

入门指南

We Move the World

DMC-9940

Manual Rev. 1.0c

By Powerly Motion Technologies

www.mtt.com.tw

Rev 06-03

We Move the World

第一章 简介	错误！未定义书签。
一. 楔子	2
二. 推荐 DMC-9940	2
三. 介绍 DMC-9940	2
第二章 快速入门	错误！未定义书签。
一. 所需的组件:	5
二. 安装:	5
三. 使用软件:	7
第三章 硬件连接	错误！未定义书签。
一. 极限开关	9
二. 原点开关	9
三. 急停输入	9
四. 一般输入	9
五. TTL 输出	9
第四章 联机	错误！未定义书签。
前言:	错误！未定义书签。
一. 速度信息说明:	10
二. QZ 指令:	10
三. 中断:	10
四. 软件工具 及 函式库:	10
第五章 指令使用入门	错误！未定义书签。
一. 基本指令	10
二. ASCII 指令	10
三. Binary 指令	11
第六章 程序范例	错误！未定义书签。
实例一 梯型速度曲线位移	错误！未定义书签。
实例二 多轴运动	错误！未定义书签。
实例三 独立运动	错误！未定义书签。
实例四 位置及误差查询	错误！未定义书签。
实例五 绝对位移	错误！未定义书签。
实例六 速度控制	错误！未定义书签。
实例七 使用编辑器	错误！未定义书签。
实例八 程序循环	错误！未定义书签。
实例九 程序中之触发点	错误！未定义书签。
实例十 控制变量	错误！未定义书签。
实例十一 直线补间	错误！未定义书签。
实例十二 圆弧补间	错误！未定义书签。
附录	错误！未定义书签。

第一章 简介

一. 楔子

我们让世界前进 (Galil Motion Control – We Move the World)

给我一根够长的杠杆，我单手便可移动世界. --阿基米德

(Give me a lever long enough ... and a single-handed I can move the world.

-- Archimedes)

Galil Motion Control于1983年创立于美国硅谷，推出世界第一套 建构于 micro processor 的运动控器，迄今已有 30 万套安装于世界各个角落，为各种精密机械及仪器，提供高精度的马达控制与完美的运动模式o

Galil Motion Control是业界公认的运动控制先驱与领航者，永不止息的从事创新,以先进而稳定的精密控制，简便的使用方式，坚实的技术支持与最佳的成本效益等四大特色，稳居运动控制领域的翘楚地位o

Galil Motion Control 的创办人兼总裁 Dr. Jacob Tal 是众所周知的学者与教授，发表文章逾一百余篇，著名的专业书籍 "Motion Control by Microprocessors", "Motion Control Applications" 以及 "Step-by-Step Design of Motion Control Systems", 均常被人广泛引用o 在实务上, Dr. Tal 卅余年来，以独到而领先的见解，为无数公司解决了运动控制上的难题o 来至全球各地的一万多名工程师，曾出席 Dr. Tal 所主持的课程或研讨会，亲炙大师的风采o

二. 推荐 DMC-9940

DMC-9940 承袭了 Galil 优质 (Optimal) 系列 DMC 1800各项卓越性能，完整保留 Galil 所有运动特性，板卡本身具有光隔离保护，仅移除模拟控制，以及一些罕用功能，如双 encoder 回馈等，可用来控制步进及伺服马达.DMC-9940 是一精心设计，严密品管的产品，绝对物超所值，超高性能/价格比，是世界级的首选运动控制器o

三. 介绍 DMC-9940

DMC-9940 是采用PCI 总线之四轴控制器，功能强大，物超所值，是您开发新设备的最佳拍档o

DMC-9940之使用亦十分方便，直觉式指令均以两个英文字母组成o丰富的支持软件，如: WSDK 用于调试与分析, ActiveX 工具箱用于Visual Basic程序开发，并提供免费C-Programmers 工具箱 o

特色

(一)快速的沟通方式

1. PCI bus接口

2. 双位或ASCII沟通模式

(二)强大的控制功能

1. 控制1~4轴
2. 编码器回馈频率: 12MHZ
3. 步进指令: 3MHZ (脉冲及方向)

(三)先进的运动模式

1. 八组独立多任务程序
2. 运动模式: 速度控制, 点对点定位, 线性及圆弧补间, 电子齿轮&凸轮, 连续路径运动, 原点复归。

(四)无以伦比的运动特性

1. 速度/加速度立即控制功能, 提供最佳之运动精度
2. 可程序之速度/加速度曲线, 并佐以速度平稳功能以消除加速度之陡变 (急冲度, jerk)
3. 可处理无限多组之直线及圆弧补间
4. 可执行双曲面及球面等连续路径运动
5. 每次运动可执行2Giga个分辨率之位移
6. 各轴高速定位连结
7. 以总线中断处理高速指令

(五)卓越之硬件配备

1. 以32位专用处理器及特制次微米网关数组, 提供最高性能
2. 可自由使用8组光耦合输入点(5-28VDC)及8组可程序输出点
3. 可选配额外64 I/O
4. 八组12bit ADC输入(选配)
5. 100-pin高密度连接器可减少电磁干扰

