
RsCMPX_UwbMeas

Release 5.0.20.22

Rohde & Schwarz

Apr 19, 2024

CONTENTS:

1	Revision History	3
1.1	RsCMPX_UwbMeas	3
1.1.1	Version history	3
2	Getting Started	5
2.1	Introduction	5
2.2	Installation	7
2.3	Finding Available Instruments	8
2.4	Initiating Instrument Session	9
2.5	Plain SCPI Communication	12
2.6	Error Checking	14
2.7	Exception Handling	14
2.8	Transferring Files	16
2.9	Writing Binary Data	16
2.10	Transferring Big Data with Progress	17
2.11	Multithreading	18
2.12	Logging	21
3	Enums	25
3.1	Connector	25
3.2	MaxSpecPowLen	26
3.3	PhrDataRate	26
3.4	PpduMode	26
3.5	Repeat	26
3.6	ResourceState	26
3.7	Result	27
3.8	ResultStatus2	27
3.9	SignalSlope	27
3.10	StopCondition	27
3.11	StsSegmentLen	27
3.12	TargetMainState	28
3.13	TargetSyncState	28
4	RepCaps	29
4.1	Instance (Global)	29
4.2	Area	29
4.3	Ppdu	29
4.4	Record	30
5	Examples	31

6	RsCMPX_UwbMeas API Structure	33
6.1	Catalog	36
6.1.1	UwbMeas	36
6.2	Configure	36
6.2.1	UwbMeas	37
6.2.1.1	MultiEval	37
6.2.1.1.1	Modulation	41
6.2.1.1.1.1	Limit	41
6.2.1.1.1.2	CcError	42
6.2.1.1.1.3	Foffset	43
6.2.1.1.1.4	Phr	43
6.2.1.1.1.5	Nrmse	44
6.2.1.1.1.6	Plevel	45
6.2.1.1.1.7	PmlWidth	45
6.2.1.1.1.8	Psdu	46
6.2.1.1.1.9	Nrmse	46
6.2.1.1.1.10	Shr	47
6.2.1.1.1.11	Nrmse	47
6.2.1.1.1.12	SlPeak	48
6.2.1.1.1.13	SmAccuracy	49
6.2.1.1.1.14	Sts	50
6.2.1.1.1.15	Nrmse	50
6.2.1.1.2	MprFrequency	51
6.2.1.1.3	Phr	51
6.2.1.1.3.1	Bitrate<Record>	51
6.2.1.1.4	PhrRate	52
6.2.1.1.5	Pmask	53
6.2.1.1.5.1	Limit	53
6.2.1.1.5.2	Area	54
6.2.1.1.5.3	Lower	54
6.2.1.1.5.4	Area<Area>	55
6.2.1.1.5.5	Upper	56
6.2.1.1.5.6	Area<Area>	56
6.2.1.1.6	Ppdu	57
6.2.1.1.7	PpLength	58
6.2.1.1.8	PrfMode	59
6.2.1.1.9	Psdu	60
6.2.1.1.9.1	Bitrate<Record>	60
6.2.1.1.10	PsFormat	61
6.2.1.1.11	Result	61
6.2.1.1.12	Spectrum	63
6.2.1.1.13	StSegments	64
6.2.1.1.14	StsGap	65
6.2.1.1.14.1	Chip	66
6.2.1.1.15	StsLength	66
6.2.1.1.16	TsMask	67
6.2.1.1.16.1	Limit	67
6.2.1.1.16.2	Area<Area>	67
6.2.1.1.16.3	TdbBandwidth	69
6.2.1.2	RfSettings	70
6.2.1.2.1	Frequency	72
6.2.1.2.1.1	Range	72
6.3	Route	73
6.3.1	UwbMeas	73

6.3.1.1	RfSettings	74
6.4	Trigger	75
6.4.1	UwbMeas	75
6.4.1.1	MultiEval	75
6.4.1.1.1	Catalog	78
6.5	UwbMeas	78
6.5.1	MultiEval	79
6.5.1.1	Ddecoding	80
6.5.1.1.1	Clength	81
6.5.1.1.2	Content	82
6.5.1.1.3	RsParity	82
6.5.1.2	Modulation	83
6.5.1.2.1	Average	84
6.5.1.2.2	CcError	85
6.5.1.2.2.1	Average	85
6.5.1.2.2.2	Current<Ppdu>	87
6.5.1.2.2.3	Extreme	88
6.5.1.2.2.4	StandardDev	89
6.5.1.2.3	Cevm	90
6.5.1.2.3.1	Average	91
6.5.1.2.3.2	Current<Ppdu>	91
6.5.1.2.3.3	Extreme	93
6.5.1.2.3.4	StandardDev	93
6.5.1.2.4	CpJitter	94
6.5.1.2.4.1	Average	94
6.5.1.2.4.2	Current<Ppdu>	95
6.5.1.2.4.3	Extreme	96
6.5.1.2.4.4	StandardDev	97
6.5.1.2.5	CtJitter	98
6.5.1.2.5.1	Average	98
6.5.1.2.5.2	Current<Ppdu>	99
6.5.1.2.5.3	Extreme	100
6.5.1.2.5.4	StandardDev	101
6.5.1.2.6	Current<Ppdu>	102
6.5.1.2.7	Extreme	103
6.5.1.2.8	Foffset	105
6.5.1.2.8.1	Average	105
6.5.1.2.8.2	Current<Ppdu>	106
6.5.1.2.8.3	Extreme	108
6.5.1.2.8.4	StandardDev	109
6.5.1.2.9	Fofh	110
6.5.1.2.9.1	Average	110
6.5.1.2.9.2	Current<Ppdu>	111
6.5.1.2.9.3	Extreme	112
6.5.1.2.9.4	StandardDev	113
6.5.1.2.10	IqOffset	114
6.5.1.2.10.1	Average	114
6.5.1.2.10.2	Current<Ppdu>	115
6.5.1.2.10.3	Extreme	116
6.5.1.2.10.4	StandardDev	117
6.5.1.2.11	Nmse	117
6.5.1.2.11.1	Average	118
6.5.1.2.11.2	Current<Ppdu>	119
6.5.1.2.11.3	Extreme	120

6.5.1.2.11.4	StandardDev	121
6.5.1.2.12	Otolerance	121
6.5.1.2.13	Phr	122
6.5.1.2.13.1	Nrmse	122
6.5.1.2.13.2	Average	123
6.5.1.2.13.3	Current<Ppdu>	124
6.5.1.2.13.4	Extreme	125
6.5.1.2.13.5	StandardDev	127
6.5.1.2.13.6	Plevel	128
6.5.1.2.13.7	Average	128
6.5.1.2.13.8	Current<Ppdu>	129
6.5.1.2.13.9	Maximum	130
6.5.1.2.13.10	Minimum	131
6.5.1.2.13.11	StandardDev	132
6.5.1.2.14	Plevel	133
6.5.1.2.14.1	Average	134
6.5.1.2.14.2	Current<Ppdu>	135
6.5.1.2.14.3	Maximum	137
6.5.1.2.14.4	Minimum	138
6.5.1.2.14.5	StandardDev	140
6.5.1.2.15	PmlWidth	141
6.5.1.2.15.1	Average	141
6.5.1.2.15.2	Current<Ppdu>	142
6.5.1.2.15.3	Extreme	144
6.5.1.2.15.4	StandardDev	145
6.5.1.2.16	Psdu	146
6.5.1.2.16.1	Nrmse	146
6.5.1.2.16.2	Average	147
6.5.1.2.16.3	Current<Ppdu>	148
6.5.1.2.16.4	Extreme	149
6.5.1.2.16.5	StandardDev	150
6.5.1.2.16.6	Plevel	151
6.5.1.2.16.7	Average	152
6.5.1.2.16.8	Current<Ppdu>	152
6.5.1.2.16.9	Maximum	154
6.5.1.2.16.10	Minimum	155
6.5.1.2.16.11	StandardDev	156
6.5.1.2.17	Rmarker	157
6.5.1.2.17.1	Average	157
6.5.1.2.17.2	Current<Ppdu>	158
6.5.1.2.17.3	Extreme	159
6.5.1.2.17.4	StandardDev	160
6.5.1.2.18	Sevm	161
6.5.1.2.18.1	Average	161
6.5.1.2.18.2	Current<Ppdu>	162
6.5.1.2.18.3	Extreme	163
6.5.1.2.18.4	StandardDev	164
6.5.1.2.19	Sfd	165
6.5.1.2.19.1	PIpolarity	165
6.5.1.2.19.2	Current<Ppdu>	165
6.5.1.2.19.3	Extreme	166
6.5.1.2.20	Shr	167
6.5.1.2.20.1	Nrmse	167
6.5.1.2.20.2	Average	168

6.5.1.2.20.3	Current<Ppdu>	169
6.5.1.2.20.4	Extreme	170
6.5.1.2.20.5	StandardDev	172
6.5.1.2.21	SlPeak	173
6.5.1.2.21.1	Average	173
6.5.1.2.21.2	Current<Ppdu>	174
6.5.1.2.21.3	Extreme	176
6.5.1.2.21.4	StandardDev	177
6.5.1.2.22	SmAccuracy	178
6.5.1.2.22.1	Average	178
6.5.1.2.22.2	Current<Ppdu>	179
6.5.1.2.22.3	Extreme	181
6.5.1.2.22.4	StandardDev	182
6.5.1.2.23	SpJitter	183
6.5.1.2.23.1	Average	183
6.5.1.2.23.2	Current<Ppdu>	184
6.5.1.2.23.3	Extreme	185
6.5.1.2.23.4	StandardDev	186
6.5.1.2.24	StandardDev	187
6.5.1.2.25	StJitter	188
6.5.1.2.25.1	Average	189
6.5.1.2.25.2	Current<Ppdu>	190
6.5.1.2.25.3	Extreme	191
6.5.1.2.25.4	StandardDev	192
6.5.1.2.26	Sts	192
6.5.1.2.26.1	Nrmse	193
6.5.1.2.26.2	Average	193
6.5.1.2.26.3	Current<Ppdu>	194
6.5.1.2.26.4	Extreme	196
6.5.1.2.26.5	StandardDev	197
6.5.1.2.26.6	Plevel	198
6.5.1.2.26.7	Average	198
6.5.1.2.26.8	Current<Ppdu>	199
6.5.1.2.26.9	Maximum	200
6.5.1.2.26.10	Minimum	202
6.5.1.2.26.11	StandardDev	203
6.5.1.2.26.12	PIPolarity	204
6.5.1.2.26.13	Current<Ppdu>	204
6.5.1.2.26.14	Extreme	205
6.5.1.2.27	Sync	206
6.5.1.2.27.1	PIPolarity	206
6.5.1.2.27.2	Current<Ppdu>	206
6.5.1.2.27.3	Extreme	208
6.5.1.3	Pmask	209
6.5.1.3.1	Lower	209
6.5.1.3.1.1	Area<Area>	209
6.5.1.3.1.2	Margin	210
6.5.1.3.2	Margin	211
6.5.1.3.2.1	Area<Area>	211
6.5.1.3.2.2	Current<Ppdu>	211
6.5.1.3.2.3	PrMonotonic	212
6.5.1.3.2.4	Current<Ppdu>	213
6.5.1.3.2.5	Extreme	214
6.5.1.3.3	Otolerance	215

6.5.1.3.4	Upper	215
6.5.1.3.4.1	Area<Area>	216
6.5.1.3.4.2	Margin	216
6.5.1.4	Power	217
6.5.1.4.1	Average	217
6.5.1.4.2	Current<Ppdu>	218
6.5.1.4.3	Dpower	220
6.5.1.4.3.1	Average	220
6.5.1.4.3.2	Current<Ppdu>	221
6.5.1.4.3.3	Maximum	222
6.5.1.4.3.4	Minimum	223
6.5.1.4.3.5	StandardDev	223
6.5.1.4.4	DpPower	224
6.5.1.4.4.1	Average	224
6.5.1.4.4.2	Current<Ppdu>	225
6.5.1.4.4.3	Maximum	226
6.5.1.4.4.4	Minimum	227
6.5.1.4.4.5	StandardDev	228
6.5.1.4.5	Maximum	229
6.5.1.4.6	Minimum	230
6.5.1.4.7	MsfPower	231
6.5.1.4.7.1	Average	231
6.5.1.4.7.2	Current<Ppdu>	232
6.5.1.4.7.3	Maximum	233
6.5.1.4.7.4	Minimum	234
6.5.1.4.7.5	StandardDev	234
6.5.1.4.8	MsPower	235
6.5.1.4.8.1	Average	235
6.5.1.4.8.2	Current<Ppdu>	236
6.5.1.4.8.3	Maximum	237
6.5.1.4.8.4	Minimum	238
6.5.1.4.8.5	StandardDev	239
6.5.1.4.9	PpdPeak	240
6.5.1.4.9.1	Average	240
6.5.1.4.9.2	Current<Ppdu>	241
6.5.1.4.9.3	Maximum	242
6.5.1.4.9.4	Minimum	243
6.5.1.4.9.5	StandardDev	243
6.5.1.4.10	Ppdu<Ppdu>	244
6.5.1.4.10.1	Average	245
6.5.1.4.10.2	Current	245
6.5.1.4.10.3	Maximum	246
6.5.1.4.10.4	Minimum	247
6.5.1.4.10.5	StandardDev	248
6.5.1.4.11	Ppower	249
6.5.1.4.11.1	Average	249
6.5.1.4.11.2	Current<Ppdu>	250
6.5.1.4.11.3	Maximum	251
6.5.1.4.11.4	Minimum	252
6.5.1.4.11.5	StandardDev	252
6.5.1.4.12	PpPower	253
6.5.1.4.12.1	Average	253
6.5.1.4.12.2	Current<Ppdu>	254
6.5.1.4.12.3	Maximum	255

6.5.1.4.12.4	Minimum	256
6.5.1.4.12.5	StandardDev	257
6.5.1.4.13	StandardDev	258
6.5.1.5	Sinfo	259
6.5.1.5.1	AsSymbols	260
6.5.1.5.2	Cindex	261
6.5.1.5.3	CsLength	261
6.5.1.5.4	Dlength	262
6.5.1.5.5	Dppdu	263
6.5.1.5.6	Drate	264
6.5.1.5.7	FcsCheck	264
6.5.1.5.8	Phr	265
6.5.1.5.8.1	AsSymbols	265
6.5.1.5.8.2	Bitrate	266
6.5.1.5.8.3	Crc	267
6.5.1.5.9	Psdu	268
6.5.1.5.9.1	Crc	268
6.5.1.5.9.2	Length	269
6.5.1.5.10	PstGap	269
6.5.1.5.11	RaBit	270
6.5.1.5.12	ReBit	271
6.5.1.5.13	Sfd	272
6.5.1.5.14	SfdLength	272
6.5.1.6	State	273
6.5.1.6.1	All	274
6.5.1.7	StsSequence	275
6.5.1.7.1	Clength	275
6.5.1.7.2	Content	276
6.5.1.8	Trace	277
6.5.1.8.1	CpJitter	277
6.5.1.8.1.1	Average	277
6.5.1.8.1.2	Current<Ppdu>	278
6.5.1.8.1.3	Xvalues	279
6.5.1.8.2	CtJitter	280
6.5.1.8.2.1	Average	280
6.5.1.8.2.2	Current<Ppdu>	281
6.5.1.8.2.3	Xvalues	282
6.5.1.8.3	NcCorr	282
6.5.1.8.3.1	Average	283
6.5.1.8.3.2	Current<Ppdu>	283
6.5.1.8.3.3	Xvalues	285
6.5.1.8.4	Pmask	285
6.5.1.8.4.1	Current<Ppdu>	286
6.5.1.8.4.2	Xvalues	287
6.5.1.8.5	PowerVsTime	287
6.5.1.8.5.1	Maximum	288
6.5.1.8.5.2	Minimum	289
6.5.1.8.5.3	Xvalues	289
6.5.1.8.6	Sbwl	290
6.5.1.8.6.1	Average	290
6.5.1.8.6.2	Current<Ppdu>	291
6.5.1.8.6.3	Xvalues	292
6.5.1.8.7	Sbws	293
6.5.1.8.7.1	Average	293

6.5.1.8.7.2	Current<Ppdu>	294
6.5.1.8.7.3	Xvalues	295
6.5.1.8.8	SpJitter	295
6.5.1.8.8.1	Average	296
6.5.1.8.8.2	Current<Ppdu>	296
6.5.1.8.8.3	Xvalues	298
6.5.1.8.9	StJitter	298
6.5.1.8.9.1	Average	299
6.5.1.8.9.2	Current<Ppdu>	299
6.5.1.8.9.3	Xvalues	300
6.5.1.8.10	TsMask	301
6.5.1.8.10.1	Average	301
6.5.1.8.10.2	Current<Ppdu>	302
6.5.1.8.10.3	Xvalues	303
6.5.1.9	TsMask	304
6.5.1.9.1	Margin	304
6.5.1.9.1.1	Area<Area>	304
6.5.1.9.1.2	Average	305
6.5.1.9.1.3	Current<Ppdu>	305
6.5.1.9.1.4	Negativ	306
6.5.1.9.1.5	Average	307
6.5.1.9.1.6	Current<Ppdu>	308
6.5.1.9.1.7	Positiv	309
6.5.1.9.1.8	Average	310
6.5.1.9.1.9	Current<Ppdu>	311
6.5.1.9.2	Otolerance	312
6.5.1.9.3	TdbBandwidth	313
6.5.1.9.3.1	Frequency	313
6.5.1.9.3.2	Fhfl	314
6.5.1.9.3.3	PsPower	315
7	RsCMPX_UwbMeas Utilities	317
8	RsCMPX_UwbMeas Logger	323
9	RsCMPX_UwbMeas Events	325
10	Index	327
	Index	329



REVISION HISTORY

1.1 RsCMPX_UwbMeas

Rohde & Schwarz CMP200 Ultra Wideband Measurement RsCMPX_UwbMeas instrument driver.

Basic Hello-World code:

```
from RsCMPX_UwbMeas import *

instr = RsCMPX_UwbMeas('TCPIP::192.168.2.101::hislip0')
idn = instr.query('*IDN?')
print('Hello, I am: ' + idn)
```

Supported instruments: CMP200

The package is hosted here: <https://pypi.org/project/RsCMPX-UwbMeas/>

Documentation: <https://RsCMPX-UwbMeas.readthedocs.io/>

Examples: <https://github.com/Rohde-Schwarz/Examples/>

1.1.1 Version history

Latest release notes summary: Update for FW 5.0.20

Version 5.0.20

- Update for FW 5.0.20

Version 4.0.171

- Fixed documentation

Version 4.0.170

- Update for FW 4.0.170

Version 4.0.80

- Update of RsCMPX_UwbMeas to FW 4.0.80 from the complete FW package 7.10.0

Version 4.0.70

- Update of RsCMPX_UwbMeas to FW 4.0.70

Version 4.0.12

- Update of RsCMPX_UwbMeas to FW 4.0.12

Version 4.0.8

- First released version

GETTING STARTED

2.1 Introduction



RsCMPX_UwbMeas is a Python remote-control communication module for Rohde & Schwarz SCPI-based Test and Measurement Instruments. It represents SCPI commands as fixed APIs and hence provides SCPI autocompletion and helps you to avoid common string typing mistakes.

Basic example of the idea:

SCPI command:

SYSTem:REFeRence:FREQuency:SOURce

Python module representation:

writing:

```
driver.system.reference.frequency.source.set()
```

reading:

```
driver.system.reference.frequency.source.get()
```

Check out this example for RsCmpx-Base and RsCmpx-Gprf:

```
"""
# GitHub examples repository path: CMXP/Python/RsCmxx_ScpiPackages

Example on how to use the python RsCmx auto-generated instrument drivers for
RsCmpx_Base and RsCmpx_Gprf (Base and GPRF) in one script with shared VISA session.
"""

from RsCMPX_Base.RsCMPX_Base import RsCMPX_Base # install from pypi.org
from RsCMPX_Base import enums as base_enums
from RsCMPX_Base import repcap as base_repcap

from RsCMPX_Gprf.RsCMPX_Gprf import RsCMPX_Gprf # install from pypi.org
from RsCMPX_Gprf.CustomFiles.reliability import ReliabilityEventArgs
from RsCMPX_Gprf import enums as gprf_enums
from RsCMPX_Gprf import repcap as gprf_repcaps
```

(continues on next page)

(continued from previous page)

```

# CMX Base init
cmx_base = RsCMPX_Base('TCPIP::10.112.1.116', False, True)
print(f'CMX Base IND: {cmx_base.utilities.idn_string}')
print(f'CMX Instrument options:\n{", ".join(cmx_base.utilities.instrument_options)}')
cmx_base.utilities.visa_timeout = 5000 # default is 10000

# Sends OPC after each command
cmx_base.utilities.opc_query_after_write = False
# Checks for syst:err? after each command / query - default value after init is True
cmx_base.utilities.instrument_status_checking = True

# Self-test
self_test = cmx_base.utilities.self_test()
print(f'CMW self-test result: {self_test} - {"Passed" if self_test[0] == 0 else "Failed"}')
↪ ''')
# Reference Frequency Source
cmx_base.system.reference.frequency.source_set(base_enums.SourceIntExt.INTERNAL)

# CMX RsCMPX_Gprf Init - reuse the session of the cmx_base, rather than creating another_
↪ one
cmx_gprf = RsCMPX_Gprf.from_existing_session(cmx_base)
cmx_gprf.utilities.visa_timeout = 5000

# Driver's Interface reliability offers a convenient way of reacting on the return value_
↪ Reliability Indicator
cmx_gprf.reliability.ExceptionOnError = True # default is 10000

# Callback to use for the reliability indicator update events
def my_reliability_handler(event_args: ReliabilityEventArgs):
    print(f'GPRF Reliability updated.\nContext: {event_args.context}\nMessage:
↪ {event_args.message}')

# We register a callback for each change in the reliability indicator
cmx_gprf.reliability.on_update_handler = my_reliability_handler

# You can obtain the last value of the returned reliability
print(f"\nReliability last value: {cmx_gprf.reliability.last_value}, context '{cmx_gprf.
↪ reliability.last_context}', message: {cmx_gprf.reliability.last_message}")

# Close the sessions
cmx_gprf.close()
cmx_base.close()

```

Couple of reasons why to choose this module over plain SCPI approach:

- Type-safe API using typing module
- You can still use the plain SCPI communication
- You can select which VISA to use or even not use any VISA at all
- Initialization of a new session is straight-forward, no need to set any other properties

- Many useful features are already implemented - reset, self-test, opc-synchronization, error checking, option checking
- Binary data blocks transfer in both directions
- Transfer of arrays of numbers in binary or ASCII format
- File transfers in both directions
- Events generation in case of error, sent data, received data, chunk data (for big files transfer)
- Multithreading session locking - you can use multiple threads talking to one instrument at the same time
- Logging feature tailored for SCPI communication - different for binary and ascii data

2.2 Installation

RsCMPX_UwbMeas is hosted on pypi.org. You can install it with pip (for example, `pip.exe` for Windows), or if you are using Pycharm (and you should be :) direct in the Pycharm Packet Management GUI.

Preconditions

- Installed VISA. You can skip this if you plan to use only socket LAN connection. Download the Rohde & Schwarz VISA for Windows, Linux, Mac OS from [here](#)

Option 1 - Installing with pip.exe under Windows

- Start the command console: WinKey + R, type `cmd` and hit ENTER
- Change the working directory to the Python installation of your choice (adjust the user name and python version in the path):

```
cd c:\Users\John\AppData\Local\Programs\Python\Python37\Scripts
```

- Install with the command: `pip install RsCMPX_UwbMeas`

Option 2 - Installing in Pycharm

- In Pycharm Menu File->Settings->Project->Project Interpreter click on the '+' button on the top left (the last PyCharm version)
- Type RsCMPX_UwbMeas in the search box
- If you are behind a Proxy server, configure it in the Menu: File->Settings->Appearance->System Settings->HTTP Proxy

For more information about Rohde & Schwarz instrument remote control, check out our [Instrument Remote Control Web Series](#).

Option 3 - Offline Installation

If you are still reading the installation chapter, it is probably because the options above did not work for you - proxy problems, your boss saw the internet bill... Here are 6 step for installing the RsCMPX_UwbMeas offline:

- Download this python script (**Save target as**): `rsinstrument_offline_install.py` This installs all the preconditions that the RsCMPX_UwbMeas needs.
- Execute the script in your offline computer (supported is python 3.6 or newer)
- Download the RsCMPX_UwbMeas package to your computer from the pypi.org: https://pypi.org/project/RsCMPX_UwbMeas/#files to for example `c:\temp\`
- Start the command line WinKey + R, type `cmd` and hit ENTER
- Change the working directory to the Python installation of your choice (adjust the user name and python version in the path):

```
cd c:\Users\John\AppData\Local\Programs\Python\Python37\Scripts
```

- Install with the command: `pip install c:\temp\RsCMPX_UwbMeas-5.0.20.22.tar`

2.3 Finding Available Instruments

Like the pyvisa's ResourceManager, the RsCMPX_UwbMeas can search for available instruments:

```
"""
Find the instruments in your environment
"""

from RsCMPX_UwbMeas import *

# Use the instr_list string items as resource names in the RsCMPX_UwbMeas constructor
instr_list = RsCMPX_UwbMeas.list_resources("?*")
print(instr_list)
```

If you have more VISAs installed, the one actually used by default is defined by a secret widget called Visa Conflict Manager. You can force your program to use a VISA of your choice:

```
"""
Find the instruments in your environment with the defined VISA implementation
"""

from RsCMPX_UwbMeas import *

# In the optional parameter visa_select you can use for example 'rs' or 'ni'
# Rs Visa also finds any NRP-Zxx USB sensors
instr_list = RsCMPX_UwbMeas.list_resources('?*', 'rs')
print(instr_list)
```

Tip: We believe our R&S VISA is the best choice for our customers. Here are the reasons why:

- Small footprint
- Superior VXI-11 and HiSLIP performance
- Integrated legacy sensors NRP-Zxx support

- Additional VXI-11 and LXI devices search
- Availability for Windows, Linux, Mac OS

2.4 Initiating Instrument Session

RsCMPX_UwbMeas offers four different types of starting your remote-control session. We begin with the most typical case, and progress with more special ones.

Standard Session Initialization

Initiating new instrument session happens, when you instantiate the RsCMPX_UwbMeas object. Below, is a simple Hello World example. Different resource names are examples for different physical interfaces.

```
"""
Simple example on how to use the RsCMPX_UwbMeas module for remote-controlling your
↳instrument
Preconditions:

- Installed RsCMPX_UwbMeas Python module Version 5.0.20 or newer from pypi.org
- Installed VISA, for example R&S Visa 5.12 or newer
"""

from RsCMPX_UwbMeas import *

# A good practice is to assure that you have a certain minimum version installed
RsCMPX_UwbMeas.assert_minimum_version('5.0.20')
resource_string_1 = 'TCPIP::192.168.2.101::INSTR' # Standard LAN connection (also
↳called VXI-11)
resource_string_2 = 'TCPIP::192.168.2.101::hislip0' # Hi-Speed LAN connection - see
↳1MA208
resource_string_3 = 'GPIB::20::INSTR' # GPIB Connection
resource_string_4 = 'USB::0x0AAD::0x0119::022019943::INSTR' # USB-TMC (Test and
↳Measurement Class)

# Initializing the session
driver = RsCMPX_UwbMeas(resource_string_1)

idn = driver.utilities.query_str('*IDN?')
print(f"\nHello, I am: '{idn}'")
print(f'RsCMPX_UwbMeas package version: {driver.utilities.driver_version}')
print(f'Visa manufacturer: {driver.utilities.visa_manufacturer}')
print(f'Instrument full name: {driver.utilities.full_instrument_model_name}')
print(f'Instrument installed options: {",".join(driver.utilities.instrument_options)}')

# Close the session
driver.close()
```

Note: If you are wondering about the missing ASRL1::INSTR, yes, it works too, but come on... it's 2023.

Do not care about specialty of each session kind; RsCMPX_UwbMeas handles all the necessary session settings for you. You immediately have access to many identification properties in the interface `driver.utilities`. Here are some of them:

- `idn_string`
- `driver_version`
- `visa_manufacturer`
- `full_instrument_model_name`
- `instrument_serial_number`
- `instrument_firmware_version`
- `instrument_options`

The constructor also contains optional boolean arguments `id_query` and `reset`:

```
driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::hislip0', id_query=True, reset=True)
```

- Setting `id_query` to `True` (default is `True`) checks, whether your instrument can be used with the RsCMPX_UwbMeas module.
- Setting `reset` to `True` (default is `False`) resets your instrument. It is equivalent to calling the `reset()` method.

Selecting a Specific VISA

Just like in the function `list_resources()`, the RsCMPX_UwbMeas allows you to choose which VISA to use:

```
"""
Choosing VISA implementation
"""

from RsCMPX_UwbMeas import *

# Force use of the Rs Visa. For NI Visa, use the "SelectVisa='ni'"
driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR', True, True, "SelectVisa='rs'")

idn = driver.utilities.query_str('*IDN?')
print(f"\nHello, I am: '{idn}'")
print(f"\nI am using the VISA from: {driver.utilities.visa_manufacturer}")

# Close the session
driver.close()
```

No VISA Session

We recommend using VISA when possible preferably with HiSlip session because of its low latency. However, if you are a strict VISA denier, RsCMPX_UwbMeas has something for you too - **no Visa installation raw LAN socket**:

```
"""
Using RsCMPX_UwbMeas without VISA for LAN Raw socket communication
"""
```

(continues on next page)

(continued from previous page)

```

from RsCMPX_UwbMeas import *

driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::5025::SOCKET', True, True, "SelectVisa=
↳ 'socket'")
print(f'Visa manufacturer: {driver.utilities.visa_manufacturer}')
print(f"\nHello, I am: '{driver.utilities.idn_string}'")

# Close the session
driver.close()

```

Warning: Not using VISA can cause problems by debugging when you want to use the communication Trace Tool. The good news is, you can easily switch to use VISA and back just by changing the constructor arguments. The rest of your code stays unchanged.

Simulating Session

If a colleague is currently occupying your instrument, leave him in peace, and open a simulating session:

```
driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::hislip0', True, True, "Simulate=True")
```

More option_string tokens are separated by comma:

```
driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::hislip0', True, True, "SelectVisa='rs',
↳ Simulate=True")
```

Shared Session

In some scenarios, you want to have two independent objects talking to the same instrument. Rather than opening a second VISA connection, share the same one between two or more RsCMPX_UwbMeas objects:

```

"""
Sharing the same physical VISA session by two different RsCMPX_UwbMeas objects
"""

from RsCMPX_UwbMeas import *

driver1 = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR', True, True)
driver2 = RsCMPX_UwbMeas.from_existing_session(driver1)

print(f'driver1: {driver1.utilities.idn_string}')
print(f'driver2: {driver2.utilities.idn_string}')

# Closing the driver2 session does not close the driver1 session - driver1 is the
↳ 'session master'
driver2.close()
print(f'driver2: I am closed now')

print(f'driver1: I am still opened and working: {driver1.utilities.idn_string}')

```

(continues on next page)

(continued from previous page)

```
driver1.close()
print(f'driver1: Only now I am closed.')
```

Note: The `driver1` is the object holding the ‘master’ session. If you call the `driver1.close()`, the `driver2` loses its instrument session as well, and becomes pretty much useless.

2.5 Plain SCPI Communication

After you have opened the session, you can use the instrument-specific part described in the RsCMPX_UwbMeas API Structure. If for any reason you want to use the plain SCPI, use the `utilities` interface’s two basic methods:

- `write_str()` - writing a command without an answer, for example `*RST`
- `query_str()` - querying your instrument, for example the `*IDN?` query

You may ask a question. Actually, two questions:

- **Q1:** Why there are not called `write()` and `query()` ?
- **Q2:** Where is the `read()` ?

Answer 1: Actually, there are - the `write_str()` / `write()` and `query_str()` / `query()` are aliases, and you can use any of them. We promote the `_str` names, to clearly show you want to work with strings. Strings in Python3 are Unicode, the `bytes` and `string` objects are not interchangeable, since one character might be represented by more than 1 byte. To avoid mixing string and binary communication, all the method names for binary transfer contain `_bin` in the name.

Answer 2: Short answer - you do not need it. Long answer - your instrument never sends unsolicited responses. If you send a set command, you use `write_str()`. For a query command, you use `query_str()`. So, you really do not need it...

Bottom line - if you are used to `write()` and `query()` methods, from pyvisa, the `write_str()` and `query_str()` are their equivalents.

Enough with the theory, let us look at an example. Simple write, and query:

```
"""
Basic string write_str / query_str
"""

from RsCMPX_UwbMeas import *

driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR')
driver.utilities.write_str('*RST')
response = driver.utilities.query_str('*IDN?')
print(response)

# Close the session
driver.close()
```

This example is so-called “*University-Professor-Example*” - good to show a principle, but never used in praxis. The abovementioned commands are already a part of the driver’s API. Here is another example, achieving the same goal:

```

"""
Basic string write_str / query_str
"""

from RsCMPX_UwbMeas import *

driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR')
driver.utilities.reset()
print(driver.utilities.idn_string)

# Close the session
driver.close()

```

One additional feature we need to mention here: **VISA timeout**. To simplify, VISA timeout plays a role in each `query_xxx()`, where the controller (your PC) has to prevent waiting forever for an answer from your instrument. VISA timeout defines that maximum waiting time. You can set/read it with the `visa_timeout` property:

```

# Timeout in milliseconds
driver.utilities.visa_timeout = 3000

```

After this time, the `RsCMPX_UwbMeas` raises an exception. Speaking of exceptions, an important feature of the `RsCMPX_UwbMeas` is **Instrument Status Checking**. Check out the next chapter that describes the error checking in details.

