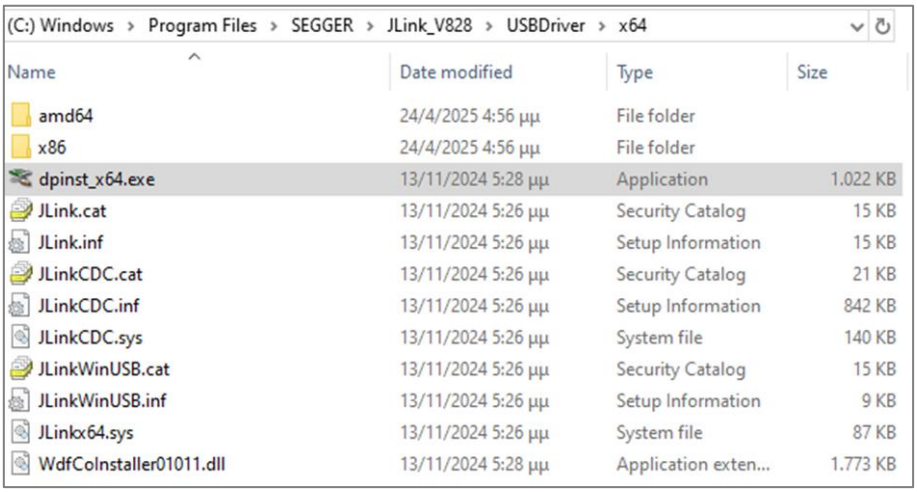



**Figure 21. Jlink driver installer**

- **NI Visa drivers** – If an external DMM or other instruments are used for tests, then NI Visa drivers are required. You can download the drivers from https://www.ni.com/en/support/downloads/drivers/download.ni-visa.html#558610.
- **Microchip SAM-BA driver** – Used for updating the firmware of the connectivity PLT hardware. You can download the driver from https://www.microchip.com/en-us/development-tool/sam-ba-in-system-programmer.

## 6.5 Update and Uninstall

To update to the newest version of the Connectivity PLT, first uninstall the previous version, and then install the needed version.

The pre-generated project files are not removed from the installation folder during the uninstall process. However, remember to keep a backup of the Configuration files, when new configuration files are introduced during the installation the old ones are going to be deleted. Figure 22 shows an example of configuration files and folders that should be backed up.

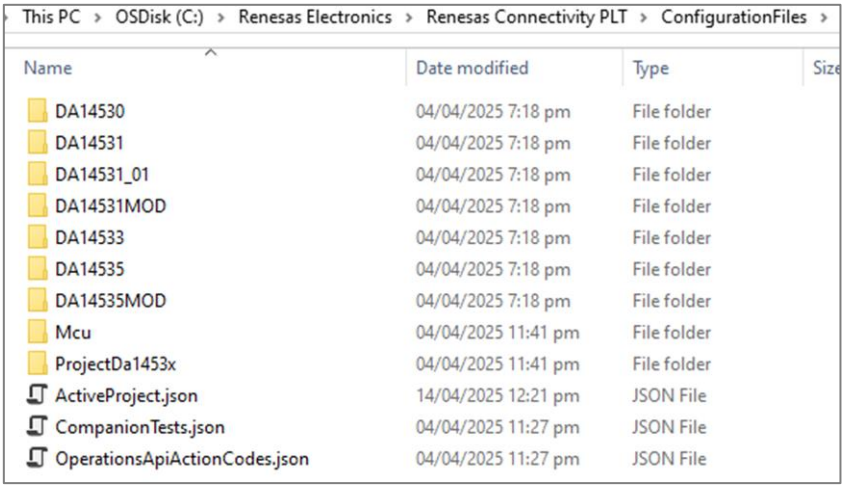



**Figure 22. Configuration files**

# 7. Database Service

The Renesas Connectivity PLT Database Service provides access to the database holding the product information as well as other details through a network interface (web service). It is used by the Connectivity PLT to identify serial numbers matched to BD addresses, store information about the test process and results, and retrieve data that is required to be written to the DUTs storage (eFlash, OTP, external Flash, EEprom).

## 7.1 Installation

Renesas Connectivity PLT Database Service can be installed using the same Connectivity PLT Installer.

The database service can be installed as a Windows service, using the same Connectivity PLT installer, and run in the background. Alternatively, it can be executed from a shell window with administrator privileges. It can be installed on the same host system as the one that executes the core service or on a remote system, if it is accessible over the network.

It is strongly recommended that it is installed as a service, in which instance no interaction with the user is required to access the database, it automatically starts during power up in the background with the correct privileges.

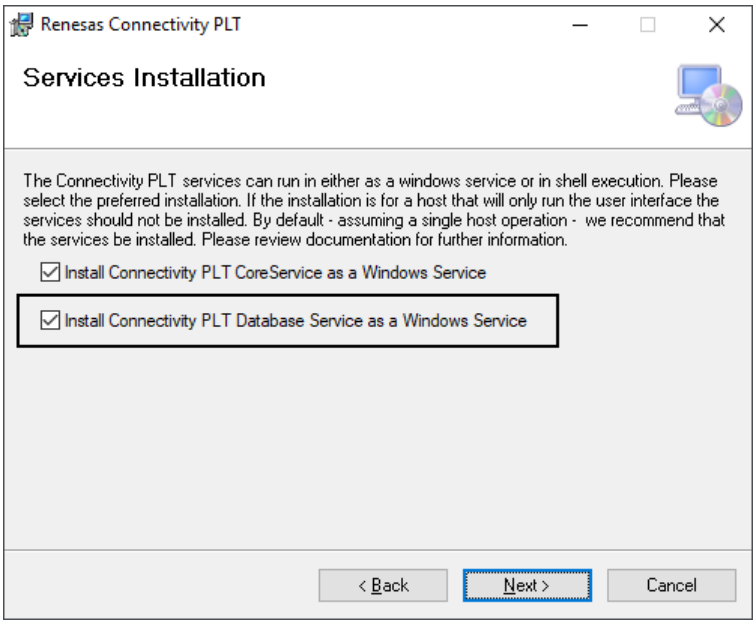



**Figure 23. Database Service Installation**

By default, the entire Connectivity PLT folder structure is installed on `C:\Renesas Electronics\Renesas Connectivity PLT`. The `database.db` file is in the `Database` folder in the root folder of the installation, and the database service is configured to locate the database at this location.

If you have changed the installation folder or want to move the database to another location, do the following:

1. Open the `DatabaseService\bin\Renesas_Connectivity_Plt_Database_Service.exe.config` file using a text editor.
2. Locate the following entry

   ```
   <connectionStrings>
       <add name="DevicesDatabase" providerName="System.Data.SQLite" connectionString="Data Source=C:\Renesas Electronics\Renesas Connectivity PLT\Database\DevicesDatabase.db; Integrated Security=True" />
   </connectionStrings>
   ```

   and then change the `DataSource` property to the correct location of the database file.
3. Save the file and restart the service.

## 7.2 Database Structure

The database contains a few different tables to hold all the necessary information.

### 7.2.1 Devices Table

The **Devices** table is the main table storing the list of serial numbers and corresponding BD addresses. After the installation, the database already contains some data for testing and demonstration purposes. Before going into production, you need to replace this data with the serial numbers and board addresses range based on your specific needs.

To manage the database directly, use SQLite tool (https://sqlitebrowser.org/).

Figure 24 shows the database **Devices** table structure in the SQLite tool.

| Devices | | CREATE TABLE "Devices" ( "SerialNumber" TEXT |
|---|---|---|
| SerialNumber | TEXT | "SerialNumber" TEXT NOT NULL UNIQUE |
| BluetoothLowEnergyAddress | INTEGER | "BluetoothLowEnergyAddress" INTEGER UNIQUE |
| BluetoothClassicAddress | INTEGER | "BluetoothClassicAddress" INTEGER UNIQUE |
| WiFiMacAddress | INTEGER | "WiFiMacAddress" INTEGER UNIQUE |
| AccessCount | INTEGER | "AccessCount" INTEGER NOT NULL |
| Timestamp | INTEGER | "Timestamp" INTEGER |
| Result | INTEGER | "Result" INTEGER NOT NULL |

**Figure 24. Database Devices table structure**

- Serial Number – Text field, must contain text representing the serial number. It must be unique, is the index.
- BluetoothLowEnergyAddress – Integer field, contains the entire Bluetooth® LE address as a 64-bit integer.
- BluetoothClassAddress – Integer field, contains the entire Bluetooth® Classic address as a 64-bit integer. Only used in devices that support Bluetooth® Classic.
- WiFiMacAddress – Integer field, contains the entire MAC address as a 64-bit integer. Only used in devices that support Wi-Fi.
- AccessCount – Integer field, contains a counter signifying how many times this entry is accessed by the Connectivity PLT.
- Timestamp – Integer field, a timestamp of the last access represented in Unix timestamp format.
- Result – Integer field, represents the results of the last test run on the Connectivity PLT.

### 7.2.2 Device Access Logs Table

The **DeviceAccessLogs** table holds information about every time the Connectivity PLT has accessed each device entry based on the serial number (index). This table should not be changed by any external processes or manually.

Figure 25 shows the Device Access Logs table structure in the SQLite tool.

| DeviceAccessLogs | | CREATE TABLE "DeviceAccessLogs" ( "Id" |
|---|---|---|
| Id | INTEGER | "Id" INTEGER NOT NULL UNIQUE |
| SerialNumber | TEXT | "SerialNumber" TEXT |
| Timestamp | INTEGER | "Timestamp" INTEGER |
| Action | INTEGER | "Action" INTEGER |
| Result | INTEGER | "Result" INTEGER |

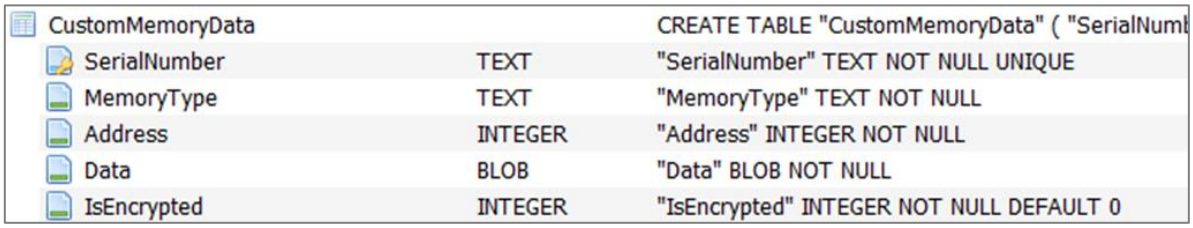
**Figure 25. Database Device Access Logs table structure**

- Id – Integer field, a unique ID of the entry (index).
- SerialNumber – Text field, Reference to the Devices Table of the entry.
- Timestamp – Integer field, a timestamp at the time of access represented in Unix timestamp format.
- Action – Integer field, an enumeration of the Connectivity PLT action log entry.
- Result – Integer field, the result of the action.

### 7.2.3 Custom Memory Data Table

The **CustomMemoryData** table contains data to be stored on the DUT tied to a specific serial number. This can be useful, for example, to store the serial number and other information that is unique to each device. For every continuous sequence of data a new entry should be made, multiple entries can exist for each serial number, each entry is programmed at the specified memory type and address specified. The write process is enabled during the configuration of the testing.

Figure 26 shows the Custom Memory Data table structure in the SQLite tool.



| CustomMemoryData | | CREATE TABLE "CustomMemoryData" ( "SerialNumb |
|---|---|---|
| SerialNumber | TEXT | "SerialNumber" TEXT NOT NULL UNIQUE |
| MemoryType | TEXT | "MemoryType" TEXT NOT NULL |
| Address | INTEGER | "Address" INTEGER NOT NULL |
| Data | BLOB | "Data" BLOB NOT NULL |
| IsEncrypted | INTEGER | "IsEncrypted" INTEGER NOT NULL DEFAULT 0 |

**Figure 26. Database Custom Memory data table**

- SerialNumber – Text, Reference to the Devices Table of the entry.
- Memory Type – Text, specifies the type of memory that the data should be stored at.
  - OqspiFlash
  - SpiFlash
  - Otp
- Address – Integer, the memory address at which the data should begin to be stored.
- Data – Blob, the data stored in binary format.
- IsEncrypted – Integer, specifies if the data entry is encrypted (1) or not (0).

Example of storing:

- Data of a random 32-bit number at address 0x10000 of the SPI Flash
- and the serial number at address 0x200000.

**Entry 1**

- SerialNumber: 0x12345678
- MemoryType: SpiFlash
- Address: 0x10000
- Data: [random number in binary blob]
- IsEncrypted: 0

**Entry 2**

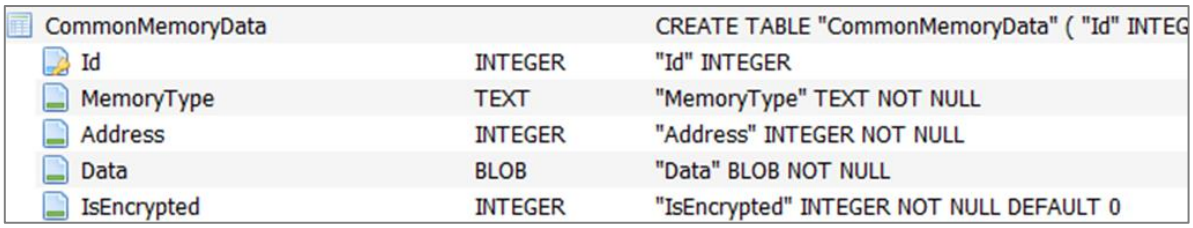
- SerialNumber: 0x12345678
- MemoryType: SpiFlash
- Address: 0x200000
- Data: [serial number in binary blob]
- IsEncrypted: 0

Another example of a different binary file loaded to the OTP for each serial number:

- SerialNumber: 0x12345678
- MemoryType: Otp
- Address: 0x1000
- Data: [0x12345678.bin]
- IsEncrypted: 0

### 7.2.4 Common Memory Data Table

The **CommonMemoryData** table contains data to be stored on every DUT. This can be useful for storing data for all the devices regardless of their serial number. For every continuous sequence of data a new entry should be made, each entry is programmed at the specified memory type and address specified. The write process is enabled during the configuration of the testing.



| CommonMemoryData | | CREATE TABLE "CommonMemoryData" ( "Id" INTEG |
|---|---|---|
| Id | INTEGER | "Id" INTEGER |
| MemoryType | TEXT | "MemoryType" TEXT NOT NULL |
| Address | INTEGER | "Address" INTEGER NOT NULL |
| Data | BLOB | "Data" BLOB NOT NULL |
| IsEncrypted | INTEGER | "IsEncrypted" INTEGER NOT NULL DEFAULT 0 |

**Figure 27. Database Common Memory Data table structure**

- Memory Type – Text, specifies the type of memory that the data should be stored at.
  - OqspiFlash
  - SpiFlash
  - OTP
- Address – Integer, the memory address at which the data should begin to be stored.
- Data – Blob, the data stored in binary format.
- IsEncrypted – Integer, specifies if the data entry is encrypted (1) or not (0).

Example of setting that is common on all devices stored in SPI flash:

- MemoryType: SpiFlash
- Address: 0x10000
- Data: [0x01 0x02 0x03 0x04]
- IsEncrypted: 0

## 7.3 Data Encryption

A basic encryption method is implemented for the data stored in the database. The data should be encrypted before storing it into the database.

The encryption method is AES on a 256-bit key.

The key for decryption is stored in `C:\Renesas Electronics\Renesas Connectivity PLT\DatabaseService\bin\ Renesas_Connectivity_Plt_Database_Service.exe.config`.

In the appSettings section, entry: `<add key="decryptionKey" value="1234567" />`

This should be changed to the key used for the encryption for the PLT to be able to decrypt and store the data.

# 8. Connectivity PLT Configuration

The Renesas Connectivity PLT Configuration is used to set up the system according to the device hardware options and select the required tests and memory actions to be performed.

The Connectivity PLT Configuration application (`Renesas_Connectivity_PLT_Configuration.exe`) is an application, which is used to appropriately configure the tests and memory operations the tool performs. Depending on the selected device chipset and the enabled actions, only appropriate options are enabled and shown. Any change made is validated before being saved to the JSON file. This prevents erroneous values from being stored in the JSON file that would harm the production procedure.

## 8.1 Connectivity PLT Configuration Window

The tree list in the side navigation is automatically created based on the folder structure in `ConfigurationFiles` folder in the application folder. Folders become parent items, and the contents (JSON configuration files) become child items. Clicking a tree item either expands the contents of the folder or displays the configuration parameters of the file. If there is an error in any of the configuration JSON files, a warning message is shown indicating which of the parameters has an error.

Figure 28 shows the initial Connectivity PLT Configuration window with the first configuration item in the tree menu on the left.

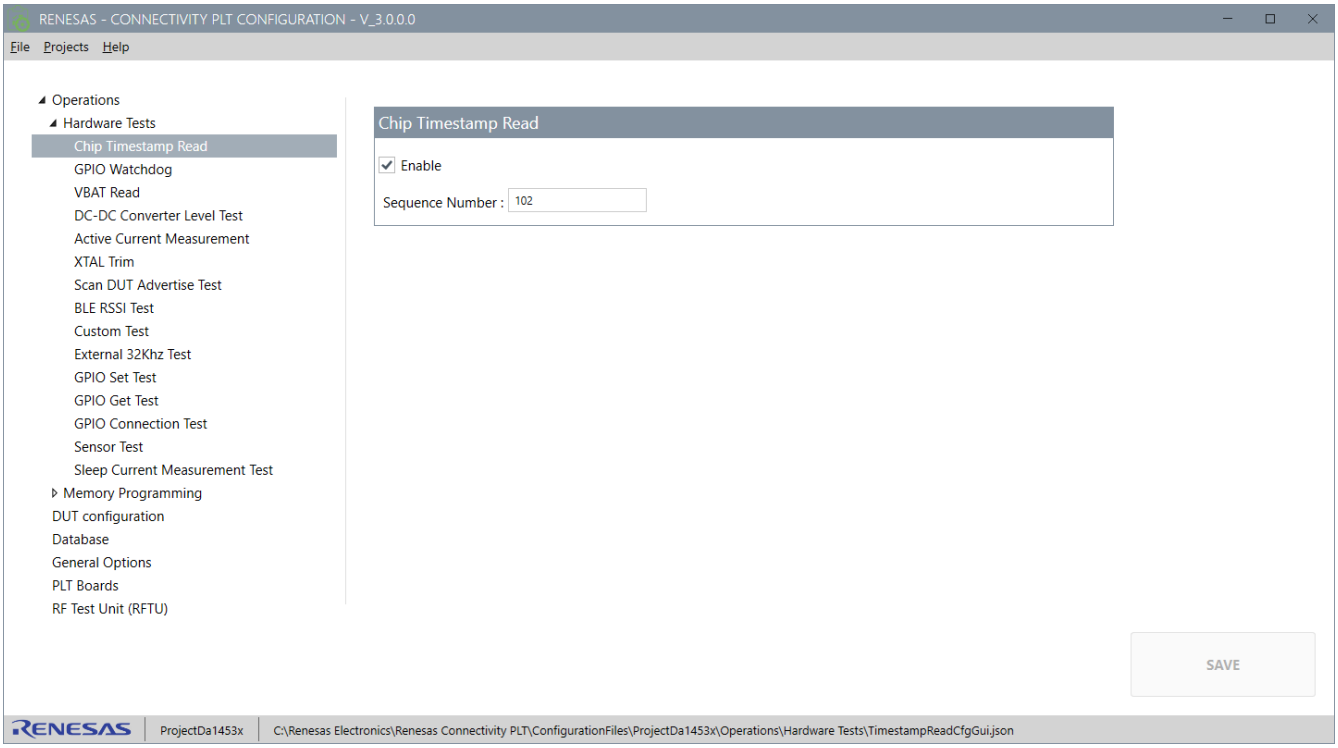



**Figure 28. Start window**

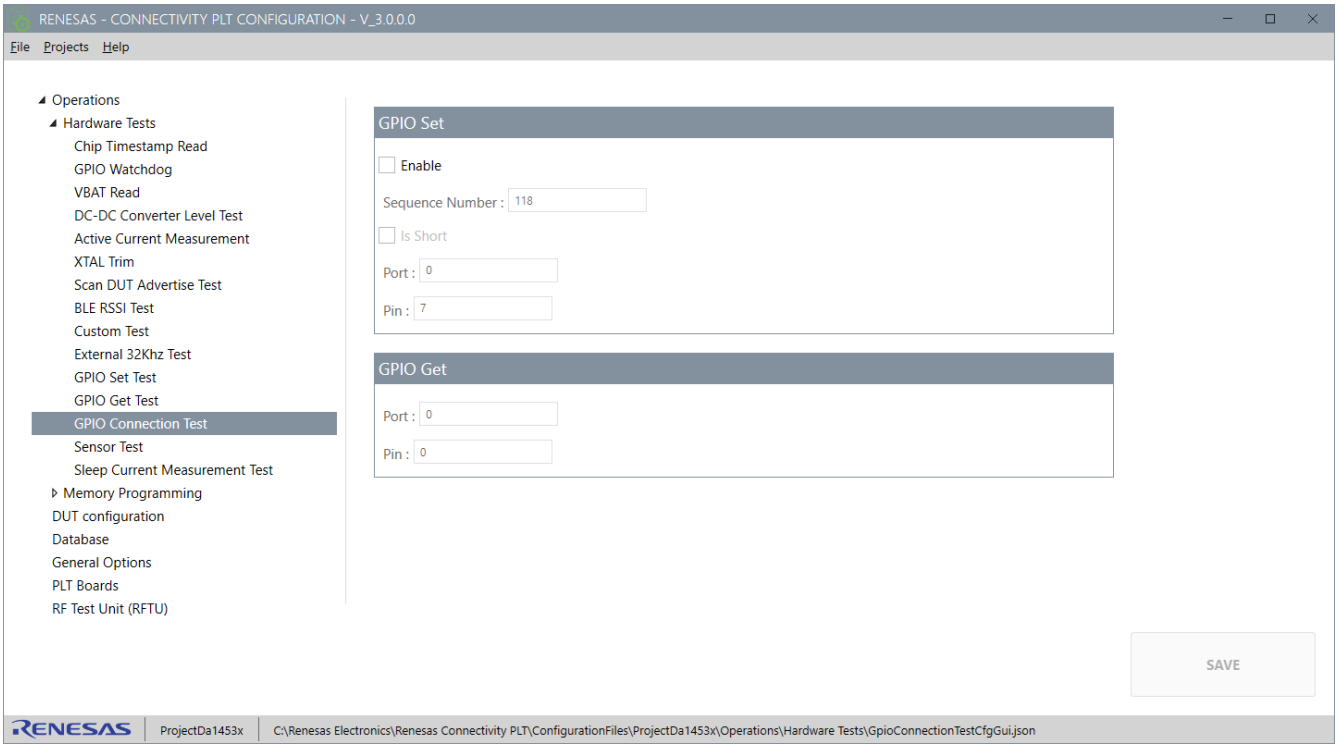



**Figure 29. Tree menu**

You can use menu options to manage projects: create new ones, open, save, reload, or edit.

At the bottom of the window, you can see the location path of the current selected JSON configuration file.

The Save button is disabled on startup. After you make a change in any field in JSON configuration files, the Save button turns red with an asterisk and the option to save the changes is available. An asterisk also appears on title bar and on Windows taskbar. You must save your changes before opening another file.

The parameter fields have a tooltip describing the field and the expected values.

All the parameter fields are protected by regex validation customized to the contents of each field. If an unsupported value or character is inserted, the field turns red and the Save button is disabled again until a valid value is entered, Figure 30.

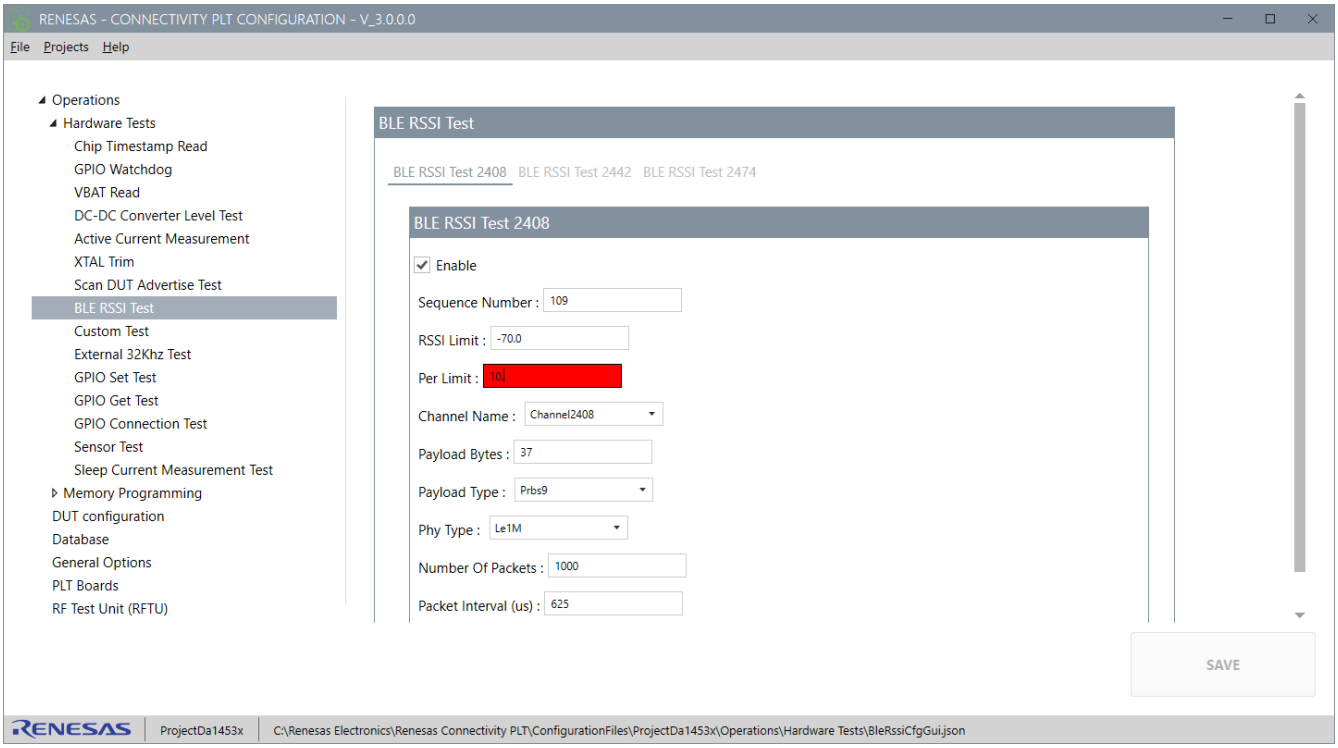



**Figure 30. Input error**

The Help menu provides access to the product documentation. When the User Manual option is selected, the PDF User Manual opens in the system's default PDF viewer.

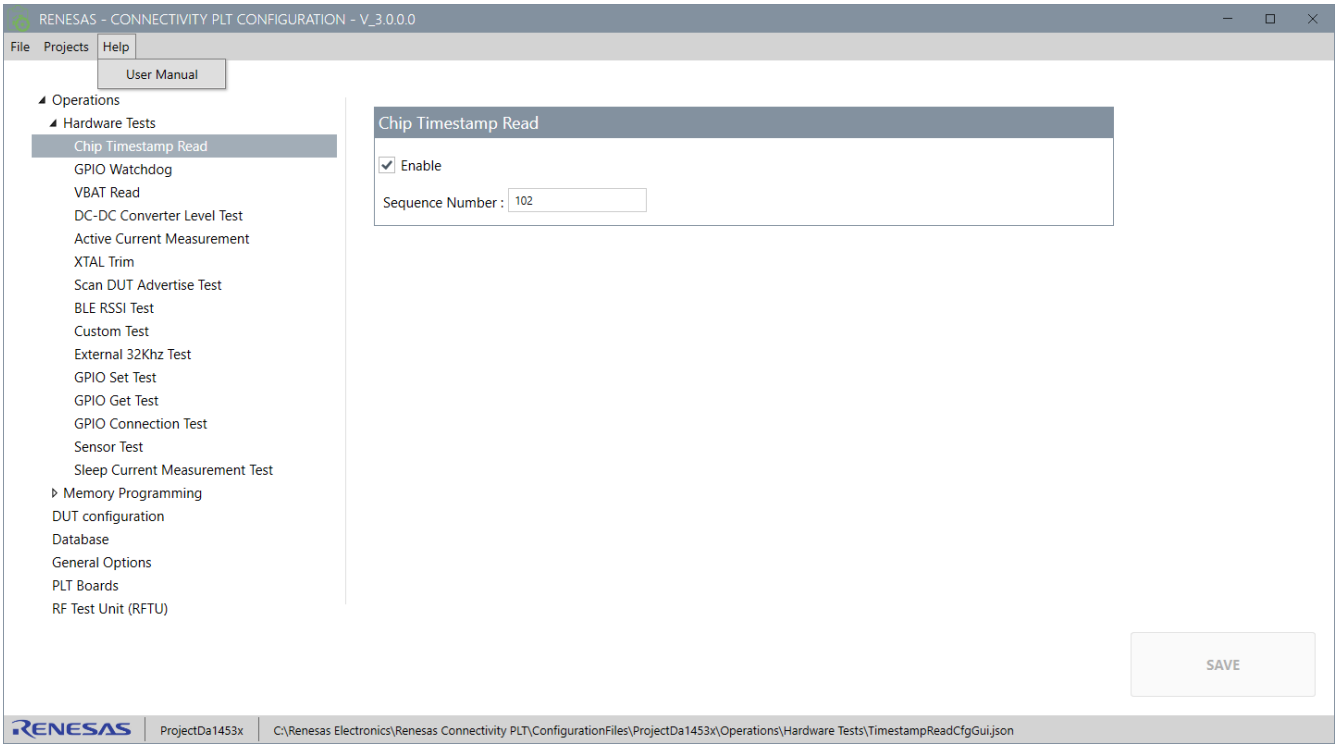



**Figure 31. Help menu**

## 8.2 Configurations

### 8.2.1 Project Configuration

You can save information into projects to easily switch between different test procedures or devices in the Connectivity PLT. Each project is a set of configuration files stored in a separate folder named as the project. By default, when Connectivity PLT is installed, a few projects representing different ICs are created automatically.

To create a new project configuration:

1. Go to **File** > **Create New Project**.
2. In the **Create New Project** dialog, select a base project that is used as a starting point, and add project name.

All the necessary files are copied to the new project folder under `ConfigurationFiles` folder and side navigation refreshes displaying the new project.

### 8.2.2 Database Configuration

Database configuration is used to store serial numbers, Bluetooth® LE addresses, Bluetooth® Classic addresses, Wi-Fi MAC addresses; information used in testing, such as common memory data and custom memory data, and statistics like passed/failed tests, access count, and timestamps. If the Connectivity PLT Database Service is installed on a separate machine, you need to set the correct URL and port to point to this machine.

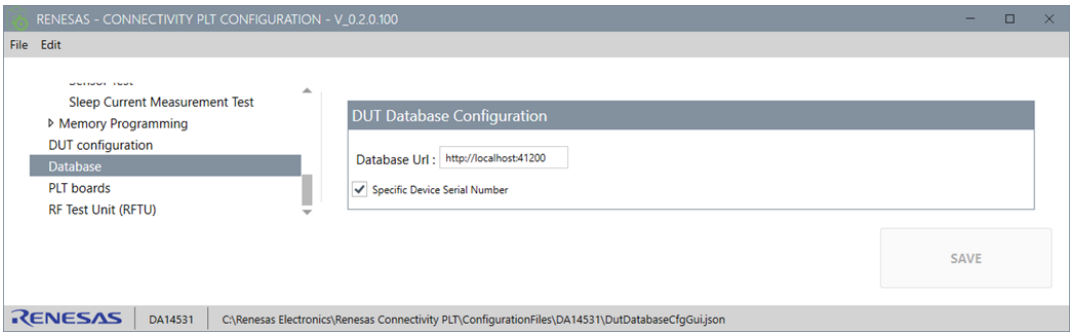



**Figure 32. Database configuration**

The Connectivity PLT used the database to assign BD addresses based on serial numbers previously entered in the database. The option to have Specific Device Serial Number enables a dialog to appear at the beginning of the test procedure to enter the serial number of the first DUT. This is usually done using a barcode scanner. For more details, see Section 7.

### 8.2.3 PLT Boards Configuration

The Connectivity PLT boards configuration has four available Connectivity PLT boards, and you can enable them and assign a location in the chain, see Figure 33. For each Connectivity PLT board, there are eight available test positions. For the minimum configuration, you must enable at least one Connectivity PLT board and one DUT.

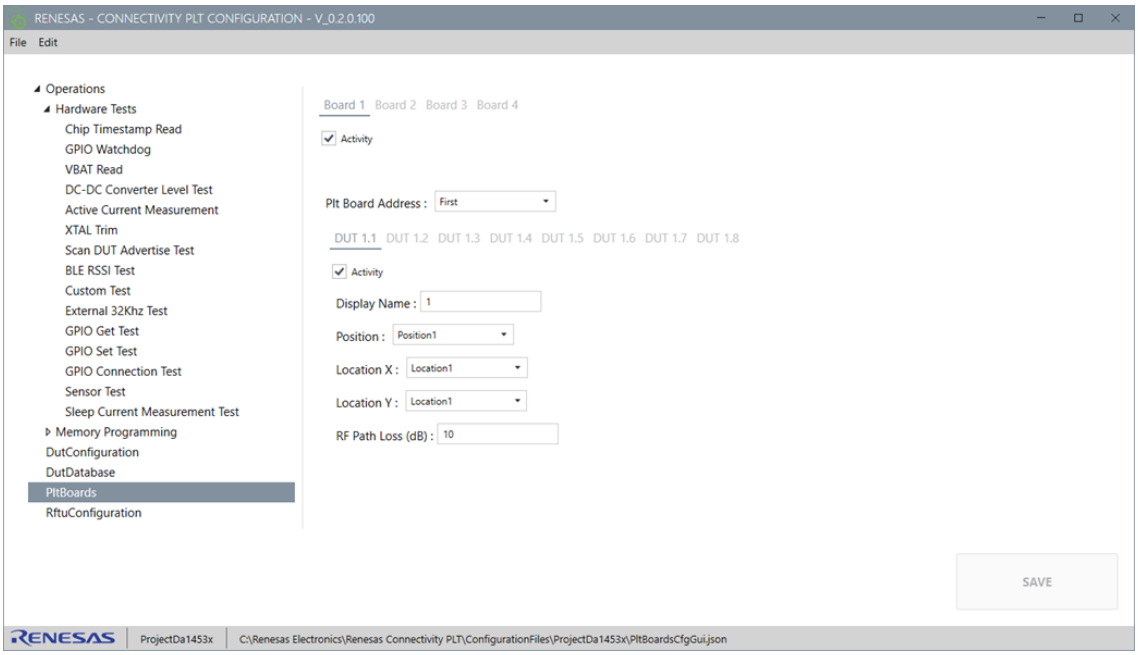



**Figure 33. PLT Boards configuration**

**Table 4. PLT Boards parameters**

| Parameter | Description |
|---|---|
| **PLT Board** | |
| Activity | Enables or disables the usage of the specific PLT Board |
| PLT Board Address | Address of the PLT Board. Available options: First, Second, Third, Fourth. |
| **DUT** | |
| Activity | Enables or disables the usage of the specific DUT |
| Display Name | Name of each DUT in the testing environment |
| Position | Position of each DUT on the PLT Board |
| Location X | Location of each DUT, in X axis, in the testing environment grid |
| Location Y | Location of each DUT, in Y axis, in the testing environment grid |
| RF Path Loss (db) | Value to be used in RF Path Loss, in dBm |

Location X/Y adjusts the view of the DUT tiles in the testing environment to simulate the actual hardware layout. Each DUT can be placed wherever needed in the Execution GUI grid. Figure 34 and Figure 35 show examples of two different views on the testing environment by adjusting Position X and Position Y.

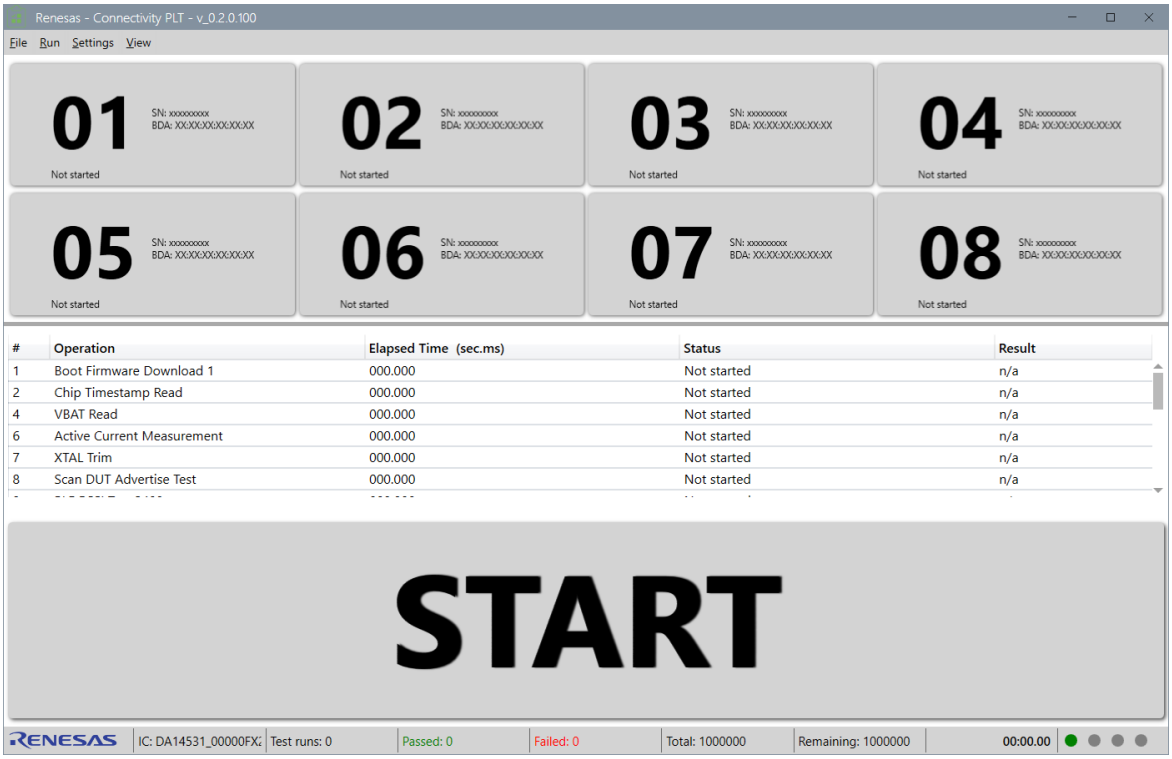



**Figure 34. Default tile configuration**

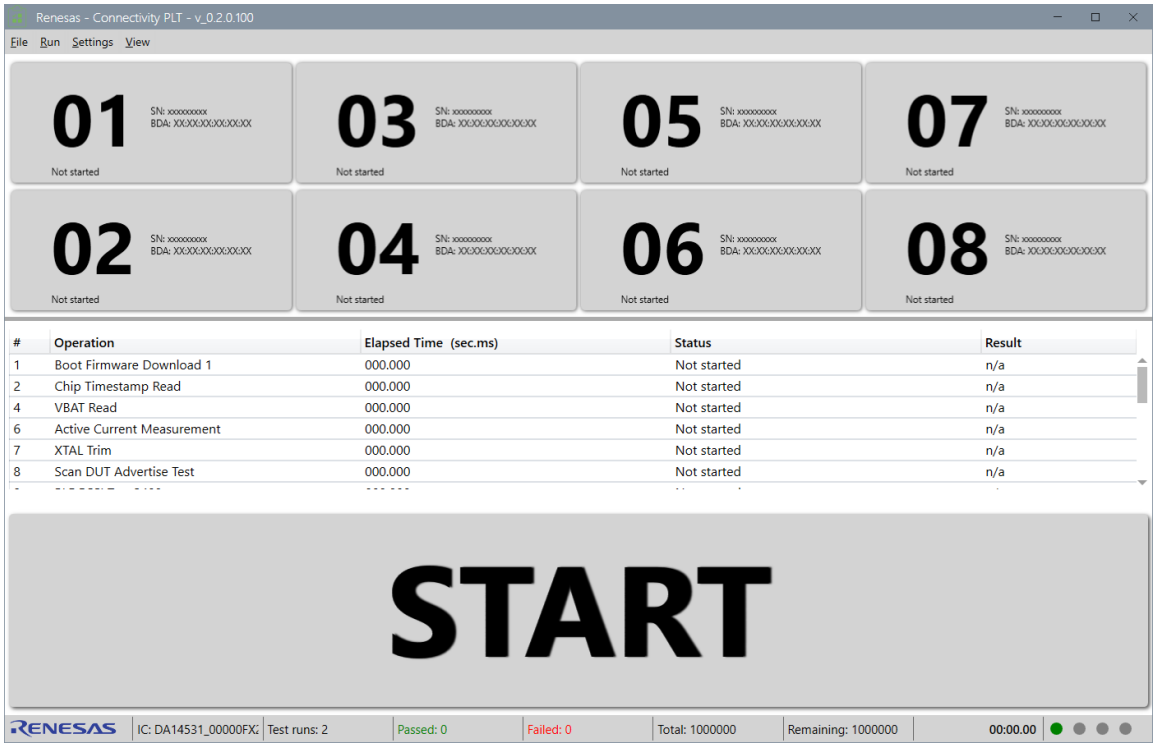



**Figure 35. 2-row tile configuration**

### 8.2.4 RFTU Bluetooth® LE Configuration

The Connectivity PLT system can be configured to use one or multiple RFTUs. Multiple RTFUs can be used for cases when the multiple boards exist on the chain and are separated by distance or other RF blocking methods. Different RFTU configurations are available. At a minimum one RFTU is necessary when RF tests are going to be performed during the test process. Figure 36 demonstrates the default and minimum RFTU configuration. The communications Interface name, in order to be automatically detected, should be set to empty.

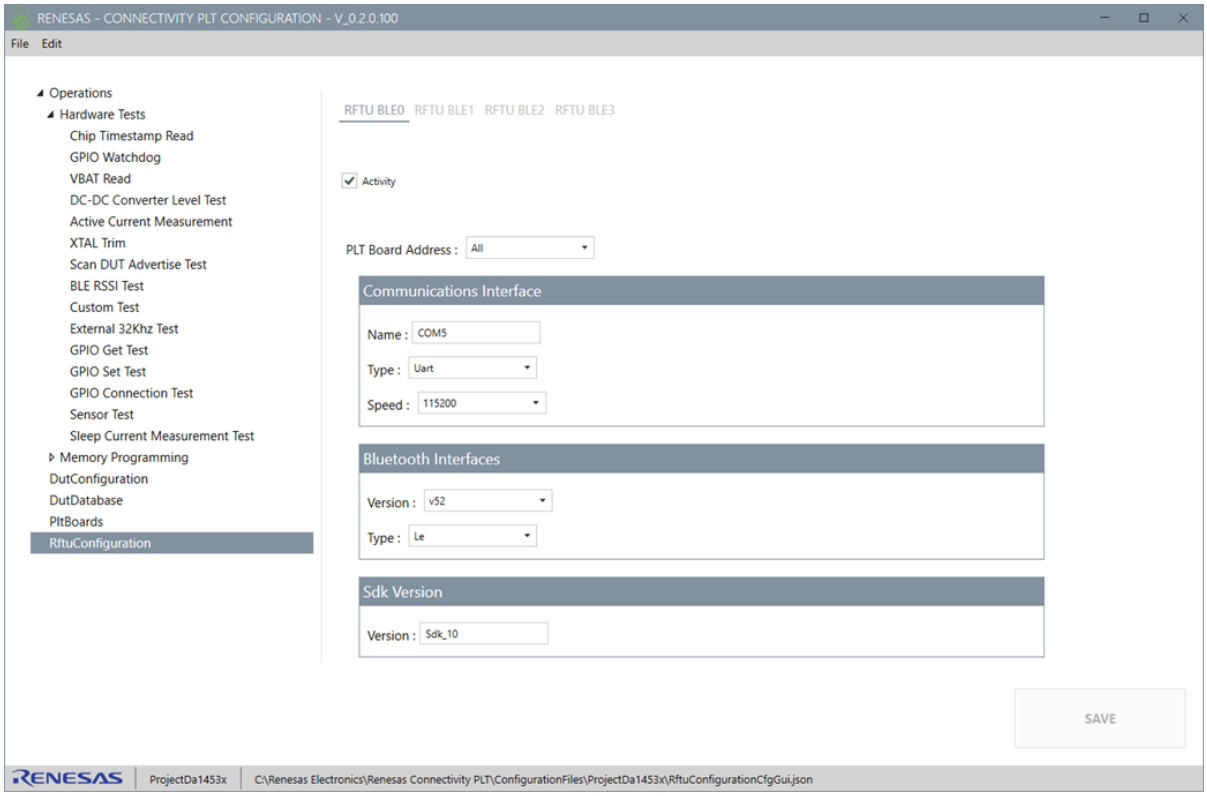



**Figure 36. RFTU Bluetooth® LE configuration**

**Table 5. RFTU Bluetooth® LE Configuration parameters**

| Parameter | Description |
|---|---|
| **RFTU BLE** | |
| Activity | Enables or disables the usage of the specific RFTU |
| PLT Board Address | All if the specific RFTU is used for all PLT boards.<br>Otherwise, the address (through the DIP switches) of the associated PLT board to be tested with the specific RFTU (first, second, third, fourth). |
| **Communications Interface** | |
| Name | Assigns a name to the communications interface |
| Type | Defines the specific interface. Available options: UART, SingleWireUART. |
| Speed | Defines speed of the communication interface, for example, 115200 |
| **Bluetooth Interfaces** | |
| Version | Version of the Bluetooth® LE interface. Available options: v40, v41, v42, v50, v51, v52, v53. |
| Type | Type of the Bluetooth® LE interface. Available options: Classic, LE. |
| **SDK Version** | |
| Version | Version of the SDK used |

### 8.2.5 RFTU Wi-Fi Configuration

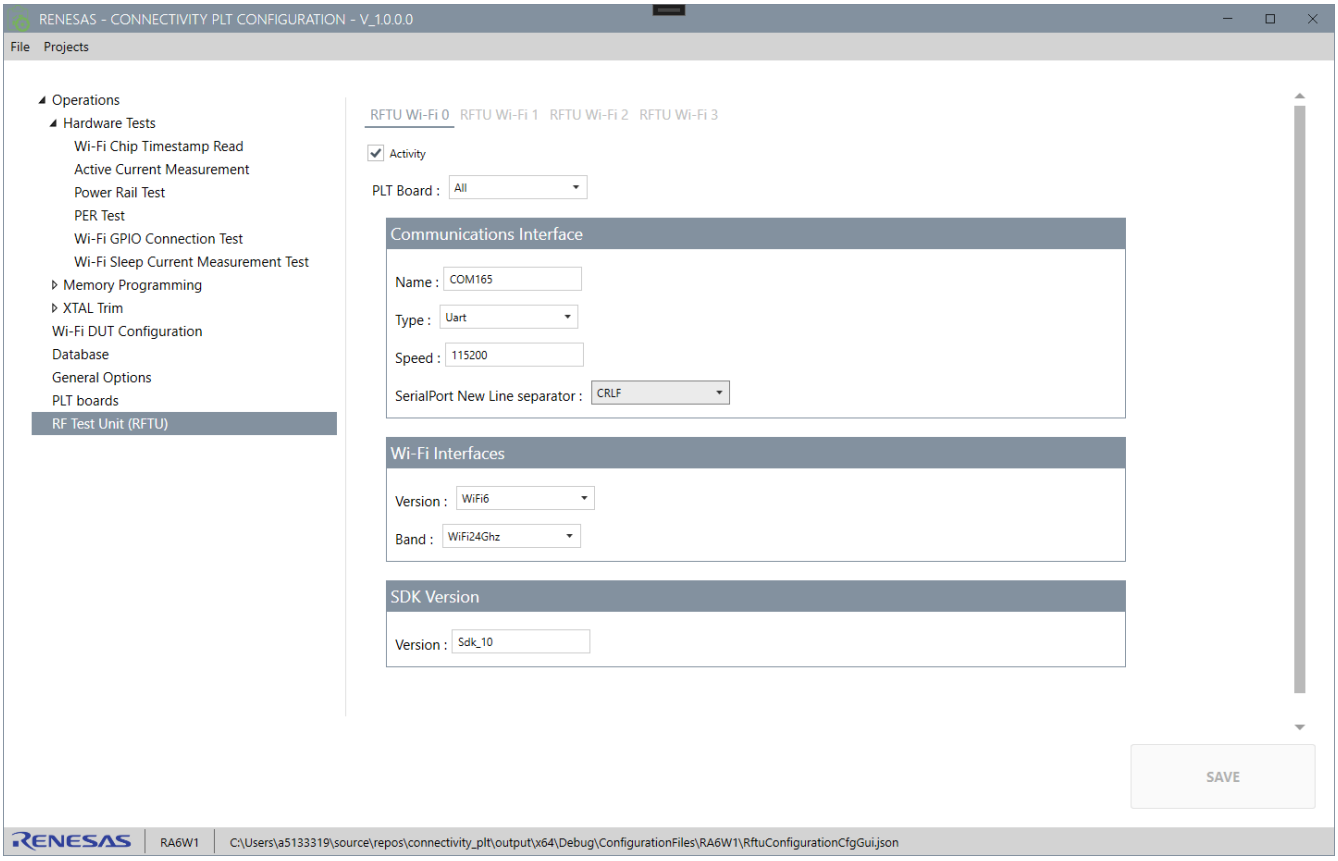



**Figure 37. RFTU Wi-Fi configuration**

**Table 6. RFTU Wi-Fi parameters**

| Parameter | Description |
|---|---|
| **RFTU Wi-Fi** | |
| Activity | Enables or disables the usage of the specific RFTU |
| PLT Board | All if the specific RFTU is used for all PLT boards.<br>Otherwise, the address (through the DIP switches) of the associated PLT board to be tested with the specific RFTU (first, second, third, fourth). |
| **Communications Interface** | |
| Name | Assigns a name to the communications interface |
| Type | Defines the specific interface. Available options: UART, SingleWireUART. |
| Speed | Defines speed of the communication interface, for example, 115200 |
| Serial Port New Line separator | Available options: CRLF, CF, LF |
| **Wi-Fi Interfaces** | |
| Version | Version of the Wi-Fi interface. Available options: WiFi1, WiFi2, WiFi3, WiFi4, WiFi5, WiFi6. |
| Band | Band of the Wi-Fi interface. Available options: WiFi24Ghz, WiFi5Ghz, WiFiAllBands. |
| **SDK Version** | |
| Version | Version of the SDK used |

### 8.2.6 RFTU Bluetooth® LE Wi-Fi Configuration

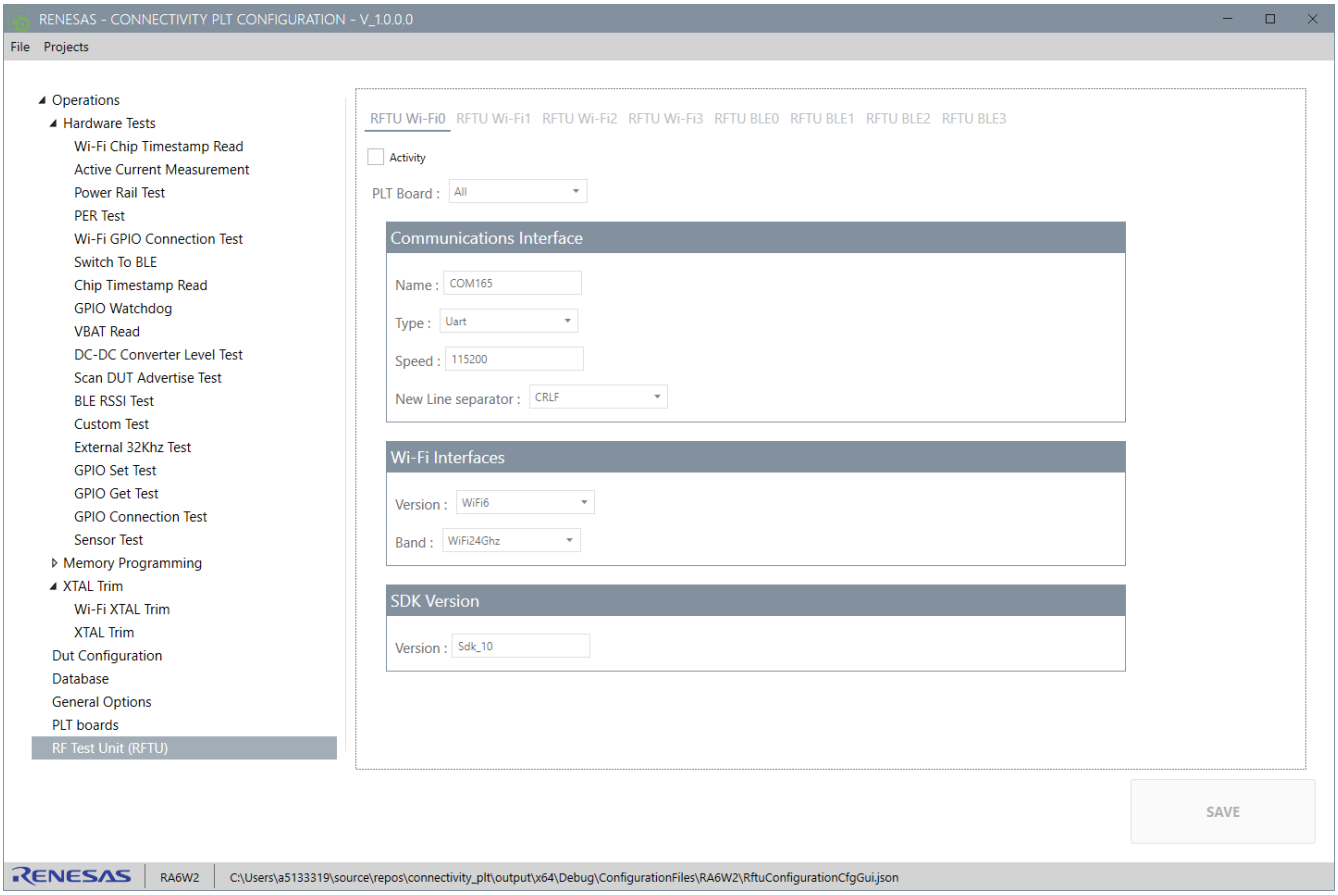



**Figure 38. Bluetooth LE® Wi-Fi RFTU configuration**

**Table 7. Bluetooth LE® Wi-Fi RFTU parameters**

| Parameter | Description |
|---|---|
| **RFTU Wi-Fi** | |
| Activity | Enables or disables the usage of the specific RFTU |
| PLT Board | All if the specific RFTU is used for all PLT boards.<br>Otherwise, the address (through the DIP switches) of the associated PLT board to be tested with the specific RFTU (first, second, third, fourth). |
| **Communications Interface** | |
| Name | Assigns a name to the communications interface |
| Type | Defines the specific interface. Available options: UART, SingleWireUART. |
| Speed | Defines speed of the communication interface, for example 115200 |
| New Line Separator | Available options: CRLF, CF, LF |
| **Wi-Fi Interfaces** | |
| Version | Version of the Wi-Fi interface. Available options: WiFi1, WiFi2, WiFi3, WiFi4, WiFi5, WiFi6. |
| Band | Band of the Wi-Fi interface. Available options: WiFi24Ghz, WiFi5Ghz, WiFiAllBands. |
| **SDK Version** | |
| Version | Version of the SDK used |
| **RFTU BLE** | |

| Parameter | Description |
|---|---|
| Activity | Enables or disables the usage of the specific RFTU |
| PLT Board Address | All if the specific RFTU is used for all PLT boards.<br>Otherwise, the address (through the DIP switches) of the associated PLT board to be tested with the specific RFTU (first, second, third, fourth). |
| **Communications Interface** | |
| Name | Assigns a name to the communications interface |
| Type | Defines the specific interface. Available options: UART, SingleWireUART. |
| Speed | Defines speed of the communication interface, for example, 115200 |
| **Bluetooth Interfaces** | |
| Version | Version of the Bluetooth® LE interface. Available options: v40, v41, v42, v50, v51, v52, v53. |
| Type | Type of the Bluetooth® LE interface. Available options: Classic, LE. |
| **SDK Version** | |
| Version | Version of the SDK used |

## 8.2.7 DUT Configuration

The DUT Configuration section is dedicated to parameters specific to the actual devices that are under test. There are numerous parameters to describe the hardware properties, communication interfaces, and so on.

