



MEDIATEK

META Calibration Library

Programming Guide

Customer Support

6001

Doc No: CS6001-H4A-PGD-V1.5EN

Version: V1.5

Release date: 2017-07-27

Classification: Confidential B

© 2008 -- 2009 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc.

Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

Specifications are subject to change without notice.

META Calibration Library
Programming Guide

MediaTek Inc.

Postal address

No. 1, Dusing 1st Rd. , Hsinchu Science
Park, Hsinchu City, Taiwan 30078

MTK support office address

No. 1, Dusing 1st Rd. , Hsinchu Science
Park, Hsinchu City, Taiwan 30078

Internet

<http://www.mediatek.com/>

Document Revision History

Revision	Date	Description
V0.1	2011-06-16	Initial Draft
V1.0	2011-06-29	Update the following sections <ol style="list-style-type: none"> General Status Code WCDMA AFC Status Code WCDMA AGC Status Code WCDMA APC Status Code WCDMA FHC Status Code WCDMA NSFT Status Code WCDMA HSDPA NSFT Status Code WCDMA HSUPA NSFT Status Code
V1.1	2012-03-29	Add notes of METACalibrationLibrary_RegisterCallBack()
V1.2	2012-06-06	<ol style="list-style-type: none"> Add the following sections <ol style="list-style-type: none"> GSM TADC Status Code GSM CAP ID Status Code GSM AFC Status Code GSM AGC Status Code GSM W Calibration Status Code GSM APC Status Code EDGE APC Status Code GSM Feedback DAC Status Code GSM Slope Skew Status Code GSM TRX offset Status Code GSM TX IQ Status Code Update calibration items Add the mapping between calibration library items and META tool factory UI calibration items
V1.3	2012-07-31	<ol style="list-style-type: none"> Add description of supporting chips and instruments Add GMSK/EPSK power check testing item Add description of share library configures
V1.4	2014-02-06	<ol style="list-style-type: none"> Add LTE part
V1.5	2017-06-14	Initial WTG external document

Mediatek Confidential

© 2017 Mediatek Inc.

Classification: Confidential B

Table of Contents

Document Revision History.....	3
Table of Contents.....	4
Lists of Tables	7
Lists of Figures	13
1 Introduction	14
1.1 Overview	14
1.2 META Calibration Library Architecture	15
1.3 Supporting Chip(s) and Instrument(s)	15
2 Error Code Listing	16
2.1 General Status Code.....	16
2.2 GSM & EDGE Calibration Status	16
2.2.1 GSM TADC Status Code	16
2.2.2 GSM CAP ID Status Code	17
2.2.3 GSM AFC Status Code.....	17
2.2.4 GSM AGC Status Code	18
2.2.5 GSM W calibration Status Code	18
2.2.6 GSM APC Status Code.....	19
2.2.7 EDGE APC Status Code	19
2.2.8 GSM AD6546(Feedback DAC and Slope Skew) Status Code.....	21
2.2.9 GSM TRX offset Status Code.....	21
2.2.10 GSM TX IQ Status Code	22
2.3 WCDMA Calibration Status Code	22
2.3.1 WCDMA TADC Status Code	22
2.3.2 WCDMA AFC Status Code.....	23
2.3.3 WCDMA AGC Status Code	23
2.3.4 WCDMA APC Status Code	24
2.3.5 WCDMA FHC Status Code	25
2.3.6 WCDMA NSFT Status Code.....	26
2.3.7 WCDMA HSDPA NSFT Status Code.....	27

2.3.8	WCDMA HSUPA NSFT Status Code	27
2.4	LTE Calibration Status Code	28
2.4.1	LTE Common Status Code	28
2.4.2	LTE TADC Status Code	28
2.4.3	LTE AFC Status Code	28
2.4.4	LTE AGC Status Code	29
2.4.5	LTE APC Status Code	30
2.4.6	LTE FHC Status Code	31
2.4.7	LTE NSFT Status Code	32
3	Exported Functions	34
3.1	The Terminology of Function Descriptions	34
3.1.1	The Meaning of Parameter Table:	34
3.2	Exported Calibration Functions	34
3.2.1	METACalibrationLibrary_Init	34
3.2.2	METACalibrationLibrary_RegisterCallBack	35
3.2.3	METACalibrationLibrary_DeInit	36
3.2.4	METACalibrationLibrary_Start	37
3.2.5	METACalibrationLibrary_StartWithRetunedStatusCode	44
3.2.6	METACalibrationLibrary_GetStatusCodeString	45
3.2.7	METACalibrationLibrary_GetStatusType	46
3.2.8	METACalibrationLibrary_GetLastError	47
4	Calibration Library Configure Settings	48
4.1	Calibration Configures	48
4.1.1	Calibration Retry Configure	48
4.2	NSFT Configures	48
4.2.1	NSFT Retry Configure	48
4.2.2	WCDMA NSFT TX Testing Items Switch	48
5	Sample code	49
5.1	META Calibration Library initialization	49
5.2	Start calibration via META Calibration Library	50

5.3	META Calibration Library Deinit.....	52
6	Instrument library	53
6.1	The return values of the Instrumnet functions	53
6.2	GPIB Control and Logging functions.....	54
6.2.1	Exported functions	54
6.2.2	Sample Code.....	56
6.3	Exported Instrument calibration control functions.....	57
6.3.1	Exported Instrument Common Control Functions	57
6.3.2	GSM & EDGE Exported Instrument Control functions	68
6.3.3	WCDMA Exported Instrument Control functions.....	100
6.3.4	TDSCDMA Exported Instrument Control functions	131
6.3.5	LTE Exported Instrument Control functions	149
6.4	Customized Instrument Porting Interface.....	177
6.4.1	Callback Register Entry	178
6.4.2	Sample Code.....	192
7	Appendix	199
7.1	The mapping between calibration library items and META tool factory UI calibration items.....	199
7.1.1	GSM/EDGE calibration items.....	199
7.1.2	WCDMA calibration items	199
7.1.3	TD-SCDMA calibration items	200
7.1.4	LTE calibration items	200



Lists of Tables

Table 3-1 General status code.....	16
Table 3-2 GSM TADC status code	16
Table 3-3 GSM CAP ID status code	17
Table 3-4 GSM AFC status code	17
Table 3-5 GSM AGC status code.....	18
Table 3-6 GSM W Calibration status code.....	18
Table 3-7 GSM APC status code	19
Table 3-8 EDGE APC status code	19
Table 3-9 GSM AD6546 status code	21
Table 3-10 GSM TRX offset status code	21
Table 3-11 GSM TX IQ status code	22
Table 3-12 WCDMA TADC status code	22
Table 3-13 WCDMA AFC status code	23
Table 3-14 WCDMA AGC status code.....	23
Table 3-15 WCDMA APC status code	24
Table 3-16 WCDMA FHC status code	25
Table 3-17 WCDMA NSFT status code	26
Table 3-18 WCDMA HSDPA NSFT status code.....	27
Table 3-19 WCDMA HSUPA NSFT status code.....	27
Table 3-20 LTE common status code.....	28
Table 3-21 LTE TADC status code.....	28
Table 3-22 LTE AFC status code.....	28
Table 3-23 LTE AGC status code	29
Table 3-24 LTE APC status code.....	30
Table 3-25 LTE FHC status code.....	31
Table 3-26 LTE NSFT status code.....	32
Table 4-1 The meaning of parameter table sample	34
Table 4-2 METACalibrationLibrary init parameter.....	35
Table 4-3 METACalibrationLibrary init return value	35
Table 4-4 METACalibrationLibrary registerCallBack parameter	35
Table 4-5 METACalibrationLibrary registerCallBack return value.....	35
Table 4-6 METACalibrationLibrary delnit parameter	37
Table 4-7 METACalibrationLibrary delnit return value.....	37

Table 4-8 METACalibrationLibrary start parameter	44
Table 4-9 METACalibrationLibrary start return value	44
Table 4-10 METACalibrationLibrary startWithRetrunedStatusCode parameter	45
Table 4-11 METACalibrationLibrary startWithRetrunedStatusCode return value	45
Table 4-12 METACalibrationLibrary getStatusCodeString parameter	46
Table 4-13 METACalibrationLibrary getStatusCodeString return value	46
Table 4-14 METACalibrationLibrary getStatusType parameter	46
Table 4-15 METACalibrationLibrary getStatusType return value	47
Table 4-16 METACalibrationLibrary getLastError parameter	47
Table 4-17 METACalibrationLibrary getLastError return value	47
Table 5-1 Calibration retry configure	48
Table 5-2 NSFT Retry configure	48
Table 5-3 WCDMA NSFT TX testing items switch	48
Table 7-1 WCDMA NSFT TX testing items switch	53
Table 7-2 RCT viQueryf parameter	54
Table 7-3 RCT viRead parameter	55
Table 7-4 RCT viWrite parameter	55
Table 7-5 GpibLogFunction parameter	56
Table 7-6 RCTLIB common initialize parameter	57
Table 7-7 RCTLIB common setApplicationFormat parameter	61
Table 7-8 RCTLIB common setOperatingMode parameter	62
Table 7-9 RCTLIB common getDeviceString parameter	64
Table 7-10 RCTLIB common getDLLVer parameter	66
Table 7-11 RCTLIB common getInsAppFormatSupport parameter	66
Table 7-12 RCTLIB common switchPort parameter	67
Table 7-13 RCTLIB occupyHandler parameter	67
Table 7-14 RCTLIB GGE Cableloss settings parameter	68
Table 7-15 RCTLIB GGE ConfigCellPower parameter	68
Table 7-16 RCTLIB GGE ConfigAnalyzerFrequencyOffset parameter	69
Table 7-17 RCTLIB GGE CAPID preSettings parameter	70
Table 7-18 RCTLIB GGE CAPID Iteration parameter	71
Table 7-19 RCTLIB GGE AFC PreSettings parameter	72
Table 7-20 RCTLIB GGE AGC preSettings parameter	73
Table 7-21 RCTLIB GGE AGC ChangeCellBand parameter	73
Table 7-22 RCTLIB GGE AGC ChangeCellBand parameter	73

Table 7-23 RCTLIB GGE APCDCOffset preSettings parameter	74
Table 7-24 RCTLIB GGE APCDCOffset ChangeCellBand parameter	74
Table 7-25 RCTLIB GGE APCDCOffset Iteration parameter	75
Table 7-26 RCTLIB GGE EDGE APCDCOffset preSettings parameter	75
Table 7-27 RCTLIB GGE EDGE APCDCOffset ChangeCellBand parameter	76
Table 7-28 RCTLIB GGE EDGE APCDCOffset iteration parameter	76
Table 7-29 RCTLIB GGE FBDAC presetsettings parameter	77
Table 7-30 RCTLIB GGE FBDAC iteration parameter	77
Table 7-31 RCTLIB GGE TXIQ presetsettings parameter	78
Table 7-32 RCTLIB GGE TXIQ ChangeBand parameter	79
Table 7-33 RCTLIB GGE TXIQ iteration parameter	79
Table 7-34 RCTLIB GGE TXSlopeSkew presetsettings parameter	80
Table 7-35 RCTLIB GGE TXSlopeSkew iteration parameter	81
Table 7-36 RCTLIB GGE TRXOffset presetsettings	82
Table 7-37 RCTLIB GGE TRXOffset InitAFC parameter	83
Table 7-38 RCTLIB GGE TRXOffset iteration parameter	83
Table 7-39 FHC three stage frames	84
Table 7-40 RCTLIB GGE FHC DTS iteration parameter	85
Table 7-41 RCTLIB GGE FHC UTS iteration parameter	88
Table 7-42 RCTLIB GGE FHC UTS FetchResult parameter	90
Table 7-43 RCTLIB GGE NSFT presetsettings parameter	91
Table 7-44 RCTLIB GGE NSFT GMSKinit parameter	91
Table 7-45 RCTLIB GGE NSFT EPSKinit parameter	92
Table 7-46 RCTLIB GGE NSFT BERinit parameter	93
Table 7-47 RCTLIB GGE NSFT ChangePCL parameter	93
Table 7-48 RCTLIB GGE NSFT ReadGMSKPerformance parameter	94
Table 7-49 RCTLIB GGE NSFT FetchBER parameter	96
Table 7-50 RCTLIB GGE NSFT ReadEPSKPerformance parameter	97
Table 7-51 RCTLIB GGE GET SpecificSettings parameter	99
Table 7-52 RCTLIB WCDMA ConfigCellPower parameter	100
Table 7-53 RCTLIB WCDMA AFC presetting parameter	101
Table 7-54 RCTLIB WCDMA AGC presetting parameter	102
Table 7-55 RCTLIB WCDMA AGC ChangeCellBand parameter	102
Table 7-56 RCTLIB WCDMA AGC ChangeChannel parameter	103
Table 7-57 RCTLIB WCDMA AGC ChangeCellPower parameter	103

Table 7-58 RCTLIB WCDMA APC presetting parameter	104
Table 7-59 RCTLIB WCDMA APC ChangeCellBand parameter.....	104
Table 7-60 RCTLIB WCDMA APC ChangeChannel parameter.....	105
Table 7-61 RCTLIB WCDMA APC ChangeExpectedPower parameter.....	105
Table 7-62 RCTLIB WCDMA APC FetchResult parameter	106
Table 7-63 RCTLIB WCDMA FHC presetting parameter	106
Table 7-64 RCTLIB WCDMA FHC StartIteration parameter	107
Table 7-65 RCTLIB WCDMA FHC FetchResult parameter	107
Table 7-66 RCTLIB WCDMA NSFT presetting parameter.....	108
Table 7-67 RCTLIB WCDMA NSFT InitiateTestCase parameter.....	110
Table 7-68 RCTLIB WCDMA NSFT TPC parameter	111
Table 7-69 RCTLIB WCDMA NSFT FetchTXPerformanceMaxPower parameter	112
Table 7-70 RCTLIB WCDMA NSFT FetchTXPerformanceMinPower parameter.....	115
Table 7-71 RCTLIB WCDMA NSFT InitiateILPCTestCase parameter.....	115
Table 7-72 RCTLIB WCDMA NSFT FetchILPCResult parameter	116
Table 7-73 RCTLIB WCDMA NSFT ChangeCellPower parameter.....	118
Table 7-74 RCTLIB WCDMA NSFT ReadBER parameter.....	119
Table 7-75 RCTLIB WCDMA NSFT PRACH presetting parameter.....	120
Table 7-76 RCTLIB WCDMA NSFT PRACH InitiateTestCase parameter.....	122
Table 7-77 RCTLIB WCDMA NSFT PRACH FetchTestCase parameter	123
Table 7-78 RCTLIB WCDMA HSDPA NSFT presetting parameter	123
Table 7-79 RCTLIB WCDMA HSDPA NSFT InitiateTestCase parameter	126
Table 7-80 RCTLIB WCDMA HSDPA NSFT FetchResult parameter.....	128
Table 7-81 RCTLIB WCDMA HSUPA NSFT presetting parameter.....	129
Table 7-82 RCTLIB WCDMA HSUPA NSFT InitiateTestCase parameter	129
Table 7-83 RCTLIB WCDMA HSUPA NSFT FetchResult parameter	130
Table 7-84 RCTLIB WCDMA GET SpecificSettings parameter.....	130
Table 7-85 RCTLIB TDSCDMA ConfigTesterBeforeAFC parameter	131
Table 7-86 RCTLIB TDSCDMA MeasureAFC parameter	132
Table 7-87 RCTLIB TDSCDMA ConfigTesterBeforeAGC parameter	133
Table 7-88 RCTLIB TDSCDMA AGC ChangeCellPower parameter	133
Table 7-89 RCTLIB TDSCDMA AGC ChangeCellBand parameter	134
Table 7-90 RCTLIB TDSCDMA AGC ChangeChannel parameter	134
Table 7-91 RCTLIB TDSCDMA ConfigTesterBeforeAPC parameter	135
Table 7-92 RCTLIB TDSCDMA APC ChangeCellBand parameter.....	135

Table 7-93 RCTLIB TDSCDMA APC MeasurePower parameter	136
Table 7-94 RCTLIB TDSCDMA ConfigTesterBeforeFHC parameter	137
Table 7-95 RCTLIB TDSCDMA FHC StartIteration parameter	137
Table 7-96 RCTLIB TDSCDMA FHC FetchResult parameter	139
Table 7-97 RCTLIB TDSCDMA ConfigTesterCommonBeforeNSFT parameter	140
Table 7-98 RCTLIB TDSCDMA ConfigTesterForNSFT parameter	140
Table 7-99 RCTLIB TDSCDMA MeasureTPCForNSFT parameter	141
Table 7-100 RCTLIB TDSCDMA ConfigTesterForNSFTBer parameter	144
Table 7-101 RCTLIB TDSCDMA MeasureLBerForNSFT parameter	145
Table 7-102 RCTLIB TDSCDMA ConfigTesterCommonBeforeFT parameter	146
Table 7-103 RCTLIB TDSCDMA ConfigTesterForFT parameter	146
Table 7-104 RCTLIB TDSCDMA MeasureTPCForFT parameter	147
Table 7-105 RCTLIB TDSCDMA ConfigTesterForFTBer parameter	147
Table 7-106 RCTLIB TDSCDMA MeasureLBerForFT parameter	148
Table 7-107 RCTLIB TDSCDMA GET SpecificSettings parameter	149
Table 7-108 RCTLIB LTE AFC presetting parameter	150
Table 7-109 RCTLIB LTE AFC FetchResult parameter	151
Table 7-110 RCTLIB LTE AGC ChangeCellBand parameter	152
Table 7-111 RCTLIB LTE AGC ChangeFrequency parameter	152
Table 7-112 RCTLIB LTE AGC ChangeCellPower parameter	153
Table 7-113 RCTLIB LTE APC presetting parameter	154
Table 7-114 RCTLIB LTE APC ChangeCellBand parameter	154
Table 7-115 RCTLIB LTE APC ChangeUIFrequency parameter	155
Table 7-116 RCTLIB LTE APC ChangeExpectedPower parameter	155
Table 7-117 RCTLIB LTE APC FetchResult parameter	156
Table 7-118 RCTLIB LTE FHC presetting parameter	157
Table 7-119 RCTLIB LTE FHC StartIteration parameter	158
Table 7-120 RCTLIB LTE FHC FetchResult parameter	161
Table 7-121 RCTLIB LTE NSFT presetting parameter	163
Table 7-122 RCTLIB LTE NSFT TX ListMode parameter	164
Table 7-123 RCTLIB LTE NSFT TX FetchListMode parameter	168
Table 7-124 RCTLIB LTE NSFT ChangeCellPower parameter	170
Table 7-125 RCTLIB LTE NSFT InitiateBER parameter	171
Table 7-126 RCTLIB LTE NSFT GetCableLoss parameter	171
Table 7-127 RCTLIB LTE GET SpecificSettings parameter	172



Lists of Tables

Table 7-128 Common callback function parameter 178

Table 7-129 GSM & EDGE callback function parameter 179

Table 7-130 WCDMA callback function parameter 183

Table 7-131 TDSCDMA callback function parameter 187

Table 7-132 LTE Callback function parameter 190



Lists of Figures

Figure 1-1 Architecture overview of META calibration library	15
Figure 7-1 Class hirachey of instrument library	53
Figure 7-2 RCTLIB GGE FHC UTS iteration	88
Figure 7-3 TX/RX arrangements of FDD band	158
Figure 7-4 TX/RX arrangements of TDD band	158
Figure 7-5 Customized instrument porting interface	178
Figure 8-1 GSM/EDGE calibration items	199
Figure 8-2 WCDMA calibration items	199
Figure 8-3 TD-SCDMA calibration items.....	200
Figure 8-4 LTE calibration items	200

1 Introduction

META Calibration Library provides the MODEM/RF calibration API for customers to adopt the MediaTek's solution in factory calibration procedure.

1.1 Overview

The META Calibration Library consists of 3 parts

1. Calibration Library

The calibration library is the core of the META Calibration library. It implements the MediaTek's RF calibration solution.

2. Instrument Library

The instrument library provides the corresponding instrument control procedure that mapped to specific RF calibration procedure. The instrument is composed of unified instrument control procedure that can easily extended by customer to add 3rd party instrument support.

3. Handset Control

The handset control is the control channel to the MediaTek's handsets.

1.2 META Calibration Library Architecture

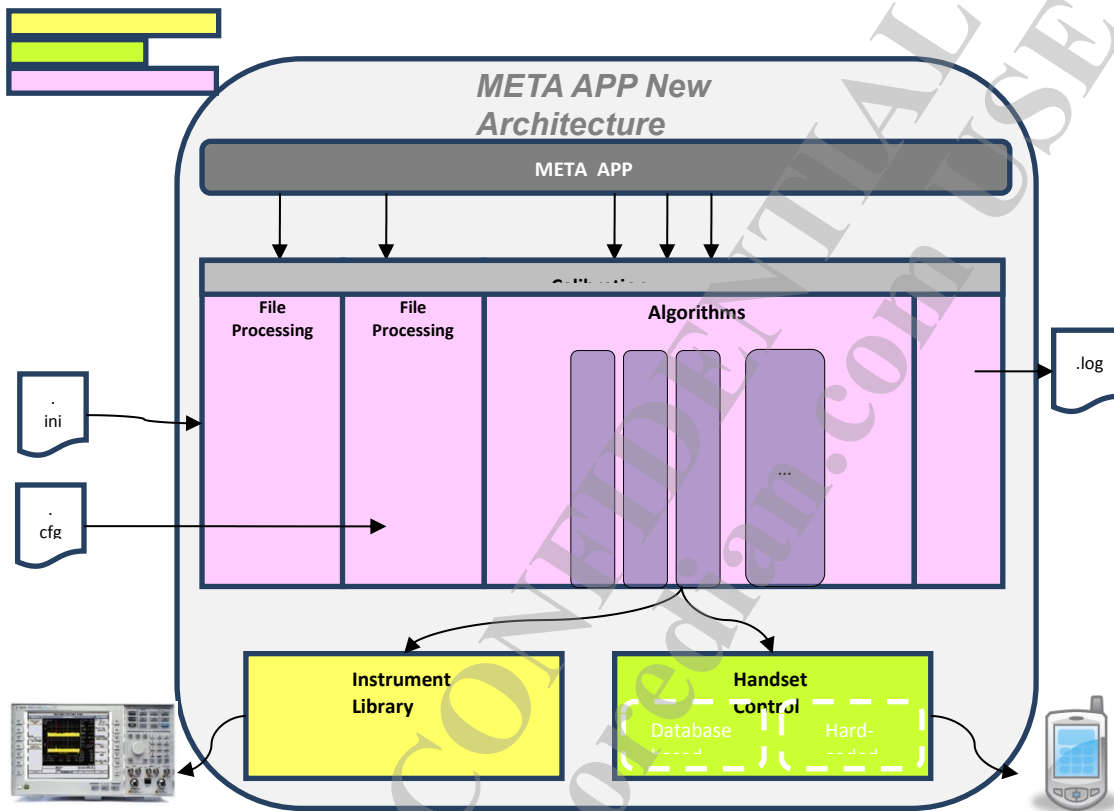


Figure 1-1 Architecture overview of META calibration library

1.3 Supporting Chip(s) and Instrument(s)

※ TC: Traditional Calibration ※ FHC: Fast Handset Calibration ※ O: Support ※ X: Not Support

Instrument	Agilent8960		CMU200		CMW500		StartPoint6010	
CHIP	TC	FHC	TC	FHC	TC	FHC	TC	FHC
MT6162	O	O	O	O	X	O	X	X
MT6163	O	O	O	O	X	O	X	X
MT6250	O	O	O	O	X	O	X	X
AST3001	O	O	X	X	X	X	O	X
AST2001	O	O	X	X	X	X	O	X

The waveforms of CMW500 should be put in "D:\WaveForm_MTK\"

2 Error Code Listing

Below are the status code used in META calibration library. Status codes are defined in METACalibrationLibrary.h

2.1 General Status Code

Table 2-1 General status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_SUCCESS	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	General: Operation is successful
E_METACalibrationLibrary_STATUS_LIB_INIT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Library initialization failed
E_METACalibrationLibrary_STATUS_INVALID_HANDLE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Access to invalid calibration handle
E_METACalibrationLibrary_STATUS_CREATE_LOG_FILE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Create log file failed
E_METACalibrationLibrary_STATUS_CREATE_REPORT_FILE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Create report file failed
E_METACalibrationLibrary_STATUS_GENERATE_OUTPUT_FILE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Create output file failed
E_METACalibrationLibrary_STATUS_CFG_FILE_INVALID_ID	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Invalid CFG settings
E_METACalibrationLibrary_STATUS_INI_FILE_INVALID_ID	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Invalid INI settings
E_METACalibrationLibrary_STATUS_GET_NVRAM_RECORD_LENGTH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Get NVRAM record length failed
E_METACalibrationLibrary_STATUS_COMPOSE_NVRAM_BUFFER_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Compose data field to NVRAM buffer failed
E_METACalibrationLibrary_STATUS_WRITE_NVRAM_ITEM_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Write NVRAM item failed
E_METACalibrationLibrary_STATUS_WRITE_OUTPUT_FILE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Write output file failed
E_METACalibrationLibrary_STATUS_TESTER_CTRL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Tester control failed
E_METACalibrationLibrary_STATUS_TESTER_CTRL_DEINIT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Instrument library de-init failed
E_METACalibrationLibrary_STATUS_USER_TERMINATED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Calibration is terminated by user
E_METACalibrationLibrary_STATUS_STOP_DUT_RF_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Stop DUT RF failed
E_METACalibrationLibrary_STATUS_READ_NVRAM_ITEM_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Read NVRAM item failed
E_METACalibrationLibrary_STATUS_GET_NVRAM_BUFFER_FIELD_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	General: Get data field from NVRAM buffer failed

2.2 GSM & EDGE Calibration Status

2.2.1 GSM TADC Status Code

Table 2-2 GSM TADC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GGE_TADC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GGE temperature sensor calibration: calibration start
E_METACalibrationLibrary_STATUS_GGE_TADC_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GGE temperature sensor calibration: measurement failed
E_METACalibrationLibrary_STATUS_GGE_TADC_GET_NVRAM_RECORD_LENGTH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GGE temperature sensor calibration: gets the NVRAM record length failed
E_METACalibrationLibrary_STATUS_GGE_TADC_COMPOSE_NVRAM_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GGE temperature sensor calibration: composes the NVRAM buffer failed
E_METACalibrationLibrary_STATUS_GGE_TADC_WRITE_NVRAM_INITIAL_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GGE temperature sensor calibration: writes the NVRAM initial value failed
E_METACalibrationLibrary_STATUS_GGE_TADC_WRITE_NVRAM_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GGE temperature sensor calibration: writes the NVRAM calibration value failed
E_METACalibrationLibrary_STATUS_GGE_TADC_INVALID_CURRENT_TEMPERATURE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GGE temperature sensor calibration: current temperature's setting is invalid
E_METACalibrationLibrary_STATUS_GGE_TADC_CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GGE temperature sensor calibration: check start
E_METACalibrationLibrary_STATUS_GGE_TADC_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GGE temperature sensor calibration: check failed
E_METACalibrationLibrary_STATUS_GGE_TADC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GGE temperature sensor calibration: calibration done
E_METACalibrationLibrary_STATUS_GGE_TADC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GGE temperature sensor calibration: parse TADC config file failed

2.2.2 GSM CAP ID Status Code

Table 2-3 GSM CAP ID status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM CAP ID Calibration: Start
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM CAP ID Calibration: Done
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM CAP ID Calibration: CAP ID calibration result range check failed
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_BAND_SELECT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM CAP ID Calibration: Set band indicator failed
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_NB_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM CAP ID Calibration: NB TX failed
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_DYNAMIC_RANGE_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM CAP ID Calibration: Dynamic range check of AFC failed
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_GET_CENTER_FREQUENCY_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM CAP ID Calibration: Get center frequency of an ARFCN failed
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_GET_CAPABILITY3_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM CAP ID Calibration: Getting software capability 3 failed
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_GET_TD_LID_LEN_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM CAP ID Calibration: Getting TD LID struct length failed
E_METACalibrationLibrary_STATUS_GSM_CAP_ID_COMPOSE_TD_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM CAP ID Calibration: Composing TD struct failed

2.2.3 GSM AFC Status Code

Table 2-4 GSM AFC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_AFC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Start

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_AFC_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Frequency error measure failed
E_METACalibrationLibrary_STATUS_GSM_AFC_CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Initial dac range check start
E_METACalibrationLibrary_STATUS_GSM_AFC_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Initial dac range check failed
E_METACalibrationLibrary_STATUS_GSM_AFC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Done
E_METACalibrationLibrary_STATUS_GSM_AFC_GET_CAPABILITY3_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Getting software capability 3 failed
E_METACalibrationLibrary_STATUS_GSM_AFC_GET_TD_LID_LEN_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Getting TD LID struct length failed
E_METACalibrationLibrary_STATUS_GSM_AFC_COMPOSE_TD_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Composing TD struct failed
E_METACalibrationLibrary_STATUS_GSM_AFC_LOW_HIGH_BAND_SWITCH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Low/High band switch failed

2.2.4 GSM AGC Status Code

Table 2-5 GSM AGC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_AFC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Start
E_METACalibrationLibrary_STATUS_GSM_AFC_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Frequency error measure failed
E_METACalibrationLibrary_STATUS_GSM_AFC_CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Initial dac range check start
E_METACalibrationLibrary_STATUS_GSM_AFC_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Initial dac range check failed
E_METACalibrationLibrary_STATUS_GSM_AFC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Done
E_METACalibrationLibrary_STATUS_GSM_AFC_GET_CAPABILITY3_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Getting software capability 3 failed
E_METACalibrationLibrary_STATUS_GSM_AFC_GET_TD_LID_LEN_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Getting TD LID struct length failed
E_METACalibrationLibrary_STATUS_GSM_AFC_COMPOSE_TD_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Composing TD struct failed
E_METACalibrationLibrary_STATUS_GSM_AFC_LOW_HIGH_BAND_SWITCH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Low/High band switch failed

2.2.5 GSM W calibration Status Code

Table 2-6 GSM W Calibration status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_AFC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Start
E_METACalibrationLibrary_STATUS_GSM_AFC_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Frequency error measure failed
E_METACalibrationLibrary_STATUS_GSM_AFC_CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Initial dac range check start
E_METACalibrationLibrary_STATUS_GSM_AFC_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Initial dac range check failed
E_METACalibrationLibrary_STATUS_GSM_AFC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Done

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_AFC_GET_CAPABILITY3_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Getting software capability 3 failed
E_METACalibrationLibrary_STATUS_GSM_AFC_GET_TD_LID_LEN_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Getting TD LID struct length failed
E_METACalibrationLibrary_STATUS_GSM_AFC_COMPOSE_TD_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Composing TD struct failed
E_METACalibrationLibrary_STATUS_GSM_AFC_LOW_HIGH_BAND_SWITCH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Low/High band switch failed

2.2.6 GSM APC Status Code

Table 2-7 GSM APC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_AFC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Start
E_METACalibrationLibrary_STATUS_GSM_AFC_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Frequency error measure failed
E_METACalibrationLibrary_STATUS_GSM_AFC_CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Initial dac range check start
E_METACalibrationLibrary_STATUS_GSM_AFC_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Initial dac range check failed
E_METACalibrationLibrary_STATUS_GSM_AFC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AFC Calibration: Done
E_METACalibrationLibrary_STATUS_GSM_AFC_GET_CAPABILITY3_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Getting software capability 3 failed
E_METACalibrationLibrary_STATUS_GSM_AFC_GET_TD_LID_LEN_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Getting TD LID struct length failed
E_METACalibrationLibrary_STATUS_GSM_AFC_COMPOSE_TD_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Composing TD struct failed
E_METACalibrationLibrary_STATUS_GSM_AFC_LOW_HIGH_BAND_SWITCH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AFC Calibration: Low/High band switch failed

2.2.7 EDGE APC Status Code

Table 2-8 EDGE APC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_EDGE_APC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	EDGE APC Calibration: Start
E_METACalibrationLibrary_STATUS_EDGE_APC_CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	EDGE APC Calibration: Tx power check start
E_METACalibrationLibrary_STATUS_EDGE_APC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	EDGE APC Calibration: Done
E_METACalibrationLibrary_STATUS_EDGE_APC_COUPLER_LOSS_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Coupler loss out of range
E_METACalibrationLibrary_STATUS_EDGE_APC_PA_GAIN_PING_PONG_DETECTED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: PA gain ping-pong detected
E_METACalibrationLibrary_STATUS_EDGE_APC_PA_GAIN_CAL_EXCEEDS_MAX_ITERATION	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: PA gain calibration exceeds max iteration
E_METACalibrationLibrary_STATUS_EDGE_APC_PA_SUBBAND_COMPENSATION_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: PA gain subband calibration out of range

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_EDGE_APC_COUPLER_SUBBAND_COMPENSATION_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Coupler loss subband compensation out of range
E_METACalibrationLibrary_STATUS_EDGE_APC_SET_PA_OCT_LEVEL_CONTROL_TO_TARGET_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: PA oct level control to target failed
E_METACalibrationLibrary_STATUS_EDGE_APC_SET_COMPENSATION_TO_TARGET_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Set compensation to target failed
E_METACalibrationLibrary_STATUS_EDGE_APC_DETECTOR_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Power detector measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Send TX signal failed
E_METACalibrationLibrary_STATUS_EDGE_APC_PA_GAIN_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: PA gain check failed
E_METACalibrationLibrary_STATUS_EDGE_APC_COUPLER_LOSS_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Coupler loss check failed
E_METACalibrationLibrary_STATUS_EDGE_APC_PA_SUBBAND_COMPENSATION_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: PA subband compensation check failed
E_METACalibrationLibrary_STATUS_EDGE_APC_COUPLER_LOSS_SUBBAND_COMPENSATION_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Coupler loss subband compenstaion check failed
E_METACalibrationLibrary_STATUS_EDGE_APC_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Initial TX power measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Fetch TX power measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_PA_CAL_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Initial TX power measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_PA_CAL_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: PA calibration, send TX signal failed
E_METACalibrationLibrary_STATUS_EDGE_APC_PA_CAL_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: PA calibration, fetch TX power measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_COUPLER_LOSS_CAL_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Coupler loss calibration, initial TX power measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_COUPLER_LOSS_CAL_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Coupler loss calibration, send TX signal failed
E_METACalibrationLibrary_STATUS_EDGE_APC_COUPLER_LOSS_CAL_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Coupler loss calibration, fetch TX power measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_SUBBAND_CAL_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: TX Subband calibration, initial TX power measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_SUBBAND_CAL_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: TX Subband calibration, send TX signal failed
E_METACalibrationLibrary_STATUS_EDGE_APC_SUBBAND_CAL_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: TX Subband calibration, fetch TX power measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_COUPLER_LOSS_CAL_DETECTOR_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Coupler loss calibration, power detector measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_SUBBAND_CAL_DETECTOR_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: TX Subband calibration, power detector measurement failed
E_METACalibrationLibrary_STATUS_EDGE_APC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Write initial data failed
E_METACalibrationLibrary_STATUS_EDGE_APC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Write calibration data failed
E_METACalibrationLibrary_STATUS_EDGE_APC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Parse config data failed
E_METACalibrationLibrary_STATUS_EDGE_APC_TESTER_PRESETTING_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Tester presetting failed
E_METACalibrationLibrary_STATUS_EDGE_APC_TESTER_CHANGE_BAND_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Tester change band failed

