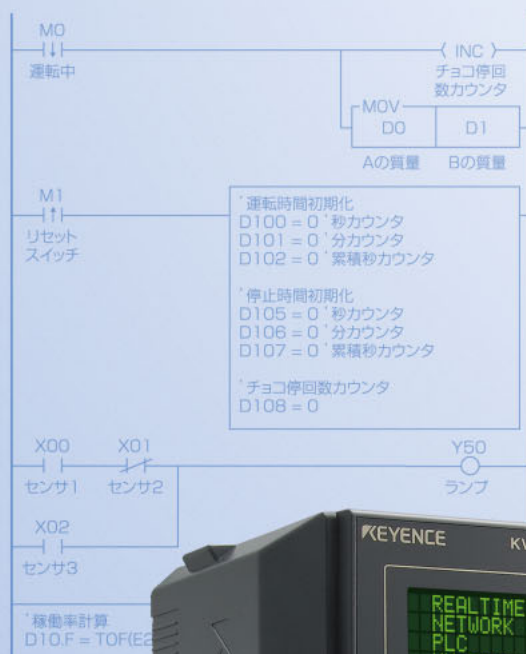


ラダーとスクリプトを融合する

KV-5000/3000/1000シリーズ

KVスクリプト 活用テクニック

即効使えるアプリケーション集2



INDEX

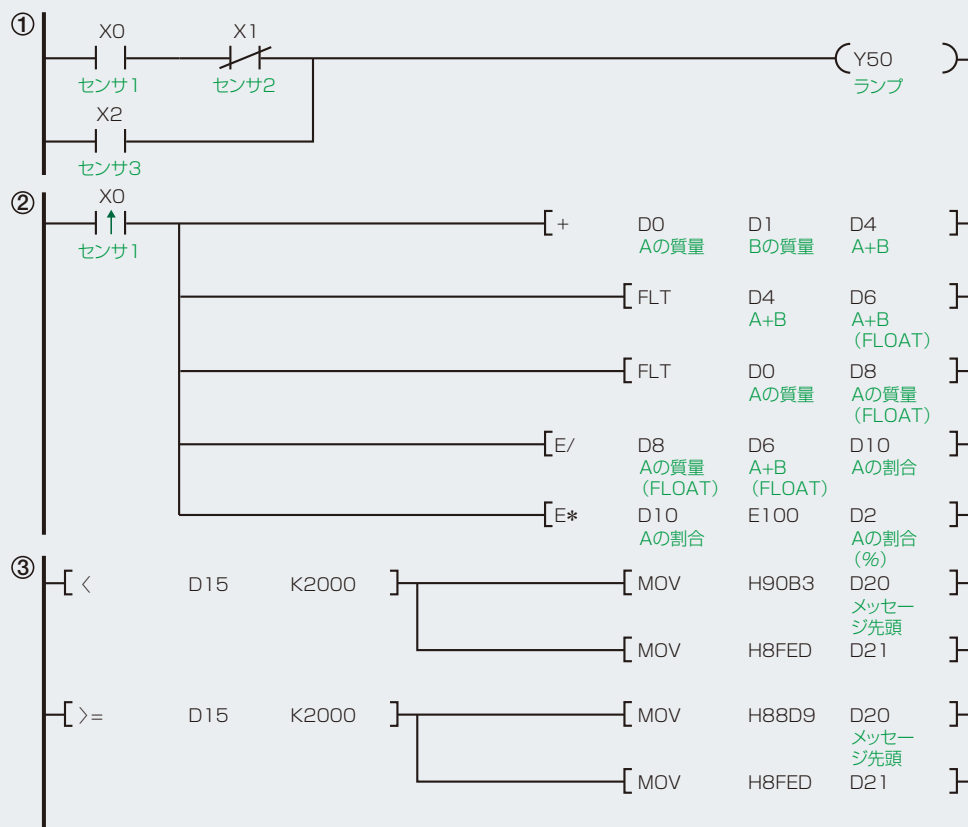
ラダーとスクリプトの融合	2
KVスクリプトの記述方法	4
充実のアシスト機能	5
KVスクリプト活用例① 稼働管理 — 四則・浮動小数点演算	6
KVスクリプト活用例② 角度計算 — 三角関数	8
KVスクリプト活用例③ 工程歩進 — 条件分岐	10
KVスクリプト活用例④ 賞味期限印字 — 文字列連結	12
KVスクリプト活用例⑤ 加工回数と目標座標 — 繰り返し処理	14

ラダーとスクリプトの融合

例えば、ある装置の仕様の一部が以下のようにになっています。

- ① センサ1 (X0) がONかつセンサ2 (X1) がOFF、またはセンサ3 (X2) がONしているときにランプ (Y50) をONします。
- ② センサ1 (X0) の立ち上がりで、Aの重量 (D0) とBの重量 (D1) の割合 (D2) を計算します。
- ③ 測定値 (D15) が2000未満の時はメッセージ (D20～) に「正常」を、2000以上の時はメッセージ (D20～) に「異常」を格納します。

従来のラダーのみで記述すると・・・



○ ビット処理 — 処理内容を直感的に把握しやすい (①)

