

BenT-CNC

可编程运动控制器 BR010 系列

N0:1

BR010 快速编程指南与实例

第二次修订 (适合 2017-9 月以后版本)

浙江俊腾（奔腾）数控自动化设备厂
ZheJiang Jun Teng (BenT) CNC factory automation equipment

前 言

尊敬的客户：

对您惠顾选用浙江奔腾数控设备公司的BR010系列可编程系统产品，我们深表感谢和荣幸！

说明书下载 与 对应视频讲解:

登入官方网址：www.btcnc.net 至“**BR010 系列**”查看电子手册与视频文件

资料说明:

本系统配备了 2 套说明书，第一本为《NO:1 BR010 快速编程指南与实例》(本册)，第二本为《NO:2 BR010 详细产品手册》。其中《NO:2 BR010 详细产品手册》主要讲解了系统每条指令的含义及其应用。建议先阅读本册，如有不懂之处，再阅读相应的《NO:2 BR010 详细产品手册》。《NO:2 BR010 详细产品手册》作为查阅使用，故不印刷发册（官网处下载）。

编程方式:

控制器支持 2 种编程方式：

- 1: 控制器可以脱机编程（所有操作在控制器上完成）
- 2: 在线编程，在电脑上编程（需要下载器），在电脑编程会直观、方便很多

软件下载地址：<http://btcnc.net/web/2moto/WEB/4.htm>

适用范围:

适合以下行业：

- | | |
|------------|-----------|
| 1: 工业自动化控制 | 2: 智能家居控制 |
| 3: 安防领域 | 4: 测试工装控制 |
| 5: 教学研究 | 6: 工业机器人 |

禁止用于以下行业：

- 1: 医疗设备
- 2: 车辆，船舶，铁路等生命攸关的领域
- 3: 电梯，消防，锅炉设备，发电设备，矿用设备等特种行业
- 4: 法律禁止的行业，如武器制造，制毒设备等。

使用温度：-5℃-60℃（无凝结） 海拔：低于 4000Km

环境：无水汽，无导电或非导电性粉尘（或使用封闭机箱）

免责声明：

任何电子设备不可能万无一失。设备拥有人应做好相应防护措施及风险管理计划，如应本设备直接或间接造成任何人身或财产损失，本公司不负责赔偿。

目 录

BENT-CNC	1
第 1 章 产 品 信 息	5
1.1 型号及意义.....	5
1.2 本品性能参数.....	5
第 2 章 电气接线与调试方法	6
2.1 【指示灯的显示】与【对比度的调节】.....	6
2.2 【电源】与【信号输入】接线方式.....	6
2.2.1 接线方法.....	6
2.2.2 调试方法.....	7
2.3 输出设备的接线方式.....	7
2.3.1 接线方法.....	7
2.3.2 调试方法.....	7
2.3.3 指示灯显示位置.....	8
2.4 步进/伺服电机的接法（如不带此功能，可略过）.....	8
2.4.1 接线方法.....	8
2.4.2 调试方法.....	9
2.4.3 指示灯显示位置.....	9
2.5 【启动】【急停】【暂停】的接线.....	9
第 3 章 系统操作方法	10
3.1 系统的编程方式.....	10
3.2 系统的启动/暂停/急停.....	10
3.3 本机上常用的操作设置.....	10
3.3.1 菜单参数修改.....	10
3.3.2 权限的介绍及其进入方法（只需了解即可）.....	11
3.3.3 其他固定的功能介绍.....	11
第 4 章 程序实例	12
4.1 程序所有指令集一览表.....	12
4.2 输出/输入（电磁阀）等编程举例.....	13
4.2.1 简单的气缸开启与关闭延迟.....	13
4.2.2 多设备同时开启与关闭.....	13
4.2.3 【在菜单内】可调延迟的开启与输出.....	13
4.2.4 感应器输入功能（气缸限位功能）.....	14
4.2.5 【循环】与【屏幕上显示计数】功能.....	14
4.3 【步进/伺服电机】编程控制的实例.....	15
4.3.1 电机的编程步骤.....	15
4.3.2 关于菜单内电机固定参数的介绍.....	15
4.3.3 【步进/伺服电机】控制语句的解释.....	17

4.3.4 【*回零语句的解释： X 机回零 003.00 速 110.00 等停】17

4.3.5 【回零语句】与【反转语句】区别：18

4.3.6 关于最高输出速度介绍18

4.3.7 电机正反转的（距离与速度）控制18

4.3.8 【菜单内可调】的电机距离与速度控制19

4.3.9 联动【XY 电机】【回零】同时运行和分开运行19

4.3.10 【XY 电机】与【输出（气缸）】的同时控制20

4.4 *控制器高级功能的演示（按需学习）21

4.4.1 逐级改速与坐标清零的方法21

4.4.2 多个程序同时运行（存多套程序）21

4.4.3 【*重要】如 1 那跳 00 否则跳 00 不返（跳转/判断语句）22

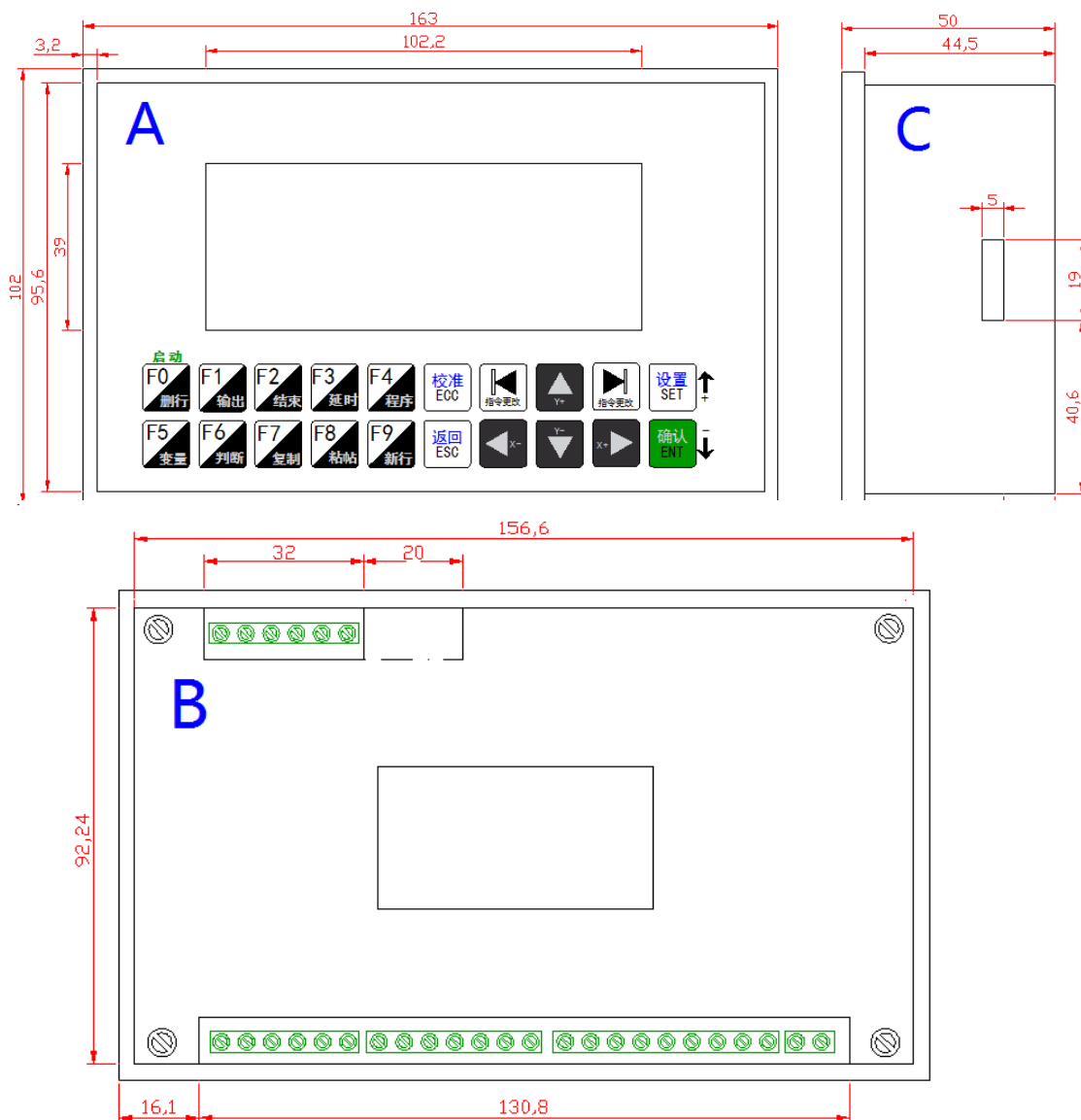
4.4.4 多个程序自动的先后运行与同时运行22

第 5 章 附:脱机编程的简介23

5.1 程序编程进入方法23

5.2 编程步骤与操作23

5.3 *文字输入方法介绍（附录）24



第1章 产品信息

1.1 型号及意义

型号: BR010 11T8X2M1C

11:输出数量, T:晶体管, 8X:输入数量, 2M:二路脉冲(选配), 1C:一路RS485通讯(待定)