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_EDGE_APC_TESTER_CHANGE_EXPECTED_POWER_IN_PA_GAIN_CAL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Tester change expected power in PA gain calibration failed
E_METACalibrationLibrary_STATUS_EDGE_APC_TESTER_CHANGE_EXPECTED_POWER_IN_SUBBAND_CAL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Tester change expected power in TX subband calibration failed
E_METACalibrationLibrary_STATUS_EDGE_APC_TESTER_CHANGE_EXPECTED_POWER_IN_COUPLER_LOSS_CAL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Tester change expected power in coupler loss calibration failed
E_METACalibrationLibrary_STATUS_EDGE_APC_LOW_HIGH_BAND_SWITCH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Switch low or high band failed
E_METACalibrationLibrary_STATUS_EDGE_APC_POWER_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: TX power check failed
E_METACalibrationLibrary_STATUS_EDGE_APC_POWER_CAL_EXCEEDS_MAX_ITERATION	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: TX power calibration exceeds max iteration
E_METACalibrationLibrary_STATUS_EDGE_APC_SET_PCL_DAC_TO_TARGET_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	EDGE APC Calibration: Set PCL DAC to target failed

2.2.8 GSM AD6546(Feedback DAC and Slope Skew) Status Code

Table 2-9 GSM AD6546 status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AD6546 Calibration: Start
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM AD6546 Calibration: Done
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_SET_SPEC_COEF_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AD6546 Calibration: Set RF special coef failed
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_SET_APC_RAMP_LEVEL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AD6546 Calibration: Set RF Ramp table apc level failed
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AD6546 Calibration: Write calibration file failed
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_GET_SEP_CH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AD6546 Calibration: Get separate channel failed
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_REQ_RF_START_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AD6546 Calibration: Require RF start to send failed
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_SLOPE_SKEW_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AD6546 Calibration: Slope Skew calibration result check failed
E_METACalibrationLibrary_STATUS_GSM_APC_AD6546_MULTI_SLOT_SUPPORT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM AD6546 Calibration: Multi-Slots is not supported

2.2.9 GSM TRX offset Status Code

Table 2-10 GSM TRX offset status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_AFC_TRXO_FFSET_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM TRX Offset Calibration: Start
E_METACalibrationLibrary_STATUS_GSM_AFC_TRXO_FFSET_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM TRX Offset Calibration: Done
E_METACalibrationLibrary_STATUS_GSM_AFC_TRXO_FFSET_SLOPE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TRX Offset Calibration: AFC Slope calibration failed
E_METACalibrationLibrary_STATUS_GSM_AFC_TRXO_FFSET_NB_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TRX Offset Calibration: Require target to send normal burst failed

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_AFC_TRXO FFSET_COUNT_LIMIT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TRX Offset Calibration: Calibration times exceeds limit
E_METACalibrationLibrary_STATUS_GSM_AFC_TRXO FFSET_TXOFFSET_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TRX Offset Calibration: AFC TRX Offset calibration failed

2.2.10 GSM TX IQ Status Code

Table 2-11 GSM TX IQ status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_TXIQ_STA RT	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM TX IQ Calibration: Start
E_METACalibrationLibrary_STATUS_GSM_TXIQ_DON E	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM TX IQ Calibration: Done
E_METACalibrationLibrary_STATUS_GSM_TXIQ_GET RFID_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TX IQ Calibration: Getting RF ID failed
E_METACalibrationLibrary_STATUS_GSM_TXIQ_GET BBTX_CFG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TX IQ Calibration: Getting BB TX cfg failed
E_METACalibrationLibrary_STATUS_GSM_TXIQ_CAL FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TX IQ Calibration: TX IQ calibration failed
E_METACalibrationLibrary_STATUS_GSM_TXIQ_MUL TI_SLOT_SUPPORT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TX IQ Calibration: Multi-Slots is not supported
E_METACalibrationLibrary_STATUS_GSM_TXIQ_GET MSCAPABILITY_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TX IQ Calibration: Get MS capability failed"

2.3 WCDMA Calibration Status Code

2.3.1 WCDMA TADC Status Code

Table 2-12 WCDMA TADC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA temperature sensor calibration: calibration start
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA temperature sensor calibration: measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ GET_NVRAM_RECORD_LENGTH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA temperature sensor calibration: gets the NVRAM record length failed
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ COMPOSE_NVRAM_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA temperature sensor calibration: composes the NVRAM buffer failed
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ WRITE_NVRAM_INITIAL_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA temperature sensor calibration: writes the NVRAM initial value failed
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ WRITE_NVRAM_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA temperature sensor calibration: writes the NVRAM calibration value failed
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ INVALID_CURRENT_TEMPERATURE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA temperature sensor calibration: current temperature's setting is invalid
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA temperature sensor calibration: check start
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA temperature sensor calibration: check failed
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA temperature sensor calibration: calibration done
E_METACalibrationLibrary_STATUS_WCDMA_TADC_ PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA temperature sensor calibration: parse TADC config file failed

2.3.2 WCDMA AFC Status Code

Table 2-13 WCDMA AFC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_GSM_TXIQ_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM TX IQ Calibration: Start
E_METACalibrationLibrary_STATUS_GSM_TXIQ_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	GSM TX IQ Calibration: Done
E_METACalibrationLibrary_STATUS_GSM_TXIQ_GET_RFID_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TX IQ Calibration: Getting RF ID failed
E_METACalibrationLibrary_STATUS_GSM_TXIQ_GET_BBTX_CFG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TX IQ Calibration: Getting BB TX cfg failed
E_METACalibrationLibrary_STATUS_GSM_TXIQ_GET_MSCAPABILITY_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	GSM TX IQ Calibration: Get MS capability failed"

2.3.3 WCDMA AGC Status Code

Table 2-14 WCDMA AGC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_AGC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA AGC calibration: start
E_METACalibrationLibrary_STATUS_WCDMA_AGC_CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA AGC calibration: check start
E_METACalibrationLibrary_STATUS_WCDMA_AGC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA AGC calibration: done
E_METACalibrationLibrary_STATUS_WCDMA_AGC_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: RSSI measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_CHECK_HIGH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: LNA high mode check failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_CHECK_MID_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: LNA middle mode check failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_CHECK_LOW_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: LNA low mode check failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_CHECK_LNA_MODE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: LNA mode consistency check failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: write AGC initial value to NVRAM failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: write AGC calibration data to NVRAM/cal file failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: parse AGC config file failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_TESTER_PRESETTING_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: tester pre-setting failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_TESTER_CHANGE_BAND_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: tester change band failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_TESTER_CHANGE_CHANNEL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: tester change channel failed
E_METACalibrationLibrary_STATUS_WCDMA_AGC_TESTER_CHANGE_POWER_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: tester change power failed

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_AGC_GAIN_OFFSET_OVERFLOW	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA AGC calibration: AGC LNA gain offset overflow

2.3.4 WCDMA APC Status Code

Table 2-15 WCDMA APC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_APC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA APC calibration: start
E_METACalibrationLibrary_STATUS_WCDMA_APC_CHECK_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA APC calibration: check start
E_METACalibrationLibrary_STATUS_WCDMA_APC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA APC calibration: done
E_METACalibrationLibrary_STATUS_WCDMA_APC_COUPLER_LOSS_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: coupler loss out-of-range
E_METACalibrationLibrary_STATUS_WCDMA_APC_PA_GAIN_PING_PONG_DETECTED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: PA-gain calibration ping-pong detected
E_METACalibrationLibrary_STATUS_WCDMA_APC_PA_GAIN_CAL_EXCEEDS_MAX_ITERATION	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: PA-gain calibration unable to converge
E_METACalibrationLibrary_STATUS_WCDMA_APC_PA_SUBBAND_COMPENSATION_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: PA sub-band compensation out-of-range
E_METACalibrationLibrary_STATUS_WCDMA_APC_COUPLER_SUBBAND_COMPENSATION_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: coupler sub-band compensation out-of-range
E_METACalibrationLibrary_STATUS_WCDMA_APC_SET_PA_OCT_LEVEL_CONTROL_TO_TARGET_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: set PA 8-level control to target failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_SET_COMPENSATION_TO_TARGET_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: set compensation to target failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_POWER_DETECTOR_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: power detector measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: DUT TX control failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_PA_GAIN_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: PA gain check failed (out-of-range)
E_METACalibrationLibrary_STATUS_WCDMA_APC_COUPLER_LOSS_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: coupler loss check failed (out-of-range)
E_METACalibrationLibrary_STATUS_WCDMA_APC_PA_SUBBAND_COMPENSATION_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: PA sub-band compensation check failed (out-of-range)
E_METACalibrationLibrary_STATUS_WCDMA_APC_COUPLER_LOSS_SUBBAND_COMPENSATION_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: coupler loss sub-band compensation check failed (out-of-range)
E_METACalibrationLibrary_STATUS_WCDMA_APC_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: init TX measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: fetch TX measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_PA_CAL_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: PA gain calibration init TX measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_PA_CAL_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: PA gain calibration DUT TX control failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_PA_CAL_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: PA gain calibration fetch TX measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_COUPLER_LOSS_CAL_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: coupler loss calibration init TX measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_COUPLER_LOSS_CAL_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: coupler loss calibration DUT TX control failed

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_APC_COUPLER_LOSS_CAL_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: coupler loss calibration fetch TX measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_SUBBAND_CAL_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: sub-band calibration init TX measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_SUBBAND_CAL_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: sub-band calibration DUT TX control failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_SUBBAND_CAL_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: sub-band calibration fetch TX measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_COUPLER_LOSS_CAL_DETECTOR_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: coupler loss calibration power detector measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_SUBBAND_CAL_DETECTOR_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: sub-band calibration power detector measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: write APC initial value to NVRAM failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: write APC calibration data to NVRAM/cal file failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: parse APC config file failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_TESTER_PRESETTING_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: tester pre-setting failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_TESTER_CHANGE_BAND_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: tester change band failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_TESTER_CHANGE_EXPECTED_POWER_IN_PA_GAIN_CAL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: tester change expected power for PA gain calibration failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_TESTER_CHANGE_EXPECTED_POWER_IN_SUBBAND_CAL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: tester change expected power for sub-band calibration failed
E_METACalibrationLibrary_STATUS_WCDMA_APC_TESTER_CHANGE_EXPECTED_POWER_IN_COUPLER_LOSS_CAL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA APC calibration: tester change expected power for coupler loss calibration failed

2.3.5 WCDMA FHC Status Code

Table 2-16 WCDMA FHC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_FHC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA FHC: start
E_METACalibrationLibrary_STATUS_WCDMA_FHC_START_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: FHC start command failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_QUERY_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: query FHC status command failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: check calibration value failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA FHC: done
E_METACalibrationLibrary_STATUS_WCDMA_FHC_AGC_CHECK_HIGH_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: LNA high mode check failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_AGC_CHECK_MID_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: LNA middle mode check failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_AGC_CHECK_LOW_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: LNA low mode check failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_AGC_LNA_MODE_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: LNA mode consistency check failed

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A PC_PA_GAIN_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: PA gain check failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A PC_COUPLER_LOSS_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: coupler loss check failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A PC_PA_SUBBAND_COMPENSATION_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: PA sub-band compensation check failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A PC_COUPLER_LOSS_SUBBAND_COMPENSATION_CH ECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: coupler loss sub-band compensation check failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A GC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: write AGC initial value to NVRAM failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A PC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: write APC initial value to NVRAM failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A GC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: write AGC calibration data to NVRAM/cal file failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A PC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: write APC calibration data to NVRAM/cal file failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A GC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: parse AGC config file failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_A PC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: parse APC config file failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_P ARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: parse FHC config file failed
E_METACalibrationLibrary_STATUS_WCDMA_FHC_T ESTER_PRESETTING_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA FHC: tester pre-setting failed

2.3.6 WCDMA NSFT Status Code

Table 2-17 WCDMA NSFT status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA NSFT: start
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ TARGET_START_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: DUT NSFT start command failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ TARGET_STOP_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: DUT NSFT stop command failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ TX_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: TX performance measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ BER_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: BER measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: performance check failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA NSFT: done
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ CONFIG_DUT_TPC_ALGO_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: configure DUT TPC algorithm failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ TX_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: TX test failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ BER_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: BER test failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ DUT_OPEN_LOOP_POWER_CONTROL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: start DUT open loop power test failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ DUT_OPEN_LOOP_POWER_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: open loop power test failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_ PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: parse config failed, please check the config file setting

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_TESTER_PRESETTING_IN_SYNC_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: tester presetting of sync test (TX/BER) failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_TESTER_PRESETTING_IN_PRACH_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: tester presetting of open loop power test failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_TESTER_TESTCASE_SETTING_IN_SYNC_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: tester test case setting of sync test (TX/BER) failed
E_METACalibrationLibrary_STATUS_WCDMA_NSFT_TESTER_TESTCASE_SETTING_IN_PRACH_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA NSFT: tester test case setting of open loop power test failed

2.3.7 WCDMA HSDPA NSFT Status Code

Table 2-18 WCDMA HSDPA NSFT status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_HSDPA_NSFT_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA HSPA NSFT: start
E_METACalibrationLibrary_STATUS_WCDMA_HSDPA_NSFT_TARGET_START_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA HSPA NSFT: DUT NSFT start command failed
E_METACalibrationLibrary_STATUS_WCDMA_HSDPA_NSFT_TARGET_STOP_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA HSPA NSFT: DUT NSFT stop command failed
E_METACalibrationLibrary_STATUS_WCDMA_HSDPA_NSFT_TX_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA HSPA NSFT: TX performance measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_HSDPA_NSFT_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA HSPA NSFT: performance check failed
E_METACalibrationLibrary_STATUS_WCDMA_HSDPA_NSFT_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA HSPA NSFT: done

2.3.8 WCDMA HSUPA NSFT Status Code

Table 2-19 WCDMA HSUPA NSFT status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_WCDMA_HSUPA_NSFT_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA HSUPA NSFT: start
E_METACalibrationLibrary_STATUS_WCDMA_HSUPA_NSFT_TARGET_START_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA HSUPA NSFT: DUT NSFT start command failed
E_METACalibrationLibrary_STATUS_WCDMA_HSUPA_NSFT_TARGET_STOP_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA HSUPA NSFT: DUT NSFT stop command failed
E_METACalibrationLibrary_STATUS_WCDMA_HSUPA_NSFT_TX_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA HSUPA NSFT: TX performance measurement failed
E_METACalibrationLibrary_STATUS_WCDMA_HSUPA_NSFT_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	WCDMA HSUPA NSFT: performance check failed
E_METACalibrationLibrary_STATUS_WCDMA_HSUPA_NSFT_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	WCDMA HSUPA NSFT: done

2.4 LTE Calibration Status Code

2.4.1 LTE Common Status Code

Table 2-20 LTE common status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_PARSE_COMMON_CFG_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	General: LTE parse common configures failed
E_METACalibrationLibrary_STATUS_LTE_GET_CAPABILITY_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	General: LTE get capability failed
E_METACalibrationLibrary_STATUS_LTE_GET_AFC_DAC_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	General: LTE get AFC DAC from target failed
E_METACalibrationLibrary_STATUS_LTE_SET_TOOL_USAGE_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	General: LTE set tool usage to target failed

2.4.2 LTE TADC Status Code

Table 2-21 LTE TADC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_TADC_START	E_METACalibrationLibrary_STATUS_CODE_TYPE_INFO	LTE temperature sensor calibration: calibration start
E_METACalibrationLibrary_STATUS_LTE_TADC_MEASUREMENT_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: measurement failed
E_METACalibrationLibrary_STATUS_LTE_TADC_GET_NVRAM_RECORD_LENGTH_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: gets the NVRAM record length failed
E_METACalibrationLibrary_STATUS_LTE_TADC_COMPOSE_NVRAM_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: composes the NVRAM buffer failed
E_METACalibrationLibrary_STATUS_LTE_TADC_WRITE_NVRAM_INITIAL_VALUE_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: writes the NVRAM initial value failed
E_METACalibrationLibrary_STATUS_LTE_TADC_WRITE_NVRAM_CAL_DATA_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: writes the NVRAM calibration value failed
E_METACalibrationLibrary_STATUS_LTE_TADC_INVALID_CURRENT_TEMPERATURE	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: current temperature's setting is invalid
E_METACalibrationLibrary_STATUS_LTE_TADC_CHECK_START	E_METACalibrationLibrary_STATUS_CODE_TYPE_INFO	LTE temperature sensor calibration: check start
E_METACalibrationLibrary_STATUS_LTE_TADC_CHECK_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: check failed
E_METACalibrationLibrary_STATUS_LTE_TADC_DONE	E_METACalibrationLibrary_STATUS_CODE_TYPE_INFO	LTE temperature sensor calibration: calibration done
E_METACalibrationLibrary_STATUS_LTE_TADC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: CFG section does not exist
E_METACalibrationLibrary_STATUS_LTE_TADC_PARSE_INI_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE temperature sensor calibration: INI section does not exist

2.4.3 LTE AFC Status Code

Table 2-22 LTE AFC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_AFC_START	E_METACalibrationLibrary_STATUS_CODE_TYPE_INFO	LTE AFC Calibration: Start

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_AFC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	LTE AFC Calibration: Done
E_METACalibrationLibrary_STATUS_LTE_AFC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	LTE AFC Calibration: Prase configure file failed
E_METACalibrationLibrary_STATUS_LTE_AFC_PARSE_INIT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	LTE AFC Calibration: Prase initial file failed
E_METACalibrationLibrary_STATUS_LTE_AFC_TESTER_PRESETTING_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	LTE AFC Calibration: Tester presetting failed
E_METACalibrationLibrary_STATUS_LTE_AFC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	LTE AFC Calibration: Write initial data failed
E_METACalibrationLibrary_STATUS_LTE_AFC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	LTE AFC Calibration: Write calibration data failed
E_METACalibrationLibrary_STATUS_LTE_AFC_WRITE_NVRAM_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	LTE AFC Calibration: Write flash failed
E_METACalibrationLibrary_STATUS_LTE_AFC_VALUE_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	LTE AFC Calibration: Calibration result check failed
E_METACalibrationLibrary_STATUS_LTE_AFC_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	LTE AFC Calibration: DUT TX failed

2.4.4 LTE AGC Status Code

Table 2-23 LTE AGC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_AGC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	"LTE AGC Calibration: Start"
E_METACalibrationLibrary_STATUS_LTE_AGC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	"LTE AGC Calibration: Done"
E_METACalibrationLibrary_STATUS_LTE_AGC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Prase configure file failed"
E_METACalibrationLibrary_STATUS_LTE_AGC_TESTER_PRESETTING_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Tester presetting failed"
E_METACalibrationLibrary_STATUS_LTE_AGC_TESTER_CHANGE_BAND_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Tester change band failed"
E_METACalibrationLibrary_STATUS_LTE_AGC_GAIN_OFFSET_OVERFLOW	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Path loss overflow (Main path)"
E_METACalibrationLibrary_STATUS_LTE_AGC_RXD_GAIN_OFFSET_OVERFLOW	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Path loss overflow (Diversity path)"
E_METACalibrationLibrary_STATUS_LTE_AGC_TESTER_CHANGE_POWER_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Tester change downlink power failed"
E_METACalibrationLibrary_STATUS_LTE_AGC_TESTER_CHANGE_DL_FREQ_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Tester change downlink frequency failed"
E_METACalibrationLibrary_STATUS_LTE_AGC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Write initial data failed"
E_METACalibrationLibrary_STATUS_LTE_AGC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Write calibration data failed"

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_AGC_VALUE_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Calibration result check failed"
E_METACalibrationLibrary_STATUS_LTE_AGC_MEASURE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: RSSI measure failed"
E_METACalibrationLibrary_STATUS_LTE_AGC_LNA_MODE_MISMATCH	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: LNA mode mismatch"
E_METACalibrationLibrary_STATUS_LTE_AGC_GET_TEMP_INFO_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE AGC Calibration: Get temperature info failed"

2.4.5 LTE APC Status Code

Table 2-24 LTE APC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_APC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	"LTE APC Calibration: Start"
E_METACalibrationLibrary_STATUS_LTE_APC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	"LTE APC Calibration: Done"
E_METACalibrationLibrary_STATUS_LTE_APC_PA_SUBBAND_COMPENSATION_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_PA_SUBBAND_COMPENSATION_OUT_OF_RANGE"
E_METACalibrationLibrary_STATUS_LTE_APC_COUPLER_SUBBAND_COMPENSATION_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_COUPLER_SUBBAND_COMPENSATION_OUT_OF_RANGE"
E_METACalibrationLibrary_STATUS_LTE_APC_SUBBAND_CAL_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_SUBBAND_CAL_INIT_TX_MEASUREMENT_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_SUBBAND_CAL_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_SUBBAND_CAL_DUT_TX_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_SUBBAND_CAL_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_SUBBAND_CAL_FETCH_TX_MEASUREMENT_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_SET_PA_OCT_LEVEL_CONTROL_TO_TARGET_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_SET_PA_OCT_LEVEL_CONTROL_TO_TARGET_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_COUPLER_LOSS_OUT_OF_RANGE	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_COUPLER_LOSS_OUT_OF_RANGE"
E_METACalibrationLibrary_STATUS_LTE_APC_PA_GAIN_PING_PONG_DETECTED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_PA_GAIN_PING_PONG_DETECTED"
E_METACalibrationLibrary_STATUS_LTE_APC_PA_GAIN_CAL_EXCEEDS_MAX_ITERATION	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_PA_GAIN_CAL_EXCEEDS_MAX_ITERATION"
E_METACalibrationLibrary_STATUS_LTE_APC_PA_CAL_INIT_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_PA_CAL_INIT_TX_MEASUREMENT_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_PA_CAL_FETCH_TX_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_PA_CAL_FETCH_TX_MEASUREMENT_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_COUPLER_LOSS_CAL_DETECTOR_MEASUREMENT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_COUPLER_LOSS_CAL_DETECTOR_MEASUREMENT_FAILED"

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_APC_DUT_TX_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_DUT_TX_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_WRITE_CAL_DATA_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_WRITE_INIT_VALUE_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_PA_GAIN_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_PA_GAIN_CHECK_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_COUPLER_LOSS_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_COUPLER_LOSS_CHECK_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_PA_SUBBAND_COMPENSATION_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_PA_SUBBAND_COMPENSATION_CHECK_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_COUPLER_LOSS_SUBBAND_COMPENSATION_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_COUPLER_LOSS_SUBBAND_COMPENSATION_CHECK_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_TESTER_PRESETTING_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_TESTER_PRESETTING_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_TESTER_CHANGE_BAND_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_TESTER_CHANGE_BAND_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_TESTER_CHANGE_EXPECTED_POWER_IN_PA_GAIN_CAL_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_TESTER_CHANGE_EXPECTED_POWER_IN_PA_GAIN_CAL_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_TESTER_CHANGE_UL_FREQUENCY_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_TESTER_CHANGE_UL_FREQUENCY_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: LTE_APC_PARSE_CONFIG_FAILED"
E_METACalibrationLibrary_STATUS_LTE_APC_GET_TPC_TABLE_INDEX_ERROR	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: Invalid LTE APC TPC table index"
E_METACalibrationLibrary_STATUS_LTE_APC_GET_TEMPERATURE_INFO_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: Get temperature info failed"
E_METACalibrationLibrary_STATUS_LTE_APC_ET_CALIBRATION_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: ET calibration failed"
E_METACalibrationLibrary_STATUS_LTE_APC_GET_ET_CALIBRATION_RESULT_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE APC Calibration: Get ET calibration result failed"

2.4.6 LTE FHC Status Code

Table 2-25 LTE FHC status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_FHC_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	"LTE FHC Calibration: Start"
E_METACalibrationLibrary_STATUS_LTE_FHC_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	"LTE FHC Calibration: Done"

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_FHC_AGC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_AGC_PARSE_CONFIG_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_APC_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_APC_PARSE_CONFIG_FAILED"
E_METACalibrationLibrary_STATUS_LTE_TESTER_SETTING_PRESETTING_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_TESTER_SETTING_PRESETTING_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_AGC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_AGC_WRITE_INIT_VALUE_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_APC_WRITE_INIT_VALUE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_APC_WRITE_INIT_VALUE_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_AGC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_AGC_WRITE_CAL_DATA_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_APC_WRITE_CAL_DATA_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_APC_WRITE_CAL_DATA_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_AGC_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_AGC_CHECK_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_APC_CHECK_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_APC_CHECK_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_START_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_START_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_AGC_EVALUATE_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_AGC_EVALUATE_FAILED"
E_METACalibrationLibrary_STATUS_LTE_FHC_AGC_LNA_MODE_MISMATCH	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE FHC Calibration: LTE_FHC_AGC_LNA_MODE_MISMATCH"

2.4.7 LTE NSFT Status Code

Table 2-26 LTE NSFT status code

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_NSFT_START	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	"LTE NSFT: start"
E_METACalibrationLibrary_STATUS_LTE_NSFT_DONE	E_METACalibrationLibrary_STAT US_CODE_TYPE_INFO	"LTE NSFT: Done"
E_METACalibrationLibrary_STATUS_LTE_NSFT_PARSE_CONFIG_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE NSFT: parse configures failed"
E_METACalibrationLibrary_STATUS_LTE_NSFT_TESTER_PRESETTING_IN_SYNC_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE NSFT: Instrument presetting failed"
E_METACalibrationLibrary_STATUS_LTE_NSFT_TARGET_RX_START_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE NSFT: RX start failed"
E_METACalibrationLibrary_STATUS_LTE_NSFT_TESTER_TESTCASE_SETTING_IN_RX_SYNC_TEST_FAILED	E_METACalibrationLibrary_STAT US_CODE_TYPE_ERROR	"LTE NSFT: Instrument RX test setting failed"

**2 Error Code Listing**

Status Code	Status Type	Status String
E_METACalibrationLibrary_STATUS_LTE_NSFT_RX_TEST_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	"LTE NSFT: RX test failed"
E_METACalibrationLibrary_STATUS_LTE_NSFT_CHECK_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	"LTE NSFT: Check failed"
E_METACalibrationLibrary_STATUS_LTE_NSFT_TARGET_TX_START_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	"LTE NSFT: TX start failed"
E_METACalibrationLibrary_STATUS_LTE_AFC_DUT_TX_FAILED	E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	LTE AFC Calibration: DUT TX failed

MediaTek Confidential

© 2017 MediaTek Inc.

Classification: Confidential B

3 Exported Functions

This chapter mentions the functions exported by META Calibration Library, and their prototypes.

3.1 The Terminology of Function Descriptions

3.1.1 The Meaning of Parameter Table:

Parameter:

Table 3-1 The meaning of parameter table sample

Parameter	Direction	Description

Parameter:

The name of parameter.

Direction:

IN: It means this parameter is used for input value.

OUT: It means this parameter is used for output value. You have to pass the address pointer of container.

Description:

The description of that parameter.

3.2 Exported Calibration Functions

3.2.1 METACalibrationLibrary_Init

Definition:

```
E_METACalibrationLibrary_RESULT __stdcall METACalibrationLibrary_Init(const int meta_handle);
```

Description:

META Calibration Library initialization function, used to initialize the library context variable.

CallBack:

NA

Parameter:

Table 3-2 METACalibrationLibrary init parameter

Parameter	Direction	Description
meta_handle	IN	The meta_handle used to control the DUT.

Return Value:

Table 3-3 METACalibrationLibrary init return value

Parameter	Description
E_METACalibrationLibrary_RESULT_SUCCESS	Success
E_METACalibrationLibrary_RESULT_FAILED	Library initialization failed.

3.2.2 METACalibrationLibrary_RegisterCallBack

Definition:

```
E_METACalibrationLibrary_RESULT __stdcall METACalibrationLibrary_RegisterCallBack(const int meta_handle,
const METACalibrationLibrary_Log_Display_CallBack cb);
```

Description:

The function could register a callback function to retrieve calibration log.

CallBack:

```
typedef void (__stdcall *METACalibrationLibrary_Log_Display_CallBack)(const char *logBuf);
```

Parameter:

Table 3-4 METACalibrationLibrary registerCallBack parameter

Parameter	Direction	Description
meta_handle	IN	The meta_handle used to control the DUT.

Return Value:

Table 3-5 METACalibrationLibrary registerCallBack return value

Parameter	Description
E_METACalibrationLibrary_RESULT_SUCCESS	Success

Sample code:

```
static void __stdcall SharelibraryCB(const char *msg)
{
    printf("%s\n", msg);
}

void main()
{
    META_RESULT mr;
    // init meta and get handler
    mr = META_GetAvailableHandle(&META_HANDLE);
    mr = META_Init_r(META_HANDLE, ErrorHandler);
    // register callback function
    METACalibrationLibrary_RegisterCallBack(META_HANDLE, SharelibraryCB);
    // start calibration library
    METACalibrationLibrary_StartWithRetrunedStatusCode(META_HANDLE,...);
    // calibration library finish
    METACalibrationLibrary_DeInit(META_HANDLE);
}
```

3.2.3 METACalibrationLibrary_DeInit

Definition:

E_METACalibrationLibrary_RESULT __stdcall METACalibrationLibrary_DeInit(const int meta_handle);

Description:

META Calibration Library de-init function, used to clean-up the library context variable.

CallBack:

NA

**Parameter:****Table 3-6 METACalibrationLibrary delnit parameter**

Parameter	Direction	Description
meta_handle	IN	The meta_handle used to control the DUT.

Return Value:**Table 3-7 METACalibrationLibrary delnit return value**

Parameter	Description
E_METACalibrationLibrary_RESULT_SUCCESS	Success
E_METACalibrationLibrary_RESULT_FAILED	Library de-init failed.

3.2.4 METACalibrationLibrary_Start

Definition:

E_METACalibrationLibrary_RESULT __stdcall METACalibrationLibrary_Start(const int meta_handle,

const METACalibrationLibrary_COMMON_CFG_T *cfg,

const METACalibrationLibrary_STATUS_CallBack cb,

int *pStopFlag);

Description:

This function is an all-in-one function to start GGE (GSM/GPRS/EDGE), WCDMA, TDSCDMA calibration and the status is notified asynchronously by the callback function.

Related data structure:

typedef struct

{

/// calibration status code to indicate the status

E_METACalibrationLibrary_STATUS_CODE m_eStatuCode;

/// calibration status type to indicate the status code is error or information

E_METACalibrationLibrary_STATUS_CODE_TYPE m_eStatusType;

```

/// status message

char m_cErrorMsgbuf[256];

}S_METACalibrationLibrary_STATUS_T;

typedef struct
{
    /// GGE Temperature sensor calibration
    bool b_gge_tadc_cal;

    /// GGE CAP ID calibration
    bool b_gge_cap_id_cal;

    /// GGE FB DAC calibration
    bool b_gge_fb_dac_cal;

    /// GGE Slope Skew calibration
    bool b_gge_slope_skew_cal;

    /// GGE AFC calibration
    bool b_gge_afc_cal;

    /// GGE AFC TRX Offset calibration
    bool b_gge_afc_trx_offset_cal;

    /// GGE AGC calibration
    bool b_gge_agc_cal;

    /// GGE APC calibration
    bool b_gge_apc_cal;

    /// GGE EDGE APC calibration
    bool b_gge_edge_apc_cal;

    /// GGE txiq calibration
    bool b_gge_txiq_cal;

    /// GGE FHC calibration attribute (to enable FHC or not)
    bool b_gge_fhc_cal;

    /// GGE TX sub-band calibration attribute (to enable sub-band calibration or not)
    bool b_gge_tpc_subband_cal;

```

```
/// GGE APC W coefficient calibration
```

```
bool b_gge_agc_w_cal;
```

```
/// GGE AD6546 APC calibration
```

```
bool b_gge_AD6546_apc_cal;
```

```
/// GGE GSM APC Power Check
```

```
bool b_gge_apc_check;
```

```
/// GGE EDGE APC Power Check
```

```
bool b_gge_edge_apc_check;
```

```
}S_METACalibrationLibrary_GGE_CAL_ITEM_T;
```

```
typedef struct
```

```
{
```

```
/// GGE NSFT R99 TX performance test
```

```
bool b_gge_nsft_gmsk;
```

```
/// GGE NSFT BER test
```

```
bool b_gge_nsft_ber;
```

```
/// GGE NSFT PRACH test (open loop power)
```

```
bool b_gge_nsft_epsk;
```

```
}S_METACalibrationLibrary_GGE_NSFT_ITEM_T;
```

```
typedef struct
```

```
{
```

```
/// WCDMA Temperature sensor calibration
```

```
bool b_wcdma_tadc_cal;
```

```
/// WCDMA DCXO AFC calibration
```

```
bool b_wcdma_dcxo_afc_cal;
```

```
/// WCDMA AFC calibration
```

```
bool b_wcdma_afc_cal;
```

```
/// WCDMA AGC calibration
```

```
bool b_wcdma_agc_cal;
```

```

/// WCDMA APC calibration

bool b_wcdma_apc_cal;

/// WCDMA FHC calibration attribute (to enable FHC or not)

bool b_wcdma_fhc_cal;

/// WCDMA TX sub-band calibration attribute (to enable sub-band calibration or not)

bool b_wcdma_tpc_subband_cal;

}S_METACalibrationLibrary_WCDMA_CAL_ITEM_T;

```

```

typedef struct
{
    /// WCDMA NSFT R99 TX performance test

    bool b_wcdma_nsft_tpc;

    /// WCDMA NSFT BER test

    bool b_wcdma_nsft_ber;

    /// WCDMA NSFT PRACH test (open loop power)

    bool b_wcdma_nsft_prach;

    /// WCDMA NSFT HSDPA performance test

    bool b_wcdma_hsdpa_nsft;

    /// WCDMA NSFT HSUPA performance test

    bool b_wcdma_hsupa_nsft;

}S_METACalibrationLibrary_WCDMA_NSFT_ITEM_T;

```

```

typedef struct
{
    /// TDSCDMA temperature sensor calibration

    bool b_tda_tadc_cal;

    /// TDSCDMA CAP ID calibration

    bool b_tda_cap_id_cal;

    /// TDSCDMA AFC calibration

    bool b_tda_afc_cal;

```



```
/// TDSCDMA AGC calibration
```

```
bool b_tda_rx_path_loss_cal;
```

```
/// TDSCDMA APC calibration
```

```
bool b_tda_tpc_cal;
```

```
/// TDSCDMA TX sub-band calibration attribute (to enable sub-band calibration or not)
```

```
bool b_tda_tpc_subband_cal;
```

```
/// TDSCDMA FHC attribute (to enable FHC or not)
```

```
bool b_tda_fhc_cal;
```

```
}S_METACalibrationLibrary_TD_CAL_ITEM_T;
```

```
typedef struct
```

```
{
```

```
/// TDSCDMA NSFT TX performance test
```

```
bool b_tda_nsft_tpc;
```

```
/// TDSCDMA NSFT BER test
```

```
bool b_tda_nsft_ber;
```

```
/// TDSCDMA FT TX performance test
```

```
bool b_tda_ft_tpc;
```

```
/// TDSCDMA FT BER test
```

```
bool b_tda_ft_ber;
```

```
}S_METACalibrationLibrary_TD_NSFT_ITEM_T;
```

```
typedef struct
```

```
{
```

```
/// LTE temperature sensor calibration
```

```
bool b_lte_tadc_cal;
```

```
/// LTE CAP ID calibration
```

```
bool b_lte_cap_id_cal;
```

```
/// LTE AFC calibration
```

```
bool b_lte_afc_cal;
```

```

/// LTE AGC calibration

bool b_lte_rx_path_loss_cal;

/// LTE APC calibration

bool b_lte_tpc_cal;

/// LTE TX sub-band calibration attribute (to enable sub-band calibration or not)

bool b_lte_tpc_subband_cal;

/// LTE FHC attribute (to enable FHC or not)

bool b_lte_fhc_cal;

} S_METACalibrationLibrary_LTE_CAL_ITEM_T;

```

```

typedef struct
{
    /// LTE NSFT TX performance test

    bool b_lte_nsft_tx;

    /// LTE NSFT RX test

    bool b_lte_nsft_rx;

} S_METACalibrationLibrary_LTE_NSFT_ITEM_T;

```

```

typedef struct
{
    /// cfg file path (*.cfg)
    char* cfg_path;

    /// ini file path (*.ini)
    char* ini_path;

    /// log file path (*.log)
    char* log_path;

    /// output file path (*.cal)
    char* output_path;

    /// result file path (report)
    char* cal_result_path;

```

```

/// smart phone nvram database path

char* sp_nvram_database_path;

/// GSM/GPRS/EDGE device type

int i_device_type;

/// WCDMA device type

int i_device_type_wcdma;

/// TD-SCDMA device type

int i_device_type_tdsdma;

/// LTE device type

int i_device_type_lte;

/// reset tester

bool resetTester;

/// RF port of tester

int i_tester_rf_port;

/// GGE calibration items

S_METACalibrationLibrary_GGE_CAL_ITEM_T ggeCalibrationItems;

/// GGE NSFT items (GMSK, BER, EPSK)

S_METACalibrationLibrary_GGE_NSFT_ITEM_T ggeNsftItems;

/// WCDMA calibration items

S_METACalibrationLibrary_WCDMA_CAL_ITEM_T wcdmaCalibrationItems;

/// WCDMA NSFT items (R99: TPC/BER/PRACH, R5, R6)

S_METACalibrationLibrary_WCDMA_NSFT_ITEM_T wcdmaNsftItems;

/// TDSCDMA calibration items (AST)

S_METACalibrationLibrary_TD_CAL_ITEM_T tdsdmaCalibrationItems;

/// TDSCDMA NSFT items (AST)

S_METACalibrationLibrary_TD_NSFT_ITEM_T tdsdmaNsftItems;

/// LTE calibration items

S_METACalibrationLibrary_LTE_CAL_ITEM_T lteCalibrationItems;

/// LTE NSFT items

S_METACalibrationLibrary_LTE_NSFT_ITEM_T lteNsftItems;

```

```
/// Utilities calibration items
```

```
S_METACalibrationLibrary_Utilities_ITEM_T  utilitiesCalibrationItems;
```

```
/// Others calibration times
```

```
S_METACalibrationLibrary_Others_ITEM_T  othersCalibrationItems;
```

```
}METACalibrationLibrary_COMMON_CFG_T;
```

CallBack:

```
typedef void (__stdcall *METACalibrationLibrary_STATUS_CallBack)(const S_METACalibrationLibrary_STATUS_T
status);
```

Parameter:

Table 3-8 METACalibrationLibrary start parameter

Parameter	Direction	Description
meta_handle	IN	The meta_handle used to control the DUT.
cfg	IN	The calibration library configuration parameters.
cb	IN	The callback function that used for status notification.
pStopFlag	IN	The application level stop flag to terminate the calibration flow.

