

目 录

1引言..... 1

1.1问题的提出..... 1

1.2国内外研究的现状..... 1

1.5任务与分析..... 2

21.5.1主要任务..... 2

21.5.2分析..... 2

2程序的主要功能..... 2

2.1参数设置功能..... 2

2.2参数显示功能..... 2

2.3打开串口功能..... 2

2.4关闭串口功能..... 2

2.5接收数据功能..... 3

2.6发送数据功能..... 3

2.7清空数据接收区和发送区功能..... 3

|                      |    |
|----------------------|----|
| 3程序运行平台              |    |
| .....                |    |
| .....                | 3  |
| 4总体设计.....           |    |
| .....                | 4  |
| 5程序变量定义说明.....       |    |
| .....                | 4  |
| 6模块分析.....           |    |
| .....                | 5  |
| 6.1初始化模块             |    |
| .....                |    |
| ..... 5 6.2参数设置和显示模块 |    |
| .....                | 6  |
| 6.3发送数据模块.....       |    |
| .....                | 8  |
| 6.4数据接收模块.....       |    |
| ..... 9 6.5清空接数据模块   |    |
| .....                |    |
| . 10                 |    |
| 6.6关闭串口模块.....       |    |
| .....                | 10 |
| 7系统测试.....           |    |
| .....                | 11 |
| 7.1测试.....           |    |
| .....                | 11 |
| 7.2总结.....           |    |
| .....                | 16 |
| .....                |    |
| ..... 16 8结论.....    |    |

## 基于串口通信的高级语言控制程序的设计

### 摘 要

随着计算机的普及~ 计算机网络和多微机系统的广泛应用~ 尽管当今USB通信技术和以太网技术快速发展~ 然而串口通信技术始终以其连接简便~ 价格低廉~ 稳定可靠等优点~ 在计算机通信领域占有一席之地。MSC Omm控件是微软公司为了用户使用方便而提供的一种便捷的控制串口编程的解决方案~ 本文主要介绍了如何利用VC + + 2008 MSComm控件的串口通信方法~ 对程序的类进行了说明~ 分析了本程序各个模块怎样实现起作用。此次设计以MFC来设计界面编制程序,同时利用MSComm控件通过串口发送和接收数据,实现了串口通信。该串口通信控制程序具有通信参数设置、通信参数显示、打开和关闭串口、接收数据、发送数据、清空发送区和接收区数据的功能。

关键词:串口通信 visual c++ 6.0 MSComm控件



# 基于串口通信的高级语言控制程序的设计

## 1引言

### 1.1问题的提出

串行通信(Serial Communication

)在现代工业控制系统中一直有着及其重要的地位,作为一种基本而又灵活方便的通信方式,它被广泛应用于PC与PC或者PC与PLC之间的数据交换以及其他工业控制与自动控制中,它不仅没有因为时代的进步而遭淘汰,反而在规格上越来越完善,应用也越来越广泛。VC++是新一代面向对象的可视化开发工具,提供了良好的界面设计能力,在PC机的串口通信方面有很强的功能。对于从事机械行业的人来说,熟悉基于RS232串口通信的高级语言编程方法显得十分重要。

### 1.2国内外研究的现状

目前,计算机的串行通信应用十分广泛,串行接口已成为计算机的必需部件和

接口之一。由于其具有连接简单、使用灵活方便、数据传递可靠等优点,在工业监

控、数据采集和实时控制系统中得到了广泛应用。当今,实现串口通信的方法以及第

例如PComm控件,Windows

API函数等等。但是这几种方法都过于三方控件有很多,

复杂,对程序员要求较高,而且可靠性并不是很好。利用Microsoft推出的MSComm

控件,使得串口通信的实现更为简单、可靠。

随着当今信息技术的快速发展，通信已成为信息技术中的关键问题。尤其在控

制领域，以何种方式实现计算机与外围设备间既简单又可靠的通信，就显得特别重

要。由于串行通信具有实现简单、使用灵活方便、数据传输可靠等优点，因而广泛

应用于工业控制系统中，是计算机与外部设备进行数据通信时经常使用的方式之一。

VC++是新一代面向对象的可视化开发工具，提供了良好的界面设计能力，在PC机的

串口通信方面有很强的功能。它具有功能强大、简便易用和代码执行速度快等特点，

在通信软件的开发中成为越来越多开发人员的首选工具。无论在工业控制中PC机和

单片机之间的通信，还是在2台Pc机之间的串口通信，都具有相同的原理。只要掌

握其中的通信本质，就能灵活地实现串口通信，串口控制。

-1-



## 基于串口通信的高级语言控制程序的设计

## 1.5 任务与分析

### 1.5.1 主要任务

- ?制定串口通信控制程序的总体构架;
- ?熟悉VC平台下的高级语言编程方法;
- ?熟悉基于RS232串口通信的高级语言编程方法;
- ?在同一台PC机上调试通过所编写的程序;

