

GUTTA Ladder Editor 安装和使用

IN1003 PLC 编程基础节选

CREATE: 2009/06/25

UPDATE: 2010/12/06

Version 1.1

GUTTA Ladder Editor Version 1.1

<http://www.plcol.com>

<http://www.visiblecontrol.com>

工具下载.....	2
软件的安装.....	2
您的第一次 PLC 之旅.....	5
更加深入的调试.....	14
主菜单介绍.....	19
快捷键一览.....	24
程序编辑介绍.....	24
梯形图编辑介绍.....	26
指令表编辑介绍.....	29
数据块编辑介绍.....	30
系统块编辑介绍.....	30
函数符号表编辑介绍.....	31
变量符号表编辑介绍.....	31
交叉引用表示用介绍.....	32
查找替换对话框使用介绍.....	32
总结.....	33

工具下载

目前市面上的 PLC 编程软件各不相同，但基本的原理是相通的。其实只要掌握其中一种 PLC 的编程，在遇到新的 PLC 系统时，相信大家都能很快的上手。本教程中所有的例题和习题都是基于 GUTTA PLC 系统平台。相关软件和文档大家可以通过访问我的站点 <http://www.plcol.com/> 免费下载。

GUTTA Ladder Editor 是一款功能全面的 PLC 编程软件。

GUTTA Simulator 是为 GUTTA Ladder Editor 配套的软件模拟器。

学习 PLC 编程和学习任何一种编程语言一样，最好的学习方法就是自己动手写程序。相信很多 PLC 初学者是在手头上没有 PLC 的情况下学习 PLC 编程的，这种情况造成了学习过程中的实践经验的严重不足。特别是做完习题，也没有办法检验结果是否正确。其实学习 PLC 编程最好的方法就是写程序、上机验证、分析现象、修改程序、再上机验证这么一个反复的过程。只有通过这么一个边动脑边动手的过程，才能真正把 PLC 的编程概念建立起来，把 PLC 编程的精髓掌握。现在借助于 GUTTA Simulator 这个软件模拟器，我们就有了一个统一的平台和实验标准。教程中的例子也基本上都能在这个模拟器上运行。因此希望大家在完成习题的时候，也尽量通过这个模拟器来检验自己的程序。

除了软件模拟器，我们还提供基于各种 CPU 平台的 PLC 实验板。相比 PLC 软件模拟器，PLC 试验板能够完成一些更高级的功能：菜单编写、通讯控制、高速计数等。除了实现工业 PLC 的所有功能外，还加入了一些方便学习和调试的特点：具有多个按键输入、跳线选择 RS232 和 RS485 等。PLC 实验板同时也可以做单片机开发板使用，用于 PLC 的固件开发。

