

脚本编程与梯形图编程相融合

## 高效的 PLC 编程技术

KV 脚本活用指南

介绍了能够显著减少 PLC 编程/调试工时的最新技术和具体应用举例。



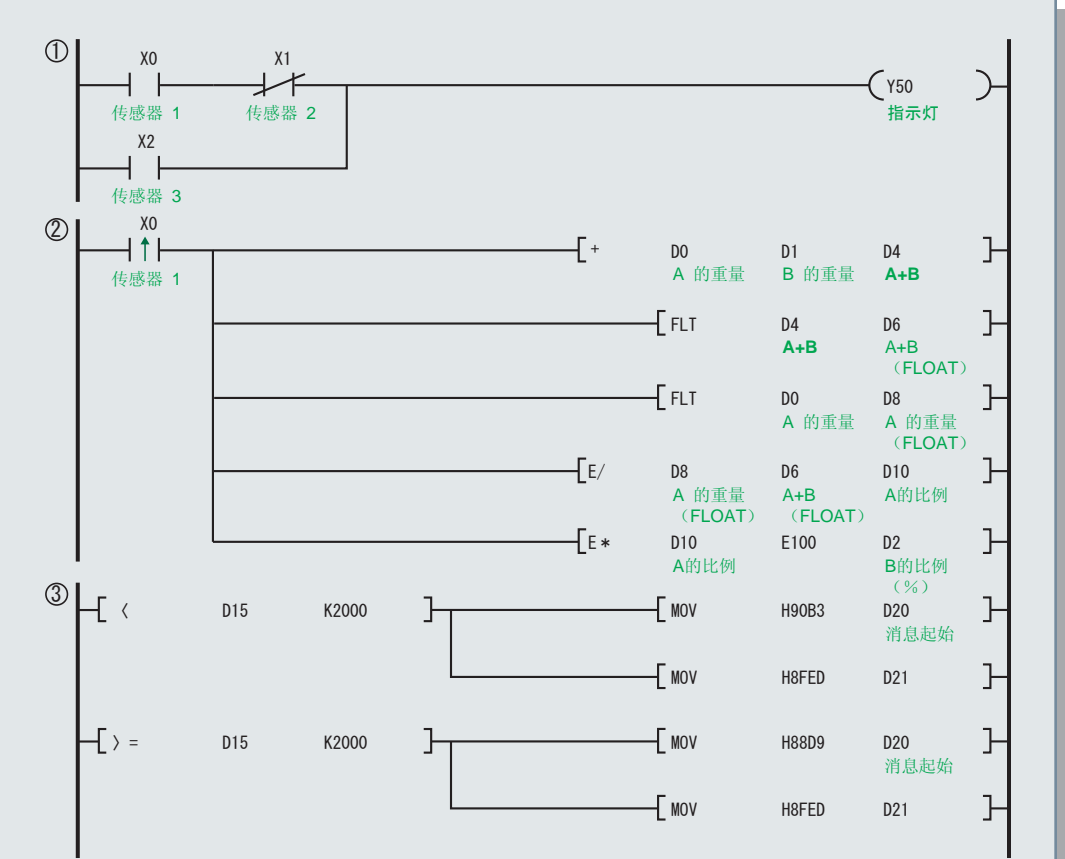
### 目录

梯形图与脚本的融合 .....	2
KV 脚本的编程方法 .....	4
丰富的增强功能 .....	5
KV 脚本活用实例①生产管理 — 四则/浮点数运算 .....	6
KV 脚本活用实例②角度计算 — 三角函数 .....	8
KV 脚本活用实例③生产排程 — 条件分支 .....	10
KV 脚本活用实例④食品有效期打印 — 字符串连接 .....	12
KV 脚本活用实例⑤加工次数与目标坐标 — 循环处理 .....	14

比如，某装置的部分技术规格如下所示。

- ① 传感器 1 (X0) ON，且传感器 2 (X1) OFF，或传感器 3 (X2) ON，  
则指示灯 (Y50) ON。
- ② 在传感器 1 (X0) 的上升沿，计算 A 的重量 (D0) 与 B 的重量 (D1) 之间的比例 (D2)。
- ③ 若测定值 (D15) 小于 2000，则在消息(D20~) 中存储 “正常”；若大于 2000，  
则在消息 (D20~) 中存储 “异常”。

传统的梯形图编程：



- 位处理 — 容易直观理解处理内容 (①)
- ✕ 字处理 — 行数较多 (②)  
容易直观理解表达式及字符串 (②③)  
需考虑运算的优先顺序 (②)  
需存储运算的整个过程 (②)