(六)最佳的程序发展环境\

1. 非挥发性程序记忆
2. 快闪EEPROM韧体更新, 及储存程序, 参数, 数组
3. 位置比较输出
4. 可程序事件驱动
5. 可执行PLC工作之I/O功能及定时器
6. 程序编写容易, 软件支持丰富, 可快速开发并修改应用程序
7. 容错处理功能: 可程序之软件极限, 错误时自动停机, 驱动器致能, 自订错误处理子程序, 看门狗定时器
8. 提供定位、速度、错误及输入之状态报告
9. 电源激活自动执行程序
10. 支持Windows Active-X工具箱
11. 支持C++ DMCWIN 工具箱

规格

(一)卓越性能

1. 执行容量:连续路径模式,全轨迹运算,每秒可执行500次移动
2. 定位精度: ± 1 个脉冲
3. 速度精度:十万分之三以下
4. 同步控制: (1)同卡之各轴完美地同步运动
(2)各卡之讯号同步

(二)参数值之范围

1. 位移范围: $\pm 2,147,483,647$ counts/move;自动更新;无限延伸之速度控制及向量模式
2. 速度范围: Up to 12,000,000 counts/sec
3. 加/减速度: 1,024 to 67,107,840 counts/sec²
4. 齿速比: ± 127.9999
5. 步进马达控制模式:全步进,半步进,微步进
6. 步进脉冲频率: 3,000,000 pulses/sec
7. 容量
 - 变量之个数: 254
 - 数组大小: 8000个数组单元, 30组数组
 - 记忆容量: 1000行 $\times$ 80字

(三)输入/输出

1. 回馈:
 - A.B相及参考点
 - 单向或差动
 - 可配置为A.B相四分割或脉冲/方向模式
2. 一般输入: 8点缓冲输入
3. 一般输出: 8个TTL 输出点
4. 各轴专属输入点:缓冲前进及后退极限,高速位置捕捉,返回原点
5. 各轴专属输出点: 脉冲与方向,驱动器致能, 位置输出比较
6. 轴数: 1~4轴

(四)沟通接口

PCI bus总线及高速双向FIFO

(五)作业平台

可用于Linux, QNX, DOS, 及所有Windows作业平台

第二章 快速入门

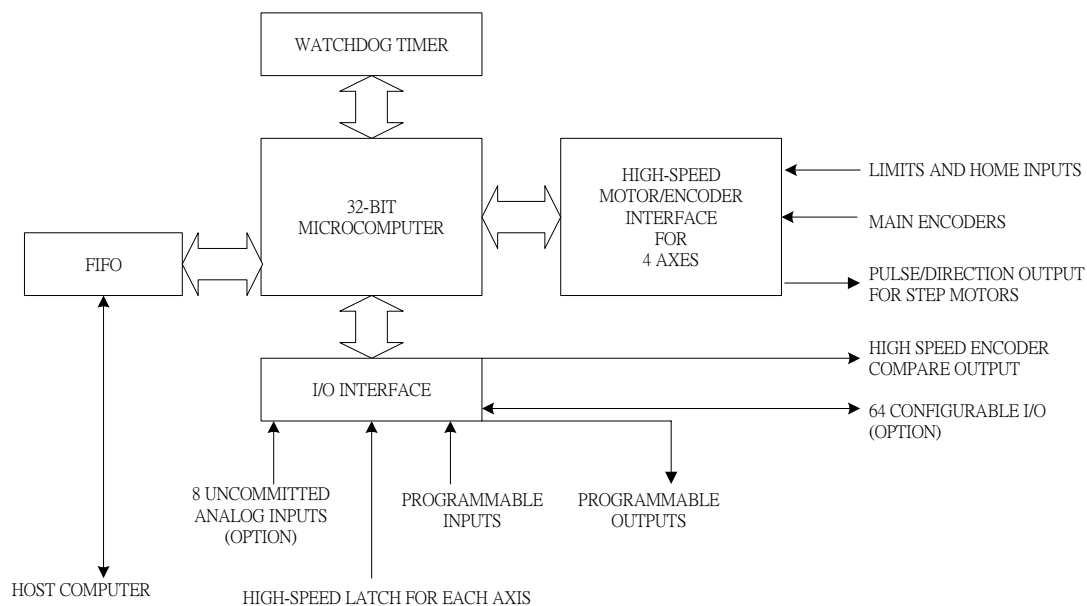
一. 所需的组件:

1. 硬件组件:

DMC9940控制卡,100 PIN缆线,100 PIN端子台

1. 马达
2. 驱动器
3. 驱动器及端子台所需的电源供应器
4. 各种相关缆线
5. 个人计算机
6. 软件

2. 功能组件图:



二. 安装:

安装DMC-9940分为以下6步骤

1. 选用马达
2. 安装工作软件
3. 将控制卡装至PC上
4. 连结控制卡与PC
5. 连结驱动器, ENCODER
6. 连结马达

1. 选用马达

选择适合的马达

2. 安装工作软件

可用以下三种作业平台来安装软件

DOS:

放入光盘片, 键入"INSTALL", 并随着指引步骤来完成安装

Windows 3.1:

放入光盘片, 并执行"DMCWIN16.exe", 并随着指引步骤来完成安装

Windows 95, 98, 2000, XP or NT:

放入光盘片, 并执行"DMCWIN32.exe", 并随着指引步骤来完成安装

3. 将控制卡装至PC上

- a. 确认PC是在关机状态下
- b. 打开机壳
- c. 将DMC-9940插在PCI插槽上, 并用螺丝锁紧
- d. 装上100PIN排线
- e. 打开电源

4. 连结控制卡与PC

可用以下三种作业平台来进行连结

DOS:

选择正确的控制器类别, 并进行连结(选择DMC-1800)

Windows 3.1:

开启GALIL软件, 并选择"FIND"键, 来进行连结(选择DMC-1800)

Windows 95, 98, 2000, XP, NT:

使用GALIL软件时, 可在随时进入登录档去增加, 删除或修改控制卡信息. 如果登录档中只有一片控制卡, 则软件会自动连结, 如有多片时请选择所要连结的卡(选择DMC-1800)

5. 连结驱动器及ENCODER

步骤:

- a. 在驱动器与控制卡不连接的状态下, 将马达连结到驱动器
- b. 将驱动器的致能讯号连接到控制卡
(警告) 若驱动器的ground未与电源线隔离, 或是不同于计算机ground的电压时, 会严重损害到计算机, 控制卡, 及驱动器
- c. 连接ENCODER
对步进马达而言, ENCODER不是必需的
- d. 确认ENCODER安装正确
移动X轴的ENCODER, 使用TPX指令去确认位置值, 在反转确认一次

6. 连结马达

- a. 将控制卡的脉冲与方向讯号连结到马达的驱动器
- b. 使用MT2指令或MT-2指令来设定马达为常开或常闭

三. 使用软件:

(以 Windows 平台为例)

DMC 9940提供 WSDK, Terminal, Editor并支持VB, VC++ 等多种软件, 可依需求选用, 提供弹性, 方便软件开发环境o

1. 使用WSDK

为选购产品, 依手册操作使用, 可用来进行系统设定与评估诊断o

2. 使用 Terminal

在执行 ‘Dterm16.exe’ 或 ‘Dterm32.exe’ 时, 系统自动产生 “Galil_Terminal” 图标 (icon), 请点选 “Galil_Terminal” 图标, 在选单中点选 “Terminal”, 系统将呈现 GALIL卡型号, 总线形式, 及卡在系统中的逻辑地址o.

点选 ‘OK’ 进入Terminal, 可键入相关指令, 并获得系统响应, 如

```
TPX
TP
PR10000,10000,3000
XQ #A
```

3. 使用程序编辑器 Editor

在 Terminal 中点选 “Editor”, 键入相关指令, 组成一个完整的可执行程序, 例如

```
#A
SP 1000000,200000
AC 20000,30000
DC 20000,30000
PR 100000,20000
BGX
AMX
WT 500
BGY
AMY
WT 500
EN
```

可上传, 下传 并储存此一程序o

DLLs with Visual Basic

在使用DLLs时, 首先要把 ‘\dmcwin\vb\DMCCOM40.BAS’ 模块加入系统o

Visual Basic 使用范例:

```

Dim Controller,hDmc,RC,ResponseLength as Integer
Dim Response as String *256
ResponseLength=256
Controller=1

//Connect to controller number 1
RC=DMCOpen(Controller,0,hDmc)

//read x axis position
RC=DMCCommand(hDmc,"TPX",Response,ResponseLength)
Text1.Text=Response

```

4. DLLs with C/C++

在程序中必须包含 'DMCCOM.H'.

C/C++ 使用范例:

```

#include <windows.h>
#include <Dmccom.h>
long rc;
HANDLEDMC hDmc;
HWND hWnd;
Int main(void)
{
    //Connect to controller number 1
    rc=DMCOpen(1,hWnd,&hDmc);
    if(rc==DMCNOERROR)
    {
        CHAR SZbUFFER[64];

        //read x axis position
        rc=DMCCommand(hDmc,"TPX",szBuffer,sizeof(szBuffer));

        //More commands

        //Disconnect from controller number 1 as the last action
        rc=DMCClose(hDmc);
    }
    return 0;
}

```

使用 ActiveX

在 VB, VC++, DELPHI 及 Labview 中开发程序
(Activex 为选购软件)

以 Delphi 为例

在使用OCX时, 首先要把 'dmccox\ocx\Dmcsell.ocx' 模块加入系统o

DELPHI 使用范例

```
//Connect to controller number 1
form1.DMCSHell1.DMCCConnect:=True;

//read x axis position
form1.DMCSHell1.DMCCCommand:='TP';
str1:=form1.DMCSHell1.Dmcreponse;
form1.edit2.text:=str1;

//Disconnect from controller number 1 as the last action
form1.DMCSHell1.DMCCConnect:=False;
```

第三章 硬件连接

一. 极限开关

前极限开关 (FLSx) 的动作是在设定马达前进时的最大行程, 后极限开关 (RLSx) 的动作是在设定马达后退时的最大行程, 可使用CN指令来设定极限开关的常态状况。

