



爱星物联
U-BASES CLOUD

MCU SDK 应用手册

V1.1.0

文件履历表

版本	日期	制定/修订内容	制定	核准
V1.1.0	2023.1.14	首次修订	卓灵敏	陈首高

目录

- 目录 3
- 1. 概述 4
 - 1.1 名词解释 4
 - 1.2 硬件资源要求 4
 - 1.3 MCU SDK 文件描述 5
 - 1.4 系统框架 5
- 2. 重点 API 描述 6
 - 2.1 MCU SDK 配置函数 6
 - 2.2 MCU SDK 数据接收函数 6
 - 2.3 MCU SDK 发送处理函数 6
- 3. MCU SDK 应用 8
 - 3.1 MCU SDK 的移植 8
 - 3.2 MCU SDK 信息配置 9
 - 3.3 设备配网 9
- 4. 附录 12

1. 概述

本文档主要是用于指导用户开发者如何使用 MCU SDK 低代码进行接入爱星云模组，并让用户开发者将 MCU SDK 直接添加至 MCU 工程代码上进行开发，实现空气净化器等项目的功能需求。该 MCU SDK 已搭建好串口协议通信，实现了设备产品信息注册、远程控制、设备状态上报等功能。用户开发者可根据该文档快速掌握 MCU SDK 的使用以及开发流程。

1.1 名词解释

名词	说明
MCU	微控制器，是设备控制芯片。用于运行所有业务需求程序的躯体。
低代码	可支持自动化配置，以较少代码集成必要功能的程序。 如 MCU SDK（可支持自动化配置及低代码量集成了串口通信协议）
MCU SDK	集成了串口通信协议，用户开发者可直接将该 SDK 添加至 MCU 工程代码上。用户只需调用低代码对应的 API，便可实现与模组的通信，并与云端通信。
RAM	MCU 芯片的片上内存，当 MCU 运行程序时，数据交互时需要用到的内存。
ROM	MCU 芯片上的 flash 内存，即 MCU 存储程序时所需的内存。

1.2 硬件资源要求

用户开发者在 MCU 工程上添加 MCU SDK 代码是，需给 MCU SDK 预留一定资源。若 MCU 资源不足时，用户可根据《串口协议文档》自行对接串口协议。

- RAM 至少：2KB
- ROM 至少：2KB

当用户开发者将 MCU SDK 添加至 MCU 工程上时，可详细阅读该文档的后续内容，了解 MCU SDK 上 API 的调用，以及如何根据该 SDK 实现业务需求的功能。

1.3 MCU SDK 文件描述

- MCU SDK 目录结构

```
inc
├── config.h
├── mcu_api.h
├── protocol.h
└── system.h
src
├── mcu_api.c
├── protocol.c
└── system.c
```

- 文件描述

文件	说明	文件是否可修改
config.h	该文件主要是配置产品信息、MCU 固件版本号、日志打印以及物模型数据等。	可修改
mcu_api.h	包含 MCU 可调用的函数声明。	不可修改
mcu_api.c	包含 MCU 可调用的函数定义，这些函数主要是实现与模组串口通信，以及其业务需求的功能。	不可修改
protocol.h	包含 MCU 可调用的函数声明，该文件提供 MCU 所需数据的函数声明。	可修改
protocol.c	该文件主要是实现物模型数据处理的函数定义或适配。该文件上的有些函数需有用户开发者自行适配完善，以达到所需功能。	可修改
system.h	该文件包含所有封装串口协议的函数声明	不可修改
system.c	该文件包含所有串口协议处理的函数定义	不可修改

1.4 系统框架



2. 重点 API 描述

此章节主要描述 MCU SDK 里重点的函数 API，指导用户开发者如何调用 MCU SDK 里的函数 API。通过该章节，用户开发者可了解如何初始化 MCU SDK，以及了解 MCU 如何与模组进行数据交互。该章节主要从 MCU SDK 的配置、MCU SDK 接收串口数据和 MCU SDK 发送串口数据这三个方向进行简要描述。

2.1 MCU SDK 配置函数

- UartProtocolInit()
 - MCU SDK 初始化, 主要是配置 MCU SDK 的接收缓冲区, 以及相关状态的初始化
 - 该函数必须在 MCU 启动后立即初始化, 如此方可接收并处理串口数据
 - 该函数是在 mcu_api.h 文件上声明。
- UartProcessPro()
 - MCU SDK 轮询处理函数, 该函数主要是校验缓冲区中数据是否符合协议格式。
 - 该函数必须以 100 - 200ms 的间隔循环调用。
 - 该函数是在 mcu_api.h 文件上声明。