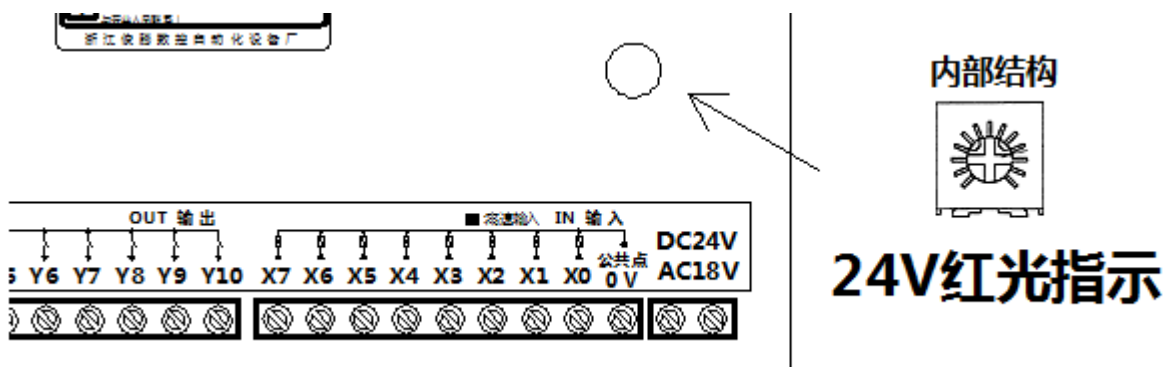
BR: 奔腾数控可编程系列产品标志 00: 经济版 01: 标准版 02: 高配版 最后 0: 对应版本的序号

1.2 本品性能参数

项目	参数	说明
工作电压	直流 12V-26V	24V, ≥2A, 48W 以上电源
输出点数	11 点	晶体管 2A 输出
输入点数	11 点	8 点自定义+3 路系统固定
可控制轴数(选配)	X/Y, 2 轴	最高速度 20KHZ, 2 轴联动
电机脉冲输出电压	5-24V 集电极压差式脉冲信号(内部已串 1.2K 电阻)	
通讯方式(待定)	1 路 RS485	功能待开发
程序储存行数	100 行, 可自由分割 n 个小程序	
菜单行数	23 行用户自定义创建菜单	
内置模拟 CPU	5 个模拟 CPU, 可低速同时运行 5 个不同程序	
指令集	多达 19 条编程指令功能	
按键材质	PVC 薄膜开关	
屏幕结构	单色点阵屏, 分辨率: 192x64	
AD/DA 模拟信号	不支持	
编码器	不支持	
高速计数器	不支持	

第2章 电气接线与调试方法

2.1 【指示灯的显示】与【对比度的调节】



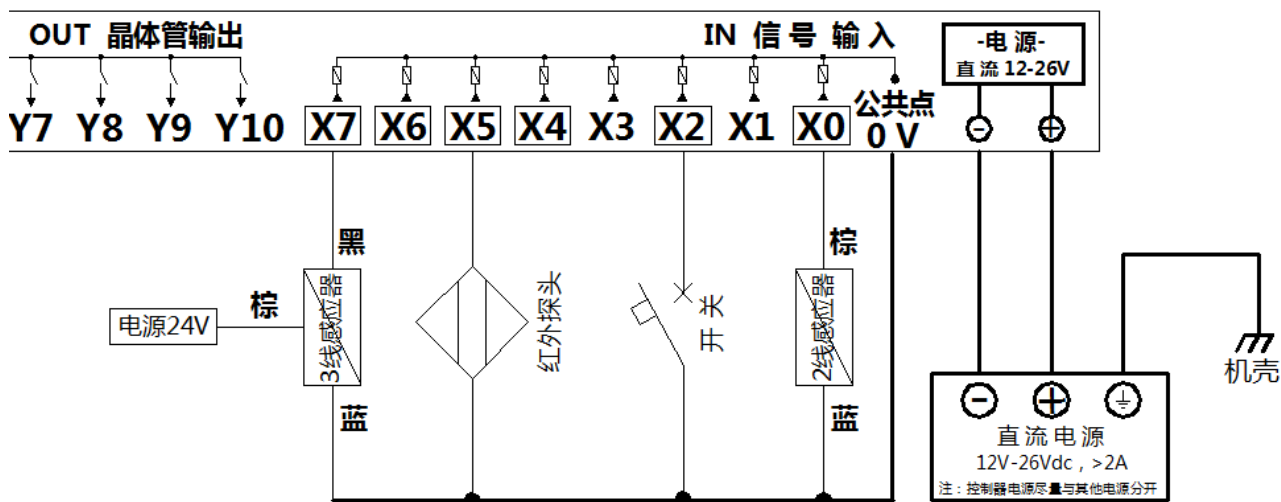
对比度：因系统搭载了一片 LCD 液晶屏幕，因此可以调节该屏幕显示字体的明暗。出厂默认是调节为正常显示。但因运输、工作环境、个人偏好，可适当自行调节。

调节方法：找到对比度旋钮孔位（上图），用小螺丝刀轻轻调节即可。

电源指示灯：对比度旋钮孔位处，还带有电源指示功能，如亮光，则表明系统有正常电源输入。

2.2 【电源】与【信号输入】接线方式

2.2.1 接线方法



***注意：**

- 1: 使用直流电源为 12-26V 电压， $\geq 2A$ 。且电源的地线尽量与机壳连接
- 2: 电源最好独立供电，不要与其他设备共用，避免干扰！

信号输入：图上的输入设备只是为举例而已，因此端口不一定为图上的设备

- 1: 二线感应器接法：蓝色（黑色）：接 0V，棕色（红色）：接信号线（上图）

- 2: 三线感应器接法: 蓝色:电源 0V, 棕色:电源 24V, 黑色:接信号线 (上图)
- 3: 信号端口以接收 0V 信号为有效信号, 故支持 12-24V 电子感应器, 行程开关, 蘑菇按钮等接触式开关
- 4: 信号线长建议不要超过 2 米, 较长距离建议使用屏蔽线
- 5: 信号线安装尽量远离强电设备

2.2.2 调试方法

控制器无需编程即可手动测试输入设备

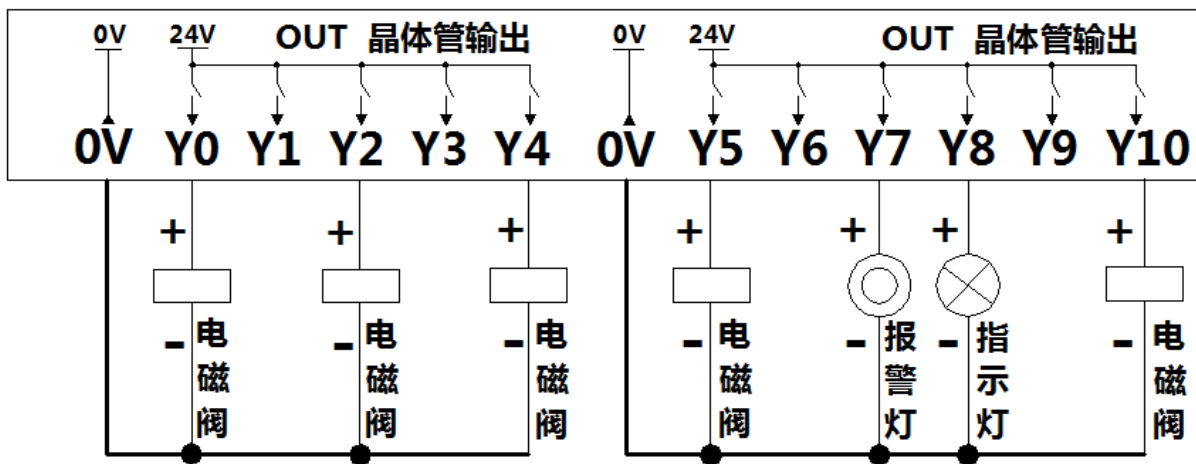
本机上调试: 主界面下按 2 次 **手动测试** 进入感应器页面。判断: 无感应为 0, 有感应为 1 (下图)



软件上调试: 打开软件, 点击  可进行输出手动测试。灯亮表明有信号 (右图)

2.3 输出设备的接线方式

2.3.1 接线方法



*提示:

- 1: 输出自带 **正极** 电压输出, 公共线为 0V, 因此请分清正负极。如上图
- 2: 接入设备的功率建议 24V, 1A 以下。以保证系统的稳定和端口的寿命
- 3: 晶体管输出无工作寿命, 但因接入电磁阀/继电器等感性负载时, 偶尔会击穿晶体管。
- 4: 如要控制高功率的设备, 则使用中间继电器等进行隔离或放大电压

2.3.2 调试方法

控制器无需编程即可手动测试输出设备

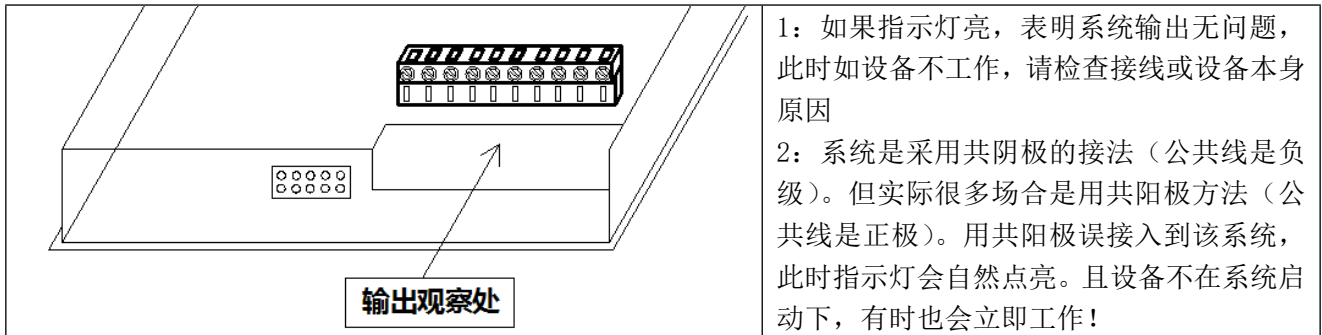
本机上调试：按【手动测试】进入界面，按【F0-F9】和【校准键】可全部进行输出手动调试（下图）。

对应接口说明：【F1-Y0】【F2-Y1】【F3-Y2】【F4-Y3】【F5-Y4】【F6-Y5】【F7-Y6】【F8-Y7】
【F9-Y8】【F0-Y9】 【校准-Y10】

F1 Y0端口	零口	F2 Y1端口	一端口	F3 Y2端口	二端口
F4 Y3端口	三口	F5 Y4端口	四端口	F6 Y5端口	五端口
F7 Y6端口	六口	F8 Y7端口	七端口	F9 Y8端口	八端口