✕ ワード処理 — 行数が多い (②)

数式や文字列が直感的に把握しにくい (② ③)

演算の優先順位を意識する必要あり (②)

演算の途中経過を格納する必要あり (②)

ラダーの長所

これからのPLCプログラミングはラダーとスクリプトの

スクリプトのみで記述すると・・・

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 = "異常"
END IF

スクリプト言語とは？

小規模なプログラムを簡単に作成するための言語で、一般のプログラミング言語よりも簡単に記述できるので習得も容易である。

✕ビット処理 — 処理内容を直感的に把握しにくい(① ②)

○ワード処理 — 行数が少ない(②)

数式や文字列が直感的に把握しやすい(② ③)

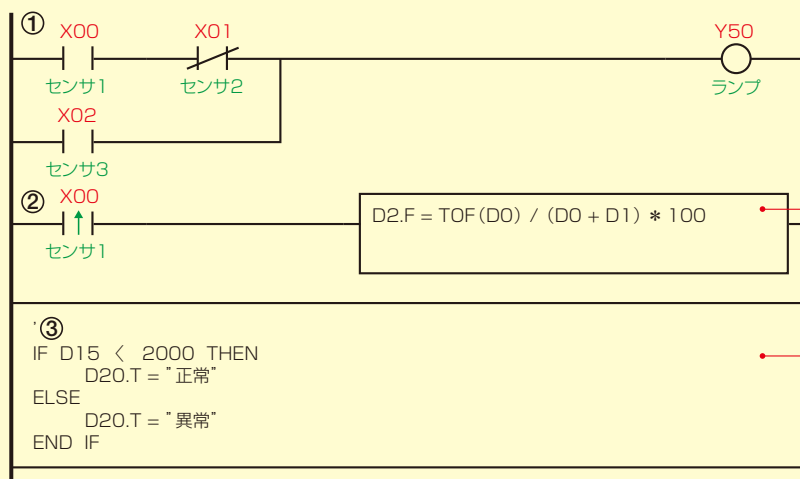
演算の優先順位を意識する必要なし(②)

演算の途中経過を格納する必要なし(②)

