

DM0:秒カウンタ DM1:分カウンタ DM2:累積秒カウンタ

DM1 = DM2 / 60 '分を求める

DM0 = DM2 MOD 60 '秒を求める

DM1000~DM1012:印字データ

'CM700:カレンダータイマ(年)

'CM701:カレンダータイマ(月)

'CM702:カレンダータイマ(日)

DM1000.T = "製造年月日:200"+STR(CM700)+"年"+STR(CM701)+"月"+STR(CM702)+"日"

'DM10:数点付き演算結果

'DM10:四捨五入結果

IF DM0.F > 0 THEN

DM10.L = DM0.F + 0.5

ELSE

DM10.L = DM0.F - 0.5

END IF

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。

'X座標:DM200 Y座標:DM202 角度:DM204 長さ:DM206

'角度計算

DM204.F = DEG(ATAN(TOF(DM200.L) / DM202.L))

'長さ計算

DM206.F = SQR( (DM200.L)^2 + (DM202.L)^2 )

測定データの標準偏差を計算します。

'平均値の計算

'平均値の計算

@EM0.F = FLOAT(AVG(DM0000.U,DM1000.U))

'分散の計算

@EM6.F = 0

FOR Z = 0 TO DM0000.U-1 STEP 1

@EM6.F = @EM6.F + (TOF(DM0000.U-1) - @EM0.F)^2

NEXT

'標準偏差の計算

@EM4.F = SQR(@EM6.F)

\*\*\* 解説 \*\*\*

DM0:平均値(移動平均法)を用いて平均した結果

DM1:平均データ個数

'(ラダーにて直前の何個のデータを平均値計算に使用するかを設定)

DM1000:A/D変換値

\*\*\* FIFOテーブル解説 \*\*\*

@DM0:現在格納されているデータ個数

'(FIFOR・FIFOW命令を実行すると自動格納)

@DM1:格納できるデータ個数最大値

'(ラダーにてFIFOR・FIFOW命令実行前に設定)

@DM2~:データ(デバイス番号の若い順に新しいデータ)

'平均データ個数 DM1をFIFOテーブルの @DM1に格納します。

@DM1 = DM1

'FIFOテーブルに格納されたA/D変換値が平均データ個数を超えれば、

'古いデータを読み出します。

IF @DM0 >= @DM1 THEN

FIFOR( @DM0 , @EM0 )

END IF

DM1000に格納されたA/D変換値を @DM0からのFIFOテーブルに格納します。

FIFOW( DM1000 , @DM0 )

'データの平均値を求めます。

DM0 = AVG( @DM2 , @DM1 )

'ゼロサプレス付き、符号有無、小数点以下桁数など桁数が可変な

'文字列データから数値部分のみを抽出して数値データに変換します。

'ダメーデータ(必要に応じてコメント化してください)

DM1000.T = "-12345.12,+1.234,0.000056"

