

目 录

DMC110A单轴运动控制器.....	2
第一部分 概述	2
第二部分 控制器使用说明	3
2.1 硬件说明.....	3
2.2 操作说明.....	6
2.3 软件说明.....	9
2.4 指令编码表.....	18
第三部分 Motion110A使用说明	21
3.1 关于软件名称 (Motion110A)	21
3.2 安装到本地.....	21
3.3 Motion110A 的编辑环境使用说明.....	21
3.4 编程操作.....	21
第四部分 编程实例	26
4.1 PC编程实例.....	26
4.2 控制器手动代码编程输入.....	26
附 录.....	29
1、 RS232 简单无硬件握手通信线缆制作	29
2、 名词解释.....	29
3、 常见问题解答.....	31

DMC110A 单轴运动控制器

第一部分 概述

非常感谢您使用 DMC110A 步进电机运动控制器。该款控制器是科瑞特自动化 DMC 系列运动控制器产品之一，采用高性能的“MCU + FPGA”控制，配合中文屏幕液晶显示，系统功能强大、外观大方，可完全替代“PLC + 文本显示”，是开发高性能数控设备的首选。

该控制器和国内外同类高档控制器相比，先进的特点如下：

1. 用户编程方便。

DMC110A 控制器指令设置合理并简单，符合人们的思维习惯，不会在指令的熟悉上花费您太多的宝贵时间；该控制器提供独立的编程环境，您可以随时对程序进行修改或重写；

2. 控制单轴步进电机。

DMC110A 控制器驱动单轴步进电机，带有两个硬件限位点；

3. 高速、高精度

控制普通步进电机，最高速度可达 5000 转/分钟，突破传统步进电机低速的概念；理想梯形加速曲线，实现平稳快速起停，定位误差为零；

4. 128×64 点阵的液晶显示。

DMC110A 控制器带有 128×64 点阵的液晶屏，显示内容可随时通过控制器自带的 RS232 接口由计算机下载；显示方式可以灵活编程，使用简单；

5. 通用 16 个输入、8 个输出点，强大的逻辑控制功能。

DMC110A 控制器除了一套完整的步进电机控制口外，另有 16 个通用输入点（包括：两个限位点，一个运行、一个停止外输入点）8 个通用输出点，功能完全可以自定义，帮你实现任意简单或复杂的逻辑、点位控制；

6. 支持手动指令代码编程和计算机编程下载。

DMC110A 控制器支持手动编程和计算机编程下载两种编程模式：手动编程界面友好、操作方便；计算机编程有单独的编译软件，并可帮你自动找错，使用简单。

第二部分 控制器使用说明



图 2-1 DMC110A 面板

2.1 硬件说明

2.1.1 功能特点

- 适用于单轴步进电机的各种场合控制应用；
- 控制轴数：单轴；
- 中文点阵液晶显示（128×64 点阵）和 21 键薄膜开关；
- 坐标参数支持相对坐标和绝对坐标；
- 独有立即数和寄存器两种寻址指令；
- 提供运算指令，可进行复杂控制；
- 16 个通用输入点(包括两个硬件限位点)、8 个输出点，实现各种复杂的逻辑、点位控制；
- 简单方便的键盘编程及上位机编程两种方式；
- 50 多条指令组成完备的指令空间，帮您完成任意功能。

2.1.2 性能指标

- 输出脉冲频率：1 ~ 100KHz/轴；
- 控制步进电机转速最高达到 5000 转/分钟；
- 12KB 用户程序空间，704 条用户指令程序；
- 4KB 用户参数存储空间，1024 个参数，参数允许范围为：±999999999。

2.1.3 电气特性

- 工作电源：DC 24V；
- 输入检测口：24V 开关信号（16 路低电平有效）；

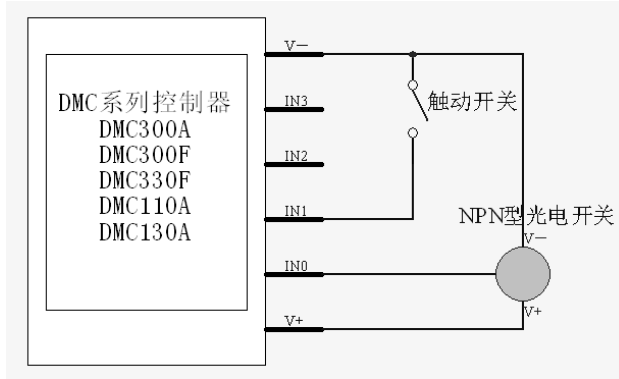


图 2-2 输入口接线方式与原理

- 输出控制口：150mA/24V (8 路常开)。

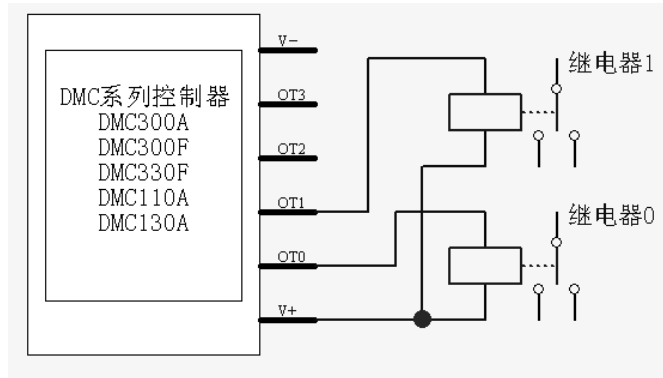


图 2-3 输出口接线方式及原理

2.1.4 控制器接线示意



图 2-4 DMC110A 背部视图

说明：工作电源 V+,V-:DC24V(外接输入电源注意正负极)；

PU：脉冲控制信号；

DR：方向控制信号；

OT0 ~ OT7：8 个点位输出信号；

IN0 ~ IN11：12 个通用输入信号；

RUN(IN14)、STOP(IN15)：运行、停止外接输入；
 L+(IN12)、L-(IN13)：步进电机前后限位输入；
 V+(24V)、V-(GND)：24V 电源和地(内部提供)； RS232: PC 通信口。

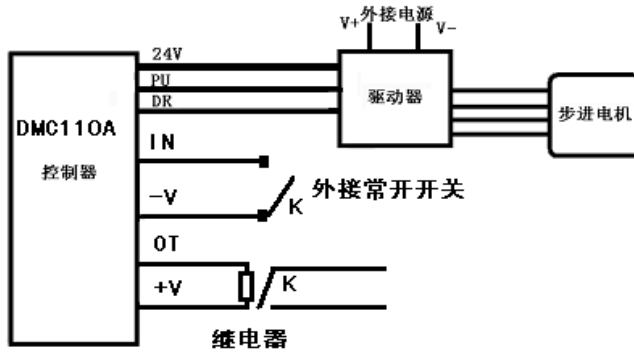


图 2-5 DMC110A 接线示意

(注:本控制器脉冲、方向信号内部已做限流处理，故连接驱动器时不用外接限流电阻)

2.1.5 安装尺寸

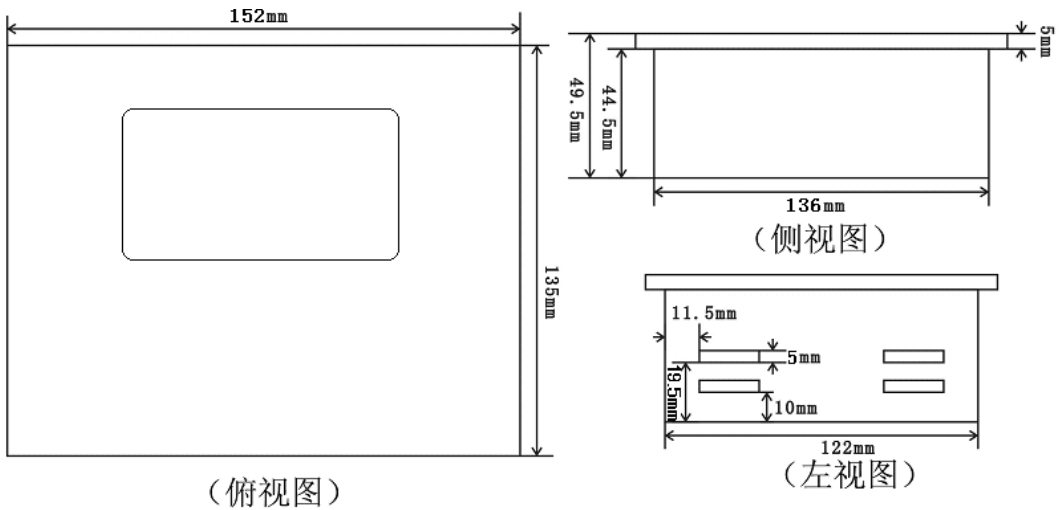


图 2-6 DMC110A 安装尺寸

(注意：在嵌入的工作台面开孔时要预留有虚位即长宽各扩宽 2~3mm，否则会装不下)

DMC110A

2.2 操作说明

2.2.1 键盘功能及显示说明

2.2.1.1 按键构成 (21 个键)

10 个数字键 (0~9), 模式选择“Mode”, 参数键“Par”, 两个光标移动键“←”“→”, 小数点键“.”, 负号键“-”, 确认键“Enter”; 运行键“Run”、急停键“Stop”、功能键“F1”、“F2”。

特殊按键说明

模式键: 用于“1. 程序编辑”、“2. 参数设置”及“3. 手动”选择; 再次按下该键后返回到待机状态;

