

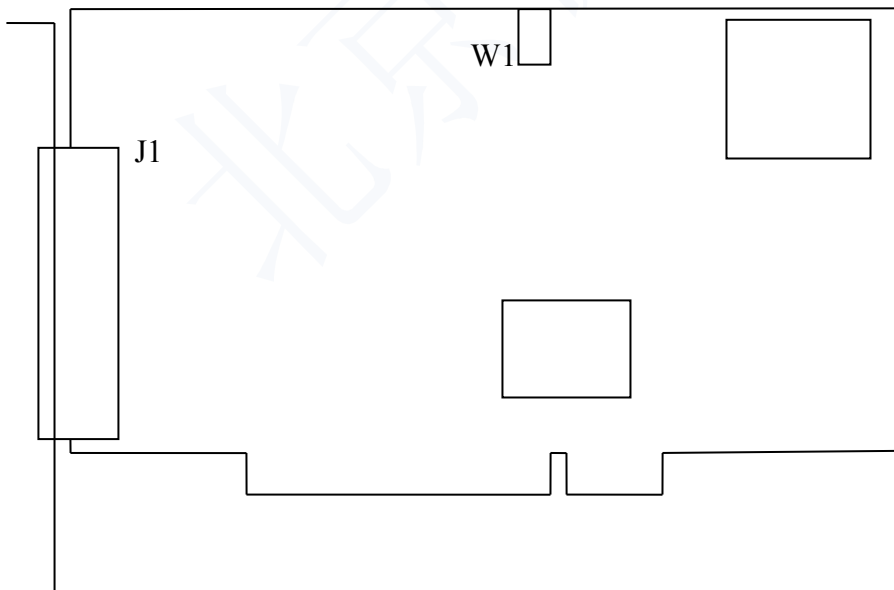
UA244 型 A/D 卡使用说明

第一章. 概述

UA244 型 A/D 卡是我公司开发的 PCI 总线数据采集产品, 该卡采用美国新型 24 位 A/D 转换芯片, 电路设计及布线讲究, 带有大容量 FIFO 缓冲存储器, 指标先进, 性能稳定。该卡的显著特点是: 高精度, 高速度, 各通道同步, 每通道带有独立程控放大器, 低噪声高动态范围, 可测量微小信号, 性价比高。可广泛应用于科学实验和工业控制领域。

一. 主要技术指标

- 分辨率 : 24bit
- 4 模入通道, 同步采集, 可差动输入, 基本量程: $\pm 6V$
- 每通道最高采样频率: 60KHz, 采样频率连续可调, 步长 1Hz
- 每通道装有独立前置控放大器: 1、2、4、8 倍 (或 1、10、100、1000 倍)
- 可任意设定采样通道
- 32KB 先进先出(FIFO)缓冲存储器, 可实现自动数据块采集
- 可外触发采样, TTL 点平上升沿。
- 提供丰富的编程支持, 适用于 WindowsXP, Win7-32, Win7-64 等操作系统



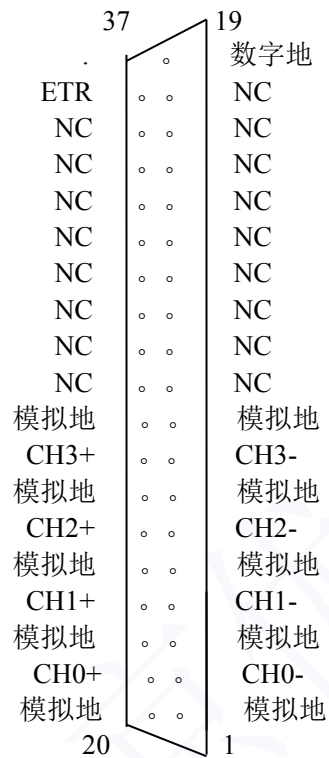
UA244 卡布局图

第二章.硬件说明

一. A/D 卡的连接

UA248 采集卡装有一只 37 芯 D 型插座, 用于连接模拟量输入定义如下:

J1 模拟输入信号插座定义 (37D 孔型):



