LoRaWAN API Function Reference Manual

`DL7612&DL7812-API-Function-Reference-V1.6`

MAXIIOT R&D Department

2018-01-31

Background & Summary

The purpose of this document is to describe for the LoRaWAN API function of DL7612 and DL7812. This document will be useful for other users to use these modules for secondary development.

Revision History

Revision	Date	Author	Descriptions
V1.6	18.01.31	Michael Li	Created

System Config

LoRaWanSetSaveConfig

Define

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

Declaration

Save the LoRaWAN configuration parameter to Flash, and the parameter will not lost when the MCU is out of power.

Parameters

NULL

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
LoRaWanSetSaveConfig();
```

LoRaWanSetRestoreFactory

Define

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

Declaration

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

Parameters

NULL

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
LoRaWanSetRestoreFactory();
```

Network Access Setting

LoRaWanGetDeviceEUI

Define

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

Declaration

Read theunique code of LoRaWAN device(DevEUI) , which is factory-generated , different for each device and can not be modified.

Parameters

uint8_t DEUI[8] : The DevEUI data that is read will be placed in the array to return

Refunds

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t DevEUI[8];

LoRaWanGetDeviceEUI(DevEUI);
```

LoRaWanSetADDR

Define

```
void LoRaWanSetADDR( uint32_t Addr );
```

Declaration

Modify the device address (DevAddr) of LoRaWAN device. The defaults value of DevAddr is the last four bytes of DevEUI.

Parameters

uint32_t Addr : Modify DevAddr to Addr.

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

LoRaWanGetADDR

Define

```
uint32_t LoRaWanGetADDR( void );
```

Declaration

Read the device address (DevAddr) of LoRaWAN device. The defaults value of DevAddr is the last four bytes of DevEUI.

Parameters

NULL

Returns

DevAddr of device

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint32_t devaddr;

devaddr = LoRaWanGetADDR();
```

LoRaWanSetAppEUI

Define

```
void LoRaWanSetAppEUI( uint8_t AEUI[8] );
```

Declaration

Modify the AppEUI value of LoRaWAN device. AppEUI value defaults to 0.

Parameters

uint8_t AEUI[8]：Modify AppEUI to AEUI.

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t APPEUI[8] = {0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08}；

LoRaWanSetAppEUI( APPEUI );
```

LoRaWanGetAppEUI

Define

```
void LoRaWanGetAppEUI( uint8_t AEUI[8] );
```

Declaration

Read the AppEUI value of LoRaWAN device. AppEUI value defaults to 0.

Parameters

uint8_t AEUI[8] ： It returns the AppEUI value.

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t AppEUI[8];

LoRaWanGetAppEUI(AppEUI);
```

LoRaWanSetAppKey

Define

```
void LoRaWanSetAppKey( uint8_t AK[16] );
```

Declaration

Modify the AppKey value of LoRaWAN device.

Parameters

uint8_t AK[16]：Modify AppKey to AK

Refund

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t AppKey[16] =
{0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08}；

LoRaWanSetAppKey( AppKey );
```

LoRaWanGetAppKey

Define

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

Declaration

Read the AppKey value of LoRaWAN device.

Parameters

uint8_t AK[16] ：it returns the AppKey value.

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t AppKey[16];

LoRaWanGetAppKey(AppKey);
```

LoRaWanSetAppSKey

Define

```
void LoRaWanSetAppSKey( uint8_t ASK[16] );
```

Declaration

Modify the AppSKey of LoRaWAN device.

Parameters

uint8_t ASK[16]：Modify AppSKey to ASK

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t AppSKey[16] =
{0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08};

LoRaWanSetAppSKey( AppSKey );
```

LoRaWanGetAppSKey

Define

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

Declaration

Read the AppSKey of LoRaWAN device.

Parameters

uint8_t ASK[16] ：it returns the AppSKey value.

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t AppSKey[16];

LoRaWanGetAppSKey(AppSKey);
```

LoRaWanSetNetworkSKey

Define

```
void LoRaWanSetNetworkSKey( uint8_t NSK[16] );
```

Declaration

Modify the NwkSKey of LoRaWAN device.

