

1. 概述

微数控 M3C 多任务运动控制器，是我司第三代通用功能类控制系统，之前我司产品只能执行单线程任务，即必须等待当前动作执行完毕才能执行下一动作，现在既能执行单线程任务，又能多个线程任务并行执行，最多可支持 20 个任务同时运行，全触摸操作，既能四轴联动，又支持非联动轴运动，编程简单，上手容易，极大的满足了用户对复杂动作的需求，广泛用于工业控制、医疗电子、节能环保、智能交通、电力系统、通讯系统、纺织行业、数控行业、汽车电子、车载设备、通信设备、网络终端等领域。欢迎广大用户定制专机，曾为客户做过八轴运动控制器。



1.1 产品特点

- 丨 中文指令，填表式编程，简单易上手
- 丨 全触控操作
- 丨 界面实时显示输入、输出状态
- 丨 自动模式实时修改程序参数
- 丨 支持掉电记忆功能，断电后坐标显示和工件计数不会清零
- 丨 支持多线程任务编程，支持 20 个线程任务同时进行
- 丨 支持串行、并行任务
- 丨 支持开机回机械零，手动回机械零，程序指令回机械零
- 丨 支持时间锁机，三级锁机模式
- 丨 定时器指令，M 寄存器指令（帮助实现多逻辑复杂动作）

- 支持外部输入口控制输出口状态
- 限位保护，支持硬限位和软限位
- 配合伺服电机、步进电机和 I/O 口实现自动执行、手动操作
- 任意输入口能定义启动、暂停、报警、限位等功能，外接开关

1.2 系统组成

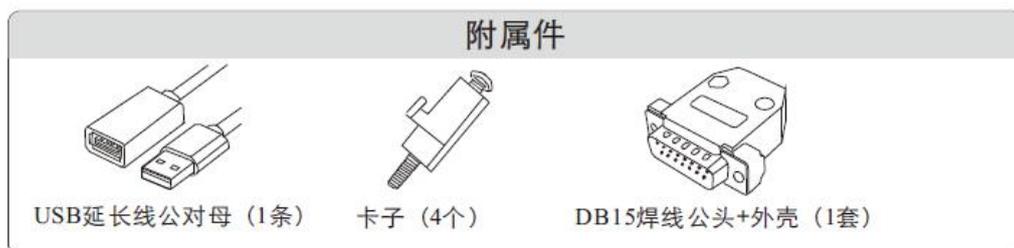
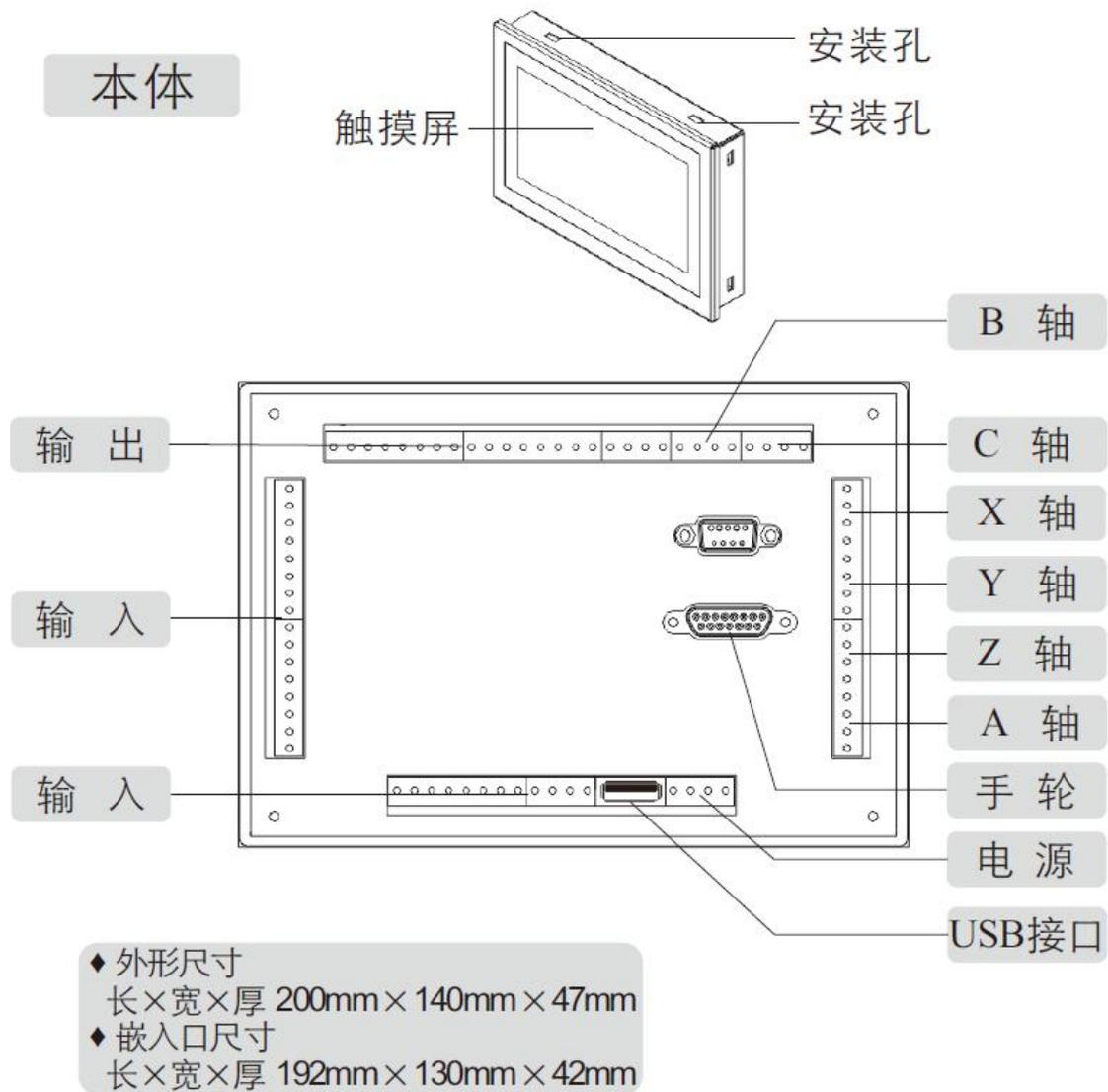
控制系统主要由以下几部分组成：

- 高精度、高精度进口 CPU
- 专门的运动控制芯片（信号 5V 脉冲+方向）
- 7 英寸彩色触摸显示屏（分辨率：800×480）
- 输入/输出（28 路光电隔离 24V 输入，18 路光电隔离 24V 输出，单通道额定电流 500mA）
- 带有一路标准 MPG 接口，外接手脉
- 壳体合金材料，坚固耐用，外观简单大方

1.3 技术指标

- 最小数据单位：0.001mm（0.001 度或 0.001 圈）
- 最大数据尺寸：±99999.999mm
- 支持 46 个中文编程指令
- 每台设备最多存储 12 个程序文件，每个程序文件可同时进行 20 个线程任务，单个线程任务最多编写 512 行程序
- 最高脉冲输出频率：350K
- 进给轴最大移动速度为 21000mm/min
- 控制轴数：1-4 轴（X, Y, Z, A）
- 联动轴：X, Y, Z, A 轴直线插补，X, Y 轴做圆弧插补
- 非联动轴：X, Y, Z, A 轴
- 进给速度倍率：10%-500%可调，通过自动模式和手动模式 F（显示进给速度）下方的“+”“-”调节，增量为 10%
- 电子齿轮：分子：1-99999, 分母：1-99999
- 系统主要功能：自动、手动、程序、参数、I0 等
- 安全保护：过压、过流、防反接
- 进度补偿：反向间隙补偿

1.4 外观及面板



1.5 接线上电

(1) 电源供电

系统采用双隔离供电方式，系统和 IO（输入、输出）分开供电

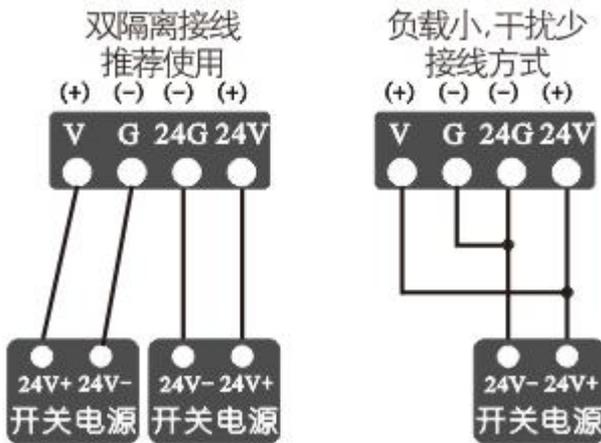
[V] 和 [G] 为 IO 供电，[V] 接 24V 开关电源正极，[G] 接 24V 开关电源负极。