'初期化

@DM0 = 0 '検索開始位置

@DM1 = 0 '数値格納位置

@DM2 = LEN(DM1000.T) '文字列長

'含まれる数値を文字列の最後まで繰り返し数値データに変換します。

IF @DM2 <> 0 THEN

DO

@DM10 = SFIND(DM1000.T,".",@DM0) 'カンマ(.)の位置検出

IF @DM10 = 65535 THEN

@DM20.T = SMID(DM1000.T,@DM0,@DM10 - @DM0) '数値部分の切り出し

IF @DM20.T = "E5535" THEN '数値データへの変換

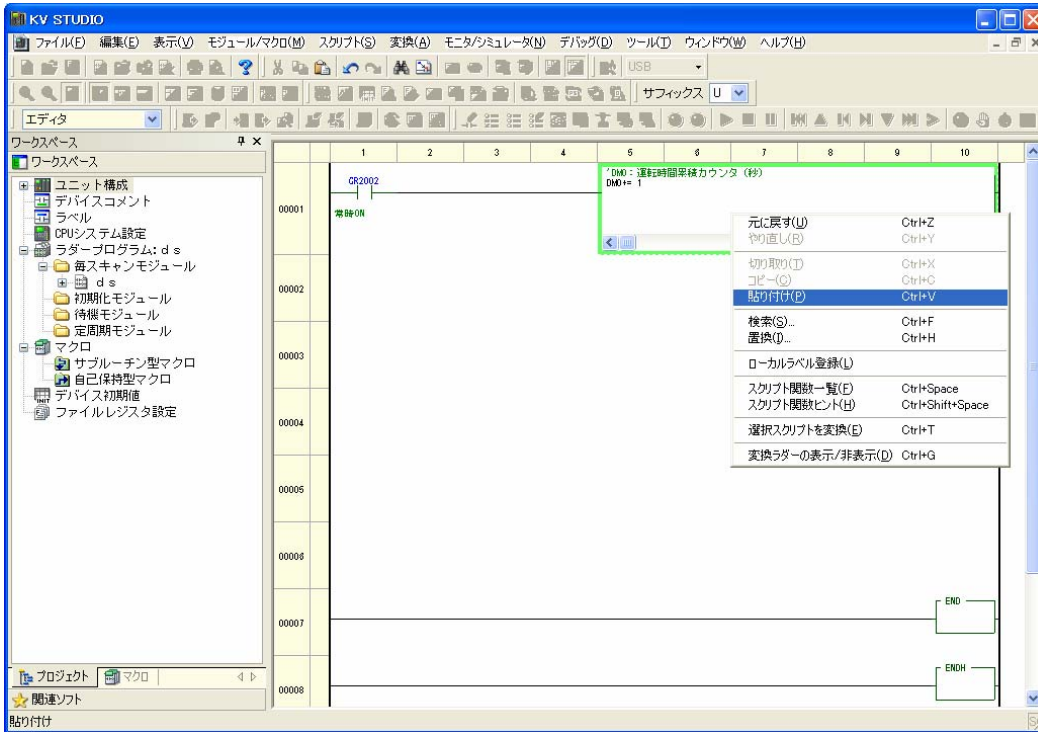
DM1000.T = DM1000.T + @DM20.T + "E5535" THEN '数値データへの変換

# 「コピペ」で活用！ 即使える！ KVスクリプト サンプルプログラム集

## 実戦で活用できる全10種のアプリケーション

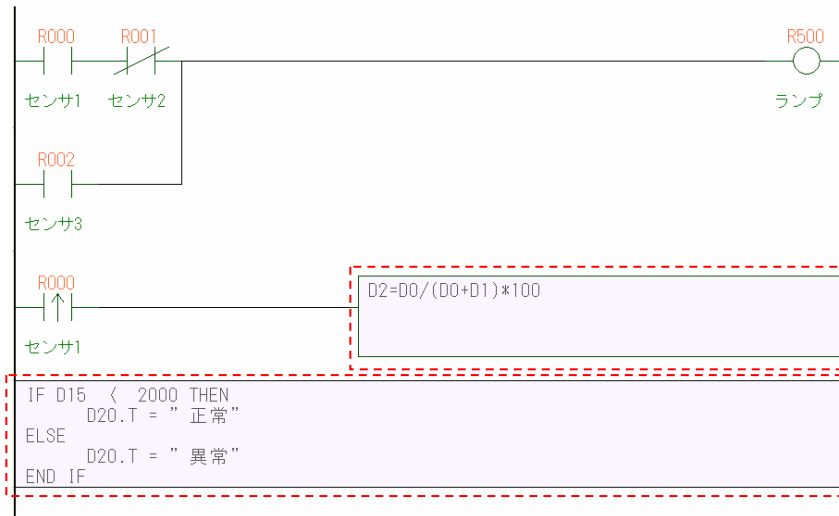


## ② KV STUDIOに貼り付けます



ラダーサポートソフト  
KV STUDIO

貼り付け先として、KVスクリプトのエリアを用意してください。



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。

アプリケーション①

稼働時間・チョコ停時間の計測  
～割り算の余りの求め方～

KVスクリプトで割り算をする場合に余りを求める演算子は「MOD」です。

◇例えば、累積秒数から何分何秒かを求めるには・・・



'DM0:秒カウンタ DM1:分カウンタ DM2:累積秒カウンタ  
DM1 = DM2 / 60 '分を求める  
DM0 = DM2 MOD 60 '秒を求める

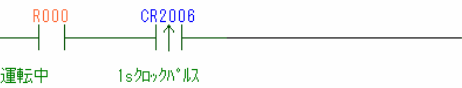


装置の運転入力で運転時間と停止時間、稼働率を算出するプロジェクト例

### 運転時間のカウント

### 運転中入力のON時間を1秒クロックでカウントする。


下記プログラム例ではDM0を16ビット符号なしデータとして扱っているため、計測時間は最大65535秒（約18時間）になります。  
より長時間の計測をおこなう場合は、デバイスにサフィックス.Dを付加して（DM0.Dと記述する）、32ビット符号なしデータとして扱えば最大4294967295秒（約100万時間）まで計測可能です。