### 8.2.7.1 DUT Configuration – Name

The Device IC under Dut Name is the generic type of the IC used.

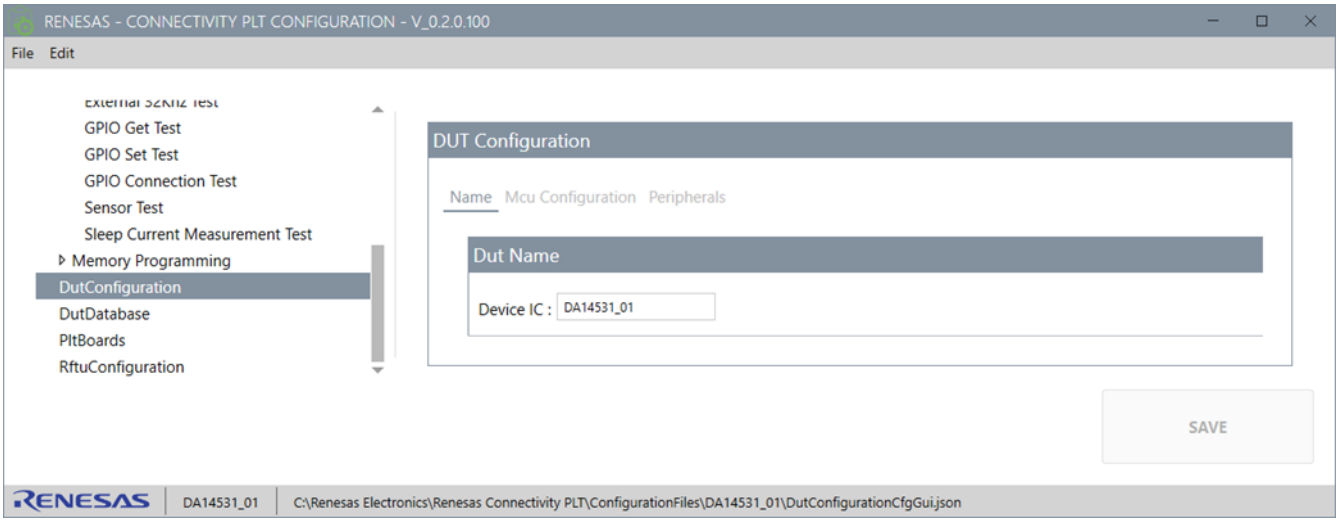



**Figure 39. DUT Configuration**

#### 8.2.7.2 DUT Configuration – MCU Configuration

The MCU configuration section sets all the parameters specific to the MCU being used by the DUT.

- MCU Name – Exact part number of the MCU used for the DUT.

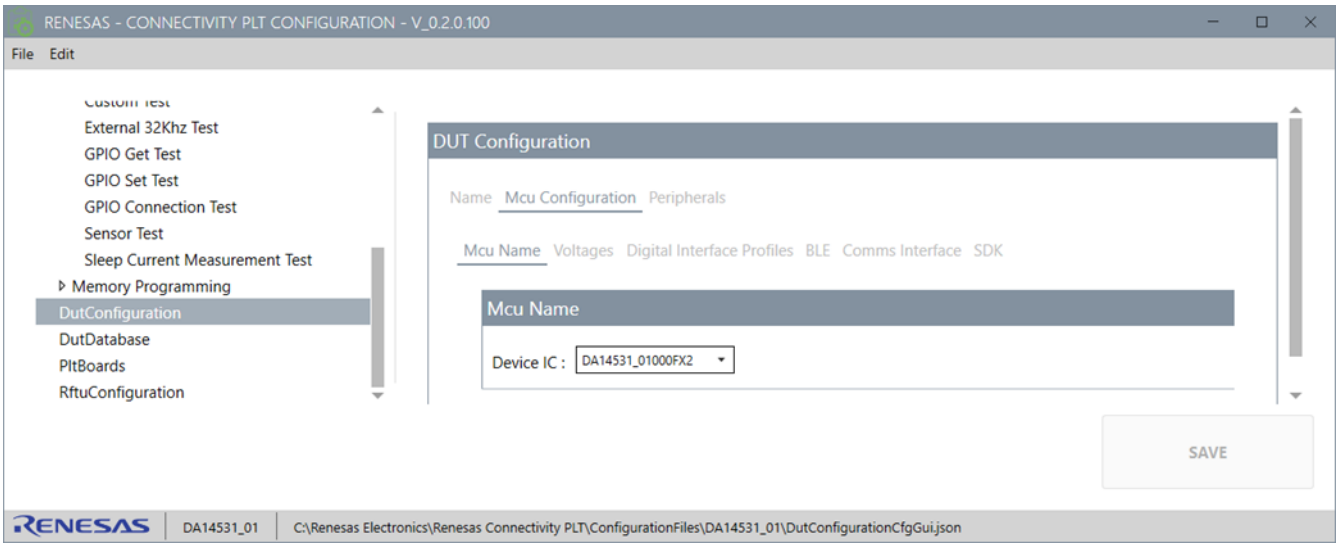



**Figure 40. MCU Name configuration**

- Voltages:
  - Vbat – Voltage that the Connectivity PLT board supplies to the DUT.
  - Vddio – Voltage of the digital interface.

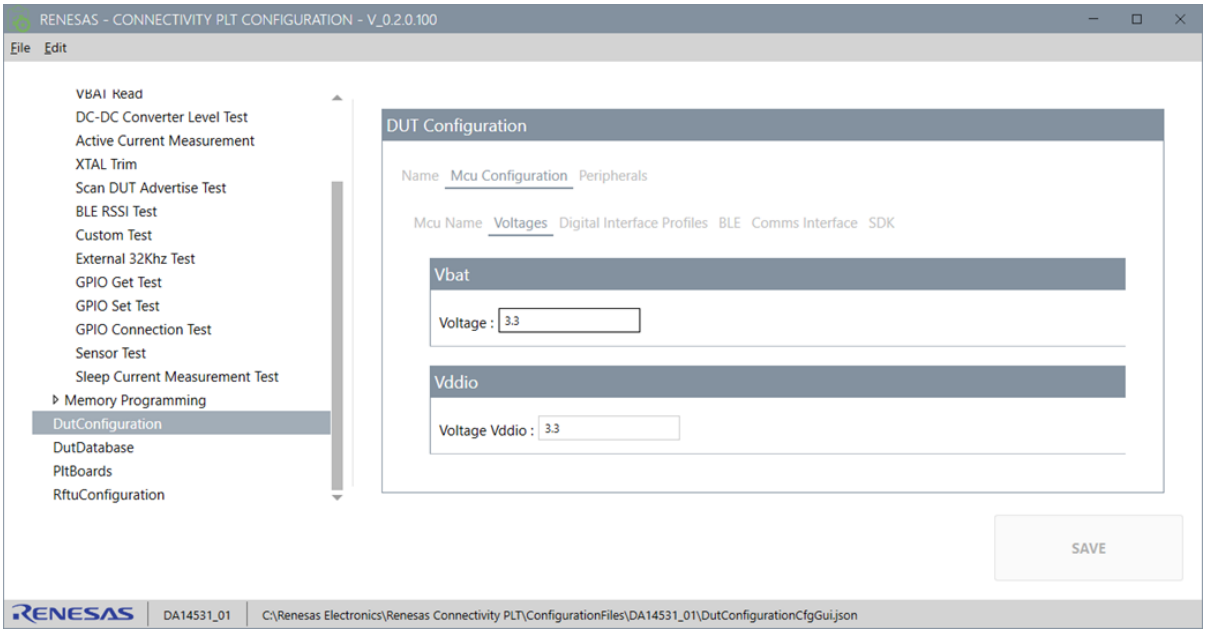



**Figure 41. Voltages configuration**

- Digital Interface Profiles:
  - **SPI Profile** – Groups a set of parameters for the SPI interface commonly used for flash memory and sensors. This is a primary configuration of the interface that is used if no further profile is defined for the specific peripheral.

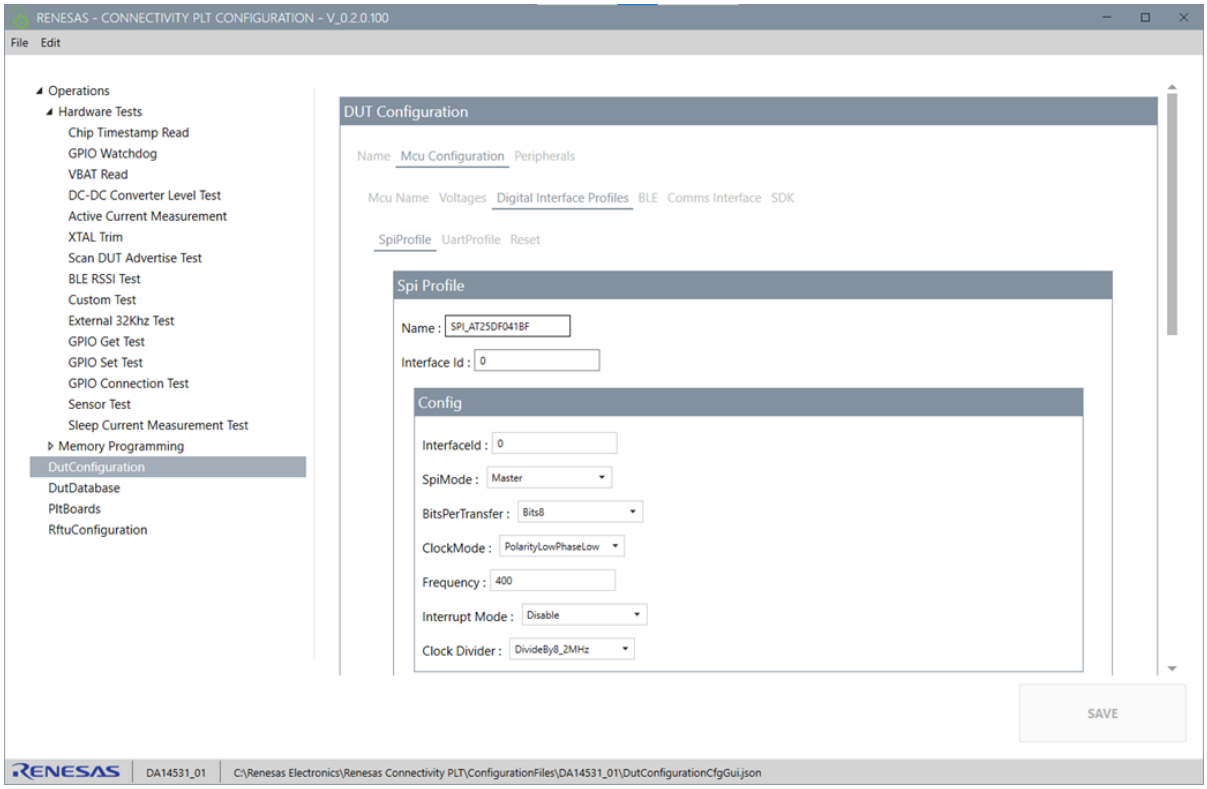



**Figure 42. SPI Profile configuration**

**Table 8. SPI Profile parameters**

| Parameter | Description |
|---|---|
| Name | SPI Profile name |
| Interface ID | ID of the SPI Profile interface |
| **Config** | |
| Interface ID | ID of the interface used |
| SPI Mode | SPI operating mode (Master/ Slave) |
| Bits Per Transfer | Available options: Bits8, Bits16 |
| Clock Mode | Available options: PolarityLowPhaseLow, PolarityLowPhaseHigh, PolarityHighPhaseLow, PolarityHighPhaseHigh |
| Frequency | SPI bus frequency, in KHz. For example, 400 |
| Interrupt Mode | Available options: Enable, Disable |
| Clock Divider | Available options: DivideBy8_2MHz, DivideBy4_4MHz, DivideBy2_8MHz, DivideBy14_16MHz |

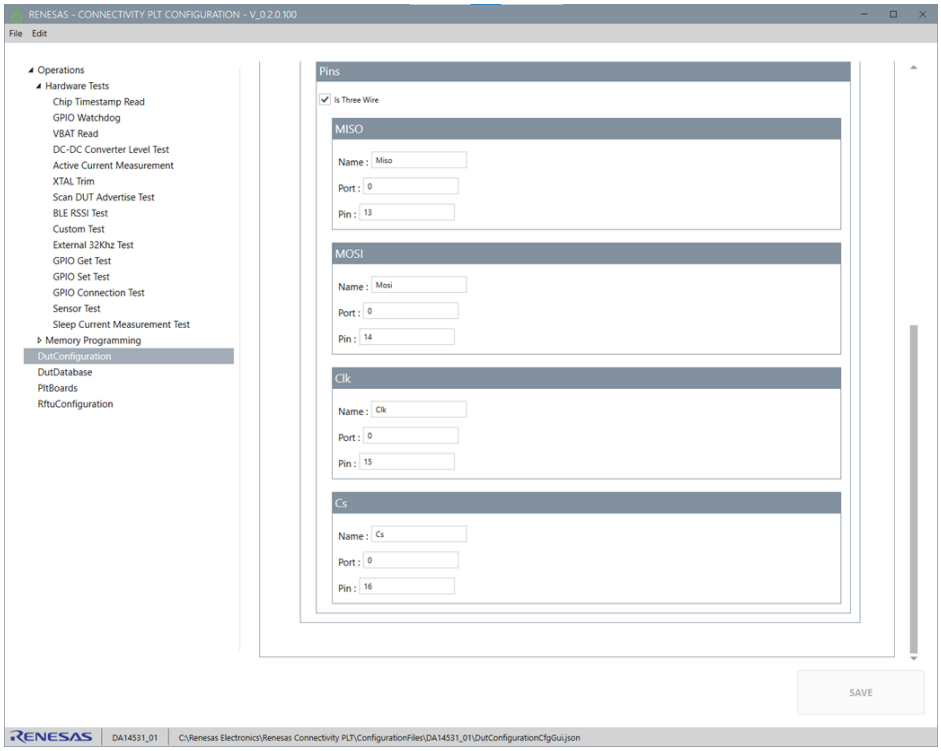



**Figure 43. Pins configuration**

**Table 9. SPI profile Pins parameters**

| Option | Description |
|---|---|
| **Pins** | |
| Is Three Wire | Enables 3-wire SPI communication |
| **MISO** | |
| Port | MISO GPIO port |
| Pin | MISO GPIO pin |
| **MOSI** | |
| Port | MOSI GPIO port |
| Pin | MOSI GPIO pin |
| **SCLK** | |
| Port | SCLK GPIO port |
| Pin | SCLK GPIO pin |
| **CS** | |
| Port | CS GPIO port |
| Pin | CS GPIO pin |

- **UART Profile**

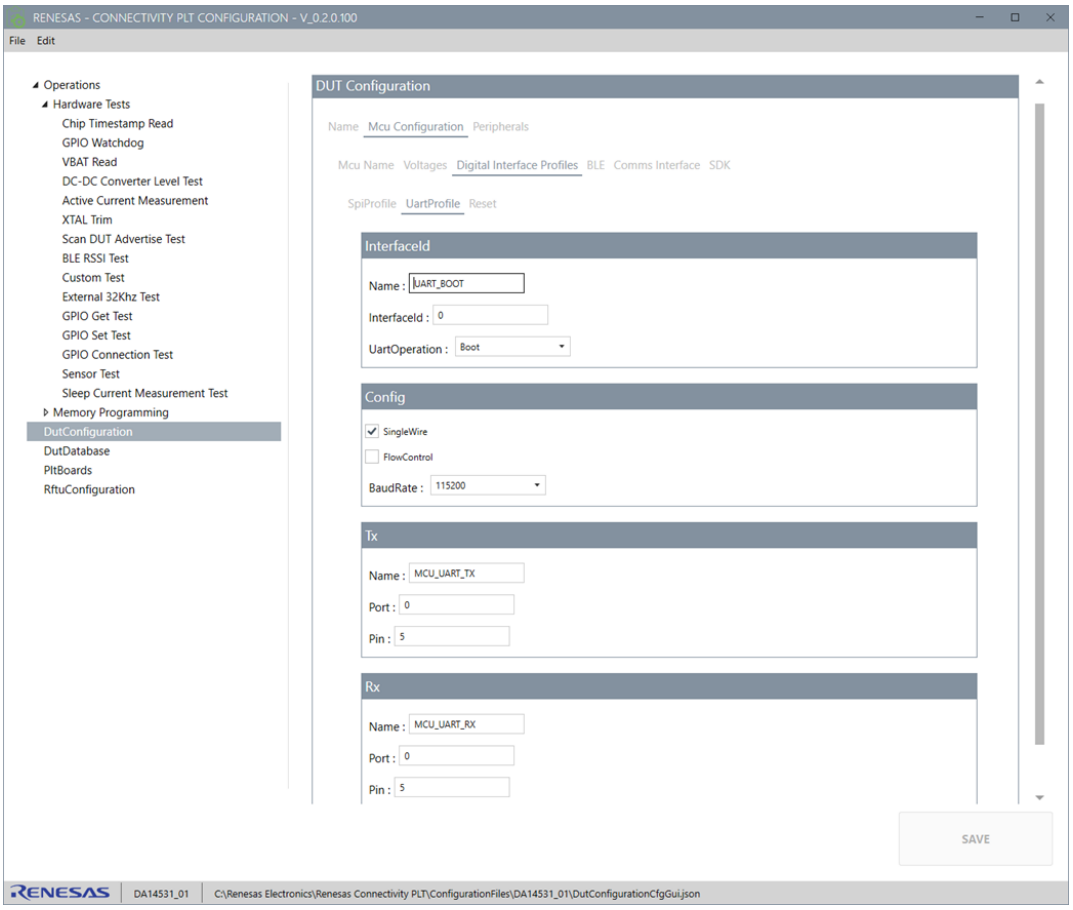



**Figure 44. UART Profile configuration**

**Table 10. UART Profile parameters**

| Parameter | Description |
|---|---|
| **Interface ID** | |
| Name | Name of the interface |
| Interface ID | ID of the interface used |
| UART Operation | Available options: Boot, Peripheral, Debug |
| **Config** | |
| Single Wire | - |
| Flow Control | - |
| Baud Rate | Available options: 1200, 2400, 4800, 9600,19200, 38400, 57600, 76800, 115200, 230400, 460800, 576000, 921600, 1M, 2M, 3M, 4M, 5M, 6M |
| **Tx** | |
| Name | MCU_UART_TX |
| Port | Tx GPIO port |
| Pin | Tx GPIO pin |
| **Rx** | |
| Name | MCU_UART_RX |
| Port | Rx GPIO port |
| Pin | Rx GPIO pin |

- **Reset**

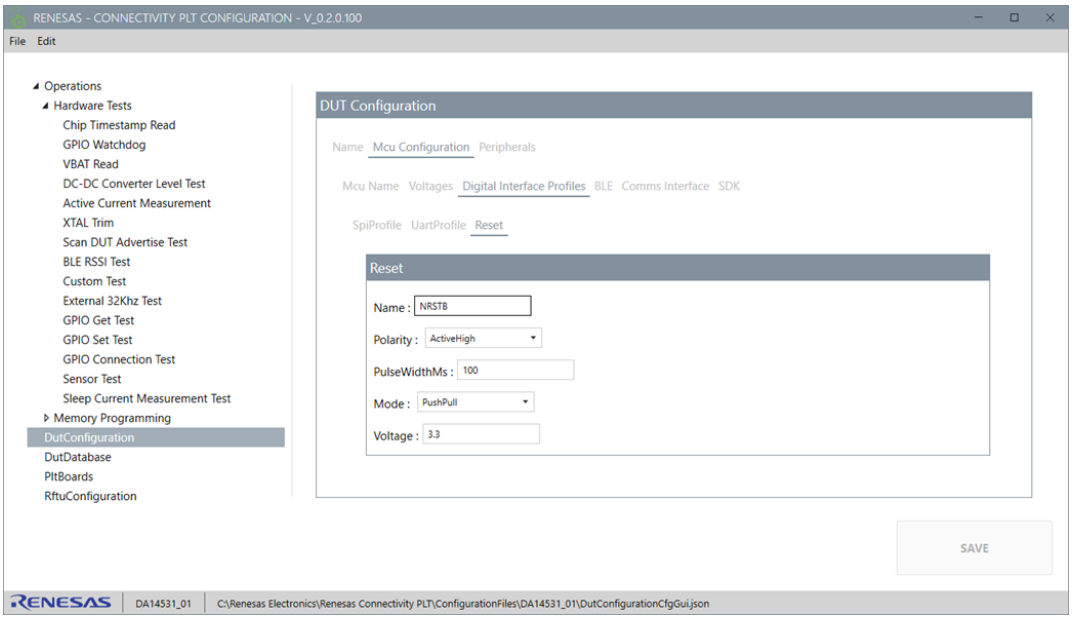



**Figure 45. Reset configuration**

**Table 11. Reset parameters**

| Option | Description |
|---|---|
| Name | NRSTB |
| Polarity | Available options: ActiveHigh, ActiveLow |
| Pulse Width (ms) | 100 |
| Mode | Available options: PushPull, OpenDrain |
| Voltage | 3.3 |

- **Bluetooth® LE**

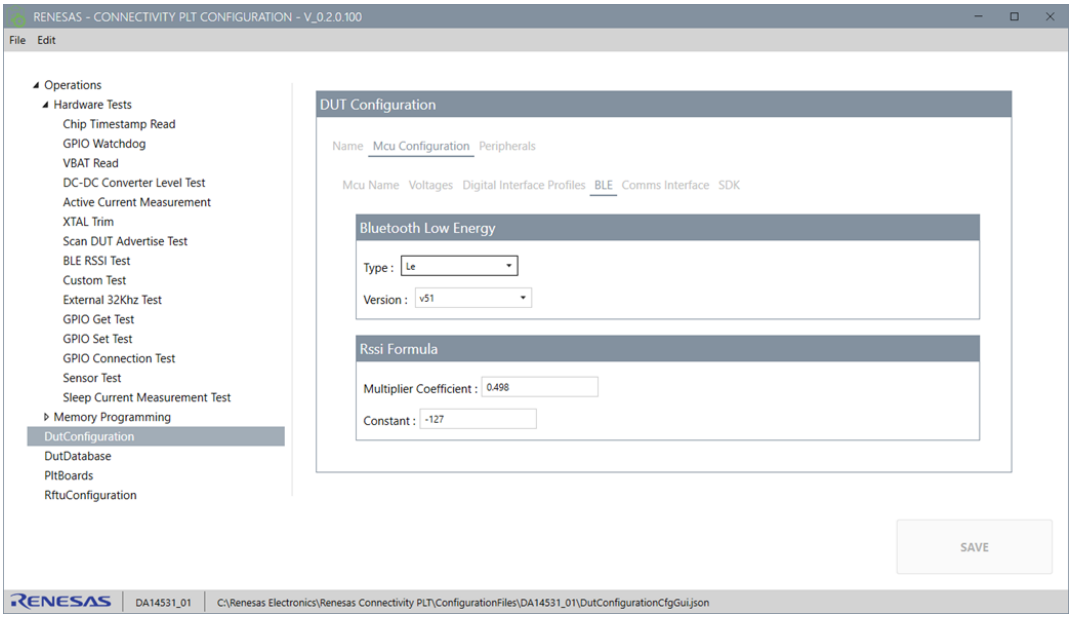



**Figure 46. Bluetooth® LE configuration**

**Table 12. Bluetooth® LE parameters**

| Parameter | Description |
|---|---|
| **Bluetooth Low Energy** | |
| Type | Type of the Bluetooth® LE interface. Available options: Classic, Le. |
| Version | Version of the Bluetooth® LE interface. Available options: v40, v41, v42, v50, v51, v52, v53. |
| **RSSI Formula** | |
| Multiplier Coefficient | The coefficient used in the calculation of the RSSI (Conversion factor) |
| Constant | The constant used in the calculation of the RSSI (Noise floor) |

- **Comms Interface**

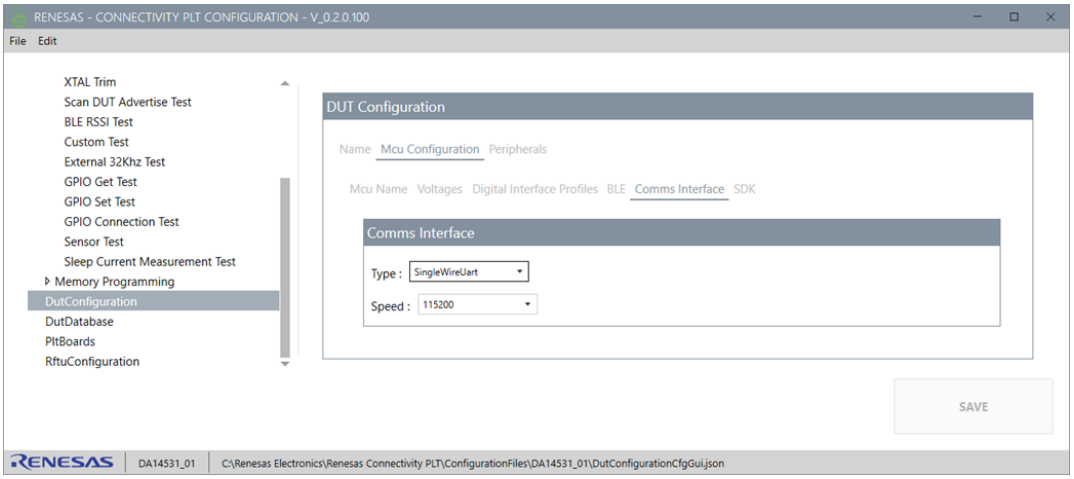



**Figure 47. Comms interface configuration**

**Table 13. Comms interface options**

| Option | Description |
|---|---|
| **Config** | |
| Type | Available options: Uart, SingleWireUart |
| Baud Rate | Available options: 1200, 2400, 4800, 9600,19200, 38400, 57600, 76800, 115200, 230400, 460800, 576000, 921600, 1M, 2M, 3M, 4M, 5M, 6M |

- **SDK**

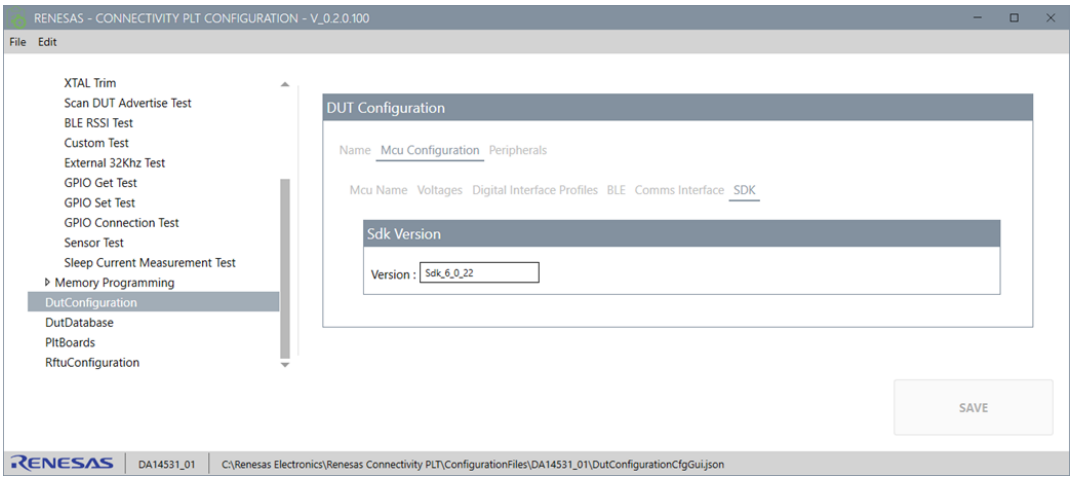



**Figure 48. SDK Version configuration**

**Table 14. SDK version parameters**

| Parameter | Description |
|---|---|
| **SDK Version** | |
| Version | Version of the SDK used |

### 8.2.7.3 DUT Configuration – Peripherals

- **Memory**

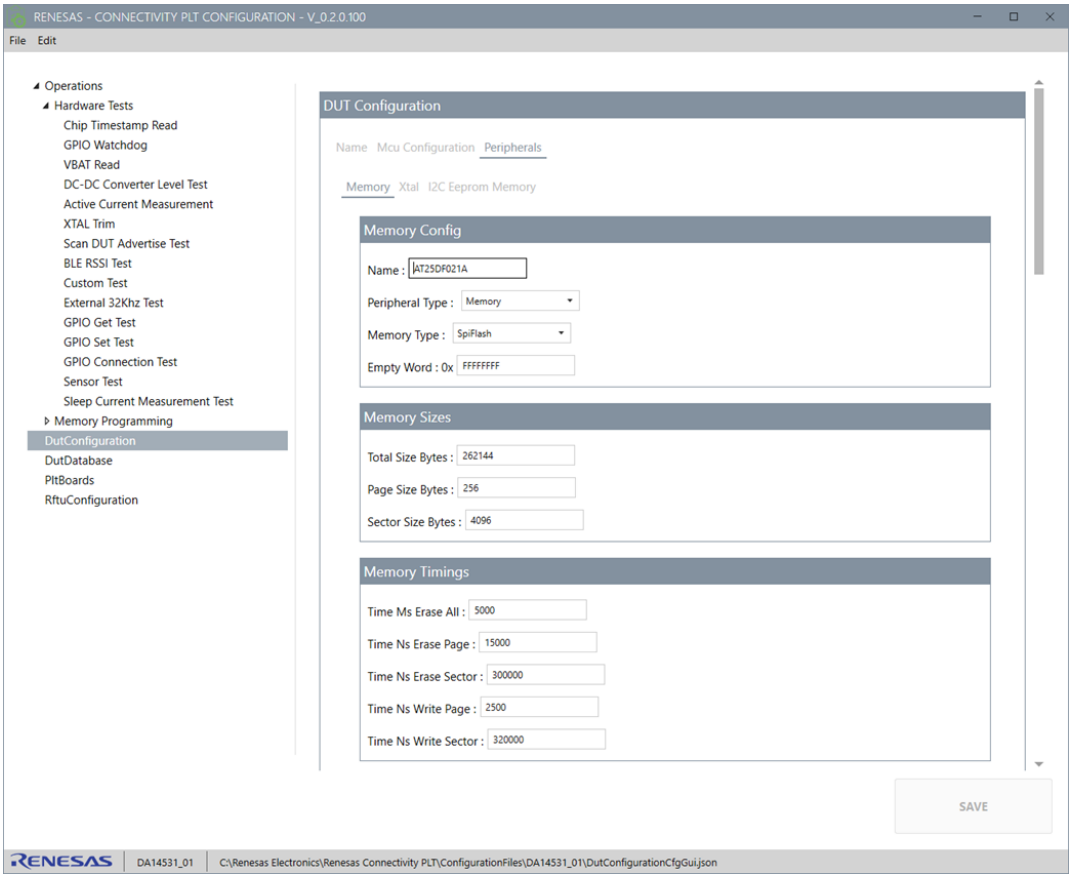



**Figure 49. Memory configuration (part 1)**

**Table 15. Memory parameters**

| Parameter | Description |
|---|---|
| **Memory Config** | |
| Name | Name of the configuration |
| Peripheral Type | Available options: Memory, Sensor, Antenna, Led, Button, Switch, Ftdi, Fpga, Dac, Adc, UsbHub, I2cGeneric, SpiGeneric, AddressSwitch, VoltageController, DutController, Nex74CB3Q3253, Ammeter, Xtal |
| Memory Type | Available options: SpiFlash, QspiFlash, OspiFlash, I2cEeprom, Otp, EmbeddedFlash, Rom, SpiEeprom |
| Empty Word | Content of empty flash: 0xFFFFFFFF |
| **Memory Sizes** | |
| Total Size Bytes | Total memory size in bytes |
| Page Size bytes | Page size in bytes |
| Sector Size Bytes | Sector size in bytes |

| Parameter | Description |
|---|---|
| **Memory Timings** | |
| Time Ms Erase All | Time to erase all (chip erase) in milliseconds |
| Time Ns Erase Page | Time to erase a single page in nanoseconds |
| Time Ns Erase Sector | Time to erase a single sector in nanoseconds |
| Time Ns Write Page | Minimum time to write a single page in nanoseconds |
| Time Ns Write Sector | Minimum time to write a single sector in nanoseconds |

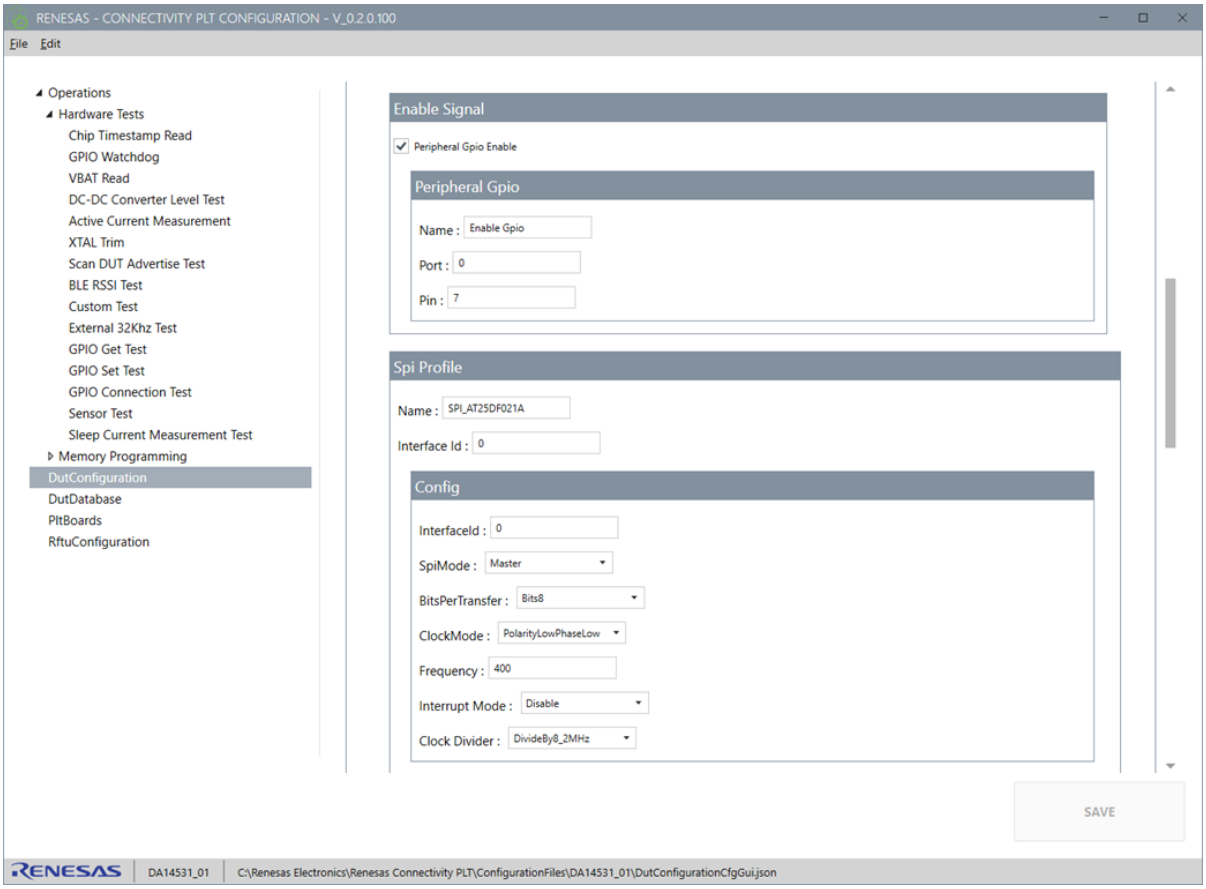



**Figure 50. Memory configuration (part 2)**

**Table 16. Peripheral GPIO parameters**

| Parameter | Description |
|---|---|
| **Enable Signal** | |
| Peripheral GPIO Enable | Enable the peripheral GPIO |
| **Peripheral GPIO** | |
| Name | Name of Peripheral GPIO |
| Port | GPIO Port |
| Pin | GPIO Pin |

**Table 17. SPI profile parameters**

| Parameter | Description |
|---|---|
| **SPI Profile** | |
| Name | Name of SPI profile |

| Parameter | Description |
|---|---|
| Interface ID | Interface ID |
| **Config** | |
| Interface ID | Interface ID |
| Bits Per Transfer | Available options: Bits8, Bits16 |
| Clock Mode | Available options: PolarityLowPhaseLow, PolarityLowPhaseHigh, PolarityHighPhaseLow, PolarityHighPhaseHigh |
| Frequency | Frequency in kHz |
| Interrupt Mode | Enable or disable Interrupt Mode |
| Clock Divider | Available options: DivideBy8_2MHz, DivideBy4_4MHz, DivideBy2_8MHz, DivideBy14_16MHz |

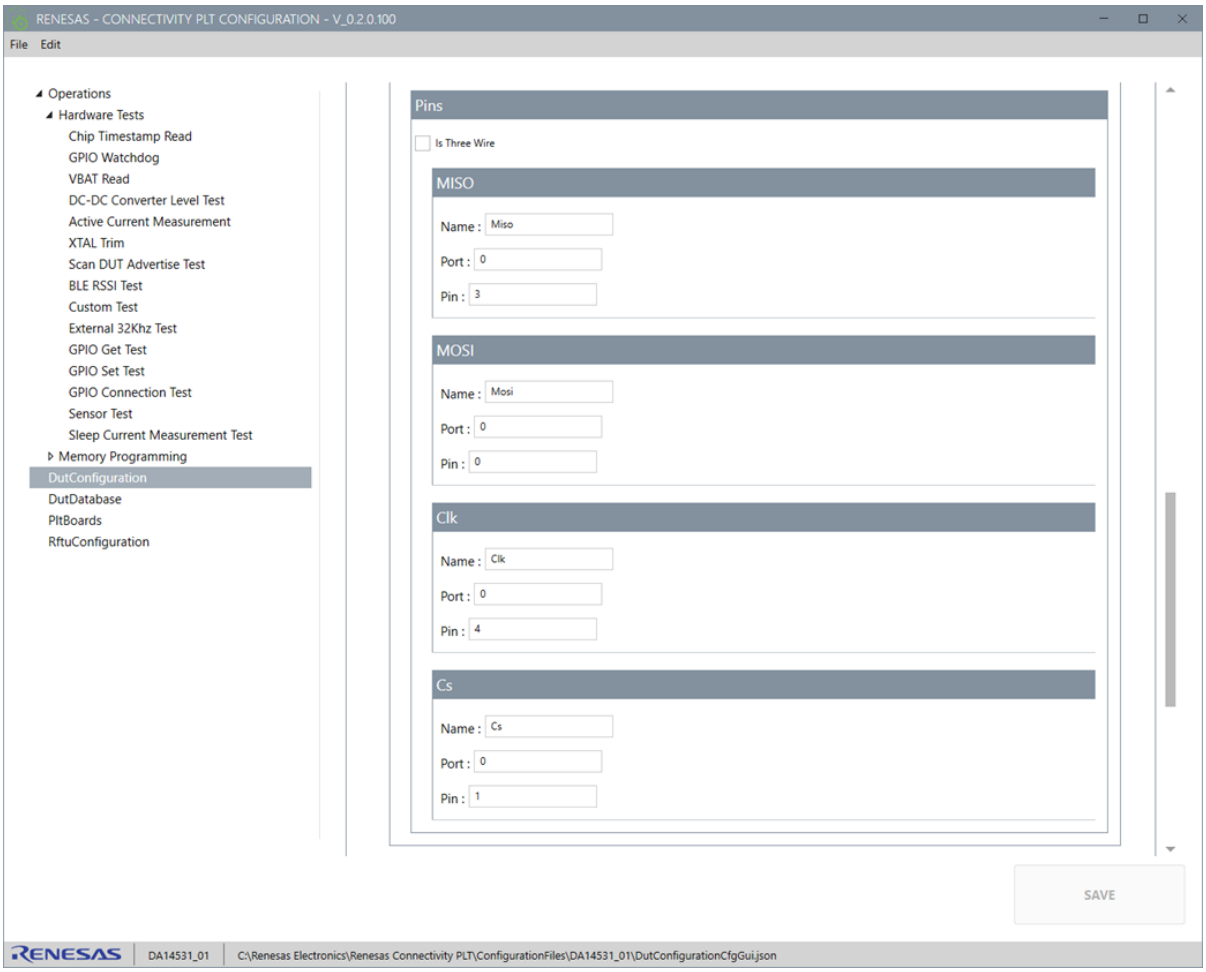



**Figure 51. Pins configuration**

**Table 18. Pins parameters**

| Parameter | Description |
|---|---|
| Is 3-wire | Enables 3-wire communication |
| **MISO** | |
| Name | Name of MISO configuration |
| Port | MISO GPIO port |
| Pin | MISO GPIO pin |
| **MOSI** | |

| Parameter | Description |
|---|---|
| Name | Name of MOSI configuration |
| Port | MOSI GPIO port |
| Pin | MOSI GPIO pin |
| **SCLK** | |
| Name | Name of SCLK configuration |
| Port | SCLK GPIO port |
| Pin | SCLK GPIO pin |
| **CS** | |
| Name | Name of CS configuration |
| Port | CS GPIO port |
| Pin | CS GPIO pin |

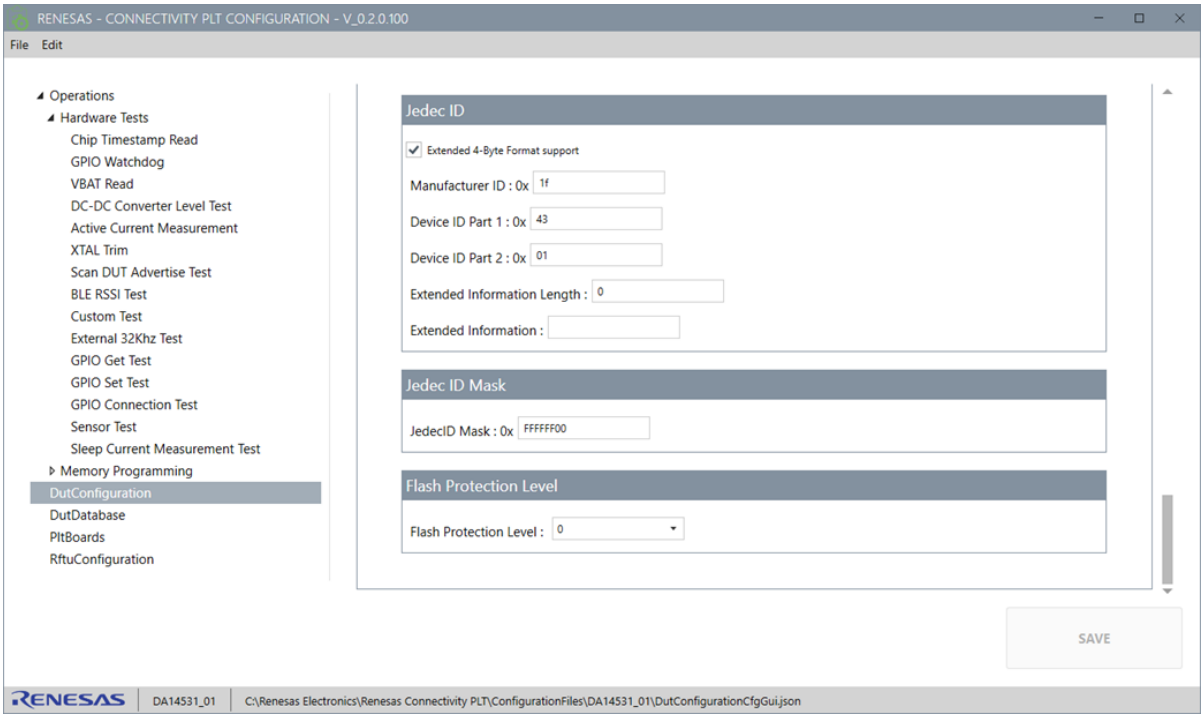



**Figure 52. JEDEC ID configuration**

**Table 19. JEDEC ID parameters**

| Parameter | Description |
|---|---|
| **JEDEC ID** | |
| Extended 4-byte Format Support | Enable or Disable 4-byte format support |
| Manufacturer ID | Part of JEDEC ID |
| Device ID Part1 | Part of JEDEC ID |
| Device ID Part2 | Part of JEDEC ID |
| Extended Information Length | Number of bytes of extended information |
| Extended Information | JEDEC ID extended information |
| **JEDEC ID Mask** | |
| JEDEC ID Mask | JEDEC ID mask to ignore bytes |

| Parameter | Description |
|---|---|
| **Flash Protection Level** | |
| Flash Protection Level | (0-15) |

- **XTAL**

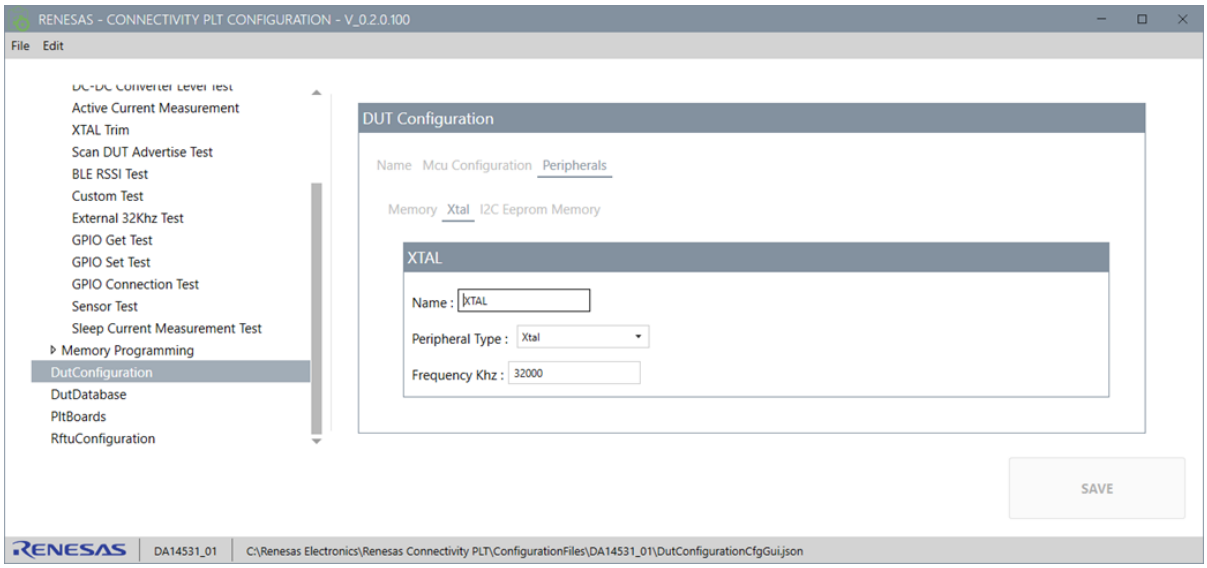



**Figure 53. XTAL configuration**

**Table 20. XTAL parameters**

| Option | Description |
|---|---|
| **XTAL** | |
| Name | Name of XTAL |
| Peripheral Type | Available options: Memory, Sensor, Antenna, Led, Button, Switch, Ftdi, Fpga, Dac, Adc, UsbHub, I2cGeneric, SpiGeneric, AddressSwitch, VoltageController, DutController, Nex74CB3Q3253, Ammeter, Xtal |
| Frequency (kHz) | XTAL Frequency in kHz |

- **I²C EEPROM Memory**

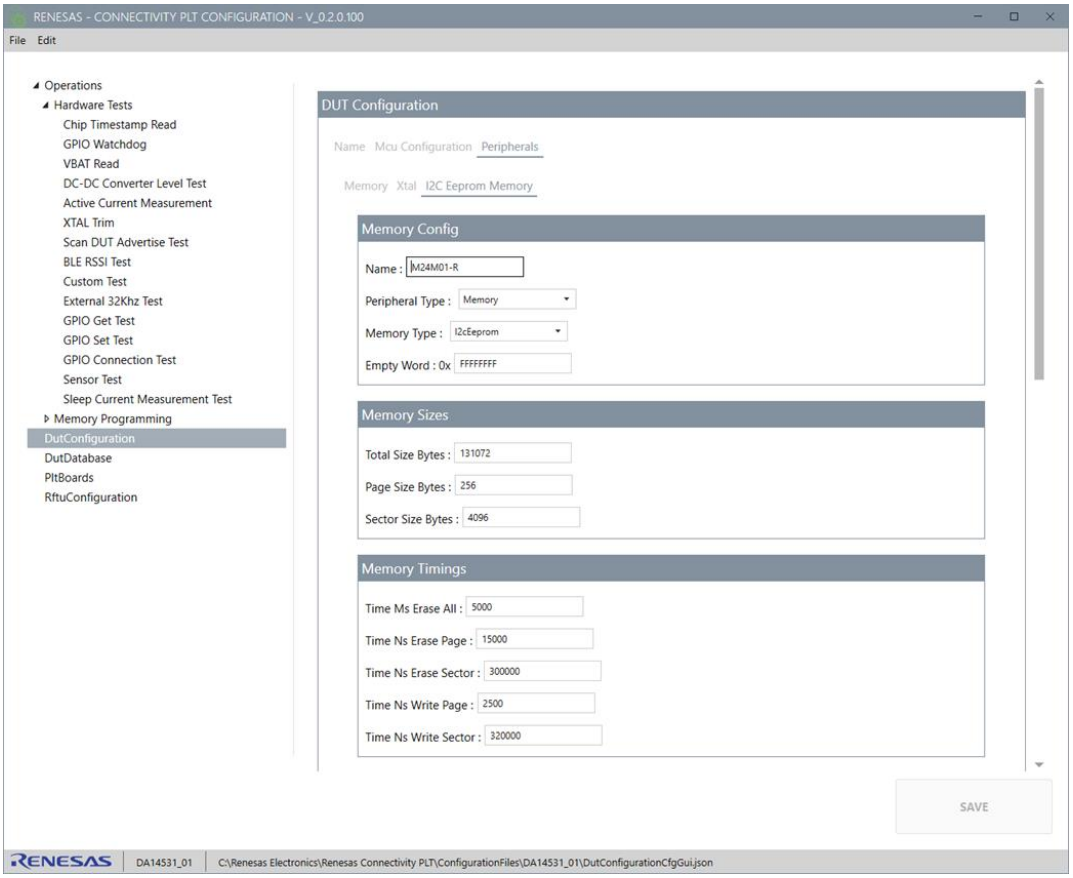



**Figure 54. I²C EEPROM configuration**

**Table 21. I²C parameters**

| Parameter | Description |
|---|---|
| **Memory Config** | |
| Name | Name of memory configuration |
| Peripheral Type | Available options: Memory, Sensor, Antenna, Led, Button, Switch, Ftdi, Fpga, Dac, Adc, UsbHub, I2cGeneric, SpiGeneric, AddressSwitch, VoltageController, DutController, Nex74CB3Q3253, Ammeter, Xtal |
| Memory Type | Available options: SpiFlash, QspiFlash, OspiFlash, I2cEeprom, Otp, EmbeddedFlash, Rom, SpiEeprom |
| Empty Word | Contents of an empty EEPROM: 0xFFFFFFFF |