[24V] 和 [24G] 为系统供电，[24V] 接 24V 开关电源正极，[24G] 接 24V 开关电源负极。

注意：严禁电源正负极接反

禁止过压、过载

禁止带电操作



(2) 接驱动器电机

系统最多提供 6 路 5V 脉冲输出（能接 6 套驱动器电机），各脉冲输出口接线方式一样，都是脉冲+方向

[P+]（脉冲正向输出）接驱动器 PUL+

[P-]（脉冲正向输出）接驱动器 PUL-

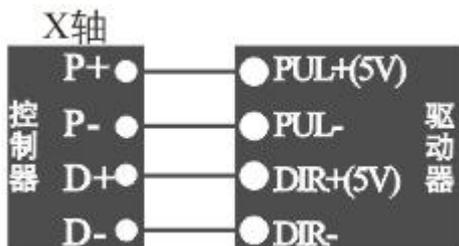
[D+]（脉冲正向输出）接驱动器 DIR+

[D-]（脉冲正向输出）接驱动器 DIR-

注意：

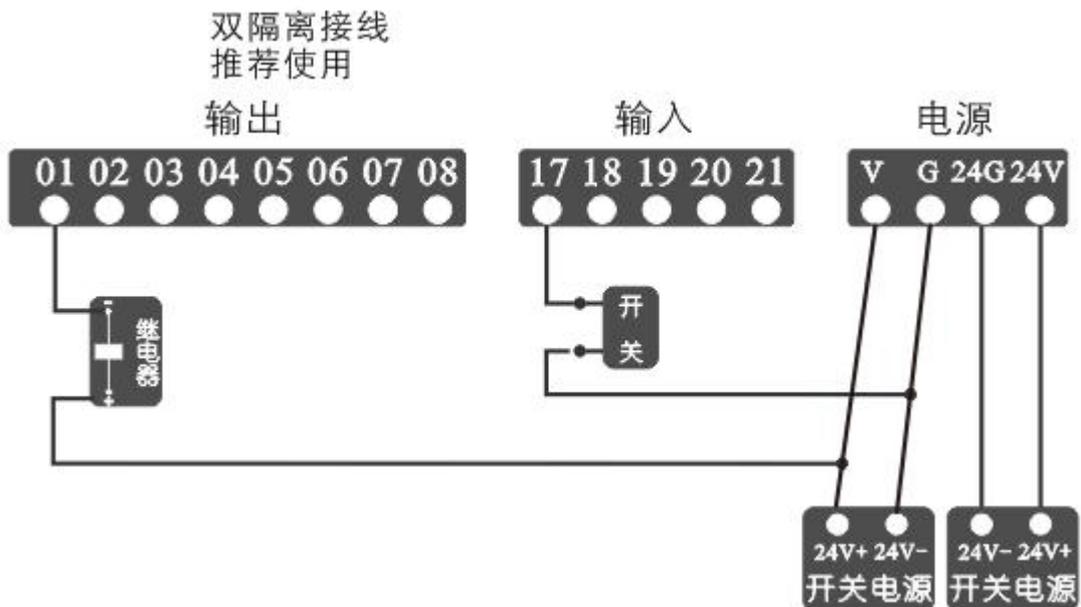
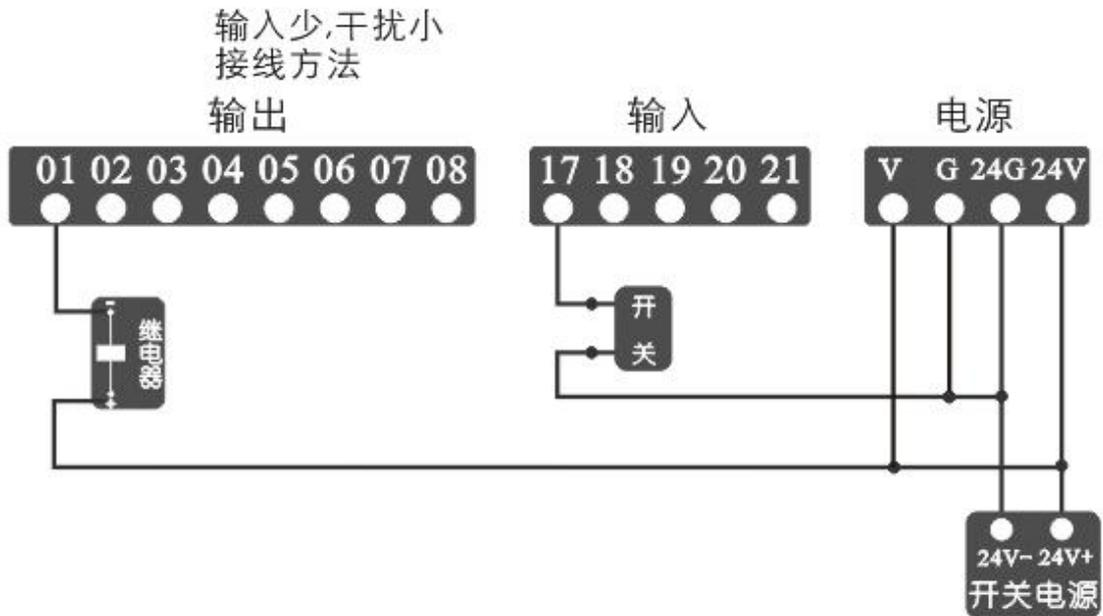
严禁将电源端子插入脉冲口

