

涂鸦云低功耗通用串口接入协议

协议生成时间：2019年03月19日 16:55

产品信息

产品名称：Lock
产品ID(PID)：b7do7sirjb0bfka0
产品功能：

dpID	功能名称	数据传输类型	数据类型	功能属性	备注
101	解锁方式	只上报	enum	枚举范围：1, 2, 3, 4	具体的编号与开门方式的对应，可以在多语言中翻译 1=人脸、2=密码、3=卡、4=机械钥匙
102	解锁ID	只上报	value	数值范围：0-999，间距：1，单位：	门锁上报远程开门请求，需要硬件主动触发 门锁本地上报远程开门请求时，携带x秒倒计时，只有在x秒内才能做出响应，点击允许，门锁将会解锁。可以x小于等于90
103	报警	只上报	enum	枚举范围：1, 2, 3, 4, 5, 6	1=低电、2=电池即将耗尽、3=防撬报警、4=密码盘锁死、5=假锁、6=高温
104	远程开门倒计时	只上报	value	数值范围：0-90，间距：1，单位：s	
105	电量状态	只上报	enum	枚举范围：high, medium, low, poweroff	高、中、低、超低（即将关机）

类型说明：

布尔型(bool)：非真即假的二值型变量，如开关功能；

数值型(value)：可线性调节数值型的功能，如温度调节20-40℃；

枚举值(enum)：自定义的有限集合值，如档位的高、中、低；

故障型(fault)：专门用于上报和统计故障的功能点，支持多故障，数据只上报。故障值，是按照对应的字节位来标识的，依次为bit0-bitx。从左到右依次对应，如E1(bit0)、E2(bit1)、F1(bit3)。多个字节位代表多个故障可同时发生，最多支持30位；

字符串型(string)：以字符串形式传输的功能点；

透传型(raw)：以二进制形式透传的功能点。

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

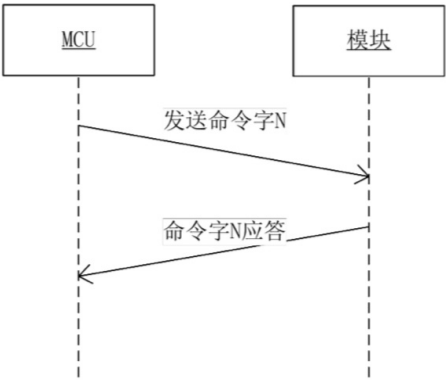
MCU：用户控制板控制芯片，与涂鸦模块通过串口对接，这里协议设计上，所有包的交互为全双工通信设计。

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：

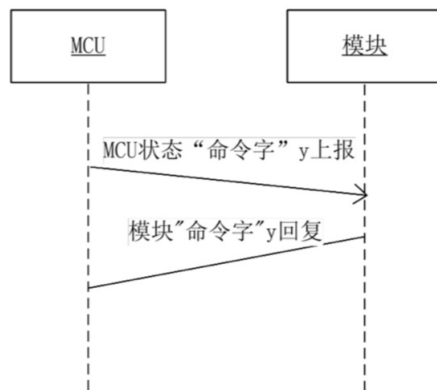
- ◆ 所有大于1个字节的数据均采用大端模式传输。
- ◆ 协议中所有举例说明数据都是十六进制数据。
- ◆ WIFI模块主动发起的通信发送包超时时间为1S，重发机制会重发三包。
- ◆ 一般情况下，采用同命令字一发一收同步机制，即一方发出命令，另一方应答，若发送方超时未收到正确的响应包，则超时传输，如下图所示：



说明：具体通信方式以“协议详述”章节中为准

- ◆ MCU状态上报则采用同步模式，MCU状态上报“命令字”为y，如下所示：

MCU统计数据上报：



协议详述

1. 查询产品信息

说明:

- 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
- 产品信息由product id、MCU软件版本构成
- MCU软件版本号格式定义: 采用点分十进制形式, "x.x.x" ($0 \leq x \leq 99$), 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

模块发送:

55 aa 00 01 00 00 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	{ "p": "vHXEcqntLpkA10sy", "v": "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

