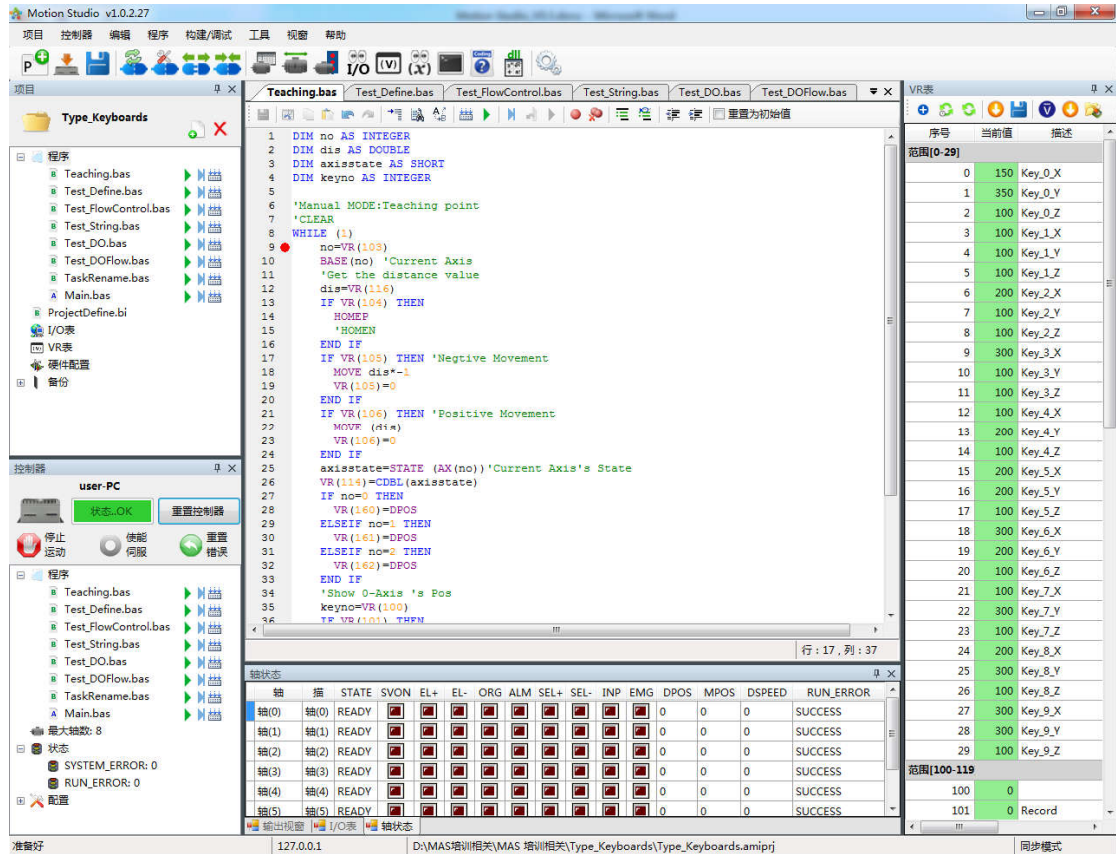


研华MAS控制器 Motion Studio手册

Version 1.3

1. 简要介绍

Motion Studio 是基于 Windows 平台下的一个应用程序,用于研华 MAS 控制器的程序开发和调试。Motion Studio 提供使用者最简单方式去设定控制器,快速开发并且能够在线监控(run-time diagnostics) 程序在控制器中的执行状态。



1.1 系统要求

运行 Motion Studio 软件的电脑配置需求如下

	最低	推荐
操作系统	Window XP SP3	Windows 7 Windows 8 Windows 10
.NET Library	.NET 4.0 及以上	
处理器		
RAM	2M	4MBytes
硬盘空间	150M	300M

1.2 操作模式

Motion Studio 编程环境有 3 种操作模式

- 断线模式 (Disconnected Mode)

Motion Studio 跟控制器未联机,所有的功能都无法使用,通讯联机为关闭状态。

- 监控模式 (Tool Mode)


选择监控模式联机，Motion Studio 可使用所有的 Tools 来监控控制器的状态与所有轴与 I/O 的状态，也可以编译/运行/停止程序的运行，但不可以在编辑器中编辑程序。

- 同步模式 (Sync Mode)

在此模式用户可以做任何的操作，包含编写程序,下载程序,执行程序 and 用测试工具测试每个轴，每个 IO 点等。

2. 开发步骤

2.1 与控制器建立连接

初次打开 Studio 后，若 Studio 和控制器不是同一台电脑，两者没有建立连接，需要先与控制器建立连接。点击主菜单的控制器—>连接设置或者点击工具按钮的, 将弹出以下对话框：



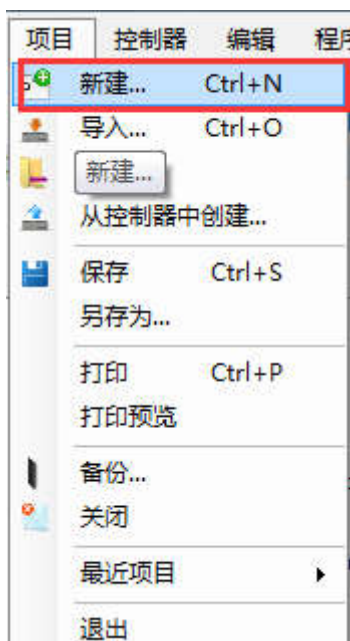
当选择接口为以太网时，输入控制器的 IP 地址和 IP 端口，点击“应用”，与控制器建立连接。此时，表示使用实际轴卡。

当选择接口为模拟器时，可选择与 MAS-324X Simulator（4 轴虚拟控制器）或者 MAS-328X Simulator（8 轴虚拟控制器）进行连接。

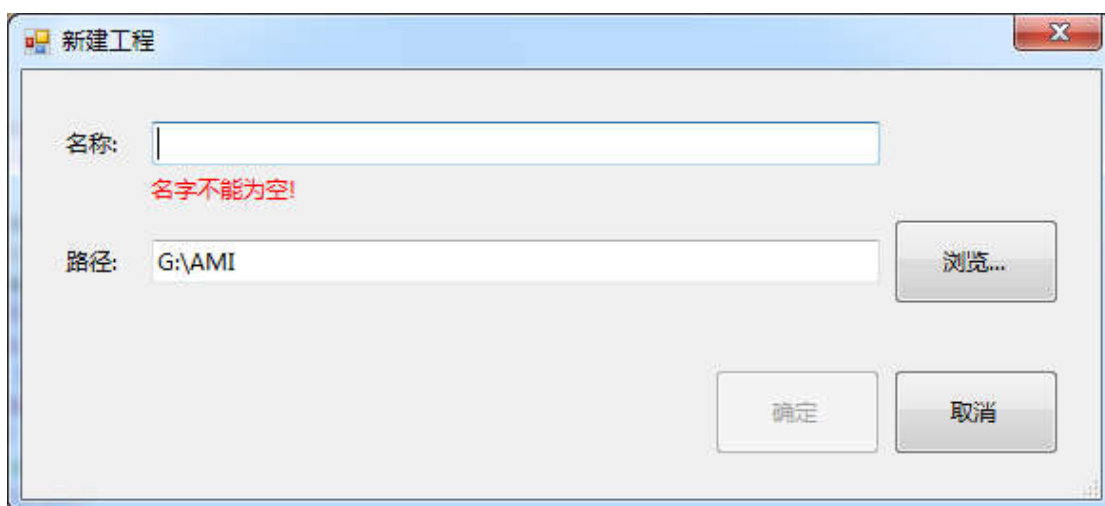


2.2 新建项目

点击主菜单——>项目——>新建...，创建新的项目



选择工程要保存的路径后，输入工程名（工程名只能是字母/数字/下划线），点击“确定”或者按 Enter 键将创建一个新的工程。

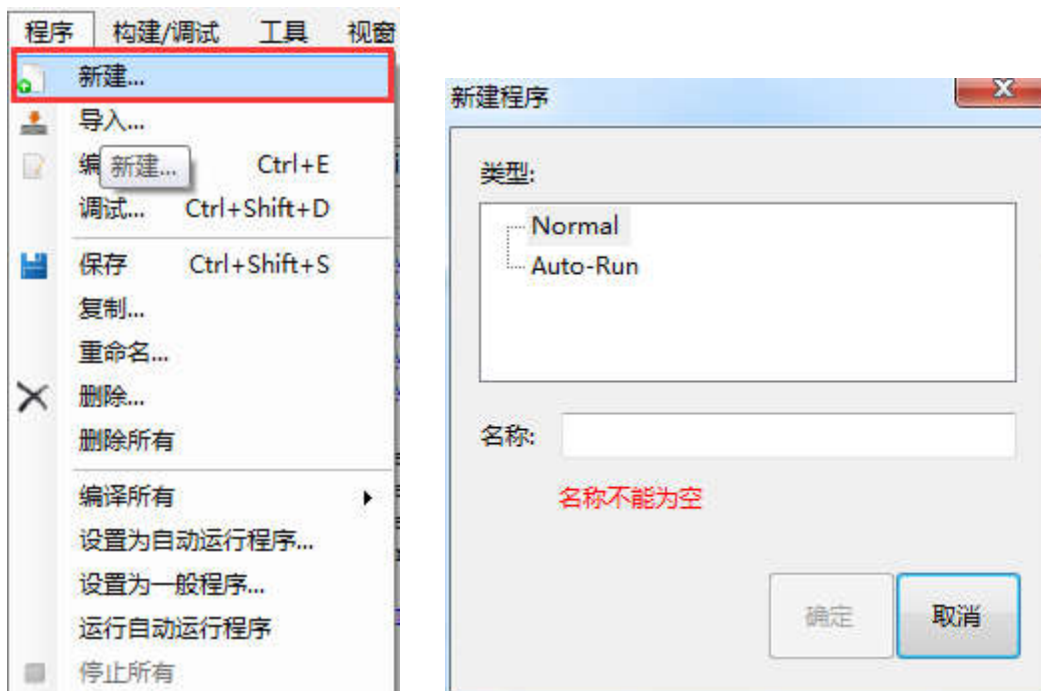


注：创建项目会同时创建一个与项目名相同的文件夹，文件夹的名称一定要与项目工程的名称一致，否则 Controller 将不识别。因此，请切记在 Studio 以外私自更改文件（文件夹）的名称/内容！

也可通过项目—>导入...，切换到另外一个已存在的项目。

2.3 新建程序

点击主菜单——>程序——>新建...，创建新的程序。在跳出的窗体中选择程序类型（默认为 Normal），点击“确定”或者按 Enter 键将创建新的程序。

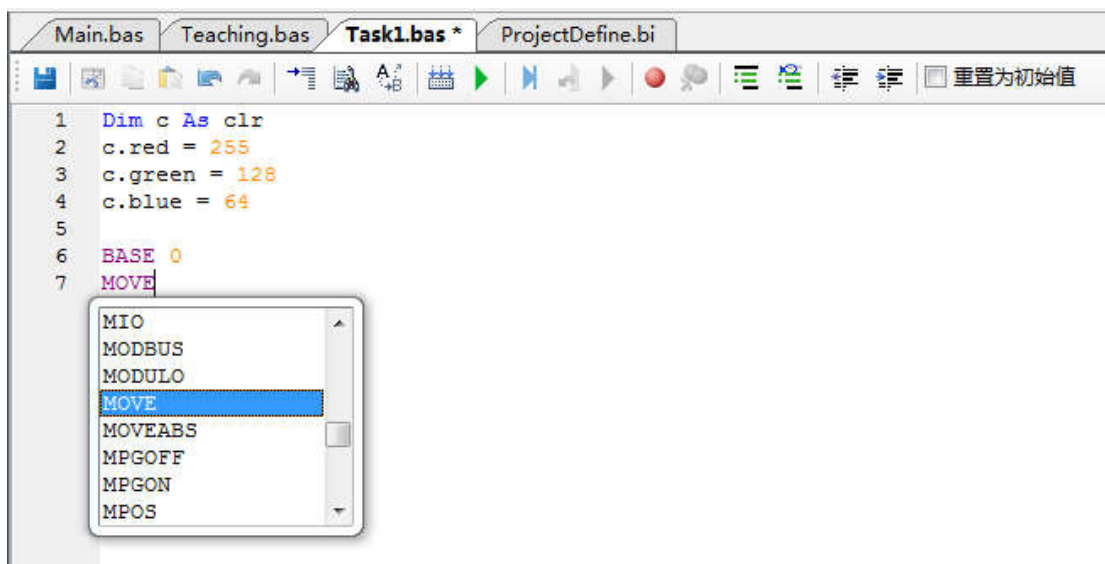


也可通过程序—>导入...，导入其他项目中已存在的程序

2.4 编写 BASIC 代码

新建程序后即可在编辑框中编辑 BASIC 指令。可使用上面的工具按钮进行相应编辑，如后面四个按钮依次含义为：注释、取消注释、减少缩进和增加缩进。


勾选“重置为初始值”，则在此.bas 运行前会将 VR 表的 VR 变量和 IO 表的 IO 变量重新设置为对应的初始值。







2.5 编译/调试/运行

编写完 BASIC 后，可点击各个编辑框的工具按钮进行编译/调试/运行。

主菜单和编辑框中编译/调试/运行相关工具按钮的含义分别为：

：编译

：发布版运行

- ：调试版运行
- ：单步
- ：停止运行

2.6工具

- 运动测试：轴测试

轴测试工具：可进行 PTP/JOG/Homing 运动测试



- 查看轴属性：轴参数

可查看配置后的指定轴的属性。

轴参数

选择轴

勾选	参数	轴 (0)	轴 (1)
<input checked="" type="checkbox"/>	UNIT_NUM	1	1
<input checked="" type="checkbox"/>	UNIT_DENOM	1	1
<input type="checkbox"/>	MODULO	0	0
Speed			
<input checked="" type="checkbox"/>	MAXVEL	5000000	5000000
<input checked="" type="checkbox"/>	MAXACC	5E+07	5E+07
<input checked="" type="checkbox"/>	MAXDEC	5E+07	5E+07
<input checked="" type="checkbox"/>	VH	8000	8000
<input checked="" type="checkbox"/>	VL	2000	2000
<input checked="" type="checkbox"/>	ACC	10000	10000
<input checked="" type="checkbox"/>	DEC	10000	10000
<input checked="" type="checkbox"/>	JK	0	0
<input checked="" type="checkbox"/>	INSTOP_DEC	100000	100000
Alarm			

- 查看轴的状态：轴状态

轴状态

轴	描述	STATE	SVON	EL+	EL-	ORG	ALM	SEL+	SEL-	INF	EMG	DPOS	MPOS	DSPEED	RUN_ERROR
轴 (0)	轴 (0)	READY										100	0	0	SUCCESS
轴 (1)	轴 (1)	READY										0	0	0	SUCCESS
轴 (2)	轴 (2)	READY										0	0	0	SUCCESS
轴 (3)	轴 (3)	READY										0	0	0	SUCCESS
轴 (4)	轴 (4)	READY										0	0	0	SUCCESS
轴 (5)	轴 (5)	READY										0	0	0	SUCCESS
轴 (6)	轴 (6)	READY										0	0	0	SUCCESS
轴 (7)	轴 (7)	READY										0	0	0	SUCCESS

- 查看 DI/DO 状态：数字输入输出查看器
可查看指定 DI/DO 的状态，可点击切换 DO 的状态。

数字输入输出查看器	
<div></div>	
DIN	描述
0	PCI-1245 Axis-0 IN1/LTC
1	PCI-1245 Axis-0 IN2/RDY
2	PCI-1245 Axis-0 IN4/TOG+
3	PCI-1245 Axis-0 IN5/TOG-
4	PCI-1245 Axis-1 IN1/LTC
5	PCI-1245 Axis-1 IN2/RDY
6	PCI-1245 Axis-1 IN4/TOG+
7	PCI-1245 Axis-1 IN5/TOG-
DOU	描述
8	PCI-1245 Axis-2 OUT4/C...
9	PCI-1245 Axis-2 OUT5/CMP
10	PCI-1245 Axis-2 OUT6/SVON
11	PCI-1245 Axis-2 OUT7/ERC
12	PCI-1245 Axis-3 OUT4/C...
13	PCI-1245 Axis-3 OUT5/CMP
14	PCI-1245 Axis-3 OUT6/SVON
15	PCI-1245 Axis-3 OUT7/ERC
DOU	描述
0	PCI-1245 Axis-0 OUT4/C...
1	PCI-1245 Axis-0 OUT5/CMP
2	PCI-1245 Axis-0 OUT6/SVON
3	PCI-1245 Axis-0 OUT7/ERC
4	PCI-1245 Axis-1 OUT4/C...
5	PCI-1245 Axis-1 OUT5/CMP
6	PCI-1245 Axis-1 OUT6/SVON
7	PCI-1245 Axis-1 OUT7/ERC
DOU	描述
16	VMCard PCI-1245 Axis-0...
17	VMCard PCI-1245 Axis-0...
18	VMCard PCI-1245 Axis-0...
19	VMCard PCI-1245 Axis-0...
20	VMCard PCI-1245 Axis-1...
21	VMCard PCI-1245 Axis-1...
22	VMCard PCI-1245 Axis-1...
23	VMCard PCI-1245 Axis-1...

