

USR-IOT 控制器指令协议

古欣、李慧斌
2014-08-14 V1.8.9

USR-IOT 控制器指令协议

目录

USR-IOT 控制器指令协议	1
目录	2
修订表	3
总则:	4
一、输出指令	6
二、输入 IO 指令	8
三、PWM 口和频率指令	9
四、红外操作指令	11
五、频率操作指令	12
六、RF 无线操作指令	12
七、寄存器指令	13
八、定时工作指令	14
九、自动控制规则	16
十、资源命名指令	17
十一、特殊指令	19

USR-IOT 控制器指令协议

修订表

编号	生成版本	修订人	修订章节与内容	修订日期
1	V1.0	古欣	生成第一个版本	2012-09-23
2	V1.2	李慧斌	增加 IO 类型与默认输出设置指令。	2013-03-03
3	V1.3	古欣	增加 PWM 和频率指令	2013-03-11
4	V1.5	王伟	增加输入 IO 状态读取 增加读取板子资源和状态指令	2013-04-23
5	V1.6	古欣	增加资源命名和寄存器操作指令	2013-06-05
6	V1.6.1	古欣	资源命名 12 字节前面增加 2 个字节，1 个用于表示设备类型，1 个备用。	2013-06-06
7	V1.6.2	古欣	增加命令 0X70 读取版本信息	2013-06-08
8	V1.7	古欣	增加定时任务相关指令	2013-06-25
9	V1.7.1	古欣	定时任务命令 bug 修正 0x52	2013-06-27
10	V1.7.2	李慧斌	增加修改设备名称指令 0x73 等	2013-07-03
11	V1.7.3	李慧斌	增加 0x75 0x76	2013-07-17
12	V1.7.4	李慧斌	0x10 11 12 13 变灰	
13	V1.7.5	王伟	0x73 修改密码返回数据由原来返回旧密码新密码改为返回成功失败	2013-07-19
14	V1.7.6	王伟	0x51 更改命令改为返回数据带有发送信息	2013-07-26
15	V1.7.7	李慧斌	0x0b 多路输出状态设置	2013-07-26
16	V1.8	古欣	增加自动控制规则	2013-08-07
17	V1.8.1	古欣	通讯协议 长度改为两位	2013-08-15
18	V1.8.2	古欣	将搜索协议添加到最后	2013-08-24
19	V1.8.3	陈超	增加寄存器清零指令	2013-10-09
20	V1.8.4	李慧斌	增加红外操作指令 0x28\29\2a\2b\2c	2013-11-08
21	V1.8.5	古欣	红外学习指令增加单路双路三路 0x29	2013-11-25
22	V1.8.5	邵慧	增加 RF 无线遥控操作指令	2014-01-17
23	V1.8.7	刘宝泉	增加 USR-WL1 的设备类型	2014-04-29
24	V1.8.8	邵慧	增加报警指令 0x15/0x95，增加 WM1/WP1 设备类型	2014-06-14
24	V1.8.9	刘宝泉	增加 USR-WL1 的控制/查询指令 0x26/0x27	2014-08-13

USR-IOT 控制器指令协议

总则:

发送命令: 包头 长度 ID 命令 参数 校验, ID 通常用于 RS485

包头(2)	长度(2)	ID(1)	命令(1)	参数(n)	校验(1)
0x55 0xaa	n+2, 长度 不包含自身 高字节在前	id	C	xxxxxx	长度(包含) 开始到参数 结束, 累加 和校验
校验包含区域					

回应命令: 包头 长度 ID 命令 参数 校验, ID 通常用于 RS485, 网络状态下只用命名+参数

包头(2)	长度(2)	ID(1)	命令(1)	参数(n)	校验(1)
0xaa 0x55 注意回应包 与发送包不 同	n+2, 长度 不包含自身 高字节在前	id	C+0x80	xxxxxx	长度(包含) 开始到参数 结束, 累加 和校验
校验包含区域					