DM0: 運転時間累積カウンタ(秒)  
DM0 += 1

### 秒単位でカウントした運転時間を時分秒に分離してDMに格納する。

32ビット符号なしデータで扱う場合は各々のデバイスにサフィックス.Dを付加してください（例 DM100.D = DM0.D / 3600）

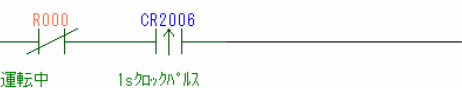


'DM0: 運転時間累積カウンタ(秒)、DM100: 運転時間(時)、DM110: 運転時間(分)、DM120: 運転時間(秒)  
DM100 = DM0 / 3600 '時を求める  
DM110 = (DM0 MOD 3600) / 60 '分を求める  
DM120 = DM0 MOD 60 '秒を求める

### 停止時間のカウント


### 運転中入力のOFF時間を1秒クロックでカウントする。

長時間計測をおこなう場合は同様にデバイスにサフィックス.Dを付加してください。



DM10: 停止時間累積カウンタ(秒)  
DM10 += 1

### 秒単位でカウントした停止時間を時分秒に分離してDMに格納する。




'DM10: 停止時間累積カウンタ(秒)、DM200: 停止時間(時)、DM210: 停止時間(分)、DM220: 停止時間(秒)  
DM200 = DM10 / 3600 '時を求める  
DM210 = (DM10 MOD 3600) / 60 '分を求める  
DM220 = DM10 MOD 60 '秒を求める

### 稼働率(%)を演算する。

稼働率 = 運転時間 / (運転時間 + 停止時間)  
小数点以下の演算結果になるため浮動小数点型実数で演算することが必要です。

### TOFについて  
TOF(浮動小数点型実数) データを浮動小数点型実数データに変換する関数です。  
演算をおこなう前に運転時間、停止時間を実数データに変換しておきます。

### サフィックス.Fについて  
.Fはデバイスデータを浮動小数点型実数データとして扱うためのサフィックスです。



'DM300: 稼働率(%)  
'DM20.F: 累積運転時間(実数)、DM30.F: 累積停止時間(実数)、DM40.F: 稼働率(実数)  
DM20.F = TOF(DM0)  
DM30.F = TOF(DM10)  
DM40.F = (DM20.F / (DM20.F + DM30.F)) \* 100  
DM300 = TOU(DM40.F)

### TOUについて  
TOU(浮動小数点型実数データ)をバイナリ(整数) データに変換する関数です。

## アプリケーション②

### ロットNo.やシリアルNo.を文字列データで出力 ～数値を文字列データに変換する方法～

KVスクリプトで数値データをアスキー文字列に変換する関数は「STR」です。



◇例えば、製造年月日のデータをプリンタに印字出力するときは・・・

```
'DM1000～DM1012: 印字データ
'CM700: カレンダータイマ(年); CM701: カレンダータイマ(月); CM702: カレンダータイマ(日)
DM1000.T = "製造年月日: 200"+STR(CM700)+"年"+STR(CM701)+"月"+STR(CM702)+"日"
```

### 製造年月日、賞味期限をアスキー文字列に変換するプロジェクト例

### 製造年月日と賞味期限をシフトJIS+アスキーコードで出力するプロジェクト例

### 製造年月日データの出力

```
### 出力データ例
### 2007年9月10日
### 2010年12月1日
### 2099年12月31日
```

CPUユニットのカレンダータイマ現在値はコントロールメモリCMに格納されます。  
データは10進数で格納されます。また、年は下2桁がCM700に格納されます。

例)  
2007年→10進数で"7"を格納  
2010年→10進数で"10"を格納  
よって2007年や2010年と出力するときは"200"または"20"を付加する必要があります。  
STRは数値データをアスキー文字列に変換する関数です。

```
'DM1000～DM1012: 出力データ格納先
'CM700: カレンダータイマ(年)西暦の下2桁
'CM701: カレンダータイマ(月)
'CM702: カレンダータイマ(日)
IF CM700 < 10 THEN
  DM1000.T = "製造年月日: 200"+STR(CM700)+"年"+STR(CM701)+"月"+STR(CM702)+"日"
ELSE
  DM1000.T = "製造年月日: 20"+STR(CM700)+"年"+STR(CM701)+"月"+STR(CM702)+"日"
END IF
```