For completion, we mention other string-based `write_xxx()` and `query_xxx()` methods - all in one example. They are convenient extensions providing type-safe float/boolean/integer setting/querying features:

```

"""
Basic string write_xxx / query_xxx
"""

from RsCMPX_UwbMeas import *

driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR')
driver.utilities.visa_timeout = 5000
driver.utilities.instrument_status_checking = True
driver.utilities.write_int('SWEEP:COUNT ', 10) # sending 'SWEEP:COUNT 10'
driver.utilities.write_bool('SOURCE:RF:OUTPUT:STATE ', True) # sending
↳ 'SOURCE:RF:OUTPUT:STATE ON'
driver.utilities.write_float('SOURCE:RF:FREQUENCY ', 1E9) # sending 'SOURCE:RF:FREQUENCY_
↳ 1000000000'

sc = driver.utilities.query_int('SWEEP:COUNT?') # returning integer number sc=10
out = driver.utilities.query_bool('SOURCE:RF:OUTPUT:STATE?') # returning boolean_
↳ out=True
freq = driver.utilities.query_float('SOURCE:RF:FREQUENCY?') # returning float number_
↳ freq=1E9

# Close the session
driver.close()

```

Lastly, a method providing basic synchronization: `query_opc()`. It sends query ***OPC?** to your instrument. The instrument waits with the answer until all the tasks it currently has in a queue are finished. This way your program waits too, and this way it is synchronized with the actions in the instrument. Remember to have the VISA timeout set

to an appropriate value to prevent the timeout exception. Here's the snippet:

```
driver.utilities.visa_timeout = 3000
driver.utilities.write_str("INIT")
driver.utilities.query_opc()

# The results are ready now to fetch
results = driver.utilities.query_str("FETCH:MEASUREMENT?")
```

Tip: Wait, there's more: you can send the ***OPC?** after each `write_xxx()` automatically:

```
# Default value after init is False
driver.utilities.opc_query_after_write = True
```

2.6 Error Checking

RsCMPX_UwbMeas pushes limits even further (internal R&S joke): It has a built-in mechanism that after each command/query checks the instrument's status subsystem, and raises an exception if it detects an error. For those who are already screaming: **Speed Performance Penalty!!!**, don't worry, you can disable it.

Instrument status checking is very useful since in case your command/query caused an error, you are immediately informed about it. Status checking has in most cases no practical effect on the speed performance of your program. However, if for example, you do many repetitions of short write/query sequences, it might make a difference to switch it off:

```
# Default value after init is True
driver.utilities.instrument_status_checking = False
```

To clear the instrument status subsystem of all errors, call this method:

```
driver.utilities.clear_status()
```

Instrument's status system error queue is clear-on-read. It means, if you query its content, you clear it at the same time. To query and clear list of all the current errors, use this snippet:

```
errors_list = driver.utilities.query_all_errors()
```

See the next chapter on how to react on errors.

2.7 Exception Handling

The base class for all the exceptions raised by the RsCMPX_UwbMeas is `RsInstrException`. Inherited exception classes:

- `ResourceError` raised in the constructor by problems with initiating the instrument, for example wrong or non-existing resource name
- `StatusException` raised if a command or a query generated error in the instrument's error queue
- `TimeoutException` raised if a visa timeout or an opc timeout is reached

In this example we show usage of all of them. Because it is difficult to generate an error using the instrument-specific SCPI API, we use plain SCPI commands:

```

"""
Showing how to deal with exceptions
"""

from RsCMPX_UwbMeas import *

driver = None
# Try-catch for initialization. If an error occurs, the ResourceError is raised
try:
    driver = RsCMPX_UwbMeas('TCPIP::10.112.1.179::hislip0')
except ResourceError as e:
    print(e.args[0])
    print('Your instrument is probably OFF...')
    # Exit now, no point of continuing
    exit(1)

# Dealing with commands that potentially generate errors OPTION 1:
# Switching the status checking OFF temporarily
driver.utilities.instrument_status_checking = False
driver.utilities.write_str('MY:MISSpelled:COMManD')
# Clear the error queue
driver.utilities.clear_status()
# Status checking ON again
driver.utilities.instrument_status_checking = True

# Dealing with queries that potentially generate errors OPTION 2:
try:
    # You might want to reduce the VISA timeout to avoid long waiting
    driver.utilities.visa_timeout = 1000
    driver.utilities.query_str('MY:WRONG:QUERY?')

except StatusException as e:
    # Instrument status error
    print(e.args[0])
    print('Nothing to see here, moving on...')

except TimeoutException as e:
    # Timeout error
    print(e.args[0])
    print('That took a long time...')

except RsInstrException as e:
    # RsInstrException is a base class for all the RsCMPX_UwbMeas exceptions
    print(e.args[0])
    print('Some other RsCMPX_UwbMeas error...')

finally:
    driver.utilities.visa_timeout = 5000
    # Close the session in any case
    driver.close()

```

Tip: General rules for exception handling:

- If you are sending commands that might generate errors in the instrument, for example deleting a file which does not exist, use the **OPTION 1** - temporarily disable status checking, send the command, clear the error queue and enable the status checking again.
 - If you are sending queries that might generate errors or timeouts, for example querying measurement that can not be performed at the moment, use the **OPTION 2** - try/except with optionally adjusting the timeouts.
-

2.8 Transferring Files

Instrument -> PC

You definitely experienced it: you just did a perfect measurement, saved the results as a screenshot to an instrument's storage drive. Now you want to transfer it to your PC. With RsCMPX_UwbMeas, no problem, just figure out where the screenshot was stored on the instrument. In our case, it is `/var/user/instr_screenshot.png`:

```
driver.utilities.read_file_from_instrument_to_pc(  
    r'/var/user/instr_screenshot.png',  
    r'c:\temp\pc_screenshot.png')
```

PC -> Instrument

Another common scenario: Your cool test program contains a setup file you want to transfer to your instrument: Here is the RsCMPX_UwbMeas one-liner split into 3 lines:

```
driver.utilities.send_file_from_pc_to_instrument(  
    r'c:\MyCoolTestProgram\instr_setup.sav',  
    r'/var/appdata/instr_setup.sav')
```

2.9 Writing Binary Data

Writing from bytes

An example where you need to send binary data is a waveform file of a vector signal generator. First, you compose your `wform_data` as bytes, and then you send it with `write_bin_block()`:

```
# MyWaveform.wv is an instrument file name under which this data is stored  
driver.utilities.write_bin_block(  
    "SOUR:BB:ARB:WAV:DATA 'MyWaveform.wv'",",  
    wform_data)
```

Note: Notice the `write_bin_block()` has two parameters:

- string parameter `cmd` for the SCPI command
 - bytes parameter `payload` for the actual binary data to send
-

Writing from PC files

Similar to querying binary data to a file, you can write binary data from a file. The second parameter is then the PC file path the content of which you want to send:

```
driver.utilities.write_bin_block_from_file(
    "SOUR:BB:ARB:WAV:DATA 'MyWaveform.wv'",
    r"c:\temp\wform_data.wv")
```

2.10 Transferring Big Data with Progress

We can agree that it can be annoying using an application that shows no progress for long-lasting operations. The same is true for remote-control programs. Luckily, the RsCMPX_UwbMeas has this covered. And, this feature is quite universal - not just for big files transfer, but for any data in both directions.

RsCMPX_UwbMeas allows you to register a function (programmers fancy name is `callback`), which is then periodically invoked after transfer of one data chunk. You can define that chunk size, which gives you control over the callback invoke frequency. You can even slow down the transfer speed, if you want to process the data as they arrive (direction instrument -> PC).

To show this in praxis, we are going to use another *University-Professor-Example*: querying the `*IDN?` with chunk size of 2 bytes and delay of 200ms between each chunk read:

```
"""
Event handlers by reading
"""

from RsCMPX_UwbMeas import *
import time

def my_transfer_handler(args):
    """Function called each time a chunk of data is transferred"""
    # Total size is not always known at the beginning of the transfer
    total_size = args.total_size if args.total_size is not None else "unknown"

    print(f"Context: '{args.context}'{'with opc' if args.opc_sync else ''}", "
          f"chunk {args.chunk_ix}, "
          f"transferred {args.transferred_size} bytes, "
          f"total size {total_size}, "
          f"direction {'reading' if args.reading else 'writing'}", "
          f"data '{args.data}'")

    if args.end_of_transfer:
        print('End of Transfer')
        time.sleep(0.2)

driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR')

driver.events.on_read_handler = my_transfer_handler
# Switch on the data to be included in the event arguments
```

(continues on next page)

(continued from previous page)

```
# The event arguments args.data will be updated
driver.events.io_events_include_data = True
# Set data chunk size to 2 bytes
driver.utilities.data_chunk_size = 2
driver.utilities.query_str('*IDN?')
# Unregister the event handler
driver.utilities.on_read_handler = None

# Close the session
driver.close()
```

If you start it, you might wonder (or maybe not): why is the `args.total_size = None`? The reason is, in this particular case the RsCMPX_UwbMeas does not know the size of the complete response up-front. However, if you use the same mechanism for transfer of a known data size (for example, file transfer), you get the information about the total size too, and hence you can calculate the progress as:

$$\text{progress [pct]} = 100 * \text{args.transferred_size} / \text{args.total_size}$$

Snippet of transferring file from PC to instrument, the rest of the code is the same as in the previous example:

```
driver.events.on_write_handler = my_transfer_handler
driver.events.io_events_include_data = True
driver.data_chunk_size = 1000
driver.utilities.send_file_from_pc_to_instrument(
    r'c:\MyCoolTestProgram\my_big_file.bin',
    r'/var/user/my_big_file.bin')
# Unregister the event handler
driver.events.on_write_handler = None
```

2.11 Multithreading

You are at the party, many people talking over each other. Not every person can deal with such crosstalk, neither can measurement instruments. For this reason, RsCMPX_UwbMeas has a feature of scheduling the access to your instrument by using so-called **Locks**. Locks make sure that there can be just one client at a time *talking* to your instrument. Talking in this context means completing one communication step - one command write or write/read or write/read/error check.

To describe how it works, and where it matters, we take three typical multithread scenarios:

One instrument session, accessed from multiple threads

You are all set - the lock is a part of your instrument session. Check out the following example - it will execute properly, although the instrument gets 10 queries at the same time:

```
"""
Multiple threads are accessing one RsCMPX_UwbMeas object
"""

import threading
from RsCMPX_UwbMeas import *
```

(continues on next page)

(continued from previous page)

```

def execute(session):
    """Executed in a separate thread."""
    session.utilities.query_str('*IDN?')

driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR')
threads = []
for i in range(10):
    t = threading.Thread(target=execute, args=(driver, ))
    t.start()
    threads.append(t)
print('All threads started')

# Wait for all threads to join this main thread
for t in threads:
    t.join()
print('All threads ended')

driver.close()

```

Shared instrument session, accessed from multiple threads

Same as the previous case, you are all set. The session carries the lock with it. You have two objects, talking to the same instrument from multiple threads. Since the instrument session is shared, the same lock applies to both objects causing the exclusive access to the instrument.

Try the following example:

```

"""
Multiple threads are accessing two RsCMPX_UwbMeas objects with shared session
"""

import threading
from RsCMPX_UwbMeas import *

def execute(session: RsCMPX_UwbMeas, session_ix, index) -> None:
    """Executed in a separate thread."""
    print(f'{index} session {session_ix} query start...')
    session.utilities.query_str('*IDN?')
    print(f'{index} session {session_ix} query end')

driver1 = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR')
driver2 = RsCMPX_UwbMeas.from_existing_session(driver1)
driver1.utilities.visa_timeout = 200
driver2.utilities.visa_timeout = 200
# To see the effect of crosstalk, uncomment this line
# driver2.utilities.clear_lock()

```

(continues on next page)

(continued from previous page)

```

threads = []
for i in range(10):
    t = threading.Thread(target=execute, args=(driver1, 1, i,))
    t.start()
    threads.append(t)
    t = threading.Thread(target=execute, args=(driver2, 2, i,))
    t.start()
    threads.append(t)
print('All threads started')

# Wait for all threads to join this main thread
for t in threads:
    t.join()
print('All threads ended')

driver2.close()
driver1.close()

```

As you see, everything works fine. If you want to simulate some party crosstalk, uncomment the line `driver2.utilities.clear_lock()`. This causes the driver2 session lock to break away from the driver1 session lock. Although the driver1 still tries to schedule its instrument access, the driver2 tries to do the same at the same time, which leads to all the fun stuff happening.

Multiple instrument sessions accessed from multiple threads

Here, there are two possible scenarios depending on the instrument's VISA interface:

- You are lucky, because your instrument handles each remote session completely separately. An example of such instrument is SMW200A. In this case, you have no need for session locking.
- Your instrument handles all sessions with one set of in/out buffers. You need to lock the session for the duration of a talk. And you are lucky again, because the RsCMPX_UwbMeas takes care of it for you. The text below describes this scenario.

Run the following example:

```

"""
Multiple threads are accessing two RsCMPX_UwbMeas objects with two separate sessions
"""

import threading
from RsCMPX_UwbMeas import *

def execute(session: RsCMPX_UwbMeas, session_ix, index) -> None:
    """Executed in a separate thread."""
    print(f'{index} session {session_ix} query start...')
    session.utilities.query_str('*IDN?')
    print(f'{index} session {session_ix} query end')

driver1 = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR')
driver2 = RsCMPX_UwbMeas('TCPIP::192.168.56.101::INSTR')

```

(continues on next page)

(continued from previous page)

```

driver1.utilities.visa_timeout = 200
driver2.utilities.visa_timeout = 200

# Synchronise the sessions by sharing the same lock
driver2.utilities.assign_lock(driver1.utilities.get_lock()) # To see the effect of
↳ crosstalk, comment this line

threads = []
for i in range(10):
    t = threading.Thread(target=execute, args=(driver1, 1, i,))
    t.start()
    threads.append(t)
    t = threading.Thread(target=execute, args=(driver2, 2, i,))
    t.start()
    threads.append(t)
print('All threads started')

# Wait for all threads to join this main thread
for t in threads:
    t.join()
print('All threads ended')

driver2.close()
driver1.close()

```

You have two completely independent sessions that want to talk to the same instrument at the same time. This will not go well, unless they share the same session lock. The key command to achieve this is `driver2.utilities.assign_lock(driver1.utilities.get_lock())`. Try to comment it and see how it goes. If despite commenting the line the example runs without issues, you are lucky to have an instrument similar to the SMW200A.

2.12 Logging

Yes, the logging again. This one is tailored for instrument communication. You will appreciate such handy feature when you troubleshoot your program, or just want to protocol the SCPI communication for your test reports.

What can you actually do with the logger?

- Write SCPI communication to a stream-like object, for example console or file, or both simultaneously
- Log only errors and skip problem-free parts; this way you avoid going through thousands lines of texts
- Investigate duration of certain operations to optimize your program's performance
- Log custom messages from your program

Let us take this basic example:

```

"""
Basic logging example to the console
"""

from RsCMPX_UwbMeas import *

```

(continues on next page)

(continued from previous page)

```

driver = RsCMPX_UwbMeas('TCPIP::192.168.1.101::INSTR')

# Switch ON logging to the console.
driver.utilities.logger.log_to_console = True
driver.utilities.logger.mode = LoggingMode.On
driver.utilities.reset()

# Close the session
driver.close()

```

Console output:

10:29:10.819	TCPIP::192.168.1.101::INSTR	0.976 ms	Write: *RST
10:29:10.819	TCPIP::192.168.1.101::INSTR	1884.985 ms	Status check: OK
10:29:12.704	TCPIP::192.168.1.101::INSTR	0.983 ms	Query OPC: 1
10:29:12.705	TCPIP::192.168.1.101::INSTR	2.892 ms	Clear status: OK
10:29:12.708	TCPIP::192.168.1.101::INSTR	3.905 ms	Status check: OK
10:29:12.712	TCPIP::192.168.1.101::INSTR	1.952 ms	Close: Closing session

The columns of the log are aligned for better reading. Columns meaning:

- (1) Start time of the operation
- (2) Device resource name (you can set an alias)
- (3) Duration of the operation
- (4) Log entry

Tip: You can customize the logging format with `set_format_string()`, and set the maximum log entry length with the properties:

- `abbreviated_max_len_ascii`
- `abbreviated_max_len_bin`
- `abbreviated_max_len_list`

See the full logger help [here](#).

Notice the SCPI communication starts from the line `driver.utilities.reset()`. If you want to log the initialization of the session as well, you have to switch the logging ON already in the constructor:

```
driver = RsCMPX_UwbMeas('TCPIP::192.168.56.101::hislip0', options='LoggingMode=On')
```

Parallel to the console logging, you can log to a general stream. Do not fear the programmer's jargon... under the term **stream** you can just imagine a file. To be a little more technical, a stream in Python is any object that has two methods: `write()` and `flush()`. This example opens a file and sets it as logging target:

```

"""
Example of logging to a file
"""

from RsCMPX_UwbMeas import *

driver = RsCMPX_UwbMeas('TCPIP::192.168.1.101::INSTR')

```

(continues on next page)

(continued from previous page)

```

# We also want to log to the console.
driver.utilities.logger.log_to_console = True

# Logging target is our file
file = open(r'c:\temp\my_file.txt', 'w')
driver.utilities.logger.set_logging_target(file)
driver.utilities.logger.mode = LoggingMode.On

# Instead of the 'TCPIP::192.168.1.101::INSTR', show 'MyDevice'
driver.utilities.logger.device_name = 'MyDevice'

# Custom user entry
driver.utilities.logger.info_raw('----- This is my custom log entry. ---- ')

driver.utilities.reset()

# Close the session
driver.close()

# Close the log file
file.close()

```

Tip: To make the log more compact, you can skip all the lines with Status check: OK:

```
driver.utilities.logger.log_status_check_ok = False
```

Hint: You can share the logging file between multiple sessions. In such case, remember to close the file only after you have stopped logging in all your sessions, otherwise you get a log write error.

For logging to a UDP port in addition to other log targets, use one of the lines:

```
driver.utilities.logger.log_to_udp = True
driver.utilities.logger.log_to_console_and_udp = True
```

You can select the UDP port to log to, the default is 49200:

```
driver.utilities.logger.udp_port = 49200
```

Another cool feature is logging only errors. To make this mode usefull for troubleshooting, you also want to see the circumstances which lead to the errors. Each driver elementary operation, for example, `write_str()`, can generate a group of log entries - let us call them **Segment**. In the logging mode **Errors**, a whole segment is logged only if at least one entry of the segment is an error.

The script below demonstrates this feature. We use a direct SCPI communication to send a misspelled SCPI command *CLS, which leads to instrument status error:

```

"""
Logging example to the console with only errors logged
"""

```

(continues on next page)

(continued from previous page)

```
from RsCMPX_UwbMeas import *

driver = RsCMPX_UwbMeas('TCPIP::192.168.1.101::INSTR', options='LoggingMode=Errors')

# Switch ON logging to the console.
driver.utilities.logger.log_to_console = True

# Reset will not be logged, since no error occurred there
driver.utilities.reset()

# Now a misspelled command.
driver.utilities.write('*CLaS')

# A good command again, no logging here
idn = driver.utilities.query('*IDN?')

# Close the session
driver.close()
```

Console output:

```
12:11:02.879 TCPIP::192.168.1.101::INSTR    0.976 ms Write string: *CLaS
12:11:02.879 TCPIP::192.168.1.101::INSTR    6.833 ms Status check: StatusException:
                                     Instrument error detected: Undefined header;
→ *CLaS
```

Notice the following:

- Although the operation **Write string: *CLaS** finished without an error, it is still logged, because it provides the context for the actual error which occurred during the status checking right after.
- No other log entries are present, including the session initialization and close, because they were all error-free.

3.1 Connector

```
# First value:
value = enums.Connector.I11I
# Last value:
value = enums.Connector.RH8
# All values (163x):
I11I | I13I | I15I | I17I | I21I | I23I | I25I | I27I
I31I | I33I | I35I | I37I | I41I | I43I | I45I | I47I
IFI1 | IFI2 | IFI3 | IFI4 | IFI5 | IFI6 | IQ1I | IQ3I
IQ5I | IQ7I | R10D | R11 | R11C | R11D | R12 | R12C
R12D | R12I | R13 | R13C | R14 | R14C | R14I | R15
R16 | R17 | R18 | R21 | R21C | R22 | R22C | R22I
R23 | R23C | R24 | R24C | R24I | R25 | R26 | R27
R28 | R31 | R31C | R32 | R32C | R32I | R33 | R33C
R34 | R34C | R34I | R35 | R36 | R37 | R38 | R41
R41C | R42 | R42C | R42I | R43 | R43C | R44 | R44C
R44I | R45 | R46 | R47 | R48 | RA1 | RA2 | RA3
RA4 | RA5 | RA6 | RA7 | RA8 | RB1 | RB2 | RB3
RB4 | RB5 | RB6 | RB7 | RB8 | RC1 | RC2 | RC3
RC4 | RC5 | RC6 | RC7 | RC8 | RD1 | RD2 | RD3
RD4 | RD5 | RD6 | RD7 | RD8 | RE1 | RE2 | RE3
RE4 | RE5 | RE6 | RE7 | RE8 | RF1 | RF1C | RF2
RF2C | RF2I | RF3 | RF3C | RF4 | RF4C | RF4I | RF5
RF5C | RF6 | RF6C | RF7 | RF7C | RF8 | RF8C | RF9C
RFAC | RFBC | RFBI | RG1 | RG2 | RG3 | RG4 | RG5
RG6 | RG7 | RG8 | RH1 | RH2 | RH3 | RH4 | RH5
RH6 | RH7 | RH8
```

3.2 MaxSpecPowLen

```
# Example value:  
value = enums.MaxSpecPowLen.MS1  
# All values (2x):  
MS1 | PPDU
```

3.3 PhrDataRate

```
# Example value:  
value = enums.PhrDataRate.DRHP  
# All values (8x):  
DRHP | DRLP | DRMD | IGN | RHHM | RHML | RSF | SYNC
```

3.4 PpduMode

```
# Example value:  
value = enums.PpduMode.MPPDu  
# All values (2x):  
MPPDu | SPPDu
```

3.5 Repeat

```
# Example value:  
value = enums.Repeat.CONTInuous  
# All values (2x):  
CONTInuous | SINGleshot
```

3.6 ResourceState

```
# Example value:  
value = enums.ResourceState.ACTive  
# All values (8x):  
ACTive | ADJusted | INValid | OFF | PENDing | QUEued | RDY | RUN
```

3.7 Result

```
# Example value:
value = enums.Result.FAIL
# All values (2x):
FAIL | PASS
```

3.8 ResultStatus2

```
# First value:
value = enums.ResultStatus2.DC
# Last value:
value = enums.ResultStatus2.ULEU
# All values (10x):
DC | INV | NAV | NCAP | OFF | OFL | OK | UFL
ULEL | ULEU
```

3.9 SignalSlope

```
# Example value:
value = enums.SignalSlope.FEDGE
# All values (2x):
FEDGE | REDGE
```

3.10 StopCondition

```
# Example value:
value = enums.StopCondition.NONE
# All values (2x):
NONE | SLFail
```

3.11 StsSegmentLen

```
# Example value:
value = enums.StsSegmentLen.L128
# All values (4x):
L128 | L256 | L32 | L64
```

3.12 TargetMainState

```
# Example value:  
value = enums.TargetMainState.OFF  
# All values (3x):  
OFF | RDY | RUN
```

3.13 TargetSyncState

```
# Example value:  
value = enums.TargetSyncState.ADJusted  
# All values (2x):  
ADJusted | PENDing
```

REPCAPS

4.1 Instance (Global)

```
# Setting:
driver.repcap_instance_set(repcap.Instance.Inst1)
# Range:
Inst1 .. Inst16
# All values (16x):
Inst1 | Inst2 | Inst3 | Inst4 | Inst5 | Inst6 | Inst7 | Inst8
Inst9 | Inst10 | Inst11 | Inst12 | Inst13 | Inst14 | Inst15 | Inst16
```

4.2 Area

```
# First value:
value = repcap.Area.Nr1
# Values (3x):
Nr1 | Nr2 | Nr3
```

4.3 Ppdu

```
# First value:
value = repcap.Ppdu.Nr1
# Range:
Nr1 .. Nr100
# All values (100x):
Nr1 | Nr2 | Nr3 | Nr4 | Nr5 | Nr6 | Nr7 | Nr8
Nr9 | Nr10 | Nr11 | Nr12 | Nr13 | Nr14 | Nr15 | Nr16
Nr17 | Nr18 | Nr19 | Nr20 | Nr21 | Nr22 | Nr23 | Nr24
Nr25 | Nr26 | Nr27 | Nr28 | Nr29 | Nr30 | Nr31 | Nr32
Nr33 | Nr34 | Nr35 | Nr36 | Nr37 | Nr38 | Nr39 | Nr40
Nr41 | Nr42 | Nr43 | Nr44 | Nr45 | Nr46 | Nr47 | Nr48
Nr49 | Nr50 | Nr51 | Nr52 | Nr53 | Nr54 | Nr55 | Nr56
Nr57 | Nr58 | Nr59 | Nr60 | Nr61 | Nr62 | Nr63 | Nr64
Nr65 | Nr66 | Nr67 | Nr68 | Nr69 | Nr70 | Nr71 | Nr72
Nr73 | Nr74 | Nr75 | Nr76 | Nr77 | Nr78 | Nr79 | Nr80
```

(continues on next page)

(continued from previous page)

Nr81	Nr82	Nr83	Nr84	Nr85	Nr86	Nr87	Nr88
Nr89	Nr90	Nr91	Nr92	Nr93	Nr94	Nr95	Nr96
Nr97	Nr98	Nr99	Nr100				

4.4 Record

```
# First value:
value = repcap.Record.Nr1
# Range:
Nr1 .. Nr20
# All values (20x):
Nr1 | Nr2 | Nr3 | Nr4 | Nr5 | Nr6 | Nr7 | Nr8
Nr9 | Nr10 | Nr11 | Nr12 | Nr13 | Nr14 | Nr15 | Nr16
Nr17 | Nr18 | Nr19 | Nr20
```


EXAMPLES

For more examples, visit our Rohde & Schwarz Github repository.

```

"""
# GitHub examples repository path: CMXP/Python/RsCmxp_xxx_ScpiPackages

Example on how to use the python RsCmx auto-generated instrument drivers for
RsCmpx_Base and RsCmpx_Gprf (Base and GPRF) in one script with shared VISA session.
"""

from RsCMPX_Base.RsCMPX_Base import RsCMPX_Base # install from pypi.org
from RsCMPX_Base import enums as base_enums
from RsCMPX_Base import repcap as base_repcap

from RsCMPX_Gprf.RsCMPX_Gprf import RsCMPX_Gprf # install from pypi.org
from RsCMPX_Gprf.CustomFiles.reliability import ReliabilityEventArgs
from RsCMPX_Gprf import enums as gprf_enums
from RsCMPX_Gprf import repcap as gprf_repcaps

# CMX Base init
cmx_base = RsCMPX_Base('TCPIP::10.112.1.116', False, True)
print(f'CMX Base IND: {cmx_base.utilities.idn_string}')
print(f'CMX Instrument options:\n{" ".join(cmx_base.utilities.instrument_options)}')
cmx_base.utilities.visa_timeout = 5000 # default is 10000

# Sends OPC after each command
cmx_base.utilities.opc_query_after_write = False
# Checks for syst:err? after each command / query - default value after init is True
cmx_base.utilities.instrument_status_checking = True

# Self-test
self_test = cmx_base.utilities.self_test()
print(f'CMW self-test result: {self_test} - {"Passed" if self_test[0] == 0 else "Failed"}')
# Reference Frequency Source
cmx_base.system.reference.frequency.source_set(base_enums.SourceIntExt.INTERNAL)

# CMX RsCMPX_Gprf Init - reuse the session of the cmx_base, rather than creating another
cmx_gprf = RsCMPX_Gprf.from_existing_session(cmx_base)

```

(continues on next page)

(continued from previous page)

```
cmx_gprf.utilities.visa_timeout = 5000

# Driver's Interface reliability offers a convenient way of reacting on the return value.
↳ Reliability Indicator
cmx_gprf.reliability.ExceptionOnError = True # default is 10000

# Callback to use for the reliability indicator update events
def my_reliability_handler(event_args: ReliabilityEventArgs):
    print(f'GPRF Reliability updated.\nContext: {event_args.context}\nMessage:
↳ {event_args.message}')

# We register a callback for each change in the reliability indicator
cmx_gprf.reliability.on_update_handler = my_reliability_handler

# You can obtain the last value of the returned reliability
print(f"\nReliability last value: {cmx_gprf.reliability.last_value}, context '{cmx_gprf.
↳ reliability.last_context}', message: {cmx_gprf.reliability.last_message}")

# Close the sessions
cmx_gprf.close()
cmx_base.close()
```

RSCMPX_UWBMEAS API STRUCTURE

Global RepCaps

```
driver = RsCMPX_UwbMeas('TCPIP::192.168.2.101::hislip0')
# Instance range: Inst1 .. Inst16
rc = driver.repcap_instance_get()
driver.repcap_instance_set(repcap.Instance.Inst1)
```

class RsCMPX_UwbMeas(*resource_name: str, id_query: bool = True, reset: bool = False, options: str = None, direct_session: object = None*)

554 total commands, 5 Subgroups, 0 group commands

Initializes new RsCMPX_UwbMeas session.

Parameter options tokens examples:

- **Simulate=True** - starts the session in simulation mode. Default: **False**
- **SelectVisa=socket** - uses no VISA implementation for socket connections - you do not need any VISA-C installation
- **SelectVisa=rs** - forces usage of RohdeSchwarz Visa
- **SelectVisa=ivi** - forces usage of National Instruments Visa
- **QueryInstrumentStatus = False** - same as **driver.utilities.instrument_status_checking = False**. Default: **True**
- **WriteDelay = 20**, **ReadDelay = 5** - Introduces delay of 20ms before each write and 5ms before each read. Default: **0ms** for both
- **OpcWaitMode = OpcQuery** - mode for all the opc-synchronised write/reads. Other modes: **StbPolling**, **StbPollingSlow**, **StbPollingSuperSlow**. Default: **StbPolling**
- **AddTermCharToWriteBinBlock = True** - Adds one additional LF to the end of the binary data (some instruments require that). Default: **False**
- **AssureWriteWithTermChar = True** - Makes sure each command/query is terminated with termination character. Default: Interface dependent
- **TerminationCharacter = "\r"** - Sets the termination character for reading. Default: **\n** (LineFeed or LF)
- **DataChunkSize = 10E3** - Maximum size of one write/read segment. If transferred data is bigger, it is split to more segments. Default: **1E6** bytes
- **OpcTimeout = 10000** - same as **driver.utilities.opc_timeout = 10000**. Default: **30000ms**
- **VisaTimeout = 5000** - same as **driver.utilities.visa_timeout = 5000**. Default: **10000ms**

- `ViClearExeMode` = Disabled - `viClear()` execution mode. Default: `execute_on_all`
- `OpcQueryAfterWrite` = True - same as `driver.utilities.opc_query_after_write` = True. Default: False
- `StbInErrorCheck` = False - if true, the driver checks errors with `*STB?` If false, it uses `SYST:ERR?`. Default: True
- `ScpiQuotes` = double'. - for SCPI commands, you can define how strings are quoted. With single or double quotes. Possible values: `single` | `double` | `{char}`. Default: ```single`
- `LoggingMode` = On - Sets the logging status right from the start. Default: Off
- `LoggingName` = 'MyDevice' - Sets the name to represent the session in the log entries. Default: 'resource_name'
- `LogToGlobalTarget` = True - Sets the logging target to the class-property previously set with `RsCMPX_UwbMeas.set_global_logging_target()` Default: False
- `LoggingToConsole` = True - Immediately starts logging to the console. Default: False
- `LoggingToUdp` = True - Immediately starts logging to the UDP port. Default: False
- `LoggingUdpPort` = 49200 - UDP port to log to. Default: 49200

Parameters

- **resource_name** – VISA resource name, e.g. 'TCPIP::192.168.2.1::INSTR'
- **id_query** – if True, the instrument's model name is verified against the models supported by the driver and eventually throws an exception.
- **reset** – Resets the instrument (sends `*RST` command) and clears its status subsystem.
- **options** – string tokens alternating the driver settings.
- **direct_session** – Another driver object or pyVisa object to reuse the session instead of opening a new session.

static `assert_minimum_version(min_version: str) → None`

Asserts that the driver version fulfills the minimum required version you have entered. This way you make sure your installed driver is of the entered version or newer.

classmethod `clear_global_logging_relative_timestamp() → None`

Clears the global relative timestamp. After this, all the instances using the global relative timestamp continue logging with the absolute timestamps.

close() → None

Closes the active `RsCMPX_UwbMeas` session.

classmethod `from_existing_session(session: object, options: str = None) → RsCMPX_UwbMeas`

Creates a new `RsCMPX_UwbMeas` object with the entered 'session' reused.