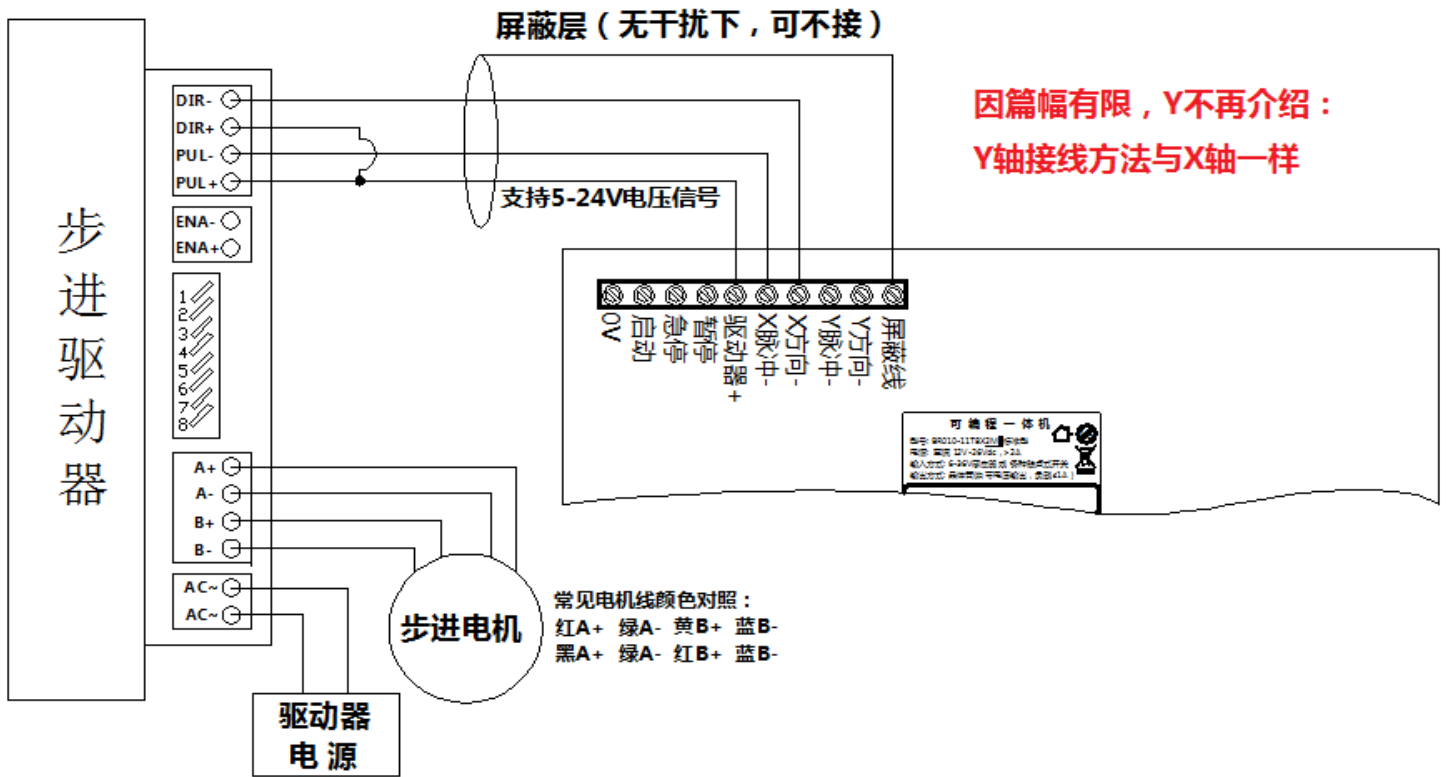
软件上调试：打开软件，点击  可进行输入手动测试

2.3.3 指示灯显示位置



2.4 步进/伺服电机的接法（如不带此功能，可略过）

2.4.1 接线方法



*注意:

- 1:脉冲工作信号电压: 5-24V 集电极压差式脉冲信号(内部已串 1.2K 电阻), 可直接连接驱动器信号端
- 2:电机输出信号类型:脉冲 PUL 和 方向 DIR, 无使能(EN)信号
- 3:电机信号线长尽量控制在 1 米之内, 尽量采用屏蔽线材, 尽量远离电机/高频/电磁等高辐射设备
- 4:市场上驱动器常见的标注示例: 脉冲: PUL PLS CP CW PP 方向: DIR CCW (伺服 SIGN PD)
- 5:如果步进电机方向转反了, 二相电机只需调换 A+A-线即可。三相电机对换任意 2 条线即可

2.4.2 调试方法

本机上调试:

快捷操作:在主界面上, 直接按 或 键即可运行。此时屏幕右下角会显示当前坐标。

常规方法:在主界面上, 按“确认”按钮, 进入电机界面, 按 或 键即可运行

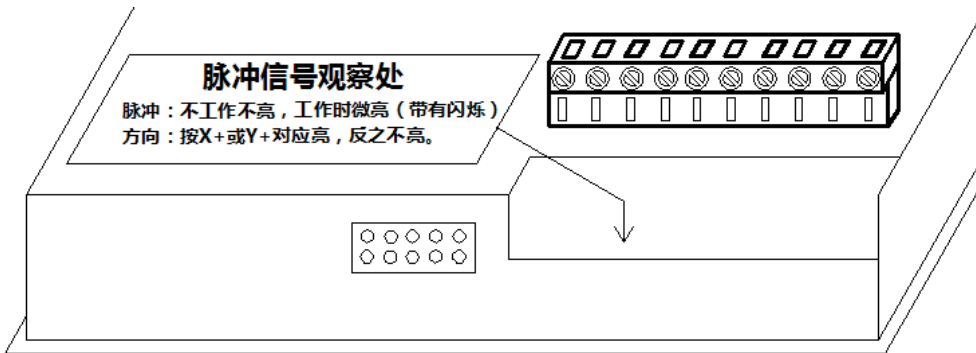
速度调整:按 键, 即可调整手动测试的移动速度, 屏幕的最下处, 会显示当前速度和频率。

软件上调试: 打开软件, 点击图标进行电机手动测



试

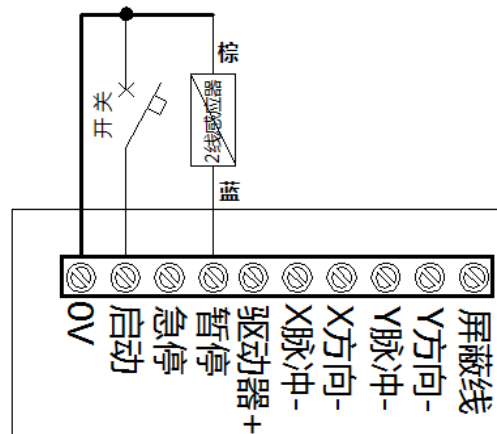
2.4.3 指示灯显示位置



2.5 【启动】【急停】【暂停】的接线

注意事项：

- 1: 因【启动】【急停】【暂停】的输入性质与普通端口输入的一样，故接法与上文讲的输入接法一致
 - 2: 这里的【启动】【急停】【暂停】是系统固定的接口，当然系统也支持可编程的方式（下文有讲）
 - 3: 急停采用的是常开接法，**在设备开机前，不允许闭合**。系统会提示，并弹出密码。此时只需断开急停，并重启即可。
- 密码作用：输入之后（00000），会恢复备份区的菜单内容及设置值。备份方法：开机前按 F7，密码 00000



第3章 系统操作方法

3.1 系统的编程方式

在本册<前言>中已经提到，系统支持 2 种编程方式：

- 1: 控制器可以脱机编程（所有操作在控制器上完成）
 - 2: 在线编程，在电脑上编程（需要下载器），在电脑编程会直观、方便很多
- 软件下载地址：<http://btcnc.net/web/2moto/WEB/4.htm>

为了减小篇幅，这里不再详细介绍，2 种的视频操作方法：

- 1: 脱机编程：<http://btcnc.net/web/2moto/WEB/6.htm>（本册最后的附页中，有大致介绍）
- 2: 在线编程：<http://btcnc.net/web/2moto/WEB/4.htm>（实在过于简单，稍微琢磨即可了解）

3.2 系统的启动/暂停/急停

说明：为了系统的适用性和方便原则，提供了 2 种方式：**系统固定的方式**与**可编程的方式**

- ① **系统固定的启动/暂停/急停方式** 优点：不需编程直接使用。缺点：功能固定，不可更改，不可编程

启动：支持 3 种启动方法

- 1:按本机上的 **F0** 按钮
- 2:触发本机上的**启动**输入接口
- 3:在软件上，点击



急停：支持 3 种急停方法

- 1:按本机上的**返回**按钮
- 2:触发本机上的**急停**输入接口
- 3:在软件上，点击



注意：急停分 2 种状态，在软件上可设置：1.急停后动作归位到初始状态（推荐）2.急停后动作保持当前状态

暂停：支持 3 种暂停方法

- 1:按本机上的 **F9** 按钮
- 2:触发本机上的**暂停**输入接口
- 2:在软件上，点击



注意：暂停后，需按**启动**才能继续运行。**步进/伺服电机在运行当中不会立即暂停，而是执行完当前任务后暂停**

- ② **可编程的方式** 优点：用户可编程，可按照实际要求编程 缺点：对新手来说，带来不便

启动：运用系统的子程序语句（见下文举例） **急停/暂停**：运用急停/暂停线程语句（不建议使用）

3.3 本机上常用的操作设置

3.3.1 菜单参数修改

- ①菜单界面的操作方法

1:在主界面下，按**设置键**进入菜单界面。2:按**上下键**可进行菜单上下滚动选择 3: 按**返回键**，返回到主界面

- ②菜单数据的修改

选中某行菜单项，按**设置**此时数据上会出现小光标 **01.00**，按**左右键**可移动光标，按**上下键**可修改光标内数值，或直接输入 F1-F9 的数值。数据修改完毕，按**确认**键，小光标消失，修改成功！按**返回键**回到主界面

