# 安全プログラミング

## 命令語

TOYOPUC-Nano Safetyは、基本命令19種, タイマー命令8種, カウンタ命令6種、応用命令(PS/PN共通):5種、応用命令(PNのみ):40種の命令が使用できます。

### 基本命令(19種)

|  |  |  |  |
| --- | --- | --- | --- |
| No | シンボル | 言語 | 機能 |
| 1 |  | STR | 演算開始(A接点) |
| 2 |  | STR NOT | 演算開始(B接点) |
| 3 |  | AND | 直列接続(A接点) |
| 4 |  | AND NOT | 直列接続(B接点) |
| 5 |  | OR | 並列接続(A接点) |
| 6 |  | OR NOT | 並列接続(B接点) |
| 7 |  | AND STR | 論理ブロック間の直列接続 |
| 8 |  | OR STR | 論理ブロック間の並列接続 |
| 9 |  | OUT | コイル出力 |
| 10 |  | PTS | 立上り微分 |
| 11 |  | NTS | 立下り微分 |
| 12 |  | FPS | マルチコイル分岐開始 |
| 13 |  | FRD | マルチコイル分岐 |
| 14 |  | FPP | マルチコイル分岐終了 |
| 15 |  | FST | 無条件出力用 |
| 16 |  | NOT | 条件反転 |
| 17 |  | NOP | 無処理 |
| 18 |  | SET | キープリレーセット(PNのみ) |
| 19 |  | RST | キープリレーリセット(PNのみ) |

表 10‑1　基本命令(19種)

### タイマー命令(8種)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | 分類 | 機能 | ニーモ  ニック | シンボル例 | 演算内容 |
| 1 | タイマー | 直接指定  10msタイマー | TMRH | T000  TMRH K=655.35 | 設定値が655.35秒の10msタイマーです。 |
| 2 | 間接指定  10msタイマー | TMRH | T001  TMRH S=D0010 | D0010の内容を設定値とする10msタイマーです。 |
| 3 | 直接指定  100msタイマー | TMR | T002  TMR K=6553.5 | 設定値が6553.5秒の100msタイマーです。 |
| 4 | 間接指定  100msタイマー | TMR | T003  TMR S=D0011 | D0011の内容を設定値とする100msタイマーです。 |
| 5 | 積算タイマー | 直接指定  10ms積算タイマー | TMRSH | T004  T  TMRSH K=123.45  R | 設定値が123.45秒の10ms積算タイマーです。(※1) |
| 6 | 間接指定  10ms積算タイマー | TMRSH | T005  T  TMRSH S=D0012  R | D0012の内容を設定値とする10ms積算タイマーです。(※1) |
| 7 | 直接指定  100ms積算タイマー | TMRS | T006  T  TMRS K=1234.5  R | 設定値が1234.5秒の100ms積算タイマーです。(※1) |
| 8 | 間接指定  100ms積算タイマー | TMRS | T007  T  TMRS S=D0013  R | D0013の内容を設定値とする100ms積算タイマーです。(※1) |

(※1)積算タイマーのデータは電源OFF時に消去されます。

表 10‑2　タイマー命令(8種)

### カウンタ命令(6種)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | 分類 | 機能 | ニーモニック | シンボル例 | 演算内容 |
| 1 | アップ  カウンタ | 直接指定  アップカウンタ | CNT | C008  CK  CNT K=65535  R | 設定範囲が1-65535のアップカウンタです。  設定値に達すると指定されたデバイス(C000~C1FF)をONします。 |
| 2 | 間接指定  アップカウンタ | CNT | C009  CK  CNT S=D0014  R | D0014の内容を設定値とするアップカウンタです。  設定値に達すると指定されたデバイス(C000~C1FF)をONします。 |
| 3 | ダウン  カウンタ | 直接指定  ダウンカウンタ | CNTD | C00A  CK  CNT K=65535  R | 設定範囲が1-65535のダウンカウンタです。  ０に達すると指定されたデバイス(C000~C1FF)をONします。 |
| 4 | 間接指定  ダウンカウンタ | CNTD | C00B  CK  CNT S=D0015  R | D0015の内容を設定値とするアップカウンタです。  設定値に達すると指定されたデバイス(C000~C1FF)をONします。 |
| 5 | アップ  ダウン  カウンタ | 直接指定  アップダウン  カウンタ | CNTH | C00C  CK  CNTH K=65535  U/D  R | 設定範囲が1-65535のアップダウンカウンタです。  現在値と指定値が一致すると指定されたデバイス(C000~C1FF)をONします。 |
| 6 | 間接指定  アップダウンカウンタ | CNTH | C00D  CK  CNTH S=D0016  U/D  R | D0016の内容を設定値とするアップカウンタです。  現在値と指定値が一致すると指定されたデバイス(C000~C1FF)をONします。 |

表 10‑3　カウンタ命令(6種)

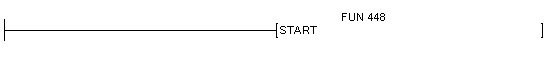
### 応用命令(PS/PN共通)(5種)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | 分類 | | 番号 | 命令語 | シンボル | | 機能 |
| 1 | ラベル | メインプログラム開始 | FUN448 | START | START  PEND | メインシーケンスプログラムの開始を示す。 | |
| 2 | メインプログラム終了 | FUN452 | END | END | メインシーケンスプログラムの終了を示す。 | |
| 3 | プログラム終了 | FUN456 | PEND |  | サブルーチンを含むシーケンスプログラムの終了を示す。 | |
| 4 | データ転送 | １バイト直接転送 | FUN90 | MOVE | MOVE  WMOVE  S  D | SのﾃﾞｰﾀをDに転送 | |
| 5 | ２バイト直接転送 | FUN0 | WMOVE |

表 10‑4　応用命令(PS/PN共通)(5種)

1.START：メインプログラム開始(FUN448)

1. シンボル



1. 機能

メインプログラムの開始を示します｡

“END”命令実行後、プログラムの実行は本命令のあるステップに戻ります｡

1. フラグ：変換なし
2. プログラム例

START, END, PEND命令に対し、条件(接点)を入れることはできません。

イニシャルプログラムは、電源投入時またはリセット/スタート時1回しか実行しません。

START命令は実行してもなにも行いません。



2.END：メインプログラム終了(FUN452)

(1)シンボル

CMD_End

(2)機能

メインプログラムの終了を示します｡

本命令にて実際に各モジュールのI/Oをリフレッシュします｡

本命令実行終了後は、START命令よりプログラムの実行を開始します｡(メインプログラムの先頭から再開｡)

(3)フラグ：変化なし

(4)プログラム例

START命令を参照して下さい。

3.PEND：プログラム終了(FUN456)

(1)シンボル

CMD_Pend

(2)機能

ユーザプログラム領域の終了を示します｡

(3)フラグ：変化なし

(4)プログラム例

START命令を参照して下さい。

4.MOVE：１バイト直接転送(FUN90)

(1)シンボル



(2)機能

OP1のレジスタのデータをOP2のレジスタへ転送します。このとき、OP1のレジスタのデータは保存されます。

(3)フラグ：変化なし

(4)プログラム例

M000がONしていれば、データレジスタD0000の下位8ビットのデータをデータレジスタD0100の上位8ビットに転送するプログラム

PEX_Move.bmp

5.WMOVE：２バイト直接転送(FUN0)