例: { "p": "vHXEcqntLpkA10sy", "v": "1.0.0" }

p 表示产品ID为vHXEcqntLpkA10sy, 这里参数为用户在涂鸦官网创建的产品PID

v 表示MCU版本为1.0.0

MCU返回例子信息：

55 aa 00 01 00 24 7b 22 70 22 3a 22 76 48 58 45 63 71 6e 74 4c 70 6b 41 6c 4f 73
79 22 2c 22 76 22 3a 22 31 2e 30 2e 30 22 7d bf

2. 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	x04

说明：

- i. 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。
- ii. 当模块的WIFI状态发生变化，则会主动下发WIFI状态至MCU。
- iii. 这里的状态包用于MCU设备得知模块目前的状态，这里两种配置状态需要MCU做相关的配网显示。
- iv. 这里原则上我们要求我们的用户需要做两种配网方式，第一种配网方式存在少部分路由兼容性问题，第二种配网方式是确保设备可以联网成功的保障。

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	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

模块发送：

55 aa 00 02 00 01 04 06（已连上路由器且连接到云端）

MCU返回：

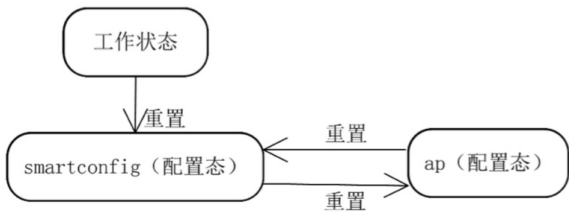
字段	长度（byte）	说明
帧头	2	0x55aa
版本	1	0x00
命令字	1	0x02
数据长度	2	0x0000
数据	0	无
校验和	1	从帧头开始按字节求和得出的结果对256求余

MCU返回：

55 aa 00 02 00 00 01

3. 重置WIFI

重置WIFI状态转化如下图所示：



MCU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	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发送：

55 aa 00 03 00 00 02

模块返回：

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

模块返回：

55 aa 00 03 00 00 02

4. 重置WIFI-选择配置模式

说明：

- i. 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
- ii. MCU接入用户可选择性的实现该协议

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	0x0001
数据	1	0x00: 进入smartconifg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

MCU发送控制模块进入AP配置模式：

55 aa 00 04 00 01 01 05

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

模块返回：

55 aa 00 04 00 00 03

5. 实时状态上报

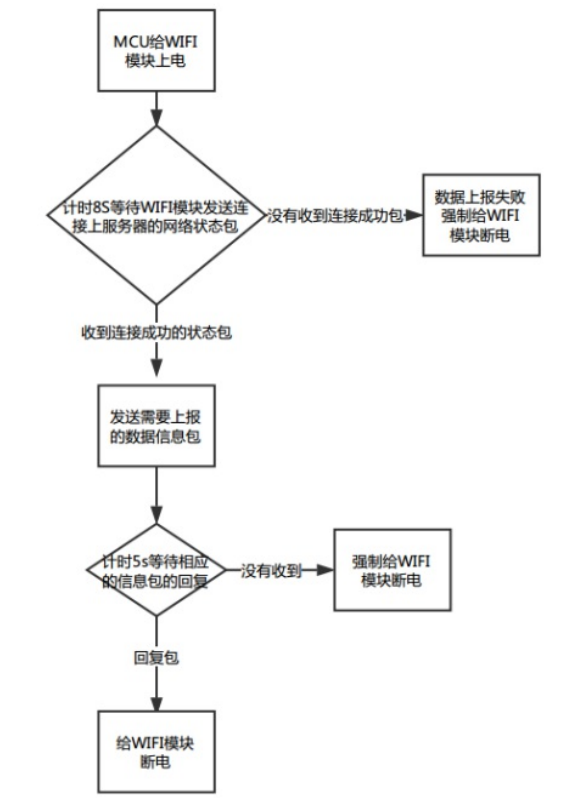
使用场景：

对于类似报警器要求实时推送的设备，这里我们就可以通过本条命令上报相关状态数据。

说明：当MCU需要上报实时状态的数据时，可通过改条协议上报数据

- i. 状态数据会直接上报到云端，故设备联网状态必须为已连接到云端，否则数据上报失败
- ii. 该条指令为同步指令，MCU数据上报后需等待模块返回上报结果，等待超时时间5秒
- iii. 需要等待模块发送连接上服务器的网络状态包，再发送需要发送的统计数据包，这里的数据上传没有储存功能，这强制等待8s还没收到连接上服务器的网络状态包，也强制给设备断电。
- iv. 这里可以支持多个数据单元上报，和单个数据单元上报，用户根据自己的需要来选择发送组包的方式。具体数据包可以下面的例子。