梯形图的优势

如果只使用脚本进行编程：

```
①
IF (X0 AND NOT X1) OR X2 THEN
  Y50 = ON
ELSE
  Y50 = OFF
END IF

②
IF LDP (X0) THEN
  D2.F = TOF (D0) / (D0 + D1) * 100
END IF

③
IF D15 < 2000 THEN
  D20.T = "正常"
ELSE
  D20.T = "异常"
END IF
```

什么是脚本语言？  
脚本语言是一种用于简单创建小型程序的语言，与普遍的编程语言相比，能够更简单地进行编程，也比较容易掌握。

✕ 位处理 — 不容易直观理解处理内容 (①②)

○ 字处理 — 行数较少 (②)  
很容易直观理解表述式及字符串 (②③)  
无需考虑运算的优先顺序 (②)  
无需存储运算的整个过程 (②)

脚本的优势

如果使用梯形图+脚本：

能够在同一程序中夹杂使用梯形图和脚本

框脚本  
使用梯形图编程执行条件。

区域脚本  
因没有执行条件，因此，总是执行。

○ 使用梯形图编程位处理 — 容易直观理解处理内容

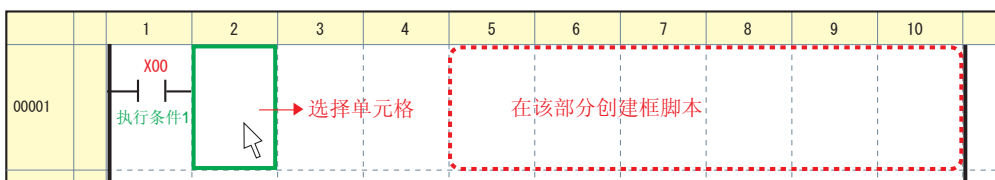
○ 使用脚本编程字处理 — 行数较少  
容易直观理解表述式及字符串  
无需考虑运算的优先顺序  
无需存储运算的整个过程

今后的 PLC 编程兼具梯形图与脚本的优点，可显著减少设计、调试和更改时间。

下面介绍新建框脚本的步骤

1

使用梯形图编辑器编程执行条件，点击要创建框脚本的单元格。



2

在右键菜单中依次选择“编辑脚本(X)”→“插入框脚本(B)”

编辑脚本(X)

编辑 CPU 定位参数(I)...  
软元件注释编辑窗口(K)... Ctrl+F7  
软元件使用列表(Z)... Ctrl+E

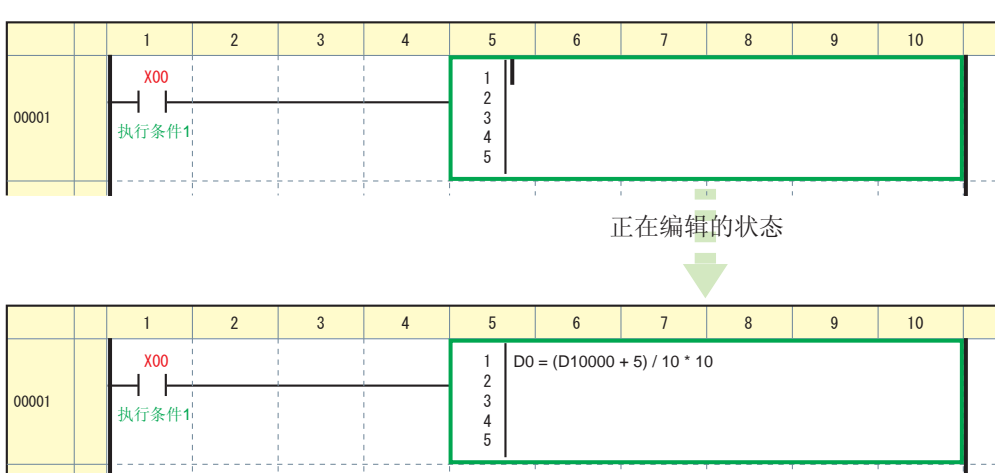
插入域脚本(A) Ctrl+R  
插入框脚本(B) Ctrl+B  
转换选择的脚本(S) Ctrl+T  
转换所有脚本(C)  
显示/隐藏转换梯形图(Y) Ctrl+G  
全部显示转换梯形图(X)  
全部折叠转换梯形图(O)

步骤

- **Ctrl** 键 + **B** 键
- 在菜单中依次选择“脚本(S)”→“插入框脚本(B)”

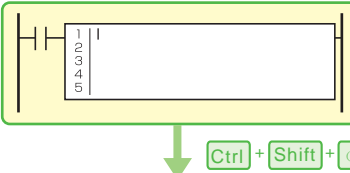
3

框脚本被插入，请编写程序。完成编程后，请按下 **Ctrl** 键 + **Enter** 键或 **Esc** 键，或点击当前正在编辑的框脚本以外的梯形图编辑窗口，退出编程。



