

掌控宝硬件设备 GPIO 控制协议

目录

掌控宝硬件设备 GPIO 控制协议	1
目录	2
1 协议格式.....	3
1.1 名词定义.....	3
1.2 控制端发送设备接收协议帧格式.....	3
1.3 设备返回/设备发送控制端接收协议帧格式	4
1.4 指令分类.....	4
1.5 设备控制流程.....	5
2 IO 指令	8
2.1 断开一路输出 IO	8
2.2 闭合一路输出 IO	8
2.3 取反一路输出 IO	8
2.4 断开所有输出 IO	9
2.5 闭合所有输出 IO	9
2.6 取反所有输出 IO	9
2.7 读取所有输出 IO 状态	10
2.8 读取所有输入 IO 状态	10
3 寄存器指令.....	11
3.1 寄存器指令说明.....	11
3.2 读取所有寄存器数据.....	11
3.3 读取单一寄存器数据.....	11
3.4 读取指定段寄存器数据.....	12
4 定时工作指令.....	13
4.1 读取定时任务列表.....	14
4.2 增加定时任务.....	14
4.3 启用/停用/删除单个定时任务.....	14
4.4 读取设备时间.....	15
4.5 设置系统时间.....	15
5 资源命名指令.....	16
5.1 设置单个资源名称.....	16
5.2 读取单路资源名称.....	17
5.3 读取所有资源名称.....	17
6 特殊指令.....	18
6.1 读取设备资源配置.....	18
6.2 保存当前状态.....	18
6.3 ID 设置指令	19
6.4 延时设置指令.....	19
7 修订表:	20
8 免责声明.....	20

1 协议格式

1.1 名词定义

控制端：包括手机 APP、PC 端控制软件、WEB 端等发起控制的终端；

设备端：掌控宝产品系列下所有的硬件产品；

0xXX：0x 表示十六进制数字格式，XX 表示数字内容。本文档中，0x 可能会省略。

1.2 控制端发送设备接收协议帧格式

帧定义

包头 (2)	长度 (2)	ID (1)	命令 (1)	参数 (n)	校验 (1)
0x55 0xaa	N+2	ID	CMD	XX XX ...	累加和
		长度值所包含的区域			
	校验所包含的区域				

字段解析

包头：2 个字节，固定为 0x55 0xaa；

长度：如 1.2.1 表格所标记区域，长度的数值为长度值包含区域的字节数；

ID：设备 ID，暂未启用，默认值为 0；

命令：控制设备的命令字，此字节的值指明了设备具体执行的动作；

参数：命令携带的参数，不定长， $n \geq 0$ ；此参数指明了设备执行命令所定义的动作所涉及的范围（包括输出/输入 IO 或寄存器的范围等）；

校验：自长度开始，到参数的最后一个字节结束，所有字节累加，保留一个字节作为校验字，超出一个字节的舍弃；

1.3 设备返回/设备发送控制端接收协议帧格式

帧定义

包头 (2)	长度 (2)	ID (1)	命令 (1)	参数 (n)	校验 (1)
0xaa 0x55	N+2	ID	CMD+0X80	XX XX ...	累加和
		长度值所包含的区域			
	校验所包含的区域				

