

# 门锁通讯协议

V1.6

## 版权声明

本手册版权归属深圳市飞比电子科技有限公司（简称“飞比科技”）所有，并保留一切权利。非经飞比科技**书面同意**，任何单位及个人不得擅自摘录本手册部分或全部内容。

## 免责声明

由于产品版本升级或其他原因，本手册内容会不定期更新。除非另有约定，本手册仅作为使用指导，本手册所有陈述、信息和建议不构成任何明示或暗示的担保。

## 商标声明



为深圳市飞比电子科技有限公司的商标。本文提及其他所有商标和注册商标，归各自的所有人所有。

## 版本信息

版本	时间	更新内容	更新者
V1.1		飞比与中控讨论版本	Daniel
V1.2	2016.08.29	所有协议增加回复，增加门锁主动获取时间 修改本地开锁上报（包含胁迫报警）	Daniel
V1.3	2016.09.04	增加启用开门常开/取消门锁常开/双人验证/胁迫报警/ 验证进入管理员菜单等状态到本地开锁上报 兼容老的胁迫报警协议 0x25 命令 修改模块发送给门锁的常开命令到 0x86，原 0x53	Parke
V1.4	2016.09.29	远程开锁回复命令，增加开锁持续时间	Parke
V1.5	2016.11.30	修正老胁迫报警命令中用户 ID 解释错误问题 增加部分的详细备注描述，增加远程加密方法描述 增加报警解除的命令	Ricky Nicholas
V1.6	2018.03.20	修改页眉页脚	Tina

## 目录

1	接口定义.....	1
2	数据交互.....	1
3	数据包格式.....	2
3.1	发送数据包.....	2
3.2	回复确认包.....	2
4	具体通讯数据.....	3
4.1	无线加网（门锁发送给模块）.....	3
4.2	无线退网（门锁发送给模块）.....	3
4.3	非法操作报警（门锁发送给模块）.....	4
4.4	强拆报警（门锁发送给模块）.....	4
4.5	假锁报警（门锁发送给模块）.....	5
4.6	未关门报警（门锁发送给模块）.....	5
4.7	胁迫报警（门锁发送给模块）.....	6
4.8	低压报警（门锁发送给模块）.....	6
4.9	本地开锁上报（门锁发送给模块）.....	7
4.10	门锁唤醒模块（门锁发送给模块）.....	8
4.11	远程密码开锁（模块发送给门锁）.....	9
4.12	远程密码开锁回复（门锁发送给模块）.....	10
4.13	时间同步（模块发送给门锁）.....	10
4.14	模块入网退网提示（模块发送给门锁）.....	11
4.15	模块网络状态显示（模块发送给门锁）.....	12
4.16	门锁获取时间（门锁发送给模块）.....	12
4.17	门锁常开（模块发送给门锁）.....	13
4.18	报警解除（门锁发送给模块）.....	14
附录 1	UTC 时间转换公式.....	14
附录 2	远程开锁密码加密解密方法.....	18

## 1 接口定义

需提供 6 线接口:

1: VCC (电池正)

2: GND (电池负)

3: 唤醒输出 (无线模块对锁主板的唤醒) P0.6 默认为低

4: 唤醒输入 (锁主板对无线模块的唤醒) P0.7 上拉

5: TXD (串口发送信号)

6: RXD (串口接收信号)

通讯参数: 波特率 57600, 数据位 8, 无校验, 1 停止位, 无流控。

唤醒信号线及串口都为 3.3V 电平。

考虑到功耗问题, 唤醒信号线需上拉为高电平。

ZIGBEE 模块与电子锁主板采用串口通讯, 进行数据交互。

ZIGBEE 模块, 以下简称模块, 门锁电路板, 以下简称门锁。

## 2 数据交互

门锁发送数据给模块:

1) 门锁先给 WakeUp\_Mod 一个 10ms 的高电平来唤醒模块, 20ms 之后在发送串口数据;

2) 模块接收到正确协议数据后, 回复“确认指令”协议数据, 若接收协议数据不正确, 回复“确认错误指令”数据协议, 再次等待门锁发送数据, 若三次接收数据错误则自动进入睡眠;

