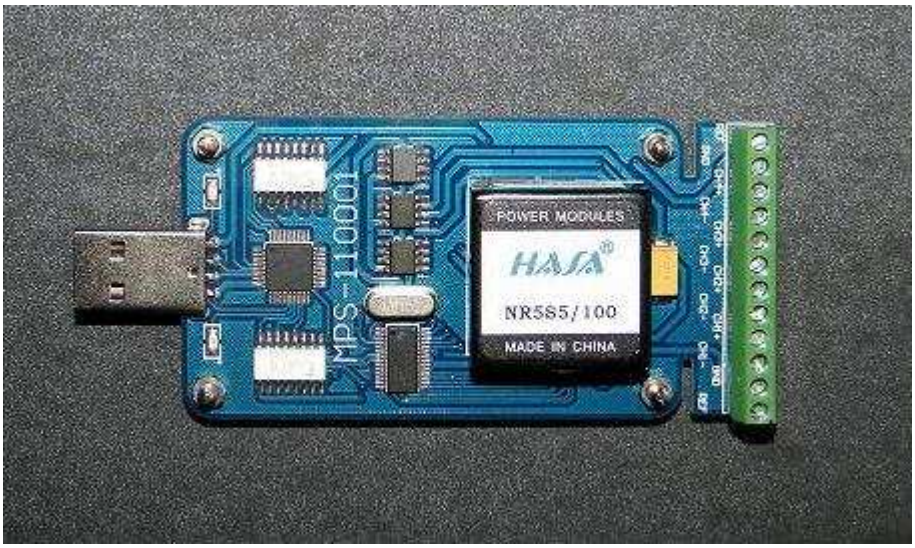
MPS-110001

24 位带隔离微弱信号采集卡

使用说明

Ver 1.1

第一章 产品概述



MPS-110001 24 位带隔离微弱信号采集卡

一、 *产品简介*

MPS-110001 数据采集卡是一款基于 USB 总线的带隔离的 24 位微弱信号采集卡。其信号端与通信端有全面隔离，从而允许接入特殊的高共模电压信号。板卡有 4 路信号输入通道，采样噪声峰峰值仅为 2 微伏，真正实现了百万分之一的超高分辨率。板卡内置有 32 倍和 64 倍两档可编程的增益放大器，在不经过外部放大的情况下可以直接测量 mV 级的微弱信号。本卡适用于实验室信号分析、产品质量检测和大专院校教学科研等各种领域，尤其在微弱信号采集方面具有无与伦比的优势。

MPS-110001 采用 USB2.0 全速总线接口，兼容 USB1.1 接口。总线极具易用性，即插即用，是便携式系统用户的最佳选择，可以完全取代以往的 PCI 卡。

MPS-110001 可工作在 Win9X/Me、Win2000/XP 等常用操作系统中，并提供可供 VB, VC, C++Builder, Dephi，LabVIEW，Matlab 等常用编程语言调用的动态链接库，编程函数接口简单易用，易于编写应用程序。

二、 *性能指标*

2.1、USB 总线性能

- USB2.0 全速总线传输
- 向下兼容 USB1.1 接口
- 使用方便，能够实现自动配置，支持设备的热插拔即插即用

2.2、模拟信号输入

- 输入通道: 1 路差分或 4 路差分(扫描)
- 输入量程： 差分±1V
- 端口耐压： 0V—4V(PGA = 1),1.5V-2.5V(PGA = 32 或 64)
- 输入阻抗: 100M 欧姆(PGA = 1)
- 采样率： 单通道：10sps 或 80sps；四通道：0.6sps 或 4.5sps(每通道)
- 可编程增益： 0.5,1，32，64
- 有效分辨率： 23.1bit(PGA = 1,10sps)；21.5(PGA = 1，80sps)；21.1(PGA = 64，10sps)；17.2(PGA = 64，80sps)
- 噪声峰峰值： 2uV(PGA = 1,10sps)；8uV(PGA = 1,80sps)；200nV(PGA = 64，

10sps)；500nV(PGA = 64，80sps)

