



步控 开发平台使用说明

BK_Dev1.2



东莞市步控电子科技有限公司

技术支持 QQ: 3081562115 公司电话: 0769-33329550

目录

- 1. 步骤程序..... 2
 - 1.1 编程简介..... 2
 - 1.2 系统架构..... 2
 - 1.3 优点..... 3
- 2. BK_Dev..... 4
 - 2.1 BK_Dev 开发平台简介.....4
 - 2.2 主界面..... 5
 - 2.3 配置窗口..... 11
 - 2.4 选项框..... 14
 - 2.5 程序按钮..... 17
 - 2.6 编辑.....22
 - 2.7 对象.....24
 - 2.8 编译.....29
 - 2.9 下载.....31

1. 步骤程序

1.1 编程简介

步骤程序的编程方式是选择式，填表式。

步骤程序的编程思维，理解方式，执行方式，类似于汇编语言，C 语言，C++ 等通用电脑编程语言。不同于 PLC 梯形图。

程序相当于转换为 C 语言或汇编语言。然后下载到控制器执行。可以用汇编语言 C 语言及一些通用电脑编程语言的思维去理解步骤程序，不能按梯型图的编程思维去理解步骤程序。

1.2 系统架构

整个步骤编程系统的原理是，将程序分为两部分，系统程序和应用程序。系统程序类似于电脑的操作系统，这部分事先做好并下载到控制器中。应用程序就类似于电脑的应用软件，就是用户的步骤程序编译而来，再将应用程序下载到控制器中。最后控制器中的应用程序和系统程序交互执行，应用程序调用系统程序的底层驱动，驱动硬件。系统程序对应用程序进行管理，实现多任务。

步骤编程系统是多任务的，所以步骤程序里的每个任务都是同时执行的。这点类似于 C 语言里的多线程。如果任务之间需要互相通信的话，则可利用公共对象区里的公共变量。（如下图）



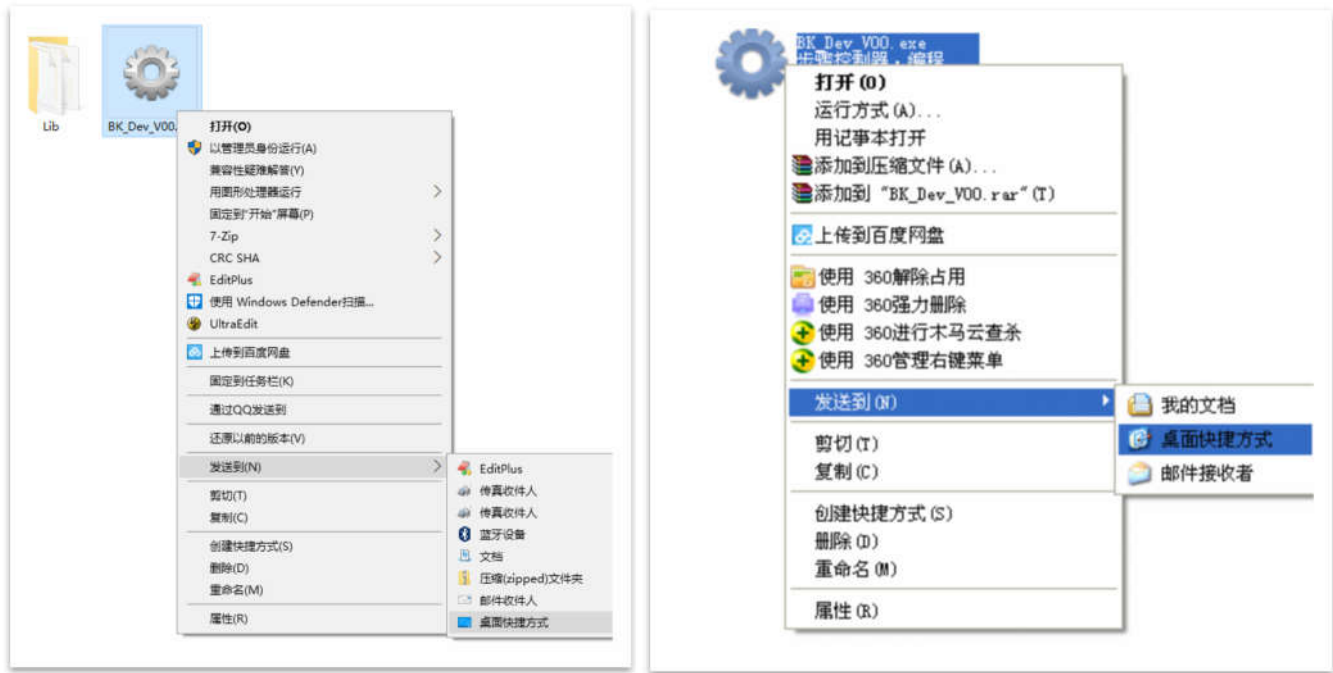
1.3 优点

- 1、全中文编程，编程无障碍。
- 2、选择式编程，每个选项都有提示，不用记任何命令。
- 3、多任务编程，每个任务独立运行就是嵌入式操作系统里的一个线程。
- 4、面向功能，以功能为主。编程主要是告诉控制器要执行的功能，用户不用介入每个功能的实现过程。
- 5、表格式查看方便，理解容易，非常适合需要编制简单动作程序，而又没有任何编程基础的人。