一、对于使用网络通讯或者串口通讯非常稳定的时候, 可以只用【命令 参数】以简化通讯, 但是会存在不知道参数长度的情况, 容易出现误判。

二、回应的数据, 基本做到可以通过回应的内容判断出发送的命令的情况, 并且可以根据回应的内容正确的更新控制界面的显示, 而不需要记录自己发送的什么控制命令, 这也意味着模块可以主动回应数据。

三、假定, 所有的 IO 口都是从第 1 路开始到第 N 路, 共计 N 路, 如果出现跳跃情况, 空白区域填 0。

四、协议的内容不需要所有的产品都支持所有协议, 不同的产品实现不同的协议即可。

五、通用指令: 0xff + cmd。如果设备不支持当前指令, 将返回: 0xff + cmd 告知对方不支持本指令。

关于连续的数值表示 IO 状态的说明, 以 12 个继电器为例:

USR-IOT 控制器指令协议

字节数据	Byte1								Byte2			
路数	1	2	3	4	5	6	7	8	9	10	11	12
数值	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit0	Bit1	Bit2	Bit3
举例	1	0	0	0	1	1	1	1	1	1	0	0
	0xF1								0x03,空白处填 0			

表格一：连续 IO 表示说明（后续关于 BBBB 的描述请参考此表格）

向硬件设备发送命令有两种方式，

- 1.作为 TCP Client 直接去连接设备的 TCP Server，8899 端口，硬件的 IP 地址通过搜索协议搜索。
- 2.向云服务器发送指令，需要相应的用户名，密码和 MAC 地址，详见相关文档。

注意：本地去控制设备时，建立 TCP 连接后需要先发送密码+0x0D+0x0A，系统回应 OK 或者 NO 表示密码正确或者错误，密码正确才能继续操作。

下面只针对【命令，参数加以说明】

单片机收到命令数据时，如果系统忙回复 0x7F 0x7F，其他原因处理失败回复 0x00 0x00。

命令中 N 表示一个字节，通常选择通道，D 表示一个字节，表示值，BBBB 表示可变长度的相连的多个 Bits，DDDD 表示可变长度的相连的多个字节，H 表示高位，L 表示低位。

