气象环境监视子系统控制软件

设计与测试

**目 录**

[一、 设计方案 1](#_Toc125136914)

[1. 概述 1](#_Toc125136915)

[2. 依赖环境 2](#_Toc125136916)

[3. 运行模式 3](#_Toc125136917)

[4. 配置参数 3](#_Toc125136918)

[5. 电源管理 4](#_Toc125136919)

[6. 时间服务 4](#_Toc125136920)

[7. 云量相机 4](#_Toc125136921)

[8. 云量相机处理软件 5](#_Toc125136922)

[9. 气象站 7](#_Toc125136923)

[10. 夜天光 7](#_Toc125136924)

[11. 数据存储 7](#_Toc125136925)

[12. 网络通信：数据交互 7](#_Toc125136926)

[13. 网络通信：控制指令 9](#_Toc125136927)

[二、 测试记录 10](#_Toc125136928)

[1. 测试项 10](#_Toc125136929)

[2. boost::circular\_buff效率 12](#_Toc125136930)

[3. 气象站编码格式 13](#_Toc125136931)

# 设计方案

## 概述

“气象环境监视子系统”（以下简称：子系统）控制软件，实现对构成子系统各硬件设备的控制、监视功能，并通过网络接口向外部用户提供实时测量结果。

图示

描述已自动生成

图 1：硬件设备及连线拓扑结构

表格 1：IP和端口默认分配

|  |  |  |  |
| --- | --- | --- | --- |
| **设备** | **IP** | **掩码** | **网关** |
| 路由器 | 192.168.0.1 | 24 | 192.168.0.1 |
| PDU | 192.168.0.2 |
| NTP | 192.168.0.3 |
| 树莓派 | 192.168.0.4 |
| 控制机 | 192.168.0.5 |
| 夜天光 | 192.168.0.6 |
| 组播 | 224.1.1.10:5000 |  |  |

表格 2：PDU端口默认分配

|  |  |
| --- | --- |
| **设备** | **端口** |
| 路由器 | 1 |
| NTP | 2 |
| 树莓派 | 3 |
| 控制机 | 4 |
| 监测设备 | 5 |
| 其它 | 6 |

## 依赖环境

* x86\_64内核Linux
* upsmonitor：UPS监视及电源管理软件
* wemon：气象环境监测设备控制软件
* cloudage：可见光全天云量分布计算软件
* 工具：
  + gcc-c++
  + gcc-gfortran
  + make
  + make
  + pip
  + iraf
  + pyraf
  + sextractor
* C/C++库：
  + boost
  + cfitsio
  + libusb
  + python3-devel
* Python库：
  + time
  + datetime
  + matplotlib
  + astropy
  + numpy
  + scipy
  + pathlib
  + shutil
  + math
  + glob
  + tokenize
  + sys
  + 2to3
  + optparse
  + socket
  + skimage

## 运行模式

* 系统服务：开机自启动
* 组播：224.1.1.10:5000。使用UDP组播，定时向外发送测量结果
* 在路由器上配置NAT：

1. 组播
2. 服务端口
3. SSH（22）

## 配置参数

* 组播地址：224.1.1.11:5000
* 广播信息格式。1：JSON格式；2：结构体协议
* UDP服务端口：响应树莓派发来的关机指令；响应用户发来的设备断电、重启指令；其它指令
* 智能PDU地址：向PDU发送对应端口的开启、关闭、重启指令
* PDU的端口-设备对应表，设备包括：
  + 树莓派
  + NTP
  + 监测设备
* 夜天光仪地址
* 云量相机控制策略
  + 文件存储目录
  + 工作时间区间
  + 曝光时间调整策略
  + 饱和值
  + 制冷温度
* 采样周期

## 电源管理

1. 控制机BIOS配置：上电自启
2. 解析来自upsmonitor的power offline指令，执行：
   1. 向PDU端口发送关闭指令，切断设备的电源
   2. 控制机软关机

