

**Doctors of Intelligence &Technology(DOIT)**

**DoHome**

（version: V3.8.5）

**2019-06-10**

目录

[1、发送信息格式 9](#_Toc10108670)

[2、TCP连接 10](#_Toc10108671)

[3、TCP包格式 11](#_Toc10108672)

[3.1 ESP->App 11](#_Toc10108673)

[3.2 App->ESP 11](#_Toc10108674)

[4、手机与灯的通讯（App<->ESP） 12](#_Toc10108675)

[4.1 获取周边WiFi扫描结果 12](#_Toc10108676)

[4.2 修改AP热点名字 13](#_Toc10108677)

[4.3 ESP重启 14](#_Toc10108678)

[4.4 获取设备信息 15](#_Toc10108679)

[4.5 开关灯 16](#_Toc10108680)

[4.6 调色 17](#_Toc10108681)

[4.7 设置动态模式 18](#_Toc10108682)

[4.8 自定义动态模式 20](#_Toc10108683)

[4.9 获取设备时间 21](#_Toc10108684)

[4.10 同步设备时间 22](#_Toc10108685)

[4.11 设置上电开灯状态 23](#_Toc10108686)

[4.12 设置为记住断电前状态 25](#_Toc10108687)

[4.13 设置定时关灯 26](#_Toc10108688)

[4.14 设置定时开灯 27](#_Toc10108689)

[4.15远程操作使能 31](#_Toc10108690)

[4.16 设置连接路由 31](#_Toc10108691)

[4.17 设置延时关灯 32](#_Toc10108692)

[4.18 OTA 33](#_Toc10108693)

[4.19 查询模块是否已连上路由 34](#_Toc10108694)

[4.20 查询固件版本号 34](#_Toc10108695)

[4.21 查询设备上的定时器信息 35](#_Toc10108696)

[4.22 查询延时关灯设置 37](#_Toc10108697)

[4.23 取消定时器（也可用于取消延时关灯） 38](#_Toc10108698)

[4.24 查询上电开灯状态 39](#_Toc10108699)

[4.25 查询当前灯状态 42](#_Toc10108700)

[4.26 修改定时器时间（延时关灯时间无需修改，直接设置新延时时间） 44](#_Toc10108701)

[4.27 预设动态模式组合 46](#_Toc10108702)

[4.28 重置AP 47](#_Toc10108703)

[4.29 时区设置 47](#_Toc10108704)

[4.30 开启中继功能 48](#_Toc10108705)

[4.31 开启Portal功能 49](#_Toc10108706)

[4.32 修改Portal页面显示内容 50](#_Toc10108707)

[4.33 调节亮度百分比 51](#_Toc10108708)

[4.34 设置亮度为某百分比值 52](#_Toc10108709)

[4.35 设置灯带RGB管脚 53](#_Toc10108710)

[4.36 查询当前固件slot 53](#_Toc10108711)

[5、灯与远程服务器的通讯（ESP<->Server） 54](#_Toc10108712)

[5.1 控制设备： 54](#_Toc10108713)

[5.2 保持心跳(每隔120秒): 55](#_Toc10108714)

[5.3数据上传，基于http 55](#_Toc10108715)

[5.4 修改设备远程控制权限 55](#_Toc10108716)

[5.5 获取时间 56](#_Toc10108717)

[5.6 实时通知手机终端 56](#_Toc10108718)

[6、手机客户端设备发现过程 57](#_Toc10108719)

[7、模块复位 58](#_Toc10108720)

[8、时区设置 59](#_Toc10108721)

[9、模块获取当前时间 60](#_Toc10108722)

[10、OTA升级（新增包序号） 61](#_Toc10108723)

[10.1 OTA数据传输协议 61](#_Toc10108724)

[10.2 OTA流程 64](#_Toc10108725)

[10.3 OTA Python客户端 65](#_Toc10108726)

[10.4 OTA校验 66](#_Toc10108727)

[10.5 App中实现OTA功能 66](#_Toc10108728)

[11、设备主动广播UDP 67](#_Toc10108729)

[12、UDP广播控制方式 68](#_Toc10108730)

[13、注意事项 69](#_Toc10108731)

