



深圳市雷赛控制技术有限公司
SHENZHEN LEADSHINE CONTROL TECHNOLOGY CO.,LTD

IOC-0640/1280

PCI 总线 I/O 扩展卡

用 户 手 册

Version 1.3

目录

IOC-0640/1280PCI 总线 I/O 扩展卡	1
用户手册.....	1
0 关于本手册	1
1 引言.....	2
2 硬件系统概述.....	3
2.1 硬件结构尺寸	4
2.2 硬件配置	4
2.3 硬件接口电路	6
2.3.1 通用数字输入信号接口 INPUT	6
2.3.2 通用数字输出信号接口 OUTPUT.....	6
2.4 硬件安装	8
3 驱动程序安装.....	9
3.1 驱动程序概况	9
3.2 驱动程序安装步骤	9
4 软件系统概述.....	12
4.1 驱动程序概况:	12
4.2 WIN98/2000/NT/7 下 IOC-0640/1280 的中断机制.....	12
4.3 演示程序	15
4.4 I/O 控制函数库说明	15
4.4.1 初始化函数	16
4.4.2 通用 I/O 控制函数.....	16
4.4.3 板卡中断函数	17
4.4.4 I/O 口中断设置函数	18
4.4.5 I/O 口软件滤波函数	20
4.4.6 输出口电平翻转函数 (仅 IOC-1280 支持)	20
4.4.7 运动函数错误码说明	20
4.5 例子程序	21
5 演示软件及应用.....	22
6 用户系统开发.....	23
6.1 基于 WINDOWS 平台的应用软件结构	23
6.2 VISUAL BASIC 环境下编程	24
6.3 VISUAL C++环境下编程.....	26
7 附录.....	30
7.1 IOC0640/1280 主板硬件信号接线板端口定义表	30
7.2 IOC0640/1280 主板硬件信号接线盒端口定义表	31

7.3	IOC0640 扩展板采用 IOB-3721 接口定义表.....	32
7.4	IOC1280 扩展板采用 IOB-6421 接口定义表.....	32
7.5	版次说明	32

0 关于本手册

本手册旨在帮助你学习 IOC-0640/1280 控制卡的使用，包括软件函数的调用，参数的设置，硬件接口的接线和应用例程软件的编写等。本手册总共分七大章节：

第一章：引言。关于本产品的大概描述和关于本产品的相关申明。

第二章：硬件系统概述。关于本产品硬件相关介绍，包括详细硬件结构尺寸，输出口电平设置，接口电路的接线和硬件的安装。

第三章：驱动程序的安装。介绍 IOC-0640/1280 驱动程序的详细安装步骤。

第四章：软件系统概述。关于本产品软件相关介绍，包括本产号所支持的的系统驱动程序，演示软件

的描述，函数库详细说明和例子子程序的说明。

第五章：演示软件及应用。介绍演示软件的使用和操作。

第六章：用户系统开发。详细介绍用户在 VB 和 VC 开发环境下的简单的应用软件的开发。

第七章：附录。提供详细硬件信号接口表。

1 引言

雷赛 IOC-0640/1280 控制卡是一款基于 ASIC 技术的高性能，高可靠的 PCI 总线 I/O 控制卡，最大支持 (IOC-0640:47 路/ IOC-1280:64 路)通用输入，(IOC-0640:48 路/ IOC-1280:64 路)通用输出，支持输入口中断功能。输入输出均采用光电隔离和滤波技术，可以有效隔离外部电路的干扰，以提高系统的可靠性，并且软件有专用的可调用的滤波函数进行软件滤波。

同时雷赛公司为 IOC-0640/1280 设计了一套易学易用、功能丰富的 I/O 函数库，大大缩短了用户应用软件开发、调试时间。随卡免费提供的 MOTION0640/1280 测试软件，可以演示和测试 IOC-0640/1280 的绝大多数控制功能，方便客户在系充开发时的调试。

特别申明:

本手册版权归深圳市雷赛控制技术有限公司所有，未经雷赛公司书面许可，任何人不得翻印、翻译和抄袭本手册中的任何内容。涉及 IOC-0640/1280 控制卡软件的详尽资料以及每个函数的介绍和范例，请参阅 IOC-0640/1280 控制卡软件使用手册。本手册中的信息资料仅供参考。由于改进设计和功能等原因，雷赛公司保留对本资料的最终解释权；内容如有更改，恕不另行通知！

2 硬件系统概述

雷赛 IOC-0640/1280 是一款兼容 PCI V2.2 标准的 32Bit PCI 标准半长卡，具体外形图如下：

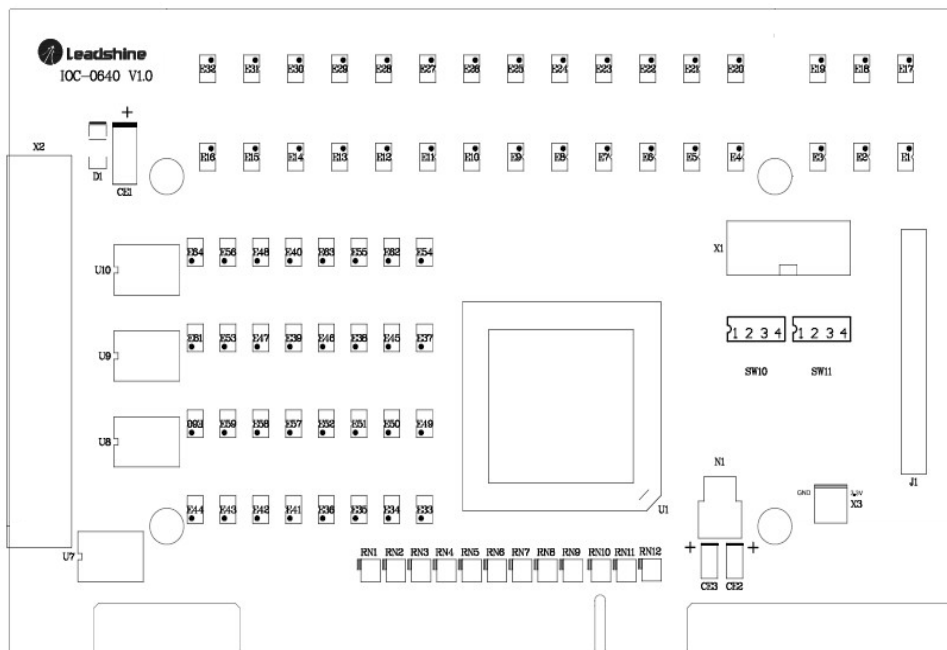


图 2-1 IOC-0640/1280 控制卡外观图

雷赛 IOC-0640/1280 卡为用户提供了最多 (IOC-0640:47 路/ IOC-1280:64 路) 通用数字输入信号和最多 (IOC-0640:48 路/ IOC-1280:64 路) 的通用输出信号，所有输入输出信号接口均采用光电隔离和滤波技术，这样可以有效隔离外部电路的干扰，以提高系统的可靠性，并且软件有专用的滤波函数进行软件滤波，所有输出口可以进行初始电平的设置。硬件系统框图如图 2-2 所示：

IOC-0640/1280 控制卡的硬件系统框图：

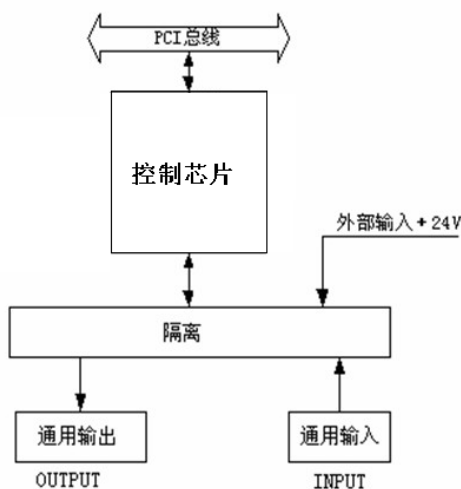


图 2-2 IOC-0640/1280 控制卡系统框图

2.1 硬件结构尺寸

结构尺寸如下：

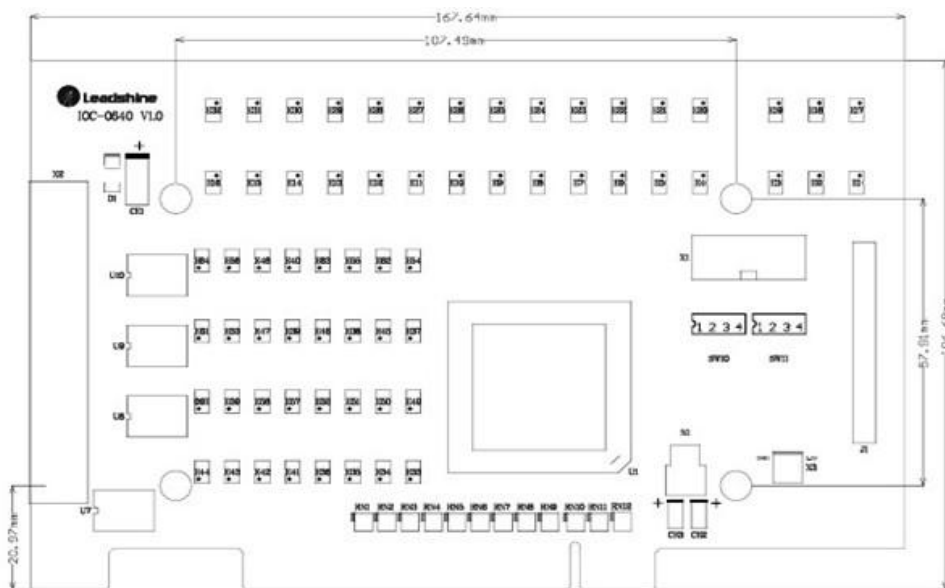


图 2-3 IOC-0640/1280 控制卡结构尺寸图

2.2 硬件配置

IOC-0640/1280 控制卡上面有 2 个拨码开关，分别是 SW10 和 SW11。通过它可以设置通用输出口初始电平的状态，选择“ON”时输出初始电平为低，选择”OFF”时输出初始电平为高。如图 2.4 所示, 以下是设置的详细说明：

