RTSx4 19" Indoor Radio User Guide

Version 1.1
Date 21st June 2018

SAFETY NOTICES

●●●

A CAUTION notice denotes a hazard. It calls attention to an operational procedure, practice or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood & met.

A WARNING notice denotes a hazard. it calls attention to an operational procedure, practice or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully met.

Caution: This document is confidential and may contain concepts that are novel and which may be the subject of current or future invention disclosures, and thus should not be distributed further

CONTENTS

0　INTRODUCTION

This document contains information relating to the use and configuration of the RTSx4 19" Indoor Radio product.

AceAxis RRHs and related products are configured and controlled via a command line interface (CLI), which is controlled using an Ethernet connection. This test interface is referred to as the TIF. This document contains a description of the TIF, instructions for use, an explanation of the setup scripts used during the system integration, and other useful items of information.

1　USER INTERFACE DESCRIPTION

1.1　Establishing a TIF session

A TIF session is established using Telnet (port number 49500) as shown:

```
$ telnet <unit_IP_address> 49500
```

This does not require password authentication. However, as it is designed for machine-to-machine (M2M) communications. Up to 3 simultaneous TIF sessions are supported.

The radio should only be configured once the CPRI is stable, also if the CPRI is disconnected and reconnected the radio should be configured again. Failure to do this could result in a changing and random delay between the two boards, as the delay is measured and compensated for when configured, this can change if the CPRI resyncs and will not be compensated for.

1.2　Usage Overview

The user interface presents a simple two level "directory": The top-level directory contains all of the devices that can be used in the test interface. This list will consist of common devices along with (possibly) devices that are specific to the particular board.

Each device represents a model of that class within the software, and usually represents some physical device (or group of registers used to control a device, for example a carrier). There are also special devices which can be used for testing.

Each device behaves like a "directory", each containing a group of commands that can be performed on that device.

Note that the command prompt will also indicate whether you are at the top level or inside a device. The top level prompt is '->', whereas the device prompt is of the form 'carrier->'.

Device names are case insensitive so that they can be typed all lowercase to make lab use easier.

1.3   Navigating

The following commands can be used to navigate the device directory structure:

- **help** - At any level, running 'help' shows help information

- **cd** - Used either as a stand alone command 'cd' to return to the top level directory, or with a device name (e.g. cd carrier) to change into that device directory

- **ls** - Similar to help, but less verbose. Just shows a list of the entries in the directory along with a one line comment. When displayed by 'ls', devices are shown with a trailing '/' ; whereas commands are shown plain.

1.4   Executing Commands

To execute a command you can either cd to the device then run the command, or specify the device as part of the command name.

For example:

```
->tx.setlfatten 1 10
```

is equivalent to:

```
->cd tx
tx->setlfatten 1 10
```

Note that the available command names depend on the current working directory. If you cd to a subdirectory you will not be able to see commands at the top level. In order to run these commands, the name should be prefixed with '.' (which then makes the command name an 'absolute path'). If you get into the habit of always using the '.' prefix then the absolute command name will always work in whatever working directory you are in:

For example:

```
->.tx.setlfatten 1 10
```

Note: The path separator can be specified either as '/' or as '.' interchangeably. So, assuming we are at the top-level of a hierarchy, the following are all equivalent:

```
->ls /a/b/c/d
->ls a/b/c/d
->ls a.b.c.d
->ls /a.b.c.d
```

1.5   Command Arguments

Commands show in their help the arguments they expect and the order in which they expect them. Commands can be called with either the arguments position dependent (i.e. without argument names) or as name/value pairs in any order:

```
->tx.setlfatten 1 10
```

is equivalent to:

```
->tx.setlfatten index=1 value=10
```

Obviously position dependent calling is simpler for small numbers of arguments, but the name/value pairs approach is easier as the number of arguments rises.

The interface checks that all of the required arguments have been specified and that they are in the correct format (e.g. an INT argument has been set with an integer value rather than a string).

Integers can be specified either as decimal or a hex value starting '0x'. The command will be defined to report return values as either decimal or hex.

NOTE: Like the device names, argument names are case insensitive.

1.6   History / TAB Completion

When a telnet session is started up on the test interface, the interface initially negotiates with the telnet client to put it into character mode. If successful, this allows you to access the history and TAB completion.

Every command you run is added to a circular history buffer (i.e. it is a fixed length buffer that fills up and when the newest command is added, the oldest command is removed). You can cycle through this history using the up and down arrows.

The command line can be edited using the normal backspace, delete, home, end, left, and right arrows.