Parameters

- **session** – can be another driver or a direct pyvisa session.
- **options** – string tokens alternating the driver settings.

classmethod `get_global_logging_relative_timestamp() → datetime`

Returns global common relative timestamp for log entries.

classmethod `get_global_logging_target()`

Returns global common target stream.

get_session_handle() → object

Returns the underlying session handle.

get_total_execution_time() → timedelta

Returns total time spent by the library on communicating with the instrument. This time is always shorter than `get_total_time()`, since it does not include gaps between the communication. You can reset this counter with `reset_time_statistics()`.

get_total_time() → timedelta

Returns total time spent by the library on communicating with the instrument. This time is always shorter than `get_total_time()`, since it does not include gaps between the communication. You can reset this counter with `reset_time_statistics()`.

static `list_resources(expression: str = '?*::INSTR', visa_select: str = None)` → List[str]

Finds all the resources defined by the expression

- `'?*' - matches all the available instruments`
- `'USB::?*' - matches all the USB instruments`
- `'TCPIP::192?*' - matches all the LAN instruments with the IP address starting with 192`

Parameters

- **expression** – see the examples in the function
- **visa_select** – optional parameter selecting a specific VISA. Examples: `'@ivi'`, `'@rs'`

reset_time_statistics() → None

Resets all execution and total time counters. Affects the results of `get_total_time()` and `get_total_execution_time()`

restore_all_repcaps_to_default() → None

Sets all the Group and Global repcaps to their initial values

classmethod `set_global_logging_relative_timestamp(timestamp: datetime)` → None

Sets global common relative timestamp for log entries. To use it, call the following:
`io.utilities.logger.set_relative_timestamp_global()`

classmethod `set_global_logging_relative_timestamp_now()` → None

Sets global common relative timestamp for log entries to this moment. To use it, call the following:
`io.utilities.logger.set_relative_timestamp_global()`.

classmethod `set_global_logging_target(target)` → None

Sets global common target stream that each instance can use. To use it, call the following:
`io.utilities.logger.set_logging_target_global()`. If an instance uses global logging target, it automatically uses the global relative timestamp (if set). You can set the target to None to invalidate it.

Subgroups

6.1 Catalog

class CatalogCls

Catalog commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.catalog.clone()
```

Subgroups

6.1.1 UwbMeas

SCPI Command :

```
CATalog:UWB:MEAS<instance>:SPATH
```

class UwbMeasCls

UwbMeas commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_spath() → List[str]

```
# SCPI: CATalog:UWB:MEAS<instance>:SPATH
value: List[str] = driver.catalog.uwbMeas.get_spath()
```

Returns the names of the available RF connections.

return
name_signal_path: Comma-separated list of strings, one string per RF connection.

6.2 Configure

class ConfigureCls

Configure commands group definition. 50 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.clone()
```

Subgroups

6.2.1 UwbMeas

class UwbMeasCls

UwbMeas commands group definition. 50 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.clone()
```

Subgroups

6.2.1.1 MultiEval

SCPI Commands :

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:PTRacking
CONFIGure:UWB:MEASurement<Instance>:MEValuation:PMODE
CONFIGure:UWB:MEASurement<Instance>:MEValuation:SCONdition
CONFIGure:UWB:MEASurement<Instance>:MEValuation:TOUT
CONFIGure:UWB:MEASurement<Instance>:MEValuation:SCount
CONFIGure:UWB:MEASurement<Instance>:MEValuation:REPetition
CONFIGure:UWB:MEASurement<Instance>:MEValuation:MOEXception
CONFIGure:UWB:MEASurement<Instance>:MEValuation:CAPLength
CONFIGure:UWB:MEASurement<Instance>:MEValuation:EOfFset
```

class MultiEvalCls

MultiEval commands group definition. 44 total commands, 16 Subgroups, 9 group commands

get_cap_length() → float

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:CAPLength
value: float = driver.configure.uwbMeas.multiEval.get_cap_length()
```

Defines the length to capture the signal.

```
return
capture_length: No help available
```

get_eoffset() → float

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:EOfFset
value: float = driver.configure.uwbMeas.multiEval.get_eoffset()
```

Specifies which time period is excluded from the measurement at the beginning of the capture length.

```
return
eval_offset: No help available
```

get_mo_exception() → bool

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:MOEXception
value: bool = driver.configure.uwbMeas.multiEval.get_mo_exception()
```

Specifies whether measurement results identified as faulty or inaccurate are rejected.

return

meas_on_exception: OFF: Faulty results are rejected. ON: Results are never rejected.

get_pmode() → PpduMode

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PMODE
value: enums.PpduMode = driver.configure.uwbMeas.multiEval.get_pmode()
```

Selects the measurement mode.

return

ppdu_mode: SPPDu: single PPDU packet analysis MPPDu: multi PPDU packet analysis

get_ptracking() → bool

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PTRacking
value: bool = driver.configure.uwbMeas.multiEval.get_ptracking()
```

Enables or disables phase tracking.

return

enable: No help available

get_repetition() → Repeat

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:REPetition
value: enums.Repeat = driver.configure.uwbMeas.multiEval.get_repetition()
```

Specifies the repetition mode of the measurement. The repetition mode specifies whether the measurement is stopped after a single shot or repeated continuously. Use CONFIGure::MEAS<i>::SCOunt to determine the number of measurement intervals per single shot.

return

repetition: SINGleshot: Single-shot measurement CONTinuous: Continuous measurement

get_scondition() → StopCondition

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:SCONdition
value: enums.StopCondition = driver.configure.uwbMeas.multiEval.get_scondition()
```

Qualifies whether the measurement is stopped after a failed limit check or continued. SLFail means that the measurement is stopped and reaches the RDY state when one of the results exceeds the limits.

return

stop_condition: NONE: Continue irrespective of the limit check. SLFail: Stop the measurement on limit failure.

get_scount() → int


```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:SCount
value: int = driver.configure.uwbMeas.multiEval.get_scount()
```

Specifies the statistic count of the measurement. The statistic count is equal to the number of measurement intervals per single shot. The statistic count applies to TX modulation and jitter measurements.

return
 statistic_count: No help available

get_timeout() → float

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:TOUT
value: float = driver.configure.uwbMeas.multiEval.get_timeout()
```

Defines a timeout for the measurement. The timer is started when the measurement is initiated via a READ or INIT command. It is not started if the measurement is initiated manually. When the measurement has completed the first measurement cycle (first single shot), the statistical depth is reached and the timer is reset. If the first measurement cycle has not been completed when the timer expires, the measurement is stopped. The measurement state changes to RDY. The reliability indicator is set to 1, indicating that a measurement timeout occurred. Still running READ, FETCH or CALCulate commands are completed, returning the available results. At least for some results, there are no values at all or the statistical depth has not been reached. A timeout of 0 s corresponds to an infinite measurement timeout.

return
 tcd_timeout: No help available

set_cap_length(capture_length: float) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:CAPLength
driver.configure.uwbMeas.multiEval.set_cap_length(capture_length = 1.0)
```

Defines the length to capture the signal.

param capture_length
 No help available

set_eoffset(eval_offset: float) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:EOffset
driver.configure.uwbMeas.multiEval.set_eoffset(eval_offset = 1.0)
```

Specifies which time period is excluded from the measurement at the beginning of the capture length.

param eval_offset
 No help available

set_mo_exception(meas_on_exception: bool) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:MOEXception
driver.configure.uwbMeas.multiEval.set_mo_exception(meas_on_exception = False)
```

Specifies whether measurement results identified as faulty or inaccurate are rejected.

param meas_on_exception
 OFF: Faulty results are rejected. ON: Results are never rejected.

set_pmode(*ppdu_mode*: *PpduMode*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PMODE
driver.configure.uwbMeas.multiEval.set_pmode(ppdu_mode = enums.PpduMode.MPPDu)
```

Selects the measurement mode.

param ppdu_mode

SPPDu: single PPDU packet analysis MPPDu: multi PPDU packet analysis

set_ptracking(*enable*: *bool*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PTRacking
driver.configure.uwbMeas.multiEval.set_ptracking(enable = False)
```

Enables or disables phase tracking.

param enable

No help available

set_repetition(*repetition*: *Repeat*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:REPetition
driver.configure.uwbMeas.multiEval.set_repetition(repetition = enums.Repeat.
↳CONTINUOUS)
```

Specifies the repetition mode of the measurement. The repetition mode specifies whether the measurement is stopped after a single shot or repeated continuously. Use CONFIGure:...:MEAS<i>:...:SCOUNT to determine the number of measurement intervals per single shot.

param repetition

SINGleshot: Single-shot measurement CONTINUOUS: Continuous measurement

set_scondition(*stop_condition*: *StopCondition*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:SCONdition
driver.configure.uwbMeas.multiEval.set_scondition(stop_condition = enums.
↳StopCondition.NONE)
```

Qualifies whether the measurement is stopped after a failed limit check or continued. SLFail means that the measurement is stopped and reaches the RDY state when one of the results exceeds the limits.

param stop_condition

NONE: Continue irrespective of the limit check. SLFail: Stop the measurement on limit failure.

set_scount(*statistic_count*: *int*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:SCOUNT
driver.configure.uwbMeas.multiEval.set_scount(statistic_count = 1)
```

Specifies the statistic count of the measurement. The statistic count is equal to the number of measurement intervals per single shot. The statistic count applies to TX modulation and jitter measurements.

param statistic_count

No help available

set_timeout(*tcd_timeout: float*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:TOUT
driver.configure.uwbMeas.multiEval.set_timeout(tcd_timeout = 1.0)
```

Defines a timeout for the measurement. The timer is started when the measurement is initiated via a READ or INIT command. It is not started if the measurement is initiated manually. When the measurement has completed the first measurement cycle (first single shot), the statistical depth is reached and the timer is reset. If the first measurement cycle has not been completed when the timer expires, the measurement is stopped. The measurement state changes to RDY. The reliability indicator is set to 1, indicating that a measurement timeout occurred. Still running READ, FETCh or CALCulate commands are completed, returning the available results. At least for some results, there are no values at all or the statistical depth has not been reached. A timeout of 0 s corresponds to an infinite measurement timeout.

param tcd_timeout
No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.clone()
```

Subgroups

6.2.1.1.1 Modulation

class ModulationCls

Modulation commands group definition. 10 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.modulation.clone()
```

Subgroups

6.2.1.1.1.1 Limit

class LimitCls

Limit commands group definition. 10 total commands, 10 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.modulation.limit.clone()
```

Subgroups

6.2.1.1.1.2 CcError

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:MODulation:LIMit:CCERor
```

class CcErrorCls

CcError commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class CcErrorStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → CcErrorStruct

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEvaluation:MODulation:LIMit:CCERor
value: CcErrorStruct = driver.configure.uwbMeas.multiEval.modulation.limit.
↪ ccError.get()
```

Activates and defines an upper limit for the chip clock error.

return

structure: for return value, see the help for CcErrorStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEvaluation:MODulation:LIMit:CCERor
driver.configure.uwbMeas.multiEval.modulation.limit.ccError.set(enable = False,
↪ limit = 1.0)
```

Activates and defines an upper limit for the chip clock error.

param enable

No help available

param limit

No help available

6.2.1.1.1.3 Foffset

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:FOFFset
```

class FoffsetCls

Foffset commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FoffsetStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → FoffsetStruct

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:FOFFset
value: FoffsetStruct = driver.configure.uwbMeas.multiEval.modulation.limit.
↳ foffset.get()
```

Activates and defines an upper limit for the center frequency error.

return

structure: for return value, see the help for FoffsetStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:FOFFset
driver.configure.uwbMeas.multiEval.modulation.limit.foffset.set(enable = False,
↳ limit = 1.0)
```

Activates and defines an upper limit for the center frequency error.

param enable

No help available

param limit

No help available

6.2.1.1.1.4 Phr

class PhrCls

Phr commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.modulation.limit.phr.clone()
```

Subgroups

6.2.1.1.1.5 Nrmse

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEvaluation:MODulation:LIMit:PHR:NRMSe
```

class NrmseCls

Nrmse commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class NrmseStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → NrmseStruct

```
# SCPI: CONFigure:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:LIMit:PHR:NRMSe
value: NrmseStruct = driver.configure.uwbMeas.multiEval.modulation.limit.phr.
↳nrmse.get()
```

Activates and defines an upper limit for the NRMSE of the PHR.

return

structure: for return value, see the help for NrmseStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:LIMit:PHR:NRMSe
driver.configure.uwbMeas.multiEval.modulation.limit.phr.nrmse.set(enable =
↳False, limit = 1.0)
```

Activates and defines an upper limit for the NRMSE of the PHR.

param enable

No help available

param limit

No help available

6.2.1.1.1.6 Plevel

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:PLEvel
```

class PlevelCls

Plevel commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PlevelStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → PlevelStruct

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:PLEvel
value: PlevelStruct = driver.configure.uwbMeas.multiEval.modulation.limit.
↳plevel.get()
```

Activates and defines a symmetrical upper and lower limit for the pulse levels relative to the SHR pulse level.

return

structure: for return value, see the help for PlevelStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:PLEvel
driver.configure.uwbMeas.multiEval.modulation.limit.plevel.set(enable = False,
↳limit = 1.0)
```

Activates and defines a symmetrical upper and lower limit for the pulse levels relative to the SHR pulse level.

param enable

No help available

param limit

No help available

6.2.1.1.1.7 PmlWidth

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:PMLWidth
```

class PmlWidthCls

PmlWidth commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PmlWidthStruct

Response structure. Fields:

- Enable: bool: No parameter help available

- Limit: float: No parameter help available

get() → PmlWidthStruct

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↪ :MEValuation:MODulation:LIMit:PMLWidth
value: PmlWidthStruct = driver.configure.uwbMeas.multiEval.modulation.limit.
↪ pmlWidth.get()
```

Activates and defines a lower limit for the pulse mainlobe width.

return

structure: for return value, see the help for PmlWidthStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↪ :MEValuation:MODulation:LIMit:PMLWidth
driver.configure.uwbMeas.multiEval.modulation.limit.pmlWidth.set(enable = False,
↪ limit = 1.0)
```

Activates and defines a lower limit for the pulse mainlobe width.

param enable

No help available

param limit

No help available

6.2.1.1.1.8 Psdu

class PsduCls

Psdu commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.modulation.limit.psdu.clone()
```

Subgroups

6.2.1.1.1.9 Nrmse

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:PSDU:NRMSe
```

class NrmseCls

Nrmse commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class NrmseStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → NrmseStruct

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:LIMit:PSDU:NRMSe
value: NrmseStruct = driver.configure.uwbMeas.multiEval.modulation.limit.psdu.
↳nrmse.get()
```

Activates and defines an upper limit for the NRMSE of the PSDU.

return

structure: for return value, see the help for NrmseStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:LIMit:PSDU:NRMSe
driver.configure.uwbMeas.multiEval.modulation.limit.psdu.nrmse.set(enable =
↳False, limit = 1.0)
```

Activates and defines an upper limit for the NRMSE of the PSDU.

param enable

No help available

param limit

No help available

6.2.1.1.10 Shr**class ShrCls**

Shr commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.modulation.limit.shr.clone()
```

Subgroups**6.2.1.1.11 Nrmse****SCPI Command :**

```
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:MODulation:LIMit:SHR:NRMSe
```

class NrmseCls

Nrmse commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class NrmseStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → NrmseStruct

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:LIMit:SHR:NRMSe
value: NrmseStruct = driver.configure.uwbMeas.multiEval.modulation.limit.shr.
↳nrmse.get()
```

Activates and defines an upper limit for the NRMSE of the SHR.

return

structure: for return value, see the help for NrmseStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:LIMit:SHR:NRMSe
driver.configure.uwbMeas.multiEval.modulation.limit.shr.nrmse.set(enable =
↳False, limit = 1.0)
```

Activates and defines an upper limit for the NRMSE of the SHR.

param enable

No help available

param limit

No help available

6.2.1.1.1.12 SIpeak**SCPI Command :**

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:SLPeak
```

class SIpeakCls

SIpeak commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class SIpeakStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → SIpeakStruct

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:SLPeak
value: SIpeakStruct = driver.configure.uwbMeas.multiEval.modulation.limit.
↳slpeak.get()
```

Activates and defines an upper limit for the pulse sidelobe peak.

return

structure: for return value, see the help for SlPeakStruct structure arguments.

set(*enable: bool, limit: float*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:SLPeak
driver.configure.uwbMeas.multiEval.modulation.limit.slPeak.set(enable = False,
↪ limit = 1.0)
```

Activates and defines an upper limit for the pulse sidelobe peak.

param enable

No help available

param limit

No help available

6.2.1.1.1.13 SmAccuracy

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:MODulation:LIMit:SMACcuracy
```

class SmAccuracyCls

SmAccuracy commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class SmAccuracyStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → SmAccuracyStruct

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↪:MEValuation:MODulation:LIMit:SMACcuracy
value: SmAccuracyStruct = driver.configure.uwbMeas.multiEval.modulation.limit.
↪ smAccuracy.get()
```

Activates and defines a lower limit for the symbol modulation accuracy.

return

structure: for return value, see the help for SmAccuracyStruct structure arguments.

set(*enable: bool, limit: float*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↪:MEValuation:MODulation:LIMit:SMACcuracy
driver.configure.uwbMeas.multiEval.modulation.limit.smAccuracy.set(enable =
↪ False, limit = 1.0)
```

Activates and defines a lower limit for the symbol modulation accuracy.

param enable

No help available

param limit
No help available

6.2.1.1.1.14 Sts

class StsCls

Sts commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.modulation.limit.sts.clone()
```

Subgroups

6.2.1.1.1.15 Nrmse

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:MODulation:LIMit:STS:NRMSe
```

class NrmseCls

Nrmse commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class NrmseStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → NrmseStruct

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:LIMit:STS:NRMSe
value: NrmseStruct = driver.configure.uwbMeas.multiEval.modulation.limit.sts.
↳nrmse.get()
```

Activates and defines an upper limit for the NRMSE of the STS.

return

structure: for return value, see the help for NrmseStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:LIMit:STS:NRMSe
driver.configure.uwbMeas.multiEval.modulation.limit.sts.nrmse.set(enable =
↳False, limit = 1.0)
```

Activates and defines an upper limit for the NRMSE of the STS.

param enable
No help available

param limit
No help available

6.2.1.1.2 MprFrequency

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:MPRFrequency<Record>
```

class MprFrequencyCls

MprFrequency commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(record=Record.Nr1) → str

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEvaluation:MPRFrequency<Record>
value: str = driver.configure.uwbMeas.multiEval.mprFrequency.get(record =
↳repcap.Record.Nr1)
```

Queries the mean pulse repetition frequency.

param record
optional repeated capability selector. Default value: Nr1

return
mpr_frequency: No help available

6.2.1.1.3 Phr

class PhrCls

Phr commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.phr.clone()
```

Subgroups

6.2.1.1.3.1 Bitrate<Record>

RepCap Settings

```
# Range: Nr1 .. Nr20
rc = driver.configure.uwbMeas.multiEval.phr.bitrate.repcap_record_get()
driver.configure.uwbMeas.multiEval.phr.bitrate.repcap_record_set(repcap.Record.Nr1)
```

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEvaluation:PHR:BITRate<Record>
```

class BitrateCls

Bitrate commands group definition. 1 total commands, 0 Subgroups, 1 group commands Repeated Capability: Record, default value after init: Record.Nr1

get(record=Record.Default) → str

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEvaluation:PHR:BITRate<Record>
value: str = driver.configure.uwbMeas.multiEval.phr.bitrate.get(record = repcap.
↳Record.Default)
```

Queries the data rate of the PHR.

param record

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Bitrate')

return

phr_bitrate: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.phr.bitrate.clone()
```

6.2.1.1.4 PhrRate**SCPI Command :**

```
CONFigure:UWB:MEASurement<Instance>:MEvaluation:PHRRate<Record>
```

class PhrRateCls

PhrRate commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(record=Record.Nr1) → PhrDataRate

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEvaluation:PHRRate<Record>
value: enums.PhrDataRate = driver.configure.uwbMeas.multiEval.phrRate.
↳get(record = repcap.Record.Nr1)
```

Specifies the data rate of the PHY header (PHR) or selects the type of signal if there is no PHR.

param record

optional repeated capability selector. Default value: Nr1

return

phr_data_rate: DRMD: 110 kb/s or 850 kb/s (DRMDR) DRLP: 850 kb/s (DRBM_LP) DRHP: 6.8 Mb/s (DRBM_HP) RHML: 3.9 Mb/s or 7.8 Mb/s (DRHM_LR) RHMH: 15.6 Mb/s or 31.2 Mb/s (DRHM_HR) SYNC: PPDU with SYNC field only (SYNC_ONLY) RSF: ranging sequence fragment IGN: ignore selected PPDU SYFD: PPDU with SYNC and SFD fields only (SYNC_SFD) NBAND: narrowband signal

set(*phr_data_rate*: *PhrDataRate*, *record*=*Record.Nr1*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PHRRate<Record>
driver.configure.uwbMeas.multiEval.phrRate.set(phr_data_rate = enums.
↪PhrDataRate.DRHP, record = repcap.Record.Nr1)
```

Specifies the data rate of the PHY header (PHR) or selects the type of signal if there is no PHR.

param phr_data_rate

DRMD: 110 kb/s or 850 kb/s (DRMDR) DRLP: 850 kb/s (DRBM_LP) DRHP: 6.8 Mb/s (DRBM_HP) RHML: 3.9 Mb/s or 7.8 Mb/s (DRHM_LR) RHMH: 15.6 Mb/s or 31.2 Mb/s (DRHM_HR) SYNC: PPDU with SYNC field only (SYNC_ONLY) RSF: ranging sequence fragment IGN: ignore selected PPDU SYFD: PPDU with SYNC and SFD fields only (SYNC_SFD) NBANd: narrowband signal

param record

optional repeated capability selector. Default value: Nr1

6.2.1.1.5 Pmask

class PmaskCls

Pmask commands group definition. 3 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.pmask.clone()
```

Subgroups

6.2.1.1.5.1 Limit

class LimitCls

Limit commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.pmask.limit.clone()
```

Subgroups

6.2.1.1.5.2 Area

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEValuation:PMASk:LIMit:AREA
```

class AreaCls

Area commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class AreaStruct

Response structure. Fields:

- Enable_Lower: bool: Enables the check of lower limits
- Enable_Upper: bool: Enables the check of upper limits

get() → AreaStruct

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:PMASk:LIMit:AREA
value: AreaStruct = driver.configure.uwbMeas.multiEval.pmask.limit.area.get()
```

Enables limit checks for the pulse mask.

return

structure: for return value, see the help for AreaStruct structure arguments.

set(enable_lower: bool, enable_upper: bool) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:PMASk:LIMit:AREA
driver.configure.uwbMeas.multiEval.pmask.limit.area.set(enable_lower = False,
enable_upper = False)
```

Enables limit checks for the pulse mask.

param enable_lower

Enables the check of lower limits

param enable_upper

Enables the check of upper limits

6.2.1.1.5.3 Lower

class LowerCls

Lower commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.pmask.limit.lower.clone()
```

Subgroups

6.2.1.1.5.4 Area<Area>

RepCap Settings

```
# Range: Nr1 .. Nr3
rc = driver.configure.uwbMeas.multiEval.pmask.limit.lower.area.repcap_area_get()
driver.configure.uwbMeas.multiEval.pmask.limit.lower.area.repcap_area_set(repcap.Area.
↳Nr1)
```

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEvaluation:PMASk:LIMit:LOWer:AREA<nr>
```

class AreaCls

Area commands group definition. 1 total commands, 0 Subgroups, 1 group commands Repeated Capability: Area, default value after init: Area.Nr1

get(area=Area.Default) → float

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEvaluation:PMASk:LIMit:LOWer:AREA
↳<nr>
value: float = driver.configure.uwbMeas.multiEval.pmask.limit.lower.area.
↳get(area = repcap.Area.Default)
```

Defines lower limits for the three areas of the pulse mask.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

return

area_limit: No help available

set(area_limit: float, area=Area.Default) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEvaluation:PMASk:LIMit:LOWer:AREA
↳<nr>
driver.configure.uwbMeas.multiEval.pmask.limit.lower.area.set(area_limit = 1.0,
↳area = repcap.Area.Default)
```

Defines lower limits for the three areas of the pulse mask.

param area_limit

No help available

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.pmask.limit.lower.area.clone()
```

6.2.1.1.5.5 Upper**class UpperCls**

Upper commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.pmask.limit.upper.clone()
```

Subgroups**6.2.1.1.5.6 Area<Area>****RepCap Settings**

```
# Range: Nr1 .. Nr3
rc = driver.configure.uwbMeas.multiEval.pmask.limit.upper.area.repcap_area_get()
driver.configure.uwbMeas.multiEval.pmask.limit.upper.area.repcap_area_set(repcap.Area.
↪Nr1)
```

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:PMASk:LIMit:UPPer:AREA<nr>
```

class AreaCls

Area commands group definition. 1 total commands, 0 Subgroups, 1 group commands Repeated Capability: Area, default value after init: Area.Nr1

get(area=Area.Default) → float

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEvaluation:PMASk:LIMit:UPPer:AREA
↪<nr>
value: float = driver.configure.uwbMeas.multiEval.pmask.limit.upper.area.
↪get(area = repcap.Area.Default)
```

Defines upper limits for the three areas of the pulse mask.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

return

area_limit: Area 2 fixed to 1. Area 3 minimum = 0.

set(area_limit: float, area=Area.Default) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PMASk:LIMit:UPPer:AREA
↳<nr>
driver.configure.uwbMeas.multiEval.pmask.limit.upper.area.set(area_limit = 1.0,
↳area = repcap.Area.Default)
```

Defines upper limits for the three areas of the pulse mask.

param area_limit

Area 2 fixed to 1. Area 3 minimum = 0.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.pmask.limit.upper.area.clone()
```

6.2.1.1.6 Ppdu

SCPI Commands :

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:RECORDs
CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:SRECORD
CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:NUMBER
```

class PpduCls

Ppdu commands group definition. 3 total commands, 0 Subgroups, 3 group commands

get_number() → int

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:NUMBER
value: int = driver.configure.uwbMeas.multiEval.ppdu.get_number()
```

No command help available

return

ppdu_number: No help available

get_records() → int

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:RECORDs
value: int = driver.configure.uwbMeas.multiEval.ppdu.get_records()
```

Defines the number of PPDU's for configuration.

return
number_pp_du: No help available

get_srecord() → int

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:SREcord
value: int = driver.configure.uwbMeas.multiEval.ppdu.get_srecord()
```

Selects one PPDU for display and configuration via the GUI.

return
selected_record: No help available

set_number(ppdu_number: int) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:NUMBER
driver.configure.uwbMeas.multiEval.ppdu.set_number(ppdu_number = 1)
```

No command help available

param ppdu_number
No help available

set_records(number_pp_du: int) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:RECORDs
driver.configure.uwbMeas.multiEval.ppdu.set_records(number_pp_du = 1)
```

Defines the number of PPDU's for configuration.

param number_pp_du
No help available

set_srecord(selected_record: int) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPDU:SRECORD
driver.configure.uwbMeas.multiEval.ppdu.set_srecord(selected_record = 1)
```

Selects one PPDU for display and configuration via the GUI.

param selected_record
No help available

6.2.1.1.7 PpLength

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPLength<Record>
```

class PpLengthCls

PpLength commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(record=Record.Nr1) → int

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPLength<Record>
value: int = driver.configure.uwbMeas.multiEval.ppLength.get(record = repcap.
↳Record.Nr1)
```

Specifies the bit length of the PHR payload length field. This setting is only relevant in HPRF mode (RHML or RHHM set via CONFIGure:UWB:MEAS<i>:MEValuation:PHRRate<Record>).

param record

optional repeated capability selector. Default value: Nr1

return

phr_payload_len: No help available

set(phr_payload_len: int, record=Record.Nr1) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PPLength<Record>
driver.configure.uwbMeas.multiEval.ppLength.set(phr_payload_len = 1, record =
↳repcap.Record.Nr1)
```

Specifies the bit length of the PHR payload length field. This setting is only relevant in HPRF mode (RHML or RHHM set via CONFIGure:UWB:MEAS<i>:MEValuation:PHRRate<Record>).

param phr_payload_len

No help available

param record

optional repeated capability selector. Default value: Nr1

6.2.1.1.8 PrfMode

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:PRFMode<Record>
```

class PrfModeCls

PrfMode commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(record=Record.Nr1) → str

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:PRFMode<Record>
value: str = driver.configure.uwbMeas.multiEval.prfMode.get(record = repcap.
↳Record.Nr1)
```

Queries the pulse repetition frequency mode.

param record

optional repeated capability selector. Default value: Nr1

return

prf_mode: 'BPRF', 'HPRF' or '—'

6.2.1.1.9 Psdu

class PsduCls

Psdu commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.psdu.clone()
```

Subgroups

6.2.1.1.9.1 Bitrate<Record>

RepCap Settings

```
# Range: Nr1 .. Nr20
rc = driver.configure.uwbMeas.multiEval.psdu.bitrate.repcap_record_get()
driver.configure.uwbMeas.multiEval.psdu.bitrate.repcap_record_set(repcap.Record.Nr1)
```

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:PSDU:BITRate<Record>
```

class BitrateCls

Bitrate commands group definition. 1 total commands, 0 Subgroups, 1 group commands Repeated Capability: Record, default value after init: Record.Nr1

get(record=Record.Default) → str

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEvaluation:PSDU:BITRate<Record>
value: str = driver.configure.uwbMeas.multiEval.psdu.bitrate.get(record = ↵
↵repcap.Record.Default)
```

Queries the PSDU bit rate.

param record

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Bitrate’)

return

psdu_bitrate: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.psd.bitrates.clone()
```

6.2.1.1.10 PsFormat

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:PSFormat<Record>
```

class PsFormatCls

PsFormat commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(record=Record.Nr1) → int

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEvaluation:PSFormat<Record>
value: int = driver.configure.uwbMeas.multiEval.psFormat.get(record = reprecap.
↳Record.Nr1)
```

Specifies the PPDU STS packet structure configuration. See also ‘HRP-ERDEV’.

param record

optional repeated capability selector. Default value: Nr1

return

ppdu_sts_format: No help available

set(ppdu_sts_format: int, record=Record.Nr1) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEvaluation:PSFormat<Record>
driver.configure.uwbMeas.multiEval.psFormat.set(ppdu_sts_format = 1, record =
↳reprecap.Record.Nr1)
```

Specifies the PPDU STS packet structure configuration. See also ‘HRP-ERDEV’.

param ppdu_sts_format

No help available

param record

optional repeated capability selector. Default value: Nr1

6.2.1.1.11 Result

SCPI Commands :

```
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:RESult:TSMask
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:RESult:PVTime
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:RESult:EMODulation
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:RESult:DDECoding
```

class ResultCls

Result commands group definition. 4 total commands, 0 Subgroups, 4 group commands

get_ddecoding() → bool

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:RESult:DDECoding
value: bool = driver.configure.uwbMeas.multiEval.result.get_ddecoding()
```

Enables or disables the evaluation of the PPDU payload contents.

return

enable: OFF: Do not evaluate results. ON: Evaluate the results.

get_emodulation() → bool

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:RESult:EMODulation
value: bool = driver.configure.uwbMeas.multiEval.result.get_emodulation()
```

Enables or disables the evaluation of modulation and jitter results.

return

enable: OFF: Do not evaluate results. ON: Evaluate the results.

get_power_vs_time() → bool

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:RESult:PVTime
value: bool = driver.configure.uwbMeas.multiEval.result.get_power_vs_time()
```

Enables or disables the evaluation of power results.

return

enable: OFF: Do not evaluate results. ON: Evaluate the results.

get_ts_mask() → bool

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:RESult:TSMask
value: bool = driver.configure.uwbMeas.multiEval.result.get_ts_mask()
```

Enables or disables the evaluation of spectrum results.

return

enable: OFF: Do not evaluate results. ON: Evaluate the results.

set_ddecoding(enable: bool) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:RESult:DDECoding
driver.configure.uwbMeas.multiEval.result.set_ddecoding(enable = False)
```

Enables or disables the evaluation of the PPDU payload contents.

param enable

OFF: Do not evaluate results. ON: Evaluate the results.

set_emodulation(enable: bool) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:RESult:EMODulation
driver.configure.uwbMeas.multiEval.result.set_emodulation(enable = False)
```

Enables or disables the evaluation of modulation and jitter results.

param enable

OFF: Do not evaluate results. ON: Evaluate the results.

set_power_vs_time(*enable: bool*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:RESult:PVTime
driver.configure.uwbMeas.multiEval.result.set_power_vs_time(enable = False)
```

Enables or disables the evaluation of power results.

param enable

OFF: Do not evaluate results. ON: Evaluate the results.

set_ts_mask(*enable: bool*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:RESult:TSMask
driver.configure.uwbMeas.multiEval.result.set_ts_mask(enable = False)
```

Enables or disables the evaluation of spectrum results.

param enable

OFF: Do not evaluate results. ON: Evaluate the results.