参数键: 输入和修改运行程序所对应的 S 寄存器参数, 在 (准) 待机状态下有效, 再次按下后退出该状态;

复位键: “F1”待机情况下该键按下, 系统状态恢复到刚上电时的状态;

小数点: 浮点输入时使用;

光标移动键: 参数输入时用于切换到临近的 S 寄存器; 手动状态时用做正反向发脉冲;

确认键: 在参数输入时用于确认当前参数的输入数值, 并转入下一参数的输入;

运行键: 该键按下后运行用户程序;

急停键: 执行用户程序时终止程序运行 (立即响应, 当前状态不保留);

功能键: F1 做复位键, 准待机情况下该键按下, 系统状态恢复待机状态

F1、F2 在参数输入设置时做上下翻页键;

F1、F2 在手动控制时用做速度倍乘倍减;

2.2.1.2 显示模式

DMC110A 控制器的显示功能主要有以下几种:

1). 欢迎界面

界面内容自行绘制并下载。用户可以使用 Windows 自带的画图工具, 绘制任意自己喜欢的 128*64 像素黑白位图 (内容可以是文字也可以是图形); 保存时要顺时针旋转 90 度 (详细操作请参见“第三部分 Motion110A 使用说明”); 参数界面和显示状态界面的制作相同。

2). 8 屏参数界面 (可视参数界面)

界面内容自行绘制并下载。支持 32 (8*4) 个参数; 规则如下:

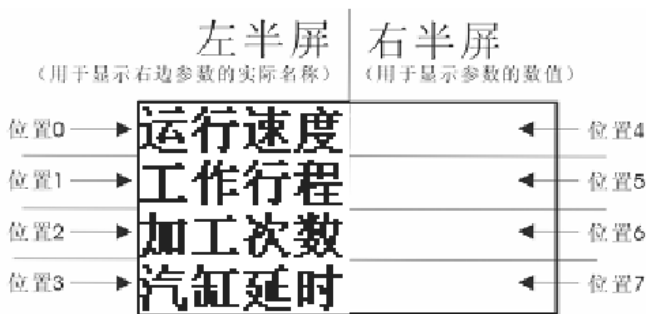


图 2- 7128*64 像素黑白位图参数界面视图示例

- 左半屏用于显示参数的实际名称 (文字提示), 右半屏用于显示并设置参数数值 (4 个 S 寄存器);
- 每个界面 (128*64 像素) 有四个编辑参数位置, 如上图所示的位置 4~7;
- 只需在参数界面下载时设定 4 个 S 寄存器序号, 系统则自动完成半屏文字标识与 S 型变

量（数值）的对应关系；

- 为了视觉效果好，建议四个位置平均分配（64*16 像素）；
- 每个参数支持范围为：-999999999 ~ +999999999；

（具体操作细节请参见 Motion110A 软件说明）。

3). 4 屏状态界面（运行状态信息显示界面）

界面可编制下载，可细分为 32(4*8)个界面或 4 屏内容，并有显示指令支持，可指定内容显示于指定位置。规则如下：

0	4
1	5
2	6
3	7

（状态界面区域分布示意图）

- 每个页面（128*64 像素）平分 8 个单元内容（64*16 像素）；
- 显示单元分布按“从上到下，从左到右”的顺序排列：第一个界面（界面 0,指令表述为 DISPLAY 0,0,0）分为 0-7 单元，第二个界面（界面 1,指令表述为 DISPLAY 0,1,0）分为 8-15 单元.....第 4 个界面（界面 3,指令表述为 DISPLAY 0,3,0）分为 24-31 单元；
- 位置分布同显示单元分布：共可细分为 8 个位置（0-7）；
（使用说明请参见后面的指令详解）

2.2.2 系统了解

DMC110A 控制器系统是基于高性能 MCU 设计的功能强大、用户二次开发简单易用、修改方便、适用范围广的通用型步进电机运动控制系统。该系统共有六类、五十余条指令，程序设计灵活，并有精确的运动指令和显示指令，能充分满足用户的项目要求。针对不同的项目要求，用户不必掌握全部指令，只需能够运用其中的十余条指令甚至几条指令就可圆满实现你的工程任务。

2.2.3 项目分析

要求用户能够根据项目的实际情况，列出控制流程，分析程序量大小、参数更改频繁程度等，以决定选用不同的指令及不同的编程方法。例如：对于程序量小、参数固定的项目编程，尽量选用常数型指令；对于参数经常需要修改的项目编程，可选用变量型指令，参数修改在模式功能中进行；程序量大的项目，在 PC 机上用 Motion110A 软件编程，可降低编程出错，提高编程效率。

（建议用户多在 PC 机上使用软件编程）

2.2.4 程序输入

当用户对项目进行了详细分析，并熟悉了必要的指令或指令代码后，可进行程序编程：

1、PC 编程：在 PC 机上打开 Motion110A 软件编程，指令直接以助记符输入。输入编程后进行编译再下载到控制器。

2、面板指令代码编程：控制器待机状态下，按下“Mode”键，出现操作提示后，再按下‘1’：选择“1. 程序编辑”；出现“输入密码”后，正确输入程序密码后（密码为“070805”）进入程序编程状态，用户可参照代码程序清单进行程序输入。输入完成后按“编程”键退出。

2.2.5 参数输入

密码保护模式 S 寄存器参数输入：控制器（准）待机状态下，按下“Mode”键，出现操作提示后，再按下“2”：选择“2. 参数设置”；出现“输入密码”字样后，输入参数密码（密码为“200708”）后进入参数设置状态（参数设置不分先后）；输入寄存器标号（0000~1023）如：S10 输入“0010”，输入参数数值如：“1000”并回车，进入下一个参数设置，参数设置完毕，按“Mode”键退出参数设置状态。

可视参数输入：将 128*64 像素黑白位图参数界面（文字标注、对应 S 寄存器位号）下载到控制器后，按下“参数”键后按‘↑’‘↓’光标移动键至“参数页面”，输入相对应的参数页序号（0~7），“Enter”确认后移动光标或按“F4”“F5”上下整屏翻页将光标移至要修改的参数位置，输入参数数值并按“Enter”键确认；输入设置完毕后按“参数”键退出。

2.2.6 待机状态及准待机状态

控制器刚上电后的状态为待机状态：

- A) 所有的 M 型变量、B 型位变量清零；
- B) 所有输出口无输出（无效），计数器值、当前坐标为零；
- C) 直接响应用户的其他操作；

控制器执行有效动作后返回的状态为准待机状态：

- A) 所有的 M 型变量、B 型位变量保留之前值；
- B) 所有输出口无输出（无效），计数器值、当前坐标保留之前值；
- C) 直接响应用户的其他操作；

2.2.7 自动运行（按下“运行”键或通用输入“Run”与“V-”短接）

检查系统接线无误后，控制器上电；待程序及参数输入完毕后，按“运行”键（或通用输入“Run”与“V-”短接），进入程序运行状态，系统将自动运行用户程序程序，直至遇到“END”指令后返回至准待机状态。

注：如果程序复杂，并且用户程序中有逻辑错误，运行结果可能出乎你的预料。

2.2.8 手动调整

控制器系统（准）待机状态下，按下“Mode”键，出现操作提示后，再按下‘3’：选择“3. 手动”，进入手动状态；按“←”“→”键可以对步进电机进行正反转控制；按“F1”“F2”进行手动速度倍乘倍减。此状态，系统实时显示当前坐标；

再次按下“Mode”键退出手动返回至准待机状态；

2.2.9 手动参数设置

控制器待机状态下，按下“Mode”键，出现操作提示后，再按下‘2’：选择“2. 参数设置”；出现“输入密码”字样后，输入正确的参数密码（密码为“168888”）后进入系统参数设置状态（参数设置不分先后）；可以对系统参数（SY0、SY1、SY2、SY3、SY4、SY5、SY6、SY20 等）进行设置：

- S0---起始速度（脉冲数：1~20000 典型值：1000）；
- S1---加速时间（毫秒：100~3000 典型值：400）；
- S2---最高速度（脉冲数：1~100000 典型值：10000）；
- S3---点动脉冲（脉冲数：1~20000 典型值：10）；
- S4---手动时按键不松开所发出最大脉冲数（典型值：90000）；
- S5---手动时按键松开后判断需减速的临界速度（典型值：2000）；
- S6---设定浮点型 S 变量数量；（默认为零）

S20---设置输入参数页面显示屏数(0~8屏)(该设置为限定输入参数显示屏数,根据实际使用的输入参数数量而定)。(默认为3)

移动光标可以切换所要设置的参数;再次按下“Mode”键退出手动参数设置;(注:输入或更改参数数据后要按“Enter”键确认)

2.2.10 第一原点设置

在手动调整状态下,将系统调整至机械原点(在限位点即第一原点位置);(按下“Mode”键)退出手动状态,按下‘F1’,即设定当前位置为第一原点(坐标清零);

2.2.11 下载程序

在此状态下,用户可以进行上位机的下载实现;请参见“第三部分 Motion110A 使用说明”。

注意:与控制器下载通讯时,应先确保控制器处于(准)待机状态,并直接在PC机上进行【连接控制器】或【断开连接】即可。

2.2.12 急停、复位

程序“自动”运行时,按下“Stop”键,程序退出,所有运动停止,进入准待机状态;在准待机状态下按下“F1”键,系统转入待机状态。

2.3 软件说明

2.3.1 变量说明及参数约定

DMC110A 控制器系统提供给用户:

- **1024 个 S 型变量 (S0 ~ S1023)**
(用户程序运行中只可读,不可写及程序修改,参见参数更改),代号为 S0 ~ S1023 (注:手动编程时标号为 1000 ~ 2023);
- **16 个 M 型变量(M0-M15):**
标号 M0 - M15(注:手动编程时标号为 0 ~ 15 共计 16 个),用户程序运行中可读可写,可程序修改,但只能在用户程序中赋值;
- **坐标变量 (M100):**
当前绝对坐标,这个变量可读、可清零,即:可以进行判断,执行跳转指令;可以被“SETC”及“CLR”指令清零,但不能在其他运算指令中作为被操作数进行修改(注:手动编程时标号为 100);
- **100 个位 B 变量 (B0 ~ B99)**
 - 0 ~ 15 (B0 ~ B15)** 为通用 B 型(位)变量,程序运行时可读可写可以用作标志位;
 - 16 ~ 23 (I0 ~ I7)** 为输入口(位)变量,可读出输入口状态,对其写无意义;
 - 32 ~ 39 (O0 ~ O7)** 为输出口(位)变量,对其写可进行开关量输出,读无意义。
 - 48 ~ 58 (J0 ~ J10)** 为键盘输入(位)变量(J0 ~ J9 对应 0 ~ 9 键),可对其进行输入有效检测,对其写无意义,“CLRB J10”为清键值指令;
 - 96 (B96)** 为运动标志,当步进电机正在运动时有效(有效 1),当步进电机运动停止时无效(无效 0),可读,写无意义;

说明:因为 DMC110A 控制器操作面板上没有字母键,为了能使用户输入不同类型的数、变量, S 型、M 型变量及 B 型、I 型、O 型变量控制器内部作了统一编码,当你在控制器上直接编程,需要输入统一编码的序号。

例如: 指令(DRVIM) 0115 0,1000第二参数为 S0;

指令 (DRVIM)	0115	0, 1第二参数为M1;
指令 (JB)	0302	12, 5第二参数为B5;
指令 (JB)	0302	12, 16第二参数为I0;
指令 (CLRB)	0406	35第一参数为O3;

(指令含义请参照指令详解。)

特别注意!!!：本文的指令描述中：

- COM 表示指令名称，使用上位 PC 机 Motion110A 编译软件编程时表示指令标示符，控制器键盘编程时表示其指令码；
- #data1,#data2、#data3、#data4 等表示常数；
- S#1,S#2 表示变量（或表示为寄存器），若无特别说明，可以是 M 或 S 型变量,M#1 一般表示为 M 型变量；
- B#1, B#2 表示位变量（或表示为位寄存器），若无特别说明，可以是通用位变量（B0 ~ B15），输入口位变量（I0 ~ I7），输出口位变量（O0 ~ O7），键盘数字键位变量（J0 ~ J10）；
- 其它符号参照指令中的说明；

2.3.2 指令详解

2.3.2.1 运动类

1).单轴运动 格式：COM #data1 / COM S#1

#data1、S#1: 单轴运动数值，S#1 可以为 M 或 S 型变量；

指令：DRVAD /0112	绝对地址/常数值单轴运动
DRVAM /0113	绝对地址/变量值单轴运动
DRVID /0114	相对地址/常数值单轴运动
DRVIM /0115	相对地址/变量值单轴运动

说明：指令执行时，受控目标将沿单轴方向，移动#data(S#1)个脉冲数或移动到#data(S#1)位置（相对原点）。

例：单轴运动，当前坐标 1000，S130=3000。

指令：“DRVAD 3000” 执行结果：系统从当前点，平移到 3000 点，如图 2-8-A；

指令：“DRVAM S130” 执行结果：系统从当前点，平移到 3000 点，如图 2-8-A；

指令：“DRVID 3000” 执行结果：系统从当前点，平移到 4000 点，如图 2-8-B；

指令：“DRVIM S130” 执行结果：系统从当前点，平移到 4000 点，如图 2-8-B；



图 2-8

相对地址与绝对地址的判定：(参考附录)

2).运动停止 格式：COM

DMC110A

指令：STOP / 0116 中止所指定的运动
说明：指令执行后，控制器关断步进电机的运动脉冲。
例：指令：“STOP” 执行结果：电机停止动作；

2.3.2.1 命令控制类

1).速度设置 格式：**COM #data1, #data2, #data3 / COM S#1, S#2, S#3**
 #data1/S#1：最低速度； #data2/S#2：加速时间；#data3/S#3：最高速度；
 S#1,S#2,S#3 可以为 M 或 S 型变量；

指令：SPEED / 0200 常数格式速度参数设置
 SPEEDM / 0201 变量值格式速度参数设置

- 说明：**1. 该指令对其后的运动有效；
 2. 速度曲线为标准梯形加速，最高速度需大于最低速度；
 3. 速度参数单位为脉冲频率，最低速度>1,最高速度<100000；
 4. 加速时间单位为毫秒，加速时间越短，其速度曲线越陡峭，见图 2-9；
 5. 参数必须同为常数或者变量形式。

例：S140=800,S141=500,S142=4000 (见图 2-9)

指令：“SPEED 800,500,4000” 执行结果 :起始速度为 800 赫 ,加速时间为 500 毫秒, 最高速度为 4000 赫；

指令：“SPEEDM S140,S141,S142” 执行结果:起始速度为 800 赫，加速时间为 500 毫秒, 最高速度为 4000 赫；

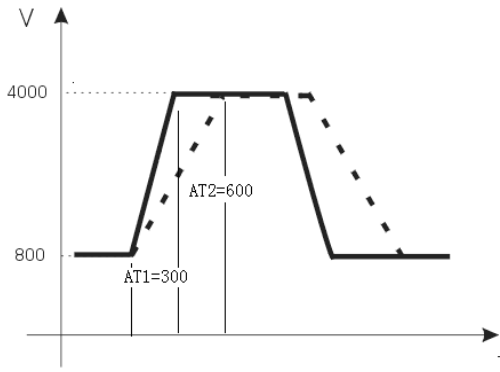


图 2-9 加速曲线示意图

2).暂停等待 格式：**COM**

指令：PAUSE / 0204
说明：指令执行后，系统直到当前运动执行结束，再执行下一条指令。
例：指令：“PAUSE” 执行结果：等待运动停止，然后执行下一条指令，如果没有运动，则直接执行下一条程序；

3).延时 格式：**COM #data / COM S#1**

#data 或 S#1: 延时时间，S#1 可以为 M 或 S 型变量；

指令：DELAY / 0206 固定时间延时
 DELAYM / 0207 可变时间延时

说明：指令执行后，系统暂停 M1 毫秒的时间，“Delay 0”用作空指令(注:1 秒=1000 毫秒)；
例： S150=3000；

指令：“DELAY 200” 执行结果：延时 200 毫秒；
 指令：“DELAYM S150” 执行结果：延时 3 秒；

4).程序结束 格式：COM

指令：END / 0208

说明：程序结束指令，用来标示用户程序结束，用户编程必需。

例：指令：“END” 执行结果：当系统执行到该条指令后，程序停止执行，返回至准待机状态；

5).显示指令 格式：COM #data1, #data2, #data3 / COM #data1, S#2, #data3

#data1：显示格式；#data2 或 S#2：显示内容；#data3：显示位置。

指令：DISPLAY / 0210

说明：调用显示指令，用于显示程序运行中用户想要看到的信息。【阅读本条指令，请先熟悉“键盘功能及显示说明”中的显示说明部分。】

显示格式：“0”显示整屏(DISPLAY 0,X,0)；可以整屏显示 4 屏 (X=0~3)；

“1”固定格式显示坐标值(DISPLAY 1,0,0)；

“2”固定格式显示加工(完成)次数(DISPLAY 2,M#2,0)；加工次数可以设置为 M0~15 中的某个 M 型变量，用户需要在程序的循环中用加一指令来维护；

“4”在液晶屏的指定位置显示 4 个状态界面的哪个单元内容(4*8 个单元之一)(DISPLAY 4, #data2, #data3)；

“6”在液晶屏的指定位置显示哪个 S/M 变量的数值(DISPLAY 6,S#2, #data3)；

“9”清屏(DISPLAY 9,0,0)。

显示内容：当显示格式为 0：指定所要显示 0~3 整屏画面；

当显示格式为 2、6：指定在液晶屏的指定位置所要显示的为哪个 M/S 寄存器:(M0~15、S0~1023)；

当显示格式为 4：指定所要显示的为 4×8 个状态界面中哪个单元(0~31)；

当显示格式为其它：内容为任意值；

显示位置：当显示格式为 4、6：指定所要显示的内容在显示屏中的哪个位置(0~7)。

当显示格式为其它：位置为任意值(建议使用 0)；

例：当前坐标(1234)，M10=100；

指令：“DISPLAY 1,0,0” 执行结果：该指令执行后液晶显示效果如下所示：

手 动			
当 前 坐 标			
1	2	3	4

指令：“DISPLAY 2,M10,0” 执行结果：该指令执行后液晶显示效果如下所示：

执 行 次 数			
1	0	0	

(实例请参阅 P15，使用方法基本一致)