一、输出指令

1、0x01 N 清零(关)单一 IO

回应: 0x81 N 0

举例: 发送 0x01 0x01 回应 0x81 0x01 0x00 表示将第一路 IO 清零

N 的范围 1~255

2、0x02 N 置一(开)单一 IO

回应: 0x82 N 1

3、0x03 N 取反单一 IO

回应: 0x83 N 1/0

4、0x04 无参数 清零所有输出型 IO

回应: 0x84 0

5、0x05 无参数 置一所有输出型 IO

回应: 0x85 1

6、0x06 无参数 取反所有输出型 IO

回应: 0x86 BBBB, BBBB 表示出所有 IO 的当前状态。

BBBB 参考, 表格一: 连续 IO 表示说明

7、0x07 BBBB 选择多路清零

回应: 0x87 BBBB

BBBB 表示被选择要清零操作的继电器

8、0x08 BBBB 选择多路置一

回应: 0x88 BBBB

USR-IOT 控制器指令协议

BBBB 表示被选择要置一操作的继电器

9、0x09 BBBB 选择多路取反

回应: 0x89 BBBB

BBBB 表示出所有 IO 的当前状态

10、0x0a 无参数 读取所有输出口的 IO 状态, 无执行动作

回应 0x8a BBBB

BBBB 表示所有输出型 IO 的当前状态

11、0x0b BBBB 设置所有输出 IO 的状态

回应 0x8b BBBB

BBBB 表示命令执行后所有输出 IO 的状态

二、输入 IO 指令

IO 类型与默认输出设置指令：

1、0x10 BBBB 设置多路 IO 为输入类型

回应：0x90 BBBB 当前 io 状态

同时保存 IO 类型，立即生效

2、0x11 BBBB 设置多路 IO 为输出类型

回应：0x91 BBBB 当前 io 状态

同时保存 IO 类型，立即生效

3、0x12 BBBB 设置多路输出 IO 的默认值

回应：0x92 BBBB 当前 io 状态

保存 IO 默认值，重新上电生效

4、0x14 无参数 读取所有输入口的 IO 状态，无执行动作

回应 0x94 BBBB

BBBB 表示所有输入型 IO 的当前状态

[注：]输入与输出 IO 通用指令

5、0x13 N 读取 IO 口的当前状态

回应 0x93 N 1/0

特别的：设备可以主动发送本指令以通知应用程序，当前的状态发生了变化，此变化有可能是外部输入产生，也可能是程序自动控制逻辑产生。

5、0x15 无参数 读取所有输入口的 IO 报警状态，无执行动作

回应 0x95 BBBB

BBBB 表示所有输入型 IO 的当前状态

【注：】0x95 命令用于设备在检测到报警信息后，主动上报报警状态

三、PWM 口和频率指令

注：PWM 输出无单位，可以是 0~100 用百分比，也可以是 0~255 用 RGB 表示三原色。

1、0x20 读取所有 PWM 占空比状态

回应： 0xa0 DDDD 当前的 PWM 值，每个字节表示一路 PWM 的占空比
如回应 0xA0 0x01 0x30 表示第一路占空比为 1，第二路占空比为 48。

2、0x21 DDDD 设置所有 PWM 值

回应： 0xa1 DDDD 当前的 PWM 值，每个字节表示一路 PWM 的占空比
如发送 0x21 0x01 0x30 表示设置第一路占空比为 1，设置第二路占空比为 48。

3、0x22 N D 设置指定通道的占空比输出

N 表示操作的通道，D 为实际数值

回应 0xa2 N D，比如发送 0x22 0x01 0x10 表示设置第一路占空比为 16。

4、0x23 N 读取指定通道的占空比输出

N 表示操作的通道

回应 0xa3 N D，比如发送 0x23 0x01，回应 0xa3 0x01 0x10，表示读取到第一路占空比为 16

5、0x24 读取所有 PWM 的占空比与频率状态

回应 0xa4 DD DH DL ...

每路 PWM 有 1 个字节占空比，2 个字节的频率，多路的话依次往后排。

6、0x25 DD DH DL...设置所有 PWM 的占空比与频率

回应 0xa5 DD DH DL...

每路 PWM 有 1 个字节占空比，2 个字节的频率，多路的话依次往后排。

7、0x26 S M DDDD DDDD DDDD DDDD 设置模式指令

S 表示速度(0~100),速度设置为 0 时，表示在静态模式，若模式为跳变或是渐变，此处 S 不能为 0。M 表示模式类型，0 表示静态，1 表示跳变，2 表示渐变。

