

ADT-852

三轴运动控制卡

用户手册

深圳市众为兴数控技术有限公司

版权声明

本用户手册的所有部分，其著作财产权归属众为兴数控技术有限公司（以下简称众为兴）所有，未经众为兴许可，任何人不可任意地仿制，拷贝、誊抄或转译。本用户手册没有任何形式的担保，立场表达或其他暗示。若有任何因本用户手册或其所提到之产品的所有信息，所引起的直接或间接的资料流出，利益损失或事业终止，众为兴及其所属员工恕不担负任何责任。除此之外，本用户手册提到的产品规格及资料仅供参考，内容有可能会更新，恕不另行通知。

商标声明

用户手册中所涉及到的产品名称仅作识别之用，而这些名称可能是属于其它不同的商标或版权，在此声明如下：

- ※ INTEL, PENTIUM 是 INTEL 公司的商标。
- ※ WINDOWS, MS-DOS 是 MICROSOFT 公司产品标识。
- ※ ADT-852 是众为兴公司的商标。
- ※ 其它未提到的标识，均属各注册公司所拥有。

版权所有，不得翻印。

众为兴数控技术有限公司

目 录

一、	概要	3
二、	硬件安装	4
三、	电气连接	5
四、	软件安装	13
五、	ADT852 库函数列表	20
六、	ADT852 库函数详解	21
七、	运动控制开发编程示例	25

备注：本说明书适用于 ADT-852 卡之 C 版。

一、概要

产品简介

ADT852 卡是基于 PCI 总线的三轴伺服/步进控制卡。

脉冲输出方式可为（脉冲+方向）方式，最大脉冲频率 50KHz。

位置管理采用计数器位数高达 32 位，最大范围

-2,147,483,648~+2,147,483,647。

速度控制可用定速和直线加减速。

每轴有 8 个输入信号，包括 2 个正负限位信号，6 个通用输入信号。

提供 DOS、WINDOWS95/98/NT/2000 开发库，可用 VC++、VB、BC++ 等进行软件开发。

主要性能:

- 3 轴伺服/步进电机控制，每轴可独立控制，互不影响
- PCI 总线 即插即用
- **最大脉冲输出频率为 200KHz**
- 脉冲输出为（脉冲+方向）方式
- **3 轴 32 位计数，最大计数范围**
-2,147,483,648~+2,147,483,647
- 直线加/减速
- 2-3 轴直线插补
- 运动中可以实时读出逻辑位置、驱动速度
- 每轴有 8 入 4 出数字 I/O，除 2 个限位信号外均可作为通用 I/O 使用，数字输出可用于伺服开启，伺服报警复位等信号
- 支持 DOS、WINDOWS95/98/NT/2000 等操作系统

应用范围:

1. PCB 钻孔铣边
2. 基于 PC 的数控系统

二、硬件安装

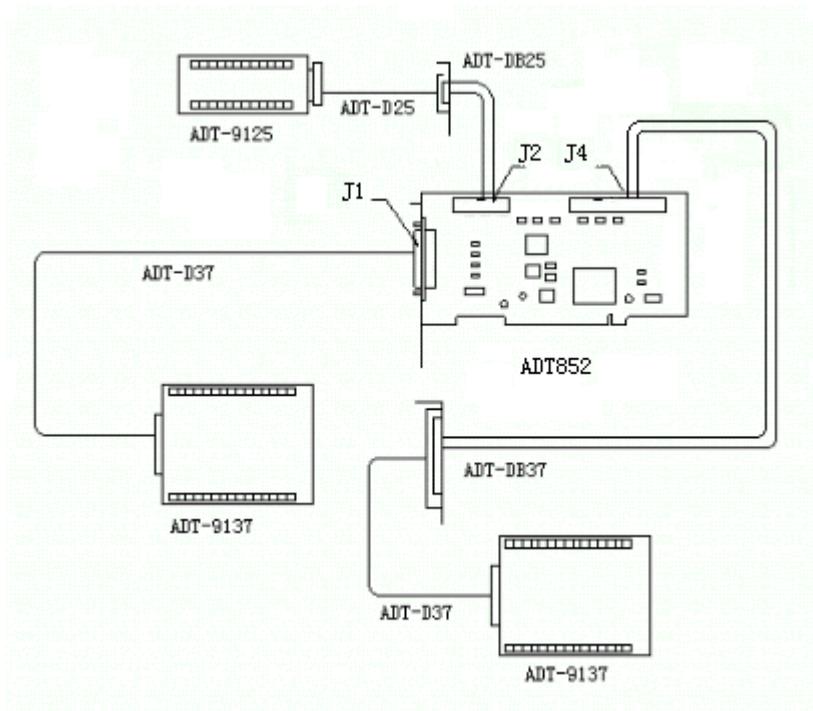
(一) 检查配件

1. ADT-852 用户手册（本手册）
2. ADT-852 四轴 PCI 总线高性能运动控制卡
3. ADT-852 用户光盘
4. ADT-9137 37 芯信号接线板 1 块
5. ADT-D37 37 芯屏蔽连接线 1 条
6. ADT-9137 37 芯信号接线板 1 块（选件）
7. ADT-D37 37 芯屏蔽连接线 1 条（选件）
8. ADT-9125 25 芯开关信号接线板 1 块（选件）
9. ADT-D25 25 芯屏蔽连接线 1 条（选件）
10. ADT-DB25 25 芯扁平线 1 条（选件）
11. ADT-DB37 37 芯扁平线 1 条（选件）

(二) 安装

1. 关闭电脑电源（注：ATX 电源需总电源关闭）。
2. 打开电脑机箱后盖。
3. 选择一条未占用的 PCI 插槽,插入 ADT-852 卡。
4. 检查 ADT-852 卡的金手指是否完整插入 PCI 插槽,拧整螺丝。
5. 根据用户情况决定是否安装 J2、J4 接口线。

三、电气连接



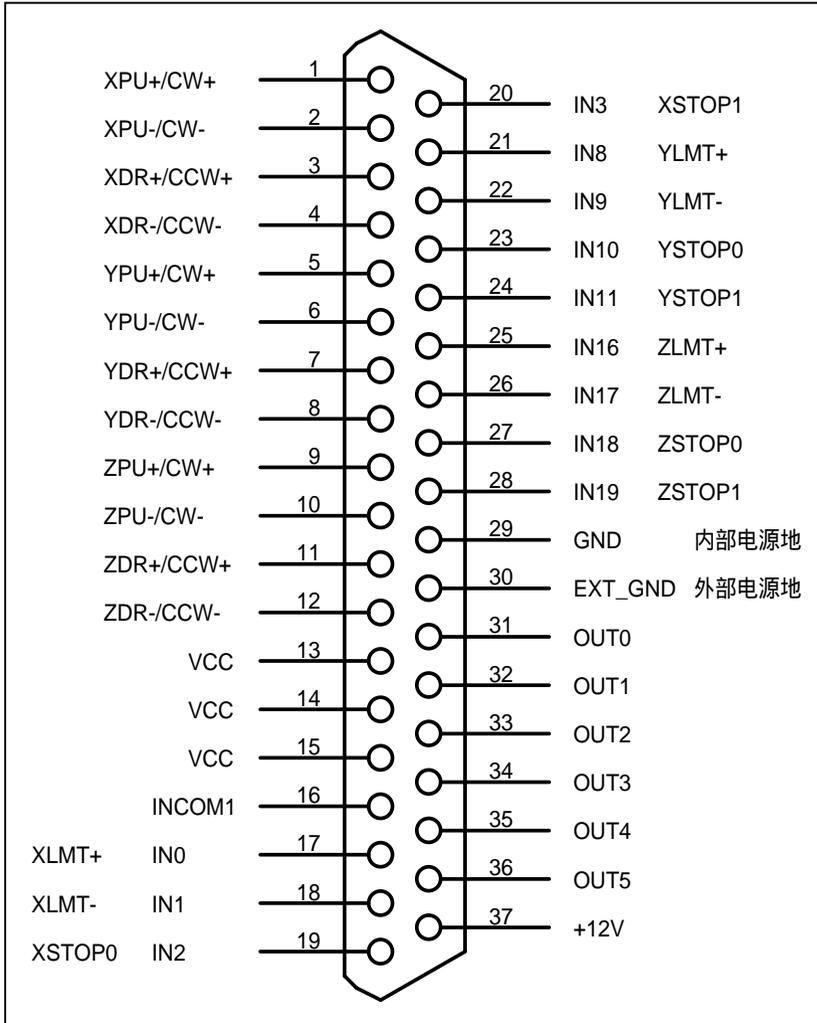
一块 Adt852 卡有三个输入/输出接口，其中 J1、J4 为 37 针插座，J2 为 25 针插座。

