

Prerequisite

About TLKCore

1. Please reference: <https://github.com/tmytek/tlkcore-examples>
2. Supported Python version: 3.6/3.8/3.10/3.12 on Windows/Linux platform.
3. Python dependency

Unset

```
pip install -r requirements.txt
```

4. Place the calibration file such as "D2152E040-28_28GHz.csv" into "files/"
5. Please refer to public enums from "tlkcore.TMYPublic.py"
 - RFMode/CellRFMode for switch TX/RX mode
 - RetCode to see what happened while processing example code(main.py).
 - All data structure of return function
6. 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

Change History

Date	Doc Version	Change List	Writer
2022/3/24	v0.0.1	First edition	Alin Su
2022/4/1	v0.0.2	Add <ul style="list-style-type: none"> • getBeamGainList • getBeamPhaseList • UD series 	Alin Su
2022/4/20	v0.0.3	Add <ul style="list-style-type: none"> • getFastParallelMode • setFastParallelMode • getBeamIdStorage • getBeamPattern • setBeamPattern 	Alin Su
2022/4/22	v0.0.4	Change return type	Alin Su
2022/4/27	v0.0.5	BBox series support <ul style="list-style-type: none"> • queryHWVer • queryLoaderVer Support BBoard series: <ul style="list-style-type: none"> • getTemperatureADC 	Alin Su

		<ul style="list-style-type: none"> • setTCConfig • setChannelPhaseStep • setChannelGainStep • setComGainStep Add description: <ul style="list-style-type: none"> • selectAAKit 	
2022/5/6	v0.0.6	Add <ul style="list-style-type: none"> • getUDDeltaFreq Change <ul style="list-style-type: none"> • getHarmonic 	Alin Su
2022/5/12	v0.0.7	TLKCore v0.0.8 Add <ul style="list-style-type: none"> • getRecommendLO 	Alin Su
2022/6/14	v0.0.8	TLKCore v0.0.9 Add <ul style="list-style-type: none"> • exportDevLog Change <ul style="list-style-type: none"> • Description of getUDFreq 	Alin Su
2022/7/28	v0.0.9	TLKCore v1.0.1 Add <ul style="list-style-type: none"> • getDevTypeName Change <ul style="list-style-type: none"> • Return type description of queryTCEnable 	Alin Su
2022/10/24	v0.0.10	TLKCore v1.0.1.4 Add <ul style="list-style-type: none"> • Device type description • Beam table supports channel configs • Add all channel checking function Change <ul style="list-style-type: none"> • Fix description of setRFMode • Fix to the correct func name initDev from init • Fix to the correct func name DelnitDev from initDev • Fix to the correct func name reboot from Reboot • Support getTemperatureADC to BBox 5G series • Disable support getUDFreq 	Alin Su
2023/1/30	v0.1.0	TLKCore v1.1.0 Add <ul style="list-style-type: none"> • 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	Alin Su

		<ul style="list-style-type: none"> Add DevInterface: COMPORT, USB or ALL for scanDevices() 	
2023/6/26	v0.1.2	TLKCore v1.1.2 Change <ul style="list-style-type: none"> Add common_gain into setIcChannelGain() 	Alin Su
2023/6/27	v0.1.3	TLKCore v1.1.3(TBD) Add <ul style="list-style-type: none"> getUDFreqLimit() getUDFreqRange() unlockUDFreqRange() getRefSource() getRefFrequencyList() setRefSource() getOutputReference() setOutputReference() Change <ul style="list-style-type: none"> Support UDM, and add UDMState into getUDState() getHarmonic() getRecommendLO() getUDFreq() setUDFreq() 	Alin Su
2023/7/6	v0.1.4	TLKCore v1.1.3(TBD) Add <ul style="list-style-type: none"> getUDFreqLimit() Change <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> Support Python 3.10 Change <ul style="list-style-type: none"> About TLKCore: some descriptions 	Alin Su
2023/9/12	v0.1.6	Change <ul style="list-style-type: none"> Part of 'About TLKCore' Description of setIcChannelGain Description of setUDFreq getRecommendLO Fix description of queryTCConfig Description of setTCConfig 	Alin Su

