MAXIIOT-LRWAN110-API-Function-Reference-V2_0

`MAXIIOT-LRWAN1.1.0-API-Function-Reference-V2.0`

Company: MAXIIOT Co.Ltd.

Department: R&D Team

Web: http://www.maxiiot.com/index.html

Date: 2019-01-16

Background & Summary

The purpose of this document is to describe for the LoRaWAN API function of MAXIIOT LoRaWAN products. This document will be useful for engineers to use these modules for secondary development.

Revision History

Revision	Date	Author	Descriptions
V2.0	19.01.16	Michael Lee	Designed for LoRaWAN 1.1.0

Table of Contents

1.System Configuration

LoRaWanSetSaveConfig

Define

```
void LoRaWanSetSaveConfig( void );
```

Description

Save the LoRaWAN configuration parameters to the flash, the parameters will not be lost after the mcu is powered off.

parameters

void

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaWanSetSaveConfig();
```

LoRaWanSetRestoreFactory

Define

```
void LoRaWanSetRestoreFactory( void );
```

Description

All parameters of the module are restored to the factory configuration and saved to the flash.

parameters

void

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaWanSetRestoreFactory();
```

2.Activation Configuration

LoRaWanSetClassType

Define

```
void LoRaWanSetClassType(DeviceClass_t Class);
```

Description

Set the target LoRaWAN class type of the terminal.

parameters

- DeviceClass_t Class
  - The target LoRaWAN class type as input

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaWanSetClassType( CLASS_C );
```

LoRaWanGetClassType

Define

```
void LoRaWanGetClassType(DeviceClass_t *Class);
```

Description

Get the target LoRaWAN class type of the terminal.

parameters

- DeviceClass_t *class
  - The target LoRaWAN class type will be placed in the pointer to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
DeviceClass_t class;

LoRaWanGetClassType( &class );
```

LoRaWanSetClass

Define

```
LoRaMacStatus_t LoRaWanSetClass( DeviceClass_t Class );
```

Description

Modify the current LoRaWAN class type of the terminal immediately.

parameters

- DeviceClass_t Class
  - The target LoRaWAN class type as input

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;

status = LoRaWanSetClass( CLASS_C );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetClass

Define

```
LoRaMacStatus_t LoRaWanGetClass( DeviceClass_t *Class );
```

Description

Get the current LoRaWAN class type of the terminal immediately.

parameters

- DeviceClass_t *class
    - The current LoRaWAN class type will be placed in the pointer to return.

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
DeviceClass_t class;

status = LoRaWanGetClass( &class );
if  (status == LORAMAC_STATUS_OK)
{
    printf("%d", class);
}
```

LoRaWanSetNwkAVTWay

Define

```
void LoRaWanSetNwkAVTWay(LoRaJoinMode_t mode);
```

Description

Modify the target network activation way of the terminal.

parameters

- LoRaJoinMode_t mode
  - The target network activation way as input
- LoRaJoinMode_t define

```
typedef enum{
    ABP_JOIN = 0,
    OTAA_JOIN,
}LoRaJoinMode_t;
```

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaWanSetNwkAVTWay(OTAA_JOIN);
```

LoRaWanGetNwkAVTWay

Define

```
void LoRaWanGetNwkAVTWay(LoRaJoinMode_t *mode);
```

Description

Get the target network activation way of the terminal.

parameters

- LoRaJoinMode_t *mode
  - The target network activation way will be placed in the pointer to return.
- LoRaJoinMode_t define

```
typedef enum{
    ABP_JOIN = 0,
    OTAA_JOIN,
}LoRaJoinMode_t;
```

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaJoinMode_t mode;

LoRaWanGetNwkAVTWay(&mode);
```

LoRaWanSetNwkAVT

Define

```
LoRaMacStatus_t LoRaWanSetNwkAVT( ActivationType_t activation );
```

Description

Modify the network activation way of the terminal. Terminal will rejoin or quit activation immediately.

parameters

- ActivationType_t activation
  - The network activation way as input
- ActivationType_t define

```
/*!
 * End-Device activation type
 */
typedef enum eActivationType
{
    /*!
     * None
     */
    ACTIVATION_TYPE_NONE = 0,
    /*!
     * Activation By Personalization (ACTIVATION_TYPE_ABP)
     */
    ACTIVATION_TYPE_ABP = 1,
    /*!
     * Over-The-Air Activation (ACTIVATION_TYPE_OTAA)
     */
    ACTIVATION_TYPE_OTAA = 2,
}ActivationType_t;
```

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;

status = LoRaWanSetNwkAVT(ACTIVATION_TYPE_OTAA);
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetNwkAVT

Define