Return Value:

Table 3-9 METACalibrationLibrary start return value

Parameter	Description
E_METACalibrationLibrary_RESULT_SUCCESS	Calibration is successful for all items.
E_METACalibrationLibrary_RESULT_FAILED	Calibration failed.

3.2.5 METACalibrationLibrary_StartWithRetrunedStatusCode

Definition:

```
E_METACalibrationLibrary_RESULT __stdcall METACalibrationLibrary_StartWithRetrunedStatusCode(const int
meta_handle,
```

```
const METACalibrationLibrary_COMMON_CFG_T *cfg,
```

```
S_METACalibrationLibrary_STATUS_T* status,
```

```
int *pStopFlag);
```

Description:

This function is an all-in-one function to start GGE (GSM/GPRS/EDGE), WCDMA, TDSCDMA calibration and the status is notified synchronously at the function returns.

Related data structure:

Refer to 3.2.4

CallBack:

NA

Parameter:

Table 3-10 METACalibrationLibrary startWithRetrunedStatusCode parameter

Parameter	Direction	Description
meta_handle	IN	The meta_handle used to control the DUT.
cfg	IN	The calibration library configuration parameters.
status	IN/OUT	The status code that indicates the error state (if any). If the calibration result is successful, the status remains as the last informational code.
pStopFlag	IN	The application level stop flag to terminate the calibration flow.

Return Value:

Table 3-11 METACalibrationLibrary startWithRetrunedStatusCode return value

Parameter	Description
E_METACalibrationLibrary_RESULT_SUCCESS	Calibration is successful for all items.
E_METACalibrationLibrary_RESULT_FAILED	Calibration failed.

3.2.6 METACalibrationLibrary_GetStatusCodeString

Definition:

```
const char* __stdcall METACalibrationLibrary_GetStatusCodeString(E_METACalibrationLibrary_STATUS_CODE code);
```

Description:

Query the status string by supplying status code.

CallBack:

NA

Parameter:

Table 3-12 METACalibrationLibrary getStatusCodeString parameter

Parameter	Direction	Description
code	IN	The status code to be checked.

Return Value:

Table 3-13 METACalibrationLibrary getStatusCodeString return value

Parameter	Description
const pointer to the status string	The status string corresponding to the status code.

3.2.7 METACalibrationLibrary_GetStatusType

Definition:

E_METACalibrationLibrary_STATUS_CODE_TYPE __stdcall
 METACalibrationLibrary_GetStatusType(E_METACalibrationLibrary_STATUS_CODE code);

Description:

Query the type of a status code to determine whether the status code is information or error.

CallBack:

NA

Parameter:

Table 3-14 METACalibrationLibrary getStatusType parameter

Parameter	Direction	Description
code	IN	The status code to be checked.

Return Value:

Table 3-15 METACalibrationLibrary getStatusType return value

Parameter	Description
E_METACalibrationLibrary_STATUS_CODE_TYPE_INFO	The input status code is information type.
E_METACalibrationLibrary_STATUS_CODE_TYPE_ERROR	The input status code is an error code.

3.2.8 METACalibrationLibrary_GetLastError

Definition:

```
const S_METACalibrationLibrary_STATUS_T* __stdcall METACalibrationLibrary_GetLastError(const int meta_handle);
```

Description:

Get last status code.

CallBack:

NA

Parameter:

Table 3-16 METACalibrationLibrary getLastError parameter

Parameter	Direction	Description
meta_handle	IN	The meta_handle used to control the DUT.

Return Value:

Table 3-17 METACalibrationLibrary getLastError return value

Parameter	Description
const pointer to status variable	The last status code remained in the library context.

4 Calibration Library Configure Settings

4.1 Calibration Configures

4.1.1 Calibration Retry Configure

Table 4-1 Calibration retry configure

Section	Key	Value	Note
[ShareLibrary Settings]	Calibration Retry Count	[0 ~ N]	The retry count of calibration. Default value is 0.

4.2 NSFT Configures

4.2.1 NSFT Retry Configure

Table 4-2 NSFT Retry configure

Section	Key	Value	Note
[ShareLibrary Settings]	NSFT Retry Count	[0 ~ N]	The retry count of NSFT. Default value is 0.

4.2.2 WCDMA NSFT TX Testing Items Switch

Table 4-3 WCDMA NSFT TX testing items switch

Section	Key	Value	Note
WCDMA NSFT Common Settings]	Max Power Test	Y	Turn on MAX Power Test in NSFT TX performance testing. (Default value)
		N	Turn off MAX Power Test in NSFT TX performance testing.
	Min Power Test	Y	Turn on MAX Power Test in NSFT TX performance testing. (Default value)
		N	Turn off MIN Power Test in NSFT TX performance testing.

5 Sample code

5.1 META Calibration Library initialization

```

/*****
 * Init META DLL
 *****/

META_RESULT mr;

int metaHandle;

mr = META_GetAvailableHandle(&metaHandle);

if(mr != META_SUCCESS)
{
    // error handling
}

mr = META_Init_r(metaHandle, ErrorHandler);

if(mr != META_SUCCESS)
{
    // error handling
}

/*****
 * Connect with target
 *****/

// ...

/*****
 * Init META Calibration Library
 *****/

E_METACalibrationLibrary_RESULT m_eCalLibResult = METACalibrationLibrary_Init(metaHandle);

if(m_eCalLibResult != E_METACalibrationLibrary_RESULT_SUCCESS)
{
    // error handling
}

```



```
// ...
```

5.2 Start calibration via META Calibration Library

```
/******  
 * Init META Calibration Library  
*****/  
  
// ...  
  
/******  
 * Set CFG/INI/CAL/Result path  
*****/  
  
METACalibrationLibrary_COMMON_CFG_T calibrationCfg;  
  
// cfg file path (file)  
calibrationCfg.cfg_path = "C:\\\\CAL\\Test.cfg";  
  
// ini file path (file)  
calibrationCfg.ini_path = "C:\\\\CAL\\Test.ini";  
  
// log file path (file)  
calibrationCfg.log_path = "C:\\\\CAL\\Test_cal.log";  
  
// output file path (file)  
calibrationCfg.output_path = "C:\\\\CAL\\Test_output.cal";  
  
// result file path (folder)  
calibrationCfg.cal_result_path = "C:\\\\CAL";  
  
/******  
 * Set Device type  
*****/  
  
calibrationCfg.i_device_type = RCTLIB_DEVICE_AGILENT_8960;  
calibrationCfg.i_device_type_wcdma = RCTLIB_DEVICE_AGILENT_8960;  
calibrationCfg.i_device_type_tdsdma = RCTLIB_DEVICE_AGILENT_8960;  
calibrationCfg.resetTester = false;  
  
/******  
 * Set GGE calibration item  
*****/
```

```
memset(&(calibrationCfg.ggeCalibrationItems), 0, sizeof(S_METACalibrationLibrary_GGE_CAL_ITEM_T));

memset(&(calibrationCfg.ggeNsftItems), 0, sizeof(S_METACalibrationLibrary_GGE_NSFT_ITEM_T));

/*****

* Set WCDMA calibration item

*****/

memset(&(calibrationCfg.wcdmaCalibrationItems), 0, sizeof(S_METACalibrationLibrary_WCDMA_CAL_ITEM_T));

memset(&(calibrationCfg.wcdmaNsftItems), 0, sizeof(S_METACalibrationLibrary_WCDMA_NSFT_ITEM_T));

// enable WCDMA Temperature ADC calibration

calibrationCfg.wcdmaCalibrationItems.b_wcdma_tadc_cal = true;

/*****

* Set TDSCDMA calibration item

*****/

memset(&(calibrationCfg.tdscdmaCalibrationItems), 0, sizeof(S_METACalibrationLibrary_TD_CAL_ITEM_T));

memset(&(calibrationCfg.tdscdmaNsftItems), 0, sizeof(S_METACalibrationLibrary_TD_NSFT_ITEM_T));

S_METACalibrationLibrary_STATUS_T sta;

E_METACalibrationLibrary_RESULT m_rMetaLibResult = METACalibrationLibrary_Start(metaHandle,

    &calibrationCfg,

    &sta,

    m_piMetaLibStop);

if(m_rMetaLibResult != E_METACalibrationLibrary_RESULT_SUCCESS)

{

    // error handling

    /* Get status code */

    sta.m_eStatusCode;

    // ...

    /* Get status string */

    sta.m_cErrorMsgbuf;

    // ...

}
```

5.3 META Calibration Library Deinit

```

/*****
 * Calibration procedure
 *****/

// ...

/*****
 * De-init META Calibration Library
 *****/

E_METACalibrationLibrary_RESULT m_eCalLibResult = METACalibrationLibrary_DeInit(metaHandle);
if(m_eCalLibResult != E_METACalibrationLibrary_RESULT_SUCCESS)
{
    // error handling
}

```

6 Instrument library

The instrument library provides the corresponding instrument control procedure that mapped to specific RF calibration procedure. The instrument is composed of unified instrument control procedure that can easily extended by customer to add 3rd party instrument support.

In this section, we gonna to introduce

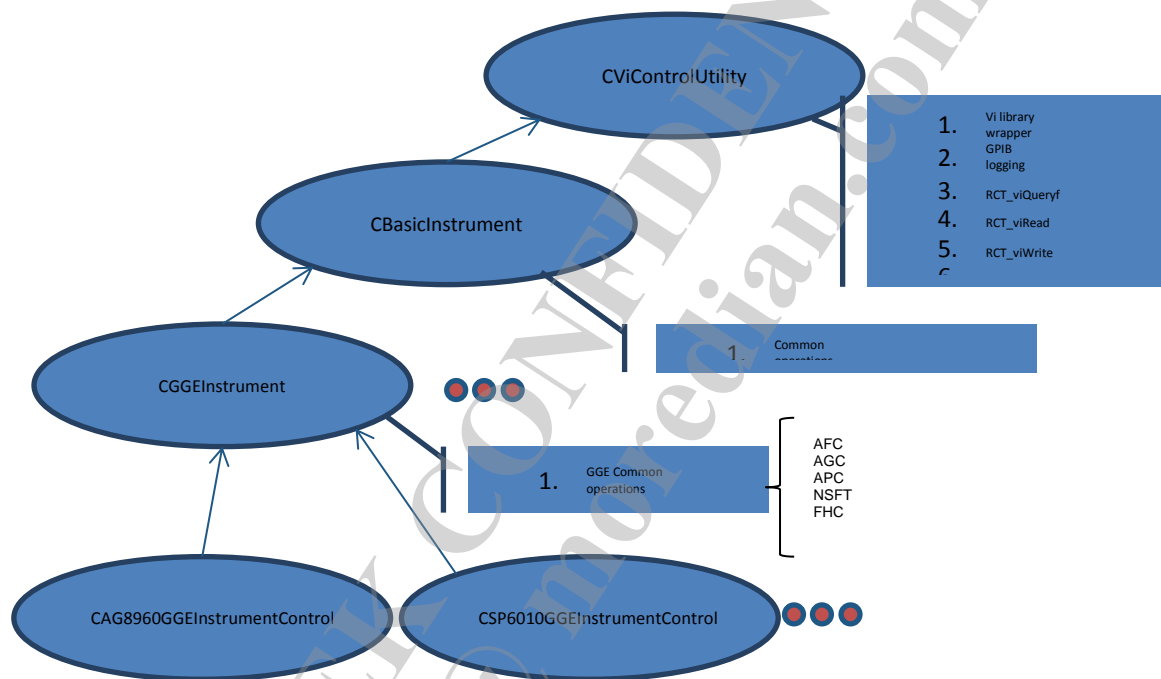


Figure 6-1 Class hirachey of instrument library

6.1 The return values of the Insturment functions

Table 6-1 WCDMA NSFT TX testing items switch

Avoid	Use Instead
RCTLIB_SUCCESS	RCT return status code successful
RCTLIB_ERROR	RCT return status code failed or error
RCTLIB_ABORTED	RCT return status code error aborted
RCTLIB_NOT_YET_IMPLEMENTED	RCT retrun status code not yet implemented



6.2 GPIB Control and Logging functions

The instrument library provides an abstraction layer of VI GPIB control functions (vi_open, vi_write, vi_read..etc) for providing an easier logging method. When the users turn on debugging information, the GPIB logs will also showing in the calibration log. Using these functions to porting instruments help you more easier to debugging.

6.2.1 Exported functions

These functions are all inherited from the class CViControlUtility as show in Figure 6-1.

6.2.1.1 RCT_viQueryf

Definition:

ViStatus RCT_viQueryf(ViSession vi_handle, ViString cmd, ViString read, ...);

Description:

The functionality is same as viVQueryf.

Parameter:

Table 6-2 RCT viQueryf parameter

Parameter	Direction	Parameter
vi_handle	IN	The VI GPIB handler
cmd	IN	GPIB command
read	IN/OUT	A pointer pointing to a space for storing the result of GPIB command.

6.2.1.2 RCT_viRead

Definition:

ViStatus RCT_viRead(ViSession vi, ViPBuf buf, ViUInt32 count, ViPUInt32 retCount);

Description:

The functionality is same as viRead.

Parameter:



Table 6-3 RCT viRead parameter

Parameter	Direction	Parameter
vi	IN	The VI GPIB handler
buf	IN/OUT	A pointer pointing to a space for storing the result of GPIB command
count	IN	Number of bytes should be read
retCount	IN/OUT	A pointer pointing to a space for storing the number of bytes actually read

6.2.1.3 RCT_viWrite

Definition:

ViStatus RCT_viWrite(ViSession vi, ViBuf buf, ViUInt32 count, ViPUI32 retCount);

Description:

The functionality is same as viWrite.

Parameter:

Table 6-4 RCT viWrite parameter

Parameter	Direction	Parameter
vi	IN	The VI GPIB handler
buf	IN	GPIB command
count	IN	The length of GPIB command (Unit: byte)
retCount	IN/OUT	A pointer pointing to a space for storing the number of bytes actually wrote

6.2.1.4 RCT_ResultTokenization

Definition:

void RCT_Vi_ResultTokenization(unsigned int max_count = UINT_MAX);

Description:

Separate the result buffer "m_ViResultBuffer", the member of class CViControlUtility, into string tokens among commas

6.2.1.5 GpibLogFunction

Definition:

int __cdecl GpibLogFunction(const char* fmt, ...);

int GpibLogFunction(const char* fmt, va_list arg);



Description:

The functions provides to add log in the log of calibration flow.

Parameter:

Table 6-5 GpibLogFunction parameter

Parameter	Direction	Parameter
fmt	IN	Output format
arg	IN	Output paramters

6.2.2 Sample Code

```

sprintf_s(m_ViCmdBuffer, RCT_VI_BUFFER_LENGTH, "CALL:TCH 65");
if(VI_SUCCESS != RCT_viWrite(GetViHandle(),
                             (unsigned char*)m_ViCmdBuffer, strlen(m_ViCmdBuffer), &length))
{
    return _VI_ERROR;
}

sprintf_s(m_ViCmdBuffer, RCT_VI_BUFFER_LENGTH, "READ:ETXP?");
if(VI_SUCCESS != RCT_viQueryf (GetViHandle(), m_ViCmdBuffer, "%s", m_ViResultBuffer))
{
    return _VI_ERROR;
}

RCT_Vi_ResultTokenization();
return atof(m_ViResultArray[1]); /// return measured power

```

MediaTek Confidential
© 2017 MediaTek Inc.
This document contains information that is proprietary to MediaTek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
Classification:Confidential B



6.3 Exported Instrument calibration control functions

6.3.1 Exported Instrument Common Control Functions

6.3.1.1 RCTLIB_Common_Initialize

Definition:

int __stdcall RCTLIB_Common_Initialize(const S_RCTLIB_INIT_CFG_T *, size_t arg_size);

Description:

The function takes the parameter as the RCT configuration file path and initialize the GPIB connection as configured.

Parameter:

Table 6-6 RCTLIB common initialize parameter

Parameter	Direction	Parameter
arg	IN	The input argument (Please reference to Releated Data Structure)
arg_size	IN	The size of the input argument

Related Data Structure:

```
typedef struct
{
    /// full file path to the CFG file
    char *cfg_file_path;

    /// the instrument type
    int i_device_type;

    /// init GGE tester
    bool b_init_GGE;

    /// the instrument type for WCDMA calibration
    int i_device_type_wcdma;

    /// init WCDMA tester
    bool b_init_WCDMA;

    /// the instrument type for TD calibration
    int i_device_type_tdscdma;

    /// init TDSCDMA tester
    bool b_init_TDSCDMA;

    /// the instrument type for LTE calibration
    int i_device_type_lte;

    /// init LTE tester
    bool b_init_LTE;

    /// application handle
    void *applicationHandle;

    /// callback function pointer for application;
    ApplicationLoggingCallback_T LogFunction;

    /// callback function to check the user termination
    ApplicationStopCallback_T CheckStopFunction;
}S_RCTLIB_INIT_CFG_T;
```

```

/// the instrument type
typedef enum
{
    RCTLIB_DEVICE_UNDEF = -1
    ,RCTLIB_DEVICE_AGILENT_8960 = 0
    ,RCTLIB_DEVICE_CMU_200
    ,RCTLIB_DEVICE_CMW_500
    ,RCTLIB_DEVICE_StarPoint_6010
    ,RCTLIB_DEVICE_MT_8820
    ,RCTLIB_DEVICE_CALLBACK /// Customized Instrument Interface
    ,RCTLIB_DEVICE_DUMMY
    ,RCTLIB_DEVICE_NUM
}E_RCTLIB_DEVICE_TYPE;
    
```

6.3.1.2 RCTLIB_Common_Connect

Definition:

```
int __stdcall RCTLIB_Common_Connect(void);
```

Description:

The instrument connection will be established when **RCTLIB_Common_Initialize** return success. Currently, this function is a dummy function which always return success.

Parameter:

N/A

6.3.1.3 RCTLIB_Common_Disconnect

Definition:

```
int __stdcall RCTLIB_Common_Disconnect(void);
```

Description:

Disconnect the connection of instrument.

Parameter:

N/A

6.3.1.4 RCTLIB_Common_Reset

Definition:

```
int __stdcall RCTLIB_Common_Reset(void);
```

Description:

Reset the connected instrument.

Parameter:

N/A



6.3.1.5 RCTLIB_Common_SetApplicationFormat

Definition:

```
int __stdcall RCTLIB_Common_SetApplicationFormat(unsigned int application_format);
```

Description:

The format of instrument is controlled by calibration flow in calibration library. Currently, this function is a dummy function which always return success.

Parameter:

Table 6-7 RCTLIB common setApplicationFormat parameter

Parameter	Direction	Parameter
application_format	IN	The format of modulation.

Related Data Structure:

```
/* Appliaction format */
typedef enum
{
    E_INSTRMENT_APP_FORMAT_GSM_GPRS = 0
    ,E_INSTRMENT_APP_FORMAT_WCDMA    = 1
    ,E_INSTRMENT_APP_FORMAT_TDSCDMA  = 2
    ,E_INSTRMENT_APP_FORMAT_LTE      = 3
    ,E_INSTRMENT_APP_FORMAT_COUNT
}E_INSTRMENT_APP_FORMAT;
```

6.3.1.6 RCTLIB_Common_SetOperatingMode

Definition:

```
int __stdcall RCTLIB_Common_SetOperatingMode(unsigned int operating_mode);
```

Description:



The operating mode of instrument is controlled by calibration flow in calibration library. Currently, this function is a dummy function which always return success.

Parameter:

Table 6-8 RCTLIB common setOperatingMode parameter

Parameter	Direction	Parameter
operating_mode	IN	The operation mode of instrument

Related Data Structure:

```
/// RCT operating mode enum value (Active Cell)
const unsigned int RCTLIB_OPERATING_MODE_ACTIVE = 0;

/// RCT operating mode enum value (GSM BCH slot:0 BCCH, other: dummy)
const unsigned int RCTLIB_OPERATING_MODE_GSM_BCH = 1;

/// RCT operating mode enum value (GSM BCH slot:0 BCCH, TCH and dummy)
const unsigned int RCTLIB_OPERATING_MODE_GSM_BCH_TCH = 2;

/// RCT operating mode enum value (continuous wave)
const unsigned int RCTLIB_OPERATING_MODE_GSM_CW = 3;

/// RCT operating mode enum value (GPRS BCH slot:0 BCCH other:dummy)
const unsigned int RCTLIB_OPERATING_MODE_GPRS_BCH = 4;

/// RCT operating mode enum value (GPRS BCH slot:0 BCCH, PDTCH and dummy)
const unsigned int RCTLIB_OPERATING_MODE_GPRS_BCH_PDTCH = 5;

/// RCT operating mode enum value (EGPRS BCH slot:0 other:dummy)
const unsigned int RCTLIB_OPERATING_MODE_EGPRS_BCH = 6;
```

```
/// RCT operating mode enum value (GPRS BCH slot:0 BCCH, PDTCH and dummy)
```

```
const unsigned int RCTLIB_OPERATING_MODE_EGPRS_BCH_PDTCH = 7;
```

```
/// RCT operating mode enum value (GSM fast handset calibration mode)
```

```
const unsigned int RCTLIB_OPERATING_MODE_FDT = 8;
```

```
/// RCT operating mode enum value (WCDMA FDD test mode)
```

```
const unsigned int RCTLIB_OPERATING_MODE_WCDMA_FDD = 9;
```

```
/// RCT operating mode enum value (cell off)
```

```
const unsigned int RCTLIB_OPERATING_MODE_CELL_OFF = 10;
```

```
/// RCT operating mode enum value (TD test mode)
```

```
const unsigned int RCTLIB_OPERATING_MODE_TD_TEST = 11;
```

```
/// RCT operating mode enum value (Signaling mode)
```

```
const unsigned int RCTLIB_OPERATING_MODE_SIGNALING = 12;
```

```
/// RCT operating mode enum value (Nonsignaling mod)
```

```
const unsigned int RCTLIB_OPERATING_MODE_NONSIGNALING = 13
```

6.3.1.7 RCTLIB_Common_Deinitialize

Definition:

```
int __stdcall RCTLIB_Common_Deinitialize(void);
```

Description:

Deinitialize the instrument those initialized and free the resources.

Parameter:

N/A

6.3.1.8 RCTLIB_Common_CheckStat

Definition:

```
int __stdcall RCTLIB_CheckStat(int status);
```



Description:

The function is called by calibration flow in calibration library. This function translate the return state of VI GPIB control functions to the related return state of instrument library.

Parameter:

N/A

6.3.1.9 RCTLIB_Common_GetDeviceString

Definition:

const char* __stdcall RCTLIB_Common_GetDeviceString(E_RCTLIB_DEVICE_TYPE type);

Description:

This function return the char string name of the instrument which is specific in parameter “E_RCTLIB_DEVICE_TYPE type”.

Parameter:

Table 6-9 RCTLIB common getDeviceString parameter

Parameter	Direction	Parameter
type	IN	The instrument type

Related Data Structure:

```
/// the instrument type
```

```
typedef enum
```

```
{
```

```
    RCTLIB_DEVICE_UNDEF = -1,
```

```
    RCTLIB_DEVICE_AGILENT_8960 = 0,
```

```
    RCTLIB_DEVICE_CMU_200 = 1,
```

```
    RCTLIB_DEVICE_CMW_500 = 2,
```

```
    RCTLIB_DEVICE_StarPoint_6010 = 3,
```

```
    RCTLIB_DEVICE_MT_8820 = 4,
```

```
    RCTLIB_DEVICE_MT_8870 = 5,
```

```
    RCTLIB_DEVICE_CTP_3110 = 6,
```

```
    RCTLIB_DEVICE_Transcom_6280 = 7,
```

```
    RCTLIB_DEVICE_AGILENT_EXT = 8,
```

```
    RCTLIB_DEVICE_LP_IQXSTREAM = 9,
```

```
    RCTLIB_DEVICE_PXI_3000 = 10,
```

```
    RCTLIB_DEVICE_CALLBACK = 11, /// Customized Instrument Interface
```

```
    RCTLIB_DEVICE_DUMMY,
```

```
    RCTLIB_DEVICE_NUM
```

```
}E_RCTLIB_DEVICE_TYPE;
```

6.3.1.10 RCTLIB_Common_GetDLLVer

Definition:

```
int __stdcall RCTLIB_Common_GetDLLVer(unsigned int *major_ver, unsigned int *minor_ver, unsigned int *build_num, unsigned int *patch_num);
```

Description:

This function return the char string name of the instrument which is specific in parameter "E_RCTLIB_DEVICE_TYPE type".

Parameter:

Table 6-10 RCTLIB common getDLLVer parameter

Parameter	Direction	Parameter
major_ver	IN/OUT	A pointer pointing to a space for storing the DLL major version
minor_ver	IN/OUT	A pointer pointing to a space for storing the DLL minor version
build_num	IN/OUT	A pointer pointing to a space for storing the DLL build version
patch_num	IN/OUT	A pointer pointing to a space for storing the DLL patch version

6.3.1.11 RCTLIB_Common_GetInsAppFormatSupport

Definition:

```
void __stdcall RCTLIB_Common_GetInsAppFormatSupport(E_RCTLIB_DEVICE_TYPE eDeviceNum,
E_INSTRUMENT_APP_FORMAT eAppFormat, bool* bTrandCalSupport, bool* bFastCalSupport);
```

Description:

This function is used to query the supporting of traditional calibration and fast handset calibration by specified instrument and application format.

Parameter:

Table 6-11 RCTLIB common getInsAppFormatSupport parameter

Parameter	Direction	Parameter
eDeviceNum	IN	The instrument type
eAppFormat	IN	The application format
bTrandCalSupport	IN/OUT	A pointer pointing to a space for storing the traditional calibration capability.
bFastCalSupport	IN/OUT	A pointer pointing to a space for storing the fast handset calibration capability.

6.3.1.12 RCTLIB_Common_SwitchPort

Definition:

```
int __stdcall RCTLIB_Common_SwitchPort(E_INSTRUMENT_APP_FORMAT eRctToChange, E_RCTLIB_PORT_TYPE
ePort)
```

```
/* Instrument port enum */
```

```
typedef enum
```

```
{
```

```
    RCTLIB_RF_MAIN_INOUT_PORT = 0,
```

```
    RCTLIB_RF_RXD_OUTPUT_PORT = 1,
```

```
    RCTLIB_RF_INOUT_PORT1 = 2,
```

```
RCTLIB_RF_INOUT_PORT2 = 3,
RCTLIB_RF_INOUT_PORT3 = 4,
RCTLIB_RF_INOUT_PORT4 = 5,
RCTLIB_RF_PORT_TYPE_END
} E_RCTLIB_PORT_TYPE;
```

Description:

Command Instrument to switch RF port, if instrument is supported.

Parameter:

Table 6-12 RCTLIB common switchPort parameter

Parameter	Direction	Parameter
eRctToChange	IN	The specific application format to switch RF port.
ePort	IN	The specific RF port to switch to.

6.3.1.13 RCTLIB_OccupyHandler

Definition:

```
int __stdcall RCTLIB_OccupyHandler(int meta_handle);
```

Description:

This function is used to occupy the multi-thread handler for multi- instrument control. Users must use reentrant function with the handler which is occupied by this function. The naming of reentrant functions are added with r. For example: RCTLIB_Common_Initialize_r(...)

Parameter:

Table 6-13 RCTLIB occupyHandler parameter

Parameter	Direction	Parameter
meta_handle	IN	The handler number to occupy.



6.3.2 GSM & EDGE Exported Instrument Control functions

6.3.2.1 RCTLIB_GGE_Cableloss_Settings

Definition:

```
int __stdcall RCTLIB_GGE_Cableloss_Settings(S_RCTLIB_GGE_Cableloss_CONFIG_T cable_loss);
```

Description:

Set the cable loss to the instrument of GSM & EDGE application format

Parameter:

Table 6-14 RCTLIB GGE Cableloss settings parameter

Parameter	Direction	Parameter
cable_loss	IN	Uplink/Downlink cableloss of a specified band

Related Data Structure:

```
typedef struct
{
    /// frequency band
    unsigned band;
    double m_ul_cableloss[4];
    double m_dl_cableloss[4];
}S_RCTLIB_GGE_Cableloss_CONFIG_T;
```

6.3.2.2 RCTLIB_GGE_ConfigCellPower

Definition:

```
int __stdcall RCTLIB_GGE_ConfigCellPower(double power);
```

Description:

Set the cell power to the instrument of GSM & EDGE application format

Parameter:

Table 6-15 RCTLIB GGE ConfigCellPower parameter



Parameter	Direction	Parameter
power	IN	Cell power (dBm)

6.3.2.3 RCTLIB_GGE_ConfigDefaultSettings

Definition:

```
int __stdcall RCTLIB_GGE_ConfigDefaultSettings(void);
```

Description:

Set the instrument to the init state

Parameter:

N/A

6.3.2.4 RCTLIB_GGE_ConfigAnalyzerFrequencyOffset

Definition:

```
int __stdcall RCTLIB_GGE_ConfigAnalyzerFrequencyOffset(const  
S_RCTLIB_GGE_FREQUENCY_OFFSET_SETTINGS_T* pSettings, unsigned int sz);
```

Description:

Set the analyzer frequency offset

Parameter:

Table 6-16 RCTLIB GGE ConfigAnalyzerFrequencyOffset parameter

Parameter	Direction	Parameter
pSettings	IN	The structure for setting analyzer frequency offset
sz	IN	Reserved parameter

Related Data Structure:



```
typedef struct
{
    /// center frequency (Unit: MHz)
    double centerFrequency;
    /// frequency offset to center frequency (Unit: MHz)
    double frequencyOffset;
    /// manual control of the analyzer frequency offset (manual: 1, the center frequency/ frequency offset is
    /// used; auto: 0, both are not used)
    unsigned char manualControl;
}S_RCTLIB_GGE_FREQUENCY_OFFSET_SETTINGS_T;
```

6.3.2.5 RCTLIB_GGE_CAPID_PreSettings

Definition:

```
int __stdcall RCTLIB_GGE_CAPID_PreSettings(const S_RCTLIB_GGE_CAPID_SETTINGS_T* pSettings);
```

Description:

Config Instrument application format, band, channel, expected power, training sequence, cable loss, and measurement count for measuring UE’s frequency error

Parameter:

Table 6-17 RCTLIB GGE CAPID preSettings parameter

Parameter	Direction	Parameter
pSettings	IN	The structure for setting CAP ID calibration settings

Related Data Structure:

```
typedef struct
{
    /// band indicator (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4: PCS1900)
    unsigned int band;
    /// measurement ARFCN
    unsigned int arfcn;
    /// expected MS TX PCL
    unsigned int pcl;
    /// expected MS TX TSC
    unsigned int tsc;
    /// number of multiple measurement
    unsigned int measurementCount;
    /// measurement timeout setting (ms)
    double timeout;
}S_RCTLIB_GGE_CAPID_SETTINGS_T;
```

6.3.2.6 RCTLIB_GGE_CAPID_Iteration

Definition:

```
int __stdcall RCTLIB_GGE_CAPID_Iteration(double *frequency_error);
```

Description:

Initiate instrument frequency measurement and fetch frequency error

Parameter:

Table 6-18 RCTLIB GGE CAPID Iteration parameter

Parameter	Direction	Parameter
frequency_error	IN/OUT	A pointer pointing to a space for storing the frequency error.

6.3.2.7 RCTLIB_GGE_AFC_PreSettings

Definition:

```
int __stdcall RCTLIB_GGE_AFC_PreSettings(const S_RCTLIB_GGE_AFC_SETTINGS_T* pSettings);
```

Description:



Config Instrument GSM application format, operating mode, band, channel, cable loss, and downlink power.
When the CWmode set to true, the instrument should using continuous waveform mode.

Parameter:**Table 6-19 RCTLIB GGE AFC PreSettings parameter**

Parameter	Direction	Parameter
pSettings	IN	The structure for setting AFC calibration settings

Related Data Structure:

```
typedef struct
{
    /// band indicator (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4: PCS1900)
    unsigned int band;
    /// BCH ARFCN
    unsigned int arfcn;
    /// downlink power for used timeslot (dBm)
    double d_used;
    /// downlink power for unused timeslot (dB)
    double d_unused;
    /// frequency offset
    double d_offset;
    /// Continuous wave mode indicator
    unsigned char CWmode;
    /// TSC
    unsigned int tsc;
}S_RCTLIB_GGE_AFC_SETTINGS_T;
```

6.3.2.8 RCTLIB_GGE_AGC_PreSettings**Definition:**

```
int __stdcall RCTLIB_GGE_AGC_PreSettings(double d_power);
```

Description:

Config Instrument application format, operating mode, cable loss, and cell power.