スクリプトの長所

ラダー＋スクリプトなら

同一プログラム中でラダーとスクリプトを混在して使用可能



ボックススクリプト

実行条件をラダーで記述します。

エリアスクリプト

実行条件を持たないので、常時実行します。

○ビット処理はラダーで記述 — 処理内容を直感的に把握しやすい

○ワード処理はスクリプトで記述 — 行数が少ない

数式や文字列が直感的に把握しやすい

演算の優先順位を意識する必要なし

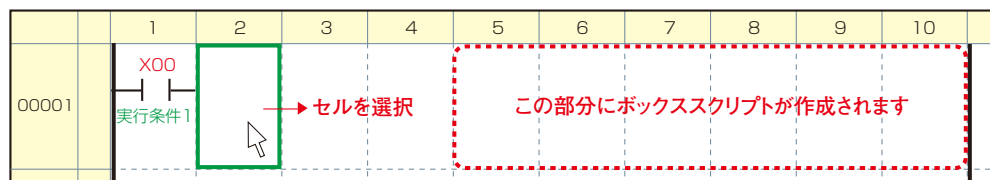
演算の途中経過を格納する必要なし

メリットを両立して設計・デバッグ・変更工数を削減できます

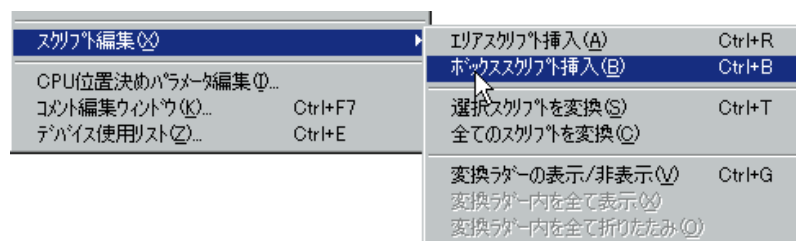
KVスクリプト記述方法

ボックススクリプトを新規作成する手順を説明します。

- 1** ラダーエディタで実行条件を記述し、ボックススクリプトを作成したいセルをクリックします。

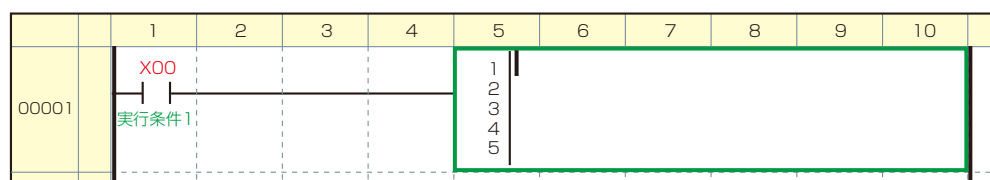


- 2** 右クリックメニューから【スクリプト編集 (X)】→【ボックススクリプト挿入 (B)】を選択します。

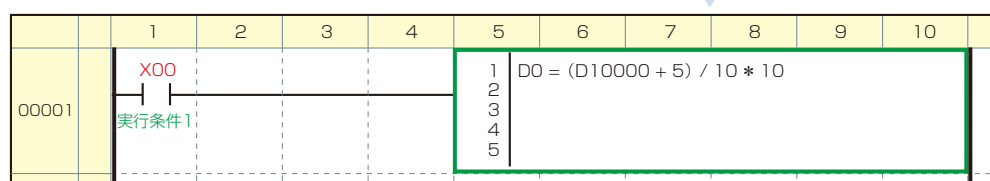


- 別手順**
- ・**Ctrl** キー + **B** キー
 - ・メニューから【スクリプト (S)】→【ボックススクリプト挿入 (B)】

- 3** ボックススクリプトが挿入されますので、プログラムを記述してください。記述が終わったら、**Ctrl** キー + **Enter** キーもしくは **Esc** キーを押すか、編集集中のボックススクリプト以外のラダー編集ウィンドウをクリックし、記述を完了してください。



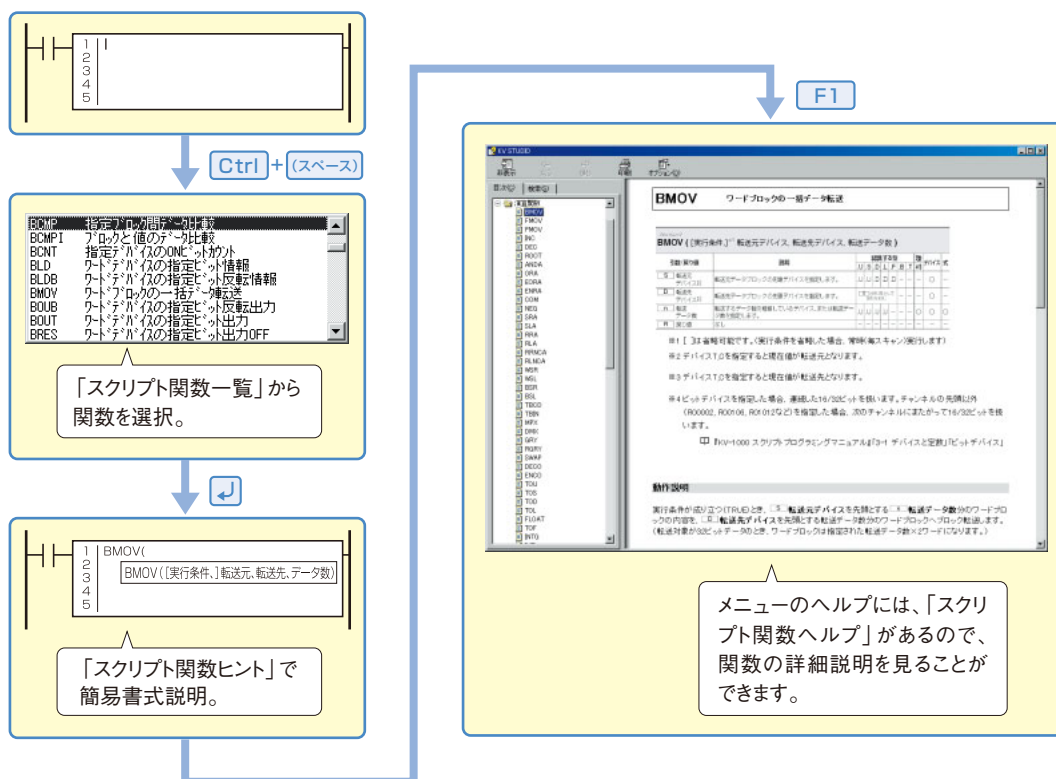
編集集中の状態です



※ **Ctrl** キー + **Enter** キーもしくは **Esc** キーで完了。

充実のアシスト機能

スクリプト関数もマニュアルレスで簡単に記述できます。



コメントの記述方法

スクリプトでは、「'」（シングルクォート）を入力すると、「'」～行末までをコメントとして扱います。行頭に「'」を記述すれば、行全てがコメントになり、演算式の後ろに「'」を記述すれば演算式の注釈となります。