6.2.1.12 Spectrum

SCPI Commands :

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:SPECtrum:SCount
CONFIGure:UWB:MEASurement<Instance>:MEValuation:SPECtrum:MSPLength
```

class SpectrumCls

Spectrum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

get_msp_length() → MaxSpecPowLen

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:SPECtrum:MSPLength
value: enums.MaxSpecPowLen = driver.configure.uwbMeas.multiEval.spectrum.get_
↳msp_length()
```

Selects the time interval for measuring the maximum spectral power.

return

max_spec_pow_len: PPDU: entire PPDU MS1: 1 ms, starting with the PPDU

get_scount() → int

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:SPECtrum:SCount
value: int = driver.configure.uwbMeas.multiEval.spectrum.get_scount()
```

Specifies the statistic count of the measurement. The statistic count is equal to the number of measurement intervals per single shot. The statistic count applies to spectrum measurements.

return

statistic_count: No help available

set_msp_length(*max_spec_pow_len: MaxSpecPowLen*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:SPECTrum:MSPLength
driver.configure.uwbMeas.multiEval.spectrum.set_msp_length(max_spec_pow_len =
↳ enums.MaxSpecPowLen.MS1)
```

Selects the time interval for measuring the maximum spectral power.

param max_spec_pow_len

PPDU: entire PPDU MS1: 1 ms, starting with the PPDU

set_scount(*statistic_count: int*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:SPECTrum:SCount
driver.configure.uwbMeas.multiEval.spectrum.set_scount(statistic_count = 1)
```

Specifies the statistic count of the measurement. The statistic count is equal to the number of measurement intervals per single shot. The statistic count applies to spectrum measurements.

param statistic_count

No help available

6.2.1.1.13 StSegments

SCPI Command :

```
CONFIGure:UWB:MEASurement<Instance>:MEValuation:STSegments<Record>
```

class StSegmentsCls

StSegments commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(*record=Record.Nr1*) → int

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:STSegments<Record>
value: int = driver.configure.uwbMeas.multiEval.stSegments.get(record = repcap.
↳ Record.Nr1)
```

Specifies the number of STS segments inserted according to the STS packet configuration.

param record

optional repeated capability selector. Default value: Nr1

return

no_sts_segments: No help available

set(*no_sts_segments: int, record=Record.Nr1*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:MEValuation:STSegments<Record>
driver.configure.uwbMeas.multiEval.stSegments.set(no_sts_segments = 1, record =
↳ repcap.Record.Nr1)
```

Specifies the number of STS segments inserted according to the STS packet configuration.

param no_sts_segments

No help available

param record

optional repeated capability selector. Default value: Nr1

6.2.1.1.14 StsGap

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEValuation:STSGap<Record>
```

class StsGapCls

StsGap commands group definition. 2 total commands, 1 Subgroups, 1 group commands

get(record=Record.Nr1) → int

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:STSGap<Record>
value: int = driver.configure.uwbMeas.multiEval.stsGap.get(record = repcap.
↳Record.Nr1)
```

Specifies additional gaps between the payload and the STS in units of 4 chips. This setting is only relevant for PPDU STS packet structure configuration two set via CONFigure:UWB:MEAS<i>:MEValuation:PSFormat<Record>).

param record

optional repeated capability selector. Default value: Nr1

return

sts_gap: No help available

set(sts_gap: int, record=Record.Nr1) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:STSGap<Record>
driver.configure.uwbMeas.multiEval.stsGap.set(sts_gap = 1, record = repcap.
↳Record.Nr1)
```

Specifies additional gaps between the payload and the STS in units of 4 chips. This setting is only relevant for PPDU STS packet structure configuration two set via CONFigure:UWB:MEAS<i>:MEValuation:PSFormat<Record>).

param sts_gap

No help available

param record

optional repeated capability selector. Default value: Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.stsGap.clone()
```

Subgroups

6.2.1.1.14.1 Chip

SCPI Command :

CONFigure:UWB:MEASurement<Instance>:MEValuation:STSGap:CHIP<Record>

class ChipCls

Chip commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(record=Record.Nr1) → int

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:STSGap:CHIP<Record>
value: int = driver.configure.uwbMeas.multiEval.stsGap.chip.get(record = repcap.
↳Record.Nr1)
```

Queries the number of chips for the configured STS gap, resulting from CONFigure:UWB:MEAS*<i>*:MEValuation:STSGap<Record>.

param record

optional repeated capability selector. Default value: Nr1

return

sts_gap_chip: No help available

6.2.1.1.15 StsLength

SCPI Command :

CONFigure:UWB:MEASurement<Instance>:MEValuation:STSLength<Record>

class StsLengthCls

StsLength commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(record=Record.Nr1) → StsSegmentLen

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:STSLength<Record>
value: enums.StsSegmentLen = driver.configure.uwbMeas.multiEval.stsLength.
↳get(record = repcap.Record.Nr1)
```

Specifies the length of the STS segment in units of 512 chips.

param record

optional repeated capability selector. Default value: Nr1

return

sts_segment_len: No help available

set(sts_segment_len: StsSegmentLen, record=Record.Nr1) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEValuation:STSLength<Record>
driver.configure.uwbMeas.multiEval.stsLength.set(sts_segment_len = enums.
↳StsSegmentLen.L128, record = repcap.Record.Nr1)
```

Specifies the length of the STS segment in units of 512 chips.

param sts_segment_len

No help available

param record

optional repeated capability selector. Default value: Nr1

6.2.1.1.16 TsMask

class TsMaskCls

TsMask commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.tsMask.clone()
```

Subgroups

6.2.1.1.16.1 Limit

class LimitCls

Limit commands group definition. 2 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.tsMask.limit.clone()
```

Subgroups

6.2.1.1.16.2 Area<Area>

RepCap Settings

```
# Range: Nr1 .. Nr3
rc = driver.configure.uwbMeas.multiEval.tsMask.limit.area.repcap_area_get()
driver.configure.uwbMeas.multiEval.tsMask.limit.area.repcap_area_set(repcap.Area.Nr1)
```

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEvaluation:TSMask:LIMit:AREA<nr>
```

class AreaCls

Area commands group definition. 1 total commands, 0 Subgroups, 1 group commands Repeated Capability: Area, default value after init: Area.Nr1

class AreaStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Area_Limit: float: No parameter help available

get(*area=Area.Default*) → AreaStruct

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEvaluation:TSMask:LIMit:AREA<nr>
value: AreaStruct = driver.configure.uwbMeas.multiEval.tsMask.limit.area.
↪get(area = repcap.Area.Default)
```

Activates and defines an upper limit for the two areas of the spectral mask.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

return

structure: for return value, see the help for AreaStruct structure arguments.

set(*enable: bool, area_limit: float = None, area=Area.Default*) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:MEvaluation:TSMask:LIMit:AREA<nr>
driver.configure.uwbMeas.multiEval.tsMask.limit.area.set(enable = False, area_
↪limit = 1.0, area = repcap.Area.Default)
```

Activates and defines an upper limit for the two areas of the spectral mask.

param enable

No help available

param area_limit

No help available

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.multiEval.tsMask.limit.area.clone()
```

6.2.1.1.16.3 TdbBandwidth

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:MEvaluation:TSMask:LIMit:TDBBandwidth
```

class TdbBandwidthCls

TdbBandwidth commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class TdbBandwidthStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Limit: float: No parameter help available

get() → TdbBandwidthStruct

```
# SCPI: CONFigure:UWB:MEASurement<Instance>
↳:MEvaluation:TSMask:LIMit:TDBBandwidth
value: TdbBandwidthStruct = driver.configure.uwbMeas.multiEval.tsMask.limit.
↳tdbBandwidth.get()
```

Activates and defines an upper limit for the -10 dB bandwidth.

return

structure: for return value, see the help for TdbBandwidthStruct structure arguments.

set(enable: bool, limit: float) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>
↳:MEvaluation:TSMask:LIMit:TDBBandwidth
driver.configure.uwbMeas.multiEval.tsMask.limit.tdbBandwidth.set(enable = False,
↳ limit = 1.0)
```

Activates and defines an upper limit for the -10 dB bandwidth.

param enable

No help available

param limit

No help available

6.2.1.2 RfSettings

SCPI Commands :

```
CONFigure:UWB:MEASurement<Instance>:RFSettings:CHANnel
CONFigure:UWB:MEASurement<Instance>:RFSettings:ENPower
CONFigure:UWB:MEASurement<Instance>:RFSettings:EATTenuation
CONFigure:UWB:MEASurement<Instance>:RFSettings:UMARgin
```

class RfSettingsCls

RfSettings commands group definition. 6 total commands, 1 Subgroups, 4 group commands

get_channel() → int

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:RFSettings:CHANnel
value: int = driver.configure.uwbMeas.rfSettings.get_channel()
```

Selects the channel number.

return
analyzer_chan: No help available

get_eattenuation() → float

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:RFSettings:EATTenuation
value: float = driver.configure.uwbMeas.rfSettings.get_eattenuation()
```

Defines an external attenuation (or gain, if the value is negative) , to be applied to the input connector. For measurement of UWB signals.

return
rf_input_ext_att: No help available

get_envelope_power() → float

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:RFSettings:ENPower
value: float = driver.configure.uwbMeas.rfSettings.get_envelope_power()
```

Sets the expected nominal power of the measured UWB signal.

return
exp_nominal_power: The range of the expected nominal power can be calculated as follows: Range (Expected Nominal Power) = Range (Input Power) + External Attenuation - User Margin The input power range is stated in the specifications document.

get_umargin() → float

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:RFSettings:UMARgin
value: float = driver.configure.uwbMeas.rfSettings.get_umargin()
```

Sets the margin that the measurement adds to the expected nominal power to determine the reference power. The reference power minus the external input attenuation must be within the power range of the selected input connector. Refer to the specifications document. For measurement of UWB signals.

return
user_margin: No help available

set_channel(*analyzer_chan: int*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:RFSettings:CHANnel
driver.configure.uwbMeas.rfSettings.set_channel(analyzer_chan = 1)
```

Selects the channel number.

param analyzer_chan

No help available

set_eattenuation(*rf_input_ext_att: float*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:RFSettings:EATTenuation
driver.configure.uwbMeas.rfSettings.set_eattenuation(rf_input_ext_att = 1.0)
```

Defines an external attenuation (or gain, if the value is negative) , to be applied to the input connector. For measurement of UWB signals.

param rf_input_ext_att

No help available

set_envelope_power(*exp_nominal_power: float*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:RFSettings:ENPower
driver.configure.uwbMeas.rfSettings.set_envelope_power(exp_nominal_power = 1.0)
```

Sets the expected nominal power of the measured UWB signal.

param exp_nominal_power

The range of the expected nominal power can be calculated as follows: Range (Expected Nominal Power) = Range (Input Power) + External Attenuation - User Margin

The input power range is stated in the specifications document.

set_umargin(*user_margin: float*) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:RFSettings:UMARgin
driver.configure.uwbMeas.rfSettings.set_umargin(user_margin = 1.0)
```

Sets the margin that the measurement adds to the expected nominal power to determine the reference power. The reference power minus the external input attenuation must be within the power range of the selected input connector. Refer to the specifications document. For measurement of UWB signals.

param user_margin

No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.rfSettings.clone()
```

Subgroups

6.2.1.2.1 Frequency

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:RFSettings:FREquency
```

class FrequencyCls

Frequency commands group definition. 2 total commands, 1 Subgroups, 1 group commands

get_value() → float

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:RFSettings:FREquency
value: float = driver.configure.uwbMeas.rfSettings.frequency.get_value()
```

Selects the center frequency of the measured carrier for UWB signals. For the supported frequency range, see 'Frequency ranges'.

return
analyzer_freq: No help available

set_value(analyzer_freq: float) → None

```
# SCPI: CONFigure:UWB:MEASurement<Instance>:RFSettings:FREquency
driver.configure.uwbMeas.rfSettings.frequency.set_value(analyzer_freq = 1.0)
```

Selects the center frequency of the measured carrier for UWB signals. For the supported frequency range, see 'Frequency ranges'.

param analyzer_freq
No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.uwbMeas.rfSettings.frequency.clone()
```

Subgroups

6.2.1.2.1.1 Range

SCPI Command :

```
CONFigure:UWB:MEASurement<Instance>:RFSettings:FREquency:RANGe
```

class RangeCls

Range commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class RangeStruct

Response structure. Fields:

- Min_Py: int: No parameter help available

- Max_Py: int: No parameter help available

get() → RangeStruct

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:RFSettings:FREQuency:RANGe
value: RangeStruct = driver.configure.uwbMeas.rfSettings.frequency.range.get()
```

No command help available

return

structure: for return value, see the help for RangeStruct structure arguments.

set(min_py: int, max_py: int) → None

```
# SCPI: CONFIGure:UWB:MEASurement<Instance>:RFSettings:FREQuency:RANGe
driver.configure.uwbMeas.rfSettings.frequency.range.set(min_py = 1, max_py = 1)
```

No command help available

param min_py

No help available

param max_py

No help available

6.3 Route

class RouteCls

Route commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.route.clone()
```

Subgroups

6.3.1 UwbMeas

SCPI Command :

```
ROUTE:UWB:MEAS<instance>:SPATH
```

class UwbMeasCls

UwbMeas commands group definition. 2 total commands, 1 Subgroups, 1 group commands

get_spath() → str

```
# SCPI: ROUTe:UWB:MEAS<instance>:SPATH
value: str = driver.route.uwbMeas.get_spath()
```

Selects the RF connection (signal input path) for the measured signal. For possible connection names, see method RsCMPX_UwbMeas.Catalog.UwbMeas.spath.

return
signal_path: No help available

set_spath(signal_path: str) → None

```
# SCPI: ROUTe:UWB:MEAS<instance>:SPATH
driver.route.uwbMeas.set_spath(signal_path = 'abc')
```

Selects the RF connection (signal input path) for the measured signal. For possible connection names, see method RsCMPX_UwbMeas.Catalog.UwbMeas.spath.

param signal_path
No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.route.uwbMeas.clone()
```

Subgroups

6.3.1.1 RfSettings

SCPI Command :

```
ROUTE:UWB:MEASurement<Instance>:RFSettings:CONNECTor
```

class RfSettingsCls

RfSettings commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_connector() → Connector

```
# SCPI: ROUTe:UWB:MEASurement<Instance>:RFSettings:CONNECTor
value: enums.Connector = driver.route.uwbMeas.rfSettings.get_connector()
```

No command help available

return
connector: No help available

set_connector(connector: Connector) → None

```
# SCPI: ROUTe:UWB:MEASurement<Instance>:RFSettings:CONNECTor
driver.route.uwbMeas.rfSettings.set_connector(connector = enums.Connector.I11I)
```

No command help available

param connector
No help available

6.4 Trigger

class TriggerCls

Trigger commands group definition. 7 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.trigger.clone()
```

Subgroups

6.4.1 UwbMeas

class UwbMeasCls

UwbMeas commands group definition. 7 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.trigger.uwbMeas.clone()
```

Subgroups

6.4.1.1 MultiEval

SCPI Commands :

```
TRIGger:UWB:MEASurement<Instance>:MEValuation:SOURce
TRIGger:UWB:MEASurement<Instance>:MEValuation:THReshold
TRIGger:UWB:MEASurement<Instance>:MEValuation:TOUT
TRIGger:UWB:MEASurement<Instance>:MEValuation:MGAP
TRIGger:UWB:MEASurement<Instance>:MEValuation:SLOPe
TRIGger:UWB:MEASurement<Instance>:MEValuation:DElay
```

class MultiEvalCls

MultiEval commands group definition. 7 total commands, 1 Subgroups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEValuation:DElay
value: float = driver.trigger.uwbMeas.multiEval.get_delay()
```

Defines a time delaying the start of the measurement relative to the trigger event. This setting has no influence on free run measurements.

return

delay: No help available

get_mgap() → float

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:MGAP
value: float = driver.trigger.uwbMeas.multiEval.get_mgap()
```

Sets a minimum time during which the IF signal must be below the trigger threshold before the trigger is armed so that an IF power trigger event can be generated.

return
minimum_gap: No help available

get_slope() → SignalSlope

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:SLOPe
value: enums.SignalSlope = driver.trigger.uwbMeas.multiEval.get_slope()
```

No command help available

return
slope: No help available

get_source() → str

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:SOURce
value: str = driver.trigger.uwbMeas.multiEval.get_source()
```

Selects the source of the trigger events. Some values are always available. They are listed below. Depending on the installed options, additional values are available. You can query a list of all supported values via TRIGger:... :CATalog:SOURce?.

return
source: - 'Free Run': Free run without synchronization - 'IF Power': Power trigger (received RF power)

get_threshold() → float

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:THReshold
value: float = driver.trigger.uwbMeas.multiEval.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return
threshold: No help available

get_timeout() → float

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:TOUT
value: float or bool = driver.trigger.uwbMeas.multiEval.get_timeout()
```

Selects the maximum time that the measurement waits for a trigger event before it stops in remote control mode or indicates a trigger timeout in manual operation mode. This setting has no influence on Free Run measurements.

return
timeout: (float or boolean) No help available

set_delay(delay: float) → None

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:DElay
driver.trigger.uwbMeas.multiEval.set_delay(delay = 1.0)
```

Defines a time delaying the start of the measurement relative to the trigger event. This setting has no influence on free run measurements.

param delay
No help available

set_mgap(*minimum_gap: float*) → None

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:MGAP
driver.trigger.uwbMeas.multiEval.set_mgap(minimum_gap = 1.0)
```

Sets a minimum time during which the IF signal must be below the trigger threshold before the trigger is armed so that an IF power trigger event can be generated.

param minimum_gap
No help available

set_slope(*slope: SignalSlope*) → None

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:SLOPe
driver.trigger.uwbMeas.multiEval.set_slope(slope = enums.SignalSlope.FEDGE)
```

No command help available

param slope
No help available

set_source(*source: str*) → None

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:SOURce
driver.trigger.uwbMeas.multiEval.set_source(source = 'abc')
```

Selects the source of the trigger events. Some values are always available. They are listed below. Depending on the installed options, additional values are available. You can query a list of all supported values via TRIGger:... :CATalog:SOURce?.

param source

- ‘Free Run’: Free run without synchronization
- ‘IF Power’: Power trigger (received RF power)

set_threshold(*threshold: float*) → None

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:THReshold
driver.trigger.uwbMeas.multiEval.set_threshold(threshold = 1.0)
```

Defines the trigger threshold for power trigger sources.

param threshold
No help available

set_timeout(*timeout: float*) → None

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEvaluation:TOUT
driver.trigger.uwbMeas.multiEval.set_timeout(timeout = 1.0)
```

Selects the maximum time that the measurement waits for a trigger event before it stops in remote control mode or indicates a trigger timeout in manual operation mode. This setting has no influence on Free Run measurements.

param timeout

(float or boolean) No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.trigger.uwbMeas.multiEval.clone()
```

Subgroups

6.4.1.1.1 Catalog

SCPI Command :

```
TRIGger:UWB:MEASurement<Instance>:MEValuation:CATalog:SOURce
```

class CatalogCls

Catalog commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_source() → List[str]

```
# SCPI: TRIGger:UWB:MEASurement<Instance>:MEValuation:CATalog:SOURce
value: List[str] = driver.trigger.uwbMeas.multiEval.catalog.get_source()
```

Lists all trigger source values that can be set using method RsCMPX_UwbMeas.Trigger.UwbMeas.MultiEval.source.

return

trigger_sources: Comma-separated list of all supported values, one string per value.

6.5 UwbMeas

class UwbMeasCls

UwbMeas commands group definition. 494 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.clone()
```


Subgroups

6.5.1 MultiEval

SCPI Commands :

```
INITiate:UWB:MEASurement<Instance>:MEvaluation
STOP:UWB:MEASurement<Instance>:MEvaluation
ABORt:UWB:MEASurement<Instance>:MEvaluation
```

class MultiEvalCls

MultiEval commands group definition. 494 total commands, 9 Subgroups, 3 group commands

abort(opc_timeout_ms: int = -1) → None

```
# SCPI: ABORt:UWB:MEASurement<Instance>:MEvaluation
driver.uwbMeas.multiEval.abort()
```

INTRO_CMD_HELP: Starts, stops **or** aborts the measurement:

- INITiate... starts **or** restarts the measurement. The measurement enters the RUN state.
- STOP... halts the measurement immediately. The measurement enters the RDY state. Measurement results are kept. The resources remain allocated to the measurement.
- ABORt... halts the measurement immediately. The measurement enters the OFF state. All measurement values are **set** to NAV. Allocated resources are released.

Use FETCh...STATe? to query the current measurement state.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

initiate(opc_timeout_ms: int = -1) → None

```
# SCPI: INITiate:UWB:MEASurement<Instance>:MEvaluation
driver.uwbMeas.multiEval.initiate()
```

INTRO_CMD_HELP: Starts, stops **or** aborts the measurement:

- INITiate... starts **or** restarts the measurement. The measurement enters the RUN state.
- STOP... halts the measurement immediately. The measurement enters the RDY state. Measurement results are kept. The resources remain allocated to the measurement.
- ABORt... halts the measurement immediately. The measurement enters the OFF state. All measurement values are **set** to NAV. Allocated resources are released.

Use FETCh...STATe? to query the current measurement state.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

stop(opc_timeout_ms: int = -1) → None

```
# SCPI: STOP:UWB:MEASurement<Instance>:MEvaluation
driver.uwbMeas.multiEval.stop()
```

INTRO_CMD_HELP: Starts, stops **or** aborts the measurement:

- INITiate... starts **or** restarts the measurement. The measurement enters the RUN state.
- STOP... halts the measurement immediately. The measurement enters the RDY state. Measurement results are kept. The resources remain allocated to the measurement.
- ABORT... halts the measurement immediately. The measurement enters the OFF state. All measurement values are **set** to NAV. Allocated resources are released.

Use FETCh...STATe? to query the current measurement state.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.clone()
```

Subgroups

6.5.1.1 Ddecoding

class DdecodingCls

Ddecoding commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.ddecoding.clone()
```

Subgroups

6.5.1.1.1 Clength

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CLENGTH<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CLENGTH<PPDU>

```

class ClengthCls

Clength commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliabilty: int: 'Reliability indicator'
- Content_Length: int: No parameter help available

fetch(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CLENGTH<PPDU>
value: ResultData = driver.uwbMeas.multiEval.ddecoding.clength.fetch(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the decoded data length.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CLENGTH<PPDU>
value: ResultData = driver.uwbMeas.multiEval.ddecoding.clength.read(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the decoded data length.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.1.2 Content

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CONTENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CONTENT<PPDU>
```

class ContentCls

Content commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Content: List[str]: Comma-separated list of hexadecimal values. The number of values can be queried via [CMDLINKRESOLVED UwbMeas.MultiEval.Ddecoding.Clength#Read CMDLINKRESOLVED].

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CONTENT<PPDU>
value: ResultData = driver.uwbMeas.multiEval.ddecoding.content.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the binary data content as a list of hexadecimal values.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CONTENT<PPDU>
value: ResultData = driver.uwbMeas.multiEval.ddecoding.content.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the binary data content as a list of hexadecimal values.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.1.3 RsParity

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:RSParity<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:RSParity<PPDU>
```

class RsParityCls

RsParity commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Solomon_Parity: enums.Result: Indicates the passed or failed check verdict. The parity check is invalid, if no Reed-Solomon encoding is detected.

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:RSParity<PPDU>
value: ResultData = driver.uwbMeas.multiEval.ddecoding.rsParity.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the result of the parity check of the Reed-Solomon encoding.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:RSParity<PPDU>
value: ResultData = driver.uwbMeas.multiEval.ddecoding.rsParity.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the result of the parity check of the Reed-Solomon encoding.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.2 Modulation**class ModulationCls**

Modulation commands group definition. 267 total commands, 27 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.clone()
```

Subgroups

6.5.1.2.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:AVERage<PPDU>  
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Freq_Offset_Hz: float: No parameter help available
- Freq_Offset: float: No parameter help available
- Chip_Clock_Error: float: No parameter help available
- Pulse_Nsme: float: No parameter help available
- Sym_Mod_Accuracy: float: No parameter help available
- Side_Lobe_Peak: float: No parameter help available
- Pulse_Ml_Width: float: No parameter help available
- Sym_Time_Jitter: float: No parameter help available
- Sym_Phase_Jitter: float: No parameter help available
- Chip_Time_Jitter: float: No parameter help available
- Chip_Phase_Jitter: float: No parameter help available
- Symbol_Evm: float: No parameter help available
- Chip_Evm: float: No parameter help available
- Rmarker: float: RMARKER time
- Shr_Nrmse: float: NRMSE for SHR
- Phr_Nrmse: float: NRMSE for PHR
- Psdu_Nrmse: float: NRMSE for PSDU
- Sts_Nrmse: float: NRMSE for STS
- Sync_Pulse_Loc_Pol: enums.Result: No parameter help available
- Sfd_Pulse_Loc_Pol: enums.Result: No parameter help available
- Sts_Pulse_Loc_Pol: enums.Result: No parameter help available
- Iq_Offset: float: I/Q offset

fetch(*ppdu=PPDU.Nr1*) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:AVERage<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Return the current, average, extreme and standard deviation single value modulation results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:AVERage<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Return the current, average, extreme and standard deviation single value modulation results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.2.2 CcError

class CcErrorCls

CcError commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.ccError.clone()
```

Subgroups

6.5.1.2.2.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(*ppdu*=*Ppdu.Nr1*) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:CCERor:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.ccError.
↳average.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(*ppdu*=*Ppdu.Nr1*) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:CCERor:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ccError.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(*ppdu*=*Ppdu.Nr1*) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CCERor:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ccError.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.2.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.ccError.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.ccError.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:CCERor:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CCERor:CURRent<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEValuation:MODulation:CCERor:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:CCERor:CURRent<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.ccError.
↳current.calculate(ppdu = repcap.Ppdu.Default)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

error: No help available

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:CCERor:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ccError.current.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

error: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ccError.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.ccError.current.clone()
```

6.5.1.2.2.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:CCERor:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.ccError.
↳extreme.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ccError.extreme.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ccError.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.2.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:SDEviation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCERor:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:CCERor:SDEviation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.ccError.
↳standardDev.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCEror:SDEViation
↪<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ccError.standardDev.
↪fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCEror:SDEViation
↪<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ccError.standardDev.
↪read(ppdu = repcap.Ppdu.Nr1)
```

Returns the positive or negative chip clock error.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.3 Cevm

class CevmCls

Cevm commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.cevm.clone()
```

Subgroups

6.5.1.2.3.1 Average

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:AVERage<PPDU>

```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cevm.average.fetch(ppdu =↳
↳repcap.Ppdu.Nr1)

```

Returns the chip EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

evm: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cevm.average.read(ppdu =↳
↳repcap.Ppdu.Nr1)

```

Returns the chip EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

evm: No help available

6.5.1.2.3.2 Current<Ppdu>

RepCap Settings

```

# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.cevm.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.cevm.current.repcap_ppdu_set(repcap.Ppdu.Nr1)

```

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:CEVM:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CEVM:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:CEVM:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cevm.current.fetch(ppdu =
↳repcap.Ppdu.Default)

```

Returns the chip EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

evm: No help available

read(ppdu=Ppdu.Default) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CEVM:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cevm.current.read(ppdu =
↳repcap.Ppdu.Default)

```

Returns the chip EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

evm: No help available

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.cevm.current.clone()

```

6.5.1.2.3.3 Extreme

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:EXTRemE<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:EXTRemE<PPDU>

```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:EXTRemE
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cevm.extreme.fetch(ppdu =↳
↳repcap.Ppdu.Nr1)

```

Returns the chip EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

evm: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:EXTRemE<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cevm.extreme.read(ppdu =↳
↳repcap.Ppdu.Nr1)

```

Returns the chip EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

evm: No help available

6.5.1.2.3.4 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:SDEVIation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:SDEVIation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:SDEviation
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cevm.standardDev.fetch(ppdu,
↳ repcap.Ppdu.Nr1)
```

Returns the chip EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

evm: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM:SDEviation
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cevm.standardDev.read(ppdu,
↳ repcap.Ppdu.Nr1)
```

Returns the chip EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

evm: No help available

6.5.1.2.4 CpJitter

class CpJitterCls

CpJitter commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.cpJitter.clone()
```

Subgroups

6.5.1.2.4.1 Average

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:AVERage<PPDU>
```


class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:CPJitter:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cpJitter.average.fetch(ppdu,
↳repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over all chips of the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CPJitter:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cpJitter.average.read(ppdu =,
↳repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over all chips of the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.4.2 Current<Ppdu>**RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.cpJitter.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.cpJitter.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:CPJitter:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CPJitter:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cpJitter.current.fetch(ppdu,
↳repcap.Ppdu.Default)
```

Returns the RMS phase jitter value averaged over all chips of the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cpJitter.current.read(ppdu,
↳repcap.Ppdu.Default)
```

Returns the RMS phase jitter value averaged over all chips of the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.cpJitter.current.clone()
```

6.5.1.2.4.3 Extreme

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cpJitter.extreme.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over all chips of the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cpJitter.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over all chips of the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.4.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJitter:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:CPJitter:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cpJitter.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over all chips of the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:CPJitter:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.cpJitter.standardDev.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over all chips of the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.5 CtJitter

class CtJitterCls

CtJitter commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.ctJitter.clone()
```

Subgroups

6.5.1.2.5.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJittter:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJittter:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJittter:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ctJitter.average.fetch(ppdu_
↳= repcap.Ppdu.Nr1)
```

Returns the chip time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CTJitter:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ctJitter.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the chip time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.5.2 Current<Ppdu>**RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.ctJitter.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.ctJitter.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:CTJitter:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CTJitter:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:CTJitter:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ctJitter.current.fetch(ppdu,
↳= repcap.Ppdu.Default)
```

Returns the chip time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

jitter: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJitter:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ctJitter.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the chip time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.ctJitter.current.clone()
```

6.5.1.2.5.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJitter:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJitter:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJitter:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ctJitter.extreme.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the chip time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CTJitter:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ctJitter.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the chip time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.5.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:CTJitter:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:CTJitter:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:CTJitter:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ctJitter.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the chip time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:CTJitter:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.ctJitter.standardDev.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the chip time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

```
return
    jitter: No help available
```

6.5.1.2.6 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Freq_Offset_Hz: float: No parameter help available
- Freq_Offset: float: No parameter help available
- Chip_Clock_Error: float: No parameter help available
- Pulse_Nsme: float: No parameter help available
- Sym_Mod_Accuracy: float: No parameter help available
- Side_Lobe_Peak: float: No parameter help available
- Pulse_Ml_Width: float: No parameter help available
- Sym_Time_Jitter: float: No parameter help available
- Sym_Phase_Jitter: float: No parameter help available
- Chip_Time_Jitter: float: No parameter help available
- Chip_Phase_Jitter: float: No parameter help available
- Symbol_Evm: float: No parameter help available
- Chip_Evm: float: No parameter help available
- Rmarker: float: RMARKER time
- Shr_Nrmse: float: NRMSE for SHR
- Phr_Nrmse: float: NRMSE for PHR
- Psdu_Nrmse: float: NRMSE for PSDU
- Sts_Nrmse: float: NRMSE for STS

- Sync_Pulse_Loc_Pol: enums.Result: No parameter help available
- Sfd_Pulse_Loc_Pol: enums.Result: No parameter help available
- Sts_Pulse_Loc_Pol: enums.Result: No parameter help available
- Iq_Offset: float: I/Q offset

fetch(ppdu=Ppdu.Default) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CURRENT<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.current.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Return the current, average, extreme and standard deviation single value modulation results.

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Default) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CURRENT<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Return the current, average, extreme and standard deviation single value modulation results.