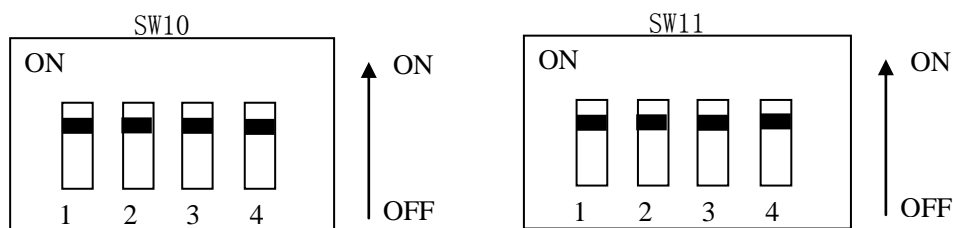


图 2.4 拨码开关 SW10/ SW11 的设置示意图

表 2-1 IOC-0640 输出口拨码开关表

开关号	输出口号	高电平	低电平
SW10.1(主板)	OUT1—OUT6	ON	OFF (出厂设置)
SW10.2(主板)	OUT7—OUT12	ON	OFF (出厂设置)
SW10.3(主板)	OUT13—OUT18	ON	OFF (出厂设置)
SW10.4(主板)	OUT19—OUT24	ON	OFF (出厂设置)
SW11.1(主板)	OUT25—OUT30	ON	OFF (出厂设置)
SW11.2(主板)	OUT31—OUT36	ON	OFF (出厂设置)
SW11.3(主板)	OUT37—OUT42	ON	OFF (出厂设置)
SW11.4(主板)	OUT43—OUT48	ON	OFF (出厂设置)

表 2-2 IOC-1280 输出口拨码开关表

开关号	输出口号	高电平	低电平
SW10.1(主板)	OUT1—OUT6	ON	OFF (出厂设置)
SW10.2(主板)	OUT7—OUT12	ON	OFF (出厂设置)
SW10.3(主板)	OUT13—OUT18	ON	OFF (出厂设置)
SW10.4(主板)	OUT19—OUT24	ON	OFF (出厂设置)
SW11.1(主板)	OUT25—OUT30	ON	OFF (出厂设置)
SW11.2(主板)	OUT31—OUT32	ON	OFF (出厂设置)
SW11.3(主板)	保留		
SW11.4(主板)	保留		
SW10.1(背板)	OUT33—OUT38	ON	OFF (出厂设置)
SW10.2(背板)	OUT39—OUT44	ON	OFF (出厂设置)
SW10.3(背板)	OUT45—OUT50	ON	OFF (出厂设置)
SW10.4(背板)	OUT51—OUT56	ON	OFF (出厂设置)
SW11.1(背板)	OUT57—OUT62	ON	OFF (出厂设置)
SW11.2(背板)	OUT63—OUT64	ON	OFF (出厂设置)
SW11.3(背板)	保留		
SW11.4(背板)	保留		

2.3 硬件接口电路

IOC-0640/1280 卡为用户提供了最多 (IOC-0640:47 路/ IOC-1280:64 路) 通用数字输入信号和最多 (IOC-0640:48 路/ IOC-1280:64 路) 的通用输出信号, 所有输入输出信号接口均采用光电隔离和滤波技术, 这样可以有效隔离外部电路的干扰, 以提高系统的可靠性, 并且软件有专用的滤波函数进行软件滤波。

2.3.1 通用数字输入信号接口 INPUT

IOC-0640/1280 卡为用户提供了通用数字输入信号, 用于开关信号、传感器信号或其它信号的输入。其接口电路加有光电隔离元件, 可以有效隔离外部电路的干扰, 以提高系统的可靠性。

通用数字输入信号接口原理图如图 2-5 所示:

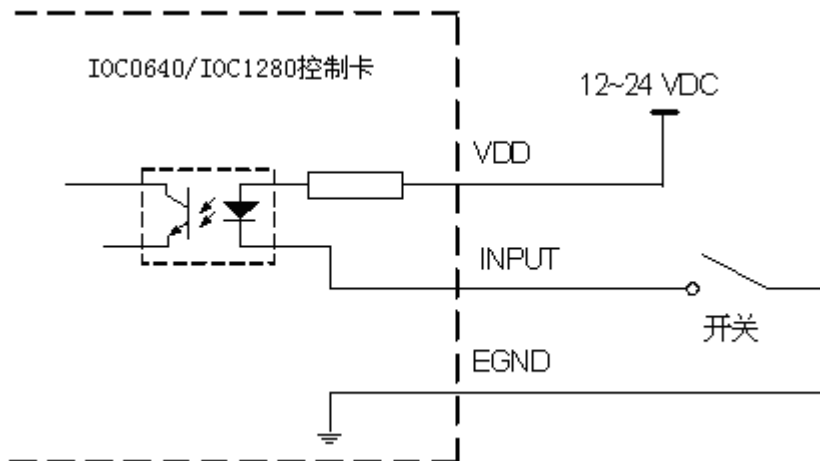


图 2-5 INPUT 信号输入原理图

2.3.2 通用数字输出信号接口 OUTPUT

IOC-0640/1280 控制卡为用户提供了通用数字输出信号, 由 ULN2803 驱动, 可用于对继电器、电磁阀、信号灯或其它设备的控制。其接口电路都加有光电隔离元件, 可以有效隔离外部电路的干扰, 提高了系统的可靠性。

输出端口 OUTPUT 可设置上电时的初始电平, 详见: [2.2 硬件配置](#)

IOC-0640/1280 卡的通用数字输出信号控制常用元器件的接法如下:

1、通发光二极管

通用数字输出端口控制发光二极管时, 需要接一限流电阻 R, 限制电流在 10MA 左右, 电阻值大约在 2K 到 5K 左右, 根据使用的电源来选择, 电压越高, 使用的电阻值越大些。原理图如图 2-6 所示:

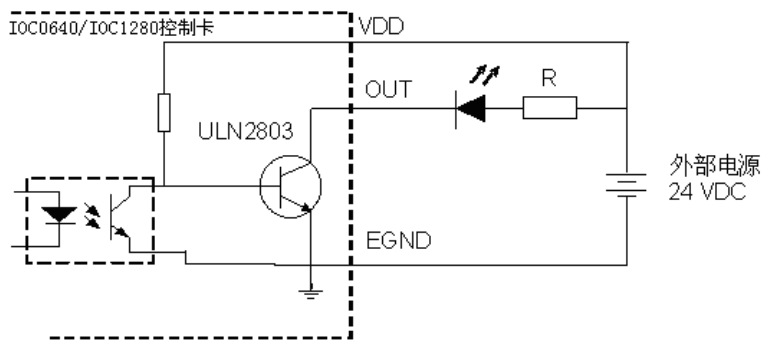


图 2-6 输出口接发光二极管

2、灯丝型指示灯：

通用数字输出端口控制灯丝型指示灯时，为提高指示灯的寿命，需要接预热电阻 R，电阻值的大小，以电阻接上后，输出口为 1 时，灯不亮为原则。原理图如图 2-7 所示：

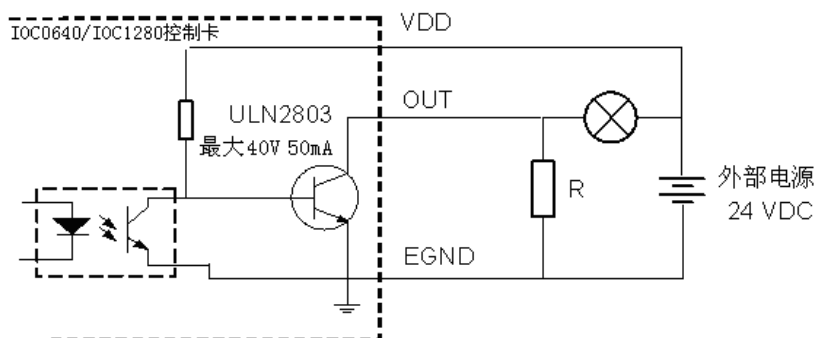


图 2-7 输出口接灯丝型指示灯

3、小型继电器：

继电器为感性负载，必须并联一个续流二极管，以保护 IOC-0640/1280 卡的输出口驱动元件 ULN2803。继电器接线图如图 2-8：

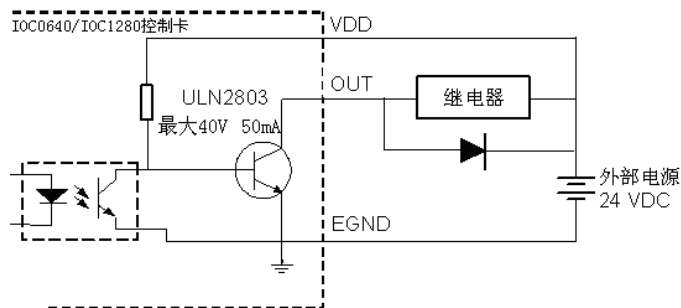


图 2-8 接小型继电器的原理图



注意： 1. 在使用通用数字输出端口时，切勿把外部电源直接接至通用数字输出端口上，否则，会损坏 ULN2803。

2.4 硬件安装

雷赛 (LeiSai) IOC-0640/1280 控制卡硬件结构遵从 32bit PCI 卡结构标准，其安装方法类同普通 32bit PCI 卡的安装，具体参考步骤如下：