Parameters

uint8_t NSK[16]：Modify NwkSKey to NSK

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t NwkSKey[16] =
{0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08};

LoRaWanSetNetworkSKey( NwkSKey );
```

LoRaWanGetNetworkSKey

Define

```
void LoRaWanGetNetworkSKey( uint8_t NSK[16] );
```

Declaration

Read the NwkSKey of LoRaWAN device.

Parameters

uint8_t NSK[16] ：it returns the NwkSKey value.

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t NwkSKey[16];

LoRaWanGetNetworkSKey(NwkSKey);
```

LoRaWanSetClass

Define

```
bool LoRaWanSetClass( DeviceClass_t Class );
```

Declaration

Modify the class of LoRaWAN Device.

Parameters

- DeviceClass_t Class：the class of LoRaWAN Device to be modify.
- The define of DeviceClass_t

  ```
  typedef enum eDeviceClass
  {
      /*!
       * LoRaWAN device class A
       *
       * LoRaWAN Specification V1.0, chapter 3ff
       */
      CLASS_A,
      /*!
       * LoRaWAN device class B
       *
       * LoRaWAN Specification V1.0, chapter 8ff
       */
      CLASS_B,     //CLASS_B disable
      /*!
       * LoRaWAN device class C
       *
       * LoRaWAN Specification V1.0, chapter 17ff
       */
      CLASS_C,
  }DeviceClass_t;
  ```

-

Returns

Modify successfully return 1 , failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file.

Example

```
LoRaWanSetClass( CLASS_C );
```

LoRaWanGetClass

Define

```
uint8_t LoRaWanGetClass( void );
```

Declaration

Read the class of LoRaWAN Device.

Parameters

NULL

Returns

- the class of LoRaWAN Device

Returns	Class
0	CLASS_A
2	CLASS_C

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t class;

class = LoRaWanGetClass();
```

LoRaWanSetISMBand

Define

```
bool LoRaWanSetISMBand( uint8_t ISMBand );
```

Declaration

Modify the ISM Band of LoRaWAN device.

Parameters

uint8_t ISMBand：the types of ISM band.

You can choose one from the enum type variable---ISM_Band_TYPE_t.

Returns

Set successful return 1, failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file.
- The define of ISM_Band_TYPE_t can be find in ISM_Band1.h file ;

```
typedef enum
{
    EU863_870 = 0,
    US902_928,
    CN779_787,
    CN470_510,
    EU433,
    CUSTOMIZE,
}ISM_Band_TYPE_t;
```

- The channel frequency and data rate of each ISM band are different. Please check the document for details.<>.

Example

```
uint8_t flag = 0；
ISM_Band_TYPE_t ismband；
ismband = EU433；

flag = LoRaWanSetISMBand( ismband );
```

LoRaWanGetISMBand

Define

```
uint8_t LoRaWanGetISMBand( void );
```

Declaration

Read the ISM band of LoRaWAN device.

Parameters

NULL

Returns

LoRaWAN ISM band of device ，Values from 0 to 5 ,correspond to members of the enumeration variable ISM_Band_TYPE_t.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file；
- ISM_Band_TYPE_t enum Define in ISM_Band1. file ；

```
typedef enum
{
    EU863_870 = 0,
    US902_928,
    CN779_787,
    CN470_510,
    EU433,
    CUSTOMIZE,
}ISM_Band_TYPE_t;
```

- The channel frequency and data rate of each ISM band are different. Please check the document for details<>.

Example

```
ISM_Band_TYPE_t isband；

isband = LoRaWanGetISMBand();
```

LoRaWanSetAutoJoinMode

Define

```
void LoRaWanSetAutoJoinMode( LoRaAutoJoinMode_t mode );
```

Declaration

Modify Join Mode of LoRaWAN device

Parameters

- LoRaAutoJoinMode_t mode：Join mode parameter
- LoRaAutoJoinMode_t Define

  ```
  typedef enum{
      ABP_JOIN = 0,   //ABP jion
      OTAA_JOIN,       //OTAA jion
  }LoRaAutoJoinMode_t;
  ```

Returns

Successful return 1, failure return 0.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
LoRaWanSetAutoJoinMode( OTAA_JOIN );
```

