# 第2章 通信协议

## 2.1 通信方式

## 2.2 数据格式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 帧头  （1Byte） | 从机地址（1Byte） | 命令组（1Byte） | 命令码(1Byte) | 数据长度(1Byte) | 数据  (NByte) | 校验和  (1Byte) |
| 固定为0xA6 | 根据实际地址填写 |  |  | 0~255 |  | 帧头到数据的校验和 |

PS:命令组表示不同功能大类的命令，命令码为某一命令组下面的功能码

通信中涉及到多个字节表示一个数值时,高字节在前。

从机地址：取低6位作为地址,范围为0~63。

|  |  |  |
| --- | --- | --- |
| 高2位 | 从机地址意义 | 备注 |
| 00 | 表示实际地址 | 适用于问答式同步通信 |
| 10 | 低6位表示通信的帧ID，应答的帧ID要与请求的帧ID一致，用来区别当前应答的是哪个命令 | 适用于异步通信，应答的帧标识ID来区分应答的是哪条命令； |

M:主机设备

S:从机设备

### 2.5.2 上报报警信息(完成)

|  |  |
| --- | --- |
| 命令 | 0x01,0x11 |
| 长度 | 6 |
| 数据区 | typedef struct  {  /\*  0x00 无报警  0x01 防拆  0x02 胁迫  0x03 钥匙开锁  0x04 低电压  0x05 假锁报警  0x06：门铃事件  0x07：静默报警1  0x08：静默报警2  0x09：防拆报警  0x0a：非法用户  \*/  uchar ucAlarmCode;  /\*  0x01 START  0x02 STOP  0x03 PULSE  \*/  uchar ucAction;  /\*动作时间\*/  ulong ulTime;  }ALARMINFO\_S; |
| 备注 |  |



### 2.5.18 远程记录查询（含风险完成）

|  |  |
| --- | --- |
| 命令 | 0x10,0x41 |
| 长度 | 1 |
| 数据区 | Typedef struct BtRecReq // 远程记录请求  {  BYTE btType; // 事件类型  // 0x00：门铃事件  // 0x01：静默报警1  // 0x02：静默报警2  // 0x03：防拆报警  }TBtRecReq,\*LpBtRecReq; |
| 备注 |  |

|  |  |
| --- | --- |
| 命令 | 0x01,0x17 |
| 长度 | n |
| 数据区 | Typedef struct RecInfo // 记录信息  {  DWORD dwRecTime; // 记录的时间 UTC  }TRecInfo,\*LPRecInfo;  Typedef struct BtRecRes // 记录返回  {  BYTE btCount; // 记录条目  BYTE btRecType; // 事件类型  TRecInfo Record[n]; // 记录类型 见 具体结构  }TBtRecRes,\*LpBtRecRes; |
| 备注 |  |



### 2.5.20 用户关锁/应答( 完成)

|  |  |
| --- | --- |
| 命令 | 0x10,0x43 |
| 长度 | 12 |
| 数据区 | Typedef struct BtUnlock // 用户关锁请求  {  BYTE btDevMac[6]; // 关锁用户的mac（蓝牙无需MAC）  BYTE btDelay; // 关锁延时  BYTE btTimeZone; // 时区  DWORD dwCurTime; // 关锁当前时间 （用于校时 ）UTC  }TBtUnlock,\*LpBtUnlock; |
| 备注 |  |

|  |  |
| --- | --- |
| 命令 | 0x01,0x18 |
| 长度 | 1 |
| 数据区 | Typedef struct BtUnlockRes // 锁体关锁应答  {  BYTE btRst;  }TBtUnlockRes,\*LpBtUnlockRes; |
| 备注 |  |



### 2.5.21无线报警控制(完成)

|  |  |
| --- | --- |
| 命令 | 0x10,0x44 |
| 长度 | 2 |
| 数据区 | Typedef struct BtWLControlReq  {  BYTE btType; // 报警类型 0：急促  BYTE btAction; // 动作类型 0：停止 1：开始  }TBtWLControlReq,\*Lp BtWLControlReq; |
| 备注 |  |



### 2.5.22 用户删除设备(完成)

|  |  |
| --- | --- |
| 命令 | 0x06,0x12 |
| 长度 | 7 |
| 数据区 | Typedef struct BtDelDev // 用户设备删除  {  BYTE bDelType; // 删除方式：0-单独 1-指定 2-所有  BYTE btDevMac[6]; // 对应删除mac（删除全部时，不用填）  }TBtDelDev,\*LpDelDev; |
| 备注 |  |



