

PCI 总线

200Ksps/12 位 32 通道 AD

4 通道 DA 板

4 通道脉冲计数

8 通道数字输入/8 通道数字输出

RBH7272

使用说明书

北京瑞博华控制技术有限公司

二 00 七年十月

200Ksps/12 位 32 通道 AD
4 通道 DA 板
4 通道脉冲计数（1 个旋转编码器）
8 通道数字输入/8 通道数字输出
RBH7272

使用说明书

一、性能特点：

本板采用 PCI 总线接口。

本板通过采用高速高精度 AD 芯片、高精度的仪器放大器、高密度 CPLD 逻辑芯片、精细地布线以及优良的制版工艺，实现了高速、高精度实时数据采集，具有以下性能特点：

- 1、AD 高精度：综合误差小于 $\pm 1\text{LSB}$ 。
- 2、4 通道脉冲功能：第一通道脉冲与第二通道脉冲可以构成一个旋转编码器输入，实时直接给出正向与反向脉冲数；第一通道同时输出脉冲的瞬态周期，时间分辨率为 1 微秒；第二、第三、第四通道输出 16 位的计数值，输入最高脉冲频率为采样频率 $\times 255$ ，如采样频率 10000，则输入最高脉冲频率为 2.55MHz。
- 3、开关量同步采集输入：8 路开关量与 4 路脉冲量合成第一通道与模拟量同步采集。便于用户对开关量的高速采集。可以用于外部同步触发等功能。
- 4、DA 通道数：4 通道独立锁存，精度 12 位
- 5、AD 高速度：多通道采集速度达到 200Ksps (Sample Per Second)，单通道方式也能够达到 70Ksps 以上，特别适合于工业控制中的数据采集。
- 6、程控放大器功能，可以设置放大倍数为 1、2、4、8 或 1、10、100、1000。
- 7、AD 硬件定时：板上提供硬件定时器，可以根据需要发出定时中断，采集软件在定时中断程序中采集，从而保证准确地时间基准，适于大部分的工业实时控制和高速数据采集的应用，特别是在 WINDOWS95/98/2000 的环境下，由于 PC 系统很难提供高精度的定时，采用本板的定时器，能够提供高精度的定时，同时能够实现高精度的数据采集，因此，在 WINDOWS 环境下采用本板具有特别的优点。
- 8、电流监测功能：本板只需焊接上检测电阻，就能够实现电流检测。用户可以按要求自己焊接，也可由本公司帮助焊接。
- 9、灵活扩展：可以根据用户需要定制 CPLD 内容，从而实现不同的功能要求。特别是对产品化开发的用户，可以进行个性化的定制服务。

二、功能与指标

AD 的性能指标：

- AD 采样精度：12 位
- AD 通道数：单端方式 32 通道，双端方式 16 通道
- AD 系统数据采集实际贯通率：200K/S
- AD 芯片转换速度：1000K/S
- AD 采样幅值综合误差： $\pm 1\text{LSB}$
- AD 输入电压范围：-5V 到 +5V 或 0—10V

- AD 输入阻抗：100 千欧
- 中断源：硬件定时器中断
- 触发方式：硬件定时触发

DA 的性能指标：

- 通道数：4 路独立输出
- 输出方式：电压输出， $-5V$ — $+5V$ 或 0 — $10V$
- 精度：12 位

接口：

- 总线方式：32 位 PCI 总线
- 接头方式：DB37（针式）

开关量指标：

- 8 路数字量输入，输入电平为 TTL 电平，与模拟量同步采集
- 8 路数字量输出，复位清零功能，TTL 电平
- 4 路脉冲计数功能，TTL 电平。第一路与第二路构成旋转编码器输入，采集正向脉冲数与反向脉冲数；输出的第一路的脉冲周期，时间分辨率为 1 微秒；输出第二、第三、第四路的计数值。
- 工作温度： 0 — 70°C

订货指南：

- RBH7272-02 表示与 RBH7202 兼容的采集板，功能与 RBH7202 完全相同，软件兼容。
- RBH7272-00 表示与 RBH7200 兼容的采集板，功能与 RBH7200 完全相同，软件兼容。
- RBH7272-07 为脉冲计数型，包括 32 通道的模拟量采集和 4 通道的 DA 输出，并包括 4 通道的脉冲计数功能，带一路脉冲编码器采集功能。
- RBH7272-XX 为用户定制型号，可以根据用户特殊需要进行 CPLD 定制。