软件的安装

安装程序包下载下来以后，看起来应该是这个样子的：

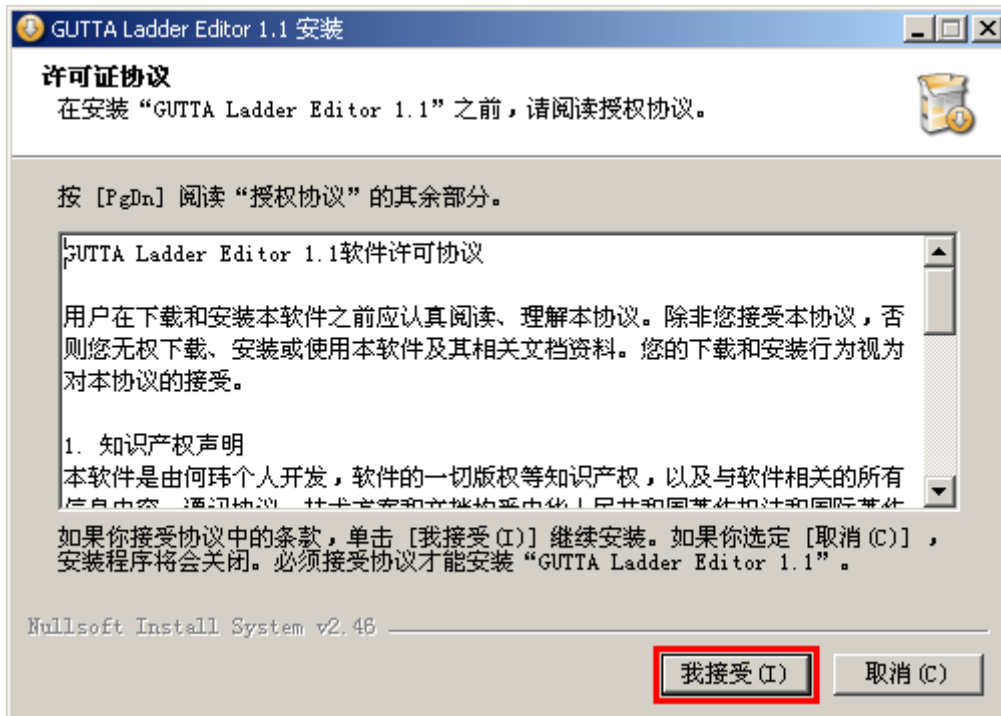


GUTTA Ladder
Editor 1.1.exe

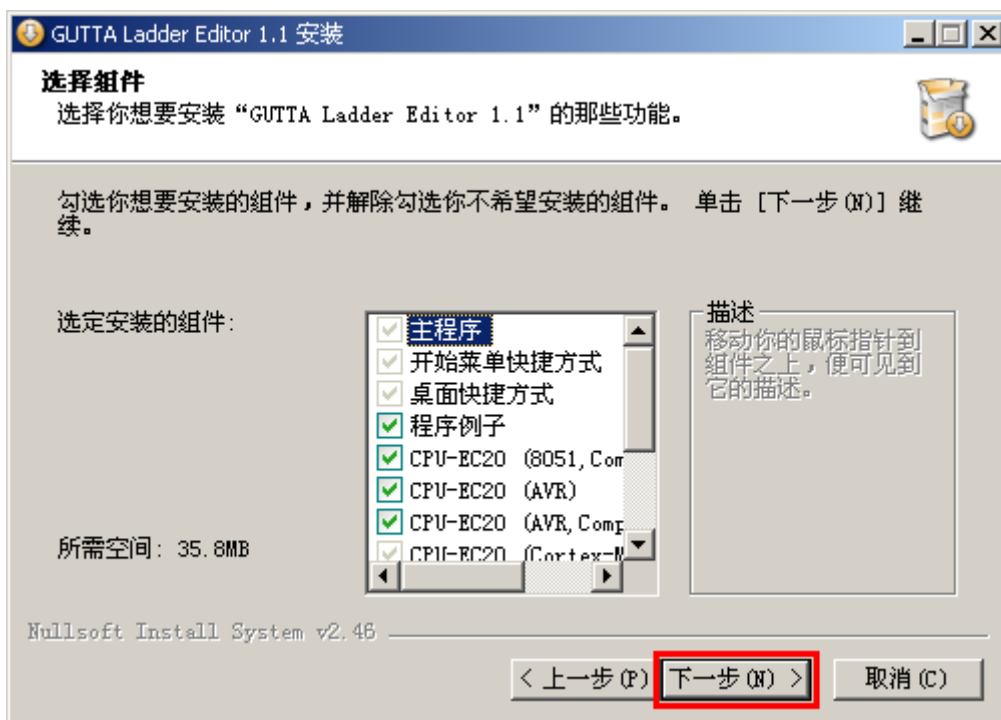
双击这个图标运行安装程序，出现语言选择对话框。
这里确定您需要安装的语言版本。



这里我们选择“Simplified Chinese”，代表简体中文，点击“OK”按钮继续。



这里需要阅读软件的许可协议，如果同意，请点击“我接受 (I)”按钮继续。

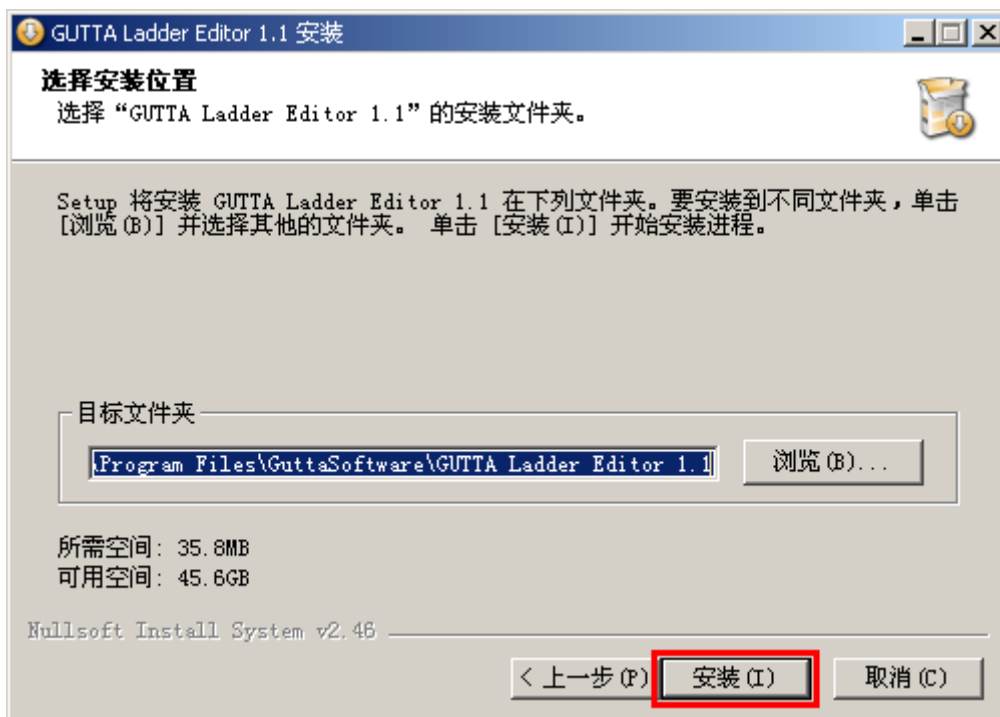


这里是 GUTTA Ladder Editor 软件的组件选择。

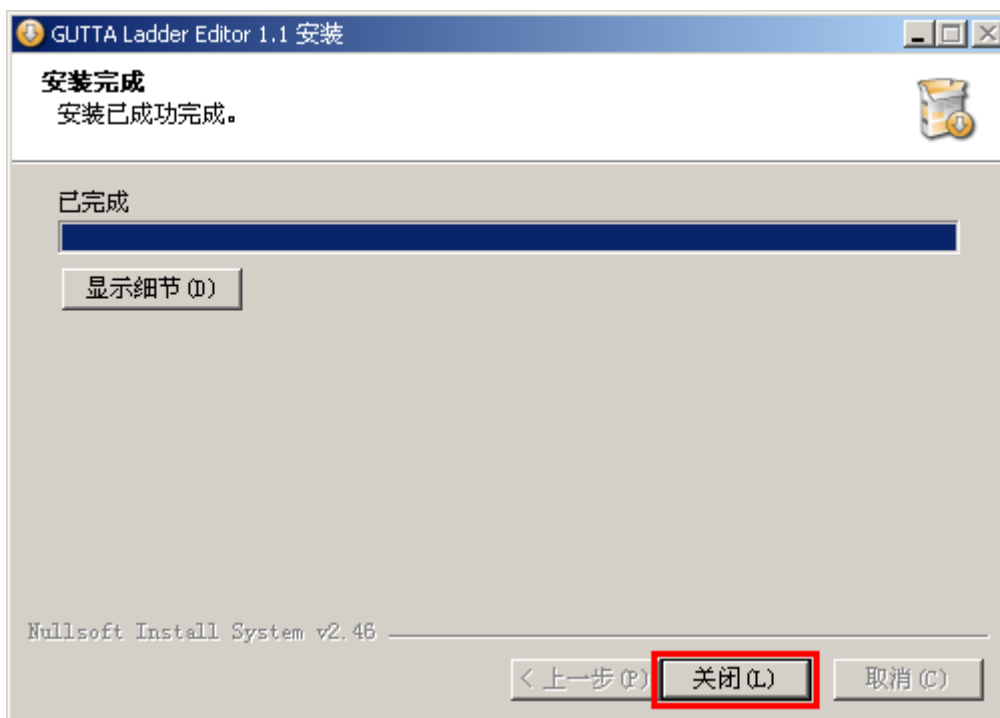
- 主程序
- 开始菜单快捷方式
- 桌面快捷方式
- CPU-EC20 (Cortex-M3)
- CPU-EC20 (Cortex-M3,Compile)

这 5 个组件是必选的。模拟器包含在 CPU-EC20 (Cortex-M3) 组件中。若您没有购买相

关硬件，其余 PLC 类型组件可以不安装。若您购买了基于 8051 的实验版，您应该选上 CPU-EC20 (8051,Compile) 组件，以用于支持 8051 的开发和调试。



这里修改软件的安装路径，确认后请点击“安装 (I)”按钮继续。



然后就是文件的解压过程，根据您安装组件的多少需要花费一定的时间。解压完成后，整个安装过程也就结束了。点击“关闭 (L)”按钮关闭安装程序。

您的第一次 PLC 之旅

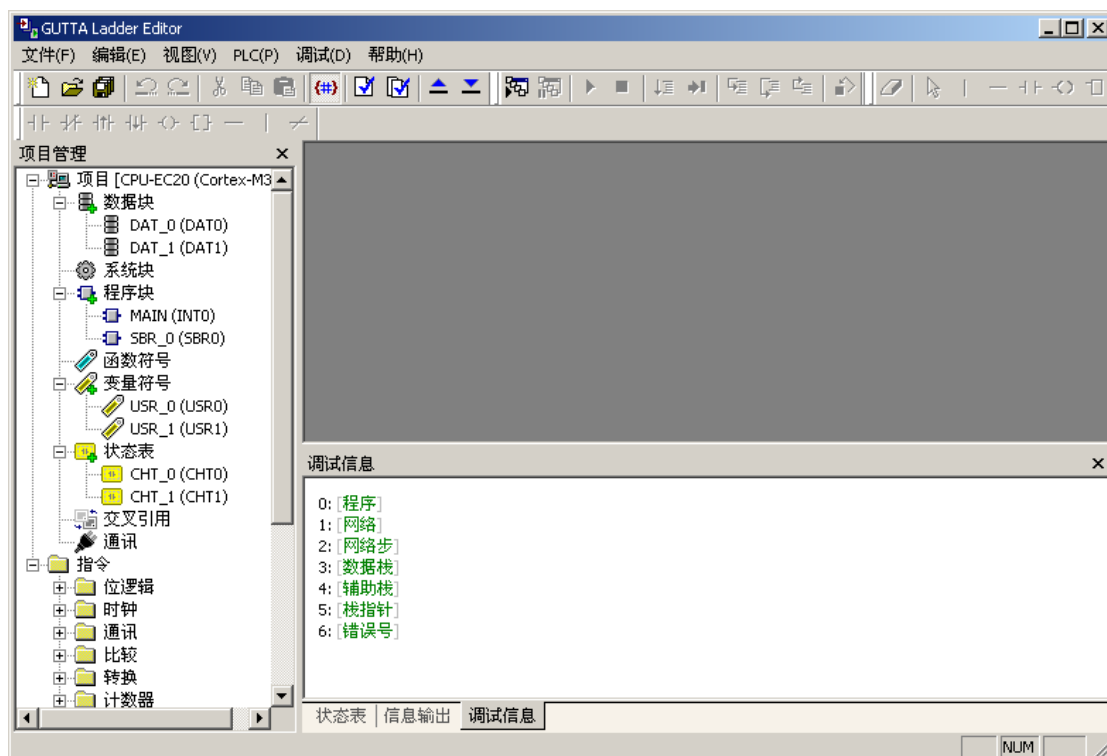
现在我们就开始动手吧！

还记得那个走廊灯两地控制的例子吗？我们就从这个例子开始。

首先，确认安装好了 GUTTA Ladder Editor 这个软件。双击这个软件的桌面图标就能运行软件 GUTTA Ladder Editor 了，软件的桌面图标看起来应该是这个样子的：



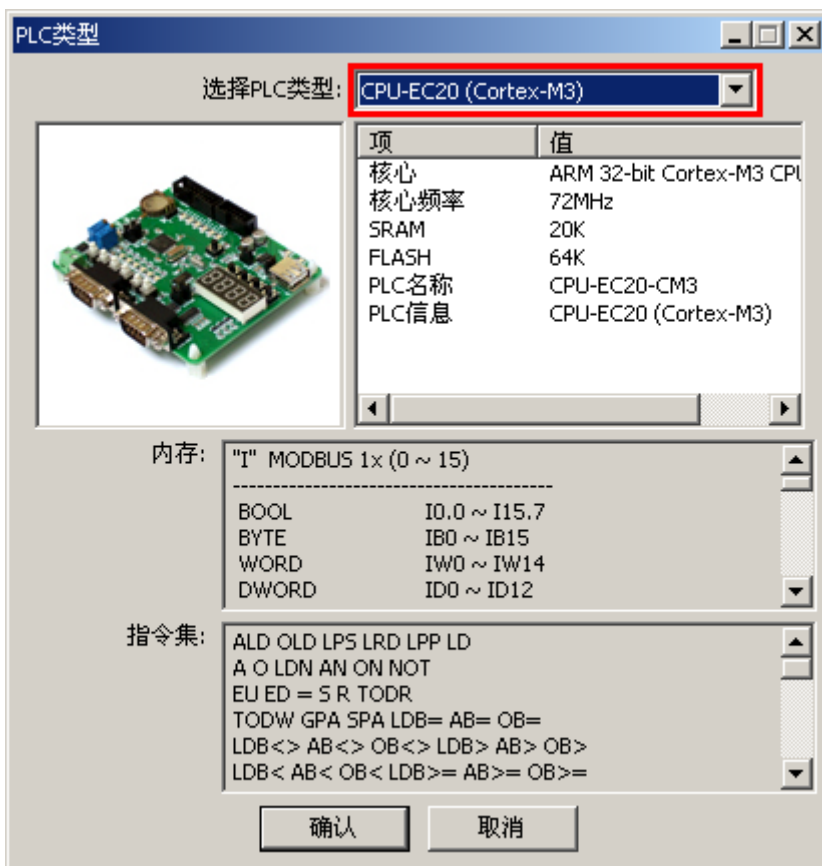
双击这个图标，如果一切顺利的话，就会出现下面的程序界面：



确认项目管理浮动窗口中的 PLC 类型为 CPU-EC20 (Cortex-M3)。如果不是需要修改当前的 PLC 类型。点击主菜单中 PLC 项的类型命令：



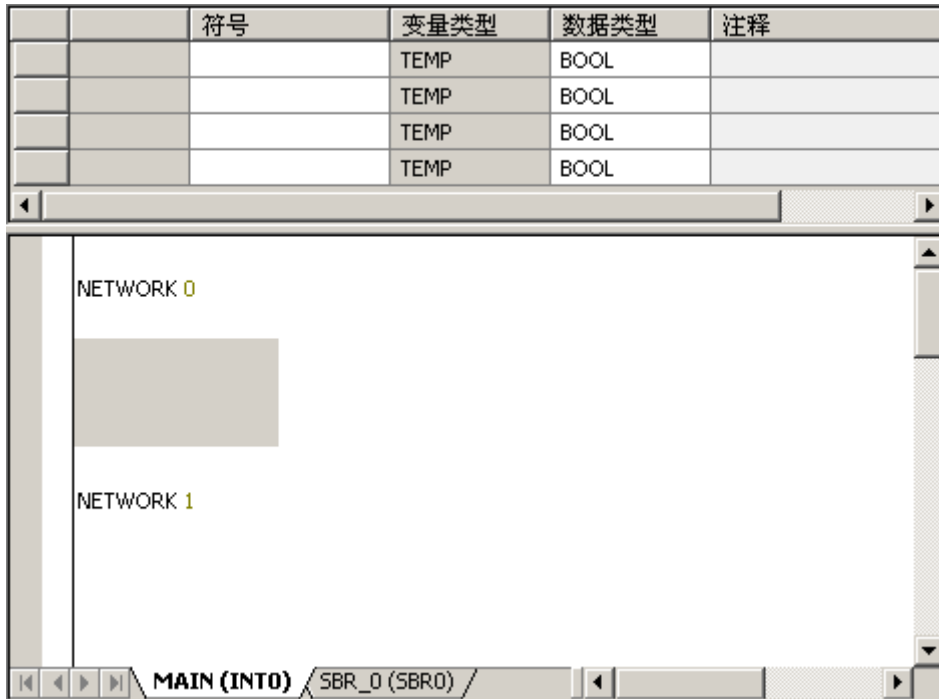
在 PLC 类型设置对话框中将 PLC 类型设置为 CPU-EC20 (Cortex-M3):



然后点击确定。确定好 PLC 类型后,我们就开始进行 PLC 梯形图的编写了。由于 GUTTA Ladder Editor 支持梯形图 (LAD) 和指令表 (STL) 两种编程方式,在编写程序之前先要确定编程方式。点击主菜单中视图项的 LAD 命令:



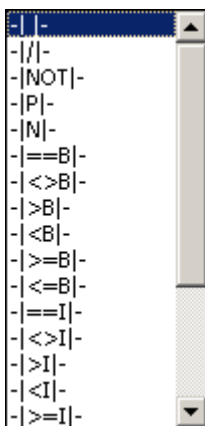
就进入了梯形图编程模式。同时软件也会弹出梯形图编辑窗口：



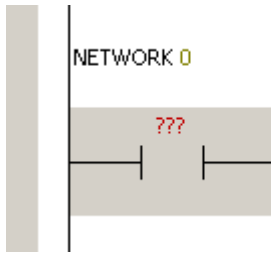
然后按下指令工具栏中的放置开关图标，进入放置开关模式：



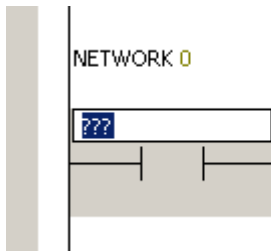
这个时候，鼠标就变成了放置开关的形状。将鼠标移动到第 1 个梯级的第 1 行、第 1 列，按下左键，就出现了开关类型选择列表：



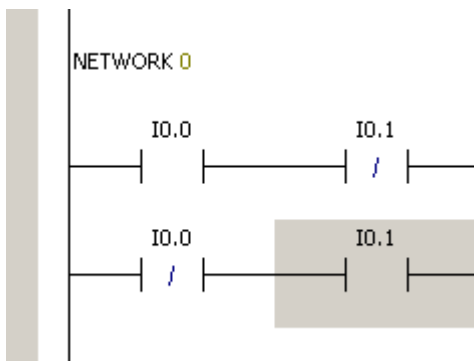
选择列表中的第一个条目，按下回车 (Enter) 键，那么就在第 1 个梯级的第 1 行、第 1 列放置了一个常开开关：



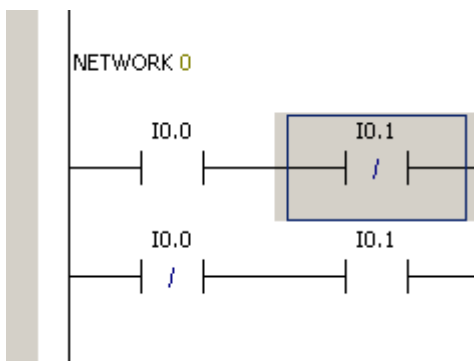
按下退出 (ESC) 键, 取消当前的放置开关模式。左键点击梯形图编辑窗口中的红色的问号字符, 开始编辑开关地址:



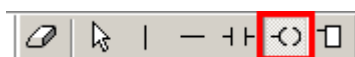
输入开关地址 **I0.0**, 按下回车 (Enter) 键后, 就完成了第一个开关的编辑。用同样的方法, 放置另外三个开关并编辑开关地址:



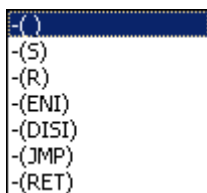
在第 1 行、第 2 列的开关的附近点击鼠标左键, 将焦点设置在这里。然后按下 **V** 键, 设置一条垂直线:



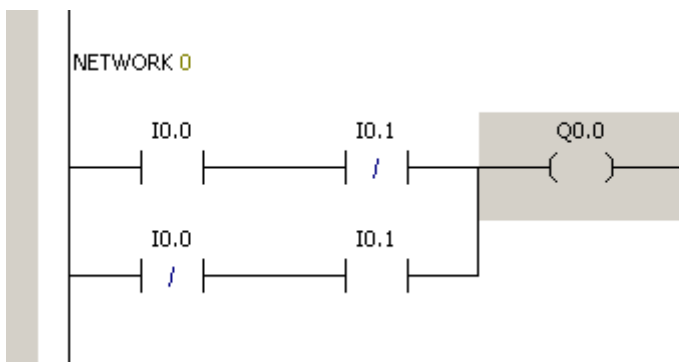
按下指令工具栏中的放置线圈图标, 进入放置线圈模式:



这个时候, 鼠标就变成了放置线圈的形状。将鼠标移动到第 1 行、第 3 列, 按下左键, 就出现了线圈类型选择列表:



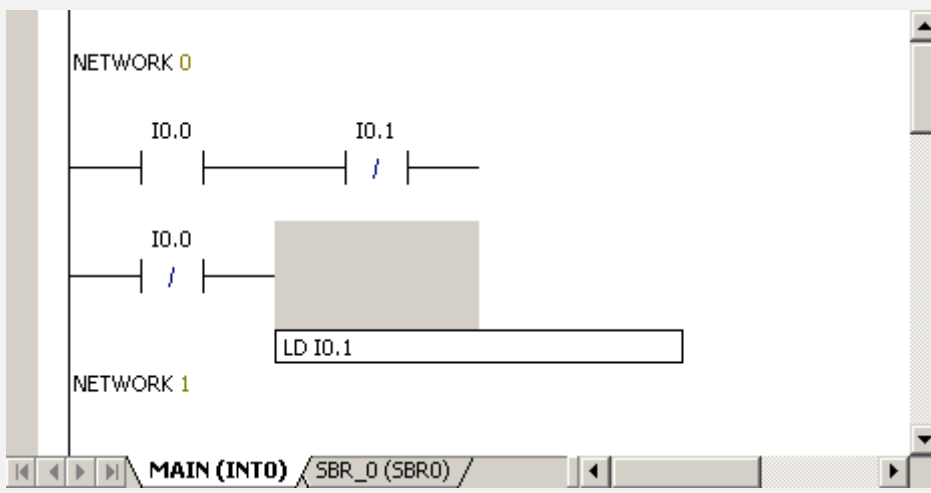
选择列表的第一个条目，按下回车（Enter）键，那么就在第 1 行、第 3 列放置了一个线圈。按下退出（ESC）键，取消当前放置线圈模式。点击线圈上的红色问号，编辑线圈地址为 Q0.0:



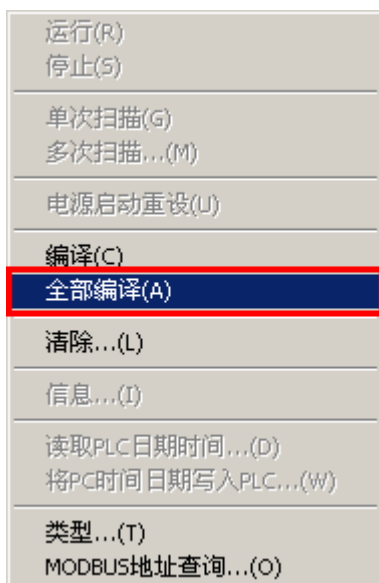
到这里，梯形图程序的编辑就完成了。

小提示

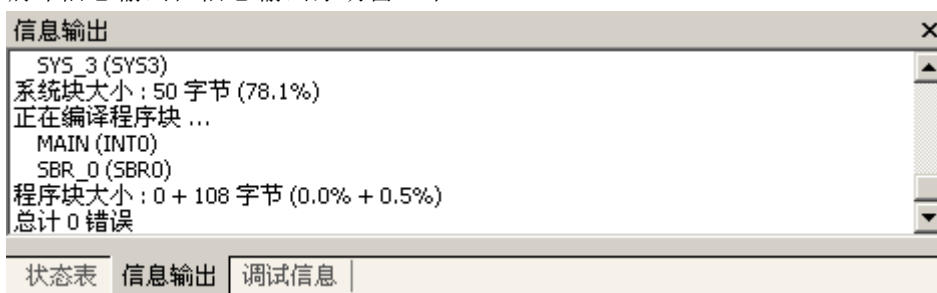
在梯形图模式下，也可以直接输入指令表指令并按回车（Enter），软件会在焦点位置放置指令表对应的梯形图元件。



点击主菜单中 PLC 项的全部编译命令:

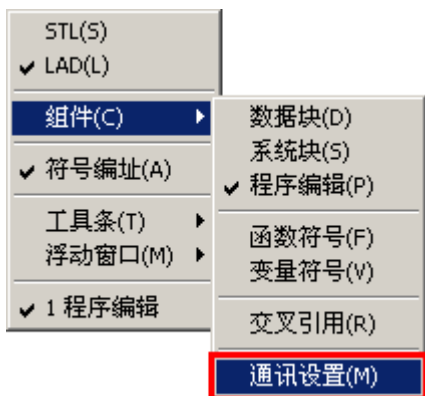


软件会先将梯形图转换成指令表，然后将指令表编译成 PLC 可以执行的二进制数据。并将编译信息输出在信息输出浮动窗口中：



信息输出最后提示总计 0 错误，表示梯形图编译成功。

点击主菜单中视图项的通讯设置命令：



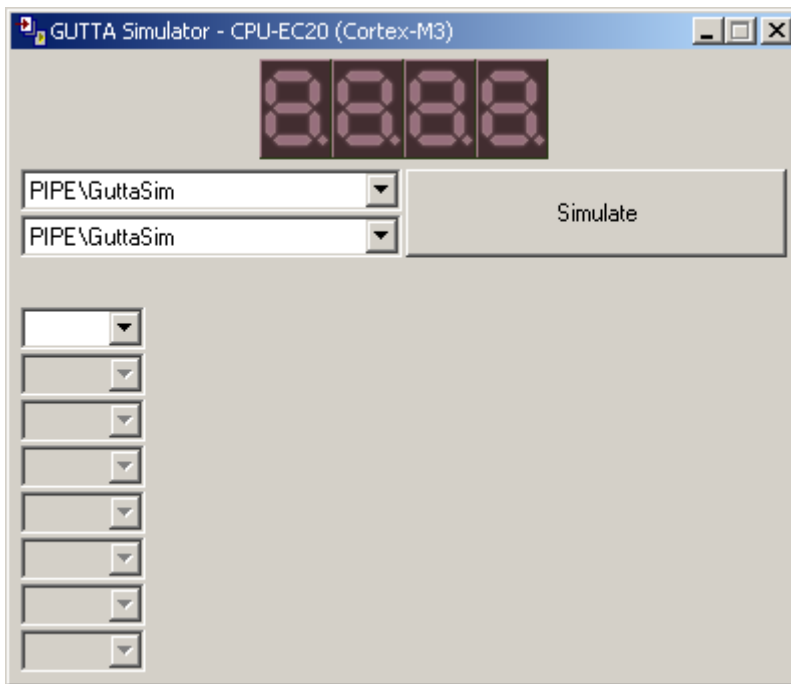
将站号设置为 0；端口使用 PIPE\GuttaSim；波特率为 38400bps；校验为 EVEN：