- 1) 打开 IOC-0640/1280 的包装，参考 [3.1 章节](#) 的说明，按照实际应用的需求，完成硬件配置。
- 2) 使用辅助接口的用户，请将辅助接口与 IOC0640/1280 对应的插座连接，并确保连接牢固，可靠。
- 3) 触摸地线，完全释放操作员身上的静电，带好防静电手套。
- 4) 确定 PC 机已经关闭，以及一切与 PC 相连的设备也已关闭。
- 5) 打开 PC 机的机箱。
- 6) 选择一个靠近处理器的 32bit PCI 插槽，将 IOC0640/1280 垂直插入插槽中。
- 7) 将 IOC-0640/1280 用螺钉紧固在 PC 机机箱上，确保紧固、可靠。
- 8) 使用辅助接口的用户，请将辅助接口也用螺钉紧固在 PC 机机箱上，确保紧固、可靠。
- 9) 盖上 PC 机机箱，至此硬件安装完毕！

3 驱动程序安装

3.1 驱动程序概况

雷赛公司为IOC-0640/1280用户配套提供Windows NT/2000/XP/7等操作系统环境下的卡的驱动程序和运动控制函数动态链接库，为客户在不同的应用系统中能开发出适合自己的应用控制系统。

3.2 驱动程序安装步骤

雷赛 (LeiSai) IOC-0640/1280运动控制卡的驱动程序遵从32bit PCI卡驱动标准，其安装方法类同普通32bit PCI卡驱动程序的安装方法。下面以Windows XP操作系统下的安装为例：

- 1) 确保已经通读本手册，并参照 [2.4 硬件安装](#) 将 IOC-0640/1280 硬件安装好。
- 2) 启动 PC 机，进入 Windows XP 操作系统。
- 3) 操作系统系统提示发现新硬件，并弹出图 3-1 所示的“找到新硬件安装向导”界面。
- 4) 将 IOC-0640/1280 所配光盘放入光驱中，在相应的目录下，打开“inf_winxp”资料夹，找到 regist2K.bat 批处理文件，双击运行。（注意：Windows7 操作系统下，则在相应的目录下，打开“inf_Win7”资料夹，找到“REG_Win7.bat”批处理文件，单击鼠标右键，选择“以管理员身份运行(A)”）
- 5) 选择“从列表或指定位置安装（高级）（S）”，点击“下一步”，系统将弹出“找到新硬件安装向导”另一界面，如图 3.2



图3-1 新的硬件安装向导界面

- 6) 在图3-2所示的向导中，点选“在这些位置中搜索最佳驱动程序”并勾选上“在搜索中包括这个位置”，点击浏览。
- 7) 在弹出的“浏览文件夹”的窗口中，找到光盘中 IOC-0640/1280 \INF 目录后，点击“确定”按钮后系统将弹出“找到新硬件安装向导”另一界面，如图 3.3 所示：

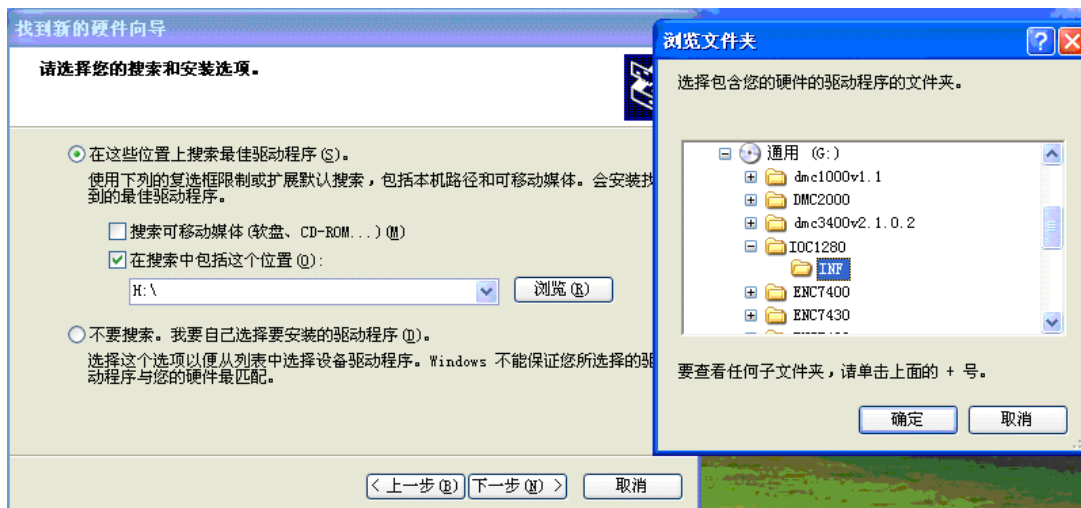


图3-2浏览并选择安装文件夹INF

8) 在图3-3中，点击“下一步”按钮后系统将弹出“找到新硬件安装向导”另一界面，如图3.4:

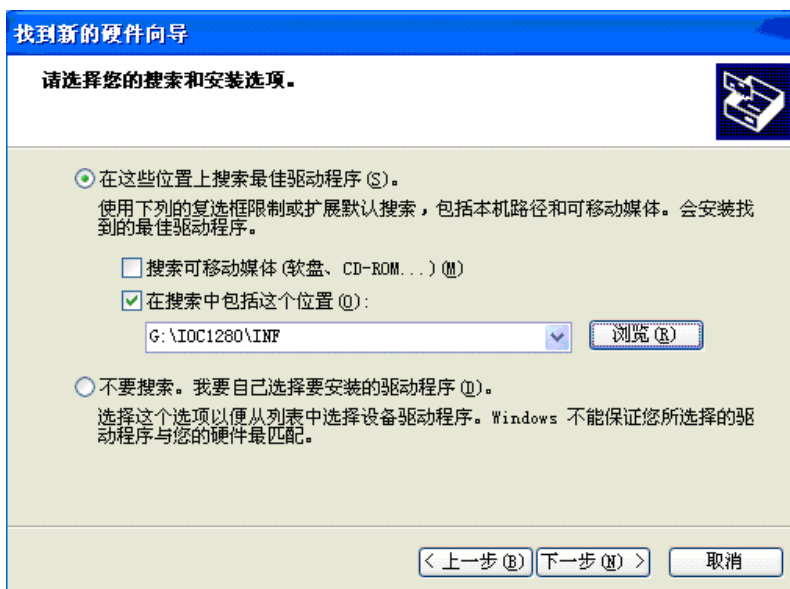


图3-3 开始安装驱动软件

9) 如图3-4所示，向导正在安装LeiSai IOC-0640/1280运动控制卡的驱动程序。



图3-4 安装驱动软件进行中

10) 等待安装完成，显示如图3-5所示界面，点击完成。



图3-5 完成IOC-0640/1280控制卡的安装

11) 至此雷赛（LeiSai）IOC-0640/1280 运动控制卡的驱动程序安装完成，可以正常使用了。

4 软件系统概述

雷赛 (LeiSai) IOC-0640/1280 控制卡软件系统包括：软件驱动程序，中断说明,演示程序， I/O 控制函数库，例子程序。

4.1 驱动程序概况：

雷赛公司为IOC-0640/1280用户配套提供Windows NT/2000/XP/7等操作系统环境下的卡的驱动程序和运动控制函数动态链接库，为客户在不同的应用系统中能开发出适合自己的应用控制系统，卡驱动程序具体安装方法请参考：[3 驱动程序安装](#)

4.2 WIN98/2000/NT/7 下 IOC-0640/1280 的中断机制

IOC-0640/1280 的中断机制如图 4-1 所示, IO 中断触发板卡中断, 然后板卡向上位机提出中断请求, 最后上位机执行中断服务程序对中断处理。IOC0-640 中断功能默认为关闭，因此使用 IOC0640/1280 的中断功能之前，其板卡中断必须打开，同时至少要有一路 IO 中断是打开的。

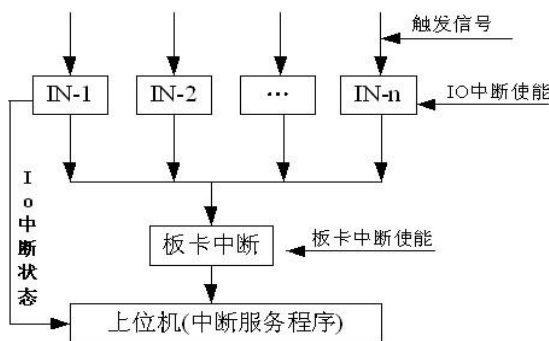


图 4-1 IOC0640/1280 中断响应机制

IOC-0640/1280 卡全部输入口都支持中断功能，中断触发方式支持上升沿触发或下降沿触发，触发方式可以通过软件设置, 默认为上升沿触发。

中断服务程序开放给用户自己编写，在设置板卡中断的时候挂接。中断服务程序中必须要清除中断，否则中断一直有效导致中断服务程序持续执行。读取 IO 的中断状态，则中断解除。

受系统效率和中断频率的影响，并不保证每次中断都能激活上位机的中断服务程序，当中断频率很高，而操作系统线程调度慢时，有可能出现多次中断堆叠。一般要求中断频率 $\leq 10\text{KHz}$ 。