While you are typing, pressing <TAB> will auto-complete the command if there is a matching unique command that starts with the characters you've typed so far. If there are multiple alternatives, pressing <TAB> a second time will list them.

1.7   Command Response

Every command (including the built-in commands) will respond with the 'Status' enumerated variable. This can take the values:

- **OK** - Command completed successfully
- **ERROR** - Command failed

If the command fails for any reason, a string variable 'Message' will be shown with the error message.

Also, if the command has extra information to report (if, for example, the test reporting level is set to verbose) then the 'Info' string will also be shown containing the information.

2   INSTRUCTIONS FOR USE

2.1   Basic procedure

The RTSx4 Indoor Radio will provide basic Tx and Rx functionality via the CPRI link.

1. Connect RF ports, it is not necessary to connect the Rx ports if only interested in Tx behaviour (and vice-versa).

2. Connect the Fiber ports

3. Apply mains power and wait for the unit to boot, remember the unit should only be configured once the CPRI is stable.

4. Configure the unit as required. Type or paste the required configuration script.

2.2   Disabling Tx and Rx

The following commands can be used to disable either Tx or Rx:

```
.tx.setEnable 1 off
.tx.setEnable 2 off
.rx.setEnable 1 off
.rx.setEnable 2 off
```

2.3   Adjusting Tx Power

The output power can be adjusted by the following means[1]:

```
.tx.setIfatten 1 <value>
```

Where *<value>* is 0 to 31.5 dB in 0.5 dB steps (default = 31.5dB).

An additional VVA-based adjustment is available[2]:

```
.tx.setRfatten 1 <value>
```

Where *<value>* is 0 to 31 dB (default = 31dB)

2.4   Adjusting Rx Gain

The Rx gain adjustment and optimisation is performed automatically.

2.5   Changing carrier object settings

It should be noted that the carrier object parameters (.carrier.set) are only mapped to the hardware when a .carrier.set X state=ENABLE command is executed.

Therefore, to change the carrier settings the command .carrier.set X state=DISABLE should be issued first. Alternatively the system can be rebooted.

2.6   System reboot

To reboot the system enter .system.reboot.

3   EXAMPLE CONFIGURATION SCRIPT

3.1   Script details

Example configuration scripts are provided in Figure 1, 2 and 3. In Figure 1, the unit is configured for 5MHz carrier bandwidth, DL centre frequency of 2655 MHz, and with an UL centre frequency of 2535 MHz. Figure 2 and 3 shows the same configuration but for a 10MHz and 20MHz carrier respectively.

Figure 1 – Example configuration script (5 MHz carrier)

#RTS4 test setup (the maximum RX input is limited to -25dBm)

===

#5MHz carriers

Set up for LTE5, DL=2655 MHz, UL=2535 MHz

#Radio 0

.carrier.set 0 LTE5 TxPower=-30 TxFreq=2655000 RxFreq=2535000 GroupUp=2 GroupDown=2 TxScaling=-13.97 TxContainer[0]=1 TxContainer[1]=3 RxContainer[0]=1 RxContainer[1]=3 state=DISABLE

.carrier.set 0 state=ENABLE

#Radio 1

.carrier.set 0 LTE5 TxPower=-30 TxFreq=2655000 RxFreq=2535000 GroupUp=2 GroupDown=2 TxScaling=-13.97 TxContainer[0]=5 TxContainer[1]=7 RxContainer[0]=5 RxContainer[1]=7 state=DISABLE

.carrier.set 0 state=ENABLE

Figure 2 – Example configuration script (10 MHz carrier)

=====================================

#10MHz carriers

Set up for LTE10, DL=2655 MHz, UL=2535 MHz

#Radio 0

.carrier.set 0 LTE10 TxPower=-30 TxFreq=2655000 RxFreq=2535000 GroupUp=4 GroupDown=4 TxScaling=-13.97 TxContainer[0]=1 TxContainer[1]=5 RxContainer[0]=1 RxContainer[1]=5 state=DISABLE

.carrier.set 0 state=ENABLE

#Radio 1

.carrier.set 0 LTE10 TxPower=-30 TxFreq=2655000 RxFreq=2535000 GroupUp=4 GroupDown=4 TxScaling=-13.97 TxContainer[0]=9 TxContainer[1]=13 RxContainer[0]=9 RxContainer[1]=13 state=DISABLE

