


RTX2300

Interface Specification


RTX	Technical Documentation Rtx2300	Interface Specification		
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 1 of 54 pages

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


1	History	5
2	References	5
3	Terms and abbreviations	5
4	Introduction	6
4.1	Interface	6
4.2	RTX2300 driver software	6
4.3	Rtx2300 target software	6
5	Generic types.....	6
6	System Interface	7
6.1	Mail primitive overview	7
6.2	Confirmation mails	8
6.3	Function interface	8
7	Dll interface.....	8
8	Initializing Rtx2300	8
8.1	Version consistency	9
9	Firmware update	9
10	Using the internal pulse generator	9
10.1	Userdefined pulse generation.....	10
11	Using the input monitor.....	10
12	Interface description	11
12.1	Measuring	11
12.1.1	A/D converter.....	11
12.1.2	Audio levels	12
12.1.3	Audio distortion	12
12.2	Signal generation/processing	12
12.2.1	D/A converter.....	12
12.2.2	Audio Generator.....	13
12.2.3	PWM generator	14
12.3	Control	14
12.3.1	Interrupt inputs	14
12.3.2	RF Switch.....	16
12.3.3	Relays.....	16
12.3.4	Digital outputs.....	18
12.3.5	Digital inputs	19
12.3.6	Fixture	19
12.3.7	USB switch	23
12.3.8	UUT serial communication	24
12.3.9	SCB expansion bus	24
12.4	Power Supply.....	25
12.4.1	Switching power supply on/off	25
12.4.2	Selecting internal/external power supply	26
12.4.3	Output voltage	27
12.4.4	Output current	27
12.4.5	Get peak current measurement	29

	Technical Documentation			Interface Specification
	Rtx2300			
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 2 of 54 pages

12.4.6	Power supply indications	29
12.5	Pulse generator.....	30
12.6	Input monitor	31
12.7	General indications	32
12.8	House keeping	32
12.8.1	Initializing the system: RTX2300_INIT	32
12.8.2	Resetting the system: RTX2300_RESET.....	33
12.8.3	Requesting system status: RTX2300_GET_STATUS	33
12.8.4	Requesting firmware version.....	33
12.8.5	Requesting main firmware information.....	34
12.8.6	Getting/setting fixture type.....	Error! Bookmark not defined.
12.8.7	Getting/setting serial number	34
12.8.8	Getting the current temperature.....	36
12.8.9	User data handling.....	36
12.8.10	Feature configuration	37
12.8.11	Access mode.....	38
12.9	Misc commands.....	39
13	Debug interface	40
13.1	High level debug interface	40
13.1.1	Attenuator driver	40
13.1.2	ADC task driver	40
13.1.3	DAC task driver	41
13.2	Low level debug interface.....	41
13.2.1	IO expander.....	41
14	Types.....	42
14.1	Rtx2300SystemInfoType	42
14.2	Rtx2300PulseModeType	43
14.3	Rtx2300PulseDestinationType	43
14.4	Rtx2300InputMonitorSourceType	44
14.5	Rtx2300AdcCfgType.....	44
14.6	Rtx2300DacChannelType.....	46
14.1	Rtx2300AudioLvlMeasureModeType	46
14.2	Rtx2300DistortionMeasureModeType	46
14.3	Rtx2300CurrentRangeType	46
14.4	Rtx2300PeakCurrentMeasuresureTimeType	46
14.5	Rtx2300AudioInputType	47
14.6	Audio attenuator constants.....	47
14.7	Rtx2300AudioGenChannelType	47
14.8	Rtx2300InterruptMaskType.....	47
14.9	Rtx2300InterruptNoType.....	48
14.10	Rtx2300RelayNoType	48
14.11	Rtx2300RelayMaskType	48
14.12	Rtx2300OutputNoType	48
14.13	Rtx2300OutputMaskType	49
14.14	Rtx2300InputNoType	49
14.15	Rtx2300InputMaskType	50
14.16	Rtx2300FrontLedNoType.....	50
14.17	Rtx2300FrontLedMaskType.....	50
14.18	Rtx2300FixtureControlNoType.....	50
14.19	Rtx2300FixtureControlMaskType.....	51
14.20	Rtx2300UserDataTypes	51
14.21	User data constants	51
14.22	Rtx2300UsbControlNoType	51
14.23	Rtx2300UutSercomControlType	51
14.24	Rtx2300FeatureType	51
14.25	Rtx2300StatusType.....	52
14.26	Pulse pattern constants.....	52

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 3 of 54 pages	

14.27 Rtx2300PulsePatternValueType	52
14.28 Rtx2300PulsePatternDefinitionType	52
14.29 Rtx2300ExpanderNoType	53
14.30 Rtx2300CapAddrType.....	53
14.31 Rtx2300CapDeviceAddrType	53
14.32 Rtx2300ScbBusCfgType	53

	Technical Documentation Rtx2300	Interface Specification		
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 4 of 54 pages

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

1 History

Date	Initials	Rev.	Description
01/02/2010	MHP	1.0	Initial version
10/03/2010	MHP	1.1	Tons of new members.
28/04/2010	MHP	1.2	Dbg interface updated.
09/08/2010	MHP	1.3	Agilent autodetection removed from RF switch API.
13/09/2010	MHP	1.4	Added API to set/get the serial number
24/11/2010	MHP	1.5	Common types moved to separate document.
17/8/2012	MHP	1.6	Added TestCounter feature Removed Get/SetId commands. Primitives 515d to 5160 are now unused! Added sense interrupts.

2 References

- Rtx2300 Configuration.pdf (still in progress)
- Rtx2300 Common Interface.pdf

3 Terms and abbreviations

Term	Description
Task	A self-contained major software component in the RTX standard software environment
Master	The software and or system controlling the Rtx2300 hardware, typically an application running on a PC.
Target	The RTX2300 box including the board containing the circuitry and the software running it.
Firmware	The software running in the target.
FWU	FirmWare Update. The process of updating all the softwares in the system to a newer version.
Request	A command sent to the Rtx2300, e.g. measure voltage.
Confirm	The result of the request, returned by Rtx2300.

4 Introduction

This document describes the SW interface between master PC running the Rtx2300 driver software and the Rtx2300 hardware. An overview of the system is shown in Figure 1.

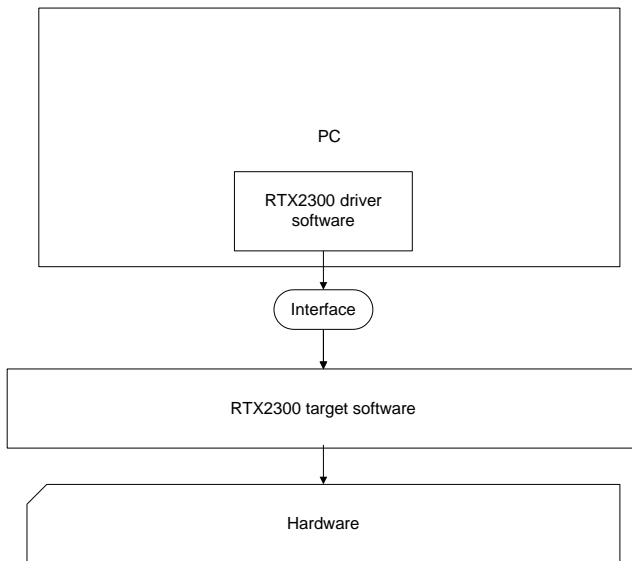


Figure 1

4.1 Interface

This is the interface described in this document.

4.2 RTX2300 driver software

This is the PC software that controls the Rtx2300 by using the interface described in this document.

4.3 Rtx2300 target software

This software is described in the *Rtx2300 Software Design* document.

5 Generic types

The Rtx2300 interface uses RTX Telecom standard platform independent types. These types must be defined in accordance with the platform used.

Type name	Typical definition	Description
rsuint8	typedef unsigned char rsuint8;	unsigned 8 bit
rsint8	typedef signed char rsint8;	signed 8 bit
rsuint16	typedef unsigned short rsuint16;	unsigned 16 bit
rsint16	typedef signed short rsint16;	signed 16 bit
rsuint32	typedef unsigned long rsuint32;	unsigned 32 bit
rsint32	typedef signed long rsint32;	signed 32 bit
rsbitfield	typedef unsigned char rsbitfield;	bitfield designator

	Technical Documentation			Interface Specification
	Rtx2300			
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 6 of 54 pages

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

6 System Interface

This section describes the Rtx2300 interface, which is mail based.

6.1 Mail primitive overview

The Rtx2300 is controlled by use of mails. A mail is, in this context, an instance of a set of data types, one type for each request and confirmation. Each type contains a field called the primitive, which identifies the type, e.g. a `Rtx2300InitReqType` will always contain a `RTX2300_INIT_REQ` primitive. The primitive allows the receiver of the mail to identify the type. Request mails are recognized by the `_REQ` postfix in the mail primitive while confirms uses a `_CFM` postfix.

Three types of mails exist: requests, confirmations and indications. Requests contain commands to the Rtx2300 while confirmations (or short: confirms) contain the result of the command. Indications are sent uninvited from Rtx2300 to the master whenever something happens that the master needs to know, e.g. a key is pressed. The major part of this document discusses the requests, confirms and indications sent to and from the Rtx2300. The Rtx2300 header file is generated from this document as part of the software build process. The interface definitions generated by each mail description is described in this fictive example:

Description:	Measure an AC voltage	
Primitive:	RTX2300_MEASURE_AC_REQ	
Parameters:		
Type	Name	Description
Rtx2300_MeasModeType	MeasurementMode	The mode used for measuring
Rtx2300_ChannelType	Channel	The channel used for measuring

Description:	An AC measurement has ended	
Primitive:	RTX2300_MEASURE_AC_CFM	
Parameters:		
Type	Name	Description
RsuInt16	Value	The value measured
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR: the measurement was done with no errors RTX2300_ERR_RANGE: the measured voltage was too large.


This example specifies the request used to start an AC voltage measurement, and the confirmation returned by Rtx2300 when the measurement has finished. The requests and confirms uses the same notation in the document and are also treated equally when the header file contents are generated. The types used in the mails are documented in section 15 and proper type definitions are also generated in the header file for the types.

The header file contains an *enum* definition named `Rtx2300PrimitiveType`, which defines all the primitives used by Rtx2300. Each request and confirm generates an entry in the enum:

```
typedef enum
{
    .....
    RTX2300_MEASURE_AC_REQ,
    RTX2300_MEASURE_AC_CFM,
    .....
} Rtx2300PrimitiveType;
```

The types used by the request and confirm mails are also defined:

```
/* Measure an AC voltage */
typedef struct
{
    Rtx2300PrimitiveType Primitive;
```

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
				Page: 7 of 54 pages

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

```

    Rtx2300_MeasModeType MeasurementMode;
    Rtx2300_ChannelType Channel;
} Rtx2300MeasureAcReqType;

/* An AC measurement has ended */
typedef struct
{
    Rtx2300PrimitiveType Primitive;
    Rsuint16 Value;
    Rtx2300ErrorType ErrorCode;
} Rtx2300MeasureAcCfmType;

```

6.2 Confirmation mails

Confirmation mails are sent from the task to the master to notify the master that a request has been executed.

All confirmations may return the following error codes:

Error code	Description
RTX2300_ERR_NO_ERROR	Request completed successfully
RTX2300_ERR_UNSUPPORTED	This request is not supported.
RTX2300_ERR_BUSY	The request cannot be accepted because the previous one has not completed or the Rtx2300 system is otherwise not able to handle the request right now.
RTX2300_ERR_VERSION	Version inconsistency detected. All version must be the same!

Please note: these error codes are not described in the confirm specifications below, as they are common to all confirms.

6.3 Function interface

The Rtx2300 interface is mailbased but a functioninterface also exists. This interface provides a function for every mail that can be sent from the PC to the Rtx2300 system. These functions, which are all listed in the IRtx2300.h headerfile, are all named after the mail they send. So the RTX2300_MEASURE_AC_REQ mail is sent using the *SendRtx2300MeasureAcReq* function.

A second function interface also exists. This is called the blocking function interface, and it binds together the sending of the request with the reception of the confirmation, i.e. the functions accept data for the request as parameters and return the confirmation mail. The disadvantage is that the function does no return until a confirmation has been received or a timeout has occurred. This is why it is called the blocking interface. All blocking functions have the word *Blocking* appended to the function name.


7 DLL interface

The main interface to the system is the function interfaces described above, but an interface to manage the DLL also exists. It is decribed in the *Rtx2300PcIntf.h* headerfiler. All functions in this interface start with the word *Rtx2300Intf_*

The DLL interface requires an instance number to identify each user of the DLL. This instance number is attached to all mails to and from the Rtx2300 system.

8 Initializing Rtx2300

The Rtx2300 must be initialized properly and this requires calling the *SendRtx2300InitReq* function (or one of its blocking variants) . This initialization includes testing the required hardware, checking the feature configuration to determine which features are enabled and handling authentication to ensure the firmware is genuine. Please note that this must be done exactly one time, and before sending any other requests to Rtx2300! Until the Rtx2300 has finished it's initialization and returned a confirmation, all other requests (except for a status request) will result in a RTX2300_ERR_UNSUPPORTED confirmation!

	Technical Documentation Rtx2300			Interface Specification	
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 8 of 54 pages

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

8.1 Version consistency

The Rtx2300 system contains several different firmwares and applications. These pieces of software can only communicate if they agree on the format of the data they exchange. To ensure this it is checked during startup that all the softwares contain identical version numbers. If not the confirmation returned by the call to *SendRtx2300InitReq()* contains the errorcode `RTX2300_ERR_VERSION`. Please note that the system cannot be used in this state, and all requests will return errorcode `RTX2300_ERR_VERSION`.