三、AD 板工作原理简介

RBH7272 的实物如图 1 所示。

图中左侧为 37 针模拟量接口 J1，包括 32 通道的 AD 输入和 4 通道的 DA 输出，针 19 为模拟地线。信号定义在 J1 的上方有标注。

板卡的上中为 8 路开关量输出、8 通道开关量输入，4 通道的脉冲输出。

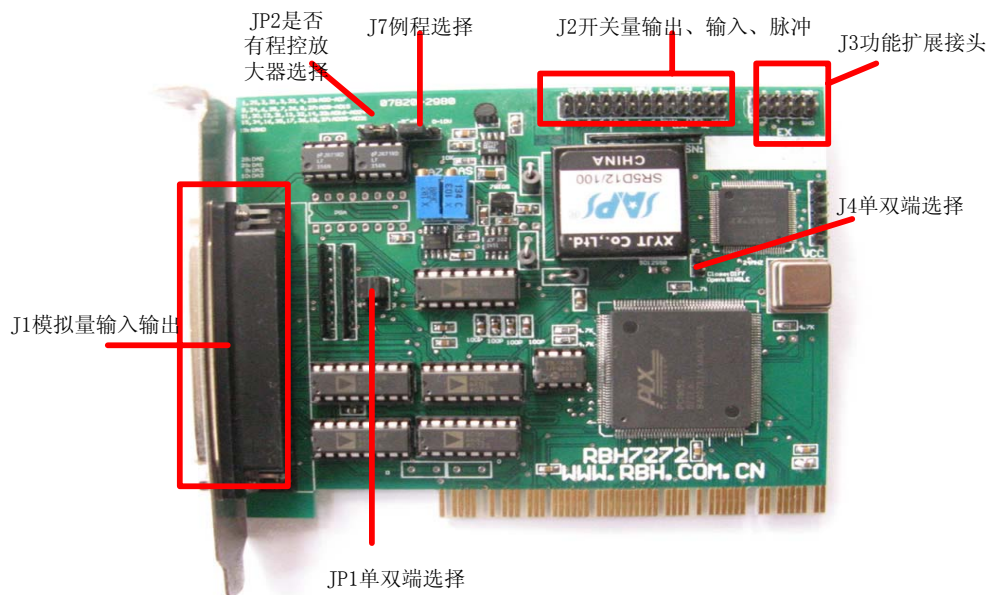


图 1 RBH7272 实物照片

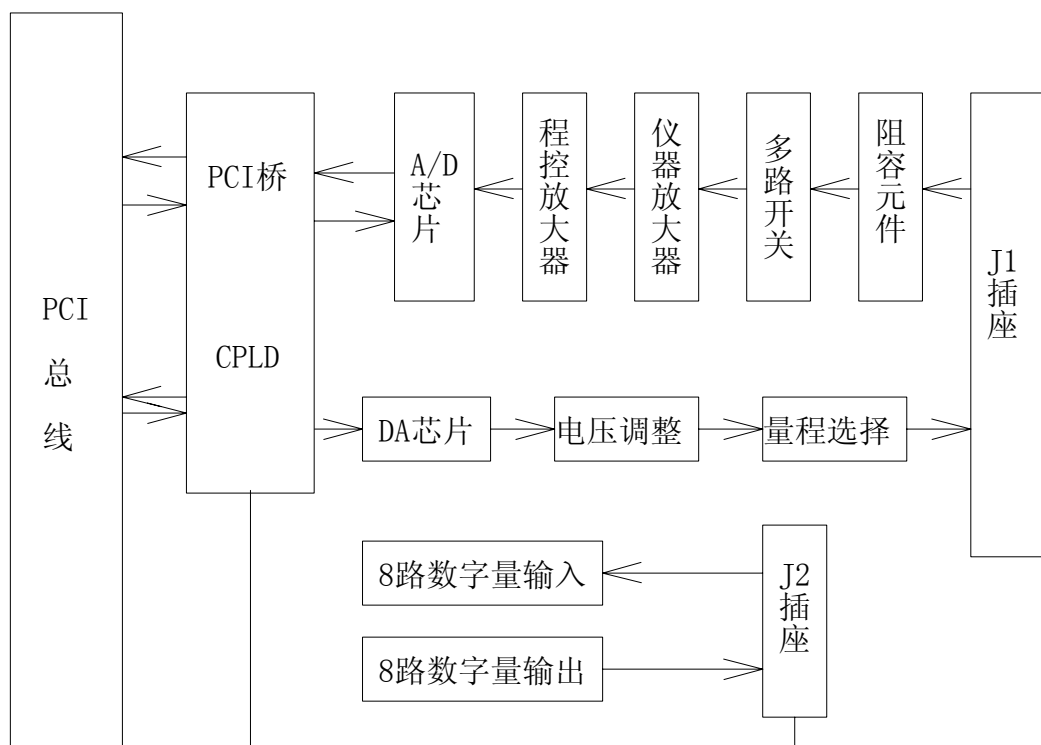


图 2 RBH7272 的原理框图

RBH7272 板的硬件组成原理框图如图 2 所示。信号从模拟量输入接头 J1 输入，然后经过阻容元件、多路开关进入仪器放大器，经过仪器放大器实现阻抗匹配和干扰抑制，再送到程控放大器，然后送到 A/D 芯片。PC 机首先选通相应的通道，然后触发 A/D，A/D 完成后，读取 A/D 结果。

DA 芯片采样 AD 公司的 12 位独立 DA 转换器，实现 4 路 DA 的独立输出。由于有复

位控制功能，当计算机复位或计算机重新启动时，DA 输出自动降到最低电压，能够保证系统的安全，这在很多工控项目中都非常重要。

板上的定时器定时给计算机发出中断，软件通过响应中断，实现实时控制功能。

一般信号采集直接采用单端即可，将信号的地线与本板的模拟地线相接，将信号线接本板的通道线。由于本板有很强的共模噪声抑制能力，将信号直接接采集板能够保证有很高的采集精度。对于噪声特别严重的信号，可以采用双端的方式输入，首先将板配置成双端采集模式，然后将信号的两端接通道的两端。本板出库时，设置为单端方式。

阻容元件是根据用户需要，可以灵活配置的元件，接线如图 3 所示。

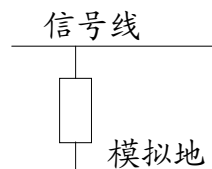