.carrier.set 0 state=ENABLE

Figure 3 – Example configuration script (20 MHz carrier)

```
==================================
#20MHz carriers
# Set up for LTE20, DL=2655 MHz, UL=2535 MHz
#Radio 0
.carrier.set 0 LTE20 TxPower=-30 TxFreq=2655000 RxFreq=2535000 GroupUp=8 GroupDown=8
TxScaling=-13.97 TxContainer[0]=1 TxContainer[1]=9 RxContainer[0]=1 RxContainer[1]=9 state=DISABLE
.carrier.set 0 state=ENABLE
#Radio 1
.carrier.set 0 LTE20 TxPower=-30 TxFreq=2655000 RxFreq=2535000 GroupUp=8 GroupDown=8
TxScaling=-13.97 TxContainer[0]=17 TxContainer[1]=25 RxContainer[0]=17 RxContainer[1]=25
state=DISABLE
.carrier.set 0 state=ENABLE
```

Some words of explanation are as follows:

1. .carrier.set sets the properties of the transmit and receiver carriers on Tx1/Tx2/Rx1/Rx2.

    - The first 0 parameter (Index) is a handle to the carrier object. In other RRH products, this would be used to configure multi-carrier scenarios (where each carrier is assigned a unique index). For a single-carrier product, the index can remain set to 0 for all use-cases.

    - Existing carrier parameters can be queried using .carrier.get *<Index>*.

    - The "TxPower" parameter specifies the combined fully-loaded power of the Tx output. However, this has no significance for the RTSx4 19" Indoor Radio since no PAs are fitted.

    - GroupUp and GroupDown specify the number of AxC containers (i.e. samples) in an AxC Container Block according to CPRI 6.0 Mapping Method 1. For LTE5 or 10. This is effectively samples per CPRI frame and should be set to 2 or 4 for these two bandwidth settings.

    - TxScaling is the rms reference level for a fully loaded DL carrier on the CPRI, units dBFS. It therefore controls the downlink digital scaling. The RTSx4 19" Indoor Radio shall use -13.97 dBFS.

    - TxContainer[0] and TxContainer[1] identify the AxC Container Indices (according to CPRI 6.0 Mapping Method 1) used for the DL carriers on Tx1.

        - Any Container Index number is a function of the lower level parameters W, B and CPRI line rate.

        - These parameters use a subscript base of 1 so TxContainer[x]=1 specifies the first container in a CPRI frame.

        - Setting TxContainer[0]=0 is a reserved value which disables that carrier.

- Similarly, RxContainer[0] and RxContainer[1] identify the AxC Container Indices used for the UL carriers on Rx1 and Rx2.

- Performing a .carrier.set 0 state=ENABLE triggers the software to configure the FPGA according to the parameters of the carrier object 0.

2. .tx.setIfatten 1 6 sets the first stage attenuator in Tx1 to 6dB.

3. .tx.setRfatten 1 6 sets the second stage attenuator in Tx1 to 6 dB.

4. .tx.setEnable 1 0 turns Tx1 on.

5. The .carrier.show 0 debug command queries the carrier setup parameters in place. Figure 4 shows the output from a successful execution.

Figure 4 – Example output from .carrier.show 0

```
Status=OK Info="
Carrier [0] DL UL
 * State...........: ENABLE
 * Type.............: LTE10
 * Signal Type.....: TX
 * Loopback........: OFF
 * TX Shutdown.....: [0]:0
 * Tx Centre Freq..: 1987500 kHz
 * Power...........: 30dB
 * Tx Scaling......: -13.97
 * DL Group size...: 4
 * TX Containers...: [0]:1 [1]:5
 * TX Fibres.......: [0]:0 [0]:0
 * Rx Centre Freq..: 2177500 kHz
 * UL Noise........: 0dB
 * RX Containers...: [0]:1 [1]:5
 * RX Fibres.......: [0]:0 [1]:0
 * UL Group size...: 4
```

3.2   Carrier Mapping

The LTE5, LTE10 and LTE20 carrier mappings is achieved using the scripts shown in Figure 1, 2 and 3 however, the unit is capable of being configured with different carrier mappings.

The RTSx4 19" Indoor Radio uses CPRI Mapping Method 1, 15 bit 2's complement IQ data, N_A=1, packed AxCs. The .carrier.set command takes AxC Container Indices with a subscript base of 1, as is common practice among OEMs.

As the word width is a function of CPRI line rate, [W, B] parameters are only applicable to a particular line rate and [W,B] settings which are correct for one line rate may not be legal at another line rate.

Table 1 shows how to convert [W,B] parameters to the Container Indices accepted by the TIF, for CPRI rate 3.