当前极限开关动作时, 正在执行中的程序会中断, 然后自动跳到 #LIMSWI 的标记位置去执行极限子程序。

二. 原点开关

DMC-9940有提供三种原点子程序: Find Edge(FE), Find Index(FI), and Stardard Home(HM)。

三. 急停输入

急停输入的功能: 当输入点的逻辑在做转换时, 立即停止控制器的所有动作。

四. 一般输入

DMC-9940有提供八个光隔离输入, 这些输入可使用 @IN[x] 或 TI去读取状态 (x表示你所要读取的输入点编号1-8)。

五. TTL输出

DMC-9940提供八个标准TTL输出跟一个Error讯号。

请参考SB(Set Bit) 、CB(Clear Bit) 、OB(Output Bit)、OP(Output Port) 等指令。

第四章 联机

前言:

DMC-9940可以从PC/XT/AT或是其它兼容的计算机接受控制指令

一. 速度信息说明:

当使用指令 TV(Tell Velocity)会从控制器响应速度信息。相关资料请查阅指令手册。

二. QZ 指令:

使用QZ 指令传回 4 bytes 信息。

BYTE#	INFORMATION
0	轴数
1	Data Record共同区 的Bytes数
2	Data Record 中 坐标平面区块的Bytes数
3	Data Record 中 各轴区块的Bytes数

三. 中断:

DMC9940提供硬件的中断线路去中断计算机轮询 (poll) PCI 总线 事件(event) 之出现.
请参考 EI, UI 等指令

四. 软件工具 及 函式库:

DMC9940提供API(Application Programming Interface)软件, 可用来发展应用程序。
亦可选购 ActiveX 工具箱开发应用程序。

第五章 指令使用入门

一. 基本指令

GALIL DMC-9940提供100多个指令, 用来设定运动及机械参数。所有ASCII指令均由两个英文大写字母组成, 程序编写简易快捷。除ASCII外亦可使用Binary指令。

二. ASCII指令

例如, “BG” 指令 (begin的缩写) 表示开始运动, 而 “ST” 指令(stop的缩写)表示停止运动, “SP2000, 4000” (speed的缩写) 设定X轴的速度为 “2000”, Y轴的速度为 “4000”。

举例1: (PR = Position relative, 相对位移)

PR 1000 设定X轴移动1000 单位

PR,2000 设定Y轴移动2000单位

PR,,3000 设定Z轴移动3000单位

PR,,,4000 设定W轴移动4000单位

PR 2000,4000,6000,8000 设定X,Y,Z W轴移动数值

PR,8000,,9000 设定Y, W轴移动数值

PR?,?.,? 查询X,Y,Z,W,轴移动数值

PR,? 查询Y轴移动数值

*PR x,y,z,w 表 设定 X, Y, Z, W 各轴数值

*逗号前无数值

- 1)若曾指定数值, 维持原指定值
- 2)若未指定数值, 维持系统预设(default)值

* “?” 表查询该轴数值

另一种表示法为

PRX=1000 指定X轴相对位移为1000

ACY=200000 指定Y轴加速度为200000

举例2: (BG=Begin)

BGX 开始X轴动作

BGY 开始Y轴动作

BGXYZW 开始每一轴动作

BGYW 开始Y和W轴动作

BG 开始每一轴动作

举例3: 多轴同步运动

BGS S坐标之运动

BGSW 开始 S坐标及W轴之运动

BGT 开始T坐标之运动

(请参考 指令手册 CA 指令, S、T 各代表一组多轴同动坐标系统)

三. Binary指令

Binary 沟通模式的执行速度比 ASCII 快速, 且仅能由 PC 传送, 不能嵌套在应用程序中. Binary 指令由 4 个 byte 开头形成采十六进制, 其后紧接数据域位.

1. Byte 1 指令名称

COMMAND	NO.	COMMAND	NO.	COMMAND	NO.
AF	85	LM	b0	AD	cd
ER	88	LI	b1	AP	ce
MT	8b	VP	b2	AR	cf
CE	8c	CR	b3	AS	d0
OE	8d	TN	b4	AI	d1
FL	8e	LE,VE	b5	AT	d2
BL	8f	VT	b6	WT	d3
AC	90	VA	b7	WC	d4
DC	91	VD	b8	RP	d8
SP	92	VS	b9	TP	d9
IT	93	VR	ba	TE	da
GR	96	CM	bd	TD	db
DP	97	CD	be	TV	dc
DE	98	DT	bf	RL	dd
GM	9a	ET	c0	TS	df
BG	a0	EM	c1	TI	e0
ST	a1	EP	c2	SC	e1

AB	a2	EG	c3	CN	e6
HM	a3	EB	c4	LZ	e7
FE	a4	EQ	c5	OP	e8
FI	a5	EC	c6	OB	e9
PA	a6	AM	c8	SB	ea
PR	a7	MC	c9	CB	eb
JG	a8	TW	ca	II	ec
MO	a9	MF	cb	EI	ed
SH	aa	MR	cccb	AL	ee