详细编程可参考以下代码（在 VC++ 环境下编写），代码的功能是当输入口 1 有中断时，打开输出口 1：

//定义中断服务程序的参数结构（用户可以根据自己的需要定义）

```
typedef struct IOC0640_OPERATE_DATA
{
    WORD cardno; // number of card

    WORD bitno; // number of IO bit
```

```
WORD on_off;    // bit data

} IOC0640_OPERATE_DATA, *IOC0640_OPERATEDATA_HANDLE;

//全局变量

IOC0640_OPERATE_DATA  m_data;

//开启中断函数代码

void CIOC0640 Dlg::OnBnIntEnable()

{

    UpdateData();

    //使能 IO 中断

    ioc_config_intporten(0, 0, 0xffffffff);

    //设置中断服务程序参数

    m_data.bitno = 1;

    m_data.cardno = 0;

    m_data.on_off = 1;

    //使能板中断，并挂接中断服务程序 ioc_int_callback

    ioc_int_enable(nCardNo, ioc_int_callback, (PVOID) &m_data);

}

//关闭中断函数代码

Void CIOC0640 Dlg::OnBnIntDisable()

{

    ioc_int_disable(nCardNo);

}

//中断服务程序

UINT ioc_int_callback(PVOID operate_data)

{
```



```
IOC0640_OPERATEDATA_HANDLE m_OperateData= (IOC0640_OPERATEDATA_HANDLE) operate_data;

DWORD status;

//读取中断状态，同时解除中断

status = ioc_read_intportstatus(nCardNo, 0);

if(status&0x1)//输入口 1 中断有效

{ //打开输出口 1

    ioc_write_outbit(m_OperateData->cardno, m_OperateData->bitno, m_OperateData->on_off);

    return 0;

}

}
```

4.3 演示程序

IOC-0640/1280控制卡演示软件的设计，大大简化了用户的调试过程。将IOC-0640/1280的软件CD盘插进计算机光驱，在相应的目录下，例如“演示界面”，将其全部拷贝到计算机硬盘的任意指定位置后，运行motion1280.exe，即可对控制卡的各项主要功能进行检测、学习、还可以借此软件对您的整个自动化系统I/O进行初步的调试，详细资料参考：[5.演示软件及应用](#)。

4.4 I/O 控制函数库说明

雷赛（LeiSai）为客户能够开发适合自己的应用控制系统，提供了丰富的 I/O 功能函数，用户可以根据自己系统的需要灵活调用不同的 I/O 函数。

IOC-0640/1280 卡的控制函数库包括 5 类 18 个库函数。在本节中，我们对这些函数分类进行详细的说明。如前所述，用户在具体应用中很可能只需调用函数库中的部分函数便可满足其实际需求。

库函数列表如下：

	函数名	描述
初始化函数	ioc_board_init	初始化控制卡
	ioc_board_close	关闭控制卡
通用I/O控制函数	ioc_read_inbit	读取输入口的状态
	ioc_write_outbit	设置输出口的状态
	ioc_read_outbit	读取输出口的状态
	ioc_read_inport	读取输入端口的值
	ioc_read_outport	读取输出端口的值
	ioc_write_outport	设置输出端口的值
板卡中断设置函数	ioc_int_enable	允许控制卡中断
	ioc_int_disable	禁止控制卡中断
I/O口中断设置函数	ioc_config_intbitmode	设置输入口的中断使能和触发方式
	ioc_read_intbitmode	读取输入口的中断使能和触发方式
	ioc_read_intbitstatus	读取输入口的中断状态
	ioc_config_intporten	设置输入端口的中断使能
	ioc_config_intportlogic	设置输入端口的中断触发方式
	ioc_read_intportmode	读取输入端口的中断使能和触发方式
	ioc_read_intportstatus	读取输入端口的中断状态
滤波函数	ioc_set_filter	设置输入口滤波
翻转函数	ioc_reverse_outbit	输出口状态翻转/延时复位（仅IOC-1280支持）

下面我们对这些函数分类进行详细说明：

4.4.1 初始化函数

int ioc_board_init(void)

功 能：为控制卡分配系统资源，并初始化 IOC0640/1280 卡。

参 数：无

返回值：卡数,(0 - 7)，其中 0 表示没有卡

void ioc_board_close(void)

功 能：关闭 IOC0640/1280 卡，释放系统资源

参 数：无

返回值：无

4.4.2 通用 I/O 控制函数

int ioc_read_inbit(WORD cardno, WORD bitno)

功 能：读取指定控制卡的某一位输入口的电平状态

参 数： cardno 指定控制卡号，范围（0 — N - 1 ,N 为卡数）

bitno 指定输入口位号（取值范围：1—47）

返回值：0 表示低电平；1 表示高电平

WORD ioc_write_outbit(WORD cardno, WORD bitno,WORD on_off)

功 能：对指定控制卡的某一位输出口置位

参 数： cardno 指定控制卡号，范围（0 — N - 1 ,N 为卡数）

bitno 指定输出口位号（取值范围：1—48）

on_off 输出电平：0—表示输出低电平，1—表示输出高电平。

返回值：错误代码

int ioc_read_outbit(WORD cardno, WORD bitno)

功 能：读取指定控制卡的某一位输出口的的电平状态

参 数： cardno 指定控制卡号，范围（0 — N - 1 ,N 为卡数）

bitno 指定输入口位号（取值范围：1—47）

返回值：0 表示低电平；1 表示高电平。

long ioc_read_inport(WORD cardno,WORD m_PortNo)

功 能: 读取指定控制卡的全部通用输入口的电平状态

参 数: cardno 指定控制卡号, 范围 (0 — N - 1 ,N 为卡数)

m_PortNo 端口号,范围(0-1)

返回值: m_PortNo 为0时,bit0–bit31 位值分别代表第 1–32 号输入端口值。

m_PortNo 为1时,bit0–bit14 位值分别代表第 33–47 号输入端口值。

long ioc_read_outport(WORD cardno,WORD m_PortNo)

功 能: 读取指定控制卡的全部通用输出端的电平状态

参 数: cardno 指定控制卡号, 范围 (0 — N - 1 ,N 为卡数)

m_PortNo 端口号,范围(0-1)

返回值: m_PortNo 为0时,bit0–bit31 位值分别代表第 1–32 号输出端口值。

m_PortNo 为1时,bit0–bit15 位值分别代表第 33–48 号输出端口值。

DWORD ioc_write_outport(WORD cardno,WORD m_PortNo, DWORD port_value)

功 能: 指定控制卡的全部通用输出端的电平状态

参 数: cardno 指定控制卡号, 范围 (0 — N - 1 ,N 为卡数)

m_PortNo 端口号,范围(0-1)

port_value m_PortNo 为0时,bit0–bit31 位值分别代表第 1–32 号输出端口值。

m_PortNo 为1时,bit0–bit15 位值分别代表第 1–48 号输出端口值。

返回值: 错误代码

4.4.3 板卡中断函数

DWORD ioc_int_enable(WORD cardno, IOC0640_OPERATE funcIntHandler, PVOID operate_data)

功 能: 允许指定控制卡的中断

参 数: cardno 指定控制卡号, 范围 (0 — N - 1 ,N 为卡数)

funcIntHandler 中断回调函数指针,

函数指针类型: UINT (*IOC0640_OPERATE)(PVOID operate_data)

operate_data 回调函数传递参数

返回值: 错误代码

DWORD ioc_int_disable(WORD cardno)

功 能: 禁止指定控制卡的中断

参 数: cardno 指定控制卡号, 范围 (0 - N - 1, N 为卡数)

返回值: 错误代码

4.4.4 I/O 口中断设置函数

DWORD ioc_config_intbitmode(WORD cardno, WORD bitno, WORD enable, WORD logic)

DWORD ioc_read_intbitmode(WORD cardno, WORD bitno, WORD *enable, WORD *logic)

功 能: 设置/读取指定控制卡输入口的使能设置和触发方式。

参 数: cardno 指定控制卡号, 范围 (0 - N - 1, N 为卡数)

bitno: 输入口位号(1-47)

enable 输入口中断使能, 0-无效, 1 有效

logic 输入口中断触发逻辑, 0—表示中断信号上升沿有效, 1—表示中断信号下降沿有效。

返回值: 错误代码

int ioc_read_intbitstatus(WORD cardno, WORD bitno)

功 能: 读取指定控制卡输入口的中断状态

参 数: cardno 指定控制卡号, 范围 (0 - N - 1, N 为卡数)

bitno 指定输入口位号 (取值范围: 1—47)

返回值: 输入口中断状态: 0-无效, 1 有效

DWORD ioc_config_intporten(WORD cardno, WORD m_PortNo, DWORD port_en)

功 能: 设置指定控制卡输入端口的使能

参 数: cardno 指定控制卡号, 范围 (0 - N - 1, N 为卡数)

m_PortNo 端口号, 范围(0-1)

port_en m_PortNo 为0时, bit0 - bit31 位值分别代表第 1 - 32 号输入端口的中断使能

m_PortNo 为1时, bit33 - bit47 位值分别代表第 33 - 47 号输入端口的中断使能位值, 0—表示输入口中断禁止, 1—表示输入口中断允许。

返回值: 错误代码

DWORD ioc_config_intportlogic(WORD cardno, WORD m_PortNo, DWORD port_logic)

功 能: 设置指定控制卡输入端口的触发方式

参 数: cardno 指定控制卡号, 范围 (0 - N - 1, N 为卡数)

m_PortNo 端口号, 范围(0-1)。

port_logic 端口号为0时, bit0 - bit31 位值分别代表第1 - 32 号输入端口的中断触发方式。

端口号为1时, bit33 - bit47 位值分别代表第33 - 47 号输入端口的中
断触发方式。

位值, 0—表示输入口上升沿有效, 1—表示输入口下降沿有效。

返回值: 错误代码

DWORD ioc_read_intportmode(WORD cardno, uint8 m_PortNo, DWORD *enable, DWORD *logic)

