名词解释：  
**海计算机运行时系统**：Things as a computer（tOS），物端开放平台，可以部署到家居环境下的任何一台设备上，可以是电脑，手机，或者具有操作系统的路由器上。

**网关**：gateway，通常指接入节点，用来组网，负责转发子设备和tOS的协议数据，也可接入internet。

**设备**：智能家居设备，iOS设备，android设备。

**命令**：command，tOS对本地设备信息提出的控制要求，例如tOS下发命令要求重启设备。（命令又分为两种格式：一种是面向命令类型设备，即只需要简单的控制命令，如开，关。另外一种是命令数据型设备，即需要命令的同时还需要一些数据，比如对于温度控制器，命令为温度升高，数据为3，即表示命令温度控制器将温度升高3度。）

**事件**：event，本地设备提醒tOS需要关注的警示信息，或者是信息处理后反馈信息。例如温度控制器上报的高温异常事件，或者是温度控制器收到打开命令后反馈给tOS的ACK。

**状态**：status，当前设备的描述参数。例如温度控制器当前的温度值。

**数据包内容**

**数据包格式：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标识位 | 消息头 | 消息体 | 校验码 | 标识位 |

**消息头格式**：

|  |  |  |  |
| --- | --- | --- | --- |
| 起始字节 | 字段 | 数据类型 | 描述与要求 |
| 0 | 消息ID | WORD | 见下表。 |
| 2 | 消息类型 | WORD | 数据包类型，command或者event。 |
| 4 | 消息长度 | WORD | 表示数据包携带的消息体长度。 |
| 6 | 设备ID | WORD | 消息所关联的设备ID。 |
| 8 | 数据包序号 | Byte[n] | 数据包序号，用于重传。 |

**平台通用消息类型**

1. 平台通用事件：0X0001（事件是设备主动向平台发送消息）

|  |  |  |
| --- | --- | --- |
| 事件描述 | 消息ID | 消息体 |
| 设备上线事件 | 65535 | 时间戳，记录设备上线时间 |
| 设备下线事件 | 65534 | 时间戳，记录设备下线时间 |
| 命令反馈事件 | 65533 | 命令执行结果 |
| 故障反馈事件 | 65532 | 故障描述 |
| 警示提醒事件 | 65531 | 设备异常描述 |

1. 平台通用命令：0X0002（命令是平台向设备发送控制命令）

|  |  |  |
| --- | --- | --- |
| 命令描述 | 消息ID | 消息体 |
| 软重启（restart） | 65535 | 空 |
| 事件送达反馈 | 65534 | 空 |
| 恢复原厂设置（reset） | 65533 | 空 |
| 时钟同步 | 65532 | 时间戳 |
| 通知 | 65531 | 前一个字节代表控制命令码，比如温度升降，开关等；后一个字节对于命令型设备为0X00，对于命令数据型设备，如果需要则为具体数值。比如对于温度控制器，开温度控制器消息体为0X0000，将温度升高3度，则消息体为0X0203 |
| 实时数据查询 | 65530 | 平台要查询的设备状态属性ID（例如温度计，开关状态属性ID：0X10，当前温度属性ID：0X20） |

**设备状态属性命令码**

1.照明设备系统控制码（通知命令中使用的控制码来控制设备的）

|  |  |
| --- | --- |
| 命令 | 含义 |
| 0x10 | 开灯 |
| 0x11 | 关灯 |
| 0x20 | 调亮 |
| 0x21 | 调暗 |
| 0x30 | 灯光颜色变换 |

照明设备系统状态命令码

|  |  |
| --- | --- |
| 状态命令码 | 含义 |
| 0x10 | 开关状态 0x11：开 0x10：关 |
| 0x20 | 明亮程度 0x20-0x2F：可定义16种明亮程度，依次从暗到亮 |
| 0x30 | 灯光颜色 0x30-0x3F：可定义16种颜色变换 |

tOS发送相应的状态命令码来实时查询设备当前属性数据。

例：照明设备灯光颜色状态

|  |  |
| --- | --- |
| 状态码 | 含义 |
| 0x30 | 白色 |
| 0x31 | 黄色 |
| 0x32 | 蓝色 |
| 0x33 | 红色 |
| 0x34 | 绿色 |

