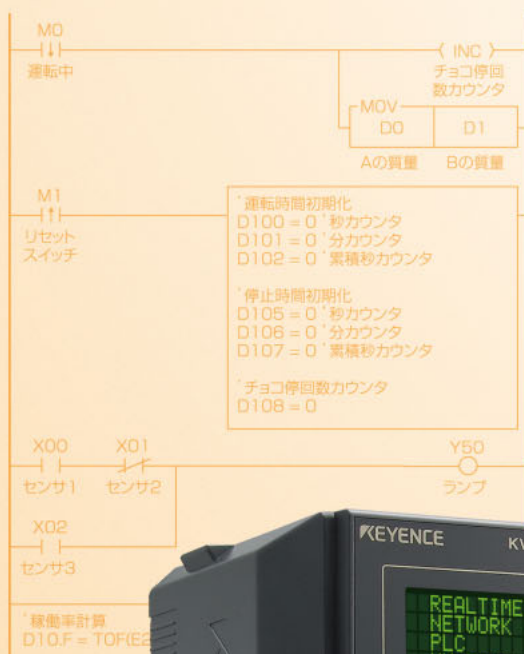


ラダーとスクリプトを融合する

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 1



INDEX

アプリケーション①	数値の四捨五入 - 四則演算 -	2
アプリケーション②	不良率のppm表記 - 浮動小数点演算 -	4
アプリケーション③	測定値に応じたメッセージ表示 - 文字列格納 -	6
アプリケーション④	バーコードの一致判定 - 文字列比較 -	8
アプリケーション⑤	測定値の連続保存 - インデックス修飾 -	10

数値の四捨五入

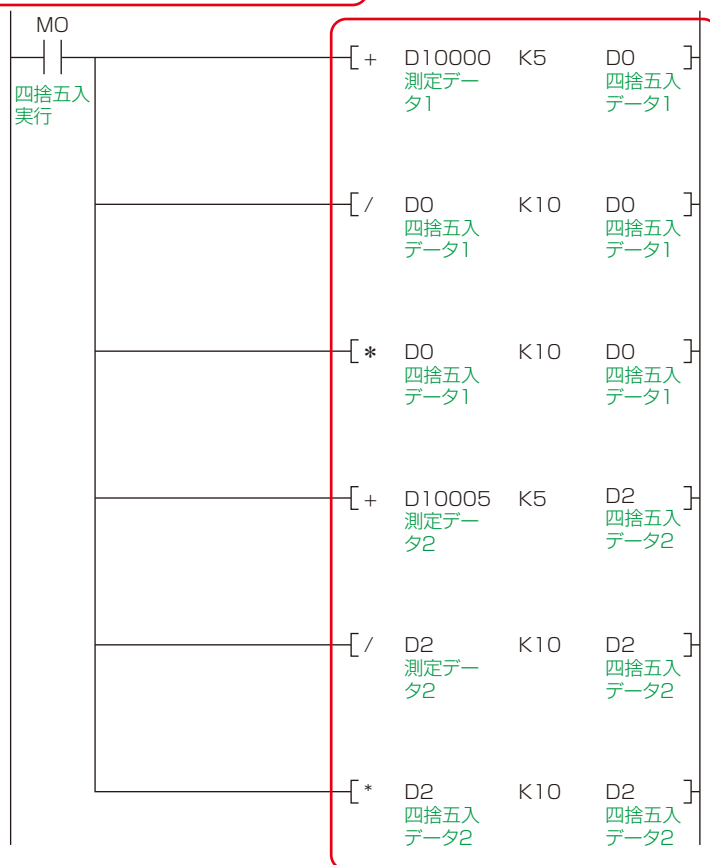
D10000、D10005に取り込んだ測定値（アナログデータ）の下一桁を四捨五入してD0、D2に格納します。