脉冲+方向接线方式用于伺服的位置模式
控制系统本身提供 5V 脉冲输出

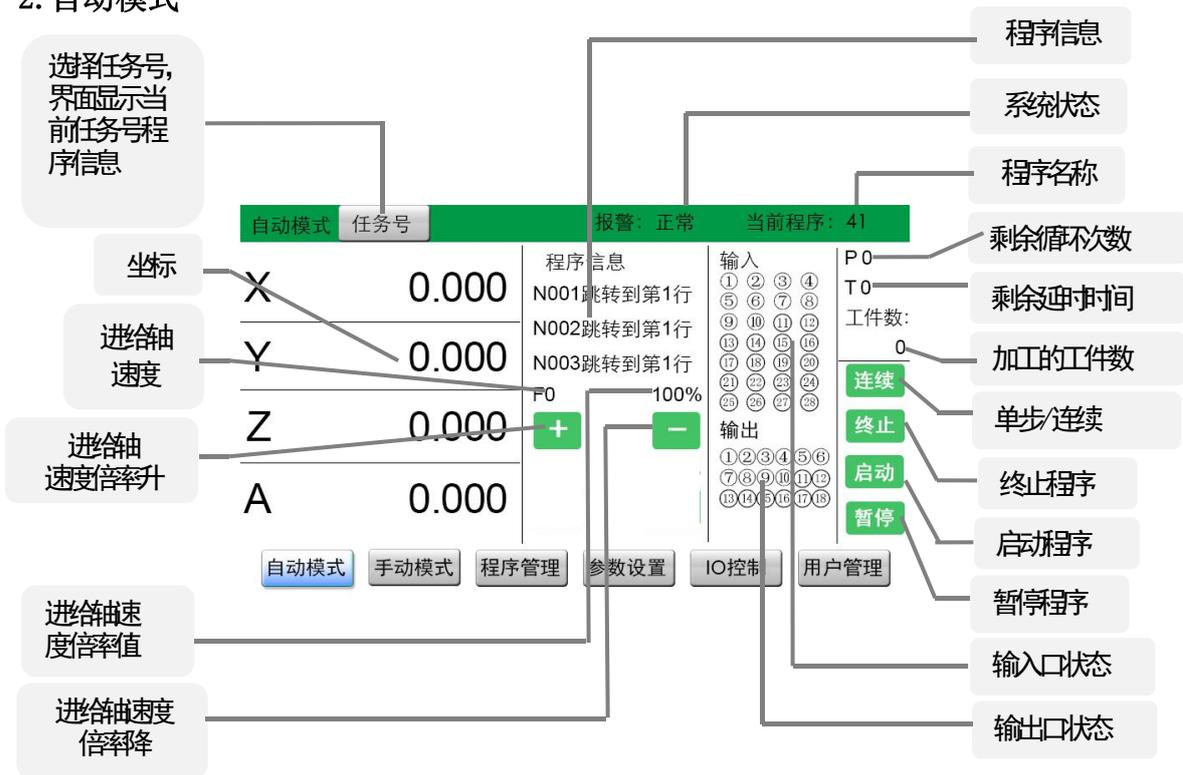


(3) 输入、输出接线

系统拥有 28 路输入，18 路输出，接线方式参考下图。



2. 自动模式

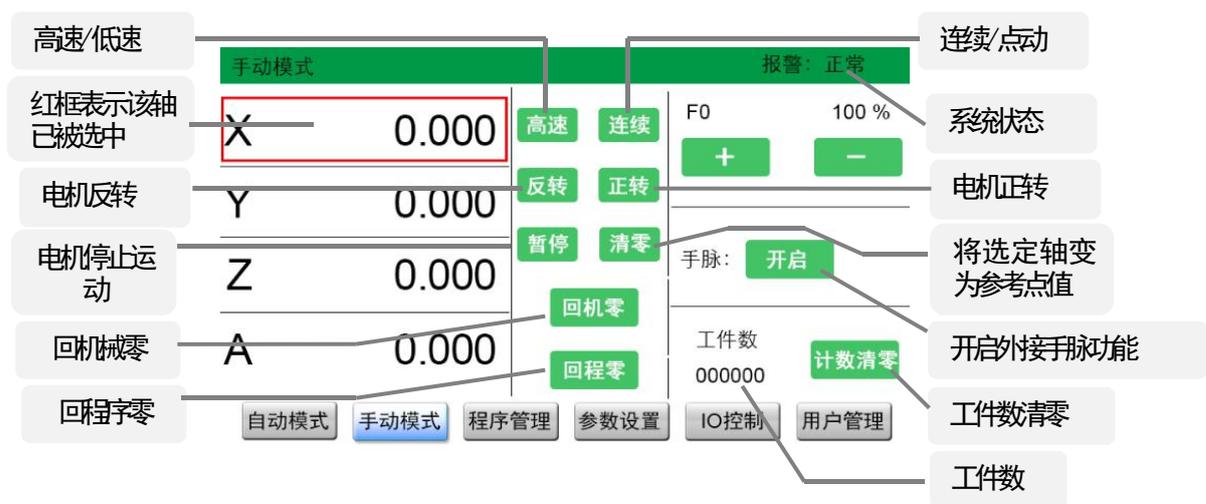


自动模式下, 按启动按钮, 程序开始运行, 运行的程序为最后一次打开的程序文件, 暂停按钮有效

连续模式: 连续时, 为连续加工; 按连续按钮变为单步显示时, 按一次启动, 执行一行指令

终止程序: 程序运行时, 按下终止按钮程序运行终止, 并跳到程序第一行

3. 手动模式



手动高速: 高速显示时以高速运动, 按高速按钮低速显示时, 以低速运动, 参数设置-速度中可设置

点动操作：按连续按钮点动显示时，以设定值为步长，按一次方向键（指正转、反转按钮）运动一次，参数-控制中可设置

正转：按下电机正向运动（与高速/低速，连续/点动配合）

反转：按下电机反向运动（与高速/低速，连续/点动配合）

清坐标零：按清零按钮将选择的轴坐标变为参考点值

回机械零：按回机零按钮让选中的轴返回机械零点，方向在参数-输入可设置

回程序零：按回程零按钮让选择的轴返回程序零点

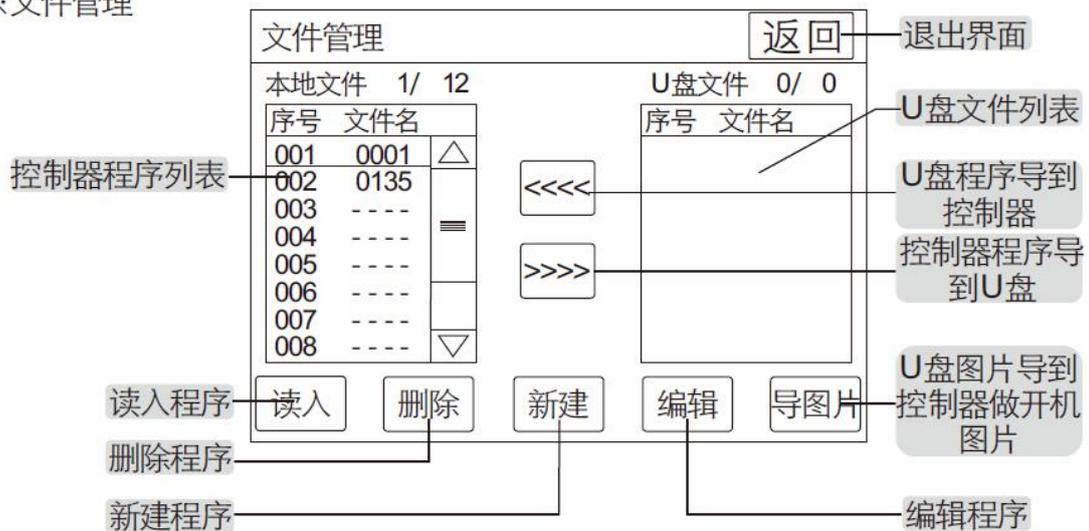
开启手脉：按开启按钮，开启外接手脉功能

计数清零：按下，工件数变为零

4. 程序管理

系统最多可储存 12 个程序文件，点击本地文件下方的加工程序文件名，选中文件

※文件管理



程序读入：按读入按钮读入系统中保存的加工程序

删除程序：按删除按钮删除光标所在位置的程序文件

程序新建：按新建按钮新建一个不带文件名的程序

程序编辑：按编辑按钮编辑已读入的加工程序

插入 U 盘，U 盘文件下方区域显示 U 盘中的文件。

导出程序：选中系统中的加工程序，按>>>>按钮，输入文件名，将控制系统中的文件导到 U 盘中

点击 U 盘的文件名，选中文件。

导入开机图片；选中格式合适的开机图片，然后按下导图片按钮。

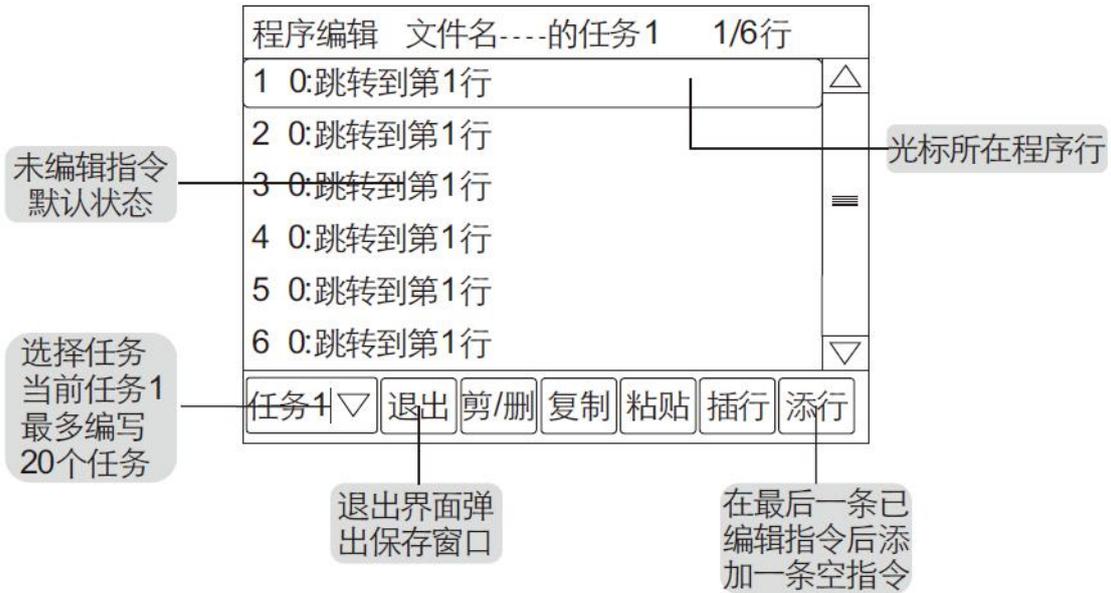
图片名称格式要求：像素为 800*480，尾缀为 bmp

导入程序：选中 U 盘中的加工程序，按<<<<按钮，将 U 盘程序导到控制系统中

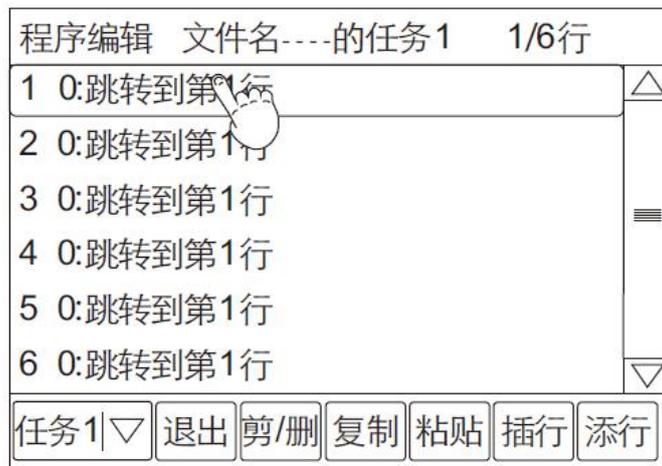
4.1 程序编辑

程序新建和程序编辑都可以进入该界面，现在以程序新建为例。

※程序编辑界面



※编辑一条指令 M 设定 M1=0



加工指令			返回
联动相对 (等待)	联动连续模式	输出	M运算
联动绝对 (等待)	非联动速度	工件置数	定时器比较
联动快速 (等待)	非联动相对	工件计数	定时器启动
回机械零 (等待)	非联动绝对	判断跳转	定时器存入M
联动顺圆 (等待)	非联动缓停	绝对跳转	子程调用
联动逆圆 (等待)	非联动急停	循环	子程开始
顺圆I (等待)	非联动判OK	机械零判OK	子程结束
逆圆I (等待)	非联动等待OK	跳转到第1行	结束
延时	坐标比较	空指令	暂停
联动相对	设定坐标	M比较	回机械零
联动绝对	联动倍率设定	M设定	
联动判OK	非联动倍率设定	M逻辑	

程序编辑 文件名----的任务1 1/6行

1	0:M设定 M□ = 0	△
2	0:跳转到第1行	
3	0:跳转到第1行	≡
4	0:跳转到第1行	
5	0:跳转到第1行	
6	0:跳转到第1行	▽

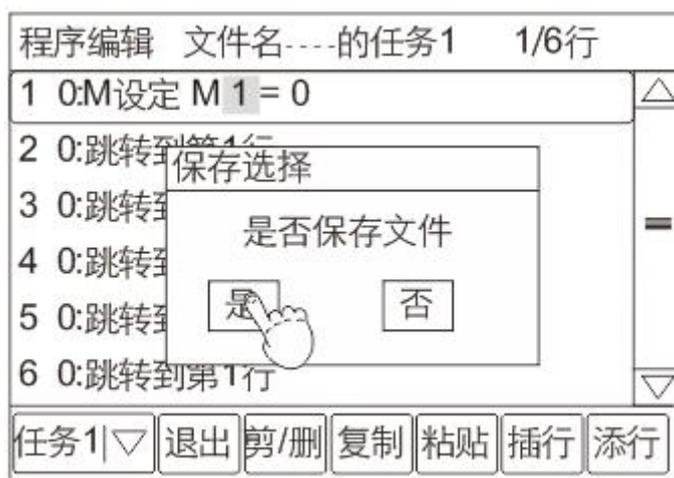
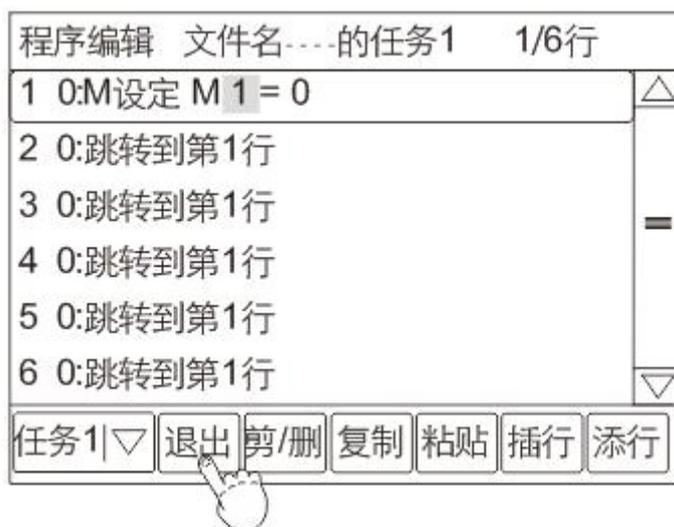
任务1 ▾ 退出 剪/删 复制 粘贴 插行 添行