### 賞味期限の計算

SECは年、月、日、時、分、秒の順に10進数で連続するデバイスに格納されている  
カレンダータイマのデータを2000年1月1日0時0分0秒を基準とする秒の10進数に  
変換する関数です。

### 現在の日付の10日後（10日\*24時間\*3600秒）を賞味期限として計算する。

RSECは2000年1月1日0時0分0秒を基準とする秒の10進数データを  
年、月、日、時、分、秒の順に10進数で連続するデバイスのカレンダータイマ  
のデータに変換して格納する関数です。

```
EM0.D = SEC(CM700)
EM100.D = EM0.D + 10*24*3600
RSEC(EM100.D, EM200)

'EM0: カレンダータイマ現在値(秒形式)
'EM100: 賞味期限(秒形式)
'EM200: 賞味期限(年)
'EM201: 賞味期限(月)
'EM202: 賞味期限(日)
'EM203: 賞味期限(時)
'EM204: 賞味期限(分)
'EM205: 賞味期限(秒)
```

### 賞味期限データの出力

```
'DM2000～DM2011: 出力データ格納先
IF EM200 < 10 THEN
  DM2000.T = "賞味期限: 200"+STR(EM200)+"年"+STR(EM201)+"月"+STR(EM202)+"日"
ELSE
  DM2000.T = "賞味期限: 20"+STR(EM200)+"年"+STR(EM201)+"月"+STR(EM202)+"日"
END IF
```

## アプリケーション③

### 数値データ、文字データの抽出方法 ～計測データ、バーコードのデータを抽出する方法～

KVスクリプトで特定の文字を検出する関数は「SFIND」です。

◇例えば、カンマ(,)を検出する場合は…



@DM10 = SFIND(DM1000.T,"",@DM0)'カンマ(,)の位置検出



### 文字列データを抽出して数値データに変換するプロジェクト例



'ゼロサプレス付き、符号有無、小数点以下桁数など桁数が可変な  
'文字列データから数値部分のみを抽出して数値データに変換します。

'ダミーデータ(必要に応じてコメント化してください)  
DM1000.T = "-12345,12,+1.234,0.000056"

'初期化

@DM0 = 0

'検索開始位置

@DM1 = 0

'数値格納位置

@DM2 = LEN(DM1000.T)

'文字列長

'含まれる数値を文字列の最後まで繰り返し数値データに変換します。

IF @DM2 <> 0 THEN

DO

@DM10 = SFIND(DM1000.T,"",@DM0)

'カンマ(,)の位置検出

IF @DM10 = 65535 THEN

@DM10 = @DM2

'最後尾の場合の処理

END IF

@DM20.T = SMID(DM1000.T,@DM0,@DM10 - @DM0)

'数値部分の切り出し

IF FIND(@DM20.T,".") = 65535 THEN

'数値データへの変換

DM0.L:@DM1 = RDASC(@DM20.T)

'整数データの場合

ELSE

DM0.F:@DM1 = RFASC(@DM20.T)

'実数データの場合

END IF

@DM0 = @DM10 + 1

'検索開始位置の加算

@DM1 += 2

'数値格納位置の加算

UNTIL @DM0 > @DM2

END IF

## アプリケーション④

### ソーティング

### ～収集データを昇順、降順で並び替える（ソーティング）方法～

KVスクリプトで制御文として「IF文」が使用できます。

#### 収集データを昇順でソーティングするプロジェクト例



```
***** ソーティング(昇順) *****  
***** 初期設定  
IF LDP(MR0) THEN  
  
  'ソーティング前デバイス : DM0～  
  'ソーティング後デバイス : DM10～  
  'デバイス数 : DM100  
  
  DM100 = 5  
  
  ADRSET(DM0,@DM0.D)  
  ADRSET(DM10,@DM2.D)  
  Z01 = DM100 - 1  
  Z02 = 0  
  FMOV(0*@DM2,DM100) 'ソーティング後デバイスクリア  
  BMOV(*@DM0,@DM10,DM100) '作業領域へコピー  
  IF *@DM0 = @DM10 THEN 'ソーティング前デバイス番号の検索  
    @DM4 = SER(*@DM0,@DM10,#1)  
  END IF  
END IF  
  
***** 並べ替え処理  
IF MR0 = ON THEN  
  
  IF MAX(@DM10,DM100) >= 1 THEN '最大値が0以外であることの確認  
    *@DM2:Z01 = MAX(@DM10,DM100) '最大値を作業領域 先頭へコピー  
    IF *@DM2:Z01 >= 1 THEN '作業領域が0以外であることの確認  
      Z02 = SER(*@DM2:Z01,@DM10,DM100) - @DM4 'ソーティング前領域で最大値があるデバイス位置を確認  
      @DM10:Z02 = 0 'ソーティング前領域の最大値を0に書き換え  
      DEC(Z01) '作業領域の先頭位置を1つ進める。  
    END IF  
  END IF  
END IF
```