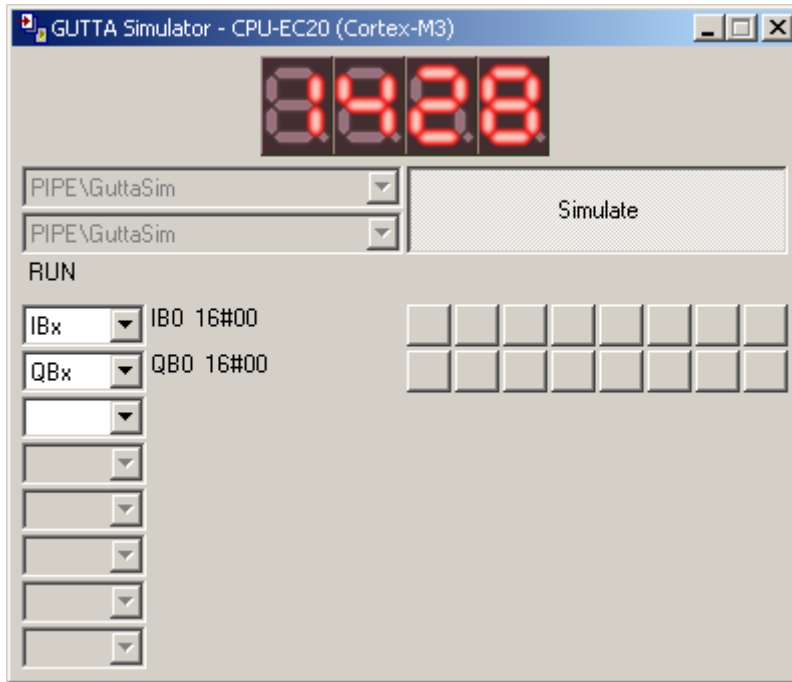
下面运行 GUTTA Simulator 这个软件进行 PLC 的软件模拟仿真。双击这个软件的桌面图标就能运行了，软件的桌面图标看起来应该是这个样子的：



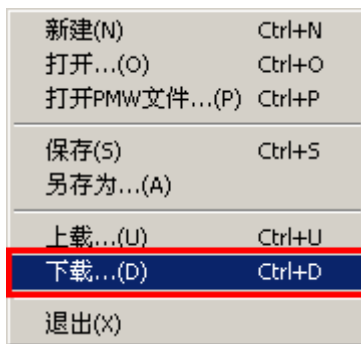
双击这个图标，如果一切顺利的话，就会出现下面的程序界面：



将两个端口都绑定为 PIPE\GuttaSim；把第一个 I/O 槽位配置为类型 IBx；把第二个 I/O 槽位配置为类型 QBx。这就相当于在 PLC 的 CPU 模块上挂接了一个离散量输入模块和一个离散量输出模块。然后点击 Simulate 按钮，让模拟器开始模拟运行：



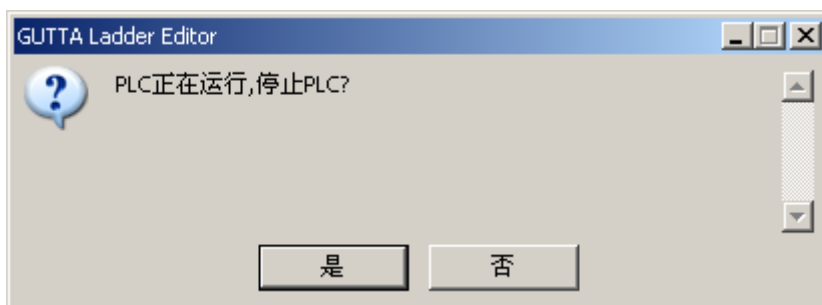
模拟器已经配置好了，回到编程软件 GUTTA Ladder Editor，点击主菜单中文件项的下载命令：



如果是第一次运行模拟器，默认是空密码。如果以前下载过程序，而且程序的系统块中也设置了密码，会弹出密码询问对话框，询问密码：



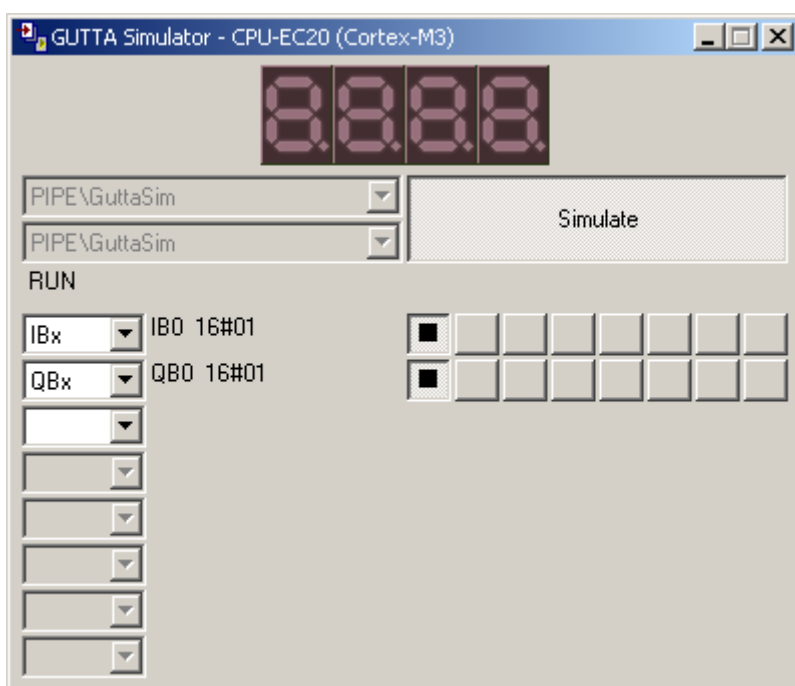
然后系统监测到模拟器正在运行，问是否停止 PLC? 这里直接点是：



于是软件开始利用 Windows 内部管道下载程序。由于程序非常简单，下载过程将非常短暂。下载完成后，点击主菜单中调试项的连线命令，进入在线状态：

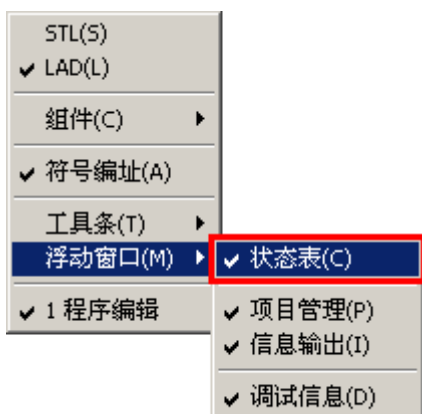


在在线状态，可以在 GUTTA Ladder Editor 中观察、编辑所有变量的值。不过这里，我们先通过模拟器来调试程序。切换到模拟器，按下 I0.0 对应的按键，观察 Q0.0 的变化：



多次试验就能发现，不论 I0.0 是否按下，再次按下 I0.1，Q0.0 的输出状态都会发生翻转。同样的，不论 I0.1 是否按下，再次按下 I0.0，Q0.0 的输出状态也都会发生翻转。假设 I0.0 对应走廊一头的控制开关，I0.1 对应走廊另一头的控制开关，而 Q0.0 就用于控制走廊灯的亮灭，是不是就实现了在走廊的两端都能对走廊灯的控制呢。

现在我们尝试在 GUTTA Ladder Editor 中调试程序，点击主菜单中视图项的状态表命令：



软件会将状态表窗口置顶：



选中第 1 行，第 1 列，输入地址 I0.0 - I0.1。回车确认后就在状态表窗口中加入了 2 个条目。用同样的方法，在第 3 行，第 1 列，输入 Q0.0。回车确认后就在状态表窗口中加入了第 3 个条目。选中 I0.0 这个条目的强制项；回车后，弹出牵制列表；选中列表中的强制，回车确定。用同样的方法，选中 I0.1 这个条目的强制项；回车后；弹出强制列表；选中列表中的强制，回车确定：



编辑 I0.0 和 I0.1 的值，观察 Q0.0 的值的变化的。
 祝贺您，到目前为止，您的第一次 PLC 之旅已经完成！

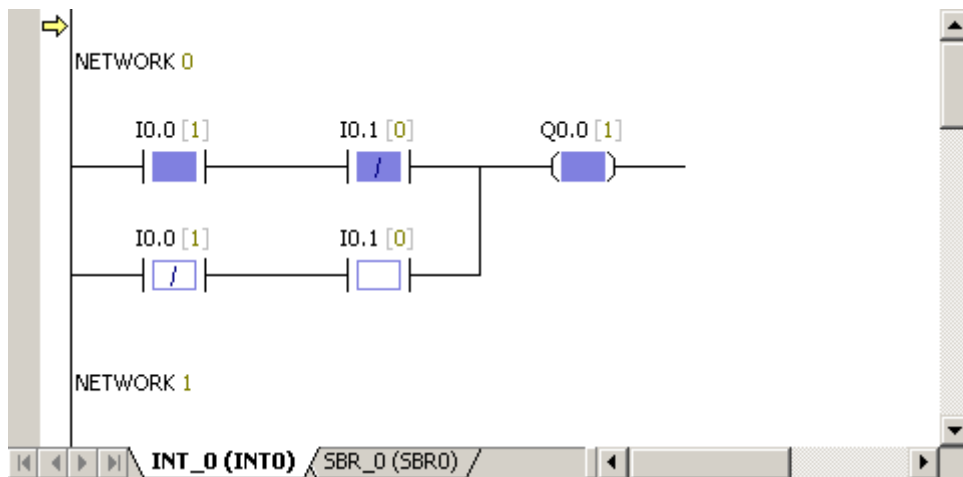
更加深入的调试

和普通的 PLC 编程软件相比，GUTTA Ladder Editor 具有更为强大的调试功能。通过以后的学习将会知道，PLC 在执行梯形图程序的时候实际上是一条指令一条指令执行的。这一点是与普通继电器控制电路的最大区别。也就是说，梯形图中各元件的执行是有先后顺序的。

这个特点一定要深入理解。由于指令顺序执行的原因：一方面，某些现有的继电器控制电路不能直接照搬到梯形图中使用，另一方面，我们也可以利用这个特点完成更多普通继电器控制电路不能够完成的功能。

目前绝大多数 PLC 编程软件及其配套 PLC 硬件只能进行运行中的状态调试。其调试手段就是在 PLC 连续运行中观察和修改各个变量的值。而这些变量值都是主循环扫描结束后的结果。PLC 程序的执行过程并不能观察。在 GUTTA Ladder Editor 软件中，除了普通的变量观察和修改功能，我们还能打断 PLC 指令的执行，一条指令一条指令的调试，并且观察每条指令执行后数据的具体变化。