(1)シンボル



(2)機能

OP1のレジスタのデータをOP2のレジスタへ転送します。このとき、OP1のレジスタのデータは保存されます。

(3)フラグ：変化なし

(4)プログラム例

PEX_Wmove.bmpX001がOFFからONになった時、データレジスタD0123のデータをY020～YO2Fに転送するプログラム

### 応用命令(PNのみ)(40種)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | 分類 | | 命令語 | シンボル | | 機能 |
| 1  =H  W=H  S  H | 比較 | 16進2桁定数比較（=） | =H |  | ﾚｼﾞｽﾀと定数またはﾚｼﾞｽﾀとﾚｼﾞｽﾀの比較を行ない,S=H, S1=S2ならば,接点は導通状態となる. | |
| 2 | 16進4桁定数比較（=） | W=H |
| 3  =N  W=N  S1  S2 | バイトデータ比較（=） | =N |  |
| 4 | ワードデータ比較（=） | W=N |
| 5  <>H  W<>  S  H | 16進2桁定数比較（≠） | <>H |  | ﾚｼﾞｽﾀと定数またはﾚｼﾞｽﾀとﾚｼﾞｽﾀの比較を行ない,S≠H, S1≠S2ならば,接点は導通状態となる. | |
| 6 | 16進4桁定数比較（≠） | W<>H |
| 7  <>N  W<>N  S1  S2 | バイトデータ比較（≠） | <>N |  |
| 8 | ワードデータ比較（≠） | W<>N |
| 9  >H  W>H  S  H | 16進2桁定数比較（＞） | >H |  | ﾚｼﾞｽﾀと定数またはﾚｼﾞｽﾀとﾚｼﾞｽﾀの比較を行ない,S>H, S1>S2ならば,接点は導通状態となる. | |
| 10 | 16進4桁定数比較（＞） | W>H |
| 11  >N  W>N  S1  S2 | バイトデータ比較（＞） | >N |  |
| 12 | ワードデータ比較（＞） | W>N |
| 13  >=H  W>=H  S  H | 16進2桁定数比較（≧） | >=H |  | ﾚｼﾞｽﾀと定数またはﾚｼﾞｽﾀとﾚｼﾞｽﾀの比較を行ない,S≧H, S1≧S2ならば,接点は導通状態となる. | |
| 14 | 16進4桁定数比較（≧） | W>=H |
| 15  >=N  W>=N  S1  S2 | バイトデータ比較（≧） | >=N |  |
| 16 | ワードデータ比較（≧） | W>=N |
| 17  <H  W<H  S  H | 16進2桁定数比較（＜） | <H |  | ﾚｼﾞｽﾀと定数またはﾚｼﾞｽﾀとﾚｼﾞｽﾀの比較を行ない,S<H, S1<S2ならば,接点は導通状態となる. | |
| 18 | 16進4桁定数比較（＜） | W<H |
| 19  <N  W<N  S1  S2 | バイトデータ比較（＜） | <N |  |
| 20 | バイトデータ比較（＜） | W<N |
| 21  <=H  W<=H  S  H | 16進2桁定数比較（≦） | <=H |  | ﾚｼﾞｽﾀと定数またはﾚｼﾞｽﾀとﾚｼﾞｽﾀの比較を行ない,S≦H, S1≦S2ならば,接点は導通状態となる. | |
| 22 | 16進4桁定数比較（≦） | W<=H |
| 23  <=N  W<=N  S1  S2 | バイトデータ比較（≦） | <=N |  |
| 24 | ワードデータ比較（≦） | W<=N |
| 25  MOV  WMOV  H  D | データ転送 | 16進2桁定数転送 | MOV | BCD  WBCD  S  D | 16進定数H(2,4桁)をDに転送. | |
| 26 | 16進4桁定数転送 | WMOV |
| 27 | データ変換 | 2進8ビット→BCD2桁変換 | BCD |  | Sに格納された2進ﾃﾞｰﾀをにBCDﾃﾞｰﾀ変換して,Dに格納. | |
| 28 | 2進16ビット→BCD4桁変換 | WBCD |
| 29 | BCD2桁→2進8ビット変換 | BIN | BIN  WBIN  S  D | Sに格納されたBCDﾃﾞｰﾀを2進ﾃﾞｰﾀに変換して,Dに格納. | |
| 30 | BCD4桁→2進16ビット変換 | WBIN |
| 31 | 算術  演算 | ワードデータ2進加算 | W+ | W+  S1  S2  D | S1の内容とS2の内容を加算し,結果をDに格納.ﾃﾞｰﾀは全て2進数として扱われる. | |
| 32 | ワードデータ2進減算 | W- | W\*  S1  S2  D  W-  S1  S2  D | S1の内容からS2の内容を引き,結果をDに格納.ﾃﾞｰﾀは全て2進数として扱われる. | |
| 33 | ワードデータ2進乗算 | W\* |  | S1の内容とS2の内容を乗算し,結果をDに格納.ﾃﾞｰﾀは全て2進数として扱われる. | |
| 34 | ワードデータ2進除算 | W/ | W/  S1  S2  D | S1の内容をS2の内容で割り,商をDに余りをD+1に格納.ﾃﾞｰﾀは全て2進数として扱われる. | |
| 35 | ワードデータ2進インクリメント | WINC | WDEC  D  WINC  S  D | Dの内容を+1した後,S1の内容と比較.ﾃﾞｰﾀは2進数扱い. | |
| 36 | ワードデータ2進デクリメント | WDEC |  | Dの内容を-1する.ﾃﾞｰﾀは2進数扱い. | |

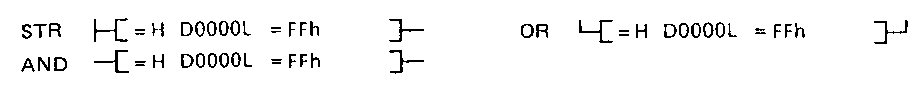
表 10‑5　応用命令(PN)(40種)　1/2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | 分類 | | 命令語 | シンボル | | 機能 |
| 37 | 論理  演算 | バイトデータ論理積 | AND | AND  S1  S2  D | S1の内容とS2の内容の論理積(AND)を行ない,結果をDに格納. | |
| 38 | バイトデータ論理和 | OR | OR  S1  S2  D | S1の内容とS2の内容の論理和(OR)を行ない,結果をDに格納. | |
| 39 | バイトデータ反転 | NOT | NOT  S  D | Sの内容を反転(各ﾋﾞｯﾄを1なら0に,0なら1にする)を行ない,結果をDに格納. | |
| 40 | バイトデータ排他的  論理和 | XOR | XOR  S1  S2  D | S1の内容とS2の内容の排他的論理和(XOR)を行ない,結果をDに格納. | |

表 10‑6　応用命令(PN)(40種)　2/2

#### 比較

1.=H：16進2桁定数比較（=）

(1) シンボル

(2) 機能

レジスタの8ビットデータと、16進2桁定数を比較し、一致していれば導通状態となります。

(3) プログラム例

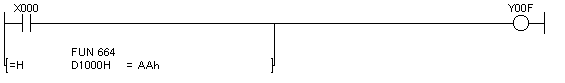
① X000~X007がBCDデータの“23”の時、Y00FをONします。

PEX_eqH_1

② X000がONで、データレジスタD1000の下位８ビットが16進の“FF”の時Y00FをONします。

PEX_eqH_2

③ X000がONであるか、または、データレジスタD1000の上位８ビットが16進の“AA”の時Y00FをONします。



2.W=H：16進4桁定数比較（=）

(1) シンボル



(2) 機能

レジスタの16ビットデータと、16進4桁定数を比較し、一致していれば導通状態となります。

(3) プログラム例

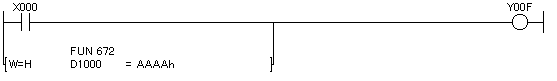
① X000~X00Fが16進の“1234”の時、Y00FをONします。

PEX_WeqH_1

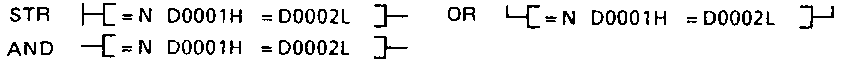
② X000がONで、データレジスタD1000が16進の“FFFF”の時Y00FをONします。

