

433 通用网关接入流程与协议

文件状态：	文件标识：	
<input checked="" type="checkbox"/> 草稿	当前版本：	1.0.0
<input type="checkbox"/> 正式发布	作者：	刘星辰
<input type="checkbox"/> 正在修改	完成日期：	2019.05.23



涂鸦智能

全球化智能平台



一键实现智能化



了解智能趋势

版本历史

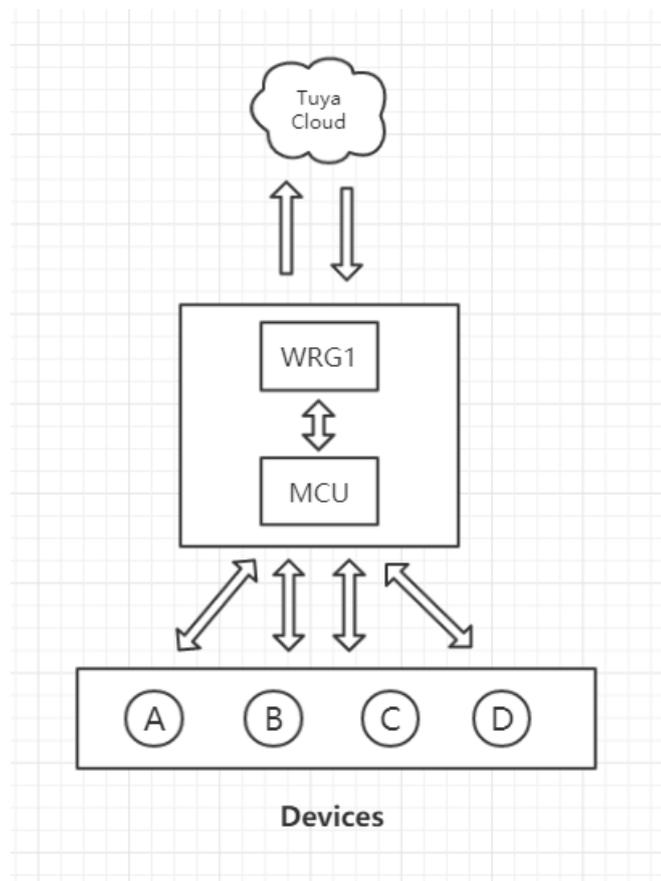
版本/状态	作者	参与者	审核	日期	备注
1.0.0	刘星辰			2019.05.23	

版权声明

本文档及其所包含内容均为涂鸦智能（杭州涂鸦信息技术有限公司）所有。除非经本公司书面授权，任何单位及个人不得擅自摘抄、编撰及复制本文档内容的部分或者全部，不得以任何形式（包括并不限于图片、链接、镜像等）进行传播，违者将被依法追究法律责任。

一、产品简介

此 433 网关通用固件基于 WRG1 模块开发，芯片平台 8711AM。涂鸦可提供 SDK 和对接协议给到客户做开发，客户需采购涂鸦的模块，对接自己设备的 MCU，使其设备具有网关的功能，能够连接涂鸦云，使用涂鸦 App，并可通过私有协议连接自己的子设备。