- 查看/设置 VR 值：VR 表
可查看/设置 VR 的值，并可为每个自定义变量设置初始值。若使用在 Modbus 中，可设置对应 Modbus 的起始位置以及 Modbus 中的数据类型。

VR表			
序号	当前值	描述	初始值
范围[0-59]			
0	-214...	SCARA_X Zer...	-21474100
1	-214...	SCARA_Y Zero...	-21473700
2	-214...	SCARA_Z Zero...	-21474500
3	-101...	SCARA_R Zer...	-10130900
4	27000	ShellPickUpPos	27000
5	3500	ShellTempPos	3500
6	1000	ShellPutDown...	1000
7	0	AboveShellGetZ	0
8	92500	UnderShellGetZ	92500
9	6400...	ShellSetX	640000
10	-74000	ShellSetY	-74000
11	33000	ShellSetR	33000
12	95000	UnderShellSetZ	95000
13	48000	AboveShellSetZ	48000
14	0	ConveyorBSp...	0
15	30000	ShellGetX	30000
16	-690...	ShellGetY	-690000
17	0		0
18	0		0
19	0		0
20	0		0
21	0		0

- 查看变量值：观察变量
选中变量后，可通过快捷键 Ctrl+W 或者拖拉到观察变量中添加变量。
程序运行时，可（实时）查看变量（全局变量/局部变量/Motion 参数）值。

观察变量		
名称	值	上下文
aa	Invalid Value	Task1.bas
VR(0)	-2147410000	VR
DPOS AX(0)	0	AXIS(0)
STATE AX(0)	1	AXIS(0)

- 终端下指令：终端
可通过此工具下单行指令，进行相关测试/设定。
可直接输入，也可事先设定宏按钮的命令，点击宏按钮会直接下对应的命令。



3 主菜单

3.1 项目

提供项目操作相关功能，如新建工程（New），打开工程（Open）等，具体如下：

菜单名称	操作说明
新建	创建一个新的工程并清除控制器的内容
导入	导入一个已存在的项目到控制器
更改	切换到另一个不同的项目，并与控制器的内容进行比较
从控制器中创建	根据控制器中的内容创建一个新的项目
保存	保存当前项目信息
另存为	将当前项目另存为另一个工程
打印	打印当前活动编辑程序
打印预览	预览当前活动编辑程序
备份	打开 "备份管理 " 工具进行创建备份或管理备份的操作
关闭	关闭当前项目（此操作后将进入监控模式）
最近项目	列出最近操作的项目，可通过选择相应项目进行项目切换
退出	退出应用程序

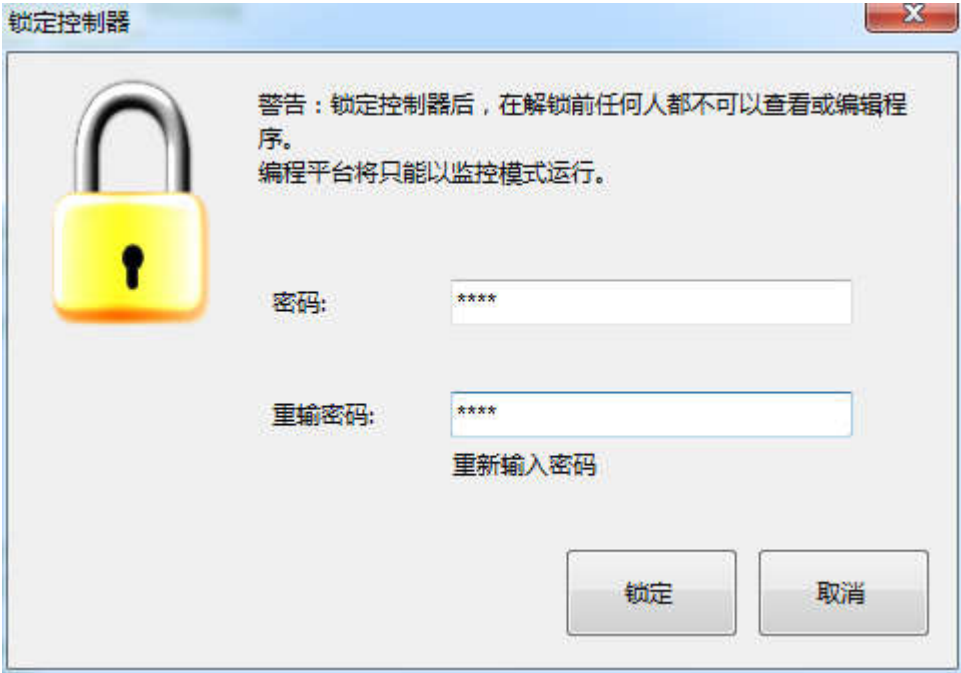
3.2 控制器

提供控制器操作相关功能，如操作模式变换，与控制器通信连接配置等，具体如下：

菜单名称	操作说明
同步模式连接	与控制器进行同步模式连接
监控模式连接	与控制器进行监控模式连接
断开连接	与控制器断开连接
连接设置	更改与控制器通信的连接设置
重置控制器	重置控制器以进行热启动
锁定控制器	输入密码锁定控制器
解锁控制器	输入密码解锁控制器
目录	列表显示控制器上的程序相关信息
日期时间	设置/查看控制器上日期和时间

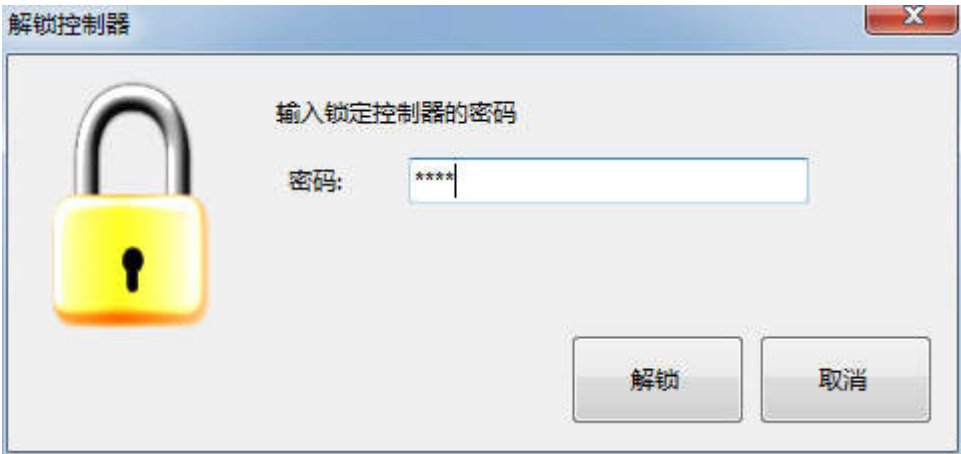
- 锁定控制器

可通过设定密码锁定控制器。控制器被锁定后，Studio 会进入监控模式。



● 解锁控制器

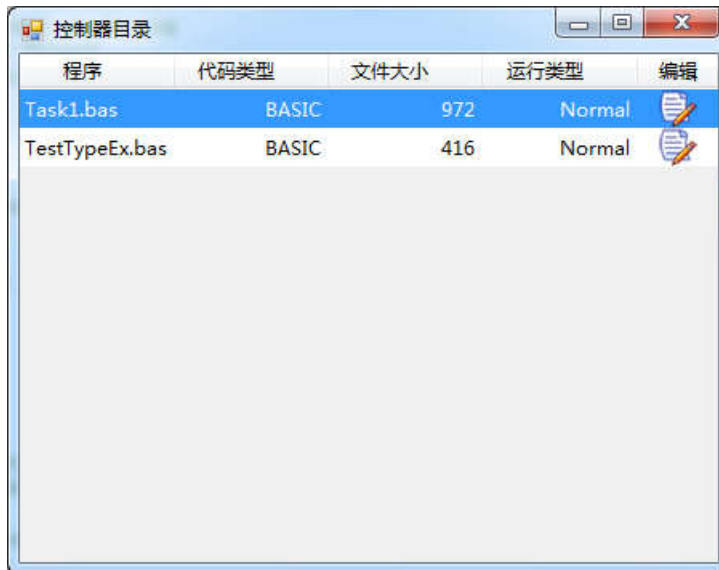
锁定控制器后，点击解锁控制器，设定密码，点击“解锁”后，控制器解除锁定。解锁控制器后会自动切换回同步模式。



● 目录

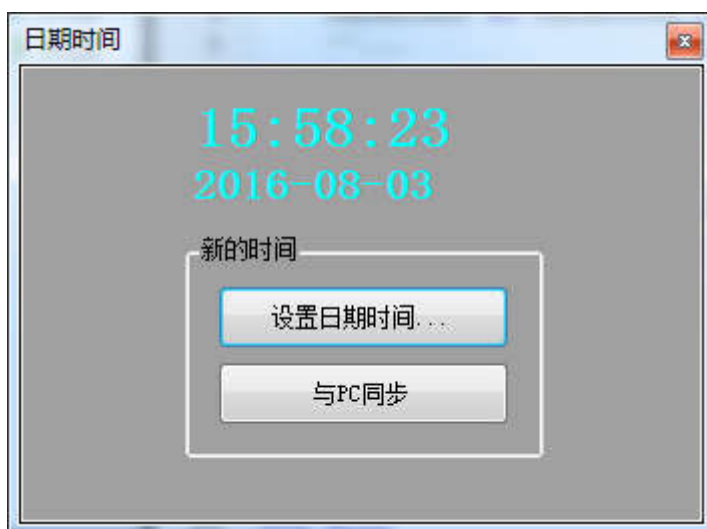
列表显示控制器上的程序相关信息。

列名	含义
程序	程序的名称
代码类型	目前均为 BASIC
文件大小	此程序的大小，单位（字节）
运行类型	即程序的类型：Normal（一般程序）或 Autorun（自动运行程序）
编辑	是否可编辑，目前均为可编辑图标

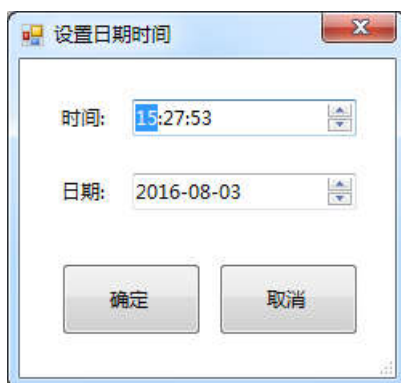


- 日期和时间

设置/查看控制器上日期和时间



点击“设置日期时间”，会弹出如下界面：



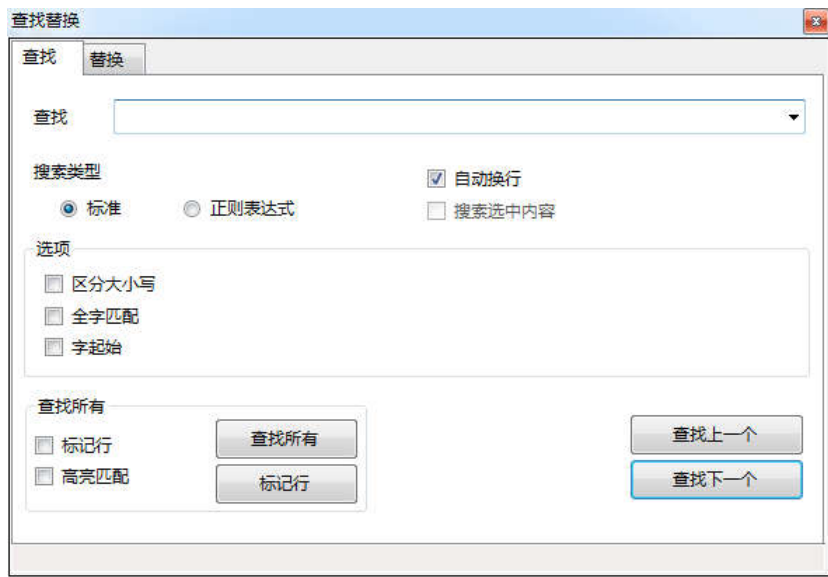
设置日期和时间后，点击“确定”，将按照所设定值重新设置控制器的日期和时间。
点击“与 PC 同步”，会设置控制器的日期和时间与 Studio 所在电脑一致。

3.3.编辑

提供对编辑框操作相关功能，如剪切、复制、粘贴等，具体如下：

菜单名称	操作说明
撤销	撤销最后一次的编辑操作
重做	重新操作最后一次取消的编辑操作
剪切	将当前选定的文本剪切到剪贴板
复制	将当前选定的文本复制到剪贴板中
粘贴	从剪贴板中粘贴文本
选择所有	选择文档中的所有文本
选择无	取消当前的选择
查找和替换	使用“查找和替换”查询/替换指定文字
转到	跳转到指定行
行号	显示/取消显示行号
书签	书签的操作包含以下： 切换书签：在当前行添加/移除书签 上一个书签：跳转到上一个书签 下一个书签：跳转到下一个书签 清除书签：清除所有书签
高级	高级操作包含以下： 注释取消注释 增加/缩小行缩进 向前/向后定位 大写/小写选中内容

● 查找和替换



3.4 程序

提供对程序相关操作功能，如添加新的/已存在的程序到当前项目，具体如下：

菜单名称	操作说明
新建	新建一个空的程序
导入	导入一个已存在的程序到当前项目中
编辑	编辑当前项目中的一个/多个程序
调试	调试当前项目中的一个/多个程序
保存	保存当前程序
复制	复制当前项目中一个程序
重命名	重命名当前项目中的一个程序
删除	删除选中的当前项目中的程序
删除所有	删除当前项目中的所有程序
编译所有	编译当前项目中的所有程序，可选择调试版或发布版编译
设置为自动运行程序	设置选中的当前项目中的程序为自动运行程序
设置为一般程序	设置选中的当前项目中的程序为一般程序
运行自动运行程序	运行所有设置为自动运行的程序
停止所有	停止所有正常运行的程序

3.5 构建/调试

提供对 BASIC 程序进行编译、调试相关功能操作。具体如下：

菜单名称	操作说明
编译	编译选中的程序（如有改动，会先保存）
运行（发布版）	发布版运行/停止当前活动的程序
运行（调试版）	调试版运行/停止当前活动的程序
单步（不进入子程序）	单步调试（不进入子程序）
继续运行	继续运行当前活动的程序
切换断点	在当前行上设置/移除断点
删除断点	删除所有断点
查看变量	将当前选中的变量加入到变量监测工具中

