

USB-8255 卡安装使用说明

1. 安装驱动程序

- 1-1 安装驱动程序
- 1-2 确认驱动程序的安装

2. USB-8255 卡设置

- 2-1 运行 USB-8255 演示程序
- 2-2 设置 COM 口
- 2-3 设置芯片 (U2, U3) 的输出和输入端口 (PA, PB, PC)
- 2-4 如何由下位机端口 (PA, PB, PC) 读出数据并在上位机显示
- 2-5 如何从上位机写入数据到下位机端口 (PA, PB, PC)

3. USB-8255 程序简单分析

4. USB-8255 上位机自检程序分析

1. 驱动程序

确保已经成功下载相关文件。

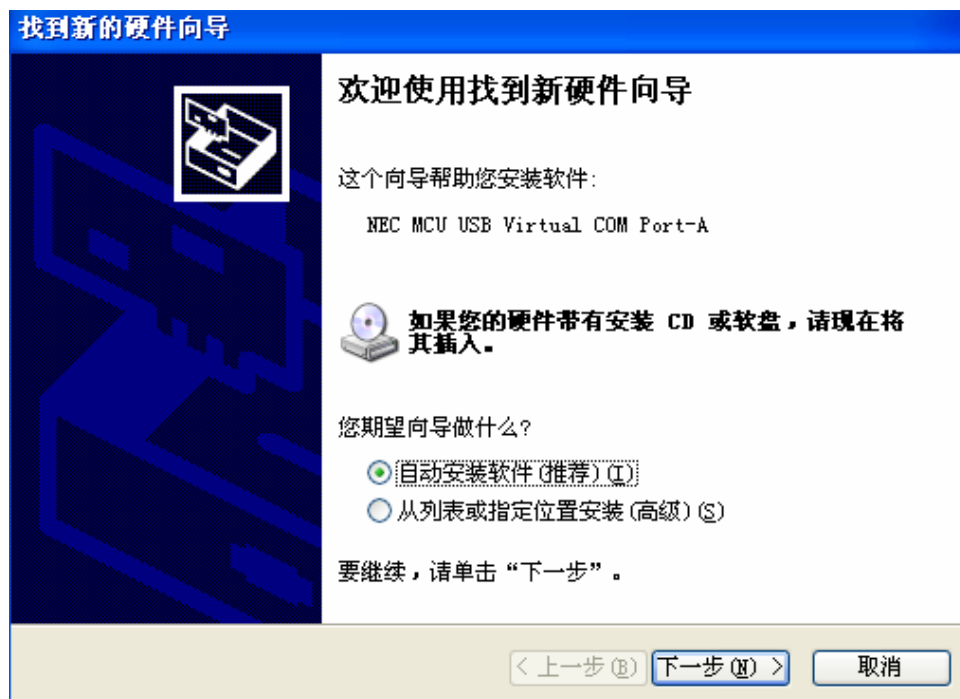
在 “Driver” 文件夹中，包括下列文件。

