RTX2300

Smart ATE

User Manual

Version: 1.00

General

Information contained in this document is subject to change without notice. RTX A/S makes no warranty of any kind regarding this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. RTX A/S shall not be liable for errors contained herein or for incidental or consequential damages about the furnishings, performance, or use of this material.

Warranty

This instrument is warranted against defects in material and Workman ship for a period of one year from date of shipment. During the warranty period, RTX A/S will at its option, either repair or replace products, which prove to be defective. For warranty service or repair, this product must be returned to a service facility designated by RTX A/S. Buyer shall prepay shipping charges to RTX A/S and RTX A/S shall pay shipping charges, duties, and taxes for products returned to RTX A/S from another country. RTX A/S warrants that its software and firmware designated by RTX A/S for use with an instrument will execute its programming instructions when properly installed on that instrument. RTX A/S does not warrant that the operation of the instrument or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED.RTX A/S SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

General information

This document and the information contained, is property of RTX A/S, Denmark. Unauthorized copying is not allowed. The information in this document is believed to be correct at the time of writing. RTX A/S reserves the right at any time to change said content, circuitry and specifications.

The general safety precautions, according to the information provided in the RTX2300 User's Manual, must be observed during all phases of operation. RTX A/S assumes no liability for the customer's failure to comply with these requirements.

The purpose of the document is to provide guidance to users of the RTX2300 Smart ATE. The User's manual describes general functions of the test equipment and describes the use of the related Windows® based interface, as well as interfacing with a production application program.

Safety information

The following general safety precautions must be observed during all phases of operation and service of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. RTX A/S assumes no liability for the customer's failure to comply with these requirements.

DO NOT operate the product in an explosive atmosphere or in the presence of flammable gasses or fumes.

DO NOT use repaired fuses or short-circuited fuse holders: For continued protection against fire, replace the line fuse(s) only with fuse(s) of the same voltage and current rating and type.

DO NOT perform procedures involving cover or shield removal unless you are qualified to do so – it is therefore strongly emphasized here that operating personnel must not remove equipment covers or shields. Procedures involving the removal of covers and shields are for use by service-trained personnel at RTX A/S only.

Electrostatic Discharge

Electrostatic discharge (ESD) can damage electronic test equipment. Working with electronic components or test equipment should always be performed at a static-safe place.

High Voltage

Some power supplies can generate high voltage, which can damage the all the port of the unit. To prevent damage to the RTX2300 Smart ATE please make sure that the device is properly earthed.

Documentation information

This User's Manual contains essential items of information needed for general-purpose use of the test equipment along with a detailed description for high throughput production purposes. In this document you will find valuable information on how to unpack, install and operate your RTX2300 Smart ATE unit.

The User's Manual provides programming guidance to users of the RTX2300 Smart ATE who would like to write their own test programs. A few examples on how to use the DLL function calls in your source code are also outlined. However, please note that the examples and code fragments are included for informational reasons only and should only be used as a guidance to ease test program development. *It is therefore strongly emphasized here that RTX takes no responsibility for debugging and verification of the actual test program developed by the customer.*

Conventions Used in this Manual

The following text conventions are used in this guide:

Parameter	used to represent a parameter, value or data in an entry field
RUN	used to represent the text in the Windows® based user interface

Abbreviations Used in this Manual

The following abbreviations are used in this guide:

API	Application Programming Interface
ATE	Automatic Test Equipment
CCB	Customized Connector Block (physically the same as SCB but with customized connectors)
DECT	Digital Enhanced Cordless Telecommunications
DUT	Device Under Test
GUI	Graphical User Interface
LED	Light Emitting Diode
PSU	Power Supply Unit
QSK	Quick-Swap Kit
REPS	RTX EAI Port Server
RF	Radio Frequency
SCB	Standard Connector Block (PCB with 2 connectors – one 2x25 pin header and 3x14 pin + 6RF option + 2 pneumatic)
SCPI	Standard Commands for Programmable Instruments
SMPS	Switch Mode Power Supply
THD	Total Harmonic Distortion
DUT	Devicenit Under Test

CONTENT

1. Getting Started

A. **Introduction**

The RTX2300 Smart ATE is a highly optimized automated test solution for calibration and functional test in the production of wireless devices (including DECT, GSM, Wi-Fi, Bluetooth™, etc.). The design of the RTX2300 is based on extensive experience within development and installation of Automated Test Equipment (ATE) systems, and hence, key parameters as handling-time, flexibility and size have been considered as part of the design process.

The RTX2300 Smart ATE is a multipurpose solution to simplify and cost-reduce the production test environment dramatically compared to traditional “rack and stack” wireless test implementations. The RTX2300 reduces the complexity and size of the ATE system through the inclusion of modular instrumentation internal to the test unit, including a digital volt meter, tone generator, and signal multiplexer. Optional internal instrumentation includes a programmable power supply, and a frequency counter. These components allow for the calibration and test of the baseband and audio portions of a Device Under Test (DUT) without the need for additional external equipment.

The flexible nature of the RTX2300 Smart ATE enables a swift physical reconfiguration and it can streamline the manufacturing process, since in literally a matter of seconds, it can be physically reconfigured to accommodate different products and test types. The RF shielded compartment of the RTX2300 can be customized with a device-specific probe fixture, including a pneumatic capture unit to automatically hold the DUT in place during test. Furthermore, the RTX2300 can decrease overall product test time since it has been designed to allow several units to be set up in parallel, enabling optimal use of RF, baseband, acoustic test resource, and effectively reducing the handling time of the DUTs.

The RTX2300 Smart ATE can be used for Printed Circuit Board (PCB) as well as final product testing. It features a customization area that enables the interconnection between internal and external measurement functionality and the DUT, hence reducing both space requirements and complexity for the total test system. The flexibility of the RTX2300 simplifies the customization of the test platform, and enables economical replication. A production test system can be implemented by using one or more RTX2300 Smart ATE units in combination with an RF communication tester (such as the RTX2011/RTX2012 HS DECT/DECT 6.0/CAT-iq RF Tester or Agilent 8960) and a PC for executing the test application.

B. Unpacking the RTX2300 Smart ATE

i. Initial Inspection

Please inspect the shipping container for damage. If the shipping container or packaging material is damaged, it should be kept until the contents have been checked mechanically and electrically. If any mechanical or electrical damage is observed, please notify RTX A/S. Please refer to the description on how to contact RTX A/S provided in this document. Please also keep the damaged shipping materials (if any) for inspection by the carrier and an RTX A/S representative.

NOTE: The handle on the shield must not be used for lifting the RTX2300 out of the box.

ii. Box content

When unpacking the RTX2300 Smart ATE please verify that the items listed below are included in the box:

- RTX2300 Smart ATE unit
- Power Supply
- USB cable (for communication between the RTX2300 unit and a PC)
- Optional items (modules, fixtures, QSK kit etc.) – if ordered

C. General overview of the RTX2300 Smart ATE Unit

The RTX2300 Smart ATE unit provides a wide range of external as well as internal connectors. The connectors and functions can be divided into the following categories:

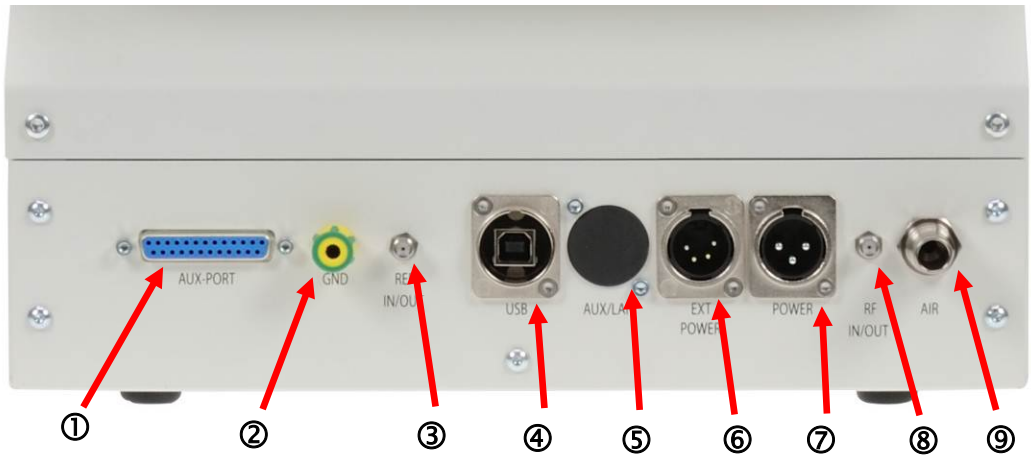
- **Rear panel Connectors**
- **Internal Connectors**
  - *Shielded Fixture Bay Connectors*
  - *Controller Chamber Connectors*
- **Front Panel Functions**

Each of these categories will be descibed in more detail in the following.

i. Rear Panel Connectors

The RTX2300 Smart ATE provides rear panel inputs/outputs for many functions. Below a brief overview of the available connectors is provided along with a description of each of the connectors.

Available rear panel



- **AUX Port (①):** Connector (type 25-pin DSUB female) for external acquisition unit measurement channels (e.g. Agilent 34970A or similar). For a detailed pin description please refer to section v on page 112. The AUX Port can be used for connecting external measurement equipment to the DUT, and it provides the following internal routing:
  - 10 voltage measurement channels (20 wires) are routed to the Customized Connector Block (CCB) selection matrix (for more information regarding the CCB block please see chapter 5)
  - One channel (2-wire) for current measurement is routed directly to the Standard Connector Block (SCB) matrix (for more information regarding the SCB block please see chapter 5)
  - One channel from the internal AD converter can be routed to the acquisition unit

- **Ground connector (②):** This ground connector is used for connecting all instrumentation together to avoid ESD problems.

- **CLK Port (③):** Connector (type SMA female) for connecting to external equipment. The port has the following functions:
  - DUT CLK output to external frequency counter (e.g. Agilent 53181A)
  - 10MHz reference in (a reference signal from an external source is used as reference for the optional internal frequency counter to measure the DUT Clock)
  - 10MHz reference out (a source signal from the optional internal frequency counter and to external equipment)

- **USB Interface (④):** This interface should be used together with a Windows® based PC to control the test set and to query data from the test set. The USB interface on the RTX2300 unit is a Type B receptacle, and hence, a standard USB cable with Type A-B plugs can be used to connect the RTX2300 unit to a PC. Please note that the external USB connector is internally connected to a 7-port USB HUB to provide both USB connection and a serial connection. Furthermore, 4 of these ports are available as USB ports for the 4 expansion slots. The USB ports support USB 1.1.

- **LAN/PSTN (⑤) (optional):** This connector (RJ45 type) is optional and is a connector for Ethernet testing options. Furthermore, this connector can also be used for PSTN/FXO/FXS testing options.

- **DUT Ext. Power supply (⑥):** Connector (type is XLR-4 male) for the external power supply of the DUT and should provide V+, V-, V+ sense and V- sense. *Please note that double pins for V+ and GND must be used.*

- **Fixture power supply connections (⑦):** Connector (type XLR-3 male) for the fixture power supply (+12V, 5A) – i.e. the main positive supply for the RTX2300 unit.

- **RF IN/OUT (⑧):** Connector (type SMA female) for the RF interface to the DUT. In the standard configuration the RF is routed directly to the SCB block to enable measurements of one antenna. If measurements of more antennas are needed an RF Switch Module (option) can be installed in the RTX2300, hence providing access to more than one antenna at the SCB/CCB block through the RF IN/OUT connector on the rear panel. When connecting the SMA male connector at the RF IN/OUT connector it must be mounted with a maximum torque of 1.4 Nm.

- **Air connector (⑨):** Connection for air supply for the pneumatic activators in the RTX2300 Smart ATE.

ii. Internal Connectors

The RTX2300 is divided into two chambers:

- **Shielded Fixture Bay Chamber** – connectors in the internal shielded chamber are used for connecting the fixture to the RTX2300.

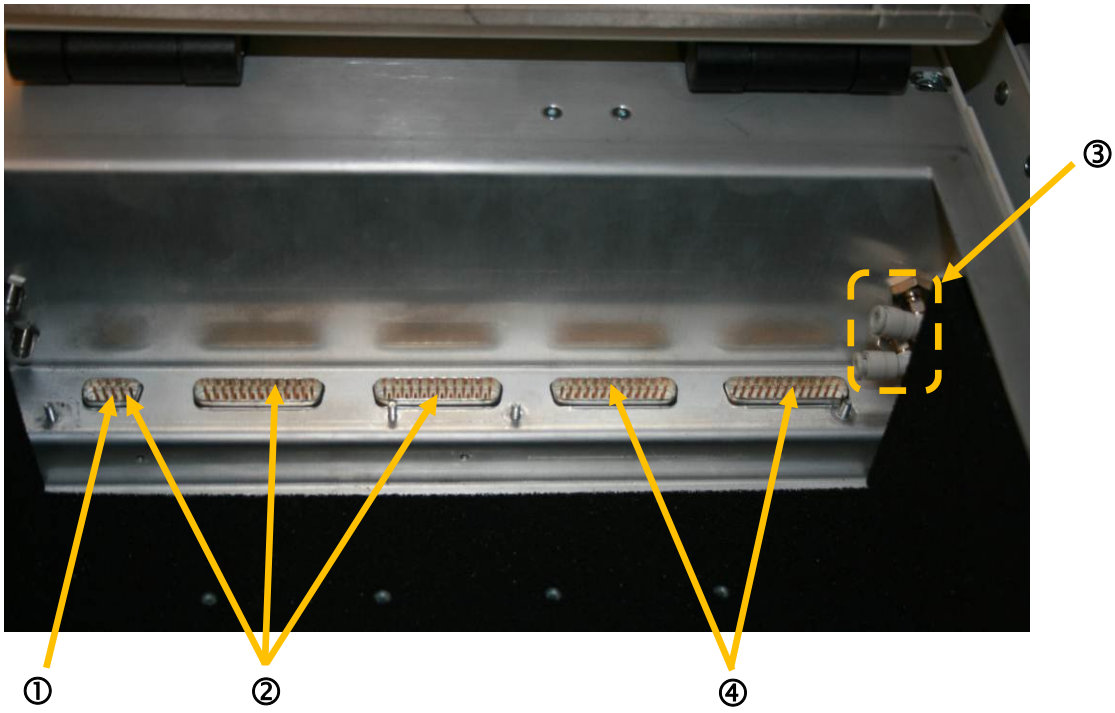
- **Controller Chamber** – connectors in the main controller chamber are used for connecting optional modules and customizing the internal routing of signals.

The connectors in each of the chambers are briefly described in the following.

Shielded Fixture Bay Connectors

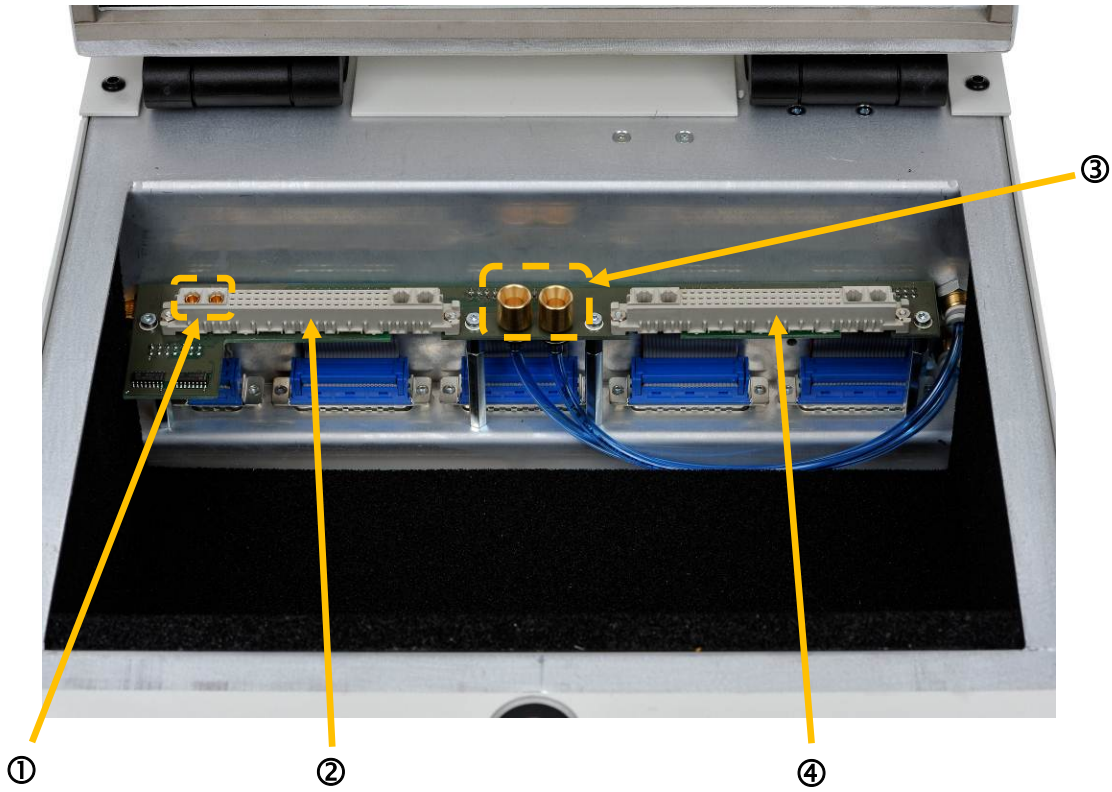
The fixture bay is prepared with interfaces to the Smart ATE functionalities and external RF equipment. The signal lines are filtered and can be accessed either at the connectors in the bottom part of the fixture bay (standard interface in the basic unit) or the optional Quick-Swap Kit (QSK) connectors. Consequently, these connectors are used for connecting the fixture to the signal lines. In the figures below the fixture bay connectors are shown in both the basic configuration and with a QSK (fixture bay part) installed.

Available Fixture Bay Connectors (basic configuration)



- **Power supply for DUT (①):** In the basic RTX2300 configuration the power supply connectors for the DUT are in the 9-pin male DSUB connector. With a Quick-Swap Kit installed the power supply for the DUT is accessible through two connectors Type M-Flat female connectors (GND and Power).

Available Fixture Bay Connectors (with fixture bay Quick-Swap Kit installed)



- **Standard Connection Block (SCB) (②):** The SCB is used for general purpose connections to the test-interface. In the basic configuration the SCB signals are distributed over the 9-pin male DSUB and two 25-pin male DSUB's. With a QSK installed the SCB connector is a 60 + 4 Type M female. Please note that the power supply for the DUT is available from this connector (as mentioned above).

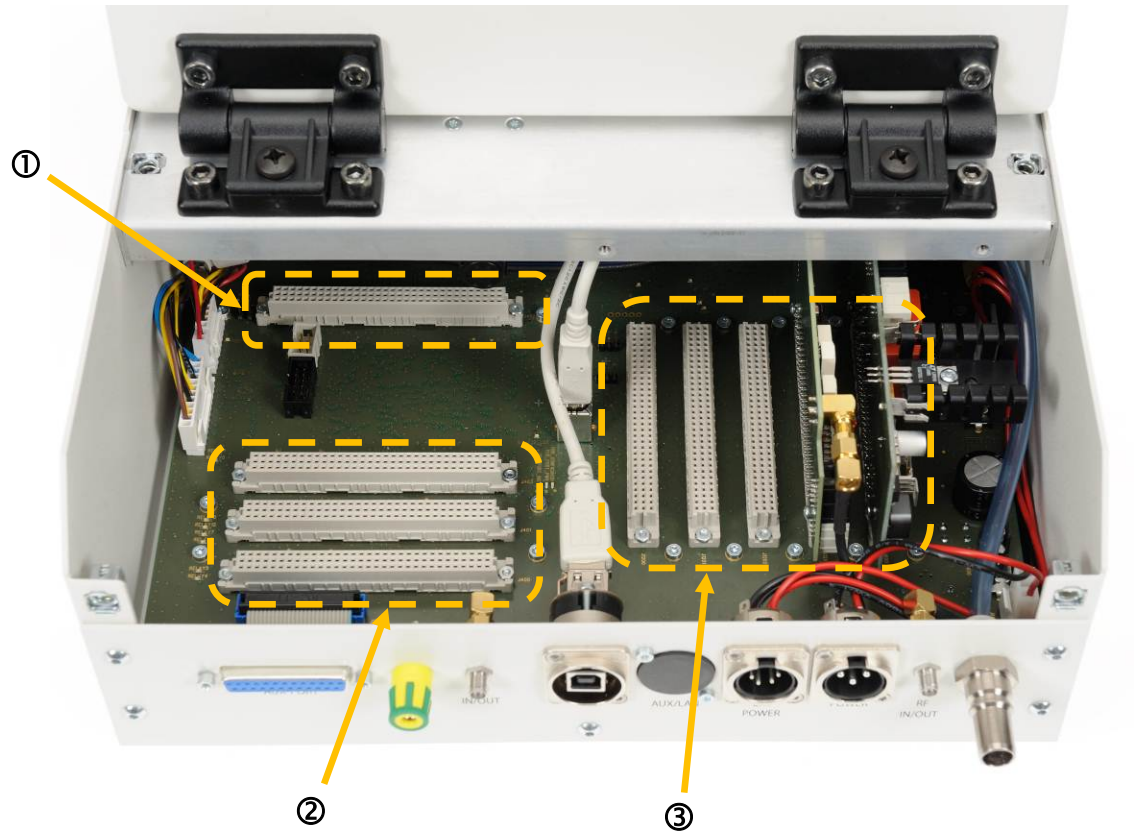
- **Pneumatic connectors (③):** Connectors for pneumatic control of fixture. In the basic RTX2300 configuration the pneumatic connectors are placed to the right on the bay wall. With a QSK installed the pneumatic connectors are placed on the QSK PCB for easy access.

- **Customization Connection Block (CCB) (④):** The CCB is used for user configurable connections to the test-interface. In the basic configuration the CCB signals are distributed over two 25-pin male DSUB's. With a QSK installed the CCB connector is a 60 + 4 Type M female.

Controller Chamber Connectors

The controller chamber can be accessed by removing the back cover (please see section 0 on page 24 for details on how to remove the back cover). In the controller chamber of the RTX2300 main board provides two main categories of connectors – connectors for modules and connectors for customization boards (input and output). Please note that two optional module cards have been installed in the figure below – i.e. they are not part of a basic RTX2300 configuration.

Controller Chamber Connectors (with two optional modules installed)



- **Customization connector (output) (①):** This connector (2 x 32 pins female) is used for the customization board for output signals (i.e. customization of the CCB functionality).

- **Customization connectors (input) (②):** In all three customization input connectors (2 x 32 pins female) are available. The connector closest to the rear panel is also connected to the external AUX-port.

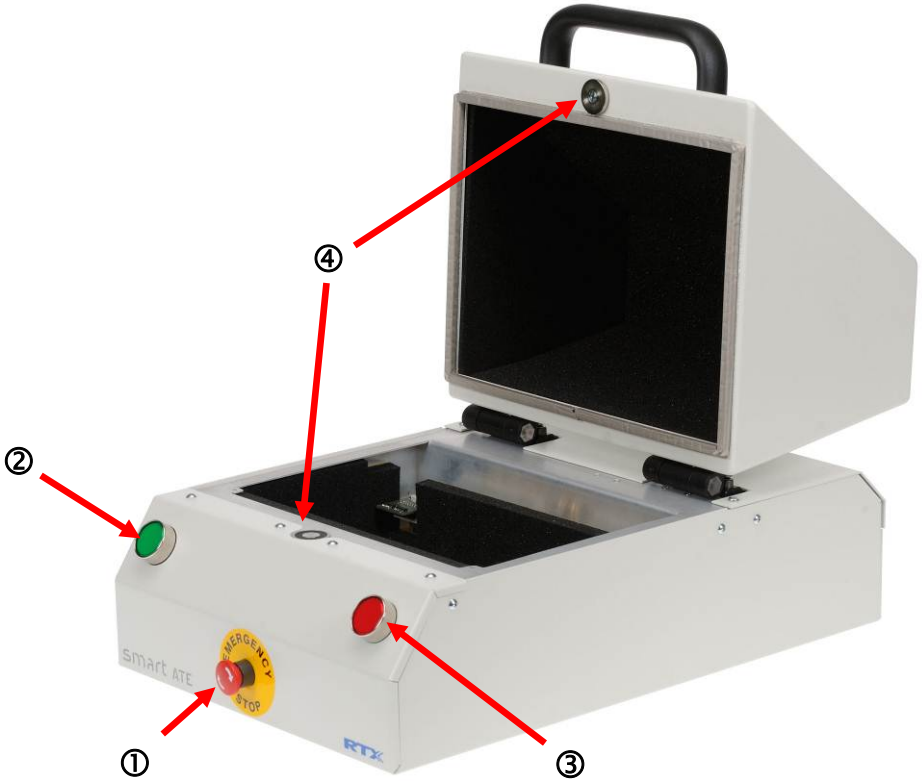
- **Extension Module connectors (③):** In total there are five connectors (4 pcs. 2 x 32 pins female and 1 pcs. 60 + 4 pins female) available for (optional) extension modules. Please note that the basic RTX2300 Smart ATE is not equipped with any extension modules.

Please note that the customization connectors also are referred to as the option matrix while the extension connectors are referred to as the instrumentation area.

iii. Front Panel Functions

The front panel on the RTX2300 Smart ATE provides many buttons for controlling the unit (e.g. start/stop of the test sequence) – each of these are briefly described below. Please note that the RTX2300 can be delivered both with (standard) or without (optional) the shielded lid.

Front panel functions



- **Emergency stop (①):** The emergency button is a safety precaution. When pressed all power to the RTX2300 is cut and all pneumatic valves controlled by the RTX2300 are released to disconnect the DUT.

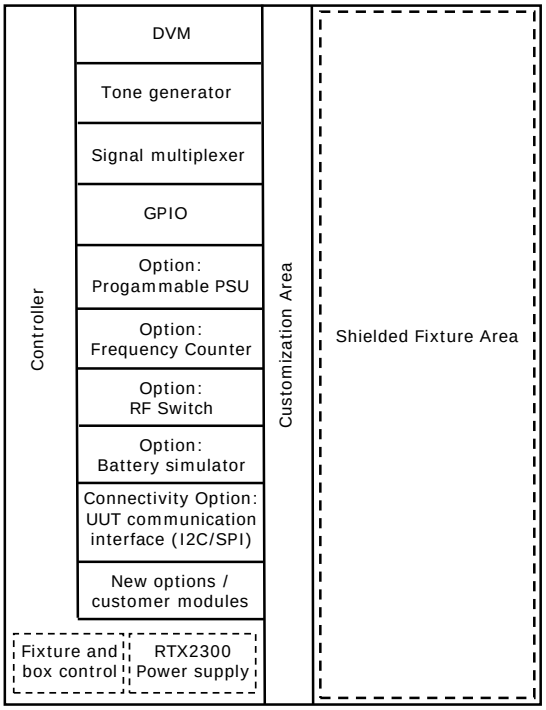
- **Push buttons / indicators (② & ③):** The two push buttons are intended to be used as Start/Stop buttons. However, the button functionality can be customized through the SW API. Furthermore, each of the buttons can be configured as normal input or as interrupt inputs with a programmable de-bounce time. The indicators in the buttons can be used for showing the status of a test sequence (i.e. PASS / FAIL). They are controlled from the SW API, and hence, the functionality of the indicators can be customized as well.

- **Top-lid closing mechanism (④):** The top-lid closing mechanism is a magnetic "lock" connected to an electrical motor which will pull down the lid. In the standard configuration the test sequence is started when closing the lid and upon completion of the test sequence the electrical motor will lift the lid. One signal for controlling this mechanism is available from the SW API.

D. **Overview of instrumentation and functional blocks**

The figure below provides an overview of the instrumentation in the standard configuration, main functional blocks of the RTX2300 and the available optional modules.

Overview of RTX2300 main functional blocks and instrumentation



The RTX2300 is a very flexible and well-equipped platform which includes the following instrumentation in the standard configuration:

- Digital AC & DC Voltmeter (DVM)
- Tone generator
- Signal multiplexer
- GPIO

In addition to the standard instrumentation the RTX2300 Smart ATE also supports installation of optional modules and features like:

- Programmable Power Supply Unit (PSU)
- Frequency counter
- RF switch
- DUT communication interface (I2C / SPI)
- New options / customer modules

The instrumentation will be outlined in more detail in the following sections including an overview of the main functional blocks. For more information on the optional modules and features listed above please refer to chapter 4 (especially section A on page 81).

i. Digital AC & DC Voltmeter (DVM)

The DVM is divided into two parts – an AC part and a DC part. The DC part is an 8-channel digital voltmeter with an operational range from -10V to 10V in 10mV steps. It can operate in two modes (static / differential) and it supports a wide range of scale configurations.

The AC part is a 2-channel digital voltmeter. It can operate in static or differential mode and it can be configured to measure RMS or peak value. A built-in attenuator enables attenuation of the input signals and distortion measurements are also supported. Please refer to the formula for THD outlined in section v on page 131.

ii. Tone Generator

The RTX 2300 has tree signal generators – one with high-level output and two with low-level outputs. The two low-level outputs can be configured to operate in differential mode. The level, frequency and configuration of the generators can be changed from the SW API.

iii. Signal multiplexer

The RTX2300 Basic unit has 8 relays for signal switching. They are all available through the option matrix (i.e. the customization connectors) and can be routed to the CCB connectors. Each relay has two sets of contacts.

In addition to the relays there are 16 connections to the expansion slots. These connections can be used to route signals from the option-board in the expansion slot to the test bed via the option matrix.

iv. GPIO

The RTX2300 Smart ATE has 16 digital output ports (8 sink outputs an 8 source outputs) and 8 digital input ports. The port state of all ports can be controlled and read from the SW API. The 8 digital inputs are 5V TTL logic and the sink and source outputs are as outlined below.

Simplified sink (left) and source (right) output schematic

In addition to the digital input and output ports the RTX2300 also includes two DACs. Please note that these DACs are only intended for DC values (i.e. they do not support audio signalling).

v. Main functional blocks in the RTX2300

From a functional perspective the functionality of the RTX2300 Smart ATE can be divided into the following categories:

- Fixture and box control
- Power supply
- Communication interfaces

Each of these categories will be outlined in more detail in the following.

Fixture and box control

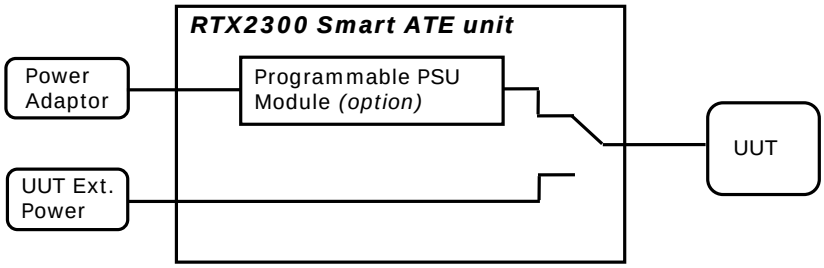
To control the fixture functionality the following signals are available:

- 4 sense signals (e.g. for detecting fixture position)
- 4 control signals (e.g. for motor or relay control)
- 1 signal for controlling the lid closing mechanism
- Signals for controlling the pneumatic slide is available through a pin header connector

Power supply

The main power supply for the RTX2300 Smart ATE unit is a 12V external switch-mode adaptor connected to the power-plug. This adaptor supplies all the internal circuitry including the installed modules. By adding an optional Programmable PSU module to the RTX2300 unit the DUT can also be powered through the RTX2300. A more detailed description of this optional module is given on page 82. An alternative way to supply the DUT with power is to connect an external power supply to the DUT Ext. Power connector on the rear panel of the RTX2300 (see figure below). The power supply to use for the DUT can be controlled from the SW API.

Power supply switching for DUT



Communication interfaces

The RTX2300 Smart ATE is equipped with one USB connector which serves as the only communication interface between the PC with the test application and the RTX2300 unit. Inside the RTX2300 the USB connection is split into 7 USB ports through in an internal full speed USB Hub. Furthermore, a serial communication port is available for DUT communication – i.e. the following communication interfaces are supported at the DUT:

- Four USB ports are available for the expansion-slots (i.e. one for each slot)
- One USB port is converted to a UART for communication channel to the RTX2300 main board CPU
- One USB port is used as communication channel to the CCB
- One USB port is used as communication channel to the DUT (i.e. from the PC with the test program directly to the DUT) through the SCB

Expansion slot USB interfaces

Each expansion slot (except the PSU module slot) is configured with a separate USB connection, and hence, these are used for communicating with the installed modules.

RTX2300 control interface

This interface is used for communication to control the functions in the RTX2300 unit.

CCB USB interface

This interface is a general-purpose USB connection and it is connected directly to the CCB interface.

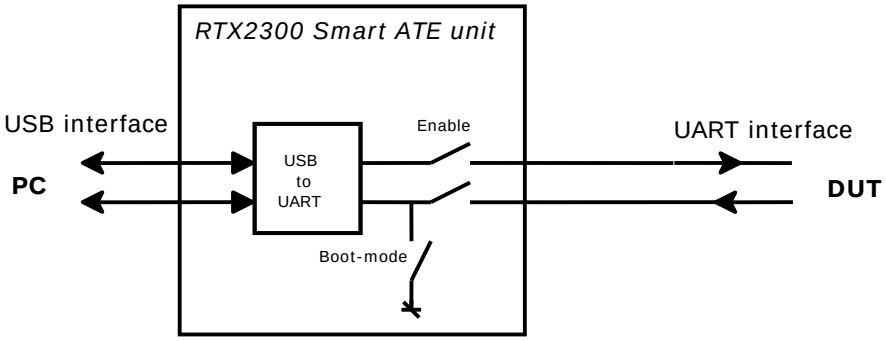
SCB USB interface

This interface is a general-purpose USB connection and it is connected directly to the SCB interface.

DUT serial communication configuration

To communicate with the DUT the RTX2300 has an UART for serial communication. Seen from the PC the UART is a USB com-port. The UART is implemented as outlined below and it has automatic level detection.

Simplified DUT UART Port



As indicated in the figure above the UART has tree modes (can be controlled from SW):

- **Disabled** - In Disabled mode the UART is disconnected from the DUT. This is the default mode after power up of the RTX2300.
- **Normal** - In Normal mode the UART is connected to the DUT and ready for communication.
- **Boot-mode** - In boot-mode the RX-line is connected to ground-level to bring the DUT in boot-mode (RTX equipment). When the DUT is in boot-mode the UART must be set to “Normal” before communication can be established.

E. Installing RTX2300 Options

As mentioned before, optional components for the RTX2300 are available – these can be categorized into four different categories:

- Modules
- Quick-Swap Kit
- Fixture Kits
- Connectivity options

Installation of the first three categories will be outlined in more detail in the following.

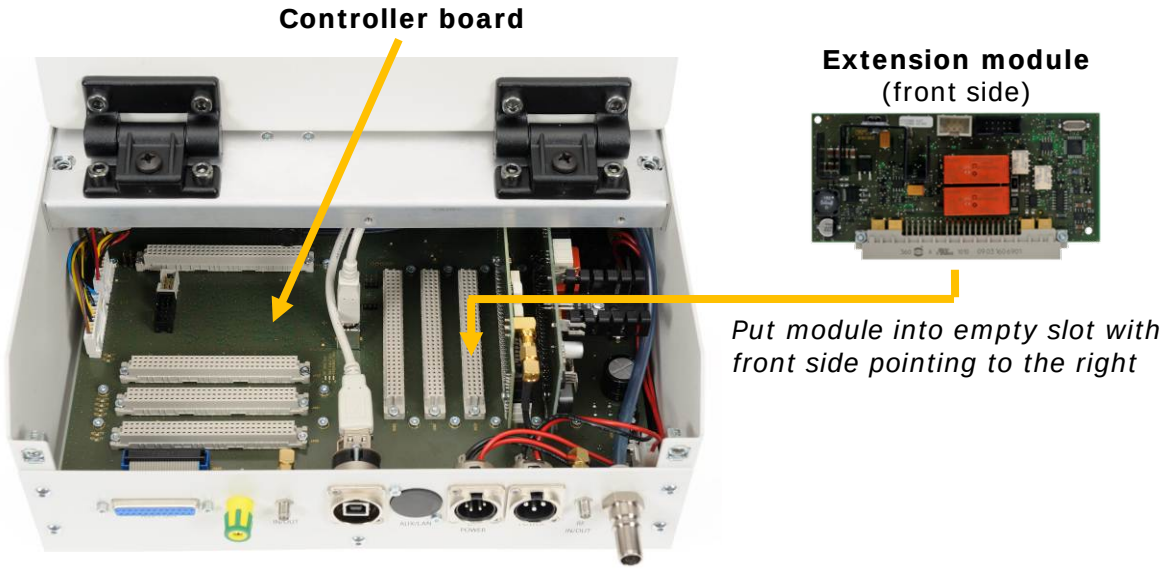
i. Installing modules

The modules are installed in the controller chamber, and hence, the lid to the controller chamber must be removed (see figure below). Please use a screwdriver to remove the four screws holding the lid.

Removing the four screws holding the controller chamber lid

After removing the lid, the controller chamber is now accessible and the extra module(s) can be installed in the empty extension slots (see figure below). Please be aware that the right most connector is different from the other four connectors. This connector is reserved for the programmable PSU module. Please make sure to install the module correctly in the connector – i.e. with the front side of the module (i.e. the top-side of the PCB with all main components mounted) pointing to the right. To ensure correct operation of the module please also make sure to press the module tightly into the connector on the controller board.

Controller chamber - installing a module in an empty extension module slot



After installing the module(s) please put the controller chamber lid back on again and tighten the four screws holding the lid. To use the modules from the test program a SW package installation might be necessary. For more details on this issue please refer to the specific module descriptions in chapter 4.

ii. Installing Quick-Swap Kit

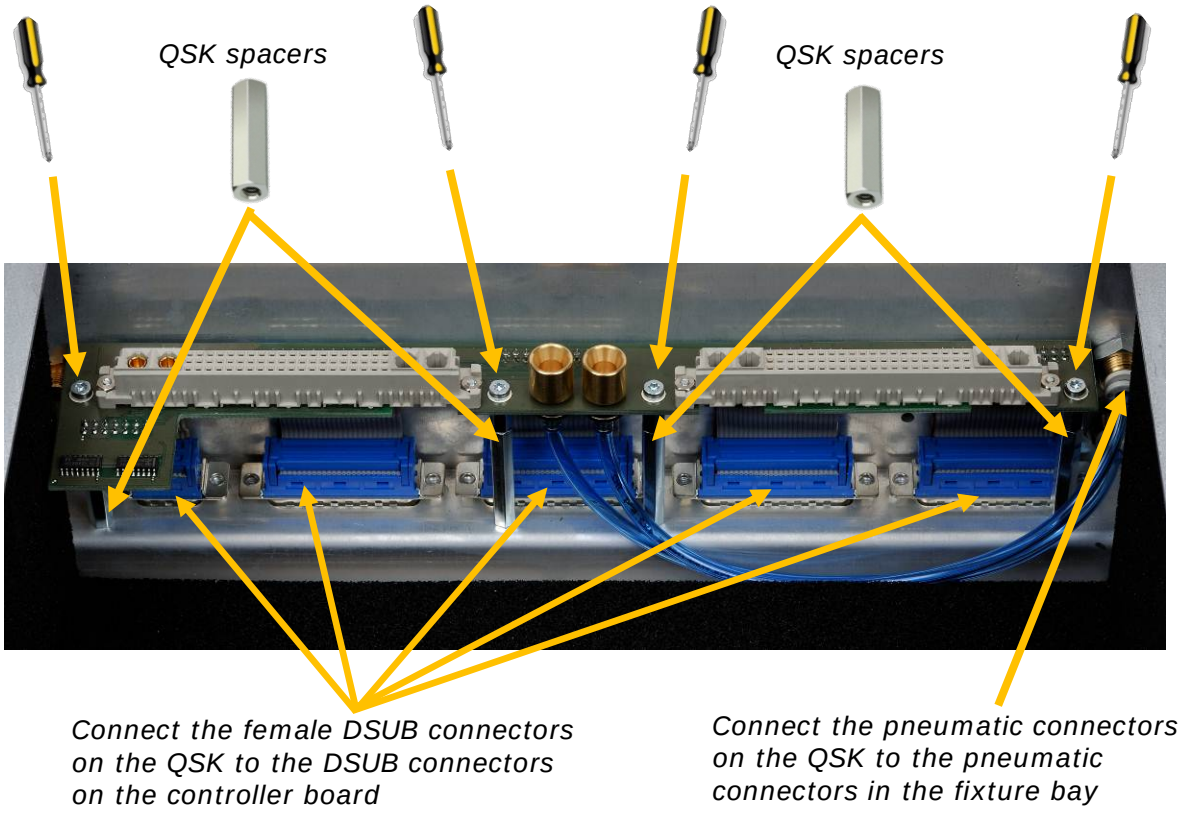
The Quick-Swap Kit (QSK) is divided into two parts – a fixture bay part and a fixture part. Installation of each of these parts will be outlined in the following.

Installing the fixture bay part of the Quick-Swap Kit

The first step is to remove the fixture part (if installed) to gain access to the fixture bay. Place the QSK in the fixture bay close to the standard DSUB (male) connectors in the fixture bay and connect the DSUB connectors on the QSK to the standard connectors in the fixture bay. Please ensure that the connectors are connected correctly.

Next step is to connect the two pneumatic connectors on the QSK to the connectors in the fixture bay. When all connectors have been connected please mount the four spacers (included in the QSK fixture bay part package) in the fixture bay and place the QSK on top of the spacers. Please ensure that the spacers are tightened. Mount and tighten the four screws (also included in the QSK fixture bay part package) to fasten the QSK on top of the spacers.