3.3.2 权限的介绍及其进入方法（只需了解即可）

- ① 用户模式界面（开机默认）：**设备运行**，**手动测试（控制输出/输入/电机）**，**菜单参数修改**
 - ② 管理员界面：**用户模式的全部功能**，**可修改隐藏菜单的参数**，合适【调机与维修，管理员】使用
 - ③ 程序员界面：**【用户+管理员模式】【菜单创建】【各界面的命名】【程序编程】【控制器固定功能使用】**
- 管理员模式进入：**在用户模式主界面下（开机默认就是用户模式），按**确认**键3秒，进入**管理员模式**
- 程序员模式进入：**在管理员模式主界面下，按**编程**键，自动弹出进入**程序员界面**的密码（默认是12345），按**确认**键进入。程序员模式拥有系统最高的权限！（下图）**如要退出程序员模式，重启系统即可。**



3.3.3 其他固定的功能介绍

- ① **设备初始化：**开机之前按 **F5** 键，密码为 00000，可恢复到出厂状态(内容全部删除)
- ② **默认值备份：**开机之前按 **F7** 键，可将当前使用中的**菜单数据及名称**备份到备份区(密码 00000)
- ③ **恢复到默认：**开机之前按**返回**键，将备份区的**数据及名称**读出并写入当前的菜单项(密码 00000)
- ④ **重置菜单/手动名称：**开机之前按 **F8** 键，密码为 00000，只删除菜单/手动内容(不删除程序)
- ⑤ **永久工作计数器清零：**开机之前按 **F9**，即可归零计数器。（临时计数器重启会自动清零）
前提:在菜单 33 行【★使用次数限制 x100】设置为 0 次才可以允许清零
注意:工作计数器与【★使用次数限制 x100】相关，请看下文介绍

菜单固定功能介绍：须在管理员或程序员模式下，进入菜单，才可以看得到

菜单行数	菜单名称	默认值	含义
31	★按 F 键直接控端口?	是 否	如设置为“是”，则系统在停机或运行当中，直接按 F0-F9，有对应的端口输出，方便在运行当中同步调试。为了安全起见，一般为关闭。
32	★急停后动作归位?	是 否	急停分 2 种状态，.是：急停后动作归位到初始状态（推荐） 否：急停后动作保持当前状态，再按一次急停，动作归位。
33	★使用次数限制 x100	00000	系统允许使用次数（催款用），0 次：无数次。设置为其他次数，如数值 1，则代表允许工作 100 次。设置完成后，每次重启开机，系统会提示还剩余多少次。达到数值之后，自动锁屏且提醒“试用时间已到”，此时需要管理员密码解屏！解屏之后设置的使用次数变为 0！（无数次 PS：次数是如何获得：以程序内运行“计数加一”语句为参考。
34	★按键音开启吗?	是 否	控制器按键声音的开关
35	★管理员密码修改为	12345	进入程序员模式（最高权限）的密码，解屏密码。需牢记。 如果密码忘记，会导致无法编程或解锁屏幕。此时只能设备初始化（见上文）
36	总底行滚条次数剩时	11161	系统默认主界面的显示内容，在这里可以选择是否关闭。 对应方式：1=总，1=底行，1=滚条，6=工作次数位数，1=剩时。 1：表示开。0：表示关。具体操纵建议到软件上进行更加方便。
37	次数显示位置 xxayb	00101	x=00, a=1, y=0, b=1 a: 计数器类型，0: 永久计数器(重启还保存次数), 1:临时计数器（重启后不保存次数，但控制器后台还在累加计数，用于使用次数（催款用）） b: 计数器是否反白显示，1:反白, 0:不反白 x 或 y: 表示把工作计数器显示在主屏幕的 X 或 Y 的位置
38	★显示几号线程进度	0	在控制器主界面的滚动条上显示的是哪个 CPU 的信息动态

第4章 程序实例

前言:

- 1: 本篇只是讲到了常见的程序编程举例, 并不是说没讲到的, 就不能达到某些功能。望能够举一反三。在官网上, 会不断更新一些编程技巧和常见有用的实例, [地址: WWW.BTCNC.NET](http://WWW.BTCNC.NET) 找到 BR010 产品
- 2: 在上文中, 也提到了系统支持 2 种编程方式, 因此在编程之前, 请先学习各自的编程操作方法

4.1 程序所有指令集一览表

19 种程序的名称

程序名称	用途
子程序 A---1 线---F1 触发	程序开始语句。可设置程序启动方式和启动接口 可分配程序在哪个 CPU 上运行 (共 5 个 CPU)
=====结束或返跳	程序一旦碰到此语句, 程序就会结束 (如有跳转情况就会返跳)
Y01=1	输出端口语句。可设置 Y 端口输出或关闭
等待 X0=1 过 00.0 后 ->b	输入信号检测语句。可检测信号状态 或 死等信号状态
如 1 那跳 00 否则跳 00 不返	1:判断语句 2:输入信号检测语句 3:跳转语句 4:循环语句
01.00 秒	固定延迟。常用于程序的固定延迟。不可在菜单内修改延迟
等:菜单 00 xxxxxxxx	可变延迟。常用于程序的可变延迟。可在菜单内修改延迟
循环开始 次 00000	循环开始语句。可设置固定循环次数 和 菜单内设置的次数
循环结束	循环结束语句。循环内容选定:循环开始(头)和循环结束 (尾)
计数加一	计数器加 1 专用语句。可设置计数器永久累加 或 临时累加。
X 机正转 01000 速 01000 等停	唯一的步进/伺服电机(脉冲)语句。正转、反转、速度设置等
a=00010	直接数值赋值。常用于变量的初始值
b<=>变量:b	相互赋值语句。常用于读取菜单的数据
a=a+00001	十则运算语句。加减乘除等
急停:线程 5	线程控制语句。用于可编程的急停, 暂停等功能。不推荐使用
显示:菜单 00 在 0 行 00 列正显	显示文字语句。用于显示程序的工作状态
跳(0=0)?A:+23 不返	高级判断语句。条件判断、数据赋值、信号检测等
01 行空白	空白语句。无任何意义的合法语句
单步	单步语句。功能暂不齐全
如要详细了解每条指令的含义和使用方法: 登入网页: http://btcnc.net/web/2moto/2MOTO.htm - 《NO:2 BR010 详细产品手册》	

4.2 输出/输入（电磁阀）等编程举例

4.2.1 简单的气缸开启与关闭延迟

程序内容	解释
子程序 A---0 线---X2 触发 Y00=1 02.50 秒 Y00=0 =====结束或返跳	程序开头语句。程序名：A，程序配定在 0 号 CPU 内运行，X2 启动该程序。 Y00 输出开启。其中 Y00 为端口号，后面的 1 开，0 关，2 取反 固定延迟 2.5 秒 Y00 输出关闭。 程序结束语句。

效果：按启动开关（X2）后，Y00 开启，2.5 秒钟后，Y00 自动关闭

如系统只存有 1 套程序，则子程序语句可省略，系统会默认分配一个子程序语句。如果系统内要有多套程序，即需要出现多个子程序。则子程序语句不能省略。

要求：程序名不能重名。同个 CPU 上不能同时运行 2 个及以上的程序。设置为 0 号 CPU 时，运行主界面上会显示进度条。子程序不能用同一编号的触发（启动）方式。关于子程序语句登入网页查看详细说明！

4.2.2 多设备同时开启与关闭

程序内容	解释
子程序 A---0 线---X2 触发 Y00=1 Y01=1 03.00 秒 Y00=0 Y01=0 =====结束或返跳	程序开头语句。程序配定在 0 号 CPU 内运行，X2 启动该程序。 Y00 输出开启。（1 开，0 关，2 取反） Y01 输出开启。 固定延迟 3 秒 Y00 输出关闭。 Y01 输出关闭。 程序结束语句。

解释：程序是从上往下流水式运行，每行间隔时间 00.01 秒！

所以这里【Y00 开启】与【Y01 开启】只花费 00.01 秒时间，可视为同步！

效果：按启动开关（X2）后，【Y00】与【Y01】同时开启，3 秒钟后自动同时关闭

4.2.3 【在菜单内】可调延迟的开启与输出

先菜单的设定			
00 行	Y00 开启时间	02.00	（00 行：表示菜单位置，数据采用 00.00 数据格式）
01 行	Y00 关闭时间	01.00	
程序内容	解释		
子程序 A---1 线---X2 触发 Y00=1 等：菜单 00 Y00 开启时间 Y00=0	程序开头语句。程序配定在 1 号 CPU 内运行，X2 启动该程序。 Y00 输出开启。（1 开，0 关，2 取反） 可变延迟。延迟时间读取 00 行菜单的数据：02.00 Y00 输出关闭。		

等: 菜单 01 Y00 关闭时间 -----结束或返跳	可变延迟。延迟时间读取 01 行菜单的数据: 01.00 程序结束语句。
---------------------------------	---

效果: 按启动开关 (X2) 后, 【Y00】开启, 2 秒钟后 (菜单设定的时间), 【Y00】关闭, 关闭时间为 01.00 秒 (菜单设定的时间)。程序结束!

4.2.4 感应器输入功能 (气缸限位功能)

程序内容	解释
子程序 B---0 线---X2 触发 Y00=1 03.00 秒 死等 X0=1 过 05.00S 后开 Y02 Y00=0 04.00 秒 死等 X1=1 过 04.00S 后开 Y02 =====结束或返跳	程序开头语句。程序配定在 0 号 CPU 内运行, X2 启动该程序。 Y00 输出开启。(1 开, 0 关, 2 取反) 固定延迟 3 秒 死等 X0 伸限位是否有输入, 如 5 秒内无输入, 自动开启 Y02 (报警) Y00 输出关闭。(1 开, 0 关, 2 取反) 固定延迟 4 秒 死等 X1 回限位是否有输入, 如 4 秒内无输入, 自动开启 Y02 (报警) 程序结束语句。

效果: 按启动开关 (X2) 后, 【Y00 气缸】开启, 气缸伸出时间 3 秒, 再检测气缸有没有伸到位 (X0), 检测时间为 5 秒, 如果 5 秒内没检测到信号, 则开启报警页面并输出 Y02 (报警器)。如果有信号, 则运行下面的程序, 【Y00 气缸】关闭, 时间 4 秒。再检测气缸有没有回到位 (X1), 检测时间为 4 秒, 如果 4 秒内没检测到信号, 开启报警页面并输出 Y02 (报警器)。如果有信号, 则运行下面的程序, 即程序结束!

