

PC-6333 多功能模入模出接口卡技术说明书

1. 概述:

PC-6333 多功能模入模出接口卡适用于具有 ISA 总线的 PC 系列微机, 具有很好的兼容性, CPU 从目前广泛使用的 64 位处理器直到早期的 16 位处理器均可适用, 操作系统可选用经典的 MS-DOS, 目前流行的 Windows 系列, 高稳定性的 Unix 等多种操作系统以及专业数据采集分析系统 LabVIEW 等软件环境。在硬件的安装上也非常简单, 使用时只需将接口卡插入机内任何一个 ISA 总线插槽中, 信号电缆从机箱外部直接接入。也可插入我所研制的 PC 扩展箱内使用。本卡可广泛应用于工业过程控制系统以及实验室数据采集系统。

PC-6333 多功能模入模出接口卡安装使用方便, 程序编制简单。其模入模出及 I/O 信号均由卡上的 37 芯 D 型插头与外部信号源及设备连接。对于模入部分, 用户可根据实际需要选择单端或双端输入方式。对于模出部分, 用户可根据控制对象的需要选择电压或电流输出方式以及不同的量程。本卡上的 A/D、D/A 转换均为 12 位, 同时还备有 6 路数字量输入和 6 路数字量输出接口, 三路 16 位字长的计数/定时器, 以及 1MHz 的基准时钟。本卡的 A/D 转换启动方式可以选用程序触发、定时器自动触发、外同步触发等方式, 转换状态可以用程序查询, 也可以用中断方式通知 CPU 读取转换结果。

2. 主要技术参数:

2.1 模入部分:

2.1.1 输入通道数: 单端 16 路; * (标*为出厂标准状态, 下同)
双端 8 路

2.1.2 输入信号范围: $0V \sim 10V^*$; $-5V \sim +5V$

2.1.3 输入阻抗: $\geq 10M\Omega$

2.1.4 A/D 转换分辨率: 12 位

2.1.5 A/D 转换速度: $10\mu S$

2.1.6 A/D 启动方式: 程序启动 / 定时触发启动 / 外触发启动

2.1.7 A/D 转换结束识别: 程序查询 / 中断方式

2.1.8 A/D 转换非线性误差: $\pm 1LSB$

2.1.9 A/D 转换输出码制: 单极性原码* / 双极性偏移码

2.1.10 系统综合误差: $\leq 0.1\% F.S$

2.2 模出部分:

2.2.1 输出通道数: 1 路

2.2.2 输出范围:

电压方式: $0 \sim 5V$; $0 \sim 10V^*$; $-5V \sim +5V$; $-2.5V \sim +2.5V$

电流方式: $0 \sim 10mA$; $4 \sim 20mA$

2.2.3 输出阻抗: $\leq 2\Omega$ (电压方式)

2.2.4 D/A 转换器件: DAC1210

2.2.5 D/A 转换分辨率: 12 位

2.2.6 D/A 转换输入码制: 二进制原码 (单极性输出方式时) *
二进制偏移码 (双极性电压输出方式时)

2.2.7 D/A 转换综合建立时间: $\leq 2\mu S$

2.2.8 D/A 转换综合误差: 电压方式: $\leq 0.1\% F.S$

电流方式: $\leq 0.5\% F.S$

2.2.9 电压输出方式负载电流: $\leq 5mA$

2.2.10 电流输出方式负载电阻范围:

使用机内 +12V 电源时: $0 \sim 250\Omega$

外加 +24V 电源时: $0 \sim 750\Omega$

2.3 数字量输入输出部分:

2.3.1 DI: 6 路 / DO: 6 路; TTL 电平

2.3.2 16 位字长计数 / 定时器: 3 路, 用户可外接使用一路计数 / 定时通道

2.3.3 基准时钟: 1MHz, 占空比 50%

2.4 电源功耗: $+5V(\pm 10\%) \leq 500mA$

$+12V(\pm 10\%) \leq 100mA$ (D/A $4 \sim 20mA$ 输出, 并使用机内电源时)

$-5V(\pm 10\%) \leq 10mA$

$-12V(\pm 10\%) \leq 50mA$

2.5 使用环境要求: 工作温度: $10^\circ C \sim 40^\circ C$

相对湿度: 40%~80%

存储温度：-55℃~+85℃

2.6 外型尺寸：（不含档板）

长×高=185.5mm×106.7 mm （7.3英寸×4.2英寸）

3. 工作原理：

PC-6333 模入模出接口卡主要由模数转换电路、数模转换电路、数字量输入输出电路，接口控制逻辑电路构成。

3.1 工作原理框图：

PC-6333 模入模出接口卡工作原理框图见图 1。

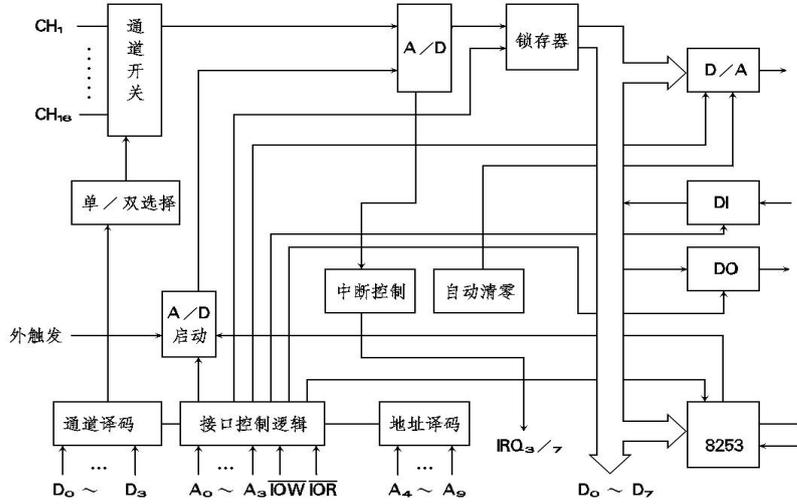


图 1 工作原理框图

3.2 模入部分：外部模拟信号经多路转换开关选择后送入高速放大器处理。放大器前后设有单 / 双端输入选择跨接器 KJ1、KJ2 和转换码制选择跨接器 KJ3，处理后的信号送入模数转换器进行转换。模数转换器的启动可以使用程序启动方式或者定时器定时触发启动方式，也可用外部触发方式启动。其转换状态和结果可用程序查询和读出。转换结束信号也可用中断方式通知 CPU 进行处理。

3.3 模出部分：

模拟量输出部分由 DAC1210 D/A 转换器件和有关的基准源、运放、阻容件和跨接选择器组成。依靠改变跨接套的连接方式，可分别选择电压或电流输出方式以及不同的输出量程。当采用电流输出方式时，本卡可直接外接 II、III 型执行器。D/A 部分具有加电自动清零功能。

3.4 数字量输入输出部分：

数字量输入输出电路为用户提供 6 路 DI 及 6 路 DO 的信号，并具备加电 DO 清零功能。

3.5 计数 / 定时器部分：

计数 / 定时器电路由一片可编程定时 / 计数器 8253 芯片和基准时钟电路以及有关的跨接选择器组成。可为用户提供 3 个 16 位字长的计数 / 定时通道和 1MHz、占空比为 50% 的基准时钟，其中两路计数 / 定时通道供本卡内部使用，用户可外接使用一路计数 / 定时通道。