## 时间服务

利用chronyd服务，维持控制机时钟与NTP时钟的一致性

## 云量相机

1. 基本功能：连接，断开，制冷，曝光，存储文件
   1. 故障诊断点1：温度采集异常
   2. 故障诊断点2：图像读出异常
2. 磁盘管理：清理历史数据
3. 曝光周期：在设定时段内开始、结束曝光序列，并写入日志
4. 曝光时间：依据图像统计结果，在范围内调节曝光时间
5. 读出异常时重连、重启相机，并写入日志
6. 图像文件存储路径：
   1. 在/data下创建Y<ccyy>目录，记录ccyy年的原始图像文件
   2. 在/data/Y<ccyy>下创建<prefix><yymmdd>目录。<prefix>是目录名前缀，默认是WMC，yymmdd是UTC时间的年月日
   3. 在/history/observed.list中，记录UTC时间当日图像文件列表。文件采用文本格式，每行对应一帧图像、包含2列。

第一列：文件所在的目录名列表，示例：/data/Y2023/WMC230110

第二列：文件名，格式：C<ccyymmdd>T<hhmmss>.fit，示例：C20230110T100102.fit

1. FITS文件头：

|  |  |  |
| --- | --- | --- |
| **关键字** | **数值类型** | **说明** |
| BITPIX | 整数 | 像元数据类型 |
| BZERO | 整数 | 默认：32768 |
| BSCALE | 整数 | 默认：1 |
| NAXIS | 整数 | 默认：2 |
| NAXIS1 | 整数 | 图像宽度 |
| NAXIS2 | 整数 | 图像高度 |
| CCDTYPE | 字符串 | 曝光类型，默认：OBJECT |
| DATE-OBS | 字符串 | 格式：CCYY-MM-DD，UTC日期 |
| TIME-OBS | 字符串 | 格式：hh:mm:ss.ssssss，曝光起始时间，UTC |
| TIME-END | 字符串 | 格式：hh:mm:ss.ssssss，曝光结束时间，UTC |
| JD | 实数 | 儒略日，曝光起始时间，UTC |
| EXPTIME | 实数 | 曝光时间，秒 |
| GAIN | 实数 | 增益档位 |
| TEMPSET | 整数 | 制冷温度，摄氏度 |
| TEMPGET | 整数 | 芯片温度，摄氏度 |
| TERMTYPE | 字符串 | 终端类型，默认：CloudCamera |
| TELFOCUS | 整数 | 焦距，毫米 |
| FRAMENO | 整数 | 当日帧序号 |
| DEVID | 字符串 | 设备编号 |
| SITENAME | 字符串 | 测站名称 |
| SITELON | 实数 | 地理经度，角度，东经为正 |
| SITELAT | 实数 | 地理纬度，角度，北纬为正 |
| SITEALT | 实数 | 海拔，米 |

## 云量相机处理软件

* 云量定义：



图 2：全天/超广角云量相机天区分割定义示意

云量分为10级，即L0-L9，每级之间相差0.4星等。分别对应：

|  |
| --- |
| L0-L3：执行跟踪型计划  L4-L7：优化定点型计划  L8-L9：该天区不可观测 |

* 与控制软件的交互机制：

1. 在/history目录维护observed.list，追加更新当日产生的图像文件；
2. 在/history目录维护cloudage.txt，记录最后一幅图像文件的处理结果。文件内容分为注释和正文两部分：
   1. 注释：以字符#为行起始符，记录用户自定义信息和规定信息。规定信息主要是3项：
      1. 设备编号：关键字：ID，内容：字符串编号

示例：

# ID 001

* + 1. 地理位置：关键字：SITE，内容：地理经度，地理纬度，海拔

示例：

# SITE 109.62514 18.34 44

* + 1. 天区步长：关键字：STEP，内容：方位步长，高度步长

# STEP 15 10

1. 正文第一行：状态。0表示有效，其它表示无效
2. 正文第二行：UTC时间，格式：CCYY-MM-DDThh:mm:ss
3. 正文第三行至文件末尾：每行记录一个天区的处理结果。每行包括3项内容，分别是：中心方位角 中心高度角 云量