Parameter:



Table 6-20 RCTLIB GGE AGC preSettings parameter

Parameter	Direction	Parameter
d_power	IN	Instrument cell power

6.3.2.9 RCTLIB_GGE_AGC_ChangeCellBand

Definition:

```
int __stdcall RCTLIB_GGE_AGC_ChangeCellBand(unsigned int band);
```

Description:

Config Instrument band setting.

Parameter:

Table 6-21 RCTLIB GGE AGC ChangeCellBand parameter

Parameter	Direction	Parameter
band	IN	band indicator (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4: PCS1900)

6.3.2.10 RCTLIB_GGE_AGC_ChangeChannel

Definition:

```
int __stdcall RCTLIB_GGE_AGC_ChangeChannel(unsigned int arfcn);
```

Description:

Config Instrument channel

Parameter:

Table 6-22 RCTLIB GGE AGC ChangeCellBand parameter

Parameter	Direction	Parameter
arfcn	IN	Channel



6.3.2.11 RCTLIB_GGE_APCDCOffset_PreSettings

Definition:

int __stdcall RCTLIB_GGE_APCDCOffset_PreSettings(unsigned int tsc);

Description:

Config Instrument application format, operating mode, band, cable loss, and training sequence.

Parameter:

Table 6-23 RCTLIB GGE APCDCOffset preSettings parameter

Parameter	Direction	Parameter
tsc	IN	Training sequence

6.3.2.12 RCTLIB_GGE_APCDCOffset_ChangeCellBand

Definition:

int __stdcall RCTLIB_GGE_APCDCOffset_ChangeCellBand(unsigned int band)

Description:

Config Instrument band setting

Parameter:

Table 6-24 RCTLIB GGE APCDCOffset ChangeCellBand parameter

Parameter	Direction	Parameter
tsc	IN	Training sequence

6.3.2.13 RCTLIB_GGE_APCDCOffset_Iteration

Definition:

int __stdcall RCTLIB_GGE_APCDCOffset_Iteration(unsigned int arfcn, double expected_power,int PCL, double *d_power);

Description:



Config Instrument channel and expected_power(or PCL). And, measure UE's TX power.

Parameter:

Table 6-25 RCTLIB GGE APCDCOffset Iteration parameter

Parameter	Direction	Parameter
arfcn	IN	Channel
expected_power	IN	Instrument expected power (dB)
PCL	IN	Instrument expected power in PCL
d_power	IN/OUT	A pointer pointing to a space for storing the UE's power.

6.3.2.14 RCTLIB_GGE_EDGE_APCDCOffset_PreSettings

Definition:

```
int __stdcall RCTLIB_GGE_APCDCOffset_PreSettings(unsigned int tsc);
```

Description:

Config Instrument application format, band, cable loss, and training sequence.

Parameter:

Table 6-26 RCTLIB GGE EDGE APCDCOffset preSettings parameter

Parameter	Direction	Parameter
tsc	IN	Training sequence

6.3.2.15 RCTLIB_GGE_EDGE_APCDCOffset_ChangeCellBand

Definition:

```
int __stdcall RCTLIB_GGE_APCDCOffset_ChangeCellBand(unsigned int band)
```

Description:

Config Instrument band setting

Parameter:

Table 6-27 RCTLIB_GGE_EDGE_APCDCOffset_ChangeCellBand parameter

Parameter	Direction	Parameter
band	IN	band indicator (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4: PCS1900)

6.3.2.16 RCTLIB_GGE_EDGE_APCDCOffset_Iteration

Definition:

```
int __stdcall RCTLIB_GGE_APCDCOffset_Iteration(unsigned int arfcn, double expected_power,int PCL, double *d_power);
```

Description:

Config Instrument channel and expected_power(or PCL). And, measure UE's TX power.

Parameter:

Table 6-28 RCTLIB_GGE_EDGE_APCDCOffset iteration parameter

Parameter	Direction	Parameter
arfcn	IN	Channel
expected_power	IN	Instrument expected power (dB)
PCL	IN	Instrument expected power in PCL
d_power	IN/OUT	A pointer pointing to a space for storing the UE's power.

6.3.2.17 RCTLIB_GGE_FBDAC_PreSettings

Definition:

```
int __stdcall RCTLIB_GGE_FBDAC_PreSettings(const S_RCTLIB_GGE_FBDAC_SETTINGS_T* pSettings);
```

Description:

Config Instrument GSM application format, operating mode, band, TCH/BCH channel, cable loss, and expected power.

**Parameter:****Table 6-29 RCTLIB GGE FBDAC presettings parameter**

Parameter	Direction	Parameter
pSettings	IN	The structure for setting FB DAC calibration settings

Related Data Structure:

```
typedef struct
{
    /// band indicator (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4: PCS1900)
    unsigned int band;
    /// BCH & TCH ARFCN
    unsigned int arfcn;
    /// expected MS TX PCL
    unsigned int pcl;
    /// TSC
    unsigned int tsc;
    /// Measurement count
    unsigned int iMeasureCount;
    /// config power time out
    unsigned int iTimeout;
}S_RCTLIB_GGE_FBDAC_SETTINGS_T;
```

6.3.2.18 RCTLIB_GGE_FBDAC_Iteration**Definition:**

```
int __stdcall RCTLIB_GGE_FBDAC_Iteration(double* d_power);
```

Description:

Measure UE's power

Parameter:**Table 6-30 RCTLIB GGE FBDAC iteration parameter**

Parameter	Direction	Parameter
pSettings	IN	The structure for setting FB DAC calibration settings



6.3.2.19 RCTLIB_GGE_TXIQ_PreSettings

Definition:

int __stdcall RCTLIB_GGE_TXIQ_PreSettings(const S_RCTLIB_GGE_TXIQ_SETTINGS_T* pSettings);

Description:

Config Instrument GSM application format, operating mode, training sequence, measurement count, measurement time out, cable loss, and coding scheme.

Parameter:

Table 6-31 RCTLIB GGE TXIQ presets parameter

Parameter	Direction	Parameter
pSettings	IN	The structure for setting TX IQ measurement settings

Related Data Structure:

```
typedef struct
{
    /// band indicator (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4: PCS1900)
    unsigned int band;
    /// TCH & BCH ARFCN
    unsigned int arfcn;
    /// expected MS TX PCL
    unsigned int pcl;
    /// TSC
    unsigned int tsc;
    /// coding scheme
    unsigned int mcs;
    /// EPSK (1: EPSK, 0:GMSK)
    unsigned int epsk;
    /// config power time out
    unsigned int iTimeout;
    /// measurement count
    unsigned int iMeasureCount;
}S_RCTLIB_GGE_TXIQ_SETTINGS_T;
```



6.3.2.20 RCTLIB_GGE_TXIQ_ChangeBand

Definition:

```
int __stdcall RCTLIB_GGE_TXIQ_ChangeBand(const S_RCTLIB_GGE_TXIQ_SETTINGS_T* pSettings);
```

Description:

Config Instrument band, TCH/BCH channel, and expected power.

Parameter:

Table 6-32 RCTLIB GGE TXIQ ChangeBand parameter

Parameter	Direction	Parameter
pSettings	IN	The structure for setting TX IQ measurement settings

6.3.2.21 RCTLIB_GGE_TXIQ_Iteration

Definition:

```
int __stdcall RCTLIB_GGE_TXIQ_Iteration(const S_RCTLIB_GGE_TXIQ_SETTINGS_T* pSettings, S_RCTLIB_GGE_TXIQ_RESULT_T *pTxIqResult);
```

Description:

Measure the average IQ imbalance result

Parameter:

Table 6-33 RCTLIB GGE TXIQ iteration parameter

Parameter	Direction	Parameter
pSettings	IN	The structure for setting TX IQ measurement settings
pTxIqResult	IN/OUT	The structure for fetching TX IQ measurement results

Related Data Structure:

Mediatek Confidential
This document contains information that is proprietary to Mediatek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
© 2017 Mediatek Inc.
Classification: Confidential B



```
typedef struct
{
    double d_sbs; // carrier frequency
    double d_oos; // +67.7083 kHz
}S_RCTLIB_GGE_TXIQ_RESULT_T;
```

6.3.2.22 RCTLIB_GGE_TXSlopeSkew_PreSettings

Definition:

int __stdcall RCTLIB_GGE_TXSlopeSkew_PreSettings(const S_RCTLIB_GGE_TXSLOPESKEW_SETTINGS_T* pSettings);

Description:

Config Instrument GSM application format, operating mode, band, training sequence, TCH/BCH channel, coding scheme, cable loss, and expected power.

Parameter:

Table 6-34 RCTLIB GGE TXSlopeSkew presettings parameter

Parameter	Direction	Parameter
pSettings	IN	The structure for setting TX Slope Skew calibration settings

Related Data Structure:

MediaTek Confidential
This document contains information that is proprietary to MediaTek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
© 2017 MediaTek Inc.
Classification: Confidential B

```
typedef struct
{
    /// band indicator (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4: PCS1900)
    unsigned int band;
    /// TCH & BCH ARFCN
    unsigned int arfcn;
    /// expected MS TX PCL
    unsigned int pcl;
    /// TSC
    unsigned int tsc;
    /// MCS
    unsigned int mcs;
    /// Measurement count
    unsigned int iMeasureCount;
    /// config power time out
    unsigned int iTimeout;
}S_RCTLIB_GGE_TXSLOPESKEW_SETTINGS_T;
```

6.3.2.23 RCTLIB_GGE_TXSlopeSkew_Iteration

Definition:

```
int __stdcall RCTLIB_GGE_TXSlopeSkew_Iteration(double* d_mod_depth);
```

Description:

Measure the maximum section { 28US, 56US } in power v.s time and the average of minimum section {15US, 42US} and { 42US, 70US } in power v.s time. And then, calculate the TX slope skew by the maximum section minus the average of two minimum section.

Parameter:

Table 6-35 RCTLIB GGE TXSlopeSkew iteration parameter

Parameter	Direction	Parameter
d_mod_depth	IN	A pointer pointing to a space for storing the UE's TX slope skew.

6.3.2.24 RCTLIB_GGE_TRXOffset_PreSettings

Definition:



int __stdcall RCTLIB_GGE_TRXOffset_PreSettings(const S_RCTLIB_GGE_TRXOFFSET_SETTINGS_T* pSettings);

Description:

Config Instrument GSM application format, operating mode, cable loss, and training sequence.

Parameter:

Table 6-36 RCTLIB GGE TRXOffset presetsings

Parameter	Direction	Parameter
pSettings	IN	The structure for setting TRX Offset calibration settings

Related Data Structure:

```
typedef struct
{
    /// band indicator (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4: PCS1900)
    unsigned int band;
    /// TCH & BCH ARFCN
    unsigned int arfcn;
    /// expected MS TX PCL
    unsigned int pcl;
    /// cell power (dbm)
    double d_used;
    /// TSC
    unsigned int tsc;
    /// time slot
    unsigned int iTimeslot;
    /// Measurement count
    unsigned int iMeasureCount;
    /// config power time out
    unsigned int iTimeout;
}S_RCTLIB_GGE_TRXOFFSET_SETTINGS_T;
```


6.3.2.25 RCTLIB_GGE_TRXOffset_InitAFC

Definition:

```
int __stdcall RCTLIB_GGE_TRXOffset_InitAFC(const S_RCTLIB_GGE_TRXOFFSET_SETTINGS_T* pSettings);
```

Description:

Config Instrument band, cell power, TCH/BCH channel, TCH timeslot, expected power in PCL, and phase error detection.

Parameter:

Table 6-37 RCTLIB GGE TRXOffset InitAFC parameter

Parameter	Direction	Parameter
pSettings	IN	The structure for setting TRX Offset calibration settings

6.3.2.26 RCTLIB_GGE_TRXOffset_Iteration

Definition:

```
int __stdcall RCTLIB_GGE_TRXOffset_Iteration(double *frequency_err);
```

Description:

Measure the average frequency error in Hz

Parameter:

Table 6-38 RCTLIB GGE TRXOffset iteration parameter

Parameter	Direction	Parameter
pSettings	IN	The structure for setting TRX Offset calibration settings

6.3.2.27 RCTLIB_GGE_FHC_DTS_PreSettings

Definition:

```
int __stdcall RCTLIB_GGE_FHC_DTS_PreSettings();
```

Description:

Config Instrumnet GSM application format, fast handset calibration downlink mode, and cable loss.

Parameter:

N/A

6.3.2.28 RCTLIB_GGE_FHC_DTS_Iteration

Definition:

int __stdcall RCTLIB_GGE_FHC_DTS_Iteration(S_RCTLIB_FHC_DL_List_T List, int ListLength);

Description:

The downlink of fast handset calibration is arranged the UE expected frames at first to decrease the GPIB command handshake time. After the Instrumnet arrangments is funished, UE can start to receice these arranged frames when FHC start. Therefore, the instrumnet must follows the "List" structure frame settings to send RX frames to UE, or the calibration will fail.

The format of the FHC frames can be divided into three stages: the first stage is contained with FCB for frequency error adjustment, the second stage is contained with SCB for timing adjustment, and the last stage contained with all dummy slots for path loss calibration as shown below.

FCB frame:

DUMMY	FCB	DUMMY	DUMMY	DUMMY	DUMMY	DUMMY
-------	-----	-------	-------	-------	-------	-------

SCB frame:

SCB	DUMMY	DUMMY	DUMMY	DUMMY	DUMMY	DUMMY
-----	-------	-------	-------	-------	-------	-------

Dummy frame:

DUMMY	DUMMY	DUMMY	DUMMY	DUMMY	DUMMY	DUMMY
-------	-------	-------	-------	-------	-------	-------

For the instrumnet which is using waveform style RF generator, the recommanded waveform format should be like the table as shown below.

FHC three stage frames:

Table 6-39 FHC three stage frames



Frequency error adjustment	Timing adjustment	Path loss calibration
FCB frames(MAX: 80 frames)	SCB frames(MAX:51 frames)	Dummy frames(MAX: 80 frames)

However, not all MTK solution will take all the FCB frames and SCB frames when FHC. Therefore, the SCB frame number in the waveform must set in the *RCTLIB_GGE_GET_SpecificSettings()* for assigning the SCB frame number to the UE. Others, those remaining FCB frames, not used in frequency error adjustment, will be skipped automatically by UE until met the first SCB frame.

Parameter:**Table 6-40 RCTLIB GGE FHC DTS iteration parameter**

Parameter	Direction	Parameter
List	IN	The structure for setting FHC RX calibration settings
ListLength	IN	The total number of arrangements in the List structure

Related Data Structure:

```
typedef enum
{
    RCTLIB_DB_FCB_DB = 0,    /// DUMMY,FCB,DUMMY,DUMMY,DUMMY,DUMMY
    RCTLIB_FSB_DB,          /// FSB,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY
    RCTLIB_DB                /// DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY
} E_RCTLIB_FHC_DL_BURST_TYPE_T;

typedef struct
{
    /// The expected frame type of arrangement Xth
    E_RCTLIB_FHC_DL_BURST_TYPE_T BurstType[100];
    /// The expected channel(in frequency) of arrangement Xth
    double    Freq_MHz[100];
    /// The expected repeat count of arrangement Xth
    int        RepeatCount[100];
    /// The expected cell power (in dBm) of arrangement Xth
    double    Power_dBm[100];
    /// The expected band of arrangement Xth. (0: GSM450 / 1: GSM850 / 2: GSM900 / 3: DCS1800 / 4:
    PCS1900)
    unsigned char band[100];
    /// reserved
    short    arfcn;
    /// The UE's step length
    int        step_count;
    /// Flag to indicate this round will calibrate the frequency error or not
    bool        doAFC;
} S_RCTLIB_FHC_DL_List_T;
```

6.3.2.29 RCTLIB_GGE_FHC_DTS_START

Definition:

```
int __stdcall RCTLIB_GGE_FHC_DTS_START();
```

Description:

Trigger the instrument starting to send arranged RX frames to UE

Parameter:

N/A

6.3.2.30 RCTLIB_GGE_FHC_DTS_STOP

Definition:

```
int __stdcall RCTLIB_GGE_FHC_DTS_STOP ();
```

Description:

Stop the instrument starting to send arranged RX frames to UE

Parameter:

N/A

6.3.2.31 RCTLIB_GGE_FHC_UTS_PreSettings

Definition:

```
int __stdcall RCTLIB_GGE_FHC_DTS_PreSettings();
```

Description:

Config Instrumnet GSM application format, fast handset calibration uplink mode, and cable loss.

Parameter:

N/A

6.3.2.32 RCTLIB_GGE_FHC_UTS_Iteration

Definition:

```
int __stdcall RCTLIB_GGE_FHC_UTS_Iteration( S_RCTLIB_GGE_FHC_TX_UTS_T *uts);
```

Description:

The uplink of fast handset calibration is arranged to measure the UE's TX power of each frame. The frame step of each round should be arranged like the following figure to measure UE's TX power.

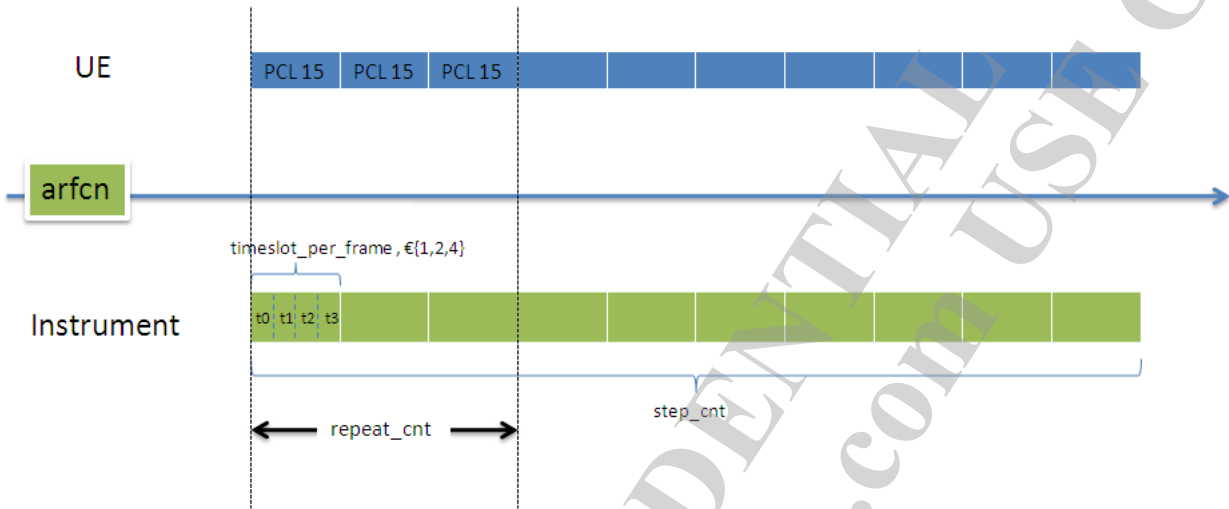


Figure 6-2 RCTLIB GGE FHC UTS iteration

Parameter:

Table 6-41 RCTLIB GGE FHC UTS iteration parameter

Parameter	Direction	Parameter
uts	IN	The structure for storing GGE FHC TX UTS request (50 USS)

Related Data Structure:

```
typedef struct
{
    /// frequency band indicator for the USS
    unsigned char    band;
    /// TCH/PDTCH ARFCN for the USS
    unsigned int     arfcn;
    /// TCH/PDTCH MHz for the USS
    double           d_mHz;
    /// mult-slot configuration for each frame in the USS
    char             timeslot_per_frame;
    /// ===== UE's settings =====
    /// APC DAC or PCL (0: PCL 1: DAC)
    unsigned char    apc_dac_pcl_sel;
    /// APC DAC selection for the USS (1: APC DAC 0: PCL)
    short            apc_dac_pcl_value[4];
    /// PA vbias setting
    unsigned char    pa_vbias_val[4];
    /// low PCL
    unsigned char    is_low_pcl[4];
    /// =====
    /// repeat frame count in the USS
    int              repeat_cnt;
}S_RCTLIB_GGE_FHC_TX_USS_T;

typedef struct
{
    /// number of USS in the UTS
    unsigned char    step_cnt;
    /// modulation (0:GMSK 1: EPSK)
    unsigned char    modulation;
    /// total samples
    short            s_measure_samples;
    /// UTS type (0: APC DAC calibration, 1: TX subband cal, 2: PCL check)
    unsigned char    uts_type;
    /// max expected nominal power
    double           d_max_expected_power;
    /// the number of uplink TDMA frames required in an uplink sequence step
    int              num_frame;
    /// USS configurations
    S_RCTLIB_GGE_FHC_TX_USS_T    ApcUSS[50];
}S_RCTLIB_GGE_FHC_TX_UTS_T;
```



6.3.2.33 RCTLIB_GGE_FHC_UTS_FetchResult

Definition:

```
int __stdcall RCTLIB_GGE_FHC_UTS_FetchResult(const S_RCTLIB_GGE_FHC_TX_UTS_T *uts, S_RCTLIB_GGE_FHC_TX_UTS_RESULT_T* uts_result);
```

Description:

Fetch results from the instrument and fill in uts_result

Parameter:

Table 6-42 RCTLIB GGE FHC UTS FetchResult parameter

Parameter	Direction	Parameter
uts	IN	The structure for storing GGE FHC TX UTS request (50 USS)
uts_result	IN/OUT	The structure for storing GGE FHC TX UTS measurement result (upto 50 USS)

Related Data Structure:

```
typedef struct
{
    /// TX power measurement of slot 0 ~ slot 3
    double d_power[4];
}S_RCTLIB_GGE_FHC_TX_USS_RESULT_T;

typedef struct
{
    /// step counts in the UTS measurement
    unsigned char step_cnt;
    /// measurement results for each USS
    S_RCTLIB_GGE_FHC_TX_USS_RESULT_T uss_result[50];
}S_RCTLIB_GGE_FHC_TX_UTS_RESULT_T;
```

6.3.2.34 RCTLIB_GGE_NSFT_PreSettings

Definition:


```
int __stdcall RCTLIB_GGE_NSFT_PreSettings(unsigned int measurement_count, unsigned int ber_count);
```

Description:

Config the instrument into non-signaling test mode and initial the basic settings

Parameter:

Table 6-43 RCTLIB GGE NSFT presettings parameter

Parameter	Direction	Parameter
measurement_count	IN	The measurement count of power measurement
ber_count	IN	The BER test measurement count

6.3.2.35 RCTLIB_GGE_NSFT_GMSKInit

Definition:

```
int __stdcall RCTLIB_GGE_NSFT_GMSKInit(const S_RCTLIB_GGE_NSFT_TESTCONFIG_T* nsft_config);
```

Description:

Config GSM NSFT testing setting, e.g. band, cell power, TCH/BCH channel, expected power, training sequence, and coding scheme.

Parameter:

Table 6-44 RCTLIB GGE NSFT GMSKInit parameter

Parameter	Direction	Parameter
nsft_config	IN	The structure for storing GGE NSFT TEST config

Related Data Structure:

```
typedef struct
{
    /// frequency band
    unsigned band;
    /// cell power
    double bch_power;
    /// TCH power reduction ( [BCH CELL POWER in BER] – [TCH CELL POWER in BER] )
    double tch_reduction;
    /// BCH arfcn
    unsigned int bch;
    /// TCH arfcn
    unsigned int tch;
    /// initial PCL
    unsigned int pcl;
    /// timeslot
    unsigned int timeslot;
    /// TSC
    unsigned int tsc;
    /// coding scheme
    unsigned int cs;
}S_RCTLIB_GGE_NSFT_TESTCONFIG_T;
```

6.3.2.36 RCTLIB_GGE_NSFT_EPSKinit

Definition:

```
int __stdcall RCTLIB_GGE_NSFT_EPSKinit(const S_RCTLIB_GGE_NSFT_TESTCONFIG_T* nsft_config);
```

Description:

Config EPSK NSFT testing setting, e.g. band, cell power, TCH/BCH channel, expected power, training sequence, and coding scheme.

Parameter:

Table 6-45 RCTLIB GGE NSFT EPSKinit parameter

Parameter	Direction	Parameter
nsft_config	IN	The structure for storing GGE NSFT TEST config

6.3.2.37 RCTLIB_GGE_NSFT_BERInit

Definition:

```
int __stdcall RCTLIB_GGE_NSFT_BERInit(const S_RCTLIB_GGE_NSFT_TESTCONFIG_T* nsft_config);
```

Description:

Config BER NSFT testing setting, e.g. band, cell power, TCH/BCH channel, expected power, training sequence, and coding scheme.

Parameter:

Table 6-46 RCTLIB GGE NSFT BERInit parameter

Parameter	Direction	Parameter
nsft_config	IN	The structure for storing BER NSFT TEST config

6.3.2.38 RCTLIB_GGE_NSFT_ChangePCL

Definition:

```
int __stdcall RCTLIB_GGE_NSFT_ChangePCL(int b_EPSK, unsigned int pcl);
```

Description:

Change Instrument expected power of TCH channel

Parameter:

Table 6-47 RCTLIB GGE NSFT ChangePCL parameter

Parameter	Direction	Parameter
b_EPSK	IN	EPSK/GMSK indicator (1: EPSK, 0: GMSK)
pcl	IN	Expected power in PCL

6.3.2.39 RCTLIB_GGE_NSFT_ReadGMSKPerformance

Definition:

```
int __stdcall RCTLIB_GGE_NSFT_ReadGMSKPerformance(S_RCTLIB_GGE_NSFT_GMSK_RESULT_T *gmsk_result);
```

Description:

Measure GGE NSFT TX power v.s time match mask or not, power, phase and frequency error, and output RF spectrum.

Parameter:

Table 6-48 RCTLIB GGE NSFT ReadGMSKPerformance parameter

Parameter	Direction	Parameter
nsft_config	IN	The structure for storing BER NSFT TEST config

Related Data Structure:

```
typedef struct
{
    S_RCTLIB_GGE_NSFT_TXPOWER_RESULT_T txp_result;
    S_RCTLIB_GGE_NSFT_PFER_RESULT_T pfer_result;
    S_RCTLIB_GGE_NSFT_ORFS_RESULT_T orfs_result;
}S_RCTLIB_GGE_NSFT_GMSK_RESULT_T;

/// The structure for storing GGE NSFT TX measurement result (power/pvt part)
typedef struct
{
    /// burst match or not
    bool b_burst_match;
    /// TX power value
    double tx_power;
}S_RCTLIB_GGE_NSFT_TXPOWER_RESULT_T;

/// The structure for storing GGE NSFT TX measurement result (PFER part)
typedef struct
{
    /// min frequency error
    double MinFErr;
    /// max frequency error
    double MaxFErr;
    /// average frequency error
    double AvgFErr;
    /// worst peak frequency error
    double WorstFErr;
    /// min peak phase error
    double MinPKPErr;
    /// max peak phase error
    double MaxPKPErr;
    /// average peak phase error
    double AvgPKPErr;
    /// min RMS phase error
    double MinRMSPErr;
    /// max RMS phase error
    double MaxRMSPErr;
    /// average RMS phase error
    double AvgRMSPErr;
}S_RCTLIB_GGE_NSFT_PFER_RESULT_T;
```

/// The structure for storing GGE NSFT TX measurement result (ORFS part)

```
typedef struct
{
    /// modulation specturm (relative)
    double mod_spectrum[22];
    /// modulation specturm (absolute)
    double mod_spectrum_abs[22];
    /// switching specturm
    double switch_spectrum[8];
    /// ref. power of modulation spectrum
    double mod_ref_power;
}S_RCTLIB_GGE_NSFT_ORFS_RESULT_T;
```

6.3.2.40 RCTLIB_GGE_NSFT_InitiateBER

Definition:

```
int __stdcall RCTLIB_GGE_NSFT_InitiateBER();
```

Description:

Initial BER measurement of the instrument

Parameter:

N/A

6.3.2.41 RCTLIB_GGE_NSFT_FetchBER

Definition:

```
int __stdcall RCTLIB_GGE_NSFT_FetchBER(S_RCTLIB_GGE_NSFT_BER_RESULT_T* ber_result);
```

Description:

Fetch BER measurement result

Parameter:

Table 6-49 RCTLIB GGE NSFT FetchBER parameter



Parameter	Direction	Parameter
ber_result	IN/OUT	The structure for storing GGE NSFT BER measurement result

Related Data Structure:

```
typedef struct
{
    /// integrity
    long FBERIntegrity;
    /// tested bits
    double FBERBitsTested;
    /// BER
    double FBERRatio;
    /// Bit error count
    double FBERCount;
    /// progress
    double progress;
    /// number of CRC errors
    double crc_error;
}S_RCTLIB_GGE_NSFT_BER_RESULT_T;
```

6.3.2.42 RCTLIB_GGE_NSFT_ReadEPSPPerformance**Definition:**

```
int __stdcall RCTLIB_GGE_NSFT_ReadEPSPPerformance(S_RCTLIB_GGE_NSFT_EPSK_RESULT_T *epsk_result);
```

Description:

Measure EPSP NSFT TX power v.s time match mask or not, power, EPSP multislot modulation accuracy, and output RF spectrum.

Parameter:

Table 6-50 RCTLIB GGE NSFT ReadEPSPPerformance parameter

Parameter	Direction	Parameter
result	IN/OUT	The structure for storing GGE NSFT EPSP measurement result

Related Data Structure:

```
typedef struct
{
    S_RCTLIB_GGE_NSFT_TXPOWER_RESULT_T txp_result;
    S_RCTLIB_GGE_NSFT_EMAC_RESULT_T ema_result;
    S_RCTLIB_GGE_NSFT_ORFS_RESULT_T orfs_result;
}S_RCTLIB_GGE_NSFT_EPSK_RESULT_T;

/// The structure for storing GGE NSFT EPSK measurement result (EVM part)
typedef struct
{
    /// EVM 95 percentile
    double EVM_95P;
    /// magnitude error 95 percentile
    double MagErr_95P;
    /// phase error 95 percentile Agilent8960 is not supported
    double Pherr_95P;
    /// peak EVM Agilent8960 is not supported
    double PK_EVM;
    /// RMS EVM
    double RMS_EVM;
    /// peak magnitude error
    double PK_MagErr;
    /// RMS magnitude error
    double RMS_MagErr;
    /// peak RMS magnitude error
    double RMS_MagErrPeak;
    /// peak phase error
    double PK_Pherr;
    /// RMS phase error
    double RMS_Pherr;
    /// peak RMS phase error
    double RMS_PherrPeak;
    /// original offset
    double orig_offset;
    /// frequency error
    double FErr;
    /// Amplitude droop
    double Amp_Droop;
    /// IQ imbalance
    double IQ_Imbalance;
    /// Timing Alignment (Agilent8960 need new application (application #?) CMU supported)
    double TA;
    /// AM/PM Alignment
```




```
typedef struct
{
    S_RCTLIB_GGE_NSFT_TXPOWER_RESULT_T txp_result;
    S_RCTLIB_GGE_NSFT_EMAC_RESULT_T ema_result;
    S_RCTLIB_GGE_NSFT_ORFS_RESULT_T orfs_result;
}S_RCTLIB_GGE_NSFT_EPSK_RESULT_T;

double AmPm;
}S_RCTLIB_GGE_NSFT_EMAC_RESULT_T;
```

6.3.2.43 RCTLIB_GGE_GET_SpecificSettings

Definition:

int __stdcall RCTLIB_GGE_GET_SpecificSettings(S_RCTLIB_GGE_SPECIFIC_SETTINGS_T *pSettings);

Description:

Get the special constant settings from instrument. e.g, TSC, midamble.. etc.

Parameter:

Table 6-51 RCTLIB GGE GET SpecificSettings parameter

Parameter	Direction	Parameter
pSettings	IN/OUT	the structure to get GSM & EDGE intrument's spcial settings

Related Data Structure:

Mediatek Confidential
This document contains information that is proprietary to MediaTek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
© 2017 Mediatek Inc.
Classification:Confidential B



```
typedef struct
{
    /// TX IQ normal burst type
    int iTxIQ_BurstTypeNB;
    /// NSFT GSM time slot
    int iNSFT_GSM_TimeSlot;
    /// NSFT EDGE time slot
    int iNSFT_EDGE_TimeSlot;
    /// NSFT GSM tsc
    int iNSFT_GSM_tsc;
    /// NSFT EDGE tsc
    int iNSFT_EDGE_tsc;
    /// FHC fine sync SCB number
    int iFHC_fsSCB;
}S_RCTLIB_GGE_SPECIFIC_SETTINGS_T;
```

6.3.3 WCDMA Exported Instrument Control functions

6.3.3.1 RCTLIB_WCDMA_ConfigCellPower

Definition:

int __stdcall RCTLIB_WCDMA_ConfigCellPower(double power);

Description:

Config the instrument cell power. Currently, this function is a dummy function which always return success.

Parameter:

Table 6-52 RCTLIB WCDMA ConfigCellPower parameter

Parameter	Direction	Parameter
pSettings	IN/OUT	the structure to get GSM & EDGE intrument's spcial settings

6.3.3.2 RCTLIB_WCDMA_ConfigDefaultSettings

Definition:

int __stdcall RCTLIB_WCDMA_ConfigDefaultSettings(void);



Description:

Set the instrument to the init state

Parameter:

N/A

6.3.3.3 RCTLIB_WCDMA_AFC_PreSetting

Definition:

int __stdcall RCTLIB_WCDMA_AFC_PreSetting(const S_RCTLIB_WCDMA_AFC_PRESETTINGS_T* pSettings, unsigned int size);

Description:

Configure the instrument channel and cell power.

Parameter:

Table 6-53 RCTLIB WCDMA AFC presetting parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA AFC presetting
size	IN	The size of WCDMA AFC presetting structure

Related Data Structure:

```
typedef struct
{
    /// UARFCN for AFC calibration
    unsigned int uarfcn_dl;
    /// cell power for AFC calibration
    double d_cell_power;
}S_RCTLIB_WCDMA_AFC_PRESETTINGS_T;
```

6.3.3.4 RCTLIB_WCDMA_AGC_PreSetting

Definition:

int __stdcall RCTLIB_WCDMA_AGC_PreSetting(const S_RCTLIB_WCDMA_AGC_PRESETTINGS_T* pSettings);



Description:

Configure the instrument cell power, CPICH power, PICH power, PCCPCH power, and DPCH power.

Parameter:

Table 6-54 RCTLIB WCDMA AGC presetting parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA AGC presetting

Related Data Structure:

```
typedef struct
{
    /// cell power for AGC calibration
    double cellPower;
    /// CPICH power (dB)
    double cpichPower;
    /// PICH power (dB)
    double pichPower;
    /// PCCPCH power (dB)
    double pccpchPower;
    /// DPCH power (dB)
    double dpchPower;
}S_RCTLIB_WCDMA_AGC_PRESETTINGS_T;
```

6.3.3.5 RCTLIB_WCDMA_AGC_ChangeCellBand

Definition:

int __stdcall RCTLIB_WCDMA_AGC_ChangeCellBand(unsigned int band);

Description:

Configure the instrument band setting.