測定値の分解能が高すぎてタッチパネル上の表示が安定しない場合などに有効です。

参考 正の値の場合、下一桁の四捨五入は「5」を足してから下一桁を切り捨てることで実現できます。

●従来のラダーで記述すると…

四捨五入1:D0 = (D10000 + 5) / 10 * 10
四捨五入2:D2 = (D10005 + 5) / 10 * 10



ラダーを見ても数式の内容が一目で把握できないため、行コメント（ステートメント）が必要です。しかも、ラダー変更しておかないと…

記述順や一時格納を意識して記述しないと正しい演算結果は得られません。

●こんな処がラダーの弱点

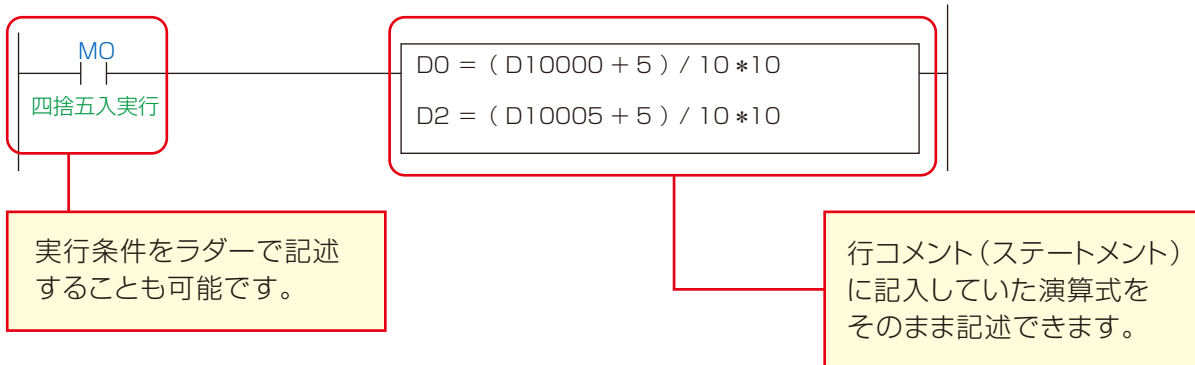
- ・ラダーでの記述が難しい演算処理や文字列処理はプログラムが冗長になってしまい、行コメント（ステートメント）がないとすぐに処理内容が把握できません。
- ・別の担当者が作成したラダーや自分が過去に作成したラダーを変更しようとする、目的の行に到達するまでに時間がかかります。

超高速・高精度
A/D変換ユニット
KV-5000/3000/1000 KV-AD40 (V/G)



タッチパネルディスプレイ
VT3シリーズ

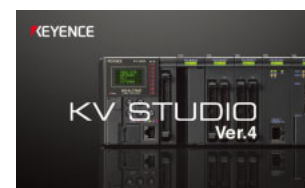
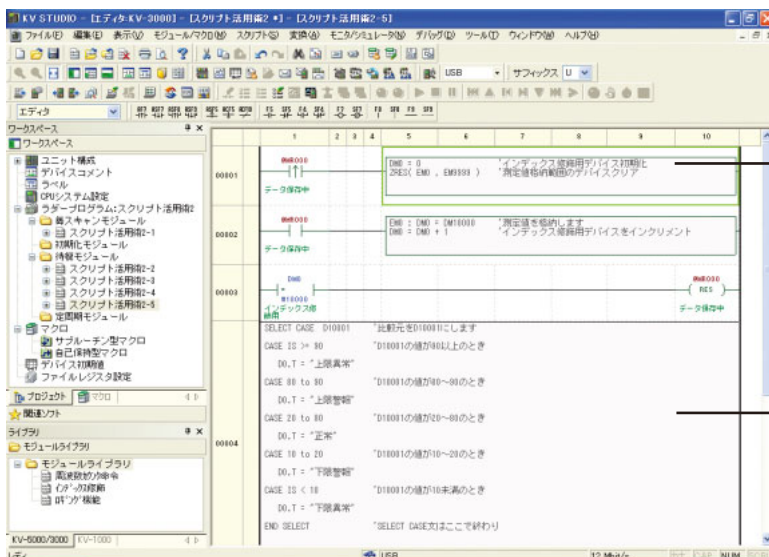
●KVスクリプトならこんなに簡単



●KVスクリプトのメリット

演算式をそのまま記述できるので、処理内容が直感的に把握できます。プログラムの設計時だけでなく、デバッグや変更時にも工数を削減できます。
四則演算をスクリプト言語で記述することによりプログラムがシンプルになります。

