



# USB2I2C 转换器 产品使用说明书

产品型号：USB2I2C转换器

手册版本：V1.2

[www.huanor.com](http://www.huanor.com)



## 目 录

目录.....	2
一、产品简介.....	3
1.1 典型应用.....	3
1.2 性能与技术指标.....	4
1.3 典型应用.....	4
1.4 通信协议转换.....	4
1.5 产品销售清单.....	4
1.6 技术支持与服务.....	4
二、外形与接口描述.....	5
2.1 产品外形.....	5
2.2 适配器对外接口定义.....	5
三、功能说明.....	7
四、驱动程序的安装.....	8
五、上位PC机应用软件开发.....	9
5.1 设备管理API.....	9
5.2 中断处理API.....	10
5.3 I2C串口数据传输API.....	10

## 一、产品简介

### 1.1 产品描述

USB2I2C是一个USB总线的转接适配器，可实现：

- USB总线到100KHz标准I2C模式；
- USB总线到400KHz快速I2C模式；
- USB总线到750KHz高速（HS）I2C模式；
- USB总线到20KHz慢速I2C模式

USB2I2C提供主I2C接口，实现PC上位机和下位控制器之间的直接数据输入输出，而不再需要单片机/DSP/MCU等的监控。在同步串口方式下，USB2I2C还支持兼容I2C（IIC）总线的其它2线制TWI/SMBUS同步串口，提供SCL线和SDA线。PC上位机可以方便地对I2C/IIC/TWI/SMBUS接口器件进行读写。

通过USB2I2C转换器用户可以非常方便地实现PC机USB总线和下位机端各种I2C/IIC设备之间的通信：

- ATMEL公司的AT24CXX系列EEPROM；
- I2C总线8位并行IO口扩展芯片PCF8574/JLC1562；
- I2C接口实时时钟芯片DS1307/PCF8563/SD2000D/M41T80/ME901/ISL1208/；
- I2C数据采集ADC芯片MCP3221（12bitADC）/ADS1100（16bitADC）/ADS1112（16bitADC）/MAX1238（12bitADC）/MAX1239（12bitADC）；
- I2C接口数模转换DAC芯片DAC5574（8bitDAC）/DAC6573（10bitDAC）/DAC8571（16bitDAC）；
- I2C接口温度传感器TMP101/TMP275/DS1621/MAX6625

USB2I2C还支持对非标准I2C协议的器件进行读写

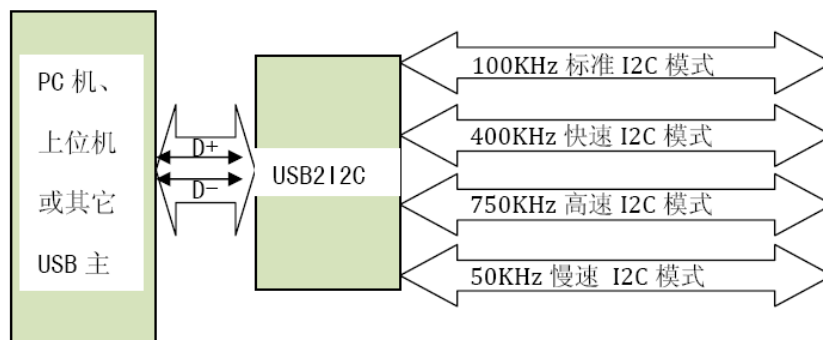


图 1 - USB2I2C转换器功能结构



## 1.2 性能与技术指标

- USB 2.0转I2C接口，USB总线供电，无需外部电源；
- 2线制I2C/IIC/TWI/SMBUS接口，支持20KHz/100KHz/400KHz/750KHz 4种传输速度；
- 提供电源输出：+3.3v ,+5V；
- 接口信号：SCL、SDA、GND、+5V、+3.3V；
- 2线制同步串口I2C/IIC/TWI/SMBUS是主动式串口，只能作为I2C/IIC/TWI/SMBUS总线上的Host或Master主机端，在计算机端的过程控制下，可以直接从外部电路输入输出数据，一般不需要外接单片机/DSP/MCU；
- 工作温度：-20°C - +70°C。