【死等 X0=1 过 02.00 秒 开 Y10】语句的介绍:

意思: 检测 X0 是什么状态, 1: 有感应, 0: 无感应。02.00 秒是检测时间, 如果在 2 秒时间内, 没有检测到对应的条件, 则开启报警 Y10。并且屏幕上显示报警提示。此时需要人工的选择下步的工作状态。如果设置为 00.00 秒, 则屏幕上立即显示报警提示, 但 Y10 不会输出, 而且实时检测条件是否满足, 如满足, 就立即自动往下运行!

4.2.5 【循环】与【屏幕上显示计数】功能

程序内容	解释
子程序 A---0 线---X2 触发 循环开始 00020 次 Y00=1 02.50 秒 Y00=0 计数加一 循环结束 =====结束或返跳	程序开头语句。程序配定在 0 号 CPU 内运行, X2 启动该程序。 循环开始 (头), 次数: 20 次 Y00 输出开启。(1 开, 0 关, 2 取反) 固定延迟 2.5 秒 Y00 输出关闭。 系统自带的计数器。 循环结束 (尾) 程序结束语句。

效果: 按启动开关 (X2) 后, 循环的进行电磁阀 (Y00) 开启, 2.5 秒钟后, Y00 自动关闭。循环次数为 20 次。此时系统计数器碰到 (计数加一) 就自动开始+1 计数, 并显示在屏幕上。

循环次数: 0, 则不运行循环的内容。控制器支持大循环内的小循环 (10 层循环)

循环内容的圈定: 循环开始 xxx 次 -----【循环的内容】-----循环结束。循环开始与循环结束犹如一个括号, 圈定了要循环的内容, 因此二者不能缺一!

4.3 【步进/伺服电机】编程控制的实例

4.3.1 电机的编程步骤

因为步进或伺服电机属于特殊电机，因此在编程之前，需事先设置固定的参数。设置好之后，再到程序内运用语句，进行编程！

一套完整的电机控制需要 2 面的设置：

第一方面:先在菜单里面设置好固定的参数！如：螺距，零点，加减速，限位等

第二方面:在程序内运用指令语句，进行动作编程！（系统唯一控制语句 **X 机正转 01000 速 01000 等停**）

4.3.2 关于菜单内电机固定参数的介绍

进入方法：在管理员模式或程序员模式下，按 **设置** 进入菜单。软件编程：直接在 **【参数设置】** 内进行

菜单行数	菜单名称	默认值	备注
39	★ X 轴零点是几号口？	00008	如零点开关接的是 5 号输入接口，则设置为 00005 (1 和 3 号不得用于零点)，如设置在 7 号以上的端口，则说明系统不使用 机械零点 ，而使用 软件零点 机械零点: 回零点时，电机必须碰到零点开关才会停止，然后系统才标记位置为 0mm 软件零点: 开机时，系统马上标记位置为 0mm，回零时自动回到开机 0mm 的位置！ 强烈建议用机械零点：能有效消除机械误差！具体使用方法，看下文介绍
40	★ X 轴按 ESC 回零速度	04000	不建议修改。 如设置为允许按[返回]回零点的，则回零点的速度在这里调。设置方法在菜单 47 行
41	★ X 轴加减速(0-100)	00004	不建议修改。 电机从启动到正常运转的加速度，与减速度的平缓程度。建议为 4（越大启动越稳定）当电机起步 【吃力】 或 【启动时抖动共振明显】 下，可适当增大数值。
42	★ X 轴右限位（急刹法）	00008	系统支持左右限位。左限位：零点（菜单 39 行）。当设置为 8 或 9 时，表示关闭该功能！
43	★ X 轴点动距离 毫米	000.00	不建议修改。 手动测试点动用的距离。在手动测试时，短按是点动效果，长按是连续旋转。长按时，系统先进行点动效果，再执行长按效果（连续旋转）。 00000 表示关闭该功能
44	★ X 轴寻零速度	00400	不建议修改。 电机退到设置好的寻零点的位置后，会以该数值(HZ)设定的速度慢慢寻找零点。此功能作用于机械零点，软件零点基本没用。该速度不宜过快，使用方法在讲回零语句的时候有介绍
45	★ X 轴手动按键反向	是 <input checked="" type="checkbox"/>	不建议修改。 修改 手动测试 时，电机运行的方向。对程序内电机运行方向不会受影响。如要修改程序内电机运行的方向 步进只需对调电机线的顺序（在讲接线时有提到），伺服则在驱动器内设置转向！
46	★ X 轴主界面显坐标	08062	不建议修改。 关于系统主界面右下角坐标显示位置与样式的设置： 08 为屏幕 X 位置，06 为屏幕 Y 位置，2 反白显示(1 不反，0 为关闭显示)
47	★ X 轴按返回回零点	是 <input checked="" type="checkbox"/>	不建议修改。 按控制器的 返回 键，电机是否回零点。
48	★ X 轴预留功能	65535	不建议修改。备用
49	★ X 轴坐标清 0 启用点	65535	不建议修改。 仅对机械零点有效。意思：65535 为脉冲数，当设置为 65535 时，表示电机在 回零 的时候，不管任何位置，只要是碰到零点开关，电机停下，并标记为 0mm。设置为其他数值，如 10000，表示系统在 10000 个脉冲之外，即使有碰到感应器也无效。在 10000 个脉冲之内任意位置碰到感应器就标记为 0mm，电机停下。有什么用？当感应器装在分度盘或圆盘的时候，回零当中可能会碰到 N 次感应器，而电机停下必须要求是机械 0 度的位置。此时就要设置该数值，来过滤掉中间碰到感应器的信号，仅保留最后 1 次或 2 次的有效信号。对于丝杆，就没意义了。

50	★ X 轴 1 毫米=几脉冲?	00100	
<p>电子齿轮比的设置。用处：电子比例一旦设置好，控制器会自动在后台进行脉冲与实际距离的转换。今后在控制器上显示的距离与程序设置的距离都是实际距离。使得以后编程和调试变得非常方便和简单 因为有些概念对新手比较难理解，这里会详细介绍：</p>			
<p>设置方法：应填数值 = 电机一圈的脉冲数 ÷ 机械螺距 最终数值建议范围 100-3000 之内 关于各部分的解释下文有介绍</p>			
<p>为了照顾有些客户实在不会换算的情况下，我们就用 1 种超简单方法：凑量法，优点：方法简单 缺点：过程繁琐 先把整套设备装到机器上，然后让系统归零（按【返回】或手动按【X-或 Y-】），归零之后，此时系统右下角的坐标为 000.00，然后手动按 X+或 Y+，让电机移动一定距离，如 1mm 或 1cm，此时电机就会旋转并带动机械运动，但实际距离肯定不是与系统对应的。用尺测量机械运动的距离，对比系统显示的距离。如少走，则增加移 1mm 需要多少脉冲的数值。多走，则减少。反复修改之后，就可以基本对准电子齿轮比。</p>			
<p>系统不支持小数点输入，实际也是不允许小数点存在（电机不存在旋转半下的概念），因此当出现小数点的时候，采用四舍五入的方式，经过四舍五入之后，系统的坐标与实际会有一点点的偏差（四舍的话是少走，五入的话是多走），此时系统的坐标只能作为参考价值。</p>			
<p>程序内电机最大填写数值为 655.35，即默认电机最大单次行走的距离是 655.35mm。如果单次行走距离超过 655.35mm，采用 2 种方法 1：分 2 次走，比如第一次先走 500mm，第二次走 500mm（建议此方法，缺点：换段时有小的停顿） 2：修改电子比例值：在原先设置好的基础上，x10。如本来是 100 的，现在设置为 1000。这样本来是 655.35 的距离单位为 mm，现在是 655.35cm。缺点：距离分辨率降低，系统可控制的速度范围减小。因此建议最终的数值不要超过 3000。可以减小驱动器细分脉冲数来使得数值减小</p>			
<p>【电机一圈的脉冲数】解释：在驱动器上（不是本系统），可以设置脉冲数，一般驱动器上有 1 个表格，按照对应的拨码开关，进行拨码。伺服驱动器的话，直接在驱动器上，设置分子、分母比例即可。怎么验证是否对？如设置为 1000，意思就是 1000 个脉冲电机为 1 圈，此时只要让系统发 1000 个脉冲，看看是不是 1 圈即可。让系统发 1000 个脉冲方法：先在这里填写 1000，然后在程序内或手动测试，让它走 1mm。此时系统就会发出 1000 个脉冲。然后看看电机是不是刚好 1 圈。如少走，表示 1 圈不止 1000 个脉冲。多走，表示 1 圈不需要 1000 个脉冲，此时需重新验证驱动器是否设置对！我们建议设置驱动器的细分范围 400-1600。驱动器细分设置越大，则电机行走的分辨率越高（不是精度），旋转越平稳、抖动越低，但速度也越慢。故在懂得道理之后，按照实际情况来设置！</p>			
<p>【机械螺距】解释：电机装在机器上，让电机转 1 圈，带动物体是多少毫米。在有些场合不知道机械螺距的情况下，我们就用上文讲的方法，使系统让电机旋转 1 圈，然后用尺测量具体走了多少 mm 即可。这个值就是机械螺距值。</p>			
<p>一般情况下，都是用丝杆结构，这里罗列了丝杆的挑选方法：</p>			
<p>1：丝杆直径（关乎到寿命）越粗越好，当然价格也越贵 2：丝杆螺距（丝杆转一圈实际带动物体是多少 mm）</p>			
<p>如 1605，它的螺距为 5mm，意思就是说丝杆转动一圈，物体走 5mm。当然有些厂家型号写法不一。</p>			
<p>强烈建议：一定要购买【1000÷丝杆螺距=整数】的丝杆。这样控制器可以比较精准的距离控制。如果不是整数的，出现小数点，不利于距离的控制。上文有讲到原因。</p>			
<p>3：丝杆螺距越大，则带动物体速度越快，但距离分辨率降低，寿命降低。不然则反</p>			
<p>电机圈数的控制：因系统只支持 mm 为单位（没有角度单位，市场上大多控制器也是不支持的）。因步进电机步距角 1.8 度或 1.2 度，指的是电机走 1 步是 1.8 度或 1.2 度。如要走 30 度，则电机应该动 $30 \div 1.8 = 16.7$ 次，但实际上电机只能走 17 次或 16 次，而 0.7 次是走不起来的。如换成 1.2 度的电机就可以：$30 \div 1.2 = 25$ 次。所以在控制角度的时候，要事先算下能不能走的到！如有驱动器细分的情况下：如设置为 1200 细分，电机转 30 度。则可以算出 $360 \div 30 = 12$ 等分，故 $1200 \div 12 = 100$，即 100 个脉冲为 30 度。知道该参数之后，在（1mm 移动需多少脉冲）设置为 100，在程序内只要运行 1mm，就是电机走 30 度了。</p>			
<p>建议：在（1mm 移动需多少脉冲）最好设置为 10，在程序内运行 10mm，效果一致，但交给用户可调的范围就会变得广了</p>			
<p>菜单 51 行 至 62 行 是关于 Y 轴的设置，因设置原理和方法与 X 轴的一样，这里不再介绍。</p>			
63	★手动移动速度 mm/s	050.00	X 或 Y 手动测试时移动的速度，这速度也可在电机手动测试界面内，按  键修改
64	★最高限制速度 mm/s	200.00	电机运行的最高速度。如超过该速度值，则以该数值为准！目的是防止工人乱调。