其中 CH0- CH0+ 为 0 通道差动输入负端和正端, 其余类推。
单端输入时可将负端接模拟地。

ETR 为外部触发信号输入, 对应地为数字地。外触发信号为 5V TTL 电平, 上升沿有效。

二. A/D 卡的调整

精密独立通道结构 A/D 卡一般用软件校正零点，软件校准绝对电压精度。

卡上装有 1 只电位器 W1，用于 A/D 模入范围大致调整，同时作用于 4 个通道。

第三章. 安装驱动程序及应用软件

一. **安装驱动程序.** UA244 在 WINDOWS XP/ WIN7 下工作时，需要安装设备驱动程序。一台计算机在第一次安装 UA244 卡时需要人工安装设备驱动程序，以后再开机时设备驱动程序会自动安装。

第一次安装驱动程序：在关机状态下，把 UA244 卡插入计算机，然后后打开计算机电源，当 WINDOWS 正常运行时会发现新设备，此时按屏幕提示安装光盘中根目录下的驱动程序。

二. **安装应用软件.** UA244 卡还提供了其它软件工具，如：动态连接库，演示应用软件等，应该把它们拷入硬盘独立目录以便使用。

三. **运行程序.** 安装软件后，您就可以运行程序，测试一下采集卡了。双击硬盘中 UA244 文件夹，运行其中 UA244.exe 文件即可进入双通道采集示波界面，此时在模拟通道 0 或 1 接入信号源或用手触摸输入端便可见到采集波形显示。

点击“Fr.”按钮可改变采集频率。

点击“CH+”按钮可改变通道。

点击“数据.”按钮可显示电压值。

点击“G+.”按钮可改变放大倍数。

点击画面可冻结波形，再次点击画面可回复动态采集。

其余可按照界面提示操作。

第四章. UA244 卡的编程

UA244 可使用各种 WINDOWS 编程工具编程, 如: VC++, VB, C#... 等. 为方便使用我们随板提供 24 位 PCI 采集卡专用的动态连接库 UA240.DLL. 在此动态连接库中提供了许多简洁高效的采集和控制函数, 支持 UA244 采集卡的各种功能, 用户可简单方便地调用这些函数完成各种数据采集工作. 本章对 UA244 卡用到的函数进行详细说明, 并给出调用例子。

一. 函数一览:

OpenUA240	打开 UA240 设备
CloseUA240	关闭 UA240 设备
start_24	设置参数启动采集
read_24	读取采集数据
stop_24	停止采集
get_statu	读采集状态
outb	I/O 输出一字节
inb	I/O 读一字节

二. 函数的详细说明

1. OpenUA240

函数形式:

HANDLE_stdcall OpenUA240(int kn)

说 明: 该函数用于打开 UA244 设备, 取得设备句柄, 对 UA244 操作时, 应首先执行该函数.

输入参数: kn 卡号从 0 开始, 用于多卡操作, 一块卡时 kn=0

返 回 值: 返回设备句柄, 这个句柄在调用其他函数时需要使用.

2. CloseUA240

函数形式:

HANDLE_stdcall CloseUA200(HANDLE h204)

说 明: 该函数用于关闭 UA244, 在结束对 UA244 操作时, 应执行该函数关闭设备.

输入参数: 在打开设备时得到的设备句柄.

返 回 值: 可忽略.

3. set_gain

函数形式:

HANDLE_stdcall set_gain(HANDLE h204, UCHAR ch, UCHAR gn)

说 明: 该函数用于设置各通道前置放大器倍数.

输入参数: ch 通道号, 可以是 0、1、2、3
gn 增益代码 0-- 1 倍; 1-- 2 倍; 2-- 4 倍, 3-- 8 倍
返回值: 无.

4. start_24

函数形式: void stdcall start_24(HANDLE h240, UCHAR ch, UCHAR chn, long fr, UCHAR tr)

说明: 该函数用于设置参数启动采集, 执行后采集卡开始采集, 数据存储内部缓存, 此函数执行后可用 read_24 函数读取数据, 可用 get_statu 函数读取采集状态。

输入参数: h240 在打开设备时得到的设备句柄.

ch 采集通道, 该字节定义如下:

bit7							bit0
0	0	0	0	CH3	CH2	CH1	CH0

该字节 8bit 对应 4 个通道位置如上, 对应通道设 1 为采集, 0 为不采集, 可任意设置。设置后, 采集数据按从低通道到高通道顺序循环放置。

chn 采集通道数

fr 通道采集频率, 可以是 1--60000 (Hz)

tr 外触发 0, 不用外触发; 1, 使用外触发

返回值: 无.