```
LoRaMacStatus_t LoRaWanGetNwkAVT( ActivationType_t *activation );
```

Description

Get the status of activation.

parameters

- ActivationType_t *activation
  - The activation status will be placed in the pointer to return.
- ActivationType_t define

```
/*!
 * End-Device activation type
 */
typedef enum eActivationType
{
    /*!
     * None
     */
    ACTIVATION_TYPE_NONE = 0,
    /*!
     * Activation By Personalization (ACTIVATION_TYPE_ABP)
     */
    ACTIVATION_TYPE_ABP = 1,
    /*!
     * Over-The-Air Activation (ACTIVATION_TYPE_OTAA)
     */
    ACTIVATION_TYPE_OTAA = 2,
}ActivationType_t;
```

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
ActivationType_t activation;

status = LoRaWanGetNwkAVT( &activation );
if  (status == LORAMAC_STATUS_OK)
{
    printf("%d", activation);
}
```

LoRaWanSetADDR

Define

```
LoRaMacStatus_t LoRaWanSetADDR( uint32_t Addr );
```

Description

Modify the terminal's LoRaWAN Devaddr.DevAddr defaults to the last four bytes of DevEUI.

parameters

- uint32_t Addr: The device address as input

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;

status = LoRaWanSetADDR(0x00000001);
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetADDR

Define

```
LoRaMacStatus_t LoRaWanGetADDR( uint32_t *Addr );
```

Description

Read the terminal's LoRaWAN Devaddr. DevAddr defaults to the last four bytes of DevEUI.

parameters

- uint32_t *Addr
  - The device address will be placed in the pointer to return.

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint32_t address;

status = LoRaWanGetADDR( &address );
if  (status == LORAMAC_STATUS_OK)
{
    printf("%08x", address);
}
```

LoRaWanGetDeviceEUI

Define

```
void LoRaWanGetDeviceEUI( uint8_t DEUI[8] );
```

Description

Read the LoRaWAN unique code DevEUI of the terminal device. This value is generated at the factory. The value of each terminal is different and cannot be modified.

parameters

- uint8_t DEUI[8] :The DevEUI data will be placed in the array to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t DevEUI[8];

LoRaWanGetDeviceEUI(DevEUI);
```

LoRaWanSetJoinEUI

Define

```
void LoRaWanSetJoinEUI( uint8_t JoinEUI[8] );
```

Description

Modify the LoRaWAN JoinEUI value of the terminal.

parameters

- uint8_t JoinEUI[8]: The JoinEUI value as input

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t joineui[8] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};

LoRaWanSetJoinEUI( joineui );
```

LoRaWanGetJoinEUI

Define

```
void LoRaWanGetJoinEUI( uint8_t JoinEUI[8] );
```

Description

Readthe LoRaWAN JoinEUI value of the terminal.

parameters

- uint8_t JoinEUI[8]
    - The JoinEUI will be placed in the array to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t joineui[8];

LoRaWanGetJoinEUI( joineui );
```

LoRaWanSetAppKey

Define

```
LoRaMacStatus_t LoRaWanSetAppKey( uint8_t AppKey[16] );
```

Description

Modify the LoRaWAN AppKey value of the terminal.

parameters

- uint8_t AppKey[16]: The AppKey value as input

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t Key[16] = {0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6,0xAB, 0xF7, 0x15,
0x88, 0x09, 0xCF, 0x4F, 0x3C};

status = LoRaWanSetAppKey( Key );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetAppKey

Define

```
void LoRaWanGetAppKey( uint8_t AppKey[16] );
```

Description

Readthe LoRaWAN AppKey value of the terminal.

parameters

- uint8_t AppKey[16]
  - The AppKey will be placed in the array to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t Key[16];

LoRaWanGetAppKey( Key );
```

LoRaWanSetNwkKey

Define

```
LoRaMacStatus_t LoRaWanSetNwkKey( uint8_t NwkKey[16] );
```

Description

Modify the LoRaWAN NwkKey value of the terminal.

parameters

- uint8_t NwkKey[16]: The NwkKey value as input

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t Key[16] = {0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6,0xAB, 0xF7, 0x15,
0x88, 0x09, 0xCF, 0x4F, 0x3C};

status = LoRaWanSetNwkKey( Key );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetNwkKey

Define