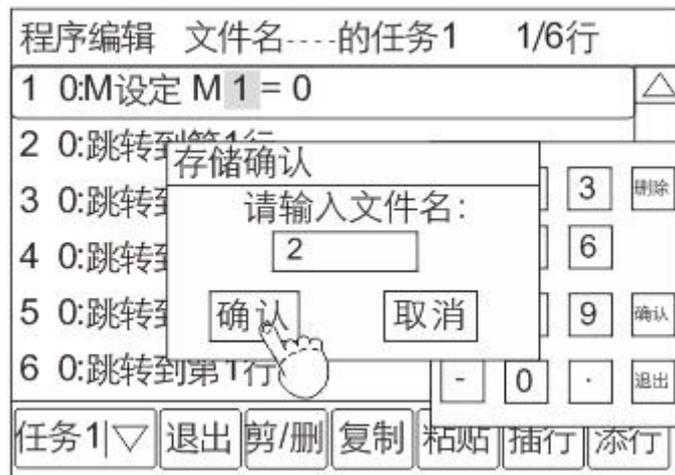
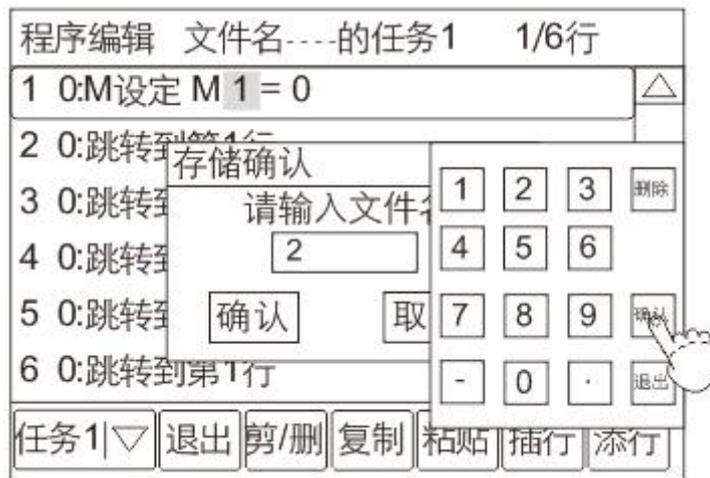
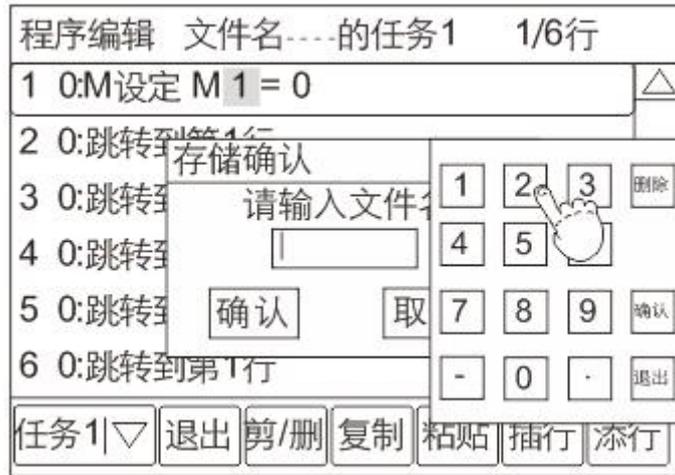
1	0:M设定 M□ = 0	△
2	0:跳转到第1行	
3	0:跳转到第1行	≡
4	0:跳转到第1行	
5	0:跳转到第1行	
6	0:跳转到第1行	▽

M	参	无效	←	→	1	2	3	4	5	-	▽
T	子	清空	删除	6	8	9	0	+			

退出软键盘

※保存程序





5. 用户管理

需要设置参数时，首先要进入用户管理界面，登陆用户密码或厂商密码后，才能修改参数。不同功能，用不同的密码登陆。用户管理里有修改系统时间，用户各级密码和时间锁机的功能

※未登陆状态

用户管理 返回

用户密码 显示密码

厂商密码

组态密码

版本信息
软件: CHA-V1.20
软件: MAIN-V2.0

用户登陆 用户退出 时间显示

用户密码：登陆用户密码，设置用户级使用参数，默认出厂密码为：0

※登陆用户密码

用户管理 返回

用户密码 显示密码

厂商密码

版本信息
软件:CHA-V1.20
硬件:MAIN-V2.0

用户登陆 用户退出 时间显示

用户管理 返回

用户密码 显示密码

厂商密码

1 2 3 删除
4 5 6
7 8 9 确认
- 0 . 退出

用户登陆 用户退出 时间显示

用户管理 返回

用户密码* 显示密码

厂商密码

1 2 3 删除
 4 5 6
 7 8 9 清除
 - 0 . 退出

用户管理 返回

用户密码* 显示密码

厂商密码

版本信息
 软件:CHA-V1.20
 硬件:MAIN-V2.0

用户管理 返回

用户已登录 显示密码

版本信息
 软件:CHA-V1.20
 硬件:MAIN-V2.0

厂商密码：登陆厂商密码，设置厂商级使用功能，时间锁机，厂商密码暂不告知用户

登陆各级密码的方法：

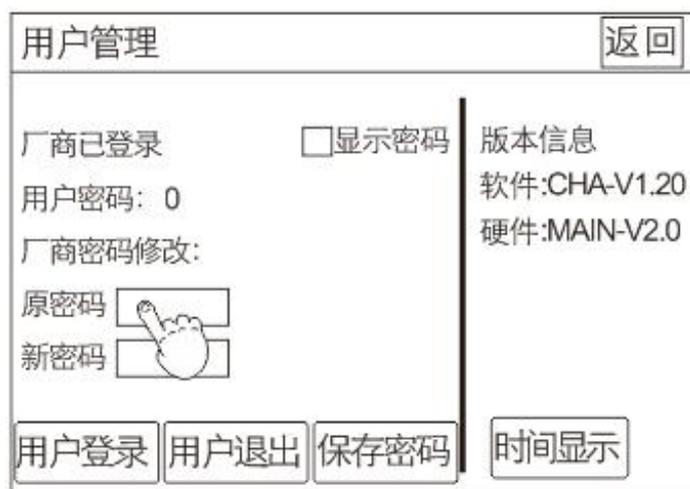
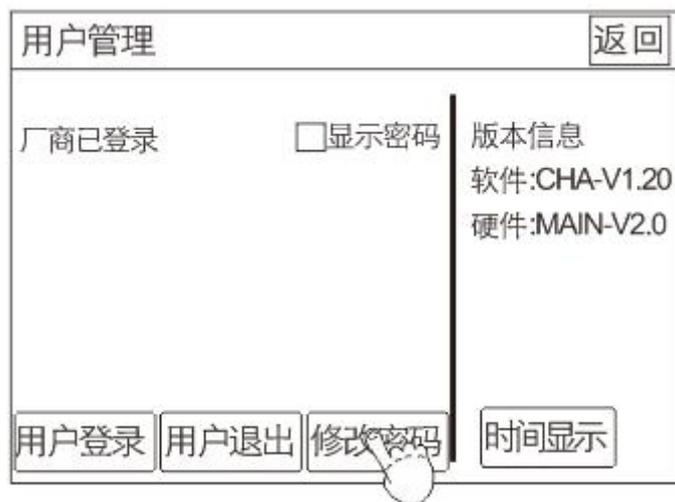
点击用户管理界面密码旁边的框→弹出的数字窗中按顺序输入密码数字→按确