3) 门锁接收到“确认正确指令”协议数据后, 进入休眠状态; 若接收到“确认错误指令”协议数据或没有收到回复确认数据, 则再次发送协议数据, 三次重发, 每次间隔 100ms, 三次重发结束门锁也进入睡眠。(重发时不需要再发激活信号)

模块发送数据给门锁:

1) 模块先给 WakeUp\_MCU 一个 250ms 的高电平唤醒门锁之后直接发送数据, 之后

再每 800ms 向门锁发送串口数据，模块每次发送数据前均会发送激活信号 250ms；

2) 模块发送 7 次每 800ms 发送一次，若收到回复确认指令就停止发送进入睡眠，若失败或者没有收到则 7 次发送完成后模块进入睡眠。

### 3 数据包格式

协议定义如下：

协议数据为十六进制。

#### 3.1 发送数据包

开始标记 0xAA(1)	有效数据 长度 (1)	命 令 字 (1)	命令字自 增 ID(4)	应答(1)	有效数据(N)	异或校验(1)	结束标记 0x55(1)
-----------------	----------------	--------------	-----------------	-------	---------	---------	-----------------

开始标记：固定为 0xaa,二进制为 10101010,在电路上表现为整齐的峰谷峰谷波形，容易与干扰数据做区别。

命令字：执行操作的命令。

命令字自增 ID：每次命令自增 1，保证每条命令是唯一的，保证发出去的命令和收到的应答是对应的，范围 0x00000001~0xFFFFFFFF；注意此字段为（低字节在后，高字节在前）；

应答：0x00 表示发送数据包

有效数据长度：有效数据部分的字节数 N

有效数据：具体的操作数据(低字节在前，高字节在后)。

异或校验：对从“开始标记”开始到“有效数据”结束的内容进行异或计算的结果值。

结束标记：固定为 0x55，二进制为 1010101

#### 3.2 回复确认包

开始标记 0xAA(1)	有效数据 长度 (1)	命 令 字 (1)	命令字自 增 ID(4)	应答(1)	有效数据(1)	异或校验(1)	结束标记 0x55(1)
-----------------	----------------	--------------	-----------------	-------	---------	---------	-----------------

开始标记：固定为 0xaa,二进制为 10101010,在电路上表现为整齐的峰谷峰谷波形，容易与干扰数据做区别。

命令字：执行操作的命令，与数据包的命令字一致。

命令字自增 ID：每次命令自增 1，保证每条命令是唯一的，保证发出去的命令和收到的应答是对应的，该 ID 与接收到的数据包 ID 保持一致（对应的数据包和确认包的 ID 一致）

应答：0x01 表示回复确认包

有效数据长度：有效数据部分的字节数，默认 0x01

有效数据：0x00 表示接收成功，0x01~0xFE 表示接收失败。

异或校验：对从“开始标记”开始到“有效数据”结束的内容进行异或计算的结果值。

结束标记：固定为 0x55，二进制为 1010101

## 4 具体通讯数据

### 4.1 无线加网（门锁发送给模块）

表 4-1-1 无线加网数据包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x0A	0x40	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0xE1	0x55

命令字为 0x40，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0。

表 4-1-2 无线加网回复包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x01	0x40	0x00 00 00 01	0x01	0x00	0xEB	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

### 4.2 无线退网（门锁发送给模块）

表 4-2-1 无线退网数据包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)

0xAA	0x0A	0x41	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0xE0	0x55
------	------	------	---------------	------	---------------------------------	------	------

命令字为 0x41，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0。

表 4-2-2 无线退网回复包

开始 标记 (1)	有效数 据长度 (1)	命 令 字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x01	0x41	0x00 00 00 01	0x01	0x00	0xEA	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

### 4.3 非法操作报警（门锁发送给模块）

表 4-3-1 非法操作数据包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x0A	0x20	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0x81	0x55

命令字为 0x20，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0。

表 4-3-2 非法操作回复包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x01	0x20	0x00 00 00 01	0x01	0x00	0x8B	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

### 4.4 强拆报警（门锁发送给模块）

表 4-4-1 强拆报警数据包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x0A	0x22	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0x83	0x55