```
void LoRaWanGetNwkKey( uint8_t NwkKey[16] );
```

Description

Readthe LoRaWAN NwkKey value of the terminal.

parameters

- uint8_t NwkKey[16]
    - The NwkKey will be placed in the array to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t Key[16];

LoRaWanGetNwkKey( Key );
```

LoRaWanSetFNwkSIntKey

Define

```
LoRaMacStatus_t LoRaWanSetFNwkSIntKey( uint8_t FNwkSIntKey[16] );
```

Description

Modify the LoRaWAN FNwkSIntKey value of the terminal.

parameters

- uint8_t FNwkSIntKey[16]: The FNwkSIntKey value as input

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t Key[16] = {0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6,0xAB, 0xF7, 0x15,
0x88, 0x09, 0xCF, 0x4F, 0x3C};

status = LoRaWanSetFNwkSIntKey( Key );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetFNwkSIntKey

Define

```
void LoRaWanGetFNwkSIntKey( uint8_t FNwkSIntKey[16] );
```

Description

Readthe LoRaWAN FNwkSIntKey value of the terminal.

parameters

- uint8_t FNwkSIntKey[16]
    - The FNwkSIntKey will be placed in the array to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t Key[16];

LoRaWanGetFNwkSIntKey( Key );
```

LoRaWanSetSNwkSIntKey

Define

```
LoRaMacStatus_t LoRaWanSetSNwkSIntKey( uint8_t SNwkSIntKey[16] );
```

Description

Modify the LoRaWAN SNwkSIntKey value of the terminal.

parameters

- uint8_t SNwkSIntKey[16]: The SNwkSIntKey value as input

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t Key[16] = {0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6,0xAB, 0xF7, 0x15,
0x88, 0x09, 0xCF, 0x4F, 0x3C};

status = LoRaWanSetSNwkSIntKey( Key );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetSNwkSIntKey

Define

```
void LoRaWanGetSNwkSIntKey( uint8_t SNwkSIntKey[16] );
```

Description

Readthe LoRaWAN SNwkSIntKey value of the terminal.

parameters

- uint8_t SNwkSIntKey[16]
    - The SNwkSIntKey will be placed in the array to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t Key[16];

LoRaWanGetSNwkSIntKey( Key );
```

LoRaWanSetNwkSEncKey

Define

```
LoRaMacStatus_t LoRaWanSetNwkSEncKey( uint8_t NwkSEncKey[16] );
```

Description

Modify the LoRaWAN NwkSEncKey value of the terminal.

parameters

- uint8_t NwkSEncKey[16]: The NwkSEncKey value as input

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t Key[16] = {0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6,0xAB, 0xF7, 0x15,
0x88, 0x09, 0xCF, 0x4F, 0x3C};

status = LoRaWanSetNwkSEncKey( Key );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetNwkSEncKey

Define

```
void LoRaWanGetNwkSEncKey( uint8_t NwkSEncKey[16] );
```

Description

Readthe LoRaWAN NwkSEncKey value of the terminal.

parameters

- uint8_t NwkSEncKey[16]
  - The NwkSEncKey will be placed in the array to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t Key[16];

LoRaWanGetNwkSEncKey( Key );
```

LoRaWanSetAppSKey

Define

```
LoRaMacStatus_t LoRaWanSetAppSKey( uint8_t AppSKey[16] );
```

Description

Modify the LoRaWAN AppSKey value of the terminal.

parameters

- uint8_t AppSKey[16]: The AppSKey value as input

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t Key[16] = {0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6,0xAB, 0xF7, 0x15,
0x88, 0x09, 0xCF, 0x4F, 0x3C};

status = LoRaWanSetAppSKey( Key );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetAppSKey

Define

```
void LoRaWanGetAppSKey( uint8_t AppSKey[16] );
```

Description

Read the LoRaWAN AppSKey value of the terminal.

parameters

- uint8_t AppSKey[16]
    - The AppSKey will be placed in the array to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t Key[16];

LoRaWanGetAppSKey( Key );
```

3.General Configuration

LoRaWanSetRX2Channel

Define

```
LoRaMacStatus_t LoRaWanSetRX2Channel( Rx2ChannelParams_t rx2channel);
```

Description

Modify the RXWIN2 parameters of the LoRaWAN terminal.