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

structure: for return value, see the help for ResultData structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.current.clone()
```

6.5.1.2.7 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Freq_Offset_Hz: float: No parameter help available
- Freq_Offset: float: No parameter help available
- Chip_Clock_Error: float: No parameter help available
- Pulse_Nsme: float: No parameter help available
- Sym_Mod_Accuracy: float: No parameter help available
- Side_Lobe_Peak: float: No parameter help available
- Pulse_Ml_Width: float: No parameter help available
- Sym_Time_Jitter: float: No parameter help available
- Sym_Phase_Jitter: float: No parameter help available
- Chip_Time_Jitter: float: No parameter help available
- Chip_Phase_Jitter: float: No parameter help available
- Symbol_Evm: float: No parameter help available
- Chip_Evm: float: No parameter help available
- Rmarker: float: RMARKER time
- Shr_Nrmse: float: NRMSE for SHR
- Phr_Nrmse: float: NRMSE for PHR
- Psdu_Nrmse: float: NRMSE for PSDU
- Sts_Nrmse: float: NRMSE for STS
- Sync_Pulse_Loc_Pol: enums.Result: No parameter help available
- Sfd_Pulse_Loc_Pol: enums.Result: No parameter help available
- Sts_Pulse_Loc_Pol: enums.Result: No parameter help available
- Iq_Offset: float: I/Q offset

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:EXTreme<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.extreme.fetch(ppdu =
↪repcap.Ppdu.Nr1)
```

Return the current, average, extreme and standard deviation single value modulation results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:EXTreme<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.extreme.read(ppdu =
↳ repcap.Ppdu.Nr1)
```

Return the current, average, extreme and standard deviation single value modulation results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.2.8 Foffset

class FoffsetCls

Foffset commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.foffset.clone()
```

Subgroups

6.5.1.2.8.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:FOFFset:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.foffset.
↳ average.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:AVErage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.foffset.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:AVErage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.foffset.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

6.5.1.2.8.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.foffset.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.foffset.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:CURRENT<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(*ppdu*=*Ppdu.Default*) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEValuation:MODulation:FOFFset:CURRent<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.foffset.
↳ current.calculate(ppdu = repcap.Ppdu.Default)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

offset: No help available

fetch(*ppdu*=*Ppdu.Default*) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:FOFFset:CURRent
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.foffset.current.fetch(ppdu =
↳ repcap.Ppdu.Default)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

offset: No help available

read(*ppdu*=*Ppdu.Default*) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:FOFFset:CURRent
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.foffset.current.read(ppdu =
↳ repcap.Ppdu.Default)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

offset: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.foffset.current.clone()
```

6.5.1.2.8.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:FOFFset:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.foffset.
↳extreme.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.foffset.extreme.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.foffset.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

6.5.1.2.8.4 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:SDEViation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:FOFFset:SDEViation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.foffset.
↳standardDev.calculate(ppdu = repcap.Ppdu.Nr1)

```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:FOFFset:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.foffset.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFFset:SDEVIation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.foffset.standardDev.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in ppm.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

6.5.1.2.9 Fofh

class FofhCls

Fofh commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.fofh.clone()
```

Subgroups

6.5.1.2.9.1 Average

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.fofh.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in Hz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

read(*ppdu*=*Ppdu.Nr1*) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:FOFH:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.fofh.average.read(ppdu =
↳ repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in Hz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

6.5.1.2.9.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.fofh.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.fofh.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:FOFH:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:FOFH:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(*ppdu*=*Ppdu.Default*) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:FOFH:CURRent
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.fofh.current.fetch(ppdu =
↳ repcap.Ppdu.Default)
```

Returns the frequency error relative to the channel center frequency in Hz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

offset: No help available

read(*ppdu*=*Ppdu.Default*) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.fofh.current.read(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the frequency error relative to the channel center frequency in Hz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

offset: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.fofh.current.clone()
```

6.5.1.2.9.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:EXTreme
↵<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.fofh.extreme.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in Hz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:EXTreme<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.fofh.extreme.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the frequency error relative to the channel center frequency in Hz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

6.5.1.2.9.4 StandardDev

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:SDEViation
↪<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.fofh.standardDev.fetch(ppdu,
↪repcap.Ppdu.Nr1)

```

Returns the frequency error relative to the channel center frequency in Hz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:FOFH:SDEViation
↪<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.fofh.standardDev.read(ppdu,
↪repcap.Ppdu.Nr1)

```

Returns the frequency error relative to the channel center frequency in Hz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

offset: No help available

6.5.1.2.10 IqOffset

class IqOffsetCls

IqOffset commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.iqOffset.clone()
```

Subgroups

6.5.1.2.10.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:AVERage
↳<PPDU>
value: int = driver.uwbMeas.multiEval.modulation.iqOffset.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the I/Q offset.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

iq_offset: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:AVERage
↳<PPDU>
value: int = driver.uwbMeas.multiEval.modulation.iqOffset.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the I/Q offset.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

iq_offset: No help available

6.5.1.2.10.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.iqOffset.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.iqOffset.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:CURRENT
↳<PPDU>
value: int = driver.uwbMeas.multiEval.modulation.iqOffset.current.fetch(ppdu =,
↳repcap.Ppdu.Default)
```

Returns the I/Q offset.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

iq_offset: No help available

read(ppdu=Ppdu.Default) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:CURRENT
↳<PPDU>
value: int = driver.uwbMeas.multiEval.modulation.iqOffset.current.read(ppdu =,
↳repcap.Ppdu.Default)
```

Returns the I/Q offset.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

iq_offset: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.iqOffset.current.clone()
```

6.5.1.2.10.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:EXTreme
↳<PPDU>
value: int = driver.uwbMeas.multiEval.modulation.iqOffset.extreme.fetch(ppdu =,
↳repcap.Ppdu.Nr1)
```

Returns the I/Q offset.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

iq_offset: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:EXTreme
↳<PPDU>
value: int = driver.uwbMeas.multiEval.modulation.iqOffset.extreme.read(ppdu =,
↳repcap.Ppdu.Nr1)
```

Returns the I/Q offset.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

iq_offset: No help available

6.5.1.2.10.4 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:IQOffset:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```

# SCPI: FETCH:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:IQOffset:SDEViation<PPDU>
value: int = driver.uwbMeas.multiEval.modulation.iqOffset.standardDev.
↳ fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the I/Q offset.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

iq_offset: No help available

read(ppdu=Ppdu.Nr1) → int

```

# SCPI: READ:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:IQOffset:SDEViation<PPDU>
value: int = driver.uwbMeas.multiEval.modulation.iqOffset.standardDev.read(ppdu,
↳ repcap.Ppdu.Nr1)

```

Returns the I/Q offset.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

iq_offset: No help available

6.5.1.2.11 Nmse

class NmseCls

Nmse commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.nmse.clone()
```

Subgroups

6.5.1.2.11.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.nmse.average.fetch(ppdu =↳
↳repcap.Ppdu.Nr1)
```

Returns the pulse NMSE (normalized mean square error between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

nmse: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.nmse.average.read(ppdu =↳
↳repcap.Ppdu.Nr1)
```

Returns the pulse NMSE (normalized mean square error between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

nmse: No help available

6.5.1.2.11.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.nmse.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.nmse.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.nmse.current.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Returns the pulse NMSE (normalized mean square error between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

nmse: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:CURRENT<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.nmse.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the pulse NMSE (normalized mean square error between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

nmse: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.nmse.current.clone()
```

6.5.1.2.11.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.nmse.extreme.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the pulse NMSE (normalized mean square error between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

nmse: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:EXTreme<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.nmse.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the pulse NMSE (normalized mean square error between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

nmse: No help available

6.5.1.2.11.4 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:SDEViation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.nmse.standardDev.fetch(ppdu,
↳repcap.Ppdu.Nr1)

```

Returns the pulse NMSE (normalized mean square error between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

nmse: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:NMSE:SDEViation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.nmse.standardDev.read(ppdu =,
↳repcap.Ppdu.Nr1)

```

Returns the pulse NMSE (normalized mean square error between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

nmse: No help available

6.5.1.2.12 Otolerance

SCPI Commands :

```

READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:OTOLerance
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:OTOLerance

```

class OtoleranceCls

Otolerance commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:OTOLerance
value: float = driver.uwbMeas.multiEval.modulation.otolerance.fetch()
```

Returns the out of tolerance result for modulation measurements. It indicates the percentage of measurement intervals of the statistic count for modulation measurements exceeding the specified TX Modulation limits or Pulse Mask limits.

Suppressed linked return values: reliability

return
out_of_tolerance: No help available

read() → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:OTOLerance
value: float = driver.uwbMeas.multiEval.modulation.otolerance.read()
```

Returns the out of tolerance result for modulation measurements. It indicates the percentage of measurement intervals of the statistic count for modulation measurements exceeding the specified TX Modulation limits or Pulse Mask limits.

Suppressed linked return values: reliability

return
out_of_tolerance: No help available

6.5.1.2.13 Phr

class PhrCls

Phr commands group definition. 26 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.phr.clone()
```

Subgroups

6.5.1.2.13.1 Nrmse

class NrmseCls

Nrmse commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.phr.nrmse.clone()
```

Subgroups

6.5.1.2.13.2 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PHR:NRMSe:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.phr.nrmse.
↳average.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.nrmse.average.fetch(ppdu,
↳ repcap.Ppdu.Nr1)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:PHR:NRMSe:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.nrmse.average.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.13.3 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.phr.nrmse.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.phr.nrmse.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:PHR:NRMSe:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:PHR:NRMSe:CURRENT<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEValuation:MODulation:PHR:NRMSe:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:PHR:NRMSe:CURRENT<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.phr.nrmse.
↳current.calculate(ppdu = repcap.Ppdu.Default)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

error: No help available

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.nrmse.current.fetch(ppdu,
↳= repcap.Ppdu.Default)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.nrmse.current.read(ppdu,
↳= repcap.Ppdu.Default)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.phr.nrmse.current.clone()
```

6.5.1.2.13.4 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PHR:NRMSe:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.phr.nrmse.
↳extreme.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.nrmse.extreme.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.nrmse.extreme.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.13.5 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:SDEviation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:NRMSe:SDEviation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PHR:NRMSe:SDEviation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.phr.nrmse.
↳ standardDev.calculate(ppdu = repcap.Ppdu.Nr1)

```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PHR:NRMSe:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.nrmse.standardDev.
↳ fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PHR:NRMSe:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.nrmse.standardDev.
↳ read(ppdu = repcap.Ppdu.Nr1)

```

Returns the NRMSE for PHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

```
return
    error: No help available
```

6.5.1.2.13.6 Plevel

class PlevelCls

Plevel commands group definition. 14 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.phr.plevel.clone()
```

Subgroups

6.5.1.2.13.7 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:PHR:PLEvel:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.average.
↪fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:AVERage
↪<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.average.read(ppdu,
↪= repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.13.8 Current<Ppdu>**RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.phr.plevel.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.phr.plevel.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:CURRENT<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PHR:PLEvel:CURRENT<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.phr.plevel.
↳current.calculate(ppdu = repcap.Ppdu.Default)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

level: No help available

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PHR:PLEvel:CURRENT<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.current.
↳fetch(ppdu = repcap.Ppdu.Default)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

level: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:CURRent
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.current.read(ppdu,
↳ = repcap.Ppdu.Default)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

level: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.phr.plevel.current.clone()
```

6.5.1.2.13.9 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:MAXimum<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PHR:PLEvel:MAXimum<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.phr.plevel.
↳ maximum.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PHR:PLEvel:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.maximum.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:MAXimum
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.maximum.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.13.10 Minimum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:MINimum<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:MINimum<PPDU>
```

class MinimumCls

Minimum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PHR:PLEvel:MINimum<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.phr.plevel.
↳minimum.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PHR:PLEvel:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.minimum.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:MINimum
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.minimum.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.13.11 StandardDev**SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:SDEviation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PHR:PLEvel:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PHR:PLEvel:SDEviation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.phr.plevel.
↳standardDev.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:PHR:PLEvel:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.standardDev.
↪fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:PHR:PLEvel:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.phr.plevel.standardDev.
↪read(ppdu = repcap.Ppdu.Nr1)
```

Returns the PHR pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.14 Plevel

class PlevelCls

Plevel commands group definition. 15 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.plevel.clone()
```

Subgroups

6.5.1.2.14.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: enums.ResultStatus2: PHR pulse level
- Psdu_Plevel: enums.ResultStatus2: PSDU pulse level
- Sts_Plevel: enums.ResultStatus2: STS pulse level

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: float: PHR pulse level
- Psdu_Plevel: float: PSDU pulse level
- Sts_Plevel: float: STS pulse level

calculate(ppdu=Ppdu.Nr1) → CalculateStruct

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↪ :MEvaluation:MODulation:PLEvel:AVERage<PPDU>
value: CalculateStruct = driver.uwbMeas.multiEval.modulation.plevel.average.
↪ calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for CalculateStruct structure arguments.

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:AVERage
↪<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.average.
↪fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:AVERage
↪<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.average.
↪read(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.2.14.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.plevel.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.plevel.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:CURRENT<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: enums.ResultStatus2: PHR pulse level

- Psdu_Plevel: enums.ResultStatus2: PSDU pulse level
- Sts_Plevel: enums.ResultStatus2: STS pulse level

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: float: PHR pulse level
- Psdu_Plevel: float: PSDU pulse level
- Sts_Plevel: float: STS pulse level

calculate(ppdu=Ppdu.Default) → CalculateStruct

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PLEvel:CURRENT<PPDU>
value: CalculateStruct = driver.uwbMeas.multiEval.modulation.plevel.current.
↳calculate(ppdu = repcap.Ppdu.Default)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for CalculateStruct structure arguments.

fetch(ppdu=Ppdu.Default) → ResultData

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:CURRENT
↳<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.current.
↳fetch(ppdu = repcap.Ppdu.Default)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Default) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:CURRENT
↳<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.current.
↳read(ppdu = repcap.Ppdu.Default)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for ResultData structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.plevel.current.clone()
```

6.5.1.2.14.3 Maximum**SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MAXimum<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: enums.ResultStatus2: PHR pulse level
- Psdu_Plevel: enums.ResultStatus2: PSDU pulse level
- Sts_Plevel: enums.ResultStatus2: STS pulse level

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: float: PHR pulse level
- Psdu_Plevel: float: PSDU pulse level
- Sts_Plevel: float: STS pulse level

calculate(ppdu=Ppdu.Nr1) → CalculateStruct

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↪ :MEvaluation:MODulation:PLEvel:MAXimum<PPDU>
value: CalculateStruct = driver.uwbMeas.multiEval.modulation.plevel.maximum.
↪ calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for CalculateStruct structure arguments.

fetch(*ppdu*=*Ppdu.Nr1*) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MAXimum
↪<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.maximum.
↪fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(*ppdu*=*Ppdu.Nr1*) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MAXimum
↪<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.maximum.
↪read(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.2.14.4 Minimum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MINimum<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MINimum<PPDU>
```

class MinimumCls

Minimum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: enums.ResultStatus2: PHR pulse level
- Psdu_Plevel: enums.ResultStatus2: PSDU pulse level
- Sts_Plevel: enums.ResultStatus2: STS pulse level

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: float: PHR pulse level

- Psdu_Plevel: float: PSDU pulse level
- Sts_Plevel: float: STS pulse level

calculate(ppdu=Ppdu.Nr1) → CalculateStruct

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PLEvel:MINimum<PPDU>
value: CalculateStruct = driver.uwbMeas.multiEval.modulation.plevel.minimum.
↳ calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for CalculateStruct structure arguments.

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MINimum
↳ <PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.minimum.
↳ fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:MINimum
↳ <PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.minimum.
↳ read(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.2.14.5 StandardDev

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEVel:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEVel:SDEViation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEVel:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: enums.ResultStatus2: PHR pulse level
- Psdu_Plevel: enums.ResultStatus2: PSDU pulse level
- Sts_Plevel: enums.ResultStatus2: STS pulse level

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Phr_Plevel: float: PHR pulse level
- Psdu_Plevel: float: PSDU pulse level
- Sts_Plevel: float: STS pulse level

calculate(ppdu=Ppdu.Nr1) → CalculateStruct

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:PLEVel:SDEViation<PPDU>
value: CalculateStruct = driver.uwbMeas.multiEval.modulation.plevel.standardDev.
↪calculate(ppdu = repcap.Ppdu.Nr1)

```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for CalculateStruct structure arguments.

fetch(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEVel:SDEViation
↪<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.standardDev.
↪fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PLEvel:SDEviation
↪<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.plevel.standardDev.
↪read(ppdu = repcap.Ppdu.Nr1)
```

Returns pulse levels according to the FIRA specification, relative to the SHR pulse level.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.2.15 PmlWidth**class PmlWidthCls**

PmlWidth commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.pmlWidth.clone()
```

Subgroups**6.5.1.2.15.1 Average****SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:PMLWidth:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.pmlWidth.
↪average.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.pmlWidth.average.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.pmlWidth.average.read(ppdu =,
↳repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

6.5.1.2.15.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.pmlWidth.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.pmlWidth.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```


SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:CURRent<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PMLWidth:CURRent<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.pmlWidth.
↳current.calculate(ppdu = repcap.Ppdu.Default)

```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

width: No help available

fetch(ppdu=Ppdu.Default) → float

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.pmlWidth.current.fetch(ppdu,
↳= repcap.Ppdu.Default)

```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

width: No help available

read(ppdu=Ppdu.Default) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.pmlWidth.current.read(ppdu =
↳repcap.Ppdu.Default)

```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

width: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.pmlWidth.current.clone()
```

6.5.1.2.15.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PMLWidth:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.pmlWidth.
↳extreme.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PMLWidth:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.pmlWidth.extreme.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:PMLWidth:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.pmlWidth.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

6.5.1.2.15.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:PMLWidth:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:PMLWidth:SDEViation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEValuation:MODulation:PMLWidth:SDEViation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:PMLWidth:SDEViation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.pmlWidth.
↳standardDev.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:PMLWidth:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.pmlWidth.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:PMLWidth:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.pmlWidth.standardDev.
↪read(ppdu = repcap.Ppdu.Nr1)
```

Returns the pulse mainlobe width.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

width: No help available

6.5.1.2.16 Psdu

class PsduCls

Psdu commands group definition. 26 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.psdu.clone()
```

Subgroups

6.5.1.2.16.1 Nrmse

class NrmseCls

Nrmse commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.psdu.nrmse.clone()
```

Subgroups

6.5.1.2.16.2 Average

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:AVERage<PPDU>

```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PSDU:NRMSe:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.psdn.nrmse.
↳ average.calculate(ppdu = repcap.Ppdu.Nr1)

```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PSDU:NRMSe:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psdn.nrmse.average.
↳ fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:AVERage
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psdn.nrmse.average.read(ppdu,
↳ repcap.Ppdu.Nr1)

```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.16.3 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.psdU.nrmse.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.psdU.nrmse.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:CURRent<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:NRMSe:CURRent<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.psdU.nrmse.
↳current.calculate(ppdu = repcap.Ppdu.Default)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:NRMSe:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psdU.nrmse.current.
↳fetch(ppdu = repcap.Ppdu.Default)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psdU.nrmse.current.read(ppdu,
↳= repcap.Ppdu.Default)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.psdU.nrmse.current.clone()
```

6.5.1.2.16.4 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:NRMSe:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.psdU.nrmse.
↳extreme.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PSDU:NRMSe:EXTReme<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.nrmse.extreme.
↳ fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:EXTReme
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.nrmse.extreme.read(ppdu,
↳ repcap.Ppdu.Nr1)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.16.5 StandardDev

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:SDEviation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:NRMSe:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:PSDU:NRMSe:SDEviation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.psd.nrmse.
↳ standardDev.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:PSDU:NRMSe:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psdn.nrmse.standardDev.
↪fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:PSDU:NRMSe:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psdn.nrmse.standardDev.
↪read(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for PSDU, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.16.6 Plevel**class PlevelCls**

Plevel commands group definition. 14 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.psdn.plevel.clone()
```

Subgroups

6.5.1.2.16.7 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEvel:AVERage<PPDU>  
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEvel:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>  
↳:MEvaluation:MODulation:PSDU:PLEvel:AVERage<PPDU>  
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.average.  
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>  
↳:MEvaluation:MODulation:PSDU:PLEvel:AVERage<PPDU>  
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.average.  
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.16.8 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100  
rc = driver.uwbMeas.multiEval.modulation.psd.plevel.current.repcap_ppdu_get()  
driver.uwbMeas.multiEval.modulation.psd.plevel.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:PSDU:PLEVel:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:PSDU:PLEVel:CURRent<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEValuation:MODulation:PSDU:PLEVel:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:PSDU:PLEVel:CURRent<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.psd.plevel.
↳current.calculate(ppdu = repcap.Ppdu.Default)

```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

level: No help available

fetch(ppdu=Ppdu.Default) → float

```

# SCPI: FETCh:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:PSDU:PLEVel:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.current.
↳fetch(ppdu = repcap.Ppdu.Default)

```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

level: No help available

read(ppdu=Ppdu.Default) → float

```

# SCPI: READ:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:PSDU:PLEVel:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.current.
↳read(ppdu = repcap.Ppdu.Default)

```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

level: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.psd.plevel.current.clone()
```

6.5.1.2.16.9 Maximum**SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEvel:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEvel:MAXimum<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEvel:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEvel:MAXimum<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.psd.plevel.
↳maximum.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEvel:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.maximum.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEVel:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.maximum.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.16.10 Minimum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEVel:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEVel:MINimum<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEVel:MINimum<PPDU>
```

class MinimumCls

Minimum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEVel:MINimum<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.psd.plevel.
↳minimum.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEVel:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.minimum.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEVel:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.minimum.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.16.11 StandardDev**SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEVel:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEVel:SDEViation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:PSDU:PLEVel:SDEViation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEVel:SDEViation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.psd.plevel.
↳standardDev.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEVel:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:PSDU:PLEvel:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.psd.plevel.standardDev.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the PSDU pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.17 Rmarker

class RmarkerCls

Rmarker commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.rmarker.clone()
```

Subgroups

6.5.1.2.17.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:RMARker:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:RMARker:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.rmarker.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the RMARKER time.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

rmarker: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.rmarker.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the RMARKER time.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

rmarker: No help available

6.5.1.2.17.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.rmarker.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.rmarker.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float


```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.rmarker.current.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Returns the RMARKER time.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

rmarker: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.rmarker.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the RMARKER time.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

rmarker: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.rmarker.current.clone()
```

6.5.1.2.17.3 Extreme

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:EXTReMe<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:EXTReMe<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.rmarker.extreme.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the RMARKER time.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

rmarker: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.rmarker.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the RMARKER time.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

rmarker: No help available

6.5.1.2.17.4 StandardDev

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:RMARker:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:RMARker:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.rmarker.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the RMARKER time.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

rmarker: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:RMarker:SDEviation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.rmarker.standardDev.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the RMARKER time.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

rmarker: No help available

6.5.1.2.18 Sevm

class SevmCls

Sevm commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sevm.clone()
```

Subgroups

6.5.1.2.18.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sevm.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the symbol EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbol: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sevm.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the symbol EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbol: No help available

6.5.1.2.18.2 Current<Ppdu>**RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.sevm.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.sevm.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:CURREnt<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:CURREnt<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:CURREnt
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sevm.current.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Returns the symbol EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

symbol: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sevm.current.read(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the symbol EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

symbol: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sevm.current.clone()
```

6.5.1.2.18.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:EXTRemE<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:EXTRemE<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:EXTRemE
↵<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sevm.extreme.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the symbol EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbol: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:EXTRemE<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sevm.extreme.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the symbol EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbol: No help available

6.5.1.2.18.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:SDEViation<PPDU>  
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:SDEViation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:SDEViation  
↪<PPDU>  
value: float = driver.uwbMeas.multiEval.modulation.sevm.standardDev.fetch(ppdu,  
↪repcap.Ppdu.Nr1)
```

Returns the symbol EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbol: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SEVM:SDEViation  
↪<PPDU>  
value: float = driver.uwbMeas.multiEval.modulation.sevm.standardDev.read(ppdu =,  
↪repcap.Ppdu.Nr1)
```

Returns the symbol EVM RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbol: No help available

6.5.1.2.19 Sfd

class SfdCls

Sfd commands group definition. 4 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sfd.clone()
```

Subgroups

6.5.1.2.19.1 PlPolarity

class PlPolarityCls

PlPolarity commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sfd.plPolarity.clone()
```

Subgroups

6.5.1.2.19.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.sfd.plPolarity.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.sfd.plPolarity.current.repcap_ppdu_set(repcap.Ppdu.
↪Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SFD:PLPolarity:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SFD:PLPolarity:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → Result

```
# SCPI: FETCh:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:SFD:PLPolarity:CURRENT<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sfd.plPolarity.
↳ current.fetch(ppdu = repcap.Ppdu.Default)
```

Returns the result of the check for correct pulse location and polarity, for SFD.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

sfd_polarity: No help available

read(ppdu=Ppdu.Default) → Result

```
# SCPI: READ:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:SFD:PLPolarity:CURRENT<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sfd.plPolarity.
↳ current.read(ppdu = repcap.Ppdu.Default)
```

Returns the result of the check for correct pulse location and polarity, for SFD.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

sfd_polarity: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sfd.plPolarity.current.clone()
```

6.5.1.2.19.3 Extreme

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:SFD:PLPolarity:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SFD:PLPolarity:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → Result


```
# SCPI: FETCh:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SFD:PLPolarity:EXTreme<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sfd.plPolarity.
↳extreme.fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the result of the check for correct pulse location and polarity, for SFD.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

sfd_polarity: No help available

read(ppdu=Ppdu.Nr1) → Result

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SFD:PLPolarity:EXTreme<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sfd.plPolarity.
↳extreme.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the result of the check for correct pulse location and polarity, for SFD.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

sfd_polarity: No help available

6.5.1.2.20 Shr

class ShrCls

Shr commands group definition. 12 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.shr.clone()
```

Subgroups

6.5.1.2.20.1 Nrmse

class NrmseCls

Nrmse commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.shr.nrmse.clone()
```

Subgroups

6.5.1.2.20.2 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:SHR:NRMSe:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.shr.nrmse.
↳ average.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:AVERage
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.shr.nrmse.average.fetch(ppdu,
↳ repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SHR:NRMSe:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.shr.nrmse.average.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.20.3 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.shr.nrmse.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.shr.nrmse.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:SHR:NRMSe:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SHR:NRMSe:CURRENT<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEValuation:MODulation:SHR:NRMSe:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:SHR:NRMSe:CURRENT<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.shr.nrmse.
↳current.calculate(ppdu = repcap.Ppdu.Default)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

error: No help available

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.shr.nrmse.current.fetch(ppdu,
↳= repcap.Ppdu.Default)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.shr.nrmse.current.read(ppdu,
↳= repcap.Ppdu.Default)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.shr.nrmse.current.clone()
```

6.5.1.2.20.4 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SHR:NRMSe:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.shr.nrmse.
↳extreme.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.shr.nrmse.extreme.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.shr.nrmse.extreme.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.20.5 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:SDEviation<PPDU>  
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:SDEviation<PPDU>  
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SHR:NRMSe:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>  
→ :MEvaluation:MODulation:SHR:NRMSe:SDEviation<PPDU>  
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.shr.nrmse.  
→ standardDev.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>  
→ :MEvaluation:MODulation:SHR:NRMSe:SDEviation<PPDU>  
value: float = driver.uwbMeas.multiEval.modulation.shr.nrmse.standardDev.  
→ fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>  
→ :MEvaluation:MODulation:SHR:NRMSe:SDEviation<PPDU>  
value: float = driver.uwbMeas.multiEval.modulation.shr.nrmse.standardDev.  
→ read(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for SHR, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return
error: No help available

6.5.1.2.21 SIPeak

class SIPeakCls

SIPeak commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sIPeak.clone()
```

Subgroups

6.5.1.2.21.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:SLPeak:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sIPeak.average.
↪calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:AVERage
↪<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sIPeak.average.fetch(ppdu =
↪repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.slPeak.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

6.5.1.2.21.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.slPeak.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.slPeak.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FEtCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:CURRent<PPDU>
REA:D:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:CURRent<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SLPeak:CURRent<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.slPeak.current.
↳calculate(ppdu = repcap.Ppdu.Default)
```


Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

peak: No help available

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.slPeak.current.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

peak: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.slPeak.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

peak: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.slPeak.current.clone()
```

6.5.1.2.21.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
→:MEvaluation:MODulation:SLPeak:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.slPeak.extreme.
→calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:EXTreme
→<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.slPeak.extreme.fetch(ppdu =
→repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.slPeak.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

6.5.1.2.21.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:SDEViation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:SDEViation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SLPeak:SDEViation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.slPeak.
↳standardDev.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:SDEViation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.slPeak.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SLPeak:SDEviation
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.slPeak.standardDev.read(ppdu,
↳ = repcap.Ppdu.Nr1)
```

Returns the sidelobe peak (largest magnitude of sidelobe peaks of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak: No help available

6.5.1.2.22 SmAccuracy**class SmAccuracyCls**

SmAccuracy commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.smAccuracy.clone()
```

Subgroups**6.5.1.2.22.1 Average****SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:SMACcuracy:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.smAccuracy.
↳ average.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

accuracy: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SMACcuracy:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.smAccuracy.average.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

accuracy: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.smAccuracy.average.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

accuracy: No help available

6.5.1.2.22.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.smAccuracy.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.smAccuracy.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:SMACcuracy:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SMACcuracy:CURRent<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEValuation:MODulation:SMACcuracy:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEValuation:MODulation:SMACcuracy:CURRent<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.smAccuracy.
↳ current.calculate(ppdu = repcap.Ppdu.Default)

```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

accuracy: No help available

fetch(ppdu=Ppdu.Default) → float

```

# SCPI: FETCh:UWB:MEASurement<Instance>
↳ :MEValuation:MODulation:SMACcuracy:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.smAccuracy.current.
↳ fetch(ppdu = repcap.Ppdu.Default)

```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

accuracy: No help available

read(ppdu=Ppdu.Default) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SMACcuracy:CURRent
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.smAccuracy.current.read(ppdu,
↳ = repcap.Ppdu.Default)

```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

accuracy: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.smAccuracy.current.clone()
```

6.5.1.2.22.3 Extreme**SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SMACcuracy:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.smAccuracy.
↳extreme.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

accuracy: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SMACcuracy:EXTreme<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.smAccuracy.extreme.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return
accuracy: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:EXTreme
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.smAccuracy.extreme.read(ppdu,
↳ = repcap.Ppdu.Nr1)
```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
accuracy: No help available

6.5.1.2.22.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:SDEviation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:SMACcuracy:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:SMACcuracy:SDEviation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.smAccuracy.
↳ standardDev.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
accuracy: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:SMACcuracy:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.smAccuracy.standardDev.
↳ fetch(ppdu = repcap.Ppdu.Nr1)
```


Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

accuracy: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SMACcuracy:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.smAccuracy.standardDev.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the symbol modulation accuracy (magnitude of the normalized cross-correlation between the measured pulse and the reference pulse) .

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

accuracy: No help available

6.5.1.2.23 SpJitter

class SpJitterCls

SpJitter commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.spJitter.clone()
```

Subgroups

6.5.1.2.23.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SPJitter:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SPJitter:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:AVERage
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.spJitter.average.fetch(ppdu,
↳ repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:AVERage
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.spJitter.average.read(ppdu =,
↳ repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.23.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.spJitter.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.spJitter.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.spJitter.current.fetch(ppdu,
↳= repcap.Ppdu.Default)
```

Returns the RMS phase jitter value averaged over the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:CURRent
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.spJitter.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the RMS phase jitter value averaged over the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.spJitter.current.clone()
```

6.5.1.2.23.3 Extreme

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.spJitter.extreme.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.spJitter.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.23.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SPJitter:SDEviation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:SPJitter:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.spJitter.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SPJitter:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.spJitter.standardDev.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the RMS phase jitter value averaged over the detected preamble symbols.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.24 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SDEViation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Freq_Offset_Hz: float: No parameter help available
- Freq_Offset: float: No parameter help available
- Chip_Clock_Error: float: No parameter help available
- Pulse_Nsme: float: No parameter help available
- Sym_Mod_Accuracy: float: No parameter help available
- Side_Lobe_Peak: float: No parameter help available
- Pulse_ML_Width: float: No parameter help available
- Sym_Time_Jitter: float: No parameter help available
- Sym_Phase_Jitter: float: No parameter help available
- Chip_Time_Jitter: float: No parameter help available
- Chip_Phase_Jitter: float: No parameter help available
- Symbol_Evm: float: No parameter help available
- Chip_Evm: float: No parameter help available
- Rmarker: float: RMARKER time

- Shr_Nrmse: float: NRMSE for SHR
- Phr_Nrmse: float: NRMSE for PHR
- Psdu_Nrmse: float: NRMSE for PSDU
- Sts_Nrmse: float: NRMSE for STS
- Sync_Pulse_Loc_Pol: enums.Result: No parameter help available
- Sfd_Pulse_Loc_Pol: enums.Result: No parameter help available
- Sts_Pulse_Loc_Pol: enums.Result: No parameter help available
- Iq_Offset: float: I/Q offset

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SDEviation<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.standardDev.fetch(ppdu,
↳ repcap.Ppdu.Nr1)
```

Return the current, average, extreme and standard deviation single value modulation results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SDEviation<PPDU>
value: ResultData = driver.uwbMeas.multiEval.modulation.standardDev.read(ppdu,
↳ repcap.Ppdu.Nr1)
```

Return the current, average, extreme and standard deviation single value modulation results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.2.25 StJitter

class StJitterCls

StJitter commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.stJitter.clone()
```

Subgroups

6.5.1.2.25.1 Average

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:STJitter:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:STJitter:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:STJitter:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.stJitter.average.fetch(ppdu,
↳repcap.Ppdu.Nr1)
```

Returns the symbol time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:STJitter:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.stJitter.average.read(ppdu,
↳repcap.Ppdu.Nr1)
```

Returns the symbol time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.25.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.stJitter.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.stJitter.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STJitter:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STJitter:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STJitter:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.stJitter.current.fetch(ppdu,
↳ repcap.Ppdu.Default)
```

Returns the symbol time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

jitter: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STJitter:CURRENT
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.stJitter.current.read(ppdu =
↳ repcap.Ppdu.Default)
```

Returns the symbol time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

jitter: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.stJitter.current.clone()
```

6.5.1.2.25.3 Extreme

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:STJitter:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:STJitter:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:STJitter:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.stJitter.extreme.fetch(ppdu,
↳repcap.Ppdu.Nr1)
```

Returns the symbol time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:STJitter:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.stJitter.extreme.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the symbol time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.25.4 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STJitter:SDEViation<PPDU>  
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STJitter:SDEViation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>  
↪:MEvaluation:MODulation:STJitter:SDEViation<PPDU>  
value: float = driver.uwbMeas.multiEval.modulation.stJitter.standardDev.  
↪fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the symbol time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>  
↪:MEvaluation:MODulation:STJitter:SDEViation<PPDU>  
value: float = driver.uwbMeas.multiEval.modulation.stJitter.standardDev.  
↪read(ppdu = repcap.Ppdu.Nr1)
```

Returns the symbol time jitter RMS value.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

jitter: No help available

6.5.1.2.26 Sts

class StsCls

Sts commands group definition. 30 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sts.clone()
```

Subgroups

6.5.1.2.26.1 Nrmse

class NrmseCls

Nrmse commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sts.nrmse.clone()
```

Subgroups

6.5.1.2.26.2 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:AVERage<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:STS:NRMSe:AVERage<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sts.nrmse.
↪average.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:NRMSe:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.nrmse.average.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:NRMSe:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.nrmse.average.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.26.3 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.sts.nrmse.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.sts.nrmse.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:NRMSe:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:NRMSe:CURRent<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:NRMSe:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:STS:NRMSe:CURRent<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sts.nrmse.
↳ current.calculate(ppdu = repcap.Ppdu.Default)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:CURRent
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.nrmse.current.fetch(ppdu,
↳ = repcap.Ppdu.Default)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:CURRent
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.nrmse.current.read(ppdu,
↳ = repcap.Ppdu.Default)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

error: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sts.nrmse.current.clone()
```

6.5.1.2.26.4 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:EXTreme<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:STS:NRMSe:EXTreme<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sts.nrmse.
↳extreme.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.nrmse.extreme.fetch(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:EXTreme
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.nrmse.extreme.read(ppdu,
↳= repcap.Ppdu.Nr1)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.26.5 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:SDEViation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:NRMSe:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:STS:NRMSe:SDEViation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sts.nrmse.
↳standardDev.calculate(ppdu = repcap.Ppdu.Nr1)

```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:STS:NRMSe:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.nrmse.standardDev.
↳fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:STS:NRMSe:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.nrmse.standardDev.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the NRMSE for STS, according to FIRA specification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

error: No help available

6.5.1.2.26.6 Plevel

class PlevelCls

Plevel commands group definition. 14 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sts.plevel.clone()
```

Subgroups

6.5.1.2.26.7 Average

SCPI Commands :

```
FEtCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEVel:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEVel:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FEtCh:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:STS:PLEVel:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.average.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:PLEvel:AVERage
↳<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.average.read(ppdu,
↳ = repcap.Ppdu.Nr1)
```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.26.8 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.sts.plevel.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.sts.plevel.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:PLEvel:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:PLEvel:CURRENT<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEValuation:MODulation:STS:PLEvel:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

calculate(ppdu=Ppdu.Default) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEValuation:MODulation:STS:PLEvel:CURRENT<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sts.plevel.
↳current.calculate(ppdu = repcap.Ppdu.Default)
```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

level: No help available

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:STS:PLEVel:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.current.
↳ fetch(ppdu = repcap.Ppdu.Default)
```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

level: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEVel:CURRent
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.current.read(ppdu,
↳ = repcap.Ppdu.Default)
```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

level: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sts.plevel.current.clone()
```

6.5.1.2.26.9 Maximum

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEVel:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEVel:MAXimum<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEVel:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(*ppdu*=*Ppdu.Nr1*) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:STS:PLEvel:MAXimum<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sts.plevel.
↳ maximum.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

fetch(*ppdu*=*Ppdu.Nr1*) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:STS:PLEvel:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.maximum.
↳ fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(*ppdu*=*Ppdu.Nr1*) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEvel:MAXimum
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.maximum.read(ppdu,
↳ = repcap.Ppdu.Nr1)
```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.26.10 Minimum

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEvel:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEvel:MINimum<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEvel:MINimum<PPDU>

```

class MinimumCls

Minimum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:STS:PLEvel:MINimum<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sts.plevel.
↳ minimum.calculate(ppdu = repcap.Ppdu.Nr1)

```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>
↳ :MEvaluation:MODulation:STS:PLEvel:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.minimum.
↳ fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEvel:MINimum
↳ <PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.minimum.read(ppdu,
↳ repcap.Ppdu.Nr1)

```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return
level: No help available

6.5.1.2.26.11 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEvel:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEvel:SDEViation<PPDU>
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLEvel:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```

# SCPI: CALCulate:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:STS:PLEvel:SDEViation<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.modulation.sts.plevel.
↪standardDev.calculate(ppdu = repcap.Ppdu.Nr1)

```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
level: No help available

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:STS:PLEvel:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.standardDev.
↪fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
level: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:STS:PLEvel:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.modulation.sts.plevel.standardDev.
↪read(ppdu = repcap.Ppdu.Nr1)

```

Returns the STS pulse level according to the FIRA specification, relative to the SHR pulse level.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

level: No help available

6.5.1.2.26.12 PlPolarity**class PlPolarityCls**

PlPolarity commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sts.plPolarity.clone()
```

Subgroups**6.5.1.2.26.13 Current<Ppdu>****RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.sts.plPolarity.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.sts.plPolarity.current.repcap_ppdu_set(repcap.Ppdu.
↪Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLPolarity:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLPolarity:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → Result

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:STS:PLPolarity:CURRENT<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sts.plPolarity.
↪current.fetch(ppdu = repcap.Ppdu.Default)
```

Returns the result of the check for correct pulse location and polarity, for STS.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return
 sts_polarity: No help available

read(ppdu=Ppdu.Default) → Result

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:STS:PLPolarity:CURRENT<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sts.plPolarity.
↳current.read(ppdu = repcap.Ppdu.Default)
```

Returns the result of the check for correct pulse location and polarity, for STS.