大致操作的流程图：



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	取决于“状态数据单元”类型以及个数
数据	N	一个或多个组合“19 状态数据单元”组
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	0x05
数据长度	2	0x0001
数据	1	0x00 成功 0x01 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

“单个状态数据单元”上报例子：

DP点109 bool型变量，数值为1

55 aa 00 05 00 05 6d 01 00 01 01 79

“多个状态数据单元”上报例子：

DP点109 bool型变量，数值为1

DP点102 string型变量，“201804121507”（具体传输对应ASCII值）

55 aa 00 05 00 15 6d 01 00 01 01 66 03 00 0c 32 30 31 38 30 34 31 32 31 35 30
37 5d

6. 记录型状态上报

使用场景：

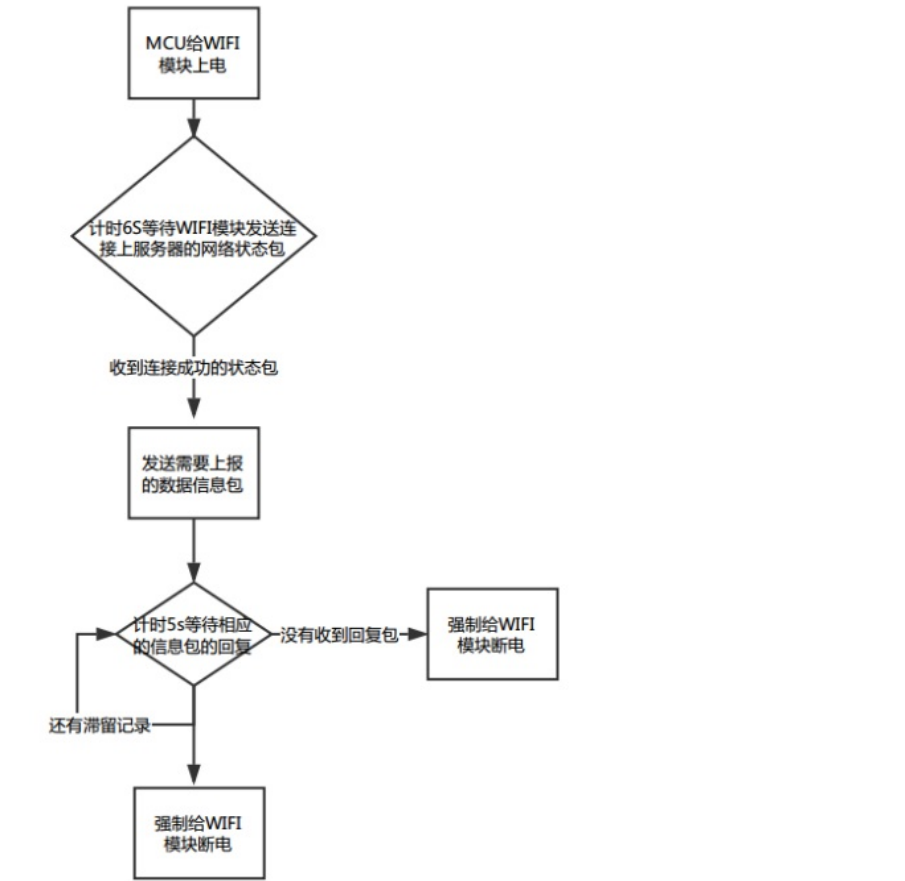
对于类似门锁等记录型的设备，包含多个DP点数据需要服务端作为整条记录处理，在短暂断网的情况下本条命令会保存下无法上报成功的数据，本条命令可以满足记录型设备的上报需求。这里的上报需要带上时间为本地时间。

说明：

当MCU需要记录型数据是由多个DP点组合的整条数据，需要整体上报，便可以通过这条命令往WIFI模块发送，并且当由数据上报时此时设备断网了的时候这条数据储存下来当下次有数据上报的时候模块再上传这条数据之后，并把之前的储存数据上报成功。

