

青莲云 MCU SDK 接口说明文档

版本记录：

版本	编写/修订说明	修订人	修订日期	备注
2.0.0	创建文档	何朋涛	20171218	
3.0.0	增加用户绑定与解绑接口 增加用户获取和通知接口 数据传输接口支持子设备 增加子设备管理接口	何朋涛	20181217	

目录

1. 概要.....	4
2. SDK 初始化.....	5
2.1. 初始化数据结构.....	6
2.2. 初始化 SDK 功能函数.....	6
3. 系统功能接口.....	7
3.1. 模组初始化.....	7
3.2. 时间同步.....	7
3.3. 模组状态报告.....	8
3.4. 重置模组.....	8
3.5. 产测模式.....	8
3.6. 发送数据回调.....	9
3.7. 用户绑定.....	9
3.8. 用户解绑.....	9
3.9. 用户分享信息改变.....	10
3.10. 获取用户信息.....	10
4. 功能点数据传输接口.....	11
4.1. 上传功能点数据.....	11
4.2. 接收功能点数据.....	11
5. OTA 升级接口.....	13
5.1. 设置 OTA 属性.....	13

5.2.	接收 OTA 数据.....	14
5.3.	接收升级指令.....	14
6.	高级功能接口.....	15
6.1.	自定义数据上传.....	15
6.2.	自定义数据接收.....	15
7.	子设备功能.....	16
7.1.	子设备上线.....	16
7.2.	子设备离线.....	17

1. 概要

青连云作为物联网后端云服务，适配了 ESP8266 等模组，为了让开发者不必关心交互协议的细节，青连云提供了 MCU 与模组间通过串口等接口交互的 SDK，此 SDK 目前支持 Arduino 平台和其他 C 语言平台。开发者在对 MCU 编程时，只需要使用 SDK 提供的接口来收发数据，就可以通过模组连接青连云。本文档介绍青连云提供的 MCU SDK 收发数据接口的使用方法。SDK 以源码方式提供。该 SDK 提供以下功能：

- 模组初始化
- 时间同步
- 模组状态报告
- 重置模组
- 产测模式
- 数据回调处理
- 用户绑定与解绑
- 用户信息管理
- 功能点数据传输
- OTA 升级
- 自定义数据传输
- 子设备管理

模组联网和设备发现的功能，不需要用户做任何的实现，该两项功能在模组的 SDK 内部逻辑已经实现。同时，断开重连以及 AP 重启导致的连接重连都在模组 SDK 内部逻辑实现，不需要开发者做任何的额外开发。

MCU 与模组之间的交互方式主要是“一问一答”，MCU 通过串口向模组发送封装好的数据，模组处理后作出回复，MCU 收到模组的回复数据，解析后调用相应的回调函数。

2.1. 初始化数据结构

PRODUCT_ID	云平台生成, 4字节的无符号整型数字, 全网唯一产品ID
PRODUCT_KEY	云平台生成, 16字节的十六进制编码, 全网唯一产品密钥
MCU_VER	mcu 固件版本, "xx.xx", $0 \leq x \leq 9$

2.2. 初始化 SDK 功能函数

◆ 协议初始化

```
void protocol_init (void)
```

说明:

该函数用于初始化接收队列和回调函数, 请在串口接收启动前调用。

◆ 发送数据函数

```
void tx_buf (const uint8_t* pBuf,
            uint8_t Length)
```

说明:

此函数用于向模组发送数据, 需要用户实现该函数的功能, SDK 发送数据时会调用此函数。

参数	长度	说明
pBuf	n	待发送数据的指针
Length	4	数据长度

◆ 接收数据函数

```
void rx_byte (uint8_t Byte)
```

说明:

此函数用于接收模组发送到 MCU 的数据, 请在接收数据的接口中调用此函数, 将收到的每个字节传入。

参数	长度	说明
Byte	1	收到的每个字节
Length	4	数据长度

◆ 处理接收数据

```
void rx_device_frame (void)
```

说明:

此函数用于对收到的数据进行解析处理, 请在主循环函数中调用此接口。

3. 系统功能接口

MCU 在与云端通信之前，需要先对模组做一些初始化相关的工作，并且要等待模组进入正常联网状态，之后，才能进入正常使用云端功能。

3.1. 模组初始化

模组在初始化完成前，定时向 MCU 发送初始化请求，MCU 收到初始化请求报文后，需要发送携带 PRODUCT_ID、PRODUCT_KEY、MCU_VER 信息的回应报文。

◆ 接收初始化命令

```
Voidiot_init_cb ( uint8_t* ModuleMac ,  
                 uint8_t* ModuleVersion )
```

说明：

该函数在收到模组的初始化命令后，由 SDK 调用。发送返回信息的代码已经添加。

参数	长度	说明
ModuleMac	6	模组的 MAC 地址
ModuleVersion	5	模组的固件版本，xx.xx 格式

3.2. 时间同步

当 MCU 需要与云端同步时间时，可以主动发起时间同步请求。

◆ 发送时间请求

```
voidiot_get_onlinetime (void)
```

说明：

该函数用于向云端发送时间请求。

◆ 接收时间信息

```
voidiot_get_onlinetime_cb ( uint32_t timestamp )
```

说明：

该函数在收到云端返回的时间信息后，由 SDK 调用。

参数	长度	说明
timestamp	4	网络时间戳

3.3. 模组状态报告

```
voidiot_status_cb ( uint8_t dev_status ,  
                   uint32_t timestamp )
```

