M190

LoRaWAN Module Library API Manual

V1.0.0

EasyLinkin

Sensing the World

Catalogue

1. Document description

1.1 Brief description

The main purpose of the document is to describe the API of development library, summarizing the description of each API interface, description of some customized types and various error codes returned by API. It's convenient for developers to query.

1.2 Targeted users

Technicians who use the library for open source development.

1.3 Document convention

The following two public header files define all types used in the documents, except for the customized types.

stdint.h	uint8_t uint16_t uint32_t int8_t int16_t int32_t
stdbool.h	bool

2. Detailed description of API library

The api described in this chapter is basically the same as the description in the header file. If there are differences, please contact Easylinkin technical staff to check and update.

2.1 Public api interface

2.11 Library initialization

This function needs to be put at the front of the client code.

void	**eli_lib_init** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	-	-	-

2.12 Library main cycle

The customer needs to put this function in the while main loop, which includes sleep, and stops in this loop after entering the low-power mode.

void	**lib_loop_main** (void)		
	type	parameter values	remark
input parameter	-	-	-

output parameter	-	-	-
return code	-	-	-

2.13 Get the library version

err_code_t	**lib_get_version** (uint8_t pvers[16])		
	type	parameter values	remark
input parameter	uint8 *	pointer, to return the version information	16 Byte
output parameter	uint8 *	version information,string format	"Vx.x.x .x" form
return code	err_code_t	➢ ERR_NULL- -Null pointer ➢ ERR_NONE	-

2.14 Reset system

void	**lib_system_reset** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	-	-	-

2.2 LoRaWAN communication configuration api

2.2.1 Register callback function

err_code_t	**lorawan_register_cb** (lorawan_user_cb_st *pcb)		
	type	parameter values	remark
input parameter	lorawan_user_cb_st	pcb construct pointer refer to appendix for structure content	User callback function ➢ join- join network successfully/failed callback ➢ sent- send successfully callback ➢ recv- receive successfully callback
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE -- ok ➢ ERR_OTHER -- fail	-

2.2.2 Join LoRaWAN network (join)

err_code_t	**lorawan_join** (join_params_st *params)		
	type	parameter values	remark
input	join_params_st	Params, network access parameter configuration	refer to

parameter		NULL, uses default configuration parameters ➢ retries- - number of network retries ➢ interval- - network access interval ➢ rejoin_time-- the time to rejoin the network after the failure to join the network	appendix for structure content
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_JOINING- -joining network	-

2.2.3 Stop LoRaWAN network access (stop join)

err_code_t	**lorawan_join_stop** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_UNJOINING-- not joining network, do not need to stop	-

2.2.4 Get LoRaWAN network access status

join_state_emt	**lorawan_join_state** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	join_state_emt	➢ RUN_UNJOIN ➢ RUN_JOINING ➢ RUN_JOINED	return current network access status

2.2.5 Send data

err_code_t	**lorawan_send_data** (bool is_confirm, uint8_t *pbuf, uint16_t size, uint8_t retries)		
	type	parameter values	remark
input parameter	bool	➢ true-- confirm type data ➢ false-- unconfirm type data	➢ confirm type data, stop sending until get response from NS ➢ uncomfirm type data, send with fixed number of times according to the settings. If the correct data is received, it will stop sending.
	uint8_t *	data buffer	
	uint16_t	data length	
	uitn8_t	number of send data retries	

output parameter	-	-	-
return code	err_code_t	➢ ERR_UNJOIN-- not joined network ➢ ERR_TX_BUSY- - data is transmitting ➢ ERR_LEN_OUT- - data length exceeds limit (maximum data length differs in different data rates)	-

2.2.6 Set device DEVEUI (device number needs to be unique)

err_code_t	**lorawan_deveui_set** (uint8_t *p_deveui)		
	type	parameter values	remark
input parameter	uint8_t *	device DEVEUI, size: 8Byte	-
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_NULL ➢ ERR_JOINING-- joining network, not configurable	-