parameters

- Rx2ChannelParams_t rx2channel: The rxwin2 value as input
- Rx2ChannelParams_t define

```
/*!
 * LoRaMAC receive window 2 channel parameters
 */
typedef struct sRx2ChannelParams
{
    /*!
     * Frequency in Hz
     */
    uint32_t Frequency;
    /*!
     * Data rate
     *
     * LoRaWAN Regional Parameters V1.0.2rB
     *
     * The allowed ranges are region specific. Please refer to \ref DR_0 to \ref DR_15
for details.
     */
    uint8_t  Datarate;
}Rx2ChannelParams_t;
```

Return

LoRaMacStatus_t: The result of the function

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.
- The details of rxwin2 parameter reference to `<<LoRaWAN Regional Parameters v1.1.0>>`.

Example

```
LoRaMacStatus_t status;
Rx2ChannelParams_t param;

param.Frequency = 434500000;
param.Datarate = 0;

status = LoRaWanSetRXWIN2( param );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetRX2Channel

Define

```
LoRaMacStatus_t LoRaWanGetRX2Channel( Rx2ChannelParams_t *rx2channel );
```

Description

Read the RXWIN2 parameters of the LoRaWAN terminal.

parameters

- Rx2ChannelParams_t *rx2channel
  - The rxwin2 parameter will be placed in the array to return.
- Rx2ChannelParams_t define

```
/*!
 * LoRaMAC receive window 2 channel parameters
 */
typedef struct sRx2ChannelParams
{
    /*!
     * Frequency in Hz
     */
    uint32_t Frequency;
    /*!
     * Data rate
     *
     * LoRaWAN Regional Parameters V1.0.2rB
     *
     * The allowed ranges are region specific. Please refer to \ref DR_0 to \ref DR_15
for details.
     */
    uint8_t  Datarate;
}Rx2ChannelParams_t;
```

Return

void

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.
- The details of rxwin2 parameter reference to `<<LoRaWAN Regional Parameters v1.1.0>>`.

Example

```
LoRaMacStatus_t status;
Rx2ChannelParams_t param;

status = LoRaWanGetRXWIN2( &param );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanSetChannelsMask

Define

```
LoRaMacStatus_t LoRaWanSetChannelsMask( uint16_t ChannelsMask[6] );
```

Description

Modify the channel masking variable of the LoRaWAN terminal.

parameters

- uint16_t ChannelsMask[6]:
  - The channel mask value as input
  - Each member of the ChannelsMask array controls the opening and blocking of 16 channels;
  - One bit member corresponds to one channel, a bit of 1 represents the channel is turned on, and a bit of 0 represents the channel is turned off.
  -

ChannelsMask	Channels (From low to high bits)
0	CH0 --- CH15
1	CH16 --- CH15
2	CH32 --- CH31
3	CH48--- CH63
4	CH64 --- CH79
5	CH80 --- CH95

Return

LoRaMacStatus_t: The result of the function

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.
- The number of channels in each ISM band is different. If the maximum number of channels is exceeded, errors are likely to occur.

- The details of ISM band channel mask parameter reference to `<<LoRaWAN Regional Parameters v1.1.0>>`.

Example

```
LoRaMacStatus_t status;
uint16_t ChMask[6] = {0x0007,0x0000,0x0000,0x0000,0x0000,0x0000};

status = LoRaWanSetChannelsMask( ChMask );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetChannelsMask

Define

```
LoRaMacStatus_t LoRaWanGetChannelsMask( uint16_t ChannelsMask[6] );
```

Description

Read the channel masking variable of the LoRaWAN terminal.

parameters

- uint16_t ChannelsMask[6]
  - The channel mask value will be placed in the array to return.

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint16_t ChMask[6];

status = LoRaWanGetChannelsMask( ChMask );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanSetDataRate

Define

```
LoRaMacStatus_t LoRaWanSetDataRate( uint8_t DataRate );
```

Description

Modify the data rate of uplink frame.

parameters

- uint8_t DataRate: The data rate value as input

DataRate	Description
0	DR0
1	DR1
2	DR2
3	DR3
4	DR4
5	DR5
6	DR6
7	DR7
8	DR8
9	DR9
10	DR10
11	DR11
12	DR12
13	DR13
14	DR14
15	DR15

Return

LoRaMacStatus_t: The result of the function

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.
- The details of data rate parameter reference to `<<LoRaWAN Regional Parameters v1.1.0>>`.

Example

```
LoRaMacStatus_t status;

status = LoRaWanSetDataRate( DR_5 );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetDataRate

Define

```
LoRaMacStatus_t LoRaWanGetDataRate( uint8_t *DataRate );
```

Description

Read the data rate of uplink frame.

parameters

- uint8_t *DataRate
    - The uplink data rate value will be placed in the pointer to return.

