



# **TLKCore Reference Guide**

Version 0.1.1

20230601

# Prerequisite

---

## 1. About TLKCore

- a. Supported Python version: 3.6~3.8
- b. Python dependency
  - `pip install -r requirements.txt`
- c. Place the calibration file such as "D2152E040-28\_28GHz.csv" into "TLKCore/files/"
- d. Please refer to public enums from "TLKCore/lib/TMYPublic.py"
  - RFMode for switch TX/RX mode
  - RetCode to see what happened while process function
  - All data structure of return function
- e. All return values follow the below 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 <ul style="list-style-type: none"><li>● getBeamGainList</li><li>● getBeamPhaseList</li><li>● UD series</li></ul>	Alin Su
2022/4/20	v0.0.3	Add <ul style="list-style-type: none"><li>● getFastParallelMode</li><li>● setFastParallelMode</li><li>● getBeamIdStorage</li><li>● getBeamPattern</li></ul>	Alin Su

		<ul style="list-style-type: none"> <li>● setBeamPattern</li> </ul>	
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"> <li>● queryHWVer</li> <li>● queryLoaderVer</li> </ul> Support BBoard series: <ul style="list-style-type: none"> <li>● getTemperatureADC</li> <li>● setTCConfig</li> <li>● setChannelPhaseStep</li> <li>● setChannelGainStep</li> <li>● setComGainStep</li> </ul> Add description: <ul style="list-style-type: none"> <li>● selectAAKit</li> </ul>	Alin Su
2022/5/6	v0.0.6	Add <ul style="list-style-type: none"> <li>● getUDDeltaFreq</li> </ul> Change <ul style="list-style-type: none"> <li>● getHarmonic</li> </ul>	Alin Su
2022/5/12	v0.0.7	TLKCore v0.0.8 Add <ul style="list-style-type: none"> <li>● getRecommendLO</li> </ul>	Alin Su
2022/6/14	v0.0.8	TLKCore v0.0.9 Add <ul style="list-style-type: none"> <li>● exportDevLog</li> </ul> Change <ul style="list-style-type: none"> <li>● Description of getUDFreq</li> </ul>	Alin Su
2022/7/28	v0.0.9	TLKCore v1.0.1 Add <ul style="list-style-type: none"> <li>● getDevTypeName</li> </ul> Change <ul style="list-style-type: none"> <li>● Return type description of queryTCEnable</li> </ul>	Alin Su
2022/10/24	v0.0.10	TLKCore v1.0.1.4 Add <ul style="list-style-type: none"> <li>● Device type description</li> </ul>	Alin Su

		<ul style="list-style-type: none"> <li>● Beam table supports channel configs</li> <li>● Add all channel checking function</li> </ul> <p>Change</p> <ul style="list-style-type: none"> <li>● Fix description of setRFMode</li> <li>● Fix to the correct func name initDev from init</li> <li>● Fix to the correct func name DelInitDev from initDev</li> <li>● Fix to the correct func name reboot from Reboot</li> <li>● Support getTemperatureADC to BBox 5G series</li> <li>● Disable support getUDFreq</li> </ul>	
2023/1/30	v0.1.0	<p>TLKCore v1.1.0</p> <p>Add</p> <ul style="list-style-type: none"> <li>● Device type description</li> <li>● Beam table supports channel configs</li> <li>● Add all channel checking function</li> </ul>	Alin Su
2023/6/1	v0.1.1	<p>TLKCore v1.1.1</p> <p>Add</p> <ul style="list-style-type: none"> <li>● Add DevInterface: COMPORT, USB or ALL for scanDevices()</li> </ul>	Alin Su

# Index of TLKCore APIs

---

[\[All\] scanDevices](#)

[\[All\] initDev](#)

[\[All\] Delnit](#)

[\[BBox Series\] checkCaliTableLocation](#)

[\[BBox Series\] getFrequencyList](#)

[\[All\] queryMAC](#)

[\[All\] queryTLKCoreVer](#)

[\[All\] queryFWVer](#)

[\[All\] queryHWVer](#)

[\[BBox Series\] queryCaliTableVer](#)

[\[BBox Series\] getOperatingFreq](#)

[\[BBox Series\] setOperatingFreq](#)

[\[BBox Series\] getDR](#)

[\[BBox 5G Series\] getCOMDR](#)

[\[BBox 5G Series\] getELEDR](#)

[\[BBox Series\] getAAKitList](#)

[\[BBox Series\] getAAKitInfo](#)

[\[BBox Series\] setAAKitInfo](#)

[\[BBox Series\] deleteAAKitInfo](#)

[\[BBox Series\] saveAAKitFile](#)

[\[BBox Series\] selectAAKit](#)

[\[BBox Series\] getRFMode](#)

[\[BBox Series\] setRFMode](#)

[\[BBox 5G Series\] getFastParallelMode](#)

[\[BBox 5G Series\] setFastParallelMode](#)

[\[BBox Series\] switchChannel](#)

[\[BBox Series\] getChannelSwitch](#)

[\[BBox Series\] setChannelGainPhase](#)

[\[BBox Series\] setIcChannelGain](#)