LoRaWanGetAutoJoinMode

Define

```
LoRaAutoJoinMode_t LoRaWanGetAutoJoinMode( void );
```

Declaration

Read join mode of LoRaWAN device

Parameters

NULL

Returns

Join mode parameter

Returns	Represents
0	ABP_JOIN
1	OTAA_JOIN

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
LoRaAutoJoinMode_t mode;

mode = LoRaWanGetAutoJoinMode();
```

LoRaWanSendOTAAJoinReq

Define

```
OTAAReturn_Type_t LoRaWanSendOTAAJoinReq( uint8_t *devEui, uint8_t *appEui, uint8_t *appKey );
```

Declaration

LoRaWAN device send an OTAA（Over-The-Air-Activation） Join Request

Parameters

uint8_t *devEui：input device DevEUI.

uint8_t *appEui：input device AppEUI

uint8_t *appKey：input device AppKey

Returns

-

Returns	Represents
0	OK
1	BUSY
2	NO_NETWORK_JOINED
3	LENGTH_ERROR
4	SERVICE_UNKNOWN
6	DEVICE_OFF

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t DevEui[8];
uint8_t AppEui[8];
uint8_t AppKey[16];

LoRaWanGetDeviceEUI(DevEui);
LoRaWanGetAppEUI(AppEui);
LoRaWanGetAppKey(AppKey);

LoRaWanSendOTAAJoinReq( DevEui, AppEui, AppKey );
```

LoRaWanSetABPJoinReq

Define

```
void LoRaWanSetABPJoinReq( uint32_t netID, uint32_t devAddr, uint8_t *nwkSKey, uint8_t *appSKey
);
```

Declaration

Set device to ABP（Activation By Personalization） Join mode .

Parameters

uint32_t netID：input network id

uint32_t devAddr：input DevAddr

uint8_t *nwkSKey：input nwkSKey

uint8_t *appSKey：input appSKey

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint32_t DevAddr;
uint8_t NwkSKey[16];
uint8_t AppSKey[16];

DevAddr = LoRaWanGetADDR();
LoRaWanGetNetworkSKey(NwkSKey);
LoRaWanGetAppSKey(AppSKey);

LoRaWanSetABPJoinReq( 0x000000, DevAddr, NwkSKey, AppSKey );
```

LoRaWanGetNetworkJoined

Define

```
bool LoRaWanGetNetworkJoined( void );
```

Declaration

Check if the device is already on the network.

Parameters

NULL

Returns

accepted returns 1，failure returns 0.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint8_t flag;

flag = LoRaWanGetNetworkJoined();
```

RX Setting

LoRaWanSetJoinAcceptDelay1

Define

```
void LoRaWanSetJoinAcceptDelay1( uint32_t delayus );
```

Declaration

Modify join accept delay time of RXWIN1 (JoinAcceptDelay1) .

Parameters

uint32_t delayus：Modify join accept delay time of RXWIN1 to delayus ，Unit us.

Returns

NULL

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file.
- It is not recommended to modify this parameter. Otherwise, may not able to receive gateway information.

Example

```
uint32_t JoinAcceptDelay1 ;
JoinAcceptDelay1 = 5000000 ;

LoRaWanSetJoinAcceptDelay1( JoinAcceptDelay1 );
```

LoRaWanGetJoinAcceptDelay1

Define

```
uint32_t LoRaWanGetJoinAcceptDelay1( void );
```

Declaration

Read join accept delay time of RXWIN1 (JoinAcceptDelay1).

Parameters

NULL

Returns

Join accept delay time of RXWIN1(JoinAcceptDelay1)，Unit us.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file.

Example

```
uint32_t JoinAcceptDelay1；

JoinAcceptDelay1 = LoRaWanGetJoinAcceptDelay1();
```

LoRaWanSetJoinAcceptDelay2

Define

```
void LoRaWanSetJoinAcceptDelay2( uint32_t delayus );
```

Declaration

Modify join accept delay time of RXWIN2 (JoinAcceptDelay2) .

JoinAcceptDelay2 default to JoinAcceptDelay1+1000000us.

Parameters

uint32_t delayus：Modify join accept delay time of RXWIN2 to delayus，Unit us.

Returns

NULL

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- It is not recommended to modify this parameter. Otherwise, may not able to receive gateway information.

Example

```
uint32_t JoinAcceptDelay2 ;
JoinAcceptDelay2 = 6000000 ;

LoRaWanSetJoinAcceptDelay2( JoinAcceptDelay2 );
```

LoRaWanGetJoinAcceptDelay2

Define

```
 uint32_t LoRaWanGetJoinAcceptDelay2( void );
```

Declaration

Read join accept delay time of RXWIN2 (JoinAcceptDelay2) .