#### 収集データを降順でソーティングするプロジェクト例



```
***** ソーティング(降順) *****  
***** 初期設定  
IF LDP(MR1) THEN  
  
  'ソーティング前デバイス : DM0～  
  'ソーティング後デバイス : DM10～  
  'デバイス数 : DM100  
  
  DM100 = 5  
  
  ADRSET(DM0,@TM0.D)  
  ADRSET(DM10,@TM2.D)  
  Z01 = 0  
  Z02 = 0  
  FMOV(0*@TM2,DM100) 'ソーティング後デバイスクリア  
  BMOV(*@TM0,@TM10,DM100) '作業領域へコピー  
  IF *@TM0 = @TM10 THEN 'ソーティング前デバイス番号の検索  
    @TM4 = SER(*@TM0,@TM10,#1)  
  END IF  
END IF  
  
***** 並べ替え処理  
IF MR1 = ON THEN  
  
  IF MAX(@TM10,DM100) >= 1 THEN '最大値が0以外であることの確認  
    *@TM2:Z01 = MAX(@TM10,DM100) '最大値を作業領域 後端へコピー  
    IF *@TM2:Z01 >= 1 THEN '作業領域が0以外であることの確認  
      Z02 = SER(*@TM2:Z01,@TM10,DM100) - @TM4 'ソーティング前領域で最大値があるデバイス位置を確認  
      @TM10:Z02 = 0 'ソーティング前領域の最大値を0に書き換え  
      INC(Z01) '作業領域の後端位置を1つ進める。  
    END IF  
  END IF  
END IF
```

## アプリケーション⑤

### 四捨五入 ～数値の下1桁を四捨五入する方法～

外部機器のアナログデータがばらつく場合などに有効です。

#### 計測データを四捨五入するプロジェクト例

MR000  
↑  
四捨五入演算

```
'DM0 : 小数点付き演算結果  
'DM10: 四捨五入結果  
IF DM0.F > 0 THEN  
  DM10.L = DM0.F + 0.5  
ELSE  
  DM10.L = DM0.F - 0.5  
END
```

## アプリケーション⑥

### X-Y座標から極座標（長さ・角度）への変換 ～三角関数を使用した座標変換の方法～

KVスクリプトでは三角関数も使用できます。



X-Y座標から極座標（長さ・角度）に簡単に変換することが可能です。

#### X-Y座標から極座標（長さ・角度）へ変換するプロジェクト例

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



## アプリケーション⑦

### 表示レンジの変更

### ～アナログデータの計測データのレンジへの変換方法～

KVスクリプトで条件分岐する制御文は「SELECT～CASE」です。

#### アナログ信号をスケーリングしてレンジ変更するプロジェクト例



'変位計などから入力されるアナログ信号を、  
'変位の最小値と最大値でスケーリングします。

'ダミーデータ(必要に応じて値を変更してください。)