2. BK_Dev

2.1 BK_Dev 开发平台简介

BK_Dev 开发平台是一款免安装的绿色软件。将压缩文件解压到任何目录下都可以运行。建议将其在桌面上建立一个快捷方式，方便后续使用。

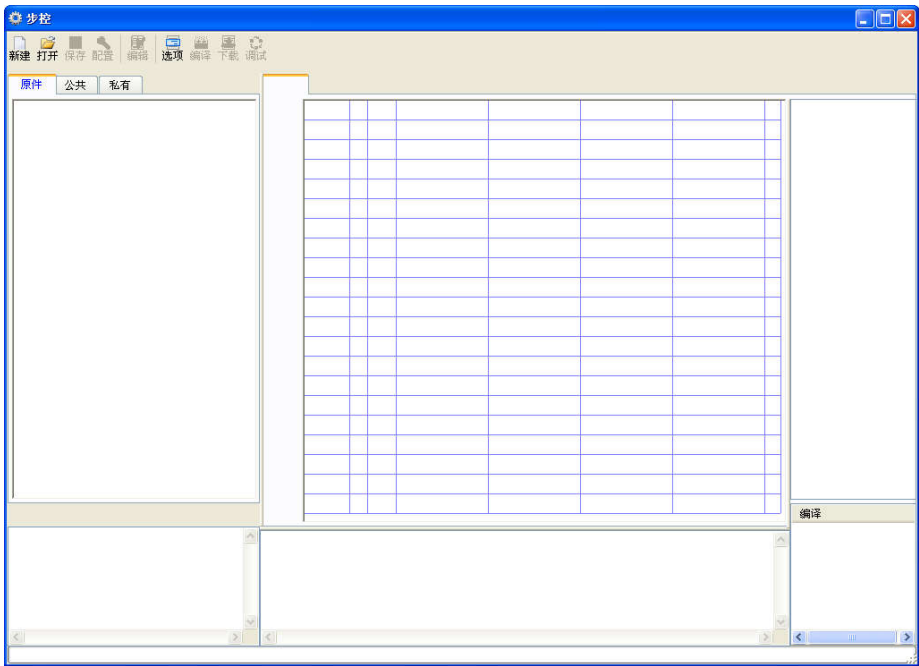


在 BK_Dev 的目录下，有个名为” Lib” 的文件夹, 请不要删除此文件夹，并让其和 BK_Dev 在同一文件夹下。



有些杀毒软件会误认为此软件为病毒。使用时请临时关闭杀毒软件。

2.2 主界面



以上是刚打开软件的初始界面。如果打开一个任务后，其各个组成部分如下。



标题栏：最上面最蓝条部分, 会显示当前已被打开的项目的控制器型号, 以及被打开的项目的路径。

工具栏：工具栏上的命令按钮。直接根据其名称即可知道其功能。

〈新建〉就是新建一个项目，建议为每个项目建立一个独立的文件夹。也就是在你想建立项目的位置，先建立一个文件夹，将文件夹命名，然后再进入文件夹中建立项目。



工具栏上的〈配置〉和〈选项〉两个按钮。〈配置〉是对项目程序进行配置。而〈选项〉则是对下载调试连接进行设定。其区别为：与打开的项目相关的设置都在配置中，比如项目的控制器型号。与打开的项目无关的设置都在〈选项〉中，比如下载的端口号，波特率。

对象区：所有能选择对象都再此区域类。先选中程序区中的对象列，然后就再点击对象中的对象即可将对象选取到对应的对象列中。

对象区有 3 个部分：原件，公共，私有。

原件：原件对象区包含该控制器所具有的所有对象。可直接选用。当然有写是不能直接选用的。比如变量里的对象都不能直接选用。需要定义为一个确切的变量对象后才能用。

公共：在公共对象区可定义所需的对象。比如可以定义一个对象，其名称为“累计个数”。然后将其定义为暂存。那么“累计个数”是一个暂存对象了。也可定义为指定的具体对象，比如可以定义一个对象，其名称为“上升气缸”。然后将其定义为 Y00。那么这个“上升气缸”就对应到 Y00。开启或关闭“上升气缸”，实际上就是对 Y00 进行开启或关闭。在公共区定义的对象在所有任务中都可以看到和使用，相当于 C 语言里的全局变量。

私有：私有对象区也可定义所需的对象，与公共对象区不同的是：在私有对象区定义的对象只能在本任务中都可以看到和使用，相当于 C 语言里的局部变量。

对象区的底部有 4 个按钮：〈新建〉，〈↑ 上移〉，〈下移 ↓〉，〈删除〉。这 4 个按钮用于在〈公共对象区〉或〈私有对象区〉里的对象。对于〈原件对象区〉，这 4 个按钮无效。

任务栏：任务栏会列出项目所包含的全部任务。每个项目最多可以有 16 个任务。项目的添加移除等需要到〈配置〉中操作。

程序区：程序区里的程序是表格式的。分为很多列。

地址列: 左边第一列, 可以在其中输入任何名称, 用于标示所在行的地址, 相当于 C 语言或汇编语言里的标号。选中后下面的主说明框会显示所在行的注解。

控制列: 第二列, 会空起来或者显示 ‘等’ ‘如’ ‘是’ ‘否’。用于流程控制。

命令列: 第三列, 表示所在行的命令方式。‘条件’ ‘动作’ ‘跳转’。

然后往后分别为对象列, 方法列, 参数 1, 参数 2。

			任务			
			新步骤			
	如					
		条件	输入x00	接通		
	是					
		动作	输出y00	开启		
	否					

对象列: 所在行的命令主体。选中后, 然后在点击对象区里的对象, 即可将被点击的对象选取到对鞋列。对象只能通过选取, 不能打字输入, 编辑修改。

方法列: 该对象所具有的所有方法都列出来供选择。用户不用记得每个方法的具体名称, 只要知道其含义即可。即便不知道其含义, 选中后主说明框也会提示被选中的方法的含义以及用法。

参数 1: 有的方法有参数, 有的没有参数。若没有参数, 此列就会显示灰色。如果有的话, 可以直接输入数值, 也可像对象列一样选择对象。

参数 2: 如果所在行的命令是动作的话, 其含义更参数 1 一样, 有的命令有可能需要两个参数, 就可以直接输入数值, 也可像对象列一样选择对象。如果不需要两个参数的话, 此列就会显示灰色。

如果所在行的命令是单独条件的话，此列就会显示灰色：

	如				
		条件	输入X00	接通	
	是				
		动作	输出Y00	开启	
	否				

如果是组合条件的话(下面一行也是条件), 那么就会出现选择框，供选择同下面一行条件的关系：

			等待信号		
	如	条件	输入X01	接通	#并且
		条件	输入X02	接通	#并且
	是				#或者

			等待信号		
	如	条件	输入X01	接通	#并且
		条件	输入X02	接通	
	是				

如果所在行的命令是单个跳转的话，那么参数 2 就表示需要跳转的目标地址。
目标地址可以是步骤，也可以是本步骤内的某个地址(以 ‘@’ 开头)。如果所在行的命令是组合跳转的话(下面那一行也是跳转)，就会像条件一样，出现选择框，供选择同下面一行跳转的关系(是 ‘并且’ 还是 ‘或者’ 的关系)。



