

# MPC05GA 运动控制器



## 乐创自动化技术有限公司

LEETRO AUTOMATION CO., LTD.

## 版权申明

### 乐创自动化技术有限公司 保留所有权利

乐创自动化技术有限公司(以下简称乐创自动化公司)保留在不事先通知的情况下, 修改本手册中的产品和产品规格等的权力。

### 乐创自动化公司不承担由于使用本手册或本产品不当,所造成直接的、 间接的、附带的或相应产生的损失或责任。

乐创自动化公司具有本产品及其软件的专利权、版权和其它知识产权。 未经授权,不得直接或间接地复制、制造、加工、使用本产品及其相关 部分。

注意:运动中的机器有危险!使用者有责任在机器中设 计有效的出错处理和安全保护机制。

### 目 录

1		前言	•••••••••••••••••••••••••••••••••••••••	1
2	1	概	述	2
	2.1	简介	·	2
	2.2	型号	说明	2
	2.3	技术	特性	3
	2.4	控制	器组成	3
3		快速	使用指南	4
	3.1	准备		4
	3.2	使用	步骤	4
4		运动	控制系统的开发	5
5		<b></b> 個件:		6
J	,	Ψ <b>C</b> TT,		•••••••••••••••••••••••••••••••••••••••
	5.1	MPC	C05/CPU 板上接口	6
	5.2	MPC	205/MC 板上接口	
	5.5	SMC	.02 扩展 IO 说明 <i>抽給</i>	10
	5. 5.	3.1 · 3.2	刃形	
	5.	3.3 ;	接口说明	
	5.	3.4 5	SMC02 与MPC05GA 的连接	
	5.4	P30	转接板	11
	5.	4.1	P30 接线板的示意图	
	Э.	4.2	P30 接线疝于的按口正义	
6	6	计代码	加工文件	
	6.1	系统	坐标系	13
	6.2	绝对	编程和增量编程	14
	6.3	G 代	码加工文件	15
	6.	3.1	标准G 代码指令	
	6.	3.2	步展G代码指令	
7		系统	配置文件	25
	7.1	系统	配置文件格式	25
	7.2	配置	文件中特殊参数详细说明	

7.2.1	<i>轴号设置参数</i>	
7.2.2	轨迹运动参数设置	
7.2.3	系统上电后执行的按键号	
8 人	机界面	
8.1 t	办议及系统设置	
8.2 ±	也址	31
8.3 N	MPC05GA的基本功能示例	32
9 高	速平面轨迹运动	
9.1 I	功能描述	
9.2 蒼	参数说明	33
9.3 约	扁程实现	34
10 编	译下载	
10.1	应用程序中集成编译下载功能	
10.1.1	文件编译	
10.1.2	通过串口下载编译后文件	
10.2	用U盘实现编译后文件下载	
10.2.1		
10.2.2	动态链接库的使用	
10.3.1	用 Visual C++开发控制系统	
10.3.2	? 用Visual Basic 开发控制系统	
11 函	数描述	41
11.1	脉冲当量设置函数	41
11.2	串口通讯控制函数	42
11.3	文件编译和下载运行函数	43
11.4	下载文件的管理	45
11.5	版本查询函数	46
12 错	误处理	
12.1	控制器指示灯显示错误代码	48
12.1.1	运行状态指示灯的位置	
12.1.2	2. 运行状态指示灯显示的信息	
13 问	题及解决方法	
13.1	增加外接 U 盘读数据指示灯	50

13.1.1	功能说明	
13.1.2	实现方法	
13.2	开机回原点功能的实现	
13.2.1	功能说明	
13.2.2	<i>实现方法</i>	

## 1 前言

感谢您选用乐创自动化技术有限公司生产的 MPC05GA 运动控制器! 该用户手册包括使用 MPC05GA 和基于 MPC05GA 开发控制系统的的详细说明。 在使用 MPC05GA 前,请务必仔细阅读用户手册内容,这些内容对于您正确使 用 MPC05GA 运动控制器非常重要。

若您在使用 MPC05GA 运动控制器开发控制系统过程中遇到问题,您可以通过 以下途径获得技术支持:

电话: (028) 85149977 传真: (028) 85187774



	在没有清楚地了解手册内容前,	请不要随意接线并上电使
	用;	
禁止		

$\bigcirc$	未获得授权的情况下请不要自行拆卸以及更换控制卡上的 芯片和元器件;
禁止	

$\bigcirc$	不要让产品碰到水溅,	腐蚀性溶液或气体;
禁止		

### 2概述

### 2.1 简介

MPC05GA 是一款通用型独立式运动控制器,可广泛应用于需要脱机运行的控制设备。

基于 DSP 和 FPGA 的硬件结构确保高性能的运动控制的实现和稳定的系统运行。运动控制的所有细节,包括插补算法,脉冲方向信号的输出,自动升降速的处理,原点、限位等信号的检测处理均在控制器中进行。控制器内部实现了速度前瞻处理算法,通过简单的编程设定即可开发稳定可靠的高性能高速连续轨迹运动控制系统。

MPC05GA 主要针对以下类型的应用:预先根据加工工艺编制加工代码(以 G 代码描述),加工代码下载到控制器后可实现完全脱机运行;

MPC05GA 采用先进的软硬件架构,丰富的开发接口,既可以满足快速的简单应用开发需求,也可以实现更复杂的控制系统的开发。并可以根据用户需求进行特殊功能定制,以适应特殊工艺要求。

MPC05GA 采用开放式结构,软件方面 MPC05GA 配备标准编译下载和系统配置软件,用户也可以利用函数库接口自行开发以便将该功能集成到应用软件中。硬件上配备可选的基于 Modbus 协议的人机界面快速实现面板控制,也可以根据软硬件接口自行开发适合不同应用的控制面板。

### 2.2 型号说明



2.3 技术特性

功能	描述
接口	串口/USB(U盘)
控制轴数	4
通用输入	16路(通过扩展模块 SMC02)
通用输出	3路(控制器)16路(通过扩展模块 SMC02)
最大脉冲输出频率	7.5M
输出信号	脉冲/方向信号
直线、圆弧插补	支持
原点、减速、限位信号	每轴正负限位、原点,该型号无减速信号
多卡共用	不支持
高速轨迹运动	平面高速轨迹运动,带速度前瞻的自动升降速处理
电源	单 24VDC

### 2.4 控制器组成

MPC05GA 运动控制器组成如图所示:



## 3 快速使用指南

### 3.1 准备

基于 MPC05GA 组成一个简单的控制系统,需要准备如下设备:

- MPC05GA 控制器
- 24V 开关电源
- 步进或伺服电机以及驱动器和电源

若要采用 PC 软件面板作为控制面板,则需要:

## 至少带一个串口(COM)的PC一台,Windows98及以上操作系统 一条连接PC与MPC05的电缆(该电缆的定义见硬件接口一章)

若要全脱机运行,则需要:

- 人机界面(支持 Modbus 协议的 MODBUS RTU 类型的设备)
- 一条连接人机界面与 MPC05 的电缆(该电缆的定义见硬件接口一章)
- U盘

### 3.2 使用步骤

在开始使用 MPC05GA 运动控制器前,请先仔细阅读硬件接口一章的内容,并 按照该部分正确连接控制器、电源、电机及驱动器、人机界面和 PC。

- (1) 检查并确认所有连接正确后分别给电机及驱动器、MPC05上电;
- (2) 在 PC 上运行 MPC05 编译下载器;
- (3) 使用 MPC05 编译下载器编译配置文件;
- (4) 使用 U 盘或串口将编译好的配置文件下载到 MPC05GA 卡中并运行, 将配置信息写入到控制器使之生效;
- (5) 使用 MPC05 编译下载器编译 G 代码加工文件;
- (6) 使用 U 盘或串口将编译好的加工文件下载到 MPC05GA 卡中并运行。

## 4 运动控制系统的开发

MPC05GA 以 G 代码文件方式提供用户开发接口,用户只需要根据系统工艺要求编制或生成 G 代码加工文件以及配置文件,利用编译下载器即可将其下载到 MPC05GA 存储运行。

用户可以根据系统工艺要求手工编写 G 代码文件,也可以编程将 PLT 文件或 AutoCAD 图形文件转化为 G 代码文件,还可以使用第三方软件自动生成标准的 G 代码文件。

系统提供的 G 代码编译下载器可以自动生成配制文件,并可以对配置文件或加 工文件进行编译、下载、运行,提供了下载文件的管理和控制面板功能。开发人员 还可以利用系统提供的动态链接库接口函数将编译下载功能集成到应用程序中,并 且可以通过串口连接实现下载文件的管理,并用软件实现控制面板功能。动态链接 库的调用可以参见 10.3 节的动态链接库的使用。接口函数详细介绍参见 11 章的函 数描述。

## 5 硬件接口

MPC05GA 控制器由两部分组成:

- 1) MPC05/MC 运动控制子板
- 2) MPC05/CPU CPU 主板

电路板上有相应的标识。

### 5.1 MPC05/CPU 板上接口



J2: 人机接口 (RS232), 支持 EasyView。

J4: PC 机接口(RS232)

J2、J4 均采用 9 芯 DB 插头:

引脚	1	2	3	4	5	6	7	8	9
定义		TXD	RXD		GND				+5V

J4 与 PC 机 RS232 接线:

/J4

J2 与触摸屏 EasyView 的 RS232 (PLC) 接线:

EasyVi	MPC05/J2		
2 3 5		2 3 5	

J2 与人机界面 BYDseries HMI 的 RS232 (PLC) 接线:

BYDse	BYDseries HMI (PLC)			
2 3 5			2 3 5	

### 5.2 MPC05/MC 板上接口



J1: 输入电源接口(24VDC), MPC05采用单 24VDC供电。



J2: 光隔输出口。	(DB15/M)
------------	----------

编号	名称	定 义	编号	名称	定 义
1	PUL1	1轴脉冲	10	OUT2	通用输出口2
2	DC5V	输出 5V 电源	11	PUL2	2 轴脉冲
3	DC5V	输出 5V 电源	12	PUL3	3 轴脉冲
4	DC5V	输出 5V 电源	13	PUL4	4 轴脉冲
5	OUT4	保留	14	OUT1	通用输出口1
6	DIR1	1 轴方向	15	OUT3	通用输出口3
7	DIR2	2 轴方向			
8	DIR3	3 轴方向			
9	DIR4	4 轴方向			

 9
 DIR4
 4 轴方向

 注:除通用输入/输出信号外,信号名称中的1、2、3、4分别对应控制器上的第1、

 2、3、4 轴。



DC5V 为 MPC05 输出的 5V 电源,可供驱动器的共阳极,请 不要用外部 5V 电源接入 2/3/4 引脚。

#### J3: 光隔输入口 (DB15/F)

编号	名称	定 义	编号	名称	定 义
1		保留	10	EL1-	1 轴负向限位
2	EL4+	4 轴正向限位	11		保留
3	EL3+	3 轴正向限位	12	ORG4	4 轴原点
4	EL2+	2 轴正向限位	13	ORG3	3 轴原点
5	EL1+	1 轴正向限位	14	ORG2	2轴原点
6	GND	输出 5V 电源地	15	ORG1	1 轴原点
7	EL4-	4 轴负向限位			
8	EL3-	3 轴负向限位			
9	EL2-	2 轴负向限位			

()
注意

输入输出口均采用单端方式

### 5.3 SMC02 扩展 IO 说明

### 5.3.1 功能

SMC02 作为 MPC05GA 的 I/O 扩展模块,可以扩展 16 路输入和 16 路输出。

### 5.3.2 尺寸

单位: mm



### 5.3.3 接口说明

- J1 与 MPC05 的 J7 通过 20 芯扁平线相连;
- J5 为输入端口: X1~X16;
- J6 为输出端口: Y1~Y16;
- VCC 为输入电源正极, GND 为输入电源负极。输入电源为 24V。



### 5.3.4 SMC02 与 MPC05GA 的连接

SMC02

MPC05/MC

### 5.4 P30 转接板

### 5.4.1 P30 接线板的示意图



引脚号	名称	定义	备注	引脚号	名称	定义	备注
1	EL1+	正向限位1	输入	2	EL1-	负向限位1	输入
3	ORG1	原点1	输入	4	EL2+	正向限位2	输入
5	EL2-	负向限位2	输入	6	ORG2	原点 2	输入
7	EL3+	正向限位 3	输入	8	EL3-	负向限位3	输入
9	ORG3	原点 3	输入	10	EL4+	正向限位4	输入
11	EL4-	负向限位4	输入	12	ORG3	原点 4	输入
13		保留	输入	14		保留	输入
15	GND	接地	输入	16	Out1	通用输出1	输出
17	Out3	通用输出 3	输出	18	Out2	通用输出 2	输出
19	Out4	通用输出4	输出	20	PUL4	脉冲 4	输出
21	DIR4	方向 4	输出	22	PUL3	脉冲 3	输出
23	DIR3	方向 3	输出	24	V5	5V 输出	输出
25	V5	5V 输出	输出	26	PUL2	脉冲 2	输出
27	DIR2	方向 2	输出	28	DIR1	方向1	输出
29	PUL1	脉冲 1	输出	30	Gnd	接地	输出

### 5.4.2 P30 接线端子的接口定义

## 6 G 代码加工文件

本章详细地描述了 MPC05GA 可以处理的 G 代码功能描述。其中,在 G 代码中使用的单位可以有如下两种情况:

一、以脉冲数为单位:

- 1. 位移(或距离)的单位为P(Pulse),即脉冲数;
- 2. 速度的单位是 PPS (Pulse/sec), 即脉冲/秒;
- 3. 加速度和减速度的单位是PPSS (Pulse/sec<sup>2</sup>),即脉冲 /  $\psi^2$ 。

#### 二、以长度(如 mm)为单位

- 1. 位移(或距离)的单位为mm,即毫米;
- 2. 速度的单位是 mm/sec, 即毫米/秒;
- 3. 加速度和减速度的单位是mm/sec<sup>2</sup>,即毫米/秒<sup>2</sup>。

### 6.1 系统坐标系

系统按照以右手法则确定的笛卡儿直角坐标系(the Right-handed Rectanglar Cartesian Coordinate System)作为编程的标准坐标系。



系统坐标系是系统内部唯一的坐标系,一般与设备机架固连,其坐标系原点通 常与设备的机械原点开关位置对应。

 当以 G29 指令回机械原点后,可以用 M90 Q09 指令设定坐标原点。

 以下假定 X、Y、Z 轴分别与控制器中的 1、2、3 轴相对应

 G29 X1 Y1 Z1
 ; X、Y、Z 轴正向回机械原点

 M90 Q09 C1 S0
 ; 设定 X 轴的原点坐标为 0

 M90 Q09 C2 S0
 ; 设定 Z 轴的原点坐标为 0

 M90 Q09 C3 S0
 ; 设定 Z 轴的原点坐标为 0

若设备没有安装机械原点,则不能按以上代码完成回原点操作,则系统上电启 动后的当前位置为坐标原点,或者在完成一些初始化运动定位后可以用以下代码完 成原点设定:

M90 Q09 C1 S0	;	设定 X 轴的原点坐标为 0
M90 Q09 C2 S0	;	设定Y轴的原点坐标为0
M90 Q09 C3 S0	;	设定 Z 轴的原点坐标为 0

为便于加工过程中的定位,可以用 G92 指令设定加工起点。

G92 X1000 Y1000 Z1000 ; 设定加工起点,其坐标值为在系统坐标系中的绝对坐标。程序中任何地方可以用 G28 指令回到加工起点。



加工起点和机械原点的关系如图所示:

### 6.2 绝对编程和增量编程

对于 G00、G01、G02、G03 等运动指令,可以采用绝对编程和增量编程两种方式。

● 绝对编程

当程序中以 G90 设定为绝对编程方式时,其后的运动指令坐标参数均表示在系统坐标系中的绝对坐标值,直到用 G91 设定为增量编程方式为止;

● 增量编程

当程序中以 G91 设定为增量编程方式时,其后的运动指令坐标参数均表示相对 于当前位置的相对坐标值(即相对位移量),直到用 G90 设定为绝对编程方式为止; 以下示例说明绝对编程和相对编程的区别和使用方法:





### 6.3 G 代码加工文件

一个完整的加工文件由一段主程序和多个子程序组成,其结构如下示例代码:

;以下是主程序 N02 G91 G01 X1000 Y0 N03 G01 X1000 Y1000 N04 G01 X0 Y1000 N05 M98 P00011000 ;调用子程序 1 N06 G01 X5000 Y-5000 N07 M98 P00012000 ;调用子程序 2 N07 G01 X2000 Y0

N08 M02	; 主程序结束
; 以下是子程序 1	
O1000	; 子程序1入口标记
N1001 G01 X2000 Y2000	
N1002 G01 X2000 Y0	
N1003 G01 X-2000 Y-2000	
N1004 G01 X-2000 Y0	
N1005 M99	; 子程序结束
; 以下是子程序 2	
O2000	; 子程序2入口标记
N2001 G01 X4000 Y0	
N2002 G01 X-4000 Y-4000	
N2003 G01 X-4000 Y0	
N2004 G01 X4000 Y4000	
N2005 M99	; 子程序结束

G代码加工文件为纯文本文件,其每一行表示一条指令或一个标记; 以分号;开始的一行为注释行;

基本指令结构如下:



为了实现某些特殊功能, MPC05GA 对标准 G 代码指令进行了扩展, 扩展 G 代码指令结构如下:

对 M90 辅助功能 G 代码进行扩展,以子功能号 Q\_\_区分不同的扩展功能。



### 6.3.1 标准 G 代码指令

由于 MPC05GA 卡为四轴控制卡,所以在 G 代码中启用第四轴。第四轴的坐标 指令采用字符 "A"表示,第四轴对应的圆心以 "L"表示。如:1 轴和 4 轴做顺时 针 90 度圆弧插补,则 G 代码指令为:G02 X1000 A1000 I1000 L0.