- i. 这里当有滞留记录每次上报成功一条，模块就会主动发送一条09命令字数据为01的回复包。这里MCU可以根据这个回复做超时断电处理！这里当发送一条记录这里强制等待5s当没有收到模块任何回复，认为模块异常也要断电模块。
- ii. 网络正常下模块上电后正常需要4s左右连接上服务器，这里每当有记录产生给设备上电后，可以等WIFI模块发送协议文档中3.2所描述的的设备联网状态包，当强制等待6s以上还没收到模块连接上服务器的包，也将该条指令MCU数据上报后需等待模块返回上报结果。
- iii. 单次记录上报的数据区域(多个状态数据单元)最大长度为80，这里根据实际的DP点的数据最后组合实际储存的长度会有所变化，这里当没有网络的状态下，超过限制长度WIFI模块会返回记录发送失败。
- iv. 可以储存历史记录的条数最长限制到20条，当超过20条从最早储存的记录开始覆盖，如此循环覆盖。
- v. 这里当WIFI模块接受到一条数据成功推送，或者当没有网络状态下，记录成功存储进flash中也会当做推送成功（00）。当有网络推送成功一条，还有滞留记录的话会返回01. 其他的情况均返回推送失败02。
- vi. 这里的时间数据的应用是为了保证记录数据在设备没有网络的情况下和实际发生的事件一致。这里如果以记录到达服务端的时间为准，当设备暂时处于没有网络的状态下产生的记录数据，当下次使用网络恢复正常时，这个时候上传的记录时间便和实际发生的事件不符合。所以这里当设备第一次和WIFI模块配网的时候，网络环境可以保证的情况下，可以通过协议获取到想要的时间数据。每次上传记录的时候带上当前设备的时间数据，服务端的记录时间便会以设备给的时间为准。
- vii. 这里由于设备在全世界各地涉及到时区和夏令时等事件因素，这里记录的时间的处理也给与两个时间的协议区分。当设备没有显示屏的需求，这里设备通过协议获取格林时间在设备中运行，当有相关上报数据产生的时候便按照协议给与当前设备运行的格林时间的数据。这里有些设备，比如门锁需要显示当地的时间，这里门锁可以通过协议获取本地时间，这个时间模块有加上时区包括夏令时的因素，确保时间和当地的时间可以对上。当有相关上报数据产生的时候便按照协议给与当前设备运行的本地时间的数据。

上报大致操作流程圖：



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	取决于“19. 状态数据单元”类型以及个数
数据	7	数据长度为7字节： Data[0]为这条数据传输是否带本地时间标志位： 0表示这条数据不带门锁给的时间后面的时间模块认为数据无效不处理 1表示后面的时间数据有效，时间数据为设备所在的当地时间。 2表示后面的时间数据有效，时间数据为格林时间。 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59结束
	N	一个或多个组合“12. 状态数据单元”组
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	0x08
数据长度	2	0x0001
数据	1	0x00 上报成功 0x01 当前记录上报成功，并且还有滞留记录需要上报 0x02 上报失败
校验和	1	从帧头开始按字节求和得出的结果对256求余

具体上报数据包举例：

“单个状态数据单元” 上报例子：

DP点109 bool型变量，数值为1

```
// 服务端当前时间为准
55 aa 00 08 00 0c 00 12 04 13 0d 04 14 6d 01 00 01 01 d1

// 设备本地时间（2018/04/19/ 13点3分29秒）为准
55 aa 00 08 00 0c 01 12 04 13 0d 03 1d 6d 01 00 01 01 da

// 设备当前格林时间（2018/04/19/ 5点3分29秒）为准
55 aa 00 08 00 0c 02 12 04 13 05 03 1d 6d 01 00 01 01 d3
```