NEC_MCU_USB.inf
NEC_MCU_USB_DV.sys
NEC_MCU_USB_VC.sys

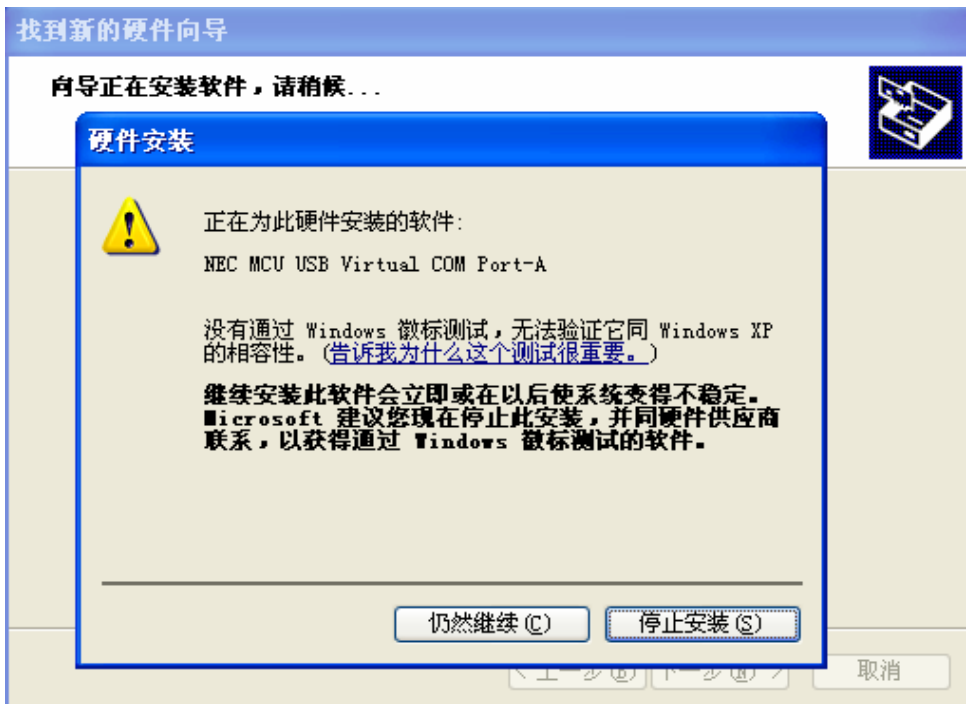
注：所有软件都支持 Windows 2000 或 Windows XP。

1-1 安装驱动程序

通过 USB 连接器将 USB-8255 卡连接至 PC。当操作系统识别到 USB 设备时，将显示下列对话框。（下列示例针对 Windows xp）



点击” 下一步”

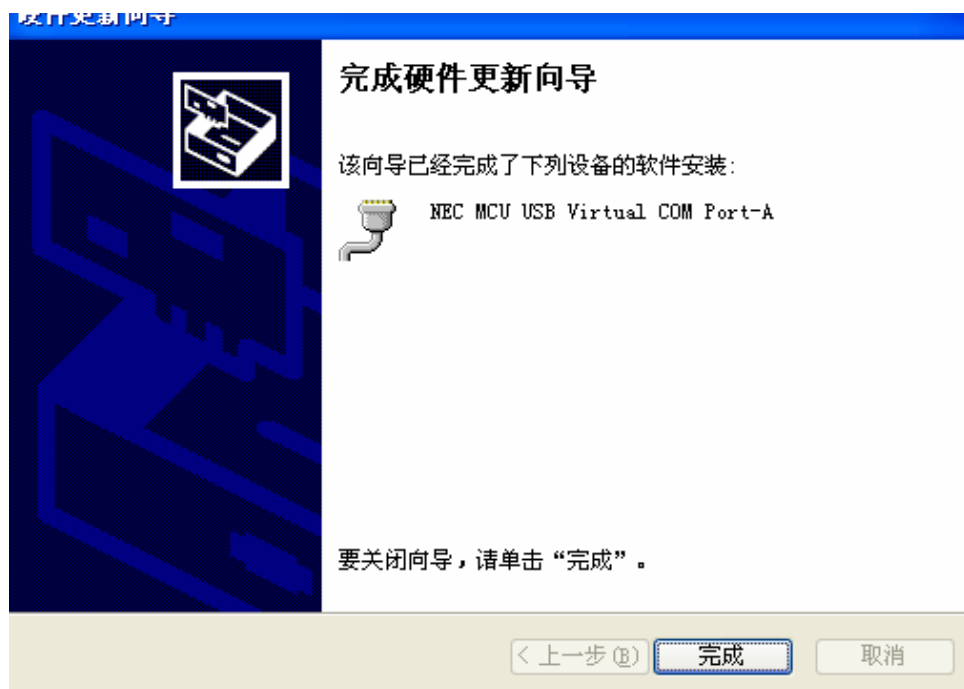


选择”仍然继续，



指定驱动文件路径，然后点击 “确定”。

(注：上图中，d:\fm\driver 是指定的文件夹，而不是一个示例。)



