**电力线监测系统**

1：系统架构

系统的物理线路连接如下图所示：



采集终端仅仅作为一个485总线协议与电力载波/zigbee协议之间的一个转换，所以整个系统的逻辑框图可以理解为下图：



系统采用多层次的方式。

1. 采集终端

采集终端作为一个设备节点，作为485总线协议与zigbee/电力载波协议之间的转换设备。

在485总线上挂载的设备有：

* 温度、水传感器
  + 4路温度传感器，能够返回锁存值和当前值
  + 1路水浸传感器，能够返回锁存值和当前值
  + 1路湿度传感器，能够返回锁存制和当前值
* 控制外围断路器的开关量信号以及开关量状态反馈
* 多功能仪表的读写端口
* 外部modBus设备的读写端口（暂时认为多功能仪表均为485总线+ModBus协议）

所有设备均为支持485读写的设备。

1. 中继器

中继器作为子母线与主母线之间的数据桥梁。

每个中继器均有唯一的ID编号。

与采集器通过电力载波或者zigbee通信。

通过电力载波方式与集中器进行通信

中继器在上电后持续轮询所有采集终端的温湿度信息并缓存

1. 集中器

集中器作为一个楼宇或者一个区域的数据转发设备。

通过电力载波方式与中继器进行通信

通过局域网与中心服务器进行通信

集中器在进入局域网后应该被分配一个固定的IP地址。

1. 中心服务器

保存所有的终端数据

1. 管理终端

实时查看和分析终端的当前状态以及历史趋势

**PS：下述文字除非特殊说明，否则温湿度的描述实际包含温度、湿度、水浸信息**

本协议为中心服务器与电力线监测系统的集中器之间的网络交互协议。

**网络端口：**

中继器开放5001号端口，开启TCP 服务。

基本包格式：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 包头 | 数据包类型 | 数据长度 | 数据 | 校验 |
| 3 Bytes | 1 Byte | 2 Bytes | N Bytes | 1 Byte |

服务器下发包头固定为字符串“TY$”，设备返回的包头固定为字符“TB$”

数据包类型表示当前的数据包类型

校验位为数据部分的异或校验结果值

数据协议：

综上，设备的指令协议包括了两个部分

1. 与中继器的指令（该指令实际是由集中器来判断下发至哪个ID的中继器）

|  |  |  |  |
| --- | --- | --- | --- |
| 指令名称 | 指令编号 | 返回编号 | 备注 |
| 采集终端搜索指令 | [0x10](#C_10) | [0x50](#C_50) | 中继器搜索温湿度检测设备的编号，并反馈搜索结果 |
| 温湿度信息反馈 | [0x11](#C_11) | [0x51](#C_51) | 中继器反馈自动轮询出来的所有设备温湿度信息 |
| 温湿度锁存指令 | [0x12](#C_12) | [0x52](#C_52) | 中继器对当前管理的所有采集终端下发要求锁存温湿度的指令 |
| 温湿度报警阈值设置 | [0x13](#C_13) | [0x53](#C_53) | 设置集中器下所有采集终端的温湿度报警阈值 |
|  |  |  |  |

1. 与中继器的指令

|  |  |  |  |
| --- | --- | --- | --- |
| 指令名称 | 指令编号 | 返回编号 | 备注 |
| 温湿度获取指令 | [0x31](#C_31) | [0x71](#C_71) | 获取中继器下某个采集终端下的温湿度信息 |
| 485 总线操作指令 | [0x32](#C_32) | [0x72](#C_72) | 通过485接口发送字符串  返回485接口读取到的返回结果串 |

三、主动报警协议

|  |  |  |
| --- | --- | --- |
| 指令名称 | 指令编号 | 备注 |
| 温湿度报警 | [0x91](#C_91) | 温湿度超过阈值时报警 |
| 状态变更报警 | [0x92](#C_92) | 设备离线 |
| 其它报警 | [0x93](#C_93) | 根据具体的设备定义协议内容：  断路器报警  遥调开关手动切换状态报警  等等 |

0x10

中继器节点搜索

服务器->中继器

用于要求某个集中器对管理的温湿度设备进行一次有效性扫描。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TY$ | 0x10 | 001 | 0 |  | 0x00 |

0x50

设备搜索结果反馈

中继器->服务器

上报当前所有的设备有效性状态

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TB$ | 0x50 | 001 | N\*(struct size) |  |  |

数据字段按照如下机构进行数据组织：

Struct inod\_info{

Uint8 inodID; 节点的ID

Uint8 useable; 节点有效性

}inod\_list[N]