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t datarate;

status = LoRaWanGetDataRate( &datarate );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanSetADR

Define

```
LoRaMacStatus_t LoRaWanSetADR( bool enable );
```

Description

Enable or disable the adaptive data rate of the terminal.

parameters

- bool enable: The enable value as input
  - true: enable
  - false: disable

Return

LoRaMacStatus_t: The result of the function

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;

status = LoRaWanSetADR( true );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetADR

Define

```
LoRaMacStatus_t LoRaWanGetADR( bool *enable );
```

Description

Read the enable or disable status of the adaptive data rate of the terminal.

parameters

- bool *enable
  - The enable status value will be placed in the pointer to return.

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
bool enable;

status = LoRaWanGetADR( &enable );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanSetTxPower

Define

```
LoRaMacStatus_t LoRaWanSetTxPower( uint8_t TxPower );
```

Description

Modify the tx power of uplink frame.

parameters

- uint8_t TxPower: The tx power value as input
  - 0: TX_POWER_0 Max EIRP

    1: TX_POWER_1 Max EIRP – 2dB

    2: TX_POWER_2 Max EIRP – 4dB

    3: TX_POWER_3 Max EIRP – 6dB

    4: TX_POWER_4 Max EIRP – 8dB

    5: TX_POWER_5 Max EIRP – 10dB

    6: TX_POWER_6

    7: TX_POWER_7

    8: TX_POWER_8

    9: TX_POWER_9

Return

LoRaMacStatus_t: The result of the function

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.
- Different end-device's maximum power is different, please choose the right power to use.
- The details of tx_power parameter reference to the **TX power table** of `<<LoRaWAN Regional Parameters v1.1.0>>`.

Example

```
LoRaMacStatus_t status;

status = LoRaWanSetTxPower( 1 );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetTxPower

Define

```
LoRaMacStatus_t LoRaWanGetTxPower( uint8_t *TxPower );
```

Description

Read the tx power of uplink frame.

parameters

- uint8_t *TxPower
    - The uplink tx power value will be placed in the pointer to return.

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t txpower;

status = LoRaWanGetTxPower( &txpower );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanSetPingSlotDR

Define

```
LoRaMacStatus_t LoRaWanSetPingSlotDR( uint8_t DateRate );
```

Description

Modify the ping slot data rate of the terminal.

parameters

- uint8_t DataRate: The data rate value as input

DataRate	Description
0	DR0
1	DR1
2	DR2
3	DR3
4	DR4
5	DR5
6	DR6
7	DR7
8	DR8
9	DR9
10	DR10
11	DR11
12	DR12
13	DR13
14	DR14
15	DR15

Return

LoRaMacStatus_t: The result of the function

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.
- The details of data rate parameter reference to `<<LoRaWAN Regional Parameters v1.1.0>>`.

Example

```
LoRaMacStatus_t status;

status = LoRaWanSetPingSlotDR( DR_3 );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetPingSlotDR

Define

```
LoRaMacStatus_t LoRaWanGetPingSlotDR( uint8_t *DateRate );
```

Description

Read the ping slot data rate of the terminal.

parameters

- uint8_t *DataRate
    - The ping slot data rate value will be placed in the pointer to return.

Return

LoRaMacStatus_t: The result of the function

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t datarate;

status = LoRaWanGetPingSlotDR( &datarate );
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanSetPingSlotPY

Define

```
void LoRaWanSetPingSlotPY( uint8_t Periodicity );
```

Description

Modify the target ping slot periodicity of the terminal. It needs to send a command - `PingSlotInfot` to take effect.

parameters

- uint8_t Periodicity: The target ping slot periodicity value as input
  - Range from 0 to 7.
  - Ping slot periodicity is equal to 2^ seconds.
  - For example: 2^3 = 8 seconds. The end-device will open an Rx ping slot every 8 seconds.