功 能: 读取指定控制卡输入端口的使能和触发方式

参 数: cardno 指定控制卡号, 范围 (0 - N - 1, N 为卡数)

m_PortNo 端口号, 范围(0-1)

port_en m_PortNo 为0时, bit0 - bit31 位值分别代表第1 - 32 号输入端口的中断使能。

m_PortNo 为1时, bit33 - bit47 位值分别代表第33 - 47 号输入端口的中断使能。

位值, 0—表示输入口中断禁止, 1—表示输入口中断允许。

port_logic 端口号为0时, bit0 - bit31 位值分别代表第1 - 32 号输入端口的中断触发方式。

端口号为1时, bit33 - bit47 位值分别代表第33 - 47 号输入端口的中
断触发方式。

位值, 0—表示输入口上升沿有效, 1—表示输入口下降沿有效。

返回值: 错误代码

long ioc_read_intportstatus(WORD cardno, WORD m_PortNo)

功 能: 读取指定控制卡输入端口的中断状态

参 数: cardno 指定控制卡号, 范围 (0 - N - 1, N 为卡数)

m_PortNo 端口号, 范围(0-1)

返回值: m_PortNo 为0时,bit0-bit31 位值分别代表第1-32号输入端口的中断状态。

m_PortNo 为1时,bit0-bit14 位值分别代表第33-47号输入端口的中断状态。

4.4.5 I/O口软件滤波函数

DWORD ioc_set_filter(WORD cardno, double filter)

功 能: 设置指定控制卡的输入口滤波

参 数: cardno 指定控制卡号, 范围(0 - N - 1, N为卡数)。

filter, 滤波频率, 设置范围 1-100KHZ。

返回值: 错误代码。

4.4.6 输出口电平翻转函数 (仅 IOC-1280 支持)

DWORD ioc_reverse_outbit(WORD cardno, WORD bitno, double ms_time)

功 能: 翻转指定控制卡的某一位输出口电平, 然后延时指定的翻转时间后, 恢复原来的电平状态。以第一个输出口 OUT1, 翻转时间为 200 毫秒为例: OUT1 当前电平为低电平 (0) 时, 调用 ioc_reverse_outbit (cardno, 1, 200) 后, OUT 电平翻转为高电平 (1), 延时 200 毫秒后, OUT1 自动恢复为低电平 (0); OUT1 当前电平为高电平 (1) 时, 调用 ioc_reverse_outbit (cardno, 1, 200) 后, OUT 电平翻转为低电平 (0), 延时 200 毫秒后, OUT1 自动恢复为高电平 (1);

输入: cardno:卡号,从0开始

bitno: 指定输出口位号 (取值范围: 1-64)

ms_time:输出电平翻转时间, 单位毫秒 (1-2000)

输出: 无

返回值: 错误码

4.4.7 运动函数错误码说明

错误码	名称	含义
0	ERR_NOERR	成功
其它	ERR_UNKNOWN	未知错误

4.5 例子程序

为了方便用户利用VC或VB等编程工具对IOC-0640/1280进行开发，针对典型的操作，例如：输入/输出口状态的读出，输出口的位操等，雷赛公司提供了示例源代码，用户可以直接将软件CD中相应目录中的代码直接拷贝到您的程序工程中使用。下面就例程源代码所能实现的功能作简要概述：

通用专用输入输出例程(VB)

此例程为 IOC-0640/1280 控制卡通用数字 I/O 信号的检测例程。程序里有所有通用输入/输出信号的检测图标，可以对每个输出口的输出状态进行操作，并可以检测每个输入输出信号的状态。详细代码请参考光盘例程“例 1_通用专用输入输出”。

通用输入输出例程主界面如下图 4-1：



图 4-1 通用输入输出例程主界面

通用专用输入输出例程(VC):

此例程为 IOC-0640/1280 控制卡通用数字 I/O 信号的检测例程。程序里有前 4 个通用输出信号的复选框，点击相应的复选择框可以有相应的输出口信号输出，有前 4 个输入信号的标签框的显示，可以测试前 4 个输入信号的检测。详细代码请参考光盘例程“例 1_通用输入输出”。

通用输入输出例程主界面如下图 4-2：

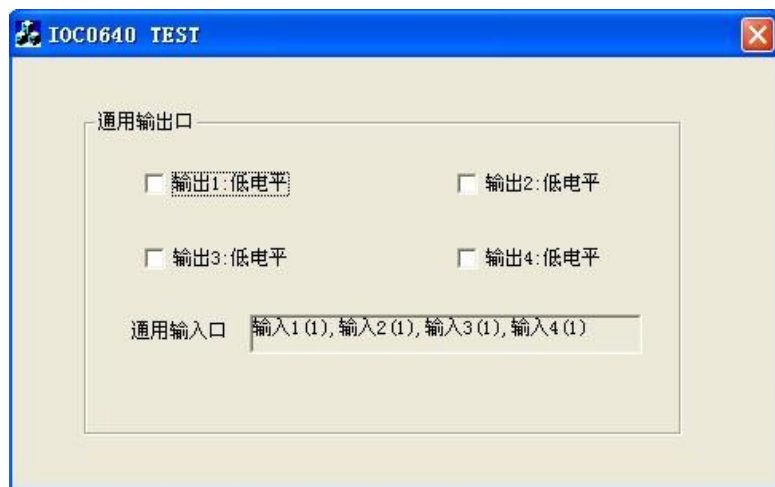


图 4-2 通用输入输出例程主界面

5 演示软件及应用

Motion 0640/1280 是雷赛公司为了便于用户熟悉 IOC-0640/1280 卡 IO 控制功能和 DLL 相关函数开发的一个演示测试软件。利用这个软件，用户可以很快地熟悉 IOC0640/1280 卡的软硬件功能。

通过测试软件界面，能够很直观的了解各通用 I/O 的实时状态，还可以通过按钮设置每一个通用输出口的电平，如图 4-7 所示

IOC-0640/1280 卡各输入口的中断，在本演示测试软件中，各个输入口的中断功能默认是开启的，还可以通过“刷新中断”获取各个输入口的中断状态。如图 5-1 所示。

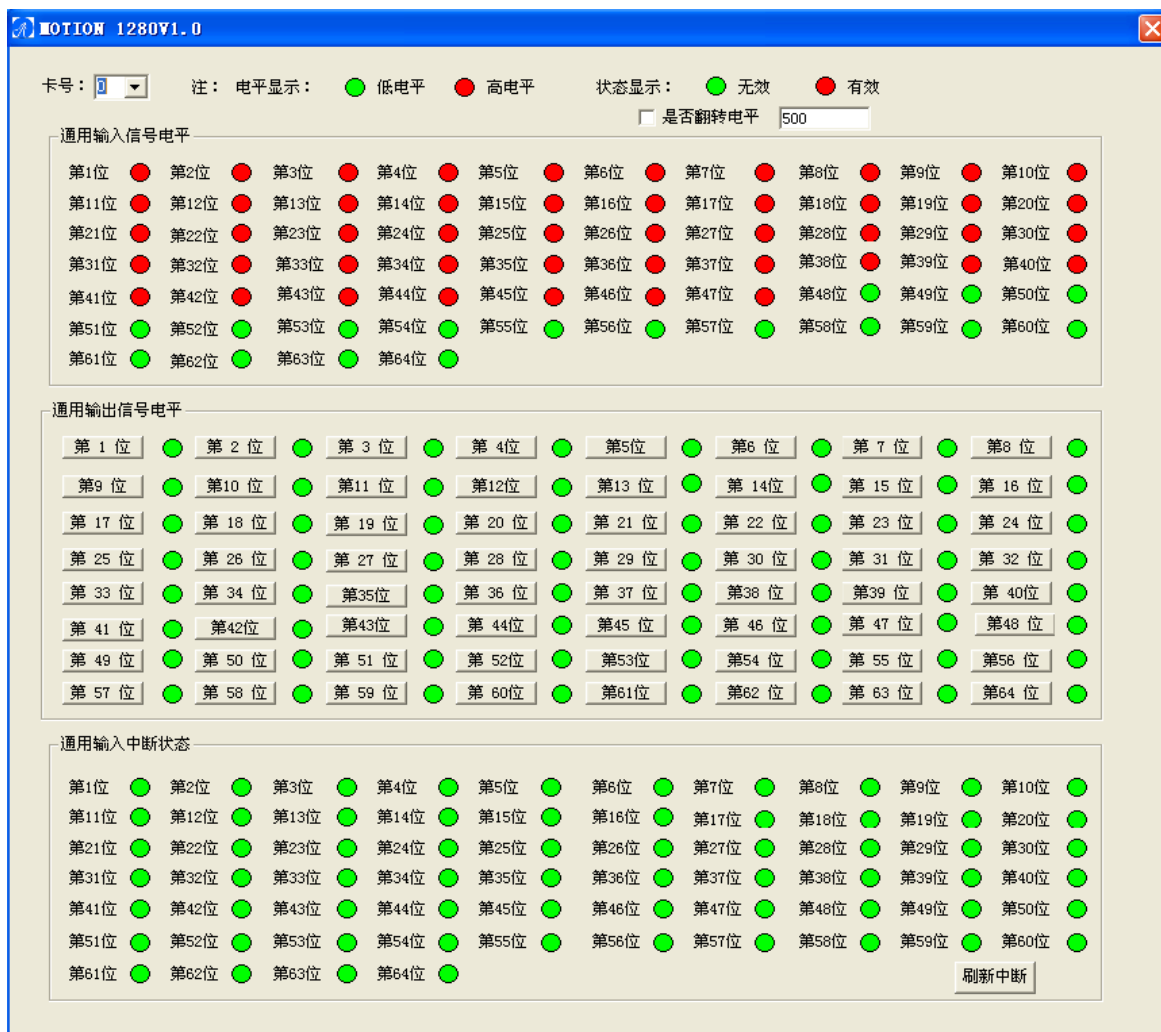


图 5-1 演示软件界面

6 用户系统开发

[概述]:

如果您对C、C++、Visual Basic等程序语言一点都不了解的话，我们建议您先花几天时间去阅读至少一本该语言的培训教材，并且通过练习掌握该语言的基本技巧，例如如何编写简单的程序，如何创建窗体和调用函数。如果您曾用C、C++、Visual Basic等程序语言进行过运动控制软件的开发，并具有丰富的经验，那么可以浏览索引中的函数，并找到所需要的函数描述页码，跳转到“第八章 运动函数说明” 查阅所需要的相关函数信息。

6.1 基于 windows 平台的应用软件结构

使用雷赛控制卡的机器控制系统构架如图6-1所示:

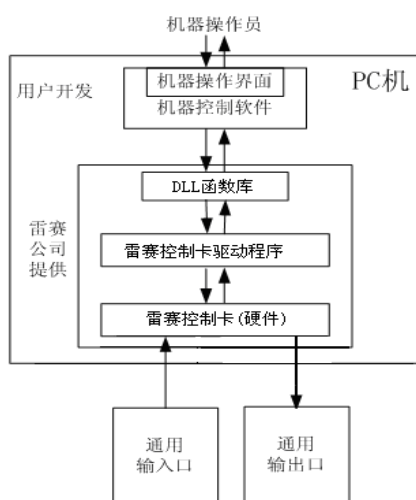


图6-1 基于雷赛IOC-0640/1280控制卡的机器控制系统构架

从上面的示意图可以看出，控制系统的工作原理可以简单描述为：

1. 操作员的操作信息通过操作接口（包括显示屏和键盘）传递给机器控制软件；
2. 机器控制软件将操作信息转化为控制参数并根据这些参数调用DLL库中的I/O操作函数；
3. 控制函数调用雷赛I/O控制卡驱动程序发出控制指令给控制卡；
4. 雷赛I/O控制卡再根据控制指令发出相应的信号读写通用输入输出接口。

用户在开发应用软件（即机器控制软件）的过程中所需要做的就是针对上面所说的第1步和第2步进行编程。雷赛公司已提供支持IOC0640/1280控制卡的硬件驱动程序和DLL运动函数库，包括控制卡初始化，输入/输出接口的读写等函数。这些函数提供了所有控制卡的相关功能，使用极为方便。用户不需要更多了解硬件电路的细节，就能够使用C、C++、Visual Basic等程序语言调用这些函数来快速开发出自己的应用软件。

用户编写的机器控制软件的典型流程如图6-2所示：

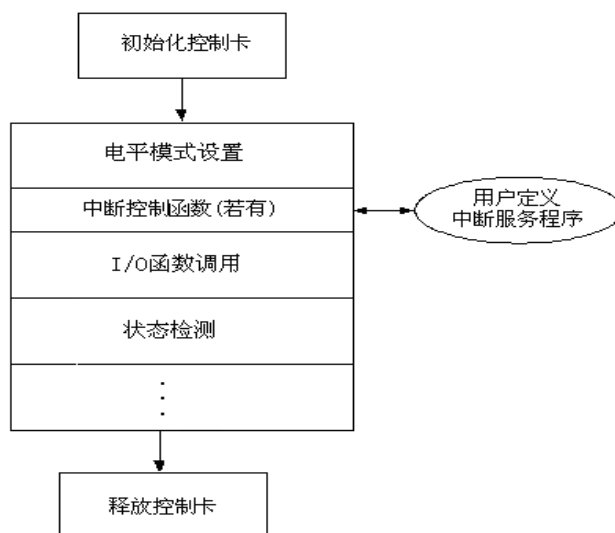


图6-2 控制软件的典型流程

6.2 Visual Basic 环境下编程

Visual Basic6.0 环境下编程，以控制 IOC-0640/1280 第一个输出口为例：

- (1) 在磁盘上新建一个目录，如 E:\test1
- (2) 打开 Visual Basic 6.0
- (3) 新建一个“标准 EXE”工程，在对话框上添加按钮“开启”（名称修改为 CB_Start）和“关闭”（名称修改为 CB_Stop），如图 6-3 所示：

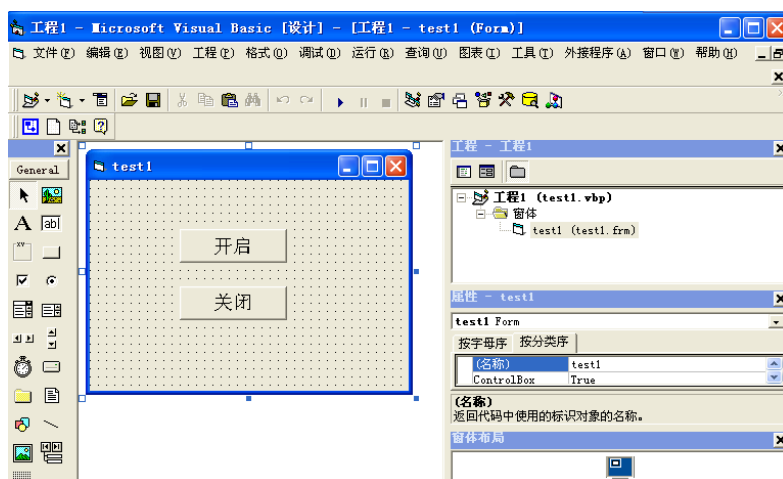


图6-3 添加按键

- (4) 工程保存在 E:\test1 目录下。
- (5) 在资料光盘相应目录下找到 IOC0640.bas 文件，拷贝到 test1 目录下。
- (6) 选择“工程”->“添加模块”->“现存”，找到 test1 目录下的 IOC0640.bas 文件，添加到工程中，如图 6-4 所示：

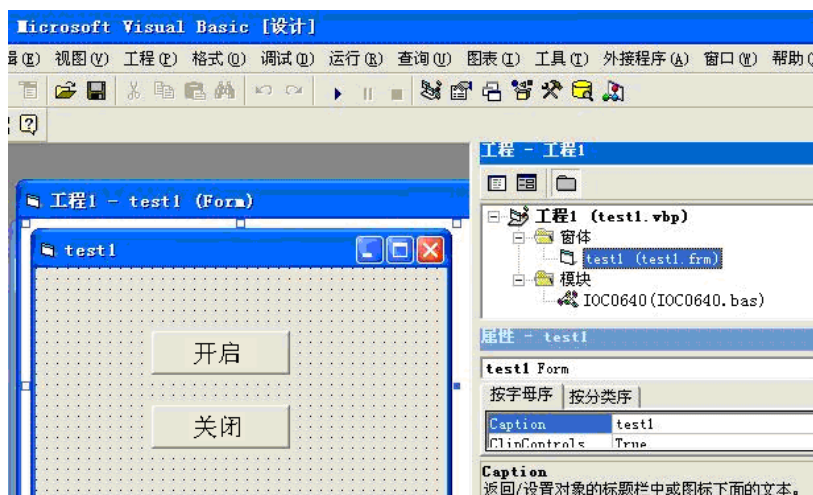


图6-4 添加模块IOC-0640/1280卡的IOC0640.bas

(7) 双击窗口控件，在 Form_Load 事件中添加代码：

```
ioc_board_init
```

选择 UnLoad 事件，在 Form_Unload 事件中添加代码：

```
ioc_board_close
```

双击“开启”按钮，在 CB_Start_Click 事件中添加代码：

```
ioc_write_outbit(0,1,0)
```

双击“关闭”按钮，在 CB_Stop_Click()事件中添加代码：

```
ioc_write_outbit(0,1,1)
```

代码如图 6-5 所示：

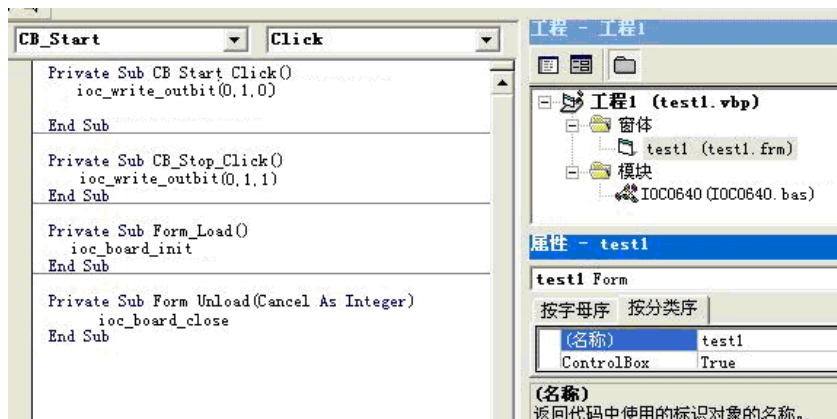


图6-5 程序中调用运动控制卡库函数

(8) 运行编好的程序，按下“开启”按钮，IOC-0640/1280 的第一个输出口将输出一个 OUPUT 信号；之后可以按下“关闭”按钮关闭 IOC-0640/1280 的第一个输出口信号的输出。程序界面如图 6-6 所示：