当tOS发送命令要求查询设备颜色状态时，设备返回相应的颜色状态码。

2.温控设备系统命令码

|  |  |
| --- | --- |
| 命令 | 含义 |
| 0x10 | 开启 |
| 0x11 | 关闭 |
| 0x20 | 制冷升温 |
| 0x21 | 制冷降温 |
| 0x30 | 制热升温 |
| 0x31 | 制热降温 |
| 0x40 | 换气 |
| 0x50 | 高风 |
| 0x51 | 中风 |
| 0x52 | 低风 |
| 0x60 | 自动 |

温控设备系统状态命令码

|  |  |
| --- | --- |
| 状态命令码 | 含义 |
| 0x10 | 开关状态 0x11：开 0x10：关 |
| 0x20 | 温度状态 0x20-0x2F：可表示16个档次的温度，15度-30度 |
| 0x50 | 风速状态 0x50-0x5F：可表示16个档次的风速。 |

3.播放器设备系统命令码

|  |  |
| --- | --- |
| 命令 | 含义 |
| 0x10 | 开启播放器 |
| 0x11 | 关闭播放器 |
| 0x20 | 音量+ |
| 0x21 | 音量- |
| 0x30 | 快进>> |
| 0x31 | 快退<< |
| 0x40 | 下一个> |

播放器设备系统状态命令码

|  |  |
| --- | --- |
| 状态命令码 | 含义 |
| 0x10 | 开关状态 0x11：开 0x10：关 |
| 0x20 | 当前音量 |
| 0x30 | 当前播放内容 |

4.简单命令型设备系统命令码

|  |  |
| --- | --- |
| 命令 | 含义 |
| 0x10 | 开启 |
| 0x11 | 关闭 |

简单命令型设备指只具备智能开启关闭简单功能的设备，如智能扫地机器人，智能窗帘，智能摄像头等，只需要开启或者关闭，这类设备就可以自己完成工作，不需要其他的设定。其他类型设备，具体设备可根据具体功能自定义命令码。

**各功能系统设备ID**

|  |  |
| --- | --- |
| 设备ID | 功能子系统 |
| 0xAFF0 | 照明系统设备 |
| 0xBFF0 | 温控系统设备 |
| 0xCFF0 | 播放器系统设备 |
| 0xDFF0 | 简单命令型设备 |
| 0xXFF0 | 可扩展设备 |

照明系统设备ID

|  |  |
| --- | --- |
| 设备ID | 照明系统设备 |
| 0xAFF0 | 客厅灯 |
| 0xAFF1 | 主卧灯 |
| 0xAFF2 | 次卧灯 |
| 0xAFF3 | 客卧灯 |
| 0xAFF4 | 卫生间灯 |
| 0xAFF5 | 餐厅灯 |
| 0xAFF6 | 阳台灯 |
| 0xAFF7 | 厨房灯 |
| 0xAFFX | 可扩展 |

由于很多智能设备所用的系统资源都是受限的，所以在与这些嵌入式设备做数据交流时，选择使用本方案提出的数据协议，用二进制的方式传输数据。

0XFF - FFFB- 0002- 0002- AFF0- 0000-校验码-FFFF

开始位-消息ID（命令通知，65531）-命令型数据包-消息体长度2字节-设备ID（客厅灯）-开灯-校验码-结束位

{

msgHead:{

msgID:”65531”;

msgType:”0x0002”;

msgLength:”2”;

deviceID:”0XAFFF0”;

}

msgBody:{

status:”0000”;

}

}

**设备联动消息体：**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 数据类型 | 长度 | 描述及要求 |
| 策略ID | byte[n] | 16 | 策略的唯一标识 |
| 启动状态 | byte[n] | 1 | 0-关闭，1-启用 |
| 触发条件列表 | String | n | conditionList，见下表 |
| 触发条件关系 | byte[n] | 1 | conditionRelation(or,and)多个触发条件可以是或的关系，也可以是与的关系 |
| 设备动作集合 | String | n | actionList,满足触发条件后需要执行的动作，见下表。 |