9 Firmware update

The firmware in the system should be updated whenever a newer version exists. An update consists of a number of steps:

- 1) Get a new firmware. This is usually done by downloading it from the RTX Telecom homepage.
- 2) Install it on the PC.
- 3) The application that uses the system must call *SendRtx2300GetFwuStatusReq* or one of it's blocking variants. The information returned tells if the system has found a new firmware.
- 4) If the new firmware should be installed the application must call *SendRtx2300StartFwuReq()*. Please note that the *firmware* parameter is not used and may contain any value. When the firmwares has been updated the DLL will terminate the application.
- 5) If the application is not running from the Rtx2300 system install folder, usually *c:\Program Files\Rtx Telecom\Rtx2300*, the DLL and it's associated files must be copied manually to application folder before restarting the application.


10 Using the internal pulse generator

The Rtx2300 system contains a pulse generator that can be used to control various outputs. The pulse generator can be used by the master to let output ports pulse autonomously, i.e. without direct instruction from the master. Supported behaviour ranges from simply keeping the output active for X miliseconds to flashing infinitely using a user-defined frequency and dutycycle.

The pulse generation is based on an internal 10ms timebase. All timing is defined in multiplums of 10ms, known as a *tick*. Please note that using an internal timebase with 10ms resolution does not guarantee 10ms precision of the actual output signal! The timing may vary with the load of the system and other factors.

The supported pulse modes are defined in the `Rtx2300PulseModeType`:

Pulse type	Description
<code>RTX2300_PULSEMODE_OFF,</code>	Output is constantly off. This effectively disables the pulse generator for that output.
<code>RTX2300_PULSEMODE_PULSE_SHORT,</code>	Output is on for 10 ms then goes off
<code>RTX2300_PULSEMODE_PULSE_LONG,</code>	Output is on for 1000 ms then goes off
<code>RTX2300_PULSEMODE_FLASH_SLOW,</code>	Output flashes constantly with 0.5 HZ and 50% dutycycle, i.e. 1000ms on/1000ms off
<code>RTX2300_PULSEMODE_FLASH_MEDIUM,</code>	Output flashes constantly with 1 HZ and 50% dutycycle, i.e. 500ms on/500ms off
<code>RTX2300_PULSEMODE_FLASH_QUICK,</code>	Output flashes constantly with 2.5 HZ and 50% dutycycle, i.e. 200ms on/200ms off
<code>RTX2300_PULSEMODE_FLASH_LONG_SLOW,</code>	Output flashes constantly with 0.5 HZ and 90% dutycycle, i.e. 1800ms on/200ms off
<code>RTX2300_PULSEMODE_FLASH_LONG_MEDIUM,</code>	Output flashes constantly with 1 HZ and 90% dutycycle, i.e. 900ms on/100ms off
<code>RTX2300_PULSEMODE_FLASH_LONG_QUICK,</code>	Output flashes constantly with 2.5 HZ and 90% dutycycle, i.e. 360ms on/40ms off
<code>RTX2300_PULSEMODE_FLASH_SHORT_SLOW,</code>	Output flashes constantly with 0.5 HZ and 10% dutycycle, i.e. 200ms on/1800ms off
<code>RTX2300_PULSEMODE_FLASH_SHORT_MEDIUM,</code>	Output flashes constantly with 1 HZ and 10% dutycycle, i.e. 100ms on/900ms off
<code>RTX2300_PULSEMODE_FLASH_SHORT_QUICK,</code>	Output flashes constantly with 2.5 HZ and 10% dutycycle, i.e. 40ms on/360ms off
<code>RTX2300_PULSEMODE_USER_DEFINED_0,</code>	User defined pulse generation, see below.
<code>RTX2300_PULSEMODE_USER_DEFINED_1,</code>	User defined pulse generation, see below.

	Technical Documentation			Interface Specification
	Rtx2300			
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 9 of 54 pages

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

RTX2300_PULSEMODE_USER_DEFINED_2,	User defined pulse generation, see below.
RTX2300_PULSEMODE_USER_DEFINED_3,	User defined pulse generation, see below.

The supported output destinations are defined in the Rtx2300PulseDestinationType:

RTX2300_PULSEDEST_DIGOUT_0 - 15	Digital out put 0 - 15
RTX2300_PULSEDEST_FRONT_LED_0 - 2	Front LED 0 - 2
RTX2300_PULSEDEST_FIXTURE_CONTROL_0 - 3	Fixture control 0 - 3
RTX2300_PULSEDEST_EXPANSION_CONTROL_0 - 3	Expansion control 0- 3

10.1 Userdefined pulse generation

Four pulse modes use a userdefined pulse generation pattern. Each mode may be used by one output at a time. It is possible to redefine all other modes as well, if needed.

A userdefined mode consists of 16 values, and each value may hold between 1 and 0x3FFF ticks. The two most significant bits (i.e. bit 15 and 14) are control bits:

- Bit15 controls the state of the output. If 1 the output will be set active for the time specified by the value, if 0 the output will be set to the inactive state.
- Bit14 controls looping. If 1 control is returned to the first value after the time specified by the value has elapsed.

The pattern ends when a value of zero is reached or the last value in the set has been processed.

The sets are programmed using the API, see 13.4

11 Using the input monitor

The Rtx2300 system contains an input monitor that can be used to monitor various inputs. The monitor can be used by the master to let state changes on input ports being reported directly to the master, i.e. without master having to continuously request the state of the port.

The input monitor is based on an internal 10ms timebase. The ports are thus checked every 10ms. The timing may vary with the load of the system and other factors. Whenever an input changes state the monitor checks if the change is to be reported. If so an RTX2300_INPUTMONITOR_IND indication is sent to the master.

The supported state change modes are defined in the Rtx2300StateChangeType:


Pulse type	Description
None	No changes are reported, i.e. monitoring is suspended.
Activated	Changes from inactive to active are reported.
Deactivated	Changes from active to inactive are reported.
Both	Changes in both directions are reported.

12 Interrupt sense control

The interrupt sense can be used to generate indications when a *Sense* interrupt input change state. It allows detection of input state change, the length of a pulse on an input, or the time between activities on two different inputs. Only *Sense* inputs *SENSE_IN_0* and *SENSE_IN_1* can be used. The inputs can be configured independently.

Five different modes are available:

Mode	Indication trigger(s)
Rising edge	Rising edge detected on Sense input
Falling edge	Falling edge detected on Sense input
Rising edge / falling edge	1. indication: rising edge detected on Sense input

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 10 of 54 pages	

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

	2. indication: falling edge detected on Sense input
Falling edge /rising edge	1. indication: falling edge detected on Sense input 2. indication: rising edge detected on Sense input
Disabled	

It is possible to configure sensing to *single shot* mode, in which sensing is automatically disabled when all indications specified in the mode has been sent, or *continuous mode*, in which sensing continues until manually disabled by the master. Caution: using continuous mode with a high frequency input may cause internal overloading and possibly instability!

Warning: allthough it is possible to use interrupt sensing on both inputs simultaneously, great care should be taken. If both inputs generate an interrupt at the exact same time only one will be detected.

13 Interface description

The following lists the primitives and mailtypes used to communicate with the Rtx2300.

13.1 Measuring


13.1.1 A/D converter

Description:	Measure DC level using A/D converter.	
Primitive:	RTX2300_GET_ADC_REQ	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300AdcCfgType	Cfg	If channel 0 is specified the channel is automatically connected to ADC_0_IN if in use by the acquisition unit. The input will be restored when done.

Description:	The ADC measurement has finished	
Primitive:	RTX2300_GET_ADC_CFM = 0x5002	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: illegal parameter(s) found
Rtx2300SignalLvlType	Value	The measured voltage. Unit: mV

Description:	Select ADC0 signal route: to internal ADC or external aquisition unit. If a route is not explicitly set using this mail the internal ADC is selected as default.	
Primitive:	RTX2300_SET_ADC0_ROUTE_REQ = 0x5003	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	UseAdc	If true the ADC_0_IN signal is routed to the internal ADC. If not it is routed to the acquisition unit.

Description:	The route was selected	
Primitive:	RTX2300_SET_ADC0_ROUTE_CFM = 0x5004	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 11 of 54 pages	

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

13.1.2 Audio levels

Description:	Measure audio level using internal DSP.	
Primitive:	RTX2300_GET_AUDIOLVL_REQ = 0x5010	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300AudioInputType	Input	
Rtx2300AudioAttenuationType	Attenuation	Attenuation in dB.
Rtx2300AudioLvlMeasureModeType	Mode	

Description:	The measurement has finished	
Primitive:	RTX2300_GET_AUDIOLVL_CFM = 0x5011	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300SignalLvlType	Value	The integer part of the measured voltage. Unit: mV (see Mode parameter)
Rtx2300SignalLvlType	Value_Fraction	The fractional part of the actual output voltage. The value is the number of uV, and is always a single-digit integer.
Rtx2300SignalLvlType	PeakValue	The measured peak voltage. Unit: mVp. The voltage represents the targets input voltage, positive or negative, measured over a period of appr. 1 second. This voltage may help avoid overloading the input.
rsbool	Overload	If true: the input has been overloaded and the measurement is not valid. This flag is reset at the start of a new 1 second period.

13.1.3 Audio distortion


Description:	Measure audio distortion using internal DSP.	
Primitive:	RTX2300_GET_DISTORTION_REQ = 0x5012	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300AudioInputType	Input	
Rtx2300AudioAttenuationType	Attenuation	Attenuation in dB.
Rtx2300DistortionMeasureModeType	Mode	

Description:	The measurement has finished	
Primitive:	RTX2300_GET_DISTORTION_CFM = 0x5013	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300DistortionLvlType	Distortion	The measured distortion. Unit: o/oo (per mille)
Rtx2300SignalLvlType	PeakValue	The measured peak voltage. Unit: mVp. The voltage represents the targets input voltage, positive or negative, measured over a period of appr. 1 second. This voltage may help avoid overloading the input.
rsbool	Overload	If true: the input has been overloaded and the measurement is not valid. This flag is reset at the start of a new 1 second period.

13.2 Signal generation/processing

13.2.1 D/A converter

Description:	Set DC level using the D/A converter.
---------------------	---------------------------------------

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
				Page: 12 of 54 pages

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

Primitive:	RTX2300_SET_DAC_REQ = 0x5020		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300DacChannelType	Channel		
Rtx2300SignalLvlType	Value	The output voltage, units = mV	

Description:	The DC level has been set		
Primitive:	RTX2300_SET_DAC_CFM = 0x5021		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: illegal parameter(s) found	


Description:	Get the current setting of the D/A converter.		
Primitive:	RTX2300_GET_DAC_REQ = 0x5022		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300DacChannelType	Channel		

Description:	The current D/A setting		
Primitive:	RTX2300_GET_DAC_CFM = 0x5023		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR	
Rtx2300SignalLvlType	Value	The output voltage, units = mV	

13.2.2 Audio Generator

Description:	Set up the audio generator.		
Primitive:	RTX2300_SET_AUDIO_GENERATOR_REQ = 0x5030		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300AudioGenChannelType	Channel	The channel to generate the audio on.	
Rtx2300FrequencyType	Frequency	The signal frequency. Unit: Hz	
Rtx2300SignalLvlType	Level	The output voltage. Unit: mVrms	

Description:	The generator has been set up		
Primitive:	RTX2300_SET_AUDIO_GENERATOR_CFM = 0x5031		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: a parameter was outside the legal range.	

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
Page: 13 of 54 pages				

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

13.2.3 PWM generator

Description:	Set up the PWM generator	
Primitive:	RTX2300_SET_PWM_GENERATOR_REQ = 0x5040	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsuint8	Ratio	Valid range: 0-100.
Rtx2300FrequencyType	Frequency	Valid range: 20Hz – 25kHz. Unit: Hz. This range is guaranteed for all ratios. Please note that , although the resulting frequency will be set as close as possible to the specified frequency, it may, especially at high frequencies, be off by several percent.

Description:	The PWM generator has been set up	
Primitive:	RTX2300_SET_PWM_GENERATOR_CFM = 0x5041	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: a parameter was outside the legal range.

Description:	Get the PWM generator setup	
Primitive:	RTX2300_GET_PWM_GENERATOR_REQ = 0x5042	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.


Description:	Get the PWM generator setup	
Primitive:	RTX2300_GET_PWM_GENERATOR_CFM = 0x5043	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsuint8	Ratio	
Rtx2300FrequencyType	Frequency	Unit: Hz.

13.3 Control

13.3.1 Interrupt inputs

Description:	Get the input state of an interrupt input	
Primitive:	RTX2300_GET_INTERRUPT_INPUT_REQ = 0x5050	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300InterruptNoType	InterruptNo	

Description:		
Primitive:	RTX2300_GET_INTERRUPT_INPUT_CFM = 0x5051	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	Active	True = input active

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
Page: 14 of 54 pages				

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

Description:	Get the input state of multiple interrupt inputs	
Primitive:	RTX2300_GET_INTERRUPT_INPUTS_REQ = 0x5052	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300InterruptMaskType	Mask	

Description:		
Primitive:	RTX2300_GET_INTERRUPT_INPUTS_CFM = 0x5053	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300InterruptMaskType	Values	The current values of the inputs


13.3.1.1 Interrupt sensing

Description:	Set up interrupt sense	
Primitive:	RTX2300_SET_INTERRUPT_SENSE_REQ = 0x5054	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300InterruptNoType	Source	The interrupt input to set
Rtx2300InterruptSenseModeType	Mode	The sense mode for interrupt input
rsbool	Continuous	0: single shot mode 1: continuous mode

Description:		
Primitive:	RTX2300_SET_INTERRUPT_SENSE_CFM = 0x5055	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: illegal parameter. RTX2300_ERR_BUSY: interrupt sensing is already in use.