## 1.3 典型应用

I2C 接口的元器件寄存器读写；

I2C 接口的EEPROM 读写；

## 1.4 通信协议转换

USB 转I2C 总线接口转换。

## 1.5 产品销售清单

USB转I2C适配器一台；

USB连接线一根；

光盘1张（包括PC驱动、接口函数动态库及示例代码、使用说明书等）

## 1.6 技术支持与服务

电话：010—82534219 传真：010—82534219

E-mail:huanor\_supports@hotmail.com

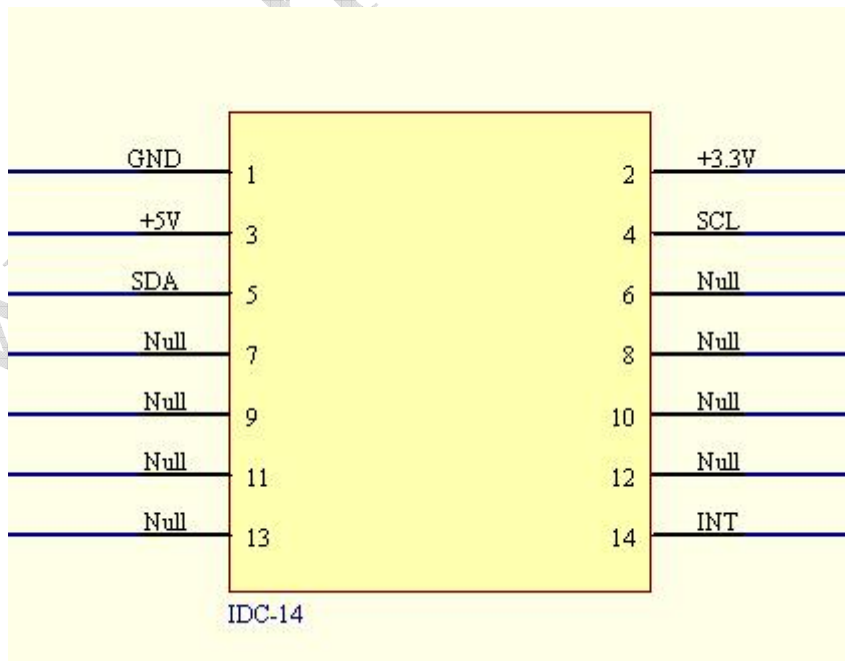
## 二、外形与接口描述

### 2.1 产品外形



### 2.2 适配器对外接口定义

适配器直接的输出接口为14pin的针式接口，通过14芯的直连排线延长后可得到孔式接口。





PIN 脚描述:

引脚序号	名称	功能描述
PIN1	GND	电源地与信号地
PIN2	+3.3V	+3.3V电源输出 (电流小于100mA)
PIN3	+5V	+5V电源输出 (电流小于100mA)
PIN4	SCL	2线串口的时钟输出, 内置上拉电阻
PIN5	SDA	2线串口的数据输入输出, 内置上拉电阻
PIN6	Null	空脚
PIN7	Null	空脚
PIN8	Null	空脚
PIN9	Null	空脚
PIN10	Null	空脚
PIN11	Null	空脚
PIN12	Null	空脚
PIN13	Null	空脚
PIN14	INT	中断请求输入, 上升沿有效, 内置上拉电阻

注1: 如果从机的电路上已有供电, 则请不要使用适配器提供的电源。

因适配器提供电源信号的驱动电流比较小(小于100mA), 请用户使用时谨慎评估, 以防PC 的 USB 接口损坏。



### 三、功能说明

由USB2I2C转换器转换的2线制同步串口I2C/IIC/TWI/SMBUS是主动式串口,只能作为I2C/IIC/TWI/SMBUS总线上的Host或Master主机端,在计算机端的过程控制下,可以直接从外部电路输入输出数据,一般不需要外接单片机/DSP/MCU。

2线制同步串口I2C/IIC/TWI/SMBUS的主要引脚包括SCL引脚、SDA引脚。SCL用于单向输出同步时钟,开漏输出且内置上拉电阻,SDA用于准双向数据输入输出,开漏输出及输入且内置上拉电阻。