#### • G00

指令说明:快速定位指令,各轴之间没有插补关系,运动同时启动但可能不同时结束,各轴按照系统设置缺省速度运动或按照扩展G代码指令设置速度运动

指令格式: G00 X1000 G00 X1000Y-2000 G00 X1000 Y3000 Z-1000 G00 X1000 Y3000 A-1000

参数说明: X\_、Y\_、Z\_、A\_分别表示 X、Y、Z、A 轴运动的绝对坐标或增量坐标

#### • G01

- 指令说明: 直线插补指令, 各轴之间按照系统设置缺省矢量速度或按照扩展 G 代码 指令设置矢量速度做直线插补运动
- 指令格式: G01 X1000 Y-2000 ; XY 两轴做直线插补 G01 X1000 Y3000 Z-1000 ; XYZ 三轴做直线插补 G01 X1000 Y3000 A-1000 ; XYA 三轴做直线插补
- 参数说明: X\_、Y\_、Z\_、A\_分别表示 X、Y、Z、A 轴运动的绝对坐标或增量坐标

#### • G02

- 指令说明:顺圆弧方向圆弧插补指令,可以采用圆心坐标+终点坐标的编程方式或终 点坐标加半径的编程方式。
- 指令格式: G02 X1000 Y0 I500 J0 ; 终点坐标+圆心坐标 G02 X1000 Y0 R500 ; 终点坐标+半径
- 参数说明: X\_、Y\_、Z\_、A\_分别表示 X、Y、Z、A 轴运动的绝对坐标或增量坐标 I\_、J\_、K\_、L\_分别表示圆心点相对于起点的增量坐标 R\_ 表示圆弧半径
- 特殊说明: 若采用终点+圆心方式编程,则终点标识 XYZA 分别与圆心标识的 IJKL ——对应,否则会出现最终走出来的结果不正确。

### • G03

- 指令说明: 逆圆弧方向圆弧插补指令, 可以采用圆心坐标+终点坐标的编程方式或终 点坐标加半径的编程方式
- 指令格式: G03 X1000 Y0 I500 J0 ; 终点坐标+圆心坐标 G03 X1000 Y0 R500 ; 终点坐标+半径
- 参数说明: X\_、Y\_、Z\_、A\_分别表示 X、Y、Z、A 轴运动的绝对坐标或增量坐标 I\_、J\_、K\_、L\_分别表示圆心点相对于起点的增量坐标 R\_ 表示圆弧半径
- 特殊说明: 若采用终点+圆心方式编程,则终点标识 XYZA 分别与圆心标识的 IJKL ——对应,否则会出现最终走出来的结果不正确。

#### • G04

指令说明: 延时指令 指令格式: G04 X3.5 ; 延时 3.5 s 参数说明: X\_表示以秒为单位的延时时间

#### • G28

- 指令说明:回加工起点指令,该指令用于回到预先设定的加工起点位置,加工起点 设定由 G92 指令设定
- 指令格式: G28 X0 ; X 轴回加工起点 G28 X0 Y0 ; X、Y 轴同时回加工起点 G28 X0 Y0 Z0 ; X、Y、Z 轴同时回加工起点 G28 X0 Y0 A0 ; X、Y、A 轴同时回加工起点
- 参数说明: X0、Y0、Z0、A0分别表示X、Y、Z、A轴从当前位置运动到设定的加工起点,参数固定为0

#### • G29

指令说明:回机械原点,该指令用于回到机械原点位置,指令执行时将一直按照指 定的方向运动,直到遇到机械原点开关信号有效则该指令执行完毕

指令格式:	G29 X1	;	X 轴回原点
	G29 X1 Y1	;	X、Y 轴同时回原点
	G29 X1 Y1 Z-1	;	X、Y、Z 轴同时回原点
	G29 X1 Y1 A-1	;	X、Y、A 轴同时回原点
4. NO			

参数说明: X\_、Y\_、Z\_、A\_ 分别表示 X、Y、Z、A 轴回原点的方向,1表示正向回原点,-1表示负向回原点

#### • G90

- 指令说明:设定编程方式为绝对编程方式,当该指令执行后,其后的所有坐标值将 采用绝对坐标,直到执行 G91 指令为止
- 指令格式: G90 G01 X1000 Y2000 ; 转换到绝对编程方式 G01 X3000 Y-1000 ; 仍然采用绝对编程方式

参数说明:

#### • G91

指令说明:设定编程方式为增量编程方式,当该指令执行后,其后的所有坐标值将 采用相对坐标,直到执行 G90 指令为止

指令格式: G91 G01 X1000 Y2000 ;转换到增量编程方式 G01 X3000 Y-1000 : 仍然采用增量编程方式 参数说明:

#### • G92

指令说明:设定加工起点位置,以配合G28回加工起点指令使用

指令格式: G92 X1000 G92 X1000 Y-2000 G92 X1000 Y3000 Z-1000 G92 X1000 Y3000 A-1000

参数说明: X\_、Y\_、Z\_、A\_ 分别表示 X、Y、Z、A 轴加工起点位置的绝对坐标

#### • M02

指令说明:	主程序结束标记		
指令格式:	G90 X1000 Y-1200	;	主程序最后一条指令
	M02	;	主程序结束
参数说明:	无		

#### **M09**

- 指令说明:通用输出指令,改变某通道的输出状态
- 指令格式: M09 C2 S1 ; 将通道 2 状态设置为 ON M09 C2 S0 ; 将通道 2 状态设置为 OFF
- 参数说明: C 表示通道号; S 表示该通道的状态(1表示 ON; 0表示 OFF)。通 道号取值范围为 1~3 时, 对应 MPC05/MC 板上 J2 口的通用输出口 1, 2, 3: 当取值范围为 1001~1016 时,对应 IO 扩展板 SM02 上的通用输出口 1~16.

#### M98

指令说明:调用子程序 指令格式: M98 P00010800 ; 调用标记 O0800 开始处的子程序 参数说明: P\_P 后面有 8 位数,其中前四位数表示调用次数;后四位数表示子程序 入口地址,即代码中以字母 O 开始标记的子程序入口地址

#### • M99

指令说明:子程序代码结束 指令格式: M99 参数说明:无

### 6.3.2 扩展 G 代码指令

为了描述一些特殊功能的实现, MPC05GA 对标准 G 代码进行了扩展, 扩展 G 代码以 M90 加子功能号的方式描述, 具体如下:

#### • M90 Q00

指令说明:设定 X、Y、Z、A 轴对应的控制器上的轴号 指令格式: M90 Q00 B2 D3 E4 S1 参数说明: B\_、D\_、E\_、S\_ 分别表示 X、Y、Z、A 轴对应的控制器上的轴号

#### • M90 Q01

指令说明:	轨迹运动开始		
指令格式:	M90 Q01	;	轨迹运动开始
	G01 X100 Y1000	;	轨迹运动的第一条线段
	G01 X100 Y1000	;	轨迹运动的第二条线段

参数说明: 无

#### • M90 Q02

指令说明:	轨迹运动结束		
指令格式:			
	G01 X100 Y1000	;	轨迹运动的倒数第二条线段
	G01 X100 Y1000	;	轨迹运动的倒数第一条线段
	M90 Q02	;	轨迹运动结束
会粉道明	Ŧ		

参数说明: 九

#### • M90 Q03

- 指令说明:设置轨迹运动的矢量速度
- 指令格式: M90 Q03 L1000 H10000 A20000
  - ; 轨迹运动矢量速度
- 参数说明: L\_ 矢量起始速度; H\_ 矢量高速运动速度; A\_ 矢量加速度

#### • M90 Q04

- 指令说明:设置轨迹运动拐弯许用最大加速度,该加速度因矢量速度方向变化而产 生。
- 指令格式: M90 Q04 A2000
  - ; 轨迹运动拐弯许用加速度
- 参数说明: A\_ 拐弯许用加速度,该值越大,拐弯点处速度越高,若超过机械允许的 值,则可能在拐弯点处产生机械冲击。该值越小,则拐弯点处速度越低。 因此该值应该通过实验测定,以不产生机械冲击情况下的最大值为设定 值,以避免拐弯点处速度下降过多导致总体效率下降。

#### • M90 Q05

- 指令说明:设置轨迹运动拆分精度,即轨迹拆分出的直线段长度。在轨迹运动中需要设置该参数,该参数通常设置范围为10-50。
- 指令格式: M90 Q05 C30
- 参数说明: C\_ 拆分精度

#### • M90 Q07

- 指令说明:设置单指令插补运动的矢量梯形速度
- 指令格式: M90 Q07 L1000 H10000 A20000
- 参数说明: L\_ 矢量起始速度; H\_ 矢量高速运动速度; A\_ 矢量加速度

#### • M90 Q08

- 指令说明:设置单指令插补运动的矢量常速度
- 指令格式: M90 Q08 L1000
- 参数说明: L\_ 矢量常速度

#### • M90 Q09

- 指令说明:设置某轴的绝对位置。控制器内部有一个基础坐标系,初始化各轴绝对 位置为 0,当执行运动指令后,其坐标将发生变化,该指令可以设置某 轴的绝对坐标而不会产生运动
- 指令格式: M90 Q09 C1 S2000 ; 设置 X 轴的绝对坐标位置
- 参数说明: C\_ 轴号; S\_ 绝对位置 轴号范围是 1~4, 其中 1、2、3、4 分别代表 X、Y、Z、A 轴

#### • M90 Q10

- 指令说明:手动点动定位运动,主要用于按键功能编程。该指令执行后指定轴将按 照系统配置文件中定义的点动速度设置运动,直到调用 M90 Q11 指令才 会停止;或者到达软限位位置而停止,但若没有回过原点,则软限位将 不会起作用,因此该指令将一直运动直到调用 M90 Q11 指令才会停止
- 指令格式: M90 Q10 C1 S1 ; X 轴正向点动
- 参数说明: C\_ 轴号; S\_ 运动方向: 1 为正向; 0 为负向 轴号范围是 1~4, 其中 1、2、3、4 分别代表 X、Y、Z、A 轴

#### • M90 Q11

指令说明:手动点动定位运动停止,主要用于按键功能编程。该指令执行后指定轴 点动运动将会停止。

指令格式: M90 Q11 C1 ; X 轴点动停止

参数说明: C\_ 轴号 轴号范围是 1~4, 其中 1、2、3、4 分别代表 X、Y、Z、A 轴

#### • M90 Q12

- 指令说明:设置指定轴运动的梯形速度
- 指令格式: M90 Q12 C1 L100 H10000 A20000
- 参数说明: C\_ 轴号; L\_ 起始速度; H\_ 高速运动速度; A\_ 加速度 轴号范围是 1~4, 其中 1、2、3、4 分别代表 X、Y、Z、A 轴

