

# 完成端口技术在路灯监控软件中的应用

刘墩东 黄祚 孙洪飞

(厦门大学自动化系, 福建厦门 361005)

**摘要:** 针对大规模的路灯监控系统, 设计了一套基于 C/S 模式的监控系统软件构架, 并且应用完成端口通信技术, 对监控系统的服务器进行网络通信优化, 使其能够承载大量的客户端, 提高了数据传输的高效性, 可靠性。

**关键词:** 监控系统; 完成端口; C/S 模式; Visual C++

## The Application of Completion Port Technology for Road Lighting Control Software

Liu Tundong Huang Zuo Sun Hongfei

(Automation Department, Xiamen University, Xiamen 361005)

### Abstract

For Monitoring system for large-scale road lighting, designed a set control system software frame based on C/S mode, and the system for monitoring network communications servers is optimized by the application of completion port communication technology so that it can carry a large number of clients and improve the data transmission efficiency and reliability.

**Key words:** control system; completion port; C/S mode; Visual C++

## 1 引言

目前, 全国很多城市的路灯监控系统受到区域限制, 仍停留在小规模监控模式上, 使得各地区的监控标准不统一, 管理混乱, 同时也占用了大量的人力和物力资源。因此, 将各区域的路灯监控系统进行统一的管理, 形成一个大规模统一的监控体系, 已成为将来路灯监控发展的趋势。传统的 SOCKET 通信模型有着客户端数量的限制, 当实际的客户端超过限制, 将会出现数据阻塞和丢失, 甚至是服务器软件崩溃的情况, 而引入了完成端口技术的通信模型没有客户端数量的限制, 并且拥有着高效的数据处理能力, 能够在大规模路灯监控系统内发挥优势, 保障了数据传输的高效性和可靠性。在 Visual C++ 2008 编程环境下, 通过完成端口技术的应用, 将原有的基于 C/S 模式的路灯监控系统软

件进行优化, 使得整套系统可以应用于大数量客户端的场合, 并且仍能保持通信系统较高的稳定性。

## 2 监控系统软件的总体构架<sup>[1]</sup>

路灯监控系统分为远程终端设备和监控软件两个部分。远程终端设备安装在路灯控制现场, 是实现监控功能的主要硬件设备。远程终端通过 GPRS 无线通信网络与服务器相连<sup>[2]</sup>, 根据用户的设置参数, 实现定时开关灯, 采集数据和事故报警等功能。根据不同地区的情况, 其数量可能非常的庞大, 传输到服务器的数据量也会非常庞大。监控软件是一套在 Visual C++ 2008 开发平台下, 基于 Client/Server 模式的网络通信软件<sup>[3]</sup>, 由服务端软件和客户端软件两个部分组成, 后台数据库选用 MS SQL Server 2005。监控系统结构图如图 1 所示。

监控软件的服务端安装并工作于服务器上, 负

责接收监控终端设备传输而来的数据，对数据进行分析，并存入数据库；同时与软件的客户端进行通信，并且将软件客户端的指令数据，转发到相应的监控终端设备，对被监控对象的进行管理与控制。监控软件的客户端工作在用户电脑上，通过网络与服务端和数据库相连，为少数特定的路灯监控管理员提供服务。客户端为这些管理员用户提供了一个功能齐全的图形界面。用户可以通过客户端查询数据，发送控制指令，也可以通过客户端的电子地图功能和柜体监控动画实时的了解各个远程终端的工作状态。