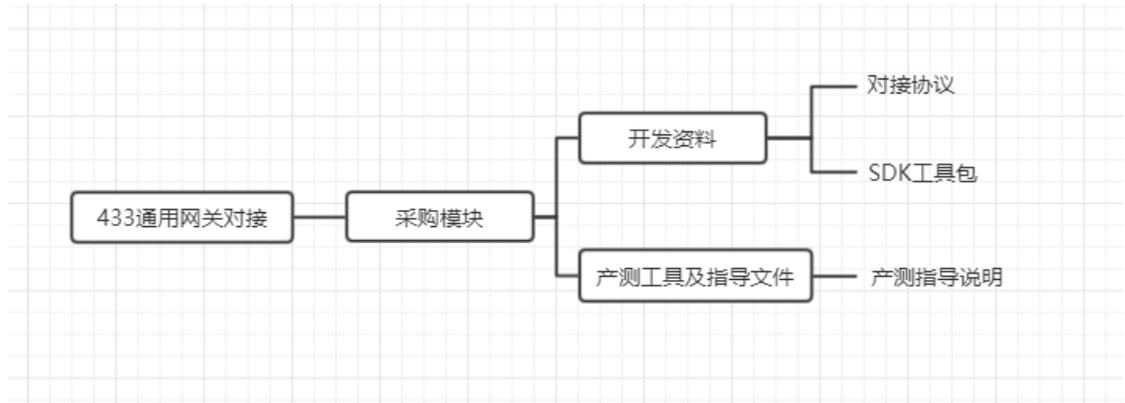


二、固件信息

固件 key : keyth3n5me5jqu95

三、合作方式

- 1、涂鸦提供：WRG1 模块、SDK 工具包、对接协议；
- 2、客户需一次性付出：云服务费+模块费用（默认连接 3 个子设备）；



四、项目对接流程

- 1、客户经理向客户介绍对接流与方式；
- 2、给客户寄送烧写好固件的 WRG1 模块，释放 SDK、对接协议文档；
- 3、客户根据自己需求开发，完成后与我方联调；
- 4、客户试产时提供产测工具和产测指导文档。

五、对接协议

1 串口通信约定

波特率：115200



数据位：8

奇偶校验：无

停止位：1

数据流控：无

MCU：用户控制板控制芯片，与涂鸦模块通过串口对接

2 帧格式说明

字段	长度 (byte)	说明
帧头	2	固定为 0x55aa
版本	1	升级扩展用
命令字	1	具体帧类型
数据长度	2	大端
数据	N	
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

说明：

3 基础协议详述

3.1 查询产品信息

说明：



- 1) product ID : 对应涂鸦开发者平台 PID(产品标识) , 由涂鸦云开发者平台生成 , 用于云端记录产品相关信息
- 2) 产品信息由 product ID、MCU 软件版本构成
- 3) MCU 软件版本号格式定义 采用点分十进制形式 , “ x.x.x” (0<=x<=99) , x 为十进制数

模块发送 :

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x01
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

例 : 0x55aa 00 01 0000 00

MCU 返回 :

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x01



数据长度	2	N
数据	N	{ "v" : " 1.0.0" , " m" : 1, " cap" :0}
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

例 : { "v" : " 1.0.0" , " m" :1, " cap" :0}

v 表示 MCU 版本为 1.0.0

m 表示配网方式: 0 默认配网、1 低功耗、2 特殊配网

cap 表示设备能力: bit0(本地群组) , bit1(本地场景) , bit2(网关有功能 dp), bit3(是 sigmesh)

例 1 : 设备支持本地群组 , cap 为 0x01:

例 2 : 设备既支持本地群组又支持本地场景 , cap 为 0x03;

例 3 : 设备不支持本地群组 , 本地场景 , 但是网关上有功能 dp. , cap 为 0x04

3.2 查询 MCU,设定模块的工作模式

说明 :

1) **模块工作模式主要针对如何指示 WIFI 的工作状态以及如何重置 WIFI 而言** ,

主要分两种情况 :

- a. MCU 与模块配合处理 , 即模块通过串口通知 MCU WIFI 当前的工作状态 , 由 MCU 提供显示支持 ; MCU 检测出 WIFI 的重置需求 , 通过串口通知模块重置 WIFI。



b. 模块自处理 :WIFI 模块的工作状态通过 WIFI 的 GPIO 引脚驱动 LED 状态显示 ; WIFI 重置通过检测 GPIO 输入需求处理。

模块自处理 WIFI 重置方法为 :WIFI 检测 GPIO 入口低电平持续 5s 以上触发 WIFI 重置。指示灯与按钮所使用的 GPIO 管脚由以下命令配置。

模块发送 :

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x02
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

例 : 0x55aa 00 02 0000 01

MCU 返回 :

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x02

数据长度	2	<p>0x0000/0x0004</p> <p>0x0000：指示模块工作为“MCU 与模块配合处理”模式，MCU 需实现上述“说明”中提及的功能。</p> <p>0x0004：指示模块工作为“模块自处理”模式。</p>
数据	0/2	<p>数据长度为 4：</p> <p>Data[0]:WIFI 指示灯 port,</p> <p>Data[1]: WIFI 指示灯 pin,</p> <p>Datat[2]:重置按键 port,</p> <p>Datat[3]:重置按键 pin,</p> <p>Port 范围 :0-3 依次代表 A-D</p> <p>Pin 范围 : 0-7</p>
校验和	1	<p>从帧头开始按字节求和得出的结果对 256 求余</p>

例：0x55aa 00 02 0000 04 (MCU 与模块配合处理)



0x55aa 00 02 0004 01030102 0C (模块自处理 , 指示灯 0x0103 : PB_3 ,
重置按钮 0x0102 : PB_2)