DDDD 表示一组颜色值，每个字节分别表示 R、G、B、W。

例如：发送 0x26 0x05 0x01 0x03 0x00 0x05 0x03 0x00 0x05 0x03 0x00 0x05 0x02 0x03 0x05

成功返回：0x26 1

失败返回：0x26 0

8、0x27 读取当前状态

返回： 0x27 S M DDDD DDDD DDDD DDDD 占位字节含义参考 0x26 指令。

USR-IOT 控制器指令协议

举例：0x27 0x00 0x00 0x02 0x02 0x03 0x01 表示静态模式，颜色参数为 0x02 0x02 0x03 0x01。

注：第七条和第八条仅适用于 USR-WL1 产品。

四、红外操作指令

1、0x28 创建新的一路红外数据

回应 $0xA8 + N$

N 为创建了第几路红外数据，范围 1 ~ 127，0xff 为创建失败

2、0x29 + N+X 学习第 N 路红外数据，X 范围 1~3，代表单键，双键，三键学习

回应 $0xA9 + N + M$

M 为学习状态：0 为收到命令，准备接收第 N 路红外数据；1 为接收到红外数据，学习第 N 路成功；2 为学习个数超过 127；3 为学习超时，自动退出学习

注：回应时应依次返回 正在学习 + 学习结果

学习命令区分单键/双键/三键，但是返回命令无此标记，都当成学习处理。

3、0x2A + N 发送第 N 路红外数据

回应 $0xAA + N + M$

M 为发送状态，0 为正在发射红外数据；1 为发送完成；2 为该路为学习过红外数据；3 为发射个数超过 127

4、0x2B + N 清除第 N 路红外数据

回应 $0xAB + N + M$

M 为清除结果，0 为收到命令，删除进行中；1 为删除完成；2 为删除个数超过 127

5、0x2C 清除所有的红外数据

回应 $0xAC + M$

M 为清除状态，0 为收到命令，正在清除中；1 为清除完成

注意：红外指令，会出现一发两回，或者一发多回的情况。

五、频率操作指令

频率操作命令和 PWM 操作命令相通，但是频率的参数是两个字节，高位在前，低位在后。

1、0x30 读取所有频率状态

回应： 0xb0 DHDL DHDL 当前的频率值，每两个字节表示一路频率的值

如回应 0xB0 0x00 0x30 0x10 0x00 表示第一路频率为 $0*256+48=48$ ，第二路频率为 $16*256+0=256$ 。

2、0x31 DHDL 设置所有频率值

回应： 0xb1 DHDL DHDL 当前的频率值，每两个字节表示一路频率值

如发送 0x31 0x01 0x30 0x00 0x20 表示设置第一路频率为 $1*256+48=304$ ，设置第二路频率为 $0+32=32$ 。

3、0x32 N D 设置指定通道的频率值

N 表示操作的通道，DHDL 为实际数值

回应 0xb2 N DH DL，比如发送 0x22 0x01 0x10 0x02 表示设置第一路占空比为 $16*256+2=4098$ 。

4、0x33 N 读取指定通道的频率值

N 表示操作的通道

回应 0xb3 N DH DL，比如发送 0x23 0x01，回应 0xb3 0x01 0x10 0x02，表示读取到第一路占空比为 $16*256+2=4098$

六、RF 无线操作指令

1、0x3A 创建一路无线遥控码资源

回应： 0xBA N

N 为创建了第几路无线遥控码，范围 1 ~ 127，0xff 为创建失败

2、0x3B N 删除第 N 路无线遥控码资源

回应： 0xBB N F

N 为第几路无线遥控资源，F 为删除结果标识，1 为成功，0xFF 为失败

3、0x3C N 发送第 N 路无线遥控码

回应： 0xBC N F

N 为第几路无线遥控码；F 为状态标识，0 为无第 N 路无线遥控码，1 为正在发送中，2 为发送成功，0xFF 则为发送失败

命令 0x3D、0x3E、0x3F 预留

七、寄存器指令

每个寄存器数据为两个字节，用来表示 AD 模拟量输入及所有传感器数据，最高位表示正负，=1 表示负数；=0 表示正数，数据部分除以 10 表示，范围从-3276.7 到+3276.7，举例：

比如收到 0x80 0x10 表示： -1.6

比如收到 0x01 0xaa 表示： +42.6

1、0x40 读取所有寄存器数据

返回： 0xC0 DDDD ... 依次返回所有寄存器数据

2、0x41 N 读取单一寄存器数据

返回： 0xC1 N DH DL 单个寄存器数据

3、0x42 S N 读取指定寄存器段数据

返回： 0xC2 S N DH DL DH DL ... 指定段的寄存器数据

描述： S 一个字节，表示寄存器起始地址（0-255）

N 一个字节，表示寄存器数量（1-255）

4、0x43 N 清零单一寄存器数据

返回： 0xC3 N 清零的寄存器路数

5、0x44 清零所有寄存器数据

返回 0xC4 0

USR-IOT 控制器指令协议

八、定时工作指令

系统自定义一个 55 字节的存储区间，用于存储 0~4 的 5 条定时命令，位置个序列固定。【暂定 5 条，视系统处理能力，也可以多】

注意使用本功能需要在 0x70 命令回应中标识是否支持本功能。

类目	任务序号 id	类型 使能 type	时间 time	执行命令 cmd	周使能 week
字节数	1	1	4	4	1
说明	从 1 开始，1~5 最多 5 个	最高位表示使能与否，1 为使能，添加时默认使能。低位表示循环类型，见备注	Unix 时间戳，高位在前	执行本协议表中的命令，通常指输出控制指令，不足的字节补 0	Bit 0~6 分别代表星期天到星期六，1 为允许动作，0 为不允许，最高位请写 0。
举例	0x01	0x80	0x51C8E925	0x05 00 00 00	0x7F
含义	定时任务一	使能 单次任务	2013-06-25-08:49:41	输出全开	每天均允许动作