3.6 8253 可编程计数 / 定时器应用简介：

3.6.1 8253 芯片管脚图如图 2（图 2 见下页）。

3.6.2 8253 功能及框图：

8253 是 INTEL 公司微型计算机系统中的一个部件，可以将 8253 作为一个具有四个输入 / 输出接口的器件处理，其中三个是计数器，一个是可程序工作方式的控制寄存器。其内部结构图如图 3 所示



图 2 8253 芯片管脚图

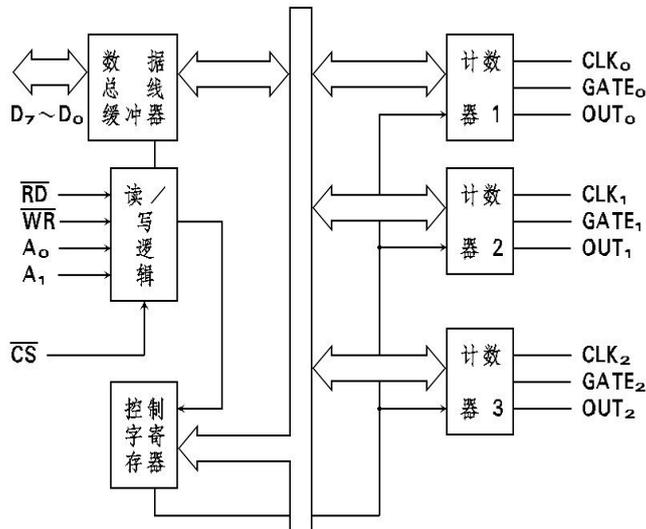


图 3 8253 内部结构图

3.6.3 8253 可编程计数 / 定时器编程要点:

8253 的全部功能是由 CPU 编程设定的。CPU 通过输出指令给 8253 装入控制字，从而设定其功能。8253 控制字格式如下:

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

各位的功能见表 1~表 4:

表 1 SC₁、SC₀ — 计数器选择

SC1	SC0	选择计数器
0	0	选择 0#
0	1	选择 1#
1	0	选择 2#
1	1	非法

表 2 RL₁、RL₀—CPU 读 / 写操作

RL1	RL0	操作类型
0	0	计数器封锁操作
0	1	读 / 写计数器低 8 位
1	0	读 / 写计数器高 8 位
1	1	先读 / 写低 8 位, 后读 / 写高 8 位

表 3 M₂、M₁、M₀ — 工作方式选择

M3	M2	M1	计数工作方式
0	0	0	方式 0
0	0	1	方式 1
0	1	0	方式 2
0	1	1	方式 3
1	0	0	方式 4
1	0	1	方式 5

表 4 BCD—计数方式选择

BCD	数码形式
0	十六位二进制计数
1	四位十进制 (BCD) 码计数

8253-5 的三个计数器是独立的 16 位减法计数器。计数器的工作方式由工作方式寄存器确定。计数器在编程写入初始值后, 在某些方式下计数到 0 后自动预置, 计数器连续工作。CPU 访问计数器时, 必须先设定工作方式控制字中的 RL1、RL0 位。计数器对 CLK 计数输入端的输入信号进行递减计数。选通信号 GATE 控制计数工作的进行, 其功能如表 5 所示。

表 5 选通信号 GATE 的功能

	低电平或进入低电平	上升边沿	高电平
方式 0	禁止计数	-----	允许计数
方式 1	-----	1. 初始化和计数 2. 下一个时钟后清除输出	-----
方式 2	1. 禁止计数 2. 使输出立即变为高电平	1. 重新装入计数器 2. 启动计数	允许计数
方式 3	1. 禁止计数 2. 使输出立即变为高电平	初始化和计数	允许计数
方式 4	禁止计数	计数未结束时初始化和计数	允许计数
方式 5	-----	初始化和计数	-----

8253-5 的三个计数器按照各工作方式寄存器中控制字的设置进行工作。可以选择的工作方式有六种。这六种方式是：

方式 0：计数结束时中断。编程后自动启动，计数器减 1 计数，计数到终点（减至 0）后输出高电平，可用于中断请求信号，GATE 为低电平时停止计数，回到高电平后继续往下计数。再次启动要重新装入计数值或重新编程。

方式 1：可编程单脉冲输出。GATE 上升沿进行初始化并开始计数。输出低电平的宽度等于计数时间。单脉冲输出可用 GATE 上升沿多次触发。

方式 2：比率发生器。编程后重复地循环计数。计数到终点时输出一个时钟周期宽度的低电平脉冲，自动初始化后继续计数。用 GATE 的上升沿初始化，并开始计数。GATE 为低电平时停止计数。

方式 3：方波发生器。这种方式是在编程后重复地循环计数，输出波形为方波。如果初始计数值为偶数，每个时钟输入脉冲使计数器减 2，达到计数终点时输出电平改变。如果初始计数值为奇数，则输出高电平时第一个时钟输入脉冲使计数器减 1，随后每个输入脉冲使计数器减 2；输出为低电平时第一个时钟输入脉冲使计数器减 3，随后每个输入脉冲使计数器减 2，到达计数终点时输出电平改变，计数器自动初始化后继续计数。用 GATE 的上升沿初始化并开始计数，GATE 为低电平时停止计数。

方式 4：软件启动选通脉冲输出。编程后自动启动，计数到终点后输出一个时钟周期的低电平脉冲。用 GATE 的上升沿初始化并开始计数，GATE 为低电平时停止计数。

方式 5：硬件启动选通脉冲输出。编程后，等待 GATE 上升沿进行初始化并开始计数，计数到终点后输出一个时钟周期的低电平脉冲，计数器开始计数后不受 GATE 信号电平的影响，这种选通脉冲的输出可用 GATE 的上升沿多次触发。在工作方式控制字中，如果设置计数器锁存操作，则该控制字中工作方式选择位 M1、M0 和计数方式选择位 BCD 无效。即设置锁存操作时不影响计数器的工作方式，计数器锁存操作，是在计数器计数过程中，在不影响正在进行的计数操作的条件下，把当前的计数值锁存到寄存器，供 CPU 读取，这时在工作方式控制字中，SC1、SC0 指定要锁存的计数器，RL1、RL0=00 表示锁存操作，其余 4 位无效，计数器按原来设定的方式工作。当本卡 A/D 转换选择定时器定时触发启动工作方式时，一般将 8253 的工作方式设置为方式 2（即比率发生器），以保证符合 A/D 转换启动信号的要求。

4. 安装及使用注意：

4.1 安装：

本卡的安装十分简便，只要将主机机壳打开，在关电情况下，将本卡插入主机的任何一个空余扩展槽中，再将档板固定螺丝压紧即可。37 芯 D 型插头可从主机后面引出并与外设连接。

4.2 本卡采用的模拟开关是 COMS 电路，容易因静电击穿或过流造成损坏，所以在安装或用手触摸本卡时，应事先将人体所带静电荷对地放掉，同时应避免直接用手接触器件管脚，以免损坏器件。 4.3