修改日志

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改内容 | 修改日期 | 修改人 | 备注 |
| 1.5.2 | 增加定时重复功能 | 2018.1.25 | 王炜 | 无 |
| 1.5.3 | 增加动态模式重复执行功能 | 2018.1.26 | 王炜 | 无 |
| 1.5.4 | 补充设备发现内容 | 2018.1.30 | 王炜 | 无 |
| 1.6.0 | 增加时区设置功能 | 2018.2.3 | 王炜 | 无 |
| 1.6.1 | (5.6节)增加实时通知手机接口 | 2018.2.26 | 李洪刚 | 无 |
| 2.0.0 | (4.30)增加中继功能 | 2018.2.27 | 王炜 | 待定 |
| 2.0.1 | 底层改为使用espconn后，app与模块的tcp连接15秒内必须有数据通信，否则tcp连接自动断开。不限制TCP连接数量。 | 2018.2.27 | 王炜 | 无 |
| 2.1.0 | 修改定时器设置功能。App设置定时器时发送加上字段ts，取消定时器时App通过发送ts取消某个定时器。需要修改延时关灯时，直接设置新延时时间即可，模块会先取消旧的延时定时器，然后设置新的。 | 2018.3.2 | 王炜 | 无 |
| 2.1.1 | 4.4添加repeater字段，表示中继功能是否开启。 | 2018.3.6 | 王炜 | 无 |
| 3.0.0 | 4.31增加Portal功能，4.32增加修改Portal页面文字功能 | 2018.3.7 | 王炜 | 无 |
| 3.0.1 | 4.4添加Portal字段表示portal功能是否开启 | 2018.3.7 | 王炜 | 无 |
| 3.1.0 | 4.33增加调节亮度百分比功能（该功能供Alexa使用） | 2018.3.7 | 王炜 | 无 |
| 3.2.0 | 4.34增加设置亮度百分比功能（该功能供Alexa使用） | 2018.3.8 | 王炜 | 无 |
| 3.3.0 | 11节增加OTA升级说明 | 2018.3.18 | 王炜 | 无 |
| 3.4.0 | 修改OTA功能 | 2018.4.12 | 王炜 | 无 |
| 3.5.0 | 给App添加OTA功能 | 2018.4.20 | 王炜 | 无 |
| 3.6.0 | 设备连上路由后主动向端口6095广播udp消息，告诉App设备信息 | 2018.6.19 | 王炜 | 无 |
| 3.6.1 | 增加灯带管脚设置功能 | 2018.6.19 | 王炜 | 无 |
| 3.6.2 | 灯带设置修改 | 2018.7.9 | 王炜 | 无 |
| 3.6.3 | 增加app关灯记忆功能。App中再次开灯后显示上次关灯状态。App发送cmd：5开关灯 | 2018.7.23 | 王炜 | 无 |
| 3.7.0 | 支持UDP广播命令，设备也可以通过UDP命令控制；修改开关灯命令方式，增加了on字段，兼容老固件。做了数据备份区，防止数据读写失败造成的数据丢失 | 2018.12.6 | 王炜 | 无 |
| 3.7.1 | 设备收到UDP广播命令后返回UDP应答，cmd=6，即调色命令不返回 | 2018.12.7 | 王炜 | 无 |
| 3.7.2 | Cmd4返回内容增加tz字段，表示设备当前使用的时区。向服务器上传的扩展信息中也带有tz字段，用于app远程获取设备时区 | 2018.12.12 | 王炜 | 无 |
| 3.8.0 | 修改OTA协议，增加包序号字段。并且在写入每一包数据之后再读取一次数据确认写入内容是否正确！ | 2018.12.24 | 王炜 | 无 |
| 3.8.1 | Cmd6增加temp字段，区分调彩光还是冷暖光；cmd25返回的type字段增加类型4，表示当前处于色温模式。 | 2019.01.12 | 王炜 | 无 |
| 3.8.2 | 增加Cmd36，udp广播自定义消息的功能；增加my9291驱动；固件修改频闪灯光效果 | 2019.05.17 | 王炜 | 无 |
| 3.8.3 | Cmd16增加bssid字段 | 2019.5.27 | 王炜 | 无 |
| 3.8.5 | 对于rtl7697，其Cmd4返回内容中加上chip字段，上传的扩展字段中也加上该字段 |  |  |  |
| 3.8.6 | 增加文件上传获取接口 | 2020.6.12 | 李洪刚 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 1、发送信息格式

统一使用JSON格式发送消息，返回结果。例如，app发送一个命令：

{

“cmd”:1,

”arg”:”arguments”

}

ESP返回操作结果：

{

“res”:0,

“cmd”:1

}

## 2、TCP连接

手机连上模块热点后建立TCP通信，端口5555。IP：192.168.4.1

## 3、TCP包格式

### 3.1 ESP->App

每个消息以“\r\n”结尾

例如：{“cmd”:9}\r\n

### 3.2 App->ESP

同上

## 4、手机与灯的通讯（App<->ESP）

### 4.1 获取周边WiFi扫描结果

#### 4.1.1 App->ESP

获取Wifi信息，App发送cmd=1给ESP。如下：

{

“cmd”:1,

}

#### 4.1.2 ESP->App

ESP给App返回操作结果，res包含错误码、wifi字段包含扫描结果

{

“res”:0,

“cmd”:1,

“wifi”:

[

{“ssid”:”aaa”, ”enc”:0, “rssi”:56},

{“ssid”:”bbb”, ”enc”:0, “rssi”:67},

{“ssid”:”ccc”, ”enc”:1, “rssi”:89},

]

}

Res字段表示扫描结果，0表示扫描成功，非0表示错误码。

“wifi”字段包含扫描到的wifi信息，如果res不为0，则本字段为空，wifi字段中每组数据中ssid字段为wifi的名称；enc字段表示是否加密，0是没加密，非零是加密；rssi字段为信号强度

### 4.2 修改AP热点名字

#### 4.2.1 App->ESP

App发送cmd=2，ssid填写新热点名；

{

“cmd”:2,

“ssid”:”abc”,

“pass”:”88888888”

}

如果同时修改ssid和密码则填写ssid和pass字段，pass为新密码，否则ssid不填，或者pass不填，则表示仅修改一项，例如只修改ssid：

{

“cmd”:2,

“ssid”:”abc”,

}

注意，密码一定要大于8位数

#### 4.2.2 ESP->App

{

“res”:0,

“cmd”:2

}

设置成功，ESP返回cmd=0，否则返回非零值。

### 4.3 ESP重启

#### 4.3.1 App->ESP

Cmd=3

{

“cmd”:3,

}

#### 4.3.2 ESP->App

重启ESP，没有返回。重启一定成功

### 4.4 获取设备信息

#### 4.4.1 App->ESP

{

“cmd”:4

}

#### 4.4.2 ESP->App

{

“res”:0,

“cmd”:4,

“tz”:19,

“dev\_id”:”AB0352EB”,

“ver”:”1.1.0”,

“conn”:1,

“remote”:0,

“save\_off\_stat”:0,

“repeater”:0,

“portal”:0,

“chip”:”rtl8710” // esp8266没有该字段。没有该字段app默认该设备是8266

}

Res=0表示成功，非零值表示失败，info字段包含设备信息。Conn字段表示是否连上路由，0表示未连上，1连上。Dev\_id为设备id，是字符串；ver字段是固件版本号。Remote字段表示远程控制是否使能，1表示使能。Save\_off\_stat字段表示是否记录断电前状态，1表示记录，0不记录。Repeater字段表示中继功能是否开启，1表示已经开启。Portal字段表示portal功能是否开启，1表示开启。Tz字段是4.0.9以后的版本添加的，表示时区需要其他信息后续可以添加。