3.3 报告设备联网状态

设备联网状态	描述	状态值
状态 1	smartconfig 配置状态	0x00
状态 2	AP 配置状态	0x01
状态 3	WIFI 已配置但未连上路由器	0x02
状态 4	WIFI 已配置且连上路由器	0x03
状态 5	已连上路由器且连接到云端	0x04
状态 6	WIFI 设备处于低功耗模式	0x05

说明：

- 1) 设备联网状态 :1 smartconfig 配置状态 2 AP 配置状态 3 WIFI 配置成功但未连上路由器 4 WIFI 配置成功且连上路由器 5 设备连接到路由器且连接到云端。“模块自处理”工作模式相应的 LED 显示分别为 :1 间隔闪烁 250ms ; 2 间隔闪烁 1500ms ; 3 长暗状态 ; 4 或 5 长亮状态
- 2) 当模块检测到 MCU 重启或 MCU 断线再上线的过程，则主动下发 WIFI 状态至 MCU
- 3) 当模块的 WIFI 状态发生变化，则主动下发 WIFI 状态至 MCU
- 4) 如设置模块工作模式为“模块自处理”，则 MCU 无需实现该协议

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x03
数据长度	2	0x0001
数据	1	指示 WIFI 工作状态： 0x00:状态 1 0x01:状态 2 0x02:状态 3 0x03:状态 4 0x04:状态 5
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

例：0x55aa 00 03 0001 00 03

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00

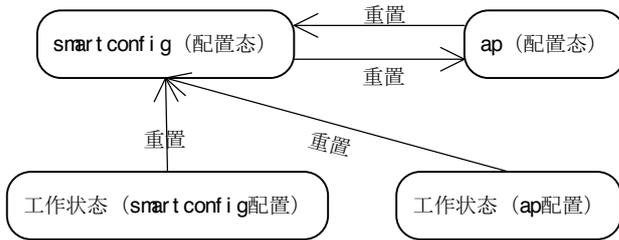
命令字	1	0x03
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

例：0x55aa 00 03 0000 02

3.4 重置 WIFI

说明：

1) 重置 WIFI 状态转化如下图所示：



模块自处理 WIFI 重置方法为：WIFI 检测 GPIO 入口低电平持续 5s 以上触发 WIFI 重置。

2) 如设置模块工作模式为“模块自处理”，则 MCU 无需实现该协议

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00



命令字	1	0x04
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

例：0x55aa 00 04 0000 03

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x04
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

例：0x55aa 00 04 0000 03

3.5 重置 WIFI-选择配置模式

说明：

- 1) 相对于“3.5”节“重置 WIFI 而言”，此帧提供 MCU 根据自身需求选择性

选择重置 WIFI 后的配置方式

- 2) MCU 接入用户可选择性的实现该协议
- 3) 如设置模块工作模式为“模块自处理”，则 MCU 无需实现该协议

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x05
数据长度	2	0x0001
数据	1	0x00 :进入 smartconfig 配置模式 0x01 :进入 AP 配置模式
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

例：0x55aa 00 05 0001 00 05，进入 smartconfig 配置模式

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00

命令字	1	0x05
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

例：0x55aa 00 05 0000 04

3.6 允许子设备入网

说明：

- 1) APP 发送允许入网指令给 GW，一定时间内允许子设备入网。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x06
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x06
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.7 关闭子设备入网

说明：

APP 发送关闭入网指令给 GW。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x07
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x07
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.8 子设备添加

说明：

- (1) MCU 发送子设备入网请求给 WIFI。
- (2) pk_type 备用字段，可不加。默认为 1

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x08
数据长度	2	N

<p>数据</p>	<p>“pk_type” :xx, “sub_id” :“ xxxx” “ , “pid” :“ xxxx” , “ver” :“ x.x.x”</p>	<p>用户产品分类 pk_type: 用户自定义产品类型，可不传。 子设备标识 sub_id: 一般为 mac 地址，芯片 id 等，需要保证唯一性.不可以是“ 0000” .“ 0000” 表示网关本身。 产品标识 pid : 产品 ID (ProductID) ，涂鸦云创建产品时生产版本号 ver : 如 : 1.0.0</p>
<p>校验和</p>	<p>1</p>	<p>从帧头开始按字节求和得出的结果对 256 求余</p>

模块返回：

字段	长度 (byte)	说明
<p>帧头</p>	<p>2</p>	<p>0x55aa</p>

版本	1	0x00
命令字	1	0x08
数据长度	2	0x0001
数据	1	0x00 接收请求 0x01 拒绝请求 (子设备实际是否入网成功，详见 3.19)
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