Suppressed linked return values: reliability

param ppdu
 optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return
 sts_polarity: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sts.plPolarity.current.clone()
```

6.5.1.2.26.14 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLPolarity:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:STS:PLPolarity:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → Result

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:STS:PLPolarity:EXTreme<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sts.plPolarity.
↳extreme.fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the result of the check for correct pulse location and polarity, for STS.

Suppressed linked return values: reliability

param ppdu
 optional repeated capability selector. Default value: Nr1

return
 sts_polarity: No help available

`read(ppdu=Ppdu.Nr1) → Result`

```
# SCPI: READ:UWB:MEASurement<Instance>
↪:MEvaluation:MODulation:STS:PLPolarity:EXTreme<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sts.plPolarity.
↪extreme.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the result of the check for correct pulse location and polarity, for STS.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

sts_polarity: No help available

6.5.1.2.27 Sync

class SyncCls

Sync commands group definition. 4 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sync.clone()
```

Subgroups

6.5.1.2.27.1 PIPolarity

class PlPolarityCls

PIPolarity commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sync.plPolarity.clone()
```

Subgroups

6.5.1.2.27.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.modulation.sync.plPolarity.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.modulation.sync.plPolarity.current.repcap_ppdu_set(repcap.Ppdu.
↪Nr1)
```


SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEValuation:MODulation:SYNC:PLPolarity:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:MODulation:SYNC:PLPolarity:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

class FetchStruct

Response structure. Fields:

- Reliabilty: int: No parameter help available
- Sync_Pulse_Loc_Pol: enums.Result: No parameter help available

fetch(ppdu=Ppdu.Default) → FetchStruct

```

# SCPI: FETCh:UWB:MEASurement<Instance>
↪:MEValuation:MODulation:SYNC:PLPolarity:CURRent<PPDU>
value: FetchStruct = driver.uwbMeas.multiEval.modulation.sync.plPolarity.
↪current.fetch(ppdu = repcap.Ppdu.Default)

```

Returns the result of the check for correct pulse location and polarity, for SYNC.

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for FetchStruct structure arguments.

read(ppdu=Ppdu.Default) → Result

```

# SCPI: READ:UWB:MEASurement<Instance>
↪:MEValuation:MODulation:SYNC:PLPolarity:CURRent<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sync.plPolarity.
↪current.read(ppdu = repcap.Ppdu.Default)

```

Returns the result of the check for correct pulse location and polarity, for SYNC.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

sync_pulse_loc_pol: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.modulation.sync.plPolarity.current.clone()
```

6.5.1.2.27.3 Extreme

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:SYNC:PLPolarity:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:SYNC:PLPolarity:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class FetchStruct

Response structure. Fields:

- Reliabilty: int: No parameter help available
- Sync_Pulse_Loc_Pol: enums.Result: No parameter help available

fetch(ppdu=Ppdu.Nr1) → FetchStruct

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SYNC:PLPolarity:EXTreme<PPDU>
value: FetchStruct = driver.uwbMeas.multiEval.modulation.sync.plPolarity.
↳extreme.fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the result of the check for correct pulse location and polarity, for SYNC.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for FetchStruct structure arguments.

read(ppdu=Ppdu.Nr1) → Result

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:MODulation:SYNC:PLPolarity:EXTreme<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.modulation.sync.plPolarity.
↳extreme.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the result of the check for correct pulse location and polarity, for SYNC.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

sync_pulse_loc_pol: No help available

6.5.1.3 Pmask

class PmaskCls

Pmask commands group definition. 11 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.clone()
```

Subgroups

6.5.1.3.1 Lower

class LowerCls

Lower commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.lower.clone()
```

Subgroups

6.5.1.3.1.1 Area<Area>

RepCap Settings

```
# Range: Nr1 .. Nr3
rc = driver.uwbMeas.multiEval.pmask.lower.area.repcap_area_get()
driver.uwbMeas.multiEval.pmask.lower.area.repcap_area_set(repcap.Area.Nr1)
```

class AreaCls

Area commands group definition. 2 total commands, 1 Subgroups, 0 group commands Repeated Capability: Area, default value after init: Area.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.lower.area.clone()
```

Subgroups

6.5.1.3.1.2 Margin

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:PMASK:LOWer:AREA<nr>:MARGin<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:PMASK:LOWer:AREA<nr>:MARGin<PPDU>

```

class MarginCls

Margin commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Margin_Lower_X: float: X-position of the margin for the lower area no
- Margin_Lower_Y: float: Y-value of the margin for the lower area no

fetch(area=Area.Default, ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:PMASK:LOWer:AREA<nr>:MARGin
↳<PPDU>
value: ResultData = driver.uwbMeas.multiEval.pmask.lower.area.margin.fetch(area_
↳= repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)

```

Returns the margin values between the transmitted pulse trace and the pulse mask for the lower area <no>. A negative margin indicates that the trace is located below the limit line, i.e. the limit is exceeded. See also 'Pulse Mask square'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(area=Area.Default, ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:PMASK:LOWer:AREA<nr>:MARGin
↳<PPDU>
value: ResultData = driver.uwbMeas.multiEval.pmask.lower.area.margin.read(area_
↳= repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)

```

Returns the margin values between the transmitted pulse trace and the pulse mask for the lower area <no>. A negative margin indicates that the trace is located below the limit line, i.e. the limit is exceeded. See also 'Pulse Mask square'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.3.2 Margin**class MarginCls**

Margin commands group definition. 5 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.margin.clone()
```

Subgroups**6.5.1.3.2.1 Area<Area>****RepCap Settings**

```
# Range: Nr1 .. Nr3
rc = driver.uwbMeas.multiEval.pmask.margin.area.repcap_area_get()
driver.uwbMeas.multiEval.pmask.margin.area.repcap_area_set(repcap.Area.Nr1)
```

class AreaCls

Area commands group definition. 1 total commands, 1 Subgroups, 0 group commands Repeated Capability:
Area, default value after init: Area.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.margin.area.clone()
```

Subgroups**6.5.1.3.2.2 Current<Ppdu>****RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.pmask.margin.area.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.pmask.margin.area.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Command :

```
CALCulate:UWB:MEASurement<Instance>:MEvaluation:PMASk:MARGin:AREA<nr>:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Margin_Upper_Y: enums.ResultStatus2: Limit check result for upper area no.
- Margin_Lower_Y: enums.ResultStatus2: Limit check result for lower area no.

calculate(area=Area.Default, ppdu=Ppdu.Default) → CalculateStruct

```
# SCPI: CALCulate:UWB:MEASurement<Instance>:MEvaluation:PMASk:MARGin:AREA<nr>
↪:CURRent<PPDU>
value: CalculateStruct = driver.uwbMeas.multiEval.pmask.margin.area.current.
↪calculate(area = repcap.Area.Default, ppdu = repcap.Ppdu.Default)
```

Returns the limit check results for the pulse mask measurement, for the pulse mask area <no>. See also 'Pulse Mask square'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for CalculateStruct structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.margin.area.current.clone()
```

6.5.1.3.2.3 PrMonotonic**class PrMonotonicCls**

PrMonotonic commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.margin.prMonotonic.clone()
```

Subgroups

6.5.1.3.2.4 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.pmask.margin.prMonotonic.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.pmask.margin.prMonotonic.current.repcap_ppdu_set(repcap.Ppdu.
↳Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:PMASK:MARGIN:PRMonotonic:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:PMASK:MARGIN:PRMonotonic:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → Result

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:PMASK:MARGIN:PRMonotonic:CURRENT<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.pmask.margin.prMonotonic.current.
↳fetch(ppdu = repcap.Ppdu.Default)
```

Returns the result of the check for monotonic rise of the pulse.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

pr_monotonic: No help available

read(ppdu=Ppdu.Default) → Result

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:PMASK:MARGIN:PRMonotonic:CURRENT<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.pmask.margin.prMonotonic.current.
↳read(ppdu = repcap.Ppdu.Default)
```

Returns the result of the check for monotonic rise of the pulse.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

pr_monotonic: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.margin.prMonotonic.current.clone()
```

6.5.1.3.2.5 Extreme**SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:PMASK:MARGIN:PRMonotonic:EXTreme<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:PMASK:MARGIN:PRMonotonic:EXTreme<PPDU>
```

class ExtremeCls

Extreme commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → Result

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↳:MEvaluation:PMASK:MARGIN:PRMonotonic:EXTreme<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.pmask.margin.prMonotonic.extreme.
↳fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the result of the check for monotonic rise of the pulse.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

pr_monotonic: No help available

read(ppdu=Ppdu.Nr1) → Result

```
# SCPI: READ:UWB:MEASurement<Instance>
↳:MEvaluation:PMASK:MARGIN:PRMonotonic:EXTreme<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.pmask.margin.prMonotonic.extreme.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the result of the check for monotonic rise of the pulse.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

pr_monotonic: No help available

6.5.1.3.3 Otolerance

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:PMASK:OTOLerance
READ:UWB:MEASurement<Instance>:MEvaluation:PMASK:OTOLerance

```

class OtoleranceCls

Otolerance commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:PMASK:OTOLerance
value: float = driver.uwbMeas.multiEval.pmask.otolerance.fetch()

```

No command help available

Suppressed linked return values: reliability

```

return
    out_of_tolerance: No help available

```

read() → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:PMASK:OTOLerance
value: float = driver.uwbMeas.multiEval.pmask.otolerance.read()

```

No command help available

Suppressed linked return values: reliability

```

return
    out_of_tolerance: No help available

```

6.5.1.3.4 Upper

class UpperCls

Upper commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.upper.clone()

```

Subgroups

6.5.1.3.4.1 Area<Area>

RepCap Settings

```
# Range: Nr1 .. Nr3
rc = driver.uwbMeas.multiEval.pmask.upper.area.repcap_area_get()
driver.uwbMeas.multiEval.pmask.upper.area.repcap_area_set(repcap.Area.Nr1)
```

class AreaCls

Area commands group definition. 2 total commands, 1 Subgroups, 0 group commands Repeated Capability:
Area, default value after init: Area.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.pmask.upper.area.clone()
```

Subgroups

6.5.1.3.4.2 Margin

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:PMASk:UPPer:AREA<nr>:MARGin<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:PMASk:UPPer:AREA<nr>:MARGin<PPDU>
```

class MarginCls

Margin commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Margin_Lower_X: float: X-position of the margin for the upper area no
- Margin_Lower_Y: float: Y-value of the margin for the upper area no

fetch(area=Area.Default, ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:PMASk:UPPer:AREA<nr>:MARGin
↳<PPDU>
value: ResultData = driver.uwbMeas.multiEval.pmask.upper.area.margin.fetch(area_
↳= repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)
```

Returns the margin values between the transmitted pulse trace and the pulse mask for the upper area <no>. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. See also 'Pulse Mask square'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(area=Area.Default, ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:PMASK:UPPer:AREA<nr>:MARGIn
↳<PPDU>
value: ResultData = driver.uwbMeas.multiEval.pmask.upper.area.margin.read(area_
↳= repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)
```

Returns the margin values between the transmitted pulse trace and the pulse mask for the upper area <no>. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. See also 'Pulse Mask square'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.4 Power

class PowerCls

Power commands group definition. 90 total commands, 13 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.clone()
```

Subgroups

6.5.1.4.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Preamble_Power: float: No parameter help available
- Pre_Peak_Power: float: No parameter help available
- Data_Power: float: No parameter help available
- Data_Peak_Power: float: No parameter help available
- Max_Spec_Power: float: No parameter help available
- Max_Spec_50_Power: float: No parameter help available
- Ppdu_Power: float: Mean power of the PPDU.
- Ppdu_Peak_Power: float: Peak power of the PPDU.

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:AVERage<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.average.fetch(ppdu = repcap.
↪Ppdu.Nr1)
```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:AVERage<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.average.read(ppdu = repcap.
↪Ppdu.Nr1)
```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.4.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEValuation:POWer:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:POWer:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Preamble_Power: float: No parameter help available
- Pre_Peak_Power: float: No parameter help available
- Data_Power: float: No parameter help available
- Data_Peak_Power: float: No parameter help available
- Max_Spec_Power: float: No parameter help available
- Max_Spec_50_Power: float: No parameter help available
- Ppdu_Power: float: Mean power of the PPDU.
- Ppdu_Peak_Power: float: Peak power of the PPDU.

fetch(ppdu=Ppdu.Default) → ResultData

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:POWer:CURRent<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.current.fetch(ppdu = repcap.
↳Ppdu.Default)

```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Default) → ResultData

```

# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:POWer:CURRent<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.current.read(ppdu = repcap.
↳Ppdu.Default)

```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for ResultData structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.current.clone()
```

6.5.1.4.3 Dpower

class DpowerCls

Dpower commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.dpower.clone()
```

Subgroups

6.5.1.4.3.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.average.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.average.read(ppdu = ↵
↵Ppdu.Nr1)
```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.3.2 Current<Ppdu>**RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.dpower.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.dpower.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.current.fetch(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.current.read(ppdu = repcap.
↵Ppdu.Default)
```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.dpower.current.clone()
```

6.5.1.4.3.3 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.maximum.fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.maximum.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.3.4 Minimum

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:MINimum<PPDU>

```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.minimum.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)

```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.minimum.read(ppdu = repcap.
↵Ppdu.Nr1)

```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.3.5 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPOWer:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:POWer:DPOWer:SDEviation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.standardDev.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:POWer:DPOWer:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpower.standardDev.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the mean power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.4 DpPower

class DpPowerCls

DpPower commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.dpPower.clone()
```

Subgroups

6.5.1.4.4.1 Average

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:POWer:DPPower:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:POWer:DPPower:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpPower.average.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpPower.average.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.4.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.dpPower.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.dpPower.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:CURRENT<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpPower.current.fetch(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpPower.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.dpPower.current.clone()
```

6.5.1.4.4.3 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpPower.maximum.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return
power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpPower.maximum.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

6.5.1.4.4.4 Minimum

SCPI Commands :

```
FEtCh:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:MINimum<PPDU>
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FEtCh:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpPower.minimum.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.dpPower.minimum.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

6.5.1.4.4.5 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:SDEViation<PPDU>  
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:SDEViation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:SDEViation  
↪<PPDU>  
value: float = driver.uwbMeas.multiEval.power.dpPower.standardDev.fetch(ppdu =  
↪repcap.Ppdu.Nr1)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:DPPower:SDEViation  
↪<PPDU>  
value: float = driver.uwbMeas.multiEval.power.dpPower.standardDev.read(ppdu =  
↪repcap.Ppdu.Nr1)
```

Returns the peak power of the data part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

6.5.1.4.5 Maximum

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MAXimum<PPDU>

```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Preamble_Power: float: No parameter help available
- Pre_Peak_Power: float: No parameter help available
- Data_Power: float: No parameter help available
- Data_Peak_Power: float: No parameter help available
- Max_Spec_Power: float: No parameter help available
- Max_Spec_50_Power: float: No parameter help available
- Ppdu_Power: float: Mean power of the PPDU.
- Ppdu_Peak_Power: float: Peak power of the PPDU.

fetch(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MAXimum<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.maximum.fetch(ppdu = repcap.
↳Ppdu.Nr1)

```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MAXimum<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.maximum.read(ppdu = repcap.
↳Ppdu.Nr1)

```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.4.6 Minimum

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MINimum<PPDU>

```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Preamble_Power: float: No parameter help available
- Pre_Peak_Power: float: No parameter help available
- Data_Power: float: No parameter help available
- Data_Peak_Power: float: No parameter help available
- Max_Spec_Power: float: No parameter help available
- Max_Spec_50_Power: float: No parameter help available
- Ppdu_Power: float: Mean power of the PPDU.
- Ppdu_Peak_Power: float: Peak power of the PPDU.

fetch(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MINimum<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.minimum.fetch(ppdu = repcap.
↳Ppdu.Nr1)

```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MINimum<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.minimum.read(ppdu = repcap.
↳Ppdu.Nr1)

```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.4.7 MsfPower

class MsfPowerCls

MsfPower commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.msfPower.clone()
```

Subgroups

6.5.1.4.7.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfPower.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfPower.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.7.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.msfpower.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.msfpower.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfpower.current.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfpower.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.msfpower.current.clone()
```

6.5.1.4.7.3 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfpower.maximum.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfpower.maximum.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.7.4 Minimum

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:MINimum<PPDU>

```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfpower.minimum.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)

```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfpower.minimum.read(ppdu = ↵
↵repcap.Ppdu.Nr1)

```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.7.5 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:SDEVIation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSFPower:SDEVIation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:POWer:MSFPower:SDEViation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfpower.standardDev.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:POWer:MSFPower:SDEViation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.msfpower.standardDev.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum peak spectral power measured with 50-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.8 MsPower

class MsPowerCls

MsPower commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.msPower.clone()
```

Subgroups

6.5.1.4.8.1 Average

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:POWer:MSPower:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:POWer:MSPower:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.average.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.average.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.8.2 Current<Ppdu>**RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.msPower.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.msPower.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.current.fetch(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.current.read(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.msPower.current.clone()
```

6.5.1.4.8.3 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.maximum.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.maximum.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.8.4 Minimum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:MINimum<PPDU>
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.minimum.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.minimum.read(ppdu =
↳repcap.Ppdu.Nr1)
```


Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.8.5 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:SDEviation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:SDEviation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:SDEviation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.standardDev.fetch(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:MSPower:SDEviation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.msPower.standardDev.read(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the maximum mean spectral power measured with 1-MHz RBW.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.9 PpdPeak

class PpdPeakCls

PpdPeak commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.ppdPeak.clone()
```

Subgroups

6.5.1.4.9.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.average.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ppdu_peak_power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.average.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ppdu_peak_power: No help available

6.5.1.4.9.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.ppdPeak.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.ppdPeak.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:CURREnt<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:CURREnt<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:CURREnt<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.current.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

ppdu_peak_power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:CURREnt<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

ppdu_peak_power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.ppdPeak.current.clone()
```

6.5.1.4.9.3 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.maximum.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ppdu_peak_power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.maximum.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ppdu_peak_power: No help available

6.5.1.4.9.4 Minimum

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:MINimum<PPDU>

```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.minimum.fetch(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ppdu_peak_power: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.minimum.read(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ppdu_peak_power: No help available

6.5.1.4.9.5 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:SDEviation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.standardDev.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ppdu_peak_power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDPeak:SDEviation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdPeak.standardDev.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the peak power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ppdu_peak_power: No help available

6.5.1.4.10 Ppdu<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.ppd.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.ppd.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

class PpduCls

Ppdu commands group definition. 10 total commands, 5 Subgroups, 0 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.ppd.clone()
```

Subgroups

6.5.1.4.10.1 Average

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:AVERage<PPDU>

```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Default) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.average.fetch(ppdu = repcap.
↳Ppdu.Default)

```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Ppdu')

return

ppdu_power: No help available

read(ppdu=Ppdu.Default) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.average.read(ppdu = repcap.
↳Ppdu.Default)

```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Ppdu')

return

ppdu_power: No help available

6.5.1.4.10.2 Current

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.current.fetch(ppdu = repcap.
↳ Ppdu.Default)
```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Ppdu')

return

ppdu_power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.current.read(ppdu = repcap.
↳ Ppdu.Default)
```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Ppdu')

return

ppdu_power: No help available

6.5.1.4.10.3 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.maximum.fetch(ppdu = repcap.
↳ Ppdu.Default)
```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Ppdu')


```

    return
    ppdu_power: No help available
read(ppdu=Ppdu.Default) → float

```

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.maximum.read(ppdu = repcap.
↳Ppdu.Default)

```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

```

    param ppdu
        optional repeated capability selector. Default value: Nr1 (settable in the interface
        'Ppdu')

    return
    ppdu_power: No help available

```

6.5.1.4.10.4 Minimum

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:MINimum<PPDU>

```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

```

fetch(ppdu=Ppdu.Default) → float

```

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.minimum.fetch(ppdu = repcap.
↳Ppdu.Default)

```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

```

    param ppdu
        optional repeated capability selector. Default value: Nr1 (settable in the interface
        'Ppdu')

    return
    ppdu_power: No help available

```

```

read(ppdu=Ppdu.Default) → float

```

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.minimum.read(ppdu = repcap.
↳Ppdu.Default)

```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Ppdu')

return

ppdu_power: No help available

6.5.1.4.10.5 StandardDev

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:SDEViation<PPDU>
```

```
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:SDEViation<PPDU>
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.standardDev.fetch(ppdu =
↳repcap.Ppdu.Default)
```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Ppdu')

return

ppdu_power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPDU:SDEViation<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppdu.standardDev.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the mean power of the PPDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Ppdu')

return

ppdu_power: No help available

6.5.1.4.11 Ppower

class PpowerCls

Ppower commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.ppower.clone()
```

Subgroups

6.5.1.4.11.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.average.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.average.read(ppdu = ↵
↵Ppdu.Nr1)
```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.11.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.ppower.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.ppower.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPoWer:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPoWer:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPoWer:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.current.fetch(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPoWer:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.current.read(ppdu = ↵
↵Ppdu.Default)
```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.ppower.current.clone()
```

6.5.1.4.11.3 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.maximum.fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.maximum.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.11.4 Minimum

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:MINimum<PPDU>

```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.minimum.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)

```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.minimum.read(ppdu = ↵
↵Ppdu.Nr1)

```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.11.5 StandardDev

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:SDEviation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.standardDev.fetch(ppdu =↳
↳repcap.Ppdu.Nr1)
```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPOWer:SDEviation<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppower.standardDev.read(ppdu =↳
↳repcap.Ppdu.Nr1)
```

Returns the mean power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.12 PpPower

class PpPowerCls

PpPower commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.ppPower.clone()
```

Subgroups

6.5.1.4.12.1 Average

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.average.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:AVERage<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.average.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: No help available

6.5.1.4.12.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.power.ppPower.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.power.ppPower.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.current.fetch(ppdu = ↵
↵repcap.Ppdu.Default)
```


Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: No help available

read(ppdu=Ppdu.Default) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:CURRent<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.power.ppPower.current.clone()
```

6.5.1.4.12.3 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.maximum.fetch(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return
power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:MAXimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.maximum.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

6.5.1.4.12.4 Minimum

SCPI Commands :

```
FEtCh:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:MINimum<PPDU>
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FEtCh:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.minimum.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:PPPower:MINimum<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.minimum.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

6.5.1.4.12.5 StandardDev

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEValuation:POWer:PPPower:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:POWer:PPPower:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:POWer:PPPower:SDEViation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.standardDev.fetch(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:POWer:PPPower:SDEViation
↳<PPDU>
value: float = driver.uwbMeas.multiEval.power.ppPower.standardDev.read(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the peak power of the preamble part.

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
power: No help available

6.5.1.4.13 StandardDev

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:SDEViation<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:POWer:SDEViation<PPDU>

```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Preamble_Power: float: No parameter help available
- Pre_Peak_Power: float: No parameter help available
- Data_Power: float: No parameter help available
- Data_Peak_Power: float: No parameter help available
- Max_Spec_Power: float: No parameter help available
- Max_Spec_50_Power: float: No parameter help available
- Ppdu_Power: float: Mean power of the PPDU.
- Ppdu_Peak_Power: float: Peak power of the PPDU.

fetch(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:POWer:SDEViation<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.standardDev.fetch(ppdu =
↳repcap.Ppdu.Nr1)

```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:POWer:SDEViation<PPDU>
value: ResultData = driver.uwbMeas.multiEval.power.standardDev.read(ppdu =
↳repcap.Ppdu.Nr1)

```

Return the current, average, extreme and standard deviation single value power results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.5 Sinfo

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:SINfo<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo<PPDU>

```

class SinfoCls

Sinfo commands group definition. 36 total commands, 14 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Psdu_Bitrate: float: No parameter help available
- Phr_Crc: enums.Result: Verification of PHR checksum (SECDED)
- Analysed_Sync_Sym: int: No parameter help available
- Cs_Length: int: Length of the code sequence
- Psdu_Length: int: Length of the PSDU
- Delta_Length: int: No parameter help available
- Code_Index: int: No parameter help available
- Detected_Ppdus: int: No parameter help available
- Payload_Sts_Gap_A_0: int: No parameter help available
- Payload_Sts_Gap_A_1: int: No parameter help available
- Ranging_Bit: int: Decoded ranging bit of the PHR
- Reserved_Bit: int: Decoded reserved bit of the PHR
- Fcs_Check: enums.Result: Verification of MAC FCS checksum
- Sync_Sym_Phr: int: Number of symbols in the SYNC field, read from the PHR
- Sfd_Value: int: SFD value as defined in IEEE Std 802.15.4z-2020
- Sfd_Length: int: Length of the SFD sequence
- Phr_Bitrate: float: No parameter help available

fetch(ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:SINfo<PPDU>
value: ResultData = driver.uwbMeas.multiEval.sinfo.fetch(ppdu = repcap.Ppdu.Nr1)

```

Return the current single value signal information results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo<PPDU>
value: ResultData = driver.uwbMeas.multiEval.sinfo.read(ppdu = repcap.Ppdu.Nr1)
```

Return the current single value signal information results.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.sinfo.clone()
```

Subgroups

6.5.1.5.1 AsSymbols

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:ASSYmbols<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:ASSYmbols<PPDU>
```

class AsSymbolsCls

AsSymbols commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:ASSYmbols<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.asSymbols.fetch(ppdu = repcap.Ppdu.
↳Nr1)
```

Returns the number of symbols detected in the SYNC field of the SHR.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbols: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:ASSYmbols<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.asSymbols.read(ppdu = repcap.Ppdu.
↳Nr1)
```

Returns the number of symbols detected in the SYNC field of the SHR.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbols: No help available

6.5.1.5.2 Cindex**SCPI Commands :**

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:CINdex<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:CINdex<PPDU>

```

class CindexCls

Cindex commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:CINdex<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.cindex.fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the code index.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

index: No help available

read(ppdu=Ppdu.Nr1) → int

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:CINdex<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.cindex.read(ppdu = repcap.Ppdu.Nr1)

```

Returns the code index.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

index: No help available

6.5.1.5.3 CsLength**SCPI Commands :**

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:CSLength<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:CSLength<PPDU>

```

class CsLengthCls

CsLength commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:CSLength<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.csLength.fetch(ppdu = repcap.Ppdu.
↳Nr1)
```

Returns the length of the code sequence.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

cs_length: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:CSLength<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.csLength.read(ppdu = repcap.Ppdu.
↳Nr1)
```

Returns the length of the code sequence.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

cs_length: No help available

6.5.1.5.4 Dlength

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:DLEnGth<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:DLEnGth<PPDU>
```

class DlengthCls

Dlength commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:DLEnGth<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.dlength.fetch(ppdu = repcap.Ppdu.
↳Nr1)
```

Returns the oversampling factor (delta length) of the code sequence.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

length: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:DLEnGth<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.dlength.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the oversampling factor (delta length) of the code sequence.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

length: No help available

6.5.1.5.5 Dppdu

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:DPPDu
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:DPPDu
```

class DppduCls

Dppdu commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:DPPDu
value: int = driver.uwbMeas.multiEval.sinfo.dppdu.fetch()
```

Returns the number of detected PPDU's in the capture length.

Suppressed linked return values: reliability

return

detected_ppdu: No help available

read() → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:DPPDu
value: int = driver.uwbMeas.multiEval.sinfo.dppdu.read()
```

Returns the number of detected PPDU's in the capture length.