**Table 22. I²C size parameters**

| Parameter | Description |
|---|---|
| **Memory Sizes** | |
| Total Size Bytes | Memory total size in bytes |
| Page Size bytes | Memory page size in bytes |
| Sector Size Bytes | Memory sector size in bytes |

**Table 23. I²C timings parameters**

| Parameter | Description |
|---|---|
| **Memory Timings** | |
| Time Ms Erase All | Time to erase all (chip erase) in milliseconds |

| Parameter | Description |
|---|---|
| Time Ns Erase Page | Time to erase a single page in nanoseconds |
| Time Ns Erase Sector | Time to erase a single sector in nanoseconds |
| Time Ns Write Page | Minimum time to write a single page in nanoseconds |
| Time Ns Write Sector | Minimum time to write a single sector in nanoseconds |

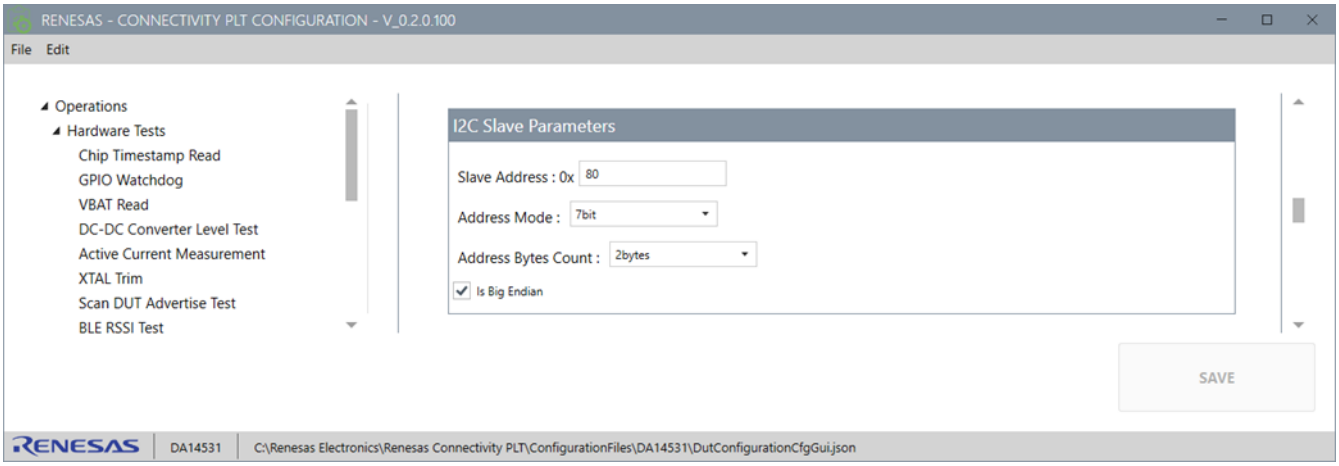



**Figure 55. I²C Slave Parameters configuration**

**Table 24. I²C Slave parameters**

| Parameter | Description |
|---|---|
| **I²C Slave Parameters** | |
| Slave Address | Slave address in hexadecimal format |
| Address Mode | Available options: 7bit, 10bit |
| Address Byte Count | Available options: 1byte, 2bytes, 3bytes |
| Is Big Endian | Set if Big Endian |

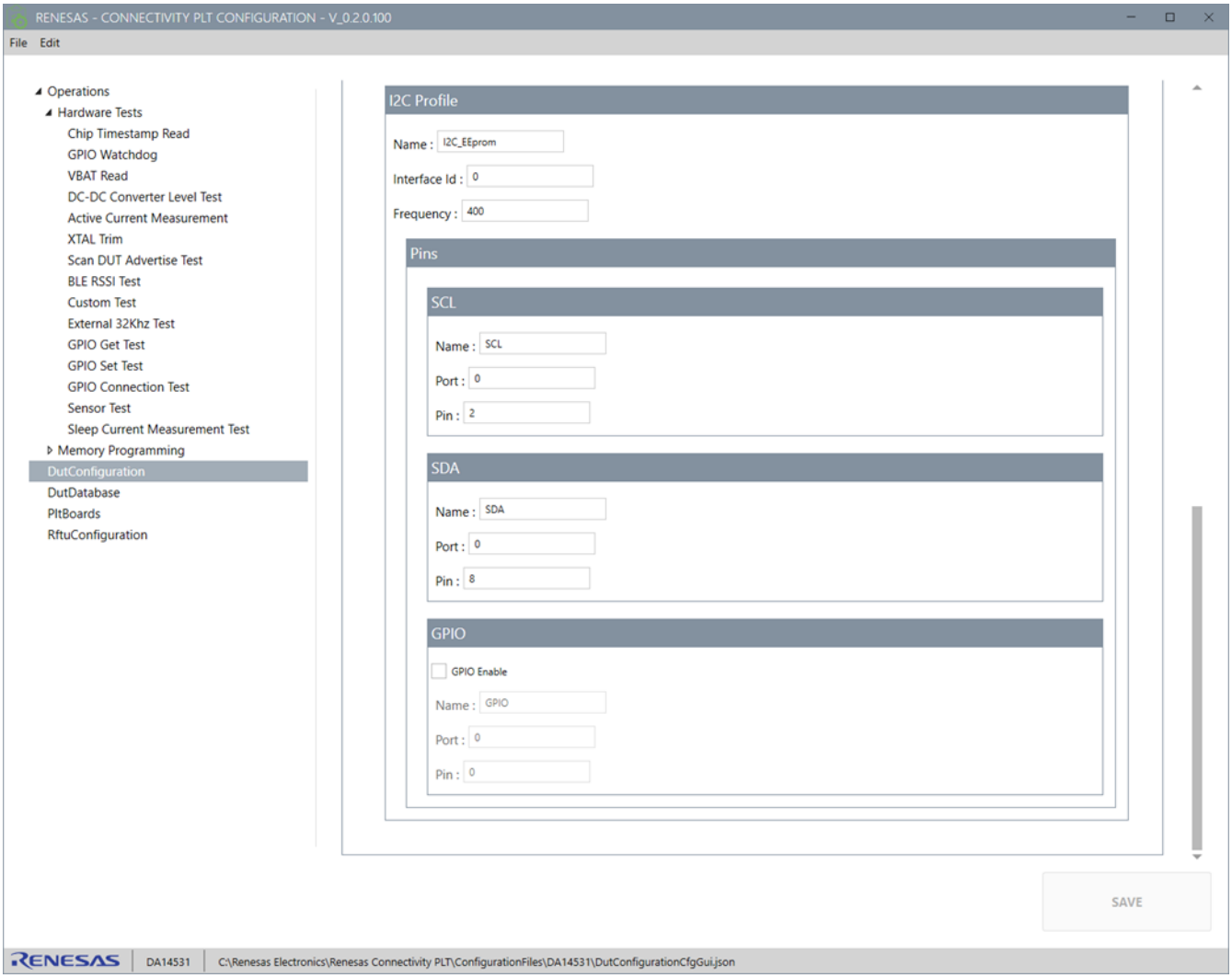



**Figure 56. I²C Profile configuration**

**Table 25. I²C Profile parameters**

| Parameter | Description |
|---|---|
| **I²C Profile** | |
| Name | Profile name |
| Interface ID | Interface ID |
| Frequency | Profile frequency in kHz |

**Table 26. I²C Pins parameters**

| Option | Description |
|---|---|
| **SCL** | |
| Name | SCL name |
| Port | SCL GPIO port |
| Pin | SCL GPIO pin |
| **SDA** | |
| Name | SDA name |
| Port | SDA GPIO port |
| Pin | SDA GPIO pin |

| Option | Description |
|---|---|
| **GPIO** | |
| GPIO Enable | Enable GPIO |
| Name | GPIO name |
| Port | GPIO port |
| Pin | GPIO pin |

### 8.2.8 General Options

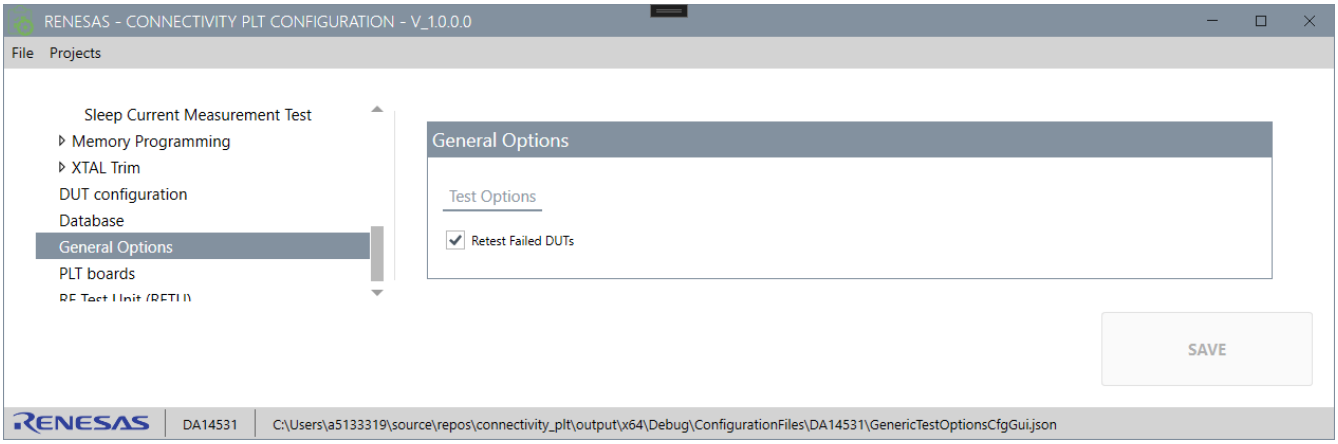



**Figure 57. General Options cocnfiguration**

**Table 27. General Options parameters**

| Parameter | Description |
|---|---|
| **General Options** | |
| Retest failed DUTs | Automatically retests only the failed DUTs |

### 8.2.9 Configure Sequence Number

A sequence number is assigned to every test and represents the order in which the tests are executed. The order is reflected in the side navigation panel. You can edit the sequence number.

If the sequence number is duplicated, an error message is displayed informing that the specific sequence number is in use and the test name that uses it, Figure 58.

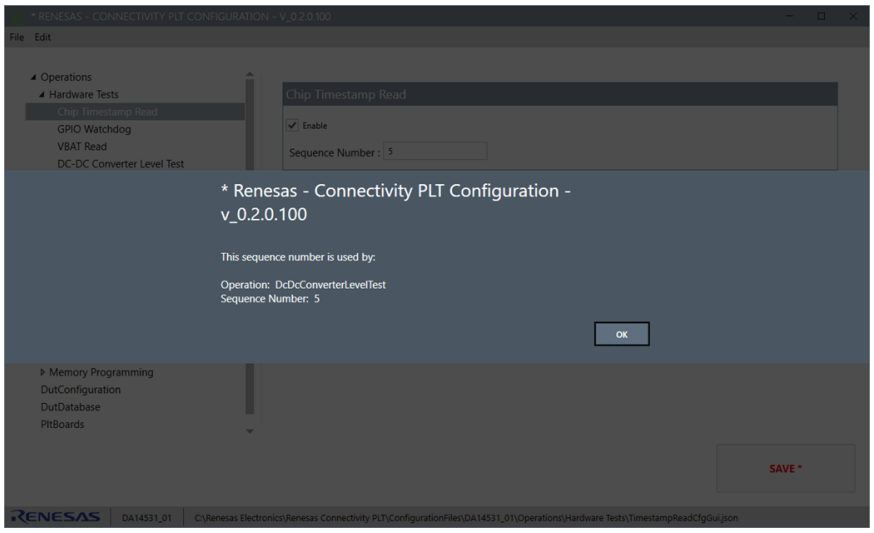



**Figure 58. Sequence number configuration**

## 8.3 Hardware Tests Configurations

This section groups all the tests that are not related to memory programming, see Table 28. Typically, each test represents a hardware function and has defined pass and fail criteria.

**Table 28. Type of hardware tests**

| Operation type | Description |
|---|---|
| Chip Timestamp Read | Not an actual test but is used mainly to log the information that was stored in the CS area of the OTP during factory programming. See Section 8.3.1. |
| GPIO Watchdog | Enables a periodic pulse at a specific CPU pin that can be utilized by the specific product as an external watchdog monitor. See Section 8.3.2. |
| VBAT Read | Sends a command to the DUTs to measure their VBAT using their internal ADC. See Section 8.3.3. |
| DC-DC Converter Level Test | Used to measure the output of the on-board DC-DC converter. See Section 8.3.4.<br>A lower and upper limit are used to define the pass/fail criteria. |
| Active Current Measurement | Measures the current consumed by the DUT at this stage of testing. In most cases this is the current consumption of the entire product. See Section 8.3.5. |
| XTAL Trim | Calculates the correct crystal oscillator trim value. The trim value of the crystal is important to achieve accurate frequency of the Bluetooth® LE radio. See Section 8.3.6. |
| Wi-Fi XTAL Trim | Calculates the correct crystal oscillator trim value. See Section 8.3.7. |
| Scan DUT Advertise | Tests the transmission of Bluetooth® LE advertising packets for each DUT. See Section 8.3.8. |
| Bluetooth LE RSSI Test | Tests the reception of Bluetooth® LE signals by the DUTs. See Section 8.3.9. |
| Custom Test | Used for vendor specific operations. These tests are developed within the production test firmware and follow a specific format. The tests are initiated by a single byte HCI command and report back a byte indicating the status of the test. See Section 8.3.10. |
| External 32-kHz Test | Checks if the 32-kHz crystal is present and operating correctly. See Section 8.3.11. |
| GPIO Get Test | Evaluates the state of a specific pin. This test is commonly used to check interoperability between other devices and the CPU in the customers' product; and can also be used to check for short circuit between pins. See Section 8.3.12. |
| GPIO Set Test | Sets a CPU pin to a specific state. It is commonly used for interoperability between other devices and the CPU in the customers' product; and can also be used to check for short circuit between pins in conjunction with a GPIO get Test. See Section 8.3.13. |
| GPIO Connection Test | Combines set and get GPIO operations to check connectivity between CPU pins. See Section 8.3.14. |
| Sensor Test | Tests interoperability between the CPU and attached sensors to the $I^2C$ or SPI bus. See Section 8.3.15. |
| Sleep Current Measurement Test | Measures the current consumption of the DUTs during sleep mode. Due to the extremely low currents, an external DMM instrument is required for this operation to make very accurate measurements. The measurement is performed on the VDUT power supply of the Connectivity PLT and is cumulative of the current from all DUTs. The result is divided by the number of active DUTs, report and compared to pass/fail criteria. See Section 8.3.16. |
| Read Power Values | Measures the current consumption of the DUTs during sleep mode. Due to the extremely low currents, an external DMM instrument is required for this operation to make very accurate measurements. The measurement is performed on the VDUT power supply of the Connectivity PLT and is cumulative of the current from all DUTs. The result is divided by the number of active DUTs, report and compared to pass/fail criteria. See Section 8.3.19. |

**Table 29. Tests and projects applicability**

| Hardware tests | DA14530, DA14531, DA14531MOD, DA14531-01, DA14533, DA14535, DA14530MOD | RA6W1, RRQ6100x | RA6W2, RRQ6105x | DA14850, DA14970 | DA14495, DA14AVDDECT |
|---|---|---|---|---|---|
| **Hardware tests** | | | | | |
| Active Current Measurement | ✓ | ✓ | ✓ | ✓ | |
| Chip Timestamp Read | ✓ | | ✓ | ✓ | |
| GPIO Watchdog | ✓ | | ✓ | ✓ | |
| VBAT Read | ✓ | | ✓ | ✓ | |
| DC-DC Converter Level Test | ✓ | | ✓ | ✓ | |
| Scan DUT Advertise Test | ✓ | | ✓ | ✓ | |
| Bluetooth RSSI Test | ✓ | | ✓ | ✓ | |
| Custom Test | ✓ | | ✓ | ✓ | |
| External 32 kHz Test | ✓ | | ✓ | ✓ | |
| GPIO Get Test | ✓ | | ✓ | ✓ | |
| GPIO Set Test | ✓ | | ✓ | ✓ | |
| GPIO Connection Test | ✓ | ✓ | ✓ | ✓ | |
| Sensor Test | ✓ | | ✓ | | |
| Sleep Current Measurement Test | ✓ | | ✓ | ✓ | |
| Power Rail Test | | ✓ | ✓ | | |
| Packet Error Rate Test | | ✓ | ✓ | | |
| Read Power Values | | | | ✓ | |
| **Memory programming** | | | | | |
| UART Baud Rate Set | ✓ | | | | |
| SPI Flash Initialize | ✓ | | | | |
| SPI Flash Erase All | ✓ | | | | |
| SPI Flash Erase Bock | ✓ | | | | |
| SPI Flash Write Binary | ✓ | | | | |
| SPI Flash Read | ✓ | | | | |
| EEPROM Memory Write | ✓ | | | | |
| EEPROM Memory Read | ✓ | | | | |
| OTP Memory Read | ✓ | ✓ | ✓ | ✓ | |
| OTP Memory Write | ✓ | ✓ | ✓ | ✓ | |
| OTP Encryption Key Write | ✓ | | | | |
| Custom Memory Data Write | ✓ | ✓ | ✓ | ✓ | |

| Hardware tests | DA14530, DA14531, DA14531MOD, DA14531-01, DA14533, DA14535, DA14530MOD | RA6W1, RRQ6100x | RA6W2, RRQ6105x | DA14850, DA14970 | DA14495, DA14AVDDECT |
|---|---|---|---|---|---|
| Common Memory Data Write | ✓ | ✓ | ✓ | ✓ | |
| Configuration Script Write | ✓ | ✓ | | ✓ | |
| Header Write | ✓ | ✓ | | ✓ | |
| QSPI Initialize | | | | | ✓ |
| QSPI Flash Erase All | | ✓ | ✓ | | ✓ |
| QSPI Flash Block Erase | | ✓ | ✓ | | ✓ |
| QSPI Flash Direct Write Binary | | ✓ | ✓ | | ✓ |
| QSPI Flash Read | | ✓ | ✓ | | ✓ |
| QSPI Test | | ✓ | ✓ | | |
| QSPI Flash Write COLA Image | | | | | ✓ |
| Wi-Fi Configuration Script Write | | ✓ | ✓ | | |
| OQSPI Flash Block Erase | | | | ✓ | |
| OQSPI Flash Direct Write Binary | | | | ✓ | |
| OQSPI Flash Read | | | | ✓ | |
| OQSPI Flash Erase All | | | | ✓ | |
| OQSPI Get State | | | | ✓ | |
| NVS Initialize | | | | | ✓ |
| NVS Defaults | | | | | ✓ |
| NVS Set Mode | | | | | ✓ |
| NVS Get Mode | | | | | ✓ |
| NVS Get RFPI | | | | | ✓ |
| NVS Set Memory | | | | | ✓ |
| NVS Read Memory | | | | | ✓ |
| NVS Set Image Configuration | | | | | ✓ |
| **XTAL Trim** | | | | | |
| XTAL Trim | ✓ | | ✓ | ✓ | |
| Wi-Fi XTAL Trim | | ✓ | ✓ | | |

### 8.3.1 Chip Timestamp Read

Chip Timestamp Read test is available in DA14530, DA14531, DA14531MOD, DA14531_1, DA14533, DA14535, DA14530MOD, RA6W2, DA14850, and DA14870, and can be useful for traceability and quality control.

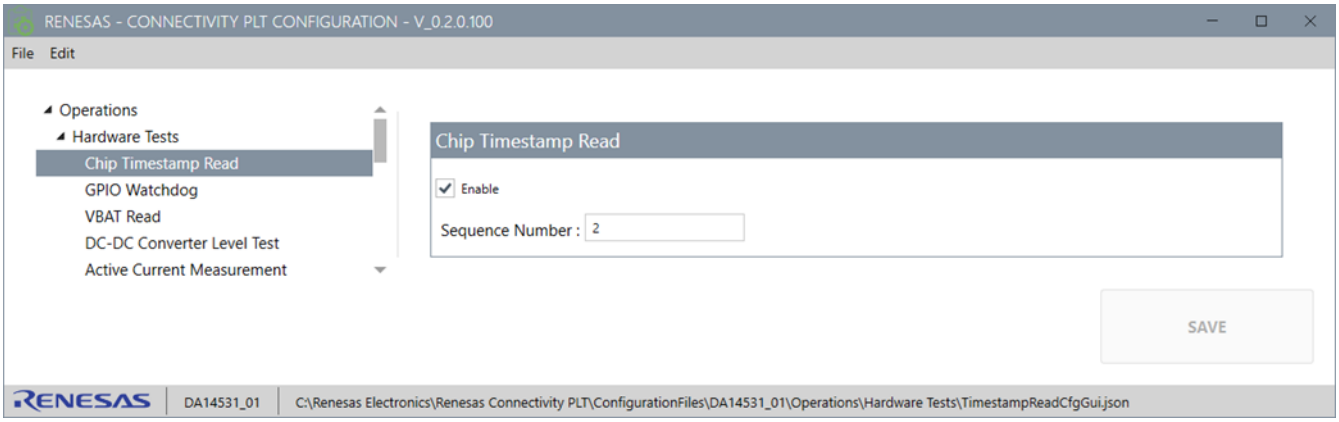



**Figure 59. Chip Timestamp Read configuration**

### 8.3.2 GPIO Watchdog

GPIO Watchdog test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

The pin output toggles throughout the entire test process except for the firmware download process. Following the firmware download process the watchdog output resumes.

The pulse on the GPIO has approximately 0.75% duty cycle and 0.5 Hz frequency.

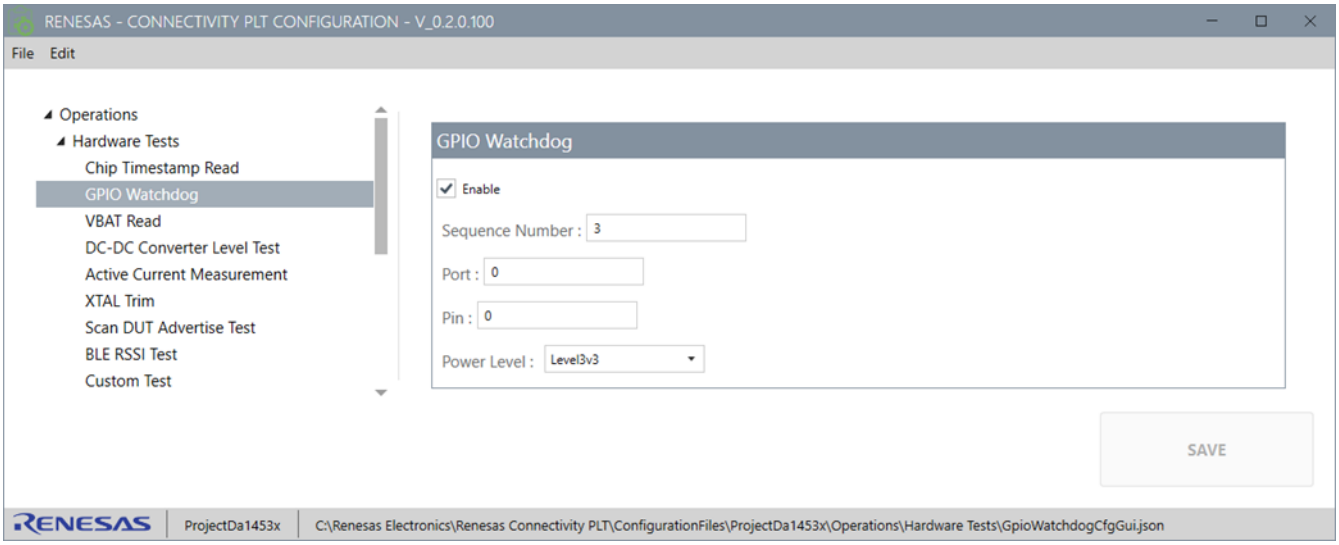



**Figure 60. GPIO Watchdog configuration**

### 8.3.3 VBAT Read

VBAT Read test is available in DA14530, DA14531, DA14531MOD, DA14531_1, DA14533, DA14535, DA14530MOD, RA6W2, RRQ6105x, DA14850, and DA14870.

This tool can be used to verify the operation and accuracy of the internal ADC. The VBAT level is further logged for traceability or any other purpose.

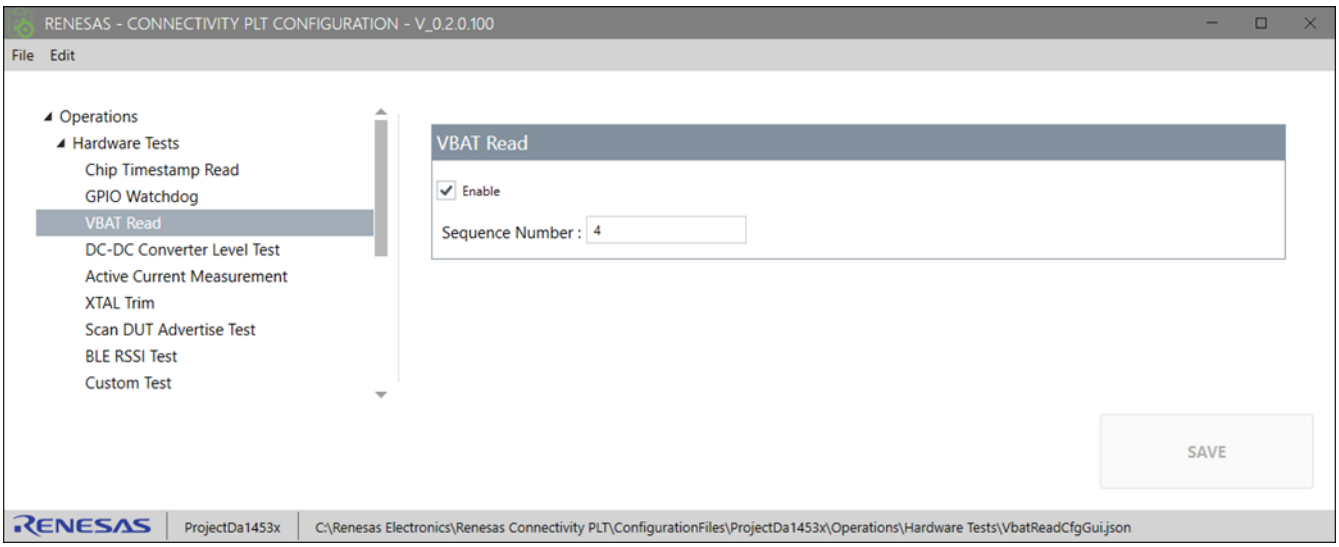



**Figure 61. VBAT Read configuration**

### 8.3.4 DC-DC Converter Level Test

DC-DC Converter Level Test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

This test verifies the operation of this function of the MCU and logs its operating characteristics. A lower and upper limit are used to define the pass/fail criteria.

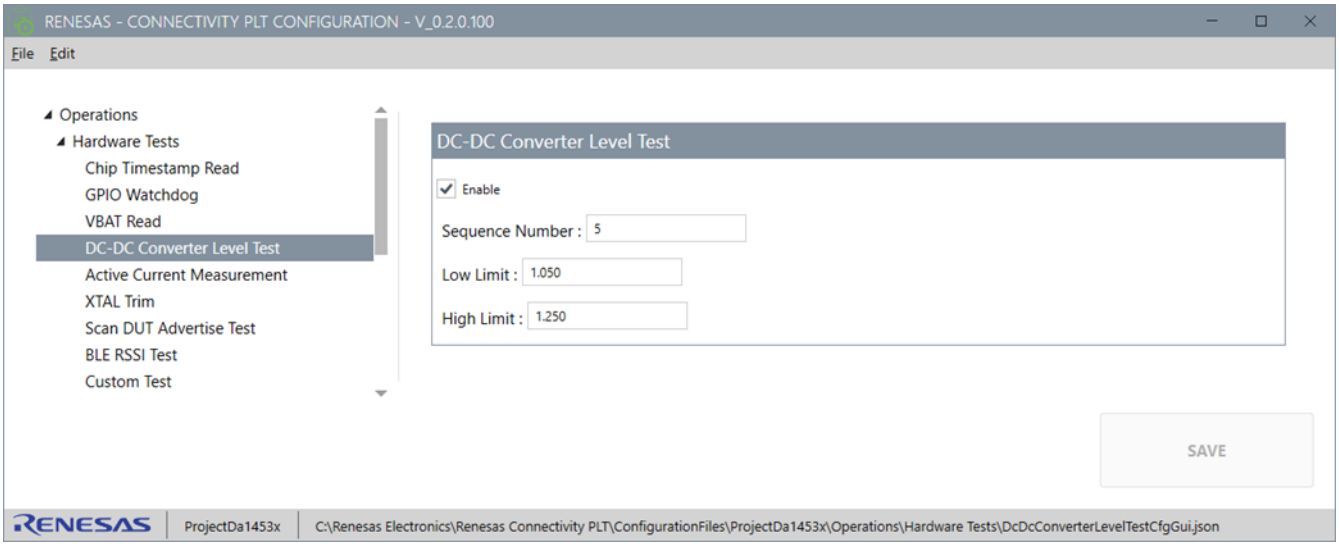



**Figure 62. DC-DC Converter Level Test configuration**

### 8.3.5 Active Current Measurement

Active Current Measurement test is available in DA1453x, RA6W1, RA6W2, RRQ6100x, RRQ6105x, DA14850, and DA14870.

This diagnostic test measures current consumption, which can be a significant indicator of DUT performance. An upper and lower limit are used to define the pass/fail criteria.

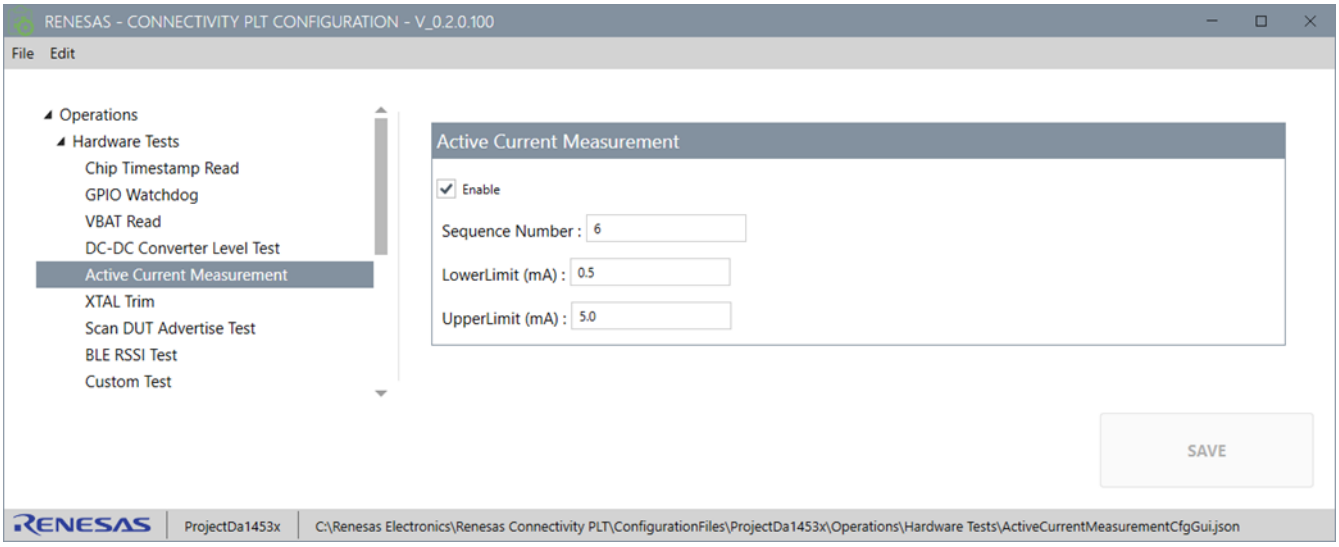



**Figure 63. Active Current Measurement configuration**

### 8.3.6 XTAL Trim

XTAL Trim test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

It is recommended that all devices with RF radio go through the trim process to maintain their frequency offset to minim PPM deviation. The XTAL trim process is completed using a combination of the Connectivity PLT Hardware clock signal and an internal calibration method of the DUT. For this process, a pin of the MCU is used to receive the calibration pulse. By default, this pin is the UART RX pin. There is also an option to use another pin to receive the pulse.

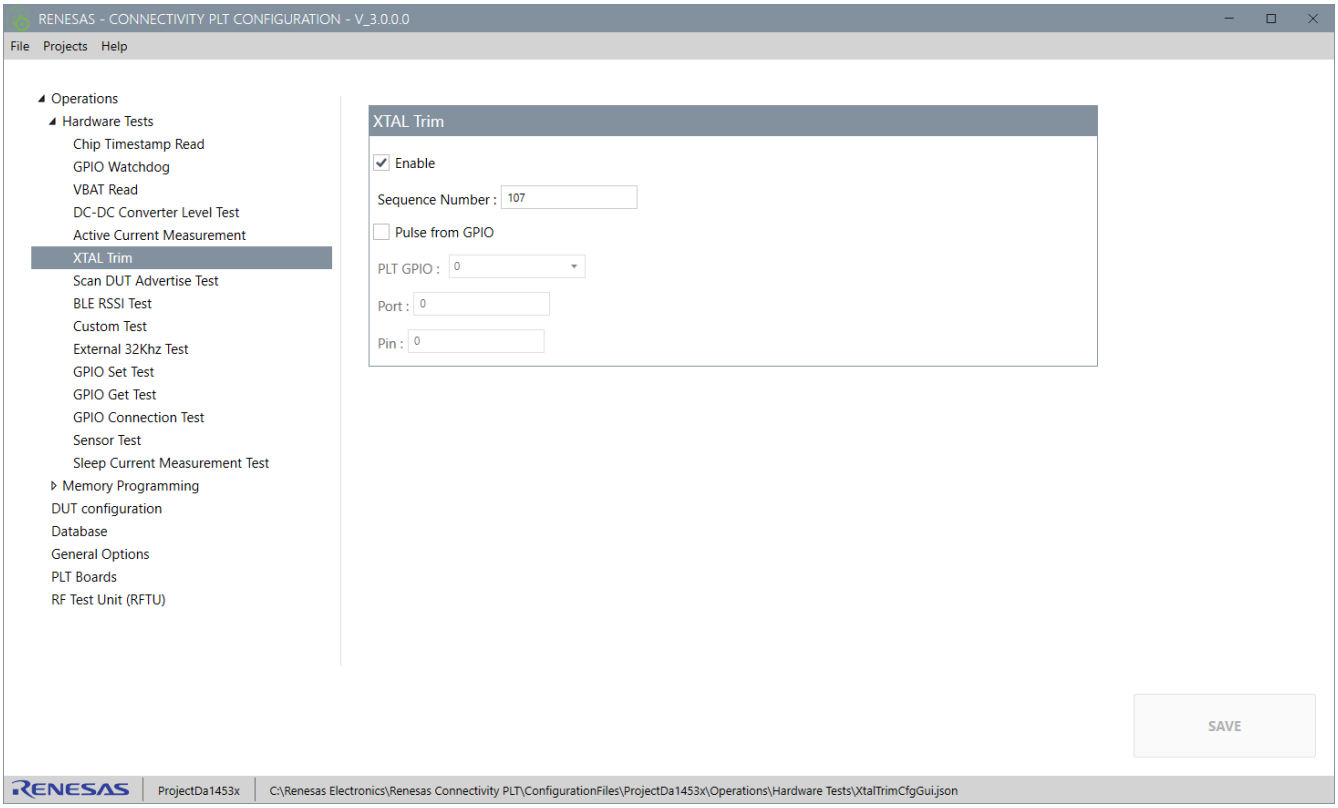



**Figure 64. XTAL Trim configuration**

### 8.3.7 Wi-Fi XTAL Trim

Wi-Fi XTAL Trim test is available in RA6Wx and RRQ6100x.

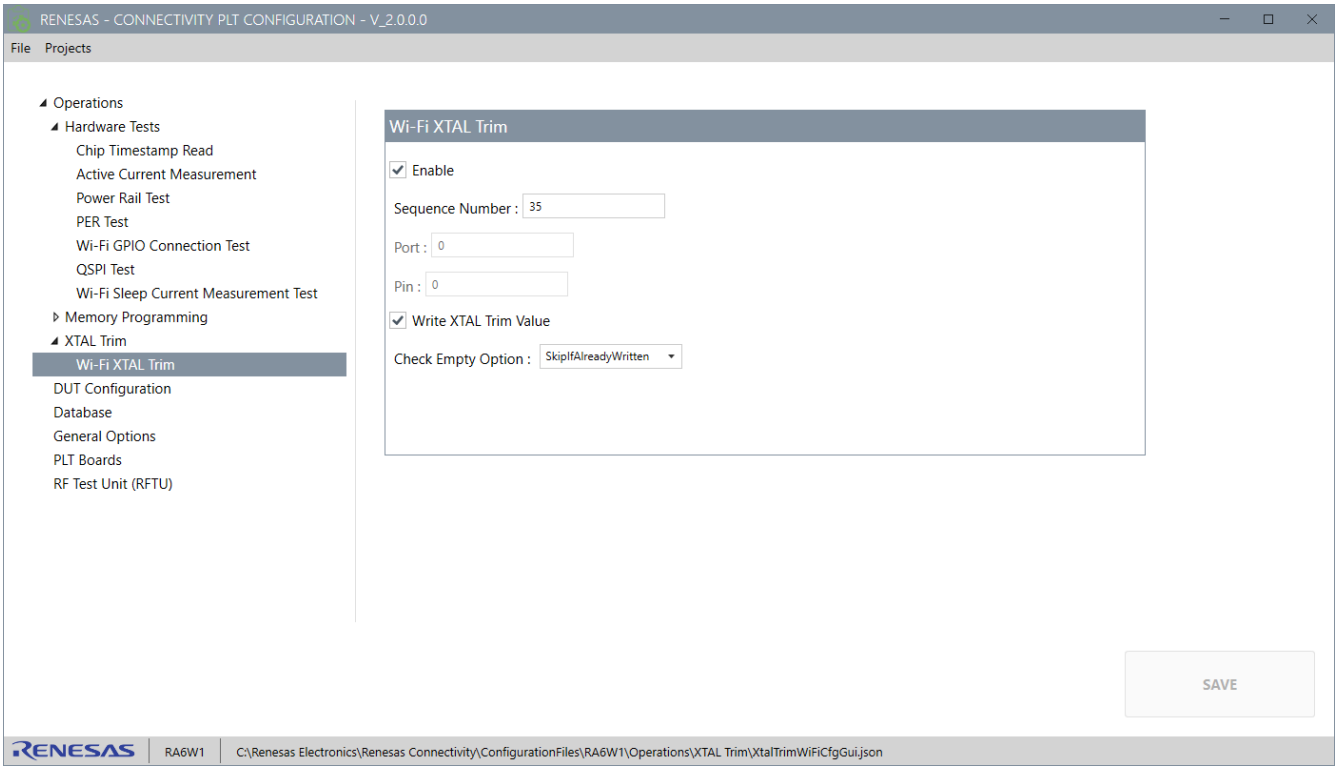



**Figure 65. Wi-Fi XTAL Trim configuration**

### 8.3.8 Scan DUT Advertise Test

Scan Dut Advertise Test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

This is a core test of the RF functionality of DUT. All DUTs are set in Bluetooth® LE advertising mode through standard HCI commands whilst the RFTU is set in reception mode looking to capture all the BD addresses that are advertising. During the initiation of the test temporary BD addresses are issued for the DUTs, these addresses are then compared to the received ones. The pass/fail criterial is the actual reception of the advertising packet and the RSSI (signal strength) of it.

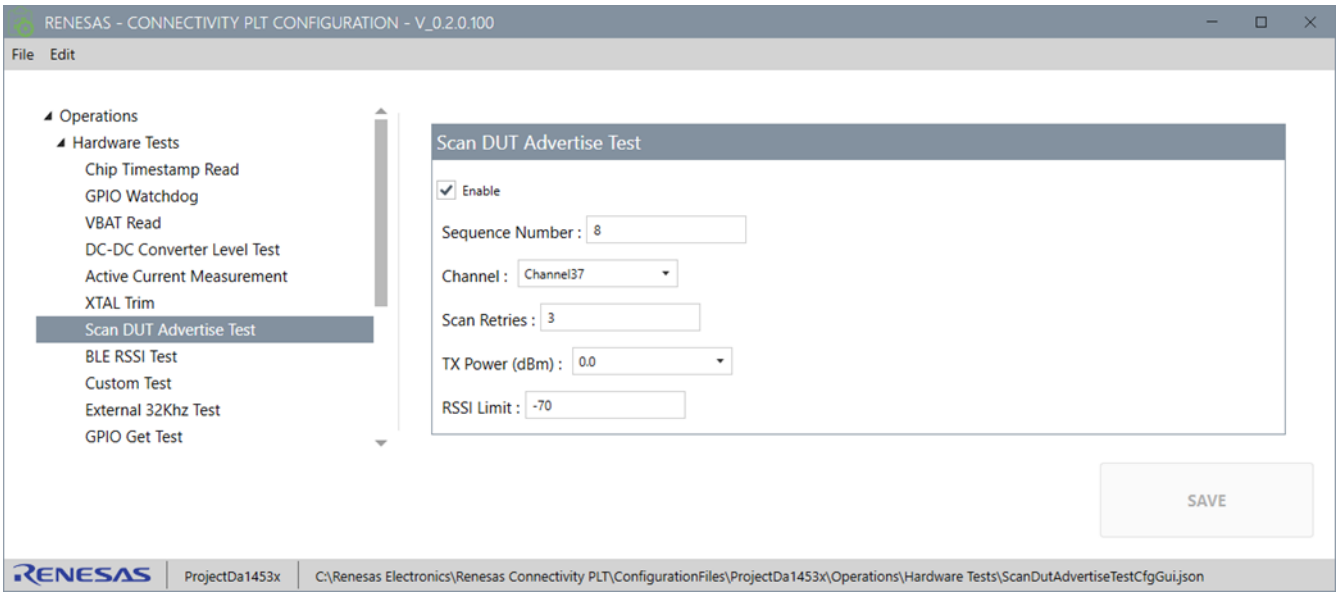



**Figure 66. Scan DUT Advertise Test configuration**

### 8.3.9 Bluetooth® LE RSSI Test

Bluetooth® LE RSSI Test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

A set of packets is transmitted by the RTFU using the specified settings and received by all the DUTs. The metrics of the received packets are used to determine if the reception is within the acceptable range. The set metrics tested are the average RSSI and the total PER.

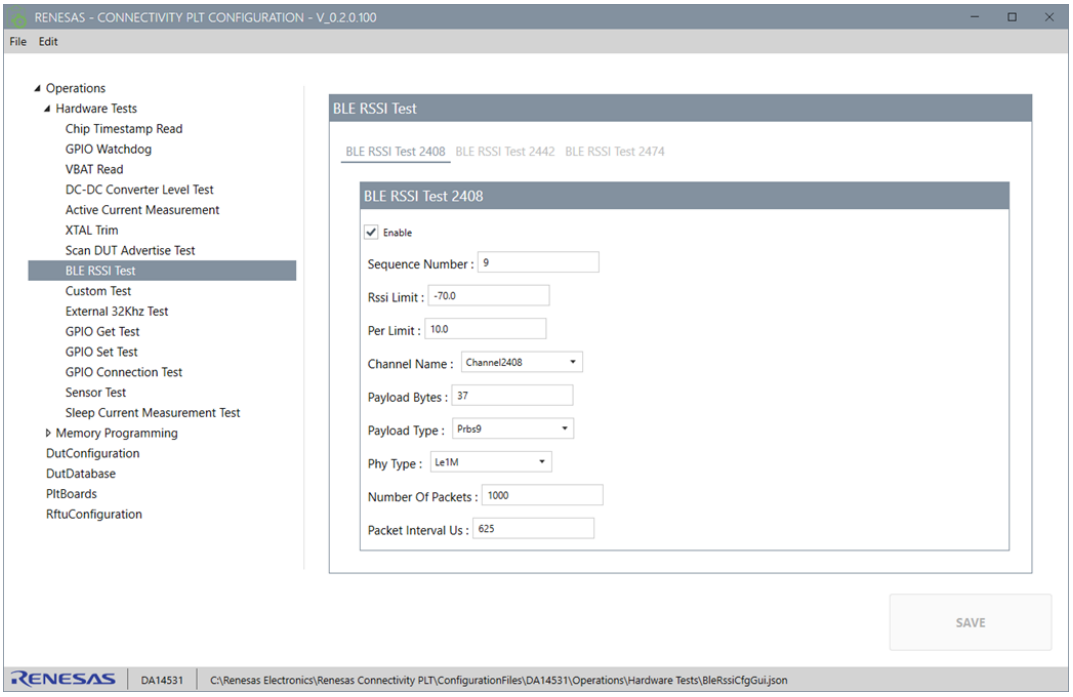



**Figure 67. Bluetooth® LE RSSI Test configuration**

The RSSI Limit is the lowest acceptable level for the DUT to pass the test. The result is calculated by averaging the RSSI measurement of all packets.

The PER Limit if the highest acceptable level for the DUT to pass the test. The result is calculated by the total packet errors.

### 8.3.10 Custom Test

Custom Test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

The tests are commonly used for extended sensor test procedures that require more than a single read-write operation. By default, a return value same as the command value indicates a pass status.

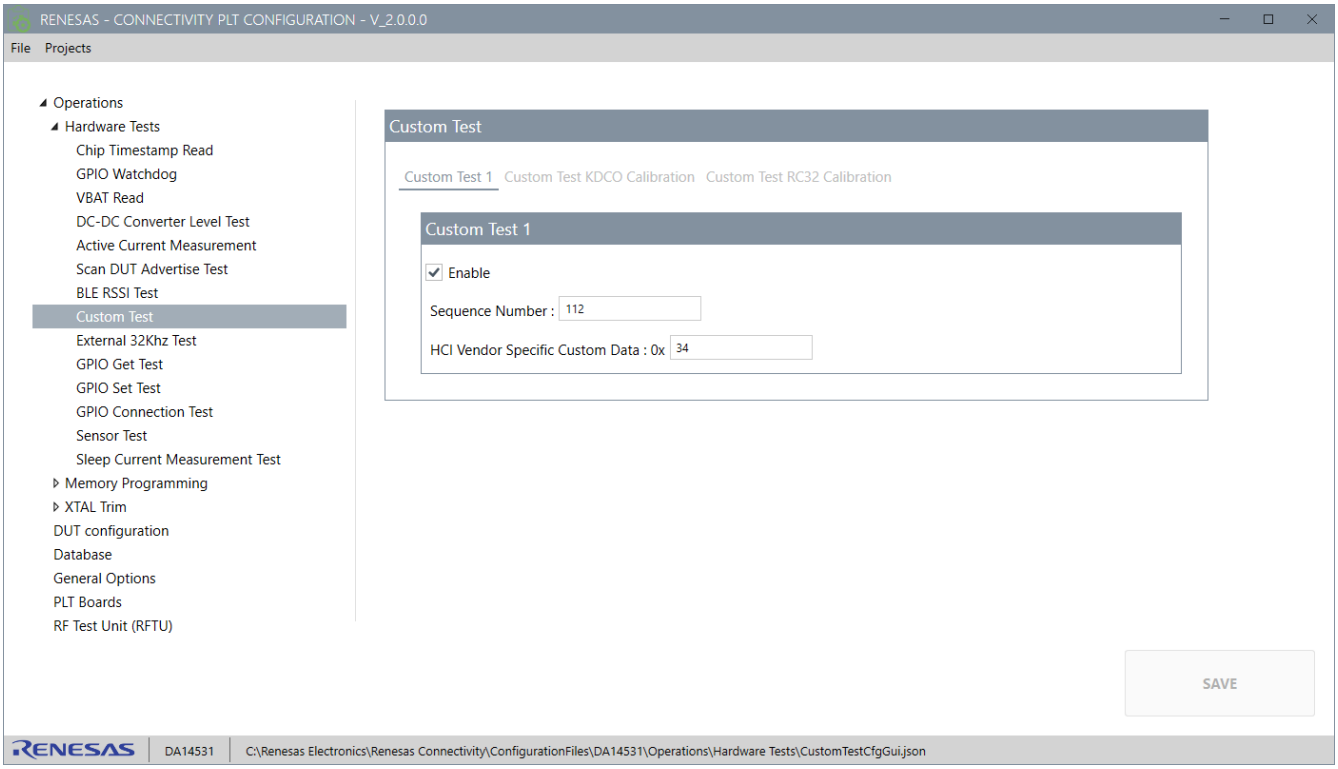



**Figure 68. Custom Test configuration**

### 8.3.11 External 32-kHz Test

External 32-kHz Test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

Its frequency is measured against the main clock source of the DUT. This test is only applicable on devices that support secondary oscillators.

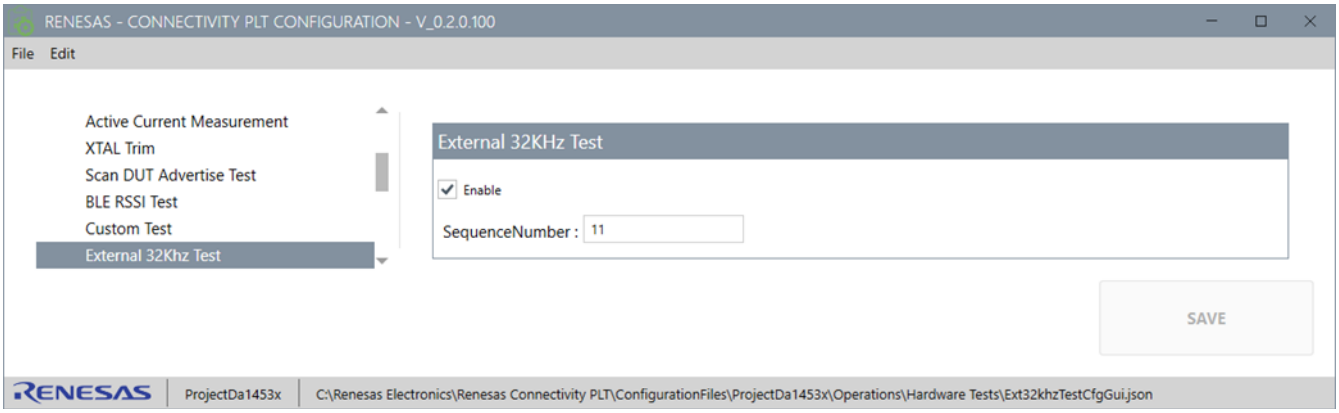



**Figure 69. External 32-kHz Test configuration**

### 8.3.12 GPIO Get Test

GPIO Get Test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

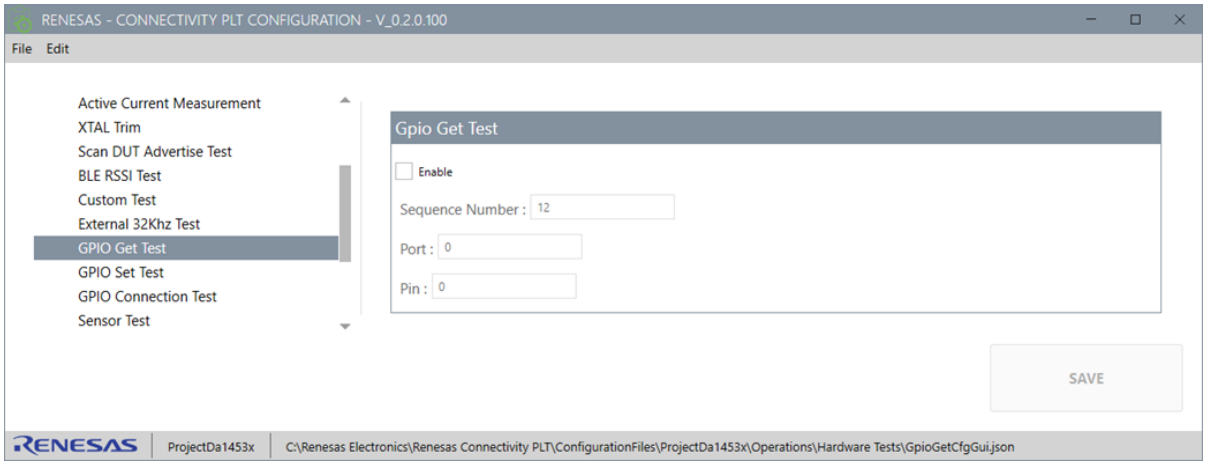



**Figure 70. GPIO Get Test configuration**

### 8.3.13 GPIO Set Test

GPIO Set Test is available in DA1453x, RA6W2, RRQ6105x, DA14850, and DA14870.

