附录 E 第 18 章 VC 程序代码

```
// test_comDlg.h : header file
//
//{{AFX_INCLUDES()
#include "mscomm.h"
//}}AFX_INCLUDES
….   …..
class CTest_comDlg : public CDialog
{
// Construction
public:
    CTest_comDlg(CWnd* pParent = NULL);    // standard constructor

// Dialog Data
    //{{AFX_DATA(CTest_comDlg)
    enum { IDD = IDD_TEST_COM_DIALOG };
    CComboBox     m_ctrlComboxMoto;
    CComboBox     m_ctrlComboxHeat;
    CMSComm       m_ctrlComm;
    CString    m_strStat;
    int        m_nDown;
    int        m_nUp;
    CString    m_strTemper;
    COleDateTime  m_oleDate;
    COleDateTime  m_oleTime;
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CTest_comDlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
    //{{AFX_MSG(CTest_comDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
```

```
	afx_msg void OnOnCommMscomm1();
	afx_msg void OnBtnReadTime();
	afx_msg void OnBtnSetTime();
	afx_msg void OnBtnReadTemper();
	afx_msg void OnBtnReadUpdown();
	afx_msg void OnBtnSetUpdown();
	afx_msg void OnBtnSetMoto();
	afx_msg void OnBtnSetHeat();
	afx_msg void OnBtnReadMoto();
	afx_msg void OnBtnReadHeat();
	afx_msg void OnDestroy();
	DECLARE_EVENTSINK_MAP()
	//}}AFX_MSG
	DECLARE_MESSAGE_MAP()
private:
	void DealProtocol();
	int m_nRecvLen;
	byte *m_pBuff;
};
```

```
// test_comDlg.cpp : implementation file
//

#include "stdafx.h"
#include "test_com.h"
#include "test_comDlg.h"
/////////////////////////////////////////////////////////////////////////////
// CTest_comDlg dialog
//该函数的内容在ClassWizard中添加成员变量、设定初始值时，系统会自动添加，不需要我们去编写
CTest_comDlg::CTest_comDlg(CWnd* pParent /*=NULL*/)
	: CDialog(CTest_comDlg::IDD, pParent)
{
	//{{AFX_DATA_INIT(CTest_comDlg)
	m_strStat = _T("");
	m_nDown = 0;
	m_nUp = 0;
	m_strTemper = _T("");
	m_oleDate = COleDateTime::GetCurrentTime();
	m_oleTime = COleDateTime::GetCurrentTime();
	//}}AFX_DATA_INIT
	// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
```

```
//该函数的内容在ClassWizard中将控件绑定到成员变量时，系统自动添加，不需要我们编写
void CTest_comDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CTest_comDlg)
	DDX_Control(pDX, IDC_COMBO_MOTO, m_ctrlComboxMoto);
	DDX_Control(pDX, IDC_COMBO_HEAT, m_ctrlComboxHeat);
	DDX_Control(pDX, IDC_MSCOMM1, m_ctrlComm);
	DDX_Text(pDX, IDC_EDIT_STAT, m_strStat);
	DDX_Text(pDX, IDC_EDIT_DOWN, m_nDown);
	DDV_MinMaxInt(pDX, m_nDown, -55, 100);
	DDX_Text(pDX, IDC_EDIT_UP, m_nUp);
	DDV_MinMaxInt(pDX, m_nUp, -10, 120);
	DDX_Text(pDX, IDC_EDIT_TEMPER, m_strTemper);
	DDX_DateTimeCtrl(pDX, IDC_DATETIMEPICKER_DATE, m_oleDate);
	DDX_DateTimeCtrl(pDX, IDC_DATETIMEPICKER_TIME, m_oleTime);
	//}}AFX_DATA_MAP
}

//这部分内容是我们双击界面按钮添加函数时，系统自动添加的映射
BEGIN_MESSAGE_MAP(CTest_comDlg, CDialog)
	//{{AFX_MSG_MAP(CTest_comDlg)
	ON_WM_SYSCOMMAND()
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	ON_BN_CLICKED(IDC_BTN_READ_TIME, OnBtnReadTime)
	ON_BN_CLICKED(IDC_BTN_SET_TIME, OnBtnSetTime)
	ON_BN_CLICKED(IDC_BTN_READ_TEMPER, OnBtnReadTemper)
	ON_BN_CLICKED(IDC_BTN_READ_UPDOWN, OnBtnReadUpdown)
	ON_BN_CLICKED(IDC_BTN_SET_UPDOWN, OnBtnSetUpdown)
	ON_BN_CLICKED(IDC_BTN_SET_MOTO, OnBtnSetMoto)