J1 为 X、Y、Z 轴的脉冲输出、开关量输入和开关量输出 OUT0-OUT5 的信号接线，

J4 为开关量输入的信号接线。

J2 为开关量输出 OUT6-OUT15 信号的接线
信号定义如下：

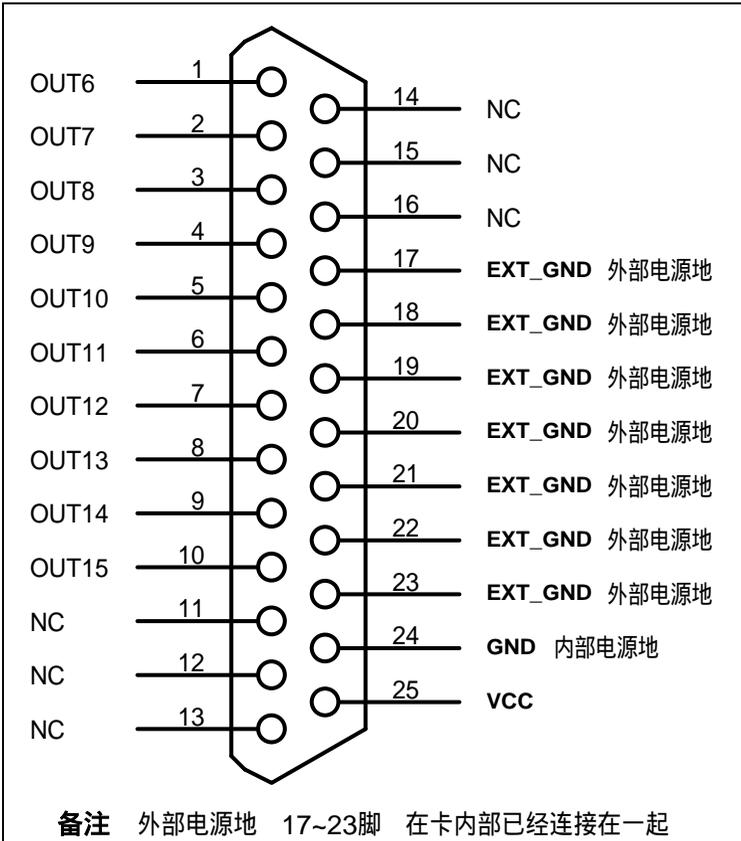
J1 线号说明



线号	符号	说明
1	XPU+/CW+	X 脉冲信号+
2	XPU-/CW-	X 脉冲信号-
3	XDR+/CCW+	X 方向信号+
4	XDR-/CCW-	X 方向信号-
5	YPU+/CW+	Y 脉冲信号+
6	YPU-/CW-	Y 脉冲信号-
7	YDR+/CCW+	Y 方向信号+
8	YDR-/CCW-	Y 方向信号-
9	ZPU+/CW+	Z 脉冲信号+
10	ZPU-/CW-	Z 脉冲信号-
11	ZDR+/CCW+	Z 方向信号+
12	ZDR-/CCW-	Z 方向信号-
13	VCC	内部+5V 电源正端 不可接外接电源
14	VCC	内部+5V 电源正端 不可接外接电源
15	VCC	内部+5V 电源正端 不可接外接电源
16	INCOM1	光耦输入公共端（下面的信号）
17	IN0 (XLMT+)	X 正向限位信号
18	IN1 (XLMT-)	X 反向限位信号
19	IN2	通用输入信号 2
20	IN3	通用输入信号 3
21	IN8 (YLMT+)	Y 正向限位信号
22	IN9 (YLMT-)	Y 反向限位信号
23	IN10	通用输入信号 10
24	IN11	通用输入信号 11
25	IN16 (ZLMT+)	Z 正向限位信号
26	IN17 (ZLMT-)	Z 反向限位信号
27	IN18	通用输入信号 18
28	IN19	通用输入信号 19