解説 ラダーサポートソフトでの編集イメージ



ボックス スクリプト

実行条件をラダーで記述することができるスクリプトです。

ラダーとKVスクリプトは同一プログラム中で混在して使えます。

エリア スクリプト

実行条件を持たない、常時実行型スクリプトです。

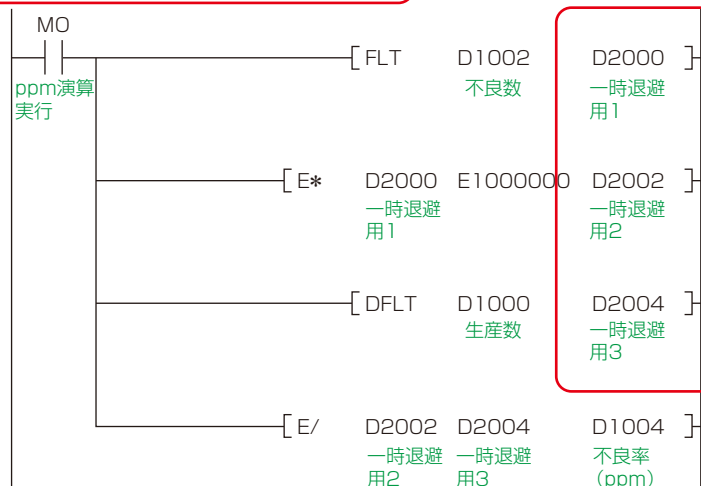
不良率のppm表記

生産数 (D1000) と不良数 (D1002) から不良率をppm単位で計算して、D1004に格納します。

参考 ppmはparts per millionの略で、1 ppmは百万分の1になります。

●従来のラダーで記述すると…

不良率 (ppm) = 不良数 * 1000000 / 生産数
D1004 = D1002 * 1000000 / D1000



行コメント（ステートメント）がないと、演算式の内容がすぐわかりません。

不良数・生産数それぞれを演算の途中経過を一旦、別のデバイスに格納する必要があります。

解説 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 ≤ N ≤ － 1.4E－ 45 N = 0 1.4E－ 45 ≤ N ≤ 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

(例) 符号付き16ビットの演算 A.S = B.S + C.S



●KVスクリプトならこんなに簡単

$D1004.F = TOF(D1002) * 1000000 / D1000$

または

$D1004.F = FLOAT (D1002) * 1000000 / D1000$

KVスクリプトならわずか1行で記述
できます。
通常の小数点表記を含んだ表記に
加えて指数表記も可能です。
(例) 1000000 → 1.0E + 6

●KVスクリプトのメリット

演算式をそのまま記述できるので、処理内容が直感的に把握できます。プログラムの設計時だけでなく、デバッグや変更時にも工数を削減できます。
浮動小数点演算をスクリプト言語で記述することによりプログラムがシンプルになります。

解説 型宣言 TYPE

型宣言をおこなうと、演算式入力時のサフィックス記述を省略することができます。
型宣言の記述規則は以下のとおりです。

TYPE記述	解 説
TYPE DMO.F	DMOのみを浮動小数型として扱う。
TYPE DMO ~ DM10.F	DM0~DM10を浮動小数型として扱う。
TYPE DM.F	全てのDMを浮動小数型として扱う。

型宣言は、スクリプトで演算記述をおこなう前に記述する必要があります。
また、宣言した型は、宣言したスクリプト枠内でのみ有効です。

●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

測定値に応じたメッセージ表示

D10001に取り込んだタンクの液量に従い、ディスプレイにメッセージ(D0～)を表示します。

例) 液量が90%以上の場合、「上限異常」と表示します。

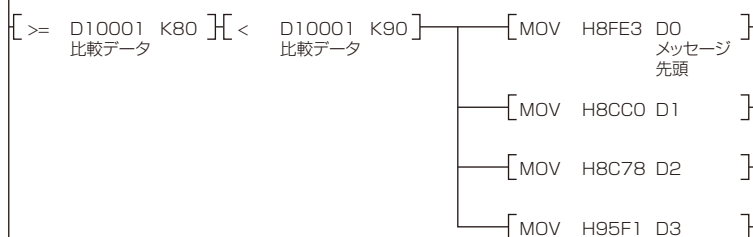
●従来のラダーで記述すると...