2.2 MCU SDK 数据接收函数

- UartRevOneByte()
 - 该函数为串口接收函数, 即通过该函数, 可将串口数据存储于缓冲区中。
 - 当 MCU 的串口数据是以字节流的方式接收数据时, 可通过该函数处理字节流数据, 将数据接收于 MCU SDK 的缓冲区中处理。
 - 该函数是在 mcu_api.h 文件上声明。
- UartRevStream()
 - 该函数为串口接收函数, 即通过该函数, 可将串口数据存储于缓冲区中。
 - 当 MCU 的串口接收数据是以缓冲区的方式接收数据时, 可通过该函数处理串口缓冲区中的数据, 并将数据接收于 MCU SDK 的缓冲区中处理。
 - 该函数是在 mcu_api.h 文件上声明。
- 注意:
 - 当 MCU 需调用 MCU SDK 函数进行接收串口数据时, 对于 UartRevOneByte 和 UartRevStream 这两个函数需根据 MCU 串口数据处理择优选择, 但只能选择一个函数去处理串口数据。
 - 当用户开发者在使用 MCU SDK 时, 在处理串口数据之前, 需在 config.h 文件上配置串口缓冲区的大小。而 MCU SDK 默认的串口缓冲区大小为 1K。

2.3 MCU SDK 发送处理函数

- UartWriteData()
 - 该函数是串口发送函数, 需由用户开发者根据不同 MCU 进行适配发送接口。
 - 该函数主要是被 MCU SDK 中其他函数进行调用, 实现将数据发送到模组中。
 - 该函数已在 protocol.c 中定义了函数名, 以及在 protocol.h 中声明了函数。需由用户开发者补充适配串口发送接口。
- UartReportBoolTypeData() 和 UartSynReportBoolTypeData()
 - 这两个函数主要是处理 MCU 向模组上报 bool 类型的数据。

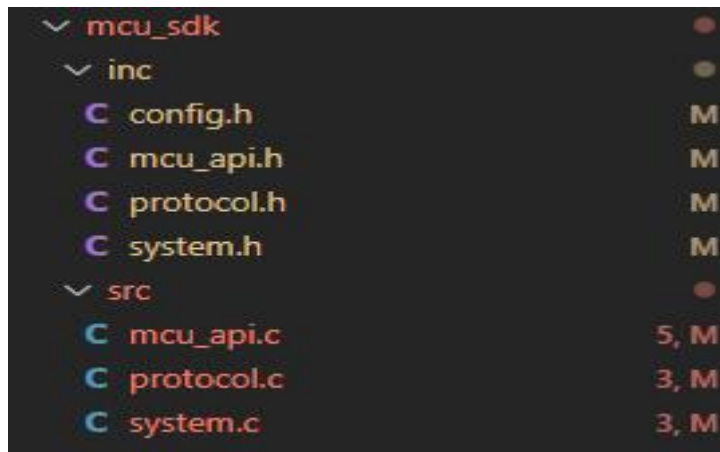
- UartReportBoolTypeData 这个函数是处理异步上报的，通过该函数上报，模组没有任何的回复。即 MCU 向模组上报频率不大的情况下，可通过该函数进行上报数据。
- UartSynReportBoolTypeData 这个函数是处理同步上报的，通过函数上报数据时，模组需将 MCU 返回处理情况。即 MCU 在短时间多次上报时，可通过该函数处理上，有利于数据传输的稳定性，以及模组有效处理数据。
- 这两个函数是在 mcu_api.h 文件上声明。
- UartReportValueTypeData() 和 UartSynReportValueTypeData()
 - 这两个函数主要是处理 MCU 向模组上报整数类型的数据。
 - UartReportValueTypeData 这个函数是处理异步上报的，通过该函数上报，模组没有任何的回复。即 MCU 向模组上报频率不大的情况下，可通过该函数进行上报数据。
 - UartSynReportValueTypeData 这个函数是处理同步上报的，通过函数上报数据时，模组需将 MCU 返回处理情况。即 MCU 在短时间多次上报时，可通过该函数处理上，有利于数据传输的稳定性，以及模组有效处理数据。
 - 这两个函数是在 mcu_api.h 文件上声明。
- UartReportStringTypeData() 和 UartSynReportStringTypeData()
 - 这两个函数主要是处理 MCU 向模组上报字符串类型的数据。
 - UartReportStringTypeData 这个函数是处理异步上报的，通过该函数上报，模组没有任何的回复。即 MCU 向模组上报频率不大的情况下，可通过该函数进行上报数据。
 - UartSynReportStringTypeData 这个函数是处理同步上报的，通过函数上报数据时，模组需将 MCU 返回处理情况。即 MCU 在短时间多次上报时，可通过该函数处理上，有利于数据传输的稳定性，以及模组有效处理数据。
 - 这两个函数是在 mcu_api.h 文件上声明。
- UartReportEnumTypeData() 和 UartSynReportEnumTypeData()
 - 这两个函数主要是处理 MCU 向模组上报枚举类型的数据。
 - UartReportEnumTypeData 这个函数是处理异步上报的，通过该函数上报，模组没有任何的回复。即 MCU 向模组上报频率不大的情况下，可通过该函数进行上报数据。
 - UartSynReportEnumTypeData 这个函数是处理同步上报的，通过函数上报数据时，模组需将 MCU 返回处理情况。即 MCU 在短时间多次上报时，可通过该函数处理上，有利于数据传输的稳定性，以及模组有效处理数据。
 - 这两个函数是在 mcu_api.h 文件上声明。
- PacketAllTypeData() 和 UartReportAllTypeData()
 - PacketAllTypeData 该函数是将所有类型的数据打包于数据缓冲区中，即可通过该函数将所有类型的数据暂存于数据缓冲区中。当调用 UartReportAllTypeData 该函数时，将会情况缓冲区中的数据。
 - UartReportAllTypeData 该函数是数据缓冲区中的数据通过串口发送函数发送给模组，并且清空数据缓冲区的函数。
 - 这两个函数是在 mcu_api.h 文件上声明。
 - 当调用 PacketAllTypeData 这个函数进行打包数据时，一定要调用 UartReportAllTypeData 进行数据处理。