图 3 阻容元件接线

- 滤波：当阻容元件是电容式，该元件起滤波作用
- 下拉：当阻容元件为大电阻，如 10K 欧，则表示将信号下拉，当外部接线断路时，数据采集的结果仍然为 0，这在一些控制系统中非常有用。
- 电流探测：当采用精密的电阻，用以检测电流时，由于本板的输入阻抗非常大，因此信号源的电流大部分都流经探测电阻，电阻两端的电压可以由采集板检测，从而根据与电流的关系，计算出电流的大小。

本板的阻容元件出库时空接，用户可以根据自己的需要焊接，元件的规格是表面贴元件 0805 系列。

四、硬件使用方法

1、操作元件布置

如图 1 所示，操作元件都已经标注。JP1 和 J4 联合控制单端与双端的选择。JP2 用于程控放大器设置，当没有程控放大器时，JP2 的 1、2 短接；当有程控放大器时，JP2 的 2、3 短接。出库时不带程控放大器，JP2 的 1、2 短接。J1 是 AD 模拟信号的输入和 DA 的模拟信号输出。J7 用于选择 AD 输入的模拟信号类型，当 J7 的 1，2 短接时，输入的信号为单极性信号，输入的 AD 信号范围是从 0V 到 10V，当 J7 的 2，3 短接时，输入的信号为双极性信号，输入的 AD 信号范围是从 -5V 到 +5V；

4 通道的 DA 输出方式完全相同。输出电平选择相同，零点与增益调整也完全系统。

WR1 用于 DA 的零点调整，板上标注为 DAZ (Digit To Analog Zero)。

WR2 用于 DA 增益调整，板上标注为 DAS (Digit To Analog Span)

J7 用于 DA 的输出极性选择，当 J7 的 1，2 闭合时，输出的电压范围是 -5V 到 +5V，当 J7 的 2，3 闭合时，输出的电压范围是 0—10V。