### 4.5 开关灯

#### 4.5.1 App->ESP

{

“cmd”:5,

“op”:0

}

Op字段表示App开关灯操作。0表示关灯；1开灯。

#### 4.5.2 ESP->App

{

“res”:0,

“cmd”:5

}

Res=0为操作成功，非零表示操作失败

### 4.6 调色

#### 4.6.1 App->ESP

{

“cmd”:6,

“r”:0,

“g”:1,

“b”:2,

“w”:3,

“m”:4,

“on”:1,

“temp”:1,

“smooth”:1,

“t”:1,

}

R，g，b，w，m几个字段分别表示红、绿、蓝、白、暖光的颜色值，颜色值取值范围是0~5000；其中on字段是4.0.5版本固件后增加的，on=1表示这命令是app开灯。固件会恢复上次关灯状态；on=0表示这是app关灯，固件会先保存关灯前状态再关灯。4.0.5版本以前的固件会忽略该字段。Temp字段是4.2.0版本以后的增加的字段，表示app设置色温模式。因为有些灯的冷暖光是用红绿蓝三基色混合出来的，所以必须用一个字段区别当前app发送的指令是调彩光还是调冷暖光。

Smooth:使能平滑开关，=1:使用平滑，0:不使用平滑，直接跳转到该颜色

t:选填，单步平滑时间，不填写时使用默认，单位ms，推荐范围0-20

#### 4.6.2 ESP->App

{

“res”:0,

“cmd”:6

}

Res=0为操作成功，非零表示操作失败

### 4.7 设置动态模式

#### 4.7.1 App->ESP

{

“cmd”:7,

“index”:0,

“freq”:22

}

index字段表示选择的模式编号。Freq为速度值

预设的动态模式索引分别对应如下：

Index=1：七彩渐变

2：红色渐变

3：绿色渐变

4：蓝色渐变

5：黄色渐变

6：青色渐变

7：紫色渐变

8：白色渐变

9：红色频闪

10：绿色频闪

11：蓝色频闪

12：黄色频闪

13：红绿渐变

14：红蓝渐变

15：绿蓝渐变

16：红绿跳变

17：红蓝跳变

18：绿蓝跳变

19：红绿频闪

20；红蓝频闪

21：绿蓝频闪

22：七彩跳变

23：七彩频闪

24：白色频闪

25：三色渐变

26：三色跳变

27：三色频闪

#### 4.7.2 ESP->App

{

“res”:0,

“cmd”:7

}

Res=0为操作成功，非零表示操作失败

### 4.8 自定义动态模式

#### 4.8.1 App->ESP

{

“cmd”:8,

“colors”:

[

{“r”:1,”g”:2,”b”:3,”w”:4},

{“r”:1,”g”:2,”b”:3,”w”:4},

{“r”:1,”g”:2,”b”:3,”w”:4},

……

],

“mode”:1,

“freq”:22

}

Colors字段是16个颜色值；mode=1表示设置为渐变，2为跳变，3为频闪；freq是速度值

#### 4.8.2 ESP->App

{

“res”:0,

“cmd”:8

}

### 4.9 获取设备时间

#### 4.9.1 App->ESP

{

“cmd”:9

}

#### 4.9.2 ESP->App

{

“res”:0,

“cmd”:9,

“year”:2018,

“month”:1,

“day”:10,

“hour”:19,

“minute”:20,

“second”:10

}

Res=0为操作成功，非零表示操作失败。year字段是年份，24小时格式

### 4.10 同步设备时间

#### 4.10.1 App->ESP

{

“cmd”:10,

“year”:2018,

“month”:1,

“day”:10,

“hour”:19,

“minute”:20,

“second”:10

}

year字段是年。用于在没有网络的情况下，手机给设备同步时间。如果设备已经连上网络，则设备会定时从服务器同步时间（默认同步为北京时间，通过设定时区可以修改同步时间），此时手机同步时间功能将失效。

#### 4.10.2 ESP->App

{

“res”:0,

“cmd”:10

}

Res=0为操作成功，非零表示操作失败。

### 4.11 设置上电开灯状态

#### 4.11.1 App->ESP

（1）设置开灯显示预定义动态模式

如果设置为预定义动态模式则index为1-20，此时只需cmd、freq、index、type字段，其他字段可以去掉，设置预定义模式时，type=1，例如发送下面数据设置预定义动态模式：

{

“cmd”:11,

“freq”:22,

“index”:1,

“type”:1

}

（2）设置开灯显示自定义动态模式

如果使用自定义动态模式，则需填cmd、freq、mode、colors字段，例如：

{

“cmd”:11,

“type”:2,

“colors”:

[

{“r”:1,”g”:2,”b”:3,”w”:4},

{“r”:1,”g”:2,”b”:3,”w”:4},

{“r”:1,”g”:2,”b”:3,”w”:4},

……

],

“mode”:1,

“freq”:22

}

mode=1表示设置为渐变，2为跳变，3为频闪；freq是速度值；index为模式值。

（3）设置开灯显示固定颜色

此时type字段填3，只有如下字段，例如：

{

“cmd”:11,

“type”:3,

“r”:0,

“g”:1,

“b”:2,

“w”:3,

“m”:4

}

#### 4.11.2 ESP->App

{

“res”:0,

“cmd”:11

}

Res=0为操作成功，非零表示操作失败。

### 4.12 设置为记住断电前状态

#### 4.12.1 App->ESP

{

“cmd”:12,

“op”:1

}

Op字段表示操作，0表示不记住断电前状态；1表示记住。

#### 4.12.2 ESP->App

{

“res”:0,

“cmd”:12

}

Res=0为操作成功，非零表示操作失败。

### 4.13 设置定时关灯

#### 4.13.1 App->ESP

{

“cmd”:13,

“ts”:12344556,

“year”:2018,

“month”:1,

“day”:10,

“hour”:19,

“minute”:20,

“second”:10,

“repeat”:0

}

year、month关机时间。repeat字段表示是否重复，1为重复。如果repeat=1，则year、month、day无用。Ts字段为App生成的唯一值，用来取消定时器。这里可能会改

