**智能家居相关终端控制协议V1.0**

**——by Q.Bryan**

***MMMMMMMMMM MMMMMMMMMMMM MMMMMMMMMMMM MMMMMMMMMMMMMMMM***

***MMMM MMMM MMMMMM MMMMMM MMMMMM MMMMMM***

***MMMM MM MMMM MMMM MMMM MMMM***

***MMMM MM MMMM MMMM MMMM MMMM***

***MMMMMM MMMM MMMM MMMM MMMM***

***MMMMMMMM MMMM MMMM MMMM MMMMMM***

***MMMMMMMM MMMMMMMMMMMMMMMMMMMMMM MMMMMMMMMMMM***

***MMMMMM MMMM MMMM MMMM***

***MMMMMM MMMM MMMM MMMM***

***MM MMMM MMMM MMMM MMMM***

***MM MMMM MMMM MMMM MMMM***

***MMMM MMMM MMMMMM MMMMMM MMMMMM***

***MMMMMMMM MMMMMMMMMMMM MMMMMMMMMMMM MMMMMMMMMMMM***

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改内容 | 修改人 | 日期 |
| V0.1 | 创建此文档 | 邱兵 | 2013-08-15 |
| V0.1 | 添加Binary、JSON实例 | 邱兵 | 2013-08-16 |
| V0.1 | 添加协商协议 | 邱兵 | 2013-08-20 |
| V0.1 | 完善协商协议 | 邱兵 | 2013-08-21 |
| V0.1 | 添加术语说明 | 孟令坤 | 2013-08-27 |
| V0.1 | 添加原子类型操作 | 邱兵 | 2013-08-27 |
| V0.1 | 修改部分术语 | 邱兵、孟令坤 | 2013-09-07 |
| V0.1 | 添加心跳包协议 | 邱兵 | 2013-09-16 |
| V0.1 | 删除桑拿房实例，添加README | 邱兵 | 2013-09-16 |
|  |  |  |  |
|  |  |  |  |

# 模型说明



图 1 总体模型

本协议是应用层协议，不涉及网络路由等细节。一般可以使用已有的底层协议（例如TCP/IP协议）。

协议约定所有字符串编码格式为GBK编码格式！！！！！！

推荐的JSON格式在线检测网站：**http://www.kjson.com/**

## SHP原理图



## SHP树形层级结构



# 术语

**服务器端、服务器、路由器服务器**

均指在普通路由器或者定制路由器中添加智能家居服务程序（包括WebServer、C/S服务器端）而形成的整体，起着连接**终端**和**执行器**的作用。

**控制端、控制台、终端、智能终端**

指通过连接**服务器端**对家居设备进行控制的设备及运行在其上的程序，如智能手机、平板、PC等。

**执行器、设备、家居设备**

侧重指添加了支持本协议的控制器的传统家居设备。

**设备拓扑**

设备的层级结构

**控制链路**

指连接终端和服务器的通信线路。

**执行链路**

指连接服务器和执行器的通信线路。

**Item、原子项**

指对执行器进行原子拆分后的控制部件的抽象，如开关类型、单变量类型等。

**Item抽象**

指根据协议的**Item**对智能家居设备进行分解，并对分解后的原子部件进行抽象。

**Item集、Item集合**

指对智能家居设备进行**Item抽象**后形成的Item集合。

**Item树集**

有些非原子设备部件可能包括其他非原子设备部件和原子设备部件，由此可能形成Item的树结构的集合，简称Item树集。

# 详细协议

## 协议总览

### 终端收发协议

头：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MAGIC-Q | 包长（含包头） | MAGIC-B | content格式 | 保留 |
| ------1字节------ | ------2字节---- | ------1字节------ | ------1字节------ | ------1字节------ |

实际包：

|  |  |
| --- | --- |
| 头 | 实际内容 |
| ------sizeof(头) 6字节------ | --------------不定长------------ |

C语言描述的头结构：

|  |
| --- |
| 1. #pragma pack(push,1) //表示1字节对齐 2. #define MAGIC1 'Q' 3. #define MAGIC2 'B' 4. typedef struct HeadForSend 5. { 6. U8 magic1; 7. U16 pkg\_len; 8. U8 magic2; 9. U8 content\_type; 10. U8 reserve; 11. }HeadForSend,HeadForRecv; |

