

TOSHIBA

汎用プログラマブルコントローラ

本体機能説明書

PROSEC

T3/T3H

安全のために次のことは必ず守って下さい

このたびは東芝プログラマブルコントローラ T3H/T3 をお買上げ頂き、誠にありがとうございます。

本書は T3H/T3 の仕様、機能、取り扱いや注意事項について説明しています。

T3H/T3 を安心して使用して頂くために、取り付け、運転、保守、点検の前に必ず本書と関連取扱説明書をすべて熟読し、機器の知識、安全情報、そして留意事項について習熟してから正しく使用して下さい。

【警告マークについて】

本書では、安全事項ランクを「注意」として区別してあります。



注意

： 取り扱いを誤った場合に、危険な状況が起こりえて、中程度の傷害や軽傷を受ける可能性が想定される場合及び物的損害の発生が想定される場合。

なお に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。

いすれも重要な内容を記載していますので必ず守って下さい。

安全のために次のことは必ず守って下さい

⚠ 注意

1. 運転中のプログラム変更、強制出力、RUN(運転)、HALT(停止)などの操作は十分安全を確認して行って下さい。
操作ミスや安全確認の怠りにより、機械の破損や事故が起こる恐れがあります。
2. I/O モジュールの交換は必ず電源を切った状態で行って下さい。
感電、誤動作、事故の原因となることがあります。
やむをえずI/O活線着脱機能を使用する場合は、作業の安全に十分注意して下さい。
3. 取扱説明書中に記載のサンプルプログラムは、お客様にて動作確認を行った後使用して下さい。
誤動作による事故を防ぐために運用前に十分確認を行って下さい。
4. 安全のため、作業、操作を行う前に上記と共に別冊の「T3/T3H 本体ハードウェア説明書」に記載している**安全上のご注意**を必ず熟読して下さい。

本文中での使用マークについて

【本文中でのマークについて】

補足

このワクは本書の中で必ず読んでいただきたい箇所についています。
T3/T3H の取扱い方や操作方法などで特に留意していただきたいことが書かれています。
必ずお読み下さい。

本書の目的

この説明書は、プログラマブルコントローラ T3/T3H の本体機能 (CPU 及び基本部ハードウェアによって実現できる機能) について、機能の内容と使用方法を説明したものです。またユーザプログラムを作成するために最低限必要な情報についても説明しています。

この説明書は以下の通り 3 部構成となっています。

本書の構成

第1部 基本プログラミング T3/T3H の基本機能のみを使用して、シーケンス制御主体のプログラムを作成する場合を想定し、プログラム作成に必要な情報及びプログラミングの手順を述べます。

第2部 本体機能説明 T3/T3H の機能をよく理解していくために、まず CPU の内部動作について説明し、次に機能について詳しく説明します。

第3部 ユーザプログラム説明 T3/T3H の機能をフルに活用したプログラムを作成するための情報について説明します。またプログラム言語として、ラダー図、SFC について説明します。第1部では省略した情報についても詳しく説明します。

T3/T3H を初めてご使用になる方は、プログラミングの基本を理解していくために、まず第1部をお読み下さい。その上で第2部と第3部を読んでいただければ、T3/T3H が提供する高度な制御機能と応用的な使用方法についてスムーズに理解していただけます。

T3/T3H を使用された経験のある方は、第1部を読み飛ばして、第2部と第3部を目的に応じてお読み下さい。このために巻末に索引を付けてあります。

構成上、第1部と第3部で内容が重複している部分がありますが、第1部での説明は理解しやすさのために部分的省略を行っている場合がある点をご承知おき下さい。

本書を読み始める前に

T3/T3H 説明書体系 T3/T3H システムの説明書として以下の説明書を準備しています。

「T3/T3H ハードウェア説明書」

T3/T3H 基本部、基本 I/O について、仕様、取り扱い方法、保守、保全方法を述べています。

「T3/T3H 本体機能説明書」(本書)

T3/T3H の CPU が持っている機能とその使用方法を説明し、ユーザプログラムを作成するために必要な情報について説明しています。

「T シリーズ命令語説明書 — ラダー図、SFC 編」

プログラム言語である、ラダー図と SFC について、各命令語の仕様詳細を説明しています。

「T シリーズプログラマ操作説明書 — 入門編」

パーソナルコンピュータに、T シリーズプログラム開発システム(T-PDS)を組み込み、プログラマとして機能させるための手順、及びプログラミングの基本操作について説明しています。

「T シリーズプログラマ操作説明書 — 応用編」

T シリーズプログラム開発システム(T-PDS)を使用して、プログラムを作成、デバッグ、プリントアウト及び保存するための操作方法について説明しています。

「T シリーズコンピュータリンク機能説明書」

CPU が内蔵しているコンピュータリンク機能について、仕様、取り扱い方法を述べています。

- * 「T3 アナログ入力モジュール取扱説明書」
アナログ入力モジュールの仕様、取り扱い方法について説明しています。
 - * 「T3 アナログ出力モジュール取扱説明書」
アナログ出力モジュールの仕様、取り扱い方法について説明しています。
 - * 「T3 パルス入力モジュール取扱説明書」
パルス入力モジュールの仕様、取り扱い方法について説明しています。
 - * 「T3 ASCII モジュール取扱説明書」
ASCII モジュールの仕様、取り扱い方法について説明しています。
 - * 「T3 位置決めモジュール取扱説明書」
2軸位置決めモジュールの仕様、取り扱い方法について説明しています。
 - * 「T3 状態変化検出付ディジタル入力モジュール」
状態変化検知モジュールの仕様、取り扱い方法について説明しています。
- 「データ伝送装置 TOSLINE-S20 概説書」
TOSLINE-S20 によるデータリンクシステムの中で T シリーズを使用する場合のシステム構成、機器構成及び TOSLINE-S20 の機能、性能、取り扱い方法について説明しています。
- 「データ伝送装置 TOSLINE-F10 概説書」
TOSLINE-F10 によるリモート I/O システムを構成する場合の、システム構成、機器構成、及び TOSLINE-F10 の機能、性能、取り扱い方法について説明しています。

*印: 「T3……説明書」となっておりますが T3/T3H で使用できます。

目次

- 安全上のご注意
- 本書を読み始める前に

第1部 基本プログラミング

1. 概要	
1.1 システム設計手順.....	3
1.2 基本プログラミング手順.....	4
2. 本体動作概説	
2.1 動作モードと機能.....	7
2.2 モード遷移条件.....	8
2.3 本体動作フロー概説.....	10
3. 入出力割り付け	
3.1 入出力割り付け.....	13
3.2 入力レジスタと出力レジスタ.....	14
3.3 入出力割り付けのルール.....	16
3.4 ユニット先頭アドレス設定機能.....	20
4. ユーザプログラム	
4.1 ユーザプログラムの構成.....	22
4.2 システム情報.....	23
4.3 ユーザプログラム.....	24
4.4 プログラムの実行順序.....	26
5. ユーザデータ	
5.1 ユーザデータの種別と機能.....	27
5.2 データ初期化条件.....	30
6. プログラム例とプログラミング手順	
6.1 シーケンス例.....	31
6.2 入出力割り付け.....	32
6.3 プログラム例.....	34
6.4 プログラミング手順.....	38

第2部 本体機能説明

1. 概要	
1.1 T3 システム構成	63
1.2 機能仕様一覧	64
2. 本体動作	
2.1 基本動作フロー	65
2.2 システム初期化処理	66
2.3 モード制御処理	68
2.4 スキャン制御	73
2.4.1 スキャンモード	75
2.4.2 一括入出力処理	77
2.4.3 タイマ更新処理	79
2.5 周辺サポート処理	80
2.6 プログラミングサポート機能	81
3. ユーザプログラム実行制御	
3.1 プログラム種別	84
3.2 メイン/サブプログラム実行制御	85
3.3 割り込みプログラム実行制御	92
4. 周辺メモリサポート機能	
4.1 EEPROM サポート	94
4.2 IC メモリカードサポート	95
5. RAS 機能	
5.1 概要	96
5.2 自己診断	96
5.3 イベント履歴登録機能	99
5.4 メモリプロテクト機能	101
5.5 瞬停検出機能	102
5.5.1 瞬停時間設定機能	102
5.5.2 瞬停継続機能	104
5.6 I/O エラーマップ作成機能	105
5.7 I/O 活線着脱機能	106
5.8 実行状態モニタ機能	108
5.9 サンプリングトレース機能	109

目次

5.10	ステータスラッチ機能.....	115
5.11	デバッグ支援機能.....	116
5.11.1	フォース機能.....	116
5.11.2	オンラインプログラム変更機能.....	117
5.11.3	デバッグモード機能.....	119
5.12	システム診断機能.....	128
5.13	パスワード機能.....	132

第3部 ユーザプログラム説明

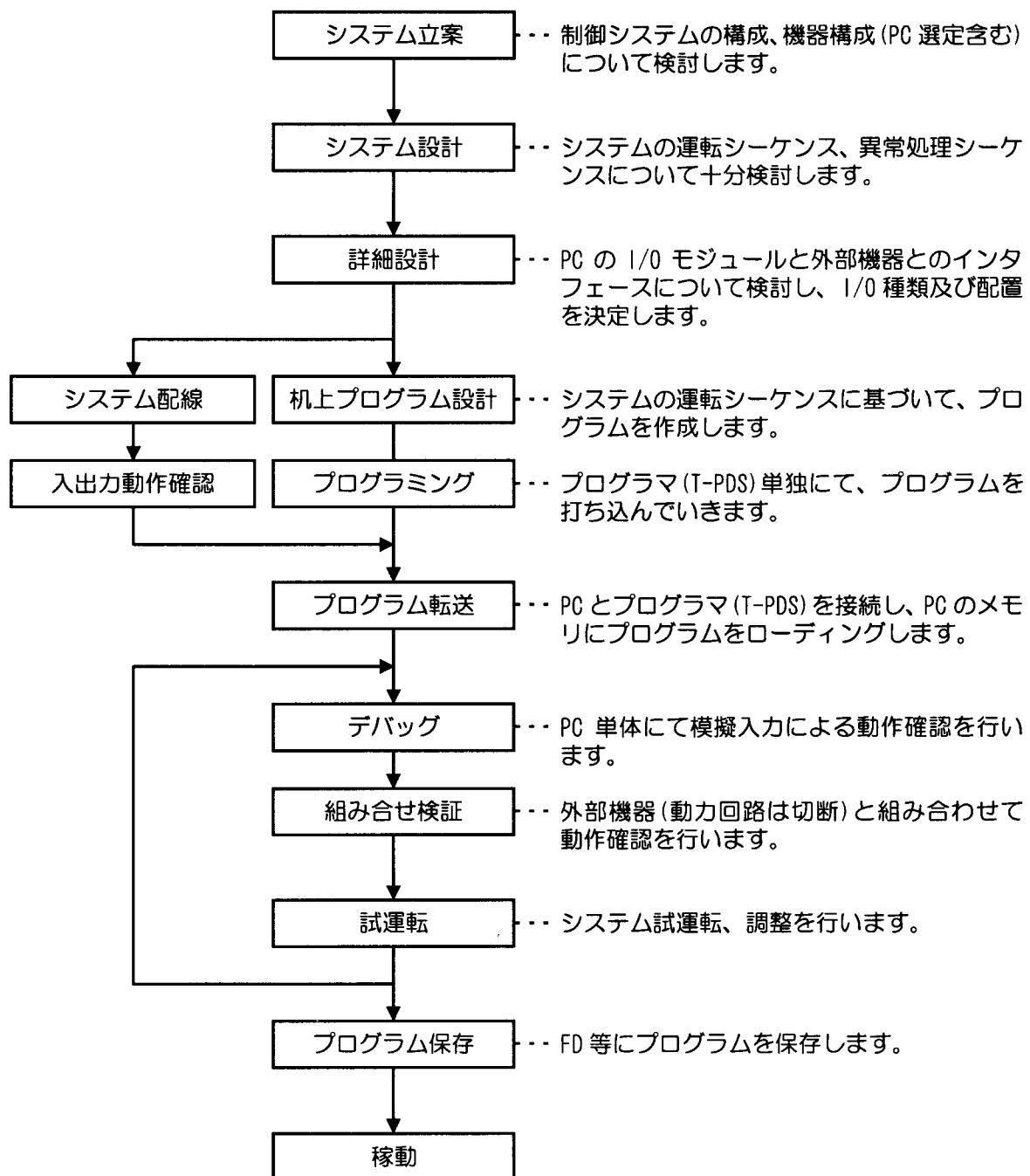
1. 概要	
1.1 第3部の目的	135
1.2 ユーザメモリ構成	135
2. ユーザプログラム構成	
2.1 ユーザプログラムの概要	137
2.2 システム情報	139
2.3 ユーザプログラム	142
2.3.1 メインプログラム	143
2.3.2 サブプログラム	144
2.3.3 割り込みプログラム	146
2.3.4 サブルーチン	149
2.4 コメント	151
3. ユーザデータ	
3.1 ユーザデータの概要	152
3.2 レジスタとデバイス	155
3.3 レジスタデータの扱い	184
3.4 インデックス修飾	191
3.5 衔指定	195
4. 入出力割り付け	
4.1 入出力割り付けの概要	200
4.2 入出力割り付けの方法	201
4.3 レジスタとモジュールの対応	205
4.4 伝送入出力割付け	209
5. プログラム言語	
5.1 プログラム言語の概要	217
5.2 ラダー図	220
5.3 SFC(シーケンシャルファンクションチャート)	227
5.4 プログラム作成上の補足	242
5.5 命令語一覧	244
索引	272

第1部

基本プログラミング

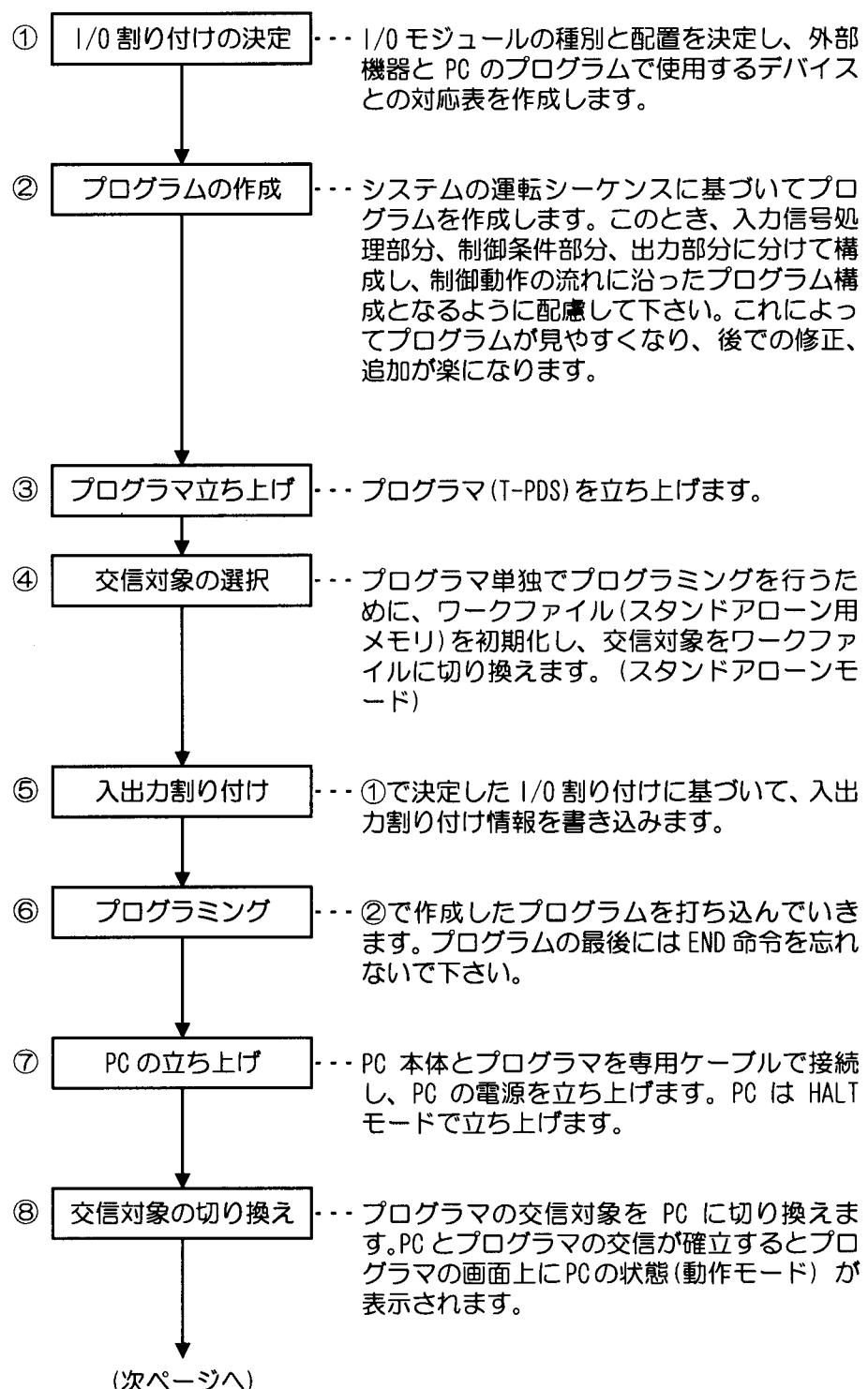
1.1 システム設計手順

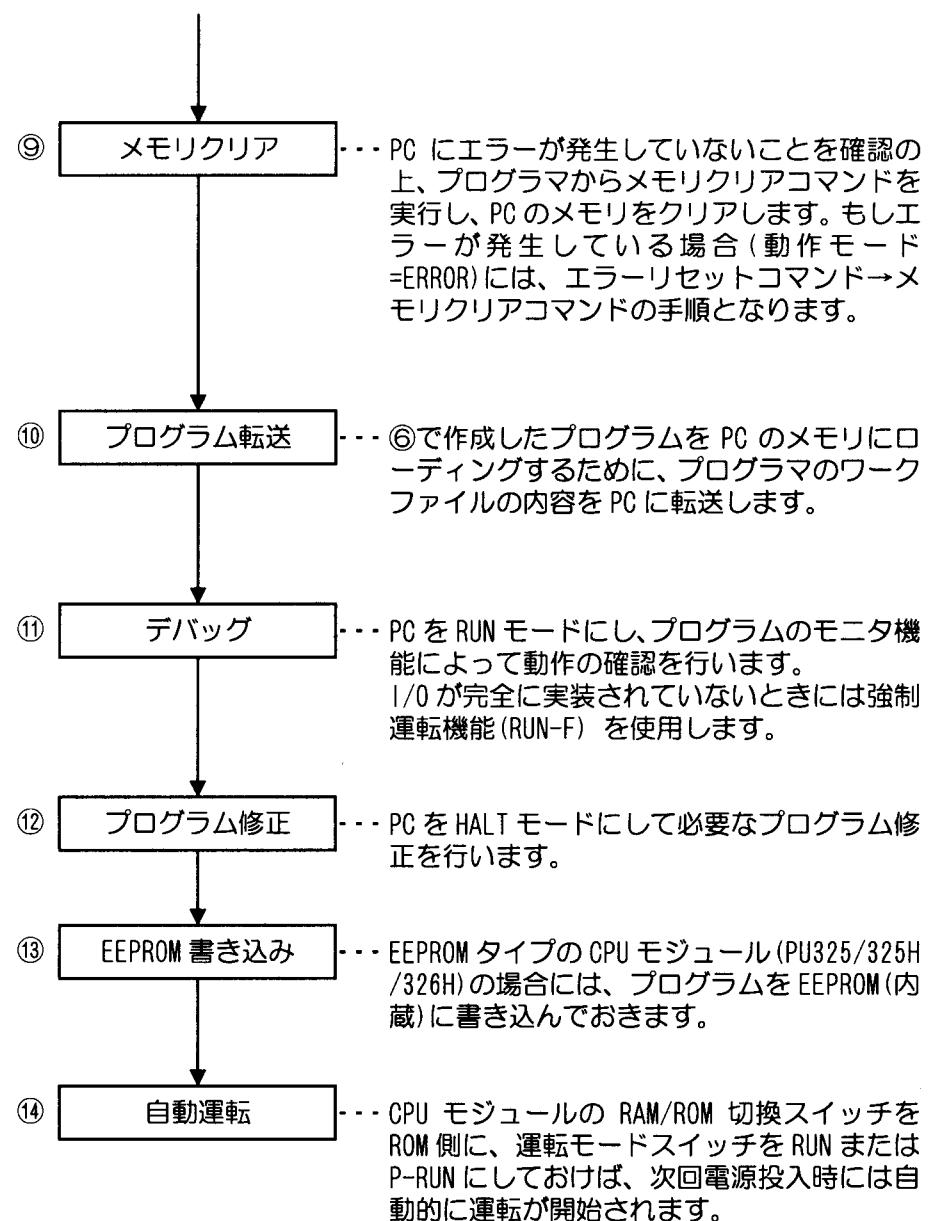
T3/T3H(以降、必要のない限り PC と称す)を適用した制御システムの設計は、通常以下の手順で行います。



1.2 基本プログラミング手順

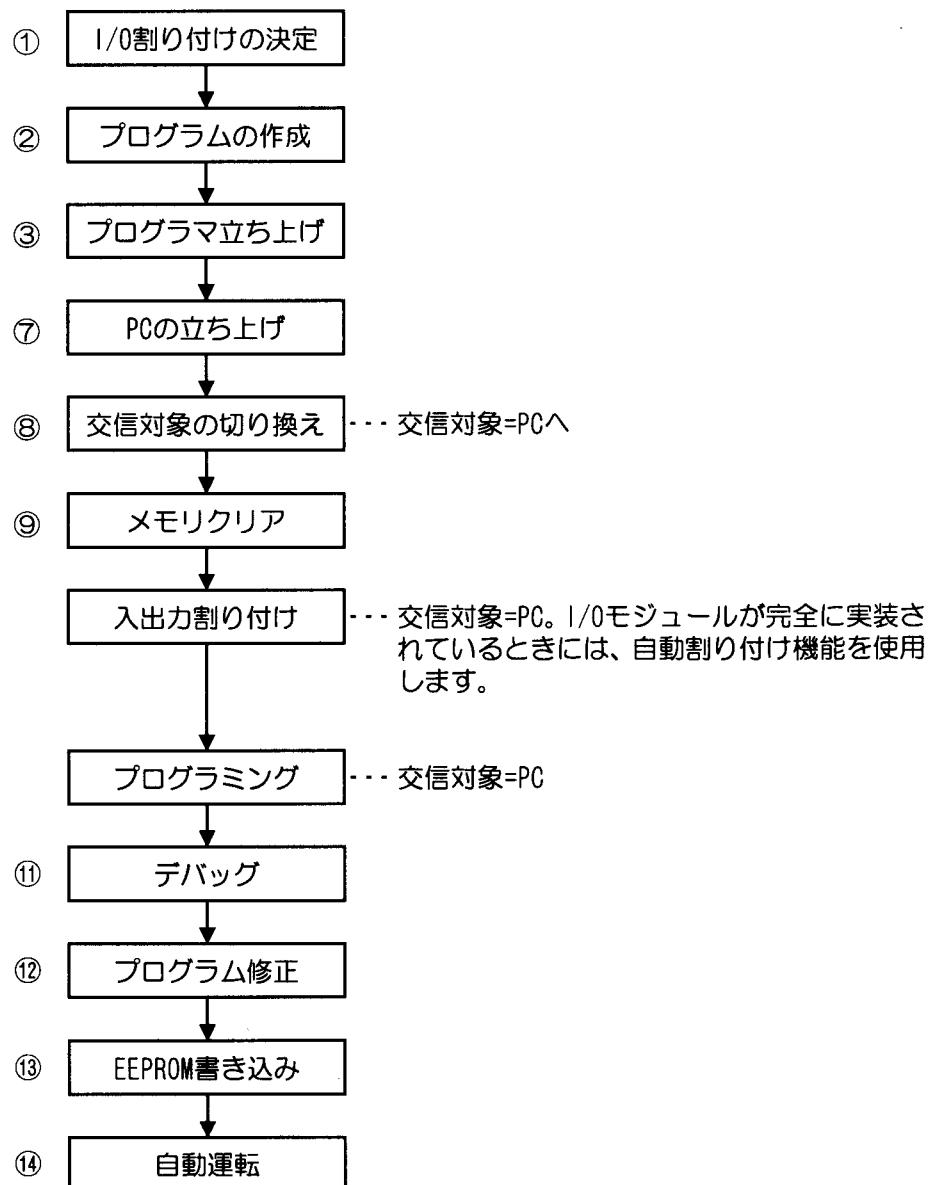
PC のプログラムを作成し、CPU に内蔵されているメモリにプログラムを書き込むための、基本的な手順を以下に示します。





上図の手順は、プログラマ(T-PDS)単独でプログラムを作成し、後で PC 本体にローディングする場合(オフラインプログラミング)の手順です。

PC とプログラマ (T-PDS) を接続した状態で、PC 本体のメモリに対してプログラムを書き込んでいく場合の手順は次のようにになります。
(数字は前ページのフローに対応)



— 補足 —
プログラムデバッグ及び試運転にあたっては、周囲の安全に十分注意して下さい。

2.1

動作モードと機能

PC の基本的な動作モードとして、RUN, HALT 及び ERROR の3つのモードがあります。また RUN モードのバリエーションとして、デバッグ用に RUN-F モードが準備されています。

RUN モード: 運転実行モードです。PC は外部入力の読み込み、ユーザプログラムの実行、外部出力の状態決定を繰り返し行います。(この 1 まわりのことをスキャンと呼びます)
プログラマを用いて、プログラム実行状態のモニタ、強制入出力などが行えます。

RUN-F モード: I/O モジュールが未実装の状態でも運転実行が可能なモード (ランエフ) です。(通常の RUN モードでは I/O 応答異常となります)
プログラムデバッグのために使用します。

HALT モード: 運転停止モードです。PC は出力を全て OFF にし、ユーザプログラムの実行を停止します。通常プログラミングはこのモードで行います。また EEPROM への書き込み (PU325/325H/326H) は、このモードでのみ可能です。

ERROR モード: エラーダウン状態です。PC は自己診断にて運転継続不可能な (エラー) エラーを検出すると、出力を全て OFF にし、ユーザプログラムの実行を停止して、ERROR モードに入ります。

ERROR モードでは PC 本体に対する書き込み操作は全て禁止されます。このモードを抜けるためには、プログラマからエラーリセットを行うか、または電源再投入が必要です。

*1) RUN モード及び RUN-F モードでもプログラムの変更は可能ですが(これをオンラインプログラム変更機能と呼びます)。ただし第1部では通常の HALT モードでのプログラミングだけを取り上げます。オンラインプログラム変更機能については第2部をご覧下さい。

*2) 上記の4つのモードの他に、実際にはさらに HOLD(ホールド)モードと DEBUG(デバッグ)モードがあります。これらについては第2部で説明します。

2.2

モード遷移条件

本体動作モードを決定/変更する要因としては、CPU モジュール上のモードスイッチ、プログラマの PC 制御コマンド、及び本体の自己診断があります。また、CPU モジュール上の RAM/ROM 切換スイッチは 2.3 節に述べるイニシャルロード機能の選択スイッチですが、T3 では電源投入後の動作モード決定にも影響を持ちます。

これらについて以下に説明し、次ページにモード遷移表を示します。

- モードスイッチ … HALT/RUN/P-RUN の3ポジションキースイッチ

スイッチ位置	動作モード
HALT	モードスイッチを RUN または P-RUN から HALT に切り換えると本体動作モードは HALT モードとなります。 また、モードスイッチを HALT にして電源を投入すると HALT モードで立ち上がります。
RUN	モードスイッチを HALT から RUN に切り換えると本体動作モードは RUN モードとなります。P-RUN から RUN に切り換えたときには、動作モードは変わりません。 RUN 位置で電源投入したときのモードは、T3 の場合 RAM/ROM 切換スイッチ、T3H の場合はプログラマ(システム情報)で設定するスタートモードにより決まります。
P-RUN	プロテクト RUN の意味です。動作モードとの関係は RUN の位置と同じですが、P-RUN の場合は、プログラム全体とデータレジスタの先頭 4K ワード (D0000～D4095) が書き込みプロテクトとなります。

- RAM/ROM 切換スイッチ (T3 の場合)

スイッチ位置	動作モード
RAM	電源投入時に RAM 位置にあると、モードスイッチが RUN または P-RUN であっても HALT モードで立ち上がります。
ROM	本体動作モードには影響を与えません。

- プログラマによるスタートモード(立上げモード)の設定 (T3H の場合)

設 定	動作モード
スタンバイ	モードスイッチが RUN または P-RUN であっても HALT モードで立ち上がります。
オート	本体動作モードには影響を与えません。

- プログラマの PC 制御コマンド … 停止/運転/強制運転

コマンド	コマンド実行後の動作モード
停止	HALT モードに切り換わります(モードスイッチが RUN または P-RUN のときのみ有効)
運転	RUN モードに切り換わります(モードスイッチが RUN または P-RUN のときのみ有効)
強制運転	RUN-F モードに切り換わります(モードスイッチが RUN または P-RUN のときのみ有効)

遷移前の状態			モード遷移要因	遷移後の動作モード	備考	
動作モード	RAM/ROM	モード SW				
電源投入前	T ₃ の場合	RAM	HALT/RUN	電源投入	HALT	イニシャルロードなし
			P-RUN	電源投入	HALT	イニシャルロードなし(プロテクト状態)
		ROM	HALT	電源投入	HALT	イニシャルロード実行
			RUN	電源投入	RUN	イニシャルロード後 RUN モード遷移
			P-RUN	電源投入	RUN	イニシャルロードなし(プロテクト状態)
	T _{3H} の場合	RAM	HALT	電源投入	HALT	イニシャルロードなし
			RUN	スタンバイ設定 電源投入	HALT	イニシャルロードなし
			オート設定	電源投入	RUN	イニシャルロードなし
			P-RUN	スタンバイ設定 電源投入	HALT	イニシャルロードなし(プロテクト状態)
		ROM	オート設定	電源投入	RUN	イニシャルロードなし(プロテクト状態)
			HALT	電源投入	HALT	イニシャルロード実行
HALT	RAM	RUN	スタンバイ設定	電源投入	HALT	イニシャルロード実行
			オート設定	電源投入	RUN	イニシャルロード実行後 RUN モード遷移
		P-RUN	スタンバイ設定	電源投入	HALT	イニシャルロードなし(プロテクト状態)
			オート設定	電源投入	RUN	イニシャルロードなし(プロテクト状態)
			—	初期化処理にてエラー検出	ERROR	
	ROM	HALT	モード SW →RUN	RUN		
		RUN/P-RUN	コマンド 運転	RUN		
			コマンド 強制運転	RUN-F		
		HALT	モード SW →RUN	RUN		イニシャルロード後 RUN モード遷移
		RUN	コマンド 運転	RUN		イニシャルロード後 RUN-F モード遷移
	—	P-RUN	コマンド 強制運転	RUN-F		イニシャルロードなし(プロテクト状態)
			モード SW →RUN	RUN		モード変化なし
		RUN	モード SW →P-RUN	HALT		モード変化なし(プロテクト状態)
			コマンド 停止	HALT		モード変化なし(プロテクト解除)
			HALT	コマンド(停止/運転/強制運転)	HALT	
	RUN	RUN/P-RUN	コマンド 停止	HALT		コマンド無効(モード変化なし)
		—	自己診断にてエラー検出	ERROR		
		—	モード SW →HALT	HALT		
			モード SW →P-RUN	RUN		モード変化なし(プロテクト状態)
			モード SW →RUN	RUN		モード変化なし(プロテクト解除)
		RUN/P-RUN	コマンド 停止	HALT		
			コマンド 運転	RUN		
			コマンド 強制運転	RUN		コマンド無効(モード変化なし)
			自己診断にてエラー検出	ERROR		
			モード SW →HALT	HALT		
RUN-F	—	RUN	モード SW →P-RUN	RUN-F		モード変化なし(プロテクト状態)
			モード SW →RUN	RUN-F		モード変化なし(プロテクト解除)
		RUN/P-RUN	コマンド 停止	HALT		
			コマンド 運転	RUN-F		
			コマンド 強制運転	RUN-F		コマンド無効(モード変化なし)
		—	自己診断にてエラー検出	ERROR		
ERROR	—	—	モード SW(HALT/RUN/P-RUN)	ERROR		
		—	コマンド(停止/運転/強制運転)	ERROR		無効
		—	コマンド エラーリセット	HALT		HALT モードに復帰

*1) 表中 RAM/ROM は RAM/ROM 切換スイッチを、モード SW はモードスイッチを示します。

*2) 一はスイッチの状態に依存しないことを示します。

*3) イニシャルロードについては次ページをご覧下さい。

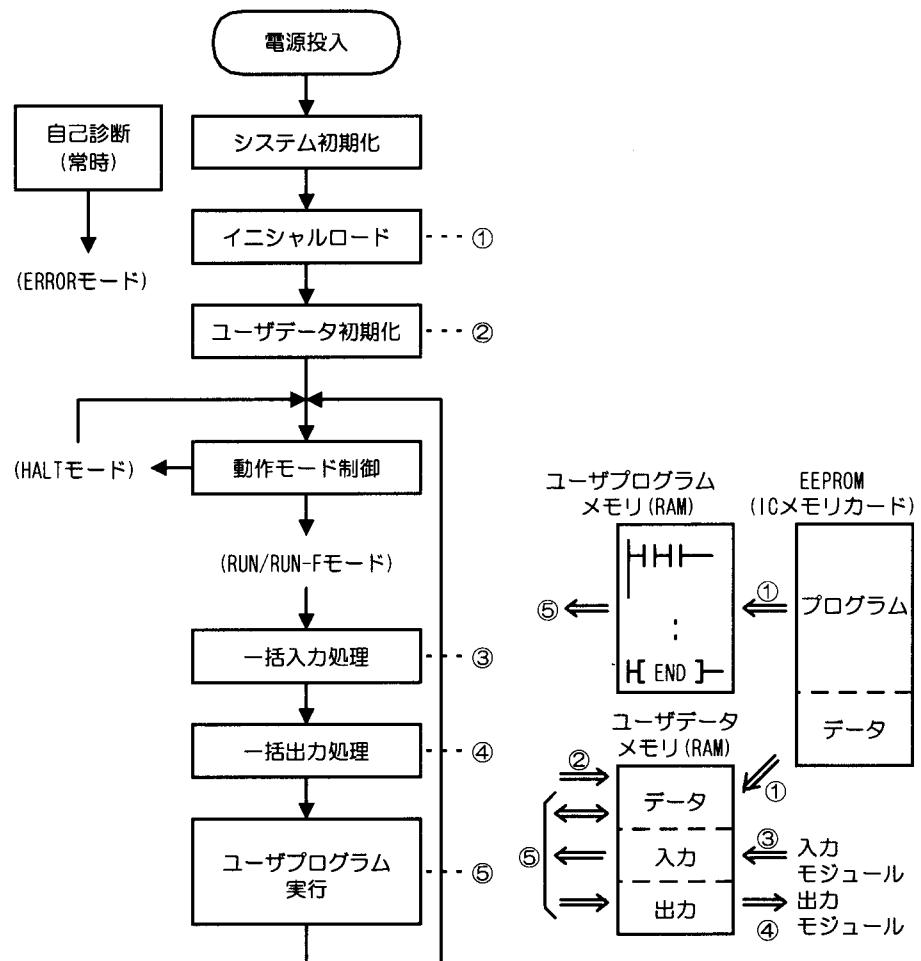
*4) スタンバイ設定、オート設定はプログラマで設定します。(スタートモードの設定)

2.3

本体動作フロー概説

PC本体の内部処理を意識しなくてもユーザプログラムは作れます。内部処理の概要を理解しておくことは、より効率の良いプログラムを作るため、また適切なデバッグを行うためにも有効です。

下図に大きな内部処理のイメージを示します。



①: イニシャルロード

EEPROM内蔵のCPUモジュール(PU325/325H/326H)において、RAM/ROM切換スイッチがROM側で、モードスイッチがP-RUN以外のとき、EEPROMに記憶されたプログラムとデータレジスタの前半(D0000～D4095)の内容がユーザプログラムメモリ(RAM)とユーザデータメモリ(RAM)に転送されます。

また、RAM/ROM切換スイッチがROM側で、RUNまたはRUN-Fモードに遷移する瞬間にもイニシャルロードが行われます(モードスイッチP-RUNのときを除く)。

(ICメモリカード装着時はICメモリカードがEEPROMに優先して転送元となります。ICメモリカードについては第2部で説明します)

②: ユーザデータ初期化

ユーザデータ(データレジスタ、タイマ、カウンタ、入力レジスタ、出力レジスタなど)を初期化します。ユーザデータについては第5章で説明します。

③: 一括入力処理

入力モジュールから外部入力信号の状態を読み込んで、ユーザデータの入力レジスタに格納します。(入力レジスタのことを入力イメージテーブルと呼ぶこともあります)

④: 一括出力処理

ユーザデータの出力レジスタの状態を出力モジュールに書き込みます。出力モジュールはこれに基づいて出力の ON/OFF 状態を決定します。(出力レジスタのことを出力イメージテーブルと呼ぶこともあります)

⑤: ユーザプログラム実行

ユーザプログラムメモリに記憶された命令語を1つづつ読み込んで、ユーザデータの内容を参照しながら出力レジスタの内容を書き換えていきます。PC の本質的な機能です。

動作モード制御からユーザプログラム実行までの1まわりを1スキャンと呼び、この1スキャンに要する時間をスキャン周期(またはスキャンタイム)と呼びます。

一般に、スキャン周期が短ければ短いほど、入力信号の変化に対する出力の応答が速いということになります。

補足

PC の RAM/ROM 切換スイッチ、モードスイッチと動作モード、イニシャルロード動作との関連で注意すべき項目を以下にまとめます。

- (1) T3 では電源投入時 RAM/ROM 切換スイッチが RAM 側であれば、常に HALT モードで立ち上がります。(自動運転には入りません)
従って、一般的な使用条件においては、CPU の種別にかかわらず、稼働時は RAM/ROM 切換スイッチを ROM 側にして使用することをお勧めします。
- (2) モードスイッチが P-RUN のときは、データレジスタの前半 4K ワード (D0000～D4095) は書き込みプロテクトとなり、一部の命令語についてはプログラム実行によるデータ書き込みに対しても書き込み禁止となります。(第2部 5.4 節参照)
- (3) EEPROM 内蔵の CPU (PU325/325H/326H) 使用時、または IC メモリカード装着時は、RAM/ROM 切換スイッチ ROM 側かつモードスイッチ P-RUN 以外の条件で、電源投入時及び RUN 起動時にイニシャルロードが行われます。従って、上記条件で使用する場合には、プログラムを変更したら必ず EEPROM または IC メモリカードに書き込んでおく必要があります。
- (4) イニシャルロードの対象となるのは、プログラム全領域、及び D レジスタの前半 4K ワード (D0000～D4095) です。従って D0000～D4095 を停電保持指定した場合であっても、イニシャルロードが行われた場合にはこれらのデータは EEPROM または IC メモリカードの内容に初期化されてしまいます。
- (5) EEPROM 内蔵の CPU (PU325/325H/326H)、または IC メモリカード装着において、RAM/ROM 切換スイッチ ROM 側の状態でモードスイッチを HALT →RUN→P-RUN と連続的に切り換えた場合、イニシャルロードの途中でプロテクト状態となるため正常な転送が行われません。このような場合には、HALT から RUN に切り換えた段階で、RUN モードに入ったことを確認してから、P-RUN に切り換えるようにして下さい。

3.1

入出力割り付け

2.3節で述べた通り、入力モジュールや出力モジュールとユーザプログラムとの間の信号のやり取りは、入力レジスタ及び出力レジスタを介して行われます。

入出力割り付けとは、入力/出力モジュールが、入力/出力レジスタのどの番地(アドレス)に割り当てられるかを決めてやることであり、基本的にはモジュールの実装状態によって自動的に決まります。従って、CPUにモジュールの実装状態を教えてやることを、入出力割り付けを行う、といいます。

入出力割り付けを行うためには、次の2つの方法があります。なおいずれの方法も、PCがHALTモードかつモードスイッチがP-RUN以外の位置であることが必要です。

(1) 自動割り付け

プログラマからPCに対して自動割り付けコマンドを実行します。PCのCPUは実装されているI/Oモジュールの種別(次ページの表参照)を読み取り、入出力割り付け情報として、ユーザプログラムメモリに記憶します。

(2) 個別割り付け

プログラマのI/Oカード割付情報の画面で、I/Oモジュールの装着位置とモジュールの種別を設定し、PC本体に書き込みます。

個別割り付けは、I/Oモジュールが完全に実装されていない状態でプログラミングを行う場合や、3.4節で述べるユニット先頭アドレス設定を行う場合に使用します。また、オフラインプログラミングを行う場合も個別割り付けを使用します。

これらの方法で、入出力割り付け情報がPCのメモリ内に記憶されると、I/Oモジュールと入力/出力レジスタの対応は、3.3節に述べるルールで自動的に決まります。

*) 個別割り付けによって、実際には次ページの表に示すモジュール種別以外の特殊割り付けを行うことができますが、ここでは説明を省略します。詳細は第3部にて説明します。

I/O モジュールの種別は、下表の通り、機能区分(X:入力、Y:出力、X+Y:入出力混合)と占有するレジスタ数(W)の組み合わせによって表わされます。

形 式	概 要	種 別
DI334/334H	32 点 DC 入力	X 2W
DI335/335H	64 点 DC 入力	X 4W
IN354/364	32 点 AC 入力	X 2W
D0333	16 点 DC 出力	Y 1W
D0334	32 点 DC 出力	Y 2W
D0335	64 点 DC 出力	Y 4W
AC363	16 点 AC 出力	Y 1W
AC364	32 点 AC 出力	Y 2W
R0364	32 点接点出力	Y 2W
R0363S	16 点独立接点出力	Y 1W
AD368	8チャネルアナログ入力	X 8W
DA364/374	4チャネルアナログ出力	Y 4W
CD332	状態変化検出付き8点 DC 入力	iX 1W
P1312	2チャネルパルス入力	iX+Y 2W
AS311	ASCII モジュール	iX+Y 4W
MC352	2軸位置決めモジュール	X+Y 4W
SN321/322/323/325	TOSLINE-S20 データ伝送モジュール	TL-S
MS311/RS311	TOSLINE-F10 データ伝送モジュール	TL-F
EN311	イーサネットモジュール	OPT

*) SN325, EN311 は T3H ver1.13 以降でサポート

3.2

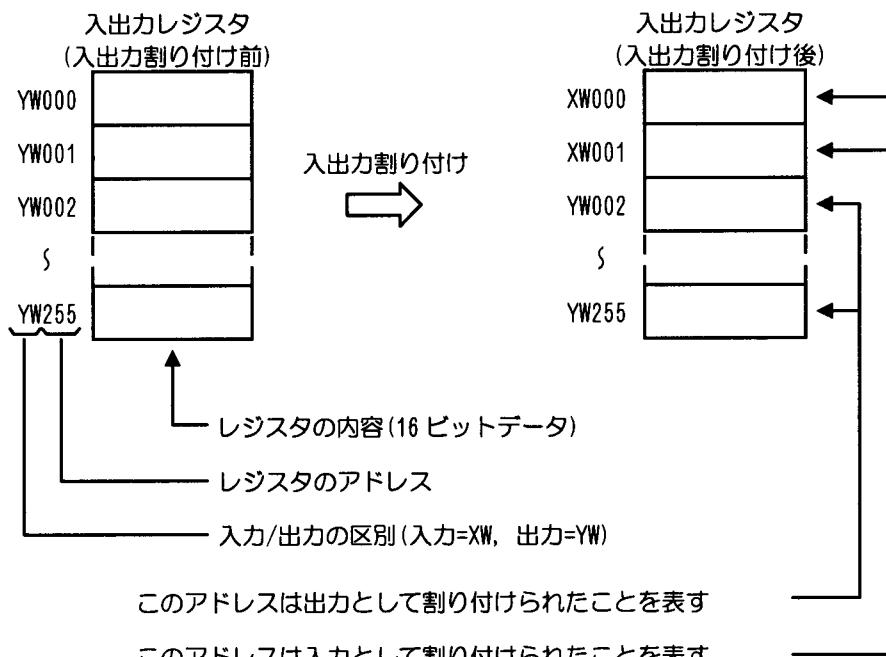
入力レジスタと 出力レジスタ

入出力割り付けとは、I/O モジュールと入力/出力レジスタとの間の対応付けを行うことであるということは前節で述べました。ここでは、入力レジスタ、出力レジスタの構成とアドレス表現方法などについて説明します。

ここまで説明では、入力レジスタと出力レジスタを別々のものとして扱ってきましたが、メモリ構成上は、これは正確ではありません。

実際には、入力レジスタと出力レジスタは、入出力レジスタと呼ぶ同一のメモリ領域を使用します。言い換えると、入出力割り付けを行う前は、入出力レジスタは入力と出力の色分けがされておらず、入出力割り付けを行うことによって、レジスタ単位(16 ビット単位)で入力と出力の色分けが行われます。(割り付け前は、内部的には全て出力レジスタとなっています)

イメージ的には下図のように考えることができます。



3

入出力レジスタは 16 ビットのレジスタであり、レジスタの個数としては T3 で 256 個、T3H で 512 個準備されています。(16 ビットということは、16 点分の ON/OFF 情報を格納するということです)

入出力レジスタは、ユーザプログラムで使用する場合、次のように表現します。

入力レジスタの場合 XW □□□

出力レジスタの場合 YW □□□

上記の□□□はレジスタのアドレス(またはレジスタ番号とも呼びます)を表し、000 から 511(T3 は 255)までの 10 進数です。

また入出力レジスタ内の各ビット(これをデバイスと呼びます)を次のように表現します。

入力レジスタ内のビット(入力デバイス)の場合 X □□□■

出力レジスタ内のビット(出力デバイス)の場合 Y □□□■

上記□□□はレジスタのアドレスを表し、■はレジスタ内のビット位置を表します。

ビット位置としては、0, 1, …, 9, A, B, C, D, E, F の 16 位置があります。

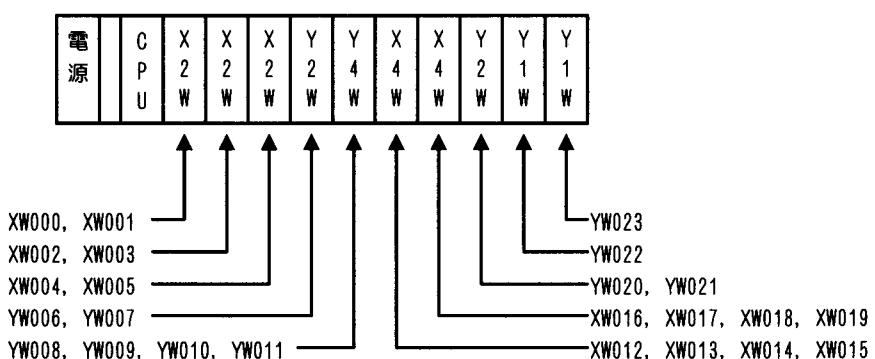
3.3

入出力割り付けのルール

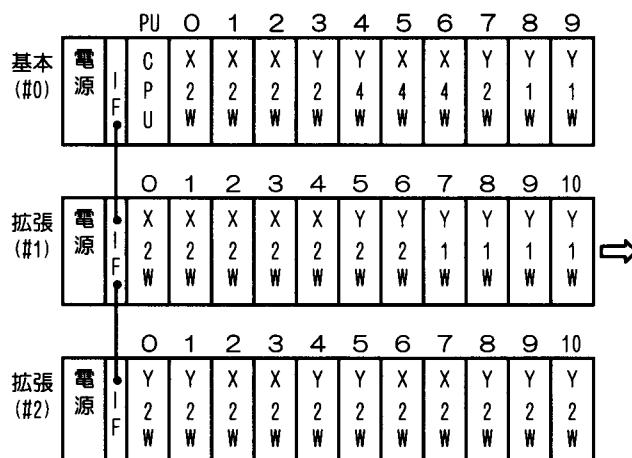
自動割り付け、または個別割り付けのいずれかの方法で、入出力割り付けが行われると、ユーザプログラムメモリ内に入出力割り付け情報（どの位置に、どの種別のモジュールが実装されるかという情報）が作成され、以下のルールにしたがってレジスタと I/O モジュールとの対応付けが行われます。

(1)～(7)は T3/T3H 共通の説明で、(8)は T3H のみの適用となります。

(1) 基本ユニットでは、CPU の右隣のモジュールから右へ、若いレジスタアドレスから順に割り付けられます。



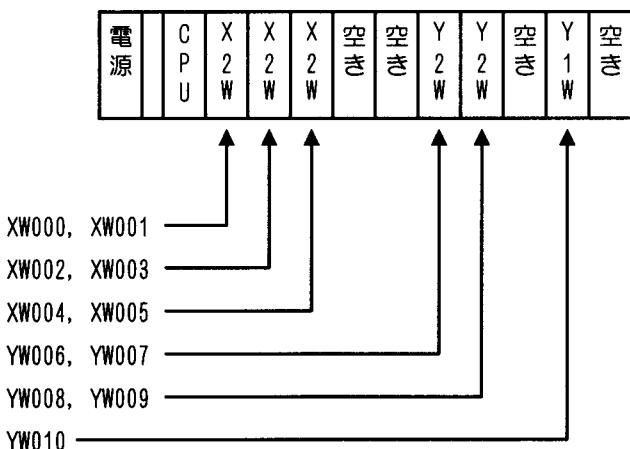
(2) 拡張ユニットの場合は、前段のユニットに引き続いて、左端のモジュールから順に右側のモジュールへ割り当てられます。



レジスタ割付表			
ユニット	スロット	種別	レジスタ
O	O	X 2W	XW000, XW001
S	S	S	S
9	Y 1W	YW023	
1	O X 2W	XW024, XW025	
1	X 2W	XW026, XW027	
2	X 2W	XW028, XW029	
3	X 2W	XW030, XW031	
4	X 2W	XW032, XW033	
5	Y 2W	YW034, YW035	
6	Y 2W	YW036, YW037	
7	Y 1W	YW038	
8	Y 1W	YW039	
9	Y 1W	YW040	
10	Y 1W	YW041	
2	O Y 2W	YW042, YW043	
S	S	S	S

*) 入出力割り付けにおいては、便宜上、モジュール実装位置をユニット番号とスロット番号の組み合わせで表します。
 ユニット番号 … 基本から順に #0, #1, #2, #3
 スロット番号 … 左端のモジュール装着位置から順に
 0, 1, 2, … 9 (拡張は10まで)

(3) モジュールが実装されていないスロット(個別割り付けにおいては、種別を設定しないスロット)はレジスタを占有しません。これを空きスロットと呼びます。



(4) I/O 5スロットの基本ベース(BU315)の場合は、入出力割り付け上は、I/O 10スロットの基本ベース(BU31A)と同じ扱いとなり、スロット5～9が空きと見なされます。

I/O 6スロットの拡張ベース(BU356)の場合も同様の扱いで、スロット6～10が空きと見なされます。

基本 (#0)	PU	O	1	2	3	4		
	電源	I F U	C P U	X 2W	X 2W	Y 2W	Y 1W	
拡張 (#1)			O	1	2	3	4	5
	電源	I F U	X 2W	X 2W	X 2W	Y 2W	Y 2W	Y 1W