2线制同步串口I2C/IIC/TWI/SMBUS的基本操作元素包括:起始位、停止位、位输出、位输入。

起始位定义为当SDA为高电平时,SCL输出下降沿(从高电平切换为低电平)。

停止位定义为当SDA为高电平时,SCL输入上升沿(从低电平切换为高电平)。

位输出定义为当SCL为低电平时,SDA输出位数据,然后SCL输出高电平脉冲。

位输入定义为SCL输出高电平脉冲,在下降沿之前从SDA输入位数据。

字节输出定义为8个位输出及1个位输入用于应答。

字节输入定义为8个位输入及1个位输出用于应答。

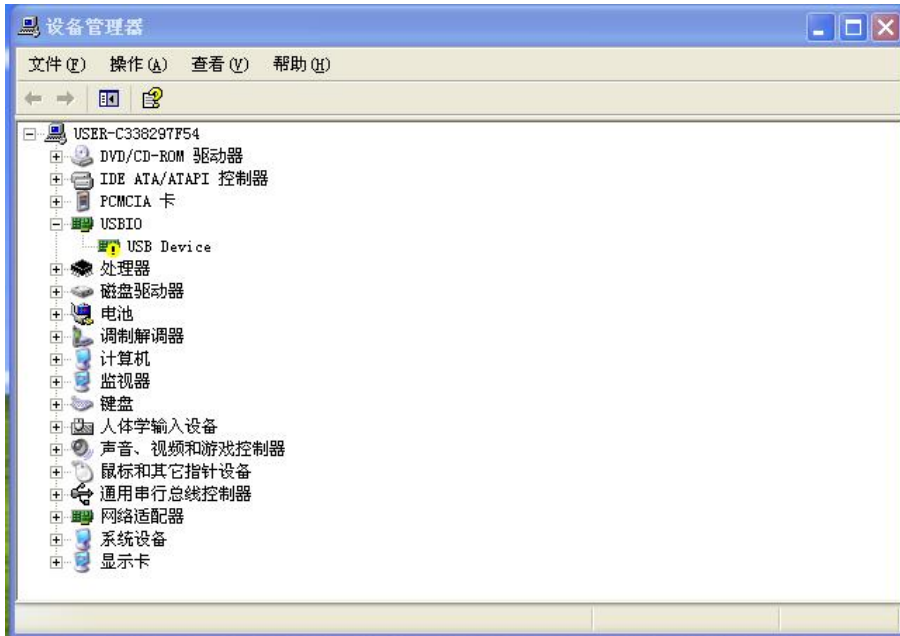
I2C总线的数据输入和输出以字节为单位,每个字节含8个位,高位在前。

其2线制同步串口支持大部分的标准或兼容I2C/IIC/TWI/SMBUS的设备,如:

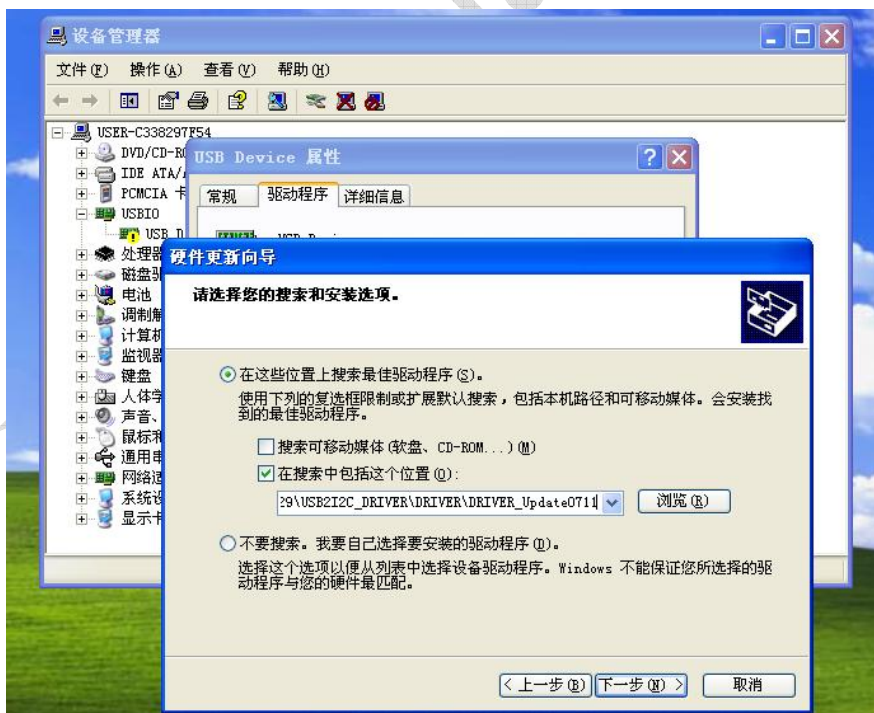
- I2C界面EEPROM: 24C01A到24C16、24C32到24C1024等;
- I2C总线8位并行IO口扩展芯片PCF8574/JLC1562;
- I2C接口实时时钟芯片DS1307/PCF8563/SD2000D/M41T80/ME901/ISL1208/;
- I2C数据采集ADC芯片MCP3221(12bitADC)/ADS1100(16bitADC)/ADS1112(16bitADC)/MAX1238(12bitADC)/MAX1239(12bitADC);
- I2C接口数模转换DAC芯片DAC5574(8bitDAC)/DAC6573(10bitDAC)/DAC8571(16bitDAC);
- I2C接口温度传感器TMP101/TMP275/DS1621/MAX6625,等