步骤列表：程序中的步骤会很多，为了方便管理，所有的步骤都会在列在步骤列表中。双击的话程序区会立即显示对应的步骤。方便查看！

对象说明框：当选中对象框里的对象时，对象说明框中会立即出现此对象的说明文字。方便用户理解该对象的含义及用法。

主说明框：当选中程序区里的对象时或者方法时，主说明框中会立即出现说明文字。方便用户理解其含义及用法。

当选中程序区里的地址栏时，主说明框会显示所在行的注释，注释可以修改。
编译完成后，会在主说明框中显示编译信息。

编译框：编译完成。如果有错误，编译框会以列表的形式将错误列出来。用户双击错误。对象区或程序区会立即定位到错误处。方便排除错误！

2.3 配置窗口

新建一个项目时，或者点击工具栏上的配置按钮就会出现如下窗口！



控制器型号下拉框用于给用户选择合适的型号，选择的型号与实际使用的型号如果不一致的话会导致程序下载不进去，不能调试。

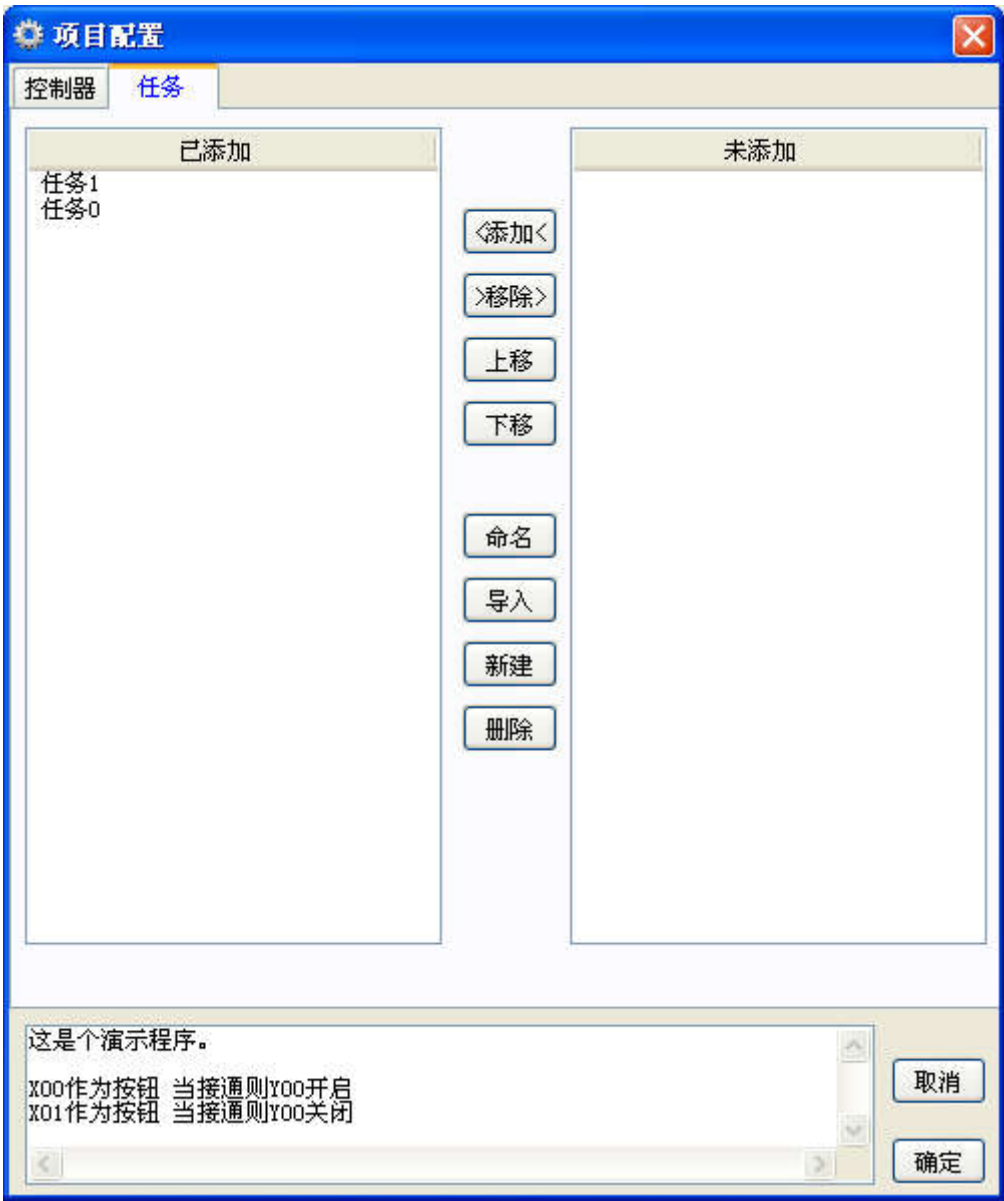
图片框里会出现项目对应的控制器型号的图片。

控制器说明框会对该控制器进行说明，方便用户选型。

最下面的是项目说明框，是本项目的注解，用户在此处写上此项目的说明，以

及一些重要事宜，备忘！

如果选中<任务>，则会出现如下界面：



项目文件所在的目录下的全部任务都会列出来。如果已经被添加进项目则会显示在左边<已添加>列表中，否则就出现在右边<未添加>列表中。

通过中间的按钮可以对任务进行操作。

通过<上移><下移>可以给已添加的任务排序。

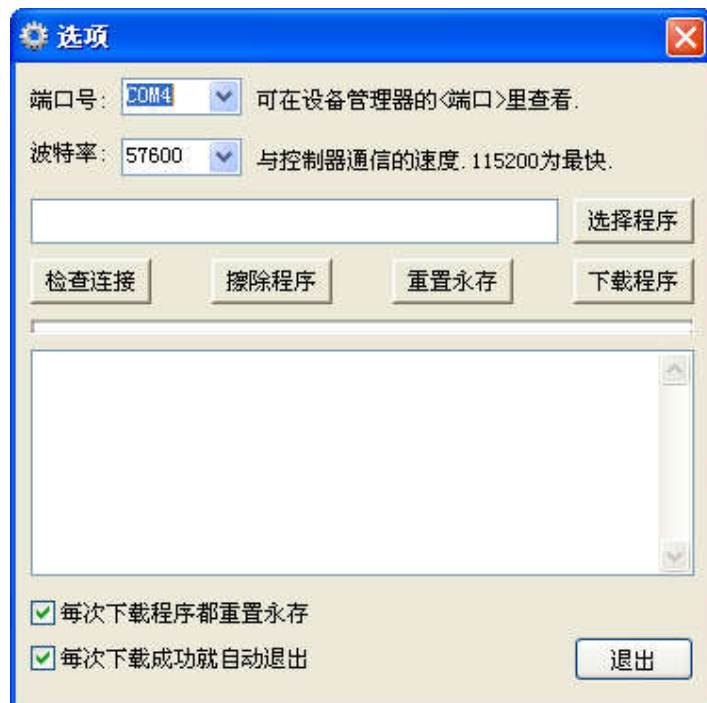
导入的意思是从其它文件夹复制到本项目文件夹里面。比如建立了一个项目，

我们需要一个任务，而那个任务在别的项目已经有，那么就可以导入过来。导入时如果本项目文件夹下已经有同名的文件，那么被导入的任务就会自动重命名。也可导入本项目文件夹下已有的任务，相当于复制一份任务。

移除只是不包含进项目，任务本身还是存在的，后续如果需要还可以被添加。而删除就是将任务彻底消灭，删除了就没有了。

2.4 选项框

点击工具栏上的选项按钮就会出现如下选项窗口：



端口号：通常情况程序能自动检测到端口，检测的原理就是先看选中的端口是否有效，如果有效就认为此端口，否则就从 COM1 到 COM16 依次扫描。只要检测到有效端口就用此端口。所以如果系统中有多多个串口，则自动检测到的端口可能并不是我们用来下载的串口，那就只能手动选择了端口号了。