2.2.7 Get device DEVEUI

uint8_t *	**lorawan_deveui_get** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	uint8_t *	pointer, pointing to the deveui	-

2.2.8 Configure OTAA mode parameters

err_code_t	**lorawan_otaa_params_set** (lorawan_otaa_params_st *params)		
	type	parameter values	define
input parameter	lorawan_otaa_params_s t *	network parameters for otaa mode	refer to appendix for details
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_NULL ➢ ERR_JOINING-- joining network, not configurable	-

2.2.9 Get OTAA mode parameters

err_code_t	**lorawan_otaa_params_get** (lorawan_otaa_params_st *params)

	type	parameter values	remark
input parameter	-	-	-
output parameter	lorawan_otaa_params_st	return otaa mode parameters (deveui /appeui /appkey)	refer to appendix for details
return code	err_code_t	ERR_NONE ERR_NULL	-

2.2.10 Configure ABP mode parameters

err_code_t	**lorawan_abp_params_set** (lorawan_abp_params_st *params)		
	type	parameter values	remark
input parameter	lorawan_abp_params_st *	network parameters for abp mode	refer to appendix for details
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_NULL ➢ ERR_JOINING- -joining network,parameter obtained may be incorrect	-

2.2.11 Get ABP mode parameters (also used to get network parameters in OTAA mode after joined network)

err_code_t	**lorawan_abp_params_get** (lorawan_abp_params_st *params)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	lorawan_abp_params_st	return abp mode parameters	refer to appendix for details
return code	err_code_t	➢ ERR_NONE ➢ E RR_NULL ➢ ERR_JOINING- -joining network,parameter obtained may be incorrect	-

2.2.12 Configure current access network mode (required before join network)

err_code_t	**lorawan_join_mode_set** (bool is_otaa)		
	type	parameter values	remark

input parameter	bool	➢ true--OTAA mode ➢ false--ABP mode	-
output parameter	-	-	-
return code	err_code_t	ERR_NONE	-

2.2.13 Network access mode (either OTAA or ABP)

bool	**lorawan_is_otaa_mode** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	bool	➢ true--OTAA mode ➢ false--ABP mode	-

2.2.14 Enable ADR (automatic rate adjustment)

err_code_t	**lorawan_adr_on** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	ERR_NONE	-

2.2.15 Disable ADR

err_code_t	**lorawan_adr_off** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	ERR_NONE	-

2.2.16 Configure transmission data rate

err_code_t	**lorawan_datarate_set** (uint8_t dr)		
	type	parameter values	remark
input parameter	uint8_t	transmission data rate different configurable range in different region/band generally DR0-DR5, the larger the faster and the shorter the distance	refer to appendix for details
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_ADR_ON-- can't be configured when ADR is on ➢ ERR_OTHER- -an unknown error	-

2.2.17 Get current transmission data rate

uint8_t	**lorawan_datarate_get** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	uint8_t	return current data rate	different configurable range in different region/band, refer to the appendix

2.2.18 Configure device type:Class(A/C)

err_code_t	**lorawan_class_set** (uint8_t class_type)		
	type	parameter values	remark
input parameter	uint8_t	➢ 0 -- ClassA ➢ 1 -- ClassC	-
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_PARAMS-- illegal parameters	-

2.2.19 Get device type

uint8_t	**lorawan_class_get** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	uint8_t	➢ 0 -- ClassA ➢ 1 -- ClassC	-

2.2.20 Configure transmission power level

err_code_t	**lorawan_power_set** (uint8_t pwr)		
	type	parameter values	remark
input parameter	uint8_t	power level ➢ <15--0, maximum power lever for current region, with a 2dbm reduction for every 1 increase ➢ 0xF0--0xFF, maximum power lever of current RF chip, with a 1dbm reduction for every 1 decrease	power levels differs in different region/band, refer to appendix for details
output parameter	-	-	-

return code	err_code_t	➢ ERR_NONE ➢ ERR_PARAMS-- illegal power ➢ ERR_ADR_ON-- power cannot be manually configured when ADR is on ➢ ERR_OTHER-- an unknown error	-