## 四、驱动程序的安装

先将设备接入PC 或笔记本电脑的USB 接口，根据提示安装我们提供的驱动程序。接入USB2I2C 设备到PC。在“我的电脑”右键“属性”中选择设备管理器，可以看到



双击该USB-API 设备，手动添加该设备的驱动程序，如下图，选择驱动程序所在的目录进行安装。过程中会提示该驱动程序是否确认安装，点确认认可。



安装完成后，设备管理器中会指示有“USB2I2C”。





## 五、上位PC机应用软件开发

在计算机端的Windows操作系统下,USB212C的并口驱动程序和动态链接库USBIOX.DLL向应用程序提供了应用层接口,包括:设备管理API、并口数据传输API、同步串口数据传输API、中断处理API。有关API参数的说明请参考USBIOX.H,主要API如下。

有关DLL中各个API的使用实例请参考USB212C评估板数据中的各个源程序及例子。

### 5.1 设备管理API

- ✓ **USBIO\_OpenDevice**(//打开USB212C设备,返回句柄,出错则无效  
    **ULONG iIndex**); //指定USB212C设备序号,0对应第一个设备  
说明:将USBIO\_作为设备,使用前必须先打开,然后才能使用。
  
- ✓ **USBIO\_CloseDevice**(//关闭USB212C设备  
    **ULONG iIndex**); //指定USB212C设备序号  
说明:用完USBIO\_后,或者应用程序退出前,应该关闭USB212C设备。
  
- ✓ **USBIO\_SetDeviceNotify**(//设定设备事件通知程序  
    **ULONG iIndex**, //指定USB212C设备序号,0对应第一个设备  
    **PCHAR iDeviceID**, //可选参数,指向字符串,指定被监控的设备的ID,字符串以\0终止  
    **mPUSBIO\_\_NOTIFY\_ROUTINE iNotifyRoutine**); //指定设备事件回调程序  
说明:用于应用程序监控USB212C设备的插拔事件,确保应用程序随时知道USB设备是否存在,防止在USB设备拔出后收发数据,并及时响应USB设备的插入。
  
- ✓ **USBIO\_GetStatus**(//通过USB212C直接输入数据和状态,类似的API还有  
    **USBIO\_GetInput**  
    **ULONG iIndex**, //指定USB212C设备序号  
    **PULONG iStatus**); //指向一个双字单元,用于保存状态数据  
说明:获取的状态数据中:位7-位0对应USB212C的D7-D0引脚,位8对应USB212C的ERR#引脚,位9对应USB212C的PEMP引脚,位10对应USB212C的INT#引脚,位11对应USB212C的SLCT引脚,位13对应USB212C的WAIT#引脚,位14对应USB212C的DS#引脚,位15对应USB212C的AS#引脚,位23对应USB212C的SDA引脚。
  