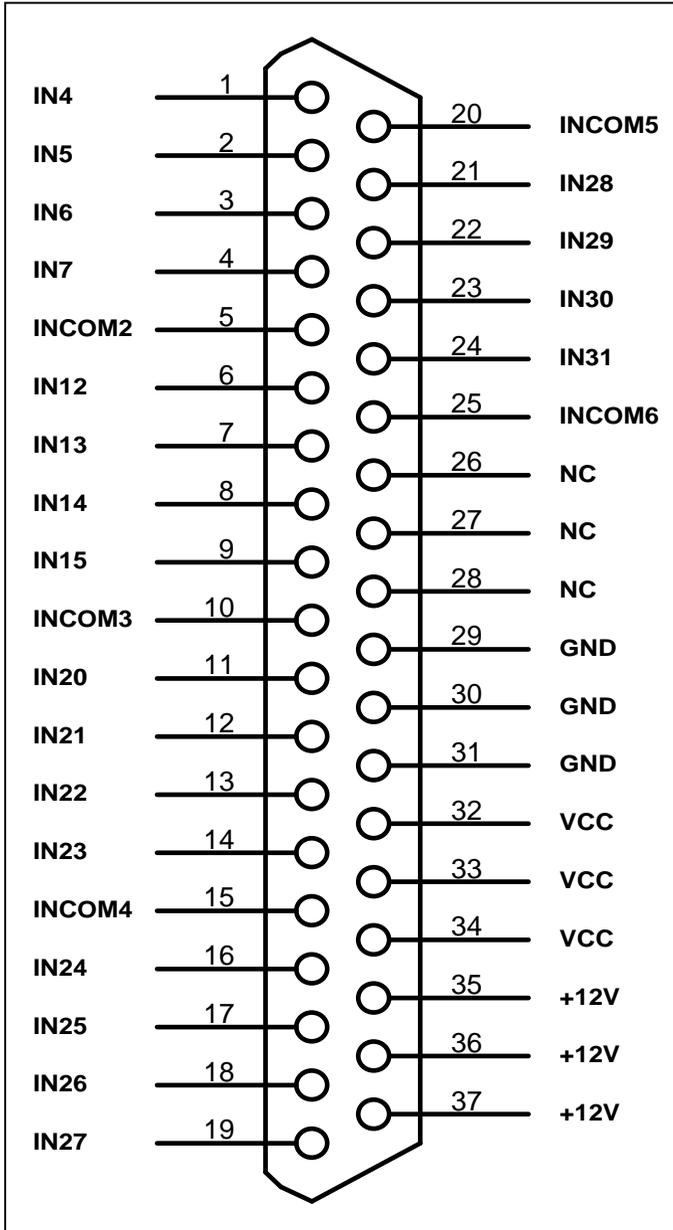
29	GND	内部电源地线
30	EXT_GND	外部电源地线
31	OUT0	开关量输出点
32	OUT1	
33	OUT2	
34	OUT3	
35	OUT4	
36	OUT5	
37	+12V	内部+12V 电源正端 不可接外接电源

J2 线号说明



线号	符号	说明
1	OUT6	开关量输出点
2	OUT7	
3	OUT8	
4	OUT9	
5	OUT10	
6	OUT11	
7	OUT12	
8	OUT13	
9	OUT14	
10	OUT15	
11	NC	未用
12	NC	未用
13	NC	未用
14	NC	未用
15	NC	未用
16	NC	未用
17	EXT_GND	外部电源地线（卡内部已经短接）
18	EXT_GND	
19	EXT_GND	
20	EXT_GND	
21	EXT_GND	
22	EXT_GND	
23	EXT_GND	
24	GND	内部电源地线
25	VCC	内部+5V 电源正端

J4 线号说明



线号	符号	说明
1	IN4	通用输入信号 4
2	IN5	通用输入信号 5
3	IN6	通用输入信号 6
4	IN7	通用输入信号 7
5	INCOM2	光耦输入公共端 (IN4-IN7)
6	IN12	通用输入信号 12
7	IN13	通用输入信号 13
8	IN14	通用输入信号 14
9	IN15	通用输入信号 15
10	INCOM3	光耦输入公共端 (IN12-IN15)
11	IN20	通用输入信号 20
12	IN21	通用输入信号 21
13	IN22	通用输入信号 22
14	IN23	通用输入信号 23
15	INCOM4	光耦输入公共端 (IN20-IN23)
16	IN24	通用输入信号 24
17	IN25	通用输入信号 25
18	IN26	通用输入信号 26
19	IN27	通用输入信号 27
20	INCOM5	光耦输入公共端 (IN24-IN27)
21	IN28	通用输入信号 28
22	IN29	通用输入信号 29
23	IN30	通用输入信号 30
24	IN31	通用输入信号 31
25	INCOM6	光耦输入公共端 (IN28-IN31)
26	NC	未用
27	NC	未用
28	NC	未用

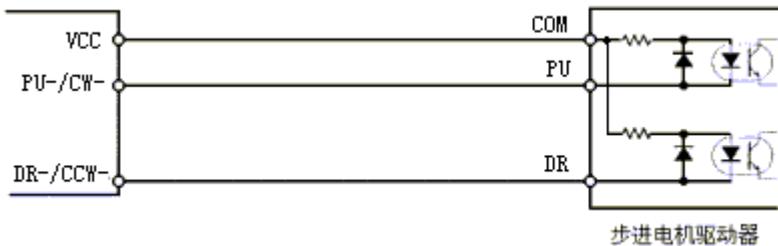
29	GND	内部电源地线
30	GND	内部电源地线
31	GND	内部电源地线
32	VCC	内部+5V 电源正端 不可接外接电源
33	VCC	内部+5V 电源正端 不可接外接电源
34	VCC	内部+5V 电源正端 不可接外接电源
35	+12V	内部+12V 电源正端 不可接外接电源
36	+12V	内部+12V 电源正端 不可接外接电源
37	+12V	内部+12V 电源正端 不可接外接电源