Parameters

NULL

Returns

Join accept delay time of RXWIN2 (JoinAcceptDelay2) ，Unit us.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file

Example

```
uint32_t JoinAcceptDelay2 ;

JoinAcceptDelay2 = LoRaWanGetJoinAcceptDelay2();
```

LoRaWanSetReceiveDelay1

Define

```
void LoRaWanSetReceiveDelay1( uint32_t delayus );
```

Declaration

Modify receive delay time of RXWIN1(ReceiveDelay1) .

Parameters

uint32_t delayus：Modify receive delay time of RXWIN1 to delayus，Unit us.

Returns

NULL

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file；
- It is not recommended to modify this parameter. Otherwise, may not able to receive gateway information.

Example

```
uint32_t ReceiveDelay1；
ReceiveDelay1 = 1000000；

LoRaWanSetReceiveDelay1( ReceiveDelay1 );
```

LoRaWanGetReceiveDelay1

Define

```
 uint32_t LoRaWanGetReceiveDelay1( void );
```

Declaration

Read receive delay time of RXWIN1(ReceiveDelay1).

Parameters

NULL

Returns

Receive delay time of RXWIN1(ReceiveDelay1)，Unit us.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file

Example

```
uint32_t ReceiveDelay1；

ReceiveDelay1 = LoRaWanGetReceiveDelay1();
```

LoRaWanGetReceiveDelay2

Define

```
 uint32_t LoRaWanGetReceiveDelay2( void );
```

Declaration

Read receive delay time of RXWIN2(ReceiveDelay2).

ReceiveDelay2 default to ReceiveDelay1+1000000us，not allowed to modify.

Parameters

NULL

Returns

Receive delay time of RXWIN2(ReceiveDelay2)，Unit us.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file

Example

```
uint32_t ReceiveDelay2；

ReceiveDelay2 = LoRaWanGetReceiveDelay2();
```

LoRaWanGetUpLinkCounter

Define

```
uint32_t LoRaWanGetUpLinkCounter( void );
```

Declaration

Read the number of uplink lorawan data frames sent by the device after powering on.

Parameters

NULL

Returns

Data frame uplink times of enabled LoRaWAN device

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file

Example

```
uint32_t UpLinkCounter ;

UpLinkCounter = LoRaWanGetUpLinkCounter();
```

LoRaWanGetDownLinkCounter

Define

```
uint32_t LoRaWanGetDownLinkCounter( void );
```

Declaration

Read the number of downlink lorawan data frames received by the device after powering on.

Parameters

NULL

Returns

Data frame downlink times of enabled LoRaWAN device

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file

Example

```
uint32_t DownLinkCounter ;

DownLinkCounter = LoRaWanGetDownLinkCounter();
```

LoRaWanSetRXWIN2

Define

```
bool LoRaWanSetRXWIN2( Rx2ChannelParams_t param );
```

Declaration

Modify RXWIN2 parameters of LoRaWAN device

Parameters

- Rx2ChannelParams_t param：RXWIN2 Parameters
- Rx2ChannelParams_t type Define

```
typedef struct sRx2ChannelParams
{
    /*!
     * Frequency in Hz
     */
    uint32_t Frequency; //RXWIN2 Usage frequency
    /*!
     * Data rate
     */
    uint8_t  Datarate; //data rate of RXWIN2
}Rx2ChannelParams_t;
```

Returns

Modify successfully return 1 , failure return 0

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file；

Data rates represent different meanings in different ISM bands.

Example

```
Rx2ChannelParams_t param;
param.Frequency = 434500000;
param.Datarate = 0;

LoRaWanSetRXWIN2( param );//Set frequency of RXWIN2 to 434.5MHz and data rate to DR0
```

LoRaWanGetRXWIN2

Define

```
Rx2ChannelParams_t LoRaWanGetRXWIN2(void);
```

Declaration

Read RXWIN2 parameters of LoRaWAN device

Parameters

NULL

Returns

- RXWIN2 parameters of LoRaWAN device
- Rx2ChannelParams_type Define

```
typedef struct sRx2ChannelParams
{
    /*!
     * Frequency in Hz
     */
    uint32_t Frequency; //RXWIN2 usage frequency
    /*!
     * Data rate
     */
    uint8_t  Datarate; //data rate of RXWIN2
}Rx2ChannelParams_t
```

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file ;

Data rates represent different meanings in different ISM bands.