3.MCU SDK 应用

3.1 MCU SDK 的移植

3.1.1 MCU SDK 文件的移植

- 在 MCU 项目工程中，完成 MCU 相关外设初始化，包括串口、定时器、GPIO 等。
- 将 MCU SDK 中的.c 和.h 文件添加至 MCU 项目工程中。添加后文件结构如下：



- 将 UartProtocolInit 这个 MCU SDK 初始化函数放于项目程序中进行初始化。
- 将 UartProcessPro 这个轮询函数放于项目程序的 while 循环中或是定时器中断函数中，进行循环调用该函数。
- 将 UartRevOneByte 或 UartRevStream 这两个函数中其中一个放于串口接收中断或是串口数据接收处，进行接收串口数据。

经过以上操作后，已成功将 MCU SDK 完全移植到 MCU 项目工程中。MCU SDK 亦可进行与模组初步的测试了。若是需进行产品项目功能测试，还需进行完成后续步骤。

3.1.2 MCU SDK 文件的修改

- 根据产品项目功能，需完成 protocol.c 文件上定义的函数控制功能。
如空气净化器项目上，需完善以下函数响应的执行动作：
ControlSwitchState: 开关控制
ControlModeState: 模式设置
ControlSpeedState: 风速设置
.....
ControlFilterResetState: 滤芯复位

3.2 MCU SDK 信息配置

3.2.1 产品信息设置

- 在爱星云平台上，创建好一个产品后，可获得一个产品 ID 和产品标识。产品 ID 和产品标识通过串口协议上传给模组后；模组将在配网模式下，将产品 ID 和产品标识按规则组装成热点名和 BLE 广播名。且产品 KEY 是模组在 MQTT 主题订阅和产品激活时都是需要用到的数据。
- 将产品 ID 和产品标识写在 MCU SDK 中的 config.h 文件上，如：

```
/* 开发平台创建产品后生成的产品ID，用户可根据创建不同产品而获得不同的产品ID，可在该处进行修改*/
#define PRODUCT_ID "PKhyQ4bI"

/* 产品标识，该标识是用户在创建不同产品时将自定义的产品标识 */
#define PRODUCT_FLAG "ZMXX"
```

- 注意：
在 config.h 上 PRODUCT_ID 和 PRODUCT_FLAG 这两个宏定义不可修改，只可修改其定义的产品 ID 内容。

3.2.2 MCU 软件版本设置

- MCU 软件版本主要是模组用于检查 MCU 固件是否需要 OTA 升级。
- 该软件版本协议 MCU SDK 中的 system.h 文件上，如：

```
/* MCU 软件版本号，用户可根据MCU版本而自定义该软件的版本号 */
#define MCU_SOFTWARE_VER "1.0.0"
```

- 注意：
 - a. 在 system.h 上的 MCU_SOFTWARE_VER 这个宏定义名不可修改，只可修改其定义的版本号。
 - b. 该版本的 MCU OTA 功能暂未支持。