5. read_24

函数形式: void stdcall read_24(HANDLE h240, int *addat, unsigned long leng)

说明: 该函数用于读取数据采集. 在执行 start_24 后, 可多次调用该函数读取数据, 只要间隔不是太长, 数据将是连续的, 可根据 get_statu 获得采集状态读取. 采集缓冲区为 32KB, 半满为 16KB. 但你可以一次读取更多的数据。

输入参数: h240 在打开设备时得到的设备句柄.

leng 采样数据的总点数(一点为 32 位字), 该参数是长整数, 意味着只要数据数组开的足够大, 可以一次采集几十兆字节以上的数据.

输出参数: *addat 数据数组, 为 32 位有符号数组, 采集数据存放于数据数组. 24 位采集数据放在低 24bit (右对齐) 数据与电压对应关系为: -8388608 to +8388607 代表 -6V - +6V 电压 (放大倍数为 1 时)

返回值: 无

6. get_statu

函数形式: UCHAR stdcall get_statu(HANDLE h240)

说明: 读取 UA244 缓冲存储器状态

输入参数: h240 在打开设备时得到的设备句柄.

返回值: 状态字节: bit6 0 存储器空; 1 非空

bit7 0 存储器半满; 1 存储器未半满
bit5 0 存储器满; 1 存储器未半满
其余 bit 无定义

7. outb

函数形式: void _stdcall outb(HANDLE h240, UCHAR addr, UCHAR dat)

说明: 在 UA244 I/O 口输出一字节

输入参数: h240 在打开设备时得到的设备句柄

addr ua244 I/O 口地址

dat 输出的数据字节

返回值: 无

8. inb

函数形式: UCHAR _stdcall inb(HANDLE h240, UCHAR addr)

说明: 在 UA244 I/O 口读一字节

输入参数: h240 在打开设备时得到的设备句柄

addr ua244 I/O 口地址

返回值: 8 位数据

以上描述了 UA240.DLL 中的基本函数, 您可以调用这些函数进行各种数据采集工作. 在 UA240 资料上您可以看到调用这些函数的例子.

二. 函数调用例子

1. VC++ 6 调用例子

这是一个控制台方式的调用例子. 编译时需加上 UA240.h, UA240.lib

```
// UA244T.cpp
#include "windows.h"
#include "stdafx.h"
#include "ua240.h"

HANDLE h240;

int main(int argc, char* argv[])
{
    short int ch;
    short int i;
```

```

int addat[8192];

printf("Hello World!\n");

h240=OpenUA240(0); // 打开设备,得到句柄
set_gain(h240,0,0); // 通道 0 增益 1
set_gain(h240,1,0); // 通道 1 增益 1
set_gain(h240,2,0); // 通道 2 增益 1
set_gain(h240,3,0); // 通道 3 增益 1

start_24(h240, 15,4 , 1000, 0); // 4 个通道采集, 1000Hz, 不用外触发

read_24(h240, addat, 1024); //读数据

// 显示数据
for(i=0;i<8;i++){
for(ch=0;ch<4;ch++){
printf("%2.5f ",addat[i*4+ch]*0.000305176 / 256*.6);
}
printf("\n");
}

stop_24(h240); //停止采集

CloseUA240(h240);
}

```

头部文件:

```

//ua240.h
HANDLE _stdcall OpenUA240(int kn);
HANDLE _stdcall CloseUA240(HANDLE h240);
UCHAR _stdcall get_statu(HANDLE h240);
void _stdcall set_gain(HANDLE h240, UCHAR ch, UCHAR gn);
void _stdcall start_24(HANDLE h240, UCHAR ch, UCHAR chn, long fr, UCHAR tr);
void _stdcall read_24(HANDLE h240, int *addat, unsigned long leng);
void _stdcall stop_24(HANDLE h240);
void _stdcall outb(HANDLE h240, UCHAR addr, UCHAR dat);
UCHAR _stdcall inb(HANDLE h240, UCHAR addr);

```