

TBASIC 関数及びステートメント

1. 整数変数 Integer 変数

TRiLOGi Ver4.0 TBASIC はすべての演算処理で 32 ビット Integer 演算を行うことができます。TBASIC の組込み変数 A~Z の 26 個の 32 ビット長として $-2^{31} \sim +2^{31}$ の数値を表すことができます。またデータメモリー DM[n] にデータを格納する場合は、すべて 16 ビット値 2^{16} の $-32,768 \sim +32,767$ の整数としてデータを格納します。16 ビット変数は次の通りです。

DM[1]~DM[4000]	データメモリー値
Input[n]	入力値
Output[n]	出力値
Relay[n]	リレー値
Timerbit[n]	タイマビット値
Ctrbit[n]	カウンタビット値
TimerPV[n]	タイマセット値
CtrPV[n]	カウンタセット値
HSCPV[n]	高速カウンタセット値
TIME[n]	時間値
DATE[n]	日付値

すべての数値演算は 32 ビット整数値のビット長として演算を行うことができます。また TBASIC はすべての変数をグローバル変数として定義できます。これはカスタム関数 Custom Function #1~#128 のどのカスタム関数からでも A~Z の変数を参照・使用することができます。

2. 数値演算子 Integer Operators

a) 算術演算子 Arithmetic Operators

演算子	意味
+	加算
-	減算
*	乗算
/	除算
MOD	整数除算のあまり

b) ビット演算子 Bitwise Logical Operators

演算子	意味	使用例
&	理論積	IF Input[1] & &H02 THEN
	理論和	Output[1] = A &H08
^	排他的理論和	A = RELAY[2] ^ B
~	不定ビット反転	A = ~ TimerPV[1]

c) 関係・理論演算子 Relational Operators

演算子	意味	使用例
=	等しい	IF A = 100 THEN
<>	等しくない	WHILE CTRPV[0] <> 0
>	大きい	IF B > C / (D + 10)
<	小さい	IF TIME[3] < 59
>=	大きいか等しい	WHILE X >= 10
<=	小さいか等しい	IF DM[1] <= 5678
AND	かつ	IF A > B AND C <= D
OR	または	IF A <> 0 OR B = 1000

3. 文字列データ String Data

文字列データはダブルクォテーションマーク"□"で囲まれた 70 キャラクター(半角英数字)の 1 文字 8 ビット ASCII コードを連続した文字列として扱えます。

```
"TBASIC mode PLC Numeric a piece of cake!"
"$102,345.00"
```

また、TBASIC の組込み文字列変数 A\$, B\$~Z\$までの 26

個に、1 つの文字変数に 0 (Null 文字)~70 キャラクター(半角英数字)の文字変数を格納できます。

4. 文字列演算 String Operators

a) 文字列変数への代入

整数変数値(A~Z)、データメモリー値(DM[1]~DM[4000])、システムデータ値等を文字列データに格納できます。

```
A$ = "Hello, Welcome TO TBASIC"
B$ = MID(A$, 3, 5)
```

b) 文字列の連結

2 つ以上の文字列データを結合演算子 + を使って表現できます。

```
A$ = James
B$ = TBASIC
M$ = "Hello " + A$ + ", Welcome TO " + B$
(M$の文字列は"Hello James, Welcome TO TBASIC")
```

c) 文字列の比較

2 つの文字を、組込み関数 STRCMP(A\$,B\$)を使用して、文字列の大小関係を比較して、結果を得ることができます。しかし、文字列の比較には関係演算子 = <> は使用できません。