端口号需要到设备管理器里查看，请先确定安装了 USB 转串口工具，打开设备管理器，然后插入 USB 转串口，即可看到对应的端口号。

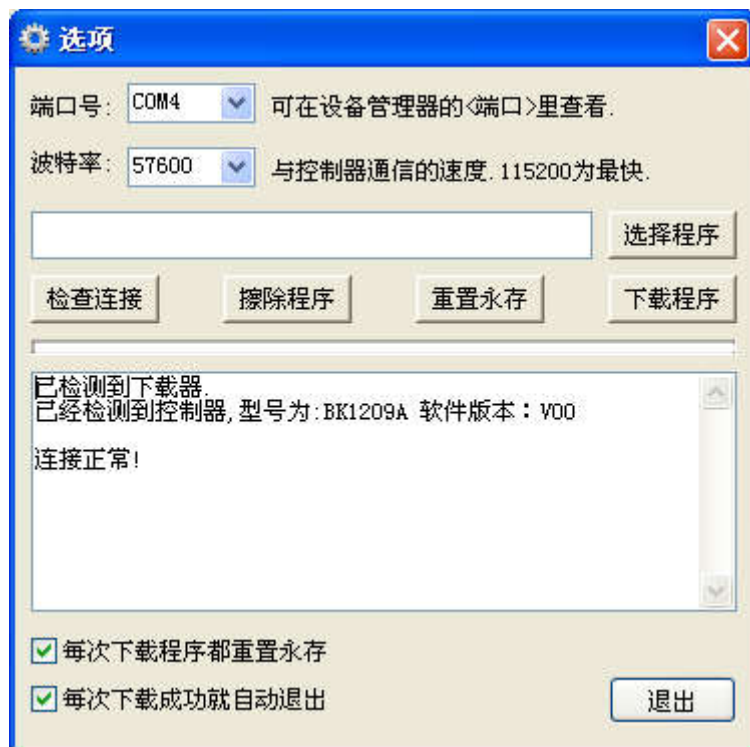


波特率：此波特率指的是下载程序的波特率，根据测试。最大只能达到 57600。如果程序量不大的话，建议用稍微小一点的波特率。否则有时有可能下载失败。如果下载失败，可以将波特率调小再重试(点击按钮<下载程序>)。也可以直接重试，也有机会下载成功！

选择程序：按下<选择程序>按钮，则立即出现文件打开对话框，用于打开想下载的程序。打开后，会在前面的编辑框中显示被打开程序的路径。

下载程序：程序选取后按下<下载程序>按钮，则立即下载程序。

检查连接：用于检查下载线及控制器是否接通。同时可以看出连接的控制器型号。如果连接正常，则显示如下：



擦除程序：将程序擦除，控制器内部就没有程序了。

重置永存：永存就是掉电保持存储器。其本身可以设个初始值，重置永存就是将其恢复为初值。比如有个永存，里面保存的是还能使用的次数，每次上电将其减一。如果为零，则不再动作。那我们需要将其赋个初值(比如 100)才可以。用了几次后我们若想将其恢复为初值，那么只需执行重置永存即可。

勾选框：如果想每次下载程序的时候的重置永存的话，勾选<每次下载程序都重置永存>。成功下载完程序后，下载窗口还留着就没什么用了，如果需要自动退出的话，勾选<每次下载成功就自动退出>。

2.5 程序按钮

在程序区的左侧有如下这些按钮：



步骤: 为程序添加一个步骤，如果已经选中了程序中的某一行，则会在其所在的步骤后面再添加一个步骤。没有选中某一行，则会在程序的最后面添加一个步骤。步骤的名称就是此步骤的标示，所以同一任务下，步骤不允许重名。步骤相当于一个段落。将程序分为多个步骤，能够更清晰的理解程序的流程。

等待：就是等待其内部的条件满足了才继续往下执行，否则就一直等。

如下程序，如果 X00，X01 同时接通则往下执行，开启输出 Y00。

等	条件	输入X00	接通		#并且
	条件	输入X01	接通		
	动作	输出Y00	开启		

相当于 C 语言：

```
while(1)
{
    if(Check1(xx,xx))
    if(Check2(xx,xx))
        break;
}
```

可以看出，如果条件满足则继续往下执行，否则就在此死等。如果两个条件是#或者的关系的话，则相当于如下 C 语言：

```
while(1)
{
    if(Check1(xx,xx))
        break;
    if(Check2(xx,xx))
        break;
}
```

也就是说，最后一个条件，一定是相当于：

```
if (Check (xx, xx))
    break;
```

而前面的条件，如果与下一条的关系是#或者，也相当于：

```
if (Check2 (xx, xx))
    break;
```

前面的条件，如果与下一条的关系是#并且，那么就没有“break”；仅仅相当于：

```
if (Check2 (xx, xx))
```

所以如果你懂 C 语言的话，请记住，“等待”实际上就是一个循环。里面的条件满足就会跳出此循环，然后往下执行。

如果“等待”中只有一个条件，如下图，等待 X00 接通，然后开启输出 Y00。

			新步骤		
	等				
		条件	输入X00	接通	
		动作	输出Y00	开启	

这样的写法是完全合理的。只是通常如果只有一个条件的话，我们希望就不要添加“等待”了。而是直接这样：

			新步骤		
		条件	输入X00	接通	
		动作	输出Y00	开启	

这种方式就是前面那种方式的简化版，两者具有完全相同的效果。而后者占用空间更少，更简洁。如果只是等待单个条件的话建议用后者。

如果：“如果”也是判断其内部的条件是否满足，与“等待”的区别在于，“等待”是当其内部的条件满足了才继续往下执行，否则就一直等。而“如果”则是不论内部条件是否满足都会往下执行，只是执行的路径不同，满足条件则往“是”这

边执行，不满足条件则往“否”这条路执行。所以“如果”的后面都会跟着“是”和“否”。

			新步骤			
	如					
		条件	输入X00	接通		
	是					
		动作	输出Y00	开启		
	否					
		动作	输出Y00	关闭		

其相当于 C 语言：

```
if (InputX(X00, ON))
{
    OutputY(Y00, On) ;
}
else
{
    OutputY(Y00, Off) ;
}
```

“如果”的内部也可以有多个条件，比如：

			新步骤			
	如	条件	输入X00	接通		#或者
		条件	输入X01	接通		
	是					
		动作	输出Y00	开启		
	否					
		动作	输出Y00	关闭		

两个条件之间是#或者的关系，那么相当于 C 语言的：

```
if (InputX(X00, ON) || InputX(X01, ON))
{
    OutputY(Y00, On) ;
}
else
{

```

```
OutputY(Y00, Off);  
}
```

不论执行的“是”这边还是“否”这边，执行完后都会接着“否”的后面继续往下执行。“是”和“否”必须和“如果”一起配套，“是”和“否”内部允许为空。

动作，条件，跳转：此三项是命令，表示执行的方式，“动作”就是做需要执行的具体事项，比如‘开启’‘关闭’。而“条件”则是检测，在前面已提到过。“跳转”，跳转分为条件跳转和直接跳转：



上图的意思是，如果 X00 和 X01 任意接通一个就跳转到步骤<开启 Y00>。如果没有的话，则又直接跳转到<循环检测>处。前面两个跳转是条件跳转, 都是有条件的，条件满足才跳转。后面的是直接跳转，不需要条件。

灵活的运用跳转就可以做全部流程控制，比如“等待”，“如果”控制都可以用“跳转”来实现，在 C 语言中，“if”语句的里面允许嵌套。而步骤程序不允许嵌套，如果需要做嵌套，那么就可以用“跳转”来实现。