Table 1 - Conversion of [W,B] parameters to Container indices at CPRI Rate 3 (2.4576 Gsps)

LTE Channel Bandwidth	[W,B] parameters		.carrier .set parameters	
	W	B	Group Up/ Group Down	TxContainer/ RxContainer
1.4 / 3	1	0	1	1
1.4 / 3	1	30	1	2
1.4 / 3	2	28	1	3
1.4 / 3	3	28	1	4
1.4 / 3	4	24	1	5
1.4 / 3	5	22	1	6
1.4 / 3	6	20	1	7
1.4 / 3	7	18	1	8
1.4 / 3	8	16	1	9
1.4 / 3	9	14	1	10
1.4 / 3	10	12	1	11
1.4 / 3	11	10	1	12
1.4 / 3	12	8	1	13
1.4 / 3	13	6	1	14
1.4 / 3	14	4	1	15
1.4 / 3	15	2	1	16
5	1	0	2	1
5	2	28	2	3
5	4	24	2	5
5	6	20	2	7
5	8	16	2	9
5	10	12	2	11
5	12	8	2	13
5	14	4	2	15
10	1	0	4	1
10	4	24	4	5
10	8	16	4	9
10	12	8	4	13
20	1	0	8	1
20	8	16	8	9

3.3   Setting up a MIMO pair on independent carrier objects

Under some circumstances it may be desirable to enable and disable carriers on the two ANT ports independently.   This is possible by setting them up on two separate carrier objects.

See Figure 5 for an example. Carrier index 0 is used to support Tx and Rx on antenna line 1 (via TxContainer[0] and RxContainer[0] settings) and carrier index 1 is used to support Tx and Rx on antenna line 2 (via TxContainer[1] and RxContainer[1] settings).   In both cases the TxContainer[] and RxContainer[] antenna carriers not required on that carrier object are set to zero to ensure they are disabled.

Figure 5 – Example of configuring one MIMO pair on two carrier objects

```
.carrier.set Index=0 Type=LTE5 State=DISABLE SigType=NONE TxPower=10 TxFreq=743500
RxFreq=713500 GroupUp=2 GroupDown=2 TxScaling=-13.97 UplinkNoise=-67.1 Loopback=off
TxContainer[0]=1 TxContainer[1]=0 RxContainer[0]=1 RxContainer[1]=0 TxFibre[0]=0 TxFibre[1]=0
RxFibre[0]=0 RxFibre[1]=0

.carrier.set Index=1 Type=LTE5 State=DISABLE SigType=NONE TxPower=10 TxFreq=743500
RxFreq=713500 GroupUp=2 GroupDown=2 TxScaling=-13.97 UplinkNoise=-67.1 Loopback=off
TxContainer[0]=0 TxContainer[1]=3 RxContainer[0]=0 RxContainer[1]=3 TxFibre[0]=0 TxFibre[1]=0
RxFibre[0]=0 RxFibre[1]=0

.carrier.set 0 state=enable

.carrier.set 1 state=enable
```

One advantage of configuring a MIMO pair in this way is that if one antenna line fails due to a Fault, attempts can be made to recover it without having to disable the other line still running.

3.4 Multicarrier Operation

The RRH supports multicarrier operation with up to 2 antenna carriers on each antenna line.

This is achieved by defining multiple carrier objects each with an antenna carrier on the required antenna line. When all carrier objects have been configured, they should be enabled. The RRH SW will then weight the antenna carriers appropriately, initialize the Power Control Loop for the composite DL power, enable the Tx and set the UL gains correctly for all antenna carriers.

Figure 6 shows a multicarrier configuration example. The following points should be noted:

1. Carrier bundles should be contiguous, i.e. adjacent carriers should be spaced from each other by their nominal carrier bandwidth so that the only gaps are associated with in-channel LTE guard space. (Present Releases permit configuration with a gap but the system is not guaranteed to meet 3GPP mask specifications with split bundles).
2. A maximum carrier bundle bandwidth of 20 MHz should be observed.
3. Antenna Carrier CPRI data on adjacent channels of the same antenna line should not be identical as the high cross correlation may increase the peak power unacceptably and degrade spectral mask. A different cellID (or time delay offset of >5 CPRI samples on the same CPRI signal) is sufficient to achieve the required decorrelation.
4. The TxPower parameter on each carrier object relates to the individual Tx antenna carriers defined in that object. Therefore the power sum of the TxPower parameters across two objects on one antenna line should add up to no more than +40.0 dBm.