Suppressed linked return values: reliability

return

detected_ppdu: No help available

6.5.1.5.6 Drate

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:DRATe<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:DRATe<PPDU>

```

class DrateCls

Drate commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:DRATe<PPDU>
value: float = driver.uwbMeas.multiEval.sinfo.drate.fetch(ppdu = repcap.Ppdu.
↳Nr1)

```

Returns the data rate of the PHY payload.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

psdu_bitrate: No help available

read(ppdu=Ppdu.Nr1) → float

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:DRATe<PPDU>
value: float = driver.uwbMeas.multiEval.sinfo.drate.read(ppdu = repcap.Ppdu.Nr1)

```

Returns the data rate of the PHY payload.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

psdu_bitrate: No help available

6.5.1.5.7 FcsCheck

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:FCSCheck<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:FCSCheck<PPDU>

```

class FcsCheckCls

FcsCheck commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → Result

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:FCSCheck<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.sinfo.fcsCheck.fetch(ppdu =
↳repcap.Ppdu.Nr1)

```

Returns the result of the MAC FCS checksum verification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

fcs_check: No help available

read(ppdu=Ppdu.Nr1) → Result

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:FCSCheck<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.sinfo.fcsCheck.read(ppdu =
↳repcap.Ppdu.Nr1)
```

Returns the result of the MAC FCS checksum verification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

fcs_check: No help available

6.5.1.5.8 Phr

class PhrCls

Phr commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.sinfo.phr.clone()
```

Subgroups

6.5.1.5.8.1 AsSymbols

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:ASSymbols<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:ASSymbols<PPDU>
```

class AsSymbolsCls

AsSymbols commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:ASSymbols<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.phr.asSymbols.fetch(ppdu = repcap.
↳Ppdu.Nr1)
```

Returns the number of symbols in the SYNC field, read from the PHR.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbols: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:ASymbols<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.phr.asSymbols.read(ppdu = repcap.
↳Ppdu.Nr1)
```

Returns the number of symbols in the SYNC field, read from the PHR.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

symbols: No help available

6.5.1.5.8.2 Bitrate

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:BITRate<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:BITRate<PPDU>
```

class BitrateCls

Bitrate commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:BITRate<PPDU>
value: float = driver.uwbMeas.multiEval.sinfo.phr.bitrate.fetch(ppdu = repcap.
↳Ppdu.Nr1)
```

Returns the data rate of the PHR.

param ppdu

optional repeated capability selector. Default value: Nr1

return

phr_bitrate: No help available

read(ppdu=Ppdu.Nr1) → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:BITRate<PPDU>
value: float = driver.uwbMeas.multiEval.sinfo.phr.bitrate.read(ppdu = repcap.
↳Ppdu.Nr1)
```

Returns the data rate of the PHR.

param ppdu

optional repeated capability selector. Default value: Nr1

return

phr_bitrate: No help available

6.5.1.5.8.3 Crc**SCPI Commands :**

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:CRC<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:CRC<PPDU>

```

class CrcCls

Crc commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → Result

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:CRC<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.sinfo.phr.crc.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)

```

Returns the result of the PHR checksum (SECDDED) verification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

crc: No help available

read(ppdu=Ppdu.Nr1) → Result

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PHR:CRC<PPDU>
value: enums.Result = driver.uwbMeas.multiEval.sinfo.phr.crc.read(ppdu = repcap.
↵Ppdu.Nr1)

```

Returns the result of the PHR checksum (SECDDED) verification.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

crc: No help available

6.5.1.5.9 Psdu

class PsduCls

Psdu commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.sinfo.psdu.clone()
```

Subgroups

6.5.1.5.9.1 Crc

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSDU:CRC<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSDU:CRC<PPDU>
```

class CrcCls

Crc commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → bool

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSDU:CRC<PPDU>
value: bool = driver.uwbMeas.multiEval.sinfo.psdu.crc.fetch(ppdu = repcap.Ppdu.
↳Nr1)
```

No command help available

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

crc: No help available

read(ppdu=Ppdu.Nr1) → bool

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSDU:CRC<PPDU>
value: bool = driver.uwbMeas.multiEval.sinfo.psdu.crc.read(ppdu = repcap.Ppdu.
↳Nr1)
```

No command help available

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

crc: No help available

6.5.1.5.9.2 Length

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSDU:LENGth<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSDU:LENGth<PPDU>

```

class LengthCls

Length commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSDU:LENGth<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.psdu.length.fetch(ppdu = repcap.
↳Ppdu.Nr1)

```

Returns the length of the PSDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

length: No help available

read(ppdu=Ppdu.Nr1) → int

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSDU:LENGth<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.psdu.length.read(ppdu = repcap.Ppdu.
↳Nr1)

```

Returns the length of the PSDU.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

length: No help available

6.5.1.5.10 PstGap

SCPI Commands :

```

READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSTGap<PPDU>
FETCh:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSTGap<PPDU>

```

class PstGapCls

PstGap commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'

- Payload_Sts_Gap_A_0: int: No parameter help available
- Payload_Sts_Gap_A_1: int: No parameter help available

class ReadStruct

Response structure. Fields:

- Reliabilty: int: No parameter help available
- Payload_Sts_Gap_A_0: int: No parameter help available
- Payload_Sts_Gap_A_1: int: No parameter help available

fetch(ppdu=Ppdu.Nr1) → FetchStruct

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSTGap<PPDU>
value: FetchStruct = driver.uwbMeas.multiEval.sinfo.pstGap.fetch(ppdu = repcap.
↳Ppdu.Nr1)
```

Returns the decoded bits A0 and A1 of the PHR.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for FetchStruct structure arguments.

read(ppdu=Ppdu.Nr1) → ReadStruct

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:PSTGap<PPDU>
value: ReadStruct = driver.uwbMeas.multiEval.sinfo.pstGap.read(ppdu = repcap.
↳Ppdu.Nr1)
```

Returns the decoded bits A0 and A1 of the PHR.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ReadStruct structure arguments.

6.5.1.5.11 RaBit

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:RABit<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:RABit<PPDU>
```

class RaBitCls

RaBit commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:RABit<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.raBit.fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the decoded ranging bit of the PHR.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ranging_bit: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:RABit<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.raBit.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the decoded ranging bit of the PHR.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

ranging_bit: No help available

6.5.1.5.12 ReBit**SCPI Commands :**

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:REBit<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:REBit<PPDU>
```

class ReBitCls

ReBit commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:REBit<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.reBit.fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the decoded reserved bit of the PHR.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

reserved_bit: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:REBit<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.reBit.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the decoded reserved bit of the PHR.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return
reserved_bit: No help available

6.5.1.5.13 Sfd

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:SFD<PPDU>  
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:SFD<PPDU>
```

class SfdCls

Sfd commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:SFD<PPDU>  
value: int = driver.uwbMeas.multiEval.sinfo.sfd.fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the SFD value as defined in IEEE Std 802.15.4z-2020, table 15-7c (SFD #) .

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
sfd_value: No help available

read(ppdu=Ppdu.Nr1) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:SFD<PPDU>  
value: int = driver.uwbMeas.multiEval.sinfo.sfd.read(ppdu = repcap.Ppdu.Nr1)
```

Returns the SFD value as defined in IEEE Std 802.15.4z-2020, table 15-7c (SFD #) .

Suppressed linked return values: reliability

param ppdu
optional repeated capability selector. Default value: Nr1

return
sfd_value: No help available

6.5.1.5.14 SfdLength

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:SFDLength<PPDU>  
READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:SFDLength<PPDU>
```

class SfdLengthCls

SfdLength commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*ppdu*=*Ppdu.Nr1*) → int

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:SINfo:SFDLength<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.sfdLength.fetch(ppdu = repcap.Ppdu.
↳Nr1)
```

Returns the length of the SFD sequence.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

sfd_length: No help available

read(*ppdu*=*Ppdu.Nr1*) → int

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:SINfo:SFDLength<PPDU>
value: int = driver.uwbMeas.multiEval.sinfo.sfdLength.read(ppdu = repcap.Ppdu.
↳Nr1)
```

Returns the length of the SFD sequence.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

sfd_length: No help available

6.5.1.6 State

SCPI Command :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:STATe
```

class StateCls

State commands group definition. 2 total commands, 1 Subgroups, 1 group commands

fetch(*timeout*: float = None, *target_main_state*: TargetMainState = None, *target_sync_state*: TargetSyncState = None) → ResourceState

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:STATe
value: enums.ResourceState = driver.uwbMeas.multiEval.state.fetch(timeout = 1.0,
↳ target_main_state = enums.TargetMainState.OFF, target_sync_state = enums.
↳TargetSyncState.ADJusted)
```

Queries the main measurement state. Without query parameters, the state is returned immediately. With query parameters, the state is returned when the <TargetMainState> and the <TargetSyncState> are reached or when the <Timeout> expires.

param timeout

No help available

param target_main_state

Target MainState for the query Default is RUN.

param target_sync_state

Target SyncState for the query Default is ADJ.

return

meas_state: Current state or target state of ongoing state transition OFF: measurement
off RUN: measurement running RDY: measurement completed

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.state.clone()
```

Subgroups

6.5.1.6.1 All

SCPI Command :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:STATe:ALL
```

class AllCls

All commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState = None) → List[ResourceState]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:STATe:ALL
value: List[enums.ResourceState] = driver.uwbMeas.multiEval.state.all.
↪ fetch(timeout = 1.0, target_main_state = enums.TargetMainState.OFF, target_
↪ sync_state = enums.TargetSyncState.ADJusted)
```

Queries the main measurement state and the measurement substates. Without query parameters, the states are returned immediately. With query parameters, the states are returned when the <TargetMainState> and the <TargetSyncState> are reached or when the <Timeout> expires.

param timeout

No help available

param target_main_state

Target MainState for the query Default is RUN.

param target_sync_state

Target SyncState for the query Default is ADJ.

return

meas_state: No help available

6.5.1.7 StsSequence

class StsSequenceCls

StsSequence commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.stsSequence.clone()
```

Subgroups

6.5.1.7.1 Clength

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:STSSequence:CLEngth<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:STSSequence:CLEngth<PPDU>
```

class ClengthCls

Clength commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Content_Length: int: No parameter help available

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:STSSequence:CLEngth<PPDU>
value: ResultData = driver.uwbMeas.multiEval.stsSequence.clength.fetch(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the content length of the STS sequence received from the DUT.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:STSSequence:CLEngth<PPDU>
value: ResultData = driver.uwbMeas.multiEval.stsSequence.clength.read(ppdu = ↵
↵repcap.Ppdu.Nr1)
```

Returns the content length of the STS sequence received from the DUT.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.7.2 Content

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:STSSequence:CONTENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:STSSequence:CONTENT<PPDU>
```

class ContentCls

Content commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Content: List[str]: No parameter help available

fetch(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:STSSequence:CONTENT<PPDU>
value: ResultData = driver.uwbMeas.multiEval.stsSequence.content.fetch(ppdu =
↪repcap.Ppdu.Nr1)
```

Returns the content of the STS sequence received from the DUT.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:STSSequence:CONTENT<PPDU>
value: ResultData = driver.uwbMeas.multiEval.stsSequence.content.read(ppdu =
↪repcap.Ppdu.Nr1)
```

Returns the content of the STS sequence received from the DUT.

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.8 Trace

class TraceCls

Trace commands group definition. 58 total commands, 10 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.clone()
```

Subgroups

6.5.1.8.1 CpJitter

class CpJitterCls

CpJitter commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.cpJitter.clone()
```

Subgroups

6.5.1.8.1.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:AVERage
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.cpJitter.average.fetch()
```

Returns the y-values of the average chip phase jitter trace. See also 'Chip Jitter square'.

Suppressed linked return values: reliability

return

jitter: Comma-separated list of chip jitter values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.cpJitter.average.read()
```

Returns the y-values of the average chip phase jitter trace. See also ‘Chip Jitter square’.

Suppressed linked return values: reliability

return

jitter: Comma-separated list of chip jitter values.

6.5.1.8.1.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.cpJitter.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.cpJitter.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:CURREnt<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:CURREnt<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:CURREnt<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.cpJitter.current.fetch(ppdu,
↳ repcap.Ppdu.Default)
```

Returns the y-values of the current chip phase jitter trace. See also ‘Chip Jitter square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: Comma-separated list of chip jitter values.

read(ppdu=Ppdu.Default) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:CURREnt<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.cpJitter.current.read(ppdu,
↳ repcap.Ppdu.Default)
```

Returns the y-values of the current chip phase jitter trace. See also ‘Chip Jitter square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: Comma-separated list of chip jitter values.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.cpJitter.current.clone()
```

6.5.1.8.1.3 Xvalues**SCPI Commands :**

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:XVALues
```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.cpJitter.xvalues.fetch()
```

Returns the x-values of the chip phase jitter trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of phase values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CPJitter:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.cpJitter.xvalues.read()
```

Returns the x-values of the chip phase jitter trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of phase values.

6.5.1.8.2 CtJitter

class CtJitterCls

CtJitter commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.ctJitter.clone()
```

Subgroups

6.5.1.8.2.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:CTJitter:AVERage
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CTJitter:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:CTJitter:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.ctJitter.average.fetch()
```

Returns the y-values of the average chip time jitter trace. See also 'Chip Jitter square'.

Suppressed linked return values: reliability

return
jitter: Comma-separated list of chip jitter values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CTJitter:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.ctJitter.average.read()
```

Returns the y-values of the average chip time jitter trace. See also 'Chip Jitter square'.

Suppressed linked return values: reliability

return
jitter: Comma-separated list of chip jitter values.

6.5.1.8.2.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.ctJitter.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.ctJitter.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEValuation:TRACe:CTJitter:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEValuation:TRACe:CTJitter:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEValuation:TRACe:CTJitter:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.ctJitter.current.fetch(ppdu,
↳ repcap.Ppdu.Default)
```

Returns the y-values of the current chip time jitter trace. See also ‘Chip Jitter square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: Comma-separated list of chip jitter values.

read(ppdu=Ppdu.Default) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEValuation:TRACe:CTJitter:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.ctJitter.current.read(ppdu,
↳ repcap.Ppdu.Default)
```

Returns the y-values of the current chip time jitter trace. See also ‘Chip Jitter square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: Comma-separated list of chip jitter values.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.ctJitter.current.clone()
```

6.5.1.8.2.3 Xvalues

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:CTJitter:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CTJitter:XVALues
```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:CTJitter:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.ctJitter.xvalues.fetch()
```

Returns the x-values of the chip time jitter trace.

Suppressed linked return values: reliability

return
values: Comma-separated list of time values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:CTJitter:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.ctJitter.xvalues.read()
```

Returns the x-values of the chip time jitter trace.

Suppressed linked return values: reliability

return
values: Comma-separated list of time values.

6.5.1.8.3 NcCorr

class NcCorrCls

NcCorr commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.ncCorr.clone()
```

Subgroups

6.5.1.8.3.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:AVERage
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.ncCorr.average.fetch()
```

Returns the y-values of the average normalized cross-correlation trace. See also ‘Normalized Cross Correlation square’.

Suppressed linked return values: reliability

return

correlation: Comma-separated list of cross-correlation values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.ncCorr.average.read()
```

Returns the y-values of the average normalized cross-correlation trace. See also ‘Normalized Cross Correlation square’.

Suppressed linked return values: reliability

return

correlation: Comma-separated list of cross-correlation values.

6.5.1.8.3.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.ncCorr.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.ncCorr.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.ncCorr.current.fetch(ppdu = ↵
↵repcap.Ppdu.Default)

```

Returns the y-values of the current normalized cross-correlation trace. See also ‘Normalized Cross Correlation square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

correlation: Comma-separated list of cross-correlation values.

read(ppdu=Ppdu.Default) → List[float]

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.ncCorr.current.read(ppdu = ↵
↵repcap.Ppdu.Default)

```

Returns the y-values of the current normalized cross-correlation trace. See also ‘Normalized Cross Correlation square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

correlation: Comma-separated list of cross-correlation values.

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.ncCorr.current.clone()

```

6.5.1.8.3.3 Xvalues

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:XVALues

```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.ncCorr.xvalues.fetch()

```

Returns the x-values of the normalized cross-correlation trace.

Suppressed linked return values: reliability

return
values: Comma-separated list of time values.

read() → List[float]

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.ncCorr.xvalues.read()

```

Returns the x-values of the normalized cross-correlation trace.

Suppressed linked return values: reliability

return
values: Comma-separated list of time values.

6.5.1.8.4 Pmask

class PmaskCls

Pmask commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.pmask.clone()

```

Subgroups

6.5.1.8.4.1 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.pmask.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.pmask.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASk:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASk:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASk:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.pmask.current.fetch(ppdu =
↳ repcap.Ppdu.Default)
```

Returns the y-values of the pulse mask trace. See also ‘Pulse Mask square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

values: Comma-separated list of normalized magnitude values.

read(ppdu=Ppdu.Default) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASk:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.pmask.current.read(ppdu =
↳ repcap.Ppdu.Default)
```

Returns the y-values of the pulse mask trace. See also ‘Pulse Mask square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

values: Comma-separated list of normalized magnitude values.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.pmask.current.clone()
```

6.5.1.8.4.2 Xvalues

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASk:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASk:XVALues
```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASk:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.pmask.xvalues.fetch()
```

Returns the x-values of the pulse mask trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of normalized time values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASk:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.pmask.xvalues.read()
```

Returns the x-values of the pulse mask trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of normalized time values.

6.5.1.8.5 PowerVsTime

class PowerVsTimeCls

PowerVsTime commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.powerVsTime.clone()
```

Subgroups

6.5.1.8.5.1 Maximum

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MAXimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MAXimum<PPDU>
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MAXimum<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.powerVsTime.maximum.
↪ fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the y-values of the power vs time trace. The minimum and maximum values can be retrieved. See also 'Power vs Time square'.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: Comma-separated list of power values.

read(ppdu=Ppdu.Nr1) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MAXimum<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.powerVsTime.maximum.
↪ read(ppdu = repcap.Ppdu.Nr1)
```

Returns the y-values of the power vs time trace. The minimum and maximum values can be retrieved. See also 'Power vs Time square'.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: Comma-separated list of power values.

6.5.1.8.5.2 Minimum

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MINimum<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MINimum<PPDU>

```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → List[float]

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MINimum<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.powerVsTime.minimum.
↪ fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the y-values of the power vs time trace. The minimum and maximum values can be retrieved. See also 'Power vs Time square'.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: Comma-separated list of power values.

read(ppdu=Ppdu.Nr1) → List[float]

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MINimum<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.powerVsTime.minimum.
↪ read(ppdu = repcap.Ppdu.Nr1)

```

Returns the y-values of the power vs time trace. The minimum and maximum values can be retrieved. See also 'Power vs Time square'.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

power: Comma-separated list of power values.

6.5.1.8.5.3 Xvalues

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:XVALues

```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.powerVsTime.xvalues.fetch()
```

Returns the x-values of the power vs time trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of time values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.powerVsTime.xvalues.read()
```

Returns the x-values of the power vs time trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of time values.

6.5.1.8.6 Sbw1

class Sbw1Cls

Sbw1 commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.sbw1.clone()
```

Subgroups

6.5.1.8.6.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:AVERage
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.sbw1.average.fetch()
```

Returns the y-values of the average 50-MHz spectrum trace. See also ‘1MHz / 50MHz Spectrum square’.

Suppressed linked return values: reliability

return

power: Comma-separated list of power values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.sbwL.average.read()
```

Returns the y-values of the average 50-MHz spectrum trace. See also ‘1MHz / 50MHz Spectrum square’.

Suppressed linked return values: reliability

return

power: Comma-separated list of power values.

6.5.1.8.6.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.sbwL.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.sbwL.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.sbwL.current.fetch(ppdu = ↵
↵ repcap.Ppdu.Default)
```

Returns the y-values of the current 50-MHz spectrum trace. See also ‘1MHz / 50MHz Spectrum square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: Comma-separated list of power values.

read(ppdu=Ppdu.Default) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.sbwL.current.read(ppdu =
↳repcap.Ppdu.Default)
```

Returns the y-values of the current 50-MHz spectrum trace. See also ‘1MHz / 50MHz Spectrum square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

power: Comma-separated list of power values.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.sbwL.current.clone()
```

6.5.1.8.6.3 Xvalues

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:XVALues
```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.sbwL.xvalues.fetch()
```

Returns the x-values of the 50-MHz spectrum trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of frequency values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWL:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.sbwL.xvalues.read()
```

Returns the x-values of the 50-MHz spectrum trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of frequency values.

6.5.1.8.7 Sbws

class SbwsCls

Sbws commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.sbws.clone()
```

Subgroups

6.5.1.8.7.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:AVERage
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.sbws.average.fetch()
```

Returns the y-values of the average 1-MHz spectrum trace. See also ‘1MHz / 50MHz Spectrum square’.

Suppressed linked return values: reliability

return
power: Comma-separated list of power values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.sbws.average.read()
```

Returns the y-values of the average 1-MHz spectrum trace. See also ‘1MHz / 50MHz Spectrum square’.

Suppressed linked return values: reliability

return
power: Comma-separated list of power values.

6.5.1.8.7.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.sbws.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.sbws.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:CURRENT<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:CURRENT<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:CURRENT<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.sbws.current.fetch(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the y-values of the current 1-MHz spectrum trace. See also '1MHz / 50MHz Spectrum square'.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

power: Comma-separated list of power values.

read(ppdu=Ppdu.Default) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:CURRENT<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.sbws.current.read(ppdu = ↵
↵repcap.Ppdu.Default)
```

Returns the y-values of the current 1-MHz spectrum trace. See also '1MHz / 50MHz Spectrum square'.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

power: Comma-separated list of power values.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.sbws.current.clone()
```

6.5.1.8.7.3 Xvalues

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:XVALues
```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.sbws.xvalues.fetch()
```

Returns the x-values of the 1-MHz spectrum trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of frequency values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.sbws.xvalues.read()
```

Returns the x-values of the 1-MHz spectrum trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of frequency values.

6.5.1.8.8 SpJitter

class SpJitterCls

SpJitter commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.spJitter.clone()
```

Subgroups

6.5.1.8.8.1 Average

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:AVERage
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.spJitter.average.fetch()
```

Returns the y-values of the average symbol phase jitter trace. See also ‘Symbol Jitter square’.

Suppressed linked return values: reliability

return

jitter: Comma-separated list of symbol jitter values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.spJitter.average.read()
```

Returns the y-values of the average symbol phase jitter trace. See also ‘Symbol Jitter square’.

Suppressed linked return values: reliability

return

jitter: Comma-separated list of symbol jitter values.

6.5.1.8.8.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.spJitter.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.spJitter.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.spJitter.current.fetch(ppdu,
↳= repcap.Ppdu.Default)

```

Returns the y-values of the current symbol phase jitter trace. See also ‘Symbol Jitter square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: Comma-separated list of symbol jitter values.

read(ppdu=Ppdu.Default) → List[float]

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:CURRent<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.spJitter.current.read(ppdu,
↳= repcap.Ppdu.Default)

```

Returns the y-values of the current symbol phase jitter trace. See also ‘Symbol Jitter square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: Comma-separated list of symbol jitter values.

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.spJitter.current.clone()

```

6.5.1.8.8.3 Xvalues

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:XVALues
```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.spJitter.xvalues.fetch()
```

Returns the x-values of the symbol phase jitter trace.

Suppressed linked return values: reliability

return
values: Comma-separated list of phase values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.spJitter.xvalues.read()
```

Returns the x-values of the symbol phase jitter trace.

Suppressed linked return values: reliability

return
values: Comma-separated list of phase values.

6.5.1.8.9 StJitter

class StJitterCls

StJitter commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.stJitter.clone()
```

Subgroups

6.5.1.8.9.1 Average

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:AVERage
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:AVERage

```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```

# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.stJitter.average.fetch()

```

Returns the y-values of the average symbol time jitter trace. See also 'Symbol Jitter square'.

Suppressed linked return values: reliability

return

jitter: Comma-separated list of symbol jitter values.

read() → List[float]

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.stJitter.average.read()

```

Returns the y-values of the average symbol time jitter trace. See also 'Symbol Jitter square'.

Suppressed linked return values: reliability

return

jitter: Comma-separated list of symbol jitter values.

6.5.1.8.9.2 Current<Ppdu>

RepCap Settings

```

# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.stJitter.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.stJitter.current.repcap_ppdu_set(repcap.Ppdu.Nr1)

```

SCPI Commands :

```

FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:CURRent<PPDU>

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:CURRENT<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.stJitter.current.fetch(ppdu,
↳ = repcap.Ppdu.Default)
```

Returns the y-values of the current symbol time jitter trace. See also ‘Symbol Jitter square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: Comma-separated list of symbol jitter values.

read(ppdu=Ppdu.Default) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:CURRENT<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.stJitter.current.read(ppdu,
↳ = repcap.Ppdu.Default)
```

Returns the y-values of the current symbol time jitter trace. See also ‘Symbol Jitter square’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

jitter: Comma-separated list of symbol jitter values.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.stJitter.current.clone()
```

6.5.1.8.9.3 Xvalues

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:XVALues
```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.stJitter.xvalues.fetch()
```

Returns the x-values of the symbol time jitter trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of time values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.stJitter.xvalues.read()
```

Returns the x-values of the symbol time jitter trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of time values.

6.5.1.8.10 TsMask

class TsMaskCls

TsMask commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.tsMask.clone()
```

Subgroups

6.5.1.8.10.1 Average

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:AVERage
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.tsMask.average.fetch()
```

Returns the y-values of the average transmit spectrum trace. See also ‘Narrowband results’.

Suppressed linked return values: reliability

return

ratio: Comma-separated list of power values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:AVERage
value: List[float] = driver.uwbMeas.multiEval.trace.tsMask.average.read()
```

Returns the y-values of the average transmit spectrum trace. See also ‘Narrowband results’.

Suppressed linked return values: reliability

return

ratio: Comma-separated list of power values.

6.5.1.8.10.2 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.trace.tsMask.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.trace.tsMask.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:CURREnt<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:CURREnt<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

fetch(ppdu=Ppdu.Default) → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:CURREnt<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.tsMask.current.fetch(ppdu =
↳ repcap.Ppdu.Default)
```

Returns the y-values of the current transmit spectrum trace. See also ‘Narrowband results’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

ratio: Comma-separated list of power values.

read(ppdu=Ppdu.Default) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:CURREnt<PPDU>
value: List[float] = driver.uwbMeas.multiEval.trace.tsMask.current.read(ppdu =
↳ repcap.Ppdu.Default)
```

Returns the y-values of the current transmit spectrum trace. See also ‘Narrowband results’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

ratio: Comma-separated list of power values.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.trace.tsMask.current.clone()
```

6.5.1.8.10.3 Xvalues**SCPI Commands :**

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:XVALues
READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:XVALues
```

class XvaluesCls

Xvalues commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.tsMask.xvalues.fetch()
```

Returns the x-values of the transmit spectrum trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of frequency values.

read() → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:XVALues
value: List[float] = driver.uwbMeas.multiEval.trace.tsMask.xvalues.read()
```

Returns the x-values of the transmit spectrum trace.

Suppressed linked return values: reliability

return

values: Comma-separated list of frequency values.

6.5.1.9 TsMask

class TsMaskCls

TsMask commands group definition. 17 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.clone()
```

Subgroups

6.5.1.9.1 Margin

class MarginCls

Margin commands group definition. 10 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.margin.clone()
```

Subgroups

6.5.1.9.1.1 Area<Area>

RepCap Settings

```
# Range: Nr1 .. Nr3
rc = driver.uwbMeas.multiEval.tsMask.margin.area.repcap_area_get()
driver.uwbMeas.multiEval.tsMask.margin.area.repcap_area_set(repcap.Area.Nr1)
```

class AreaCls

Area commands group definition. 10 total commands, 4 Subgroups, 0 group commands Repeated Capability: Area, default value after init: Area.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.margin.area.clone()
```

Subgroups

6.5.1.9.1.2 Average

SCPI Command :

```
CALCulate:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:AVERage<PPDU>
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Margin_Avg_Neg_Y: enums.ResultStatus2: Limit check result for area no with negative frequency offset.
- Margin_Avg_Pos_Y: enums.ResultStatus2: Limit check result for area no with positive frequency offset.

calculate(area=Area.Default, ppdu=Ppdu.Nr1) → CalculateStruct

```
# SCPI: CALCulate:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↳:AVERage<PPDU>
value: CalculateStruct = driver.uwbMeas.multiEval.tsMask.margin.area.average.
↳calculate(area = repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)
```

Returns the limit check results for the current and average traces, for the transmit spectrum mask area <no>. See also 'Narrowband results'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for CalculateStruct structure arguments.

6.5.1.9.1.3 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.tsMask.margin.area.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.tsMask.margin.area.current.repcap_ppdu_set(repcap.Ppdu.Nr1)
```

SCPI Command :

```
CALCulate:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Margin_Curr_Neg_Y: enums.ResultStatus2: Limit check result for area no with negative frequency offset.
- Margin_Curr_Pos_Y: enums.ResultStatus2: Limit check result for area no with positive frequency offset.

calculate(area=Area.Default, ppdu=Ppdu.Default) → CalculateStruct

```
# SCPI: CALCulate:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↪:CURRent<PPDU>
value: CalculateStruct = driver.uwbMeas.multiEval.tsMask.margin.area.current.
↪calculate(area = repcap.Area.Default, ppdu = repcap.Ppdu.Default)
```

Returns the limit check results for the current and average traces, for the transmit spectrum mask area <no>. See also 'Narrowband results'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for CalculateStruct structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.margin.area.current.clone()
```

6.5.1.9.1.4 Negativ**class NegativCls**

Negativ commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.margin.area.negativ.clone()
```

Subgroups

6.5.1.9.1.5 Average

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:NEGativ:AVERAge<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:NEGativ:AVERAge<PPDU>
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Margin_Aver_Neg_X: float: X-position of the margin for the area no
- Margin_Aver_Neg_Y: float: Y-value of the margin for the area no

fetch(area=Area.Default, ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↪:NEGativ:AVERAge<PPDU>
value: ResultData = driver.uwbMeas.multiEval.tsMask.margin.area.negativ.average.
↪fetch(area = repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)
```

Returns the margin values between the result trace and the transmit spectrum mask for the area <no> with negative frequency offset. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. The current and average values can be retrieved. See also 'Narrowband results'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(area=Area.Default, ppdu=Ppdu.Nr1) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↪:NEGativ:AVERAge<PPDU>
value: ResultData = driver.uwbMeas.multiEval.tsMask.margin.area.negativ.average.
↪read(area = repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)
```

Returns the margin values between the result trace and the transmit spectrum mask for the area <no> with negative frequency offset. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. The current and average values can be retrieved. See also ‘Narrowband results’.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Area’)

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.9.1.6 Current<Ppdu>

RepCap Settings

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.tsMask.margin.area.negativ.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.tsMask.margin.area.negativ.current.repcap_ppdu_set(repcap.Ppdu.
↪Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:NEGativ:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:NEGativ:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

class ResultData

Response structure. Fields:

- Reliability: int: ‘Reliability indicator’
- Margin_Curr_Neg_X: float: No parameter help available
- Margin_Curr_Neg_Y: float: No parameter help available

fetch(area=Area.Default, ppdu=Ppdu.Default) → ResultData

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↪:NEGativ:CURRent<PPDU>
value: ResultData = driver.uwbMeas.multiEval.tsMask.margin.area.negativ.current.
↪fetch(area = repcap.Area.Default, ppdu = repcap.Ppdu.Default)
```

Returns the margin values between the result trace and the transmit spectrum mask for the area <no> with negative frequency offset. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. The current and average values can be retrieved. See also ‘Narrowband results’.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Area’)

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

structure: for return value, see the help for ResultData structure arguments.

read(area=Area.Default, ppdu=Ppdu.Default) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↪:NEGativ:CURRent<PPDU>
value: ResultData = driver.uwbMeas.multiEval.tsMask.margin.area.negative.current.
↪read(area = repcap.Area.Default, ppdu = repcap.Ppdu.Default)
```

Returns the margin values between the result trace and the transmit spectrum mask for the area <no> with negative frequency offset. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. The current and average values can be retrieved. See also ‘Narrowband results’.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Area’)

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

structure: for return value, see the help for ResultData structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.margin.area.negative.current.clone()
```

6.5.1.9.1.7 Positiv**class PositivCls**

Positiv commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.margin.area.positive.clone()
```

Subgroups

6.5.1.9.1.8 Average

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:POSitiv:AVERage<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:POSitiv:AVERage<PPDU>

```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Margin_Aver_Pos_X: float: X-position of the margin for the area no
- Margin_Aver_Pos_Y: float: Y-value of the margin for the area no

fetch(area=Area.Default, ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↳:POSitiv:AVERage<PPDU>
value: ResultData = driver.uwbMeas.multiEval.tsMask.margin.area.positiv.average.
↳fetch(area = repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)

```

Returns the margin values between the result trace and the transmit spectrum mask for the area <no> with positive frequency offset. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. The current and average values can be retrieved. See also 'Narrowband results'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

read(area=Area.Default, ppdu=Ppdu.Nr1) → ResultData

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↳:POSitiv:AVERage<PPDU>
value: ResultData = driver.uwbMeas.multiEval.tsMask.margin.area.positiv.average.
↳read(area = repcap.Area.Default, ppdu = repcap.Ppdu.Nr1)

```

Returns the margin values between the result trace and the transmit spectrum mask for the area <no> with positive frequency offset. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. The current and average values can be retrieved. See also 'Narrowband results'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1

return

structure: for return value, see the help for ResultData structure arguments.

6.5.1.9.1.9 Current<Ppdu>**RepCap Settings**

```
# Range: Nr1 .. Nr100
rc = driver.uwbMeas.multiEval.tsMask.margin.area.positiv.current.repcap_ppdu_get()
driver.uwbMeas.multiEval.tsMask.margin.area.positiv.current.repcap_ppdu_set(repcap.Ppdu.
↳Nr1)
```

SCPI Commands :

```
FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:POSitiv:CURRent<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>:POSitiv:CURRent<PPDU>
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: Ppdu, default value after init: Ppdu.Nr1

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Margin_Curr_Pos_X: float: No parameter help available
- Margin_Curr_Pos_Y: float: No parameter help available

fetch(area=Area.Default, ppdu=Ppdu.Default) → ResultData

```
# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↳:POSitiv:CURRent<PPDU>
value: ResultData = driver.uwbMeas.multiEval.tsMask.margin.area.positiv.current.
↳fetch(area = repcap.Area.Default, ppdu = repcap.Ppdu.Default)
```

Returns the margin values between the result trace and the transmit spectrum mask for the area <no> with positive frequency offset. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. The current and average values can be retrieved. See also 'Narrowband results'.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Area')

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface 'Current')

return

structure: for return value, see the help for ResultData structure arguments.

read(area=Area.Default, ppdu=Ppdu.Default) → ResultData

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:AREA<nr>
↪:POSitiv:CURRent<PPDU>
value: ResultData = driver.uwbMeas.multiEval.tsMask.margin.area.positiv.current.
↪read(area = repcap.Area.Default, ppdu = repcap.Ppdu.Default)
```

Returns the margin values between the result trace and the transmit spectrum mask for the area <no> with positive frequency offset. A negative margin indicates that the trace is located above the limit line, i.e. the limit is exceeded. The current and average values can be retrieved. See also ‘Narrowband results’.

param area

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Area’)

param ppdu

optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Current’)

return

structure: for return value, see the help for ResultData structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.margin.area.positiv.current.clone()
```

6.5.1.9.2 Otolerance

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:OTOLerance
READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:OTOLerance
```

class OtoleranceCls

Otolerance commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → float

```
# SCPI: FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:OTOLerance
value: float = driver.uwbMeas.multiEval.tsMask.otolerance.fetch()
```

Returns the out of tolerance result for transmit spectrum mask measurements. It indicates the percentage of measurement intervals of the statistic count for spectrum measurements exceeding the specified Transmit Spectrum Mask limits.