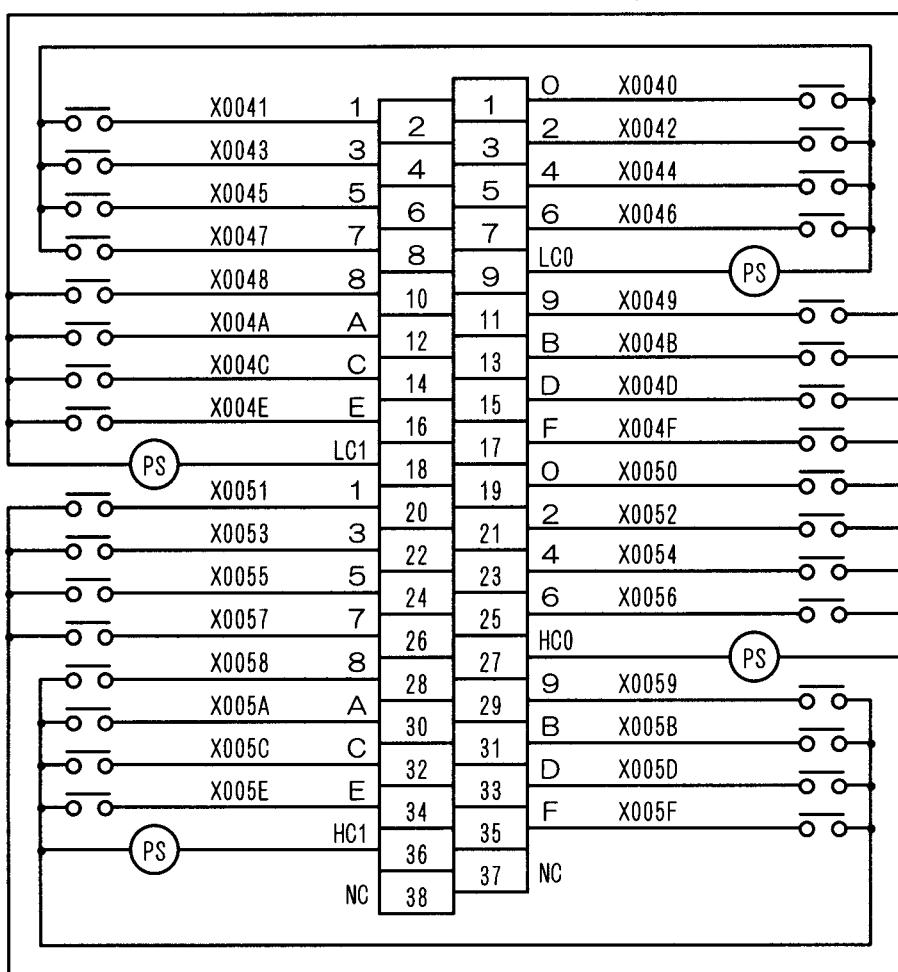
レジスタ割り付け表

ユ ニ ッ ト	ス ロ ッ ト	種別	レジスタ
0	O	X 2W	XW000, XW001
	1	X 2W	XW002, XW003
	2	Y 2W	YW004, YW005
	3	Y 1W	YW006
	4	Y 1W	YW007
	5	空き	—
	6	空き	—
	7	空き	—
	9	X 2W	—
	10	空き	—
1	O	X 2W	YW008, XW009
	1	X 2W	YW010, XW011
	2	X 2W	YW012, XW013
	3	Y 2W	YW014, YW015
	4	Y 2W	YW016, YW017
	5	Y 1W	YW018
	6	空き	—
	7	空き	—
	10	空き	—

- (5) I/O モジュールに入出力レジスタが割り付けられると、モジュール上の個々の外部信号はレジスタ上の各ビット(デバイス)に割り当てられます。このとき、複数レジスタが割り付けられるモジュールでは、LOW 側(LC 側)の信号に若いレジスタアドレスが割り当てられます。

(例)

32 点入力モジュール(X2W)に XW004 と XW005 が割り付けられた場合の入力信号と入力デバイスの対応



- (6) データ伝送モジュールなどの特殊機能モジュール(モジュール種別として、X, Y, X+Y, iX, iY, iX+Y が付かないモジュール)は、入出力レジスタを占有しません。
- (7) 割り付けされない入出力レジスタは、内部的には出力レジスタとなり、プログラム上で補助レジスタ/リレーと同じように使用することができます。

(8) T3H では拡張ユニットを2系統(最大3台/1系統)接続することができます。この場合、1系統目のユニットに引き続いて2系統目のユニットに装着されたモジュールに順番に割り当てられます。

	PU	0	1	2	3	4	5	6	7	8	9	10
基本 (#0)	電源	C P U	X 2 W	X 2 W	X 4 W	Y 4 W	X 2 W	X 2 W	Y 2 W	Y 2 W	Y 2 W	
		F	I	P	U							
1系統目												
拡張 (#1)	電源	X 4 W	X 4 W	X 4 W	X 2 W	Y 2 W	Y 2 W	Y 1 W	Y 1 W	Y 1 W	Y 1 W	
		F	I	P	U							
2系統目												
拡張 (#2)	電源	Y 2 W	Y 2 W	Y 2 W	X 2 W	Y 2 W	X 2 W	X 2 W	Y 2 W	Y 2 W	Y 2 W	
		F	I	P	U							
拡張 (#3)	電源	X 2 W	X 2 W	X 2 W	X 2 W	Y 2 W	Y 2 W	Y 2 W	Y 2 W	Y 2 W	Y 2 W	
		F	I	P	U							
拡張 (#4)	電源	X 2 W										
		F	I	P	U							
拡張 (#5)	電源	Y 2 W	Y 1 W	Y 1 W	Y 1 W							
		F	I	P	U							
拡張 (#6)	電源	X 2 W	X 4 W	X 4 W	X 2 W	X 2 W	Y 4 W	Y 4 W	Y 4 W	Y 2 W	Y 2 W	
		F	I	P	U							

レジスタ割付表

ユニット	スロット	種別	レジスタ
O	O	X 2W	XW000, XW001
	S	S	S
	9	X 2W	YW024, YW025
1	O	X 2W	XW026, XW027
	S	S	S
	10	Y 1W	YW047
2	O	Y 2W	XW048, YW049
	S	S	S
	10	Y 2W	YW068, YW069
3	O	X 2W	XW070, XW071
	S	S	S
	10	Y 2W	YW090, YW091

ユニット	スロット	種別	レジスタ
4	O	X 2W	XW092, XW093
	S	S	S
	10	X 2W	YW112, XW113
5	O	Y 2W	XW114, XW115
	S	S	S
	10	Y 1W	YW132
6	O	X 2W	XW133, XW134
	S	S	S
	10	Y 2W	YW163, YW164

3.4 ユニット先頭アドレス設定機能

入出力割り付けの特殊な機能として、各ユニットの先頭レジスタアドレスを設定することができる機能があります。これをユニット先頭アドレス設定機能と呼びます。

この機能は、プログラマからの入出力個別割り付けによって実現できます。この機能を使用すれば、将来モジュール追加を行った場合にも、レジスタアドレスはずれませんので、そのような場合に有効です。

基本 (#0)	電源	PU	O	1	2	3	4	5	6	7	8	9
		C P U	X 4 W	X 4 W	X 4 W	X 2 W	X 2 W	X 2 W	X 空 き	空 き	空 き	空 き
拡張 (#1)	電源	O	1	2	3	4	5	6	7	8	9	10
		Y 4 W	Y 4 W	Y 2 W	Y 2 W	Y 2 W	Y 2 W	Y 2 W	空 き	空 き	空 き	空 き

拡張 (#2)	電源	O	1	2	3	4	5	6	7	8	9	10
		X 2 W	X 2 W	X 2 W	X 2 W	X 2 W	X 2 W	X 空 き	空 き	空 き	空 き	空 き
拡張 (#3)	電源	O	1	2	3	4	5	6	7	8	9	10
		Y 2 W	Y 2 W	Y 1 W	Y 1 W	Y 1 W	Y 1 W	空 き	空 き	空 き	空 き	空 き

レジスタ割付表

ユ ニ ット	ユニット先頭アドレス指定	ス ロ ット	種別	レジスタ
0	000	PU	CPU	—
		O	X 4W	XW000～XW003
		1	X 4W	XW004～XW007
		2	X 4W	XW008～XW011
		3	X 2W	XW012, XW013
		4	X 2W	XW014, XW015
		5	X 2W	XW016, XW017
		6	X 2W	XW018, XW019
		7	空き	—
		8	空き	—
		9	空き	—
1	050	O	Y 4W	YW050～YW053
		1	Y 4W	YW054～YW057
		2	Y 2W	YW058, YW059
		3	Y 2W	YW060, YW061
		4	Y 2W	YW062, YW063
		5	Y 2W	YW064, YW065
		6	Y 2W	YW066, YW067
		7	空き	—
		8	空き	—
		9	空き	—
		10	空き	—

ユ ニ ット	ユニット先頭アドレス指定	ス ロ ット	種別	レジスタ
2	100	O	X 2W	XW100, XW101
		1	X 2W	XW102, XW103
		2	X 2W	XW104, XW105
		3	X 2W	XW106, XW107
		4	X 2W	XW108, XW109
		5	X 2W	XW110, XW111
		6	X 2W	XW112, XW113
		7	空き	—
		8	空き	—
		9	空き	—
3	150	O	Y 2W	YW150, YW151
		1	Y 2W	YW152, YW153
		2	Y 1W	YW154
		3	Y 1W	YW155
		4	Y 1W	YW156
		5	Y 1W	YW157
		6	空き	—
		7	空き	—
		8	空き	—
		9	空き	—
		10	空き	—

補足

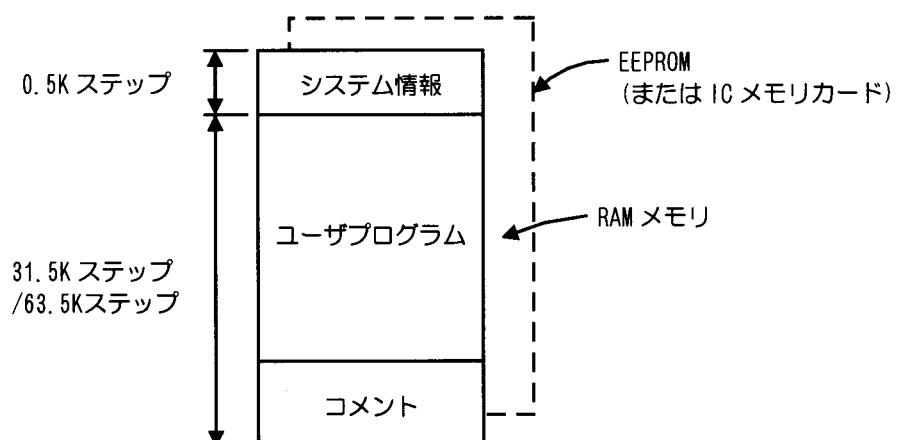
- (1) ユニット先頭アドレス設定機能を使用した場合も、ユニット間でレジスタアドレスが飛びこと以外は、3.3 節で述べた入出力割り付けのルールに従います。
- (2) 後段のユニットが若いレジスタアドレスとなるような設定はできません。例えば、ユニット#1 の先頭アドレス指定が 200 で、ユニット#2 の先頭アドレス指定が 100 というような設定はできません。
- (3) 入出力自動割り付けを行った場合には、全ユニットについて先頭アドレス指定なしとなり、レジスタは連続して割り付けられます。(3.3 節に述べた通り)

4.1 ユーザプログラムの構成

制御を行うための命令語の集まりをユーザプログラムと言います。これをアプリケーションプログラム、シーケンスプログラム、またはシーケンス回路と呼ぶこともありますが、本書ではユーザプログラムと呼ぶことにします。

ユーザプログラムを格納するメモリ領域をユーザプログラムメモリと言います。T3 は 32K ステップ(PU315, PU325)、T3H は 32K ステップ(PU325H)と 64K ステップ(PU326H)の容量があります。ただし、このうち 0.5K ステップはユーザプログラム付隨情報(これをシステム情報と呼びます)を記憶するために使用されますので、実際のユーザプログラムの容量は 31.5K ステップ / 63.5K ステップということになります。また、プログラムにコメントを付けた場合には、コメントもこの領域に記憶されます。

なお、ステップとは、命令語を構成する最小単位であり、命令語の種類により、1~10 ステップ/命令語となります。



- *1) EEPROM は、EEPROM 内蔵形の CPU (PU325/325H/326H) の場合です。IC メモリカード装着時は、EEPROM に優先して IC メモリカードが対象となります。第 1 部では説明を省略します。IC メモリカードについては第 2 部をご覧下さい。
- *2) EEPROM(または IC メモリカード)から RAM への転送(イニシャルロード)の条件については、2.3 節を参照下さい。
- *3) コメントについては第 3 部で説明します。

4.2 システム情報

システム情報とは、ユーザプログラムを実行する場合の実行制御パラメータやユーザプログラム管理情報を格納する領域で、0.5K ステップを占有します。システム情報には以下の内容が含まれます。

- (1) 本体パラメータ(機種タイプ、メモリ容量)
- (2) ユーザプログラム管理情報(プログラム ID、システムコメント、使用ステップ数、他)
- (3) 実行制御パラメータ(スキャンモード、サブプログラム実行条件、割り込みプログラム実行条件)
- (4) 停電保持範囲情報
- (5) 10mS タイマ範囲情報(T3H のみ)
- (6) スタートモード情報(T3H のみ)
- (7) 入出力割り付け情報
- (8) I/O 割り込み管理情報
- (9) データ伝送モジュール入出力情報
- (10) コンピュータリンク伝送パラメータ
- (11) ユーザ診断機能実行条件

これらのうち、(1)の本体パラメータと(2)の中の使用ステップ数については、CPU が自動的に設定/更新します。これ以外の項目は、ユーザがプログラマから設定します。

ここでは、上記(4)の停電保持範囲情報と、(7)の入出力割り付け情報についてのみ説明します。他の項目については第2部及び第3部で説明します。

- **停電保持範囲情報**

ユーザデータのうち、補助レジスタ(RW)、タイマレジスタ(T)、カウンタレジスタ(C)、及びデータレジスタ(D)について、停電/復電時に停電前のデータを保持する範囲を設定します。ここで設定した範囲以外のデータは、電源立ち上げ時のデータ初期化処理において、0クリア(デバイスについては OFF)されます。範囲設定は、上記レジスタ各々について、先頭アドレス(0)から指定アドレスまでを保持指定するという方式です。(詳細は 5.2 節を参照して下さい)

- **入出力割り付け情報**

第3章で説明した通り、自動割り付け、または個別割り付けを行うことによって、入出力割り付け情報がここに格納されます。CPU はこの情報に基づいて、入力/出力レジスタの割り付けを行い、また自己診断として割り付け情報にあるモジュールが正しく実装されているかどうかのチェックを行います。

4.3 ユーザプログラム

ユーザプログラムは、制御を行うための命令語の集まりです。

各PCの容量は次の通りです。

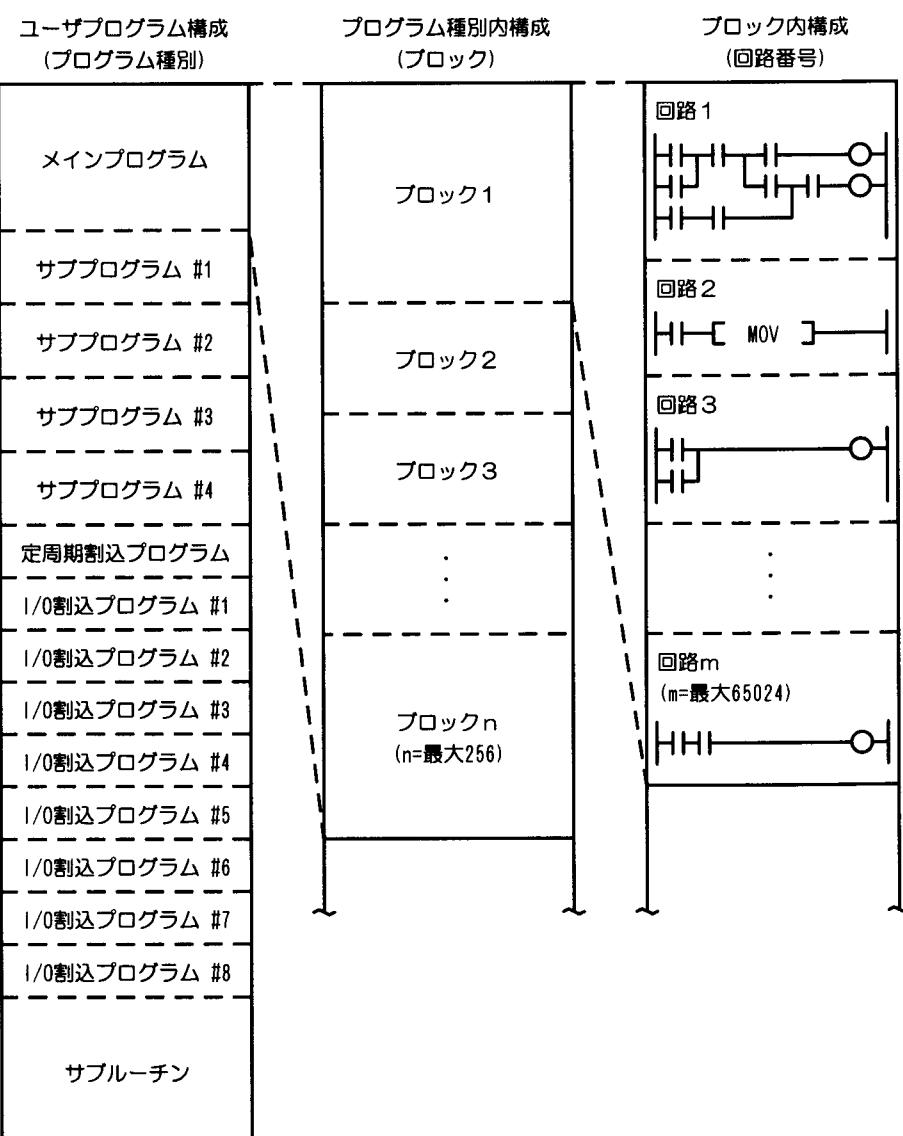
T3: PU315/PU325→31.5Kステップ

T3H: PU325H→31.5Kステップ

PU326H→63.5Kステップ

ユーザプログラムは下図のように各プログラム種別毎に記憶され、各々のプログラム種別の中ではブロックという単位で管理されます。また1つのブロックの中では回路番号によって管理されます(ラダー図の場合)。

従って、ユーザプログラムのモニタ/作成/編集においては、プログラム種別、ブロック番号及び回路番号を指定することによって、所定の回路を呼び出すことになります。



- プログラム種別

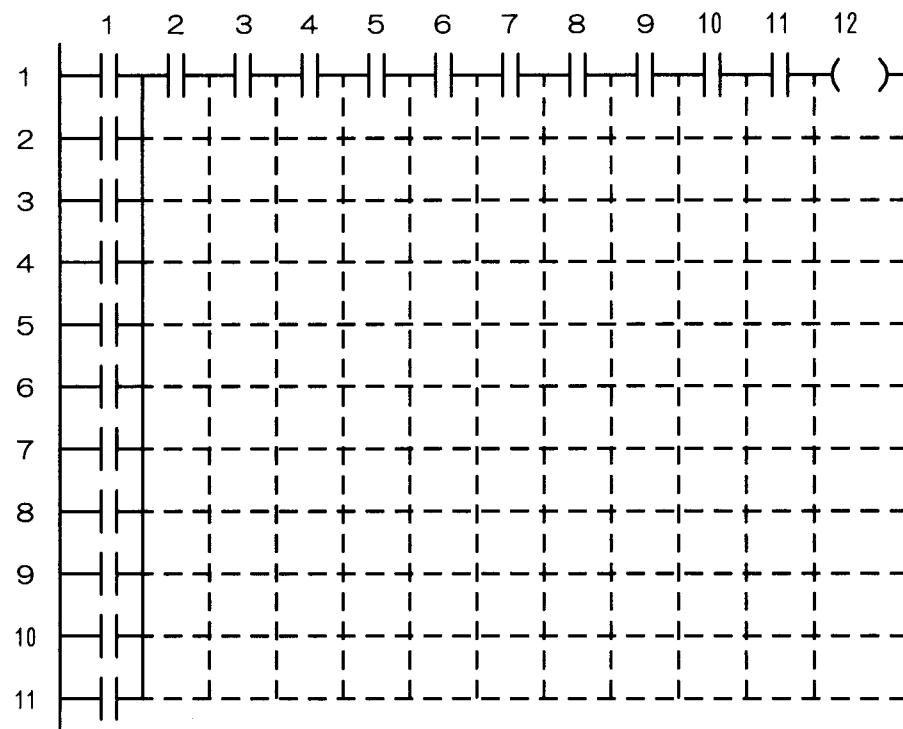
メインプログラム、サブプログラム (#1～#4)、定周期割込プログラム、I/O 割込プログラム (#1～#8)、及びサブルーチンの各種別があります。合計 31.5K/63.5K ステップ以内の容量制限はありますが、各プログラム種別毎の容量制限はありません。

- ブロック

ブロック番号としては、1 から 256 までが有効です。ブロック毎の容量制限(ステップ数制限)はありません。PC ではラダー図の他に、SFC といったプログラム言語が使用できますが、1 つのブロックの中で複数の言語を使用することはできません。つまり複数の言語を使用する場合にはブロックを切り換える必要があります。ラダー図のみを使用する場合には、ブロックを分ける必要はありません。

- 回路

ブロック内では、ユーザプログラムは回路番号によって管理されます。(ラダー図の場合)。回路とは、左右の母線以外の接続線によって接続された 1 かたまりを意味します。1 ブロック内に最大 65024 個の回路を作成することができます(実質無制限)。1 回路の大きさは、下図のように 11 行×12 列(最大 132 ステップ)に制限されます。



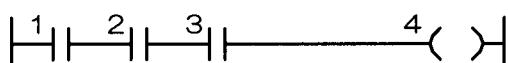
4.4 プログラムの実行順序

メインプログラムは毎スキャン実行されるユーザプログラムの本体であり、最低1つのEND命令がなければいけません。ここでは、メインプログラムのみの場合について、プログラムの実行順序を説明します。他のプログラム種別の動作については第2部で説明します。

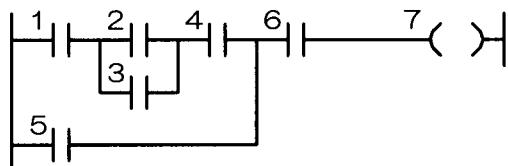
ユーザプログラムは次の順序で実行されます。

- ① メインプログラムの先頭ブロック(最も若い番号のブロック)から END命令が含まれるブロックまで順に実行されます。
- ② 1つのブロック内では先頭回路(回路1)から順に最終回路まで(END命令のあるブロックではEND命令のある回路まで)実行されます。
- ③ 1つの回路では次のルールに従って実行されます。

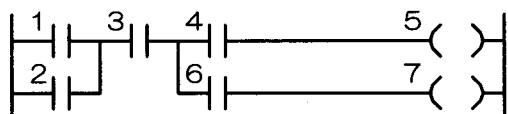
- (1) 上下接続がないときは、左から右へ実行されます。



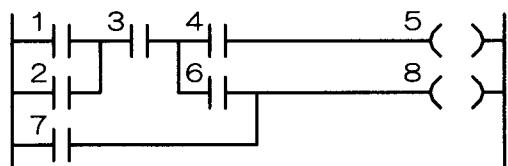
- (2) OR接続があるときは、OR論理部が先に実行されます。



- (3) 分流があるときは、上の行から下の行の順に実行されます。



- (4) 上記(2)と(3)の組み合わせ



補足

- (1) ブロック番号は連続していなくてもかまいません。つまり途中に空きブロックがあってもかまいません。
- (2) 回路番号は連続している必要があります。つまり途中に空白の回路は作成できません。

5.1 ユーザデータの種別と機能

外部入出力信号の状態や制御パラメータ、ユーザプログラムの実行途中演算結果などを格納するメモリ領域で、ユーザプログラムで使用できるものをユーザデータと呼びます。

ユーザデータは、扱いの上で、レジスタとデバイスの2つに分けて考えることができます。

レジスタとは、16ビットのデータを格納しておく場所であり、機能によって次の種類があります。

記号	名 称	機 能	T3		T3H	
			個 数	アドレス範囲	個 数	アドレス範囲
XW	入力レジスタ	入力モジュールからの入力データを格納(一括入力)	合計 256 ワード	WX000～ XW255	合計 512 ワード	WX000～ XW511
YW	出力レジスタ	出力モジュールへの出力データを格納(一括出力)		YX000～ YW255		YX000～ YW511
IW	直接入力レジスタ	入力モジュールからの直接入力データ(直接入力)		IW000～ IW255		IW000～ IW511
OW	直接出力レジスタ	出力モジュールへの直接出力データ(直接出力)		OW000～ OW255		OW000～ OW511
RW	補助レジスタ	ユーザプログラムの実行途中結果の一時記憶等に使用	512 ワード	RW000～ RW511	1000 ワード	RW000～ RW999
SW	特殊レジスタ	エラーフラグ、実行制御フラグ、カレンダデータ、タイミングクロック等を格納	256 ワード	SW000～ SW255	256 ワード	SW000～ SW255
T	タイマレジスタ	タイマ命令実行中の経過時間格納	512 ワード	T000～T511	1000 ワード	T000～T999
C	カウンタレジスタ	カウンタ命令実行中の現在カウント値を格納	512 ワード	C000～C511	512 ワード	C000～C511
D	データレジスタ	制御パラメータの格納や、実行結果の一時記憶等に使用	8192 ワード	D0000～ D8191	8192 ワード	D0000～ D8191
W	リンクレジスタ(TOSLINE-S20用)	データ伝送装置(TOSLINE-S20)とのデータ交換エリア	1024 ワード	W0000～ W1023	2048 ワード	W0000～ W2047
LW	リンクリレーレジスタ(TOSLINE-F10用)	データ伝送装置(TOSLINE-F10)とのデータ交換エリア	256 ワード	LW000～ LW255	256 ワード	LW000～ LW255
F	ファイルレジスタ	制御パラメータの格納や集収データの格納等に使用	8192 ワード	F0000～ F8191	32768 ワード	F00000～ F32767
I	インデックスレジスタ	命令語で使用するレジスタやデバイスのアドレス修飾に使用	1 ワード	I (アドレス はなし)	1 ワード	I (アドレス はなし)
J			1 ワード	J (アドレス はなし)	1 ワード	J (アドレス はなし)
K			1 ワード	K (アドレス はなし)	1 ワード	K (アドレス はなし)

*1) PCシステムでは、1ワード=16ビットとして扱い、レジスタ個数としてワードという単位を使用します。

*2) レジスタのアドレスは全て10進数です。

*3) タイマレジスタは、T3の場合T000～T063は0.01秒単位で増加(0.01秒タイマ)し、T064～T511は0.1秒単位で増加(0.1秒タイマ)します。

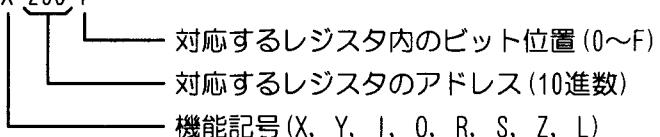
T3Hの場合、0.01秒タイマはT000～任意の範囲で指定できます。
(初期範囲設定はT000～T063になっています。)

一方、デバイスとは、1ビットデータ(ON/OFF情報)を格納する場所であり、機能によって次の種類があります。

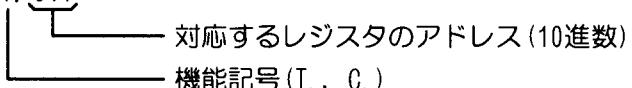
記号	名 称	機 能	T3		T3H	
			個 数	アドレス範囲	個 数	アドレス範囲
X	入力デバイス	入力モジュールからの入力データを格納(一括入力) XW レジスタ内の1ビットに対応	合計 4096 点	X0000～ X255F	合計 8192 点	X0000～ X511F
Y	出力デバイス	出力モジュールへの出力データを格納(一括出力) YW レジスタ内の1ビットに対応		Y0000～ Y255F		Y0000～ Y511F
I	直接入力デバイス	入力モジュールからの直接入力データ(直接入力)		I0000～ I255F		I0000～ I511F
O	直接出力デバイス	出力モジュールへの直接出力データ(直接出力)		00000～ 0255F		00000～ 0511F
R	補助リレー	シーケンスの内部中継用に使用 RW レジスタ内の1ビットに対応	8192 点	R0000～ R511F	16000 点	R0000～ R999F
S	特殊リレー	エラーフラグ、実行制御フラグ、タイミングリレー等を格納 SW レジスタ内の1ビットに対応	4096 点	S0000～ S255F	4096 点	S0000～ S255F
T.	タイマリレー	タイマ命令の実行結果を反映 同じアドレスのTレジスタ動作に対応	512 点	T. 000～T. 511	1000 点	T. 000～T. 999
C.	カウンタリレー	カウンタ命令の実行結果を反映 同じアドレスのCレジスタ動作に対応	512 点	C. 000～C. 511	512 点	C. 000～C. 511
Z	リンクレジスタリレー (TOSLINE-S20 用)	データ伝送装置(TOSLINE-S20)とのデータ交換エリア W レジスタの上位 512 ワード内の1ビットに対応	8192 点	Z0000～ Z511F	16000 点	Z0000～ Z999F
L	リンクリレー (TOSLINE-F10 用)	データ伝送装置(TOSLINE-F10)とのデータ交換エリア LW レジスタ内の1ビットに対応	4096 点	L0000～ L255F	4096 点	L0000～ L255F

デバイスのアドレス表現は以下の通りです。

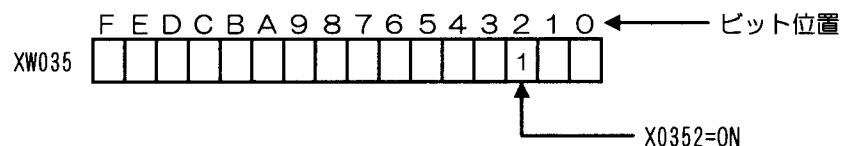
T. と C. 以外 X 255 F



T. と C. T. 511



従って、例えばデバイス X0352 とはレジスタ XW035 のビット2を表し、X0352 がON とは XW035 のビット2が1であることを示します。



*1) レジスタで数値を扱うときの最下位ビットはビット0です。

*2) 直接入力レジスタ/デバイス(IW/I)を命令語で使用した場合、その命令語を実行した時に、入力モジュールから直接、入力データが読み込まれます。(この方式を直接入力方式といいます)

これに対し、入力レジスタ(XW)には、ユーザプログラム実行の前に一括して、対応する入力モジュールから入力データが読み込まれます。(この方式を一括入力方式といいます)

入出力割り付け上は、同じアドレスの IW と XW は同じ入力モジュールに対応します。

*3) 直接出力レジスタ/デバイス(OW/O)を命令語で使用した場合、その命令語を実行した時に、直接そのデータが出力モジュールに出力されます。(この方式を直接出力方式といいます)

これに対し、出力レジスタ(YW)の内容は、ユーザプログラム実行の前に一括して、対応する出力モジュールに出力されます。(この方式を一括出力方式といいます)

入出力割り付け上は、同じアドレスの OW と YW は同じ出力モジュールに対応します。

*4) レジスタ/デバイスの詳細は第3部をご覧下さい。

5.2

データ初期化条件

ユーザデータは、電源立ち上げ時及びRUNモード遷移時に、下表の条件に従って初期化されます。

また、データレジスタの先頭4Kワード(D0000～D4095)はイニシャルロードの対象となり、イニシャルロード条件が成立している場合には、イニシャルロード→データ初期化の順で初期化が行われます。(イニシャルロードについては2.3節を参照下さい)

レジスタ/デバイス	初期化状態
入力レジスタ/デバイス(XW/X)	フォースされた入力デバイスについては以前の状態を保持し、それ以外は0クリア
出力レジスタ/デバイス(YW/Y)	コイルフォースされた出力デバイスについては以前の状態を保持し、それ以外は0クリア
補助レジスタ/リレー(RW/R)	停電保持指定されたレジスタ及びコイルフォースされたデバイスについては以前の状態を保持し、それ以外は0クリア
特殊レジスタ/リレー(SW/S)	SW000～SW063はCPU設定部は初期化、ユーザ設定部は保持、SW064以降は初期化
タイマレジスタ/リレー(T/T.)	停電保持指定されたレジスタとそれに対応するデバイスについては以前の状態を保持し、それ以外は0クリア
カウンタレジスタ/リレー(C/C.)	
データレジスタ(D)	停電保持指定されたレジスタについては以前の状態を保持し、それ以外は0クリア また、モードスイッチがP-RUNのときは先頭4Kワード(D0000～D4095)は保持
リンクレジスタ/リレー(W/Z)(TOSLINE-S20用)	コイルフォースされたリンクレジスタリレー(送信エリア)については以前の状態を保持し、それ以外は0クリア
リンクリレー/レジスタ(L/LW)(TOSLINE-F10用)	コイルフォースされたリンクリレー(送信エリア)については以前の状態を保持し、それ以外は0クリア
ファイルレジスタ(F)	全て保持
インデックスレジスタ(I, J, K)	全て0クリア

*) 停電保持指定は、RW, T, C, Dの各レジスタについて可能です。

プログラマのシステム情報設定機能により、保持領域を指定します。

各レジスタについて、先頭アドレス(0)から指定したアドレスまでが停電保持領域となります。

プログラマの停電保持指定画面

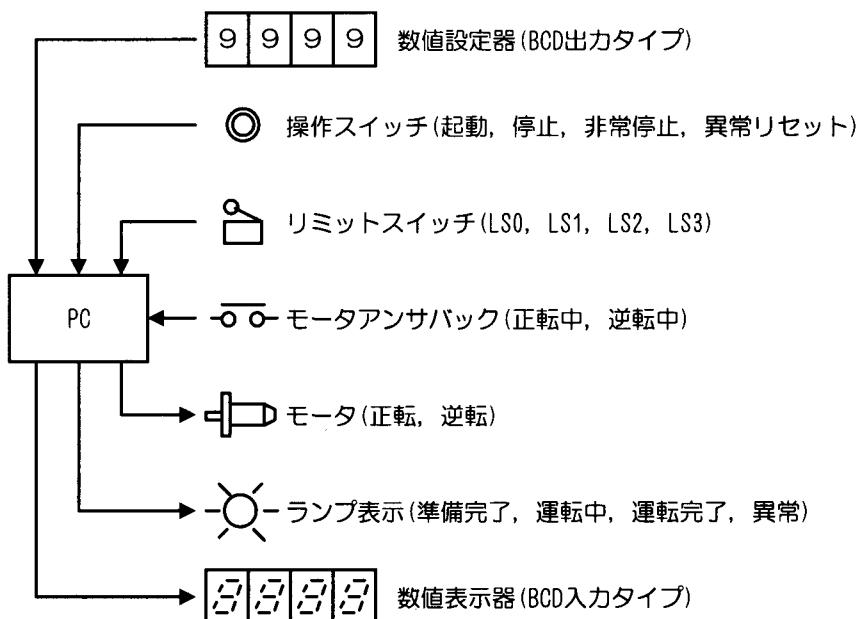
13. 停電保持範囲指定

RW000	～	[]
T000	～	[]
C000	～	[]
D0000	～	[]

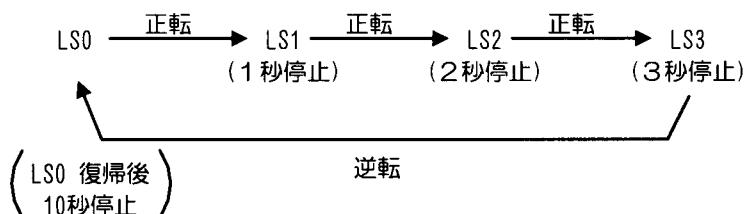
6.1 シーケンス例

この章では、簡単なシーケンスを例にあげて、入出力割り付け、プログラム作成、さらに実際のプログラミング操作手順を示します。PC を使用する場合の参考にしてください。

例として、下図のシーケンスを考えてみます。



- ① LS0 が ON している状態で起動スイッチが押されると次の動作を行う。



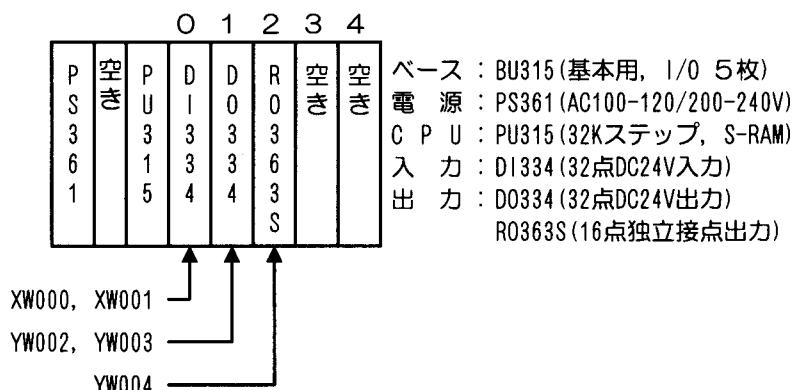
- ② 上記動作を数値設定器で設定された回数だけ繰り返し、動作中は運転中ランプを点灯させると共に数値表示器にその時点の実行回数を表示する。動作完了すると運転中ランプを消灯し、運転完了ランプを点灯させる。
- ③ 運転中に停止スイッチが押されると、モータをその場で停止させ、1秒後逆転を開始させる。LS0 が ON したらモータを停止させ、1秒後準備完了ランプを点灯させる。

- ④ 運転中以外の状態で LS0 が ON しているときには、準備完了ランプを点灯させる。起動スイッチは準備完了ランプが点灯しているときのみ有効とする。
 - ⑤ 非常停止スイッチが押された場合には、モータをその場で停止させ、異常ランプを点灯させる。この状態で、異常リセットスイッチが押されたら、異常ランプを消灯させる。

6.2

まず、モジュール構成を決定し、外部信号とレジスタ/デバイスとの対応表を作ります。ここでは、下記のモジュール構成、割り付けとします。

- #### • モジュール構成とレジスタ割り付け



- #### • 入出力マップ

XW000(数値設定器)		XW001(スイッチ類)	
X0000		X0010	非常停止(常時ON)
01	$\times 10^0$	11	異常リセット
02		12	起動
03		13	停止
04		14	
05	$\times 10^1$	15	
06		16	
07		17	
08		18	LS0
09	$\times 10^2$	19	LS1
0A		1A	LS2
0B		1B	LS3
0C		1C	アンサバック正転中
0D	$\times 10^3$	1D	アンサバック逆転中
0E		1E	
0F		1F	

YW002(数値表示器)		YW003(ランプ表示)		YW004(モータ)	
Y0020	$\times 10^0$	Y0030	異常	Y0040	正転
21		31	準備完了	41	逆転
22		32	運転中	42	
23		33	運転完了	43	
24	$\times 10^1$	34		44	
25		35		45	
26		36		46	
27		37		47	
28	$\times 10^2$	38		48	
29		39		49	
2A		3A		4A	
2B		3B		4B	
2C	$\times 10^3$	3C		4C	
2D		3D		4D	
2E		3E		4E	
2F		3F		4F	

6.3

プログラム例

このシーケンスのプログラム例を次ページ以降に示します。プログラム作成にあたっては、条件を整理し、できるだけ動作の流れに沿ったプログラムとなるように考慮します。

ここでは、基本的な命令語だけを使用してプログラムを構成していますが、この中で使用している命令語について以下に簡単に説明します。

 入力  出力 : a接点

入力が ON でデバイス④の状態が ON の時、出力を ON にする。

 入力  出力 : b接点

入力が ON でデバイス④の状態が OFF のとき、出力を ON にする。

入力  出力 : 立ち上がり微分接点

前回スキャン時の入力が OFF で今回スキャン時の入力が ON のときだけ、出力を ON にする。

 入力  出力 : コイル

入力が ON のときデバイス④を ON にし、入力が OFF のときデバイス④を OFF にする。

入力  [④ TON Tnnn] 出力 : オンディレイタイマ

入力が OFF から ON に変化してから④で示される時間経過後に出力を ON にする。またこのとき対応するタイマリレーを ON にする。(次ページの例では 0.1 秒タイマ)

カウント入力  出力 : カウンタ
イネーブル入力 

イネーブル入力が ON の状態でカウント入力が ON する回数をカウントして、カウンタレジスタ Cnnn に格納する。④と Cnnn の値が等しくなったら出力を ON にする。イネーブル入力が OFF のときは Cnnn をクリアし、出力を OFF にする。

入力  [④ BIN ④] 出力 : バイナリ変換

入力が ON のとき、④に格納された BCD の値を 2進数に変換して④に格納する。

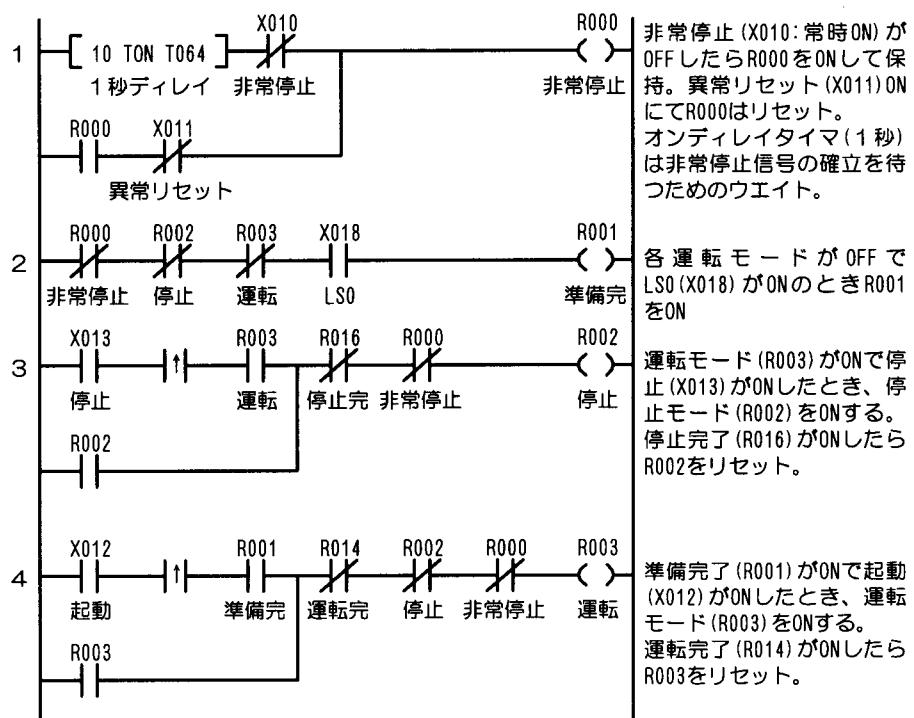
入力  [④ BCD ④] 出力 : BCD変換

入力が ON のとき、④の値を BCD に変換して④に格納する。

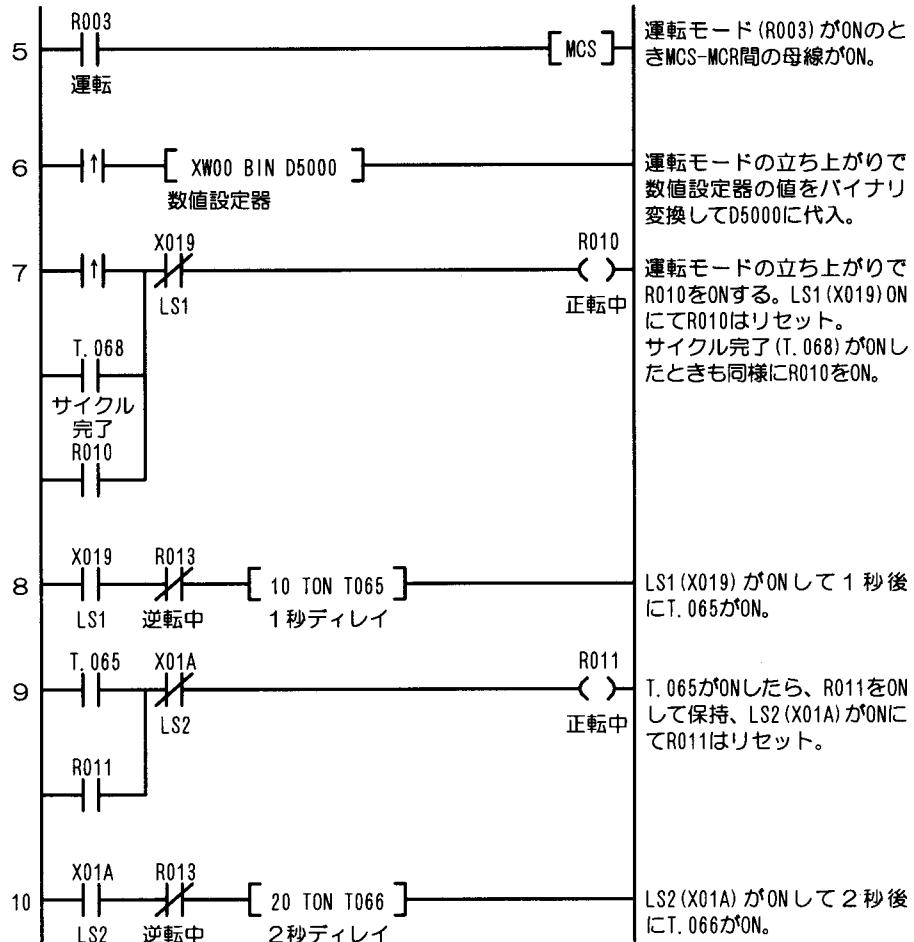
入力  [MCS] / [MCR] : マスターントロールセット/リセット

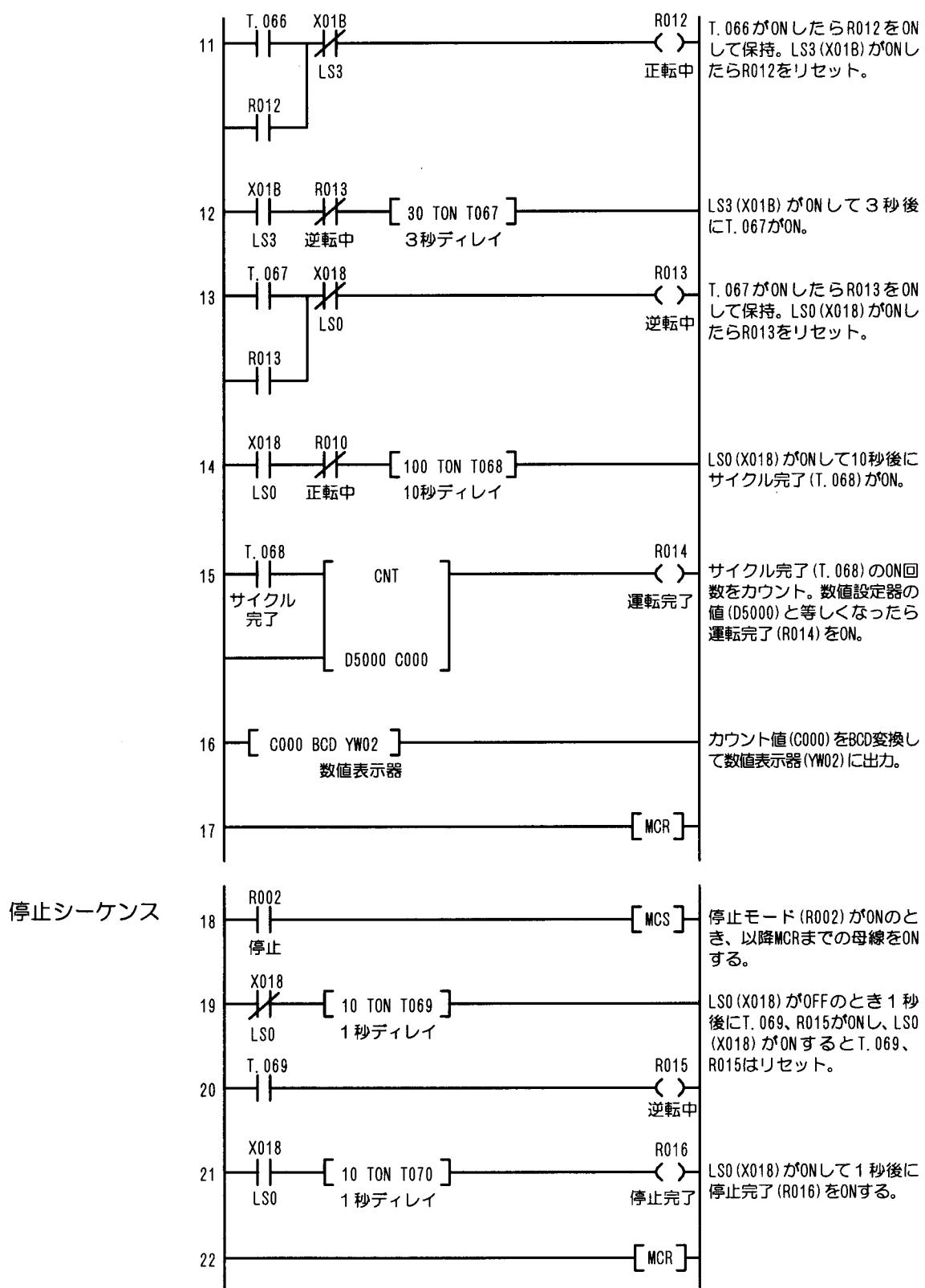
MCS の入力が ON のときだけ、MCS-MCR 間の母線を ON にする。

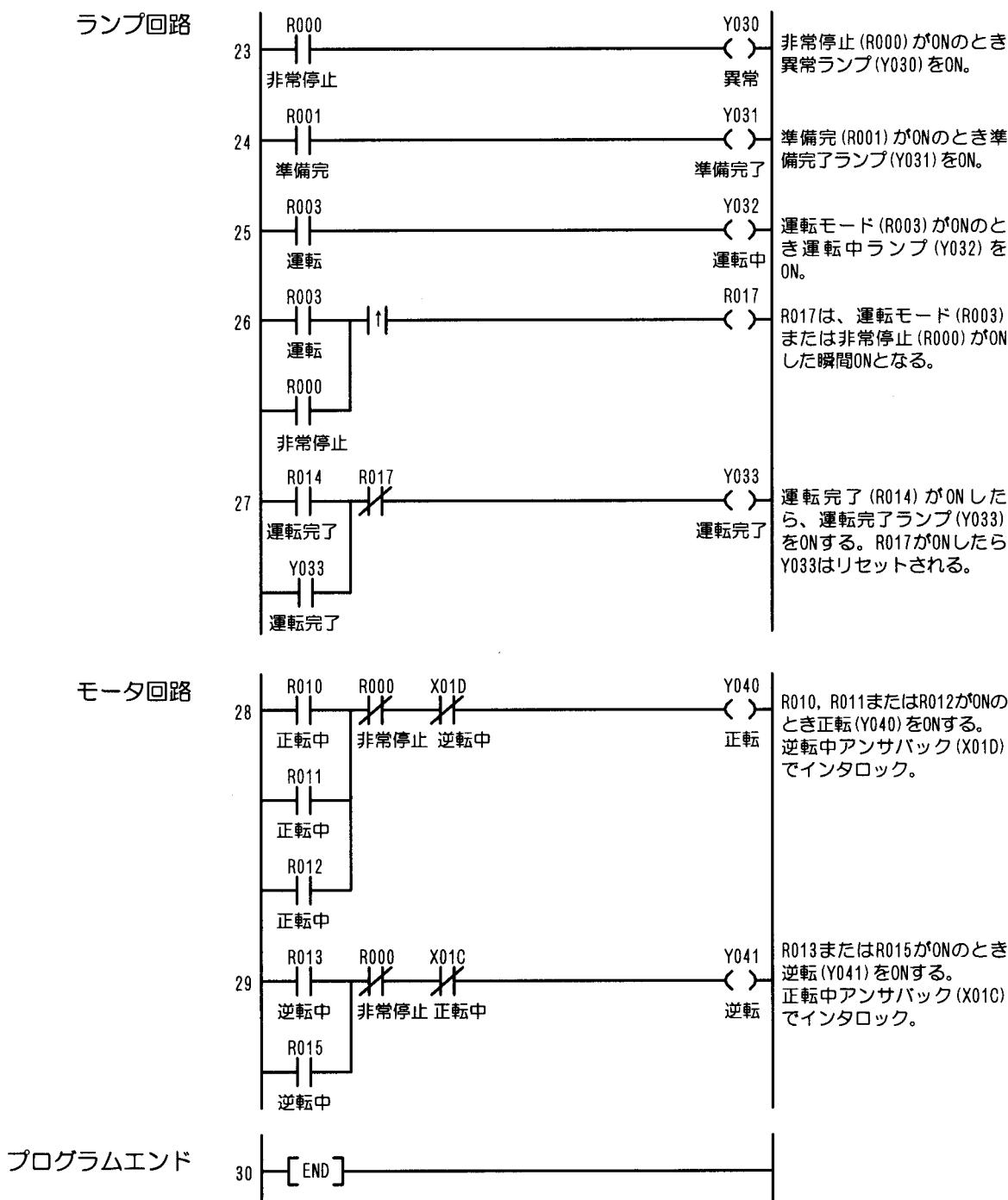
運転モード設定部分



運転シーケンス







⚠ 注意

取扱説明書中に記載のサンプルプログラムは、お客様にて動作確認を行った後使用して下さい。
誤動作による事故を防ぐために運用前に十分確認を行って下さい。

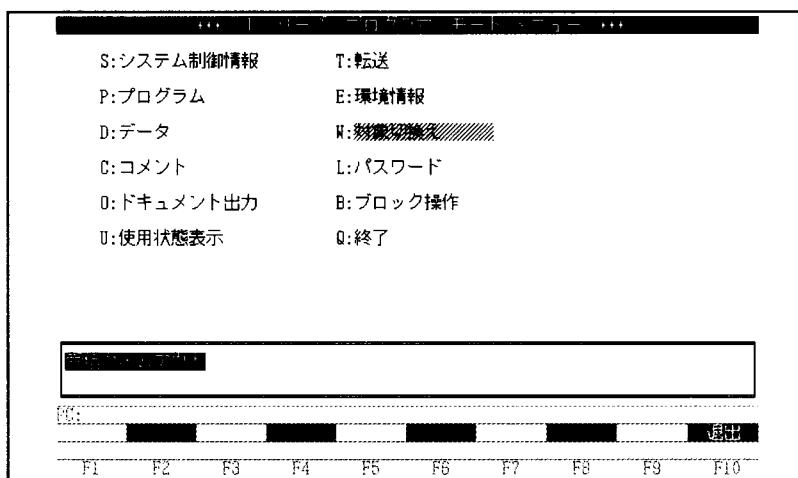
6.4 プログラミング手順

ここでは、プログラマ(T-PDS)を使用して、実際にこのプログラムを書き込む手順を示します。(オフラインプログラミング、T-PDS バージョン 2.0 の操作例)