\* **Ctrl** 键 + **Enter** 键，或 **Esc** 键，完成。

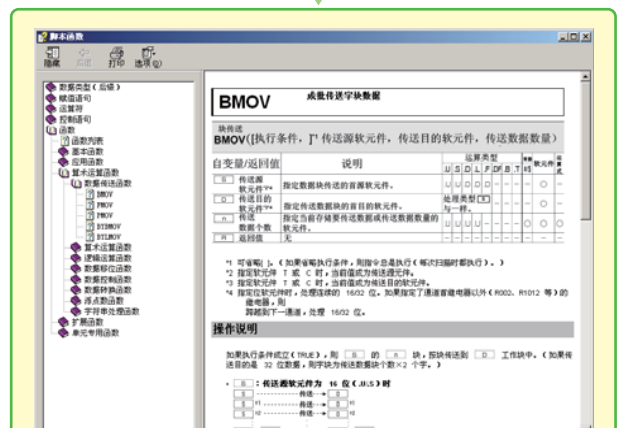
不借助手册，也能够对脚本函数进行简便编程。



**Ctrl** + **Shift** + (空格)

在“脚本函数列表”中选择函数

使用“脚本函数提示”，介绍简易格式



在菜单的帮助中含有“脚本函数帮助”，可浏览函数的详细介绍。

注释的编程方法

在脚本中输入“'”（单引号），则会将“'”到行末之前的内容作为注释进行处理。若在行头编写“'”，则整行成为注释；若在运算公式的后面编写“'”，则公式后面的内容成为运算公式的注释

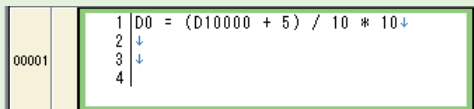
例如 '该行全部作为注释进行处理。  
D0 = D1 + D2 '将 D1和 D2合并，存储到 D0

可选择的编辑方法

若想输入/编辑脚本，可选择“直接输入”和“窗口”。

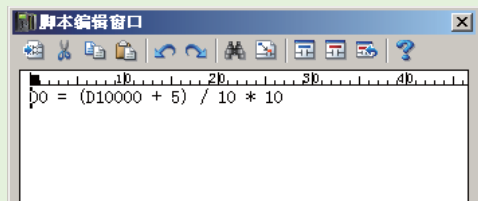
“直接输入”

可以使用使用梯形图程序一样输入脚本。适用于输入较短运算公式的脚本。



“窗口”

显示脚本编辑专用窗口。适用于输入包含条件分支的较长脚本。



4 www.keyence.co.jp/seigyo

www.keyence.co.jp/seigyo 5

管理装置的运转/停止时间，计算利用率

管理装置的累积运转/停止时间，计算利用率。  
同时采集停机次数。

例

累积运转时间	185: 19
累积停止时间	1: 55
利用率	98.9%
停机次数	3次



人机界面  
VT3 系列

<使用的软元件>

DM100···秒计数器（运转）	DM105···秒计数器（停止）
DM101···分计数器（运转）	DM106···分计数器（停止）
DM102···累积秒计数器（运转）	DM107···累积秒计数器（停止）
DM108···停机次数	MR100···运转中标志
MR101···复位开关	CR2006···1秒时钟脉冲

说明

◎ 算术运算符

返回 2 个值的算术运算结果

符号	处理内容
^	求乘方
*	求 2 个值的积（乘法）
/	求 2 个值的商（除法）
MOD*1	进行 2 个值的除法，返回余值
+	求 2 个值的和（加法）
-	求 2 个值的差（减法）