	ON_BN_CLICKED(IDC_BTN_SET_HEAT, OnBtnSetHeat)
	ON_BN_CLICKED(IDC_BTN_READ_MOTO, OnBtnReadMoto)
	ON_BN_CLICKED(IDC_BTN_READ_HEAT, OnBtnReadHeat)
	ON_WM_DESTROY()
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CTest_comDlg message handlers
//灰色的内容是系统自动添加的
BOOL CTest_comDlg::OnInitDialog()
{
	CDialog::OnInitDialog();
```

```
    // 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())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // Set the icon for this dialog.  The framework does this automatically
    //  when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);            // Set big icon
    SetIcon(m_hIcon, FALSE);       // Set small icon

    // TODO: Add extra initialization here
    m_ctrlComboxHeat.SetCurSel(0);
    m_ctrlComboxMoto.SetCurSel(0);
    m_nDown = 10;
    m_nUp = 28;
    m_strTemper = "";
    m_strStat = "";
    UpdateData(FALSE);

    m_nRecvLen = 0;
    m_pBuff = new byte[100];

    if(m_ctrlComm.GetPortOpen())           //如果串口是打开的，则行关闭串口
    {
        m_ctrlComm.SetPortOpen(FALSE);
    }
    m_ctrlComm.SetCommPort(2);             //选择 COM2
    m_ctrlComm.SetInBufferSize(1024);      //接收缓冲区
    m_ctrlComm.SetOutBufferSize(1024);     //发送缓冲区
    m_ctrlComm.SetInputLen(0);             //设置当前接收区数据长度为 0,表示全部读取
```

```
        m_ctrlComm.SetInputMode(1);            //以二进制方式读写数据
        m_ctrlComm.SetRThreshold(1);           //接收缓冲区有 1 个及 1 个以上字符时，将引发接收数据的
OnComm 事件
        m_ctrlComm.SetSettings("19200,n,8,1");//波特率 9600 无检验位，8 个数据位，1 个停止位
        if(!m_ctrlComm.GetPortOpen())          //如果串口没有打开则打开
        {
                m_ctrlComm.SetPortOpen(TRUE);//打开串口
        }else
        {
                m_ctrlComm.SetOutBufferCount(0);
                AfxMessageBox("Open The Serial COM4 Fail!");
        }

        return TRUE;   // return TRUE   unless you set the focus to a control
}

//系统自动添加的函数
void CTest_comDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
        if ((nID & 0xFFF0) == IDM_ABOUTBOX)
        {
                CAboutDlg dlgAbout;
                dlgAbout.DoModal();
        }
        else
        {
                CDialog::OnSysCommand(nID, lParam);
        }
}

// If you add a minimize button to your dialog, you will need the code below
//   to draw the icon.   For MFC applications using the document/view model,
//   this is automatically done for you by the framework.

void CTest_comDlg::OnPaint()
{
        if (IsIconic())
        {
                CPaintDC dc(this); // device context for painting

                SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

                // Center icon in client rectangle
                int cxIcon = GetSystemMetrics(SM_CXICON);
```

```
		int cyIcon = GetSystemMetrics(SM_CYICON);
		CRect rect;
		GetClientRect(&rect);
		int x = (rect.Width() - cxIcon + 1) / 2;
		int y = (rect.Height() - cyIcon + 1) / 2;

		// Draw the icon
		dc.DrawIcon(x, y, m_hIcon);
	}
	else
	{
		CDialog::OnPaint();
	}
}

// The system calls this to obtain the cursor to display while the user drags
//  the minimized window.