機種は T3 を、プログラマは T-PDS Ver2.0 を例にとります。

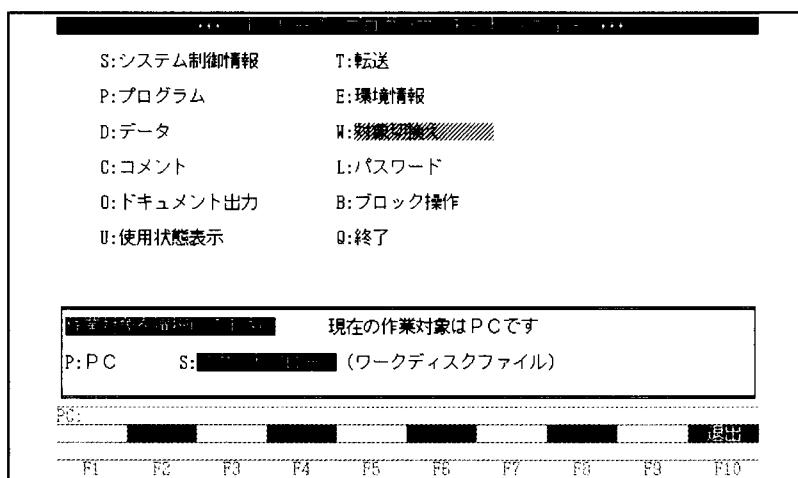
- ① プログラマの電源を ON し、TPDS□とキーインして T-PDS を立ち上げます。T-PDS のタイトル画面が表示され、T-PDS 初期メニュー画面が現れます。

(T-PDS 初期メニュー画面)

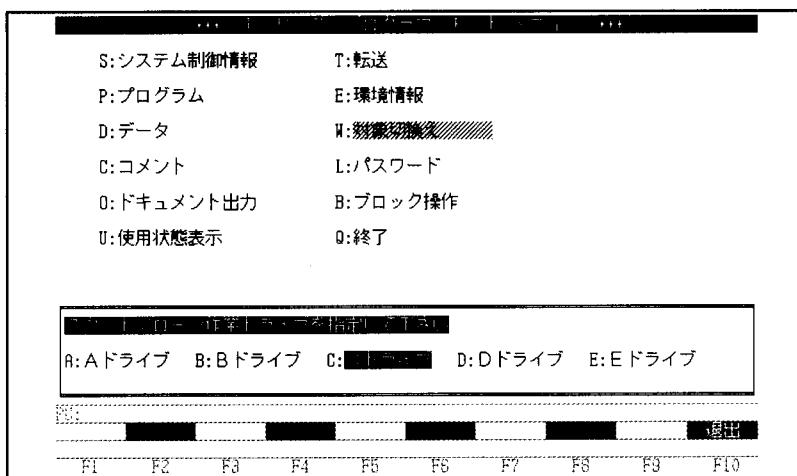


T-PDS の初期状態では PC との交信モードで立ち上がりりますので、PC と未接続の状態では画面上に “受信タイムアウト” と表示されるはずです。

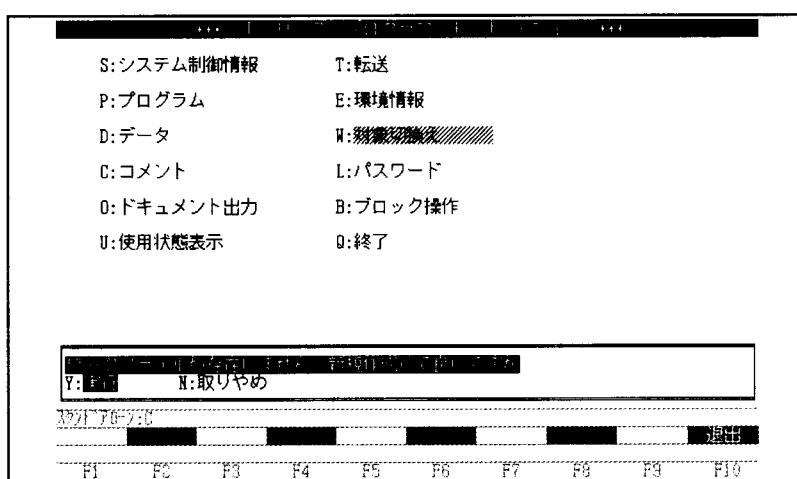
- ② オフラインプログラミングを行うために、作業対象をワークファイルに切り替えます。「W:対象切換え」を選択するために、Wとキーインします。



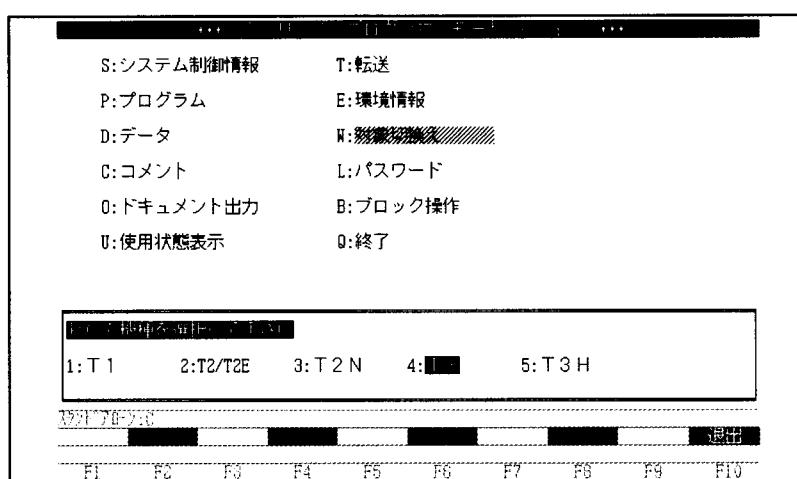
ここで「S:スタンドアローン」を選択します。
Sとキーインします。



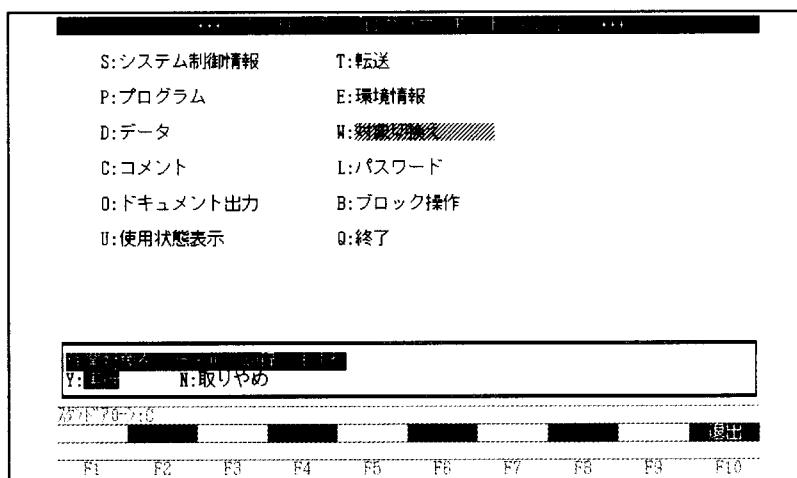
ワークファイルを作成するドライブの選択モードとなります。ここではCとキーインして「C:C ドライブ」を選択します。



ワークファイルの作成確認待ちになりますので、Yとキーインします。



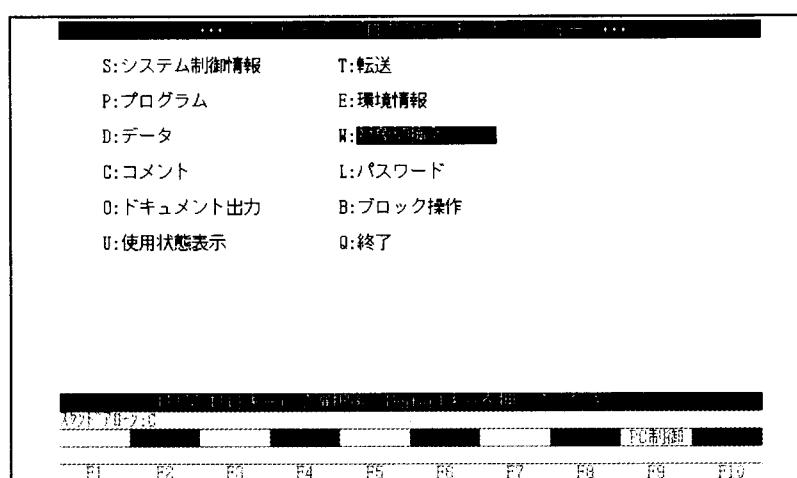
T1、T2/T2E、T2N、T3、T3H の機種選択となります。「4:T3」を選択しますので4とキーインします。



ここまででの操作でスタンドアローンモードへの切り換えは終了し、画面左下には“スタンドアローン C:”と表示されています。

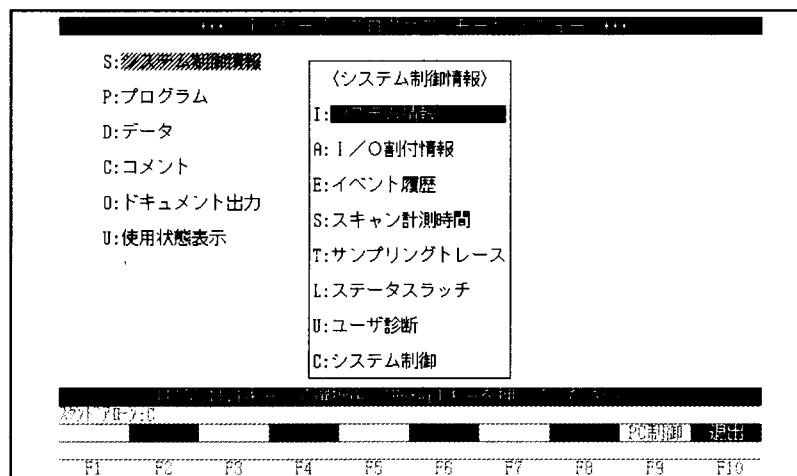
また、ここで設定した作業対象 (C ドライブ) を環境ファイルに記録するか聞いてきますので、「Y: 実行」を選択します。

これにより、次回 T-PDS 立ち上げ時には C ドライブのワークファイルを作業対象として立ち上ります。

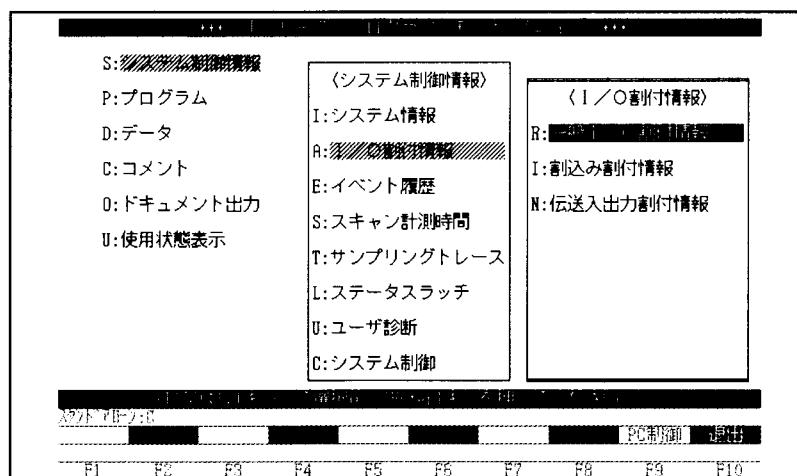


以上でスタンドアローンモードへの切り換えは完了です。

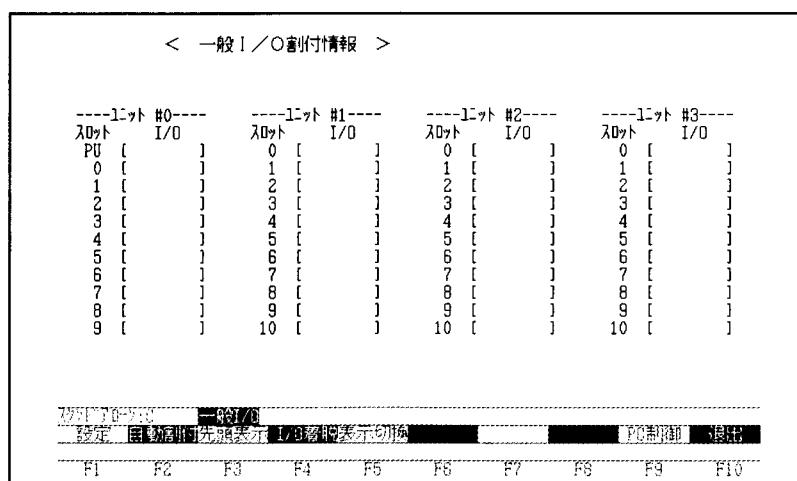
- ③ 次に入出力割り付けを行います。初期メニューの状態から「S:システム制御情報」を選択します。Sとキーインします。



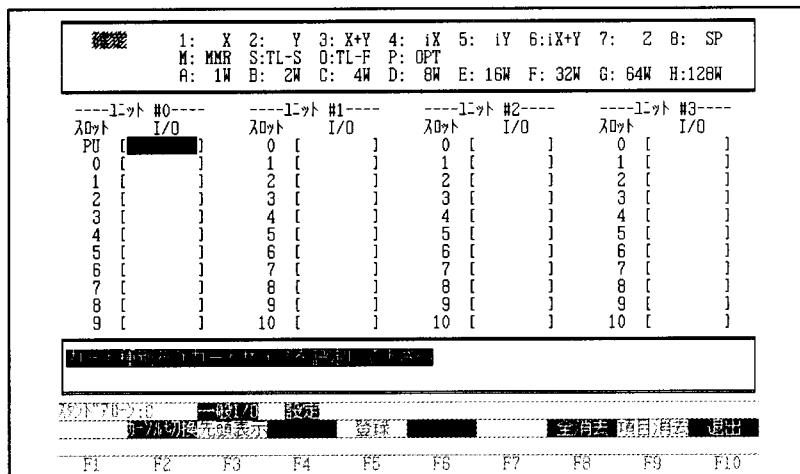
ここで「A:I/O割付情報」を選択します。Aとキーインします。



さらに「R:一般 I/O 割付情報」を選択するためにRとキーインします。



ここでは個別入出力割り付けを行いますので、コマンドラインの F1(設定)を選択します。画面上にはカーソルが現れます。



6.2 節のモジュール構成とするため、入出力割り付けとして次のような設定を行います。

- - -ユニックット #0- - -

スロット I/O

PU []

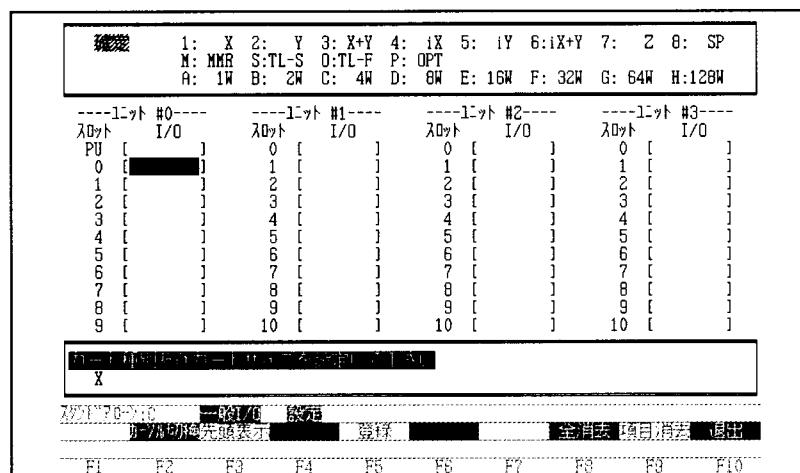
0 [X 2W] ← DI334(32点入力)

1 [Y 2W] ← DO334(32点出力)

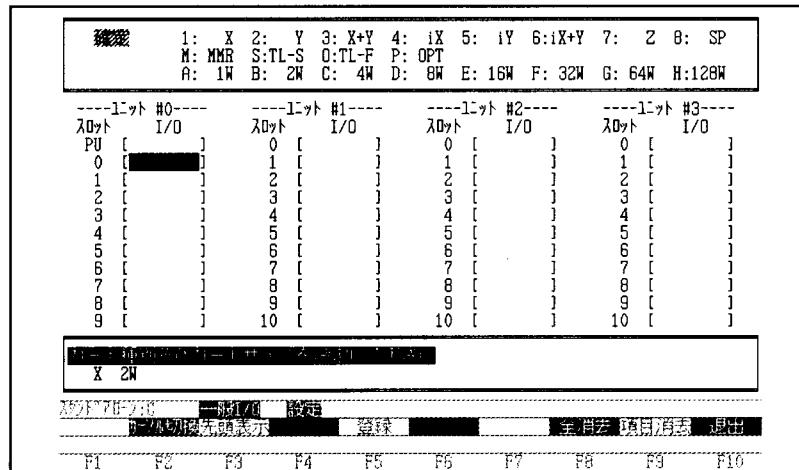
2 [Y 1W] ← RO363S(16点出力)

モジュール種別の設定は、カーソルを所定のスロット位置に移動させた後、画面上部に表示されている選択リストにより機能区分(X, Y 等)と占有レジスタ数(1W, 2W 等)の組み合せで指定します。

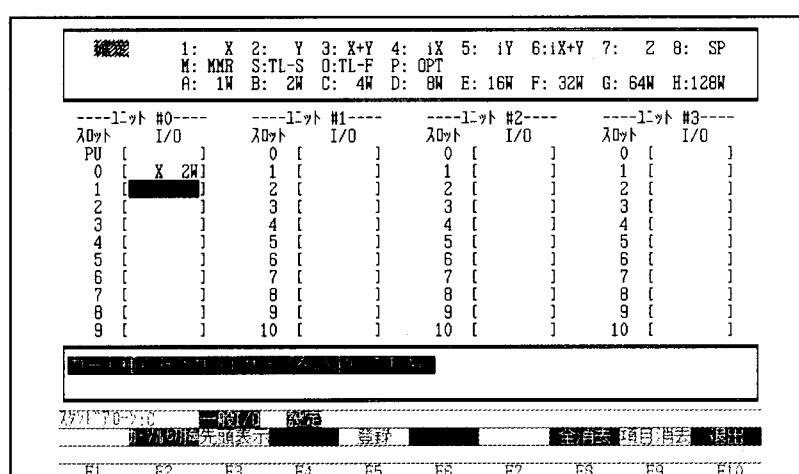
まず、カーソルキー(←↑↓→)によりカーソルをスロット#0 に移動させます。次に「1:X」を指定するために1とキーインします。



続いて「B:2W」を指定しますのでBとキーインします。

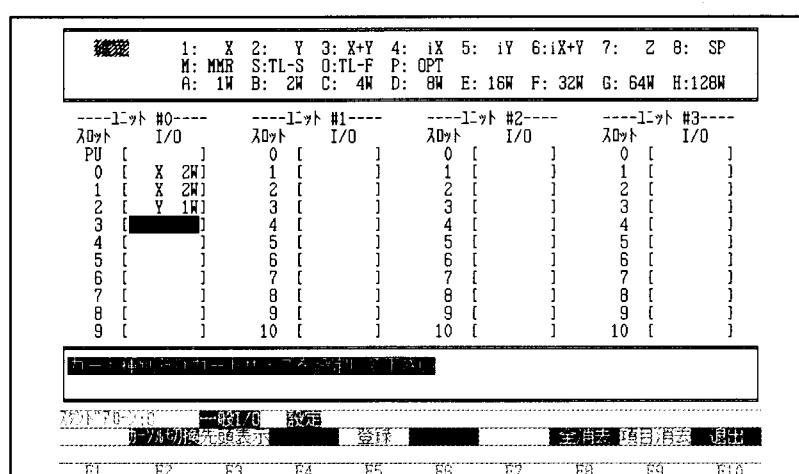


画面下部にモジュール種別として“X 2W”と表示されているはずです。
これをカーソル位置のスロットに設定するために□とキーインします。

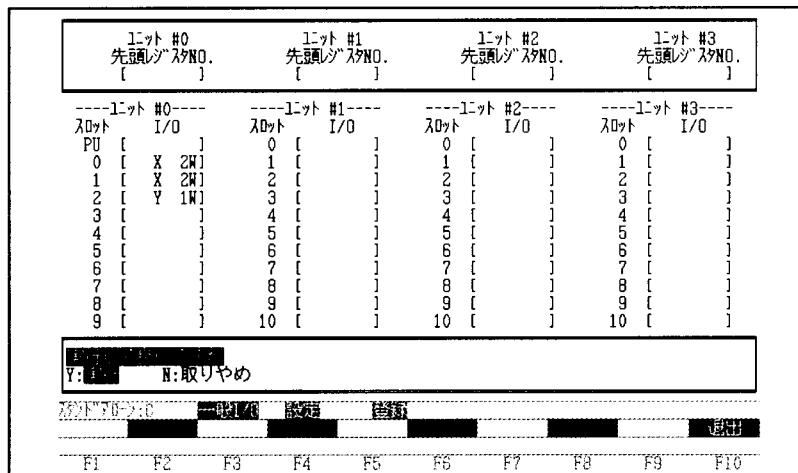


ユニット#0、スロット#0 の位置に “X 2W” と表示され、カーソルは次のスロットに移ります。

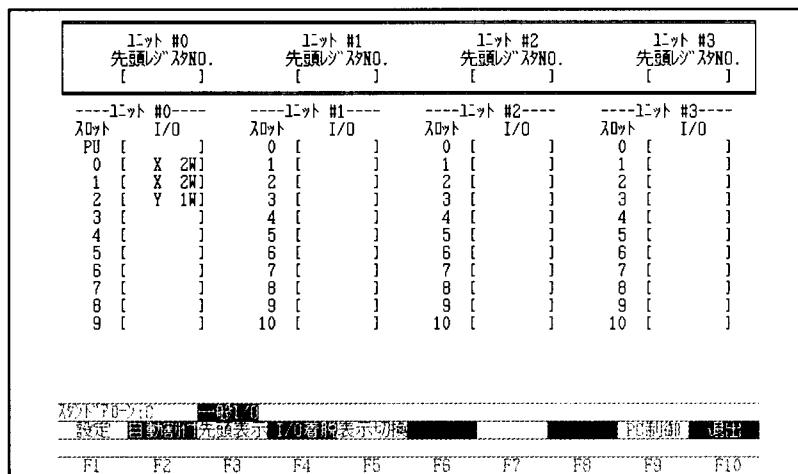
以下同じ要領で必要な設定を行います。



必要な設定が終ったら、この情報をワークファイルに登録します。
コマンドラインから F5 (登録) を選択します。

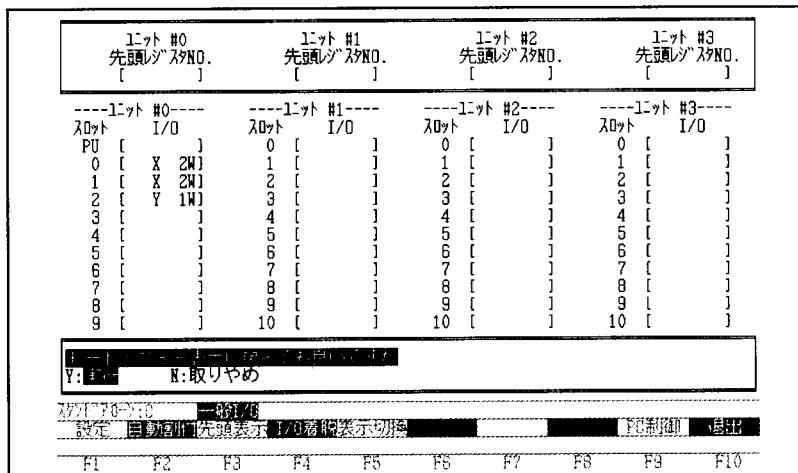


実行確認待ちになりますので、Yとキーインします。

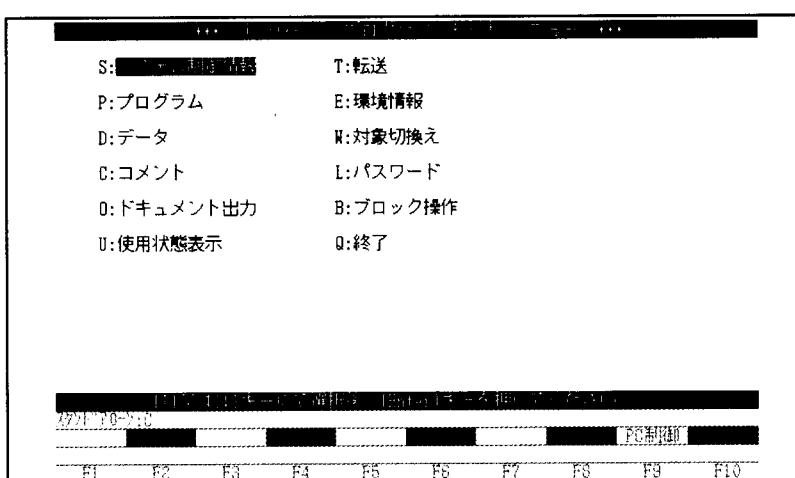


これで入出力割り付けは完了です。

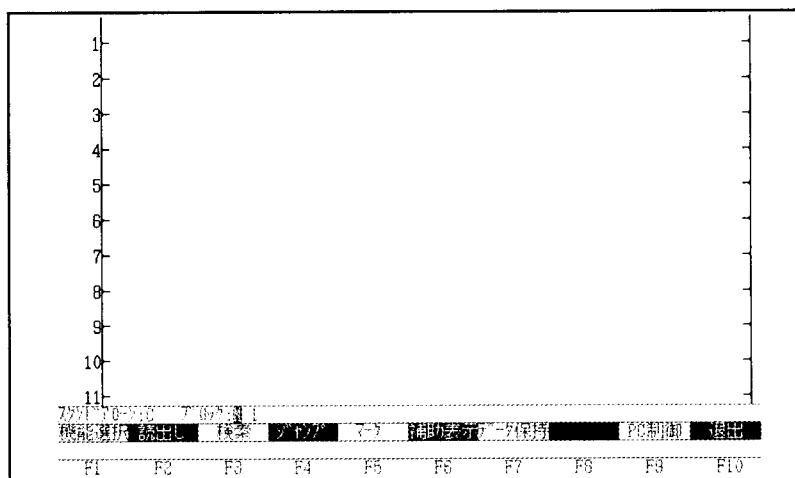
- ④ 今度はプログラムの打ち込みに入ります。まず初期メニューに戻すために Esc キーを押します。



確認待ちになりますので、Yとキーインします。

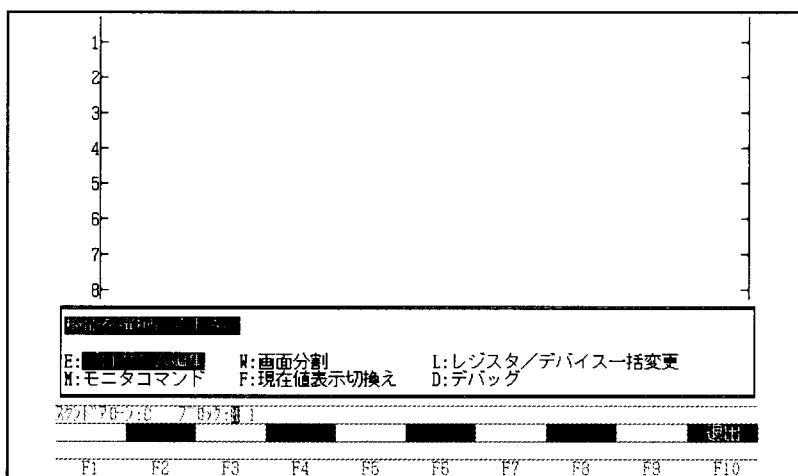


ここで「P:プログラム」を選択します。Pとキーインします。

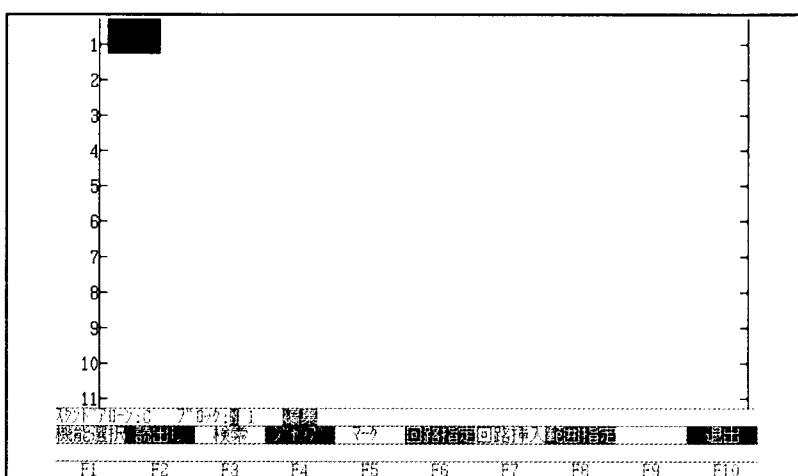


自動的にメインプログラムのブロック1が選択され画面に“ブロック:M1”と表示されているはずです。(もちろん初期状態ですので回路は何も現れません)

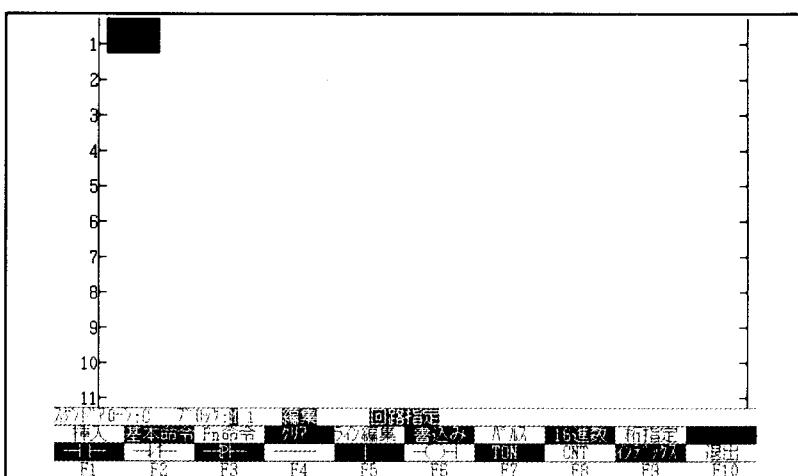
ここでコマンドラインから F1(機能選択)を選択します。



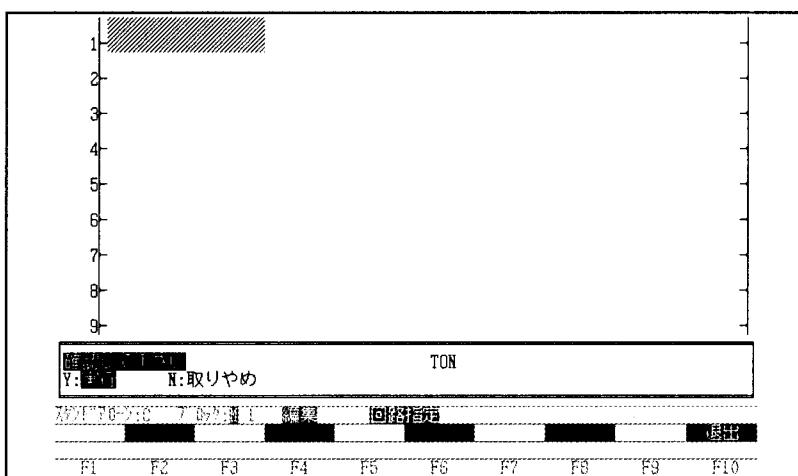
さらにメニューwindowから「E:プログラム編集」を選択します。Eとキーインします。画面上にはカーソルが現れます。



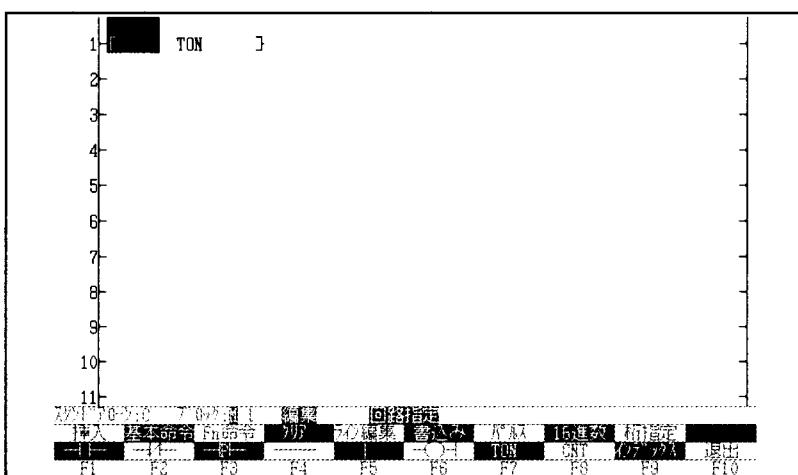
ここでコマンドラインから F6(回路指定)を選択します。すると画面のコマンドラインにはリレーシンボルが現れます。



これから 6.3 節のプログラムを打ち込んでいきます。
まず F7(TON) を押します。



確認待ちになりますので、Yとキーインします。

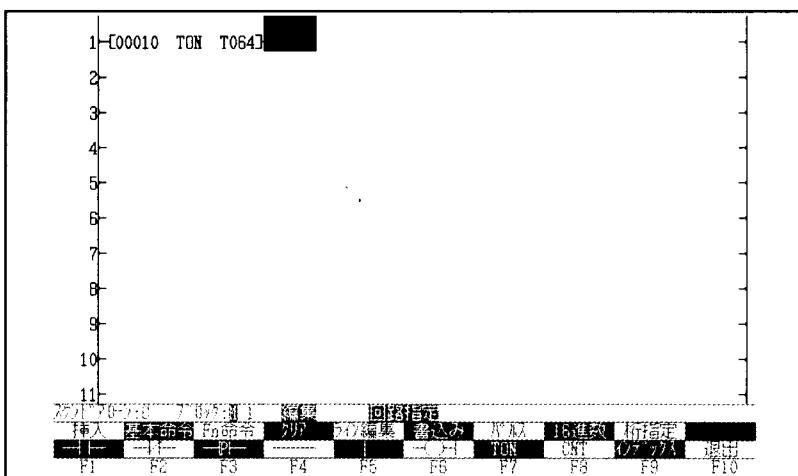


オペランド(設定値とタイマレジスタ)を入力します。

10

T64

とキーインします。間違ったときにはスペースキーで取り消して入力し直します。

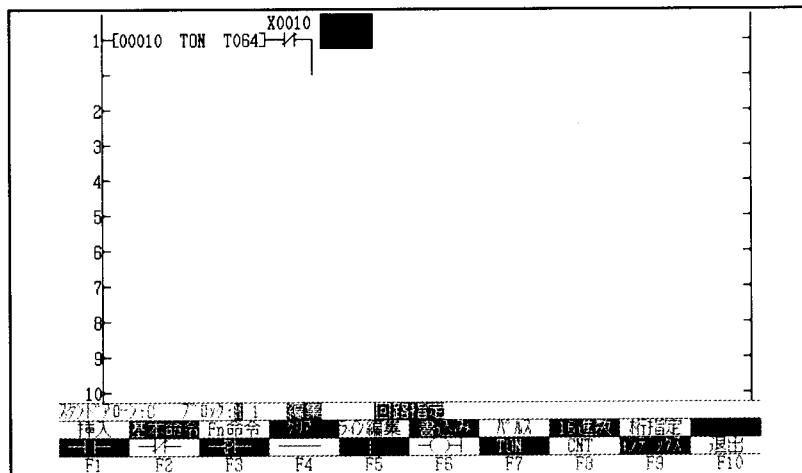


次に X010 の b 接点(縦接続付き)を入力します。

F2(++)

F5(|)

X10[]とキーインします。



以下同様の手順で第1回路を完成させます。

F6(<)(-)R0[]

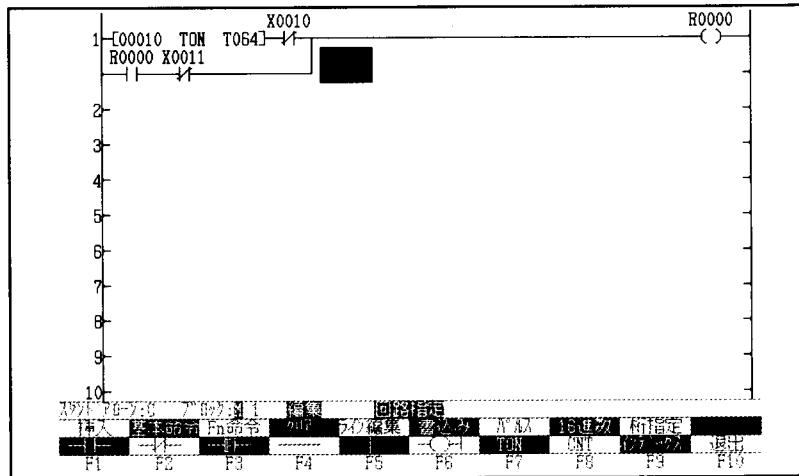
F1(+)(+)R0[]

F2(++)X11[]

F4(-)[]

F4(-)[]

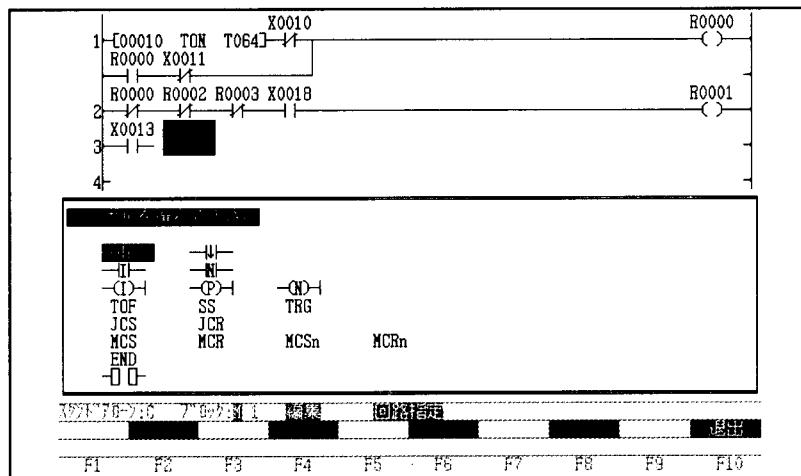
とキーインします。



次にカーソルキー(←↑↓→)によって第2回路の先頭に移動させて、第1回路と同じ要領で第2回路、さらに第3回路、第4回路と入力していきます。

途中、立ち上がり微分接点のようにコマンドラインにシンボルが表示されていない命令語を選択する場合には、Shift+F2(基本命令)と押してメニューウィンドウを表示させて選択します。

(Shift+F2(基本命令)を押した画面状態)

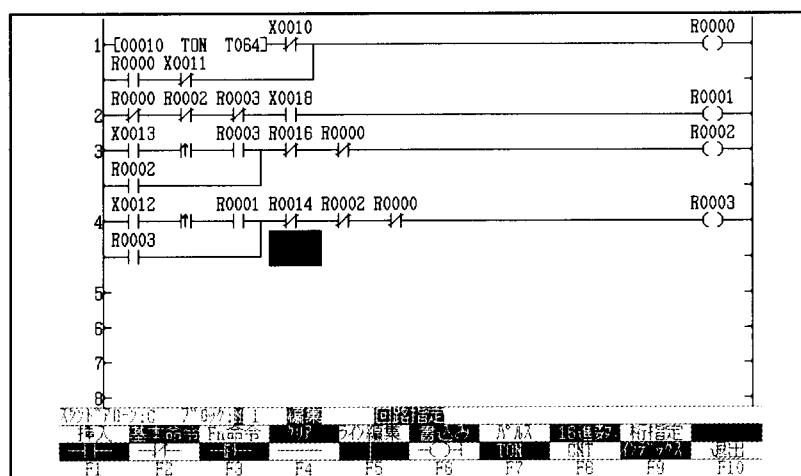


第4回路まで入力し終えたところで、第1～第4回路をワークファイルに書き込みます。

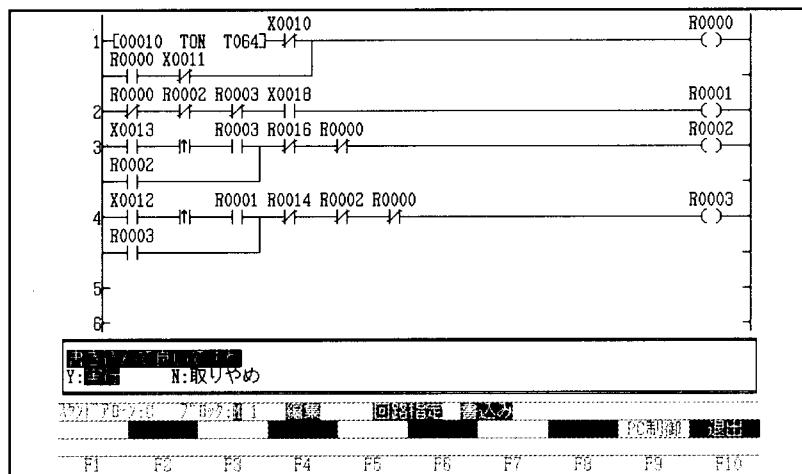
つまり、1度にワークファイルに書き込めるのは1画面分の大きさ(11行×12列)ですので、回路の区切りのいいところでワークファイルに書き込んでやるわけです。(PC本体を対象としたときも同じ)
(回路編集画面ではカーソルも画面範囲内でしか動きません)

ワークファイルへの書き込み操作は次のように行います。

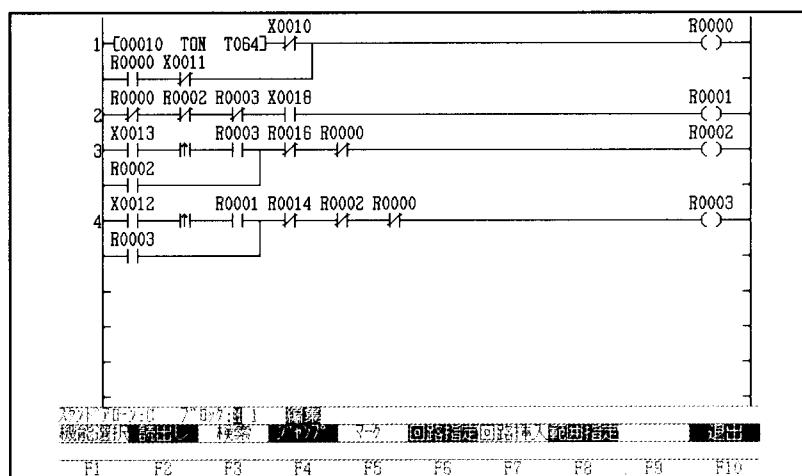
(第4回路まで入力が終わった状態)



Shift+F6(書き込み)とキーインします。

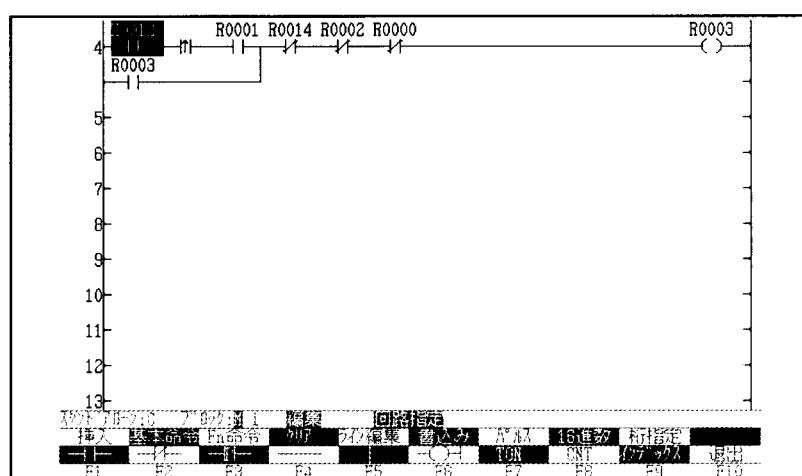


実行確認待ちとなりますので、Yとキーインします。



これで第1回路から第4回路までがワークファイルに書き込まれました。次に第5回路以降の入力に移ります。カーソルキーでカーソルを第4回路上に移動させ(カーソルは存在する回路上しか動きません)。

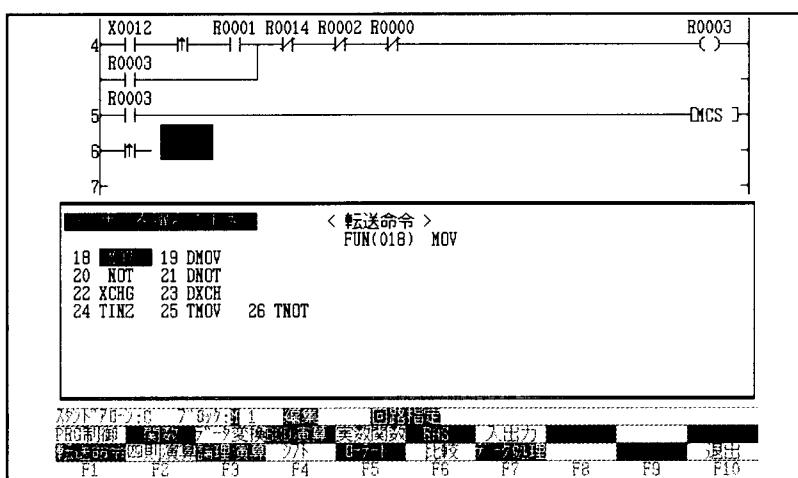
F6(回路指定)を押します。すると画面は第4回路を先頭とする回路編集画面となります。



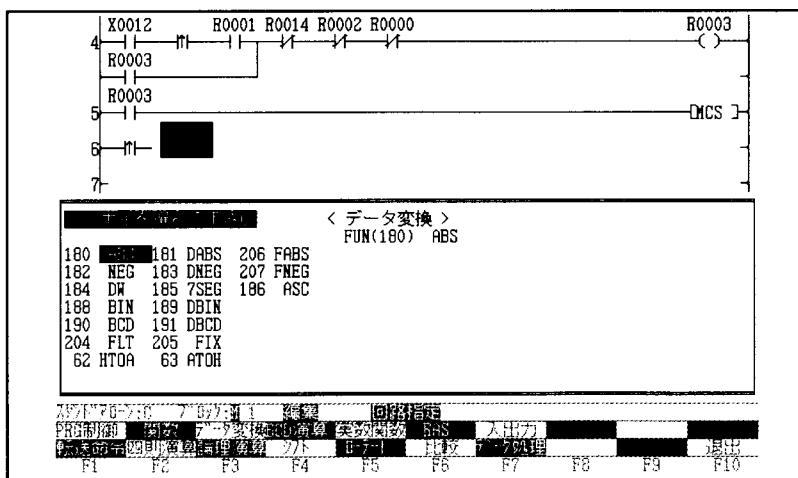
カーソルを第5回路の先頭に移動させて、以降のプログラムを作成していきます。

BIN 命令などのファンクション命令を入力する場合には、Shift+F3(Fn 命令)と押してファンクション命令の選択状態とし、さらに命令語グループを選択してから所定の命令語を選択します。

(Shift+F3(Fn 命令)を押した画面状態)



(上図の状態から Shift+F3(データ変換)を押した状態)



以下第4回路までと同様の手順で、適宜ワークファイルに書き込みながら、全ての回路を書き込んでいきます。

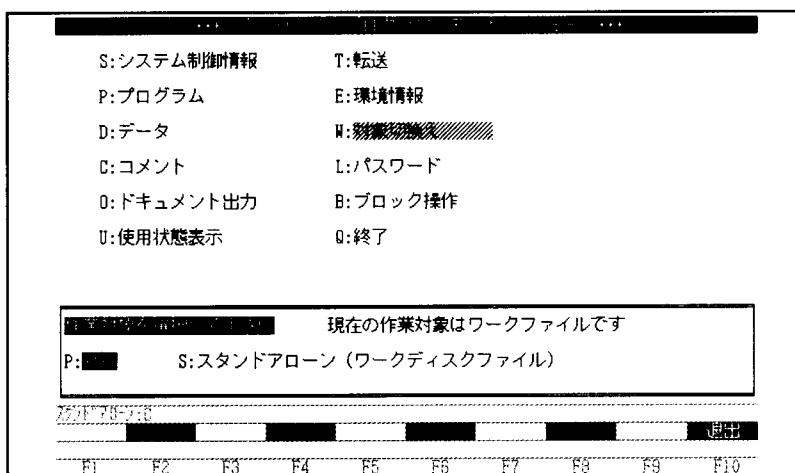
- ⑤ ここまで操作でワークファイルにプログラムが全て書き込まれたら、次に T3 本体にプログラムをローディングします。
 まず T3 と T-PDS を専用ケーブルで接続します。(T3 には 6.2 節のモジュールが実装されているものとして説明します)
 次に T3 の RAM/ROM 切換スイッチを RAM 側に、モードスイッチを RUN の位置にして T3 に電源を投入します。(T3 は HALT モードで立ち上がり RUN 待機状態になります)

補足

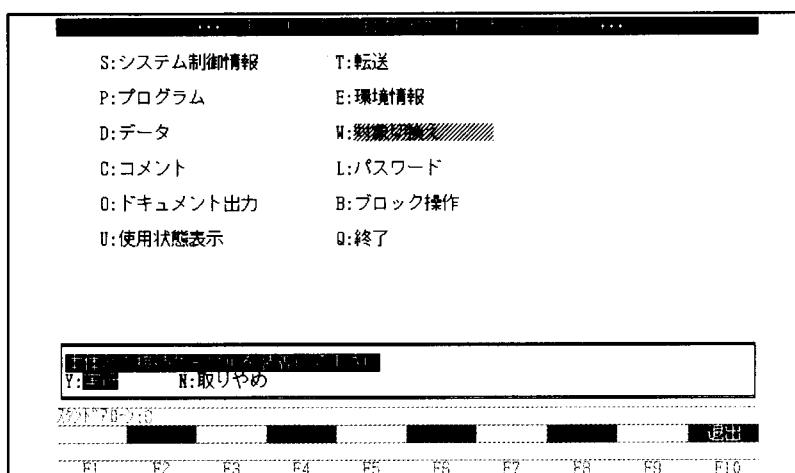
T3H の場合、立ち上げ時のモードはプログラマで設定するスタートモードによって決められます。この例題を T3H で行う場合はスタートモードをスタンバイに設定してください。

- ⑥ T-PDS 側を PC(T3) との交信モードにします。

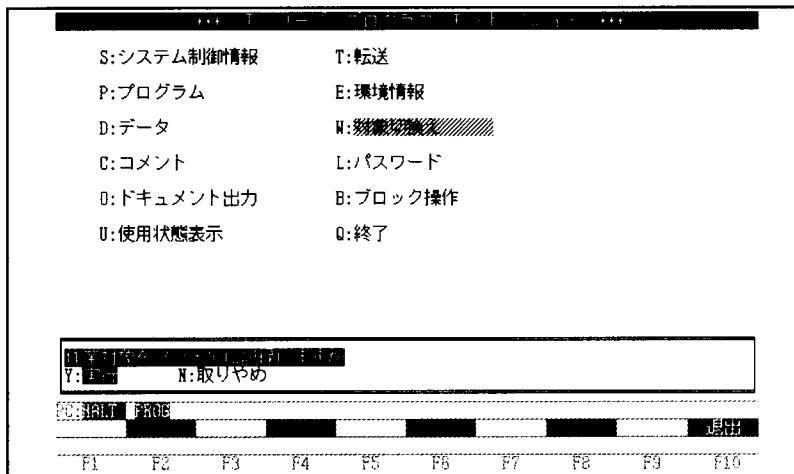
まず Esc [] と押して T-PDS の初期メニューを表示させてから、「W:対象切換え」を選択します。