3.6 工具

提供相关工具以便于调试或查看控制器运行的状态。具体如下：

菜单名称	操作说明
------	------

轴属性	通过“轴属性”工具查看选定轴的属性
轴测试	通过“轴测试”工具进行轴运动测试
轴状态	通过“轴测试”工具查看所有轴的状态
数字输入输出表	通过“数字输入输出表”工具查看选定的数字输入/输出的状态
VR 表	通过“VR 表”工具查看/设定自定义变量
变量监测	通过“变量监测”工具查看变量值
终端	通过“终端”工具可以与控制器进行 BASIC 指令交互
编程助手	通过“编程助手”可查看 BASIC 指令的简单说明、用法和范例
自定义库函数	通过“自定义库函数”工具可以在 BASIC 中使用第三方 dll (C/C++) 的 API
选项	通过“选项”可对 Studio 的一些操作进行设置

具体请参看 9 工具的各小节。

3.7 视窗

显示或隐藏相关界面/窗体，具体如下：

菜单名称	操作说明
工具栏	显示/隐藏工具栏
状态栏	显示/隐藏状态栏
控制器	显示/隐藏控制器窗体
项目	显示/隐藏项目窗体
输出视窗	显示/隐藏输出视窗
重置窗体布局	重置为默认的窗体布局

3.8 帮助

提供帮助文档链接相关功能，具体如下：

菜单名称	操作说明
Motion Studio 帮助	显示 Motion Studio 手册
Motion BASIC 帮助	显示 Motion BASIC 手册
关于 Motion Studio	对 Motion Studio 的版本/版权进行简要说明

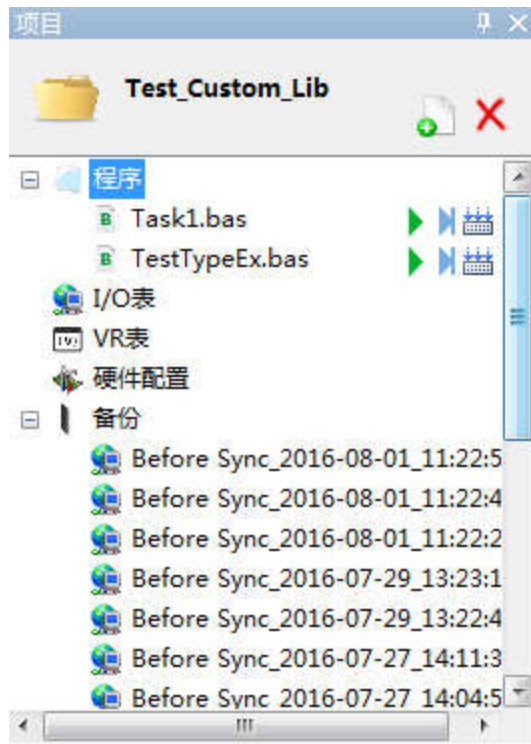
4 工具按钮



工具按钮提供快捷使用常用功能/工具的方式。

图标	功能	说明
	新建工程	创建一个新的工程并清除控制器的内容
	导入工程	导入一个已存在的项目到控制器
	保存工程	保存当前工程的所有信息
	同步模式连接	与控制器进行同步模式连接
	监控模式连接	与控制器进行监控模式连接
	断开连接	与控制器断开连接
	连接设置	更改与控制器通信的连接设置
	轴属性	显示/隐藏“轴属性”工具
	轴测试	显示/隐藏“轴测试”工具
	轴状态	显示/隐藏“轴测试”工具
	数字输入输出表	显示/隐藏“数字输入输出表”工具
	VR 表	显示/隐藏“VR 表”工具
	变量监测	显示/隐藏“变量监测”工具
	终端	显示/隐藏“终端”工具
	编程助手	显示/隐藏“编程助手”
	自定义库函数	打开“自定义库函数”工具
	选项	打开“选项”对话框

5 项目



5.1 程序

程序节点下列出目前项目所包含的所有程序。

单击每个程序名，同步模式下，编辑器将切换到对应的程序。可点击图标对相应程序进行编译/发布版运行/调试版运行的操作。

单击程序，右键，会出现如下菜单：



可在项目中新建/导入程序。或者对当前已存在的程序进行编译/停止或删除的操作。

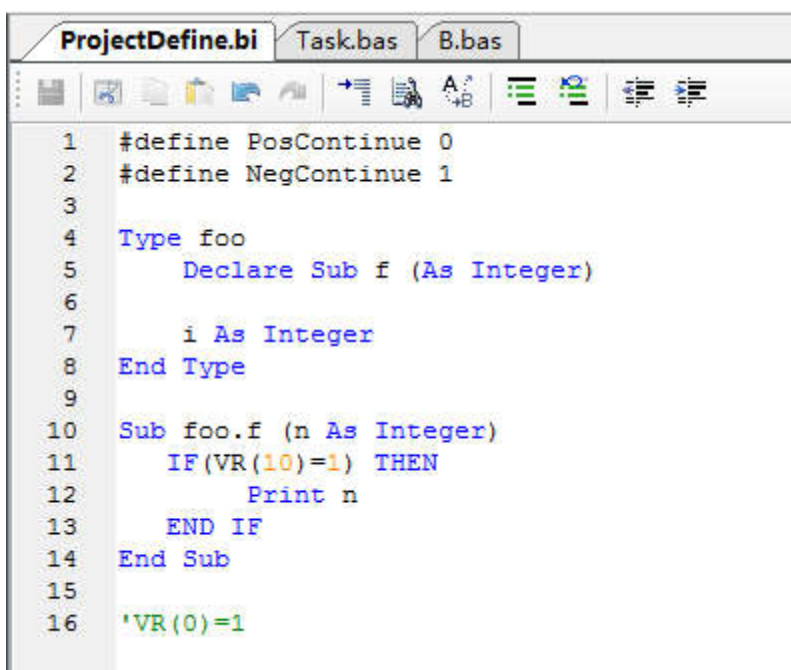
选中某个具体程序，右键，会出现如下菜单：



可对选中的项目进行编译/运行/单步/继续运行以及重命名/复制/删除/设置为自动运行程序（设置为一般程序）等编辑操作。

5.2 ProjectDefine.bi

点击此节点，会在编辑框中显示 ProjectDefine.bi。



添加 bi 编辑器的功能有以下几种：

1. 进行宏定义

如在 ProjectDefine.bi 中添加以下宏定义：

```
#DEFINE X_Pos 0
#DEFINE Y_Pos 1
#DEFINE Z_Pos 2
```

在.bas 中使用：

```
VR(X_Pos)=100
VR(Y_Pos)=200
VR(Z_Pos)=300
```

2. 自定义数据类型，如枚举类型/结构体...

如在 ProjectDefine.bi 添加如下定义：

```
Type clr
    red As UByte
    green As UByte
    blue As UByte
```

End Type

在.bas 中使用：

```
Dim c As clr
c.red = 255
c.green = 128
c.blue = 64
```

注：上述定义或声明可使用在所有 Task 中。

5.3I/O 表

创建硬件配置后，将自动生成对应的 I/O 表：

名称	当前值	描述	初始值	Modbus	数据类型
Motion_PCI-1245_12					
OUT (0)	0	PCI-1245 Axi...	0	1	BOOL
OUT (1)	0	PCI-1245 Axi...	0	2	BOOL
OUT (2)	0	PCI-1245 Axi...	0	3	BOOL
OUT (3)	0	PCI-1245 Axi...	0	4	BOOL
OUT (4)	0	PCI-1245 Axi...	0	5	BOOL
OUT (5)	0	PCI-1245 Axi...	0	6	BOOL
OUT (6)	0	PCI-1245 Axi...	0	7	BOOL
OUT (7)	0	PCI-1245 Axi...	0	8	BOOL
OUT (8)	0	PCI-1245 Axi...	0	9	BOOL
OUT (9)	0	PCI-1245 Axi...	0	10	BOOL
OUT (10)	0	PCI-1245 Axi...	0	11	BOOL
OUT (11)	0	PCI-1245 Axi...	0	12	BOOL
OUT (12)	0	PCI-1245 Axi...	0	13	BOOL
OUT (13)	0	PCI-1245 Axi...	0	14	BOOL
OUT (14)	0	PCI-1245 Axi...	0	15	BOOL
OUT (15)	0	PCI-1245 Axi...	0	16	BOOL

其中，列名与含义如下：

名称：I/O 变量名，在程序中可直接使用

当前值：对应变量的当前值：0/1

描述：对应变量的描述

初始值：对应变量的初始值，默认为 0。用户可根据实际情况设置初始值，在程序运行前，可通过勾选“重置为初始值”将变量重新设置为初始值。

Modbus：对应变量的 Modbus 地址。

数据类型：目前只支持数字输入/输出，因为数据类型都是 BOOL 类型。

工具按钮功能如下：



图标	功能	说明
	定时创新	使能/禁用定时创新 I/O 表的当前值
	刷新	点击后将刷新一次 I/O 表的当前值
	重置为初始值	点击后将会把 DO 变量的初始值设置为当前值
	下载到控制器	点击后将会把 I/O 表下载到控制器，控制器端将会对 I/O 进行 Modbus 地址等匹配

5.4VR 表

VR 表即用户自定义的全局变量表。
VR 变量的格式为 VR(x)，x 的值为 0-9999，即最大支持 10000 个。
VR 的数据类型默认为 Double 型，若 VR 是需要与 WOP 等进行关键，则需要自行根据实际情况配置 VR 的数据类型与 Modbus 地址（注：此两项必须与 WOP 上配置一致，否则可能出现 Modbus 通信不正常或者值显示不正常的情况。）
VR 表如下：

VR表						
序号	当前值	描述	初始值	Modbus	数据类型	
范围[0-59]						
0	-214...	SCARA_X Zero...	-2147410000	40001	BIT_32_FLOAT	▼
1	-214...	SCARA_Y Zero...	-2147370000	40003	BIT_32_FLOAT	▼
2	-214...	SCARA_Z Zero...	-2147450000	40005	BIT_32_FLOAT	▼
3	-101...	SCARA_R Zero...	-10130900	40007	BIT_32_FLOAT	▼
4	27000	ShellPickUpPos	27000	40009	BIT_32_FLOAT	▼
5	3500	ShellTempPos	3500	40011	BIT_32_FLOAT	▼
6	1000	ShellPutDownPos	1000	40013	BIT_32_FLOAT	▼
7	0	AboveShellGetZ	0	40015	BIT_32_FLOAT	▼
8	92500	UnderShellGetZ	92500	40017	BIT_32_FLOAT	▼
9	640000	ShellSetX	640000	40019	BIT_32_FLOAT	▼
10	-74000	ShellSetY	-74000	40021	BIT_32_FLOAT	▼
11	33000	ShellSetR	33000	40023	BIT_32_FLOAT	▼
12	95000	UnderShellSetZ	95000	40025	BIT_32_FLOAT	▼

其中，列名与含义如下：
名称：变量名，在程序中可直接使用
当前值：对应变量的当前值

描述：对应变量的描述

初始值：对应变量的初始值，默认为 0。用户可根据实际情况设置初始值，在程序运行前，可通过勾选“重置为初始值”将变量重新设置为初始值。

Modbus：对应变量的 Modbus 地址。

数据类型：可在创建 Range 时，选择数据类型或者单个配置（注：程序未运行时）

目前数据类型只支持

BIT_16_UINT

BIT_32_FLOAT

工具按钮功能如下：



图标	功能	说明
	新建范围	点击后将跳出子对话框，进行配置后创建新的 VR 范围
	定时创新	使能/禁用定时创新 I/O 表的当前值
	刷新	点击后将刷新一次 I/O 表的当前值
	从.csv 文件中导入	从.csv 文件中导入已保存的 VR 相关信息显示到 VR 表中
	保存为.csv 文件	将 VR 表保存为.csv 文件
	重置为初始值	点击后将会把 DO 变量的初始值设置为当前值
	下载到控制器	点击后将会把 I/O 表下载到控制器，控制器端将会对 I/O 进行 Modbus 地址等匹配

新建范围的对话框如下：

添加范围

从

到

0

Modbus起始地址:

0

VR数据类型:

确定

取消

5.5硬件配置

点击“配置”，可查看当前的硬件配置或者根据实际硬件情况配置项目所用板卡（运动控制卡+I/O卡）。

配置分为三块：输入配置，输出配置和运动配置。

输入配置包含数字输入配置和模拟输入配置（暂时保留）。目前主要支持

DAQ卡：PCI-1750（16DI）和 PCI-1756（32DI）

Motion卡：MVP-3245-MAS（16DI）

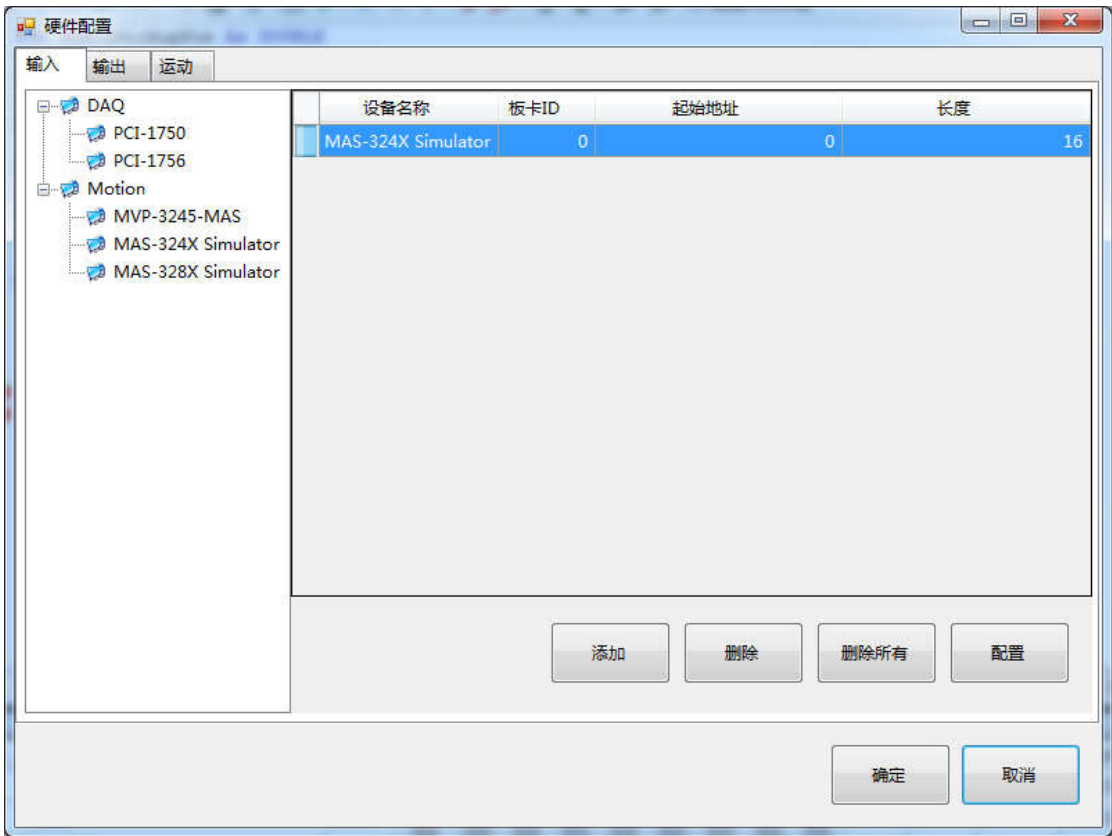
虚拟控制器：MAS-324X Simulator 和 MAS-328X Simulator