#### 4.13.2 ESP->App

{

“res”:0,

“cmd”:13

}

Res=0为操作成功，非零表示操作失败。

### 4.14 设置定时开灯

#### 4.14.1 App->ESP

可以设置开灯时的状态为：固定颜色值、预设动态模式、自定义动态模式

{

“cmd”:14,

“ts”:12344556,

“year”:2018,

“month”:1,

“day”:10,

“hour”:19,

“minute”:20,

“second”:10,

“r”:1,

“g”:2,

“b”:3,

“w”:4,

“m”:5,

“type”:1,

“repeat”:0

}

Index字段为选择的模式编号；mode=1表示设置为渐变，2为跳变，3为频闪；type=1表示设定开灯时为固定颜色值

如果设置为预定义动态模式则index为1-16，此时只需cmd、freq、index字段，其他字段可以去掉，例如发送下面数据设置开灯为预定义动态模式：

{

“cmd”:14,

“year”:2018,

“month”:1,

“day”:10,

“hour”:19,

“minute”:20,

“second”:10,

“freq”:22,

“index”:1,

“type”:2,

“repeat”:0

}

如果使用自定义动态模式，则需填cmd、freq、mode、colors字段，例如：

{

“cmd”:14,

“year”:2018,

“month”:1,

“day”:10,

“hour”:19,

“minute”:20,

“second”:10,

“colors”:

[

{“r”:1,”g”:2,”b”:3,”w”:4},

{“r”:1,”g”:2,”b”:3,”w”:4},

{“r”:1,”g”:2,”b”:3,”w”:4},

……

],

“mode”:1,

“freq”:22,

“type”:3,

“repeat”:0

}

mode=1表示设置为渐变，2为跳变，3为频闪；freq是速度值；index为模式值。

#### 4.14.2 ESP->App

{

“res”:0,

“cmd”:14

}

Res=0为操作成功，非零表示操作失败。

### 4.15远程操作使能

#### 4.15.1 App->ESP

{

“cmd”:15,

“op”:0

}

Op=1表示开启远程操作，0表示关闭远程

#### 4.15.2 ESP->App

{

“res”:0,

“cmd”:15

}

Res=0为操作成功，非零表示操作失败。

### 4.16 设置连接路由

#### 4.16.1 App->ESP

{

“cmd”:16,

“ssid”:”test\_ssid”,

“pass”:”12345567”

“bssid”:”12:b3:e8:33:90:7d”

}

Bssid是路由器的bssid

#### 4.15.2 ESP->App

{

“res”:0,

“cmd”:16,

“dev\_id”:fsdfsdg\_LED\_DOIT,

“ssid”:”Doit”,

“pass”:”doit3305”

}

Res=0为操作成功，非零表示操作失败。Ssid和pass字段用于给app显示，让用户知道是否输错路由密码

### 4.17 设置延时关灯

#### 4.17.1 App->ESP

{

“cmd”:17,

“time”:15,

“ts”:12344556

}

Time是定时关灯的时间，单位是分钟

#### 4.17.2 ESP->App

{

“res”:0,

“cmd”:17

}

Res=0为操作成功，非零表示操作失败。

### 4.18 OTA

OTA使用udp方式实现，手机App，或者python客户端，建立udp连接（通过连接模块，或者通过路由器广播都可以），端口为6091。

广播字符串“cmd=ota\_begin\r\n”模块开始进入接收bin文件状态，然后持续发送bin文件数据。每包发送256字节。每次延迟50ms发送下一包。客户端发送完bin文件后，广播字符串“cmd=ota\_end\r\n”通知模块结束升级，然后模块自动重启运行新固件。

### 4.19 查询模块是否已连上路由

#### 4.19.1 App->ESP

{

“cmd”:19

}

#### 4.19.2 ESP->App

{

“res”:0,

“cmd”:19,

“ip”:”192.168.1.100”

}

Res=0为连接路由成功，并带有IP字段，ip字段为分配到的ip值；如果res为非零表示没有连接路由，此时ip字段为空。

### 4.20 查询固件版本号

#### 4.20.1 App->ESP

{

“cmd”:20

}

#### 4.20.2 ESP->App

{

“res”:0,

“cmd”:20,

“ver”:”1.1.0”

}

### 4.21 查询设备上的定时器信息

#### 4.21.1 App->ESP

{

“cmd”:21

}

查询所有已经设置的定时器，包括定时开关灯、延时关灯的定时器。

#### 4.21.2 ESP->App

{

“res”:0,

“cmd”:21,

“timers”:

[

{“index”:1,”type”:0,”year”:2018,”mon”:1,”day”:2,”hour”:10,”min”:2,”sec”:20,”repeat”:0},

{“index”:2,”type”:1,”r”:1,”g”:2,”b”:3,”w”:5,”m”:9, ”year”:2018,”mon”:1,”day”:2,”hour”:10,”min”:2,”sec”:20,”repeat”:1},

{“index”:3,”type”:2,”index2”:3,”freq”:30, ”year”:2018,”mon”:1,”day”:2,”hour”:10,”min”:2,”sec”:20,”repeat”:1}