2. Byte 2 设定数据域位的 byte数

- 00 无数据域位
- 01 每一个字段均为一个byte
- 02 每一个字段均为二个byte
- 04 每一个字段均为四个byte
- 06 4 bytes 整数, 2 bytes 小数

3. Byte 3 选择运动模式

- 00 个轴运动
- 01 多轴同动

4. Byte 4 指定动作轴或数据域位

- Bit 7 = 第 8 个数据域位
- Bit 6 = 第 7 个数据域位
- Bit 5 = 第 6 个数据域位
- Bit 4 = 第 5 个数据域位
- Bit 3 = D 轴 或 第 4 个数据域位
- Bit 2 = C 轴 或 第 3 个数据域位
- Bit 1 = B 轴 或 第 2 个数据域位
- Bit 0 = A 轴 或 第 1 个数据域位

5. 数据域位

数据域位应与上述各 byte 的定义相符合

举例1: PR1000,-500表示法为A7 02 00 05 03 E8 FE OC

- A7 代表PR
- 02 代表数据域位为2bytes
- 00 代表S 坐标系统不执行 PR 指令
- 05 代表($2^0 + 2^2 = 5$) A和C轴执行 PR 指令
- 03 E8 代表A轴数值为1000
- FE OC 代表C轴数值为-500

举例2: ST XYZS表示法为A1 00 01 07

- A1 代表ST(STOP)
- 00 代表不需数据域位
- 01 代表停止同动坐标系统 S 之运动

07 代表($2^0+2^1+2^2=7$)指定X(bit 0),Y(bit 1),Z(bit 2)轴停止运动

6. 回传指令错误讯息

举例:

将BG<enter>指令打成bg<enter>
DMC-9940会回传 “?” 表示无效指令

使用 TC (tell error code) 指令 传回 错误讯息码
键入TC1<enter>
DMC-9940 回传1, 表无法辨识的指令 (Unrecognized command)
(请 参考指令手TC 之说明)

7. 查询 键入 控制器 应答指令

查询指令汇总

RP	Report Command Position
RL	Report Latch
^R^V	Firmware Revision Information
SC	Stop Code
TB	Tell Status
TC	Tell Error Code
TD	Step counts output
TE	Tell Error
TI	Tell Input
TP	Tell Position
TR	Trace
TS	Tell Switches
TV	Tell Velocity

举例:

1. TPX<enter> 请告知X轴位置 DMC-9940回复为 0000000000
- 2.TPX Y<enter> 请告知XY轴位置
DMC-9940回复为 0000000000,0000000000

询问现在指令数值

举例:

1. PR?,?,? 请告知现在X Y Z W 轴数值
2. PR,? 请告知现在Y 轴数值

8. 操作数 (Operands)

举例:

1. MG ‘operand’ 表 显示讯息
- 6.V=_TPX 为指定变量V为X轴现值

若某一指令有相对应的操作数, 在指令手册 (Command Reference)中均逐一注明.

9. DMC-9940 指令集

Stepper Motor

DE	定义encoder位置	Define encoder position
DP	设定步进马达参考点位置	Define reference position
KS	设定步进平滑常数(平稳的步进运动)	Stepper motor smoothing
MT	选择马达型式及驱动信号的极性	Motor type
RP	回传所有指令位置	Report commanded position
TD	回传控制器所输出的步进脉冲数	Step counts output
TP	回传位置	Tell position of encoder

I/O

AL	激活高速位置捕捉(or位置截取/锁存/栓锁)	Arm latch
CB	清除I/O点之状态(low)	Clear bit
CO	规划输出/入点 (extended I/O)	Configure I/O points
II	激活输入点中断功能	Input interrupt
OB	定义输出点	Define output bit
OC	产生"比较"脉冲讯号	Output compare function
OP	传送资料至控制器输出端口	Output port
SB	设定各输出点状态(high)	Set bit
UI	产生 IRQ 中断信号	User interrupts

System Configuration

BN	将参数存入EEPROM	Burn parameters
BP	将程序存入EEPROM	Burn program
BV	将变量及数组存入EEPROM	Burn variables and arrays
CE	规划encoder型式	Configure encoder type
CN	规划极限,原点,Latch输入的极性; 规划急停或一般输入点	Configure switches
CO+	规划输出/入点(extended I/O)	Configure I/O points
CW	显示版本信息	Copyright information
DP	定义位置	Define position
EI	激活中断模式	Enable interrupts
EO	关闭回传	Echo off
IT	设定单轴时间平滑常数(平稳运动 ,S curve)	Independent smoothing
LZ	显示/隐藏 有效值之前的0值	Leading zeros format
MO	关闭马达控制功能,控制器仍在侦测马达位置	Motor off
MT*	选择马达型式及驱动信号的极性	Motor type
PF	定义位置值显示格式	Position format
QD	下载数组	Download array
QU	上传数组	Upload array
RS	回复成开机时的参数值及控制器状态	Reset
SH	开启马达控制功能,清除位置指令与误差	Servo Here
^R ^ S	清除Eprom值,控制卡设定为出厂默认值	Master reset
VF	设定变量格式	Variable format

+ 重复列表 (I/O) * 重复列表 (Stepper Motor)

