**通讯协议**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **说明** | **编者** | **时间** |
| V1.0 | 初稿 | gh | 2022/08/10 |
|  |  |  |  |
|  |  |  |  |

1. **概述**

通过本协议主要实现以下功能：

* 控制运行(停止、关机、重启)
* 获取或修改设置
* 获取设备状态
* 固件信息获取，固件升级
* 设备异常报告

本协议的所有数据均为1字节对齐，且在表中的从上到下的排序均为内存中由低到高的顺序。

1. **帧头定义**

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类型** | **变量名** | **说明** | |
| UINT32 | magic | 魔数，固定为：0xDEADBEEF | |
| UINT8 | type | 数据类型，详细说明见下 | 0：设备命令 |
| 1：设备状态 |
| 2：ADC数据 |
| 3：通讯回应 |
| 4：设置参数 |
| 5：全部参数 |
| 6：错误信息 |
| 7：固件升级 |
| 8：心跳包 |
| UINT8 | flag | 标志位，预留备用 |  |
| UINT8 | askAck | 是否要求对方ack，在要求ack时，固件层无  论执行正确与否都将进行回应，否则只在出错时回应 | |
| UINT16 | dataLen | 尾部data长度，不含checksum字节 |  |
| UINT8 | …… | 尾部数据 | 详见数据帧定义 |
| UINT8 | checksum | 校验和 | 本字节前的所有字节的累加和 |
|  |  |  |  |

注：该帧头是所有数据的公用帧头，不同数据通过type来进行区分

|  |  |
| --- | --- |
| **type** | **ack** |
| 0：设备命令 | 置位askAck时，则100毫秒内收不到回应则重发，连发**3次**则停发 |
| 1：设备状态 | 设备主动按一定频率上报，不置位askAck |
| 2：通讯回应 | 回应数据包，一般不置位askAck |
| 3：设置参数 | 置位askAck时，则100毫秒内收不到回应则重发，连发**3次**则停发 |
| 4：全部参数 | 双方均应置位askAck，收不到回应将一直发送 |
| 5：错误信息 | 该数据由设备发送，将置位askAck，收不到回应将一直发送 |
| 6：固件升级 | 升级数据由APP发给设备，每包应置位askAck，100毫秒内收不到回应或是收到错误则应重发，连发**3次**则停止发送并报错 |
| 7：心跳包 | 用来侦测对方是否还在线 |

1. **数据帧定义**
   1. **命令数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类型** | **变量名** | **说明** | |
| UINT8 | cmd | 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 | 设备重启 |
| 5 | 恢复出厂设置 |
| 6 | 设备关机。 |
| UINT32 | para |  | 命令对应的参数，不同命令，参数意义不同，见上面说明 |

* 1. **状态数据**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **变量名** | **说明** |
| UINT8 | sysState | 设备状态，取值有：  0：设备停止运行  1：设备工作中  2：设备升级状态  设备接收到升级包时，设备处于升级状态，在升级状态时将忽略除升级外的其它所有数据 |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. **回应数据**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **变量名** | **说明** |
| UINT8 | type | 指明该回应是针对哪种类型数据的回应，详见帧头定义中的type |
| UINT8 | error | 错误码，详见error定义文件 |

*说明：本次改动又恢复到之前的格式，主要原因是在调试升级过程中，光有回应还不够，如果数据出错或异常均需要APP重发，故又将其进行合并*

注：

1. 置位askAck时，接收方应不论执行成功与否，均需给予回应
2. 未置位askAck时，固件在执行命令时出错时会给予回应，回应数据中的error标识错误码，该回应格式与错误报告格式相同，所不同的是回应数据是用于主动通讯，而错误上报则是固件系统侦测错误的自动行为，两者的类型不同
3. 对于置位askAck的通讯，如果100ms内接收方无回应，则应重试**3次**，如果**3次**均无回应，则应报错并在界面予以呈现
4. 设备的状态由app控制发送节奏，app发送状态请求(后面空数据)，设备立即返回所有状态数据
   1. **错误数据**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **变量名** | **说明** |
| UINT8 | type | 指明该错误是针对哪种类型数据的错误，详见帧头定义中的type |
| UINT8 | error | 错误码，详见error定义文件 |

注：

该错误信息由设备主动上报，会置位askAck位，如果收不到回应将会一直发送

* 1. **全部参数数据**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **变量名** | **说明** |
| 固件信息 | fwInfo | 参见后面定义 |
| 全部设置参数 | setts | 待补充 |
|  |  |  |

注：增加全部参数传递，避免中途固件信息传递，升级时可以直接进行版本检查而无需向设备获取

* 1. **固件信息**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **变量名** | **说明** |
| UINT32 | magic | 固件魔数，固定为：0xFACEBEAD |
| UINT8, [20] | version | 固件版本，类型为字符串，依照已有的版本命令规范进行 |
| UINT8, [20] | buildTime | 固件构建时间，类型为字符串，如：20200408 |
|  |  |  |

* 1. **设置参数数据**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **变量名** | **说明** |
| UINT8 | mode | 待补充 |
|  | |  |
|  | |  |
|  | |  |
|  | |  |

说明：

1. 向设备发送参数数据时，后面需要带上参数数据；向设备请求参数数据时，后面无需带数据，据此可以判断读取还是写入, 根据后面参数的长度
2. 开机后会主动向APP发送全部设置参数，并索要回应，如果未收到回应，设备将一直发送，否则停止发送，同理APP刚运行起来时也应向设备索要全部设置参数直至收到参数，采取该方式的原因在任一方重启时可以保证两边数据同步
3. 用户修改设置参数后应立即发送给设备
   1. **升级文件头**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **变量名** | **说明** |
| UINT8, **[32]** | md5 | md5校验数据，即从固件信息开始到整个文件结束的md5码  需要检查该数据段，该数据与自己计算的数据不一致则报错 |
| 固件信息 | fwInfo | 详见第4页说明 |
| UINT8 | …… | 二进制固件数据 |

* 1. **固件信息**

详见第4页

说明：

APP可以根据文件固件信息头与参数固件信息进行比对，以决定是否进行升级

* 1. **固件数据帧**

|  |  |  |
| --- | --- | --- |
| **数据类型** | **变量名** | **说明** |
| UINT16 | pkts | 总包数 |
| UINT16 | pid | 包id号，循环递增 |
| UINT16 | dataLen | 尾数据的长度，单位为字节 |
| U8 | …… | 尾数据 |

说明：

1. 发送升级固件数据时，第一包数据应只发升级包帧头，不附具体固件数据，系统收到此帧后将重启进入Bootloader中进行升级操作，进入bootloader后会主动给APP发送一包设备状态的数据，用于通知APP：设备已经进入升级状态，APP可以继续发送固件数据了
2. 发送升级固件数据时，应从固件信息开始发送数据，单个数据包的**总长**(含包头、包尾)不能超过1000字节。发送第一包固件数据的ID号应为0，之后依次递增
3. APP在升级数据发送完毕后，应等待升级完成；升级成功后，设备将会重启(如有需要还将更新参数)，重启后，设备会主动向APP发送全部参数，APP在收到全部参数后，可以在界面上提示升级成功，并用新的参数来刷新界面
4. 因为单片机内存太小的缘故，无法将固件全部缓存下来，因此，一旦开始升级不可中途退出，否则将导致升级失败，重启后设备将会自动进入升级状态