記述式 'この行は全てコメントとして扱われます。

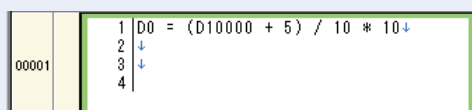
D0 = D1 + D2 'D1とD2を足し合わせD0に格納

選べる編集方法

スクリプトの入力・編集には「直接入力」と「ウィンドウ」を選ぶことができます。

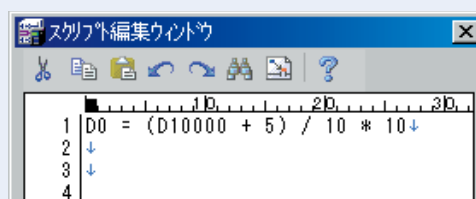
「直接入力」

ラダープログラムと同じ感覚でスクリプトの入力が可能。短い演算式のスクリプト入力に適しています。



「ウィンドウ」

スクリプト編集専用ウィンドウが表示されます。条件分岐を含めた長いスクリプトの入力に適しています。



KVスクリプト活用例①

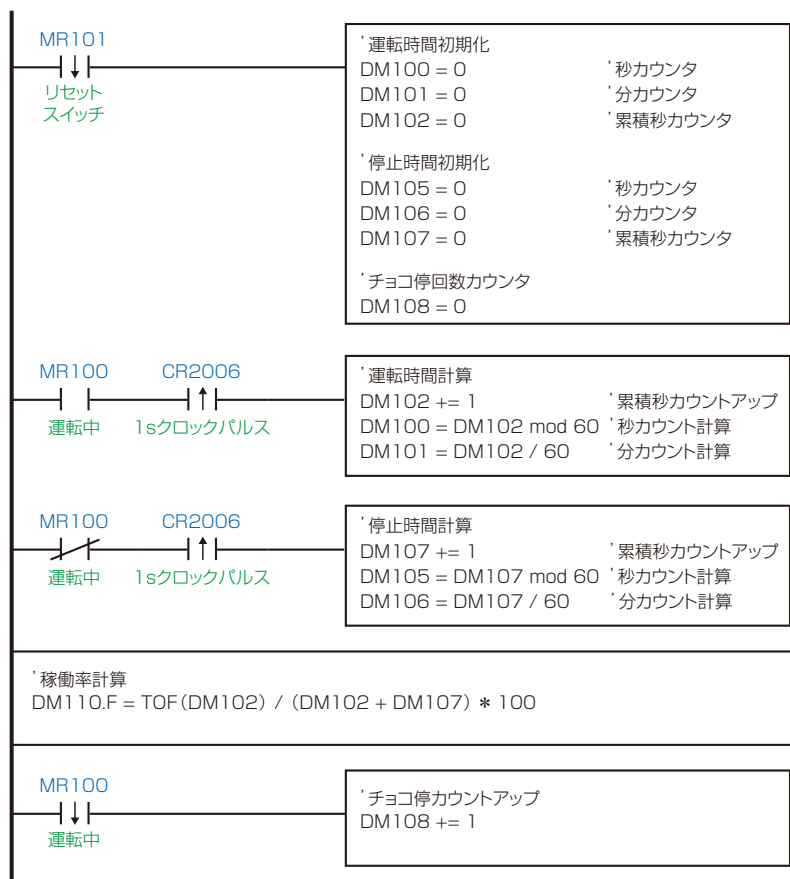
装置の運転・停止時間を管理して稼働率を計算する

装置の累積運転・停止時間を管理して、稼働率を計算します。
同時にチョコ停回数も取得します。

例	累積運転時間	185:19
	累積停止時間	1:55
	稼働率	98.9%
	チョコ停回数	3回

【使用デバイス】	DM100・・・秒カウンタ (運転)	DM105・・・秒カウンタ (停止)
	DM101・・・分カウンタ (運転)	DM106・・・分カウンタ (停止)
	DM102・・・累積秒カウンタ (運転)	DM107・・・累積秒カウンタ (停止)
	DM108・・・チョコ停回数	DM110・・・稼働率
	MR100・・・運転中フラグ	MR101・・・リセットスイッチ

KVスクリプトを使用すればこんなに見やすい





タッチパネルディスプレイ
VT3シリーズ

解説

◎ 算術演算子

2つの値の算術計算の結果を返します。

記号	処理内容
^	累乗を求めます
*	2つの値の積を求めます (乗算)
/	2つの値の商を求めます (除算)
MOD※1	2つの値の除算を行ない、その剰余を返します
+	2つの値の和を求めます (加算)
-	2つの値の差を求めます (減算)

※1: 記号の前後に半角スペースが必要です。

◎ サフィックスについて

KV-5000/3000/1000のラダー言語では各命令語にサフィックスを付けることで、命令の動作の型が決まるのに対して、KVスクリプトではデバイスや定数にサフィックスを付けることで、関数の動作の型や、デバイスに格納されている値の意味が決まります。

サフィックスの種類

サフィックス	型	扱う範囲
.U	16ビット符号無しデータ型	0~65535
.S	16ビット符号付きデータ型	-32768~+32767
.D	32ビット符号無しデータ型	0~4294967295
.L	32ビット符号付きデータ型	-2147483648~+2147483647
.F	浮動小数点型	$-3.4E38 \leq N \leq -1.4E-45$ $N = 0$ $1.4E-45 \leq N \leq 3.4E38$ (有効7桁)
.B	ビット型 (ブール値)	1、0、ON、OFF、TRUE、FALSE
.T	文字列型	文字列

◎ データ型変換関数

TOU (変換対象デバイス) ……データの型を.U に変換
 TOS (変換対象デバイス) ……データの型を.S に変換
 TOD (変換対象デバイス) ……データの型を.D に変換
 TOL (変換対象デバイス) ……データの型を.L に変換
 TOF (変換対象デバイス) ……データの型を.F に変換
 FLOAT (変換対象デバイス) ……データの型を.F に変換