PEX_WeqH_2

③ X000がONであるか、または、データレジスタD1000が16進の“AAAA”の時Y00F をONします。



3.=N：バイトデータ比較（=）

(1) シンボル

(2) 機能

2つの8ビットデータを比較し、一致していれば導通状態となります。

(3) プログラム例

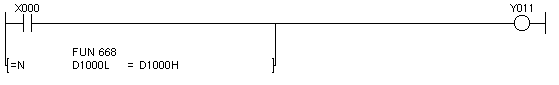
① データレジスタD0001の上位8ビットと、データレジスタD0002の下位8ビットが一致していればY011をONします。

PEX_eqN_1

② X000がONで、データレジスタD1000の下位8ビットとX010~X017が一致していればY011をONします。

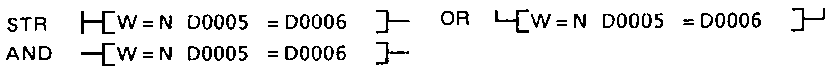
PEX_eqN_2

③ X000がONであるか、または、データレジスタD1000の下位8ビットとD1000の上位8ビットが一致していればY011をONします。



4.W=N：ワードデータ比較（=）

(1) シンボル



(2) 機能

2つの16ビットデータを比較し、一致していれば導通状態となります。

(3) プログラム例

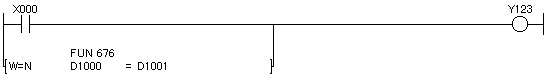
① カウンタC000の現在値(現在値レジスタN000の値)と、カウンタC001の現在値(現在値カウンタN001の値)が一致した時Y123をONします。

PEX_WeqN_1

② X000がONで、X010~X01FとD1000の内容が一致していればY123をONします。

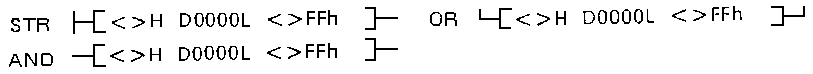
PEX_WeqN_2

③ X000がONであるか、または、データレジスタD1000の内容とデータレジスタD1001の内容が一致していればY123をONします。



5.<>H：16進2桁定数比較（≠）

(1) シンボル



(2) 機能

レジスタの8ビットデータと、16進2桁定数を比較し、一致していなければ導通状態となります。

(3) プログラム例

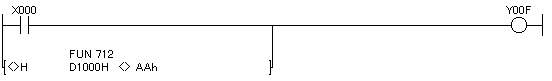
① X000~X007が16進の“23”でない時、Y00FをONします。

PEX_neH_1

② X000がONで、データレジスタD1000の下位8ビットが16進の“FF”でない時Y00FをONします。

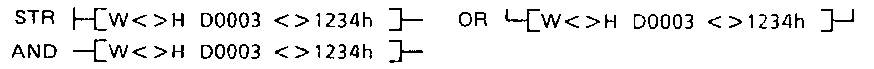
PEX_neH_2

③ X000がONであるか、または、データレジスタD1000の上位8ビットが16進の“AA”でない時Y00FをONします。



6.W<>H：16進4桁定数比較（≠）

(1) シンボル



(2) 機能

レジスタの16ビットデータと、16進4桁定数を比較し、一致していなければ導通状態となります。

(3) プログラム例

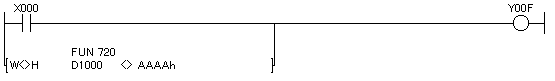
① X000~X00Fが16進の“1234”でない時、Y00FをONします。

PEX_WneH_1

② X000がONで、データレジスタD1000が16進の“FFFF”でない時Y00FをONします。

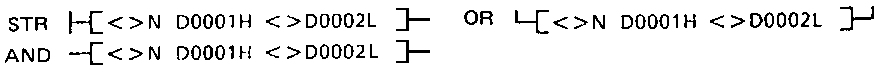
PEX_WneH_2

③ X000がONであるか、または、データレジスタD1000が16進の“AAAA”でない時Y00FをONします。



7.<>N：バイトデータ比較（≠）

(1) シンボル



(2) 機能

2つの8ビットデータを比較し、一致していなければ導通状態となります。

(3) プログラム例

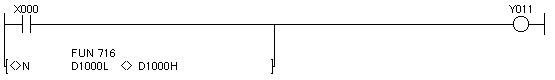
① データレジスタD0001の上位8ビットと、データレジスタD0002の下位8ビットが一致していなければY011をONします。

PEX_neN_1

② X000がONで、データレジスタD1000の下位8ビットとX010~X017が一致していなければY011をONします。

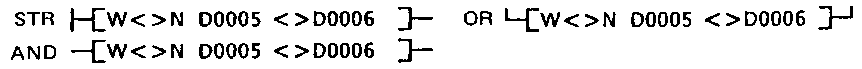
PEX_neN_2

③ X000がONであるか、または、データレジスタD1000の下位8ビットとD1000の上位8ビット一致していなければY011をONします。



8.W<>N：ワードデータ比較（≠）

(1) シンボル



(2) 機能

2つの16ビットデータを比較し、一致していなければ導通状態となります。

(3) プログラム例

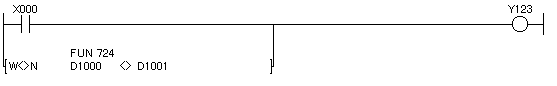
① カウンタC000の現在値(現在値レジスタN000の値)とカウンタC001の現在値(現在値カウンタN001の値)が一致しない時、Y123をONします。

PEX_WneN_1

② X000がONで、X010~X01FとD1000の内容が一致していなければY123をONします。

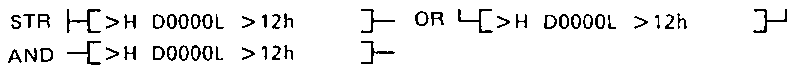
PEX_WneN_2

③ X000がONであるか、または、データレジスタD1000の内容とデータレジスタD1001の内容が一致していなければY123をONします。



9.>H：16進2桁定数比較（＞）

(1) シンボル



(2) 機能

レジスタの8ビットデータと、16進2桁定数を比較し、レジスタデータが定数より大きければ導通状態となります。

(3) プログラム例

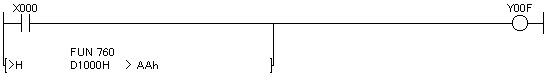
① X000~X007が16進の“23”より大きい時、Y00FをONします。

PEX_gtH_1

② X000がONで、データレジスタD1000の下位8ビットが16進の“12”より大きい時Y00FをONします。

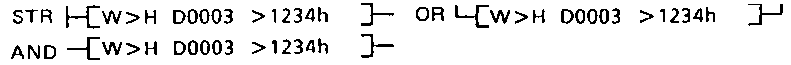
PEX_gtH_2

③ X000がONであるか、または、データレジスタD1000の上位8ビットが16進の“AA”より大きい時Y00FをONします。



10.W>H：16進4桁定数比較（＞）

(1) シンボル



(2) 機能

レジスタの16ビットデータと、16進4桁定数を比較し、レジスタデータが定数よりきければ導通状態となります。

(3) プログラム例

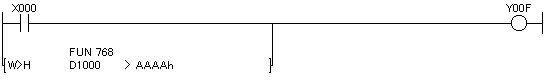
① X000~X00Fが16進の“1234”より大きい時、Y00FをONします。

PEX_WgtH_1

② X000がONで、データレジスタD1000が16進の“5555”より大きい時Y00FをONします。

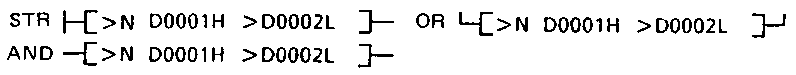
PEX_WgtH_2

③ X000がONであるか、または、データレジスタD1000が16進の“AAAA”より大きい時Y00FをONします。



11.>N：バイトデータ比較（＞）

(1) シンボル



(2) 機能

2つの８ビットデータを比較し、1つ目のデータ値が２つ目のデータ値より大きければ導通状態になります。

(3) プログラム例

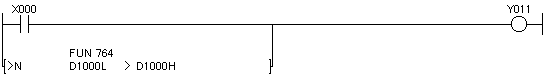
① データレジスタD0001の上位8ビットが、データレジスタD0002の下位8ビットより大きければY011をONします。