2023/11/20	v0.1.7	TLKCore v1.2.0 Add <ul style="list-style-type: none"> • getScanInfo • New optional parameters in initDev • getBoardCount • getChannelCount Change <ul style="list-style-type: none"> • Description of selectAAKit • Fix description of getDR • setBeamPattern • getBeamPattern 	Alin Su
2024/1/16	v0.1.8	TLKCore v1.2.0.7 Add <ul style="list-style-type: none"> • Description of PD • Description of reboot • Description of processDFU • getUDFreq for UDM->UD series Change <ul style="list-style-type: none"> • Descriptions: KHZ->kHz • Description of scanDevices • Description of setStaticIP • Description of setBeamPattern 	Alin Su
2025/1/13	v0.2.0	TLKCore v2.1.0 Add <ul style="list-style-type: none"> • getRFMode • getOperatingConfig • getOperatingStatus • switchChannel for multi polarizations • setIcComGain Change <ul style="list-style-type: none"> • Follow TMYTEK document format • Part of 'About TLKCore' • Description of setChannelGainPhase • Description of setBeamAngle • Description of setBeamPattern 	Alin Su
2025/2/5	v0.2.1	TLKCore v2.2.0 RIS part Add <ul style="list-style-type: none"> • setRISModuleInfo • setRISPattern • getRISPattern • setRISAngle • getNetInfo • setIPMode • setSubnetMsk • setGateway 	Cheryl Lin

2025/2/20	v0.2.2	<p>TLKCore v2.2.0 UDB-0630 part</p> <p>Add</p> <ul style="list-style-type: none">• unlockUDFreqRange for UDB-0630• querySN• getLOConfig• setLOConfig• processDFU for RIS/UDB <p>Change</p> <ul style="list-style-type: none">• getHarmonic: removed ``freq_rf`` parameter• Description of setUDState	Cheryl Lin
-----------	--------	---	---------------

TLKCore API List

[All] scanDevices

Signature

```
list 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 tlkcore.TMPublic.py	DevInterface.LAN

Returns

Declaration type	Description	Values
list	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

- Return all scan results with dict format if not assign SN(dict)
 - getScanInfo()
- Return scan result for specific SN(tuple).
 - 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	Device SN	"D2104L001-28"
string(Optional)	Device Address	"192.168.100.111"
int(Optional)	Device Type	9

Returns

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

[All] DelnitDev

Signature

```
RetCode DelnitDev(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(string sn)
```

Description

Get TLKCore version

Parameters

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

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] querySN

Signature

```
RetCode querySN(string sn )
```

Description

Query SN for device

Parameters

Declaration type	Description	Parameters
string	Device SN	

Returns

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

[All] 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	

[All] setStaticIP

Signature

```
RetCode setStaticIP(string sn, string ip, bool reboot=False)
```

Description

Set static IP settings device for Ethernet interface, then auto reboot or not (Default not reboot).

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
string	Static IP settings	"192.168.100.111"
bool	Auto reboot after set IP	True/False

Returns

Declaration type	Description	Values
------------------	-------------	--------

RetCode	OK, ERROR, ...etc	
---------	-------------------	--

[BeamForm Series][UDB][RIS] processDFU

Signature

```
RetCode processDFU(string sn, string file_path)
```

Description

Device firmware upgrade for BBox 5G series.

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
string	FW image file path	"BBoxOne_v1.2.16.bin"

Returns

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

[BeamForm 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"
--------	-------------------	-------------------

[BeamForm Series] queryTCEnable

Signature

```
int 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

[BeamForm Series] queryTCConfig

Signature

```
list 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 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"

[BeamForm 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	Frequency Point	27.5

[BeamForm 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