@DM0 = 5000 '変位計最小値

@DM1 = 12000 '変位計最大値

@DM2 = 6000 '変位計最小値時のAD変換値

@DM3 = 14000 '変位計最大値時のAD変換値

'ch0のAD変換値(DM10101)の値で条件分岐します。

SELECT CASE DM10101.S

'変位計最小値時のAD変換値(下限値)以下のデータは-30000とします。

CASE IS < @DM2.S

DM1000.S = -30000

'変位計最大値時のAD変換値(上限値)以上のデータは30000とします。

CASE IS > @DM3.S

DM1000.S = 30000

'変位計最小値時のAD変換値(下限値)以上、

'変位計最大値時のAD変換値(上限値)以下の場合、スケーリングします。

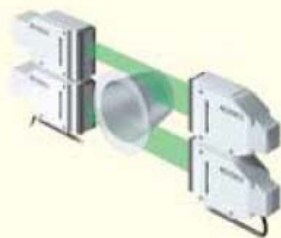
CASE ELSE

DM1000.S = (@DM1.S-@DM0.S)\*(DM10101.S-@DM2.S)/(@DM3.S-@DM2.S)+@DM0.S

END SELECT



ハンダペースト厚み測定



樹脂容器寸法測定



シュリンク温度管理



HDD振れ測定

## アプリケーション⑧

### 平均値、標準偏差の計算

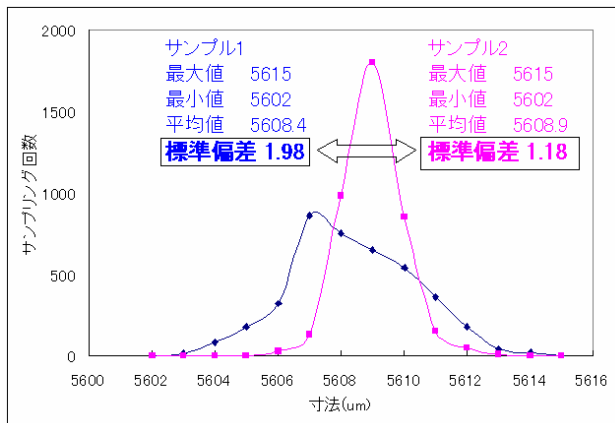
#### ～計測データの平均値、標準偏差を計算する方法～

KVスクリプトで数値データの平均値を計算する関数は「AVG」です。

◇例えば、数値データの平均値を計算するときは・・・



EM0.F = FLOAT(AVG(DM0000.U,DM1000.U))



### 平均値、標準偏差を計算するプロジェクト例

###平均値・分散・標準偏差を求めるプロジェクト例

###動作説明①

収集データの格納場所は、DM0000からとします。  
収集データ数は、DM1000に格納します。

分散 =  $\sum [(要素k - 平均値)^2] / n$   
標準偏差 =  $\sqrt{\text{分散}}$

###動作説明②

@MR0がONになった立上りで計算を実施します。

@MR000

↑↑

標準偏差実行



'測定データの平均値・分散・標準偏差を計算します。

'平均値:@EM0 分散:@EM2 標準偏差:@EM4

'平均値の計算

@EM0.F = FLOAT(AVG(DM0000.U,DM1000.U))

'分散の計算

@EM6.F = 0

FOR Z1 = 0 TO DM1000-1 STEP 1

@EM6.F = @EM6.F + (TOF(DM0000:Z1) - @EM0.F)^2

NEXT

@EM2.F = @EM6.F / (TOF(DM1000))

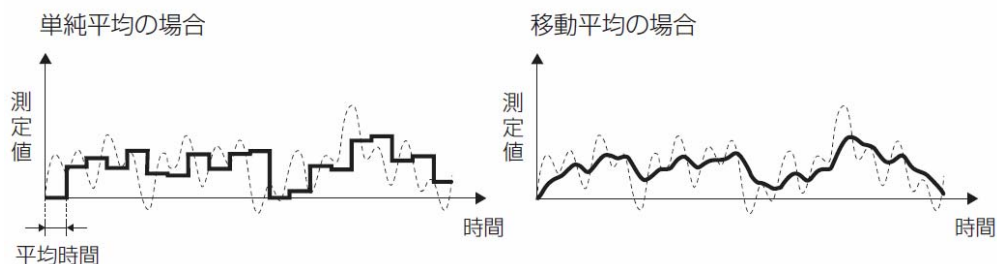
'標準偏差の計算

@EM4.F = ROOT(@EM2.F)

## アプリケーション⑨

### 移動平均の計算 ～時間を設定して移動平均を計算する方法～

KVスクリプトで「FIFO」関数を使うと移動平均を簡単に計算できます。



### 移動平均で平均値を計算するプロジェクト例

@T0  
↑



\*\*\* デバイス解説 \*\*\*

'DM0:平均値 (移動平均法を用いて平均した結果を格納)

'DM1:平均データ個数

' (ラダーにて直前の何個のデータを平均値計算に使用するかを設定)

'DM1000:A/D変換値

\*\*\* FIFOテーブル解説 \*\*\*

'@DM0:現在格納されているデータ個数

' (FIFOR・FIFOW命令を実行すると自動格納)