触发条件

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 数据类型 | 长度 | 描述 |
| 设备ID | Byte[n] | 16 | 联动的设备ID，deviceID |
| 状态命令码 | Byte[n] | 8 | statusCode,读取指定设备的状态值 |
| 运算符 | Byte[n] | 1 | operation,</>/= |
| 比较值 | Byte[n] | 16 | compareValue,与读取到的状态值比较的参数值。 |

设备动作集合

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 数据类型 | 长度 | 描述 |
| 设备ID | Byte[n] | 16 | 联动的设备ID，deviceID |
| 控制命令码 | Byte[n] | 8 | commandCode，满足触发条件后对联动设备作出的控制动作。 |

设备联动Json示例如下

设备联动消息体={

strategyID:”sid\_01”;

strategyState:”1”;

conditionList:{

deviceID:”0xDFF0”;

queryStatusCode:”0x10”;

operation:”=”;

compareValue:”0x11”

}

conditionRelation:”0”//如果只有一个条件，则不用关注这个选项。

actionList:{

deviceID:”0xAFF0”；

commandCode:”0x10”；

}

}

自定义情景模式：

用户可以根据需要对家中的某些或者所有智能设备进行设置，产生特定的情景模式，当选择该情景后，只需一键，就可以将家居环境中的设备全部设定为相应的状态。也可以将所有设备都设置为自动状态，则家中的设备，会根据当前的环境状况，做相应的调节。

系统实现方面：可以将用户所做的设置进行保存，然后当用户点击按钮需要应用该模式时，则向情景中的所有设备发送控制命令。

智能电饭锅的一个应用场景：买回来的一个半成品饭菜，放在电饭锅中，手机扫描饭菜包装上的二维码，电饭锅就能自动进入相应的烹饪模式。

业务场景主要包括三个方面：

1. 智能控制 2.情景模式 3.设备联动

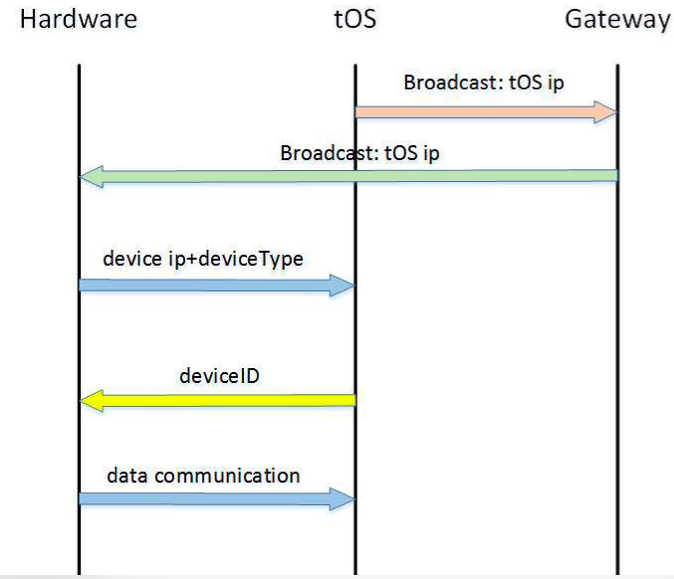
**平台接入设计**

注意：开始所有流程前，运行时系统需要开启，家居环境下的所有设备都要通过wifi连接到同一个局域网中。

设备接入采用的方案是：在设备内部嵌入联通协议，直接连接到家庭网关。这样做的好处就是，不用设计并购买专门的家庭网关控制中心，只要设备开发者按照tOS系统设计的接入流程及数据格式发送消息，即可被系统识别并进行注册。

设备接入流程（注册流程）：

当运行时系统启动之后，就定期向局域网广播发送数据包，告诉连接到局域网里的设备tOS系统所在设备的ip地址。局域网内的设备收到这个数据包之后，将数据包内容进行保存，然后向这个地址发送心跳包，告诉tOS系统自己的设备类型以及IP地址进行注册。tOS系统收到消息后存储相应信息，并为该设备分配唯一的deviceID，并告知该设备。此后所有的数据交流，都通过设备ID进行唯一性识别。