通常情况下，选择单元格，然后点击某个命令按钮(“条件”“动作”“跳转”)，即可在选中行的后面添加对应的命令。

如果选中的行是个流程控制内部的命令，比如“等待”“如果”“是”“否”。那

么点击某个命令按钮则还是将命令添加到流程控制内部。如果需要添加到控制的后面，则需选择控制栏，然后在点击某个命令按钮。

如果要添加一个“动作”到“等待”后面，那么需要先选中“等”，然后点击“动作”，就会将“动作”添加到“等待”的后面。如下图：

步骤			新步骤	
等待				
	1 等	条件	输入X00	接通
如果		条件	输入X01	接通
		动作	3	被添加到等待的外面
2 动作		等	条件	输入X00
条件		条件	输入X01	断开
添加动作		动作	输出Y00	关闭
跳转		跳转		新步骤

添加一个“条件”到“等待”内部的“条件”的下面，就需要先选中任意列(不要选中“等”即可)，然后点击“条件”，就会将“条件”添加到“条件”的后面，在“等待”的内部。

步骤			新步骤	
等待				
	1 条件	条件	输入X00	接通
如果		条件	输入X01	接通
		条件	3	被添加到等待的里面
动作		动作	输出Y01	开启
2 条件		等	条件	输入X00
添加条件		条件	输入X01	断开
跳转		动作	输出Y00	关闭
		跳转		新步骤

上移，下移，删除：通过这些按钮可以将程序挪动位置，或者删掉。

2.6 编辑

名称：在“等待”的上面会有个首行，在此处可以给此“等待”起个名称，方便理解！

			新步骤			
			等待X00, X01两个开关同时按下			
	等	条件	输入X00	接通		#并且
		条件	输入X01	接通		
		动作	输出Y01	开启		
			等待X00, X01两个开关同时松开			
	等	条件	输入X00	断开		#并且
		条件	输入X01	断开		
		动作	输出Y00	关闭		
		跳转				新步骤


“如果”“是”“否”“步骤”也都可以首行起个名称。“步骤”的名称是必须的，而“等待”“如果”“是”“否”，首行的名称则可以空。

步骤程序的首列就是地址列，表示此行的地址。可以起个名称，从而对此行进行标识。然后就可以作为跳转的目标。

如果要将地址作为跳转目标的时候，那么需要先选中参数 2，然后按住 Ctrl 键再点击目标地址，才可以。

按住Ctrl			关闭Y00			
		动作	输出Y00	关闭		
2	循环检测	跳转	输入X00	接通		#或者
		跳转	输入X01	接通		开启Y00
		跳转				@循环检测 1
			开启Y00			
		动作	输出Y00	开启		

如果要将步骤作为跳转目标的时候，那么需要先选中参数 2，然后点击右边步骤列表中的步骤(不用按住 Ctrl 了)。

选取：当选中程序区<对象栏>，<参数 1>或<参数 2>后，如果鼠标指针变为手形  则表示处于选取模式。就可以选取左边的对象，或右边的步骤了。

隐藏：双击控制列即可将对应的控制块内容隐藏起来，相当于折叠起来。

			新步骤		
			X00 X01 同时接通		
	如	条件	输入X00	接通	#或者
		条件	输入X01	接通	
	是	开启			
		动作	输出Y00	开启	
	否	关闭			
		动作	输出Y00	关闭	
		跳转			新步骤

双击

双击后就可以将内容隐藏起来：

			任务1		
			新步骤		
	如	隐藏	X00 X01 同时接通		
	是	隐藏	开启		
	否	隐藏	关闭		
		跳转			新步骤

内容被隐藏了

除了“如果”“是”“否”“等待”的内容可以被隐藏外，“步骤”也有此功能！

2.7 对象

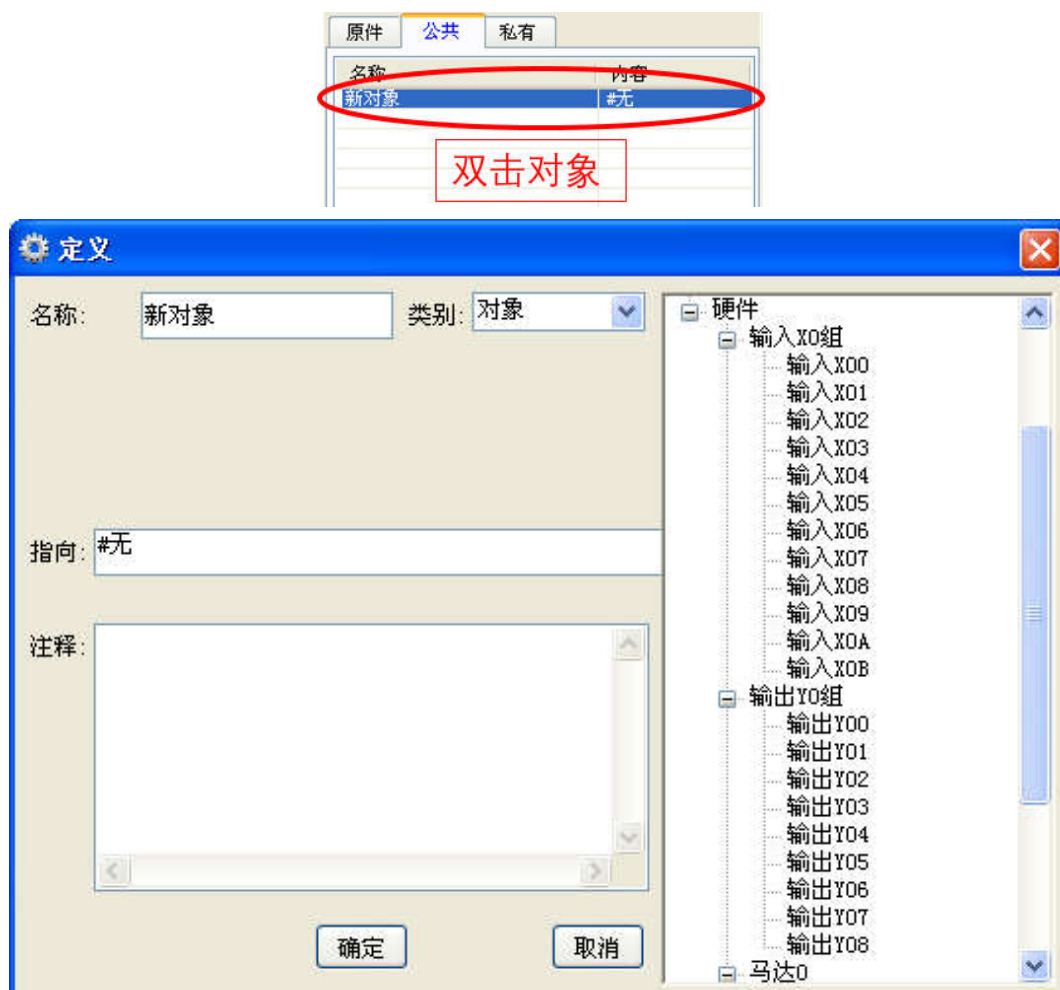
对象是程序每个命令的主体。程序的每个命令都需要有一个对象。全部都是通过选取而进入程序的。