2.2.21 Get current transmission power level

uint8_t	**lorawan_power_get** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	uint8_t	return current power level	-

2.2.22 Send a heartbeat payload

err_code_t	**lorawan_send_heartbeat** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_TX_BUSY-- in communication, no time to send a heartbeat payload(try again later)	-

2.2.23 Enable automatic sending heartbeat payload

err_code_t	**lorawan_heartbeat_on** (uint16_t period)		
	type	parameter values	remark
input parameter	uint16_t	➢ heartbeat cycle (unit:S) ➢ 0-- heartbeat cycle does not change, just enable heartbeat	-
output parameter	-	-	-
return code	err_code_t	ERR_NONE	-

2.2.24 Disable automatic sending heartbeat payload

err_code_t	**lorawan_heartbeat_off** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	ERR_NONE	-

2.2.25 Set up channel (different regions differ)

err_code_t	**lorawan_channel_set** (void *tx, void *rx1, void *rx2)		
	type	parameter values	remark
input parameter	void *	tx-- configure transmitting frequency point	fill in NULL if no settings
	void *	rx1 reception frequency point (some region does not require configuration)	-
	void *	rx2- -rx2 receiving frequency point	-
output parameter	-		-
return code	err_code_t	➢ ERR_NONE ➢ ERR_FREQ-- illegal frequency	-

2.2.26 Get channel configuration

err_code_t	**lorawan_channel_get** (void *tx, void *rx1, void *rx2)		
	type	parameter values	remark
input parameter	-	-	➢ fill in NULL if no need to get the value; ➢ no answer if unsupported channel.
output parameter	void *	tx-- return the frequency point set before	-
	void *	rx1-- return rx1 frequency point (not	-

		supported by some region)	
	void *	rx2-- return rx2 frequency point	-
return code	err_code_t	ERR_NONE	-

2.2.27 Configure LBT on/off (listen before talk, supported by some region)

err_code_t	**lorawan_lbt_set** (bool on)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	ERR_NONE	-

2.2.28 Get LBT status

err_code_t	**lorawan_lbt_get** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	bool	ERR_NONE ERR_DR -- illegal DR	-

2.2.29 Configure RX2 data rate

err_code_t	**lorawan_rx2dr_set** (uint8_t dr)		
	type	parameter values	remark
input parameter	uint8	data rate	-
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_DR-- illegal DR value, and set failed	-

2.2.30 Get RX2 data rate configuration

uint8_t lorawan_rx2dr_get (void)			
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	uint8_t	return current RX2 DR value	-

2.2.31 Check network connection status

err_code_t	**lorawan_link_check** (void)		
	type	parameter values	remark
input parameter	-	-	-

output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_TX_BUSY-- in communication, can not perform linkcheck (try again later)	This function depends on communication with NS, please do not request frequently

2.2.32 Execute time synchronization

err_code_t	**lorawan_time_sync** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE ➢ ERR_TX_BUSY-- in communication, can not perform time synchronization (try again later)	It is recommended to perform it once before time query. This function depends on communication with NS, please do not request frequently

2.2.33 Get current time (time synchronization is required before the acquisition)

uint32_t	**lorawan_time_get** (DateTime_t *p_dt)		
	type	parameter values	remark
input parameter	DateTime_t *	pointer (input null pointer does not output DateTime_t format time)	-
output parameter	DateTime_t	output current time	-
return code	uint32_t	return current time (Unit:S)	-

2.2.34 Format saved configuration (device restarted automatically after formatted)

err_code_t	**lorawan_config_format** (void)		
	type	parameter values	remark
input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE	-

2.2.35 Save configuration

err_code_t	**lorawan_config_save** (void)		
	type	parameter values	remark

input parameter	-	-	-
output parameter	-	-	-
return code	err_code_t	➢ ERR_NONE	-

3. Customize type summary description

In order to query quickly, the customized types used in each part are summarized below, if any additions or deletions,need to be updated in this section.