Return

void

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaWanSetPingSlotPY(3);
```

LoRaWanGetPingSlotPY

Define

```
void LoRaWanGetPingSlotPY( uint8_t *Periodicity );
```

Description

Read the target ping slot periodicity of the terminal.

parameters

- uint8_t *Periodicity
  - The target ping slot periodicity value will be placed in the pointer to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t periodicity;

LoRaWanGetPingSlotPY( &periodicity );
```

LoRaWanSetPort

Define

```
bool LoRaWanSetPort( uint8_t Port );
```

Description

Modify the uplink frame port of the terminal.

parameters

- uint8_t Port: The port value as input
  - Range from 1 to 255

Return

- bool: The result of the function
  - true: ok
  - false: fail

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaWanSetPort(3);
```

LoRaWanGetPort

Define

```
void LoRaWanGetPort( uint8_t *Port );
```

Description

Read the uplink frame port of the terminal.

parameters

- uint8_t *Port
  - The port value will be placed in the pointer to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint8_t port;

LoRaWanGetPort( &port );
```

LoRaWanSetFrameType

Define

```
void LoRaWanSetFrameType( Mcps_t type );
```

Description

Modify the uplink frame type of the terminal.

Parameters

- Mcps_t type: The type value as input

Return

- void

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaWanSetFrameType(MCPS_CONFIRMED);
```

LoRaWanGetFrameType

Define

```
void LoRaWanGetFrameType( Mcps_t *type );
```

Description

Read the uplink frame type of the terminal.

parameters

- Mcps_t *type:
    - The frame type value will be placed in the pointer to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
Mcps_t type

LoRaWanGetFrameType( &type );
```

LoRaWanGetCSQ

Define

```
void LoRaWanGetCSQ( int16_t *rssi, int8_t *snr );
```

Description

Read the rssi and snr value of the newest received downlink frame of the terminal.

parameters

- int16_t *rssi
  - The newest rssi value will be placed in the pointer to return.
  - unit: dBm
- int8_t *snr
  - The newest snr value will be placed in the pointer to return.
  - unit: dB

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
int16_t rssi;
int8_t snr;

LoRaWanGetCSQ( &rssi, &snr );
```

LoRaWanGetUTC

Define

```
void LoRaWanGetUTC( SysTime_t *time );
```

Description

Read the UTC time currently recorded by the terminal from network server. It requires the device has been activated and gateway is running the gps module.

parameters

- SysTime_t *time
  - The current UTC time value will be placed in the pointer to return.
- SysTime_t define

```
/*!
 * \brief Structure holding the system time in seconds and milliseconds.
 */
typedef struct SysTime_s
{
    uint32_t Seconds;//uint:s
    int16_t  SubSeconds;//unit:ms
}SysTime_t;
```

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
SysTime_t time;

LoRaWanGetUTC( &time );
```

LoRaWanCheckFlag

Define

```
uint16_t LoRaWanCheckFlag(void);
```

Description

Read the status of lorawan of the terminal. Matching to global variables `LoRaWanRxInfo`, engineer can know when the device receives data and what the data is.

parameters

void

Return

- uint16_t status: Each bit represents a flag sign.

```
typedef enum
{
    NONE = 0,
    TXTIMEOUT,
    RXDONE,
    RXTIMEOUT,
    RXERROR,
    CONFIRM_ACK,
    CONFIRM_NOACK,
    UNCONFIRM_DONE,
    OTAA_JOINOK,
    CLASSB_ENTER,
    CLASSB_QUIT,
    BEACON_FOUND,
    BEACON_NOFOUND,
}LoRaWanFLAG_Type_t;
```

Note

- The function is declared in the `LoRaWAN_api_v1.h` file.
- Once queried, the status flag is immediately assigned the value NONE.
- LoRaWanRxInfo define

```
typedef struct{
    int16_t rssi;
    int8_t snr;
    uint8_t port;
    uint8_t rxslot;
    uint16_t size;
    uint8_t buf[256];
}LoRaWanRxInfo_Type_t;
```