Example

```
Rx2ChannelParams_t param;

param = LoRaWanGetRXWIN2();
```

LoRaWanCheckFlag

Define

```
LoRaWanFLAG_Type_t LoRaWanCheckFlag(void);
```

Declaration

Check the user prompt status flag of device

Parameters

NULL

Returns

- User prompt status flag
- LoRaWanFLAG_Type_tDefine

  ```
  typedef enum
  {
      NONE = 0,
      TXDONE,
      TXTIMEOUT,
      RXDONE,
      RXTIMEOUT,
      RXERROR,
      ACK_RECEIVE,
      ACK_UNRECEIVE,
      CMD_RECEIVE,
      OTAA_JOINOK,
  }LoRaWanFLAG_Type_t;
  ```

  A query will only return one status flag, once inquired, the status flag is immediately assigned to NONE. The system will automatically modify the status flag when the system recognizes the new status. And then the user can query again.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file ;

Example

```
LoRaWanFLAG_Type_t flag;

loraflag = LoRaWanCheckFlag();
if(loraflag==OTAA_JOINOK)
{
    //OTAA join successfully
else if(loraflag==RXDONE)
{
    //Received downlink application layer data frame
}
else if(loraflag == TXDONE){
    //Uplink data frame  Sent successfully
}
```

LoRaWanGetRSSI_SNR

Define

```
void LoRaWanGetRSSI_SNR( int16_t *rssi, uint8_t *snr);
```

Declaration

Read the RSSI and SNR value of latest received LoRaWan data frame by device

Parameters

int16_t *rssi：Get the latest RSSI of LoRa data frame via rssireadloar pointer

uint8_t *snr：Get the latest SNR of LoRa data frame via snr pointer

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file

Example

```
int16_t *RSSI;
uint8_t *SNR;

LoRaWanGetRSSI_SNR( RSSI, SNR );
```

LoRaWanSetOriginalRxFramePrint

Define

```
void LoRaWanSetOriginalRxFramePrint( OriginalRxFramePrint_t printflag );
```

Declaration

By default, when the device receives a LoRaWAN data frame, the serial port will only print the decoded application layer data.

When this function is enabled, the entire contents of the LoRaWAN protocol frame will be printed, helping users to know the whole process of LoRaWAN communication.

Parameters

- OriginalRxFramePrint_t printflag：choose on or off
- OriginalRxFramePrint_t Define

  ```
  typedef enum{
      NoPrint = 0,//do not print the entire contents of LoRaWAN protocol frame
      Print,//print the entire contents of LoRaWAN protocol frame
  }OriginalRxFramePrint_t;
  ```

Returns

NULL

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file；

Example

```
LoRaWanSetOriginalRxFramePrint( NoPrint );
```

TX Settings

LoRaWanSetChannelMask

Define

```
void LoRaWanSetChannelMask( uint16_t ChMask[6] );
```

Declaration

Modify LoRaWAN device channel mask variables.

Parameters

uint16_t ChMask[6]：Each member of the ChMask array controls the on or off of 16 channels;

One bit of a member corresponds to one channel, a bit of 1 means open channel, a bit of 0 means closed channel.

ChMask	Corresponding control channel (from low to high)
ChMask[0]	Channel 0 to Channel 15
ChMask[1]	Channel 16 to Channel 31
ChMask[2]	Channel 32 to Channel 47
ChMask[3]	Channel 48 to Channel 63
ChMask[4]	Channel 64 to Channel 78
ChMask[5]	Channel 80 to Channel 95

Returns

NULL

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file；
- The number of channels contained in each ISM band is different. If the channel number of the ChMask array is greater than the maximum number of channels in the ISM band, the channel corresponding to this part number is not opened.

Example

```
uint16_t ChMask[6] = {0x0007,0x0000,,0x0000,0x0000,0x0000,0x0000}//Open channel 0 to channel 2

LoRaWanSetChannelMask(ChMask);
```

LoRaWanGetChannelMask

Define

```
uint16_t* LoRaWanGetChannelMask( void );
```

Declaration

Read LoRaWAN device channel mask variables.