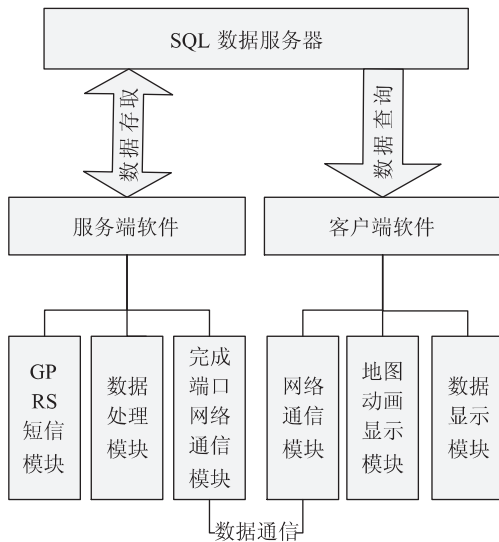


图1 系统结构图

### 3 服务端完成端口通信模型的实现<sup>[4]</sup>

#### 3.1 完成端口原理

##### 3.1.1 完成端口简介

网络通信模块是整个系统最核心的部分，由于要负责大规模的数据传输与处理，因此对软件的性能的高效性提出了挑战，而完成端口通信技术的应用解决了这一难题。

完成端口（I/O Completion Port）是一个 Windows NT 执行子系统的核心对象。通过将完成端口与任意 I/O 句柄（文件或 Socket 等）关联，使得用户可以通过完成端口，异步的获取并处理 I/O 的结果。

完成端口是由系统直接提供并行优化支持的，

在完成端口上建立几个并行的服务线程，一般数量为 CPU 数，它们为到达完成端口的服务请求提供服务。当有服务请求到达时，如果有可用的服务线程，则激活该线程，如果没有可用服务线程，则将服务请求加入请求队列，该队列采用先进先出（FIFO）的策略，来保证这些请求得到公平的服务。服务线程的建立和请求队列的 FIFO 策略，减少了 CPU 在不同线程间切换的次数，降低线程上下文切换所造成的开销。

##### 3.1.2 重叠 I/O

完成端口的设计原理是让应用程序使用重叠的数据结构，一次投递一个或多个 I/O 请求，当这些请求完成后，应用程序可以为它们提供服务。这就要求我们在使用完成端口时必须使用重叠 I/O。重叠 I/O，即当 I/O 功能调用时，不论 I/O 是否完成，函数马上返回，由操作系统底层处理 I/O 的实际工作，而应用程序（进程）可以继续做其他事情。因而，完成端口是处理完成重叠 I/O 的一种高效的机制<sup>[5]</sup>。

##### 3.1.3 工作线程

除了工作在完成端口上的服务线程外，在关联套接字之前，还必须创建一个或多个工作线程，以便在 I/O 请求投递给完成端口对象后，为完成端口提供服务。工作线程的个数取决于应用程序的总体设计情况。创建的工作线程由完成端口管理。当有 I/O 完成通知到来，则由完成端口唤醒一个工作线程接收 I/O 完成通知，并对其进行处理。完成端口自动对工作线程进行调度，唤醒哪个工作线程则由完成端口决定。若无 I/O 完成通知，则所有的工作线程都在等待。根据经验，工作线程的数量一般为 CPU 数量的两倍再加上 2。

#### 3.2 完成端口的程序实现<sup>[6]</sup>

网络通信模块通过 CreateIoCompletionPort 函数创建完成端口对象，并将接收到的 SOCKET 对象与完成端口关联，启动一定数量的工作线程，通过 GetQueuedCompletionStatus 函数获取完成端口上 SOCKET 的当前状态，并将收到的数据从缓存中取出。完成端口的主要工作流程图如图 2 所示。

主线程：

1) 程序启动的时候，初始化网络并且创建完成端口句柄：

```
CompletionPort = CreateIoCompletionPort ( INVA-
```

```

LID_HANDLE_VALUE, NULL, 0, 0);
2) 启动 2 * N + 2 个工作线程, N 为 CPU 数量:
for (int i=0; i < sys_CPU_NUM * 2 + 2; i++)
{
HANDLE ThreadHandle;
DWORD ThreadID;
ThreadHandle = CreateThread (
    NULL,
    0,
    ServerWorkerProc,
    this,
    0,
    &ThreadID);
CloseHandle (ThreadHandle);
}
    
```