Parameter:

Table 6-55 RCTLIB WCDMA AGC ChangeCellBand parameter

Mediatek Confidential
© 2017 Mediatek Inc.
This document contains information that is proprietary to Mediatek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
Classification: Confidential B



Parameter	Direction	Parameter
band	IN	Band indicator

6.3.3.6 RCTLIB_WCDMA_AGC_ChangeChannel

Definition:

```
int __stdcall RCTLIB_WCDMA_AGC_ChangeChannel(unsigned int uarfcn);
```

Description:

Configure the instrument channel number

Parameter:

Table 6-56 RCTLIB WCDMA AGC ChangeChannel parameter

Parameter	Direction	Parameter
uarfcn	IN	Channel number

6.3.3.7 RCTLIB_WCDMA_AGC_ChangeCellPower

Definition:

```
int __stdcall RCTLIB_WCDMA_AGC_ChangeCellPower(double cellPower);
```

Description:

Change cell power of the instrument

Parameter:

Table 6-57 RCTLIB WCDMA AGC ChangeCellPower parameter

Parameter	Direction	Parameter
cellPower	IN	Cell power (dBm)

6.3.3.8 RCTLIB_WCDMA_APC_PreSetting

Definition:

```
int __stdcall RCTLIB_WCDMA_APC_PreSetting(const S_RCTLIB_WCDMA_APC_PRESETTINGS_T* pSettings);
```

**Description:**

Config WCDMA application format, FDD operation mode, cable loss, TX power measurement timeout, interval, and trigger method of the instrument

Parameter:

Table 6-58 RCTLIB WCDMA APC presetting parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA APC preSetting

Related Data Structure:

```
typedef struct
{
    /// measurement timeout setting (ms)
    double timeout;
    /// measurement interval (us)
    double measurementInterval;
    /// trigger delay (us)
    double triggerDelay;
}S_RCTLIB_WCDMA_APC_PRESETTINGS_T;
```

6.3.3.9 RCTLIB_WCDMA_APC_ChangeCellBand**Definition:**

```
int __stdcall RCTLIB_WCDMA_APC_ChangeCellBand(unsigned int band);
```

Description:

Change band setting of the instrument

Parameter:

Table 6-59 RCTLIB WCDMA APC ChangeCellBand parameter

Parameter	Direction	Parameter
band	IN	Band indicator



6.3.3.10 RCTLIB_WCDMA_APC_ChangeChannel

Definition:

int __stdcall RCTLIB_WCDMA_APC_ChangeChannel(unsigned int uarfcn);

Description:

Change channel number of the instrument

Parameter:

Table 6-60 RCTLIB WCDMA APC ChangeChannel parameter

Parameter	Direction	Parameter
uarfcn	IN	Channel number

6.3.3.11 RCTLIB_WCDMA_APC_ChangeExpectedPower

Definition:

int __stdcall RCTLIB_WCDMA_APC_ChangeExpectedPower(int expectedPower);

Description:

Change expected power of the instrument

Parameter:

Table 6-61 RCTLIB WCDMA APC ChangeExpectedPower parameter

Parameter	Direction	Parameter
expectedPower	IN	Expected power (dB)

6.3.3.12 RCTLIB_WCDMA_APC_Initiate

Definition:

int __stdcall RCTLIB_WCDMA_APC_Initiate(void);

Description:

Initiate TX power measurement of the instrument



Parameter:

N/A

6.3.3.13 RCTLIB_WCDMA_APC_FetchResult

Definition:

int __stdcall RCTLIB_WCDMA_APC_FetchResult(double *outputPower);

Description:

Fetch TX power measurement result from the instrument

Parameter:

Table 6-62 RCTLIB WCDMA APC FetchResult parameter

Parameter	Direction	Parameter
outputPower	IN/OUT	A pointer pointing to a space for storing the result of TX power measurement.

6.3.3.14 RCTLIB_WCDMA_FHC_PreSetting

Definition:

int __stdcall RCTLIB_WCDMA_FHC_PreSetting(const S_RCTLIB_WCDMA_FHC_PRESETTINGS_T *pSettings);

Description:

Fetch TX power measurement result from the instrument

Parameter:

Table 6-63 RCTLIB WCDMA FHC presetting parameter

Parameter	Direction	Parameter
outputPower	IN/OUT	A pointer pointing to a space for storing the result of TX power measurement.

6.3.3.15 RCTLIB_WCDMA_FHC_StartIteration

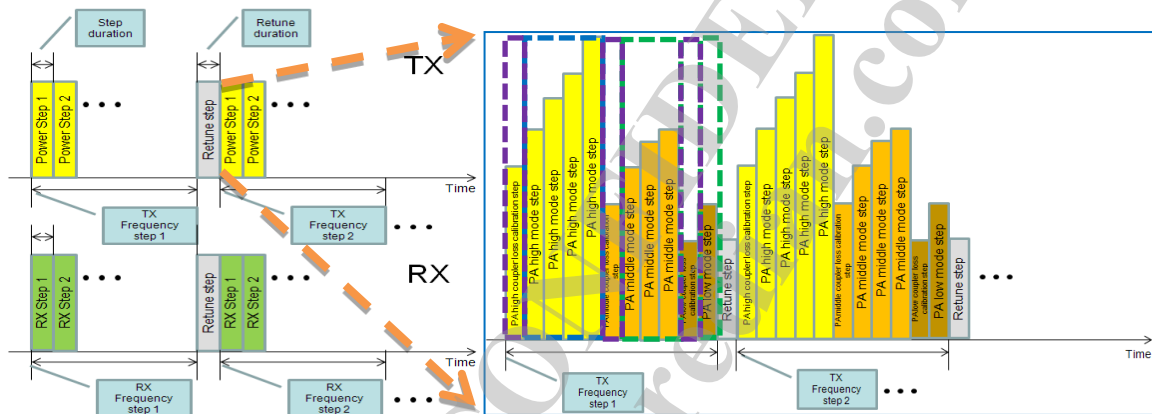
Definition:

int __stdcall RCTLIB_WCDMA_FHC_StartIteration(const S_RCTLIB_WCDMA_FHC_MEASUREMENT_PARAM_T *pSettings);

Description:

Fetch TX power measurement result from the instrument

- The calibration plan is composed of TX/RX frequency steps and TX/RX power steps
 - number of frequency steps * power steps
- The number of TX frequency steps and RX frequency steps must be the same. Each RX step does RSSI measurement and checks the LNA mode tracking result.



Parameter:

Table 6-64 RCTLIB_WCDMA_FHC StartIteration parameter

Parameter	Direction	Parameter
outputPower	IN/OUT	A pointer pointing to a space for storing the result of TX power measurement.

6.3.3.16 RCTLIB_WCDMA_FHC_FetchResult

Definition:

```
int __stdcall RCTLIB_WCDMA_FHC_FetchResult(const S_RCTLIB_WCDMA_FHC_MEASUREMENT_PARAM_T
*pSettings, S_RCTLIB_WCDMA_FHC_MEASUREMENT_RESULT_T* pResult);
```

Description:

Fetch TX power measurement result from the instrument

Parameter:

Table 6-65 RCTLIB_WCDMA_FHC_FetchResult parameter



Parameter	Direction	Parameter
outputPower	IN/OUT	A pointer pointing to a space for storing the result of TX power measurement.

6.3.3.17 RCTLIB_WCDMA_NSFT_PreSetting

Definition:

```
int __stdcall RCTLIB_WCDMA_NSFT_PreSetting(const S_RCTLIB_WCDMA_NSFT_PRESETTINGS_T* pSettings);
```

Description:

Config WCDMA application format, FDD operating mode, RMC12.2k, power measurement count (5 times), SEM measurement count (5 times), ACLR measurement count (5 times), WWQ measurement count (5 times), measurement time out, BER measurement count, ILPC segments upper and lower limitation, slot, TFCI, midamble and downlink power levels.

Parameter:

Table 6-66 RCTLIB WCDMA NSFT presetting parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA NSFT presetting

Related Data Structure:

```
typedef struct
{
    /// measurement timeout value (ms)
    double timeout;
    /// cell power for UE NSFT sync
    double cellPower;
    /// BER test count;
    unsigned int ber_bit_count;
    /// TFCI
    unsigned int tfci;
    /// SC code
    unsigned int sc_code;
    /// OVSF
    unsigned int ovsf;
    /// downlink data pattern
    unsigned int dtch_data_type;
    /// CPICH power (dB)
    double cpichPower;
    /// PICH power (dB)
    double pichPower;
    /// PCCPCH power (dB)
    double pccpchPower;
    /// DPCH power (dB)
    double dpchPower;
    /// ILPC config
    S_RCTLIB_WCDMA_NSFT_ILPC_CONFIG_T ilpcConfig;
}S_RCTLIB_WCDMA_NSFT_PRESETTINGS_T;
```

```
typedef struct
{
    char seg;
    double start_power;
    double stop_power;
    // single step error
    double single_step_error_algo1_1dB_upper[2]; // TPC cmd: +1, -1
    double single_step_error_algo1_1dB_lower[2]; // TPC cmd: +1, -1
    double single_step_error_algo1_2dB_upper[2]; // TPC cmd: +1, -1
    double single_step_error_algo1_2dB_lower[2]; // TPC cmd: +1, -1
    double single_step_error_algo2_upper[3]; // +1, 0, -1
    double single_step_error_algo2_lower[3]; // +1, 0, -1
    // aggregate step error (10 steps)
    double aggregate_step_error_algo1_1dB_upper[2]; // +1, -1
    double aggregate_step_error_algo1_1dB_lower[2]; // +1, -1
    double aggregate_step_error_algo1_2dB_upper[2]; // +1, -1
    double aggregate_step_error_algo1_2dB_lower[2]; // +1, -1
    double aggregate_step_error_algo2_upper[3]; // +1, 0, -1
    double aggregate_step_error_algo2_lower[3]; // +1, 0, -1
}S_RCTLIB_WCDMA_NSFT_ILPC_CONFIG_T;
```

6.3.3.18 RCTLIB_WCDMA_NSFT_InitiateTestCase

Definition:

int __stdcall RCTLIB_WCDMA_NSFT_InitiateTestCase(const S_RCTLIB_WCDMA_NSFT_CONFIG_T* pSettings);

Description:

Config cableloss, uplink channel, downlink channel, and change cell power.

Parameter:

Table 6-67 RCTLIB WCDMA NSFT InitiateTestCase parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA NSFT test case config

Related Data Structure:

```
typedef struct
{
    /// uplink UARFCN
    unsigned int uarfcn_ul;
    /// downlink UARFCN
    unsigned int uarfcn_dl;
    /// uplink cable loss
    double cableloss_ul;
    /// downlink cable loss
    double cableloss_dl;
    /// init cell power
    double cell_power;
    /// ILPC Test segment
    unsigned char testSegment;
}S_RCTLIB_WCDMA_NSFT_CONFIG_T;
```

6.3.3.19 RCTLIB_WCDMA_NSFT_TPC

Definition:

```
int __stdcall RCTLIB_WCDMA_NSFT_TPC(const S_RCTLIB_WCDMA_NSFT_TPC_REQUEST_T *tpc_request);
```

Description:

Setup the power control mode, power control algorithm, power step length, and expected power.

Parameter:

Table 6-68 RCTLIB WCDMA NSFT TPC parameter

Parameter	Direction	Parameter
tpc_request	IN/OUT	The structure for storing WCDMA NSFT TPC control used in TX performance

Related Data Structure:



```
typedef struct
{
    /// specify the test pattern (0: All up for UE max power, 1: All down for UE min power, 2: active for specified power
    int i_Pattern;
    /// TPC algorithm (1: algo1 2: algo2)
    unsigned int u_Algorithm;
    /// TPC step size
    int i_Step;
    /// the target power if the i_Pattern is 2, expected power if the tpc pattern is 0 or 1 (all up or all down)
    double d_TargetPower;
}S_RCTLIB_WCDMA_NSFT_TPC_REQUEST_T;
```

6.3.3.20 RCTLIB_WCDMA_NSFT_FetchTXPerformanceMaxPower

Definition:

```
int __stdcall
RCTLIB_WCDMA_NSFT_FetchTXPerformanceMaxPower(S_RCTLIB_WCDMA_NSFT_TX_PERF_MAX_POWER_RES
ULT_T *tx_perf_result);
```

Description:

Fetch NSFT TX performance at Max Power

Parameter:

Table 6-69 RCTLIB WCDMA NSFT FetchTXPerformanceMaxPower parameter

Parameter	Direction	Parameter
tx_perf_result	IN/OUT	The structure for storing WCDMA NSFT TX performance (max power) measurement result

Related Data Structure:

Mediatek Confidential
© 2017 Mediatek Inc.
This document contains information that is proprietary to Mediatek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
Classification: Confidential B

```
typedef struct
{
    /// UE power
    double m_dUEPower;

    /// SEM test result (emission mask)
    S_RCTLIB_WCDMA_NSFT_SEM_RESULT_T semResult;

    /// WWQ test result
    S_RCTLIB_WCDMA_NSFT_WWQ_RESULT_T wwqResult;

    /// spectrum test result (ACLR, OBW)
    S_RCTLIB_WCDMA_NSFT_SPECTRUM_RESULT_T spectrumResult;
}S_RCTLIB_WCDMA_NSFT_TX_PERF_MAX_POWER_RESULT_T;
```

/// The structure for storing WCDMA NSFT SEM result

```
typedef struct
{
    /// pass flag (overall)
    bool bPass;

    /// SEM level results
    double d_sem_level[8];

    /// frequency offset for each SEM level
    double d_sem_freq_offset[8];

    /// margin for each SEM result
    double d_sem_mask_margin[8];

    /// SEM pass flag
    int i_sem_pass[8];
}S_RCTLIB_WCDMA_NSFT_SEM_RESULT_T;
```

```
/// The structure for storing WCDMA NSFT WWQ result
```

```
typedef struct
```

```
{
```

```
    /// EVM (average, dB)
```

```
    double d_evm_rms_average;
```

```
    /// max frequency error (max, Hz)
```

```
    double d_evm_max_freq_error;
```

```
    /// OOS (average, dB)
```

```
    double d_evm_origin_offset;
```

```
    /// PCDE (max, dB)
```

```
    double d_pcde;
```

```
}S_RCTLIB_WCDMA_NSFT_WWQ_RESULT_T;
```

```
/// the structure for storing WCDMA NSFT spectrum measurement result
```

```
typedef struct
```

```
{
```

```
    /// ACLR at negative 10 MHz (dB)
```

```
    double m_dOffsetn10;
```

```
    /// ACLR at negative 5 MHz (dB)
```

```
    double m_dOffsetn5;
```

```
    /// ACLR at positive 5 MHz (dB)
```

```
    double m_dOffsetp5;
```

```
    /// ACLR at positive 10 MHz (dB)
```

```
    double m_dOffsetp10;
```

```
    /// Occupied bandwidth (MHz)
```

```
    double m_dOBW;
```

```
}S_RCTLIB_WCDMA_NSFT_SPECTRUM_RESULT_T;
```




6.3.3.21 RCTLIB_WCDMA_NSFT_FetchTXPerformanceMinPower

Definition:

int __stdcall
RCTLIB_WCDMA_NSFT_FetchTXPerformanceMinPower(S_RCTLIB_WCDMA_NSFT_TX_PERF_MIN_POWER_RES
ULT_T *tx_perf_result);

Description:

Fetch NSFT TX performance at Min Power

Parameter:

Table 6-70 RCTLIB WCDMA NSFT FetchTXPerformanceMinPower parameter

Parameter	Direction	Parameter
tx_perf_result	IN/OUT	the structure for storing WCDMA NSFT TX performance (min power) measurement result

Related Data Structure:

```
typedef struct  
{  
    /// UE power  
    double m_dUEPower;  
}S_RCTLIB_WCDMA_NSFT_TX_PERF_MIN_POWER_RESULT_T;
```

6.3.3.22 RCTLIB_WCDMA_NSFT_InitiateILPCTestCase

Definition:

int __stdcall RCTLIB_WCDMA_NSFT_InitiateILPCTestCase(unsigned char testSegment);

Description:

Config the ILPC in WCDMA NSFT segment and ready for measurement ILPC

Parameter:

Table 6-71 RCTLIB WCDMA NSFT InitiateILPCTestCase parameter



Parameter	Direction	Parameter
testSegment	IN	ILPC test segment

6.3.3.23 RCTLIB_WCDMA_NSFT_FetchILPCResult

Definition:

```
int __stdcall RCTLIB_WCDMA_NSFT_FetchILPCResult(S_RCTLIB_WCDMA_NSFT_ILPC_CONFIG_T*
ilpcConfig,S_RCTLIB_WCDMA_NSFT_ILPC_RESULT_T* result);
```

Description:

This function fetch the ILPC test result in WCDMA NSFT

Parameter:

Table 6-72 RCTLIB WCDMA NSFT FetchILPCResult parameter

Parameter	Direction	Parameter
ilpcConfig	IN	ILPC test segment
result	IN/OUT	The structure for storing WCDMA NSFT ILPC result

Related Data Structure:

```
typedef struct
{
    /// measurement timeout value (ms)
    double timeout;

    /// cell power for UE NSFT sync
    double cellPower;

    /// BER test count;
    unsigned int ber_bit_count;

    /// TFCI
    unsigned int tfci;

    /// SC code
    unsigned int sc_code;

    /// OVSF
    unsigned int ovfs;

    /// downlink data pattern
    unsigned int dtch_data_type;

    /// CPICH power (dB)
    double cpichPower;

    /// PICH power (dB)
    double pichPower;

    /// PCCPCH power (dB)
    double pccpchPower;

    /// DPCH power (dB)
    double dpchPower;

    /// ILPC config
    S_RCTLIB_WCDMA_NSFT_ILPC_CONFIG_T ilpcConfig;
}S_RCTLIB_WCDMA_NSFT_PRESETTINGS_T;
```

```
*
*/
typedef struct
{
    /// integrity
    int i_Integrity;
    /// pass flag (overall)
    bool bPass;
    /// number of slots
    int i_NumSlots;
    /// power (ABS)
    double d_Absolute[100];
    /// delta value between each step
    double d_Delta[100];
    bool bCal_fail;
}S_RCTLIB_WCDMA_NSFT_ILPC_RESULT_T;
```

6.3.3.24 RCTLIB_WCDMA_NSFT_ChangeCellPower

Definition:

int __stdcall RCTLIB_WCDMA_NSFT_ChangeCellPower(double d_cell_power);

Description:

This function change the downlink cell power in WCDMA NSFT

Parameter:

Table 6-73 RCTLIB WCDMA NSFT ChangeCellPower parameter

Parameter	Direction	Parameter
d_cell_power	IN	Cell power (dBm)

6.3.3.25 RCTLIB_WCDMA_NSFT_InitiateBER

Definition:

```
int __stdcall RCTLIB_WCDMA_HSDPA_NSFT_PreSetting(const int __stdcall RCTLIB_WCDMA_NSFT_InitiateBER());
```

Description:

This function initiate the loopback BER in WCDMA NSFT

Parameter:

N/A

6.3.3.26 RCTLIB_WCDMA_NSFT_ReadBER

Definition:

```
int __stdcall RCTLIB_WCDMA_NSFT_ReadBER(S_RCTLIB_WCDMA_NSFT_BER_RESULT_T* result);
```

Description:

This function fetch the loopback BER in WCDMA NSFT

Parameter:

Table 6-74 RCTLIB WCDMA NSFT ReadBER parameter

Parameter	Direction	Parameter
result	IN/OUT	The structure for storing WCDMA NSFT BER result

Related Data Structure:



```
typedef struct
{
    /// BER
    double d_ber;
    /// bit error count
    int i_err_cnt;
    /// total test count
    int i_total_cnt;
}S_RCTLIB_WCDMA_NSFT_BER_RESULT_T;
```

6.3.3.27 RCTLIB_WCDMA_NSFT_PRACH_PreSetting

Definition:

int `__stdcall` RCTLIB_WCDMA_NSFT_PRACH_PreSetting(const S_RCTLIB_WCDMA_NSFT_PRACH_PRESETTING_T* pSettings);

Description:

This function do the PRACH pre-setting in WCDMA NSFT

Parameter:

Table 6-75 RCTLIB WCDMA NSFT PRACH presetting parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA NSFT PRACH presetting

Related Data Structure:

MediaTek Confidential
© 2017 MediaTek Inc.
This document contains information that is proprietary to MediaTek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
Classification: Confidential B

```
typedef struct
{
    /// timeout (ms)
    double timeout;

    /// PRACH Test Uplink Interference (dBm)
    double uplinkInterference;

    /// PRACH Test Primary CPICH Power (dBm)
    double pcPichPower;

    /// PRACH Test Constant Value
    double constantValue;

    /// PRACH Test CPICH Level offset (dB)
    double cpichLevelOffset;
}S_RCTLIB_WCDMA_NSFT_PRACH_PRESETTING_T;
```

6.3.3.28 RCTLIB_WCDMA_NSFT_PRACH_CasePreSetting

Definition:

```
int __stdcall RCTLIB_WCDMA_NSFT_PRACH_CasePreSetting();
```

Description:

This function do the PRACH test case pre-setting in WCDMA NSFT

Parameter:

N/A

6.3.3.29 RCTLIB_WCDMA_NSFT_PRACH_InitiateTestCase

Definition:

```
int __stdcall RCTLIB_WCDMA_NSFT_PRACH_InitiateTestCase(const
S_RCTLIB_WCDMA_NSFT_PRACH_CONFIG_T* pSettings);
```



Description:

This function initiate the PRACH test case in WCDMA NSFT

Parameter:

Table 6-76 RCTLIB WCDMA NSFT PRACH InitiateTestCase parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA NSFT PRACH test case config

Related Data Structure:

```
typedef struct
{
    /// uplink UARFCN
    unsigned int uarfcn_ul;
    /// downlink UARFCN
    unsigned int uarfcn_dl;
    /// uplink cable loss
    double cableloss_ul;
    /// downlink cable loss
    double cableloss_dl;
}S_RCTLIB_WCDMA_NSFT_PRACH_CONFIG_T;
```

6.3.3.30 RCTLIB_WCDMA_NSFT_PRACH_FetchTestCase

Definition:

```
int __stdcall RCTLIB_WCDMA_NSFT_PRACH_FetchTestCase(S_RCTLIB_WCDMA_NSFT_PRACH_RESULT_T
*result);
```

Description:

This function fetch the PRACH test case in WCDMA NSFT

Parameter:



Table 6-77 RCTLIB WCDMA NSFT PRACH FetchTestCase parameter

Parameter	Direction	Parameter
result	IN/OUT	The structure for storing WCDMA NSFT PRACH result

Related Data Structure:

```
typedef struct
{
    /// TX on power
    double d_on_power;

    /// TX power before TX on
    double d_before_power;

    /// TX power after TX on
    double d_after_power;

    /// error between the actual power and expected power
    double d_error;
}S_RCTLIB_WCDMA_NSFT_PRACH_RESULT_T;
```

6.3.3.31 RCTLIB_WCDMA_HSDPA_NSFT_PreSetting

Definition:

```
int __stdcall RCTLIB_WCDMA_HSDPA_NSFT_PreSetting(const S_RCTLIB_WCDMA_HSPA_NSFT_PRESETTINGS_T*
pSettings);
```

Description:

This function is similar with 6.3.3.17 RCTLIB_WCDMA_NSFT_PreSetting, but further config the power levels of HSDPA channels

Parameter:

Table 6-78 RCTLIB WCDMA HSDPA NSFT presetting parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA HSPA NSFT presetting



Related Data Structure:

```
typedef struct
{
    /// HSDSCH UE category
    unsigned int ueCategory;

    /// measurement timeout value (ms)
    double timeout;

    /// delta ACK
    int deltaAck;

    /// delta NACK
    int deltaNack;

    /// delta CQI
    int deltaCqi;

    /// CQI feedback cycle
    int cqiFeedbackCycle;

    /// CQI repetition factor
    int cqiRepetitionFactor;

    int ovsf256;

    int ovsf128;

    int eagchChannelizationCode;
    int ehichChannelizationCode;
    int firstHspdschChannelizationCode;
    int hsscch1ChannelizationCode;
    double hspaFddTestCpichLevel;
    double hspaFddTestPccpchLevel;
    double hspaFddTestPichLevel;
    double hspaFddTestDpchLevel;
    double hspaFddTestEagchLevel;
    double hspaFddTestEhichLevel;
    double hspaFddTestHspdschsLevel;
```



```
double hspaFddTestHsscch1Level;

/// R99 parameters

S_RCTLIB_WCDMA_NSFT_PRESETTINGS_T r99Param;

}S_RCTLIB_WCDMA_HSPA_NSFT_PRESETTINGS_T;
```

6.3.3.32 RCTLIB_WCDMA_HSDPA_NSFT_InitiateTestCase

Definition:

int __stdcall RCTLIB_WCDMA_HSDPA_NSFT_InitiateTestCase(const S_RCTLIB_WCDMA_HSPA_NSFT_CONFIG_T* pConfig);

Description:

Config HSDPA uplink channel, downlink channel, power control mode, power control algorithm, power control gain of DPCCCH Bc and Bd, and expected power.

Parameter:

Table 6-79 RCTLIB WCDMA HSDPA NSFT InitiateTestCase parameter

Parameter	Direction	Parameter
pConfig	IN	The structure of WCDMA HSPA NSFT test case configuration

Related Data Structure:

```
typedef struct
{
    /// uplink test channel
    unsigned int uplinkUarfcn;
    /// uplink test channel
    unsigned int downlinkUarfcn;
    /// input(DUT uplink) cable loss
    double inputCableLoss;
    /// output(DUT downlink) cable loss;
    double outputCableLoss;
    /// HSPA sub test number: (1 ~ 4)
    unsigned int subTest;
    /// betaC
    unsigned int betaC;
    /// betaD
    unsigned int betaD;
    /// closed-loop target power for TPC (dBm)
    int closedLoopTargetPower;
    /// expected power for RF analyzer (dBm)
    int expectedPower;
    /// callback function for DUT control loop (for adding delay)
    DutWaitingCallback delayCallback;
}S_RCTLIB_WCDMA_HSPA_NSFT_CONFIG_T;

typedef void (__stdcall *DutWaitingCallback)(void);
```

6.3.3.3 RCTLIB_WCDMA_HSDPA_NSFT_FetchResult

Definition:

```
int __stdcall RCTLIB_WCDMA_HSDPA_NSFT_FetchResult(S_RCTLIB_WCDMA_HSPA_NSFT_RESULT_T* pResult);
```



Description:

Fetch WCDMA HSPA NSFT results of TX maximum power, the spectrum emissions mask measurement, and adjacent channel leakage ratio (ACLR) measurement

Parameter:

Table 6-80 RCTLIB WCDMA HSDPA NSFT FetchResult parameter

Parameter	Direction	Parameter
pResult	IN/OUT	The structure for storing WCDMA HSPA NSFT Test result

Related Data Structure:

```
typedef struct
{
    /// max power
    double maxPower;

    /// SEM
    S_RCTLIB_WCDMA_NSFT_SEM_RESULT_T semResult;

    /// ACLR
    S_RCTLIB_WCDMA_NSFT_SPECTRUM_RESULT_T aclrResult;
}S_RCTLIB_WCDMA_HSPA_NSFT_RESULT_T;
```

6.3.3.34 RCTLIB_WCDMA_HSUPA_NSFT_PreSetting

Definition:

```
int __stdcall RCTLIB_WCDMA_HSUPA_NSFT_PreSetting(const
S_RCTLIB_WCDMA_HSPA_NSFT_PRESETTINGS_T* pSettings);
```

Description:

Same with 6.3.3.31 RCTLIB_WCDMA_HSDPA_NSFT_PreSetting

**Parameter:****Table 6-81 RCTLIB WCDMA HSUPA NSFT presetting parameter**

Parameter	Direction	Parameter
pSettings	IN	The structure of WCDMA HSPA NSFT presetting

Related Data Structure:

Same with 6.3.3.31 RCTLIB_WCDMA_HSDPA_NSFT_PreSetting

6.3.3.35 RCTLIB_WCDMA_HSUPA_NSFT_InitiateTestCase**Definition:**

```
int __stdcall RCTLIB_WCDMA_HSUPA_NSFT_InitiateTestCase(const S_RCTLIB_WCDMA_HSPA_NSFT_CONFIG_T* pConfig);
```

Description:

Same with 6.3.3.35 RCTLIB_WCDMA_HSUPA_NSFT_InitiateTestCase

Parameter:**Table 6-82 RCTLIB WCDMA HSUPA NSFT InitiateTestCase parameter**

Parameter	Direction	Parameter
pConfig	IN	The structure of WCDMA HSPA NSFT test case configuration

Related Data Structure:

Same with 6.3.3.35 RCTLIB_WCDMA_HSUPA_NSFT_InitiateTestCase

6.3.3.36 RCTLIB_WCDMA_HSUPA_NSFT_FetchResult**Definition:**

```
int __stdcall RCTLIB_WCDMA_HSUPA_NSFT_FetchResult(S_RCTLIB_WCDMA_HSPA_NSFT_RESULT_T* pResult);
```

Description:

Same with 6.3.3.33 RCTLIB_WCDMA_HSDPA_NSFT_FetchResult

Parameter:

Table 6-83 RCTLIB WCDMA HSUPA NSFT FetchResult parameter

Parameter	Direction	Parameter
pResult	IN/OUT	The structure for storing WCDMA HSPA NSFT test result

Related Data Structure:

Same with 6.3.3.33 RCTLIB_WCDMA_HSDPA_NSFT_FetchResult

6.3.3.37 RCTLIB_WCDMA_GET_SpecificSettings

Definition:

```
int __stdcall RCTLIB_WCDMA_GET_SpecificSettings(S_RCTLIB_WCDMA_SPECIFIC_SETTINGS_T *pSettings);
```

Description:

Get the special constant settings from instrument. e.g. ctfc and tfci.

Parameter:

Table 6-84 RCTLIB WCDMA GET SpecificSettings parameter

Parameter	Direction	Parameter
pSettings	IN/OUT	the structure to get WCDMA instrument's special settings

Related Data Structure:

```
typedef struct
{
    /// WCDMA NSFT CTFC
    unsigned char ucNSFT_ctfc[4];
    /// WCDMA NSFT TFci
    unsigned short usNSFT_tfci;
}S_RCTLIB_WCDMA_SPECIFIC_SETTINGS_T;
```




6.3.4 TDSCDMA Exported Instrument Control functions

6.3.4.1 RCTLIB_TDSCDMA_ConfigTesterCommonBeforeCal

Definition:

int __stdcall RCTLIB_TDSCDMA_ConfigTesterCommonBeforeCal(void);

Description:

Config the common settings of TDSCDMA of the instrument . Currently, this function is a dummy function which always return success.

Parameter:

N/A

6.3.4.2 RCTLIB_TDSCDMA_ConfigTesterBeforeAFC

Definition:

int __stdcall RCTLIB_TDSCDMA_ConfigTesterBeforeAFC(const S_RCTLIB_TDSCDMA_AFC_TESTER_CONFIG_BEFORE_CAL_T *p_rConfig);

Description:

Get the special constant settings from instrument. e.g. ctfc and tfci.

Parameter:

Table 6-85 RCTLIB TDSCDMA ConfigTesterBeforeAFC parameter

Parameter	Direction	Parameter
p_rConfig	IN	the structure of TDSCDMA AFC configures

Related Data Structure:

Mediatek Confidential
This document contains information that is proprietary to Mediatek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
© 2017 Mediatek Inc.
Classification: Confidential B



```
typedef struct
{
    /// cell power seetings
    double    d_cell_power;

    /// expected power
    int       expected_power;

    /// Average Count
    unsigned int  ui_average_cnt;

    /// UARFCN
    unsigned short us_uarfcn;
}S_RCTLIB_TDSCDMA_AFC_TESTER_CONFIG_BEFORE_CAL_T;
```

6.3.4.3 RCTLIB_TDSCDMA_MeasureAFC

Definition:

int __stdcall RCTLIB_TDSCDMA_MeasureAFC(double *p_dFreqOffset);

Description:

Get the special constant settings from instrument. e.g. ctfc and tfci.

Parameter:

Table 6-86 RCTLIB TDSCDMA MeasureAFC parameter

Parameter	Direction	Parameter
p_dFreqOffset	IN/OUT	A pointer pointing to a space for storing the result of frequency error measurement.

6.3.4.4 RCTLIB_TDSCDMA_ConfigTesterBeforeAGC

Definition:

int __stdcall RCTLIB_TDSCDMA_ConfigTesterBeforeAGC(const S_RCTLIB_TDSCDMA_AGC_TESTER_CONFIG_BEFORE_CAL_T *p_rConfig);

Mediatek Confidential
This document contains information that is proprietary to Mediatek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
© 2017 Mediatek Inc.
Classification: Confidential B

**Description:**

Get the special constant settings from instrument. e.g. ctfc and tfci.

Parameter:

Table 6-87 RCTLIB TDSCDMA ConfigTesterBeforeAGC parameter

Parameter	Direction	Parameter
p_rConfig	IN	the structure of TDSCDMA AGC configures

Related Data Structure:

```
typedef struct
{
    /// cell power for the calibration (dBm)
    double d_cell_power;
}S_RCTLIB_TDSCDMA_AGC_TESTER_CONFIG_BEFORE_CAL_T;
```

6.3.4.5 RCTLIB_TDSCDMA_AGC_ChangeCellPower

Definition:

```
int __stdcall RCTLIB_TDSCDMA_AGC_ChangeCellPower(double cell_power);
```

Description:

Change cell power of the instrument

Parameter:

Table 6-88 RCTLIB TDSCDMA AGC ChangeCellPower parameter

Parameter	Direction	Parameter
cell_power	IN	Cell power (dBm)



6.3.4.6 RCTLIB_TDSCDMA_AGC_ChangeCellBand

Definition:

```
int __stdcall RCTLIB_TDSCDMA_AGC_ChangeCellBand(unsigned int band);
```

Description:

Change band setting of the instrument

Parameter:

Table 6-89 RCTLIB TDSCDMA AGC ChangeCellBand parameter

Parameter	Direction	Parameter
band	IN	Band indicator

6.3.4.7 RCTLIB_TDSCDMA_AGC_ChangeChannel

Definition:

```
int __stdcall RCTLIB_TDSCDMA_AGC_ChangeChannel(unsigned int uarfcn);
```

Description:

Change channel number of the instrument

Parameter:

Table 6-90 RCTLIB TDSCDMA AGC ChangeChannel parameter

Parameter	Direction	Parameter
uarfcn	IN	Channel number

6.3.4.8 RCTLIB_TDSCDMA_ConfigTesterBeforeAPC

Definition:

```
int __stdcall RCTLIB_TDSCDMA_ConfigTesterBeforeAPC(const  
S_RCTLIB_TDSCDMA_APC_TESTER_CONFIG_BEFORE_CAL_T *p_rConfig);
```

Description:



Get the special constant settings from instrument. e.g. ctfc and tfci.