Description:	Get interrupt sense setup	
Primitive:	RTX2300_GET_INTERRUPT_SENSE_REQ = 0x5056	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300InterruptNoType	Source	The interrupt input to set

Description:		
Primitive:	RTX2300_GET_INTERRUPT_SENSE_CFM = 0x5057	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: illegal parameter. RTX2300_ERR_BUSY: interrupt sensing is already in use.
Rtx2300InterruptSenseModeType	Mode	The sense mode for interrupt input
rsbool	Continuous	0: single shot mode 1: continuous mode

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
Page: 15 of 54 pages				

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

Description:	Interrupt sense indication: the Rtx2300 interrupt sensing has detected a state change on the specified input.		
Primitive:	RTX2300_INTERRUPT_SENSE_IND = 0x5058		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300InterruptNoType	Source	The sense input causing the indication	
rsbool	Rising	0: a rising edge was detected 1: a falling edge was detected	
rsuint32	TimeStamp	The value of the internal system timer when the edge was detected. The resolution is 1 ms.	

13.3.2 RF Switch

Description:	Setup the RF switch.		
Primitive:	RTX2300_SET_RF_SWITCH_REQ = 0x5060		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
rsuint16	Setting	The new setting of the RF switch.	
rsbool	DirectMode	If true the four least significant bits of <i>Setting</i> parameter are used to control the four data lines in the RF Switch bus. Bit0: data, bit1: oe, bit2:clk and bit3: strobe.	


Description:	The RF switch has been set up		
Primitive:	RTX2300_SET_RF_SWITCH_CFM = 0x5061		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR	

Description:	Get the RF switch setting		
Primitive:	RTX2300_GET_RF_SWITCH_REQ = 0x5062		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	

Description:			
Primitive:	RTX2300_GET_RF_SWITCH_CFM = 0x5063		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR	
rsuint16	Setting	The current setting of the RF switch	

13.3.3 Relays

Description:	Setup a relay.		
Primitive:	RTX2300_SET_RELAY_REQ = 0x5190		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300RelayNoType	No	The relay to set.	
rsbool	Active	True = relay active	

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
Page: 16 of 54 pages				

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

Description:	The relay has been set up	
Primitive:	RTX2300_SET_RELAY_CFM = 0x5191	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get a relay setting	
Primitive:	RTX2300_GET_RELAY_REQ = 0x5192	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300RelayNoType	No	The relay to set.


Description:		
Primitive:	RTX2300_GET_RELAY_CFM = 0x5193	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	Active	True = relay active

Description:	Setup the relays.	
Primitive:	RTX2300_SET_RELAYS_REQ = 0x5194	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300RelayMaskType	Mask	The relays to set. Multiple relay numbers may be OR'ed to control more than one relay.
Rtx2300RelayMaskType	State	The state to set. Multiple relays may be OR'ed to control more than one relay.

Description:	The relays has been set up	
Primitive:	RTX2300_SET_RELAYS_CFM = 0x5195	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get the relay settings	
Primitive:	RTX2300_GET_RELAYS_REQ = 0x5196	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300RelayMaskType	Mask	The relays to get. Multiple relay numbers may be OR'ed to control more than one relay.

Description:		
Primitive:	RTX2300_GET_RELAYS_CFM = 0x5197	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

	Technical Documentation			Interface Specification
	Rtx2300			
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 17 of 54 pages

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

Rtx2300RelayMaskType	Values	The current setting of the relays
----------------------	--------	-----------------------------------

13.3.4 Digital outputs

Description:	Setup a digital output.	
Primitive:	RTX2300_SET_OUTPUT_REQ = 0x51A0	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300OutputNoType	No	The outputs to set.
rsbool	Active	True = output active

Description:	The output has been set up	
Primitive:	RTX2300_SET_OUTPUT_CFM = 0x51A1	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR


Description:	Get the state of a digital output	
Primitive:	RTX2300_GET_OUTPUT_REQ = 0x51A2	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300OutputNoType	No	The output to get.

Description:		
Primitive:	RTX2300_GET_OUTPUT_CFM = 0x51A3	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	Active	True = output active

Description:	Setup a group of digital outputs.	
Primitive:	RTX2300_SET_OUTPUTS_REQ = 0x51A4	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300OutputMaskType	Mask	The outputs to set. Multiple outputs numbers may be OR'ed to control more than one output.
Rtx2300OutputMaskType	State	The state to set. Multiple outputs may be OR'ed to control more than one output.

Description:	The outputs has been set up	
Primitive:	RTX2300_SET_OUTPUTS_CFM = 0x51A5	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get the state of a group of digital outputs.	
---------------------	--	--

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 18 of 54 pages	

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

Primitive:	RTX2300_GET_OUTPUTS_REQ = 0x51A6		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300OutputMaskType	Mask	The outputs to get. Multiple output numbers may be OR'ed to control more than one output.

Description:			
Primitive:	RTX2300_GET_OUTPUTS_CFM = 0x51A7		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
	Rtx2300OutputMaskType	Values	The current setting of the selected outputs

13.3.5 Digital inputs

Description:	Get the current state of a digital input.		
Primitive:	RTX2300_GET_INPUT_REQ = 0x5090		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300InputNoType	No	The input to get.


Description:			
Primitive:	RTX2300_GET_INPUT_CFM = 0x5091		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
	rsbool	Active	True = input active

Description:	Get the current state of a group of digital inputs.		
Primitive:	RTX2300_GET_INPUTS_REQ = 0x5092		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300InputMaskType	Mask	The inputs to get. Multiple input numbers may be OR'ed to get more than one input.

Description:			
Primitive:	RTX2300_GET_INPUTS_CFM = 0x5093		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
	Rtx2300InputMaskType	Values	The current values of the inputs

13.3.6 Fixture

Description:	Set the air valve.		
Primitive:	RTX2300_SET_AIRVALVE_REQ = 0x50A0		
Parameters:			

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
				Page: 19 of 54 pages

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

Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Active	If true the valve is activated

Description:	The airvalve has been set up	
Primitive:	RTX2300_SET_AIRVALVE_CFM = 0x50A1	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get the airvalve setting	
Primitive:	RTX2300_GET_AIRVALVE_REQ = 0x50A2	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.


Description:		
Primitive:	RTX2300_GET_AIRVALVE_CFM = 0x50A3	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	Active	The current setting of the airvalve

Description:	Set the magnet. Beware: activating the magnet will automatically increment the mainboard test counter, see 13.8.13	
Primitive:	RTX2300_SET_MAGNET_REQ = 0x50A4	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Active	If true the magnet is activated
rsbool	AutoDeactivate	If true the magnet will automatically deactivate after a fixed period of time.

Description:	The magnet has been set up	
Primitive:	RTX2300_SET_MAGNET_CFM = 0x50A5	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get the magnet setting	
Primitive:	RTX2300_GET_MAGNET_REQ = 0x50A6	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

Description:		
Primitive:	RTX2300_GET_MAGNET_CFM = 0x50A7	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	Active	The current setting of the magnet

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2

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

Description:	Set a front LED.		
Primitive:	RTX2300_SET_FRONT_LED_REQ = 0x50A8		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300FrontLedNoType	No	The LED to set.
	rsbool	Active	The new state of the selected front LED.

Description:	The LED has been set		
Primitive:	RTX2300_SET_FRONT_LED_CFM = 0x50A9		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR


Description:	Get the front LED setting		
Primitive:	RTX2300_GET_FRONT_LED_REQ = 0x50AA		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300FrontLedNoType	No	The LED number to get.

Description:			
Primitive:	RTX2300_GET_FRONT_LED_CFM = 0x50AB		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
	rsbool	Active	The current state of the front LED

Description:	Set the front LED's.		
Primitive:	RTX2300_SET_FRONT_LEDS_REQ = 0x50AC		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300FrontLedMaskType	Mask	The LED's to set. Multiple LED numbers may be OR'ed to control more than one LED.
	Rtx2300FrontLedMaskType	State	The new state of the selected front LED's. Multiple LED numbers may be OR'ed to control more than one LED.

Description:	The LED has been set		
Primitive:	RTX2300_SET_FRONT_LEDS_CFM = 0x50AD		
Parameters:			
	Type	Name	Description
	Rtx2300InstanceNoType	InstNo	The instance number.
	Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get the front LED setting		
Primitive:	RTX2300_GET_FRONT_LEDS_REQ = 0x50AE		

	Technical Documentation			Interface Specification
	Rtx2300			
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 21 of 54 pages

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

Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300FrontLedMaskType	Mask	The LED numbers to get.

Description:		
Primitive: RTX2300_GET_FRONT_LEDS_CFM = 0x50AF		
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300FrontLedMaskType	State	The current state of the front LEDs


Description: Set the fixture control output.		
Primitive: RTX2300_SET_FIXTURE_CONTROL_REQ = 0x50B0		
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300FixtureControlNoType	No	The fixture control output number to set.
rsbool	Active	The new state.

Description: The fixture control output has been set		
Primitive: RTX2300_SET_FIXTURE_CONTROL_CFM = 0x50B1		
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_BUSY: the state is already in use

Description: Get the fixture control output setting		
Primitive: RTX2300_GET_FIXTURE_CONTROL_REQ = 0x50B2		
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300FixtureControlNoType	No	The output number to get.

Description:		
Primitive: RTX2300_GET_FIXTURE_CONTROL_CFM = 0x50B3		
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	Active	The current setting of the fixture output

Description: Set the fixture control outputs.		
Primitive: RTX2300_SET_FIXTURE_CONTROLS_REQ = 0x50B4		
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300FixtureControlMaskType	Mask	The fixture control outputs number to set.
Rtx2300FixtureControlMaskType	State	The new states. This will affect all outputs specified in the Mask argument.

	Technical Documentation Rtx2300			Interface Specification	
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 22 of 54 pages

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

Description:	The fixture control output has been set	
Primitive:	RTX2300_SET_FIXTURE_CONTROLS_CFM = 0x50B5	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_BUSY: the state is already in use

Description:	Get the fixture control outputs settings	
Primitive:	RTX2300_GET_FIXTURE_CONTROLS_REQ = 0x50B6	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300FixtureControlMaskType	Mask	The output numbers to get.

Description:		
Primitive:	RTX2300_GET_FIXTURE_CONTROLS_CFM = 0x50B7	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300FixtureControlMaskType	State	The current setting of the fixture outputs


13.3.7 USB switch

Description:	Set USB switches	
Primitive:	RTX2300_SET_USB_CONTROL_REQ = 0x50C0	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300UsbControlNoType	UsbNo	The USB switch to set.
rsbool	Active	True = USB active

Description:	The USB switch has been set	
Primitive:	RTX2300_SET_USB_CONTROL_CFM = 0x50C1	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_BUSY: the state is already in use

Description:	Get the USB switch setting	
Primitive:	RTX2300_GET_USB_CONTROL_REQ = 0x50C2	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300UsbControlNoType	UsbNo	The USB switch to get.

Description:		
Primitive:	RTX2300_GET_USB_CONTROL_CFM = 0x50C3	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	Active	True = USB active

	Technical Documentation Rtx2300		Interface Specification	
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2

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

13.3.8 UUT serial communication

Description:	Set up the UUT serial communication. The USB channel 6 is used, which is shared with the CCB connector. Disabling the UUT serial communication is equal to enabling the USB channel 6, and vice versa!	
Primitive:	RTX2300_SET_UUT_SERCOM_REQ = 0x50D0	
Parameters:		
	Type	Name
	Rtx2300InstanceNoType	InstNo
	Rtx2300UutSercomControlType	Mode
		Description
		The instance number.
		The required mode.

Description:	The UUT sercom has been set	
Primitive:	RTX2300_SET_UUT_SERCOM_CFM = 0x50D1	
Parameters:		
	Type	Name
	Rtx2300InstanceNoType	InstNo
	Rtx2300ErrorType	ErrorCode
		Description
		The instance number.
		RTX2300_ERR_NO_ERROR

Description:	Get the UUT sercom setting	
Primitive:	RTX2300_GET_UUT_SERCOM_REQ = 0x50D2	
Parameters:		
	Type	Name
	Rtx2300InstanceNoType	InstNo
		Description
		The instance number.

Description:		
Primitive:	RTX2300_GET_UUT_SERCOM_CFM = 0x50D3	
Parameters:		
	Type	Name
	Rtx2300InstanceNoType	InstNo
	Rtx2300ErrorType	ErrorCode
	Rtx2300UutSercomControlType	Mode
		Description
		The instance number.
		RTX2300_ERR_NO_ERROR
		The current mode.

13.3.9 SCB expansion bus

The SCB connector contains a serial bus that is controlled by the Rtx2300 firmware. This bus may be connected to devices in the fixture, e.g. output port expanders, shift registers, D/A converters, etc. It is a read/write bus, fully configurable and includes *Strobe* and *Output Enable* lines, besides the clock and data lines. It supports data words from 1 to 32 bits. Figure 2 shows a typical output pattern from the bus:

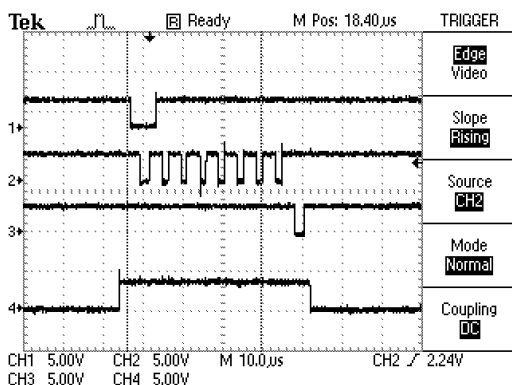


Figure 2

The four lines may also be used as individual IO ports, see the Rtx2300ScbBusCfgType for supported configurations of the bus.

Data may be clocked in from the SENSE_IN_2 input. The clock rate is set internally too approximately 10kHz.

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 24 of 54 pages	

Description:	Configure the SCB serial bus.	
Primitive:	RTX2300_SET_SCB_BUS_CFG_REQ = 0x50E0	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ScbBusCfgType	Cfg	The new configuration

Description:	The SCB serial bus has been configured	
Primitive:	RTX2300_SET_SCB_BUS_CFG_CFM = 0x50E1	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get configuration of the SCB serial bus.	
Primitive:	RTX2300_GET_SCB_BUS_CFG_REQ = 0x50E2	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

Description:	The SCB serial bus configuration has been read	
Primitive:	RTX2300_GET_SCB_BUS_CFG_CFM = 0x50E3	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300ScbBusCfgType	Cfg	The current configuration
rsuint32	WriteData	The data last written to the bus.