- 非线性误差： ±0.0002%
- 偏移误差： ±0.2ppm
- 温漂： 0.3μV/℃

2.3、隔离电压

- 1000Vrms

2.4、工作温度

- 0℃ - 70℃

三、 应用领域

微弱信号采集

隔离数据采集

便携式仪表和测试设备

传感器信号采集与分析

四、 软件支持

提供 Windows95/98/NT/2000 下的驱动程序，提供通用 DLL 文件，并提供在 LabVIEW 图像化语言编写的应用软件范例程序。

五、 销售清单

[1] MPS-110001 USB 总线数据采集卡一张；

[2] 高屏蔽 USB 数据传输电缆一根；

[3] 保修卡一张；

六、 售后服务

一年内免费保修，终身维修。

第二章 设备安装

一、 *MPS-110001 数据采集卡硬件接口说明*

GND： 采集卡地线端口

REF： 参考电压输出(2.048V)

CHx+: 模拟信号差分输入正输入端口

CHx-: 模拟信号差分输入负输入端口

二、 *MPS-110001 数据采集卡指示灯状态说明*

指示灯状态由软件编程决定，用户自定义。上电时绿色 LED 作为设备自检状态指示(自检成功 LED 亮)，自检完成后绿色 LED 可由用户编程控制。

三、 *驱动安装*

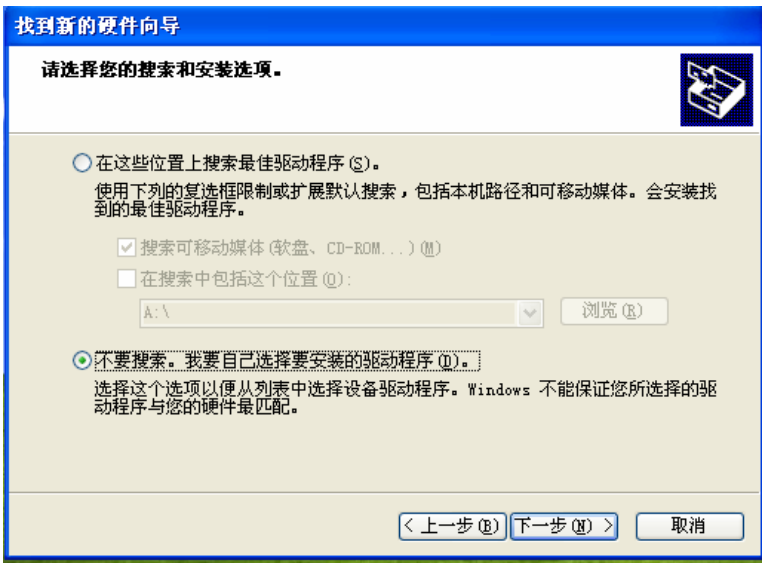
1. 首次使用本卡时，计算机将提示“发现新硬件”，如下图所示。选择“否，暂时不”，并点击“下一步”。