点击“完成”结束驱动程序安装。并重新启动 PC，确保驱动有效。

1-2 确认驱动程序的安装

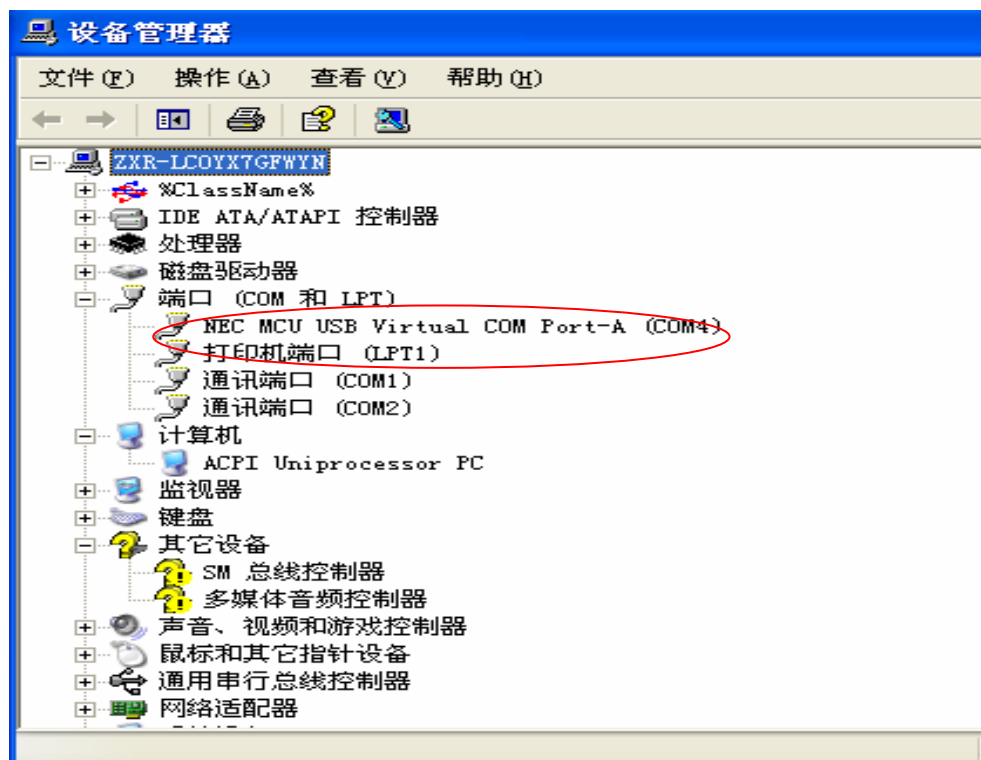
在“我的电脑”上点击右键，选择“属性”，



选择“硬件”标签，然后点击“设备管理器”。



如果你能够在“设备管理器”中的“端口 (Com 和 LPT)”中看到“NEC MCU USB Virtual COM Port-A (COMx)”，则驱动程序已经被正确安装。（这里，“COMx”中的“x”代表COM口的端口号，依赖于系统配置。请记住这个端口号，因为在“8255卡设置程序”中必须指定这个端口号。）



2. USB-8255 卡设置

2-1 运行 USB-8255 演示程序

首先，将 8255 卡 JP1 的 2-3 脚短接，然后连接至 PC 机的 USB 端口。双击保存在 Debug 文件夹（已经下载的）中的 8255USB.exe 文件。你将看到下列显示的画面。



演示程序

2-2 设置 COM 口



，将”设备管理器“中显示的“COMx”中‘x’输入“连接端口号“旁边的文本框中。

2-3 设置芯片 (U2, U3) 的输出和输入端口 (PA, PB, PC)


1. 设置芯片 U2, U3 的 PA, PB, PC 口为写入:

先设置好”连接端口号”，然后根据你所要写入的芯片 (U2, U3) 选择输入端口 (PA, PB, PC)，在它们对应的文本框中输入数字或者二进制中打勾，它们的改变是同步进行的。选择”写入”，即可写入。如下列所示:

芯片 U2 的 PA 口输入



在演示程序中，

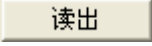
如果你要设置芯片 U2 的 PA 口为输入，在下位机数据线连接后，你只须在 U2A 的二进制框中打勾或者文本框输入数字 (0--255)，然后点击 U2A 文本框旁边的写入按钮 ，即可将你所输入的数字在下位机中以高低电平的形式表示出来。

2. 设置芯片 U2, U3 的 PA, PB, PC 口为输入:

先设置好”连接端口号”，然后根据你所要读出的芯片 (U2, U3) 选择输出端口 (PA, PB, PC), 选择”读出”，结果显示在它对应的文本框和二进制框中，如图所示:



如芯片 U2 的 PA 口输出

如果你要设置芯片 U2 的 PA 口为输出，在下位机数据线连接后，点击对应的  按钮，即可在 U1A 文本框和二进制框中显示输出结果。

2-4 如何由下位机端口 (PA, PB, PC) 读出数据并在上位机显示 (VC++代码)

1. 由 U2PA 口读入数据并显示出来:

```
usb->read_data(0x04, &buff);  
m_U2PA_EDIT=buff;
```

2. 由 U2PB 口读入数据并显示出来:

```
usb->read_data(0x14, &buff);  
m_U2PB_EDIT=buff;
```

3. 由 U2PC 口读入数据并显示出来:

```
usb->read_data(0x24, &buff);  
m_U2PC_EDIT=buff;
```

4. 由 U3PA 口读入数据并显示出来:

```
usb->read_data(0xc4, &buff);  
m_U3PA_EDIT=buff;
```

5. 由 U3PB 口读入数据并显示出来:

```
usb->read_data(0xD4, &buff);  
m_U3PB_EDIT =buff;
```

6. 由 U3PC 口读入数据并显示出来:

```
usb->read_data(0xE4, &buff);  
m_U3PC_EDIT =buff;
```

2-5 如何从上位机写入数据到下位机端口 (PA, PB, PC)

1. 写入数据到下位机的 U2PA 口并显示出来:

```
usb->write_data(0x08, m_U2PA_EDIT);
```

2. 写入数据到下位机的 U2PB 口并显示出来:

```
usb->write_data(0x18, m_U2PB_EDIT);
```

3. 写入数据到下位机的 U2PC 口并显示出来:

```
usb->write_data(0x28, m_U2PC_EDIT);
```

4. 写入数据到下位机的 U3PA 口并显示出来:

```
usb->write_data(0xC8, m_U3PA_EDIT);
```

5. 写入数据到下位机的 U3PB 口并显示出来:

```
usb->write_data(0xD8, m_U3PB_EDIT);
```

6. 写入数据到下位机的 U3PC 口并显示出来:

```
usb->write_data(0xE8, m_U3PC_EDIT);
```

3. USB-8255 程序简单分析

USB-8255 演示程序的核心在 operate 类中, 它包含了 ComOpen(BYTE Comport), ComOpen(BYTE Comport), write_data(unsigned char adr, unsigned char data), read_data(unsigned char adr, BYTE *data).

```
bool CUsbOperate::ComOpen(BYTE Comport) //打开 com 口
{
    CString strCom;
    strCom.Format("\\\\. \\COM%d", Comport);
    hCom=CreateFile(strCom, GENERIC_READ|GENERIC_WRITE, 0/*FILE_SHARE_READ|FILE_SHARE_WRITE*/, NULL, OPEN_EXISTING, NULL, NULL);
    if( hCom == INVALID_HANDLE_VALUE ){
        return FALSE;}

#ifdef 1
        COMMTIMEOUTS timeouts;
        timeouts.ReadIntervalTimeout =500;
        timeouts.ReadTotalTimeoutConstant=100;
        timeouts.ReadTotalTimeoutMultiplier=100;
        SetCommTimeouts(hCom, &timeouts);
#endif

    return TRUE;
}

bool CUsbOperate::ComClose()//关闭 com 口
{
    if( hCom != INVALID_HANDLE_VALUE ){
        CloseHandle(hCom);
        return true;
        hCom = INVALID_HANDLE_VALUE;
    }

    else

        return false;
}

写操作函数:
bool CUsbOperate::write_data(unsigned char adr, unsigned char data)//传入 adr 写入命令和 data
{
    BYTE cmd[16];
    DWORD dwWritten;

    cmd[0] = 0xAA;
    cmd[1] = 0x03;
    cmd[2] = 0x01;
    cmd[3] = adr;
```

```
cmd[4] = data;
```

```
if( !WriteFile(hCom, cmd, 5, &dwWritten, NULL) ){  
    return false;  
}  
else  
    return true;  
}
```