3.9 子设备删除

说明：

APP 发送关闭入网指令给 GW。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x09
数据长度	2	N
数据	{ "sub_id" : " xxxx"	要删除的子设备 ID

	}	
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x09
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

3.10 心跳检测

说明：

WIFI 模组定时 (1-3 分钟) 检测子设备在线状态。

模块发送：

字段	长度 (byte)	说明
----	-------------	----

帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0A
数据长度	2	N
数据	{ "sub_id" : "xxxx" }	sub_id : 子设备 ID
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0A
数据长度	2	N
数据	{ "sub_id" : "xxxx" , "hb_time" :xx }	sub_id : 子设备 ID 心跳间隔 hb_time (单 位 : 秒) : 如 : 60 表示 60 秒 , 0 不

	}	支持心跳，永久在线。若心跳间隔小于 30s 当 30s 处理（除 0 以外）
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

说明：

在线的设备需要回复心跳，连续 2 个心跳周期不回复则认为该设备离线。

3.11 状态查询

说明：

用于同步设子设备状态。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0B
数据长度	2	N
数据	0	无
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

MCU 返回：(3.13 状态上报)

3.12 命令下发

说明：

控制命令下发。

datapoint 命令状态单元：

1) datapoint 命令/状态数据单元如下所示：

数据段	长度 (byte)	说明			
dpid	1	datapoint 序号			
type	1	对应开放平台上某 datapoint 具体的数据类型,通过如下“表示值”标识			
		类型	表示值	长度(字节)	说明
		raw	0x00	N	对应于 raw 型 datapoint (模块透传)
		bool	0x01	1	value 范围：0x00/0x01
		value	0x02	4	对应 int 类型，大端表示
		string	0x03	N	对应于具体字符串
		enum	0x04	1	枚举类型，范围 0-255
		bitma	0x05	1/2/4	长度大于 1 字节时，大端表示



		p			
len	2	长度对应 value 的字节数			
value	1/2/4/N	hex 表示，大于 1 字节采用大端传输			

- 2) datapoint 命令/状态数据单元除“ raw” 类型外，其他类型均属于 “obj” 型 datapoint
- 3) “命令下发” 可含多个 datapoint “命令数据单元”
- 4) “命令下发” 为异步处理协议，对应于 MCU 的 datapoint “状态上报”

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0C
数据长度	2	N
数据	id_len(1byte) + sub_id(nbyte)+ datapoint	id_len: 子设备 id 长度 sub_id: 子设备 id 当 sub_id 为“ 0000” 时 ,代表这个 命令是下发网关本身的

		dp (主设备)。 datapoint: 控制命令数据点集合
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

3.13 状态上报

说明：

子设备受到命令下发后回复或状态变化后主动推送该指令。

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0D
数据长度	2	N
数据	id_len(1byte) + sub_id(nbyte)+ datapoint	id_len: 子设备 id 长度 sub_id: 子设备 id 当 sub_id

		为“0000”时,代表是上报网关本身的 dp (主设备)。 datapoint: 状态命令数据点集合
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

3.14 群组子设备加入 (可选)

说明：本地群组子设备加入，支持本地建组的设备。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0E
数据长度	2	N
数据	{ "gid" : " 01" ,	gid 表示群组 id cids : 要加入群组的子设

	<pre>"cids" :[" 12" ," 34" ," 56"], }</pre>	<p>备集合</p> <p>例：子设备 " 12" ," 34" ," 56" 要加入群组" 01"</p>
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x08
数据长度	2	N
数据	<pre>{ "gid" : " 12" , " cids" :[" 12" ," 34" ," 56"], "rets" :[0,0,0]; }</pre>	<p>gid 表示群组 id</p> <p>cids : 要加入群组的子设备集合</p> <p>返回结果结合 rets :</p> <p>0:添加成功</p> <p>1:超过群组数上限</p> <p>2:子设备超时</p>

		3:设置值超出范围 4:写文件错误 5:其他错误
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.15 群组子设备删除 (可选)

说明：本地群组子设备删除。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0F
数据长度	2	N
数据	<pre>{ "gid" : " 01" , "cids" : [" 12"], }</pre>	gid 表示群组 id cids : 要删除群组的子设备集合 例 : 子设备 " 12" 要从群组 " 01" 中删除