PEX_gtN_1

② X000がONで、データレジスタD1000の下位8ビットとX010~X017で表される値より大きければY011をONします。

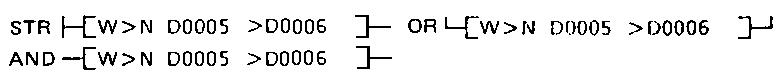
PEX_gtN_2

③ X000がONであるか、または、データレジスタD1000の下位8ビットがD1000の上位8ビットより大きければY011をONします。



12.W>N：ワードデータ比較（＞）

(1) シンボル



(2) 機能

2つの16ビットデータを比較し、1つ目のデータ値が2つ目のデータ値より大きければ導通状態となります。

(3) プログラム例

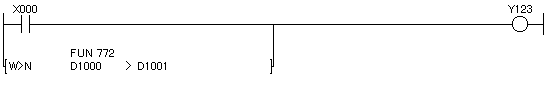
① カウンタC000の現在値(現在値レジスタN000の値)が、カウンタC001の現在値(現在値カウンタN001の値)より大きい時、Y123をONします。

PEX_WgtN_1

② X000がONで、X010~X01Fで表される値がD1000の内容より大きければY123をONします。

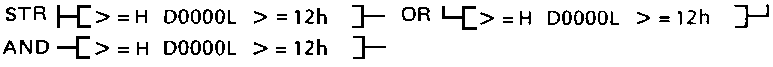
PEX_WgtN_2

③ X000がONであるか、または、データレジスタD1000の内容がデータレジスタD1001の内容より大きければY123をONします。



13.>=H：16進2桁定数比較（≧）

(1) シンボル



(2) 機能

レジスタの8ビットデータと、16進2桁定数を比較し、レジスタデータが定数以上であれば導通状態となります。

(3) プログラム例

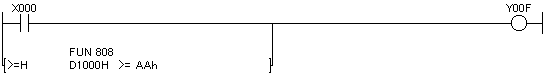
① X000~X007が16進の“23”以上の時、Y00FをONします。

PEX_geH_1

② X000がONで、データレジスタD1000の下位8ビットが16進の“12”以上の時Y00FをONします。

PEX_geH_2

③ X000がONであるか、または、データレジスタD1000の上位8ビットが16進の“AA”以上の時Y00FをONします。



14.W>=H：16進4桁定数比較（≧）

(1) シンボル



(2) 機能

レジスタの16ビットデータと、16進4桁定数を比較し、レジスタデータが定数以上であれば導通状態となります。

(3) プログラム例

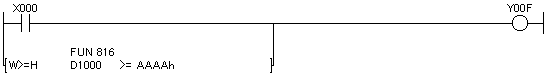
① X000~X00Fが16進の“1234”以上の時、Y00FをONします。

PEX_WgeH_1.bmp

② X000がONで、データレジスタD1000が16進の“5555”以上の時Y00FをONします。

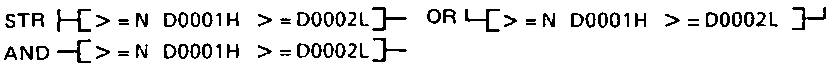
PEX_WgeH_2.bmp

③ X000がONであるか、または、データレジスタD1000が16進の“AAAA”以上の時Y00FをONします。



15.>=N：バイトデータ比較（≧）

(1) シンボル



(2) 機能

2つの8ビットデータを比較し、1つ目のデータ値が2つ目のデータ値以上であれば導通状態となります。

(3) プログラム例

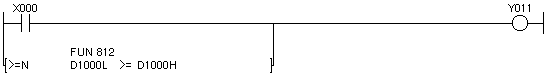
① データレジスタD0001の上位8ビットが、データレジスタD0002の下位8ビット以上であればY011をONします。

PEX_geN_1.bmp

② X000がONで、データレジスタD1000の下位8ビットがX010~X018で表される値以上であればY011をONします。

PEX_geN_2.bmp

③ X000がONであるか、または、データレジスタD1000の下位8ビットがD1000の上位8ビット以上であればY011をONします。



16.W>=N：ワードデータ比較（≧）

(1) シンボル



(2) 機能

2つの16ビットデータを比較し、1つ目のデータ値が2つ目のデータ値以上であれば導通状態となります。

(3) プログラム例

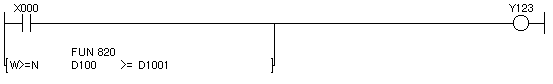
① カウンタC000の現在値(現在値レジスタN000の値)がカウンタC001の現在値(現在値カウンタN001の値)以上の時、Y123をONします。

PEX_WgeN_1

② X000がONで、X010~X01Fで表される値がD1000の内容以上であればY123をONします。

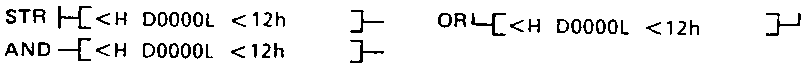
PEX_WgeN_2

③ X000がONであるか、または、データレジスタD100の内容がデータレジスタD1001の内容以上であればY123をONします。



17.<H：16進2桁定数比較（＜）

(1) シンボル



(2) 機能

レジスタの8ビットデータと、16進2桁定数を比較し、レジスタデータが定数より小

さければ導通状態となります。

(3) プログラム例

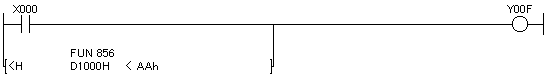
① X000~X007で表される値が16進の“23”より小さい時、Y00FをONします。

PEX_ltH_1.bmp

② X000がONで、データレジスタD1000の下位8ビットが16進の“12”より小さい時Y00FをONします。

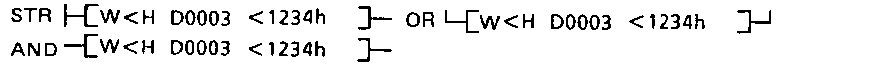
PEX_ltH_2.bmp

③ X000がONであるか、または、データレジスタD1000の上位8ビットが16進の“AA”より小さい時Y00FをONします。



18.W<H：16進4桁定数比較（＜）

(1) シンボル



(2) 機能

レジスタの16ビットデータと、16進4桁定数を比較し、レジスタデータが定数より小さければ導通状態となります。

(3) プログラム例

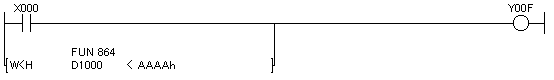
① X000~X00Fが16進の“1234”より小さい時、Y00FをONします。

PEX_WltH_1.bmp

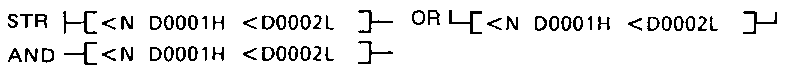
② X000がONで、データレジスタD1000が16進の“5555”より小さい時Y00FをONします。

PEX_WltH_2.bmp

③ X000がONであるか、または、データレジスタD1000が16進の“AAAA”より小さい時Y00FをONします。



19.<N：バイトデータ比較（＜）

(1) シンボル

(2) 機能

2つの8ビットデータを比較し、1つ目のデータ値が2つ目のデータ値より小さければ導通状態となります。

(3) プログラム例

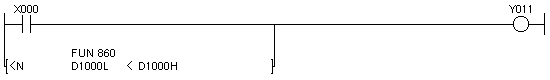
① データレジスタD0001の上位8ビットより、データレジスタD0002の下位8ビットが小さければY011をONします。