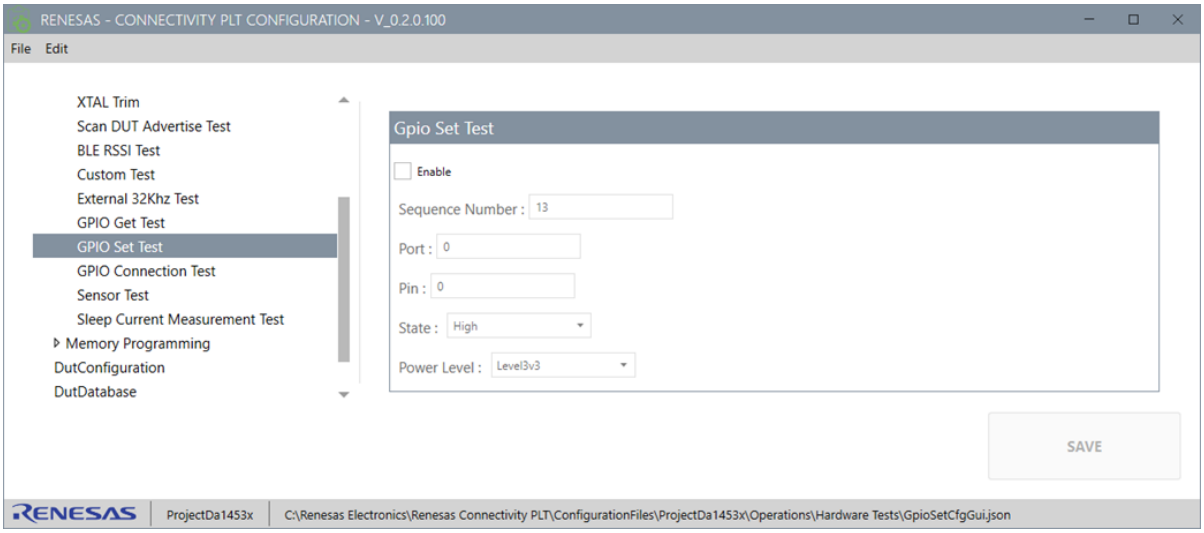



**Figure 71. GPIO Set Test configuration**

### 8.3.14 GPIO Connection Test

GPIO Connection Test is available in DA1453x, RA6W1, RA6W2, RRQ6100x , RRQ6105x, DA14850, and DA14870.

If the **Is Short** option is enabled, the operation checks whether **Set Pin** has the same level as **Get Pin** for all retries. Alternatively, the operation checks whether **Get Pin** is always low no matter what the **Set Pin** level is.

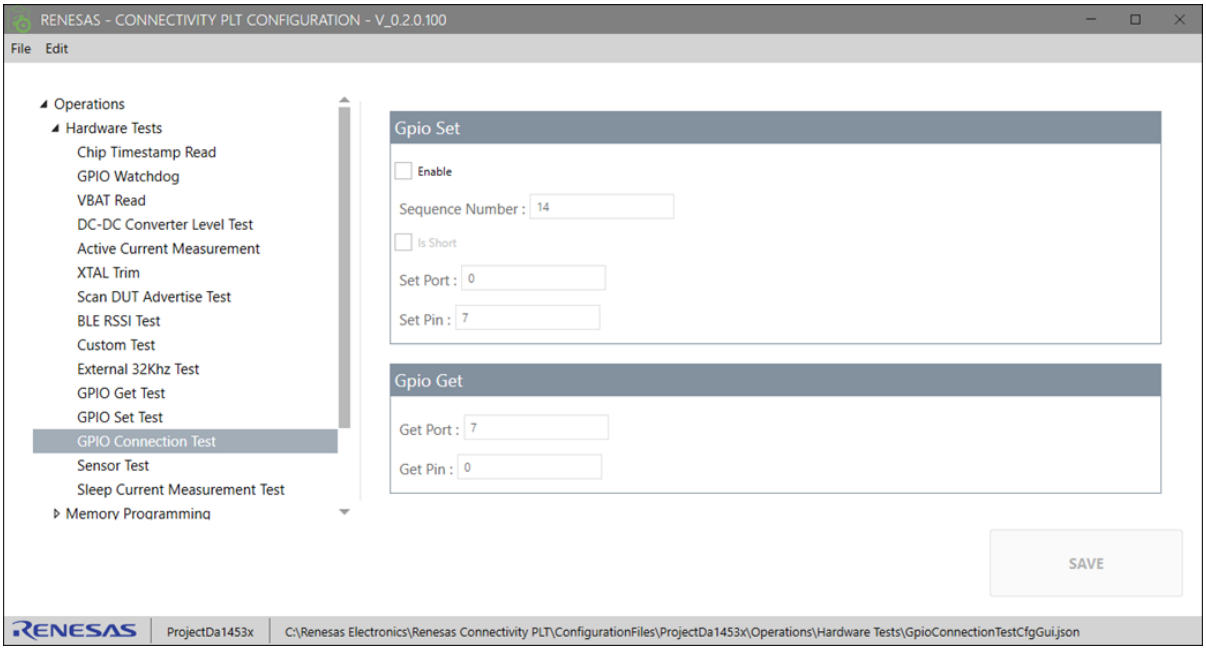



**Figure 72. GPIO Connection Test configuration**

### 8.3.15 Sensor Test

Sensor Test is available in DA1453x, RA6W2, and RRQ6105x.

The operation can perform read and write functions to the sensor at a specified register address. In the case of $I^2C$ devices the $I^2C$ address needs to be specified in 7-bit format. If the read function is used the **Expected Data** field is compared with the result to determine the success of the test. When the write function is used, it is followed by a read test to verify that the written value is correct, this determines the test result.

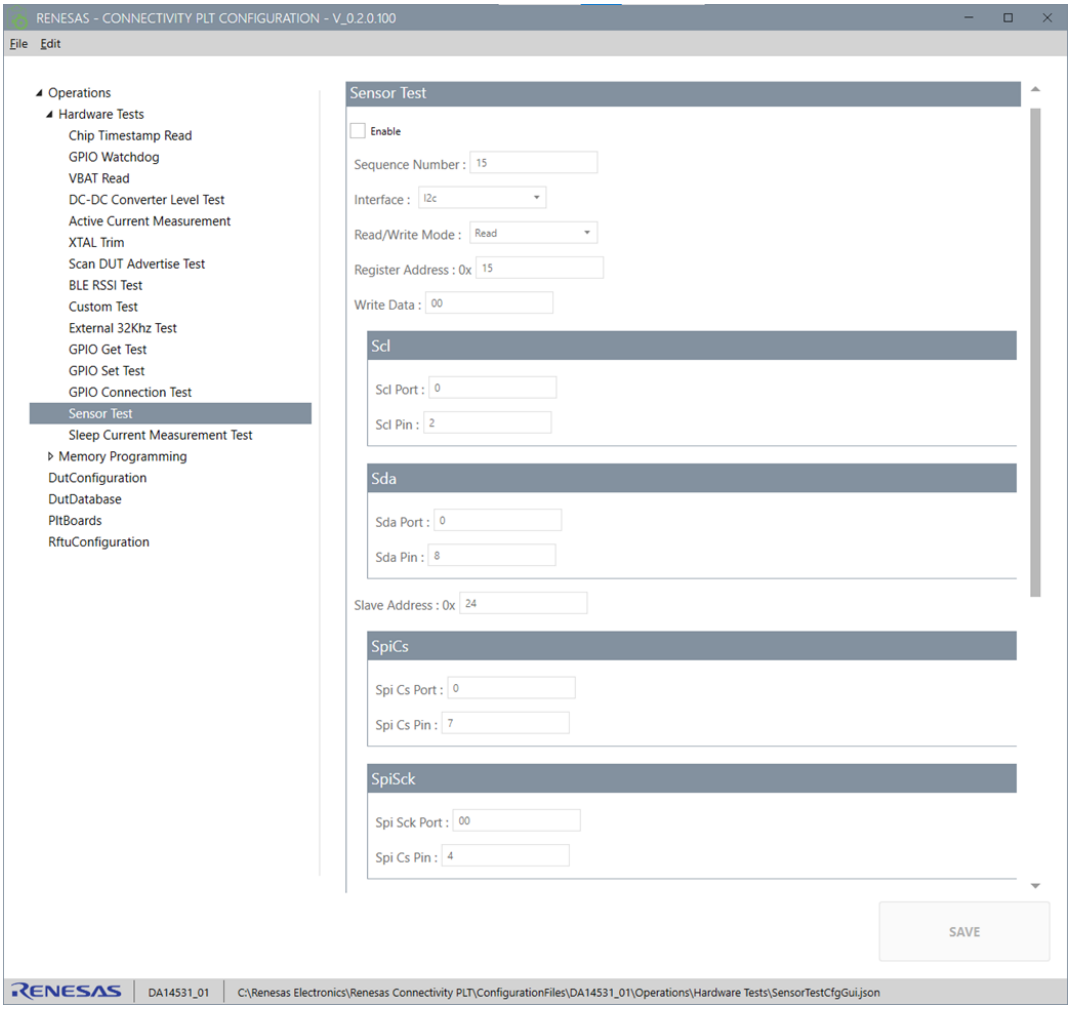



**Figure 73. Sensor Test configuration (part 1)**

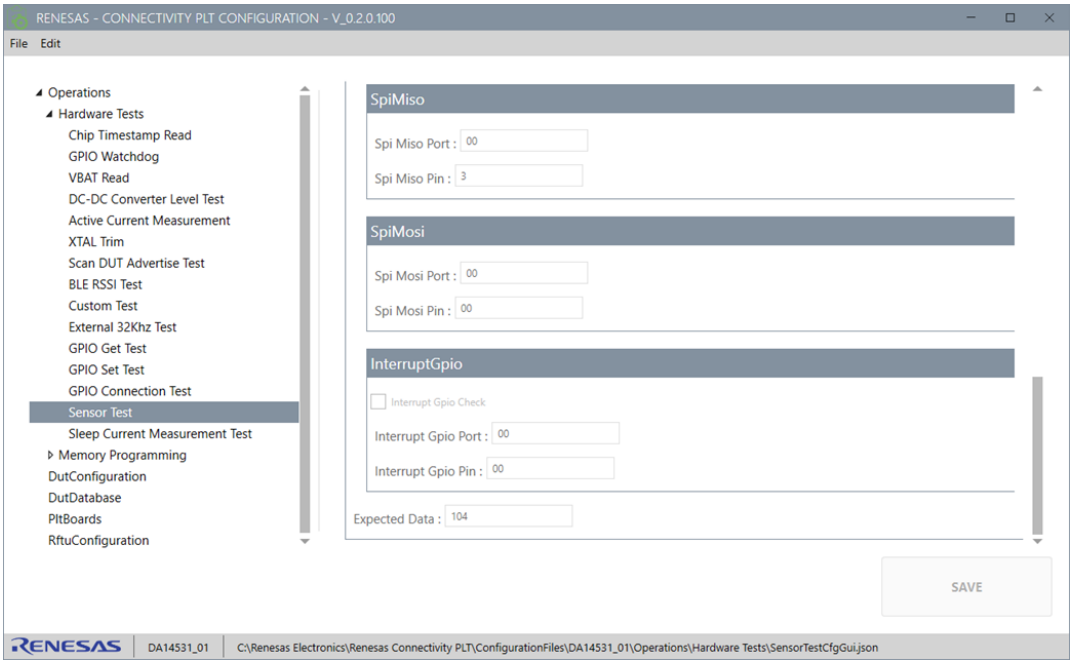



**Figure 74. Sensor test configuration (part 2)**

### 8.3.16 Sleep Current Measurement Test

Sleep Current Measurement Test is available in DA1453x, DA14850, and DA14870.

Due to the extremely low currents, an external DMM instrument is required for this operation to make very accurate measurements. For details on the instrument and connectivity, see Section 5.4.2. The measurement is performed on the VDUT power supply of the Connectivity PLT and is cumulative of the current from all DUTs. The result is divided by the number of active DUTs, report and compared to pass/fail criteria.

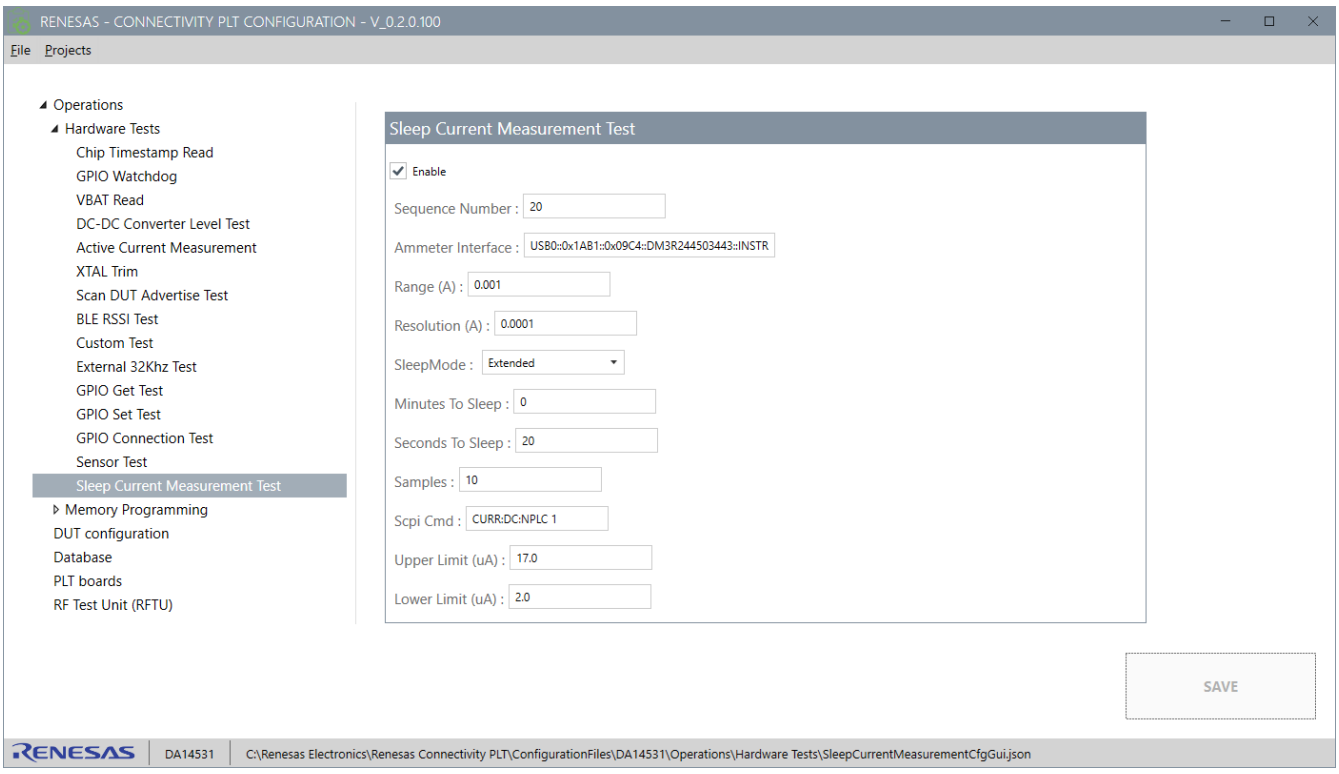



**Figure 75. Sleep Current Measurement Test configuration**

Configuration details:

- Ammeter Interface – The NI VISA address of the instrument (can be obtained by the NI Max application).
- Range – The set range of the instrument. The actual range values depends on the instrument. Typical values are 1 µA, 10 µ A, 100 µ A, 1 mA, 10 mA, 100 mA, 1 A, 3 A, 10 A. The value must be entered in ampere. Default value is 1 mA. This range must be able to accommodate the sleep current of all DUTs connected to a single Connectivity PLT board.
- Resolution – The set resolution for the instrument. The actual resolution depends on the instrument. The acceptable values are between 0 and 7 decimal places. Note that many instruments do not use this command and the range sets the resolution as well.
- Sleep mode – Sleep mode depends on the CPU supported modes, typically an active, extended and deep sleep mode is support. Current consumption differs between these modes.
- Minutes/Seconds to sleep – The time the device is set to wake up after switching to sleep mode. The CPU of the DUTs uses a timer interrupt to wake up since for the sleep operation most I/O is commonly disabled including the reset pin. These settings also determine the length of time after which the measurement is made which happens just before the device wakes up.
- Sample – The number of samples to be taken and averaged.
- SCPI CMD – Additional SCPI command, usually specific to the instrument used. The default CURR:DC:NPLC 1 sets the sampling type to one NPLC cycle.
- Upper/Lower limits – Sets the pass/fail criteria for the sleep current value per DUT.

## 8.3.17 Power Rail Test

Power Rail Test is available in RA6W1, RA6W2, RRQ6100x, and RRQ6105x.
The aim of this test is to verify that the power supplies of the chip function properly. There are seven power domains: VDDIO1, VDDIO2, VDDIO_FDIO, VBAT, VDDD, DCDC_PA_OUT, and DCDC_ANA_OUT.

### 8.3.17.1 DC-DC Analog

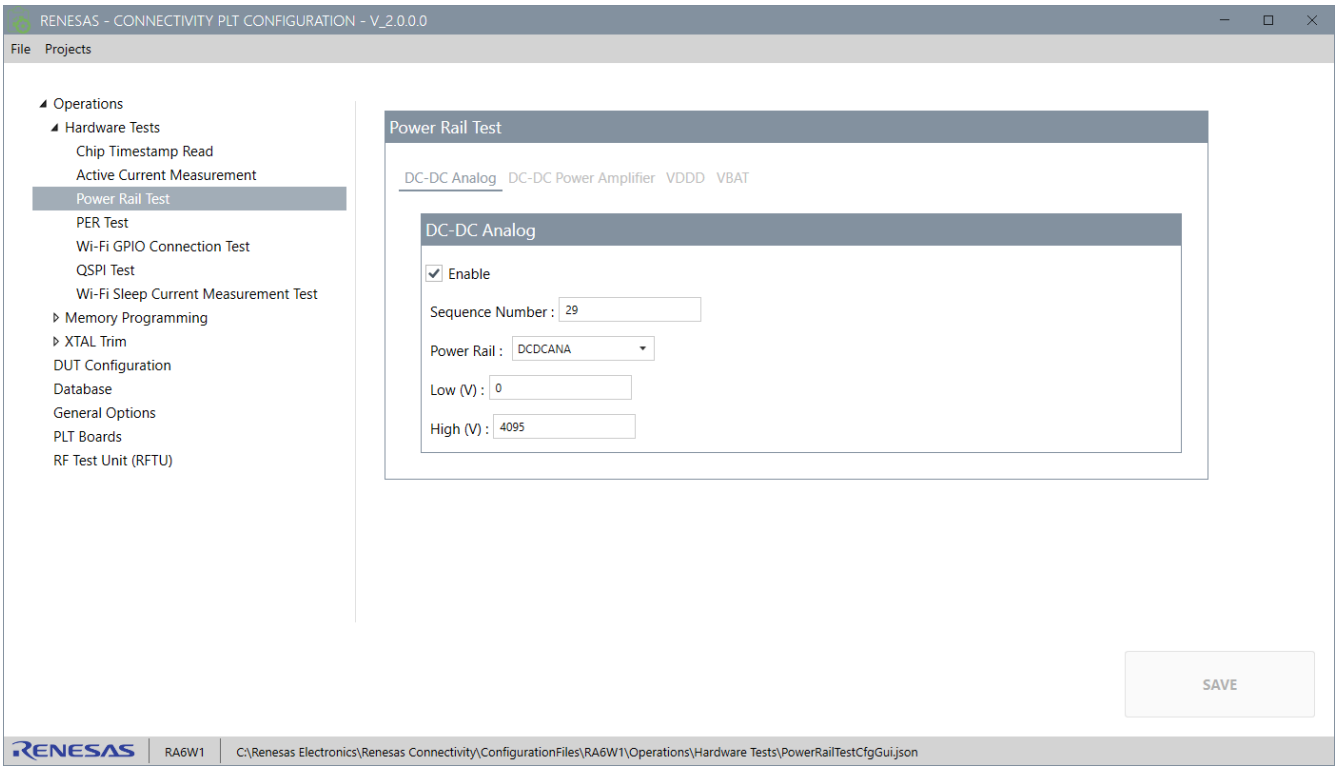



**Figure 76. DC-DC Analog configuration**

**Table 30. DC-DC Analog parameters**

| Parameter | Description |
|---|---|
| Enable | This enables the specific memory reading operation |
| Sequence Number | Order in which the test is run in execution |
| Power Rail | Power rail to configure |
| ADC Value Read – Min Limit | Minimum accepted value of ADC |
| ADC Value Read – Max Limit | Maximum accepted value of ADC |

### 8.3.17.2 DC-DC Power Amplifier

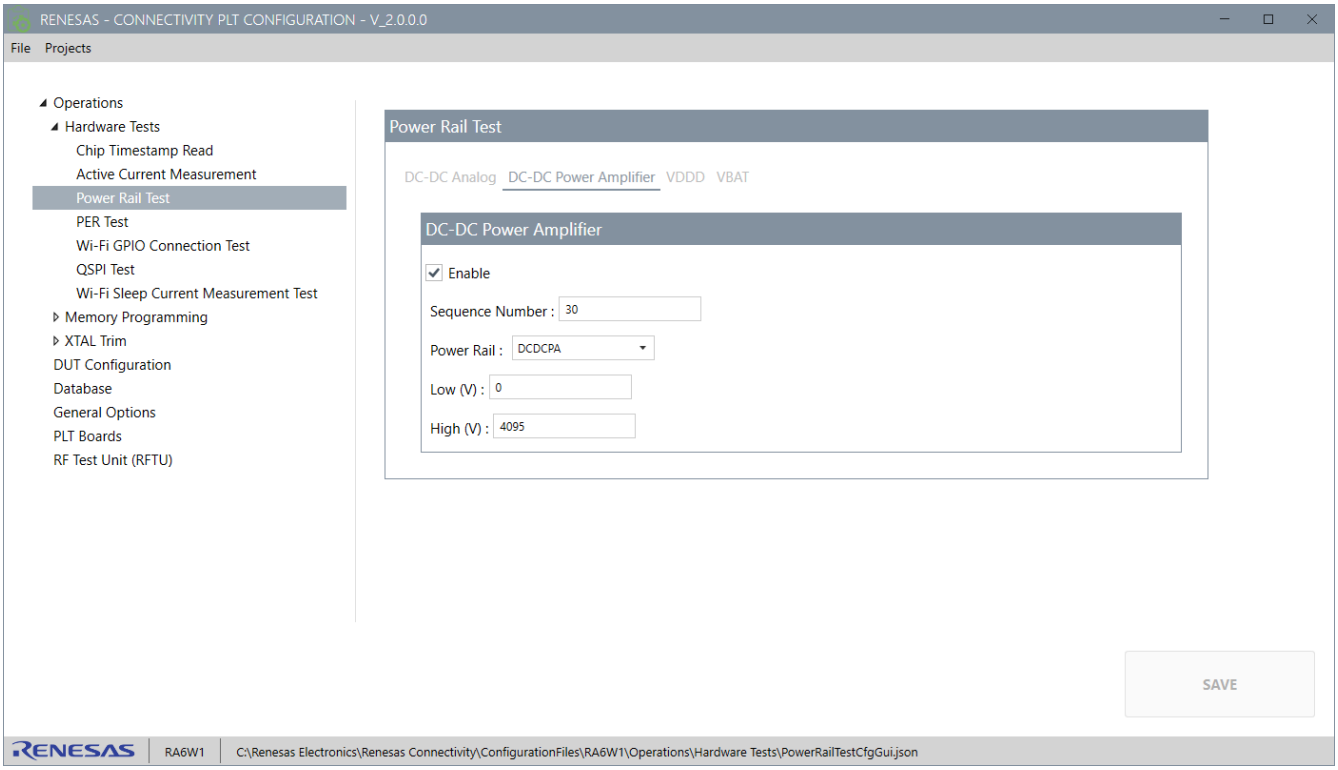



**Figure 77. DC-DC Power Amplifier configuration**

**Table 31. DC-DC Power Amplifier parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific operation |
| Sequence Number | Order in which the test is run in execution |
| Power Rail | Power rail to configure |
| ADC Value Read – Min Limit | Minimum accepted value of ADC |
| ADC Value Read – Max Limit | Maximum accepted value of ADC |

### 8.3.17.3 DC-DC VDDD

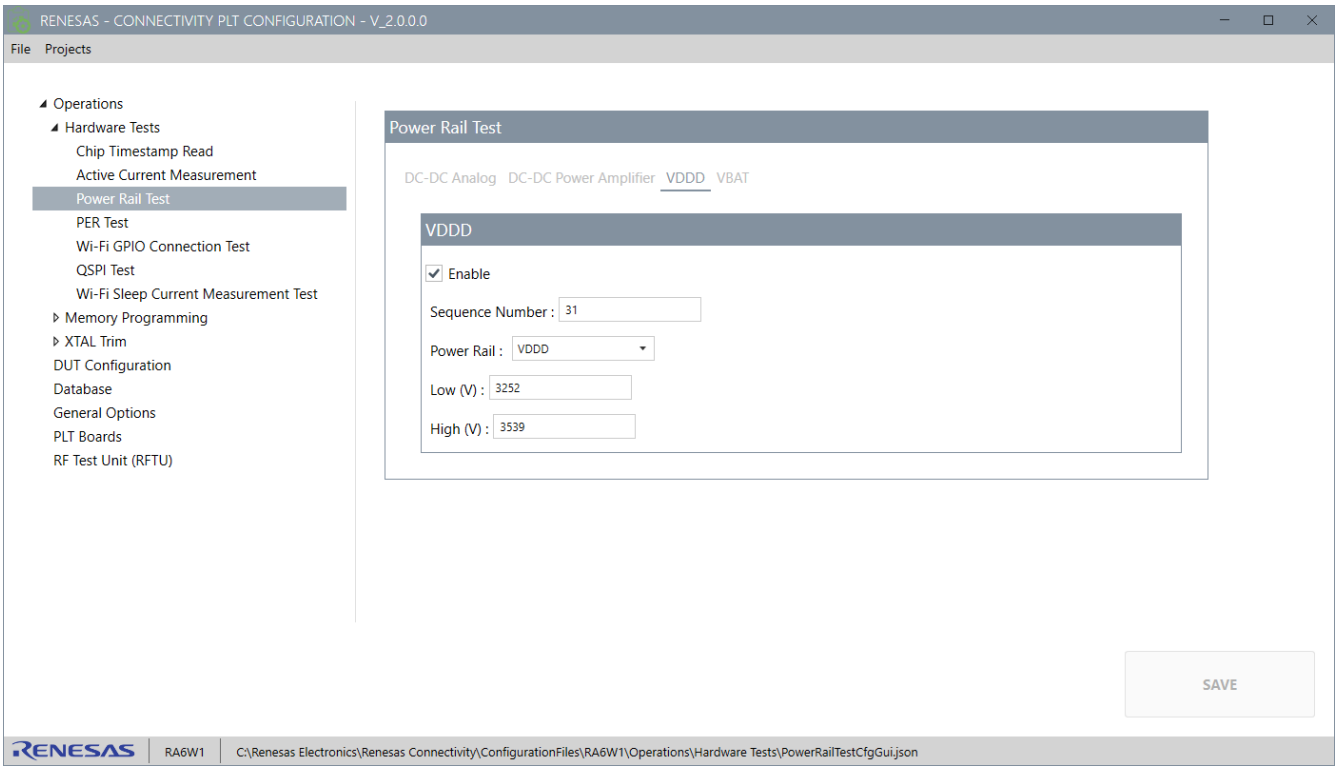



**Figure 78. DC-DC VDD configuration**

**Table 32. DC-DC VDDD parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific operation |
| Sequence Number | Order in which the test is run in execution |
| Power Rail | Power rail to configure |
| Low (V) | Minimum accepted value - Min Limit |
| High (V) | Maximum accepted value - Max Limit |

### 8.3.17.4 DC-DC VBAT

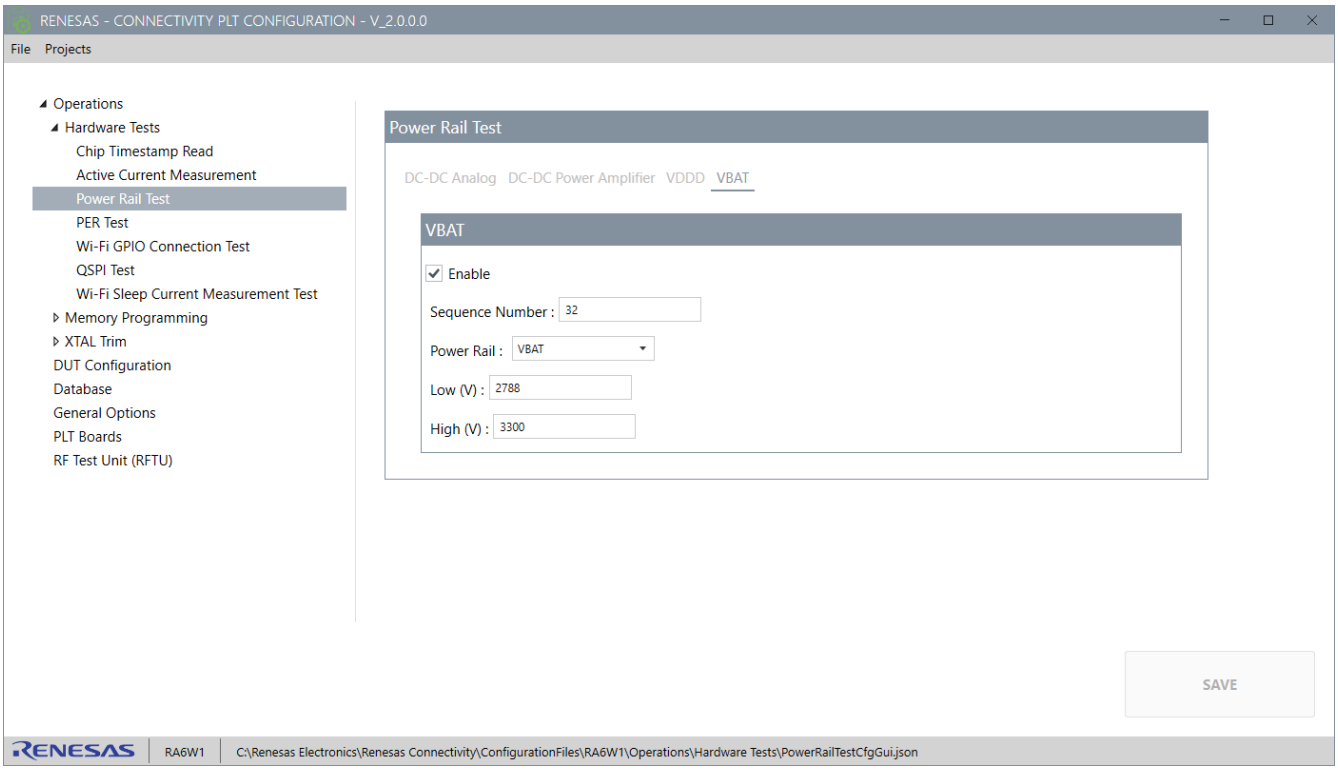



**Figure 79. DC-DC VBAT configuration**

**Table 33. DC-DC VBAT parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific operation |
| Sequence Number | Order in which the test is run in execution |
| Power Rail | Power rail to configure |
| Low (V) | Minimum accepted value – Min Limit |
| High (V) | Maximum accepted value – Max Limit |

### 8.3.18 Packet Error Rate (PER) Test

Packet Error Rate (PER) Test is available in RA6W1, RA6W2, RRQ6100x, and RRQ6105x.

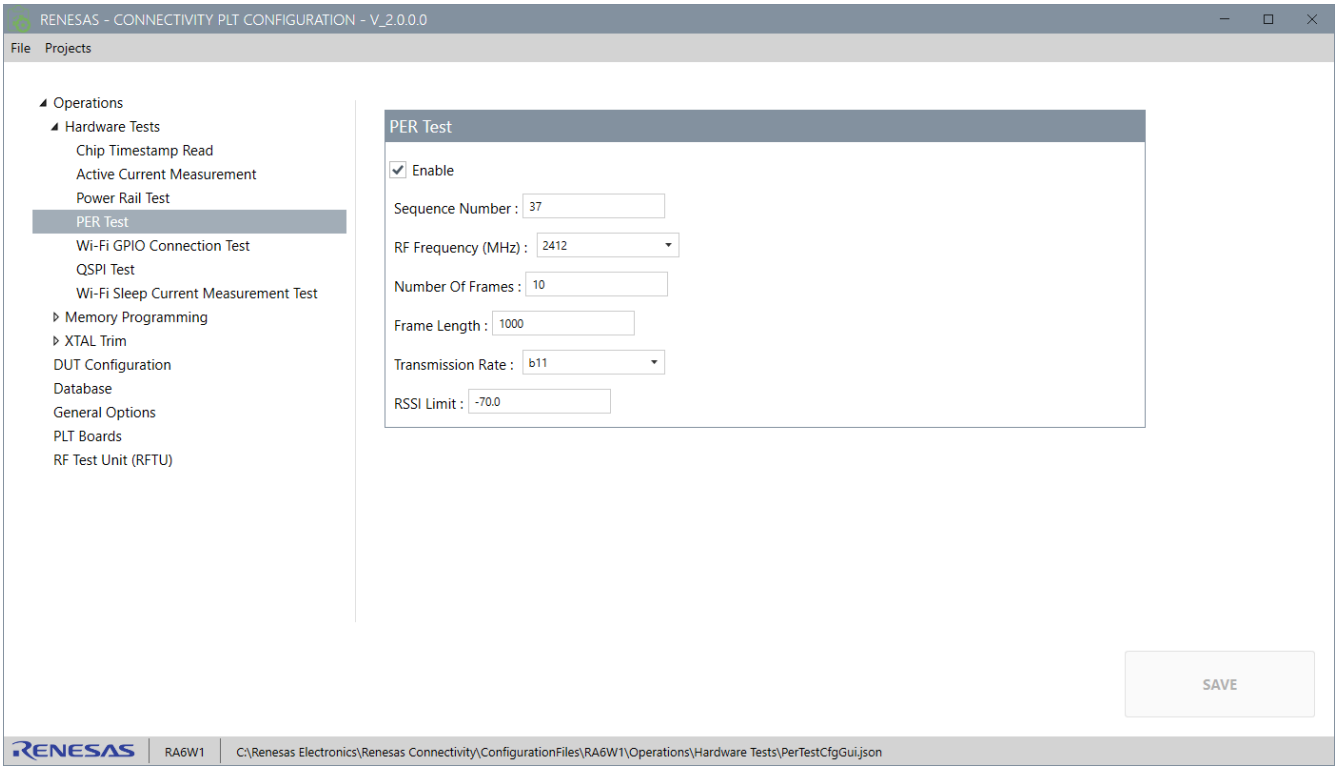



**Figure 80. PER Test configuration**

**Table 34. PER Test parameters**

| Option | Description |
|---|---|
| Enable | Enables the specific memory reading operation |
| RF Frequency (MHz) | The RF frequency in MHz |
| Number of Frames | The number of frames |
| Frame Length | The length of the frame |
| Transmission Rate | Transmission rate |
| RSSI Limit | RSSI limit |

### 8.3.19 Read Power Values

Read Power Values test is available in DA14850 and DA14870.

The aim of this test is to measure the power supplies of the chip to verify proper function. Table 35 summarizes the available measurements.

**Table 35. Power Values measurements**

| Name | Description |
|---|---|
| Battery Voltage | Battery voltage |
| Battery Charging Current | Battery charging current |
| Battery Discharging Current | Battery discharging current |
| Charger Voltage | Charger voltage |
| Charger Current | Charger current |
| SIMO 1 Voltage | Radio supply (VDDRF - 1.05 V) |

| Name | Description |
|---|---|
| SIMO 2 Voltage | Digital core supply (VDDD, 0.65-0.8 V) |
| SIMO 3 Voltage | I/O reference and external peripheral supply (Codec) (VDDIO18 - 1.8 V) |
| SIMO 4 Voltage | I/O reference and external peripheral supply (Codec) (VDDIO12 - 1.2 V) |
| LDO 1 Voltage | RF Power Amplifier supply (VDDRFPA - 0.8 V) |
| LDO 2 Voltage | Memory supply voltage (VCS - 0.8 V) |
| LDO 3 Voltage | Analog core supply (VDDA_18 - 1.8 V) |
| LDO 4 Voltage | USB PHY block supply (VDDU - 3.3 V) |

All Read Power Values options share the same configuration parameters.

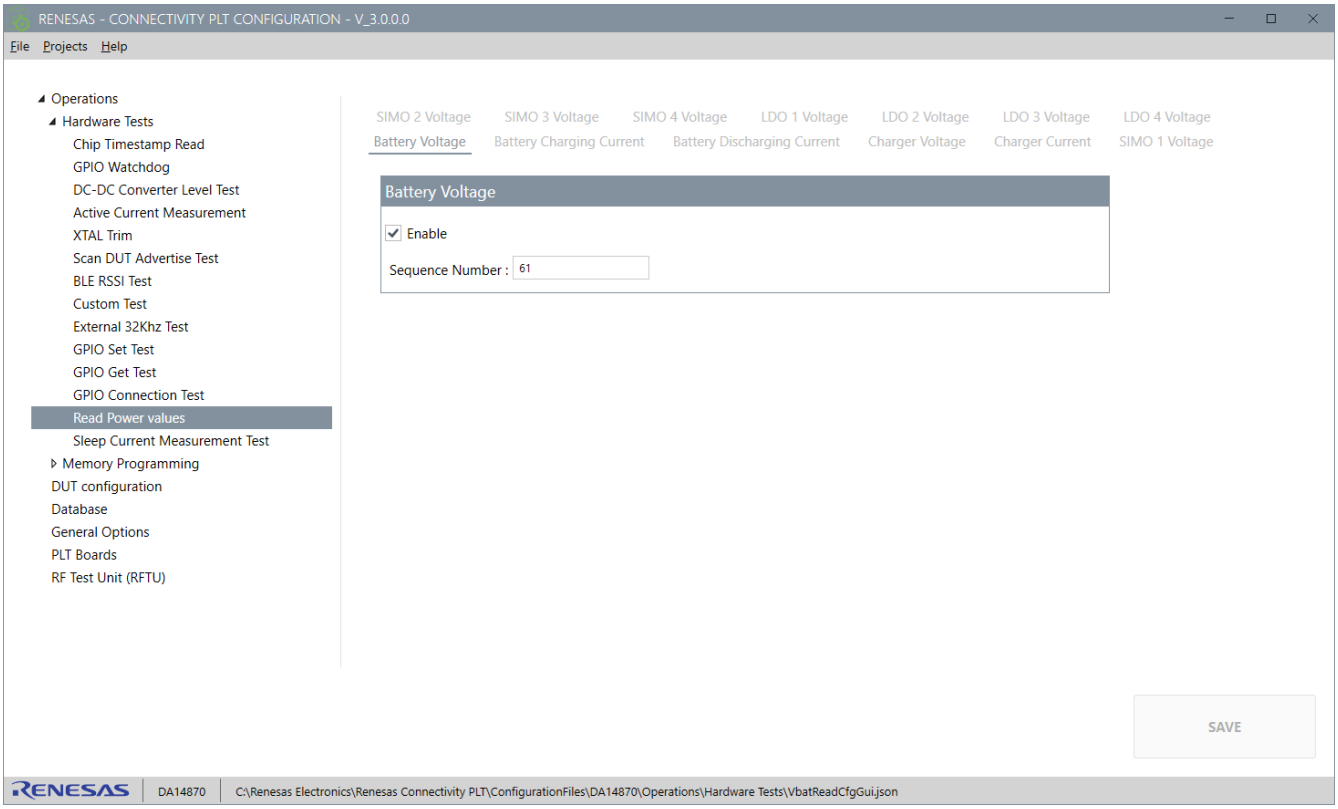



**Figure 81. Read Power Values configuration**

**Table 36. Read Power Values parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific operation. |
| Sequence Number | Order in which the test is run in execution. |

## 8.4 Memory Programming Configuration

The memory programming group contains operations for reading and writing to different types of memories internal and external to the MCU. In addition to simple binary read/write operations, special memory areas can be configured and programmed. Table 37 summarizes all memory programing operations.

**Table 37. Types of memory tests**

| Operation type | Description |
|---|---|
| UART Baud Rate Set | Sets the baud rate of the UART for all memory operations. See Section 8.4.1. |
| SPI Flash Initialize | Initializes the external flash memory controller which internally defines the pins used and other settings as configured in the peripheral configuration page. See Section 8.4.2. |
| SPI Flash Erase All | Performs a chip erase on the flash memory. See Section 8.4.3. |
| SPI Flash Erase Block | Erases specific memory blocks. See Section 8.4.4. |
| SPI Flash Write Binary | Writes an entire binary to flash memory. See Section 8.4.5. |
| SPI Flash Read | Reads the flash memory with the specified settings. See Section 8.4.6. |
| EEPROM Memory Write | Writes an entire binary to an external EEPROM memory. See Section 8.4.7. |
| EEPROM Memory Read | Reads data from external EEPROM memory. See Section 8.4.8. |
| OTP Memory Write | Writes an entire binary to the OTP memory. See Section 8.4.9. |
| OTP Memory Read | Reads data from the OTP memory. See Section 8.4.10. |
| Header Write | Writes in the special memory area of OTP Header. See Section 8.4.11. |
| Configuration Script Write | Writes entries to the configuration script area of the OTP. See Section 8.4.12. |
| Custom Memory Data Write | Writes data to Flash or OTP memory that are stored in the devices database and are specific for each serial number. See Section 8.4.13. |
| Common Memory Data Write | Writes data to Flash or OTP memory that are stored in the devices database. See Section 8.4.14. |
| QSPI Flash Block Erase | Erases specific QSPI flash memory blocks. See Section 8.4.15. |
| QSPI Flash Write Binary | Writes an entire binary to QSPI flash memory. See Section 8.4.16. |
| QSPI Flash Read | Reads the QSPI flash memory with the specified settings. See Section 8.4.17. |
| QSPI Flash Erase All | Performs a chip erase on the QSPI flash memory. See Section 8.4.18. |
| QSPI Test | Gets the JEDEC ID from QSPI flash memory. See Section 8.4.19. |
| Wi-Fi Configuration Script Write | Writes entries to the configuration script area of the OTP. See Section 8.4.20. |
| QSPI Flash Initialize | Reads flash info and initializes QSPI flash memory. See Section 8.4.21. |
| QSPI Flash Write Image | Writes an IntelHex application image file to QSPI flash memory. See Section 8.4.22. |
| QSPI Flash Write COLA Image | Writes an IntelHex COLA firmware image file to QSPI flash memory. See Section 8.4.23. |
| OQSPI Flash Erase All | Performs a chip erase on the OQSPI flash memory. See Section 8.4.24. |
| OQSPI Flash Block Erase | Erases specific OQSPI flash memory blocks. See Section 8.4.25. |
| OQSPI Flash Write Binary | Writes an entire binary to OQSPI flash memory. See Section 8.4.26. |
| OQSPI Flash Read | Reads the OQSPI flash memory with the specified settings. See Section 8.4.27. |
| OQSPI Flash State | Performs a chip erase on the OQSPI flash memory. See Section 8.4.28. |
| NVS Initialize | Initializes NVS protocol. See section 8.4.29. |
| NVS Defaults | Sets NVS to the factory settings. See section 8.4.30. |
| NVS Set Mode | Sets operation region. See section 8.4.31. |
| NVS Get Mode | Gets operation region. See section 8.4.32. |

| Operation type | Description |
|---|---|
| NVS Get RFPI/IPEI | Gets Device Identity. See section 8.4.33. |
| NVS Set Memory | Sets parameters in NVS memory. See section 8.4.34. |
| NVS Read Memory | Reads parameters from NVS memory. See section 8.4.35. |
| NVS Set Image Configuration | Sets run image. See section 8.4.36. |

### 8.4.1 UART Baud Rate Set

UART Baud Rate Set is available in DA1453x.

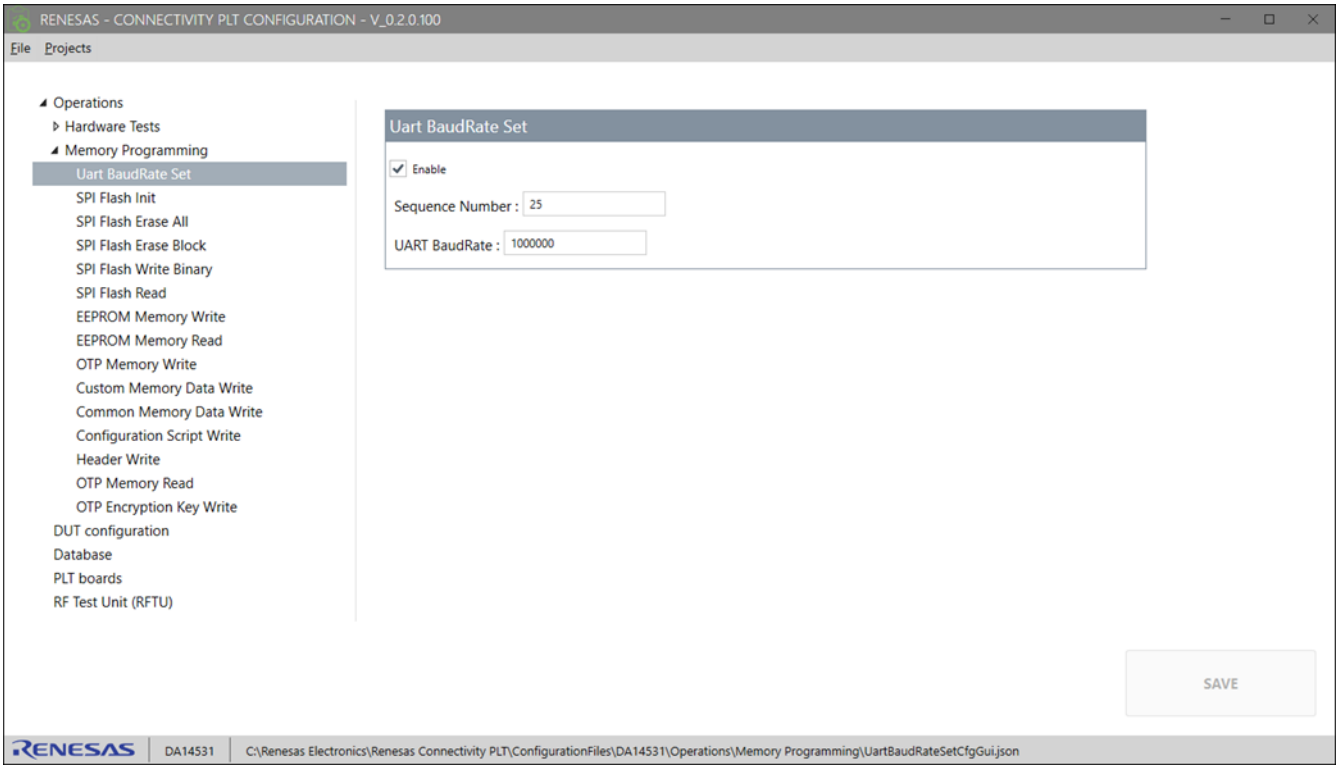



**Figure 82. UART Baud Rate Set configuration**

**Table 38. UART Baud Rate Set parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the UART Set baud rate operation |
| Sequence Number | Order in which the test is run in execution |
| UART Baud Rate | Baud rate to set up in bps. Supported values: 9600, 19200, 57600, 115200, 1000000.<br>Note: If the default 115200 is used, the operation does not need to be enabled. |

### 8.4.2 SPI Flash Initialize

SPI Flash Initialize is available in DA1453x.

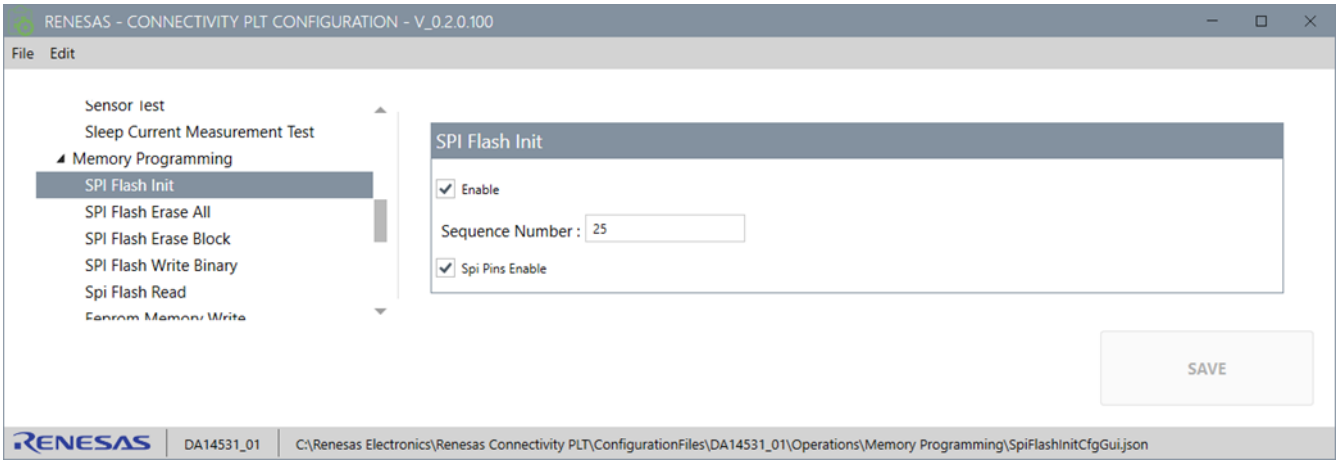



**Figure 83. SPI Flash Initialize configuration**

**Table 39. SPI Flash Initialize parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the SPI Flash Initialize operation |
| Sequence Number | Order in which the test is run in execution |
| SPI Pins Enable | Enables the SPI pins |

### 8.4.3 SPI Flash Erase All

SPI Flash Erase All is available in DA1453x.

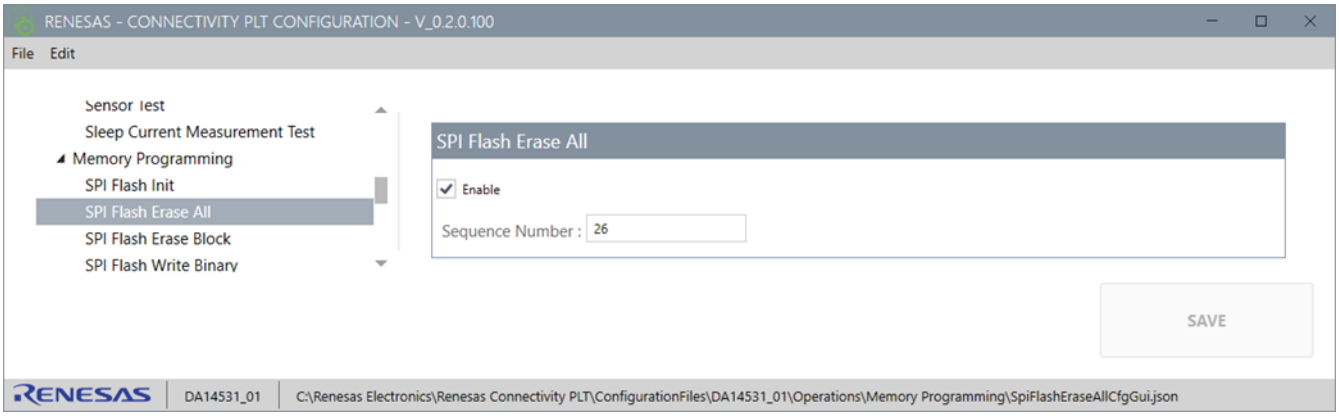



**Figure 84. SPI Flash Erase All configuration**

**Table 40. SPI Flash Erase All parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the SPI flash chip erase operation |
| Sequence Number | Order in which the test is run in execution |

### 8.4.4 SPI Flash Erase Block

SPI Flash Erase Block is available in DA1453x.

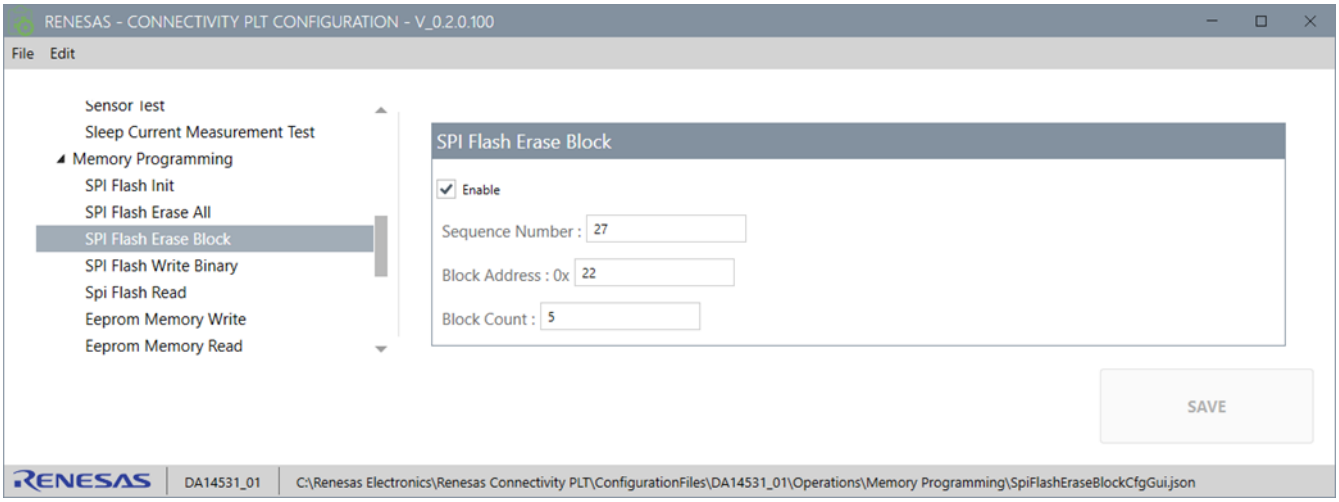



**Figure 85. SPI Flash Erase Block configuration**

**Table 41. Flash Erase Block parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific memory erase test |
| Sequence Number | Order in which the test is run in execution |
| Block Address | Address of the first block |
| Block Count | Number of memory blocks to erase |

### 8.4.5 SPI Flash Write Binary

SPI Flash Write Binary is available in DA1453x.