90以上のとき「上限異常」

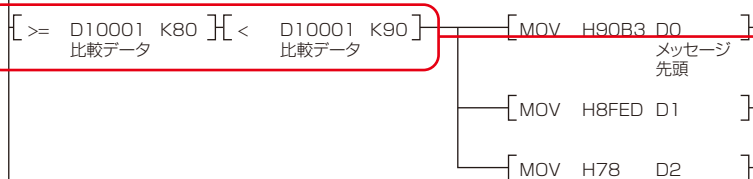


全角文字はそのまま記述できないため、文字コードを一文字分ずつ調べて打ち込む必要があります。

80～90のとき「上限警報」

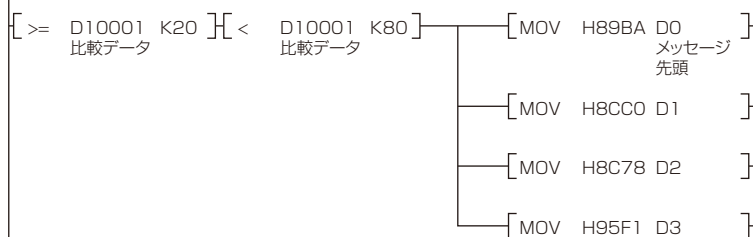


20～80のとき「正常」



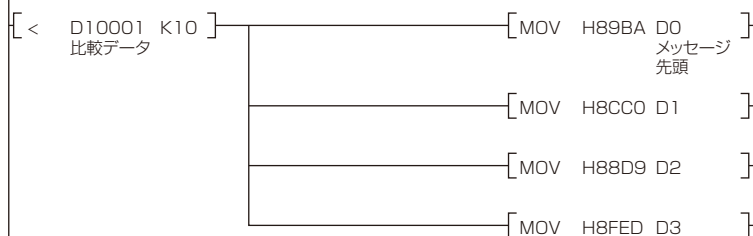
条件の大小関係が直感的にわかりづらいです。

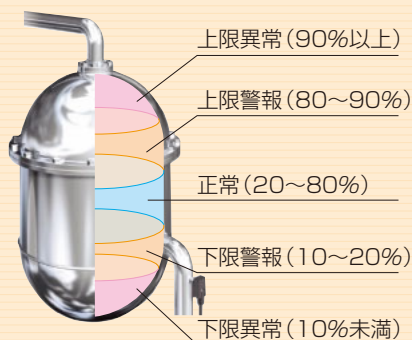
10～20のとき「下限警報」



縦に長くなってしまうので、行コメント(ステートメント)がないと、条件分岐のラダーがどこまで続いているのかわかりません。

10未満のとき「下限異常」





タッチパネルディスプレイ
VT3シリーズ

●KVスクリプトならこんなに簡単

SELECT CASE D10001	'比較元をD10001にします'
CASE IS >= 90	'D10001の値が90以上のとき'
DO.T = "上限異常"	
CASE 80 to 90	'D10001の値が80~90のとき'
DO.T = "上限警報"	
CASE 20 to 80	'D10001の値が20~80のとき'
DO.T = "正常"	
CASE 10 to 20	'D10001の値が10~20のとき'
DO.T = "下限警報"	
CASE IS < 10	'D10001の値が10未満のとき'
DO.T = "下限異常"	
END SELECT	'SELECT CASE文はここで終わり'

文字列をそのまま記述
できます。

条件の大小関係が直感的に
把握できます。

条件分岐に関するスクリプトの
範囲がわかりやすいです。

●KVスクリプトのメリット

スクリプトでは文字列がそのまま記述できるので、プログラム作成時や変更時に工数が削減できます。また、ディスプレイに表示する前にミスがないかを簡単にチェックできるため、無駄なミスも防げます。

解説 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。

SELECT CASE ~	
基本書式	書式説明
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
CASE 定数1 ^{※3}	定数1とワードデバイスの値が等しい時
処理文1	処理文1を実行します。
CASE 定数2	定数2とワードデバイスの値が等しい時
処理文2	処理文2を実行します。
CASE 定数3	定数3とワードデバイスの値が等しい時 ^{※2}
⋮	⋮
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、
処理文n ^{※1}	処理文nを実行します。
END SELECT	セレクト文は、ここで終わります。

※1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: 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。

書式例	解説
SELECT CASE DM1000	'DM1000に格納されている値を基準とします'
CASE 1	'DM1000 = 1 のとき'
DM2000 = 1000	'DM2000に1000を格納します'
CASE 2 TO 10	'DM1000 = 2~10のとき'
DM2000 = 2000	'DM2000に2000を格納します'
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 IS > 100 [※]	'DM1000が100より大きい時'
DM2000 = 4000	'DM2000に4000を格納します'
CASE ELSE	'DM1000の値が上記以外のとき'
DM2000 = 5000	'DM2000に5000を格納します'
END SELECT	