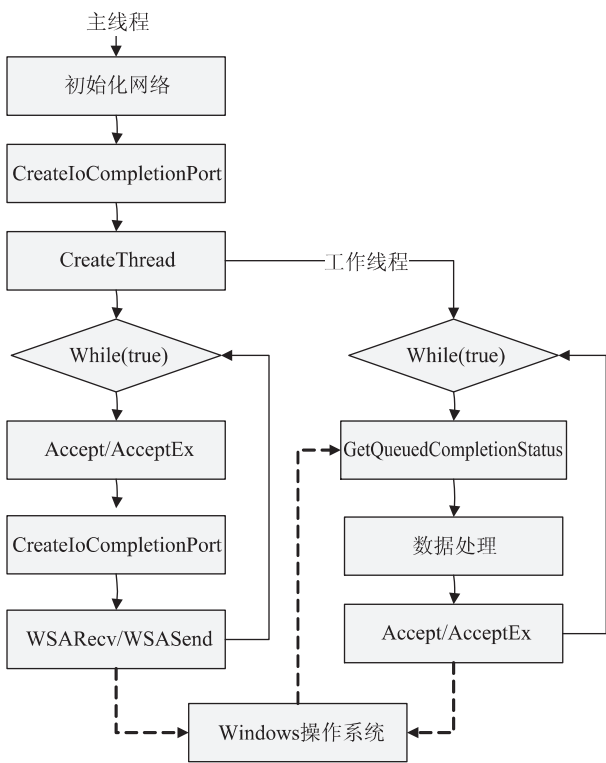


图 2 完成端口模块流程图

- 3) 进入一个监听循环, 开始监听客户端连接请求;
- 4) 将接收到的客户端 SOCKET 与完成端口对象绑定;

- 5) 发出一个异步的 WSARecv 或是 WSASend 操作, 实际的接收和发送数据操作会由操作系统完成。
- 6) 重复以上 3) 到 5) 的操作。

工作线程:

- 1) 进入循环, 通过 GetQueuedCompletionStatus 函数, 从完成端口上取得 WSASend/WSARecv 的操作结果;
 

```

GetQueuedCompletionStatus (
    CompletionPort,
    &ByteTransferred,
    (LPDWORD) &PerHandleData,
    (LPOVERLAPPED*) Per_IO,
    INFINITE);
            
```
- 2) 根据完成端口上 I/O 状态, 进行数据的处理;
- 3) 提交一个新的 WSASend/WSARecv 操作请求;
- 4) 重复以上 1) 到 4) 的操作。

### 3.3 通信规约设计

整个监控系统采用 TCP (Transmission Control Protocol, 传输控制协议) 进行数据传输, 在此基础上设计了一套监控系统规约, 来完成服务端与远程终端, 服务端与客户端的通信。根据路灯监控的实际需求, 数据报文包括以下几种形式。

- 1) 远程终端主动向软件服务端发送的连接认证数据报文, 如表 1 所示。

表 1 连接认证数据报文格式

字节数	数据内容
1 个字节	起始位
6 个字节	6 位认证 ID
2 个字节	CRC 校验
1 个字节	结束位

- 2) 远程终端定时向软件服务端发送的现场数据报文, 主要包括路灯监控现场采集到的电流, 电压, 温度, 开关状态, 报警信息等数据信息, 如表 2 所示。

- 3) 软件客户端发送给服务端, 并由服务端转发到相应远程终端的参数设置报文, 根据不同的功能号, 报文发送不同的参数信息, 包括开关灯时间,

报警阈值，数据采集周期等如表 3 所示。

表 2 现场数据报文

字节数	数据内容
1 个字节	起始位
1 个字节	数据位长度 N
N 个字节	数据位
2 个字节	CRC 校验
1 个字节	结束位

表 3 参数设置报文

字节数	数据内容
1 个字节	起始位
1 个字节	功能号
1 个字节	数据位长度 N
N 个字节	数据位
2 个字节	CRC 校验
1 个字节	结束位

### 3.4 完成端口通信的优化

#### 3.4.1 内存池的设计

完成端口模型采用异步通信模式，每次调用 `WSASend` 和 `WSARecv` 函数都需要在内存创建一个结构体空间，函数调用完毕后，再销毁这个结构体空间。频繁的创建和销毁内存空间占用了大量的系统资源，因此，在设计完成端口程序时，根据需求创建一定数量的结构体空间，并将其放入一个统一的空闲队列，当调用 `WSASend` 和 `WSARecv` 函数时，从队列中取用一个结构体空间，使用完毕，再将其放回队列。