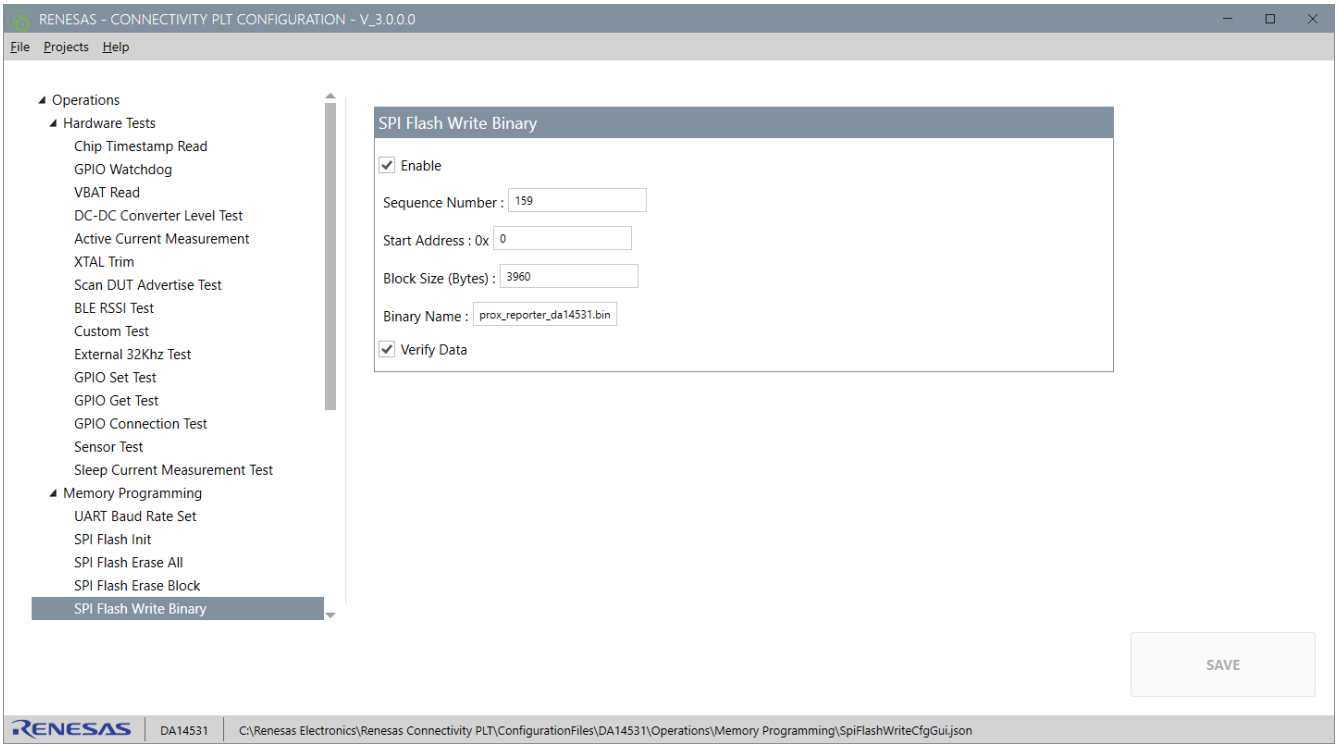



**Figure 86. SPI Flash Write Binary configuration**

**Table 42. Flash Write Binary parameters**

| Parameter | Description |
|---|---|
| Enable | Enables specific flash image programming operation |
| Sequence Number | Order in which the test is run in execution |
| Start Address | Flash start address the image is written to |
| Block Size | Chunk size for writing |
| Verify Data | By selecting this option, the PLT software reads back the contents of the flash memory and compares them to the original image file. If these do not match, the memory programming fails. |
| Binary Name | The filename of the binary to be written.<br>Note: All binary files need to be stored in the Firmware folder under the installation folder and registered in the specific MCU JSON file. In any other case, full file path should be provided. |

### 8.4.6 SPI Flash Read

SPI Flash Read is available in DA1453x.

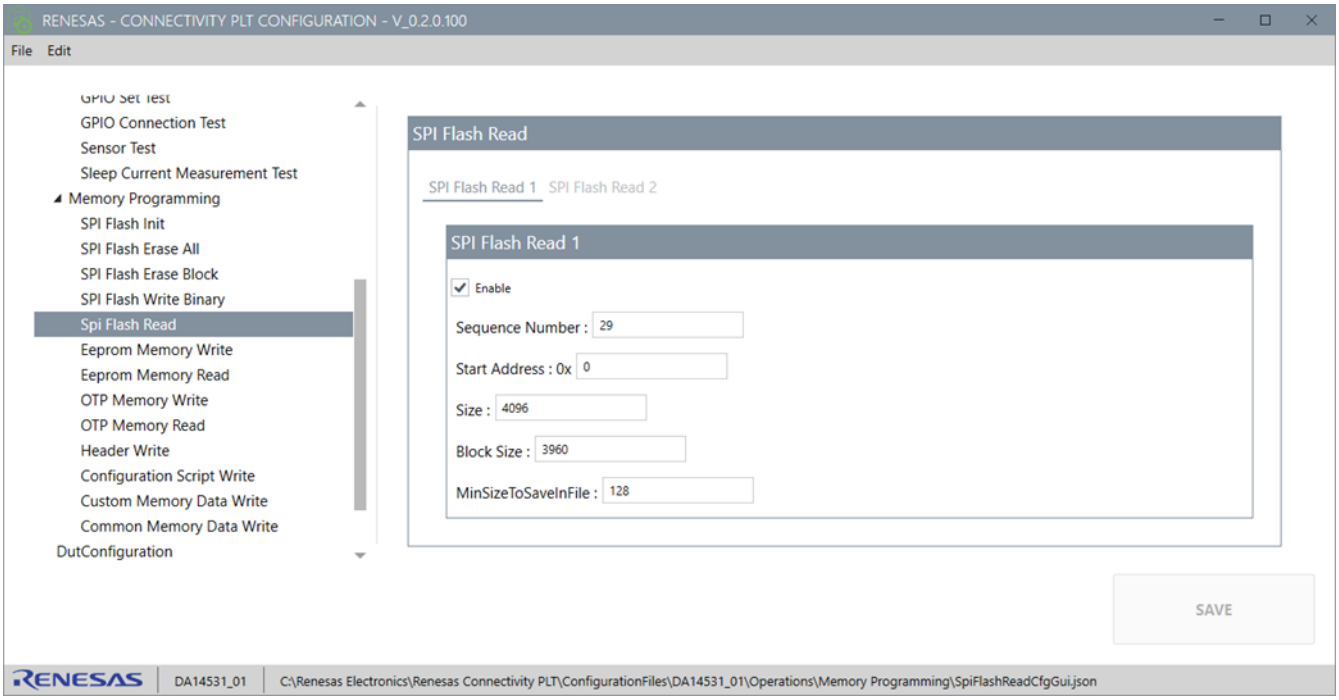



**Figure 87. SPI Flash Read configuration**

**Table 43. SPI Flash Read parameters**

| Parameter | Description |
|---|---|
| Enable | Enables specific memory reading operation |
| Sequence Number | Order in which the test is run in execution |
| Start address | Address the read operation begins from |
| Size | Number of bytes to read, up to 64 MB |
| Block Size | Size of the chunk that the data is split during reading |
| Minimum Size to Save in File | Minimum number of bytes required for the read operation to create a file. The file is stored under `logs\folder mem_read_test` in the Connectivity PLT installation folder. If no file is created, the data is available in the logs files. |

### 8.4.7 EEPROM Memory Write

EEPROM Memory Write is available in DA1453x.

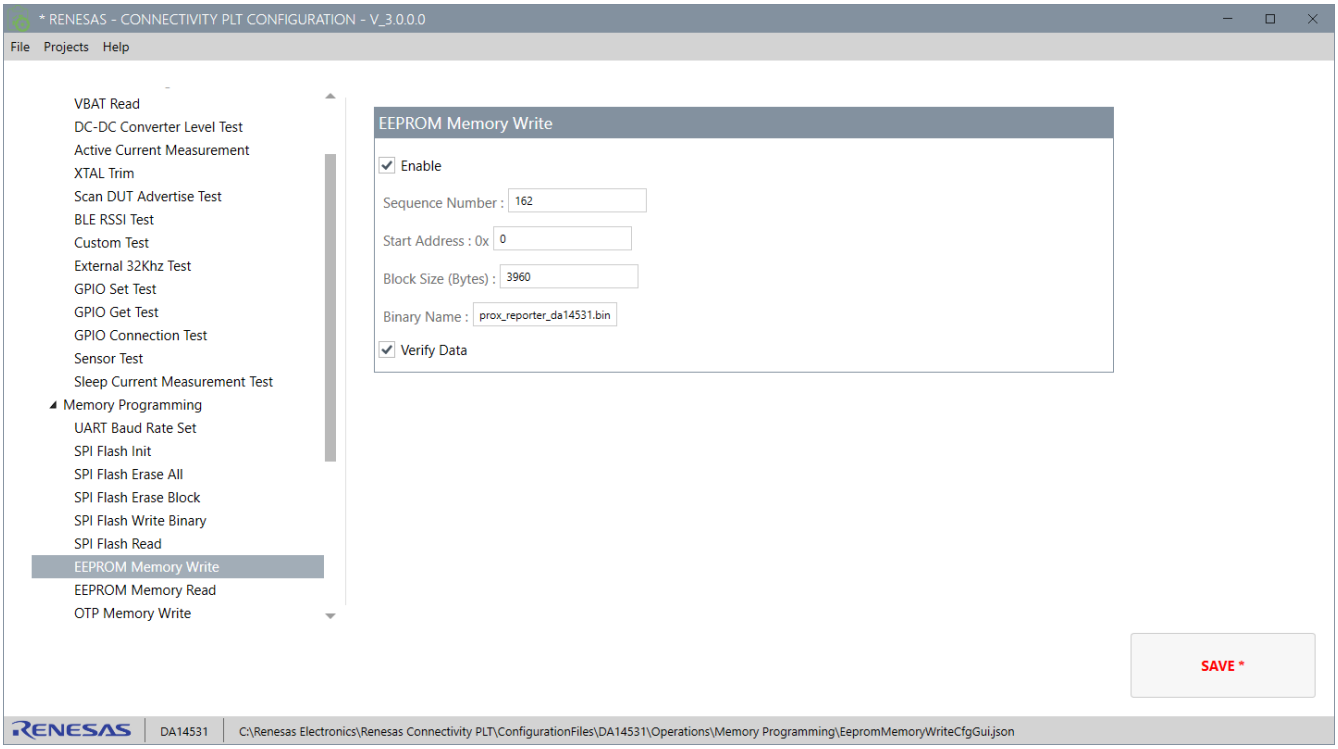



**Figure 88. EEPROM Memory Write configuration**

**Table 44. EEPROM Memory Write parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific memory reading operation |
| Sequence Number | Order in which the test is run in execution |
| Start address | Start address in hexadecimal format |
| Block Size | Size of the chunk that the data is split during writing |
| Binary Name | Filename of the binary file.<br>Note: All binary files need to be stored in the Firmware folder under the installation folder and registered in the specific MCU JSON file. |
| Verify Data | By selecting this option, the PLT software reads back the contents of the Eeprom memory and compares them to the original image file. If these do not match, the memory programming fails. |

### 8.4.8 EEPROM Memory Read

EEPROM Memory Read is available in DA1453x.

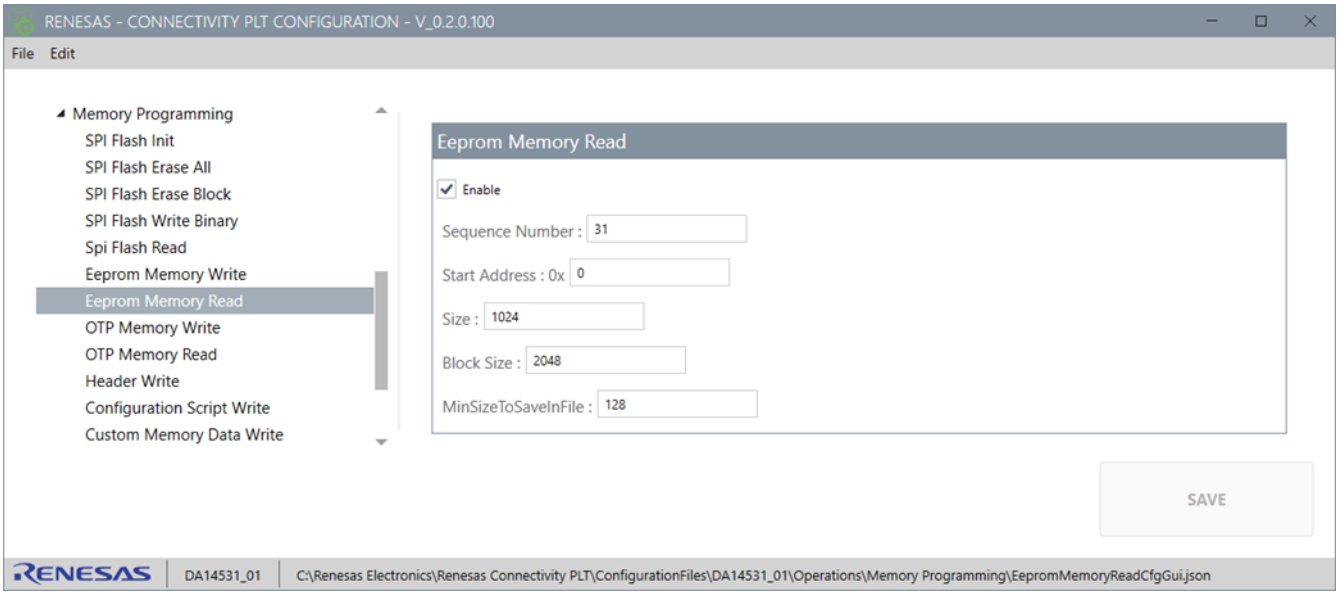



**Figure 89. EEPROM Memory Read configuration**

**Table 45. EEPROM Memory Read parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific memory reading operation |
| Sequence Number | Order in which the test is run in execution |
| Start address | Start address in hexadecimal format |
| Size | Number of bytes to read, up to 64 MB |
| Block Size | Size of the chunk that the data is split during reading |
| MinSizeToSaveInFile | Minimum number of bytes required for the read operation to create a file. The file is stored under `logs\folder mem_read_test` in the Connectivity PLT installation folder. If no file is created, the data is available in the logs files. |

### 8.4.9 OTP Memory Write

OTP Memory Write is available in DA1453x, RA6W1, RA6W2, RRQ6100x, RRQ6105x, DA14850 and DA14870.

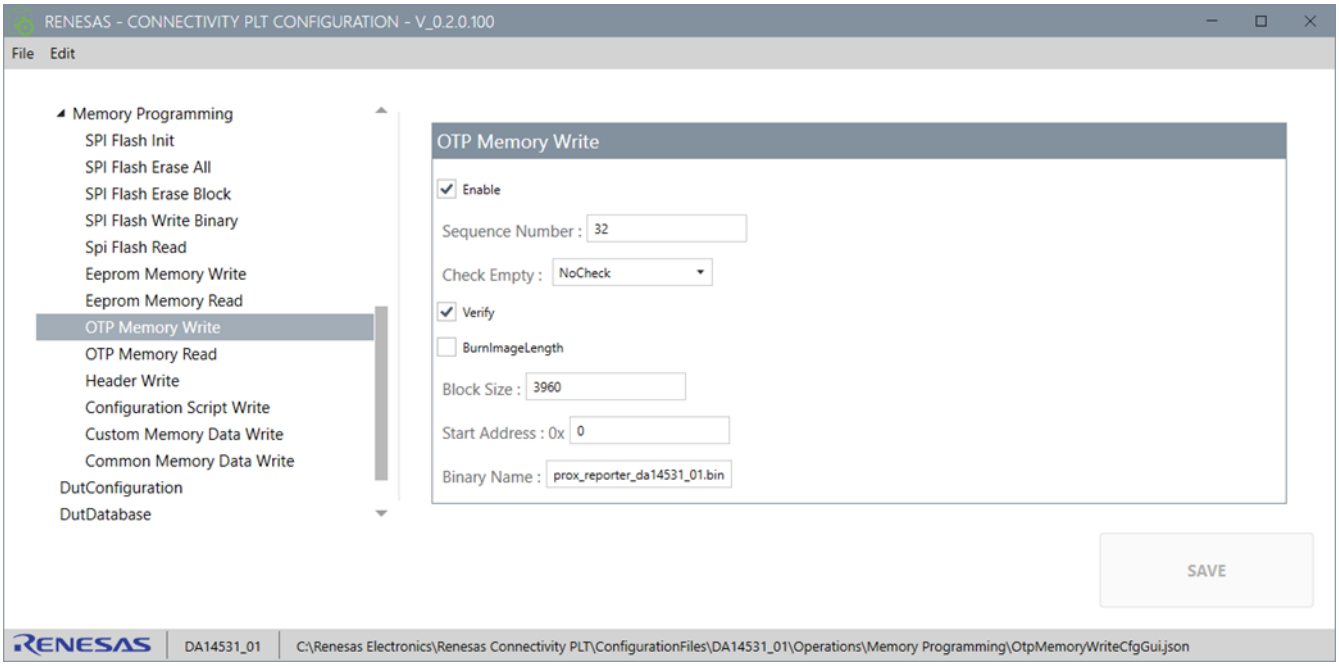



**Figure 90. OTP Memory Write configuration**

**Table 46. OTP Memory Write parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific memory reading operation |
| Sequence Number | Order in which the test is run in execution |
| Check Empty | Available options:<br>▪ NoCheck: Writes the data without reading first.<br>▪ ErrorIfNotEmpty: Fails the operation if the OTP area to be written is not empty.<br>▪ PassifNotEmpty: Skips the writing<br>▪ PassifMatch: Fails the operation if the contents of the OTP differ from the binary file. |
| Verify | When selected, reads back the contents of the OTP memory and compares them to the original image file. If these do not match, the operation fails. |
| Burn Image Length | Writes the size of the image to the OTP Header |
| Block Size | Size of the chunk that the data is split during the operation |
| Start Address | Start address in hexadecimal format |
| Binary Name | The filename of the binary to be written.<br>Note: All binary files should be stored in the Firmware folder under the installation folder and registered in the specific MCU JSON file. In not, full file path should be provided. |

### 8.4.10 OTP Memory Read

OTP Memory Read is available in DA1453x, RA6W1, RA6W2, RRQ6100x, RRQ6105x, DA14850 and DA14870.

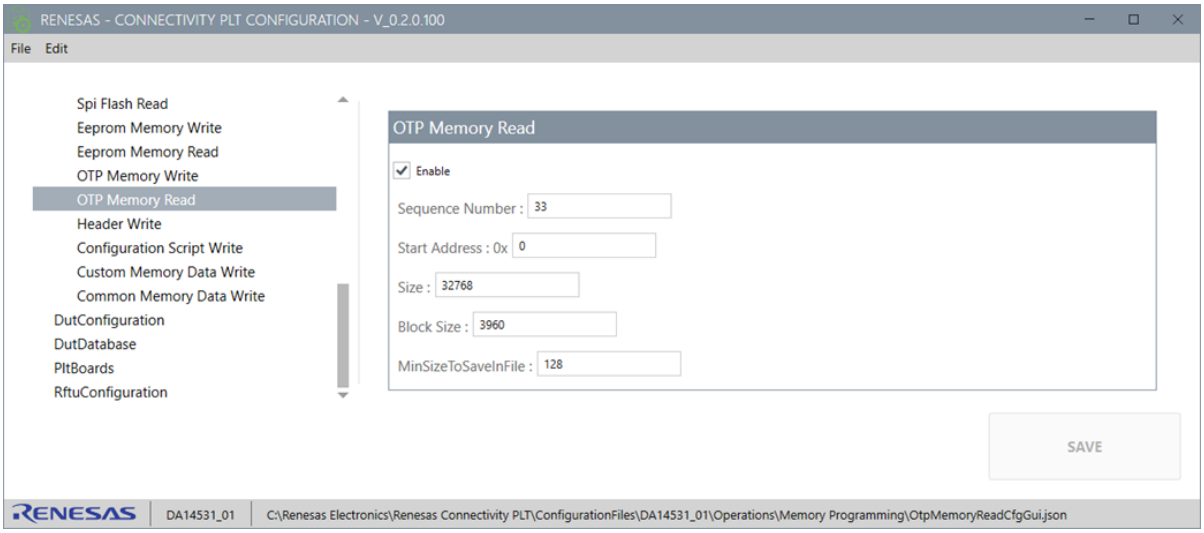



**Figure 91. OTP Memory Read configuration**

**Table 47. OTP Memory Read parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the specific memory reading operation |
| Sequence Number | Order in which the test is run in execution |
| Start address | Start address in hexadecimal format |
| Size | Number of bytes to read, up to 64 MB |
| Block Size | Size of the chunk that the data is split during the operation |
| Minimum Size to Save in File | Minimum number of bytes required for the read operation to create a file. The file is stored under logs\folder mem_read_test in the Connectivity PLT installation folder. If no file is created the data is available in the logs files. |

### 8.4.11 Header Write

Header Write is available in DA1453x, RA6W1, RRQ6100x.

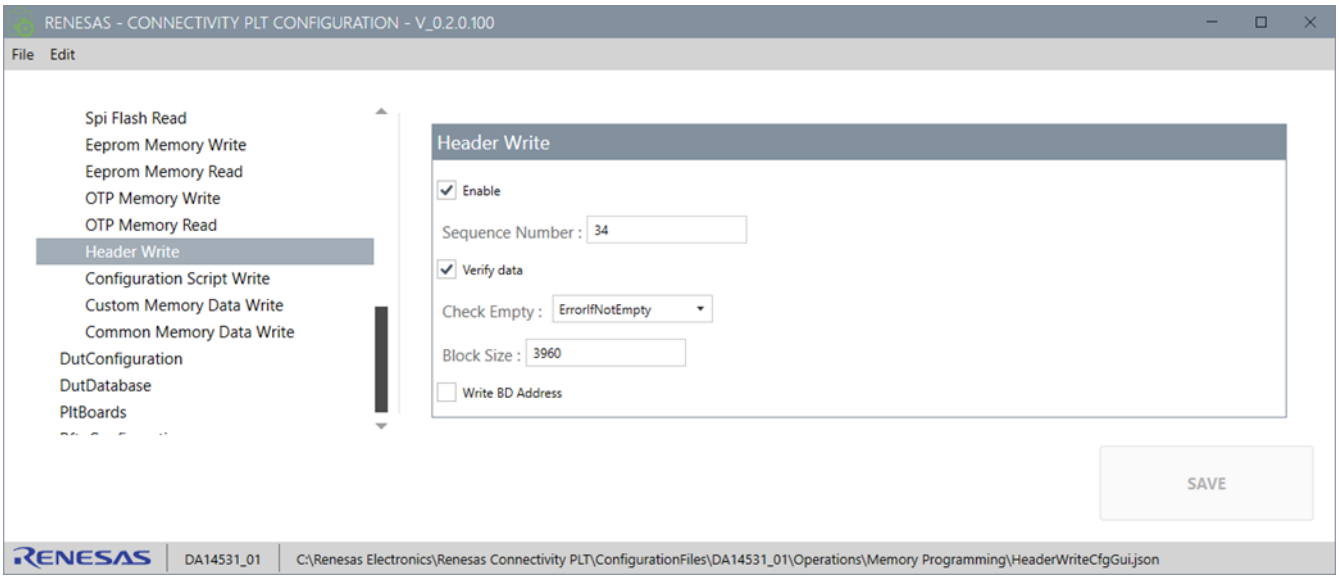



**Figure 92. Header Write configuration**

**Table 48. Header Write parameters**

| Parameter | Description |
|---|---|
| Enable | This enables the header write operation |
| Sequence Number | The order in which the test is run in execution |
| Verify Data | When selected, reads back the contents of the flash memory and compares them to the original image file. If these do not match, the memory programming fails. |
| Check Empty | Available options:<br>▪ NoCheck: Writes the data without reading first.<br>▪ ErrorIfNotEmpty: Fails the operation if the OTP area to be written is not empty.<br>▪ PassifNotEmpty: Skips the writing and report the operation as pass.<br>▪ PassifMatch: Fails the operation if the contents of the OTP differ from the one set to write. |
| Block Size | Size of the chunk that the data is split during the operation |
| Write DB Address | Enables BD address writing in header area |

## 8.4.12 Configuration Script Write

Configuration Script Write is available in DA1453x, RA6W1, RRQ6100x, DA14850 and DA14870.

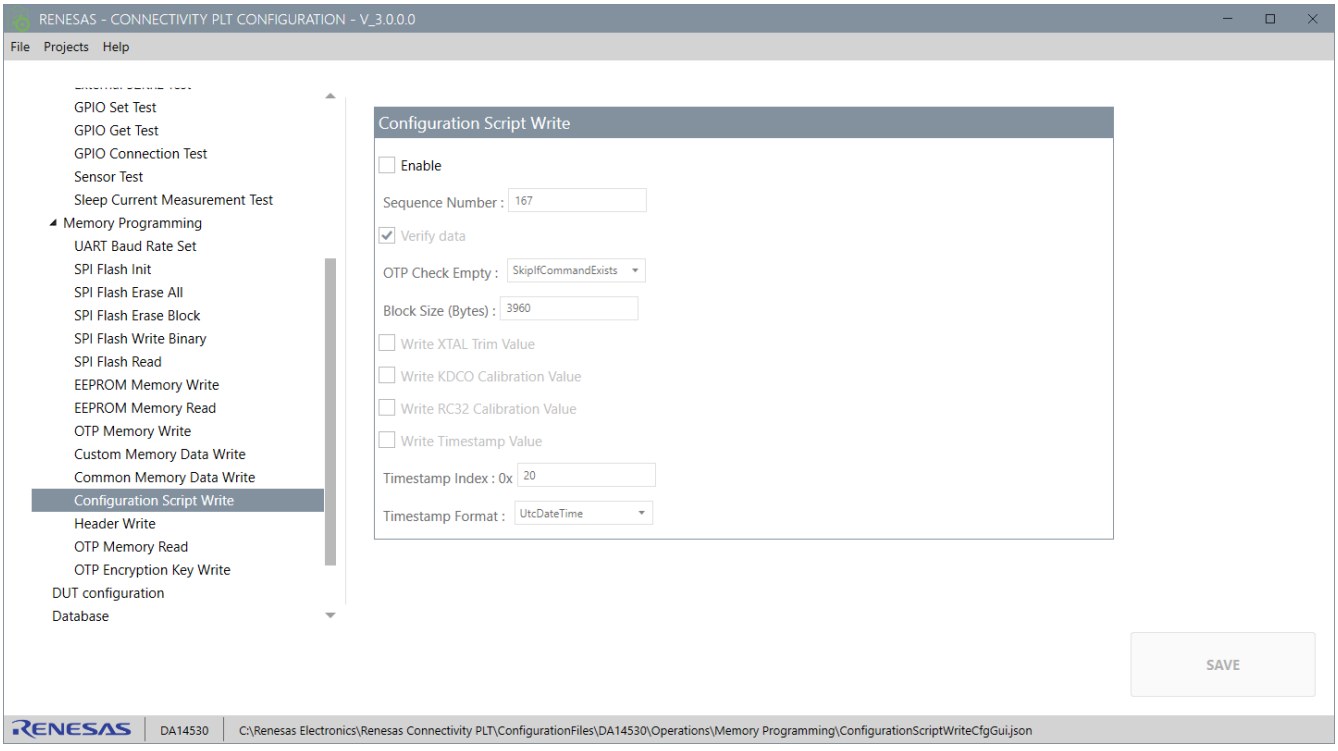



**Figure 93. Configuration Script Write**

**Table 49. Configuration Script Write parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the Configuration Script write operation |
| Sequence Number | Order in which the test is run in execution |
| Verify Data | When selected, reads back the contents of the flash memory and compares them to the original image file. If these do not match, the memory programming fails. |
| OTP Check Empty | Available options:<br>▪ No Check: Writes the data without reading first. |

| Parameter | Description |
|---|---|
| | ▪ ErrorIfCommandExists: Fails the operation if the CS entry command already exists in the CS area.<br>▪ SkipIfCommandExists: Skips the command-value set if it already exists in the CS area.<br>▪ SkipIfTotalMatch: Fails the operation if the command-value set does not match what is already stored in CS area. |
| Block Size | Size of the chunk that the data is split during the operation |
| Write XTAL Trim Value | Write the XTAL Trim Value (acquired from XTAL trim test) |
| Write KDCO Calibration Value | Write the KDCO calibration value (acquired from the KDCO calibration test) |
| Write RC32 Calibration Value | Write the RC32 calibration value (acquired from the RC32 calibration test) |
| Write Timestamp Value | Write the timestamp value |
| Timestamp Index | Index in the CS area to write the timestamp |
| Timestamp Format | The format of the timestamp to write (UTC Date Time/UNIX Epoch) |

### 8.4.13 Custom Memory Data Write

Custom Memory Data Write is available in DA1453x, RA6W1, RA6W2, RRQ6100x, RRQ6105x, DA14850 and DA14870.

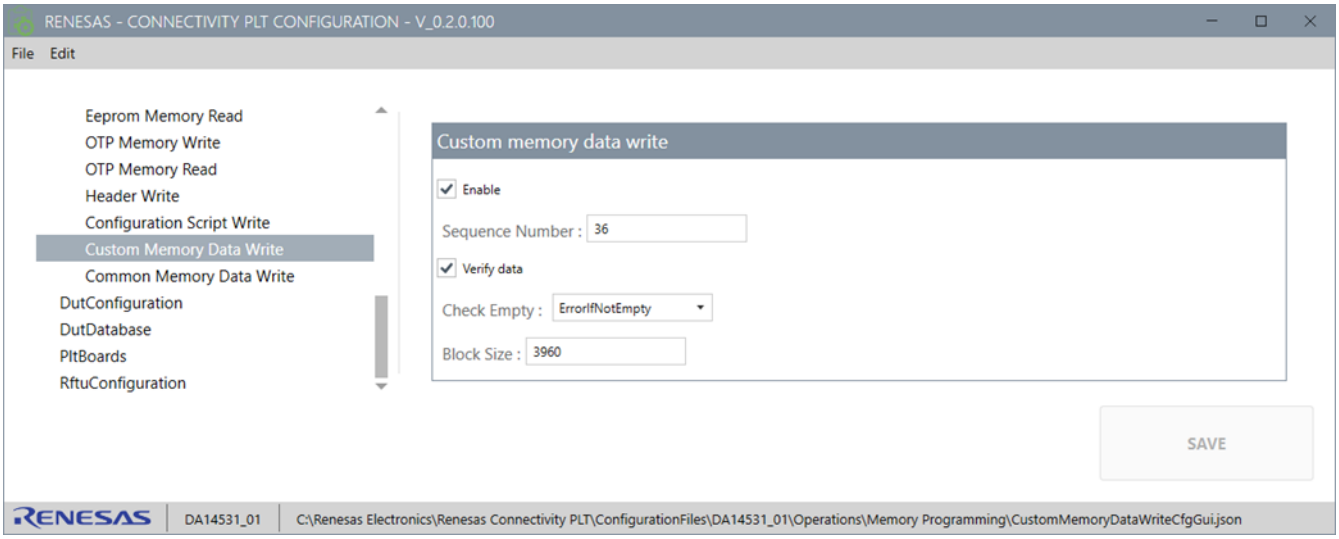



**Figure 94. Custom Memory Write configuration**

**Table 50. Custom Memory Write parameters**

| Parameter | Description |
|---|---|
| Enable | Enables custom memory data write operation |
| Sequence Number | Order in which the test is run in execution |
| Verify Data | When selected, reads back the contents of the flash memory and compares them to the original image file. If these do not match, the memory programming fails. |
| Check Empty | Not implemented for this operation – Set to No Check |
| Block Size | Size of the chunk that the data is split during the operation |

### 8.4.14 Common Memory Data Write

Common Memory Data Write is available in DA1453x, RA6W1, RA6W2, RRQ6100x, RRQ6105x, DA14850 and DA14870.

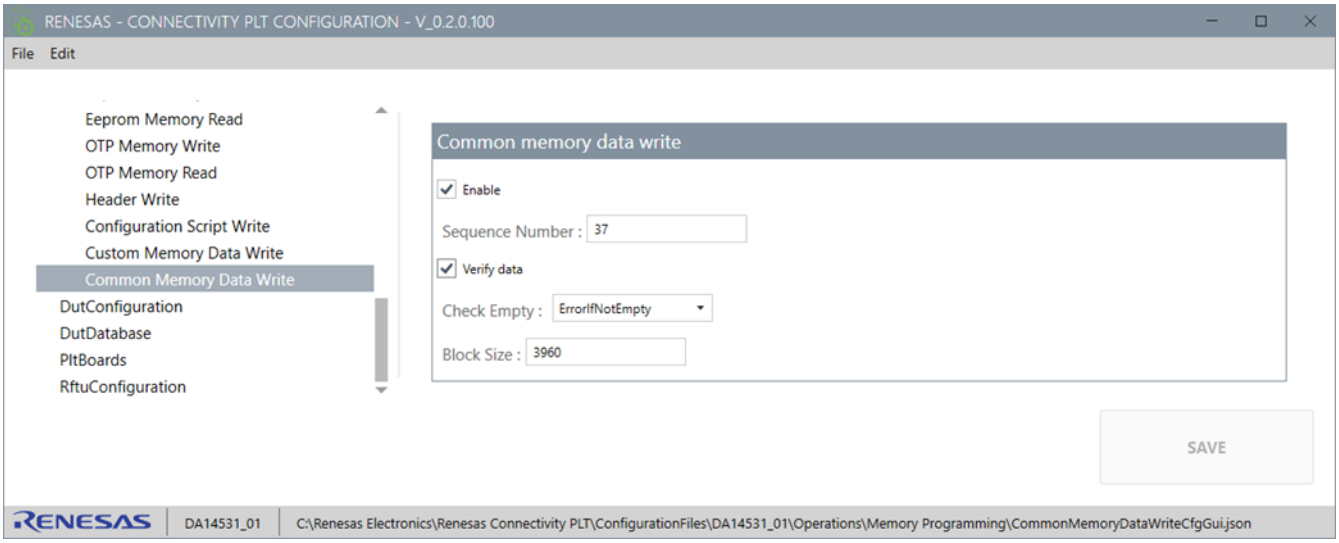



**Figure 95. Common Memory Write configuration**

**Table 51. Common Memory Write parameters**

| Parameter | Description |
|---|---|
| Enable | Enables common memory data write operation |
| Sequence Number | Order in which the test is run in execution |
| Verify Data | When selected, reads back the contents of the flash memory and compares them to the original image file. If these do not match, the memory programming fails. |
| Check Empty | Not implemented for this operation – Set to No Check |
| Block Size | Size of the chunk that the data is split during the operation |

### 8.4.15 QSPI Flash Block Erase

QSPI Flash Block Erase is available in RA6W1, RA6W2, RRQ6100x, RRQ6105x, DA14495, and DA14AVDDECT.

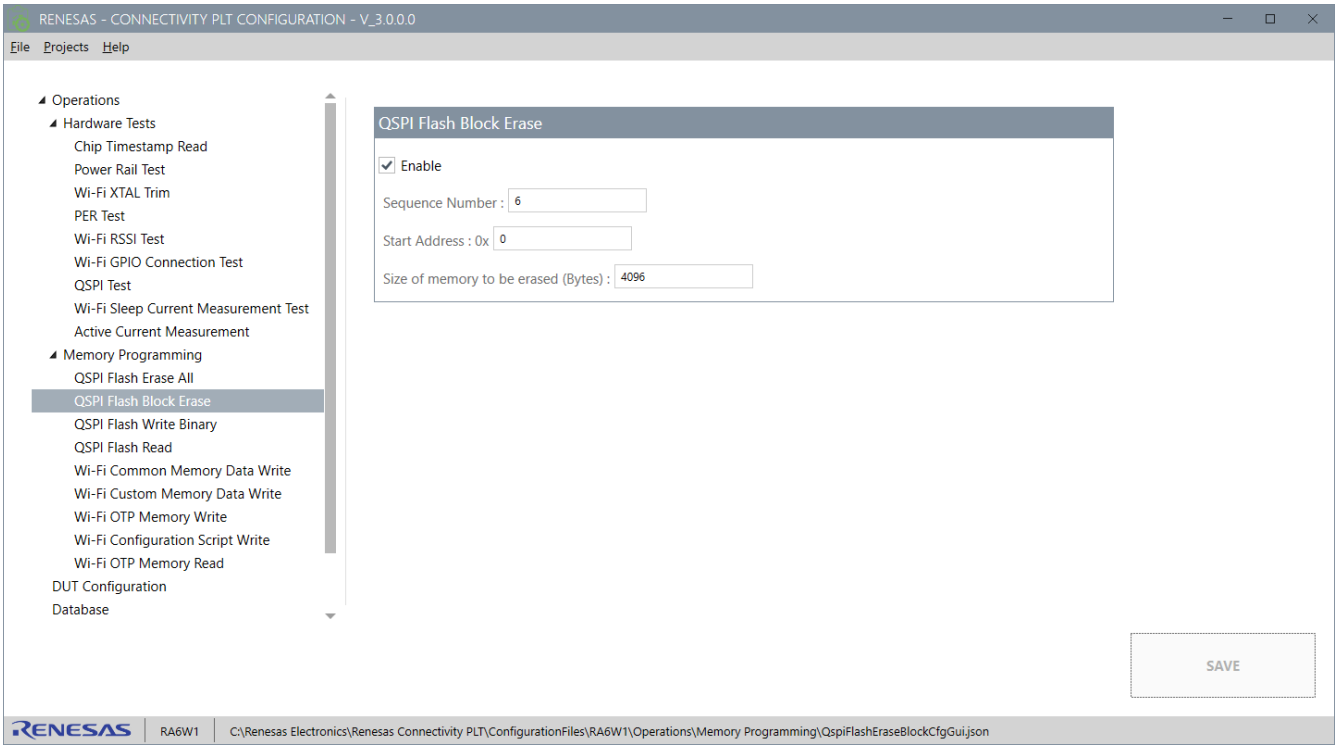



**Figure 96. QSPI Flash Block Erase configuration**

**Table 52. QSPI Flash Block Erase parameters**

| Parameter | Description |
|---|---|
| Enable | Enables QSPI Flash Block Erase operation |
| Sequence Number | Order in which the test is run in execution |
| Start Address | Start address in hexadecimal format |
| Size of memory to be erased (Bytes) | Number of bytes to erase |

### 8.4.16 QSPI Flash Write Binary

QSPI Flash Write Binary is available in RA6W1, RA6W2, RRQ6100x, and RRQ6105x.

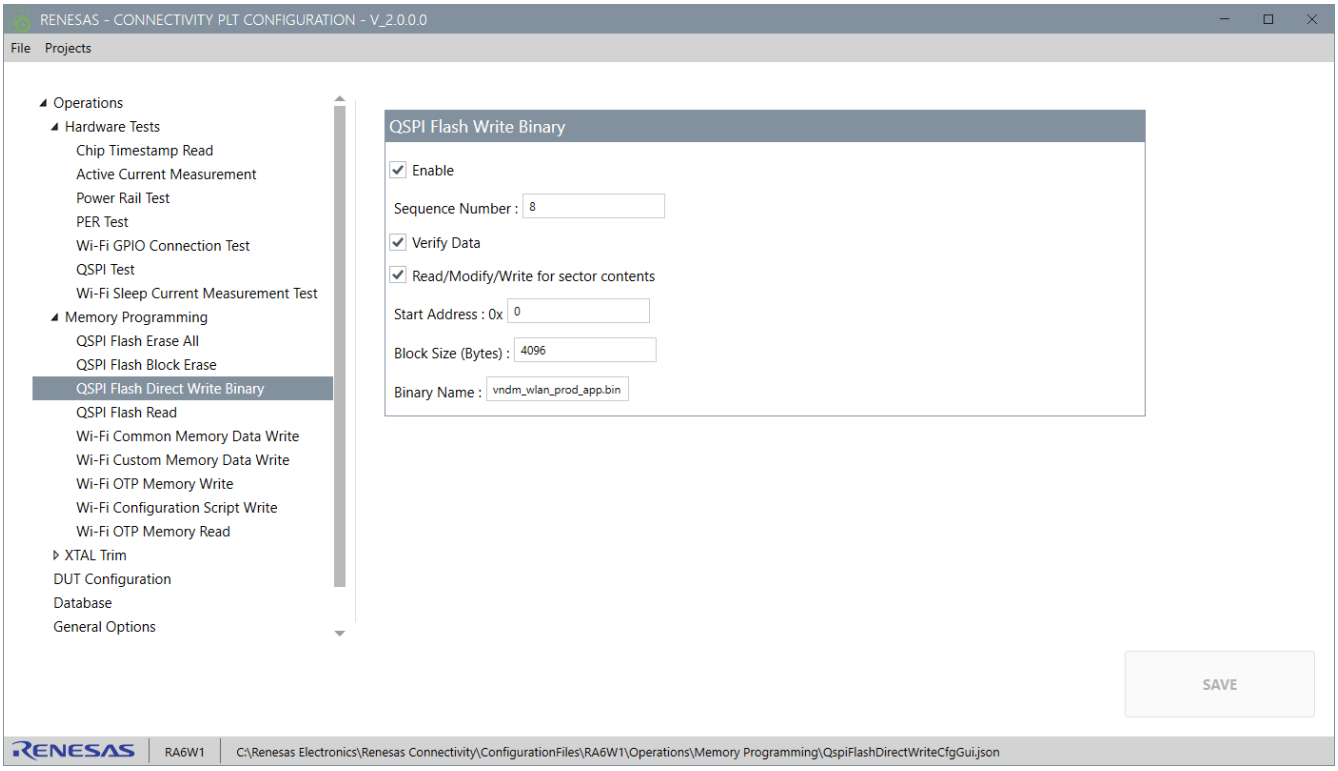



**Figure 97. QSPI Flash Write configuration**

**Table 53. QSPI Flash Write parameters**

| Parameter | Description |
|---|---|
| Enable | Enables QSPI Flash Write operation |
| Sequence Number | Order in which the test is run in execution |
| Verify Data | When selected, reads back the contents of the flash memory and compares them to the original image file. If these do not match, the memory programming fails. |
| Read/Modify/Write for Sector Contents | Add Read/Write/Modify to sector contents |
| Start Address | Start address in hexadecimal format |
| Block Size | Block size in bytes |
| Binary Name | Binary name |

### 8.4.17 QSPI Flash Read

QSPI Flash Read is available in RA6W1, RA6W2, RRQ6100x, and RRQ6105x.

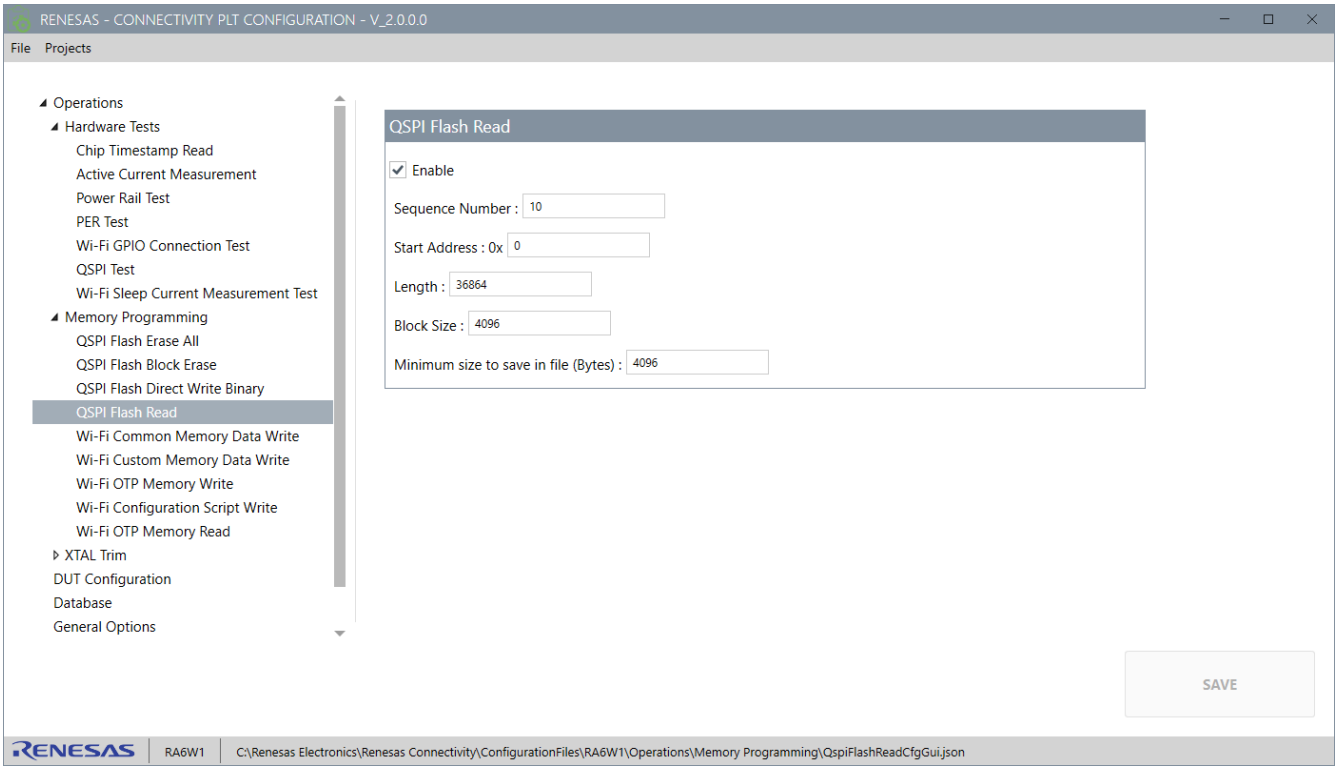



**Figure 98. QSPI Flash Read configuration**

**Table 54. QSPI Flash Read parameters**

| Parameter | Description |
|---|---|
| Enable | Enables QSPI Flash Read operation |
| Sequence Number | Order in which the test is run in execution |
| Start Address | Start address in hexadecimal format |
| Length | Length of read |
| Block Size | Block size in bytes |
| Minimum Size to Save in File (Bytes) | Minimum size in bytes |

### 8.4.18 QSPI Flash Erase All

QSPI Flash Erase All is available in RA6W1, RA6W2, RRQ6100x, and RRQ6105x.

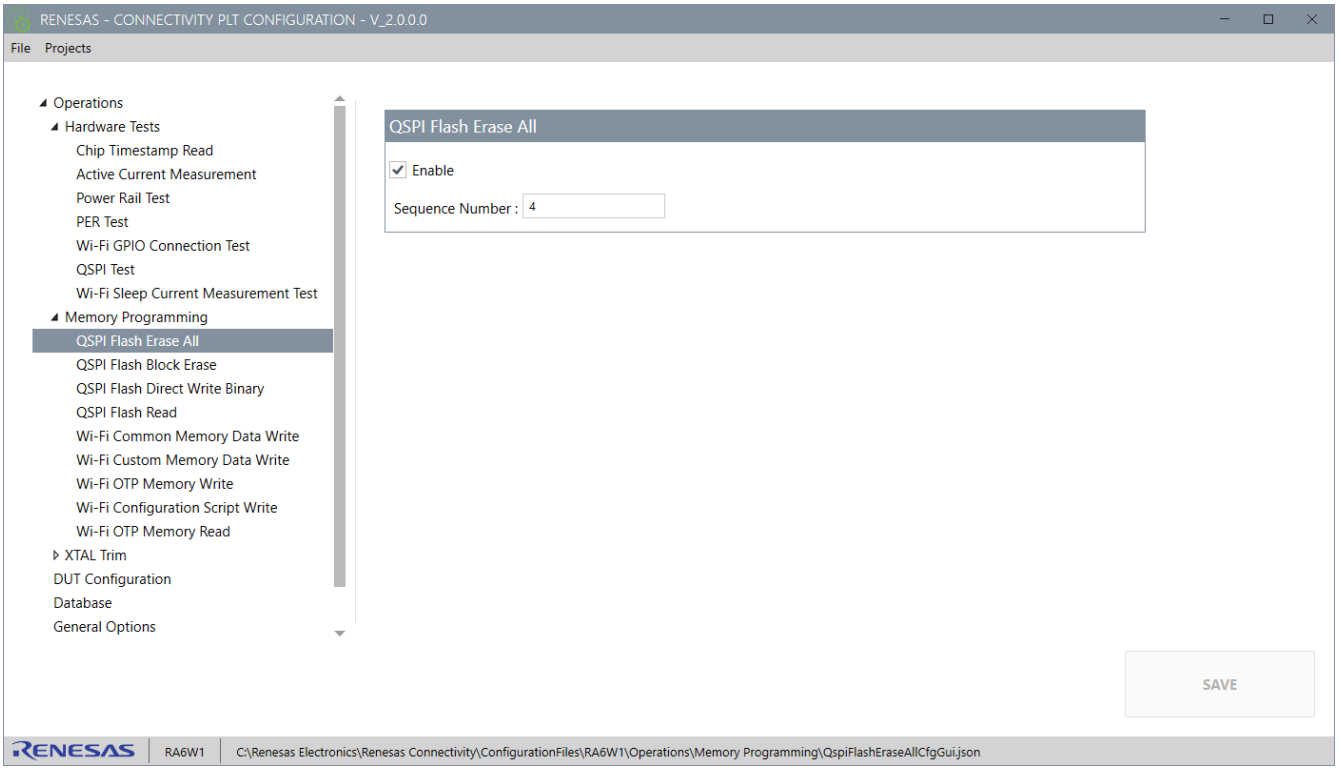



**Figure 99. QSPI Flash Erase All configuration**

**Table 55. QSPI Erase All parameters**

| Parameter | Description |
|---|---|
| Enable | Enables QSPI Flash Erase All operation |
| Sequence Number | Order in which the test is run in execution |

### 8.4.19 QSPI Test

QSPI Test is available in RA6W1, RA6W2, RRQ6100x, and RRQ6105x.

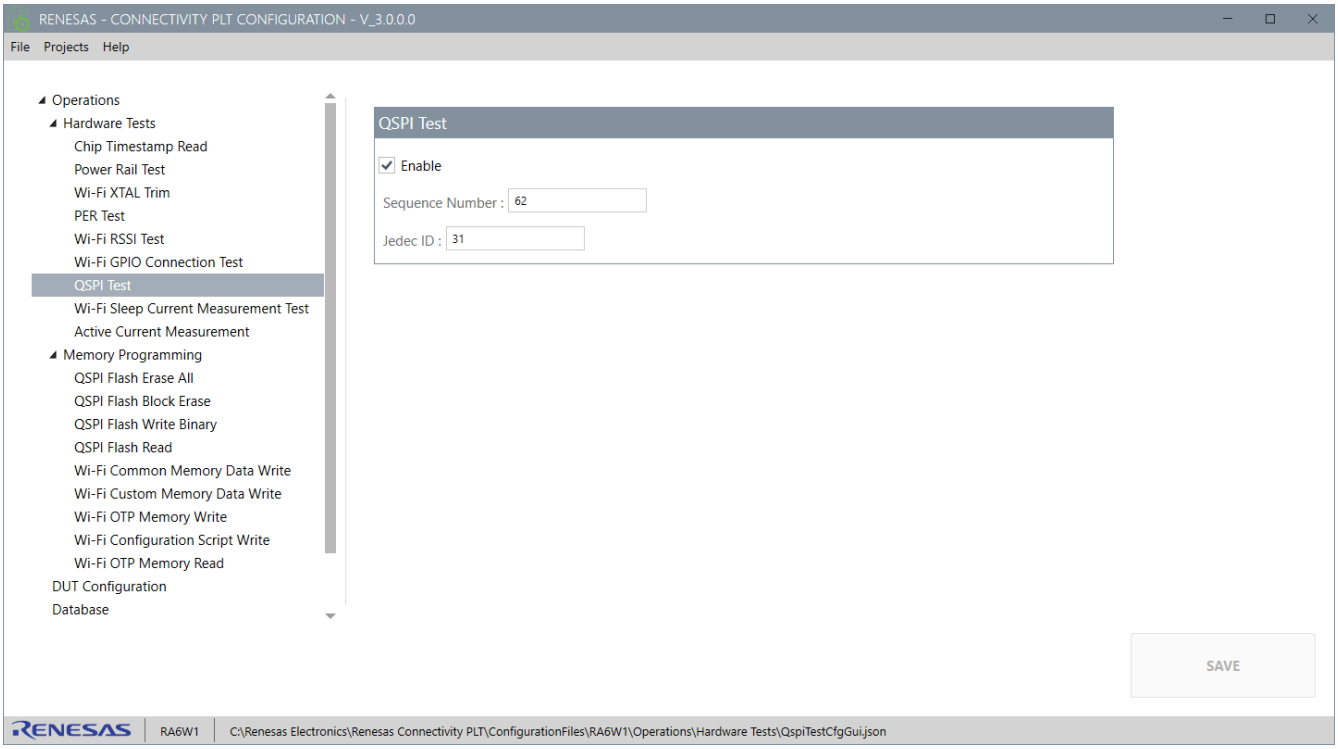



**Figure 100. QSPI Test configuration**

**Table 56. QSPI Test parameters**

| Parameter | Description |
|---|---|
| Enable | Enables QSPI Test operation |
| Sequence Number | Order in which the test is run in execution |
| JEDEC ID | JEDEC ID |

## 8.4.20 Wi-Fi Configuration Script Write

Wi-Fi Configuration Script Write is available in RA6W1, RA6W2, RRQ6100x, and RRQ6105x.