命令字为 0x22，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0。

表 4-4-2 强拆报警回复包



开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异 惑 校 验 (1)	结束标记 (1)
0xAA	0x01	0x22	0x00 00 00 01	0x01	0x00	0x89	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

## 4.5 假锁报警（门锁发送给模块）

表 4-5-1 假锁报警数据包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x0A	0x23	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0x82	0x55

命令字为 0x23，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0。

表 4-5-2 假锁报警回复包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结 束 标 记 (1)
0xAA	0x01	0x23	0x00 00 00 01	0x01	0x00	0x88	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

## 4.6 未关门报警（门锁发送给模块）

表 4-6-1 未关门数据包

开始 标记 (1)	有效数 据长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x0A	0x24	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0x85	0x55

命令字为 0x24，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0。

表 4-6-2 未关门回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异 惑 校 验 (1)	结束标记 (1)
0xAA	0x01	0x24	0x00 00 00 01	0x01	0x00	0x8F	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

## 4.7 胁迫报警（门锁发送给模块）

表 4-7-1 胁迫报警数据包

开始标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x0A	0x25	0x00 00 00 01	0x00	0x00 01 03 01 00 00 00 00 00 00	0x87	0x55

命令字为 0x25，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位上传：用户编号 0x0100（2byte）（低字节在前，高位在后），验证方式(1byte，1 为密码，2 为卡，3 为指纹)以及电量（1byte，电量为 01-04，表示电量从高到低，04 代表低压，作用同低压报警）。表中有效数据含义：用户编号 0x0100，验证方式 0x03，电量 0x01，其他数据保留默认为 0。（注：此条协议仅为兼容老版本智能锁，此协议有缺陷如没有用户编号，开锁时长，没有开锁时间等，并且发送此条后还必须发送一条本地开锁记录，因此将本条与本地开锁上报整合到一起，新锁设计请参考本地开锁上报中的胁迫报警协议，不再使用此条老协议）

表 4-7-2 胁迫报警回复包

开始标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x01	0x25	0x00 00 00 01	0x01	0x00	0x8E	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

## 4.8 低压报警（门锁发送给模块）

表 4-8-1 低压报警数据包

开始标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)

0xAA	0x0A	0x30	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0x91	0x55
------	------	------	---------------	------	---------------------------------	------	------

命令字为 0x30，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0。

表 4-8-2 低压报警回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异 惑 校 验 (1)	结束标记 (1)
0xAA	0x01	0x30	0x00 00 00 01	0x01	0x00	0x9B	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。（此条可尽量与本地开锁上报一起发送）

## 4.9 本地开锁上报（门锁发送给模块）

表 4-9-1 本地开锁数据包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异 惑 校 验 (1)	结束标记 (1)
0xAA	0x0A	0x80	0x00 00 00 01	0x00	0x01 00 02 04 05 98 c0 1b 7d 1f	0x02	0x55

命令字为 0x80，应答为 0x00；命令字自增 ID 为 0x00000001（高位在前，低位在后）；有效数据分别包括：开锁的用户编号 0x0001（2byte）（高位在后，低位在前）；验证方式 0x02（1byte，1 为密码，2 为卡，3 为指纹，4 多重验证）（多重验证：是指要求用户开门时必须多次验证，比如要求必须使用该用户的指纹和密码同时验证通过才能开锁）；电量等级 0x04（1byte，电量为 01-04，表示电量从高到低，04 代表低压，作用同低压报警）；开锁持续时间 0x05（1byte，单位秒，可以为 0x00）（为开锁后门锁隔多长时间自动关闭的时间，模块会在此时间后自动发送门锁关闭数据包给网关，如果此字节为 0x00：则 Zigbee 模块不会自动发送门锁关闭数据包给网关，节省电量）；门锁状态（1byte，0x98=10011000 位模式，最高位 Bit7：胁迫报警，Bit4：双人验证模式，Bit3：验证管理员进入菜单，Bit1：取消门锁常开，Bit0：启用门锁常开；）（启用门锁常开时：开锁持续时间必须发 0x00，模块不能再往网关发送已关锁信息；取消门锁常开时：门锁会在开锁持续时间后关锁，可以通过管理员菜单取消门锁常开，或在常开状态下使用正确的指纹密码卡验证成功后同时取消门锁常开状态）（双人验证模式：门锁默认为单人验证模式，如果用户选择使用双人验证模式，则要求必须两个用户同时验证通过后方能开锁，不限制用户验证方式，但必须两个用户同时验证通过，比如 A 用户指纹与 B 用户密码同时验证通过，门锁发送两条