### 2.5.23 用户开锁/应答（完成）

|  |  |
| --- | --- |
| 命令 | 0x06,0x20 |
| 长度 | 12 |
| 数据区 | Typedef struct BtUnlock // 用户开锁请求  {  BYTE btDevMac[6]; // 开锁用户的mac  BYTE btDelay; // 开锁保持时间  BYTE btTimeZone; // 时区  DWORD dwCurTime; // 开锁当前时间 （用于校时 ）UTC  }TBtUnlock,\*LpBtUnlock; |
| 备注 |  |

|  |  |
| --- | --- |
| 命令 | 0x06,0x21 |
| 长度 | 22 |
| 数据区 | Typedef struct BtUnlockRes // 锁体开锁应答  {  BYTE btRst;  BYTE btBatState; // 电池电量  DWORD dwPswd[5]; // 分享临时 秘钥 5组 8位秘钥 (十六进制表示)  }TBtUnlockRes,\*LpBtUnlockRes; |
| 备注 |  |



### 2.5.24开锁记录查询（完成）

|  |  |
| --- | --- |
| 命令 | 0x06,0x22 |
| 长度 | 09 |
| 数据区 | Typedef struct BtRecReq // 开锁记录请求  {  BYTE btType; // 查询 方式  // 0x00: 时间段  // 0x 01：当日  // 0x 02：昨日  // 0x03: 最近100条  // 0x04: 最近200条  // 0x 10：所有 （暂定500条）  DWORD dwStartTime; // 起始时间  DWORD dwEndTime; // 结束时间  }TBtRecReq,\*LpBtRecReq; |
| 备注 |  |

|  |  |
| --- | --- |
| 命令 | 0x06,0x23(0x24) |
| 长度 | n |
| 数据区 | Typedef struct RecInfo // 开锁信息  {  BYTE btType; // 记录类型 0：刷卡 1：密码 2：指纹 3：蓝牙  DWORD dwRecTime; // 记录的时间 UTC  DWORD dwId; // 操作设备 对应为：卡号、密码、指纹ID、蓝牙MAC低4字节  }TRecInfo,\*LPRecInfo;  Typedef struct BtRecRes // 开锁记录返回  {  BYTE btCount; // 记录条目  TRecInfo Record[n]; // 记录类型 见 具体结构  }TBtRecRes,\*LpBtRecRes; |
| 备注 | 开锁记录开始和结束 仅指令不同，内容一致。 |



### 2.5.25智能锁版本查询/应答（完成）

|  |  |
| --- | --- |
| 命令 | 0x06,0x60 |
| 长度 | 0 |
| 数据区 |  |
| 备注 |  |

|  |  |
| --- | --- |
| 命令 | 0x06,0x61 |
| 长度 | 13 |
| 数据区 | Typedef struct UnlockVerRes // 锁体版本信息应答  {  BYTE btDevType; // 设备类型  BYTE btIcType; // 芯片类型  //CHAR strVersion[]; // 版本信息  //General\_VTH1X0XX\_MCU\_V5.210.0000.0.R.150724  BYTE btMVer; // 主版本  BYTE btSVer; // 次版本  WORD wOEMVer; // OEM 版本  BYTE btEVer; // 修订 版本  BYTE btBootVer; // boot 版本 （预留）  BYTE btYear; // 年  BYTE btMonth; // 月  BYTE btDay; // 日  }TUnlockVerRes, \*LpUnlockVer; |
| 备注 |  |



### 2.5.28 应答

|  |  |
| --- | --- |
| 命令 | 0x20,0x51 |
| 长度 | 2 |
| 数据区 | typedef struct  {  uchar ucCmd; //表示应答的是什么命令  uchar ucRet; //是否成功 0x01-成功,0x02失败  }COMACK\_S |
| 备注 |  |

## 2.7蓝牙模块命令解析

### 2.7.1 测试/测试应答(完成)

|  |  |
| --- | --- |
| 命令 | 0x06,0x01 |
| 长度 | 0 |
| 数据区 | Typedef struct BtTest  {  BYTE btResver; // 保留  }TBtTest,\*LpBtTest; |
| 备注 | 智能锁通知蓝牙模块进入测试 |

|  |  |
| --- | --- |
| 命令 | 0x06,0x02 |
| 长度 | 1 |
| 数据区 | Typedef struct BtTestRes  {  BYTE btState; // 状态  }TBtTestRes,\*LpBtTestRes; |
| 备注 | 蓝牙模块应答测试 |