6).速度改变 格式：COM #data1, #data2 / COM S#1, S#2

#data1, S#1：加速时间,#data2, S#2：最高速度

指令：CHSPEED / 0212 常数格式速度改变指令
 CHSPEEDM / 0213 变量格式速度改变指令

说明：该类指令适用于运动过程中的速度改变，例如遇到减速点。该指令执行后，所指定的运动类型的运动将以 SA 的加速度改变至 SH。可以加速，也可以减速。

例：当前速度为 8000 赫，S160=500,S161=500;

指令：“CHSPEED 500,500” 执行结果：速度由 8000 赫，在 500ms 内，降为 500 赫，速度与时间曲线图如图 2-11A；

指令：“CHSPEEDM S160,S161” 执行结果：同上；

例：当前速度为 2000 赫/秒，S160=600,S161=8000;

指令：“CHSPEED 600,8000” 执行结果：速度由 2000 赫，在 600ms 内，升为 8000 赫，速度与时间曲线图如图 2-11B；

指令：“CHSPEEDM S160,S161” 执行结果：同上；

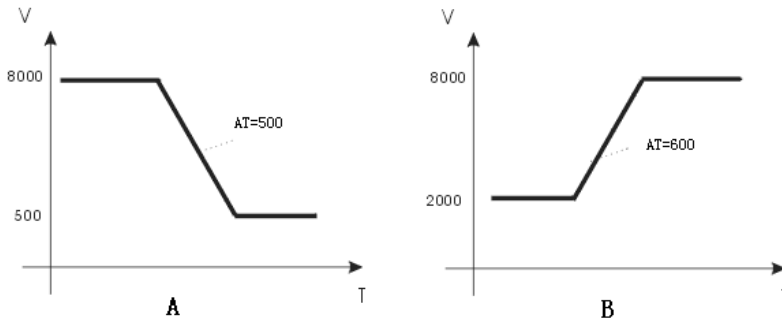


图 2-10 速度改变功能示意

7). 设定零点 格式：COM

指令：SETC / 0214 设置当前位置为零点

说明：该指令执行后，当前坐标值清零。

例：当前坐标 500

指令：“SETC” 执行结果：当前坐标 0

8). 调用子程序 格式：COM #line

指令：CALL / 0216

说明：跳转至#line 行执行程序（自动保留当前程序行号），与“RET”指令配合使用；说明见下一指令。

9). 子程序返回 格式：COM

指令：RET / 0217

说明：返回上次“CALL”调用指令的下一行程序执行；

例：行号 指令

```

.....
20      CALL 100                          ‘跳转到“100”行程序执行
21      MOV M1,10
.....
100     DRVID 0,2000
.....
120     RET                                ‘返回到“21”行程序执行
    
```

2.3.2.1 跳转类

1). 无条件跳转 格式: COM #line / COM M#1

DMC110A

#line:跳转的目的行号, M#1 寄存器指定的行号;

指令: JMP/0300 跳转到指定行
 JMPM/0301 跳转到 M 型变量值指定的行号

说明: 1. 执行该条指令, 程序将跳转#line 行或 M/S 型变量指定的行接着执行。
 2. 可以跳转当前行的前面也可跳转到当前行的后面, 但行号不能大于 704, 不能小于 1 (以下所有跳转指令相同)。
 3. 当程序中有出现多级的程序调用, 即程序嵌套调用时, 子程序返回建议使用“JMPM”;

例: 当前执行程序 100 行, 变量 M10=11,
 指令: “JMP 10” 执行结果: 程序跳转到 10 行执行;
 指令: “JMPM M10” 执行结果: 程序跳转到 11 行执行;

2). 位跳转 格式: COM #line, B#1

B#1: 位变量; #line: 目的行号;

指令: JB / 0302 如果有效跳转;
 JNB / 0303 如果无效跳转

说明: 1. 这两条指令是位跳转指令, 可以检测通用 B 型变量、输入口、键盘数字键 (请参见变量说明);
 2. 通用位变量 (B0-B15) 1 有效, 0 无效; 输入口位变量 (I0-I15) 0 有效, 1 无效;
 3. 条件满足时跳转到指定行运行, 否则执行下一条指令。

例: 指令: “JB 10, B2” 执行结果: 如果 B2 = 1, 跳转到第 10 行程序;
 如果 B2 = 0, 接着执行下一条程序;
 指令: “JNB 10, B2” 执行结果: 如果 B2 = 0, 跳转到第 10 行程序; 如果 B2 = 1, 接着执行下一条程序;
 指令: “JB 15, I1” 执行结果: 如果 I1 输入点有效 (I1 为低电平), 跳转到第 15 行程序; 如果 I1 输入点无效 (I1 为高电平), 接着执行下一条程序;
 指令: “JNB 15, I1” 执行结果: 如果 I1 输入点无效 (I1 为高电平), 跳转到第 15 行程序; 如果 I1 输入点有效 (I1 为低电平), 接着执行下一条程序;

3). 比较跳转 格式: COM #line, S#1, #data / COM #line, S#1, S#2

S#1, #data 或 S#1, S#2: 相比较的两个数 (可以为 M 或 S 变量), #line 跳转的目的行号;

指令: JGD / 0304 变量值与常数值比较, 大于则跳转,
 JGM / 0305 变量值与变量值比较, 大于则跳转,
 JLD / 0306 变量值与常数值比较, 小于则跳转,
 JLM / 0307 变量值与变量值比较, 小于则跳转,
 JED / 0308 变量值与常数值比较, 等于则跳转,
 JEM / 0309 变量值与变量值比较, 等于则跳转,
 JNED / 0310 变量值与常数值比较, 不等于则跳转,
 JNEM / 0311 变量值与变量值比较, 不等于则跳转,

说明: 当相应条件成立时跳转到指定行, 否则执行下一条指令。相比较的两个数必须同为常数或变量

例: M1=550, S100=600,
 指令: “JGD 20, M1, 500” 执行结果: 跳转到 20 行 (条件成立);
 指令: “JGM 30, M1, S100” 执行结果: 转下一条指令 (条件未成立);
 指令: “JLD 20, M1, 500” 执行结果: 转下一条指令 (条件未成立);

指令：“JLM 30,M1,S100”	执行结果：跳转到 30 行（条件成立）；
指令：“JED 20,M1,550”	执行结果：跳转到 30 行（条件成立）；
指令：“JEM 30,M1,S100”	执行结果：转下一条指令（条件未成立）；
指令：“JNED 20,M1,550”	执行结果：转下一条指令（条件未成立）；
指令：“JNEM 30,M1,S100”	执行结果：跳转到 30 行（条件成立）；

4. 循环跳转 格式: COM #line, M#1

循环跳转至#line 行#data/M#1 次；

指令： LOOP / 0312 循环跳转 N 次 (N = #Data)；
 LOOPM / 0313 循环跳转 M#1 次；

- 说明：1. 这两条指令用于建立循环，可以用常数指明次数，也可以用变量型指令以方便次数更改；
2. 当采用变量型指令时，变量为 M 型变量，并且在跳转目的行和当前行之间不能用指令改变该变量的值；
3. LOOP 指令不允许嵌套，即 Loop 指令与目的行之间不能再次出现该指令；如果必须使用嵌套循环，请使用 JMPM 指令实现。

再次强调：跳转的目的行号必须为有效数值：1 ~ 682，否则会出现不可预知的错误

例： M10 = 100 ,

指令：“LOOP 50,20” 执行结果：循环跳转到 50 行 20 次；

指令：“LOOPM 30,M10” 执行结果：循环跳转到 30 行 100 次；

2.3.2.1 逻辑(位操作)

1). 与 格式: COM B#1, B#2

指令：AND / 0400

说明：双操作数都为位变量。B#1 位变量与 B#2 位变量“与”，结果存于 B#1；当 B#1 为输出口位变量时，该指令影响输出口状态；B#1 不能为 I 型或 J 型位变量，B#2 不能为 O 型位变量。

例：当前 B1=1;B2=0, I1=0,O1=1,

指令：“AND B1,B2” 执行结果：B1=B1&B2=1&0=0,

指令：“AND O1,I1” 执行结果：O1=O1&I1=1&0=0,输出口 O1 有效；

2). 或 格式: COM B#1, B#2;

指令：OR / 0402

说明：双操作数都为位变量。B#1 位变量和 B#2 位变量“或”，结果存于 B#1；当 B#1 为输出口位变量时，该指令影响输出口状态；B#1 不能为 I 型或 J 型位变量，B#2 不能为 O 型位变量。

例：当前 B1=0;B2=1, I1=1,O1=0,

指令：“OR B1,B2” 执行结果：B1=B1|B2=0|1=1;

指令：“OR O1,I1” 执行结果：O1=O1|I1=0|1=1,输出口 O1 无效；

3). 非 格式: COM B#1;

指令：CPL / 0404

说明：操作数为位变量。该指令对 B#1 位变量的值取反；当 B#1 为输出口位变量时，该指令改变输出口状态；B#1 不能为 I 型或 J 型位变量。

例：B1=0,O3=1;

指令：“CPL B1” 执行结果：B1 = !B = !0 =1;

指令：“CPL O3” 执行结果：O3= !O3 = !1 = 0,输出口 O3 反转；

4). 置位 格式: COM B#1;

指令：SETB / 0406

DMC110A

说明：操作数为位变量。该指令对 B#1 位变量的值置 1；当 B#1 为输出口位变量时，该指令使输出口有效（低电平）；B#1 不能为 I 型或 J 型位变量。

例：指令：“SETB B1” 执行结果：B1=1；
指令：“SETB O6” 执行结果：O6=0,输出口 O6 有效；

5).清零 格式: COM B#1;