HCURSOR CTest_comDlg::OnQueryDragIcon()
{
	return (HCURSOR) m_hIcon;
}

BEGIN_EVENTSINK_MAP(CTest_comDlg, CDialog)
    //{{AFX_EVENTSINK_MAP(CTest_comDlg)
	ON_EVENT(CTest_comDlg, IDC_MSCOMM1, 1 /* OnComm */, OnOnCommMscomm1, VTS_NONE)
	//}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()

//串口接收数据
void CTest_comDlg::OnOnCommMscomm1()
{
	// TODO: Add your control notification handler code here
	VARIANT varInput;
	COleSafeArray arrayInput;
	long size = 0, k;
	unsigned char recvData[1024];	//设置 BYTE 数组

	int event = m_ctrlComm.GetCommEvent();
	switch(event)
	{
	case 1:			// comEvSend 发送数据
		break;
	case 2:			// comEvReceive 读取数据
		varInput = m_ctrlComm.GetInput();	//读缓冲区
```

```
        arrayInput = varInput;              //VARIANT 型变量转换为 ColeSafeArray 型变量
        size = arrayInput.GetOneDimSize();      //得到有效数据长度
        //判断接收数据是否够一帧
        if(size < 1)
        {
            break;
        }

        for(k=0; k<size; k++)
        {
            arrayInput.GetElement(&k, recvData+k);      //提取接收字符
            if(k<0)
                break;
            //判断数据头
            if(recvData[0] == 0xEB && m_nRecvLen == 0)
            {
                m_pBuff[m_nRecvLen] = 0xEB;
                m_nRecvLen ++;
            }else if(m_nRecvLen != 0){
                m_pBuff[m_nRecvLen] = recvData[k];
                m_nRecvLen ++;
            }
        }

        DealProtocol();         //处理通信数据
        break;
    default:                    // 传输事件出错
        m_ctrlComm.SetOutBufferCount(0);
        break;
    }
    UpdateData(FALSE);          //更新界面显示
}

//我们自己的协议处理函数
void CTest_comDlg::DealProtocol()
{
    int num = 0;
    int year = 0;
    int month = 0;
    int day = 0;
    char int_tmp, dec_tmp;
    int tmp = 0;

    num = m_pBuff[1];
```

```
if(m_nRecvLen < 6)
{
        return;
}

if(m_nRecvLen < num)
{
        return;
}
m_nRecvLen = 0;

if(m_pBuff[num-1] != 0x90)
{
        m_strStat = "尾字符错误！";
        return;
}

//判断校验和
unsigned char chkSum = 0;
for(int i=1; i<num-2; i++)
{
        chkSum += m_pBuff[i];
}
if(chkSum != m_pBuff[num-2])
{
        m_strStat = "校验和错误！";
        return;
}

//解析协议
if(m_pBuff[3] == 1)    //设置
{
        switch(m_pBuff[2])
        {
        case 0x02:
                if(m_pBuff[4] == 0)
                        m_strStat = "设置温度上下限成功！";
                else
                        m_strStat = "设置温度上下限失败！";
                break;
        case 0x03:
                if(m_pBuff[4] == 0)
                        m_strStat = "设置系统时间成功！";
                else
```

```
            m_strStat = "设置系统时间失败！";
        break;
    case 0x04:
        if(m_pBuff[4] == 0)
            m_strStat = "控制风扇成功！";
        else
            m_strStat = "控制风扇失败！";
        break;
    case 0x05:
        if(m_pBuff[4] == 0)
            m_strStat = "控制加热器成功！";
        else
            m_strStat = "控制加热器失败！";
        break;
    default :
        break;
    }
}else
{
    switch(m_pBuff[2])
    {
    case 0x01:
        int_tmp = m_pBuff[5];
        dec_tmp = m_pBuff[6];
        if((int_tmp&0x80) != 0)
        {
            dec_tmp = 0x0F-dec_tmp;//04->0B
            //int_tmp = ~int_tmp;
            dec_tmp++;
            if(dec_tmp == 0x10)
            {
                int_tmp -= 1;
                dec_tmp = 0;
            }
        }
        tmp = dec_tmp *625;
        if((int_tmp&0x80) != 0)
        {
            int_tmp = ~int_tmp;