\*1: 需在符号的前后留半角空格

◎ 关于后缀

与梯形图语言可通过对各指令添加后缀来决定命令的动作类型相反，KV 脚本通过对设备及常数添加后缀来决定函数的动作类型及设备中存储的值的意

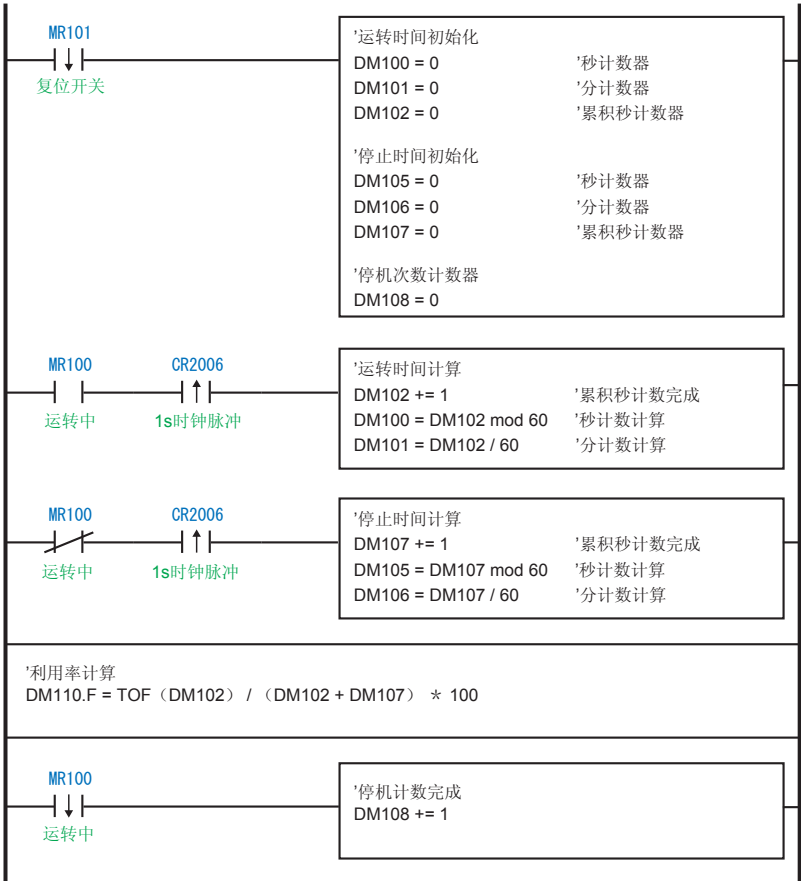
后缀的种类

后缀	类型	可处理范围
.U	16 位无符号数据类型	0~65535
.S	16 位带符号数据类型	-32768~+32767
.D	32 位无符号数据类型	0~4294967295
.L	32 位带符号数据类型	-2147483648~+2147483647
.F	浮点型	-3.4E38 ≤ N ≤ -1.4E -45 N = 0 1.4E -45 ≤ N ≤ 3.4E38 (有效 7 位)
.B	位型（布尔值）	1、0、ON、OFF、TRUE、FALSE
.T	字符串型	字符串

◎ 数据类型转换函数

TOU（转换对象软元件）·····将数据的类型转换成.U  
TOS（转换对象软元件）·····将数据的类型转换成.S  
TOD（转换对象软元件）·····将数据的类型转换成.D  
TOL（转换对象软元件）·····将数据的类型转换成.L  
TOF（转换对象软元件）·····将数据的类型转换成.F  
FLOAT（转换对象软元件）·····将数据的类型转换成.F

使用 KV 脚本，查看非常方便



根据目标坐标计算搬运机械臂的角度和长度

根据已输入的目标坐标，计算搬运机械臂的移动角度和长度

例

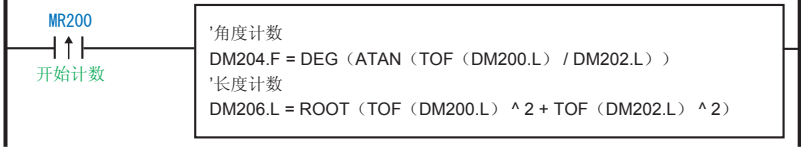
X 坐标: 40000	Y 坐标: 30000
↓ 计数	
角度: 53.13°	长度: 50000



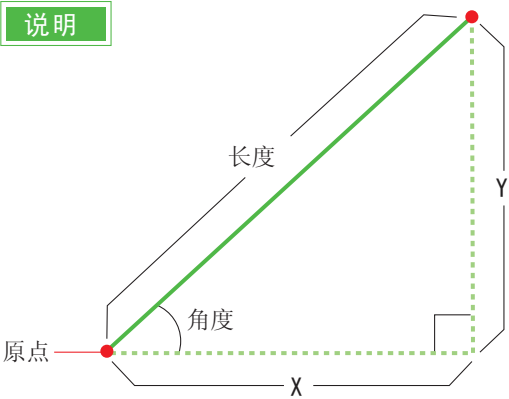
人机界面  
VT3 系列

<使用软元件>    DM200···X 坐标    DM202···Y 坐标  
                      DM204···角度    DM206···长度

使用 KV 脚本，变得如此简单



说明



公式

反正切（正切的反函数）  
角度 $\theta = \tan^{-1} \left( \frac{Y}{X} \right)$

勾股定理（毕达哥拉斯定理）  
长度 $= \sqrt{X^2 + Y^2}$

角度  $\theta$ ，使用正切（对边  $Y \div$  底边  $X$ ）的反函数 **ATAN**（反正切）进行计算。  
长度  $R$  为  $[X^2+Y^2]$  的平方根，可使用 **ROOT** 函数。