中心位置采用地平系，方位角采用正北作为零点。

1. 示例：

# User’s customized comments

# ID 001

# SITE 109.62514 18.34 44

# STEP 15 10

#

0

2023-01-10T13:01:02

0 20 5

30 20 4

## 气象站

1. 软件启动后扫描串口，依据协议类型查找对应的串口设备名称。若所有串口都无法获得信息，则记录异常
2. 定时读取气象信息。若无法获得信息，则记录异常

## 夜天光

1. 软件启动后，定时读取信息。若无法获得信息，则记录异常

## 数据存储

1. 测量结果写入日志
2. 测量结果记入内存，并以组播形式对外广播

## 网络通信：数据交互

以组播形式，按照配置参数实时广播气象环境测量结果。

* 通信协议格式1：1字节对齐结构体

struct CloudAge {

float azi, ele; ///< 中心指向

int level; ///< 云量等级

};

typedef std::vector<CloudAge> CloudAgeVec;

struct EnvInfo {

int typeProto; // 协议类型 == 8

char id[12]; // 设备编号

char utc[20]; // UTC时间, 格式: CCYYMMThhmmss.sss

float magBk; // 夜天光亮度，量纲：星等@平方角秒

float temp; // 环境温度，量纲：℃

float humidity; // 相对湿度

float airp; // 大气压，量纲：百帕

float windSpd; // 风速，量纲：米/秒

int windDir; // 风向。0°：正北

int rainy; // 雨量。0：无雨

int nZone; // 全天云量相机划分天区数量

float azStep; // 云量相机方位轴步长

float elStep; // 云量相机高度轴步长

CloudAgeVec cloudAge; // 云量

};

* 通信协议格式2：JSON格式

{

“id”: “01”,

"utc": "2022-01-01T01:00:00",

"Weather":

{

"state": 0,

"utc": "2022-01-01T00:59:00",

"temperature": 20.0,

"humidity": 40.0,

"pressure": 1008.0,

"windSpeed": 1.5,

"windDirection": 300.1,

"rainfall": 0

},

"SQM":

{

"state": 0,

"utc": "2022-01-01T00:59:00",

"mpsas": 18.5

},

"Cloudage":

{

"state": 0,

"utc": "2022-01-01T00:59:00",

"azStep": 10,

"elStep": 10,

"distribution": {

"azi": 0,

"ele": 0,

"level": 1

}

}

}

## 网络通信：控制指令

控制指令的通信协议格式为：

1. 采用字符流，以换行符结束
2. 由operator-keyword-value组成，形式为：

operator keyword1=value1, …, keywordn=valuen<LN>

* upsmonitor=>wemon：

1. poweroff
   * + 功能：UPS输入电源异常，需要对系统设备软关机
     + 指令：poweroff
     + 参数：无

* 控制台（调试）=>wemon

1. reset
   * + 功能：重启PDU对应端口电源
     + 指令：reset
     + 参数：port
     + 示例：

reset port=1

# 测试记录

## 测试项

|  |  |  |
| --- | --- | --- |
| **名称** | **内容** | **结果** |
| 配置参数 | 1. 加载 2. 初始化和保存 | 通过 |
| 编译QHY链接库 | /usr/bin/ld: /usr/local/lib/libqhyccd.so: undefined reference to `cvCanny'  …   1. 需要OpenCV链接库但未定义 2. 在cmakefiles.txt中，定义： | 与环境有关。   1. ARM Linux提示该错误 2. Ubuntu 22提示该错误 |
| 晨昏蒙影 | 1. 计算UTC日期变更时的晨昏蒙影。日期初始设置为-1，强制软件启动时做一次检测 2. 在Mac下，timezone==0 3. 在Linux下，ptime::date().year()==65579 |  |
| 定时断开与相机的连接 | 疑似与相机断开时软件崩溃 | 继承类调用shared\_ptr<>.reset()，没有完全执行析构函数。  显示调用Disconnect()，问题解决  通过 |
| SQM | 1. 在晨昏蒙影时段工作 2. 文件记录在<root dir>/SQM/Y<CCYY>/SQM\_CCYYMMDD.log中 3. 文件内容：<utc string> <mpass> |  |
| 气象站 | 1. 全天时工作 2. 文件记录在<root>/Weather/Y<CCYY>/Weather\_CCYYMMDD.log中 3. 文件内容：   <utc string> 温度 湿度 气压 风速 风向 雨量 |  |
| 云量相机自动采集 | （1）在晨光前-昏影后时段工作 |  |
| 云量计算结果 |  |  |
| 组播协议 |  |  |
| 组播跨路由器 | 需要路由器支持组播功能 | TP-Link工业路由器不支持组播功能 |
|  |  |  |

