**数据协议V0.1（待完善）**

1. **指令协议说明**

在原本协议是用在自定义串口助手与下位机的通信测试所用，实现的功能是关灯与开灯。

**1.1指令帧格式**

固件指令由帧头、帧类型、指令代码、指令数据长度、指令参数、校验码和帧尾组成，均为十六进制表示。 例如：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
| BB | 00 | 01 | 00 | 01 | 03 | 05 | 7E |

帧头 Header: 0xBB

帧类型 Type: 0x00

指令代码 Command: 0x07

指令参数长度 PL: 0x0001

指令参数 Parameter: 0x01

校验位 Checksum: 0x05

帧尾 End: 0x7E

校验位 Checksum 为从帧类型 Type 到最后一个指令参数 Parameter 累加和，并只取累加和最低一个字节(LSB)。

**1.2指令帧类型**

|  |  |
| --- | --- |
| Type | Description |
| 0x00 | 命令帧: 由上位机发送给锁 |
| 0x01 | 响应帧: 由Cpu发回给上位机 |
| 0x02 | 通知帧: 由Cpu发回给上位机 |

每一条指令帧都有对应的响应帧。响应帧表示指令是否已经被执行了。

**2.指令定义**

**2.1 获取表1长度的指令**

2.1.1 命令帧定义

获取模块的软硬件版本和制造商信息

帧类型： 0x00

命令码： 0x02

参数：1字节 //3字节

默认0x00 保留字节

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

例：获取信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
| BB | 00 | 01 | 00 | 01 | 03 | 05 | 7E |

帧类型 Type: 0x00 /

指令代码 Command: 0x02 //代表设定表2的长度的命令

指令参数长度 PL: 0x0001

指令参数 Parameter: 0x03

校验位 Checksum: 0x05

2.1.2 响应帧定义

帧类型： 0x01

命令码： 0x01

参数：18字节 //20字节

锁唯一编码：12字节（原8字节）

软件版本号：2字节

电量： 2字节

随机数： 2字节

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

字符串表示：\*000100010305~

**2.2 设置表1的元素值**

2.2.1 命令帧定义

获取模块的软硬件版本和制造商信息

帧类型： 0x00

命令码： 0x02

参数：1字节 //3字节

默认0x00 保留字节

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

例：获取信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
| BB | 00 | 02 | 00 | 01 | 04 | 07 | 7E |

帧类型 Type: 0x00 /

指令代码 Command: 0x02 //代表设定表2的长度的命令

指令参数长度 PL: 0x0003

指令参数 Parameter: 0x04

校验位 Checksum: 0x07

2.2.2 响应帧定义

帧类型： 0x01

命令码： 0x02

参数：18字节 //20字节

锁唯一编码：12字节（原8字节）

软件版本号：2字节

电量： 2字节

随机数： 2字节

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

字符串表示：\*000300030407~

**2.2 设置表1的元素值**

2.2.1 命令帧定义

获取模块的软硬件版本和制造商信息

帧类型： 0x00

命令码： 0x03

参数：1字节 //3字节

默认0x00 保留字节

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

例：获取信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
| BB | 00 | 03 | 00 | 03 | 02 03 03 | 0e | 7E |

帧类型 Type: 0x00 /

指令代码 Command: 0x03 //代表设定表2的长度的命令

指令参数长度 PL: 0x0003

指令参数 Parameter: 0x00

校验位 Checksum: 0x0e

2.2.2 响应帧定义

帧类型： 0x01

命令码： 0x01

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

字符串表示：\*000300030203030e~

**2.3 设置表2的元素值**

2.3.1 命令帧定义

获取模块的软硬件版本和制造商信息

帧类型： 0x00

命令码： 0x03

参数：1字节 //3字节

默认0x00 保留字节

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

例：获取信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
| BB | 00 | 04 | 00 | 04 | 01 03 03 04 | 13 | 7E |

帧类型 Type: 0x00 /

指令代码 Command: 0x03 //代表设定表2的长度的命令

指令参数长度 PL: 0x0004

指令参数 Parameter: 0x01 0x03 0x03 0x04

校验位 Checksum: 0x13

2.3.2 响应帧定义

帧类型： 0x01

命令码： 0x04

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

字符串表示：\*000400040103030413~

**2.2 合并**

2.1.1 命令帧定义

获取模块的软硬件版本和制造商信息

帧类型： 0x00

命令码： 0x05

参数：1字节 //3字节

默认0x00 保留字节

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

例：获取信息

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | Type | Command | PL(MSB) | PL(LSB) | Parameter | Checksum | End |
| BB | 00 | 05 | 00 | 01 | 00 | 0x06 | 7E |

帧类型 Type: 0x00 /

指令代码 Command: 0x05 //代表设定表2的长度的命令

指令参数长度 PL: 0x0001

指令参数 Parameter: 0x00

校验位 Checksum: 0x06

2.1.2 响应帧定义

帧类型： 0x01

命令码： 0x05

参数：18字节 //20字节

锁唯一编码：12字节（原8字节）

软件版本号：2字节

电量： 2字节

随机数： 2字节

//CRC16： 2字节（暂未添加此功能，因原数据定义有报文数据外部校验错误和内部错误两种结果码，可以考虑在负载中使用CRC校验作为内部校验，外部由数据包校验）

字符串表示：\*000500010006~