指令：CLRB / 0408

说明：操作数为位变量。该指令对 B#1 位变量的值清零；当 B#1 为输出口位变量时，该指令使输出口无效（高电平）；B#1 不能为 I 型或 J 型位变量。

例：指令：“CLRB B1” 执行结果：B1=0
指令：“CLRB O6” 执行结果：O6=1,输出口 O6 无效；

2.3.2.1 运算类

1).加 格式:COM M#1, S#1;

指令：ADD / 0500 变量加常数值
ADDM / 0501 变量加变量值（M 或 S 型变量）

说明：运算指令的目的操作数必须为 M 型变量（可读可写），原操作数可为 M 或 S 型变量或常数。“加”运算结果：目的操作数加上源操作数，结果存于目的操作数（M 型变量）。

例：M1=200,M2=300,S100=400;
指令：“ADD M1,500” 执行结果：M1=M1+500=200+500=700;
指令：“ADDM M1,M2” 执行结果：M1=M1+M2=200+300=500;
指令：“ADDM M1,S100” 执行结果：M1=M1+S100=200+400=600;

2).减 格式: COM M#1, S#1;

指令：SUB / 0502 变量减常数值
SUBM / 0503 变量减变量值（M 或 S 型变量）

说明：“减”运算结果：目的操作数减去源操作数，结果存于目的操作数 - M 型变量；
目的操作数：M#1 为 M 型变量； 原操作数：M2 可为 M 或 S 型变量。

例：M1=200,M2=300,S100=400;
指令：“SUB M1,500” 执行结果：M1=M1-500=200-500=-300;
指令：“SUBM M1,M2” 执行结果：M1=M1-M2=200-300=-100;
指令：“SUBM M1,S100” 执行结果：M1=M1-S100=200-400=-200;

3).增 1 格式: COM M#1;

指令：INC / 0504 M 型变量 M#1 的值加 1;

说明：“增一”运算结果：M 型变量 M#1 每执行该指令一次，其值增一。多用于循环、计数。

例：当前 M12=10，
指令：“INC M12” 执行结果：M12=M12+1=11;

4).减 1 格式: COM M#1;

指令：DEC / 0506 M 型变量 M#1 的值减 1;

说明：“减一”运算结果：M 型变量 M1 每执行该指令一次，其值减一，多用于循环、计数。

例：当前 M12=10，
指令：“DEC M12” 执行结果：M12=M12-1=9;

5).清零 格式: COM M#1;

指令：CLR / 0508 M 型变量 M#1 值清零;

说明：“清零”运算结果：M 型变量 M#1 的值清零。

例：指令：“CLR M12” 执行结果：M12=0;

6).乘 格式:COM M#1, S#1;

指令：MUL / 0510 M 型变量乘以常数值

MULM / 0511 M 型变量乘以变量值 (M 或 S 型变量)

说明：运算指令的目的操作数必须为 M 型变量 (可读可写)，原操作数可为 M 或 S 型变量或常数 (其它运算指令同)。“乘”运算结果：目的操作数加上源操作数，结果存于目的操作数 (M 型变量)。

例：M1=200,M2=12,S100=13;

指令：“MUL M1,11”	执行结果：M1=M1*11=200*11=2200;
指令：“MULM M1,M2”	执行结果：M1=M1*M2=200*12=2400;
指令：“MULM M1,S100”	执行结果：M1=M1*S100=200*13=2600;

7).除 格式: COM M#1, S#1;

指令：DIV / 0512 M 型变量除以常数值

DIVM / 0513 M 型变量除以变量值 (M 或 S 型变量)

说明：“除”运算结果：目的操作数除以原操作数，结果存于目的操作数；结果有小数的，舍去小数部分。

例：M1=2000,M2=12,S100=13;

指令：“DIV M1,10”	执行结果：M1=M1/10=2000/10=200;
指令：“DIVM M1,M2”	执行结果：M1=M1/M2=2000/12=166;
指令：“DIVM M1,S100”	执行结果：M1=M1/S100=2000/13=153;

2.3.2.1 数据传送类

1).赋值 格式:COM M1, #data

指令：MOV/0600 常数送入变量 M1;
MOV M/0601 变量 M2 的值送入变量 M1;

说明：常数值或变量值送入 M 型变量。

例：M0=101,S100=202,

指令：“MOV M2,100”	执行结果：M2=100；
指令：“MOV M2,M0”	执行结果：M2=M0=101；
指令：“MOV M2,S100”	执行结果：M2=S100=202；

2).指定地址取值 格式:COM M#1, M#2

指令：LP / 0604

说明：以变量 M#2 的值为“统一编码序号”的变量，将其数值送入 M#1 变量中；该指令可以认为是指针操作；

例：M0=2,M3=1200,S200=1890；M4=3；

指令：“LP M0,M3”	执行结果：M0=S200=1890；
“LP M0, M4”	执行结果：M0=M3=1200；

2.3.2.1 DMC110A 浮点功能补充说明

【浮点型 F 变量】

增加 16 个浮点型 F 变量，分别为 F0、F1、F2、……、F15；
(在键盘手动编程时，分别对应为 M200 ~ M215)

【浮点数应用于运动的特殊性】

1. 浮点数不能直接应用于运动指令的脉冲数；
2. 在运动指令前，将浮点数赋值给 M 变量：‘FMOV M10, F0’
3. 浮点型数值赋给 M 型变量时，小数部分略掉：

```
FMOV F0, 2.58
FMOV M1, F0
执行结果：F0=2.58, M1=2；
```

4. 浮点型 S 变量的可靠用法：参加运算前赋值给 F 变量，运动指令前赋值给 M 变量；

```
FMOV    F0, S2
FMUL    F0, 2.35
FMOVM   M1, F0
DRVIM   M1
```

【浮点运算指令】

在兼容以前版本指令的基础上，增加了以下指令：

```
FADD / FADDM /
FSUB / FSUBM /
FMUL / FMULM /
FDIV / FDIVM /
FMOV / VMOV /
FLP /
```

注：浮点运算只适用于 PC 编程下载，不能用手动代码输入且在代码程序中看不到浮点运算内容

2.4 指令编码表

DMC110A 控制器共 50 余条指令，表中采用了下述符号：

- #data:表示为一个数值，范围为-999999999 ~ 999999999；
- S#1, S#2, S#3, S#4：若无特别说明，表示为某个 M 或 S 型变量；
- M#1, M#2：表示为某个 M 型变量；
- #line：表示为某一行程序的行号；
- B#1、B#2：表示为某个 B 型位变量。

(一) 运动类指令

助记符	代码	指令参数	说明
DRVAD	0112	#data	运动到指定坐标
DRVAM	0113	#data	同上，但坐标来自于 S 或 M 变量中
DRVID	0114	#data	运动若干数值
DRVIM	0115	S#1	同上，但数值来自于 S 或 M 变量中
STOP	0116		停止运动

(二) 命令控制类指令

助记符	代码	指令参数	说明
SPEED	0200	#data1,#deta2,#deta3	设定某个轴、某类运动的起始、最高速度
SPEEDM	0201	S#1,S#2,S#3	同上，但速度值来自于 S 或 M 变量中
PAUSE	0204		等待某个轴、某类运动的脉冲发完
DELAY	0206	#data1	延时指令，延时多少时间，毫秒为单位
DELAYM	0207	S#1	同上，但时间值来自于 S 或 M 变量中
END	0208		程序结束
DISPLAY	0210	#data1,#data2,#data3	显示指令
CHSPEED	0212	#data1,#deta2,	在设定的时间内速度改变为设定值

CHSPEEDM	0213	S#1,S#2	同上，但速度值来自于 S 或 M 变量中
SETC	0214		设定原点，即坐标值清零
CALL	0216	#line	调用某行的程序
RET	0217		返回 CALL 指令调用的下一行

(三) 跳转类指令

助记符	代码	指令参数	说明
JMP	0300	#line	跳转到某行程序执行
JMPM	0301	#M	同上，但行号取自于 M 变量中
JB	0302	#line, #B	某个位变量有效的话，跳转到某一行
JNB	0303	#line, #B	某个位变量无效的话，跳转到某一行
JGD	0304	#line, S#1, #data1	如果指定的 M 或 S 变量大于某个数值的话，跳转到某一行
JGM	0305	#line, S#1, S#2	如果指定的 M、S 变量大于另一个 M、S 变量跳转到某一行
JLD	0306	#line, S#1, #data1	如果指定的 M 或 S 变量小于某个数值的话，跳转到某一行
JLM	0307	#line, S#1, S#2	如果指定的 M、S 变量小于另一个 M、S 变量跳转到某一行
JED	0308	#line, S#1, #data1	如果指定的 M 或 S 变量等于某个数值的话，跳转到某一行
JEM	0309	#line, S#1, S#2	如果指定的 M、S 变量等于另一个 M、S 变量跳转到某一行
JNED	0310	#line, S#1, #data1	如果指定的 M 或 S 变量不等于某个数值的话，跳转到某一行
JNEM	0311	#line, S#1, S#2	如果指定的 M、S 变量不等于另一个 M、S 变量跳转到某一行
LOOP	0312	#line, #data1	指定跳转到某一行若干次
LOOPM	0313	#line, M#1	同上，但次数存于 M 或 S 变量中

(四) 逻辑类指令 (对位操作)