若自行配置，则选择板卡类型后，点击“添加”即可。

若是要查看配置，则双击配置行或者选中配置行，点击“配置”即可。

若是要删除配置，则选中配置行，点击“删除”。点击“删除所有”将删除所有的输入配置

注：由于目前会根据实际可操作板卡创建配置文件，因此不建议用户自行配置或删除配置！



输出配置包含数字输出配置和模拟输出配置（暂时保留）。目前主要支持 DAQ卡：

DAQ卡：PCI-1750（16DO）和 PCI-1756（32DO）

Motion卡：MVP-3245-MAS（16DO）

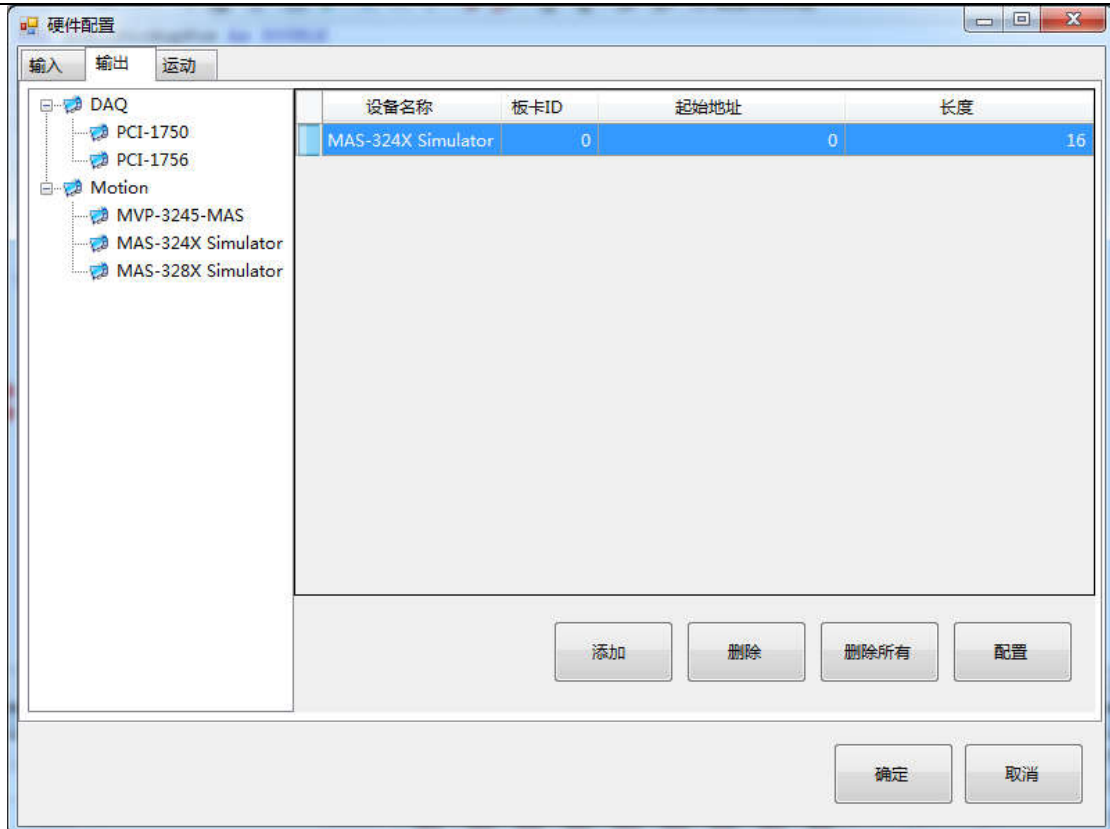
虚拟控制器：MAS-324X Simulator 和 MAS-328X Simulator

若自行配置，则选择板卡类型后，点击“添加”即可。

若是要查看配置，则双击配置行或者选中配置行，点击“配置”即可。

若是要删除配置，则选中配置行，点击“删除”。点击“删除所有”将删除所有的输出配置。

注：由于目前会根据实际可操作板卡创建配置文件，因此不建议用户自行配置或删除配置！



运动控制主要是对运动控制卡的配置，其中自动包含对每个轴上的 DI/DO 的配置。目前主要支持：

实体板卡：PCI-1245-MAS、PCI-1285-MAS 和 MVP-3245-MAS

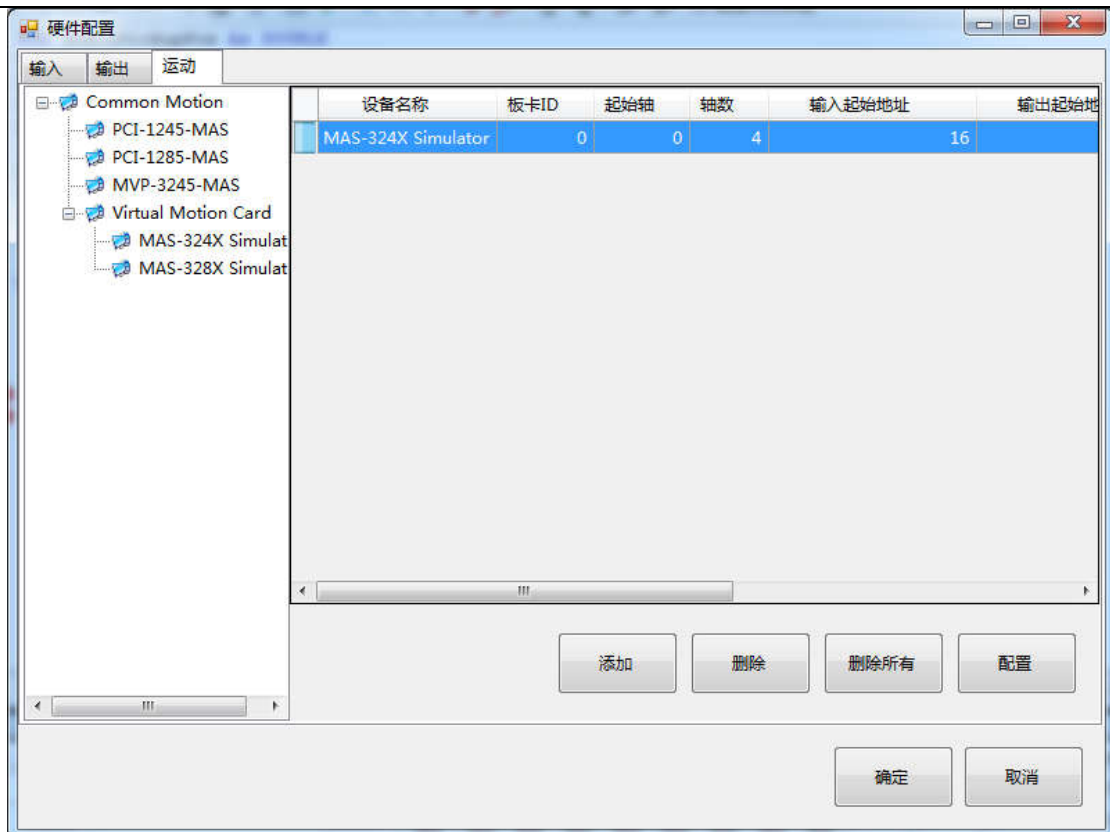
虚拟控制器：MAS-324X Simulator 和 MAS-328X Simulator

若自行配置，则选择板卡类型后，点击“添加”即可。

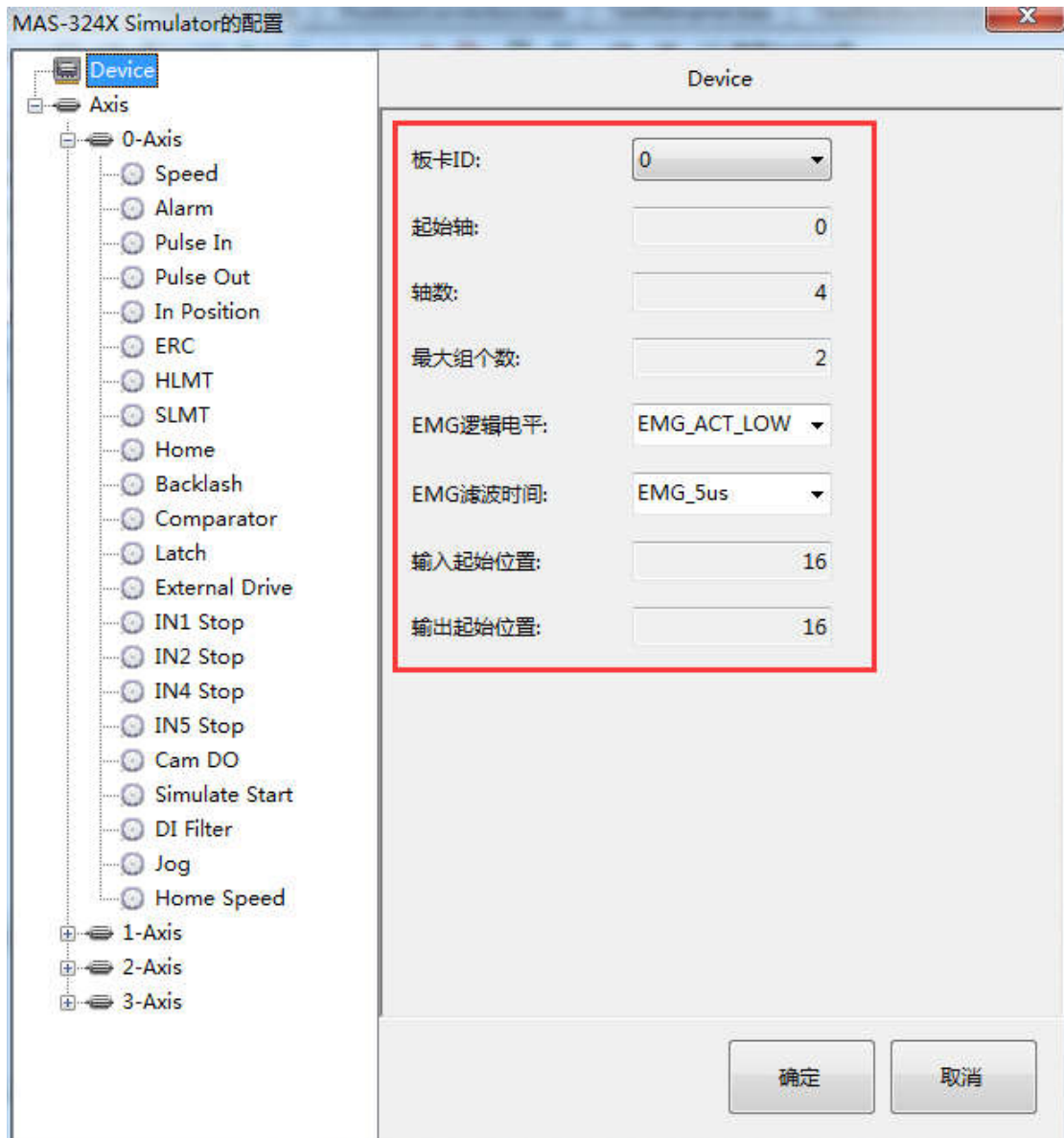
若是要查看配置，则双击配置行或者选中配置行，点击“配置”即可。

若是要删除配置，则选中配置行，点击“删除”。点击“删除所有”将删除所有的运动配置。

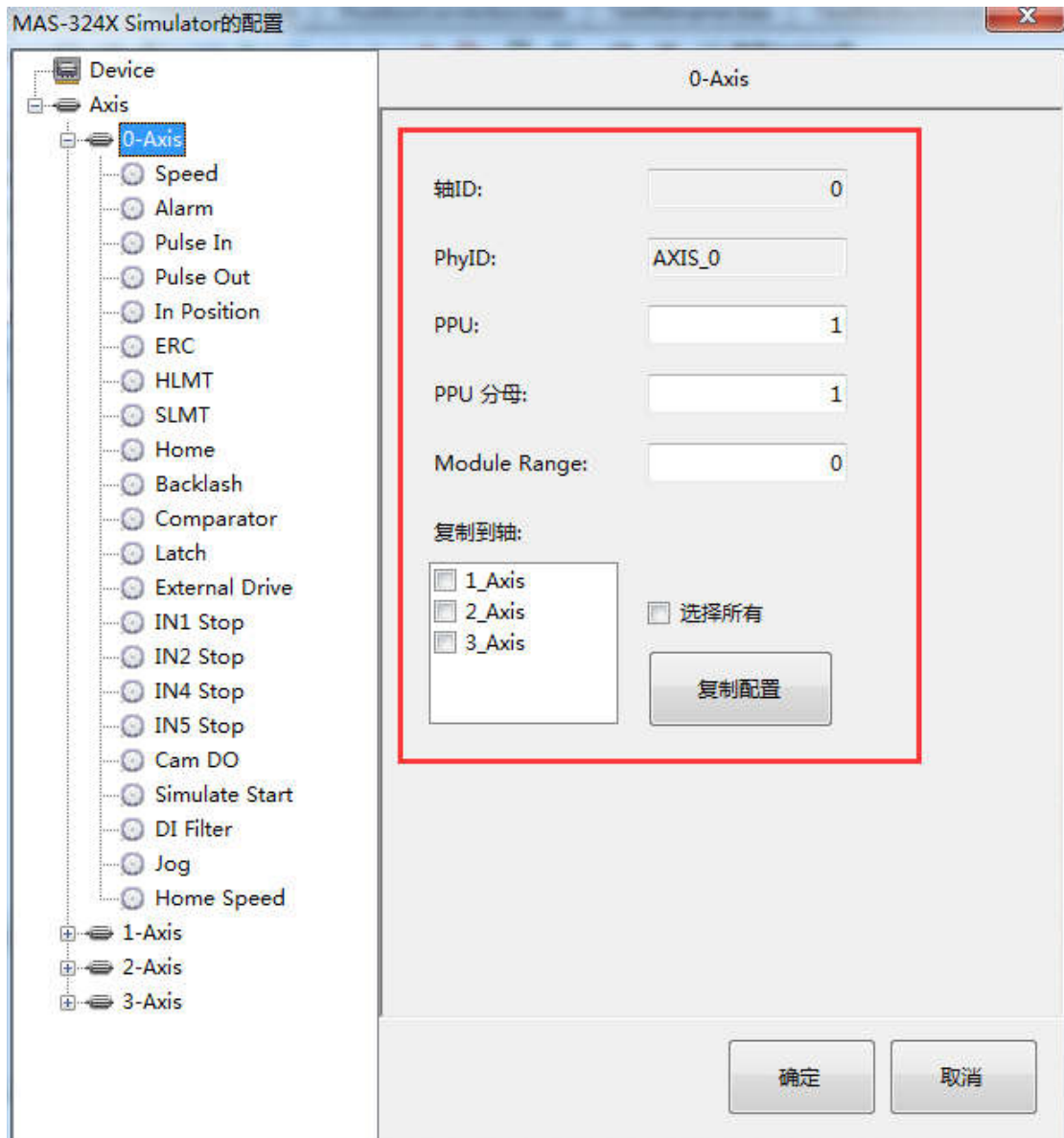
注：由于目前会根据实际可操作板卡创建配置文件，因此不建议用户自行配置或删除配置！