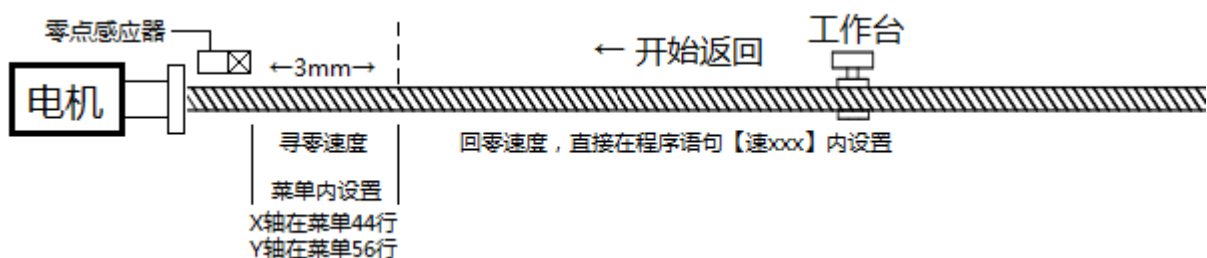
4.3.3 【步进/伺服电机】控制语句的解释

控制器唯一的电机控制语句：X 机正转 010.00 速 010.00 等停

<u>X 机</u>	<u>正转</u>	<u>010.00</u>	<u>速 010.00</u>	<u>等停</u>
X X 电机	正转 电机正转（相对位置，以上次电机停下为参考点）	000.00（运行距离） 程序固定的数值	000.00（运行速度） 程序固定的数值	等停 等电机停下才运行下面的程序。 (就是说当运行下面的程序时，电机肯定停下了)
Y Y 电机	反转 电机反转（相对位置，以上次电机停下为参考点）	菜单 0-8 数值从菜单内读出	菜单 0-8 数值从菜单内读出	
	转到 电机正转到位置（绝对位置，以系统 OMM 位置为参考点）	变量 a-z 数值从变量内读出	变量 a-z 数值从变量内读出	
	回零 电机反转并寻找零点。 (如设置为机械零点，则寻找感应器。如设置为软件零点，则回到 0mm 的位置) 见下文			
	到点 （此功能已关闭）			
	改速 修改电机运行中的速度。不允许电机运行后，马上进行改速，一般是延迟后，再进行			
	缓停 电机有加减速曲线的停下			
	急停 电机立即停下，可能会失步			不等 无需等待电机停下，就运行下面的程序 (就是说当运行下面的程序时，电机还在运行)
	等停 等电机停下，才运行下面程序			

4.3.4 【*回零语句的解释： X 机回零 003.00 速 110.00 等停】

回零采用机械零点下: 003.00 为寻零点的距离。即电机要在感应器前 3mm 的位置, 减速下来找感应器, 3mm 之外的速度为程序语句内设置的任意速度。如采用软件零点（不使用感应器）: 系统则一步到位以程序内设置的速度回 0mm。如下图:



注意:寻零速度不宜过快，因为它要校准零点感应器。一般在 200-600 之间!要想快速回零请在程序内设置回零速度!感应器的寻零距离，是在程序语句内给定，寻零距离不建议直接设置为 0mm。这样回零就没有了校准距离，导致校准不准确。

4.3.5 【回零语句】与【反转语句】区别:

- 1: 【回零语句】: 电机反转，并寻找 0mm。【反转语句】: 电机反转。因此 2 者都是反转的效果。但反转语句可以是反转任意距离。而回零语句是不管在什么地方，执行时，都将回到 0MM 位置。
- 2: 回零语句如使用机械零点，回零当中会寻找感应器，如软件零点，则一步到位至 0mm。因此在使用机械零点时，有校准并消除机械误差的作用（软件零点就无法消除）。而反转语句没有此功能
- 3: 在使用机械零点时，如反转语句执行的距离超过 0MM 位置，则系统会强制在感应器位置停下并标记为 0MM。但标记 0MM 的精度与回零校准的差很多。如使用软件零点，则会允许超过 0MM 位置。如多走 -10mm。因系统不支持负坐标。因此系统就认为-10mm 的位置为 0MM。以后都是以该位置为 0MM

**电机在运行前都要进行一次校准:这是工业习惯，为了安全与精度考虑!
建议采用机械零点（感应器,行程开关等），尽量不要用软件零点。**

4.3.6 关于最高输出速度介绍

系统支持最大脉冲输出能力为 23KHZ，如 2 轴同时运行，则是 X+Y 最大为 23KHZ。

超过该能力系统会报警（报警了不代表不能使用，而会出现意外转速现象，但距离还是准的）常见的原因【电机设置的速度】x【1mm 需要多少脉冲（螺距）】>23000，出现该问题系统会立马报警。消除报警方法：重新启动系统。如问题依旧存在，则还会报警。

解决方法:

- 1: 手动测速的速度是否太高？（一般手动测试时会提示）
 - 2: 程序内电机设置的速度是否太高？（一般是启动运行时提示）
 - 3: 菜单 40 行，按 ESC 返回速度是否太高？Y 轴的设置方法一样。（一般是按【返回】键的时候会提示）
- 以上设置的方法都是降低电机设置的速度，其实最有效的方法是降低电子齿轮比（上文有讲解）。我们知道电子齿轮比是由【机械螺距】和【驱动器细分】组成。因机械螺距改变不容易。所以我们从驱动器细分（电机 1 圈需多少脉冲数）设置下手是最简单的方法，建议驱动器细分设置为 400-1600

4.3.7 电机正反转的（距离与速度）控制

程序内容	解释
子程序 A---0 线---X1 触发	程序开头语句。程序配定在 0 号 CPU 内运行，X1 启动该程序。
X 机正转 040.00 速 010.00 等停	X 电机正转 40mm，速度为 10mm/s，等停：等电机运行完才运行下面语句
03.00 秒	固定延迟 3 秒
X 机反转 030.00 速 031.00 等停	X 电机反转 30mm，速度为 31mm/s，等停：等电机运行完才运行下面语句
=====结束或返跳	程序结束语句

效果: 按 X1 启动按钮后, X 电机正转(距离: 40mm, 速度 10mm/s), 40mm 走完停留 3 秒。然后电机反转(距离: 30mm, 速度 31mm/s)

4.3.8 【菜单内可调】的电机距离与速度控制

先菜单的设定	
00 行 X 轴行走距离	010.00 (00 行:表示菜单位置, 数据采用 000.00, 唯一的电机数据格式)
01 行 X 轴行走速度	020.00
02 行 Y 轴行走距离	030.00
03 行 Y 轴行走速度	040.00
程序内容	解释
子程序 A---0 线---X1 触发 a<=菜单 02 Y 轴行走距离 b<=菜单 03 Y 轴行走速度 X 机正转 菜单 0 速 菜单 1 等停 03.00 秒 Y 机正转 a 速 b 等停 =====结束或返跳	程序开头语句。程序配定在 0 号 CPU 内运行, X1 启动该程序。 把 02 行菜单的数值读出来 (030.00) 赋值给 a 变量。 把 03 行菜单的数值读出来 (040.00) 赋值给 b 变量。 X 电机正转 10mm, 速度为 20mm/s, 等电机运行完才运行下面语句 固定延迟 3 秒 Y 电机正转 30mm, 速度为 40mm/s, 等电机运行完才运行下面语句 程序结束语句

效果: 按 X1 启动按钮后, X 电机正转(距离: 10mm, 速度 20mm/s, 数值从菜单内读出), 正转完后停留 3 秒, 然后 Y 电机正转(距离: 30mm, 速度 40mm/s, 从变量读出, 而变量的值从菜单内读出), 程序结束。

这里 Y 轴为什么要变量读出? (只是举例而已, 当然也可以直接从菜单读出)