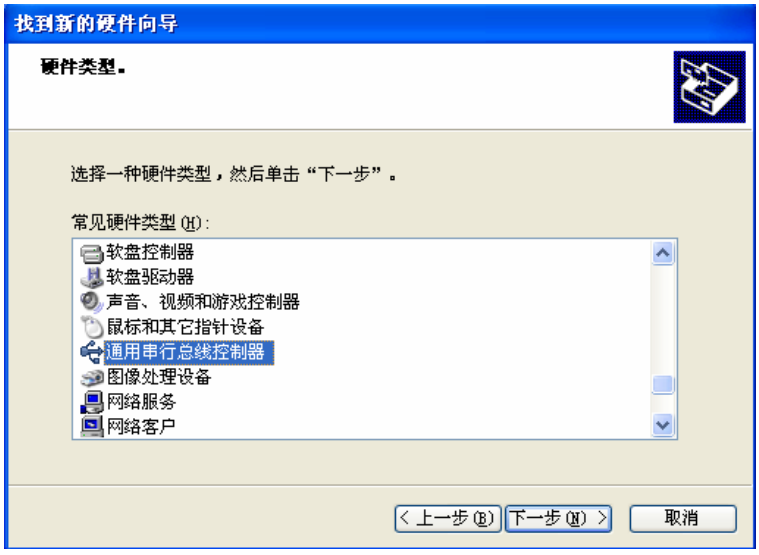
2. 选择“从列表或指定位置安装(高级)”，点击“下一步”。



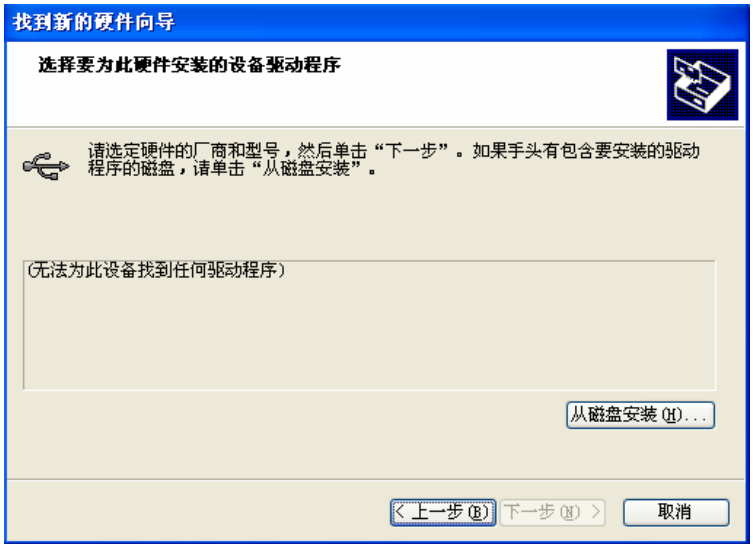
3. 选择“不要搜索。我要自己选择安装的驱动程序”，点击下一步。



4. 选择“通用串行总线控制器”，点击下一步。



5. 在弹出的对话框中选择“从磁盘安装”。



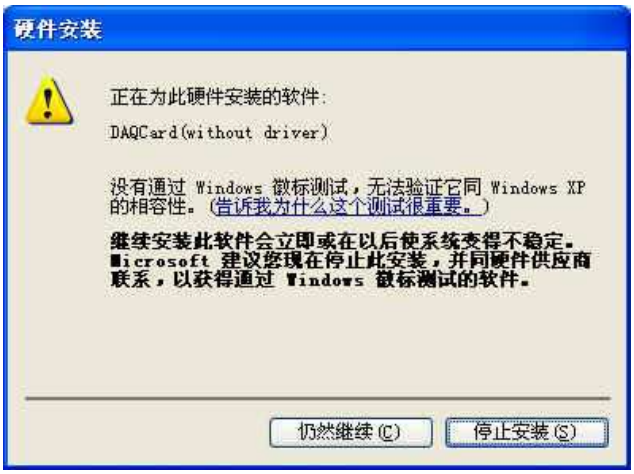
6. 选择 MPS-110001 采集卡驱动所在的目录，选中文件“MPS-1100xx.inf”。点击确定。



7. 选中“MPS-1100xx”，点击下一步。



8. 系统开始安装驱动，若弹出如下对话框，选择“仍然继续”。



9. 驱动安装完成。



四、 基本功能测试



MPS-110001 各项基本功能测试：

1. 将 MPS-110001 数据采集卡与计算机的 USB 接口连接。
2. 按第三小节所描述的过程安装驱动程序。
3. 打开范例程序中的“MPS-110001 综合测试程序”(需先安装 LabVIEW 软件)。
4. 点击左上角的执行按钮(白色箭头)，运行程序。
5. 波形图中显示采集到的电压的历史数据波形，波形图右侧控件显示当前电压值，单位为 mV。
6. 。配置“通道模式”、“增益设置”和“速度模式”可观察到采集模式发生相应变化。
7. 点击更改红色 LED 与绿色 LED 的状态，可看到采集卡上对应 LED 状态的改变。
8. 点击“STOP”退出程序。
9. 更多测试程序请参考所提供的范例程序包。

第三章 用户编程

一、 *动态链接库(DLL)*

MPS-110001 采用 DLL(Dynamic Linkable Library,动态链接库)的方式来进行编程驱动。DLL 的编制与具体的编程语言及编译器无关，只要遵循约定的 DLL 接口规范和调用方式，用各种语言编写的 DLL 都可以相互调用。