ここで作業対象として「P:PC」を選択します。P とキーインします。



接続状態を確認して、Yとキーインします。

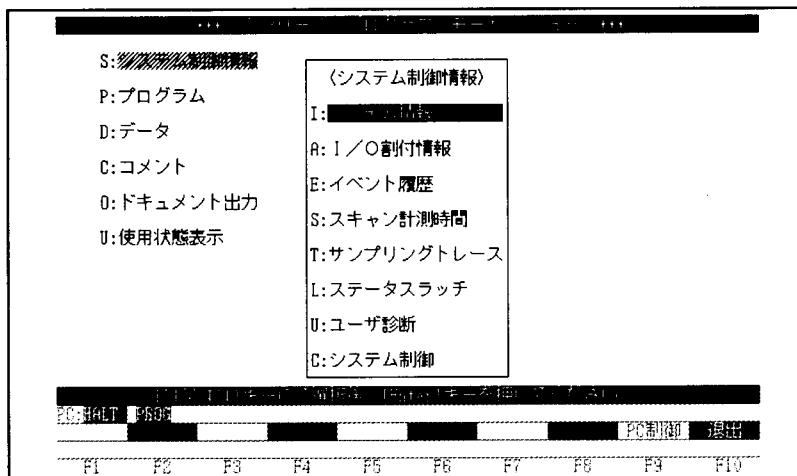


T3 と T-PDS の通信回線が正常に接続されると、画面左下には“PC:HALT”と表示されているはずです。

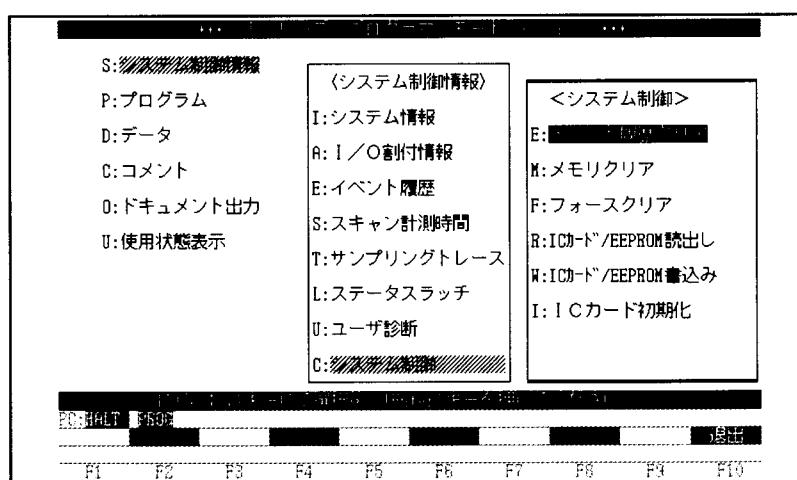
作業対象の記録は必ずしも必要ではありませんが、とりあえずここでは「Y:実行」を選択しておきます。

これで T-PDS はオンラインモードになりました。

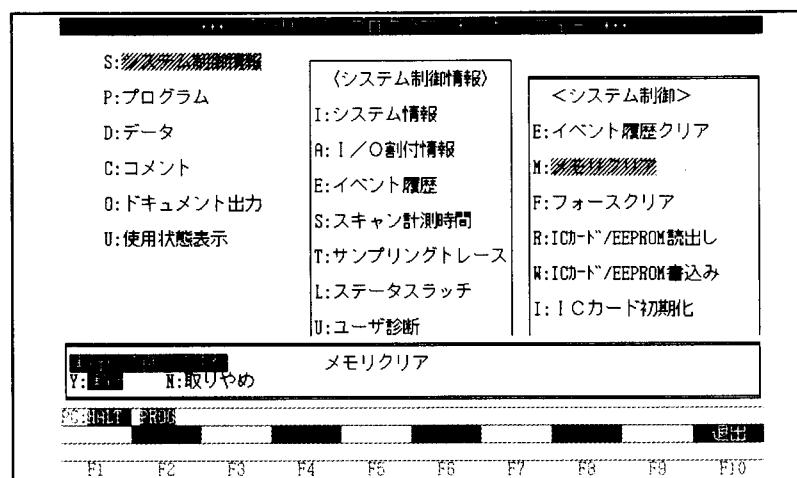
- ⑦ 次に T3 本体のメモリをクリアします。「S:システム制御情報」を選択します。Sとキーインします。



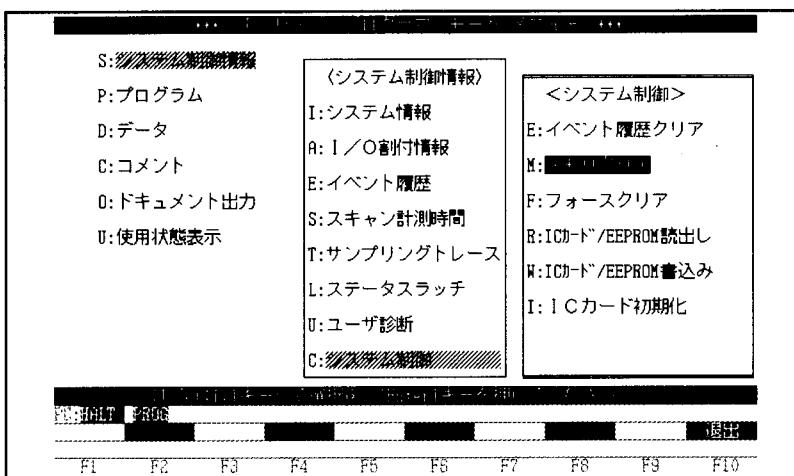
ここで<システム制御情報>のメニューの中から「C:システム制御」を選択します。Cとキーインします。



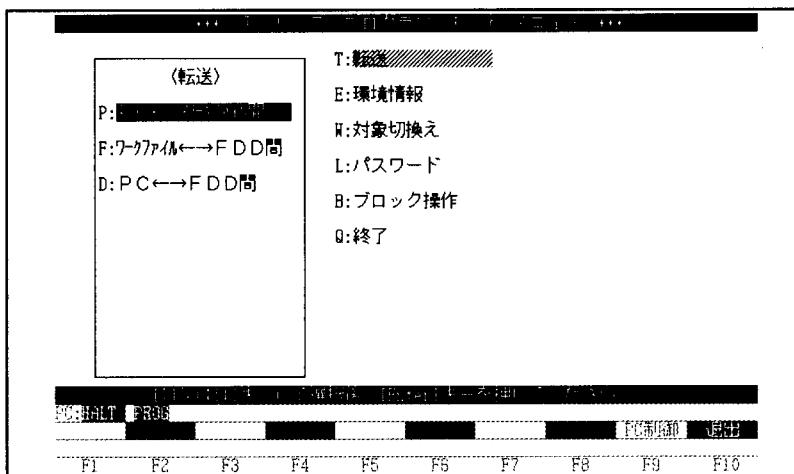
さらに<システム制御>メニューの中から「M:メモリクリア」を選択します。Mとキーインします。



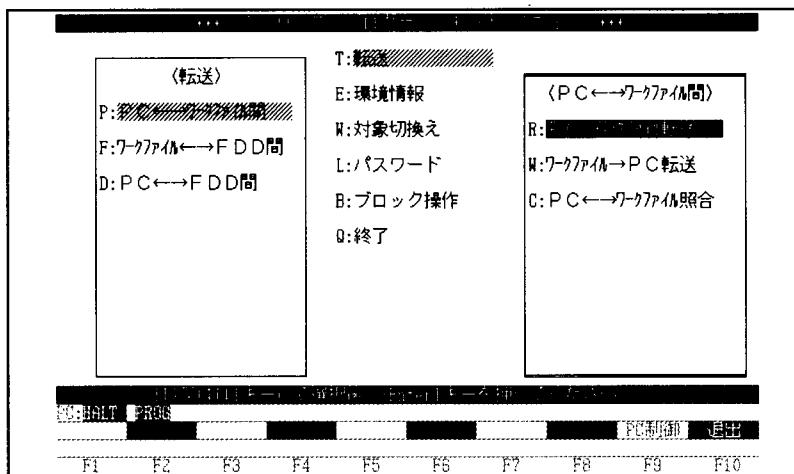
実行確認待ちになりますので、Yとキーインしてメモリクリアを実行します。



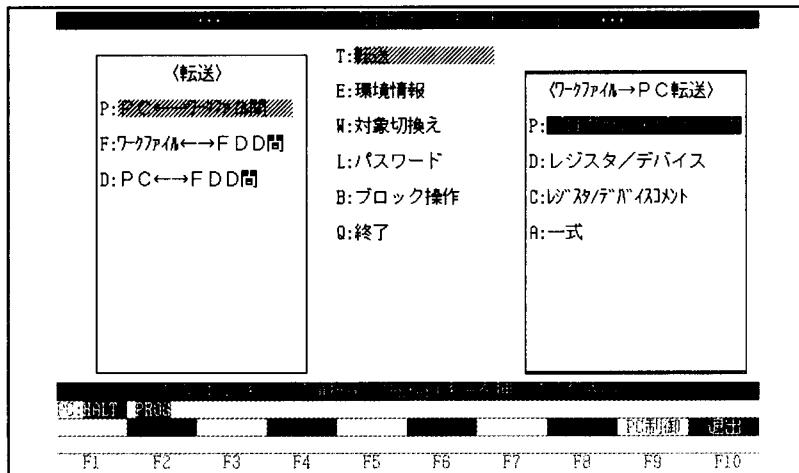
- ⑧ 次にワークファイルに作成したプログラムを T3 本体に転送(ローディング)します。まず Esc□と押して初期メニューを表示させてから「T: 転送」を選択します。



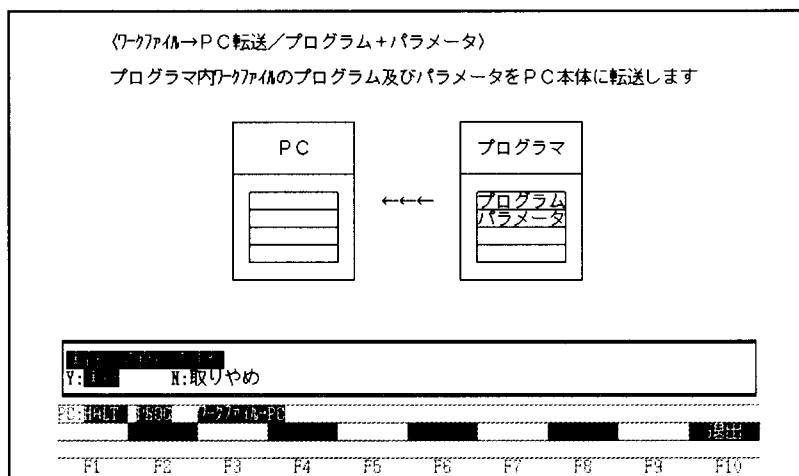
ここで<転送>メニューの中から「P:PC↔ワークファイル間」を選択します。



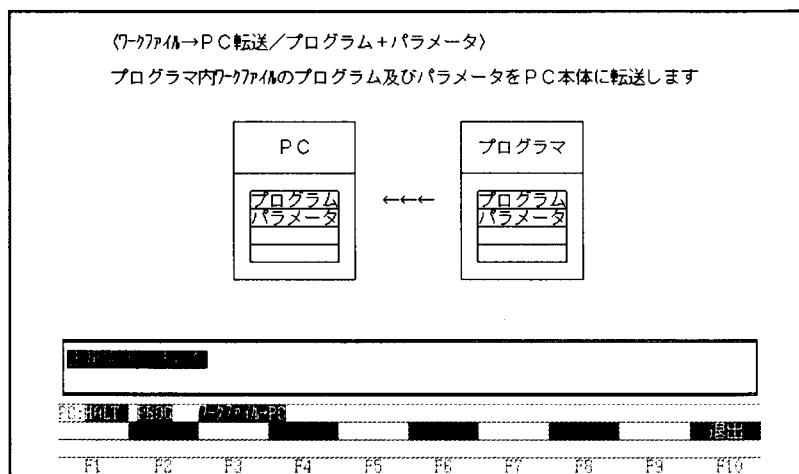
さらに<PC↔ワークファイル間>メニューの中から「W:ワークファイル→PC転送」を選択します。



転送内容の選択メニューが表示されますが、この例では単純にプログラムのみですので「P:プログラム+パラメータ」を選択します。



実行確認待ちになりますので、Yとキーインします。



正常に転送が行われると“正常完了しました”と表示されます。

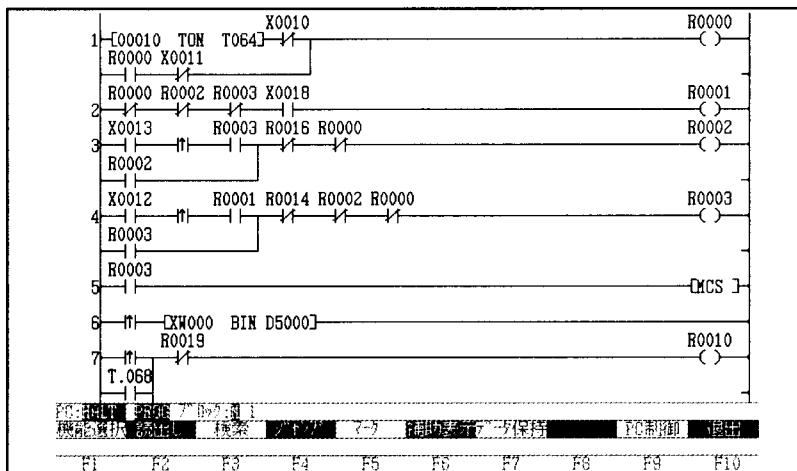
- ⑨ 以上の操作でプログラムのローディングが完了したら、T3 を運転させ(RUN モードにして)プログラムのデバッグを行います。

⚠ 注意

運転中のプログラム変更、強制出力、RUN(運転)、HALT(停止)などの操作は十分安全を確認して行って下さい。
操作ミスや安全確認の怠りにより、機械の破損や事故が起こる恐れがあります。

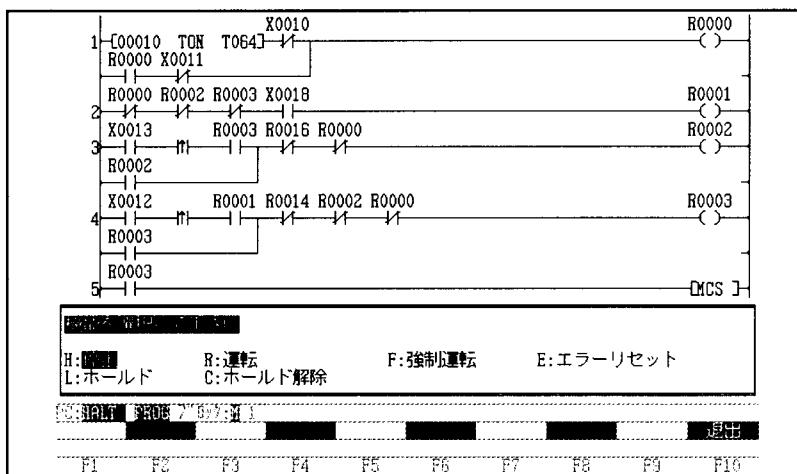
ここでは T-PDS の運転コマンドによって T3 を運転させてみます。

まず ESC□と押して初期メニューを表示させてから、「P:プログラム」を選択します。

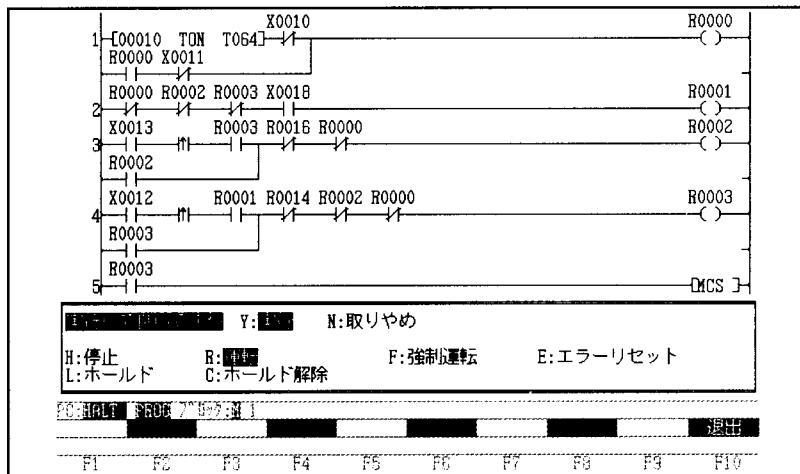


T3 本体に記憶されたプログラムのモニタ状態となります。

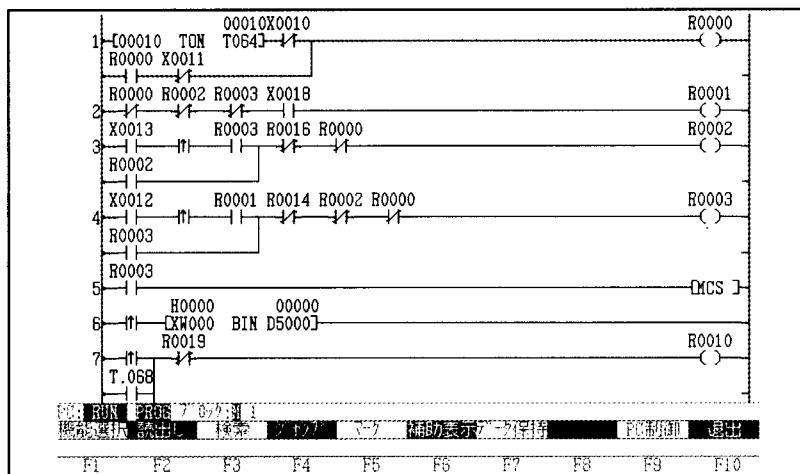
ここでコマンドラインから F9(PC 制御)を選択します。



メニューウィンドウから「R:運転」を選択します。



実行確認待ちとなりますので、再度周囲の安全を確認した上でYとキーインします。



画面には“PC:RUN”と表示され、プログラムの実行状態のモニタ画面となります。

外部のシミュレーションスイッチや、T-PDS の模擬入力機能(フォース状態)を使用して動作の確認を行って下さい。(操作については別冊のプログラマの説明書をご覧下さい)

プログラムの修正/変更を行う場合には、一旦 T3 を停止させ(HALT モードにして)、T3 本体内のプログラムを修正/変更して下さい。

オンラインモードの状態でプログラムの作成/変更を行う場合も、操作としてはワークファイルを対象とする場合と同じです。

補足

プログラムデバッグ、試運転の目的で PC を RUN モードにする場合には、動力回路の電源を OFF するなど、安全には十分注意して下さい。

- ⑩ プログラムの修正、動作確認が終了したら、T3 本体のプログラムを FD 等に保存し、T3 本体の電源を落とします。

T-PDS を終了するには、Esc□と押して初期メニューを表示させた状態で「Q:終了」を選択します。これで T-PDS を立ち上げる前の状態に戻ります。

以上でプログラミングの操作は終了です。次回電源投入時から PC を自動運転させるには、次のような設定をしておきます。

T3 の場合 : RAM/ROM 切換スイッチ =ROM 側

モードスイッチ =RUN または P-RUN

T3H の場合 : プログラマによるスタートモードの設定 =オート

モードスイッチ =RUN

または P-RUN

補足

EEPROM 内蔵の CPU (PU325/325H/326H) の場合には、上記手順⑩の前にプログラムを EEPROM に書き込んでおいて下さい。

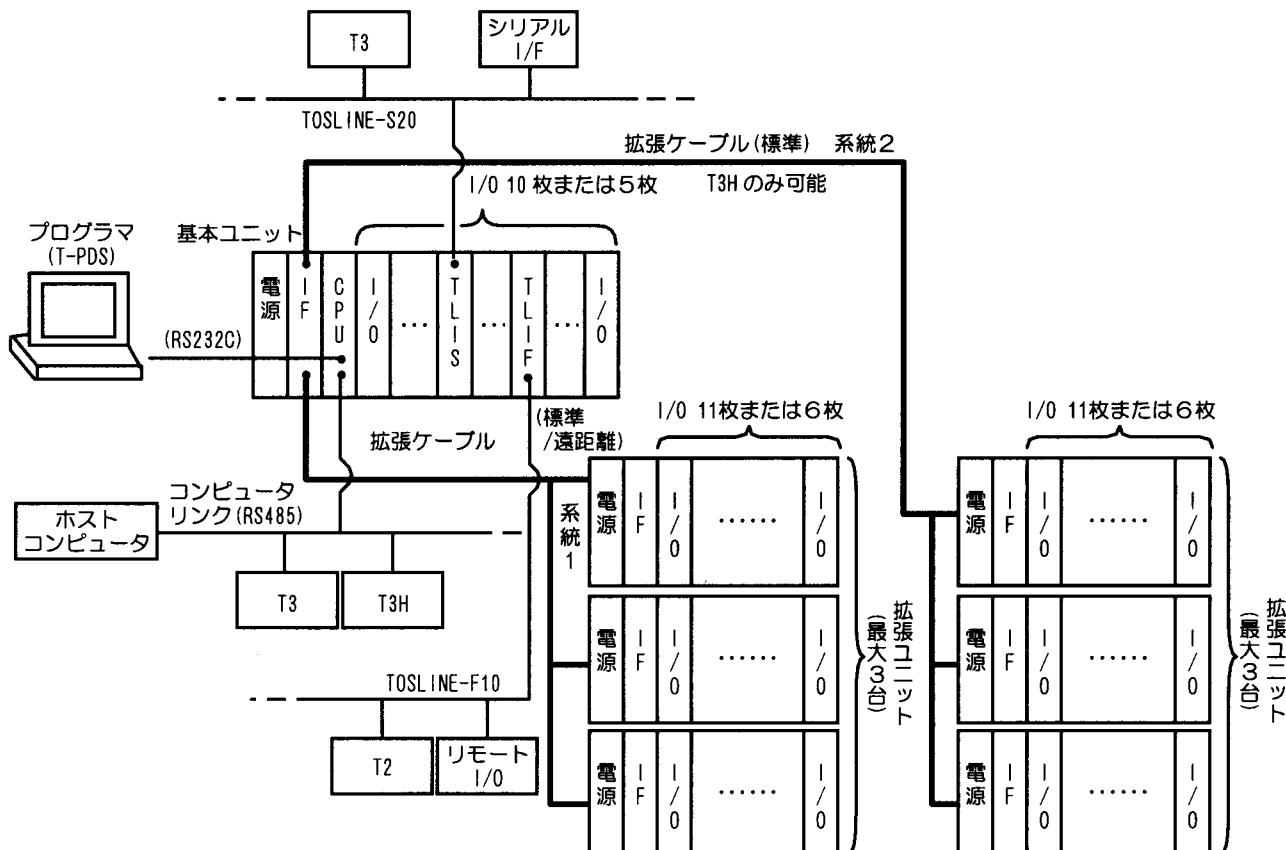
操作は、<システム制御>メニューの中から「W:IC カード/EEPROM 書き込み」を選択実行することにより行われます。（手順⑦の画面側参照）

第2部

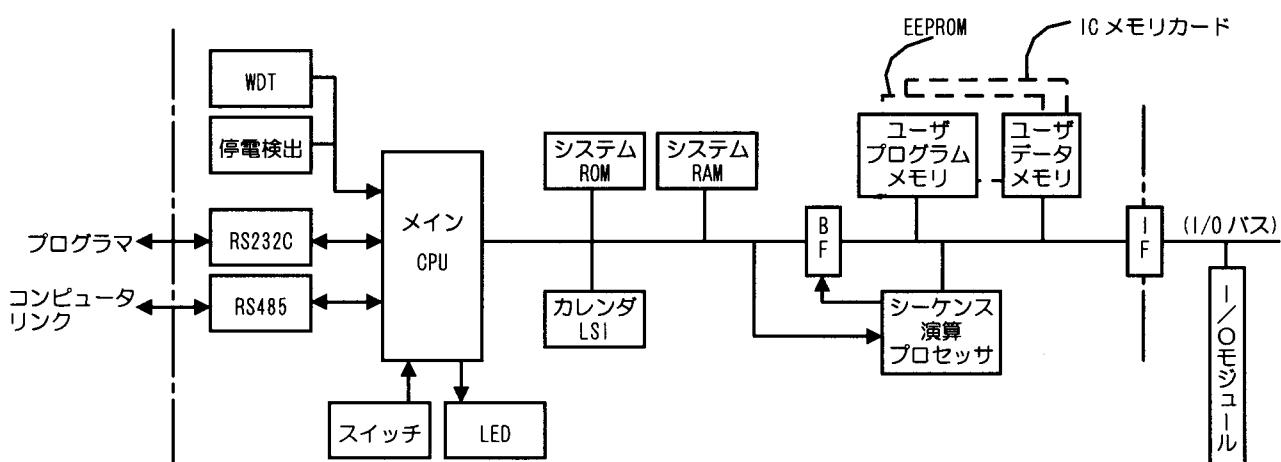
本体機能説明

1.1 システム構成

下図にシステム構成を示します。第2部では、PC システムが実現できる機能のうち、CPU の機能を主体に説明します。



PC の CPU 部のブロック図を下図に示します。



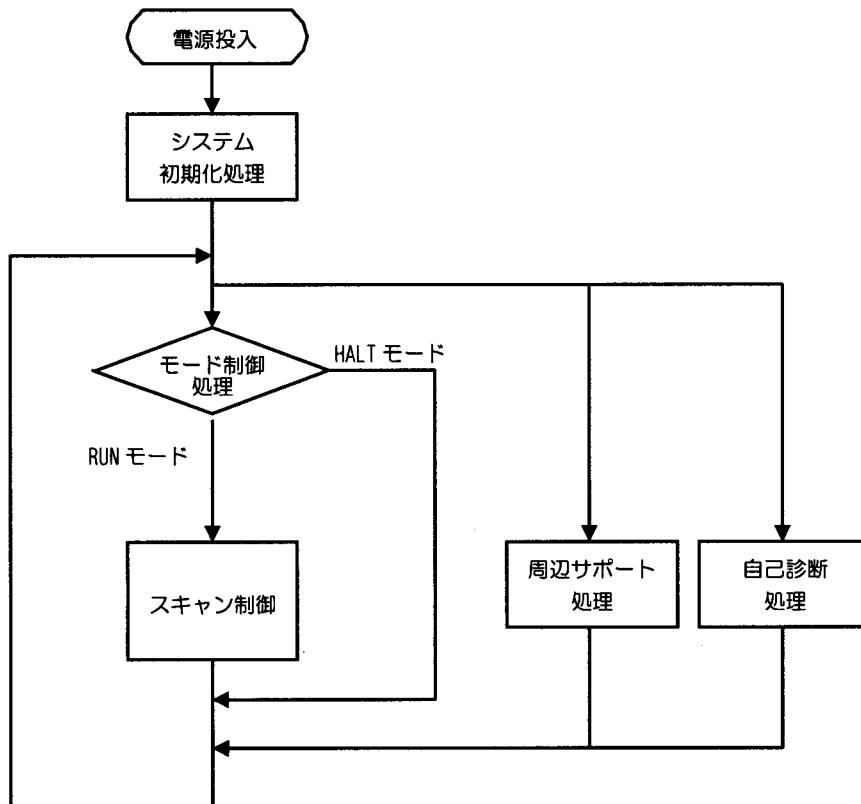
PC の CPU 部は全体の動作を管理するメイン CPU と、シーケンス演算(ビット演算、ワード演算)を実行するシーケンス演算プロセッサ(LP)の2つのプロセッサにより構成され、スキャン実行中は並列動作を行います。

1.2 機能仕様一覧

項目	仕 様		
	T3	T3H	
制御方式	ストアードプログラム・サイクリックスキャン方式	←	
プロセッサ	実行管理:16ビットマイクロプロセッサ シーケンス演算:専用演算プロセッサ(LP)	←	
入出力方式	一括入出力(リフレッシュ)方式/直接入出力 (ダイレクト)方式併用	←	
入出力点数	1376 点(32 点 I/O 使用時) 2752 点(64 点 I/O 使用時) ローカル I/O 最大 4096 点/256 ワード (1 ワード=16 ビット)	2432 点(32 点 I/O 使用時) 4864 点(64 点 I/O 使用時) ローカル I/O 最大 8192 点/512 ワード (1 ワード=16 ビット)	
ユ ニ カ ブ ロ グ ラ ム	プログラム言語	ラダー図(リレーションボル+ファンクションブロック) SFC(シーケンシャルファンクションチャート)	←
	プログラム容量	32K ステップ(プログラムコメント含む) (1ステップ=24 ビット)	32K ステップ(PU325H)/64K ステップ(PU326H)
	メモリ	メインメモリ:SRAM(バッテリバックアップ) 周辺メモリ:EEPROM/I/O メモリカード	←
	命令語	基本命令 24 種、応用命令 201 種 • 転送(単長/倍長/レジスタテーブル) • 四則演算(単長/倍長、バイナリ/BCD、符号有/無、浮動小数点可) • 論理演算(単長/倍長/レジスタテーブル/ビットファイル) • 比較(単長/倍長、符号有/無、浮動小数点可) • 実行制御(ジャンプ/FOR-NEXT/サブルーチン/SFC 他) • 関数(リミット/三角関数/積分/PID/関数発生、浮動小数点可) • 変換(ASCII, BCD, アセグメント他) その他	基本命令 24 種、応用命令 206 種
	演算速度	0.15μs/接点、0.3μs/コイル 0.9μs/転送、2.25μs/加算	0.1μs/接点、0.2μs/コイル 0.54μs/転送、0.9μs/加算
	スキャン方式	フローティングスキャン/定刻スキャン (定刻:10~200ms, 10ms 単位)	←
	マルチタスク機能	メインプログラム 1 本、サブプログラム 4 本 定周期割り込み 1 本(2~1000ms, 1ms 単位) I/O 割り込み 8 本(割り込み応答 500μs 以下)	メインプログラム 1 本、サブプログラム 4 本 定周期割り込み 1 本(1~1000ms, 1ms 単位) I/O 割り込み 8 本(割り込み応答 500μs 以下)
ユ ニ カ ブ ロ グ ラ ム タ ー	入出力リレー/ レジスタ	4096 点/256 ワード (X/Y, XW/YW は一括入出力) (I/O, IW/OW は直接入出力) (1 ワード=16 ビット、以下同様)	8192 点/512 ワード (X/Y, XW/YW は一括入出力) (I/O, IW/OW は直接入出力) (1 ワード=16 ビット、以下同様)
	補助リレー/ レジスタ	8192 点/512 ワード	16000 点/1000 ワード
	特殊リレー/ レジスタ	4096 点/256 ワード	←
	タイマリレー/ レジスタ	512 点 (T000~T063 の 64 点は 0.01 秒タイマ) (T064~T511 の 448 点は 0.1 秒タイマ)	1000 点 (T000~T063 の 64 点は 0.01 秒タイマ(初期値)) (0.01 秒タイマの本数は可変、プログラマで範囲を指定)
	カウンタリレー/ レジスタ	512 点	←
	データレジスタ	8192 ワード	←
	リンクレジス	TOSLINE-S20 用 8192 点/1024 ワード (前半 512 ワードはビット指定可能)	TOSLINE-S20 用 16000 点/2048 ワード (前半 1000 ワードはビット指定可能)
	リンクリレー/ レジスタ	TOSLINE-F10 用 4096 点/256 ワード	←
	ファイルレジ	8192 ワード	32768 ワード
	インデックス レジスタ	I, J, K(計 3 ワード)	←
R A S	停電保持	補助リレー/レジスタ、タイマリレー/レジスタ、カウンタリレー/レジスタ、及びデータレジスタについて、停電保持領域の指定が可能	
	自己診断	瞬停チェック、基本/拡張電圧チェック、バッテリレベル、I/O バス、I/O 応答、I/O 照合、I/O パリティ、ウォッチドッグタイマ、イリガール命令、LP チェック、他	
	監視機能	イベント履歴、実行時間計測、シーケンス渋滞検出、他	
R A S	デバッグ/ メンテナンス	オンライントレースモニタ、フォース機能、I/O 活線着脱、サンプリングトレース、ステータスラッチ、ブレークポイント設定、他	

2.1 基本動作フロー

PC 本体の基本動作フローを下図に示します。



PC は、電源が投入されると、最初にシステム初期化処理を実行し、異常がなければモード制御処理を行います。このとき、RUN モードに入る条件が整っていれば、続いてスキャン制御に入ります。スキャン制御とは PC の基本機能であるユーザプログラム実行を行う処理です。モード制御処理において、RUN モードに入る条件が整わなければ、HALT モードとなり、ユーザプログラム不実行状態となります。

周辺サポート処理とは、プログラマやコンピュータリンクからの要求を受け付け、処理を行うもので、他の処理の空き時間に実行します。

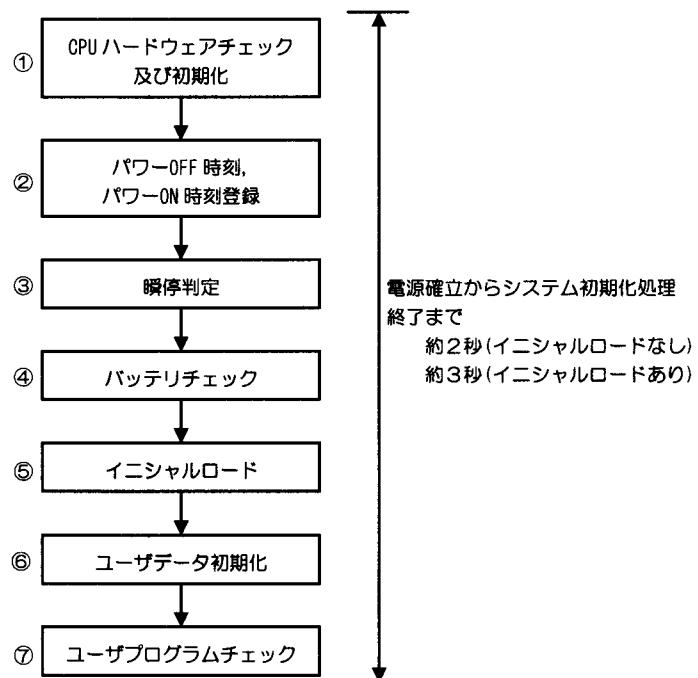
自己診断処理は、各処理の中で個別にも実行しますが、上図は、他の処理の空き時間に実行する自己診断処理を表わしています。

この章では、各処理の内容について以降に説明します。また、自己診断については、第5章 RAS 機能の中で説明します。

2.2 システム初期化処理

システム初期化処理は、電源投入後最初に実行する処理で、ハードウェアの自己診断、初期化を行い、その後、システムの初期化を行います。

下図に処理の流れを示し、各々の内容について説明します。



① CPU ハードウェアチェック及び初期化

システム ROM チェック、システム RAM のチェックと初期設定、周辺 LSI のチェックと初期設定、カレンダ LSI チェック、及びシーケンス演算プロセッサ (LP) のチェックを行います。

② パワーOFF 時刻、パワーON 時刻登録

イベント履歴テーブルに、前回電源 OFF 日時を登録し、さらにカレンダ LSI から読み出した現在日時をパワーON 時刻として登録します。また特殊レジスタ (SW007～SW013) に現在日時をセットします。

③ 瞬停判定

瞬停継続モード設定時 (S0400=ON のとき)、前回電源 OFF 時刻から今回電源 ON 時刻までの時間を計算し、これが2秒以内のときは、瞬停継続と判断し以降のイニシャルロードとユーザデータ初期化を行いません。
(前回電源 OFF 時 RUN モードだったことが条件)

④ バッテリチェック

ユーザプログラム及びユーザデータバックアップ用のバッテリの電圧をチェックします。バッテリ電圧が規定値以下のときには、イベント履歴テーブルに「バッテリ電圧低下異常」のメッセージを登録すると共に、特殊リレーのバッテリアラームフラグ (S000F) をセットします。

⑤ イニシャルロード

ユーザプログラム実行に先立って、ユーザプログラム及びデータレジスタの先頭 4K ワード (D0000～D4095) の内容を、周辺メモリ (IC メモリカードまたは EEPROM) からメインメモリ (RAM) に転送することをイニシャルロードと呼びます。

イニシャルロードは、RAM/ROM 切換スイッチが ROM 側でかつ運転モードスイッチが HALT 又は RUN の状態で電源が投入されたときに実行されます。

- IC メモリカードと EEPROM のいずれも存在しないときには、イニシャルロードは行いません。
- IC メモリカードと EEPROM の両方が存在するときには、IC メモリカードが転送元となります。
- IC メモリカードまたは EEPROM にユーザプログラムが書き込まれていて、その内容が壊れているとき (BOC エラー検出時) には、イニシャルロードは行いません。
- IC メモリカードにユーザプログラムが書き込まれていないときは、EEPROM が転送元となります。
- ③の瞬停判定で瞬停継続と判定されたときには、イニシャルロードは行いません。

⑥ ユーザデータ初期化

ユーザデータ (レジスタとデバイス) を下表の条件に従って初期化します。なお、③の瞬停判定で瞬停継続と判定された場合には、この処理は行いません。

レジスタ/デバイス	初期化状態
入力レジスタ/デバイス (XW/X)	入力フォースされた入力デバイスについては以前の状態を保持し、それ以外は0クリア
出力レジスタ/デバイス (YW/Y)	コイルフォースされた出力デバイスについては以前の状態を保持し、それ以外は0クリア
補助レジスタ/リレー (RW/R)	停電保持指定されたレジスタ及びコイルフォースされたデバイスについては以前の状態を保持し、それ以外は0クリア
特殊レジスタ/リレー (SW/S)	SW000～SW063 は CPU 設定部は初期化、ユーザ設定部は保持、SW064 以降は初期化
タイマレジスタ/リレー (T/T.)	停電保持指定されたレジスタとそれに対応するデバイスについては以前の状態を保持し、それ以外は0クリア
カウンタレジスタ/リレー (C/C.)	
データレジスタ (D)	停電保持指定されたレジスタについては以前の状態を保持し、それ以外は0クリア また、モードスイッチが P-RUN のときは先頭 4K ワード (0000～D4095) は保存

リンクレジスタ/リレー (W/Z) (TOSLINE-S20 用)	入カフォースされたリンクレジスタリレー(受信エリア)及びコイルフォースされたリンクレジスタリレー(送信エリア)については以前の状態を保持し、それ以外はOクリア
リンクリレー/レジスタ (L/LW) (TOSLINE-F10 用)	入カフォースされたリンクリレー(受信エリア)及びコイルフォースされたリンクリレー(送信エリア)については以前の状態を保持し、それ以外はOクリア
ファイルレジスタ (F)	全て保持
インデックスレジスタ (I, J, K)	全てOクリア

*1) フォース機能については、5.8 デバッグ機能を参照して下さい。

*2) 停電保持指定については、第3部 2.2 節を参照して下さい。

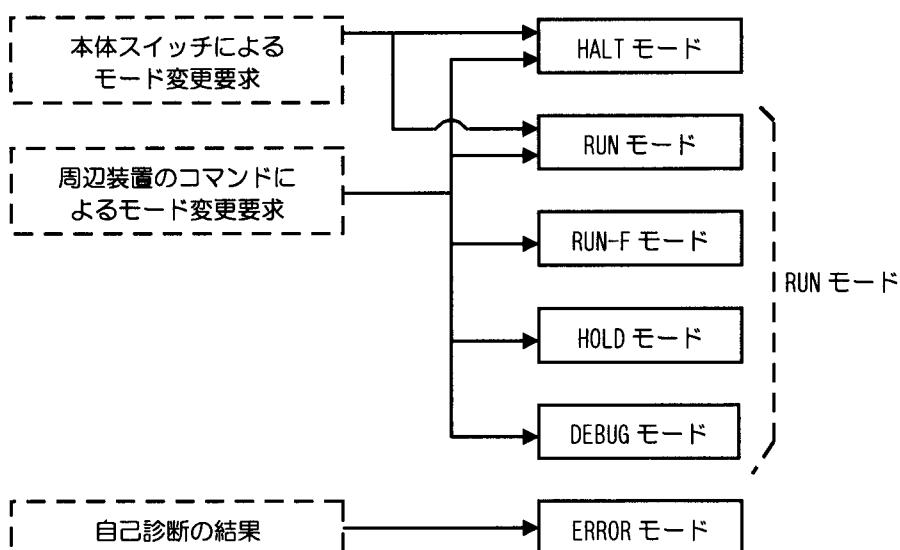
⑦ ユーザプログラムチェック

メインメモリ (RAM) 上のユーザプログラムの内容を BCC によりチェックします。

2.3 モード制御処理

モード制御処理では、CPU モジュール上のモードスイッチの状態と周辺装置 (プログラマ、コンピュータリンク、伝送装置) からのモード変更要求をチェックし、PC の動作モードを決定します。

PC の動作モードとしては、大きく分けると、RUN モード、HALT モード、及び ERROR モードの3つがあります。このうち、ERROR モードとは、各種自己診断の結果、所定の動作が継続できないと判断される場合に入るモードであり、ここで説明するモード制御処理のみでモード遷移条件が判定されるものではありません。また、RUN モードの中には、通常の RUN モードの他に、主にデバッグ/調整のために、RUN-F, HOLD, DEBUG の各モードが準備されています。



各モードでの動作について以下に説明し、次にこれらのモードに入るための条件(モード遷移条件)について説明します。

- | | |
|-------|---|
| HALT | :外部出力を全て OFF とし、ユーザプログラム実行及び入出力処理を停止します。HALT モードでは、モード制御処理を定期的(50ms 毎)に実行し、空き時間は周辺サポート処理及び自己診断処理に制御を移します。外部的には、ユーザプログラムの作成/変更を行うモードです。 |
| RUN | :イニシャルロード(必要時)、ユーザデータ初期化(必要時)、I/O モジュール実装状態チェック、ユーザプログラムチェック、及びスキャンモードの決定を行ってから RUN モードに入ります。
RUN モードでは、モード制御処理、一括入出力処理、タイマ更新処理、ユーザプログラム実行を繰り返し実行します。これをスキャン制御と呼びます。
スキャンの方法には、連続的に繰り返すフローティングスキャン方式と一定の周期で繰り返す定刻スキャン方式の2つがあり、この選択をスキャンモードの選択と呼びます。スキャン制御については、2.4 節及び第3章で詳しく説明します。 |
| RUN-F | :強制運転モードです。上記 RUN モードと異なる点は、起動時の I/O モジュール実装状態チェックにおいて、割り付けされたモジュールが未実装の状態でもスキャン制御に入る点です(割り付けと異なる種別のモジュールが実装されているときは運転不可)。
その他の処理は上記 RUN モードと同じです。 |
| HOLD | :スキャン一時停止モードです。スキャン制御の各処理のうち、一括入出力処理のみを実行し、タイマ更新処理、ユーザプログラム実行処理は停止します。スキャンモードは、それまで実行していた状態を継続します。
データモニタ/設定機能により、入出力モジュールのテストが行えます。 |
| DEBUG | :プログラムデバッグのための機能(ステップ実行、回路実行、Nスキャン実行、ブレークポイント設定、外部入力/出力処理の禁止等)が使用できるモードです。
DEBUG モードの中に D-HALT, D-STOP 及び D-RUN の3つのモードがあります。デバッグ機能については、5.11.3 節をご覧下さい。 |
| ERROR | :各種自己診断において、何らかの異常が検出され、所定のリトライ処理によても回復できないとき、正常動作不可能と判断して、このモードに入れます。ERROR モードでは、出力は全て OFF となり、プログラマからのエラーリセットコマンドのみが有効となります(エラーリセットコマンドによって HALT モードに復帰)。自己診断の内容については、第5章 RAS 機能をご覧下さい。 |

各モードへの遷移条件を以下に示します。

- HALT モードへの遷移条件

遷移前の状態				モード遷移要因	遷移後の動作モード	備考
動作モード	スタートモード	RAM/ROM	モードSW			
T3 の場合 (電源 OFF)		RAM	—	電源投入	HALT	INZ 実行
		ROM	HALT	電源投入		IL, INZ 実行
T3H の場合 (電源 OFF)	スタンバイ	RAM	—	電源投入	HALT	INZ 実行
		ROM	—	電源投入		IL, INZ 実行
ERROR	—	—	—	コマンド エラーリセット		
上記以外		—	RUN	モード SW → HALT		
			RUN/P-RUN	コマンド 停止		

- RUN モードへの遷移条件

遷移前の状態				モード遷移要因	遷移後の動作モード	備考
動作モード	スタートモード	RAM/ROM	モードSW			
T3 の場合 (電源 OFF)		ROM	RUN	電源投入	RUN	IL, INZ 実行
			P-RUN	電源投入		INZ 実行
		—	RUN/P-RUN	電源投入 (瞬停継続)		
T3H の場合 (電源 OFF)		オート	RAM	RUN	RUN	INZ 実行
			P-RUN	電源投入		INZ 実行
		ROM	RUN	電源投入	RUN	IL, INZ 実行
			P-RUN	電源投入		INZ 実行
HALT		—	—	RUN/P-RUN	電源投入 (瞬停継続)	
			RAM	HALT	モード SW → RUN	INZ 実行
		—	RUN/P-RUN	コマンド 運転		INZ 実行
			RAM	HALT	モード SW → RUN	IL, INZ 実行
			ROM	RUN	コマンド 運転	IL, INZ 実行
		ROM	P-RUN	コマンド 運転		INZ 実行
HOLD	—	—	RUN/P-RUN	コマンド ホールド解除	RUN または RUN-F	ホールド前のモードに復帰

- RUN-F モードへの遷移条件

遷移前の状態			モード遷移要因	遷移後の動作モード	備考
動作モード	RAM/ROM	モード SW			
HALT	RAM	RUN/P-RUN	コマンド 強制運転	RUN-F	INZ 実行
	ROM	RUN	コマンド 強制運転		IL, INZ 実行
		P-RUN	コマンド 強制運転		INZ 実行
HOLD	—	RUN/P-RUN	コマンド ホールド解除	RUN-F または RUN	ホールド前のモードに復帰

- HOLD モードへの遷移条件

遷移前の状態			モード遷移要因	遷移後の動作モード	備考
動作モード	RAM/ROM	モード SW			
RUN	—	RUN/P-RUN	コマンド ホールド	HOLD	
	—	RUN/P-RUN	コマンド ホールド		
	—	RUN/P-RUN	コマンド ホールド		

• DEBUG モードのモード遷移条件

遷移前の状態			モード遷移要因	遷移後の動作モード	備考
動作モード	RAM/ROM	モード SW			
HALT	—	RUN/P-RUN	コマンド デバッグ	D-HALT	
D-STOP	—	RUN/P-RUN	デバッグコマンド モード遷移		
D-HALT	—	RUN/P-RUN	デバッグコマンド 初回実行	INZ 実行	
			デバッグコマンド 繼続実行		
			デバッグコマンド ステップ実行		
			デバッグコマンド 回路実行		
D-STOP	—	RUN/P-RUN	デバッグコマンド 初回実行	D-RUN	INZ 実行
			デバッグコマンド 繼続実行		
			デバッグコマンド ステップ実行		
			デバッグコマンド 回路実行		
HOLD	—	RUN/P-RUN	コマンド ホールド解除		
D-RUN	—	RUN/P-RUN	設定スキャン回数到達	D-STOP	
			ブレーキポイント検出		
			停止条件成立		
			ステップ実行終了		
			回路実行終了		
			デバッグコマンド 強制停止		
D-HALT	—	RUN	モード SW → HALT	HALT	
		RUN/P-RUN	コマンド 停止		
D-STOP	—	RUN	モード SW → HALT		
		RUN/P-RUN	コマンド 停止		
D-RUN	—	RUN	モード SW → HALT		

*1) RAM/ROM 及びモード SW は、CPU モジュール上のスイッチの状態を示します。

*2) ーはスイッチの状態に依存しないことを示します。

*3) モード遷移要因中、「モード SW→××」はモードスイッチを××の位置に切り換えることを示します。

また「コマンド××」は周辺装置から××というコマンドを実行することを示します。(コマンドは、モードスイッチが RUN または P-RUN のときのみ有効)

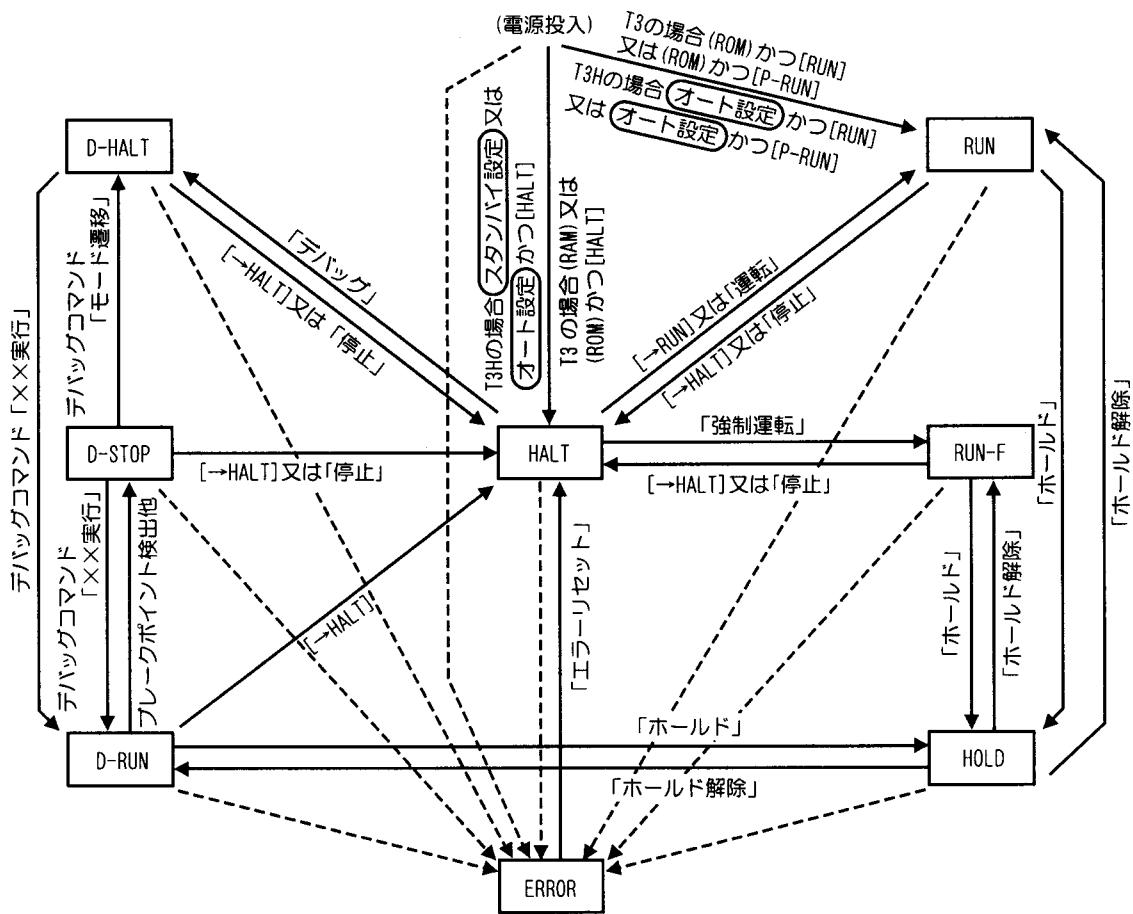
*4) モードスイッチ RUN↔P-RUN の切り換えは、動作モードには影響を与えません。(ただしプロテクト状態は切り換わります。5.4 節参照)

*5) 備考中、IL はイニシャルロードを、INZ はユーザデータの初期化を意味します。

*6) モード遷移要因が電源投入である場合の、イニシャルロードとユーザデータ初期化は、システム初期化処理の中で行われます。(2.2 節参照)

*7) デバック機能の詳細については、5.11.3 節をご覧下さい。

下図にモード遷移条件を図示します。



*1) ---はエラー発生時

*2) [→××]はモードスイッチを××の位置に切り換えることを示す

*3) 「××」は周辺装置からPC制御コマンド××を実行することを示す(モードスイッチRUNまたはP-RUNのときのみ有効)