[\[BBox Series\] setBeamAngle](#)

[\[BBox Series\] getBeamGainList](#)

[\[BBox Series\] getBeamPhaseList](#)

[\[BBox Series\] queryStaticIP](#)

[\[All\] setStaticIP](#)

[\[BBox Series\] reboot](#)

[\[BBox Series\] queryTCEnable](#)

[\[BBox Series\] queryTCConfig](#)

[\[BBox 5G Series\] getBeamIdStorage](#)

[\[BBox 5G Series\] getBeamPattern](#)

[\[BBox 5G Series\] setBeamPattern](#)

[\[BBoard Series\] getTemperatureADC](#)

[\[BBoard Series\] setTCConfig](#)

[\[BBoard Series\] setChannelGainStep](#)

[\[BBoard Series\] setComGainStep](#)

[\[BBoard Series\] setChannelPhaseStep](#)

[\[All\] queryHWVer](#)

[\[UDBox\] getUDDeltaFreq](#)

[\[UDBox\] getHarmonic](#)

[\[UDBox\] getRecommendLO](#)

[\[UDBox\] setUDFreq](#)

[\[UDBox\] getUDFreq](#)

[\[UDBox\] getUDState](#)

[\[UDBox\] setUDState](#)

# 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] initDev

### Signature

```
RetCode initDev(string sn)
```

### Description

Initialize device mode and device settings

### Parameters

Declaration type	Description	Parameters
string	sn	"D2104L001-28"

### Returns

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

## [All] DelInitDev

### Signature

```
RetCode DelInitDev(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	

## [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]

## [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	"00G"

## [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	Frequencny 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[][]	[  [ TX_MIN_GAIN, TX_MAX_GAIN ], [ RX_MIN_GAIN, RX_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], [RX_COM_MIN_GAIN, RX_COM_MAX_GAIN] ]	[  [0, 4], [-1, 2] ]

## [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, TX_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: <ul style="list-style-type: none"> <li>● "DEV_TYPE"</li> <li>● "TYPE",</li> </ul>	

	<ul style="list-style-type: none"> <li>● "SPACING",</li> <li>● "STEERING_H",</li> <li>● "STEERING_V",</li> <li>● "OFFSET_TX",</li> <li>● "OFFSET_RX",</li> </ul>	
--	--	--

## [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

#### Parameters

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

#### 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 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 setAllChannelGain(string sn, int board, float[] ch\_db)

### Description

Set four channel Gain settings in the specific board

### 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]

### Returns

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

## [BBox Series] setBeamAngle

### Signature

RetCode setBeamAngle(string sn, float db, int theta, int phi)

### Description

Set the specific beam angle with (db, theta, phi) for device with sn

### Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
float	Gain Settings	Dynamic range
int	Theta	AAkit spacing
int	Phi	0 - 359

### 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"

## [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] 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
dict	TC config with dict format	N/A

# [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, int beamId, RFMode mode)
```

## Description

Get Beam Pattern config by BeamId

## Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
RFMode	mode	RFMode.TX/RX
int	BeamId	1 - limit of beamId storage ref. getBeamIdStorage()

## Returns

Declaration type	Description	Values
string	BeamPattern in json format	"{'beamId': 64, 'mode': 1, 'db': 12.0, 'theta': 0, 'phi': 0, 'channel_switch': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]}"

## [BBox 5G Series] setBeamPattern

### Signature

RetCode setBeamPattern(string sn, RFMode mode, int beamId, float db, int theta, int phi)

### Description

Set device beam pattern with beamId, mode, db, theta and phi

### Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
RFMode	mode	RFMode.TX/RX
int	beamId	1 - limit of beamId storage ref. getBeamIdStorage()
float	db(In dynamic range)	7.5
int	theta	15
int	phi	0-359

**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



## Parameters

Declaration type	Description	Parameters
string	Device SN	"D2104L001-28"
int[] as list	TC_Config: 1. TX_C 2. TX_Q 3. RX_C 4. 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	

## [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	"00G"

---

## [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	

## [UDBox] 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"><li>● True: harmonic</li><li>● False: not harmonic</li></ul>	

## [UDBox] getRecommendLO

### Signature

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

### Description

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

### Parameters

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

## Returns

Declaration type	Description	Values
dict	dict contains keys: <ul style="list-style-type: none"><li>● "USBLo"</li><li>● "LSBLo",</li><li>● "Recommend",</li></ul>	dict contains values from keys

## [UDBox] setUDFreq

### Signature

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

### Description

Set UDBox Frequency.

### Parameters

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

## Returns

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

## [UDBox] getUDFreq

### Signature

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

### Description

Set UDBox Frequency.

### 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"> <li>● "UDFreq"</li> <li>● "RFFreq",</li> <li>● "IFFreq",</li> </ul>	Gets value 0 from dict if old UD FW ver



---

# [UDBox] getUDState

## Signature

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

## Description

UDBox current status.

## Parameters

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

## Returns

Declaration type	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] setUDState

## Signature

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

## Description

Set UDBox status.

## Parameters

Declaration type	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

Declaration type	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