

# 涂鸦通用串口接入协议

版本记录:

| 版本    | 编写/修订说明 | 修订人 | 修订日期     | 备注  |
|-------|---------|-----|----------|---|
| 1.0.0 | 创建文档    | 聂哲元 | 20151010 |   |
| 1.0.1 | Modify  | 聂哲元 | 20151013 | 将查询 product key 改为查询模块信息，新增设备版本信息返回   |
| 1.0.2 | Modify  | 聂哲元 | 20151017 | <ol style="list-style-type: none"><li>心跳检测协议中新增 MCU 重启检测功能</li><li>心跳检测间隔调整为 10s</li><li>模块 WIFI 状态上报更新为模块主动上报状态至 MCU</li></ol> |
| 1.0.3 | Modify  | 刘康  | 20151114 | <ol style="list-style-type: none"><li>增加 MCU 获取时间功能</li><li>增加 MCU 获取时区功能</li><li>增加进入产测功能</li></ol>                            |

# 1 串口通信约定

波特率：9600

数据位：8

奇偶校验：无

停止位：1

数据流控：无

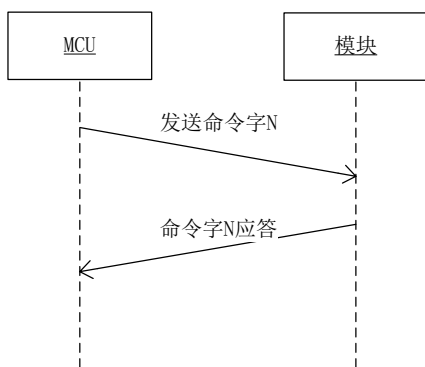
MCU：用户控制板控制芯片，与涂鸦模块通过串口对接

## 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   | xxxx      |                         |
| 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：

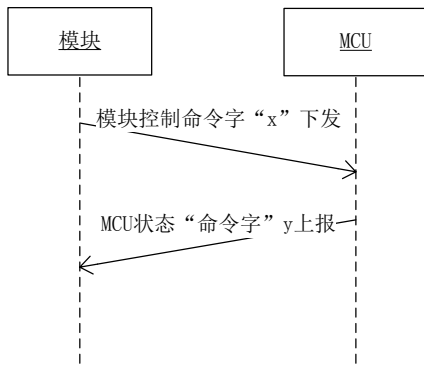
- 所有大于 1 个字节的数据均采用大端模式传输。
- 一般情况下，采用同命令字一发一收同步机制，即一方发出命令，另一方应答，若发送方超时未收到正确的响应包，则超时传输，如下图所示：



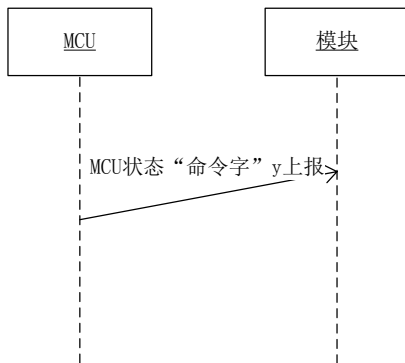
说明：具体通信方式以“协议详述”章节中为准

- 模块控制命令下发及 MCU 状态上报则采用异步模式，假设模块控制命令下发“命令字”为 x，MCU 状态上报“命令字”为 y，如下所示：

1) 模块控制命令下发：



2) MCU 状态上报:



## 3 协议详述

### 3.1 心跳检测

说明:

- 1) 模组上电后，以 10s 的间隔定期发送心跳，若在超时时间（3s）内，未收到 MCU 的回应，则认为 MCU 离线。
- 2) MCU 侧也可依据此心跳定期检测模块是否正常工作，若模块无心跳下发，则 MCU 可通过模块提供的硬件复位引脚复位模块。

模块发送:

| 字段 | 长度 (byte) | 说明     |
|----|-----------|--------|
| 帧头 | 2         | 0x55aa |
| 版本 | 1         | 0x00   |

|      |      |                         |
|------|------|-------------------------|
| 命令字  | 1    | 0x00                    |
| 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 |