开锁记录给模块，但这两条开锁记录会标记为双人验证模式）（Bit3 置位代表此条信息不是开锁记录而是验证管理员进入菜单）（Bit1 取消与 Bit0 启用门锁常开不能同时置位）；时间戳 0x1F7D1BC0（4byte，UTC 时间，低位在前，高位在后）。有效数据：用户编号 0x0001，验证方式 0x02，电量 0x04，开锁持续时间 0x05，门锁状态（位模式 0x98：验证进入管理员菜单时，胁迫报警，双人验证模式）（0x11：双人验证模式，启用门锁常开模式）（0x12：双人验证模式，取消门锁常开模式）（0x88：验证进入管理员菜单时，胁迫报警，单人验证模式）（0x90：普通开门记录，胁迫报警，双人验证模式），时间戳：0x1F7D1BC0（低位在前，高位在后）（2016/08/27 12/00/00）UTC 时间转换，具体转换公式请看附录 1。有效数据：用户编号 0x0001，验证方式 0x02，电量 0x04，开锁持续时间 0x05，门锁状态含义（位模式：0x11：双人验证模式，启用门锁常开模式）（0x12：双人验证模式，取消门锁常开模式）（0x88：验证进入管理员菜单时，胁迫报警，单人验证模式）（0x98：验证进入管理员菜单时，胁迫报警，双人验证模式）（0x90：普通开门记录，胁迫报警，双人验证模式），时间戳：0x1F7D1BC0（2016/08/26 12/00/00）UTC 时间转换，具体转换公式请看附录 1。

表 4-9-2 本地开锁回复包

开始标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x01	0x80	0x00 00 00 01	0x01	0x00	0x2B	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

备注：状态字节含义，组合含义如下：（用户编号，验证方式，电量等按实际情况填写）

## 4.10 门锁唤醒模块（门锁发送给模块）

表 4-10-1 门锁唤醒模块数据包

开始标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标记 (1)
0xAA	0x0A	0x77	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0xD6	0x55

命令字为 0x77，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0。

在远程操作前，需先唤醒模块才能远程操作，模块唤醒后的一分钟内若接收到正确操作指令，待处理完成操作命令 2 秒之后进入睡眠，若没有收到任何操作则一分钟之后自动进

入睡眠。

表 4-10-2 门锁唤醒模块回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异感 校验 (1)	结束标 记 (1)
0xAA	0x01	0x77	0x00 00 00 01	0x01	0x00	0xDC	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xFE 表示接收数据失败。

## 4.11 远程密码开锁（模块发送给门锁）

表 4-11-1 远程密码开锁数据包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异感 校验 (1)	结束标 记 (1)
0xAA	0x0A	0x60	0x00 00 00 01	0x00	0x01 02 03 04 05 06 00 00 00 00	0xC6	0x55

命令字为 0x60，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位：目前密码长度为 6byte，其他为默认为 0。模块及门锁接收到的 6 位密码是经过加密的，请参考本文档最后部分附录 2；

表 4-11-2 远程密码开锁回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异感 校验 (1)	结束标 记 (1)
0xAA	0x01	0x60	0x00 00 00 01	0x01	0x00	0xCB	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xFE 表示接收数据失败。

远程开锁命令过程如下：

1. 门锁发送 0x77 命令唤醒模块，模块回复确认；
2. 模块发送 0x84 显示网络状态，门锁回复确认；
3. 模块醒来一分钟，等待 APP 端输入密码；
4. 然后 APP 端输入密码，发送给模块，模块再发送给门锁（0x60），门锁回复确认；
5. 门锁判断密码是否正确，发送远程开锁回复（0x54），模块回复确认。