不同区的对象允许重名，比如：公共区有个对象叫计时器，而私有区也可以有个对象叫计时器。而在程序的对象栏中，都是叫计时器。只有通过鼠标选取软件才知道指的是哪个区的‘计时器’。所以对象只能通过鼠标选取，否则软件不知道其指的是那个区的对象。

注意：对象只能是通过选取，不能通过打字上去，或直接打字来修改！

对象可以通过对象区底下的按钮来新建或删除。上移或下移。

当建立好一个对象后，双击对象，即可对对象进行定义：

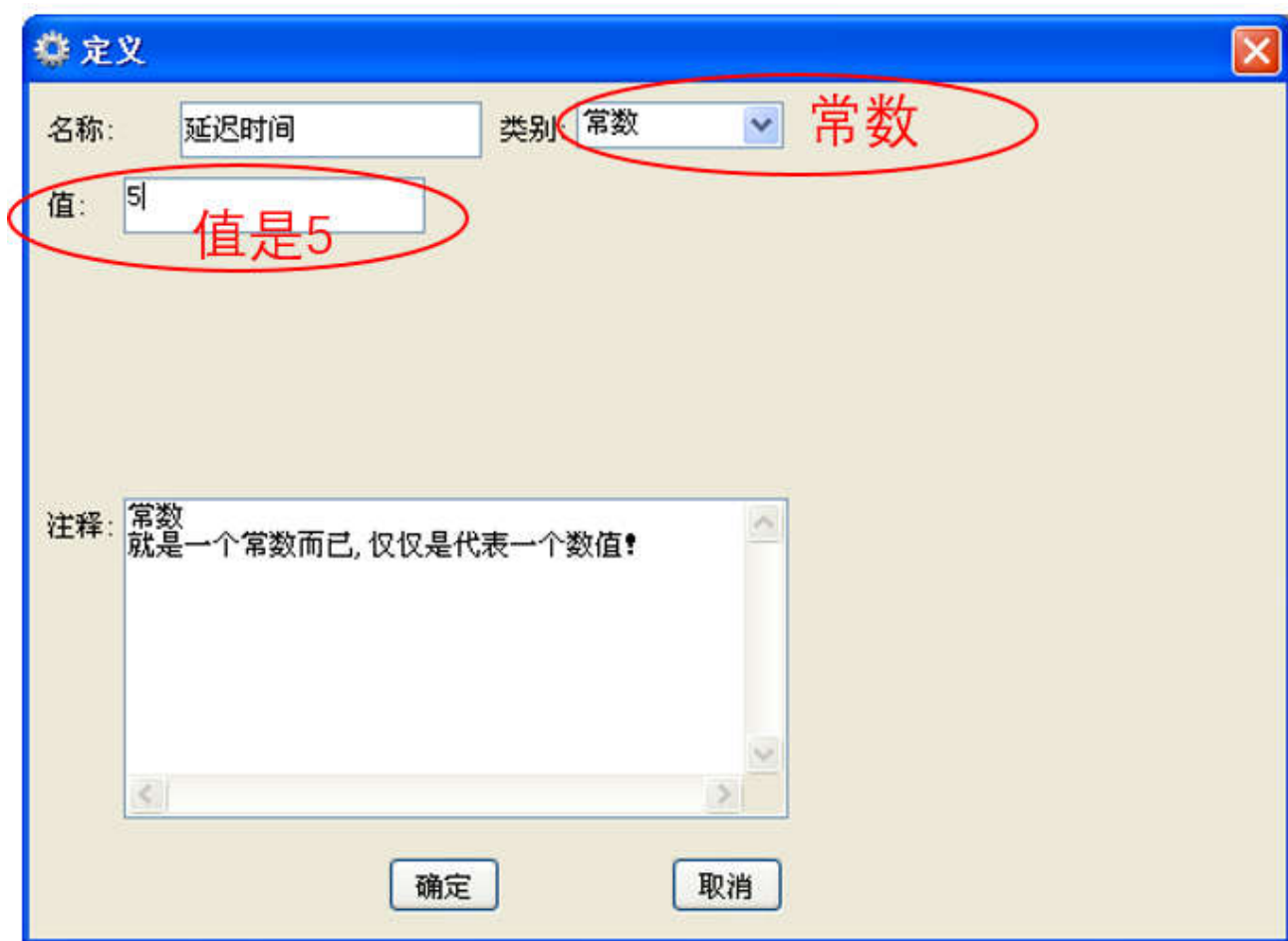


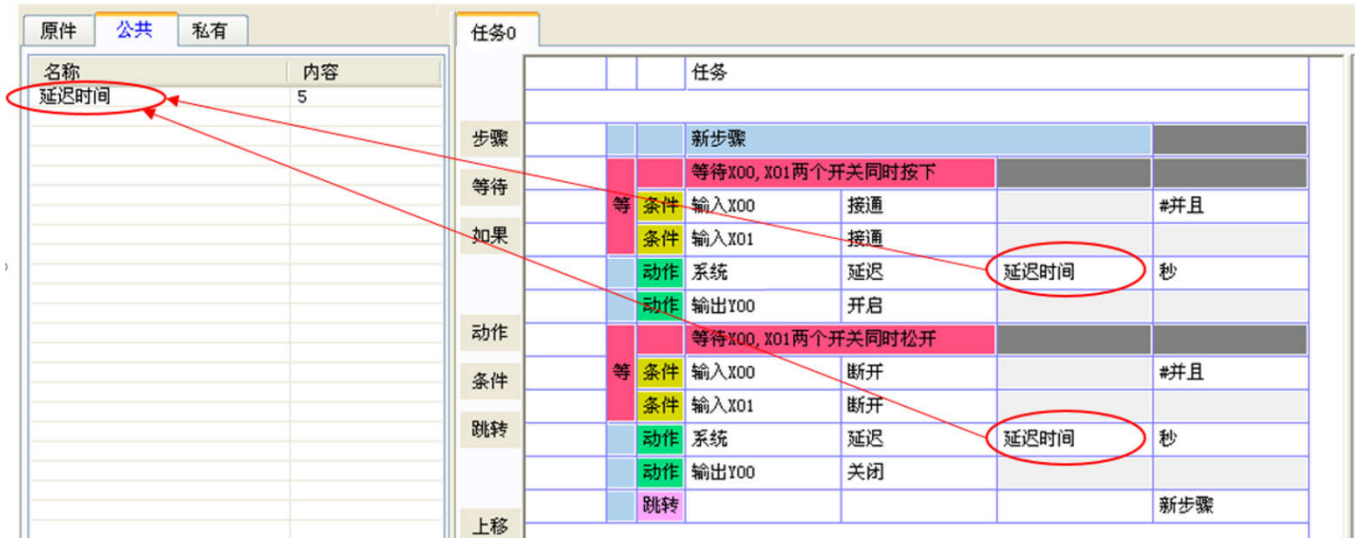