点击查看配置，可对运动控制卡是板卡配置或者轴的属性进行配置，双击配置行，会跳出如下界面：
如板卡的配置：

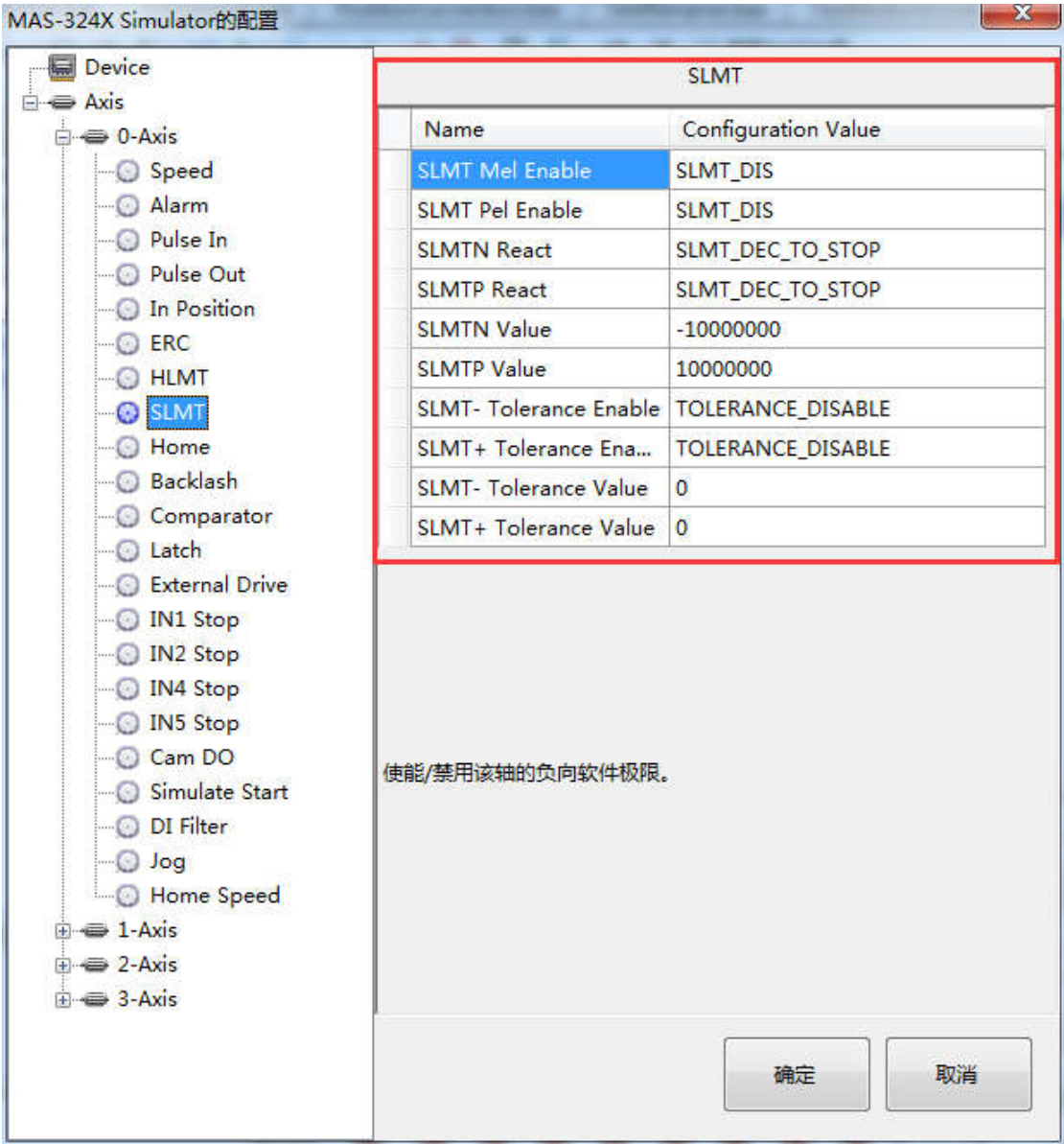


如轴的基本配置:



可选择将此轴的配置复制到其他选中轴。

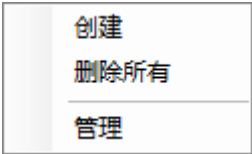
如配置轴的指定功能属性：



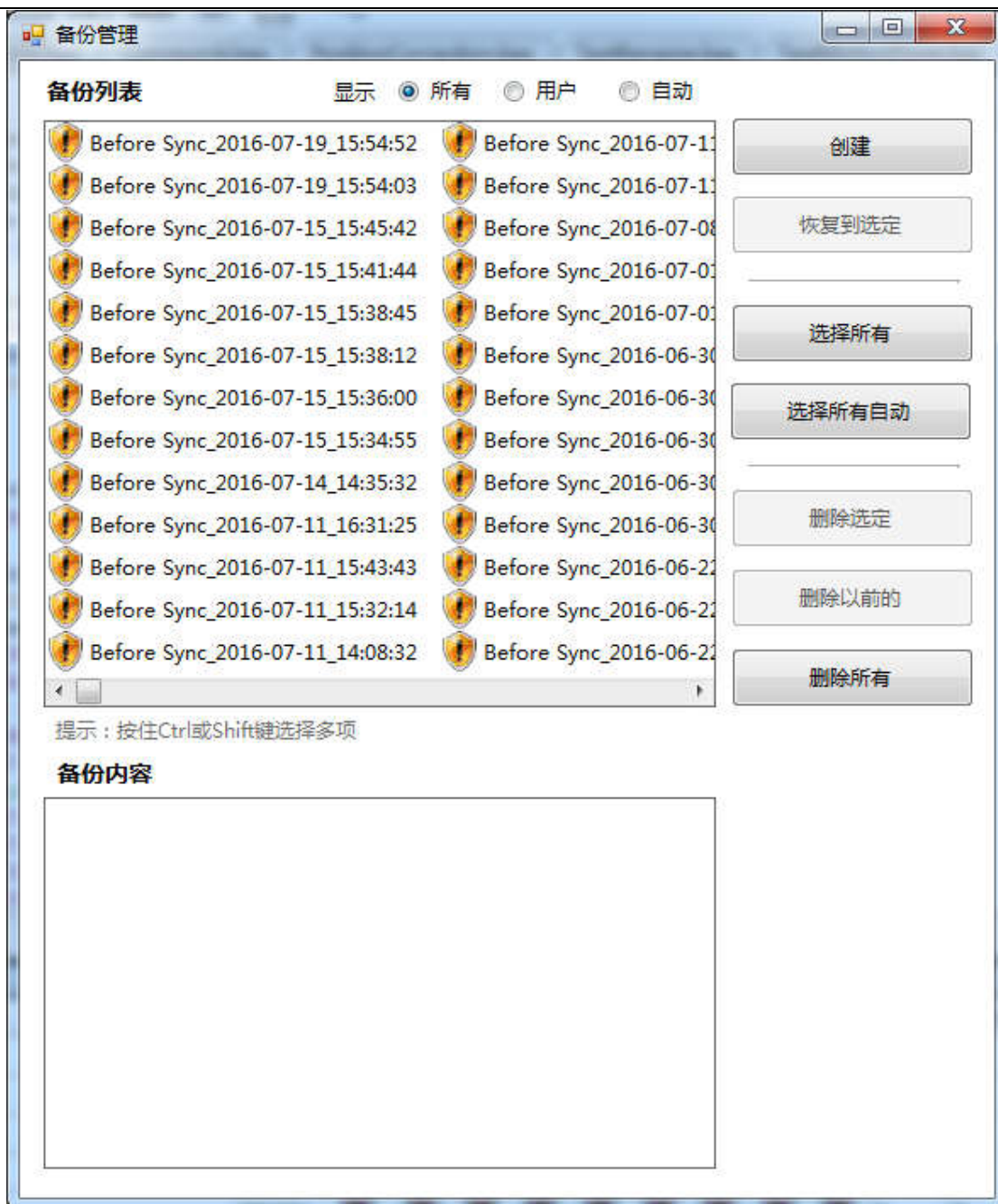
添加或更新配置后，点击“确定”将会保存配置，并将配置下载到控制器进行实体配置。

5.6 备份

展开备份节点，将显示所有备份信息的子节点，点击右键，会有以下右键菜单：



- 点击“创建”将新建指定名称的备份
- 点击“删除所有”将删除所有备份
- 点击“管理”，将跳出“备份管理”对话框，如下：



主体分为：备份列表，操作按钮和备份内容三大部分。

其中：

1) 备份列表：根据上面的单选框显示备份列表

所有：显示所有备份列表

用户：显示所有用户自定义的备份列表

自动：显示所有 Studio 自动备份列表

2) 操作按钮

创建：创建新的自定义备份

恢复到选定：选择某一个备份，点击后将恢复到选定备份

选择所有：将选择所有备份

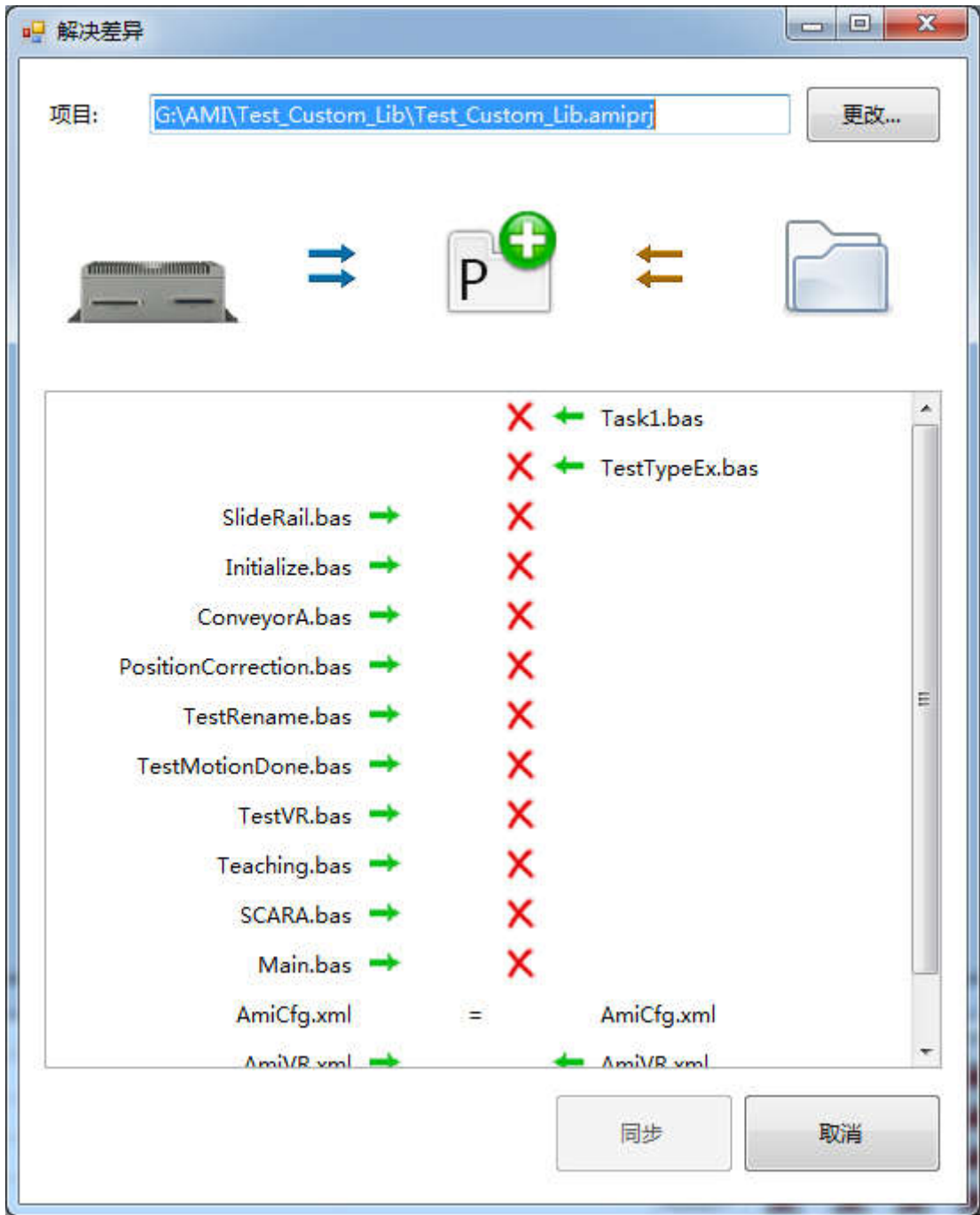
选择所有自动：将选择所有自动备份

删除选定：选择某一个备份，点击后将删除此备份

删除以前的：选择某一个备份后，点击后将删除此备份时间之前的所有备份
删除所有：将删除所有备份

5.7项目检查

当更改项目时，切换到同步模式前，会先比较当前的项目与控制器中内容，如果两者之间不相同，则会跳出以下对话框，解决差异。



项目：为目前选择的项目，可通过点击“更改”更改项目
各个图标及相关含义如下：

图标	功能	说明
----	----	----

	新建工程	新建一个工程
	下载项目	控制器的内容与项目相同
	上传项目	项目与控制器的内容相同
	下载文件	下载文件到控制器
	上传文件	上传文件到项目
	删除文件	删除文件
	比较文件	使用配置的比较工具比较文件后下载

对不同项操作后，对于的图标会灰阶。所有不同处都处理后，点击“同步”进行项目端与控制器端的内容同步。

特别说明：

不同处比较除了用户编辑的程序外，还有配置文件（对应配置模块，文件名为 AmiCfg.xml），输入输出变量信息（对应 I/O 表，文件名为 AmiVarDetail.xml），自定义变量信息（对应 VR 表，文件名为 AmiVR.xml）以及关于自定义库函数相应配置信息。由于输入输出变量信息与配置文件是相互关联的，因此列表中只列出 AmiCfg.xml；同理自定义库函数相关信息是相互关联的，因此列表中只列出 AmiCustomLib.bi。也即：

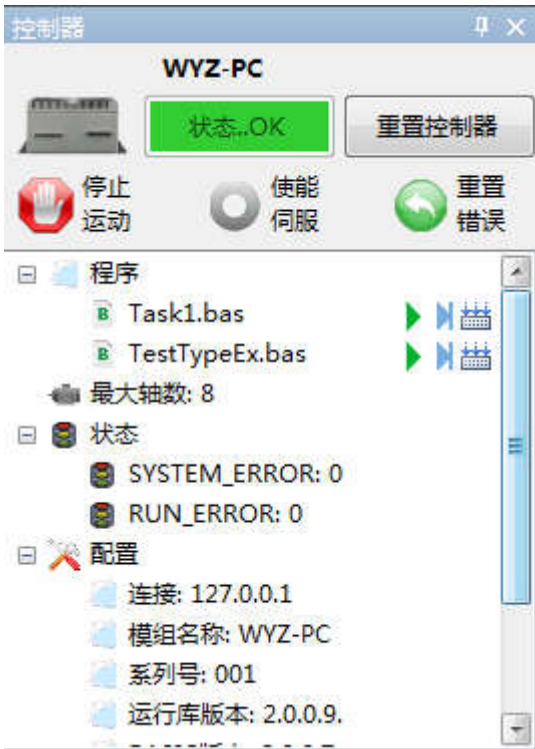
AmiCfg.xml：配置信息与输入输出变量信息

AmiVR.xml：自定义变量信息

AmiCustomLib.bi：自定义库函数相关信息

注：目前此三类文件不支持删除和比较，只支持上传/下载。

6 控制器



6.1 控制器相关操作

重置控制器：当控制器出现异常时，可通过点击重置控制器。

停止运动：点击后，所有运动将停止

使能伺服/禁用伺服：对伺服进行使能/禁用

重置错误：当运动轴卡的轴状态为 Error_Stopped 时，点击后轴的状态将恢复为 Ready

控制器状态：

颜色	文字	含义
	状态..OK	当下面的状态节点的子节点 SYSTEM_ERROR 和 RUN_ERROR 的错误信息都为 0,即没有任何错误时)
	系统执行错误	当下面的状态节点的子节点 SYSTEM_ERROR 和 RUN_ERROR 有错误时,但轴的状态不是 DISABLED 或者 Error_Stopped 时
	运行状态错误	轴的状态为 DISABLED 或者 Error_Stopped 时

注意：由于上述的操作，实际是通过 BASIC 指令完成，因为可能会更改默认的 BASE 对应的轴（操作轴）。因为若是在终端下命令，需要再下 BASE 命令，指定操作轴。