以上设置已经标识在板卡上。

3、AD 的单端与双端输入方式选择

通过短接器 JP1 和 J4 实现单端与双端的转换。

单端方式：JP1 的 1、2 短接，4、5 短接；J4 的 1、2 短接。双端方式：JP1 的 2、3

短接、5、6 短接；J4 的 1、2 断开。

出库时，设置成单端方式。

4、AD 的增益调整与零点调整

AD 输入的零点与增益调整采用软件调整方式。

当用户软件中的启动采集函数中的 AmpGain 赋值为 0 时，输出的模拟量结果为原始值，当 AmpGain 不为 0 时，输出的模拟量结果为校准后的结果。

5、AD 电压信号与 AD 输出数码的关系

输出采用偏移码方式。当输入为 $-5V$ 至 $+5V$ ：输入 $-5.000V$ 时，对应的数码是 0H；当输入是 0 电压时，输出的数码为 800H；当输入的电压为 $+5.000V$ 时，输出的数码为 FFFH；当输入为 $0V$ 至 $+10V$ ：输入 $0V$ 时，对应的数码是 0H；当输入是 5V 电压时，输出的数码为 800H；当输入的电压为 $+10.000V$ 时，输出的数码为 FFFH。

6、程控放大器的设置

本板可以接程控放大器，程控放大器可以是 PGA204、PGA205 或 PGA206。当选择 PGA204 时，4 种放大倍数是 1、10、100、1000。当选择程控放大器 PGA205 时，4 种放大倍数是 1、2、4、8。当选择程控放大器 PGA206 时，4 种放大倍数是 1、2、5、10。程控放大器由两个控制端 A0 和 A1 控制，当 A1A0 为二进制的 00、01、10、11 时，分别选择 4 种放大倍数 1、10、100、1000 或 1、2、4、8 或 1、2、5、10。

7、接线插座的信号定义

图 4 是开关量接线图，D00-D07 是 8 路开关量输出，DI0-DI7 是 8 路开关量输入，Fout 是 500K 的方波，Out0，Out1 备用输出，CLK1，CLK2，CLK3，CLK4 是 4 路脉冲输入，针 25 输出高电平，针 26 是地线。针 23，24 悬空，没有内部连接。

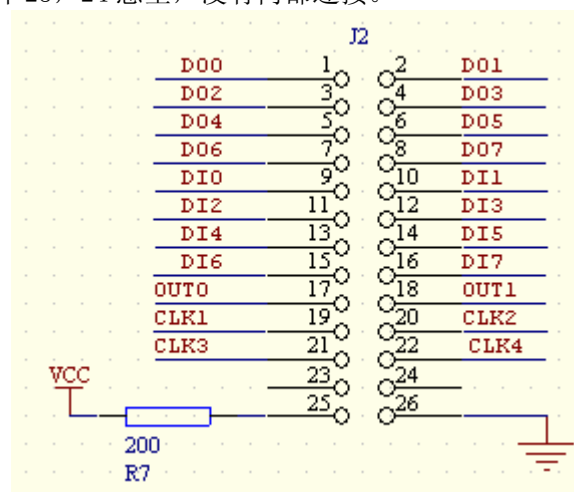


图 4 开关量接线图

J1 是 37 芯的 D 形接头，模拟量输入与模拟量输出接口，如图 5 所示，针脚的定义是：

针 19 是模拟地线，是模拟量输入 AD 与模拟量输出 DA 的公共地线。

针 9, 28, 10, 29 是 4 路 DA 输出。当 DA 输出为 4 通道模式时，9, 28, 10, 29 对应 0, 1,

2, 3 这四个通道, 当只有 2 通道时, 与 RBH7202 模式相同, 28 为 DA0, 29 为 DA1, 9 和 10 为地线。

当采用单端时, 标号 J_1 至 J_16 对应通道 0—15; 标号 J_20 至 J_35 对应通道 16 至 31。

当采用双端时, 标号 J_1 与 J_20 组成第一通道、J_2 与 J_21 组成第二通道、依此类推, J_16 与 J_35 组成最后一个通道。

需要注意的是, 图 5 所示标号与 J1 事物标号完全一致, J1 接近 PCI 金手指一端为 1, 远离金手指的一端为 19。图 5 所示为从外观观察 J1 时的定义。