Installing the Quick-Swap Kit (fixture bay part)



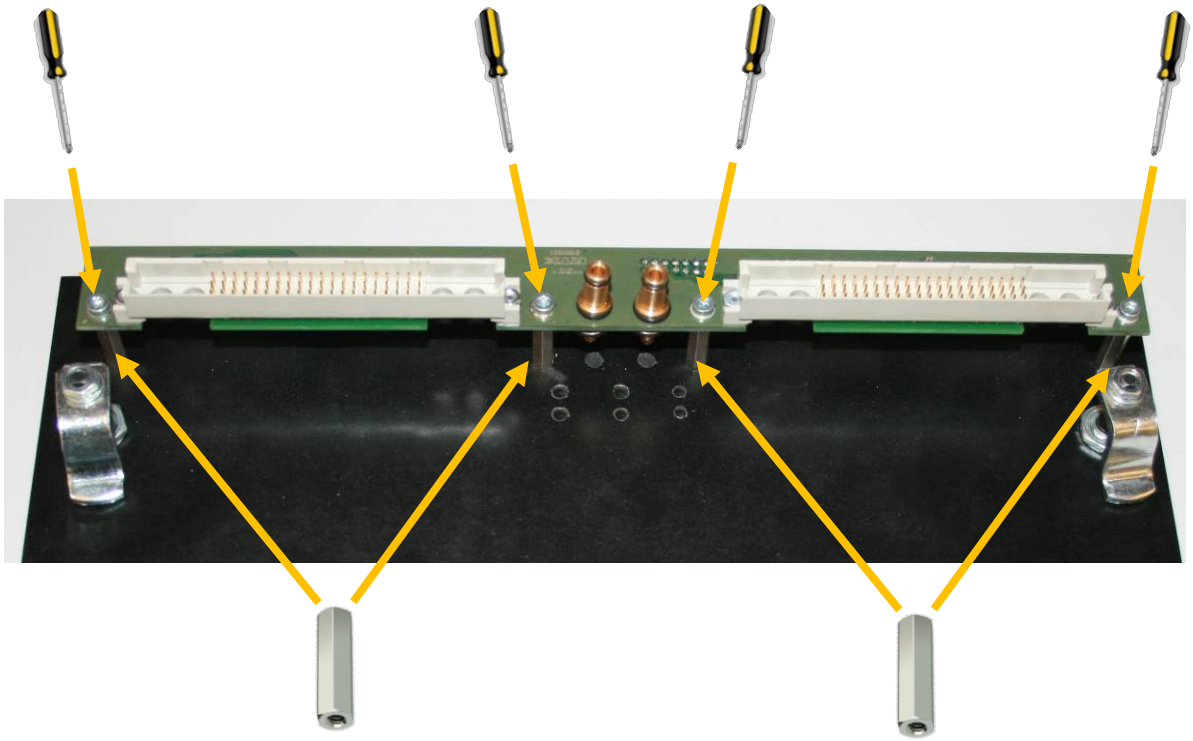
Installing the fixture part of the Quick-Swap Kit

To enable easy installation of the fixture part QSK a new fixture plate should be used (i.e. without any connectors or other equipment installed). However, the installation steps outlined below should apply to all fixtures, but it could be necessary to remove some of the items installed on the fixture.

The first step is to mount the QSK spacers (included in the QSK fixture part package) on the fixture. Place the QSK on top of the spacers. Mount and tighten the four screws as shown in the figure below.

Please note that only the SCB connector, CCB connector and one of the connectors on the top side of the QSK are mounted, and hence, connecting the QSK interface to the fixture pins must also be done. This is, however, outside the scope of this section but for more information on customizing the fixture please refer to chapter 6 (especially section D on page 115).

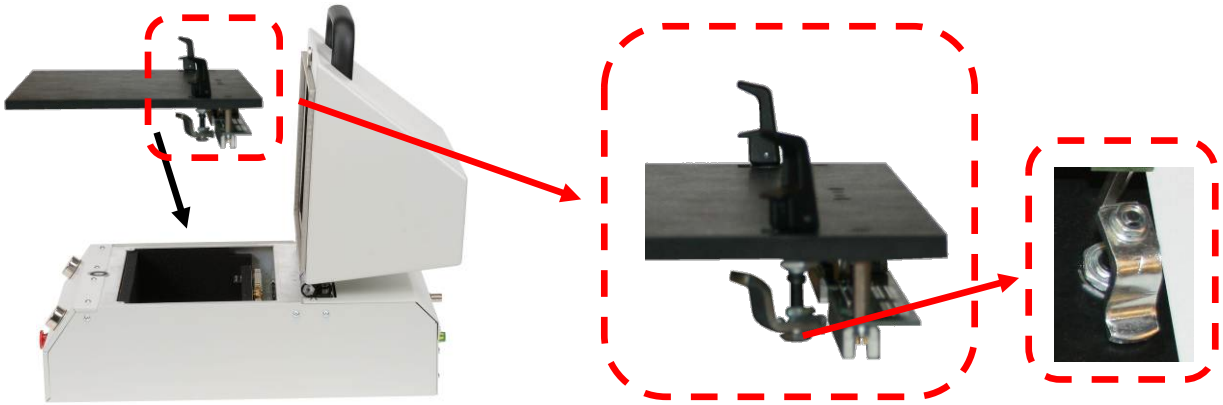
Installing the Quick-Swap Kit (fixture part)



iii. Installing a fixture kit

The first step is to remove the fixture part (if installed), hence unlocking the two locks on each side of the fixture and removing it from the RTX2300 unit. The next step is to prepare the new fixture and mount it in the fixture bay.

Installing a fixture kit in the RTX2300 Smart ATE unit



The installation process defers slightly depending on whether or not a QSK has been installed:

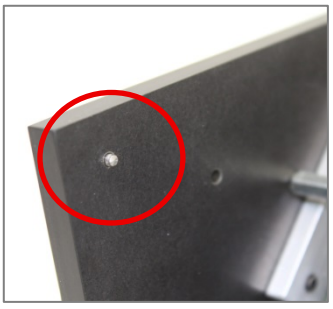
With QSK installed:

If a QSK has been installed the installation process is quite straightforward. Just make sure the (pneumatic, CCB and SCB) connectors on the fixture QSK part are connected properly to the fixture bay part QSK connectors and secure the fixture by locking the two fixture locks.

It is recommended that following steps are followed to mount the fixture insert into the fixture bay of the RTX2300 unit.

1) The fixture must have a guide-pin (see picture 1) mounted at each corner opposite to the quick-swap-connectors of the fixture insert.

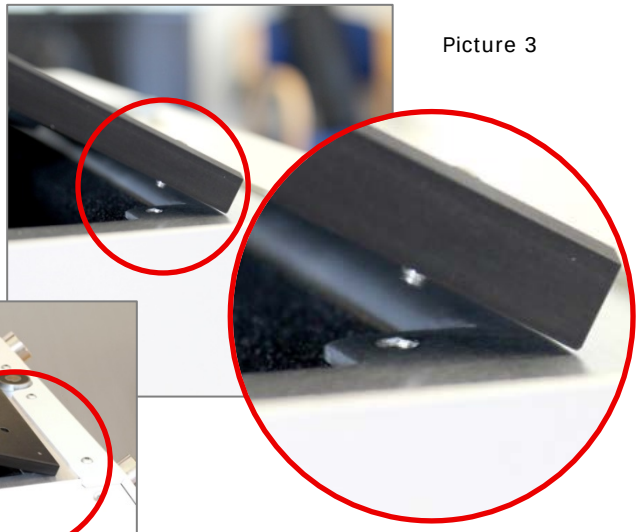
Picture 1



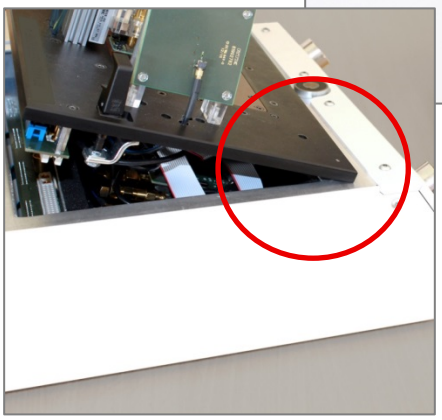
2) In the fixture bay are there two guide-holes (see picture pins in which the guide-pins must fit.

Picture 2

3) The mounting of the fixture insert in the fixture bay must begin with an alignment of the guide-pins with the guide-holes (see picture 3).
When the guide-pins and holes are aligned the connectors can be pressed in place (see picture 4)



Picture 3



Picture 4

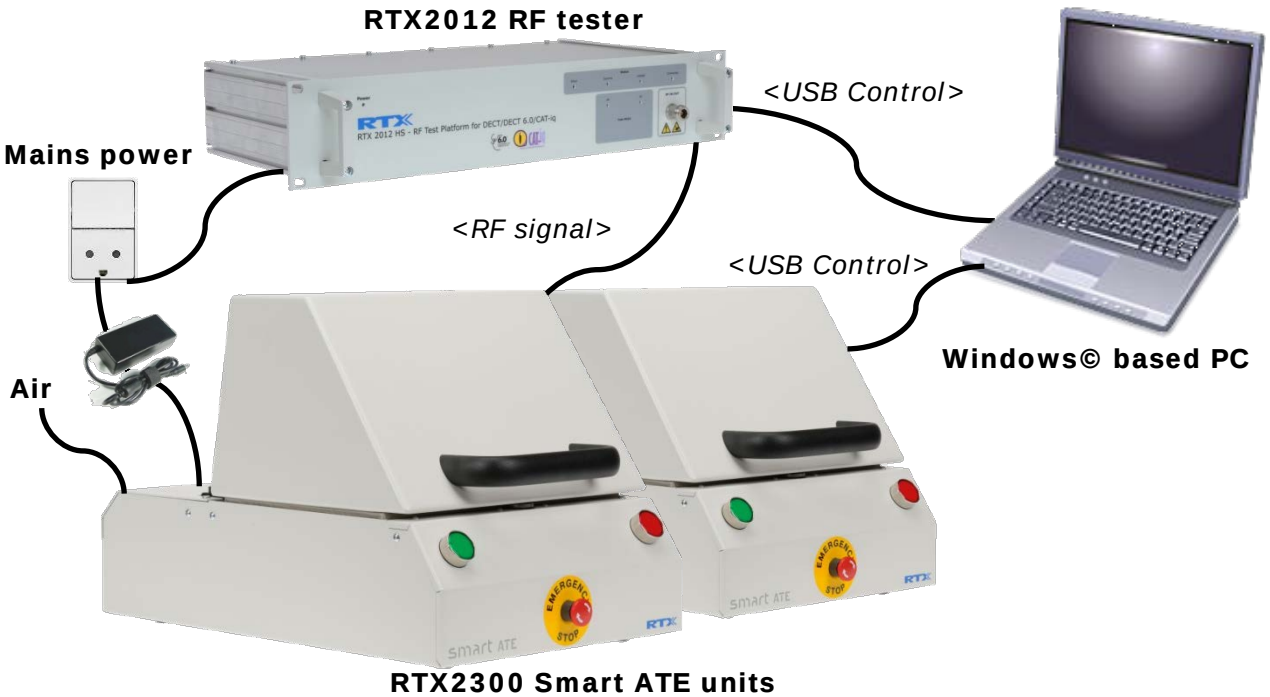
Without QSK installed:

To connect a fixture without a QSK installed the connectors on the fixture must be connected directly to the standard interface in the fixture bay. If the fixture includes a pneumatic slide the pneumatic tubes must be connected to the pneumatic connectors on the back panel of the fixture bay (see the figure on page 14 for an overview of the standard fixture bay connectors). The next step is to connect the fixture SCB and CCB connectors to the connectors in the bottom of the fixture bay. Please make sure to connect the fixture correctly and secure the fixture by locking the two fixture locks.

F. System setup and installation of the RTX2300

The RTX2300 is the key component in an automatic test system setup and it deliverers most of the instrumentation (PSU, DVM, relays etc) needed to configure a production test setup for a large variety of applications. The only external parts required are a Windows based PC and an RF communication tester (if RF testing is required). Hence, a typical setup could look like the one outlined in the figure below. It consists of a Windows based PC, an RTX2012 RF tester (in case of a DECT product) and two RTX2300 Smart ATE units. Please note that one PC can control more than one RTX2300 unit (as indicated in the figure).

Simple automatic production test setup with RTX2300 and RTX2012



i. Connecting the RTX2300 Smart ATE

To configure a simple production test system (as the one outlined above) please follow the four steps below to setup the above RTX2300 Smart ATE system.

1. **Connect the RTX2300 units to the PC –** the first step is to connect the RTX2300 units to a PC using the provided USB cables. In the example above both units are connected to the same PC.

2. **Connect external equipment to the RTX2300 units –** if external equipment is required (for example an RF tester) this must be connected to the relevant connectors on the rear panel. In the setup above an RTX2012 RF tester has been connected to the two RTX2300 units (through the RF port on the rear panel).

3. **Connect air for the pneumatic controls in the RTX2300 units –** to enable the pneumatic functionality inside the RTX2300 air must be connected to the air connector on the rear panel of the RTX2300 units.

4. **Connect power to the RTX2300 units –** the last step is to connect the RTX2300 to the mains power using the supplied adaptors. Before switching on power to the RTX2300 units please ensure that the emergency button is released on both units. If this is the first time the RTX2300 units have been connected to the PC installation of the RTX2300 driver package will be initiated (i.e. the PC will ask for driver setup when switching on power to the equipment). Please follow the instructions in section G on page 31 (below) for installing the necessary drivers.

In addition to the above steps please also make sure to connect the RTX2300 to a proper ground level to avoid problems with ESD.

G. Installing the PC Software

In this section the installation process of the supplied SW is outlined. However, before installing the SW (which can be downloaded from the RTX Tester Download Center at www.rtx.dk/testers/downloadcenter) please confirm that the PC on which to install the SW adheres to the minimum requirements outlined below. If it does not, then successful operation of the RTX2300 Windows applications cannot be guaranteed.

System Part	Minimum requirement
CPU	1 GHz processor (Intel© or AMD) or faster
RAM	1 GB (2 GB for 64-bit operation)
Available disc space	75 MB
Monitor resolution	1280 x 1024 pixels or higher
CD-ROM drive	YES
Ports	Available USB port
Operating System	Windows XP/Windows 7 (32- or 64-bit)

The RTX2300 SW package includes the necessary drivers, supporting Windows applications, RTX2300 DLL, source code for demo applications, and interface documentation. The source code for demo applications and the interface documentation will be described in more detail in chapter 7. The following drivers and supporting Windows applications are included:

- RTX2300 DUT UART Driver
- RTX2300 Basic Unit Communication Interface Driver
- RTX EAI Port Server
- RTX2300 Detective debug application
- RTX2300 DLL
- RTX2300 USB Bridge Driver
- RTX2300 .net

In the following instructions for installing the above items are outlined. Please note that the instructions provided here outlines installation of SW for the basic RTX2300 unit. Consequently, if extension modules have been installed in the RTX2300 please refer to section A in chapter 4 for information on how to install specific module related SW packages (if relevant). However, one exception is the driver for the Programmable PSU; this driver is included in the RTX2300 Basic Unit Communication Interface, since it is tightly coupled to the operation and interfaces of the RTX2300 main board. Furthermore, the RTX2300 Detective application also provides specific PSU functions.

The typical installation procedure is outlined below and will be described in more detail in the next sections:

1. **Install the RTX2300 Basic Unit SW package:** The first step in a typical RTX2300 installation process is to install the RTX2300 Basic Unit SW package. This package will also install the RTX2300 DLL and the RTX2300 Detective debug application along with extract driver files and documentation. Hence, this should be done before switching on the RTX2300 unit for the first time. Please note that both the RTX2300 USB Bridge Driver and the RTX EAI Port Server can be installed as part of the installation process.
2. **Install the RTX2300 USB Bridge Driver:** Install the RTX2300 USB Bridge Driver to be able to communicate with the RTX2300 unit over the USB interface. The driver installation can be initiated as part of the RTX2300 Basic Unit SW package or started manually.

3. **Install the RTX EAI Port Server:** Install the RTX EAI Port Server to be able to map the COM-ports made available by the RTX2300 DUT UART Driver and the RTX2300 Basic Unit Communication Interface Driver. The driver installation can be initiated as part of the RTX2300 Basic Unit SW package or started manually.

4. **Install the RTX2300 DUT UART Driver:** Install the RTX2300 DUT UART Driver to enable communication directly with the DUT. Normally this driver is installed upon switching on the RTX2300 unit for the first time.

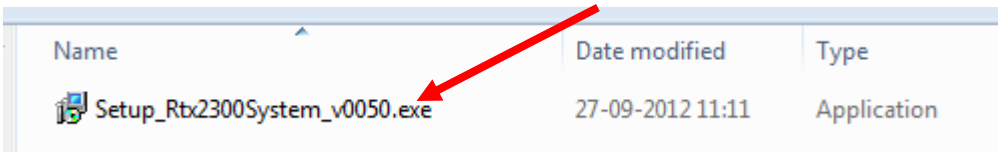
5. **Install the RTX2300 Basic Unit Communication Interface Driver:** Install the RTX2300 Basic Unit Communication Driver to enable communication with the RTX2300 unit (i.e. this is the driver for controlling the RTX2300 unit). Normally this driver is installed upon switching on the RTX2300 unit for the first time.

i. Installing the RTX2300 Basic Unit SW

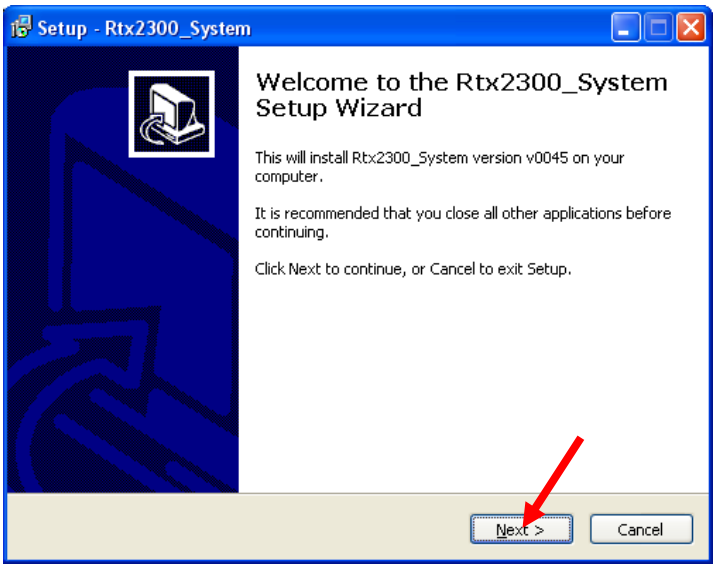
Turn on the PC and download the RTX2300 SW package from the RTX Download Center (http://www.rtx.dk/testers/downloadcenter) and save the file on the PC. Please note that the outlined screenshots and the procedure described in this paragraph are based on a PC using Windows XP. Hence, if you use another Windows OS on your PC the setup procedure can deviate a bit from the one shown here. As noted above the RTX2300 DLL and RTX2300 Detective debug application will be installed as part of the RTX2300 Basic Unit SW package installation process.

❶ Select and unzip the downloaded RTX2300 SW package.

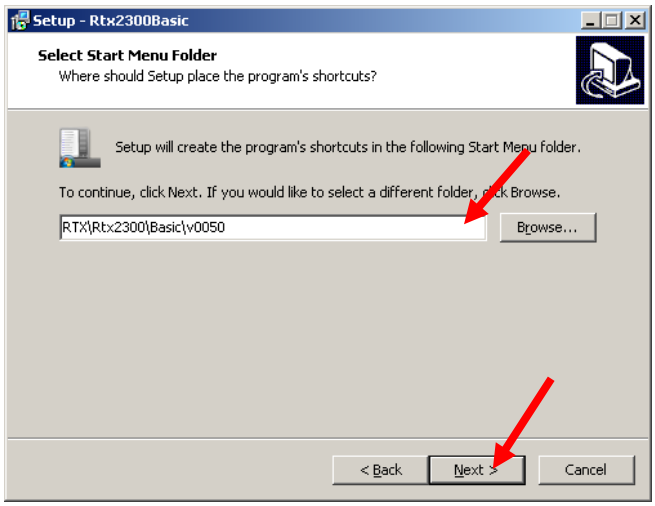
❷ Click on the Setup_Rtx2300System.exe file.



❸ Click **Next** to continue the installation process

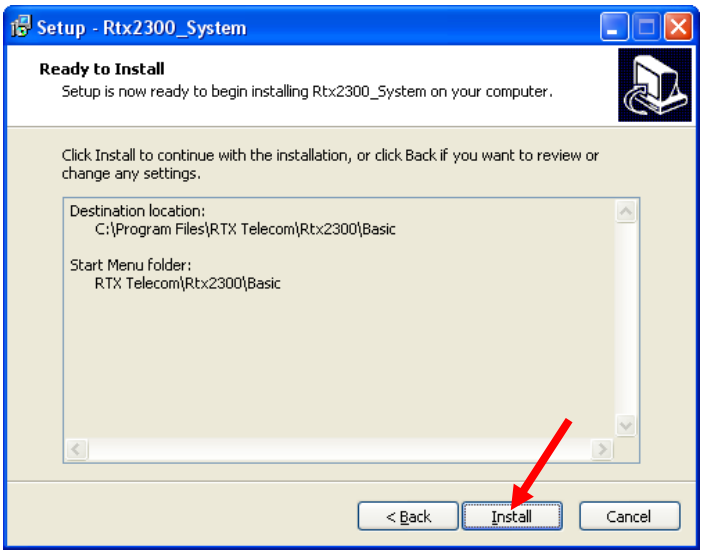


❹ Enter the destination location in which to install the RTX2300 Basic Unit SW package



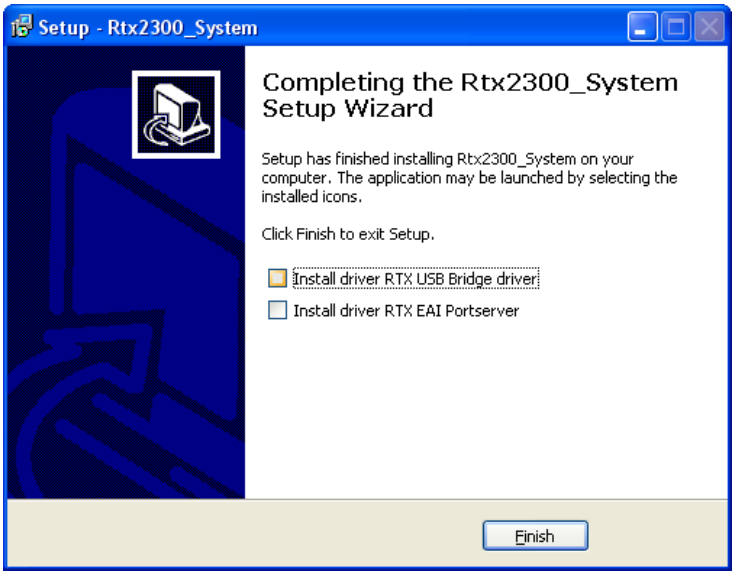
As indicated above the RTX2300 Basic Unit SW package will be installed into the C:\Program files\RTX\Rtx2300\Version\ directory as default. If the SW should be installed in another location, please enter a new location or browse for a suitable location. Click **Next** to continue the installation process. The next step is to select the location of the programs in the start menu folder (through a similar dialogue box as the one above). The default location is “All Programs/RTX/Rtx2300/Basic”. Click **Next** to continue the installation process.

❺ Validate the installation information selected in the prior steps and click **Install**



After clicking **Install** the installation process will continue and the RTX2300 Basic Unit SW package will be installed into the locations chosen as part of the steps performed above. Upon completion of the installation of the RTX2300 Basic Unit SW package the **Setup Complete** dialogue box offers the possibility to install the RTX EAI Port Server and/or the RTX USB Bridge Driver.

❻ Install RTX USB Bridge Driver and/or the RTX EAI Port Server if necessary

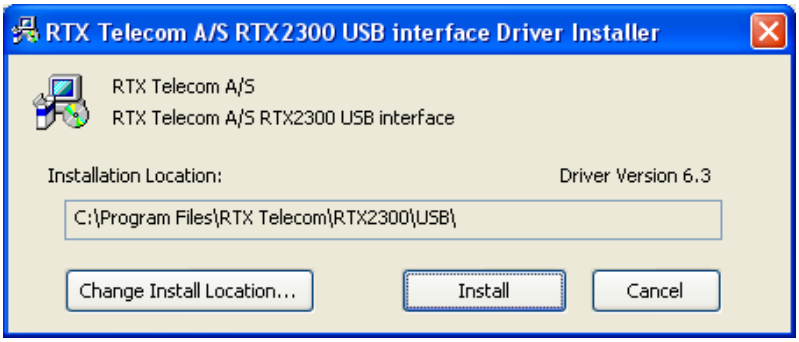


Click the check box to select the relevant items to install and click **Finish**. Please refer to paragraph ii and 0 in this chapter in relation to the installation process for the RTX2300 USB Bridge driver and the RTX EAI Port Server, respectively. If these have already been installed just click **Finish** to complete the installation process.

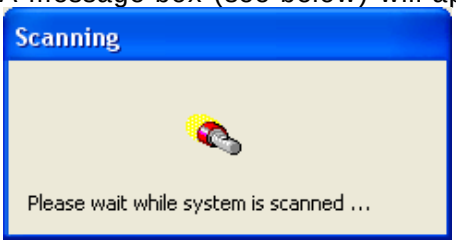
ii. Installing the RTX2300 USB Bridge Driver

As noted above the installation process of the RTX2300 USB Interface Driver can be initiated as part of the installation process of the RTX2300 Basic Unit SW package or manually. To start the process manually please open a windows explorer and go to the location in which the RTX2300 Basic Unit SW has been installed (e.g. the default location `C:\Program Files\RTX\Rtx2300\Basic`). Open the `Driver` directory and double click on `RTX2300VCPInstaller.exe` to start the installation process. In both cases the installer window (see below) will appear.

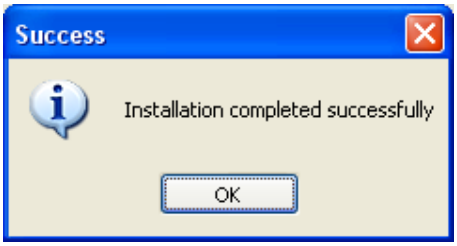
❶ Select installation location and click **Install** to start installation of the driver



A message box (see below) will appear.



The installation process is quite straightforward and after a moment a status message will appear (see below). If the installation fails, please try to install the driver again.



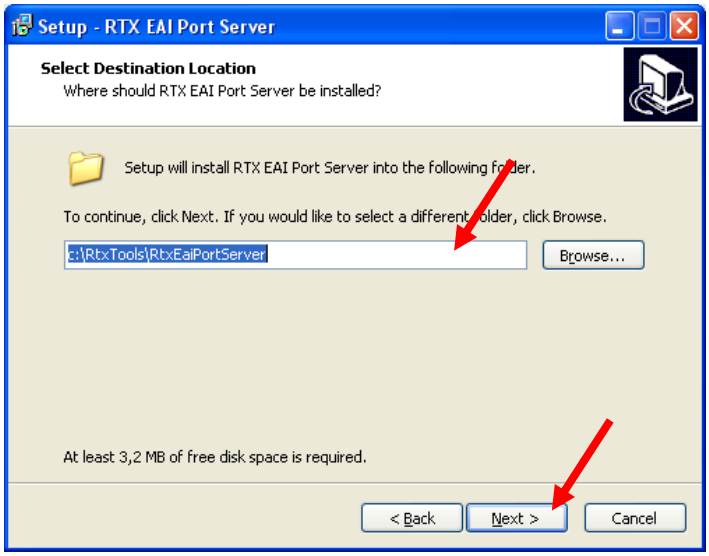
Click OK to complete the installation of the RTX2300 USB Interface Driver.

iii. Installing the RTX EAI Port Server

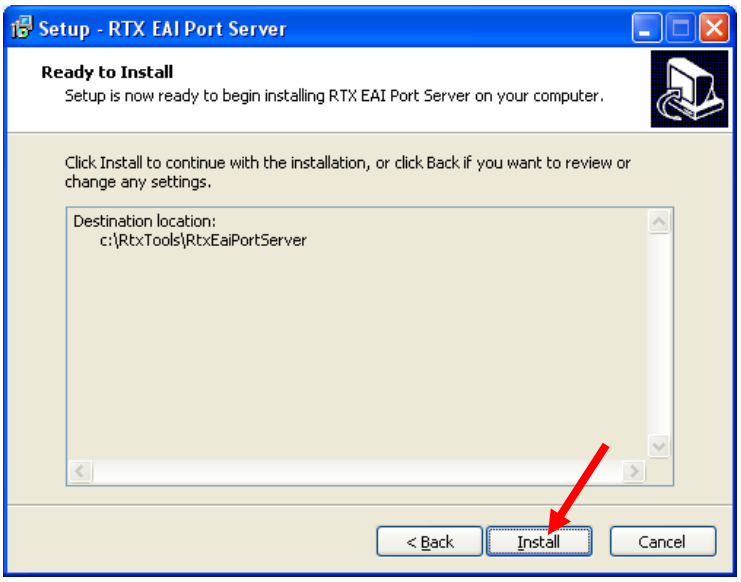
The RTX EAI Port Server is needed to connect the RTX2300 Detective debug application to the COM port on which the RTX2300 Basic Communication Interface has been installed. The installation process of the RTX EAI Port Server can be started either as a step in the RTX2300 Basic Unit SW installation process or directly by double clicking on `RtxEaiPortServer-setup.exe` (located in the `\basic\driver` directory) in a Windows Explorer (or alternatively running it directly from the command line).

❶ Click **Next** to continue

❷ Enter the destination location and click **Next**



❸ Validate the installation information selected in the prior steps and click **Install**



The installation of RTX EAI Port Server will now start. Upon completion of the installation process the 'Setup Complete' dialogue box will appear. Click Finish to close the setup wizard. Please note the check-box 'Launch RTX EAI Port Server' – if selected the RTX EAI Port Server is started and it will appear as an icon in the notification area of the Windows toolbar.

❹ Click **Finish** to close the setup wizard – please note the check-box

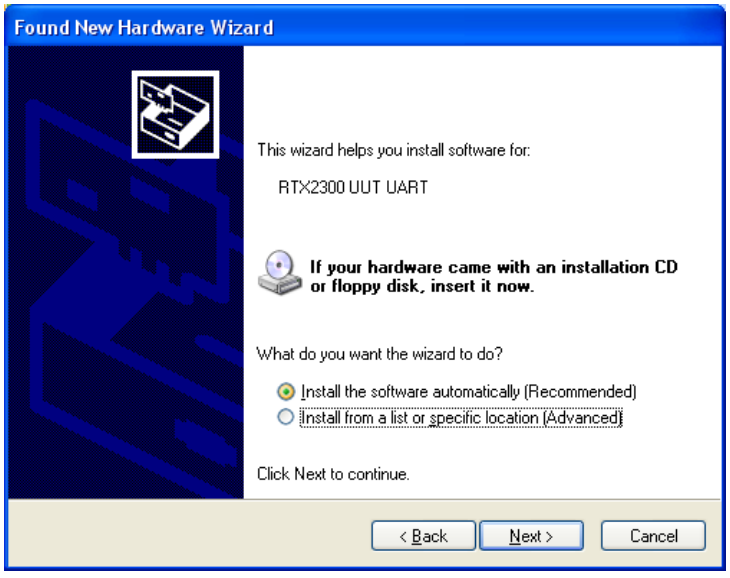
iv. Installing the RTX2300 DUT UART driver

To communicate with the DUT from a PC through the RTX2300 unit the DUT UART Driver must be installed. *Please note that the driver has not yet been digitally signed by Microsoft.* To install this driver please turn on your PC and do the following:

1) Connect one end of the supplied USB cable into the USB port of the RTX2300 unit (on rear panel) and connect the other end to a USB port on your Windows PC (if not already done).

2) Connect the RTX2300 unit to a power source and turn on the RTX2300 unit. The **Found New Hardware Wizard** window should now appear (see below). Please note that the **Found New Hardware Wizard** window for the RTX2300 Basic Communication Interface Driver could appear as well. Select **No, not this time** and click **Next**.

3) Select **Install the software automatically (Recommended)**.



4) Upon completion of the RTX2300 DUT UART driver installation the **Completing the Found Hardware Wizard** is displayed (see below).

5) Click **Finish** to close the wizard.

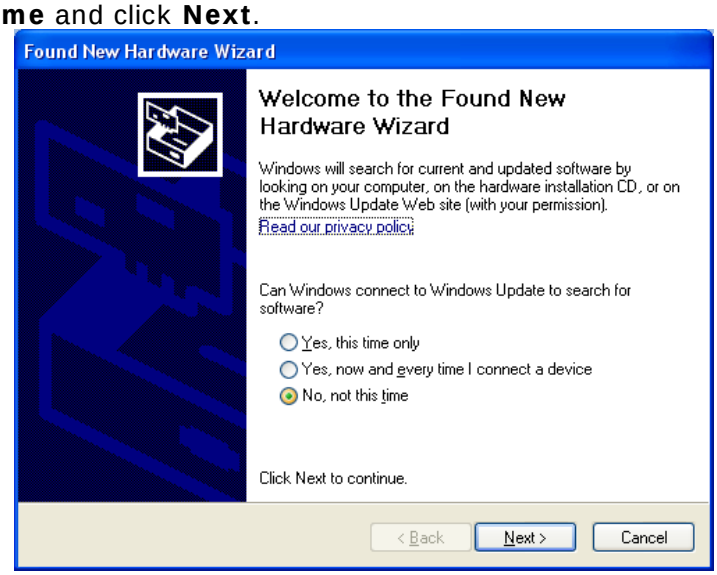
The RTX2300 DUT UART driver should now be successfully installed and a new COM port for DUT access should now be available.

v. Installing the RTX2300 Basic Unit Communication driver

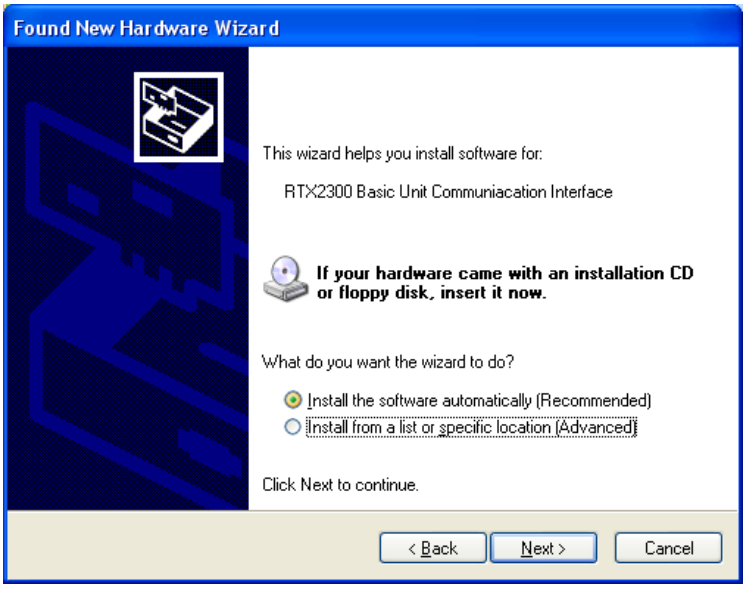
To communicate with the RTX2300 unit from a PC the RTX2300 Basic Unit Communication Interface driver must be installed. *Please note that the driver has not yet been digitally signed by Microsoft.* The install process is the same as for the RTX2300 DUT UART driver and it will in fact be initiated at the same time (i.e. when the RTX2300 is turned on the first time). Hence, to install the RTX2300 Basic Unit Communication Interface Driver please perform the following steps:

1) Connect one end of the supplied USB cable into the USB port of the RTX2300 unit (on rear panel) and connect the other end to a USB port on your Windows PC (if not already done).

2) Connect the RTX2300 unit to a power source and turn on the RTX2300 unit. The **Found New Hardware Wizard** window will now appear (see below). Select **No, not this time** and click **Next**.



3) Select **Install the software automatically (Recommended)**.



4) Upon completion of the RTX2300 Basic Unit Communication Interface driver installation the **Completing the Found Hardware Wizard** is displayed (see below).

5) Click **Finish** to close the wizard.

The RTX2300 Basic Unit Communication Interface driver should now be successfully installed and a new COM port for RTX2300 unit access should now be available.

2. RTX2300 Basic Unit SW Package

A. Introduction

In this chapter an overview of the contents of the Windows package is provided along with a more detailed description of the two central RTX2300 applications (i.e. RTX2300 Detective and RTX EAI Port Server).

B. Contents of the RTX2300 Basic Unit SW Package

The RTX2300 Basic Unit SW package includes all necessary drivers and applications for installing the RTX2300 unit on a Windows PC. In addition to the drivers and applications some documents, source code (i.e. program examples, interface header files) and binaries (`Rtx2300PcIntf.dll`, `Rtx2300PcIntf.lib`) are also included. After installing the RTX2300 Basic Unit SW package the RTX2300 files are placed in the following directory structure (relative to the install directory)

`Basic`
Includes the RTX2300 Detective application, interface documentation, source code (program examples and interface header files), and binaries (i.e. the Rtx2300PcIntf.dll and Rtx2300PcIntf.lib files).

- **`Drivers`**
  Includes installers for RTX EAI Port Server and the RTX2300 USB Bridge driver.
- **`USB`**
  Includes USB driver for the RTX2300 DUT UART and the RTX2300 Basic Unit Communication Interface (both drivers for x86 and x64 are included)

In the following paragraphs the central windows applications, documentation, included source code and binaries will be briefly outlined.

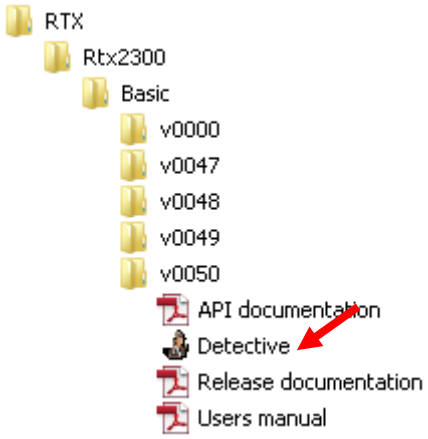
i. Windows applications

The central applications in relation to the basic RTX2300 unit are the RTX EAI Port Server and the RTX2300 Detective debug application. These applications will be outlined in more detail in section C and D in this chapter, respectively. After successful installation the two applications can be started by selecting (also outlined in the figure below):

`All Programs->RTX->RTX EAI Port Server` (default location)

AND

`All Programs->RTX->Rtx2300->Basic->Vxxx->Detective` (default location)



ii. Documentation

The included documentation includes the following documents:

- Firmware Update Procedure
- Release Notes
- RTX2300 Common Interface (types)
- RTX2300 Interface (API)

Please also refer to section A on page 118 for more information regarding the interface documentation.

iii. Source Code and Binaries

The included source code is:

- Interface header files
  - `DllInterface.h`
  - `IRtx2300.h`
  - `IRtx2300IntfCommon.h`
  - `Rtx2300BasicTypes.h`
  - `Rtx2300PcIntf.h`
  - `Rtx2300Primitives.h`
  - `Rtx2300SysTypes.h`
- Binaries
  - `Rtx2300PcIntf.dll`
  - `Rtx2300PcIntf.lib`
- Example Source code
  - Firmware Update (`FirmwareUpdate.zip`)
  - Simple test application (`Simple.zip`)
  - Test application with multiple RTX2300 units (`MultiInst.zip`)

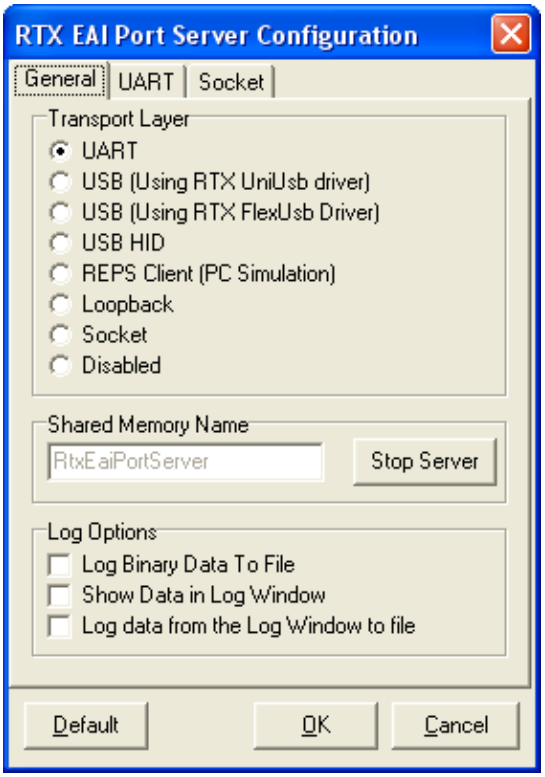
Please also refer to chapter 7 for more information regarding usage of the binaries and source code.

C. RTX EAI Port Server

The RTX EAI Port Server is a central element in the communication channel between the RTX2300 unit and the RTX2300 Detective debug SW. Since only parts of the functionality provided by the RTX EAI Port Server is required in the RTX2300 system, the application will only be described briefly in the following. The most important parts are the **Transport Layer** selection on the **General** tab and the settings (specifically **Port**, **Stop bits**, **Parity** and **RTS Input Flow Control**) on the **UART** tab.

i. Setup - General Page Overview

The user interface always opens with the **General** page displayed (see below) and the main functionality that can be set through this page is related to the transport interface and logging. it is only necessary to check that the transport layer is set to UART (default value), and hence, the rest of the functionality is not needed for normal operation of the RTX2300.