PEX_ltN_1.bmp

② X000がONで、データレジスタD1000の下位8ビットがX010~X017で表される値より小さければY011をONします。

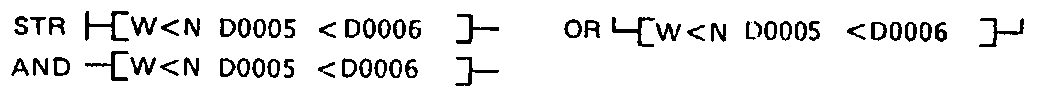
PEX_ltN_2.bmp

③ X000がONであるか、または、データレジスタD1000の下位8ビットがD1000の上位8ビットより小さければY011をONします。



20.W<N：バイトデータ比較（＜）

(1) シンボル



(2) 機能

2つの16ビットデータを比較し、1つ目のデータ値が2つ目のデータ値より小さければ導通状態となります。

(3) プログラム例

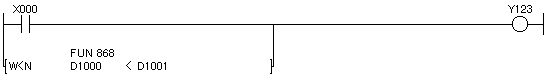
① カウンタC000の現在値(現在値レジスタN000の値)がカウンタC001の現在値 (現在値カウンタN001の値)より小さければ時、Y123をONします。

PEX_WltN_1.bmp

② X000がONで、X010~X01FとD1000の内容より小さければY123をONします。

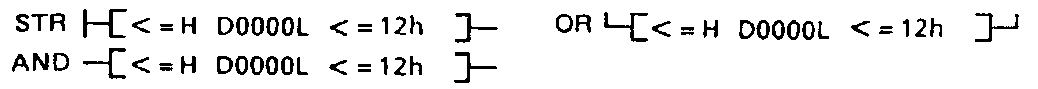
PEX_WltN_2.bmp

③ X000がONであるか、または、データレジスタD1000の内容がデータレジスタD1001の内容より小さければY123をONします。



21.<=H：16進2桁定数比較（≦）

(1) シンボル



(2) 機能

レジスタの8ビットデータと、16進2桁定数を比較し、レジスタデータが定数以下であれば導通状態となります。

(3) プログラム例

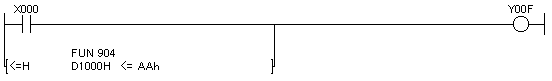
① X000~X007が16進の“23”以下の時、Y00FをONします。

PEX_leH_1.bmp

② X000がONで、データレジスタD1000の下位8ビットが16進の“55”以下の時Y00FをONします。

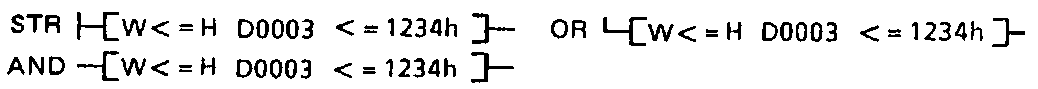
PEX_leH_2.bmp

③ X000がONであるか、または、データレジスタD1000の上位8ビットが16進の“AA”以下の時Y00FをONします。



22.W<=H：16進4桁定数比較（≦）

(1) シンボル



(2) 機能

レジスタの16ビットデータと、16進4桁定数を比較し、レジスタデータが定数以下であれば導通状態となります。

(3) プログラム例

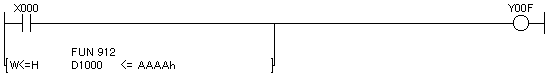
① X000~X00Fが16進の“1234”以下の時、Y00FをONします。

PEX_WleH_1.bmp

② X000がONで、データレジスタD1000が16進の“5555”以下の時Y00FをONします。

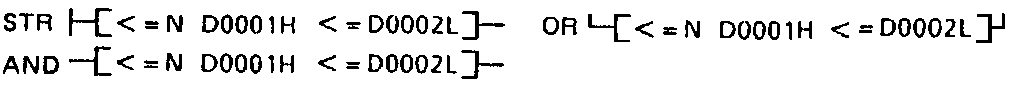
PEX_WleH_2.bmp

③ X000がONであるか、または、データレジスタD1000が16進の“AAAA”以下の時Y00FをONします。



23.<=N：バイトデータ比較（≦）\*

(1) シンボル



(2) 機能

2つの8ビットデータを比較し、1つ目のデータ値が2つ目のデータ値以下ならば導通状態になります。

(3) プログラム例

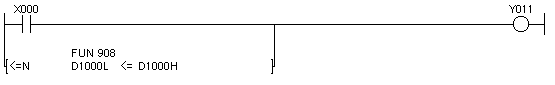
① データレジスタD0001の上位8ビットが、データレジスタD0002の下位8ビット以下であればY011をONします。

PEX_leN_1.bmp

② X000がONで、データレジスタD1000の下位8ビットがX010~X017で表される値以下であればY011をONします。

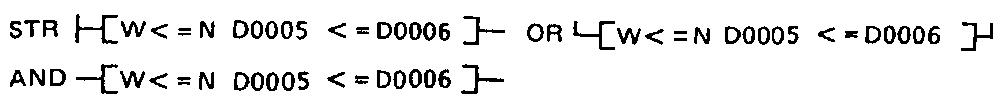
PEX_leN_2.bmp

③ X000がONであるか、または、データレジスタD1000の下位8ビットがD1000の上位8ビット以下ならばY011をONします。



24.W<=N：ワードデータ比較（≦）\*

(1) シンボル



(2) 機能

2つの16ビットデータを比較し、1つ目のデータ値が2つ目のデータ値以下ならば導通状態になります。

(3) プログラム例

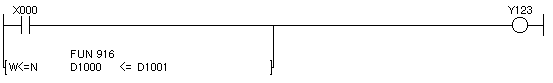
① カウンタC000の現在値(現在値レジスタN000の値)がカウンタC001の現在値 (現在値カウンタN001の値)以下であればY123をONします。

PEX_WleN_1.bmp

② X000がONで、X010~X01Fで表される値がD1000の内容以下であればY123をONします。

PEX_WleN_2.bmp

③ X000がONであるか、または、データレジスタD1000の内容がデータレジスタD1001の内容以下であればY123をONします。



#### データ転送

25.MOV：16進2桁定数転送　(FUN 100)

(1) シンボル

(2) 機能

16進定数をレジスタに格納します。定数の範囲は0~FFです。

(3) フラグ 変化なし

(4) プログラム例

M000がONしていれば、データレジスタD0000の下位8ビットに16進の“3F”を格納するプログラム

PEX_Mov

26.WMOV：16進4桁定数転送　(FUN 101)

(1) シンボル

(2) 機能

16進定数をレジスタに格納します。定数の範囲は0~FFFFです。

(3) フラグ 変化なし

(4) プログラム例

X001がOFFからONになった時、Y010～Y01Fに16進の“AAAA”を転送するプログラム

PEX_Wmov

#### データ変換

27.BCD：2進8ビット→BCD2桁変換　(FUN 154)

(1) シンボル



(2) 機能

OP1のレジスタに格納された2進データをBCDデータに変換して、OP2のレジスタに格納します。

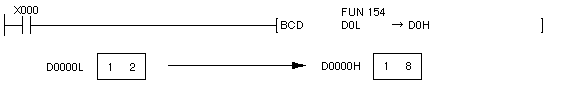
(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  |  |  |  |  | ↑ |

ｴﾗｰﾌﾗｸﾞ(ER) ： OP1のレジスタに格納されたデータをBCDコードに変換した際に､最大値99を超える時ONします｡

(4) プログラム例

X000がONの時、D0000Lの下位バイトに格納されている2進データをBCD2桁データに変換してD0000Hに格納します。



28.WBCD：2進16ビット→BCD4桁変換　(FUN 4)