{“index”:4,”type”:3,”mode”:1,”freq”:30 ,”repeat”:1,”year”:2018,”mon”:1,”day”:2,”hour”:10,”min”:2,”sec”:20,”colors”:[{“r”:1,”g”:3,”b”:4,”w”:2,”m”:2},{“r”:1,”g”:3,”b”:4,”w”:2,”m”:2}, ……]},

{“index”:5,”type”:4,”year”:2018,”mon”:7,”day”:23,”hour”:19,”min”:2,”sec”:2, ,”repeat”:0 },

]

}

Index字段表示定时器索引，可用于取消定时器。index2是定时预设动态模式索引号。Type表示定时操作类型，type=0表示定时关灯；1表示定时开灯，并且开灯后显示指定颜色；2表示定时开灯，且开灯后显示预定义的动态模式；3表示定时开灯，且开灯后显示自定义动态模式；4表示延时关灯，延时关灯的repeat字段只能是0，year，mon，day等字段表示定时到期时间，24小时制。

如果没有设置定时器，则timers字段为空数组，例如

{

“res”:0,

“cmd”:21,

“timers”:[]

}

### 4.22 查询延时关灯设置

#### 4.22.1 App->ESP

{

“cmd”:22

}

查询结果仅仅是延时关灯的定时器，不包括定时开灯的定时器。

#### 4.22.2 ESP->App

{

“res”:0,

“cmd”:22,

“timers”:

[

{“index”:5,”year”:2018,”mon”:7,”day”:23,”hour”:19,”min”:2,”sec”:2,”ts”:4234566},

]

}

### 4.23 取消定时器（也可用于取消延时关灯）

#### 4.23.1 App->ESP

{

“cmd”:23,

“ts”:3344356

}

ts字段是App设置timer时传给模块的ts值，该值也可从4.22中查询到，该值是App生成并使用的。

#### 4.23.2 ESP->App

{

“res”:0,

“cmd”:23

}

取消成功，返回res=0

### 4.24 查询上电开灯状态

#### 4.24.1 App->ESP

{

“cmd”:24

}

#### 4.24.2 ESP->App

{

“res”:0,

“cmd”:24,

“type”:0,

……

}

Type字段与4.11.1中的type字段一样，当type=1，表示开灯显示预定义动态模式，返回结果如下；

{

“res”:0,

“cmd”:24,

“type”:1,

“freq”:22,

“index”:1

}

Index为预定义动态模式索引值，对应4.7.1中所述各种模式；

Type=2表示开灯显示自定义动态模式，返回如下；

{

“res”:0,

“cmd”:24,

“type”:2,

“freq”:22,

“mode”:1,

“colors”:

[

{“r”:1,”g”:3,”b”:4,”w”:2,”m”:2},

{“r”:1,”g”:3,”b”:4,”w”:2,”m”:2},

……

]

}

mode=1表示设置为渐变，2为跳变，3为频闪；

type=3表示开灯显示固定颜色。此时返回如下字段：

{

“res”:0,

“cmd”:24,

“type”:3,

“r”:1,

“g”:9,

“b”:2,

“w”:8,

“m”:0,

}

如果没有设置上电状态，则type=0，并且没有其他字段，例如：

{

“res”:0,

“type”:0,

}

### 4.25 查询当前灯状态

#### 4.25.1 App->ESP

{

“cmd”:25

}

#### 4.25.2 ESP->App

当前灯处于固定颜色值状态，返回如下：

{

“res”:0,

“cmd”:25,

“r”:2,

“g”:30,

“b”:2,

“w”:4,

“m”:8,

“type”:1,

}

当前灯处于预设动态模式，返回如下信息：

{

“res”:0,

“cmd”:25,

“index”:2,

“freq”:30,

“type”:2,

}

当前灯处于自定义动态模式

{

“res”:0,

“cmd”:25,

“mode”:2,

“freq”:30,

“colors”:

[

{“r”:3,”g”:3,”b”:3,”w”:3,”m”:3},

{“r”:3,”g”:3,”b”:3,”w”:3,”m”:3},

……

],

“type”:3,

}

当前灯处于色温模式：

{

“res”:0,

“cmd”:25,

“r”:2,

“g”:30,

“b”:2,

“w”:4,

“m”:8,

“type”:4

}

### 4.26 修改定时器时间（延时关灯时间无需修改，直接设置新延时时间）

#### 4.26.1 App->ESP

{

“cmd”:26,

“index”:1,

“year”:2018,

“mon”:8,

“day”:8,

“hour”:11,

“min”:22,

“sec”:1,

“repeat”:1,

}

修改定时器时间，hour字段为24小时制，index字段为定时器索引，范围0~6，总共七个定时器。当修改的是用于延时关灯的定时器时，repeat字段无效。

#### 4.26.2 ESP->App

{

“res”:0,

“cmd”:26,

}

### 4.27 预设动态模式组合

#### 4.27.1 App->ESP

{

“cmd”:27,

“repeat”:1,

“list”:

[

{“index”:1,”freq”:30,”loop”:3},

{“index”:2,”freq”:40,”loop”:6},

……

]

}

List字段包含了各个预设动态模式的信息，index表示动态模式索引，loop是指该动态模式执行多少次。repeat字段表示动态模式组合是否重复执行，0表示只执行一遍，1表示重复执行。最多设置16个动态模式组合！

#### 4.27.2 ESP->App

{

“res”:0,

“cmd”:27,

}

### 4.28 重置AP

#### 4.28.1 App->ESP

{

“res”:0,

“cmd”:28,

}

#### 4.28.2 ESP->App

无返回，模块直接重启

### 4.29 时区设置

#### 4.29.1 App->ESP

{

“cmd”:29,

“offset”:2,

}

offset字段表示时区偏移量，是与北京时间，东八区的偏移。默认使用北京时间，其offset是19。相邻时区间差正好1小时。24个时区描述见第九节。这里offset取值从0至23。例如设置使用东京时间，位于东九区，则offset=20。