认按钮→按用户登陆按钮

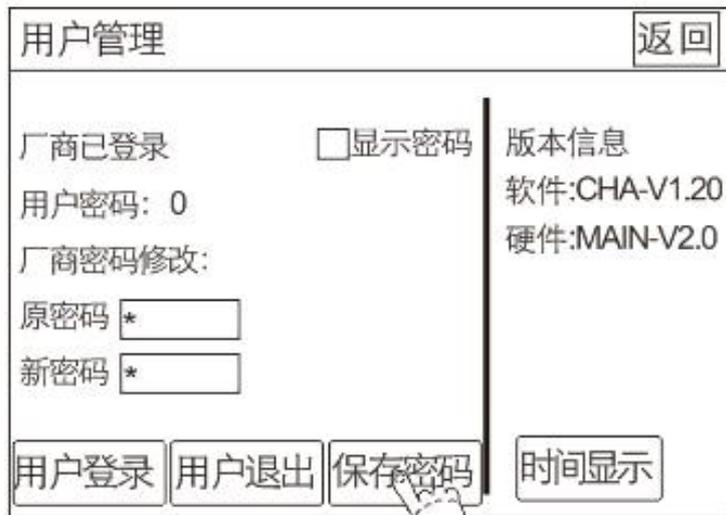
修改各级密码的方法：

进入已登陆用户的用户管理界面（想修改那个密码就登陆那个账号）→按修改密码按钮→输入原密码和新密码→按保存密码按钮

※修改厂商密码

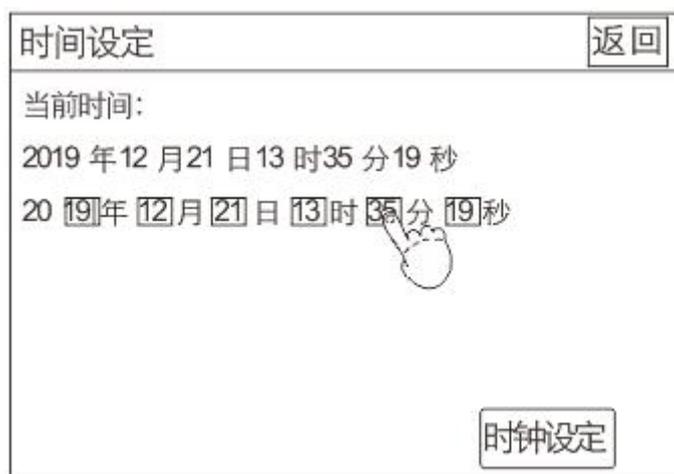
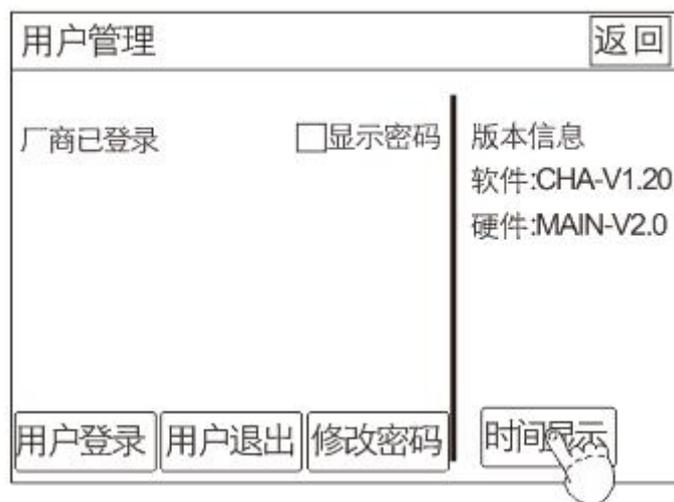


..... (略)



修改系统时间的方法:

登陆厂商密码→在参数设置-速度中找到时间锁机，保证此参数功能无效（未打对勾），退出时保存→用户管理界面→按时间显示按钮



..... (略)

时间设定	返回
当前时间: 2019年12月21日13时35分19秒 20 19年 12月 21日 13时 34分 19秒	
<input type="button" value="时间设定"/>	

用户退出的方法:

进入已登陆用户的用户管理界面→按用户退出按钮

时间锁机的方法

登陆厂商密码→在参数设置-速度中找到时间锁机，保证此参数功能有效（打对勾），退出时保存→用户管理界面→按时间显示按钮

※设定时间锁机 1

用户管理	返回
厂商已登录	<input type="checkbox"/> 显示密码
版本信息 软件:CHA-V1.20 硬件:MAIN-V2.0	
<input type="button" value="用户登录"/>	<input type="button" value="用户退出"/>
<input type="button" value="修改密码"/>	<input type="button" value="时间显示"/>

锁机设定	返回
当前时间: 2019年12月21日13时35分19秒	
锁机时间1: 20 11年 1月 1日	期限密码1: 1
锁机时间2: 20 11年 1月 1日	期限密码2: 2
锁机时间2: 20 11年 1月 1日	期限密码2: 0
	<input type="button" value="锁机1设定"/>
	<input type="button" value="锁机2设定"/>
	<input type="button" value="锁机3设定"/>

..... (略)

锁机设定		返回
当前时间: 2019年12月21日13时35分19秒		
锁机时间1: 2019年12月23日	期限密码1: 888888	锁机1设定
锁机时间2: 2011年1月1日	期限密码2: 2	锁机2设定
锁机时间2: 2011年1月1日	期限密码2: 0	锁机3设定

锁机设定		返回
当前时间: 2019年12月21日13时35分19秒		
锁机时间1: 2019年12月23日	期限密码1: 888888	锁机1设定
锁机时间2: 2011年1月1日	期限密码2: 2	锁机2设定
锁机时间2: 2011年1月1日	期限密码2: 0	锁机3设定

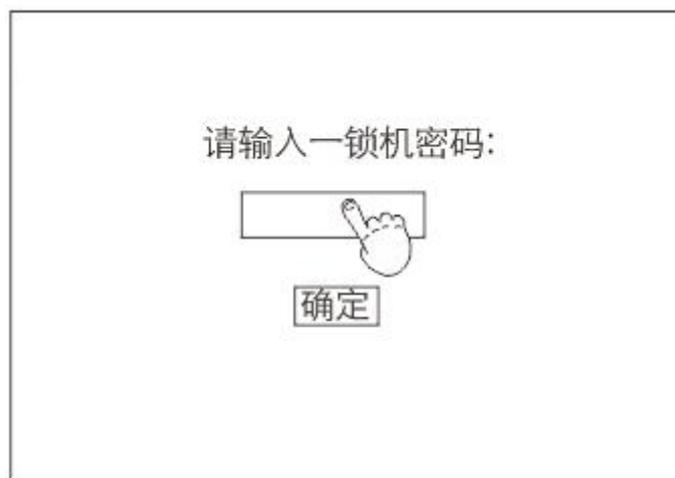
密码保存

是否保存密码

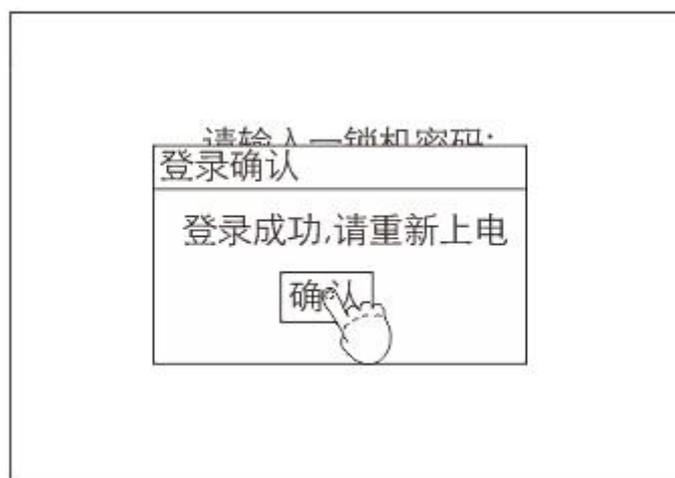
是 否

锁机设定		返回
当前时间: 2019年12月21日13时35分19秒		
锁机时间1: 2019年12月23日	期限密码1: 888888	锁机1已设
锁机时间2: 2011年1月1日	期限密码2: 2	锁机2设定
锁机时间2: 2011年1月1日	期限密码2: 0	锁机3设定

※解除时间锁机设定（到达锁机时间）



..... (略)



其余两个时间锁机/解锁设置方式同上。

6. 参数设置

系统正常工作时参数的设置，包括控制参数、速度参数、输入参数、输出参数和程序参数。进入界面需先登录用户

6.1 控制参数

控制参数		返回
升速时间:	<input type="text" value="200"/>	Y轴分子: <input type="text" value="1"/>
点动距离:	<input type="text" value="0.100"/>	Y轴分母: <input type="text" value="1"/>
X轴参考点:	<input type="text" value="0.000"/>	Z轴参考点: <input type="text" value="0.000"/>
X轴分子:	<input type="text" value="1"/>	Z轴分子: <input type="text" value="1"/>
X轴分母:	<input type="text" value="1"/>	Z轴分母: <input type="text" value="1"/>
Y轴参考点:	<input type="text" value="1"/>	A轴参考点: <input type="text" value="0.000"/>
控制		速度
输入		输出
程参		上一页
		下一页
		1/4页