```
list getDR(string sn, RFMode mode=None)
```

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, default is None to return all DR list	RFMode.TX
--------	---	-----------

Returns

Declaration type	Description	Values
list	Nested array with T/R, (H/V), Min/Max sequence. i.g. [[TX_MIN_GAIN, TX_MAX_GAIN], [RX_MIN_GAIN, RX_MAX_GAIN]] or [MIN_GAIN, MAX_GAIN] if assign mode	[[0, 15], [-2, 9.5]]

[BBox Series] getCOMDR

Signature

list 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
list	Nested array with T/R, (H/V), Min/Max sequence. i.g. [[TX_COM_MIN_GAIN, TX_COM_MAX_GAIN],]]	[[0, 4], [-1, 2]]

	[RX_COM_MIN_GAIN, RX_COM_MAX_GAIN]]	
--	--	--

[BBox Series] getELEDR

Signature

```
list 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
list	[[TX_ELEMENT_GAIN, RX_ELEMENT_GAIN]]	[[3, 2]]

[BBox Series] getAAKitList

Signature

```
list getAAKitList(string sn)
```

Description

Get AAKit name string list from aakit table

Parameters

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

Returns

Declaration type	Description	Values
list	AAKIT NAME Array	["TMYTEK_C2104", "TMYTEK_C2105",]

[BBox Series] getAAKitInfo**Signature**

```
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	Selected AAKit Name	"TMYTEK_C2104"

Returns

Declaration type	Description	Values
dict	dict contains keys: <ul style="list-style-type: none"> • "DEV_TYPE" • "TYPE", • "SPACING", • "STEERING_H", • "STEERING_V", • "OFFSET_TX", • "OFFSET_RX", 	{ "DEV_TYPE": 9, "TYPE": 1, "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	

[BeamForm Series] getRFMode**Signature**

```
RFMode getRFMode(string sn)
CellRFMode getRFMode(string sn)
```

Description

Get Device Operating Mode

Parameters

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

Returns

Declaration type	Description	Values
RFMode/CellRFMode	TX/RX mode imported from TMYPublic, and CellRFMode add Standby mode.	RFMode.TX or CellRFMode.TX

[BeamForm Series] setRFMode**Signature**

```
RetCode setRFMode(string sn, Union[RFMode, CellRFMode, int] mode)
```

Description

Set Device operating mode

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
RFMode/CellRFMode	TX/RX mode imported from TMYPublic, and CellRFMode add Standby mode.	RFMode.TX or CellRFMode.TX

Returns

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

[BBox 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 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	

[BeamForm 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

[BeamForm Series] getChannelCount

Signature

```
int getChannelCount(string sn, inBoard=False)
```

Description

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

Parameters

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

```
RetCode 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]]

[BeamForm Series] setChannelGainPhase**Signature**

```
RetCode setChannelGainPhase(string sn, int channel, float gain_db, int phase_deg,  
Union[POLARIZATION, int] polar=POLARIZATION.DUAL)
```

Description

Set Gain and Phase setting for the specific channel or ALL channels, board/channel usually starts from 1 (0 means ALL channels), gain_db range is from DR, phase_deg range is 0-359

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
int	Channel number	0 (ALL channels) 1 (start value)
float	gain_db	in Dynamic range
int	phase_deg	0 - 359
POLARIZATION(Optional)	polarization	POLARIZATION.HORIZON

Returns

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

[BeamForm Series] setlcChannelGain

Signature

RetCode setlcChannelGain(string sn, int board, float[] ch_gain, float common_gain=None, Union[POLARIZATION, int] polar=POLARIZATION.DUAL)

Description

Set all 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
POLARIZATION(Optional)	polarization	POLARIZATION.HORIZON

Returns

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

[BeamForm Series] setBeamAngle

Signature

```
RetCode setBeamAngle(string sn, float gain_db, int theta, int phi, Union[POLARIZATION, int]
polar=POLARIZATION.DUAL, 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	0 - 60
int	Phi	0 - 359
POLARIZATION(Optional)	polarization	POLARIZATION.HORIZON
RFMode(Optional)	mode	RFMode.TX/RX

Returns

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

[BeamForm Series] getBeamGainList

Signature

