

文档名	SENSORO SWC11X-S SDK API Reference Manual
版本	
模板编号	
页	

SENSORO SWC11X-S SDK API Reference Manual

SENSORO

产品名	SWC11X-S	日期	2018-01-10
产品编号		日期	
作者		日期	
审核		日期	

修订历史

日期	版本	描述	作者
2018-01-10	1.0	文档创建	Huayuan

SENSORO

目录

修订历史.....	2
目录	3
1. 简介	9
2. 默认频段参数	9
2.1. CN470.....	9
2.1.1. 默认信道	9
2.1.2. Datarate	10
2.1.3. TxPower	10
2.1.4. 默认参数	11
2.2. EU433.....	11
2.2.1. 默认信道	11
2.2.2. Datarate	12
2.2.3. TxPower	12
2.2.4. 默认参数	13
2.3. EU868.....	13
2.3.1. 默认信道	13
2.3.2. Datarate	14

SENSORO

2.3.3.	TxPower	14
2.3.4.	默认参数	15
2.4.	CN779.....	15
2.4.1.	默认信道	15
2.4.2.	Datarate	16
2.4.3.	TxPower	16
2.4.4.	默认参数	17
2.5.	US915_Hybrid.....	17
2.5.1.	默认信道	17
2.5.2.	Datarate	18
2.5.3.	TxPower	18
2.5.4.	默认参数	19
2.6.	AS923	19
2.6.1.	默认信道	19
2.6.2.	Datarate	20
2.6.3.	TxPower	20
2.6.4.	默认参数	21
2.7.	IN865.....	21
2.7.1.	默认信道	21
2.7.2.	Datarate	22

SENSORO

2.7.3.	TxPower	22
2.7.4.	默认参数	23
3.	APIs.....	24
3.1.	LoRa API	24
3.1.1.	LoRa API list.....	24
3.1.2.	LoRaFSM	27
3.1.3.	LoRaInit.....	27
3.1.4.	LoRaJoin	28
3.1.5.	LoRaNetworkJoinStatusGet	28
3.1.6.	LoRaUnconfirmedPacketSend.....	29
3.1.7.	LoRaConfirmedPacketSend.....	30
3.1.8.	LoRaDevEUISet.....	31
3.1.9.	LoRaDevEUIGet.....	31
3.1.10.	LoRaAppEUISet.....	32
3.1.11.	LoRaAppEUIGet	32
3.1.12.	LoRaAppKeySet.....	33
3.1.13.	LoRaAppKeyGet	33
3.1.14.	LoRaAppSKeySet.....	34
3.1.15.	LoRaAppSKeyGet	34
3.1.16.	LoRaNwkSKeySet	35

SENSORO

3.1.17.	LoRaNwkSKeyGet	35
3.1.18.	LoRaDevAddrSet.....	36
3.1.19.	LoRaDevAddrGet	36
3.1.20.	LoRaDeviceClassSet	37
3.1.21.	LoRaDeviceClassGet	37
3.1.22.	LoRaActivationSet.....	38
3.1.23.	LoRaActivationGet	38
3.1.24.	LoRaTxDatarateSet	39
3.1.25.	LoRaTxDatarateGet.....	39
3.1.26.	LoRaTxPowerSet	40
3.1.27.	LoRaTxPowerGet	40
3.1.28.	LoRaAdrStatusSet	41
3.1.29.	LoRaAdrStatusGet	41
3.1.30.	LoRaDutyCycleStatusSet.....	42
3.1.31.	LoRaDutyCycleStatusGet	42
3.1.32.	LoRaRx2FrequencySet	43
3.1.33.	LoRaRx2FrequencyGet	43
3.1.34.	LoRaRx2DatarateSet	44
3.1.35.	LoRaRx2DatarateGet	44
3.1.36.	LoRaRx1DelaySet	45

SENSORO

3.1.37.	LoRaRx1DelayGet	45
3.1.38.	LoRaRx2DelayGet	46
3.1.39.	LoRaJoinRx1DelaySet.....	46
3.1.40.	LoRaJoinRx1DelayGet.....	47
3.1.41.	LoRaJoinRx2DelayGet.....	47
3.1.42.	LoRaUplinkCounterSet	48
3.1.43.	LoRaUplinkCounterGet.....	48
3.1.44.	LoRaDownlinkCounterSet.....	49
3.1.45.	LoRaDownlinkCounterGet	49
3.1.46.	LoRaChannelMaskSet	50
3.1.47.	LoRaChannelMaskGet	51
3.1.48.	LoRaMaxNbChannelsGet.....	52
3.1.49.	LoRaChannelsGet	52
3.1.50.	LoRaChannelAdd	53
3.1.51.	LoRaChannelRemove.....	54
3.1.52.	LoRaWANCertif.....	55
3.1.53.	LoRaWANCertifStop	55
3.2.	RunTime API	56
3.2.1.	RunTime API list	56
3.2.2.	RunTimeInit	57