备注：

循环类型说明：0 单次，1 分循环，2 时循环，3 天循环，4 月循环。系统在执行完定时任务后，根据循环类型自动将定时时间调整为下一个需要执行的时间。民用 APP 只考虑单次和天循环{通过周使能功能来选择哪几天运行}即可。

Unix 时间戳：是从 1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的秒数，不考虑闰秒。如：0x51C8E925 = 1372121381 = 2013 年 06 月 25 日 08:49:41

请参考在线转换工具：<http://shijianchuo.911cha.com/>

1、0x50 N 读取定时任务列表

参数：N：用于指示读取第 N 路 IO 的定时任务，N 为 0 时表示读取所有定时任务，N 不为 0 时表示读取用于第 N 路 IO 的定时任务列表。N 取值 0~255；

返回：0xD0 + 符合条件任务数量{小于等于 5 个，0 则是无列表}+任务列表，依次排列，格式如上面表格。

2、0x51 增加定时任务

参数：Type Time CMD Week 如上表中所述

返回：0xD1 ID Type Time CMD Wee，ID 表示存储后所在的位置序号，FF 表示存储已满，存储失败。

USR-IOT 控制器指令协议

3、0x52 M N 单个定时任务启用 停用 删除

参数: M 表示要操作的任务号 1~5

N 表示操作类型 1: 启用 2: 停用 3: 删除

返回: 0xD2 M N

4、0x53 读取系统时间

参数: 无参数

返回: 0xD3 Time Time 为 Unix 时间戳

5、0x54 设置系统时间

参数: Time

返回: 0xD4 Result Time

Result 1 为成功, 0 为失败。{这里有可能因为硬件原因, 不支持或者设定时间失败。} Time 为时间戳。

USR-IOT 控制器指令协议

九、自动控制规则

类目	规则存储 ID	使能	资源类型	资源位置	比较符 ASCII 字符	比较值	执行命令
字节数	1	1	1	1	1	2	4
说明	从 1 开始, 1~5 最多 5 个	使用最低位, 1 为使能, 0 为禁用, 其余位备用。	0-3, 0 表示输出; 1 表示输入依次类推	即第几路资源, 1~255	运算符 < > = ! 分别表示小于 大于 等于 不等于 专用于: IO 输入与输出跟随 ~ 取反 & 同相跟随	0~65535, IO 值和寄存器值通用	执行本协议表中的命令, 通常指输出控制指令, 不足的字节补 0
举例							

建议下位机程序处理时, 做 10ms 的状态机确认机制, 确认是**第一次准确**的变化后执行动作。但是当改变不是由内部检测产生而是由命令控制时, 应该先执行自动控制规则再执行命令, 特别用于输出互锁指令, 比如防止两个方向的继电器同时工作损坏电机。

1、0x5F N 读取自动控制规则列表

参数: N: 读取指定存储位置的自动控制规则, 即下一条命令返回的 ID, N 为 0 是读取所有。

返回: 0xDF +N+符合条件任务数量{小于等于 5 个, 0 则是无列表}+任务列表, 依次排列, 格式如上面表格。

2、0x5E 增加自动控制规则

参数: Enable Type Position Compare value command 如上表中所述

返回: 0xDE ID Enable Type Position Compare value command, ID 表示存储后所在的位置序号, FF 表示存储已满, 存储失败。