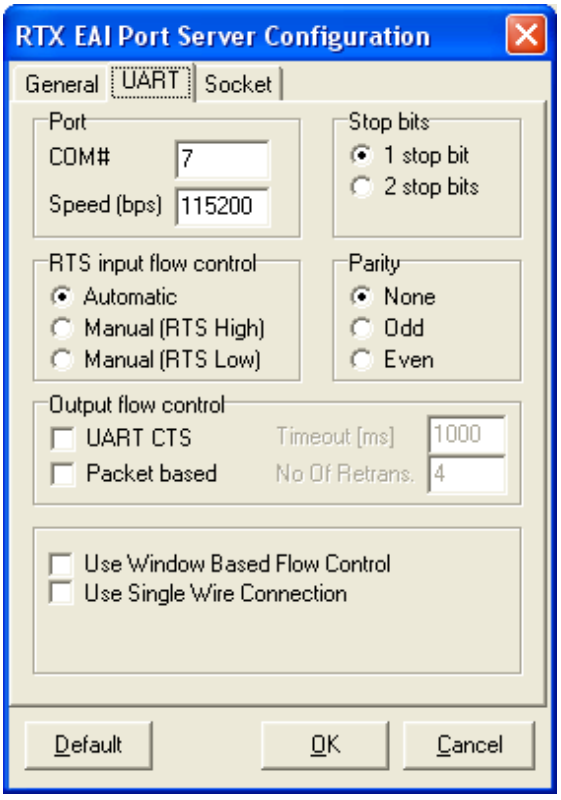


The **General** page contains the following panels:

Panel	Parameters	Short description
Transport Layer	UART / USB (IniUsb) / USB (FlexUsb) / USB HID / REPS Client / Loopback / Socket / Disabled	Set the transport layer type
Shared Memory Name	-	Name of the shared memory area used by the server
Log Options	Log Binary Data to File, Show Data in Log Window, Log data from Log Window to file	Settings for logging of data

ii. Setup - UART Page Overview

The **UART** page is mainly for configuring the UART parameters. In relation to normal RTX2300 operation the port settings are the most important parameters along with the other serial settings like stop bits, parity and flow control.

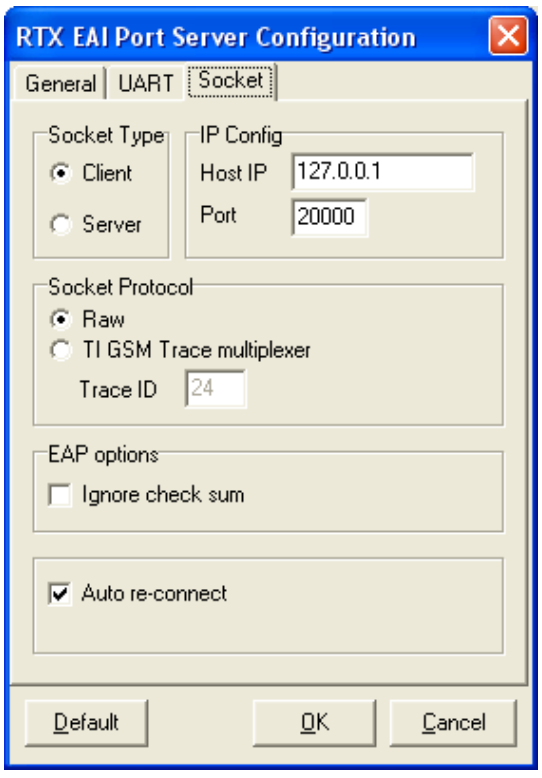


The **UART** page contains the following panels:

Panel	Parameters	Short description
Port	<port number>	Port number of the COM port to connect to
Stop Bits	1 or 2 stop bits	Indicates how many stop bits should be used
RTS input flow control	Automatic, RTS High, RTS Low	Setup of RTS input flow-control
Parity	None, Even or Odd	Indicates whether or not the parity bit is used – if used the parity is set to either odd or even
Output flow control	UART CTS, Packet based	Enabling and configuring output flow-control
Use Windows Based Flow Control	Enable/disable	-
Use Single Wire Connection	Enable/disable	-

iii. Setup – Socket Page Overview

The **Socket** page is mainly for configuring the socket interface. However, this functionality is not used in normal operation of the RTX2300 and will not be described in more detail here.

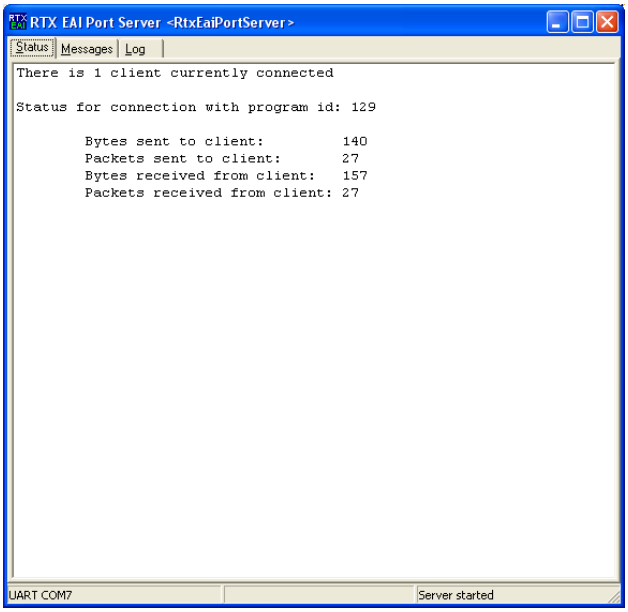


The **Socket** page contains the following panels:

Panel	Parameters	Short description
Socket Type	Client / Server	Configuring the type of socket connection
IP Config	<Host IP address>, <Port number>	Configuring host IP address and port number
Socket Protocol	Raw, GI GSM Trace multiplexer	Selecting the protocol type for the connection
EAP Options	-	Enable / disable ignore of the check sum
Auto re-connect	-	Enable / disable automatic re-connect upon loss of connection

iv. Status Window Overview

A status window for the RTX EAI Port Server also exists (see figure below). To open the status window please right click on the REPS icon in the system tray and select **Status**. In the status window central items of information regarding clients and communication can be seen. It is an essential source for information if communication problems are encountered.



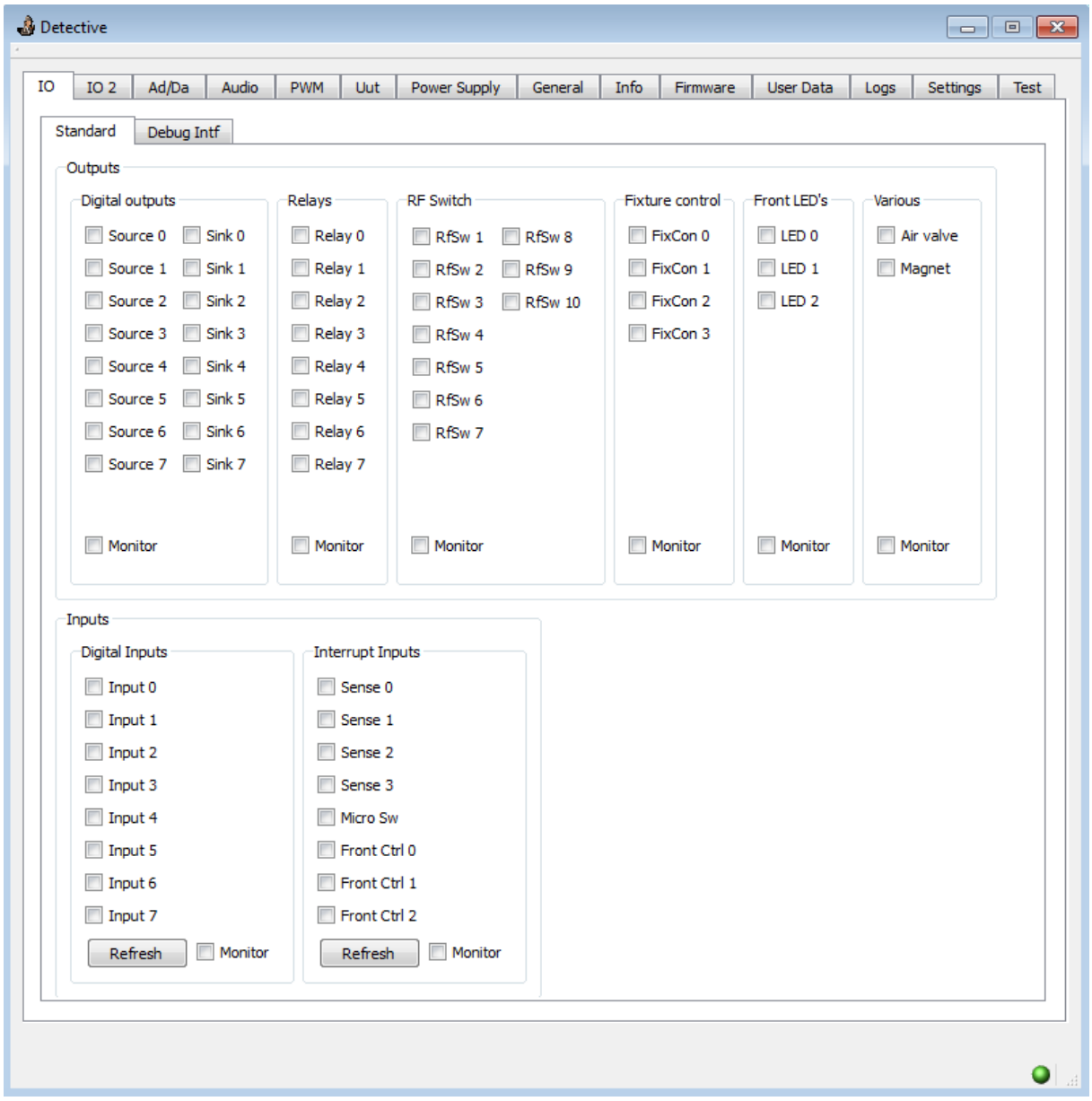
D. RTX2300 Detective debug application

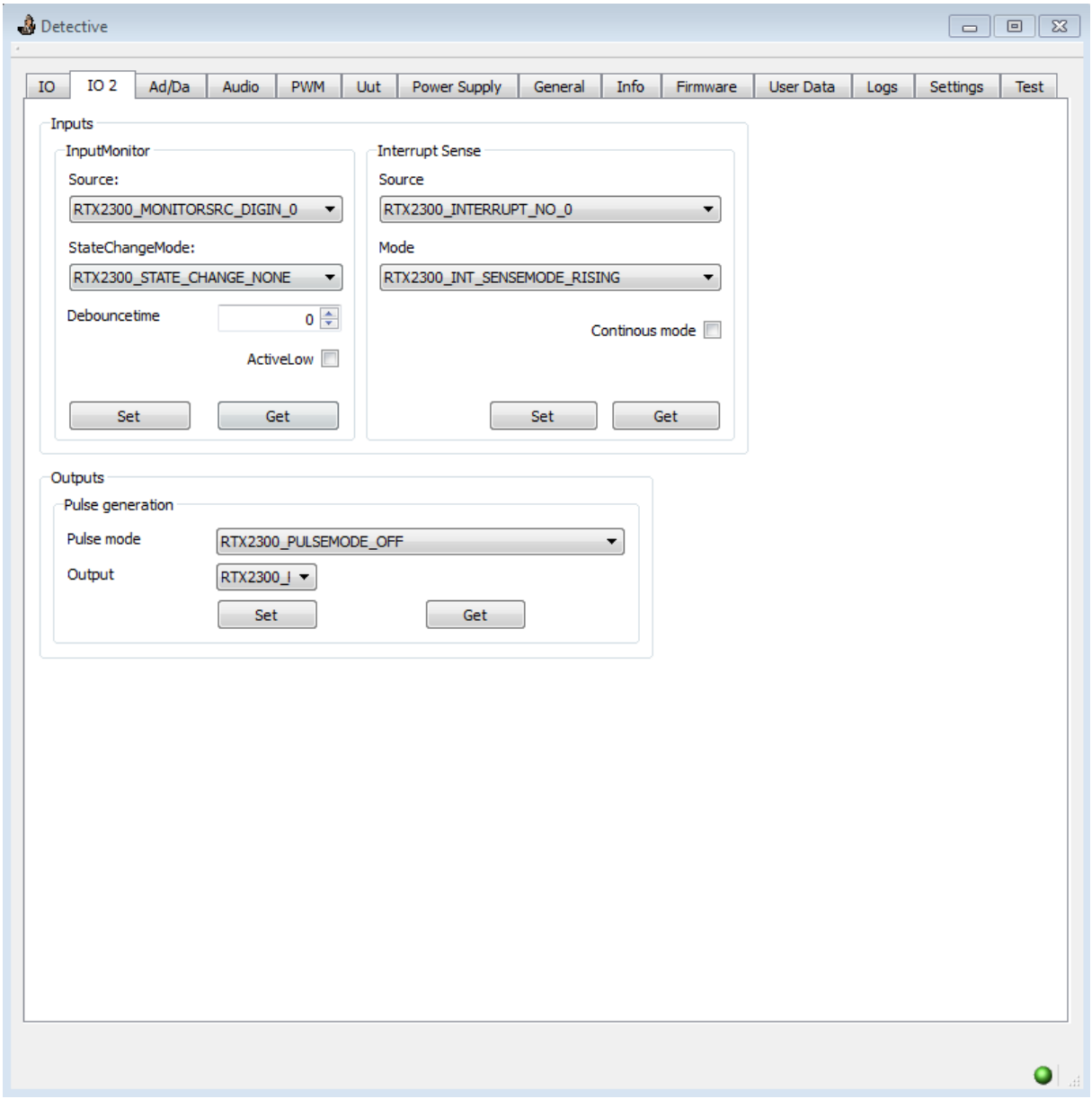
The RTX2300 Detective debug application is a valuable tool when debugging customization boards. it is a tool for monitoring and controlling the inputs and outputs of the RTX2300 directly. In the following each of the pages in the RTX2300 Detective application will be outlined in more detail.

i. I/O Page Overview

The **I/O** page is the start page of the RTX2300 Detective application and it is related to the I/O functionality of the RTX2300. From this page the complete I/O functionality can be controlled and monitored. The **I/O** page contains the following panels:

Panel	Parameters	Short description
Relays	-	Select / de-select or monitor relays
RF Switch	-	Select / de-select or monitor RF Switch outputs
Fixture control	-	Select / de-select or monitor fixture control outputs
Front LED's	-	Select / de-select or monitor front LED's
Various	-	Select / de-select or monitor the air valve or magnet outputs
Pulse generation	<Pulse mode>, <Output>	Set pulse mode and output
Interrupt inputs	-	Select / de-select or monitor interrupt inputs
InputMonitor	<Source>, <StateChangeMode>, <Debouncetime>, <ActiveLow>	Configure input monitor – set input source, set state change mode, set de-bounce time (in ms) and enable / disable active low


Detective
IO
IO 2
Ad/Da
Audio
PWM
Uut
Power Supply
General
Info
Firmware
User Data
Logs
Settings
Test
Standard
Debug Intf
Outputs
Digital outputs
Source 0
Source 1
Source 2
Source 3
Source 4
Source 5
Source 6
Source 7
Sink 0
Sink 1
Sink 2
Sink 3
Sink 4
Sink 5
Sink 6
Sink 7
Monitor
Relays
Relay 0
Relay 1
Relay 2
Relay 3
Relay 4
Relay 5
Relay 6
Relay 7
Monitor
RF Switch
RfSw 1
RfSw 2
RfSw 3
RfSw 4
RfSw 5
RfSw 6
RfSw 7
RfSw 8
RfSw 9
RfSw 10
Monitor
Fixture control
FixCon 0
FixCon 1
FixCon 2
FixCon 3
Monitor
Front LED's
LED 0
LED 1
LED 2
Monitor
Various
Air valve
Magnet
Monitor
Inputs
Digital Inputs
Input 0
Input 1
Input 2
Input 3
Input 4
Input 5
Input 6
Input 7
Refresh
Monitor
Interrupt Inputs
Sense 0
Sense 1
Sense 2
Sense 3
Micro Sw
Front Ctrl 0
Front Ctrl 1
Front Ctrl 2
Refresh
Monitor



There are two main groups of functionalities on these two pages – outputs and inputs. Although the usage of the I/O pages is quite straightforward, each of the main groups will be described in more detail below. Please also note the **Monitor** check box in all the output and input panes. Checking this box will enable monitoring of the outputs / inputs in the specific pane (i.e. if the **Monitor** check-box is checked in the **Interrupt Inputs** pane the **Front Ctrl 0** check-box will change when pressing and releasing the red button on the front panel). Furthermore, the outputs and inputs can be set/reset by checking/ un-checking the check boxes for each of the inputs and outputs.

Output pane

- **Digital outputs:** This category enables direct control of the 16 digital outputs (8 sink and 8 source outputs).
- **Relays:** This category enables direct control of the 8 relays in the RTX2300 unit.
- **RF Switch:** This category enables direct control of the 10 RF Switch outputs.

- **Fixture control:** This category enables direct control of the four fixture control outputs.
- **Front LED's:** This category enables direct control of the three front LED's – please note only two of these are used in the basic RTX2300 configuration.
- **Various:** This category enables direct control of the air valve and the magnet control (i.e. the air valve can be opened or closed, and the magnet can be enabled/ disabled).
- **Pulse generation:** Through this category it is possible to set/get the pulse mode for each of the outputs (including the four expansion control outputs). The **Pulse mode** can be one of the following:
    - RTX2300_PULSEMODE_OFF
    - RTX2300_PULSEMODE_ON
    - RTX2300_PULSEMODE_PULSE_SHORT
    - RTX2300_PULSEMODE_PULSE_PULSE_MEDIUM
    - RTX2300_PULSEMODE_PULSE_PULSE_LONG
    - RTX2300_PULSEMODE_FLASH_SLOW
    - RTX2300_PULSEMODE_FLASH_MEDIUM
    - RTX2300_PULSEMODE_FLASH_QUICK
    - RTX2300_PULSEMODE_FLASH_LONG_SLOW
    - RTX2300_PULSEMODE_FLASH_LONG_MEDIUM
    - RTX2300_PULSEMODE_FLASH_LONG_QUICK
    - RTX2300_PULSEMODE_FLASH_SHORT_SLOW
    - RTX2300_PULSEMODE_FLASH_SHORT_MEDIUM
    - RTX2300_PULSEMODE_FLASH_SHORT_QUICK
    - RTX2300_PULSEMODE_USER_DEFINED_0
    - RTX2300_PULSEMODE_USER_DEFINED_1
    - RTX2300_PULSEMODE_USER_DEFINED_2
    - RTX2300_PULSEMODE_USER_DEFINED_3

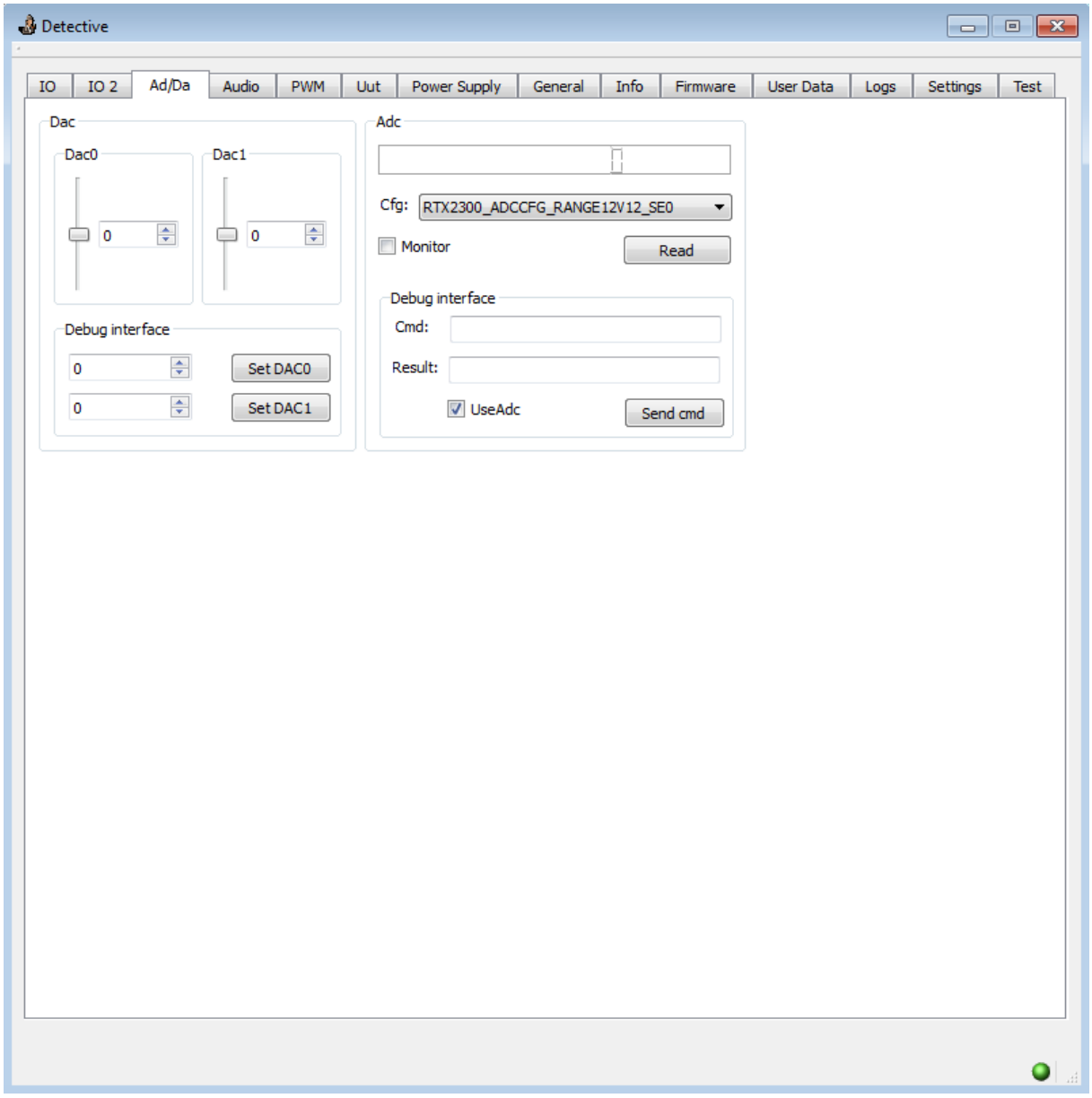
Input pane

- **Digital inputs:** This category enables direct control of the 8 digital inputs.
- **Interrupt inputs:** This category enables direct control of the 4 sense signal inputs and the SW interrupt inputs (including front control interrupts).
- **InputMonitor:** Through this category it is possible to get/set monitoring criteria on all the inputs. The StateChangeMode can be one of the following:
    - RTX2300_STATE_CHANGE_NONE
    - RTX2300_STATE_CHANGE_ACTIVATED
    - RTX2300_STATE_CHANGE_DEACTIVATED
    - RTX2300_STATE_CHANGE_BOTH

ii. Ad/Da Page Overview

The **Ad/Da** page is related to control of the 2 DAC's and one ADC in the RTX2300. The page contains the following panels:

Panel	Parameters	Short description
DAC	<DAC0 value>, <DAC1 value>	Set values for the DAC's
ADC	<ADC value>	Read out the value of the ADC – please note that the range can be configured through the **Cfg** pull-down list



DAC

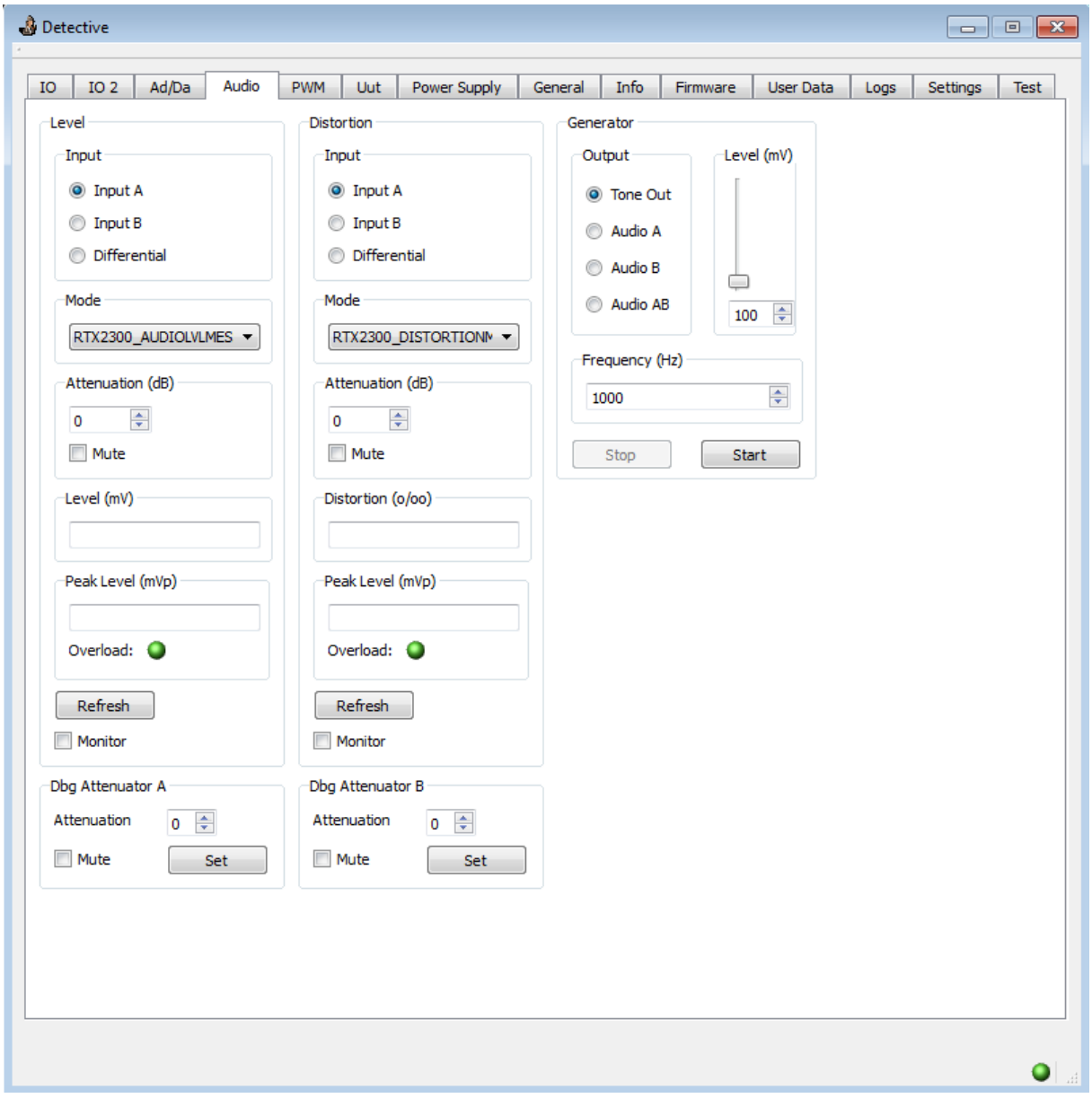
- **Dac0 & Dac1:** The value can be set on both the DAC's by either using the slider, the arrows or using the input field (value range: -10000 to 10000).

ADC

- **Adc:** The ADC value is displayed in the output field and can be monitored by checking the **Monitor** check-box. The value can also be read by clicking the **Read** button. The configuration of the ADC can be changed through selecting one of the pre-configured configurations in the **Cfg** pull-down list. In total there are 69 different configurations (7 x 7 static voltage configurations, 3 x 4 differential voltage configurations).

iii. Audio Page Overview

The **Audio** page is related to controls for the audio functionality of the RTX2300.



The Audio page contains the panels listed in the table below. These will be outlined in more detail in the following. Please note that both the level and distortion can be monitored by checking the **Monitor** check-box. Furthermore, the Tone generator can be started and stopped by clicking the **Start** and **Stop** buttons.

Panel	Parameters	Short description
Level	-	Monitor or refresh level settings
Input (Level)	Input A, Input B, Differential	Set the input
Mode (Level)	<AudioMessMode>	Set the level mode
Attenuation (Level)	<Attenuation in dB>	Set the attenuation
Level (Level)	<Level in mV>	Set the level
Peak Level (Level)	<Peak level in mVp>	Set the peak level
Distortion	-	Monitor or refresh distortion settings
Input (Distortion)	Input A, Input B, Differential	Set the input
Mode (Distortion)	<DistMessMode>	Set the mode
Attenuation (Dist.)	<Attenuation in dB>	Set the attenuation
Distortion (Dist.)	<Distortion in o/oo>	Set the distortion
Peak Level (Dist.)	<Peak level in mVp>	Set the peak level
Generator	-	Start and stop of the tone generator
Output (Generator)	Tone Out, Audio A, Audio B, Audio AB	Set the output for the tone generator
Level (Generator)	Level in mV	Set the level for the output of the tone generator
Frequency (Gen.)	Frequency in Hz	Set the frequency of the tone generator

Level

- **Input:** Select the input – Input A, Input B or Differential.
- **Mode:** The level mode can be set through the **Mode** pull-down list. Two values have been defined:
  - `RTX2300_AUDIOLVLMESS_MODE_RMS`
  - `RTX2300_AUDIOLVLMESS_MODE_PP`
- **Attenuation:** Set the attenuation for the level input by using either the arrows or the input field. The audio can be muted by checking the **Mute** check-box.
- **Level:** Set the level on the input.
- **Peak Level:** Set the maximum peak level on the input. The *Overload* indicator will indicate whether or not the peak level has been exceeded (red = overload detected, green = peak level not exceeded).

Distortion

- **Input:** Select the input – Input A, Input B or Differential.
- **Mode:** The distortion mode can be set through the **Mode** pull-down list. However, only one value is defined (`RTX2300_DISTORTIONMESS_MODE_THD`).
- **Attenuation:** Set the attenuation for the level input by using either the arrows or the input field. The audio can be muted by checking the **Mute** check-box.
- **Distortion:** Set the distortion level.
- **Peak Level:** Set the maximum peak level on the input. The *Overload* indicator will indicate whether or not the peak level has been exceeded (red = overload detected, green = peak level not exceeded).

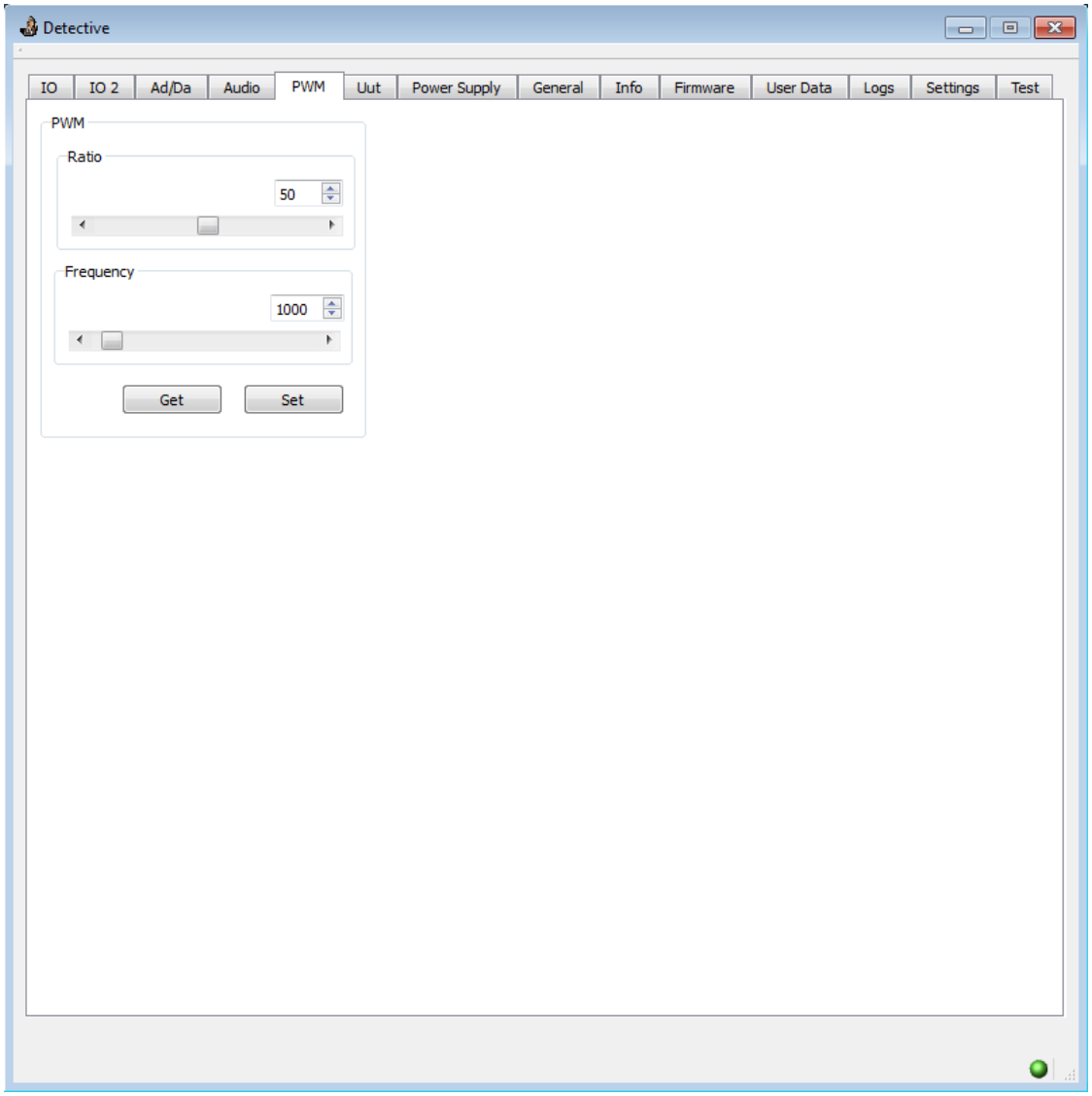
Generator

- **Output:** Set the output for the tone generator – Tone out, Audio A, Audio B or Audio AB.
- **Level:** Set the level of the tone generator by using the slider, input field or the arrows.
- **Frequency:** Set the frequency of the Tone generator.

iv. PWM Page Overview

The PWM page contains the following panels:

Panel	Parameters	Short description
Ratio	<Ratio>	Set the ratio
Frequency	<Frequency>	Set the frequency

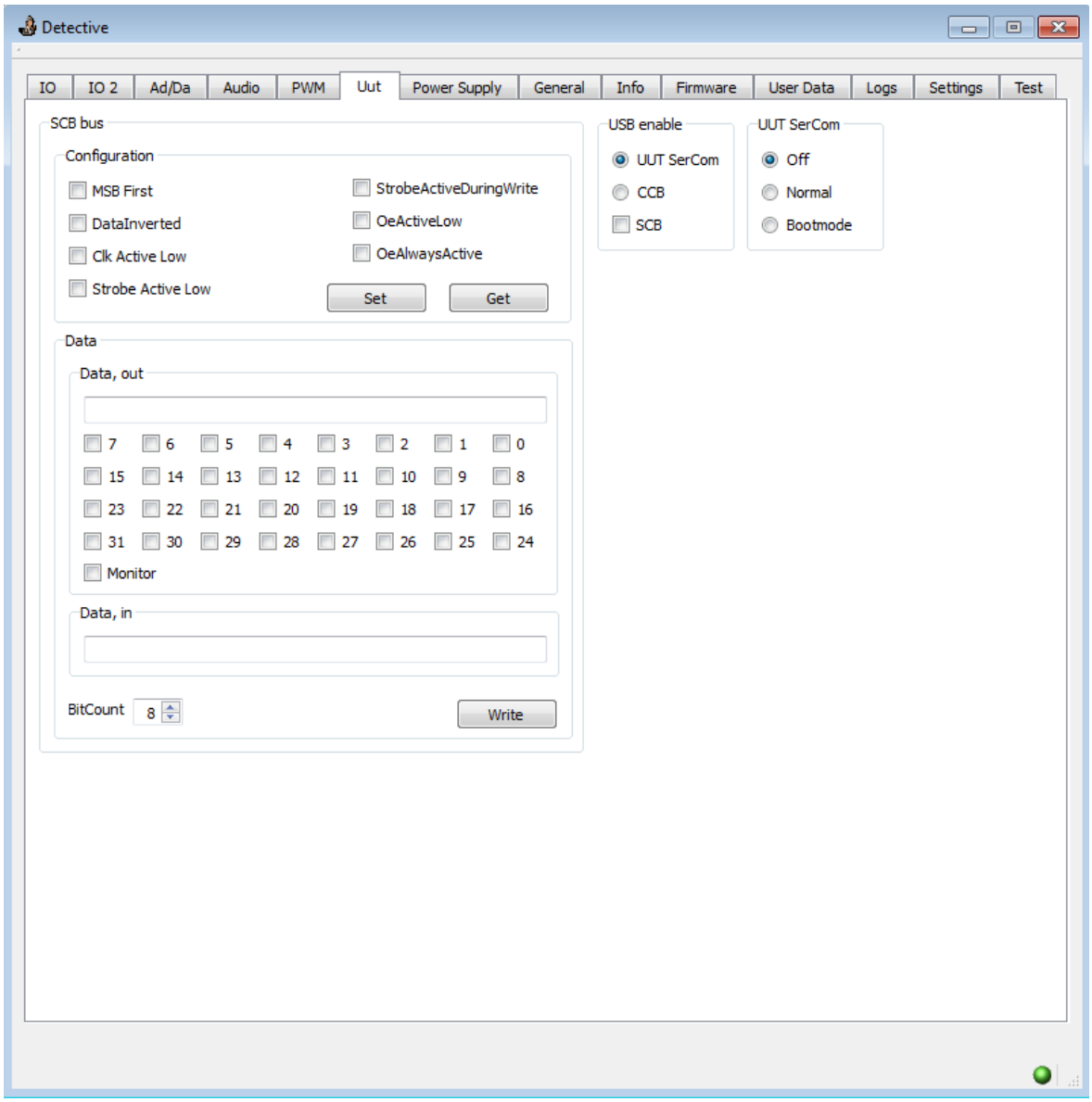


- **Ratio:** Set the PWM ratio by using the slider, arrows or input field.
- **Frequency:** Set the PWM frequency by using the slider, arrows or input field.

v. DUT Page Overview

The **DUT** page contains central controls for the DUT interface and it contains the following panels:

Panel	Parameters	Short description
SCB bus – Configuration	-	Configuration of the SCB bus
SCB bus – Data	-	Write specific value to the SCB bus
USB enable	DUT SerCom/CCB, SCB	Enable/disable USB at SCB interface and select either CCB or DUT SerCom
DUT SerCom	Off, Normal, Boot-mode	Select operation mode for the DUT SerCom interface



SCB bus

- **Configuration:** The SCB bus can be configured through the controls in this pane. Check the check-box to enable the different configuration items.
- **Data:** A specific value can be written directly to the SCB bus by using the Data input and clicking the **Write** button.

USB enable

The USB interface at the DUT can be configured through this panel. The SCB bus USB interface can be enabled or disabled. Furthermore, the second USB interface can be set to either CCB (i.e. USB interface enabled at the CCB interface) or DUT SerCom. If the latter is enabled, it must be configured through the **DUT SerCom** panel (see below).

DUT SerCom

The radio buttons in this panel is used for configuring the DUT SerCom mode. As mentioned in chapter 1 on page 22 the DUT Serial Communication interface has three modes; 1) Off, 2) Normal, and 3) Boot-mode.

vi. Power Supply Page Overview

The **Power Supply** page includes controls for the internal PSU and it contains the following panels:

Panel	Parameters	Short description
Voltage	<Voltage>	Set the output voltage
Current	<Current>	Set the output current
Current range	<Current range>	Set configuration of the current limiter
Power On / Off	-	Turn PSU off/on
PSU Selection	Internal / External	Select PSU – internal or external
Measurements (voltage, current, peak current)	-	Voltage and current measurements

Please note that the PSU can be turned off/on by clicking the **Power Off**/**Power On** button.

Voltage

The voltage output of the PSU can be set through the control or by using the input field or the arrows. The voltage can be set in mV steps in the range 0 – 15000 mV.

Current

The current output of the PSU can be set through the control or by using the input field or the arrows. The current can be set in mA steps in the range 0 – 2000 mA.