SENSORO

3.2.3. RunTimeGet.....	57
3.2.4. RunTimeElapsedGet.....	58

SENSORO

1. 简介

本文描述了 SWC11X-S_SDK LoRa 库的相关信息及 API

2. 默认频段参数

2.1. CN470

2.1.1. 默认信道

信道	上行	下行	DR Range
80	486.3	506.7	DR0 - DR5
81	486.5	506.9	DR0 - DR5
82	486.7	507.1	DR0 - DR5
83	486.9	507.3	DR0 - DR5
84	487.1	507.5	DR0 - DR5
85	487.3	507.7	DR0 - DR5
86	487.5	507.9	DR0 - DR5
87	487.7	508.1	DR0 - DR5
RX2		505.3	DR0

- SDK 支持标准 CN470 的 96 个信道，默认开启信道 80~87。启用或关闭其他信道，详见 API [3.1.47 LoRaChannelMaskGet](#)

SENSORO

2.1.2. Datarate

DR	配置	Bit rate[bit/s]
0	SF12 / 125 kHz	250
1	SF11 / 125 kHz	440
2	SF10 / 125 kHz	980
3	SF9 / 125 kHz	1760
4	SF8 / 125 kHz	3215
5	SF7 / 125 kHz	5470

2.1.3. TxPower

TxPower	配置
0	MaxEIRP
1	MaxEIRP – 2dB
2	MaxEIRP – 4dB
3	MaxEIRP – 6dB
4	MaxEIRP – 8dB
5	MaxEIRP – 10dB
6	MaxEIRP – 12dB
7	MaxEIRP – 14dB

SENSORO

2.1.4. 默认参数

参数	值
TX Datarate	DR3
TX Power	2
MaxEIRP	17 dBm
RX1 delay	1000 ms
RX2 delay	2000 ms
Join RX1 delay	5000 ms
Join RX2 delay	6000 ms
ADR Enabled	False

2.2. EU433

2.2.1. 默认信道

信道	上行	下行	DR Range
[0]	433.175	433.175	DR0 - DR5
[1]	433.375	433.375	DR0 - DR5
[2]	433.575	433.575	DR0 - DR5
3	433.775	433.775	DR0 - DR5
4	433.975	433.975	DR0 - DR5
5	434.175	434.175	DR0 - DR5
6	434.375	434.375	DR0 - DR5
7	434.575	434.575	DR0 - DR5
RX2	--	434.665	DR0

SENSORO

2.2.2. Datarate

DR	配置	Bit rate[bit/s]
0	SF12 / 125 kHz	250
1	SF11 / 125 kHz	440
2	SF10 / 125 kHz	980
3	SF9 / 125 kHz	1760
4	SF8 / 125 kHz	3215
5	SF7 / 125 kHz	5470
6	SF7 / 250 kHz	11000
7	FSK 50 kbps	50000

2.2.3. TxPower

TxPower	配置
0	MaxEIRP
1	MaxEIRP – 2dB
2	MaxEIRP – 4dB
3	MaxEIRP – 6dB
4	MaxEIRP – 8dB
5	MaxEIRP – 10dB

SENSORO

2.2.4. 默认参数

参数	值
TX Datarate	DR0
TX Power	0
MaxEIRP	12 dBm
RX1 delay	1000 ms
RX2 delay	2000 ms
Join RX1 delay	5000 ms
Join RX2 delay	6000 ms
ADR Enabled	False

2.3. EU868

2.3.1. 默认信道

信道	上行	下行	DR Range
[0]	868.1	868.1	DR0 - DR5
[1]	868.3	868.3	DR0 - DR5
[2]	868.5	868.5	DR0 - DR5
3	867.1	867.1	DR0 - DR5
4	867.3	867.3	DR0 - DR5
5	867.5	867.5	DR0 - DR5
6	867.7	867.7	DR0 - DR5
7	867.9	867.9	DR0 - DR5
RX2	--	869.525	DR0

SENSORO

2.3.2. Datarate

DR	配置	Bit rate[bit/s]
0	SF12 / 125 kHz	250
1	SF11 / 125 kHz	440
2	SF10 / 125 kHz	980
3	SF9 / 125 kHz	1760
4	SF8 / 125 kHz	3215
5	SF7 / 125 kHz	5470
6	SF7 / 250 kHz	11000
7	FSK 50 kbps	50000

2.3.3. TxPower

TxPower	配置
0	MaxEIRP
1	MaxEIRP – 2dB
2	MaxEIRP – 4dB
3	MaxEIRP – 6dB
4	MaxEIRP – 8dB
5	MaxEIRP – 10dB
6	MaxEIRP – 12dB
7	MaxEIRP – 14dB

SENSORO

2.3.4. 默认参数

参数	值
TX Datarate	DR0
Tx Power	1
MaxEIRP	16 dBm
Rx1 delay	1000 ms
RX2 delay	2000 ms
Join RX1 delay	5000 ms
Join RX2 delay	6000 ms
ADR Enabled	False

