

---

# IN1002 GUTTA 指令说明

---

**CREATE: 2008/12/25**

**UPDATE: 2010/12/04**

**Version 1.1**

**GUTTA Ladder Editor Version 1.1**

<http://www.plcol.com>

<http://www.visiblecontrol.com>

---

指令的分类.....	2
指令的编辑.....	3
指令列表.....	4

## 指令的分类

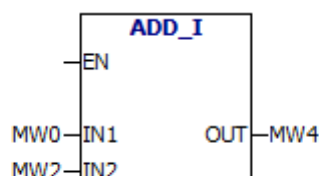
GUTTA PLC 支持梯形图 (LAD) 和指令表 (STL) 两种编程模式，同时支持这两种模式的相互转换。因此 GUTTA PLC 有两类指令，一类是梯形图指令，一类是指令表指令。梯形图指令实际上就是梯形图中的元件。PLC 固件不能够直接执行梯形图指令。梯形图指令必须经过 GUTTA Ladder Editor 软件转换成指令表指令后才能下载到 PLC 中执行 (不论解释型 PLC 还是编译型 PLC)。因此指令表指令是 PLC 执行程序的基本单元。

绝大部分梯形图指令和指令表指令是一一对应的关系，但也有指令例外。

第一种情况是系统指令表指令，它们是：ALD、OLD、LPS、LRD、LPP。ALD 表示串联连接、OLD 表示并联连接、LPS、LRD、LPP 用来表示复合输出连接。因为连接方式在梯形图中不需要用元件表达，故这五条指令没有梯形图形式。

还有一种情况就是指令的复合与分解。

例如梯形图中的加法指令：

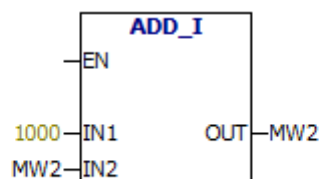


转换成指令表指令时，被分解成：

```
MOVW    MW0, MW4
+I      MW2, MW4
```

梯形图指令表示 MW4 等于 MW0 加上 MW2，这个指令有 3 个操作数。而对应的指令表指令为两步，首先 MW4 等于 MW0，然后 MW4 加上 MW2 后结果存入 MW4。这两条指令都只有 2 个操作数。由于这两条指令是连续执行的且都为输出指令，执行过程中数据栈站顶的值不会发生变化，故要么就都执行，要么就都不执行，最终数据结果和梯形图中的功能块一致。

类似 ADD\_I 这样的指令，IN2 操作数不能和 OUT 变量使用同一个变量，例如：



转换成指令表指令时，被分解成：

```
MOVW    1000, MW4
+I      MW4, MW4
```

梯形图指令的操作是  $MW2 = MW2 + 1000$ ，但是指令表的实际操作是  $MW2 = 2000$ 。在 GUTTA Ladder Editor 1.0 中，目前还不能发现这类错误。需要在编辑梯形图的时候特别留意。

指令的复合与分解还有一种形式，就是输入指令和系统指令的复合，在后面详细解释。

## 指令的编辑

梯形图编辑窗口中，可以放置三类元件（不包括横线和竖线），他们是：开关、线圈、功能块。开关有一个输入能流端和一个输出能流端。开关有两种状态，导通或者断开。开关导通则能量可以流过，开关断开则能量不能流过。线圈则只有一个能流输入端。线圈根据其输入能流端的状态确定对应的操作。功能块一般具有一个能流输入端，个别时候也有多个（例如向上计数器指令 CTUD）。虽然 GUTTA Ladder Editor 软件支持具有输出能流的功能块，但是在 EC20/EC30 系统中，没有实现这种功能块指令。

开关，线圈，功能块只是外部形状而已。GUTTA Ladder Editor 在进行梯形图到语句表的转换时，会忽略形状信息。在转换系统中，只认为有下面几类指令：

- 输出指令：输出指令表示只有一个能流输入端，没有能流输出端。输出指令可能是线圈，也可能是功能块。
- 输入指令：输入指令表示只有一个能流输入端，只有一个能流输出端。输入指令可以是开关，也可能是功能块。
- 复合输出指令：复合输出指令表示有一个以上能流输入端，没有能流输出端。复合输出指令只能是功能块。
- 复合输入指令：复合输入指令表示有一个以上能流输入端，只有一个能流输出端。复合输入指令只能是功能块。

在添加自定义指令的时候，需要特别注意：GUTTA Ladder Editor 软件只能保证输出指令的前端能流为正确的值。因此所有与能流对应的操作只能在输出指令中完成。输入指令的所有操作数必须是输入操作数，指令不可修改操作数的内容，亦不可根据输入能流执行 PLC 动作，**因为输入指令前端能流的值并不能正确地反映梯形图中的能流的逻辑。**

在 GUTTA Ladder Editor 软件中，有 3 种指令不符合上述规则。第 1 个是能流取反开关-|NOT|-。这条指令不能理解为普通开关，它在输入能流为 1 时输出 0，在输入能流为 0 时输出 1。第 2 是正跳变开关-|P|-。第 3 个是负跳变开关-|N|-。正跳变开关和负跳变开关都具有一个输出操作数，用于记录上一次扫描时，输入能流的值。这三条指令在 GUTTA Ladder Editor 的指令转换系统中做了特殊的检查，因此可以避免上述问题。自己定义的指令，由于 GUTTA Ladder Editor 指令转换系统不能区别对待，所以必须符合上述对输入指令的限制。

在指令表编辑模式，所有的指令都以文本的形式进行输入。借助于文本编辑器的一些功能，指令表编程模式显得更为灵活。指令表指令没有能流的概念，与之对应的是 PLC 位栈的操作。PLC 有两个位栈，一个是数据栈，一个是辅助栈。数据栈用于当前的串并联逻辑运算，辅助栈用于复合输出时的能流保存。

在指令表中：

- 输出指令：不改变数据栈的值，只能根据数据栈栈顶的值进行对应的操作。
- 输入指令：可将一个值压入数据栈，压入的值由输入指令的输入操作数决定。
- 复合输出指令：不改变数据栈的值，可根据数据栈栈顶的几个值进行对应的操作。
- 复合输入指令：可取出数据栈栈顶的几个值（数量为输入能流数量），并将一个值压入数据栈，压入的值由复合输入指令的输入操作数决定。

输入指令可以和系统指令 OLD、ALD 进行复合。例如：

```
LD    M0.0
OLD
```

可以复合成一条指令：

```
O    M0.0
```

这条指令可以看作上面两条指令的组合，实际上 PLC 在执行时作了一些优化。例如原本的操作为：

<b>LD</b>	M0.0	① 将 M0.0 的值压入数据栈
<b>OLD</b>		② 将数据栈栈顶的两个值取出进行或运算
		③ 或运算结果压入数据栈

变成了操作：

<b>O</b>	M0.0	① 若 M0.0 为 1，置位栈顶的值
----------	------	---------------------

而后一种操作大大简化了 PLC 的运算，提高了效率。

## 指令列表