(1) シンボル



(2) 機能

OP1のレジスタに格納された2進データをBCDデータに変換して、OP2のレジスタに格納します。

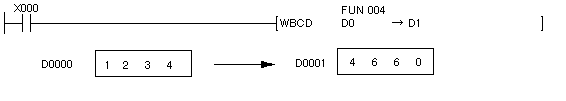
(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  |  |  |  |  | ↑ |

ｴﾗｰﾌﾗｸﾞ(ER) ： OP1のレジスタに格納されたデータをBCDコードに変換した際に､最大値9999を超える時ONします｡

(4) プログラム例

X000がONの時、D0000に格納されている2進データをBCD4桁に変換してD0001に格納します｡



29.BIN：BCD2桁→2進8ビット変換　(FUN 152)

(1) シンボル



(2) 機能

OP1のレジスタに格納されたBCDデータを2進データに変換して、OP2のレジスタに格納します。

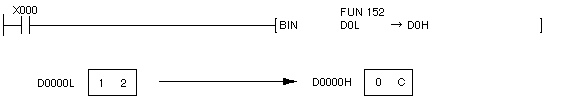
(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  |  |  |  |  | ↑ |

ｴﾗｰﾌﾗｸﾞ(ER) ： OP1のレジスタに格納されたデータがBCDフォーマットでない時ONします｡

(4) プログラム例

X000がONの時、D0000の下位バイトに格納されているBCD2桁データを2進データに変換してD0000の上位バイトに格納します。



30.WBIN：BCD4桁→2進16ビット変換　(FUN 3)

(1) シンボル



(2) 機能

OP1のレジスタに格納されたBCDデータを2進データに変換して、OP2のレジスタに格納します。

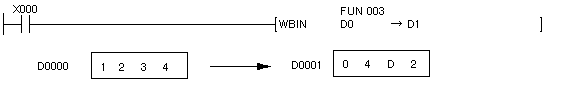
(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  |  |  |  |  | ↑ |

ｴﾗｰﾌﾗｸﾞ(ER) ： OP1のレジスタに格納されたデータがBCDフォーマットでない時ONします｡

(4) プログラム例

X000がONの時、D0000に格納されているBCD4桁データを2進データに変換してD0001に格納します｡



#### 算術演算

31.W+：ワードデータ2進加算　(FUN 92)

(1) シンボル

(2) 機能

OP1のレジスタの内容とOP2のレジスタの内容を加算し、結果をOP3のレジスタに格納します。データは全て2進数として扱われます。

(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
| ↑ |  | ↑ |  |  |  |  |

ｷｬﾘｰﾌﾗｸﾞ(CY) ： 加算の結果､桁上げが発生した時ONします｡

ｾﾞﾛﾌﾗｸﾞ(Z) ： 加算の結果が0の時ONします｡

(4) プログラム例

X001がONの時､カウンタC000の現在値(N000の内容)とカウンタC001の現在値 (N001の内容)を加算し､結果をD0234に格納します｡

PEX_Wadd

32.W-：ワードデータ2進減算　(FUN 93)

(1) シンボル

(2) 機能

OP1のレジスタの内容からOP2のレジスタの内容を引き、結果をOP3のレジスタに格納します。データは全て2進数として扱われます。

(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  | ↑ | ↑ |  |  |  |  |

ﾎﾞﾛｰﾌﾗｸﾞ(BO) ： 減算の結果桁下げが発生した（OP1の内容よりOP2の内容の方が大きかった）時にONします｡

ｾﾞﾛﾌﾗｸﾞ(Z) ： 減算の結果が0の時ONします｡

(4) プログラム例

X001がONの時､カウンタC000の現在値(N000の内容)から、カウンタC001の現在値(N001の内容)を引き､結果をD0234に格納します｡

PEX_Wsub

33.W\*：ワードデータ2進乗算　(FUN 94)

(1) シンボル



(2) 機能

OP1のレジスタの内容とOP2のレジスタの内容を乗算し、結果をOP3のレジスタに格納します。データは全て2進数として扱われます。

(3) フラグ

変化なし

(4) プログラム例

X001がONの時､カウンタC000の現在値(N000の内容)と、カウンタC001の現在値 (N001の内容)を乗算し､結果をD0234・D0235に格納します｡

PEX_Wmul

34.W/：ワードデータ2進除算　(FUN 175) #

(1) シンボル

(2) 機能

OP1のレジスタの内容をOP2のレジスタの内容で割り、商をOP3のレジスタに、余りをOP3の次のレジスタに格納します。

(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  |  |  |  |  | ↑ |

ｴﾗｰﾌﾗｸﾞ(ER) ： OP2のレジスタの内容が0の時ONします｡

(4) プログラム例

X001がONの時､カウンタC000の現在値(N000の内容)をカウンタC001の現在値 (N001の内容)で割り、商をD0234に、余りをD0235に格納します｡

PEX_Wd

35.WINC：ワードデータ2進インクリメント　(FUN 63)

(1) シンボル



(2) 機能

OP2のレジスタの内容を+1した後、OP1のレジスタの内容との比較を行います。データは2進数で扱います。

(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
| ↑ |  | ↑ |  | ↑ |  |  |

ｷｬﾘｰﾌﾗｸﾞ(CY) ： 演算後､OP2のレジスタの内容がオーバーフローした時ONします｡

ｾﾞﾛﾌﾗｸﾞ(Z) ： 演算後､OP2のレジスタの内容が0の時ONします｡

ｲｺｰﾙﾌﾗｸﾞ(=) ： 演算後､OP2のレジスタの内容がOP1のレジスタの内容と一致した時ONします｡

(4) プログラム例

X001がOFFからONになった時、D0123の内容を+1します。

PEX_Winc

36.WDEC：ワードデータ2進デクリメント　(FUN 64)

(1) シンボル

(2) 機能

OP1のレジスタの内容を-1します。データは2進数で扱います。

(3) フラグ

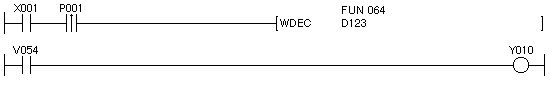
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  | ↑ | ↑ |  |  |  |  |

ﾎﾞﾛｰﾌﾗｸﾞ(BO) ： 演算結果が負になった時､すなわち演算前のデータが0であった時ONします｡

ｾﾞﾛﾌﾗｸﾞ(Z) ： 演算結果が0の時ONします｡

(4) プログラム例

X001がOFFからONになった時、D0123の内容を-1し、結果が0であればY010をONします。



#### 論理演算

37.AND：バイトデータ論理積　(FUN 13)

(1) シンボル



(2) 機能

OP1のレジスタの内容とOP2のレジスタの内容の論理積(AND)を行い、結果をOP3のレジスタに格納します。

(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  | ↑ |  |  |  |  |

ｾﾞﾛﾌﾗｸﾞ(Z) ： 演算結果が0の時ONします｡

(4) プログラム例

X000がONの時、X010～X017のデータとD1234の下位8ビットとの論理積を行い、結果をY018～Y01Fに出力します｡

PEX_And

38.OR：バイトデータ論理和　(FUN 14)

(1) シンボル



(2) 機能

OP1のレジスタの内容とOP2のレジスタの内容の論理和(OR)を行い、結果をOP3のレジスタに格納します。

(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  | ↑ |  |  |  |  |

ｾﾞﾛﾌﾗｸﾞ(Z) ： 演算結果が0の時ONします｡

(4) プログラム例

X000がONの時、X010～X017のデータとD0123の下位8ビットとの論理和を行い、結果をY028～Y02Fに出力します｡

PEX_Or

39.NOT：バイトデータ反転　(FUN 9)

(1) シンボル

(2) 機能

OP1のレジスタの内容を反転（各ビットを1ならば0に、0ならば1にする）したデータをOP2のレジスタに格納します。

(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  | ↑ |  |  |  |  |

ｾﾞﾛﾌﾗｸﾞ(Z) ： 演算結果が0の時ONします。

(4) プログラム例

X000がONの時、X010～X017のON/OFFデータを全て反転し、結果をY028～Y02Fに出力します｡

PEX_Not

40.XOR：バイトデータ排他的論理和　(FUN 18)