Parameter:

Table 6-91 RCTLIB TDSCDMA ConfigTesterBeforeAPC parameter

Parameter	Direction	Parameter
p_rConfig	IN	the structure of TDSCDMA APC configures

Related Data Structure:

```
typedef struct
{
    /// timeout value (unit: ms)
    double timeout;
    /// meausurement count
    int measurement_count;
}S_RCTLIB_TDSCDMA_APC_TESTER_CONFIG_BEFORE_CAL_T;
```

6.3.4.9 RCTLIB_TDSCDMA_APC_ChangeCellBand

Definition:

```
int __stdcall RCTLIB_TDSCDMA_APC_ChangeCellBand(unsigned int band);
```

Description:

Change band setting of the instrument

Parameter:

Table 6-92 RCTLIB TDSCDMA APC ChangeCellBand parameter

Parameter	Direction	Parameter
band	IN	Band indicator



6.3.4.10 RCTLIB_TDSCDMA_APC_MeasurePower

Definition:

```
int __stdcall RCTLIB_TDSCDMA_APC_MeasurePower(S_RCTLIB_TDSCDMA_APC_MEASUREMENT_PARAM_T
*param);
```

Description:

N/A

Parameter:

Table 6-93 RCTLIB TDSCDMA APC MeasurePower parameter

Parameter	Direction	Parameter
param	IN/OUT	the structure for configing and storing the TX Power measurement result

Related Data Structure:

```
typedef struct
{
    /// UARFCN
    unsigned int uarfcn;
    /// expected power (dBm)
    int expected_power;
    /// [IN/OUT] channel power (dBm)
    double output_power;
}S_RCTLIB_TDSCDMA_APC_MEASUREMENT_PARAM_T;
```

6.3.4.11 RCTLIB_TDSCDMA_ConfigTesterBeforeFHC

Definition:

```
int __stdcall RCTLIB_TDSCDMA_ConfigTesterBeforeFHC(const
S_RCTLIB_TDSCDMA_FHC_TESTER_CONFIG_BEFORE_CAL_T *p_rConfig);
```

Description:



N/A

Parameter:**Table 6-94 RCTLIB TDSCDMA ConfigTesterBeforeFHC parameter**

Parameter	Direction	Parameter
p_rConfig	IN	the structure of TDSCDMA FHC presetting

Related Data Structure:

```
typedef struct
{
    /// timeout value (unit: ms)
    double timeout;
}S_RCTLIB_TDSCDMA_FHC_TESTER_CONFIG_BEFORE_CAL_T;
```

6.3.4.12 RCTLIB_TDSCDMA_FHC_StartIteration**Definition:**

```
int __stdcall RCTLIB_TDSCDMA_FHC_StartIteration(const S_RCTLIB_TDSCDMA_FHC_MEASUREMENT_PARAM_T
*pSettings);
```

Description:

N/A

Parameter:**Table 6-95 RCTLIB TDSCDMA FHC StartIteration parameter**

Parameter	Direction	Parameter
pSettings	IN	the structure of TDSCDMA FHC configures

Related Data Structure:

```
typedef struct
{
    /// number of frequency steps (Max 20)
    unsigned int freq_num;
    /// frequency steps (used in both TX/RX since it's TDD) (MHz)
    double freq_steps[40];
    /// number of tx power steps
    unsigned int tx_power_steps;
    /// expected power level of tx power step (Max 40)
    double tx_expected_power[40];
    /// number of rx power steps
    unsigned int rx_power_steps;
    /// power level of rx power step (Max 40)
    double rx_power_level[40];
    /// RX retune step length (unit sub-frame)
    unsigned int rx_retune_step;
    /// TX retune step length (unit sub-frame)
    unsigned int tx_retune_step;
    /// power step length (sub-frame)
    unsigned int power_step_length;
}S_RCTLIB_TDSCDMA_FHC_MEASUREMENT_PARAM_T;
```

6.3.4.13 RCTLIB_TDSCDMA_FHC_FetchResult

Definition:

```
int __stdcall RCTLIB_TDSCDMA_FHC_FetchResult(const S_RCTLIB_TDSCDMA_FHC_MEASUREMENT_PARAM_T
*pSettings, S_RCTLIB_TDSCDMA_FHC_MEASUREMENT_RESULT_T* pResult);
```

Description:

N/A



Parameter:

Table 6-96 RCTLIB_TDSCDMA_FHC_FetchResult parameter

Parameter	Direction	Parameter
pSettings	IN	The structure of TDSCDMA FHC configures
pResult	IN/OUT	The structure for storing the results of TDSCDMA FHC TX power measurement

Related Data Structure:

```
typedef struct
{
    int integrity;
    /// number of frequency steps
    int freq_steps;
    /// number of tx power steps
    int tx_power_steps;
    /// number of tx power samples
    int tx_power_samples;
    /// TX power measurement result
    double tx_power[400];
}S_RCTLIB_TDSCDMA_FHC_MEASUREMENT_RESULT_T;
```

6.3.4.14 RCTLIB_TDSCDMA_ConfigTesterCommonBeforeNSFT

Definition:

```
int __stdcall
RCTLIB_TDSCDMA_ConfigTesterCommonBeforeNSFT(S_RCTLIB_TD_NSFT_TESTER_COMMON_CONFIG_T
*common_cfg);
```

Description:

N/A



Parameter:

Table 6-97 RCTLIB TDSCDMA ConfigTesterCommonBeforeNSFT parameter

Parameter	Direction	Parameter
common_cfg	IN	The structure of TDSCDMA NSFT common configures

Related Data Structure:

```
typedef struct
{
    unsigned char sc_code;

    // PRBS9 (0) | PRBS15 (1) | INCRement (2) | ZERos (3) | ONEs (4) | ALTerating (5)

    int      m_iDtchType;
}S_RCTLIB_TD_NSFT_TESTER_COMMON_CONFIG_T;
```

6.3.4.15 RCTLIB_TDSCDMA_ConfigTesterForNSFT

Definition:

```
int __stdcall RCTLIB_TDSCDMA_ConfigTesterForNSFT(const S_RCTLIB_TD_NSFT_TESTER_CONFIG_T *cfg);
```

Description:

N/A

Parameter:

Table 6-98 RCTLIB TDSCDMA ConfigTesterForNSFT parameter

Parameter	Direction	Parameter
cfg	IN	The structure of TDSCDMA NSFT configures

Related Data Structure:

```
typedef struct
{
    /// UARFCN
    unsigned short m_usTD_NSFT_UARFCN;

    /// Cable loss
    double m_dTD_NSFT_CableLoss;

    /// Cell power (dBm)
    double m_dTD_NSFT_Cell_Power;

    /// Reserved
    double m_dTD_NSFT_MAX_PWR_AVG_HIGH;
}S_RCTLIB_TD_NSFT_TESTER_CONFIG_T;
```

6.3.4.16 RCTLIB_TDSCDMA_MeasureTPCForNSFT

Definition:

```
int __stdcall RCTLIB_TDSCDMA_MeasureTPCForNSFT(
    const S_RCTLIB_TD_NSFT_TPC_CONFIG_ENTRY_T *req,
    S_RCTLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T *res);
```

Description:

N/A

Parameter:

Table 6-99 RCTLIB TDSCDMA MeasureTPCForNSFT parameter

Parameter	Direction	Parameter
req	IN	The structure of TDSCDMA NSFT configures
res	IN/OUT	The structure for storing the results of TDSCDMA NSFT measurements

Related Data Structure:

```
typedef struct
{
    double d_min_avg_pwr;
    double d_max_avg_pwr;
    double d_obw_max;

    S_RCTLIB_TDSCDMA_NSFT_SEM_RESULT_T r_sem_result;
    S_TDSCDMA_NSFT_ACLR_RESULT_T r_aclr_result;
    S_TDSCDMA_NSFT_WQ_RESULT_T r_wq_result;
    S_TDSCDMA_NSFT_WQ_RESULT_T r_wq_result_at_low_power;
    S_TDSCDMA_NSFT_OOP_RESULT_T r_oop_result;
    S_TDSCDMA_NSFT_CLP_RESULT_T r_clp_result;

    double d_ACLR_Low5;
    double d_ACLR_Up5;
    double d_ACLR_Low10;
    double d_ACLR_Up10;

    double d_ber_from_target;
    double d_ber_from_tester; // the ber measured by tester
}S_RCTLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T;
```

```
typedef struct
{
    bool bPass;

    double d_sem_level[RCTLIB_TD_NSFT_SEM_MARGIN_NUM];

    double d_sem_freq_offset[RCTLIB_TD_NSFT_SEM_MARGIN_NUM];

    double d_sem_margin[RCTLIB_TD_NSFT_SEM_MARGIN_NUM];

    bool b_sem_pass[RCTLIB_TD_NSFT_SEM_MARGIN_NUM];

    double d_in_channel_pwr[RCTLIB_TD_NSFT_SEM_MARGIN_NUM];

}S_RCTLIB_TDSCDMA_NSFT_SEM_RESULT_T;
```

```
typedef struct
{
    bool bPass;

    double d_aclr_result[4]; // -1.6/1.6/-3.2/3.2 MHz Offset

    bool b_aclr_pass[4]; // -1.6/1.6/-3.2/3.2 MHz Offset

}S_TDSCDMA_NSFT_ACLR_RESULT_T;
```

```
typedef struct
{
    double d_evm_rms;

    double d_evm_max_freq_error;

    double d_evm_origin_offset;

    double d_max_pcde;

}S_TDSCDMA_NSFT_WQ_RESULT_T;
```



```
typedef struct
{
    bool bPass;

    double d_avg_on_pwr;

    bool b_oop_pass[3];

    double d_avg_pwr[3];

}S_TDSCDMA_NSFT_OOP_RESULT_T;
```

```
typedef struct
{
    bool bPass;

    double d_max_pwr;

    double d_min_pwr;

    int i_worst_step_index[2]; // [0]: REL1POW, [1]:REL10POW

    double d_worst_abs_pwr[2]; // [0]: REL1POW, [1]:REL10POW

    double d_worst_pwr[2]; // [0]: REL1POW, [1]:REL10POW

}S_TDSCDMA_NSFT_CLP_RESULT_T;
```

6.3.4.17 RCTLIB_TDSCDMA_ConfigTesterForNSFTBer

Definition:

```
int __stdcall RCTLIB_TDSCDMA_ConfigTesterForNSFTBer(const S_RCTLIB_TD_NSFT_BER_CONFIG_ENTRY_T
*cfg);
```

Description:

N/A

Parameter:

Table 6-100 RCTLIB TDSCDMA ConfigTesterForNSFTBer parameter



Parameter	Direction	Parameter
cfg	IN	The structure of TDSCDMA NSFT BER configures

Related Data Structure:

```
typedef struct
{
    double d_cell_pwr;

    // loopback BER
    bool b_loopback_ber;

    // valid when b_loopback_ber = true;
    int i_bit_cnt;
}S_RCTLIB_TD_NSFT_BER_CONFIG_ENTRY_T;
```

6.3.4.18 RCTLIB_TDSCDMA_MeasureLBerForNSFT**Definition:**

```
int __stdcall RCTLIB_TDSCDMA_MeasureLBerForNSFT(S_RCTLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T *res);
```

Description:

N/A

Parameter:**Table 6-101 RCTLIB TDSCDMA MeasureLBerForNSFT parameter**

Parameter	Direction	Parameter
res	IN/OUT	The structure for storing the results of TDSCDMA NSFT measurements

Related Data Structure:

Same with 6.3.4.16 RCTLIB_TDSCDMA_MeasureTPCForNSFT



6.3.4.19 RCTLIB_TDSCDMA_ConfigTesterCommonBeforeFT

Definition:

```
int  
RCTLIB_TDSCDMA_ConfigTesterCommonBeforeFT(S_RCTLIB_TD_NSFT_TESTER_COMMON_CONFIG_T  
*common_cfg);
```

Description:

N/A

Parameter:

Table 6-102 RCTLIB TDSCDMA ConfigTesterCommonBeforeFT parameter

Parameter	Direction	Parameter
common_cfg	IN	The structure of TDSCDMA signaling mode common configures

Related Data Structure:

Same with 6.3.4.14 RCTLIB_TDSCDMA_ConfigTesterCommonBeforeNSFT

6.3.4.20 RCTLIB_TDSCDMA_ConfigTesterForFT

Definition:

```
int __stdcall RCTLIB_TDSCDMA_ConfigTesterForFT(const S_RCTLIB_TD_NSFT_TESTER_CONFIG_T *cfg);
```

Description:

N/A

Parameter:

Table 6-103 RCTLIB TDSCDMA ConfigTesterForFT parameter

Parameter	Direction	Parameter
cfg	IN	The structure of TDSCDMA signaling mode configures

Related Data Structure:

MediaTek Confidential
© 2017 MediaTek Inc.
This document contains information that is proprietary to MediaTek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
Classification: Confidential B



Same with 6.3.4.15 RCTLIB_TDSCDMA_ConfigTesterForNSFT

6.3.4.21 RCTLIB_TDSCDMA_MeasureTPCForFT

Definition:

```
int __stdcall RCTLIB_TDSCDMA_MeasureTPCForFT(const S_RCTLIB_TD_NSFT_TPC_CONFIG_ENTRY_T *req,  
                                              S_RCTLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T *res);
```

Description:

N/A

Parameter:

Table 6-104 RCTLIB TDSCDMA MeasureTPCForFT parameter

Parameter	Direction	Parameter
req	IN	The structure of TDSCDMA signaling mode configures
res	IN/OUT	The structure for storing the results of TDSCDMA signaling mode measurements

Related Data Structure:

Same with 6.3.4.16 RCTLIB_TDSCDMA_MeasureTPCForNSFT

6.3.4.22 RCTLIB_TDSCDMA_ConfigTesterForFTBer

Definition:

```
int __stdcall RCTLIB_TDSCDMA_ConfigTesterForFTBer(const S_RCTLIB_TD_NSFT_BER_CONFIG_ENTRY_T *cfg);
```

Description:

N/A

Parameter:

Table 6-105 RCTLIB TDSCDMA ConfigTesterForFTBer parameter



Parameter	Direction	Parameter
cfg	IN	The structure of TDSCDMA NSFT signaling mode configures

Related Data Structure:

Same with 6.3.4.17 RCTLIB_TDSCDMA_ConfigTesterForNSFTBer

6.3.4.23 RCTLIB_TDSCDMA_MeasureLBerForFT**Definition:**

```
int __stdcall RCTLIB_TDSCDMA_MeasureLBerForFT(S_RCTLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T *res);
```

Description:

N/A

Parameter:

Table 6-106 RCTLIB TDSCDMA MeasureLBerForFT parameter

Parameter	Direction	Parameter
res	IN/OUT	The structure for storing the results of TDSCDMA signaling mode measurements

Related Data Structure:

Same with 6.3.4.23 RCTLIB_TDSCDMA_MeasureLBerForFT

6.3.4.24 RCTLIB_TDSCDMA_GET_SpecificSettings**Definition:**

```
int __stdcall RCTLIB_TDSCDMA_GET_SpecificSettings(S_RCTLIB_TDSCDMA_SPECIFIC_SETTINGS_T *pSettings);
```

Description:

N/A

Parameter:



Table 6-107 RCTLIB TDSCDMA GET SpecificSettings parameter

Parameter	Direction	Parameter
pSettings	IN/OUT	the structure to get TDSCDMA instrument's special settings

Related Data Structure:

```
typedef struct
{
    /// Traditional calibration support
    bool bTCSupport;
    /// Traditional calibration support
    bool bFHCSupport;
    /// NSFT UL midamble config
    unsigned char usNSFT_midamble;
}S_RCTLIB_TDSCDMA_SPECIFIC_SETTINGS_T;
```

6.3.5 LTE Exported Instrument Control functions

6.3.5.1 RCTLIB_LTE_ConfigDefaultSettings

Definition:

```
int __stdcall RCTLIB_LTE_ConfigDefaultSettings(void);
```

Description:

Initialize the instrument.

Parameter:

N/A

Related Data Structure:

N/A



6.3.5.2 RCTLIB_LTE_AFC_PreSetting

Definition:

```
int __stdcall RCTLIB_LTE_AFC_PreSetting(const S_RCTLIB_LTE_AFC_PRESETTINGS_T* pSettings);
```

Description:

Preset the instrument to ready for meansuring single tone frequency error.

Parameter:

Table 6-108 RCTLIB LTE AFC presetting parameter

Parameter	Direction	Parameter
pSettings	IN	Frequency error measurement parameter structure

Related Data Structure:

```
typedef struct
{
    unsigned char band;
    unsigned short usFrequency; /* the frequency to siwtch. Unit: 100KHz */
    short sFreqOffset; /* the frequency to offset. Unit: KHz */
    double dExpectedPower; /* the expected power, dBm */
    char cMeasurementCount; /* measurement count */
} S_RCTLIB_LTE_AFC_PRESETTINGS_T;
```

6.3.5.3 RCTLIB_LTE_AFC_Initiate

Definition:

```
int __stdcall RCTLIB_LTE_AFC_Initiate(void);
```

Description:

Initial frequency error measurement function of the instrument.

Parameter:



N/A

Related Data Structure:

N/A

6.3.5.4 RCTLIB_LTE_AFC_FetchResult**Definition:**

```
int __stdcall RCTLIB_LTE_AFC_FetchResult(double* freqerror);
```

Description:

Fetch frequency error measurement result from the instrument in Hz unit.

Parameter:*Table 6-109 RCTLIB LTE AFC FetchResult parameter*

Parameter	Direction	Parameter
freqerror	IN	The double pointer to store the frequency error result

Related Data Structure:

N/A

6.3.5.5 RCTLIB_LTE_AGC_PreSetting**Definition:**

```
int __stdcall RCTLIB_LTE_AGC_PreSetting();
```

Description:

Preset the instrument to ready for generating single tone signal to calibrate DUT's pathlosses.

Parameter:

N/A

Related Data Structure:

N/A

6.3.5.6 RCTLIB_LTE_AGC_ChangeCellBand

Definition:

```
int __stdcall RCTLIB_LTE_AGC_ChangeCellBand(unsigned int band);
```

Description:

Command the instrument to set cableloss for specific band or frequency range.

Parameter:

Table 6-110 RCTLIB LTE AGC ChangeCellBand parameter

Parameter	Direction	Parameter
band	IN	The band index to set the user configured cableloss.

Related Data Structure:

N/A

6.3.5.7 RCTLIB_LTE_AGC_ChangeFrequency

Definition:

```
int __stdcall RCTLIB_LTE_AGC_ChangeFrequency(const S_RCTLIB_LTE_FREQUENCY_T* pSettings);
```

Description:

Command the instrument to change RX frequency. The RX path loss are calibrated by single tone signal, therefore using frequency offset to eliminate DC effect to improve RSSI measurement accuracy.

Parameter:

Table 6-111 RCTLIB LTE AGC ChangeFrequency parameter

Parameter	Direction	Parameter
pSettings	IN	The single tone frequency parameter.

Related Data Structure:

```
typedef struct
{
    unsigned short usFrequency; /* the frequency to siwtch. Unit: 100KHz */
    short sFreqOffset; /* the frequency to offset. Unit: KHz */
} S_RCTLIB_LTE_FREQUENCY_T;
```

6.3.5.8 RCTLIB_LTE_AGC_ChangeCellPower

Definition:

```
int __stdcall RCTLIB_LTE_AGC_ChangeCellPower(double cellPower);
```

Description:

Command the instrument to change downlink power.

Parameter:

Table 6-112 RCTLIB LTE AGC ChangeCellPower parameter

Parameter	Direction	Parameter
cellPower	IN	The downlink power to change to. (Unit: dBm)

Related Data Structure:

N/A

6.3.5.9 RCTLIB_LTE_APC_PreSetting

Definition:

```
int __stdcall RCTLIB_LTE_APC_PreSetting(const S_RCTLIB_LTE_APC_PRESETTINGS_T* pSettings);
```

Description:

Preset the instrument for single tone power measurement.

Parameter:

Table 6-113 RCTLIB LTE APC presetting parameter

Parameter	Direction	Parameter
pSettings	IN	The parameters of single tone power measurement.

Related Data Structure:

```
typedef struct
{
    char cMeasurementCount; /* measurement count */
    char cTriggerMode; /*0: rising trigger 1: free run */
    unsigned int uiTriggerDelay; /* trigger delay (Unit: ms) */
} S_RCTLIB_LTE_APC_PRESETTINGS_T;
```

6.3.5.10 RCTLIB_LTE_APC_ChangeCellBand

Definition:

```
int __stdcall RCTLIB_LTE_APC_ChangeCellBand(unsigned int band);
```

Description:

Command the instrument to set cableloss for specific band or frequency range.

Parameter:

Table 6-114 RCTLIB LTE APC ChangeCellBand parameter

Parameter	Direction	Parameter
band	IN	To set the cable loss for specific band index.

Related Data Structure:

N/A

6.3.5.11 RCTLIB_LTE_APC_ChangeUIFrequency

Definition:

```
int __stdcall RCTLIB_LTE_APC_ChangeUIFrequency(const S_RCTLIB_LTE_FREQUENCY_T* pSettings);
```


Description:

Command the instrument to change frequency for single tone power measurement. Using frequency offset to eliminate DC effect.

Parameter:

Table 6-115 RCTLIB LTE APC ChangeUIFrequency parameter

Parameter	Direction	Parameter
pSettings	IN	The single tone frequency parameter.

Related Data Structure:

```
typedef struct
{
    unsigned short usFrequency; /* the frequency to switch. Unit: 100KHz */
    short sFreqOffset; /* the frequency to offset. Unit: KHz */
} S_RCTLIB_LTE_FREQUENCY_T;
```

6.3.5.12 RCTLIB_LTE_APC_ChangeExpectedPower

Definition:

```
int __stdcall RCTLIB_LTE_APC_ChangeExpectedPower(int expectedPower);
```

Description:

To set the expected power to the instrument.

Parameter:

Table 6-116 RCTLIB LTE APC ChangeExpectedPower parameter

Parameter	Direction	Parameter
expectedPower	IN	The expected power to set to the instrument. (Unit: dBm)

Related Data Structure:

N/A



6.3.5.13 RCTLIB_LTE_APC_Initiate

Definition:

```
int __stdcall RCTLIB_LTE_APC_Initiate(void);
```

Description:

Initial single tone power measurement function of the instrument, and ready for measurement.

Parameter:

N/A

Related Data Structure:

N/A

6.3.5.14 RCTLIB_LTE_APC_FetchResult

Definition:

```
int __stdcall RCTLIB_LTE_APC_FetchResult(double* outputPower);
```

Description:

Fetch single tone power measurement result from the instrument.

Parameter:

Table 6-117 RCTLI LTE APC FetchResult parameter

Parameter	Direction	Parameter
outputPower	OUT	The single tone power measurement result fetch from the instrument.

Related Data Structure:

N/A



6.3.5.15 RCTLIB_LTE_FHC_PreSetting

Definition:

```
int __stdcall RCTLIB_LTE_FHC_PreSetting(const S_RCTLIB_LTE_FHC_PRESETTINGS_T* pSettings);
```

Description:

Preset the instrument to ready for fast handset calibration mode.

Parameter:

Table 6-118 RCTLIB LTE FHC presetting parameter

Parameter	Direction	Parameter
pSettings	IN	Reserved for future using.

Related Data Structure:

```
typedef struct
{
    char dummy; /// reserved
} S_RCTLIB_LTE_FHC_PRESETTINGS_T;
```

6.3.5.16 RCTLIB_LTE_FHC_StartIteration

Definition:

```
int __stdcall RCTLIB_LTE_FHC_StartIteration(const S_RCTLIB_LTE_FHC_MEASUREMENT_PARAM_T* pSettings);
```

Description:

Setup the arrangement to the instruments. The band arrangement(S_RCTLIB_LTE_FHC_BAND_PARAM_T) can be split into two categories: FDD and TDD. In the FDD band, the TX single tone power measurement and RX downlink single tone are executing synchronously as shown in the **Error! Reference source not found..** Others, TDD band, are executing asynchronously due to frequency the same as shown in the **Error! Reference source not found..** Each band arrangement are composed of TX and RX frequency arrangements(S_RCTLIB_LTE_FHC_FREQ_STEP_U). In the RX frequency part, the arrangements tell the instrument when to generate the single tone down link power and the strength of the down link power(Unit:

Classification: Confidential B





Parameter:

Table 6-119 RCTLIB LTE FHC StartIteration parameter

Parameter	Direction	Parameter
pSettings	IN	The arrangements of TX single tone power measurement and RX downlink single tone power.

Related Data Structure:

The meaning of each parameters in the following structure can be found in the **Error! Reference source not found.** and **Error! Reference source not found.**

```
typedef struct
{
    unsigned short  usFrequency; /* the frequency to siwtch. Unit: 100KHz */
    short          sFreqOffset; /* the frequency to offset. Unit: KHz */
    double         dPowerSteps[27]; /// TX expected power level (dBm) or RX downlink power (dBm)
    unsigned char   ucPwrStepNum;
} S_RCTLIB_LTE_FHC_FREQ_STEP_U;

/**
 * The structure for storing LTE FHC request by band
 */

typedef struct
{
    /// band
    unsigned int   uiBand; // index start from 0 ==> band1
    /// 0:TDD, 1:FDD
    unsigned char  duplex_mode;
    /// number of frequency step
    int            iNumberOfFrequencySteps;
    /// UE TX frequency setting in each frequency step (KHz + offset)
    S_RCTLIB_LTE_FHC_FREQ_STEP_U sTxFreqSteps[22];
    /// UE RX frequency setting in each frequency step (KHz + offset)
    S_RCTLIB_LTE_FHC_FREQ_STEP_U sRxFreqSteps[22];
} S_RCTLIB_LTE_FHC_BAND_PARAM_T;

/**
 * The structure for storing LTE FHC request
 */

typedef struct
{

```

```

unsigned char fdd_tx_to_rx_time;//ms unit
unsigned char tdd_tx_to_rx_time;//ms unit
unsigned char freq_switch_time;//ms unit
unsigned char band_switch_time;//ms unit
unsigned char tx_step_width;//ms unit
unsigned char tdd_to_fdd_switch_time;//ms unit
unsigned char fdd_to_tdd_switch_time;//ms unit
unsigned char band_num;

/// TX + RX Band parameter

S_RCTLIB_LTE_FHC_BAND_PARAM_T sTrxBandParam[10];
} S_RCTLIB_LTE_FHC_MEASUREMENT_PARAM_T
    
```

6.3.5.17 RCTLIB_LTE_FHC_FetchResult

Definition:

```

int __stdcall RCTLIB_LTE_FHC_FetchResult(const S_RCTLIB_LTE_FHC_MEASUREMENT_PARAM_T* pSettings,
S_RCTLIB_LTE_FHC_MEASUREMENT_RESULT_T* pResult);
    
```

Description:

Fetch the single tone power measurement results of TX arrangements.

Parameter:

Table 6-120 RCTLIB LTE FHC FetchResult parameter

Parameter	Direction	Parameter
pSettings	IN	The arrangements of TX single tone power measurement and RX downlink single tone power.
pResult	OUT	The TX single tone power measurement results of TX arrangements.

Related Data Structure:

```
typedef struct
{
    double      dMeasuredPower[27]; /// TX measured power level (dBm)
    unsigned char  ucPwrStepNum;
} S_RCTLIB_LTE_FHC_FREQ_RESULT_U;
```

```
typedef struct
{
    /// integrity of the measurement result
    int      integrity;

    /// band
    unsigned int  uiBand; // index start from 0 ==> band1
    /// frequency step
    int      iNumberOfFrequencySteps;
    S_RCTLIB_LTE_FHC_FREQ_RESULT_U sTxFreqResult[22];
} S_RCTLIB_LTE_FHC_BAND_RESULT_U;
```

```
/**
 * The structure for storing LTE FHC TX power result
 */
```

```
typedef struct
{
    /// TX Band Result
    S_RCTLIB_LTE_FHC_BAND_RESULT_U sTxBandResult[10];
} S_RCTLIB_LTE_FHC_MEASUREMENT_RESULT_T
```

6.3.5.18 RCTLIB_LTE_NSFT_PreSetting

Definition:

```
int __stdcall RCTLIB_LTE_NSFT_PreSetting(const S_RCTLIB_LTE_NSFT_PRESETTINGS_T* pSettings);
```


**Description:**

Preset the instrument LTE FDD/TDD EVM/ACLR/Ripple/SEM/Power measurements.

Parameter:

Table 6-121 RCTLIB LTE NSFT presetting parameter

Parameter	Direction	Parameter
pSettings	IN	Preset to the bandwidth.

Related Data Structure:

```
typedef struct
{
    unsigned char ucBandWidth; //0: 1.4MHz, 1: 3MHz, 2: 5MHz, 3: 10MHz, 4: 15MHz, 5: 20MHz
} S_RCTLIB_LTE_NSFT_PRESETTINGS_T;
```

6.3.5.19 RCTLIB_LTE_NSFT_TX_ListMode**Definition:**

```
int __stdcall RCTLIB_LTE_NSFT_TX_ListMode(const S_RCTLIB_LTE_NSFT_TX_CONFIG_T* pSettings);
```

Description:

Initial the instrument for ready to measure the PUSCH TX arrangements of DUT. The test items, which is formed as a bit map, can be found in the S_RCTLIB_LTE_NSFT_TEST_TX_STEP_REQ_T structure. The DUT TX arrangements are shown in the Figure. 1. The arrangement of each band are started from a downlink synchronous procedure in a specific duration(usFddDSyncTime and usTddDSyncTime) except first band in the arrangements. Others, each band arrangement are composed of many frequencies(S_RCTLIB_LTE_NSFT_TX_FREQ_CONFIG_T). And, different frequency contains PUSCH TX steps to measure EVM/ACLR/Ripple/SEM/Power.

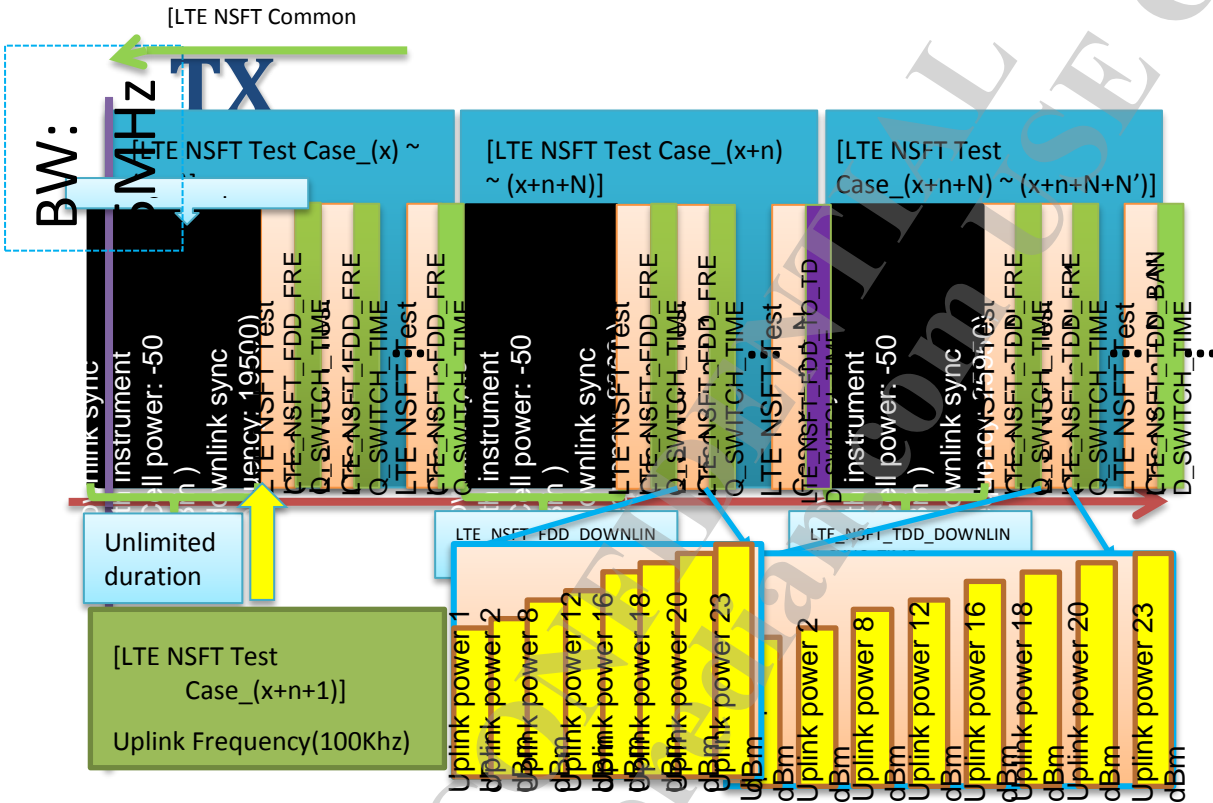


Figure. 1 LTE NSFT PUSCH TX DUT arrangements.

Parameter:

Table 6-122 RCTLIB LTE NSFT TX ListMode parameter

Parameter	Direction	Parameter
pSettings	IN	TX NSFT step arrangements structure

Related Data Structure:

```
typedef struct
{
    /* TX test check items */
    bool            bTxPowerCheck;

    /* MS setting */
    unsigned char    ucTxTestStep;
    char            cRbOffset1;    //start VRB for the first period
    unsigned char    ucRbLength1;
    char            cRbOffset2;    //start VRB for the second period
    unsigned char    ucRbLength2;
    unsigned char    ucNoStep2ChangeVRB; //0: Tx VRB will be changed to p2 at step 0
    float            fStartTargetPower;
    float            fStepPower;
    float            fEndTargetPower;
} S_RCTLIB_LTE_NSFT_TEST_TX_SWEEP_REQ_T;
```

```
typedef struct
{
    /* TX test check items */
    bool            bTxPowerCheck[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    bool            bEvmCheck[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    bool            bAclrCheck[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    bool            bSemCheck[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    bool            bFlatnessCheck[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    bool            bGainErrCheck[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    bool            bPhaseErrCheck[LTE_NSFT_MAX_TX_STEP_TEST_NUM];

    /* MS setting */
    unsigned char    ucTxTestStep;
    char            cRbOffset[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
```

```

unsigned char    ucRbLength[LTE_NSFT_MAX_TX_STEP_TEST_NUM];

float            fTargetPower[LTE_NSFT_MAX_TX_STEP_TEST_NUM];

} S_RCTLIB_LTE_NSFT_TEST_TX_STEP_REQ_T;

/* TX NSFT test command */

typedef struct
{
    /* Inst. setting and RX test check items */

    union
    {
        S_RCTLIB_LTE_NSFT_TEST_TX_STEP_REQ_T    txStepCmd;
        S_RCTLIB_LTE_NSFT_TEST_TX_SWEEP_REQ_T    txSweepCmd;
    };

    /* TX test command */

    unsigned short ulFrequency;

    char            cRequestOpetions; //0: follow power steps, 1: sweep powers with step (dBm)

} S_RCTLIB_LTE_NSFT_TX_FREQ_CONFIG_T;

```

```

typedef struct
{
    unsigned char    ucDuplexMode; /* Refer to Tx duplexMode - tdd,fdd */

    unsigned char    ucBand;

    unsigned char    mcsMode; /* Refer to Tx mcsMode */

    unsigned char    ucFreqNum;

    unsigned short    usDlFrequency; // unit: 100KHz

    S_RCTLIB_LTE_NSFT_TX_FREQ_CONFIG_T    txFreqCmd[13];

} S_RCTLIB_LTE_NSFT_TX_BAND_CONFIG_T;

```

```

typedef struct
{

```

```

unsigned char   ucFddTxStepWidth;//ms unit
unsigned char   ucTddTxStepWidth;//ms unit
unsigned char   ucFddToTddSwitchTime;//ms unit
unsigned char   ucFddFreqSwitchTime;//ms unit
unsigned char   ucTddFreqSwitchTime;//ms unit
unsigned char   ucFddBandSwitchTime;//ms unit
unsigned char   ucTddBandSwitchTime;//ms unit
unsigned char   ucBandNum;
unsigned short  usFddDISyncTime;//ms unit
unsigned short  usTddDISyncTime;//ms unit
double         dDownlinkSyncPowerDbm;
unsigned char   ucBandwidth; /* Refer to Tx ulBandwidth - 6,15,25,50,75,100RB */
S_RCTLIB_LTE_NSFT_TX_BAND_CONFIG_T txBandCmd[10];
} S_RCTLIB_LTE_NSFT_TX_CONFIG_T;

```



6.3.5.20 RCTLIB_LTE_NSFT_TX_FetchListMode

Definition:

```
int __stdcall RCTLIB_LTE_NSFT_TX_FetchListMode(const S_RCTLIB_LTE_NSFT_TX_CONFIG_T* pSettings, S_RCTLIB_LTE_NSFT_TX_RESULT_T* pResult);
```

Description:

Fetch TX NSFT measurement results from instrument.