Math/Special Functions

@SIN[x]	正弦函数	Sine of x
@COS[x]	余弦函数	Cosine of x
@COM[x]	1 的补码	1's compliment of x

@ASIN[x]	反正弦函数	Arc sine of x
@ACOS[x]	反余弦函数	Arc cosine of x
@ATAN[x]	正切函数	Arc tangent of x
@ABS[x]	绝对值	Absolute value of x
@FRAC[x]	小数	Fraction portion of x
@INT[x]	整数	Integer portion of x
@RND[x]	四舍五入	Round of x
@SQR[x]	平方根	Square root of x
@IN[x]	回传数字输入状态	State of digital input x
@OUT[x]	回传数字输出状态	State of digital output x
@AN[x]	回传模拟输入值	Value of analog input x

Interrogation

LA	回传所有数组名及大小	List arrays
LL	回传程序之所有旗标	List labels
LS	回传程序内容	List program
LV	回传所有变量的名称及记忆值	List variables
MG	显示讯息	Message command
QR	回传控制器信息记录	Data record
QZ	回传特定轴相关信息	Return Data Record information
RL	回传上一个截取位置	Report latch
RP	回传命令位置	Report command position
^R ^V	显示韧体信息	Firmware revision information
SC	回传马达停止的原因代码	Stop code
TB	回传控制器状态	Tell status
TC	回传错误代码	Tell error code
TI	回传输入点状态	Tell input
TIME		Time operand or Value of the internal clock
TP	回传马达位置	Tell position
TR	记录指令用于侦错	Trace program
TS	回传原点/极限开关, 错误极限, 锁存, 马状及运动状态	Tell switches
TV	回传运动轴速度	Tell velocity

Programming

DA	清除/释放 变量及数组记忆值	Deallocate variables/arrays
DL	下载程序	Download program
DM	定义一维数组名及大小	Dimension arrays
ED	进入编辑模式	Edit program
ELSE	条件式_"IF"叙述不成立时, 执行下一行指令	Conditional statement
ENDIF	结束IF叙述指令	End of cond. statement
EN	结束程序	End program
HX	中断程序执行	Halt execution
IF	假设叙述	If statement
IN	输入变量	Input variable
JP	跳至卷标执行程序	Jump
JS	跳至子程序	Jump to subroutine
NO	作批注.不执行任何动作	No-operation—for remarks
RA	选择需要记录的数组	Record array
RC	开始记录数组,指定时间间隔及记录笔数	Record interval
RD	指定记录资料的类别	Record data

REM	批注	Remark program
RI	(由中断子程序)返回主程序	Return form Interrupt Routine
UI	传送中断讯号至所选定之IRQ	User interrupt
UL	从控制器上传程序至计算机	Upload program
ZS	清除堆栈区	Zero stack

Error Control

BL	设定软件后极限	Backward software limit
ER	设定(最大)误差极限	Error limit
FL	设定前极限	Forward software limit
OE	激活/关闭"误差极限"处理功能	Off-on-error function
TW	设定各轴"定位超时"的时间	Timeout for in-position

Trippoint

AD	到位后(马达激活后的增量位移)	After distance
AI	指定输入点状态改变后	After input
AM	运动完成后	After motion profiler
AP	到位后(绝对位置)	After absolute position
AR	到位后(上次执行AR/AD 后的增量位移)	After relative distance
AS	达到指定速度后	At speed
AT	达到某相对时间后	After time
AV	到位后(向量位置)	After vector distance
MC	控制器送出"指令位置"所需的所有脉冲后	Motion complete
MF	顺向行进至某一个位置后	After motion—forward
MR	逆向行进至某一个位置后	After motion—reverse
WC	等待轨迹资料	Wait for contour data
WT	暂停	Wait for time

Independent Motion

AB	急停(运动 or 运动及程序)	Abort
AC	设定加速度	Acceleration
BG	开始运动	Begin motion
DC	设定减速度	Deceleration
FE	寻找(原点开关)的边缘	Find edge
FI	寻找(编码器)原点	Find index
HM	返回原点	Home
HX	中断程序或执行绪(thread)	Halt Execution
IP	即刻改变位置(在运动状态下)	Increment position
IT*	设定单轴时间平滑常数(平稳运动 ,S curve)	Smoothing time constant
JG	设定速度控制模式各轴的稳态速度	Jog mode
PA	设定各轴的绝对位置(位移)	Position absolute
PR	设定各轴的相对位置(位移)	Position relative
SP	设定速度	Speed
ST	停止指定轴运动(以减速度停止)	Stop

*重复列表 (System Configuration)

Contour Mode

CD	指定轨迹资料(连续路径之位移增量)	Contour data
	CM	Contour mode
DT	设定取样时间间隔(轨迹模式)	Contour time interval
WC	等待轨迹资料	Wait for contour data