升速时间：电机升速时间，电机将以启动速度运行经过设定的升速时间后，运行到设定的F值速度（单位：毫秒）

点动距离：手动模式下，点动增量数值

X轴参考点：手动模式下，按下清零按钮，将当前坐标清空并显示为此设定数值，或回机械零碰到开关停止时显示此设定数值

X轴分子：X轴电子齿轮分子（取值范围 1-99999）

X轴分母：X轴电子齿轮分母（取值范围 1-99999）

Y轴参考点：手动模式下，按下清零按钮，将当前坐标清空并显示为此设定数值，或回机械零碰到开关停止时显示此设定数值

Y轴分子：Y轴电子齿轮分子（取值范围 1-99999）

Y轴分母：Y轴电子齿轮分母（取值范围 1-99999）

Z轴参考点：手动模式下，按下清零按钮，将当前坐标清空并显示为此设定数值，或回机械零碰到开关停止时显示此设定数值

Z轴分子：Z轴电子齿轮分子（取值范围 1-99999）

Z轴分母：Z轴电子齿轮分母（取值范围 1-99999）

A轴参考点：手动模式下，按下清零按钮，将当前坐标清空并显示为此设定数值，或回机械零碰到开关停止时显示此设定数值

A轴分子：A轴电子齿轮分子（取值范围 1-99999）

A轴分母：A轴电子齿轮分母（取值范围 1-99999）

X 轴间隙：X 轴反向间隙值（根据设备实际测量获得）

Y 轴间隙：Y 轴反向间隙值（根据设备实际测量获得）

Z 轴间隙：Z 轴反向间隙值（根据设备实际测量获得）

A 轴间隙：C 轴反向间隙值（根据设备实际测量获得）

X 轴回零模式：回机械零的两种方式， 经过开关， 不过开关

Y 轴回零模式：回机械零的两种方式， 经过开关， 不过开关

Z 轴回零模式：回机械零的两种方式， 经过开关， 不过开关

A 轴回零模式：回机械零的两种方式， 经过开关， 不过开关

X 轴软限位使能： 功能有效， 功能无效，决定系统是否开启软限位功能

Y 轴软限位使能： 功能有效， 功能无效，决定系统是否开启软限位功能

Z 轴软限位使能： 功能有效， 功能无效，决定系统是否开启软限位功能

A 轴软限位使能： 功能有效， 功能无效，决定系统是否开启软限位功能

X 轴正限位：X 轴软限位的正向坐标

Y 轴正限位：Y 轴软限位的正向坐标

Z 轴正限位：Z 轴软限位的正向坐标

A 轴正限位：A 轴软限位的正向坐标

X 零开机启动： 功能有效， 功能无效，决定开机时系统优先运行回机械零动作

Y 零开机启动： 功能有效， 功能无效，决定开机时系统优先运行回机械零动作

Z 零开机启动： 功能有效， 功能无效，决定开机时系统优先运行回机械零动作

A 零开机启动： 功能有效， 功能无效，决定开机时系统优先运行回机械零动作

开机回零顺序 1：多轴回机械零时，决定第一个回机械零的轴，0 X 轴，1 Y 轴，2 Z 轴，3 C 轴

开机回零顺序 2：多轴回机械零时，决定第二个回机械零的轴，0 X 轴，1 Y 轴，2 Z 轴，3 C 轴

开机回零顺序 3：多轴回机械零时，决定第三个回机械零的轴，0 X 轴，1 Y 轴，2 Z 轴，3 C 轴

开机回零顺序 4：多轴回机械零时，决定第四个回机械零的轴，0 X 轴，1 Y 轴，

2 Z 轴，3 C 轴

手脉增量：外接手脉移动的最小距离，单位微米

6.2 速度参数

速度参数		返回
启动速度:	<input type="text" value="200"/>	Y轴手动高速: <input type="text" value="1000"/>
XYZA最高速:	<input type="text" value="2000"/>	Y轴手动低速: <input type="text" value="500"/>
回零高速:	<input type="text" value="1000"/>	Z轴手动高速: <input type="text" value="1000"/>
回零低速:	<input type="text" value="500"/>	Z轴手动低速: <input type="text" value="500"/>
X轴手动高速:	<input type="text" value="1000"/>	A轴手动高速: <input type="text" value="1000"/>
X轴手动低速:	<input type="text" value="500"/>	A轴手动低速: <input type="text" value="500"/>
控制		速度
输入		输出
程参		上一页
		下一页 <small>1/2 页</small>

启动速度：电机启动时，会以此速度运行，在升速时间内运行至程序设定中的 F 值速度

XYZA 最高速：系统运行时，合成速度 F 不会超过此速度

回零高速：回机械零时的高速，回机械零时，首先以回零高速运行

回零低速：回机械零时的低速，回机械零时，首先以回零高速运行，碰到机械零开关后再以回零低速运行

X 轴手动高速：手动模式下，X 轴电机高速时的速度

X 轴手动低速：手动模式下，X 轴电机低速时的速度

Y 轴手动高速：手动模式下，Y 轴电机高速时的速度

Y 轴手动低速：手动模式下，Y 轴电机低速时的速度

Z 轴手动高速：手动模式下，Z 轴电机高速时的速度

Z 轴手动低速：手动模式下，Z 轴电机低速时的速度

A 轴手动高速：手动模式下，A 轴电机高速时的速度

A 轴手动低速：手动模式下，A 轴电机低速时的速度

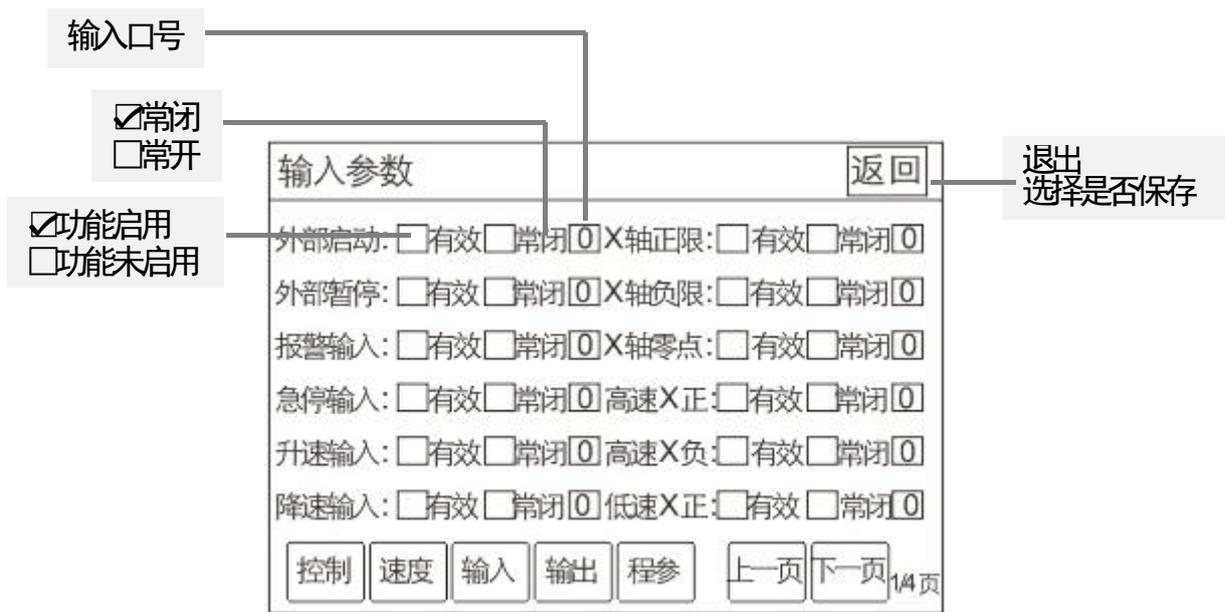
6.3 输入参数

外部启动：外部开关启动程序功能的设定

外部暂停：外部开关暂停功能的设定

报警输入：报警信号的输入设定

急停输入：急停信号的输入设定



升速输入：外部开关进给轴速度倍率升速按钮设定

降速输入：外部开关进给轴速度倍率降速按钮设定

X轴正限：X轴的正限位设定

Y轴负限：Y轴的负限位设定

X轴零点：X轴零点开关的设定

高速X正：X轴手动高速正向转动外部开关的设定

高速X负：X轴手动高速反向转动外部开关的设定

低速X正：X轴手动低速正向转动外部开关的设定

低速X负：X轴手动低速反向转动外部开关的设定

X零启动：外部开关启动X轴回机械零功能设定

Y轴正限：Y轴的正限位设定

Y轴负限：Y轴的负限位设定

Y轴零点：Y轴零点开关的设定

高速Y正：Y轴手动高速正向转动外部开关的设定

高速Y负：Y轴手动高速反向转动外部开关的设定

低速Y正：Y轴手动低速正向转动外部开关的设定

低速Y负：Y轴手动低速反向转动外部开关的设定

Y零启动：外部开关启动Y轴回机械零功能设定

Z轴正限：Z轴的正限位设定

Z轴负限：Z轴的负限位设定

Z 轴零点：Z 轴零点开关的设定

高速 Z 正：Z 轴手动高速正向转动外部开关的设定

高速 Z 负：Z 轴手动高速反向转动外部开关的设定

低速 Z 正：Z 轴手动低速正向转动外部开关的设定

低速 Z 负：Z 轴手动低速反向转动外部开关的设定

Z 零启动：外部开关启动 Z 轴回机械零功能设定

A 轴正限：A 轴的正限位设定

A 轴负限：A 轴的负限位设定

A 轴零点：A 轴零点开关的设定

高速 A 正：A 轴手动高速正向转动外部开关的设定

高速 A 负：A 轴手动高速反向转动外部开关的设定

低速 A 正：A 轴手动低速正向转动外部开关的设定

低速 A 负：A 轴手动低速反向转动外部开关的设定

A 零启动：外部开关启动 A 轴回机械零功能设定

6.4 输出参数

输出参数		返回
输出01: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	输出07: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	
输出02: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	输出08: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	
输出03: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	输出09: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	
输出04: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	输出10: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	
输出05: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	输出11: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	
输出06: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	输出12: <input type="checkbox"/> 有效 <input type="checkbox"/> 通	

控制 速度 输入 输出 程参 上一页 下一页 1/4页

自动模式下，设置程序运行暂停时输出口的状态，重新启动程序后，输出口恢复成暂停之前的状态

6.5 程序参数

事先在程序里指定的位置写上参 X，该界面的功能是设置参 X 代表的实际数值，适用于程序行数多，需要经常修改程序里某项指令参数。

程序参数				返回
参1:	<input type="text" value="0.000"/>	参7:	<input type="text" value="0.000"/>	
参2:	<input type="text" value="0.000"/>	参8:	<input type="text" value="0.000"/>	
参3:	<input type="text" value="0.000"/>	参9:	<input type="text" value="0.000"/>	
参4:	<input type="text" value="0.000"/>	参10:	<input type="text" value="0.000"/>	
参5:	<input type="text" value="0.000"/>	参11:	<input type="text" value="0.000"/>	
参6:	<input type="text" value="0.000"/>	参12:	<input type="text" value="0.000"/>	
<input type="button" value="控制"/> <input type="button" value="速度"/> <input type="button" value="输入"/> <input type="button" value="输出"/> <input type="button" value="程参"/>				<input type="button" value="上一页"/> <input type="button" value="下一页"/>