            m_strTemper.Format("-%d.%d", int_tmp, tmp);
        }else
        {
            m_strTemper.Format("%d.%d", int_tmp, tmp);
        }
        m_strStat = "读取当前温度成功！";
        break;
    case 0x02:
        if((m_pBuff[5]&0x80) != 0)
        {
            m_pBuff[5] = (~m_pBuff[5]) + 1;
            m_nDown = 0-m_pBuff[5];
        }else{
            m_nDown = m_pBuff[5];
        }

        if((m_pBuff[6]&0x80) != 0)
        {
            m_pBuff[6] = (~m_pBuff[6]) + 1;
            m_nUp = 0-m_pBuff[6];
        }else{
            m_nUp = m_pBuff[6];
        }

        if(m_pBuff[4] == 0)
            m_strStat = "读取温度上下限成功！";
        else
            m_strStat = "读取温度上下限失败！";
        break;
    case 0x03:
        year = 2000+m_pBuff[5];
        month = m_pBuff[6];
        day = m_pBuff[7];

        m_oleDate.SetDateTime(year, month, day, (int)m_pBuff[8], (int)m_pBuff[9], (int)m_pBuff[10]);
        m_oleTime.SetTime(m_pBuff[8], m_pBuff[9], m_pBuff[10]);
        if(m_pBuff[4] == 0)
            m_strStat = "读取系统时间成功！";
        else
            m_strStat = "读取系统时间失败！";
        break;
    case 0x04:
        m_ctrlComboxMoto.SetCurSel(m_pBuff[5]);
        if(m_pBuff[4] == 0)
            m_strStat = "读取风扇状态成功！";
        else
            m_strStat = "读取风扇状态失败！";
        break;
    case 0x05:
            m_ctrlComboxHeat.SetCurSel(m_pBuff[5]);
            if(m_pBuff[4] == 0)
                m_strStat = "读取加热器状态成功！";
            else
                m_strStat = "读取加热器状态失败！";
            break;
        default :
            break;
        }
    }
}

//读时间按钮
void CTest_comDlg::OnBtnReadTime()
{
    // TODO: Add your control notification handler code here
    int i = 0;
    unsigned char pSendBuff[] = {0xEB,0x06,0x03,0x00,0x09,0x90};
    CByteArray byteArray;

    byteArray.RemoveAll();
    byteArray.SetSize(6);

    for(i =0; i<6; i++)
    {
       byteArray.SetAt(i, pSendBuff[i]);
    }

    m_ctrlComm.SetOutput(COleVariant(byteArray));
}

//设定时间按钮
void CTest_comDlg::OnBtnSetTime()
{
    // TODO: Add your control notification handler code here
    int i = 0;
    unsigned char pSendBuff[13] = {0xEB,13,0x03,0x01,0x08,0x90};
    CByteArray byteArray;

    pSendBuff[0] = 0xEB;
    pSendBuff[12] = 0x90;
    pSendBuff[1] = 13;//len
    pSendBuff[2] = 0x03;//cmd
    pSendBuff[3] = 0x01;//set
    UpdateData(TRUE);
    pSendBuff[4] = m_oleDate.GetYear()-2000;
    pSendBuff[5] = m_oleDate.GetMonth();
    pSendBuff[6] = m_oleDate.GetDay();
    pSendBuff[10] = m_oleDate.GetDayOfWeek();

    pSendBuff[7] = m_oleTime.GetHour();
    pSendBuff[8] = m_oleTime.GetMinute();
    pSendBuff[9] = m_oleTime.GetSecond();
    pSendBuff[11] = 0;
    for(i=1; i<11; i++)
    {
            pSendBuff[11] += pSendBuff[i];
    }

    byteArray.RemoveAll();
    byteArray.SetSize(13);

    for(i =0; i<13; i++)
    {
      byteArray.SetAt(i, pSendBuff[i]);
    }

    m_ctrlComm.SetOutput(COleVariant(byteArray));
}

//读温度按钮
void CTest_comDlg::OnBtnReadTemper()
{
    // TODO: Add your control notification handler code here
    int i = 0;
    unsigned char pSendBuff[] = {0xEB,0x06,0x01,0x00,0x07,0x90};
    CByteArray byteArray;

    byteArray.RemoveAll();
    byteArray.SetSize(6);

    for(i =0; i<6; i++)
    {
      byteArray.SetAt(i, pSendBuff[i]);
    }

    m_ctrlComm.SetOutput(COleVariant(byteArray));
```

```
}

//读温度上下限按钮
void CTest_comDlg::OnBtnReadUpdown()
{