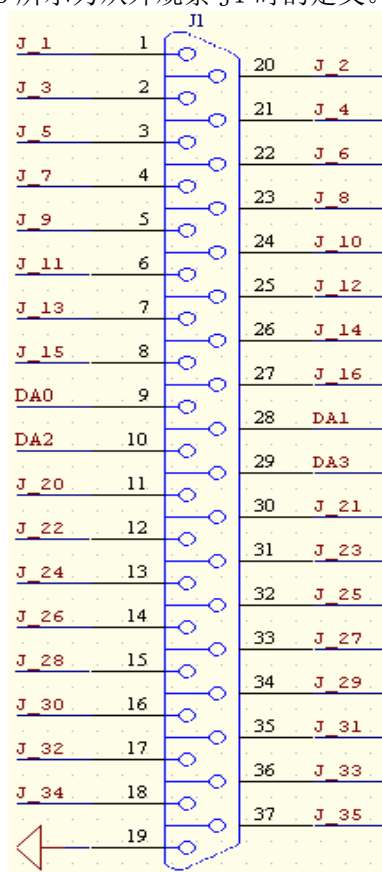


图 5 J1 模拟信号接线插座

五、软件使用说明

pci 总线有即插即用的特点, 为用户使用本卡提供了很多方便, 对于大多少用户, 可以直接采用本公司提供的驱动软件, 可以实现数据采集功能。

在工业控制中, 往往需要软件独立控制硬件, 并且需要了解硬件的物理地址, 以便于实现可靠的控制和采集功能。有些用户的软件在 Dos 下编制, 也需要了解板卡的起始地址和中断号, 因此, 这里介绍获取板卡口地址和中断号的方法。建议新使用的用户不要使用这个方法, 建议使用 WINDOWS 下的编程方法。

以下为了说明方便, 设定口地址为 IOBase。并用 C 语言介绍。

1、程控放大器设置

程控放大器有两种类型可以选择, 一种是 PGA204, 增益分别是 1、10、100、1000; 另一种是 PGA205, 放大倍数分别是 1、2、4、8。两种放大器的放大倍数由控制端 A0、A1

控制，当 A1A0 分别是 00、01、10、11 时，对应的放大倍数为 1、10、100、1000 或 1、2、4、8。

IOBase 的 D5 位=A0;

IOBase 的 D6 位=A1;

2、 软件触发启动 A/D 的方法

outportb(IOBase+10, 0); //立即启动 AD

3、 A/D 完成的查询方法

while(! (inportb(IOBase+6) &0x2)); // 等待 bit1==1, 表明 A/D 芯片正在工作，当为高电平时，表明 A/D 完成。

4、 读取 A/D 结果的方法

inportb(IOBase+4) //读取结果低8位

(inportb(IOBase+5))//读取结果高 4 位;

5、 AD 综合编程实例

假设程控放大器是PGA205，设定增益=2，那么A1A0=01;

int ControlByte;//控制字节，对应的地址是IoBase+9。其D0—D4是通道号，D5是PGA的A0，D6是PGA的A1。

ControlByte=0;//控制字节初始化

ControlByte=ControlByte &0xBF;//A1=D6=0;

ControlByte=ControlByte|0x20;//A0=D5=1;

outportb(IOBase+9, ControlByte);//使程控放大器的A0=0 A0=1

ControlByte=ControlByte&0xE0;

outportb(IOBase+9, ControlByte);//选择通道0

//设置定时器

outportb(IOBase+3, 0x36);//设置定时器的控制字

i = MainFreq/FrqSamp0;

outportb(IOBase, i& 0xff); //送控制字节

outportb(IOBase, (i>>8)&0xff);

for(i=0;i<NumChn+1;i++) {采集所有通道的信号

outportb(IOBase+10, 0); //立即启动 AD

ControlByte=ControlByte|((i+1)&0x1f);

outportb(IOBase+9, ControlByte); //选择下一个通道

while(! (inportb(IOBase+6) &0x2)); // 等待 bit1==1,

if(i>=1) //跳过第一次转换

TmpADBuff[i-1] = inportb(IOBase+4) +//读取结果

((inportb(IOBase+5)&0xf) <<8);

}

ControlByte=ControlByte&0xe0;

outportb(IOBase+9, ControlByte); //设置到通道0

6、开关量输入编程说明

8路开关量输入的口地址是IOBase+7。

对应的D0-D7是8路开关量输入的电平状态，当对应的针是高电平时，读入的是1，当对应的针脚是低电平时，输入的是0。对应的针脚是J1的9至16，在板上有文字标注，特别要注意J1的方向，J1的下方为第1针。