Content格式说明：

|  |  |  |
| --- | --- | --- |
| 格式名 | 值 | 备注 |
| Binary | 0x42 | 二进制格式，0x42==’B’ |
| Json | 0x4a | Json格式，0x4a==’J’ |
| Xml | 0x58 | XML格式，0x58==’X’ |
| Consultation [[1]](#footnote-1) | 0x43 | 协商类型，0x43==’C’ |

Consultation（协商）类型说明

1. 所有终端必须实现此种类型的协议。

此类型协议主要用来报告终端设备ID、名称、类型，方便集中管理。

1. 设备与服务器连接成功后，或者设备拓扑改变后，在服务器响应此类型协议之前，每隔一定时间发送一次此协议。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | |  |
| ------ 6字节------ | | --------------46字节------------ | | |  |
|  | |  |  |  |  |
| PID | ID | 设备类型 | 设备名称 | 协议版本 | 支持的内容协议 |
| -2Bytes—0001- | -48位MAC - | --2字节-- | ---32字节--- | ---1字节--- | ---3字节--- |

* **ID：**暂时使用自定义的随机48位MAC地址
* **设备类型：**待定
* **设备名称：**标示设备的名称
* **协议版本：**标示当前设备支持的协议的版本，高4 位表示主版本号，低4位表示此版本号。
* **支持的内容协议：**当前设备支持的Content协议类型，例如BJX、-JX、B—等组合。

1. 服务器收到Consultation类型数据包，向设备回复如下Consultation类型S0/No确认数据包。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 头 | 实际内容 | | | |  |
| ------ 6字节------ | --------------10字节------------ | | | |  |
|  |  |  | |  |  |
| PID | ID | | 服务器是否支持 | | 保留 |
| --2Bytes—0002-- | -48位MAC - | | ---2字节--- S0 or NO | | ---2字节--- |

1. 设备收到Consultation类型确认S0数据包，回复Consultation类型自我描述数据包，向服务器报告自身拓扑结构及功能部件。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 头 | PID | | 实际内容 | | |  |
| ------ 6字节------ | --2Bytes—0003-- | | -不定长的树状JSON格式- | | |  |
|  |  |  | |  | |  |
| PID ： --2Bytes—0003-- | | | | | |  | |
| { | | | | | | //start | |
| "NAME": "SmartHeater", | | | | | | //view name | |
| "MAC": "ZJU003", | | | | | | //48位mac地址-BCD码表示 | |
| "TYPE": "DEV", | | | | | | //DEV or SET or TERM | |
| "CHILD": [ | | | | | | //child array start | |
| { | | | | | |  | |
| "NAME": "Light", | | | | | |  | |
| "TYPE": "SET", | | | | | | //SET or TERM | |
| "CHILD": [ | | | | | |  | |
| { | | | | | |  | |
| "NAME": "ReadLight", | | | | | |  | |
| "TYPE": "TERM", | | | | | | //SET or TERM | |
| "DEV\_ID":0 | | | | | |  | |
| "DEVTYPE": "SWITCH" | | | | | |  | |
| }, | | | | | |  | |
| { | | | | | |  | |
| "NAME": "DecorateLight", | | | | | |  | |
| "TYPE": "TERM", | | | | | | //SET or TERM | |
| "DEV\_ID":1 | | | | | |  | |
| "DEVTYPE": "SWITCH" | | | | | |  | |
| } | | | | | |  | |
| ] | | | | | |  | |
| }, | | | | | |  | |
| { | | | | | |  | |
| "NAME": "CDPlayer", | | | | | |  | |
| "TYPE": "TERM", | | | | | | //SET or TERM | |
| "DEV\_ID":2 | | | | | |  | |
| "DEVTYPE": "SWITCH" | | | | | |  | |
| } | | | | | |  | |
| ] | | | | | | //child array end | |
| } | | | | | | //end | |