a. 脉冲/方向输出信号的连接

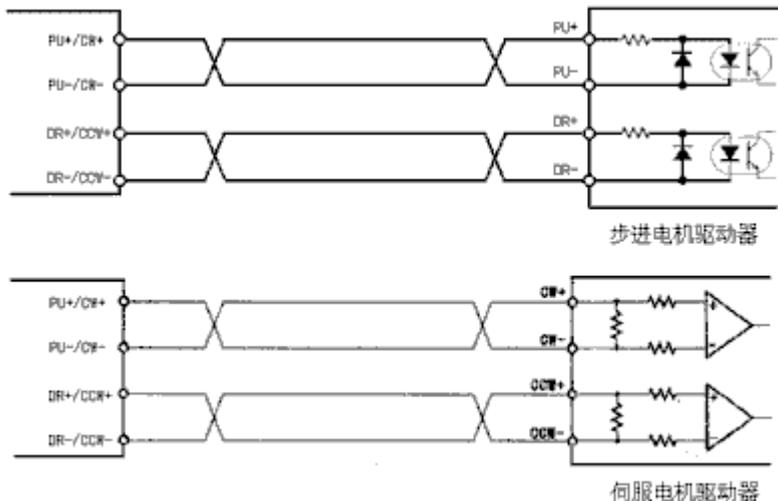
脉冲输出为差动输出方式

可与步进/伺服驱动器很方便的连接

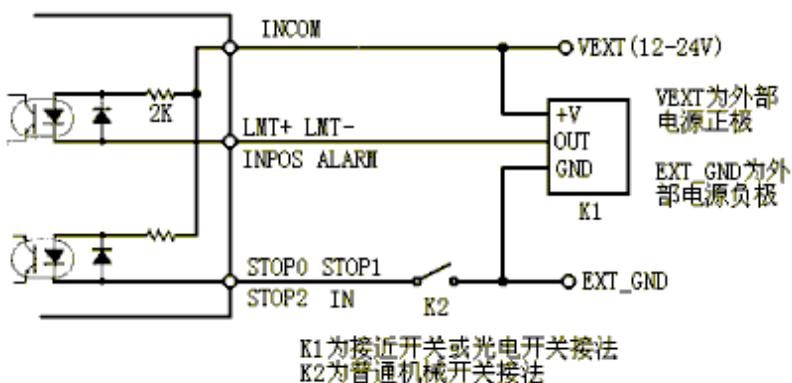
下图为脉冲与方向的阳极已连通的接法



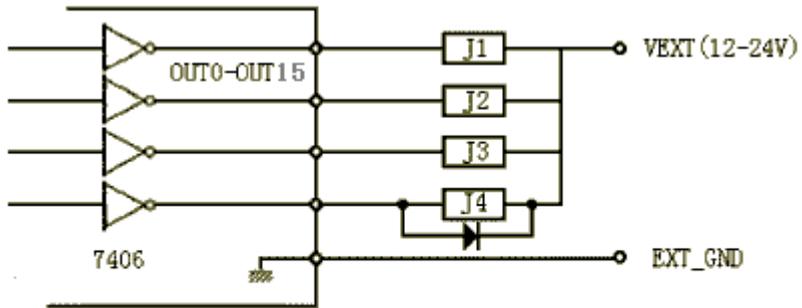
下图为脉冲与方向信号独立的接法，建议采用此种方法，因为是差动接法，抗干扰性强。



b. 数字输入的连接



c. 数字输出的连接



如果为感性负载，如继电器等，应在负载两端加续流二极管，如J4所示

一、 软件安装

ADT852 卡在 WINDOWS95/98/NT/2000 下必须安装驱动程序才能使用，在 DOS 下则无须安装驱动程序。

■ WIN95/98 下驱动程序的安装

以下用 Windows98 中文版为例，说明驱动程序的安装。

在将 ADT852 卡安装到电脑上的 PCI 插槽后，电脑开机后应发现新硬件，出现如下图的提示：



单击“下一步”后出现如下画面：



按上图所示选择, 再单击“下一步”, 出现如下画面:



以上假设 E 盘为光盘, 按上图输入路径, 或者使用“浏览”寻找 ADT850.INF 文件所在的路径, 单击“下一步”后, 应出现如下画面



单击“下一步”后出现如下画面：



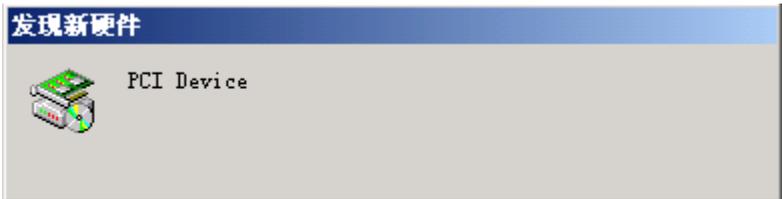
单击“完成”后，即完成 ADT852 卡的安装。

■ Winnt 下驱动程序的安装

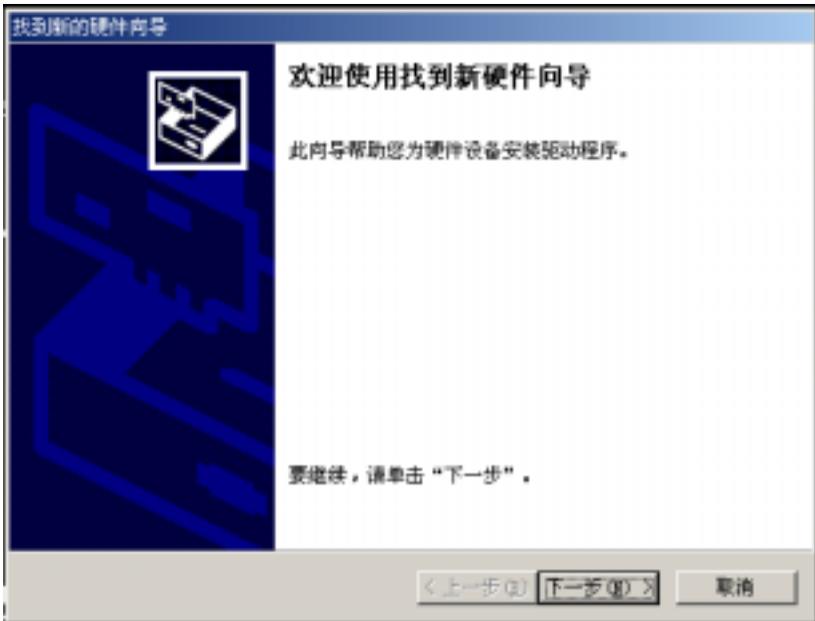
■ Win2000 下驱动程序的安装

以下用 Windows2000 Professional 中文版为例,说明驱动程序的安装,其余版本的 Windows2000 与此类似。

在将 ADT852 卡安装到电脑上的 PCI 插槽后,开机时应以管理员身份登陆,电脑开机后应发现新硬件,出现如下图的提示:



然后出现如下画面:



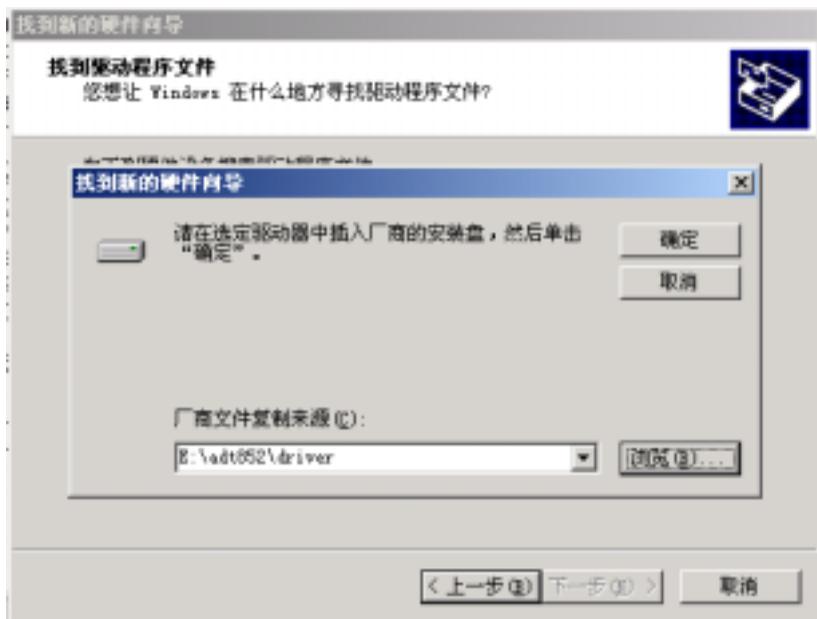
单击“下一步”后,再显示如下画面,



按上图选择后，再单击“下一步”后，出现如下画面



再按上图选择“指定一个位置”，单击“下一步”，出现如下画面



以上假设 E 盘为光盘，或者使用“浏览”寻找 ADT850.INF 文件所在的路径，单击“确定”后，应出现如下画面



单击“下一步”后出现如下画面:



单击“完成”后, 即完成 ADT852 卡的安装。

二、ADT852 库函数列表

```
int init_board();
```

```
int set_startv(int axis,unsigned long v);
```

```
int set_speed(int axis,unsigned long v);
```

```
int set_add(int axis,unsigned long v);
```

```
int set_mode(int axis,int v);
```

```
int set_inp_startv(unsigned long v);
```

```
int set_inp_speed(unsigned long v);
```

```
int set_inp_add(unsigned long v);
```

```
int pmove(int axis,long pos);
```

```
int sudden_stop(int axis);
```

```
int dec_stop(int axis);
```

```
int inp_move2(int axis1,int axis2,long pos1,long pos2);
```

```
int inp_move3(int axis1,int axis2,int axis3,long pos1,long  
pos2,long pos3);
```

```
int get_status(int axis,int *value);
```

```
int get_speed(int axis,unsigned long *speed);
```

```
int get_pos(int axis,long *pos);
```

```
int reset_pos(int axis);
```

```
int write_bit(int a,int v);
```

```
int read_bit(int a);
```

三、ADT852库函数详解

1. 初始化 adt852 卡

int init_board(void);

返回值 0: 正确
1: 错误, 可能未插卡或地址设置不正确

2. 设置起始速度

int set_startv(int axis,unsigned long v);

axis 轴号 (1-3)
v 起始速度值, 范围 20-50000;
返回值 0: 正确 1: 参数错误 2: 卡错误

3. 设置驱动速度

int set_speed(int axis,unsigned long v);

axis 轴号 (1-3)
v 驱动速度值, 范围 20-50000;
返回值 0: 正确 1: 参数错误 2: 卡错误

4. 设置加速度

int set_add(int axis,unsigned long v);

axis 轴号 (1-3)
v 加速度值
返回值 0: 正确 1: 参数错误 2: 卡错误

5. 设置驱动模式

int set_mode(int axis,int v);

axis 轴号 (1-3)
v 0: 正逻辑 1: 负逻辑
返回值 0: 正确 1: 参数错误 2: 卡错误

如果驱动方向与实际方向相反, 可用此函数将运动方向反转

6. 设置插补起始速度

int set_inp_startv(unsigned long v);

v 起始速度值，范围 20-50000;

返回值 0: 正确 1: 参数错误 2: 卡错误

7. 设置插补驱动速度

int set_inp_speed(unsigned long v);

v 驱动速度值，范围 20-50000;

返回值 0: 正确 1: 参数错误 2: 卡错误

8. 设置插补加速度

int set_inp_add(unsigned long v);

v 加速度值

返回值 0: 正确 1: 参数错误 2: 卡错误

9. 定量驱动

int pmove(int axis,long pos);

axis 轴号 (1-3)

pos 脉冲数，范围 (-268435455 to 268435455)

返回值 0: 正确 1: 参数错误 2: 卡错误

10. 立即停止驱动

int sudden_stop(int axis);

axis 轴号 (1-3)

返回值 0: 正确 1: 参数错误 2: 卡错误

11. 减速停止驱动

int dec_stop(int axis);

axis 轴号 (1-3)

返回值 0: 正确 1: 参数错误 2: 卡错误

12. 两轴直线插补

int inp_move2(int axis1,int axis2,long pos1,long pos2);

axis1,axis2 轴号 (1-3)

pos1,pos2 脉冲数

返回值 0: 正确 1: 参数错误 2: 卡错误

13. 三轴直线插补

```
int inp_move3(int axis1,int axis2,int axis3,long pos1,long pos2,long pos3);
```

axis1,axis2,axis3 轴号 (1-3)

pos1,pos2,pos3 脉冲数

返回值 0: 正确 1: 参数错误 2: 卡错误

14. 获取轴状态

```
int get_status(int axis,int *value);
```

axis 轴号 (1-3)

value 状态值的指针

1: 正在驱动 0: 驱动结束

返回值 0: 正确 1: 参数错误 2: 卡错误

15. 获取轴当前速度

```
int get_speed(int axis,unsigned long *speed);
```

axis 轴号 (1-3)

value 速度值的指针

返回值 0: 正确 1: 参数错误 2: 卡错误

16. 获取当前位置

```
int get_pos(int axis,long *pos);
```

axis 轴号 (1-3)

value 位置值的指针

返回值 0: 正确 1: 参数错误 2: 卡错误

17. 当前位置清零

```
int reset_pos(int axis);
```

axis 轴号 (1-3)

返回值 0: 正确 1: 参数错误 2: 卡错误

18. 输出单点

int write_bit(int a,int v);

a 输出点 (0-15)

v 0: 低 1: 高

返回值 0: 正确 1: 参数错误 2: 卡错误
输出数对应接线图相应的输出号。

19. 读单个输入点

int read_bit(int a);

a 输入点 (0-31)

返回值 0: 低电平

 1: 高电平

 -1: 错误

输入点数含义如下:

IN0: X轴正限位 IN1: X轴负限位

IN8: Y轴正限位 IN9: Y轴负限位

IN16: Z轴正限位 IN17: Z轴负限位

其余与接线图对应, 为通用输入点

四、运动控制开发编程示例

所有函数均为立即返回，而不是等待驱动结束后才返回，所有驱动操作都是由ADT852卡完成，上位机主要做实时监控及界面处理的工作。

1. 开发DOS下的运动控制系统（Borland C++ 3.1）

DOS下的开发库共有三个文件，在配套光盘的\adt852\dos\lib\目录下，一个是头文件adt852.h，另外两个是库函数，一个是大模式库adt852l.lib（large memory model），一个是巨模式库adt852h.lib（huge memory model），可根据不同的编译环境选择。

a. 单轴定量匀速运动

目的：

让X轴的步进电机以1000 pps的速度运动10000步：

程序如下：

```
#include "adt852.h"
void main()
{
    if(init_board()!=0) return;           //未安装ADT852卡
                                           //以下只对X轴操作

    set_startv(1,1000);
    set_speed(1,1000);                   //如果起始速度大于或等于驱
                                           //动速度，则为匀速运动

    pmove(1,10000);                       //开始驱动

    int s;
    while(1)
    {
        get_status(1,&s);                 //读驱动状态
        if(s==0)break;                   //驱动结束跳出
    }
}
```

```

        ..... //可执行读键盘，显示位置等函数
    }
    return ;
}

```

b. 单轴定量加/减速运动

目的:

让X轴以下列速度运动20000步

起始速度: 2000 pss

驱动速度: 20000 pss

加/减速度: 40000 pss

程序如下:

```

#include "adt852.h"
void main()
{
    if(init_board()!=0) return; //未安装ADT832卡
                                // X轴操作

    set_startv(1,400); //起始速度
    set_speed(1,4000); //驱动速度
    set_acc(1,40000); //加/减速度
    pmove(1,20000); //开始驱动
    int s;
    while(1)
    {
        get_status(1,&s); //读驱动状态
        if(s==0)break; //驱动结束跳出
        ..... //可执行读键盘，显示位置等函数
    }
    return ;
}

```

c. 多轴运动

以上虽为单轴操作，但实际上可同时设置另外几轴的数据，互相之间并不影响，如在X轴驱动时，设置好Y轴的参数，然后驱动Y轴，对X轴的运动不会有任何影响，如此可独立操作san三轴，

d. 两轴直线插补

直线插补与单轴运动基本类似

程序如下：

```
#include "adt852.h"
void main()
{
    if(init_board()!=0) return;          //未安装ADT832卡
    set_inp_startv(1000);
    set_inp_speed(1000);
    inp_move2(1,2,10000,-20000);      //X-Y开始插补
                                        //X正向移动10000步
                                        //Y反向移动20000步

    int s1,s2;
    while(1)
    {
        get_status(1,&s1);              //读X状态
        get_status(2,&s2);              //读Y状态
        if(s1==0 && s2==0)break;      //结束跳出
        .....                          //可执行读键盘，显示位置等函数
    }
    return ;
}
```

e. 三轴直线插补

程序如下：

```
#include "adt852.h"
void main()
{
    if(init_board()!=0) return;          //未安装ADT832卡
    set_inp_startv(1000);
    set_inp_speed(1000);
```

```

inp_move3(1,2,3,10000,-20000,30000);    //X-Y开始插补
                                           //X正向移动10000步
                                           //Y反向移动20000步
                                           //Z正向移动30000步

int s1,s2,s3;
while(1)
{
    get_status(1,&s1);    //读X状态
    get_status(2,&s2);    //读Y状态
    get_status(2,&s3);    //读Z状态
    if(s1==0 && s2==0 && s3==0)break; //结束跳出
    .....                //可执行读键盘，显示位置等函数
}
return ;
}
    
```

2. 用VB开发WINDOWS下的运动控制系统

VB下的开发库共有三个文件，在配套光盘的\adt852\win\vb\目录下，一个是动态库ADT852.DLL，一个是模块文件ADT852.BAS，一个是WindowsNT/2000使用的文件winio.sys，动态库中的函数已在模块文件ADT852.BAS中有声明，基本使用方法与DOS下的编程相同。

详细的例子在用户光盘上。

3. 用VC开发WINDOWS下的运动控制系统

VC下的开发库共有三个文件，在配套光盘的\adt852\win\vc\目录下，一个是静态库ADT852.LIB，一个是头文件ADT852.H，一个是WindowsNT/2000使用的文件winio.sys，动态库中的函数已在头文件ADT852.H中有声明，基本使用方法与DOS下的编程相同。

详细的例子在用户光盘上。