0x11

节点温湿度数据查询

服务器->中继器

服务器向中继器要求获取所有有效节点的温湿度数据。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TY$ | 0x11 | 001 | 0 |  | 0x00 |

0x51

节点温湿度反馈结果

中继器->服务器

用于中继器向服务器上报当前所有有效节点的温湿度数据。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TB$ | 0x51 | 001 | Size of coll\_info |  |  |

数据字段按照如下机构进行数据组织：

Struct inod\_tem\_info{

Uint8 flag; //固定为0x55 用于做数据是否错乱的判断依据

Uint8 inodID; //节点ID

Uint8 humidity; //节点湿度 这里需要讨论一下湿度的精度要求，如果为0.1以下，则需要扩大字节数

Uint8 water; //水浸传感器状态

Uint8 temp[4]; //4个温度采样节点的值 这里需要讨论一下温度的精度要求，如果为0.1以下，则需要扩大字节数

Uint8 humidity\_locked; //锁存的数据

Uint8 water\_locked; //锁存的数据

Uint8 temp\_locked[4];

}inod\_list[N]

0x12

温度锁存指令

服务器->中继器

要求中继器下发温度采集终端的锁存指令，收到该指令后，中继器广播下发温度锁存指令，所有的温湿度采集点锁存当前状态数据。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TY$ | 0x12 | 001 | 0 |  |  |

0x52

锁存指令收发状态反馈

中继器->服务器

中继器确认收到指令

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TB$ | 0x52 | 001 | 0 |  |  |

0x13

温度报警阈值

服务器->中继器

服务器设置温湿度报警的阈值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TY$ | 0x13 | 001 | 0 |  |  |

Struct inod\_tem\_info{

Uint8 water\_enable; //开启水浸报警

Uint8 humidity\_enable; //开启节点湿度报警

Uint8 humidity; //节点湿度报警阈值 0为保持不变

Uint8 temp\_enable; //开启温度报警

Uint8 temp; //温度报警阈值 0为保持不变

}

0x53

温度报警阈值设置反馈

中继器->服务器

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TY$ | 0x53 | 001 | 0 |  |  |

**与中继器指令：**

0x31

温湿度获取指令

服务器->中继器

获取中继器下某节点的设备温湿度信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 节点ID | 数据 | 校验 |
| TY$ | 0x31 | 001 | 0 |  |  |

0x71

节点温湿度返回值

中继器->服务器

返回所查询节点的温湿度信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TB$ | 0x71 | 001 | 0 |  |  |

Struct inod\_tem\_info{

Uint8 flag; //固定为0x55 用于做数据是否错乱的判断依据

Uint8 inodID; //节点ID

Uint8 humidity; //节点湿度 这里需要讨论一下湿度的精度要求，如果为0.1以下，则需要扩大字节数

Uint8 water; //水浸传感器状态

Uint8 temp[4]; //4个温度采样节点的值 这里需要讨论一下温度的精度要求，如果为0.1以下，则需要扩大字节数

Uint8 humidity\_locked; //锁存的数据

Uint8 water\_locked; //锁存的数据

Uint8 temp\_locked[4];

}inod\_info

0x32

485总线操作指令

服务器->中继器

向某个中继器下的485总线发送字符串

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据 | 校验 |
| TY$ | 0x31 | 001 |  |  |

0x72

485总线发返回结果

中继器->服务器

返回485总线的操作结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 命令编号 | 中继器ID | 数据长度 | 数据 | 校验 |
| TB$ | 0x72 | 001 | 0 |  |  |

对于多功能仪表，遥调开关，智能断路器等的操作，均由上层软件封装485操作协议来收发、解析。

**报警操作协议：**

0x91

温湿度报警

中继器->服务器

数据指令：0x91

某个节点出现了温湿度的异常状况。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 包头 | 数据包类型 | 中继器ID | 设备ID | 数据 | 校验 |
| TB$ | 0x91 | 001 |  |  | 0x00 |

反馈信息内容同 Struct inod\_tem\_info

0x92

节点离线报警

中继器->服务器

当节点下的485操作设备离线(无反馈)时报警

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 包头 | 数据包类型 | 中继器ID | 设备ID | 数据长度 | 数据 | 校验 |
| TB$ | 0x92 | 001 |  | N |  | 0x00 |

返回信息为无反馈的485设备ID。

0x93

状态变更报警

中继器->服务器

外围的485设备节点有状态变更时主动触发向服务器中心发送状态变化，数据的具体定义更具不同的设备来决定。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 包头 | 数据包类型 | 中继器ID | 设备ID | 数据长度 | 数据 | 校验 |
| TB$ | 0x93 | 001 |  | N |  | 0x00 |