*4) 電源投入時のモードスイッチの状態は[××]で、RAM/ROM切換スイッチの状態は(××)で示す

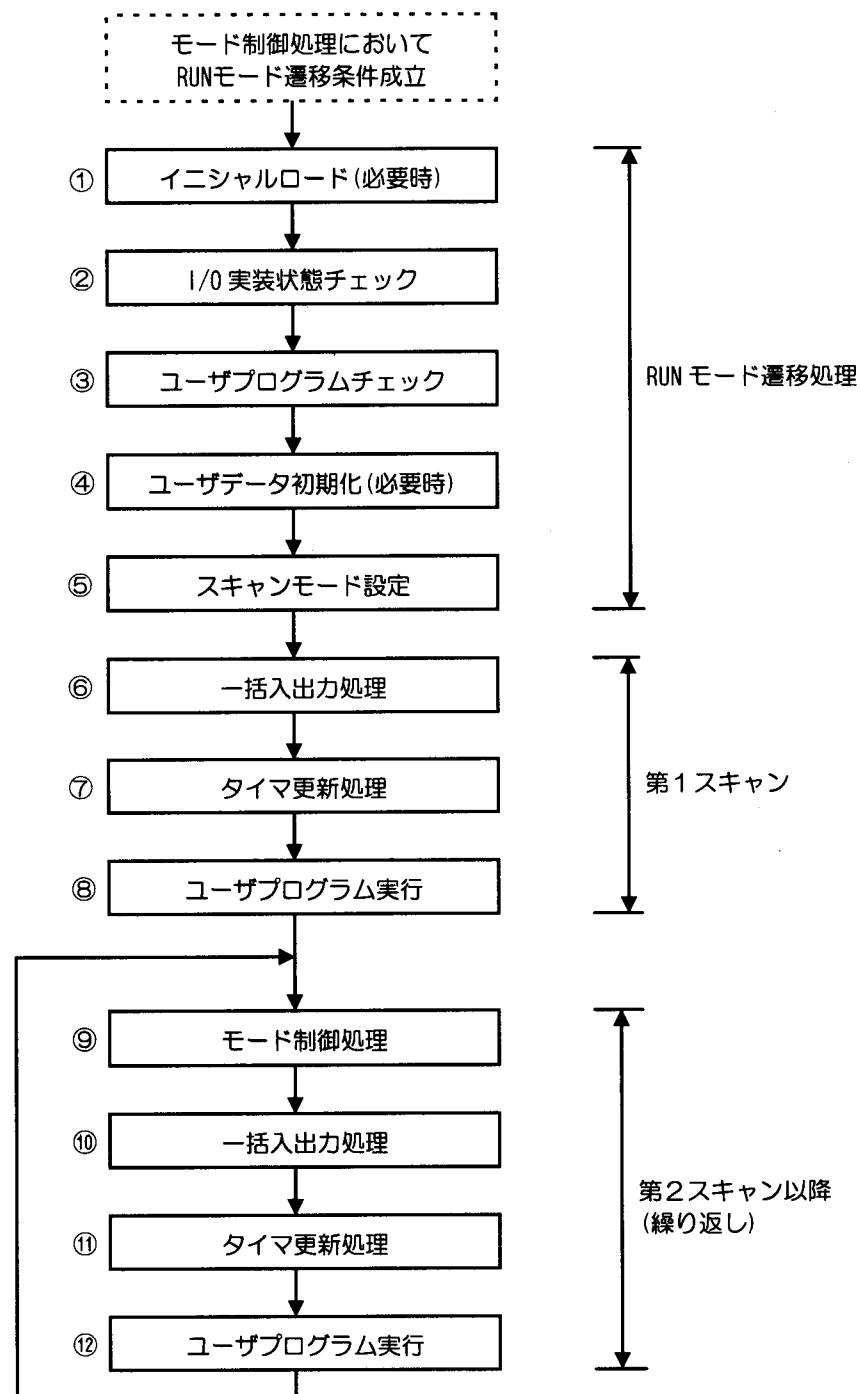
*5) (××)はT3H使用時、T-PDSによるスタートモードの設定状態です。

2.4 スキャン制御

PC の基本機能であるスキャン制御について説明します。

2.3 節に説明したように、モード制御処理において RUN モードに遷移する条件が整うと、イニシャルロード（必要時）、ユーザデータ初期化（必要時）、I/O 実装状態チェック、プログラムチェック、スキャンモード決定を行った上で、スキャン制御に入ります。スキャン制御では、モード制御処理、一括入出力処理、タイマ更新処理、ユーザプログラム実行処理を繰り返し実行します。

下図にスキャン制御の処理の流れを示します。



① イニシャルロード

CPU モジュール上の RAM/ROM 切換スイッチが ROM 側かつモードスイッチが RUN の状態で RUN 起動がかけられたとき、以下の条件に従って、周辺メモリ (IC メモリカードまたは EEPROM) からメインメモリ (RAM) へ、ユーザプログラムとデータレジスタの先頭 4K ワード (D0000～D4095) の内容を転送します。(瞬停継続時は不実行)

- IC メモリカードと EEPROM のいずれも存在しないときには、イニシャルロードは行いません。
- IC メモリカードと EEPROM の両方が存在するときには、IC メモリカードが転送元となります。
- 装着された IC メモリカードにユーザプログラムが書き込まれていないときには、EEPROM が転送元となります。
- IC メモリカードまたは EEPROM にユーザプログラムが書き込まれていて、その内容が壊れているとき (BCC エラー検出時) には、イニシャルロードは行わず、ERROR モードに遷移します。

② I/O 実装状態チェック

入出力割り付け情報を基に、I/O モジュールの実装状態をチェックします。(詳細は第5章 RAS 機能をご覧下さい)

③ ユーザプログラムチェック

メインメモリ (RAM) 上のユーザプログラムを対象に BCC チェックを行います。(詳細は第5章 RAS 機能をご覧下さい)

④ ユーザデータ初期化

HALT モードから RUN モードに遷移したとき、ユーザデータの初期化を行います。初期化の内容については、2.2 システム初期化処理を参照して下さい。(瞬停継続時は不実行)

⑤ スキャンモード設定

スキャンモード (フローティングスキャンまたは定刻スキャン) の設定を行います。

スキャンモードについては、2.4.1 節で説明します。

⑥, ⑩一括入出力処理

入出力割り付け情報を基に、入出力イメージテーブル(入出力レジスタ/デバイス)と I/O モジュール間でデータのやり取りを行います。また、データ伝送装置(TOSLINE-S20, TOSLINE-F10)とのデータ交信も行います。なお、第1スキャンでは I/O モジュールからの入力処理のみを行います。I/O モジュールへの出力処理、データ伝送装置とのデータ交信については、第2スキャンから行います。

一括入出力処理については、2.4.2 節で説明します。

⑦, ⑪タイマ更新処理

タイマ命令で使用しているタイマレジスタの更新、及び特殊リレーのタイミングリレー(S0040～S0047)の更新を行います。

タイマ更新処理については、2.4.3 節で説明します。

⑧, ⑫ユーザプログラム実行

ユーザプログラムの命令語を先頭から END 命令まで順に実行します。ここで実行の対象となるユーザプログラムは、メインプログラムとサブプログラムです。

割り込みプログラムは割り込み要因発生時、他の処理を中断して即座に実行されます。

ユーザプログラムの実行制御については、第3章で詳しく説明します。

⑨ モード制御処理

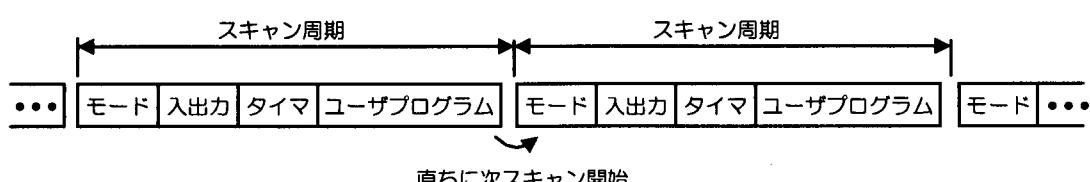
モードスイッチ及びプログラマからのモード変更コマンドをチェックし、動作モード変更の処理を行います。また、スキャン周期及びユーザプログラム実行時間の計測を行い、スキャンのタイミング制御を行います。

2.4.1 スキャンモード

PC ではスキャンの方式として、フローティングスキャンと定刻スキャンの選択が可能です。

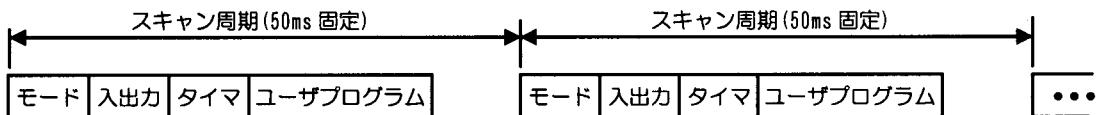
フローティングスキャンとは、1 スキャンの処理終了後直ちに次スキャンの処理を開始する処理方式です。最短のスキャン周期となります。スキャン周期はユーザプログラムの実行状態によって伸び縮みします。

フローティングスキャンの動作を下図に示します。



一方、定刻スキャンとは、設定された時間周期でスキャンの処理を実行する処理方式です。周期の設定範囲は 10~200ms (10ms 単位) です。スキャン周期のばらつきをなくしたいときに使用します。

周期 50ms に設定したときの定刻スキャンの動作を下図に示します。

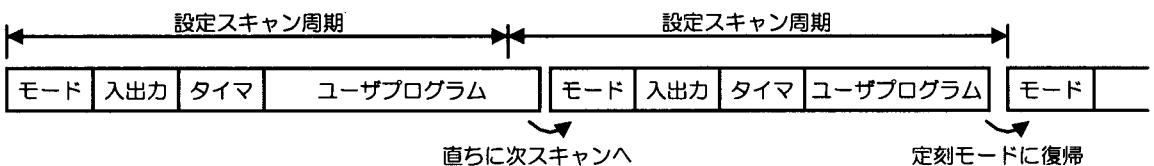


スキャンモードの選択は、プログラマから、システム情報のスキャン時間の項目にスキャン周期を設定することによって行われます。

フローティングスキャンの場合は、スキャン時間を設定なし(ブランク)とします。定刻スキャンの場合は、スキャン時間に 10~200ms (10ms 単位) の範囲でスキャン周期を設定します。

— 補足 —

定刻スキャンにおいて、1スキャンの処理時間が設定周期を越えた場合には、フローティングスキャンとなり、定刻スキャン渋滞フラグ(特殊リレー S0008)が ON となります。なお、スキャン処理時間が設定周期以内に戻った場合には、当初のスキャン周期で定刻スキャンに復帰します。



2.4.2

一括入出力処理

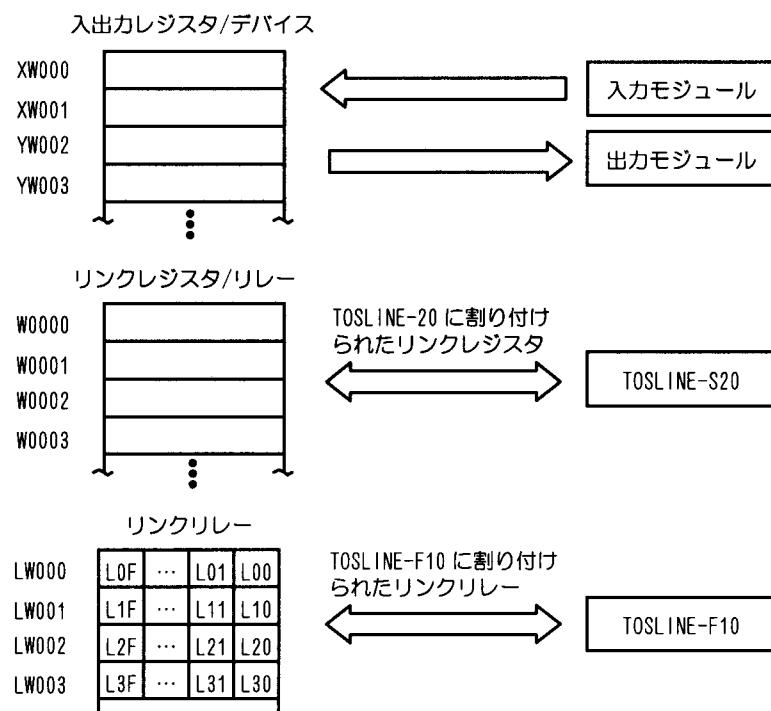
入力モジュールに入力された外部信号の状態を入力レジスタ/デバイス(XW/X)に読み込み、出力レジスタ/デバイス(YW/Y)の状態を出力モジュールに出力にします。この処理をユーザプログラム実行に先立って、一括して実行しますので一括入出力処理と呼びます。一括入出力の対象は次の通りです。

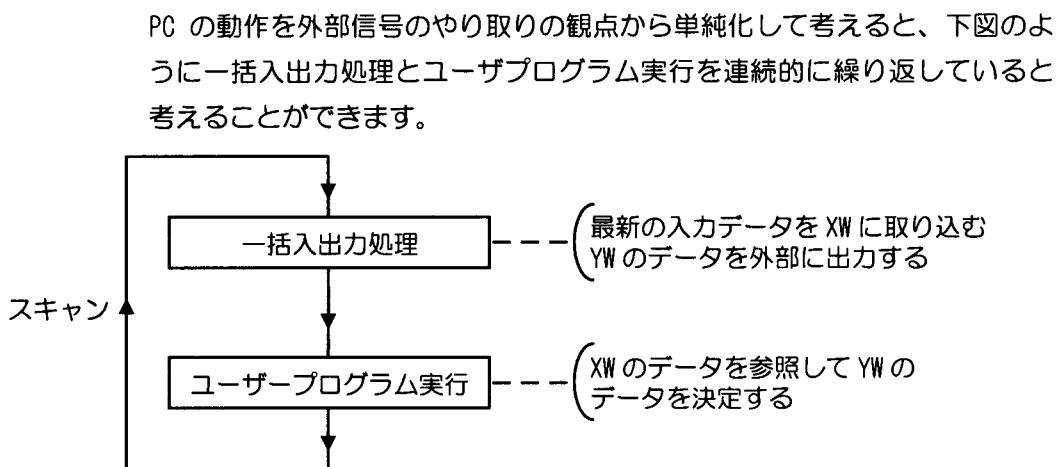
一括入力の対象 … 入出力割り付け上 i 指定の付いていない入力モジュールの信号で、かつフォース指定されていない入力レジスタ/デバイス(XW/X)

一括出力の対象 … 入出力割り付け上 i 指定の付いていない出力モジュールに対応する出力レジスタ/デバイス(YW/Y)

2

また、データ伝送装置(TOSLINE-S20, TOSLINE-F10)とCPUモジュール内のリンクレジスタ/リレー(W/Z及びLW/L)とのデータ交信もこの処理で実行します。





このように、ユーザプログラム実行の中には、基本的に、I/O モジュールとのデータやり取りを行わないため、高速スキャンを実現することができ、またユーザプログラム実行の中では XW のデータは変化しないためプログラムの論理が考えやすいという特長があります。この方式を一括入出力方式(リフレッシュ方式)と呼びます。

なお PC では、XW/X の代わりに IW/I を、YW/Y の代わりに OW/O を使用することによって、ユーザプログラム実行の中にも I/O モジュールとのデータやり取りを行う方式をとることもできます。この方式を直接入出力方式(ダイレクト方式)と呼びます。この場合には、直接入出力で使用する I/O モジュールを一括入出力の対象から外し(入出力割り付け上 i 指定を付けます)、一括入出力処理の処理時間を短くすることをお勧めします。

補足

- (1) 一括入出力処理の処理時間は下記を目安にして下さい。

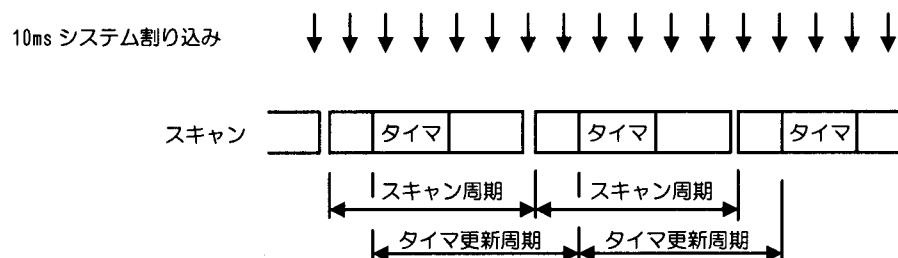
	T3	T3H	
入力 (XW)	54μs/1 レジスタ当たり	基本	12μs/1 レジスタ当たり
		拡張	17μs/1 レジスタ当たり
出力 (YW)	43μs/1 レジスタ当たり	基本	12μs/1 レジスタ当たり
		拡張	17μs/1 レジスタ当たり
リンク (W)	30μs/1 レジスタ当たり	基本	9μs/1 レジスタ当たり
		拡張	15μs/1 レジスタ当たり

- (2) 入出力割り付け上、i 指定の付いた I/O モジュール(iX, iY, iX+Y)は、一括入出力の対象外となります。入出力割り付けについては、第3部をご覧下さい。
- (3) フォース指定がかかった入力デバイス(X)、リンクレジスタリレー(Z)、及びリンクリレー(L)は一括入出力の対象外となります。フォース機能については、第5章にて説明します。
- (4) データ伝送装置へのリンクレジスタ/リレー(W/Z 及び LW/L)の割り付けは、データ伝送装置の説明書をご覧ください。
- (5) 直接入出力方式とした場合、ビット(0)で指定してもレジスタ単位で出力されます。直接入出力レジスタについては、第3部をご覧下さい。

2.4.3 タイマ更新処理

タイマ命令で使用しているタイマレジスタの更新(加算)処理、及び特殊リレー中のタイミングリレー(S0040～S0047)の更新を行います。

- タイマレジスタの更新



タイマ更新周期(≈スキャン周期)内に発生したシステム割り込みの回数をカウントし、タイマ命令(TON, TOF, SS, TRG)で起動がかかっているタイマレジスタの内容に、このカウント値を加算します。

0.01秒タイマについては10ms割り込みを使用し、0.1秒タイマについては10ms割り込みを積算して使用します。タイマのリセット及びタイムアップの判定はタイマ命令実行時に行います。

タイマ種別	タイマレジスタ (タイマリレー)		設定値範囲	備考
	T3	T3H		
0.01秒タイマ	T000～T063 (T. 000～T. 063)	T000～T999 (可変) (T. 000～)	0～32767 (0～327.67秒)	オンディレイタイマ(TON) オフディレイタイマ(TOF)
0.1秒タイマ	T064～T511 (T. 064～T. 511)	T000～T999 (可変) (～T. 999)	0～32767 (0～3276.7秒)	シングルショットタイマ(SS) タイマトリガ(TRG)

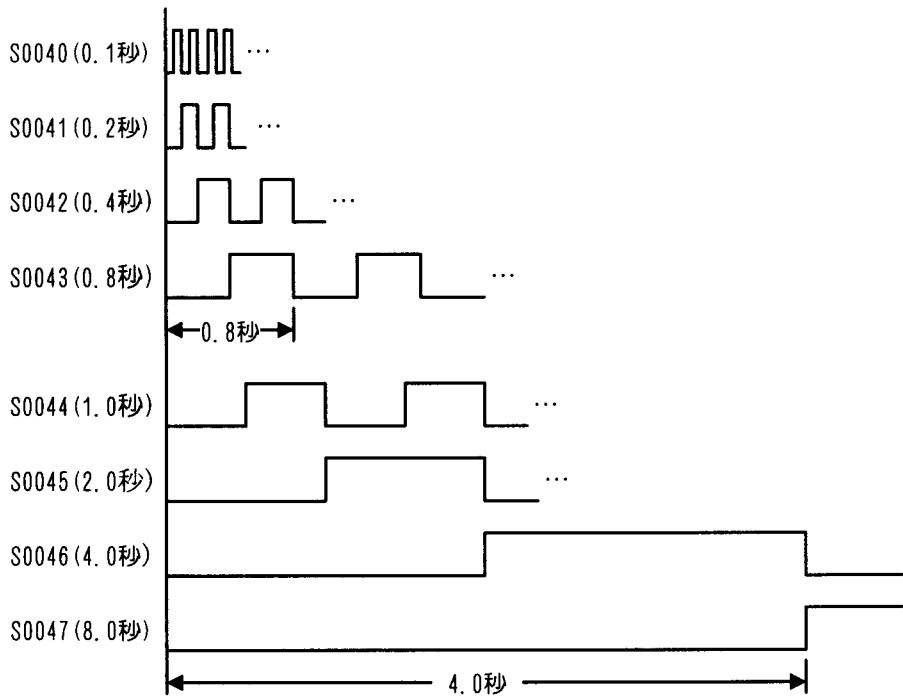
*) タイマレジスタ更新の処理時間は下記を目安に考えて下さい。

[T3] 27μs/タイマ1点当たり(更新時)

[T3H] 7μs/タイマ1点当たり(更新時)

- タイミングリレーの更新

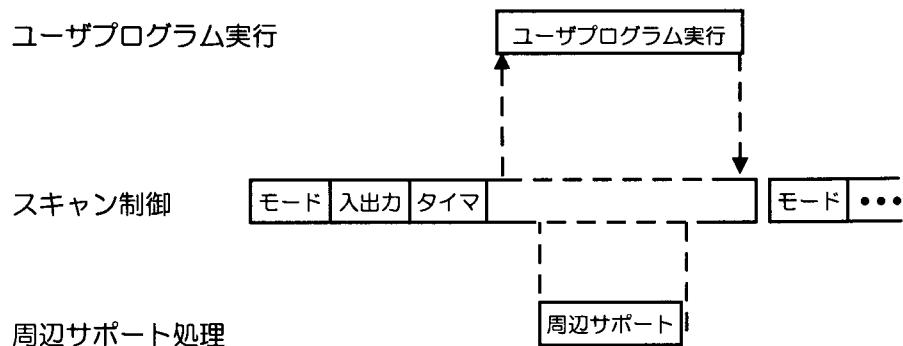
10msシステム割り込みを使用して、特殊リレー中のタイミングリレー(S0040～S0047)のON/OFF状態を制御します。次ページのようにバイナリカウンタを構成します。(RUN起動時は全てOFF)



2.5 周辺サポート処理

周辺サポート処理は、周辺装置(プログラマ、コンピュータリンク、伝送装置)からの要求コマンドを解釈し、要求処理を実行し、返答を返す処理を行います。

ユーザプログラム実行中は、実際の命令語実行はシーケンス演算プロセッサ(LP)が受け持つますので、メインCPUには空き時間が生じます。メインCPUはこの空き時間を利用して周辺サポート処理を行います。



*1) 周辺サポート処理において、データの同期が必要なコマンドに対しては、コマンドの解釈は並列処理にて行い、データ収集等の処理はスキャン終了後に一括して行います。さらに返信テキストの作成は次の空き時間に処理します。

*2) 2つ以上の要求元から同時に要求コマンドを受け付けたときの処理の優先順は次の通りです。

プログラマ > コンピュータリンク > TOSLINE-S20 (ch1) > TOSLINE-S20 (ch2)

2.6 プログラミングサポート機能

プログラミングサポート機能は、周辺サポート処理の結果として実現される機能の一部です。プログラミングサポート機能の詳細は別冊のプログラマ操作説明書にて説明しますが、ここではこれらの機能の概要と T3 本体の動作モードとの関連について説明します。

(1) メモリクリア

メモリクリアコマンドを受け付け、ユーザプログラムメモリ (RAM) の内容を初期化し、ユーザデータメモリ (RAM) の内容を 0 クリアします。

(2) 入出力自動割り付け

入出力自動割り付けコマンドを受け付け、実装されている I/O モジュールの種別を読み出して、入出力割り付け情報としてシステム情報内に格納します。(システム情報はユーザプログラムメモリ内)

(3) 入出力割り付け情報読み出し

システム情報から入出力割り付け情報を読み出して、周辺装置に送ります。

(4) 入出力割り付け情報書き込み

周辺装置から送られてきた入出力割り付け情報を受け取り、システム情報内に格納します。

(5) システム情報読み出し

システム情報(プログラム ID、停電記憶指定、使用ステップ数、スキンモード指定他)を読み出して、周辺装置に送ります。

(6) システム情報書き込み

周辺装置から送られてきたシステム情報(ユーザ設定項目)を受け取り、システム情報に格納します。

(7) プログラム読み出し

周辺装置からの要求に応じて、所定の範囲のユーザプログラムメモリの内容を読み出して、周辺装置に送ります。

(8) プログラム書き込み

周辺装置から送られてきた所定範囲のユーザプログラムを受け取り、ユーザプログラムメモリの内容を書き換えます。また、書き換えた後で BCC(チェックコード)を作り直します。

(9) オンラインプログラム変更 (T3 ではバージョン 1.2 以降)

RUN モード中に、ユーザプログラムメモリの内容を書き換え(追加/変更/挿入/削除)、BCC(チェックコード)を作り直します。この処理は、1スキャン終了後に行いますので、処理の間、スキャン周期は伸びます。

オンラインプログラム変更には、以下の制約があります。

- 実行制御にかかる命令語(下記)の個数や実行順序が変わるような変更はできません。
END, MCS, MCR, JCS, JCR, JUMP, LBL, FOR, NEXT, CALL, SUBR, RET, IRET
- SFC プログラム部分については、SFC 構造の変更はできませんが、ステップ及び遷移に対応する詳細部(ラダー図)部分は変更できます。

(10) プログラム一括読み出し

ユーザプログラムメモリの内容(システム情報含む)を読み出して、周辺装置に送ります。

プログラムのアップロード(PC→プログラマ→FD 等)に使用します。

(11) プログラム一括書き込み

周辺装置から送られてきたユーザプログラム(システム情報含む)を受け取り、ユーザプログラムメモリに格納します。

プログラムのダウンロード(FD→プログラマ→PC 等)に使用します。

(12) 検索

周辺装置から指定された命令語/オペランドをユーザプログラムメモリから探し出し、そのアドレスを周辺装置に送ります。

(13) プログラムチェック

プログラムチェックコマンドを受け付け、ユーザプログラムの文法チェックを行います。そしてチェック結果を周辺装置に送ります。

(14) データ読み出し

周辺装置からの要求に応じて、ユーザデータメモリから指定のデータ内容を読み出して、周辺装置に送ります。

(15) データ書き込み

周辺装置から送られてきたユーザデータのアドレスとデータ内容を受け取り、ユーザデータメモリに格納します。

(16) IC メモリカード (EEPROM) 読み出し

ICメモリカードの装着有無及びEEPROMの内蔵有無をチェックし、ICメモリカードまたはEEPROMの内容をチェックした上でメインメモリ(RAM)のユーザプログラムメモリとユーザデータメモリ(RW, T, C, D)に転送します。ICメモリカードとEEPROMの両方が存在するときには、ICメモリカードを対象とします。

(17) ICメモリカード(EEPROM)書き込み

IC メモリカードの装着有無及び EEPROM の内蔵有無をチェックし、いずれか存在するときには、ユーザプログラムメモリとユーザデータメモリ (RW, T, C, D) の内容を IC メモリカードまたは EEPROM に転送します。IC メモリカードと EEPROM の両方が存在するときには、IC メモリカードを対象とします。

(18) IC メモリカード初期化 (T3H のみ可能)

IC メモリカードに存在するユーザプログラムとユーザデータもしくはサンプリングデータを初期化(0クリア)します。

これらの機能の実行条件を下表に示します。

機能	実行条件
入出力割り付け情報読み出し	常時実行可能(ただしシステム初期化処理で異常を検出し、周辺装置との通信不能な場合を除く)
システム情報読み出し	
プログラム読み出し	
データ読み出し	
プログラム一括読み出し	ERROR モード時以外実行可能
検索	
プログラムチェック	HALT モード時のみ実行可能
IC カード(EEPROM)書き込み	
IC カード初期化*	
メモリクリア	HALT モードかつモードスイッチ P-RUN 以外のとき実行可能
入出力自動割り付け	
入出力割り付け情報書き込み	
システム情報書き込み	
プログラム書き込み	
プログラム一括書き込み	
IC カード(EEPROM)読み出し	
オンラインプログラム変更	ERROR モード以外かつ P-RUN 以外のとき実行可能
データ書き込み	ERROR モード以外実行可能、ただし P-RUN のときはデータレジスタの 1 部(D0000~D4095)は書き込み禁止

* T3H のみ可能

補足

本体にパスワードが設定された場合には、パスワードの許可レベルにより実行可能なコマンドが決まります。パスワード機能については 5.13 節をご覧下さい。

3.1

プログラム種別

PC では種別の異なる複数のプログラムを並列に実行することができます(この機能をマルチタスク機能と呼びます)。この機能により、制御対象に応じた最適の応答時間を実現することができます。

プログラム種別としては以下の通り 3 種類、合計 14 本があります。

- メインプログラム(1本)

毎スキャン必ず実行されるプログラムであり、スキャン制御の主体となります。

- サブプログラム(4本)

他のプログラム(メインプログラム、他のサブプログラム、割り込みプログラム)からの起動要求によって実行されるプログラムであり、#1～#4 の4本が準備されています。(ただし#1 は機能固定)

サブプログラムの実行は、フローティングスキャンの場合には、メインプログラムに統いて実行時間制限(ユーザ設定)付きで実行され、定刻スキャンの場合には、メインプログラム実行終了から次スキャン開始までの空き時間に実行されます。

時間のかかるデータ処理などをサブプログラムにて実行させることによって、メインプログラムのスキャン周期を伸ばさずにデータ処理を行うことや、またメインプログラムを高速スキャン、サブプログラムを低速スキャンとして並列実行させることなどが可能となります。

- 割り込みプログラム(9本)

割り込み要因が発生したときに他の処理を中断して即座に実行されるプログラムです。ユーザが設定した周期で起動されるプログラム(定周期割り込みプログラム)が1本、割り込み機能付き I/O モジュールからの割り込み信号によって起動されるプログラム(I/O 割り込みプログラム)が8本(#1～#8)の合計9本が準備されています。

定周期割り込みによって、細かい時間制御、効率的なマルチタスク制御を行うことや、I/O 割り込みによって、スキャン周期に影響されない入出力応答を実現することなどが可能となります。

この章では、サブプログラム、割り込みプログラムの実行方法、実行条件について説明します。

3.2 メイン/サブプログラム実行制御

サブプログラムとしては、下表のように#1～#4 の4本を登録することができ、下表の実行条件によって実行されます。

サブプログラム#1(以下サブ#1と略します)は、RUN 起動時にメインプログラム(以下メインと略します)に先立って1回だけ実行されます。サブ#2 は設定により、一般モードと特殊モードの選択が可能です。サブ#3 とサブ#4 は一般モードに固定です。また一般モードにおいては、起動要求発生時に1回だけ実行される1回実行モードと、起動要求がある間、設定したスキャン回数毎に1回実行されるサイクリック実行モードの選択が可能です。

No.	一般モード /特殊モード	1回実行/サイクリック実行	動作
サブ#1	—	—	RUN 起動時の第1スキャンにメインの前に実行します(一括入力後)。
サブ#2	S0403=0のとき 一般モード	S0405=0 のとき 1回実行モード	S0409=1 のとき実行します。 (S0409 はシステム側でリセット)
		S0405=1 のとき サイクリック 実行モード	S0409=1 の間、SW042 に設定されたスキャン回数毎に1回実行します。 (S0409 はユーザプログラムでリセット)
	S0403=1 のとき特殊 モード	—	停電時間が2秒以内で立ち上がった場合で、S0400=1 のとき、RUN 起動時の第1スキャンにメインの前に実行します(一括入力後)。 このときサブ#1 は不実行となります。
サブ#3	一般モードのみ	S0406=0 のとき1回実行モード	S040A=1 のとき実行します。 (S040A はシステム側でリセット)
		S0406=1 のときサイク リック実行モード	S040A=1 の間、SW043 に設定されたスキャン回数毎に1回実行します。 (S040A はユーザプログラムでリセット)
サブ#4	一般モードのみ	S0407=0 のとき1回実行モード	S040B=1 のとき実行します。 (S040B はシステム側でリセット)
		S0407=1 のときサイク リック実行モード	S040B=1 の間、SW044 に設定されたスキャン回数毎に1回実行します。 (S040B はユーザプログラムでリセット)

各サブプログラムに関連するフラグ(特殊リレー/レジスタ)を下表にまとめます。

サブ No.	関連フラグ(名称)	機能		
サブ#1	S0410(サブ#1 実行中)	0:不実行	1:実行中	(ステータス)
サブ#2	S0400(瞬停処理選択)	0:通常モード	1:継続モード	(設定)
	S0403(サブプログラムモード)	0:一般モード	1:特殊モード	(設定)
	S0405(サブ#2 実行モード)	0:1回実行	1:サイクリック実行	(設定)
	S0409(サブ#2 起動要求)	0:要求なし	1:要求あり	(コマンド)
	SW042(サブ#2 スキャン回数)	設定回数毎に実行(サイクリック時)		
	S0411(サブ#2 実行中)	0:不実行	1:実行中	(ステータス)
	S0415(サブ#2 渋滞中)	0:正常	1:渋滞中	(ステータス)
サブ#3	S0406(サブ#3 実行モード)	0:1回実行	1:サイクリック実行	(設定)
	S040A(サブ#3 起動要求)	0:要求なし	1:要求あり	(コマンド)
	SW043(サブ#3 スキャン回数)	設定回数毎に実行(サイクリック時)		
	S0412(サブ#3 実行中)	0:不実行	1:実行中	(ステータス)
	S0416(サブ#3 渋滞中)	0:正常	1:渋滞中	(ステータス)
サブ#4	S0407(サブ#4 実行モード)	0:1回実行	1:サイクリック実行	(設定)
	S040B(サブ#4 起動要求)	0:要求なし	1:要求あり	(コマンド)
	SW044(サブ#4 スキャン回数)	設定回数毎に実行(サイクリック時)		
	S0413(サブ#4 実行中)	0:不実行	1:実行中	(ステータス)
	S0417(サブ#4 渋滞中)	0:正常	1:渋滞中	(ステータス)

*) 表中、(設定)とは動作モード設定のためにユーザが設定するフラグを、(コマンド)とは起動をかけるためにユーザが ON するフラグを示します。(ステータス)は実行状態モニタ用フラグです。

(1) サブ#1 の動作

サブ#1 は RUN 起動時の第 1 スキャンにメイン実行前に 1 回だけ実行されます。従って、サブ#1 は運転開始時の初期設定プログラムとして利用して下さい。

サブ#1 が存在する場合で、サブ#2 の特殊モードが起動されない場合の動作を下図に示します。



(2) サブ#2 特殊モードの動作

サブ#2 の特殊モードは、停電時間が2秒以内で、かつ瞬停処理選択フラグ(S0400)がONのとき(瞬停継続モード設定)に、RUN起動時の第1スキャンにメイン実行前に1回だけ実行されます。(サブ#2 特殊モード実行時はサブ#1は実行されません)

また、瞬停継続モード設定時に停電時間が2秒以内で立ち上がった場合には、システム初期化処理において、イニシャルロード及びユーザデータ初期化は行われません。

従って、サブ#2 の特殊モードは、瞬停再始動時の初期設定プログラムとして利用して下さい。

下図にサブ#2 特殊モードの動作を示します。



3

(3) 一般モード(サブ#2, サブ#3, サブ#4)

サブプログラムの実行は、メインプログラムの実行終了から次スキャン開始までの間に行われます。従ってフローティングスキャンモードの場合と定刻スキャンモードの場合では動作が異なります。

フローティング …… フローティングスキャンの場合は、システム情報スキャン時にサブプログラム実行時間を設定します。設定は1~100ms(1ms単位)です。起動要求があったサブプログラムは、この設定時間内でメインプログラムの後に実行されます。設定時間内に実行が終了しないときには、実行が中断され、次スキャンに継続実行されます。

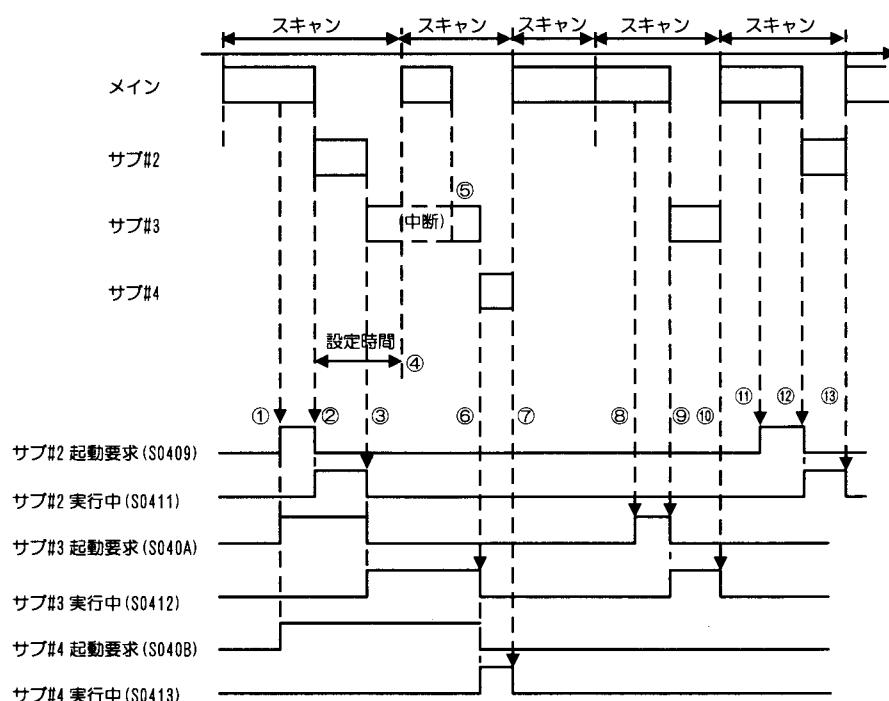
定刻スキャン時 …… 定刻スキャンの場合は、メインプログラム実行終了から次スキャン開始までの時間、つまり定刻周期の残り時間にサブプログラムが実行されます。定刻周期になっても実行が終了しないときには、実行が中断され、次スキャンに継続実行されます。

また、サブプログラムの実行モードとして、起動要求があったら1回だけ実行する1回実行モードと、起動要求がある間、指定スキャン回数毎に1回づつ実行するサイクリック実行モードの2つがあります。これらの実行モードは、サブプログラム各々について個別に設定可能です。

(3-a) 1回実行モード

起動要求のチェックは、メインプログラム実行終了後、及びサブプログラム実行終了後に行います。このとき同時に2つ以上のサブプログラムに対して起動要求がONしている場合には、サブ#2→サブ#3→サブ#4の優先順で要求を受け付けます(ただし、1スキャン内で起動要求チェックは一巡のみ)。起動要求フラグは、要求受け付け時(起動時)にリセットします(ユーザリセット不要)。起動要求は、メインプログラム、他のサブプログラム、及び割り込みプログラムから、かけることができます。

- フローティングスキャンの場合の動作を下図に示します。

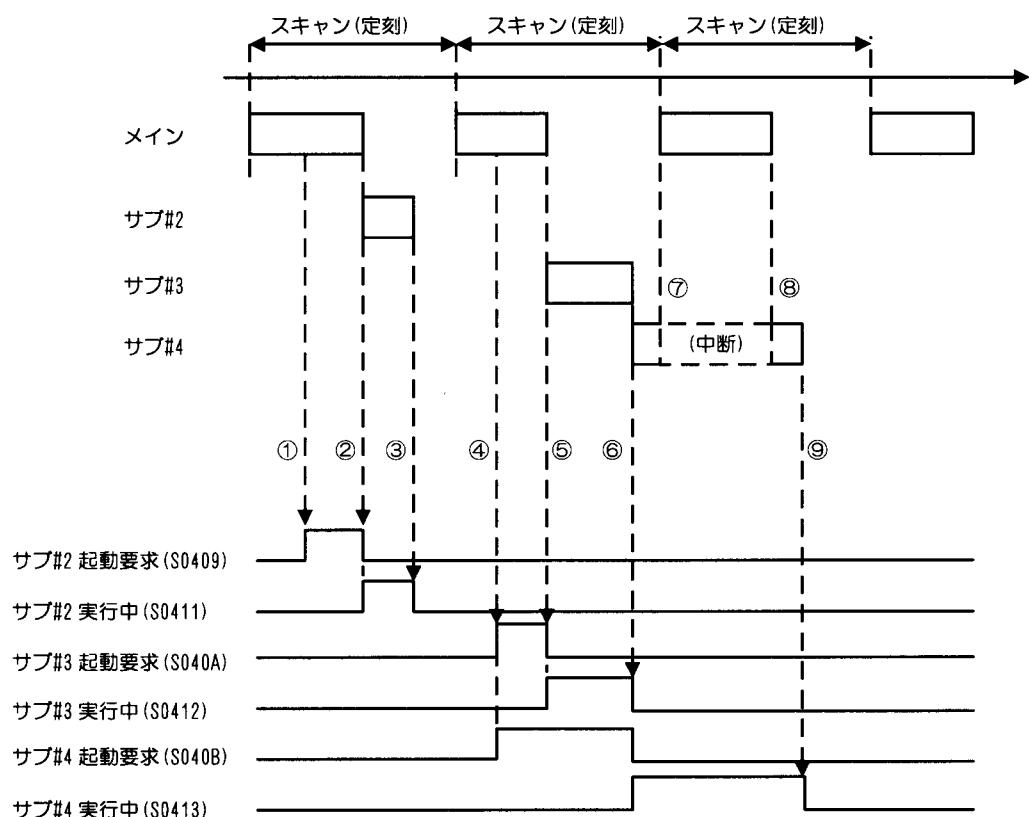


図中のタイミングは次の通りです。

- ① … メインから、サブ#2, サブ#3, サブ#4 に同時に起動要求
- ② … メイン終了後、サブ#2 起動(サブ#2 起動要求フラグは自動リセット)
- ③ … サブ#2 終了、サブ#3 起動(サブ#3 起動要求フラグは自動リセット)
- ④ … 設定時間経過によりサブ#3 実行中断、次スキャン開始、(サブ#3 実行中フラグ継続)
- ⑤ … メイン終了後、サブ#3 繼続実行開始
- ⑥ … サブ#3 終了、サブ#4 起動(サブ#4 起動要求フラグは自動リセット)
- ⑦ … サブ#4 終了、次スキャン開始
- ⑧ … メインからサブ#3 に起動要求
- ⑨ … メイン終了後、サブ#3 起動(サブ起動要求フラグは自動リセット)

- ⑩ … サブ#3 終了、次スキャン開始
- ⑪ … メインからサブ#2 起動要求
- ⑫ … メイン終了後、サブ#2 起動(サブ#2 起動要求フラグは自動リセット)
- ⑬ … サブ#2 終了、次スキャン開始

- 定刻スキャンの場合の動作を下図に示します。



図中のタイミングは次の通りです。

- ① … メインからサブ#2 に起動要求
- ② … メイン終了後、サブ#2 起動(サブ#2 起動要求フラグは自動リセット)
- ③ … サブ#2 終了
- ④ … メインからサブ#3、サブ#4 に起動要求
- ⑤ … メイン終了後、サブ#3 起動(サブ#3 起動要求フラグは自動リセット)
- ⑥ … サブ#3 終了、サブ#4 起動(サブ#4 起動要求フラグは自動リセット)
- ⑦ … 定刻スキャン周期到達、サブ#4 実行中断
- ⑧ … メイン終了後、サブ#4 繼続実行開始
- ⑨ … サブ#4 終了

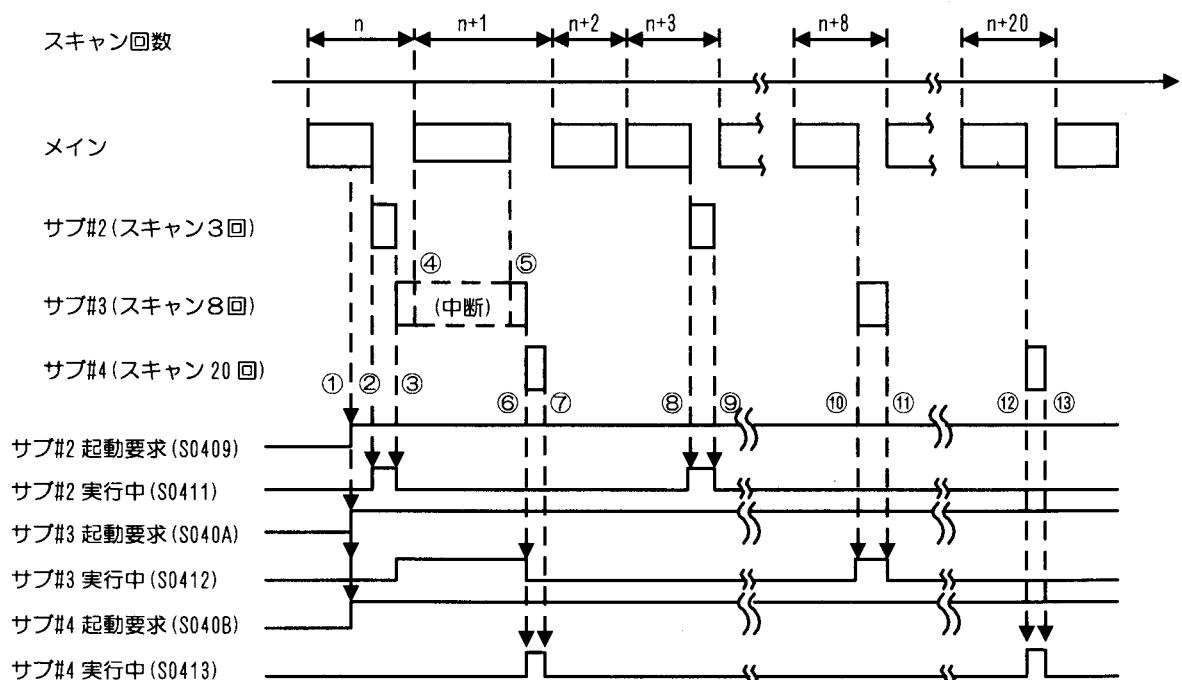
(3-b) サイクリック実行モード

起動要求フラグが ON している間、設定されたスキャン回数毎に 1 回づつ繰り返し実行します。複数のサブプログラムで起動タイミングが重なった場合の優先度は、サブ#2→サブ#3→サブ#4 の順です。サイクリック実行モードによって、多重スキャン(高速スキャン、中速スキャン、低速スキャン等)を実現することができます。

なお、起動要求フラグのセット/リセットはユーザプログラムで行います。

また、設定されたスキャン回数で、サブプログラムの実行が完了しない場合には、サブプログラム渋滞中フラグ(S0415, S0416, S0417)が ON します。

- フローティングスキャンの場合の動作を下図に示します。

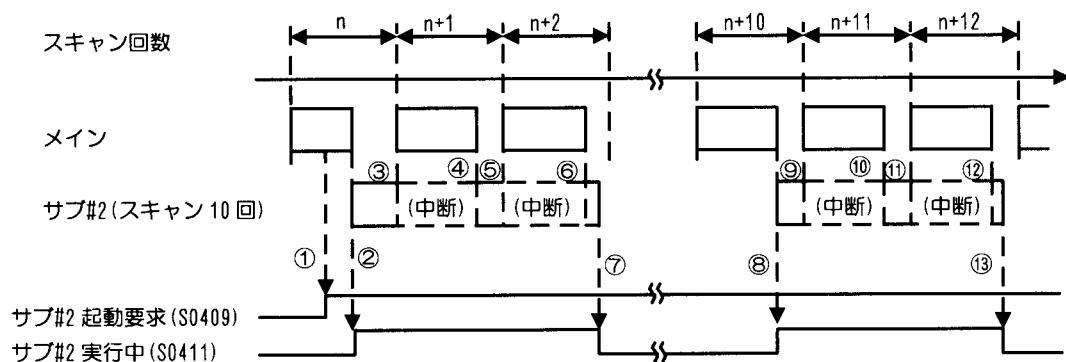


図中のタイミングは次の通りです。

- ① … メインから、サブ#2, サブ#3, サブ#4 に起動要求
- ② … メイン終了後、サブ#2 起動
- ③ … サブ#2 終了、サブ#3 起動
- ④ … 設定時間経過により、サブ#3 実行中断
- ⑤ … メイン終了後、サブ#3 繼続実行開始
- ⑥ … サブ#3 終了、サブ#4 起動
- ⑦ … サブ#4 終了
- ⑧ … サブ#2 起動要求後3スキャン目のメイン終了後にサブ#2 起動
- ⑨ … サブ#2 終了

- ⑩ … サブ#3 起動要求後8スキャン目のメイン終了後にサブ#3 起動
- ⑪ … サブ#3 終了
- ⑫ … サブ#4起動要求後20スキャン目のメイン終了後にサブ#4 起動
- ⑬ … サブ#4 終了

- 定刻スキャンの場合の動作を下図に示します。(基本的な考え方方は同じためサブ#3, サブ#4 は省略)



図中のタイミングは次の通りです。

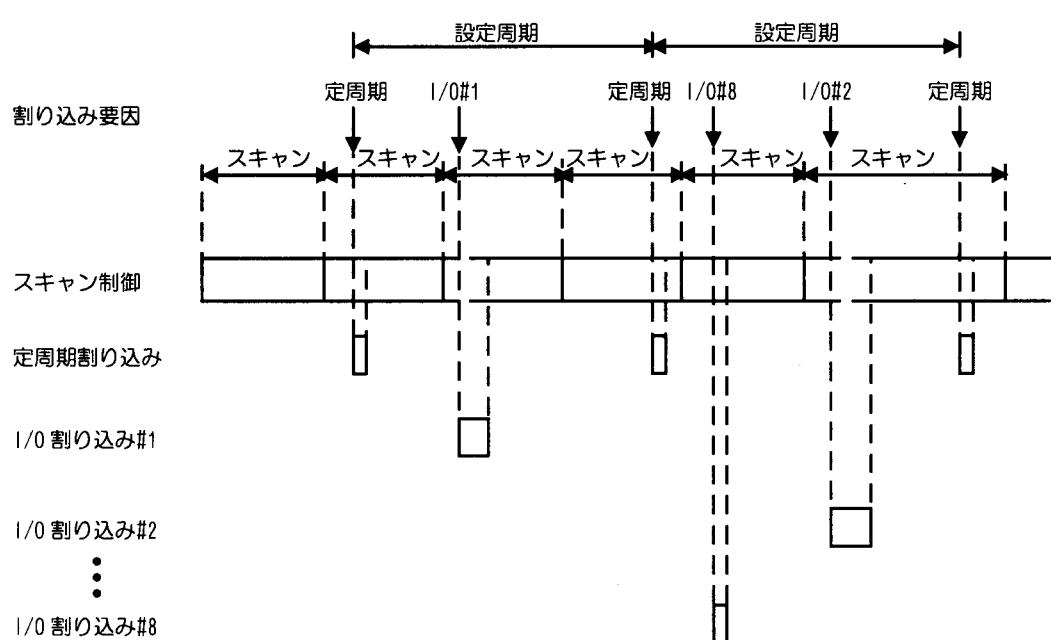
- ① … メインからサブ#2 に起動要求
- ② … メイン終了後、サブ#2 起動
- ③ … 定刻スキャン周期到達、サブ#2 実行中断
- ④ … メイン終了後サブ#2 繼続実行開始
- ⑤ … 定刻スキャン周期到達、サブ#2 実行中断
- ⑥ … メイン終了後サブ#2 繼続実行開始
- ⑦ … サブ#2 終了
- ⑧ … サブ#2 起動要求後 10 スキャン目のメイン終了後にサブ#2 起動
- ⑨ … 定刻スキャン周期到達、サブ#2 実行中断
- ⑩ … メイン終了後サブ#2 繼続実行開始
- ⑪ … 定刻スキャン周期到達、サブ#2 実行中断
- ⑫ … メイン終了後サブ#2 繼続実行開始
- ⑬ … サブ#2 終了

3.3

割り込みプログラム
実行制御

割り込みプログラムは、割り込み要因発生時に他の処理を中断して即座に実行されます。割り込みプログラムとしては、下表のように、システム情報内に設定した周期で起動される定周期割り込みが1本、割り込み機能付きI/Oモジュールからの割り込み信号で起動されるI/O割り込みが#1～#8の8本、登録できます。

割り込みプログラム	動作
定周期割り込み	システム情報内に設定した割り込み周期ごとに実行されます。割り込み周期の設定はT3で2～1000ms、T3Hで1～1000ms(1ms単位)です。
I/O割り込み#1～I/O割り込み#8	状態変化検出付きDC入力(CD332)やパルス入力(P1312)などの割り込み機能付きI/Oモジュールからの割り込み信号発生によって、対応するI/O割り込みプログラムが実行されます。



(1) 割り込みプログラムの実行優先度

複数の割り込み要因が同時に発生したときには、下表の優先度の高い(数値が小さい方が優先度が高い)ものから順に実行されます。また、割り込みプログラム実行中に他の割り込み要因が発生した場合には、割り込み要因は保留され、実行中の割り込みプログラム終了後に、優先度順に実行されます。

割り込みプログラム	優先度	種別内優先度
定周期割り込み	○	—
I/O割り込み#1		○(初期状態)
I/O割り込み#2		1(II)
I/O割り込み#3		2(II)
I/O割り込み#4		3(II)
I/O割り込み#5	1	4(II)
I/O割り込み#6		5(II)
I/O割り込み#7		6(II)
I/O割り込み#8		7(II)