◎ **DEG** 弧度→度（°）单位转换

返回值 = <sup>度数</sup>**DEG**（转换对象软元件）

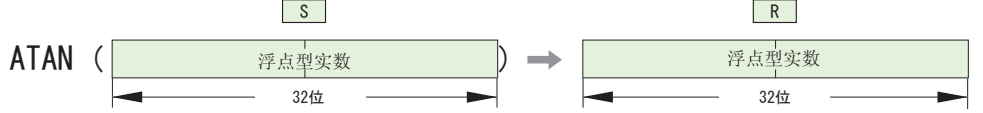
**[S]** 将转换对象软元件中存储的 32 位浮点型实数的角度（弧度单位）转换成度（°）单位。结果变成 32 位浮点型实数值，存储到 **[R]** 返回值中。



◎ **ATAN** 正接值→角度（弧度）

返回值 = <sup>反正切</sup>**ATAN**（运算对象软元件）

**[S]** 根据运算对象软元件中存储的 32 位浮点型实数的正切（**TAN**）值计算角度（弧度单位）。结果变成 32 位浮点型实数值，存储到 **[R]** 返回值中。

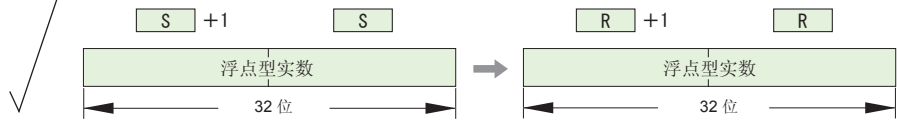


◎ **ROOT** 平方根

返回值 = <sup>平方根</sup>**ROOT**（运算对象软元件）

**[S]** 计算运算对象软元件中存储数据的平方根，将结果存储到 **[R]** 返回值中。

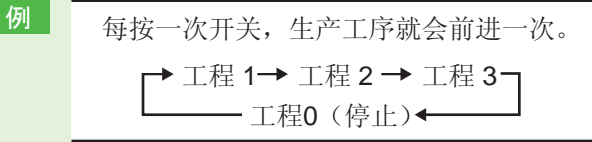
**[S]** 运算对象软元件为浮点型实数（.F）时





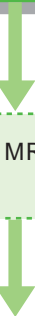
简便编程生产排程

无需通过继电器传送，即可非常简便地对生产排程的状态迁移进行编程。

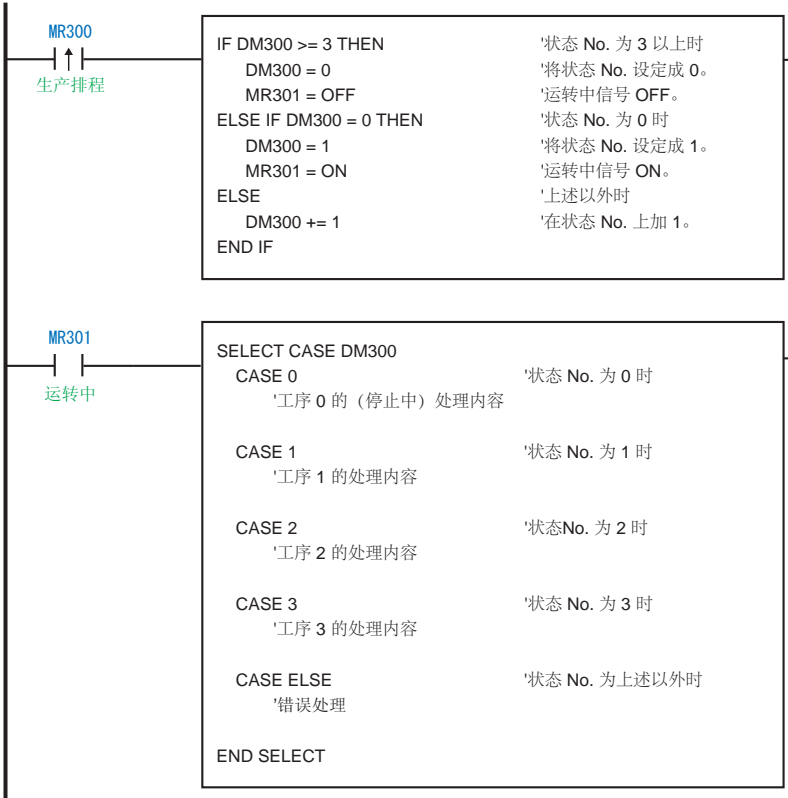


人机界面  
VT3 系列

<使用的软元件> DM300···状态 No.  
MR301···运转中信号



使用 KV 脚本，将变得如此简单



说明

◎ IF～ELSE JF～语句（条件分支）

若不满足最初的条件式，则使用下一个条件式再次进行判定。  
该语句是拥有多个条件的条件分支语句。

基本格式	格式说明
IF<条件式1> THEN	条件式 1 成立时，
<处理语句1>	执行处理语句 1。
ELSE IF<条件式2> THEN	条件式 1 不成立，但条件式 2 成立时
<处理语句2>	执行处理语句 2。
ELSE IF<条件式3> THEN	条件式 1 和 2 不成立，但条件式 3 成立时
⋮	⋮
ELSE*1	所有条件式都不成立时，
<处理语句n>*1	执行处理语句 n。
END JF	IF 语句至此结束。

\* □ 表示插入半角空格。  
\* 有标记的地方，请务必插入换行。  
\* 1: 可以省略。  
\* 2: 可设定任意数量的条件式和处理语句。没有上限。

◎ SELECT CASE～语句（多分支控制）

该语句是能够根据成为基准的字软元件中存储的值来选择处理的条件分支语句。

基本格式	格式说明	格式示例	说明
SELECT CASE 字软元件	以字软元件的值为基准，进行条件分支	SELC CASE DM1000	'将 DM1000 中存储的值设为基准
CASE 常数1*3	常数 1 和字软元件的值相等时	CASE 1	'DM1000 = 1 时