Description:	Write data to the SCB serial bus	
Primitive:	RTX2300_WRITE_SCB_BUS_REQ = 0x50E4	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsuint32	Data	The data to send
rsuint8	BitCount	The number of bits to send

Description:		
Primitive:	RTX2300_WRITE_SCB_BUS_CFM = 0x50E5	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: a parameter was out of range
rsuint32	Data	The data read from the bus
rsuint8	BitCount	The number of bits received

13.4 Power Supply

13.4.1 Switching power supply on/off

Description:	Switch power supply on or off	
Primitive:	RTX2300_SET_PSU_SWITCH_REQ = 0x5100	
Parameters:		
Type	Name	Description

	Technical Documentation		Interface Specification	
	Rtx2300			
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 25 of 54 pages

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

Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	State	True = power supply on False = power supply off

Description:	The power supply switch has been set	
Primitive:	RTX2300_SET_PSU_SWITCH_CFM = 0x5101	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get power supply switch state	
Primitive:	RTX2300_GET_PSU_SWITCH_REQ = 0x5102	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

Description:		
Primitive:	RTX2300_GET_PSU_SWITCH_CFM = 0x5103	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	SupplyOn	True = power supply is on


13.4.2 Selecting internal/external power supply

Description:	Select which power supply to use	
Primitive:	RTX2300_SET_PSU_SELECTION_REQ = 0x5104	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Internal	True = internal powers upply selected, false = internal

Description:	The power supply has been selected	
Primitive:	RTX2300_SET_PSU_SELECTION_CFM = 0x5105	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

Description:	Get power supply selection	
Primitive:	RTX2300_GET_PSU_SELECTION_REQ = 0x5106	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

Description:		
Primitive:	RTX2300_GET_PSU_SELECTION_CFM = 0x5107	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
rsbool	Internal	True = internal powers upply selected, false = internal

	Technical Documentation Rtx2300			Interface Specification	
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 26 of 54 pages

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

13.4.3 Output voltage

Description:	Set the power supply output voltage	
Primitive:	RTX2300_SET_PSU_VOLTAGE_REQ = 0x5108	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300VoltageType	Voltage	

Description:	The output voltage has been set	
Primitive:	RTX2300_SET_PSU_VOLTAGE_CFM = 0x5109	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: invalid voltage RTX2300_ERR_BUSY: too many pending requests. RTX2300_ERR_NO_ACCESS: PSU module not found.


Description:	Get power supply output voltage.	
Primitive:	RTX2300_GET_PSU_VOLTAGE_REQ = 0x510A	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

Description:		
Primitive:	RTX2300_GET_PSU_VOLTAGE_CFM = 0x510B	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_BUSY: too many pending requests RTX2300_ERR_NO_ACCESS: PSU module not found.
Rtx2300VoltageType	Voltage_Set	The set output voltage.
Rtx2300VoltageType	Voltage_Out	The output voltage.
Rtx2300VoltageType	Voltage_SwMode	The output voltage from the internal switch mode regulator.

13.4.4 Output current

Description:	Set the power supply current limiter	
Primitive:	RTX2300_SET_PSU_CURRENT_REQ = 0x510C	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300CurrentType	Current	The max allowed current
Rtx2300CurrentRangeType	Range	The range to use while measuring the current. If <i>Range</i> is set to AUTO the range will automatically be set to the most suitable range.

Description:	The current limiter has been set	
Primitive:	RTX2300_SET_PSU_CURRENT_CFM = 0x510D	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: invalid current setting RTX2300_ERR_BUSY: too many pending requests RTX2300_ERR_NO_ACCESS: PSU module not found.

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 27 of 54 pages	

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


Description:	Get the present current output and the current limiter setting. The resolution depends of the selected range: 2000mA:2mA, 1000mA:1mA, 500mA:0,5mA, 100mA:0,2mA.		
Primitive:	RTX2300_GET_PSU_CURRENT_REQ = 0x510E		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	

Description:			
Primitive:	RTX2300_GET_PSU_CURRENT_CFM = 0x510F		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_BUSY: too many pending requests RTX2300_ERR_NO_ACCESS: PSU module not found.	
Rtx2300CurrentRangeType	Range	The range used while measuring the current	
Rtx2300CurrentType	Current_Set	The current limiter setting.	
Rtx2300CurrentType	Current	The integer part of actual output current.	
Rtx2300CurrentType	Current_Fraction	The fractional part of the actual output current. The value is the number of 1/10 of a mA, and is always a single-digit integer.	
rsuint16	Adc	The value of the current measurement ADC. For internal use only.	

13.4.4.1 Get average current measurement

Description:	Get the present average current output and the current limiter setting. The resolution depends of the selected range: 2000mA:2mA, 1000mA:1mA, 500mA:0,5mA, 100mA:0,2mA. This measurement uses a special averaging circuitry to measure the average current. In addition to this it is possible to add configurable software averaging.		
Primitive:	RTX2300_GET_PSU_AVG_CURRENT_REQ = 0x5110		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
rsuint8	SwAvgCount	The number of samples to average over. If 0 no software averaging is done, only hardware averaging. Max count is 100;	

Description:			
Primitive:	RTX2300_GET_PSU_AVG_CURRENT_CFM = 0x5111		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_BUSY: too many pending requests RTX2300_ERR_NO_ACCESS: PSU module not found.	
Rtx2300CurrentRangeType	Range	The range used while measuring the current	
Rtx2300CurrentType	Current_Set	The current limiter setting.	
Rtx2300CurrentType	Current	The integer part of actual output current.	
Rtx2300CurrentType	Current_Fraction	The fractional part of the actual output current. The value is the number of 1/10 of a mA, and is always a single-digit integer.	
rsuint16	Adc	The value of the current measurement ADC. For internal use only. If software averaging is applied this value will be the ADC reading of the last sample made.	

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 28 of 54 pages	

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

13.4.4.2 Get peak current measurement

Description:	Get power supply output peak voltage. The resolution depends of the selected range: 2000mA:2mA, 1000mA:1mA, 500mA:0,5mA, 100mA:0,2mA.		
Primitive:	RTX2300_GET_PSU_PEAK_CURRENT_REQ = 0x5112		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300PeakCurrentMeasureTimeType	Time	The time to measure over.	


Description:			
Primitive:	RTX2300_GET_PSU_PEAK_CURRENT_CFM = 0x5113		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_BUSY: too many pending requests RTX2300_ERR_NO_ACCESS: PSU module not found.	
Rtx2300CurrentRangeType	Range	The range used while measuring the current	
Rtx2300CurrentType	Current	The integer part of the peak output peak current.	
Rtx2300CurrentType	Current_Fraction	The fractional part of the peak output current. The value is the number of 1/10 of a mA, and is always a single-digit integer.	

13.4.5 Power supply indications

Description:	Power supply over-current indication. When received the internal current limiter in the PSU module has triggered, the internal switch-mode regulator has been disabled, and the output voltage has been disconnected from the DUT (see RTX2300_SET_PSU_SWITCH_REQ). To restore the output voltage you must reset the current limiter, see RTX2300_RESET_PSU_CURRENTLIM_REQ below.		
Primitive:	RTX2300_PSU_OVERCURRENT_IND = 0x5114		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
rsbool	Overcurrent	True: an overcurrent is detected. False: an over-current is no longer detected.	

Description:	Restore the power supply after the current limiter has triggered.		
Primitive:	RTX2300_RESET_PSU_CURRENTLIM_REQ = 0x5115		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
rsbool	SwitchVoltageOn	False: the PSU will enable the output using the voltage previously set. True: the output voltage will also be reconnected to the DUT.	

Description:	The current limiter has been reset		
Primitive:	RTX2300_RESET_PSU_CURRENTLIM_CFM = 0x5116		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	

	Technical Documentation Rtx2300			Interface Specification	
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 29 of 54 pages

Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: invalid current setting RTX2300_ERR_BUSY: too many pending requests RTX2300_ERR_NO_ACCESS: PSU module not found.
------------------	-----------	---

13.5 Pulse generator

Description:	Set up a output pulse on the specified output, see 10.1	
Primitive:	RTX2300_SET_PULSE_REQ = 0x5120	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300PulseDestinationType	Output	The output to generate the pulse on.
Rtx2300PulseModeType	PulseMode	The pulse generation pattern to apply.


Description:	The pulse has been set	
Primitive:	RTX2300_SET_PULSE_CFM = 0x5121	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: unknown pulse or output

Description:	Get the output pulse on the specified output, see 10.1	
Primitive:	RTX2300_GET_PULSE_REQ = 0x5122	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300PulseDestinationType	Output	The output to get the pulse setting from

Description:	The pulse setting has been retrieved	
Primitive:	RTX2300_GET_PULSE_CFM = 0x5123	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: unknown pulse or output
Rtx2300PulseModeType	PulseMode	The pulse generation pattern currently applied to the output

Description:	Set up an output pulse pattern for a user defined state. See 10.1	
Primitive:	RTX2300_SET_PULSE_PATTERN_REQ = 0x5124	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300PulseModeType	State	The pulse generation pattern to define. Must be one of the RTX2300_PULSEMODE_USER_DEFINED_X states!
Rtx2300PulsePatternDefinitionType	Pattern	The definition of it.

Description:	The pattern has been set	
Primitive:	RTX2300_SET_PULSE_PATTERN_CFM = 0x5125	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: unknown pattern

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 30 of 54 pages	

Description:	Get a output pulse pattern	
Primitive:	RTX2300_GET_PULSE_PATTERN_REQ = 0x5126	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300PulseModeType	State	The pulse generation pattern to get. Must be one of the RTX2300_PULSEMODE_USER_DEFINED_X states!

Description:		
Primitive:	RTX2300_GET_PULSE_PATTERN_CFM = 0x5127	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: unknown pattern
Rtx2300PulsePatternDefinitionType	Pattern	The definition


13.6 Input monitor

Description:	Set up a monitor for a specified input, see paragraph 11	
Primitive:	RTX2300_SET_INPUT_MONITOR_REQ = 0x5130	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300InputMonitorSourceType	Source	The input to monitor.
Rtx2300StateChangeType	StateChangeMode	The state change to report
rsbool	ActiveLow	False: the input is considered active when high. True: the input is considered active when low.
rsuint8	DebounceTime	The number of 10ms system ticks that the signal must be stable before being reported to the master. Use 0 if debouncing is not wanted. Caution: disabling debouncing on a rapidly changing input may cause huge amount of indications being sent!

Description:	The monitor has been set up.	
Primitive:	RTX2300_SET_INPUT_MONITOR_CFM = 0x5131	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: unknown input or change.

Description:	Get the monitor setting for a specified input, see paragraph 11	
Primitive:	RTX2300_GET_INPUT_MONITOR_REQ = 0x5132	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300InputMonitorSourceType	Source	The input monitor setting to get.

Description:	The input monitor setting	
Primitive:	RTX2300_GET_INPUT_MONITOR_CFM = 0x5133	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: unknown input or change.
Rtx2300StateChangeType	StateChangeMode	The state change to report

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 31 of 54 pages	

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

rsbool	ActiveLow	False: the input is considered active when high. True: the input is considered active when low.
rsuint8	DebounceTime	The number of 10ms system ticks that the signal must be stable before being reported to the master. Use 0 if debouncing is not wanted. Caution: disabling debouncing on a rapidly changing input may cause huge amount of indications being sent!

Description:	Input monitor indication: the Rtx2300 input monitor has detected a state change on the specified input.	
Primitive:	RTX2300_INPUT_MONITOR_IND = 0x5134	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300InputMonitorSourceType	Source	The input on which a change was detected.
Rtx2300StateChangeType	StateChange	The change that was detected

13.7 General indications

The following indications may be sent from the Rtx2300 to the masters at any time after initialization. Please note that indications are broadcast to all masters, so any PC program using the system must be prepared to handle indications it may not have asked for!


Description:	Firmware outdated. A newer firmware has been discovered	
Primitive:	RTX2300_FWU_OUTDATED_IND = 0x5140	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Local	True: the firmware is located locally on the PC False: the firmware is located remote. Please note: this is just information, it is not possible to actually upgrade to the remote firmware. In order to upgrade to the remote firmware it must be downloaded and located locally on the PC.
Rtx2300VersionInfoType	CurVersionNo	The current version number.
Rtx2300VersionInfoType	NewVersionNo	The version number to update to.

Description:	This indication is used to inform the master of various events in the system.	
Primitive:	RTX2300_SYSTEM_INFO_IND = 0x5141	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300SystemInfoType	Info	The info
rsuint32	AddInfo	Additional info. Contains 0 if not applicable.

13.8 House keeping

13.8.1 Initializing the system: RTX2300_INIT

Description:	Initialize the Rtx2300	
Primitive:	RTX2300_INIT_REQ = 0x5079	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300VersionNoType	Version	Reserved, specify 0.

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 32 of 54 pages	

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

Description:	The initialization has finished	
Primitive:	RTX2300_INIT_CFM = 0x507A	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_VERSION: one or more firmwares contain different version numbers. The system cannot be used before a firmware update has been made!

13.8.2 Resetting the system: RTX2300_RESET

Description:	Reset the Rtx2300 system. Please note that this will reset the entire system, including expansion modules. The system must be initialized again afterwards! No confirmation will be returned as the system is being reset, but a RTX2300_RESET_IND will be sent when the system restarts. Please note that resetting the system is not required for normal use.	
Primitive:	RTX2300_RESET_REQ = 0x5152	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

13.8.3 Requesting system status: RTX2300_GET_STATUS


Description:	Get the current status of Rtx2300.	
Primitive:	RTX2300_GET_STATUS_REQ = 0x507C	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

Description:	Return the current status	
Primitive:	RTX2300_GET_STATUS_CFM = 0x507D	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300StatusType	Status	Rtx2300 status

13.8.4 Requesting firmware version

Description:	Get version info for installed firmware. The info consists of a firmware defined NULL terminated string, and a 16 bit version number. This request can be used to interrogate the firmware version in case of version inconsistency problems. It can also be used to detect if a PSU module is mounted in the system.	
Primitive:	RTX2300_GET_VERSION_REQ = 0x507E	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300FirmwareType	Firmware	The firmware to request.

