文章编号:1007 - 2985(2011)04 - 0068 - 03

多串口多线程交通信号灯数据采集软件系统设计*

张艳霞,胡双炎
(娄底职业技术学院,湖南 娄底 417000)

摘 要:提出了一种交通信号灯数据采集方法,采用多串口多线程技术实时采集多个路口的红绿灯信息,利用同步技术解决了各路数据的干扰. 通信双方采用自定义的通信协议,从而提高了通信的可靠性和可维护性. 中心端 PC 机实时接收各路信息,同时发送各种控制命令给路口,大大减弱了各个路口信号灯管理人员的劳动强度.

关键词:数据采集;多线程;多串口;红绿灯信息;通信协议
中图分类号:TP319 **文献标志码**:A

计算机与单片机的通信一般采用串口进行,上位机(计算机)与下位机(单片机)的串口线长度一般不超过 3 m,当然也可以通过延长串口连接线的方法来解决. 目前有许多学者研究基于串口的通信,其中 32 位下串口通信程序可以利用利用 ActiveX 控件和 API 通信函数 2 种方法实现. 笔者在参考文献的基础上,提出了多串口多线程交通信号灯数据采集的设计方法,并在 VC++中运用面向对象方法设计了 1 个基于 Win32 的 API 的串口封装类 CSerialEx,然后基于多线程父类 CWinThread 设计了子类 CSerialThread,在子类中运用了 CSerialEx 来进行多线程多串口的通信,最后封装成 DLL 组件文件,从而方便升级与维护.

1 设计自定义串口操作类 CSerialEx

为了提高灵活性和访问速度,串口操作类 CSerialEx 封装了 Win32 API 访问串口的函数,CSerialEx 类的定义:

```
class CSerialEx
{public:
CSerialEx();
virtual ~CSerialEx();public:
void setStatusPort(BOOL on_off); // 设置串口是否打开或关闭
BOOL closePort(); //关闭串口操作
BOOL openPort(DCB dcb, // 打开串口操作
const char* portName ="COM1"); // 定义串口变量名称,缺省为"COM1"
//读串口通信数据
BOOL read_scc(char* inputData, //读串口数据的输入缓冲区
const unsigned int& sizeBuffer, // 缓冲区最大长度
unsigned long& length); // 实际接收缓冲区数据的长度
//向串口写数据
BOOL write_scc(LPCVOID data, // 写数据的缓冲区指针
const unsigned int& sizeBuffer, // 写缓冲区的最大长度
unsigned long& length); // 写缓冲区的实际长度
HANDLE getHandlePort(); // 获取串口的句柄
BOOL getStatusPort(); // 获取串口的打开状态
private:
BOOL statusPort_; // 串口的打开状态
HANDLE handlePort_; // 串口的句柄
DCB config_; // 串口通信的结构变量
}
```

CSerialEx 类的具体实现过程在此不再赘述.

* 收稿日期:2011 - 05 - 26
作者简介:张艳霞(1984 -),女,湖南常德人,娄底职业技术学院教师,硕士,主要从事计算机应用研究.

2　设计自定义多线程操作串口类 CSerialThread

类 CSerialThread 的定义为：

```
class CSerialThread:public CWinThread//从线程基类继承
{ DECLARE_DYNCREATE(CSerialThread)
protected:
CSerialThread(); // 保护构造函数
public:
CSerialExm_SerialEx;//定义串口类的变量
DCB configSerial_;//定义串口的参数结构变量
CView* p_View;//定义数据要显示的视图
public:
public:
virtual BOOL InitInstance();
virtual int ExitInstance();
virtual int Run();//线程运行函数，作用就是不断地接收串口返回的数据并做处理
DECLARE_MESSAGE_MAP()
}
```

3　在视图类中调用 CSerialThread

3.1 视图类中要做的工作

在 VC++的视图类中定义指针变量：

```
CSerialThread* serialProcess;
```

在视图类的初始函数 OnInitialUpdate()初始化指针变量：

```
serialProcess = (CSerialThread*)::AfxBeginThread(RUNTIME_CLASS(CSerialThread));
serialProcess->p_View=this;//把视图对象指针传入线程中，方便显示线程中的数据
```

在视图类中定义函数 OnReceiveData 来把线程中接收的数据显示到视图界面中：

```
OnReceiveData(char * buff,int length)//buff 是接收数据的缓冲区，length 指缓冲区的长度.
```

3.2 视图类中的执行过程

当程序运行时，视图类中的初始函数 OnInitialUpdate 会自动调用，就启动线程 serialProcess，然后线程中的 Run 运行函数就不断运行，接收所打开的串口（第 1 个串口是“COM1”）的数据，并调用视图中的函数 OnReceiveData(char * buff,int length)来把接收的数据显示到视图界面中.