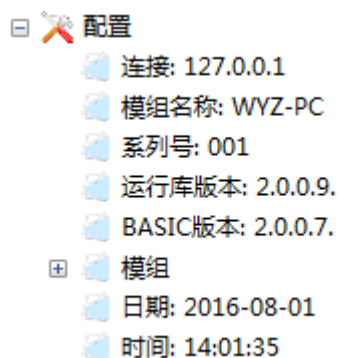
6.2 节点

程序：同项目中的程序

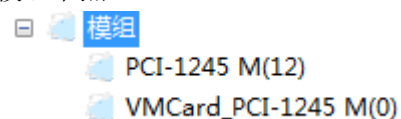
最大轴数：显示目前可操作的最大轴数

状态：显示目前系统的错误信息，目前包含系统错误（SYSTEM_ERROR）和运行错误（RUN_ERROR）两个部分。其中 SYSTEM_ERROR 为系统相关错误（对应指令 SYSTEM_ERROR）；RUN_ERROR 为运动相关的错误（包含所有轴的 RUN_ERROR 指令信息）

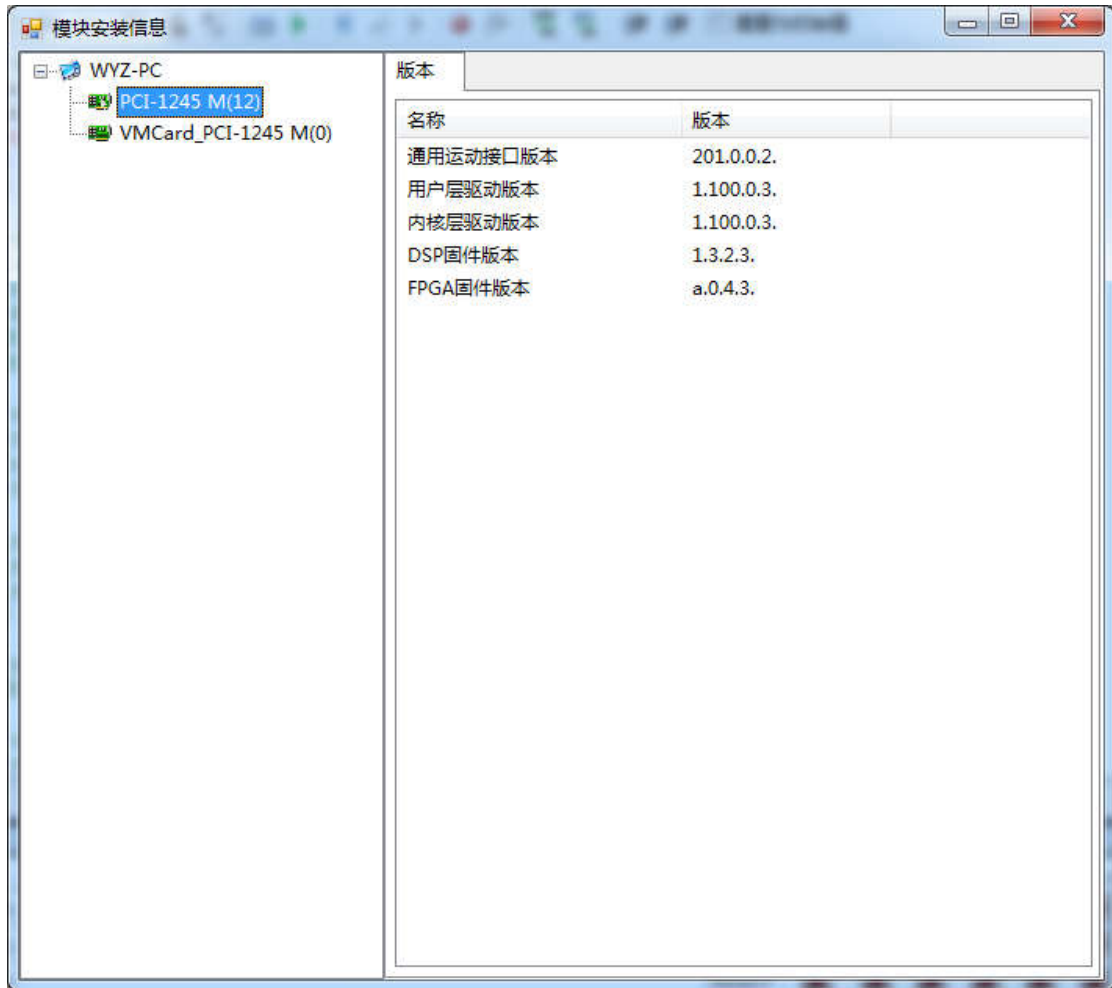
配置：



当只有一个板卡时，所有版本信息将作为单个节点列出；当板卡个数超过两个时，将会添加模组节点：



显示所有可用的板卡，双击节点，会跳出对话框，显示指定板卡的具体版本信息：



7 程序编辑器

程序编辑器用于编辑 BASIC 程序文件，并可对 BASIC 程序编译/调试/运行。

7.1 编辑功能



程序编辑器的编辑功能主要包含以下：

- 文字编辑（测试/重做/剪切/复制/粘贴/全选/不选）
- 关键字高亮显示
- 智能提示
- 高亮显示当前行
- 显示行号
- 查找/替换
- 跳转到指定行
- 注释/取消注释选中行
- 行(块)增加/减少缩进
- 向前/向后定位
- 行标识 (书签)
- 文本缩放：按住 Ctrl+滚轮上/下滚动

7.2 构建/调试功能

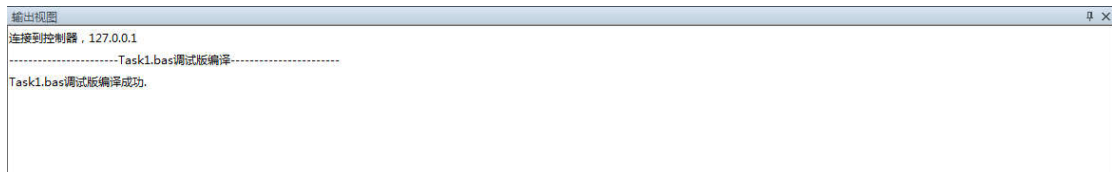
程序编辑器上的工具按钮提供编辑/编译/调试/运行的快捷方式。



图标	功能	说明
	保存	（若有改动）保存程序，并下载到控制器（同步模式）
	剪切	将当前选定的文本剪切到剪贴板
	复制	将当前选定的文本复制到剪贴板中
	粘贴	从剪贴板中粘贴文本
	撤销	撤销最后一次的编辑操作
	重做	重新操作最后一次取消的编辑操作
	跳转	跳转到指定行

	查找	跳出“查找”对话框查找指定文字
	替换	跳出“替换”对话框替换指定文字
	编译	编译当前活动的程序（如有改动，会先保存）
	发布版编译运行	发布版运行/停止当前活动的程序
	调试版运行	调试版运行/停止当前活动的程序
	单步	单步调试（不进入子程序）
	继续运行	继续运行当前活动的程序
	切换断点	在当前行上设置/移除断点
	清除断点	删除所有断点
	注释	注释当前选中内容
	取消注释	取消注释当前选中内容
	减少缩进	减少缩进当前选中行/块
	增加缩进	增加缩进当前选中行/块
	重置为初始值	此功能与 I/O 表，VR 表有关联：当勾选此复选框时，在此程序运行前，会把设置 I/O 表和 VR 表中的变量值设置为对应的初始值（若未设置，则默认为 0）

8 输出视窗



输出视窗显示所有提示信息、错误信息以及运行程序中所有的打印信息（调试版运行时）。

9 工具

9.1 轴属性

轴属性工具用于查看选定轴的当前属性值。

第一列：复选框若勾选，该属性默认为正常显示；若不勾选，则该属性会隐藏显示

第二列：参数名，与参数的 BASIC 关键字一致

后续每列为每个选定轴对应的属性值。



工具按钮

图标	功能	说明
	显示所有参数	切换是否显示所有参数
	选择轴	跳出对话框，显示可操作轴序号，可更改选择轴
	定时刷新	使能/禁用定时刷新
	刷新数据	点击后刷新一次数据



9.2轴测试

轴测试工具用于测试选定轴运动是否正常。



按顺序每列的列名与含义分别如下：









































































图标	功能	说明
	轴	轴序号，轴名称
	伺服	表示当前伺服的状态，也可点击切换SVON/SVOFF
	负向点动运动	按下鼠标开始运动，松开停止运动

	正向点动运动	按下鼠标开始运动，松开停止运动
	选择 PTP 运动 /Homing 运动	背景色为蓝色的表示为当前的运动模式
	负向运动	根据前面的选择进行负向 PTP 运动/Homing 运动
	正向运动	根据前面的选择进行正向 PTP 运动/Homing 运动
	停止运动	停止该轴的当前运动（PTP 运动/Homing 运动）
	点到点距离	PTP 运动的参数：单次运动的距离
	DPOS	当前轴的理论位置
	配置运动参数	配置运动参数。可配置 PTP/Homing/Jog 运动的参数

点击选择轴，可显示所有可操作轴，可切换操作轴（同轴参数工具中的选择轴）。

9.3轴状态

轴状态工具显示所有可用轴的当前状态（包含常用的 DI/DO 状态）。

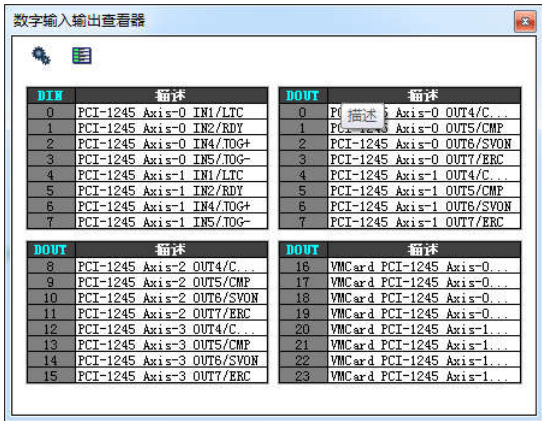
轴状态															
轴	描述	STATE	SVON	EL+	EL-	ORG	ALM	SEL+	SEL-	INP	EMG	DPOS	MPOS	DSPEED	RUN_ERROR
轴 (0)	轴 (0)	READY										100	0	0	SUCCESS
轴 (1)	轴 (1)	READY										0	0	0	SUCCESS
轴 (2)	轴 (2)	READY										0	0	0	SUCCESS
轴 (3)	轴 (3)	READY										0	0	0	SUCCESS
轴 (4)	轴 (4)	READY										0	0	0	SUCCESS
轴 (5)	轴 (5)	READY										0	0	0	SUCCESS
轴 (6)	轴 (6)	READY										0	0	0	SUCCESS
轴 (7)	轴 (7)	READY										0	0	0	SUCCESS

列名	说明
轴	轴序号，轴的名称
描述	对轴的描述说明，默认为轴名称
STATE	当前轴的状态，具体含义请参数 BASIC 指令 STATE
SVON	当前轴的 SVON 状态：ON/OFF，ON 为亮，OFF 为暗
EL+	当前轴的正向硬件极限（EL+）的状态
EL-	当前轴的负向硬件极限（EL+）的状态
ORG	当前轴的原点（ORG）的状态
ALM	当前轴的 Alarm 信号（ALM）的状态
SEL+	当前轴的正向软件极限（SEL+）的状态
SEL-	当前轴的负向硬件极限（SEL-）的状态
INP	当前轴的到位信号（INP）的状态
EMG	当前轴的紧急停止信号（EMG）的状态
DPOS	当前轴的理论位置

MPOS	当前轴的实际位置
DSPEED	当前轴的当前理论速度
RUN_ERROR	当前轴运动的最后错误，具体含义请参看 BASIC 指令 RUN_ERROR

9.4数字输入输出表

数字输入输出表可查看指定 DI/DO 的状态，可点击切换 DO 的状态。



其中，每一块 DI/DO 的显示都以 Byte 为单位，即 8 个 DI/DO。

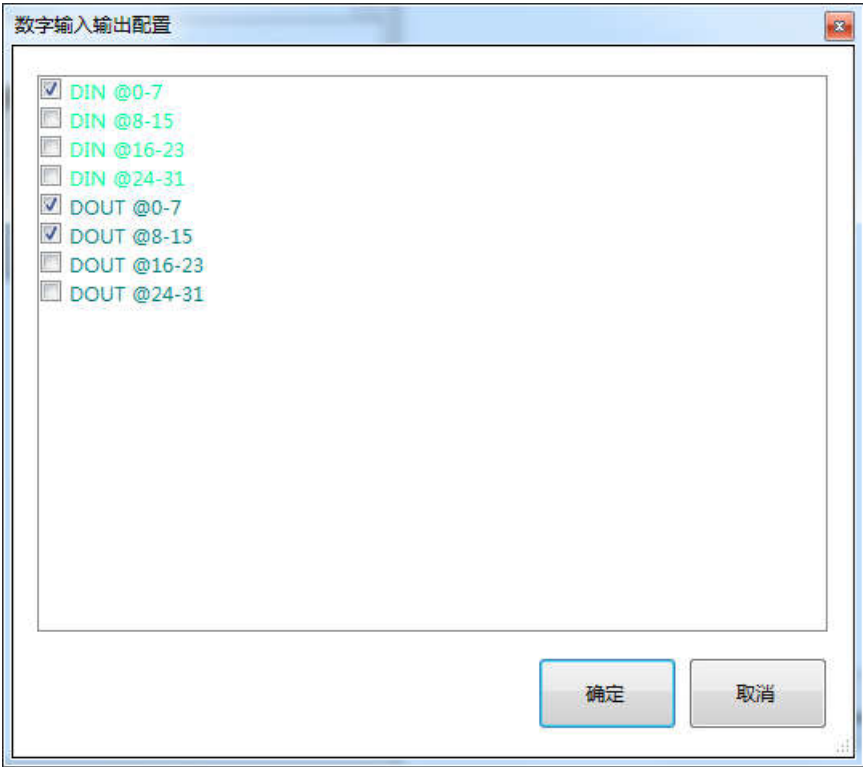
列名：DIN/DOU 表示为数字输入或者输出，值即为此数字输入/输出序号。

描述：对应数字输入/输出的说明。此说明与 I/O 表中的说明一致，可在 I/O 表中对应栏编辑，此处只读。

工具按钮

图标	功能	说明
	配置	配置要显示的 DIN/DOU
	显示/隐藏	切换显示/隐藏描述栏

点击配置，弹出的对话框如下：



9.5VR 表

具体请参见 5.3 VR 表

9.6观察变量



观察变量工具用于查看变量（全局变量/局部变量/Motion 参数）值。

观察变量		
名称	值	上下文
aa	Invalid Value	Task1.bas
VR(0)	-2147410000	VR
DPOS AX(0)	0	AXIS(0)
STATE AX(0)	1	AXIS(0)

