MYTEK

TLKCore Reference Guide

Version 0.1.7

20231120

Prerequisite

1. About TLKCore

a. Supported Python version: 3.6/3.8/3.10 on Windows/Linux

b. Python dependency

- pip install -r requirements.txt

c. Place the calibration file such as "D2152E040-28_28GHz.csv" into "files/"

d. Please refer to public enums from "lib/TMYPublic.py"

- RFMode for switch TX/RX mode
- RetCode to see what happened while processing example code(main.py).
- All data structure of return function

e. All return values follow the below dict struct:

- RetCode: Enumeration list in TMYPublic.py
- RetMsg: String type, shows message if not RetCode.OK
- RetData: Returns data in function if RetCode.OK

2. Document Changelist

Date	Doc Version	Change list	Changed by
2022/3/24	v0.0.1	First edition	Alin Su
2022/4/1	v0.0.2	Add ● getBeamGainList ● getBeamPhaseList ● UD series	Alin Su
2022/4/20	v0.0.3	Add ● getFastParallelMode ● setFastParallelMode ● getBeamIdStorage ● getBeamPattern	Alin Su

		● setBeamPattern	
2022/4/22	v0.0.4	Change return type	Alin Su
2022/4/27	v0.0.5	BBox series support ● queryHWVer ● queryLoaderVer Support BBoard series: ● getTemperatureADC ● setTCConfig ● setChannelPhaseStep ● setChannelGainStep ● setComGainStep Add desscription: ● selectAAKit	Alin Su
2022/5/6	v0.0.6	Add ● getUDDeltaFreq Change ● getHarmonic	Alin Su
2022/5/12	v0.0.7	TLKCore v0.0.8 Add ● getRecommendLO	Alin Su
2022/6/14	v0.0.8	TLKCore v0.0.9 Add ● exportDevLog Change ● Description of getUDFreq	Alin Su
2022/7/28	v0.0.9	TLKCore v1.0.1 Add ● getDevTypeName Change ● Return type description of queryTCEnable	Alin Su
2022/10/24	v0.0.10	TLKCore v1.0.1.4 Add ● Device type description	Alin Su

		● Beam table supports channel configs ● Add all channel checking function Change ● Fix description of setRFMode ● Fix to the correct func name initDev from init ● Fix to the correct func name DeInitDev from initDev ● Fix to the correct func name reboot from Reboot ● Support getTemperatureADC to BBox 5G series ● Disable support getUDFreq	
2023/1/30	v0.1.0	TLKCore v1.1.0 Add ● Device type description ● Beam table supports channel configs ● Add all channel checking function	Alin Su
2023/6/1	v0.1.1	TLKCore v1.1.1 Add ● Add DevInterface: COMPORT, USB or ALL for scanDevices()	Alin Su
2023/6/26	v0.1.2	TLKCore v1.1.2 Change ● Add common_gain into setIcChannelGain()	Alin Su
2023/6/27	v0.1.3	TLKCore v1.1.3(TBD) Add	Alin Su

		● getUDFreqLimit() ● getUDFreqRange() ● unlockUDFreqRange() ● getRefSource() ● getRefFrequencyList() ● setRefSource() ● getOutputReference() ● setOutputReference() Change ● Support UDM, and add UDMState into getUDState() ● getHarmonic() ● getRecommendLO() ● getUDFreq() ● setUDFreq()	
2023/7/6	v0.1.4	TLKCore v1.1.3(TBD) Add ● getUDFreqLimit() Change ● getRefSource -> getRefConfig() ● Correct description of setRefSource() ● Correct description of getOutputReference() ● Correct description of setOutputReference()	Alin Su
2023/7/26	v0.1.5	TLKCore v1.1.3 Add ● Support Python 3.10 Change ● About TLKCore: some descriptions	Alin Su
2023/9/12	v0.1.6	Change ● Part of ‘About TLKCore’	Alin Su

		● Description of setIcChannelGain ● Description of setUDFreq ● getRecommendLO ● Fix description of queryTCConfig ● Description of setTCConfig	
2023/11/20	v0.1.7	TLKCore v1.2.0 Add ● getScanInfo ● New optional parameters in initDev ● getBoardCount ● getChannelCount Change ● Description of selectAAKit ● Fix description of getDR ● setBeamPattern ● getBeamPattern	Alin Su

Index of TLKCore APIs

[UDM] setOutputReference

TLKCore APIs

[All] scanDevices

Signature

string[] scanDevices(string sn, interface=DevInterface.LAN)

Description

Scan the active devices to obtain its information including SN, IP and device type.