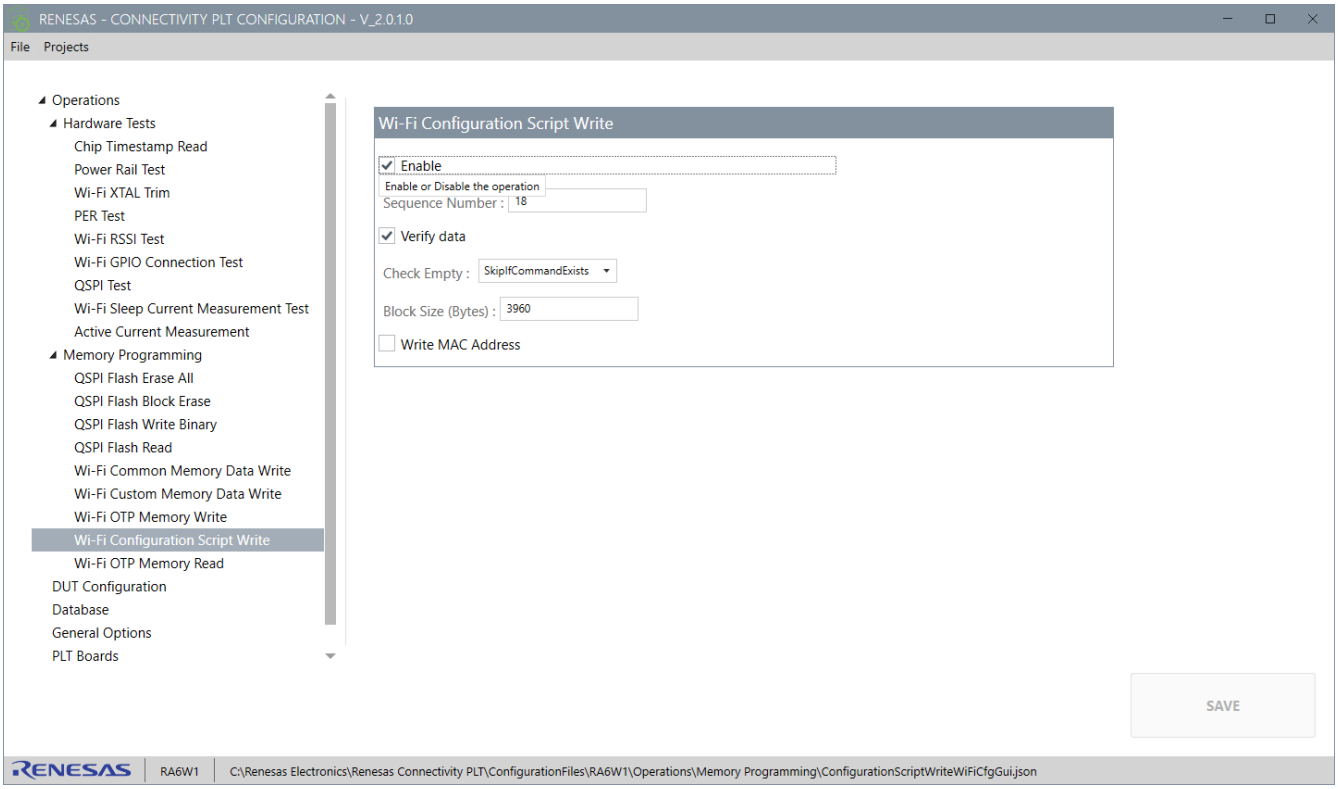



**Figure 101. Wi-Fi Configuration Script**

**Table 57. Wi-Fi Configuration Script Write parameters**

| Parameter | Description |
|---|---|
| Enable | Enables write operation |
| Verify Data | Verify Written data |
| Check Empty | Available options:<br>▪ NoCheck: Writes the data without reading first.<br>▪ ErrorIfNotEmpty: Fails the operation if the OTP area to be written is not empty.<br>▪ PassifNotEmpty: Skips the writing<br>PassifMatch: Fails the operation if the contents of the OTP differ from the binary file. |
| Block Size | Block size in bytes |
| Write MAC Address | Enables MAC address writing in Configuration Script area |

### 8.4.21 QSPI Flash Initialize

QSPI Flash Initialize is available in DA14495 and DA14AVDECT.

This operation retrieves flash characteristics, such as Manufacturer ID, Device ID, flash total size, and sector size and configures the flash for future write operations.

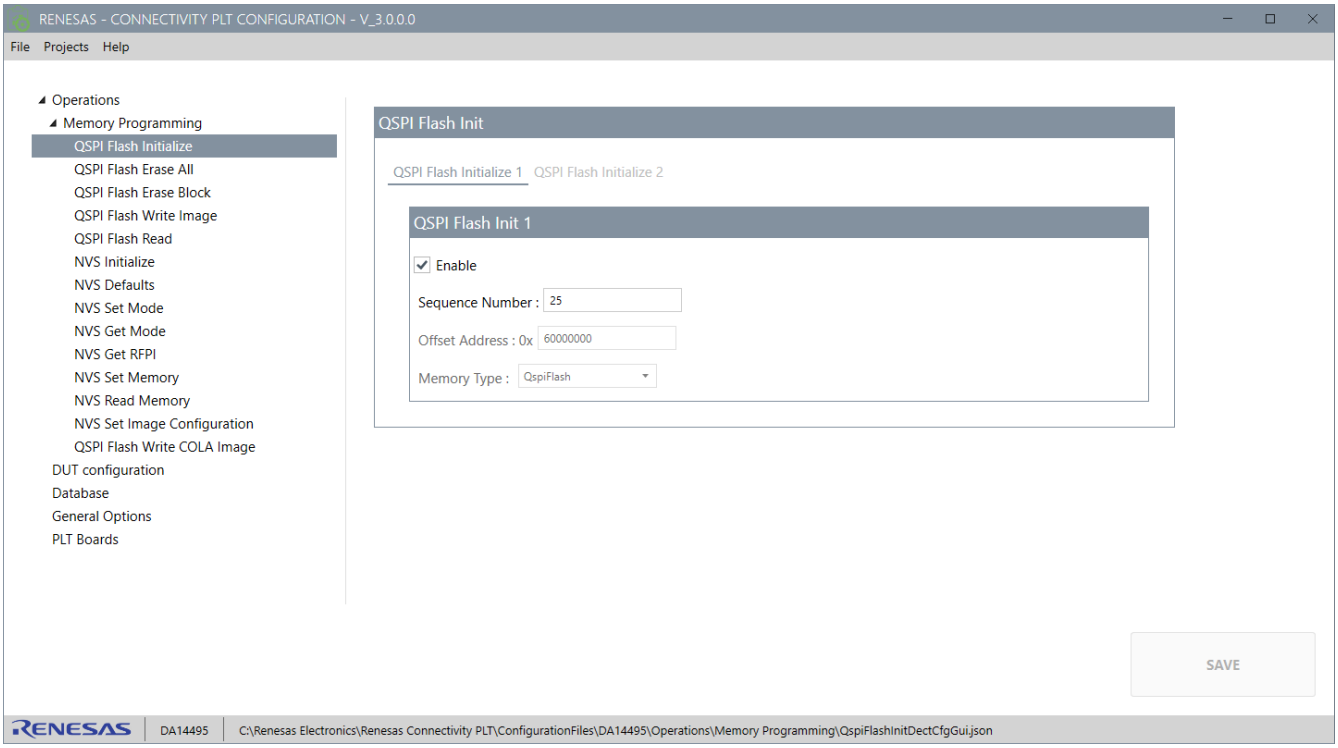



**Figure 102. QSPI Flash Initialize configuration**

**Table 58. SPI Flash Initialize parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the QSPI Flash Initialize operation |
| Sequence Number | Order in which the test is run in execution |
| Offset Address | Offset address for memory write operations (non-editable field) |
| Memory Type | The type of Flash memory. QSPI Flash is the default type (non-editable field). |

### 8.4.22 QSPI Flash Write Image

QSPI Flash Write Image is available for DA14495 and DA14AVDDECT.

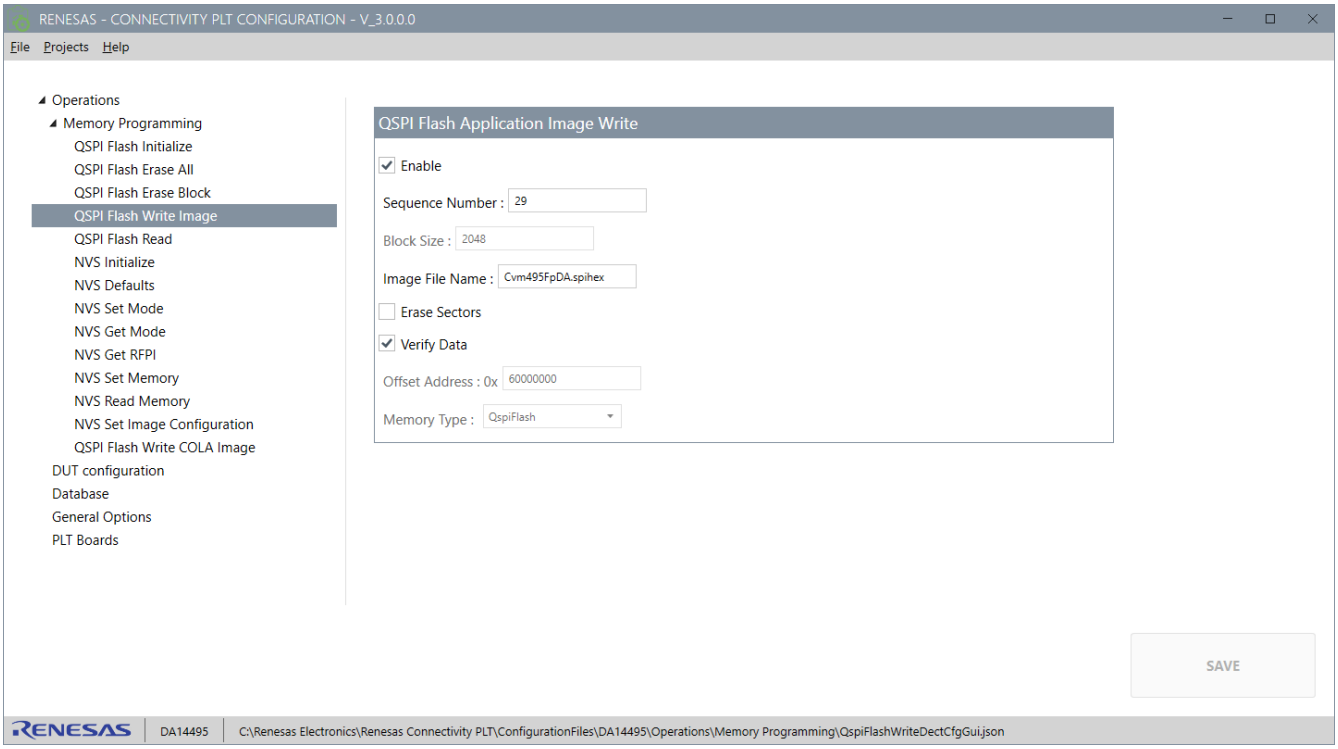



**Figure 103. QSPI Flash Write Image configuration**

**Table 59. QSPI Flash Write Image parameters**

| Parameter | Description |
|---|---|
| Enable | Enables QSPI Flash Write Image operation |
| Sequence Number | Order in which the test is run in execution |
| Block Size | Block size in bytes (fixed value, non-editable field) |
| Image File Name | Application image filename |
| Erase Sectors | Erases the relevant flash sectors before programming |
| Verify Data | Verify or not written data |
| Offset Address | Offset address for memory write operations (non-editable field) |
| Memory Type | The type of Flash memory. QSPI Flash is the default type (non-editable field). |

### 8.4.23 QSPI Flash Write COLA Image

QSPI Flash Write Co-Located application (COLA) Image is available for DA14495 and DA14AVDDECT.

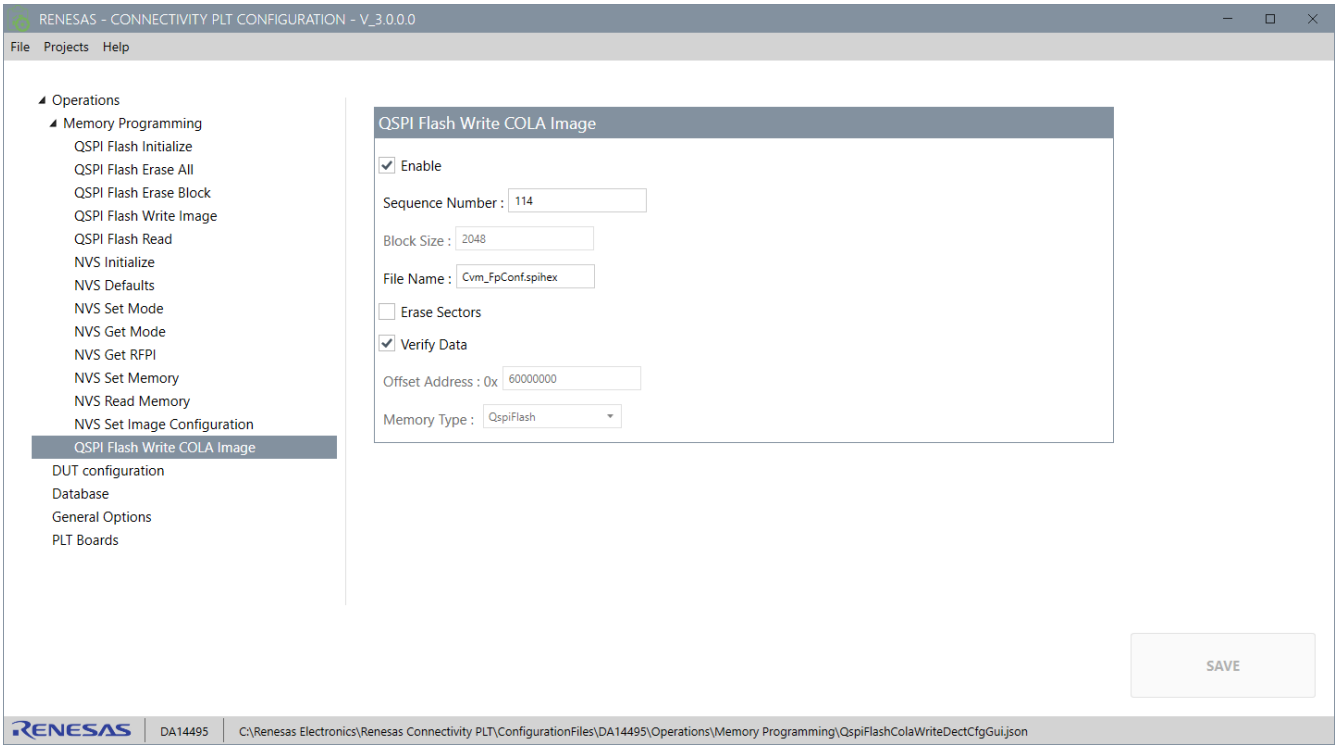



**Figure 104. QSPI Flash Write COLA Image configuration**

**Table 60. QSPI Flash Write COLA Image parameters**

| Parameter | Description |
|---|---|
| Enable | Enables QSPI Flash Write Image operation |
| Sequence Number | Order in which the test is run in execution |
| Block Size | Block size in bytes (fixed value, non-editable field) |
| File Name | COLA image filename |
| Erase Sectors | Erases the relevant flash sectors before programming |
| Verify Data | Verify or not written data |
| Offset Address | Offset address for memory write operations (non-editable field) |
| Memory Type | The type of Flash memory. QSPI Flash is the default type (non-editable field) |

### 8.4.24 OQSPI Flash Erase All

OQSPI Flash Erase All is available in DA14850 and DA14870.

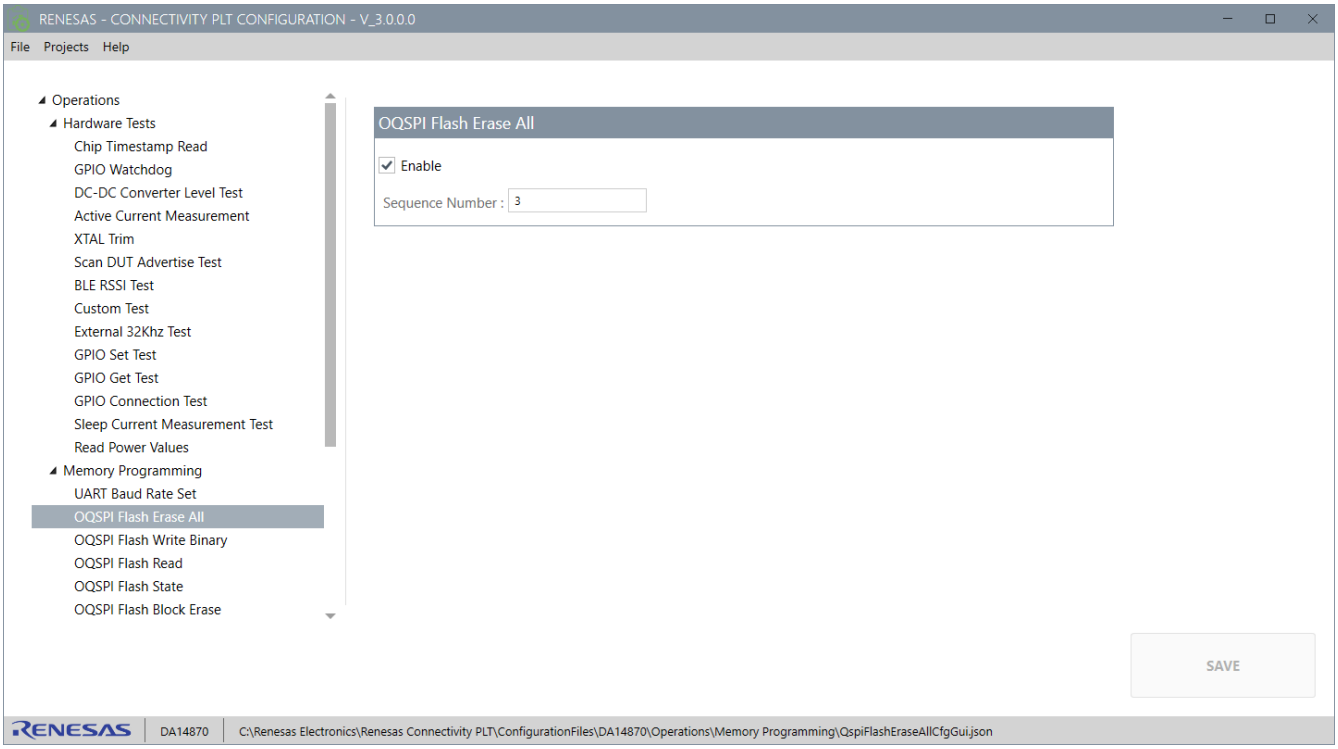



**Figure 105. OQSPI Flash Erase All configuration**

**Table 61. OQSPI Flash Erase All parameters**

| Parameter | Description |
|---|---|
| Enable | Enables OQSPI Flash Erase All operation |
| Sequence Number | Order in which the test is run in execution |

### 8.4.25 OQSPI Flash Block Erase

OQSPI Flash Block Erase is available for DA14850 and DA14870.

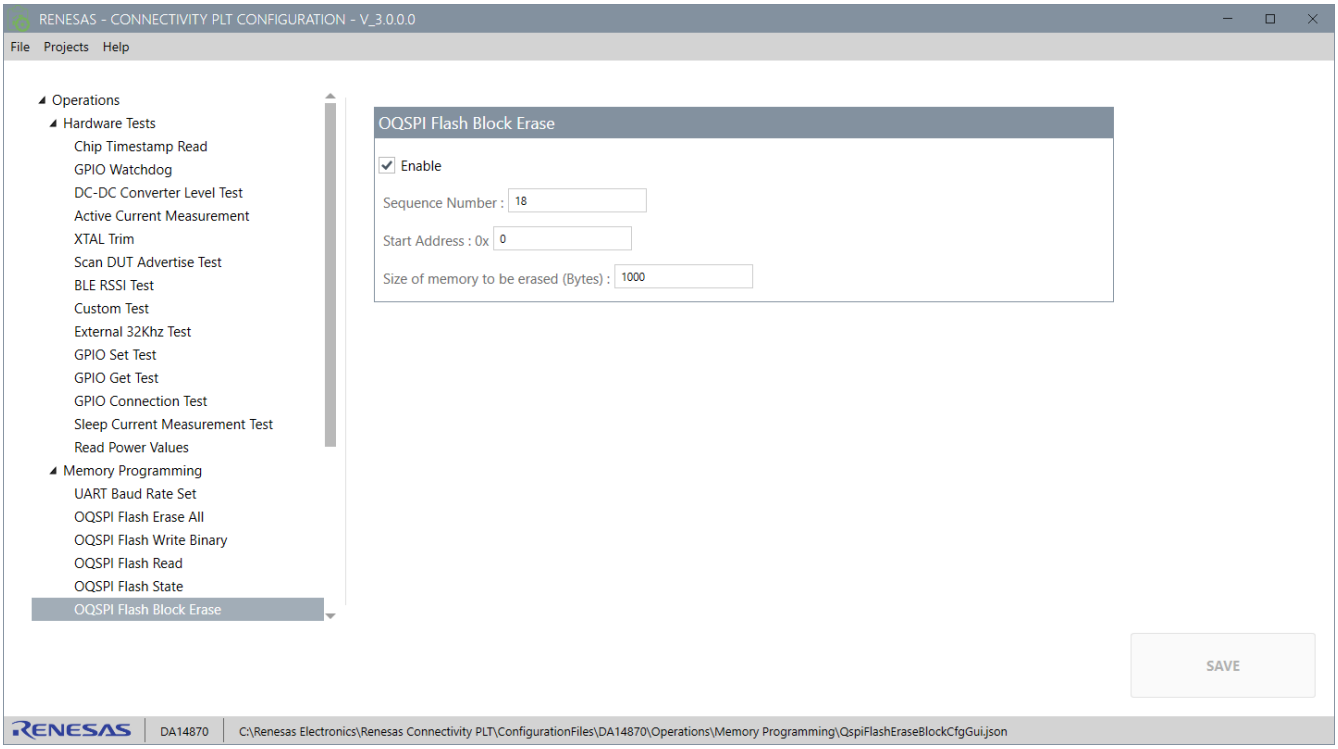



**Figure 106. OQSPI Flash Block Erase configuration**

**Table 62. OQSPI Flash Block Erase parameters**

| Parameter | Description |
|---|---|
| Enable | Enables OQSPI Flash Block Erase operation |
| Sequence Number | Order in which the test is run in execution |
| Start Address | Start address in hexadecimal format |
| Size of memory to be erased (Bytes) | Number of bytes to erase |

### 8.4.26 OQSPI Flash Write Binary

OQSPI Flash Write Binary is available in DA14850 and DA14870.

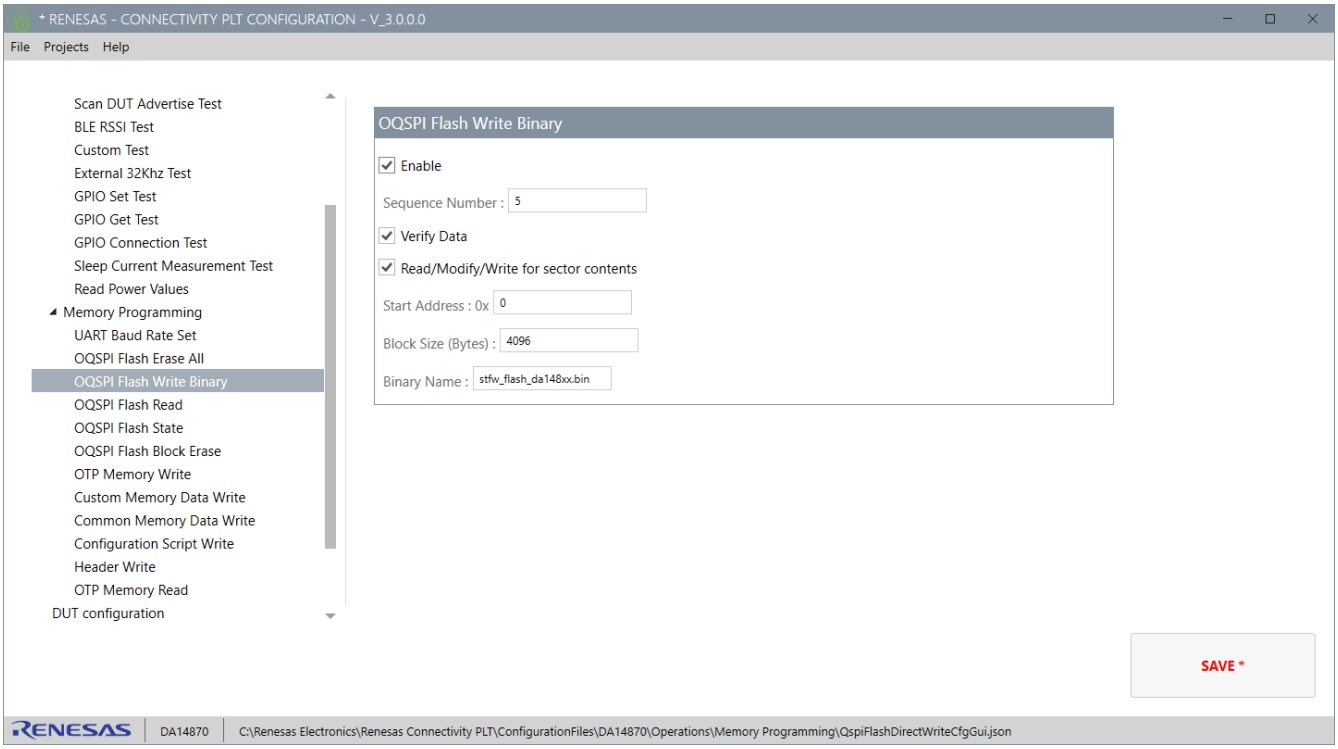



**Figure 107. OQSPI Flash Write Binary configuration**

**Table 63. OQSPI Flash Write Binary parameters**

| Parameter | Description |
|---|---|
| Enable | Enables OQSPI flash write operation |
| Sequence Number | Order in which the test is run in execution |
| Verify Data | Verify or not written data |
| Read/Modify/Write for Sector Contents | Add Read/Write/Modify to sector contents |
| Start Address | Start address in hexadecimal format |
| Block Size | Block size in bytes |
| Binary Name | Binary name |

### 8.4.27 OQSPI Flash Read

OQSPI Flash Read is available in RA6W1, RA6W2, RRQ6100x, and RRQ6105x.

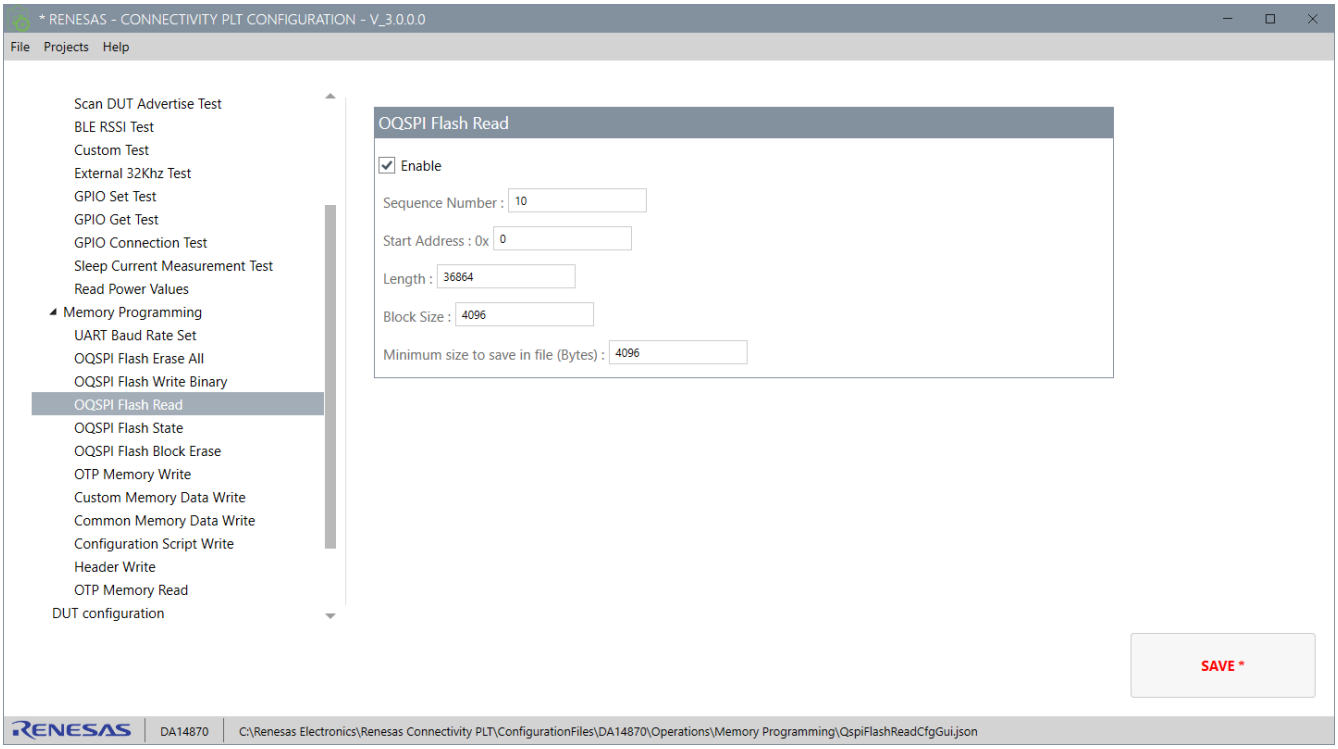



**Figure 108. QSPI Flash Read configuration**

**Table 64. QSPI Flash Read parameters**

| Parameter | Description |
|---|---|
| Enable | Enables OQSPI flash write operation |
| Sequence Number | Order in which the test is run in execution |
| Start Address | Start address in hexadecimal format |
| Length | Length of read |
| Block Size | Block size in bytes |
| Minimum Size to Save in File (Bytes) | Minimum size in bytes |

### 8.4.28 OQSPI Flash State

QSPI Flash State is available in DA14850 and DA14870.

This test can be used to retrieve flash state information, such as manufacturer ID, device type, density, and whether driver is configured or not.

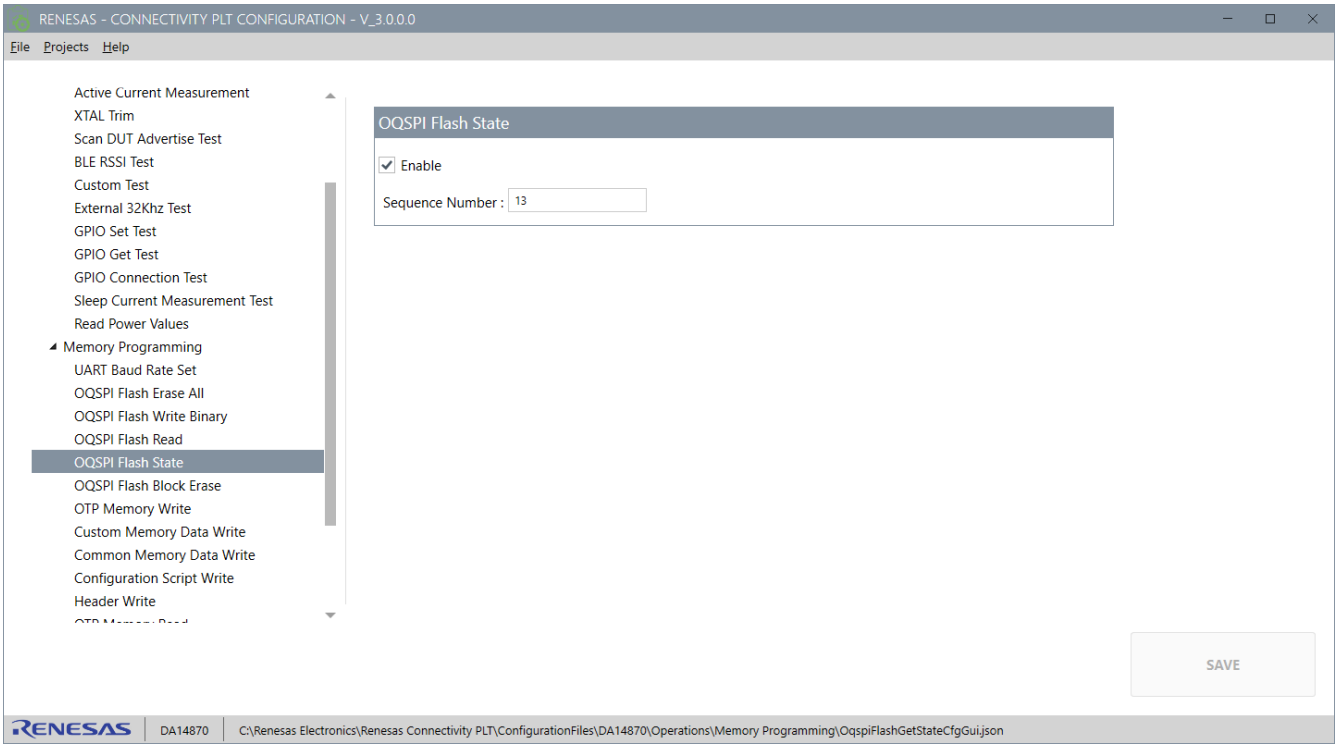



**Figure 109. OQSPI Flash State configuration**

**Table 65. OQSPI Flash State parameters**

| Parameter | Description |
|---|---|
| Enable | Enables common memory data write operation |
| Sequence Number | Order in which the test is run in execution |

### 8.4.29 NVS Initialize

NVS Initialize is available in DA14495 and DA14AVDDECT.

This operation is necessary to initialize Mail communication with the DUTs for NVS programming operations.

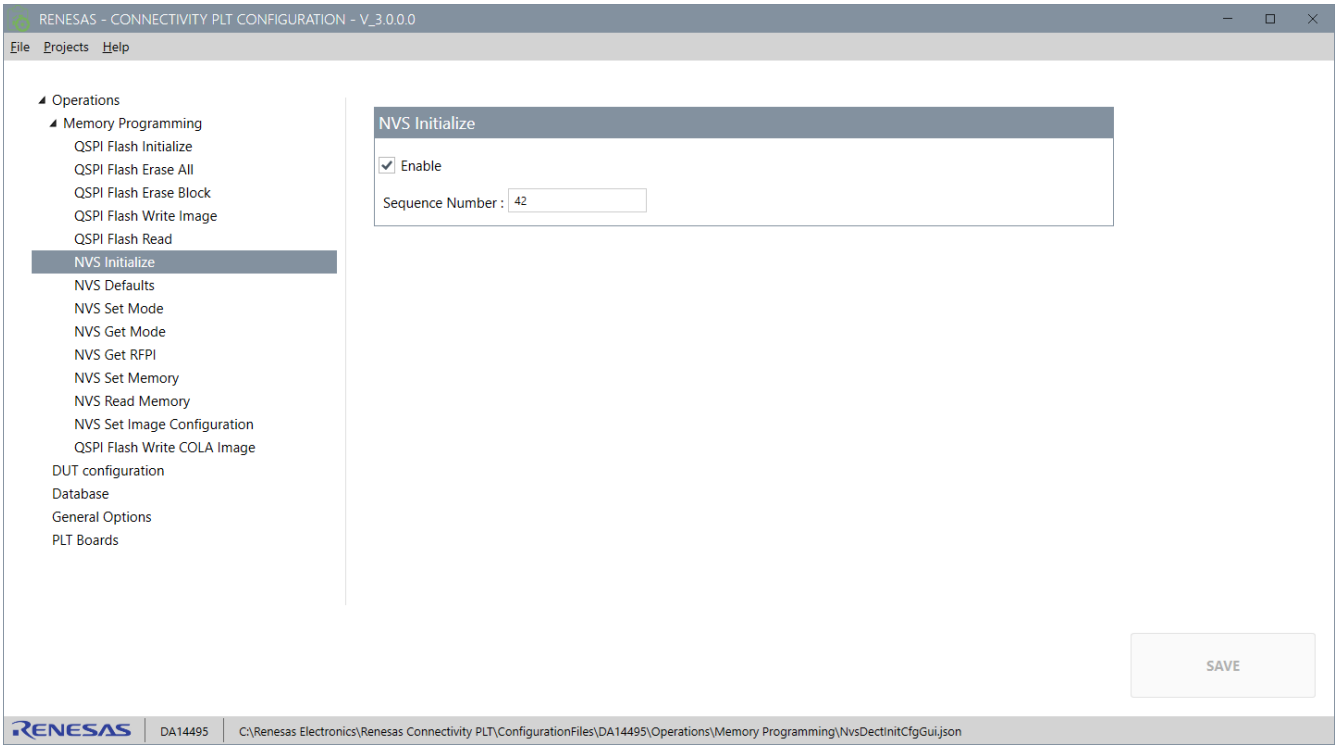



**Figure 110. NVS Initialize configuration**

**Table 66. NVS Initialize parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the NVS Initialize operation |
| Sequence Number | Order in which the test is run in execution |

### 8.4.30 NVS Defaults

NVS Defaults is available in DA14495, DA14AVDDECT.

Using this command, the NVS can be set to the factory settings. Every NVS parameter has a specific type (factory default/standard default), which defines how that affects NVS Defaults command.

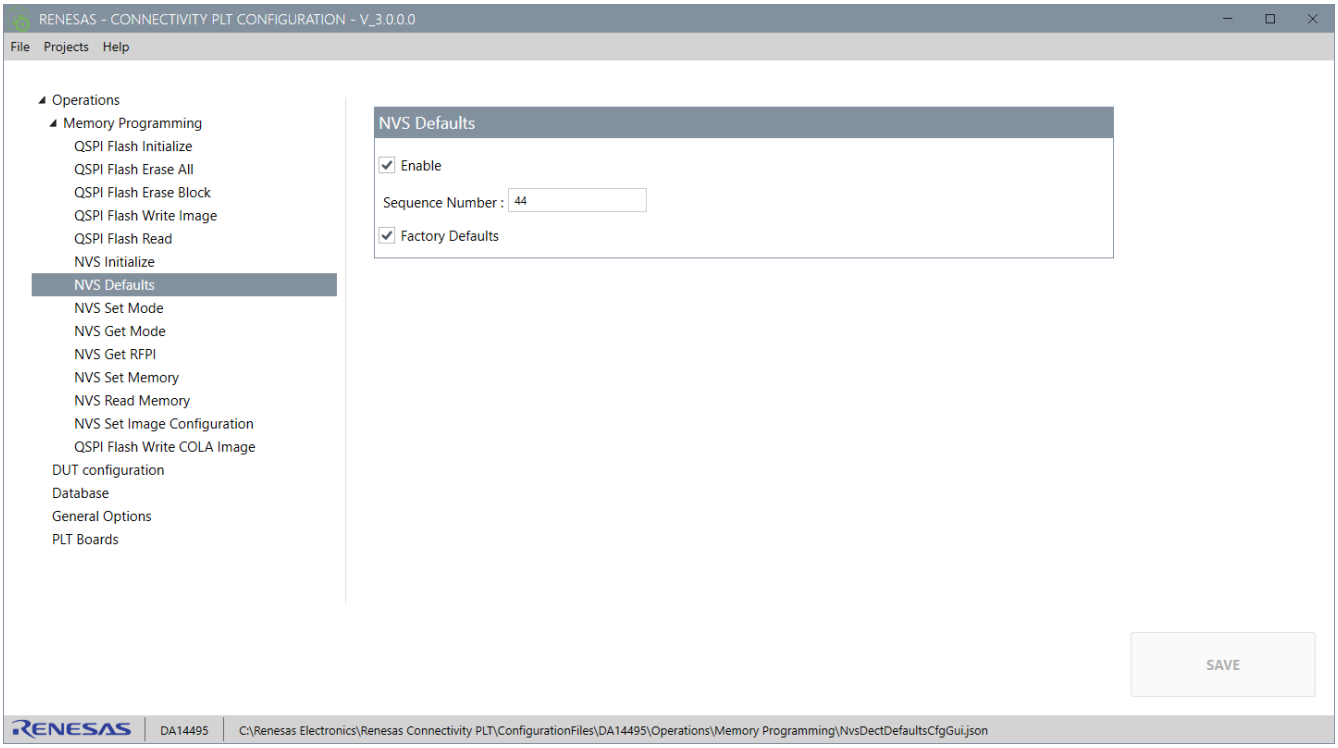



**Figure 111. NVS Defaults configuration**

**Table 67. NVS Defaults parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the NVS Defaults operation |
| Sequence Number | Order in which the test is run in execution |
| Factory Defaults | When selected, both Factory and Standard default type parameters are set to default value.<br>If not selected, Standard default type parameters are set to default, while Factory default type parameters are not affected. |

### 8.4.31 NVS Set Mode

NVS Set Mode is available in DA14495 and DA14AVDDECT.

This operation sets the corresponding value for the region chosen.

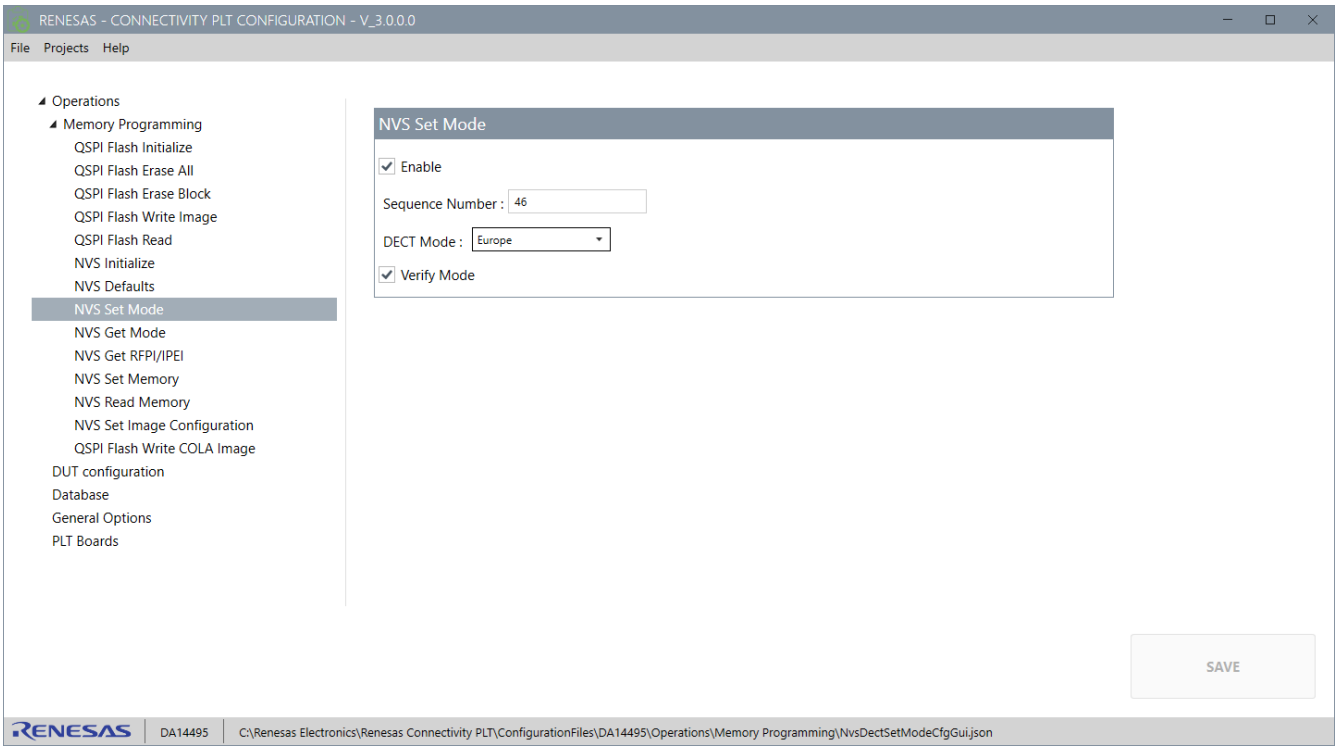



**Figure 112. NVS Set Mode configuration**

**Table 68. NVS Set Mode parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the NVS Set Mode operation |
| Sequence Number | Order in which the test is run in execution |
| DECT Mode | Region used for DECT operation |
| Verify Mode | Verifies that the DECT Mode has been set correctly |

### 8.4.32 NVS Get Mode

NVS Get Mode is available in DA14495 and DA14AVDDECT.

This operation returns the region selected.

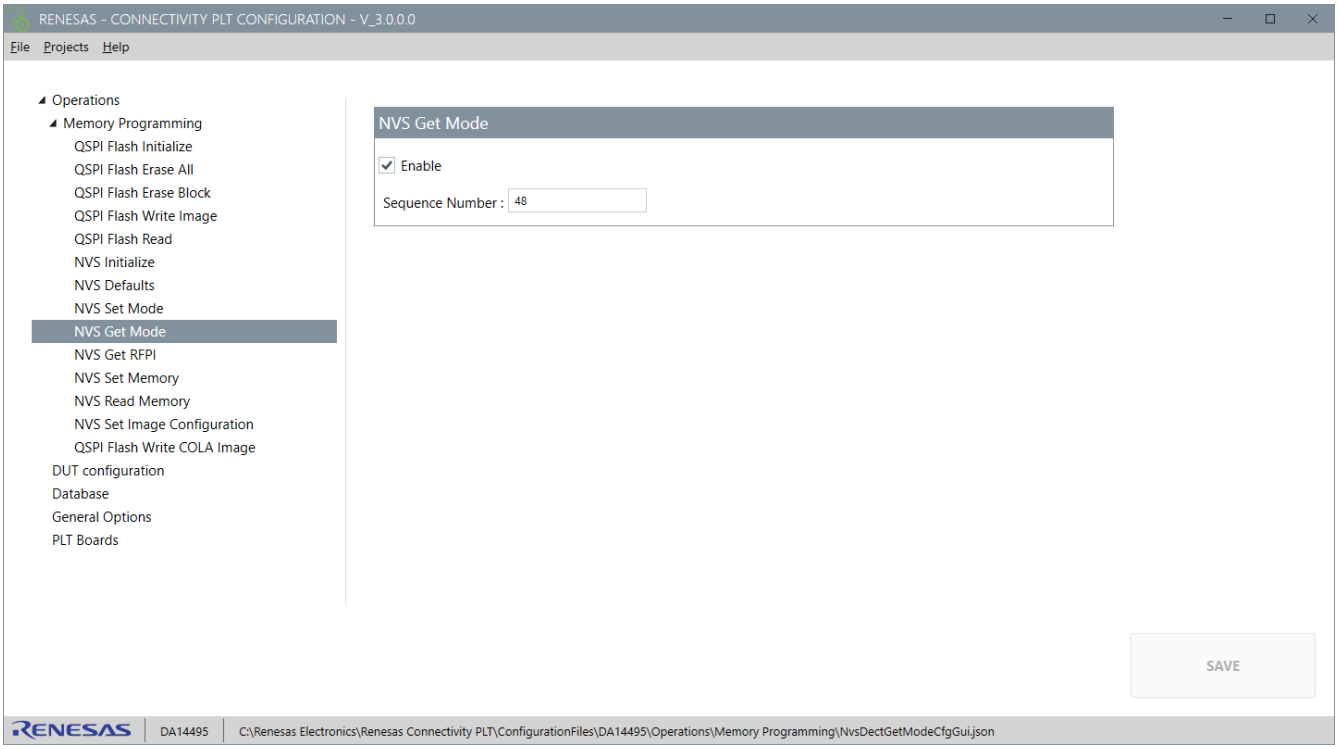



**Figure 113. NVS Get Mode configuration**

**Table 69. NVS Get Mode parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the NVS Get Mode operation |
| Sequence Number | Order in which the test is run in execution |

## 8.4.33 NVS Get RFPI/IPEI

NVS Get RFPI is available in DA14495, DA14AVDDECT.

This operation returns the Radio Fixed Part Identity (RFPI) of the FP device or the International Portable Part Equipment Identity (IPEI) of the PP device.

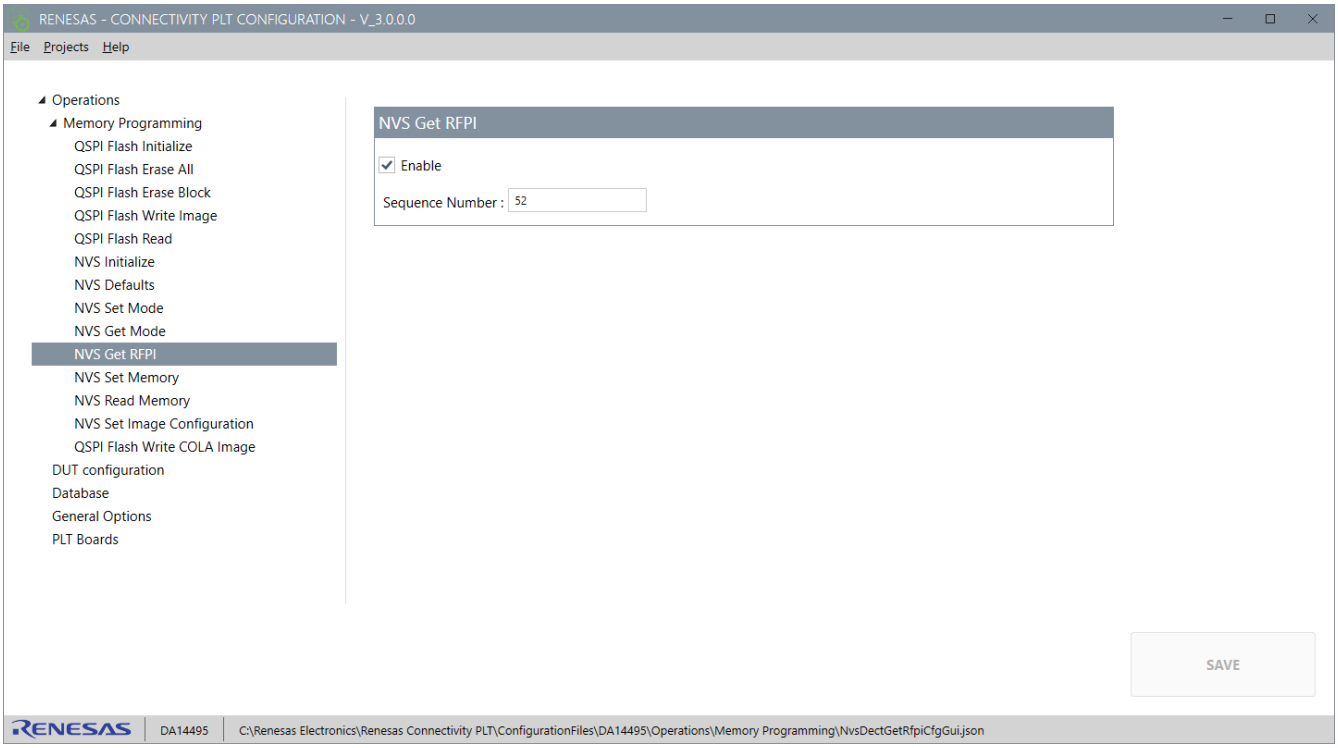



**Figure 114. NVS Get RFPI configuration**

**Table 70. NVS Get RFPI parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the NVS Get RFPI operation |
| Sequence Number | Order in which the test is run in execution |

## 8.4.34 NVS Set Memory

NVS Set Memory is available in DA14495 and DA14AVDDECT.

The aim of this test is to set the desired values to specific NVS positions to control certain parameters. Table 71 summarizes the available options.

**Table 71. NVS Set memory options**

| Name | Description |
|---|---|
| Diversity Mode | Sets antenna diversity mode |
| Power Low RSSI Limit | Sets RSSI Low Limit |
| Power High RSSI Limit | Sets RSSI High Limit |
| Extended FP Cap | Configures extended FP capability |
| Used RF Carriers 5G | Sets the RF Carrier settings |
| Delete IPUI | Deletes International Portable User Identity |
| Early Encryption Keys | Sets encryption keys array |
| RF Power Level | Sets the RF power level from level 0 to 5 |
| Control GFSK Modulation | Sets GFSK/DBPSK modulation |

| Name | Description |
|---|---|
| Control External Host App | Enables or disables the external host application |
| Control Battery Charging Algorithm | Enables or disables battery charging algorithm |
| Battery Charging Algorithm | Sets battery charging algorithm |

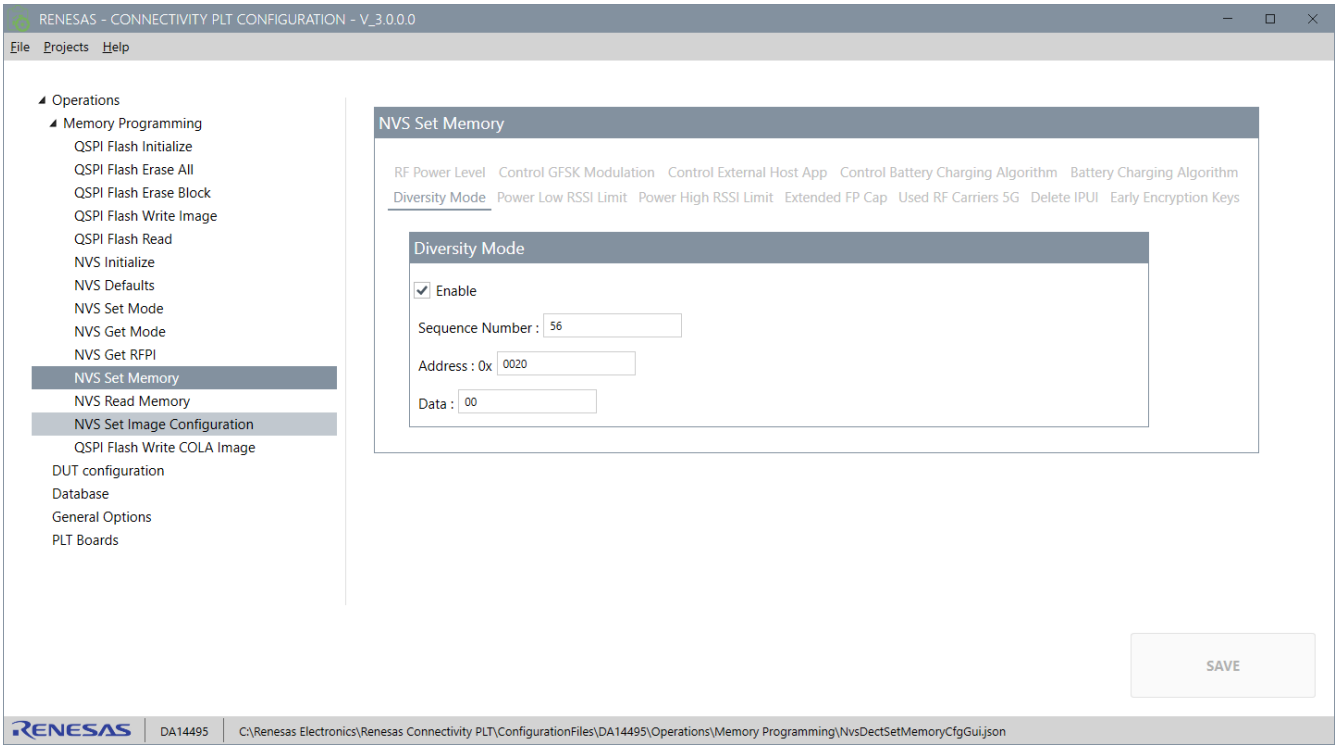



**Figure 115. NVS Set Memory configuration**

**Table 72. NVS Set Memory parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the NVS Set Mode operation |
| Sequence Number | Order in which the test is run in execution |
| Address | NVS memory address (in hexadecimal format) to write the data |
| Data | Data to be written in hexadecimal format |

## 8.4.35 NVS Read Memory

NVS Read Memory is available in DA14495 and DA14AVDDECT.