Description:	Return the version info	
Primitive:	RTX2300_GET_VERSION_CFM = 0x507F	
Parameters:		
Type	Name	Description

	Technical Documentation		Interface Specification	
	Rtx2300			
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 33 of 54 pages

Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_NO_ACCESS: firmware not found. This could happen if a module is not mounted.
Rtx2300VersionInfoType	VersionInfo	The version info.

13.8.5 Requesting main firmware information

Description:	Get additional version info for main firmware.	
Primitive:	RTX2300_GET_FIRMWARE_INFO_REQ = 0x5080	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.


Description:	Return the firmware info	
Primitive:	RTX2300_GET_FIRMWARE_INFO_CFM = 0x5081	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300DateType	LinkDate	The link date
Rtx2300VersionLabelType	VersionLabel	This field contains the version label as a zero terminated string.

13.8.6 Getting/setting insert/CCB info

Description:	Get info for installed insert or customization board. The info consists of a type identifier, an id and version info. The interpretation of all three is unknown to the Rtx2300.	
Primitive:	RTX2300_GET_INSERT_INFO_REQ = 0x5159	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Insert	True: get info for insert False: get info for the CCB

Description:	Return the insert/ccb info	
Primitive:	RTX2300_GET_INSERT_INFO_CFM = 0x515A	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_NO_ACCESS: fixture/CCB not found.
rsuint32	TypeInfo	The type info.
rsuint16	Id	The ID number.
Rtx2300VersionInfoType	VersionInfo	The version number.

Description:	Set the info for installed insert or customization board.	
Primitive:	RTX2300_SET_INSERT_INFO_REQ = 0x515B	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Insert	True: set info for insert False: set info for the CCB
rsuint32	TypeInfo	The type info.
rsuint16	Id	The ID number.

	Technical Documentation Rtx2300			Interface Specification	
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 34 of 54 pages

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

Rtx2300VersionInfoType	VersionInfo	The insert version number. If the insert does not supply a version number specify 0. The type info. If a type info string is not available the first char in the string must contain 0.
------------------------	-------------	--

Description:	The insert version info was set	
Primitive:	RTX2300_SET_INSERT_INFO_CFM = 0x515C	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_NO_ACCESS: fixture or CCB not found or coprocessor not present. RTX2300_ERR_BUSY: coprocessor busy because of too many requests.. RTX2300_ERR_AUTHENTICATION: wrong access mode, at least <i>Admin</i> is required.


13.8.7 Getting/setting serial number

Description:	Set serial number information. The serial number information is not used by the firmware. The primary serial number is a number that uniquely identifies this particular RTX2300 system. The secondary serial number may be used for any purpose. It requires <i>Manufacturer</i> access rights to change the primary serial number, while the secondary serial number requires <i>Admin</i> access rights.	
Primitive:	RTX2300_SET_SERIALNO_REQ = 0x5161	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	PsuSerial	True: set the serial number in the PSU module. False: set the serial number in the mainboard.
rsbool	SetPrimary	True: set the primary serial number False: set the secondary serial number
Rtx2300SerialNumberType	SerialNo	The serial number.

Description:	The serial number has been set	
Primitive:	RTX2300_SET_SERIALNO_CFM = 0x5162	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: the user does not have the required privilege to do this. RTX2300_ERR_NO_ACCESS: PSU module not found.

Description:	Get the serial number	
Primitive:	RTX2300_GET_SERIALNO_REQ = 0x5163	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	PsuSerial	True: get the serial number from the PSU module. False: get the serial number from the mainboard.

Description:		
Primitive:	RTX2300_GET_SERIALNO_CFM = 0x5164	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 35 of 54 pages	

Rtx2300SerialNumberType	PrimSerialNo	The primary serial number.
Rtx2300SerialNumberType	SecSerialNo	The secondary serial number.

13.8.8 Getting the Manufacturer Information

The Manufacturer Information is a set of information that describes the system. The information is stored during manufacturing and cannot be changed.

Description:	Get the Manufacturer Information	
Primitive:	RTX2300_GET_MANUFACTURER_INFO_REQ = 0x5165	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Psu	True: get the Manufacturer Information from the PSU module. False: get the Manufacturer Information from the mainboard.

Description:		
Primitive:	RTX2300_GET_MANUFACTURER_INFO_CFM = 0x5166	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR
Rtx2300ManufacturerInfoType	Info	The Manufacturer Information.

13.8.9 Getting the current temperature


Description:	Request current temperature from the device	
Primitive:	RTX2300_GET_TEMPERATURE_REQ = 0x5167	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	PsuTemp	True: return the temperature in the PSU module. False: return the temperature in the mainboard.

Description:	The version info has been returned from the device	
Primitive:	RTX2300_GET_TEMPERATURE_CFM = 0x5168	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ACCESS: PSU module not found.
Rtx2300TemperatureType	Temperature	The current temperature in degrees Celsius. Accuracy is +/- 10 degrees.

13.8.10 User data handling

These requests allow the client to access the user area of the on-board EEPROM. See also RTX2300_USERDATA_SIZE.

Description:	Write data to the UserData area. <i>Admin</i> access rights are required.	
Primitive:	RTX2300_WRITE_USERDATA_REQ = 0x5169	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Fixture	True: access the user data in the fixture False: access the user data on the mainboard
rsuint16	Addr	The offset into the user data area.

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 36 of 54 pages	

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

rsuint8	ByteCount	The number of bytes to write
Rtx2300UserDataType	Data	The data to write

Description:	The data has been written	
Primitive:	RTX2300_WRITE_USERDATA_CFM = 0x516A	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: attempt to access outside the user area. RTX2300_ERR_AUTHENTICATION: the user does not have the required privilege to do this.


Description:	Read data from the UserData area	
Primitive:	RTX2300_READ_USERDATA_REQ = 0x516B	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	Fixture	True: access the user data in the fixture False: access the user data on the mainboard
rsuint16	Addr	The offset into the user data area.
rsuint8	ByteCount	The number of bytes to read, max 16 bytes

Description:	The data has been read	
Primitive:	RTX2300_READ_USERDATA_CFM = 0x516C	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: attempt to access outside the user area.
rsuint8	ByteCount	The number of bytes read
Rtx2300UserDataType	Data	The data read.

13.8.11 Feature configuration

Description:	Set the availability of a feature. Access mode must be set to <i>Admin</i> mode	
Primitive:	RTX2300_SET_FEATURE_ACCESS_REQ = 0x516D	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300FeatureType	Feature	The feature to set
rsbool	Enable	If true the feature is enabled.
Rtx2300PasswordType	Password	The password required to set the feature.

Description:	Feature access has been set	
Primitive:	RTX2300_SET_FEATURE_ACCESS_CFM = 0x516E	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: wrong password or access mode. RTX2300_ERR_RANGE: unknown feature

	Technical Documentation Rtx2300			Interface Specification	
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 37 of 54 pages

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

Description:	Get a feature access setting	
Primitive:	RTX2300_GET_FEATURE_ACCESS_REQ = 0x516F	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300FeatureType	Feature	The feature to set

Description:	The feature access setting	
Primitive:	RTX2300_GET_FEATURE_ACCESS_CFM = 0x5170	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: unknown feature
rsbool	Enable	If true the feature is enabled.


13.8.12 Access mode

Description:	Set the access mode	
Primitive:	RTX2300_SET_ACCESS_MODE_REQ = 0x5171	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300AccessModeType	AccessMode	The required access mode
Rtx2300PasswordType	Password	The password required to enable the mode. No password is required to enable user mode, use 0.

Description:	Access mode has been enabled	
Primitive:	RTX2300_SET_ACCESS_MODE_CFM = 0x5172	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: wrong password RTX2300_ERR_NO_ACCESS: 3 consecutive failed attempts to set the access mode was registered. Power-cycle the system and retry. RTX2300_ERR_RANGE: illegal or no password specified RTX2300_ERR_BUSY: coprocessor busy because of too many requests..

Description:	Get the access mode	
Primitive:	RTX2300_GET_ACCESS_MODE_REQ = 0x5173	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

Description:	The current access mode	
Primitive:	RTX2300_GET_ACCESS_MODE_CFM = 0x5174	
Parameters:		

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
Page: 38 of 54 pages				

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

Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_NO_ACCESS: coprocessor not present. RTX2300_ERR_BUSY: coprocessor busy because of too many requests.
Rtx2300AccessModeType	AccessMode	The current access mode

13.8.13 Test counters

The Rtx2300 system supports a set of counters that can be used to count the number of tests performed. This information is important in cases where the equipment has a limited life in terms of test cycles. Counters exist in the Rtx2300 system itself, the inserts and the customization boards. The counters in the Rtx2300 system always exist, while the inserts and customization boards requires an onboard EEPROM.

Please note that activating the magnet (see 13.3.6) will automatically increment the system test counter!

Description:	Get the current value of a test counter	
Primitive:	RTX2300_GET_TEST_COUNTER_REQ = 0x5175	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300TestCounterType	Counter	The counter to get
rsbool	Increment	If true the counter will be incremented before the counter value is returned.


Description:	The current test counter value	
Primitive:	RTX2300_GET_TEST_COUNTER_CFM = 0x5176	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_NO_ACCESS: no hardware support found.
Rtx2300TestCounterValueType	CounterValue	The current counter value

13.9 Misc commands

13.9.1 Simulation Configuration

Description:	Configure the simulation. This command has no effect unless the target is running as a simulation. It is for internal use only.	
Primitive:	RTX2300_SET_SIM_CFG_REQ = 0x51B0	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300PrimitiveType	CfgPrimitive	The command to configure.
rsuint16	Mode	The configuration mode.
Rtx2300SimCfgDataType	Data	The configuration data

Description:	Simulation configuration has been set.	
Primitive:	RTX2300_SET_SIM_CFG_CFM = 0x51B1	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.

	Technical Documentation Rtx2300			Interface Specification
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
Page: 39 of 54 pages				

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

Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_RANGE: illegal parameter. RTX2300_ERR_UNSUPPORTED: feature is not supported or not running a simulation.
------------------	-----------	---

14 Debug interface

The debug interfaces is for debugging the target hardware and software. It is by no means intended for normal use by the master software. To ensure this the debug interfaces must be enabled before being available. When enabled parts of the normal functionality will be reduced or disabled. Leaving debug mode resets the target to ensure that the system is in working condition.

The following high and low level debug interfaces all require that debug mode has been enabled.

14.1 High level debug interface

14.1.1 Attenuator driver


Description:	Set attenuator		
Primitive:	RTX2300_DBG_SET_ATTENUATOR_REQ = 0x5180		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
rsuint8	Channel	The channel to set	
rsuint8	Attenuation	The new value	

Description:	The attenuator has been set		
Primitive:	RTX2300_DBG_SET_ATTENUATOR_CFM = 0x5181		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: debug mode not enabled	

14.1.2 ADC task driver

Description:	Read ADC		
Primitive:	RTX2300_DBG_READ_ADC_REQ = 0x5182		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
rsuint8	Cfg	The input configuration, see Max1300 datasheet	

Description:	The ADC has been read		
Primitive:	RTX2300_DBG_READ_ADC_CFM = 0x5183		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 40 of 54 pages	

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

Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: debug mode not enabled
rsuint16	Value	The read ADC value

14.1.3 DAC task driver

Description:	Set up a DAC	
Primitive:	RTX2300_DBG_SET_DAC_REQ = 0x5184	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
rsbool	DacCh	False: DAC1 True: DAC2
rsuint16	Data	The data to write

Description:	The DAC has been set up	
Primitive:	RTX2300_DBG_SET_DAC_CFM = 0x5185	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: debug mode not enabled


14.2 Low level debug interface

14.2.1 IO expander

Description:	Write to an output expander	
Primitive:	RTX2300_DBG_WRITE_OUTPUT_EXPANDER_REQ = 0x5186	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ExpanderNoType	ExpanderNo	0: ACS1 based expander 1: ACS2 based expander 2: ACS3 based expander
rsuint16	Data	The data to write.

Description:	Data has been written to the output expander	
Primitive:	RTX2300_DBG_WRITE_OUTPUT_EXPANDER_CFM = 0x5187	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: debug mode not enabled

Description:	Read from the output expander	
Primitive:	RTX2300_DBG_READ_OUTPUT_EXPANDER_REQ = 0x5188	
Parameters:		
Type	Name	Description
Rtx2300InstanceNoType	InstNo	The instance number.
Rtx2300ExpanderNoType	ExpanderNo	0: ACS1 based expander 1: ACS2 based expander 2: ACS3 based expander

	Technical Documentation Rtx2300			Interface Specification	
	Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 41 of 54 pages

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

Description:	Data has been read from the output expander		
Primitive:	RTX2300_DBG_READ_OUTPUT_EXPANDER_CFM = 0x5189		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: debug mode not enabled	
rsuint16	Data	The data read.	


Description:	Read from the input expander		
Primitive:	RTX2300_DBG_READ_INPUT_EXPANDER_REQ = 0x518A		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	

Description:	Data has been read from the input expander		
Primitive:	RTX2300_DBG_READ_INPUT_EXPANDER_CFM = 0x518B		
Parameters:			
Type	Name	Description	
Rtx2300InstanceNoType	InstNo	The instance number.	
Rtx2300ErrorType	ErrorCode	RTX2300_ERR_NO_ERROR RTX2300_ERR_AUTHENTICATION: debug mode not enabled	
rsuint8	Data	The data read.	