        // TODO: Add your control notification handler code here
        int i = 0;
        unsigned char pSendBuff[] = {0xEB,0x06,0x02,0x00,0x08,0x90};
        CByteArray byteArray;

        byteArray.RemoveAll();
        byteArray.SetSize(6);

        for(i =0; i<6; i++)
        {
             byteArray.SetAt(i, pSendBuff[i]);
        }

        m_ctrlComm.SetOutput(COleVariant(byteArray));
}

//设定温度上下限按钮
void CTest_comDlg::OnBtnSetUpdown()
{
        // TODO: Add your control notification handler code here
        int i = 0;
        unsigned char pSendBuff[8] = {0xEB,8,0x02,0x01,0x08,0x90};
        CByteArray byteArray;

        pSendBuff[0] = 0xEB;
        pSendBuff[7] = 0x90;
        pSendBuff[1] = 8;//len
        pSendBuff[2] = 0x02;//cmd
        pSendBuff[3] = 0x01;//set

        UpdateData(TRUE);
        pSendBuff[4] = m_nDown;
        pSendBuff[5] = m_nUp;

        pSendBuff[6] = 0;
        for(i=1; i<6; i++)
        {
                pSendBuff[6] += pSendBuff[i];
        }
```

```
    byteArray.RemoveAll();
    byteArray.SetSize(8);

    for(i =0; i<8; i++)
    {
        byteArray.SetAt(i, pSendBuff[i]);
    }

    m_ctrlComm.SetOutput(COleVariant(byteArray));
}

//设定风扇控制按钮
void CTest_comDlg::OnBtnSetMoto()
{
    // TODO: Add your control notification handler code here
    int i = 0;
    unsigned char pSendBuff[7] = {0xEB,7,0x04,0x01,0x00, 0x08,0x90};
    CByteArray byteArray;

    UpdateData(TRUE);
    pSendBuff[4] = m_ctrlComboxMoto.GetCurSel();

    pSendBuff[5] = 0;
    for(i=1; i<5; i++)
    {
         pSendBuff[5] += pSendBuff[i];
    }

    byteArray.RemoveAll();
    byteArray.SetSize(7);

    for(i =0; i<7; i++)
    {
        byteArray.SetAt(i, pSendBuff[i]);
    }

    m_ctrlComm.SetOutput(COleVariant(byteArray));
}

//设定加热控制按钮
void CTest_comDlg::OnBtnSetHeat()
{
    // TODO: Add your control notification handler code here
    int i = 0;
    unsigned char pSendBuff[7] = {0xEB,7,0x05,0x01,0x00, 0x08,0x90};
    CByteArray byteArray;

    UpdateData(TRUE);
    pSendBuff[4] = m_ctrlComboxHeat.GetCurSel();

    pSendBuff[5] = 0;
    for(i=1; i<5; i++)
    {
        pSendBuff[5] += pSendBuff[i];
    }

    byteArray.RemoveAll();
    byteArray.SetSize(7);

    for(i =0; i<7; i++)
    {
      byteArray.SetAt(i, pSendBuff[i]);
    }

    m_ctrlComm.SetOutput(COleVariant(byteArray));
}

//读取风机状态按钮
void CTest_comDlg::OnBtnReadMoto()
{
    // TODO: Add your control notification handler code here
    int i = 0;
    unsigned char pSendBuff[] = {0xEB,0x06,0x04,0x00,0x0A,0x90};
    CByteArray byteArray;

    byteArray.RemoveAll();
    byteArray.SetSize(6);

    for(i =0; i<6; i++)
    {
      byteArray.SetAt(i, pSendBuff[i]);
    }

    m_ctrlComm.SetOutput(COleVariant(byteArray));
}

//读取加热系统状态按钮
```

```
void CTest_comDlg::OnBtnReadHeat()
{
        // TODO: Add your control notification handler code here
        int i = 0;
        unsigned char pSendBuff[] = {0xEB,0x06,0x05,0x00,0x0B,0x90};
        CByteArray byteArray;

        byteArray.RemoveAll();
        byteArray.SetSize(6);

        for(i =0; i<6; i++)
        {
             byteArray.SetAt(i, pSendBuff[i]);
        }

        m_ctrlComm.SetOutput(COleVariant(byteArray));
}

//窗口注销函数，通过 ClassWizard 添加，主要是释放我们申请的资源
void CTest_comDlg::OnDestroy()
{
        CDialog::OnDestroy();

        // TODO: Add your message handler code here
        delete m_pBuff;          //释放资源
        m_pBuff = NULL;
}
```