KVスクリプト活用例②

目標座標から搬送用アームの角度と長さを計算する

入力した目標座標から搬送用アームの移動する角度と長さを計算します。

例	X座標:40000	Y座標:30000
	↓計算	
	角度:53.13°	長さ:50000

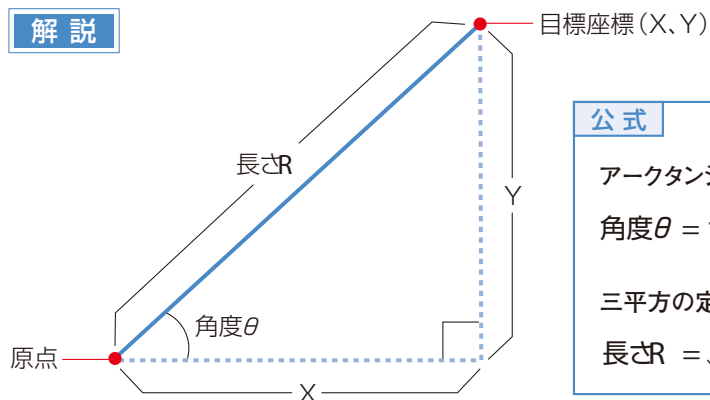
【使用デバイス】 DM200・・・X座標 DM202・・・Y座標
DM204・・・角度 DM206・・・長さ

KVスクリプトならこんなにシンプル

MR200
↑↑↑
計算開始

```
'角度計算  
DM204.F = DEG(ATAN(TOF(DM200.L) / DM202.L))  
'長さ計算  
DM206.L = ROOT(TOF(DM200.L) ^ 2 + TOF(DM202.L) ^ 2)
```

解説



公式

アークタンジェント (タンジェントの逆関数)

$$\text{角度}\theta = \tan^{-1}\left(\frac{Y}{X}\right)$$

三平方の定理 (ピタゴラスの定理)

$$\text{長さ}R = \sqrt{X^2 + Y^2}$$

角度θはタンジェント (対辺Y÷底辺X) の逆関数であるATAN (アークタンジェント) 関数を使用して求めます。
長さRは $X^2 + Y^2$ の平方根なので、ROOT (ルート) 関数を使用します。

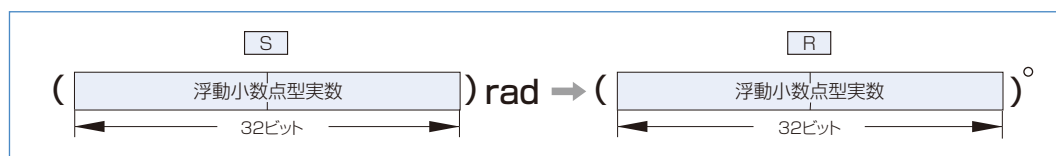


タッチパネルディスプレイ
VT3シリーズ

◎ DEG ラジアン→度(°)単位変換

ディグリー
戻り値 = **DEG** (変換対象デバイス)

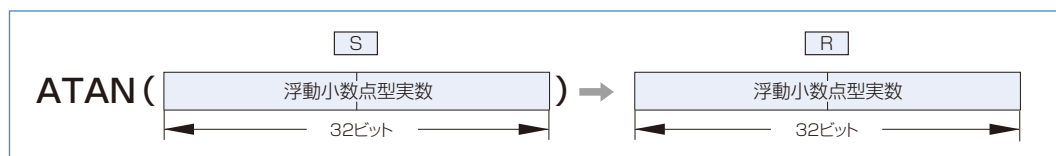
[S] 変換対象デバイスに格納された32ビット浮動小数点型実数の角度(ラジアン単位)を度(°)単位に変換します。結果は32ビット浮動小数点型実数値となり、**[R]** 戻り値に格納されます。



◎ ATAN 正接値→角度(ラジアン)

アークタンジェント
戻り値 = **ATAN** (演算対象デバイス)

[S] 演算対象デバイスに格納された32ビット浮動小数点型実数の正接(TAN)値から角度(ラジアン単位)を求めます。結果は32ビット浮動小数点型実数値となり、**[R]** 戻り値に格納されます。

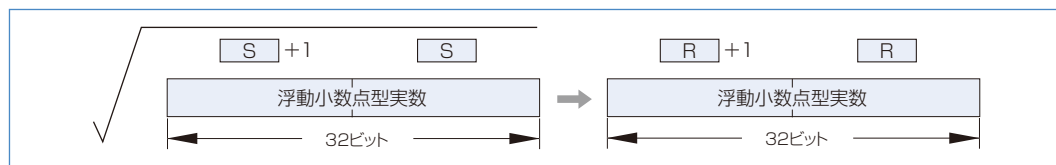


◎ ROOT 平方根

平方根
戻り値 = **ROOT** (演算対象デバイス)

[S] 演算対象デバイスに格納されたデータの平方根を求め、結果を**[R]** 戻り値に格納します。

[S] 演算対象デバイスが浮動小数点型実数(.F)の場合



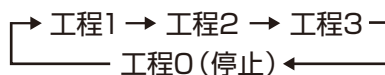
KVスクリプト活用例 ③

簡単に工程歩進のプログラムを組みたい

工程歩進の状態遷移をリレーの受け渡しなしで簡単にプログラムします。

例

スイッチを押すたびに工程が歩進します



【使用デバイス】 DM300・・・状態No.
MR301・・・運転中信号

KVスクリプトならこんなに簡単

MR300
↑↑↑
工程歩進

```
IF DM300 >= 3 THEN
  DM300 = 0
  MR301 = OFF
ELSE IF DM300 = 0 THEN
  DM300 = 1
  MR301 = ON
ELSE
  DM300 += 1
END IF
```