#### 4.29.2 ESP->App

{

“res”:0,

“cmd”:29,

}

成功返回0，失败返回非零

### 4.30 开启中继功能

#### 4.30.1 App->ESP

{

“cmd”:30,

“en”:1,

}

En=1表示开启中继功能，反之，关闭。开启后，模块重启进入中继模式。手机连上模块后可以打开网页访问网络。前提是要配置连接路由。所以App里面应该先判断是否已经连上路由，然后才能允许用户使能中继功能。查询中继功能是否开启可以使用4.4节的命令查询。该功能默认不开启。

#### 4.30.2 ESP->App

{

“res”:0,

“cmd”:30,

}

返回0表示操作成功。返回44表示模块没有连上路由，返回其他非零值都是操作失败。

### 4.31 开启Portal功能

#### 4.31.1 App->ESP

{

“cmd”:31,

“en”:1,

}

En=1表示开启Portal功能，0为关闭。

#### 4.31.2 ESP->App

{

“res”:0,

“cmd”:31,

}

返回0表示成功，非零为失败

### 4.32 修改Portal页面显示内容

#### 4.32.1 App->ESP

{

“cmd”:32,

“text”:”hello world”,

}

Text字段是Portal页面显示的字符串内容

#### 4.32.2 ESP->App

{

“res”:0,

“cmd”:32,

}

返回0表示成功，非零为失败

### 4.33 调节亮度百分比

#### 4.33.1 App(Server)->ESP

{

“cmd”:33,

“sign”:0,

“val”:30

}

Sign表示符号，val表示百分比，上例表示亮度在现有基础上-30%，sign=1表示正百分比。这是调节白光亮度，红绿蓝被关闭。

#### 4.33.2 ESP-> App(Server)

{

“res”:0,

“cmd”:32,

}

返回0表示成功，非零为失败

### 4.34 设置亮度为某百分比值

#### 4.34.1 App(Server)->ESP

{

“cmd”:34,

“val”:30

}

val表示百分比，表示亮度设置为30%，范围0~100。这是调节白光亮度，红绿蓝被关闭。

#### 4.34.2 ESP->APP(Server)

{

“res”:0,

“cmd”:34,

}

返回0表示成功，非零为失败

### 4.35 设置灯带RGB管脚

#### 4.35.1 App->ESP

{

“cmd”:35,

“type”:1

}

Type的值为0、1、2、3、4、5，即RGB管脚排列有六种类型。该命令仅用于灯带。

0对应BGR（IO4、IO15、IO13）；1对应BRG；2对应GBR；3对应GRB；4是RGB；5是RBG

#### 4.35.2 ESP->App

{

“res”:0,

“cmd”:35,

}

返回0表示成功，非零为失败

### 4.36 查询当前固件slot

Udp广播发送“cmd=bin\_slot\r\n”，模块返回当前slot，返回1表示当前使用的是user1；2则为user2

### 4.37 唤醒局域网内主机

通过wake-on-lan技术唤醒局域网内电脑主机，当前仅插座支持

#### 4.37.1 App->ESP

|  |
| --- |
| {  "cmd": 37,  "ip": "192.168.2.255",  "mac": "C8:60:00:77:F1:44",  "port": 7,  "count": 1  } |

Ip：广播地址或者目标主机ip，

Mac：目标主机mac地址

Port：目标主机端口，通常为7或9

Count：广播次数，默认为8

#### 4.37.2 ESP->App

{

“res”:0,

“cmd”:37

}

返回0表示成功，非零为失败

### 4.38

#### 4.37.1 App->ESP

|  |
| --- |
| {  "cmd": 38,  } |

#### 4.37.2 ESP->App

{

“res”:0,

“cmd”:38

}

返回0表示成功，非零为失败

### 4.39 设置homekit功能使能

打开关闭homekit功能，开启后模块进入homekit配对模式，关闭后清除homekit配置信息

#### 4.39.1 App->ESP

|  |
| --- |
| {  "cmd": 39,  "en": 1,  } |

en：homekit使能，1开启，0关闭

#### 4.39.2 ESP->App

{

“res”:0,

“cmd”:39

}

返回0表示成功，非零为失败

### 4.40 查询homekit功能使能

查询homekit功能使能状态

#### 4.40.1 App->ESP

|  |
| --- |
| {  "cmd": 40  } |

#### 4.40.2 ESP->App

{

“res”:0,

“cmd”:40,

“en”:1,

}

res：返回0表示成功，非零为失败

en：homekit使能，1开启，0关闭

### 4.41 查询保存的配网信息

查询保存的配网信息

#### 4.41.1 App->ESP

|  |
| --- |
| {  "cmd": 41  } |

#### 4.41.2 ESP->App

{

“res”:0,

“cmd”:41,

“ssid”:”xxxx”,

“password”:”xxxxxxxx”,

}

res：返回0表示成功，非零为失败

ssid：wifi名称

password:wifi密码

### 4.42 设置ota使能

开启ota功能，默认关闭ota功能，需要发送本条指令开启ota功能

#### 4.42.1 App->ESP

|  |
| --- |
| {  "cmd":200,  "otaen":1  } |

#### 4.42.2 ESP->App

{

“res”:0,

“cmd”:200

}

res：返回0表示成功，非零为失败

### 4.42 恢复出厂设置

清除所有配置信息，恢复出厂设置

#### 4.41.1 App->ESP

|  |
| --- |
| {  "cmd":201,  "en":1  } |

#### 4.41.2 ESP->App

{

“res”:0,

“cmd”:201

}

res：返回0表示成功，非零为失败

## 5、灯与远程服务器的通讯（ESP<->Server）

* 平台基于TCP通讯，服务器IP: led\_iot.doit.am 端口：8899
* 每个命令后\r\n结尾
* 设备采用STA的MAC地址+设备类型+companyID做为设备id 例如：00dd67891a\_DT-YRGB\_DOIT
* 通过下面规则生成设备 key，即字符串：设备ID+“doitxf”，的MD5值