Example

```
uint16_t lorawanflag;

lorawanflag = LoRaWanCheckFlag();
if(lorawanflag&(1<<RXDONE))
{
    const char *slotStrings[] = { "1", "2", "C", "Ping-Slot", "Multicast Ping-Slot" };

    //receive a app downlink frame from lora server
    lora_printf("LORAWAN:    RXDONE\r\n");
    lora_printf("downlink len:%d\r\n",LoRaWanRxInfo.size);
    lora_printf("downlink frame:%s\r\n",LoRaWanRxInfo.buf);
    lora_printf("downlink win:%s\r\n",slotStrings[LoRaWanRxInfo.rxslot]);
    lora_printf("downlink port:%d\r\n",LoRaWanRxInfo.port);
```

```
    lora_printf("downlink rssi:%d,snr:%d\r\n",LoRaWanRxInfo.rssi,LoRaWanRxInfo.snr);
}
if(lorawanflag&(1<<TXTIMEOUT))
{
    lora_printf("LORAWAN:    TXTIMEOUT\r\n");
}
else if(lorawanflag&(1<<RXTIMEOUT))
{
    lora_printf("LORAWAN:    RXTIMEOUT\r\n");
}
else if(lorawanflag&(1<<RXERROR))
{
    lora_printf("LORAWAN:    RXERROR\r\n");
}
if(lorawanflag&(1<<OTAA_JOINOK))
{
    lora_printf("LORAWAN:    OTAA_JOINOK\r\n");
}
if(lorawanflag&(1<<CONFIRM_ACK))
{
    lora_printf("LORAWAN:    CONFIRM_ACK\r\n");
}
else if(lorawanflag&(1<<CONFIRM_NOACK))
{
    lora_printf("LORAWAN:    CONFIRM_NOACK\r\n");
}
else if(lorawanflag&(1<<UNCONFIRM_DONE))
{
    lora_printf("LORAWAN:    UNCONFIRM_DONE\r\n");
}
if(lorawanflag&(1<<CLASSB_ENTER))
{
    lora_printf("LORAWAN:    CLASSB_ENTER\r\n");
}
else if(lorawanflag&(1<<CLASSB_QUIT))
{
    lora_printf("LORAWAN:    CLASSB_QUIT\r\n");
}
if(lorawanflag&(1<<BEACON_FOUND))
{
    lora_printf("LORAWAN:    BEACON_FOUND\r\n");
    lora_printf("beacon rssi:%d,snr:%d\r\n",LoRaWanRxInfo.rssi,LoRaWanRxInfo.snr);
}
else if(lorawanflag&(1<<BEACON_NOFOUND))
{
    lora_printf("LORAWAN:    BEACON_NOFOUND\r\n");
}
```

LoRaWanSendBuf

Define

```
LoRaMacStatus_t LoRaWanSendBuf( Mcps_t type, uint8_t *buf, int size );
```

Description

The terminal sends an uplink frame once.

parameters

- Mcps_t type
  - The uplink frame type value as input.
- uint8_t *buf
  - The buffer prepared to be sent.
- int size
  - The buffer size,

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
LoRaMacStatus_t status;
uint8_t *buf = {"12345"};
int size = 5;

status = LoRaWanSendBuf(MCPS_CONFIRMED, buf, size);
if  (status == LORAMAC_STATUS_OK)
{
    printf("OK");
}
```

LoRaWanGetULCounter

Define

```
void LoRaWanGetULCounter( uint32_t *ulcounter);
```

Description

Read the uplink frame counter of the terminal.

parameters

- uint32_t *ulcounter
  - The ulcounter value will be placed in the pointer to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint32_t counter

LoRaWanGetULCounter( &counter );
```

LoRaWanGetDLCounter

Define

```
void LoRaWanGetDLCounter( uint32_t *dlcounter);
```

Description

Read the downlink frame counter of the terminal.

parameters

- uint32_t *dlcounter
  - The dlcounter value will be placed in the pointer to return.

Return

void

Note

The function is declared in the `LoRaWAN_api_v1.h` file.

Example

```
uint32_t counter

LoRaWanGetDLCounter( &counter );
```

4.Parameter definition list

DeviceClass_t

```
/*!
 * LoRaWAN devices classes definition
 *
 * LoRaWAN Specification V1.0.2, chapter 2.1
 */
typedef enum eDeviceClass
{
    /*!
     * LoRaWAN device class A
     *
     * LoRaWAN Specification V1.0.2, chapter 3
     */
    CLASS_A = 0x00,
    /*!
     * LoRaWAN device class B
     *
     * LoRaWAN Specification V1.0.2, chapter 8
     */
    CLASS_B = 0x01,
    /*!
     * LoRaWAN device class C
     *
     * LoRaWAN Specification V1.0.2, chapter 17
     */
    CLASS_C = 0x02,
}DeviceClass_t;
```