MCU 返回:

| 字段   | 长度 (byte) | 说明  |
|------|-----------|---|
| 帧头   | 2         | 0x55aa  |
| 版本   | 1         | 0x00  |
| 命令字  | 1         | 0x00  |
| 数据长度 | 2         | 0x0001  |
| 数据   | xxxx      | 0x00: MCU 重启后第一次心跳返回值, 仅发送一次, 用于模块判断工作过程中 MCU 是否重启<br>0x01: 除 MCU 重启后第一次返回 0 外, 其余均返回此值 |
| 校验和  | 1         | 从帧头开始按字节求和得出的结果对 256 求余   |

## 3.2 查询产品信息

说明:

- 1) product key: 固定为 16 字节, 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 key、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), 同时支持 "x.x" 及 "x" 形式, 模块会自动扩展成 "0.x.x" 或 "0.0.x" 格式处理

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   | 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 |

MCU 返回:

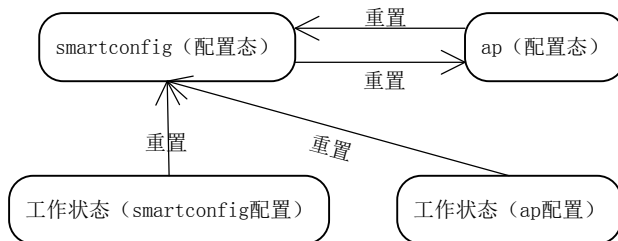
| 字段 | 长度 (byte) | 说明 |
|----|-----------|----|
|----|-----------|----|

|      |      |                         |
|------|------|-------------------------|
| 帧头   | 2    | 0x55aa                  |
| 版本   | 1    | 0x00                    |
| 命令字  | 1    | 0x01                    |
| 数据长度 | 2    | 0x0010 < len <= 0x18    |
| 数据   | xxxx | prodect key+mcu version |
| 校验和  | 1    | 从帧头开始按字节求和得出的结果对 256 求余 |

### 3.3 查询 MCU 设定的模块工作模式

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，然后 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 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。
  - c. WIFI 工作状态：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。对应的“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 长亮状态
  - d. 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。

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   | 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 |

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    | 0x0000/0x0002<br>1 数据长度为 0: 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。<br>2 数据长度为 2: 指示模块工作为“模块自处理”模式。 |
| 数据   | xxxx | 数据长度为 2: 首字节为 WIFI 状态指示 GPIO 序号; 次字节为 WIFI 重置按键 GPIO 序号  |
| 校验和  | 1    | 从帧头开始按字节求和得出的结果对 256 求余  |

### 3.4 报告 WIFI 工作状态

说明:

- 1) WIFI 工作状态: 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。对应的“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 长亮状态
- 2) 当模块检测到 MCU 重启或 MCU 断线再上线的过程, 则主动上报 WIFI 状态至 MCU
- 3) 当模块检测到 WIFI 状态发生变化, 则主动上报 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   | xxxx      | 指示 WIFI 工作状态:<br>0x00:状态 1<br>0x01:状态 2<br>0x02:状态 3<br>0x03:状态 4 |
| 校验和  | 1         | 从帧头开始按字节求和得出的结果对 256 求余   |

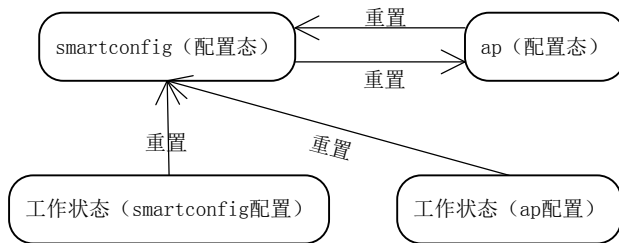
MCU 返回:

| 字段   | 长度 (byte) | 说明                      |
|------|-----------|-------------------------|
| 帧头   | 2         | 0x55aa                  |
| 版本   | 1         | 0x00                    |
| 命令字  | 1         | 0x03                    |
| 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 |

### 3.5 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   | 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         | 0x04          |
| 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 |
|--|--|------------|

### 3.6 重置 WIFI-选择模式

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

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   | xxxx      | 0x00: 进入 smartconifg 配置模式<br>其他: 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                     |

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   | 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 |