“TYPE”说明表：

|  |  |  |
| --- | --- | --- |
| TYPE | 值 | 备注 |
| DEV | 0 | 指当前节点是设备类型 |
| SET | 1 | 指当前节点是虚拟的设备集合 |
| TERM | 2 | 指当前节点是真实的执行控制器 |

1. 服务器收到Consultation类型自我描述数据包，回复Consultation类型S1/No确认数据包。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 头 | 实际内容 | | | |  |
| ------ 6字节------ | --------------10字节------------ | | | |  |
|  |  |  | |  |  |
| PID | ID | | 服务器是否支持 | | 保留 |
| --2Bytes—0004-- | -48位MAC - | | ---2字节--- S0 or NO | | ---2字节--- |

**协商协议之后，系统进入正式通信阶段。双方可以用各自都支持的协议进行通信。[[2]](#footnote-2)**

## 原子类型定义

智能控制系统都是由最终的执行器构成的，对这些执行器进行抽象定义，作为原子类型来组装智能控制系统，可以很好的提高系统的扩展性和兼容性。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | PID | 说明 | 实例 |
| SWITCH | 1001 | | 开关 | 普通电灯、插座、开关等 |
| T\_SENSOR | 1002 | | 温度传感器 | 18B20 |
| G\_SENSOR | 1003 | | 重力传感器 |  |
| VARIABLE | 1004 | | 单变量类型部件 | 变速电机、可变电阻 |
| KEYBOARD | 1005 | | 单字母输入部件 | 遥控器接收器 |
| CAMERA | 1006 | | 摄像头 |  |
| CUSTUM | \*\*\*\* | | 自定义 |  |

### 协议1001 for SWITCH

#### 协议1001二进制版本

控制器请求协议：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | | |  | |
| ------ 6字节------ | | ---------不定长--------- | | | |  | |
|  | |  |  | |  |  | | |
| PID | DEV\_ID | OP\_TYPE | | Path\_size | | 保留 | Path |
| --2字节-恒为1001- | --2字节-- | --1字节-- | | --2字节--- | | --2字节--- | - Path\_size - |

MAC DEV\_ID Path

1. MAC用来唯一标识一个设备。（在发送发现协议时向服务器报告，用于在设备重连时查询历史信息）。

2. DEV\_ID用来唯一标识一个设备里的执行部件，

看图

注：

DEV\_ID：在当前设备中须唯一

Path\_size：当前操作的路径长度，含结尾\0字符。当设备是执行器时，Path\_size可以为0。

Path：型如/smarthome/light1的路径。

OP\_TYPE：可以取值0关闭，1打开，2读取状态。详细见下表：

|  |  |  |  |
| --- | --- | --- | --- |
| OP\_TYPE | Binary取值 | Json取值 | Xml取值 |
| 打开 | 1 | OPEN | OPEN |
| 关闭 | 0 | CLOSE | CLOSE |
| 读取 | 2 | READ | READ |

当OP\_TYPE== READ时，执行器回复如下数据包：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | |  | |
| ------ 6字节------ | | ---------不定长--------- | | |  | |
|  | |  |  |  |  | | |
| PID | DEV\_ID | OP\_TYPE | | STATE | VISIT | 保留 |
| --2字节-恒为1001- | --2字节-- | --1字节恒为2-- | | --1字节-- | -1字节-- | ---2字节--- |

STATE可取值：1代表当前状态为打开的；0代表当前状态为关闭的

#### 协议1001 JSON版本

控制器请求协议：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 头 | 实际内容 | | | |  |
| ------ 6字节------ | --------------不定长的JSON格式------------ | | | |  |
|  |  |  |  | |  |
| { | | | | //start | |
| PID: 1001, | | | | //package id | |
| DEV\_ID:3 | | | | // destination id | |
| PATH:”/smarthome/light1”, | | | | //destination path | |
| OP\_TYPE:OPEN | | | | //operation type | |
| } | | | | //end | |