The aim of this test is to read contents from specific NVS positions. Table 73 summarizes the available options.

**Table 73. NVS Read Memory options**

| Name | Description |
|---|---|
| Diversity Mode | Reads antenna diversity mode |
| Power Low RSSI Limit | Reads RSSI Low Limit |
| Power High RSSI Limit | Reads RSSI High Limit |
| Extended FP Cap | Reads extended FP capability |
| Used RF Carriers 5G | Reads the RF Carrier settings |
| Delete IPUI | Reads International Portable User Identity |
| Early Encryption Keys | Reads encryption keys array |

| Name | Description |
|---|---|
| RF Power Level | Reads the RF power level from level 0 to 5 |
| Control GFSK Modulation | Reads GFSK/DBPSK modulation control value |
| Control External Host App | Reads external host application control value |
| Control Battery Charging Algorithm | Reads battery charging algorithm control value |
| Battery Charging Algorithm | Reads battery charging algorithm |

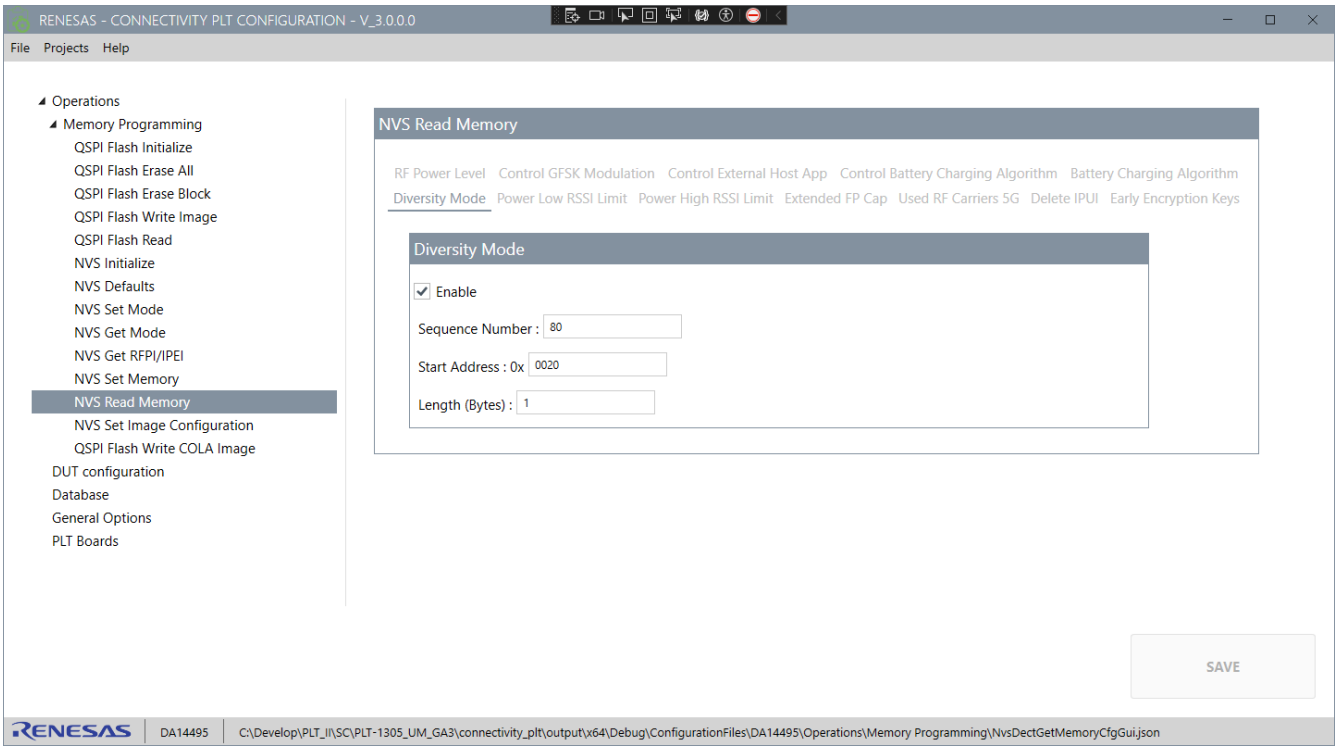



**Figure 116. NVS Read Memory configuration**

**Table 74. NVS Read Memory parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the NVS Set Mode operation |
| Sequence Number | Order in which the test is run in execution |
| Start Address | NVS memory address (in hexadecimal format) to read data from |
| Length | Number of bytes to read |

### 8.4.36 NVS Set Image Configuration

NVS Set Image Configuration is available in DA14495 and DA14AVDDECT.

This operation is used to select the image to run.

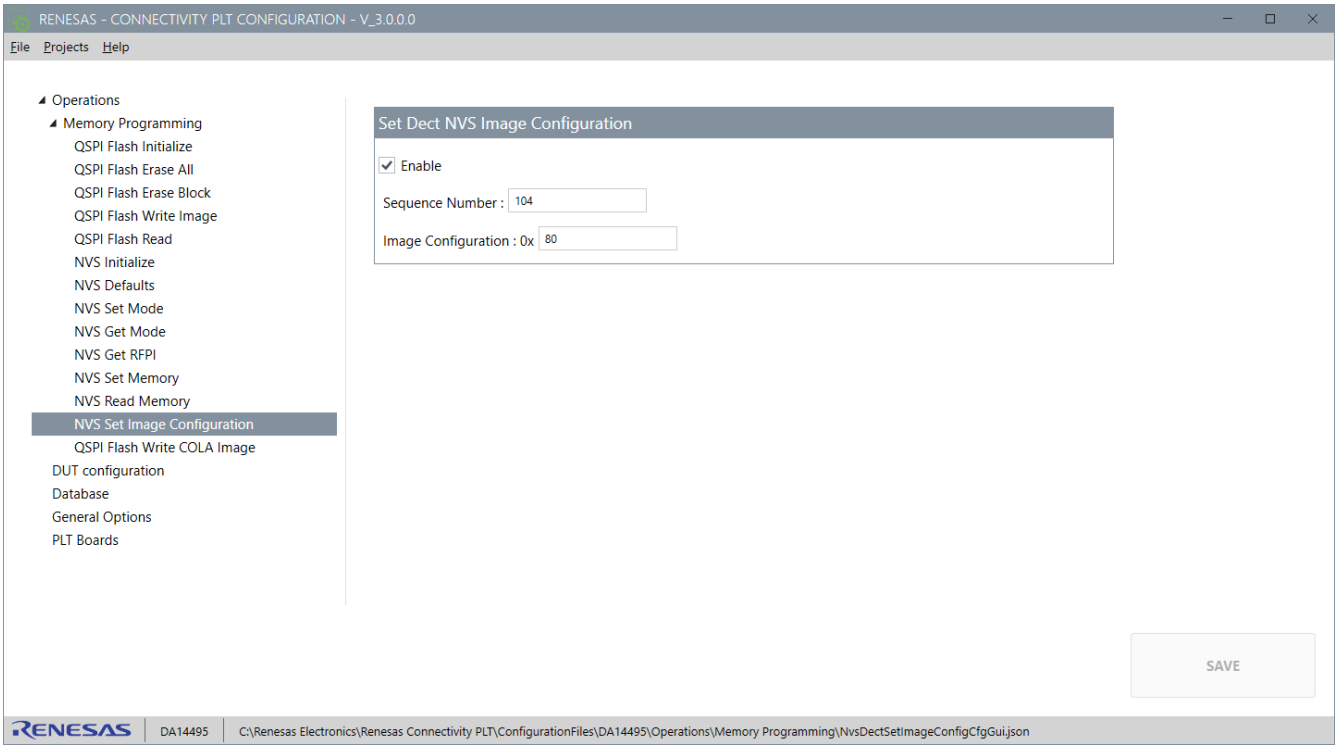



**Figure 117. NVS Set Image configuration**

**Table 75. NVS Set Image parameters**

| Parameter | Description |
|---|---|
| Enable | Enables the NVS Set Mode operation |
| Sequence Number | Order in which the test is run in execution |
| Image configuration | Value that defines the active image.<br>▪ Bit 7(0x80) Run Co-Located Application (COLA) image<br>▪ 0x00 or any other value: Run image 1. |

# 9. Connectivity PLT Application

The Renesas Connectivity PLT application serves as the primary graphical user interface for executing tests and programming tasks. It provides operators with a console to initiate and monitor the testing process. The application connects remotely to the Connectivity PLT Core service and functions as the front end of the PLT system. It can be launched either from the Windows Start menu or by directly running the executable file located in the installation directory. (`Renesas_Connectivity_PLT.exe`).

## 9.1 Connectivity PLT Application Main Execution Window

The main application window consists of tiles, each representing a DUT and its status, as well as a list of operations and their status, a start/finish button, and several other information fields. Figure 118 shows an example of a setup of a single Connectivity PLT board with 8 DUTs.

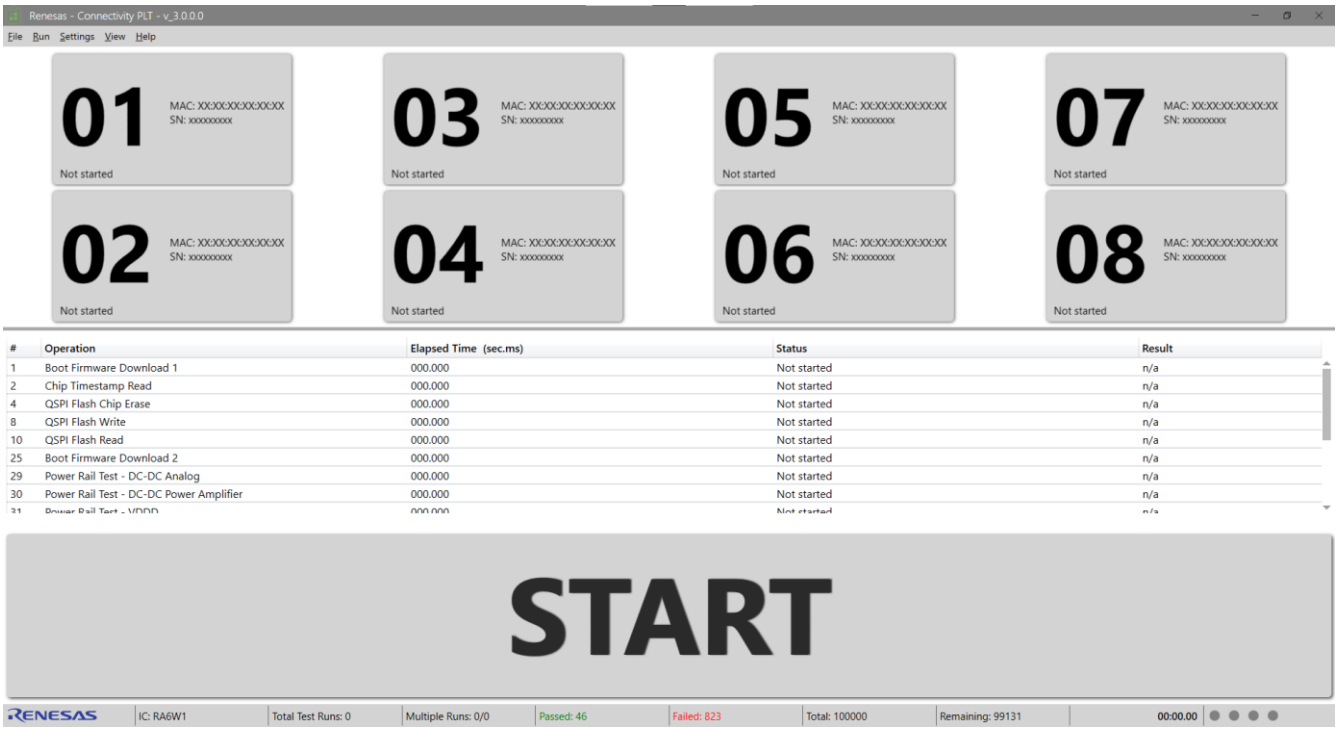



**Figure 118. Test execution start window**

You can configure the testing using the menu items at the top of window. Brief testing results are displayed at the bottom of the window, including the four LEDs for RFTU status (grey for not enabled, green for enabled and detected, and red for enabled but not detected).

Information tiles with numbers represent DUTs in your setup (Figure 119) and can accommodate from one to four Connectivity PLT boards consisting of one to thirty-two DUT tiles (one to eight DUTS per Connectivity PLT board). The layout is configured using X/Y axis in PLT boards Configuration, see Section 8.2.3.


**Figure 119. DUTs grid**

Each DUT tile contains the following:

- a – Display name (for example, 01) as configured in Connectivity PLT boards settings.
- b – Acquired Serial Number either automatically or from the initial dialog (only while tests are running).
- c – Acquired board address as fetched automatically form the devices database (either based on entered serial number or automatic base on next available) (only while tests are running).
- d – Current operation executing (only while tests are running), see Figure 120. If any of the DUTs fails, the current command field indicates the failed test command.

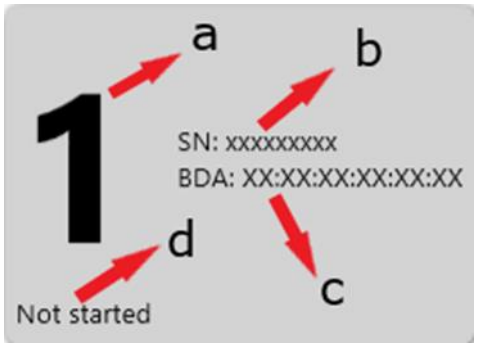



**Figure 120. Single DUT tile**

A splitter control allows the display area to be maximized for showing more of the operations table or more tile space depending on the user's needs.

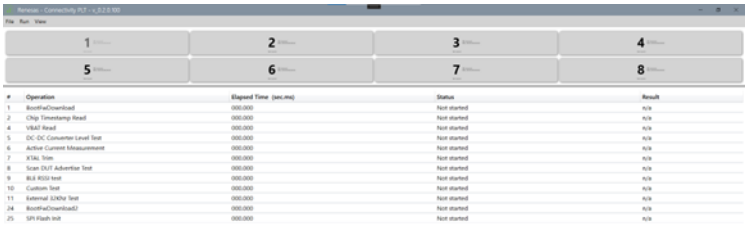



**Figure 121. Grid splitter top position**


**Figure 122. Grid splitter bottom position**

The operations grid is the table that shows the operations that are enabled to run. This table provides data about execution time, status, and results (Figure 123).

| # | Operation | Elapsed Time (sec.ms) | Status | Result |
|---|---|---|---|---|
| 1 | BootFwDownload | 000.000 | Not started | n/a |
| 2 | Chip Timestamp Read | 000.000 | Not started | n/a |
| 4 | VBAT Read | 000.000 | Not started | n/a |
| 5 | DC-DC Converter Level Test | 000.000 | Not started | n/a |
| 6 | Active Current Measurement | 000.000 | Not started | n/a |
| 7 | XTAL Trim | 000.000 | Not started | n/a |
| 9 | BLE RSSI test | 000.000 | Not started | n/a |
| 10 | Custom Test | 000.000 | Not started | n/a |
| 11 | External 32Khz Test | 000.000 | Not started | n/a |
| 24 | BootFwDownload2 | 000.000 | Not started | n/a |
| 25 | SPI Flash Init | 000.000 | Not started | n/a |

**Figure 123. Operations grid**

Start button has four states:

- **Idle** – Grey and enabled; right click the button or press space bar to start testing.
- **Enter Device Serial Number** – At this state, a dialog is displayed requesting to enter the serial number.
- **Running** – Blue and disabled; testing is in process.
- **Finish** – Green if testing is successful, or red if testing failed.

You can also use keyboard shortcuts: to start testing, select S or the Spacebar; and to finish testing, select F.

Finally, when selecting the Help menu, you can open the User Manual, using the system's default PDF viewer (Figure 124).

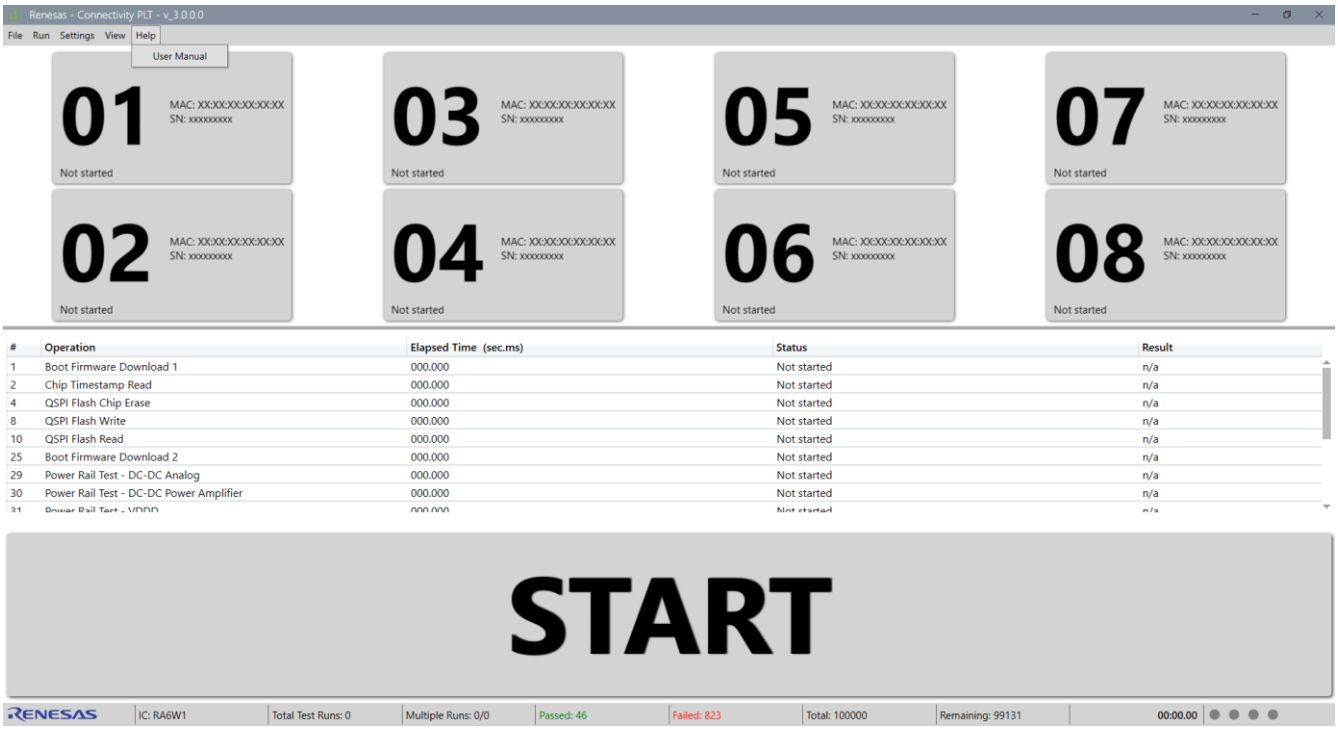



**Figure 124. User Manual under Help menu**

## 9.2 Test Execution

The test execution process has the following stages:

1. **Serial numbers assignment (optional)**

   If the **Device Specific Serial Number** option was enabled, a dialog to enter the first serial number is displayed when the test process is initiated, see Figure 125.

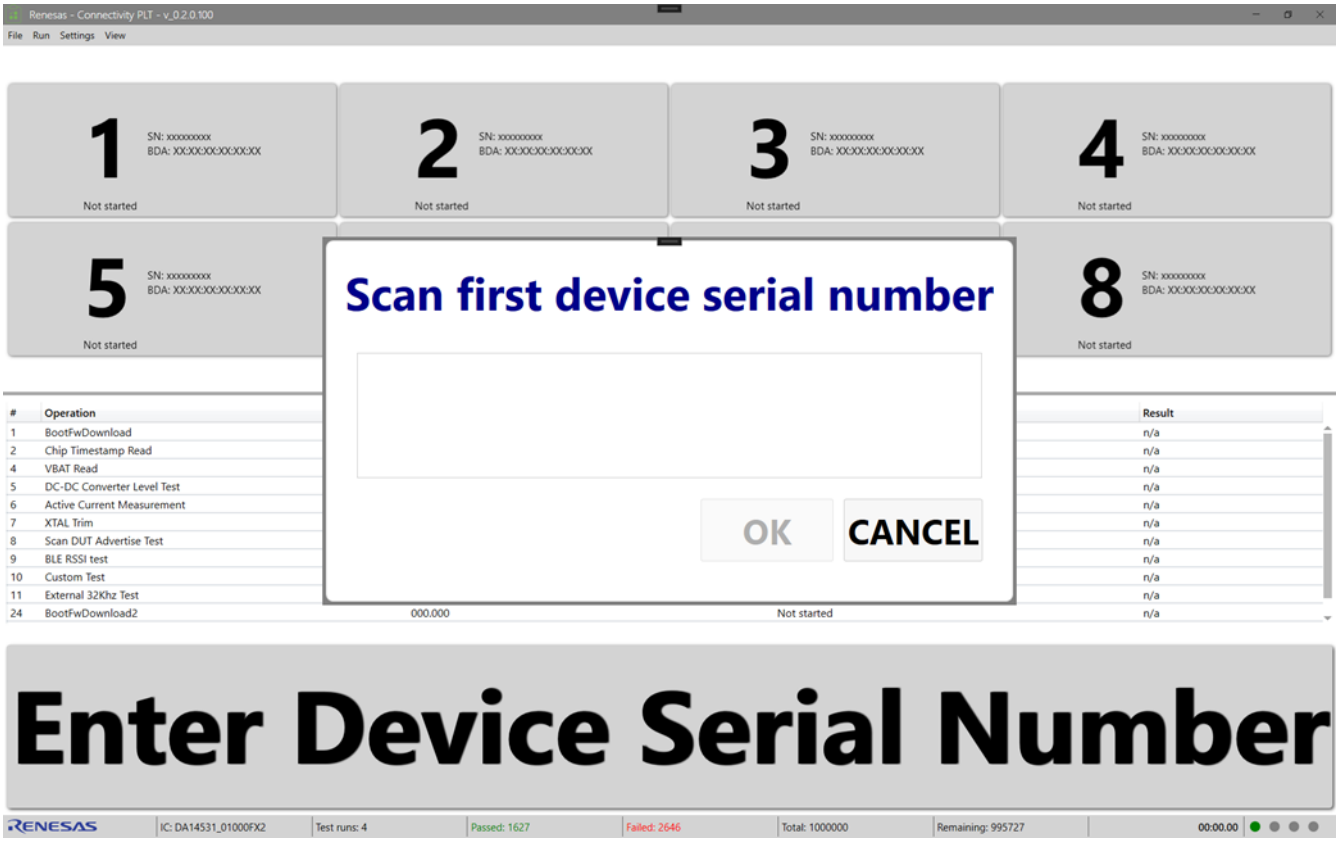



**Figure 125. Enter serial number screen**

   After the serial number is entered, all DUTs are assigned with a sequential serial number retrieved from the database along with the corresponding Bluetooth® address.

2. **Configuration**

   Connectivity PLT fetches the latest configuration available in the current project. If changes are made to the configuration, reload is required to apply these changes (go to **File** > **Reload Configuration** or press **F5**). Even if manual reload is not executed, the configuration is going to be reloaded at the beginning of the test process and a warning message appears, see Figure 126.

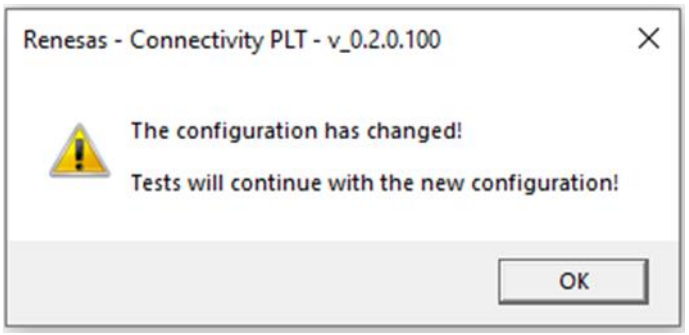



**Figure 126. Configuration reload warning**

3. **OTP Write Warning**

   If any operation includes writing to the OTP memory, a warning is displayed at the start of the test process to avoid accidental writes since the OTP cannot be re-written. The warning can be suppressed for future runs from the same dialog by selecting the **Don't ask again for the current session** checkbox.

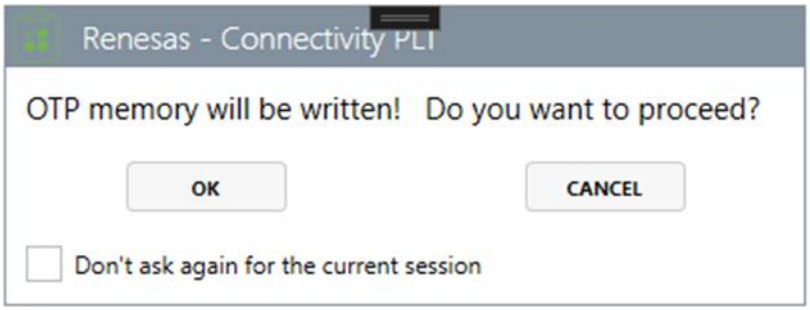



**Figure 127. OTP write warning message**

4. **Tests Execution**

   When the test process starts, the background color of DUT tiles changes to blue to indicate the "in progress" status. The active operation row text color changes to blue also to indicate "in progress" while also the dedicated test timer starts. The start button is disabled but serves as a progress bar colored blue while executing, and green or red depending on the result, see Figure 128.

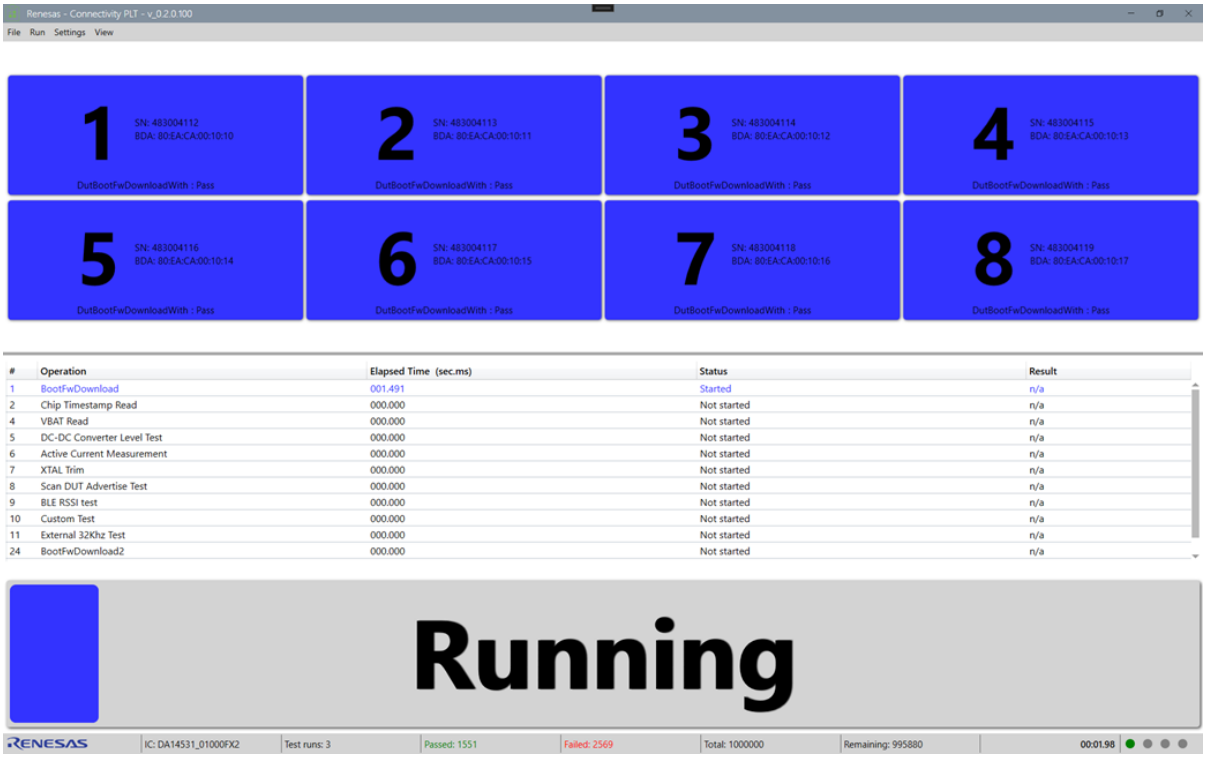



**Figure 128. Test execution**

   If any DUT fails on any operation, the tile background becomes red, the failed command is displayed at the bottom of the tile, and the operation row text color changes to red to indicate that at least one DUT failed this test.

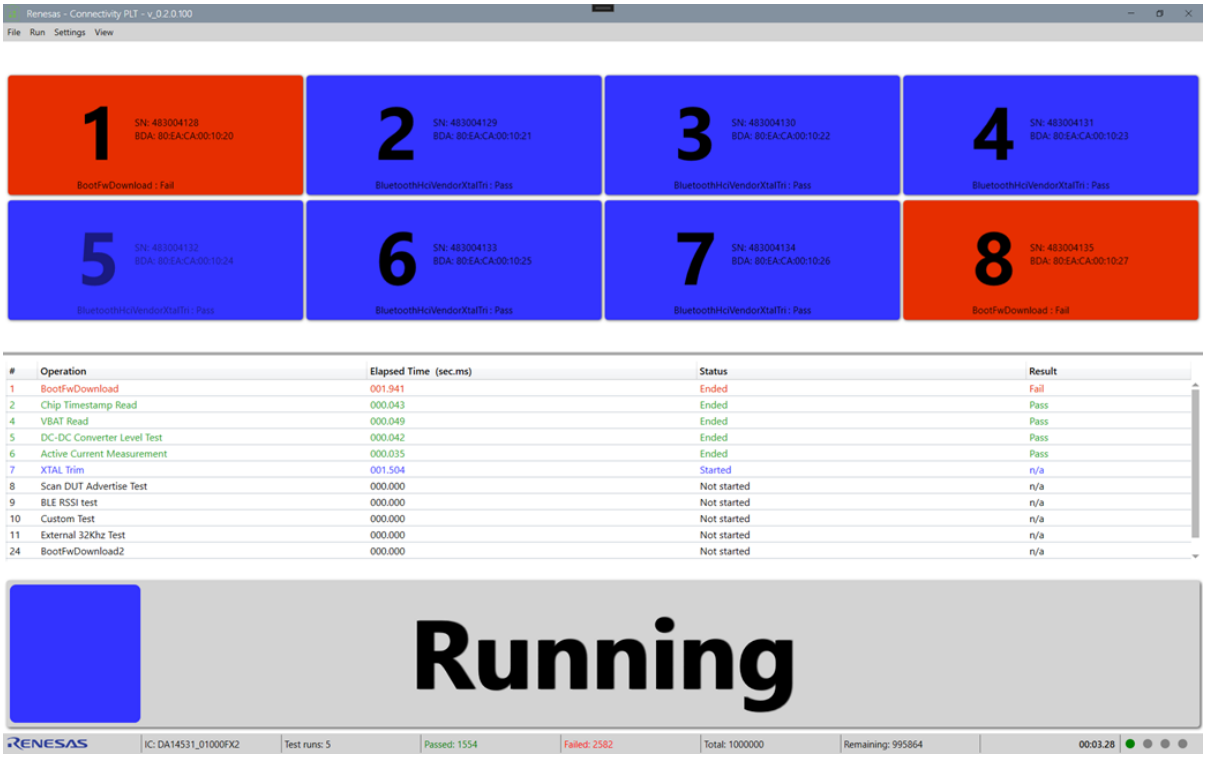



**Figure 129. DUTs fail during execution**

5. **Test process completion**

- Successful completion

  If all operations were successful (Pass) for all the DUTs tiles, lines in the operations table, and the start button are green.

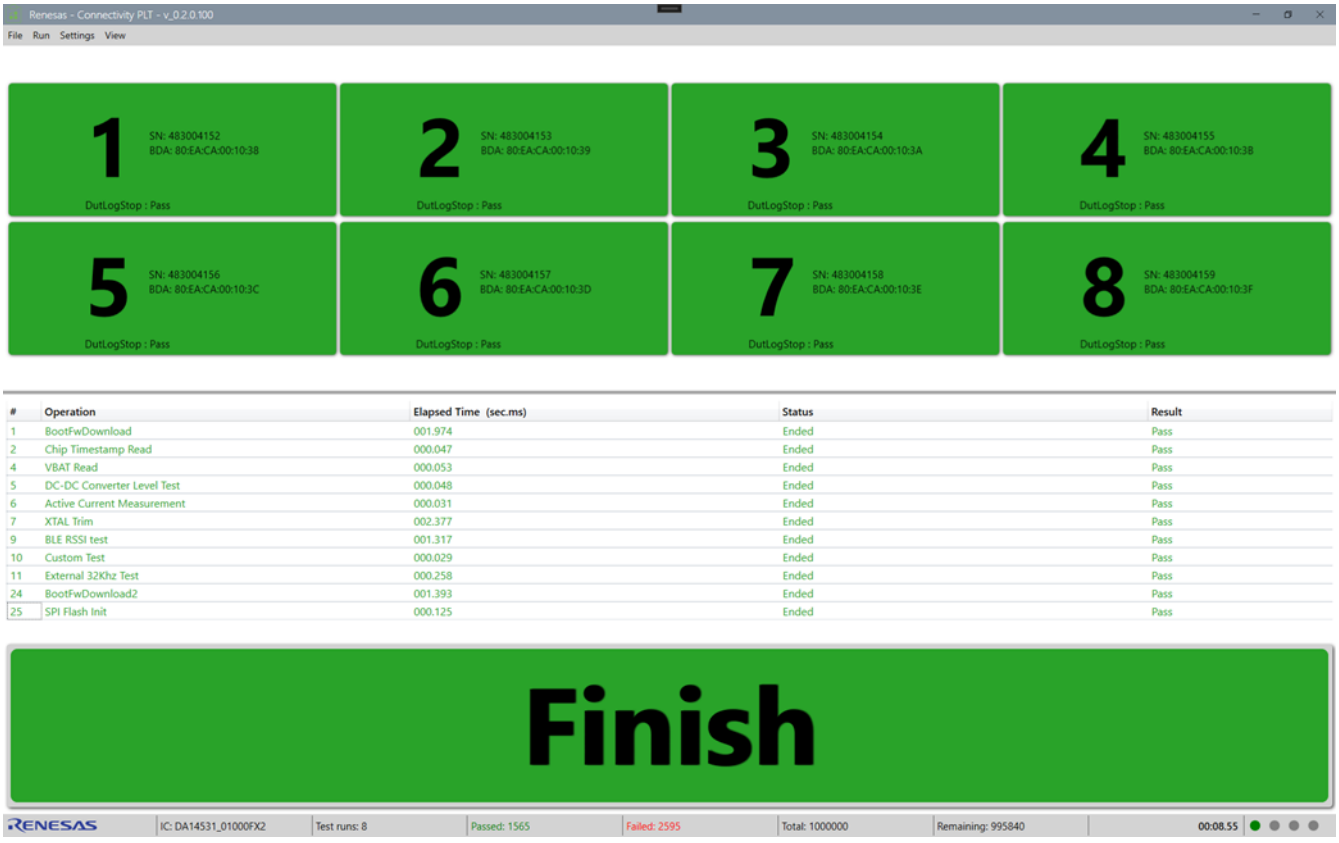



**Figure 130. Successful test sequence**

- Incomplete testing due to failures

  If during the test process all DUTs have at least one test that failed, then the test process stops as no more DUTs are available to continue. For this case, all DUT tiles are red whilst the operations table displays in red all the operations that failed for at least one DUT. Figure 131 shows that DUTs 1 and 6 failed on GPIO Connection Test and DUTs 2,3,4,5,7, and 8 failed in BootFwDownload. Failed command is indicated on each DUT tile.

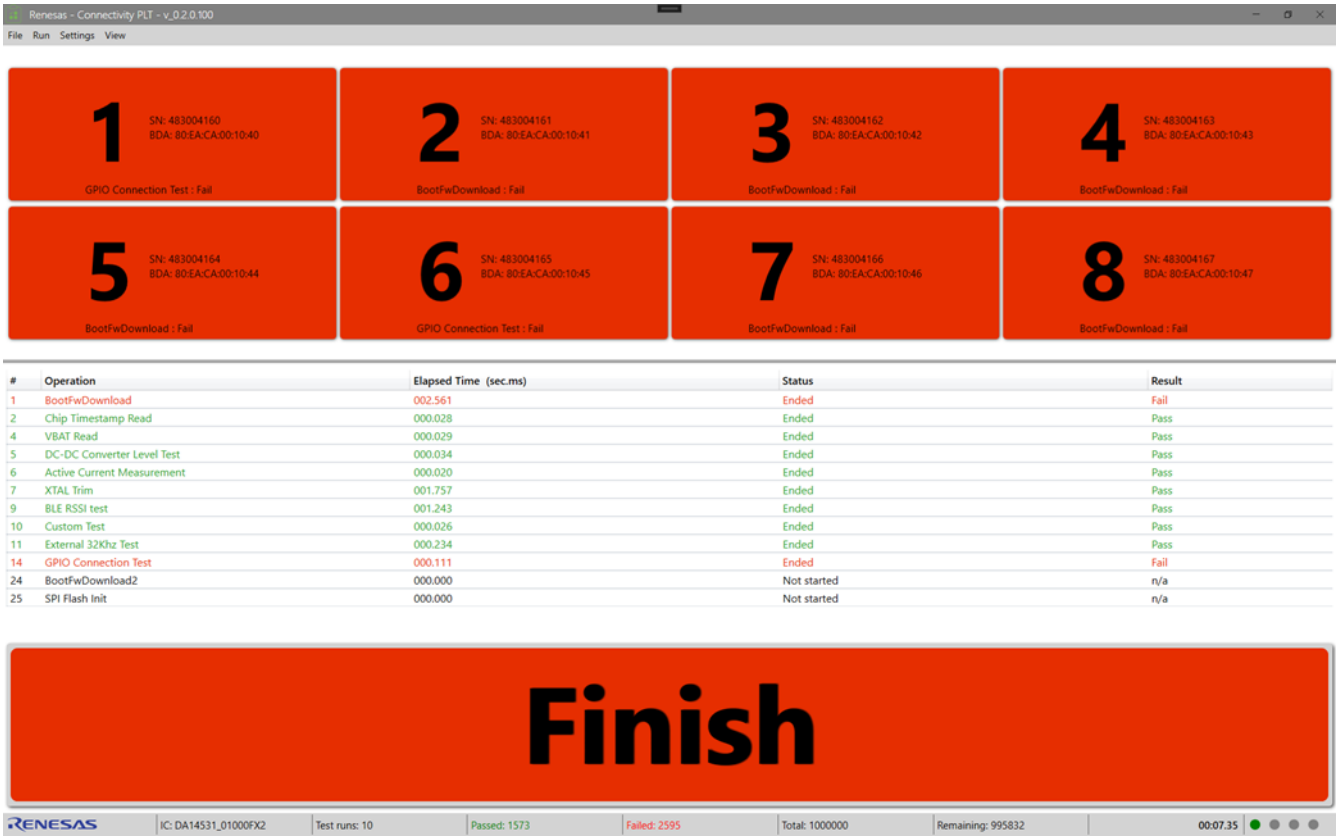



**Figure 131. Failed test sequence**

- Partially successful completion

  The testing is considered partially successful when there are both passed and failed DUTs. Figure 132 shows that DUT 4 failed on BLE RSSI test but 1,2,3,5,6,7,8 passed the test sequence. Successful DUTs are green, failed are red, failed operation and start button is red.

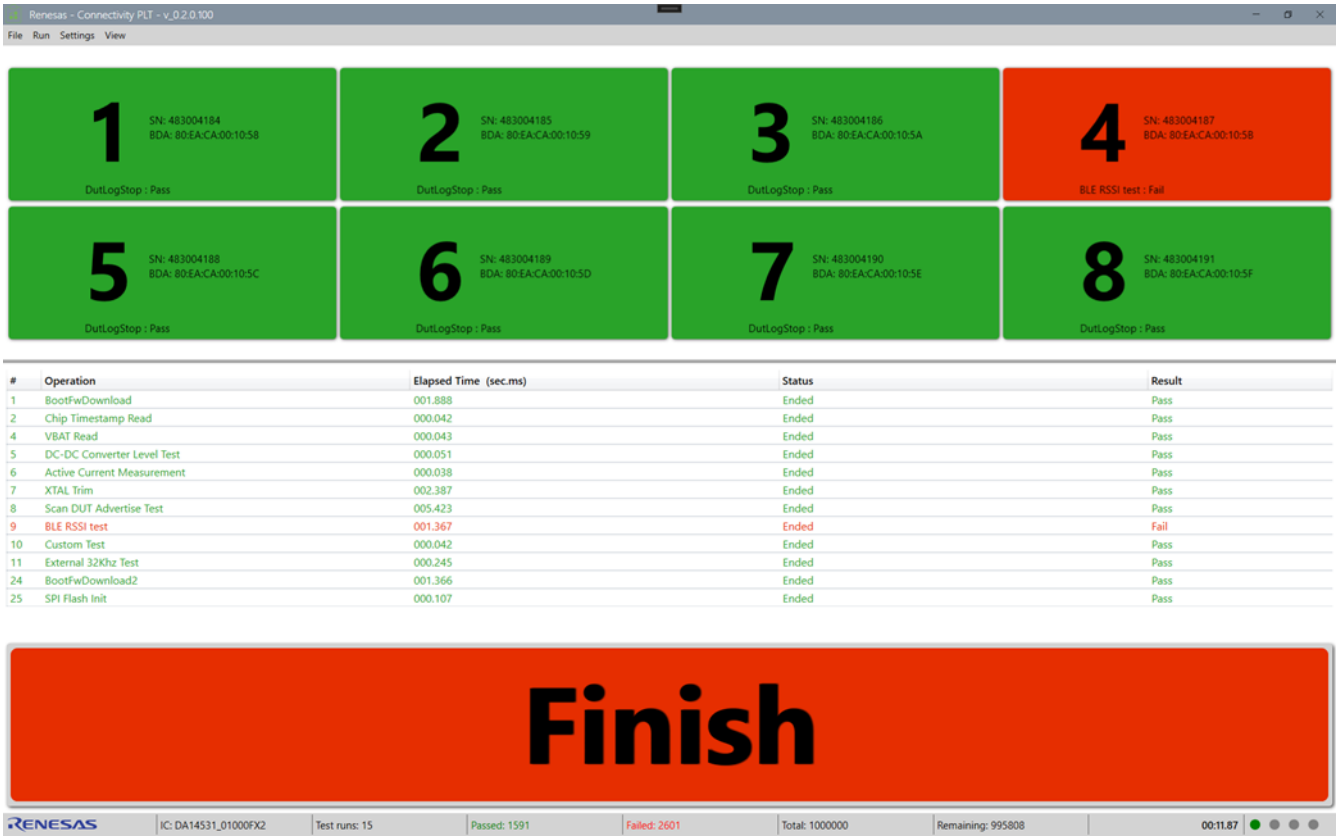



**Figure 132. Partially successful test sequence**

6. **Test Logs viewing**

   After the termination of the test sequence, clicking any tile brings up the log file for the specific DUT (Figure 133) or by opening the logs folder under the installation folder (Figure 134).

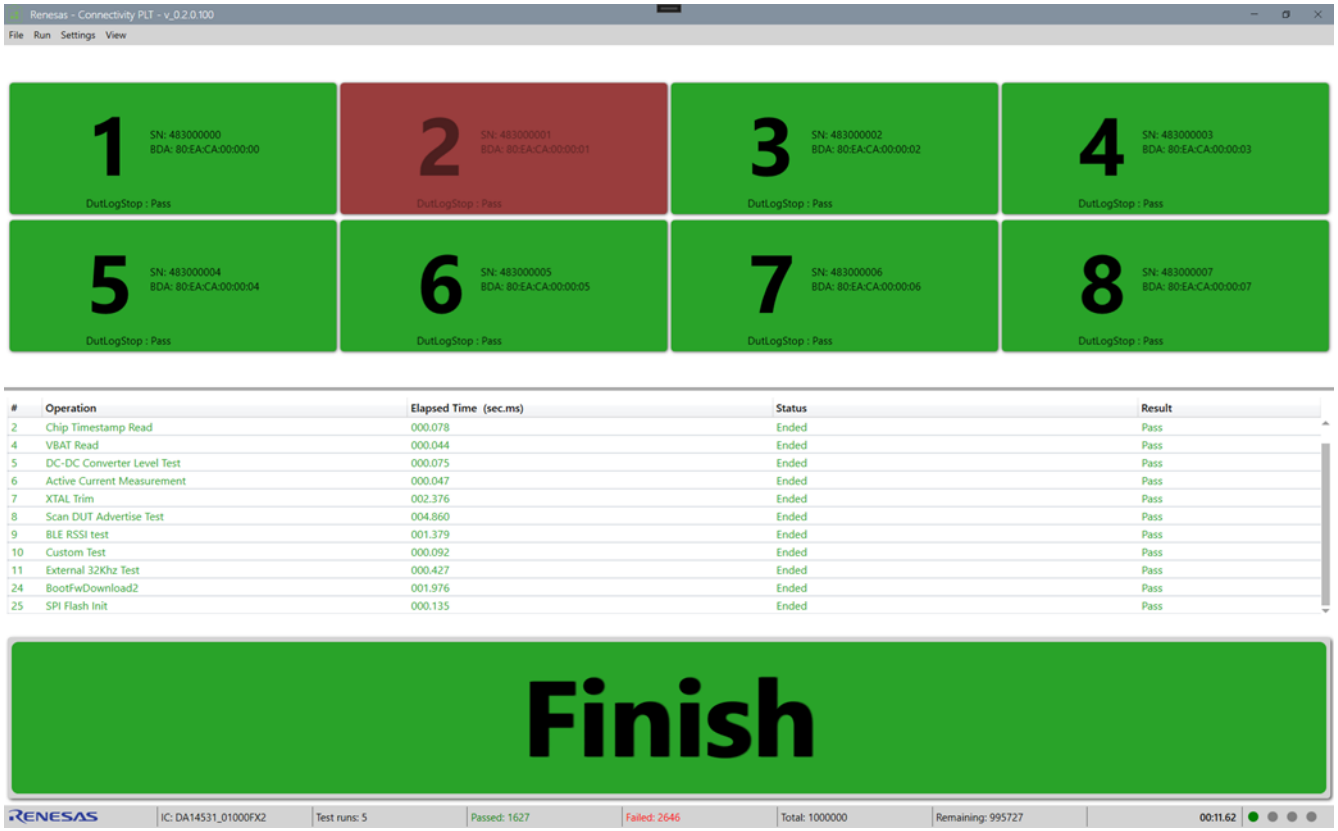



**Figure 133. The selected DUT tile**

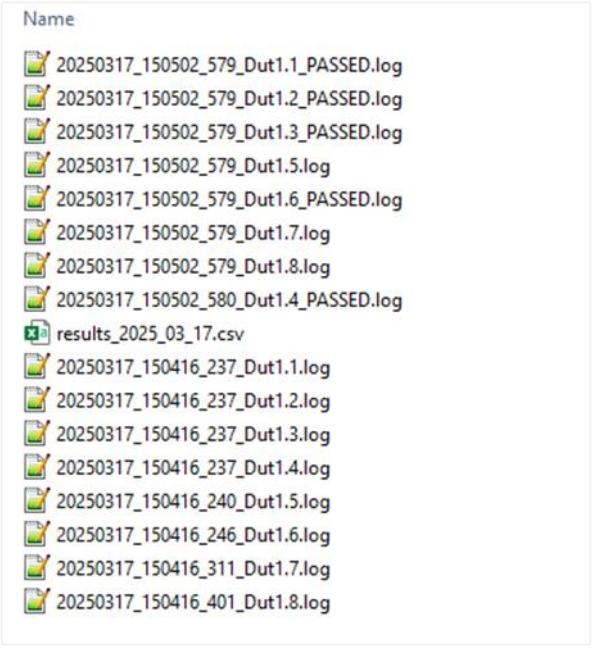



**Figure 134. Example of the log folder contents**

## 7. DUT Log viewing

The log file for each DUT contains a detailed log of the test process and the test results. It can be found in the logs folder or by clicking the DUT tile at the completion of the test process.

A header is included in the log file providing details about connectivity PLT hardware and the software. It includes the firmware and software versions, the station name and test dates and times. It also provides information about DUT, such as the physical connection to the Connectivity PLT hardware and the virtual COM port used, the serial number and BD address assigned. See Figure 135.

```
SoftwareName: Renesas_Connectivity_PLT_CoreService.exe
SoftwareVersion: 0.2.0.100
Date: 2025-04-04
CommunicationInterfaceName: TBA
BleAddress: 80:EA:CA:00:00:00
TestStationId: 01
SerialNumber: 483000000
DutNumber: 1.1
StartTime: 19:21:51.045
EndTime: 19:22:21.841

|<time>        |<operation>                                |<action>      |<result>  |<info>
#########################################################################################################################################################
|19:21:51.069  |RftuListGet                                |Initialized   |Pass      |
|19:21:51.095  |RftuListGet                                |Started       |Pass      |
|19:21:51.108  |RftuListGet                                |Ended         |Pass      |
|19:21:51.118  |RftuBleListValidate                        |Initialized   |Pass      |
|19:21:51.122  |RftuBleListValidate                        |Started       |Pass      |
|19:21:51.129  |RftuBleListValidate                        |Ended         |Pass      |
|19:21:54.486  |DutPowerDisable                            |Initialized   |Pass      |
|19:21:54.490  |DutPowerDisable                            |Started       |Pass      |Switch off DUT VBAT power
|19:21:54.532  |DutPowerDisable                            |Ended         |Pass      |
|19:21:54.580  |PltBoardSerialNumberGet                    |Initialized   |Pass      |
|19:21:54.583  |PltBoardSerialNumberGet                    |Started       |Pass      |
|19:21:54.607  |PltBoardSerialNumberGet                    |Ended         |Pass      |Serialnumber=[20322050-3055-4b4a-2039-303236303331]
|19:21:54.616  |PltBoardFtdiInit                           |Initialized   |Pass      |
|19:21:54.620  |PltBoardFtdiInit                           |Started       |Pass      |
|19:22:06.624  |PltBoardFtdiInit                           |Ended         |Pass      |
|19:22:06.627  |DutEnumerate                               |Initialized   |Pass      |
|19:22:06.630  |DutEnumerate                               |Started       |Pass      |
|19:22:07.597  |DutEnumerate                               |Ended         |Pass      |Communication interface found=[COM5]
|19:22:07.601  |DutCommsInterfaceUpdate                    |Initialized   |Pass      |
|19:22:07.607  |DutCommsInterfaceUpdate                    |Started       |Pass      |Communication interface is=[COM5]
|19:22:07.612  |DutCommsInterfaceUpdate                    |Ended         |Pass      |Communication interface is=[COM5]
|19:22:07.615  |RftuBleEnumerate                           |Initialized   |Pass      |
|19:22:07.618  |RftuBleEnumerate                           |Started       |Pass      |
|19:22:07.625  |RftuBleEnumerate                           |Ended         |Pass      |
|19:22:07.629  |DutCommsConnect                            |Initialized   |Pass      |
|19:22:07.633  |DutCommsConnect                            |Started       |Pass      |Connect DUT interface to PC, through the FTDI and FPGA
|19:22:07.687  |DutCommsConnect                            |Ended         |Pass      |
|19:22:07.691  |DutCommsDisconnect                         |Initialized   |Pass      |
|19:22:07.692  |DutCommsDisconnect                         |Started       |Pass      |Disconnect DUT interface from PC. Close connection in FPGA and close PC operating system handlers
|19:22:07.827  |DutCommsDisconnect                         |Ended         |Pass      |
|19:22:07.830  |PltBoardFpgaSpiConfiguration               |Initialized   |Pass      |
|19:22:07.835  |PltBoardFpgaSpiConfiguration               |Started       |Pass      |SPI bits per transfer=[8]. Clock mode=[PolarityLowPhaseLow]. FrequencyKhz=[400]
|19:22:07.856  |PltBoardFpgaSpiConfiguration               |Ended         |Pass      |
|19:22:07.858  |Ina231Init                                 |Initialized   |Pass      |
|19:22:07.861  |Ina231Init                                 |Started       |Pass      |
|19:22:07.904  |Ina231Init                                 |Ended         |Pass      |
|19:22:07.907  |Delay                                      |Initialized   |Pass      |
|19:22:07.910  |Delay                                      |Started       |Pass      |Delay=[100]ms
|19:22:08.026  |Delay                                      |Ended         |Pass      |
|19:22:08.033  |PltBoardVoltageVddioSet                    |Initialized   |Pass      |
```

**Figure 135. DUT log example**

For the DUTs that have passed all tests, the log file is renamed with the word "_PASSED" 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 file is updated with the status of each test 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.