Current range

The RTX2300 internal PSU includes a current limiter and it is configured through the pull-down list. The limiter can be set to use one of the following configurations:

- `RTX2300_CURRENT_RANGE_AUTO` (current limiter disabled)
- `RTX2300_CURRENT_RANGE_100MA`
- `RTX2300_CURRENT_RANGE_500MA`
- `RTX2300_CURRENT_RANGE_1000MA`
- `RTX2300_CURRENT_RANGE_2000MA`

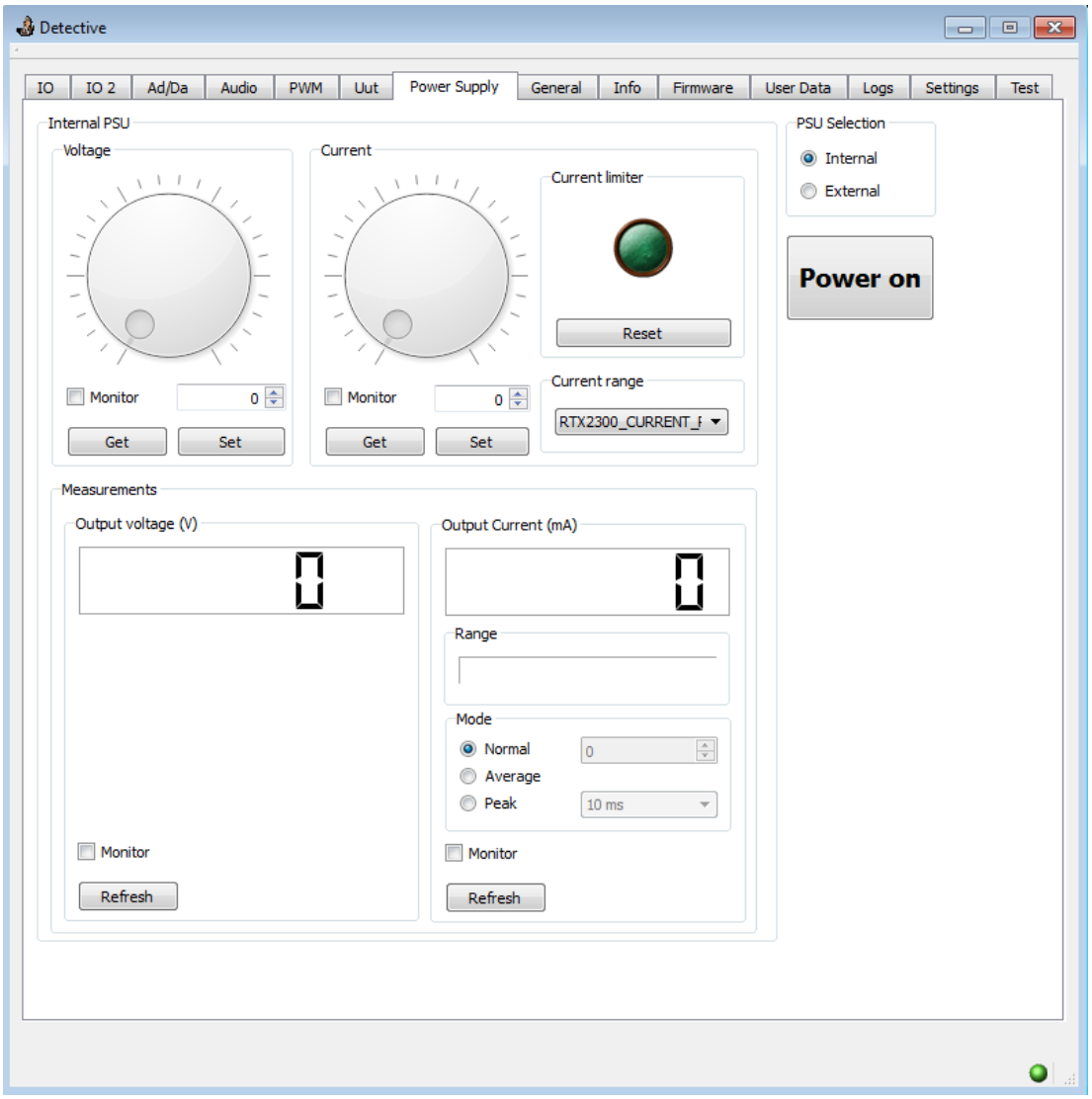
If the current limiter has been set the indicator in the Current limiter pane will indicate whether the current limit has been reached. If the current limit has not been reached it will remain green otherwise it will become red.

PSU Selection

The internal PSU can also be disabled by choosing External. In this case the external PSU is used (i.e. an external PSU connected to the Ext. Power connector on the rear panel of the RTX2300).

Measurements

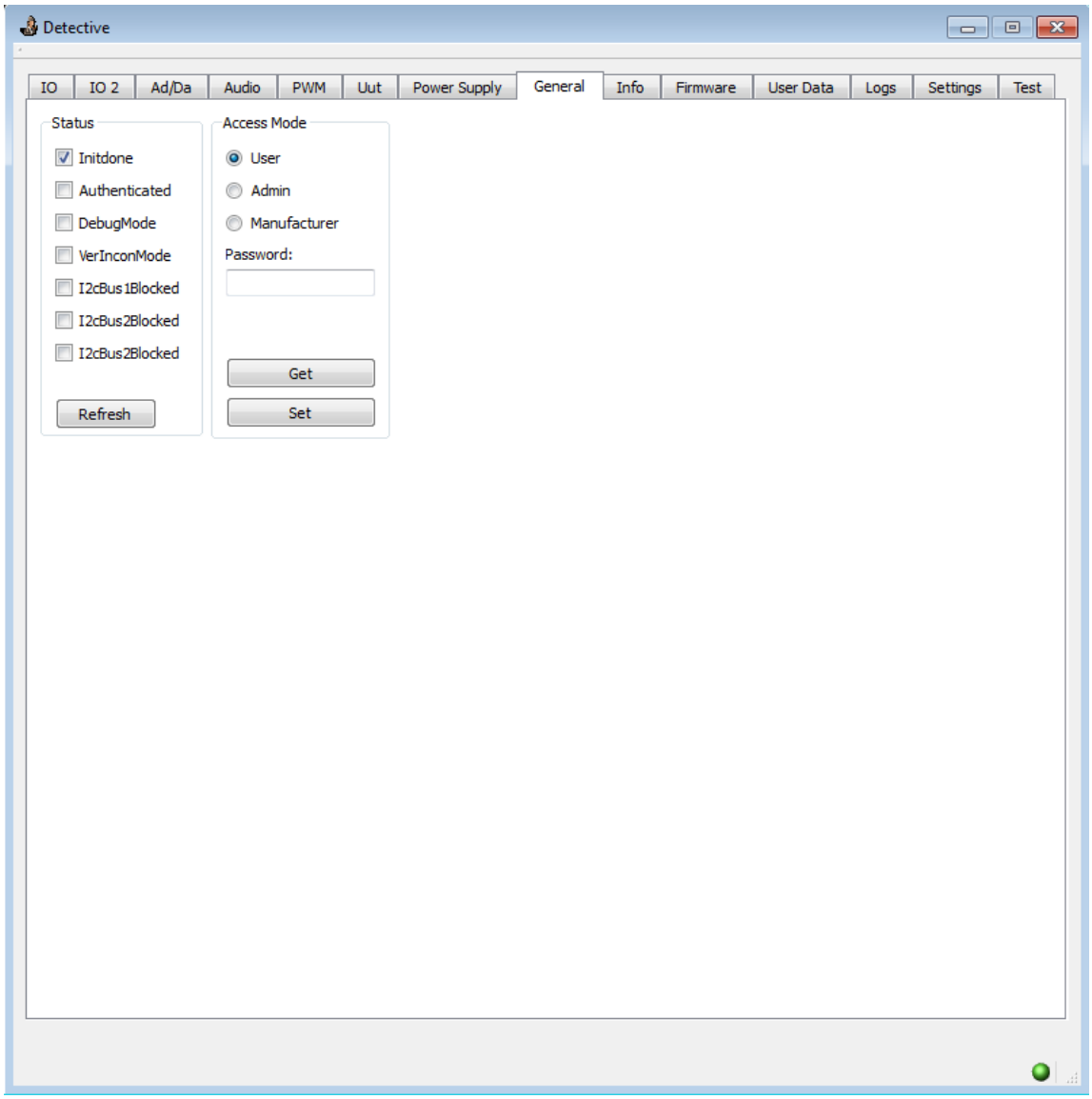
The controls under the measurement panel are related to voltage and current (including peak) measurements. Each can be monitored (i.e. the measurements will be updated automatically) by checking the **Monitor** check-box.



vii. General Page Overview

The **General** page includes general controls and it contains the following panels:

Panel	Parameters	Short description
Status	-	Status indicators
Access Mode	User/Admin	Setting of access mode and password



Status

The status panel provides an overview of some items of information.

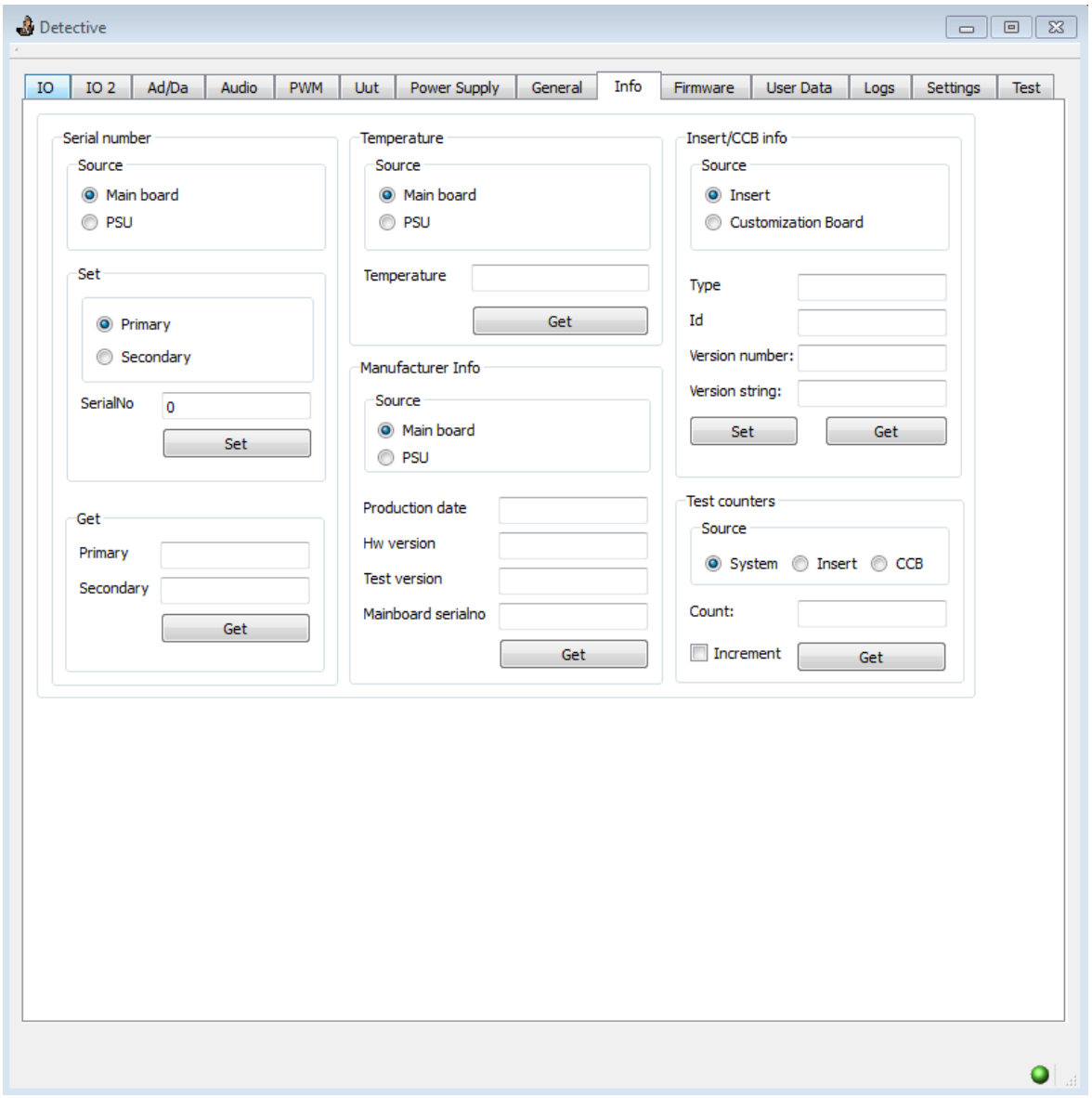
Access Mode

In the access mode panel, the access mode can be set to either User or Admin, and furthermore, the password can also be set.

viii. Info Page Overview

The **Info** page provides an overview of the Serial number, Temperature, Manufacturer Info, Insert/CCB info, and Test counters.
The **Info** page contains the following panels:

Panel	Parameters	Short description
Serial number	<Serial number>	Set and get primary/secondary serial number of both main-board and PSU
Temperature	-	Get the temperature of either the main board or the PSU
Serial number	-	Get the manufacturer information for either the main board or the PSU. - Production date - HW version - Test version - Main board serial no
Insert/CCB info	<Version number>, <Version string>, <Type>, <Id>	Set and get version number and version string for the Insert as well as the Type and Id of the Insert
Test counters	-	Get the number test cycles performed by either System, Insert, or CCB.



Serial number

the serial number pane is divided into three sub-panes:

- **Destination** – In this sub-pane the destination for the get/set operation selected. Please note that it is only possible to get/set the primary and secondary serial numbers for the main-board and the PSU module (if present).
- **Set** – This sub-pane includes functionality for setting the primary and secondary serial number. Use the radio-buttons to select primary or secondary serial number setting, enter a number in the input field and click the **Set** button to set the serial number.

- **Get** – This sub-pane includes the functionality for reading the primary and secondary serial numbers. Select a destination (main-board or PSU) and click the **Get** button to read the serial numbers from the destination. The serial numbers are displayed in both decimal and hex-format (i.e. in the form '<serial number as decimal number> / <serial number in hex-format>').

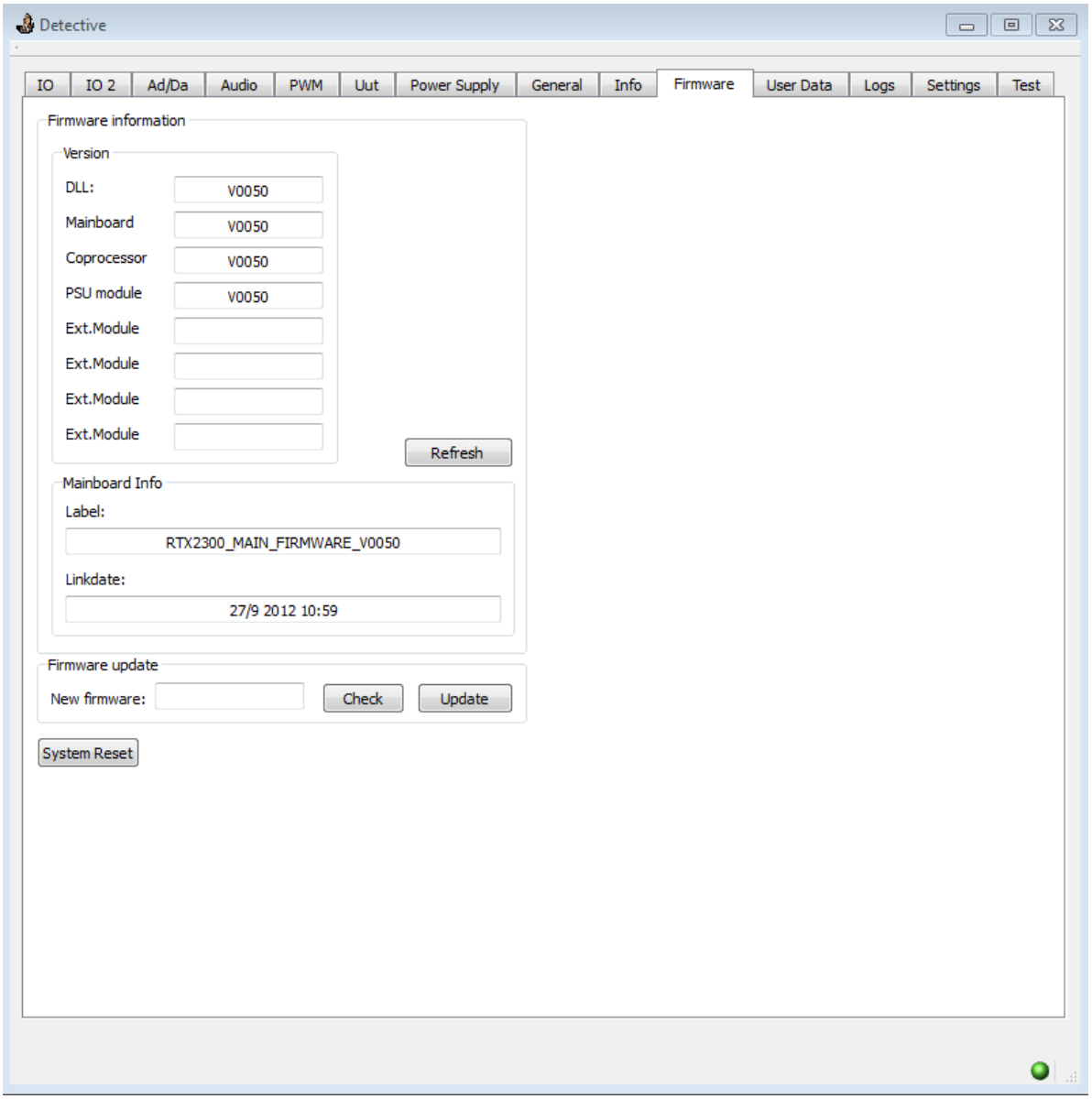
Insert/CCB Info

The Insert version number and Insert version string can be set by entering information in the two input fields and clicking the **Set** button. The same information can be read from the Insert by clicking the **Get** button.

ix. Firmware Page Overview

The **Firmware** page provides an overview of the firmware version of the main-board, PSU Module, Co-processor and other extension boards along with the SW version of the RTX2300 DLL.
Please also note the **System Reset** button in the bottom of the **Firmware** page. Clicking this will reset the system. The **Firmware** page contains the following panels:

Panel	Parameters	Short description
Firmware information	-	DLL SW version and firmware version of main-board (including link date and label information), co-processor, PSU module and extension modules
Firmware update	<Firmware file>	Check and upload firmware file
Mainboard Info	<Label string>	Read the label information of the firmware loaded onto the mainboard
	<Linkdate>	Date of the firmware upload



Firmware information

The firmware information pane provides an overview of the firmware version of the following items:

- DLL
- Main-board
- Coprocessor
- PSU Module (if not present 'N/A' will be shown – as indicated in the figure below)
- Other extension modules

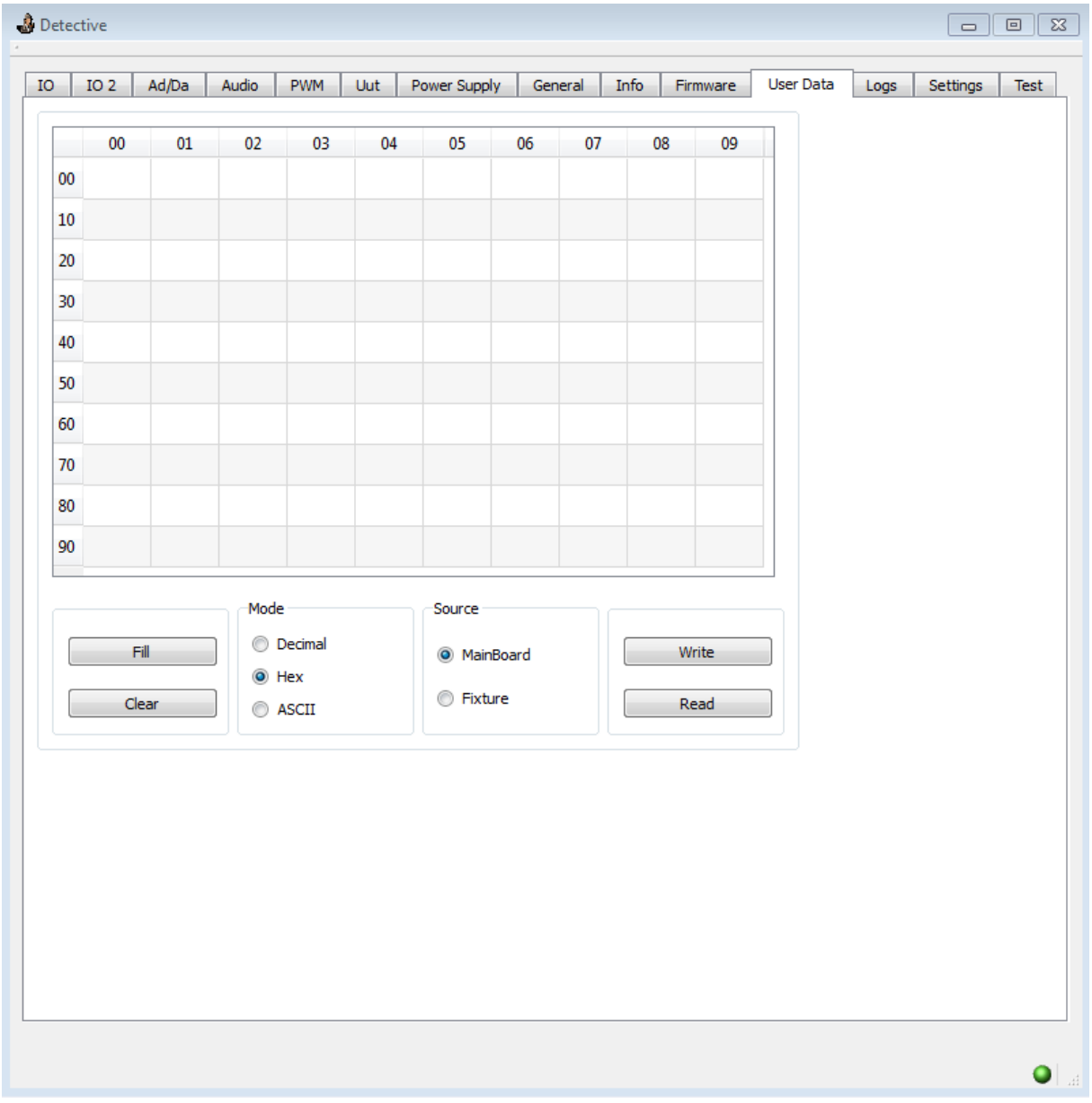
Furthermore, other relevant items of information (e.g. label and link-date) related to the main-board firmware is also included in the info area in the firmware information pane.

Firmware update

Firmware can be checked / updated by clicking **Check** / **Update**. Clicking **Update** will start the firmware update process (i.e. the RTX2300 Detective will update the firmware of the RTX2300 with the file located in the installation directory). Clicking **Check** will make the RTX2300 Detective check the current firmware version and compare it to the firmware file located in the installation directory.

x. User Data Page Overview

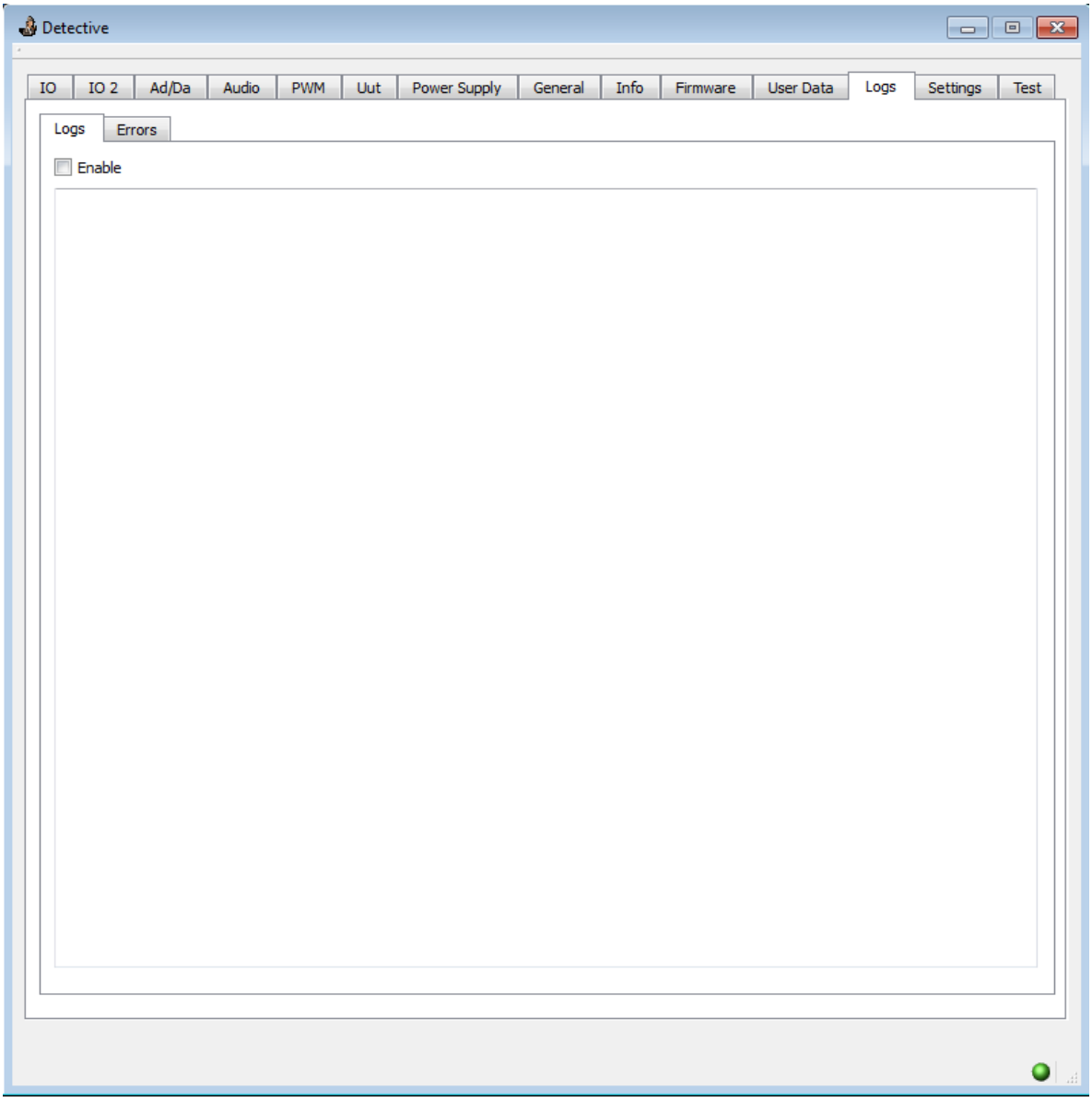
the **User Data** page is used for retrieving/modifying values in the user data area (100 bytes). The user data area can both be set for the main-board and the fixture. Furthermore, the values can be shown/entered in Decimal, Hex, or ASCII.



xi. Logs Page Overview

The **Logs** page provides detailed information in relation to the communication between the RTX2300 Detective SW and the RTX2300 unit. It contains the following panels:

Panel	Parameters	Short description
Errors	-	Overview of errors
Logs	-	Detailed logs



Logs

In this window the detailed communication between the RTX2300 Detective application and the RTX2300 unit is outlined. Both the message name and message values are shown.

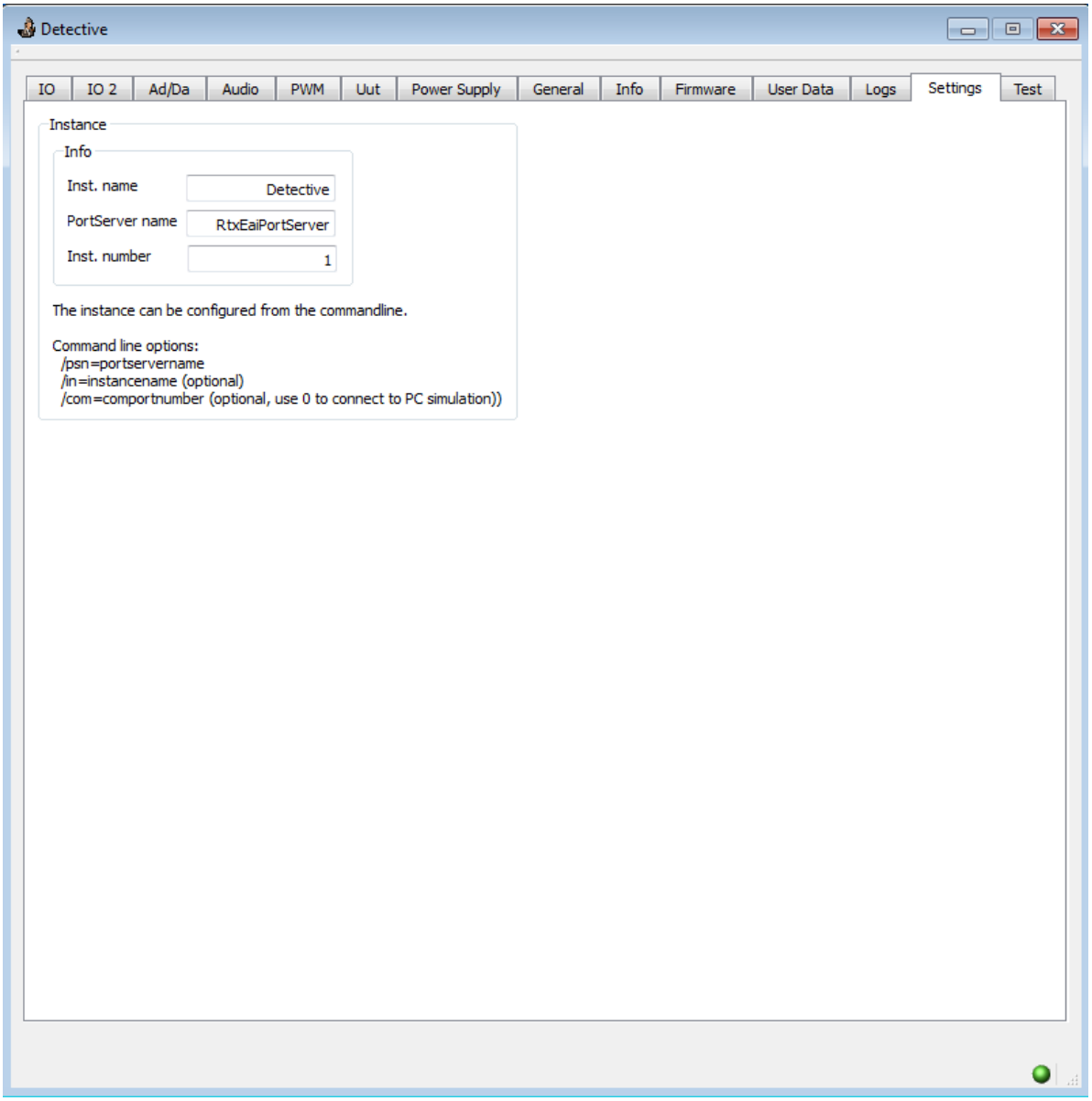
Errors

This window will show a log of errors encountered.

xii. Settings Page Overview

The **Settings** page is used for connecting the RTX2300 Detective application to the RTX EAI Port Server. it consists of three items of information:

- **Inst. name:** The name of the program (i.e. in this case it is set to Detective).
- **PortServer name:** The name of the port server to connect to.
- **Inst. number:** The instance number of this program.



3. Using the RTX2300 Windows SW

A. Introduction

In this chapter, the focus is on using the supplied RTX2300 SW. The following items of information will be described:

- **Using the RTX2300 Detective debug application** – This section includes important items of information in relation to launching the RTX2300 Detective application. Moreover, a brief description of how to use the application when debugging customization cards is also included.

- **Performing unit tests with the RTX2300** – This section includes a brief description on how to perform unit tests with the RTX2300.

B. Using the RTX2300 Detective debug application

This paragraph outlines how to launch and use the RTX2300 Detective debug application. Prior to launching the RTX2300 Detective debug application, please ensure that:

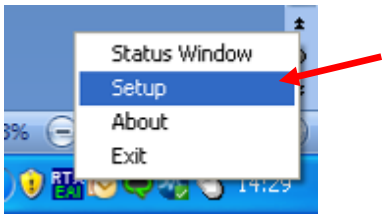
- The RTX2300 Detective debug application has been correctly installed on your PC (as part of the RTX2300 Basic Unit SW package)
- The RTX2300 unit is connected to your PC using the USB interface
- The RTX2300 unit is switched on
- The RTX EAI Port Server has been correctly configured

It is strongly emphasized here that it is crucial to configure the RTX EAI Port Server correctly. Hence, if an error is encountered when launching the RTX2300 Detective debug application it is most likely related to an erroneous configuration of the port server. Please refer to paragraph i in this section for instructions on how to configure the RTX EAI Port Server correctly (i.e. select the correct transport layer, setup the correct COM port and other communication settings).

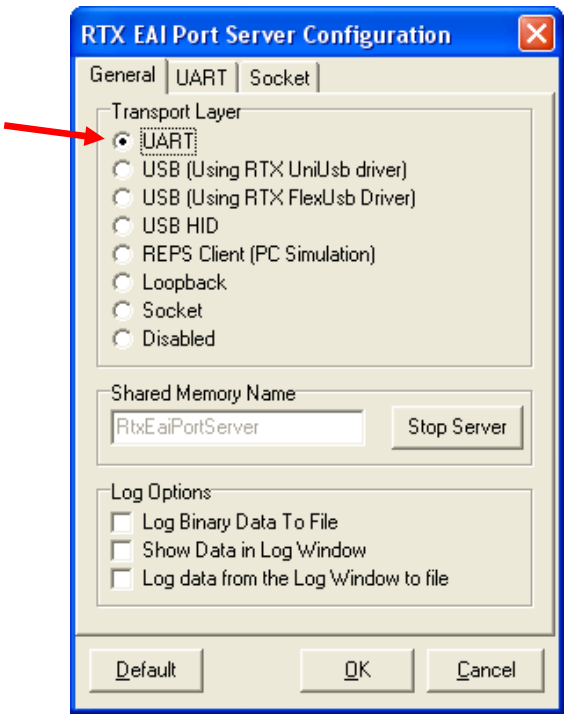
i. Configuring the RTX EAI Port Server (REPS)

The RTX EAI Port Server (REPS) is started automatically when needed by the RTX2300 DLL. It can also be started manually by selecting the `RTX EAI Port Server` from the `Start->All Programs` menu (specific location depends on the location entered under installation of the REPS – default is `Start->All Programs->RTX`). Once started the REPS will run in minimized mode and a RTX EAI Port Server icon will appear in the notification area of the Windows task bar (as shown in the figure below).

The Setup window can be opened by selecting **Setup** in the context menu. To open the context menu move the cursor to the REPS icon and click on the right mouse button (see below).



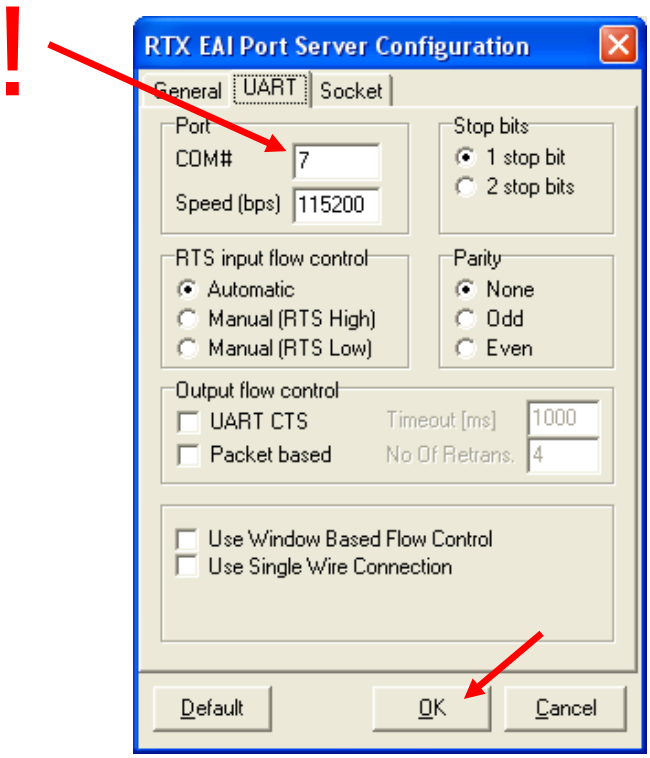
Select **Setup** by clicking the left mouse button and the **RTX EAI Port Server Configuration** window should now appear (see below).



Select the **UART** radio button on the **General** tab to select the UART Transport Layer and click **OK**. Next select the **UART** tab and select the correct COM port to use for communicating with the RTX2300 and ensure that the following is set (as also indicated in the figure below):

- **Speed (bps)** is set to 115200
- **Stop bits** set to **1 stop bit**
- **Automatic RTS input flow control** set to **Automatic**

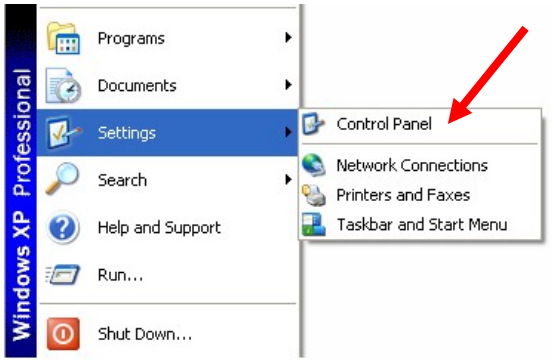
In this example COM port number 7 is the correct port number but it is strongly emphasized here that the correct COM port number is related to a specific PC and the COM port number of the installed RTX2300 Basic Unit Communication Interface driver must be checked in the Windows Device Manager (steps on how to do this is given later in this paragraph).



Press **OK** and the REPS should now be able to communicate with the RTX2300 (if connected to the PC and the correct COM settings have been entered). The RTX2300 Smart ATE unit should therefore now be ready for use.

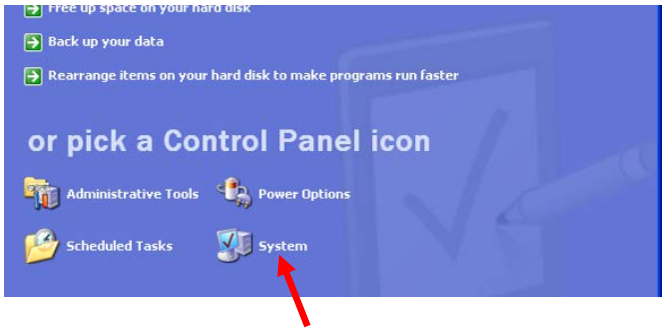
As mentioned above it is crucial to configure the REPS correctly – i.e. using the correct communication settings and COM port. To figure out which COM port to use for RTX2300 unit communication the available ports on the PC needs to be checked in the Windows Device Manager. To open the Device Manager please refer to the steps outlined below.

❶ Select **Settings** from **Start** button. Select **Control Panel**.

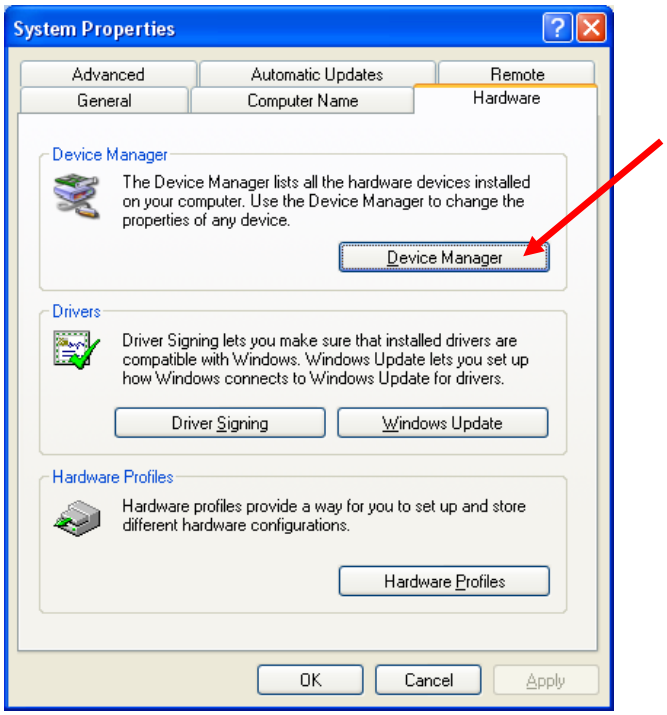


❷ Click **Performance and Maintenance**

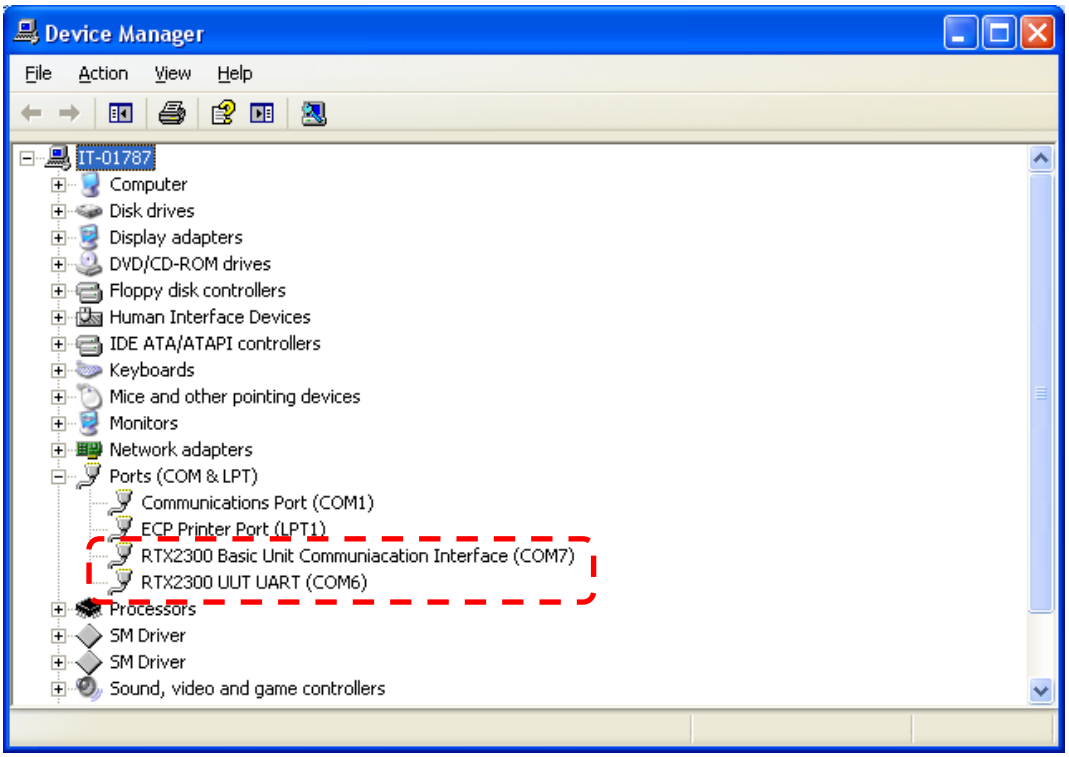
❸ Click **System**



❸ Select the **Hardware** pane and click **Device Manager**



The Device Manager window will now be opened. Locate the `Ports (COM & LPT)` category and click on the + to open the list with available ports.