Parameters

NULL

Returns

Pointer to device channel mask variables.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file；

Example

```
uint16_t ChMask[6]；

ChMask = LoRaWanGetChannelMask();
```

LoRaWanSetChannelState

Define

```
bool LoRaWanSetChannelState(uint8_t StartCH_Num, uint8_t EndCH_Num, ChannelState_Type_t state);
```

Declaration

Modify channel state of LoRaWAN device (open or mask) ，from "StartCH_Num" to "EndCH_Num"，Unify channel to "state".

Parameters

uint8_t StartCH_Num：Start channel number ,ranges from 0 to 95.

uint8_t EndCH_Num：End channel number ，ranges fomr StartCH_Num to 95.EndCH_Num, larger than StartCH_Num.

ChannelState_Type_t state：Channel State ，0 represent close channel , 1 represent open channel

Returns

NULL

Additional Declarationdefine

- The function is declared in the LoRaWan_api_v1.h file；
- The number of channels contained in each ISM band is different. If the channel number of the ChMask array is greater than the maximum number of channels in the ISM band, the channel corresponding to this part number is not opened.
- ChannelState_Type_tDefine、

```
typedef enum
{
    OFF = 0,
    ON,
}ChannelState_Type_t;
```

Example

```
uint16_t ChMask[6] = {0x0007,0x0000,,0x0000,0x0000,0x0000,0x0000}//Open channel 0 to channel 2

LoRaWanSetChannelMask(ChMask);
```

LoRaWanAddChannel

Define

```
bool LoRaWanAddChannel(uint8_t CH_Num, ChannelParams_t param);
```

Declaration

Add or modify device ISM band channels.

Parameters

- uint8_t CH_Num：Channel Number ,Ranges from 0 to 95，Depend on the ISM band .
- ChannelParams_t param：

```
typedef struct sChannelParams
{
    /*!
     * Frequency in Hz
     */
    uint32_t Frequency;
    /*!
     * Data rate definition

     */
    DrRange_t DrRange;
    /*!
     * Band index
     */
    uint8_t Band;   //This value does not affect, can be ignored
}ChannelParams_t;
```

Returns

Modify successfully return 1 , failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- CH_Num can not exceed the maximum number of channels in the ISM band.
- Some channels cannot be modified, for details.<>.

Example

```
ChannelParams_t param;
param.Frequency = 433700000;
param.DrRange.Fields.Min = DR_0;
param.DrRange.Fields.Max = DR_5;

LoRaWanAddChannel(3, param);//add Channel 3, Channel frequency 43.7MHz , DR from 0 to 5.
```

LoRaWanDelChannel

Define

```
bool LoRaWanDelChannel(uint8_t CH_Num);
```

Declaration

Delete one of the ISM band channels .

Parameters

- uint8_t CH_Num： Channel number ，Ranges from 0 to 95，Depend on the ISM band

Returns

Modify successfully return 1 , failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- CH_Num can not exceed the maximum number of channels in the ISM band.
- Some channels cannot be modified, for details.<>.

Example

```
LoRaWanDelChannel(3);//Delate Channel 3
```

LoRaWanSetChannelDR

Define

```
bool LoRaWanSetChannelDR( uint8_t TxChDR );
```

Declaration

Modify the Tx channel data rate of LoRaWAN device

Parameters

- uint8_t TxChDR：Tx Channel data rate .

TxChDR value	Data rate
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

Returns

Modify successfully return 1 , failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- Data rates represent different meanings in different ISM bands. for details , see doc <>

Example

```
uint8_t TxChDR = 2;

LoRaWanSetChannelDR(TxChDR);
```

LoRaWanGetChannelDR

Define

```
uint8_t LoRaWanGetChannelDR( void );
```

Declaration

Read the Tx channel data rate of LoRaWAN device.