读操作函数:

```
bool CUsbOperate::read_data(unsigned char adr, BYTE *data)//传入 adr 输出命令,返回*data  
{  
    BYTE cmd[16], rbuff[16];  
    DWORD dwWritten, dwRead;  
    BYTE tmp;  
    cmd[0] = 0xAA;  
    cmd[1] = 0x01;  
    cmd[2] = 0x02;  
    cmd[3] = adr;  
    if(!WriteFile(hCom, cmd, 4, &dwWritten, NULL)){  
        return false;  
    }  
    if(!ReadFile(hCom, rbuff, 5, &dwRead, NULL) ){  
        return false;  
    }  
    memcpy(data, rbuff, 5);  
    tmp=data[4];  
    *data=tmp;  
    return true;  
}
```

写入下位机的操作函数为 : write_data(unsigned char adr, unsigned char data);

例如: 写入下位机的 U2PB 口: usb->write_data(0x18, m_U2PB_EDIT); //usb 是 operate 实例化

由下位机读数据的操作函数为: read_data(unsigned char adr, BYTE *data);

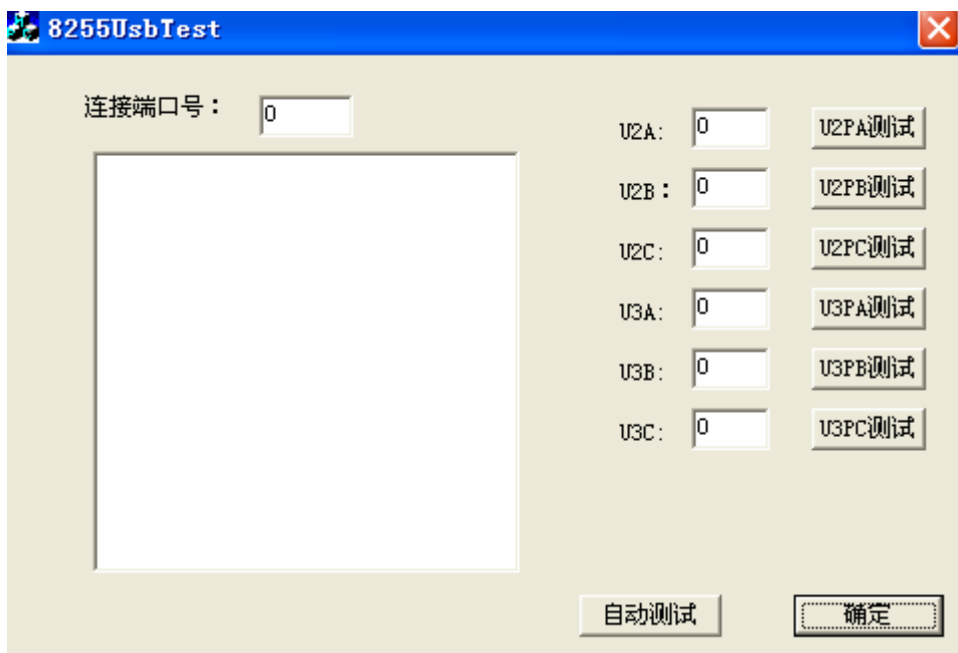
例如: 由下位机的 U2PB 口读: usb->read_data(0x14, &buff);

```
m_U2PB_EDIT =buff; //显示值
```

对下位机的读和写操作都可以直接调用上面的读和写函数。

4. USB-8255 上位机自检程序分析

USB-8255 上位机自检程序, 主要是通过循环的写入和读书来检测 U2, U3 的 PA, PB, PC 口是否正常。首先, 将 8255 卡 JP1 的 2-3 脚短接, 然后连接至 PC 机的 USB 端口。双击保存在 8255UsbTest/Debug 文件夹(已经下载的)中的 8255UsbTest.exe 文件。你将看到下列显示的画面。



上位机自检程序

在下位机线连接好和输入“端口号”以后, 点击“自动测试”即可。测试结果显示在左边的显示框中。

如: U2PA Write and U3PA Read Test OK

U2PA 测试—U3PC 测试 都是自动读入信号源的操作, 选择后, 即可将数据显示在左边的文本框和显示框中(在没有接入信号源时也可以读出默认的数字)

如: U2PA Read 255 OK!