电机语句的数值从菜单读取只能从【0 菜单】至【8 菜单】。那【8 菜单】以后的菜单怎么办呢? 其实后面的菜单也是可以作为电机的使用。方法如上文 Y 轴的演示。意思就是需要用变量(a, b)作为一个媒介, 起到传输的作用。PS: 其实菜单 8 之前的原理, 系统内部也是用这种方法而已的。

因此: 关于电机的菜单创建, 尽量**靠前**创建(方便)。后面的菜单则需要用变量作为媒介。变量赋值在程序当中很浪费系统 CPU。所以一套程序当中, 尽量只赋值一遍即可。一般把它放在程序运行之前赋值好。

4.3.9 联动【XY 电机】【回零】同时运行和分开运行

程序内容	解释
子程序 A---0 线---X1 触发 X 机回零 003.00 速 010.00 等停 Y 机回零 003.00 速 010.00 等停 X 机正转 040.00 速 040.00 等停 Y 机正转 050.00 速 070.00 等停 03.00 秒 X 机反转 020.00 速 060.00 不等 Y 机反转 030.00 速 050.00 等停 =====结束或返跳	程序开头语句。程序配定在 0 号 CPU 内运行, X1 启动该程序。 X 电机回零校准, 速度为 10mm/s 等停: 等电机运行完才运行下面语句 Y 电机回零校准, 速度为 10mm/s 等停: 等电机运行完才运行下面语句 X 正转 40mm, 速度为 40mm/s 等停: 等电机运行完才运行下面语句 Y 正转 50mm, 速度为 70mm/s 等停: 等电机运行完才运行下面语句 固定延迟 3 秒 X 反转 20mm, 速度为 60mm/s 不等: 不等电机运行完就运行下面语句 Y 反转 30mm, 速度为 50mm/s 等停: 等电机运行完才运行下面语句 程序结束语句

等停: 等电机运行完才运行下面语句 (就是说当运行下面的程序时, 电机肯定停下了)

不等: 不等电机运行完就运行下面语句 (就是说当运行下面的程序时, 电机还在运行)

因为程序是从上往下流水式运行, 每行间隔时间 00.01 秒!

所以【X 机反转 020.00 速 060.00 不等】到【Y 机反转 030.00 速 050.00 等停】2 电机同时启动间隔为 0.01 秒, **视为同步运行**

效果 按 X1 启动按钮后，先 X 电机回零校准，完成到位后，再 Y 电机回零校准。等 Y 电机校准完成后 X 电机正转到位置 40mm 处，完成后，再 Y 正转到位置 50mm 处。等 Y 电机到位后，延迟 3 秒。然后 X 电机返回，在 X 电机返回开始时，Y 电机也同时开始返回（先后 2 个电机启动时差为 00.01 秒。2 电机可视为同时返回）。等 Y 返回到位后，程序结束！

在这个例程中，首次用到回零语句，解释一下【X 机回零 003.00 速 010.00 等停】的意思：

X 机回零：顾名思义就是 X 电机**反转并回零**。机械零点：反转寻找零点限位开关。软件零点：直接回到 0mm

003.00：机械零点：在限位开关前 3mm 位置内减速下来，慢速寻找感应器。软件零点：直接回到 0mm

速 010.00：在限位开关前 3mm 位置以外的反转运行速度

等停：等电机回零动作完成，才运行下面的语句。

注意：当电机执行当前任务的时候，又让同个电机执行下面其他的动作，这时系统将舍弃下面的任务。所以在写电机不等的时候，一定要理清动作逻辑。如下图：

错误写法 ×	正确写法 √
X 机正转 xx 速 xx 不等 Y 机正转 xx 速 xx 不等 X 机正转 xx 速 xx 不等	X 机正转 xx 速 xx 不等 Y 机正转 xx 速 xx 不等 X 机 等停
效果 :因为都是不等,所以程序会马上触发这 3 条语句,但 X 轴触发了 2 次的运行,换句话说,当运行下次 X 轴的时候,上一次的 X 电机任务还没结束。此时系统会舍弃下面的 X 轴的任务	效果 :当 X 轴运行的时候,因为不等,马上触发 Y 轴的运行,然后等待 X 轴运行完,才进行下 1 次的 X 轴的动作任务。所以这里的【X 机 等停】是很有必要

4.3.10 【XY 电机】与【输出（气缸）】的同时控制

程序内容	解释
子程序 A---0 线---X1 触发	程序开头语句。程序配定在 0 号 CPU 内运行，X1 启动该程序。
X 机回零 003.00 速 010.00 等停	X 电机回零校准，速度为 10mm/s 等停：等电机运行完才运行下面语句
Y 机回零 003.00 速 010.00 等停	Y 电机回零校准，速度为 10mm/s 等停：等电机运行完才运行下面语句
Y00=1 (电磁阀)	Y00 电磁阀开启
X 机正转 040.00 速 050.00 等停	X 正转 40mm，速度为 50mm/s 等停：等电机运行完才运行下面语句
Y01=1 (指示灯)	Y01 指示灯开启
Y 机正转 050.00 速 070.00 等停	Y 正转 50mm，速度为 70mm/s 等停：等电机运行完才运行下面语句
Y00=0 (电磁阀)	Y00 电磁阀关闭
03.00 秒	固定延迟 3 秒
Y02=1 (报警灯)	Y02 报警灯开启
X 机反转 020.00 速 060.00 不等	X 反转 20mm，速度为 60mm/s 不等：不等电机运行完就运行下面语句
Y 机反转 030.00 速 030.00 等停	Y 反转 30mm，速度为 30mm/s 等停：等电机运行完才运行下面语句
Y01=0 (指示灯)	Y01 指示灯关闭
Y02=0 (报警灯)	Y02 报警灯关闭
=====结束或返跳	程序结束语句

效果：按 X1 启动，先 X 电机进行回零校准，完成后，再 Y 电机进行回零校准，Y 电机校准完成后。电磁阀（Y00）开启，**同时** X 电机正转（距离 40mm，速度 50mm/s），等 X 电机走完停下。指示灯（Y01）开启，**同时** Y 电机正转（距离 50mm，速度 70mm/s），等 Y 电机走完停下。电磁阀(Y00)关闭。延迟 3 秒后，报警灯(Y02)开启，并**同时** X 电机反转（距离 20mm，速度 60mm/s），在 X 电机反转开始时，Y 电机也开

始同时反转（距离 30mm，速度 30mm/s）。等 Y 电机反转到位后。指示灯（Y01）关闭，同时报警灯(Y02)关闭。程序结束！

4.4 *控制器高级功能的演示（按需学习）

4.4.1 逐级改速与坐标清零的方法

上文中我们学会了变量（a-z）赋值的方法，因此将进行更深一步的控制延伸：

程序内容	解释
子程序 A---0 线---X1 触发	程序开头语句。程序配定在 0 号 CPU 内运行，X1 启动该程序。
a->系统 00: X 轴当前位置	把 a（默认是 0）赋值给系统 00 功能区，即 X 轴当前的坐标
X 机正转 050.00 速 005.00 不等	X 轴正转 50mm。速度 5mm/s 不等：无需等电机停下，就运行下面语句
循环开始 00003 次	设置循环（上文中举例），次数 3 次
01.00 秒	延迟 1 秒
X 机 改速 b	修改 X 轴运行中的速度。速度值为读取 b 变量的数值
b = b+1000	运行加法语句。b（默认是 0），自加 1000。则循环 3 次分别为 1000，2000，3000
循环结束	程序结束语句
=====结束或返跳	

效果：先把 X 轴坐标清零（系统 00 功能区看《详细说明书》-互相赋值语句的介绍）。然后开启 X 轴旋转。因为不等，故可以马上运行下面的语句。下面的内容是个循环，循环内容是：先延迟 1 秒，然后改 X 轴的速度。**改速语句的要求（1：改速前，电机必须是运行的 2：电机运行后，不能马上改速）**，改速的速度值读取 b 变量的数值。而 b 变量的数值是自加的，即越加值越大。以上内容循环 3 次。所以 b 值的变换过程为 1000-2000-3000，填入到改速的值里面为 010.00-020.00-030.00。因此最终结果为，X 轴最初的速度为 5mm/s，以后是 010.00-020.00-030.00，改速变换时间为 1 秒。

Ps：清零功能使用后，系统就认为清零后的位置为 0MM。如使用的是软件零点，则系统回零就到该位置。如使用的是机械零点，因为回零时必须碰到限位开关，系统才标记为 0MM。因此如果在中途坐标被清零，在回零的时候系统刚开始认为中途的位置是 0MM。但会判断到这个中途 0MM 的位置没有找到感应器，故系统还会继续寻找感应器，但此时寻找的速度为寻零速度（很慢）

4.4.2 多个程序同时运行（存多套程序）

用途：应用于 2 套以上完全不同的设备独立工作。所有的动作轨迹和参数都是完全不同

程序内容	解释
子程序 A---0 线---X1 触发	程序开头语句。程序配定在 0 号 CPU 内运行，X1 启动该程序。
-----程序内容-----	
=====结束或返跳	程序结束语句
子程序 B---1 线---X2 触发	程序开头语句。程序配定在 1 号 CPU 内运行，X2 启动该程序。
-----程序内容-----	
=====结束或返跳	程序结束语句

子程序 C---2 线---X3 触发 -----程序内容----- =====结束或返跳	程序开头语句。程序配定在 2 号 CPU 内运行，X3 启动该程序。 程序结束语句
---	--

解释: 子程序 A---0 线---X1 触发，其中子程序 A 为程序名（所有程序名不得重名）。0 线为 CPU 号（共 0-4 号，控制器允许 5 个 CPU 同时运行）。X1 为触发此子程序的条件。