2.4. CN779

2.4.1. 默认信道

信道	上行	下行	DR Range
[0]	779.5	779.5	DR0 - DR5
[1]	779.7	779.7	DR0 - DR5
[2]	779.9	779.9	DR0 - DR5
3	780.1	780.1	DR0 - DR5
4	780.3	780.3	DR0 - DR5
5	780.5	780.5	DR0 - DR5
6	780.7	780.7	DR0 - DR5
7	780.9	780.9	DR0 - DR5
RX2	--	786	DR0

SENSORO

2.4.2. Datarate

DR	配置	Bit rate[bit/s]
0	SF12 / 125 kHz	250
1	SF11 / 125 kHz	440
2	SF10 / 125 kHz	980
3	SF9 / 125 kHz	1760
4	SF8 / 125 kHz	3215
5	SF7 / 125 kHz	5470
6	SF7 / 250 kHz	11000
7	FSK 50 kbps	50000

SENSORO

2.4.3. TxPower

TxPower	配置
0	MaxEIRP
1	MaxEIRP – 2dB
2	MaxEIRP – 4dB
3	MaxEIRP – 6dB
4	MaxEIRP – 8dB
5	MaxEIRP – 10dB

SENSORO

2.4.4. 默认参数

参数	值
TX Datarate	DR0
TX Power	0
MaxEIRP	12 dBm
RX1 delay	1000 ms
RX2 delay	2000 ms
Join RX1 delay	5000 ms
Join RX2 delay	6000 ms
ADR Enabled	False

2.5. US915_Hybrid

2.5.1. 默认信道

信道	上行	DR Range	下行	DR Range
0	902.3	DR0 - DR3	923.3	DR8 - DR13
1	902.5	DR0 - DR3	923.9	DR8 - DR13
2	902.7	DR0 - DR3	924.5	DR8 - DR13
3	902.9	DR0 - DR3	925.1	DR8 - DR13
4	903.1	DR0 - DR3	925.7	DR8 - DR13
5	903.3	DR0 - DR3	926.3	DR8 - DR13
6	903.5	DR0 - DR3	926.9	DR8 - DR13
7	903.7	DR0 - DR3	927.5	DR8 - DR13
RX2	--	--	923.3	DR8

SENSORO

2.5.2. Datarate

DR	配置	Bit rate[bit/s]	方向
0	SF10 / 125 kHz	980	Uplink
1	SF9 / 125 kHz	1760	Uplink
2	SF8 / 125 kHz	3215	Uplink
3	SF7 / 125 kHz	5470	Uplink
4	SF8 / 500 kHz	12500	Uplink
5 : 7	RFU		
8	SF12 / 500 kHz	980	Downlink
9	SF11 / 500 kHz	1760	Downlink

2.5.3. TxPower

TxPower	配置
0*	30 dBm
1*	28 dBm
2*	26 dBm
3*	24 dBm
4*	22 dBm
5	20 dBm
6	18 dBm
7	16 dBm
8	14 dBm
9	12 dBm
10	10 dBm

SENSORO

2.5.4. 默认参数

参数	值
TX Datarate	DR1
TX Power	5
RX1 delay	1000 ms
RX2 delay	2000 ms
Join RX1 delay	5000 ms
Join RX2 delay	6000 ms
ADR Enabled	False

- 硬件最大支持 20 dBm，若设置 TxPower 为 0~4，将以 20 dBm 进行 TX

2.6. AS923

2.6.1. 默认信道

信道	上行	下行	DR Range
[0]	923.2	923.2	DR0 - DR5
[1]	923.4	923.4	DR0 - DR5
2	923.6	923.6	DR0 - DR5
3	923.8	923.8	DR0 - DR5
4	924.0	924.0	DR0 - DR5
5	924.2	924.2	DR0 - DR5
6	924.4	924.4	DR0 - DR5
7	924.6	924.6	DR0 - DR5
RX2	--	923.2	DR2

SENSORO

2.6.2. Datarate

DR	配置	Bit rate[bit/s]
0	SF12 / 125 kHz	250
1	SF11 / 125 kHz	440
2	SF10 / 125 kHz	980
3	SF9 / 125 kHz	1760
4	SF8 / 125 kHz	3215
5	SF7 / 125 kHz	5470
6	SF7 / 250 kHz	11000
7	FSK 50 kbps	50000

2.6.3. TxPower

TxPower	配置
0	MaxEIRP
1	MaxEIRP – 2dB
2	MaxEIRP – 4dB
3	MaxEIRP – 6dB
4	MaxEIRP – 8dB
5	MaxEIRP – 10dB
6	MaxEIRP – 12dB
7	MaxEIRP – 14dB

SENSORO

2.6.4. 默认参数

参数	值
TX Datarate	DR0
TX Power	0
MaxEIRP	16 dBm
RX1 delay	1000 ms
RX2 delay	2000 ms
Join RX1 delay	5000 ms
Join RX2 delay	6000 ms
ADR Enabled	False