定周期割り込みが最も優先度が高く、次いでI/O割り込みの順となります。I/O割り込みの中の優先度は、CPUに近い割り込み機能付きI/Oモジュールからの割り込みが最も優先度が高くなります。割り込みプログラムとI/Oモジュールの対応は下記(3)を参照してください。

(2) 割り込みの許可/禁止

ユーザプログラムのDI命令(割り込み禁止)及びEI命令(割り込み許可)を実行することによって、割り込み実行の禁止/許可を切り換えることができます。DI命令実行後の割り込み禁止状態に発生した割り込み要因は保留され、EI命令実行によって割り込み許可状態となった瞬間に実行されます。

また、RUNモード遷移時の第1スキャン中は割り込み禁止状態となっており、第2スキャンから割り込み許可となります。

(3) I/O割り込みプログラムの割り付け

割り込み機能付きI/OモジュールとI/O割り込みプログラムの対応は、初期状態では、CPUに近いモジュールから番号の若いプログラム順に割り付けられます。ただしプログラマの割り込み割り付け設定機能によって、割り付けを変更することができます。

割り込み機能付きI/Oモジュールの実装位置には制約はありません。
(第3部2.3.3節参照)

補足

I/O割り込み応答時間(割り込み要因発生から割り込みプログラム起動まで)は、常時割り込み許可かつ他の割り込みプログラム不実行状態の条件で500μs以内です。ただし、浮動小数点演算など処理時間の長い命令を使用している場合には、最悪その命令の処理時間+500μsとなります

4.1

EEPROM サポート

EEPROM 内蔵の CPU モジュール (PU325, PU325H, PU326H) の場合には、ユーザプログラム及びユーザデータの内容を EEPROM に格納し、イニシャルロード機能またはプログラマ操作によって、メインメモリ (RAM) に読み出すことができます。また、ユーザプログラムにて EEPROM 内のデータレジスタ領域の書き込み/読み出しを行うことも可能です。

EEPROM を使用することによって、バッテリレス運転が可能となる他、万一のプログラム破壊時の復旧も容易となります。

EEPROM に対して、次の機能が可能です。

機能	内 容	実行条件
プログラム書き込み	メインメモリ (RAM) のユーザプログラム (システム情報含む) 及び RW, T, C, D の全レジスタの内容を EEPROM に書き込みます。	I/O メモリカード非実装時、HALT モードの状態にて、プログラマから「I/O カード (EEPROM) 書き込み」コマンドを実行したとき
プログラム読み出し	EEPROM の内容をメインメモリ (RAM) のユーザプログラムメモリ、RW, T, C, D の全レジスタに転送します。	I/O メモリカード非実装時、HALT モードかつモードスイッチ P-RUN 以外の状態にて、プログラマから「I/O カード (EEPROM) 読み出し」コマンドを実行したとき
イニシャルロード	EEPROM の内容をメインメモリ (RAM) のユーザプログラムメモリ、データレジスタの先頭 4K ワード (D0000～D4095) に転送します。	システム初期化処理時： I/O メモリカード非実装時、RAM/ROM 切換スイッチ ROM かつモードスイッチが P-RUN 以外の状態にて電源が投入されたとき RUN モード遷移時： I/O メモリカード非実装時、RAM/ROM 切換スイッチ ROM かつモードスイッチ RUN の状態で RUN モードに遷移したとき
EEPROM 内データレジスタ読み出し/書き込み (T3 ではバージョン 1.3 以降)	ユーザプログラムにて、EEPROM 内のデータレジスタとメインメモリ (RAM) の任意のレジスタとの間でデータの転送を行います。	拡張データ転送命令 (XFER) を使用します。

*1) イニシャルロード機能については、2.2 システム初期化処理及び 2.4 スキャン制御を参照して下さい。

*2) EEPROM の繰り返し書き込み回数には、ハードウェアとして 10 万回 (保証値) の制限があります。PC は、EEPROM 書き込み実行回数をカウントしており、10 万回を超過すると EEPROM 警告フラグ (S0007) が ON となります。ただし、拡張データ転送命令 (XFER) による EEPROM 内データレジスタ書き込みに対しては、この書き込み回数チェック機能は無効ですので注意して下さい。

4.2

IC メモリカード
サポート

CPU モジュールに IC メモリカードを装着することによって、EEPROM サポート機能と同様に、ユーザプログラム+ユーザデータのロード/セーブが可能となります。これよりプログラムの一括変更、コピー等が容易に行えます。また、IC メモリカードを拡張メモリとして設定し、サンプリングトレースのトレースバッファ及び拡張ファイルレジスタとして使用することもできます。

IC カードを使用することによって可能となる機能を下表に示します。

使用形態	機能	内 容	実行条件
プログラム 補助メモリ	プログラム書き込み	メインメモリ (RAM) のユーザプログラム及び RW, T, C, D の全レジスタの内容を IC メモリカードに書き込みます。	HALT モードの状態にて、プログラマから「IC カード (EEPROM) 書き込み」コマンドを実行したとき
	プログラム読み出し	IC メモリカードの内容をメインメモリ (RAM) のユーザプログラム、RW, T, C, D の全レジスタに転送します。	HALT モードかつモードスイッチ P-RUN 以外の状態にて、プログラマから「IC カード (EEPROM) 読み出し」コマンドを実行したとき
IC カード 初期化 (T3H のみ)	IC メモリカードの内容を初期化 (Oクリア) します。		HALT モードの状態にて、プログラマから「IC カード初期化」コマンドを実行したとき
イニシャル ロード	IC メモリカードの内容をメインメモリ (RAM) のユーザプログラム、データレジスタ先頭 4K ワード (D0000 ~D4095) に転送します。		システム初期化処理時： RAM/ROM 切換スイッチ ROM かつモードスイッチが P-RUN 以外の状態にて電源が投入されたとき
			RUN モード遷移時： RAM/ROM 切換スイッチ ROM かつモードスイッチ RUN の状態で RUN モードに遷移したとき
拡張メモリ	サンプリング トレースのト レースバッフ ア (T3 ではバ ージョン 1.2 以降)	サンプリングトレース機能の収集データ格納用に使用します。	入出力割り付けにて CPU スロットに MMR と設定されている状態にてサンプリングトレース機能が実行されたとき
	拡張ファイル レジスタ (T3 ではバージョ ン 1.3 以降)	ユーザプログラムにて、120k ワードの拡張ファイルレジスタとして読み出し、書き込みできます。制御パラメータの格納、フィールド収集データの格納など自由に使用できます。	入出力割り付けにて CPU スロットに MMR と設定されている状態にて、拡張データ転送命令 (XFER) によって読み出し/書き込みを行います。

*1) イニシャルロード機能については、2.2 システム初期化処理及び 2.4 スキャン制御を参照して下さい。

*2) サンプリングトレース機能については、5.9 節を、入出力割り付けの MMR 設定については、第3部 4.2 節を参照して下さい。

5.1

概要

RAS とは、Reliability=信頼性、Availability=可用性、Serviceability=保守性を意味し、PC を適用したシステムの信頼性、保守性を高め、システムの稼動をサポートするために、PC が備えている機能を総称して RAS 機能と呼びます。

この章では、PC が備えている自己診断機能、メンテナンス機能、デバッグ機能、及びユーザが PC を使用してシステムのチェックを行うためのシステム診断機能について説明します。

5.2

自己診断

PC が PC 自身の健全性をチェックし、異常動作を起こさないようにするための自己診断の内容、診断実施のタイミング、及び異常検出時の動作について以下に示します。

システム設計にあたっては、PC が異常を検出した場合のシステム動作の安全性(フェイルセーフ)、及びシステム動作のバックアップについて、十分に検討して下さい。

なお、以下の説明中、異常登録とは異常内容と発生時刻をイベント履歴テーブルに格納することを、エラーダウンとは出力を全て OFF に落とし ERROR モードに遷移することを、またアラームとは異常登録、特殊リレーのセットを行い、動作は継続することを意味します。

(1) システム初期化処理時(電源投入時)診断項目

診断項目	診断内容	異常検出時の動作
システム ROM BOC チェック	システム ROM の正当性を BOC にてチェックします。	異常登録を行い、FAULT, I/O LED を点滅させます。(プログラマ交信不能)
システム RAM チェック	システム RAM のリード/ライトチェックを行います。	異常登録を行い、FAULT LED を点滅させます。(プログラマ交信不能)
周辺 LSI チェック	周辺 LSI が正常に初期化できたかチェックします。(リードバックチェック)	異常登録を行い、FAULT LED を点滅させ I/O LED を点灯させます。(プログラマ交信不能)
LP チェック	LP(シーケンス演算プロセッサ)が正常に初期化できたかチェックします。	異常登録を行い、ERROR モードで立ち上がります。(エラーリセットコマンド無効)
ユーザプログラムメモリチェック	ユーザプログラムメモリの内容の正当性を BOC にてチェックします。(周辺メモリが存在する場合はイニシャルロード後にチェックします)	異常登録を行い、ERROR モードで立ち上がります。
ユーザデータメモリチェック	ユーザデータメモリのリード/ライトチェックを行います。	異常登録を行い、ERROR モードで立ち上がります。(エラーリセットコマンド無効)
周辺メモリチェック	イニシャルロードの対象となる周辺メモリ(I ² C メモリカードまたは EEPROM)の正当性を BOC によりチェックします。	異常登録を行い、ERROR モードで立ち上がります。

カレンダ LSI チェック	カレンダ LSI からの読み出しデータ(日付、時刻)の有効性チェックを行い、特殊レジスタにデータをセットします。	アラーム。カレンダ再設定まで日付・時刻データ(特殊レジスタ内)は HFF とします。
バッテリチェック	メモリバックアップ用のバッテリの電圧をチェックします。	アラーム。ユーザプログラムメモリの BCC が正常であれば正常に立ち上がります。(ただし停電保持指定されたユーザデータの内容は保証外)

(2) RUN 起動時診断項目

診断項目	診断内容	異常検出時の動作
I/O 照合チェック	I/O 割り付け情報と I/O モジュールの実装状態を照合し、一致していることをチェックします。	異常登録し、エラーダウン。ただし、プログラマの運転コマンドによる起動時は、メッセージ表示のみで HALT モードのまま、異常登録もしません。
I/O バスチェック	I/O バスが正常であることをチェックします。	異常登録し、エラーダウン。ただし、プログラマの運転コマンドによる起動時は、メッセージ表示のみで HALT モードのまま、異常登録もしません。
拡張ユニット 電源チェック	拡張ユニットの電源が正常であることをチェックします。	異常登録し、エラーダウン。ただし、プログラマの運転コマンドによる起動時は、メッセージ表示のみで HALT モードのまま、異常登録もしません。
I/O 応答チェック	I/O モジュールアクセス時に応答が規定時間内に返ってくることをチェックします。	異常登録し、エラーダウン。ただし、プログラマの運転コマンドによる起動時は、メッセージ表示のみで HALT モードのまま、異常登録もしません。
プログラム チェック	ユーザプログラムの文法チェックを行います。	異常登録し、エラーダウン。ただし、プログラマの運転コマンドによる起動時は、メッセージ表示のみで HALT モードのまま、異常登録もしません。

(3) スキャン実行中診断項目

診断項目	診断内容	異常検出時の動作
I/O バスチェック	I/O バスが正常であることをチェックします。(一括入出力時)	異常登録後エラーダウン。 (ただし、規定個数のリトライ処理によって回復した場合には、登録のみでエラーダウンはしません)
拡張ユニット 電源チェック	拡張ユニットの電源が正常であることをチェックします。(一括入力時)	異常登録後エラーダウン。 (ただし、規定個数のリトライ処理によって回復した場合には、登録のみでエラーダウンはしません)
I/O 応答チェック	I/O モジュールアクセス時に応答が規定時間内に返ってくることをチェックします。(一括入力時及び直接入出力命令実行時)	異常登録後エラーダウン。 (ただし、規定個数のリトライ処理によって回復した場合には、登録のみでエラーダウンはしません)

I/Oバス/parity チェック	I/O モジュールアクセス時のバスパリティをチェックします。(一括入出力及び直接入出力命令実行時)	異常登録後エラーダウン。 (ただし、規定回数のリトライ処理によって回復した場合には、登録のみでエラーダウンはしません)
LP機能チェック	LP(シーケンス演算プロセッサ)に所定のテストプログラムを実行させ、正常な結果が得られるかをチェックします。(ユーザプログラム実行時)	異常登録後エラーダウン。 (ただし、規定回数のリトライ処理によって回復した場合には、登録のみでエラーダウンはしません)
LPイリーガル命令検出 チェック	LP(シーケンス演算プロセッサ)にて不正命令検出の有無をチェックします。(ユーザプログラム実行時)	異常登録後エラーダウン
スキャンタイムオーバ チェック	スキャン周期が設定値(200ms)を超えていないかチェックします。 ただし設定値はユーザ命令(WDT)にて変更できます。(ユーザプログラム実行時)	異常登録後エラーダウン

(4) 常時診断項目(動作モードにかかわらず定期的に実行)

診断項目	診断内容	異常検出時の動作
システム ROM BCC チェック	システム ROM の正当性を BCC にてチェックします。	異常登録後エラーダウン(エラーリセットコマンド無効)
システム RAM チェック	システム RAM のリード/ライトチェックを行います。	異常登録後エラーダウン(エラーリセットコマンド無効)
周辺 LSI チェック	周辺 LSI の設定値が正常であることをチェックします。	異常登録後エラーダウン(エラーリセットコマンド無効)
ウォッチドッグ タイムチェック	ウォッチドッグタイマにてシステムの暴走をチェックします。(設定350ms)	システムリセット後、異常登録してERRORモードに遷移します。
ユーザメモリチェック	ユーザメモリ(RAM)のリード/ライトチェックを行います。	異常登録後エラーダウン(リトライあり)
LP チェック	LP(シーケンス演算プロセッサ)のリード/ライトチェックを行います。	異常登録後エラーダウン
バッテリチェック	メモリバックアップ用のバッテリの電圧をチェックします。	アラーム
カレンダ LSI チェック	300ms 毎にカレンダ LSI から日付・時刻データを読み出し、有効性チェックの上、特殊レジスタにデータをセットします。	アラーム カレンダ再設定まで、日付・時刻データは HFF とします。

- *) 異常発生時にイベント履歴テーブルに登録される内容、及びセットされる特殊リレーのアドレスについては、別冊の「T3/T3H 本体ハードウェア説明書」のトラブルシューティングの章で説明しますので併せてご覧下さい。

5.3

イベント履歴登録
機能

PC が自己診断にて異常を検出した場合、異常内容と発生時刻がイベント履歴テーブルに登録されます(異常の他に、電源 ON/OFF 時刻も登録されます)。イベント履歴テーブルには、最新の 30 個までの発生内容が登録され、新しい内容が登録されるとそれまでの登録内容は順に繰り下がり、最も古い内容は捨てられます。

イベント履歴テーブルは、プログラマを接続することによって、下図の例のように内容を表示させ確認することができますので、メンテナンス情報として利用して下さい。なお、イベント履歴テーブルの内容は、プログラマからイベント履歴クリアコマンド、またはメモリクリアコマンドを実行するまで保持されます。

< イベント履歴 >						
日付	時刻	発生事象	回数	情報1	情報2	情報3
1. 98-07-23	08:23:15	システム電源パワーオン	1			INIT.
2. 98-07-23	08:22:55	システム電源パワーフェイル	1			HALT.
3. 98-07-23	08:17:39	システム電源パワーオン	1			INIT.
4. 98-07-22	17:10:49	システム電源パワーフェイル	1			HALT.
5. 98-07-22	17:01:45	システム電源パワーオン	1			INIT.
6. 98-07-22	17:00:20	システム電源パワーフェイル	1			HALT.
7.						
8.						
9.						
10.						
11.						
12.						
13.						
14.						
15.						

FC HALT PROG バック
負切換 PC制御 戻出

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10

表示画面中の各項目の意味は次の通りです。

(1) 番号(1~30)

発生の順序を示します。1番が最新の発生内容となります。

5

(2) 日付(年-月-日)

発生の日付を示します。もしカレンダ LSI が異常だった場合には、??-??-??と表示されます。

(3) 時刻(時:分:秒)

発生の時刻を示します。もしカレンダ LSI が異常だった場合には、?:?:?:?と表示されます。

(4) 発生事象

検出された異常の内容を示します。(システム電源パワーオンは電源 ON を、システム電源パワーフェイルは電源 OFF を示します)

(5) 回数

異常検出回数を示します。例えば、ある処理中に異常を検出し、再試行(リトライ)を3回繰り返しても異常内容が変わらずエラーダウンした場合には、回数が4と表示され、運転モードの項目に**ダウン**と表示されます。

(6) 情報1, 情報2, 情報3

異常発生内容の補足情報を示します。例えば、I/O 異常の場合には、異常発生した I/O モジュールの位置(ユニット No, スロット No)と読み書きしたレジスタアドレスなどが表示されます。

(7) 運転モード

異常を検出したときの本体モードを示します。また、この異常検出によってエラーダウンに到った場合には、**ダウン**と表示されます。

なお、モード表示中、INIT. とは電源投入後のシステム初期化処理を示します。

- *) 異常検出時の表示内容(発生事象, 情報1, 情報2, 情報3)及びこの意味と処理方法については、別冊の「T3/T3H 本体ハードウェア説明書」のトラブルシューティングの項をご覧ください。

5.4 メモリプロテクト機能

CPU モジュール上のモードスイッチは、HALT/RUN/P-RUN の3ポジションキースイッチを使用しており、P-RUN 位置はメモリプロテクトの機能を持っていきます。

モードスイッチを P-RUN の位置にしておくことによって、プログラマから下記操作を行っても実行されず、プログラマの画面上にプロテクト状態であることがメッセージ表示されます。

P-RUN 位置(メモリプロテクト状態)にすることによって禁止される操作

- (1) メモリクリア
- (2) 入出力自動割り付け
- (3) 入出力割り付け情報書き込み
- (4) システム情報書き込み
- (5) プログラム編集(オンライン変更含む)
- (6) フロッピー等から PC 本体へのプログラムダウンロード
- (7) IC メモリカード (EEPROM) 読み出し(イニシャルロード含む)
- (8) データレジスタの先頭 4K ワード (D0000～D4095) へのデータ書き込み

メモリプロテクト機能を使用することにより、プログラマの誤操作によるプログラム破壊を防ぐことができます。

補足

T3 ではモードスイッチが P-RUN のときには、下記命令語によってデータレジスタの先頭 4K ワード (D0000～D4095) にデータを書き込むことはできませんので注意して下さい。但し他の命令語 (MOV 等) での書き込みは可能です。

- カレンダ演算 (CLDS, FUN 155)
- 拡張データ転送 (XFER, FUN 236)
- 特殊モジュールデータ入力 (READ, FUN 237)

また、ユーザデータ初期化処理においても、D0000～D4095 はクリアされず、以前の内容が残りますので注意して下さい。

5.5

瞬停検出機能

PC の電源に瞬停が発生した場合の動作を決定する機能として、瞬停時間設定機能と瞬停継続機能を備えています。

5.5.1

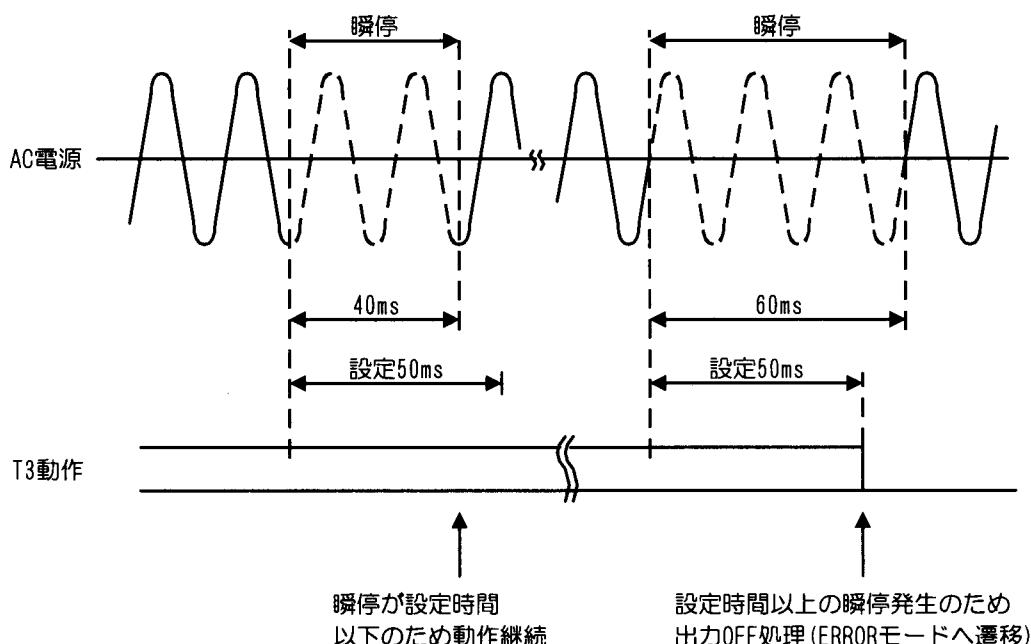
瞬停時間設定機能

通常 PC は、電源に停電が発生すると、内部 5V 電源の充電状態と負荷状態に応じた時間だけ動作を継続し、その後電源 OFF 状態(通常は全出力 OFF)に移行します。この動作継続時間は、内部 5V 電源の状態(電源電圧、I/O モジュールの数、種類及び入出力 ON/OFF 状態など)によって左右されるため、システム構成及び停電発生の状況によってばらつくことになります。制御対象によっては、このばらつきが問題となる場合があります。

PC は、電源条件(電源電圧、内部 5V 負荷)によって決まる最大動作継続時間の範囲内で許容瞬停時間を設定できる機能を備えています。この機能を瞬停時間設定機能と呼びます。(T3 ではバージョン 1.2 以降)

許容瞬停時間の設定は、プログラマからユーザ診断項目内の瞬停検出時間に値を設定すること、または特殊レジスタ(SW045)に直接値を設定することにより行います。(設定値:10~255ms, 1ms 単位)

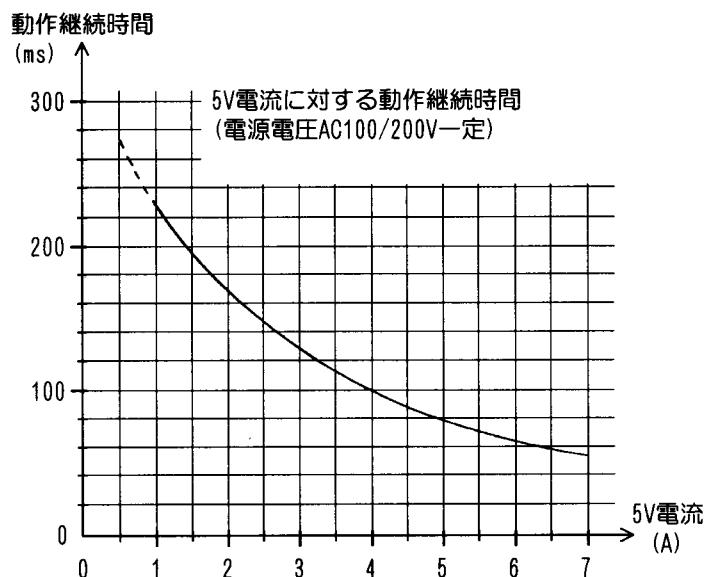
瞬停時間設定機能の動作例を下図に示します。



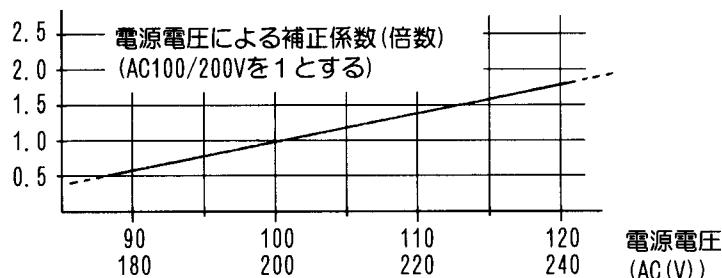
補足

瞬停検出時間が設定されていない場合、及び設定時間が最大動作継続時間(電源条件によって決まる)を超える場合には、この機能は働きません。次ページに最大動作継続時間の算出方法を示します。

最大動作継続時間は、電源電圧及び使用しているモジュールで消費する内部 5V 電流によって決まります。内部 5V 消費電流と電源電圧から、下図によって最大動作継続時間を求め、瞬停検出時間はこの値以下となるように設定して下さい。(各モジュールの内部 5V 消費電流については、別冊「T3/T3H 本体ハードウェア説明書」及び使用特殊モジュールの関連説明書をご覧下さい)



補正係数



補足

- (1) 停電時間が設定時間以上となったとき、PC は即座に全出力を OFF します。(内部的には ERROR モードに遷移)
この状態からすぐに(内部 5V が低下する前に)復電したときには ERROR モードが継続し、自動運転には戻りません。
- (2) 瞬停動作中に拡張ユニット側の電源が内部 5V 電圧低下を起こさないように(拡張側電源の方が負荷が重いことがないように)注意して下さい。

5.5.2

瞬停継続機能

制御対象によっては、瞬停によってPC停止→PC再始動となる場合、再始動時の状態に注意が必要な場合があります。

PCでは、サブプログラムの特殊モード(3.2節参照)を使用することによって瞬停再始動時の状態をユーザが決定することができます。

サブプログラム#2が特殊モード(S0403=1)かつ瞬停継続モード(S0400=1)の設定のときに、停電時間2秒以内で復電した場合、サブプログラム#2はメインプログラムの実行前に1回だけ実行されます。

従って、入力状態(制御対象の状態)によって、レジスタ/デバイスの初期状態設定を行うように、サブプログラム#2を構成しておけば、下表のように再始動時の状態を選択/設定することができます。

停電時間	再始動時の状態	実現方法
2秒超過	初期化後再始動 (通常の起動状態)	—
2秒以内	初期化後再始動 (通常の起動状態)	瞬停継続モードの設定を行わない(S0400=0)
	レジスタ/デバイスにあらかじめ決められた状態を設定してから再始動	サブプログラム#2で所定の設定を行う
	入力状態に応じた初期設定をしてから再始動	サブプログラム#2の内容による
	前回電源OFF時の状態のまま再始動	サブプログラム#2で何も初期設定を行わない

補足

- (1) 停電時間が2秒を超える場合には、レジスタ/デバイス内容は、停電保持指定の領域、及びフォース指定されたもの以外は0クリア(ユーザデータの初期化)されます。
- (2) 瞬停継続モードの設定(S0400=1)で停電時間が2秒以内の場合には、ユーザデータの初期化は行われませんので、サブプログラム#2の内容によって、上記のような動作選択が可能となります。
- (3) 瞬停継続モードの設定は、特殊リレー(S0400)に直接データを設定しておく以外に、プログラマのユーザ診断モードによっても行うことができます。

5.6 I/O エラーマップ 作成機能

I/O モジュールに、内蔵ヒューズ溶断等の異常が発生した場合、どのモジュールに異常が発生したのかを検出する機能があります。この機能を I/O エラーマップ作成機能と呼びます。なお、ここで対象とするのは、PC の CPU がエラーダウンとはならない、モジュール異常発生時です。

I/O エラーマップ作成機能を有効にするためには、あらかじめ特殊リレー S0150 を ON にします。これによって PC の CPU は、実装された各 I/O モジュールの状態をチェックし、異常を検出した場合には I/O 異常フラグ (S0009) を ON にセットすると共に、異常モジュールの実装位置を特殊レジスタ (SW046～SW049) に登録します。

異常検出の対象となるモジュールと検出内容は下表の通りです。

形 式	概 要	検出内容
D0333	16 点 DC 出力	ヒューズ溶断、外部電源異常
D0334/D0344	32 点 DC 出力	ヒューズ溶断、外部電源異常
AC363	16 点 AC 出力	ヒューズ溶断、外部電源異常
AC364	32 点 AC 出力	ヒューズ溶断、外部電源異常
R0364	32 点接点出力	外部電源異常
R0363S	16 点独立接点出力	外部電源異常
AD368	8 チャネルアナログ入力	モジュール異常
DA364/374	4 チャネルアナログ出力	モジュール異常、外部電源異常
MC352	2 軸位置決め	モジュール異常

— 補足 —

- (1) I/O エラーマップ作成機能を有効にしたときには、スキャンのオーバーヘッドが多少長くなります。
- (2) I/O エラーマップ作成機能は T3 ではバージョン 1.3 以降で可能です。

5.7

I/O 活線着脱機能

運転中に I/O モジュールの単体故障が発生した場合、システムの運転を継続しながら故障モジュールを交換することができます。この機能を I/O 活線着脱機能と呼びます。

1台の PC で複数ステーション/ブロックの制御を行っているときなどに、モジュール故障が発生したステーション/ブロックのみ運転を停止して、モジュール交換を行う場合などに有効です。

I/O 活線着脱を行う場合には、まずプログラマの I/O 着脱指定機能によって交換したいモジュール位置に着脱指定をかけます。

モジュールに着脱指定がかけられると、PC の CPU は、着脱指定のかけられたモジュールを一括入出力及び直接入出力の対象から外します。

この後、モジュール交換を行い、最後にプログラマから着脱指定を解除します。以上の操作によって、I/O 活線着脱が実現できます。

I/O 活線着脱が有効なモジュール単体故障としては以下が考えられます。

- 入力モジュールで特定の信号が読み込めなくなった（入力フォトカプラ故障など）
- 出力モジュールで特定の出力が出なくなった（出力素子故障など）
- 出力モジュールのヒューズが溶断した

I/O 活線着脱が可能なモジュールは下表の通りです。

形 式	概 要	形 式	概 要
DI334/334H	32 点 DC 入力	D0335	64 点 DC 出力
DI335/335H	64 点 DC 入力	AC363	16 点 AC 出力
IN354/364	32 点 AC 入力	AC364	32 点 AC 出力
D0333	16 点 DC 出力	R0364	32 点 AC 出力
D0334	32 点 DC 出力	R0363S	16 点独立接点出力

上記以外のモジュールに対して活線着脱を行わないで下さい。
活線着脱を行った場合、故障及び誤動作の可能性があります。

⚠ 注意

I/O モジュールの交換は必ず電源を切った状態で
行って下さい。
感電、誤動作、故障の原因となることがあります。
やむを得ず I/O 活線着脱機能を使用する場合は、作
業の安全に十分注意して下さい。

補足

- (1) 通常、モジュール交換は可能な限り電源を切った状態で行って下さい。
やむを得ず活線着脱を行う場合には、作業の安全に十分注意して下さい。
- (2) モジュールの交換は、同じモジュール種別(X 2W 等)でのみ可能です。
例えば、32 点入力を 64 点入力に変えることはできません。

5.8

実行状態モニタ機能

PC のスキャン制御実行状態をモニタ (監視) するために、本体がサポートする機能として以下の機能があります。(操作については、別冊のプログラマ説明書をご覧下さい)

(1) 実行時間計測機能

次の実行時間を計測します。このデータはプログラマで読み出して確認することができます。

- スキャン周期 … 現在値、最大値、最小値(1ms 単位)
- メインプログラム実行時間 … 現在値、最大値、最小値(1ms 単位)
- サブプログラム実行時間(サブ#1～#4) … 現在値、最大値、最小値(1ms 単位)
- 定周期割り込み実行時間 … 最新値、最大値、最小値(0.1ms 単位)
- I/O 割り込み実行時間(#1～#8) … 最新値、最大値、最小値(0.1ms 単位)

— 補足 —

- (1) スキャン周期は、スキャンのオーバヘッド及びスキャン中に発生した割り込みも全て含んだ値です。
- (2) メインプログラム及びサブプログラム実行時間は、割り込み発生によって中断した時間は除きます。

(2) オンライントレース機能

プログラマでモニタ中の回路範囲について、命令実行中の状態をトレースし、プログラマ画面上に実行状態を表示(回路活線表示、レジスタ値表示)します。

1スキャン終了後のデータではなく、命令実行時点のデータを表示しますので、プログラムデバックにも有効です。

(3) ステータスマニタ機能

プログラマのオンライントレース表示画面にて、補助表示機能によって指定された、8点までのデバイス/レジスタの状態を、上記オンライントレース実行直後の時点で収集し、画面上に表示します。

5.9

サンプリング
トレース機能

プログラマから設定されたサンプリング条件が成立したときの指定デバイス/レジスタのデータを収集し、時系列的にサンプリングバッファに格納します。

サンプリングの条件判定とデータ収集は各スキャンの終了時に行います。

サンプリングしたデータはプログラマに読み出して、トレンドグラフ(レジスタ)またはタイムチャート(デバイス)の形で表示することができます。(操作については別冊のプログラマ説明書をご覧下さい)

サンプリングバッファ サンプリングデータの格納先は、ICメモリカードまたはPC本体のファイルレジスタ(F)となります。

① ICメモリカードをサンプリングバッファとする場合

ICメモリカードをCPUモジュールに装着し、入出力割り付けにてCPUスロットにMMRの設定を行います。これによりICメモリカードがサンプリングバッファとなります。サンプリングバッファ容量は8Kワード固定です。

② ファイルレジスタをサンプリングバッファとする場合

入出力割り付けにてCPUスロットにMMRの設定がされていないとき、PC本体のファイルレジスタがサンプリングバッファとなります。

サンプリングバッファの容量は、システム情報にて設定します。設定は1~8Kワード(1Kワード単位)が可能で、設定された容量分だけ、ファイルレジスタの大きいアドレス側からサンプリングバッファとして使用されます。(この領域をユーザプログラムで使用している場合には結果が異常となりますので、領域が重ならないように注意して下さい)

[T3は8KWの後方から、T3Hは32KWの後方から設定容量分確保します]

サンプリング対象 サンプリングする対象(レジスタ/デバイス)の組み合わせとしては、

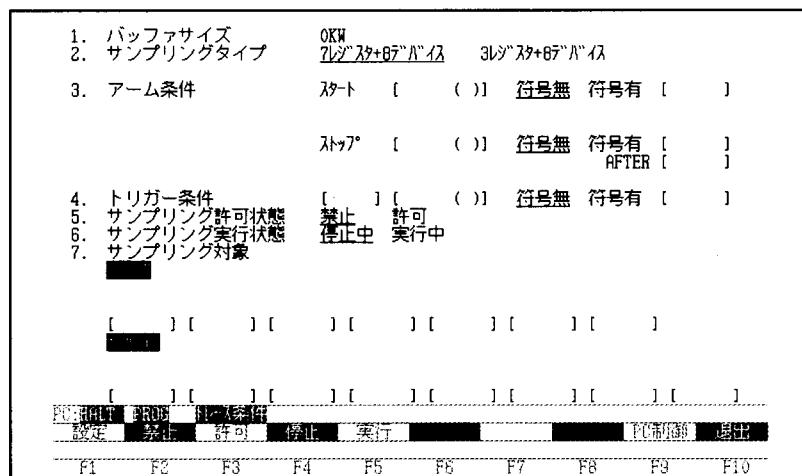
- ① 3レジスタ+8デバイス
- ② 7レジスタ+8デバイス

の選択が可能です。①の場合には、256回/1Kワード(最大2048回サンプル)、②の場合には、128回/1Kワード(最大1024回サンプル)のデータ蓄積が可能です。

サンプリング対象の設定は、プログラマのサンプリングトレース条件・対象画面にて行います。設定は、PC本体がHALTモードのとき、またはRUNモードでサンプリング禁止状態のときに可能です。

- サンプリング条件** サンプリングトレースを実行するための条件として、アーム条件及びトリガ条件を設定します。
- アーム条件にはスタート条件とストップ条件があり、アームスタート条件成立からアームストップ条件成立までの間、データ収集が可能となります。ただしアームストップ条件に AFTER 回数が設定されている場合には、アームストップ条件成立から AFTER 回数分のスキャンだけ、データ収集可能期間が継続されます。
- また、トリガ条件には実際にデータを収集するタイミング条件を設定します。つまり、アーム条件が成立していてかつトリガ条件が成立した瞬間に、データが収集されます。
- なお、サンプリングトレース実行の前提条件として、サンプリング許可になっていることが必要です。

サンプリング条件の設定は、サンプリングトレース条件・対象画面(下記)にて行います。設定は、PC 本体が HALT モードのとき、または RUN モードでサンプリング禁止状態のときに可能です。



各条件の設定方法は以下の通りです。

アームスタート条件:

スタート [()] 符号無 符号有 []

対象がレジスタであって、判定条件が=のときの比較値

判定条件:

対象がレジスタのとき、	対象がデバイスのとき、
ブランク …… =	1 …… 立ち上がり
1 …… 増加	2 …… 立ち下がり
2 …… 減少	3 …… 変化
3 …… 変化	

条件判定の対象、レジスタまたはデバイス

アームストップ条件 :

ストップ [()] 符号無 符号有 []

AFTER []

ストップ条件成立後、設定したスキャン回数(プログラムスキャン回数)だけアーム条件を継続

ブランク …… 指定なし
1~65535 …… 指定スキャン回数

*) 他の設定はアームスタート条件と同様

トリガ条件:

[] [()] 符号無 符号有 []

成立回数:

設定した回数だけトリガ条件が成立したときにデータ収集を実行

ブランク …… 設定なし(回数=1)
1~65535 …… 設定回数

*1) 他の設定はアームスタート条件と同様

*2) トリガ条件設定なしのときは毎スキャン実行

補足

判定条件の成立は、前回スキャン時の値からの増加、減少、変化によって判断されます。(レジスタの=条件時を除く)

5. RAS機能

第2部 本体機能説明

実行例 サンプリング条件・対象設定例

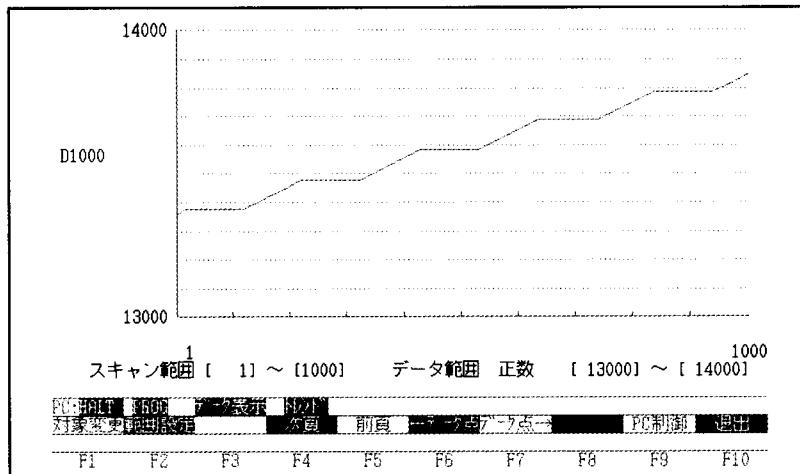
1. バッファサイズ	OKW									
2. サンプリングタイプ	レジスト+ゼータバイト	<u>レジスト+ゼータバイト</u>								
3. アーム条件	スタート	[D2001(3)]	符号無	符号有	[]					
	ストップ	[R0100(1)]	符号無	符号有	[]	AFTER	[]	10		
4. トリガー条件	[]	[]	()	符号無	符号有	[]				
5. サンプリング許可状態	禁止	許可								
6. サンプリング実行状態	停止中	実行中								
7. サンプリング対象	[]									
[YW008] [D1000] [D2001]										
[S0041] [Y0104] [Y0105] [Y0106] [R0100] [] [] []										
PC HALT	PROG	フレーム待								
設定	禁止	許可	停止	実行	[]				PG 制御	退出
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	

上記画面の設定の場合

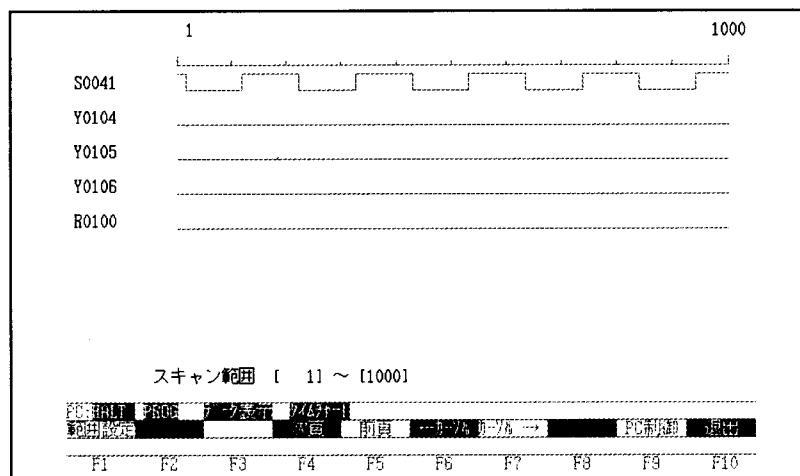
D2001 の値が変化してから、R0100 が OFF→ON に立ち上がって 10 スキャン後までの間、毎スキャン、YW008、D1000、D2001、S0041、Y0104、Y0105、Y0106、R0100 のデータを収集する。
ということになります。

結果画面の表示例 1 (収集データ表示)

結果画面の表示例2(レジスタのトレンドグラフ表示)



結果画面の表示例3(デバイスのタイムチャート表示)



なお、アーム条件の設定を行わない場合には、サンプリング条件・対象画面にて、手動でサンプリングトレースの実行(F5)、停止(F4)を行うこともできます。

補足

サンプリングトレース機能はT3ではバージョン1.2以降で可能です。

サンプリングコピー
(T3Hのみ) T3Hにおいては、ICメモリカードにサンプリング条件とサンプリングデータをコピーすることができます。

これにより、T3Hで設定/サンプリングを行った情報が他のT3Hでも利用可能となります。

ICメモリカードにサンプリング条件とサンプリングデータをコピーする場合は特殊リレーS0620を1(ON)にセットしてサンプリングトレースを行います。S0620をONにしてコピーしたサンプリングデータは他のT3Hでも読み出すことができます。

T3Hでのサンプリングトレース時のICメモリカードコピーのルールは次のようになっています。

(I/O割付のCPU欄にMMRを設定した場合)

- サンプリング条件設定画面を表示した際に、ICメモリカードにすでにサンプリング条件が入っている場合は、その条件がT3H本体に転送されます。
ICメモリカードが初期化状態の場合は、T3H本体に記憶されている条件が表示されます。
- サンプリング条件を設定し、書き込みを実行すると、T3H本体に記憶される条件データが、そのままICメモリカードにコピーされます。
- 特殊リレーS0620をOFFでサンプリングトレースを実行した場合は、サンプリングデータはICメモリカードを使用して行われますが、記憶(コピー)はされません。
(T3と同様のサンプリングトレース=他のT3Hからサンプリングデータの読み出し不可)
- 特殊リレーS0620をONにしてサンプリングトレースを実行した場合は、サンプリングデータはICメモリカードに記憶されます。
他のT3Hでもこのサンプリングデータを読み出すことができます。

(I/O割付のCPU欄にMMRを設定しない場合)

- MMRを設定せずにT3H本体のFレジスタでサンプリングしたデータをICメモリカードにコピーする場合は、サンプリングトレース完了後、特殊リレーS0620をONにし、サンプリング条件設定画面を表示します。
この時にT3H本体のサンプリング条件とサンプリングデータがICメモリカードにコピーされ、S0620は自動的にOFFになります。
他のT3HでMMR設定を行うことで、このサンプリング条件とサンプリングデータを読み出すことができます。

5.10

ステータスラッチ
機能

プログラマにて設定されたラッチ条件成立時、またはラッチ命令(STLS)実行時の、指定デバイス/レジスタデータをラッチデータ格納エリアに一括転送します。最大32個のデバイス/レジスタの指定が可能です。

ラッチ条件の判定とデータ収集はスキャン終了後に行います。

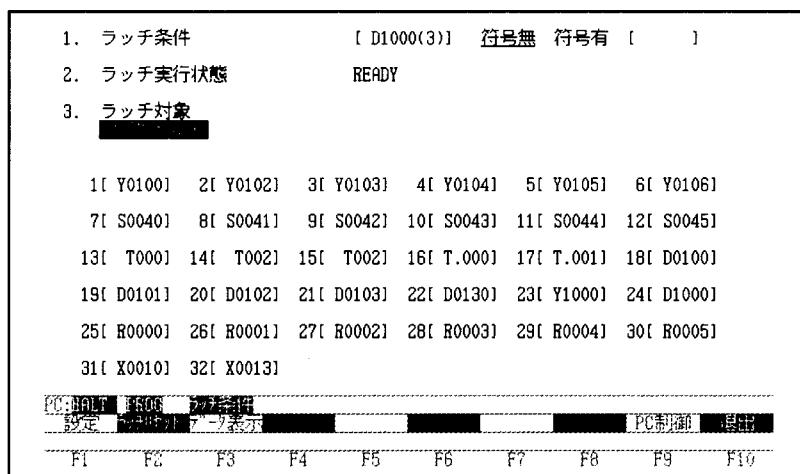
ただしラッチ命令の場合は、命令実行時にデータの収集を行います。

ラッヂしたデータはプログラマに読み出して表示することができます。

また、ラッヂ状態はプログラマのラッヂリセットコマンド、またはラッヂリセット命令(STLR)の実行によって解除されます。

(操作については別冊のプログラマ説明書をご覧下さい)

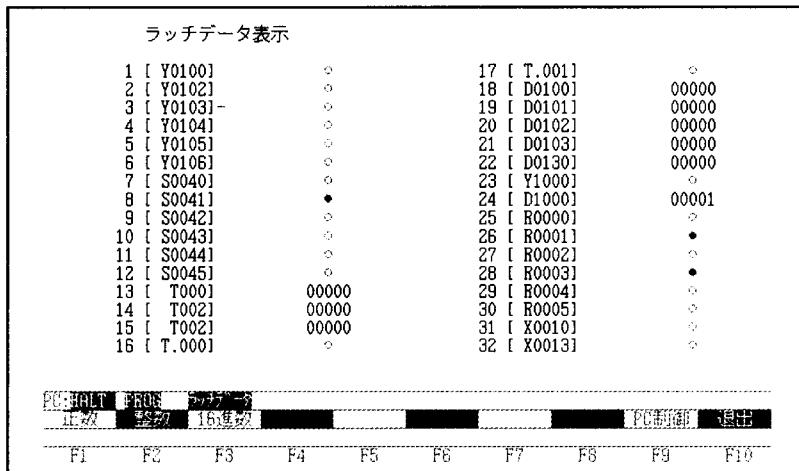
下図にラッヂ条件及びラッヂ対象の設定画面例を示します。



ラッヂ条件の設定方法は、サンプリングトレース機能のアーム条件の設定方法と同じです。(5.9節参照)

この例の場合には、R0100 が OFF→ON に立ち上がったとき、指定した 32 個のデバイス/レジスタのデータをラッヂデータ格納エリアに一括転送することになります。

ラッチデータの表示画面例を下図に示します。



補足

ステータスラッチ機能はT3ではバージョン1.2以降で可能です。

5.11 デバッグ支援機能

5.11.1 フォース機能

ユーザプログラムのデバッグを行うために、本体がサポートする機能を以下に示します。（操作については、別冊のプログラマ説明書をご覧下さい）

フォース機能には、入力フォースとコイルフォースの2つの機能があります。入力フォース指定されたレジスタ/デバイスについては、一括入力処理においてデータの更新を行いません。入力フォースが可能なレジスタ/デバイスとしては、入力レジスタ/デバイス(XW/X)、受信エリアのリンクレジスタ/リレー(W/Z)、及び受信エリアのリンクリレー(L)があります。

一方、コイルフォース指定されたコイル命令については、命令実行時に処理を行いませんので、コイルのデバイスは回路の実行状態にかかわらず、以前の状態を保持します。コイルフォースが可能なデバイスは、出力デバイス(Y)、補助リレー(R)、送信エリアのリンクレジスタリレー(Z)、及び送信エリアのリンクリレー(L)です。

入力フォース/コイルフォース機能とデータ設定機能を併用することによって、模擬入力、模擬出力として利用することができます。

フォース指定されたデバイスは、プログラマ(T-PDS)では、以下の様な表示となります。

- (1) 一部デバイスがフォース指定有りの場合、… デバイス種別のみが小文字表示されます。

X/YW	→	x/yW
LW	→	lW
ZW	→	zW
W	→	w

- (2) 全デバイスにフォース指定有りの場合、… デバイス種別+Wが小文字表示されます。

X/YW	→	x/yw
LW	→	lw
ZW	→	zw
W	→	w

5.11.2

オンライン プログラム 変更機能

オンライン中(RUN中)にユーザプログラムの変更を行う機能です。変更処理は1スキャン終了後に行いますので、処理の間スキャン周期は伸びます。

オンラインプログラム変更には以下の制約があります。

- 実行制御にかかる命令(下記)の個数や順序が変わらるような変更は行えません。
END, MCS, MCR, JCS, JCR, JUMP, LBL, FOR, NEXT, CALL, SUBR, RET, IRET
- SFCプログラム部分については、SFC構造の変更はできませんが、ステップや遷移に対応する詳細部(ラダー図)部分は変更できます。

また、オンラインプログラム変更機能の1つとして、定数変更機能があります。これはタイマ、カウンタ命令の定数(設定値)、及びファンクション命令に使用している定数を直接オンライン(RUN中)で変更する機能です。

タイマ、カウンタの設定値については、メモリプロテクト状態(P-RUN時)でも変更が可能です。

△注意

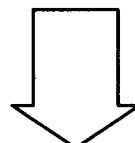
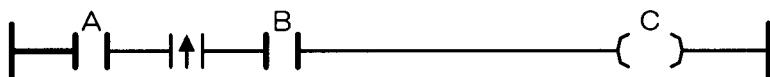
運転中のプログラム変更、強制出力、RUN(運転)、HALT(停止)などの操作は十分安全を確認して行って下さい。

操作ミスや安全確認の怠りにより機械の破損や事故の恐れがあります。

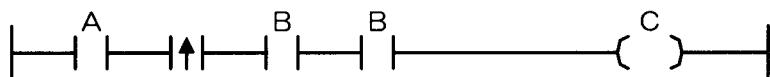
補足

1. オンラインプログラム変更機能はT3ではバージョン1.2以降で可能です。
2. パルス実行機能命令を含む回路にRUN中オンライン書き込みを行うと、その瞬間にパルス命令が1度だけ実行されます。

例)



A, Bが閉、CがOFFの状態で回路を次のように
変更し、RUN中オンライン書き込みを行う。



RUN中オンライン書き込みを行うと、コイルCがその瞬間(1ス
キャン)ONになります。

よって回路の一部を変更しただけでも予想外の動作をすることがありますので、パルス実行機能命令が含まれる回路を RUN 中オンライン変更するときには、十分注意してください。

パルス実行機能命令: $\neg P \uparrow$, $\neg N \uparrow$, $\neg(P) \downarrow$, $\neg(N) \downarrow$,

$\neg \uparrow \uparrow$, $\neg \downarrow \downarrow$

ファンクション命令のパルス化指定

5.11.3 デバッグ モード機能

PCは、プログラムデバッグを支援するための専用のモードとして DEBUG モードを持っています。DEBUG モードにすることによって次の各機能が可能となり、効率的なプログラムデバッグが行えます。(T3ではバージョン 1.3 以降可能)

- ブレークポイント設定機能
ブレークポイントが設定された命令の直前まで実行して停止します。
- シングルステップ実行機能
1命令単位で実行、停止を繰り返します。
- 回路実行機能
1回路単位で実行、停止を繰り返します。
- Nスキャン実行機能
設定されたスキャン回数(1~65535)だけ実行して停止します。
- 停止条件設定機能
設定された停止条件(レジスタ/デバイスの状態設定)が成立するまで実行して停止します。

DEBUG モード DEBUG モードへの移行は HALT モードからのみ可能です。また DEBUG モードの中には、D-HALT、D-RUN、D-STOP の3つのサブモードがあります。

D-HALT モード : HALT モードから DEBUG モードに移行したとき、及び D-STOP モードでデバッグコマンド「モード遷移」が実行されたとき、このモードとなります。プログラム実行は停止です。
上記デバッグモード機能の実行条件設定はこのモードで行います。

D-RUN モード : プログラム実行モードです。上記デバッグモード機能の各停止条件成立により D-STOP モードに遷移します。

D-STOP モード : プログラム実行一時停止モードです。D-STOP モードに遷移した要因はプログラマにて確認できます。

各モードの遷移条件詳細については、2.3 節を参照して下さい。

入出力処理選択 D-RUN モードにおいて、I/O モジュールとの交信を行うかどうかの選択が可能です。入出力処理禁止と設定した場合には、一括入出力処理は行わず、直接受け入出力命令も不実行となります。また、直接入出力カーバイス/レジスタ(I/O 及び IW/OW)を使用した場合もイメージテーブル(X/Y 及び XW/YW)が対象となります。