6. 如果密码正确，模块直接进入休眠；否则，模块继续等待 APP 端的密码，直到超时。

## 4.12 远程密码开锁回复（门锁发送给模块）

表 4-12-1 远程密码开锁回复数据包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结束标 记 (1)
0xAA	0x0A	0x54	0x00 00 00 01	0x00	0x00 01 00 00 00 00 00 00 00 00	0xF1	0x55

命令字为 0x54，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位上传门锁状态 0x00(1byte, 00 表示密码正确，01 为密码错误，02 为远程开锁未允许(锁本地拨码开关未打上))，用户编号 0x0001(2byte)，保留 0x00(1byte)，开锁持续时间 0x05(1byte，单位秒，此字节直接填 0x00 代表不上传门锁开启时间)，其他字节都为 0x00。有效数据：远程开锁成功 (0x00)，用户编号 (0x0001)，保留字节 (0x00)，开锁持续时间 (0x05)；(备注：除了模块在未收到确认包会重发外，网关为了可靠也会把此包发给模块及门锁 3 次 (间隔 500ms 每次)，因此对于此数据包，门锁回复确认包后可能还会收到模块的重发包，这些重发包都需要回复确认包，但不需要重复回复此条开锁成功与失败的数据包给模块，避免 APP 和服务端重复记录开锁数据)

表 4-12-2 远程密码开锁回复确认包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异 惑 校 验 (1)	结束标 记 (1)
0xAA	0x01	0x54	0x00 00 00 01	0x01	0x00	0xFF	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xFE 表示接收数据失败。

## 4.13 时间同步（模块发送给门锁）

表 4-13-1 时间同步数据包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑 校验 (1)	结 束 标 记 (1)
-----------------	----------------	------------	-----------------	-----------	-------------	-----------------	----------------



0xAA	0x0A	0x62	0x00 00 00 01	0x00	0xE0 07 07 1A 0B 1E 2D 00 00 00	0x01	0x55
------	------	------	---------------	------	---------------------------------	------	------

命令字为 0x62, 应答为 0x00, 命令字自增从 0x01 开始累加, 有效数据位包括年(2byte), 月(1byte), 日(1byte), 时(1byte), 分(1byte), 秒(1byte)。

有效数据: 2016 年(0x07E0), 7 月(0x07), 26 日(0x1A), 12 时(0x0B), 30 分(0x1E), 45 秒(0x2D)。

表 4-13-2 时间同步回复包

开始 标记 (1)	有效数据 长度(1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验(1)	结 束 标 记(1)
0xAA	0x01	0x62	0x00 00 00 01	0x01	0x00	0xC9	0x55

命令字, 命令字自增与接收到的协议数据包一致, 应答为 0x01, 有效数据位 0x00 表示数据接收成功, 0x01~0xFE 表示接收数据失败。

备注: 此条命令会在门锁向模块发送门锁获取时间命令后的几秒内回复。

## 4.14 模块入网退网提示(模块发送给门锁)

表 4-14-1 模块入网退网提示数据包

开始 标记 (1)	有效数据 长度(1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验(1)	结束标 记(1)
0xAA	0x0A	0x83	0x00 00 00 01	0x00	0x01 01 00 00 00 00 00 00 00 00	0x22	0x55

命令字为 0x83, 应答为 0x00, 命令字自增从 0x01 开始累加, 操作方法: 1byte 有效数据位, 入网操作或退网操作(1byte, 0x00 入网, 0x01 退网), 反馈状态: 1byte 有效数据, 代表成功还是失败(1byte, 0x00 成功, 0x01 失败), 其它字节默认为 0。有效数据位: 退网操作 0x01, 退网失败 0x01。

- ◆ 模块收到无线入网命令(4.1)后, 会持续找网一分钟, 一分钟之内若入网成功则会立即发送入网成功指令给门锁(有效数据 0x00 00 00 00 00 00 00 00 00 00); 若在一分钟之内没有入网成功则会在一分钟结束时发送入网失败指令(有效数据 0x00 01 00 00 00 00 00 00 00 00);
- ◆ 模块在收到无线退网命令(4.2)后, 会在 5 秒之后发送退网状态给门锁。(模块自己清除入网状态并自动重启, 网关通过心跳判断是否在线, APP 显示离线需 1 小时以上;)