处理语句1	执行处理语句 1。	DM2000 = 1000	'在 DM2000 中存储 1000
CASE 常数2	常数 2 和字软元件的值相等时	CASE 2 TO 10	'DM1000 = 2～10 时
处理语句2	执行处理语句 2。	DM2000 = 2000	'在 DM2000 中存储 2000
CASE 常数3	常数 3 和字软元件的值相等时	CASE 11,15,60	'DM1000 = 11、15 或 60 时
⋮	⋮	DM2000 = 3000	'在 DM2000 中存储 3000
CASE ELSE*1	常数和字软元件的值均不相等时，	CASE IS > 100	'DM1000 的值大于 100 时
处理语句n*1	执行处理语句 n。	DM2000 = 4000	'在 DM2000 中存储 4000
END SELECT	SELECT 语句至此结束。	CASE ELSE	'DM1000 的值是上述以外时
		DM2000 = 5000	'在 DM2000 中存储 5000
		END SELECT	

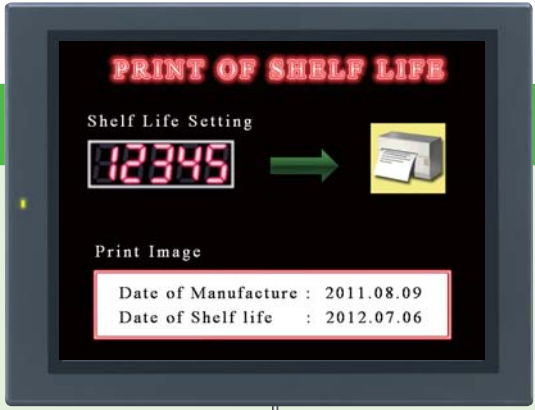
\* □ 表示插入半角空格。  
\* 有标记的地方，请务必插入换行。  
\* 1: 可以省略。  
\* 2: 《CASE 常数（ELSE）》最多可编程200个。  
\* 3: 无法在常数1中指定字符串及处理字符串的设备、位型设备  
\* 能够在常数部分编写使用比较运算符的条件式。  
此时，使用“IS”进行比较。  
（例）IS >= 100（100 以上）、IS < 30（除 30 以外）

创建食品有效期的打印数据

根据当前时间和设定的有效期，计算食品有效期，创建打印数据。

例

设定有效期	36 时间
制造日期	2005 年 12 月 31 日 18 时
食品有效期	2006 年 1 月 2 日 06 时



人机界面  
VT3 系列

<使用的软元件> DM400·····有效期      DM410~·····制造日期  
DM430~·····食品有效期

使用 KV 脚本，字符串处理非常轻松

'创建制造日期数据 DM410.T = "制造日期: " + STR(2000 + CM700) + "年" DM410.T += STR(CM701) + "月" + STR(CM702) + "日" DM410.T += RIGHT(DASC(CM703,2),2) + "时" + CHR(13) + CHR(10)
'食品有效期计算 RSEC((SEC(CM700) + DM400 * 60 * 60),DM450)
'创建食品有效期数据 DM430.T = "食品有效期: " + STR(2000 + DM450) + "年" DM430.T += STR(DM451) + "月" + STR(DM452) + "日" DM430.T += RIGHT(DASC(DM453,2),2) + "时" + CHR(13) + CHR(10)

