一种无线自组网通信协议设计

张远

**摘**要：本文是基于ZigBee自组网协议设计的一种新型的无线自组网通信协议，该通信协议简单明了、配置灵活、方便测试，可以在各种环境下传输信息。该协议创建了一种蜂窝状网络，实现点对点的信息传输，经过实践操作，测试得其运行稳定，模块化好，达到了设计的目的。

**关键词：**ZigBee，蜂窝状网络，自组网，路由表，退避算法

1. **协议的目的及意义**

近年来，ZigBee自组网协议广泛应用于物联网无线数据终端，本协议基于此设计一种类ZigBee的新型自组网协议，使其更好的贴合日常生活的使用，实现了学生宿舍的信息高效传输，整体结构清晰明了节点安装方便。可以根据实际需求扩展节点的功能，适用于不同环境，在学生宿舍方面的应用与推广有较好的前景。

**二.总体设计**

**系统方案**

系统整体框架图如下图1所示，

设 备 中 继 终 端

本系统基于物联网工程，使用类ZigBee的方式自建组网实现上下行的数据传输。整个系统自下而上结构为设备，中继，终端三层结构。

**通信协议设计**

数据传输由十三位字节组成，其中第一位和第十三位作为起始和终止的标志位，定义起始位为0xF1表示正常上行，0xF2表示正常下行，0xF3表示组网请求上行，0xF4表示组网请求下行，0xF5表示测试指令，对应的终止位为其起始位的最后一位取反，分别为0xFE、0XFD、0xFC、0xFB、0xFA；第二位第三位共同表示信号发出的设备标识号，第四位第五位共同表示数据最初来源的设备表示号；第六位表示操作数；第七位是一个计数位；第八、九、十、十一、十二合表示中继ID，这五位初始值为F6，自组网成功后其从左至右的顺序表示其在系统中传输数据的路径。

每次正常上行工作可由设备发出十三位十六进制的数据，一级中继接收到这个数据后首先会判断这是否是上行消息且是否为自己处理，如果是则进行修改再重新发出，否则自动忽略这条信息，二级中继接收到消息后所做出的事和一级中继是一样的，判断改编后再发出，最后是终端控制机，接收到数据判断后将信息写入数据库中，然后根据数据库内容显示在专门的网页界面上，使用者便可看到设备进行了什么操作变化。

正常下行数据传输时，首先使用者可通过网页中的按键选项对设备进行一种操作，此时这种携带着操作指令的九位十六进制的消息将从终端控制机这一级传至二级中继。二级中继判断其是否为下行消息且是否为自己处理，如果是，则对这段数据进行改编再传至下一级，否则自动忽略这条消息，一级中继接收到这个消息后也和二级中继做一样的判断并处理数据，传输到设备这一级时，设备根据发送来的指令做出相应的动作并根据需求返回数据。

**稳定性实现**

为了保障数据传输的稳定性，我们采取了退避算法，当数据传向下一级后，若下一级收到这个信息，自动向上一级发出确认收到的回复，上一级在设定时间内收到特定的回复，则停止发送消息，若超出设定时间未收到特定回复，此时表示下一级未收到消息，上一级也将停止发送消息，在0-1024之中选择任意时间退避等待，再重新发送消息，等待回复，若仍未收到回复，继续退避等待，以此类推。