SENSORO

2.7. IN865

2.7.1. 默认信道

信道	上行	下行	DR Range
[0]	865.0625	865.0625	DR0 - DR5
[1]	865.4025	865.4025	DR0 - DR5
[2]	865.9850	865.9850	DR0 - DR5
RX2	--	866.550	DR2

SENSORO

2.7.2. Datarate

DR	配置	Bit rate[bit/s]
0	SF12 / 125 kHz	250
1	SF11 / 125 kHz	440
2	SF10 / 125 kHz	980
3	SF9 / 125 kHz	1760
4	SF8 / 125 kHz	3215
5	SF7 / 125 kHz	5470
6	RFU	RFU
7	FSK 50 kbps	50000

2.7.3. TxPower

TxPower	配置
0	MaxEIRP
1	MaxEIRP – 2dB
2	MaxEIRP – 4dB
3	MaxEIRP – 6dB
4	MaxEIRP – 8dB
5	MaxEIRP – 10dB
6	MaxEIRP – 12dB
7	MaxEIRP – 14dB
8	MaxEIRP – 16dB
9	MaxEIRP – 18dB
10	MaxEIRP – 20dB

2.7.4. 默认参数

参数	值
TX Datarate	DR0
TX Power	5
MaxEIRP	30 dBm
RX1 delay	1000 ms
RX2 delay	2000 ms
Join RX1 delay	5000 ms
Join RX2 delay	6000 ms
ADR Enabled	False

- 硬件最大支持 20 dBm，若设置 TxPower 为 0~4，将以 20 dBm 进行 TX

SENSORO

3. APIs

3.1. LoRa API

LoRa API 声明在 *lorawan.h* 里

3.1.1. LoRa API list

API	描述
LoRaFSM	状态机
LoRaInit	初始化
LoRaJoin	入网
LoRaNetworkJoinStatusGet	入网状态查询
LoRaUnconfirmedPacketSend	发送 Unconfirmed 数据包
LoRaConfirmedPacketSend	发送 Confirmed 数据包
LoRaDevEUISet	设置 Device EUI
LoRaDevEUIGet	查询 Device EUI
LoRaAppEUISet	设置 Application EUI
LoRaAppEUIGet	查询 Application EUI
LoRaAppKeySet	设置 Application key
LoRaAppKeyGet	查询 Application key
LoRaAppSKeySet	设置 Application session key
LoRaAppSKeyGet	查询 Application session key
LoRaNwkSKeySet	设置 Network session key
LoRaNwkSKeyGet	查询 Network session key
LoRaDevAddrSet	设置 Device address
LoRaDevAddrGet	查询 Device address
LoRaDeviceClassSet	设置 Device Class
LoRaDeviceClassGet	查询 Device Class

SENSORO

LoRaActivationSet	设置 入网方式
LoRaActivationGet	查询 入网方式
LoRaTxDatarateSet	设置 发送 Datarate
LoRaTxDatarateGet	查询 发送 Datarate
LoRaTxPowerSet	设置 TxPower
LoRaTxPowerGet	查询 TxPower
LoRaAdrStatusSet	设置 ADR 状态
LoRaAdrStatusGet	查询 ADR 状态
LoRaDutyCycleStatusSet	设置 Duty cycle 状态
LoRaDutyCycleStatusGet	查询 Duty cycle 状态
LoRaRx2FrequencySet	设置 Rx2 频率
LoRaRx2FrequencyGet	查询 Rx2 频率
LoRaRx2DatarateSet	设置 Rx2 Datarate
LoRaRx2DatarateGet	查询 Rx2 Datarate
LoRaRx1DelaySet	设置 Rx1 Delay
LoRaRx1DelayGet	查询 Rx1 Delay
LoRaRx2DelayGet	查询 Rx2 Delay
LoRaJoinRx1DelaySet	设置 Join Rx1 Delay
LoRaJoinRx1DelayGet	查询 Join Rx1 Delay
LoRaJoinRx2DelayGet	查询 Join Rx2 Delay
LoRaUplinkCounterSet	设置 Uplink counter
LoRaUplinkCounterGet	查询 Uplink counter
LoRaDownlinkCounterSet	设置 Downlink counter
LoRaDownlinkCounterGet	查询 Downlink counter
LoRaChannelMaskSet	设置 Channel mask
LoRaChannelMaskGet	查询 Channel mask
LoRaMaxNbChannelsGet	查询 最大信道数量

SENSORO

LoRaChannelsGet	查询 信道参数
LoRaChannelAdd	新增信道或修改信道参数
LoRaChannelRemove	移除信道
LoRaSNRGet	查询 上一条接收到的数据的 SNR
LoRaRSSIGet	查询 上一条接收到的数据的 RSSI
LoRaWANCertif	进入认证模式
LoRaWANCertifStop	退出认证模式

SENSORO

SENSORO

3.1.2. LoRaFSM

运行状态机

- 原型:

```
void LoRaFSM(void)
```

- 参数: 无
- 返回值: 无

3.1.3. LoRaInit

初始化函数

- 原型:

```
void LoRaInit(LoRaRegion_t region, LoRaCallback_t *callbacks)
```

- 参数:
region
[in] LoRa 频段, 可选值:

频段	值
CN470	LORA_REGION_CN470
CN779	LORA_REGION_CN779
EU433	LORA_REGION_EU433
EU868	LORA_REGION_EU868
US915_HYBRID	LORA_REGION_US915_HYBRID
AS923	LORA_REGION_AS923
IN865	LORA_REGION_IN865

SENSORO

callbacks:

[in] 回调函数结构体

- 返回值：无

3.1.4. LoRaJoin

- 原型:

```
LoRaStatus_t LoRaJoin(uint32_t maxTrials)
```

- 参数:

maxTrials

[in] 最大发送入网请求次数

当 **Activation** 方式为 **OTAA** 时，若发送入网请求没收到正确回复，则会重发，总共最多发送 **maxTrials** 次入网请求。当 **maxTrials** 等于 0 时，API 自动加 1

当 **Activation** 方式为 **ABP** 时，API 忽略此参数

- 返回值:

LORA_STATUS_OK: 成功

LORA_STATUS_BUSY: 失败，上一次入网程序尚未结束

3.1.5. LoRaNetworkJoinStatusGet

查询入网状态

- 原型:

```
bool LoRaNetworkJoinStatusGet(void);
```

- 参数：无

SENSORO

- 返回值：
true: 已入网
false: 未入网

3.1.6. LoRaUnconfirmedPacketSend

发送 Unconfirmed 数据包

- 原型:

```
LoRaStatus_t LoRaUnconfirmedPacketSend(uint8_t port, uint8_t *pData, uint16_t dataSize);
```

- 参数:
port
[in] 数据端口, 取值范围 1 ~ 223
pData
[in] 待发送的数据
dataSize
[in] 待发送的数据长度
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次发送尚未结束
LORA_STATUS_PARAMETER_INVALID: 失败, 非法参数

SENSORO

3.1.7. LoRaConfirmedPacketSend

发送数据

- 原型:

```
LoRaStatus_t LoRaConfirmedPacketSend(uint8_t port, uint8_t maxTrials, uint8_t * pData,
uint16_t dataSize);
```

- 参数:

port

[in] 数据端口, 取值范围 1 ~ 223

maxTrials

[in] 最大发送次数

终端发送 Confirmed 数据包, 期望 Server 端响应 ACK, 若终端没有收到 ACK, 则重发, 总共最多发送 maxTrials 次。maxTrials 等于 0 时, API 自动加 1

pData

[in] 待发送的数据

dataSize

[in] 待发送的数据长度

- 返回值:

LORA_STATUS_OK: 成功

LORA_STATUS_BUSY: 失败, 上一次发送尚未结束

LORA_STATUS_PARAMETER_INVALID: 失败, 非法参数

SENSORO

3.1.8. LoRaDevEUISet

设置 Device EUI

- 原型:

```
LoRaStatus_t LoRaDevEUISet(uint8_t eui[8]);
```

- 参数:
eui[8]
[in] 8 字节的 EUI
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_STATE_INVALID: 失败, 设备已入网

3.1.9. LoRaDevEUIGet

查询 Device EUI

- 原型:

```
uint8_t *LoRaDevEUIGet(void);
```

- 参数: 无
- 返回值:
设备当前的 Device EUI

SENSORO

3.1.10. LoRaAppEUISet

设置 Application EUI

- 原型:

```
LoRaStatus_t LoRaAppEUISet(uint8_t eui[8]);
```

- 参数:
eui[8]
[in] 8 字节的 EUI
- 返回值: 无
LORA_STATUS_OK: 成功
LORA_STATUS_STATE_INVALID: 失败, 设备已入网

3.1.11. LoRaAppEUIGet

查询 Application EUI

- 原型:

```
uint8_t *LoRaAppEUIGet(void);
```

- 参数: 无
- 返回值:
设备当前的 Application EUI

3.1.12. LoRaAppKeySet

设置 Application key

- 原型:

```
LoRaStatus_t LoRaAppKeySet(uint8_t key[16]);
```

- 参数:
 - key[16]
[in] 16 字节的 key
- 返回值:
 - LORA_STATUS_OK: 成功
 - LORA_STATUS_STATE_INVALID: 失败, 设备已入网

3.1.13. LoRaAppKeyGet

查询 Application key

- 原型:

```
uint8_t *LoRaAppKeyGet(void);
```

- 参数: 无
- 返回值:
 - 设备当前的 Application key

3.1.14. LoRaAppSKeySet

设置 Application session key

注意，在 OTAA 激活方式下，LoRaWAN 网络会给设备重新分配 AppSKey，NwkSKey，DevAddr，通过此 API 设置的 AppSKey 会被覆盖

- 原型：

```
LoRaStatus_t LoRaAppSKeySet(uint8_t key[16]);
```

- 参数：
key[16]
[in] 16 字节的 key
- 返回值：
LORA_STATUS_OK: 成功
LORA_STATUS_STATE_INVALID: 失败，设备已入网