当OP\_TYPE== READ时，执行器回复如下数据包：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 头 | 实际内容 | | | |  |
| ------ 6字节------ | --------------不定长的JSON格式------------ | | | |  |
|  |  |  |  | |  |
| { | | | | //start | |
| PID: 1001, | | | | //package id | |
| DEV\_ID:3 | | | | // destination id | |
| VISIT:WR, | | | | //writeable and readable | |
| OP\_TYPE:READ | | | | //always READ | |
| STATE:0 | | | | //0 or 1 | |
| } | | | | //end | |

STATE可取值：1代表当前状态为打开的；0代表当前状态为关闭的

#### 协议1001 XML版本

### 协议1002 for T-Sensor

#### 协议1002二进制版本

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | | |  | | |
| ------ 6字节------ | | ---------不定长--------- | | | |  | | |
|  | |  |  | |  |  | | | |
| PID | DEV\_ID | OP\_TYPE | | Path\_size | | TEMP | 保留 | Path |
| -2字节-恒为1002 | --2字节-- | --1字节-- | | --2字节--- | | --2字节-- | --2字节- | - Path\_size - |

注：

DEV\_ID：在当前设备中须唯一

Path\_size：当前操作的路径长度，含结尾\0字符。当设备是执行器时，Path\_size可以为0。

Path：型如/smarthome/air\_temperature的路径。

OP\_TYPE： 2读取当前温度，精度0.1摄氏度；3写入期望达到的温度，精度0.1摄氏度。

|  |  |  |  |
| --- | --- | --- | --- |
| OP\_TYPE | Binary取值 | Json取值 | Xml取值 |
| 读取 | 2 | READ | READ |
| 写入 | 3 | WRITE | WRITE |

当OP\_TYPE== READ时，执行器回复如下数据包：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | | |  | |
| ------ 6字节------ | | ---------不定长--------- | | | |  | |
|  | |  |  | |  |  | | |
| PID | DEV\_ID | OP\_TYPE | | TEMP | | VISIT | 保留 |
| --2字节-恒为1001- | --2字节-- | --1字节==2-- | | --2字节-- | | -1字节-- | ---1字节--- |

TEMP单位为十分之一摄氏度。例如TEMP=300，代表当前温度为30.0摄氏度。

#### 协议1002 JSON版本

控制器请求协议：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 头 | 实际内容 | | | |  |
| ------ 6字节------ | --------------不定长的JSON格式------------ | | | |  |
|  |  |  |  | |  |
| { | | | | //start | |
| PID: 1002, | | | | //package id | |
| DEV\_ID:3 | | | | // destination id | |
| PATH:”/smarthome/light1”, | | | | //destination path | |
| OP\_TYPE: READ | | | | //operation type | |
| TEMP:350 | | | | // | |
| } | | | | //end | |

当OP\_TYPE== WRITE时，代表期望执行器达到的温度，执行器可以忽略；

当OP\_TYPE== READ时，执行器回复如下数据包：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 头 | 实际内容 | | | |  |
| ------ 6字节------ | --------------不定长的JSON格式------------ | | | |  |
|  |  |  |  | |  |
| { | | | | //start | |
| PID: 1002, | | | | //package id | |
| DEV\_ID:3 | | | | // destination id | |
| VISIT:WR, | | | | //writeable and readable | |
| OP\_TYPE:READ | | | | //always READ | |
| TEMP:350 | | | | //temperature | |
| } | | | | //end | |

#### 协议1002 XML版本

### 协议1004 for Variable

#### 协议1004 二进制版本

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | | |  | |
| ------ 6字节------ | | ---------不定长--------- | | | |  | |
|  | |  |  | |  |  | | |
| PID | DEV\_ID | OP\_TYPE | | Path\_size | | Value | Path |
| --2字节-恒为1004- | --2字节-- | --1字节-- | | --2字节--- | | ---4字节--- | - Path\_size - |

注：

DEV\_ID：在当前设备中须唯一

Path\_size：当前操作的路径长度，含结尾\0字符。当设备是执行器时，Path\_size可以为0。

Path：型如/smarthome/light\_brightness的路径。

OP\_TYPE： 2读取当前温度，精度0.1摄氏度；3写入期望达到的温度，精度0.1摄氏度。

|  |  |  |  |
| --- | --- | --- | --- |
| OP\_TYPE | Binary取值 | Json取值 | Xml取值 |
| 读取 | 2 | READ | READ |
| 写入 | 3 | WRITE | WRITE |