### 1.5.2 分析

目前Win32 中基于VC + + 6.0

的常用串口通信程序一般可以用两种方法实现: ?利用MSComm  
ActiveX 串行通信控件; ?使用Win2dows API

通信函数。虽然在Windows下Win32API提供了使用文件操作方式的函数进行串口操作的方法,但是在编程实现上还是比较复杂。Windows平台先进的ActiveX技术使得对串口编程不再需要处理烦琐的细节。利用已有的AxtiveX控件,只需要编写少量的代码,就可以轻松高效地完成任务。因此程序员不需要花时间去了解复杂的API函数,只需要利用MSCom  
m控件就可实现串口编程。

## 2 程序的主要功能

### 2.1 参数设置功能

设置和修改串口通信参数,包括端口,波特率,奇偶校验位,数据位,停止位。 2.2参数显示功能

显示当前串口通信参数,包括端口号,波特率,奇偶校验位,数据位,停止位。 2.3打开串口功能

打开串口,实现通信。

## 2.4 关闭串口功能

关闭串口，停止通信。

-2-



# 基于串口通信的高级语言控制程序的设计

## 2.5 接收数据功能

能够进行单个或多个数据的接收和显示。

## 2.6 发送数据功能

能够进行单个或多个数据的发送。

## 2.7 清空数据接收区和发生区功能

删除接收区和发生区所有数据。

## 3 程序运行平台

Visual c++ 6.0

具体操作如下：

(1)新建MScComm项目，选择MFC应用程序，应用程序类型选择“基于对话框”，

生成的类中选择 “CMSCCommDlg” ；

(2)为MSComm项目添加新建一个Dialog设置对话框；

(3)加载MSComm控件，添加相应的源程序，加载必要的控件；

(4)编写程序代码，再编译，链接，调试，执行等，生成应用程序。

-3-



## 基于串口通信的高级语言控制程序的设计

### 4总体设计

图4.1系统总体框架图

主 函 数

参打发接清清

数开送收空空

设和数数接发

置关据 据 收送

和闭区 区

显串

示 口

## 5程序变量定义说明

控件名 程序中控件名 变量名 变量类型 串口选择组合框 IDC\_COMBO1 m\_td  
CComboBox 波特率选择组合框 IDC\_COMBO2 m\_btl CComboBox  
校检位组合框 IDC\_COMBO3 m\_xjw CComboBox 数据位组合框  
IDC\_COMBO4 m\_sjw CComboBox 停止位组合框 IDC\_COMBO5 m\_tzw  
CComboBox 接收编辑框 IDC\_EBIT1 m\_str\_recv CString 发送编辑框  
IDC\_EBIT2 m\_str\_send CString 复选框 IDC\_CHECK1 // COM控件  
IDC\_MSCOMM1 m\_msomm CMMSCComm

-4-



## 基于串口通信的高级语言控制程序的设计

### 6模块分析

#### 6.1初始化模块

代码如下:

```
BOOL CMyDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    m_msomm.SetInBufferSize(1024); //设置接收缓冲区的大小
    m_msomm.SetOutBufferSize(512); //设置发送缓冲区的大小
    if(m_msomm.GetPortOpen()) //判断串口是否打开
```

```
m_msomm.SetPortOpen(FALSE); //串口如果打开就关闭它

m_msomm.SetCommPort(2); //设置通道为COM2

if(!m_msomm.GetPortOpen()) //判断串口是否打开

m_msomm.SetPortOpen(TRUE); //如果没有打开就重新开启

else

AfxMessageBox("serial port open error");

m_msomm.SetInputMode(1); //设置以二进制方式接收

m_msomm.SetSettings("9600,n,8,1");
//设置波特率9600，无校验位，8位数据

位，1位停止位

m_msomm.SetRThreshold(1); //串口接收缓冲区有多于或等于一个字

符是产生接收数据的OnOnComm

m_msomm.SetInputLen(0); //设置当前接收区长度为零

m_msomm.GetInput(); //预读取缓冲区以清空残留数据

//组合框的初始化

m_td.SetCurSel(1);

m_btl.SetCurSel(5);

m_xjw.SetCurSel(2);

m_sjw.SetCurSel(0);

m_tzw.SetCurSel(0);

// Add "About..." menu item to system menu.

// IDM_ABOUTBOX must be in the system command range.
```

```

ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);

ASSERT(IDM_ABOUTBOX < 0xF000);

CMenu* pSysMenu = GetSystemMenu(FALSE);

if (pSysMenu != NULL)
{
    CString strAboutMenu;

    strAboutMenu.LoadString(IDS_ABOUTBOX);

    if (!strAboutMenu.IsEmpty())
    {

```

-5-



## 基于串口通信的高级语言控制程序的设计

```

pSysMenu->AppendMenu(MF_SEPARATOR);

pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
strAboutMenu);

}

}

```

## 6.2 参数设置和显示模块

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。  
如要下载或阅读全文，请访问：<https://d.book118.com/728062122107006057>