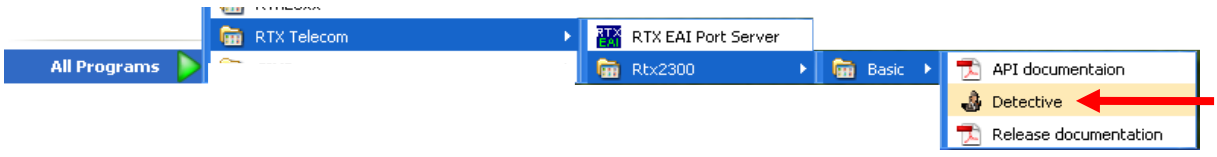


If the drivers have been correctly installed and the RTX2300 unit is connected to the PC the list should include the two indicated COM ports for the RTX2300 Basic Unit Communication Interface and the RTX2300 DUT UART, respectively. Please note that the RTX2300 Basic Unit Communication Interface is available through COM port number 7 on this PC, and hence, this is the port number to use in the REPS configuration.

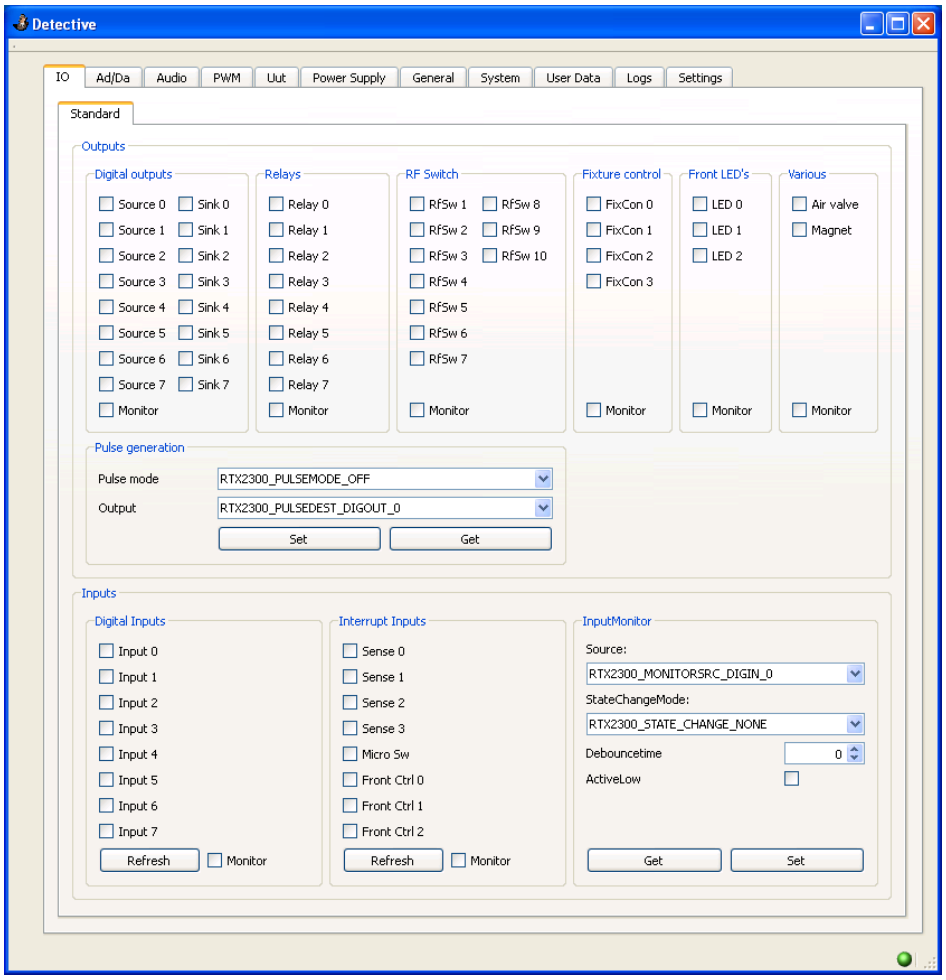
ii. Launching and use of the RTX2300 Detective application

Launching RTX2300 Detective

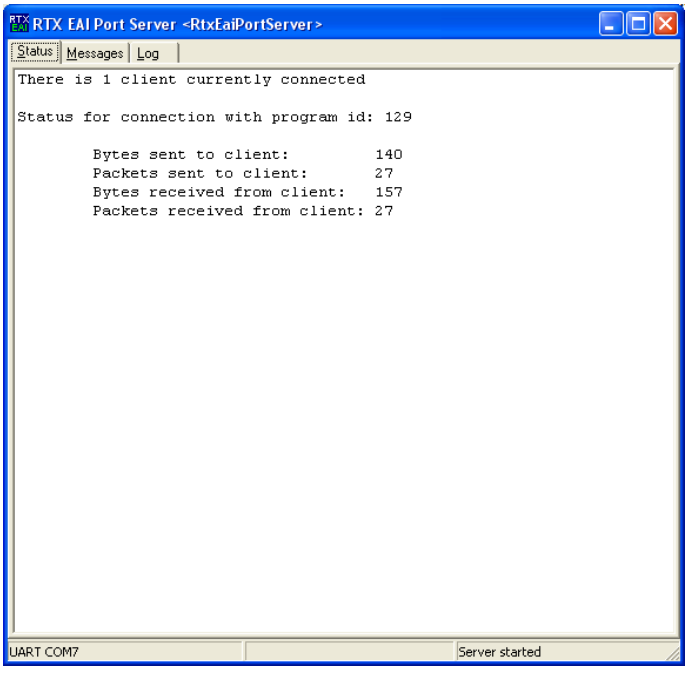
After checking that the REPS configuration is correct the RTX2300 Detective debug application can be launched. The RTX2300 Detective application can be launched either through `All Programs->RTX->Rtx2300->Basic->Detective` (see below) or by double clicking the `Detective.exe` file in a windows explorer (default location is `C:\Program Files\RTX\Rtx2300\Basic`).



The RTX2300 Detective application should start-up after a few seconds and the main application window should appear (see figure below).



Please notice the green indicator in the bottom right corner. This indicates that the connection to the RTX2300 unit has been established. Please note that the status for the connection can be seen in the **Status** page in REPS (see below).



Using the RTX2300 Detective application for debugging purposes

The RTX2300 Detective application is an essential tool when debugging for example customization boards or performing specific measurements on DUT test points. From this application the user has complete control over the functional blocks of the RTX2300, and hence, for example the signal routing can be effectively controlled by switching internal relays. Furthermore, the support for monitoring of internal RTX2300 signals also makes the Detective application valuable debug tool.

The RTX2300 Detective application is quite straightforward to use, and furthermore, the specific usage of the application depends to a large extent on the configuration and usage of the RTX2300 unit itself. Consequently, it is not possible to provide exact instructions on how to use the application but in general terms a debugging procedure typically involves three steps:

1. Execute steps (i.e. the test program) to bring the DUT into the correct state – i.e. the state in which measurements should be performed

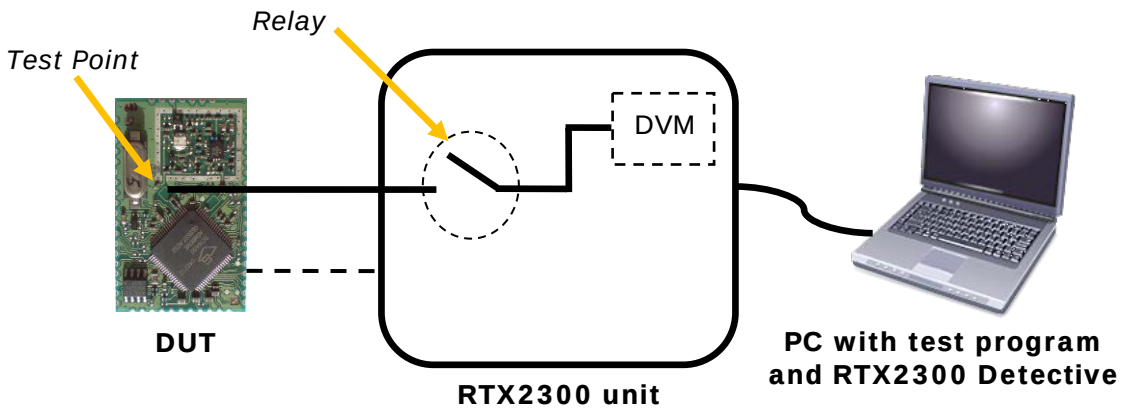
2. Configure the signal path by switching the correct relays in the RTX2300 through the RTX2300 Detective application

3. Perform measurements with the RTX2300 Detective application

To illustrate how to use the RTX2300 Detective application using the three step debugging approach outlined above a small example will be included below. The example will not include any screen dumps but instead refer to the relevant RTX2300 Detective pages and selections.

In this example it is assumed that the test program will put a DC voltage on a test point and that the task is to check the DC voltage at the test point (see figure below). Furthermore, it is assumed that the test setup is correctly connected and operating (i.e. the DUT is placed in the fixture and ready for testing).

Example – setup for measurement of DC voltage at a test point



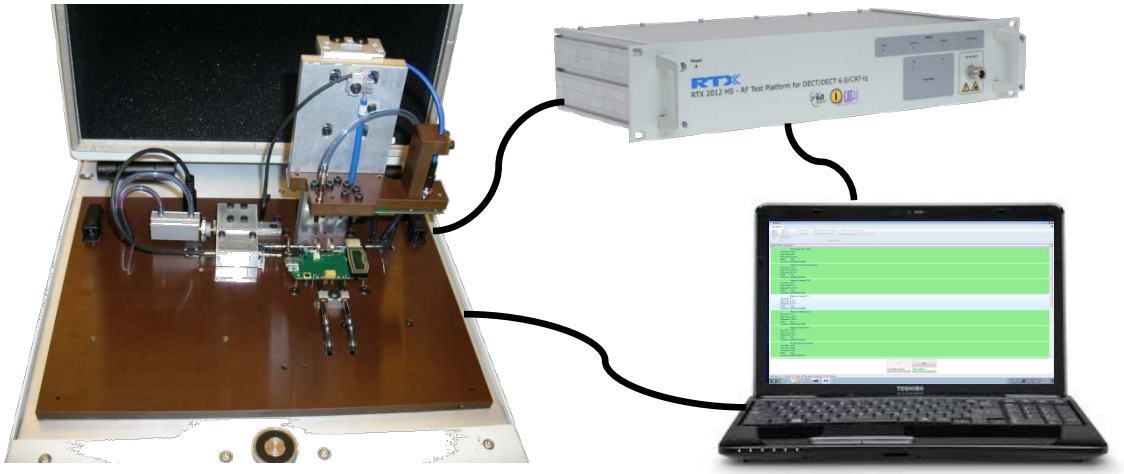
The first step is to execute the test program (from the connected PC) to bring the DUT into the correct state – i.e. so that the DC voltage will be set at the test point. Next the RTX2300 Detective application is started and the relevant relays are set in the **Relay** panel of the **Output** category on the **IO** page (i.e. the startup page of the RTX2300 Detective

application). To measure the DC voltage at the test point, go to the **Ad/Da** page (i.e. the DC DVM functionality) and check the measured DC voltage in the **Adc** panel. Please ensure to use the correct range configuration. If the DUT should output say 1.55V at the test point, the measured value in the **Adc** pane should be 1.55V.

C. **Performing unit tests with the RTX2300**

Performing unit tests with the RTX2300 requires as a minimum a RTX2300 unit and a PC with the test application along with a DUT. Furthermore, in a typical test setup for wireless devices an RF tester is also included in the test system (see the figure below). Additional components are typically not needed since the RTX2300 includes all the necessary basic functional blocks and extension modules can be added if needed.

Example test setup – RTX2300, RTX2012, PC with test application, and a DUT

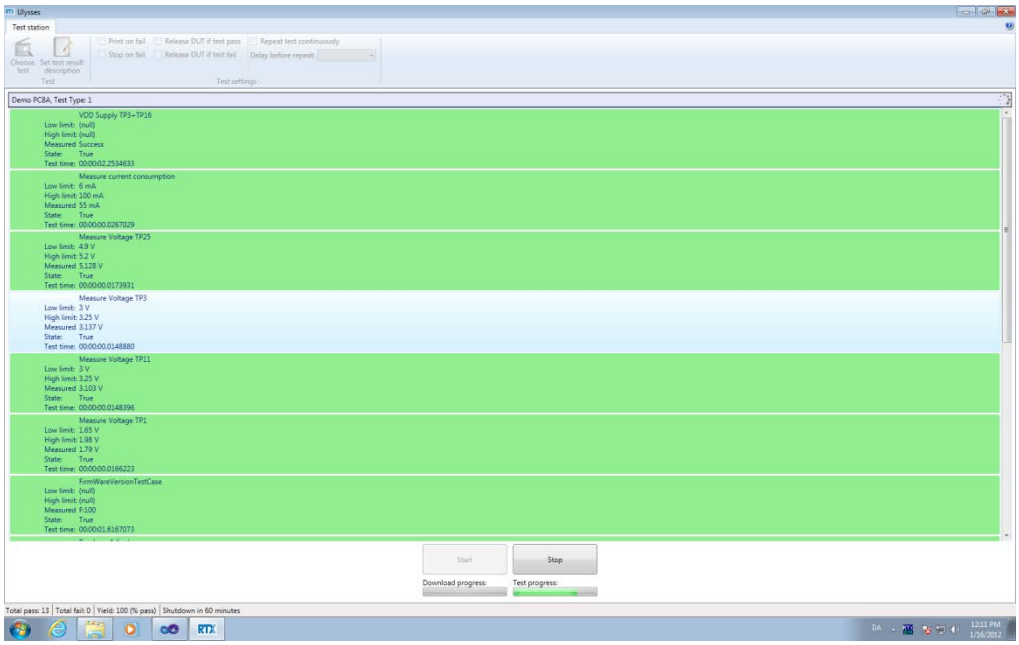


A central part of the system is the PC test application. this application controls all aspects of the test procedure. Please note that the visual appearance of the PC test application (i.e. the GUI) as well as the test suites must be implemented by the customer since this is not a part of the RTX2300 SW package. It is therefore strongly emphasized here that the PC test application described in this section is an example only. The main objectives are to show typical functionality of a PC test application and to provide inspiration for custom PC test applications. Please note that example PC test application is built on top of the RTX2300 API, and hence, customers can create test applications with exactly the same features as the one outlined here. The GUI part of the PC test application can be implemented using Java Swing, Microsoft .Net or similar frameworks.

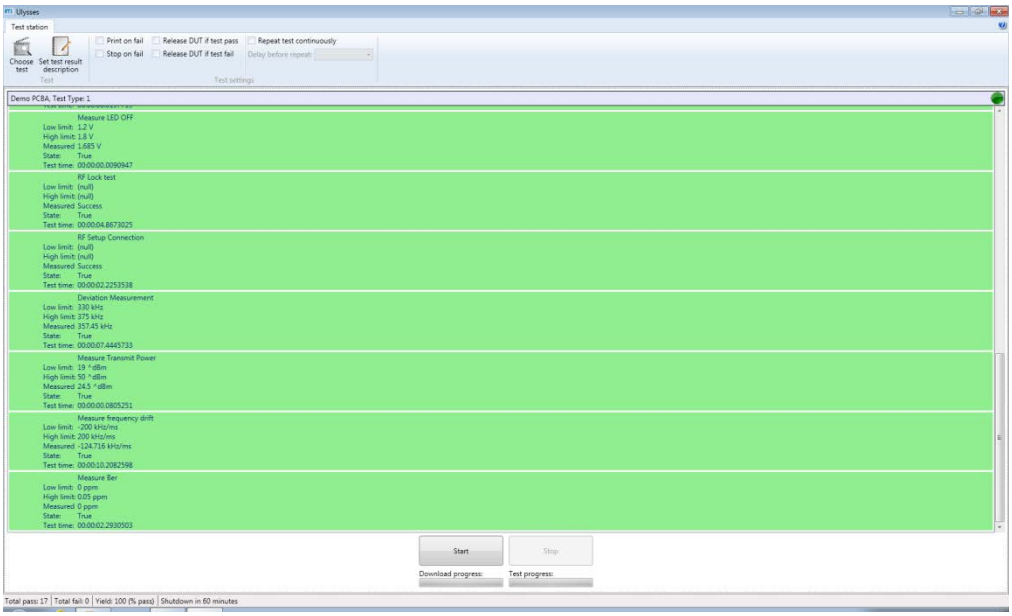
In the following the unit test procedure using the above test setup will be outlined. However, since the detailed unit test procedure to a large extent depends on the specific devices to test, the procedure will be outlined in general terms only. It is assumed that the above system is operational and ready for test execution (i.e. all equipment is switched on and a suitable test suite can be executed from the implemented PC test application).

The first step is to place the DUT in the fixture and start the test from either the RTX2300 unit (by pressing the start button) or the test application on the PC. The test suite will start executing and from the PC test application it is possible to keep track of both the current test case and the completed test cases. If test cases pass the background of the test cases will be green and likewise if test cases fail the background will be red. Below some screen dumps from the example PC test application are outlined.

The first screen dump shows a running test case – please note the test progress status bar in the bottom of the page. The test execution can be stopped by pressing the **Stop** button on the PC test application (or on the RTX2300).

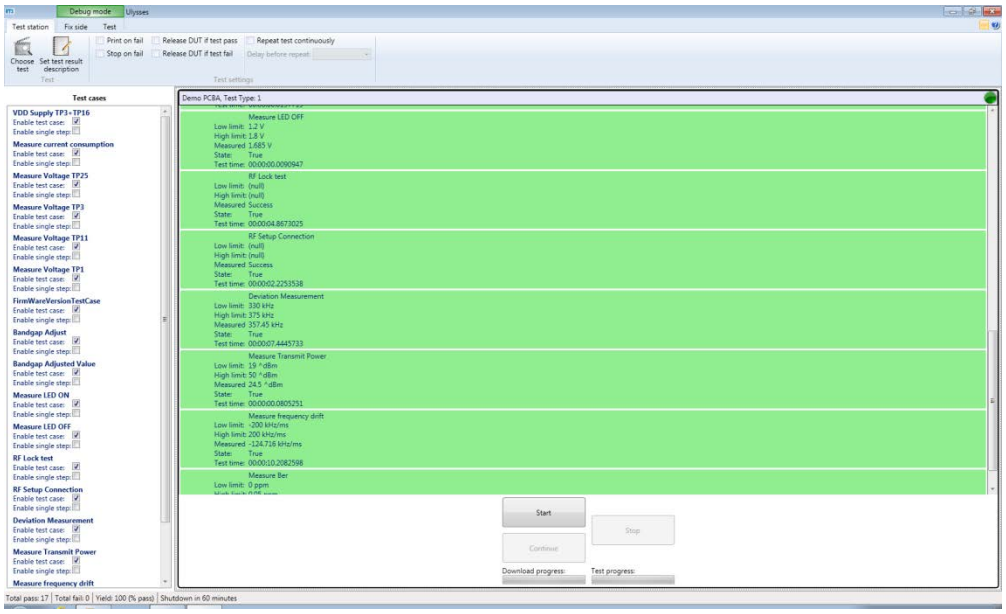


The test suite will run for some time hence performing the defined test cases. Upon completion of the test suite the results can be seen in the PC test application (see the screen dump below).

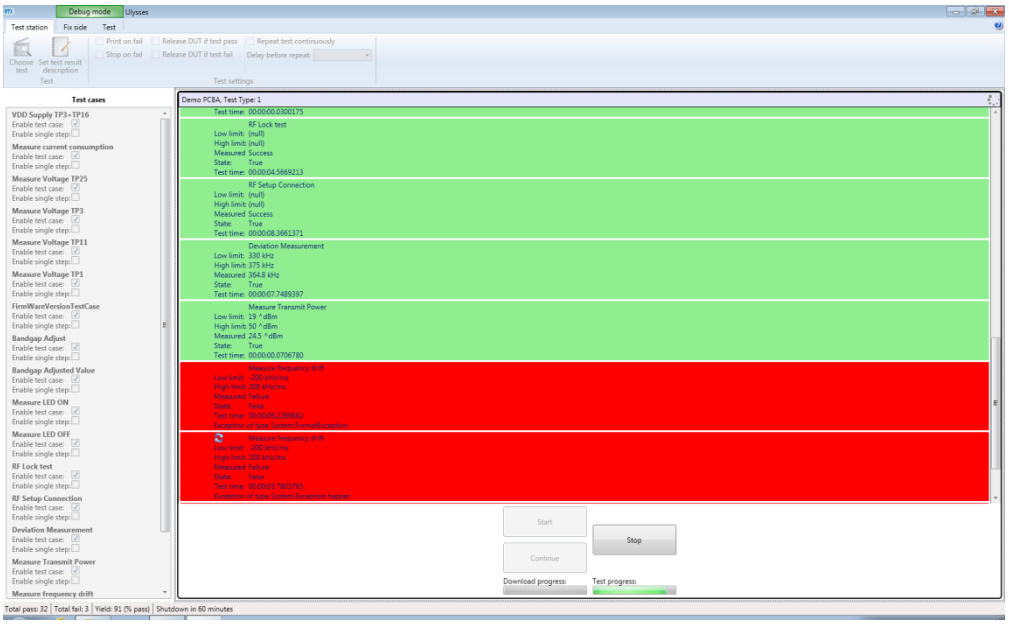


Please notice the items of information in the bottom left of the window regarding passed test cases. The test suite can be restarted again by pressing the **Start** button on the PC application (or on the RTX2300). The two screen dumps above show the flow in normal mode.

For debugging purposes, it could be feasible to implement some level of debugging functionality in the PC test application (or in a separate PC application). The example PC test application outlined here includes a special debug mode. In this mode the user has access to some advanced features for debugging purposes. For example, it is possible to enable single step, hence stepping through a test case one step at a time. Furthermore test cases can also be enabled/disabled on-the-fly. A screen dump from a running test suite in debug mode is outlined below. Please notice the extra functionality in the left pane of the window and the extra **Continue** button in the bottom of the window.



Below another screen dump with the test application in debug mode is shown. Here the test suite is almost completed (91% progress) but some of the test cases failed (i.e. indicated with the red background). Please note the reason code in the last line of the test case information.



If all test cases in the test suite passed, then the test run was successful. If some of the test cases failed the reason for this must be identified. Here the single step feature is a valuable tool.

4. RTX2300 Smart ATE Options

The basic configuration of the RTX2300 includes a wide range of functionality but it can be further extended through some options. These options can be categorized into four different categories:

- **Modules** (Programmable PSU, frequency counter, RF switch)
- **Quick-Swap Kit** (fixture bay part and fixture part)
- **Fixture Kits** (with or without pneumatic slide)
- **Connectivity options** (RJ45 connection to the DUT and SPI or I2C DUT interface)

The available options in each of the categories will be outlined in more detail in the following sections. In addition to the options above the RTX2300 can also be delivered without the shielded lid (option).

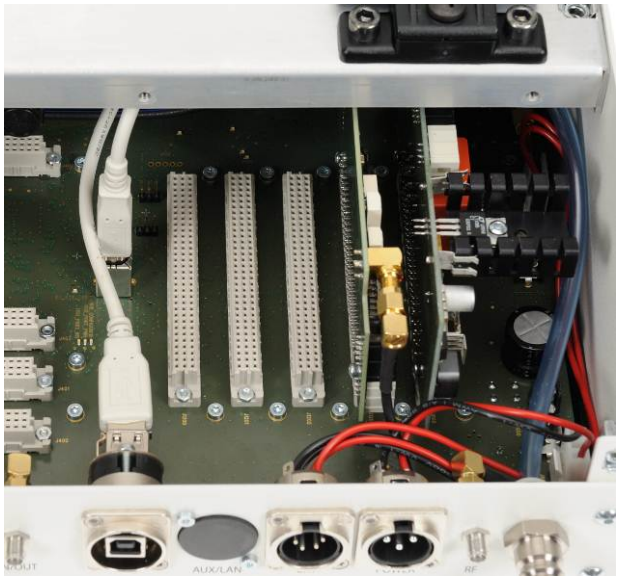
A. **Modules**

The first category is modules and currently four types of optional modules are available:

- Programmable PSU
- Frequency Counter (standard and high-stability)
- RF Switch (4 or 8 channels)

The RTX2300 Smart ATE supports up to five modules (i.e. one PSU and four other modules). The optional modules are installed in the extension slots in the controller chamber (see figure below).

RTX2300 equipped with two optional modules in the controller chamber



i. Programmable PSU Module

The programmable PSU module is an adjustable power supply for the DUT. It is adjustable from 0 V to 15 V and can supply max. 2 A with an adjustable current limiter.

Programmable PSU Module

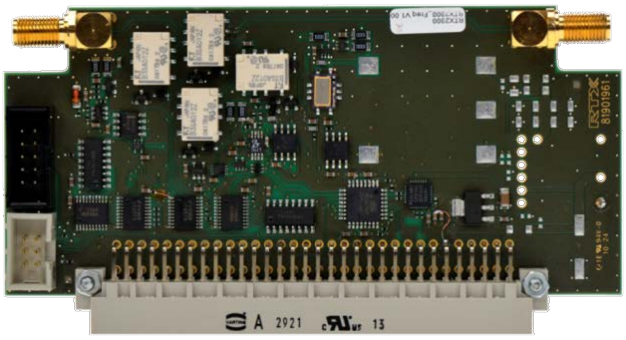
The programmable PSU module includes the following functionalities:

- 0-15 V 2A / 10-15V 1A output voltage in 10 mV steps
- Current limiter 0-2 A current limiter
- Output voltage measurement
- RMS current measurement
- Peak current measurement

ii. Frequency Counter Module

The frequency counter module is available in two versions – a standard grade and a high-stability version. The high-stability version has a high precision time base (<0.1 ppm). Both are prepared for external reference and have an impedance of 50 Ω.

Frequency Counter Module



Standard Frequency Counter

The standard grade frequency counter module includes a frequency counter up to 50 MHz with a time base accuracy of 1 ppm (internal reference).

High-stability Frequency counter

The high-stability frequency counter module includes a frequency counter up to 50 MHz with a time base accuracy of 0.1 ppm (internal reference).

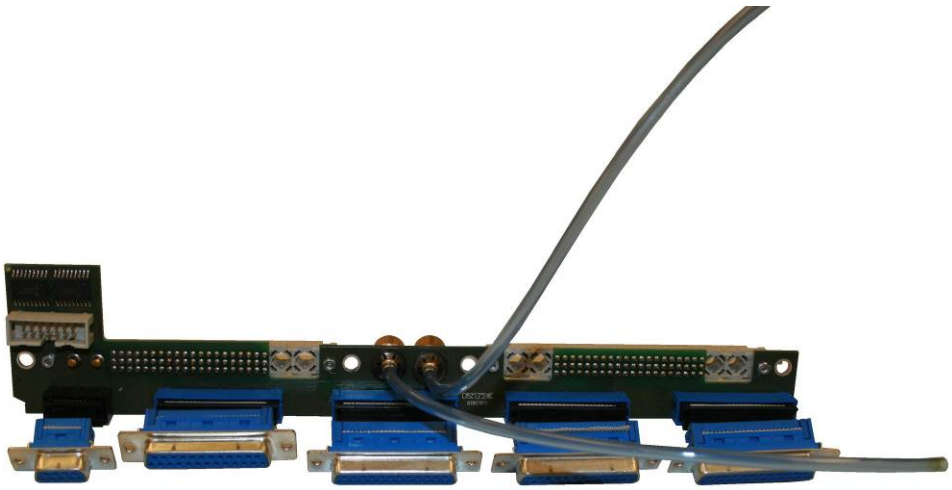
B. Quick-Swap Kit (QSK)

The Quick-Swap Kit (QSK) is a flexible solution for making replacement of fixtures easy and it is divided into two parts – a fixture bay part and a fixture part. Each of these is described in more detail in the following.

i. Fixture bay part

The fixture bay part QSK is a connector block for installation in the fixture bay (see figure).

Fixture bay part QSK



The QSK fixture bay package includes:

- QSK fixture bay PCB
- Spacers and screws (4 pcs.)

For installation instruction please refer to section ii on page 25.

ii. Fixture part

The fixture part QSK is a connector block for installation on the fixture (see figure). it is a PCB with two 60 + 4 Type M male connectors on the bottom side and 4 25-pins Type C male DSUB connectors (not mounted) along with one 14-pin male connector on the top side.

Fixture part QSK

QSK fixture part (bottom)

QSK fixture part (top)

The QSK fixture package includes:

- QSK fixture PCB
- Spacers and screws (4 pcs.)

For installation instruction please refer to section ii on page 25.

C. Fixture Kits

A wide range of fixture kits are available for the RTX2300. The fixture can be delivered both with and without a pneumatic slide. Furthermore, fixtures with different slide sizes are also available.

i. Standard Fixture Kit with pneumatic slide

The standard fixture kit with a pneumatic slide is available in three versions with different pneumatic slide sizes (Ø6, Ø8 or Ø12).

Standard fixture with pneumatic slide

ii. Standard Fixture Kit without pneumatic slide

The standard fixture plate is the same as the one with the pneumatic slide, and hence, the only difference is that the pneumatic slide is not included in this kit.

D. Connectivity options

Some connectivity options are available for the DUT. Two types of connectivity options are available – interfaces at the DUT and extra connectors (including added interface functionality at the DUT) on the rear panel of the RTX2300 unit.

i. DUT Interface

To provide flexible interfaces the RTX2300 supports addition of connectivity options to the DUT, hence providing a SPI or an I2C interface.

SPI interface to DUT

One of the internal RTX2300 USB interfaces can be converted to a SPI interface which can be routed to the DUT. The conversion will be made in a SPI interface module card, hence making use of the USB interface towards the module card to provide the SPI functionality.

I2C interface to DUT

One of the internal RTX2300 USB interfaces can be converted to a I2C interface which can be routed to the DUT. the conversion will be made in a I2C interface module card, hence making use of the USB interface towards the module card to provide the I2C functionality.

ii. Rear panel

An extra RJ45 connector can be added to the RTX2300 unit, hence enabling PSTN/FXO/ FXS testing options.

RJ45 connection

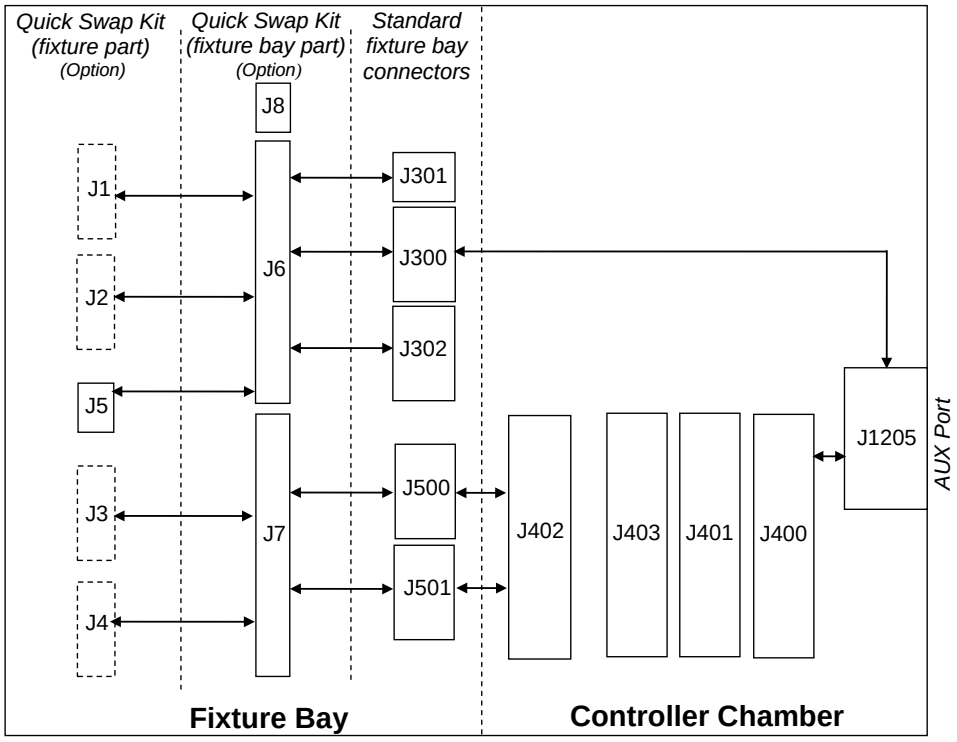
This option includes addition of an RJ45 connector in the empty slot on the rear panel of the RTX2300 unit. The connector is directly connected to the fixture bay.

5. Fixture Bay Connector Interfaces

The objective of this chapter is to provide an overview of both the available interfaces in the fixture bay and a list of pin functions in each connector. Throughout this user manual SCB and CCB have been mentioned but in the following these two interface blocks will be described in more detail.

As already mentioned a Quick-Swap Kit (QSK) for the fixture bay is available for the RTX2300. The QSK consists of both a fixture bay part and a fixture part, and hence, three different interface levels are available (as indicated in the figure below). Please note that the CCB interface is connected to the customization output connector in the controller chamber and the SCB interface is connected to the external AUX Port (providing an interface for external current measurements).

Overview of central connectors in the RTX2300 Smart ATE



As indicated the three interface levels are:

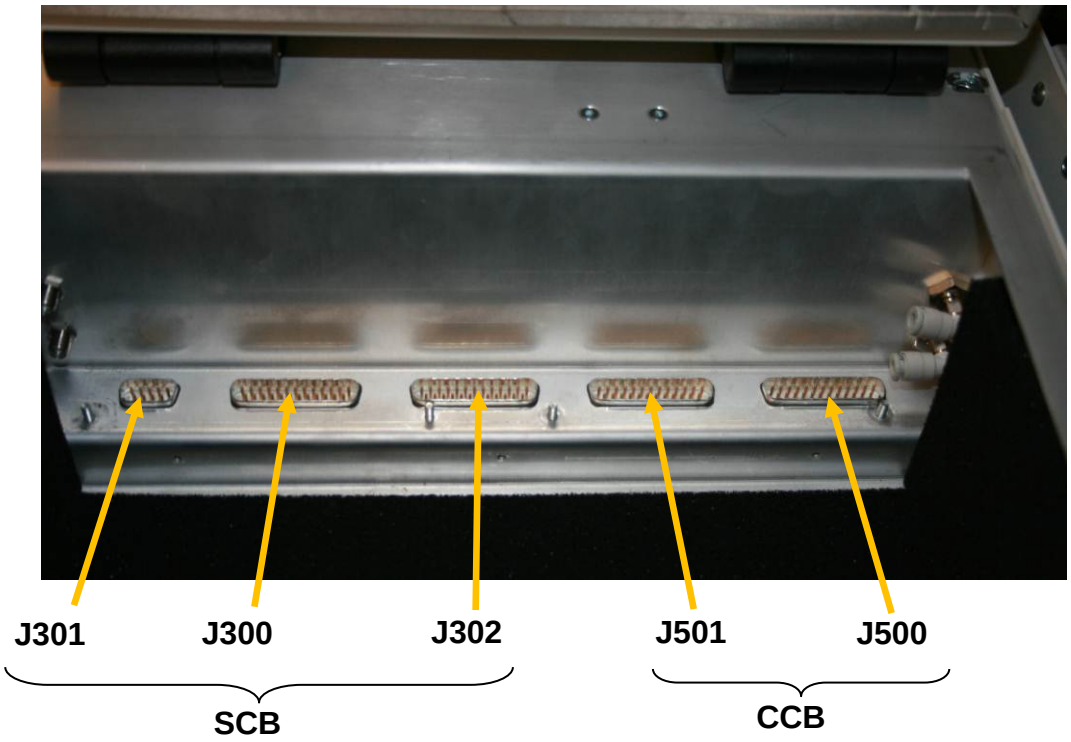
- Standard Fixture Bay Connectors
- Quick-Swap Kit Connectors (fixture bay part)
- Quick-Swap Kit Connectors (fixture part)

In the following these interfaces will be described in more detail, hence describing both the SCB and CCB interface connectors for each interface level. Furthermore, a complete SCB and CCB pin overview is given along with an overview of the electrical characteristics of the fixture bay interfaces.

A. **Standard Fixture Bay Connectors**

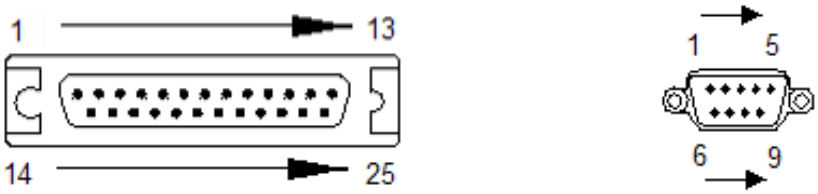
In the standard configuration the fixture bay connectors are placed in the bottom of the fixture bay. In total five connectors are available – four 25-pin male DSUB connectors and one 9-pin male DSUB connector (see figure below). The standard connectors are also referred to as connectors J301, J300, J302, J500 and J501. Please note that these connector names are used in the pin overview sections in this chapter.

Standard Fixture Bay interface connectors



The interface pin numbering for each of the connector types in the standard configuration is outlined in the figure below.

Standard interface connector types - 9-pin and 25-pin DSUB pin numbering



i. SCB

The Standard Connector Block (SCB) for the standard interface in the fixture bay is made up of the three connectors J301 (9-pin male DSUB), J300 (25-pin male DSUB) and J302 (25-pin male DSUB).

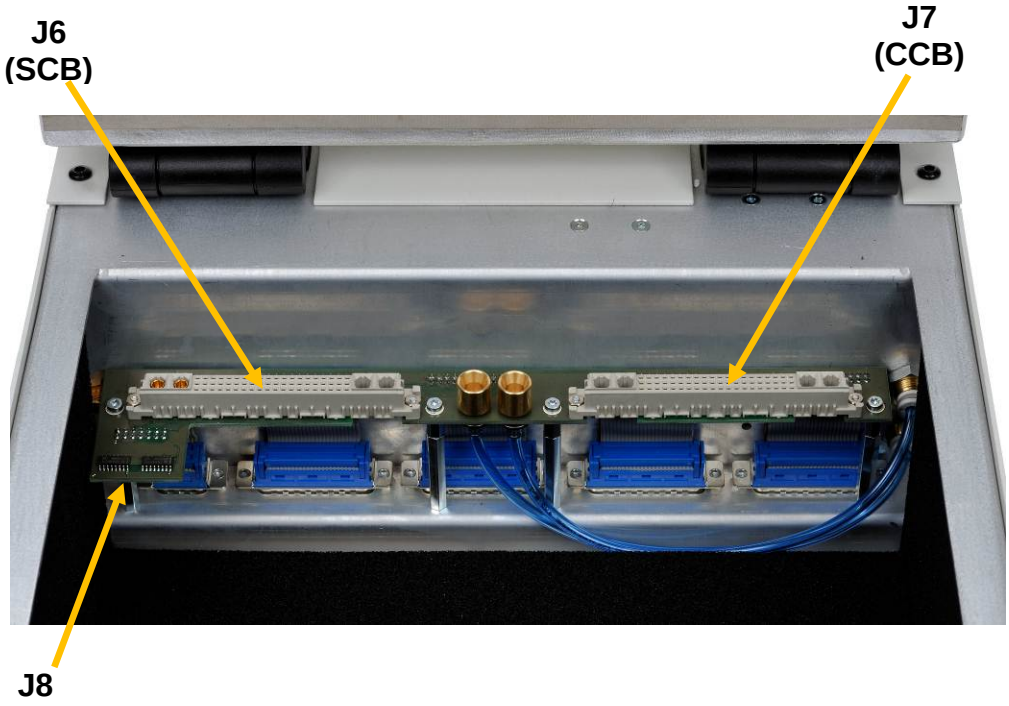
ii. CCB

The Customization Connector Block (CCB) for the standard interface in the fixture bay is made up of the two connectors J501 (25-pin male DSUB) and J500 (25-pin male DSUB).

B. Quick-Swap Kit Connectors – fixture bay part (Option)

The Quick-Swap Kit (QSK) provides easy access to the SCB and CCB interfaces. The main interface connectors are the two 60 + 4 Type M female connectors on the top side of the PCB. These are referred to as J6 and J7.

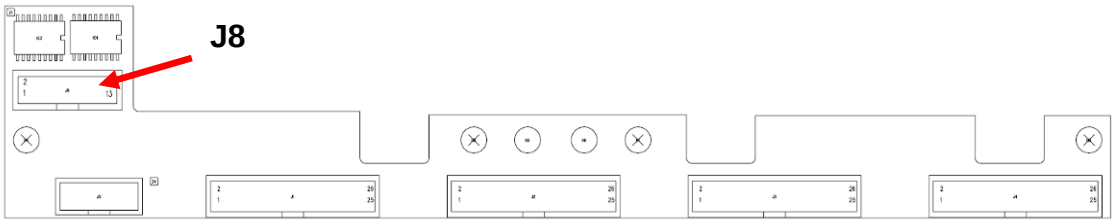
Fixture Bay Connectors with Quick-Swap Kit (fixture bay part) installed



Please note the J8 connector on the bottom side of the QSK PCB (used for RF Switch functionality) as outlined in the figure below. Please also note the pneumatic connectors in the centre of the QSK PCB.

Quick-Swap Kit Connector Interface – fixture bay part (bottom and top view)

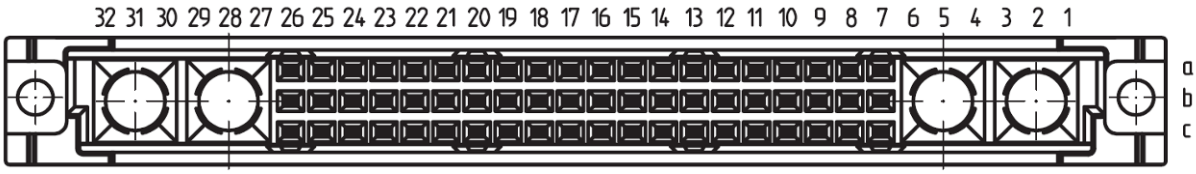
Top part of QSK PCB



Bottom part of QSK PCB

The interface pin numbering for the connector type used for the SCB and CCB in the QSK configuration is outlined in the figure below.