LoRaMacStatus_t

```
/*!
 * LoRaMAC Status
 */
typedef enum eLoRaMacStatus
{
    /*!
     * Service started successfully
     */
    LORAMAC_STATUS_OK,
    /*!
     * Service not started - LoRaMAC is busy
     */
    LORAMAC_STATUS_BUSY,
    /*!
     * Service unknown
     */
    LORAMAC_STATUS_SERVICE_UNKNOWN,
    /*!
     * Service not started - invalid parameter
     */
```

```
    LORAMAC_STATUS_PARAMETER_INVALID,
    /*!
     * Service not started - invalid frequency
     */
    LORAMAC_STATUS_FREQUENCY_INVALID,
    /*!
     * Service not started - invalid datarate
     */
    LORAMAC_STATUS_DATARATE_INVALID,
    /*!
     * Service not started - invalid frequency and datarate
     */
    LORAMAC_STATUS_FREQ_AND_DR_INVALID,
    /*!
     * Service not started - the device is not in a LoRaWAN
     */
    LORAMAC_STATUS_NO_NETWORK_JOINED,
    /*!
     * Service not started - payload length error
     */
    LORAMAC_STATUS_LENGTH_ERROR,
    /*!
     * Service not started - the specified region is not supported
     * or not activated with preprocessor definitions.
     */
    LORAMAC_STATUS_REGION_NOT_SUPPORTED,
    /*!
     * The application data was not transmitted
     * because prioritized pending MAC commands had to be sent.
     */
    LORAMAC_STATUS_SKIPPED_APP_DATA,
    /*!
     * ToDo
     */
    LORAMAC_STATUS_DUTYCYCLE_RESTRICTED,
    /*!
     *
     */
    LORAMAC_STATUS_NO_CHANNEL_FOUND,
    /*!
     *
     */
    LORAMAC_STATUS_NO_FREE_CHANNEL_FOUND,
     /*!
      * ToDo
      */
    LORAMAC_STATUS_BUSY_BEACON_RESERVED_TIME,
     /*!
      * ToDo
      */
    LORAMAC_STATUS_BUSY_PING_SLOT_WINDOW_TIME,
     /*!
      * ToDo
```

```
     */
    LORAMAC_STATUS_BUSY_UPLINK_COLLISION,
    /*!
     * An error in the cryptographic module is occurred
     */
    LORAMAC_STATUS_CRYPTO_ERROR,
    /*!
     * An error in the frame counter handler module is occurred
     */
    LORAMAC_STATUS_FCNT_HANDLER_ERROR,
    /*!
     * An error in the MAC command module is occurred
     */
    LORAMAC_STATUS_MAC_COMMAD_ERROR,
    /*!
     * An error in the Class B module is occurred
     */
    LORAMAC_STATUS_CLASS_B_ERROR,
    /*!
     * An error in the Confirm Queue module is occurred
     */
    LORAMAC_STATUS_CONFIRM_QUEUE_ERROR,
    /*!
     * Undefined error occured
     */
    LORAMAC_STATUS_ERROR
}LoRaMacStatus_t;
```

Mcps_t

```
/*!
 *
 * \brief   LoRaMAC data services
 *
 * \details The following table list the primitives which are supported by the
 *          specific MAC data service:
 *
 * Name                  | Request | Indication | Response | Confirm
 * --------------------- | :-----: | :--------: | :------: | :-----:
 * \ref MCPS_UNCONFIRMED | YES     | YES        | NO       | YES
 * \ref MCPS_CONFIRMED   | YES     | YES        | NO       | YES
 * \ref MCPS_MULTICAST   | NO      | YES        | NO       | NO
 * \ref MCPS_PROPRIETARY | YES     | YES        | NO       | YES
 *
 * The following table provides links to the function implementations of the
 * related MCPS primitives:
 *
 * Primitive        | Function
 * ---------------- | :--------------------:
 * MCPS-Request     | \ref LoRaMacMlmeRequest
 * MCPS-Confirm     | MacMcpsConfirm in \ref LoRaMacPrimitives_t
```

```
 * MCPS-Indication  | MacMcpsIndication in \ref LoRaMacPrimitives_t
 */
typedef enum eMcps
{
    /*!
     * Unconfirmed LoRaMAC frame
     */
    MCPS_UNCONFIRMED,
    /*!
     * Confirmed LoRaMAC frame
     */
    MCPS_CONFIRMED,
    /*!
     * Multicast LoRaMAC frame
     */
    MCPS_MULTICAST,
    /*!
     * Proprietary frame
     */
    MCPS_PROPRIETARY,
}Mcps_t;
```