名称: 该对象的名称。可以起个通俗易懂的名字。但同一对象区不允许同名的对象。

类别: 类别中有两项,“对象”和“常数”,对象是具体的命令执行主体。而常数就是固定的数值,相当于C语言的“#define DELAY_TIME 5”。

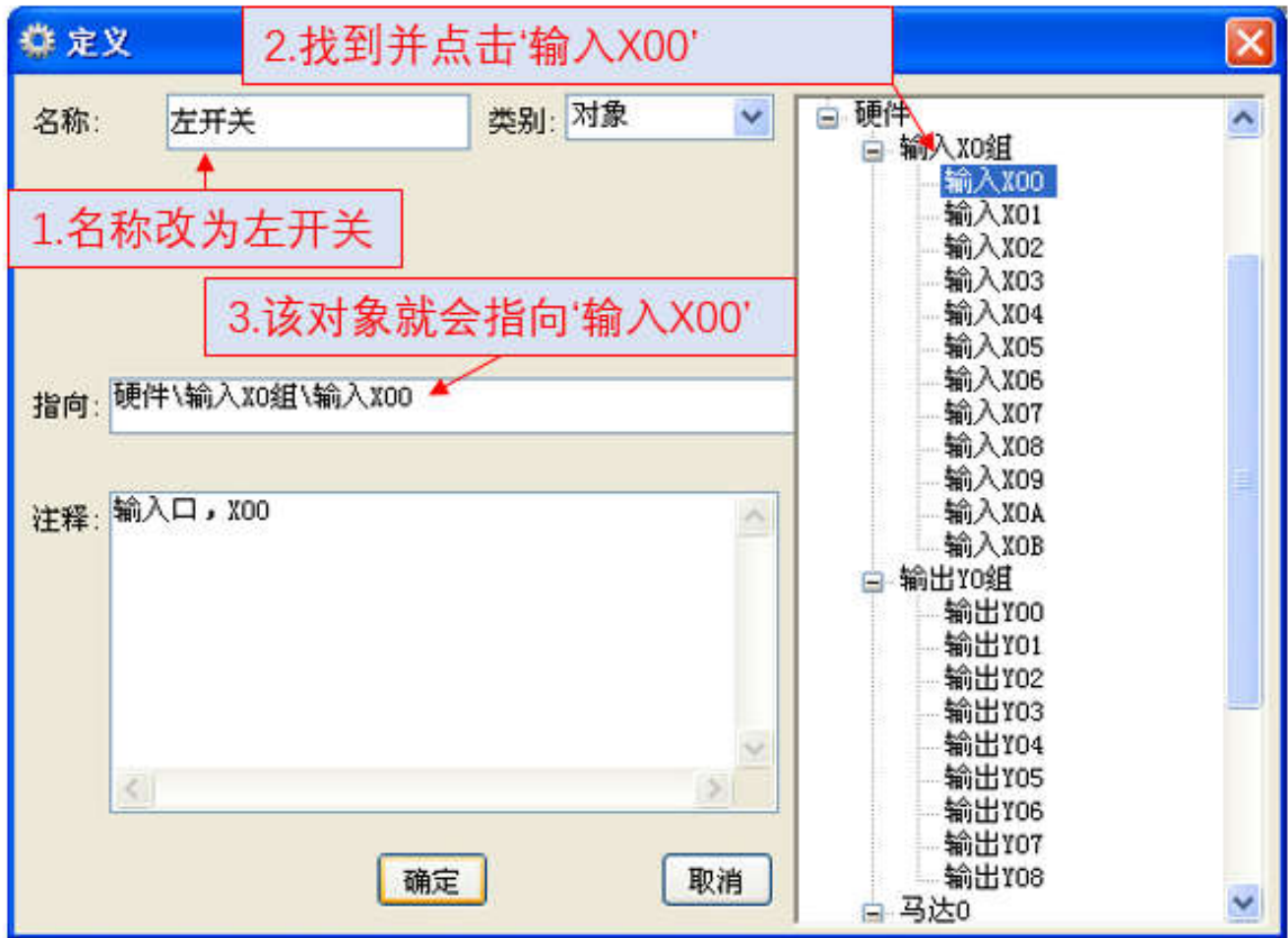
如果我们的程序在很多地方都需要延迟延迟一下,时间先确定为5秒。后来发现不够,需要改为8秒,那么就需要在多个地方都将5改为8。

那么如果我们定义一个常量(名字就为:延迟时间),其值是5。然后在需要延迟的地方延迟的参数栏都为‘延迟时间’。如果需要改为8的话只需要将其值由5改为8即可。就不用多个地方改了。