$k=$device\_id+"doitxf";

$device\_key=md5($k);

### 5.1 控制设备：

#### 5.1.1 订阅设备

wifi模块发送：

cmd=subscribe&device\_id= S7865ABC\_LED\_DOIT&device\_key=bltw9gBbSMrjxv8Ph4YH3mQz

服务器应答：

cmd=subscribe&res=1

当有指令时收到，cmd=publish&message=消息体

### 5.2 保持心跳(每隔120秒):

cmd=keep&deivce\_id=12387&device\_key=liqin

应答：

cmd=keep&res=1

### 5.3数据上传，基于http

#### 5.3.1 状态上传（基本状态50秒传一次，扩展状态1分钟一次）

<http://xinfeng.doit.am:8008/?cmd=stat&device_id=Doit-LED-S7865ABC&device_key=bltw9gBbSMrjxv8Ph4YH3mQz&&status>=状态上传

应答:Ok

cmd=stat代表基本状态

cmd=ext\_stat 代表扩展状态

7697设备在扩展状态中加上字段“chip”:”rtl7697”

### 5.4 修改设备远程控制权限

[http://xinfeng.doit.am/mobile\_app/device\_remote.php? &device\_id=1234&device\_key=bltw9gBbSMrjxv8Ph4YH3mQz &remote\_control=1(1](http://xinfeng.doit.am/mobile_app/device_remote.php?%20&device_id=1234&device_key=bltw9gBbSMrjxv8Ph4YH3mQz%20&remote_control=1(1)代表开启远程控制、0代表关闭远程)

### 5.5 获取时间

<http://xinfeng.doit.am/iot_api/get_time.php>

### 5.6 实时通知手机终端

http://xinfeng.doit.am/mobile\_app/device\_notice.php? &device\_id=1234&device\_key=bltw9gBbSMrjxv8Ph4YH3mQz &message=通知内容( 如果绑定的手机在线，实时通知手机)

### 5.7 查询文件上传接口

手机APP显示：

<http://f.doiting.com/?id>=用户id

请求地址：

<http://dohome.doit.am/mobile_app/f.php?cmd=query&id>=用户id

参数说明：

Id:为用户上传界面的参数，通常为用户注册账号的@前面部分,

比如用户注册为 [test@doit.am，id](mailto:test@doit.am，id)为test

返回json数组：

当

{"ret":0} 代表无文件

当

Ret值为1，代表有文件，从url地址解析下载，然后调用durl删除文件

{"name":"zui.mp3","size":9884944,"type":"audio\/mpeg","url":"http:\/\/tui.doit.am\/t\/u\/server\/php\/files\/zui.mp3","size\_desc":"9.43 MB","ret":1, "durl":"http:\/\/tui.doit.am\/t\/u\/server\/php\/files\/zui.mp3"}

## 6、手机客户端设备发现过程

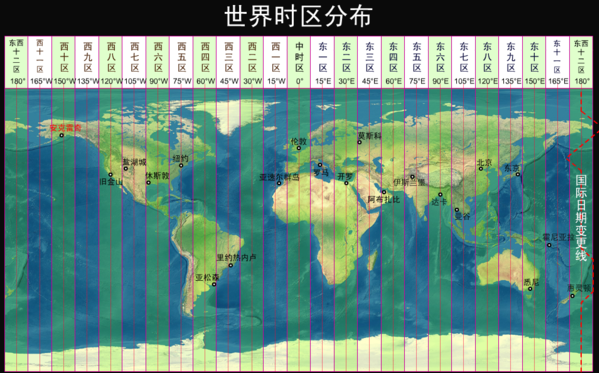
App通过向端口6091广播字符串“cmd=ping\r\n”搜索设备。设备收到该广播后返回类似如下内容“cmd=pong&host\_ip=192.168.4.1&sta\_ip=192.168.1.2&device\_id=sdfsd&device\_name=aabbcc&device\_type=\_LED&compandy\_id=\_DOIT”

## 7、模块复位

按下开关，按下面顺序可以实现设备复位“开-关-开-关-开-关-开”。两次开灯间隔不能大于10秒钟，反复开关5次后恢复出厂设置，然后可重新配网连接新路由器。

## 8、时区设置

相邻时区间相差正好1小时。下图中的城市为各个时区代表城市。



## 9、模块获取当前时间

模块发送GET请求：<http://xinfeng.doit.am/iot_api/get_iot_time.php>

## 10、OTA升级（新增包序号）

OTA升级使用TCP传输新固件。每次更新固件之前服务端可以发送“cmd \_bin\_slot\r\n”查询当前模块中运行的固件slot索引号，然后决定更新哪个slot中的固件。只能更新当前没使用的slot中的固件。

模块在端口6093建立了 TCP server，客户端连上后，通过该连接发送固件。

### 10.1 OTA数据传输协议

OTA数据包格式如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prefix | Type | Length | Payload | Checksum | Suffix |
| 包头（2字节） | 包类型（1字节） | 数据长度（1个字节表示，最大数据长度为255，默认传240个字节） | 数据 | 校验 | 包尾（1个字节，固定为0x7E） |

新的App发送包格式如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Prefix | Type | Length | Sequence | Payload | Checksum | Suffix |
| 包头（2字节） | 包类型（1字节） | 数据长度（1个字节表示，最大数据长度为255，默认传240个字节） | 包序号（占4个字节） | 数据 | 校验 | 包尾（1个字节，固定为0x7E） |