Parameter:

Table 6-123 RCTLIB LTE NSFT TX FetchListMode parameter

Parameter	Direction	Parameter
pSettings	IN	TX NSFT step arrangements structure
pResult	OUT	TX NSFT measurement results of each step

Related Data Structure:

```
typedef struct
{
#define LTE_NSFT_MAX_TX_STEP_TEST_NUM      8

    double          dPoutPower[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    S_RCTLIB_LTE_NSFT_EVM_RESULT_T   sEVMResult[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    S_RCTLIB_LTE_NSFT_RIPPLE_RESULT_T sRpResult[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    S_RCTLIB_LTE_NSFT_ACP_RESULT_T   sACPResult[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
    S_RCTLIB_LTE_NSFT_SEM_RESULT_T   sSEMResult[LTE_NSFT_MAX_TX_STEP_TEST_NUM];
} S_RCTLIB_LTE_NSFT_TEST_TX_STEP_RESULT_T;

/**
 * the structure for storing TX list mode sweep result
 */
typedef struct
{
#define LTE_NSFT_MAX_TX_SWEEP_TEST_NUM      63

    double          dPoutPower[LTE_NSFT_MAX_TX_SWEEP_TEST_NUM];
} S_RCTLIB_LTE_NSFT_TEST_TX_SWEEP_RESULT_T;

/* TX NSFT test results */
typedef struct
{
    unsigned char    ucMeasureStep;
    char             cRequestOptions;

    /* Inst. setting and RX test check items */
    union
    {
        S_RCTLIB_LTE_NSFT_TEST_TX_STEP_RESULT_T   txStepResult;
        S_RCTLIB_LTE_NSFT_TEST_TX_SWEEP_RESULT_T   txSweepResult;
    };
} S_RCTLIB_LTE_NSFT_TX_TEST_FREQ_RESULT_T;
```



```
typedef struct
{
    S_RCTLIB_LTE_NSFT_TX_TEST_FREQ_RESULT_T    txFreqResults[13];
} S_RCTLIB_LTE_NSFT_TX_TEST_BAND_RESULT_T;

typedef struct
{
    S_RCTLIB_LTE_NSFT_TX_TEST_BAND_RESULT_T    txBandResults[10];
} S_RCTLIB_LTE_NSFT_TX_RESULT_T;
```

6.3.5.21 RCTLIB_LTE_NSFT_ChangeCellPower

Definition:

int __stdcall RCTLIB_LTE_NSFT_ChangeCellPower(double d_cell_power);

Description:

Command the instrument to set the specific downlink power.

Parameter:

Table 6-124 RCTLIB LTE NSFT ChangeCellPower parameter

Parameter	Direction	Parameter
d_cell_power	IN	Downlink power

Related Data Structure:

N/A

6.3.5.22 RCTLIB_LTE_NSFT_InitiateBER

Definition:

int __stdcall RCTLIB_LTE_NSFT_InitiateBER(const S_RCTLIB_LTE_NSFT_RX_BER_T* pSettings);

Description:

Mediatek Confidential
© 2017 Mediatek Inc.
This document contains information that is proprietary to Mediatek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.
Classification: Confidential B



Command the instrument to initial settings for BER test.

Parameter:

Table 6-125 RCTLIB LTE NSFT InitiateBER parameter

Parameter	Direction	Parameter
pSettings	IN	The parameters of BER initial settings.

Related Data Structure:

```
typedef struct
{
    unsigned char    ucBandWidth; //0: 1.4MHz, 1: 3MHz, 2: 5MHz, 3: 10MHz, 4: 15MHz, 5: 20MHz
    unsigned char    ucDuplexMode; /* Refer to Tx duplexMode - tdd,fdd */
    unsigned char    ucBand;
    unsigned char    mcsMode; /* Refer to Tx mcsMode */
    unsigned short   usDIFrequency; // unit: 100KHz
} S_RCTLIB_LTE_NSFT_RX_BER_T;
```

6.3.5.23 RCTLIB_LTE_NSFT_GetCableLoss

Definition:

```
int __stdcall RCTLIB_LTE_NSFT_GetCableLoss(unsigned int band, double* dOutputAttenuation, double* dInputAttenuation);
```

Description:

Acquire input/output attenuations of the instrument for specific band index.

Parameter:

Table 6-126 RCTLIB LTE NSFT GetCableLoss parameter

Parameter	Direction	Parameter
band	IN	The specific band to acquire attenuations of the instrument.
dOutputAttenuation	OUT	The output attenuation
dInputAttenuation	OUT	The input attenuation



Related Data Structure:

N/A

6.3.5.24 RCTLIB_LTE_GET_SpecificSettings

Definition:

int __stdcall RCTLIB_LTE_GET_SpecificSettings(S_RCTLIB_LTE_SPECIFIC_SETTINGS_T* pSettings);

Description:

Get specific setting of the instrument to use in the calibration flow.

Parameter:

Table 6-127 RCTLIB LTE GET SpecificSettings parameter

Parameter	Direction	Parameter
pSettings	IN	The structure to store the specific setting of the instrument.

Related Data Structure:

typedef struct

```
{
    /// ===== FHC instrument parameters =====
    /// FHC RX Max Step Count
    unsigned int uiFHC_RX_MaxStepCount;
    /// FHC TX Max Step Count
    unsigned int uiFHC_TX_MaxStepCount;
    /// FHC Multiple Band Support
    unsigned char ucFHC_MultiBand; // 0: single band, 1: multiple band support
    /// FHC FDD to TDD switch time (ms)
    unsigned char ucFHC_Fdd2TddSwitchTimeMS;
    /// FHC TDD to FDD switch time (ms)
    unsigned char ucFHC_Tdd2FddSwitchTimeMS;
    /// FHC FDD TX to RX switch time (ms)
    unsigned char ucFHC_FddTx2RxSwitchTimeMS;
    /// FHC TDD TX to RX switch time (ms)
    unsigned char ucFHC_TddTx2RxSwitchTimeMS;
    /// FHC frequency switch time (ms)
    unsigned char ucFHC_FreqSwitchTimeMS;
    /// FHC band switch time (ms)
    unsigned char ucFHC_BandSwitchTimeMS;
    /// FHC tx step width (ms)
    unsigned char ucFHC_TxStepWidthMS;
```

```
    /// ===== NSFT instrument parameters =====
    /// NSFT RX Max Step Count
    unsigned int uiNSFT_RX_MaxStepCount;
    /// NSFT TX Max Step Count
    unsigned int uiNSFT_TX_MaxStepCount;
    /// NSFT Multiple Band Support
```

typedef struct

```

{
    /// ===== FHC instrument parameters =====
    /// FHC RX Max Step Count
    unsigned int uiFHC_RX_MaxStepCount;
    /// FHC TX Max Step Count
    unsigned int uiFHC_TX_MaxStepCount;
    /// FHC Multiple Band Support
    unsigned char ucFHC_MultiBand; // 0: single band, 1: multiple band support
    /// FHC FDD to TDD switch time (ms)
    unsigned char ucFHC_Fdd2TddSwitchTimeMS;
    /// FHC TDD to FDD switch time (ms)
    unsigned char ucFHC_Tdd2FddSwitchTimeMS;
    /// FHC FDD TX to RX switch time (ms)
    unsigned char ucFHC_FddTx2RxSwitchTimeMS;
    /// FHC TDD TX to RX switch time (ms)
    unsigned char ucFHC_TddTx2RxSwitchTimeMS;
    /// FHC frequency switch time (ms)
    unsigned char ucFHC_FreqSwitchTimeMS;
    /// FHC band switch time (ms)
    unsigned char ucFHC_BandSwitchTimeMS;
    /// FHC tx step width (ms)
    unsigned char ucFHC_TxStepWidthMS;

    unsigned char ucNSFT_MultiBand; // 0: single band, 1: multiple band support
    /// NSFT FDD TX step width (ms) bandwidth[1.4MB, 5MB, 10MB, 15MB, 20MB]
    unsigned char ucNSFT_FddTxSetpWidthMS[5];
    /// NSFT TDD TX step width (ms) bandwidth[1.4MB, 5MB, 10MB, 15MB, 20MB]
    unsigned char ucNSFT_TddTxSetpWidthMS[5];
    /// NSFT FDD to TDD switch time (ms)

```

typedef struct

```
{
    /// ===== FHC instrument parameters =====
    /// FHC RX Max Step Count
    unsigned int uiFHC_RX_MaxStepCount;
    /// FHC TX Max Step Count
    unsigned int uiFHC_TX_MaxStepCount;
    /// FHC Multiple Band Support
    unsigned char ucFHC_MultiBand; // 0: single band, 1: multiple band support
    /// FHC FDD to TDD switch time (ms)
    unsigned char ucFHC_Fdd2TddSwitchTimeMS;
    /// FHC TDD to FDD switch time (ms)
    unsigned char ucFHC_Tdd2FddSwitchTimeMS;
    /// FHC FDD TX to RX switch time (ms)
    unsigned char ucFHC_FddTx2RxSwitchTimeMS;
    /// FHC TDD TX to RX switch time (ms)
    unsigned char ucFHC_TddTx2RxSwitchTimeMS;
    /// FHC frequency switch time (ms)
    unsigned char ucFHC_FreqSwitchTimeMS;
    /// FHC band switch time (ms)
    unsigned char ucFHC_BandSwitchTimeMS;
    /// FHC tx step width (ms)
    unsigned char ucFHC_TxStepWidthMS;
    unsigned char ucNSFT_Fdd2TddSwitchTimeMS;
    /// NSFT FDD frequency switch time (ms)
    unsigned char ucNSFT_FddFreqSwitchTimeMS;
    /// NSFT TDD frequency switch time (ms)
    unsigned char ucNSFT_TddFreqSwitchTimeMS;
    /// NSFT FDD band switch time (ms)
```

typedef struct

```

{
    /// ===== FHC instrument parameters =====
    /// FHC RX Max Step Count
    unsigned int uiFHC_RX_MaxStepCount;
    /// FHC TX Max Step Count
    unsigned int uiFHC_TX_MaxStepCount;
    /// FHC Multiple Band Support
    unsigned char ucFHC_MultiBand; // 0: single band, 1: multiple band support
    /// FHC FDD to TDD switch time (ms)
    unsigned char ucFHC_Fdd2TddSwitchTimeMS;
    /// FHC TDD to FDD switch time (ms)
    unsigned char ucFHC_Tdd2FddSwitchTimeMS;
    /// FHC FDD TX to RX switch time (ms)
    unsigned char ucFHC_FddTx2RxSwitchTimeMS;
    /// FHC TDD TX to RX switch time (ms)
    unsigned char ucFHC_TddTx2RxSwitchTimeMS;
    /// FHC frequency switch time (ms)
    unsigned char ucFHC_FreqSwitchTimeMS;
    /// FHC band switch time (ms)
    unsigned char ucFHC_BandSwitchTimeMS;
    /// FHC tx step width (ms)
    unsigned char ucFHC_TxStepWidthMS;
    unsigned char ucNSFT_FddBandSwitchTimeMS;
    /// NSFT TDD band switch time (ms)
    unsigned char ucNSFT_TddBandSwitchTimeMS;
    /// NSFT FDD Downlink sync time (ms)
    unsigned short usNSFT_FddDownlinkSyncTimeMS;
    /// NSFT TDD Downlink sync time (ms)

```

typedef struct

```
{
    /// ===== FHC instrument parameters =====
    /// FHC RX Max Step Count
    unsigned int uiFHC_RX_MaxStepCount;
    /// FHC TX Max Step Count
    unsigned int uiFHC_TX_MaxStepCount;
    /// FHC Multiple Band Support
    unsigned char ucFHC_MultiBand; // 0: single band, 1: multiple band support
    /// FHC FDD to TDD switch time (ms)
    unsigned char ucFHC_Fdd2TddSwitchTimeMS;
    /// FHC TDD to FDD switch time (ms)
    unsigned char ucFHC_Tdd2FddSwitchTimeMS;
    /// FHC FDD TX to RX switch time (ms)
    unsigned char ucFHC_FddTx2RxSwitchTimeMS;
    /// FHC TDD TX to RX switch time (ms)
    unsigned char ucFHC_TddTx2RxSwitchTimeMS;
    /// FHC frequency switch time (ms)
    unsigned char ucFHC_FreqSwitchTimeMS;
    /// FHC band switch time (ms)
    unsigned char ucFHC_BandSwitchTimeMS;
    /// FHC tx step width (ms)
    unsigned char ucFHC_TxStepWidthMS;
    unsigned short usNSFT_TddDownlinkSyncTimeMS;
} S_RCTLIB_LTE_SPECIFIC_SETTINGS_T;
```

6.4 Customized Instrument Porting Interface

The main goal of customized instrument porting interface is helping the users to create the new instrument calibration control themselves or cooperate with instrument supplier. Users could register the instrument control functions of GSM, WCDMA, or TDSCDMA mentioned in the last section by the callback register entry as shown

below figure. After the callback functions registered callback functions, the calibration library will call the instrument control callback function in the user’s application through the instrument library.

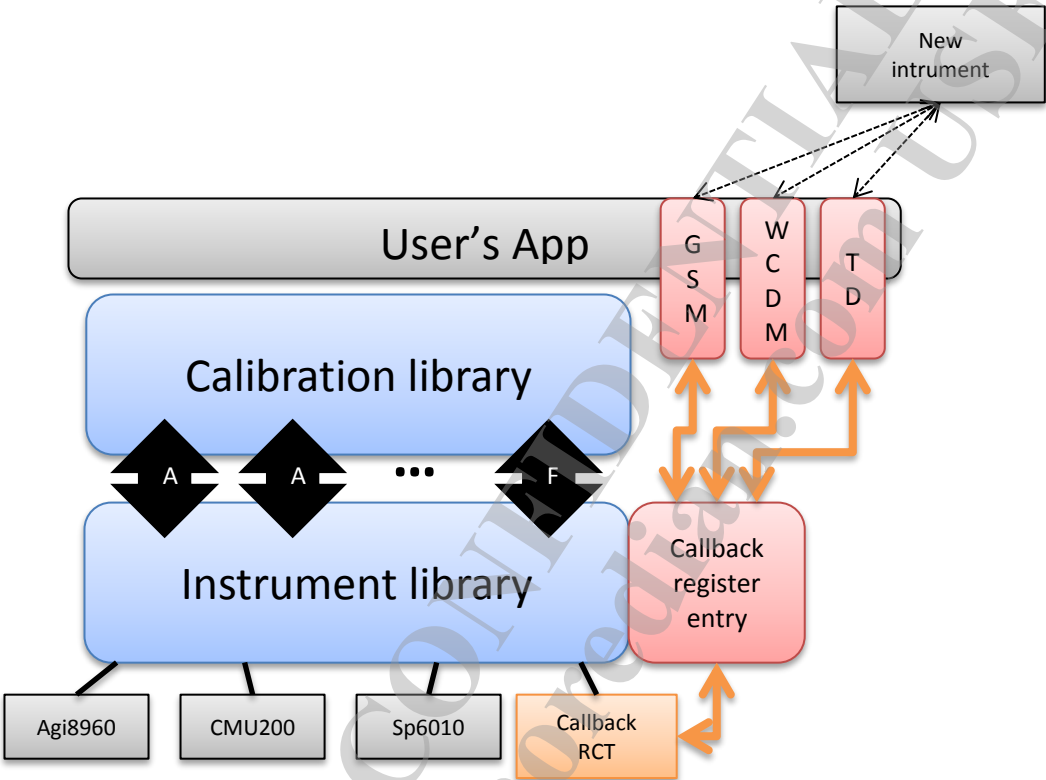


Figure 6-5 Customized instrument porting interface

6.4.1 Callback Register Entry

The callback functions can be split into four categories as mentioned in section 6.3. Therefore, the callback register entry also have four register functions for each category. Please reference the section 6.3 or the existing instrument control code for the functionality of callback instrument control functions

6.4.1.1 Common Callback Function Definitions

Callback Register Function Definition:

```
int __stdcall RCTLIB_SetCommonInstrumentInstance(RCTLIB_COMMON_CALLBACKS_CFG_T * instCallbacks);
```

Description:

Set instrument common callback functions.

Parameter:

Table 6-128 Common callback function parameter



Parameter	Direction	Parameter
instCallbacks	IN	the structure to get TDSCDMA instrument's special settings

Related Callback Function Definitions:

```
typedef int (*RCTLIB_InitializeConfiguration_CALLBACK)(const char * cfg_file_path);  
typedef int (*RCTLIB_ConnectRCT_CALLBACK)(void);  
typedef int (*RCTLIB_DisconnectRCT_CALLBACK)(void);  
typedef int (*RCTLIB_QueryCurrentApplicationFormat_CALLBACK)(char *buf, int buf_len);  
typedef int (*RCTLIB_ConfigApplicationFormat_CALLBACK)(unsigned int format);  
typedef int (*RCTLIB_ConfigOperatingMode_CALLBACK)(unsigned int OperatingMode);  
typedef int (*RCTLIB_Reset_CALLBACK)(void);
```

Related Data Structure:

```
typedef struct  
{  
    RCTLIB_InitializeConfiguration_CALLBACK InitializeConfiguration_CALLBACK;  
    RCTLIB_ConnectRCT_CALLBACK ConnectRCT_CALLBACK;  
    RCTLIB_DisconnectRCT_CALLBACK DisconnectRCT_CALLBACK;  
    RCTLIB_QueryCurrentApplicationFormat_CALLBACK QueryCurrentApplicationFormat_CALLBACK;  
    RCTLIB_ConfigApplicationFormat_CALLBACK ConfigApplicationFormat_CALLBACK;  
    RCTLIB_ConfigOperatingMode_CALLBACK ConfigOperatingMode_CALLBACK;  
    RCTLIB_Reset_CALLBACK Reset_CALLBACK;  
}RCTLIB_COMMON_CALLBACKS_CFG_T;
```

6.4.1.2 GSM & EDGE Callback Function Definitions

```
int __stdcall RCTLIB_SetGGEInstrumentInstance(RCTLIB_GGE_CALLBACKS_CFG_T * instCallbacks);
```

Description:

Set GSM/GPRS/EPSC instrument control callback functions.

Parameter:

Table 6-129 GSM & EDGE callback function parameter



Parameter	Direction	Parameter
instCallbacks	IN	the structure to get TDSCDMA instrument's special settings

Related Callback Function Definitions:

```
typedef int (*RCLIB_GGE_InstrumentInit_CALLBACK)(void);

typedef int (*RCLIB_GGE_Cableloss_Settings_CALLBACK)(S_RCLIB_GGE_Cableloss_CONFIG_T cable_loss);

typedef int (*RCLIB_GGE_ConfigCellPower_CALLBACK)(double power);

typedef int (*RCLIB_GGE_ConfigDefaultSettings_CALLBACK)(void);

typedef int (*RCLIB_GGE_ConfigAnalyzerFrequencyOffset_CALLBACK)(const S_RCLIB_GGE_FREQUENCY_OFFSET_SETTINGS_T* pSettings, unsigned int sz);

typedef int (*RCLIB_GGE_CAPID_PreSettings_CALLBACK)(const S_RCLIB_GGE_CAPID_SETTINGS_T* pSettings);

typedef int (*RCLIB_GGE_CAPID_Iteration_CALLBACK)(double *frequency_error);

typedef int (*RCLIB_GGE_AFC_PreSettings_CALLBACK)(const S_RCLIB_GGE_AFC_SETTINGS_T* pSettings);

typedef int (*RCLIB_GGE_AGC_PreSettings_CALLBACK)(double d_power);

typedef int (*RCLIB_GGE_AGC_ChangeCellBand_CALLBACK)(unsigned int band);

typedef int (*RCLIB_GGE_AGC_ChangeChannel_CALLBACK)(unsigned int arfcn);

typedef int (*RCLIB_GGE_APCDCOffset_PreSettings_CALLBACK)(unsigned int tsc);

typedef int (*RCLIB_GGE_APCDCOffset_ChangeCellBand_CALLBACK)(unsigned int band);

typedef int (*RCLIB_GGE_APCDCOffset_Iteration_CALLBACK)(unsigned int arfcn, double expected_power, int PCL, double *d_power);

typedef int (*RCLIB_GGE_EDGE_APCDCOffset_PreSettings_CALLBACK)(unsigned int tsc);

typedef int (*RCLIB_GGE_EDGE_APCDCOffset_ChangeCellBand_CALLBACK)(unsigned int band);

typedef int (*RCLIB_GGE_EDGE_APCDCOffset_Iteration_CALLBACK)(unsigned int arfcn, double expected_power, int PCL, double *d_power);

typedef int (*RCLIB_GGE_FHC_DTS_PreSettings_CALLBACK)();

typedef int (*RCLIB_GGE_FHC_DTS_Iteration_CALLBACK)(S_RCLIB_FHC_DL_List_T List, int ListLength);

typedef int (*RCLIB_GGE_FHC_DTS_START_CALLBACK)();

typedef int (*RCLIB_GGE_FHC_DTS_STOP_CALLBACK)();

typedef int (*RCLIB_GGE_FHC_UTS_PreSettings_CALLBACK)();
```

```
typedef int (*RCTLIB_GGE_FHC_UTS_Iteration_CALLBACK)(S_RCTLIB_GGE_FHC_TX_UTS_T *uts);

typedef int (*RCTLIB_GGE_FHC_UTS_FetchResult_CALLBACK)(const S_RCTLIB_GGE_FHC_TX_UTS_T *uts,
S_RCTLIB_GGE_FHC_TX_UTS_RESULT_T *uts_result);

typedef int (*RCTLIB_GGE_FBDAC_PreSettings_CALLBACK)(const S_RCTLIB_GGE_FBDAC_SETTINGS_T*
pSettings);

typedef int (*RCTLIB_GGE_FBDAC_Iteration_CALLBACK)(double* d_power);

typedef int (*RCTLIB_GGE_TXIQ_PreSettings_CALLBACK)(const S_RCTLIB_GGE_TXIQ_SETTINGS_T* pSettings);

typedef int (*RCTLIB_GGE_TXIQ_ChangeBand_CALLBACK)(const S_RCTLIB_GGE_TXIQ_SETTINGS_T* pSettings);

typedef int (*RCTLIB_GGE_TXIQ_Iteration_CALLBACK)(const S_RCTLIB_GGE_TXIQ_SETTINGS_T* pSettings,
S_RCTLIB_GGE_TXIQ_RESULT_T *pTxIqResult);

typedef int (*RCTLIB_GGE_TXSlopeSkew_PreSettings_CALLBACK)(const
S_RCTLIB_GGE_TXSLOPESKEW_SETTINGS_T* pSettings);

typedef int (*RCTLIB_GGE_TXSlopeSkew_Iteration_CALLBACK)(double* d_mod_depth);

typedef int (*RCTLIB_GGE_TRXOffset_PreSettings_CALLBACK)(const S_RCTLIB_GGE_TRXOFFSET_SETTINGS_T*
pSettings);

typedef int (*RCTLIB_GGE_TRXOffset_InitAFC_CALLBACK)(const S_RCTLIB_GGE_TRXOFFSET_SETTINGS_T*
pSettings);

typedef int (*RCTLIB_GGE_TRXOffset_Iteration_CALLBACK)(double *frequency_err);

typedef int (*RCTLIB_GGE_NSFT_PreSettings_CALLBACK)(unsigned int measurement_count, unsigned int
ber_count);

typedef int (*RCTLIB_GGE_NSFT_GMSKInit_CALLBACK)(const S_RCTLIB_GGE_NSFT_TESTCONFIG_T*
nsft_config);

typedef int (*RCTLIB_GGE_NSFT_EPSKInit_CALLBACK)(const S_RCTLIB_GGE_NSFT_TESTCONFIG_T* nsft_config);

typedef int (*RCTLIB_GGE_NSFT_BERInit_CALLBACK)(const S_RCTLIB_GGE_NSFT_TESTCONFIG_T* nsft_config);

typedef int (*RCTLIB_GGE_NSFT_ChangePCL_CALLBACK)(int b_EPSK,unsigned int pcl);

typedef int (*RCTLIB_GGE_NSFT_ReadGMSKPerformance_CALLBACK)(S_RCTLIB_GGE_NSFT_GMSK_RESULT_T
*gmsk_result);

typedef int (*RCTLIB_GGE_NSFT_InitiateBER_CALLBACK)();

typedef int (*RCTLIB_GGE_NSFT_FetchBER_CALLBACK)(S_RCTLIB_GGE_NSFT_BER_RESULT_T* ber_result);

typedef int (*RCTLIB_GGE_NSFT_ReadEPSKPerformance_CALLBACK)(S_RCTLIB_GGE_NSFT_EPSK_RESULT_T
*epsk_result);

typedef int (*RCTLIB_GGE_GET_SpecificSettings_CALLBACK)(S_RCTLIB_GGE_SPECIFIC_SETTINGS_T *pSettings);
```

Related Data Structure:

typedef struct

```
{
    RCTLIB_GGE_InstrumentInit_CALLBACK InstrumentInit_CALLBACK;
    RCTLIB_GGE_Cableloss_Settings_CALLBACK Cableloss_Settings_CALLBACK;
    RCTLIB_GGE_ConfigCellPower_CALLBACK ConfigCellPower_CALLBACK;
    RCTLIB_GGE_ConfigDefaultSettings_CALLBACK ConfigDefaultSettings_CALLBACK;
    RCTLIB_GGE_ConfigAnalyzerFrequencyOffset_CALLBACK ConfigAnalyzerFrequencyOffset_CALLBACK;
    RCTLIB_GGE_CAPID_PreSettings_CALLBACK CAPID_PreSettings_CALLBACK;
    RCTLIB_GGE_CAPID_Iteration_CALLBACK CAPID_Iteration_CALLBACK;
    RCTLIB_GGE_AFC_PreSettings_CALLBACK AFC_PreSettings_CALLBACK;
    RCTLIB_GGE_AGC_PreSettings_CALLBACK AGC_PreSettings_CALLBACK;
    RCTLIB_GGE_AGC_ChangeCellBand_CALLBACK AGC_ChangeCellBand_CALLBACK;
    RCTLIB_GGE_AGC_ChangeChannel_CALLBACK AGC_ChangeChannel_CALLBACK;
    RCTLIB_GGE_APCDCOffset_PreSettings_CALLBACK APCDCOffset_PreSettings_CALLBACK;
    RCTLIB_GGE_APCDCOffset_ChangeCellBand_CALLBACK APCDCOffset_ChangeCellBand_CALLBACK;
    RCTLIB_GGE_APCDCOffset_Iteration_CALLBACK APCDCOffset_Iteration_CALLBACK;
    RCTLIB_GGE_EDGE_APCDCOffset_PreSettings_CALLBACK EDGE_APCDCOffset_PreSettings_CALLBACK;
    RCTLIB_GGE_EDGE_APCDCOffset_ChangeCellBand_CALLBACK
        EDGE_APCDCOffset_ChangeCellBand_CALLBACK;
    RCTLIB_GGE_EDGE_APCDCOffset_Iteration_CALLBACK EDGE_APCDCOffset_Iteration_CALLBACK;
    RCTLIB_GGE_FHC_DTS_PreSettings_CALLBACK FHC_DTS_PreSettings_CALLBACK;
    RCTLIB_GGE_FHC_DTS_Iteration_CALLBACK FHC_DTS_Iteration_CALLBACK;
    RCTLIB_GGE_FHC_DTS_START_CALLBACK FHC_DTS_START_CALLBACK;
    RCTLIB_GGE_FHC_DTS_STOP_CALLBACK FHC_DTS_STOP_CALLBACK;
    RCTLIB_GGE_FHC_UTS_PreSettings_CALLBACK FHC_UTS_PreSettings_CALLBACK;
    RCTLIB_GGE_FHC_UTS_Iteration_CALLBACK FHC_UTS_Iteration_CALLBACK;
    RCTLIB_GGE_FHC_UTS_FetchResult_CALLBACK FHC_UTS_FetchResult_CALLBACK;
    RCTLIB_GGE_FBDAC_PreSettings_CALLBACK FBDAC_PreSettings_CALLBACK;
    RCTLIB_GGE_FBDAC_Iteration_CALLBACK FBDAC_Iteration_CALLBACK;
}
```

```

RCTLIB_GGE_TXIQ_PreSettings_CALLBACK TXIQ_PreSettings_CALLBACK;

RCTLIB_GGE_TXIQ_ChangeBand_CALLBACK TXIQ_ChangeBand_CALLBACK;

RCTLIB_GGE_TXIQ_Iteration_CALLBACK TXIQ_Iteration_CALLBACK;

RCTLIB_GGE_TXSlopeSkew_PreSettings_CALLBACK TXSlopeSkew_PreSettings_CALLBACK;

RCTLIB_GGE_TXSlopeSkew_Iteration_CALLBACK TXSlopeSkew_Iteration_CALLBACK;

RCTLIB_GGE_TRXOffset_PreSettings_CALLBACK TRXOffset_PreSettings_CALLBACK;

RCTLIB_GGE_TRXOffset_InitAFC_CALLBACK TRXOffset_InitAFC_CALLBACK;

RCTLIB_GGE_TRXOffset_Iteration_CALLBACK TRXOffset_Iteration_CALLBACK;

RCTLIB_GGE_NSFT_PreSettings_CALLBACK NSFT_PreSettings_CALLBACK;

RCTLIB_GGE_NSFT_GMSKInit_CALLBACK NSFT_GMSKInit_CALLBACK;

RCTLIB_GGE_NSFT_EPSKInit_CALLBACK NSFT_EPSKInit_CALLBACK;

RCTLIB_GGE_NSFT_BERInit_CALLBACK NSFT_BERInit_CALLBACK;

RCTLIB_GGE_NSFT_ChangePCL_CALLBACK NSFT_ChangePCL_CALLBACK;

RCTLIB_GGE_NSFT_ReadGMSKPerformance_CALLBACK NSFT_ReadGMSKPerformance_CALLBACK;

RCTLIB_GGE_NSFT_InitiateBER_CALLBACK NSFT_InitiateBER_CALLBACK;

RCTLIB_GGE_NSFT_FetchBER_CALLBACK NSFT_FetchBER_CALLBACK;

RCTLIB_GGE_NSFT_ReadEPSPPerformance_CALLBACK NSFT_ReadEPSPPerformance_CALLBACK;

RCTLIB_GGE_GET_SpecificSettings_CALLBACK GET_SpecificSettings_CALLBACK;

}RCTLIB_GGE_CALLBACKS_CFG_T;
    
```

6.4.1.3 WCDMA Callback Function Definitions

Callback Register Function Definition:

```
int __stdcall RCTLIB_SetWCDMAInstrumentInstance(RCTLIB_WCDMA_CALLBACKS_CFG_T * instCallbacks);
```

Description:

Set WCDMA instrument control callback functions.