**三.具体实施方案**

**自组网实例**

当一个中继（假设ID为‘0’‘1’）要自组网时，此时操作数应为0xF3即发送组网请求，计数位为初始值0x01，他将发送消息：

0xF3 0x30 0x31 0x30 0x31 0xF3 0x01 0xF6 0xF6 0xF6 0xF6 0xF6 0xFC

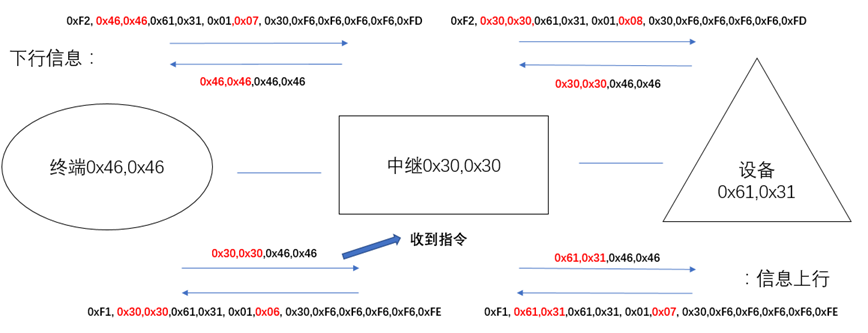
当终端收到这个组网消息时，初始位与终止位满足对应关系是有效数据，同时返回给中继收到消息的通知：0x30 0x31 0x46 0x46，中继收到此消息时发现是自己的ID并以0x46 0x46结尾，则停止发送组网消息，终端通过0xF3发现是申请加入组网的请求，判别计数位为0x01初始值则此中继未加入组网，终端将其中继ID的第二位0x31写入数据第八位即路由表的第二位，将计数位改为0x06，再把这个传输路径写入数据库，同时终端将给中继返回组网数据，操作数0xF6表示组网成功，下行数据计数位加1，则消息为：

0xF4 0x46 0x46 0x30 0x31 0xF6 0x07 0x31 0xF6 0xF6 0xF6 0xF6 0xFB

当中继收到这个组网消息时，初始位与终止位满足对应关系是有效数据，同时返回给中继收到消息的通知：0xFF 0xFF 0x46 0x46，终端收到此消息时发现是自己的ID并以0x46 0x46结尾，则停止发送组网消息，中继通过0xF6发现是组网成功的请求，判断计数位的值与其指向数据数组的值相等，则将路由表保留到自己的数据里，以后发送消息的七到十二位为：

0x07 0x31 0xF6 0xF6 0xF6 0xF6

以上即为中继与终端上行自组网的全部数据传输过程。相似的可以得到设备，中继，终端三者的组网过程，如下图所示。



**数据传输实例**

现假设已经存在了一条数据传输网络终端（id为’F’‘F’）、中继（ID为‘0’‘0’）和设备（ID为‘a’‘1’）。

首先具体描述一般信息下行，即信息由设备产生发出经由中继到终端的过程。

首先是设备0x61,0x31向中继发送指令：当起始位为0xF1时，设备请求上行，请求由‘a’‘1’发出，则第二、三位按其编号顺序分别为0x61,0x31；原始请求由设备提出，则第四、五位分别对应设备编号的两位：0x61,0x31；第六位为指令位，这里假设为0x01；第七位指针位的P本为8，上行左移一位变为7；第8、9、10、11、12位分别为已组网成功的目标路由表，即0x30,0xF6,0xF6,0xF6,0xF6；终止位为0xFE。

则该设备发出来的整条数据串即为：0xF1,0x61,0x31，0x61,0x31,0x01,0x07,0x30,0xF6,0xF6,0xF6,0xF6,0xFE

数据发送出去后，设备会在一个时间间隔内等待直到收到0x61,0x31,0x46,0x46的下级（即中继0x30,0x30）回复，否则重发，或者重新申请组网。

中继0x30,0x30收到上条数据串时，通过识别起始位为0xF1，和指针位所指向的a[7]路由表第一位0x30,为自己，然后会进向相关处理，即向上级回复收到指令和向下级发送指令。具体相关处理为：（1）将第二、三位0x61,0x31添加在0x46,0x46前，将其发出，即为回复收到指令。（2）将第二、三位0x61,0x31替换为自己的设备号0x30,0x30，再将指针位减一变为6，然后将其发送出去，即为向下级发送指令。

则该设备发出来的整条数据串即为：

0xF1,0x30,0x30,0x61,0x31,0x01,0x06,0x30,0xF6,0xF6,0xF6,0xF6,0xFE。

数据发送出去后，设备会在一个时间间隔内等待直到收到0x30,0x30,0x46,0x46的下级（即终端0x46,0x46）回复，否则重发，或者重新申请组网。

终端0x46,0x46收到上条数据串时，通过识别起始位为0xF1，和指针位P为0x06,为自己要处理的信息，然后会进向相关处理，和向上级回复收到指令0x30,0x30,x46,0x46。

以上即为一般上行全部数据传输过程。相似的可以得到一般的信息下行过程，如图中所示。