助记符	代码	指令参数	说明
AND	0400	B#1, B#2	两个位变量进行与，存于第一个位变量
OR	0402	B#1, B#2	两个位变量进行或，存于第一个位变量
CPL	0404	B#1	将某个位变量取反
SETB	0406	B#1	将某个位变量置位
CLRB	0408	B#1	将某个位变量清零

(五) 运算类指令

助记符	代码	指令代码	说明
ADD	0500	M#1, #data	指定的 M 变量加上某一数值
ADDM	0501	M#1, M#2	同上，但数值存于另一个 M 或 S 变量中
SUB	0502	M#1, #data	指定的 M 变量减去某一数值
SUBM	0503	M#1, M#2	同上，但数值存于另一个 M 或 S 变量中
INC	0504	M#1	指定的 M 变量加 1
DEC	0506	M#1	指定的 M 变量减 1
CLR	0508	M#1	清指定的 M 变量

MUL	0510	M#1,#data	指定的 M 变量乘于某一数值
MULM	0511	M#1,M#2	同上，但数值存于另一个 M 或 S 变量中
DIV	0512	M#1,#data	指定的 M 变量除以某一数值
DIVM	0513	M#1,M#2	同上，但数值存于另一个 M 或 S 变量中
FADD	0520	FM#1,#fdata	浮点加
FADDM	0521	FM#1,M#2	浮点加
FSUB	0522	FM#1,#fdata	浮点减
FSUBM	0523	FM#1,M#2	浮点减
FMUL	0530	FM#1,#fdata	浮点乘
FMULM	0531	FM#1,M#2	浮点乘
FDIV	0532	FM#1,#fdata	浮点除
FDIVM	0533	FM#1,M#2	浮点除

(六) 数据传送类指令

助记符	代码	指令代码	说明
MOV	0600	M#1,#data	将指定的 M 变量赋于指定的数值
MOVM	0601	M#1,M#2	同上，但数值存于另一个 M 或 S 变量中
LP	0604	M#1,M#2	将第二 M 变量中位置所代表的 M 或 S 变量的值置给第一个 M 变量
FMOV	0620	FM#1,#fdata	浮点赋值
FMOVM	0621	FM#1,M#2	浮点赋值

第三部分 Motion110A 使用说明

3.1 关于软件名称（Motion110A）

Motion 表明软件是用于运动控制程序编制。110A 表示这一个版本是适用于科瑞特自动化 DMC110A 控制器。科瑞特自动化始终专注于工控领域，今后还将推出其它的运动控制产品。

声明: Motion XXXX 系列软件和 DMC 系列控制产品的版权归深圳市科瑞特自动化技术有限公司所有。任何单位及个人不得对深圳市科瑞特自动化技术有限公司的软硬件产品进行反向工程，反向编译和各种解密。

3.2 安装到本地

Motion110A 为绿色软件，当您获得 Motion110A 程序副本，只需将 Motion110A.exe 复制到任意目录，即可执行。它不会在您的系统添加任何额外信息。

3.3 Motion110A 的编辑环境使用说明

科瑞特自动化 DMC 系列控制器提供了充足的程序指令，用户使用这些指令的组合可以完成丰富的功能。通常对控制器编程有三个步骤：程序输入；程序编译；下载到控制器。另外，有时候还要对控制器进行一些设置，以下逐步说明。

打开 Motion110A.exe，先显示欢迎界面然后进入程序输入窗口。在这里可以输入程序代码，它提供了和一般文本编辑器相似的复制、粘贴、剪切、查找、文件保存等功能。

3.4 编程操作

3.4.1 程序输入

科瑞特自动化 DMC 系列控制器提供了丰富的指令和寻址空间，足以编辑强大的控制程序。编程规则如下：

1. DMC 系列控制器指令系统是由类似汇编的语句及语法组成的完善的指令系统；
 2. 在编辑器中，每行只能书写一条语句，语句中不能换行；
 3. 语句格式为：助记符 + 操作数。首先写指令的助记符，然后紧跟操作数。每个指令最多跟 4 个操作数。指令与操作数，操作数与操作数之间用空格分隔；
 4. 程序必须用 END 指令结束。
 5. 程序注释为以英文单引号 ‘ 打头的字符串。
 6. 编程时，用户可以使用行标号，方法是：一个字符串加冒号；
 7. 程序输入不分大小写；
- （注意：行号标号不能有重复）

见下图所示例子：

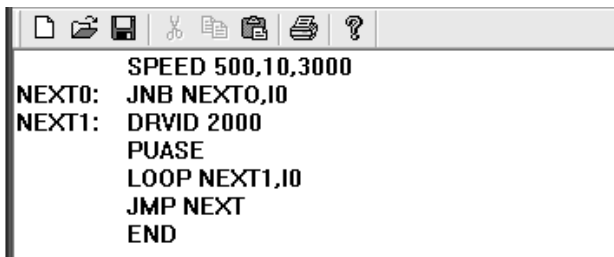


图 2- 11

关于指令的说明和每个操作数的含义，请参阅【指令详解及指令表】。

3.4.2 程序编译

程序源代码写好之后，就可以执行编译了。点菜单的【工具】，然后选择【程序编译】。将弹出对话框，如图 2-13 所示。



图 2- 12 程序编译窗口

点击【浏览】，在弹出的对话框中，选择文件存放路径，并输入文件名；点击【程序编译】即可执行编译。这时，会弹出输出窗口，可以看到编译信息。如果完成正确编译，将提示编译成功；否则会弹出编译出错信息。图 2-14 为编译成功一个程序后，输出窗口的样子。



图 2- 13 程序编译成功后的输出窗口

3.4.3 程序下载

程序编译完毕之后，必须下载到控制器中才能运行。下载之前必须先建立和控制器的通讯连接。

3.3.3.1 通讯接口设置

请先确保你的计算机和控制器之间已经有物理连接。科瑞特自动化 DMC 系列控制器和计算机之间通过标准 RS232 串行通信。其连接线是 DB9S 标准接头，采用简单的无硬件握手协议，连线方式请参阅附录一。

串行通讯必须选择合适的波特率和串口。点击【控制器会话】菜单，选择【串口设置...】，会弹出下载对话框（如图 2-15 所示）。



图 2-14 串口设置对话框

选择与控制器有物理连接的串口，波特率选 9600，点击【确定】，就完成了串口设置。这里要说明的是，务必保证没有其它程序打开选定的串口。

3.3.3.2 建立连接

串口设置完毕之后，就可以开始建立与控制器的通讯了。点击【控制器会话】菜单，选择【连接控制器】，上位机将呼叫控制器。控制器响应后，建立连接，这时可以看到【控制器会话】下拉菜单中，【下载程序】等项使能（有一定的时延，可以重复点击【连接控制器】）。如果【下载程序】等项仍为灰色，并且等待一段时间后，弹出“控制器无应答！”，请检查控制器与 PC 机的 RS232 接口是否连接正常，如果确认接口正常，设置正确，仍无法与控制器建立连接，请与我技术部联系。图 2-16 为与控制器成功建立通讯连接的输出窗口显示内容。

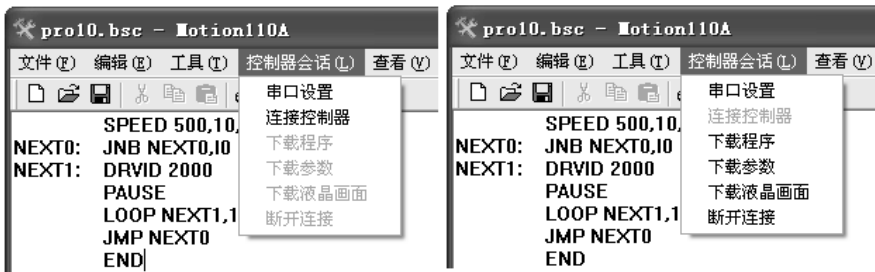


图 2-15 建立通讯连接前后的输出窗口对比

3.3.3.3 下载程序

通讯建立之后，点击【控制器会话】菜单，选择【下载程序】，将弹出程序下载对话框，如图 2-17 所示：



图 2-16 下载程序对话框

点击【浏览...】选择要编译好的程序文件。点击【开始下载】，就可将程序下载到控制器对应的程序序号中。在输出窗口中，仍然可以看到下载时的控制器应答和通讯状态，当下载时，会显示消息窗口提示下载进度，如图 2-18 所示：

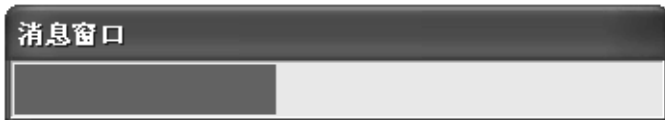


图 2-17 程序下载下载进度输出窗口

如果提示下载不成功，请检查上位机与控制器间的 RS232 连线是否有断线，是否无意使控制器退出下载状态。

3.4.4 下载液晶画面

3.4.4.2 制作图片

要下载的图片要求必须是 128×64 像素的黑白位图。另外，还需要把图片顺时针转 90 度。下面以 windows 自带的画图工具为例说明如下：其他图形绘制软件类似，但建议使用 Windows 自带画图工具。

打开画图程序，新建一副图片；

- 点击【图象】菜单，选择【属性】。在对话框中设置：图象宽=128,高=64；单位：像素；颜色：黑白，点击【确定】。
- 在空白画布上画好图形。
- 点击【图象】菜单，选择【翻转/旋转】。在对话框中选择“按一定角度旋转”，选择“90度”，点确定。
- 存盘退出。