3.3 设备配网

3.3.1 MCU 向模组重置网络

- 当 MCU 通过心跳包与模组建立连接后，MCU 可通过发送重置网络命令，让模组进入配网模式。
 - MCU 可通过 MCU SDK 中的 UartReportResetState 这个函数，向模组发送重置网络命令，让模组进入配网模式。
 - MCU 亦可通过串口发送以下命令数据，让模组进入配网模式。
5a a5 10 04 00 00 13

- 当模组进入配网模式后，需用爱星云 APP 进行对其配网。配网流程如下：
 - 当进入爱星云 APP 后，可通过点击添加设备而进入搜索设备。如下图：



- 进入搜索设备后，若是模组已进入配网模式，即可搜索到设备。

正在搜索附近的设备
请确保设备处于配网状态，扫描不到建议手动添加



- 当搜索到设备后，点击设备图标，即可进入配网。

正在搜索附近的设备
请确保设备处于配网状态，扫描不到建议手动添加



- APP 进入配网状态后，则需选择 WiFi 和输入 WiFi 的密码，点击下一步，APP 便对设备进行配网了。





- 注意：
 - 模组一共有两种配网方式，即 AP 配网和 BLE 配网。现模组默认进入配网模式时，AP 配网和 BLE 配网是处于共存的状态。
 - UartReportResetState 这个函数在 mcu_api.h 文件中声明。
 - 当 MCU 未与模组建立连接时，MCU 通过 UartReportResetState 这个函数向模组发送重置网络命令是无效的。
 - 当模组进入配网后，模组将会通过报告联网状态的命令告知 MCU，模组已进入哪种配网方式。当在配网过程中，亦会持续向 MCU 报告网络连接情况。

3.3.2 MCU 向模组指定配网方式

- MCU 向模组指定配网，该方式基本和重置网络的方式是一样的。其区别仅仅是 MCU 向模组发送不同的指令而已。
- 当 MCU 通过心跳包与模组建立连接后，MCU 可通过发送重置网络命令，让模组进入配网模式。
 - MCU 可通过 MCU SDK 中的 UartReportWifiConfig 这个函数，向模组发送重置网络命令，让模组进入配网模式。
 - ◆ 当通过 UartReportWifiConfig 设置参数为 0 时，模组即进入 BLE 配网模式。
 - ◆ 当通过 UartReportWifiConfig 设置参数为 1 时，模组即进入 AP 配网模式。
 - MCU 亦可通过串口发送以下命令数据，让模组进入配网模式。
 - ◆ 当 MCU 通过以下指令时，模组将进入 BLE 配网模式
5a a5 10 05 00 01 00 15
 - ◆ 当 MCU 通过以下指令时，模组将进入 AP 配网模式
5a a5 10 05 00 01 01 16
- 当模组进入配网模式后，需用爱星云 APP 进行对其配网。而 APP 配网流程可在 3.3.1 章节出查看。
- 注意：
 - 指定配网方式，现模组暂未进行选择进入指定配网方式。即 MCU 发送该选择指定配网指令时，模组仍然进入 AP 配网和 BLE 配网共存模式。
 - UartReportWifiConfig 这个函数在 mcu_api.h 文件中声明。

3.3.3 MCU 向模组指定路由连接

- 当用户开发者需测试模组的 WiFi 功能性时，可通过 WiFi 功能性测试指令进行测试模组连接 WiFi 的情况。
- 在 MCU SDK 上，可通过 UartReqConnectWifi 这个函数，向模组发送指定路由信息。模组将自行去连接指定路由。模组亦会将网络连接情况主动报告给 MCU。
- 注意：
 - UartReqConnectWifi 这个函数在 mcu_api.h 文件中声明。

4. 附录

- 当 MCU 通过心跳包与模组建立连接后，模组将会主动报告模组的网络状态。当模组首次启动后未进行配网过或是连接路由时，模组完成初始化后，将自动进入配网模式。
- 当模组已通过 APP 配网过或是以通过指定连接路由后，模组第二次启动时，将不会进入配网模式，而是直接进入连接之前连接过的路由。而需要重新配网时，可参考 3.3 章节的设备配网。
- 当模组进入配网后，模组将同时开启 AP 广播和 BLE 广播。当通过爱星云 APP 搜索不到设备时，可通过手机搜索模组开放的热点名、或是通过 BLE 测试工具查询 BLE 广播名，查看模组是否处于配网模式。模组的广播名格式如下：

```
axy_产品标识-产品 ID_p1_MAC 地址
```