#### • M90 Q13

指令说明:设置指定轴运动的常速度

- 指令格式: M90 Q13 C1 L1000
- 参数说明: C\_ 轴号 L\_ 常速度 轴号范围是 1~4, 其中 1、2、3、4 分别代表 X、Y、Z、A 轴

#### • M90 Q14

- 指令说明:设置运动的速度类型是常速还是快速运动,当执行该指令后,后面所有 的运动指令均按照设定的速度类型运动,直到新的设定指令重新进行设 定
- 指令格式: M90 Q14 S1 ; 快速运动
- 参数说明: S\_ 运动类型, 1 为快速; 0 为常速

## 7 系统配置文件

系统配置文件主要保存如下信息:

- 系统参数
- 缺省运动参数
- 按键功能定义代码

### 7.1 系统配置文件格式

一个完整的系统配置文件由系统参数段和按键功能定义代码段组成,其结构及 内容如下所示:

其中参数段等号右边的值可根据系统的实际情况更改,等号左边的参数名称不 得随意变动。

当在加工文件中没有指定的速度和加速度的情况下系统将采用配置文件中缺省速度和加速度值。

[ParamSection]	;参数段
ParamNum=60	;参数个数,即该参数段参数的个数
ChxNo=1	;X 轴轴号,加工文件中可以用 M90 Q00 指令
	指定 X 轴轴号,指定轴号将代替该缺省值
ChyNo=2	;Y 轴轴号,加工文件中可以用 M90 Q00 指令
	指定 Y 轴轴号,指定轴号将代替该缺省值
ChzNo=3	;Z 轴轴号,加工文件中可以用 M90 Q00 指令
	指定 Z 轴轴号,指定轴号将代替该缺省值
ChaNo=4	; A 轴轴号,加工文件中可以用 M90 Q00 指令
	指定 A 轴轴号,指定轴号将代替该缺省值
LookAheadLow=2000	;轨迹运动的起始速度,加工文件中可以用 M90
	Q03 指令指定该速度,指定速度将代替该缺省值
LookAheadHigh=20000	;轨迹运动的高速度,加工文件中可以用 M90
	Q03 指令指定该速度,指定速度将代替该缺省值
LookAheadAccel=50000	; 轨迹运动的加速度, 加工文件中可以用 M90
	Q03 指令指定该加速度,指定加速度将代替该缺
	省值
LookAheadMaxAccel=5000	;轨迹运动的拐弯许用加速度,加工文件中可以
	用 M90 Q04 指令指定该加速度,指定加速度将
	代替该缺省值
LookAheadMultiLength=30	;轨迹运动的拆分微线段长度,加工文件中可以

用 M90 Q05 指令指定该长度,指定的拆分长度 将代替该缺省值 ;缺省矢量常速度,加工文件中可以用 M90 Q08 DefaultVecConspeed=10000 指令指定速度,指定速度将代替该缺省值 :缺省矢量起始速度,加工文件中可以用 M90 DefaultVecStartSpeed=600 Q07 指令指定速度,指定速度将代替该缺省值 ;缺省矢量工作速度,加工文件中可以用 M90 DefaultVecWorkSpeed=30000 Q07 指令指定速度,指定速度将代替该缺省值 ; 缺省矢量空程速度 DefaultVecMoveSpeed=40000 DefaultVecAccel=80000 :缺省矢量加速度,加工文件中可以用 M90 O07 指令指定速度,指定速度将代替该缺省值 : 系统上电后执行该参数指定按键的功能代码, OnButtonPowerOn=7 该按键对应的功能代码在后面的按键功能定义 段。若指定的按键在后面没有定义,则系统将不 做任何操作 [Ch1Para] CH1RangeMin=-80000 ; 1 轴运动范围最小值 CH1RangeMax=80000 : 1 轴运动范围最大值 ; 1 轴输入有效电平状态 CH1LevelIn=0 ;1 轴缺省常速度,加工文件中可以用 M90 Q13 CH1DefaultConspeed=20000 指令指定速度,指定速度将代替该缺省值 ;1轴缺省起始速度,加工文件中可以用 M90 Q12 CH1DefaultLowSpeed=600 指令指定速度,指定速度将代替该缺省值 ;1轴缺省工作速度,加工文件中可以用 M90 Q12 CH1DefaultHighSpeed=40000 指令指定速度,指定速度将代替该缺省值 :1 轴缺省加速度,加工文件中可以用 M90 O12 CH1DefaultAccel=80000 指令指定速度,指定速度将代替该缺省值 ;1 轴缺省减速度,加工文件中可以用 M90 Q12 CH1DefaultDecel=80000 指令指定速度,指定速度将代替该缺省值 CH1PMoveLowSpeed=100 ;1轴点动起始速度,即M90Q10指令运动的速 度 CH1PMoveHighSpeed=1000 ;1轴点动高速运动速度,即 M90 Q10 指令运动 的速度 CH1PMoveAccel=2000 ;1轴点动加速度,即M90Q10指令运动的加速 度 [Ch2Para] CH2RangeMin=-80000 : 2 轴运动范围最小值

CH2RangeMax=80000	; 2 轴运动范围最大值
CH2LevelIn=0	; 2 轴输入有效电半状态
CH2DefaultConspeed=20000	; 2 轴缺省常速度,加工文件中可以用 M90 Q13
	指令指定速度,指定速度将代替该缺省值
CH2DefaultLowSpeed=600	;2轴缺省起始速度,加工文件中可以用 M90 Q12
	指令指定速度,指定速度将代替该缺省值
CH2DefaultHighSpeed=40000	;2轴缺省工作速度,加工文件中可以用M90Q12
	指令指定速度,指定速度将代替该缺省值
CH2DefaultAccel=80000	;2轴缺省加速度,加工文件中可以用 M90 Q12
	指令指定速度,指定速度将代替该缺省值
CH2DefaultDecel=80000	;2轴缺省减速度,加工文件中可以用 M90 Q12
	指令指定速度, 指定速度将代替该缺省值
CH2PMoveLowSpeed=100	;2轴点动起始速度,即M90Q10指令运动的速
	度
CH2PMoveHighSpeed=1000	;2轴点动高速运动速度,即 M90 Q10 指令运动
	的速度
CH2PMoveAccel=2000	;2轴点动加速度,即M90Q10指令运动的加速
	度
[Ch3Para]	
CH3RangeMin=-80000	; 3 轴运动范围最小值
CH3RangeMax=80000	:3轴运动范围最大值
CH3LevelIn=0	; 3 轴输入有效电平状态
CH3DefaultConspeed=20000	:3 轴缺省常速度,加工文件中可以用 M90 O13
1	指令指定速度,指定速度将代替该缺省值
CH3DefaultLowSpeed=1000	:3轴缺省起始速度,加工文件中可以用 M90 O12
I	指令指定速度,指定速度将代替该缺省值
CH3DefaultHighSpeed=40000	:3轴缺省工作速度,加工文件中可以用 M90 O12
5 - <u>1</u>	指令指定速度,指定速度将代替该缺省值
CH3DefaultAccel=80000	:3 轴缺省加速度,加工文件中可以用 M90 O12
	指令指定速度,指定速度将代替该缺省值
CH3DefaultDecel=80000	· 3 轴缺省减速度,加丁文件中可以用 M90 O12
	指今指定速度,指定速度将代替该缺省值
CH3PMoveLowSpeed=100	$\cdot$ 3 轴占动起始速度,即 M90 O10 指今运动的速
enter movelo mopeca-100	度
CH3PMoveHighSpeed=600	✓ · 3 轴占动高速运动速度. 即 M90 O10 指今运动
errer menepeed=000	
	的速度
CH3PMoveAccel-1000	的速度 •3 轴占动加速度,即 M90 O10 指今运动的加速

### 度

[Ch4Para]	
CH4RangeMin=-80000	; 4 轴运动范围最小值
CH4RangeMax=80000	; 4 轴运动范围最大值
CH4LevelIn=0	; 4 轴输入有效电平状态
CH4DefaultConspeed=20000	;4 轴缺省常速度,加工文件中可以用 M90 Q13 指令指定速度,指定速度将代替该缺省值
CH4DefaultLowSpeed=600	;4轴缺省起始速度,加工文件中可以用 M90 Q12 指令指定速度,指定速度将代替该缺省值
CH4DefaultHighSpeed=40000	;4轴缺省工作速度,加工文件中可以用 M90 Q12 指令指定速度,指定速度将代替该缺省值
CH4DefaultAccel=80000	;4 轴缺省加速度,加工文件中可以用 M90 Q12 指令指定速度,指定速度将代替该缺省值
CH4DefaultDecel=80000	;4 轴缺省减速度,加工文件中可以用 M90 Q12 指令指定速度,指定速度将代替该缺省值
CH4PMoveLowSpeed=100	;4 轴点动起始速度,即 M90 Q10 指令运动的速度
CH4PMoveHighSpeed=1000	;4 轴点动高速运动速度,即 M90 Q10 指令运动的速度
CH4PMoveAccel=2000	;4轴点动加速度,即 M90 Q10 指令运动的加速 度
[ButtonDefineSection]	; 按键功能定义段
ButtonNum=3	; 自定义功能按键个数
Button005_OnClick={	; 按键5按下后执行的代码
N10 M90 Q10 C1 S1 N20 M99	; X 轴正向点动运动
Button005 OnClickUn={	: 按键 5 松开后执行的代码
N10 M90 O11 C1	<b>· X</b> 轴点动停止
N20 M99	
}	
Button006_OnClick={	; 按键 6 按下后执行的代码
N10 M90 Q10 C1 S0 N20 M99	; X 轴负向点动运动
Button006_OnClickUp={	; 按键 6 松开后执行的代码