15 Types

15.1 Rtx2300SystemInfoType

Description:	This type is used to return system info to the client.		
C-syntax:			
typedef enum			
{			
RTX2300_SYSINFO_RESET,		The system has been reset and is starting to initialize.	
RTX2300_SYSINFO_READY,		The system is initialized and ready for use.	
RTX2300_SYSINFO_VERSION_INCONSISTENCY,		Firmware version inconsistency! One or more softwares in the system are having different version numbers. The firmware must be updated before the system can be used.	
RTX2300_SYSINFO_I2CBUS1_BLOCKED,		The I2C bus to the fixture EEPROM is blocked.	
RTX2300_SYSINFO_I2CBUS2_BLOCKED,		The I2C bus to the PSU module and/or the expansion modules is blocked.	
RTX2300_SYSINFO_I2C_DEVICE_BLOCKED,		A device on the I2C bus is blocked (it's handshake line to the central CPU is permanently asserted) and the module has been disabled. The AddInfo parameter contains additional internal info.	
RTX2300_SYSINFO_UNKNOWN_REQ,		An unknown request has been received. As it is unknown it cannot be replied to, and has thus been discarded. The AddInfo contains the primitive of the request.	
RTX2300_SYSINFO_SYSINT_FAULT,		System Integrity Fault mode has been entered. Please contact RTX Telecom.	
RTX2300_SYSINFO_PSU_UNDERVOLTAGE,		An undervoltage has been detected on the system powersupply.	
RTX2300_SYSINFO_PSU_OVERVOLTAGE,		An overvoltage has been detected on the system powersupply.	
RTX2300_SYSINFO_COUNT			
} RSENUM8 (Rtx2300SystemInfoType);			

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 42 of 54 pages	


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

15.2 Rtx2300PulseModeType

Description:	Defines the possible modes for the internal pulse generator.	
C-syntax:		
typedef enum		
{		
RTX2300_PULSEMODE_OFF,		Output is constantly off
RTX2300_PULSEMODE_ON,		Output is constantly on
RTX2300_PULSEMODE_PULSE_SHORT,		Output is on for 10 ms then goes off
RTX2300_PULSEMODE_PULSE_MEDIUM,		Output is on for 100 ms then goes off
RTX2300_PULSEMODE_PULSE_LONG,		Output is on for 1000 ms then goes off
RTX2300_PULSEMODE_FLASH_SLOW,		Output flashes constantly with 0.5 HZ and 50% dutycycle, i.e. 1000ms on/1000ms off
RTX2300_PULSEMODE_FLASH_MEDIUM,		Output flashes constantly with 1 HZ and 50% dutycycle, i.e. 500ms on/500ms off
RTX2300_PULSEMODE_FLASH_QUICK,		Output flashes constantly with 2.5 HZ and 50% dutycycle, i.e. 200ms on/200ms off
RTX2300_PULSEMODE_FLASH_LONG_SLOW,		Output flashes constantly with 0.5 HZ and 90% dutycycle, i.e. 1800ms on/200ms off
RTX2300_PULSEMODE_FLASH_LONG_MEDIUM,		Output flashes constantly with 1 HZ and 90% dutycycle, i.e. 900ms on/100ms off
RTX2300_PULSEMODE_FLASH_LONG_QUICK,		Output flashes constantly with 2.5 HZ and 90% dutycycle, i.e. 360ms on/40ms off
RTX2300_PULSEMODE_FLASH_SHORT_SLOW,		Output flashes constantly with 0.5 HZ and 10% dutycycle, i.e. 200ms on/1800ms off
RTX2300_PULSEMODE_FLASH_SHORT_MEDIUM,		Output flashes constantly with 1 HZ and 10% dutycycle, i.e. 100ms on/900ms off
RTX2300_PULSEMODE_FLASH_SHORT_QUICK,		Output flashes constantly with 2.5 HZ and 10% dutycycle, i.e. 40ms on/360ms off
RTX2300_PULSEMODE_USER_DEFINED_0,		The output will pulse or flash according to user defined settings.
RTX2300_PULSEMODE_USER_DEFINED_1,		The output will pulse or flash according to user defined settings.
RTX2300_PULSEMODE_USER_DEFINED_2,		The output will pulse or flash according to user defined settings.
RTX2300_PULSEMODE_USER_DEFINED_3,		The output will pulse or flash according to user defined settings.
RTX2300_PULSEMODE_COUNT,		
} RSENUM8 (Rtx2300PulseModeType);		

15.3 Rtx2300PulseDestinationType

Description:	Defines the possible outputs that may use the internal pulse generator.	
C-syntax:		
typedef enum		
{		
RTX2300_PULSEDEST_DIGOUT_0,		Digital out put 0 / Source output 0
RTX2300_PULSEDEST_DIGOUT_1,		Digital out put 1 / Source output 1
RTX2300_PULSEDEST_DIGOUT_2,		Digital out put 2 / Source output 2
RTX2300_PULSEDEST_DIGOUT_3,		Digital out put 3 / Source output 3
RTX2300_PULSEDEST_DIGOUT_4,		Digital out put 4 / Source output 4
RTX2300_PULSEDEST_DIGOUT_5,		Digital out put 5 / Source output 5
RTX2300_PULSEDEST_DIGOUT_6,		Digital out put 6 / Source output 6
RTX2300_PULSEDEST_DIGOUT_7,		Digital out put 7 / Source output 7
RTX2300_PULSEDEST_DIGOUT_8,		Digital out put 8 / Sink output 0
RTX2300_PULSEDEST_DIGOUT_9,		Digital out put 9 / Sink output 1
RTX2300_PULSEDEST_DIGOUT_10,		Digital out put 10 / Sink output 2
RTX2300_PULSEDEST_DIGOUT_11,		Digital out put 11 / Sink output 3
RTX2300_PULSEDEST_DIGOUT_12,		Digital out put 12 / Sink output 4
RTX2300_PULSEDEST_DIGOUT_13,		Digital out put 13 / Sink output 5
RTX2300_PULSEDEST_DIGOUT_14,		Digital out put 14 / Sink output 6
RTX2300_PULSEDEST_DIGOUT_15,		Digital out put 15 / Sink output 7
RTX2300_PULSEDEST_RELAY_0,		Relay 0
RTX2300_PULSEDEST_RELAY_1,		Relay 1
RTX2300_PULSEDEST_RELAY_2,		Relay 2
RTX2300_PULSEDEST_RELAY_3,		Relay 3
RTX2300_PULSEDEST_RELAY_4,		Relay 4
RTX2300_PULSEDEST_RELAY_5,		Relay 5
RTX2300_PULSEDEST_RELAY_6,		Relay 6
RTX2300_PULSEDEST_RELAY_7,		Relay 7

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 43 of 54 pages	

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


RTX2300_PULSEDEST_FRONT_LED_0,	Front LED 0
RTX2300_PULSEDEST_FRONT_LED_1,	Front LED 1
RTX2300_PULSEDEST_FRONT_LED_2,	Front LED 2
RTX2300_PULSEDEST_FIXTURE_CONTROL_0,	Fixture control 0 (FIX_CON_0)
RTX2300_PULSEDEST_FIXTURE_CONTROL_1,	Fixture control 1 (FIX_CON_1)
RTX2300_PULSEDEST_FIXTURE_CONTROL_2,	Fixture control 2 (FIX_CON_2)
RTX2300_PULSEDEST_FIXTURE_CONTROL_3,	Fixture control 3 (FIX_CON_3)
RTX2300_PULSEDEST_EXPANSION_CONTROL_0,	Expansion control 0 (EXP_CTRL_0)
RTX2300_PULSEDEST_EXPANSION_CONTROL_1,	Expansion control 1 (EXP_CTRL_1)
RTX2300_PULSEDEST_EXPANSION_CONTROL_2,	Expansion control 2 (EXP_CTRL_2)
RTX2300_PULSEDEST_EXPANSION_CONTROL_3,	Expansion control 3 (EXP_CTRL_3)
RTX2300_PULSEDEST_COUNT,	
} RSENUM8 (Rtx2300PulseDestinationType);	

15.4 Rtx2300InputMonitorSourceType

Description:	Defines the possible inputs that may be used the input monitor .
C-syntax:	
typedef enum	
{	
RTX2300_MONITORSRC_DIGIN_0,	Digital input 0
RTX2300_MONITORSRC_DIGIN_1,	Digital input 1
RTX2300_MONITORSRC_DIGIN_2,	Digital input 2
RTX2300_MONITORSRC_DIGIN_3,	Digital input 3
RTX2300_MONITORSRC_DIGIN_4,	Digital input 4
RTX2300_MONITORSRC_DIGIN_5,	Digital input 5
RTX2300_MONITORSRC_DIGIN_6,	Digital input 6
RTX2300_MONITORSRC_DIGIN_7,	Digital input7
RTX2300_MONITORSRC_SENSE_0,	Digital interrupt input 0
RTX2300_MONITORSRC_SENSE_1,	Digital interrupt input 1
RTX2300_MONITORSRC_SENSE_2,	Digital interrupt input 2
RTX2300_MONITORSRC_SENSE_3,	Digital interrupt input 3
RTX2300_MONITORSRC_MICROSW,	Digital interrupt input 4
RTX2300_MONITORSRC_FRONT_CONTROL_0,	Digital interrupt input 5
RTX2300_MONITORSRC_FRONT_CONTROL_1,	Digital interrupt input 6
RTX2300_MONITORSRC_FRONT_CONTROL_2,	Digital interrupt input 7
RTX2300_MONITORSRC_COUNT	
} RSENUM8 (Rtx2300InputMonitorSourceType);	

15.5 Rtx2300AdcCfgType

Description:	Defines the available signal inputs and ranges for ADC measurements
C-syntax:	
typedef enum	
{	