#### 3.4.2 连接池的设计

当用传统的 `accept` 函数接收客户端时，`accept` 函数会创建一个 `socket` 作为返回值，分配给客户端。客户端断开连接时，创建的 `socket` 会被销毁。创建和销毁 `socket` 的过程会占用大量的系统资源，因此在接收客户端时，采用 `acceptEx` 函数代替 `accept`，该函数可以把一个事先创建好的 `socket` 对象，分配给接收到的客户端。首先，创建好一定数量的 `socket` 对象，形成一个连接池，当接收到客户端的连接请求时，从连接池中取出空闲 `socket` 对象，分配给该客户端，断开连接时，再将 `socket` 放回连接池队列。连接池的设计减少了客户端 `SOCKET` 的不断创建与销毁，节省了大量的系统资源。

#### 3.4.3 线程池的设计

完成端口本身就应用了线程池技术，线程池中的线程不仅包括了工作者线程，还包括了工作上完

成端口上的服务线程。有效的对这些线程进行管理，能够减少 CPU 在不同线程间的频繁切换，降低了切换线程上下文所耗费的时间。

#### 3.4.4 数据池的设计

完成端口模块接收到的数据，要根据通信规约进行处理与分析，并将数据存储到相应的数据库中。由于完成端口网络通信的数据传输总是不平稳的，常常会出现短时间内接收到大量数据，而另一段时间内只接收到少量数据要的情况。为了防止服务器在短时间内超负荷工作，造成的数据意外丢失或是程序崩溃的情况，在进行数据处理时，预先建立了数据存储队列，形成一个数据池，将未处理的数据加入队列，并采用 FIFO 的策略来分配 CPU 时间，这就使得 CPU 资源得到充分的利用，提高了数据处理的安全性和可靠性。

## 4 客户端软件设计

客户端软件通过一般的 `SOCKET` 通信方式与服务器相连，主要是功能是为用户提供一个简洁，便利的用户功能界面。地图显示模块通过对 GIS 电子地图的绘制，将城市地图及路灯系统的分布图直观的显示给用户，使得用户能够大体的了解到整个路灯系统的运行状态。动画显示模块通过 `FLASH` 编程技术，将单个远程终端所控制的配电柜示意图展示给用户，用户可以了解到现场的实时数据并对具体的监控点进行设置，开关灯等操作。数据显示模块与数据库相连，用户可以查询到各个监控点的历史数据以及当前的设置参数，了解路灯系统的具体工作状态。软件客户端主界面如图 3 所示。

## 5 完成端口服务器软件的性能测试<sup>[7]</sup>

### 5.1 测试对象

完成端口通信模型与传统通信模型相比，拥有更大的数据吞吐量和客户端数目，并且通过线程池、连接池、内存池的设计和应用，节省了系统资源，提高了服务器软件的数据处理效率。在对传统通信模型和完成端口通信模型的性能测试和比较中，选取饥饿的客户端和每秒线程上下文切换次数两个重要指标为测试对象。饥饿的客户端定义为同一时间向服务器申请连接并发送数据的客户端中，未被服