字段解析

包头：2 个字节，固定为 0x55 0xaa；

长度：如 1.2.1 表格所标记区域，长度的数值为长度值包含区域的字节数；

ID：设备 ID，暂未启用，默认值为 0；

命令：设备端返回的命令字，控制指令字的最高位置一返回；此字节的值指明了设备执行了什么动作；

参数：命令携带的参数，不定长， $n \geq 0$ ；此参数指明了设备执行命令所定义的动作所涉及的范围（包括输出/输入 IO 或寄存器的范围等）；

校验：自长度开始，到参数的最后一个字节结束，所有字节累加，保留一个字节作为校验字，超出一个字节的舍弃；

1.4 指令分类

GPIO 协议目前主要使用的分类有：IO 指令、寄存器指令、定时工作指令、资源命名指令和一些特殊指令。

1.5 设备控制流程

1.5.1 设备搜索协议

使用 UDP 向局域网发送广播，端口为 1901，设备接收后即返回设备信息，包括 IP/MAC/名称等。

搜索指令为（十六进制）：**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。

1.5.2 设备连接和权限验证

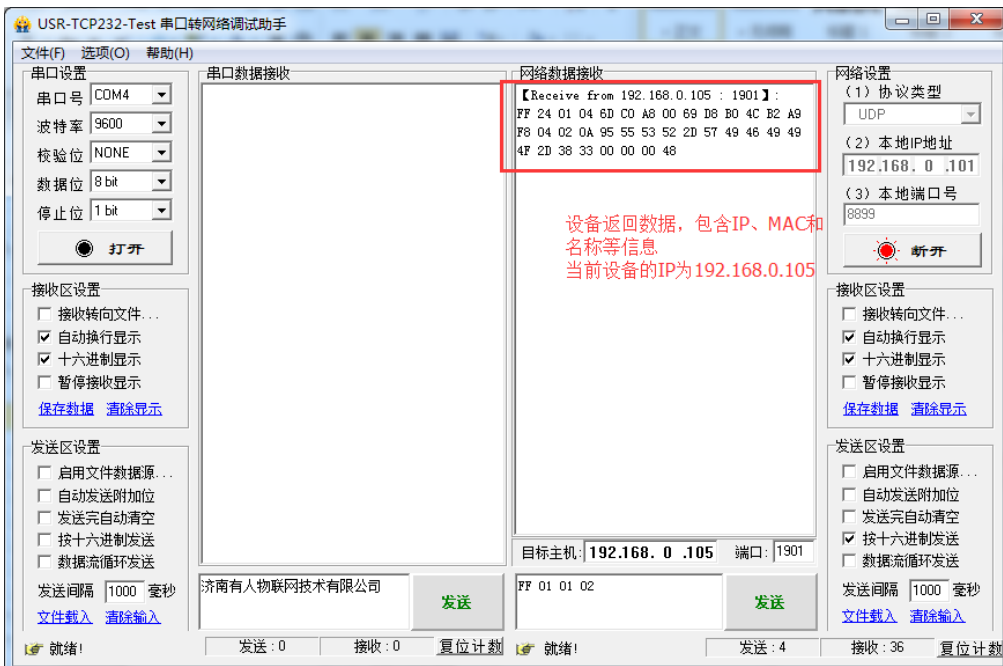
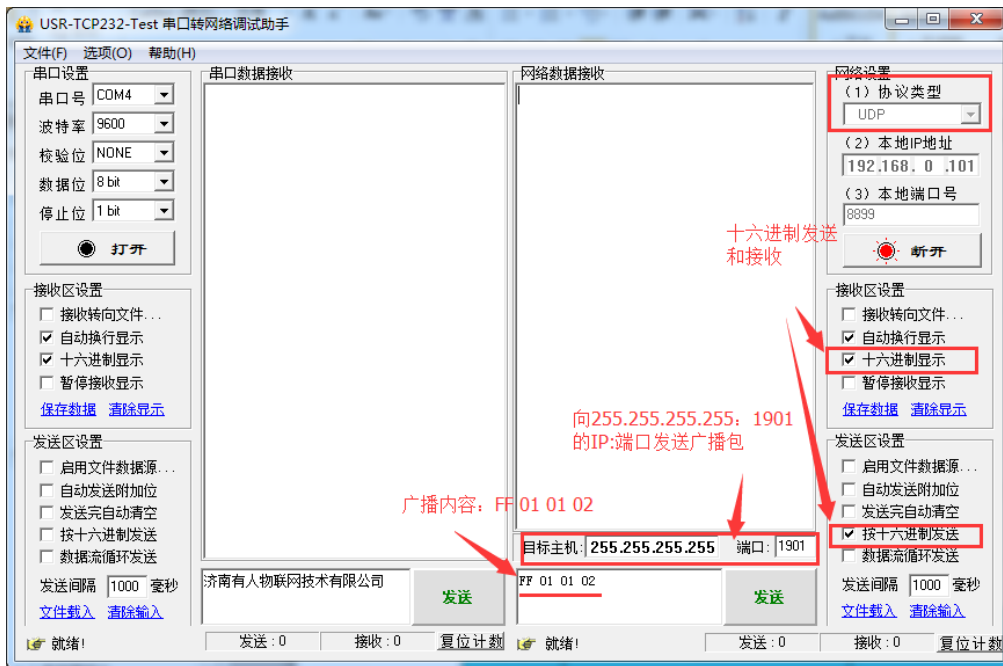
在通过 1.5.1 的搜索过程完成搜索后，可以获得设备的 MAC/IP/名称的对应关系。