说明

◎ 字符串连接运算符

返回 2 个字符串的连接结果。

符号	处理内容
&. +*1	连接 2 个字符串

\*1: “+”根据编程内容自动判断算术运算和字符串连接。

◎ STR 数值数据→ASCII 字符串转换

十进制 ASCII 码转换  
返回值 = STR (转换对象软元件)

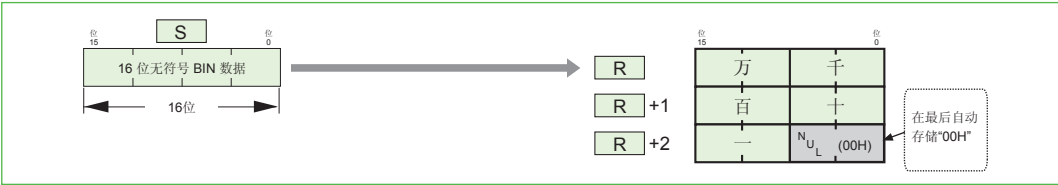
进行与 DASC 函数相同的操作

◎ DASC BIN 数据→十进制 ASCII 字符串转换

十进制 ASCII 码转换  
返回值 = DASC (转换对象软元件、转换参数)

[S1] 将转换对象软元件中存储的二进制数据按照 [S2] 指定的转换参数，转换成十进制 ASCII 码字符串，再将结果存储到 [R] 返回值。

[S1]：转换对象软元件为 16 位无符号数据 (.U) 时



◎ RIGHT 右截取字符串

右截取字符串  
返回值 = DASC (截取对象软元件、截取字符数)

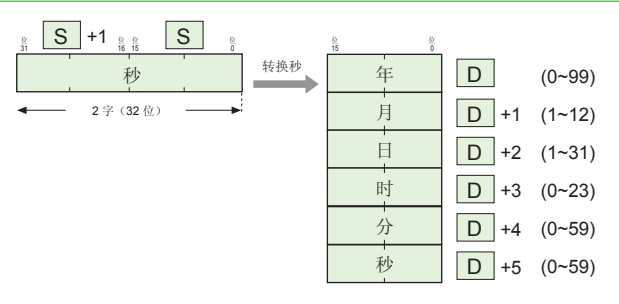
将从 [S] 截取对象软元件开始进行存储的字符串从右侧截取以 [n] 指定的截取字符数 (字节数)，添加结束符 NUL (00H) 后，将结果存储到 [R] 返回值。

◎ RSEC 将秒格式转换成日期数据

秒→日期转换  
RSEC ( [执行条件], \*转换源软元件、转换目的软元件 )

将以 [S] 转换源软元件指定的、以 2000 年 1 月 1 日 0 时 0 分 0 秒为基准日期的总秒数 (32 位二进制数据 (无符号)) 转换成日期数据，存储到从 [D] 转换目的软元件开始的 6 字 (96 位) 中。

\*[]可以省略。



计算加工次数和各目标坐标

根据测定的薄板长度，计算可加工的张数和各目标坐标。

例	薄板长度：1250cm 加工长度：160cm 开始坐标：3cm
	↓ 计算
	可加工张数：7张 剩余：130cm 坐标 1：3cm 坐标 2：163cm . . .



人机界面  
VT3 系列

<使用的软元件>	DM500 . . . 薄板长度	DM501 . . . 剩余薄板长度
	DM502 . . . 加工长度	DM503 . . . 可加工张数
	FM0 . . . . 开始坐标	FM2 ~ . . . 各目标坐标
	MR500 . . . 开始计算	

使用 KV 脚本，将变得如此简单

MR500	↑↑	'初始化
加工次数计算		FMOV (0,FM2,10000) '从 FM2 开始初始化 10000 个。
		DM503 = 0 '初始化可加工张数。
		'复制剩余薄板长度
		DM501 = DM500 '将薄板长度复制到剩余长度。
		'循环加工次数
		WHILE DM501 >= DM502 '剩余薄板长度大于加工长度时
		DM503 += 1 '在可加工张数中加 1。
		DM501 -= DM502 '从剩余长度中扣除加工长度。
		FM0.L: (DM503 * 2) = FM0.L + EM502 * EM503 '存储各目标坐标。
		END WHILE

说明

◎ WHILE 语句（首先判断循环条件，然后执行循环控制）

循环条件为“TRUE（真）”时，执行循环处理语句。

基本格式	格式说明	基本格式	说明
WHILE <条件式>	条件式成立时，*1	WHILE DM1000 < 100	'DM1000 的值小于 100 时，*2
<处理语句>	执行处理语句。	DM1000 += 5	'在 DM1000 中加 5。
END WHILE	返回到语句开头(WHILE)	END WHILE	'返回到语句开头（WHILE）