新OTA升级机制中，如果固件检测到下一个包序号与当前接收到的包中的包序号不相等，说明有包丢失，固件会中止升级重启模块。

回应包的格式：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prefix | Type | Length | Payload | Checksum | Suffix |
| 包头（2字节） | 包类型（1字节） | 数据长度（1个字节表示，最大数据长度为255，默认传240个字节） | 数据 | 校验 | 包尾（1个字节，固定为0x7E） |

对于Client发给ESP的数据包，包头固定是两个字节：0xF1 0xF1；对于ESP回应的数据包，回应内容不需要包序号。包头固定是两个字节：0xF2 0xF2。

包尾固定是0x7E。

Type字段为一个字节。有以下几种类型：

1. 0x01：该类型包由Client发给ESP，通知ESP进入OTA模式。ESP收到后，进入OTA模式，同时回应0xf2 0xf2 0x07 0x01 0x01 0x09 0x7e。之所以回应包里面数据长度设置为0x1，数据内容设置为0x1是因为如果数据长度设置为0x0，app可能认为0x0是结束符号，而把后面内容忽略掉。Client收到应答后开始发送数据包
2. 0x02：该类型包由Client发给ESP，表示该包是正常数据包。ESP收到后处理数据，同时回应0xf2 0xf2 0x08 0x01 0x01 0x0a 0x7e。Client收到应答后开始发送下一包
3. 0x03：该类型包由Client发给ESP，通知ESP发送结束。ESP收到后重启并运行新固件
4. 0x04：该类型包由ESP发给Client，表示接收正常。
5. 0x05：该类型包由Client发给ESP，查询ESP当前使用的Slot
6. 0x06：该类型包由ESP发给Client，回应Client当前使用的Slot
7. 0x07：该类型包由ESP发给Client，回应Client当前ESP已经进入OTA模式
8. 0x08：该类型包由ESP发给Client，通知Client发送下一包
9. 0x09：该类型包由Client发给ESP，表示该包是最后一个包
10. 0x0a：该类型包由ESP发给Client，通知Client已经成功收到最后一包
11. 0x20：该类型包由ESP发给Client，表示接收出现异常。ESP发送该包后中止升级，重新运行原有固件。此时，Client需要重新进行升级

### 10.2 OTA流程

下面是OTA流程：

（1）Python client给ESP发送：f1 f1 01 00 01 7e

（2）ESP进入OTA模式，并给client回应：f2 f2 07 00 07 7e（新机制回应：f2 f2 07 01 01 09 7e）

（3）Client开始发送一个数据包：f1 f1 02 f0 xx xx xx …… 89 7e

（4）ESP收到一个数据包后如果校验通过并写入flash成功，给client回应：f2 f2 08 00 08 7e，用来请求client发送下一包。如果有校验失败，则回应f2 f2 20 00 20 7e，并停止升级，重启运行老固件

重复（3）

……

重复（4）

……

……

……

（5）Client发送最后一包：f1 f1 09 f0 xx xx …… 98 7e

（6）ESP收到最后一包后应答：f2 f2 0a 00 0a 7e

（7）Client发送结束标志包：f1 f1 03 00 03 7e

（8）ESP收到结束包后重启，并运行新固件

### 10.3 OTA Python客户端

代码见ota\_client.py

### 10.4 OTA校验

校验值大小是一个字节，校验值是下面三个字段各个字节相加之和取低8位：type+length+payload

例如，client发送下面一个数据包：f1 f1 02 02 00 00 00 00 01 0a 0f 7e，则校验值0f是这样计算出来：02+02+00+00+00+00+01+0a=0f

### 10.5 App中实现OTA功能

在App中按照上述OTA协议可以实现设备固件无线升级。App首先应先从下面地址下载最新固件到手机：<http://xinfeng.doit.am/upgrade/u.php>

访问该URL后，返回的ver字段包含服务器上固件的版本号，url字段是固件下载地址。对比ver字段的版本号与App当前连接的设备的固件版本号，如果服务器上固件的版本号比设备固件版本号大，则应升级固件，否则无需升级。

## 11、设备主动广播UDP

设备连上路由后主动向端口6095发送UDP广播，每隔两秒发送一次，总共发送10次。广播内容为“cmd=pong……”，即应答“cmd=ping”的内容。主动告诉App设备信息。加快App设备发现过程，提高App发现设备的概率

## 12、UDP广播控制方式

固件版本4.0.5以后新增udp控制功能，防止出现设备失控现象。UDP广播命令格式如下：

cmd=ctrl&devices={[09b4,eee9]}&op={"cmd":6,"r":5000,"g":0,"b":0,"w":0,"m":0 }

cmd=ctrl表示该udp广播是控制命令。Devices字段内容是json格式，包含了该命令要控制的id，如果该字段包含多个设备，说明这是个分组控制命令。这个id是设备id中第一个下划线前四位数字或字母（之所以取四位是为了减少发送内容，这四位是wifi模组mac地址后四位，基本能唯一标示一台设备），例如设备id是1234567890\_LED\_DOIT ，则devices中的id对应为7890。Op字段是具体控制命令内容，与TCP发送的控制命令内容相同。

设备收到udp广播命令后，也通过udp方式返回应答。返回数据格式如下:

cmd=echo&dev=1234567890\_LED\_DOIT& op={"cmd":7,"freq":50,"index":5 }

调色命令即cmd=6无返回，避免返回数据过多，影响设备运行效率。

## 13、注意事项

1. 自定义颜色最多16个
2. freq字段最小设置为30，无上限。越大颜色变化越慢
3. AP密码必须大于8位数
4. 预设模式组合最多选取16个模式
5. App与模块的TCP连接15秒内必须有数据通信，否则连接断开