3.1.15. LoRaAppSKeyGet

查询 Application session key

- 原型：

```
uint8_t *LoRaAppSKeyGet(void);
```

- 参数：无
- 返回值：
设备当前的 Application session key

3.1.16. LoRaNwkKeySet

设置 Network session key

注意，在 OTAA 激活方式下，LoRaWAN 网络会给设备重新分配 AppSKey，NwkSKey，DevAddr，通过此 API 设置的 NwkSKey 会被覆盖

- 原型：

```
LoRaStatus_t LoRaNwkKeySet(uint8_t key[16]);
```

- 参数：
key[16]
[in] 16 字节的 key
- 返回值：
LORA_STATUS_OK: 成功
LORA_STATUS_STATE_INVALID: 失败，设备已入网

3.1.17. LoRaNwkKeyGet

查询 Network session key

- 原型：

```
uint8_t *LoRaNwkKeyGet(void);
```

- 参数：无
- 返回值：
设备当前的 Network session key

SENSORO

3.1.18. LoRaDevAddrSet

设置 Device address

注意，在 OTAA 激活方式下，LoRaWAN 网络会给设备重新分配 AppSKey，NwkSKey，DevAddr，通过此 API 设置的 DevAddr 会被覆盖

- 原型：

```
LoRaStatus_t LoRaDevAddrSet(uint32_t addr);
```

- 参数：
addr
[in] Device address
- 返回值：
LORA_STATUS_OK: 成功
LORA_STATUS_STATE_INVALID: 失败，设备已入网

3.1.19. LoRaDevAddrGet

查询 Device address

- 原型：

```
uint32_t LoRaDevAddrGet(void);
```

- 参数：无
- 返回值：
设备当前 Device address

SENSORO

3.1.20. LoRaDeviceClassSet

设置 Device class

- 原型:

```
LoRaStatus_t LoRaDeviceClassSet(DeviceClass_t device_class);
```

- 参数:
device_class
[in] LoRaWAN Class, 取值 CLASS_A, CLASS_C
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成

3.1.21. LoRaDeviceClassGet

查询 Device class

- 原型:

```
DeviceClass_t LoRaDeviceClassGet(void);
```

- 参数: 无
- 返回值:
设备当前 Class
默认值: CLASS_A

3.1.22. LoRaActivationSet

设置入网方式

如果设备处于已入网状态，修改不同的入网方式将使设备返回未入网状态

- 原型:

```
LoRaStatus_t LoRaActivationSet(ActivationType_t type);
```

- 参数:
type
[in] Activation type, 入网方式, 取值 ACTIVATION_ABP 或 ACTIVATION_OTAA
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成

3.1.23. LoRaActivationGet

查询设备入网方式

- 原型:

```
ActivationType_t LoRaActivationGet(void);
```

- 参数: 无
- 返回值:
设备当前入网方式
默认值: ACTIVATION_ABP

SENSORO

3.1.24. LoRaTxDataRateSet

设置发送 Datarate

- 原型:

```
LoRaStatus_t LoRaTxDataRateSet(uint32_t datarate);
```

- 参数:
datarate
[in] 发送 Datarate, 取值范围参考 [默认频段参数](#)
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成
LORA_STATUS_PARAMETER_INVALID: 失败, 非法参数

3.1.25. LoRaTxDataRateGet

查询发送 Datarate

- 原型:

```
uint32_t LoRaTxDataRateGet(void);
```

- 参数: 无
- 返回值:
设备当前发送 Datarate
默认值: 参考 [默认频段参数](#)

SENSORO

3.1.26. LoRaTxPowerSet

设置发射功率

注意：硬件最大支持 20dBm 的发射功率，根据 LoRaWAN 协议，某些频段，如 US915_HYRIDE，有大于 20dBm 的发射功率的配置，API 允许设置大于 20dBm 的配置，但硬件实际只以 20dBm 的功率工作

- 原型：

```
LoRaStatus_t LoRaTxPowerSet(uint32_t power);
```

- 参数：
power
[in] 发射功率，取值范围参考 [默认频段参数](#)
- 返回值：
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败，上一次数据发送尚未完成
LORA_STATUS_PARAMETER_INVALID: 失败，非法参数

3.1.27. LoRaTxPowerGet

查询发射功率

- 原型：

```
uint32_t LoRaTxPowerGet(void);
```

- 参数：无
- 返回值：
设备当前发射功率
默认值：参考 [默认频段参数](#)

SENSORO

3.1.28. LoRaAdrStatusSet

设置 ADR 状态

- 原型:

```
LoRaStatus_t LoRaAdrStatusSet(bool status);
```

- 参数:
status
[in] ADR 状态, true: Enable, false: disable
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成

3.1.29. LoRaAdrStatusGet

查询 ADR 状态

- 原型:

```
bool LoRaAdrStatusGet(void);
```

- 参数: 无
- 返回值:
设备当前 ADR 状态, true: Enable, false: Disable
默认值: false

SENSORO

3.1.30. LoRaDutyCycleStatusSet

设置 Duty cycle 状态，此参数仅应用于频段 EU433 和 EU868

- 原型:

```
LoRaStatus_t LoRaDutyCycleStatusSet(bool status);
```

- 参数: 无
status
[in] Duty cycle 状态, true: Enable, false: Disable
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成