入力状態はプログラマから設定できます。外部に出力を出さずにプログラムデバッグを行うときに使用します。

なお、入出力処理禁止に設定されているときには、プログラマに表示される PC の動作モードは、S-HALT(D-HALT に対応)、S-RUN(D-RUN に対応)、S-STOP(D-STOP に対応)となります。

なお、入出力処理許可/禁止の選択は、D-HALT または S-HALT の状態で行います。

トレースバック

機能 シングルステップ実行、回路実行以外のデバッグモード機能でプログラムを実行しているときは、過去 10 スキャン分のオンライントレース情報(回路活性表示、レジスタ値表示データ)を保存しています。

従って、停止条件成立などで実行停止したとき、過去 10 スキャンまで順次さかのぼって実行状態を表示し確認することができます。

補足

- (1) トレースバックできるのは現在モニタ中の画面範囲のみです。
- (2) シングルステップ実行及び回路実行では、トレースバック機能は無効です。

機能詳細

(1) ブレークポイント設定機能

実行停止状態から、ブレークポイントが設定された命令の直前まで実行して停止します。ブレークポイントの設定は 1ヶ所のみ可能です。

デバッグ実行条件設定にて、ブレークポイント検出回数に 0 以外の値が設定されたときに、ブレークポイント設定が有効となります。ブレークポイント検出回数の設定は、0~65535 の設定が可能で、0 のときはブレークポイント設定なし、それ以外のときは設定回数だけブレークポイントを検出したときに停止するという動作となります。

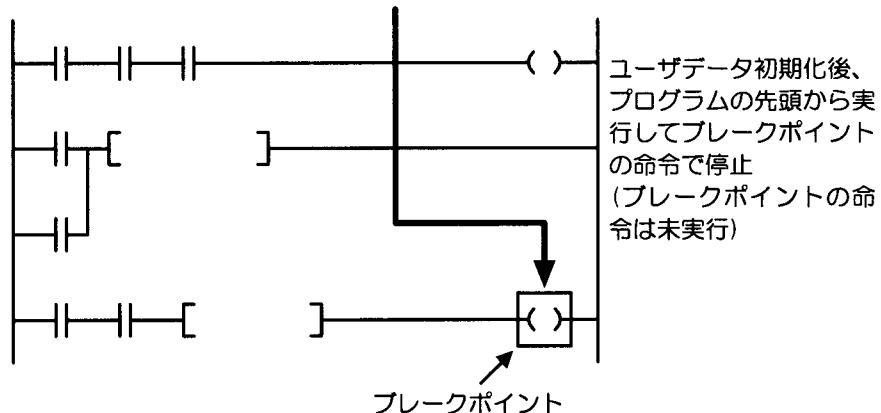
プログラム実行方法の指定は、初回実行と継続実行の指定が可能です。

- 初回実行……ユーザデータを初期化してプログラムの先頭から実行

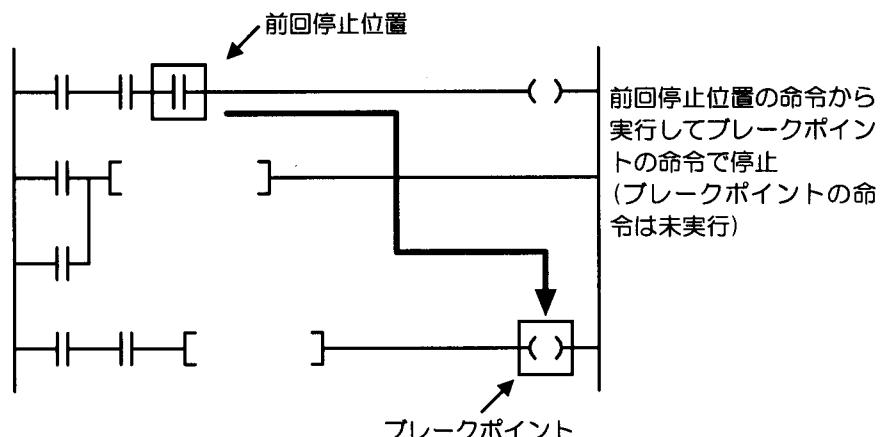
- 継続実行……実行停止位置から継続して実行を再開

なお、D-HALT からの起動時は常に初回実行となります。

実行例1(初回実行)



実行例2(継続実行)

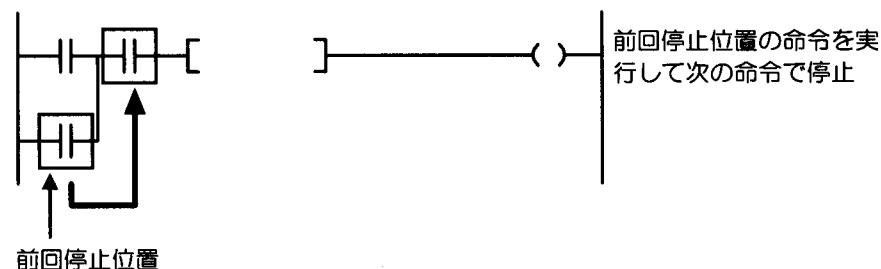


(2) シングルステップ実行機能

1 命令実行する毎にプログラム実行を停止します。D-HALT モードからシングルステップ実行を起動した場合には、ユーザデータ初期化を行い、メインプログラムの先頭命令で停止します(停止位置の命令は未実行)。D-STOP モードからの起動時は、前回停止位置の命令を実行し、次の命令で停止します。

5

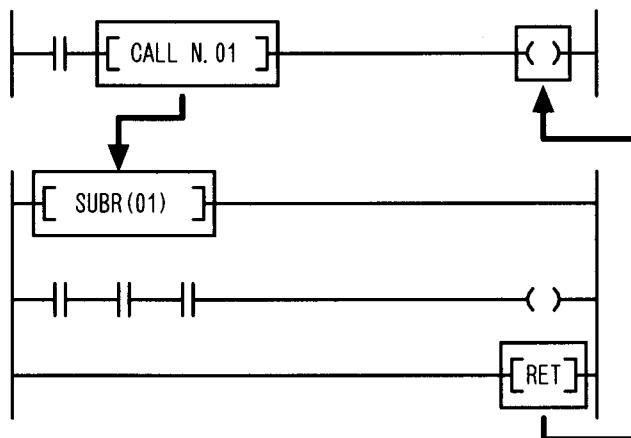
実行例1



サブルーチンコール命令(CALL)で停止し、サブルーチンへの分岐条件が成立しているときには、次の停止位置は分岐先のサブルーチンエントリ命令(SUBR)となります。

また、サブルーチンリターン命令(RET)の次は、呼び出し元のサブルーチンコール命令(CALL)の次の命令で停止します。

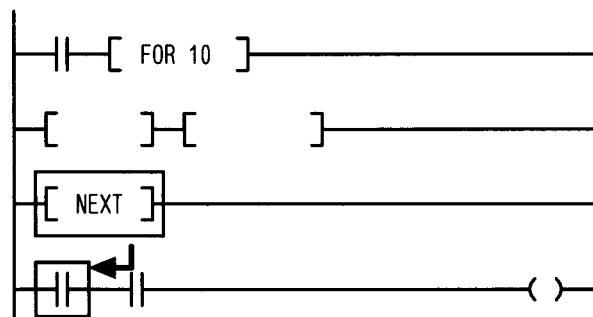
実行例2 (CALL/RET)



条件ジャンプ命令(JUMP)で停止し、ジャンプの条件が成立しているときには、次の停止位置はジャンプ先のラベル命令(LBL)となります。

FOR-NEXT 命令の場合、NEXT 命令の実行後は次の回路の先頭命令で停止します。FOR-NEXT ループの中は指定の回数だけ繰り返し実行されますが、シングルステップ実行によって追跡することはできません。(FOR-NEXT ループの中のトレース表示は1回目の実行時の状態が表示されますので注意して下さい)

実行例3 (FOR-NEXT)



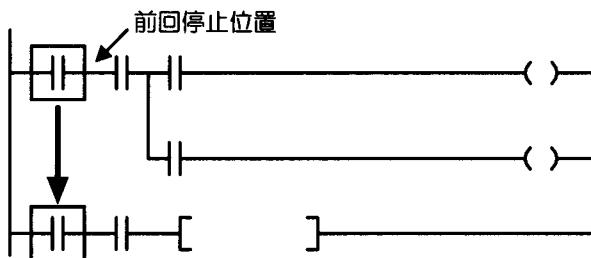
割り込みプログラムでシングルステップ実行を使用した場合（割り込みプログラム内に設定したブレークポイントで停止し、続けてシングルステップ実行を行う場合）、割り込みリターン命令（IRET）の実行後は、実行中の割り込みプログラムの先頭命令で停止します。

（割り込み発生前の実行位置には戻りませんので注意して下さい）

(3) 回路実行機能

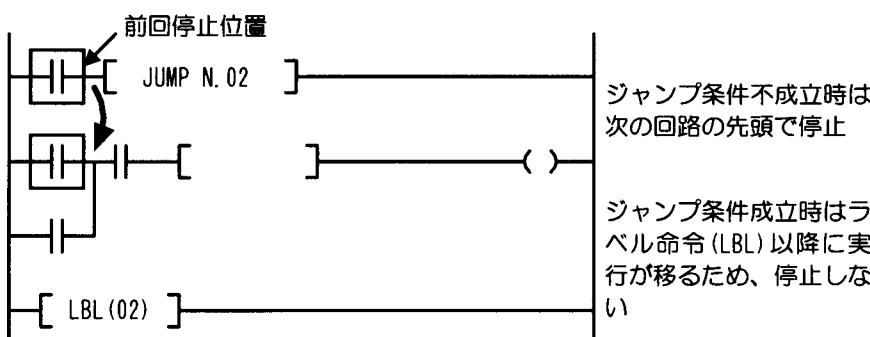
1回路実行する毎にプログラム実行を停止します。D-HALT モードから回路実行を起動した場合には、ユーザデータ初期化を行い、メインプログラムの先頭命令で停止します（停止位置の命令は未実行）。
D-STOP モードからの起動時は、前回停止位置の回路を実行し、次の回路の先頭で停止します。

実行例 1



サブルーチンコール命令（CALL）や条件ジャンプ命令（JUMP）が存在する回路の場合は、分岐の有無にかかわらず次の回路の回路の先頭命令にて停止します。従って、条件ジャンプ命令（JUMP）によって次の回路をジャンプした場合には停止しなくなりますので注意して下さい。

実行例 2 (JUMP)



FOR-NEXT 命令の場合には、シングルステップ実行と同様に、NEXT 命令の実行後は次の回路の先頭で停止します。FOR-NEXT ループの中は指定の回数だけ繰り返し実行されますが、追跡することはできません。(FOR-NEXT ループの中のトレース表示は1回目の実行時の状態が表示されますので注意して下さい)

割り込みプログラムで回路実行を使用した場合は、シングルステップ実行と同様に、割り込みリターン命令(IRET)の実行後は実行中の割り込みプログラムの先頭命令で停止します。(割り込み発生前の実行位置には戻りませんので注意して下さい)

(4) Nスキャン実行機能

設定されたスキャン回数だけプログラムを実行し、メインプログラムの終了位置で停止します。

スキャン回数の設定は、デバッグ実行条件設定にて、0~65535 の範囲で設定します。0のときはスキャン回数設定なし(連続実行)となります。

プログラム実行方法の指定は、初回実行と継続実行の指定が可能です。

- 初回実行 … ユーザデータを初期化してプログラムの先頭から実行
 - 継続実行 … 実行停止位置から継続して実行を再開
- なお、D-HALTからの起動時は常に初回実行となります。

(5) 停止条件設定機能

設定された停止条件(レジスタ/デバイスの状態設定)により、プログラム実行を停止します。停止条件のチェックポイントは、スキャン終了時点またはブレークポイント検出時点の選択が可能です。

停止条件の設定は、デバッグ実行条件設定にて停止条件有りを選択することにより有効となり、下記項目について条件を設定します。

- 停止条件チェックポイント

停止条件の成立をどの時点で行うかを設定します。

スキャン終了時またはブレークポイント検出時のいずれかを選択します。なおブレークポイント検出時を選択するときには、ブレークポイント設定が有効となっている必要があります。(このときブレークポイント検出回数に大きな値を設定しないと意味がありません)

- 判定条件成立回数

設定した回数だけ下記判定条件が成立したときに、停止条件成立となります。設定は、1～65535 の範囲です。

- 判定条件

最大4個のレジスタ/デバイスの状態の組み合わせにて設定します。組み合わせ条件は、OR 条件または AND 条件の選択が可能です。

(例1)

組み合わせ条件 AND				
チェック対象	X0100	R110F	D0050	D0311
チェックデータ	○	●	100	-2000



X0100 が OFF, R110F が ON, D0050 が 100, D0311 が-2000 の全てが成立したときに判定条件成立

(例2)

組み合わせ条件 OR				
チェック対象	X0001	D4900	Y0508	C.003
チェックデータ	●	H0100	●	●

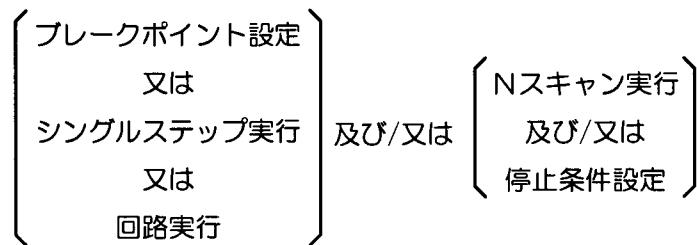


X0001 が ON, D4900 が H0100, Y0508 が ON, C.003 が ON のいずれかが成立したときに判定条件成立

プログラム実行方法の指定は、初回実行と継続実行の指定が可能です。

- 初回実行 …… ユーザデータを初期化してプログラムの先頭から実行
 - 継続実行 …… 実行停止位置から継続して実行を再開
- なお、D-HALT からの起動時は常に初回実行となります。

共通事項 (1) 各デバッグモード機能は、単独での使用の他に、下記組み合わせでの使用が可能です。



- (2) D-HALT (S-HALT) から D-RUN (S-RUN) への遷移時にはイニシャルロードは行われません。
- (3) プログラム中で使用しているタイマは、連続スキャン時は通常更新処理、シングルステップ実行時及び回路実行時は 100ms/スキャンとしてタイマ更新が行われます。ただしスキャンモードが定刻スキャンに設定されているときには、設定周期/スキャンとしてタイマ更新が行われます。
- (4) シングルステップ実行時及び回路実行時は、サブプログラムの中断処理は行われません。連続スキャン時は通常通り中断処理が行われます。
- (5) 割り込みプログラムの動作は下記のようになります。

D-HALT (S-HALT) 時 …… 禁止

D-STOP (S-STOP) 時 …… 保留(許可になった瞬間に実行)

D-RUN (S-RUN) 時 …… 許可

*) 連続スキャンとは、シングルステップ実行または回路実行以外のデバッグモード機能で、連続的にスキャンを実行している状態を示します。

制約事項 (1) SFC で作成されたプログラムブロックでは、デバッグモード機能は使用できません。シングルステップ実行や回路実行で SFC ブロックが実行対象となった場合は、SFC ブロック実行後のラダー図ブロックの最初の命令で停止します。ブレークポイントは SFC ブロック内には設定できません。また、トレースバック機能は SFC ブロックでは使用できません。

- (2) コンピュータリンク経由及び TOSLINE-S20 経由では、デバッグモード機能は使用できません。
- (3) プログラム変更は HALT モードで行って下さい。D-HALT または S-HALT モードでプログラム変更を行った場合、ブレークポイント、シングルステップ実行及び回路実行において、初回の実行時に停止しない場合があります。
- (4) Nスキャン実行にて1スキャン停止を行った場合には、割り込みプログラムは実行されません。割り込みプログラムを実行させる場合は、スキャン回数を2回以上として下さい。
- (5) 同じ命令にブレークポイントを設定して繰り返し実行を行う場合は、ブレークポイント検出にて停止後、シングルステップ実行または回路実行を行ってからブレークポイントを再設定して下さい。そうしないと、割り込みプログラムが存在し、停止中に割り込み要因が発生した場合に、メインプログラムが実行されなくなってしまいます。割り込みプログラムが存在しない場合には、上記の注意は必要ありません。

補足

D-STOP 及び D-RUN モード時は、CPU モジュール上の FAULT LED が点滅し、S-STOP 及び S-RUN モード時は FAULT LED と I/O LED が点滅しますが異常ではありません。

5.12 システム診断機能

制御対象の動作/状態を診断するために以下の機能を準備しています。これらの機能を使用することにより、容易にシステム監視機能を実現することができます。

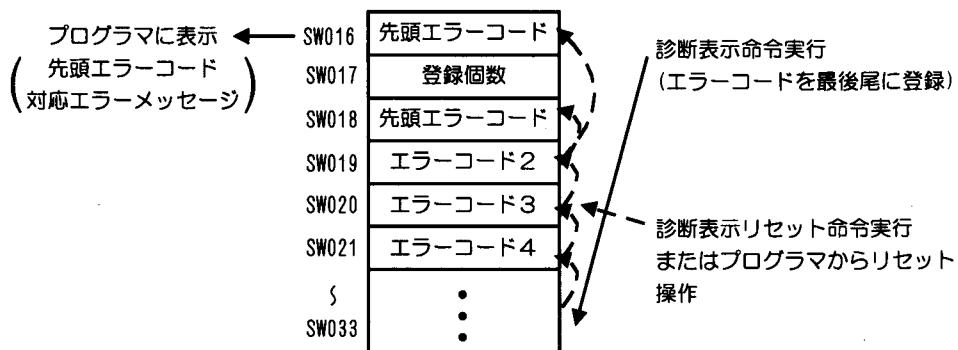
(1) 診断表示機能

ユーザプログラムにて診断表示命令(DIAG)を使用することによって、制御対象に異常が発生したときに、該当エラーメッセージ(1メッセージ最大12文字)をプログラマ上に表示させることができます。また、対応するエラーコード(1~64)は、発生順に最大16個まで、特殊レジスタ(SW016~SW033)に格納され、エラーコードに対応するアンシェータリレー(S0340~S037F)がONとなります。特殊レジスタ/リレーを利用して、外部表示器等にエラーコードを表示させることも可能です。

登録されたエラーコードは、プログラマ操作(診断確認)または診断表示リセット命令(DIAR)によって、1個づつリセット(消去して以降シフトアップ)することができます。

この機能は、以降に述べるビットパターンチェック、シーケンス渋滞検出と組み合わせて使用するとさらに効果的です。

(診断表示命令の詳細は、別冊の命令語説明書をご覧下さい)



登録されているエラーコードが、例えば、3, 10, 29, 58だったとき、各々に対応するアンシェータリレー、S0342, S0349, S0350, S0379がONとなります。

(アンシェータリレー)

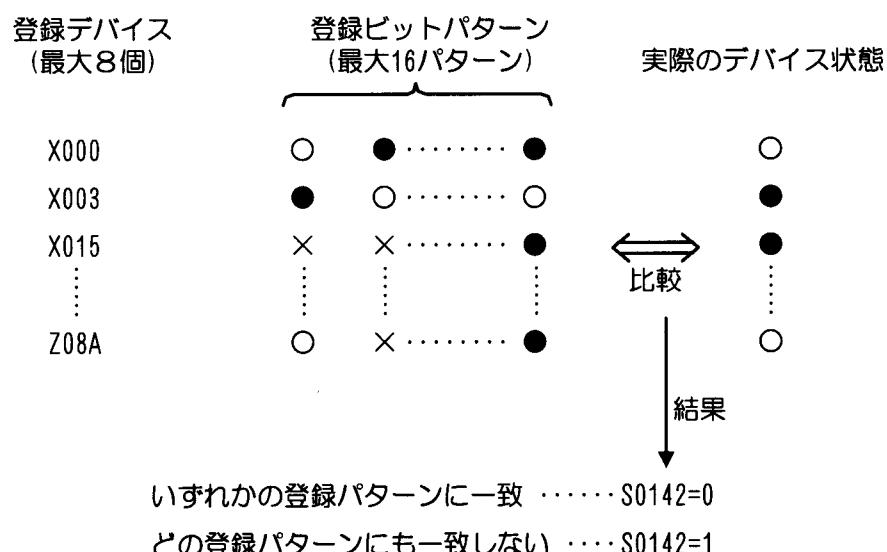
	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	O
SW034	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SW035	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
SW036	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
SW037	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

(2) ビットパターンチェック機能

複数のデバイスの ON/OFF 状態が、正常な組み合わせ（パターン）かどうかをチェックする機能です。例えば、デバイス 1, 2, 3 は同時に 2つ以上 ON することはないなどをチェックする場合に有効です。

登録は最大8個のデバイスについて、最大16パターンまで可能です。

チェックは一括入出力処理後に行われ*、結果は特殊リレーS0142に反映されます。(T3ではバージョン1.2以降可能)



なお、登録パターン中、○は OFF、●は ON、×は ON/OFF いずれでも可を表わします。

デバイス及びビットパターンの登録は、プログラマのユーザ診断モードにて行います。

ユーザプログラムで S0142 を利用することによって、異常な入力による誤動作を防止することができます。

- *) ピットパターンチェックの実行タイミングは、特殊リレーS015Fにより
変更することができます。(T3 ではバージョン 1.4 以降)

(3) レジスタ値正当性チェック機能

レジスタの値が正当な数値範囲内かどうかをチェックする機能です。レジスタは最大4個まで登録することができ、各々について最小値と最大値を登録します。なお、レジスタの値を整数(符号付き)と見なすか、正整数(符号なし)と見なすかの選択が可能です。

チェックは一括入出力処理後に行われ*、結果は特殊リレーS0143～S0146に格納されます(範囲内:0、範囲外:1)。

(T3 ではバージョン 1.2 以降可能)

登録レジスタ (最大4個)	タイプ	最小値	最大値	実際のレジスタの値
XW034	符号なし	0	400	200
XW035	符号付き	-1500	1500	2000
D0011	符号なし	H0200	H9000	H1234
W0100	符号付き	-300	600	-1000

比較

↓

結果

レジスタ及び数値範囲の登録は、プログラマのユーザ診断モードにて行います。

ユーザプログラムで S0143 から S0146 を利用することによって、異常な入力による誤動作を防止することができます。

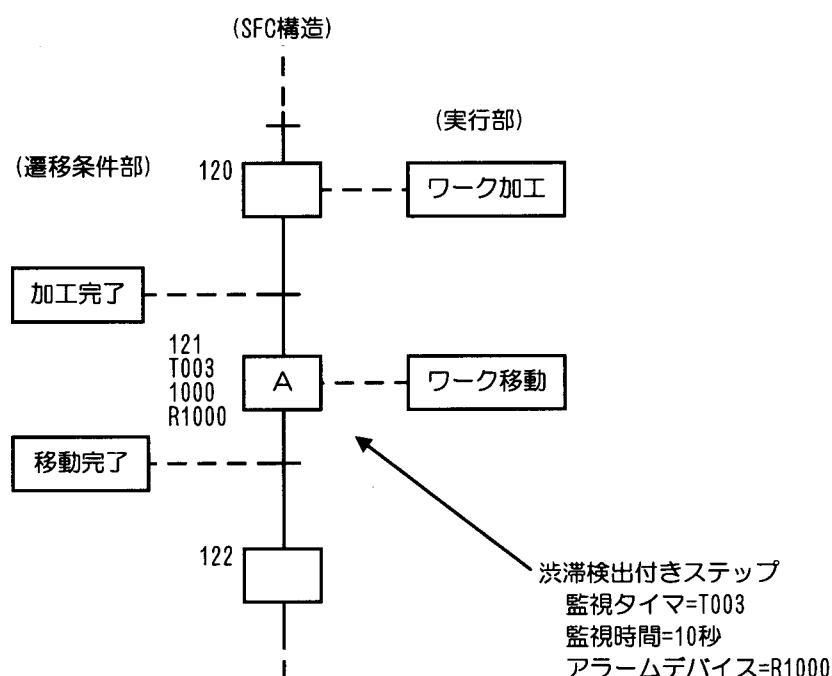
- *) レジスタ値正当性チェックの実行タイミングは、特殊リレーS015Fにより変更することができます。(T3 ではバージョン 1.4 以降)

S015F が OFF のとき	ユーザプログラム実行の直前 (一括入出力後)
S015F が ON のとき	ユーザプログラム実行の直後

(4) シーケンス渋滞検出機能

SFC(シーケンシャルファンクションチャート)の命令の1つに渋滞検出付きステップがあります。この渋滞検出付きステップは、ステップが起動されてから、設定時間内に次ステップに移る条件が整わないときに、指定デバイスをONする機能があります。

この機能を使用することによって、工程歩進制御における動作停滯検出を容易に実現することができます。



上記の例の場合、ワーク移動を開始してから10秒以内に移動完了(ワーク到達信号ON等)しない場合に、指定のアラームデバイス(R1000)がONとなります。これによって、ワーク駆動系または検出系に異常が発生したことを検出することができます。

SFCについては、本書第3部及び別冊の命令語説明書をご覧下さい。

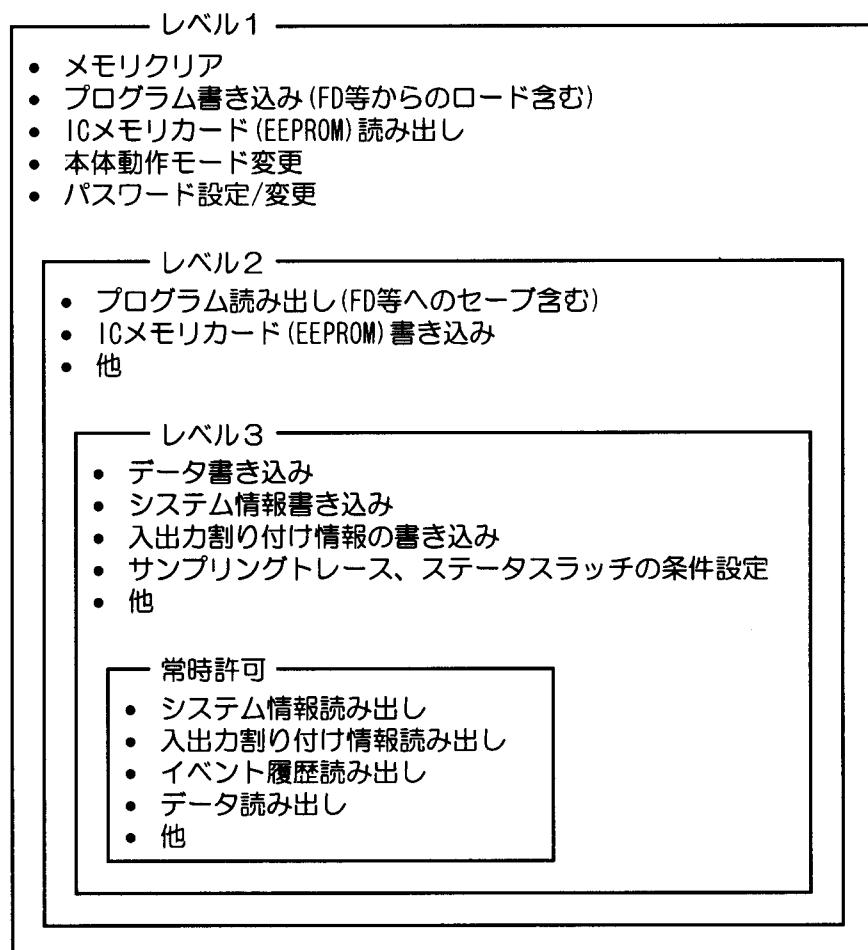
5.13

パスワード機能

PC を適用したシステムのセキュリティを保つため、PC はパスワード機能を備えています。(T3 ではバージョン 1.4 以降)

パスワードとしては、レベル1パスワード、レベル2パスワード、レベル3 パスワードの3種類が登録でき、レベルに応じてプログラマからの要求コマンドに対する制約が設けられます。

各レベルで実行可能なコマンドの分類の概略を下図に示します。



例えば、レベル1とレベル2のパスワードが登録されている状態で PC に電源が投入されたときには、レベル3以下のコマンドのみ実行可能となります。この状態でレベル2のパスワードを入力すると、レベル2以下のコマンドが実行可能となります。

(各レベルで実行可能なコマンドの詳細については、別冊のプログラマ説明書をご覧下さい)

第3部

ユーザプログラム説明

1.1

第3部の目的

PC の機能の主体は、ユーザプログラムを記憶し、記憶したユーザプログラムを実行し、実行の結果として、機械/プロセスの動作/状態を制御及び監視することにあります。

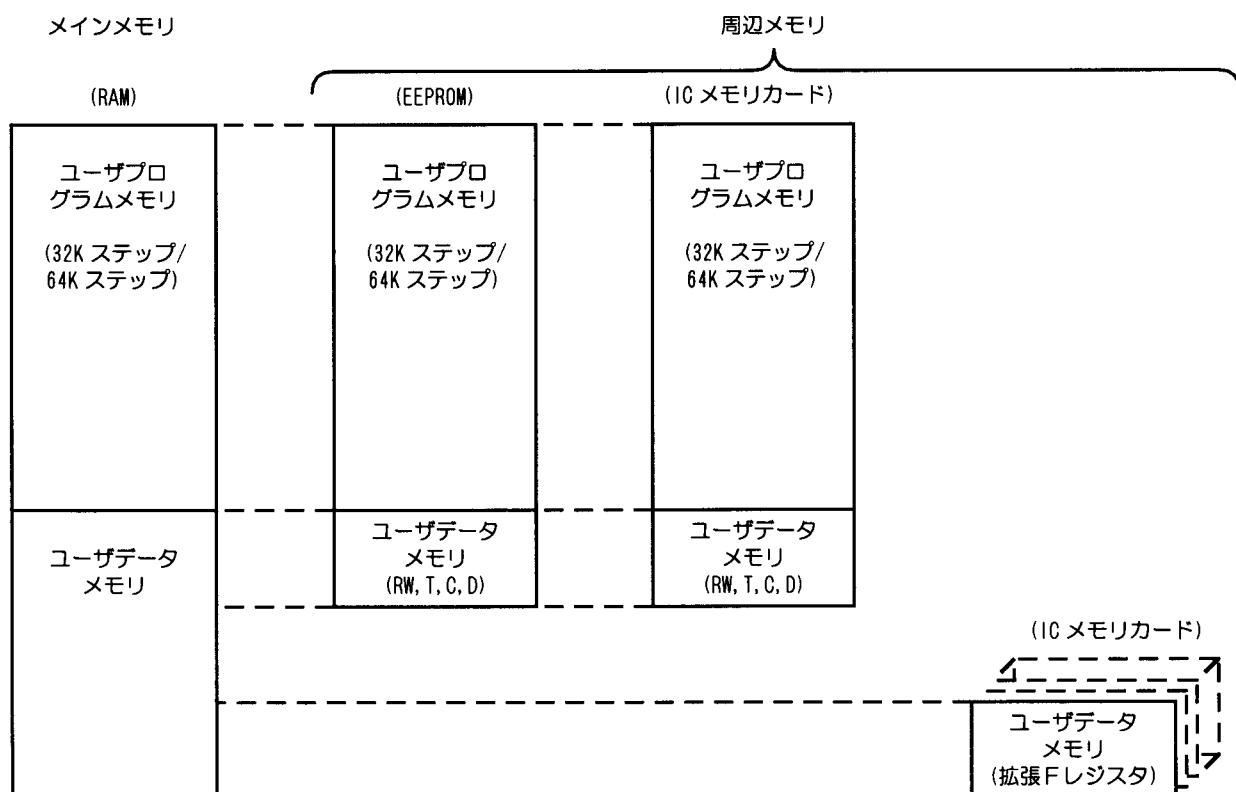
ユーザプログラムとは、要求制御機能を実現するための動作順序、動作条件、データ加工処理、オペレータとのインターフェース方法などを、一連の命令語によって記述し、ユーザプログラムメモリに記憶させたものです。ユーザプログラムを実行するということは、外部入出力データや制御パラメータが格納されているユーザデータを読み出し、各々の命令語の処理を行い、その結果をユーザデータメモリに格納するという処理を連続的に行うことと申すことができます。

第2部では、PC 本体の内部でどのような処理が行われ、ユーザプログラムはどのように実行されるのか、また PC 内部及び PC が制御する機械/プロセスが正常状態を保つために PC 本体はどのような機能をサポートしているのかなどについて説明しました。この第3部では、ユーザプログラムを作成するために必要な情報、つまりユーザデータの詳細、入出力割り付けの詳細及びプログラム言語について説明します。また PC が備えているマルチタスク機能を活用するために、ユーザプログラムの構成について整理して説明します。

1.2

ユーザメモリ構成

下図に PC のユーザメモリ構成を示します。



PCを使用する上で、ユーザが使用することのできるメモリエリアをユーザメモリと呼びます。ユーザメモリを形態で大別すると、メインメモリと周辺メモリに分けることができ、機能で分類すると、ユーザプログラムメモリとユーザデータメモリに分けて考えることができます。

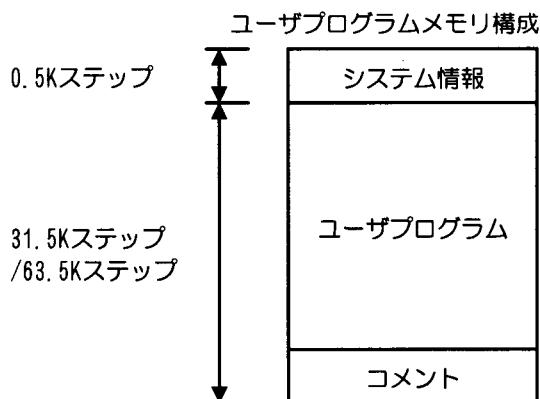
メインメモリとは、常にPCのCPUモジュール内に存在するメモリで、バッテリバックアップされたRAM(スタティックRAM)によって構成されます。一方周辺メモリとは、必要に応じてユーザが使用選択を行うことのできるメモリで、EEPROMまたはICメモリカードによって構成されます。周辺メモリは、メインメモリのバックアップ(ユーザプログラム+RW,T,C,Dの各レジスタ)、またはメインメモリの拡張エリア(拡張ファイル(F)レジスタ、ICメモリカードのみ)として使用することができます。

ユーザプログラムメモリは、CPUモジュールにより32Kステップと64Kステップ(ステップは命令語記憶単位)の容量があり、ラダー図、SFCなどのプログラム言語によって作成されたユーザプログラムを記憶します。

ユーザデータメモリは、制御のパラメータである入出力データや変数を格納するメモリで、機能によって、入出力レジスタ、データレジスタ、ファイルレジスタなど、いくつかの種別に分かれています。

2.1 ユーザプログラムの 概要

ユーザプログラムメモリは、下図のようにシステム情報格納領域、ユーザプログラム格納領域、及びコメント格納領域に分けられます。



システム情報とは、ユーザプログラムを実行する際の実行制御パラメータや、ユーザプログラム管理情報を格納する領域で、常に 0.5K ステップを占有します。

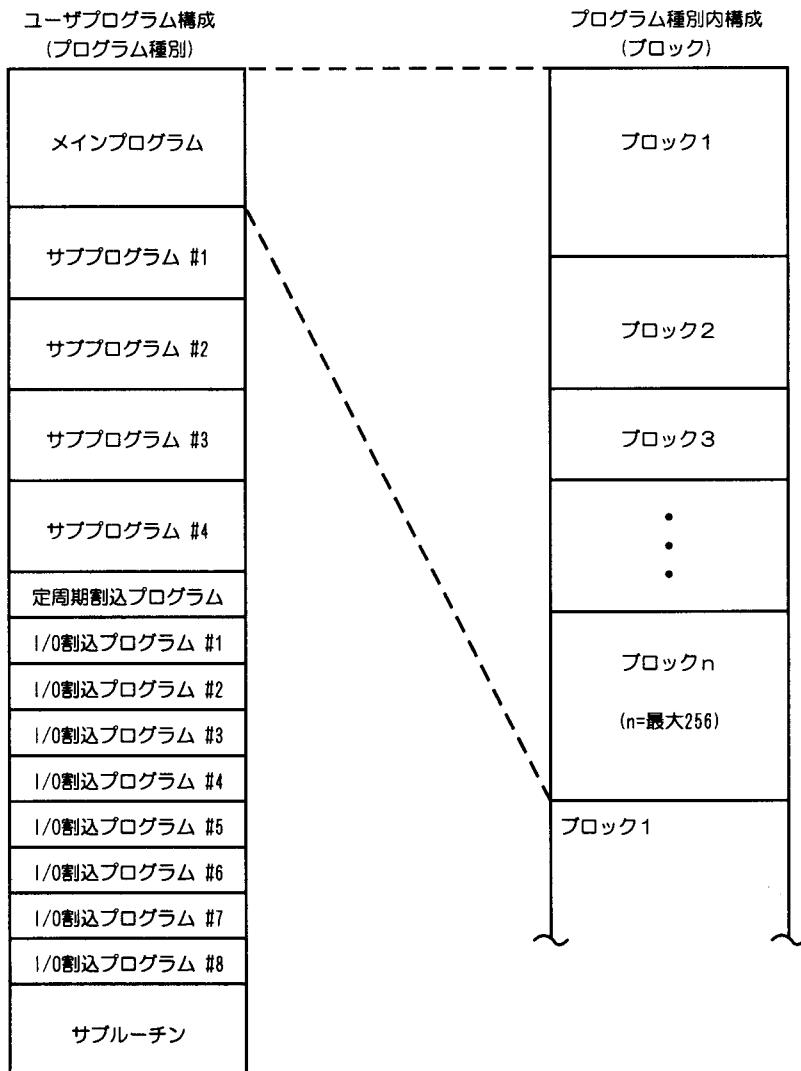
コメントとは、ユーザプログラムの読みやすさのために付けるもので、デバイスネーム及びデバイスコメントなどがあります。コメント格納領域は固定ではなく、ユーザプログラムメモリ (32K ステップまたは 64K ステップ) をシステム情報とユーザプログラムに割り当てた残り領域をコメント格納用として使用します。

ユーザプログラムは、実行形態によって、メインプログラム、サブプログラム、割り込みプログラム、及びサブルーチンの各プログラム種別に分類されます。

このうち、メインプログラムは必ず存在しなければいけないプログラムで、ユーザプログラムの中心となるものです。

一方、サブプログラムと割り込みプログラムは、メインプログラムのみでは要求制御機能を実現することが難しいときに、必要に応じて作成/使用するもので、無くともかまいません。

また、サブルーチンとは、プログラム中で同じ処理を繰り返し行う必要がある場合や、1つの機能をブロック化してプログラムを見やすくするためなどに使用するもので、必要がなければ無くともかまいません。



また、各プログラム種別の中では、ブロックという単位で、ユーザプログラムは管理されます。

内部的には、各ブロックの先頭にはブロック定義ラベルが存在しており、ブロック定義ラベルには、プログラム種別、ブロック番号、及びプログラム言語情報が付随しています(ユーザはブロック定義ラベルを意識する必要はありません)。

T3/T3H ではラダー図、SFC の2つのプログラム言語を併用することができますが、1つのブロック内では1つの言語しか使用できません。

補足

- (1) 各プログラム種別及びブロックにおいて、プログラム容量(ステップ数)の制約はありません。総容量(31.5K ステップまたは 63.5K ステップ)によってのみ制約されます。
- (2) ブロック番号は連続していなくてもかまいません。つまり途中に空きブロックがあってもかまいません。

2.2 システム情報

システム情報とは、ユーザプログラムを実行する場合の実行制御パラメータやユーザプログラム管理情報を格納する領域で、ユーザプログラムメモリの0.5Kステップを占有します。システム情報には以下の内容が含まれます。

(1) プログラム ID

ユーザプログラム識別名です。半角英数字10文字まで(漢字などの全角文字は不可)の設定ができます。プログラムIDは、プログラマのシステム情報画面にて登録/モニタできます。

(2) システムコメント

ユーザプログラムに付けるコメントです。半角英数字30文字まで(全角文字も可能)の設定ができます。システムコメントは、プログラマのシステム情報画面にて登録/モニタできます。

(3) メモリ容量

本体のメモリタイプ(ユーザプログラム容量/データレジスタ容量)を格納します。システム側が自動登録しますので、ユーザによる登録は不要です。メモリ容量は、プログラマのシステム情報画面にてモニタできます。

(4) 使用ステップ

ユーザプログラムで使用しているステップ数を格納します。ユーザプログラムの作成/変更の都度、システム側が自動更新しますので、ユーザによる登録は不要です。使用ステップは、プログラマのシステム情報画面にてモニタできます。

(5) PCタイプ

本体の機種タイプを格納します。システム側が自動登録しますので、ユーザによる登録は不要です。PCタイプは、プログラマのシステム情報画面にてモニタできます。

(6) プログラム容量

ユーザプログラム用に割り当てる容量を設定し登録します。総容量(32Kステップまたは64Kステップ)からここで登録した容量を引いた残り分が、コメント格納領域として確保されます。プログラム容量は、プログラマのシステム情報画面にて登録/モニタできます。

(7) サンプリングバッファ容量

サンプリングトレース機能によるサンプリングデータの格納容量を設定し登録します。最大設定は8Kワードです。サンプリングバッファ容量は、プログラマのシステム情報画面にて登録/モニタできます。

(8) 停電保持範囲指定

ユーザデータのうち、補助レジスタ(RW), タイマレジスタ(T), カウンタレジスタ(C)及びデータレジスタ(D)について、停電/復電時に停電前のデータを保持するアドレス範囲を設定し登録します。ここで登録した範囲は、ユーザデータ初期化処理の対象外となります。これらのレジスタ各々について、先頭アドレス(0)から指定アドレスまでの範囲が、停電保持領域となります。停電保持範囲指定は、プログラマのシステム情報画面にて登録/モニタできます。

(9) 10ms タイマ範囲指定(T3Hのみ)

T3H(PU325H/PU326H)において 10ms タイマとして使用する範囲を設定し登録します。

設定範囲は T000～T999 で、T000～設定したタイマレジスタまでが 10ms タイマとなります。

初期設定は T000～T063 です。

(10) スタートモード(T3Hのみ)

電源投入後の T3H のモードを選択します。

スタンバイを選択すると電源投入後 T3H は CPU 本体のモードスイッチの状態に関わらず HALT モードで立ち上がります。

オートを選択すると電源投入後、CPU 本体のモードスイッチが RUN もしくは P-RUN の場合、RUN もしくは P-RUN モードで立ち上がります。

(11) スキャン時間

スキャンモード(フローティング/定刻)の設定を行い登録します。スキャン時間が登録なし(ブランク)のとき、フローティングスキャンモードとなり、スキャン時間に数値が設定されたとき、その時間をスキャン周期とする定刻スキャンモードとなります。スキャン周期の設定は、10～200ms(10ms 単位)です。スキャン時間は、プログラマのシステム情報画面にて登録/モニタできます。

(12) サブプログラム実行時間

フローティングスキャンモードにおけるサブプログラム実行時間制限の設定、登録を行います。設定範囲は、1～100ms(1ms 単位)です。サブプログラム実行時間は、プログラマのシステム情報画面にて登録/モニタできます。

(13) 定周期割り込み周期

定周期割り込みプログラムの割り込み周期を設定し登録します。設定範囲は、T3 の場合 2~1000ms、T3H の場合 1~1000ms(1ms 単位)です。定周期割り込み周期は、プログラマのシステム情報画面にて登録/モニタできます。

(14) コンピュータリンク伝送パラメータ

コンピュータリンクを使用する場合の伝送パラメータを設定し登録します。プログラマのシステム情報画面にて登録/モニタできます。
伝送パラメータの項目及び設定範囲は次の通りです。

- ステーション No. …… 1~32(初期値=1)
- ポーレート(bps) …… 300, 600, 1200, 2400, 4800, 9600, 19200(初期値 9600)
- パリティ…………… 無し、奇数、偶数(初期値=奇数)
- データ長(ビット) …… 7, 8(初期値=8)
- ストップビット……… 1, 2(初期値=1)

(15) 入出力割り付け情報

入出力割り付け情報及びユニット先頭アドレス指定情報が格納されます。これらの情報は、プログラマの一般 I/O 割り付け情報画面にて、入出力自動割り付けコマンドを選択実行するか、またはスロット毎に I/O モジュール種別を設定し登録することによって作成されます。

(16) 割り込み割り付け情報

I/O 割り込みプログラムと割り込み機能付き I/O モジュールとの対応情報が格納されます。デフォルト(設定を行わない状態)では、CPU に近い割り込み機能付き I/O モジュールから順に、番号の若い I/O 割り込みプログラムが割り付けられます。割り込み割り付け情報は、プログラマの割り込み割り付け情報画面にて、設定変更/モニタすることができます。

(17) 伝送入出力割り付け情報

データ伝送装置(TOSLINE-S20, TOSLINE-F10)に割り付けられるリンクレジスタの領域、及びデータ入出力方法の情報が格納されます。伝送入出力割り付け情報は、プログラマの伝送入出力割り付け情報画面にて、登録/モニタできます。

2.3

ユーザプログラム

ユーザプログラムは、メインプログラム、サブプログラム(#1～#4)、割り込みプログラム(定周期、I/O#1～I/O#8)、及びサブルーチンの各プログラム種別から構成されます。これらのプログラム種別のうち、メインプログラムは必ず存在しなければいけませんが、他のプログラム種別は使用しなければ全く存在しなくてもかまいません。従って、ユーザプログラムはメインプログラムのみ、という構成も当然可能です。

2

また、各プログラム種別の中では、ブロックという単位でプログラムを分割することができます(必要がなければブロック分割は不要)。ブロック分割は次の場合に行います。

- ラダー図以外の言語を使用するとき(1言語/1ブロック)
- 複数本のSFCプログラムを作成するとき(SFC 1本/ブロック, 5.3節参照)
- 制御機能単位でブロック分割した方が、プログラムが見やすくなるとき

各プログラム種別及びブロックによる、プログラム容量(ステップ数)の制限はありません。(SFCの場合を除く)

ブロック番号としては、1から256までが有効です。ただしブロック番号は連続していなくてもかまいません。プログラム実行上は、若いブロック番号から順に存在するブロックのプログラムが実行されます。

プログラミングにおいては、プログラマのプログラム読み出し機能によって、プログラム種別とブロック番号を指定して所定の部分を画面上に表示させた上で、必要なプログラム作成/編集を行うことになります。

2.3.1

メインプログラム

メインプログラムは、毎スキャン必ず1回実行される、ユーザプログラムの中心となる部分です。メインプログラムとして認識されるのは次の範囲です。

プログラム種別“メインプログラム”内の

最も若いブロック番号の先頭の命令 から

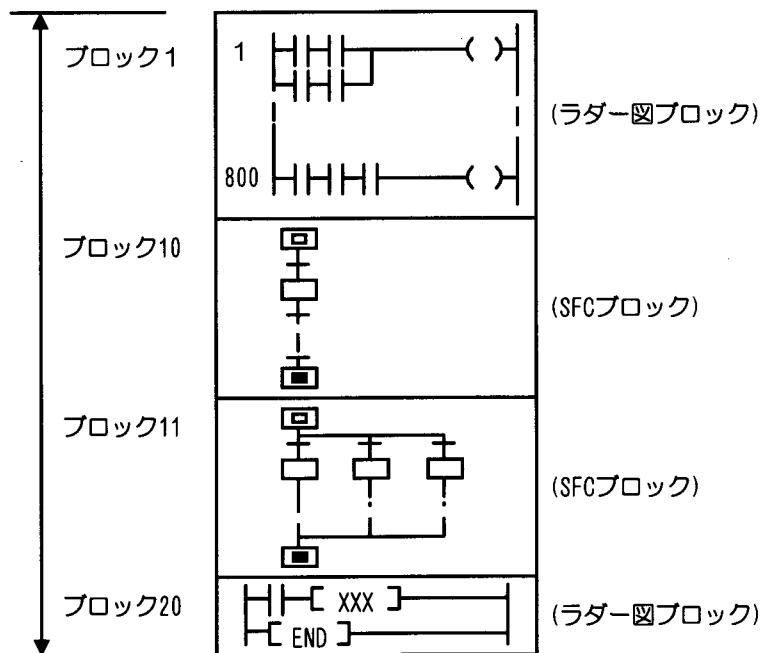
↓

ラダー図のEND命令 まで

2

なお、END命令の後に命令が存在していてもかまいませんが、その部分は実行はされません。(ただし使用ステップ数としてはカウントされます)

(メインプログラムの構成例)



2.3.2

サブプログラム

サブプログラムは、PC の特長の1つであるマルチタスク機能を実現するためのプログラム種別であり、#1～#4 の4本が作成可能です。サブプログラムの起動は、メインプログラム、割り込みプログラム及び他のサブプログラムから行うことができます。

サブプログラムを使用することによって、メインプログラムのスキャン周期を伸すことなく、オペレータインターフェースプログラムやデータ処理プログラムを実行させることができます。また、制御対象に応じた多重スキャンシステム(高速スキャン部、低速スキャン部など)を構成することもできます。

サブプログラム#1(サブ#1)は、RUN 起動時の第1スキャンにメインプログラムの前に実行されます。従ってサブ#1 は初期設定用のプログラムとして利用して下さい。

サブプログラム#2(サブ#2)は、設定によって、瞬停再始動時の初期設定プログラム(特殊モード)または他のプログラム種別からの起動要求によって起動される通常のサブプログラム(一般モード)の選択が可能です。

サブプログラム#3(サブ#3)とサブプログラム#4(サブ#4)は、通常のサブプログラム(一般モード)の機能に固定です。

またサブ#2～サブ#4 の一般モードでは、設定によって起動要求時に1回だけ起動される1回実行モードと、起動要求が ON している間設定したスキャン回数毎に繰り返し実行されるサイクリック実行モードの選択が可能です。

— 補足 —

サブプログラムの動作の詳細については、第2部 3.2 節をご覧下さい。
また、サブプログラム#2 については、第2部 5.5 節も参照して下さい。

サブプログラム#n として認識されるのは次の範囲です。

プログラム種別 “サブ#n” 内の

最も若いブロック番号の先頭の命令

から

↓

ラダー図のEND命令

まで

END 命令の後に命令が存在していてもかまいませんが、その部分は実行はされません。(ただし使用ステップ数としてはカウントされます)

サブプログラムの実行条件を下表にまとめます。

サブNo.	実行条件
サブ#1	瞬停再始動時(瞬停継続モード(S0400=1)かつ停電時間2秒以内)以外のとき実行
サブ#2	[特殊モード] 特殊モード(S0403=1)かつ瞬停継続モード(S0400=1)で瞬停再始動時(停電時間2秒以内)のとき実行(このときサブ#1は不実行)
	[1回実行モード] 一般モード(S0403=0)かつ1回実行モード(S0405=0)の状態でサブ#2起動要求(S0409=1)のとき実行(S0409はサブ#2起動時に自動リセット)
	[サイクリック実行モード] 一般モード(S0403=0)かつサイクリック実行モード(S0405=1)の状態でサブ#2起動要求(S0409=1)のときSW042に設定されたスキャン回数毎に実行(S0409のリセットはユーザプログラムによる)
サブ#3	[1回実行モード] 1回実行モード(S0406=0)の状態でサブ#3起動要求(S040A=1)のとき実行(S040Aはサブ#3起動時に自動リセット)
	[サイクリック実行モード] サイクリック実行モード(S0406=1)の状態でサブ#3起動要求(S040A=1)のときSW043に設定されたスキャン回数毎に実行(S040Aのリセットはユーザプログラムによる)
サブ#4	[1回実行モード] 1回実行モード(S0407=0)の状態でサブ#4起動要求(S040B=1)のとき実行(S040Bはサブ#4起動時に自動リセット)
	[サイクリック実行モード] サイクリック実行モード(S0407=1)の状態でサブ#4起動要求(S040B=1)のときSW044に設定されたスキャン回数毎に実行(S040Bのリセットはユーザプログラムによる)

補足

サブプログラムは数スキャンにわたって分割実行される場合があります。従って実行中の不必要的入力の変化を避けるために、サブプログラムで使用する入力モジュールについては、i 指定付きの割り付け(一括入出力禁止指定)に変更し、サブプログラムの先頭で直接入出力命令(I/O)によって入力データを読み込むことをお勧めします。

2.3.3

割り込みプログラム

割り込みプログラムとしては、システム情報に設定された周期で周期的に実行される定周期割り込みプログラム(1本)と、割り込み機能付き I/O モジュールからの割り込み信号によって起動される I/O 割り込みプログラム(#1～#8 の8本)の合計9種類があります。

- 定周期割り込みプログラム

システム情報に登録された周期で周期的に実行されます。T3 は 2～1000ms、T3H は 1～1000ms です。周期の登録がない(ブランク)ときは不実行となります。

- I/O 割り込みプログラム (#1～#8)