ECAM/Gearing

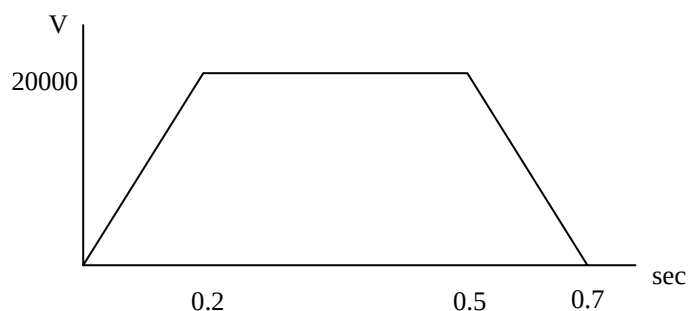
EA	选定凸轮运动之主控轴	ECAM master
EB	激活凸轮运动模式	Enable ECAM
EC	设定凸轮原点	ECAM table index
EG	开始凸轮运动	ECAM go
EM	定义受控轴之位移增量(凸轮运动)	ECAM cycle
EP	定义主控轴之起点及增量(凸轮运动)	ECAM interval
EQ	停止凸轮运动	Disengage ECAM
ET	设定受控轴位移函数(凸轮运动)	ECAM table entry
GA	指定主控轴(电子齿轮)	Master axis for gearing
GM	指定龙门运动执行轴	Gantry mode
GR	指定齿速比(电子齿轮)	Gear ratio for gearing

Vector/Linear Interpolation

CA	指定向量运动的相关平面(or系统)	Define vector plane
CR	指定圆弧运动的半径、起始角及角度增量	Circular interpolation move
CS	清除运动序列	Clear motion sequence
ES	设定椭圆轴调整比例	Ellipse scaling
LE	结束直线轨迹(or补间)运动	Linear interpolation end
LI	指定直线轨迹(补间)各轴的位移	Linear interpolation segment
LM	指定直线轨(补间)迹模式相关各轴	Linear interpolation mode
ST	停止运动	Stop motion
TN	定义正切轴(跟随刀)角度参数	Tangent
VA	设定向量加速度	Vector acceleration
VD	设定向量减速度	Vector deceleration
VE	结束向量运动序列	Vector sequence end
VM	指定向量运动的平面两轴(及正切轴)	Coordinated motion mode
VP	定义向量位移	Vector position
VR	设定向量速度的调整乘数	Vector speed ratio
VS	指定向量速度	Vector speed
VT	设定向量时间平滑常数(平稳运动)	Smoothing time constant —vector

第六章 程序范例

实例一 梯型速度曲线位移



X轴由速度等于0,在加速度100,000单位/秒²下,加速至定速20,000单位/秒² , 然后再以减速度100,000单位/秒²减速至0,总共位移10,000单位

<u>指 令</u>	<u>说 明</u>
PR10000	位移 (PR: Relative Position)
SP20000	速度 (SP: Speed)
DC100000	减速度 (DC: Deceleration)
AC100000	加速度(AC: Acceleration)
BG X	开始运动(BG: Begin)

实例二 多轴运动

4轴独立运动、分别设定位移、速度及加减速

<u>指 令</u>	<u>说 明</u>
PR 500,1000,600,-400	X,Y,Z,W之位移
SP 10000,12000,20000,10000	X,Y,Z,W定速
AC 100000,10000,100000,100000	X,Y,Z,W加速度
DC 80000,40000,30000,50000	X,Y,Z,W减速度
BG XZ	开始X,Z轴之运动
BG YW	开始Y,W轴之运动

实例三 独立运动

运动参数分别设定

<u>指 令</u>	<u>说 明</u>
PR,300,-600	Y,Z轴之位移(逗号前无数值,X轴位移未设定*)
SP,2000	Y轴定速
DC,80000	Y轴减速度
AC,100000	Y轴加速度
SP,40000	Z轴定速(二逗号前无数值,X.Y轴之定速未设定)
AC,,100000	Z轴加速度
DC,,150000	Z轴减速度
BG Z	开始Z轴运动
BG Y	开始Y轴运动

* 若先前已设定X轴位移值, 则维持原值, 若先前未设定, 则X轴位移值为0

实例四 位置及误差查询

4-1 以TP(Tell position)查询位置(位移)

<u>指 令</u>	<u>说 明</u>
TP	查询所有轴之位置
TP X	查询X轴之位置
TP Y	查询Y轴之位置

TP Z	查询Z轴之位置
TP W	查询W轴之位置

4-2 以TE(Tell error)查询指令要求之位移与实际位移间之误差值

<u>指 令</u>	<u>说 明</u>
TE	查询所有轴之误差
TE X	查询X轴之误差
TE Y	查询Y轴之误差
TE Z	查询Z轴之误差
TE W	查询W轴之误差

实例五 绝对位移

<u>指 令</u>	<u>说 明</u>
DP 0,2000	定义X轴之现在位置为0 定义Y轴之现在位置为2000(Define Position)
PA 7000,4000	设定绝对位移
BG X	开始X轴运动
BG Y	开始Y轴运动
X.Y轴运动结束后,返回0之位置	
PA 0,0	移向0,0
BG XY	开始X.Y轴运动