效果: 开机子程序 A、B、C 这 3 个程序已经同时运行（因为都是在不同 CPU），只是都在等待触发条件，一旦触发就开始独立运行！

更多【子程序 A---0 线---X1 触发】查阅 BR010-《详细产品手册》-子程序语句的介绍

4.4.3 【*重要】如 1 那跳 00 否则跳 00 不返（跳转/判断语句）

程序内容	解释
00 子程序 A---0 线---X1 触发	A 程序开头语句。程序配定在 0 号 CPU 内运行，X1 启动该子程序 A
01 xx 其他程序内容 xx	第 1 行的程序内容
02 xx 其他程序内容 xx	第 2 行的程序内容
03 如 X0 那跳 +01 否则跳 -02 不返	判断 X0 有无感应，有感应：运行+01，否则无感应运行-02 (+01,-02 代表程序行数)
04 xx 其他程序内容 xx	第 4 行的程序内容
05 xx 其他程序内容 xx	第 5 行的程序内容
=====结束或返跳	程序结束

解释: 判断 X0 有无感应（即判断 0 还是 1），有感应（1）：运行+01，无感应（0）：运行-02，其中+01 表示跳到该判断语句的下面 1 行，即第 04 行。-02 表示跳到该判断语句的上面 2 行，即第 01 行。所以这条语句意思就是判断 X0 感应器，有感应则程序往下运行，无感应则重复运行 01 至 03 的语句，直到有感应为止。另外如 05（没有+或-）表示跳到程序第 05 行。关于该语句的介绍查阅 BR010-《详细产品手册》

4.4.4 多个程序自动的先后运行与同时运行



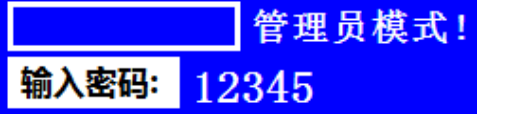
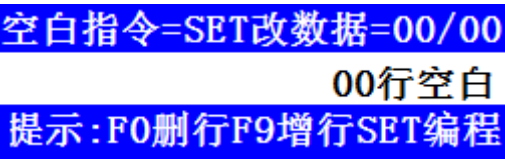

程序内容	解释
子程序 A---0 线---X1 触发 xx 其他程序内容 x	A 程序开头语句。程序配定在 0 号 CPU 内运行，X1 启动该子程序 A
如 1 那跳 B00 否则跳 00 返回 如 1 那跳 C00 否则跳 00 后台 xx 其他程序内容 xx =====结束或返跳	暂停 A 程序，跳到 B 程序运行 B 程序内容。B 程序运行完之后，跳回来继续运行 A 程序的内容 开启 C 程序。因为是后台运行。因此 A 与 C 两个程序是同时进行的。
子程序 B---1 线---无触发 -----程序内容----- =====结束或返跳	B 程序开头语句。程序配定在 1 号 CPU 内运行，无触发:其他启动方式 程序结束语句。
子程序 C---1 线---无触发 -----程序内容----- =====结束或返跳	C 程序开头语句。程序配定在 1 号 CPU 内运行，无触发:其他启动方式 程序结束语句。

效果: 开机按 X1 启动子程序 A，A 程序运行到中间的时候，执行【如 1 那跳 B00 否则跳 00 返回】语句，

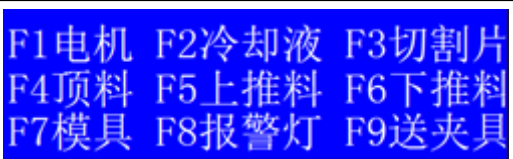
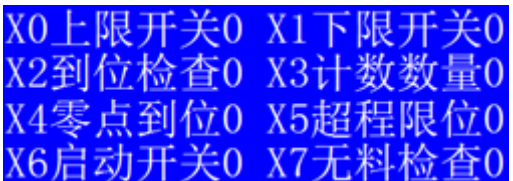
因为该语句为带**返回**属性的跳转语句，所以此时暂停 A 程序跳到 B 程序，运行 B 程序的内容。B 程序碰到【=====结束或返跳】语句时，表示已运行完 B 程序。则跳回来继续运行 A 程序的内容。当程序执行【如 1 那跳 C00 否则跳 00 后台】语句时，开启 C 程序。因为该语句为带**后台**属性的跳转，因此 A 与 C 两个程序是同时进行的（2 个程序必须为不同的 CPU）。各自的子程序都碰到相应的【=====结束或返跳】语句后，各自结束工作！

第5章 附:脱机编程的简介

5.1 程序编程进入方法

第一步	开机启动默认进入 用户模式 界面！ 此界面不能编程，仅适合【设备的使用者(工人)】使用。	
第二步	在用户模式主界面下，按 确认 键3秒，进入 管理员模式 ！ 此界面不能编程，合适【调机与维修，厂方管理员】使用	
第三步	在管理员模式主界面下，按 编程 键，自动弹出进入 程序员 界面的密码（默认为12345），按 确认 键进入 程序员模式 。 此界面拥有控制器最高的权限，可编程序动作！	
第四步	进入程序员模式后，如需编程，按 编程 键进入程序编辑界面。如无需编程，按 返回 键回到程序员主界面。 右图就是程序编程界面。	
第五步	程序员模式主界面状态图 如要退出程序员模式，重启系统即可。	

5.2 编程步骤与操作



第一步 如不想给输出起设备名称，可跳过该步骤	在 输出测试 界面里命名对应的设备名称 操作方法: 在程序员模式主界面下，按 手动测试 进入界面。按 \blacktriangle 可换行，按 确认 进行文字修改（输入方法查看下文）。 支持文字连打，即 F1, F2, F3 的名字是一起打的，系统会自动分割文字长度，并根据格式对号入座！ 按 返回 可返回上级界面 无需编程，系统即可进行输出的手动测试（右图）	 <p>对应接口说明：【F1-Y0】【F2-Y1】【F3-Y2】【F4-Y3】【F5-Y4】【F6-Y5】【F7-Y6】【F8-Y7】【F9-Y8】【F0-Y9】【校准-Y10】</p>
第二步 如不想给输入起设备名称，可跳过该步骤	在 输入测试 界面里命名对应的设备名称 操作方法: 在程序员模式主界面下，按2次 手动测试 进入界面。按 \blacktriangle 可换行，按 确认 进行文字修改（输入方法查看下文）。 支持文字连打，道理与输出的一样 按 返回 可返回上级界面 使用说明: 此功能无需编程就可使用，【0-表“无信号”输入】，【1-表“有信号”输入】	 <p>对应接口说明:【X0-X0】【X1-X1】【X2-X2】【X3-X3】【X4-X4】【X5-X5】【X6-X6】【X7-X7】</p>
解释：为什么要事先改好输出、输入的各种设备名称？ 1: 当输入输出有对应名称后，控制器就会进行绑定。以后用户进行 手动测试 和 电路接线调试 时，会很直观 2: 程序编程的时候，调用输入输出时，会自动显示对应名称，而不是单调的编号，让编程起来更加清楚方便		

<p>第三步 如以后调参数，只需在程序内修改，而不需要在菜单内进行的，可跳过此步骤</p>	<p>在菜单界面里预先创建好菜单项：（右图）</p> <p>由 菜单名称、显示格式、菜单属性三部分组成</p> <p>操作方法:在程序员模式主界面下，按设置进入菜单界面。按↵可换行，按确认进行文字修改（输入方法查看下文）。按←可更改菜单数据格式，按→可更改菜单属性。菜单的数据格式和属性请查看网页 http://btcnc.net/web/2moto/WEB/6.htm</p> <p>菜单数据的修改 选中某行菜单项，按设置此时数据上会出现小光标01.00，按←可移动光标，按↵可修改光标内数值，或直接输入 F1-F9 的数值。数据修改完毕，按确认小光标消失，修改成功！</p>	 <p>注意事项: 菜单屏幕的最上行有对应菜单项的行数显示（画线部分），此行数用于程序调用该菜单数据时的行数。</p> <p>文字编辑小技巧：当菜单文字内容相差不多时，可按F7进行复制，再按F8进行粘贴。</p>
<p>第四步 [必须用]</p>	<p>在程序界面里编写动作流程</p> <p>操作方法:在程序员模式主界面下，按编程进入编程界面。按↵可换行，按F9可增加一行程序，按F0可删除一行程序。</p> <p>编程方法:按确认或设置此时会在选中的程序上出现小光标，按⏪可选择控制器所有的指令。选中一条指令后，按←或→可移动小光标，按确认或设置可修改光标的数值。程序编程好，按返回小光标消失，再按返回提醒是否保存？按确认保存，程序编程完毕！退出到主界面。登入该说明书操作视频可观看演示</p>	 <p>注意事项: 一行程序对应填写一条指令，共可填写 100 条指令。在屏幕的最上行有总行数和当前行数的显示。</p> <p>程序是从上往下流水式运行，所以在编程之前，要事先整理好编程的动作。</p> <p>编程小技巧：当指令内容相差不多时，可按F7进行复制，再按F8进行粘贴</p>
<p>第五步 [可不用]</p>	<p>自定义主界面的文字和显示布局，让控制器更加符合你的风格。一般不建议客户去修改</p> <p>请查看网页：http://btcnc.net/web/2moto/WEB/6.htm</p>	
<p>第六步 [可不用]</p>	<p>编程完成，是否需要添加控制器的固定功能</p> <p>使用方法:本册第 3 章-3.3.3 （第 10 页）有介绍</p>	<p>如 1:使用次数的限制 2:管理员密码的修改 3: 是否要备份数据 等等</p>

5.3 *文字输入方法介绍（附录）

前言:因储存空间有限，故字体并不齐全，可向厂家联系添加！

输入对象:【输出设备的名称】【输入信号的名称】【菜单内容的名称】【主界面的文字】

 <p>首字母</p> <p>对应文字</p> <p>已输入的文字</p>	<ol style="list-style-type: none"> 选中要命名的目标，按确认弹出输入框，开始输入文字 删除已输入的文字：按F0或ESC 移动光标：F5与F6 按⏪进行首字母选择。按↵或←进行对应文字的选中 选中要输入的文字，按设置键确认输入。 <p>注：如系统内没有该字体，会出现 1 个“斧头”的图标</p> <p>5: 输入完毕，按确认完成输入并返回到界面</p>
<p>空格的输入: </p>	