禁止带电插拔本接口卡。设置接口卡开关、跨接套和安装接口带缆均应在关电状态下进行。

4.3 当模入通道不全部使用时，应将不使用的通道就近对地短接，不要使其悬空，以避免造成通道间串扰和损坏通道。

4.4 本卡跨接选择器较多，使用中应严格按照说明书进行设置操作。电压方式模拟输出时，应避免输出端对地短路。

4.5 为保证安全及采集精度，应确保系统地线（计算机及外接仪器机壳）接地良好。特别是使用双端输入方式时，为防止外界较大的共模干扰，应注意对信号线进行屏蔽处理。

4.6 对外供电端应注意加以保护，严禁短路，否则将造成主机电源损坏，使用中应特别小心。

5. 使用与操作：

5.1 主要可调整元件位置见图 4。

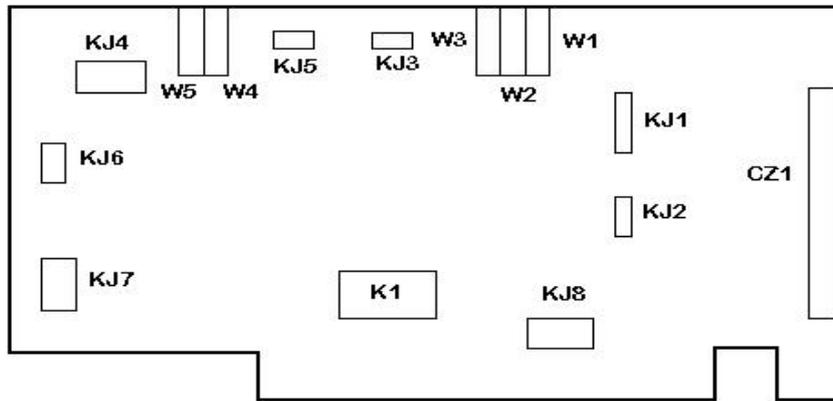


图 4 主要可调整元件位置图

5.2 输入输出插座接口定义:

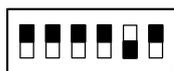
输入输出插座接口定义 (括号内表示双端输入方式时通道组成) 见表 6。

表 6 输入输出插座接口定义表

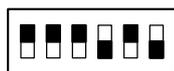
插座引脚号	信号定义	插座引脚号	信号定义
1	模拟地	20	模拟地
2	CH1 (CH1+)	21	CH9 (CH1-)
3	CH2 (CH2+)	22	CH10 (CH2-)
4	CH3 (CH3+)	23	CH11 (CH3-)
5	CH4 (CH4+)	24	CH12 (CH4-)
6	CH5 (CH5+)	25	CH13 (CH5-)
7	CH6 (CH6+)	26	CH14 (CH6-)
8	CH7 (CH7+)	27	CH15 (CH7-)
9	CH8 (CH8+)	28	CH16 (CH8-)
10	D / A	29	模拟地
11	E. T	30	+12V
12	+5V	31	数字地
13	D I0	32	D I1
14	D I2	33	D I3
15	D I4	34	D I5
16	D00	35	D01
17	D02	36	D03
18	D04	37	D05
19	-12v		

5.3 I / O 基地址选择:

I / O 基地址的选择是通过开关 K_1 进行的, 开关拨至“ON”处为 0, 反之为 1。初始地址的选择范围一般为 0100H~01FFH; 0210H~02FFH 以及 0300H~036FH 之间。用户应根据主机硬件手册给出的可用范围以及是否插入其它功能卡来决定本卡的 I / O 基地址。出厂时本卡的基地址设为 0300H, 并从基地址开始占用连续 12 个地址。现举例说明见图 5。



a. 0100H



b. 0280H



c. 0310H

图 5 I / O 基地址选择举例

5.4 跨接插座的用法:

5.4.1 输入单 / 双端方式选择:

KJ_1 、 KJ_2 为单 / 双端输入方式选择, 其使用方法见图 6。

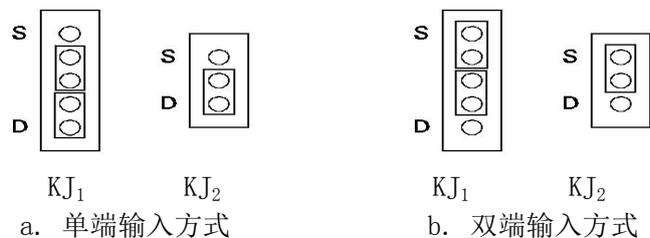


图 6 单 / 双端输入方式选择

5.4.2 转换码制选择:

KJ₃为转换码制选择插座。码制的定义参见 5.6 节。用户应根据输入信号的极性进行选择，选择方法见图 7。

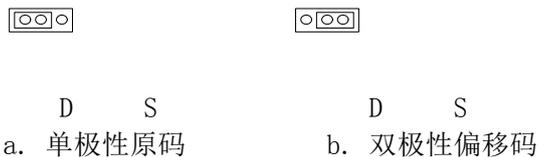


图 7 转换码制选择

5.4.3 D/A 输出量程选择:

KJ₄为 D/A 输出量程选择插座，使用时应配合 KJ₅ 输出方式选择插座共同使用。输出量程的选择方法见图 8。

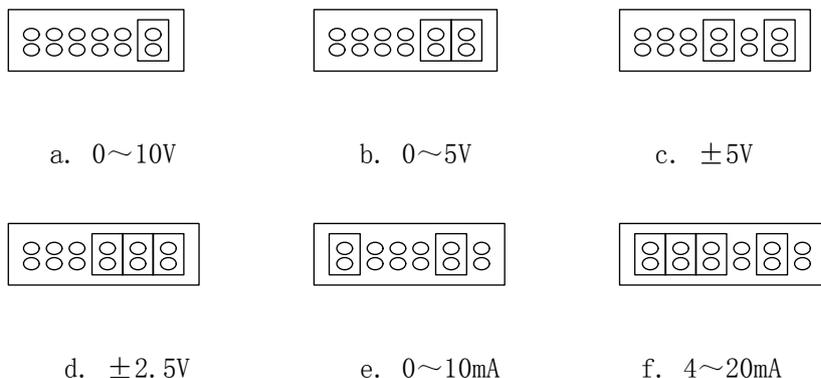


图 8 D/A 输出方式及范围选择

5.4.4 D/A 输出方式选择:

KJ₅为 D/A 输出方式选择插座，使用中应与输出量程选择插座配合使用，否则会造成错误的结果。KJ₅ 的使用方法见图 9。

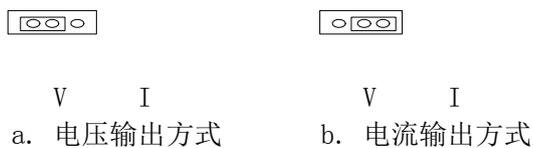


图 9 D/A 输出方式选择

5.4.5 定时触发启动选择:

KJ₆用于在定时触发启动方式时，选择 8253 的定时 / 计数器输出通道，本卡上的 8253 定时 / 计数器中，通道 0 和通道 1 已连成级连方式，即通道 0 的 CLK0 端接至 1MHz 时钟上，OUT0 端接至通道 1 的 CLK1 端，通道 0、1、2 上的 GATE 并联接程控信号。通道 2 留给用户选用，其定义及连接方法见图 10，用户可根据需要选择。

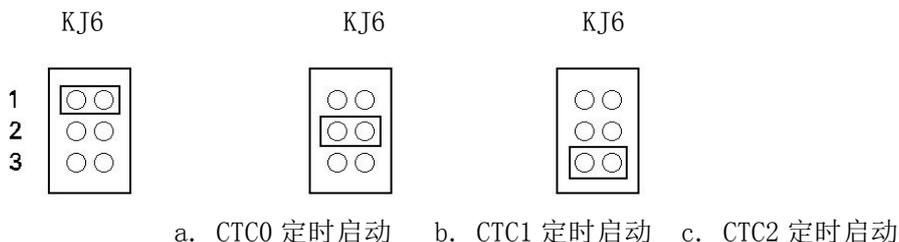


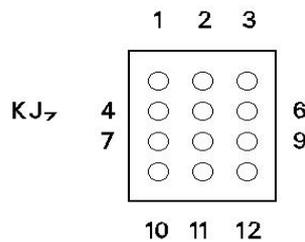
图 10 定时触发启动选择

5.4.6 触发信号及中断信号方式选择:

KJ7 用于选择 A / D 启动触发信号以及中断信号。使用中应该注意 A / D 启动触发信号和中断信号都必须是唯一的，不允许同时有多个触发和中断源。KJ7 的定义见表 7:

表 7 KJ7 的定义

功 能 定 义	使 用 选 择
P1 : 8253 通道 1 输出 P2 : 8253 通道 2 输入 P3: 外触发信号	8253 通道 2 与通道 1 级连: P1—P2 8253 通道 2 连接外部信号源: P2—P3
P4 : 外 触 发 中 断 P5 : 外 触 发 信 号 P6: 外触发启动 A / D	外触发信号产生中断: P4—P5 外触发信号启动 A / D: P5—P6
P7: 定时触发中断 P8: 定时触发信号 P9: 定时触发启动 A / D	使用 8253 定时产生中断: P7—P8 使用 8253 定时启动 A / D: P8—P9
P10: A / D 转换结束中断 P11: A / D 转换结束信号 P12: NC	A / D 转换结束后产生中断: P10—P11 A / D 转换结束后不产生中断: P11—P12



5.4.7 中断源选择:

KJ8 为中断选择插座。该插座可用来选择 IORQ3、IORQ4、IORQ5、IORQ6、IORQ7 中断源，中断源的选择见图 11。

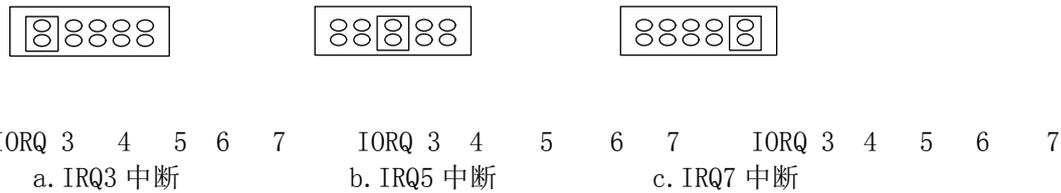


图 11 中断源的选择

5.5 控制端口地址与有关数据格式:

5.5.1 各个控制端的操作地址与功能见表 8。

5.5.2 通道代码数据格式见表 9。

5.5.3 8253 程控信号及中断申请允许信号的定义及数据格式:

8253 程控信号及中断申请允许信号的定义及数据格式见表 10。

表 8 端口地址与功能

端口操作地址	操 作 命 令	功 能
BASE+0	IOW	写通道代码，写 8253 程控及中断允许代码
BASE+0	IOR	启动 D / A 转换
BASE+1	IOW	启动 A / D 转换
BASE+2	IOR	查询 A / D 转换状态，读高 4 位转换结果
BASE+3	IOR	读 A / D 低 8 位转换结果，清除 A / D 中断标志

BASE+4	$\overline{\text{IOW}}$	写 D/A 高 8 位数据
BASE+5	$\overline{\text{IOW}}$	写 D/A 低 4 位数据
BASE+6	$\overline{\text{IOR}}$	读 D I 数据
BASE+7	$\overline{\text{IOW}}$	写 D0 数据
BASE+8	$\overline{\text{IOR}} / \overline{\text{IOW}}$	读 / 写 8253 计数器 0 通道数据
BASE+9	$\overline{\text{IOR}} / \overline{\text{IOW}}$	读 / 写 8253 计数器 1 通道数据
BASE+A	$\overline{\text{IOR}} / \overline{\text{IOW}}$	读 / 写 8253 计数器 2 通道数据
BASE+B	$\overline{\text{IOW}}$	写入 8253 控制寄存器控制字

表 10 8253 程控信号及中断申请允许信号的定义及数据格式

端口地址	操作命令	D7 D6 D5 D4	D3 D2 D1 D0	操作结果
BASE+0	$\overline{\text{IOW}}$	1 X X X	通道代码	8253 GATE 允许
BASE+0		0 X X X	通道代码	8253 GATE 禁止
BASE+0	$\overline{\text{IOW}}$	X 1 X X	通道代码	中断申请允许
BASE+0		X 0 X X	通道代码	中断申请禁止

表 9 通道代码数据格式

通道号	十进制代码	十六进制代码	输入方式	通道号	十进制代码	十六进制代码	输入方式
1	0	00H	单 / 双	9	8	08H	单
2	1	01H	单 / 双	10	9	09H	单
3	2	02H	单 / 双	11	10	0AH	单
4	3	03H	单 / 双	12	11	0BH	单
5	4	04H	单 / 双	13	12	0CH	单
6	5	05H	单 / 双	14	13	0DH	单
7	6	06H	单 / 双	15	14	0EH	单
8	7	07H	单 / 双	16	15	0FH	单

5.5.4 查询 A/D 转换状态数据格式:

查询 A/D 转换状态时的数据格式及意义见表 11 (端口地址为基地址+2) :

表 11 A/D 转换状态数据格式 (X 表示任意)

操作命令	D7	D6	D5	D4	D3	D2	D1	D0	A/D 转换状态
$\overline{\text{IOR}}$	1	X	X	X	X	X	X	X	没有或正在转换
$\overline{\text{IOR}}$	0	X	X	X	X	X	X	X	转换结束

5.5.5 A/D 转换结果数据格式:

A/D 转换结果数据格式见表 12:

表 12 A/D 转换结果数据格式

端口地址	操作命令	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	意义
基地址+2	$\overline{\text{IOR}}$	0	0	0	0	DB ₁₁	DB ₁₀	DB ₉	DB ₈	高 4 位数据
基地址+3	$\overline{\text{IOR}}$	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	低 8 位数据

注: 读取低 8 位数据的另一附加作用是清除 A/D 转换结束中断标志, 在每一个 A/D 转换程序段开始时, 应通过空读低 8 位数据初始化 A/D 转换结束中断申请标志。

5.5.6 D/A 转换数据格式:

D/A 转换数据格式见表 13:

表 13 D/A 转换数据格式 (X 表示任意)

端口地址	操作命令	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	意义
基地址+4	$\overline{\text{IOW}}$	DB ₁₁	DB ₁₀	DB ₉	DB ₈	DB ₇	DB ₆	DB ₅	DB ₄	高 8 位数据

基地址+5	IOW	DB ₃	DB ₂	DB ₁	DB ₀	×	×	×	×	低 4 位数据
-------	-----	-----------------	-----------------	-----------------	-----------------	---	---	---	---	---------

5.6 模入模出码制以及数据与模拟量的对应关系:

5.6.1 本接口卡在单极性方式工作时, 即模入模出的模拟量为 0~10V 时, 转换后和写出的 12 位数码为二进制原码。此 12 位数码表示一个正数码, 其数码与模拟电压值的对应关系为:

$$\text{模拟电压值} = \text{数码 (12 位)} \times 10(\text{V}) / 4096 \quad (\text{V})$$

即: 1LSB=2.44mV

5.6.2 本接口卡在双极性方式工作时, 转换后和写出的 12 位数码为二进制偏移码。此时 12 位数码的最高位 (DB11) 为符号位, “0” 表示负, “1” 表示正。偏移码与补码仅在符号位上定义不同, 可以先求出补码再将符号位取反就可得到偏移码。此时数码与模拟电压值的对应关系为:

模入模出信号为 -5~+5V 时:

$$\text{模拟电压值} = \text{数码} \times 10(\text{V}) / 4096 - 5 \quad (\text{V})$$

即: 1LSB=2.44mV

5.7 定时器定时触发信号的要求及使用说明:

本卡的 A/D 转换可以由本卡上的 8253 定时计数器自动定时触发进行, 其定时触发脉冲波形及参数要求见图 12。

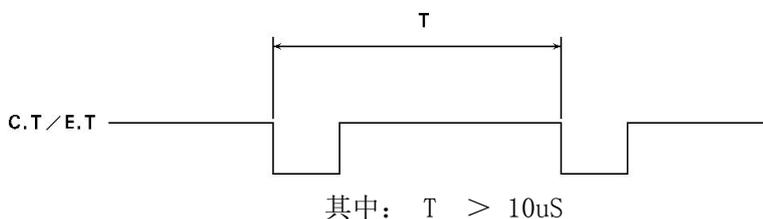


图 12 定时器脉冲或外触发信号波形图

由于 A/D 转换时定时触发启动信号的要求, 我们推荐 8253 定时通道选用原则如下:

- ①当定时间隔小于 65.535mS 时, 应选用一级定时通道来完成。
- ②当定时间隔大于 65.535mS 时, 可选用两级定时通道串联使用, 本卡在设计上已将 8253 定时/计数器的通道 0 和通道 1 接成级连方式, 用户只需将 KJ6 选为 2 连接启动, 并对通道 0、通道 1 进行编程操作即可。
- ③当定时间隔更大时, 如果需要可通过 KJ7 将通道 1 与通道 2 接成级连方式, 并将 KJ6 选为 3 连接启动, 同时对通道 0、1、2 进行编程操作。另外, 在选用定时触发方式前, 应根据定时间隔的要求根据上述通道选用原则选择 8253 的定时/计数通道, 并将 KJ6 设置好, 然后参见 8253 编程要求(说明书 3.6.3 节), 对 8253 进行编程及装入分频系数, 同时在适当的时间对 8253 程控端置位, 8253 定时通道即会产生启动 A/D 转换所需的定时负脉冲, A/D 转换的状态可以通过查询或进行中断申请操作测知。

5.8 外触发信号 E.T 的要求:

本卡的模入部分可以在外触发方式下工作。每当 E.T 有一个低电平时, A/D 就启动转换一次。使用该方式时, 应注意外信号源 E.T 信号必须符合 TTL 电平标准, 其波形也可参见图 12。

5.9 中断工作方式:

本卡的 A/D 转换结束信号可以采用中断方式通知 CPU 进行处理。改变 KJ8 的位置可以选用 IRQ3 至 IRQ7 中断。用户在使用中断方式时, 应对主机系统的 8259 中断管理器进行初始化并编制中断处理程序。并在 8259 中断允许之前, 先清除本卡的中断标志。当 A/D 转换结束时, 本卡会向 8259 中断管理器发出一个高电平的中断申请, CPU 接到中断请求后转向中断处理程序运行读数操作。当读取低 8 位转换结果时, 会自动清除中断标志。

5.10 电流输出方式的使用与扩展:

本卡模出部分可选择 0~10mA 或 4~20mA 电流输出方式以直接驱动 II、III 型执行仪表。采用电流输出方式时, 供电电源可以使用本卡提供的 +12V。也可扩展使用机外 +24V 电源。其连接使用方法见图 13。

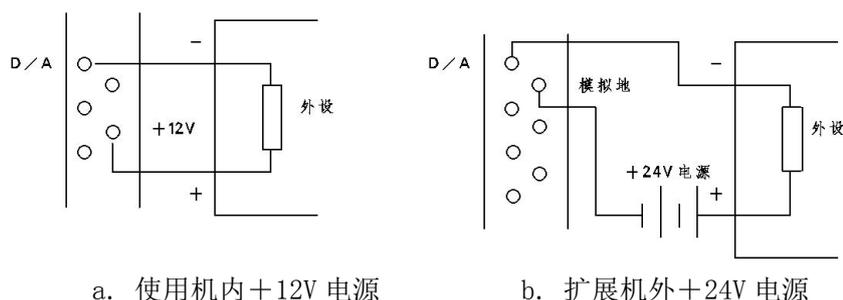


图 13 电流输出方式使用方法

5.11 调整与校准:

5.11.1 产品出厂前, 本卡的模入模出部分均已按照单极性 0~10 V 调整好, 一般情况下用户不需进行调节, 如果用户改变了工作模式及范围, 可按本节所述方法进行调整。调整时应开机预热 20 分钟以上后进行, 并准备一块 4 位半以上的数字万用表。

5.11.2 各电位器功能说明:

- W1 为输入放大器零点调节。
- W2 为 A/D 转换器满度调节。
- W3 为 A/D 转换器双极性偏移调节。
- W4 为 D/A 零点调节。
- W5 为 D/A 满度调节。

5.11.3 模入部分调整:

凡改变模入工作方式, 如果采样结果偏差大于 20mV 以上的, 需要对模入部分进行调整。

①零点调整: 使任一通道与模拟地短接, 并按实际需要设置好通道代码运行程序对该通道采样。用电压表测量 OP-37 运放的第 6 脚, 调整 W1 使电压表读数小于 100 μ V。

②A/D 转换满度调整: 在任一通道接入一接近正满度的电压信号, 运行程序对该通道采样。调整 W2 使 A/D 转换读数等于或接近外信号电压。

③A/D 转换双极性偏移调整: 在单极性方式时, W3 可用于零点辅助调整。在双极性方式时, 如果误差较大, 可在外端口分别加上正负电压信号, 调整 W3 使其对称。

5.11.4 模出部分调整:

凡改变模出部分的方式和量程后, 如果输出结果误差较大, 需要对模出部分进行调整。

①零点调整: 在单极性方式时调整 W4 使其偏差最小。

②满度调整: 在零点调整正常情况下, 如果满度偏差较大, 可通过调整 W5 使满度符合要求。

6. 驱动程序简介:

PC-6000 系列演示程序及驱动程序是为 PC-6000 系列多功能工控采集板配制的工作在中西文 Windows 95/98/NT 环境下的一组驱动程序以及使用该驱动程序组建的一个演示程序, 可以方便地使用用户在中西文 Windows 环境下检测硬件的工作状态以及帮助软件开发人员在常用的 C\C++, Visual Basic, Delphi, Borland C++ Builder, Borland Pascal for windows 等开发环境中使用 PC-6000 系列工控采集板进行数据采集和过程控制等工作。驱动程序是一个标准动态链接库 (DLL 文件)。它的输出函数可以被其它应用程序在运行时直接调用。用户的应用程序可以用任何一种可以使用 DLL 链接库的编程工具来编写。每种板卡依据其自身功能的不同具有不同的输出函数和参数定义。

驱动程序输出函数定义:

所列函数的说明格式为 VC++6.0 环境下 PC6000.D11 库函数的原函数格式, 无论使用哪一种开发工具, 务请注意数据格式的匹配及函数的返回类型, 本说明中所使用的数据类型定义如下:

```
short    ~ 16 位带符号数
unsigned long - 32 位无符号数
unsigned char - 8 位无符号数
```

模拟量输入部分:

* 函数: short APIENTRY AI6333Single(short nAdd, short nCha, short AIMode)

功能: 进行某一通道的模拟量数据采集(程序启动 A/D 方式)。

参数: nAdd 基地址
nCha 通道号: 0 - 15(单端), 0 - 7(双端)
AIMode 输入方式: 0 -- 原码值
1 -- 0, 10v
2 -- -5v, +5v

* 函数: void APIENTRY AI6333AllSingle(short nAdd, short AIMode, short *p)

功能: 单端输入方式下, 全部 16 通道的模拟量数据采集(程序启动 A/D 方式)。

参数: nAdd 基地址
AIMode 输入方式: 0 -- 原码值
1 -- 0, 10v
2 -- -5v, +5v
p 指向 16 个通道的采集结果的起始地址

- * 函数: void APIENTRY AI6333AllDouble(short nAdd, short AIMode, short *p)
 功能: 双端输入方式下, 全部 8 通道的模拟量数据采集(程序启动 A/D 方式)。
 参数: nAdd 基地址
 AIMode 输入方式: 0 -- 原码值
 1 -- 0, 10v
 2 -- -5v, +5v
 p 指向 8 个通道的采集结果的起始地址

- * 函数: void APIENTRY AI6333TimerAD0(short nAdd, short nCha, long nValue)
 功能: 非级联时, 用 8253 定时器的 0 通道定时启动某一通道, 启动 A/D 转换的时间间隔在 0-65.535ms(定时启动 A/D 方式)。
 参数: nAdd 基地址
 nCha 通道号: 0 - 15(单端), 0 - 7(双端)
 nValue 8253 定时器 0 通道预装入值(0-65535)。
 注意: 8253 定时器 0 通道的 CLK 端接至内部时钟(1MHz)。

- * 函数: void APIENTRY AI6333TimerAD1(short nAdd, short nCha, long nValue0, long nValue1)
 功能: 非级联时, 用 8253 定时器的 0, 1 通道进行级连后定时启动某一通道, 启动 A/D 的时间间隔可大于 65.535(ms)。
 参数: nAdd 基地址
 nCha 通道号: 0 - 15(单端), 0 - 7(双端)
 nValue0 8253 定时器 0 通道预装入值(0-65535)。
 nValue1 8253 定时器 1 通道预装入值(0-65535)。
 注意: 板卡将 8253 定时器 0 通道与 1 通道进行级连, 8253 定时器 0 通道的 CLK 端接至内部时钟(1MHz), 0 通道的 OUT 端接至 1 通道的 CLK 端。
 说明: 启动 A/D 的时间间隔的计算公式:

$$Q=1\text{MHz} / \text{定时器0通道分频值N} / \text{定时器1通道分频值M}(\text{其中N尽量小, 太大影响时间间隔精度, 但最小不能为1}).$$

时间间隔和编程参数推荐表

时间间隔Q	定时器0通道分频值N	定时器1通道分频值M
100uS	0002H(2)	0032H(50)
1mS	0002H(2)	01F4H(500)
10mS	0002H(2)	1388H(5000)
100mS	0002H(2)	C350H(50000)
1S	0014H(20)	C350H(50000)

- * 函数: void APIENTRY AI6333TimerAD2(short nAdd, short nCha, long nValue)
 功能: 非级联时, 用 8253 定时器的 2 通道定时启动某一通道, 启动 A/D 转换的时间间隔由外部信号确定(定时启动 A/D 方式)。
 参数: nAdd 基地址
 nCha 通道号: 0 - 15(单端), 0 - 7(双端)
 nValue 8253 定时器 2 通道预装入值(0-65535)。
 注意: 8253 定时器 2 通道接外部信号。

- * 函数: short APIENTRY AI6333TimerRead(short nAdd, short AIMode)
 功能: 非级联时, 进行某一通道的模拟量数据的读取(定时启动 A/D 方式)。
 参数: nAdd 基地址
 AIMode 输入方式: 0 -- 原码值
 1 -- 0, 10v
 2 -- -5v, +5v
 注意: 若返回值为 0, 则由基地址错误或 A/D 未转换完造成。

模拟量输出部分:

- * 函数: void APIENTRY AO6333Single(short nAdd, short nValue, short DAMode)
 功能: 进行某一通道的模拟量数据输出操作。
 参数: nAdd 基地址
 nValue 输出数据: 单位为毫伏(电压方式) 或微安(电流方式)。
 DAMode 输出方式: 0 -- 原码值
 1 -- 0, 10v
 2 -- 0, 5v
 3 -- -5, +5v
 4 -- -2.5, +2.5v
 5 -- 0, 10mA
 6 -- 4, 20mA

返回: 无返回值。

数字量输入输出部分：

- * 函数: unsigned char APIENTRY DI6333Bit(short nAdd, short nBit)
功能: 采集某一位数字量输入信号的状态。
参数: nAdd 基地址
nBit 通道号: 0-5

- * 函数: unsigned char APIENTRY DI6333All(short nAdd)
功能: 采集全部通道(6路) 数字量输入信号的状态。
参数: nAdd 基地址
返回: 返回值为6个输入信号的状态。

- * 函数: void APIENTRY DO6333Bit(short nAdd, short nBit, unsigned char nState)
功能: 进行某一个通道的数字量数据输出操作。
参数: nAdd 基地址
nBit 通道号: 0-5
nState 1表示将输出高电平, 0表示将输出低电平。
返回: 无返回值

- * 函数: void APIENTRY DO6333All(short nAdd, unsigned char nGroup)
功能: 同时进行所有 6 个通道的数字量数据输出操作。
参数: nAdd 基地址
nGroup 全部6个通道的输出状态, nGroup的D0代表Bit0, D3代表Bit3。
返回: 无返回值