3.1.31. LoRaDutyCycleStatusGet

查询 Duty cycle 状态

- 原型:

```
bool LoRaDutyCycleStatusGet(void);
```

- 参数: 无
- 返回值:
设备当前 Duty cycle 状态, true: Enable, false: Disable
默认值: false

SENSORO

3.1.32. LoRaRx2FrequencySet

设置接收窗口 2 的频率

- 原型:

```
LoRaStatus_t LoRaRx2FrequencySet(uint32_t freq);
```

- 参数:
freq
[in] 接收窗口 2 的频率
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成

3.1.33. LoRaRx2FrequencyGet

查询接收窗口 2 的频率

- 原型:

```
uint32_t LoRaRx2FrequencyGet(void);
```

- 参数: 无
- 返回值:
设备当前接收窗口 2 的频率
默认值: 参考 [默认频段参数](#)

3.1.34. LoRaRx2DatarateSet

设置接收窗口 2 的 Datarate

- 原型:

```
LoRaStatus_t LoRaRx2DatarateSet(uint32_t datarate);
```

- 参数:
datarate
[in] 接收窗口 2 的 Datarate, 取值范围参考 [默认频段参数](#)
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成
LORA_STATUS_PARAMETER_INVALID: 失败, 非法参数

3.1.35. LoRaRx2DatarateGet

查询接收窗口 2 的 Datarate

- 原型:

```
uint32_t LoRaRx2DatarateGet(void);
```

- 参数: 无
- 返回值:
当前设备接收窗口 2 的 Datarate
默认值: 参考 [默认频段参数](#)

SENSORO

3.1.36. LoRaRx1DelaySet

设置接收窗口 1 的 delay

- 原型:

```
LoRaStatus_t LoRaRx1DelaySet(uint32_t delay);
```

- 参数:
delay
[in] 接收窗口 1 的 delay, 取值范围 1000 ~ 15000 ms
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成
LORA_STATUS_PARAMETER_INVALID: 失败, 非法参数

3.1.37. LoRaRx1DelayGet

查询接收窗口 1 的 delay

- 原型:

```
uint32_t LoRaRx1DelayGet(void);
```

- 参数: 无
- 返回值:
当前设备接收窗口 1 的 delay
默认值: 1000 ms

SENSORO

3.1.38. LoRaRx2DelayGet

查询接收窗口 2 的 delay

$Rx2Delay = Rx1Delay + 1000$

- 原型:

```
uint32_t LoRaRx2DelayGet(void);
```

- 参数: 无
- 返回值:
当前设备接收窗口 2 的 delay
默认值: 2000 ms

3.1.39. LoRaJoinRx1DelaySet

设置入网接收窗口 1 的 delay

- 原型:

```
LoRaStatus_t LoRaJoinRx1DelaySet(uint32_t delay);
```

- 参数:
delay
[in] 入网接收窗口 1 的 delay, 取值范围 1000 ~ 15000 ms
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成
LORA_STATUS_PARAMETER_INVALID: 失败, 非法参数

3.1.40. LoRaJoinRx1DelayGet

查询入网接收窗口 1 的 delay

- 原型:

```
uint32_t LoRaJoinRx1DelayGet(void);
```

- 参数: 无
- 返回值:
当前设备入网接收窗口 1 的 delay
默认值: 5000 ms

3.1.41. LoRaJoinRx2DelayGet

查询入网接收窗口 2 的 delay

$\text{JoinRx2Delay} = \text{JoinRx1Delay} + 1000$

- 原型:

```
uint32_t LoRaJoinRx2DelayGet(void);
```

- 参数: 无
- 返回值:
当前设备入网接收窗口 2 的 delay
默认值: 6000 ms

3.1.42. LoRaUplinkCounterSet

设置上行 Counter 值

- 原型:

```
LoRaStatus_t LoRaUplinkCounterSet(uint32_t count);
```

- 参数:
count
[in] 上行 Counter 值
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成

3.1.43. LoRaUplinkCounterGet

查询上行 Counter 值

- 原型:

```
uint32_t LoRaUplinkCounterGet(void);
```

- 参数: 无
- 返回值:
当前设备上 行 Counter 值

3.1.44. LoRaDownlinkCounterSet

设置下行 Counter 值

- 原型:

```
LoRaStatus_t LoRaDownlinkCounterSet(uint32_t count);
```

- 参数:
count
[in] 下行 Counter 值
- 返回值:
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成