```
float[][] getBeamGainList(string sn, Union[POLARIZATION, int] polar=POLARIZATION.DUAL)
```

Description

Read all channel gain settings with current mode

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
POLARIZATION(Optional)	polarization	POLARIZATION.HORIZON

Returns

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

[BeamForm Series] getBeamPhaseList

Signature

```
int[][] getBeamPhaseList(string sn, Union[POLARIZATION, int] polar=POLARIZATION.DUAL)
```

Description

Read all channel phase settings with current mode

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
POLARIZATION(Optional)	polarization	POLARIZATION.HORIZON

Returns

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

[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 Series] getBeamPattern**Signature**

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

Description

[FBS] 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	<pre>{ '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} } } }</pre>

[BBox Series] setBeamPattern

Signature

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

Description

Set device 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/int	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, ref: Beam/Channel config	{ 'db': 6, 'theta': 0, 'phi': 0 }
string(Optional)	description	"MyConfig-1"

The following configs are Beam config & Channel config with dict format,

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 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[]	The ADC list of Temperature	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: <ul style="list-style-type: none"> • 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	

[CloverCellEVB] getOperatingConfig

Signature

```
dict getOperatingConfig(string sn, CellRFMode mode)
```

Description

Get current operating register config

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
CellRFMode	mode	CellRFMode.TX/RX

Returns

Declaration type	Param Description	Values
dict	last IC operating config in dict format	{'HORIZON': {'DISABLE': [[[0, 0, 0, 0]]], 'PHASE': [[[0, 0, 0, 0]]], 'GAIN': [[[0, 0, 0, 0]]], 'COM_STEP': [[0]]}, 'VERTICAL': {'DISABLE': [[[0, 0, 0, 0]]], 'PHASE': [[[0, 0, 0, 0]]], 'GAIN': [[[0, 0, 0, 0]]], 'COM_STEP': [[0]]}}

[CloverCellEVB] getOperatingStatus

Signature

```
dict getOperatingStatus(string sn)
```

Description

Get IC operating Status

Parameters

Declaration type	Description	Parameters
------------------	-------------	------------

string	Device SN	"D2104L001-28"
--------	-----------	----------------

Returns

Declaration type	Description	Values
dict	IC operating Status in dict format	{'pdet': {'horizon': [0, 0, 0, 0], 'vertical': [0, 0, 0, 0]}, 'temperature': 25.64, 'power': {'vddPa': {'V': 1776, 'I': 1200, 'P': 2131}, 'vdd1v8': {'V': 1800, 'I': 388, 'P': 698}}}

[CloverCellEVB] switchChannel**Signature**

```
RetCode switchChannel(string sn, int channel, bool disable, Union[POLARIZATION, int]
polar=POLARIZATION.DUAL)
```

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)
POLARIZATION(Optional)	polarization	POLARIZATION.HORIZON

Returns

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

[CloverCellEVB] setIcComGain**Signature**

```
RetCode setIcComGain(string sn, POLARIZATION polar, int board, float gain)
```

Description

Set common gain for specific polarization and board/IC

Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
POLARIZATION(Optional)	polarization	POLARIZATION.HORIZON
int	board number	1 (start value)
float	Gain	

Returns

Declaration type	Description	Values
RetCode	OK, ERROR, ...etc	0

[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	<ul style="list-style-type: none">True: harmonicFalse: 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: <ul style="list-style-type: none">• "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)	

int	UD RF frequency (kHz)	
int	UD IF frequency (kHz)	
int	UD bandwidth (kHz)	

Returns

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

[UD Series] getUDFreq**Signature**

```
dict getUDFreq(string sn)
```

Description

Get UD Frequency with kHz, and UD5G only support this function start with FW version 3.0.0

Parameters

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

Returns

Declaration type	Description	Values
dict	dict contains keys: <ul style="list-style-type: none"> "UDFreq": 6000000, "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"

Returns

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
--	--	---------------------------

[UDBox] setUDState

Signature