Parameters

Declaration type	Description	Parameters
string	sn	“D2104L001-28”
DevInterface	interface, scanning interface, this enum referenced from lib/TMPublic.py	DevInterface.LAN

Returns

Declaration type	Description	Values
string[]	Each device includes SN, IP and device type	[“SN1, IP1, Device_Type1”, “SN2, IP2, Device_Type2”]

i.e. [“SN1, IP1, Device_Type1”,“SN2, IP2, Device_Type2”,...]

[All] getScanInfo

Signature

dict getScanInfo(string sn=None)

Description

There are 2 usages

1. Return all scan results with dict format if not assign SN(dict)
   a. getScanInfo()
2. Return scan result for specific SN(tuple).
   a. getScanInfo(SN)

Parameters

Declaration type	Description	Parameters
string(Optional)	sn	“D2104L001-28”

Returns

Declaration type	Description	Values

dict	Device (dict) includes SN as key, and (Address and device type) as tuple value.	{
		SN: (Address, device_type)
		}

tuple	(Address and device type) as tuple value.	(Address, device_type)

[All] initDev

Signature

RetCode initDev(string sn, address="", dev_type=-1)

Description

Initialize device mode and settings after scanning.

Or initialize the device via passing address & dev_type without scanning(optional).

P.S. dev_type information provided by previous scanning.

Parameters

Declaration type	Description	Parameters
string	sn	“D2104L001-28”
string(Optional)	address	“192.168.100.111”
int(Optional)	dev_type	9

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[All] DeInitDev

Signature

RetCode DeInitDev(string sn)

Description

De-initialize device instance and remove connection

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[All] queryMAC

Signature

string queryMAC(string sn)

Description

Get Mac address from device with sn

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values

string	Ethernet MAC address	“00:D0:E2:11:22:33”

[All] queryTLKCoreVer

Signature

string queryTLKCoreVer()

Description

Get TLKCore version

Returns

Declaration type	Description	Values
string	TLKCore version	“0.0.1”

[All] queryFWVer

Signature

string queryFWVer(string sn)

Description

Get Device Firmware Version

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2014L001-28"

Returns

Declaration type	Description	Values
string	Device Firmware version	"v1.2.10"

[All] queryHWVer

Signature

string queryHWVer(string sn)

Description

Get BBox/UDBox hardware version.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UD-BS20343100-24"

Returns

Declaration type	Description	Values
string	Hardware version	UDBox: 0013, 0014...etc

[All] setStaticIP

Signature

RetCode setStaticIP(string sn, string ip)

Description

Set device static IPsettings

Parameters

Declaration type	Description	Parameters
string	Static IP settings	“192.168.100.111”
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[BBox Series] checkCaliTableLocation

Signature

bool checkCaliTableLocation(string sn)

Description

Check if the calibration table exists or not with the specific sn.

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values
bool	Calibration table exists or not	True/False

[BBox Series] getFrequencyList

Signature

float[] getFrequencyList(string sn)

Description

Get supported frequency points list from calibration table with the specific sn

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values
float[]	frequency point(float) in list	[26.5,27,27.5,28]

[BBox Series] queryCaliTableVer

Signature

string queryCaliTableVer(string sn)

Description

Get Calibration method version

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2014L001-28"

Returns

Declaration type	Description	Values
string	Calibration Method Version	"2.0.0"

[BBox Series] getOperatingFreq

Signature

float getOperatingFreq(string sn)

Description

Get current frequency point in calibration table with the specific sn

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values
float	Frequecny Point	27.5

[BBox Series] setOperatingFreq

Signature

RetCode setOperatingFreq(string sn, float freq)

Description

Load the calibration table with the specific SN and frequency point.