校验和	1	从帧头开始按字节求和得出的 结果对 256 求余
-----	---	-----------------------------

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x0F
数据长度	2	N
数据	{ "gid" : " 01" , "cids" : [" 12"], "rets" : [0]; }	gid 表示群组 id cids : 要加入群组的子设备集合 返回结果结合 rets : 0:删除成功 1:超过群组数上限 2:子设备超时 3:设置值超出范围 4:写文件错误 5:其他错误
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.16 获取系统时间(格林时间)

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x10
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

例：0x55aa 0010 0000 0F

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x10
数据长度	2	0x0007
数据	7	数据长度为 7 字节: Data[0]为是否获取时间 成功标志, 为 0 表示失败,

		<p>为 1 表示成功</p> <p>Data[1]为年份, 0x00 表示 2000 年</p> <p>Data[2]为月份, 从 1 开始到 12 结束</p> <p>Data[3]为日期, 从 1 开始到 31 结束</p> <p>Data[4]为时钟, 从 0 开始到 23 结束</p> <p>Data[5]为分钟, 从 0 开始到 59 结束</p> <p>Data[6]为秒钟, 从 0 开始到 59 结束</p>
校验和	1	<p>从帧头开始按字节求和得出的结果对 256 求余</p>

例 : 0x55aa 00 10 0007 01 10 04 13 05 06 07 50 (格林时间 2016 年 4 月 19 日 5 时 6 分 7 秒)

3.17 获取本地时间

MCU 发送 :

字段	长度 (byte)	说明
----	-------------	----

帧头	2	0x55aa
版本	1	0x00
命令字	1	0x11
数据长度	2	0x0000
数据	xxxx	无
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x11
数据长度	2	0x0008
数据	Data	数据长度为 8 字节： Data[0]为是否获取时间成功标志，为 0 表示失败，为 1 表示成功 Data[1]为年份，0x00 表示 2000 年 Data[2]为月份，从 1 开

		始到 12 结束 Data[3]为日期，从 1 开始到 31 结束 Data[4]为时钟，从 0 开始到 23 结束 Data[5]为分钟，从 0 开始到 59 结束 Data[6]为秒钟，从 0 开始到 15 结束 Data[7]为星期，从 1 开始到 7 结束，1 代表星期一
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

例：

(1)如设备在国内激活使用，则当地时间为北京时间(东 8 区)

0x55aa 00 11 0008 01 10 04 13 05 06 07 02 54 (北京时间 2016 年 4 月 19 日 5 时 6 分 7 秒)

(2)如果设备在国外激活使用，则当地时间为设备所处时区时间

3.18 批量添加设备

说明：批量添加同一个 pid 的多个子设备。

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x12
数据长度	2	N
数据	{ "pid" : " xxxx" , "cmds" : [" 12" , " 34" , " 56"], "ver" : " x.x.x" }	产品标识 pid : 产品 ID (ProductID) , 涂鸦云 创建产品时生产 cmds : 待添加子设备集合 版本号 ver : 如 : 1.0.0
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x12

数据长度	2	N
数据	1	00 表示接收数据成功 01 数据接收或者解析失败
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

3.19 返回添加设备结果

说明：设备添加完成后，通知 MCU 添加结果。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x13
数据长度	2	N
数据	{ "cmds" :[" 12" , " 34" , " 56"], "rets" :[0,0,0];	cids : 子设备集合 返回结果结合 rets : 0 : 成功 非 0 : 失败

	}	
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x13
数据长度	2	0000
数据	无	
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

3.20 控制群组指令下发 (可选)

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00

命令字	1	0x14
数据长度	2	N
数据	id_len(1byte) + grp_id(nbyte)+ datapoint	id_len: 子设备 id 长度 grp_id: 群组 id datapoint: 控制命令数据点集合
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x14
数据长度	2	0000
数据	无	
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.21 WIFI 功能性测试

说明：扫描指定的 SSID: tuya_mdev_test,返回扫描结果和信号强度百分比

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x15
数据长度	2	0x0000
数据	Data	无
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x15
数据长度	2	0x0002
数据	2	数据长度为 2 字节： Data[0]: 0x00 失败, 0x01 成功;

		<p>当 Data[0]为 0x01 ,即成功时 , Data[1]表示信号强度 (0-100, 0 信号最差 , 100 信号最强)</p> <p>当 Data[0]为 0x00 ,即失败时 ,</p> <p>Data[1]为 0x00 表示未扫描到指定的 ssid ,</p> <p>Data[1]为 0x01 表示模块未烧录授权 key</p>
校验和	1	<p>从帧头开始按字节求和得出的结果对 256 求余</p>