3、0x5D M N 单个自动控制规则启用 停用 删除

参数: M 表示要操作的任务号 1~5

N 表示操作类型 1: 启用 2: 停用 3: 删除

返回: 0xDD M N

十、资源命名指令

为方便用户使用设备，每个设备给每个资源可以独立命名，设备名称存储到硬件设备里供用户 APP 读取，每个通道的命名空间为 14 个字节，第一个字节用于标识该通道的实际属性(见附表)，第二个字节备用，后面 12 个字节表示名称，采用 Unicode-8 编码存储，字母最多 12 个字节，汉字最多 4 个，如果名称不足 12 字节，则以 0 视为结束，如遇全零或者全 F，APP 上应显示为未命名，硬件上可以根据自身情况主动给资源一个默认名称。

实际操作时，第 1 和第 2 字节，可以暂时留空，写 0。

1、0x60 N M DDDDDDDDDDDDDDD 配置单个资源的单路名称

N 表示资源类型 (0-3， 0 表示输出；1 表示输入依次类推，及 7E 命令表示的 XYZK)，M 表示第几路 (1-255)；D 等 14 字节数据为本路名称。

返回： 0xE0 N M DDDDDDDDDDDDDDD

2、0x61 N M 读取单路资源的单路名称

返回： 0xE1 N M DDDDDDDDDDDDDDD

解释： N 表示资源序号 (0-3)，M 表示路序号 (1-255) 单路的名称

3、0x62 DDDDDDDDDDDDDDD ... 配置所有资源的所有路名称。

返回： 0xE2 DDDDDDDDDDDDDDD ... 依次往后排列。

4、0x63 读取所有资源的所有路名称

返回： 0xE3 DDDDDDDDDDDDDDD ... 所有资源的所有路名称，依次往后排列。

5、0x64 N DDDD... 配置某一类资源的所有名称

N 表示资源序号 (0-3)

返回 0xE4 N DDDD...

6、0x 65 N 读取某一类资源的所有名称

返回 0xE5 N DDDD... 所有资源名称，依次往后排列。

USR-IOT 控制器指令协议

资源名称第一个字节的值表示的内容如下表，APP 根据这个值显示对应的图标。

0	1	2	3	4	5	6	7	8	9
预留，默认。	灯泡	插座	开关	电视	空调	冰箱	热水器	温度	湿度

10	11	12							
CO 报警器	颗粒物								

资源名称另一个字节的表示内容（比特位）如下：

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
1: 曾经成功学习红外; 0 从未成功学习过红外							

十一、特殊指令

1、0x7E 无参数 读取板子的资源情况

返回 0xFE x y z k m

X Y Z K M 分别表示输出口 输入口 PWM 口 寄存器 红外 的数量

添加：0x72 x y z k m 配置板子的资源情况

返回 0xF2 x y z k m

以下 2.3.4 指令 时序电源专用，通用产品不用考虑。

2、0x7D 无参数 读取当前电压

返回 0xFD H L

H 和 L 表示电压的高位和低位

3、0x7C 无参数 执行 Normal 开命令

收到命令后立即返回 0xFC 0x00 表示开始执行命令。

执行结束后返回 0xFC 0x01 表示执行结束。

4、0x7B 无参数 执行 Normal 关命令

收到命令后立即返回 0xFB 0x00 表示开始执行命令。

执行结束后返回 0xFB 0x01 表示执行结束。

0x77 设备推送报警内容

返回 0xF7 A DDDDDDDDD

设备在需要报警推送时，主动发送 0xF7 DDDDDDD 到服务器，DDDDDD 为报警推送内容。A 表示内容中前 A 个字节为推送标题，最大 255，剩余字节则为推送描述；DDDDDD 为 UTF-8 编码格式，最多 255*2 个字节

备注：0x77 无意义，服务器向设备发送该命令，设备无需响应

0x78 X Y Z PWM 切换到 IO 模式

返回 0xF8

PWM 切换到 IO 模式发送指令 0x78 X Y Z，X Y Z 分别表示输出口、输入口、PWM 口，模块返回 0xF8 X Y Z 例如将 4 路 PWM 全部切换到 IO 模式就是 0x78 8 0 0。例如将 7 路 PWM 切换到 IO 模式就是 0x78 7 0 1。