实例六 速度控制

驱动X,Y轴马达到达定速

<u>指 令</u>	<u>说 明</u>
JG 10000,-20000	设定速度及方向
AC 100000,40000	设定加速渡
DC 50000,50000	设定减速渡
BG XY	开始运动
几分钟后	
JG -40000	新的X轴速度及方向
TV X	查询(由控制器回传)X轴速度(Tell Velocity)
然后	
JG,20000	新的Y轴速度及方向
TV Y	查询Y轴速度及方向

停止运动

ST

停止运动(STOP)

实例七 使用编辑器

ED(Edit mode)指令开启控制器之编辑模式,可编辑并储存运动程序

以ED指令进入编辑模式,编辑器自动提供行号,以<ctrl>+Q跳出编辑模式

<u>行 号</u>	<u>指 令</u>	<u>说 明</u>
000	#A	定义旗标
001	PR700	位移
002	SP2000	速度
003	BG X	开始X轴运动
004	EN	结束程序(End program)

以XQ#A 执行程序(XQ, Execute program)

实例八 程序循环

运动程序可包含条件流程

<u>指 令</u>	<u>说 明</u>
#A	旗标
DP 0	定义现在位置为0
V1=1000	设定V1初始值
#Loop	循环旗标
PA V1	X轴移动V1单位
BG X	开始X轴运动
AM X	X轴运动完成后(AM, After motion complete)
WT 500	等500毫秒
TP X	回传X轴位置
V1=V1+1000	增加V1值
JP#Loop,V1<10001	当V1<10001时执行循环
EN	结束(EN, End program)

以<ctrl>+Q 结束编辑

以XQ#A执行#A程序

实例九 程序中之触发点

AD(After distance)、**AP(After absolute position)**均为本例之触发点

<u>指 令</u>	<u>说 明</u>
#B	旗标
DP 0,0	定义初始位置

PR 30000,60000	设定目标
SP 5000,5000	设定目标
BG X	开始X运动
AD 4000	等X轴移动4000单位后
BG Y	开始Y轴运动
AP 6000	当X轴位置为6000时
SP 2000,50000	改变速度
AP ,50000	等Y轴位置为50000时
SP ,10000	改变Y轴速度
EN	程序结束(EN, End program)
以EQ#B执行本程序	

实例十 控制变量

本例X轴移动4000单位,再反向移回2000单位,1000单位,500单位(逐次移回,移至所在位置与零点之中点)直到回到零点,其中__TPX传回X轴位置,是在TP指令前加"_"(下横线)所形成.

<u>指 令</u>	<u>说 明</u>
#A:DP0	旗标:定义现在位置
PR 4000	设定首次位移数值
SP 2000	设定速度
BG X	移动X轴
AMX	X轴运动完成后
WT 500	开始X轴运动500毫秒后
#B	旗标
V1=_TPX	将X轴位置存入V1变量
PR-V1/2	移动- V1/2
BG X	开始X轴运动
AMX	X轴运动完成后
WT 500	500毫秒/分
V1=	传回V1数值
JP#C,V1=0	若V1=0,跳到#C,结束本程序
JP#B	若V1≠0,执行循环#B
#C	旗标
EN	结束程序
以XQ#A执行本程序	

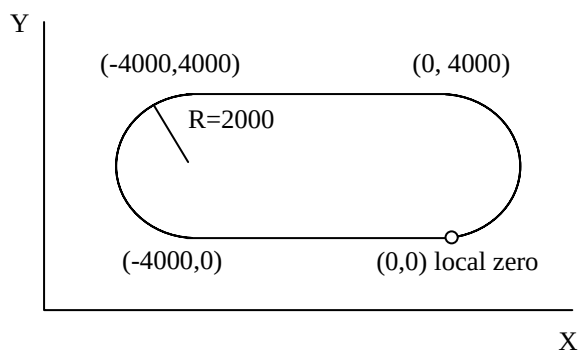
实例十一 直线补间

同时激活并停止马达,沿着直线轨迹移动7000,3000,6000单位

<u>指 令</u>	<u>说 明</u>
------------	------------

LM XYZ	设定XYZ轴补间 (LM, Linear interpolation mode)
LI 7000,3000,6000	补间相移值(LI, Linear interpolation distances)
LE	结束补间设定(Linear interpolation end)
VS 6000	向量速度(Vector speed)
VA 20000	向量加速度(Vector acceleration)
VD 20000	向量减速度(Vector deceleration)
BGS	开始运动(Begin motion + sequence)

实例十二 圆弧补间



定义XY轴向量运动的启始点位置为原点位置

<u>指 令</u>	<u>说 明</u>
VM XY	定义XY运动平面
VP -4000,0	线段
CR 2000,270,-180	圆弧 (r=2000, 由270度起, 转 -180度)
VP 0,4000	线段
CR 2000,90,-180	圆弧
VS 1000	向量速度
VA 50000	向量加速度
VD 50000	向量减速度
VE	结束定义
BGS	开始执行

附 录

电气规格说明

步进马达

脉冲 - TTL(0-5 Volts)level at 50% duty cycle
3,000,000 pulses/sec maximum frequency
方向 - TTL(0-5 Volts)

输入/输出

极限,原点,等其它输入 - TTL
从OUT(1)至OUT(8)输出 - TTL

电源

+5V	750mA
+12V	40mA
-12V	40mA