说明:

该函数在收到模组的状态报告后, 由 SDK 调用。

参数	长度	说明
dev_status	1	模组当前状态
timestamp	4	状态报告对应的时间戳

3.4. 重置模组

```
voidiot_reset_module ( uint8_t ResetMode )
```

说明:

该函数用于向模组发送重置命令。

参数	长度	说明
ResetMode	1	设置配网模式: 0x01: smartconfig 模式, 0x02: AP 模式 (暂不支持)

3.5. 产测模式

◆ 发送产测命令

```
voidiot_factory_test_mode(void)
```

说明: 该函数用于向模组发送产测命令。

◆ 接收产测结果

```
voidiot_factory_test_cb ( uint8_t TestResult,  
                          uint8_t SignalIntensity)
```

说明:

该函数在收到模组返回的产测结果后, 由 SDK 调用。

参数	长度	说明
TestResult	1	测试结果: 0x00:成功 0x01:未找到产测路由 0x02:模组扫描故障
SignalIntensity	1	信号强度, 0~100, 0 最差, 100 最强

3.6. 发送数据回调

◆ 接收数据回调

```
void iot_data_cb( uint32_t data_seq )
```

说明：该函数在收到模组返回的数据发送结果后，由 SDK 调用。

参数	长度	说明
data_seq	4	发送数据的序列号

3.7. 用户绑定

◆ 发送绑定命令

```
void iot_start_bind_user(uint32_t Timeout)
```

说明：用于设置设备允许被绑定。

参数	长度	说明
Timeout	4	允许绑定的超时时间

◆ 接收绑定结果

```
void iot_bind_cb (uint8_t result)
```

说明：该函数在收到模组返回的绑定结果后，由 SDK 调用。

参数	长度	说明
result	1	绑定结果： 0x00：设置允许绑定命令的回复 0x01：绑定成功 0x02：绑定失败（超时或设备未联网）

3.8. 用户解绑

◆ 发送解绑命令

```
void iot_unbind_user(uint32_t Timeout)
```

说明：该函数用于发送解绑命令。

参数	长度	说明
Timeout	4	解绑超时时间

◆ 接收解绑结果

```
void iot_unbind_cb ()
```

说明：该函数在收到模组返回的解绑成功结果后，由 SDK 调用。

3.9. 用户分享信息改变

◆ 用户分享信息改变回调

```
void iot_share_chg_cb (uint8_t result)
```

说明：该函数在收到模组发送的分享用户变化结果后，由 SDK 调用。

参数	长度	说明
result	1	分享用户变化情况： 0x00：分享用户增加 0x01：分享用户减少

3.10. 获取用户信息

◆ 发送获取用户信息命令

```
int32_t iot_get_info(INFO_TYPE_T InfoType)
```

说明：该函数用于发送获取用户信息指令。

参数	长度	说明
InfoType	4	获取用户信息类型： 0x00：主用户信息 0x01：分享用户信息 0x02：设备信息

◆ 接收用户信息回调

```
void iot_info_cb( INFO_TYPE_T info_type, void* info )
```

说明：该函数在收到模组返回用户信息数据后，由 SDK 调用。

参数	长度	说明
info_type	4	返回用户信息类型： 0x00：主用户信息 0x01：分享用户信息 0x02：设备信息
info		用户信息结构

4. 功能点数据传输接口

4.1. 上传功能点数据

◆ 添加上传数据

int32_t dp_up_add_int (uint8_t dpid, uint32_t value)

int32_t dp_up_add_bool (uint8_t dpid, uint8_t value)

int32_t dp_up_add_enum (uint8_t dpid, uint8_t value)

int32_t dp_up_add_float (uint8_t dpid, float32_t value)

int32_t dp_up_add_string (uint8_t dpid, const uint8_t* str, uint16_t str_len)

int32_t dp_up_add_fault (uint8_t dpid, const uint8_t* fault, uint16_t fault_len)

int32_t dp_up_add_binary (uint8_t dpid, const uint8_t* bin, uint16_t bin_len)

说明:

调用对应接口, 添加相应类型数据到发送队列

参数	长度	说明
dpid	1	数据点 ID, 由云平台生产
返回值	4	0: 添加成功 -1: 添加失败

◆ 发送上传数据

int32_t iot_upload_dps (const char sub_id[16], uint32_t* data_seq)

说明:

添加数据完成后, 调用该函数进行数据组包发送。每个数据包最多可包含 15 个数据点的数据。

参数:

参数	长度	说明
sub_id	16	自定义的子设备 id, 仅限字母数字组合
data_seq	4	传出参数, 数据包序号接收指针

4.2. 接收功能点数据

接收到云端下发的功能点数据后, SDK 解析协议报文, 根据功能点调用相应的处理函数。所有功能点处理函数由用户实现, 用户要根据功能点处理函数, 完成如下数据结构的初始化。

```
iot_download_dps_t iot_down_dps[] =
{
```

```
{ DP_ID_1, DP_TYPE_INT, dp_down_handle_xxx },
{ DP_ID_2, DP_TYPE_BOOL, dp_down_handle_xxx },
{ DP_ID_3, DP_TYPE_ENUM, dp_down_handle_xxx },
{ DP_ID_4, DP_TYPE_STRING, dp_down_handle_xxx },
{ DP_ID_5, DP_TYPE_FLOAT, dp_down_handle_xxx },
{ DP_ID_6, DP_TYPE_FAULT, dp_down_handle_xxx },
{ DP_ID_7, DP_TYPE_BIN, dp_down_handle_xxx }
};
typedef struct iot_download_dps
{
    uint8_t dpid;
    uint8_t dptype;
    dp_down_handle_t dp_down_handle;
}iot_download_dps_t;
```

说明:

dpid: 数据点 ID, 由云平台生成

dptype: 数据点类型

dp_down_handle: 数据点处理接口

5. OTA 升级接口

5.1. 设置 OTA 属性

◆ 发送设置命令

```
voidiot_ota_option_set ( uint16_t    expect_time ,  
                        uint8_t    chunk_size )
```

说明：

该函数用于设置 OTA 属性，由用户根据实际需要设置。

参数	长度	说明
chunk_size	1	每次传输块大小： 0x00:256B 0x01:512B 0x02:1024B
expect_time	2	用户期望的升级检测倒计时，单位秒，范围 120-3600

◆ 接收设置结果

```
voidiot_ota_option_set_cb ( uint8_t    Result ,  
                           uint16_t    RealTime )
```

说明：

该函数在收到模组返回的 OTA 设置回应后，由 SDK 调用。

参数	长度	说明
Result	1	设置结果： 0x00:成功 0x01:参数错误
RealTime	2	实际升级检测倒计时

5.2. 接收 OTA 数据

```
voidiot_ota_chunk_cb ( uint8_t chunk_is_last ,
                      uint32_t chunk_offset ,
                      uint16_t chunk_size ,
                      const uint8_t* chunk )
```

说明:

该函数在收到云端发送的 OTA 数据块后，由 SDK 调用。

参数	长度	说明
chunk_is_last	1	有效标志: 0x00:有效 0x01:有效且是最后一块
chunk_offset	4	本数据块相对固件块的偏移
chunk_size	2	数据块长度
chunk	4	指向数据块的指针

5.3. 接收升级指令

```
voidiot_ota_upgrade_cb (void)
```

说明:

该函数在收到云端发送的升级指令后，由 SDK 调用。

6. 高级功能接口

6.1. 自定义数据上传

◆ 发送上传数据

```
int32_t iot_tx_data ( const char sub_id[16], uint32_t* data_seq, const uint8_t* data,  
uint32_t data_len )
```

说明:

发送自定义数据接口。

参数	长度	说明
sub_id	16	自定义的子设备 id, 仅限字母数字组合
data_seq	4	接收序列号指针
data	4	自定义数据块指针
data_len	4	自定义数据长度

6.2. 自定义数据接收

```
void iot_rx_data_cb ( char sub_id[16], uint32_t data_timestamp, uint8_t* data, uint32_t  
data_len )
```

说明:

该函数在收到云端下发的自定义数据后由 SDK 调用。

参数	长度	说明
sub_id	16	自定义的子设备 id, 仅限字母数字组合
data_timestamp	4	报文发送时间戳
data	4	接收数据块
data_len	4	接收数据块长度

7. 子设备功能

7.1. 子设备上线

- ◆ 添加子设备信息到发送队列

```
int32_t sub_dev_add (const char sub_id[16], const char sub_name[32], const char
sub_version[5], uint16_t sub_type)
```

参数	长度	说明
sub_id	16	自定义的子设备 id, 仅限字母数字组合
sub_name	32	子设备名称, 仅限字母数字组合
sub_version	5	子设备固件版本, "xx.xx", 0≤x≤9
sub_type	2	子设备类型, 请与 APP 端自行约定
返回值	4	0 : 成功 -1: 失败

- ◆ 发送上线子设备上线

```
int32_t iot_sub_dev_active ( uint32_t* data_seq )
```

参数	长度	说明
data_seq	4	传出参数, 本条数据的序列号, 如果需要确定数据何时上传成功, 可记录此发送序列号, 与收到的进行对比。
返回值	4	0 : 成功; -1: 失败

7.2. 子设备离线

◆ 发送子设备离线

`int32_t iot_sub_dev_inactive (const char sub_id[16], uint32_t* data_seq)`

参数	长度	说明
sub_id	16	自定义的子设备 id, 仅限字母数字组合
data_seq	4	传出参数, 本条数据的序列号, 如果需要确定数据何时上传成功, 可记录此发送序列号, 与收到的进行对比。
返回值	4	0 : 成功; -1: 失败