务器影响的客户端数。

### 5.2 测试环境

选用两台 Intel Core2 1.9GHz 双核 CPU, 2G 内

存台式机, 一台用作服务器电脑, 一台用作客户端电脑。服务器电脑上分别安装传统通信模型的旧版路灯监控软件和完成端口模型的新版路灯监控软件,

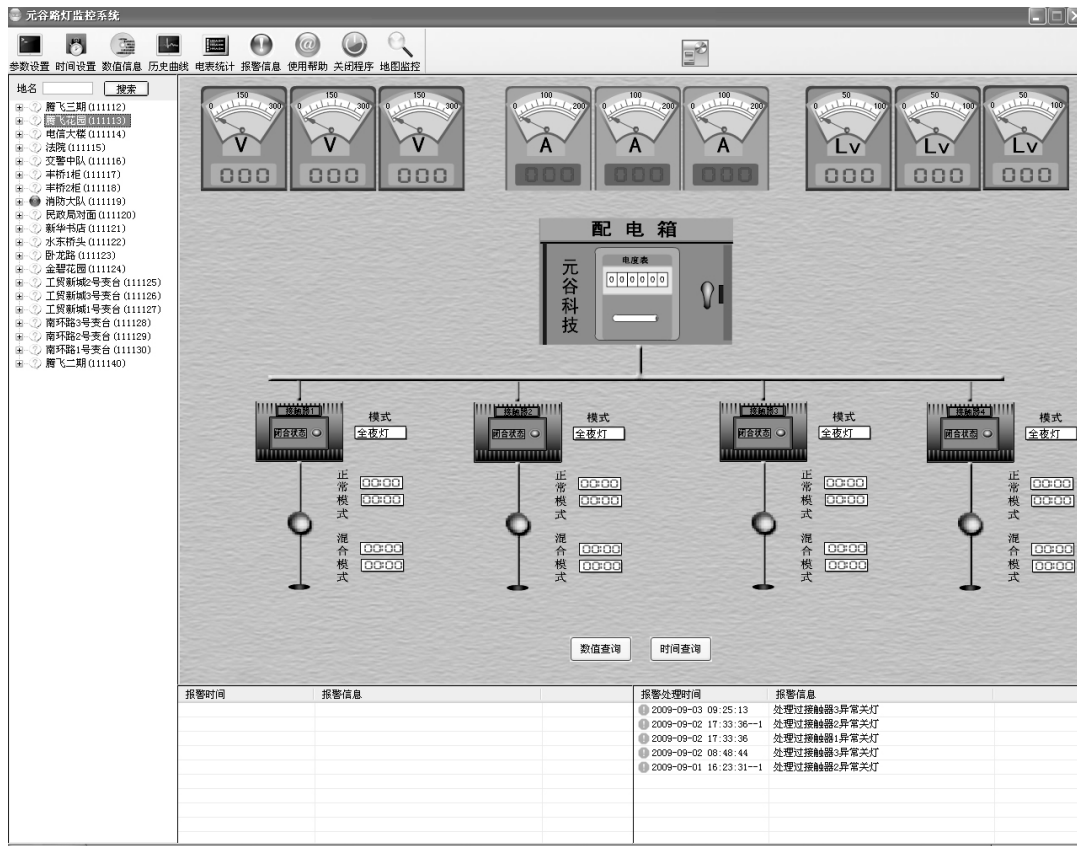


图3 客户端软件主界面

并且在软件程序中加入测试代码, 用来计算饥饿客户端数目和线程上下文的切换次数; 客户端电脑上用测试软件来模拟一定数量的终端设备的客户端, 并向服务器同时进行连接和发送数据的操作。

### 5.3 测试结果及分析

不断的改变模拟客户端的数量, 对两种通信模型进行测试, 分别记录下两种模型在不同数量的客户端下, 饥饿客户端数量和线程上下文切换的次数, 重复多次测试, 取得多组数据, 取其平均值。

如表4所示, 当模拟客户端数目逐渐增加时, 传统通信模型的饥饿客户端数量也不断增加, 这就使得大量的客户端无法得到服务器响应, 大量客户

表4 饥饿客户端测试

模拟客户端数	50	100	200	500	1000
传统通信模型	0	12	33	65	188
完成端口模型	0	0	0	0	0

端的数据无法传输, 导致数据的阻塞和丢失。而完

成端口通信模型采取了一系列的优化策略, 并不存在客户端无法得到服务的情况。

如表5所示, 在模拟客户端数量较少时, 两种通信模型的线程上下文切换次数相当; 当模拟客户端数量增加时, 传统通信模型的切换次数剧增, 而每次的切换都会导致系统资源的额外开销, 这就使的传统通信模型的数据处理效率十分低下。使用完成端口通信模型时, 线程上下文切换次数并未随着模拟客户端的增加而产生更大的变化, 因此完成端口模型更适合于大量客户端的应用场合, 并且仍可保持的数据通信的可靠性和高效性。