### 3.7 命令下发

说明:

- 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 | x       | 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 | x     | 对应于具体字符串            |
|       |          | enum                  | 0x04 | 1     | 枚举类型, 范围 0-255      |
|       |          | bitmap                | 0x05 | 1/2/4 | 长度大于 1 字节时, 大端表示    |
| len   | 2        | 长度对应于上面 “type” 说明的长度  |      |       |                     |
| value | 取决于 type | 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         | 0x06                    |
| 数据长度 | 2         | 取决于 “命令数据单元” 类型以及个数     |
| 数据   | xxxx      | “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 |

### 3.8 状态上报

说明:

- 1) datapoint 状态数据单元说明详见 “3.7 节”
- 2) “状态上报” 为异步处理协议, “状态上报” 触发机制有三类:
  - a. MCU 收到 “命令下发处理帧时, 正确执行相应 datapoint 命令, 再通过 “状态上报” 帧将变化后的 datapoint 状态发送至模块;
  - b. MCU 主动检测到 datapoint 有变化, 将变化后的 datapoint 状态发送至模块;
  - c. MCU 收到 “3.9” 节的状态查询帧时, 将所有的 datapoint 状态发送至模块。
- 3) “状态上报” 可含多个 datapoint “命令数据单元”

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         | 取决于 “状态数据单元” 类型以及个数 |
| 数据   | xxxx      | “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 |
|-----|---|-------------------------|

### 3.9 状态查询

说明:

- 1) “状态查询”为异步处理协议，主要用于模块查询 MCU 所有的“obj”类型的 datapoint 状态，当 MCU 收到此帧时，通过“3.8”节状态上报帧上报 datapoint 状态
- 2) “状态查询”发送时机主要有两种：
  - a. 模块首次上电，通过心跳与 MCU 建立连接后，查询发送；
  - b. 模块工作过程中检测到 MCU 重启或发生了离线再上线的过程，查询发送。

模块发送:

| 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         | 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 |

### 3.10 MCU 升级支持

说明:

- 1) MCU 可根据自身情况，选择性支持
- 2) 何时升级由 MCU 触发，模块仅作为支持 MCU 升级的数据传输通道
- 3) 升级支持强制升级以及手动升级模式，可通过涂鸦云开发平台上传升级固件时选择具体模式
- 4) 升级仅支持高版本升级低版本

#### 3.10.1 升级查询

MCU 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         | 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         | 0x09   |
| 数据长度 | 2         | 0x0001   |
| 数据   | xxxx      | 0x00: 不支持 MCU 升级<br>0x01: 模块未就绪<br>0x02: 云端升级信息查询失败<br>0x03: 无需升级 (云端无更新版本)<br>0x04: 需升级, 等待模块发起升级操作 |
| 校验和  | 1         | 从帧头开始按字节求和得出的结果对 256 求余  |

### 3.10.2 升级启动

说明：

- 1) 升级启动方式含自动及手动升级。当处于自动升级时，模块检测云端 MCU 有更新版本固件，则自动启动与 MCU 升级包交互流程；当处于手动升级时，通过 APP 确定，模块才启发与 MCU 升级包交互流程

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         | 0x0004                   |
| 数据   | xxxx      | 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  |

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    | 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 |

### 3.10.3 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 2 且包偏移 $\geq$ 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         | 0x0b                    |
| 数据长度 | 2         | 0x0002+包大小              |
| 数据   | xxxx      | 前两字节, 固定为包偏移            |
| 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         | 0x0b                    |
| 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 |

### 3.11 获取系统时间

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   | 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         | 0x0c   |
| 数据长度 | 2         | 0x0007   |
| 数据   | Data      | 数据长度为 7 字节:<br>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<br>Data[1]为年份, 0x00 表示 2000 年<br>Data[2]为月份, 从 1 开始到 12 结束<br>Data[3]为日期, 从 1 开始到 31 结束<br>Data[4]为时钟, 从 0 开始到 23 结束<br>Data[5]为分钟, 从 0 开始到 59 结束<br>Data[6]为秒钟, 从 0 开始到 15 结束 |
| 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 通知 MCU 进入产测模式

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         | 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 |

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   | 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 |