“多个状态数据单元” 上报例子：

DP点109 bool型变量，数值为1
DP点102 string型变量，“201804121507”（具体传输对应ASCII值）

```
// 服务端时间为准
55 aa 00 08 00 1c 00 12 04 13 0d 06 04 6d 01 00 01 01 66 03 00 0c 32 30 31 38 30
34 31 32 31 35 30 37 a7

// 设备本地时间（2018/04/19/ 13点3分29秒）为准
55 aa 00 08 00 1c 01 12 04 13 0d 08 2e 6d 01 00 01 01 66 03 00 0c 32 30 31 38 30
34 31 32 31 35 30 37 d4

// 设备格林时间（2018/04/19/ 5点3分29秒）为准
55 aa 00 08 00 1c 02 12 04 13 05 08 2e 6d 01 00 01 01 66 03 00 0c 32 30 31 38 30
34 31 32 31 35 30 37 cd
```

7. 模块命令下发

说明：

“命令下发”为异步处理协议，MCU收到相关控制包，确认接收到回复下发包后，完成相关的控制动作MCU状态反馈通过MCU的状态上报来实现。

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	取决于“命令数据单元”类型

		以及个数
数据	N	“19. 命令数据单元” 组
校验和	1	从帧头开始按字节求和得出的结果对256求余

模块下发控制命令：（系统开关对应3号DP, 使用bool型变量，开机数值为1）

55 aa 00 09 0005 03 01 00 01 01 13

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

MCU确认回复：

55 aa 03 09 00 00 0b

8. 获取本地时间

说明：

- i. 需要等待设备连接上网络（也就是收到设备3.2命令 状态5）后开始发送获取本地时间的数据包
- ii. 在设备连接上服务器后，在网络情况非常差的情况下，有可能会获取时间数据失败。这里对于依赖时间的设备（例如门锁）当本地时间没有校准过的请款下当获取失败，这里需要做一个间隔3S的获取时间值，确保时间数据获取成功。

MCU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获取本地时间：

55 aa 00 06 00 00 05

模块返回：

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	0x0008
数据	N	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59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

模块返回本地时间数据：
55 aa 00 06 00 08 01 12 09 11 10 09 05 01 59
本地时间:18年9月17日16时9分5秒星期一

- (1) 如设备在国内激活使用，则当地时间为北京时间(东8区)
- (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。

9. 获取格林时间

- i. 格林时间不带有时区和夏令时相关因素，这里带动态密码功能（门锁等）的但是不需要显示设备所在当地时间的设备，本地只需要跑格林时间，记录上传走带记录型上报的通道给的时间也需要是格林时间。
- ii. 这里带动态密码功能的设备，并且门锁也要显示设备所在当地的当前时间，本地在跑跑格林时间的同时，需要在定时获取本地时间，本地储存格林时间和当地时间的时间差，每次显示本地时间的时候在格林时间的基础上加上时间差。记录上传走带记录型上报的通道给的时间也需要是格林时间。

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

MCU获取格林时间：
55 aa 00 10 00 00 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	0x0008
		数据长度为8字节：

数据	8	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59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

模块返回设备格林时间：
55 aa 00 10 00 08 01 12 09 11 08 15 03 01 65
格林时间:2018年9月17日8时21分3秒 星期一

10. WIFI功能性测试

- 说明：
- i. 扫描指定的SSID: tuyamdev_test, 返回扫描结果和信号强度百分比。
 - ii. 产测指令需要在wifi上电工作完成初始化流程以后发送（回复3.1查询产品信息的数据包），否则导致产测失败或无结果。
 - iii. 本条命令多用于设备量产时的产品整机测试使用。
 - iv. 产测路由器密码设置不做要求（随意设置），这里路由需要为2.4G信号的路由器。

MCU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	N	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触发模块功能性测试：
55 aa 00 07 00 00 06

模块返回：

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	0x0002
数据	2	数据长度为2字节： Data[0]: 0x00失败，0x01成功； 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） 当Data[0]为0x00，即失败时： Data[1]表示失败原因

		0x00表示未扫描到指定的ssid Data[1]为0x01表示模块未烧录 授权key
校验和	1	从帧头开始按字节求和得出的 结果对256求余

模块返回测试结果：
55 aa 00 07 00 02 01 50 59
WIFI功能性测试成功信号强度为80

11. 请求WIFI模块固件升级

说明：

这里WIFI模块的电源通断是由MCU去控制的，这里当我们MCU需要去升级WIFI模块的固件便可以通过下面的命令去拉去最新固件。MCU主板根据WIFI模块的回复包决定是否需要给WIFI模块断电。这里MCU当发送0a命令等待5S没有收到相关回复便把WIFI模块断电。当模块回复正在更新固件这里MCU也需要起一个定时当处于正在更新固件超过60S没有收到固件升级成功，也强制认为固件升级失败给WIFI模块断电。