(1) シンボル

(2) 機能

OP1のレジスタの内容とOP2のレジスタの内容の排他的論理和(XOR)を行い、結果をOP3のレジスタに格納します。

(3) フラグ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CY | BO | Z | > | = | < | ER |
|  |  | ↑ |  |  |  |  |

ｾﾞﾛﾌﾗｸﾞ(Z) ： 演算結果が0の時ONします｡

(4) プログラム例

X000がONの時、X010～X017のデータとD0123の下位8ビットとの排他的論理和を行い、結果をY018～Y01Fに出力します｡

PEX_Xor

## 安全回路の基本ルール

TOYOPUC-Nano Safetyの安全制御プログラムには、安全制御に適した正しい信号を使うことが必要です。

TOYOPUC-Nano Safety用プログラミングソフトのPCwin-Safe-Nanoには、PCwin-Safe, PCwin-Safe-J, PCwin-Safe-Plusと同様、安全制御に適した正しい信号が使われているかどうかをチェックする機能が盛り込まれています。

安全制御に適した正しい信号(接点,コイル)には、”S”マークがつきます。また、安全制御に適さない使われかたをした信号(接点, コイル)には、“U”マークをつけます。この”S”マーク、“U”マークによって、安全制御設計者は、安全制御に適した正しい信号の使い方でプログラムを設計できます。

### ”S”(安全信号)マークの意味

* 安全制御に適した使い方をした安全信号の接点, コイルの脇に“S”マークを付けて表します。
* “S”マークが付いた安全信号は、非安全信号が安全信号よりも優先されてしまうような回路がないことを表します。
* 全ての出力(Q領域)が”S”マーク付きとなった場合に安全制御実行プログラムが生成され、TOYOPUC-Nano Safetyに書込みが可能になります。
* なお、“S”マーク信号のうち0016S-0.5A(1点でCat.2)、0806Sのシングル出力は”S2”と表されます。

### “U”(ユーザ確認信号)マークの意味

* 安全制御に適さない使い方をした信号の接点, コイルの脇に“U”マークを付けて表します。
* “U”マークが付いた信号は、安全信号と非安全信号のOR回路があり、非安全信号が安全信号よりも優先されてしまうようなことがあることを表し、ユーザにて回路の確認, 修正が必要になります。
* この信号は、たとえば非常停止ボタンが押されて入力信号がOFFしても、非安全信号がONしているかぎりコイルがOFFしないため、安全回路の出力(Q領域)として適切ではないため、使用することができません。

### ルールのポイント

* 安全信号“S”を非安全信号“－”でORした場合、コイルは“U”マークとなります。
* ただし、自己保持接点が非安全側にある場合のみ安全信号“S”を非安全信号“－”でORした自己保持回路は、コイルが非安全“－”となります。(原因Uマーク信号)
* “U”マーク接点が含まれた回路の場合、その出力コイルも“U”マークとなります。(派生Uマーク信号)
* ダブル入力とシングル入力のB接点は、非安全信号“－”となります。

### 基本ルール

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | "S"マークが付かない場合 | ①自己保持は原則"S"マークが付かない。 |  | |
| ②TOYOPUC-Nano Safetyの入力信号をB接点で使用すると"S"マークは付かない。 |  |  |
| ③通信領域のEL領域は"S"マークが付かない。 |  | |
| 2 | "S"マークが付く場合 | ①TOYOPUC-Nano Safetyへの出力は"S","S2"マークが付く  ※"S"マーク付きのみTOYOPUC- Nano Safetyにユーザープログラムを書込める。  ※IO-0016S-05A, IO-0806Sのシングル出力の場合、コイルは“S2“マークとなる。 |  | |
| ②TOYOPUC-Nano Safetyの入力信号は"S"マーク付きとなる。 |  |  |
| 3 | "U"マークが付く場合 | ①安全信号と非安全信号のORコイルは"U"マークが付く。 |  | |
| ②"U"接点を含んだコイルは"U"マークが付く。 |  | |
| 4 | EL領域への  出力制限なし | "S"マークなし, "U"マーク付きでも出力できる。 |  | |

表 10‑7　基本ルール

### "S"マークの付く条件

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | 識別子 | | 回路 | 説明 |
| 1 | I | ダブル  入力 |  | 実入力は"S"マークが付く。 |
| 2 |  | 実入力は"S"マークが付く。 |
| 3 |  | 実入力は"S"マークが付く。 |
| 4 |  | 実入力でも、B接点で制御するため  "S"マークが付かない。 |
| 5 | シングル  入力 |  | 実入力のA接点は"S"マークが付く。 |
| 6 |  | 単入力のB接点は"S"マークが付かない。 |
| 7 | Q | ダブル出力  もしくは  0016S-03A,  0016S-08Aの  シングル出力 |  | Q領域(出力)は、ユーザープログラム作成時、"S"マークまたは"S2"マークが付いていないと、TOYOPUC-Nano Safetyに書込むことができない。 |
| 0016S-05A,  0806Sの  シングル出力 |  |
| 8 | EL | |  | 送受信ともに"S"マークが付かない。 |
| 9-1 | M, K, T, L, D,GX/GY | |  | 接点の組み合わせ論理によって”S”  マーク付き、"S"マークなしが決まる。 |
| 9-2 |  |
| 10 | V, S, N | |  | "S"マークが付かない。 |

表 10‑8　"S"マークの付く条件 1/2

|  |  |  |  |
| --- | --- | --- | --- |
| No | 識別子 | 回路 | 説明 |
| 1 | GX/GY |  | 安全通信領域に設定した場合、  接点の組み合わせ論理によって“S“マーク付き、“S“マークなしが決まる。 |
|  |
| 2 |  | 安全通信領域に設定していない場合、“S“マークはつかない。 |

表 10‑9　"S"マークの付く条件 2/2

### コイルへのマーキング回路

接点の組み合わせ回路により、コイルは以下のように"S"マークなし, "S"マーク付き, "U"マーク付きとなります。

1. コイルに"S"マークが付かないパターン

* 回路パターン

|  |  |  |
| --- | --- | --- |
| No | 回路パターン | 説明 |
| (1)-1 | －  － | "S"マークのない接点だけで回路を組んだ場合、コイルには"S"マークが付かない。 |

表 9‑10　コイルへのマーキング回路

* 回路例

|  |  |  |
| --- | --- | --- |
| No | 回路例 | 説明 |
| 1 | M000  EL100 | EL(EL100)でコイル(M000)をおこす。 |
| 回路条件：8 |
| 回路パターン：(1)-1 |
| 2 | M001  M000 | "S"が付いていないコイル接点(M001)でコイル(M000)をおこす。 |
| 回路条件：9-2 |
| 回路パターン：(1)-1 |
| 3 | M000  ※単入力の場合も同じ  I002  I003 | B接点の入力信号領域(I002, I003)でコイル(M000)を起こす。 |
| 回路条件：4, 6 |
| 回路パターン：(1)-1 |

表 10‑11　回路例

1. コイルに"S"マークが付くパターン

* 回路パターン

|  |  |  |
| --- | --- | --- |
| No | 回路パターン | 説明 |
| (2)-1 | S  S | "S"マーク付きの接点で回路を組んだ場合、コイルに"S"が付く。 |
| (2)-2 | S  -  S | "S"マークなしと"S"マーク付きの接点でAND回路を組んだ場合、コイルに"S"が付く。 |
| (2)-3 | S  S  S | "S"マーク付きと"S"マーク付きの接点でOR回路を組んだ場合、コイルに"S"が付く。 |