・状態No.が3以上の時
・状態No.を0にします。
・運転中信号をOFFします。
・状態No.が0の時
・状態No.を1にします。
・運転中信号をONします。
・上記以外の時
・状態No.を1足します。

MR301
| |
運転中

```
SELECT CASE DM300
CASE 0
  工程0での(停止中の)処理内容
CASE 1
  工程1での処理内容
CASE 2
  工程2での処理内容
CASE 3
  工程3での処理内容
CASE ELSE
  エラー処理
END SELECT
```

・状態No.が0の時
・状態No.が1の時
・状態No.が2の時
・状態No.が3の時
・状態No.が上記以外の時



タッチパネルディスプレイ
VT3シリーズ

解説

◎ IF～ELSE、IF～文（条件分岐）

最初の条件式を満たさない場合、次の条件式にて再度判定をおこないます。
条件をいくつも持てる条件分岐文です。

基本書式	書式説明
IF<条件式1> THEN	条件式1が成り立つ時、
<処理文1>	処理文1を実行します。
ELSE IF<条件式2> THEN	条件式1は成り立たないが、条件式2が成り立つ時
<処理文2>	処理文2を実行します。
ELSE IF<条件式3> THEN	条件式1,2は成り立たないが、条件式3が成り立つ時
⋮	⋮
ELSE※1	全ての条件式が成り立たない時、
<処理文n>※1	処理文nを実行します。
END IF	IF文はここで終わります。

※ 半角スペース挿入を表します。

※ マークがあるところは、必ず改行を入れてください。

※1: 省略可能です。

※2: 任意の数の条件式と処理文を設定できます。上限は有りません。

◎ SELECT、CASE～文（多分岐制御）

基準となるワードデバイスに格納されている値により、処理を選択できる条件分岐文です。

基本書式	書式説明	書式例	解説
SELECT CASE ワードデバイス	ワードデバイスの値を基準に条件分岐	SELECT CASE DM1000	'DM1000に格納されている値を基準とします
CASE 定数1※3	定数1とワードデバイスの値が等しい時	CASE 1	'DM1000 = 1 のとき
処理文1	処理文1を実行します。	DM2000 = 1000	'DM2000に1000を格納します
CASE 定数2	定数2とワードデバイスの値が等しい時	CASE 2 TO 10	'DM1000 = 2 ～ 10 のとき
処理文2	処理文2を実行します。	DM2000 = 2000	'DM2000に2000を格納します
CASE 定数3	定数3とワードデバイスの値が等しい時	CASE 11, 15, 60	'DM1000 = 11 or 15 or 60 のとき
⋮	⋮	DM2000 = 3000	'DM2000に3000を格納します
CASE ELSE※1	定数とワードデバイスの値が何れも等しくない場合、	CASE IS > 100※	'DM1000が100より大きい時
処理文n※1	処理文nを実行します。	DM2000 = 4000	'DM2000に4000 を格納します
END SELECT	セレクト文は、ここで終わります。	CASE ELSE	'DM1000の値が上記以外 のとき
		DM2000 = 5000	'M2000に5000を格納します
		END SELECT	

※ 半角スペース挿入を表します。

※ マークがあるところは、必ず改行を入れてください。

※1: 省略可能です。

※2: (CASE 定数 (ELSE)) は最大200 個まで記述可能です。

※3: 定数1に文字列や、文字列を扱うデバイス、ビット型デバイスを指定できません。

※定数部には比較演算子を使用した条件式を書くことができます。
その場合、「IS」を用いて比較します。

(例) IS > = 100 (100以上)、IS < > 30 (30以外)

賞味期限の印字データを作成する

現在時刻と設定した賞味期間から賞味期限を求め、印字データを作成します。

例	設定賞味期間	36時間
	製造日時	2005年12月31日18時
	賞味期限	2006年1月2日06時

【使用デバイス】 DM400・・・賞味期間 DM410・・・製造日時
DM430・・・賞味期限

KVスクリプトなら文字列処理も楽々

```
'製造日時データ作成
DM410.T = "製造日時 : " + STR(2000 + CM700) + "年"
DM410.T += STR(CM701) + "月" + STR(CM702) + "日"
DM410.T += RIGHT(DASC(CM703,2),2) + "時" + CHR(13) + CHR(10)

'賞味期限計算
RSEC((SEC(CM700) + DM400 * 60 * 60),DM450)

'賞味期限データ作成
DM430.T = "賞味期限 : " + STR(2000 + DM450) + "年"
DM430.T += STR(DM451) + "月" + STR(DM452) + "日"
DM430.T += RIGHT(DASC(DM453,2),2) + "時" + CHR(13) + CHR(10)
```