'@DM1:格納できるデータ個数最大値

' (ラダーにてFIFOR・FIFOW命令実行前に設定)

'@DM2~:データ (デバイス番号の若い順に新しいデータ)

'平均データ個数 DM1をFIFOテーブルの @DM1に格納します。

@DM1 = DM1

'FIFOテーブルに格納されたA/D変換値が平均データ個数を超えれば、

'古いデータを読み出します。

IF @DM0 >= @DM1 THEN

    FIFOR( @DM0 , @DM0 )

END IF

'DM1000に格納されたA/D変換値を @DM0からのFIFOテーブルに格納します。  
FIFOW( DM1000 , @DM0 )

'データの平均値を求めます。

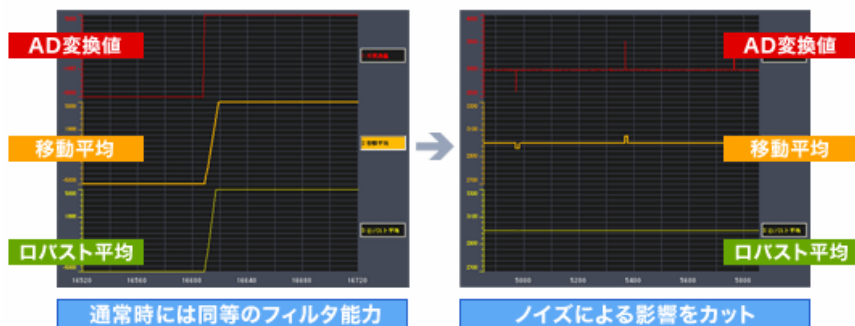
DM0 = AVG( @DM2 , @DM1 )

## アプリケーション⑩

### ロバスト平均制御

### ～アナログデータからノイズ(外乱)の影響を取り除く～

FIFO関数を使用して、ロバスト平均も簡単に計算



変位や温度などの連続したアナログデータからノイズ(外乱)の影響を除き、安定した平均値を求めるのに便利なスクリプトです。

### 測定したデータからロバスト平均法を用いて平均値を計算する例



```

*** デバイス解説 ***
'DM0:A/D変換値
'DM1:平均値(ロバスト平均法を用いて平均した結果を格納)
'DM2:平均データ個数(3～10000)
'(ラダーにて直前の何個のデータをロバスト平均値計算に使用するかを設定)
'DM3～DM15:演算に使用
*** FIFOテーブル解説 ***
'@EM0:現在格納されているデータ個数
'(FIFOR・FIFOW命令を実行すると自動格納)
'(注: @EM0は動作開始時に0クリアしてください)
'@EM1:格納できるデータ個数最大値
'(ラダーにてFIFOR・FIFOW命令実行前に設定)
'@EM2～:データ(デバイス番号の若い順に新しいデータ)
'平均データ個数 DM2をFIFOテーブルの @EM1に格納します。
@EM1.U = DM2.U
'FIFOテーブルに格納されたA/D変換値が平均データ個数を超えれば、
'古いデータを読み出します。
IF @EM0.U >= @EM1.U
  THEN
    FIFOR( @EM0.S , DM3.S )
  END IF
'DM0に格納されたA/D変換値を @EM0からのFIFOテーブルに格納します。
FIFOW( DM0.S , @EM0.S )
'最大値と最小値を除きます。
DM4.L=WSUM(@EM2.S,@EM1)-MAX(@EM2.S,@EM1)-MIN(@EM2.S,@EM1)
'最大値・最小値を除くためデータ数から2を引きます。
DM6 = @EM1-2
'データの平均値を求めます。
DM1.S = DM4.L/TOL(DM6)
    
```

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

- 制御システムに  
関するお問合せ

**06-6379-1164**

月～金曜(祝日を除く)8:30～20:00

- 情報サービス [www.keyence.co.jp](http://www.keyence.co.jp)

カタログ、取扱説明書、マニュアル、CADデータ等をダウンロードできます。

- 輸出書類サービス [www.keyence.co.jp/yushutsu](http://www.keyence.co.jp/yushutsu)

輸出に必要な書類をその場でダウンロードできます。

**株式会社 キーエンス**

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

仕様は改良のため予告なく変更することがあります。  
記載されている会社名、製品名等は、それぞれ各社の商標または登録商標です。

Copyright© 2016 KEYENCE CORPORATION.  
All rights reserved.

制御2-1036

1096-1 084435