N10 M90 Q11 C1 N20 M99	; X 轴点动停止
}	
Button007_OnClick={	; 按键7按下后执行的代码
N10 M90 Q00 B1 D2 E3	; 设置 XYZ 轴的轴号分别为 1、2、3
N20 G29 X-1 Y-1 Z-1	; XYZ 轴负向回机械原点
N70 M90 Q09 C1 S0	;将X轴坐标清零
N80 M90 Q09 C2 S0	;将Y轴坐标清零
N90 M90 Q09 C3 S0	; 将 Z 轴坐标清零
N120 M99	

}

以上配置文件中, 按键 XXX 按下表示编号为 XXX 的按键按下后执行的代码; 按键 XXX 松开表示编号为 XXX 的按键松开后执行的代码;

XXX 表示的按键编号为十进制数,必须从 5 开始连续生成,如:005,006,007,...。 其中 1~4 号键为系统定义键,依次为启动、暂停、恢复、停止。按键最大数目包括 1~4 号系统定义键在内为 29 个。

若该按键按下后什么都不做,则只需要书写如下代码:

Button005_C	OnClick={
-------------	-----------

; 按键按下后执行的代码

N06 M99

; 结束指令

}

配置文件中定义的按键为自定义按键,当控制面板中相应按键被按下后,系统 将执行配置文件中定义的按键响应代码。

在 PC 上可通过动态链接库接口函数 GC\_on\_button()调用执行相应按键功能代码。

$\bigcirc$	务必严格按照以上格式生成配置文件,按键数与实际定义
$\cup$	的按键必须一致;按键功能定义代码必须以 M99 指令行结
注意	束;所有参数名称不得随意更改,否则将导致编译失败;

$\bigcirc$	配置文件下载后运行该文件才能生效,运行后配置信息将
$\bigcirc$	会被保存到控制器内部存储器,因此运行后即可删除该配
	置文件以便更有效地利用内部存储空间;
(工忌	

### 7.2 配置文件中特殊参数详细说明

### 7.2.1 轴号设置参数

ChxNo=1 ChyNo=2

- ChzNo=3
- ChaNo=4

若 G 代码加工文件中没有采用 M90 Q00 设置 XYZA 四个轴对应的轴号,则采用配置文件中设置的轴号分别作为 G 代码文件中 XYZA 轴的轴号;若 G 代码加工 文件中有采用 M90 Q00 设置 XYZA 四个轴对应的轴号,则自采用 M90 Q00 设置时 刻起开始采用 G 代码中设置的轴号。

### 7.2.2 轨迹运动参数设置

LookAheadLow=2000 LookAheadHigh=20000 LookAheadAccel=50000 LookAheadMaxAccel=5000 LookAheadMultiLength=30

该组参数可以采用 G 代码扩展指令 M90 Q03、M90 Q04, M90 Q05 扩展指令进行设置。

该组参数主要是针对第三方软件自动生成的标准 G 代码文件的。自动生成的标准 G 代码采用轨迹运动处理,轨迹运动的参数采用配置文件中的缺省的参数。这样,标准 G 代码直接在 G 代码编译下载器上编译下载到 MPC05 卡上后,就可以直接使用,而无须在生成的标准 G 代码中增加一些扩展指令设置这些参数。

### 7.2.3 系统上电后执行的按键号

#### OnButtonPowerOn=7

该参数表示系统上电或,执行7号按键的功能代码,即首先 XYZ 轴回机械原点, 到达原点位置后将原点位置坐标清零。

若该参数为0,则表示系统上电后不做任何操作。若该参数为7,但配置文件的 自定义按键处没有定义7号按键,也不做任何操作。

## 8 人机界面

MPC05GA 与控制面板之间采用 Modbus 协议进行通讯,因此可以采用支持 Modbus 协议的人机界面标准产品来开发控制面板,以下以 MT506LV45WV 为例介 绍具体方法。

### 8.1 协议及系统设置

采用标准的 Modbus 协议通讯。
PLC 类型: Modbus RTU
波特率: 9600;
数据位: 8;
停止位: 1;
校验: 无。

### 8.2 地址

元件选项		山台	米刑	久沆
地址	设备类型	均甩	天空	宙任
1	0x	运行	按键	
2	0x	暂停	按键	
3	0x	恢复	按键	
4	0x	停止	按键	
5~29	0x	自定义功能按键(参见配置文件内容)	按键	
30	0x	删除当前文件	按键	
31	0x	文件号增,用于切换当前文件	按键	
32	0x	文件号减,用于切换当前文件	按键	
33	0x	加工状态: 1-正在加工; 0-加工结束	状态	

● PLC 继电器地址及功能定义如下:

以上按键的地址与配置文件中定义的按键编号对应,当单元值为1则执行配置 文件中"ButtonXXX\_OnClick"={...}定义的代码,当单元值为0则执行配置文件中 "ButtonXXX\_OnClickUp"={...}定义的代码,若没有"ButtonXXX\_OnClickUp"={...} 则该单元为0不做任何操作。

####	类型	数据	数据长度		读	工力会と
프린네	设备	类型	位	字	写	切肥
1						保留
2						保留
3						保留
4	4x	BIN	16	1	写	设置加工数量
5	4x	BIN	16	1	读	扩展 IO 输入状态字, 其各位分别
3						对应 16 个扩展输出口的状态
6	4x	BIN	16	1	读	已加工件数
7	4x	BIN	32	2	读	1 轴坐标
9	4x	BIN	32	2	读	2 轴坐标
11	4x	BIN	32	2	读	3轴坐标
13	4x	BIN	32	2	读	4 轴坐标
15	4x	BIN	16	1	读	已经下载到控制器中的总文件数
16	4x	BIN	16	1	读	当前文件号
17	4x	BIN	64	4	读	文件名 abcdefgh 共 8 个字节数据,
1/						以文本方式显示

● PLC 寄存器地址及功能定义如下:

### 8.3 MPC05GA 的基本功能示例

随软件发布有针对触摸屏(型号:MT506LV45WV)和人机界面(型号:BYD037L)的示例程序供开发人机界面的参考。

## 9 高速平面轨迹运动

MPC05GA 通过扩展 G 代码实现高速平面轨迹运动,控制器内部会根据轨迹信息按照设定的参数自动进行升降速处理,从而避免在高速平面轨迹运动过程中拐弯处出现机械冲击。

### 9.1 功能描述

高速平面轨迹运动即执行机构(如工作台或刀具等)沿根据加工工艺要求预先 规划的任意曲线作高速运动,为避免机械冲击,需要在拐弯前后分别进行降速和升 速的处理。

MPC05GA 内部已经包含了速度处理功能,开发人员只需要预先设定适量速度 和拐弯加速度参数,并将组成任意曲线轨迹的一系列直线段编制成加工代码,系统 即可自动完成拐弯点处升降速处理。

在轨迹运动过程中,可以实现 IO 输出操作和第三轴的运动,只需要在轨迹运动的 G 代码中插入 IO 输出指令和第三轴运动指令,这些指令的执行将不会对轨迹连续性造成影响。

### 9.2 参数说明



如图:

P 点为设定的加工起点, A 点为轨迹起始点, B 点为任意曲线的起始点, C 点为曲线中间的一个拐弯点, D 点为任意曲线的终点。

假定当前点为加工起点,要完成整个加工过程,需要在编写组成轨迹的微直线 段G代码前设定如下参数:

● 矢量速度:
 包括起始矢量速度、工作矢量速度和矢量加速度,加工过程中将从起始矢量速

度开始,按照加速度升到工作矢量速度,加工结束前将会按照加速度降到起始速度; ● 拐弯许用加速度:

当运动执行机构在拐弯点如图中的 C 点处,因矢量速度的突变会产生加速度,拐弯许用加速度用于限制拐弯处的矢量速度,该参数需要根据机械惯量通过实验确定,以拐弯处不产生机械冲击为限度。其值越小,拐弯处越平稳,反之则冲击越明显。

### 9.3 编程实现

如下 G 代码实现图中的运动轨迹(假定 123 轴分别与 XYZ 轴对应)。

M90 Q14 S0 M90 Q13 C3 L1000 G90 G01 X150 Y350 M90 Q01 M90 Q03 L2000 H30000 A100000 M90 Q04 A8000 M09 C1 S1 G01 X250 Y300 G01 X255 Y303	; 非轨迹运动采用常速方式 ; 设定 Z 轴常速度 ; 切换到绝对编程方式 ; 运动到 A 点 ; 设定轨迹运动开始 ; 设定轨迹运动矢量速度和加速度参数 ; 设定拐弯允许加速度 ; 通用输出口 1 置为 1, 落笔 ; 运动到 B 点
 G01 X495 Y154 G01 X500 Y150 G00 Z100  G01 X348 Y325 G01 X350 Y330 M09 C1 S0 M90 Q02 G28 X0 Y0	; 运动到C点 ; 从C点开始启动第三轴运动 ; ; 运动到D点 ; 通用输出口1置为0, 抬笔 ; 轨迹运动结束 ; 回加工起点

以上代码中在 C 点处 Z 轴会启动运动,并且与 XY 的轨迹运动同时进行,若要 在 C 点处停止轨迹运动,待 Z 轴运动完毕,再继续轨迹运动,则应将 C 点前后分成 两段轨迹,代码如下:

M90 Q14 S0	;	非轨迹运动采用常速方式
M90 Q13 C3 L1000	;	设定 Z 轴常速度
G90	;	切换到绝对编程方式
G01 X150 Y350	;	运动到 A 点
M90 Q01	;	第一段轨迹运动开始
M90 Q03 L2000 H30000 A100000	;	设定轨迹运动矢量速度和加速度参数
M90 Q04 A8000	;	设定拐弯允许加速度
M09 C1 S1	;	通用输出口1置为1,落笔
G01 X250 Y300	;	运动到 B 点

G01 X255 Y303	
 G01 X495 Y154 G01 X500 Y150 M90 Q02 G00 Z100 M90 Q01	; 运动到 C 点 ; 第一段轨迹运动结束 ; 从 C 点开始启动第三轴运动 ; 第二段轨迹运动开始
 G01 X348 Y325 G01 X350 Y330 M09 C1 S0 M90 Q02 G28 X0 Y0	; ; 运动到 D 点 ; 通用输出口 1 置为 0, 抬笔 ; 轨迹运动结束 ; 回加工起点



在曲线轨迹指令中,只能插入第三轴的单轴运动指令 G00 和 IO 输出指令 M09,若插入其它指令则可能导致运动不正常



## 10 编译下载

当按照工艺编制加工 G 代码文件和配置文件后,需要对这两个文件进行编译并 下载到控制器执行。

编译和下载可以用标准编译下载器完成,也可以将编译下载功能集成到应用程 序中,以下分别介绍两种方法。

### 10.1 应用程序中集成编译下载功能

应用程序可以利用提供的 DLL (GCode.dll) 接口函数实现加工文件和系统配置 文件的编译和下载,从而将编译下载功能无缝集成到应用程序中。

### 10.1.1文件编译

调用如下函数:

GC\_compile\_file (char\* SrcFile, char\* OutFile);

函数参数为以\0 结束的包含文件绝对路径和文件名的字符串,前一个为 G 代码 文件或配置文件,后一个为编译后的输出文件。

其中编译输出文件名只能由英文字母和数字组成,符合 8.3 长度限制,其后缀为.mol,即\*\*\*\*\*\*\*.mol,其中\*代表任意英文字母或 0~9 的数字。

### 10.1.2通过串口下载编译后文件

初始化部分:

int nRtn;

然后调用以下函数打开串口连接:

nRtn = GC\_open\_link(1); //串口通讯连接,参数为串口通道号,返回 0 表示 连接成功,返回非 0 表示失败。

之后可调用 GC\_download\_file()下载加工文件。

调用 GC\_on\_button()函数运行下载的文件。

在应用程序结束下载后调用 GC\_close\_link(1)关闭串口连接;

通过串口下载时,对于指定下载的文件和控制器中已经下载的文件同名情况的 处理模式可用 GC\_set\_download\_mode()函数进行设置。

### 10.2 用 U 盘实现编译后文件下载

通过 U 盘可方便地实现固件升级文件的下载升级,以及加工文件和配置文件的的下载。

### 10.2.1 固件升级操作

固件升级步骤:

将新固件文件(05FA???.FMW 和 05FA???.HDW 两个文件,?位置代表的数字随不同的版本会有所变化)拷贝到 U 盘根目录(格式化为 FAT16,建议 U 盘上不要有别的文件);

- (1) MPC05 上电, MPC05/CPU 主板上的指示灯 D1 快速闪 2 次;
- (2) D1 快闪 2 次后的 5 秒内将 U 盘插入 MPC05;
- (3) D1 一直亮(一般 2~10 秒,视固件大小而定),此时正在更新固件;若 U 盘上有指示灯,则按照 U 盘使用说明观察指示灯闪亮来判断是否正在 读数据;
- (4) D1 快速闪亮,表示更新固件结束;若U盘上有指示灯,则按照U盘使 用说明观察指示灯闪亮来判断是否读数据结束;
- (5) 拔出U盘, MPC05将正常启动 DSP 固件程序。

若拔出 U 盘后, MPC05 不能正常工作,则可能是固件更新时出错。可以重复以 上步骤重新更新固件。若 MPC05 依然不能正常工作,则与供应商联系。



请选用带指示灯的 U 盘,以便观察下载数据是否完成。否则可能引起下载数据不完整或错误。

### 10.2.2加工文件及配置文件下载

加工及配置文件的下载:

- 将 PC 上编译后生成的文件(\*.mol) 拷贝到 U 盘根目录(格式化为 FAT16)。
  - (1) MPC05 上电启动进入工作状态;
  - (2) 将 U 盘插入 MPC05;
  - (3) MPC05/CPU 主板上的指示灯 D1 一直亮(一般几秒到几分钟,视文件 大小而定),此时正在下载加工文件;
  - (4) D1 快速闪亮,表示下载加工文件结束;
  - (5) 拔出 U 盘,通过控制面板可选择并运行加工文件。

若下载的是配置文件,则通过控制面板选择并运行后新的 1 配置才能生效,运行后配置信息被写入到控制器存储器, 因此运行后可删除该配置文件。若是加工数据文件,则通 注意 过控制面板选择并运行可直接启动加工。

()	在第一次使用 MPC05GA 时,请务必按照机器参数生成配置文件,下载并运行该文件。以后当机器参数发生变化后,
注意	也应进行问样操作。



### 10.3 动态链接库的使用

### 10.3.1用 Visual C++开发控制系统

### 10.3.1.1 开发环境

用户可以使用 VC5.0 或更高版本,来进行 Windows 平台下运动控制系统开发。

### 10.3.1.2 动态链接库函数调用方法

在 VC 中调用动态链接库 DLL 中函数有两种方法:

● 隐式调用(静态加载)

隐式调用需要如下文件: DLL 函数声明头文件 GCodeMpc05ga.h; 编译连接时用的导入库文件 GCode.lib; 动态链接库文件 GCode.dll、Mpc05gaInterface.dll 和 CommM05.dll;

● 显式调用

显式调用只需要如下文件: 动态链接库文件 GCode.dll; 显式调用方法需要调用 Windows API 函数加载和释放动态链接库。方法如下:调用 Windows API 函数 LoadLibrary()动态加载 DLL;

调用 Windows API 函数 GetProcAddress()取得将要调用的 DLL 中函数的指针; 用函数指针调用 DLL 中函数完成相应功能;

在程序结束时或不再使用 DLL 中函数时,调用 Windows API 函数 FreeLibrary() 释放动态链接库。

以上在两种方法均为 VC 中调用动态链接库函数的标准方法,若要获得更具体的调用方法和帮助,请参考微软 Visual Stutio 开发文档 MSDN 或相关 VC 参考书籍中相应部分内容。

#### 10.3.1.3 示例程序的使用

提供的示例程序 VC60 能实现编译下载功能,该示例为采用 VC6.0 建立的工程, 提供完整的源代码。

可以在 Visual C++ 6.0 开发环境中加载该工程并编译运行。

### 10.3.2用 Visual Basic 开发控制系统

#### 10.3.2.1 开发环境

用户可以使用 VB5.0 或更高版本,来进行 Windows 平台下运动控制系统开发。

#### 10.3.2.2 动态链接库函数调用方法

开发一个简单的 Visual Basic 控制程序非常容易。按照如下步骤可以快速开发一个简单的控制程序。

- 1. 按照硬件接口一章内容正确连接 PC 和 MPC05GA;
- 2. 用 Visual Basic 写一个界面程序;
- 3. 将 GCodeMpc05ga.bas 文件添加到 VB Project 中去;
- 4. 在应用程序中调用函数实现编译下载功能。

所有 Visual Basic 的教材都介绍了如何写界面程序,包括按钮、对话框以及菜单等。

在 VB 中调用动态链接库(DLL)中函数应包括两部分工作:

● 函数声明

每一个动态链接库(DLL)中的函数在 VB 中的声明已经包含在

GCodeMpc05ga.bas 文件中了,该文件可在\DLL\文件夹下找到,用户只需要将该文件添加进 VB 工程中即可。

● 函数调用

若调用函数的返回值为空或不需要返回值,则按如下方法调用:

GC\_open\_link 1

或

Call GC\_open\_link (1)

若要得到函数的返回值,则按如下方法调用: Dim rtn As Long rtn = GC\_open\_link (1)

注意: 传递的参数数据类型及接收返回值的变量类型应与函数声明的数据类型一致,并且建议函数描述中所有 int 型(C语言中的整形)和 long 型(C语言中的长整形)参数及返回值均统一采用 Long 型(VB中的长整形)数据类型;所有的 float (C语言中的单精度浮点型)和 double(C语言中的双精度浮点型)参数及返回值均统一采用 Double 型(VB中的双精度浮点型)数据类型,否则将可能产生无法预料的结果。

以上方法为 VB 中调用动态链接库函数的标准方法,若要获得更具体的调用方法和帮助,请参考微软 Visual Stutio 开发文档 MSDN 或相关 VB 参考书籍中相应部分内容。

### 10.3.2.3 示例程序的使用

提供的示例程序能实现编译下载功能,该示例为采用 VB6.0 建立的工程,提供 完整的源代码。

可以在 Visual Basic 6.0 开发环境中加载该工程并编译运行。

## 11 函数描述

本章详细地描述了 MPC05GA 函数库中的每一个函数。其中,在函数库中使用的函数返回值约定通常如下:

#### 函数返回值

运动库中的大多数函数是整型函数,它们的返回意义如下:

- 0 函数执行正确;
- -1 函数执行错误。

### 11.1 脉冲当量设置函数

该类函数主要用于设置 XYZA 四个轴的脉冲当量,相关函数有:

int GC\_set\_pulse\_perunit\_chx (long pulse); /\*设置 X 轴每个长度单位代表的脉冲数\*/ int GC\_set\_pulse\_perunit\_chy (long pulse); /\*设置 Y 轴每个长度单位代表的脉冲数\*/ int GC\_set\_pulse\_perunit\_chz (long pulse); /\*设置 Z 轴每个长度单位代表的脉冲数\*/ int GC\_set\_pulse\_perunit\_cha (long pulse); /\*设置 A 轴每个长度单位代表的脉冲数\*/

函数名: GC\_set\_pulse\_perunit\_chx

目 的:用该函数设置G代码中X轴的脉冲当量。

- 语 法: int GC\_set\_pulse\_perunit\_chx (long pulse);
   pulse: 一个长度单位代表的脉冲数。
- 调用例子: GC\_set\_pulse\_perunit\_chx (150);
- 描 述:对于第三方软件生成的标准 G 代码,很多生成的 G 代码采用长度如毫米 为单位进行编程的,而控制卡要求的长度单位为脉冲数,所以在编译 G 代码文件前需要设置各轴的脉冲当量值。若 G 代码加工文件或配置文件 本身采用的脉冲数为单位进行编制的,那么需要将脉冲当量设置为 1。 需要注意的是一个 G 代码加工文件或配置文件,里面的单位应保持一致。
- 返回值:如果调用成功,函数返回0,否则返回-1。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参 见**:**

语

函数名: GC\_set\_pulse\_perunit\_chy

- 目 的:用该函数设置G代码中Y轴的脉冲当量。
  - 法: int GC\_set\_pulse\_perunit\_chy (long pulse); pulse: 一个长度单位代表的脉冲数。

调用例子: GC\_set\_pulse\_perunit\_chy (150);

描 述:对于第三方软件生成的标准 G 代码,很多生成的 G 代码采用长度如毫米 为单位进行编程的,而控制卡要求的长度单位为脉冲数,所以在编译 G 代码文件前需要设置各轴的脉冲当量值。若G代码加工文件或配置文件本身采用的脉冲数为单位进行编制的,那么需要将脉冲当量设置为1。 需要注意的是一个G代码加工文件或配置文件,里面的单位应保持一致。

- 返回值:如果调用成功,函数返回0,否则返回-1。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:
- 函数名: GC\_set\_pulse\_perunit\_chz
- 目 的:用该函数设置G代码中Z轴的脉冲当量。
- 语 法: int GC\_set\_pulse\_perunit\_chz (long pulse);
   pulse: 一个长度单位代表的脉冲数。
- 调用例子: GC\_set\_pulse\_perunit\_chz (150);
- 描述:对于第三方软件生成的标准 G代码,很多生成的 G代码采用长度如毫米为单位进行编程的,而控制卡要求的长度单位为脉冲数,所以在编译 G代码文件前需要设置各轴的脉冲当量值。若 G代码加工文件或配置文件本身采用的脉冲数为单位进行编制的,那么需要将脉冲当量设置为 1。
   需要注意的是一个 G代码加工文件或配置文件,里面的单位应保持一致。
- 返 回 值:如果调用成功,函数返回0,否则返回-1。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:
- 函数名: GC\_set\_pulse\_perunit\_cha
- 目 的:用该函数设置G代码中A轴的脉冲当量。
- 语 法: int GC\_set\_pulse\_perunit\_cha (long pulse);pulse: 一个长度单位代表的脉冲数。
- 调用例子: GC\_set\_pulse\_perunit\_cha (150);
- 描 述:对于第三方软件生成的标准 G 代码,很多生成的 G 代码采用长度如毫米 为单位进行编程的,而控制卡要求的长度单位为脉冲数,所以在编译 G 代码文件前需要设置各轴的脉冲当量值。若 G 代码加工文件或配置文件 本身采用的脉冲数为单位进行编制的,那么需要将脉冲当量设置为 1。 需要注意的是一个 G 代码加工文件或配置文件,里面的单位应保持一致。
- 返 回 值:如果调用成功,函数返回0,否则返回-1。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

### 11.2 串口通讯控制函数

该类函数主要用于 PC 通过串口对控制器进行联机控制,相关函数有:

int GC\_open\_link (int ch); /\*打开串口连接\*/

int GC\_close\_link (int ch); /\*关闭串口连接\*/

#### 函数名: GC\_open\_link

- 目 的:用 GC\_open\_link 函数打开串口连接。
- 语 法: int GC\_open\_link (int ch);

ch: 串口通道号, 一般为1或2, 对应 PC 上的 COM1 或 COM2

- 调用例子: GC\_open\_link (1); /\*打开 COM1 连接\*/
- 描述: 在使用串口与 MPC05GA 连接进行加工文件下载前必须调用该函数打开 该连接,下载完毕后调用 GC\_close\_link()关闭连接。
- 返回值:如果调用成功,GC\_open\_link函数返回0,否则返回-1。

系 统: WINDOWS98, WINDOWS2000, WINDOWS XP

- 注 释:调用下载函数、启动按键函数和下载文件管理类函数前必须调用该函数 打开串口连接,操作完之后调用 GC\_close\_link()函数关闭串口连接。
- 参见: GC\_close\_link
- 函数名: GC\_close\_link
- 目 的:用 GC\_close\_link 函数关闭串口连接。
- 语 法: int GC\_close\_link (int ch);

ch: 串口通道号, 一般为1或2, 对应 PC 上的 COM1 或 COM2

- 调用例子: GC\_close\_link (1);
- 描 述: 在调用 GC\_open\_link() 打开串口连接进行通讯操作后,必须调用该函 数关闭连接,否则再次打开连接将会失败。
- 返 回 值:如果调用成功,函数返回0,否则返回-1。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见: GC\_open\_link

### 11.3 文件编译和下载运行函数

该类函数主要用于设置下载文件模式,编译、下载加工文件或配置文件,以及 启动按键功能等等。相关函数有:

int GC\_compile\_file(char\* srcfile, char\* outfile); /\*编译加工文件或配置文件\*/ int GC\_download\_file(char \*filename); /\*下载编译后的加工文件或配置文件\*/ int GC\_on\_button(DWORD dwButtonID, int Mode); /\*启动按键功能\*/ int GC\_set\_download\_mode(int mode); /\*设置下载模式\*/

#### 函数名: GC\_compile\_file

- 目 的:用 GC\_compile\_file()函数编译加工文件或配置文件,编译后的文件必 须以.mol 为后缀。
- 语 法: int GC\_compile\_file(char\* srcfile, char\* outfile);
   srcfile: 纯文本文件格式的G代码加工文件或配置文件

outfile:编译后用于下载的文件,文件名必须以.mol为后缀

- 调用例子: GC\_compile\_file ("C:\\GFiles\\test.g","C:\\GFiles\\test.mol"); /\*将 test.g 文件编译成 test.mol 文件\*/
- 描 述: 该函数可以对加工文件或配置文件进行编译,加工文件和配置文件必须 严格按照要求的格式生成。
- 返回值:如果调用成功,GC\_compile\_file()函数返回0,否则返回非0。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

#### 函数名: GC\_download\_file

- 目 的:用 GC\_download\_file()函数下载编译后的加工文件或配置文件,编译 后的文件必须以.mol 为后缀。
- 语 法: int GC\_download\_file(char\* downfile);
   downfile: 编译后用于下载的文件,文件名只能由英文字母和数字组成,符合 8.3 长度限制,其后缀为.mol,即\*\*\*\*\*\*\*.mol,其中\*代表任意英文字母或 0~9 的数字;
- 调用例子: GC\_download\_file ("C:\\GFiles\\test.mol"); /\*将 test.mol 文件通过串口下 载到控制器\*/
- 描 述: 该函数的下载功能实现是通过串口进行,因此必须在该操作之前打开串 口;也可以用 U 盘下载而不用该函数实现下载功能。
- 返 回 值:如果调用成功,函数返回0,否则返回非0。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

#### 函数名: GC\_on\_button

- 目 的:用于控制面板按键功能调用。
- 语 法: int GC\_on\_button(unsigned int dwButtonID, int Mode);
   dwButtonID: 按键编号,从1开始,其中1~4 固定为:运行;暂停;恢复;停止;5号以上依次与配置文件中定义的按键功能对应;
   Mode: 按键状态,1:按下;0:松开;其对应功能可在配置文件中定义
- 调用例子: GC\_on\_button (5, 1); /\* 执行 5 号按键被按下的功能 \*/
- 返 回 值:如果调用成功,函数返回0,否则返回非0。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 参见:

#### 函数名: GC\_set\_download\_mode

- 目 的:设置串口下载过程中当指定下载的文件在控制器中已经存在同名文件时 的处理方式。
- 语 法: int GC\_set\_download\_mode(int mode); mode: 模式 0-允许下载同名文件; 1-覆盖同名文件; 2-不下载同名文件, 即控制器中仍然保留已经下载的文件;

调用例子: GC\_set\_download\_mode (1); /\* 设置为覆盖模式 \*/

- 描述:系统缺省为覆盖模式。应注意的若指定下载的文件名只能由英文字母和 数字组成,符合 8.3 长度限制,其后缀为.mol,即\*\*\*\*\*\*\*.mol,其中\* 代表任意英文字母或 0~9 的数字,若文件名长度超过 8.3 的限制,则系 统将自动截取前 8 个字节的字符和后缀.mol,因此对于前 8 个字节的字 符相同的两个文件名,系统将认为是同名文件。
- 返 回 值:如果调用成功,函数返回0,否则返回非0。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 参见:

### 11.4 下载文件的管理

该类函数主要用于对下载到控制器中的文件进行管理,如查询、删除、设置当 前运行文件等操作。相关函数有:

int GC\_get\_file\_num (int\* CurFileID, int\* FileNum); /\*获取已下载的文件数目和当前 打开的文件的文件号\*/

int GC\_open\_file(char fileID, unsigned int\* filelen, char\* filename); /\*根据文件号打开 文件并获取文件长度和文件名,同时该文件被设置为当前文件\*/ int GC\_del\_file(int FileID); /\*删除文件号对应的文件\*/ int GC\_del\_all\_file(); /\*删除已经下载到控制器的所有文件\*/

#### 函数名: GC\_get\_file\_num

- 目 的:用 GC\_get\_file\_num()函数获取已下载的文件数目和当前打开的文件的 文件号。
- 语 法: int GC\_get\_file\_num (int\* CurFileID, int\* FileNum);
   CurFileID: 返回当前打开的文件号
   FileNum: 返回已经下载到控制器的文件数目
- 调用例子: int CurFile, FileNum;
  - GC\_get\_file\_num (&CurFile, &FileNum);
- 描 述: 该函数可以获取已下载到控制器中的文件数目; 其中有一个当前打开的 文件, 启动加工即运行该文件, 该函数可以获取当前打开的文件号。
- 返 回 值:如果调用成功,函数返回0,否则返回非0。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

#### 函数名: GC\_open\_file

- 目 的:用 GC\_open\_file()根据文件号打开文件并获取文件长度和文件名,同 时该文件被设置为当前文件。
- 法: int GC\_open\_file(char fileID, unsigned int\* fileIen, char\* filename);
   fileID: 要打开的文件号
   fileIen: 返回该文件的大小

filename: 返回该文件的文件名

调用例子: int FileID=1, FileLen; char FileName[16];

GC\_open\_file (FileID, &FileLen, FileNum);

- 描 述: 该函数可以获取某文件号对应文件的文件大小和文件名,同时用于更改 当前打开文件为启动运行做准备。
- 返 回 值:如果调用成功,函数返回0,否则返回非0。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

函数名: GC\_del\_file

目 的:用 GC\_del\_file()删除某文件号对应的文件。

- 语 法: int GC\_del\_file(int FileID); fileID: 要删除的文件号
- 调用例子: GC\_del\_file(1); 删除文件号1对应的文件

描 述: 该函数可以删除已经下载到控制器中的文件,若要获知该文件号对应的 文件名等信息,则可调用 GC\_open\_file()函数取得。

- 返 回 值:如果调用成功,函数返回0,否则返回非0。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

#### 函数名: GC\_del\_all\_file

- 目 的:用 GC\_del\_all\_file()删除所有文件。
- 语 法: int GC\_del\_all\_file();
- 调用例子: GC\_del\_all\_file();
- 描述: 该函数可以删除已经下载到控制器中的所有文件。
- 返 回 值:如果调用成功,函数返回0,否则返回非0。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

### 11.5 版本查询函数

该类函数主要用于查询 MPC05 卡的 DSP、FPGA 和函数库的版本,以及 DSP 程序的应用类型。相关函数有:

DWORD GC_get_fmw_ver();	/*查询 DSP 版本号*/
DWORD GC_get_hdw_ver();	/*查询 FPGA 版本号*/
DWORD GC_get_dll_ver();	/*查询动态链接库的版本号*/
DWORD GC_get_fmw_type();	/*查询 DSP 的应用类型*/

#### 函数名: GC\_get\_fmw\_ver

- 目 的:用 GC\_get\_fmw\_ver 函数查询 DSP 程序的版本号。
- 语 法: DWORD GC\_get\_fmw\_ver ();
- 调用例子: dwVer = GC\_get\_fmw\_ver ();
- 描 述: 该函数查询 MPC05 卡 DSP 固件的版本号。
- 返 回 值:返回 DSP 的版本号。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

#### 函数名: GC\_get\_hdw\_ver

- 目 的:用 GC\_get\_hdw\_ver 函数查询 FPGA 程序的版本号。
- 语 法: DWORD GC\_get\_ hdw\_ver ();
- 调用例子: dwVer = GC\_get\_f hdw\_ver ();
- 描 述: 该函数查询 MPC05 卡 FPGA 的版本号。
- 返 回 值:返回 FPGA 的版本号。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注释:
- 参见:

#### 函数名: GC\_get\_dll\_ver

- 目 的:用 GC\_get\_dll\_ver 函数查询动态链接库的版本号。
- 语 法: DWORD GC\_get\_dll\_ver ();
- 调用例子: dwVer = GC\_get\_dll\_ver ();
- 描 述: 该函数查询 MPC05 卡动态链接库的版本号,即编译出来的加工文件或配 置文件的版本号。
- 返 回 值:返回动态链接库的版本号。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参 见**:**

#### 函数名: GC\_get\_fmw\_type

- 目 的:用 GC\_get\_fmw\_type 函数查询 DSP 程序的应用类型。
- 语 法: DWORD GC\_get\_fmw\_type ();
- 调用例子: dwType = GC\_get\_fmw\_type ();
- 描 述: 该函数查询 MPC05 卡 DSP 固件的应用类型。如返回 0x05010102 则表示 该版本 DSP 固件应用的类型为通用版本。
- 返 回 值:返回 DSP 的应用类型。
- 系 统: WINDOWS98, WINDOWS2000, WINDOWS XP
- 注 释:
- 参见:

## 12 错误处理

### 12.1 控制器指示灯显示错误代码

### 12.1.1运行状态指示灯的位置

在 MPC05/MC 板上有连续排列的 D1~D8 共 8 个 LED 指示灯,用于指示运行工 作状态以及显示错误代码:



### 12.1.2运行状态指示灯显示的信息

当控制器上电启动后, D8 常亮, 显示运行正常;

在正常工作情况下, D1~D4 分别显示 1~4 轴脉冲信号输出状态, 即当某轴正在 输出脉冲时, 对应指示灯亮, 脉冲发送完毕,则对应指示灭。

当系统出现错误后, D1~D8 显示编码后的错误代码, 其编码规则如下:

D1~D8 分别表示 8 个位的状态,组成一个字节,如 D1 亮,其余灯熄灭,表示 错误代码 0x01;如 D1~D4 熄灭,D5~D8 亮,则表示错误代码:0xf0。

错误代码	原因	解决办法
0xe0	当前要运行的文件与固件版本不	1) 重新下载固件
	匹配	2) 更新函数库后重新编译下载
		文件
0xe1	当前要运行的文件与固件适用控	1)重新下载固件
	制器类型不匹配	2) 更新函数库后重新编译下载
		文件
0xe2	当前要运行的加工文件与控制器	1)更新函数库后重新编译下载
	内保存的已下载配置文件不匹配	加工文件或配置文件
0xe3	当前要运行的加工文件与控制器	1)更新函数库后重新编译下载
	内保存的已下载配置文件适用控	加工文件或配置文件
	制器类型不匹配	
0xf0	当前下载的固件不能用于该控制	1)更换控制器
	器	
0xf1	配置文件错误,可能是从未下载过	1) 需要重新下载配置文件
	配置或配置与固件不匹配	
0xf2	下载的两个固件文件不匹配	1)重新下载固件

各错误代码含义如下:

()	当系统出现错误显示后,请重新上电,即可恢复到正常状态,之后可根据错误提示采取相应的解决方法
注意	

## 13 问题及解决方法

### 13.1 增加外接 U 盘读数据指示灯

### 13.1.1功能说明

由于机器安装完成后, MPC05GA 被装在机器内部, MPC05GA 控制器上的指示 灯无法看见, 当用 U 盘下载数据时, 若 U 盘上没有指示灯,则无法判断数据是否已 经下载完毕。

可以利用输出信号增加了外接U盘指示灯功能。用于将U盘读数据状态信号引出到机器面板上通过发光二极管来进行显示。

### 13.1.2实现方法

将 MPC05/MC 板上的 J2 (DB15) 接口中的 5 脚 U 盘指示灯信号和 2/3/4 脚 (5V 电源) 用导线引出,之间串接一个 1k Ω 电阻和发光二极管用于显示。

### 13.2 开机回原点功能的实现

### 13.2.1功能说明

在控制系统的开发过程中,往往需要在系统上电后自动回机械原点。

### 13.2.2实现方法

该功能可采用如下方法实现:

在配置文件中参数段定义如下参数:

#### OnButtonPowerOn = 13

系统上电后执行该参数指定按键的功能代码,该按键对应的功能代码在配置文件的按键功能定义段实现。

即若该参数为13,则按键功能定义段应该有如下内容:

Button013_OnClick = {	
M90 Q00 B1 D2 E3	; X、Y、Z 轴分别对应 1、2、3 轴
M90 Q14 S0	; 常速回原点
M90 Q13 C1 2000	; 各轴常速度

M90 Q13 C2 2000 M90 Q13 C3 2000 G29 X1 Y1 Z1 G91 G01 X-1000 Y-2000 M90 Q09 C1 S0 M90 Q09 C2 S0 M90 Q09 C3 S0

; XYZ 轴正向回原点

; 增量编程方式

; X、Y 负向运动一段距离

;将该位置作为0点位置,以后的绝对编程定位均相对于该点

M99

}

按以上内容修改配置文件并重新编译、下载运行生效,然后重新上电即可实现 开机回原点。

若要取消开机回原点,只需要将 OnButtonPowerOn 参数设为 0 然后重新编译、 下载运行生效并重新上电即可。