3.22 获取 WIFI 状态

说明：MCU 可根据此条命令获取 WIFI 状态。具体 WIFI 状态见 3.3。网关在网络初始化完成前不会恢复 wifi 状态。

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x16



数据长度	2	0x0000
数据	Data	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x16
数据长度	2	0x0001
数据	1	指示 WIFI 工作状态： 0x00:状态 1 0x01:状态 2 0x02:状态 3 0x03:状态 4 0x04:状态 5
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.23 恢复出厂设置

说明：网关恢复出厂设置，删除全部子设备并且移除配网。

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x17
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.24 报告移除状态

说明：1) 网关收到移除或者恢复出厂命令时，会向 MCU 报告状态。

2) 如果仅收到移除指令，网关只会移除配网，并不会删除子设备信息。下次联网时，子设备信息仍会在 app 上显示。

3) 如果收到恢复出厂指令，网关会移除配网并且清除所有子设备信息。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00

命令字	1	0x18
数据长度	2	0x0001
数据	1	0x00 本地恢复出厂(网 关重置并且删除所有子 设备) 0x01 远程移除 0x02 本地移除 0x03 远程恢复出厂(网 关重置并且删除所有子 设备)
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.25 本地删除子设备

说明：

MCU 发送删除子设备指令。

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x19



数据长度	2	N
数据	{ "sub_id" : " xxx" }	
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x19
数据长度	2	0x0001
数据	1	0x00 接收请求 0x01 拒绝请求
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.26 本地允许/关闭添加子设备

MCU 发送：

字段	长度 (byte)	说明
----	-------------	----

帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1A
数据长度	2	0x0003
数据	3	Data[0]: 0 : 关闭子设备添加 1 : 允许子设备添加 data[1]: 允许添加时间 (高字节) data[2]: 允许添加时间 (低字节)
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1A
数据长度	2	0x0001
数据	1	0 : 成功

		1 : 失败
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.27 获取模块内存

说明：获取 WIFI 模块剩余内存

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x03
命令字	1	0x1B
数据长度	2	0x0000
数据	Data	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1B
数据长度	2	0x0004

数据	4	数据长度为 4 字节,大端格式： 如 0x00 0x00 0x28 0x00 代表剩余 10240 字节内存
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

3.28 查询子设备列表

说明：1) MCU 可查询网关下所有子设备的列表。

2) 当列表数据大于 255byte 时，会分包发送

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1C
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1C
数据长度	2	N
数据	id(1byte)+sub_num(1 byte) sub1_len(1byte)+sub1_id+..... subn_len(1byte)+subn_id	id: bit 7: TURE 所有子设备地址已经发送完成。 FALSE 后续仍有子设备地址，还有一包数据 bit0-6:包序号从 0 开始 sub_num: 这包数据中包含的子设备数量 n sub1_len: 子设备 id 长度 sub1_id: 子设备 id
校验和	1	从帧头开始按字节求和得出的结果对 256 求余

3.29 远程升级服务

说明：

- 1) 何时升级客户涂鸦 IOT 平台自己的产品页面配置相关升级选项触发,模块仅作为支持 MCU 升级的数据传输通道,也不对数据内容做任何解析。
- 2) 目前涂鸦平台的 MCU 升级支持下面四种升级方式的配置：
 1. app 提醒升级：这个升级方式用户每次进入设备控制面板都会收到升级提醒的弹窗,是否确认升级由用户自己在 APP 确认
 2. app 静默升级：这个升级方式 app 不会有任何提醒弹窗,固件上电后一分钟内会去自动去检测升级发现有高版本的升级包会自动开始拉取相关升级包,第一次上电后模块会间隔 24 小时去云端检测一次是否有升级包配置。
 3. app 强制升级: 这个升级方式 app 端会有升级提醒弹窗,如果用户不确认升级用户就没法正常使用这个产品的控制面板。
 4. app 检测升级:这个升级方式 app 端不会有任何升级提醒的弹窗,必须要用户在 app 端自己去点击相关固件版本检测,如果有高版本的固件配置才会显示升级提示信息。