Parameter:

Table 6-130 WCDMA callback function parameter

Parameter	Direction	Parameter
instCallbacks	IN	the structure to get TDSCDMA instrument's special settings

Related Callback Function Definitions:

```
typedef int (*RCTLIB_WCDMA_InstrumentInit_CALLBACK)(void);

typedef int (*RCTLIB_WCDMA_ConfigCellPower_CALLBACK)(double power);

typedef int (*RCTLIB_WCDMA_ConfigDefaultSettings_CALLBACK)(void);

typedef int (*RCTLIB_WCDMA_AFC_PreSetting_CALLBACK)(const S_RCTLIB_WCDMA_AFC_PRESETTINGS_T*
pSettings, unsigned int size);

typedef int (*RCTLIB_WCDMA_AGC_PreSetting_CALLBACK)(const S_RCTLIB_WCDMA_AGC_PRESETTINGS_T*
pSettings);

typedef int (*RCTLIB_WCDMA_AGC_ChangeCellBand_CALLBACK)(unsigned int band);

typedef int (*RCTLIB_WCDMA_AGC_ChangeChannel_CALLBACK)(unsigned int uarfcn);

typedef int (*RCTLIB_WCDMA_AGC_ChangeCellPower_CALLBACK)(double cellPower);

typedef int (*RCTLIB_WCDMA_APC_PreSetting_CALLBACK)(const S_RCTLIB_WCDMA_APC_PRESETTINGS_T*
pSettings);

typedef int (*RCTLIB_WCDMA_APC_ChangeCellBand_CALLBACK)(unsigned int band);

typedef int (*RCTLIB_WCDMA_APC_ChangeChannel_CALLBACK)(unsigned int uarfcn);

typedef int (*RCTLIB_WCDMA_APC_ChangeExpectedPower_CALLBACK)(int expectedPower);

typedef int (*RCTLIB_WCDMA_APC_Initiate_CALLBACK)(void);

typedef int (*RCTLIB_WCDMA_APC_FetchResult_CALLBACK)(double *outputPower);

typedef int (*RCTLIB_WCDMA_FHC_PreSetting_CALLBACK)(const S_RCTLIB_WCDMA_FHC_PRESETTINGS_T
*pSettings);

typedef int (*RCTLIB_WCDMA_FHC_StartIteration_CALLBACK)(const
S_RCTLIB_WCDMA_FHC_MEASUREMENT_PARAM_T *pSettings);

typedef int (*RCTLIB_WCDMA_FHC_FetchResult_CALLBACK)(const
S_RCTLIB_WCDMA_FHC_MEASUREMENT_PARAM_T *pSettings,
S_RCTLIB_WCDMA_FHC_MEASUREMENT_RESULT_T* pResult);

typedef int (*RCTLIB_WCDMA_NSFT_PreSetting_CALLBACK)(const S_RCTLIB_WCDMA_NSFT_PRESETTINGS_T*
pSettings);

typedef int (*RCTLIB_WCDMA_NSFT_InitiateTestCase_CALLBACK)(const S_RCTLIB_WCDMA_NSFT_CONFIG_T*
pSettings);

typedef int (*RCTLIB_WCDMA_NSFT_TPC_CALLBACK)(const S_RCTLIB_WCDMA_NSFT_TPC_REQUEST_T
*tpc_request);
```



```

typedef int (*RCTLIB_WCDMA_NSFT_FetchTXPerformanceMaxPower_CALLBACK)(S_RCTLIB_WCDMA_NSFT_TX_PERF_MAX_POWER_RESULT_T *tx_perf_result);

typedef int (*RCTLIB_WCDMA_NSFT_FetchTXPerformanceMinPower_CALLBACK)(S_RCTLIB_WCDMA_NSFT_TX_PERF_MIN_POWER_RESULT_T *tx_perf_result);

typedef int (*RCTLIB_WCDMA_NSFT_InitiateILPCTestCase_CALLBACK)(unsigned char testSegment);

typedef int (*RCTLIB_WCDMA_NSFT_FetchILPCResult_CALLBACK)(S_RCTLIB_WCDMA_NSFT_ILPC_CONFIG_T* ilpcConfig, S_RCTLIB_WCDMA_NSFT_ILPC_RESULT_T* result);

typedef int (*RCTLIB_WCDMA_NSFT_ChangeCellPower_CALLBACK)(double d_cell_power);

typedef int (*RCTLIB_WCDMA_NSFT_InitiateBER_CALLBACK)();

typedef int (*RCTLIB_WCDMA_NSFT_ReadBER_CALLBACK)(S_RCTLIB_WCDMA_NSFT_BER_RESULT_T* result);

typedef int (*RCTLIB_WCDMA_NSFT_PRACH_PreSetting_CALLBACK)(const S_RCTLIB_WCDMA_NSFT_PRACH_PRESETTING_T* pSettings);

typedef int (*RCTLIB_WCDMA_NSFT_PRACH_CasePreSetting_CALLBACK)();

typedef int (*RCTLIB_WCDMA_NSFT_PRACH_InitiateTestCase_CALLBACK)(const S_RCTLIB_WCDMA_NSFT_PRACH_CONFIG_T* pSettings);

typedef int (*RCTLIB_WCDMA_NSFT_PRACH_FetchTestCase_CALLBACK)(S_RCTLIB_WCDMA_NSFT_PRACH_RESULT_T *result);

typedef int (*RCTLIB_WCDMA_HSDPA_NSFT_PreSetting_CALLBACK)(const S_RCTLIB_WCDMA_HSPA_NSFT_PRESETTINGS_T* pSettings);

typedef int (*RCTLIB_WCDMA_HSDPA_NSFT_InitiateTestCase_CALLBACK)(const S_RCTLIB_WCDMA_HSPA_NSFT_CONFIG_T* pConfig);

typedef int (*RCTLIB_WCDMA_HSDPA_NSFT_FetchResult_CALLBACK)(S_RCTLIB_WCDMA_HSPA_NSFT_RESULT_T* pResult);

typedef int (*RCTLIB_WCDMA_HSUPA_NSFT_PreSetting_CALLBACK)(const S_RCTLIB_WCDMA_HSPA_NSFT_PRESETTINGS_T* pSettings);

typedef int (*RCTLIB_WCDMA_HSUPA_NSFT_InitiateTestCase_CALLBACK)(const S_RCTLIB_WCDMA_HSPA_NSFT_CONFIG_T* pConfig);

typedef int (*RCTLIB_WCDMA_HSUPA_NSFT_FetchResult_CALLBACK)(S_RCTLIB_WCDMA_HSPA_NSFT_RESULT_T* pResult);

typedef int (*RCTLIB_WCDMA_GET_SpecificSettings_CALLBACK)(S_RCTLIB_WCDMA_SPECIFIC_SETTINGS_T *pSettings);

```

Related Data Structure:

typedef struct

```
{
    RCTLIB_WCDMA_InstrumentInit_CALLBACK InstrumentInit_CALLBACK;
    RCTLIB_WCDMA_ConfigCellPower_CALLBACK ConfigCellPower_CALLBACK;
    RCTLIB_WCDMA_ConfigDefaultSettings_CALLBACK ConfigDefaultSettings_CALLBACK;
    RCTLIB_WCDMA_AFC_PreSetting_CALLBACK AFC_PreSetting_CALLBACK;
    RCTLIB_WCDMA_AGC_PreSetting_CALLBACK AGC_PreSetting_CALLBACK;
    RCTLIB_WCDMA_AGC_ChangeCellBand_CALLBACK AGC_ChangeCellBand_CALLBACK;
    RCTLIB_WCDMA_AGC_ChangeChannel_CALLBACK AGC_ChangeChannel_CALLBACK;
    RCTLIB_WCDMA_AGC_ChangeCellPower_CALLBACK AGC_ChangeCellPower_CALLBACK;
    RCTLIB_WCDMA_APC_PreSetting_CALLBACK APC_PreSetting_CALLBACK;
    RCTLIB_WCDMA_APC_ChangeCellBand_CALLBACK APC_ChangeCellBand_CALLBACK;
    RCTLIB_WCDMA_APC_ChangeChannel_CALLBACK APC_ChangeChannel_CALLBACK;
    RCTLIB_WCDMA_APC_ChangeExpectedPower_CALLBACK APC_ChangeExpectedPower_CALLBACK;
    RCTLIB_WCDMA_APC_Initiate_CALLBACK APC_Initiate_CALLBACK;
    RCTLIB_WCDMA_APC_FetchResult_CALLBACK APC_FetchResult_CALLBACK;
    RCTLIB_WCDMA_FHC_PreSetting_CALLBACK FHC_PreSetting_CALLBACK;
    RCTLIB_WCDMA_FHC_StartIteration_CALLBACK FHC_StartIteration_CALLBACK;
    RCTLIB_WCDMA_FHC_FetchResult_CALLBACK FHC_FetchResult_CALLBACK;
    RCTLIB_WCDMA_NSFT_PreSetting_CALLBACK NSFT_PreSetting_CALLBACK;
    RCTLIB_WCDMA_NSFT_InitiateTestCase_CALLBACK NSFT_InitiateTestCase_CALLBACK;
    RCTLIB_WCDMA_NSFT_TPC_CALLBACK NSFT_TPC_CALLBACK;
    RCTLIB_WCDMA_NSFT_FetchTXPerformanceMaxPower_CALLBACK
        NSFT_FetchTXPerformanceMaxPower_CALLBACK;
    RCTLIB_WCDMA_NSFT_FetchTXPerformanceMinPower_CALLBACK
        NSFT_FetchTXPerformanceMinPower_CALLBACK;
    RCTLIB_WCDMA_NSFT_InitiateILPCTestCase_CALLBACK NSFT_InitiateILPCTestCase_CALLBACK;
    RCTLIB_WCDMA_NSFT_FetchILPCResult_CALLBACK NSFT_FetchILPCResult_CALLBACK;
    RCTLIB_WCDMA_NSFT_ChangeCellPower_CALLBACK NSFT_ChangeCellPower_CALLBACK;
    RCTLIB_WCDMA_NSFT_InitiateBER_CALLBACK NSFT_InitiateBER_CALLBACK;
```



```
RCTLIB_WCDMA_NSFT_ReadBER_CALLBACK NSFT_ReadBER_CALLBACK;

RCTLIB_WCDMA_NSFT_PRACH_PreSetting_CALLBACK NSFT_PRACH_PreSetting_CALLBACK;

RCTLIB_WCDMA_NSFT_PRACH_CasePreSetting_CALLBACK NSFT_PRACH_CasePreSetting_CALLBACK;

RCTLIB_WCDMA_NSFT_PRACH_InitiateTestCase_CALLBACK NSFT_PRACH_InitiateTestCase_CALLBACK;

RCTLIB_WCDMA_NSFT_PRACH_FetchTestCase_CALLBACK NSFT_PRACH_FetchTestCase_CALLBACK;

RCTLIB_WCDMA_HSDPA_NSFT_PreSetting_CALLBACK HSDPA_NSFT_PreSetting_CALLBACK;

RCTLIB_WCDMA_HSDPA_NSFT_InitiateTestCase_CALLBACK HSDPA_NSFT_InitiateTestCase_CALLBACK;

RCTLIB_WCDMA_HSDPA_NSFT_FetchResult_CALLBACK HSDPA_NSFT_FetchResult_CALLBACK;

RCTLIB_WCDMA_HSUPA_NSFT_PreSetting_CALLBACK HSUPA_NSFT_PreSetting_CALLBACK;

RCTLIB_WCDMA_HSUPA_NSFT_InitiateTestCase_CALLBACK HSUPA_NSFT_InitiateTestCase_CALLBACK;

RCTLIB_WCDMA_HSUPA_NSFT_FetchResult_CALLBACK HSUPA_NSFT_FetchResult_CALLBACK;

RCTLIB_WCDMA_GET_SpecificSettings_CALLBACK GET_SpecificSettings_CALLBACK;

}RCTLIB_WCDMA_CALLBACKS_CFG_T;
```

6.4.1.4 TDSCDMA Callback Function Definitions

Callback Register Function Definition:

```
int __stdcall RCTLIB_SetTDSCDMAInstrumentInstance(RCTLIB_TDSCDMA_CALLBACKS_CFG_T * instCallbacks);
```

Description:

Set TDSCDMA instrument control callback functions.

Parameter:

Table 6-131 TDSCDMA callback function parameter

Parameter	Direction	Parameter
instCallbacks	IN	the structure to get TDSCDMA instrument's special settings

Related Callback Function Definitions:

```
typedef int (*RCTLIB_TDSCDMA_InstrumentInit_CALLBACK)(void);
```

```
typedef int (*RCTLIB_TDSCDMA_ConfigTesterCommonBeforeCal_CALLBACK)(void);
```

```
typedef int (*RCTLIB_TDSCDMA_ConfigTesterBeforeAFC_CALLBACK)(const S_RCTLIB_TDSCDMA_AFC_TESTER_CONFIG_BEFORE_CAL_T *p_rConfig);
```

```

typedef int (*RCTLIB_TDSCDMA_MeasureAFC_CALLBACK)(double *p_dFreqOffset);

typedef          int          (*RCTLIB_TDSCDMA_ConfigTesterBeforeAGC_CALLBACK)(const
S_RCTLIB_TDSCDMA_AGC_TESTER_CONFIG_BEFORE_CAL_T *p_rConfig);

typedef int (*RCTLIB_TDSCDMA_AGC_ChangeCellPower_CALLBACK)(double cell_power);

typedef int (*RCTLIB_TDSCDMA_AGC_ChangeCellBand_CALLBACK)(unsigned int band);

typedef int (*RCTLIB_TDSCDMA_AGC_ChangeChannel_CALLBACK)(unsigned int uarfcn);

typedef          int          (*RCTLIB_TDSCDMA_ConfigTesterBeforeAPC_CALLBACK)(const
S_RCTLIB_TDSCDMA_APC_TESTER_CONFIG_BEFORE_CAL_T *p_rConfig);

typedef int (*RCTLIB_TDSCDMA_APC_ChangeCellBand_CALLBACK)(unsigned int band);

typedef          int          (*RCTLIB_TDSCDMA_APC_MeasurePower_CALLBACK)(S_RCTLIB_TDSCDMA_APC_MEASUREMENT_PARAM_T
*param);

typedef          int          (*RCTLIB_TDSCDMA_ConfigTesterBeforeFHC_CALLBACK)(const
S_RCTLIB_TDSCDMA_FHC_TESTER_CONFIG_BEFORE_CAL_T *p_rConfig);

typedef          int          (*RCTLIB_TDSCDMA_FHC_StartIteration_CALLBACK)(const
S_RCTLIB_TDSCDMA_FHC_MEASUREMENT_PARAM_T *pSettings);

typedef          int          (*RCTLIB_TDSCDMA_FHC_FetchResult_CALLBACK)(const
S_RCTLIB_TDSCDMA_FHC_MEASUREMENT_PARAM_T          *pSettings,
S_RCTLIB_TDSCDMA_FHC_MEASUREMENT_RESULT_T* pResult);

typedef          int          (*RCTLIB_TDSCDMA_ConfigTesterCommonBeforeNSFT_CALLBACK)(S_RCTLIB_TD_NSFT_TESTER_COMMON_CO
NFIG_T *common_cfg);

typedef          int          (*RCTLIB_TDSCDMA_ConfigTesterForNSFT_CALLBACK)(const
S_RCTLIB_TD_NSFT_TESTER_CONFIG_T *cfg);

typedef          int          (*RCTLIB_TDSCDMA_MeasureTPCForNSFT_CALLBACK)(const
S_RCTLIB_TD_NSFT_TPC_CONFIG_ENTRY_T *req, S_RCTLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T *res);

typedef          int          (*RCTLIB_TDSCDMA_ConfigTesterForNSFTBer_CALLBACK)(const
S_RCTLIB_TD_NSFT_BER_CONFIG_ENTRY_T *cfg);

typedef          int          (*RCTLIB_TDSCDMA_MeasureLBerForNSFT_CALLBACK)(S_RCTLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T
*res);

typedef          int          (*RCTLIB_TDSCDMA_ConfigTesterCommonBeforeFT_CALLBACK)(S_RCTLIB_TD_NSFT_TESTER_COMMON_CONF
IG_T *common_cfg);

```

```
typedef int (*RCLIB_TDSCDMA_ConfigTesterForFT_CALLBACK)(const S_RCLIB_TD_NSFT_TESTER_CONFIG_T
*cfg);

typedef int (*RCLIB_TDSCDMA_MeasureTPCForFT_CALLBACK)(const
S_RCLIB_TD_NSFT_TPC_CONFIG_ENTRY_T *req, S_RCLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T *res);

typedef int (*RCLIB_TDSCDMA_ConfigTesterForFTBer_CALLBACK)(const
S_RCLIB_TD_NSFT_BER_CONFIG_ENTRY_T *cfg);

typedef int (*RCLIB_TDSCDMA_MeasureLBerForFT_CALLBACK)(S_RCLIB_TDA_NSFT_MEASURE_RESULT_ENTRY_T *res);

typedef int (*RCLIB_TDSCDMA_GET_SpecificSettings_CALLBACK)(S_RCLIB_TDSCDMA_SPECIFIC_SETTINGS_T
*pSettings);
```

Related Data Structure:

```
typedef struct
{
    RCLIB_TDSCDMA_InstrumentInit_CALLBACK InstrumentInit_CALLBACK;
    RCLIB_TDSCDMA_ConfigTesterCommonBeforeCal_CALLBACK ConfigTesterCommonBeforeCal_CALLBACK;
    RCLIB_TDSCDMA_ConfigTesterBeforeAFC_CALLBACK ConfigTesterBeforeAFC_CALLBACK;
    RCLIB_TDSCDMA_MeasureAFC_CALLBACK MeasureAFC_CALLBACK;
    RCLIB_TDSCDMA_ConfigTesterBeforeAGC_CALLBACK ConfigTesterBeforeAGC_CALLBACK;
    RCLIB_TDSCDMA_AGC_ChangeCellPower_CALLBACK AGC_ChangeCellPower_CALLBACK;
    RCLIB_TDSCDMA_AGC_ChangeCellBand_CALLBACK AGC_ChangeCellBand_CALLBACK;
    RCLIB_TDSCDMA_AGC_ChangeChannel_CALLBACK AGC_ChangeChannel_CALLBACK;
    RCLIB_TDSCDMA_ConfigTesterBeforeAPC_CALLBACK ConfigTesterBeforeAPC_CALLBACK;
    RCLIB_TDSCDMA_APC_ChangeCellBand_CALLBACK APC_ChangeCellBand_CALLBACK;
    RCLIB_TDSCDMA_APC_MeasurePower_CALLBACK APC_MeasurePower_CALLBACK;
    RCLIB_TDSCDMA_ConfigTesterBeforeFHC_CALLBACK ConfigTesterBeforeFHC_CALLBACK;
    RCLIB_TDSCDMA_FHC_StartIteration_CALLBACK FHC_StartIteration_CALLBACK;
    RCLIB_TDSCDMA_FHC_FetchResult_CALLBACK FHC_FetchResult_CALLBACK;
    RCLIB_TDSCDMA_ConfigTesterCommonBeforeNSFT_CALLBACK
        ConfigTesterCommonBeforeNSFT_CALLBACK;
    RCLIB_TDSCDMA_ConfigTesterForNSFT_CALLBACK ConfigTesterForNSFT_CALLBACK;
    RCLIB_TDSCDMA_MeasureTPCForNSFT_CALLBACK MeasureTPCForNSFT_CALLBACK;
```

```
RCTLIB_TDSCDMA_ConfigTesterForNSFTBer_CALLBACK ConfigTesterForNSFTBer_CALLBACK;
RCTLIB_TDSCDMA_MeasureLBerForNSFT_CALLBACK MeasureLBerForNSFT_CALLBACK;
RCTLIB_TDSCDMA_ConfigTesterCommonBeforeFT_CALLBACK ConfigTesterCommonBeforeFT_CALLBACK;
RCTLIB_TDSCDMA_ConfigTesterForFT_CALLBACK ConfigTesterForFT_CALLBACK;
RCTLIB_TDSCDMA_MeasureTPCForFT_CALLBACK MeasureTPCForFT_CALLBACK;
RCTLIB_TDSCDMA_ConfigTesterForFTBer_CALLBACK ConfigTesterForFTBer_CALLBACK;
RCTLIB_TDSCDMA_MeasureLBerForFT_CALLBACK MeasureLBerForFT_CALLBACK;
RCTLIB_TDSCDMA_GET_SpecificSettings_CALLBACK GET_SpecificSettings_CALLBACK;
}RCTLIB_TDSCDMA_CALLBACKS_CFG_T;
```

6.4.1.5 LTE Callback Function Definitions

Callback Register Function Definition:

```
int __stdcall RCTLIB_SetLTEInstrumentInstance(RCTLIB_LTE_CALLBACKS_CFG_T* instCallbacks);
```

Description:

Set LTE instrument control callback functions.

Parameter:

Table 6-132 LTE Callback function parameter

Parameter	Direction	Parameter
instCallbacks	IN	the structure to get LTE instrument's special settings

Related Callback Function Definitions:

```
typedef int (*RCTLIB_LTE_InstrumentInit_CALLBACK)(void);
typedef int (*RCTLIB_LTE_ConfigDefaultSettings_CALLBACK)();
typedef int (*RCTLIB_LTE_AFC_PreSetting_CALLBACK)(const S_RCTLIB_LTE_AFC_PRESETTINGS_T* pSettings);
typedef int (*RCTLIB_LTE_AFC_Initiate_CALLBACK)();
typedef int (*RCTLIB_LTE_AFC_FetchResult_CALLBACK)(double* freqerror);
typedef int (*RCTLIB_LTE_AGC_PreSetting_CALLBACK)();
typedef int (*RCTLIB_LTE_AGC_ChangeCellBand_CALLBACK)(unsigned int band);
typedef int (*RCTLIB_LTE_AGC_ChangeFrequency_CALLBACK)(const S_RCTLIB_LTE_FREQUENCY_T* pSettings);
```

```
typedef int (*RCTLIB_LTE_AGC_ChangeCellPower_CALLBACK)(double cellPower);

typedef int (*RCTLIB_LTE_APC_PreSetting_CALLBACK)(const S_RCTLIB_LTE_APC_PRESETTINGS_T* pSettings);

typedef int (*RCTLIB_LTE_APC_ChangeCellBand_CALLBACK)(unsigned int band);

typedef int (*RCTLIB_LTE_APC_ChangeUIFrequency_CALLBACK)(const S_RCTLIB_LTE_FREQUENCY_T* pSettings);

typedef int (*RCTLIB_LTE_APC_ChangeExpectedPower_CALLBACK)(int expectedPower);

typedef int (*RCTLIB_LTE_APC_Initiate_CALLBACK)();

typedef int (*RCTLIB_LTE_APC_FetchResult_CALLBACK)(double* outputPower);

typedef int (*RCTLIB_LTE_FHC_PreSetting_CALLBACK)(const S_RCTLIB_LTE_FHC_PRESETTINGS_T* pSettings);

typedef int (*RCTLIB_LTE_FHC_StartIteration_CALLBACK)(const S_RCTLIB_LTE_FHC_MEASUREMENT_PARAM_T* pSettings);

typedef int (*RCTLIB_LTE_FHC_FetchResult_CALLBACK)(const S_RCTLIB_LTE_FHC_MEASUREMENT_PARAM_T* pSettings, S_RCTLIB_LTE_FHC_MEASUREMENT_RESULT_T* pResult);

typedef int (*RCTLIB_LTE_NSFT_PreSetting_CALLBACK)(const S_RCTLIB_LTE_NSFT_PRESETTINGS_T* pSettings);

typedef int (*RCTLIB_LTE_NSFT_TX_ListMode_CALLBACK)(const S_RCTLIB_LTE_NSFT_TX_CONFIG_T* pSettings);

typedef int (*RCTLIB_LTE_NSFT_TX_FetchListMode_CALLBACK)(const S_RCTLIB_LTE_NSFT_TX_CONFIG_T* pSettings, S_RCTLIB_LTE_NSFT_TX_RESULT_T* pResult);

typedef int (*RCTLIB_LTE_NSFT_ChangeCellPower_CALLBACK)(double d_cell_power);

typedef int (*RCTLIB_LTE_NSFT_InitiateBER_CALLBACK)(const S_RCTLIB_LTE_NSFT_RX_BER_T* pSettings);

typedef int (*RCTLIB_LTE_NSFT_GetCableLoss_CALLBACK)(unsigned int band, double* dOutputAttenuation, double* dInputAttenuation);

typedef int (*RCTLIB_LTE_GET_SpecificSettings_CALLBACK)(S_RCTLIB_LTE_SPECIFIC_SETTINGS_T* pSettings);
```

Related Data Structure:

```
typedef struct
{
    RCTLIB_LTE_InstrumentInit_CALLBACK InstrumentInit_CALLBACK;
    RCTLIB_LTE_ConfigDefaultSettings_CALLBACK ConfigDefaultSettings_CALLBACK;
    RCTLIB_LTE_AFC_PreSetting_CALLBACK AFC_PreSetting_CALLBACK;
    RCTLIB_LTE_AFC_Initiate_CALLBACK AFC_Initiate_CALLBACK;
    RCTLIB_LTE_AFC_FetchResult_CALLBACK AFC_FetchResult_CALLBACK;
    RCTLIB_LTE_AGC_PreSetting_CALLBACK AGC_PreSetting_CALLBACK;
```

```

RCTLIB_LTE_AGC_ChangeCellBand_CALLBACK AGC_ChangeCellBand_CALLBACK;
RCTLIB_LTE_AGC_ChangeFrequency_CALLBACK AGC_ChangeFrequency_CALLBACK;
RCTLIB_LTE_AGC_ChangeCellPower_CALLBACK AGC_ChangeCellPower_CALLBACK;
RCTLIB_LTE_APC_PreSetting_CALLBACK APC_PreSetting_CALLBACK;
RCTLIB_LTE_APC_ChangeCellBand_CALLBACK APC_ChangeCellBand_CALLBACK;
RCTLIB_LTE_APC_ChangeUIFrequency_CALLBACK APC_ChangeUIFrequency_CALLBACK;
RCTLIB_LTE_APC_ChangeExpectedPower_CALLBACK APC_ChangeExpectedPower_CALLBACK;
RCTLIB_LTE_APC_Initiate_CALLBACK APC_Initiate_CALLBACK;
RCTLIB_LTE_APC_FetchResult_CALLBACK APC_FetchResult_CALLBACK;
RCTLIB_LTE_FHC_PreSetting_CALLBACK FHC_PreSetting_CALLBACK;
RCTLIB_LTE_FHC_StartIteration_CALLBACK FHC_StartIteration_CALLBACK;
RCTLIB_LTE_FHC_FetchResult_CALLBACK FHC_FetchResult_CALLBACK;
RCTLIB_LTE_NSFT_PreSetting_CALLBACK NSFT_PreSetting_CALLBACK;
RCTLIB_LTE_NSFT_TX_ListMode_CALLBACK NSFT_TX_ListMode_CALLBACK;
RCTLIB_LTE_NSFT_TX_FetchListMode_CALLBACK NSFT_TX_FetchListMode_CALLBACK;
RCTLIB_LTE_NSFT_ChangeCellPower_CALLBACK NSFT_ChangeCellPower_CALLBACK;
RCTLIB_LTE_NSFT_InitiateBER_CALLBACK NSFT_InitiateBER_CALLBACK;
RCTLIB_LTE_NSFT_GetCableLoss_CALLBACK NSFT_GetCableLoss_CALLBACK;
RCTLIB_LTE_GET_SpecificSettings_CALLBACK GET_SpecificSettings_CALLBACK;
} RCTLIB_LTE_CALLBACKS_CFG_T;

```

6.4.2 Sample Code

```

#include "stdafx.h"
#include "METAInstrumentLibrary.h"

/*****
* VISA GPIB include headers
*****/

#include "visatype.h"
#include "visa.h"

```

```
static int cableloss[5];

static ViSession m_ViRM;

static ViSession m_ViHandle;

static char m_ViCmdBuffer[1024];


#include <string>

#include <windows.h>


static int RCT_InitializeConfiguration(
    const char *cfg_file_path /// full path to the CFG file
)
{
    printf("RCT_InitializeConfiguration\n");
    int i;
    /// setting GSM cable loss
    for(i=0;i<5;i++)
    {
        cableloss[i] = -1.0;
    }
    return 0;
}

static int ConnectRCT(void)
{
    printf("ConnectRCT\n");
    int m_ViStat;
    if(m_ViRM == NULL)
    {
        // allocate RM
        m_ViStat = viOpenDefaultRM(&m_ViRM);
    }
}
```

```

        if(m_ViStat != VI_SUCCESS && m_ViStat != VI_WARN_CONFIG_NLOADED)
        {
            return m_ViStat;
        }
    }

    if(m_ViHandle == NULL)
    {
        // allocate Handle
        m_ViStat = viOpen(m_ViRM, (char*)"GPIB0::15::INSTR", VI_NULL, VI_NULL, &m_ViHandle);

        if(m_ViStat != VI_SUCCESS &&
            m_ViStat != VI_WARN_CONFIG_NLOADED &&
            m_ViStat != VI_SUCCESS_DEV_NPRESENT)
        {
            DisconnectRCT();
            return m_ViStat;
        }
    }

    m_ViStat = viClear(m_ViHandle);
    if(m_ViStat != VI_SUCCESS)
    {
        return m_ViStat;
    }

    return VI_SUCCESS;
}

static int DisconnectRCT(void)
{
    printf("DisconnectRCT\n");
}

```



```

int m_ViStat;

if (NULL !=m_ViHandle)
{
    m_ViStat = viClose(m_ViHandle);
    if(m_ViStat != VI_SUCCESS) return m_ViStat;
    m_ViHandle = NULL;
}

if( NULL != m_ViRM)
{
    m_ViStat = viClose(m_ViRM);
    if(m_ViStat != VI_SUCCESS) return m_ViStat;
    m_ViRM = NULL;
}

return VI_SUCCESS;
}

static int RCT_ConfigApplicationFormat(unsigned int format)
{
    printf("RCT_ConfigApplicationFormat\n");
    return VI_SUCCESS;
}

static int RCT_ConfigOperatingMode(unsigned int OperatingMode)
{
    printf("RCT_ConfigOperatingMode\n");
    return VI_SUCCESS;
}

/*****
* Setup callback functions

```

```

*****/

void Setup_Callback(RCTLIB_COMMON_CALLBACKS_CFG_T *pCommonCfg)
{
    printf("Setup_Callback\n");
    pCommonCfg->InitializeConfiguration_CALLBACK = &RCT_InitializeConfiguration;
    pCommonCfg->ConnectRCT_CALLBACK = &ConnectRCT;
    pCommonCfg->DisconnectRCT_CALLBACK = &DisconnectRCT;
    pCommonCfg->ConfigApplicationFormat_CALLBACK = &RCT_ConfigApplicationFormat;
    pCommonCfg->ConfigOperatingMode_CALLBACK = &RCT_ConfigOperatingMode;
}

/*****
 * Init META Calibration Library
 *****/

// Please reference 5.2 Start calibration via META Calibration Library
/*****
 * Set CFG/INI/CAL/Result path
 *****/

// Please reference 5.2 Start calibration via META Calibration Library
/*****
 * Set Customized Device type
 *****/

calibrationCfg.i_device_type = RCTLIB_DEVICE_CALLBACK;
calibrationCfg.i_device_type_wcdma = RCTLIB_DEVICE_CALLBACK;
calibrationCfg.i_device_type_tdscdma = RCTLIB_DEVICE_CALLBACK;
calibrationCfg.resetTester = false;
/*****
 * Set GGE calibration item
 *****/

memset(&(calibrationCfg.ggeCalibrationItems), 0, sizeof(S_METACalibrationLibrary_GGE_CAL_ITEM_T));

```

```
memset(&(calibrationCfg.ggeNsftItems), 0, sizeof(S_METACalibrationLibrary_GGE_NSFT_ITEM_T));

/*****

* Set WCDMA calibration item

*****/

memset(&(calibrationCfg.wcdmaCalibrationItems), 0, sizeof(S_METACalibrationLibrary_WCDMA_CAL_ITEM_T));
memset(&(calibrationCfg.wcdmaNsftItems), 0, sizeof(S_METACalibrationLibrary_WCDMA_NSFT_ITEM_T));

// enable WCDMA Temperature ADC calibration
calibrationCfg.wcdmaCalibrationItems.b_wcdma_tadc_cal = true;

/*****

* Set TDSCDMA calibration item

*****/

memset(&(calibrationCfg.tdscdmaCalibrationItems), 0, sizeof(S_METACalibrationLibrary_TD_CAL_ITEM_T));
memset(&(calibrationCfg.tdscdmaNsftItems), 0, sizeof(S_METACalibrationLibrary_TD_NSFT_ITEM_T));
S_METACalibrationLibrary_STATUS_T sta;

/*****

* Set Callback functions

*****/

RTLIB_COMMON_CALLBACKS_CFG_T CommonCfg;
Setup_Callback(&CommonCfg)
RTLIB_SetCommonInstrumentInstance(&CommonCfg);

/*****

* Start calibration

*****/

E_METACalibrationLibrary_RESULT m_rMetaLibResult = METACalibrationLibrary_Start(metaHandle,
&calibrationCfg,
&sta,
m_piMetaLibStop);

if(m_rMetaLibResult != E_METACalibrationLibrary_RESULT_SUCCESS)
```

```
{
    // error handling
    /* Get status code */
    sta.m_eStatusCode;
    // ...
    /* Get status string */
    sta.m_cErrorMsgbuf;
    // ...
}
```

7 Appendix

7.1 The mapping between calibration library items and META tool factory UI calibration items

7.1.1 GSM/EDGE calibration items

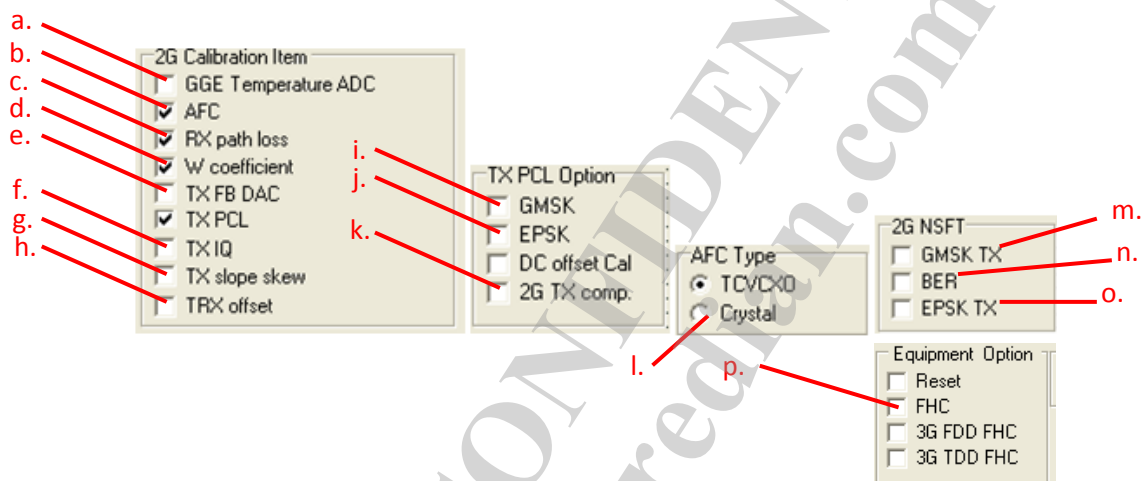


Figure 7-1 GSM/EDGE calibration items

a. b_gge_tadc_cal	i. b_gge_apc_cal
b. b_gge_afc_cal	j. b_gge_edge_apc_cal
c. b_gge_agc_cal	k. b_gge_tpc_subband_cal
d. b_gge_agc_w_cal	l. b_gge_cap_id_cal /* Crystal only */
e. b_gge_fb_dac_cal	m. b_gge_nsft_gmsk
f. b_gge_txiq_cal	n. b_gge_nsft_ber
g. b_gge_slope_skew_cal	o. b_gge_nsft_epsk
h. b_gge_afc_trx_offset_cal	p. b_gge_fhc_cal

7.1.2 WCDMA calibration items

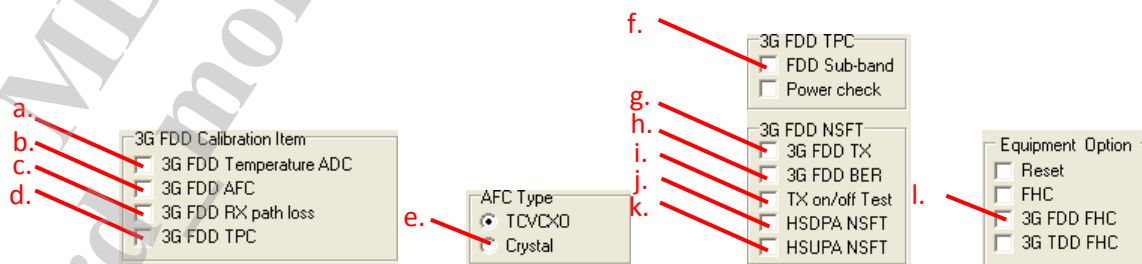


Figure 7-2 WCDMA calibration items

q. b_wcdma_tadc_cal	w. b_wcdma_nsft_tpc
r. b_wcdma_afc_cal	x. b_wcdma_nsft_ber
s. b_wcdma_agc_cal	y. b_wcdma_nsft_prach
t. b_wcdma_apc_cal	z. b_wcdma_hsdpa_nsft
u. b_wcdma_dcxo_afc_cal /* Crystal only */	aa. b_wcdma_hsupa_nsft
v. b_wcdma_tpc_subband_cal	bb. b_wcdma_fhc_cal

7.1.3 TD-SCDMA calibration items

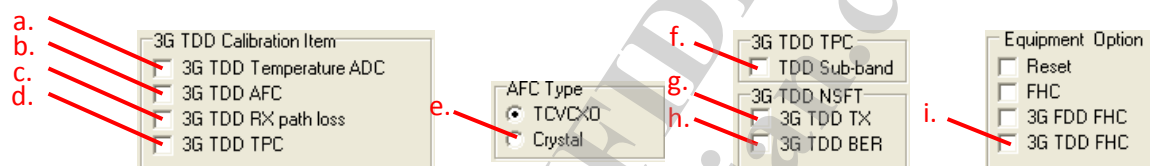


Figure 7-3 TD-SCDMA calibration items

cc. b_tda_tadc_cal	ii. b_tda_nsft_tpc
dd. b_tda_afc_cal	jj. b_tda_nsft_ber
ee. b_tda_rx_path_loss_cal	kk. b_tda_fhc_cal
ff. b_tda_tpc_cal	
gg. b_tda_cap_id_cal /* Crystal only */	
hh. b_tda_tpc_subband_cal	

7.1.4 LTE calibration items

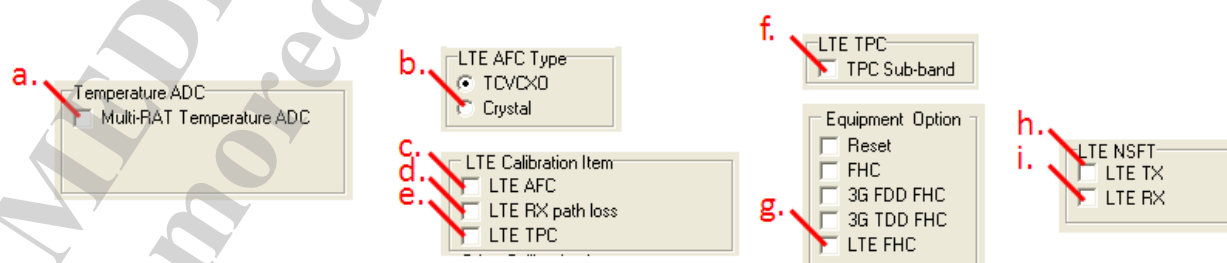


Figure 7-4 LTE calibration items



ll. b_lte_tadc_cal	rr. b_lte_fhc_cal
mm. b_lte_cap_id_cal /* Crystal only */	ss. b_lte_nsft_tx
nn. b_lte_afc_cal	tt. b_lte_nsft_rx
oo. b_lte_rx_path_loss_cal	
pp. b_lte_tpc_cal	
qq. b_lte_tpc_subband_cal	

MediaTek Confidential

© 2017 MediaTek Inc.

Classification:Confidential B

This document contains information that is proprietary to MediaTek Inc.
Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.