- ✓ **USBIO\_SetOutput**(//设置USB212C的I/O方向,并通过USB212C直接输出数据  
    **ULONG iIndex**, //指定USB212C设备序号  
    **ULONG iEnable**, //数据有效标志  
    **ULONG iSetDirOut**, //设置I/O方向,位清0则对应引脚为输入,位置1则对应引脚为输出  
    **ULONG iSetDataOut**); //输出数据,如果I/O方向为输出,那么位数据将通过引脚输出  
说明:谨慎使用该API,防止修改I/O方向使输入引脚变为输出导致与其它输出引脚之间短路



而损坏。上述的I/O方向和输出数据以32位数据表示,其中:位7-位0对应USB212C的D7-D0引脚,位8对应USB212C的ERR#引脚,位9对应USB212C的PEMP引脚,位10对应USB212C的INT#引脚,位11对应USB212C的SLCT引脚,位13对应USB212C的WAIT#引脚,位14对应USB212C的DS#/READ#引脚,位15对应USB212C的AS#引脚另外,以下引脚只能输出,不考虑I/O方向:位16对应USB212C的RESET#引脚,位17对应USB212C的WRITE#引脚,位18对应USB212C的SCL引脚,位29对应USB212C的SDA引脚。

√ USBIO\_Set\_D5\_D0(//设置USB212C的D5-D0引脚的I/O方向,并通过D5-D0引脚直接输出数据

ULONG iIndex, //指定USB212C设备序号

ULONG iSetDirOut, //设置D5-D0各引脚的I/O方向,清0则引脚为输入,置1则引脚为输出

ULONG iSetDataOut); //设置D5-D0各引脚的输出数据,仅当I/O方向为输出时生效

说明:谨慎使用该API,防止修改I/O方向使输入引脚变为输出导致与其它输出引脚之间短路而损坏。

### 5.2 中断处理API

√ USBIO\_SetIntRoutine(//设定中断服务程序

ULONG iIndex, //指定USB212C设备序号

mPUSBIO\_INT\_ROUTINE iIntRoutine); //指定中断服务程序,为NULL则取消中断服务

说明:设置USB212C的中断服务程序,iIntRoutine是一个符合mPUSBIO\_INT\_ROUTINE格式的子程序,当USB212C的INT#引脚出现上升沿时,USBIOX.DLL自动调用iIntRoutine,并向其提供一个引脚状态参数,引脚状态参数中,位为1则说明对应的引脚为高电平,位为0则说明对应的引脚为低电平,位7-位0对应USB212C的D7-D0引脚,位8对应USB212C的ERR#引脚,位9对应USB212C的PEMP引脚,位10对应USB212C的INT#引脚,位11对应USB212C的SLCT引脚。例如:主程序

```
main{
```

```
.....
```

```
USBIO_OpenDevice(0); //打开设备,针对0#设备,如果有多个,可以计数
```

```
USBIO_SetIntRoutine(0,myInterruptEvent); //设置中断服务程序
```

```
..... 读写数据,或者在接收到中断服务程序的通知后处理中断
```

```
USBIO_CloseDevice(0); //用完后关闭设备
```

```
}
```

中断服务程序,当USB212C的INT#引脚出现上升沿时,USBIOX.DLL会自动调用该子程序,

```
Void CALLBACK myInterruptEvent(unsigned long PinStatus) {
```

```
if(PinStatus & mStateBitERR) printf(“发生中断时ERR#引脚为高电平”);
```

```
else printf(“发生中断时ERR#引脚为低电平”);
```

```
..... 自己处理或者通知主程序处理
```

```
}
```

### 5.3 I2C 串口数据传输API

√ USBIO\_ReadI2C(//从两线串口读取一个字·节数据,仅适用于7位地址的设备



ULONG iIndex, //指定USB212C设备序号  
ULONG iDevice, //低7位指定设备地址  
ULONG iAddr, //指定数据单元的地址  
PULONG oByte); //指向一个字节单元,用于保存读取的字节数据

说明: 从两线串口读取一个字节数据。仅适用于7位地址的设备, 不支持带从地址的12C设备。

√ USBIO\_Write12C(//向两线串口写入一个字节数据, 仅适用于7位地址的设备

ULONG iIndex, //指定USB212C设备序号  
ULONG iDevice, //低7位指定设备地址  
ULONG iAddr, //指定数据单元的地址  
ULONG iByte); //待写入的字节数