解説

◎ 文字列連結演算子

2つの文字列の連結結果を返します。

記号	処理内容
&. +※1	2つの文字列を連結します

※1:「+」は記述内容に応じて算術演算と文字列連結に自動的に判断されます。

◎ STR 数値データ→アスキー文字列変換

10進アスキーコード変換
戻り値 = STR (変換対象デバイス)



タッチパネルディスプレイ
VT3シリーズ

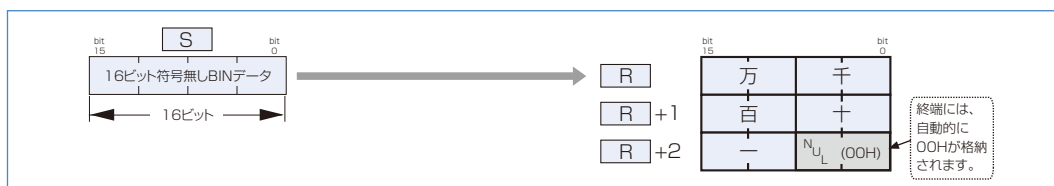
◎ DASC BINデータ→10進アスキー文字列変換

10進アスキーコード変換

戻り値 = **DASC** (変換対象デバイス、変換パラメータ)

[S1] 変換対象デバイスに格納されたバイナリデータを、[S2]で指定された変換パラメータに従い、10進アスキーコード文字列に変換して、結果を[R] 戻り値に格納します。

[S1]: 変換対象デバイスが16ビット符号無しデータ(.U)の場合



◎ RIGHT 右文字列切り出し

右文字列切り出し

戻り値 = **RIGHT** (切り出し対象デバイス、切り出し文字数)

[S] 切り出し対象デバイスで指定されたデバイスを先頭に格納されている文字列を、右端から、
[n] 切り出し文字数で指定された文字数(バイト数)分切り出し、終端コードNUL (00H)を付加して、
結果を[R] 戻り値に格納します。

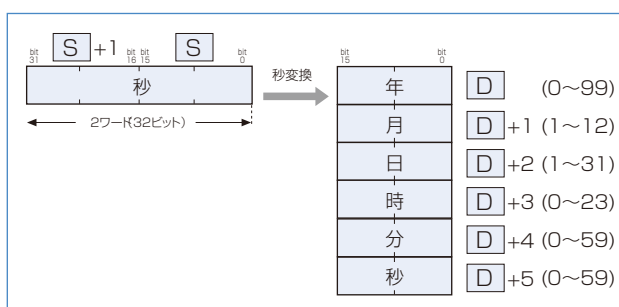
◎ RSEC 秒形式を日時データに変換

秒→日時変換

RSEC ([実行条件,] * 変換元デバイス、変換先デバイス)

[S] 変換元デバイスで指定された、2000年1月1日0時0分0秒を基準日時とした通算秒(32ビットバイナリデータ(符号無し))を日時データに変換し、[D] 変換先デバイスで指定されたデバイスを先頭とする6ワード(96ビット)に格納します。

※ [] は省略可能です。



KVスクリプト活用例 ⑤

加工回数と各目標座標を計算する

測定されたシート長から加工可能枚数と各目標座標を計算します。

例	シート長:1250cm 加工長:160cm 開始座標:3cm
	↓計算
	加工可能枚数:7枚 余り:130cm 座標1:3cm 座標2:163cm...

【使用デバイス】	DM500・・・シート長 DM502・・・加工長 FM0・・・開始座標 MR500・・・計算開始	DM501・・・余りシート長 DM503・・・加工可能枚数 FM2～・・・各目標座標
----------	---	--

KVスクリプトならこんなに簡単

MR500 ↑ 加工回数計算	<pre>' 初期化 FMOV (0,FM2,10000) ' FM2から10000個分初期化します。 DM503 = 0 ' 加工可能枚数を初期化します。 ' 余りシート長コピー DM501 = DM500 ' シート長を余り長にコピーします。 ' 加工回数分繰返し WHILE DM501 >= DM502 ' あまりシート長が加工長以上の場合 DM503 += 1 ' 加工可能枚数を1枚足します。 DM501 -= DM502 ' 余り長から加工長を引きます。 FMO.L: (DM503 * 2) = FM0.L + EM502 * EM503 ' 各目標座標を格納します。 END WHILE</pre>
----------------------	--

解説

◎ WHILE文 (前判定繰返し制御)

繰返し条件が「TRUE (真)」の間、繰返し処理文を実行する制御文です。

基本書式	書式説明	書式例	解説
WHILE ◻<条件式> <処理文> END◻WHILE	条件式が成り立つ時、*1 処理文を実行します。 文頭 (WHILE) に戻ります。	WHILE DM1000 < 100 DM1000 += 5 END WHILE	' DM1000の値が100未満の場合、*2 ' DM1000に5を加算します。 ' 文頭 (WHILE) に戻る