```
dict setUDState(string sn, item:Union[UDState, int]=UDState.NO_SET)
```

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

[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	

[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: <ul style="list-style-type: none"> "UDFreq": {"min": ?, "max": ?}, "RFFreq": {"min": ?, "max": ?}, "IFFreq": {"min": ?, "max": ?}, 	Default UDFreq is 6G = 6000000 kHz

[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: <ul style="list-style-type: none"> "UDFreq": {"min": ?, "max": ?}, "RFFreq": {"min": ?, "max": ?}, "IFFreq": {"min": ?, "max": ?}, 	Default UDFreq is 6G = 6000000 kHz

[UDM / UDB] unlockUDFreqRange**Signature**

```
RetCode unlockUDFreqRange(string sn, string unlock_key_str)
```

Description

Unlock the license key for the new frequency range with kHz to UDM/UDB, 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: <ul style="list-style-type: none">"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	

[UDB] querySN**Signature**

```
dict querySN(string sn, sn_type: Union[UD_SN_TYPE, int] = UD_SN_TYPE.UD_BOX )
```

Description

Query SN for UDB or UDM inside

Parameters

Declaration type	Description	Parameters
string	Device SN	
UD_SN_TYPE	sn_type	1: UD_BOX 2: UD_MODULE 3: UD_BOX & UD_MODULE

Returns

Declaration type	Description	Values
dict	Device SN	{ 'UD_BOX': 'UDB-2445001-0630'}

[UDB] getLOConfig**Signature**

```
RetCode getLOConfig(string sn, bool returnIntValue)
```

Description

Query SN for UDB

Parameters

Declaration type	Description	Parameters
string	Device SN	
bool	returnIntValue	False

Returns

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

[UDB] setLOConfig**Signature**

```
RetCode setLOConfig(string sn, config:Union[UD_LO_CONFIG, int])
```

Description

LO config to internal/out or external

Parameters

Declaration type	Description	Parameters
string	Device SN	
UD_LO_CONFIG	LO config	0: Internal LO 1: Internal LO and output 2: external LO input

Returns

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

[PD] setCaliConfig**Signature**

```
RetCode setCaliConfig(string sn, dict config)
```

Description

Set PD calibration config for specific frequency config with dict format.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UDM-2322001-0620"
dict	Calibration config for specific frequency.	Calibration config with dict format: { freq(str) : freq_config(dict) }, freq_config also dict type: { "lowPower": dBm(float), "lowVolt": mV(float), "highPower": dBm(float), "highVolt": mV(float) } Example: "20GHz": { "lowPower": -36, "lowVolt": 40.46, "highPower": -5, "highVolt": 936.36 } or reference to main.py

Returns

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

[PD] getVoltageValue

Signature

float getVoltageValue(string sn, float freq)

Description

Get measured voltage(mV) from the power detector.

Parameters

Declaration type	Description	Parameters
------------------	-------------	------------

string	Device SN	"UDM-2322001-0620"
float	frequency with unit: GHz	28

Returns

Declaration type	Description	Values
float	Voltage with unit: mV	121.5

[PD] getPowerValue**Signature**

```
float getPowerValue(string sn, float freq)
```

Description

Get measured power(dBm) from the power detector.

Parameters

Declaration type	Description	Parameters
string	Device SN	"UDM-2322001-0620"
float	frequency with unit: GHz	28

Returns

Declaration type	Description	Values
float	power with unit: dBm	-11

[RIS] getRISModuleInfo()**Signature**