继续使用上一小节走廊灯两地控制的例子，按照上一小节的内容，编写梯形图程序完成后，将程序下载到模拟器并进入在线模式，这个时候程序编辑窗口看起来应该是这个样子的：



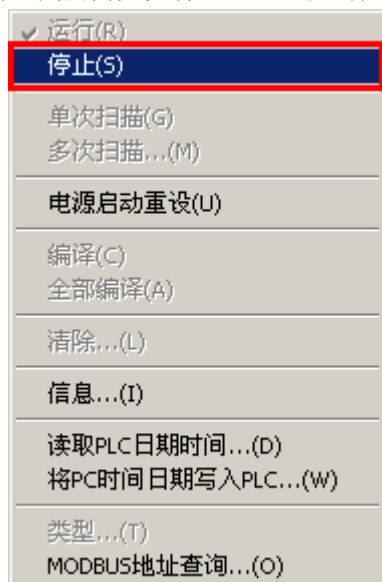
开关如果被蓝色填充，表示开关处于闭合状态，能流能够通过这个开关。线圈如果被蓝色填充，表示线圈处于得电状态，对应的 I/O 端口发生动作。上图表示在 I0.0 按下后，能流经过 I0.0 常开触点（第 1 行、第 1 列）、I0.1 常闭触点（第 1 行、第 2 列）到达线圈 Q0.0（第 1 行、第 3 列）。首先我们来看看 PLC 实际运行的指令是什么。点击主菜单中视图项的 STL 命令：



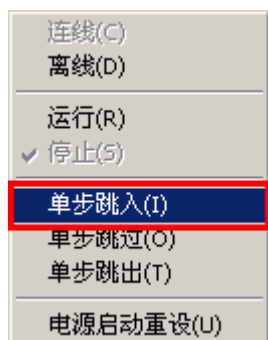
就可以切换到指令表编程模式：



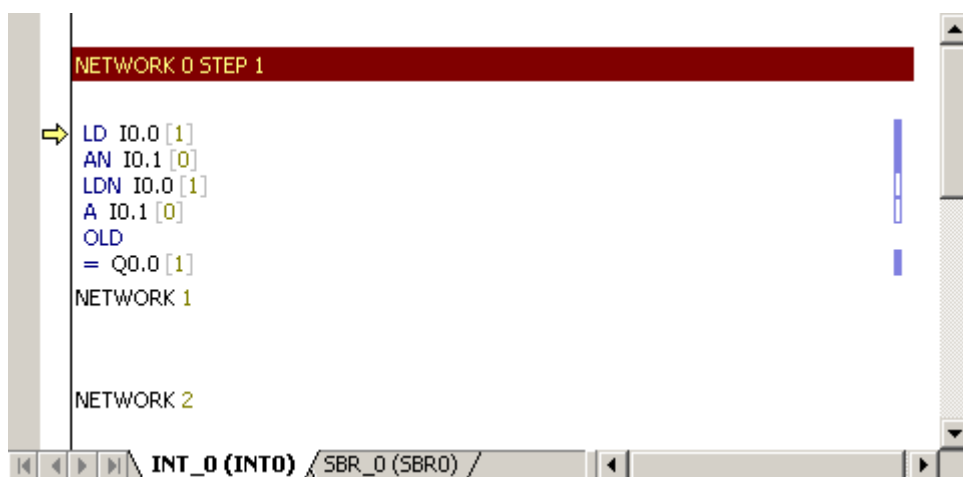
可以看出，软件将上面的梯形图程序翻译成了 6 条指令表指令。这些指令的具体含义和编写方法将会在以后进行系统的学习，目前只需要熟悉调试方法。为了能够进行逐条指令的调试，我们需要先停止 PLC 的运行。点击主菜单中 PLC 项的停止命令：



PLC 便进入了停止模式。这个时候我们就能发出指令，让 PLC 逐条执行指令了。点击主菜单中调试项的单步跳入命令：



这条命令能够触发 PLC 执行一条指令，然后立即停止：

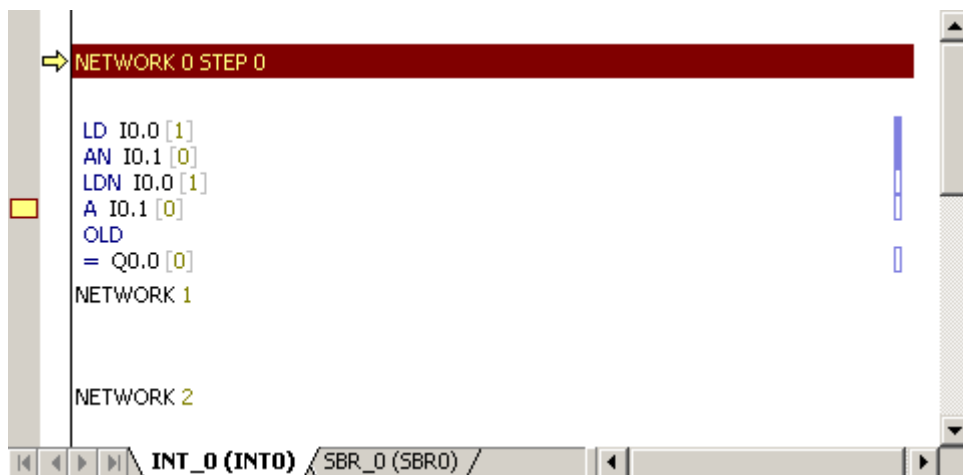


如上图，最左边黄色的小箭头发生了移动，表示第 1 条指令 NETWORK 已经执行完毕，下面将执行 LD IO.0 这条指令。继续点击主菜单中调试项的单步跳入命令，最左边黄色的小箭头将继续移动，直到程序的最末端：

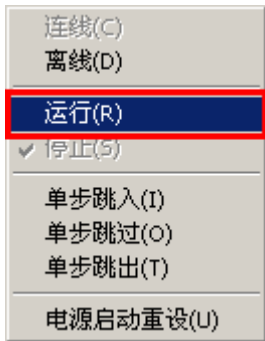


这个时候继续点击主菜单中调试项的单步跳入命令，PLC 将完成一次扫描，刷新 I/O 后将回到主循环程序的最开始。

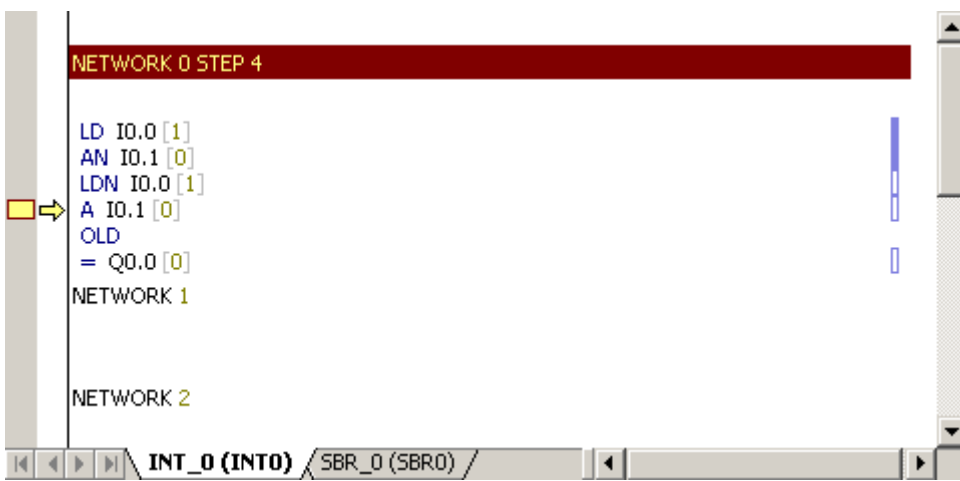
在程序很大的时候，往往我们只需要调试某一段程序。然而一条一条点击单步跳入命令使 PLC 逐条运行到我们需要调试的地方将是非常不方便的。这个时候我们可以通过设置断点的方法解决这个问题。在指令表编辑窗口最左侧的灰条上点击鼠标左键：



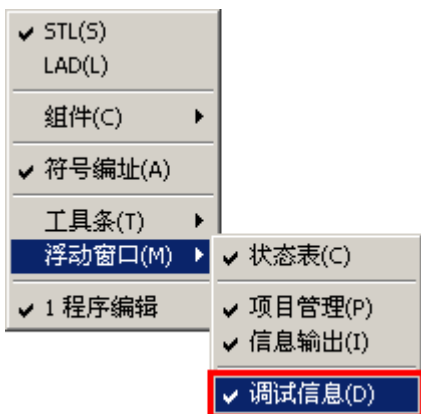
如上图,在鼠标点击的地方会出现一个黄色的小框,这个小框就是我们设置的断点位置。点击主菜单中调试项的运行命令:



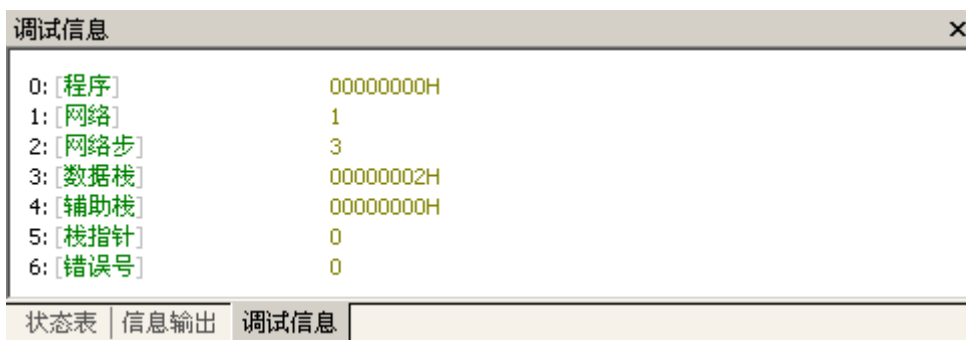
这条命令和主菜单中 PLC 项的运行命令不同, PLC 进入调试运行状态。在调试运行状态下, PLC 遇到断点将会自动停止:



如上图, PLC 停止在断点位置。此时可以继续点击主菜单中调试项的运行命令, PLC 跳过当前断点继续运行,一个扫描循环后又回到该断点并停止。也可以通过单步跳入命令开始断点处的单步调试。在单步调试过程中,可以观察到 PLC 更多的状态信息。点击主菜单中视图项的调试信息命令:



软件会将调试信息窗口置顶:



- 程序： 当前的函数代码；0 表示主循环扫描 INT_0 (INT0)。
- 网络： 当前的网络号；1 表示第 1 个梯级。
- 网络步： 当前网络中指令的步数；3 表示第 3 条指令。
- 数据栈： 当前数据栈的值。
- 辅助栈： 当前辅助栈的值。
- 栈指针： 当前函数嵌套的层数；主循环扫描是 0，即无嵌套。
- 错误号： 当前错误代码；0 表示没有错误。

主菜单介绍

这里先进行主界面的菜单命令的介绍。大家不必尝试将这些命令全部记住，大致浏览一遍就可以了。一是因为有些命令的使用频率不高，记住意义也不大；二是在没有真正用到这些命令之前，光看文字说明并不容易理解，记也是记了个字面。动手是第一位的，需要用再来查，才是正确的学习方法。