表 4-14-2 模块入网退网提示回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束标 记 (1)
0xAA	0x01	0x83	0x00 00 00 01	0x01	0x00	0x28	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xFE 表示接收数据失败。

## 4.15 模块网络状态显示（模块发送给门锁）

表 4-15-1 模块网络状态提示数据包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束标 记 (1)
0xAA	0x0A	0x84	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0x25	0x55

命令字为 0x84，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位上传网络状态（1byte，0x00 模块在线，0x01 模块掉线），其它字节默认为 0；

模块在每次被唤醒和心跳时都会去检测网络是否在线，若在线这回发送在线命令给门锁，若掉线则会发送掉线命令。门锁可以将在线状态告知用户。

表 4-15-2 模块网络状态提示回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束标 记 (1)
0xAA	0x01	0x84	0x00 00 00 01	0x01	0x00	0x2F	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xFE 表示接收数据失败。

## 4.16 门锁获取时间（门锁发送给模块）

表 4-16-1 门锁获取时间数据包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束 标记 (1)
0xAA	0x0A	0x85	0x00 00 00 01	0x00	0x00 00 00 00 00 00 00 00 00 00	0x24	0x55

命令字为 0x85，应答为 0x00，命令字自增从 0x01 开始累加，有效数据位全为 0；模块在收到该命令后，会先从网关或服务器获取时间，然后把时间发送给门锁（4.13）。



表 4-16-2 门锁获取时间回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束 标记 (1)
0xAA	0x01	0x85	0x00 00 00 01	0x01	0x00	0x2E	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xFE 表示接收数据失败。

备注：门锁定期自动向模块发此指令，从模块及网关获取最新时间；确保本地时间正确；建议每星期同步一次时间，但门锁不需要主动醒来同步时间（可以在用户开锁时，判断上次时间同步是否超过 1 星期，如果超过 1 星期则在此次开锁成功后马上同步一次时间，同步时间成功后再发送开锁上报）

## 4.17 门锁常开（模块发送给门锁）

表 4-17-1 门锁常开数据包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束 标记 (1)
0xAA	0x0A	0x86	0x00 00 00 01	0x00	0xFE FF FF FF 00 00 00 00 00 00	0x26	0x55

命令字为 0x86，应答为 0x00，命令字自增从 0x01 开始累加；有效数据分别包括：门锁保持开启的时间（4byte）高位在后（0xFFFFFFFF 表示常开），其他字节默认为 0；有效数据：0xFFFFFFFFE 表示保持开门的时间；此条数据备解释：此条主要目的，想通过 APP 设置门锁常开，APP 发送命令给网关及模块及门锁，控制门锁处于常开多长时间；实际不会产生 0xFFFFFFFFE 这种数据，但可能是保持常开状态半天。

表 4-17-2 门锁常开回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束 标记 (1)
0xAA	0x01	0x86	0x00 00 00 01	0x01	0x00	0x2D	0x55

命令字，命令字自增与接收到的协议数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xFE 表示接收数据失败。

## 4.18 报警解除（门锁发送给模块）

表 4-18-1 报警解除数据包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束 标记 (1)
0xAA	0x01	0x29	0x00 00 00 01	0x00	0x87 00 00 00 00 00 00 00 00 00	0x0F	0x55

命令字为 0x29，应答为 0x00，命令字为 0x00000001,有效数据分别包括：解除假锁报警 0x87 (1byte, 0x84 解除防拆报警, 0x85 解除未关锁报警, 0x86 解除胁迫报警, 0x87 解除假锁报警, 0xB3 解除非法操作报警, 其他值暂时无意义), 其他字节全部默认为 0；有效数据：解除假锁报警发有效数据 (0x87 00 00 00 00 00 00 00 00 00)，解除非法操作报警发有效数据 (0xB3 00 00 00 00 00 00 00 00 00)；（备注：请模块朝网关发送数据时务必直接透传此字节；门锁暂时只需要支持 0x85 及 0x87：即解除未关锁报警及解除假锁报警）