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
float	Frequency point	27.5

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[BBox Series] getDR

Signature

float[][] getDR(string sn)

float[] getDR(string sn, RFMode mode)

Description

Get current TX/RX dynamic range with the specific sn from calibration table

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
RFMode	TX/RX mode imported from TMYPublic	RFMode.TX

Returns

Declaration type	Description	Values
float[][]	[	[

	[RX_MIN_GAIN, RX_MAX_GAIN], [TX_MIN_GAIN, TX_MAX_GAIN]]	[0, 15], [-2, 9.5]]
float[]	[MIN_GAIN, MAX_GAIN] with mode	

[BBox 5G Series] getCOMDR

Signature

float[][] getCOMDR(string sn)

Description

Get current TX/RX board common-arm dynamic range with the specific sn from calibration table

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values
float[][]	[[TX_COM_MIN_GAIN, TX_COM_MAX_GAIN],	[[0, 4], [-1, 2]

	[RX_COM_MIN_GAIN, RX_COM_MAX_GAIN]]	]

[BBox 5G Series] getELEDR

Signature

float[][] getELEDR(string sn)

Description

Get current TX/RX board element-arm dynamic range with the specific sn from calibration table

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
float[][]	[[TX_ELEMENT_GAIN, RX_ELEMENT_GAIN]]	[[3, 2]]

[BBox Series] getAAKitList

Signature

string[] getAAKitList(string sn)

Description

Get AAKit name string array from aakit table

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
string[]	AAKIT NAME Array	[“TMYTEK_C2104”, “TMYTEK_C2105”,]

[BBox Series] getAAKitInfo

Signature

dict getAAKitInfo(string sn)

dict getAAKitInfo(string sn, string kitName)

Description

Get AAKit information by aakit name and sn, default is the selected aakit name

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”
string	AAKit Name	“TMYTEK_C2104”

Returns

Declaration type	Description	Values
dict	dict contains keys: ● “DEV_TYPE” ● “TYPE”, ● “SPACING”, ● “STEERING_H”, ● “STEERING_V”, ● “OFFSET_TX”, ● “OFFSET_RX”,	

[BBox Series] setAAKitInfo

Signature

```
RetCode setAAKitInfo(strin sn,
                     string kitName,
                     spacing:list,
                            steeringH:list, steeringV:list,
                            offsetTx:list, offsetRx:list)
```

Description

Set **customized** AAKit information with its parameters

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
string	AAKit Name	"TMYTEK_C2104"
float[]	Spacing	[5,5]
float[]	Steering_H	[-45,45]
float[]	Steering_V	[0,0]
int[]	TX offset	[0,0,0,0]
int[]	RX offset	[0,0,0,0]

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[BBox Series] deleteAAKitInfo

Signature

RetCode deleteAAKitInfo(string sn, string kitName)

Description

Delete **customized** AAKit information by aakitname and sn

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
string	AAKit Name	"TMYTEK_C2104"

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[BBox Series] saveAAKitFile

Signature

RetCode saveAAKitFile(string sn, string kitName)

Description

Save **customized** AAkit file with AAKit name

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
string	AAKit Name	"TMYTEK_C2104"

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[BBox Series] selectAAKit

Signature

RetCode selectAAKit(string sn, string AAKitName)

Description

Set operating AAKit by AAKitName for device with SN, please call getAAKitList() to pre-fetch available AAKit list first.

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
string	AAKitName	"TMYTEK_C2104" to select or "" to deselect

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[BBox Series] getRFMode

Signature

RFMode getRFMode(string sn)

Description

Get Device Operating Mode

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
RFMode	Imported from TMYPublic	RFMode.TX, RFMode.RX

[BBox Series] setRFMode

Signature

retCode setRFMode(string sn, RFMode mode)

Description

Set Device operating mode

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”
RFMode	Imported from TMYPublic	RFMode.TX, RFMode.RX

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[BBox 5G Series] getFastParallelMode

Signature

bool getFastParallelMode(string sn)

Description

Get fast parallel mode and external SPI status

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
bool	Fast parallel mode and external SPI enable/disable	True/False

[BBox 5G Series] setFastParallelMode

Signature

RetCode setFastParallelMode(string sn, bool en)

Description

Set fast parallel mode and external SPI enable or not

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”
bool	Enable	True , False

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[BBox 5G Series] getBoardCount

Signature

int getBoardCount(string sn)

Description

Get total number of RF board

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
int	The number of RF board	1