当OP\_TYPE== READ时，执行器回复如下数据包：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | | |  | |
| ------ 6字节------ | | ---------不定长--------- | | | |  | |
|  | |  |  | |  |  | | |
| PID | DEV\_ID | OP\_TYPE | | Value | | VISIT | 保留 |
| --2字节-恒为1004- | --2字节-- | --1字节==2-- | | ---4字节--- | | -1字节-- | ---2字节--- |

#### 协议1004 JSON版本

#### 协议1004 XML版本

### 协议1005 for KEYBOARD

#### 协议1005 二进制版本

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | | | |  | |
| ------ 6字节------ | | ---------不定长--------- | | | | |  | |
|  | |  |  | |  | |  | | |
| PID | DEV\_ID | OP\_TYPE | | Path\_size | | CHAR | 保留 | Path |
| 2字节-恒为1005 | --2字节-- | --1字节-- | | --2字节--- | | -1Byte- | -3Byte- | - Path\_size - |

注：

DEV\_ID：在当前设备中须唯一

Path\_size：当前操作的路径长度，含结尾\0字符。当设备是执行器时，Path\_size可以为0。

Path：型如/smarthome/light\_brightness的路径。

OP\_TYPE： 2读取当前温度，精度0.1摄氏度；3写入期望达到的温度，精度0.1摄氏度。

|  |  |  |  |
| --- | --- | --- | --- |
| OP\_TYPE | Binary取值 | Json取值 | Xml取值 |
| 读取 | 2 | READ | READ |
| 写入 | 3 | WRITE | WRITE |

当OP\_TYPE== READ时，执行器回复如下数据包：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 头 | | 实际内容 | | | |  | |
| ------ 6字节------ | | ---------不定长--------- | | | |  | |
|  | |  |  | |  |  | | |
| PID | DEV\_ID | OP\_TYPE | | CHAR | | VISIT | 保留 |
| --2字节-恒为1005- | --2字节-- | --1字节==2-- | | ---1字节--- | | -1字节-- | ---1字节--- |

#### 协议1005 JSON版本

#### 协议1005 XML版本

## 状态报告/请求协议

### 设备状态报告协议、心跳包协议

中央服务器维护系统的总体状态树，设备定时向其连接的中央控制服务器发送心跳包。设备向中央服务器发送的数据总体上来说是一个链表结构。包含原子执行体的id和状态信息。

设备状态报告协议、心跳包协议的作用有如下几点：

1. 给中央服务器提供自身状态，使控制端可以随时向服务器查询到当前状态。
2. 当有设备故障时，中央控制器能够及时发现。
3. 减小系统实现难度，编程过程中，可以尽量少的考虑目标机的系统状态。

#### 心跳包协议二进制版本

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 头 | | | 实际内容 | | | |  | | | |
| ------ 6字节------ | | | ---------不定长--------- | | | |  | | | |
|  | | |  |  | |  | |  | | | |
| PID+MAC | NUM | - DEV\_ID1 | offset | | DATA1 | | DEV\_ID2 | | Len | DATA2 |
| ---8Bytes-- | -2Byte- | --2字节- | --2字节-- | | --offset-- | | --2字节-- | | -2Byte- | len -4 |
|  |  |  |  | |  | |  | |  |  |
| 2Byts–恒为9999 | 6Bytes MAC | | | | | | | | | |

#### 心跳包协议JSON版本

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 头 | 实际内容 | | |  | |
| ------ 6字节------ | ---------不定长--------- | | |  | |
|  |  |  |  |  | | |
| {  “PID”：9999  “MAC”：”ZJU001”  “TICK”：[  {  “DEV\_ID”:0  “DATA”:OPENED/CLOSED  },{  “DEV\_ID”:1  “DATA”:128  }  ]  } | | | | |

**README:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Copyright Statement:

\* Copyright (c) Q.Bryan <qbing2010@163.com> All rights reserved.