3.4.4.2 下载图片

图片按规定格式制作好之后，就可以下载到控制器了。

点击【控制器会话】菜单，选择【下载液晶画面】，将弹出“下载控制器显示画面”对话框。如图 2-19 所示。



图 2-18 下载液晶画面对话框

A). 下载开机画面。

- 点击【浏览】，选择制作好的开机画面；
- 在“画面类型选择栏”点击选中“开机画面”；
- 在“至控制器画面序号”栏输入“0”；
- 点击“开始下载”；在输出窗口可以看到下载进度消息提示窗口。

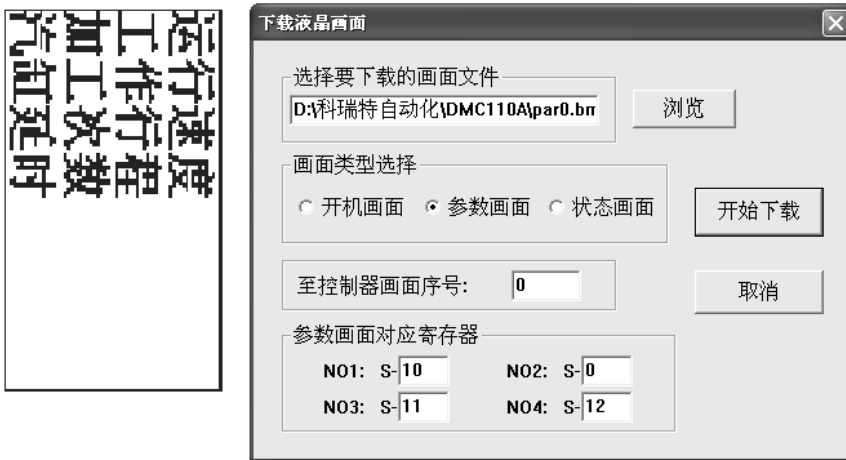
B). 下载可视参数画面。

- 点击【浏览】，选择制作好的参数输入图片；
- 在“画面类型选择栏”点击选中“参数画面”；
- 在“至控制器画面序号”栏输入“0~7”；
- 在“参数画面对应 S 寄存器”栏中四个位置，分别输入四个输入区对应的寄存器；
- 点击“开始下载”；在输出窗口可以看到下载进度消息提示窗口。

C). 下载状态画面

- 点击【浏览】，选择制作好的状态画面图片；
- 在“画面类型选择栏”点击选中“状态画面”；
- 在“至控制器画面序号”栏输入“0~3”；
- 点击“开始下载”；在输出窗口可以看到下载进度消息提示窗口。

例：128*64 黑白位图：“D:\输入参数 1.Bmp”假设：要求“运行速度”对应 S10，“工作行程”对应 S0，“加工次数”对应 S11，“汽缸延时”对应 S12，画面对应参数输入界面 0，下载控制器显示画面对话框基本如下：



A

B

图 2-19 下载可视参数画面参数设置示意

3.4.5 断开连接

与控制器通讯（包括下载程序，下载 S 寄存器，下载液晶画面）完成后，要断开连接。点击【控制器会话】菜单，选择【断开连接】，即可断开与控制器的连接。

DMC110A

第四部分 编程实例

4.1 PC 编程实例

例 1. 一个简单的运动控制系统

运行要求：按下“运行”进入程控状态后系统由 IN0 开关触发沿 X 轴方向以每秒钟 5000 个脉冲的速度反向运行 4000 步，然后返回待机状态。

程序清单：(标号无特别需要可不加)

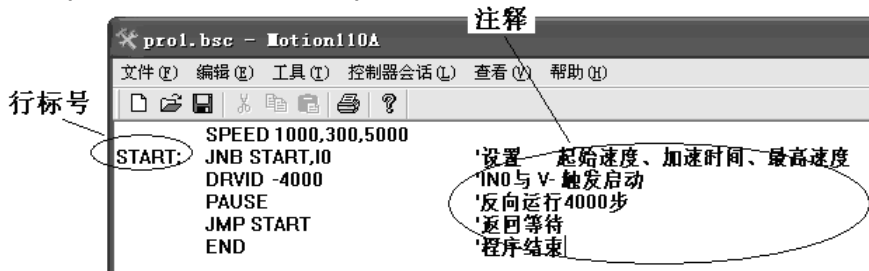


图 2- 20

例 2. 简单的定长剪切机控制系统

分析：普通的电机带动凸轮控制切刀，在凸轮的最高位安装一个感应器开关控制刀位元，步进电机带动压轮送料；输入口 I2 为刀位感应信号，输出口 O2 控制提刀和下刀输出信号；

程序清单 1/使用立即数指令：

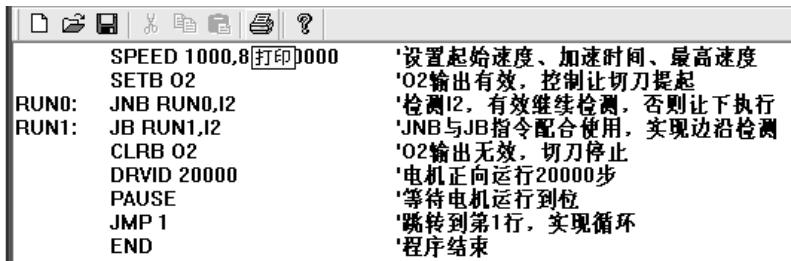


图 2- 21 使用立即数指令程序示例

程序清单 2/使用寄存器指令：

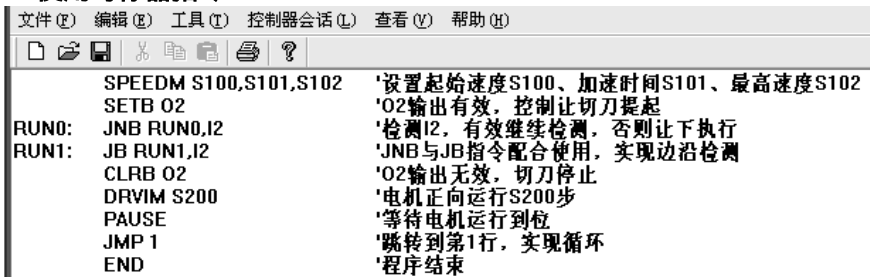


图 2- 22 使用寄存器指令程序示例

例 4. 显示功能及响应键盘按键示范

(请参阅 P33-例 4，使用方法基本相同)

4.2 控制器手动代码编程输入

我们以“编程实例”的例 2 的操作为例，让用户熟悉控制器的手动编程的操作流程。

A.程序清单 1 (使用立即数指令):

指引行	语句/程序指令	对应指令代码
	SPEED 1000,800,20000	0200
	SETB 02	0406
	DELAY 0	0206
RUN:	JNB RUN,I2	0303
	CLRB 02	0408
	DRVID 20000	0114
	PAUSE 0	0204
	JMP 1	0300
	END	0208

- 1) 将 DMC110A 控制器接通电源，系统进入待机状态；
- 2) 按“Mode”键，按“1”选“1.程序编辑”系统显示“输入密码”，正确输入密码（070805），系统显示编程界面(注:在此“←”“→” 移动光标键 “F1”“F2”上下跳程序行)；如果密码输入错误，请重复此过程；
- 3) 输入指令码“0200”，输入第一参数“1000”，按“Enter”确认参数（以下简称“回车”），输入第二参数“800”并回车，输入第三参数“20000”并回车本行程序结束，进入下一行程序；如果参数输错，按“F1”键返回，重复本行程序输入，以下同；
- 4) 输入指令码“0406”，输入第一参数“34”并回车，进入下一行程序；
- 5) 输入指令码“0206”，输入第一参数“0”并回车，进入下一行程序；
- 6) 输入指令码“0303”，输入第一参数“4”并回车，输入第二参数“18”并回车，进入下一行程序；
- 7) 输入指令码“0408”，输入第一参数“34”并回车，进入下一行程序；
- 8) 输入指令码“0114”，输入第一参数“20000”并回车，进入下一行程序；
- 9) 输入指令码“0204”，并回车，进入下一行程序；
- 10) 输入指令码“0300”，输入第一参数“1”并回车，进入下一行程序；
- 11) 输入指令码“0208”，系统记入程序并自动退出编程状态。

注：如果需要对以上程序修改，例如，需要将第六条指令：drvid (0114) 20000 的第一参数改为 30000，不必重新输入全部程序，只需修改第六条指令即可，步骤如下：

- 1) 在系统待机状态下，(按下“复位”键，确认为待机状态)，进入编程状态（步骤同上）；
- 2) a.连续按“F2”五次上翻页键，进入第六行编程（系统有行号提示）；
b.或者向上移动光标至行号位置，输入“0006”，直接进入第六行编程；
- 3) a.输入指令码“0114”，输入第一参数“30000”并回车；
b.或者向下移动光标至第一个参数，输入“30000”并回车；
- 4) 按“Mode”键，退出编程状态，程序修改完毕。

B.程序清单 2 (使用寄存器指令):

指引行	语句/程序指令	对应指令代码
	SPEEDM S100,S101,S102	0201
	SETB 02	0406
	DELAY 0	0206
RUN:	JNB RUN,I2	0303
	CLRB 02	0408
	DRVIM S200	0115
	PAUSE	0204
	JMP 1	0300
	END	0208

