Xiamen Ursalink Technology Co., Ltd.
No. 63 Wanghai Road, 2nd Software Park, Xiamen, China
Phone: +86-592-5023060 Fax: +86-592-5023065
Web: www.ursalink.com Email: sales@ursalink.com

Ursalink Python SDK API

Welcome

Thanks for choosing Ursalink UG87 LoRaWAN Gateway . This introduction is intended for users who need to conduct secondary development with UG87 LoRaWAN Gateway. You can refer to this guide for further development of some main functions. For more details, please contact your Ursalink technical engineer.

Document History

Version	Description	Date	Firmware Version	SDK Version
v1.11	/	Jul 10th, 2018	87.1.0.X	1.2.0

Interface List

Name	Description
URDBAPI	The function interface of user data cache
URMessageChannel	The function interface for asynchronous communication programming model and asynchronous clock programming model

Attention:

As SDK will be installed on external memory card, please make sure that the memory card is not corrupted before installation starts. Errors like ServiceError, InsideError, or RuntimeError may occur if system-related services are not running correctly or the SDK version is too old.

URDBAPI Module

URDBAPI module provides a DBHandle class for data cache. You can refer to the detailed description as below.

Class Method	**__init__(*dbPath*)**
Parameter	*dpPath*: database file path. Without file path, a new file path will be created.
Example	"/tmp/test.db"
Class Method	**addValue(*key, value*)**
Parameter	*Key*: the index node of data cache. *Value*: the content of data cache.
Description	Add a data cache.
Return Value	None
Example	name = "ip" value={'ipv4':'192.168.2.166','mac':'76:34:55:96:69:11', 'BoardCast':'192.168.2.255'} db.addValue(name, value)
Class Method	**delValue(*key*)**
Parameter	*key*: the index node of data is deleted.
Description	Delete a piece of data.
Return Value	None
Example	db.delValue(name)
Class Method	**showValue(*key*)**
Parameter	*key*: the index node of data to be viewed.

Description	View a piece of data.
Return Value	[{'mac': '76:34:55:96:69:11', 'BoardCast': '192.168.2.255', 'ipv4': '192.168.2.166'}]
Example	db.showValue(name)
Class Method	**getList()**
Parameter	None.
Description	Obtain the index nodes of all data.
Return Value	eg: ['ip', 'power']
Example	db.getList()
Class Method	**updateValue(key, value)**
Parameter	*Key*: the index node of data to be updated. *value*: the content of new data.
Description	Update the content of the specific data.
Return Value	None
Example	value2={'ipv4':'192.168.2.167','mac':'76:34:55:96:69:12','BoardCast':'192.168.2.255' } Db.updateValue(name, value2)
Class Method	**rmDBFile()**
Parameter	None.
Description	Delete the database file.
Return Value	None
Example	db.rmDBFile()

URMessageChannel Module

Based on nanomsg and libevent library, URMessageChannel module has encapsulated several asynchronous communication model classes (Push/Pull model, Sub/Pub model, Request/Response model) and a timer event class, which can be used directly or developed. It also provides a separate timer function that can be used directly.

Class Method	**init_base()**
Parameter	None
Description	Create an event base
Return Value	event base
Class Method	**registerTimer(*base, callback, userdata = None*)**
Parameter	*base*: event base *Callback*: callback function *userdata*: the data that sends to callback function
Description	The registration event that triggers timer
Return Value	The event object.
Class Method	**startTimer(*evt, timeout*)**
Parameter	*evt*: event object *timeout*: the duration that delay to trigger timer
Description	Start timer
Return Value	None
Class Method	**stopTimer(*evt*)**
Parameter	*evt*: event object
Description	Delete timer
Return Value	None
Class Method	**start(*base*)**
Parameter	*base*: event base
Description	Start event loop
Return Value	None
Class Method	**stop(*base, sec = 0*)**
Parameter	*base*: event base
Description	Stop event loop
Return Value	None
Class Method	**logconfig(*loglevel*)**

Parameter	*loglevel*: log level. Select from “info”, “debug”, “error”, “critical”, “fatal”, and “warn”.
Description	Define log level
Return Value	None

MessagePull Class

Class Method	**__init__(*base, callback, domain*)**
Parameter	*base*: event base *Callback*: callback function *domain*: message communication address
Return Value	None

MessagePush Class

Class Method	**__init__(*base, domain*)**
Parameter	*base*: event base *domain*: message communication address

MessagePublish Class

Class Method	**__init__(*base, domain*)**
Parameter	*base*: event base *domain*: message communication address

MessageSubscribe Class

Class Method	**__init__(*base, callback, domain*)**
Parameter	*base*: event base *Callback*: callback function *domain*: message communication address

MessageRequest Class

Class Method	__init__(*base, callback, domain*)
Parameter	*base*: event base *Callback*: callback function *domain*: message communication address

MessageResponse Class

Class Method	__init__(*base, domain*)
Parameter	*base*: event base *domain*: message communication address

Note:
Domain should be the legal address type defined by nanomsg. For example, tcp://127.0.0.1:12525, ipc:///test-pull.ipc.

Remark:
All of the above mentioned class ("MessagePull", "MessagePush", "MessagePublish", "MessageSubcribe", "MessageRequest"and "MessageReponse) are the subclass of "MessageChannel".
Here's a list of public methods of "MessageChannel". Please do not use super class directly.

MessageChannel Class

Class Method	startReading(*sec = 0*)
Parameter	*sec*: the period of delay before start
Description	Start to read event
Return Value	None
Class Method	stopReading()
Parameter	None
Description	Stop reading event
Return Value	None
Class Method	startWriting(*sec = 0*)
Parameter	*sec*: the period of delay before start

Description	Start to write event
Return Value	None
Class Method	**stopWriting()**
Parameter	None
Description	Stop writing event
Return Value	None
Class Method	**write(*data*)**
Description	Send data
Return Value	None
Class Method	**gotData(*data*)**
Parameter	*data*: the data that sends to callback function
Description	The data that receives from callback function
Return Value	None

TimerEvtHandle Class

Class Method	**__init__(*base, timer*)**
Parameter	*base*: the object of the event base *timer*: the time when triggers timer
Class Method	**timerHandle(*evt, userdata*)**
Parameter	*evt*: the event object *userdata*: the data that sends to callback function
Description	The callback function of timer. The subclass need to rewrite it.
Return Value	NotImplemented
Class Method	**startTimer(*timer = 0*)**
Parameter	*timer*: the period of delay before start
Description	Start timer
Return Value	None
Class Method	**stop()**

Parameter	None
Description	Delete timer and stop event loop.
Return Value	None
Class Method	**start()**
Parameter	None
Description	Start event loop.
Return Value	None

[END]