3) MCU 升级相关流程图：

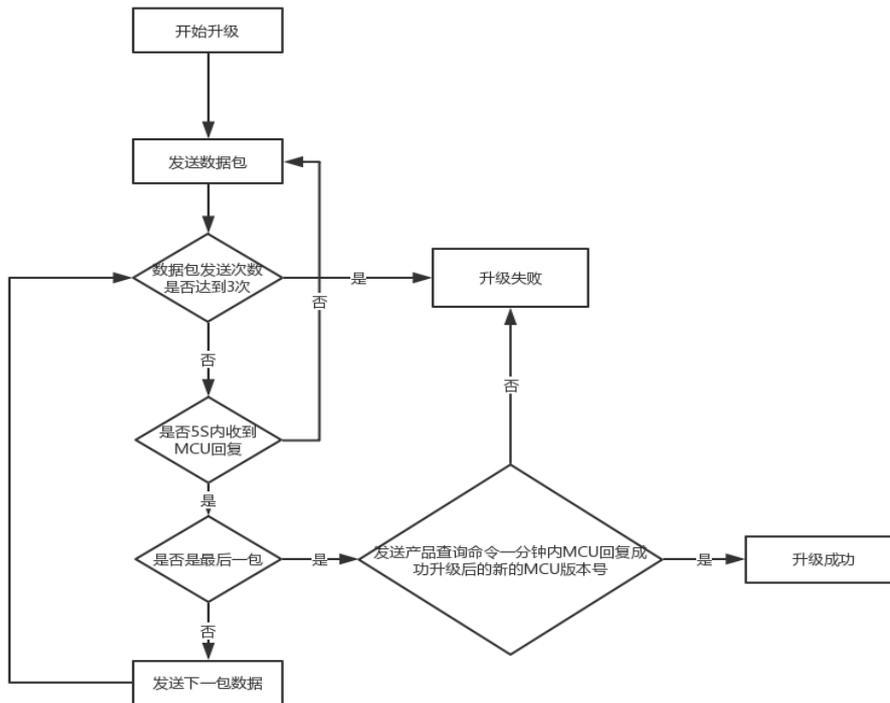
WIFI 模块发送完所有的升级包,重新发送 01 命令字 (3.2 查询产品信息)
MCU 需要在一分钟回复产品信息中的软件版本号带上升级后的 MCU 版本号,
版本号需要和在涂鸦后台配置升级的版本号保持一致。

3.29.1 MCU 启动升级

说明：

WIFI 模块发送完所有的升级包，重新发送 01 命令字 (3.1 查询产品信息)

MCU 需要在一分钟回复产品信息中的软件版本号带升级后的 MCU 版本号，版本号需要和在涂鸦后台配置升级的版本号保持一致。



模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1D
数据长度	2	0x0004
数据	file_size (4byte)	file_size: 升级文件大小， unsigned int，大端

校验和	1	从帧头开始按字节求和得出的 结果对 256 求余
-----	---	-----------------------------

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1D
数据长度	2	0x0001
数据	1	升级包分包传输大小： 0x00 :默认 256byte(兼容旧固件) 0x01 : 512byte 0x02 : 1024byte
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.29.2 MCU 升级包传输

说明：

- 1) 升级包传输数据格式：包偏移 (unsigned short) + 包数据
- 2) MCU 若收到该帧数据长度为 4 且包偏移 >= 固件大小，则包传输结束

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1E
数据长度	2	0x0004+m
数据	offset_addr(4byte)+ pack(mbyte)	offset_addr: 包偏移地址 pack : 数据包内容
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

例：

若要升级的文件大小 530Byte，(最后一包数据可不回复)

(1) 第一包数据，包偏移为 0x00000000，数据包长度为 256

0x55aa 00 1e 0104 00000000 xx...xx XX

(2) 第二包数据，包偏移为 0x00000100，数据包长度为 256

0x55aa 00 1e 0104 00000100 xx...xx XX

(3) 第三包数据，包偏移为 0x00000200，数据包长度为 18

0x55aa 00 1e 0016 00000200 xx...xx XX

(4) 最后一包，包偏移为 0x00000212，数据包长度为 0

0x55aa 00 1e 0004 00000212 xx...xx XX

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1E
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.29.3 子设备启动升级

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1F
数据长度	2	N
数据	id_len(1byte) + sub_id(nbyte)+ file_size(4byte)	id_len: 子设备 id 长度 sub_id:

		子设备 id file_size: 升级文件大小， unsigned int，大端
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x1F
数据长度	2	0x0001
数据	1	升级包分包传输大小： 0x00 :默认 256byte(兼容旧固件) 0x01 : 512byte 0x02 : 1024byte
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.29.4 子设备升级包传输