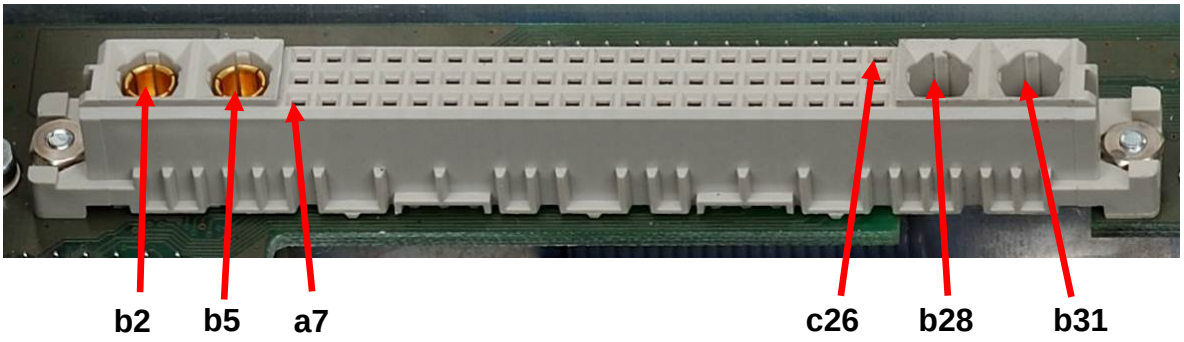
Quick-Swap Kit interface connector type – 60+4 Type M female pin numbering



i. SCB

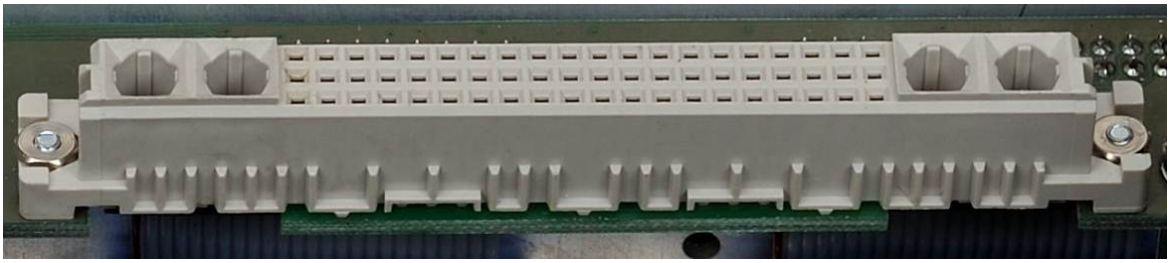
The SCB interface is provided through the J6 connector (see figure below). Please notice that the connector depicted is rotated 180 degrees in relation to the pin numbering figure above (i.e. pin B2 and B5 are to the left in the picture but to the right in the pin numbering figure above).

SCB connector on QSK (fixture bay part) – pin numbering



ii. CCB

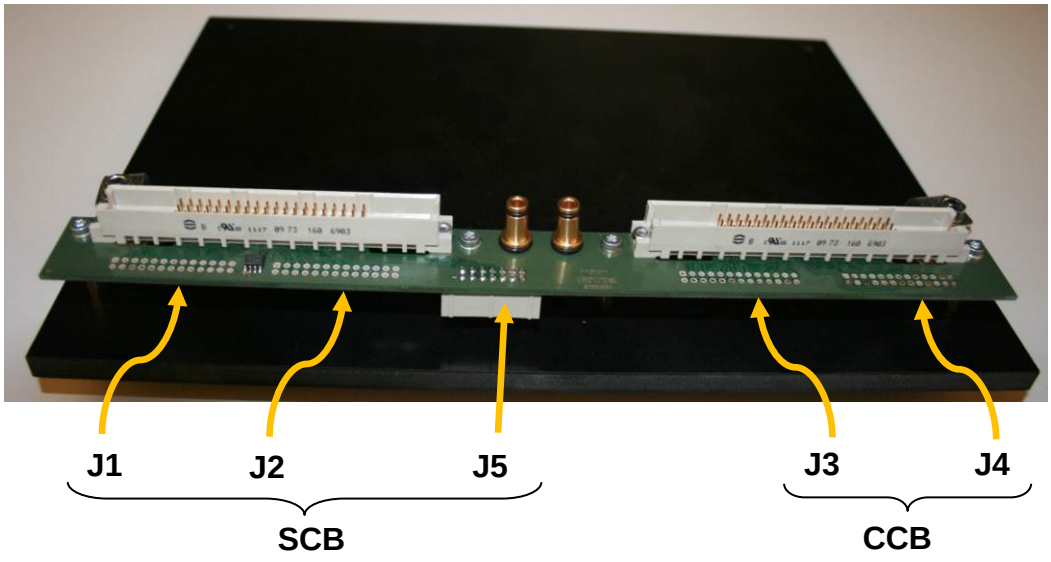
The CCB interface is provided through the J7 connector (see figure below). The pin numbering scheme from the SCB also applies to the CCB.



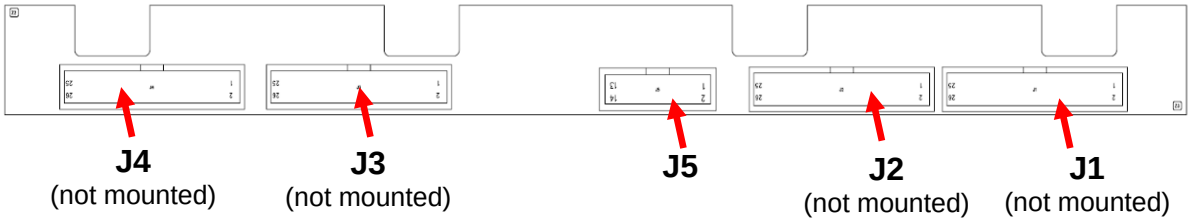
C. Quick-Swap Kit Connectors – fixture part (Option)

The fixture part of the QSK includes two 60 + 4 Type M male connectors for the SCB and CCB interfaces along with two pneumatic male connectors that fit on top of the fixture bay QSK (see figures below). On the top side of the QSK PCB five connectors provide access to the SCB and CCB interfaces. These connectors are referred to as J1, J2, J3, J4 and J5. Although only J5 has been mounted the pin numbering outlined in the figure below also applies to the contact numbering on the PCB.

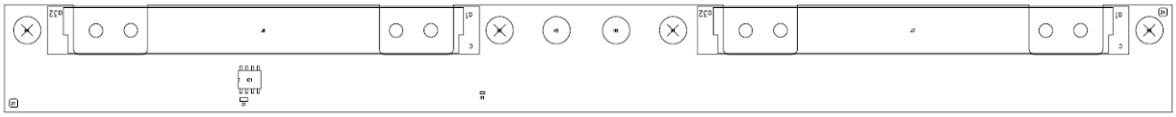
Fixture Connectors with Quick-Swap Kit (fixture part) installed



Quick-Swap Kit Connector Interface – fixture bay (bottom and top view)



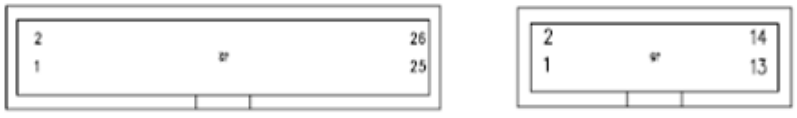
Top part of QSK PCB



Bottom part of QSK PCB

The interface pin numbering for the connector type used for the SCB and CCB in the QSK configuration is outlined in the figure below. It is emphasized here that only the 14-pin connector is mounted.

Quick-Swap Kit interface connector type – 25- and 14-pin numbering



25-pin connector **14-pin connector**

i. SCB

The SCB for the QSK interface on the fixture is made up of the three connectors J1 (25-pin connector – not mounted), J2 (25-pin connector – not mounted) and J5 (14-pin male connector).

ii. CCB

The CCB for the QSK interface on the fixture is made up of the two connectors J3 (25-pin connector – not mounted) and J4 (25-pin connector – not mounted).

D. SCB Interface Pin Overview

This section includes a pin overview of the SCB interface for all of the three interface levels (Standard, QSK fixture bay, QSK fixture). Please also notice the connection to J1205.

Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture bay (option)	Pin @ Quick-Swap Kit - fixture (option)	AUX-port (rear panel DSUB)	Signal Name	Input / Output	Description
J301-4-7	J6-b2			DUT_PSU_GND	-	GND for unit under test
J301-1-3	J6-b5			DUT_PSU_SUPPLY	O	Supply voltage for unit under test
J301-8	J6-a7	J2-9		INT_PSU_SENSE+	I	Positive sense input from test jig. If not used, it is recommended to connect to DUT_PSU_SUPPLY
J301-9	J6-b7	J2-11		INT_PSU_SENSE-	I	Negative sense input from test jig. If not used, it is recommended to connect to DUT_PSU_GND
J300-13	J6-c7	J1-25		FIX_CON_0	O	Control signal for fixture - High side current switch
J300-2	J6-a8	J1-3		ADC_IN_0	I	ADC input channel 0
J300-3	J6-b8	J1-5		ADC_IN_1	I	ADC input channel 1
J300-14	J6-c8	J1-2		FIX_CON_1	O	Control signal for fixture - High side current switch
J300-4	J6-a9	J1-7		ADC_IN_2	I	ADC input channel 2
J300-5	J6-b9	J1-9		ADC_IN_3	I	ADC input channel 3
J300-15	J6-c9	J1-4		FIX_CON_2	O	Control signal for fixture - High side current switch
J300-6	J6-a10	J1-11		ADC_IN_4	I	ADC input channel 4
J300-7	J6-b10	J1-13		ADC_IN_5	I	ADC input channel 5
		J1-1		GND	-	Ground signal
		J1-24		GND	-	Ground signal

Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture bay (option)	Pin @ Quick-Swap Kit - fixture (option)	AUX-port (rear panel DSUB)	Signal Name	Input / Output	Description
J300-16	J6-c10	J1-6		FIX_CON_3	O	Control signal for fixture - High side current switch
J300-8	J6-a11	J1-15		ADC_IN_6	I	ADC input channel 6
J300-9	J6-b11	J1-17		ADC_IN_7	I	ADC input channel 7
J300-18	J6-c11	J1-10		TONE_OUT_SCB	O	Analogue output from signal generator
J300-10	J6-a12	J1-19		ADC_COM	-	Common signal for ADC channels, connect to ground close top measurement point
J300-17	J6-b12	J1-8		DAC_1	O	DAC output channel 1
J300-19	J6-c12	J1-12		FILTER_IN_A	I	Analogue input for AC-measurements
J300-11	J6-a13	J1-21	J1205-1	I_MEAS_AC-	-	Signal feed to AUX-port. Can be used for external current measurements
J300-12	J6-b13	J1-23	J1205-2	I_MEAS_AC+	-	Signal feed to AUX-port. Can be used for external current measurements
J300-20	J6-c13	J1-14		FILTER_IN_B	I	Analogue input for AC-measurements
J300-22	J6-a14	J1-18		INT_UART_RX_SCB	I	UART for DUT (RX) - level is detected from RX signal
J300-23	J6-b14	J1-20		INT_UART_TX_SCB	O	UART for DUT (TX) - level is detected from RX signal
J300-21	J6-c14	J1-16		FILTER_OUT	O	Analogue output from signal generator
J300-24	J6-a15	J1-22		-12V	O	-12 V Supply voltage
J300-25	J6-b15	J2-1		GND	-	Ground signal
J300-1	J6-c15	J2-24		GND	-	Ground signal
J302-2	J6-a16	J2-3		+12V	O	+12 V Supply voltage
J302-3	J6-b16	J2-5		+12V	O	+12 V Supply voltage
J302-15	J6-c16	J2-4		EXT_SERIAL_RX_SCB	I	Serial communication connection from expansion slots (RX)
J302-4	J6-a17	J2-7		+5V	O	+5 V Supply voltage
J302-5	J6-b17			SCK	O	I2C clock
J302-16	J6-c17	J2-6		EXT_SERIAL_TX_SCB	O	Serial communication connection from expansion slots (TX)
J302-6	J6-a18			SDA	I/O	I2C data
J302-7	J6-b18	J2-13		SENSE_IN_3	I	Control signal from fixture
J302-17	J6-c18	J2-8		DAC_2	O	DAC output channel 2
J302-8	J6-a19	J2-15		SENSE_IN_2	I	Control signal from fixture
J302-9	J6-b19	J2-17		SENSE_IN_1	I	Control signal from fixture
J302-18	J6-c19	J2-10		SCB_RX_SCB	I	SPI communication from expansion slots

Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture bay (option)	Pin @ Quick-Swap Kit - fixture (option)	AUX-port (rear panel DSUB)	Signal Name	Input / Output	Description
J302-10	J6-a20	J2-19		SENSE_IN_0	I	Control signal from fixture
J302-11	J6-b20	J2-21		OE_SCB_EXP	O	Serial communication connection (4 Wire)
J302-19	J6-c20	J2-12		SCB_TX_SCB	O	SPI communication from expansion slots
J302-12	J6-a21	J2-23		STROBE_SCB_EXP	O	Serial communication connection (4 Wire)
J302-13	J6-b21	J2-25		DATA_SCB_EXP	O	Serial communication connection (4 Wire)
J302-20	J6-c21	J2-15		SCB_SPI_CLK	O	SPI communication from expansion slots
J302-14	J6-a22	J2-2		CLK_SCB_EXP	O	Serial communication connection (4 Wire)
J302-24	J6-b22	J2-22		V_SCB_USB5	O	USB Supply
J302-21	J6-c22	J2-16		SCB_SPI_/CS	O	SPI communication from expansion slots
J302-22	J6-a23	J2-18		SCB_USBDM_5	I/O	USB -
J302-23	J6-b23	J2-20		SCB_USBDP_5	I/O	USB +
	J6-c23 (J8-9)	J5-9		RF_SWITCH_01	O	Control signal for RF switch, open collector output
	J6-a24 (J8-8)	J5-8		RF_SWITCH_02	O	Control signal for RF switch, open collector output
	J6-b24 (J8-10)	J5-10		RF_SWITCH_03	O	Control signal for RF switch, open collector output
	J6-c24 (J8-7)	J5-7		RF_SWITCH_04	O	Control signal for RF switch, open collector output
	J6-a25 (J8-11)	J5-11		RF_SWITCH_05	O	Control signal for RF switch, open collector output
	J6-b25 (J8-6)	J5-6		RF_SWITCH_06	O	Control signal for RF switch, open collector output
	J6-c25 (J8-4)	J5-4		RF_SWITCH_07	O	Control signal for RF switch, open collector output
	J6-a26 (J8-3)	J5-3		RF_SWITCH_08	O	Control signal for RF switch, open collector output
	J6-b26 (J8-13)	J5-13		RF_SWITCH_09	O	Control signal for RF switch, open collector output
	J6-c26 (J8-14)	J5-14		RF_SWITCH_10	O	Control signal for RF switch, open collector output
	(J8-1)	J5-1		NC		
	(J8-2)	J5-2		GND		

Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture bay (option)	Pin @ Quick-Swap Kit - fixture (option)	AUX-port (rear panel DSUB)	Signal Name	Input / Output	Description
	(J8-5)	J5-5		NC		
	(J8-12)	J5-12		+12 V	O	

E. **CCB Interface Pin Overview**

This section includes a pin overview of the CCB interface for all of the three interface levels (Standard, QSK fixture bay and QSK fixture).

Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture bay (option)	Pin @ Quick-Swap Kit - fixture (option)	Customization output connector	Signal Name	Input / Output	Description
J501-3	J7-a7	J3-5		OE_RF_SWITCH	O	Serial connection to the RF switch
J501-4	J7-b7	J3-7		CLK_RF_SWITCH	O	Serial connection to the RF switch
J501-2	J7-c7	J3-3		DATA_RF_SWITCH	O	Serial connection to the RF switch
	J7-a8					
	J7-b8					
J501-5	J7-c8	J3-9		STROBE_RF_SWITCH	O	Serial connection to the RF switch
J501-6	J7-a9	J3-11	J402-a24	CBB45	I/O	Configurable through customization interface
J501-7	J7-b9	J3-13	J402-c23	CBB44	I/O	Configurable through customization interface
J501-8	J7-c9	J3-15	J402-a23	CBB43	I/O	Configurable through customization interface
J501-9	J7-a10	J3-17	J402-c22	CBB42	I/O	Configurable through customization interface
J501-10	J7-b10	J3-19	J402-a22	CBB41	I/O	Configurable through customization interface

Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture bay (option)	Pin @ Quick-Swap Kit - fixture (option)	Customization output connector	Signal Name	Input / Output	Description
J501-11	J7-c10	J3-21	J402-c21	CBB40	I/O	Configurable through customization interface
J501-12	J7-a11	J3-23	J402-a21	CBB39	I/O	Configurable through customization interface
J501-13	J7-b11	J3-25	J402-c20	CBB38	I/O	Configurable through customization interface
J501-14	J7-c11	J3-2	J402-a20	CBB37	I/O	Configurable through customization interface
J501-15	J7-a12	J3-4	J402-c19	CBB36	I/O	Configurable through customization interface
J501-16	J7-b12	J3-6	J402-a19	CBB35	I/O	Configurable through customization interface
J501-17	J7-c12	J3-8	J402-c18	CBB34	I/O	Configurable through customization interface
J501-18	J7-a13	J3-10	J402-a18	CBB33	I/O	Configurable through customization interface
J501-19	J7-b13	J3-12	J402-c17	CBB32	I/O	Configurable through customization interface
J501-20	J7-c13	J3-14	J402-a17	CBB31	I/O	Configurable through customization interface
J501-21	J7-a14	J3-16	J402-c16	CBB30	I/O	Configurable through customization interface
J501-22	J7-b14	J3-18	J402-a16	CBB29	I/O	Configurable through customization interface
J501-23	J7-c14	J3-20	J402-c15	CBB28	I/O	Configurable through customization interface
J501-24	J7-a15	J3-22	J402-a15	CBB27	I/O	Configurable through customization interface
J501-25	J7-b15	J3-24	J402-c14	CBB26	I/O	Configurable through customization interface
J500-1	J7-c15	J4-1	J402-a14	CBB25	I/O	Configurable through customization interface
J500-2	J7-a16	J4-3	J402-c13	CBB24	I/O	Configurable through customization interface
J500-3	J7-b16	J4-5	J402-a13	CBB23	I/O	Configurable through customization interface
J500-4	J7-c16	J4-7	J402-c12	CBB22	I/O	Configurable through customization interface
J500-5	J7-a17	J4-9	J402-a12	CBB21	I/O	Configurable through customization interface

Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture bay (option)	Pin @ Quick-Swap Kit - fixture (option)	Customization output connector	Signal Name	Input / Output	Description
J500-6	J7-b17	J4-11	J402-c11	CBB20	I/O	Configurable through customization interface
J500-7	J7-c17	J4-13	J402-a11	CBB19	I/O	Configurable through customization interface
J500-8	J7-a18	J4-15	J402-c10	CBB18	I/O	Configurable through customization interface
J500-9	J7-b18	J4-17	J402-a10	CBB17	I/O	Configurable through customization interface
J500-10	J7-c18	J4-19	J402-c9	CBB16	I/O	Configurable through customization interface
J500-11	J7-a19	J4-21	J402-a9	CBB15	I/O	Configurable through customization interface
J500-12	J7-b19	J4-23	J402-c8	CBB14	I/O	Configurable through customization interface
J500-13	J7-c19	J4-25	J402-a8	CBB13	I/O	Configurable through customization interface
J500-14	J7-a20	J4-2	J402-c7	CBB12	I/O	Configurable through customization interface
J500-15	J7-b20	J4-4	J402-a7	CBB11	I/O	Configurable through customization interface
J500-16	J7-c20	J4-6	J402-c6	CBB10	I/O	Configurable through customization interface
J500-17	J7-a21	J4-8	J402-a6	CBB9	I/O	Configurable through customization interface
J500-18	J7-b21	J4-10	J402-c5	CBB8	I/O	Configurable through customization interface
J500-19	J7-c21	J4-12	J402-a5	CBB7	I/O	Configurable through customization interface
J500-20	J7-a22	J4-14	J402-c4	CBB6	I/O	Configurable through customization interface
J500-21	J7-b22	J4-16	J402-a4	CBB5	I/O	Configurable through customization interface
J500-22	J7-c22	J4-18	J402-c3	CBB4	I/O	Configurable through customization interface
J500-23	J7-a23	J4-20	J402-a3	CBB3	I/O	Configurable through customization interface
J500-24	J7-b23	J4-22	J402-c2	CBB2	I/O	Configurable through customization interface
J500-25	J7-c23	J4-24	J402-a2	CBB1	I/O	Configurable through customization interface

F. **Electrical Characteristics for Fixture Interfaces**

In this section some electrical characteristics for fixture interfaces (specifically J300, J302, J501 and J5) are outlined below.

Signal Name	Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture (option)	Vmin	vmax	Vin(H) max	Vin(H) min	Vin(L) max	Vin(L) min	Vo(H) max	Vo(H) min	Vo(L) max	Iout max
DUT_PSU_SUPPLY	J301-4-7		0V	15V								1A
FIX_CON_0	J300-13									10V		50mA
FIX_CON_1	J300-14									10V		50mA
FIX_CON_2	J300-15									10V		50mA
FIX_CON_3	J300-16									10V		50mA
ADC_IN_0	J300-2				12V			-12V				
ADC_IN_1	J300-3				12V			-12V				
ADC_IN_2	J300-4				12V			-12V				
ADC_IN_3	J300-5				12V			-12V				
ADC_IN_4	J300-6				12V			-12V				
ADC_IN_5	J300-7				12V			-12V				
ADC_IN_6	J300-8				12V			-12V				
ADC_IN_7	J300-9				12V			-12V				
DAC_1	J300-17								8V	-8V		
DAC_2	J302-17								8V	-8V		
INT_UART_RX_SCB	J300-22				5V	1.8V	0.6V	0V				
INT_UART_TX_SCB	J300-23									RX-150mV		
-12V	J300-24											100mA
+12V	J302-2											300mA
+5V	J302-4											300mA
SCK	J302-5				5V				5V			
SDA	J302-6				5V				5V			
SENSE_IN_3	J302-7				5V	3.5V		1.5V				
SENSE_IN_2	J302-8				5V	3.5V		1.5V				
SENSE_IN_1	J302-9				5V	3.5V		1.5V				
SENSE_IN_0	J302-10				5V	3.5V		1.5V				
V_SCB_USB5	J302-24			5V								100mA
OE_SCB_EXP	J302-11									3.1V	0.2V	
STROBE_SCB_EXP	J302-12									3.1V	0.2V	
DATA_SCB_EXP	J302-13									3.1V	0.2V	
CLK_SCB_EXP	J302-14									3.1V	0.2V	

Signal Name	Pin @ Standard Fixture Bay Interface	Pin @ Quick-Swap Kit - fixture (option)	Vmin	vmax	Vin(H) max	Vin(H) min	Vin(L) max	Vin(L) min	Vo(H) max	Vo(H) min	Vo(L) max	Iout max
OE_RF_SWITCH	J501-3									3.1V	0.2V	
CLK_RF_SWITCH	J501-4									3.1V	0.2V	
DATA_RF_SWITCH	J501-2									3.1V	0.2V	
STROBE_RF_SWITCH	J501-5									3.1V	0.2V	
RF_SWITCH_01		J5-9									1.1V	100mA
RF_SWITCH_02		J5-8									1.1V	100mA
RF_SWITCH_03		J5-10									1.1V	100mA
RF_SWITCH_04		J5-7									1.1V	100mA
RF_SWITCH_05		J5-11									1.1V	100mA
RF_SWITCH_06		J5-6									1.1V	100mA
RF_SWITCH_07		J5-4									1.1V	100mA
RF_SWITCH_08		J5-3									1.1V	100mA
RF_SWITCH_09		J5-13									1.1V	100mA
RF_SWITCH_10		J5-14									1.1V	100mA

6. RTX2300 Customization

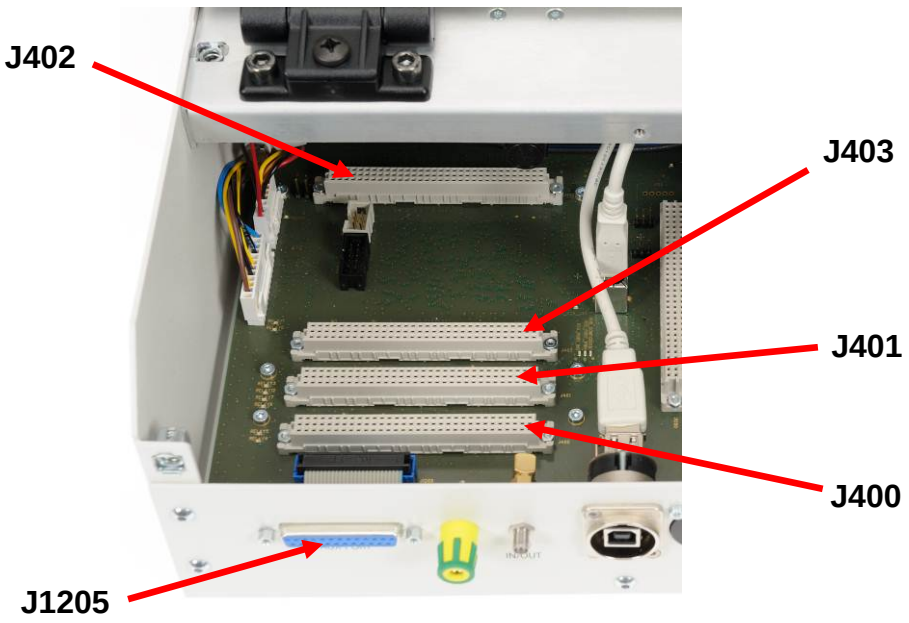
The RTX2300 Smart ATE provides a flexible customization interface which is accessible through four 2 x 32 pin connectors in the controller chamber. Furthermore, the RTX2300 supports connection of external instrumentation through the external AUX-port on the rear panel of the RTX2300 unit. Routing of signals from this port can be customized through the customization interface. A detailed overview of the pin configuration of each of the customization connectors and the external AUX-port are given in this chapter along with detailed instructions on how to create a customization board for the RTX2300 Smart ATE unit.

The RTX2300 Smart ATE unit also provides a very flexible approach to customization of the fixture. In chapter 5 all the fixture chamber interface variants were described in detail but the fixture itself can also be customized. The last part of this chapter will, hence, outline instructions on how to customize fixtures.

A. Controller Chamber Customization Interface

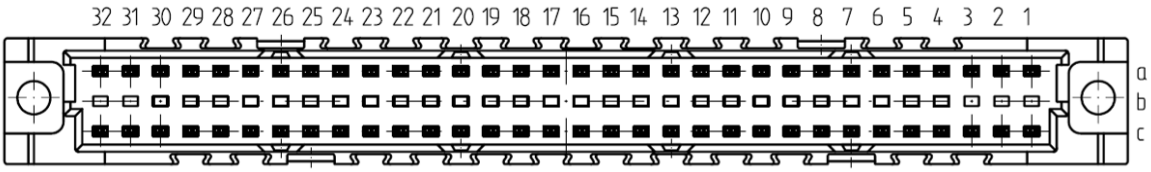
The customization interface is in the controller chamber and it is (as mentioned above) made up of four 2 x 32 pins DIN 41612 Type C female connectors on the RTX2300 main board (see figure below). One of these is the output connector (which is connected to the fixture bay interface) while the other three are input connectors. One of the input connectors (J400) is connected to the external AUX-port. The customization interface is referred to as connectors J400, J401, J402 and J403 along with the external AUX-port which is referred to as J1205.

Controller Chamber Customization Interface

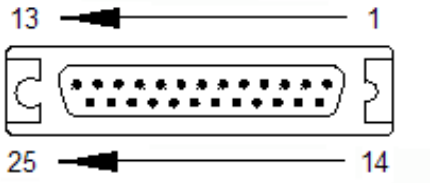


An overview of the pin numbering for the customization interfaces and the AUX-port is included below.

Customization Connectors (2 x 32 pin DIN 41612 Type C fem.) – pin numbers



AUX-port (25-pin female DSUB) – pin numbering



B. Customization Interface Pin Overview

In the following paragraphs a detailed pin overview is given for each of the interfaces outlined in the figure above (i.e. J402, J400, J401, J403 and J1205).

i. Customization Output Connector Pin Overview (J402)

This paragraph outlines the detailed pin overview of the J402 output connector. Please note that this connector is connected to the CCB interface in the fixture bay (i.e. the connectors J500 and J501).

Customization output connector	Pin @ Standard Fixture Bay Interface	Signal Name	Input / Output	Description
J402-a1		GND	-	
J402-c1		GND	-	
J402-a2	J500-25	CBB1	I/O	Configurable through customization interface
J402-c2	J500-24	CBB2	I/O	Configurable through customization interface
J402-a3	J500-23	CBB3	I/O	Configurable through customization interface
J402-c3	J500-22	CBB4	I/O	Configurable through customization interface
J402-a4	J500-21	CBB5	I/O	Configurable through customization interface
J402-c4	J500-20	CBB6	I/O	Configurable through customization interface
J402-a5	J500-19	CBB7	I/O	Configurable through customization interface
J402-c5	J500-18	CBB8	I/O	Configurable through customization interface
J402-a6	J500-17	CBB9	I/O	Configurable through customization interface
J402-c6	J500-16	CBB10	I/O	Configurable through customization interface
J402-a7	J500-15	CBB11	I/O	Configurable through customization interface
J402-c7	J500-14	CBB12	I/O	Configurable through customization interface
J402-a8	J500-13	CBB13	I/O	Configurable through customization interface
J402-c8	J500-12	CBB14	I/O	Configurable through customization interface
J402-a9	J500-11	CBB15	I/O	Configurable through customization interface
J402-c9	J500-10	CBB16	I/O	Configurable through customization interface
J402-a10	J500-9	CBB17	I/O	Configurable through customization interface
J402-c10	J500-8	CBB18	I/O	Configurable through customization interface
J402-a11	J500-7	CBB19	I/O	Configurable through customization interface
J402-c11	J500-6	CBB20	I/O	Configurable through customization interface
J402-a12	J500-5	CBB21	I/O	Configurable through customization interface
J402-c12	J500-4	CBB22	I/O	Configurable through customization interface
J402-a13	J500-3	CBB23	I/O	Configurable through customization interface
J402-c13	J500-2	CBB24	I/O	Configurable through customization interface
J402-a14	J500-1	CBB25	I/O	Configurable through customization interface
J402-c14	J501-25	CBB26	I/O	Configurable through customization interface
J402-a15	J501-24	CBB27	I/O	Configurable through customization interface
J402-c15	J501-23	CBB28	I/O	Configurable through customization interface
J402-a16	J501-22	CBB29	I/O	Configurable through customization interface
J402-c16	J501-21	CBB30	I/O	Configurable through customization interface
J402-a17	J501-20	CBB31	I/O	Configurable through customization interface
J402-c17	J501-19	CBB32	I/O	Configurable through customization interface

Customization output connector	Pin @ Standard Fixture Bay Interface	Signal Name	Input / Output	Description
J402-a18	J501-18	CBB33	I/O	Configurable through customization interface
J402-c18	J501-17	CBB34	I/O	Configurable through customization interface
J402-a19	J501-16	CBB35	I/O	Configurable through customization interface
J402-c19	J501-15	CBB36	I/O	Configurable through customization interface
J402-a20	J501-14	CBB37	I/O	Configurable through customization interface
J402-c20	J501-13	CBB38	I/O	Configurable through customization interface
J402-a21	J501-12	CBB39	I/O	Configurable through customization interface
J402-c21	J501-11	CBB40	I/O	Configurable through customization interface
J402-a22	J501-10	CBB41	I/O	Configurable through customization interface
J402-c22	J501-9	CBB42	I/O	Configurable through customization interface
J402-a23	J501-8	CBB43	I/O	Configurable through customization interface
J402-c23	J501-7	CBB44	I/O	Configurable through customization interface
J402-a24	J501-6	CBB45	I/O	Configurable through customization interface
J402-c24		-	-	
J402-a25		-	-	
J402-c25		-	-	
J402-a26		-	-	
J402-c26		-	-	
J402-a27		-	-	
J402-c27		+12V	-	
J402-a28		-	-	
J402-c28		+12V	-	
J402-a29		-	-	
J402-c29		-	-	
J402-a30		+5V	-	
J402-c30		-12V	-	
J402-a31		+5V	-	
J402-c31		+12V	-	
J402-a32		GND	-	
J402-c32		GND	-	

ii. Customization Input Connector Pin Overview (J400)

This paragraph outlines the detailed pin overview of the J400 input connector. Please note that this connector is connected to the external AUX-port (J1205).

Customization input connector	**AUX-port** (rear panel DSUB)	**Signal Name**	**Input / Output**	**Description**
J400-a1		GND	-	
J400-c1	J1205-25	GND	-	Can be used for external instrumentation
J400-a2	J1205-14	AQUI_5_B	-	Can be used for external instrumentation
J400-c2	J1205-24	AQUI_10_B	-	Can be used for external instrumentation
J400-a3	J1205-13	AQUI_5_A	-	Can be used for external instrumentation
J400-c3	J1205-23	AQUI_10_A	-	Can be used for external instrumentation
J400-a4	J1205-12	AQUI_4_B	-	Can be used for external instrumentation
J400-c4	J1205-22	AQUI_9_B	-	Can be used for external instrumentation
J400-a5	J1205-11	AQUI_4_A	-	Can be used for external instrumentation
J400-c5	J1205-21	AQUI_9_A	-	Can be used for external instrumentation
J400-a6	J1205-10	AQUI_3_B	-	Can be used for external instrumentation
J400-c6	J1205-20	AQUI_8_B	-	Can be used for external instrumentation
J400-a7	J1205-9	AQUI_3_A	-	Can be used for external instrumentation
J400-c7	J1205-19	AQUI_8_A	-	Can be used for external instrumentation
J400-a8	J1205-8	AQUI_2_B	-	Can be used for external instrumentation
J400-c8	J1205-18	AQUI_7_B	-	Can be used for external instrumentation
J400-a9	J1205-7	AQUI_2_A	-	Can be used for external instrumentation
J400-c9	J1205-17	AQUI_7_A	-	Can be used for external instrumentation
J400-a10	J1205-6	AQUI_1_B	-	Can be used for external instrumentation
J400-c10	J1205-16	AQUI_6_B	-	Can be used for external instrumentation
J400-a11	J1205-5	AQUI_1_A	-	Can be used for external instrumentation
J400-c11	J1205-15	AQUI_6_A	-	Can be used for external instrumentation
J400-a12		NC	-	Can be used for external instrumentation
J400-c12		Audio_Buf_2_In+	-	
J400-a13		+5V	-	
J400-c13		Audio_Buf_2_In-	-	
J400-a14		PWM0	-	
J400-c14		Audio_Buf_2_Out	-	
J400-a15		GND	-	
J400-c15		GND	-	
J400-a16		EXP_Slot4_0	-	
J400-c16		EXP_Slot3_0	-	
J400-a17		EXP_Slot4_1	-	
J400-c17		EXP_Slot3_1	-	
J400-a18		EXP_Slot4_2	-	
J400-c18		EXP_Slot3_2	-	
J400-a19		EXP_Slot4_3	-	

Customization input connector	AUX-port (rear panel DSUB)	Signal Name	Input / Output	Description
J400-c19		EXP_Slot3_3	-	
J400-a20		EXP_Slot4_4	-	
J400-c20		EXP_Slot3_4	-	
J400-a21		EXP_Slot4_5	-	
J400-c21		EXP_Slot3_5	-	
J400-a22		EXP_Slot4_6	-	
J400-c22		EXP_Slot3_6	-	
J400-a23		EXP_Slot4_7	-	
J400-c23		EXP_Slot3_7	-	
J400-a24		EXP_Slot4_8	-	
J400-c24		EXP_Slot3_8	-	
J400-a25		EXP_Slot4_9	-	
J400-c25		EXP_Slot3_9	-	
J400-a26		EXP_Slot4_10	-	
J400-c26		EXP_Slot3_10	-	
J400-a27		EXP_Slot4_11	-	
J400-c27		EXP_Slot3_11	-	
J400-a28		EXP_Slot4_12	-	
J400-c28		EXP_Slot3_12	-	
J400-a29		EXP_Slot4_13	-	
J400-c29		EXP_Slot3_13	-	
J400-a30		EXP_Slot4_14	-	
J400-c30		EXP_Slot3_14	-	
J400-a31		EXP_Slot4_15	-	
J400-c31		EXP_Slot3_15	-	
J400-a32		GND	-	
J400-c32		GND	-	

iii. Customization Input Connector Pin Overview (J401)

This paragraph outlines the detailed pin overview of the J401 input connector.

Customization input connector	Signal Name	Input / Output	Description
J401-a1	GND	-	
J401-c1	GND	-	
J401-a2	REL_4_A_OUT	-	Relay normal open pin
J401-c2	REL_6_A_OUT	-	Relay normal open pin
J401-a3	REL_4_A_COM	-	Relay common pin
J401-c3	REL_6_A_COM	-	Relay common pin
J401-a4	REL_4_A_OUT/	-	Relay normal closed
J401-c4	REL_6_A_OUT/	-	Relay normal closed
J401-a5	REL_4_B_OUT	-	Relay normal open pin
J401-c5	REL_6_B_OUT	-	Relay normal open pin
J401-a6	REL_4_B_COM	-	Relay common pin
J401-c6	REL_6_B_COM	-	Relay common pin
J401-a7	REL_4_B_OUT/	-	Relay normal closed
J401-c7	REL_6_B_OUT/	-	Relay normal closed
J401-a8	REL_5_A_OUT	-	Relay normal open pin
J401-c8	REL_7_A_OUT	-	Relay normal open pin
J401-a9	REL_5_A_COM	-	Relay common pin
J401-c9	REL_7_A_COM	-	Relay common pin
J401-a10	REL_5_A_OUT/	-	Relay normal closed
J401-c10	REL_7_A_OUT/	-	Relay normal closed
J401-a11	REL_5_B_OUT	-	Relay normal open pin
J401-c11	REL_7_B_OUT	-	Relay normal open pin
J401-a12	REL_5_B_COM	-	Relay common pin
J401-c12	REL_7_B_COM	-	Relay common pin
J401-a13	REL_5_B_OUT/	-	Relay normal closed
J401-c13	REL_7_B_OUT/	-	Relay normal closed
J401-a14	ATM_RXD	-	
J401-c14	ATM_TXD	-	
J401-a15	Audio_A_OUT	-	
J401-c15	Audio_B_OUT	-	
J401-a16	EXP_Slot2_0	-	
J401-c16	EXP_Slot1_0	-	
J401-a17	EXP_Slot2_1	-	
J401-c17	EXP_Slot1_1	-	
J401-a18	EXP_Slot2_2	-	
J401-c18	EXP_Slot1_2	-	
J401-a19	EXP_Slot2_3	-	

Customization input connector	Signal Name	Input / Output	Description
J401-c19	EXP_Slot1_3	-	
J401-a20	EXP_Slot2_4	-	
J401-c20	EXP_Slot1_4	-	
J401-a21	EXP_Slot2_5	-	
J401-c21	EXP_Slot1_5	-	
J401-a22	EXP_Slot2_6	-	
J401-c22	EXP_Slot1_6	-	
J401-a23	EXP_Slot2_7	-	
J401-c23	EXP_Slot1_7	-	
J401-a24	EXP_Slot2_8	-	
J401-c24	EXP_Slot1_8	-	
J401-a25	EXP_Slot2_9	-	
J401-c25	EXP_Slot1_9	-	
J401-a26	EXP_Slot2_10	-	
J401-c26	EXP_Slot1_10	-	
J401-a27	EXP_Slot2_11	-	
J401-c27	EXP_Slot1_11	-	
J401-a28	EXP_Slot2_12	-	
J401-c28	EXP_Slot1_12	-	
J401-a29	EXP_Slot2_13	-	
J401-c29	EXP_Slot1_13	-	
J401-a30	EXP_Slot2_14	-	
J401-c30	EXP_Slot1_14	-	
J401-a31	EXP_Slot2_15	-	
J401-c31	EXP_Slot1_15	-	
J401-a32	GND	-	
J401-c32	GND	-	

iv. Customization Input Connector Pin Overview (J403)

This paragraph outlines the detailed pin overview of the J403 input connector.