0x79 A B C D 修改设备的 ID

返回 0xF9 A B C D

具体指令如下：0x79+地址高位+地址低位+地址高位反码+地址低位反码，设备回应 0xF9+地址高位+地址低位+地址高位反码+地址低位反码。

0x7A 保存当前配置

返回 0xFA

USR-IOT 控制器指令协议

发下命令后，保存所有的 IO 开关状态，所有 PWM 的频率和占空比，等重启以后按照保存的配置设置 IO 和 PWM

具体指令是 0x7A，模块返回 0xFA（就是 0x7A+0x80）。

0x70 读取设备版本信息

返回 0xF0 设备功能字(2 字节) 硬件版本(2 字节) 软件版本(2 字节)

设备功能字的第 1 个字节的 8 个 bit 功能描述如下，1 为支持，0 为不支持：

bit	0	1	2	3	4	5	6	7
功能	可网页配置	资源数量可配置	WIFI	有线网络	GPRS	Smart-link	是否支持定时任务	保留

Bit5 Smart-link 位如果为 0 则对应 A11 的协议，为 1 则对应 LPB 的协议。

第二个字节表示设备类型号，标识不同的设备。

暂定目前的设备：

IOT1	1
WIFI IO MINI	2
GPRS RTU	3
WIFI-IO-83	4
IOT2	5
USR-WL1	6
USR-WM1	7
USR-WP1	8
USR-ZKB	9（掌控宝通用设备类型）
USR-HTW	10
USR-Relay16	11
数据采集卡	12
USR-WP1q	13
智能窗帘	14
智能网关	15
USR-PDU	16