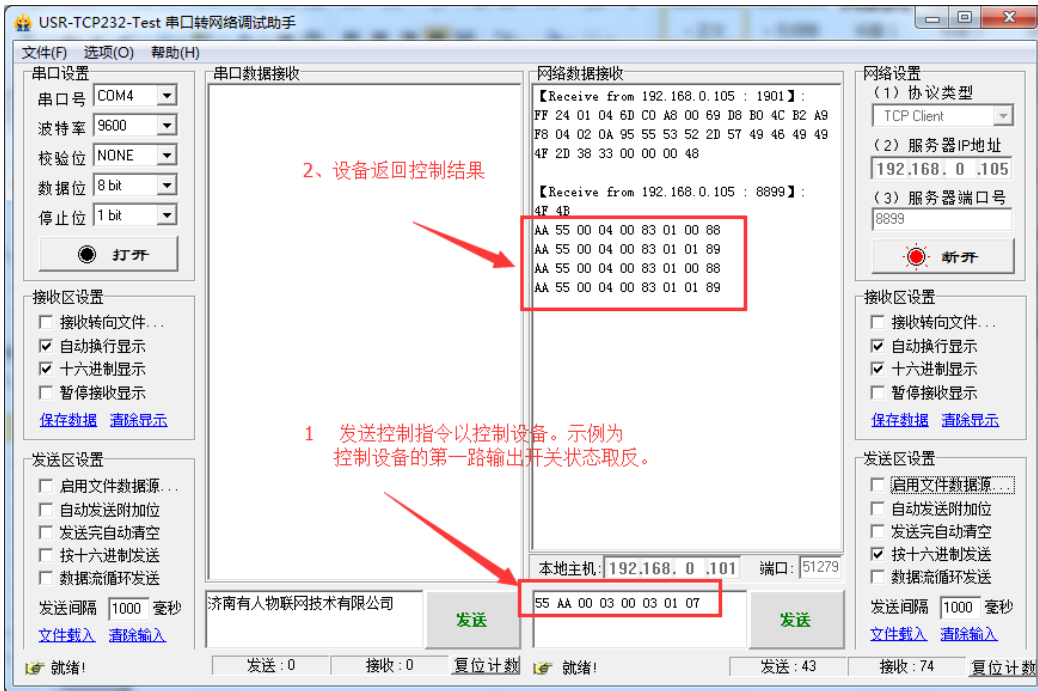
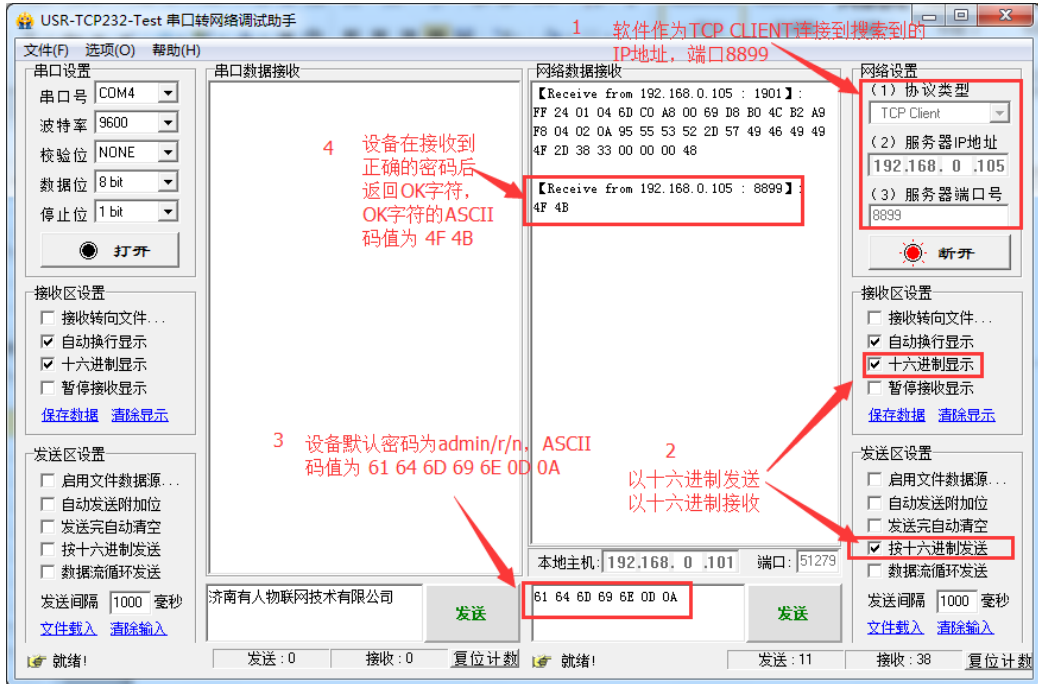
如果需要对设备进行控制，必须先发起 TCP 连接到设备并且发送设备密码。设备的控制端口为 8899，初始密码为 admin/r/n（/r/n 为回车和换行两个字符的 C 语言表述方式）。