```
dict getRISModuleInfo(string sn):
```

Description

Get all module's freq(MHz), hw version, fpga version...etc as dict format

Parameters

Declaration type	Description	Parameters
string	Device SN	"RIS-2400001-0000"

Returns

Declaration type	Description	Values
dict	Module information	<pre> info = { "1": { "freq": 28000, # MHz, uint32 "hw_ver": 0x00, # uint8 "fpga_ver": [0x00, 0x00, 0x00, 0x00], # uint8 * 4 'element_spacing': 5000, # um(micro meter), uint32 'antenna_size': [32, 32], # module row, col, for pattern generator 'driver_size': [32, 32] # size for scaling }, #... } </pre>

[RIS] setRISPattern()**Signature**

```
RetCode setRISPattern(string sn, pattern:list)
```

Description

Set the pattern of the RIS.

Parameters

Declaration type	Description	Parameters
string	Device SN	"RIS-2400001-0000"
list	Double list pattern (list): list of 0, 1	[[0, 0, 0...], [0, 0, 0...], ...]

Returns

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

[RIS] getRISPattern()

Signature

```
list getRISPattern(string sn, pattern: list )
```

Description

Get the current pattern of the RIS, and pattern format is a double list (0/1)

Parameters

Declaration type	Description	Parameters
string	Device SN	"RIS-2400001-0000"

Returns

Declaration type	Description	Values
list	pattern	[[0, 0, 0...], [0, 0, 0...], ...]

[RIS] setRISAngle()

Signature

```
RetCode setRISAngle(string sn, incident: Union[int, tuple]=(0,0), reflection: Union[int, tuple, list]=[(0,0)], module=[1], feed_distance=None, save=False)
```

Description

Set RIS pattern via Incident angle and Reflection angle, the type of angle can be int or tuple or others, the function will convert to target data type automatically.

Parameters

Declaration type	Description	Parameters
string	Device SN	"RIS-2400001-0000"
Union[int, tuple]	Incident angle. Defaults to (0,0)	theta as [(theta, 0)] or (theta, phi) as [(theta, phi)] or [(theta, phi)]
Union[int, tuple, list]	Reflection angle. Defaults to [(0,0)]	theta as [(theta, 0)] or

		(theta, phi) as [(theta, phi)] or [(theta, phi)]
list	Target module list to control. Defaults to [1]	[1,2,3,4] if modules use 1,2,3,4
float	Feed distance(m). Defaults to None	0.51, None means default feed distance for the frequency
bool	Save to csv or not. Defaults to False	False

Returns

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

[RIS] getNetInfo()**Signature**

```
dict getNetInfo(string sn)
```

Description

Get Network basic info with dict format which includes 'ip_mode', 'ip', 'static_ip', 'MAC', 'subnet_mask' and 'gateway'

Parameters

Declaration type	Description	Parameters
string	Device SN	"RIS-2400001-0000"

Returns

Declaration type	Description	Values
dict	Dictionary containing network information, the value of 'ip_mode' represent as the enum of IPMode for DHCP and STATIC_IP	{'ip_mode': <IPMode.DHCP: 0>, 'ip': '192.168.137.227', 'static_ip': '192.168.100.114', 'MAC': 'f0:e1:d2:c3:b4:a5', 'subnet_mask': '255.255.255.0', 'gateway': '192.168.100.1'}

[RIS] setIPMode()

Signature

```
RetCode setIPMode(string sn, ip_mode: Union[IPMode, int]):
```

Description

Set the IP mode of the network interface

Parameters

Declaration type	Description	Parameters
string	Device SN	"RIS-2400001-0000"
ip_mode	The IP mode to set	0: DHCP, 1: STATIC_IP

Returns

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

[RIS] setSubnetMsk()

Signature

```
RetCode setSubnetMsk(string sn, str mask):
```

Description

Set the subnet mask of the network interface

Parameters

Declaration type	Description	Parameters
string	Device SN	"RIS-2400001-0000"
string	Subnet mask	255.255.255.0

Returns

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

[RIS] setGateway()

Signature

RetCode setGateway(string sn, str gateway):

Description

Set the gateway of the network interface

Parameters

Declaration type	Description	Parameters
string	Device SN	"RIS-2400001-0000"
string	gateway	192.168.100.254

Returns

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