硬件版本和软件版本用两个字节表示，高字节表示大版本，低字节表示小版本，组合成 V.H.L。

0x73 DDDDDD DDDDDD 修改设备密码

参数：1 修改成功；0 旧密码错误，修改失败

返回 0xF3 1/0

USR-IOT 控制器指令协议

0x74 DDDDDDDDDDDDDDDDDDD 修改设备名称

返回 0xF4 DDDDDDDDDDDDDDDDDDD 参数为设备名称，16 字节。

0x75 获取设备名称

参数：无

返回 0xF5 DDDDDDDDDDDDDDDDDDD 参数为设备名称，名称长度为 16 字节

0x76 Port N DDDDDDDDD 向特定串口发送数据

参数：Port 串口号（0~2） N 数据部分的长度 DDDDDDDDD 要发送的串口数据

返回 0xF6

Todo: 数据回应部分指令待定

USR-IOT 控制器指令协议

附录： 搜索协议

搜索协议在 E45 基础上使用，使用 UDP 广播，端口 1901

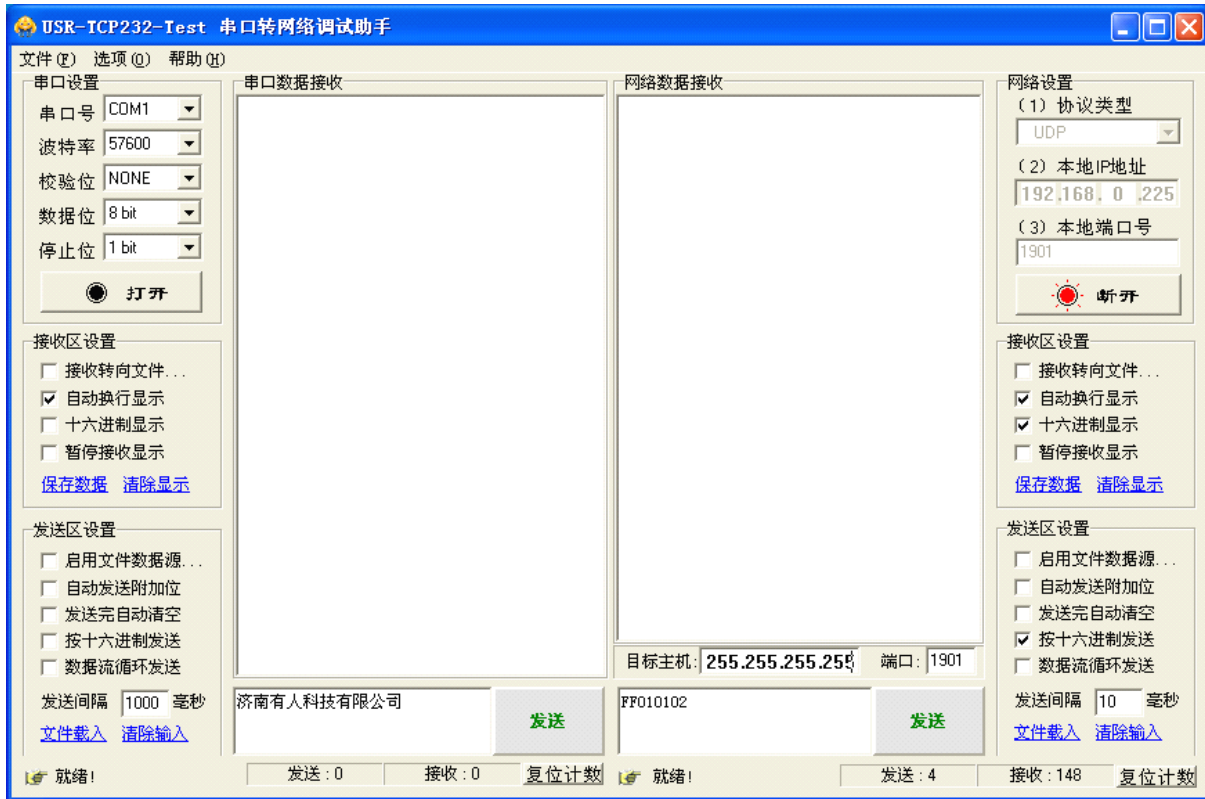
搜索发送：

ff 01 01 02

搜索返回：

字节	名称	例子	说明
0	TAG_STATUS	FF	恒为 0xff
1	Packet_length	24	包总长度（所有数据的长度，恒为 0x24）
2	CMD_DISCOVER_TARGET	01	命令字，恒为 0x01
3	Board_type	01	设备类型（1 为 IOT，2 为 WIFI io mini，3 为 GPRS rtu）
4	Board_ID	4B	设备功能字（详见 GPIO 协议，0x70 命令字）
5~8	Client_IP_address	C0 A8 00 44	设备 IP（高字节在前）
9~14	MAC_address	D8 B0 4C 00 01 64	设备 MAC（高字节在前）
15~18	Firmware_version	DA 07 01 00	DA 07：软件版本（低字节在前 2010）； 01 00：为硬件版本（低字节在前），从 1 开始计，但不能为 0x0000
19~34	Application_title	55 53 52 2D 49 4F 54 31 00 00 00 00 00 00 00 00	设备名称
35	checksum	85	校验字节。基数为 0，从字节 0（包含）开始减，减到校验字节之前，结果保留低字节。

USR-IOT 控制器指令协议



USR-IOT 控制器指令协议

简单通讯举例

以 USR-WP3 为例子，

发送命令关闭第一路 IO					
发送代码	55 AA	00 03	00	01 01	05
说明	包头	包长度 3 字节	ID, 默认零	命令和参数 命令号 01(清零命令) 参数 01	和校验 $00+03+00+01+01=05$
回应代码	AA 55	00 04	00	81 01 00	86
说明	包头	包长度 4 字节	ID, 默认零	命令和参数 命令号 01(清零命令) 第一路 01 清零	和校验

TCP 发送: 55 AA 00 03 00 02 01 06

TCP 接收: AA 55 00 04 00 82 01 01 88

发送命令打开第一路 IO					
发送代码	55 AA	00 03	00	02 01	06
说明	包头	包长度 3 字节	ID, 默认零	命令和参数 命令号 02(置一命令) 参数 01	和校验 $00+03+00+02+01=06$
回应代码	AA 55	00 04	00	82 01 01	88
说明	包头	包长度 4 字节	ID, 默认零	命令和参数 命令号 02(置以命令) 第一路 01 置一	和校验