DLL 可以方便的在 VC、VB、LabVIEW 等语言下被调用，具体方式分别为：

- VC 下调用 DLL

```
typedef void ( * FUNC )(void);                  //定义一个函数指针
FUNC Func;                                      //定义一个函数指针变量
HINSTANCE hDLL=LoadLibrary("DllTest.dll"); //加载 dll
Func=(FUNC)GetProcAddress(hDLL,"FuncInDLL"); //找到 dll 中的函数
Func();                                         //调用 dll 里的函数
```

- VB 下调用 DLL

[Public | Private] Declare Function name Lib " labname " [Alias " aliasname "] [([arglist])] [As type]"

Public(可选)用于声明在所有模块中的所有过程都可以使用的函数;Private(可选)用于声明只能在包含该声明的模块中使用的函数。

Name(必选)任何合法的函数名。动态链接库的入口处(entry points)区分大小写。

Libname(必选)包含所声明的函数动态链接库名或代码资源名。

Alias(可选)表示将被调用的函数在动态链接库 (DLL) 中还有另外的名称。当外部函数名与某个函数重名时，就可以使用这个参数。当动态链接库的函数与同一范围内的公用变量、常数或任何其它过程的名称相同时，也可以使用 Alias。如果该动态链接库函数中的某个字符不符合动态链接库的命名约定时，也可以使用 Alias。

Aliasname(可选)动态链接库。如果首字符不是数字符号 (#)，则 aliasname 是动态链接库中该函数入口处的名称。如果首字符是 (#)，则随后的字符必须指定该函数入口处的顺序号。

Arglist(可选)代表调用该函数时需要传递参数的变量表。

Type(可选)Function 返回值的数据类型；可以是 Byte、Boolean、Integer、Long、Currency、Single、Double、Decimal(目前尚不支持)、Date、String(只支持变长)或 Variant，用户定义类型，或对象类型。

arglist 参数的语法如下：

[Optional] [ByVal | ByRef] [ParaArray] varname [()] [As type]

Optional(可选)表示参数不是必需的。如果使用该选项，则 arglist 中的后续参数都必需是可选的，而且必须都使用 Optional 关键字声明。如果使用了 ParamArray，则任何参数都不能使用 Optional。

ByVal(可选)表示该参数按值传递。

ByRef(可选)表示该参数按地址传递。

- LabVIEW 下调用 DLL

在 LabVIEW 中，调用 DLL 是通过 CLF 节点来完成的。所谓 CLF 节点(Call Library Function，调用函数库节点)，是指可以在 LabVIEW 调用其他语言封装的 DLL，CLF 节点位于位于 LabVIEW 功能模板中的 Advanced 子模板中，其配置过程如下：

- 在 CLF 节点的右键菜单中选择“Configure”，弹出 CLF 节点配置对话框；

- 点击“Browse”按钮，在随后弹出的选择 DLL 文件对话框中找到你需要用的 DLL 文件，此时，LabVIEW 就会自动装载选定的 DLL 文件，并检测 DLL 文件中所包含函数。但是函数中的参数和参数的数据类型需要用户根据函数的输入、输出参数手动设置。因而在调用 DLL 文件时，要求用户对 DLL 文件有较为详细的了解。
- 在 FunctionName 下拉列表框中选定动态连接库中所包含的所需要 API 函数；
- 在 Calling Convention 下拉菜单中选择 StdCall(WINAPI)和 C 两个选项，若用户选定的是 Windows API 函数，则选用 StdCall(WINAPI)选项;若用户选用的 DLL 中的函数是非 Windows API 函数，则选用 C 选项；
- 设置函数的返回参数。函数参数的类型要与 DLL 中函数本身所定义的函数参数类型相对应，如果不对应，函数就会出现数据错误和强制类型转换；
- 根据所选函数的函数原型，设置函数的输入参数及数据类型。点击 Add a Parameter 按钮，即可以添加一个新的输入参数。

二、 *编程函数及参数*

MPS-110001 提供的驱动 DLL 文件名为 MPS-110001.dll，内部共有四个驱动函数，分别为：

- ***int SI_Open (int DeviceNum, HANDLE *Handle)***

  打开一个设备，并返回该设备的句柄。

  参数:

  1. DeviceNum—设备序号. 第一个设备为 0，第二个设备为 1，依次类推。.
  2. Handle—设备句柄的指针

  返回值：0 = SUCCESS
  6 = INVALID_PARAMETER
  255 = DEVICE_NOT_FOUND

- ***int SI_Close (HANDLE Handle)***

  关闭一个设备。

  参数：

  Handle：从 SI_Open 得到的设备句柄。

  返回值：0 = SUCCESS
  1 = INVALID_HANDLE

  注意：每次用 SI_Open 函数打开一个设备后，读写完成后必须用 SI_Close 将其关闭！

- ***int SI_Read (HANDLE Handle, unsigned char * Buffer, int NumBytesToRead, int * NumBytesReturned, int NullPara)***

  读取数据。

  参数：

  Handle：从 SI_Open 得到的设备句柄。

  Buffer：数据缓存区的首地址。函数执行后该缓存内数据被更新为读取到的数据。每次读取 16 个字节，其中每四个字节代表一个通道的数据。每个通道的四字节数据中，第一个字节表示 24 位数据的高 8 位，第二个字节表示 24 位数据的中 8 位，第三个字节表示 24 位数据的低 8 位，第四个字节表示通道号。在单通道采集模式下，只有第一个通道数据有效，表示 CH1 的数据，其他通道数据均为 0；在四通

道扫描模式下，四个通道数据均有效，分别依次代表 CH，CH2，CH3 和 CH4 的数据。

NumBytesToRead：要求函数从设备读取的数据字节数，其值为 16。

NumBytesReturned：函数实际从设备读取的数据字节数，传递的参数为数据指针，函数运行完成指针中数据被更新为实际读取的数据字节数。

NullPara：无效参数，输入时令 NullPara = 0；

返回值：0 = SUCCESS
1 = INVALID_HANDLE
3 = QUEUE_NOT_READY
6 = INVALID_PARAMETER
7 = INVALID_REQUEST_LENGTH
13 = READ_TIMED_OUT
15 = IO_PENDING

- ***int SI_Write (HANDLE Handle, unsigned char *Buffer, int NumBytesToWrite, int * NumBytesWritten, int NullPara)***

写入数据。即发送控制命令。

参数：

Handle：从 SI_Open 得到的设备句柄。

Buffer：命令码所在的地址。函数执行后从该地址内第 1 个字节起的数据将被写入到设备。每次写入一个字节，内容是以 8 位无符号整形数表示的控制命令码。

NumBytesToWrite：向设备写入的数据字节数，NumBytesToWrite = 1。

NumBytesWritten：函数实际从设备读取的数据字节数，传递的参数为数据指针，函数运行完成指针中数据被更新为实际读取的数据字节数

NullPara：无效参数，输入时令 NullPara = 0；

返回值：

0 = SUCCESS
1 = INVALID_HANDLE
4 = WRITE_ERROR
6 = INVALID_PARAMETER
7 = INVALID_REQUEST_LENGTH
15 = IO_PENDING

命令码说明：

0：无命令；
1：点亮红色 LED；
2：熄灭红色 LED；
3：点亮绿色 LED；
4：熄灭绿色 LED；
5：单通道采集模式；
6：四通道扫描模式；
7：标准模式(单通道 10sps，四通道 0.625sps)；
8：快速模式(单通道 80sps，四通道 4.5sps)；
9：无增益；

10：增益为 32；
11：增益为 64；
12：增益为 0.5；

- ***int SI_FlushBuffers (HANDLE Handle, unsigned char FlushTransmit, unsigned char FlushReceive)***

清除缓冲区数据，用于在较长时间未读取过数据时清除旧的数据缓存，以保证读到的是最新的数据。
参数：Handle：从 SI_Open 得到的设备句柄。
FlushTransmit：无效果，可设置为任意非 0 值。
FlushReceive：无效果，可设置为任意非 0 值。
注：需要清除缓存数据的情况下使用本函数，连续读取数据时可不使用。

补充说明:

- REF 参考电压输出值为 2.048V，可用于为电桥电路或传感器提供驱动电压，也可作为中值电压与输入电压组建差分对。
- 24 位数据的最高位为符号位，若符号位为 0，24 位数据为采集到数据的二进制码，若符号位为 1，24 为数据位采集到数据的二进制补码。即：VREF – 1LSB 对应的数据值为 0x7FFFFF；VREF/2 对应的数据值为 0x4000000；+1 LSB 对应的数据值为 0x000001；-1 LSB 对应的数据值为 0xFFFFFF；VREF/2 对应的数据值为 0xC00000；－VREF 对应的数据值为 0x800000。
- 假设采集到的 24 位数据为 DATA，则实际电压 U 与 DATA 的关系为：U = (DATA / 8388608) * VREF，其中 VREF 的值为 1.024。