1. 文件项：






新建(N)	Ctrl+N
打开...(O)	Ctrl+O
打开PMW文件...(P)	Ctrl+P
保存(S)	Ctrl+S
另存为...(A)	
上载...(U)	Ctrl+U
下载...(D)	Ctrl+D
退出(X)	

名称	按键	图标	说明
新建	N		新建命令用于建立一个全新的工程文件。新建命令执行后当前工程的所有内容都会被新工程覆盖。如果当前工程未保存，软件会弹出对话框询问是否保存，或者取消当前操作。
打开...	O		打开命令用于打开以前保存的工程文件。打开命令执行后当前工程的所有内容都会被以前保存的工程覆盖。如果当前工程未保存，软件会弹出对话框询问是否保存，或者取消当前操作。

打开 PWM 文件...	P		打开 PWM 文件命令用于打开三菱的程序文件, 并尝试转换成 GUTTA Ladder Editor 自己的格式。使用打开 PWM 文件命令前, 需要确认当前的 PLC 类型是三菱风格。
保存	S		保存命令用于将当前工程保存到文件。对于已经保存过的工程, 保存命令将工程保存到上次保存的文件中。对于新工程, 保存命令会弹出一个对话框询问文件名。
另存为...			另存为命令用于将当前工程保存到文件。另存为命令总会弹出一个对话框询问文件名。
上载...	U		上载命令用于将工程从 PLC 上传到软件。上载命令执行后当前工程的所有内容都会被上传的工程覆盖。如果当前工程未保存, 软件会弹出对话框询问是否保存, 或者取消当前操作。
下载...	D		下载命令用于将工程从软件下载到 PLC。
退出	X		退出命令用于退出软件。

2. 编辑项:

撤销(U)	Ctrl+Z
重复(R)	Ctrl+Y
剪切(T)	Ctrl+X
复制(C)	Ctrl+C
粘贴(P)	Ctrl+V
全选(L)	Ctrl+A
查找替换(F)	Ctrl+F

名称	按键	图标	说明
撤销	U		撤销命令用于撤销最近的操作。
重复	R		重复命令用于恢复最近被撤销的操作。
剪切	T		将当前选择的内容剪切到粘贴板中。
复制	C		将当前选择的内容复制到粘贴板中。
粘贴	P		将粘贴板中的内容粘贴到当前位置。
全选	L		全部选择。
查找替换	F		弹出查找替换对话框, 对工程中的字符串进行查找或替换等操作。

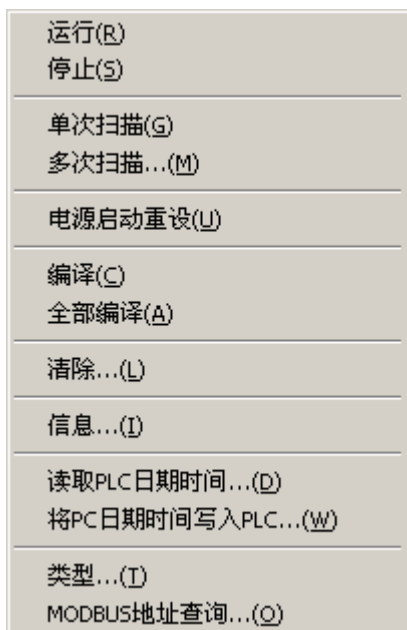
3. 视图项:

STL(S)	
✓ LAD(L)	
组件(C)	▶
✓ 符号编址(A)	
工具条(T)	▶
浮动窗口(M)	▶

名称	按键	图标	说明
STL	S		进入指令表 (STL) 编辑模式。如果当前处于梯形图 (LAD) 编辑模式。对于已经编辑好的程序, 软件会尝试编译当前梯形图并将梯形图转换成等价的指令表。转换信息会在信息输出窗口中给出。
LAD	L		进入梯形图 (LAD) 编辑模式。如果当前处于指令表 (STL) 编辑模式。对于已经编辑好的程序, 软件会尝试编译当前指令表并将指令表转换成等价的梯形图。转换信息会在信息输出窗口中给出。
组件	C		
组件 - 数据块	D		数据块命令弹出数据块窗口。在数据块窗口中, 设置 PLC 需要初始化的数据。
组件 - 系统块	S		系统块命令弹出系统块对话框。在系统块对话框中, 设置 PLC 需要用到的 PLC 系统配置数据。
组件 - 程序编辑	P		程序编辑命令弹出程序编辑窗口。在程序编辑窗口中, 可以编辑 PLC 需要执行的程序。如果处于指令表 (STL) 编辑模式, 可在此窗口中编辑指令表; 如果处于梯形图 (LAD) 编辑模式, 可在此窗口中编辑梯形图。
组件 - 函数符号表	F		函数符号表命令弹出函数符号表窗口。在函数符号表窗口中, 可以添加或删除中断函数和子程序函数, 并且为函数定义符号名。在定义了函数后, 才可以在程序编辑窗口中编辑相应的程序。
组件 - 变量符号表	V		变量符号表命令弹出变量符号表窗口。在变量符号表窗口中, 可以给绝对地址定义符号名。在软件的其它地方, 都可以通过使用符号名来引用实际的变量地址。
组件 - 交叉引用	R		交叉引用命令弹出交叉引用窗口。交叉引用窗口用于确定当前变量的使用情况。通过交叉引用, 可确认哪些变量被使用, 那些变量被空闲。
组件 - 通讯设置	M		通讯设置命令用于弹出通讯设置对话框。通讯设置对话框用于确定上载、下载、在线的通讯方式。如果是串口, 在这里还可以设定串口的波特率、校验位等。
符号编址	A		符号编址命令用于在符号编址模式和非符号编址模式间的切换。从非符号编址模式切换到符号编址模式时, 软件扫描所有的直接地址, 并且尝试将直接地址替换成对应的符号名。从符号编址模式切换到非符号编址模式时, 软件扫描所有的符号名, 并且尝试将符号名替换成对

			应的直接地址。
工具条	T		
工具条 - 标准	S		显示或隐藏标准相关的工具条。
工具条 - 调试	D		显示或隐藏调试相关的工具条。
工具条 - 指令	I		显示或隐藏指令相关的工具条。
工具条 - 指令 (Fx)	N		显示或隐藏指令相关的工具条。
浮动窗口	M		
浮动窗口 - 状态表	C		状态表命令用于弹出状态表浮动窗口。在状态表浮动窗口中，可以添加或删除需要监控的变量。在在线模式下，可以在状态表中读取或修改变量的值。
浮动窗口 - 项目管理	P		项目管理命令用于弹出项目管理浮动窗口。
浮动窗口 - 信息输出	I		信息输出命令用于弹出信息输出浮动窗口。
浮动窗口 - 调试信息	D		调试信息命令用于弹出调试信息浮动窗口。

4. PLC 项:




名称	按键	图标	说明
运行	R		运行命令用于在在线模式下对 PLC 发出运行指令，使 PLC 进入运行状态。
停止	S		停止命令用于在在线模式下对 PLC 发出停止指令，使 PLC 进入停止状态。
单次扫描	F		单次扫描命令用于在在线模式下对 PLC 发出扫描指令，使处于停止状态的 PLC 进行一次扫描。
多次扫描...	M		多次扫描命令用于在在线模式下对 PLC 发出扫描命令，使处于停止状态的 PLC 进行若干次扫描。扫描的次数由用户在对话框中确定。
电源启动重设	U		电源启动重设命令在在线模式下对 PLC 发出启

			动指令，强制 PLC 电源启动重设。
编译	C		编译命令用于编译当前程序块，编译结果在信息输出窗口中输出。
全部编译	A		全部编译命令用于编译当前数据块、系统块、程序块。编译结果在信息输出窗口中输出。
清除…	L		清除命令用于清除 PLC 中的程序。由于清除指令发送到 PLC 前不需要登陆 PLC，故清除指令能够清除有密码保护的程序，包括密码本身。
读取 PLC 日期时间…	D		在连线状态下，读取 PLC 日期时间。读取的时间用对话框显示出来。
将 PC 日期时间写入 PLC…	W		在连线状态下，将 PC 日期时间写入 PLC。写入的时间用对话框显示出来。
信息…	I		信息命令用于在在线模式下获取 PLC 的固件信息。
类型…	T		类型命令改变工程的 PLC 类型。当 PLC 类型被改变后，系统会根据新的 PLC 类型来新建工程。
MODBUS 地址查询…	O		弹出对话框，输入 PLC 变量，给出当前 PLC 类型下 PLC 变量对应的 MODBUS 变量地址。

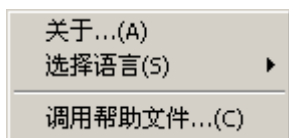
5. 调试项：



名称	按键	图标	说明
连线	C		连线命令将登入 PLC。若连接 PLC 成功，软件进入在线状态。
离线	D		离线命令将登出 PLC。若登出 PLC 成功，软件进入离线状态。
运行	R		运行命令用于在在线模式下对 PLC 发出运行指令，使 PLC 进入调试运行状态。
停止	S		停止命令用于在在线模式下对 PLC 发出停止指令，使 PLC 进入调试停止状态。
单步跳入	I		单步跳入命令使停止状态的 PLC 运行一条指令，若下一条指令是函数调用指令，则进入该函数后停止。
单步跳过	O		单步跳过命令使停止状态的 PLC 运行一条指令，若下一条指令是函数调用指令，则执行该函数后停止。
单步跳出	T		单步跳出命令使停止状态的 PLC 运行，直到停止位

			置的函数被返回。
电源启动重设	U		电源启动重设命令在在线模式下对 PLC 发出启动指令，强制 PLC 电源启动重设，重设后依然保持调试状态。

6. 帮助项:



名称	按键	图标	说明
关于...	A		弹出软件说明对话框。
选择语言	S		
选择语言 - 英语(美国)	E		切换语言到英文模式，切换语言需要重新启动软件。
选择语言 - 中文(中国)	C		切换语言到中文模式，切换语言需要重新启动软件。
调用帮助文件	C		调用帮助文件，帮助文件为 Html Help 格式。

快捷键一览

菜单命令	对应的快捷键
文件 - 新建	Ctrl + N
文件 - 打开...	Ctrl + O
文件 - 保存	Ctrl + S
文件 - 上载...	Ctrl + U
文件 - 下载...	Ctrl + D
编辑 - 撤销	Ctrl + Z
编辑 - 重复	Ctrl + Y
编辑 - 剪切	Ctrl + X
编辑 - 复制	Ctrl + C
编辑 - 粘贴	Ctrl + V
编辑 - 全选	Ctrl + A
编辑 - 查找替换	Ctrl + F