表5 每秒线程上下文切换次数

模拟客户端数	50	100	200	500	1000
传统通信模型	452	860	2410	2241	1968
完成端口模型	354	359	366	370	388

(下转第76页)

## 5 调光装置与计算机系统的通信

如图5所示,调光装置和计算机系统之间采用RS-232串行通信标准,其规定了发送端驱动器与接收端接收器的电平关系、负载要求、信号速率与连接要求等,MAX232主要用来完成TTL电平和RS-232之间电平的转换。

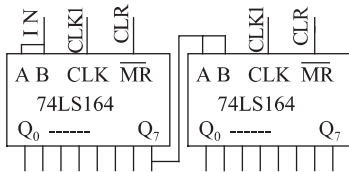


图5 调光装置与计算机系统的通信接口图

## 6 结论

设计并制作了一套LED调光装置,采用单片机通过三路D/A转换输出分别控制LED红、绿、蓝三基色输出,实现对半导体照明色彩和亮度宽范围、

高精度的调节,同时通过串并转换扫描方式使一套系统可对多个LED分别控制,实现其高精度控光和控色的基础上可组成任意的静、动态图案或画面。通过对实际制作实验模型128×64LED组成的矩阵进行图案和色彩、亮度进行8小时测试,表明LED中心光强稳定在280 lx (lx:勒克斯),随着LED温度的升高中心光强衰减低于3%,达到了预期的控制要求水平。

### 参考文献

- [1] 马磊,刘德明等. 太阳能半导体照明系统控制器的研究与设计 [J]. 半导体光电, 2007, 4 第28卷第2期: 279~282
- [2] 王新贤. 通用集成电路速查手册 [Z]. 济南. 山东科技出版社. 2004
- [3] 陈维,沈辉等. 太阳能半导体照明驱动技术研究 [J]. 照明工程学报. 2005.9 第16卷第3期: 8~10
- [4] 周杭慈. PHILIPS 51LPS系列单片机原理及应用设计 [M]. 北京. 北京航空航天大学出版社. 2001.5
- [5] 朴燕. 全色LED显示器中亮度参数的研究 [J]. 液晶与显示. 2007.10. 第22卷第5期: 503~505
- [6] AD558 Databook [Z] Analog Devices, Inc.

(上接第70页)

## 6 结束语

完成端口技术的引入,充分发挥了服务器多CPU的优势,使得整个监控系统的数据通信性能得到了极大的优化了。经过压力测试,当监控终端设备数量达5000时,系统仍然能够保持高效、稳定的运行。目前该系统应用于厦门路桥公司,龙岩长汀等地的路灯控制,取得了良好的效果。

### 参考文献

- [1] 赵炯,徐博铭,宋蕴璞. 火灾报警系统集成监控和管理软件设计 [J]. 计算机工程, 2008, 34 (16): 259~261
- [2] 王成福,唐晓强. 基于GPRS的路灯监控系统的设计与

- 实现 [J]. 电力系统通信, 2008, 29 (190): 18~21
- [3] 闫谦时,陈雷. 一种基于网络的监控软件设计与实现 [J]. 计算机与数字工程, 2009, 37 (2): 183~185
- [4] 基于IOCP机制的网络游戏服务器通信层的实现 [J]. 计算机工程与应用, 2009, 47 (7): 75~81
- [5] Gyu-baek Kim, An Effective Processing Server for Various Database Operations of Large-scale On-line Games [C], IASTED International Conference on Information and Knowledge Sharing, Arizona, U. S. A, November 2003, Vol. 1, pp. 188~192
- [6] 陈和平,王早,李晓卉. 基于单个I/O完成端口的HTTP代理方法研究 [J]. 计算机工程与设计, 2005, 26 (11): 2995~2997
- [7] 唐海娜,李俊. 网络性能监测技术综述 [J]. 计算机应用研究, 2004, 21 (8): 10~13