5. TBASIC 関数とステートメント

演算子	内容(概略)
ABS (x)	x の絶対値を返します。
ADC (n)	A/D 変換チャンネル n の値を返します。
ASC (x\$, n)	x\$文字列の n 番目の文字の ASCII コードを返します。
CALL n	他のカスタム関数 CusFn #n を呼び出します。
CHR\$ (n)	ASCIIコード(0~255) nに対応する文字を返します。
CLRBIT v, n	ビット番号 n を Integer 値 v が 0 になるようにクリアします。
FOR ... NEXT	FOR と NEXT で囲まれた範囲を所定の回数だけループします。
GetCtrSV (n)	カウンタ番号 Counter n 番のセット値(S.V)を返します。
GetTimerSV (n)	タイマー番号 Timer n 番のセット値(S.V)を返します。
GETHIGH16 (v)	32 ビット Integer の上位 16 ビットの値を返します。
GOTO @ n	現在のカスタム関数内のラベル@ n で指定される文に、無条件分岐します。
HEX\$ (n)	数値 n を 16 進で表現した文字列に変換します。
HEXVAL (x\$)	文字列 x\$を 16 進の数値に変換します。
HSTIMER n	タイマ番号 1~n 番をタイムベース 0.01sec の高速タイマ[HST]として定義します。
HSCDEF ch, fn, v	高速カウンタチャンネル ch にパラメータをセットし有効にします。
HSCOFF ch	高速カウンタチャンネル ch を無効にします。
IF .. THEN .. ELSE ENDIF	条件判断のフロー制御ステートメントを構成します。
INPUTS\$ (ch)	COMMポート番号 ch から文字列データを返します。
INTRDEF ch, fn, e	入力チャンネル番号 ch の割り込み、および割り込みルーチンの カスタム関数 CusFn fn を有効にします。
INTROFF ch	入力チャンネル番号 ch の割り込みを無効

TBASIC 関数及びステートメント

	にします。
LEN (x\$)	文字列 x\$のサイズ数を返します。
LOAD_EEPROM (addr)	ユーザ定義領域の EEPROM にストアされたアドレス addr の値を得ます。
LSHIFT i, n	16 ビット Integer 値を 1 ビット左にシフトします。
MID\$ (x\$, n, m)	文字列 x\$のスタート位置 n から m 個の文字を取り出します。
PAUSE	実行中のカスタム関数にブレークポイントを設定します。
PIDdef ch, lmt, P, I, D	PIDコントローラ機能(比例・積分・微分制御)にパラメータを設定します。
PIDcomput (ch, E)	PIDdef コマンドで設定した P・I・D 値を使って出力エラー E を演算します。
PRINT# n x\$: y; z	COMMポート番号 n にフォーマットリストの文字列を送出します。
PMON ch	PMON は入力チャンネル ch パルス計測機能を有効にします。
PMOFF ch	PMOFF はこのチャンネル ch のパルス計測機能を無効にします。
PULSEFREQUENCY	最終入力パルスの周波数 Hz を返します。
PULSEPERIOD	入力チャンネルのパルス周期を msec 単位で返します。
PULSEWIDTH	入力チャンネルのパルス幅を msec 単位で返します。
REFRESH	出力ビットを直ちにリフレッシュします。
RESET	ソフトウェア上から PLC をリセットします。
RETURN	現在のカスタム関数から無条件でステートメントを戻します。
REM (or ')	プログラムにコメントを記述します。
RSHIFT i, n	16 ビット Integer 値を 1 ビット右にシフトします。
SAVE_EEPROM data, addr	ユーザ定義領域の EEPROM のアドレス addr に 16 ビット Integer データ data をストアします。
SETBAUND ch, baud	PLC の COMM ポート n の通信速度を設定します。
SETBIT v, n	ビット番号 n を Integer 値 y が '1' になるようにセットします。
SetCtrSV n, val	カウンタ番号 Counter n のセット値(S.V)を変更します。
SetTimerSV n, val	タイマ番号 Timer n のセット値(S.V)を変更します。
SETDAC n, x	D/A 変換チャンネル n に値 x を設定します。
SETTHIGH16 v, d	32 ビット Integer 値 v の上位 16 ビットに割り当てます。
SETPWM n, x, y	PWM 制御チャンネル n をデューティ比 x% に設定します。
STEPCOUNT (ch)	STEPMOVE コマンドのステップ数をカウントします。
STEPSPEED ch, pps, acc	ステッピングモータ出力の制御パラメータを設定します。
STEPMOVE ch, count, r	パルス発生用出力チャンネルとパルスカウンタを動作させます。
STEPSTOP	ステッピングモータ用出力チャンネルの出力を停止します。
STR\$ (n)	数値 n を文字列に変換します。
STRCMP (A\$, B\$)	文字列 A\$ と B\$ の大小関係を比較し、結果を返します。
STRUPR\$ (A\$)	文字列 A\$ のをすべて大文字に変換した文字列を返します。
STRLWR\$ (A\$)	文字列 A\$ のをすべて小文字に変換した文字列を返します。
TESTBIT (v, n)	ビット n のロジック実行状態値を返します。
WHILE ... ENDWHILE	前判定反復のループ制御ステートメントを構成します。
VAL (x\$)	文字列 x\$ を数値に変換します。