※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。

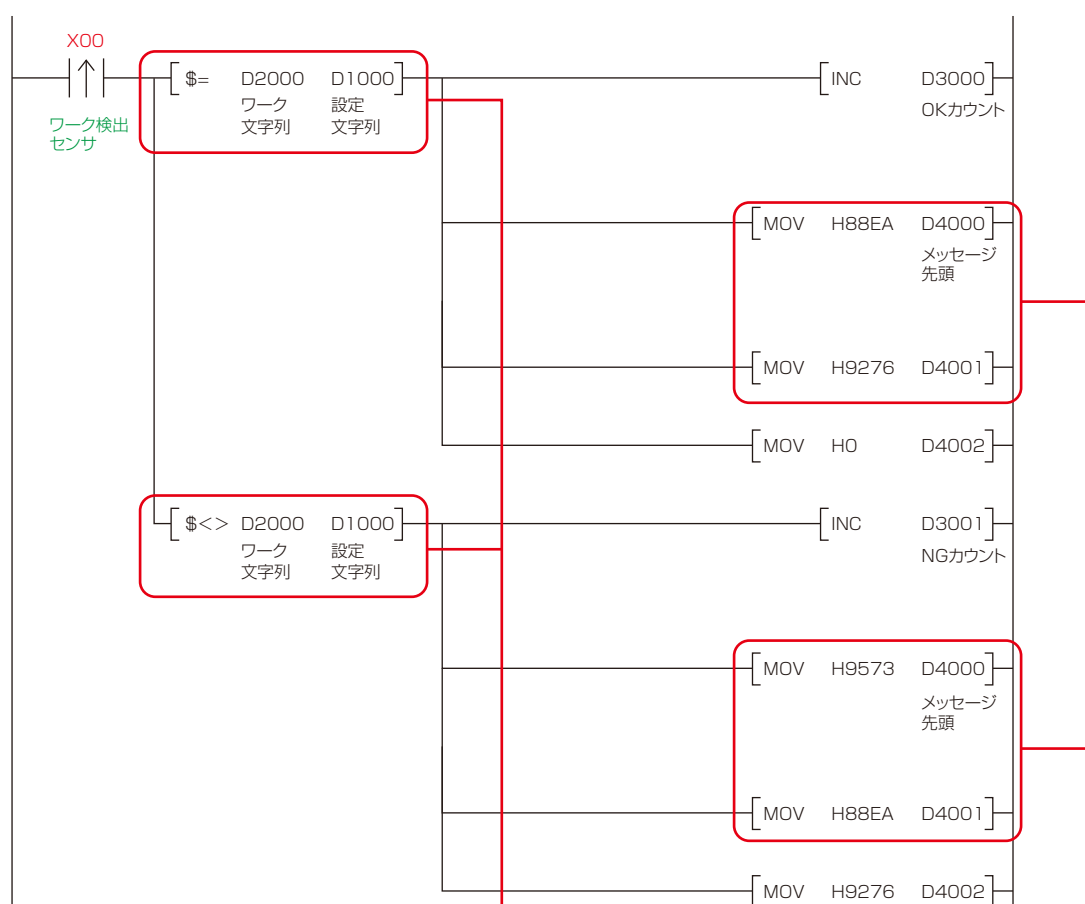
(例) IS >= 100(100以上)、IS <> 30(30以外)

アプリケーション④ ー文字列比較ー

バーコードの一致判定

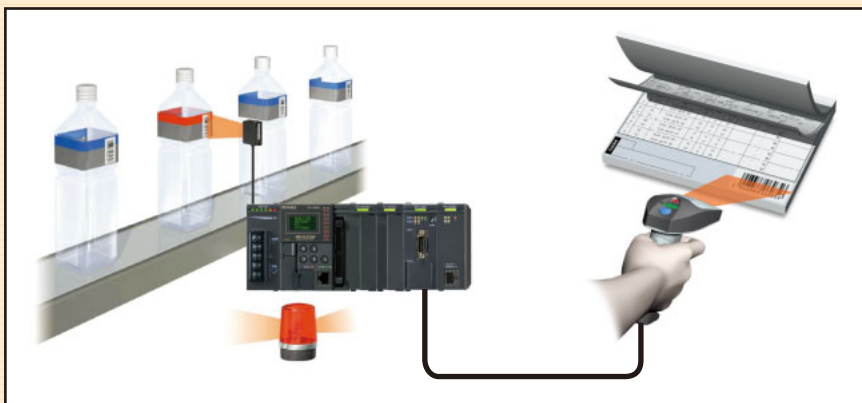
ハンディバーコードリーダで読み込んだ文字列(D1000～)とワークに貼り付けられているバーコードの文字列(D2000～)を比較します。
一致/不一致をディスプレイに表示(D4000～)して、それぞれの個数をカウント(D3000,D3001)します。