1.5.3 设备控制流程

在经过设备搜索、TCP 连接建立后，即可发送 GPIO 控制协议帧对设备进行控制。

使用网络调试软件对设备进行搜索、连接和控制流程如下：





2 IO 指令

2.1 断开一路输出 IO

指令符: 0x01

参数: N : N 表示第几路, 例如 N=0x01, 表示第一路;

返回指令符: 0x81

返回参数: N 0x00: N 表示第几路, 例如 N=0x01, 表示第一路;

协议示例:

控制端发送 (十六进制): 55 AA 00 03 00 01 01 05

设备端返回 (十六进制): AA 55 00 04 00 81 01 00 86

2.2 闭合一路输出 IO

指令符: 0x02

参数: N : N 表示第几路, 例如 N=0x01, 表示第一路;

返回指令符: 0x82

返回参数: N 0x01: N 表示第几路, 例如 N=0x01, 表示第一路;

协议示例:

控制端发送 (十六进制): 55 AA 00 03 00 02 01 06

设备端返回 (十六进制): AA 55 00 04 00 82 01 00 87

2.3 取反一路输出 IO

指令符: 0x03

参数: N : N 表示第几路, 例如 N=0x01, 表示第一路;

返回指令符: 0x83

返回参数:

N M: N 表示第几路, 例如 N=0x01, 表示第一路; M 表示 IO 状态, 1 表示闭合状态, 0 表示断开状态;

协议示例:

控制端发送 (十六进制): 55 AA 00 03 00 03 01 07

设备端返回 (十六进制): AA 55 00 04 00 83 01 00 88

2.4 断开所有输出 IO

指令符: 0x04

参数: 无参数;

返回指令符: 0x84

返回参数: 0x00: 执行成功后返回输出 IO 的状态为 0x00;

协议示例:

控制端发送 (十六进制): 55 AA 00 02 00 04 06

设备端返回 (十六进制): AA 55 00 03 00 84 00 87

2.5 闭合所有输出 IO

指令符: 0x05

参数: 无参数;

返回指令符: 0x85

返回参数: 0x01: 执行成功后返回输出 IO 的状态为 0x01;

协议示例:

控制端发送 (十六进制): 55 AA 00 02 00 05 07

设备端返回 (十六进制): AA 55 00 03 00 85 01 89

2.6 取反所有输出 IO

指令符: 0x06

参数: 无参数;

返回指令符: 0x86

返回参数:

XX XX ...: 执行成功后返回输出 IO 的状态, 返回字节按位展开, 一个位代表一个输出 IO 的状态; 第一字节表示第 1~8 路输出 IO, 第二字节表示第 9~16 路输出 IO, 按最低位到最高位排列, 依次类推。

协议示例:

控制端发送 (十六进制): 55 AA 00 02 00 06 08

设备端返回 (十六进制): AA 55 00 04 00 86 05 05 94

其中, 0x86 后第一个 05 换算为二进制: 0b00000101, 表示第 1 路和第 3 路输出 IO 为闭合状态, 其余为断开状态; 第二个 05 换算为二进制: 0b00000101, 表示第 9 路和第 11 路输出 IO 为闭合状态, 其余为断开状态。

2.7 读取所有输出 IO 状态

指令符: 0x0A

参数: 无参数;

返回指令符: 0x8A

返回参数:

XX XX ...: 执行成功后返回输出 IO 的状态, 返回字节按位展开, 一个位代表一个输出 IO 的状态; 第一字节表示第 1~8 路输出 IO, 第二字节表示第 9~16 路输出 IO, 按最低位到最高位排列, 依次类推。

协议示例:

控制端发送 (十六进制): 55 AA 00 02 00 0A 0C

设备端返回 (十六进制): AA 55 00 04 00 8A 05 05 98

其中, 0x8A 后第一个 05 换算为二进制: 0b00000101, 表示第 1 路和第 3 路输出 IO 为闭合状态, 其余为断开状态; 第二个 05 换算为二进制: 0b00000101, 表示第 9 路和第 11 路输出 IO 为闭合状态, 其余为断开状态。

2.8 读取所有输入 IO 状态

指令符: 0x14

参数: 无参数;

返回指令符: 0x94

返回参数:

XX XX ...: 执行成功后返回输出 IO 的状态, 返回字节按位展开, 一个位代表一个输出 IO 的状态; 第一字节表示第 1~8 路输出 IO, 第二字节表示第 9~16 路输出 IO, 按最低位到最高位排列, 依次类推。

协议示例:

控制端发送 (十六进制): 55 AA 00 02 00 14 16

设备端返回 (十六进制): AA 55 00 04 00 94 05 99

其中, 0x94 后的 05 换算为二进制: 0b00000101, 表示第 1 路和第 3 路输出 IO 为闭合状态, 其余为断开状态。

3 寄存器指令

3.1 寄存器指令说明

设备端可能包含有寄存器，用于存储通过传感器采集到的数据。

每个寄存器数据为两个字节，传输时高字节在前，低字节在后，用来表示数字量的具体数值。最高位为符号位，为 1 表示为负数，为 0 表示为正数。数据部分除以 10 表示实际值。寄存器的数据范围为-3276.7~+3276.6。

示例：接收到的数据位 0x80 0x10，表示数据为-1.6（最高位为 1 表示是负数，0x10 换算为十进制为 16，除以 10 等于 1.6）。

3.2 读取所有寄存器数据

指令符：0x40

参数：无参数；

返回指令符：0xC0

返回参数：

XX XX ...：依次返回所有寄存器数据，高字节在前，低字节在后。

协议示例（以设备有三个寄存器为例）：

控制端发送（十六进制）：55 AA 00 02 00 40 42

设备端返回（十六进制）：AA 55 00 08 00 C0 00 01 00 02 00 03 CE

设备的三个寄存器的值分别为 0001，0002,0003。

3.3 读取单一寄存器数据

指令符：0x41

参数：N；

返回指令符：0xC1

返回参数：

N DH DL：返回第 N 个寄存器数据，高字节在前，低字节在后。

协议示例（以设备有三个寄存器为例）：

控制端发送（十六进制，读取第 2 个寄存器值）：55 AA 00 03 00 41 02 46

设备端返回（十六进制，返回第 2 个寄存器值）：AA 55 00 05 00 C1 02 00 02 CA

设备的第 2 个寄存器的值为 0002。

3.4 读取指定段寄存器数据

指令符: 0x42

参数: S N;