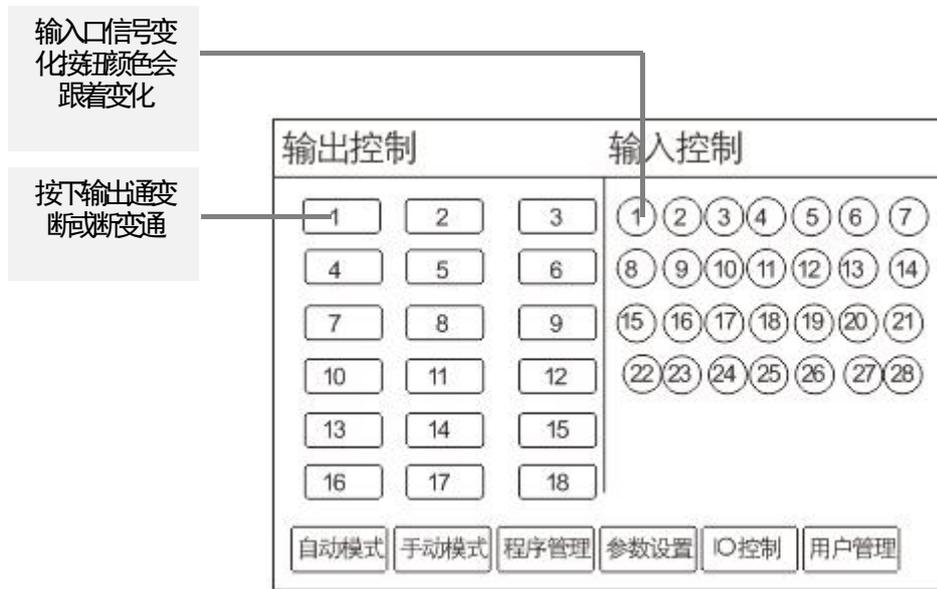
※退出并保存参数

程序参数				返回
参1:	<input type="text" value="0.000"/>	参7:	<input type="text" value="0.000"/>	
参2:	<input type="text" value="0.000"/>	参8:	<input type="text" value="0.000"/>	
参3:	<input type="text" value="0.000"/>	参9:	<input type="text" value="0.000"/>	
参4:	<input type="text" value="0.000"/>	参10:	<input type="text" value="0.000"/>	
参5:	<input type="text" value="0.000"/>	参11:	<input type="text" value="0.000"/>	
参6:	<input type="text" value="0.000"/>	参12:	<input type="text" value="0.000"/>	
<input type="button" value="控制"/> <input type="button" value="速度"/> <input type="button" value="输入"/> <input type="button" value="输出"/> <input type="button" value="程参"/>				<input type="button" value="上一页"/> <input type="button" value="下一页"/>

程序参数				返回
参1:	<input type="text" value="0.000"/>	参7:	<input type="text" value="0.000"/>	
参2:	参数保存		<input type="text" value="0.000"/>	
参3:	是否保存参数		<input type="text" value="0.000"/>	
参4:	<input type="button" value="是"/>	<input type="button" value="否"/>	<input type="text" value="0.000"/>	
参5:			<input type="text" value="0.000"/>	
参6:	<input type="text" value="0.000"/>	参12:	<input type="text" value="0.000"/>	
<input type="button" value="控制"/> <input type="button" value="速度"/> <input type="button" value="输入"/> <input type="button" value="输出"/> <input type="button" value="程参"/>				<input type="button" value="上一页"/> <input type="button" value="下一页"/>

7.10 设置

用来检测输入、输出功能是否正常



8. 程序指令

为便于使用，本控制器采用中文填表式编程，通过触摸选择操作指令，各指令采用固定程序格式，提示输入相应的程序数据，不需要可以不填

程序指令格式说明：



控制系统拥有 46 个指令，通过这些指令的配合使用，可实现串行、并行多任务动作

8.1 联动相对（等待）

走相对坐标（即每次运动的起点以上次运动的终点作为参考点），沿直线以合成速度 $F \times$ 速度倍率的速度运行，等待此指令执行完毕才能执行下一指令

1 0: 联动相对（等待） X 0.0 Y 0.0 Z 0.0 A 0.0 F 0

参数：X（X 轴相对坐标），Y（Y 轴相对坐标），Z（Z 轴相对坐标），A（A 轴相对坐标），F（运动速度）

8.2 联动绝对（等待）

走绝对坐标（即每次运动都以原点作为参考点），沿直线以合成速度 $F \times$ 速度倍率的速度运行，等待此指令执行完毕才能执行下一指令

1 0: 联动绝对（等待） X 0.0 Y 0.0 Z 0.0 A 0.0 F 0

参数：X（X 轴绝对坐标），Y（Y 轴绝对坐标），Z（Z 轴绝对坐标），A（A 轴绝

对坐标)，F（运动速度）

8.3 联动快速（等待）

走相对坐标（即每次运动的起点以上次运动的终点作为参考点），沿直线以系统设置的最高速×速度倍率的速度运行，等待此指令执行完毕才能执行下一条指令

```
G01 G00: 联动快速（等待） X 0.0 Y 0.0 Z 0.0 A 0.0
```

参数：X（X轴相对坐标），Y（Y轴相对坐标），Z（Z轴相对坐标），A（A轴相对坐标）

8.4 回机械零（等待）

根据设定好的轴和回零方向执行回机械零动作，等待此指令执行完毕才能执行下一条指令

```
G1 G00: 回机械零（等待） X 轴回机械零，负向
```

参数：X（选轴，单轴默认X），负向（回机械零方向）

8.5 联动顺圆（等待）

走相对坐标（即每次运动的起点以上次运动的终点作为参考点），沿顺时针方向以合成速度F×速度倍率的速度运动到给定的增量值，等待指令指令执行完毕才执行下一条指令

```
G1 G01: 联动顺圆（等待） X 0.0 Y 0.0 R 0.0 F 0
```

参数：X（X轴运动增量），Y（Y轴运动增量），R（圆的半径，R为负值，圆弧为劣弧，R为正值，圆弧为正值），F（运动速度）

8.6 联动逆圆（等待）

走相对坐标（即每次运动的起点以上次运动的终点作为参考点），沿逆时针方向以合成速度F×速度倍率的速度运动到给定的增量值，等待指令指令执行完毕才执行下一条指令

```
G1 G02: 联动逆圆（等待） X 0.0 Y 0.0 R 0.0 F 0
```

参数：X（X轴运动增量），Y（Y轴运动增量），R（圆的半径，R为负值，圆弧为劣弧，R为正值，圆弧为正值），F（运动速度）

8.7 联动顺圆 IJ（等待）

走相对坐标（即每次运动的起点以上次运动的终点作为参考点），沿顺时针方向以合成速度F×速度倍率的速度运动到给定的增量值，等待指令指令执行完毕才执行下一条指令

```
G1 G01: 联动顺圆 IJ（等待） X 0.0 Y 0.0 I 0.0 J 0.0 F 0
```

参数: X (X 轴方向圆弧终点相对于圆弧起点的增量值), Y (Y 轴方向圆弧终点相对于圆弧起点的增量值), I (圆心相对于圆弧起点 X 轴方向的增量值), J (圆心相对于圆弧起点 Y 轴方向的增量值), F (运动速度)

8.8 联动逆圆 IJ (等待)

走相对坐标 (即每次运动的起点以上次运动的终点作为参考点), 沿逆时针方向以合成速度 $F \times$ 速度倍率的速度运动到给定的增量值, 等待指令指令执行完毕才执行下一条指令

```
G 01 I 0.0 J 0.0 F 0.0
```

参数: X (X 轴方向圆弧终点相对于圆弧起点的增量值), Y (Y 轴方向圆弧终点相对于圆弧起点的增量值), I (圆心相对于圆弧起点 X 轴方向的增量值), J (圆心相对于圆弧起点 Y 轴方向的增量值), F (运动速度)

8.9 延时

执行到此指令, 按照设置的时间, 等待

```
G 04 P 0.0
```

参数: 0.0 (延时等待的时间)

8.10 联动相对

走相对坐标 (即每次运动的起点以上次运动的终点作为参考点), 沿直线以合成速度 $F \times$ 速度倍率的速度运行。不需要等待指令执行完毕直接执行下一条指令

```
G 01 X 0.0 Y 0.0 Z 0.0 A 0.0 F 0.0
```

参数: X (X 轴相对坐标), Y (Y 轴相对坐标), Z (Z 轴相对坐标), A (A 轴相对坐标), F (运动速度)

8.11 联动绝对

走绝对坐标 (即每次运动都以原点作为参考点), 沿直线以合成速度 $F \times$ 速度倍率的速度运行。不需要等待指令执行完毕直接执行下一条指令

```
G 00 X 0.0 Y 0.0 Z 0.0 A 0.0 F 0.0
```

参数: X (X 轴绝对坐标), Y (Y 轴绝对坐标), Z (Z 轴绝对坐标), A (A 轴绝对坐标), F (运动速度)

8.12 联动判 OK

判断联动轴是否停止运动, 停止运动执行下一条指令, 否则跳到目的标号处

```
G 01 X 0.0 Y 0.0 Z 0.0 A 0.0 F 0.0
```

参数: 0: (目的标号)

8.13 联动连续模式

选择有效时，从前一联动直线类指令到后一联动直线类指令不降速。选择禁止时，从前一联动直线类指令到后一联动直线类指令，需要先降速再升速

1 0: 联动判 OK 禁止

参数: 禁止 (选择禁止或者有效)

8.14 非联动速度

各轴以各自设定的速度同时运行，没有指定具体的位置，速度值前减号，负方向运动；无符号，正方向运动

1 0: 非联动速度 FX 0 FY 0 FZ 0 FA 0

参数: FX 0 (X 轴速度)，FY 0 (Y 轴速度)，FZ 0 (Z 轴速度)，FA 0 (A 轴速度)

8.15 非联动相对

走相对坐标（即每次运动的起点以上次运动的终点作为参考点），选定的轴以 F×速度倍率的速度运行，不需要等待指令执行完毕直接执行下一条指令

1 0: 非联动相对 X 0.0 F 0

参数: X 0.0 (某轴的运动增量)，F 0 (运行速度)

8.16 非联动绝对

走绝对坐标（即每次运动都以原点作为参考点），选定的轴以 F×速度倍率的速度运行。不需要等待指令执行完毕直接执行下一条指令

1 0: 非联动绝对 X 0.0 F 0

参数: X 0.0 (某轴的绝对坐标)，F 0 (运行速度)

8.17 非联动缓停

设置有效时，做非联动运动的轴，缓慢停止运动

1 0: 非联动缓停 X 禁止 Y 禁止 Z 禁止 A 禁止

参数: X 禁止 (X 轴功能选择是否有效)，Y 禁止 (Y 轴功能选择是否有效)，Z 禁止 (Z 轴功能选择是否有效)，A 禁止 (A 轴功能选择是否有限)