Customization intput connector	Signal Name	Input / Output	Description
J403-a1	GND	-	
J403-c1	GND	-	
J403-a2	V_CCB_USB6	-	
J403-c2	REL_0_A_OUT	-	Relay normal open pin
J403-a3	CCB_USBDM_6	-	
J403-c3	REL_0_A_COM	-	Relay common pin
J403-a4	CCB_USBDP_6	-	
J403-c4	REL_0_A_OUT/	-	Relay normal closed
J403-a5	GND	-	
J403-c5	REL_0_B_OUT	-	Relay normal open pin
J403-a6	CCB_SINK_OUT0	-	
J403-c6	REL_0_B_COM	-	Relay common pin
J403-a7	CCB_SINK_OUT1	-	
J403-c7	REL_0_B_OUT/	-	Relay normal closed
J403-a8	CCB_SINK_OUT2	-	
J403-c8	REL_1_A_OUT	-	Relay normal open pin
J403-a9	CCB_SINK_OUT3	-	
J403-c9	REL_1_A_COM	-	Relay common pin
J403-a10	CCB_SINK_OUT4	-	
J403-c10	REL_1_A_OUT/	-	Relay normal closed
J403-a11	CCB_SINK_OUT5	-	
J403-c11	REL_1_B_OUT	-	Relay normal open pin
J403-a12	CCB_SINK_OUT6	-	
J403-c12	REL_1_B_COM	-	Relay common pin
J403-a13	CCB_SINK_OUT7	-	
J403-c13	REL_1_B_OUT/	-	Relay normal closed
J403-a14	VREF	-	
J403-c14	REL_2_A_OUT	-	Relay normal open pin
J403-a15	CCB_SOURCE_OUT0	-	
J403-c15	REL_2_A_COM	-	Relay common pin
J403-a16	CCB_SOURCE_OUT1	-	
J403-c16	REL_2_A_OUT/	-	Relay normal closed
J403-a17	CCB_SOURCE_OUT2	-	
J403-c17	REL_2_B_OUT	-	Relay normal open pin
J403-a18	CCB_SOURCE_OUT3	-	
J403-c18	REL_2_B_COM	-	Relay common pin
J403-a19	CCB_SOURCE_OUT4	-	

Customization output connector	Signal Name	Input / Output	Description
J403-c19	REL_2_B_OUT/	-	Relay normal closed
J403-a20	CCB_SOURCE_OUT5	-	
J403-c20	REL_3_A_OUT	-	Relay normal open pin
J403-a21	CCB_SOURCE_OUT6	-	
J403-c21	REL_3_A_COM	-	Relay common pin
J403-a22	CCB_SOURCE_OUT7	-	
J403-c22	REL_3_A_OUT/	-	Relay normal closed
J403-a23	GND	-	
J403-c23	REL_3_B_OUT	-	Relay normal open pin
J403-a24	DIG_IN_0	-	
J403-c24	REL_3_B_COM	-	Relay common pin
J403-a25	DIG_IN_1	-	
J403-c25	REL_3_B_OUT/	-	Relay normal closed
J403-a26	DIG_IN_2	-	
J403-c26	Audio_Buf_1_IN+	-	
J403-a27	DIG_IN_3	-	
J403-c27	Audio_Buf_1_IN-	-	
J403-a28	DIG_IN_4	-	
J403-c28	Audio_Buf_1_OUT	-	
J403-a29	DIG_IN_5	-	
J403-c29	GND	-	
J403-a30	DIG_IN_6	-	
J403-c30	-	-	
J403-a31	DIG_IN_7	-	
J403-c31	GND	-	
J403-a32	GND	-	
J403-c32	GND	-	

v. External AUX Interface Connector (J1205)

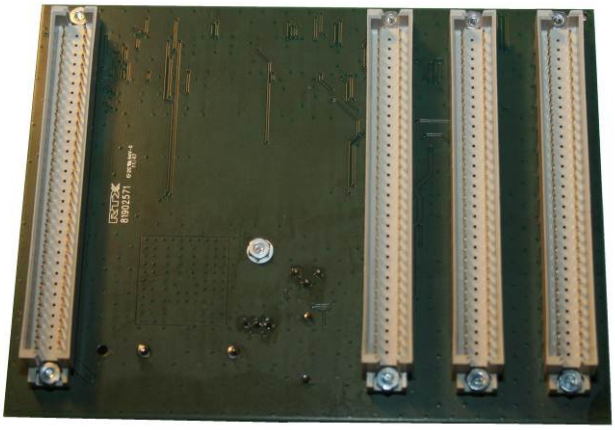
This paragraph outlines the detailed pin overview of the J1205 AUX-port connector.

AUX-port (rear panel DSUB)	**Customization output connector**	**Pin @ Standard Fixture Bay Interface**	**Signal Name**	**Input / Output**	**Description**
J1205-1		J300-11	I_MEAS_AC-	-	Signal feed to AUX-port. Can be used for external current measurements
J1205-2		J300-12	I_MEAS_AC+	-	Signal feed to AUX-port. Can be used for external current measurements
J1205-3			ADC_1_AQUI	-	Can be used for external instrumentation
J1205-4			ADC_COM	-	Can be used for external instrumentation
J1205-5	J400-a11		AQUI_1_A	-	Can be used for external instrumentation
J1205-6	J400-a10		AQUI_1_B	-	Can be used for external instrumentation
J1205-7	J400-a9		AQUI_2_A	-	Can be used for external instrumentation
J1205-8	J400-a8		AQUI_2_B	-	Can be used for external instrumentation
J1205-9	J400-a7		AQUI_3_A	-	Can be used for external instrumentation
J1205-10	J400-a6		AQUI_3_B	-	Can be used for external instrumentation
J1205-11	J400-a5		AQUI_4_A	-	Can be used for external instrumentation
J1205-12	J400-a4		AQUI_4_B	-	Can be used for external instrumentation
J1205-13	J400-a3		AQUI_5_A	-	Can be used for external instrumentation
J1205-14	J400-a2		AQUI_5_B	-	Can be used for external instrumentation
J1205-15	J400-c11		AQUI_6_A	-	Can be used for external instrumentation
J1205-16	J400-c10		AQUI_6_B	-	Can be used for external instrumentation
J1205-17	J400-c9		AQUI_7_A	-	Can be used for external instrumentation
J1205-18	J400-c8		AQUI_7_B	-	Can be used for external instrumentation
J1205-19	J400-c7		AQUI_8_A	-	Can be used for external instrumentation
J1205-20	J400-c6		AQUI_8_B	-	Can be used for external instrumentation
J1205-21	J400-c5		AQUI_9_A	-	Can be used for external instrumentation
J1205-22	J400-c4		AQUI_9_B	-	Can be used for external instrumentation
J1205-23	J400-c3		AQUI_10_A	-	Can be used for external instrumentation
J1205-24	J400-c2		AQUI_10_B	-	Can be used for external instrumentation
J1205-25	J400-c1		GND	-	Can be used for external instrumentation

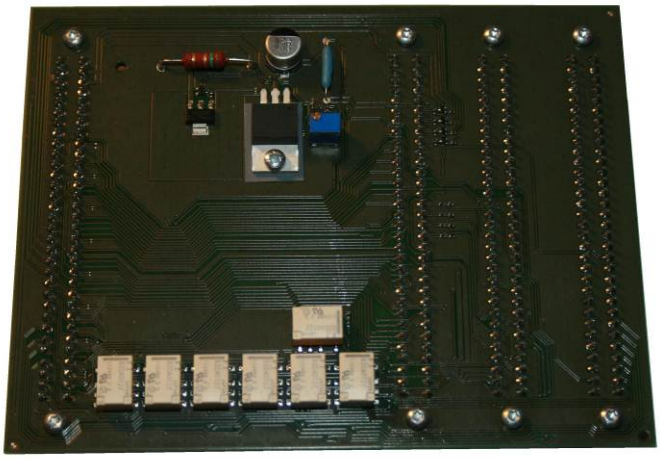
C. **Creating a customization board for the RTX2300**

In this section instructions along with essential items of information for creating a customization board for the RTX2300 are provided. Please note that a customization board is not always needed. If only a few signals need to be configured in the customization area this can be solved by just using a number of connectors and connecting these directly using some wires. However, the most efficient way to handle more complex and extensive customizations is to create a customization board (see figure below) – i.e. a PCB with connectors and the logic needed to connect signals from the input connectors (J400, J401, J403) to the output connector (J402).

Example Customization board (top and bottom)



Customization board (bottom side)



Customization board (top side)

The most important issue in either of the approaches is to be careful when connecting the customization input and output connectors. For an overview of the connector pins please refer to section B on page 103 (chapter 6). If the customization board (or wiring) does not work as intended it can be debugged by using the RTX2300 Detective debug application (see section B in chapter 3 for an explanation on how to debug customization boards – specifically the paragraph on page 76).

i. Customization board - dimensions and connector types

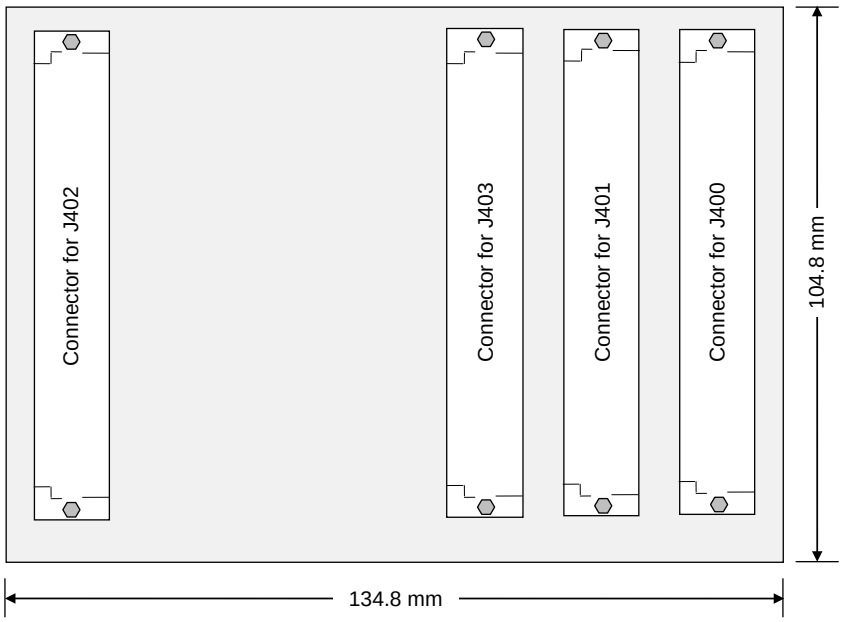
the customization board must adhere to the connector types and location on the RTX2300 main board. Consequently, the two most important items of information when creating a customization board are:

1) Board dimensions and exact location of the connectors
2) Connector types

Board dimensions and connector location

A typical customization board is a 2-sided PCB with the dimensions 104.8 mm x 134.8 mm (Width x Length) – see the figure below for a typical customization board with connectors.

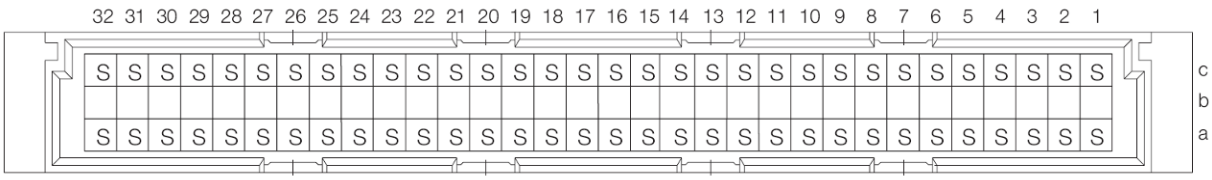
Customization board – dimensions and connector locations



Connector types

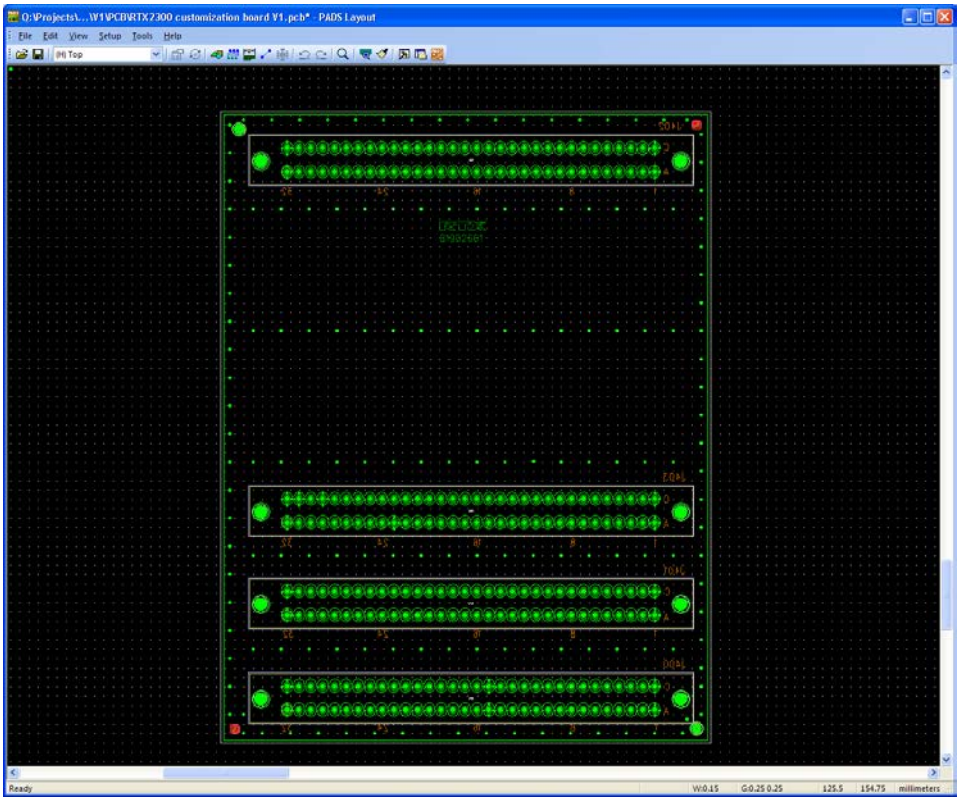
The connectors on the customization board must be of the type 2 x 32 pins DIN 41612 Type C male (see below).

Connector for customization board - 2 x 32 pin DIN 41612 Type C male



ii. Reference PADS file

To make it easy for users to create customization cards for the RTX2300 Smart ATE a PADS file is included on the installation package – i.e. available on the RTX Tester Download Center. The file is in the `RTX2300_customization_board` directory and it includes the basic PCB layout (see figure below) and schematic.

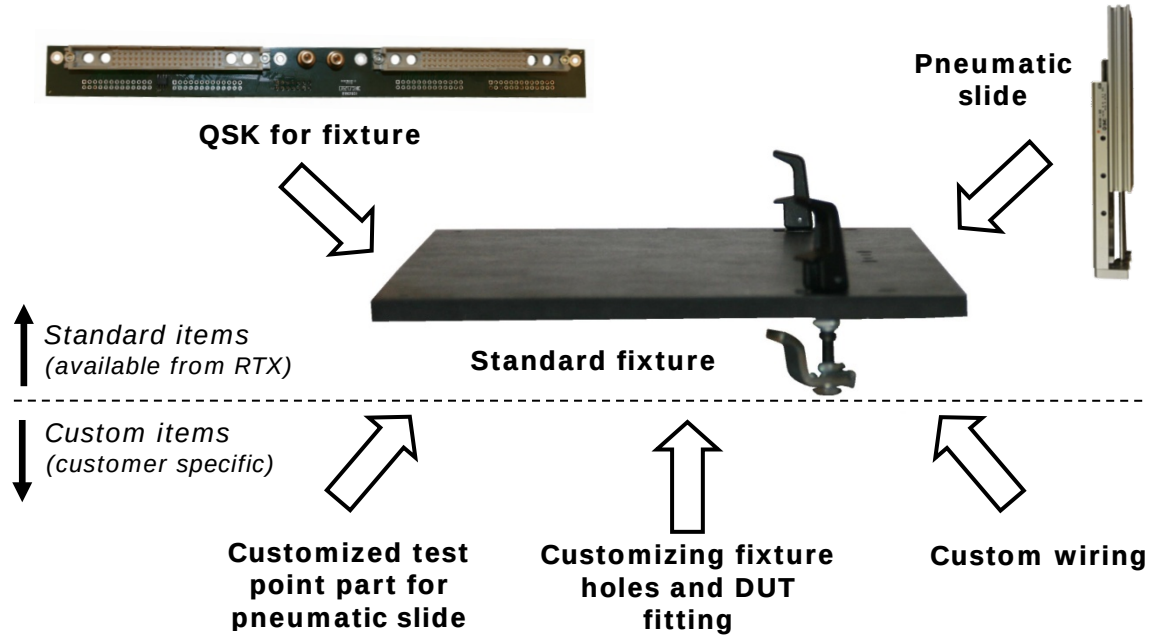


Please note that the layout only includes the pads for the connectors – i.e. components and related layout needs to be added before the customization card is complete.

D. Customization of fixtures for the RTX2300

Fixtures for the RTX2300 can be customized to very specific purposes. the standard fixture plate can be customized with both standard items and very specialized customization items like wiring, fixture holes and fittings, and a test point part (as outlined in the figure below).

Customization of fixtures – standard (available from RTX) and custom items



Consequently, the customer has a wide range of possibilities to customize the fixture to very specific purposes. Please note that the standard customization items are available from RTX. Furthermore, the customer can create a standard fixture plate by using the drawing included on the CR-ROM. Please note that it is essential to use this drawing when creating fixture plates since it includes location of the holes for the QSK option and the fixture locks. Each of the customization items from the figure above will be described briefly below.

- **Standard items** – the standard customization items are available directly from RTX and can be easily installed.
  - *Fixture plate* – A standard fixture plate is a plate with the correct dimensions and it only includes the fixture locks and relevant holes for installation of the QSK and a pneumatic slide.
  - *QSK for fixture* – The QSK for the fixture can be installed for easy replacement of fixtures. This part is also described in more detail in section C on page 91.
  - *Pneumatic slide* – If needed in the test configuration the test pins can be installed on a pneumatic slide hence enabling control of the test pins. As mentioned earlier the pneumatic slide is available in different sizes.

- **Custom items** – The custom items are items which are made by the customer.
  - *Test point plate for pneumatic slide* – The plate with test point pins are specific to each product, and hence, must be custom made. It can either be mounted on a pneumatic slide or mounted as part of the fixture itself.

- *Custom holes and fittings* – The fixture can be populated with fittings to lock the DUT in a specific position under test. Furthermore, custom holes for wiring, pneumatic items or other relevant fixture items can be made.
- *Wiring* – The wiring from the SCB and CCB interfaces in the RTX2300 depends highly on the configuration of the unit, and hence, this will also vary from product to product.
- *Other customization items* – In addition to the above customization items some very customer specific items are also possible – i.e. the above list is not exhaustive but only outlines the typical items that cover most situations.

the customer is responsible for performing customization of the fixture since the fixture configuration depends highly on the specific DUT and the requirements to test points. Therefore, the objective of this section is to provide the customer with an overview of how to customize the fixture.

7. Implementing RTX2300 test programs

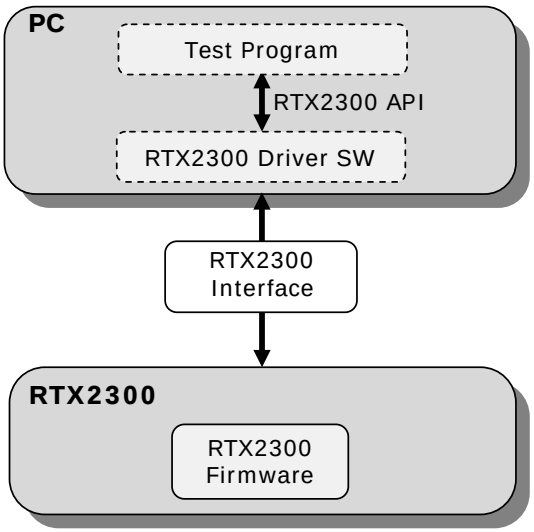
A key part in an RTX2300 Smart ATE system is the actual PC test program – i.e. the program that implements the specific production test suite(s). The RTX2300 test program is executed on a Windows PC and makes use of the functionality provided by the RTX2300 DLL (i.e. the RTX2300 API). The PC test program can be implemented in various ways – it can be implemented as a simple program controlled from the command line or as a more advanced GUI based application. The customer must make the actual design choices on how to implement a suitable PC test program for their specific use. This chapter will outline central items of information in relation to implementation of RTX2300 test programs - especially the following three subjects will be included:

- **RTX2300 API overview** – providing a brief overview of API functions and relevant API documentation.
- **How to implement RTX2300 test programs** – providing some example programs.
- **Dynamic Link Library Interfacing** – providing guidelines on how to call DLL functions and use implicit or explicit linking of the DLL.

The main purpose of this chapter is to enable SW engineers to write production test programs for a RTX2300 Smart ATE system. It is strongly emphasized here that understanding all three elements above is important. Especially the section on DLL linking is central to understand before starting implementation of a test program.

A. RTX2300 API overview

In the installation directory two interface documents are included – *"RTX2300 Common Interface"* and *"RTX2300 Interface"*. The first document includes common type definitions and the second describes the interface towards the RTX2300 – i.e. *"RTX2300 Interface"* in the figure below.



the RTX2300 API is a mapping of the primitives described in the *"RTX2300 Interface"* document to API functions. Consequently, only an overview of the API functions will be

provided here. For detailed information in relation to specific RTX2300 API parameters (and primitives) please refer to the interface documents and specific header files provided as part of the installation package.

In addition to the interface documents some header files are also included – they were also briefly mentioned in section B on page 44. A short description of each file is provided below.

- **`DllInterface.h`** – This header file includes type definitions related to the DLL interfacing.
- **`IRtx2300.h`** – This header file contains type definitions and primitive overview including function definitions of primitive sending functions. The header file is related to the RTX2300 Interface (i.e. the interface described in the *"RTX2300 Interface"* document).
- **`IRtx2300IntfCommon.h`** – This header file contains common type definitions for the RTX2300 Interface (i.e. the common interface definitions described in the *"RTX2300 Common Interface"* document).
- **`Rtx2300BasicTypes.h`** – This header file contains type definitions of few basic types.
- **`Rtx2300Primitives.h`** – This header file contains primitive ID's for all RTX2300 primitives.
- **`Rtx2300SysTypes.h`** – This header file contains general system wide type definitions.
- **`Rtx2300PcIntf.h`** – This header file is the central header file for the RTX2300 API and it contains function definitions for all RTX2300 API functions.

All the above files are relevant when developing test programs for the RTX2300. However, in relation to the RTX2300 API functions the `Rtx2300PcIntf.h` header file is central. The API functions are divided into four different categories:

- **General functionality:** This category contains functions for performing interface initialization, connection check, install/uninstall mail handler, install error handler, install log handler, mail queue access etc.
- **Firmware update functionality:** This category contains functions for checking the firmware version and to perform firmware update.
- **Debugging functionality for the DLL:** this category contains debugging functions for use in debugging DLLs, and hence, this category is not specific to the RTX2300. Furthermore, this category is typically not used when implementing test programs for the RTX2300.
- **Blocking API functionality:** This category contains most of the API functions. These functions are get/set functions which block until a response from the RTX2300 has been received.

In section B in this chapter the usage of the RTX2300 API is outlined through a few examples. Please make sure to go through the examples and the available source code in Appendix A to C.

B. How to implement a RTX2300 test program

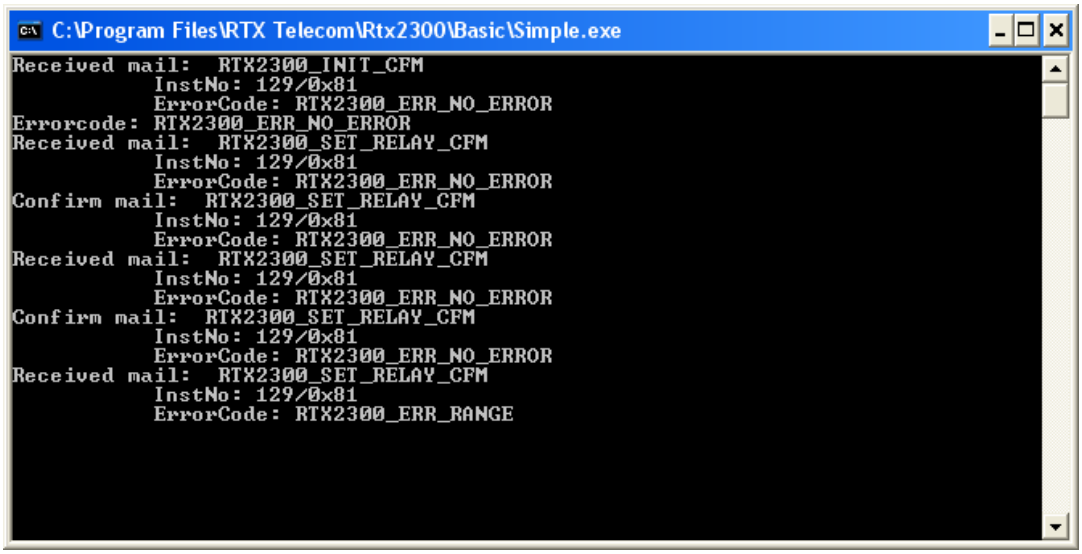
To visualize how to implement test programs three examples are included in the RTX2300 Basic Unit SW package. All examples include source code and Microsoft Visual C 2008 project files. The examples packages are in the installation directory (i.e. the three zipped files – `Simple.zip`, `MultiInst.zip` and `FirmwareUpdate.zip`). All examples will be described in more detail in the following. Please note that the source code for each of the examples is included in Appendix A to C. Prior to compiling and executing the example programs please also ensure that one or more RTX2300 units are connected to the PC and that they have been configured in the REPS.

i. Example 1 – simple test program

The first example is the `Simple` example which demonstrates basic interfacing to a basic RTX2300 Smart ATE system (i.e. consisting of only one RTX2300 unit). When executed it connects to an existing system using the port server and flips Relay 0 twice. Furthermore, the `Simple` example also shows logging and handling of received mails.

To compile the `Simple` test program the required files from the RTX2300 package must be copied to a relevant directory (i.e. the build directory) – including all `.dll`, `.h` and `.lib` files. The example code is zipped, and hence, please unzip the `Simple.zip` to the relevant directory. Build the example executable (`Simple.exe`), check that a RTX2300 unit is connected to the PC and execute the test program. Please observe that the relay is flipped and check the output in the console window (see below). If the test execution fails, please check the connection between the RTX2300 unit and the PC along with the configuration in the REPS.

Output in the console window when executing the `Simple.exe` example



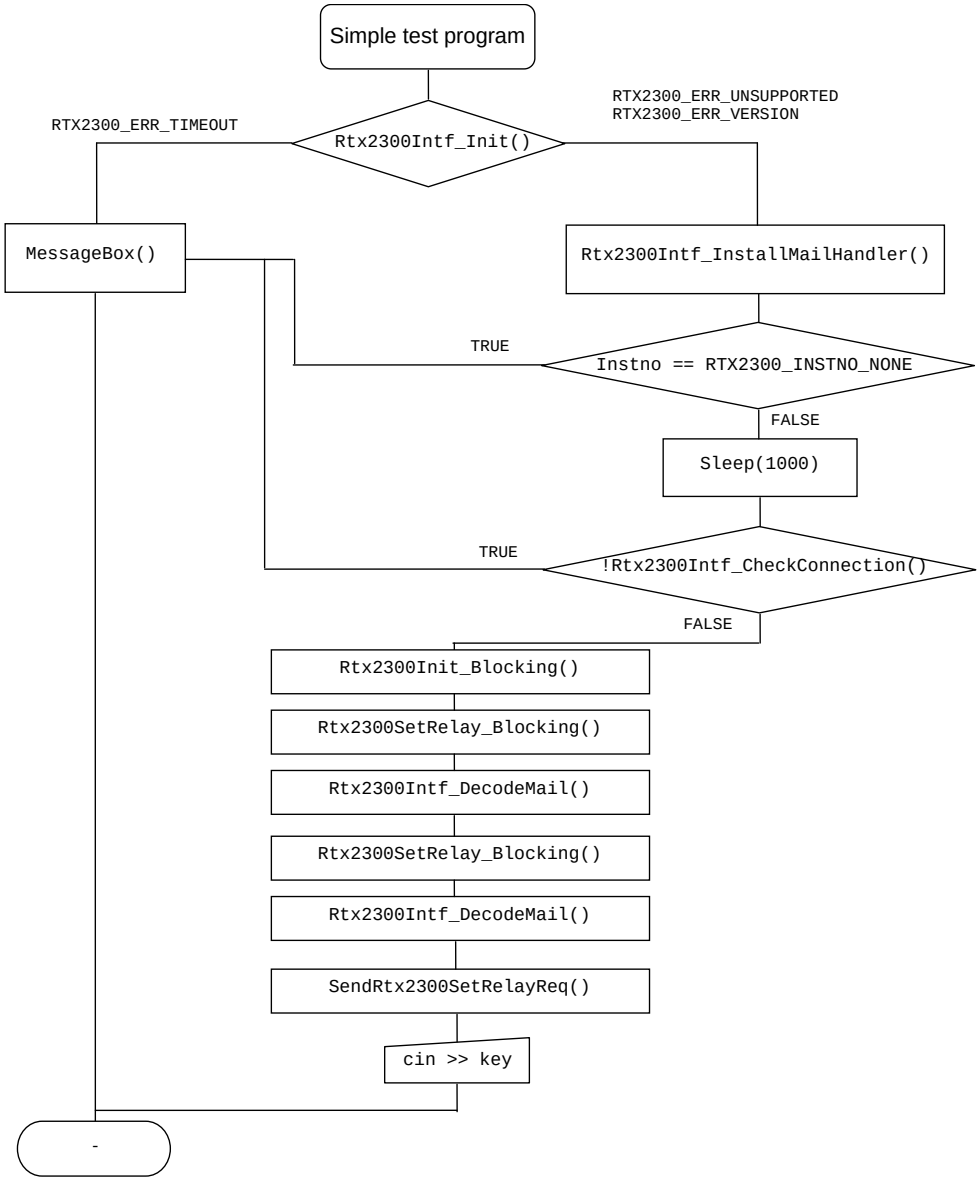
To end the test program please press any key and hit Return. Please refer to Appendix A for the source code for the test program. In the following the source code will be discussed in more detail to visualise how to use the RTX2300 API along with an overview of the logical operation of the test program (see the flow-diagram below). The line numbers mentioned in the following hence refers to the source code listing in Appendix A. For a detailed

description of the RTX2300 API functions please refer to the information in section A in this chapter.

The handler for the mail queue is defined in lines 12-20. This function will be called every time a mail has been received. Please notice the log messages with the decoded mail in the console window above. When the mail queue handler is activated it will first decode the mail by calling the `Rtx2300Intf_DecodeMail()` function (line 18) and it is logged using `cout` (line 19).

The main program (lines 23–95) includes four logical steps (see the flow diagram below).

Flow-diagram – `Simple.cpp` example test program



The first step is to initialize the interface (`Rtx2300Intf_Init()` in lines 30-31). If initialization fails due to a time-out or failure in getting the instance number, a message box will be displayed (`MessageBox()` in lines 36-37 or lines 48-49) and execution of the test program will stop.

Next step is to install the mail queue handler (`Rtx2300Intf_InstallMailHandler()` in line 43).

The third step is to check the connection to the RTX2300 unit but to allow the port server to reconfigure the transport layer a delay of 1 second is used (line 55). If the check (`Rtx2300Intf_CheckConnection()` in line 57) a message will be displayed in the console window (line 59) and the test program will be stopped.

The fourth step is to perform the actual test steps. In this case it is to flip a relay twice. However, before starting the test the unit must be initialized (`Rtx2300Init_Blocking()` in line 65). Next Relay 0 is flipped twice (`Rtx2300SetRelay_Blocking()` in lines 74-75 and lines 83-84) and `SendRtx2300SetRelayreq()` is called with a non-existing relay (i.e. relay parameter out of range). Hence, an error message will be displayed in the console window.

Please note that the interface initialization, connection check and unit initialization must be part of every RTX2300 test program while installation of a mail queue handler is optional.

After test execution the program will wait until any character has been input (line 92 and 94) before it terminates.

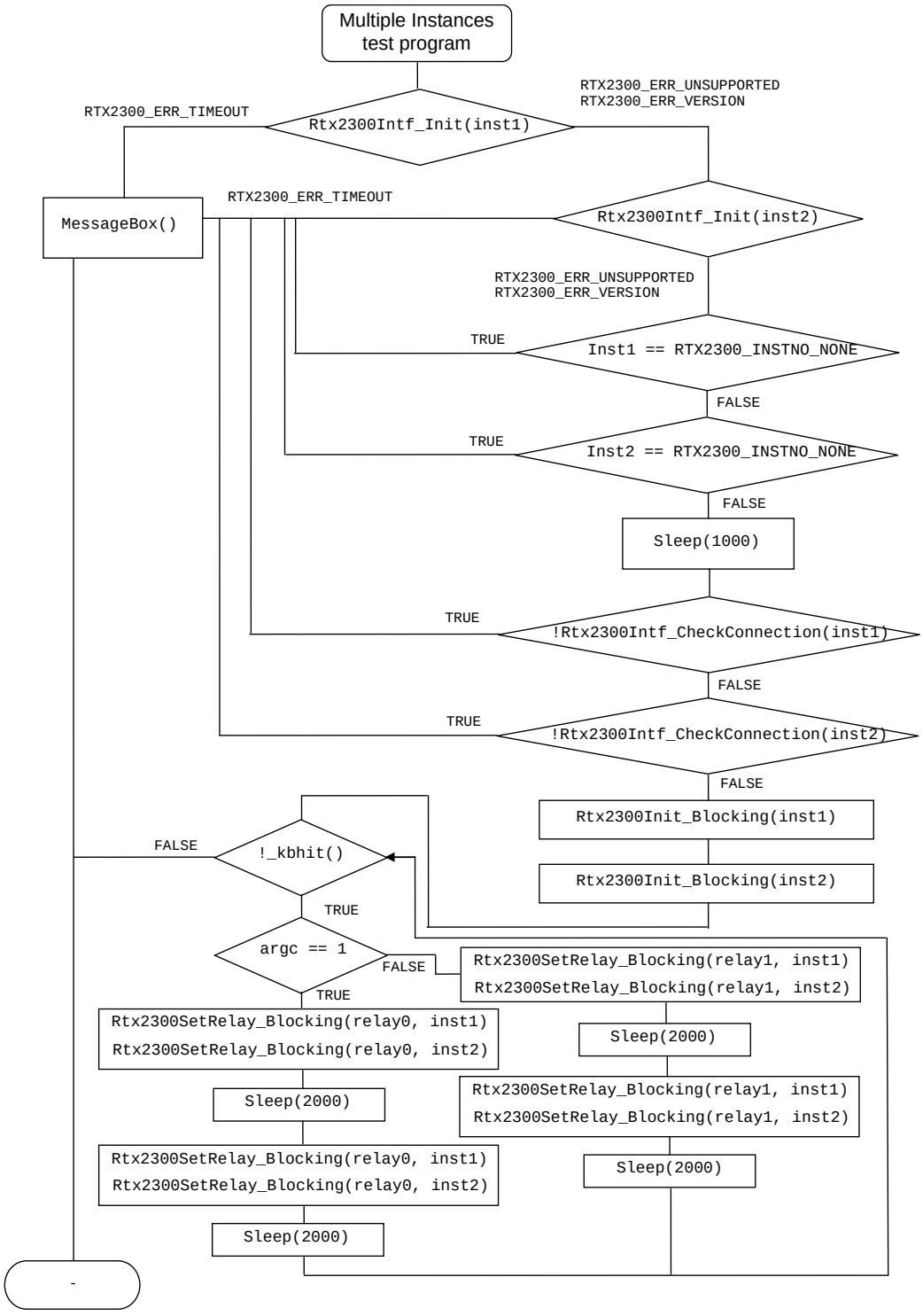
ii. Example 2 – test program controlling multiple RTX2300 units

the `MultiInst` test program (see source code listing in Appendix B) is an extended version of the `Simple` test program (see paragraph i in this section) and it demonstrates interfacing to a RTX2300 system with multiple RTX2300 units. The test program will start two instances and a relay will be flipped on both RTX2300 units. If an argument is specified on the command line (any argument will do) when starting the test program Relay 1 will be flipped – otherwise Relay 0 will be flipped. For a logic overview of the test program please see the flow-diagram below. Please also be aware of the fact that the line numbers mentioned in the following refers to the source code listing in Appendix B.

Please note that the `MultiInst.exe` test program will connect to two RTX2300 units using the port server names `"PsTarget1"` and `"PsTarget2"` (i.e. `portservername` equal to `"PsTarget1"` and `"PsTarget2"` in the initialization function calls), and hence, these ports must exist. *Consequently, please note that port servers with these names must have been configured to the correct COM port numbers prior to running this example test program!!*

To compile the `MultiInst` test program the required files from the RTX2300 package must be copied to a relevant directory (i.e. the build directory) – including all `.dll`, `.h` and `.lib` files. The example code is zipped, and hence, please unzip the `MultiInst.zip` to the relevant directory. Build the example executable (`MultiInst.exe`), check that two RTX2300 units are connected to the PC and execute the test program. Please observe that the relay is flipped on both RTX2300 units and check the output in the console window. If the test execution fails, please check the connection between the RTX2300 units and the PC along with the configuration in the REPS.

Flow-diagram – `MultiInit.cpp` example test program



The first two steps in the `MultiInst` test program is to initialize the two interfaces (`Rtx2300Intf_Init()` in lines 19-20 and 32-33) and check the connection to the two units (`Rtx2300Intf_CheckConnection()` in lines 64 and 69). If any of these fails a message box is displayed and the test program will terminate. Please also note the 1 second delay in line 62 before checking the connection to the units.

Next step is to initialize the two units (`Rtx2300Init_Blocking()` in lines 77 and 83). The actual test program contains the lines 89 to 122 and the main functionality is to flip either Relay 0 (`Rtx2300SetRelay_Blocking()` in lines 95-96 and 111-112) or Relay 1 (`Rtx2300SetRelay_Blocking()` in lines 102-103 and 117-118) on both RTX2300 units. Please notice the 2 second delay in lines 106 and 121. Flipping of the relay will continue until a key is pressed.

iii. Example 3 – firmware update program

The last example is a bit different than the other two. The firmware update program – `demo.exe` – does not flip any Relays but instead updates the firmware of one RTX2300 unit (if necessary). For a logic overview of the test program please see the flow-diagram below. Please also know the line numbers mentioned in the following refers to the source code listing in Appendix C.

To compile the `Demo` test program the required files from the RTX2300 package must be copied to a relevant directory (i.e. the build directory) – including all `.dll`, `.h` and `.lib` files. The example code is zipped, and hence, please unzip the `FirmwareUpdate.zip` to the relevant directory. Build the example executable (`Demo.exe`), check that a RTX2300 unit is connected to the PC and execute the test program. Please be aware of the difference in naming of the package and the executable (i.e. `FirmwareUpdate.zip` vs. `demo.exe`).

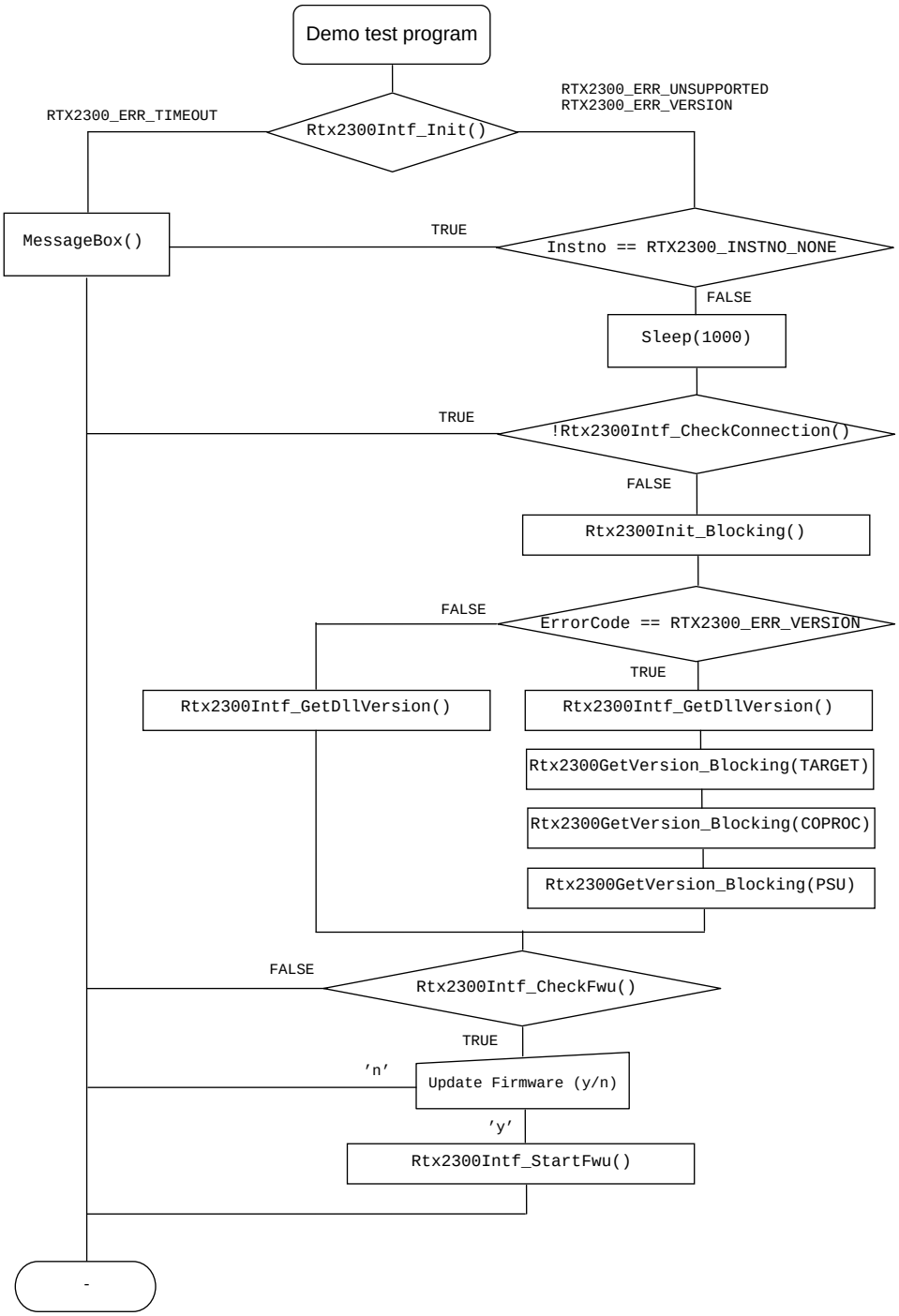
The first steps of the Demo test program are similar to the other two examples – i.e. initialization of the interface towards the RTX2300 (`Rtx2300Intf_Init()` in lines 19-20) and a connection check (`Rtx2300Intf_CheckConnection()` in line 45). If any of these fails a message box is displayed and the test program will terminate. Please also note the 1 second delay in line 62 before checking the connection to the units.

Next step is to initialize the RTX2300 unit (`Rtx2300Init_Blocking()` in line 49). The actual test program contains the lines 51 to 121 and the main functionality is to check the firmware version and perform firmware update if necessary.