Parameters

NULL

Returns

- Tx channel data rate of LoRaWAN device

Returns	Data Rate
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

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- Data rates represent different meanings in different ISM bands. for details , see doc <>

Example

```
uint8_t TxChDR;

TxChDR = LoRaWanGetChannelDR();
```

LoRaWanSetCustomDRList

Define

```
uint8_t LoRaWanSetCustomDRList( uint8_t DR, uint8_t SF, uint16_t BW, uint8_t DLDR);
```

Declaration

Modify the meaning of the data rate in a custom ISM band（ CUSTOMIZE band ）

Parameters

- uint8_t DR：The data rate to be modified.
  - from DR0 to DR15.
- uint8_t SF：Set the spread spectrum factor.
  - from7 to12，represent SF7 toSF12.
- uint16_t BW：Set band width.
  - 0 represent125KHz，1represent250KHz，2represent500KHz.
- uint8_t DLDR：Set the downlink data rate corresponding to RXWIN1.
  - from DR0 to DR15.

Returns

Modify successfully return 1 , failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- Custom ISM band parameters can be configured.

Example

```
//Set the data rate DR0 of CUSTOMIZE band to SF12 and BW125 ,set the downlink data rate of
RXWIN1 to DR0
LoRaWanSetCustomDRList( 0, 12, 0, 0);
```

LoRaWanSetADR

Define

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

Declaration

Enable and disable LoRaWAN device adaptive data rate.

Parameters

bool enable：1 is enable , 0 is disable

Returns

Modify successfully return 1 , failure return 0..

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file；

Example

```
LoRaWanSetADR( 1 ); //enable adaptive data rate
```

LoRaWanGetADR

Define

```
bool LoRaWanGetADR( void );
```

Declaration

Read whether adaptive data rate is enabled .

Parameters

NULL

Returns

1 is enable , 0 is disable

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file

Example

```
uint8_t flag;

flag = LoRaWanGetADR();
```

LoRaWanSetSendMode

Define

```
bool LoRaWanSetSendMode(LoRaSendMode_t Mode);
```

Declaration

Modify device data send mode

Parameters

- LoRaSendMode_t Mode：send mode type
- LoRaSendMode_t Define

  ```
  typedef enum{
      AUTO,    //Automatical send mode
      MANUAL, //Manual send mode
      PASSTHROUGH, //Passthrough mode
  }LoRaSendMode_t;
  ```

Returns

Modify successfully return 1 , failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- In manual mode, users are required to manually send data frames each time.

- In automatic transmission mode, the device sends data once per working cycle, and the duty cycle is set by the function (LoRaWanSetAutoSendCycle).
- In pass-through mode, each data frame sent by the device using UART will be sent as the LoRaWAN data frame, and the LoRaWAN application layer data received by the device will also be transmitted directly to UART.

Example

```
LoRaWanSetSendMode(MANUAL);
```

LoRaWanGetSendMode

Define

```
LoRaSendMode_t LoRaWanGetSendMode( void );
```

Declaration

Read device data send mode

Parameters

NULL

Returns

Device data send mode

Returns	Represents
0	AUTO
1	MANUAL
2	PASSTHROUGH

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file；

Example

```
LoRaSendMode_t mode;


mode = LoRaWanGetSendMode();
```

LoRaWanSetFrameType

Define

```
void LoRaWanSetFrameType( LoRaWanFrameMode_t mode );
```

Declaration

Modify the data frame type of LoRaWAN device in auto-send mode.

Parameters

- LoRaWanFrameMode_t mode：data frame type
- LoRaWanFrameMode_t Define

```
typedef enum{
    Confrim = 0,//The gateway needs an ACK response.
    Unconfrim,//The gateway does not need an ACK response
}LoRaWanFrameMode_t;
```

Returns

NULL

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file；

Example

```
LoRaWanSetFrameType( Unconfrim );
```

LoRaWanGetFrameType

Define

```
LoRaWanFrameMode_t LoRaWanGetFrameType( void );
```

Declaration

Read the data frame type of LoRaWAN device in auto-send mode.

Parameters

NULL

Returns

Data frame type of LoRaWAN device in auto-send mode

Returns	Represent
0	Confirm
1	Unconfirm

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file；

Example

```
LoRaWanFrameMode_t type;

type = LoRaWanGetFrameType();
```

LoRaWanSetAutoSendFrame

Define

```
void LoRaWanSetAutoSendFrame( uint8_t Buf[], uint8_t BufLen);
```

Declaration

Modify the data frame content and length of LoRaWAN device in auto-send mode

Parameters

- uint8_t Buf[]：data frame content
- uint8_t BufLen：data frame length

Returns

NULL

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;

Example

```
uint8_t Buf[10] = {"hello lora"}

LoRaWanSetAutoSendFrame(Buf, 10);
```

LoRaWanSetAutoSendCycle

Define

```
bool LoRaWanSetAutoSendCycle( uint32_t s);
```

Declaration

Modify the auto send cycle of LoRaWAN device in auto-send mode

Parameters

uint32_t s：Send Cycle ,Unit s.