8. **CSV Log viewing**

In addition to the individual DUT logs, all results are summarized in a continuous CSV log file. The CSV logs are located in the same folder as the DUT logs and are rotated in a daily basis. Every CSV file has a filename indicating the date created, for example "results_2024_10_15.csv".

| DUT | Start Time | Stop Time | BootFwDownload | TimestampRead | Data | VbatRead | VbatVoltage | DcDcConverterLevelTest | DcDcVoltage |
|---|---|---|---|---|---|---|---|---|---|
| 1.1 | 02/21/2025 12:13:19 | 02/21/2025 12:13:29 | Pass | Pass | 00-00-00-00-08-AA-02-00-11-82-56-1C-FF-FF-FF-FF-FF-FF-FF-FF | Pass | 3.331 | Pass | 1.106 |
| 1.6 | 02/21/2025 12:13:19 | 02/21/2025 12:13:29 | Pass | Pass | 00-00-00-00-06-AA-02-00-53-CB-CA-1A-FF-FF-FF-FF-FF-FF-FF-FF | Pass | 3.319 | Pass | 1.103 |
| 1.7 | 02/21/2025 12:13:19 | 02/21/2025 12:13:29 | Fail | | | | | | |
| 1.8 | 02/21/2025 12:13:19 | 02/21/2025 12:13:29 | Fail | | | | | | |
| 1.2 | 02/21/2025 12:13:19 | 02/21/2025 12:13:29 | Fail | | | | | | |
| 1.5 | 02/21/2025 12:13:19 | 02/21/2025 12:13:29 | Pass | Pass | 00-00-00-00-05-AA-02-00-EF-C8-CA-1A-FF-FF-FF-FF-FF-FF-FF-FF | Pass | 3.313 | Pass | 1.104 |
| 1.3 | 02/21/2025 12:13:19 | 02/21/2025 12:13:29 | Fail | | | | | | |
| 1.4 | 02/21/2025 12:13:19 | 02/21/2025 12:13:29 | Pass | Pass | 00-00-00-00-0C-AA-02-00-1B-33-56-1C-FF-FF-FF-FF-FF-FF-FF-FF | Pass | 3.319 | Pass | 1.099 |

**Figure 136. CSV log example (part 1)**

| ActiveCurrentMeasurement | Current | XtalTrim | TrimValue | BleRssi | Rssi | CustomTest | Ext32khzTest | SpiFlashInit |
|---|---|---|---|---|---|---|---|---|
| Pass | 0.579784845467502 | Pass | 0x00000079 | Pass | -54.79 | Pass | Pass | Pass |
| Pass | 0.599777426345692 | Pass | 0x00000074 | Pass | -37.36 | Pass | Pass | Pass |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| Pass | 0.609773716784787 | Pass | 0x00000073 | Pass | -36.86 | Pass | Pass | Pass |
| | | | | | | | | |
| Pass | 0.579784845467502 | Pass | 0x0000007D | Pass | -46.82 | Pass | Pass | Pass |
| Pass | 0.579784845467502 | Pass | 0x0000007B | Pass | -55.29 | Pass | Pass | Pass |

**Figure 137. CSV log example (part 2)**

## 9.3 Error Handling

Connectivity PLT switched to a demo mode when various configuration and hardware issues occur. Demo mode has no functionality, it only displays a basic setup screen, Figure 138.

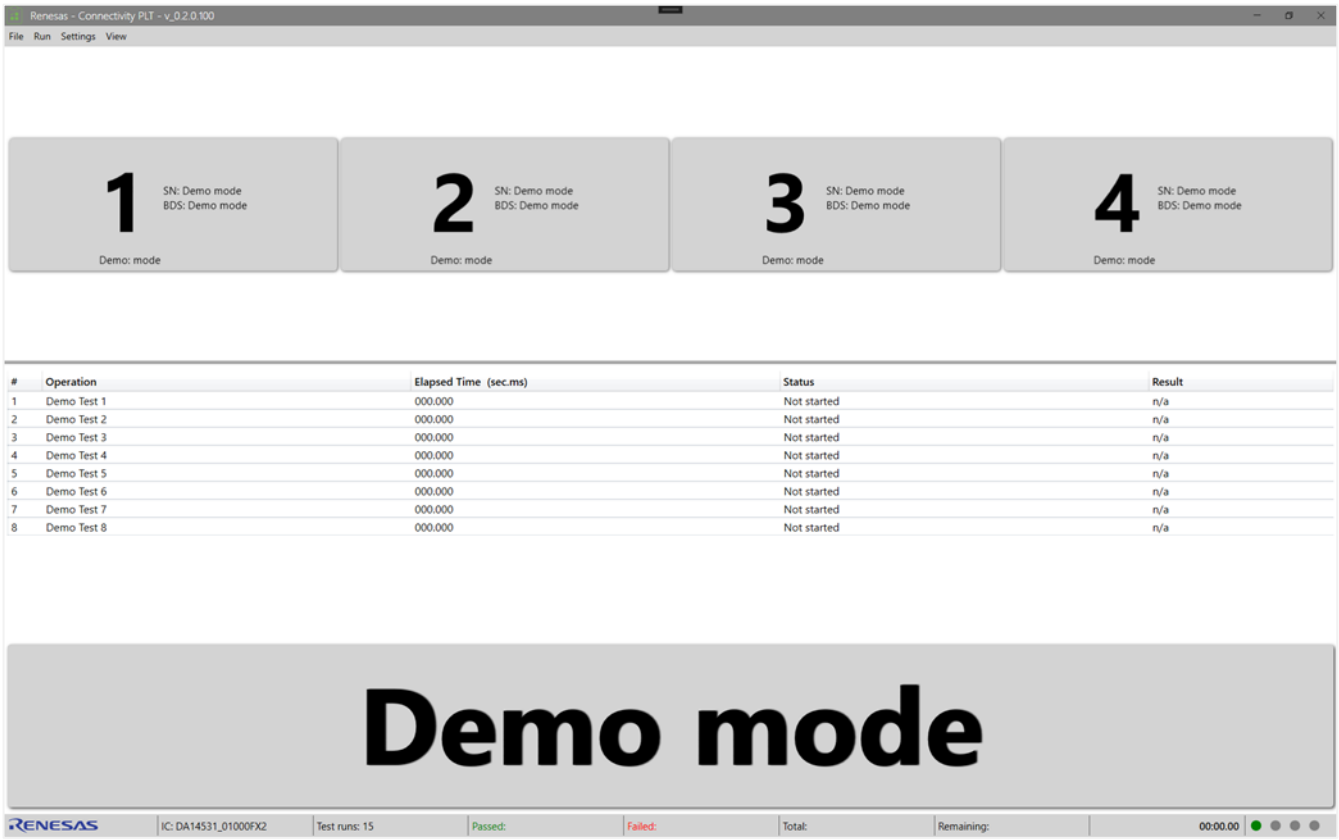



**Figure 138. Demo mode**

In some cases where there are issues in the initialization, a reload of the configuration might help resolve them. To reload the configuration, go to **File** > **Reload Configuration** or press **F5**. Common causes of a failed initialization are communication errors between the different services that can happen on the first run of the application or connectivity issues with hardware. If a reload of the configuration does not solve the issue, check hardware connectivity and that the Connectivity PLT Core Service and Connectivity PLT Database Service are running.

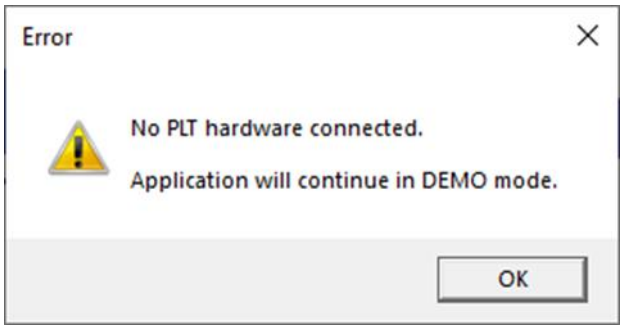



**Figure 139. Hardware communication error**

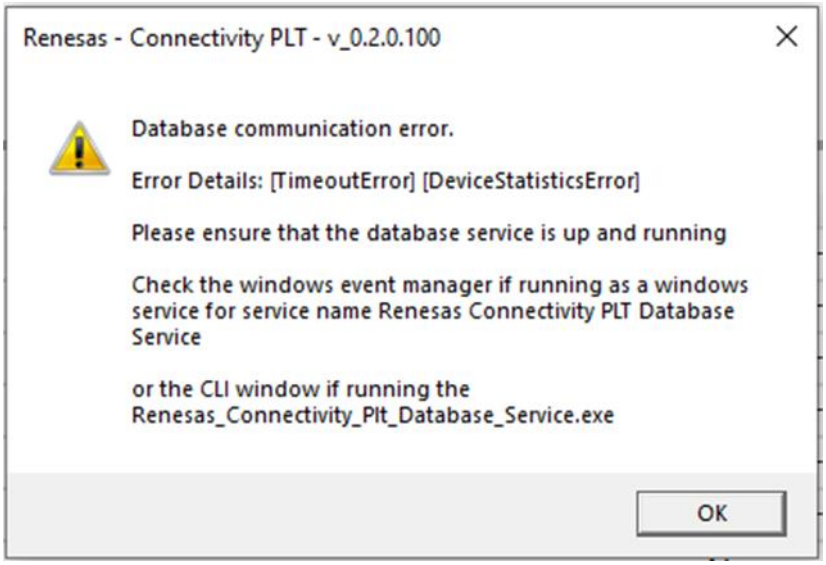



**Figure 140. Database communication error**

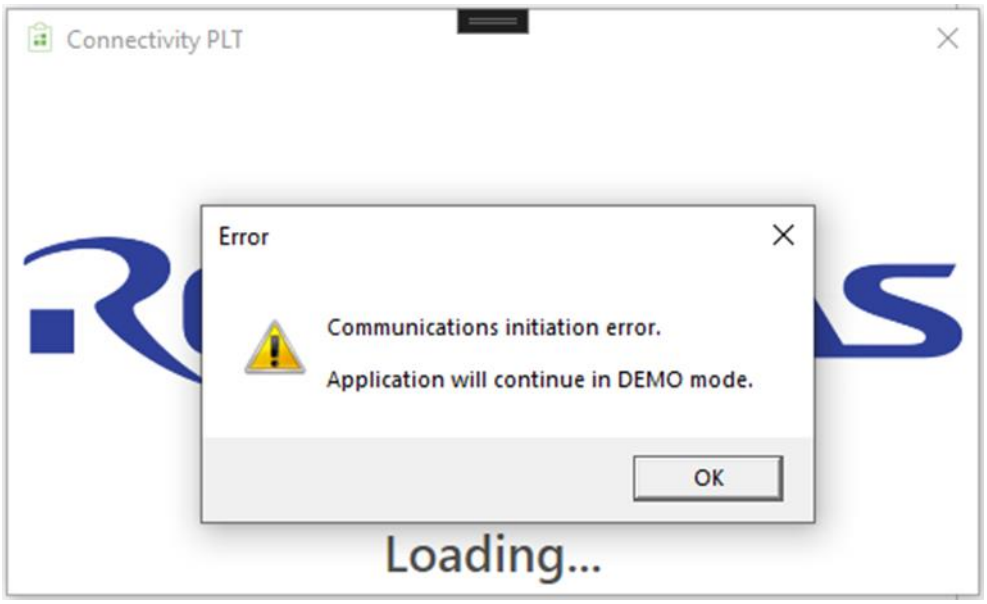



**Figure 141. Core Service communication error**

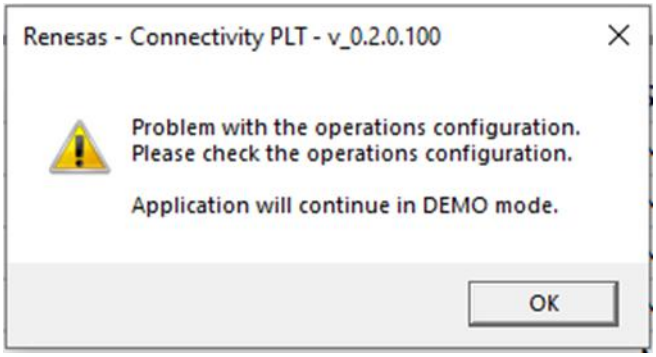



**Figure 142. Operations configuration error**

# 10. Connectivity PLT CLI Application

The Connectivity PLT CLI application (`Renesas_Connectivity_PLT_Cli.exe`) is a command line Interface application with functionality and features similar to the Connectivity PLT Application. In addition to the standard tasks for starting and monitoring the test process – based on the same configuration structure – there are additional commands available that can aid interoperability with other applications.

The application can be started from the Start menu shortcut or by executing `Renesas_Connectivity_PLT_Cli.exe` in the installation folder. A command menu is displayed when the application is started with no parameters Figure 143. All the commands stated in this menu can also be used as parameters to automatically execute from the command line.

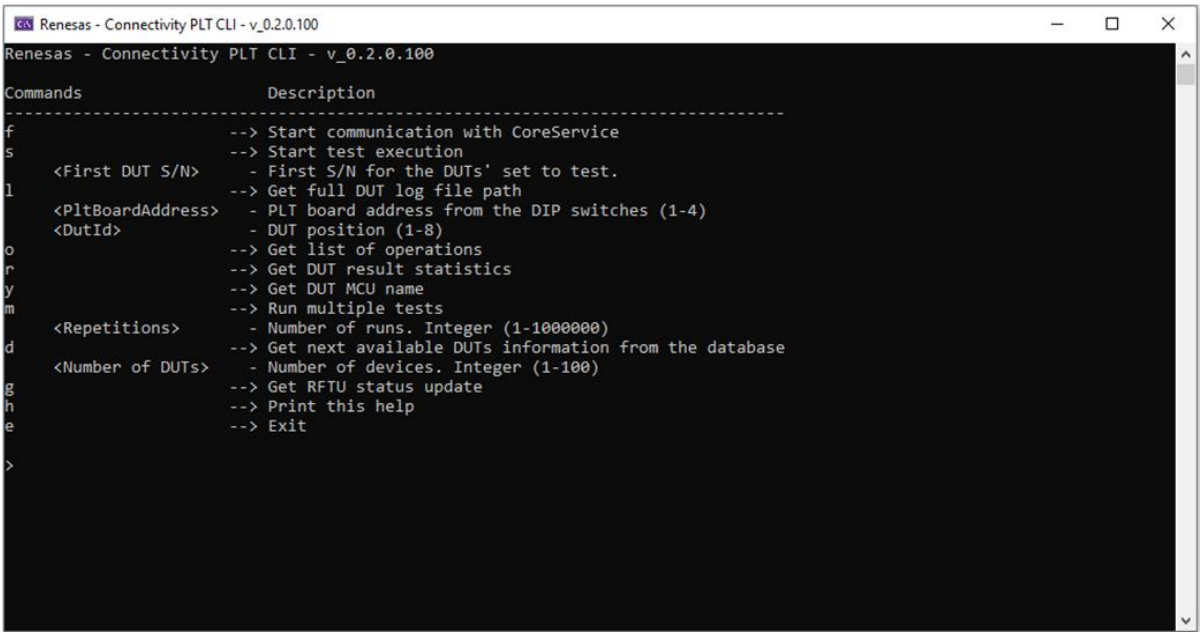



**Figure 143. Execution CLI start screen**

**Table 76. CLI commands**

| Command | Arguments | Description |
|---|---|---|
| `f` | - | Establishes communication with CoreService |
| `s` | First DUT serial number | Starts the test sequence |
| `l` | <PltBoardAddress> <DutId> | Returns the full log file path of a specific DUT.<br>Log files are accessible only after running the test sequence. |
| `o` | - | Returns the operations test sequence.<br>The operations list can be configured on Renesas_Connectivity_PLT_Configuration. |
| `r` | - | Returns the overall statistics: total devices, remaining devices, passed and failed DUTs |
| `y` | - | Returns the MCU name |
| `m` | <Repetitions> | Runs the test sequence on repeat, according to the number of repetitions entered |
| `d` | <Number of DUTs> | Gets the next available DUT information from the database (S/N and BD Address) |
| `g` | - | Gets the RFTU status (name, type, activity, connection) |
| `h` | - | Returns the menu options |
| `e` | - | Exits the application |

## 10.1 CLI Test Execution

The following is an example CLI test run result.

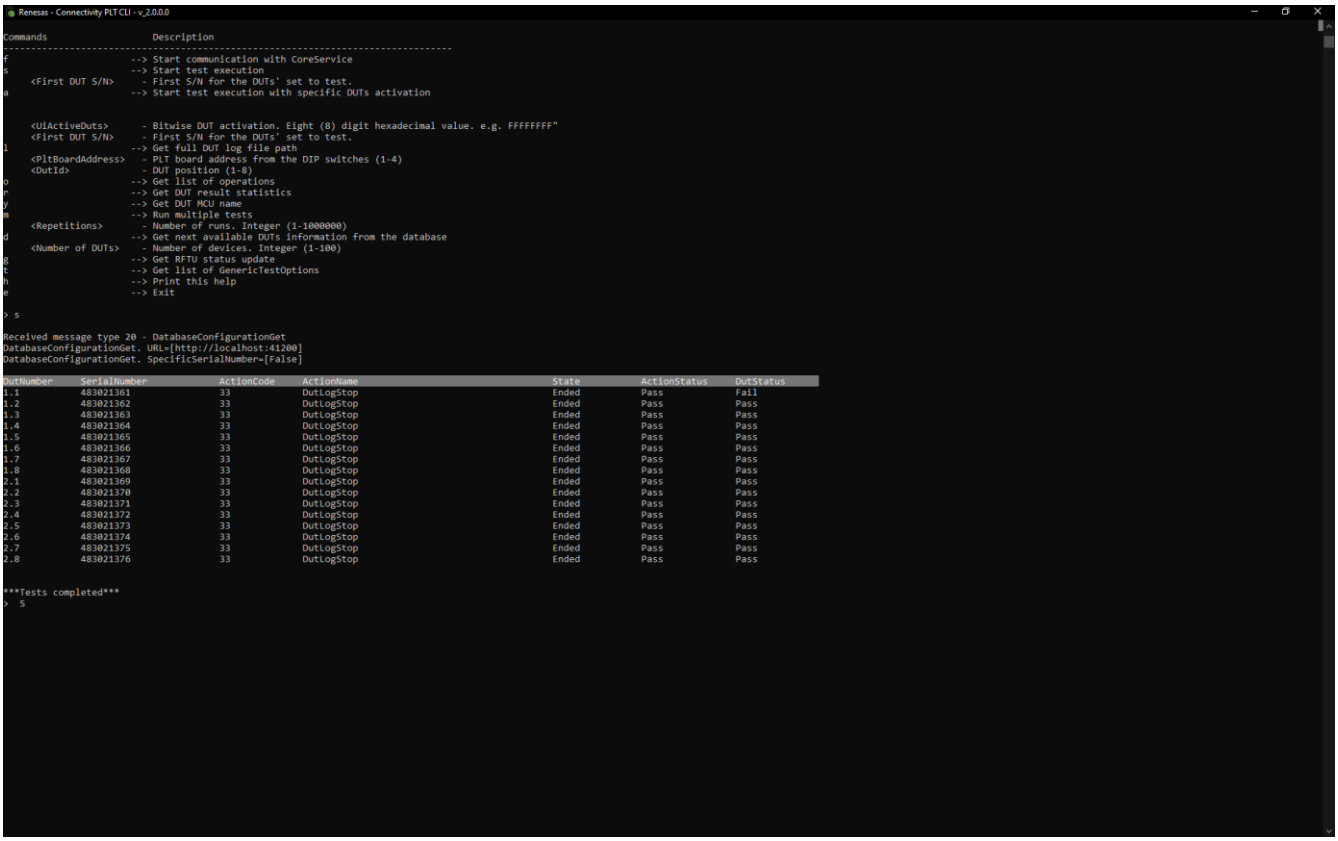



**Figure 144 CLI Test Execution**

# 11. Connectivity PLT Core Service

The Connectivity PLT Core Service is the central component of the PLT system and hosts a WebSocket service used by client applications, including the Connectivity PLT application and the Connectivity PLT CLI.

This service must be installed on the Windows machine that is directly connected to the Connectivity PLT hardware through USB. The Core Service can operate in two modes:

- **As a Windows Service** – Runs in the background automatically with no user interaction required.
- **As a CLI Application** – Runs in a command-line interface for manual control and special functions.

Under normal circumstances, the service runs silently in the background, requiring no user interaction. However, for advanced tasks or special operations within the Connectivity PLT system, the service must be run in CLI mode to provide an interface for these functions.

## 11.1 Operation in Service Mode

In Service mode, the Connectivity PLT Core Service acts as the link between the hardware and the user applications, while also serving as the executor for all operations. When a request is received from a client application, the service loads the relevant configuration files that define the parameters for the requested operation.

In addition to handling direct communication with the hardware, the service is responsible for the detection, configuration, and initialization of Connectivity PLT boards, RFTUs, and DUTs.

The Core Service remains idle until a client connects and begins issuing commands. Once a request is received, the Core Service executes the required operations, providing real-time updates to the client throughout the process. Upon completion, it returns the results to the client. The operations' sequences depends on the parameters and commands listed in the configuration files and does not depend on the continued commands form the client.

The Connectivity PLT Core Service communicates to the user front end (graphical or CLI) using messages.

Both request and reply messages consist of the Message Type ID field and parameters field. The Type Id field is an integer number, identifying the type of data. This field has the same value for both request and reply to messages.

Table 77 shows the requests and the corresponding reply messages supported by the Core Service.

**Table 77. Core service messages**

| Message type | Message type ID | Description |
|---|---|---|
| Start Test | 2 | CoreService starts test execution |
| Dut Status Update | 10 | DUT status update info is sent to the Client |
| Rftu Status Update | 11 | RFTUs' status update info is sent to the Client |
| PltBoard Configuration | 14 | PLT Boards Configuration info is sent to the Client |
| Dut Log FilePath | 15 | DUT log file path info is sent to the Client |
| Operations | 16 | Operations list info is sent to the Client |
| Device Statistics | 17 | Device Statistics info from the Database is sent to the Client |
| Dut Mcu Name | 18 | The name of the DUT's MCU is sent to the Client |
| Device Info | 19 | Device info list from the Database is sent to the Client |

Table 78 shows a list of error messages.

**Table 78. Core service error messages**

| Error message | Error ID | Description |
|---|---|---|
| NoError | 0 | Successful request, no error returned from CoreService |
| NoPltBoardError | 1 | No connected PLT board found |
| PltBoardNumberError | 2 | Error in the number of PLT boards connected |
| PltBoardAddressingError | 3 | Error in PLT Boards addressing. Probably a misconfiguration of the PLT boards' DIP switches. |
| DutDatabaseError | 4 | Error of the Database communication |
| UsbChainError | 5 | Error in the PLT board daisy chaining setup |
| DutLogFilePathError | 6 | Error in retrieving the DUT log file path |
| OperationsError | 7 | Error in getting the Operations list |
| StartTestsError | 8 | Error when attempting test execution |
| TimeoutError | 9 | Timeout error in the communication between the client and the server |
| CommsInitError | 10 | Error of CoreService - PLT Client application connection |
| DeviceStatisticsError | 11 | Error in retrieving Device Statistics from the Database |
| DutMcuNameError | 12 | Error when attempting to get the MCU name of a DUT |
| DeviceInfoError | 14 | Error in retrieving Device Info from the Database |
| RftuConfigurationError | 15 | Error in the configuration of the RFTUs |
| RftuConnectionError | 16 | Error of RFTU connection |
| SerialNumberNotFoundError | 17 | Error in getting a given Serial Number from the Database |
| OtherError | 13 | Other error cases, not described above in this table |

## 11.2 Operation in CLI Mode

If the Connectivity PLT Core Service is installed as a Windows service (see Section 6.2), it is required to stop the Windows service before starting it in CLI mode. This can be done through the Connectivity PLT Service Agent or any other of the standard Windows methods of starting and stopping services (Services Application, PowerShell, Command prompt). The service name is **Renesas Connectivity PLT CoreService**.

The Core Service can be started through the shortcut in the Start menu or by launching command prompt and executing `Renesas_Connectivity_PLT_CoreService.exe CLI` within the installation folder, note the additional parameter CLI.

Note that while in IDLE state, a client can connect to the Core Service same as if it was running in the background service, Figure 145.

Many of the functions at this stage fall outside the scope of this user manual as they are mainly used for development purposes. All the user functionality can be accessed through the dedicated Connectivity PLT CLI application (see Section 10).

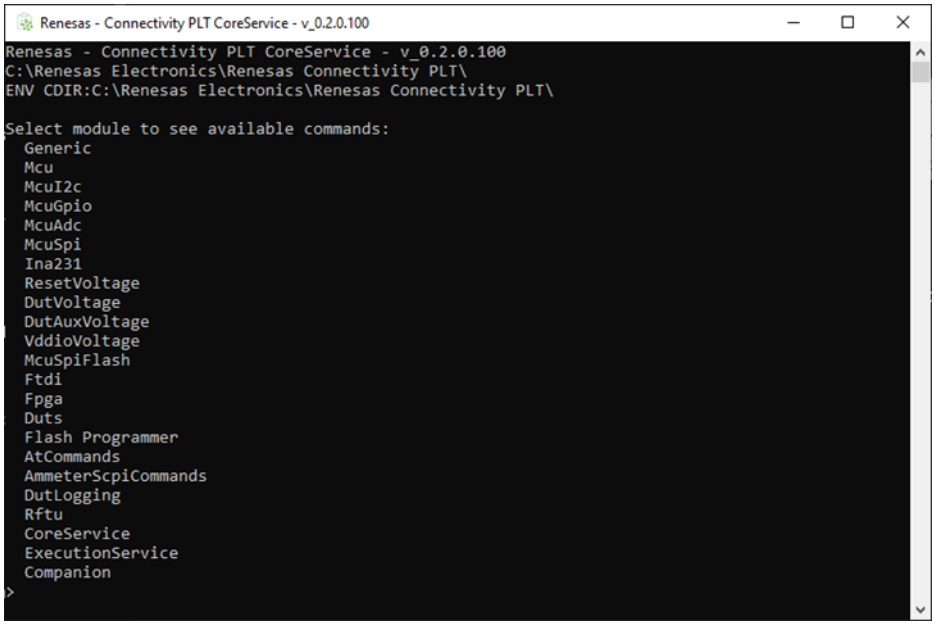



**Figure 145. Core Service in CLI mode initial screen**

Running the Core Service in CLI mode allows attaching a dedicated companion board to each Connectivity PLT main board to perform self-testing and calibration. Once the companion board is attached the test process can be initiated by the Core Service in CLI mode.

Following the main screen, type Companion to enter the companion boards test menu. Following this type StartTests 1 to start the test process. The parameter 1 is the number of iterations to perform, Figure 146.

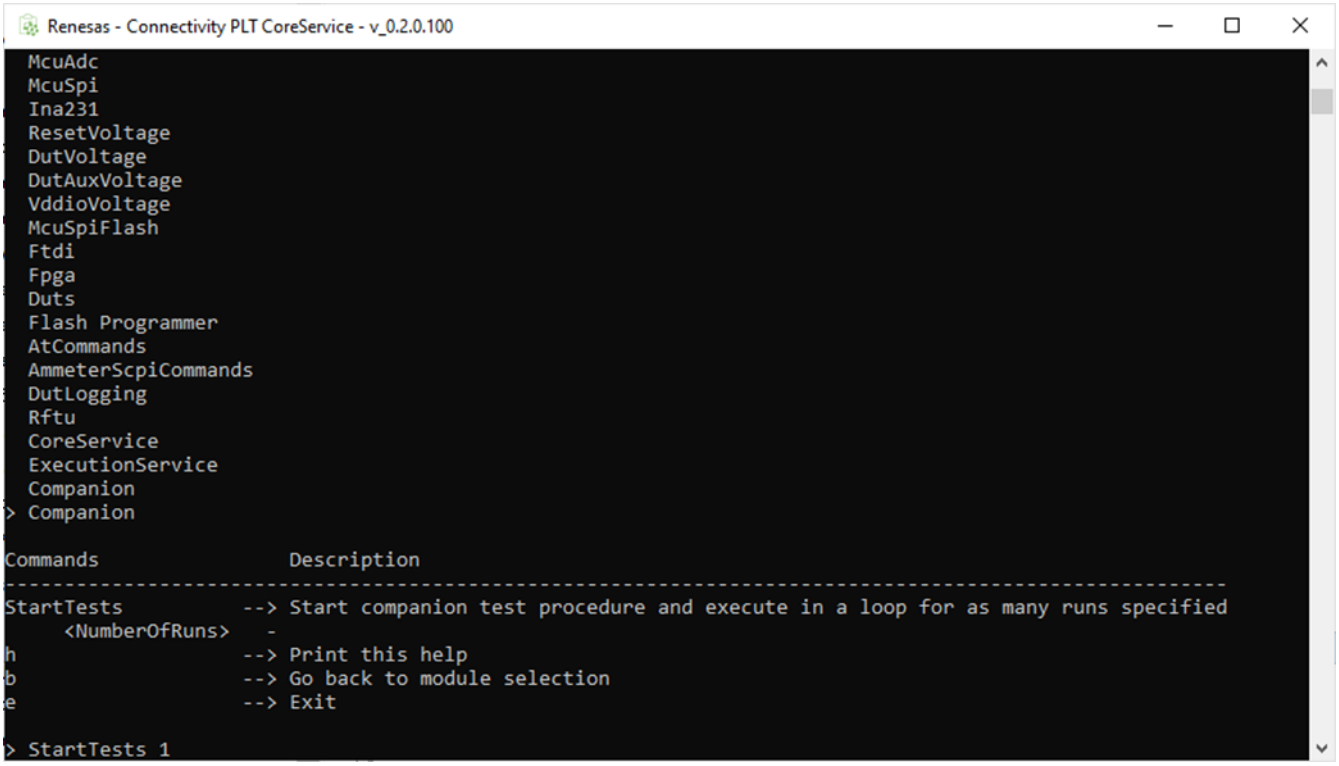



**Figure 146. Companion board tests**

The Core Service runs through a number of tests as defined in the `CompanionTests.json` in the configuration files folder. Following the completion of the test process a pass/fail message is displayed. The pass/fail criteria can be adjusted depending on the standards of the production line.

# 12. Connectivity PLT Service Agent

The Connectivity PLT Service Agent is a small application designed to be present in the system tray to provide a quick way to start and stop the Connectivity PLT Windows services as well as offer a live real time log of the Connectivity PLT Core Service execution. The application is installed automatically, and a shortcut is placed in the Windows All Users Startup folder, so that it automatically starts after a reboot of the system.

The Service Agent can be started manually from the Start menu or automatically during Windows start. The tray icon is located either directly on the visible tray or on the extended tray which can be shown by clicking the tray icon expansion arrow, Figure 147.

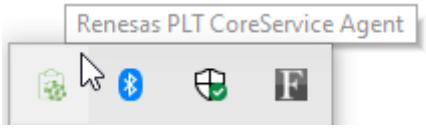



**Figure 147. Service Agent icon**

By clicking the icon, the Service Agent menu appears where the Core Service and Database services can be started and stopped, as well as a live view of the Core Service Log can be started, Figure 148.

The Core Service Console can be used to monitor the communication between the Connectivity PLT application and the Core Service as well as the status of the operations, Figure 149.

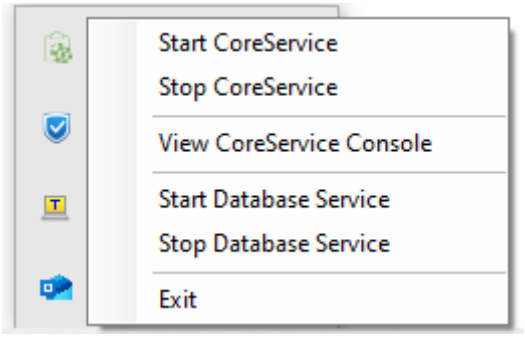



**Figure 148. Service Agent menu**

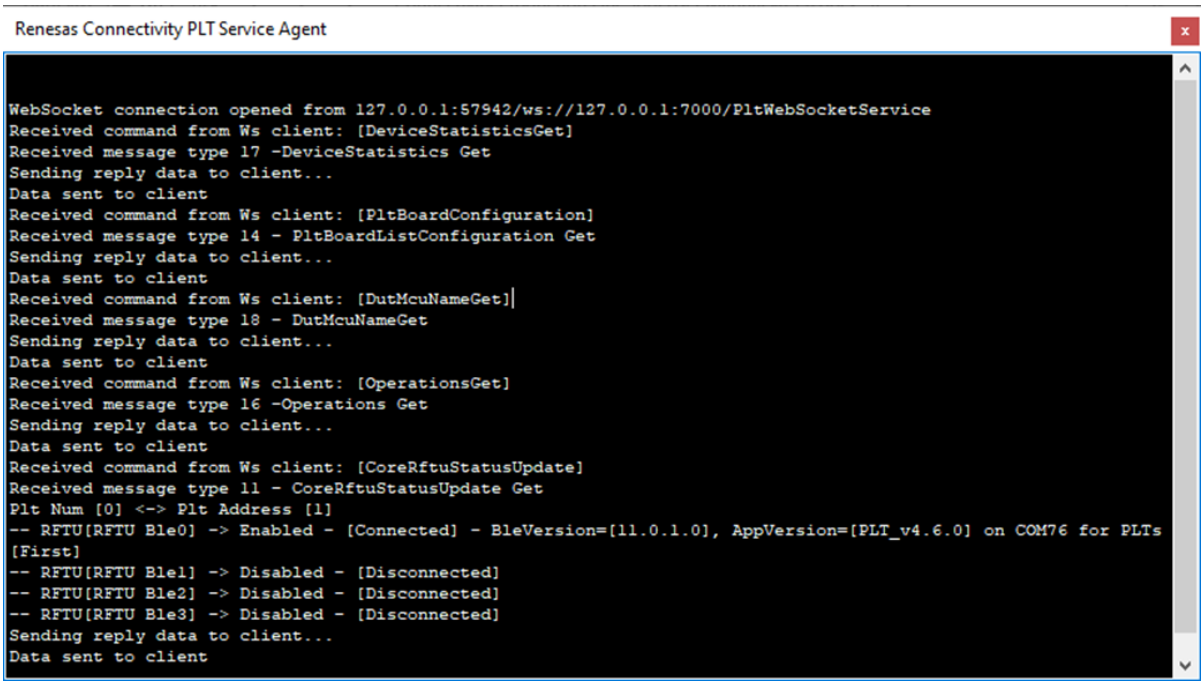



**Figure 149. Service Agent Core Service log view**

# 13. Connectivity PLT Firmware Update Application

The Connectivity PLT Firmware Update application (`Renesas_Connectivity_PLT_Firmware_Update.exe`) is a tool to update the SPI Flash memory of the MCU and the FPGA.

| NOTE |
| --- |
| For PLT Firmware Update Application to work, the SAM-Ba driver should be installed. See Section 6.4. |

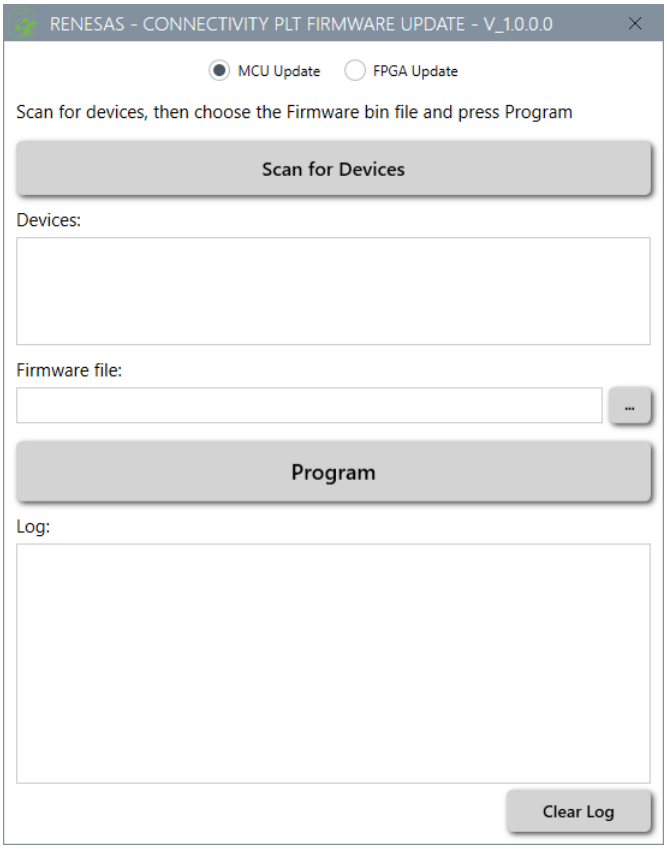



**Figure 150. Firmware update application initial screen**

## 13.1 MCU Update

To update MCU firmware:

1. In the top of **Renesas** – **Connectivity PLT Firmware Update** window, select **MCU Update**, and then click **Scan for Devices** (Figure 150).
2. If the hardware is connected correctly, you should see from one to four USB-IO devices corresponding to each connected PLT board, see Figure 151.

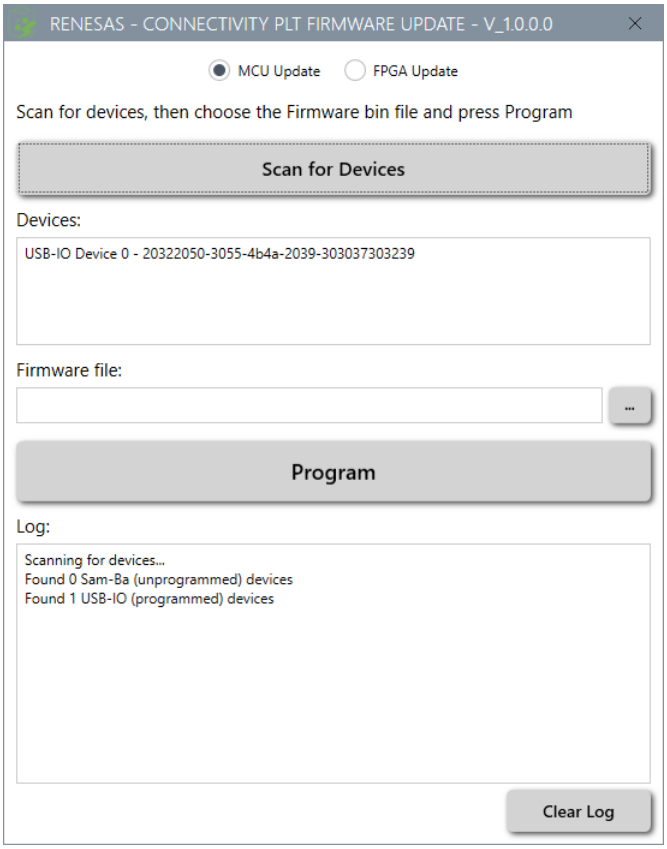



**Figure 151. Device detection**

3. Next specify the **Firmware file**, click the browse button (three dots), and select the appropriate file to flash, see Figure 152.

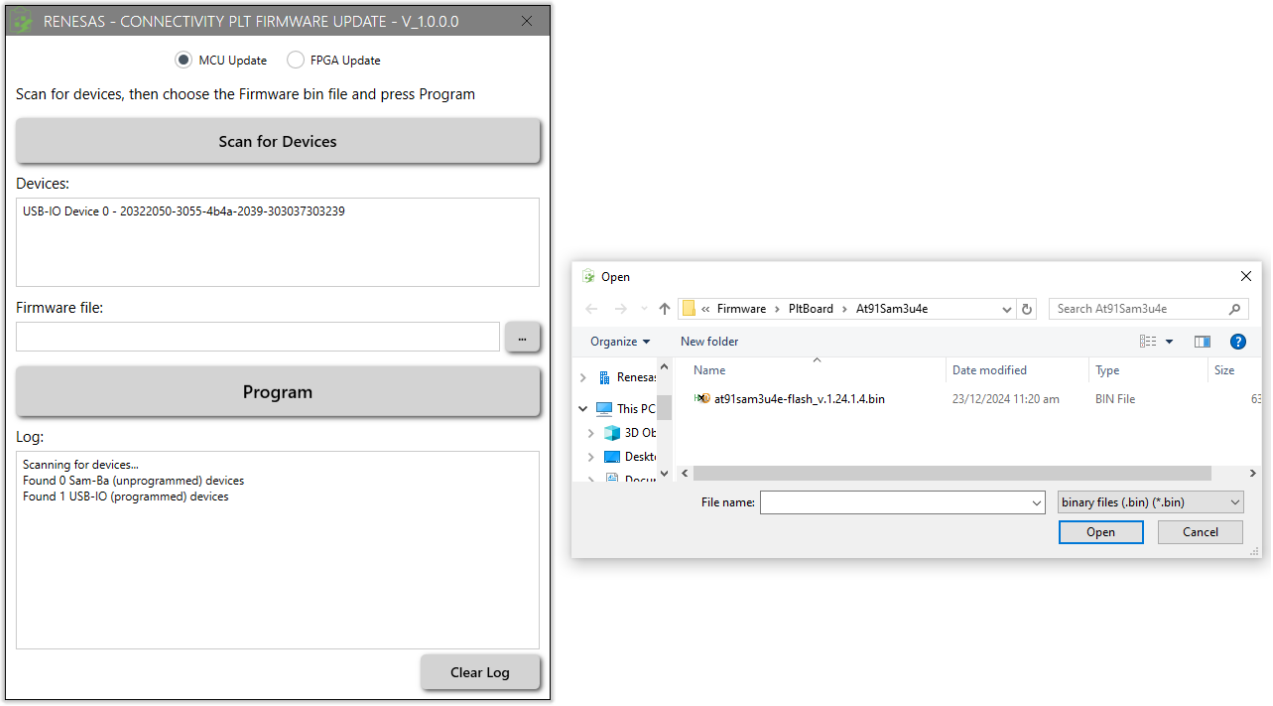



**Figure 152. Firmware update file browse**

4. Click **Program**. See Figure 153 and Figure 154.

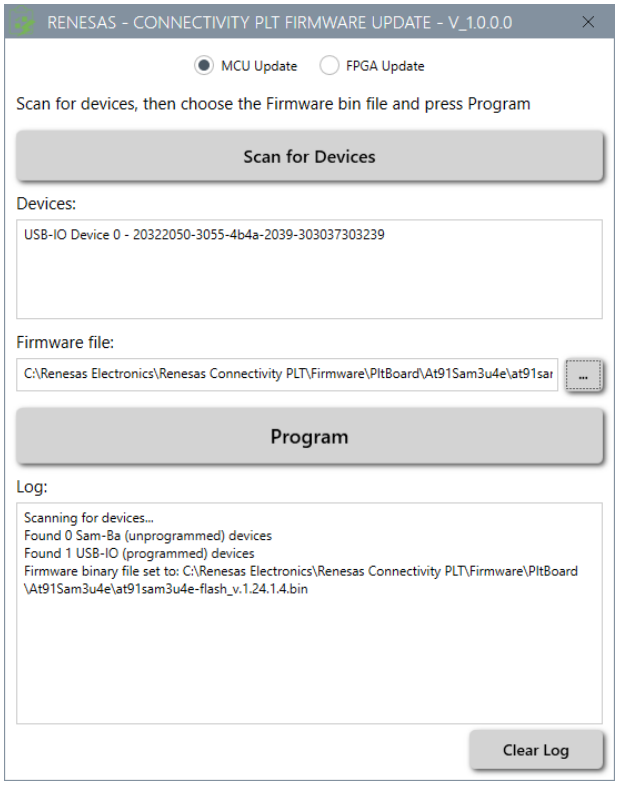



**Figure 153. Firmware update - `.bin` file loading**

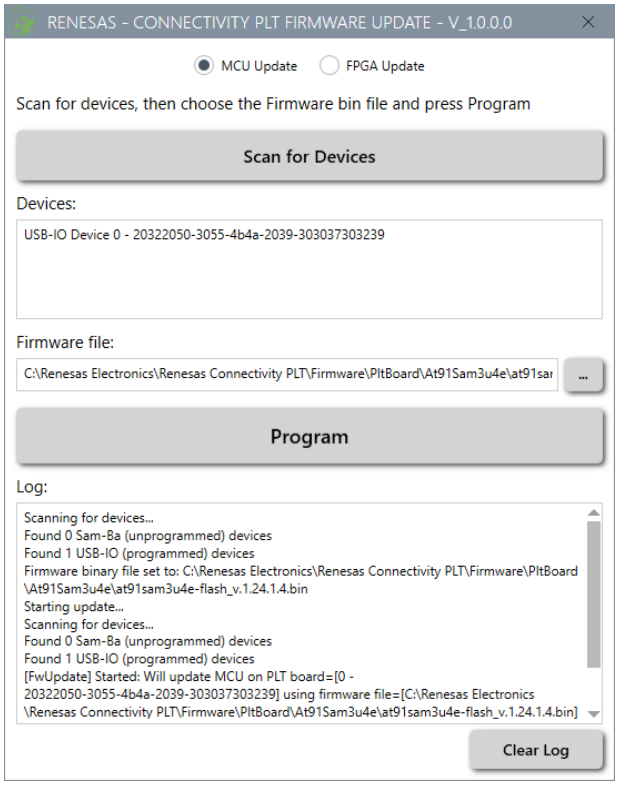



**Figure 154. Firmware update - Programing**

5. Check the log for errors.

Figure 155 shows the example log.

```
Scanning for devices...
Found 0 Sam-Ba (unprogrammed) devices
Found 1 USB-IO (programmed) devices
Starting update...
Scanning for devices...
Found 0 Sam-Ba (unprogrammed) devices
Found 1 USB-IO (programmed) devices
[FwUpdate] Started: Will update MCU on PLT board=[0 - 20322050-3055-4b4a-2039-303037303239] using
firmware file=[C:\Renesas Electronics\Renesas Connectivity PLT\Firmware\PltBoard\At91Sam3u4e
\at91sam3u4e-flash_v.1.24.1.4.bin]
[FwUpdate] BootFromRomStarted: Configuring boot from ROM, erasing flash and reseting...
[FwUpdate] BootFromRomCompleted: Completed boot from ROM and reset
[FwUpdate] FlashingStarted: Starting flashing...
[FwUpdate] Message: Found SamBa devices:
[FwUpdate] Message: COM3 - AT91 USB to Serial Converter (COM3)
[FwUpdate] Message: Will update SamBa device: COM3 - AT91 USB to Serial Converter (COM3)
[FwUpdate] SamBaScriptMessage: "arguments:"
[FwUpdate] SamBaScriptMessage: Prog_uli_with_Sam-Ba.cmd
[FwUpdate] SamBaScriptMessage: "C:\Renesas Electronics\Renesas Connectivity PLT\Firmware\PltBoard
\At91Sam3u4e\at91sam3u4e-flash_v.1.24.1.4.bin"
[FwUpdate] SamBaScriptMessage: COM3
[FwUpdate] SamBaScriptMessage: "C:\Program Files (x86)\Atmel\sam-ba_2.18\sam-ba.exe"
[FwUpdate] SamBaScriptMessage: ECHO is off.
[FwUpdate] SamBaScriptMessage: "sam-ba exit code" 1
[FwUpdate] Message: Mcu Update Successful
Scanning for devices...
Found 0 Sam-Ba (unprogrammed) devices
Found 1 USB-IO (programmed) devices
Starting update...
Scanning for devices...
Found 0 Sam-Ba (unprogrammed) devices
Found 1 USB-IO (programmed) devices
[FwUpdate] Started: Will update MCU on PLT board=[0 - 20322050-3055-4b4a-2039-303037303239] using
firmware file=[C:\Renesas Electronics\Renesas Connectivity PLT\Firmware\PltBoard\At91Sam3u4e
\at91sam3u4e-flash_v.1.24.1.4.bin]
[FwUpdate] BootFromRomStarted: Configuring boot from ROM, erasing flash and reseting...
[FwUpdate] BootFromRomCompleted: Completed boot from ROM and reset
[FwUpdate] FlashingStarted: Starting flashing...
[FwUpdate] Message: Found SamBa devices:
[FwUpdate] Message: COM3 - AT91 USB to Serial Converter (COM3)
[FwUpdate] Message: Will update SamBa device: COM3 - AT91 USB to Serial Converter (COM3)
[FwUpdate] SamBaScriptMessage: "arguments:"
[FwUpdate] SamBaScriptMessage: Prog_uli_with_Sam-Ba.cmd
[FwUpdate] SamBaScriptMessage: "C:\Renesas Electronics\Renesas Connectivity PLT\Firmware\PltBoard
\At91Sam3u4e\at91sam3u4e-flash_v.1.24.1.4.bin"
[FwUpdate] SamBaScriptMessage: COM3
[FwUpdate] SamBaScriptMessage: "C:\Program Files (x86)\Atmel\sam-ba_2.18\sam-ba.exe"
[FwUpdate] SamBaScriptMessage: ECHO is off.
[FwUpdate] SamBaScriptMessage: "sam-ba exit code" 1
[FwUpdate] Message: Mcu Update Successful
```

**Figure 155. Example for firmware update log**

## 13.2 FPGA Update

To update FPGA, follow the same steps as for the MCU update but instead select **FPGA Update** and the appropriate `.bin` file.

# Appendix A Conformity Assessment

Connectivity PLT conforms to laws and regulations that are described in the following subsections.

## A.1 FCC Notice (Applicable to Evaluation Kits not FCC-Approved)

This kit is designed to allow product developers to evaluate electronic components, circuitry, or software associated with the kit to determine whether to incorporate such items in a finished product and software developers to write software applications for use with the end product. This kit is not a finished product and when assembled may not be resold or otherwise marketed unless all required FCC equipment authorizations are first obtained. Operation is subject to the condition that this product not cause harmful interference to licensed radio stations and that this product accept harmful interference. Unless the assembled kit is designed to operate under part 15, part 18 or part 95 of this chapter, the operator of the kit must operate under the authority of an FCC license holder or must secure an experimental authorization under part 5 of this chapter.

## A.2 CE – UKCA (Europe-UK)

The Connectivity PLT is tested and found to comply with **EMC Directive (2014/30/EU)** for electromagnetic compatibility. Applicable standards are the following:

- **EN 55032 (2015)/AC (2016)/A11 (2020) & EN 55035 (2017)/A11 (2020).**

| Simplified Declaration of Conformity |
|---|
| Hereby, Renesas Design Netherlands B.V. declares that The Connectivity PLT is in compliance with **EMC Directive (2014/30/EU)**. The full texts of the EU declaration of conformity are available at the following internet address: www.renesas.com |

## A.3 WEEE Directive (2012/19/EU)

**The Waste Electrical and Electronic Equipment Regulations 2013**

### For customers in the UK and European Union

The WEEE (Waste Electrical and Electronic Equipment)

regulations put responsibilities on producers for the collection and recycling or disposal of electrical and electronic waste. Return of WEEE under these regulations is applicable in the UK and European Union.

This equipment (including all accessories) is not intended for household use. After use, the equipment cannot be disposed of as household waste, and the WEEE must be treated, recycled, and disposed of in an environmentally sound manner.

Renesas Electronics Europe GmbH can take back the end of line equipment. Register for this service at https://www.renesas.com/eu/en/support/regional-customer-support/weee.

## A.4 RoHS Compliance

Renesas Electronics' 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.

**Table 79. Product standards**

| Revision | Description |
|---|---|
| ELOT EN 55032 E2:2015 +A11:2020 | Electromagnetic compatibility of multimedia equipment - Emission Requirements |
| ELOT EN 55035 E1:2017 +A11:2020 | Electromagnetic compatibility of multimedia equipment - Immunity Requirements |

**Table 80. Test standards**

| Revision | Description |
|---|---|
| ELOT EN 61000-4-2 E2:2009 | Electromagnetic compatibility (EMC) -- Part 4-2: Testing and measurement techniques - Electrostatic discharge immunity test |
| ELOT EN IEC 61000-4-3 E4:2020 | Electromagnetic compatibility (EMC) -- Part 4-3: Testing and measurement techniques - Radiated, radio-frequency, electromagnetic field immunity test |
| ELOT EN 61000-4-8 E2:2010 | Electromagnetic compatibility (EMC) -- Part 4-8: Testing and measurement techniques - Power frequency magnetic field immunity test |

# 14. Revision History

| Revision | Date | Description |
|---|---|---|
| 01.10 | July 9, 2026 | Updated:<br>▪ Figure 1, Figure 3, Figure 6<br>▪ Table 6, Table 28, Table 29, Table 37, Table 49, Table 54, Table 55, Table 56<br>▪ Section 8.3, Section 8.4.15, Section 9.1<br>Added:<br>▪ Figure 31<br>▪ Section 5.4.4, Section 8.3.19, Section 8.4.21-Section 8.4.36 |
| 01.03 | Nov 20, 2025 | Added section about RFTUs. Added figures for Bluetooth® LE and Wi-Fi testing unit setup.<br>Updated section Connectivity PLT Configuration with new information. |
| 01.02 | Sept 12, 2025 | Fixed J11 pinout. |
| 01.01 | Aug 14, 2025 | Added Compliance information (Appendix A). |
| 01.00 | June 3, 2025 | First version. |

## IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.1 Jan 2025)

### Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

### Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

### Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/