三、 *VC 程序范例*

```
float MPS_TEST()
{
    HINSTANCE hDll = LoadLibrary("MSP-110001.dll");                    //加载DLL
    if(NULL==hDll)
    {
        AfxMessageBox("Cann't find DLL");
    }

    //函数声明
    //Open函数
    typedef  int (_stdcall *  Type_Open)(int DeviceNumber, HANDLE *Handle);
    Type_Open  MPS_Open =(Type_Open)GetProcAddress(hDll,"SI_Open");
    if(MPS_Open == NULL)
    {
        AfxMessageBox("Cann't find <SI_Open> function");
    }

    //Close函数
    typedef  int (_stdcall * Type_Close)(HANDLE Handle);
    Type_Close  MPS_Close =(Type_Close)GetProcAddress(hDll,"SI_Close");
    if(MPS_Open == NULL)
```

```
    {
        AfxMessageBox("Cann't find <SI_Close> function");
    }

    //Read函数
    typedef int(_stdcall * Type_Read)(HANDLE Handle, unsigned char * Buffer, int NumBytesToRead,
int * NumBytesReturned,  int  NullPara);
    Type_Read  MPS_Read =(Type_Read)GetProcAddress(hDll,"SI_Read");
    if(MPS_Read == NULL)
    {
        AfxMessageBox("Cann't find <SI_Read> function");
    }

    //Write函数
   typedef int(_stdcall * Type_Write)(HANDLE Handle, unsigned char *Buffer, int NumBytesToWrite,
int * NumBytesWritten, int  NullPara);
    Type_Write  MPS_Write =(Type_Write)GetProcAddress(hDll,"SI_Write");
    if(MPS_Write == NULL)
    {
        AfxMessageBox("Cann't find <SI_Write> function");
    }

//变量定义及初始化
    HANDLE Handle = NULL;
    unsigned char DataBuffer[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,};
    unsigned char Command[1] = {1,};                                    //点亮红色LED
    double Voltage = 0;
    int NumBytesReturned = 0;
    int NumBytesWritten = 0;
    int MPS_flag = 0;

//采集与输出
    MPS_flag = MPS_Open(0, &Handle);                                    //打开
    if(MPS_flag != 0)
    {
        AfxMessageBox("Open Error!Please check hardware!");             //报错
    }
    else
    {
        MPS_Read(Handle, DataBuffer, 16, & NumBytesReturned, 0);        // 采集

        Voltage = (int)DataBuffer[0] * 65536 + (int)DataBuffer[1] * 256 + DataBuffer[2];
                                                                        // 数据处理
        if(Voltage > 8388608) Voltage = Voltage - 8388608 * 2;
```

```
            Voltage = (Voltage / 8388608) * 1.024;

            //MPS_Write(Handle, Command, 1, &NumBytesWritten, 0);            // 发送命令
            //i = 100000000;while(--i);                                      // 发送延时

            //此处可添加对采集到的数据的处理代码

            MPS_Close(Handle);
        }
        return Voltage * 1000000;                                  // 单位为微伏
}
```

第四章 注意事项

- 拔插采集卡请用力适度，以免损害 USB 接口。
- 采集前务必将采集卡的 GND 端口与信号地相连。
- 模拟信号输入端口耐压范围为 0V 至+4V，允许范围之外的电压有可能对采集卡硬件造成损害。
- 使用过程中不要用手接触电路和芯片，避免人体静电对硬件造成损害。采集卡保存时请妥善保管，注意防尘防潮。
- MPS-110001 自出厂之日起，一年内凡用户遵守贮存，运输和使用要求，而产品质量低于技术指标的，凭保修卡免费维修。因违反操作规定和要求而造成损坏的，需交纳器件维修费。
- MPS 系列数据采集卡由北京启创莫非电子科技有限公司设计和生产。
- 网站：www.mps-electronic.com.cn,咨询邮箱 mail@mps-electronic.com.cn。