\*

\*-----------------------------------------------------------------------------

---------------------------------------

Smart Home Protocl

-----------------

MMMMMMMMMM MMMMMMMMMMMM MMMMMMMMMMMM MMMMMMMMMMMMMMMM

MMMM MMMM MMMMMM MMMMMM MMMMMM MMMMMM

MMMM MM MMMM MMMM MMMM MMMM

MMMM MM MMMM MMMM MMMM MMMM

MMMMMM MMMM MMMM MMMM MMMM

MMMMMMMM MMMM MMMM MMMM MMMMMM

MMMMMMMM MMMMMMMMMMMMMMMMMMMMMM MMMMMMMMMMMM

MMMMMM MMMM MMMM MMMM

MMMMMM MMMM MMMM MMMM

MM MMMM MMMM MMMM MMMM

MM MMMM MMMM MMMM MMMM

MMMM MMMM MMMMMM MMMMMM MMMMMM

MMMMMMMM MMMMMMMMMMMM MMMMMMMMMMMM MMMMMMMMMMMM

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Author: Q.Bryan 2013/09/17 17:36:41

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

简要说明

1、 系统包含Sever、Webserver、Web页、模拟器、终端执行器、手机端控制器。

目前有些只完成了一部分。

2、 第三方自己的设备兼容我们的协议可以参考SHP\_example进行。API部分的接口有待进一步完善。

3、 大体结构图可以参看《智能家居-草稿1》

4、 进一步抽象出模块后，大家合作编程。当前只是初步打通流程。

调试方法简介：

1、到SmartHomeServer目录启动sh\_server.exe程序。<如果无此程序，可通过make命令编译>。

2、到SmartHomeBoa目录下，执行make，make install ，以SmartHomeWeb为http根目录配置好WEB Server。之后可通过boa -d调试运行WEB Server。

3、到MODIS目录下，启动模拟器SmartHomeModis.exe

4、WEB server配置正确的情况下，通过127.0.0.1可以看到根据设备的连接情况生成的控制界面。右键可操作设备。

**设备端API库：**

typedef SHP\_RESULT\_T (\*SHP\_EXE\_T)(DEV\_ID dev\_id,OP\_TYPE op\_type,void \*data);

typedef SHP\_RESULT\_T (\*SHP\_GET\_T)(DEV\_ID dev\_id,void \*out\_buf,U16\* max\_size\_AND\_out\_size);

typedef struct SHP\_SYSTEM

{

//////////////////SHP System Value//////////////////////begin///////////////

SHP\_STATUS\_T state;

SHP\_RECV\_DATA\_T buf;

//////////////////SHP System Value//////////////////////end////////////////

//////The function which should be implimente at other place/////////begin///////////

S16 (\*send\_data)(U8\* bin,U16 len);

S16 (\*send\_system\_mac)(void);

S16 (\*send\_system\_topology)(void);

////The function which should be implimente at other place/////////begin///////////

}SHP\_SYSTEM;

//////////////The function which should be called at other place/////////begin///////////

//called before use any SHP function

S16 shp\_system\_init(S16 (\*send\_data)(U8\* bin,U16 len),S16 (\*send\_system\_mac)(void), S16 (\*send\_system\_topology)(void));

//Register your device operation after called init func

SHP\_RESULT\_T shp\_register\_execution(DEV\_ID dev\_id,DEV\_TYPE dev\_type,SHP\_EXE\_T exe,SHP\_GET\_T get);

SHP\_RESULT\_T shp\_register\_groud\_dev\_execution(DEV\_ID\* dev\_ids,U16 dev\_nums,SHP\_EXE\_T exe,SHP\_GET\_T get);

//typicly be called per second or 100ms

S16 shp\_system\_tick(void);

//You MUST call this fuction when data arrived!!!!!!!

S16 shp\_handle\_recviced\_data(U8\* bin,U16 len);

//////////////The function which should be called at other place/////////begin///////////

1. 与上面三种格式是不同逻辑的概念，放到此处是为了方便实现 [↑](#footnote-ref-1)
2. **在正式通信阶段，设备向服务器发送心跳包（状态包），表示正常工作和报告状态** [↑](#footnote-ref-2)