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 44 of 54 pages	

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

RTX2300_ADCCFG_RANGE3V3_SE0 = 0x00,	Range -3V - +3V, Single-ended input 0
RTX2300_ADCCFG_RANGE3V3_SE1 = 0x01,	Range -3V - +3V, Single-ended input 1
RTX2300_ADCCFG_RANGE3V3_SE2 = 0x02,	Range -3V - +3V, Single-ended input 2
RTX2300_ADCCFG_RANGE3V3_SE3 = 0x03,	Range -3V - +3V, Single-ended input 3
RTX2300_ADCCFG_RANGE3V3_SE4 = 0x04,	Range -3V - +3V, Single-ended input 4
RTX2300_ADCCFG_RANGE3V3_SE5 = 0x05,	Range -3V - +3V, Single-ended input 5
RTX2300_ADCCFG_RANGE3V3_SE6 = 0x06,	Range -3V - +3V, Single-ended input 6
RTX2300_ADCCFG_RANGE3V3_SE7 = 0x07,	Range -3V - +3V, Single-ended input 7
RTX2300_ADCCFG_RANGE6V0_SE0 = 0x08,	Range -6V - 0V, Single-ended input 0
RTX2300_ADCCFG_RANGE6V0_SE1 = 0x09,	Range -6V - 0V, Single-ended input 1
RTX2300_ADCCFG_RANGE6V0_SE2 = 0x0A,	Range -6V - 0V, Single-ended input 2
RTX2300_ADCCFG_RANGE6V0_SE3 = 0x0B,	Range -6V - 0V, Single-ended input 3
RTX2300_ADCCFG_RANGE6V0_SE4 = 0x0C,	Range -6V - 0V, Single-ended input 4
RTX2300_ADCCFG_RANGE6V0_SE5 = 0x0D,	Range -6V - 0V, Single-ended input 5
RTX2300_ADCCFG_RANGE6V0_SE6 = 0x0E,	Range -6V - 0V, Single-ended input 6
RTX2300_ADCCFG_RANGE6V0_SE7 = 0x0F,	Range -6V - 0V, Single-ended input 7
RTX2300_ADCCFG_RANGE0V6_SE0 = 0x10,	Range 0V - +6V, Single-ended input 0
RTX2300_ADCCFG_RANGE0V6_SE1 = 0x11,	Range 0V - +6V, Single-ended input 1
RTX2300_ADCCFG_RANGE0V6_SE2 = 0x12,	Range 0V - +6V, Single-ended input 2
RTX2300_ADCCFG_RANGE0V6_SE3 = 0x13,	Range 0V - +6V, Single-ended input 3
RTX2300_ADCCFG_RANGE0V6_SE4 = 0x14,	Range 0V - +6V, Single-ended input 4
RTX2300_ADCCFG_RANGE0V6_SE5 = 0x15,	Range 0V - +6V, Single-ended input 5
RTX2300_ADCCFG_RANGE0V6_SE6 = 0x16,	Range 0V - +6V, Single-ended input 6
RTX2300_ADCCFG_RANGE0V6_SE7 = 0x17,	Range 0V - +6V, Single-ended input 7
RTX2300_ADCCFG_RANGE6V6_SE0 = 0x18,	Range -6V - +6V, Single-ended input 0
RTX2300_ADCCFG_RANGE6V6_SE1 = 0x19,	Range -6V - +6V, Single-ended input 1
RTX2300_ADCCFG_RANGE6V6_SE2 = 0x1A,	Range -6V - +6V, Single-ended input 2
RTX2300_ADCCFG_RANGE6V6_SE3 = 0x1B,	Range -6V - +6V, Single-ended input 3
RTX2300_ADCCFG_RANGE6V6_SE4 = 0x1C,	Range -6V - +6V, Single-ended input 4
RTX2300_ADCCFG_RANGE6V6_SE5 = 0x1D,	Range -6V - +6V, Single-ended input 5
RTX2300_ADCCFG_RANGE6V6_SE6 = 0x1E,	Range -6V - +6V, Single-ended input 6
RTX2300_ADCCFG_RANGE6V6_SE7 = 0x1F,	Range -6V - +6V, Single-ended input 7
RTX2300_ADCCFG_RANGE12V0_SE0 = 0x20,	Range -12V - 0V, Single-ended input 0
RTX2300_ADCCFG_RANGE12V0_SE1 = 0x21,	Range -12V - 0V, Single-ended input 1
RTX2300_ADCCFG_RANGE12V0_SE2 = 0x22,	Range -12V - 0V, Single-ended input 2
RTX2300_ADCCFG_RANGE12V0_SE3 = 0x23,	Range -12V - 0V, Single-ended input 3
RTX2300_ADCCFG_RANGE12V0_SE4 = 0x24,	Range -12V - 0V, Single-ended input 4
RTX2300_ADCCFG_RANGE12V0_SE5 = 0x25,	Range -12V - 0V, Single-ended input 5
RTX2300_ADCCFG_RANGE12V0_SE6 = 0x26,	Range -12V - 0V, Single-ended input 6
RTX2300_ADCCFG_RANGE12V0_SE7 = 0x27,	Range -12V - 0V, Single-ended input 7
RTX2300_ADCCFG_RANGE0V12_SE0 = 0x28,	Range 0V - +12V, Single-ended input 0
RTX2300_ADCCFG_RANGE0V12_SE1 = 0x29,	Range 0V - +12V, Single-ended input 1
RTX2300_ADCCFG_RANGE0V12_SE2 = 0x2A,	Range 0V - +12V, Single-ended input 2
RTX2300_ADCCFG_RANGE0V12_SE3 = 0x2B,	Range 0V - +12V, Single-ended input 3
RTX2300_ADCCFG_RANGE0V12_SE4 = 0x2C,	Range 0V - +12V, Single-ended input 4
RTX2300_ADCCFG_RANGE0V12_SE5 = 0x2D,	Range 0V - +12V, Single-ended input 5
RTX2300_ADCCFG_RANGE0V12_SE6 = 0x2E,	Range 0V - +12V, Single-ended input 6
RTX2300_ADCCFG_RANGE0V12_SE7 = 0x2F,	Range 0V - +12V, Single-ended input 7
RTX2300_ADCCFG_RANGE12V12_SE0 = 0x30,	Range -12V - +12V, Single-ended input 0
RTX2300_ADCCFG_RANGE12V12_SE1 = 0x31,	Range -12V - +12V, Single-ended input 1
RTX2300_ADCCFG_RANGE12V12_SE2 = 0x32,	Range -12V - +12V, Single-ended input 2
RTX2300_ADCCFG_RANGE12V12_SE3 = 0x33,	Range -12V - +12V, Single-ended input 3
RTX2300_ADCCFG_RANGE12V12_SE4 = 0x34,	Range -12V - +12V, Single-ended input 4
RTX2300_ADCCFG_RANGE12V12_SE5 = 0x35,	Range -12V - +12V, Single-ended input 5
RTX2300_ADCCFG_RANGE12V12_SE6 = 0x36,	Range -12V - +12V, Single-ended input 6
RTX2300_ADCCFG_RANGE12V12_SE7 = 0x37,	Range -12V - +12V, Single-ended input 7
RTX2300_ADCCFG_RANGE6V6_DIF01 = 0x38,	Range -6V - +6V, differential input 0 and 1
RTX2300_ADCCFG_RANGE6V6_DIF23 = 0x39,	Range -6V - +6V, differential input 2 and 3
RTX2300_ADCCFG_RANGE6V6_DIF45 = 0x3A,	Range -6V - +6V, differential input 4 and 5
RTX2300_ADCCFG_RANGE6V6_DIF67 = 0x3B,	Range -6V - +6V, differential input 6 and 7
RTX2300_ADCCFG_RANGE12V12_DIF01 = 0x3C,	Range -12V - +12V, differential input 0 and 1
RTX2300_ADCCFG_RANGE12V12_DIF23 = 0x3D,	Range -12V - +12V, differential input 2 and 3
RTX2300_ADCCFG_RANGE12V12_DIF45 = 0x3E,	Range -12V - +12V, differential input 4 and 5
RTX2300_ADCCFG_RANGE12V12_DIF67 = 0x3F,	Range -12V - +12V, differential input 6 and 7
RTX2300_ADCCFG_RANGE24V24_DIF01 = 0x40,	Range -24V - +24V, differential input 0 and 1
RTX2300_ADCCFG_RANGE24V24_DIF23 = 0x41,	Range -24V - +24V, differential input 2 and 3
RTX2300_ADCCFG_RANGE24V24_DIF45 = 0x42,	Range -24V - +24V, differential input 4 and 5



<code>RTX2300_ADCCFG_RANGE24V24_DIF67 = 0x43,</code>	Range -24V - +24V, differential input 6 and 7
<code>RTX2300_ADCCFG_COUNT</code>	The number of possible input configurations
<code>} RSENUM8 (Rtx2300AdcCfgType);</code>	

15.6 Rtx2300DacChannelType

Description:	Defines the available channels in the DAC	
C-syntax:	typedef enum	
	{	
	<code>RTX2300_DAC_CHANNEL_0,</code>	DAC_1
	<code>RTX2300_DAC_CHANNEL_1,</code>	DAC_2
	<code>RTX2300_DAC_CHANNEL_COUNT</code>	The number of possible channels
	<code>} RSENUM8 (Rtx2300DacChannelType);</code>	

15.1 Rtx2300AudioLvlMeasureModeType

Description:	Defines the available modes when measuring audio levels	
C-syntax:	typedef enum	
	{	
	<code>RTX2300_AUDIOLVLMESS_MODE_RMS,</code>	RMS measurement
	<code>RTX2300_AUDIOLVLMESS_MODE_PP,</code>	Peak-peak measurement
	<code>RTX2300_AUDIOLVLMESS_MODE_COUNT</code>	The number of possible modes
	<code>} RSENUM8 (Rtx2300AudioLvlMeasureModeType);</code>	

15.2 Rtx2300DistortionMeasureModeType


Description:	Defines the available modes when measuring audio distortion	
C-syntax:	typedef enum	
	{	
	<code>RTX2300_DISTORTIONMESS_MODE_THD,</code>	Total Harmonic Distortion
	<code>RTX2300_DISTORTIONMESS_MODE_COUNT</code>	The number of possible modes
	<code>} RSENUM8 (Rtx2300DistortionMeasureModeType);</code>	

15.3 Rtx2300CurrentRangeType

Description:	Defines the allowed current ranges	
C-syntax:	typedef enum	
	{	
	<code>RTX2300_CURRENT_RANGE_100MA,</code>	Use the 100 mA range
	<code>RTX2300_CURRENT_RANGE_500MA,</code>	Use the 500 mA range
	<code>RTX2300_CURRENT_RANGE_1000MA,</code>	Use the 1000 mA range
	<code>RTX2300_CURRENT_RANGE_2000MA,</code>	Use the 2000 mA range
	<code>RTX2300_CURRENT_RANGE_AUTO,</code>	Set the range automatically.
	<code>RTX2300_CURRENT_RANGE_COUNT</code>	The number of possible ranges
	<code>} RSENUM8 (Rtx2300CurrentRangeType);</code>	

15.4 Rtx2300PeakCurrentMeasuresTimeType

Description:	Defines the time to measure peak current over
C-syntax:	

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 46 of 54 pages	

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

typedef enum		
{		
RTX2300 PEAK CURRENT MESSTIME_10MS,		Measure peak current over 10 ms
RTX2300 PEAK CURRENT MESSTIME_100MS,		Measure peak current over 100 ms
RTX2300 PEAK CURRENT MESSTIME_1000MS,		Measure peak current over 1000 ms
RTX2300 PEAK CURRENT MESSTIME_INFINITE,		Measure peak current since last measurement.
RTX2300 PEAK CURRENT MESSTIME_COUNT		The number of possible measurement times
} RSENUM8 (Rtx2300PeakCurrentMeasureTimeType);		

15.5 Rtx2300AudioInputType

Description:	Defines the available audio inputs	
C-syntax:		
typedef enum		
{		
RTX2300 AUDIO_INPUT_0,		Audio A input
RTX2300 AUDIO_INPUT_1,		Audio B input
RTX2300 AUDIO_INPUT_DIFF,		Differential input on audio A and B inputs
RTX2300 AUDIO_INPUT_COUNT		The number of possible inputs
} RSENUM8 (Rtx2300AudioInputType);		

15.6 Audio attenuator constants


Description:	Mute
C-syntax:	
#define RTX2300_ATT_MUTE 0xFF	Attenuator is muted

15.7 Rtx2300AudioGenChannelType

Description:	Defines the available channels in the audio generator. Please note that the A and B channels are driven by the same internal codec, so they cannot be used simultaneously. The TONE output, however, is independent of the A/B channels.	
C-syntax:		
typedef enum		
{		
RTX2300 AUDIOGEN_OFF,		No output
RTX2300 AUDIOGEN_CHANNEL_TONE_OUT,		Output on the TONE_OUT_SCB signal
RTX2300 AUDIOGEN_CHANNEL_A,		Output on the AUDIO_A_OUT signal, AUDIO_B_OUT is disabled.
RTX2300 AUDIOGEN_CHANNEL_B,		Output on the AUDIO_B_OUT signal, AUDIO_A_OUT is disabled.
RTX2300 AUDIOGEN_CHANNEL_AB,		Output on the AUDIO_A_OUT and AUDIO_B_OUT signal in differential mode.
RTX2300 AUDIOGEN_CHANNEL_COUNT		The number of possible channels
} RSENUM8 (Rtx2300AudioGenChannelType);		

15.8 Rtx2300InterruptMaskType

Description:	Defines the available interrupt inputs	
C-syntax:		
typedef enum		
{		
RTX2300 INTERRUPT_MASK_0 = RSBIT0,		
RTX2300 INTERRUPT_MASK_1 = RSBIT1,		
RTX2300 INTERRUPT_MASK_2 = RSBIT2,		
RTX2300 INTERRUPT_MASK_3 = RSBIT3,		
RTX2300 INTERRUPT_MASK_MICRO_SW = RSBIT4,		
RTX2300 INTERRUPT_MASK_FRONT_CONTROL_0 = RSBIT5,		
RTX2300 INTERRUPT_MASK_FRONT_CONTROL_1 = RSBIT6,		
RTX2300 INTERRUPT_MASK_FRONT_CONTROL_2 = RSBIT7,		
RTX2300 INTERRUPT_MASK_ALL = 0xFF		

	Technical Documentation			Interface Specification
	Rtx2300			
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 47 of 54 pages

} RSENUM8 (Rtx2300InterruptMaskType);	
---------------------------------------	--

15.9 Rtx2300InterruptNoType

Description:	Defines the available interrupt input numbers
C-syntax:	
typedef enum	
{	
RTX2300_INTERRUPT_NO_0,	
RTX2300_INTERRUPT_NO_1,	
RTX2300_INTERRUPT_NO_2,	
RTX2300_INTERRUPT_NO_3,	
RTX2300_INTERRUPT_NO_MICRO_SW,	
RTX2300_INTERRUPT_NO_FRONT_CONTROL_0,	
RTX2300_INTERRUPT_NO_FRONT_CONTROL_1,	
RTX2300_INTERRUPT_NO_FRONT_CONTROL_2,	
RTX2300_INTERRUPT_NO_COUNT	
}	
} RSENUM8 (Rtx2300InterruptNoType);	

15.10 Rtx2300RelayNoType

Description:	Defines the available relays numbers
C-syntax:	
typedef enum	
{	
RTX2300_RELAYNO_0,	
RTX2300_RELAYNO_1,	
RTX2300_RELAYNO_2,	
RTX2300_RELAYNO_3,	
RTX2300_RELAYNO_4,	
RTX2300_RELAYNO_5,	
RTX2300_RELAYNO_6,	
RTX2300_RELAYNO_7,	
RTX2300_RELAYNO_COUNT,	
}	
} RSENUM8 (Rtx2300RelayNoType);	

15.11 Rtx2300RelayMaskType

Description:	Defines the available relays
C-syntax:	
typedef enum	
{	
RTX2300_RELAYMASK_0 = RSBIT0,	
RTX2300_RELAYMASK_1 = RSBIT1,	
RTX2300_RELAYMASK_2 = RSBIT2,	
RTX2300_RELAYMASK_3 = RSBIT3,	
RTX2300_RELAYMASK_4 = RSBIT4,	
RTX2300_RELAYMASK_5 = RSBIT5,	
RTX2300_RELAYMASK_6 = RSBIT6,	
RTX2300_RELAYMASK_7 = RSBIT7,	
RTX2300_RELAYMASK_ALL = 0xFF	
}	
} RSENUM8 (Rtx2300RelayMaskType);	

15.12 Rtx2300OutputNoType

Description:	Defines the available digital outputs
C-syntax:	
typedef enum	
{	
RTX2300_OUTPUTNO_SOURCE_0,	



RTX2300_OUTPUTNO_SOURCE_1,		
RTX2300_OUTPUTNO_SOURCE_2,		
RTX2300_OUTPUTNO_SOURCE_3,		
RTX2300_OUTPUTNO_SOURCE_4,		
RTX2300_OUTPUTNO_SOURCE_5,		
RTX2300_OUTPUTNO_SOURCE_6,		
RTX2300_OUTPUTNO_SOURCE_7,		
RTX2300_OUTPUTNO_SINK_0,		
RTX2300_OUTPUTNO_SINK_1,		
RTX2300_OUTPUTNO_SINK_2,		
RTX2300_OUTPUTNO_SINK_3,		
RTX2300_OUTPUTNO_SINK_4,		
RTX2300_OUTPUTNO_SINK_5,		
RTX2300_OUTPUTNO_SINK_6,		
RTX2300_OUTPUTNO_SINK_7,		
RTX2300_OUTPUTNO_COUNT,		
} RSENUM8 (Rtx2300OutputNoType);		