※ ◻ は半角スペース挿入を表します。

※ ◻ マークがあるところは、必ず改行を入れてください。

※ 1: 条件式が成り立たない時、繰返し処理を終了します。

※ 2: DM1000の値が100以上だったら、処理文を実行せずにEND WHILE の次の行から実行します。



タッチパネルディスプレイ
VT3シリーズ

その他、DO～UNTIL・FOR～NEXTが繰り返し制御文として使用できます。
また、BREAK文で繰り返し制御から抜けることが可能です。

書式例	解説
<繰り返し制御文内> IF DM502 = 0 THEN BREAK END IF	DM502が0の場合、 この繰り返し処理を強制的に終了します。 IF分の終わりです。

注意 繰り返し制御文では、必ず繰り返し処理が終了するようにスクリプトを記述してください。繰り返し処理が終了しないと「無限グループ（永久に繰り返し処理を行なう）」となり、プログラム実行時にスキャンタイムオーバーとなります（無限グループは変換時にエラーとして検出されません）。

◎ インデックス修飾とは？

インデックス修飾とは、命令語のオペランドとして指定する各デバイスに、インデックスレジスタ(Z)の値や定数を加算した番号のデバイスを指定する方法です。
加算できる値の範囲は、符号付き32ビット (KV-1000の場合は16ビット) です。
インデックス修飾できるデバイスの種類は、R (DR)、MR、LR、B、T、C、DM、EM、FM、ZF、W、TMです。

◎ インデックスレジスタによる修飾

命令語のオペランドが参照するデバイスを運転中に変更する場合に使用します。
インデックスレジスタによるデバイス修飾の記述方法は「(デバイス番号):Z(01～10)」となります。

例 DM1000:Z01…………… Z01の値が－10の場合、DM990を参照します。
R000:Z05…………… Z05の値が＋16の場合、R100を参照します。

◎ デバイスや式によるインデックス修飾

KVスクリプトでは、インデックスレジスタ以外にもインデックス修飾ができます。
この場合、ラダー変換時に自動的にシステムが予約しているインデックスレジスタ(Z11、Z12)を使用します。

例 DM1000:(DM10+2)=100

全商品、送料無料で

当日出荷

必要な時に、必要な量だけ
在庫不要でトータルコストを削減

PLC・タッチパネルの
最新ソリューションを探せる
www.keyence.co.jp/seigyô



安全に関する注意

商品を安全にお使いいただくため、ご使用前に必ず「取扱説明書」をよくお読みください。

株式会社 キーエンス

技術相談、お問い合わせ

お客様の身近な技術営業が
ダイレクトにサポート

制御システム事業部

盛岡 019-603-0911	松本 0263-36-3911
仙台 022-224-0911	静岡 054-203-7100
山形 023-626-7311	浜松 053-454-0911
郡山 024-933-0911	豊田 0565-25-3211
宇都宮 028-610-8611	安城 0566-71-0011
長岡 0258-38-5311	名古屋 052-971-3911
高崎 027-328-1911	一宮 0586-47-7511
熊谷 048-527-0311	津 059-224-0911
川越 049-240-3211	富山 076-444-1433
浦和 048-832-1711	金沢 076-262-0911
水戸 029-302-0811	滋賀 077-526-8122
柏 04-7165-7011	京都 075-352-0911
幕張 043-296-7511	大阪北 06-6338-1471
神田 03-5825-6211	大阪中央 06-6943-6111
東京 03-5715-6211	堺 072-224-4911
立川 042-529-4911	神戸 078-322-0911
八王子 042-648-1101	岡山 086-224-1911
川崎 044-220-3011	高松 087-834-8911
横浜 045-263-1311	広島 082-261-0911
藤沢 0466-29-0711	北九州 093-511-3911
厚木 046-224-0911	福岡 092-452-8411
長野 026-237-0911	熊本 096-278-8311

制御システムに関するお問合せは

 **0120-423-723**

本社・研究所／制御システム事業部

〒533-8555 大阪市東淀川区東中島1-3-14
Tel 06-6379-1271 Fax 06-6379-1270

海外事業部

〒533-8555 大阪市東淀川区東中島1-3-14
Tel 06-6379-2211 Fax 06-6379-2131

制御5-1030

記載内容は、発売時点での弊社調べであり、
予告なく変更する場合があります。

Copyright© 2008 KEYENCE CORPORATION.
All rights reserved.

1040-1 184-168

PROGRAMMABLE CONTROLLER KV-5000/3000

KV-5000 NEW REALTIME Logic Controller



ネットワーク対応・大容量CPU

基本性能	最大I/O 3096点	プログラム容量 260kステップ	LD/OUT命令 10ns
通信ポート	Ethernet/ FL-net	USB	
内蔵I/O	24点		
システム	構造化 プログラム	KVスクリプト	ダイレクト リフレッシュ

KV-3000 NEW REALTIME Logic Controller



リアルタイム スタンダードCPU

基本性能	最大I/O 3096点	プログラム容量 160kステップ	LD/OUT命令 10ns
通信ポート	USB	シリアル	
内蔵I/O	24点		
システム	構造化 プログラム	KVスクリプト	ダイレクト リフレッシュ

KV-1000



基本性能	最大I/O 3096点	プログラム容量 160kステップ	LD/OUT命令 25ns
通信ポート	USB	シリアル	
内蔵I/O	24点		
システム	構造化 プログラム	KVスクリプト	