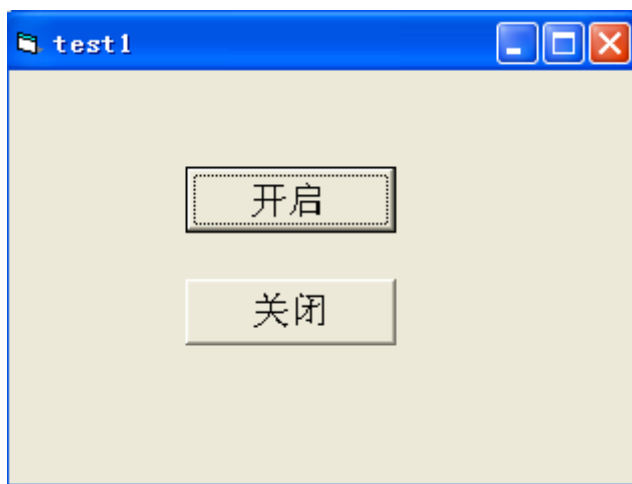


图6-6 程序运行界面

注：我们在 IOC-0640/1280 的用户光盘中还提供了以下示例供用户参考：

示例一，通用输入输出例程；

6.3 Visual C++环境下编程

Visual C++6.0 环境下编程，以控制通用输出口为例：

- 1) 打开 Visual C++ 6.0;
- (2) 新建一个工程;
- (3) 选择 MFC APPWizard(exe);
- (4) 选择工程保存路径，如：E:\;
- (5) 输入工程名，如：test1。如图 6-7 所示：

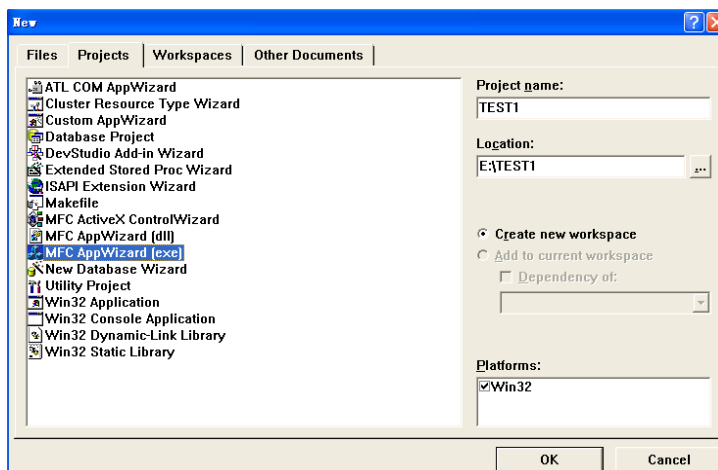


图6-7 创建新工程

- (5) 在应用程序类型中选择“基于对话框”，按“完成”键，建立工程。
- (6) 给对话框进行简单的修改，增加按钮“开启”（命名为 IDC_BUTTON_Start）和“关闭”（命名为 IDC_BUTTON_Stop），如图 6-8 所示。

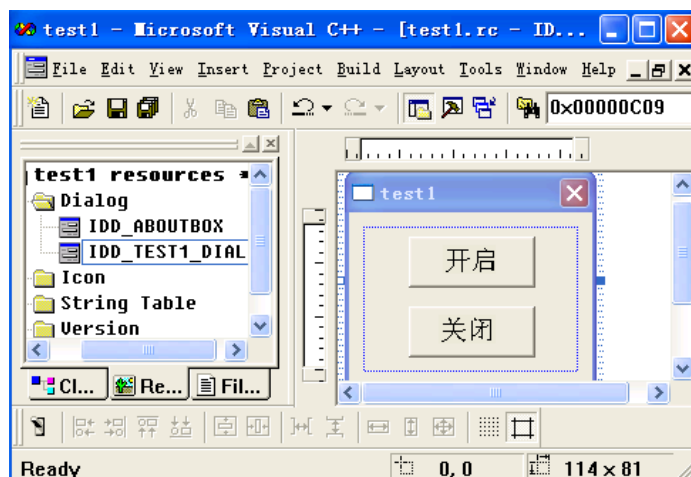


图6-8 修改对话框

- (7) 在资料光盘的相应目录下找到 IOC0640.h 和 IOC0640.lib 文件，拷贝到 E:\tes1 目录下。
- (8) 选择“工程”->“添加工程”->“文件”，选中 IOC0640.lib 文件加入到工程中。
- (9) 打开 test1.cpp 文件，在程序开始部分添加语句：`#include "IOC0640.h"`，如图 6-9 所示：

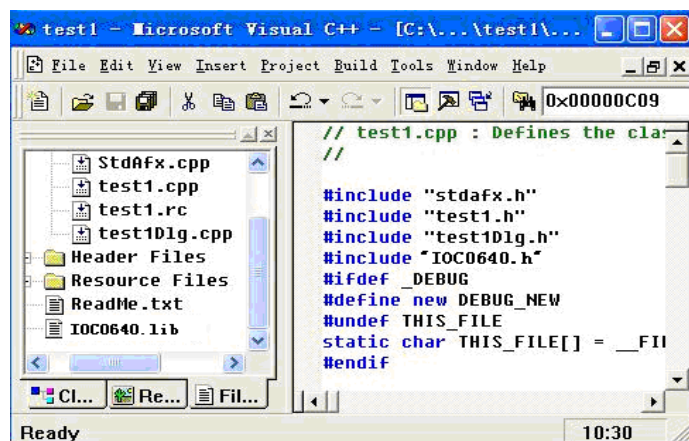


图6-9 程序增加头文件

- (10) 在 `CTest1Dlg::OnInitDialog()` 函数中添加代码：

```
ioc_board_init();
```

如图 6-10 所示：

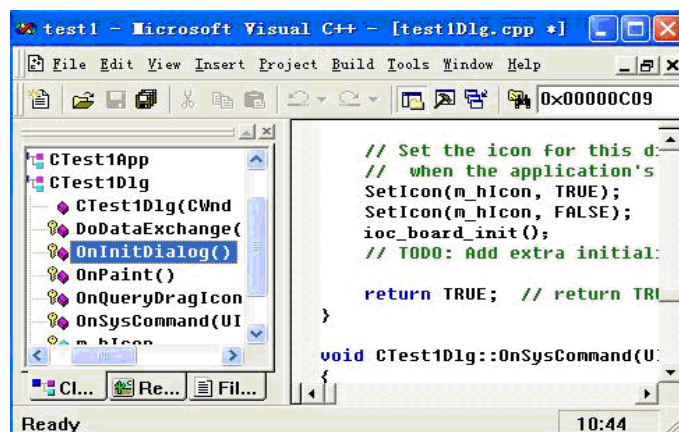


图6-10 程序增加初始化函数

(11) 在 Ctest1Dlg 中添加一个成员函数 OnCancel, 在 OnCancel 函数中添加代码:

```
ioc_board_close();
CDialog::OnCancel();
```

如图 6-11 所示:

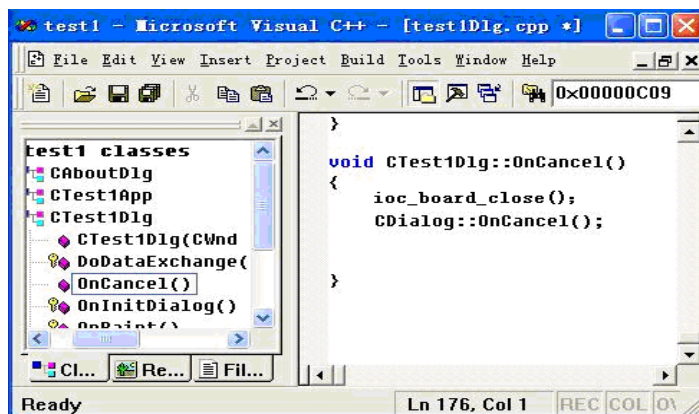


图6-11 程序增加OnCancel函数

(12) 双击“开启”按钮在按钮点击事件中输入代码:

```
ioc_write_outbit(0,1,0);
```

双击“关闭”按钮在按钮点击事件中输入代码:

```
ioc_write_outbit(0,1,1);
```

如图 6-12 所示:

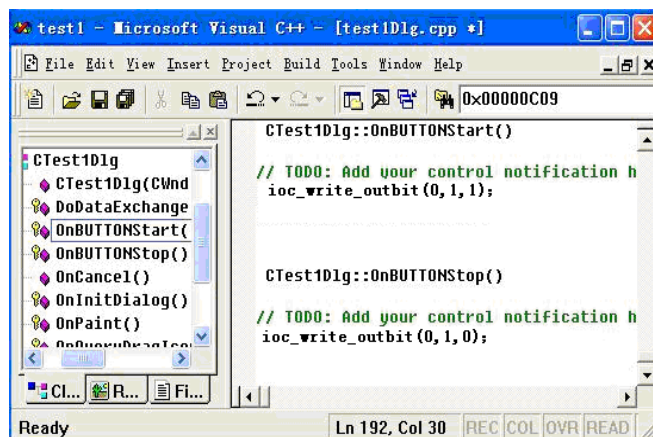


图6-12 程序中调用运动控制卡库函数

(13) 编译, 运行, 按下“开启”按钮, IOC-0640/1280 的第一个输出口将输出一个 OUPUT 信号; 之后可以按下“关闭”按钮关闭 IOC-0640/1280 第一个输出口信号的输出。界面如图 6-13 所示:

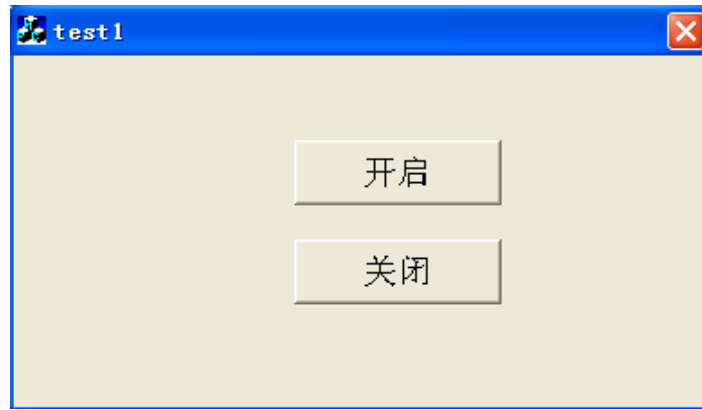


图6-13 程序运行界面

注：我们在 IOC-0640/1280 的用户光盘中还提供了以下几个示例供用户参考：

示例一，通用输入输出例程；

7 附录

7.1 IOC0640/1280 主板硬件信号接线板端口定义表

X2 是 I/O 信号的接口，为 SCSI-II 型 68 针插座。

针脚号和针脚名对应如下：

引脚号	名称	类型	说明	引脚号	名称	类型	说明
1	EXGND	I	外部电源EGND	35	E24V	I	外部 24V 电源
2	OUT1	0	隔离输出口 1	36	OUT2	0	隔离输出口 2
3	OUT3	0	隔离输出口 3	37	OUT4	0	隔离输出口 4
4	OUT9	0	隔离输出口 9	38	OUT10	0	隔离输出口 10
5	OUT11	0	隔离输出口 11	39	OUT12	0	隔离输出口 12
6	OUT17	0	隔离输出口 17	40	OUT18	0	隔离输出口 18
7	OUT19	0	隔离输出口 19	41	OUT20	0	隔离输出口 20
8	OUT25	0	隔离输出口 25	42	OUT26	0	隔离输出口 26
9	OUT27	0	隔离输出口 27	43	OUT28	0	隔离输出口 28
10	OUT5	0	隔离输出口 5	44	OUT13	0	隔离输出口 13
11	OUT6	0	隔离输出口 6	45	OUT14	0	隔离输出口 14
12	OUT7	0	隔离输出口 7	46	OUT15	0	隔离输出口 15
13	OUT21	0	隔离输出口 21	47	OUT29	0	隔离输出口 29
14	OUT22	0	隔离输出口 22	48	OUT30	0	隔离输出口 30
15	OUT23	0	隔离输出口 23	49	OUT31	0	隔离输出口 31
16	OUT8	0	隔离输出口 8	50	OUT16	0	隔离输出口 16
17	OUT24	0	隔离输出口 24	51	OUT32	0	隔离输出口 32
18	IN1	I	隔离输入口 1	52	IN2	I	隔离输入口 2
19	IN3	I	隔离输入口 3	53	IN4	I	隔离输入口 4
20	IN5	I	隔离输入口 5	54	IN6	I	隔离输入口 6
21	IN7	I	隔离输入口 7	55	IN8	I	隔离输入口 8
22	IN9	I	隔离输入口 9	56	IN10	I	隔离输入口 10
23	IN11	I	隔离输入口 11	57	IN12	I	隔离输入口 12
24	IN13	I	隔离输入口 13	58	IN14	I	隔离输入口 14
25	IN15	I	隔离输入口 15	59	IN16	I	隔离输入口 16
26	IN17	I	隔离输入口 17	60	IN18	I	隔离输入口 18
27	IN19	I	隔离输入口 19	61	IN20	I	隔离输入口 20
28	IN21	I	隔离输入口 21	62	IN22	I	隔离输入口 22
29	IN23	I	隔离输入口 23	63	IN24	I	隔离输入口 24
30	IN25	I	隔离输入口 25	64	IN26	I	隔离输入口 26
31	IN27	I	隔离输入口 27	65	IN28	I	隔离输入口 28
32	IN29	I	隔离输入口 29	66	IN30	I	隔离输入口 30
33	IN31	I	隔离输入口 31	67	IN32	I	隔离输入口 32
34	EXGND	I	外部电源 EGND	68	E24V	I	外部 24V 电源

7.2 IOC640/1280 主板硬件信号接线盒端口定义表

接线盒电源接口

VDD	I	外部电源, +12V~+24V
EXGND		外部电源地
EXGND		外部电源地

接口名称	引脚	名称	类	说明
P1	16	IN8	I	隔离输入口 8
	15	IN7	I	隔离输入口 7
	14	IN6	I	隔离输入口 6
	13	IN5	I	隔离输入口 5
	12	IN4	I	隔离输入口 4
	11	IN3	I	隔离输入口 3
	10	IN2	I	隔离输入口 2
	9	IN1	I	隔离输入口 1
	8	OUT8	O	隔离输出口 8
	7	OUT7	O	隔离输出口 7
	6	OUT6	O	隔离输出口 6
	5	OUT5	O	隔离输出口 5
	4	OUT4	O	隔离输出口 4
	3	OUT3	O	隔离输出口 3
	2	OUT2	O	隔离输出口 2
	1	OUT1	O	隔离输出口 1

接口名称	引脚	名称	类	说明
P2	16	IN16	I	隔离输入口 16
	15	IN15	I	隔离输入口 15
	14	IN14	I	隔离输入口 14
	13	IN13	I	隔离输入口 13
	12	IN12	I	隔离输入口 12
	11	IN11	I	隔离输入口 11
	10	IN10	I	隔离输入口 10
	9	IN9	I	隔离输入口 9
	8	OUT16	O	隔离输出口 16
	7	OUT15	O	隔离输出口 15
	6	OUT14	O	隔离输出口 14
	5	OUT13	O	隔离输出口 13
	4	OUT12	O	隔离输出口 12
	3	OUT11	O	隔离输出口 11
	2	OUT10	O	隔离输出口 10
	1	OUT9	O	隔离输出口 9

接口名称	引脚	名称	类	说明
P3	16	IN24	I	隔离输入口 24
	15	IN23	I	隔离输入口 23
	14	IN22	I	隔离输入口 22
	13	IN21	I	隔离输入口 21
	12	IN20	I	隔离输入口 20
	11	IN19	I	隔离输入口 19
	10	IN18	I	隔离输入口 18
	9	IN17	I	隔离输入口 17
	8	OUT24	O	隔离输出口 24
	7	OUT23	O	隔离输出口 23
	6	OUT22	O	隔离输出口 22
	5	OUT21	O	隔离输出口 21
	4	OUT20	O	隔离输出口 20
	3	OUT19	O	隔离输出口 19
	2	OUT18	O	隔离输出口 18
	1	OUT17	O	隔离输出口 17

接口名称	引脚	名称	类	说明
P4	16	IN8	I	隔离输入口 32
	15	IN7	I	隔离输入口 31
	14	IN6	I	隔离输入口 30
	13	IN5	I	隔离输入口 29
	12	IN4	I	隔离输入口 28
	11	IN3	I	隔离输入口 27
	10	IN2	I	隔离输入口 26
	9	IN1	I	隔离输入口 25
	8	OUT32	O	隔离输出口 32
	7	OUT31	O	隔离输出口 31
	6	OUT30	O	隔离输出口 30
	5	OUT29	O	隔离输出口 29
	4	OUT28	O	隔离输出口 28
	3	OUT27	O	隔离输出口 27
	2	OUT26	O	隔离输出口 26
	1	OUT25	O	隔离输出口 25

7.3 IOC0640 扩展板采用 IOB-3721 接口定义表

接口名称	名称	类	说明
P1	EGND		外部电源地
	EGND		外部电源地
	EGND		外部电源地
	E24V	I	外部电源 24V+
	E24V	I	外部电源 24V+
	EGND		外部电源地
	EGND		外部电源地
	EGND		外部电源地

接口名称	引脚	名称	类	说明
P2	16			
	15	IN47	I	隔离输入口 47
	14	IN46	I	隔离输入口 46
	13	IN45	I	隔离输入口 45
	12	IN44	I	隔离输入口 44
	11	IN43	I	隔离输入口 43
	10	IN42	I	隔离输入口 42
	9	IN41	I	隔离输入口 41
	8	IN40	I	隔离输入口 40
	7	IN39	I	隔离输入口 39
	6	IN38	I	隔离输入口 38
	5	IN37	I	隔离输入口 37
	4	IN36	I	隔离输入口 36
	3	IN35	I	隔离输入口 35
	2	IN34	I	隔离输入口 34
1	IN33	I	隔离输入口 33	

接口名称	引脚	名称	类	说明
P3	16	OUT48	O	隔离输出口 48
	15	OUT47	O	隔离输出口 47
	14	OUT46	O	隔离输出口 46
	13	OUT45	O	隔离输出口 45
	12	OUT44	O	隔离输出口 44
	11	OUT43	O	隔离输出口 43
	10	OUT42	O	隔离输出口 42
	9	OUT41	O	隔离输出口 41
	8	OUT40	O	隔离输出口 40
	7	OUT39	O	隔离输出口 39
	6	OUT38	O	隔离输出口 38
	5	OUT37	O	隔离输出口 37
	4	OUT36	O	隔离输出口 36
	3	OUT35	O	隔离输出口 35
	2	OUT34	O	隔离输出口 34
1	OUT33	O	隔离输出口 33	

7.4 IOC1280 扩展板采用 IOB-6421 接口定义表

IOC1280 扩展板 IOB-6421 的接口定义与主板相同。

7.5 版次说明

本次卡的版本号为：IOC-0640/1280 v1.3