计数定时部分：

- * 函数: void APIENTRY CT6333Start(short nAdd, long nPreLoad)
功能: 初始化PC6333, 设置8253计数器芯片第2通道的工作模式及寄存器预装值。
参数: nAdd 基地址
nPreLoad 16位寄存器预装入值(0-65535)。
返回: 无返回值
说明: 此函数将寄存器预装值装入寄存器中, 此时计数器是否开始计数, 还取决于 Gate 端的状态, 通道2工作于方式2。

- * 函数: unsigned long APIENTRY CT6333Read(short nAdd)
功能: 此函数读取某个计数器的寄存器值。
参数: nAdd 基地址
返回: 返回寄存器的当前值。
说明: 此函数不影响计数器的继续计数工作。

如有需要使用 Windows 系列及 LabVIEW 驱动程序的用户可向本公司索取, 请注明所使用的操作系统和开发软件。

编程举例:

6.1 对通道 1 连续采样 100 次, 程序启动和查询。本程序可用于 A / D 部分调校。

```
10 CLS ; 清屏
20 ADD=&H300 ; 板基地址设为 0300H
30 B=INP(ADD+3) ; 空读低 8 位, 清除转换状态及中断标志
40 CH=0 ; 对通道 1 采样
50 OUT (ADD), CH ; 送通道代码
60 FOR J=1 TO 100 ; 设采样次数
70 OUT (ADD+1), 0 ; 启动 A / D, 所送数值无关
80 IF INP(ADD+2) >= 128 THEN 80 ; 查询 A / D 转换状态
90 H=INP(ADD+2) ; 转换结束, 读高 4 位结果
100 L=INP(ADD+3) ; 读低 8 位结果
110 V=(H*256+L) * 10000 / 4096 ; 将结果转换为十进制数据
120 PRINT H, L, V; "mV" ; 显示结果, 用 "mV" 表示
130 NEXT J ; 循环 100 次
140 END
```

注: 如果是双极性信号, 则 110 句改为:

$$V = (H \times 256 + L) \times 10000 / 4096 - 5000$$

6.2 用通道 1 定时触发启动，定时间隔约为 10ms，程序查询取数，采样 1000 组。

```

10 CLS ; 清屏
20 ADD=&H300 ; 设板基地址为 0300H
30 DIM A1(1000) ; 定义数组长度
40 OUT (ADD+&HB), &H35 ; 设 8253 通道 0 工作于
; 方式 2, 十进位 BCD 计数
50 OUT (ADD+&H8), 5 ; 通道 0 分频系数为 5
60 OUT (ADD+&H8), 0
70 OUT (ADD+&HB), &H75 ; 设 8253 通道 1 工作于
; 方式 2, 十进位 BCD 计数
80 OUT (ADD+&H9), 00 ; 定时间隔为 10ms
90 OUT (ADD+&H9), 32 ; 空读低 8 位, 清除转换
; 状态及中断标志
110 CH=&H80 ; 对通道 1 采样,
; 8253 GATE 允许
120 OUT (ADD+0), CH ; 送通道代码及
; 8253 GATE 允许信号
130 FOR T=1 TO 1000 ; 定义循环次数
140 IF INP(ADD+2) >= 128 THEN 140 ; 等待转换结束
150 H=INP(ADD+2) ; 取高 4 位结果
160 L=INP(ADD+3) ; 取低 8 位结果
170 A1(T) = (H*256+L)*10000/4096-5000 ; 处理后存入数组
180 NEXT T ; 循环
; 转数据处理
.....

```

6.3 使 D/A 输出 0V、1000V、3.333V、6.666V、2.000V、8.000V。D/A 工作方式为单极性 0~10V。本程序可用于 D/A 部分的调校。

```

10 CLS ; 清屏
20 BASE%=&H300 ; 板基地址设为 0300H
30 DAH=BASE%+4: ; 设 D/A 高、低字节
DAL=BASE%+5 ; 端口地址
40 FOR I=1 TO 6 ; 设数据指针长度
50 READ A ; 取高字节数据
60 OUT DAH, A ; 送出
70 READ A ; 取低字节数据
80 OUT DAL, A ; 送出
90 READ B ; 取显示用数据
100 A=INP(BASE%+0) ; 启动 D/A 转换, 读出的数据无关
110 PRINT "TEST D/A"; DA; ;
"OUT"; B; "V" ; 显示
120 IF INKEY $ =" " THEN 130 ; 等待, 按任一键继续
130 NEXT I ; 循环送 6 组数据
140 DATA 0, 0, 0, 255, 240, 10.000, ;
85, 80, 3.333, 170, 16, 6.666, ;
51, 48, 2.000, 204, 192, 8.000 ;
150 END ;

```

6.4 读写数字量:

```

10 CLS ; 清屏
20 ADD=&H300 ; 板基地址设为 0300H
30 DO=XX ; 设数据输出为 XX
40 OUT(ADD+7), DO ; 写出并锁存
50 DI=INP(ADD+6) ; 读入数字量状态
60 PRINT DI ; 显示
70 END

```

6.5 对通道 1 连续采样 100 次，程序启动和查询。

C 语言程序：

```
#include "stdio.h"
#include "dos.h"
#include "conio.h"
main()
{
    int ch; /* 定义通道变量 */
    float value[100] /* 定义数组变量 */
    int dl,dh,i,j,base; /* 定义过程变量 */
    clrscr(); /* 清屏 */
    base=0x300; /* 设板基地址=300H */
    printf("Input channle number:"); /* 输入通道号 */
    scanf("%d",&ch);
    outportb(base,ch); /* 送通道代码 */
    for(j=0;j<100;j++){ /* 设采样次数 */
        for(i=0;i<100;i++){ /* 延时，常数由机型决定 */
            outportb(base+1,0); /* 启动 A / D，所送数值
            无关 */
            do{ /* 查询 A / D 转换状态 */
                ;
            }while(inportb(base+2)>=128);
            dh=inportb(base+2); /* 读高 4 位结果 */
            dl=inportb(base+3); /* 读低 8 位结果 */
            value[j]=(dh*256+dl)*10.0/4096.0-5.0; /* 将结果转换为十进
            制数据 */
        }
        for(j=0;j<100;j++) /* 显示结果 */
            printf("%f ",value[j]);
    }
}
```

6.6 循环采集 A / D 32 通道，程序启动和查询。

C 语言程序：

```
#include "stdio.h"
#include "dos.h"
#include "conio.h"

main()
{
    int ch; /* 定义通道变量 */
    float value[32]; /* 定义数组变量 */
    int dl,dh,i,base; /* 定义过程变量 */
    clrscr(); /* 清屏 */
    for(ch=0;ch<=31;ch++){ /* 定义循环通道数 */
        base=0x300; /* 设板基地址=300H */
        outportb(base,ch); /* 送通道代码 */
        for(i=0;i<1000;i++){ /* 延时，常数由机型决定 */
            outportb(base+1,0); /* 启动 A / D，所送数值无关 */
            do{ /* 查询 A / D 转换状态 */
                ;
            }while(inportb(base+2)>=128);
            dh=inportb(base+2); /* 转换结束，读高 4 位结果 */
            dl=inportb(base+3); /* 读低 8 位结果 */
            value[ch]=(dh*256+dl)*10.0/4096.0-5.0; /* 将结果转换为十进
            制数据 */
        }
    }
}
```

```

                                制数据    */
                                /* 下一个通道 */
}                                /* 显示结果 */
for(ch=0;ch<=31;ch++)          /* 显示结果 */
    printf("%f ",value[ch]);
}