●従来のラダーで記述すると…

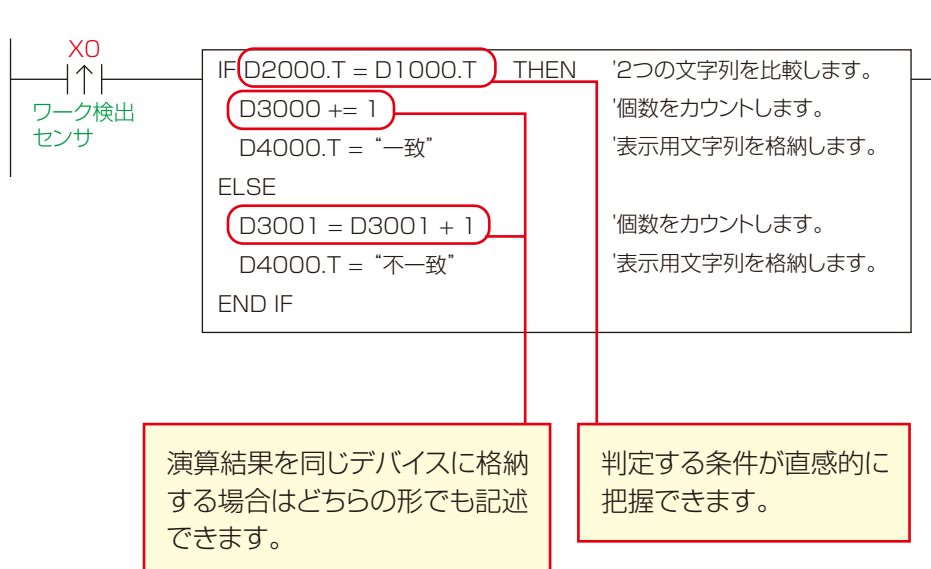


比較演算命令でそれぞれの条件を記述する必要があります。

全角文字はそのまま記述できないため、文字コードを一文字分ずつ調べて打ち込む必要があります。



●KVスクリプトならこんなにすっきり



●KVスクリプトのメリット

文字列の比較や代入処理も直感的に把握しやすく記述できます。プログラムの設計時だけでなく、デバッグや変更時にも工数を削減できます。
また、「～以外の場合」の条件を簡単に記述できます。

解説 IF～ELSE文 (条件分岐)

1つの条件式で処理方法を2通りに分けることのできる、簡単な条件分岐文です。

IF ～ THEN ～ ELSE ～ END IF		
基本書式		書式説明
IF ◻ <条件式> ◻ THEN	↙	もし条件式が成り立つ時、
<処理文1>	↙	処理文1を実行します。
ELSE	↙	条件式が成り立たない時、
<処理文2>	↙	処理文2を実行します。
END IF	↙	IF文はここまで