## boost::circular\_buff效率

* 测试条件：仿真网络通信中的缓冲区，存储100帧、每帧1024字节
* 测试功能：
  1. 写入100\*1024字节数据
  2. 每次删除1024字节数据
  3. 缓冲区线性化
  4. 共测试50次
* 结果：
  1. 删除和线性化操作耗时极少：100KB耗时约1微秒
  2. 写入操作统计平均231微秒@100KB，误差1.8微秒
  3. 延伸：对于4096\*4096\*2+2880字节大典型FITS文件，写入操作耗时0.08秒
* 测试数据：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 写入 | 删-线 | 写入 | 删-线 | 写入 | 删-线 | 写入 | 删-线 | 写入 | 删-线 |
| 526 | 0 | 230 | 0 | 230 | 1 | 231 | 0 | 235 | 0 |
| 231 | 0 | 230 | 1 | 230 | 1 | 230 | 1 | 231 | 0 |
| 230 | 0 | 232 | 0 | 232 | 0 | 230 | 0 | 234 | 0 |
| 231 | 0 | 230 | 3 | 231 | 0 | 230 | 0 | 231 | 0 |
| 230 | 0 | 230 | 1 | 230 | 1 | 230 | 1 | 230 | 1 |
| 230 | 0 | 230 | 0 | 230 | 0 | 230 | 0 | 231 | 0 |
| 231 | 0 | 230 | 1 | 240 | 1 | 230 | 0 | 230 | 0 |
| 230 | 1 | 230 | 0 | 231 | 1 | 230 | 1 | 232 | 0 |
| 230 | 0 | 232 | 0 | 232 | 1 | 230 | 0 | 231 | 0 |
| 230 | 0 | 233 | 0 | 233 | 0 | 230 | 0 | 230 | 1 |