```

6.7 在 Windows 95/98 环境下，使用 MicroSoft Visual Basic 6.0 开发环境，采取调用驱动程序的输出函数的方法定时循环采集多个 A/D 通道，并对 D/A、数字 I/O 进行操作和计数器进行操作。

注意：在 VB 6.0 中，数据类型 Integer 为 16 位带符号整数；Byte 为 8 位无符号整数；Long 为 32 位带符号整数。

首先创建一个窗口，名为 Form1。设置一个定时器，名为 Timer1；一个 Text1；一个 Text3；一个 Text2 数组，Text2[]。

```

Private Declare Function AI6333Single Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nCha As Integer,
ByVal AIMode As Integer) As Integer
Private Declare Sub AI6333AllSingle Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal AIMode As Integer,
ByRef p As Integer)
Private Declare Sub AI6333AllDouble Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal AIMode As Integer,
ByRef p As Integer)
Private Declare Sub AI6333TimerAD0 Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nCha As Integer,
ByVal nValue As Long)
Private Declare Sub AI6333TimerAD1 Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nCha As Integer,
ByVal nValue0 As Long, ByVal nValue1 As Long)
Private Declare Sub AI6333TimerAD2 Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nCha As Integer,
ByVal nValue As Long)
Private Declare Function AI6333TimerRead Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal AIMode As
Integer) As Integer
' 模出部分
Private Declare Sub A06333Single Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nValue As Integer,
ByVal DAMode As Integer)
' 数字量
Private Declare Function DI6333Bit Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nBit As Integer)
As Byte
Private Declare Function DI6333All Lib "pc6000.dll" (ByVal nAdd As Integer) As Byte
Private Declare Sub D06333Bit Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nBit As Integer, ByVal
nState As Byte)
Private Declare Sub D06333All Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nGroup As Byte)
' 计数器
Private Declare Sub CT6333Start Lib "pc6000.dll" (ByVal nAdd As Integer, ByVal nPreLoad As Long)
Private Declare Function CT6333Read Lib "pc6000.dll" (ByVal nAdd As Integer) As Long

Dim a(16) As Integer      ' 数组元素个数应>=16

Private Sub Form_Load()
    Call CT6333Start(256, 65535)
End Sub

Private Sub Timer1_Timer()

    ' 多通道采集 16 通道，0-15 通道采集结果分别存放在 a(0)-a(15) 中
    Call AI6333AllSingle(256, 1, a(0))
    For i = 0 To 15
        ' Text2(i) = a(i)      ' 界面显示 16 通道数据
    Next i

    ' 单通道采集第 0 通道

```

```

Text1 = AI6333Single(256, 0, 1)

' D/A 输出
Call A06333Single(256, AoOutValue, 1)
If AoOutValue = 10000 Then
    AoOutValue = 0
Else
    AoOutValue = AoOutValue + 100      ' 改变 D/A 输出电压值
End If

' I/O 输出
Call D06333A11(256, DoOutValue)
If DoOutValue = 0 Then
    AoOutValue = 255
Else
    AoOutValue = 0                    ' 改变数字量输出状态.
End If

' I/O 输入
Text3 = DI6333A11(256)

' 计数器计数
Label = CT6333Read(256)

End Sub

```

附 A. 名词注释：

- 1. 单端输入方式：**各路输入信号共用一个参考电位，即各路输入信号共地，这是最常用的接线方式。使用单端输入方式时，地线比较稳定，抗干扰能力较强，建议用户尽可能使用此种方式。
- 2. 双端输入方式：**各路输入信号各自使用自己的参考电位，即各路输入信号不共地。如果输入信号来自不同的信号源，而这些信号源的参考电位(地线)略有差异，可考虑使用这种接线方式。使用双端输入方式时，输入信号易受干扰，所以，应加强信号线的抗干扰处理，同时还应确保模拟地以及外接仪器机壳接地良好。而且特别注意的是，所有接入的信号，不论是高电位还是低电位，其电平相对于模拟地电位应不超过 +12V 及 -5V，以避免电压过高造成器件损坏。
- 3. 单极性信号：**输入信号相对于模拟地电位来讲，只偏向一侧，如输入电压为 0 ~ 10v.
- 4. 双极性信号：**输入信号相对于模拟地电位来讲，可高可低，如输入电压为 -5v ~ +5v.
- 5. 码制：**模拟量信号转换为数字量后，形成一组由 0 开始的连续数字，每一个数字对应着一个特定的模拟量值，这种对应关系称为编码方法或码制。依据输入信号的不同分为单极性原码与双极性偏移码。单极性输入信号对应着单极性原码，双极性信号对应着双极性偏移码。
- 6. 单极性原码：**以 12 位 A/D 为例，输入单极性信号 0~10V。转换后得到 0 ~ 4095 的数字量，数字量 0 对应的模拟量为 0V，数字量 4095 对应的模拟量为 10V，这种编码方法称为单极性原码，其数字量值与模拟电压值的对应关系可描述为：

$$\text{模拟电压值} = \text{数码}(12 \text{ 位}) \times 10(\text{V}) / 4096 \quad (\text{V})$$
 即：1LSB (1 个数码位) = 2.44mV
- 7. 双极性偏移码：**以 12 位 A/D 为例，输入双极性信号 -5~+5V。转换后得到 0 ~ 4095 的数字量，数字量 0 对应的模拟量为 -5V，数字量 4095 对应的模拟量为 +5V，这种编码方法称为双极性偏移码，其数字量值与模拟电压值的对应关系可描述为：

模拟电压值 = 数码(12位) × 10(V) / 4096 - 5 (V)

即: 1LSB (1个数码位) = 2.44mV

此时 12 位数码的最高位 (DB₁₁) 为符号位, 此位为 0 表示负, 1 表示正。偏移码与补码仅在符号位上定义不同, 如果反向运算, 可以先求出补码再将符号位取反就可得到偏移码。

8. A/D 转换速率: 表明 A/D 转换芯片的工作速度。如对 BB774 来讲, 完成一次转换所需要的时间是 10 微秒, 则它的转换速率为 100 KHz。

9. 通过率: 指 A/D 采集卡对某一路信号连续采集时的最高采集速率。

10. 初始地址: 使用板卡时, 需要对卡上的一组寄存器进行操作, 这组寄存器占用数个连续的地址, 一般将其中最低的地址值定为此卡的初始地址, 这个地址值需要使用卡上的拨码开关来设置。

附 B. 产品清单及保修:

产品清单:

1. PC-6333 多功能模入模出接口卡壹块。
2. 37 芯 D 型插头壹套。

本产品自售出之日起一年内, 凡用户遵守贮存、运输及使用要求, 而产品质量低于技术指标的, 凭保修单免费维修。因违反操作规定和要求而造成损坏的, 需交纳器件和维修费。