MCU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	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请求WIFI固件升级：
55 aa 00 0a 00 00 09

模块返回：

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	0x0001
数据	1	0x00(开始检测固件更新) 不可 断电 0x01（已经是最新固件） 断电 0x02(正在更新固件) 不可断电 0x03(固件更新成功) 断电 0x04(固件更新失败) 断电
校验和	1	从帧头开始按字节求和得出的 结果对256求余

模块升级状态返回：
55 aa 00 0a 00 01 00 0a（收到升级请求包立马返回）
55 aa 00 0a 00 01 01 0b（拉取服务端信息返回没有固件需要更新）

12. 请求MCU固件升级

说明：

- i. 这里当MCU收到模块返回固件更新完成之后，说明服务端MCU的升级文件已经全部拉取完成并且完成了串口数据传输。这里注意MCU在收到完整升级文件后，确认拉取文件没有问题，这个时候模块会重新发送查询信息，MCU需要返回相关产品信息和在升级服务器后台填写一致的新的软件版本号。
- ii. 目前支持最大为480K的MCU文件的OTA。
- iii. 这里用户再涂鸦官网配置MCU升级时，配置MCU升级方式为app静默升级

MCU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	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请求MCU固件升级：

55 aa 00 0c 00 00 0b

模块返回：

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	0x0001
数据	1	0x00(开始检测固件更新) 不可断电 0x01（已是最新固件） 断电 0x02(正在更新固件) 不可断电 0x03(固件更新完成) 断电 0x04(固件更新失败) 断电
校验和	1	从帧头开始按字节求和得出的结果对256求余

模块升级状态返回：

55 aa 00 0c 00 01 00 0c（收到升级请求包立马返回）

55 aa 00 0c 00 01 01 0d（拉取服务端信息返回没有固件需要更新）

13. 升级启动

说明：

当MCU触发了模块去升级MCU固件，并且服务端也配置了更高版本的MCU固件，升级方式也符合这里模块便会返回给MCU相关MCU需要升级的固件包的文件大小。

模块发送：

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	0x0004
数据	4	固件包字节数，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

模块发送文件包大小：
55 aa 00 0d 00 04 00 00 68 00 78（固件包长度26624，即26KB）

MCU返回：

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	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确认包回复：
55 aa 00 0d 00 00 0c

14. 升级包传输

- 说明：
- i. 升级包传输数据格式：包偏移（unsigned short） + 包数据
 - ii. 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	0x0e
数据长度	2	0x0004+数据包长度
数据	N	前四字节，固定为包偏移，后面为数据包内容
校验和	1	从帧头开始按字节求和得出的结果对256求余

模块发送文件数据：
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
55aa 00 0e 0104 00000000 xx····xx XX

(2) 第二包数据，包偏移为0x00000100，数据包长度为256
55aa 00 0e 0104 00000100 xx····xx XX

(3) 第三包数据，包偏移为0x00000200，数据包长度为18
55aa 00 0e 0016 00000200 xx····xx XX

(4) 最后一包，包偏移为0x00000212，数据包长度为0
55aa 00 0e 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	0x0e
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每包数据包确认：
55aa 00 0e 0000 0d

15. 查询当前连接路由信号强度

说明：
当需要查询设备连接路由器当前的路由的信号强度，这里前提是收到设备的网络状态包得知设备已经成功连接上路由器，否则发送此命令将返回失败的结果。

MCU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	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获取查询设备连接的路由器强度：
55 aa 00 0b 00 00 0a

模块返回：

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	0x0002
数据	2	数据长度为2字节： Data[0]: 0x00失败，0x01成功； 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） 当Data[0]为0x00，即失败时， Data[1]为0x00，表示设备未连接上路由

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

模块返回当前强度值(80):
55 aa 00 0b 00 02 01 50 5D

16. 请求云端临时密码（只支持单组）

说明:

- i. 这里需要在APP端创建当前时间的密码，设备才能拉取到当前生效的密码。
- ii. 这里这个接口只支持一组临时密码的拉取，此为早期门锁通用固件1.0.0的接口需要和对应的早期门锁面板对应结合使用，后面所有APP面板临时密码均用4.8的支持创建多组临时密码的接口获取相关密码数据。

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	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

门锁获取当前生效临时密码:
55 aa 00 11 00 00 10

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	0x0007+N (临时密码)
数据	7 + N	数据长度为7+N字节: Data[0]为是否获取密码成功标志，为0表示失败，后面不会带上相关密码和有效时间数据，数据内容长度为1，为1表示成功 密码有效截至日期（格林时间）： 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59结束 临时密码 (N) 密码数据传输字符的ASLL码
校验和	1	从帧头开始按字节求和得出的结果对256求余

模块返回当前密码:

55 aa 00 11 00 0d 01 10 04 13 05 06 07 31 32 33 34 35 36 8c

格林时间2016年4月19日5时6分7秒

密码 “123456”, 这里密码部分传输的字符的ASLL码

17. 动态密码校验

说明:

- i. 协议中的时间数据用于运算当前的动态密码，这里的动态密码运算时间都为当前的格林时间，设备需要给模块格林时间数据。
- ii. 这里APP端可以设置一个设备当前的是否要加入管理员密码用于参与动态密码的混算结果，这里设备每次再请求校验动态密码的时候，都需要给模块当前门锁所有的管理员密码。
- iii. 发送包的数据长度至少为15字节，时间数据+密码+传输的管理员密码组数（不包可选项含管理员密码）

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	0x0006+0x08+0x01+N
数据	6	数据长度为6字节: 设备当前格林时间: Data[0]为年份, 0x00表示2000年 Data[1]为月份, 从1开始到12结束 Data[2]为日期, 从1开始到31结束 Data[3]为时钟, 从0开始到23结束 Data[4]为分钟, 从0开始到59结束 Data[5]为秒钟, 从0开始到59结束
	8	用户输入的密码数据: Data[6]~Data[13] 数据内容范围为 ‘0’ ~ ‘9’ 数据传输使用ASCLL码
	1	管理员密码组数 (0~10) 有多少组管理员密码后面带上相应组数数据
	1	管理员密码长度 (长度最长不超过8个字节)
	N	第一组管理员密码部分 (密码部分由数字组成) 数据内容范围为 ‘0’ ~ ‘9’ 数据传输使用ASCLL码
	N	第二组管理员密码部分 (密码部分由数字组成) 数据内容范围为 ‘0’ ~ ‘9’ 数据传输使用ASCLL码

校验和	1	从帧头开始按字节求和得出的结果对256求余

MCU发送带两组长度为6的管理员密码的数据包:

55 aa 00 12 00 1c 12 09 11 06 22 29 31 35 39 35 30 31 35 38 02 06 35 32 31 33 31 34 35 32 31 33 31 33 b3

第一组密码数据: 521314

第二组密码数据: 521313

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	1	0x0001
数据	1	0x00（密码核对通过） 0x01（密码核对失败） //通信操作错误码 0x02（设备未激活） 0x03（数据长度错误）
校验和	1	从帧头开始按字节求和得出的结果对256求余

模块返回密码校验失败：

55 aa 00 12 00 01 01 13

18. 请求云端临时密码（支持多组）

说明：

- i. 这个临时密码这里我们APP设计上目前为了和其他密码区分开使用7位有效数据，门锁主板当检测到用户使用的是多组临时密码才去调用此接口去云端拉取相关数据，更新本地的密码数据和状态，这样可以最大程度节约功耗。
- ii. 注意这里模块给门锁给的相关时间数据都是基于格林时间，请门锁本地通过获取格林时间的接口同步自己的时钟数据。
- iii. 密码每次服务端都是全量下发，门锁需要每次根据服务端返回的所有密码和状态进行更新。

MCU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	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

门锁获取当前生效临时密码：

55 aa 00 13 00 00 12

模块返回：

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	0x0001+N（临时密码）
		数据长度为1+N字节： Data[0]为是否获取密码成功标志，为0表示失败，后面不会带上相关密码和有效时间数据，数据内容长度为1，为1表示成