添加变量的方法

- 1) 在编辑器中选中变量后，按快捷键 Ctrl+W
- 2) 在编辑器中选中变量后，拖拉到观察变量中
- 3) 直接在观察变量中的名称栏写入变量名

工具按钮

图标	功能	说明
	定时刷新	使能/禁用定时刷新
	刷新数据	点击后刷新一次变量值

特别说明：

若查看轴的属性/或状态，需要在后面添加轴序号，比如查看 0 轴的 ACC 属性，变量名为 ACC AX (0)。

9.7 终端

终端提供用户直接下 BASIC 指令的一种方式。

目前只支持单次一行指令。



指令可以为命令指令，如 MOVE 1000；也可以是询问状态或值的命令，如 ? ACC，即为查询操作轴的 ACC 属性值。

注：若是对轴的操作或查看轴的属性/状态，需要先指定操作轴，即先下 BASE 指令，如 BASE 0。

终端的工具栏（最下面）显示的是宏定义按钮。点击一次宏按钮，则终端下一次对应按到 BASIC 指令。点击右下角的图标，会出现如下对话框，可编辑宏定义按钮。



● 宏按钮

界面可分为三块：

工具栏预览：即预览编辑后在终端工具栏（最下面）宏定义按钮的显示预览，可通过拖拉宏定义按钮更改顺序

按钮列表：显示已编辑的宏定义按钮的标题与对应的 BASIC 语句

列名	说明
勾选	勾选用于表示此宏按钮是否显示在工具栏
标题	宏按钮的标题
命令	此宏按钮的 BASIC 指令

注意：标题不能重复，否则会忽略。

操作按钮：

按钮	说明
添加	点击后将添加一个宏按钮
移除	选中一行后，点击，将移除此宏按钮
取消	取消此次编辑并关闭此窗体

确定	确认此次编辑并关闭此窗体
----	--------------

菜单项

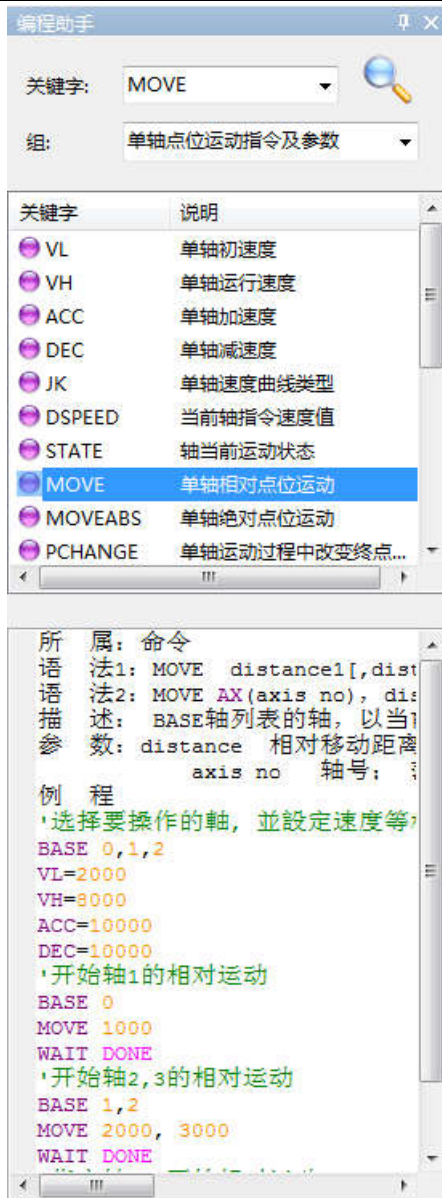


点击编辑菜单或者在 Terminal 中右键，将跳出如下菜单项

- 1) 清除所有，将清空目前 Terminal 的内容
- 2) 复制：复制目前选中的内容
- 3) 粘贴：将剪贴板的内容粘贴到 Terminal 中
- 4) 复制，粘贴和运行：将选中的内容复制，粘贴并运行

9.8编程助手

编程助手工具主要为方便用户查询 BASIC 指令的含义和使用，对每个指令有简要的说明并提供列程帮助用户理解和使用 BASIC 指令。



编程助手主界面可以分为三大块：

- 1) 用户可输入关键字点击查询或者切换组，查看对应组别包含的所有关键字
- 2) 指令列表：对应组的所有指令以及每个指令的简要说明
- 3) 指令说明与列程：输入关键字，点击查询（若存在）或者直接在指令列表中选择某个指令，下面将列出对应指令的语法，描述，参数以及例程，帮助用户理解和使用此 BASIC 指令。

关键字的输入框可进行模糊匹配，即随着输入自动下拉列出模糊匹配到的关键字。

编程助手工具的显示方式目前有两种：

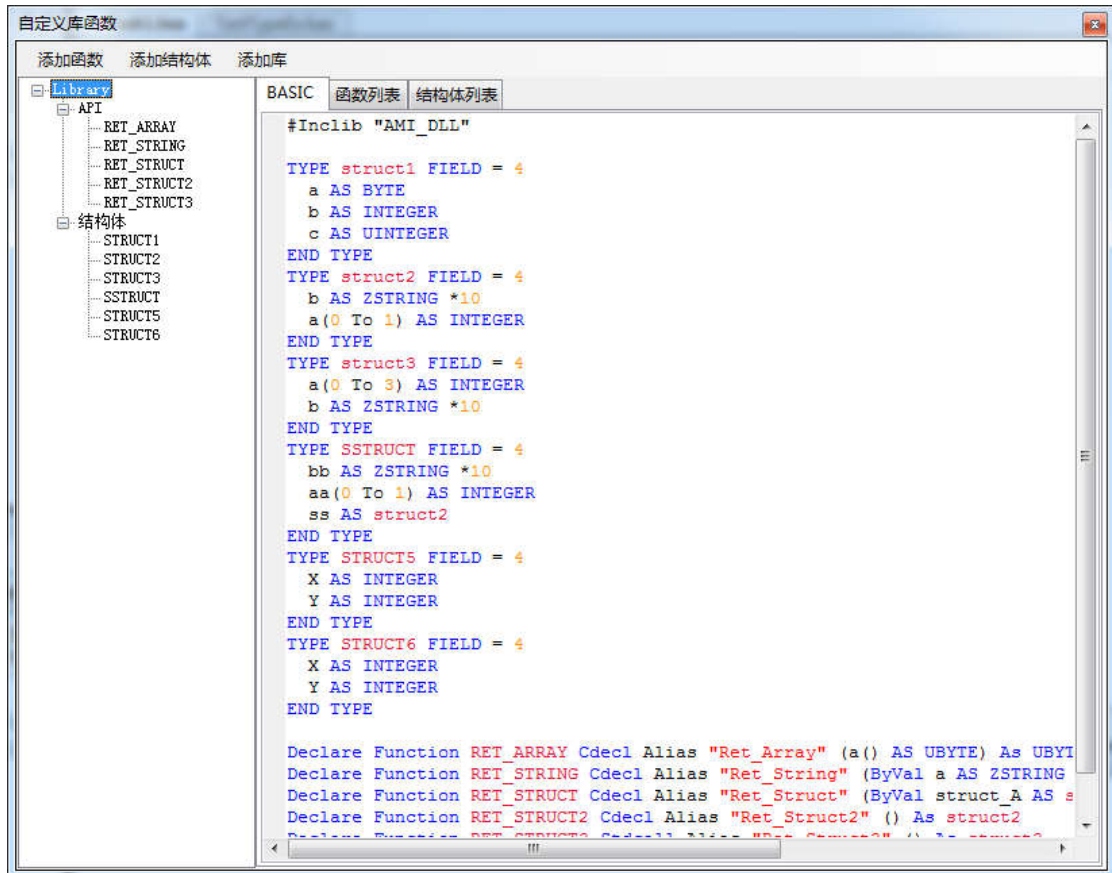
- 1) 菜单：工具——>编程助手
- 2) 在编辑框中选中关键字后，按 F12：将跳出编辑助手，且跳转到对应关键字

9.9 自定义库函数

为扩展支持其他相关硬件（如相机）的功能，AMI 提供接口方便用户自定义 C dll 中的

API 在 BASIC 中的关键字（即提供从 C 接口转换为 BASIC 接口的功能）。

● 主界面



可分为以下三大块：工具按钮，自定义库函数树和内容窗体

1) 工具按钮

添加函数：点击后将跳出“添加函数”窗体

添加结构体：点击后将跳出“添加结构体”窗体

添加库：点击后将跳出“添加库函数”窗体

2) 自定义库函数树：列出目前已定义的自定义 API 和结构体

点击“Library”节点，内容窗体三个页面：BASIC、函数列表和结构体列表

点击“API”：内容窗体将显示“函数列表”窗体

点击“结构体”：内容窗体将显示“结构体列表”窗体

点击具体 API/结构体：内容窗体将显示此 API/结构体的具体配置信息

3) 内容窗体

包含以下三个页面

BASIC：自定义库函数对应的 BASIC 接口

函数列表：显示已定义的函数列表。双击对应函数行，可查看/编辑此函数

结构体列表：显示已定义的结构体列表。双击对应结构体行，可查看/编辑此结构体

● BASIC

```
#Inclib "AMI_DLL"

TYPE struct1 FIELD = 4
  a AS BYTE
  b AS INTEGER
  c AS UINTEGER
END TYPE
TYPE struct2 FIELD = 4
  b AS ZSTRING *10
  a(0 To 1) AS INTEGER
END TYPE
TYPE struct3 FIELD = 4
  a(0 To 3) AS INTEGER
  b AS ZSTRING *10
END TYPE
TYPE SSTRUCT FIELD = 4
  bb AS ZSTRING *10
  aa(0 To 1) AS INTEGER
  ss AS struct2
END TYPE
TYPE STRUCT5 FIELD = 4
  X AS INTEGER
  Y AS INTEGER
END TYPE
TYPE STRUCT6 FIELD = 4
  X AS INTEGER
  Y AS INTEGER
END TYPE

Declare Function RET_ARRAY Cdecl Alias "Ret_Array" (a() AS UBYTE) As UBYT
Declare Function RET_STRING Cdecl Alias "Ret_String" (ByVal a AS ZSTRING
Declare Function RET_STRUCT Cdecl Alias "Ret_Struct" (ByVal struct_A AS s
Declare Function RET_STRUCT2 Cdecl Alias "Ret_Struct2" () As struct2
Declare Function RET_STRUCT3 Cdecl Alias "Ret_Struct3" () As struct3
```

● 函数列表

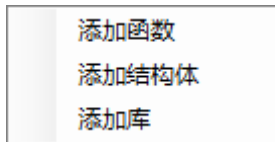
BASIC名称	函数名称	函数库名	调用方式	参数个数
RET_ARRAY	Ret_Array	AMI_DLL.dll	C_call	2
RET_STRING	Ret_String	AMI_DLL.dll	C_call	2
RET_STRUCT	Ret_Struct	AMI_DLL.dll	C_call	3
RET_STRUCT2	Ret_Struct2	AMI_DLL.dll	C_call	1
RET_STRUCT3	Ret_Struct3	AMI_DLL.dll	Stdcall	1

● 结构体列表

名称	对齐模数	成员个数
struct1	pragma_pack_4	3
struct2	pragma_pack_4	2
struct3	pragma_pack_4	2
SSTRUCT	pragma_pack_4	3
STRUCT5	pragma_pack_4	2
STRUCT6	pragma_pack_4	2

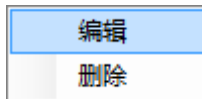
● 菜单

点击 Library/API/结构体节点，点击右键，会弹出以下右键菜单：



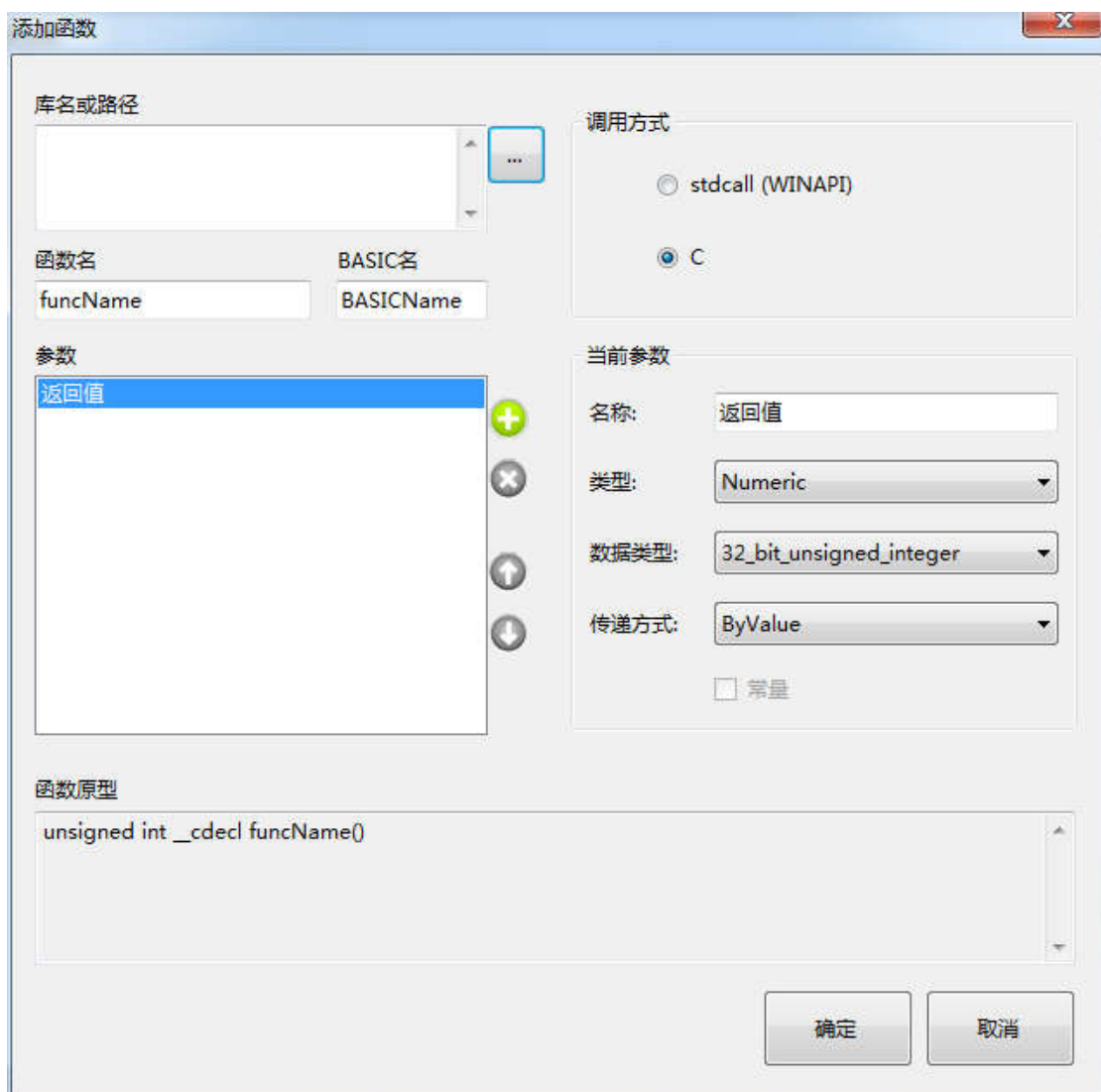
菜单功能与工具按钮的功能一致。

点击具体 API/结构体，点击右键，会弹出以下右键菜单：



可对此 API/结构体进行编辑或者删除。

● 添加函数



添加 API 提供以下输入参数：

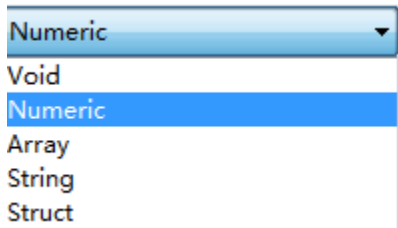
- 库名或函数：输入库函数的名字（默认首先查找 Studio EXE 所在路径；若 Studio EXE 没有找到，则查找系统路径）或者库函数的所在路径+库函数的名字。
- 函数名：API 的名字（注：必须与 dll 中的 API 名字一致）。
- BASIC 名：BASIC 中定义的 API 名字（BASIC 中的关键字，注不能与已有的 BASIC

关键字重复)。

- 调用方式：选择为 Stdcall 或者 C 调用方式，默认为 C 调用。
- 参数：API 中的参数，左边列表列出已定义的参数名，选择后，右边为当前参数的具体信息；可对参数进行添加/删除/上移/下移的操作。
- 当前参数：显示/编辑当前 Parameter 的信息，包含名称，类型，数据类型和传递方式。

名称：除第一个参数名默认为返回值（只读）外，其他函数名只能为字符，数字和下划线。

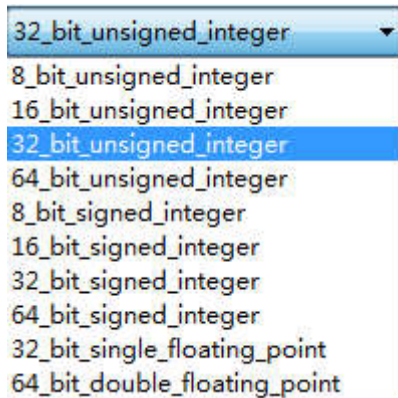
类型：此参数的类型，分为 Void（无，目前只支持返回值），Numeric（数字），Array（数组），String（字符串）和结构体（Struct）