Figure 6 – Multicarrier configuration example

```
.carrier.set Index=0 LTE5 State=DISABLE TX TxPower=0 TxFreq=738500 RxFreq=708500 GroupUp=2
GroupDown=2 TxScaling=-13.97 TxContainer[0]=1 TxContainer[1]=3 RxContainer[0]=1 RxContainer[1]=3
TxFibre[0]=0 TxFibre[1]=0 RxFibre[0]=0 RxFibre[1]=0 UplinkNoise=0 LoopBack=OFF

.carrier.set Index=1 LTE5 State=DISABLE TX TxPower=0 TxFreq=743500 RxFreq=713500 GroupUp=2
GroupDown=2 TxScaling=-13.97 TxContainer[0]=5 TxContainer[1]=7 RxContainer[0]=5 RxContainer[1]=7
TxFibre[0]=0 TxFibre[1]=0 RxFibre[0]=0 RxFibre[1]=0 UplinkNoise=0 LoopBack=OFF

.carrier.set 0 state=enable

.carrier.set 1 state=enable
```

4　MISCELLANEOUS INFORMATION

4.1　Software Image Management

4.1.1　Installing a Build

The software image components are installed within a non-volatile area of the Linux filesystem. The total amount of non-volatile storage available is 64 Mbytes.

Although the TIF provides a comprehensive set of image management commands which allow component-wise installation and manipulation of the configuration, it is intended that builds will be released as packages. Each release consists of the following files:

- A package file, which provides an inventory of all code components relevant to the packaged build.
- An application image file
- A library file
- An OS file
- MD5 hash files for verifying build integrity.

The image management commands are in the /image directory of the TIF.

To install a packaged build, the following steps are required:

1. Place the required files on an FTP server accessible to the RRH. To which AceAxis shall transfer software images when available.

2. Perform a .image.url *<link>* command,

    where *<link>* is an FTP URL pointing to the directory on the FTP server where the files are located. In the case of anonymous FTP, the URL will be of the form <ftp://IP_address:21/packages_folder_path/>. If a username and password is required then the URL will be of the form <ftp://username:password@IP_address:21/packages_folder_path/>. Note that in the above URL examples, the commonly-used default FTP port number of 21 has been used. Please modify as required by local rules.

3. Type .image.installFile *<name>* where *<name>* is the name of the package (example: ECB-RTS-1.1-5.pkg).

4. The software will then search its local directories for the package name and code components. Any parts of the build (including the overall package inventory) which are not locally available will be fetched from the FTP server: such downloads will be flagged by a status message including a % progress indication.

5. When complete the system will return with either an OK or ERROR response depending on whether the installation was successful.

6. If successful, the unit should be rebooted (.system.reboot) to complete the installation.

4.1.2   Other useful /image commands

.image.show displays the details of the currently installed image. Old software images can be removed using .image.clean which removes all images apart from the current image and the previous one.

4.2   Hostname

The 19" Indoor Radio has a hostname based on the unit's product and serial number.

By default the unit attempts to acquire an IP address for the Ethernet via DHCP when it boots.

4.3   Checking board temperatures

The temperatures of the Tx paths can be checked using .tx.getTemp 1 or .tx.getTemp 2. Similarly the temperatures of Rx1 and Rx2 can be checked using .rx.getTemp 1 and .rx.getTemp 2. The acceptable operating range for these is -12°C. to +85°C.

The FPGA junction temperature can be checked using .fpga.getTemperature. The acceptable operating range for this is -12°C. to +100°C.

5   EXTERNAL CONNECTIONS

Table 2 descripes the connectors and labels, figure 7 shows the connections provided on the front and back panels:

Front Panel	8 x N-type RF connectors, female (4 x Rx, 4 x Tx) to SMA, female (internal connector)	
	Center-Center Spacing: ≥ 38mm	
	Status LEDs: "GREEN" to indicate RF module is operational	
	Silkscreen: AceAxis logo, 8x RF connectors,	
	Labels: 2x RF module SN, Frequency Range	
Back Panel	Optical Port (CPRI, LC Single Mode, 1310nm uplink/ downlink), 2x Ethernet, 2x Serial Port, 2x Reset buttons	
	Power Input: IEC-C14; Ground Lug with bolt/washer or equivalent	
	Label(s): Safety/EMC verbage, FO-0/FO-1, CTR-0/CTR-1, AceAxis SN	
Top Lid	Label: Compliance Information	
Airflow Opening	Per thermal design	
Fan Openings	Per thermal design	

Table 2   Connectors, LED's and Labels

The front and rear panels are shown in Figure 7.

Figure 7 – RTSx4 19" Indoor Radio front and rear panel connections

END OF DOCUMENT