程序编辑介绍

下面是程序编辑介绍，函数部分我们暂时用不到，了解即可。重点是梯形图编辑器和指令表编辑器，一定要打开软件，自己实践。

程序编辑器是分页显示的，每个函数对应的程序位于不同的页面。程序编辑窗口的最下方就是页面选择表，点击每个函数名标签，就进入不同的函数编辑页面。每个函数编辑页面又分为上下两个部分。上面是函数的参数表格，下面是函数的具体实现。

参数表格用来定义当前函数的参数。在中断函数中，参数只有 TEMP 一种类型。在子程序函数中，参数除了 TEMP 类型，还有 IN、IN_OUT、OUT 三种类型，这三种类型定义了子程序函数被调用时参数的传递方向。对于每个函数，用户可以添加和删除参数（通过鼠标右键菜单中的删除 - 行、插入 - 行、插入 - 下一行来实现）。也可以通过编辑参数的符号和数据类型项来确认参数。一旦参数的符号名不为空，软件就会根据这个参数的位置，数据类型自动分配地址：

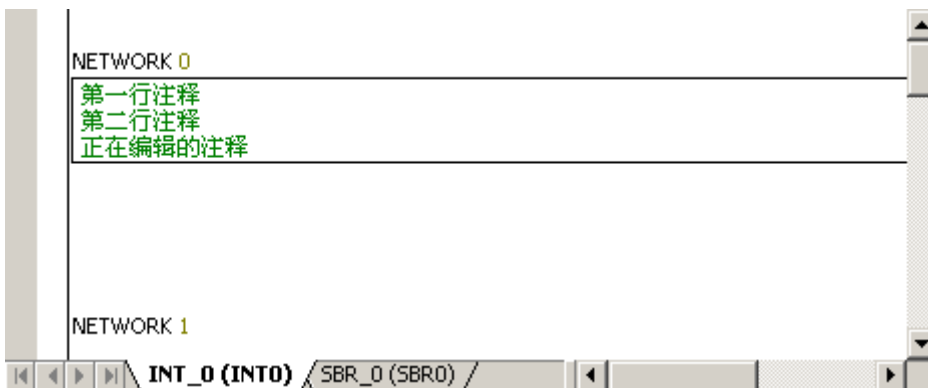
		符号	变量类型	数据类型	注释
✓	LBO	TBL1	TEMP	BYTE	
✓	LB1	TBL2	TEMP	BYTE	
			TEMP	BOOL	
			TEMP	BOOL	

在上图中，定义了两个输入参数，分配的地址分别是 LBO 和 LB1。

不论是在梯形图编辑模式下，还是在指令表编辑模式下，函数具体实现部分都是由一个个梯级组成的（NETWORK）。一个梯级有三个部分：梯级标题、梯级注释、梯级内容。

梯级标题是不可编辑的，由系统自动分配梯级号。梯级号从 0 开始。

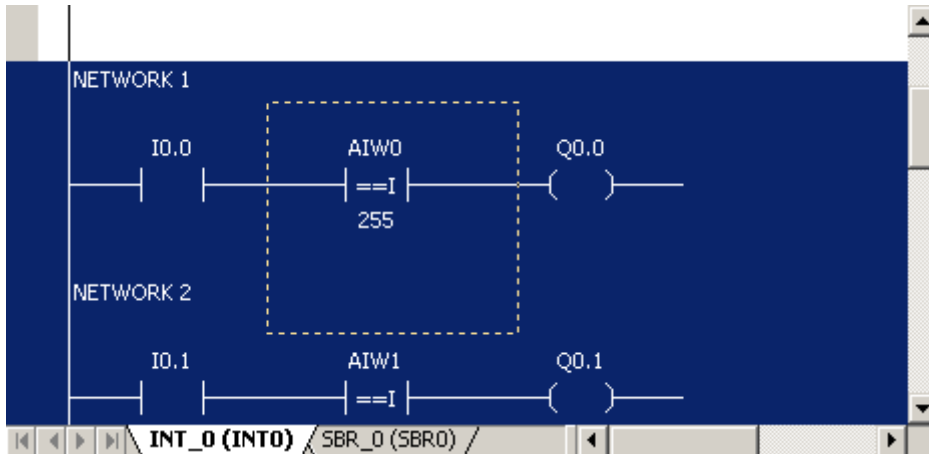
梯级注释是对本梯级的一个文字说明。可以通过直接点击梯级文字注释开始其编辑，也可以通过按键移动焦点到梯级注释。据有焦点的梯级注释自动进入编辑状态：



编辑好注释后，用向上或向下键能够继续切换焦点。也可以按取消（ESC）键取消注释的修改。若修改被取消，本次操作也不会被撤销重复记录。

梯级内容在不同的编辑模式下有不同的形式。在梯形图模式下，梯级内容由网格矩阵构成。在指令表模式下，梯级的内容为一个可以编辑的文字框构成。两种模式下的程序编辑在后面具体介绍。

可以通过单击梯级的标题以及梯级左右两边的空白来选择单个梯级。被选择的梯级被显示为蓝底白字。按下鼠标左键然后移动鼠标，就会出现一个选择框，当选择框在某个梯级内时，进行的是梯级内的选择。当选择框超出一个梯级的范围时，就进入了梯级选择模式。向下拖动鼠标，就能一次选择多个梯级：



在上图中，NETWORK 1 和 NETWORK 2 这两个梯级被选择。

如果需要进行大范围的梯级选择，也可以先将焦点设置在开始选择的梯级上。然后按下 Shift 键，在选择终止的梯级上设置新的焦点。那么在原焦点和新焦点之间的所有梯级都被选择。

在选择好了梯级之后，可以对选择的梯级进行各类操作。

可以单击指令工具栏中的擦除命令，将选择的梯级从网格中删除。

可以使用编辑菜单中的剪切、复制命令将被选择的梯级剪切、复制到公共粘贴板。然后使用粘贴命令将粘贴板中的梯级粘贴到新的位置。粘贴是可以跨软件的。您可以在两个以上正在运行的软件间复制粘贴它们。

梯形图编辑介绍

在梯形图编辑模式下，程序编辑窗口中只能进行梯形图编程。梯形图的编辑，本质上就是将各种元件（连接线、开关、线圈、功能块）放置到网格中去并进行适当的配置。梯形图被编译后，PLC 就能根据梯形图中的元件及其对应的位置来实现一定的逻辑控制。要编辑梯形图，首先要学会放置元件。

各种元件（连接线、开关、线圈、功能块）都有连续放置和独立放置两种模式。先看看连续放置元件，在指令工具栏中，有七个图标：



第 1 个图标对应的命令是擦除，这个命令在后面介绍。

第 2 个图标开始的后面 6 个图标对应的命令是选择、竖线、横线、开关、线圈、功能块。这 6 个命令其实也对应了 6 种编辑模式。在任何时候，都可以通过点击模式对应的命令在各个模式之间进行切换。同时也可以通过按下退出（ESC）键回到选择模式。

选择模式是最开始的默认模式。在选择模式下，点击鼠标不会放置任何元件到网格中去，这个模式下只能选择或编辑网格中已经存在的元件。

后面 5 种模式都是用于连续放置元件。

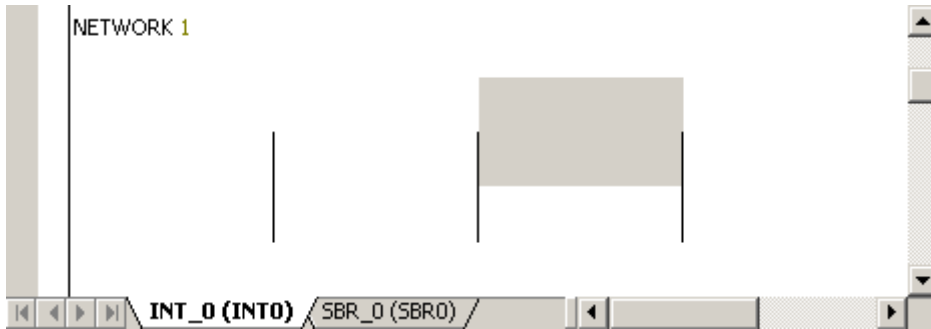
让我们点击竖线命令：



竖线命令的图标会处于下沉状态表示目前处于连续放置竖线模式。这个时候，鼠标应该看起来是这个样子的：



在这个状态下，只要是在梯形图的网格空白处按下鼠标左键，就会在鼠标对应网格空白处放置一条竖线。下面是连续放置了三条竖线的情景：



这个时候按下退出（ESC）键，就能退出放置竖线模式而回到选择模式。

横线命令的使用和竖线命令一样，请读者自行练习。

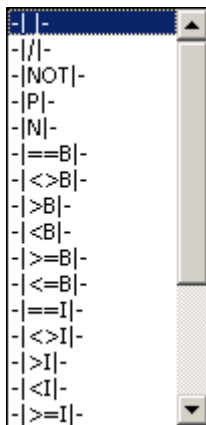
让我们点击开关命令：



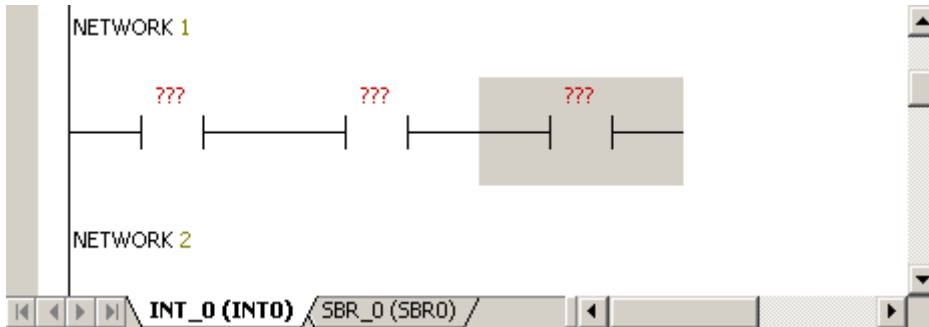
开关命令的图标就会处于下沉状态表示目前处于连续放置开关模式。这个时候，鼠标应该看起来是这个样子的：



在这个状态下，只要是在梯形图的网格空白处按下鼠标左键，就会在鼠标对应网格空白处弹出开关列表：



用鼠标或者向上键向下键来切换需要放置的开关类型。选中需要放置的开关之后，按下回车（Enter）键，就会在对应网络空白处放置 1 个开关。下面是连续放置了 3 个开关的情景：