Returns

Modify successfully return 1 , failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- Send cycle must over 3 seconds, it is recommended to be more than 10 seconds
- In automatic transmission mode, the device sends data once per working cycle, and the duty cycle is set by the function (LoRaWanSetAutoSendCycle).
- In pass-through mode, each data frame sent by the device using UART will be sent as the LoRaWAN data frame, and the LoRaWAN application layer data received by the device will also be transmitted directly to UART.

Example

```
LoRaWanSetAutoSendCycle(30);
```

LoRaWanGetAutoSendCycle

Define

```
uint32_t LoRaWanGetAutoSendCycle( void );
```

Declaration

Read the auto send cycle of LoRaWAN device in auto-send mode

Parameters

NULL

Returns

Auto send cycle of LoRaWAN device in auto-send mode，Unit s.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file；

Example

```
uint32_t cycle;

cycle = LoRaWanGetAutoSendCycle();
```

LoRaWanSetSendPort

Define

```
bool LoRaWanSetSendPort(uint8_t Port);
```

Declaration

Modify uplink port of LoRaWAN device

Parameters

uint8_t Port：uplink port , can't be 0

Returns

Modify successfully return 1 , failure return 0.

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;

Example

```
LoRaWanSetSendPort(5);
```

LoRaWanGetSendPort

Define

```
uint8_t LoRaWanGetSendPort( void );
```

Declaration

Read uplink port of LoRaWAN device

Parameters

NULL

Returns

uplink port of LoRaWAN device

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file ;

Example

```
uint8_t port;

port = LoRaWanGetSendPort();
```

LoRaWanSetTxPower

Define

```
void LoRaWanSetTxPower( uint8_t TxPower);
```

Declaration

Modify LoRa device RF Tx power

Parameters

uint8_t TxPower：Modify RF TxPower

Returns

NULL

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file；

TxPower	Power（dBm）
0	20
1	14 (Max Power for DL7612/DL7812)
2	11
3	8
4	5
5	2

Example

```
uint8_t TxPower；
TxPower = 1；

LoRaWanSetTxPower( TxPower );
```

LoRaWanGetTxPower

Define

```
uint8_t LoRaWanGetTxPower( void );
```

Declaration

Read device LoRa RF Tx power

Parameters

NULL

Returns

Tx Power of device

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file

Example

```
uint8_t TxPower ;

TxPower = LoRaWanGetTxPower();
```

LoRaWanSendBuf

Define

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

Declaration

The device manually sends a LoRaWAN data frame..

Parameters

uint8_t type：data frame type

0： need to receive the gateway ACK signal data frame after sending

1： do not need to receive the gateway ACK signal data frame after sending

uint8_t *buf：data frame to send

int size：Data length

int retry：Number of retransmissions, 0 means no retransmission

Returns

Returns	Represents
0	Success
non 0	Busy

Additional Declaration

- The function is declared in the LoRaWan_api_v1.h file ;
- This function is recommended to be used when ABP or OTAA join accept .
- The maximum payload length of a data packet varies with different spreading factors, exceeding this value will result in error

SF	Maximum payload of a data packet bytes)
SF12	60
SF11	73
SF10	115
SF9	242
SF8	242
SF7	242

Example

```
uint8_t buf[10] = {"hello lora"}

LoRaWanSendBuf( 1, buf, 10, 0);
```

LoRaWanSendLinkCheckReq

Define

```
bool LoRaWanSendLinkCheckReq( void );
```

Declaration

The device sends a LoRaWAN MAC command LinkCheckReq to check whether the device is in the network or already joined

The MAC command will be send next time when a data frame is sent(either manually or automatically).

Parameters

NULL

Returns

Successfully return 1 , failure return 0.

Additional Declaration

The function is declared in the LoRaWan_api_v1.h file ;

Example

```
LoRaWanSendLinkCheckReq( );//send a LinkCheckReq
```