8.18 非联动急停

设置有效时，做非联动运动的轴，紧急停止运动

1 0: 非联动急停 X 禁止 Y 禁止 Z 禁止 A 禁止

参数: X 禁止 (X 轴功能选择是否有效)，Y 禁止 (Y 轴功能选择是否有效)，Z 禁止 (Z 轴功能选择是否有效)，A 禁止 (A 轴功能选择是否有限)

8.19 非联动判 OK

判断选定的非联动轴是否停止运动，停止运动执行下一条指令，否则跳到目的标号处

1 0: 非联动判 OK 如果 (X 电机 OK 了) 下执, 否则跳到 0:

参数: X (选轴, 默认 X 轴), 0: (目的标号)

8.20 非联动等待 OK

等待选定的非联动轴停止运动, 才执行下一条指令

1 0: 非联动等待 OK 等待, 直到 X 轴 OK

参数: X (选轴, 单轴默认 X)

8.21 坐标比较

选定轴的坐标与设定的值作比较, 满足条件执行下一条指令, 否则跳到目的标号处

1 0: 坐标比较 如果 (X \geq 0.0) 下执, 否则跳到 0:

参数: X (选轴, 单轴默认 X), \geq (选择比较条件), 0.0 (被比较的数值)

8.22 设定坐标

执行到该指令, 将当前位置设为新坐标

1 0: 设定坐标 X 0.0 Y 0.0 Z 0.0 A 0.0

参数; X 0.0 (X 轴坐标), Y 0.0 (Y 轴坐标), Z 0.0 (Z 轴坐标), A 0.0 (A 轴坐标)

8.23 联动倍率设定

运行到此指令, 正在运行的联动轴的速度自动变为合成速度 $F \times$ 设定的联动轴速度倍率

1 0: 联动倍率设定 0 %

参数: 0 % (设定联动轴速度倍率)

8.24 非联动倍率设定

运行到此指令, 正在运行的选定的非联动轴的速度自动变为设定的速度 $F \times$ 选定的非联动轴的速度倍率

1 0: 非联动倍率设定 X 轴 0 %

参数: X 轴 (选轴, 默认 X), 0 % (设定非联动轴速度倍率)

8.25 输出

设置输出口状态, 能同时控制 5 个输出口的通断

1 0: 输出 0 0 0 0 0 断

参数: 0 (输出口号), 断 (选择通断状态)

8.26 工件置数

工件数重新设定, 与工件计数指令配合使用

1 0: 工件置数 0

参数: 0 (设定工件数)

8.27 工件计数

执行一次该指令, 工件数加 1 或者减 1

1 0: 工件计数 减 1

参数: 减 1 (选择工件数加或者减)

8.28 判断跳转

检测输入口的状态, 满足条件执行下一条指令, 否则跳到目的标号处

1 0: 判断跳转 如果 (输入 0 = 断) 下执, 否则跳到 0:

参数: 输入 0 (输入口号), 断 (选择通断状态), 0: (目的标号)

8.29 绝对跳转

执行到该指令处, 直接跳到目的标号处

1 0: 绝对跳转 直接跳到 0:

参数: 0: (目的标号)

8.30 循环

当程序执行到此处, 跳转到目的标号处, 再执行 N 次。注意: 需循环的程序段第一条指令填写标号, 循环指令前一指令是循环程序段最后一条指令

1 0: 循环 次数 0, 跳到 0:

参数: 次数 0 (循环次数), 跳到 0: (目的标号)

8.31 机械零判 OK

判断是否完成回机械零动作, 如果完成执行下一条指令, 否则跳到目的标号处

1 0: 机械零判 OK 如果 (回机械零 OK 了) 下执, 否则跳到 0:

参数: 0: (目的标号)

8.32 跳转到第 1 行

执行到该指令行, 直接跳到第一行程序继续执行

1 0: 跳转到第 1 行

参数: 无

8.33 空指令

程序执行到该指令直接执行下一行

1 0: 空指令

参数: 无

8.34 M 比较

M 寄存器是为辅助实现一些加工动作, M 寄存器值在同一个程序多个任务一直有效。将某个寄存器值与其它数值做比较, 满足条件执行下一条指令, 否则跳到目的标号处

1 0: M 比较 如果 (M 0 ≥ 0) 下执, 否则跳到 0:

参数: M 0 (寄存器的名称), ≥ (选择比较条件), 0 (被比较的数值), 0: (目的标号)

8.35 M 设定

给寄存器的赋值

1 0: M 设定 M 0 ≡ 0

参数: M 0 (寄存器的名称), 0 (数值)

8.36 M 逻辑非

将数字, 寄存器的值, 定时器的值或参 1 等取非, 赋值给寄存器。值为 0, 取非后寄存器赋值 1; 当值为除了 0 以外的其它数值, 取非后寄存器赋值 0。比如:

M2=!T1 将定时器 1 中的数值取非后赋值给寄存器 2

1 0: M 逻辑非 M 0 ≡ ! 0

参数: M 0 (寄存器的名称), 0 (数值, 寄存器, 定时器等)

8.37 M 运算

将经过数学计算的结果赋值给某寄存器, 被计算的可以是具体的数字, 寄存器, 定时器等

1 0: M 运算 M 0 ≡ 0 + 0 + 0

参数: M 0 (寄存器的名称), 0 (数值, 寄存器, 定时器等)

8.38 定时器比较

将某个定时器计时时间与其它数值做比较, 满足条件执行下一条指令, 否则跳到目的标号处

1 0: 定时器比较 如果 (T 0 ≥ 0) 下执, 否则跳到 0:

参数: T 0 (定时器的名称), ≥ (选择比较条件), 0 (被比较的数值), 0: (目

的标号)

8.39 定时器启动

设置定时器初始值，启动或停止计时

1 Q: 定时器启动 停止 T Q = Q

参数: 停止 (选择启动或停止计时), T Q (定时器名称), Q (定时器初始值)

8.40 定时器存入 M

将 M 寄存器的值设定成当前计时时间

1 Q: 定时器存入 M M Q = T Q

参数: M Q (寄存器的名称), T Q (定时器名称)

8.41 子程调用

调用子程序，子程序指令参数用子 1/子 2/子 3 代替，主程序子程调用的指令参数子 1/子 2/子 3 位置写上具体值

1 Q: 子程调用 名: Q 子 1: Q 子 2: Q 子 3: Q

参数: 名: Q (调用的子程序名称), 子 1: Q (子程序指令参数, 此处写具体数字)

8.42 子程开始

子程序的开头和结尾需要两个单独的指令组合，该指令为子程序内容的开头

1 Q: 子程开始 名: Q

参数: 名: Q (被选择调用的子程序名称)

8.43 子程结束

为子程序的开头和结尾需要两个单独的指令组合，该指令为子程序内容的结束

1 Q: 子程结束

参数: 无

8.44 结束

运行到此指令，程序结束

1 Q: 结束

参数: 无

8.45 暂停

运行到此指令，程序暂停，按启动按钮，程序继续执行下一条指令

1 Q: 暂停

8.46 回机械零

根据设定好的轴和回零方向进行回机械零动作，不需要等待指令执行完毕直接执行下一条指令

1 0: 回机械零 X 轴回机械零，负向

参数：X 轴（选轴，单轴默认 X 轴），负向（回机械零方向选择）

9. 实例

※应用案例三盏灯亮灭

前提条件：输出1外接继电器，接一盏灯(1号灯)；输出2外接继电器，接一盏灯(2号灯)；输出3外接继电器，接一盏灯(3号灯)

动作要求：1号灯亮0.5s，灭0.5s再亮，重复亮灭动作；2号灯亮1s，灭1s，重复亮灭动作；3号灯亮2s，灭2s，重复亮灭动作；三盏灯闪烁，彼此互不影响。

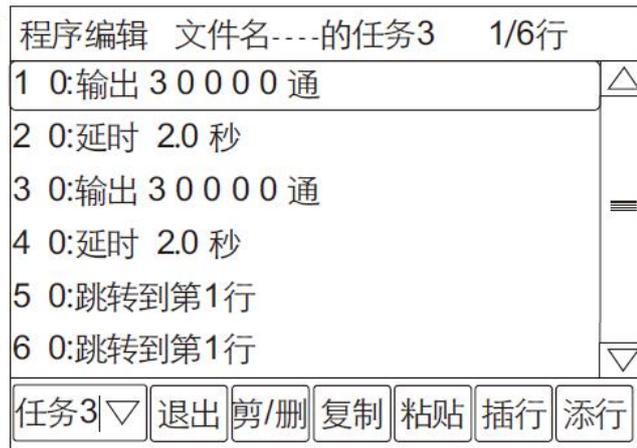
◆ 任务一 1-4程序行

程序编辑 文件名---的任务1 1/6行	
1 0:输出 1 0 0 0 通	△
2 0:延时 0.5 秒	
3 0:输出 1 0 0 0 断	≡
4 0:延时 0.5秒	
5 0:跳转到第1行	
6 0:跳转到第1行	▽
任务1 ▽ 退出 剪/删 复制 粘贴 插行 添行	

◆ 任务二 1-4程序行

程序编辑 文件名---的任务2 6/6行	
1 0:输出 2 0 0 0 通	△
2 0:延时 1.0 秒	
3 0:输出 2 0 0 0 断	≡
4 0:延时 1.0 秒	
5 0:跳转到第1行	
6 0:跳转到第1行	▽
任务2 ▽ 退出 剪/删 复制 粘贴 插行 添行	

◆ 任务三 1-4程序行



※保存成文件名为 135 的程序

按退出按钮，是否保存文件弹窗选择是，输入文件名 135，再点击确认按钮。