- 1) 将 DMC110A 控制器接通电源，系统进入待机状态；
- 2) 进入编程状态：按“Mode”键，选按“1.程序编辑”系统显示“输入密码”，正确输入密码（070805），系统将显示编程接口；如果密码输入错误，请重复此过程；

DMC110A

- 3) 输入指令码“0201”，输入第一参数“1100”并回车，输入第二参数“1101”并回车，输入第三参数“1102”并回车，进入下一行程序；
- 4) 输入指令码“0406”，输入第一参数“34”并回车，进入下一行程序；
- 5) 输入指令码“0206”，输入第一参数“0”并回车，进入下一行程序；
- 6) 输入指令码“0302”，输入第一参数“3”并回车，输入第二参数“18”并回车，进入下一行程序；
- 7) 输入指令码“0408”，输入第一参数“34”并回车，进入下一行程序；
- 8) 输入指令码“0115”，输入第一参数“1200”并回车，进入下一行程序；
- 9) 输入指令码“0204”，输入回车，进入下一行程序；
- 10) 输入指令码“0300”，输入第一参数“1”并回车，进入下一行程序；
- 11) 输入指令码“0208”，编程结束，系统记入程序并自动退出编程状态。
- 12) **参数设置**：按“Mode”键选按“2.参数设置”键，系统显示“输入密码”，输入密码（200708），系统将显示参数设置界面（参数设置不分先后）；
- 13) 输入寄存器标号“0100”，输入“1000”并回车，进入下一个参数设置；
- 14) 输入寄存器标号“0101”，输入“800”并回车，进入下一个参数设置；
- 15) 输入寄存器标号“0102”，输入“20000”并回车，进入下一个参数设置；
- 16) 输入寄存器标号“0200”，输入“2000”并回车，进入下一个参数设置；
- 17) 参数设置完毕，按“Mode”键退出参数设置状态。

注：如果需要对以上程序修改，步骤同清单 1 的修改；

如果需要对寄存器参数修改，例如，需要将第六条指令：drvim (0115) S200 的第二参数 S200 的寄存器值改为 30000，步骤如下：

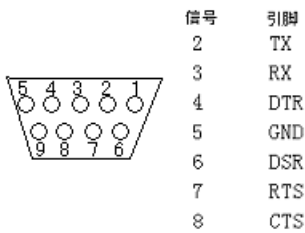
- 1) 在系统（准）待机状态下，（按下“急停”，“F1”复位键，确认为待机状态），按“Mode”键，进入参数设置状态（步骤同上 12）；
- 2) 输入寄存器标号“0200”，输入“30000”并回车，进入下一个参数设置；
- 3) 参数设置完毕，按“Mode”键退出参数设置状态。

附录

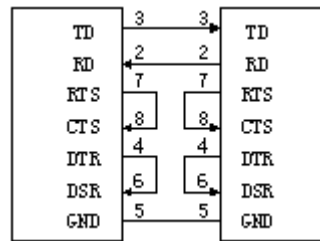
1、RS232 简单无硬件握手通信线缆制作

RS232 是外部设备连接到计算机最常用的一种接口形式。RS232 使用串行通信，一次发送一位数据，与并行通信相比的主要优点是只需要一条线来接收数据，另外一条线发送数据。RS232 是许多计算机和计算机外设公司遵守的事实标准。1962 年 EIA (Electronics Industries Association, 电子工业协会) 把它标准化。

25 针接头是 RS232 的标准接头，但是随着电器设备的越来越小巧，实际上，绝大多数计算机和外设使用的是简化的 9 针 DB9S 接头。DMC 系列运动控制器也采用 9 针 DB9S 接头。下图是 DB9S 接头的信号分配。



RS232 DB9S 信号分配图



无硬件握手的 RS232 连接图

注意：连接线长度不能超过 20 米。照上面制作出的连接线即可适用于计算机到控制器的连接。

2、名词解释

2.1 X 轴、Y 轴、Z 轴：

对于 DMC300A、DMC130A 系统，指的就是 0、1、2 轴；

2.2 坐标、绝对坐标、相对坐标：

DMC 系列控制器使用脉冲值作为坐标单位。控制器通电后，认为当前为坐标原点，某个轴正方向发了 X 个脉冲后，相应轴的坐标加 X，负方向发了 X 个脉冲后，相应轴的坐标减 X。以上坐标又称为绝对坐标；

指令“SETC”可以将当前点设为坐标原点，即绝对坐标清零；

相对坐标指的是目的点相对于当前点的位移，对于发脉冲指令，就是指定轴发指定的脉冲；

绝对坐标相对于发脉冲指令，就是指定轴运动到指定位置（离指定位置有多少，就发多少个脉冲）；

例：三轴控制器通电，当前坐标、绝对坐标为零（X = 0，Y = 0，Z = 0），

执行程序：

率与时间的关系的曲线图。

DMC 系列控制器，采用标准理想的梯形加速曲线，实现了控制步进电机快速起停、精准定位：经可靠测试，控制两相混合式步进电机，最高转速可达 5500 转/分。

3、常见问题解答

3.1 为何控制器连接不上？

答：连接控制器的正常步骤是：

1. 将控制器 RS232 串口与电脑的 RS232 串口正确连接；
2. 保证控制器处于待机或准待机状态；
3. 打开 MotionXXXA 软件，选择“控制器会话”→“串口设置”，根据你接在电脑上的串口，选择合适的串口，并选择波特率为：4800 (DMC300A)、9600(其他)；
4. 选择“控制器会话”→“连接控制器”。

如果出现问题可能的原因：

1. 你的计算机串口设置错误，例如你将控制器连接在计算机的串口 2 上，但你设置为串口 1；
2. 计算机串口波特率设置错误；
3. 串口被其它程序占用，譬如你打开了不止一个 Motion300A 或 Motion110A；
4. 串口线连接有问题，请确认你所使用的为标准 1 公 1 母直通串口线；
5. 你的计算机串口已经坏了，或软件和你的计算机的操作系统有冲突，Motion300A 已经在 Windows98 简体中文、Windows2000 简体中文、WindowsXP 简体中文下测试过，基本可以保证正常；
6. 控制器有问题，请你及时给我们反馈。

3.2 如何更改液晶欢迎画面为自己制作的画面？

答：第一步：在 PC 机的自带画图软件上制作你自己的开机显示画面

第二步：为控制器下载你自己制作的画面（定位于“开机界面”的“0”）。

3.3 如何使用显示指令？

答：请你仔细参阅：

- 1)、“操作说明”中的“键盘功能功能及显示说明”中的“显示部分”；
- 2)、“软件说明”中的“命令控制类指令”的“DISPLAY”指令；
- 3)、“MotionXXXX 编译环境使用说明”中的“设置液晶画面”、“下载”；
- 4). 理解“编程实例”中的举例。

3.4 如何清除坐标值？

答：两种办法：程序运行时使用指令：“SETC”或“CLR M100、CLR M101、CLR M102”

待机状态下：按下“停止”，再按下“复位”键（DMC110A 按下“F1”键）；

注意：在手动调整时，需要先退出手动状态，再按“停止”、“复位”；

3.5 如何改变 DMC300A“手动调整”时，三轴的速度及最小脉冲数？

答：请参见“软件说明”->“变量说明及参数说明”。做法如下（“- 1”页）：

修改 S0000，可以改变三轴运动的起始速度，一般为 1~2000 之间；

修改 S0001，可以改变三轴运动的加速时间，一般为 100~3000 毫秒之间；

修改 S0002，可以改变三轴运动的最高速度即正常速度一般 100~150000 之间；

S0003 是设置响应每次按键的最小脉冲数，你可以根据你的整机设备的要求及不同精度设置不同的数值。例如，S0003 = 10，每次快速按键（三轴进退键），发 10 个脉冲；

S0004 是设置持续按键（三轴进退键）时，所发的最大脉冲数，建议你设置为 99999999，一般情况下，这样数量的脉冲一次按键时间很难发完；

S0005 是一个临界速度：当你按键松开时，如果当前轴已经加速超过该速度，会有一个

类似于加速段的减速段，目的是为了**避免机械瞬间急停**；如果当前轴的速度速度小于该值，脉冲会立即停发。显然，如果该临界速度大于最高速度，那么，无论你按键时间有多长，都没有减速段；如果该临界速度小于起始速度，那么，无论你的按键时间有多短，都会有一个减速段。需要注意的是，这些速度设置对于三轴都有效，也就是说，手动时三轴的速度是一致的。

3.6 以 DMC300A 为例：如何使用输入、输出点？

答：DMC300A 控制器采用输入点低电平有效，即接地有效；输出点采用集电极开漏输出，即有效时输出点对地短路；输入点的使用：例如下图 A 程序功能：输入点 I2 每接地（有效）一次，X 轴发 1000 个脉冲，否则，程序一直停留在第 2 行程序；

使用如下图 B 程序实现功能：继电器闭合 2 秒，断开 2 秒，循环；

<pre> 文件(F) 编辑(E) 工具(T) 控制器会话(L) 查 [Icons] SPEED 0,500,300,10000 TEST0: JNB TEST0,I2 DRVID 0,1000 PAUSE 0 JMP TEST0 END </pre> <p style="text-align: center;">A</p>	<pre> 文件(F) 编辑(E) 工具(T) 控制器会话(L) 查 [Icons] START1: SETB O1 DELAY 2000 CLRB O1 DELAY 2000 JMP START1 END </pre> <p style="text-align: center;">B</p>
---	--