Suppressed linked return values: reliability

return

out_of_tolerance: No help available

read() → float

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:OTolerance
value: float = driver.uwbMeas.multiEval.tsMask.otolerance.read()
```

Returns the out of tolerance result for transmit spectrum mask measurements. It indicates the percentage of measurement intervals of the statistic count for spectrum measurements exceeding the specified Transmit Spectrum Mask limits.

Suppressed linked return values: reliability

```
return
    out_of_tolerance: No help available
```

6.5.1.9.3 TdbBandwidth

class TdbBandwidthCls

TdbBandwidth commands group definition. 5 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.tdbBandwidth.clone()
```

Subgroups

6.5.1.9.3.1 Frequency

SCPI Commands :

```
FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandwidth:FREquency<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandwidth:FREquency<PPDU>
```

class FrequencyCls

Frequency commands group definition. 3 total commands, 1 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → List[float]

```
# SCPI: FETCH:UWB:MEASurement<Instance>
↪:MEvaluation:TSMask:TDBBandwidth:FREquency<PPDU>
value: List[float] = driver.uwbMeas.multiEval.tsMask.tdbBandwidth.frequency.
↪fetch(ppdu = repcap.Ppdu.Nr1)
```

Returns the frequency values fM, fL, fH, fH fL for the -10 dB bandwidth limit as defined in ANSI C63.10-2013, chapter 10.1 ‘Evaluation of -10 dB bandwidth’.

Suppressed linked return values: reliability

```
param ppdu
    optional repeated capability selector. Default value: Nr1
```

```
return
    frequency: Comma-separated list of frequencies in the order fM, fL, fH, fH fL.
```

read(ppdu=Ppdu.Nr1) → List[float]

```
# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandwidth:FREquency
↳<PPDU>
value: List[float] = driver.uwbMeas.multiEval.tsMask.tdbBandwidth.frequency.
↳read(ppdu = repcap.Ppdu.Nr1)
```

Returns the frequency values fM, fL, fH, fH fL for the -10 dB bandwidth limit as defined in ANSI C63.10-2013, chapter 10.1 ‘Evaluation of -10 dB bandwidth’.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

frequency: Comma-separated list of frequencies in the order fM, fL, fH, fH fL.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.uwbMeas.multiEval.tsMask.tdbBandwidth.frequency.clone()
```

Subgroups

6.5.1.9.3.2 Fhfl

SCPI Command :

```
CALCulate:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandwidth:FREquency:FHFL<PPDU>
```

class FhflCls

Fhfl commands group definition. 1 total commands, 0 Subgroups, 1 group commands

calculate(ppdu=Ppdu.Nr1) → ResultStatus2

```
# SCPI: CALCulate:UWB:MEASurement<Instance>
↳:MEvaluation:TSMask:TDBBandwidth:FREquency:FHFL<PPDU>
value: enums.ResultStatus2 = driver.uwbMeas.multiEval.tsMask.tdbBandwidth.
↳frequency.fhfl.calculate(ppdu = repcap.Ppdu.Nr1)
```

Returns the limit check result for the 10 dB bandwidth (fH fL) 500 MHz.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

frequency: No help available

6.5.1.9.3.3 PsPower

SCPI Commands :

```

FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandwidth:PSPower<PPDU>
READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandwidth:PSPower<PPDU>

```

class PsPowerCls

PsPower commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(ppdu=Ppdu.Nr1) → List[float]

```

# SCPI: FETCh:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandwidth:PSPower
→<PPDU>
value: List[float] = driver.uwbMeas.multiEval.tsMask.tdbBandwidth.psPower.
→fetch(ppdu = repcap.Ppdu.Nr1)

```

Returns the peak spectral power results at the frequencies fM, fL, fH for the -10 dB bandwidth limit as defined in ANSI C63.10-2013, chapter 10.1 'Evaluation of -10 dB bandwidth'.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak_spectr_power: Comma-separated list of peak spectral power values at fM, fL, fH, NAV for fH fL.

read(ppdu=Ppdu.Nr1) → List[float]

```

# SCPI: READ:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandwidth:PSPower
→<PPDU>
value: List[float] = driver.uwbMeas.multiEval.tsMask.tdbBandwidth.psPower.
→read(ppdu = repcap.Ppdu.Nr1)

```

Returns the peak spectral power results at the frequencies fM, fL, fH for the -10 dB bandwidth limit as defined in ANSI C63.10-2013, chapter 10.1 'Evaluation of -10 dB bandwidth'.

Suppressed linked return values: reliability

param ppdu

optional repeated capability selector. Default value: Nr1

return

peak_spectr_power: Comma-separated list of peak spectral power values at fM, fL, fH, NAV for fH fL.

RSCMPX_UWBMEAS UTILITIES

class Utilities

Common utility class. Utility functions common for all types of drivers.

Access snippet: `utils = RsCMPX_UwbMeas.utilities`

property logger: *ScpiLogger*

Scpi Logger interface, see [here](#)

Access snippet: `logger = RsCMPX_UwbMeas.utilities.logger`

property driver_version: `str`

Returns the instrument driver version.

property idn_string: `str`

Returns instrument's identification string - the response on the SCPI command `*IDN?`

property manufacturer: `str`

Returns manufacturer of the instrument.

property full_instrument_model_name: `str`

Returns the current instrument's full name e.g. 'FSW26'.

property instrument_model_name: `str`

Returns the current instrument's family name e.g. 'FSW'.

property supported_models: `List[str]`

Returns a list of the instrument models supported by this instrument driver.

property instrument_firmware_version: `str`

Returns instrument's firmware version.

property instrument_serial_number: `str`

Returns instrument's serial_number.

query_opc(*timeout: int = 0*) → `int`

SCPI command: `*OPC?` Queries the instrument's OPC bit and hence it waits until the instrument reports operation complete. If you define `timeout > 0`, the VISA timeout is set to that value just for this method call.

property instrument_status_checking: `bool`

Sets / returns Instrument Status Checking. When True (default is True), all the driver methods and properties are sending "SYSTem:ERRor?" at the end to immediately react on error that might have occurred. We recommend to keep the state checking ON all the time. Switch it OFF only in rare cases when you require maximum speed. The default state after initializing the session is ON.

property encoding: str

Returns string<=>bytes encoding of the session.

property opc_query_after_write: bool

Sets / returns Instrument *OPC? query sending after each command write. When True, (default is False) the driver sends *OPC? every time a write command is performed. Use this if you want to make sure your sequence is performed command-after-command.

property bin_float_numbers_format: BinFloatFormat

Sets / returns format of float numbers when transferred as binary data.

property bin_int_numbers_format: BinIntFormat

Sets / returns format of integer numbers when transferred as binary data.

clear_status() → None

Clears instrument's status system, the session's I/O buffers and the instrument's error queue.

query_all_errors() → List[str]

Queries and clears all the errors from the instrument's error queue. The method returns list of strings as error messages. If no error is detected, the return value is None. The process is: querying 'SYS-Tem:ERRor?' in a loop until the error queue is empty. If you want to include the error codes, call the query_all_errors_with_codes()

query_all_errors_with_codes() → List[Tuple[int, str]]

Queries and clears all the errors from the instrument's error queue. The method returns list of tuples (code: int, message: str). If no error is detected, the return value is None. The process is: querying 'SYS-Tem:ERRor?' in a loop until the error queue is empty.

property instrument_options: List[str]

Returns all the instrument options. The options are sorted in the ascending order starting with K-options and continuing with B-options.

reset() → None

SCPI command: *RST Sends *RST command + calls the clear_status().

default_instrument_setup() → None

Custom steps performed at the init and at the reset().

self_test(timeout: int = None) → Tuple[int, str]

SCPI command: *TST? Performs instrument's self-test. Returns tuple (code:int, message: str). Code 0 means the self-test passed. You can define the custom timeout in milliseconds. If you do not define it, the default selftest timeout is used (usually 60 secs).

is_connection_active() → bool

Returns true, if the VISA connection is active and the communication with the instrument still works.

reconnect(force_close: bool = False) → bool

If the connection is not active, the method tries to reconnect to the device. If the connection is active, and force_close is False, the method does nothing. If the connection is active, and force_close is True, the method closes, and opens the session again. Returns True, if the reconnection has been performed.

property resource_name: int

Returns the resource name used in the constructor

property opc_timeout: int

Sets / returns timeout in milliseconds for all the operations that use OPC synchronization.

property visa_timeout: int

Sets / returns visa IO timeout in milliseconds.

property data_chunk_size: int

Sets / returns the maximum size of one block transferred during write/read operations

property visa_manufacturer: int

Returns the manufacturer of the current VISA session.

process_all_commands() → None

SCPI command: ***WAI** Stops further commands processing until all commands sent before ***WAI** have been executed.

write_str(cmd: str) → None

Writes the command to the instrument.

write(cmd: str) → None

This method is an alias to the write_str(). Writes the command to the instrument as string.

write_int(cmd: str, param: int) → None

Writes the command to the instrument followed by the integer parameter: e.g.: cmd = 'SELECT:INPUT' param = '2', result command = 'SELECT:INPUT 2'

write_int_with_opc(cmd: str, param: int, timeout: int = None) → None

Writes the command with OPC to the instrument followed by the integer parameter: e.g.: cmd = 'SELECT:INPUT' param = '2', result command = 'SELECT:INPUT 2' If you do not provide timeout, the method uses current opc_timeout.

write_float(cmd: str, param: float) → None

Writes the command to the instrument followed by the boolean parameter: e.g.: cmd = 'CENTER:FREQ' param = '10E6', result command = 'CENTER:FREQ 10E6'

write_float_with_opc(cmd: str, param: float, timeout: int = None) → None

Writes the command with OPC to the instrument followed by the boolean parameter: e.g.: cmd = 'CENTER:FREQ' param = '10E6', result command = 'CENTER:FREQ 10E6' If you do not provide timeout, the method uses current opc_timeout.

write_bool(cmd: str, param: bool) → None

Writes the command to the instrument followed by the boolean parameter: e.g.: cmd = 'OUTPUT' param = 'True', result command = 'OUTPUT ON'

write_bool_with_opc(cmd: str, param: bool, timeout: int = None) → None

Writes the command with OPC to the instrument followed by the boolean parameter: e.g.: cmd = 'OUTPUT' param = 'True', result command = 'OUTPUT ON' If you do not provide timeout, the method uses current opc_timeout.

query_str(query: str) → str

Sends the query to the instrument and returns the response as string. The response is trimmed of any trailing LF characters and has no length limit.

query(query: str) → str

This method is an alias to the query_str(). Sends the query to the instrument and returns the response as string. The response is trimmed of any trailing LF characters and has no length limit.

query_bool(query: str) → bool

Sends the query to the instrument and returns the response as boolean.

query_int(*query: str*) → int

Sends the query to the instrument and returns the response as integer.

query_float(*query: str*) → float

Sends the query to the instrument and returns the response as float.

write_str_with_opc(*cmd: str, timeout: int = None*) → None

Writes the opc-synced command to the instrument. If you do not provide timeout, the method uses current `opc_timeout`.

write_with_opc(*cmd: str, timeout: int = None*) → None

This method is an alias to the `write_str_with_opc()`. Writes the opc-synced command to the instrument. If you do not provide timeout, the method uses current `opc_timeout`.

query_str_with_opc(*query: str, timeout: int = None*) → str

Sends the opc-synced query to the instrument and returns the response as string. The response is trimmed of any trailing LF characters and has no length limit. If you do not provide timeout, the method uses current `opc_timeout`.

query_with_opc(*query: str, timeout: int = None*) → str

This method is an alias to the `query_str_with_opc()`. Sends the opc-synced query to the instrument and returns the response as string. The response is trimmed of any trailing LF characters and has no length limit. If you do not provide timeout, the method uses current `opc_timeout`.

query_bool_with_opc(*query: str, timeout: int = None*) → bool

Sends the opc-synced query to the instrument and returns the response as boolean. If you do not provide timeout, the method uses current `opc_timeout`.

query_int_with_opc(*query: str, timeout: int = None*) → int

Sends the opc-synced query to the instrument and returns the response as integer. If you do not provide timeout, the method uses current `opc_timeout`.

query_float_with_opc(*query: str, timeout: int = None*) → float

Sends the opc-synced query to the instrument and returns the response as float. If you do not provide timeout, the method uses current `opc_timeout`.

write_bin_block(*cmd: str, payload: bytes*) → None

Writes all the payload as binary data block to the instrument. The binary data header is added at the beginning of the transmission automatically, do not include it in the payload!!!

query_bin_block(*query: str*) → bytes

Queries binary data block to bytes. Throws an exception if the returned data was not a binary data. Returns `data:bytes`

query_bin_block_with_opc(*query: str, timeout: int = None*) → bytes

Sends a OPC-synced query and returns binary data block to bytes. If you do not provide timeout, the method uses current `opc_timeout`.

query_bin_or_ascii_float_list(*query: str*) → List[float]

Queries a list of floating-point numbers that can be returned in ASCII format or in binary format. - For ASCII format, the list numbers are decoded as comma-separated values. - For Binary Format, the numbers are decoded based on the property `BinFloatFormat`, usually float 32-bit (FORM REAL,32).

query_bin_or_ascii_float_list_with_opc(*query: str, timeout: int = None*) → List[float]

Sends a OPC-synced query and reads a list of floating-point numbers that can be returned in ASCII format or in binary format. - For ASCII format, the list numbers are decoded as comma-separated values. - For Binary Format, the numbers are decoded based on the property `BinFloatFormat`, usually float 32-bit (FORM REAL,32). If you do not provide timeout, the method uses current `opc_timeout`.

query_bin_or_ascii_int_list(*query: str*) → List[int]

Queries a list of floating-point numbers that can be returned in ASCII format or in binary format. - For ASCII format, the list numbers are decoded as comma-separated values. - For Binary Format, the numbers are decoded based on the property BinFloatFormat, usually float 32-bit (FORM REAL,32).

query_bin_or_ascii_int_list_with_opc(*query: str, timeout: int = None*) → List[int]

Sends a OPC-synced query and reads a list of floating-point numbers that can be returned in ASCII format or in binary format. - For ASCII format, the list numbers are decoded as comma-separated values. - For Binary Format, the numbers are decoded based on the property BinFloatFormat, usually float 32-bit (FORM REAL,32). If you do not provide timeout, the method uses current `opc_timeout`.

query_bin_block_to_file(*query: str, file_path: str, append: bool = False*) → None

Queries binary data block to the provided file. If `append` is `False`, any existing file content is discarded. If `append` is `True`, the new content is added to the end of the existing file, or if the file does not exist, it is created. Throws an exception if the returned data was not a binary data. Example for transferring a file from Instrument -> PC: `query = f"MMEM:DATA? '{INSTR_FILE_PATH}'"`. Alternatively, use the dedicated methods for this purpose:

- `send_file_from_pc_to_instrument()`
- `read_file_from_instrument_to_pc()`

query_bin_block_to_file_with_opc(*query: str, file_path: str, append: bool = False, timeout: int = None*) → None

Sends a OPC-synced query and writes the returned data to the provided file. If `append` is `False`, any existing file content is discarded. If `append` is `True`, the new content is added to the end of the existing file, or if the file does not exist, it is created. Throws an exception if the returned data was not a binary data.

write_bin_block_from_file(*cmd: str, file_path: str*) → None

Writes data from the file as binary data block to the instrument using the provided command. Example for transferring a file from PC -> Instrument: `cmd = f"MMEM:DATA '{INSTR_FILE_PATH}',"`. Alternatively, use the dedicated methods for this purpose:

- `send_file_from_pc_to_instrument()`
- `read_file_from_instrument_to_pc()`

send_file_from_pc_to_instrument(*source_pc_file: str, target_instr_file: str*) → None

SCPI Command: `MMEM:DATA`

Sends file from PC to the instrument

read_file_from_instrument_to_pc(*source_instr_file: str, target_pc_file: str, append_to_pc_file: bool = False*) → None

SCPI Command: `MMEM:DATA?`

Reads file from instrument to the PC.

Set the `append_to_pc_file` to `True` if you want to append the read content to the end of the existing PC file

get_last_sent_cmd() → str

Returns the last commands sent to the instrument. Only works in simulation mode

go_to_local() → None

Puts the instrument into local state.

go_to_remote() → None

Puts the instrument into remote state.

get_lock() → RLock

Returns the thread lock for the current session.

By default:

- If you create standard new RsCMPX_UwbMeas instance with new VISA session, the session gets a new thread lock. You can assign it to other RsCMPX_UwbMeas sessions in order to share one physical instrument with a multi-thread access.
- If you create new RsCMPX_UwbMeas from an existing session, the thread lock is shared automatically making both instances multi-thread safe.

You can always assign new thread lock by calling `driver.utilities.assign_lock()`

assign_lock(lock: RLock) → None

Assigns the provided thread lock.

clear_lock()

Clears the existing thread lock, making the current session thread-independent from others that might share the current thread lock.

sync_from(source: Utilities) → None

Synchronises these Utils with the source.

RSCMPX_UWBMEAS LOGGER

Check the usage in the Getting Started chapter [here](#).

class ScpiLogger

Base class for SCPI logging

mode

Sets the logging ON or OFF. Additionally, you can set the logging ON only for errors. Possible values:

- `LoggingMode.Off` - logging is switched OFF
- `LoggingMode.On` - logging is switched ON
- `LoggingMode.Errors` - logging is switched ON, but only for error entries
- `LoggingMode.Default` - sets the logging to default - the value you have set with `logger.default_mode`

default_mode

Sets / returns the default logging mode. You can recall the default mode by calling the `logger.mode = LoggingMode.Default`.

Data Type

`LoggingMode`

device_name: str

Use this property to change the resource name in the log from the default Resource Name (e.g. `TCPIP::192.168.2.101::INSTR`) to another name e.g. `'MySigGen1'`.

set_logging_target(target, console_log: bool = None, udp_log: bool = None) → None

Sets logging target - the target must implement `write()` and `flush()`. You can optionally set the console and UDP logging ON or OFF. This method switches the logging target global OFF.

get_logging_target()

Based on the `global_mode`, it returns the logging target: either the local or the global one.

set_logging_target_global(console_log: bool = None, udp_log: bool = None) → None

Sets logging target to global. The global target must be defined. You can optionally set the console and UDP logging ON or OFF.

log_to_console

Returns logging to console status.

log_to_udp

Returns logging to UDP status.

log_to_console_and_udp

Returns true, if both logging to UDP and console in are True.

info_raw(log_entry: str, add_new_line: bool = True) → None

Method for logging the raw string without any formatting.

info(start_time: datetime, end_time: datetime, log_string_info: str, log_string: str) → None

Method for logging one info entry. For binary log_string, use the info_bin()

error(start_time: datetime, end_time: datetime, log_string_info: str, log_string: str) → None

Method for logging one error entry.

set_relative_timestamp(timestamp: datetime) → None

If set, the further timestamps will be relative to the entered time.

set_relative_timestamp_now() → None

Sets the relative timestamp to the current time.

get_relative_timestamp() → datetime

Based on the global_mode, it returns the relative timestamp: either the local or the global one.

clear_relative_timestamp() → None

Clears the reference time, and the further logging continues with absolute times.

flush() → None

Flush all the entries.

log_status_check_ok

Sets / returns the current status of status checking OK. If True (default), the log contains logging of the status checking 'Status check: OK'. If False, the 'Status check: OK' is skipped - the log is more compact. Errors will still be logged.

clear_cached_entries() → None

Clears potential cached log entries. Cached log entries are generated when the Logging is ON, but no target has been defined yet.

set_format_string(value: str, line_divider: str = '\n') → None

Sets new format string and line divider. If you just want to set the line divider, set the format string value=None. The original format string is: PAD_LEFT12(%START_TIME%) PAD_LEFT25(%DEVICE_NAME%) PAD_LEFT12(%DURATION%) %LOG_STRING_INFO% %LOG_STRING%

restore_format_string() → None

Restores the original format string and the line divider to LF

abbreviated_max_len_ascii: int

Defines the maximum length of one ASCII log entry. Default value is 200 characters.

abbreviated_max_len_bin: int

Defines the maximum length of one Binary log entry. Default value is 2048 bytes.

abbreviated_max_len_list: int

Defines the maximum length of one list entry. Default value is 100 elements.

bin_line_block_size: int

Defines number of bytes to display in one line. Default value is 16 bytes.

udp_port

Returns udp logging port.

target_auto_flushing

Returns status of the auto-flushing for the logging target.

RSCMPX_UWBMEAS EVENTS

Check the usage in the Getting Started chapter [here](#).

class Events

Common Events class. Event-related methods and properties. Here you can set all the event handlers.

property before_query_handler: Callable

Returns the handler of before_query events.

Returns

current before_query_handler

property before_write_handler: Callable

Returns the handler of before_write events.

Returns

current before_write_handler

property io_events_include_data: bool

Returns the current state of the io_events_include_data See the setter for more details.

property on_read_handler: Callable

Returns the handler of on_read events.

Returns

current on_read_handler

property on_write_handler: Callable

Returns the handler of on_write events.

Returns

current on_write_handler

sync_from(source: Events) → None

Synchronises these Events with the source.

**CHAPTER
TEN**

INDEX

INDEX

A

abbreviated_max_len_ascii (*ScpiLogger attribute*),
324
abbreviated_max_len_bin (*ScpiLogger attribute*),
324
abbreviated_max_len_list (*ScpiLogger attribute*),
324
ABORT:UWB:MEASurement<Instance>:MEvaluation,
79

B

bin_line_block_size (*ScpiLogger attribute*), 324

C

CALCulate:UWB:MEASurement<Instance>:MEvaluation:
85
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
87
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
88
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
89
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
105
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
106
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
108
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
109
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
123
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
124
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
125
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
127
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
129
CALCulate:UWB:MEASurement<Instance>:MEvaluation:
130
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
131
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
132
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
134
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
135
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
137
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
138
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
140
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
141
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
143
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
144
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
145
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
147
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
148
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
149
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
150
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
153
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
154
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
155
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
156
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
168
CALCulate:UWB:MEASurement<Instance>:MEvaluation:MODulation:
169

CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	170	43
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	172	44
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	173	45
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	174	45
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	176	46
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	177	47
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	178	48
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	180	49
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	181	50
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	182	37
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	193	51
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	194	52
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	196	52
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	197	54
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	199	55
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	200	56
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	202	37
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	203	57
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	212	57
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	305	57
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	306	58
CALCulate:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:MODulation	314	59
CATalog:UWB:MEAS<instance>:SPATH, 36		Configure:UWB:MEASurement<Instance>:MEvaluation:PSDU:BITRate
clear_cached_entries() (ScpiLogger method), 324		60
clear_relative_timestamp() (ScpiLogger method), 324		Configure:UWB:MEASurement<Instance>:MEvaluation:PSFormat<Format>
Configure:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:PTRacking, 37		37
Configure:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:REPetition, 37		37
Configure:UWB:MEASurement<Instance>:MEvaluation:COMODulate:UWB:MEASurement<Instance>:MEvaluation:RESult:DDF, 42		61

CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:RSPa	61	82
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:AVE	61	84
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCF	61	85
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCF	37	87
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCF	37	88
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCF	63	89
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEV	63	91
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEV	64	92
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEV	66	93
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEV	65	93
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJ	66	94
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJ	37	95
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJ	68	96
CONFIGure:UWB:MEASurement<Instance>:MEvaluation:FETCH:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJ	69	97
CONFIGure:UWB:MEASurement<Instance>:RFSettings:CHANNEL:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJ	70	98
CONFIGure:UWB:MEASurement<Instance>:RFSettings:CHANNEL:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJ	70	99
CONFIGure:UWB:MEASurement<Instance>:RFSettings:CHANNEL:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJ	70	100
CONFIGure:UWB:MEASurement<Instance>:RFSettings:FREQUENCY:UWB:MEASurement<Instance>:MEvaluation:MODulation:CTJ	72	101
CONFIGure:UWB:MEASurement<Instance>:RFSettings:FREQUENCY:UWB:MEASurement<Instance>:MEvaluation:MODulation:CUF	72	102
CONFIGure:UWB:MEASurement<Instance>:RFSettings:FREQUENCY:UWB:MEASurement<Instance>:MEvaluation:MODulation:EXT	70	103
D		105
default_mode (<i>ScpiLogger attribute</i>), 323		106
device_name (<i>ScpiLogger attribute</i>), 323		108
E		109
error() (<i>ScpiLogger method</i>), 324		110
F		111
FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CLength<PPDU>	81	111
FETCH:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CONTENT<PPDU>	82	112

FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:EXTRemInstanPPDU>:MEvaluation:PMASK:LOWER:AF
 176 210
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:SDEvInstanPPDU>:MEvaluation:PMASK:MARGin:PPDU
 177 213
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:AVERageInstanPPDU>:MEvaluation:PMASK:MARGin:PPDU
 178 214
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:CURRentInstanPPDU>:MEvaluation:PMASK:OTOLerance:PPDU
 180 215
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:EXTRemInstanPPDU>:MEvaluation:PMASK:UPPER:AF
 181 216
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:SDEvInstanPPDU>:MEvaluation:POWER:AVERage:PPDU
 182 217
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:AVERageInstanPPDU>:MEvaluation:POWER:CURRent:PPDU
 183 219
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:CURRentInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 184 220
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:EXTRemInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 185 221
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:SDEvInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 186 222
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:AVERageInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 189 223
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:CURRentInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 190 223
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:EXTRemInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 191 224
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:SDEvInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 192 225
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:AVERageInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 193 226
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:CURRentInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 194 227
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:EXTRemInstanPPDU>:MEvaluation:POWER:DPOWER:PPDU
 196 228
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:SDEvInstanPPDU>:MEvaluation:POWER:MAXimum:PPDU
 197 229
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:AVERageInstanPPDU>:MEvaluation:POWER:MINimum:PPDU
 198 230
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:CURRentInstanPPDU>:MEvaluation:POWER:MSFPower:PPDU
 199 231
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:MAXimumInstanPPDU>:MEvaluation:POWER:MSFPower:PPDU
 200 232
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:MINimumInstanPPDU>:MEvaluation:POWER:MSFPower:PPDU
 202 233
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:SDEvInstanPPDU>:MEvaluation:POWER:MSFPower:PPDU
 203 234
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:PeLenti:CURRentInstanPPDU>:MEvaluation:POWER:MSFPower:PPDU
 204 234
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:PeLenti:EXTRemInstanPPDU>:MEvaluation:POWER:MSFPower:PPDU
 205 235
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:PeLenti:CURRentInstanPPDU>:MEvaluation:POWER:MSFPower:PPDU
 207 236
 FETCH:UWB:MEASurement<Instance>:MEvaluation:MODELCh:UWB:SPASub:PeLenti:EXTRemInstanPPDU>:MEvaluation:POWER:MSFPower:PPDU
 208 237


```

238  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:DPPDu,
239  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:DRATe<PP
240  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:FCSCheck
241  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:PHR:ASSY
242  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:PHR:BITR
243  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:PHR:CR<
243  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:PSDU:CR<
245  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:PSDU:LEN
245  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:PSTGap<P
246  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:RABit<PP
247  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:REBit<PP
248  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:SFD<PPDU
249  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:SFDLength
250  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:SINFo:<PPDU>,
251  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:STATe,
252  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:STATe:ALL,
252  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:STSSequence:CL
253  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:STSSequence:CO
254  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:TRACe:CPJitter
255  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:TRACe:CPJitter
256  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:TRACe:CPJitter
257  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:TRACe:CTJitter
258  FETCH:UWB:MEASurement<Instance>:MEvaluation:POWERCHPUWB:MEASurement<PPDU>:MEvaluation:TRACe:CTJitter
260  FETCH:UWB:MEASurement<Instance>:MEvaluation:SINFo:ASSWb:MEASurement<Instance>:MEvaluation:TRACe:CTJitter
261  FETCH:UWB:MEASurement<Instance>:MEvaluation:SINFo:ASSWb:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:A
261  FETCH:UWB:MEASurement<Instance>:MEvaluation:SINFo:ASSWb:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:C
262  FETCH:UWB:MEASurement<Instance>:MEvaluation:SINFo:ASSWb:MEASurement<Instance>:MEvaluation:TRACe:NCCorr:X

```


FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:PMASK:CURRENT<PPDU>, 324
 286
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:PMASK:XVALUES,
 287
 get_logging_target() (ScriLogger method), 323
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:PVTTime:MAXimum<PPDU>,
 288
 get_relative_timestamp() (ScriLogger method), 324
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:PVTTime:MINimum<PPDU>,
 289
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:PVTTime:XVALUES,
 289
 info() (ScriLogger method), 324
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SBWL:AVERAGE,
 290
 info_raw() (ScriLogger method), 323
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SBWL:CURRENT<PPDU>,
 291
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SBWL:XVALUES,
 292
 log_status_check_ok (ScriLogger attribute), 324
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SBWS:AVERAGE,
 293
 log_to_console (ScriLogger attribute), 323
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SBWS:CURRENT<PPDU>,
 294
 log_to_udp (ScriLogger attribute), 323
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SBWS:XVALUES,
 295
 |
 mode (ScriLogger attribute), 323
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SPJitter:AVERAGE,
 296
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SPJitter:CURRENT<PPDU>,
 297
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CLENGTH,
 298
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SPJitter:XVALUES,
 298
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:CONTENT,
 299
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SPJitter:AVERAGE,
 299
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:DDECoding:RSPAR,
 299
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SPJitter:CURRENT<PPDU>,
 299
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:AVEF,
 300
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:SPJitter:XVALUES,
 300
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCEFF,
 301
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:TSMask:AVERAGE,
 301
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCEFF,
 302
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:TSMask:CURRENT<PPDU>,
 302
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCEFF,
 303
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TRACE:TSMask:XVALUES,
 303
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CCEFF,
 307
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGIN:AREA<HF>:NEGATIV:AVERAGE<PPDU>,
 308
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM,
 308
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGIN:AREA<HF>:NEGATIV:CURRENT<PPDU>,
 310
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM,
 310
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGIN:AREA<HF>:POSITIV:AVERAGE<PPDU>,
 311
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CEVM,
 311
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGIN:AREA<HF>:POSITIV:CURRENT<PPDU>,
 312
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJ,
 312
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:OBTolerance,
 313
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJ,
 313
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:IDBbandwidth:FREQUENCY<PPDU>,
 315
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJ,
 315
 |
 FETCH:UWB:MEASurement<Instance>:MEvaluation:TSMask:IDBbandwidth:POWER<PPDU>,
 315
 |
 READ:UWB:MEASurement<Instance>:MEvaluation:MODulation:CPJ,
 315
 |

READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:CU
 225 250
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:MA
 226 251
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:MI
 227 252
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:SD
 228 252
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:A
 229 253
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:C
 230 254
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:M
 231 255
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:M
 232 256
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 233 257
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 234 258
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 234 260
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 235 261
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 236 261
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 237 262
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 238 263
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 239 264
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 240 264
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 241 265
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 242 266
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 243 267
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 243 268
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 245 269
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 245 269
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 246 270
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 247 271
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 248 272
 READ:UWB:MEASurement<Instance>:MEvaluation:POWER:PPower:PPower:S
 249 272

READ:UWB:MEASurement<Instance>:MEvaluation:STStutter:PPDU<Instance>:MEvaluation:TRACe:STJitter:PPDU
 259 299
 READ:UWB:MEASurement<Instance>:MEvaluation:STStutter:PPDU<Instance>:MEvaluation:TRACe:STJitter:PPDU
 275 300
 READ:UWB:MEASurement<Instance>:MEvaluation:STStutter:PPDU<Instance>:MEvaluation:TRACe:TSMask:AV
 276 301
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:CU
 277 302
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TRACe:TSMask:XV
 278 303
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:A
 279 307
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:A
 280 308
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:A
 281 310
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TSMask:MARGin:A
 282 311
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TSMask:OTOLerar
 283 312
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandw
 284 313
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:UWB:MEASurement<Instance>:MEvaluation:TSMask:TDBBandw
 285 315
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASK:CURRENT:PPDU (ScpiLogger method), 324
 286 ROUTe:UWB:MEAS<instance>:SPATH, 73
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PMASK:AVASurement<Instance>:RFSettings:CONNector,
 287 74
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MAXimum<PPDU>,
 288 S
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:MINimum<PPDU> (ScpiLogger method), 324
 289 323
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:PVTime:XVALues (ScpiLogger method), 324
 289 set_logging_target() (ScpiLogger method), 323
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:AVERAge
 290 323
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:CURRENT:PPDU
 291 324
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:KVALues
 292 324
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:KVALues (ScpiLogger method), 324
 292 method), 324
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:MEASurement<Instance>:MEvaluation,
 293 79
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:CURRENT<PPDU>,
 294 T
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SBWS:XVALues (ScpiLogger attribute), 324
 295 TRIGger:UWB:MEASurement<Instance>:MEvaluation:CATalog:SOU
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:AVERAge,
 296 TRIGger:UWB:MEASurement<Instance>:MEvaluation:DELay,
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:CURRENT<PPDU>,
 297 TRIGger:UWB:MEASurement<Instance>:MEvaluation:MGAP,
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:SPJitter:XVALues,
 298 TRIGger:UWB:MEASurement<Instance>:MEvaluation:SLOPe,
 READ:UWB:MEASurement<Instance>:MEvaluation:TRACe:STJitter:AVERAge,
 299

TRIGger:UWB:MEASurement<Instance>:MEValuation:SOURce,
75
TRIGger:UWB:MEASurement<Instance>:MEValuation:THReshold,
75
TRIGger:UWB:MEASurement<Instance>:MEValuation:TOUT,
75

U

udp_port (*ScpiLogger attribute*), 324