设备类型（device type）

**enum** DeviceType {

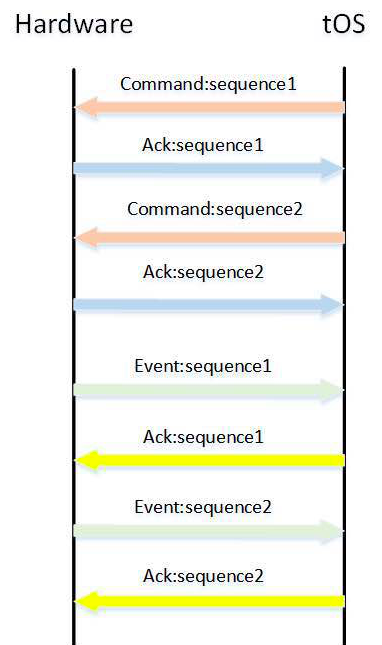
***LIGHT\_CTL***("LIGHT\_CTL",1),***TEMPERATURE\_CTL***("TEMPERATURE\_CTL",2),***PLAYER\_CTL***("PLAYER\_CTL",3)……}

除了这三种设备类型之外，可以运行时动态增加枚举类型，以动态增加设备类型。讲解说明：当一个照明设备要加入到当前系统，则在该设备收到系统广播的ip地址之后，就会向系统发送自己的ip地址以及设备类型“LIGHT\_CTL”或者是“1”，告诉系统自己是一个照明设备。然后系统根据设备类型为这个设备分配一个唯一的设备ID，假如这个照明设备是第一个加入系统的照明设备，那么系统从0xAFF0字段开始分配，由于是第一个，所以就给这个设备分配为0xAFF0，作为这个设备的ID。如果是第二个加入的，则系统顺次分配0xAFF1，以此类推。系统会将分配给设备的这个ID告诉设备，此后的数据交流都将用这个设备ID，唯一指定。系统会对注册的设备进行记录，建立设备类型，设备ID以及设备IP一一对应的关系列表。

设备登录：

首先检查设备本地是否已存储设备ID。若已有设备ID，则直接使用设备ID进行数据交流。如果没有则进入注册流程。

设备和tOS系统之间信息的传递



数据通过局域网TCP/IP协议传输，目前仅适用了crc校验确定数据的完整，采用sequence、ack的方式保证数据的可靠到达。如上图所示，当发送方发送消息后，接收方需要回复ack包进行确认，要求回复的sequence与收到的一致。如果发送方没有收到ack，则重发数据包，尝试3次后不再重传。命令，事件拥有独立的sequence,每发一个包递增1，重传数据包时sequence不变。对于设备发送的提示警告消息，若达到重传次数后仍然没有得到回复，则应该对其保存。在下次发送消息之前，先发送保存的数据包。如果是数据通讯链路断开，设备也应该先保存数据包，待数据通讯链路恢复之后，立即发送此数据包。

设备与系统之间连接的维持

在建立连接之后，设备应定期向系统发送心跳包，系统收到后，向设备发送应答消息。系统通过心跳包的接收，来判断设备是否在线。

为了便于区分管理，平台为每种类型的功能子系统的设备ID都分配了特定的字段。如下表所示。平台给出了几个基本分类，若要增加子系统类别可继续扩展。

|  |  |
| --- | --- |
| 设备ID | 功能子系统 |
| 0xAFF0 | 照明系统设备 |
| 0xBFF0 | 温控系统设备 |
| 0xCFF0 | 播放器系统设备 |
| 0xDFF0 | 简单命令型设备 |
| 0xXFF0 | 可扩展设备 |

每个功能子系统内部具体包含的设备的设备ID如下所示。以照明系统设备ID为例。

|  |  |
| --- | --- |
| 设备ID | 照明系统设备 |
| 0xAFF0 | 客厅灯 |
| 0xAFF1 | 主卧灯 |
| 0xAFF2 | 次卧灯 |
| 0xAFF3 | 客卧灯 |
| 0xAFF4 | 卫生间灯 |
| 0xAFF5 | 餐厅灯 |
| 0xAFF6 | 阳台灯 |
| 0xAFF7 | 厨房灯 |
| 0xAFFX | 可扩展 |