### 2.7.2 复位/复位应答(完成)

|  |  |
| --- | --- |
| 命令 | 0x06,0x05 |
| 长度 | 1 |
| 数据区 | Typedef struct BtReset  {  BYTE btTime; // 复位延迟 时间 秒  }TBtReset,\*LpBtReset; |
| 备注 | 智能锁通知蓝牙模块进入复位 |

|  |  |
| --- | --- |
| 命令 | 0x06,0x06 |
| 长度 | 1 |
| 数据区 | Typedef struct BtResetRes  {  BYTE btState; // 状态  }TBtResetRes,\*LpBtResetRes; |
| 备注 | 蓝牙模块ACK复位 |



### 2.7.3 已绑定蓝牙设备获取/应答(完成)

|  |  |
| --- | --- |
| 命令 | 0x06,0x03 |
| 长度 | 0 |
| 数据区 |  |
| 备注 |  |

|  |  |
| --- | --- |
| 命令 | 0x06,0x04 |
| 长度 | n |
| 数据区 | Typedef struct BDevMacRes  {  BYTE btMacCount; // MAC数量  BYTE btMac[n][6]; // MAC 内容  }TBDevMacRes,\*LpBDevMacRes; // MAC 列表应答 |
| 备注 |  |



### 2.7.4 入网许可(完成)

|  |  |
| --- | --- |
| 命令 | 0x06,0x10 |
| 长度 | 1 |
| 数据区 | Typedef struct BtNetAllow  {  BYTE bFlag; // 运行入网 0：结束；1：开始  }TBtNetAllow,\*LpBtNetAllow; |
| 备注 |  |



### 2.7.5 新入网通知(完成)

|  |  |
| --- | --- |
| 命令 | 0x06,0x11 |
| 长度 | 7 |
| 数据区 | Typedef struct BtNewDev // 新用户设备入网  {  BOOL bNew; // 是否为新 设备 （0：已存在， 1：新入）  BYTE btDevMac[6]; // 设备MAC 地址  }TBtNewDev,\*LpBtNewDev; |
| 备注 |  |



### 2.7.6 蓝牙注册请求/应答(与入网许可重复)

|  |  |
| --- | --- |
| 命令 | 0x06,0x25 |
| 长度 | 1 |
| 数据区 | Typedef struct BtReg  {  BYTE btFlag; // 0:注册开始, 1:注册结束  }T BtReg,\*Lp BtReg; |
| 备注 |  |

|  |  |
| --- | --- |
| 命令 | 0x06,0x26 |
| 长度 | 1 |
| 数据区 | Typedef struct BtRegRes  {  BYTE btFlag; // 0:蓝牙进入绑定状态, 1:蓝牙没进入绑定状态  }TBtReg Res,\*LpBtReg Res; |
| 备注 |  |



### 2.7.7蓝牙注册状态通知（与新入网通知重复）

|  |  |
| --- | --- |
| 命令 | 0x06,0x26 |
| 长度 | 7 |
| 数据区 | Typedef struct BtRegNotify  {  BYTE btState; // 0：新绑定,1：已绑定,2：超时  BYTE btMac[6]; // MAC地址  }TBtRegNotify,\*LpBtRegNotify; |
| 备注 |  |



### 2.7.8蓝牙密钥设置(6位密钥)

|  |  |
| --- | --- |
| 命令 | 0x06,0x28 |
| 长度 | 6 |
| 数据区 | Typedef struct BtKeyReq  {  BYTE btKey[6]; // 密钥  }T BtKeyReq,\*Lp BtKeyReq; |
| 备注 |  |



### 2.7.9蓝牙配对码设置(int32位整型)

|  |  |
| --- | --- |
| 命令 | 0x06,0x30 |
| 长度 | 4 |
| 数据区 | Typedef struct BtCodeReq  {  DWORD dwCode; // 密钥  }T BtCodeReq,\*Lp BtCodeReq; |
| 备注 |  |



### 2.7.11通用应答(完成)

|  |  |
| --- | --- |
| 命令 | 0x06,0x80 |
| 长度 | 2 |
| 数据区 | Typdef struct CommAck // 通用应答  {  BYTE btGroupCmd; // 针对的指令  BYTE btCmd; // 针对的指令  BYTE btRst; // 结果 0：失败 1：成功 2：  }TCommAck,\*LpCommAck; |
| 备注 | 通用应答 用于对特定指令执行结果的返回： |