表 10‑12　コイルに"S"マークが付くパターン

・未実装の場合は詰めて割り振る。

・BLANK設定で予約（16点）

・12点ユニットも16点とる

・基本をBLANK設定したとき、オプションのモジュール設定不可。BLANK設定は可

・リモート子局のBLANK設定は不可

PS,PNそれぞれ何種？

PS,PNそれぞれ何種？

あと６種は？

PN５４種は？

START

END

* 回路例

|  |  |  |
| --- | --- | --- |
| No | 回路例 | 説明 |
| 1 | M001  M000  S  S | "S"マーク付きの接点(M001)でコイル(M000)を起こす。 |
| 回路条件:9-1 |
| 回路パターン:(2)-1 |
| 2 | M000  S  S  S  I002  I003 | "S"マーク付きの入力信号領域(I002, I003)でコイル(M000)を起こす。 |
| 回路条件:1, 2, 3, 5 |
| 回路パターン:(2)-1 |
| 3 | S  S  M000  EL100  M001 | "S"マークなしと"S"マーク付きのAND回路でコイル(M000)を起こす |
| 回路条件:8, 9-1, 9-2 |
| 回路パターン:(2)-2 |
| 4 | M003  I012  M000  S  S  S | OR回路のA側, B側両方が"S"マーク付きでコイル(M000)を起こす。 |
| 回路条件:1, 2, 3, 5, 9-1 |
| 回路パターン:(2)-3 |

表 10‑13　回路例

1. コイルに”U”が付くパターン

"S"マークなしと"S"マーク付きでOR回路を組み立てるとコイルに”U”マークが付きます。

この”U”マークは安全信号の組み合わせでないことを示しています。

* 回路パターン

|  |  |  |
| --- | --- | --- |
| No | 回路パターン | 説明 |
| (3)-1 | U  S  － | "S"マーク付き、"S"マークなしでOR回路を組み立てた場合、コイルに"U"マークが付きます。"S"マークなしがあるため安全性が確保できないためです。 |

表 10‑14　コイルに”U”が付くパターン

“U”マークの付いた接点を使用した場合、その回路のコイルには必ず”U”マークが付きます。

* 回路パターン

|  |  |  |
| --- | --- | --- |
| No | 回路パターン | 説明 |
| (3)-2 | U  S  U | "U"の付いた接点と"S"の付いた接点とでAND回路を組み立てた場合、コイルに"U"マークが付く。 |
| (3)-3 | S  U  U | "S"の接点と"U"の接点とでOR回路を組み立てた場合、コイルに"U"マークが付く。 |

表 10‑15　回路パターン

* 回路例

|  |  |  |
| --- | --- | --- |
| No | 回路例 | 説明 |
| 1 | I012  M002  M000  S  U | "S"の付いた接点(I012)と"S"の付いていない接点(M002)のOR回路を組み立てた場合、コイル(M000)に"U"マークが付く。 |
| 回路パターン:(3)-1 |
| 2 | M004  M005  S  M000  U  U | "S"の付いた接点(M000)と"U"の付いた接点(M005)のAND回路を組み立てた場合、コイル(M004)に"U"マークが付く。 |
| 回路パターン:(3)-2 |
| 3 | U  U  S  I012  M000  M002 | "S"の付いた接点(I012)と"U"の付いた接点(M002)のOR回路を組み立てた場合、コイル(M000)に"U"マークが付く。 |
| 回路パターン:(3)-3 |

表 10‑16　回路例

### 自己保持回路パターン

* OR回路だけのパターン

|  |  |  |
| --- | --- | --- |
| No | 回路パターン | 説明 |
| (4)-1 | M\*\*\*  M\*\*\*  －  －  A側  B側  － | A側, B側の接点が共に"-"のOR回路の場合、自己保持回路のコイルは"-"となる。 |
| (4)-2 | M\*\*\*  M\*\*\*  S  －  A側  B側  － | A側の接点が"S"マーク付きでコイルが起きるが、B側の接点が"-"のため自己保持回路のコイルは"-"となる。 |
| (4)-3 | M\*\*\*  M\*\*\*  S  S  A側  B側  S | A側, B側の接点が共に"S"マーク付きとなる場合、自己保持回路のコイルは"S"マーク付きとなる。 |
| (4)-4 | M\*\*\*  M\*\*\*  －  S  A側  B側  U | A側の接点が"-"、B側の接点が"S"マーク付きの場合、自己保持回路のコイルは"U"マークとなる。 |
| (4)-5 | M\*\*\*  B側  U  A側  U  S  M\*\*\* | OR回路に"U"マーク付きの接点があるため自己保持回路のコイルは"U"マーク付きとなる。 |

表 10‑17　OR回路だけのパターン

* OR回路, AND回路を組み合わせたパターン

|  |  |  |
| --- | --- | --- |
| No | 回路パターン | 説明 |
| (4)-6 | B側  M\*\*\*  A側  M\*\*\*  －  －  －  －  M\*\*\* | OR回路は共に”－”のため”－”となる。  ”－”と”－”のAND回路となるためコイルは”－”となる。 |
| (4)-7 | M\*\*\*  M\*\*\*  －  S  －  － | OR回路は”S”マーク付きとなる。  “S”マーク付きと”－”のAND回路となるためコイルは一時的に”S”マーク付きとなるが、自己保持回路の接点が”－”のため”－”になる。 |
| (4)-8 | M\*\*\*  S  S  －  S | OR回路は”S”マーク付きとなる。  AND回路が”S”マーク付きになり、自己保持回路の接点が”S”マーク付きのためコイルは”S”マーク付きとなる。 |
| (4)-9 | －⇒S  －  S  S  M\*\*\*  M\*\*\* | 自己保持回路の接点は初期段階では”－”だが、コイルが”S”マーク付きとなった場合、”S”マーク付きとなる。 |
| －⇒S  －  S  S⇒U  A側  M\*\*\*  M\*\*\*  M\*\*\* | AND回路が”S”マーク付きのためコイルは”S”マーク付きとなる。コイルが”S”マーク付きのため自己保持回路の接点は”S”マーク付きとなるが、A側が”－”のためコイルが”U”マークとなる。 |
| U  －  S  U  M\*\*\* | 回路に”U”マーク付きがある場合、コイルは”U”マークとなる。 |

表 10‑18　OR回路, AND回路を組み合わせたパターン

* 回路例

|  |  |  |
| --- | --- | --- |
| No | 回路例 | 説明 |
| 1 | M000  I012  M000  M001  S  S  M014  S  S  A側  B側 | A側, B側のOR部は自己保持回路であるためOR全体で"S"が付かない。  "-"と"S"(M014)のANDであるためM000に"S"が付く。 |
| 回路パターン:(4)-2, (4)-10 |
| 2 | M000  I012  M000  M014  S  M016  S  S  S  S  A側  B側 | A側は"S"マーク付き回路であり、B側は自己保持接点(M000)と"S"マーク付き(M014)のANDであるため"S"マーク付きである。  そのためA側, B側のOR回路は"S"マーク付きとなる。  そのため自己保持回路であったとしても"S"マークとなる。 |
| 回路パターン:(4)-8 |
| 3 | M000  M000  M016  EL100  一旦”S”マーク付きになるが、自己保持接点に”S”マークを付けると”U”マークになる。  M001  U  U  M016  S  M001  EL100  M000  S  S  S  M000 | OR部は"-"と"-"の組み合わせのため"-"である。  全体では"-"と"S"マーク付きとのANDのため自己保持接点に"S"が付く。  しかしOR部が"-"と"S"マーク付きのORであるためコイルは"U"マークとなる。 |
| 回路パターン:(4)-9 |
| 4 | M000  EL100  M000  M001  S  M016  U  U  U | OR部は"U"マーク付き(M001)が含まれるため"U"マークとなり、AND部も"U"マークとなる。  そのためコイルが"U"マークとなり、自己保持接点も"U"マークとなる。 |
| 回路パターン:(4)-11 |

表 10‑19　回路