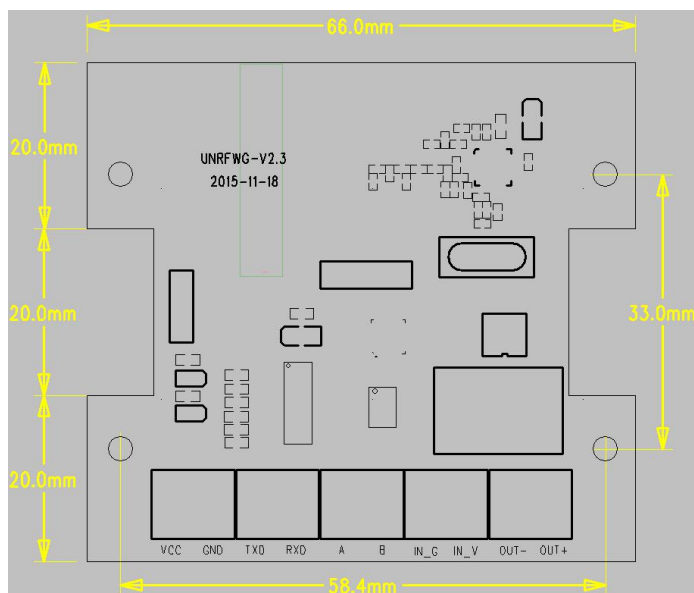


86 盒网关与智能家居协议

一、概述：

1. 86 盒网关的体质大小：66mm X 60mm，如下图：



2. RS232 或 RS485 总线，波特率 9600，一个字节十个位，一位开始位，八位数据位，一位停止位，无校验位。
3. 所有的命令都包含门锁的序列号，门锁只有收到和它的序列号相同的命令才会做出响应。
4. 无线发送接收最大通讯距离隔一个墙后一般可达 10 米。

二、接线图：

VCC	GND	TXD	RXD	A	B	IN_G	IN_V	OUT-	OUT+
接入 DC12V	地线	RS232 输出	RS232 输入	RS485 A 口	RS485 B 口	接地开门	接电源开门	继电器常开输出 1	继电器常开输出 2

- 注释：
- 1、**VCC**：电源接口，可接入直流 5V 至 12V 电压。
 - 2、**GND**：地线，需要地线的其它组合接口，都从此接入或接出。
 - 3、**TXD**、**RXD**：RS232 接通讯接口，应用时，把前面的 **GND** 一起接入使用。
 - 4、**A**、**B**：RS485 通讯接线口。
 - 5、**IN_G**：接入地线信号时开锁。可与前面的 **GND** 一起组合应用于干触点开锁，即继电器或开关的方式，要求短接时间 300ms 以上。与可视对讲结合应用的比较多。
 - 6、**IN_V**：接入电压信号时开锁。电压范围 5V 至 15V，要求信号时间 300ms 以上，接入电压信号线的时候，地线接前面的 **GND**。此接口也可做干触点接线：从前面的 **VCC** 接出电源到开关的一端，另一端接到此接口即可。此接口接线方式稳定性高。
 - 7、**OUT-**、**OUT+**：继电器干触点信号直接输出。开锁时，继电器的干触点短接，此两个端子形成回路。干触点短接时间 300ms 至 500ms。

三、命令协议对接：

1. 门锁对码：从 86 盒网关到家居系统

包头	包长度	命令码	命令内容	CRC 检验	包尾
1 字节	1 字节	1 字节	4 字节	1 字节	1 字节
0XAA	0X07	0XC0	门锁的序列号	8 位 CRC	0XBB

说明：接好线后通电，开启指纹锁按“**”再按主控指纹进入管理模式，按“6”再按“1#”进入添加服务器状态，然后输入 86 盒网关 ID 号（10 位数）以#号键结束，此时将报 ID 号语音，确认无误后按“1#”开始对码，86 盒网关会发送一个门锁对码命令，里面包含着新的门锁的序列号。

例：AA07C010425038EABB

2. 清除对码：从 86 盒网关到家居系统

包头	包长度	命令码	命令内容	CRC 检验	包尾
1 字节	1 字节	1 字节	4 字节	1 字节	1 字节
0XAA	0X07	0XC1	门锁的序列号	8 位 CRC	0XBB

说明：开启指纹锁后，进入管理模式，按“5”再按“3”进入无线设备删除状态，按“0#”即可删除对码 ID 号，同时 86 盒网关会发一串命令数据。

例：AA07C1FFFFFFFE7BB

3. 开锁：从家居系统到 86 盒网关

包头	包长度	命令码	命令内容	CRC 检验	包尾
1 字节	1 字节	1 字节	4 字节	1 字节	1 字节
0XAA	0X07	0X24	门锁的序列号	8 位 CRC	0XBB

例：AA0724104250382FBB

4. 挟持开锁：从 86 盒网关到家居系统。当前只有指纹编号为 00 的有此功能。

包头	包长度	命令码	命令内容	CRC 检验	包尾
1 字节	1 字节	1 字节	4 字节	1 字节	1 字节
0XAA	0X07	0X32	门锁的序列号	8 位 CRC	0XBB

例：AA073210425038CFBB

5. 撬门报警：从 86 盒网关到家居系统

包头	包长度	命令码	命令内容	CRC 检验	包尾
1 字节	1 字节	1 字节	4 字节	1 字节	1 字节
0XAA	0X07	0X33	门锁的序列号	8 位 CRC	0XBB

例：AA07331042503802BB

6. 开门密码开锁错误 5 次：从 86 盒网关到家居系统。当编号为 90 到 99 的钥匙开锁后，门锁会发出信息，用于监控钥匙使用者的进出情况，如小孩的放学或老人回家等。

包头	包长度	命令码	命令内容	CRC 检验	包尾
1 字节	1 字节	1 字节	4 字节	1 字节	1 字节
0XAA	0X07	0X36	门锁的序列号	8 位 CRC	0XBB

例：AA0736104250386BBB

四、CRC 算法:

1) 从包长度开始算, 到命令内容 (包括命令内容)。

2) 具体算法的 C 语言描述如下: PRT 为数组的首地址, N 为数组的长度。

```
unsigned char crc_8n(unsigned char *ptr,unsigned char n)
```

```
{  
    unsigned char crc = 0;          //CRC preset  
    unsigned char i, j;  
    const unsigned char poly = 0x8C; //CRC polynom  
  
    for (i=0;i<n;i++)  
    {  
        crc ^= ptr[i];  
        for (j=0; j<8; j++)  
        {  
            if (crc&1)  
            {  
                crc = (crc>>1) ^ poly;  
            }  
            else  
            {  
                crc = (crc>>1);  
            }  
        }  
    }  
    return crc;  
}
```