这个时候按下退出（ESC）键，就能退出放置开关模式而回到选择模式。

线圈命令和功能块命令的使用和开关命令一样，请读者自行练习。

上面学习了元件的连续放置方法，下面介绍元件的独立放置。连续放置一旦进入某种放置模式，便能放置多个同类型的元件。但是，很多时候，我们只需要单独放置一个元件，这样如果还进入放置模式，放置元件后再退出，就显得不是那么方便了。这个时候，就应该使用独立放置方法。

独立放置必须配合键盘按键来进行：

- V: 焦点位置放置一条垂直线。
- H: 焦点位置放置一条水平线。
- F9: 焦点位置放置一个开关。
- F10: 焦点位置放置一个线圈。
- F11: 焦点位置放置一个功能块。

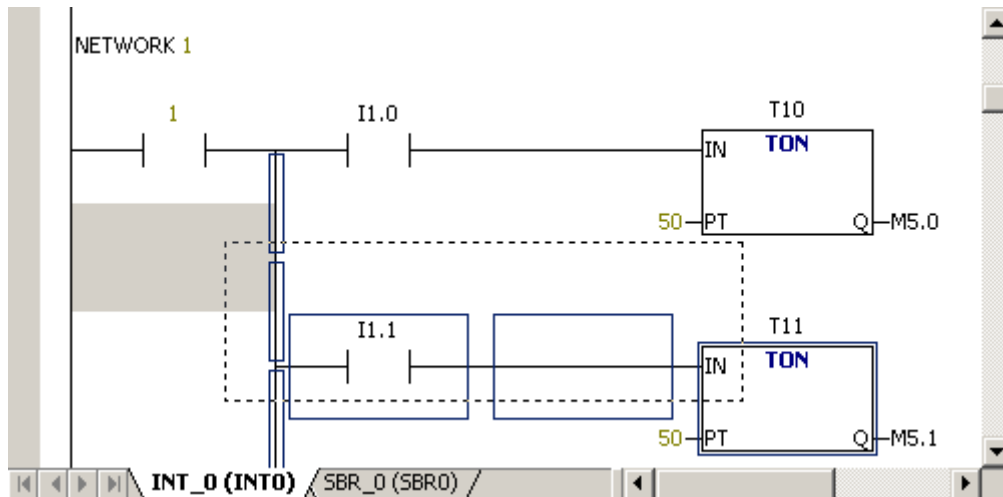
在选择模式下，按下上面列出的快捷键，软件都会尝试在当前焦点处放置对应的元件。

不论是连续放置还是独立放置，软件都会在放置前进行空间位置的判断，如果发现当前位置没有足够的空间放置指定的元件时，会尝试删除阻挡指定元件放置的其他元件。删除成功后再放置指定的元件。

介绍完了元件的放置，现在着重讲解如何对已经放置好了的元件进行配置和编辑操作。在选择模式下，点击元件的操作数，就能在操作数的位置弹出操作数编辑框。在编辑框中输入操作数后，按回车（Enter）键，操作数就被修改成新输入的值。

当焦点在梯形图网格中时，焦点用灰色的网格显示。并且可以通过向上、向下、向左、向右键来移动焦点。在移动焦点的时候，如果焦点网格中有元件，元件会被自动选择（被选择的元件被蓝色的边框包围）。在选择状态下按回车（Enter）键，编辑器会依次激活被选择元件操作数的编辑框。

在网格中，也能利用鼠标选择元件。选择状态下，在元件（包括竖线、横线）上方单击鼠标左键，就能选择单个元件。在选择状态下，按下鼠标左键，然后移动鼠标。在按下鼠标的位置到鼠标新移动的位置形成一个选择框，在选择框内的所有元件都会被选择：



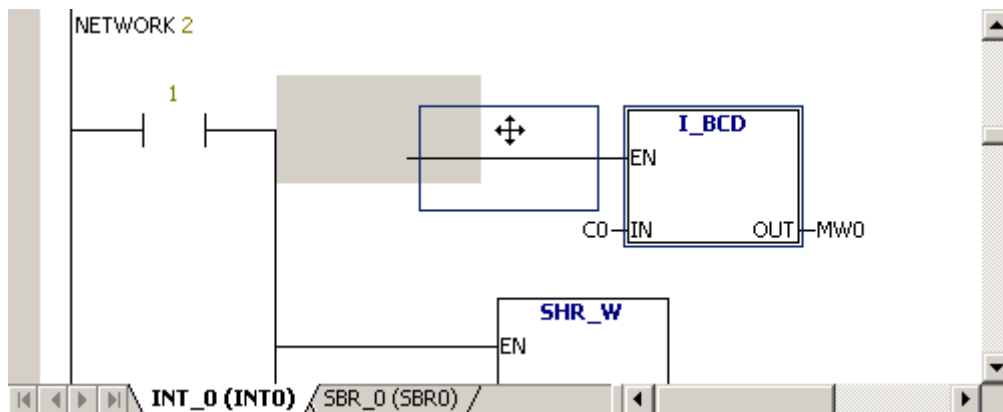
在上图中，被选择框选中的元件就有三条竖线、一条横线、一个开关和一个功能块。如果希望选择的元件不相邻，该怎么办呢，可以按下 **Ctrl** 键，然后逐个单击需要选择的元件。在 **Ctrl** 键按下后，点击元件不会取消已有的选择，而是将新的元件添加至选择列表中。

在选择好了元件之后，可以对选择的元件进行各类操作。

可以单击指令工具栏中的擦除命令，将选择的元件从网格中删除。

可以使用编辑菜单中的剪切、复制命令将被选择的元件剪切、复制到公共粘贴板。需要注意的是，在剪切、复制网格中的元件时，被操作的元件都是以焦点网格来计算相对位置的。在以后粘贴元件的时候，会根据新的焦点网格重新定位粘贴的元件。

在同一个网格内，可以对元件进行快速移动。方法是在选择好需要移动的网格之后，再次在已经选择好的元件上方按下鼠标左键（在任意蓝色选择框内）。然后移动鼠标，这个时候选择的元件就会跟随你的鼠标移动而移动了：



上面就是试图将一个横线和一个功能块向右移动一个网格的例子。在鼠标的移动过程中，软件会随时计算当前的移动位置是否有足够的空间放置移动中的元件。如果有，鼠标显示为一个移动符号；如果没有，鼠标显示为一个移动符号加一个禁止符号。在新的位置，松开鼠标左键，就结束了元件的移动。如果有足够的空间，那么元件就被移动。如果没有足够的空间，移动就会被取消。本次操作也不会被撤销重复记录。

指令表编辑介绍

在指令表编辑模式下，程序编辑窗口中只能进行指令表编程。在选择模式下，单击梯级，便能激活指令表编辑框：



指令表的编辑和一般的文本编辑没有任何区别。由于文本编辑更为灵活，一般说来以指令表的形式书写程序速度更快。

数据块编辑介绍

数据块用于设置变量的初始化值。

	地址	数据类型	值	注释
✓	M0.0	BOOL	1	
✓	M0.1	BOOL	1	
✓	MW20	WORD	16#0100	
✓	MW22	INT	2000	
		BOOL		

设置一个初始化值，需要确定数据地址、数据类型和数据的初始化值。在 EC30/EC30 中，数据块可以定义至少 16 个初始化数据页面；每个数据页面可以添加 16 个初始化值。PLC 在每次上电运行前，都会根据数据块给指定变量进行初始化。初始化值在 PLC 上电运行前只赋一次，被赋予初始化值的变量可以在程序中改变。

系统块编辑介绍

系统块用于设置 PLC 系统配置数据：



在 CPU-EC20 (Cortex-M3) 中系统块分为四个部分：

- 通讯端口页面：用于配置 PLC 的通讯端口。
- 密码页面：用于配置 PLC 的登陆密码。
- 中断时间页面：用于配置 PLC 定时中断的时间间隔。
- 强制表页面：用于配置部分 I/O 端口的强制状态。强制表页面的强制状态和状态表的强制状态是等效的。不过状态表的强制状态修改后不会被记忆，是调试时临时采用的；而强制表页面的强制状态是写入 FLASH 的，每次启动运行都会生效。

函数符号表编辑介绍

在函数符号表窗口中，可以添加或删除中断函数和子程序函数，并且为函数定义符号名。在定义了函数后，才可以在程序编辑窗口中编辑相应的程序。

	符号	地址	注释
✓	MAIN	INT0	主循环扫描
✓	SBR_0	SBR0	
✓	MyFun	SBR1	自定义函数

变量符号表编辑介绍

在变量符号表窗口中，可以给绝对地址定义符号名。在软件的其它地方，都可以通过引用符号名来引用实际的变量地址。

	符号	地址	数据类型	注释
✓	Pump_1	Q0.0	BOOL	1号泵运行
✓	Pump_2	Q0.1	BOOL	2号泵运行
			BOOL	
			BOOL	

USR_0 (USR0) / USR_1 (USR1)

交叉引用表示用介绍

交叉引用表用于统计当前变量的内存使用情况：

地址	符号	位置	关联
I0.0		MAIN (INT0) NETWORK 0 Col 1 Row 2	- -
Q0.0		MAIN (INT0) NETWORK 3 Col 2 Row 0	-()
Q0.1		MAIN (INT0) NETWORK 3 Col 2 Row 1	-()
Q0.2		MAIN (INT0) NETWORK 3 Col 2 Row 2	-()
Q0.3		MAIN (INT0) NETWORK 3 Col 2 Row 3	-()
Q0.4		MAIN (INT0) NETWORK 3 Col 2 Row 4	-()
Q0.5		MAIN (INT0) NETWORK 3 Col 2 Row 5	-()

VALUE / BIT / BYTE

交叉引用表有三个子页，分别是交叉引用、位引用、字节引用。交叉引用表中点击右键能够弹出菜单。菜单有两个命令，分别是刷新和定位。刷新用于重新统计当前变量和内存的使用情况；定位用于确定使用这个变量的位置。直接双击某个变量条目也可用于定位。位引用和字节引用用于反映当前的内存使用情况。

表格中的

字符 **b** 表示本处内存以位的形式使用；

字符 **B** 表示本处内存以字节的形式使用；

字符 **W** 表示本处内存以字的形式使用；

字符 **D** 表示本处内存以双字的形式使用。

查找替换对话框使用介绍

在梯形图和指令表的程序编辑窗口中，按下 **Ctrl+F** 就能弹出查找替换对话框：



查找替换对话框以字符的形式在所有程序中查找匹配的操作数。并将结果以列表的形式显示在对话框中。可以通过下一个按钮定位下一个操作数被使用的位置。替换按钮将当前位置的操作数替换成指定的字符串。全部替换按钮将所有找到的匹配操作数全部替换成指定的字符串。

总结

这里介绍了 GUTTA Ladder Editor 软件的方方面面。很多内容都只是一个初略的介绍，大家浏览一下，有个大致印象就可以了。很多具体的功能只有在后面真正使用时，才能深入理解。