If the return value from the initialization function call indicates `RTX2300_ERR_VERSION` in the `ErrorCode` parameter the current firmware versions of the co-processor and the PSU will be requested (lines 61-86) – i.e. the DLL version (`RTX2300Intf_GetDllVersion()` in line 57) and target firmware version (`RTX2300GetVersion_Blocking()` in lines 59-60) will be shown in the console. Otherwise the current firmware version will just be output.

Next the actual firmware update procedure will be initiated (from line 95 and forward). The first step in the firmware update process is to check whether or not a newer firmware version is available (`RTX2300Intf_CheckFwu()` in line 99). Please note that this function will check the version of the firmware file in the installation directory and acquire the current firmware version from the RTX2300. If a newer version exists, the user is asked whether the firmware should be updated (lines 103-106). If 'y' is pressed the firmware will be updated (`RTX2300Intf_StartFwu()` in line 111) and the test program will terminate upon completion.

Flow-diagram – `Demo.cpp` example test program



Please note that the lines 124-143 are not directly used in the outlined example. status information is requested from the unit (`Rtx2300GetStatus_Blocking()` in line 127) and the test program will output some items of information and terminate.

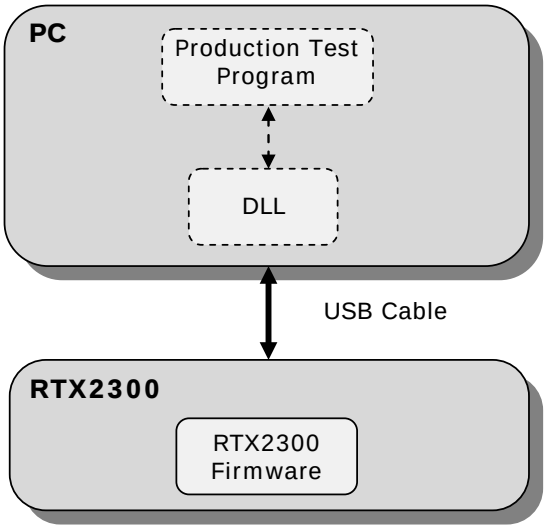
C. Dynamic Link Library Interfacing

To successfully control the RTX2300 unit using the RTX2300 API it is important to understand how to link to the supplied RTX2300 Dynamic Link Library (DLL).

Microsoft Windows provides ways to use dynamic link libraries. But it is important to know various programming/compiler tools adopt slightly different approaches to DLL linking. In this section the most common ways to perform DLL linking using Windows WIN32 C++ API are described. Minor adaptations may be necessary when other programming tools are used.

i. Dynamic Link Library Interface Description

The USB interface is used to communicate with the RTX2300 unit. To make the RTX2300 unit user-friendly, the USB communication has been encapsulated in some API functions which are provided as a Windows DLL (`Rtx2300PcIntf.dll`). As indicated in the figure below the RTX2300 Production Test Program makes use of this DLL for communication with the RTX2300 unit.



In the following paragraphs central items of information regarding use of the DLL in your own programs are provided. Issues like calling convention, command format and explicit/ implicit linking of the DLL are outlined.

Calling Convention

As mentioned above different programming languages and compilers use different approaches when performing subroutine calls. For example, the methods used to store parameters and return addresses on the stack vary. The term *calling convention* is defined to cover the different access methods. Hence, since the user application may not have been created in the same programming language as the DLL, it is crucial to know the calling convention to use when DLL functions are called from your source code.

The calling convention used by the RTX2300 unit DLL is the **__stdcall**. If C++ is used as the application programming language the calling convention is explicitly specified by the **__stdcall** keyword in the interface header file `DllInterface.h`.

Explicit DLL Linking

With *Explicit DLL Linking* the application only requires the interface header file `Rtx2300PcIntf.h` and the DLL itself. All DLL linking is done explicitly by the application program. Hence, using this approach you must first load the DLL module using the WIN API function `LoadLibrary()`. Thereafter retrieve the addresses for each DLL function explicitly through the function `GetProcAddress()`- i.e. for example:

```
DllHandle = LoadLibrary(<DllName>);
functionPtr = GetProcAddress(<DllHandle>, <function name>);
```

Implicit DLL Linking

With *Implicit Linking* the application requires the *Rtx2300PcIntf.lib* file in addition to the interface header file Rtx2300PcIntf.h and the DLL. The DLL linking is in this case done implicitly by the compiler used for creating the application program. The compiler will recognize this because of the keyword **__declspec(dllimport)** specified in the interface header file.

The information needed by the compiler to perform implicit linking is included in the .lib file, and hence, Rtx2300PcIntf.lib must therefore be included in the source file list of the application project. Unfortunately, there seems to be compiler differences between .lib formats. **Therefore, it is recommended to use implicit DLL linking only with Borland compilers**. In other cases (e.g. Microsoft) explicit DLL linking is recommended.

8. Specifications and characteristics

A. **Introduction**

This chapter includes specific details related to the functionality and performance characteristics of the RTX2300 and the available module options.

B. **RTX2300 Smart ATE basic unit**

This section includes specific details related to the functionality and performance of the RTX2300 Smart ATE Unit. It includes the following items of information:

- D/A channels
- Tone generator
- Audio buffer
- A/D channels
- Audio measurement
- GPIO
- General specifications

Characteristics for each of these will be outlined in the following paragraphs. Please note that the RTX2300 Smart ATE complies with the specifications outlined in this chapter after 2 hours of storage within the environmental temperature, and 60 minutes after turn-on.

i. D/A channels

The RTX2300 main board has 2 DAC outputs connected to the SCB. Please note that the DAC is only intended for DC values and should *not* be used for audio signalling. See the table below for an overview of the functional performance of the 2 DAC outputs.

Specification, DAC

Description	Condition	Min.	Nom.	Max.	Unit
Channels			2		
Resolution		12			Bits
Voltage range [1)]	Load > 10kΩ typical value	-10		+10	V
Voltage steps				10	mV
Tolerance				20	mV
Source current	(Output buffer)	30			mA
Sink current	(Output buffer)	30			mA
Short circuit protection	No damage upon system failure	-10		10	V

1) Limited by ±10V AFE supply and load current

ii. Tone generator

The RTX2300 main board includes three tone generator outputs with adjustable amplitude and frequency; one high level single ended output and two configurable (differential or single ended) low level outputs. Please note that the low-level outputs cannot be configured as two separate single outputs with different frequency and amplitude. See the tables below for an overview of the functional performance of the tone generator outputs.

Specification, Tone generator (high level)

Description	Condition	Min.	Nom.	Max.	Unit
Channels	*Single ended output*		1		
Output Voltage	*THD < 1%, 1kHz @R_L > 30k*			10	V_{PP}
Amplitude flatness 50Hz to 10kHz	*Relative to 1kHz*			1	dB
Amplitude flatness 50Hz to 14kHz	*Relative to 1kHz*			2	dB
Resolution Frequency			1		Hz
Frequency range		0.05		14	kHz
Harmonic distortion				1	%
Output impedance			75		Ω

Specification, Tone generator (low level)

Description	Condition	Min.	Nom.	Max.	Unit
Channels	*Single ended output*		2		
Channels	*Differential output*		1		
Single ended output voltage	*THD < 1%, 1kHz @R_L > 30K*			2,5	V_{PP}
Differential output voltage	*THD < 1%, 1kHz @R_L > 30K*			5	V_{PP}
Amplitude flatness 50Hz to 8kHz	*Relative to 1kHz*			0,5	dB
Amplitude flatness 50Hz to 13kHz	*Relative to 1kHz*			3	dB
Resolution Frequency			1		Hz
Frequency range		0.05		14	kHz
Harmonic distortion				0,2	%
Output impedance			75		Ω

iii. Audio buffer

Two differentials to single ended audio signal buffers are provided on the RTX2300 main board. The input and output signals can via the option matrix be connected to the CCB connector block. See the table below for an overview of the functional performance of the audio buffer.

Specification, Audio buffer

Description	Condition	Min.	Nom.	Max.	Unit
Channels			2		
Output voltage range	Single ended	-10		10	V_P
Input voltage range	Differential	-5		5	V_P
Buffer Lower cut-off	-3dB limit			20	Hz
Buffer higher cut-off	-3dB limit			20	kHz
Buffer Pass-band gain	100Hz to 10kHz	-0,5	0	+0,5	dB
Buffer filter order			2		
Buffer Gain		-1		+1	dB
Harmonic Distortion			0,1	0,2	%

iv. A/D channels

The RTX2300 main board provides eight 16-bit ADC ports (single ended with common GND). All the ADC channels are available at the SCB interface. See the table below for an overview of the functional performance of the A/D channels.

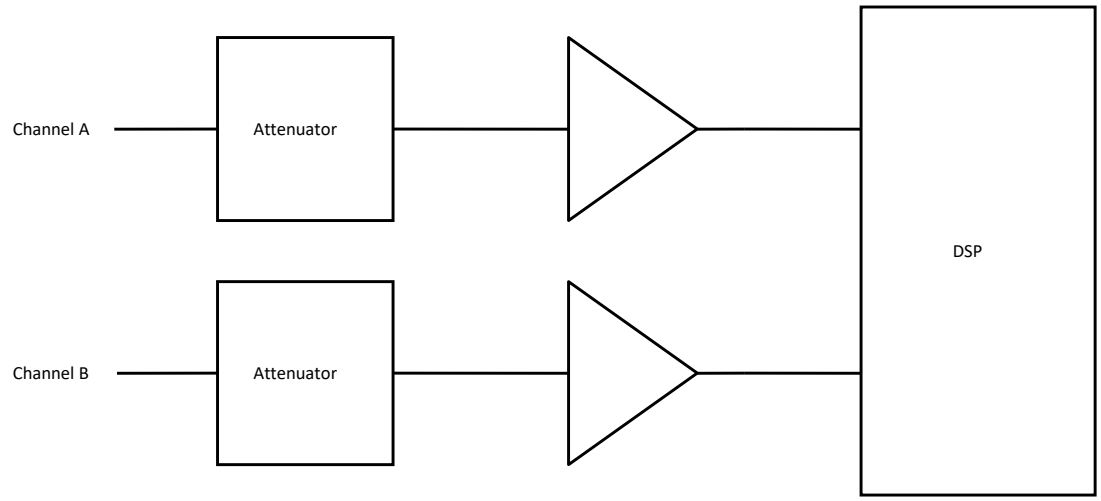
Specification, A/D

Description	Condition	Min.	Nom.	Max.	Unit
Channels	Single ended - common GND		8		
Resolution	Full scale		16		Bit
Voltage range (target input)	0 to full scale digital signal	-10		+10	V
Voltage steps	No scaling		0,37		mV
Measurement accuracy single ended	In a measurement range from -12 to +12 Volt	5	50	100	mV
	Offset	0		10	mV
Measurement accuracy differential	In a measurement range from -6 to +6 Volt		5		mV
	Offset	0		5	mV
Linearity			1		mV
Max sampling rate		10			Hz
Measurement type			DC		
Scaling range		-12		0	dB
Scaling steps	Single ended positive/ negative or differential ±		7		

v. Audio measurement

The RTX2300 includes logic for performing measurements on AC signals (two channels). This functionality is implemented in a DSP with adjustable attenuators in front of the DSP inputs (as outlined in the figure below). Please note that the maximum input level at the DSP input pins is 188mV$_{rms}$.

2-channel audio measurement logic (DSP with adjustable attenuators)



AC signals on the input are attenuated (0 - 62 dB) and buffered before they are analyzed using the DSP. The DSP is able to measure RMS voltage, Peak Voltage and Distortion.

Specification, AC measurement

Description	Condition	Min.	Nom.	Max.	Unit
Input channels			2		
Frequency range		0,1		13	kHz
Input voltage range	30dB range			±5	V_P
Attenuator range		0		-62	dB
Accuracy 0dB range	100Hz to 13kHz, Sine wave	-0,2		+0,3	dB
Accuracy -30dB range	100Hz to 13kHz, Sine wave	-0,2		+0,8	dB

Specification, THD measurement

Description	Condition	Min.	Nom.	Max	Unit
Input channels			2		
Fundamental frequency range		0,75		2	kHz
Accuracy	Fundamental frequency 1kHz	-2		+2	%
Input voltage range	Min 0dB range, Max 30dB range	±0,0		±5	V_P
Attenuator range		0		-62	dB

Due to the FFT size (256 bits) in the DSP the fundamental frequency should be a multiple of (16000/256=) 62,5Hz for distortion measurements. A good choice is to use a fundamental frequency of 1 kHz (1000/62.5=16). Next useful frequency is 1062.5 Hz and so forth.

The Total Harmonic Distortion (THD) is defined by the formula below where terms 2...N are the power levels of the harmonics and term 1 is the power level of the fundamental (pure tone).

$$\%THD = \frac{\sqrt{H_2^2 + H_3^2 + ... + H_N^2}}{\sqrt{H_1^2 + H_2^2 + H_3^2 + ... + H_N^2}} \times 100$$

vi. GPIO

The RTX2300 provides 16 outputs (8 sink/8 source) to the CCB selection matrix along with 8 digital inputs (i.e. 24-bit in total).

Specification, Output source

Description	Condition	Min.	Nom.	Max.	Unit
Output Channels			8		
Maximum output voltage source port		2	V_{Ref}	12	V
Output source mode per channel				100	mA
Max total				500	mA

Specification, Output sink

Description	Condition	Min.	Nom.	Max.	Unit
Output Channels			8		
Output sink current per channel				-100	mA
Max total				-500	mA
Maximum output voltage sink port		2	V_{Ref}	12	V

Specification, Input

Description	Condition	Min	Nom.	Max.	Unit
Input Channels			8		
Input impedance		100			kΩ
Input voltage			5		V
Input voltage, high		2.0			V
Input voltage, low				0,8	V
Input overvoltage protection		-0.5		5.5	V

vii. General Specifications

Power supply

The RTX2300 is supplied through a 12 V power supply (with GND). From this the internal supply-voltages are generated. Two Switch-mode Power Supplies (SMPS) provide the needed -12V and +5V supplies - the rest are generated from these by linear regulation. The DUT can be supplied from a separate programmable power-supply module (optional) or through the connection of an external power source.

Power adapter:
Input: 100 - 240VAC 50-60 Hz, 1.5 A
Output: 12 V, 5 A
Power consumption 60 VA maximum

Mechanical specification

Physical Dimensions of the RTX2300 Smart ATE main unit:

- 500 x 300 x 310 mm (Depth x Width x Height)

Weight of the RTX2300 Smart ATE main unit:

- 8.0 kg

Maximum DUT size supported by the RTX2300:

- 200 x 150 x 30mm (Length x Width x Height)

RF shielding:

- 1.0GHz – 6.0 GHz: typical 60 dB

Reliability requirements

Operation:
Operating Temperature range 10°C to 40°C (50°F to 104°F)
Operating Humidity up to 95% relative humidity to 40°C (104°F) (non-condensing)

Storage:
Storage Temperature range -25°C to 70°C (-13°F to 158°F)
Storage Humidity up to 100% relative humidity to 40°C (104°F) (non-condensing)

ESD:
Contact with covers closed: ±4kV
Air with covers closed: ±8kV

Input / output connectors

- AUX Port connector (25-pin DSUB female)
- Ground connector
- CLK Port connector (type SMA female)
- USB Interface (Type B receptacle)
- DUT Ext. Power supply connector (type is XLR-4 male)
- Fixture power supply connector (type XLR-3 male)
- RF IN/OUT connector (type SMA female)
- Air connector (pneumatic male connector)

C. Optional Modules

The RTX2300 supports some optional modules including:

- Programmable PSU
- Frequency Counter
- RF Switch

Specifications for each of the modules are outlined in the following paragraphs.

i. Programmable PSU

Specification, Programmable PSU module

Description	Condition	Min.	Nom.	Max.	Unit
Channels			1		
Voltage Range	$I_{min} < I < I_{max}$	1		15	V
Max ripple voltage	100 HZ pulsed max load			10	mV
Voltage programming step size				20	mV
Supply non-linearity	1-15V			5	%
Load regulation				10	mV
Voltage measurement resolution	0-15V			20	mV
Voltage measurement Tolerance	0-6V			1	%
I range 0-15V	Option for reducing max. current at voltages over 10 V due to heat dissipation	0		2	A
I_{limit} programming step size	(average)		2	2.2	ıA
I measurement resolution[1)]	0-200mA 200-2000mA			0.2 2	mA
Maximum external Voltage Range		12			V
External current Range		-6		6	A

1) Measures high range first and then low range if I<200mA and return best fit resolution.

ii. Frequency counter

Specification, Frequency Counter Module

Description	Condition	Min.	Nom.	Max.	Unit
Voltage range	Design must support as minimum	10			V_p
Signal Threshold	Minimum voltage guaranteed			100	mV_p
Measurement resolution	Relative to reference clock	0.1			Hz
Measurement accuracy	Counter accuracy (ref accuracy excluded)			0,05	ppm
Input coupling			AC/DC		
Zin (internal frequency counter)	Selectable high value Selectable low value	>100	50		Ω
Capacitive load (internal frequency counter)	Selectable high value Selectable low value			10 10	pF
Frequency maximum	Must support	50			MHz
Frequency minimum	Must support			1	Hz
Internal reference clock			10		Mhz
Internal reference clock tolerance – standard grade		-1		1	ppm
Internal reference clock tolerance – high stability		-0,1		0,1	ppm
Gate time		1	00	1000	ms

9. Maintenance

A. Introduction

This chapter describes general maintenance of the RTX2300 including central items of information in relation to return procedures. It contains the following sections:

- General customer responsibilities
- Operator Maintenance
- Contacting RTX A/S or RTX2300 distributors
- Returning your RTX2300 unit for service

B. General customer responsibilities

In general, the customer shall:

- Perform routine operator maintenance of the unit (general maintenance, replacement of consumables and cleaning) – see also next section for details
- Provide access to and use of all items of information and facilities determined necessary by RTX A/S to service and/or maintain the product(s)
- Provide access to the product(s) during the specified maintenance period(s) to enable RTX A/S personnel to perform maintenance on the product(s)
- Provide adequate working space around the products for servicing by RTX personnel

C. Operator Maintenance

This section describes general responsibilities of the customer and it includes instructions related to general maintenance (i.e. replacement of consumables and calibration of pneumatic pressure) and cleaning of the RTX2300 unit.

i. General maintenance

The customer is responsible for performing general maintenance of the RTX2300 unit including replacement of consumables (like the gasket on the lid and fuses) and calibration of the pneumatic pressure. In the following paragraphs central information is provided related to these responsibilities.

Replacing lid gasket

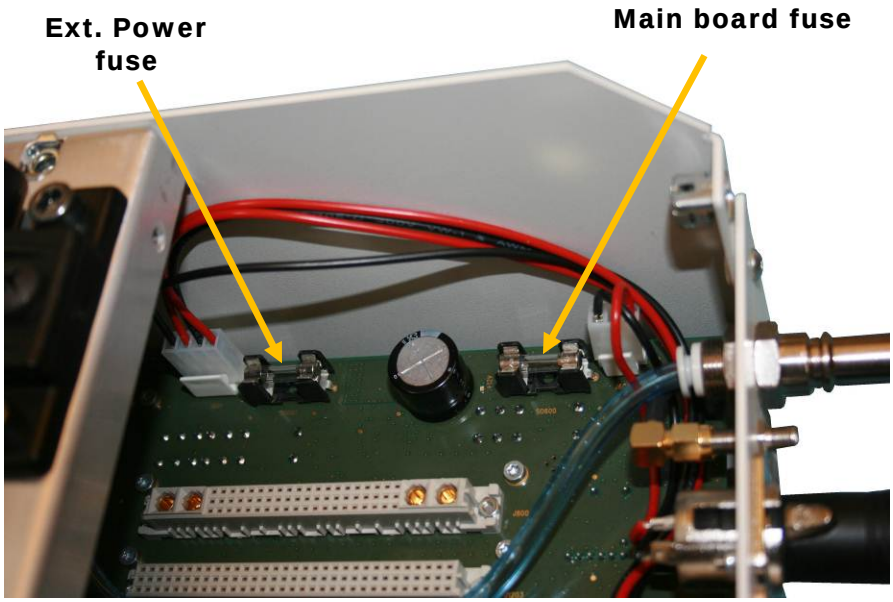
For optimal operation the gasket on the lid (see figure below) should be free from cracks and it must fit tightly to the RTX2300 main box. Consequently, it should be inspected regularly. If irregularities in the gasket are identified it must be replaced.

Gasket on the RTX2300 lid

Replacing fuses

If the RTX2300 does not operate as expected, please check the fuses on the main board (accessible from the controller chamber) and replace them when necessary. There are two fuses to check (see figure below).

Fuses on the RTX2300 main-board



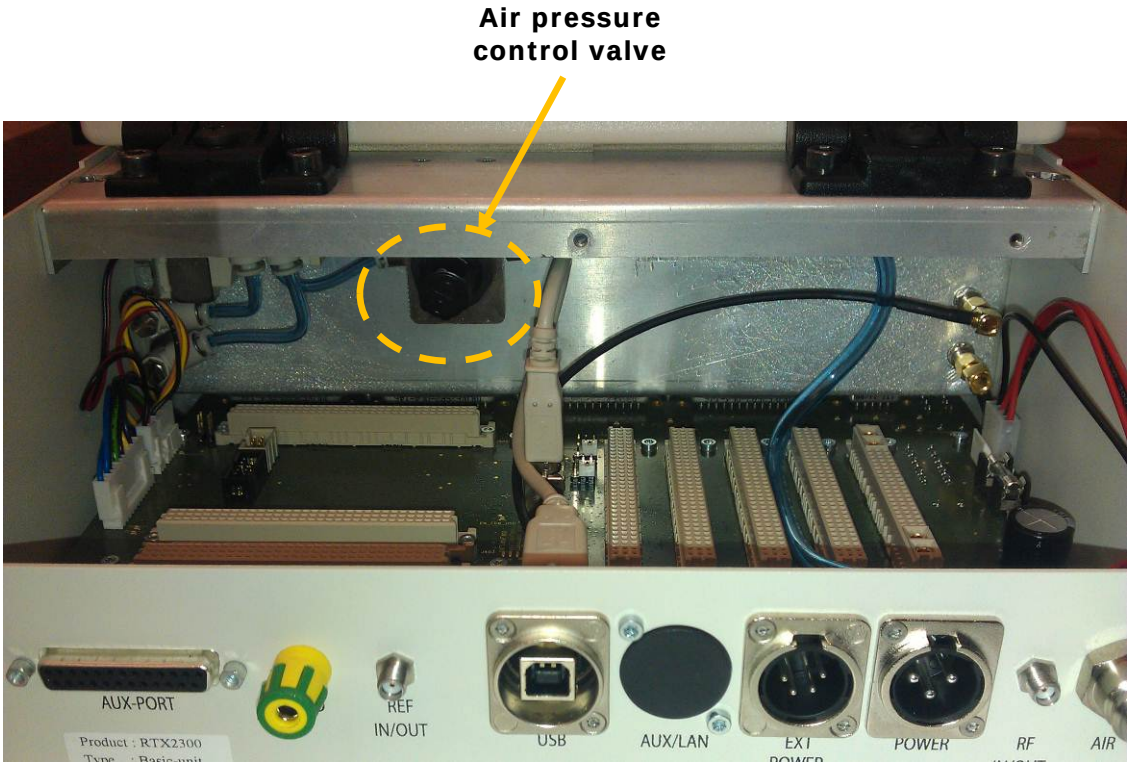
- One 2A fuse for the external power supply (i.e. an external power source connected to the Ext. Power connector on the rear panel).

- One 5A fuse for the main board power supply.

Calibration of pneumatic pressure

A pressure control valve for controlling the air pressure in the pneumatic system is placed near the back side of the controller chamber (see picture below). This valve is used for stabilizing the air pressure and to maintain constant air pressure the pressure in the pneumatic system should be calibrated regularly.

Location of air pressure control valve in the controller chamber



ii. Cleaning the RTX2300 unit

Since RTX2300 units typically will be placed in production environments they should be cleaned regularly. Before cleaning the RTX2300 unit please disconnect all power supplies. To clean the unit, use pressured air to clean both the controller chamber and the fixture chamber. Furthermore, wipe the unit with a damp cloth only.

D. Contacting RTX A/S or RTX2300 distributors

If you experience problems with your RTX2300 unit please read this section carefully before contacting RTX A/S or the distributor from which you purchased your RTX2300 unit, since important aspects and items of information in relation to support is outlined here.

If you wish to contact RTX A/S or the distributor in relation to any aspect of the unit - from service problems to ordering information - please refer to the paragraph *"Sales and Service Offices"* later in this section.

If you wish to return the unit to RTX A/S or the distributor please refer to the section: " Returning *Your RTX2300 unit for Service"* later in this chapter.

i. Before calling RTX A/S or the distributor

Before calling RTX A/S or the distributor from which you purchased your RTX2300 unit in relation to returning it for service please take a couple of minutes to go through the checklist outlined in the paragraph *"Check the Basics"* later in this section. The checklist will guide you through some basic checks to rule out some of the most common problems. Furthermore, it could help you identify the root cause of your problem(s).

If this does not solve your problem(s) then please read the warranty printed in the first pages of this User's Manual. If the problem(s) you experience seem to be covered by the warranty, please state this when contacting RTX A/S or the distributor.

If your unit is covered by a separate maintenance agreement, please look at the terms of the agreement and validate that your problem(s) is covered by the agreement. Please contact RTX A/S or the distributor for full details – see contact details in the paragraph *"Sales and Service Office"* later in this section.

If your RTX2300 unit becomes faulty and you wish to return it, please follow the description on how to return the faulty instrument in the paragraph *"Sales and Service Offices"* later in this section.

It is strongly emphasized here that the serial number of the unit is central in communication with RTX A/S or the distributor. Therefore, please check the serial number (located below the AUX-port on the rear panel of the unit) and include this in your communication.

ii. Check the Basics

To rule out some of the basic problems that could occur please take a minute to go through the checklist below. If the unit is still faulty after performing the checks please contact the RTX service office for information and support.

- Check that the line socket has power.
- Check that the unit is plugged into the proper ac power source and that there is power to the unit.
- Check that the fuses are in working condition.

- Check that the other equipment (if present), cables, and connectors are connected properly and operating correctly.
- Check the equipment settings in the procedure that was being used when the problem occurred.
- Check that the handling of the unit is within the specifications and capabilities of the unit.

iii. Sales and Service Office

You can contact one of the following Sales Offices or a registered RTX2300 distributor / service office (please refer to the list of distributors / service offices provided on our website http://www.rtx.dk). Here you can ask for a test and measurement sales representative. In any correspondence or telephone conversations, please refer to the RTX2300 unit by its model number and full serial number. With this information, the RTX A/S representative can quickly determine whether your unit is still within its warranty period.

Europe and Asia:

RTX A/S
Stroemmen 6
9400 Noerresundby
Denmark

Tel. +45 96 32 23 00
E-mail Sales: sales@rtx.dk
E-mail Service: service@rtx.dk

North America:

RTX America Inc.
10620 Treena St, Suite 230
San Diego, CA 92131
USA

Tel. +1 (858) 935 6152
E-mail Sales: sales@rtx.dk
E-mail Service: service@rtx.dk

For more information about RTX A/S test and measurement products, applications, services, and for a current sales and support offices along with a list of distributors, please visit our web site http://www.rtx.dk.

E. Returning Your RTX2300 Unit for Service

This section contains important items of information in case you need to return your RTX2300 unit to RTX A/S.

It is strongly emphasized here that all returns to RTX A/S MUST be initialized by obtaining a Return Material Authorization (RMA). Any returns without authorization cannot be handled in the normal service process and in a timely manner.

Please read the paragraphs very carefully and follow the instructions closely. The first step if you want to return your RTX2300 unit to RTX A/S is to obtain an RMA. To avoid damages to the unit when shipping it please pack your unit according to the recommendations provided in this section.

i. Obtaining an RMA for service return

To obtain an RMA (Return Material Authorization) please contact one of the RTX A/S Service office by E-mail (service@rtx.dk) or phone with the following items of information:

- Company name
- Company address
- Contact information
- Serial number of the unit
- Model Type
- A detailed error description with:
    - Any error messages generated by the unit
    - Any information on the performance of the unit
    - Fault description
    - Description of product usage and configuration
    - Block diagram or schematic of customization boards and test inserts if applicable
- Type of Service agreement or warranty

Upon receiving the above information, the Service Department will provide an RMA number. This number must be placed at a visible location on the shipping box, and furthermore, the RMA number must be used as a reference in all communication in relation to the return procedure.

ii. Packing the Unit for Shipment

Please perform the steps below when packing the unit for shipment to RTX A/S for service.

Please note that damage can result from using packaging materials other than those specified. Never use styrene pellets in any shape as packaging materials since they do not adequately cushion the unit or prevent it from moving in the box. Furthermore, styrene pellets cause damage by generating static electricity. Consequently, the original packing materials should always be used when shipping the unit.

1. Fill in a note and attach it to the unit or place it visible in the shipping box. The note must include the error description provided in the RMA request.
2. Use the original packaging material. New original packaging material can be ordered at RTX A/S.
3. Seal the shipping container securely with strong nylon adhesive tape.
4. Mark the shipping container **"FRAGILE, HANDLE WITH CARE"** to ensure careful handling.
5. Retain copies of all shipping papers.

Appendix A – Example code – Simple test program (C++)

```
1   // Simple.cpp : Defines the entry point for the console application.
2   //
3
4   #include <Rtx2300PcIntf.h>
5   #include <windows.h>
6   #include <string>
7   #include <iostream>
8   #include <iomanip>
9
10  using namespace std;
11
12  static void __stdcall MailQueueHandler(Rtx2300InstanceNoType Inst, const
13  Rtx2300MailType* Mail, rsuint16 Size)
14  {
15        char buf[1000];
16        rsuint32 color;
17
18        Rtx2300Intf_DecodeMail(Mail, buf, &color);
19        cout << "Received mail: " << buf << endl;
20  }
21
22
23  int main(int argc, char* argv[])
24  {
25     string instnamebuf = "SimpleInst";
26     string portsrvnamebuf = "RtxEaiPortServer";
27     bool verinconmode = false;
28     Rtx2300InstanceNoType instno;
29
30     switch(Rtx2300Intf_Init(&instno, instnamebuf.c_str(),
31                       portsrvnamebuf.c_str(), REPS_USE_DEFAULT_UART, 1000))
32     {
33       case RTX2300_ERR_UNSUPPORTED:
34          break;
35       case RTX2300_ERR_TIMEOUT:
36          MessageBox(NULL, "No connection to target, firmware version check
37                    not completed!", "Rtx2300 warning", MB_ICONHAND | MB_OK);
38          exit(EXIT_FAILURE);
39       case RTX2300_ERR_VERSION:
40          break;
41     }
42
43     Rtx2300Intf_InstallMailHandler(instno, MailQueueHandler,
44                                                    RTX2300_PRIMITIVE_NONE);
45
46     if(instno == RTX2300_INSTNO_NONE)
47     {
```

```
48          MessageBox(NULL, "Fatal error", "Could not find or start port
49                                                  server!", MB_ICONHAND | MB_OK);
50          exit(EXIT_FAILURE);
51       }
52
53
54       // Allow the port server some time to reconfigure the transportlayer
55       Sleep(1000);
56
57       if(!Rtx2300Intf_CheckConnection(instno, 1000) != 0)
58       {
59          cout << "No connection to target!";
60          exit(EXIT_FAILURE);
61       }
62
63       {
64          char buf[100];
65          Rtx2300InitCfmType cfm = Rtx2300Init_Blocking(instno, 0);
66
67          Rtx2300Intf_DecodeErrorCode(cfm.ErrorCode, buf);
68          cout << "Errorcode: " << buf << endl;
69       }
70
71       {
72          char buf[1000];
73          rsuint32 color;
74          Rtx2300SetRelayCfmType cfm = Rtx2300SetRelay_Blocking(instno,
75                                                    RTX2300_RELAYNO_0 , true);
76
77          Rtx2300Intf_DecodeMail((Rtx2300MailType*)&cfm, buf, &color);
78          cout << "Confirm mail: " << buf << endl;
79       }
80       {
81          char buf[1000];
82          rsuint32 color;
83          Rtx2300SetRelayCfmType cfm = Rtx2300SetRelay_Blocking(instno,
84                                                     RTX2300_RELAYNO_0 , false);
85
86          Rtx2300Intf_DecodeMail((Rtx2300MailType*)&cfm, buf, &color);
87          cout << "Confirm mail: " << buf << endl;
88       }
89
90       SendRtx2300SetRelayReq(instno, 100 , false); // non-existing -> error!
91
92       char key;
93
94       cin >> key;
95    }
```

Appendix B – Example code – Multiple RTX2300 units (C++)

```
// MultiInst.cpp : Defines the entry point for the console application.
//

#include <Rtx2300PcIntf.h>
#include <windows.h>
#include <string>
#include <iostream>
#include <iomanip>
#include <conio.h>

using namespace std;

int main(int argc, char* argv[])
{
   bool verinconmode = false;
   Rtx2300InstanceNoType instno1;
   Rtx2300InstanceNoType instno2;

   switch(Rtx2300Intf_Init(&instno1, "Inst1", "PsTarget1",
                                          REPS_USE_DEFAULT_UART, 1000))
   {
      case RTX2300_ERR_UNSUPPORTED:
         break;
      case RTX2300_ERR_TIMEOUT:
         MessageBox(NULL, "No connection to target1!", "Rtx2300 warning",
                                                   MB_ICONHAND | MB_OK);
         exit(EXIT_FAILURE);
      case RTX2300_ERR_VERSION:
         break;
   }

   switch(Rtx2300Intf_Init(&instno2, "Inst2", "PsTarget2",
                                           REPS_USE_DEFAULT_UART, 1000))
   {
      case RTX2300_ERR_UNSUPPORTED:
         break;
      case RTX2300_ERR_TIMEOUT:
         MessageBox(NULL, "No connection to target 2!", "Rtx2300 warning",
                     MB_ICONHAND | MB_OK);
         exit(EXIT_FAILURE);
      case RTX2300_ERR_VERSION:
         break;
   }
```

```
47      if(instno1 == RTX2300_INSTNO_NONE)
48      {
49         MessageBox(NULL, "Fatal error", "Could not find or start port
50                     server1!", MB_ICONHAND | MB_OK);
51         exit(EXIT_FAILURE);
52      }
53
54      if(instno2 == RTX2300_INSTNO_NONE)
55      {
56         MessageBox(NULL, "Fatal error", "Could not find or start port
57                     server2!", MB_ICONHAND | MB_OK);
58         exit(EXIT_FAILURE);
59      }
60
61      // Allow the port server some time to reconfigure the transportlayer
62      Sleep(1000);
63
64      if(!Rtx2300Intf_CheckConnection(instno1, 1000) != 0)
65      {
66         cout << "No connection to target1!";
67         exit(EXIT_FAILURE);
68      }
69      if(!Rtx2300Intf_CheckConnection(instno2, 1000) != 0)
70      {
71         cout << "No connection to target2!";
72         exit(EXIT_FAILURE);
73      }
74
75      {
76         char buf[100];
77         Rtx2300InitCfmType cfm = Rtx2300Init_Blocking(instno2, 0);
78
79         Rtx2300Intf_DecodeErrorCode(cfm.ErrorCode, buf);
80         cout << "Errorcode: " << buf << endl;
81      }   {
82         char buf[100];
83         Rtx2300InitCfmType cfm = Rtx2300Init_Blocking(instno1, 0);
84
85         Rtx2300Intf_DecodeErrorCode(cfm.ErrorCode, buf);
86         cout << "Errorcode: " << buf << endl;
87      }
88
89      while(!_kbhit())
90      {
91
92         if(argc == 1)
93         {
94            cout << "Turning relay 0 on in instance 1" << endl;
95            Rtx2300SetRelay_Blocking(instno1, RTX2300_RELAYNO_0 , true);
96            Rtx2300SetRelay_Blocking(instno2, RTX2300_RELAYNO_0 , false);
97         }
98
```

```
 99         else
100         {
101            cout << "Turning relay 1 on in instance 1" << endl;
102            Rtx2300SetRelay_Blocking(instno1, RTX2300_RELAYNO_1 , true);
103            Rtx2300SetRelay_Blocking(instno2, RTX2300_RELAYNO_1 , false);
104         }
105
106         Sleep(2000);
107
108         if(argc == 1)
109         {
110            cout << "Turning relay 0 on in instance 2" << endl;
111            Rtx2300SetRelay_Blocking(instno1, RTX2300_RELAYNO_0 , false);
112            Rtx2300SetRelay_Blocking(instno2, RTX2300_RELAYNO_0 , true);
113         }
114         else
115         {
116            cout << "Turning relay 1 on in instance 2" << endl;
117            Rtx2300SetRelay_Blocking(instno1, RTX2300_RELAYNO_1 , false);
118            Rtx2300SetRelay_Blocking(instno2, RTX2300_RELAYNO_1 , true);
119         }
120
121         Sleep(2000);
122      }
123   }
```

Appendix C – Example code – Firmware update (C++)

```
1   // Demo.cpp : Defines the entry point for the console application.
2   //
3
4   #include <Rtx2300PcIntf.h>
5   #include <windows.h>
6   #include <string>
7   #include <iostream>
8   #include <iomanip>
9
10  using namespace std;
11
12  int main(int argc, char* argv[])
13  {
14     string instnamebuf = "DemoInst";
15     string portsrvnamebuf = "RtxEaiPortServer";
16     bool verinconmode = false;
17     Rtx2300InstanceNoType instno;
18
19     switch(Rtx2300Intf_Init(&instno, instnamebuf.c_str(),
20                       portsrvnamebuf.c_str(), REPS_USE_DEFAULT_UART, 1000))
21     {
22       case RTX2300_ERR_UNSUPPORTED:
23          break;
24       case RTX2300_ERR_TIMEOUT:
25          MessageBox(NULL, "No connection to target, firmware version check
26                    not completed!", "Rtx2300 warning", MB_ICONHAND | MB_OK);
27          exit(EXIT_FAILURE);
28       case RTX2300_ERR_VERSION:
29          break;
30     }
31
32     if(instno == RTX2300_INSTNO_NONE)
33     {
34        MessageBox(NULL, "Fatal error", "Could not find or start port
35                                                server!", MB_ICONHAND | MB_OK);
36        exit(EXIT_FAILURE);
37     }
38
39     // Allow the port server some time to reconfigure the transportlayer
40     Sleep(1000);
41
42     if(!Rtx2300Intf_CheckConnection(instno, 1000) != 0)
43     {
44        cout << "No connection to target!";
45        exit(EXIT_FAILURE);
46     }
```

```
47
48      {
49         Rtx2300InitCfmType m = Rtx2300Init_Blocking(instno, 0);
50
51         if(m.ErrorCode == RTX2300_ERR_VERSION)
52         {
53            verinconmode = true;
54            cout << "Version inconsistency detected!" << endl;
55            cout << "\nVersions found:\n" << endl;
56            cout << "\tDLL:         " << "v" << hex << setfill('0') << setw(4)
57                 << Rtx2300Intf_GetDllVersion() << endl;
58            cout << "\tMainboard:   " << "v" << hex << setfill('0') << setw(4)
59                 << Rtx2300GetVersion_Blocking(instno,
60                       RTX2300_FIRMWARE_TARGET).VersionInfo.VersionNo << endl;
61            {
62               Rtx2300GetVersionCfmType m = Rtx2300GetVersion_Blocking(instno,
63                                                  RTX2300_FIRMWARE_COPROCESSOR);
64               if(m.ErrorCode == RTX2300_ERR_NO_ERROR)
65               {
66                  cout << "\tCoprocessor: " << "v" << hex << setfill('0') <<
67                          setw(4) << m.VersionInfo.VersionNo << endl;
68               }
69               else
70               {
71                  cout << "\tCoprocessor: not available" << endl;
72               }
73            }
74            {
75               Rtx2300GetVersionCfmType m = Rtx2300GetVersion_Blocking(instno,
76                                                  RTX2300_FIRMWARE_POWERSUPPLY);
77               if(m.ErrorCode == RTX2300_ERR_NO_ERROR)
78               {
79                  cout << "\tPSU module:  " << "v" << hex << setfill('0') <<
80                          setw(4) << m.VersionInfo.VersionNo << endl;
81               }
82               else
83               {
84                  cout << "\tPSU module:  not available" << endl;
85               }
86            }
87         }
88         else
89         {
90            cout << "Firmware version: " << "v" << hex << setfill('0') <<
91                    setw(4) << Rtx2300Intf_GetDllVersion() << endl;
92         }
93      }
94
95      {
96         Rtx2300VersionNoType curver;
97         Rtx2300VersionNoType newver;
98
```

```
 99         if(Rtx2300Intf_CheckFwu(instno, &curver, &newver) == TRUE)
100         {
101            char answer;
102
103            cout << "\nA newer firmware exists!\nDo you want to upgrade from
104                     v" << hex << setfill('0') << setw(4) << curver << " to v"
105                 << setfill('0') << setw(4) << newver << dec << " (y/n)? ";
106            cin >> answer;
107
108            if(answer == 'y')
109            {
110               cout << "\nStarting firmware update:\n" << endl;
111               Rtx2300Intf_StartFwu(instno, argv[0]);
112               cout << "\nFirmware update failed!" << endl;
113            }
114         }
115         else
116         {
117            cout << "\nNo newer firmware found!" << endl;
118         }
119
120         return 0;
121      }
122
123      if(!verinconmode)
124      {
125         Rtx2300GetStatusCfmType m;
126
127         m = Rtx2300GetStatus_Blocking(instno);
128
129         if(m.ErrorCode != RTX2300_ERR_NO_ERROR)
130         {
131            cout << "Error requesting status!" << endl;
132            exit(EXIT_FAILURE);
133         }
134
135         if(m.Status.Bits.InitDone)
136         {
137            cout << "Status: InitDone" << endl;
138         }
139         if(m.Status.Bits.Authenticated)
140         {
141            cout << "Status: Authenticated" << endl;
142         }
143      }
144   }
```