[BBox 5G Series] getChannelCount

Signature

int getChannelCount(string sn, inBoard=False)

Description

Get total number of channel in this device (or in one RF board)

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2230E013-28"
bool (Optional)	Get count in one board	True

Returns

Declaration type	Description	Values
int	The total channel count of device Or The total channel count of RF board	16 or 4

[BBox Series] switchChannel

Signature

string switchChannel(string sn, int channel, bool disable)

Description

Disable the specific channel power with device sn or not

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
int	Channel number	1 (start value)
bool	disable or not	True(Disable),

		False(Enable)

Returns

Declaration type	Param Description	Values
RetCode	OK, ERROR, …etc	

[BBox Series] getChannelSwitch

Signature

int[][] getChannelSwitch(string sn, RFMode mode)

Description

Read all channel disable/enable settings by the specific mode

Note: RFMode mode will be deprecated

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
RFMode	TX/RX Mode	

Returns

Declaration type	Description	Values
int[][]	Disable settings for all channel	[[0,0,0,1]

		]

[BBox Series] setChannelGainPhase

Signature

string setChannelGainPhase(string sn, int channel, float db, int deg)

Description

Set Gain and Phase setting in the specific channel

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”
int	Channel number	1 (start value)
float	Gain db	in Dynamic range
int	Phase deg	0 - 360

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[BBox Series] setIcChannelGain

Signature

RetCode setIcChannelGain(string sn, int board, float[] ch_gain, float common_gain=None)

Description

Set four channel Gain settings in the specific board, and ch_gain will be offset gain only if common_gain is not None

Parameters

Declaration type	**Description**	**Parameters**
string	Device SN	“D2104L001-28”
int	Board number	1 (start value)
float[]	a list for all channel gain settings	[3.5, 2, 1.5, 2]
float(Optional)	common_gain	2.5

Returns

Declaration type	**Description**	**Values**
RetCode	OK, ERROR, …etc	

[BBox Series] setBeamAngle

Signature

RetCode setBeamAngle(string sn, float db, int theta, int phi, RFMode mode=None)

Description

Set the specific beam angle with (db, theta, phi) for the device with sn, it follows the current RFMode, or assigns the RFMode directly, and it must selectAAKit() first.

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”
float	Gain Settings	Dynamic range
int	Theta	AAkit spacing
int	Phi	0 - 359
RFMode(Optional)	mode	RFMode.TX/RX

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[BBox Series] getBeamGainList

Signature

float[][] getBeamGainList(string sn)

Description

Read all channel gain settings with current mode

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
float[][]	All channel gain settings	[[3.5,4,3.5,4]]

[BBox Series] getBeamPhaseList

Signature

int[][] getBeamPhaseList(string sn)

Description

Read all channel phase settings with current mode

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
int[,]	All channel phase settings	[[45,60,75.90]]

[BBox Series] queryStaticIP

Signature

string queryStaticIP(string sn)

Description

Get device static IP settings

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
string	Static IP address	“192.168.100.111”

[BBox Series] reboot

Signature

RetCode reboot(string sn)

Description

Reboot device

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[BBox Series] queryTCEnable

Signature

void queryTCEnable(string sn)

Description

Get dynamic temperature compensation status

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values
int	Auto temperature compensation enabled or not.	0: Disable 1: Enable

[BBox Series] queryTCConfig

Signature

```
void queryTCConfig(string sn)
```

Description

Get dynamic temperature compensation status

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values
int[][][] as list	TC config with list format, each board includes [TXC,TXQ,RXC,RXQ]	[[[8, 6, 2, 11]]]

[BBox 5G Series] getBeamIdStorage

Signature

int getBeamIdStorage(string sn)

Description

Get BeamId storage limit for each RF mode

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

Returns

Declaration type	Description	Values
int	Beam storage limit	64 or 512...etc

[BBox 5G Series] getBeamPattern

Signature

string getBeamPattern(string sn, RFMode mode, int beamId)

Description

Get beam pattern config by beamId, default(if empty) is channel config,

Parameters

Declaration type	**Description**	**Parameters**
string	Device SN	“D2104L001-28”
RFMode	mode	RFMode.TX/RX
int	beamId: 1 ~ limit of beamId storage, ref. getBeamIdStorage()	1