说明：

3) 升级包传输数据格式：包偏移 (unsigned short) + 包数据

4) MCU 若收到该帧数据长度为 4 且包偏移 >= 固件大小，则包传输结束

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x20
数据长度	2	N
数据	id_len(1byte) + sub_id(nbyte)+ offset_addr(4byte)+ pack(mbyte)	id_len: 子设备 id 长度 sub_id: 子设备 id offset_addr: 偏移地址 pack : 数据包
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

例：

若 sub_id 为“ 1234” ，要升级的文件大小 530Byte ，(最后一包数据可不回复)

(1) 第一包数据，包偏移为 0x00000000 ，数据包长度为 256

0x55aa 00 20 0109 04 31 32 33 34 00000000 xx...xx XX

(2) 第二包数据，包偏移为 0x00000100 ，数据包长度为 256

0x55aa 00 20 0109 04 31 32 33 34 00000100 xx...xx XX

(3) 第三包数据，包偏移为 0x00000200 ，数据包长度为 18

0x55aa 00 20 001B 04 31 32 33 34 00000200 xx...xx XX

(4) 最后一包，包偏移为 0x00000212 ，数据包长度为 0

0x55aa 00 20 000904 31 32 33 34 00000212 xx...xx XX

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x20
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

3.29.5 查询子设备版本号

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x21
数据长度	2	N
数据	{ "sub_id" : " xxx" }	sub_id:子设备 id
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x21
数据长度	2	N
数据	{ "sub_id" : " xxx" ,	sub_id: 子设备 ID

	<pre> "ver" : " xx.xx.xx" } </pre>	ver: 子设备固件版本号
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

4 扩展协议

4.1 sig mesh 扩展协议

4.1.1 获取网络信息

说明：拉取 sig mesh 入网必需的网络信息。

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x80
数据长度	2	0x0000
数据	Data	无
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x80
数据长度	2	N
数据	{ "netkey" : " xxxx" , "appkey" : " xxxx" }	Netkey:16 个字节 Appkey:16 个字节
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

4.1.2 申请节点

说明：获取 sigmesh 网络中未使用的 node 列表。

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00



命令字	1	0x81
数据长度	2	0x0001
数据	1	申请节点数量
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x81
数据长度	2	2+m(id 长度)*n(n 数量)
数据	node_num (n) (1byte)+ id_len(m)(1byte)+ nodeid_1(m byte)+...+ nodeid_n(m byte)	每个 nodeid 大端
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

4.1.3 新增子设备同步

说明 客户端添加直连设备，网关需要将这个新增的直连设备信息同步给 MCU。

模块发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x82
数据长度	2	0x0001
数据	{ "sub_id ":" xxx" , "pk_type" :xx, "devkey" : " xxxxx" , "mac" : " xxx" , }	子设备信息： sub_id:子设备唯一标识 码，可使用 mac 地址。 pk_type：产品类型 devkey：设备密钥
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

MCU 返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x82



数据长度	2	0x0001
数据		0 : 成功 1 : 失败
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余

4.1.4 子设备添加

说明：

- (1) MCU 发送 sigmesh 子设备入网请求给 WIFI。
- (2) pk_type 备用字段，可不加。默认为 1

MCU 发送：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x83
数据长度	2	N
数据	前绑定： { "pk_type" :xx, "sub_id" : " xxxx" },	用户产品分类 pk_type: 用户自定义产品类型，可不传。 子设备标识 sub_id: 同一个网络需要保

	<pre> "mac" : " xxx" , "pid" : " xxxx" , "ver" : " x.x.x" , "devkey" : " xxx " } 后绑定: { "pk_type" :xx, "sub_id" : " xxxx " , "mac" : " xxx" , "ver" : " x.x.x" , "devkey" : " xxx " "firmkey" : " xxxx " , } </pre>	<p>证唯一性.不可以是" 0000" ." 0000"表示网关本身。</p> <p>产品标识 pid : 产品 ID (ProductID) , 涂鸦云 创建产品时生产 Mac: 设备唯一标识码 版本号 ver : 如 : 1.0.0 dev_key: 设备密钥 固件 firmkey: 在后台上传固件时生产 如果该设备是前绑定则传 pid 信息.若是后绑定 , 则传 firmkey, oem 信息。</p>
<p>校验和</p>	<p>1</p>	<p>从帧头开始按字节求和得出的</p>

		结果对 256 求余
--	--	------------

模块返回：

字段	长度 (byte)	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x83
数据长度	2	0x0001
数据	1	0x00 接收请求 0x01 拒绝请求 (子设备实际是否入网成功，详见 3.19)
校验和	1	从帧头开始按字节求和得出的 结果对 256 求余