3.1 Structure type

3.1.1 datatime_st

```
typedef struct datetime{
uint16_t       year;
uint8_t        month;
uint8_t        day;
uint8_t        hour;
uint8_t        minute;
uint8_t        second;
uint16_t       millisecond;
} datetime_st ;
```

3.1.2 join_params_st

```
typedef struct {
uint16_t       retries;
uint16_t       interval;
uint16_t       rejoin_time;
} join_params_st;
```

3.1.3 lorawan_otaa_params_st

```
typedef struct {
uint8_t       deveui[8];
uint8_t       appeui[8];
uint8_t       appkey[16];
} lorawan_otaa_params_st;
```

3.1.4 lorawan_abp_params_st

```
typedef struct {
uint8_t       deveui[8];
uint8_t       nwkskey[16];
uint8_t       appskey[16];
uint8_t       devaddr[4];
} lorawan_abp_params_st;
```

3.1.5 lorawan_user _cb_st

```
typedef struct {
void (*join)(bool is_joined, uint16_t cnt, uint16_t total);
void (*sent)(bool is_sent_ok, uint8_t rety_nr);
void (*recv)(uint8_t type, uint8_t *pbuf, uint8_t size, uint8_t port, int16_t rssi);
} lorawan_user_cb_st;

// when data is received,the pointer will be used.
void (*recv)(uint8_t type, uint8_t *pbuf, uint8_t size, uint8_t port, int16_t rssi);

TYPE: 1Byte, downlink data type
Bit0: 0-unconfirm, 1-confirm
Bit1: 0-non-ACK, 1-ACK
Bit2: 0-uncarried, 1-carried, indicating whether downlink data carries LINK command
response
Bit3: 0-uncarried, 1-carried, indicating whether downlink data carries TIME command
response, only when this bit is 1, it means that the time synchronization is successful
Bit4~Bit7: default 0, reserved
```

3.2 Enumeration type

3.2.1 class_type_emt

```
enum {
API_CLASS_A,
API_CLASS_C,
};
```

Appendix 1 (error code)

0	ERR_NONE	No error
-0x1001	ERR_NULL	Empty pointer error
-0x1002	ERR_PARAMS	Parameter error
-0x1003	ERR_FREQ	Illegal frequency
-0x1004	ERR_DR	Illegal data rate
-0x1005	ERR_PWR	Power level configuration
-0x1006	ERR_OTHER	Unknown error
-0x1007	ERR_UNJOINING	Unjoining network, current request is not available
-0x1008	ERR_JOINING	Joining network, current request is not available
-0x1009	ERR_UNJOIN	Not joined network, the current request is not available
-0x100A	ERR_LEN_OUT	Data length exceed limit
-0x100B	ERR_ADR_ON	ADR is on, and the configuration cannot be performed
-0x1010	ERR_TX_BUSY	It is currently in transmit and cannot be sent
-0x1100	ERR_NOT_SUPPORT	Unsupported interface
-0x1101	ERR_UNKNOWN	Unknown error, replace ERR_OTHER, both represent unknown errors

Appendix 2 (configuration list can be saved after power-off)

For easy and fast batch configuration, the library can save some parameters after power-off. In addition, you can modify default settings by importing configuration file.

Note:

The client's program needs to use API save to save the configuration, otherwise it will not be saved automatically.

class	Lorawan class
adr	Lorawan adr on/off
datarate	Lorawan tx datarate
power	Lorawan tx power
port	Lorawan tx port
ntrials	Lorawan tx retry times
rx2dr	Rx2 datarate
channel	Lorawan channel
heartbeat	Heartbeat configuration (on/off or cycle)
lbt	Listen before talk (available in some regions)