7、开关量输出编程说明

8路开关量输出的口地址是IOBase+11。

对应的D0-D7是8路开关量输出的电平状态，当对应的位输出1时，当对应的针是高电平时，当对应的位输出0时，对应的针脚输出低电平。对应的针脚是J1的1至8，在板上有文字标注，特别要注意J1的方向，J1的下方为第1针。

〈二〉WINDODW95 下的软件说明

本板提供了很完善的 WIN98/2000/NT/XP 驱动程序，采用动态链接库的方式，用户使用方便、快捷，所提供的 DEMO 软件，能满足大量的实际需要，如实时控制、波形显示、波形记录等。

在 WINDOWS 下编程使用瑞博华公司统一的软件编程接口，该接口在 WINDWOS 可以实现高速、连续、实时采集，并且适用于 WINDOWS 下的各种编程语言，包括 VB, VC, DELPHI, C++BUILDER, LABVIEW 等等，编程接口采用几个主要的接口函数，包括初始化、启动、读取数据、关闭等，这些函数的说明与使用见编程指南。

作为编程实例，主要提供 VB 编程代码，用户可以通过该软件针对具体的板卡进行参数设置。

1、读取板卡信息

由于 RBH7272 有多种分型号，因此，用户有时需要了解板卡的硬件信息。

如下面的程序所示，通过函数 IOCTL 函数的 70 号命令就可以读取板卡的信息，在返回函数中可以读出板卡型号，板卡速度，AD 位数，分型号（模式），其中模式就是板卡的分型号。函数的使用可以参见 VB 例程。一般情况下用户不必使用该函数。

```
InBuff(0) = 70 ' 命令号
InBuff(1) = 5 ' num
InBuff(2) = 0 ' offset
i =DllIOctl(100, InBuff(0), 100, OutBuff(0))
str1 = ""
Dim ilong, ilong1, ilong2 As Long
ilong1 = OutBuff(0)
ilong2 = OutBuff(1)
ilong = ilong2 * 100 + ilong1
str1 = "板卡型号: " + Str(ilong)
str1 = str1 + vbCrLf
str1 = str1 + "板卡速度 KSPS:" + Str(OutBuff(2) * 10) + vbCrLf
```

```

str1 = str1 + "AD 位数:" + Str(OutBuff(3)) + vbCrLf
str1 = str1 + "模式:" + Str(OutBuff(4)) + vbCrLf
str1 = str1 + "模式参数:" + Str(OutBuff(5)) + vbCrLf

```

2、初始化功能

```

i = DllInitial(0, 5, 0, 0) ' 初始化采集功能

```

3、启动采集

```

NumBuf = 10 ' 缓冲区个数
NumSamp = 500 ' 每个缓冲区采样点数, 一个采样点是指所有的通道采样一次
begchn = 0 ' 起始通道
NumChn = 8 ' 通道数
FrqSamp = 10000 ' 采样频率
FrqFilter = 0 ' 滤波器频率
AmpGain = 1 ' 放大器增益

```

当放大器增益=0 时, 表示采集的数据为原始数据, 数据没有经过校准。

```

i = DllStartIntr(NumBuf, NumSamp, begchn, NumChn, FrqSamp, FrqFilter,
AmpGain)

```

4、取采集结果

连续采集模式下的数据读取方法

```

Private Sub TimerAD_Timer()
Dim i
Dim j
Dim ilong
Dim NumFill As Integer
Dim iPoint As Long
NumFill = DllQueryBuf ' 填满的缓冲区个数
If (NumFill > NumBuf) Then
MsgBox ("数据丢失")
End If
For i = 1 To NumFill
j = DllADResult(ADBuf(0)) ' 首先将数据保存到临时数组 ADBuf() 中
For j = 1 To NumSamp * NumChn
' 将数据从临时缓冲区中, 保存在长缓冲区中, 去掉序列号, 所以从 1 开始, 而不是从 0 开始
RecordBuf(RecordPtr + j - 1) = ADBuf(j)
Next j
RecordPtr = RecordPtr + NumSamp * NumChn ' 指针指向下一次起始地址
If (RecordPtr >= (RecordBlock * (NumSamp * NumChn))) Then RecordPtr = 0
' 缓冲区满, 就从头开始