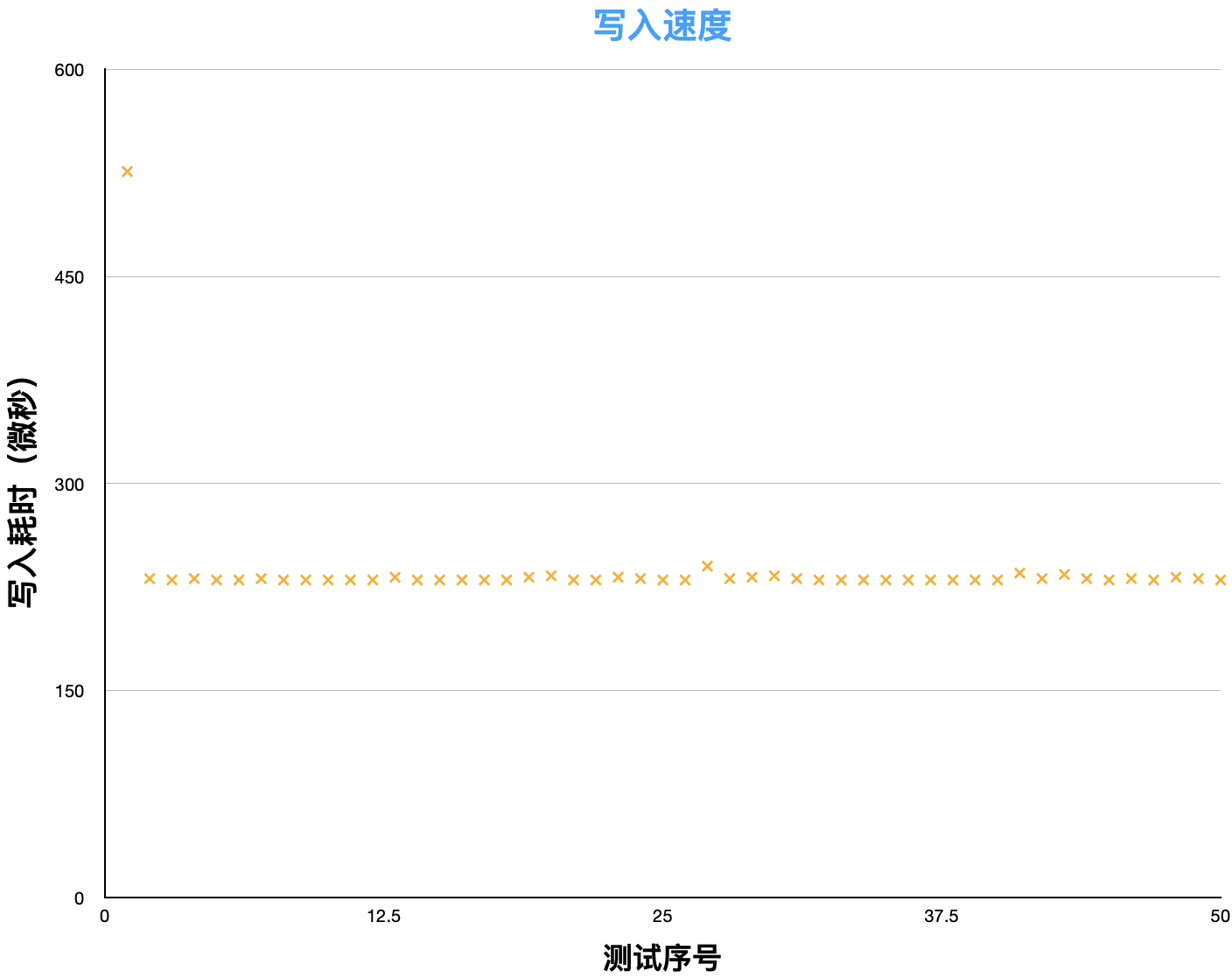


图 3：循环缓冲区写入速度

## 气象站编码格式

### 地址

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **功能** | **功能地址** | **寄存器地址** | **数据类型** | **量纲** |
| 温度 | 0x66 | 0x0000 | 1 | 0.1℃ |
| 湿度 | 0x0001 | 0.1%RH |
| 气压 | 0x0002 | 10Pa |
| 风速 | 0xC8 | 0x0000 | 1 | 0.01m/s |
| 风向 | 0x0001 | 1度 |
| 降雨 | 0xCA | 0x0048，高位  0x0049，低位 | 2 | 0.01mm |

### CRC校验码

unsigned short ModbusCRC16(unsigned char\* data, unsigned int len) {

unsigned int i, j;

unsigned short CRC16, tmp;

CRC16 = 0xFFFF;

for (i = 0; i < len; ++i) {

CRC16 ^= data[i];

for (j = 0; j < 8; ++j) {

tmp = (unsigned short) (CRC16 & 0x0001);

CRC16 >>= 1;

if (tmp == 1) CRC16 ^= 0xA001;

}

}

// CRC校验码流: 低字节在前

// unsigned char crc[2];

// crc[0] = (unsigned char) (CRC16 & 0xFF);

// crc[1] = (unsigned char) ((CRC16 >> 8) & 0xFF);

return CRC16;

}

### 查询指令编码

### 修改指令编码

### 风速和风向

### 温度、湿度、气压

### 降雨

* 读取

指令：CA 03 00 00 00 02 D4 70