说明: 从两线串口读取一个字节数据。仅适用于7位地址的设备, 不支持带从地址的12C设备。

√ USBIO\_WriteRead(//执行数据流命令, 先输出再输入

ULONG iIndex, //指定USB212C设备序号  
ULONG iWriteLength, //写长度, 准备写出的长度  
ULONG iWriteBuffer, //指向一个缓冲区, 放置准备写出的资料  
ULONG iReadStep, //准备读取的单个块的长度, 总长度为

(iReadStep\*iReadTimes)

ULONG iReadTimes, //准备读取的次数  
PULONG oReadLength, //指向长度单元, 返回后为实际读取的长度  
ULONG oReadBuffer); //指向一个足够大的缓冲区, 用于保存读取的数据

说明: 先输出数据再输入数据, 执行数据流命令, 适用于同步串口等。

√ USBIO\_SetStream(//设置同步串口流模式

ULONG iIndex, //指定USB212C设备序号  
ULONG iMode); //指定模式, 见下面的说明

说明: iMode的位1位0: 12C速度/SCL频率, 00=低速20KHz, 01=标准100KHz, 10=快速400KHz, 11=高速750KHz//位2: SPI的I/O数/I/O引脚, 0=单入单出(4线界面), 1=双入双出(5线接口)//位7: SPI字节中的位顺序, 0=低位在前, 1=高位在前//其它保留, 必须为0。

√ USBIO\_Stream12C(//处理两线串口的数据流, 适用于所有两线串口的设备

ULONG iIndex, //指定USB212C设备序号  
ULONG iWriteLength, //准备写出的数据字节数  
ULONG iWriteBuffer, //指向缓冲区, 放置准备写出的数据, 首字节是设备地址及

读写位

ULONG iReadLength, //准备读取的数据字节数  
ULONG oReadBuffer); //指向缓冲区, 返回后是读入的数据对两线串口设备进行操

作。

例如, 从24C256中3200H开始的地址读出256字节的数据:



```
ULONG OutBuf[5], InBuf[300]; //待写数据缓冲区, 读出数据缓冲区  
OutBuf[0]=0xA1;  
OutBuf[1]=0x32;  
OutBuf[2]=0x00; //待写数据: 设备地址及单元地址  
USBIO_Stream12C(0, 3, OutBuf, 256, InBuf); //针对0#设备处理两线串口的数据流
```

- √ USBIO\_ReadEEPROM(//从EEPROM中读取数据块, 速度约56K字节

```
    ULONG iIndex, //指定USB212C设备序号  
    EEPROM_TYPE iEepromID, //指定EEPROM型号  
    ULONG iAddr, //指定数据单元的地址  
    ULONG iLength, //准备读取的数据字节数  
    PULONG oBuffer); //指向一个缓冲区, 返回后是读入的数据
```

说明: 读EEPROM的API支持从24C01到24C16和从24C32到24C4096的各种型号的EEPROM内存。

- √ USBIO\_WriteEEPROM(//向EEPROM中写入数据块

```
    ULONG iIndex, //指定USB212C设备序号  
    EEPROM_TYPE iEepromID, //指定EEPROM型号  
    ULONG iAddr, //指定数据单元的地址  
    ULONG iLength, //准备写出的数据字节数  
    PULONG iBuffer); //指向一个缓冲区, 放置准备写出的资料
```

说明: 写EEPROM的API支持从24C01到24C16和从24C32到24C4096的各种型号的EEPROM内存。



# 北京华诺铭科电子科技中心

地址：北京市海淀区中关村大街28-1号中海园电子市场BF-198

（位于 海淀剧院旁 或者 中发电子大厦对面）

电话：010-82534219

传真：010-82534219

网址：[www.huanor.com](http://www.huanor.com)

技术支持：[huanor\\_supports@hotmail.com](mailto:huanor_supports@hotmail.com)

销售：[huanor\\_sales@hotmail.com](mailto:huanor_sales@hotmail.com)

郑重承诺：本公司产品，非人为损坏，三个月内包换，一年内保修！