数据	1 + N	功 Data[1] 密码组数 (0~10) 当密码组数为0的时候表示面板没有创建密码。串口传输不会带上相关密码和有效时间数据 Data[2] 密码长度 (后面每组的密码长度保持一致) //第一组密码相关数据 Data[3]密码编号 (实际编号需要在密码编号的数据上加900) (1~50) Data[4]密码有效次数 0: 表示不限次数密码-有效期内可以不限次数开门 1: 表示一次性密码-有效期内只能开一次门 Data[5]密码当前状态 0: 表示密码有效 1: 表示面板已经进行了删除操作此密码无效 密码生效日期 (格林时间): Data[6]为年份, 0x00表示2000年 Data[7]为月份, 从1开始到12结束 Data[8]为日期, 从1开始到31结束 Data[9]为时钟, 从0开始到23结束 Data[10]为分钟, 从0开始到59结束 Data[11]为秒钟, 从0开始到59结束 密码截至日期 (格林时间): Data[12]为年份, 0x00表示2000年 Data[13]为月份, 从1开始到12结束 Data[14]为日期, 从1开始到31结束 Data[15]为时钟, 从0开始到23结束 Data[16]为分钟, 从0开始到59结束 Data[17]为秒钟, 从0开始到59结束 临时密码 (具体字节数N=data[2]) (密码数据传输字符的ASLL码) //第二组密码数据
校验和	1	从帧头开始按字节求和得出的结果对256求余

WIFI模块返回当前10组密码数据密码:

55 AA 00 13 00 DF 01 0A 07
0A 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
09 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
08 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
07 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
06 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
05 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
04 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
03 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
02 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
01 00 00 12 06 1C 08 15 07 14 05 16 13 01 07 31 32 33 34 35 36 37
C6

13. 状态数据单元

说明:

i. datapoint命令/状态数据单元如下所示:

数据 字段	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 (byte)	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
		string	0x05	1/2/4	长度大于1字节时, 大端表示
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			

- ii. 0datapoint命令/状态数据单元除”raw”类型外, 其他类型均属于“obj”型datapoint
- iii. “状态数据”可含多个datapoint “命令数据单元”

实际开发中注意:

- 整套协议针对使用wifi模块且没发使用外电的设备。这里我们开发MCU程序的时候, 对于断电管理也显示尤为重要, 在我们可以完成功能的前提下尽量减少WIFI模块的上电时间, 是节约设备功耗的关键。这里对于数据上传部分我们有给出大体的控制流程图, 开发者可以根据自己设备的特性来选择协议中自己需要用到的功能。实际的开发中开发者可以根据自己的需要实时调整相关控制逻辑。
- 实际产品很多日常设备中设备不开启WIFI功能也可以正常使用, 这里不排除用户在拿到设备的同时由于种种原因不使用WIFI这个功能, 这里为了防止这种情况下给WIFI模块上电造成不必要的电能损耗, 我们在设备端可以设计一个物理按键或者相关选项, 只有当用户主动打开这个按键或则选项我们MCU才在每次数据变动的时候才去给WIFI模块上传传输相关数据。

通讯协议-功能协议

通讯协议 (产品功能部分) 指令收发表

ID	功能名称		帧头版本	命令字	数据长度	dpID	数据类型	功能长度	功能指令	校验
101	解锁方式	MCU上报	0x55aa0x00	0x07	0x00 0x05	0x65	0x04	0x00 0x01	1:0x00 2:0x01 3:0x02 4:0x03	校验和
102	解锁ID	MCU上报	0x55aa0x00	0x07	0x00 0x08	0x66	0x02	0x00 0x04	0x0-0x3e7	校验和
103	报警	MCU上报	0x55aa0x00	0x07	0x00 0x05	0x67	0x04	0x00 0x01	1:0x00 2:0x01 3:0x02 4:0x03 5:0x04 6:0x05	校验和
104	远程开门倒计时	MCU上报	0x55aa0x00	0x07	0x00 0x08	0x68	0x02	0x00 0x04	0x0-0x5a	校验和
105	电量状态	MCU上报	0x55aa0x00	0x07	0x00 0x05	0x69	0x04	0x00 0x01	high:0x00 medium:0x01 low:0x02 poweroff:0x03	校验和