返回指令符: 0xC2

返回参数:

S N DH DL DH DL ... : 返回自 S 开始的 N 个寄存器数据, 高字节在前, 低字节在后。

协议示例 (以设备有三个寄存器为例):

控制端发送 (十六进制, 读取序号 2~3 两个寄存器值):

55 AA 00 04 00 42 02 02 4A

设备端返回 (十六进制, 返回第 2~3 两个寄存器值):

AA 55 00 05 00 C2 02 02 00 02 00 03 D0

设备的第 2 个寄存器的值为 0002, 第 3 个寄存器的值为 0003。

4 定时工作指令

对于支持定时功能的设备，存在一个存储区，用于存储定时指令。并且由设备在预定时间执行预定功能。

定时工作指令包括 读取定时任务列表、增加定时、启用/停用/删除单个定时任务、读取设备当前时间、设置设备时钟 共五种定时指令。

每个定时指令的数据区包含 11 字节的存储区，用于指明定时的序号、类型、时间、所需要执行的命令等信息。定时任务的 11 个字节的含义如下表。

需要注意的是，目前所有的掌控宝设备的定时功能，在 APP 控制上都只能用于输出开关，定时指令相关的读取等功能，也是针对某一路输出开关而设置。如果用户希望使用其他方面的定时，需要自行根据协议和指令格式来做软件实现。

类目	任务序号 ID	是否使能 使能类型 TYPE	定时时间 TIME	执行命令 CMD	周使能类型 WEEK
字节数	1	1	4	4	1
说明	从 1 开始编号，表示当前设备的第几条定时任务	Type 字节二进制表示的情况下，最高位表示此条定时任务是否使能。后七位数值表示使能类型：0 为单次执行，1 位按分钟循环，2 位按小时循环，3 位按天循环，4 位按月循环。目前 APP 只支持单次、按天循环。	Unix 时间戳，高字节在前。不区分时区。无论在哪个时区，均以当前时区为 UTC 协调时区。	定时时间到达后所执行的命令。命令为本文档所描述的所有有效指令，命令内容只包括“命令+参数”两部分内容。长度不足的在后续字节填充 0	Bit0~6 位分别代表星期天到星期六，为 1 表示允许执行，为 0 表示不允许执行。最高位为 0。
例子	0x01	0x83	0x51C8E925	0x05 0x00 0x00 0x00	0x7F
例子描述	第一个定时任务	定时任务使能（有效），按天执行。	2013-06-25-08:49:41	闭合全部输出开关	一周的七天均执行此操作

4.1 读取定时任务列表

指令符: 0x50

参数: N 用于指示读取第几个输出开关的定时任务。

返回指令符: 0XD0

返回参数:

M List: M 表示符合条件的任务数量, List 表示定时任务列表。如果 M=0, 则无任务列表。

协议示例 (以设备的第一路开关有一个定时任务为例):

控制端发送 (十六进制): 55 AA 00 03 00 50 01 54

设备端返回 (十六进制): AA 55 00 0E 00 D0 01 01 83 57 E7 AE 6C 02 01 00 00 7F 3D

D0 表示返回指令为读取定时任务列表, 第一个 01 表示第一个开关有 1 个定时任务, 第二个 01 表示定时任务的序号为 01, 83 表示定时任务启用, 按天执行, 执行时间为 0x57E7AE6C, 即 2016-09-25-11:01:00, 02 01 00 00 表示闭合第一路开关, 7F 表示每周的七天均执行此动作。

4.2 增加定时任务

指令符: 0x51

参数: TYPE TIME CMD WEEK

返回指令符: 0xD1

返回参数:

设置成功返回 ID TYPE TIME CMD WEEK

设置失败返回 FF

协议示例 (以设置第二路开关在 2016-09-25-17:20:00 开启, 每周七天均执行此动作为例):

控制端发送 (十六进制): 55 AA 00 0C 00 51 83 57 E8 07 40 02 02 00 00 7F E9

设备端返回 (十六进制): AA 55 00 0D 00 D1 05 83 57 E8 07 40 02 02 00 00 7F 6F

4.3 启用/停用/删除单个定时任务

指令符: 0x52

参数:

ID 表示要操作的定时任务的 ID

N 表示操作类型 1-启用, 2-停用, 3-删除

返回指令符: 0xD2

返回参数:

ID N

协议示例 (以删除 4.2 节增加的定时任务为例):

控制端发送 (十六进制): 55 AA 00 04 00 52 05 03 5E

设备端返回 (十六进制): AA 55 00 04 00 D2 05 03 DE

4.4 读取设备时间

指令符: 0x53

参数: 无

返回指令符: 0xD3

返回参数:

TIME-当前设备时间, 以 UNIX 时间戳表示。以当前时区为 UTC 协调时, 不考虑时区。

协议示例:

控制端发送 (十六进制): 55 AA 00 02 00 53 55

设备端返回 (十六进制): AA 55 00 06 00 D3 57 E8 07 4C 6B

4.5 设置系统时间

指令符: 0x54

参数:

TIME-设置设备的时间为 APP/软件的当前时间, 以当前时区为 UTC 协调时, 不考虑时区。

返回指令符: 0xD4

返回参数:

Result-为 1 表示设置时间成功, 为 0 表示设置时间失败

TIME-设置时间的时间戳

协议示例:

控制端发送 (十六进制): 55 AA 00 06 00 54 57 E8 08 79 1A

设备端返回 (十六进制): AA 55 00 07 00 D4 01 57 E8 08 79 9C

5 资源命名指令

为了便于用户使用设备，设置了资源命名指令。用户可以使用资源命名指令来为开关或者其他的设备上的资源设置名称。设备名称是存储在硬件设备里的，用户在任何情况下都可以获取到。

每个命名空间的大小为 14 字节，第一个字节用于标记此资源的属性（暂未使用，默认为 0），第二个字节备用，后面 12 个字节存储名称，采用 Unicode-8 编码存储，字母最多可以设置 12 个字节，汉字最多允许设置 4 个汉字。如果名称不足 12 个字节，则用 0 填充。如果读取到的值是全为 0x00 或者 0xFF，APP 显示未命名。硬件可以根据设备的实际情况给资源一个初始默认名称。

5.1 设置单个资源名称

指令符：0x60

参数：

N-资源类型，为 0 表示输出开关，为 1 表示输入开关，为 2 表示 PWM，为 3 表示寄存器

M-表示第几路，例如 N=0，M=1，则表示输出开关的第一路

DDDDD-14 个字节数据，记录所设置资源的名称

返回指令符：0xE0

返回参数：

N-资源类型，为 0 表示输出开关，为 1 表示输入开关，为 2 表示 PWM，为 3 表示寄存器

M-表示第几路，例如 N=0，M=1，则表示输出开关的第一路

DDDDD-14 个字节数据，记录所设置资源的名称

协议示例：

控制端发送（十六进制）：

55 AA 00 12 00 60 00 01 00 00 54 45 53 54 00 00 00 00 00 00 00 00 00 B3

设备端返回（十六进制）：

AA 55 00 12 00 E0 00 01 00 00 54 45 53 54 00 00 00 00 00 00 00 00 00 33

5.2 读取单路资源名称

指令符: 0x61

参数:

N-资源类型, 为 0 表示输出开关, 为 1 表示输入开关, 为 2 表示 PWM, 为 3 表示寄存器

M-表示第几路, 例如 N=0, M=1, 则表示输出开关的第一路

返回指令符: 0xE1

返回参数:

N-资源类型, 为 0 表示输出开关, 为 1 表示输入开关, 为 2 表示 PWM, 为 3 表示寄存器

M-表示第几路, 例如 N=0, M=1, 则表示输出开关的第一路

DDDDD-14 个字节数据, 记录所设置资源的名称

协议示例:

控制端发送 (十六进制):

55 AA 00 04 00 61 00 01 66

设备端返回 (十六进制):

AA 55 00 07 00 E1 00 01 00 00 54 45 53 54 00 00 00 00 00 00 00 00 29

5.3 读取所有资源名称

指令符: 0x63

参数:

DDDDD-14×N 个字节数据, 记录所有可以命名的资源名称。N 为设备所支持命名的全部资源数量

返回指令符: 0xE3

返回参数:

DDDDD-14×N 个字节数据, 记录所有可以命名的资源名称。N 为设备所支持命名的全部资源数量

协议示例:

控制端发送 (十六进制):

55 AA 00 02 00 63 65

设备端返回 (十六进制):

AA 55 00 9C 00 E3 00 00 54 45 53 54 00 00 00 00 00 00 00 00 00 00 32 00 00 00 00 00 00 00 00 00 00 00 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 34 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 36 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 37 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C8

6 特殊指令

特殊指令用于一些设备信息相关的、以及一些后续扩展的指令。

6.1 读取设备资源配置

指令符: 0x7E

参数: 无

返回指令符: 0xFE

返回参数:

X Y Z K M

X 表示输出开关数量, Y 表示 开关数量, Z 表示 PWM 资源数量, K 表示寄存器数量, M 表示红外资源数量。依次排列, 红外资源没有可以缺省。

协议示例:

控制端发送 (十六进制): 55 AA 00 02 00 7E 80

设备端返回 (十六进制): AA 55 00 07 00 FE 08 03 00 00 0F

6.2 保存当前状态

指令符: 0x7A

参数: 无

功能: 保存当前的开关状态等, 在设备重新通电后仍然按当前状态运行

返回指令符: 0xFA

返回参数:

无

协议示例:

控制端发送 (十六进制): 55 AA 00 02 00 7A 7C

设备端返回 (十六进制): AA 55 00 02 00 FA FC

6.3 ID 设置指令

指令符: 0x7F

参数:

ID, 即为设备的 ID, 在设备支持此功能的情况下, 设置设备 ID 后, 协议指令必须指定所要控制的设备 ID。

功能: 设置当前设备的 ID 序号

返回指令符: 0x8F

返回参数: ID

协议示例:

控制端发送 (十六进制): 55 AA 00 06 00 7F 01 83

设备端返回 (十六进制): AA 55 00 07 00 8F 01 93

设置 ID 后, 再次发送控制指令示例 (以取反第一路输出开关状态为例):

控制端发送 (十六进制): 55 AA 00 03 01 03 01 08

设备端返回 (十六进制): AA 55 00 04 01 83 01 00 89

6.4 延时设置指令

指令符: 0x4A

参数:

DATA, 表示延时间, 在设备支持此功能的情况下, 如果设置延时时间, 那么当控制闭合某一路输出开关后, 开关在延时 DATA 秒的时间后自动关闭。延时时间 1~59 秒, 超出此范围则为正常控制方式。

功能: 设置延时时间, 以使在发送输出开关闭合指令后 DATA 秒后自动断开开关。

返回指令符: 0xCA

返回参数:

DATA

协议示例:

控制端发送 (十六进制): 55 AA 00 03 00 4A 01 83

设备端返回 (十六进制): AA 55 00 03 00 CA 01 93

设置后, 如果发送控制指令闭合某一路输出开关, 则此开关闭合 1 秒后即断开。

7 修订表:

编号	版本	修订人	修订内容	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	王伟	增加输入 IO 状态存取指令; 增加读取设备资源和状态	2013-04-23
4	V1.4	古欣	增加自动控制规则	2013-08-07
5	V1.5	李慧斌	增加搜索协议	2013-08-20
6	V2.0	陈超	修改文档架构, 完善控制指令的描述。	2016-09-27

8 免责声明

本文档提供有关 GPIO 控制协议的信息。用户可以依据本文档的描述, 使用符合用户需求的编程语言及控制方式, 来实现对掌控宝 品牌下的设备进行控制和操作。

本文档会根据 掌控宝 产品开发及升级的情况, 适当进行更新和维护。对于文档及产品的更新, 恕不另行通知。

请用户及时关注 [掌控宝](http://www.zhangkongbao.com) 网站, 以获取产品和文档的最新信息。