3.3 同时接收多串口数据的显示

上述只是启动了 1 个线程来接收 1 个串口的返回数据，如果要同时接收多个串口的数据，则可以在视图类中创建 1 个线程数组对象，具体过程如下：

在 VC++的视图类中定义指针数组变量：

```
CSerialThread* serialProcess[MAX_PORTS];
//其中 MAX_PORTS 为总的串口数量
```

在视图类的初始函数 OnInitialUpdate()初始化指针变量：

```
for(int i=0;i<MAX_PORTS;i++)
{
serialProcess[i] = (CSerialThread*)::AfxBeginThread(RUNTIME_CLASS(CSerialThread));
serialProcess[i]->p_View=this;//把视图对象指针传入线程中，方便显示线程中的数据
}
```

在视图类中定义函数 OnReceiveData 来把线程中接收的数据显示到视图界面中：

```
OnReceiveData(char * buff,int length)//buff 是接收数据的缓冲区，length 指缓冲区的长度
```

在多线程数据中的 OnReceiveData 函数中要用同步信号量（或者临界区）保证各个串口返回的数据不会混到一起.

3.4 当串口数量超过 9 以后的注意事项

当串口小于 10 时，打开串口的通用过程为：openPort(dcb,“COMn”);//其中 n 的取值为 1 到 9

当串口大于 9 以后，打开串口的通用过程为：openPort(dcb,“\\\\.\\COMn”);//其中 n 的取值为 9 之后的整数

如果要打开串口“COM10”，如果用 openPort(dcb,“COM10”);就会出错，提示不能打开串口 COM10. 如果改成 openPort(dcb,“\\\\.\\COM10”);就能成功，这是经过多次的实验才找出的规律.

4　上位机与交通控制器的通信协议

为了能方便地使上位机与交通控制器之间进行通信，要制定一个便于双方通信的通信协议. 指挥中心上位机与交通控制路口机间的数据按帧传输，每帧数据的格式如表 1 所示.

表 1 数据按帧传输表

帧名	数据长度/B	备注
起始字头 AA	1	作为一帧数据的起始,可以指定为 #0AAH
路口地址码	1	用来标志各个路口
命令字	1	进行操作的种类
保留字	1	作为将来的扩展功能用
数据块长度	1	长度为 #00～#0FFH,供接收完数据长度的判断
数据块	若干	不同命令,对应不同命令,数据块长度不一样
校验和	1	作为此次数据接收完整判断之用

(1) 校验和的作用范围应包括校验和字节之前的所有字节,但是只保留低 8 位,保证双方通信正确有效.

(2) 通讯的波特率设置. 通讯为异步串行方式,1 个起始位,8 个数据位,1 个停止位,无奇偶校验.

(3) 数据块的含义. 数据块是本数据帧所附带的与命令字相关的参数或数据. 其有效长度可为 0～120,当为 0 时即本帧无数据块或参数,这种情况出现在数据采集"命令帧"中及数据下载"应答帧"中.

(4) 传输约定. (ⅰ)每个通讯过程均由上位通讯机发起;路口机则根据通讯机所发的命令字意义给予应答,即 1 个通讯机发出的"命令帧"对应 1 个路口机返回的"应答帧";(ⅱ)通讯机的命令格式和路口机的应答格式均采用以上规定的相同的传输格式;(ⅲ)命令帧"有 2 类,其中第 1 类为监测,第 2 类为设置.

5 结语

多串口多线程数据采集方法用于交通信号灯信号数据采集过程中,中心端电脑同时采集的交通路口数量小于 30 个,能实时地接收数据并解析显示到视图界面. 当超过 30 个时,软件运行速度有些缓慢,有时接收的数据错误率比较高. 此时必须再添置 1 台电脑来接收超过 30 个以上的返回的数据. 当同时有 30 路交通信号数据返回来,同时显示到视图界面上面,出现闪烁现象. 这时可以从 2 个方面来减轻闪烁的程度:(1)在视图区中采取双缓冲区技术来显示数据;(2)在每个路口返回的红绿灯信号时间上找规律.

参考文献:

[1] 石海杰,常 虹. 基于 VC 的多线程串口通信程序设计 [J]. 可编程控制器与工厂自动化,2009(9):65－67.
[2] 邱建华,彭志豪. 串口通信技术在 VisualC＋＋中的实现 [J]. 软件工程师,2010(2):94－95.
[3] 庞军平,田梦君,陈 华. 基于串口通讯和多线程技术的应用软件开发 [J]. 机械与电子,2009(2):47－49.
[4] 李 勇. 一个多串口多线程数据采集系统软件的设计与实现 [J]. 微计算机信息,2006,22(16):152－154.
[5] 申晓宁,赵毅强,张 进,等. 多线程串口类在实时数据采集系统中的应用 [J]. 微计算时代息,2010(1):28－30.

Data Collecting Software Design for Traffic Lights' Signal Based on Multi-Serial Port and Multithreading

ZHANG Yan-xia, HU Shuang-yan
(Loudi Vocational & Technical College, Loudi 417000, Hunan China)

Abstract: An improved data collecting method for traffic lights' signal is presented, which uses multi-threading and multi-serial port technology to collect the traffic lights information from separate road junctions at the same time and uses synchronous technology to solve the interference from separate road junctions' data information. A custom communication protocol is put forward for the both communicating sides to improve communication reliability and maintainability. The center-side PC can receive all the real-time information from separate road junctions and at the same time can send various control commands to the road junctions, which can greatly reduce the labor intensity of the management staff of separate road junctions.

Key words: data collection; multi-threading; multi-serial port; the traffic lights' signal; communication protocol

(责任编辑 陈炳权)