Returns

Declaration type	**Description**	**Values**

dict	BeamPattern in dict format	{
		'beam_description': '',
		'beam_type': 1,
		'channel_config':
		{
		'board_1':
		{
		'common_db': 10.0,
		'channel_1': {'sw': 0, 'db': 3.5, 'deg': 0},
		'channel_2': {'sw': 0, 'db': 3.5, 'deg': 0},
		'channel_3': {'sw': 0, 'db': 3.5, 'deg': 0},
		'channel_4': {'sw': 0, 'db': 3.5, 'deg': 0}
		}
		}
		}

[BBox 5G Series] setBeamPattern

Signature

RetCode setBeamPattern(string sn, RFMode mode, int beamId, BeamType beamType, dict config, string description=“”)

Description

Set BBox beam pattern with mode, beamId, beamType, and configuration. The configuration which overrides the default configuration, so we can just write down the partial config which we want to modify.

For batch beam edition, you can edit the beam configuration file for main.py by enabling batch_import variable, about how to edit file, please reference [FBS] Beam configuration file.

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”
RFMode	mode	RFMode.TX/RX
int	beamId: 1 ~ limit of beamId storage, ref. getBeamIdStorage()	1
BeamType	beamType	BeamType.BEAM: whole beam(0, must selectAAKit() first), or BeamType.CHANNEL: configure each channels(1)
dict	config, choose Beam config or Channel config with dict format,	{ ‘db’: 6,

	ref: Beam/Channel config	‘theta’: 0, ‘phi’: 0 }
string(Optional)	description	“MyConfig-1”

Beam config

```
        Default
        {
                ‘db’: 6, # if Max DR is 6
                ‘theta’: 0,
                ‘phi’: 0
        }
```

Declaration type	Description	Parameters
float	db(In dynamic range)	6
int	theta (0-45), Max steering theta reference from AAKit table.	0
int	phi (0~359)	0

Channel config

```
        Default (Only ONE board if BBoxLite, ref. getBoardCount())
        {
                ‘board_1’:
                {
                        ‘common_db’: 6,
                        ‘channel_1’:
                        {
                                ‘"sw": 0,
                                  "db": 4.0, # if Max element DR is 4.0
                                  "deg": 0
                        },
                        …
                        ‘channel_4’:
```

```
        {
           ‘"sw": 0,
             "db": 4.0,
             "deg": 0
        }
    }
}
```

Declaration type	Description	Parameters
float	common_db (In common gain dynamic range)	6
int	sw, it means disable or not (0:ON, 1:OFF)	0
float	db (In element dynamic range)	4.0
int	deg (0~355, step: 5)	0

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[BBox 5G Series] getTemperatureADC

Signature

int[] getTemperatureADC(string sn)

Description

Get temperature adc value per board from device

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28”

Returns

Declaration type	Description	Values
int[]	Board ADC	0 - 50

[BBoard Series] setTCConfig

Signature

RetCode setTCConfig(string sn, int[] TC_Config)

Description

Set temperature compensation parameters if dynamic TC disabled

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
int[] as list	TC_Config: ● TX_C ● TX_Q ● RX_C ● RX_Q	range: 0 - 31

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[BBoard Series] setChannelGainStep

Signature

RetCode setChannelGainStep(string sn, int channel, int gain_step,)

Description

Set the specific channel gain step

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"

int	Channel number	1 (start number)
int	Gain Step	0 - 15

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[BBoard Series] setComGainStep

Signature

string sn setComGainStep(string sn, int board, int gain_step)

Description

Set the specific board common-arm gain with the common-arm gain step

Parameters

Declaration type	Description	Parameters
string	Device SN	“D2104L001-28“
int	Com gain Step	0 - 15

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	

[BBoard Series] setChannelPhaseStep

Signature

RetCode setChannelPhaseStep(string sn, int channel, int phase_step)

Description

Set the specific channel phase step

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
int	Channel number	1 (start number)
int	Phase Step	0 - 63

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[UDBox] getUDDeltaFreq

Signature

int getUDDeltaFreq(string sn, int freq_ud, int freq_rf, int freq_if, int bandwidth)

Description

Get the delta frequency between LO & IF from [SN].csv, please use wide range frequency for better performance in the **loopback** condition. This function will be deprecated after UD 00E/00F phased out.