3.1.45. LoRaDownlinkCounterGet

- 原型:

```
uint32_t LoRaDownlinkCounterGet(void);
```

- 参数: 无
- 返回值:
当前设备下行 Counter 值

SENSORO

3.1.46. LoRaChannelMaskSet

设置 Channel mask，Channel mask 长度 16 位，每一位表示一个信道。某一位为 1 时，对应信道启用，否则不启用

- 原型：

```
LoRaStatus_t LoRaChannelMaskSet(uint16_t *pMask);
```

- 参数：

pMask

[in] Channel mask 数组，根据协议 *LoRaWAN Regional Parameter*，在频段 EU433, EU868, CN779, AS923, IN865 下，pMask 指向 1 个元素的数组，在频段 CN470 下，pMask 指向 6 个元素的数组

- 返回值：

LORA_STATUS_OK: 成功

LORA_STATUS_BUSY: 失败，上一次数据发送尚未完成

- 示例：

根据文档，CN470 ChannelMask 对应信道如下：

ChMaskCntl	ChMask applies to
0	Channels 0 to 15
1	Channels 16 to 31
2	Channels 32 to 47
3	Channels 48 to 63
4	Channels 64 to 79
5	Channels 80 to 95
6	All channels ON

SENSORO

现需要启用 16 ~ 31 信道：

```
LoRaStatus_t status;

uint16_t channelMask[6] =
{
    0x0000,
    0xFFFF,
    0x0000,
    0x0000,
    0x0000,
    0x0000,
};

status = LoRaChannelMaskSet(channelMask);
```

3.1.47. LoRaChannelMaskGet

查询 Channel mask

- 原型：

```
uint16_t * LoRaChannelMaskGet(void);
```

- 参数：无
- 返回值：
当前设备 Channel mask

SENSORO

3.1.48. LoRaMaxNbChannelsGet

查询最大信道数

- 原型:

```
uint32_t LoRaMaxNbChannelsGet(void);
```

- 参数: 无
- 返回值:
当前频段的最大信道数
默认值:
CN470: 96
CN779: 16
EU433: 16
EU868: 16
US915_HYBRID: 16
AS923: 16
IN865: 16

3.1.49. LoRaChannelsGet

查询信道参数

- 原型:

```
ChannelParams_t * LoRaChannelsGet(void);
```

- 参数: 无
- 返回值:
当前设备的信道参数数组

SENSORO

3.1.50. LoRaChannelAdd

新增信道或修改信道，成功后，相应的 Channel mask 位会置 1

- 原型：

```
LoRaStatus_t LoRaChannelAdd(uint32_t id, ChannelParams_t param);
```

- 参数：
param
[in] 信道参数
- 返回值：
LORA_STATUS_OK: 成功
LORA_STATUS_BUSY: 失败，上一次数据发送尚未完成
LORA_STATUS_PARAMETER_INVALID: 失败，非法参数
- 注意：
不允许修改的信道：

频段	不允许修改的信道
CN470	所有信道
CN770	信道 0, 1, 2
EU433	信道 0, 1, 2
EU868	信道 0, 1, 2
US915_HYBRID	所有信道
AS923	信道 0, 1
IN865	信道 0, 1, 2

SENSORO

3.1.51. LoRaChannelRemove

移除信道

- 原型:

```
LoRaStatus_t LoRaChannelRemove(uint32_t id);
```

- 参数:
 - id
[in] 要移除的信道 id
- 返回值:
 - LORA_STATUS_OK: 成功
 - LORA_STATUS_BUSY: 失败, 上一次数据发送尚未完成
- 注意:
 - 不允许移除的信道:

频段	不允许移除的信道
CN470	所有信道
CN770	信道 0, 1, 2
EU433	信道 0, 1, 2
EU868	信道 0, 1, 2
US915_HYBRID	所有信道
AS923	信道 0, 1
IN865	信道 0, 1, 2

3.1.52. LoRaWANCertif

进入 LoRaWAN 认证模式

- 原型:

```
void LoRaWANCertif(void);
```

- 参数: 无
- 返回值: 无

3.1.53. LoRaWANCertifStop

退出 LoRaWAN 认证模式

- 原型:

```
void LoRaWANCertifStop(void);
```

- 参数: 无
- 返回值: 无

3.2. RunTime API

API 声明在 RunTime.h 里

3.2.1. RunTime API list

API	描述
RunTimeInit	RunTime 模块初始化
RunTimeGet	当前时间戳获取
RunTimeElapsedGet	计算逝去的时间

3.2.2. RunTimeInit

RunTime 模块初始化

- 原型:

```
void RunTimeInit(void);
```

- 参数: 无
- 返回值: 无

3.2.3. RunTimeGet

获取当前的时间戳, 单位 ms, 以 MCU 启动为起点 (本模块基于 RTC, 严格来说, 是以 RTC 启动为起点)

- 原型:

```
uint32_t RunTimeGet(void);
```

- 参数: 无
- 返回值:
当前时间戳, 单位 ms

3.2.4. RunTimeElapsedGet

获取逝去的时间(过去某个时间点到当前时间的差), 单位 ms

- 原型:

```
uint32_t RunTimeElapsedGet(uint32_t past);
```

- 参数:
past
[in] 过去某个时间点
- 返回值:
当前时间与传入时间点的差值, 单位 ms

SENSORO