```

```

Next i

TotalBuf = TotalBuf + NumFill ' 总共接收到的块个数
lblBlock.Caption = Str(TotalBuf) ' 显示总共接收到的块的个数
lblptr.Caption = Str(RecordPtr) ' 显示数据在大缓冲区中的指针位置

' 从最近接收缓冲区中取数据, 显示结果
If NumFill > 0 Then
    For i = 0 To NumChn - 1
        ad_chn(i) = ADBuf(1 + (NumSamp - 1) * NumChn + i) ' 取最新的数据
        v_chn(i) = (ad_chn(i) - 2048) / 4095 * 10
    Next i
    For i = 0 To 3
        lblAD(i).Caption = Format(v_chn(i), "##.000")
    Next i
End If

' 从大缓冲区 RecordBuf 中取数, 显示结果
If NumFill > 0 Then
    iPoint = 1 ' =1 表示最新的一个点, =2 表示前一个点, 依次类推
    For i = 0 To NumChn - 1
        If (RecordPtr = 0) Then
            ad_chn(i) = RecordBuf(RecordBlock * NumChn - NumChn * iPoint + i)
        Else
            ad_chn(i) = RecordBuf(RecordPtr - NumChn * iPoint + i)
        End If
        v_chn(i) = (ad_chn(i) - 2048) / 4095 * 10
    Next i

End If

' 取最后一个缓冲区, 各个通道在不同时刻的值
If NumFill <= 0 Then Exit Sub
For i = 0 To NumChn - 1
    ilong = 0
    For j = 0 To NumSamp - 1
        AD_Arr(i, j) = ADBuf(1 + j * NumChn + i)
        ilong = ilong + AD_Arr(i, j) ' 计算累加和
        V_Arr(i, j) = (AD_Arr(i, j) - 2048) / 4095 * 10
    Next j
Next i

```

```

Next j
    ilong = ilong / NumSamp ' 计算平均值
    ad_chn(i) = ilong
    v_chn(i) = (ad_chn(i) - 2048) / 4095 * 10
Next i

For i = 0 To 3
    lblAD(i).Caption = Format(v_chn(i), "##.000")
Next i

```

End Sub

查询方式下的采集结果读取

```

' -----模拟量采集例程-----
i = DllRbh_ADResult(NumChn, ADResult(0)) ' 结果放置在数组 ADResult() 中, 该
数组是一个全局变量, 在 module 中定义
' 处理结果
For i = 1 To 32
    v_chn(i - 1) = (ADResult(i - 1) - 2048) / 4095 * 10 ' 将 AD 值换算成电
压值
    ChnStr(i - 1) = Format(v_chn(i - 1), "00.000") ' 得到显示用的字符
Next i
' 显示结果, 在屏幕上显示结果
For j = 0 To 3
    lblAD(j).Caption = ""
    For i = 0 To 7
        lblAD(j).Caption = lblAD(j).Caption + " " + ChnStr(i + 8 * j)
    Next i
Next j

```

5、开关量输入功能

```

' -----读取开关量输入, 并显示结果例程-----
Dim DI0, DI1, DI2 As Integer ' 8 通道的开关量输入
i = DllRbh_DI(7, IOResult(0))
DI0 = IOResult(0)

```

6、开关量输出功能

```

DOData(0) = (Val("&H" + txtSW0.Text)) And &HFF ' 要输出的开关量字节
i = DllRbh_DO(0, DOData(0))

```

7、脉冲功能

8、DA 输出功能

```
chn = Val(txtDACH.Text)
If chn > 3 Then chn = 3
If chn < 0 Then chn = 0

DAValue = Val("&H" + txtDAValue.Text)
i = DllRbh_DA(DAValue, chn)
```

9、停止采集功能

```
i = DllStopIntr
```

六、注意事项

1、由于 RBH7272 采用 CPLD 控制, 可以实现各种功能, 因此有多种 RBH7272 分型号, 建议购买时, 先确定型号, 如 RBH7272_7200 表示与 RBH7200 兼容的版本, RBH7272_7202 表示与 RBH7202 兼容, RBH7272-7 为脉冲计数型。建议使用前阅读相应的硬件说明书。每种板卡对应不同的启动程序。

2、不要带电插拔该板。

3、长期不使用时, 建议从计算机中拔下该板, 妥善保管。

七、出库清单

1、RBH7272 板一块

2、光盘一张 (内含 demo 程序、驱动程序 SYS 等)