Parameters

Declaration type	Description	Parameters
string	Device SN	“UD-BS20343100-24”

Returns

Declaration type	Description	Values
int	Delta freq	

[UD Series] getHarmonic

Signature

bool getHarmonic(string sn, int freq_ud, int freq_rf, int freq_if, int bandwidth)

Description

Check UDBox Frequency affected by harmonic (KHz), not check equation

Parameters

Declaration type	Description	Parameters
string	Device SN	"UD-BS20343100-24"

Returns

Declaration type	Description	Values
bool	● True: harmonic ● False: not harmonic	

[UD Series] getRecommendLO

Signature

dict getRecommendLO(string sn, int freq_rf, int freq_if, int bandwidth)

Description

Get supported and recommend LO frequencies from UD RF & IF frequency.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UD-BS20343100-24"
int	UD RF frequency (KHz)	
int	UD IF frequency (KHz)	

Returns

Declaration type	Description	Values
dict	dict contains keys: ● "USBLo" ● "LSBLo", ● "Recommend",	dict contains values from keys

[UD Series] setUDFreq

Signature

RetCode setUDFreq(string sn, int freq_ud, int freq_rf, int freq_if, int bandwidth)

Description

Set UD Frequency and also call getHarmonic() for checking.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UD-BS20343100-24"
int	UD LO frequency (KHz)	TBD
int	UD RF frequency (KHz)	TBD
int	UD IF frequency (KHz)	TBD
int	UD bandwidth (KHz)	TBD

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	

[UDM] getUDFreq

Signature

dict getUDFreq(string sn)

Description

Get UD Frequency with KHz.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UD-BS20343100-24"

Returns

Declaration type	Description	Values
dict	dict contains keys: ● "UDFreq" ● "RFFreq":0, ● "IFFreq":0,	Default UDFreq is 6G = 6000000 KHz

[UD Series] getUDState

Signature

dict getUDState(string sn)

int getUDState(string sn, UDState.(Your item))

dict getUDState(string sn, UDMState(Your item 1)|UDMState(Your item 2)...)

Description

UD Box/Module current status. Please reference TMYPublic.py for more detailed information.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UD-BS20343100-24"

Return Data

UDState

Name	Description	Values
PLO_LOCK	Lock status	0 - Unlock, 1 - Lock
CH1	CH1 enable	0 - Disable, 1 - Enable
CH2	CH2 enable	0 - Disable, 1 - Enable
OUT_10M	10MHz output	0 - Disable, 1 - Enable
OUT_100M	100MHz output	0 - Disable, 1 - Enable
SOURCE_100M	100MHz source	0 - Internal, 1 - External
LED_100M	100MHz Led status	0 - Off, 1 - White, 2 -

		Blue
PWR_5V	5V output	0 - Disable, 1 - Enable
PWR_9V	9V output	0 - Disable, 1 - Enable

UDMState

Name	**Description**	**Values**
SYSTEM	System status	UDM_SYS 0 - Normal -1 - Error
PLO_LOCK	PLO lock status	UDM_SYS 0 - Lock -1 - Unlock
REF_LOCK	Reference clock lock status	UDM_REF 0 - Internal Locked 1 - External Locked -1 - Unlock
LICENSE	License unlock state	UDM_LICENSE -2 - Verified failed at flash -1 - Verified failed at digest 0 - Non license 1 - License verified pass

[UD] setUDState

Signature

dict setUDState(string sn, dict state)

dict setUDState(string sn, int state, UDState.(Your item))

Description

Set UDBox status.

Parameters

Name	**Description**	**Parameters**
string	Device SN	"UD-BS20343100-24"
PLO_LOCK	Lock status	1
CH1	CH1 enable	0 - Disable, 1 - Enable
CH2	CH2 enable	0 - Disable, 1 - Enable
OUT_10M	10MHz output	0 - Disable, 1 - Enable
OUT_100M	100MHz output	0 - Disable, 1 - Enable
SOURCE_100M	100MHz source	0 - Internal, 1 - External
LED_100M	Reserved	N/A
PWR_5V	5V output	0 - Disable, 1 - Enable
PWR_9V	9V output	0 - Disable, 1 - Enable