表 4-18-2 报警解除回复包

开始 标记 (1)	有效数据 长度 (1)	命令字 (1)	命令字自增 ID (4)	应答 (1)	有效数据 (n)	异惑校 验 (1)	结束 标记 (1)
0xAA	0x01	0x29	0x00 00 00 01	0x01	0x00	0x82	0x55

命令字为 0x29，命令字自增字与接收到的数据包一致，应答为 0x01，有效数据位 0x00 表示数据接收成功，0x01~0xfe 表示接收数据失败。

## 附录 1 UTC 时间转换公式

```
#define YearLength(yr) ((uint16_t)(IsLeapYear(yr) ? 366 : 365))

#define IsLeapYear(yr) (!((yr) % 400) || (((yr) % 100) && !((yr) % 4)))

#define BEGYEAR (2000)

#define DAY (60*60*24)

static uint8 monthLength( uint8 lpyr, uint8 mon )
{
    uint8 days = 31;
```

```
if ( mon == 1 ) // feb
{
    days = ( 28 + lpyr );
}
else
{
    if ( mon > 6 ) // aug-dec
    {
        mon--;
    }

    if ( mon & 1 )
    {
        days = 30;
    }
}

return ( days );
}

typedef struct
{
    uint8 seconds; // 0-59
    uint8 minutes; // 0-59
    uint8 hour;    // 0-23
    uint8 day;     // 0-30
    uint8 month;   // 0-11
    uint16 year;   // 2000+
} UTCTimeStruct;
```

## 1.UTC 时间转换成本地时间

```
void osal_ConvertUTCTime( UTCTimeStruct *tm, UTCTime secTime )//转换公式
{
    // calculate the time less than a day - hours, minutes, seconds
    {
        uint32 day = secTime % DAY;
        tm->seconds = day % 60UL;
        tm->minutes = (day % 3600UL) / 60UL;
        tm->hour = day / 3600UL;
    }

    // Fill in the calendar - day, month, year
    {
        uint16 numDays = secTime / DAY;
        tm->year = BEGYEAR;
        while ( numDays >= YearLength( tm->year ) )
        {
            numDays -= YearLength( tm->year );
            tm->year++;
        }

        tm->month = 0;
        while ( numDays >= monthLength( IsLeapYear( tm->year ), tm->month ) )
        {
            numDays -= monthLength( IsLeapYear( tm->year ), tm->month );
            tm->month++;
        }
    }
}
```

```
tm->day = numDays;
```

```
}
```

```
}
```

## 2.本地时间转换成 UTC 时间

```
UTCTime osal_ConvertUTCsecs( UTCTimeStruct *tm )
```

```
{
```

```
uint32 seconds;
```

```
/* Seconds for the partial day */
```

```
seconds = (((tm->hour * 60UL) + tm->minutes) * 60UL) + tm->seconds;
```

```
/* Account for previous complete days */
```

```
{
```

```
/* Start with complete days in current month */
```

```
uint16 days = tm->day;
```

```
/* Next, complete months in current year */
```

```
{
```

```
int8 month = tm->month;
```

```
while ( --month >= 0 )
```

```
{
```

```
days += monthLength( IsLeapYear( tm->year ), month );
```

```
}
```

```
}
```

```
/* Next, complete years before current year */
```

```
{
```

```
uint16 year = tm->year;
```

```
while ( --year >= BEGYEAR )
```

```
{  
    days += YearLength( year );  
}  
}  
  
/* Add total seconds before partial day */  
seconds += (days * DAY);  
}  
  
return ( seconds );  
}
```

## 附录 2 远程开锁密码加密解密方法

门锁将收到的 6 字节密码数据与 key[6] = {0x46, 0x45, 0x49, 0x42, 0x49, 0x47,}进行异或运算, 所得结果即为真正的密码。加密程序是相同的过程, 也是同这 6 个字节做异或运算。APP 往门锁发送数据时的加密函数方法如下:

```
Byte key[] = {0x46, 0x45, 0x49, 0x42, 0x49, 0x47}; //加密密钥  
Byte npass[6];  
  
for (int i = 0; i < 6; i ++)  
{  
    npass[i] = (Byte) (key[i] ^ password[i]);  
}
```

-----The End-----