\*1: 表示插入半角空格。  
\* 有 标记的地方，所图片放在上述空格处。  
\* 1: 条件式不成立时，退出循环处理。  
\* 2: DM1000 的值大于 100 时，不执行处理语句，从 END WHILE 的下一行开始执行。

而且，DO~UNTIL/FOR~NEXT 可作为循环控制语句使用。  
此外，可通过 BREAK 语句从循环控制中退出。

格式示例	说明
<循环控制语句内> IF DM502 = 0 THEN BREAK END IF	DM502 为 0 时， 强制退出该循环处理。 IF 语句结束

注 在循环控制语句中，请务必按循环处理退出那样来编写脚本。若循环处理不退出，则会变成“无限循环”，执行程序时，变成扫描定时上溢（无限循环不会检测为错误）。

◎ 什么是变址修改？

所谓变址修改 (Index Modification)，是在指定为指令操作数的各软元件中加上变址寄存器 (Z) 的值及常数后、对软元件进行编号的方法。能够进行加法运算的值的范围为带符号 16 位数值（-32768~+32767）。能够变址修改的软元件种类为 R、MR、LR、T、C、DM、EM、FM、TM。

◎ 基于变址寄存器的修改

使用该功能，可在运转过程中，更改指令操作数要参照的软元件。  
基于变址寄存器修改软元件的编程方法为：“（软元件号）：Z（01~10）”。

例	DM01000: Z01 . . . . . Z01 的值为 -10 时，参照 DM00990。 R00000: Z05 . . . . . Z05 的值为 +16 时，参照 R00100。
---	--

◎ 基于软元件及表达式的变址修改

除变址寄存器外，还可以使用 KV 脚本进行变址修改。  
此时，在转换梯形图时会自动使用系统预留的变址寄存器（Z11、Z12）。

例	DM1000: (DM10+2)=100
---	----------------------



# PROGRAMMABLE CONTROLLER KV-5000/3000

## KV-5000 NEW REALTIME Logic Controller



### 支持网络，大容量 CPU

基本性能	最大 I/O 点数 3096 点	程序容量 260k 步	LD/OUT 指令 10ns
通信端口	Ethernet/ FL-net	USB	
内置 I/O	24 点		
系统	模块化编程	KV 脚本	直接刷新

## KV-3000 NEW REALTIME Logic Controller



### 实时标准 CPU

基本性能	最大 I/O 点数 3096 点	程序容量 160k 步	LD/OUT 指令 10ns
通信端口	USB	串行	
内置 I/O	24 点		
系统	模块化编程	KV 脚本	直接刷新

## KV-1000



基本性能	最大 I/O 点数 3096 点	程序容量 160k 步	LD/OUT 指令 25ns
通信端口	USB	串行	
内置 I/O	24 点		
系统	模块化编程	KV 脚本	



www.keyence.com.cn

基恩士(中国)有限公司 最新发售情况，请咨询就近的基恩士

上海 200120 上海市浦东新区世纪大道 100 号上海环球金融中心 8 楼 电话：+86-21-5058-6228 传真：+86-21-5058-7178  
【关于产品的咨询，请致电】 电话：+86-21-3357-1001 传真：+86-21-6496-8711

北京 电话：+86-10-8447-5835 传真：+86-10-8447-5370	青岛 电话：+86-532-6677-7110 传真：+86-532-8571-8182	苏州 电话：+86-512-6809-8612 传真：+86-512-6809-8613	深圳东 电话：+86-755-2588-2550 传真：+86-755-8247-8972
天津 电话：+86-22-8319-1775 传真：+86-22-8319-1578	上海虹桥 电话：+86-21-3357-1001 传真：+86-21-6496-8711	武汉 电话：+86-27-8771-7558 传真：+86-27-8771-7557	深圳西 电话：+86-755-2588-2551 传真：+86-755-8627-1027
大连 电话：+86-411-3986-9011 传真：+86-411-3986-9010	南京 电话：+86-25-8322-9577 传真：+86-25-8322-6277	无锡 电话：+86-510-6662-8800 传真：+86-510-6605-6890	广州 电话：+86-20-3878-1155 传真：+86-20-3878-0199

基恩士(香港)有限公司 香港红磡都会道 10 号都会大厦 23 楼 2308 室 电话：+852-3104-1010 传真：+852-3104-1080



最新信息  
扫一扫关注  
基恩士微信



安全方面的注意事项  
为了安全使用商品，请务必在  
使用之前仔细阅读《使用说明书》。

咨询热线

4007-367-367  
E-mail: info@keyence.com.cn

日本語ダイヤル：+86-21-5058-7128