Returns

Name	**Description**	**Values**

PLO_LOCK	Lock status	0 - Unlock, 1 - Lock
CH1	CH1 enable	0 - Disable, 1 - Enable
CH2	CH2 enable	0 - Disable, 1 - Enable
OUT_10M	10MHz output	0 - Disable, 1 - Enable
OUT_100M	100MHz output	0 - Disable, 1 - Enable
SOURCE_100M	100MHz source	0 - Internal, 1 - External
LED_100M	100MHz Led status	0 - Off, 1 - White, 2 - Blue
PWR_5V	5V output	0 - Disable, 1 - Enable
PWR_9V	9V output	0 - Disable, 1 - Enable

[UDM] getUDFreqLimit

Signature

dict getUDFreqLimit(string sn)

Description

Get max capability of UDM frequency range with KHz.

Parameters

Declaration type	**Description**	**Parameters**
string	Device SN	“UDM-2322001-0620”

Returns

Declaration type	**Description**	**Values**
dict	dict contains keys & values: ● “UDFreq”: {“min”: ?, “max”: ?},	Default UDFreq is 6G = 6000000 KHz

	● "RFFreq": {"min": ?, "max": ?}, ● "IFFreq":{"min": ?, "max": ?},	

[UDM] getUDFreqRange

Signature

dict getUDFreqRange(string sn)

Description

Get the current available frequency range with KHz from UDM.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UDM-2322001-0620"

Returns

Declaration type	Description	Values
dict	dict contains keys & values: ● "UDFreq": {"min": ?, "max": ?}, ● "RFFreq": {"min": ?, "max": ?}, ● "IFFreq":{"min": ?, "max": ?},	Default UDFreq is 6G = 6000000 KHz

[UDM] unlockUDFreqRange

Signature

RetCode unlockUDFreqRange(string sn, string unlock_key_str)

Description

Unlock the license key for the new frequency range with KHz to UDM, then must reboot the device to activate the new range.

Parameters

Declaration type	**Description**	**Parameters**
string	Device SN	"UDM-2322001-0620"
string	unlock key	N/A

Returns

Declaration type	**Description**	**Values**
RetCode	OK, ERROR, …etc	

[UDM] getRefConfig

Signature

UDM_REF getRefConfig(string sn)

Description

Get current configuration of reference clock source, to know reference is from internal or external, and its output/input freq. (output not enabled is 0)

Parameters

Declaration type	**Description**	**Parameters**
string	Device SN	"UDM-2322001-0620"

Returns

Declaration type	Description	Values
dict	dict contains keys & values: ● "source": UDM_REF, ● "freq": 0	{"source":UDM_REF.INTERNAL, "freq": 10000}

[UDM] getRefFrequencyList

Signature

list getRefFrequencyList(string sn)

Description

Get the supported reference frequency list.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UDM-2322001-0620"

Returns

Declaration type	Description	Values
list	Support reference frequency list	[10000, 100000]

[UDM] setRefSource

Signature

RetCode setRefSource(string sn, UDM_REF source)

Description

Set reference clock source is from internal reference or external reference. If you set UDM_REF.EXTERNAL but not plug-in signal cable, you will get the reference lock status is UDM_REF.INTERNAL from getUDState().

Parameters

Declaration type	**Description**	**Parameters**
string	Device SN	"UDM-2322001-0620"
UDM_REF	UDM_REF INTERNAL - Internal Locked EXTERNAL - External Locked UNLOCK - Unlock	UDM_REF.INTERNAL
float	External reference frequency, unit is KHz, default is 0 for internal reference.	10000

Returns

Declaration type	**Description**	**Values**
RetCode	OK, ERROR, ...etc	

[UDM] getOutputReference

Signature

bool getOutputReference(string sn)

Description

Get output status of reference clock source from internal reference and always get False if from external reference.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UDM-2322001-0620"

Returns

Declaration type	Description	Values
bool	Output reference enabled or not	False

[UDM] setOutputReference

Signature

RetCode setOutputReference(string sn, bool output, float ref_freq=0)

Description

Set output of reference clock source and reference frequency from internal reference.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UDM-2322001-0620"
bool	Output reference enabled or not	True

float	reference frequency(clock) with KHz	10000

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, …etc	