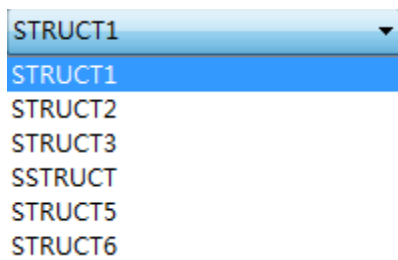
数据类型：根据所选类型而变化。

当类型为 Void 或者 String 时，没有数据类型

当类型为 Numeric 或者 Array 时，数据类型有如下几类：

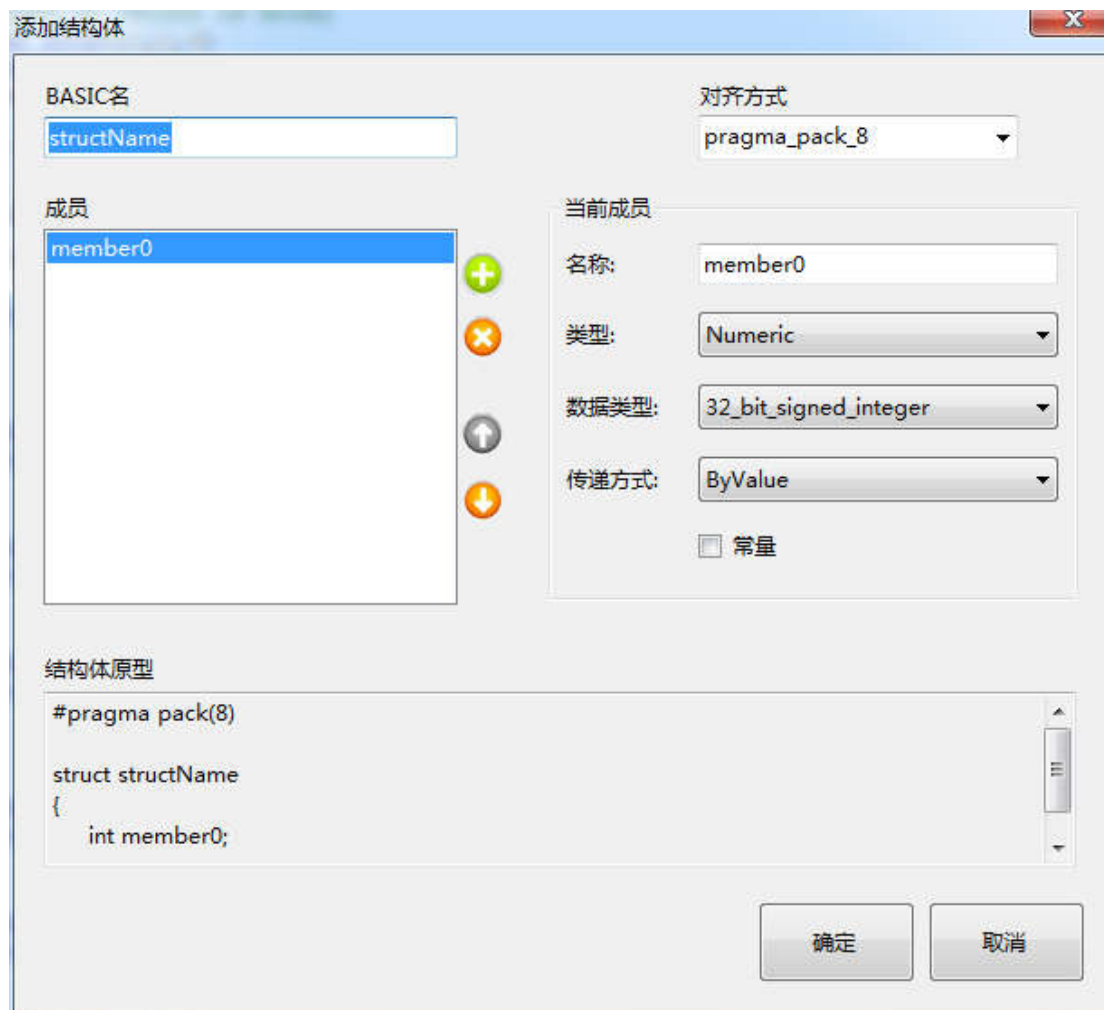


当类型为 Struct 时，数据类型将列出已添加支持的结构体类型，如下：



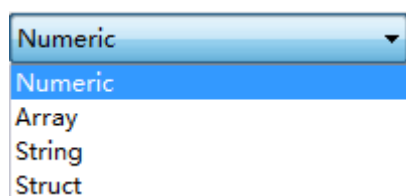
常量：勾选表示参数为常量

- 函数：只读，显示此 API 的 C 语言声明。
- 点击“确定”后，添加此 API 的 BASIC 接口；点击“取消”，取消此次编辑。
- 添加结构体



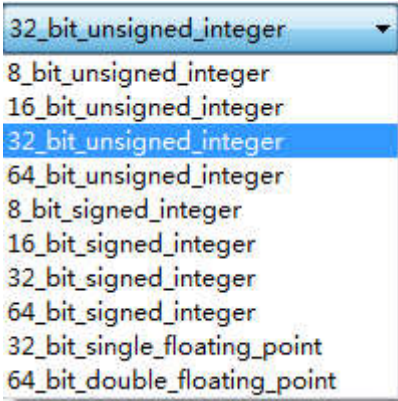
添加结构体需配置如下参数：

- 名称：结构体的名字，BASIC 关键字，不可与已存在的 BASIC 关键字重复
- 对齐模式：可选择 1，2，4，8，16。默认为 8。
- 成员：结构体中的成员，左边列表列出已定义的成员名，选择后，右边为当前成员的具体信息；可对成员进行添加/删除/上移/下移的操作。
- 当前成员：显示/编辑当前成员的信息，包含名称，类型，数据类型和传递方式。
 名称：除第一个参数名默认为返回值（只读）外，其他函数名只能为字符，数字和下划线。
 类型：此参数的类型，分为 Numeric（数字），Array（数组），String（字符串）和结构体（Struct）



数据类型：根据所选类型而变化。

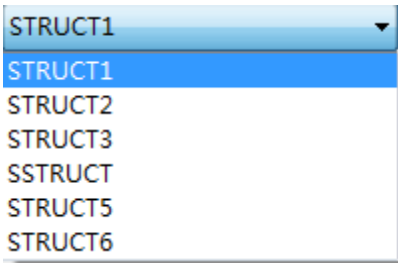
当类型为 Numeric 或者 Array 时，数据类型有如下几类：



当类型为 String 时，没有数据类型，需要填写元素个数（同数组类型），即字符串的长度

元素个数:

当类型为 Struct 时，数据类型将列出已添加支持的结构体类型，如下：

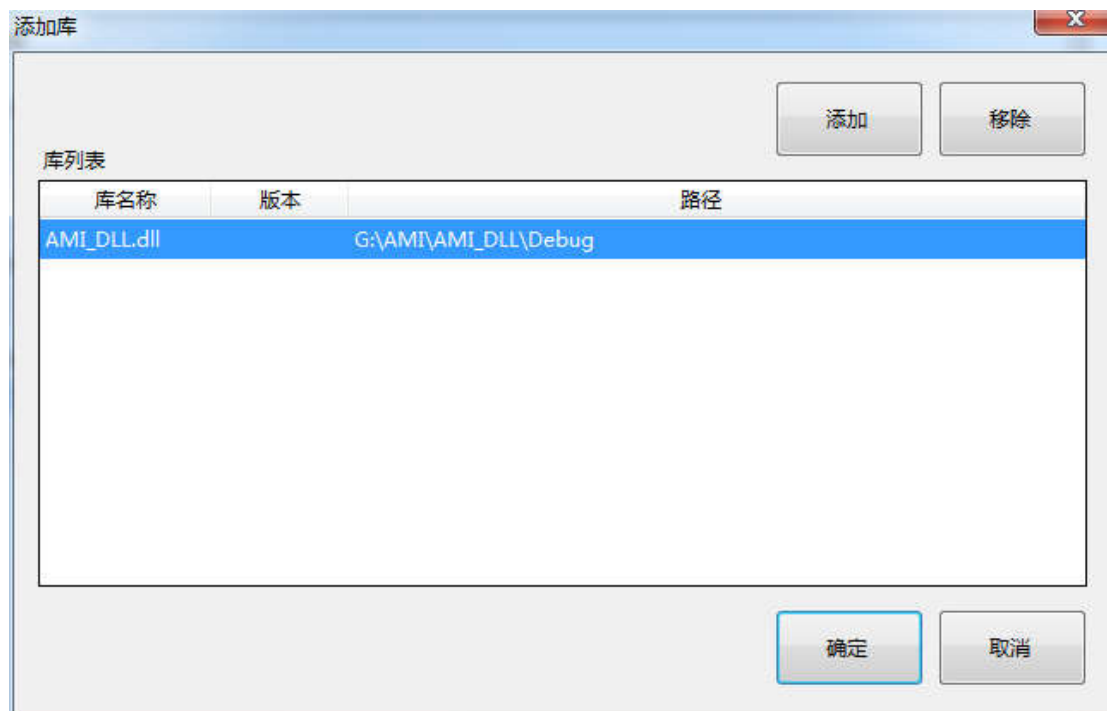


常量：勾选表示参数为常量

- 结构体原型：只读，显示此结构体的 C 语言声明。

点击“确定”后，添加此结构体的 BASIC 接口；点击“取消”，取消此次编辑。

- 添加库



添加库页面将会列出目前已添加的库列表，可添加或移除已有的库。

每一行会显示此库的名称，版本（若存在）以及库路径。

点击“确定”将确定本次编辑；点击“取消”，将取消此次编辑。

特别说明：自定义库函数工具实际是提供一个 C 到 BASIC 的转换工具，最终产生的 BASIC 页面即为对应的 BASIC 接口。具体使用还需要考虑 FreeBASIC 的语法规则。

在此，提供 FreeBASIC 手册网址：<http://www.freebasic.net/wiki/wikka.php?wakka=DocToc>。以供学习，参考。

9.10 选项

选项主要为配置 Studio 的工具的可选设置。

目前分为以下三类：

常规：关于 Studio 和工具的常用可选设置。

编辑器：关于编辑器的常用可选设置。

项目：关于项目的常用可选设置。



● 常规

主要分为两项

1) 启动模式：Studio 开启后的启动模式

选项	含义
上次关闭的模式	Studio 会记录上次关闭时的模式，打开会设置为关闭时的模式。若关闭时为断线模式，连线后会自动切换为监控模式
断线模式	Studio 打开后自动设置为断线模式
监控模式	Studio 打开后自动设置为监控模式
同步模式	用最后使用的项目与控制器建立同步连接
启动后提示	Studio 每次启动后跳出对话框，选择操作模式

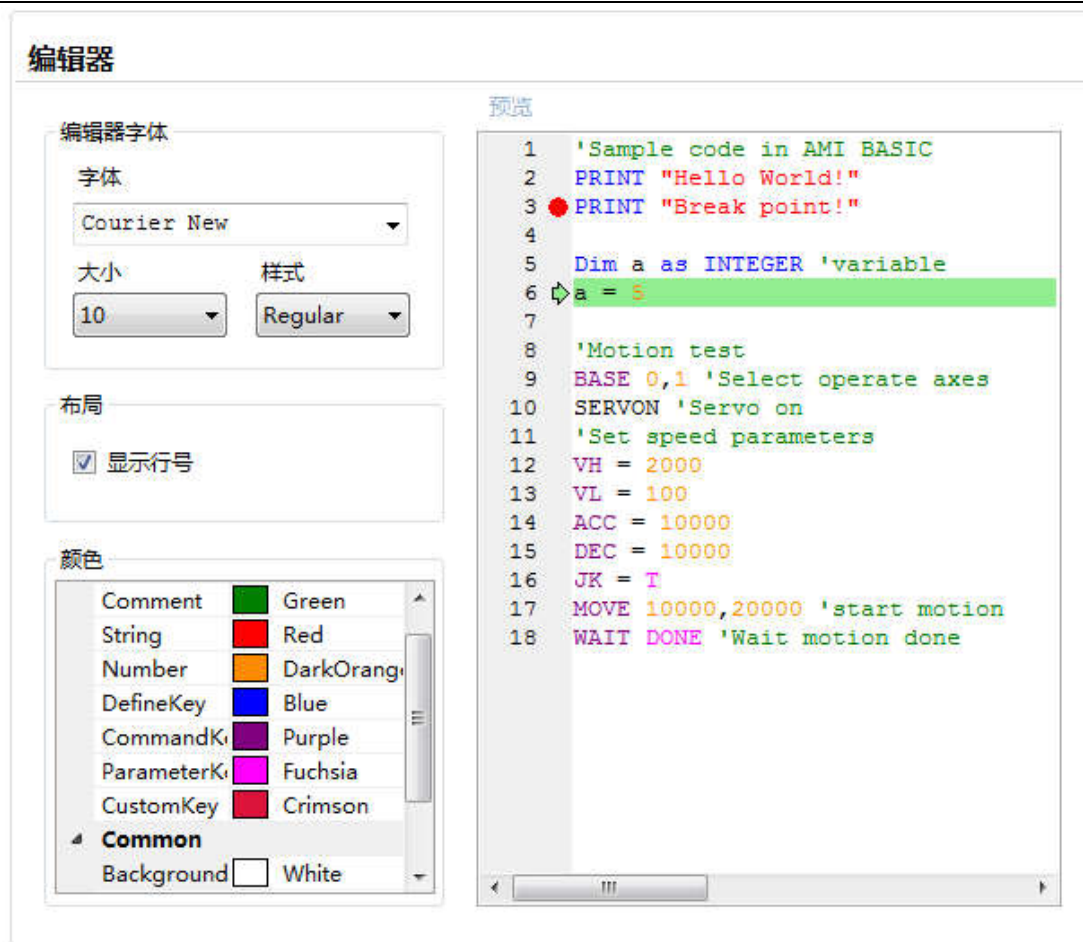
2) 刷新频率

对 Studio 中的工具设置刷新频率。

注意：刷新频率过高，可能会增加 Studio 的负荷，影响整体使用效果。因此，建议在非必须的情况下，不要设置过高的刷新频率。

● 编辑器

对编辑器的字体，关键字的颜色等进行设置



左边为设置，右边为预览图。

设置部分分为三大块：

- 1) 字体：包含字体名称，大小和样式
- 2) 布局：是否显示行号
- 3) 颜色：关键字的颜色

目前分为以下几类：

Comment：注释文字

String：字符串

Number：数字

CommandKey：MAS BASIC 命令相关以及系统相关关键字

ParameterKey：MAS BASIC 属性/状态相关关键字

CustomKey：用户自定义的库函数中的相关关键字（API 和结构体）

Background：背景色

Text：文字（非关键字）颜色

Breakpoint：背景色

DebugStep：调试单步高亮行颜色

● 项目

项目相关的常用配置

项目

同步

项目比较/合并工具:

自定义...

☒ 下载项目时列出自动运行的程序
 ☒ 更改控制器类型时显示警告

备份

从备份恢复项目时，也恢复

☐ 桌面布局
 ☒ 工具配置

1) 同步

配置项目比较/合并工具:

同步时，处理差异，点击比较项时，使用配置的比较工具进行两者文件的比较

目前 Studio 默认下载项目时列出自动运行的程序，控制器类型与配置不一致时会跳出对话框提示。

2) 备份

从备份恢复项目时，Studio 会默认恢复工具配置

注：目前桌面布局暂时不可恢复