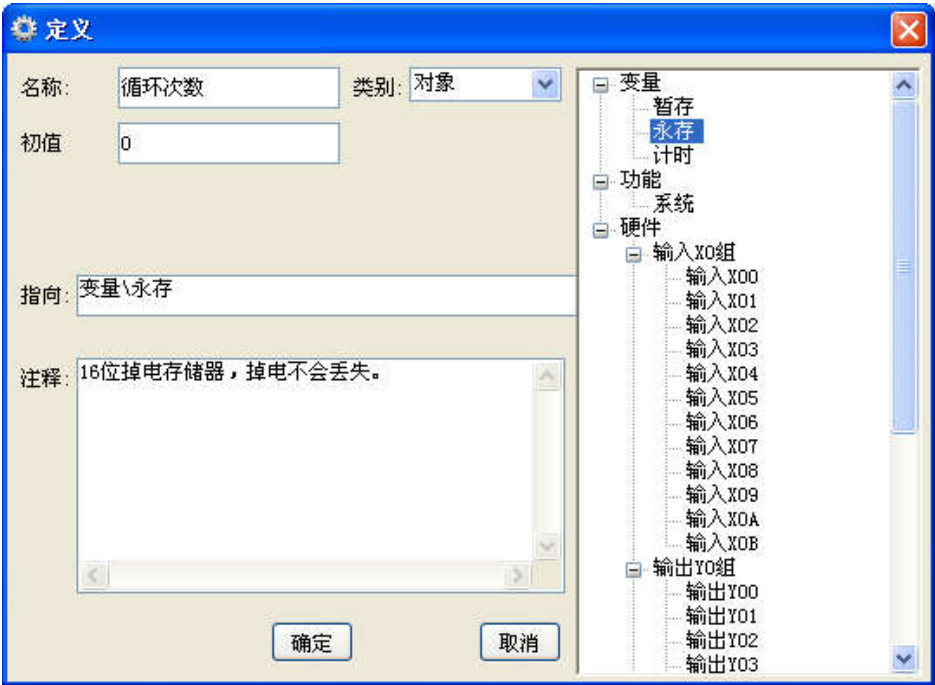


‘对象’的定义，是从右边的原件框中选取的。比如要定义一个对象，名字为‘左开关’，指的就是 X00：





也可以定义变量，有些变量是有初值的。比如‘永存’：



2.8 编译

点击工具栏中的<编译>按钮，就会立即编译。

如果没有错误就会在主说明框中显示编译信息！



如果有错误则会弹出信息框来提示编译有误，需要处理。

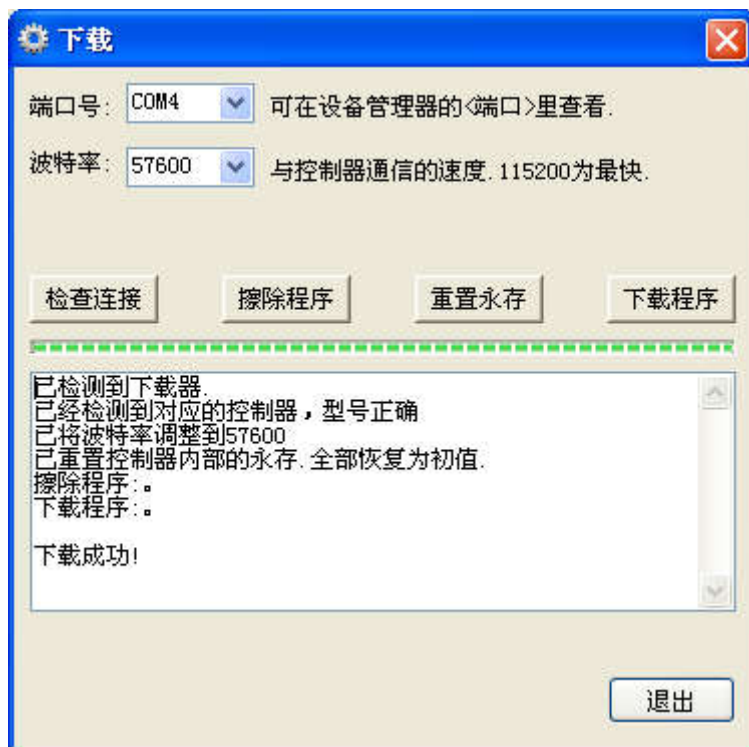


同时主说明框会显示编译结果，编译框会列出错误。双击编译框中的‘错误’
会立即定位到错误处，方便排除错误！



2.9 下载

点击工具栏中的<下载>按钮, 就会出现下载窗口, 如果连接正常的话就会下载。下载完后, 会自动关闭窗口(此功能需要在<选项>中设置)。下载失败, 若连接正常的话还可以点击<下载程序>来重试。

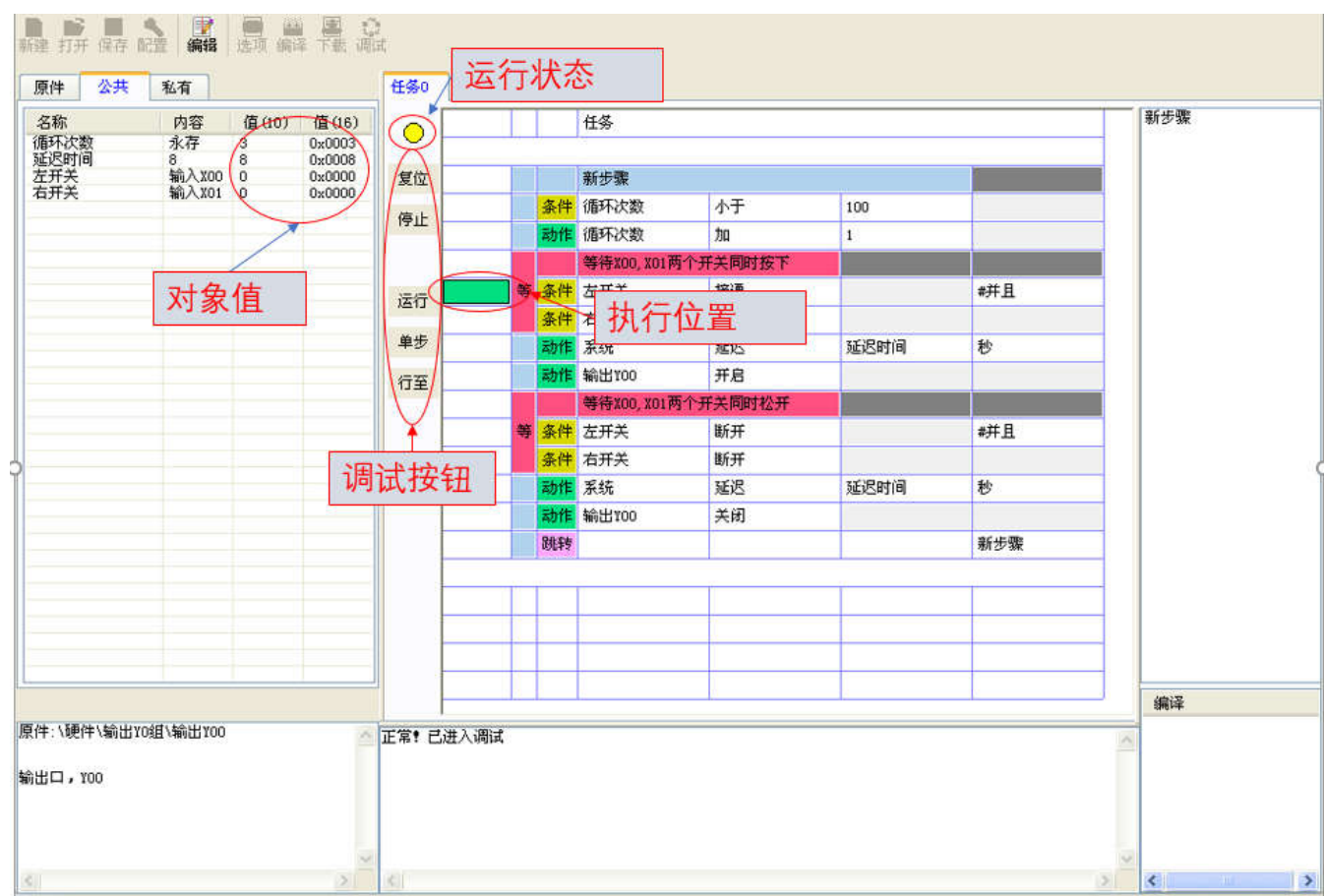


2.10 调试

点击工具栏中的<调试>按钮, 就会进入调试状态。进入调试状态需要满足以下条件:

1. 连接正常。
2. 本项目控制器型号与连接的控制型号一致。
3. 本项目程序与控制器里面的程序一致。

在调试模式下:




运行状态：绿色为正在运行，黄色为已经停止。


对象值：对象的值，分十进制和十六进制两栏。只有停止的时候才会显示。


执行位置：绿色表示执行位置，表示正准备执行此行。

调试按钮：

<复位>：相当于重启控制器，重启后立即停止。

<停止>：如果程序正在运行，按此按钮可以将其停止。相当于 

<运行>：让程序立即运行起来。相当于 

<单步>：让程序只走一个命令。相当于 

<行至>：让程序执行至某处。相当于 