15.13 Rtx2300OutputMaskType

Description:	Defines the available digital outputs as a mask, i.e. multiple outputs can be combined by OR'ing them together.	
C-syntax:		
typedef enum		
{		
RTX2300_OUTPUTMASK_SOURCE_0 = RSBIT0,		
RTX2300_OUTPUTMASK_SOURCE_1 = RSBIT1,		
RTX2300_OUTPUTMASK_SOURCE_2 = RSBIT2,		
RTX2300_OUTPUTMASK_SOURCE_3 = RSBIT3,		
RTX2300_OUTPUTMASK_SOURCE_4 = RSBIT4,		
RTX2300_OUTPUTMASK_SOURCE_5 = RSBIT5,		
RTX2300_OUTPUTMASK_SOURCE_6 = RSBIT6,		
RTX2300_OUTPUTMASK_SOURCE_7 = RSBIT7,		
RTX2300_OUTPUTMASK_SINK_0 = RSBIT8,		
RTX2300_OUTPUTMASK_SINK_1 = RSBIT9,		
RTX2300_OUTPUTMASK_SINK_2 = RSBIT10,		
RTX2300_OUTPUTMASK_SINK_3 = RSBIT11,		
RTX2300_OUTPUTMASK_SINK_4 = RSBIT12,		
RTX2300_OUTPUTMASK_SINK_5 = RSBIT13,		
RTX2300_OUTPUTMASK_SINK_6 = RSBIT14,		
RTX2300_OUTPUTMASK_SINK_7 = RSBIT15,		
RTX2300_OUTPUTMASK_ALL_SOURCE = 0x00FF,		
RTX2300_OUTPUTMASK_ALL_SINK = 0xFF00,		
RTX2300_OUTPUTMASK_NONE = 0x00,		
RTX2300_OUTPUTMASK_ALL = 0xFFFF		
} RSENUM16 (Rtx2300OutputMaskType);		

15.14 Rtx2300InputNoType

Description:	Defines the available digital inputs	
C-syntax:		
typedef enum		
{		
RTX2300_INPUTNO_0,		
RTX2300_INPUTNO_1,		
RTX2300_INPUTNO_2,		
RTX2300_INPUTNO_3,		
RTX2300_INPUTNO_4,		
RTX2300_INPUTNO_5,		
RTX2300_INPUTNO_6,		
RTX2300_INPUTNO_7,		
RTX2300_INPUTNO_COUNT,		
} RSENUM8 (Rtx2300InputNoType);		

15.15 Rtx2300InputMaskType

Description:	Defines the available digital inputs as a mask, i.e. multiple inputs can be combined by OR'ing them together.		
C-syntax:			
typedef enum			
{			
	RTX2300_INPUTMASK_0 = RSBIT0,		
	RTX2300_INPUTMASK_1 = RSBIT1,		
	RTX2300_INPUTMASK_2 = RSBIT2,		
	RTX2300_INPUTMASK_3 = RSBIT3,		
	RTX2300_INPUTMASK_4 = RSBIT4,		
	RTX2300_INPUTMASK_5 = RSBIT5,		
	RTX2300_INPUTMASK_6 = RSBIT6,		
	RTX2300_INPUTMASK_7 = RSBIT7,		
	RTX2300_INPUTMASK_ALL = 0xFF		
} RSENUM8 (Rtx2300InputMaskType);			

15.16 Rtx2300FrontLedNoType


Description:	Defines the available LED's on the front.		
C-syntax:			
typedef enum			
{			
	RTX2300_FRONT_LEDNO_0,		
	RTX2300_FRONT_LEDNO_1,		
	RTX2300_FRONT_LEDNO_2,		
	RTX2300_FRONT_LEDNO_COUNT,		
} RSENUM8 (Rtx2300FrontLedNoType);			

15.17 Rtx2300FrontLedMaskType

Description:	Defines the available LED's on the front. Multiple LED's can be combined by OR'ing them together.		
C-syntax:			
typedef enum			
{			
	RTX2300_FRONT_LEDMASK_0 = RSBIT1,		
	RTX2300_FRONT_LEDMASK_1 = RSBIT2,		
	RTX2300_FRONT_LEDMASK_2 = RSBIT3,		
	RTX2300_FRONT_LEDMASK_ALL = (RSBIT1 RSBIT2 RSBIT3)		
} RSENUM8 (Rtx2300FrontLedMaskType);			

15.18 Rtx2300FixtureControlNoType

Description:	Defines the available fixture control outputs.		
C-syntax:			
typedef enum			
{			
	RTX2300_FIXTURE_CONTROLNO_0,		
	RTX2300_FIXTURE_CONTROLNO_1,		
	RTX2300_FIXTURE_CONTROLNO_2,		
	RTX2300_FIXTURE_CONTROLNO_3,		
	RTX2300_FIXTURE_CONTROL_MASK_COUNT,		
} RSENUM8 (Rtx2300FixtureControlNoType);			

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 50 of 54 pages	

15.19 Rtx2300FixtureControlMaskType

Description:	Defines the available fixture control outputs. Multiple outputs can be combined by OR'ing them together.		
C-syntax:	<pre>typedef enum { RTX2300_FIXTURE_CONTROL_MASK_0 = RSBIT4, RTX2300_FIXTURE_CONTROL_MASK_1 = RSBIT5, RTX2300_FIXTURE_CONTROL_MASK_2 = RSBIT6, RTX2300_FIXTURE_CONTROL_MASK_3 = RSBIT7, RTX2300_FIXTURE_CONTROL_MASK_ALL = (RSBIT4 RSBIT5 RSBIT6 RSBIT7) } RSENUM8 (Rtx2300FixtureControlMaskType);</pre>		

15.20 Rtx2300UserDataTypes

Description:	This type contains data transmitted to or from the EEPROM		
C-syntax:	<pre>typedef struct Rtx2300UserDataTypes { rsuint8 Data[16]; } Rtx2300UserDataTypes;</pre>		

15.21 User data constants

Description:	User data size		
C-syntax:	<pre>#define RTX2300_USERDATA_SIZE 100</pre>		
	The number of bytes in the user data area		


15.22 Rtx2300UsbControlNoType

Description:	Defines the available expansion USB control outputs		
C-syntax:	<pre>typedef enum { RTX2300_USB_CONTROL_SCB, RTX2300_USB_CONTROL_CCB, RTX2300_USB_CONTROL_COUNT } RSENUM8 (Rtx2300UsbControlNoType);</pre>		
		Controls SCB_USB_5 in the SCB connector	
		Controls CCB_USB_6 in the CCB connector	

15.23 Rtx2300UutSercomControlType

Description:	Defines the available UUT serial communication modes		
C-syntax:	<pre>typedef enum { RTX2300_UUT_SERCOM_OFF, RTX2300_UUT_SERCOM_NORMAL, RTX2300_UUT_SERCOM_BOOTMODE, RTX2300_UUT_SERCOM_COUNT } RSENUM8 (Rtx2300UutSercomControlType);</pre>		
		UART disabled, bootmode off	
		UART enabled, bootmode off	
		UART disabled, bootmode on	

15.24 Rtx2300FeatureType

Description:	Rtx2300 feature type		
	Technical Documentation Rtx2300		Interface Specification
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2
			Page: 51 of 54 pages

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

C-syntax:	
typedef enum	
{	
RTX2300_FEATURE_PSU_STD_001,	Standard RTX power supply module
RTX2300_FEATURE_FREQCNT_STD_001,	Standard RTX frequency counter module
RTX2300_FEATURE_COUNT	
} RSENUM8 (Rtx2300FeatureType);	

15.25 Rtx2300StatusType

Description:	This type is used to return Rtx2300 status information to the PC.
C-syntax:	
typedef union Rtx2300StatusType	
{	
struct	
{	
rsbitfield InitDone : 1;	The Rtx2300 has been initialized and is ready to accept commands.
rsbitfield Authenticated : 1;	The Rtx2300 system has been authenticated by the coprocessor
rsbitfield DebugMode : 1;	Debug mode is enabled
rsbitfield VerInconMode : 1;	Version inconsistency detected
rsbitfield I2cBus1Blocked : 1;	I2c bus 1 blocking detected
rsbitfield I2cBus2Blocked : 1;	I2c bus 2 blocking detected
rsbitfield I2cDeviceBlocked : 1;	A blocked I2C device has been detected.
rsbitfield SystemIntegrityFault : 1;	A problem with the integrity of the system has been detected. Please contact RTX Telecom.
rsbitfield Reserved : 6;	
} Bits;	
rsuint16 Data;	
} Rtx2300StatusType;	

15.26 Pulse pattern constants


Description:	Pattern value set size
C-syntax:	
#define RTX2300_PATTERN_VALUE_SET_SIZE 16	The number of vaules in a set
Description:	
Number of patterns	
C-syntax:	
#define RTX2300_PATTERN_COUNT 4	The number of output pattern sets.

15.27 Rtx2300PulsePatternValueType

Description:	This type is used to hold a single value in an output pulse pattern.
C-syntax:	
typedef struct Rtx2300PulsePatternValueType	
{	
rsuint16 Time : 14;	
rsuint16 Loop : 1;	
rsuint16 Active : 1;	
} Rtx2300PulsePatternValueType;	

15.28 Rtx2300PulsePatternDefinitionType

Description:	This type is used to hold the definition of an output pulse pattern.
C-syntax:	
typedef struct Rtx2300PulsePatternDefinitionType	
{	
Rtx2300PulsePatternValueType Values[RTX2300_PATTERN_VALUE_SET_SIZE];	See 10.1
} Rtx2300PulsePatternDefinitionType;	

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 52 of 54 pages	

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

15.29 Rtx2300ExpanderNoType

Description:	Defines the possible io expanders in the system.		
C-syntax:			
typedef enum			
{			
	RTX2300_EXPANDER_NO_1 = 0,		
	RTX2300_EXPANDER_NO_2 = 1,		
	RTX2300_EXPANDER_NO_3 = 2,		
	RTX2300_EXPANDER_NO_COUNT		
} RSENUM8 (Rtx2300ExpanderNoType);			

15.30 Rtx2300CapAddrType

Description:	Rtx2300 CAP address type		
C-syntax:			
typedef rsuint8 Rtx2300CapAddrType;	Contains a NULL terminated string.		

15.31 Rtx2300CapDeviceAddrType


Description:	This type is used to hold CAP device addresses		
C-syntax:			
typedef rsuint8 Rtx2300CapDeviceAddrType;			

15.32 Rtx2300ScbBusCfgType

Description:	This type is used to configure the SCB expansion bus.		
C-syntax:			
typedef union Rtx2300ScbBusCfgType			
{			
struct			
{			
	rsbitfield MsbFirst : 1;		MSB of data is clocked out first
	rsbitfield DataInverted : 1;		Data is inverted before transmission
	rsbitfield ClkActiveLow : 1;		Data is latched on falling edge of the clock
	rsbitfield StrobeActiveLow : 1;		Strobe is pulsed low when data is complete.
	rsbitfield StrobeActiveDuringWrite : 1;		Strobe remains active when data is being clocked. Otherwise it is pulsed to active state momentarily after writing.
	rsbitfield OeActiveLow : 1;		<i>Output Enable</i> is active when low.
	rsbitfield OeAlwaysActive : 1;		<i>Output Enable</i> is always active. Otherwise it is inactive while clocking data on the bus.
	rsbitfield DirectMode : 1;		If true the four datalines in the SCB bus is controlled directly by the <i>IoData</i> , <i>IoOe</i> , <i>IoClk</i> and <i>IoStrobe</i> bits below, and all remaining bits in the configuration are ignored.
	rsbitfield IoData : 1;		See <i>DirectMode</i> bit above.
	rsbitfield IoOe : 1;		See <i>DirectMode</i> bit above.
	rsbitfield IoClk : 1;		See <i>DirectMode</i> bit above.
	rsbitfield IoStrobe : 1;		See <i>DirectMode</i> bit above.
	rsbitfield Reserved1 : 4;		Reserved, do not use
} Bits;			
rsuint16 Data;			
} Rtx2300ScbBusCfgType;			

15.33 Rtx2300ManufacturerInfoType

Description:	Rtx2300 manufacturer information type		
C-syntax:			
typedef struct Rtx2300ManufacturerInfoType			

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 53 of 54 pages	

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

{	
Rtx2300DateType ProdDate;	The date of production
Rtx2300SerialNumberType MainboardSerial;	The mainboard serial number, 0 if not applicable
Rtx2300VersionNoType HwVersion;	The hardware version
Rtx2300VersionNoType TestVersion;	The test version
} Rtx2300ManufacturerInfoType;	

15.34 Rtx2300TestCounterType


Description:	Defines the possible test counters in the system.	
C-syntax:		
typedef enum		
{		
RTX2300_TESTCOUNTER_SYSTEM,		
RTX2300_TESTCOUNTER_INSERT,		
RTX2300_TESTCOUNTER_CCB,		
RTX2300_TESTCOUNTER_COUNT		
} RSENUM8 (Rtx2300TestCounterType);		

15.35 Rtx2300TestCounterValueType

Description:	Rtx2300 test count type	
C-syntax:		
typedef rsuint32 Rtx2300TestCounterValueType;	Contains the number of tests made	

15.36 Rtx2300InterruptSenseModeType

Description:	Defines the possible interrupt sense modes	
C-syntax:		
typedef enum		
{		
RTX2300_INT_SENSEMODE_RISING,		Detect a rising edge
RTX2300_INT_SENSEMODE_FALLING,		Detect a falling edge
RTX2300_INT_SENSEMODE_RISING_FALLING,		Detect a rising edge followed by a falling edge
RTX2300_INT_SENSEMODE_FALLING_RISING,		Detect a falling edge followed by a rising edge
RTX2300_INT_SENSEMODE_DISABLED,		Disable interrupt sense
RTX2300_INT_SENSEMODE_COUNT		Disable interrupt sensing.
} RSENUM8 (Rtx2300InterruptSenseModeType);		

	Technical Documentation			Interface Specification	
	Rtx2300				
Date: 2012-09-27	File: Rtx2300Interface_v0050.doc	Ref.: MHP	Rev.: 1.2	Page: 54 of 54 pages	

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