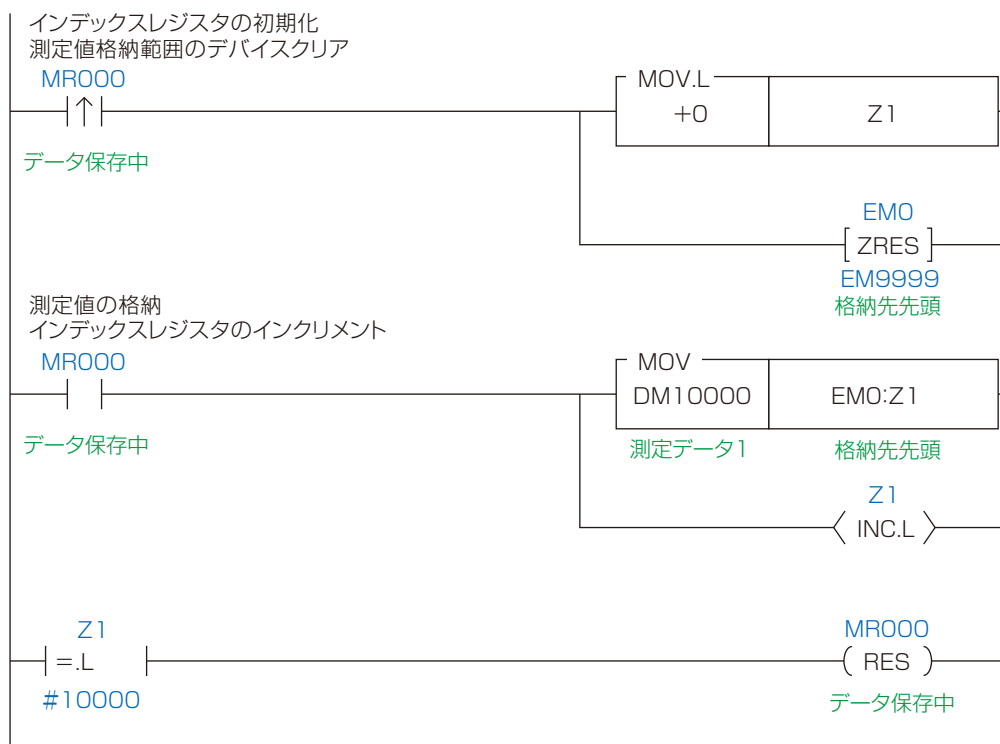
※ ◻ は半角スペース挿入を表します。

↙ マークがあるところは、必ず改行を入れてください。

測定値の連続保存

DM10000に取り込んだ測定値（アナログデータ）を1スキャンごとにEMOから連続で格納します。
格納するデバイスを間接指定するために、インデックス修飾を使用します。

●KV-5000/3000にてラダーのみで記述すると…



解説 ■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)、B、MR、LR、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)

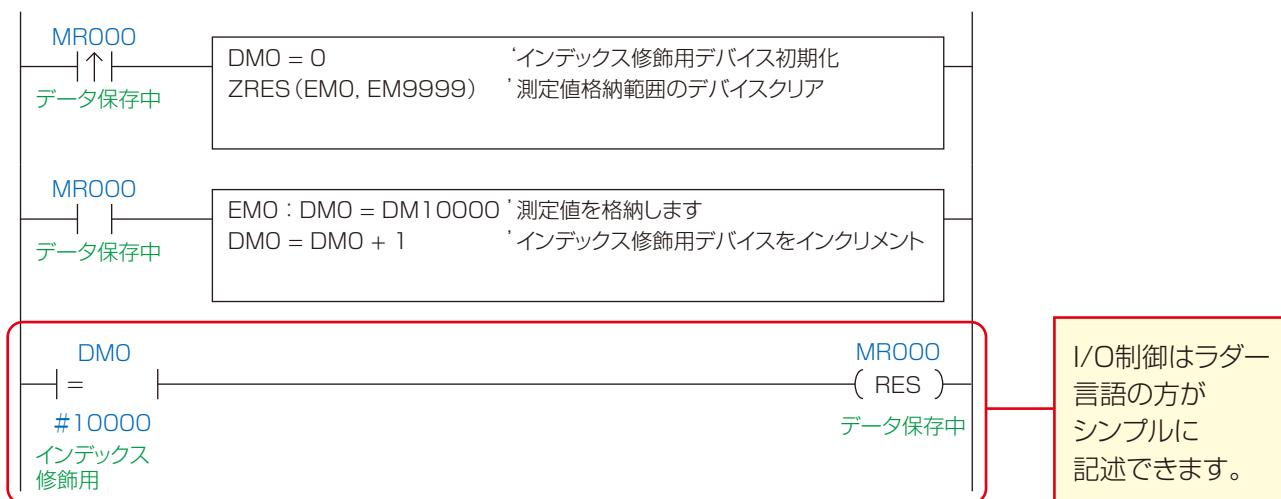
となります。

(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。



●KVスクリプトを併用すればすっきり



●KVスクリプトのメリット

ラダーとスクリプトを同一プログラム中で混在して使えますので、処理内容に合わせて言語を使い分けることができます。

解説

■デバイスや式によるインデックス修飾

スクリプトでは、インデックスレジスタ以外で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.

1040-1 084-167

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	