対応する割り込み機能付き I/O モジュールからの割り込み信号発生によって起動されます。割り込みプログラム番号と割り込み機能付き I/O モジュールとの対応を以下に示します。

補足

- (1) 割り込みプログラムの動作の詳細については第2部 3.3 節をご覧下さい。
- (2) 割り込みプログラムには SFC は使用できません。

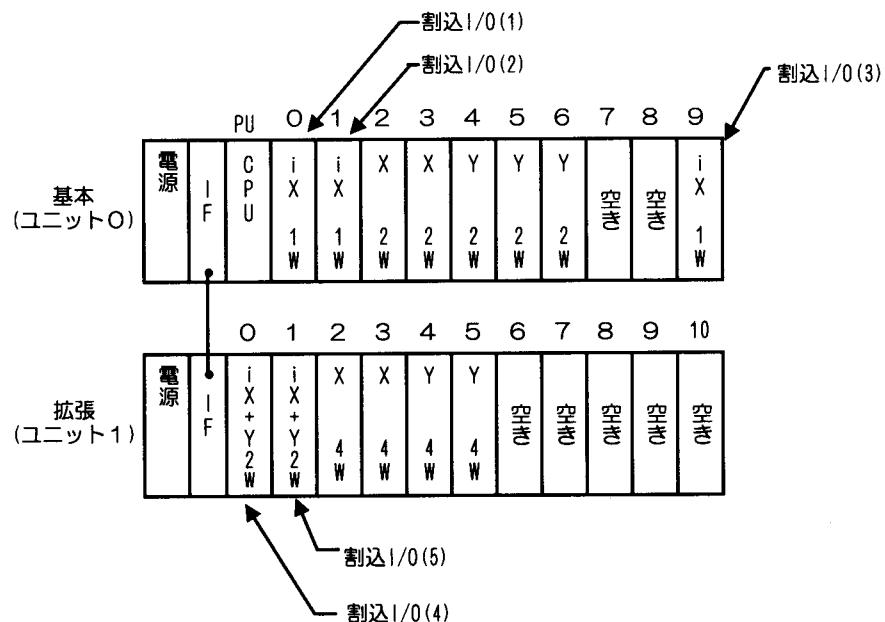
割り込み機能付き I/O モジュール(以下本章では割込 I/O と略します)としては、次のモジュールがあります。

- 2チャネルパルス入力(形式:PI312、モジュール種別:iX+Y2W)
- 状態変化検出付き8点 DC 入力(形式:CD332、モジュール種別:iX1W)

これらの割込 I/O が実装された状態で入出力自動割り付けを行ったとき、割り込みプログラム番号と割込 I/O との対応は、CPU に近い割込 I/O から順に若い番号の I/O 割り込みプログラムが割り付けられます。(次ページの例をご覧下さい)

例)

(1) モジュール実装状態



(2) レジスタ割り付け

ユニットO

スロット	モジュール種別	レジスタ
PU	—	—
O	iX 1W	XW000
1	iX 1W	XW001
2	X 2W	XW002, XW003
3	X 2W	XW004, XW005
4	Y 2W	YW006, YW007
5	Y 2W	YW008, YW009
6	Y 2W	YW010, YW011
7	空き	—
8	空き	—
9	iX 1W	XW012

ユニット1

スロット	モジュール種別	レジスタ
O	iX+Y 2W	XW013, YW014
1	iX+Y 2W	XW015, YW016
2	X 4W	XW017～XW020
3	X 4W	XW021～XW024
4	Y 4W	YW025～YW028
5	Y 4W	YW029～YW032
6	空き	—
7	空き	—
8	空き	—
9	空き	—
10	空き	—

(3) 割り込みプログラムの割り付け

プログラム種別	対応入力レジスタ	対応割込I/O	備考
I/O割り込みプログラム #1	XW000	ユニットOースロットO	割込I/O(1)
I/O割り込みプログラム #2	XW001	ユニットOースロット1	割込I/O(2)
I/O割り込みプログラム #3	XW012	ユニットOースロット9	割込I/O(3)
I/O割り込みプログラム #4	XW013	ユニット1ースロットO	割込I/O(4)
I/O割り込みプログラム #5	XW015	ユニット1ースロット1	割込I/O(5)

一方、このように決まった割り付け状態を変更することができます。システム情報の割り込み割り付け情報にて、割り込みプログラム番号を設定変更し、登録することによって行います。

例)

割り込み割り付け情報(変更前)

割り込みレベル	割り込みプログラムNo.	対応入力レジスタ
0	[1]	XW000
1	[2]	XW001
2	[3]	XW012
3	[4]	XW013
4	[5]	XW015

↓ 次のように変更し登録

割り込み割り付け情報(変更後)

割り込みレベル	割り込みプログラムNo.	対応入力レジスタ
0	[1]	XW000
1	[2]	XW001
2	[3]	XW012
3	[5]	XW013
4	[4]	XW015

この例では、XW013 からの割り込み発生で割り込みプログラム#5 が、XW015 からの割り込み発生で割り込みプログラム#4 が起動されるように、割り付けが変更されます。

補足

システム情報の割り込み割り付け情報を変更することによって、例えば、複数の割込 I/O に同一の割り込みプログラムを割り付けることも可能です。ただし、割り込みの優先度(割り込みレベル)は、ハードウェア的に固定(CPU に近い割込 I/O が優先度が高い)になっていますので、これを変更することはできません。

割り込みプログラムとして認識されるのは次の範囲です。

プログラム種別 “定周期割込” “I/O 割込#n” の各々の

最も若いブロック番号の先頭の命令	から
↓	
ラダー図のIRET命令	まで

補足

割込 I/O からの割り込み信号は、割り込みフラグを読み出すことによってリセットされます。割り込みフラグのビット位置は各割込 I/O によって異なります。I/O 割り込みプログラムの使用方法については、対応する割込 I/O の説明書をご覧下さい。

2.3.4 サブルーチン

プログラム中で、同じ処理を繰り返し行わせる必要がある場合、この処理をサブルーチンとして登録しておき、必要な場所でこのサブルーチンを呼び出して(これをサブルーチンをコールするといいます)実行させることができます。これによって、プログラムステップ数を短縮することができますと共に、機能が整理できてプログラムが見やすくなります。

サブルーチンは他のプログラム種別(メインプログラム、サブプログラム、割り込みプログラム)、及び他のサブルーチンから呼び出すことができます(SFC の実行プログラム部からの呼び出しも可能)。

サブルーチンとしては、プログラム種別 “サブルーチン” 内の、

ラダー図のSUBR命令	から
↓	
ラダー図のRET命令	まで

が、1つのサブルーチンとして登録されます。最大 256 個の登録が可能です。

SUBR 命令(サブルーチンエントリ命令)には、サブルーチン番号を付ける必要があります、有効な番号は0から 255 の範囲です。

-[SUBR (000)]-

↑

サブルーチン番号

RET 命令(サブルーチンリターン命令)にはサブルーチン番号はありません。

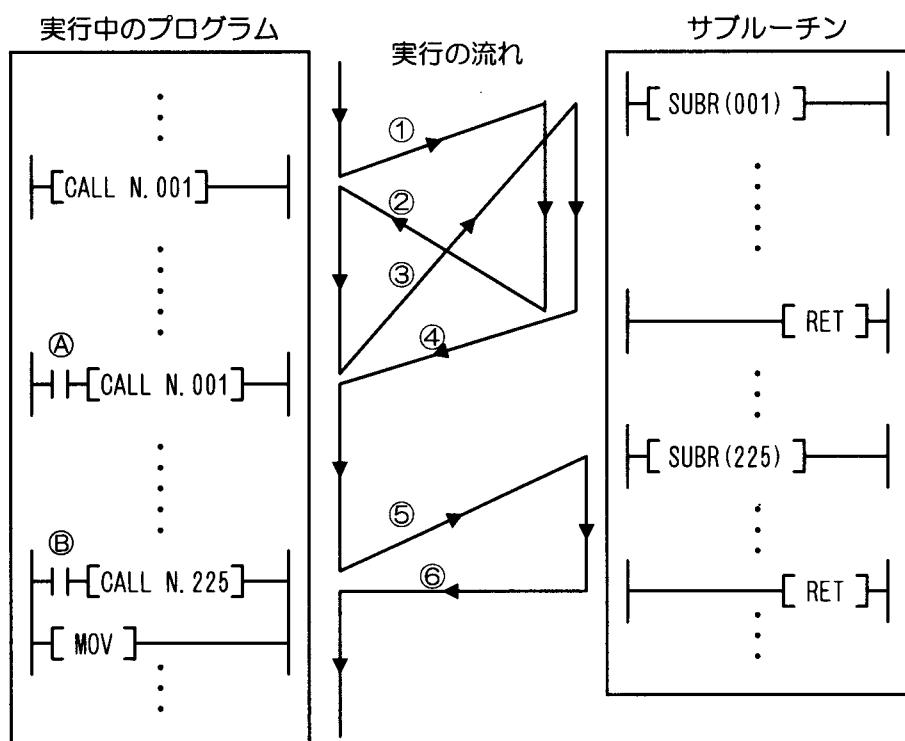
登録されたサブルーチンを呼び出す命令は、ラダー図の CALL 命令(サブルーチンコール命令)です。CALL 命令には呼び出すサブルーチン番号が必要です。

—[CALL N. 000]—

サブルーチン番号

サブルーチンを含む場合の実行順序を下図に示します。

2



- ① サブルーチン 001 の CALL 命令実行により、サブルーチン 001 に実行が移る
- ② RET 命令まで進むと、①の CALL 命令の次の命令に実行が移る
- ③ デバイスⒶが ON のとき CALL 命令が実行され、サブルーチン 001 に実行が移る
- ④ RET 命令まで進むと、③の CALL 命令の次の命令に実行が移る
- ⑤ デバイスⒷが ON のとき CALL 命令が実行され、サブルーチン 255 に実行が移る
- ⑥ RET 命令まで進むと、⑤の CALL 命令の次の命令(この例では MOV 命令)に実行が移る

補足

- (1) サブルーチンには SFC は使用できません。
- (2) サブルーチンから他のサブルーチンの呼び出し(ネスティング)は 6 重までです。
- (3) サブルーチンの実行中に同一のサブルーチンが呼び出されるような場合は動作が異常となりますので、このようなことがないように注意して下さい。
- メイン/サブプログラムでサブルーチン実行中に割り込みが発生し、割り込みの中で同一のサブルーチンを実行した場合
 - サブプログラムでサブルーチン実行中に時間制限で実行が中断され、次のメインプログラムの実行の中で同一のサブルーチンを実行した場合

2.4 コメント

ユーザプログラムを見やすくし、メンテナンスを容易にするために、プログラムにコメントを付け、PC 本体(ユーザプログラムメモリの残り領域)に格納しておくことができます。(T3 ではバージョン 1.2 以降)

PC 本体に格納できるコメントの種類としては、デバイス/レジスタ、SFC ステップ番号に対する、ネーム及びコメントがあります。

ネーム……………半角英数字 5 文字(全角文字も可能)

コメント……………半角英数字 20 文字(全角文字も可能)

PC 本体内のコメント格納領域の容量は、ユーザプログラム総容量(32K ステップもしくは 64K ステップ)からシステム情報に設定したプログラム容量を引いた残り分となります。

ネームとコメントのペアを 1 個として、格納できる個数は次式で決まります。

32K ステップの場合: $(1024 \times (32-N) - 38) / 10$ (個)

↑
システム情報に設定したプログラム容量
↓

64K ステップの場合: $(1024 \times (64-N) - 38) / 10$ (個)

補足

コメントの格納先としては、ここで説明したように PC 本体のユーザプログラムメモリを使用する場合の他に、外部コメントファイルとして FD 等に保存することもできます。コメントを作成する場合には、どの格納先を対象とするのかを確認してから行って下さい。

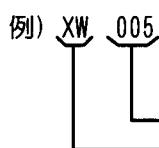
(コメント作成方法、外部コメントファイルの指定方法についての別冊のプログラマ説明書をご覧下さい)

3.1

ユーザデータの概要

ユーザプログラムで使用する、外部入出力データ、タイマ命令やカウンタ命令の現在値、及びデータ処理を行うための変数の値を格納する領域をユーザデータ領域と総称します。

ユーザデータは、機能を表す英記号(これを機能種別と呼びます)と、0から始まる一連の番号(これをアドレスと呼びます)の組み合わせによって、データの格納場所を表します。

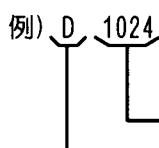


アドレス005(この場合はレジスタアドレス)
機能種別 XW=入力レジスタ

XW005の内容が100ということは、XW005が示すユーザデータメモリ上の場所に100という数値が格納されているということ。

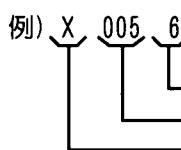
また、ユーザデータは、格納するデータの種類によって、レジスタとデバイスに分けられます。(リレーという表現も用いますが、リレーはデバイスの1種として考えて下さい)

レジスタとは、16ビットのデータ(正の整数とすれば、0から65535までの数値を表すことができる)を格納する領域で、機能種別とレジスタアドレスの組み合わせで表現されます。(レジスタアドレスは10進数)



レジスタアドレス(10進数)
機能種別 D=データレジスタ

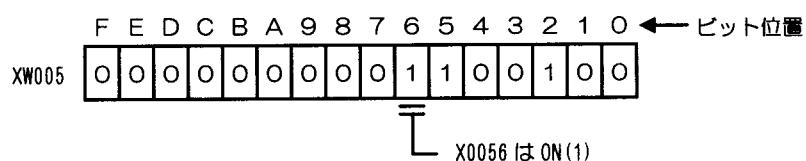
一方、デバイスとは、1ビットのデータ(1か0か、つまりONかOFFかを表す)を格納する領域で、機能種別とデバイスアドレスの組み合わせで表現されます。ただし、デバイスは、独立したメモリ領域を使用する訳ではなく、対応するレジスタの16ビット中の1ビットとして割り当てられています。従って、デバイスアドレスは、対応するレジスタアドレス+ビット位置という形で表現されます。



ビット位置(0~Fの16位置)
レジスタアドレス(10進数)
機能種別 X=入力デバイス(入力レジスタ XW に対応)

レジスタデータとデバイスデータの対応は次のように考えます。

例) XW005 の内容が 100 ということは、10 進数の 100 を2進数で表すと、1100100 となりますので、XW005 の各ビットは次のようになっていることを示しています。



このとき、XW005 のビット 6 に対応するデバイス X0056 のデータは、1、つまり X0056 は ON ということになります。

レジスタとデバイスの対応を機能種別で示します。

3

- 入力デバイス(X) …… 入力レジスタ(XW)の1ビットに対応
- 出力デバイス(Y) …… 出力レジスタ(YW)の1ビットに対応
- 補助リレー(R) …… 補助レジスタ(RW)の1ビットに対応
- 特殊リレー(S) …… 特殊レジスタ(SW)の1ビットに対応
- リンクレジスラリレー(Z) …… リンクレジスタ(W)の1ビットに対応
- リンクリレー(L) …… リンクリレーレジスタ(LW)の1ビットに対応

他のデバイス、I, O, T., C. については、扱いが多少異なります。詳しくは 3.2 節で説明します。

レジスタとデバイスの種別とアドレス範囲を下表に示します。機能及び使用方法については、3.2 節で説明します。

機能種別	種別記号	T3		T3H		表現例
		アドレス範囲	個 数	アドレス範囲	個 数	
入力レジスタ	XW	000～255	合計 256 ワード	000～511	合計 512 ワード	XW001
出力レジスタ	YW					YW034
直接入力レジスタ	IW					IW001
直接出力レジスタ	OW					OW034
入力デバイス	X	0000～255F	合計4096点	0000～511F	合計 8192 点	X001A
出力デバイス	Y					Y0348
直接入力デバイス	I					I0012
直接出力デバイス	O					O0340
補助レジスタ	RW	000～511	512 ワード	000～999	1000 ワード	RW414
補助リレー	R	0000～511F	8192 点	0000～999F	16000 点	R4141
特殊レジスタ	SW	000～255	256 ワード	000～255	256 ワード	SW014
特殊リレー	S	0000～255F	4096 点	0000～255F	4096 点	S0140
タイマレジスタ	T	000～511	512 ワード	000～999	1000 ワード	T030
タイマリレー	T.	000～511	512 点	000～999	1000 点	T.030
カウンタレジスタ	C	000～511	512 ワード	000～511	512 ワード	C299
カウンタリレー	C.	000～511	512 点	000～511	512 点	C.299
データレジスタ	D	0000～8191	8192 ワード	0000～8191	8192 ワード	D7635
リンクレジスタ	W	0000～1023	1024 ワード	0000～2047	2048 ワード	W0200
リンクレジスタリレー	Z	0000～511F	8192 点	0000～999F	16000 点	Z2001
リンクリレー レジスタ	LW	000～255	256 ワード	000～255	256 ワード	LW123
リンクリレー	L	0000～255F	4096 点	0000～255F	4096 点	L123F
ファイルレジスタ	F	0000～8191	8192 ワード	00000～ 32767	32768 ワード	F5000
インデックス レジスタ	I	なし	1ワード	なし	1ワード	I
	J	なし	1ワード	なし	1ワード	J
	K	なし	1ワード	なし	1ワード	K

*) PC では 1ワード=16 ビットとして扱い、レジスタ個数はワード単位で数えます。

3.2 レジスタとデバイス

レジスタとデバイスの各々の機能種別について、機能とアドレス範囲を以下に説明します。

入力レジスタと
入力デバイス

記 号		入力レジスタ … XW 入力デバイス … X	
ア ド レ ス	T3	入力レジスタ … 000～255(256 ワード) 入力デバイス … 0000～255F(4096 点)	出力レジスタ/ 出力デバイスと共に用
	T3H	入力レジスタ … 000～511(512 ワード) 入力デバイス … 0000～511F(8192 点)	
機 能		入出力割り付けを行うことによって、レジスタ単位(ワード単位)で、入力モジュールに割り付けられます。入力モジュールに入力された信号状態は、一括入出力のタイミングで対応する入力レジスタに格納されます(割り付け上；指定が付いているモジュールを除く)。入力デバイスは、対応する入力レジスタの1ビットを示します。 入力レジスタ/入力デバイスのデータは基本的に1スキャン中は変化しません。ただし、直接入出力命令を実行したときには、命令実行時に対応する入力モジュールからデータを読み込んで入力レジスタ/入力デバイス(XW/X)に格納しますので、スキャンの中でデータが変化する場合があります。	

出力レジスタと
出力デバイス

記 号		出力レジスタ … YW 出力デバイス … Y	
ア ド レ ス	T3	出力レジスタ … 000～255(256 ワード) 出力デバイス … 0000～255F(4096 点)	出力レジスタ/ 出力デバイスと共に用
	T3H	入力レジスタ … 000～511(512 ワード) 入力デバイス … 0000～511F(8192 点)	
機 能		入出力割り付けを行うことによって、レジスタ単位(ワード単位)で、出力モジュールに割り付けられます。出力レジスタに格納されたデータは、一括入出力のタイミングで対応する出力モジュールに書き込まれ、出力モジュールの出力信号の状態が決定されます(割り付け上；指定が付いているモジュールを除く)。出力デバイスは出力レジスタの1ビットを示します。	

直接入力レジスタと
直接入力デバイス

記 号	直接入力レジスタ … IW 直接入力デバイス … I
ア ド レ ス	T3 直接入力レジスタ … 000~255(入力レジスタ(XW)に対応) 直接入力デバイス … 0000~255F(入力デバイス(X)に対応)
	T3H 直接入力レジスタ … 000~511(入力レジスタ(XW)に対応) 直接入力デバイス … 0000~511F(入力デバイス(X)に対応)
機 能	<p>直接入力レジスタ/直接入力デバイスは、これ自体が特定のメモリを指し示す訳ではなく、このレジスタ/デバイスを使用した命令語が実行されたときに、アドレスに対応する入力モジュールから直接データを読み込む動作となります。一括入出力方式(リフレッシュ方式)ではなく、直接入出力方式(ダイレクト方式)でPCを使用するときに、このレジスタ/デバイスを使用します。</p> <p>(例)</p>  <p>10000 10000 の a 接点命令</p> <p>命令実行時に、XW000 に対応する入力モジュールから X0000 に対応するビットデータを読み出し、そのデータによって命令を実行します。 (X0000 のデータは影響を受けません)</p> <p>—[IW005 MOV RW100]— IW005 から RW100 への転送命令</p> <p>命令実行時に、XW005 に対応する入力モジュールから XW005 に対応するワードデータを読み出し、RW100 に代入します。 (XW005 のデータは影響を受けません)</p>

直接出力レジスタと
直接出力デバイス

記 号	直接出力レジスタ … OW 直接出力デバイス … O
ア ド レ ス	T3 直接出力レジスタ … 000~255(出力レジスタ(YW)に対応) 直接出力デバイス … 0000~255F(出力デバイス(Y)に対応)
	T3H 直接出力レジスタ … 000~511(出力レジスタ(YW)に対応) 直接出力デバイス … 0000~511F(出力デバイス(Y)に対応)
機 能	<p>直接出力レジスタ/直接出力デバイスを使用した命令語が実行されたとき、対応する出力レジスタ/出力デバイス(YW/Y)にデータを格納し、さらにこの出力レジスタ(YW)のデータを対応する出力モジュールに直接書き込みます。一括入出力方式(リフレッシュ方式)ではなく、直接入出力方式(ダイレクト方式)でPCを使用するときに、このレジスタ/デバイスを使用します。</p> <p>(例)</p>  <p>00020 00020 のコイル命令</p> <p>命令実行時に、直前の信号状態に応じたデータ(ON/OFF データ)を Y0020 に格納し、さらにこのときの YW002 のデータを対応する出力モジュールに書き込みます。</p>

補助レジスタと
補助リレー

記 号	補助レジスタ … RW 補助リレー … R
ア ド レ ス	T3 出力レジスタ … 000~511(512 ワード) 出力デバイス … 0000~511F(レジスタ内の1ビットに対応、8192 点)
	T3H 出力レジスタ … 000~999(1000 ワード) 出力デバイス … 0000~999F(レジスタ内の1ビットに対応、16000 点)
機 能	プログラムの実行途中結果の一時記憶などに使用できる汎用レジスタ/汎用デバイスです。補助レジスタは 16 ビットデータの格納用に使用します。補助リレーは補助レジスタ内の1ビットを示します。 補助レジスタ/リレーは停電保持領域の指定ができます。

特殊レジスタと
特殊リレー

記号	特殊レジスタ … SW 特殊リレー … S	
ア ド レ ス	T3	特殊レジスタ … 000～255(256ワード) 特殊リレー … 0000～255F(レジスタ内の1ビットに対応、4096点)
	T3H	特殊レジスタ … 000～255(256ワード) 特殊リレー … 0000～255F(レジスタ内の1ビットに対応、4096点)
機能	CPUが異常を検出したときにセットする異常クラブ(エラーダウン/警告)、CPUが更新するタイミングリレーやカレンダタイマのデータ(年、月、日、時、分、秒、曜日)、サブプログラムの動作制御を行うためにユーザが設定するフラグ/データなどの特殊機能を持ったレジスタ/デバイスです。詳細は159ページの一覧表をご覧下さい。	

タイマレジスタと
タイマリレー

記号	タイマレジスタ … T タイマリレー … T.	
ア ド レ ス	T3	タイマレジスタ … 000～511(512ワード) タイマリレー … 000～511(512点)
	T3H	タイマレジスタ … 000～999(1000ワード) タイマリレー … 000～999(1000点)
機能	タイマレジスタはタイマ命令(TON, TOF, SS, TRG)と共に使用し、タイマ動作時に経過時間(加算式)が格納されます。また、タイマリレーは同じアドレスのタイマレジスタの動作に連動し、タイマ命令の出力結果が格納されます。タイマレジスタは停電保持領域の指定ができます。 なお、T3においてタイマアドレス 000～063 は 0.01 秒タイマとして、タイマアドレス 064～511 は 0.1 秒タイマとして動作します。 又、T3Hにおいては 0.01 秒タイマの範囲がタイマアドレス 000～任意のアドレスまで設定可能になっています。	

カウンタレジスタと
カウンタリレー

記号	カウンタレジスタ … C カウンタリレー … C.	
ア ド レ ス	T3	カウンタレジスタ … 000～511 (512ワード) カウンタリレー … 000～511 (512点)
	T3H	カウンタレジスタ … 000～511 (512ワード) カウンタリレー … 000～511 (512点)
機能	カウンタレジスタはカウンタ命令(CNT, U/D)と共に使用し、カウンタ動作時にカウント現在値が格納されます。また、カウンタリレーは同じアドレスのカウンタレジスタの動作に連動し、カウンタ命令の出力結果が格納されます。カウンタレジスタは停電保持領域の指定ができます。	

データレジスタ

記号	D
ア ド レ ス	T3
	T3H
機能	演算結果の一時記憶、制御パラメータの格納などに使用できる汎用レジスタです。ビット指定ができない点以外は補助レジスタと同様の使用ができます。 データレジスタは停電保持領域の指定ができます。また周辺メモリを使用した場合にはD0000からD4095はイニシャルロードの対象となります。なお、メモリプロジェクト状態(P-RUN)ではD0000～D4095は周辺装置からのデータ書き込みが禁止されます。

リンクレジスタと
リンクレジスタリレー

記号	リンクレジスタ W リンクレジスタリレー Z
アドレス	T3 リンクレジスタ 000~1023(1024ワード) リンクレジスタリレー 0000~511F(レジスタの先頭512ワードのビットに 対応、8192点)
	T3H リンクレジスタ 000~2047(2048ワード) リンクレジスタリレー 0000~999F(レジスタの先頭1000ワードの1ビット に対応、16000点)
機能	データ伝送装置 TOSLINE-S20 によるデータリンクに使用します。T3 ではリンク レジスタの先頭 512 ワード (W0000~W0511) について、T3H では先頭 1000 ワード (W0000~W0999) について、リンクレジスタリレーとしてビット指定が可能で す。(T3:Z0000~Z511F, T3H:Z0000~Z999F) TOSLINE-S20 を使用しないとき、及び使用していても TOSLINE-S20 に割り付けて いない領域については、補助レジスタやデータレジスタと同様の使用ができます。

リンクリレーレジスタと
リンクリレー

記号	リンクリレーレジスタ LW リンクリレー L
アドレス	T3 リンクリレーレジスタ 000~255(256 ワード) リンクリレー 000~255F(レジスタ内の1ビットに対応 4096 点)
	T3H リンクリレーレジスタ 000~255(256 ワード) リンクリレー 000~255F(レジスタ内の1ビットに対応 4096 点)
機能	データ伝送装置 TOSLINE-F10 によるリモート I/O 構成用のレジスタ/リレーと して使用します。リンクリレーは、リンクリレーレジスタ内の1ビットに対応 します。 TOSLINE-F10 を使用しないとき、及び使用していても TOSLINE-F10 に割り付け ていない領域については、補助レジスタ/リレーと同様の使用ができます。

ファイルレジスタ

記号	F
アドレス	T3 000~8191(8192 ワード)
	T3H 000~32767(32768 ワード)
機能	制御パラメータの格納、フィールド収集データの格納など、データレジスタと 同様の使用ができます。ビット指定はできません。ファイルレジスタは全領域 停電保持されます。 I _C メモリカードを装着したときには、拡張データ転送命令 (XFER) によりファ イルレジスタの拡張が可能です。(I _C メモリカード内 122880 ワード) なお、T3Hにおいて F10000~F32767 をプログラム中で使用する場合、インデッ クスレジスタ (I, J, K) によってアドレスを指定します。

インデックスレジスタ

記号	I, J, K(3種類、3ワード)
	T3, T3H共にアドレスなし
機能	命令語でレジスタ (インデックスレジスタ以外) を使用する場合、通常のアレ ス指定方法 (直接アドレス指定、例えば D0100) の他に、インデックスレジスタ を用いて間接的に指定すること (間接アドレス指定、例えば D0100・I) が可能 です。(D0100・I は、I の内容が例えば 5なら、D0105を示します) 間接アドレス指定については、3.4節をご覧下さい。

特殊レジスタ/特殊リレーの一覧を以下に示します。

特殊レジスタ全体マップ

レジスタ	内 容
SW000	本体動作モード、異常フラグ、警告フラグ
SW001	CPU 異常関連フラグ
SW002	I/O 異常関連フラグ
SW003	プログラム異常関連フラグ、I/O メモリカードステータス
SW004	タイミングリレー
SW005	キャリーフラグ、エラーフラグ
SW006	プログラム実行中エラー関連フラグ
SW007 ↓ SW013	カレンダタイマデータ (年、月、日、時、分、秒、曜日)
SW014	ビットパターン/レジスタ値チェック結果(ユーザ診断)
SW015	I/O エラーマップ作成コマンド
SW016 ↓ SW033	診断表示登録(ユーザ診断)
SW034 ↓ SW037	アンシェータリレー(ユーザ診断)
SW038	リザーブ(将来用)
SW039	割り込みプログラム実行ステータス
SW040	サブプログラム実行制御コマンドフラグ
SW041	サブプログラム実行ステータス
SW042 ↓ SW044	サブプログラム実行モード設定
SW045	瞬停時間設定
SW046 ↓ SW052	I/O エラーマップ
SW053 ↓ SW060	リザーブ(将来用)
SW061	周辺サポート設定(T3H ver1.2 以降でサポート)
SW062	サンプリングトレース データコピーフラグ
SW063	EN311 CH1 ステーションステータス(T3H ver1.13 以降でサポート)
SW064	EN311 CH2 ステーションステータス(T3H ver1.13 以降でサポート)
SW065	EN311 CH3 ステーションステータス(T3H ver1.13 以降でサポート)
SW066	EN311 CH4 ステーションステータス(T3H ver1.13 以降でサポート)
SW067	PCリンク レジスタライトプロテクト (T3H ver1.13 以降でサポート)
SW068 ↓ SW077	リザーブ(将来用)

特殊レジスタ全体マップ(続き)

レジスタ	内 容
SW078 ↓ SW093	TOSLINE-F10 コマンド/ステータス
SW094 ↓ SW109	TOSLINE-F10 スキャンエラーマップ
SW110	TOSLINE-S20 CH1 ステーションステータス
SW111	TOSLINE-S20 CH2 ステーションステータス
SW112 ↓ SW115	TOSLINE-S20 CH1 オンラインマップ
SW116 ↓ SW119	TOSLINE-S20 CH2 オンラインマップ
SW120 ↓ SW123	TOSLINE-S20 CH1 スタンバイマップ
SW124 ↓ SW127	TOSLINE-S20 CH2 スタンバイマップ
SW128 ↓ SW255	TOSLINE-S20 スキャンヘルシーマップ

特殊リレー	名 称	機 能
S0000	本体動作モード	0:初期化中 1:HALT モード 2:RUN モード 3:RUN-F モード 4:HOLD モード 6:ERROR モード
S0001		9:D-HALT A:D-RUN B:D-STOP D:S-HALT E:S-RUN F:S-STOP
S0002		
S0003		
S0004	CPU 異常(ダウン)	異常発生時 ON (SW001 内の関連フラグの OR 条件)
S0005	I/O 異常(ダウン)	異常発生時 ON (SW002 内の関連フラグの OR 条件)
S0006	プログラム異常(ダウン)	異常発生時 ON (SW003 内の関連フラグの OR 条件)
S0007	EEPROM 書込回数オーバー(警告)	EEPROM 書込回数 10 万回超過時 ON (運転継続)
S0008	定刻スキャン渋滞(警告)	定刻スキャン渋滞発生時 ON (運転継続)
S0009	I/O 異常(警告)	I/O エラーマップ作成機能にて I/O 異常検出時 ON
S000A	カレンダ LSI 異常(警告)	カレンダタイマデータ異常時 ON (運転継続)
S000B	EN311 異常(警告)	EN311 異常時 ON (運転継続)
S000C		リザーブ(将来用)
S000D	TOSLINE-F10 異常(警告)	TOSLINE-F10 異常時 ON (運転継続)
S000E	TOSLINE-S20 異常(警告)	TOSLINE-S20 異常時 ON (運転継続)
S000F	バッテリ電圧異常(警告)	バッテリ電圧低下時 ON (運転継続)
S0010	システム ROM 異常(ダウン)	システム ROM 異常時 ON
S0011	システム RAM 異常(ダウン)	システム RAM 異常時 ON
S0012	プログラムメモリ異常(ダウン)	プログラムメモリ (RAM) 異常時 ON
S0013	EEPROM 異常(ダウン)	EEPROM 異常時 ON
S0014	IC メモリカード異常(ダウン)	IC メモリカード異常時 ON
S0015	LP 異常(ダウン)	シーケンス演算プロセッサ (LP) 異常時 ON
S0016	メイン CPU 異常(ダウン)	メイン CPU 異常時 ON
S0017		リザーブ(将来用)
S0018		
S0019		
S001A		
S001B		
S001C	瞬停復帰	
S001D	瞬停検出	
S001E		
S001F	ウォッチドッグタイマ異常(ダウン)	ウォッチドッグタイマエラー発生時 ON

*1) 本エリアは参照のみ(書き込みは行わないこと)

*2) 異常フラグは要因発生にて ON となり保持 (RUN 起動時にリセット)

特殊リレー	名 称	機 能
S0020	I/O バス異常(ダウン)	I/O バス異常時 ON
S0021	I/O 照合異常(ダウン)	I/O 照合異常(割り付け情報と実装状態が不一致)時 ON
S0022	I/O 応答異常(ダウン)	I/O 無応答時 ON
S0023	I/O パリティ異常(ダウン)	I/O データのパリティエラー発生時 ON
S0024	拡張 I/O 電源異常(ダウン)	拡張ユニット側 5V 電源異常時 ON
S0025		リザーブ(将来用)
S0026		
S0027		
S0028		
S0029		
S002A		
S002B		
S002C		
S002D		
S002E		
S002F		
S0030	プログラム異常	プログラム異常発生時 ON *3)
S0031	スキャンタイム異常(ダウン)	スキャン周期が設定値を超えた時に ON
S0032		リザーブ(将来用)
S0033		
S0034		
S0035		
S0036		
S0037		
S0038		
S0039		
S003A		
S003B		
S003C		
S003D	IC メモリカード実装ステータス	IC メモリカード装着時 ON
S003E	IC メモリカード書き込みプロテクト	書き込みプロテクト状態のとき ON
S003F	IC メモリカードバッテリ異常(警告)	IC メモリカードに内蔵のバッテリ電圧低下時 ON

*1) 本エリアは参照のみ(書き込みは行わないこと)

*2) 異常フラグは要因発生にて ON となり保持(RUN 起動時にリセット)

*3) キーSWによるRUN起動時のプログラムチェック異常(JUMP命令, CALL命令, END命令, IRET命令, JCS/JCR命令, MCS/MCR命令チェック)

特殊リレー	名 称	機 能	
S0040	タイミングリレー0.1s	0.05秒 OFF/0.05秒 ON(周期0.1秒) 0.1秒 OFF/0.1秒 ON(周期0.2秒) 0.2秒 OFF/0.2秒 ON(周期0.4秒) 0.4秒 OFF/0.4秒 ON(周期0.8秒) 0.5秒 OFF/0.5秒 ON(周期1.0秒) 1.0秒 OFF/1.0秒 ON(周期2.0秒) 2.0秒 OFF/2.0秒 ON(周期4.0秒) 4.0秒 OFF/4.0秒 ON(周期8.0秒)	
S0041	タイミングリレー0.2s		
S0042	タイミングリレー0.4s		
S0043	タイミングリレー0.8s		
S0044	タイミングリレー1.0s		
S0045	タイミングリレー2.0s		
S0046	タイミングリレー4.0s		
S0047	タイミングリレー8.0s		
S0048	リザーブ(将来用)		
S0049			
S004A			
S004B			
S004C			
S004D			
S004E	常時 OFF	常に OFF	
S004F	常時 ON	常に ON	
S0050	CF(キャリーフラグ)	キャリー付き命令にて使用	
S0051	ERF(エラーフラグ)	命令実行時にエラー発生にて ON(SW006 の各フラグに連動)	
S0052	リザーブ(将来用)		
S0053			
S0054			
S0055			
S0056			
S0057			
S0058			
S0059			
S005A			
S005B			
S005C			
S005D			
S005E			
S005F			

*) 本エリア(S0050, S0051 を除く)は参照のみ(書き込みは無効)

特殊リレー	名 称	機 能
S0060	イリーガル命令検出(ダウン)	不正命令検出時ON
S0061		
S0062		リザーブ(将来用)
S0063		
S0064	バウンダリエラー(警告)	間接アドレス指定にてアドレス範囲オーバーのときON(運転継続)
S0065	アドレスバウンダリエラー(警告)	CALL命令、JUMP命令にて飛び先ラベル異常時ON(運転継続)
S0066	メモリパリティエラー(ダウン)	ユーザプログラムメモリにパリティエラー発生時ON
S0067	メモリパリティエラー(ダウン)	ユーザデータメモリにパリティエラー発生時ON
S0068	除算エラー(警告)	除算命令にてエラー発生時ON(運転継続)
S0069	BCDデータエラー(警告)	BCD命令にて異常データ検出時ON(運転継続)
S006A	テーブルオペレーションエラー(警告)	テーブル操作命令にてテーブル範囲オーバーのときON(運転継続)
S006B	エンコードエラー(警告)	エンコード命令にてエラー発生時ON(運転継続)
S006C	アドレス登録エラー(警告)	CALL命令、JUMP命令にて飛び先未登録のときON(運転継続)
S006D	ネスティングエラー(警告)	CALL命令、FOR命令、MCSn命令にてネスティングオーバー時ON(運転継続)
S006E		リザーブ(将来用)
S006F		

*1) エラー要因発生にて ON となり保持(RUN 起動時にリセット)

*2) 警告フラグのリセットは必要に応じてユーザプログラムで行う。

特殊 レジスタ	名 称	機 能	
SW007	カレンダデータ(年)	西暦年下位2桁(91, 92, …)	下位8ビットに BCD コードで格納
SW008	カレンダデータ(月)	月(01~12)	
SW009	カレンダデータ(日)	日(01~31)	
SW010	カレンダデータ(時)	時(00~23)	
SW011	カレンダデータ(分)	分(00~59)	
SW012	カレンダデータ(秒)	秒(00~59)	
SW013	カレンダデータ(曜日)	曜日(日=00, 月 = 01, …土=06)	

*1) カレンダデータの設定は、カレンダ設定命令またはプログラマによるシステム情報設定によって行う。(特殊レジスタに直接データを書き込んでも無効)

*2) カレンダ LSI 異常等によりデータが正常に読み込めないときは、これらのレジスタは H00FF となる。

*3) カレンダの精度は±30 秒/月

特殊リレー	名 称	機 能
S0140	ピット/レジスタチェック許可	ONでチェック実行
S0141	ピット/レジスタチェック結果	S0142～S0146のいずれかONのときON
S0142	ピットパターンチェック結果	登録デバイスの状態が登録パターンに不一致のときON
S0143	レジスタ値チェック結果(1)	登録レジスタ1の値が登録範囲外のときON
S0144	レジスタ値チェック結果(2)	登録レジスタ2の値が登録範囲外のときON
S0145	レジスタ値チェック結果(3)	登録レジスタ3の値が登録範囲外のときON
S0146	レジスタ値チェック結果(4)	登録レジスタ4の値が登録範囲外のときON
S0147		リザーブ(将来用)
S0148		
S0149		
S014A		
S014B		
S014C		
S014D		
S014E		
S014F		
S0150	I/Oエラーマップ作成	ONでI/Oエラーマップ(異常I/O位置登録)作成
S0151		リザーブ(将来用)
S0152		
S0153		
S0154		
S0155		
S0156		
S0157		
S0158		
S0159		
S015A		
S015B		
S015C		
S015D		
S015E		
S015F	ピット/レジスタチェック 実行タイミング	OFF:プログラム実行前 ON:プログラム実行後

特殊レジスタ	名 称	機 能
SW016	先頭エラーコード	
SW017	登録個数	
SW018	エラーコード(先頭)	• 診断表示命令の実行により、指定のエラーコード(1~64)が SW018~SW033 に実行順(早いものが若いアドレス)に格納され、登録個数(SW017)が更新される。
SW019	エラーコード(2)	
SW020	エラーコード(3)	
SW021	エラーコード(4)	• 先頭エラーコード(SW016)には、登録されているエラーコードの中で最も早く発生したもの(SW018 の内容)が格納される。
SW022	エラーコード(5)	
SW023	エラーコード(6)	• 診断表示リセット命令の実行、またはプログラマによるリセット操作によって、登録エラーコードは1つづつ削除される。このとき、登録個数は1減少し、エラーコードの格納位置はシフトアップされる。
SW024	エラーコード(7)	
SW025	エラーコード(8)	
SW026	エラーコード(9)	
SW027	エラーコード(10)	
SW028	エラーコード(11)	
SW029	エラーコード(12)	
SW030	エラーコード(13)	
SW031	エラーコード(14)	
SW032	エラーコード(15)	
SW033	エラーコード(16)	

特殊リレー	名 称	機 能
S0340	アンシェータリレー1	
S0341	アンシェータリレー2	
S0342	アンシェータリレー3	• SW018~SW033 に登録されているエラーコードに対応するアンシェータリレーがONとなる。
S0343	アンシェータリレー4	
S0344	アンシェータリレー5	
S0345	アンシェータリレー6	
S0346	アンシェータリレー7	
S0347	アンシェータリレー8	
S0348	アンシェータリレー9	
S0349	アンシェータリレー10	
S034A	アンシェータリレー11	
S034B	アンシェータリレー12	
S034C	アンシェータリレー13	
S034D	アンシェータリレー14	
S034E	アンシェータリレー15	
S034F	アンシェータリレー16	

特殊リレー	名 称	機 能
S0350	アンシェータリレー17	
S0351	アンシェータリレー18	
S0352	アンシェータリレー19	
S0353	アンシェータリレー20	
S0354	アンシェータリレー21	
S0355	アンシェータリレー22	
S0356	アンシェータリレー23	
S0357	アンシェータリレー24	
S0358	アンシェータリレー25	
S0359	アンシェータリレー26	
S035A	アンシェータリレー27	
S035B	アンシェータリレー28	
S035C	アンシェータリレー29	
S035D	アンシェータリレー30	
S035E	アンシェータリレー31	
S035F	アンシェータリレー32	
S0360	アンシェータリレー33	
S0361	アンシェータリレー34	
S0362	アンシェータリレー35	
S0363	アンシェータリレー36	
S0364	アンシェータリレー37	
S0365	アンシェータリレー38	
S0366	アンシェータリレー39	
S0367	アンシェータリレー40	
S0368	アンシェータリレー41	
S0369	アンシェータリレー42	
S036A	アンシェータリレー43	
S036B	アンシェータリレー44	
S036C	アンシェータリレー45	
S036D	アンシェータリレー46	
S036E	アンシェータリレー47	
S036F	アンシェータリレー48	

特殊リレー	名 称	機 能
S0370	アンシェータリレー-49	• SW018～SW033 に登録されているエラーコードに対応するアンシェータリレーがONとなる。
S0371	アンシェータリレー-50	
S0372	アンシェータリレー-51	
S0373	アンシェータリレー-52	
S0374	アンシェータリレー-53	
S0375	アンシェータリレー-54	
S0376	アンシェータリレー-55	
S0377	アンシェータリレー-56	
S0378	アンシェータリレー-57	
S0379	アンシェータリレー-58	
S037A	アンシェータリレー-59	
S037B	アンシェータリレー-60	
S037C	アンシェータリレー-61	
S037D	アンシェータリレー-62	
S037E	アンシェータリレー-63	
S037F	アンシェータリレー-64	

特殊リレー	名 称	機 能
S0390	定周期割り込み実行ステータス	実行中 ON
S0391	I/O 割り込み#1 実行ステータス	実行中 ON
S0392	I/O 割り込み#2 実行ステータス	実行中 ON
S0393	I/O 割り込み#3 実行ステータス	実行中 ON
S0394	I/O 割り込み#4 実行ステータス	実行中 ON
S0395	I/O 割り込み#5 実行ステータス	実行中 ON
S0396	I/O 割り込み#6 実行ステータス	実行中 ON
S0397	I/O 割り込み#7 実行ステータス	実行中 ON
S0398	I/O 割り込み#8 実行ステータス	実行中 ON
S0399	リザーブ(将来用)	
S039A		
S039B		
S039C		
S039D		
S039E		
S039F		

特殊リレー	名 称	機 能
S0400	瞬停処理選択	継続モード選択時 ON に設定する
S0401	HOLD デバイス	HOLD モード中 ON (プログラムで ON にしても HOLD に遷移)
S0402		リザーブ(将来用)
S0403	サブプログラムモード	サブプログラム#2 のモードを設定 (OFF:一般モード ON:特殊モード)
S0404		リザーブ(将来用)
S0405	サブプログラム#2 実行モード	サブプログラム#2 の実行モード設定 (OFF:1回実行 ON:サイクリック実行)
S0406	サブプログラム#3 実行モード	サブプログラム#3 の実行モード設定 (OFF:1回実行 ON:サイクリック実行)
S0407	サブプログラム#4 実行モード	サブプログラム#4 の実行モード設定 (OFF:1回実行 ON:サイクリック実行)
S0408		リザーブ(将来用)
S0409	サブプログラム#2 起動要求	サブプログラム#2 の起動要求コマンド (ON で起動要求)
S040A	サブプログラム#3 起動要求	サブプログラム#3 の起動要求コマンド (ON で起動要求)
S040B	サブプログラム#4 起動要求	サブプログラム#4 の起動要求コマンド (ON で起動要求)
S040C		リザーブ(将来用)
S040D		
S040E		
S040F		
S0410	サブプログラム#1 実行ステータス	サブプログラム#1 実行中のとき ON
S0411	サブプログラム#2 実行ステータス	サブプログラム#2 実行中のとき ON
S0412	サブプログラム#3 実行ステータス	サブプログラム#3 実行中のとき ON
S0413	サブプログラム#4 実行ステータス	サブプログラム#4 実行中のとき ON
S0414		リザーブ(将来用)
S0415	サブプログラム#2 渋滞(警告)	サブプログラム#2 実行渋滞時 ON (サイクリック実行モード)
S0416	サブプログラム#3 渋滞(警告)	サブプログラム#3 実行渋滞時 ON (サイクリック実行モード)
S0417	サブプログラム#4 渋滞(警告)	サブプログラム#4 実行渋滞時 ON (サイクリック実行モード)
S0418		リザーブ(将来用)
S0419		
S041A		
S041B		
S041C		
S041D		
S041E		
S041F		

特殊レジスタ	名 称	機 能
SW042	サブプログラム#2 スキャン回数	サブプログラム#2 サイクリック実行時のスキャン回数設定
SW043	サブプログラム#3 スキャン回数	サブプログラム#3 サイクリック実行時のスキャン回数設定
SW044	サブプログラム#4 スキャン回数	サブプログラム#4 サイクリック実行時のスキャン回数設定
SW045	瞬停検出時間	瞬停時間設定機能における瞬停検出時間を設定

特殊リレー	名 称	機 能
S0460	I/O エラーマップ#0-0	ユニット 0-スロット0の I/O に異常発生時 ON
S0461	I/O エラーマップ#0-1	ユニット 0-スロット1の I/O に異常発生時 ON
S0462	I/O エラーマップ#0-2	ユニット 0-スロット2の I/O に異常発生時 ON
S0463	I/O エラーマップ#0-3	ユニット 0-スロット3の I/O に異常発生時 ON
S0464	I/O エラーマップ#0-4	ユニット 0-スロット4の I/O に異常発生時 ON
S0465	I/O エラーマップ#0-5	ユニット 0-スロット5の I/O に異常発生時 ON
S0466	I/O エラーマップ#0-6	ユニット 0-スロット6の I/O に異常発生時 ON
S0467	I/O エラーマップ#0-7	ユニット 0-スロット7の I/O に異常発生時 ON
S0468	I/O エラーマップ#0-8	ユニット 0-スロット8の I/O に異常発生時 ON
S0469	I/O エラーマップ#0-9	ユニット 0-スロット9の I/O に異常発生時 ON
S046A		リザーブ(将来用)
S046B		
S046C		
S046D		
S046E		
S046F		
S0470	I/O エラーマップ#1-0	ユニット 1-スロット0の I/O に異常発生時 ON
S0471	I/O エラーマップ#1-1	ユニット 1-スロット1の I/O に異常発生時 ON
S0472	I/O エラーマップ#1-2	ユニット 1-スロット2の I/O に異常発生時 ON
S0473	I/O エラーマップ#1-3	ユニット 1-スロット3の I/O に異常発生時 ON
S0474	I/O エラーマップ#1-4	ユニット 1-スロット4の I/O に異常発生時 ON
S0475	I/O エラーマップ#1-5	ユニット 1-スロット5の I/O に異常発生時 ON
S0476	I/O エラーマップ#1-6	ユニット 1-スロット6の I/O に異常発生時 ON
S0477	I/O エラーマップ#1-7	ユニット 1-スロット7の I/O に異常発生時 ON
S0478	I/O エラーマップ#1-8	ユニット 1-スロット8の I/O に異常発生時 ON
S0479	I/O エラーマップ#1-9	ユニット 1-スロット9の I/O に異常発生時 ON
S047A	I/O エラーマップ#1-10	ユニット 1-スロット10 の I/O に異常発生時 ON
S047B		リザーブ(将来用)
S047C		
S047D		
S047E		
S047F		

特殊リレー	名 称	機 能
S0480	I/O エラーマップ#2-0	ユニット 2-スロット0の I/O に異常発生時 ON
S0481	I/O エラーマップ#2-1	ユニット 2-スロット1の I/O に異常発生時 ON
S0482	I/O エラーマップ#2-2	ユニット 2-スロット2の I/O に異常発生時 ON
S0483	I/O エラーマップ#2-3	ユニット 2-スロット3の I/O に異常発生時 ON
S0484	I/O エラーマップ#2-4	ユニット 2-スロット4の I/O に異常発生時 ON
S0485	I/O エラーマップ#2-5	ユニット 2-スロット5の I/O に異常発生時 ON
S0486	I/O エラーマップ#2-6	ユニット 2-スロット6の I/O に異常発生時 ON
S0487	I/O エラーマップ#2-7	ユニット 2-スロット7の I/O に異常発生時 ON
S0488	I/O エラーマップ#2-8	ユニット 2-スロット8の I/O に異常発生時 ON
S0489	I/O エラーマップ#2-9	ユニット 2-スロット9の I/O に異常発生時 ON
S048A	I/O エラーマップ#2-10	ユニット 2-スロット10の I/O に異常発生時 ON
S048B		リザーブ(将来用)
S048C		
S048D		
S048E		
S048F		
S0490	I/O エラーマップ#3-0	ユニット 3-スロット0の I/O に異常発生時 ON
S0491	I/O エラーマップ#3-1	ユニット 3-スロット1の I/O に異常発生時 ON
S0492	I/O エラーマップ#3-2	ユニット 3-スロット2の I/O に異常発生時 ON
S0493	I/O エラーマップ#3-3	ユニット 3-スロット3の I/O に異常発生時 ON
S0494	I/O エラーマップ#3-4	ユニット 3-スロット4の I/O に異常発生時 ON
S0495	I/O エラーマップ#3-5	ユニット 3-スロット5の I/O に異常発生時 ON
S0496	I/O エラーマップ#3-6	ユニット 3-スロット6の I/O に異常発生時 ON
S0497	I/O エラーマップ#3-7	ユニット 3-スロット7の I/O に異常発生時 ON
S0498	I/O エラーマップ#3-8	ユニット 3-スロット8の I/O に異常発生時 ON
S0499	I/O エラーマップ#3-9	ユニット 3-スロット9の I/O に異常発生時 ON
S049A	I/O エラーマップ#3-10	ユニット 3-スロット10の I/O に異常発生時 ON
S049B		リザーブ(将来用)
S049C		
S049D		
S049E		
S049F		

特殊リレー	名 称	機 能
S0500	I/O エラーマップ#4-0	ユニット4-スロット0のI/Oに異常発生時ON
S0501	I/O エラーマップ#4-1	ユニット4-スロット1のI/Oに異常発生時ON
S0502	I/O エラーマップ#4-2	ユニット4-スロット2のI/Oに異常発生時ON
S0503	I/O エラーマップ#4-3	ユニット4-スロット3のI/Oに異常発生時ON
S0504	I/O エラーマップ#4-4	ユニット4-スロット4のI/Oに異常発生時ON
S0505	I/O エラーマップ#4-5	ユニット4-スロット5のI/Oに異常発生時ON
S0506	I/O エラーマップ#4-6	ユニット4-スロット6のI/Oに異常発生時ON
S0507	I/O エラーマップ#4-7	ユニット4-スロット7のI/Oに異常発生時ON
S0508	I/O エラーマップ#4-8	ユニット4-スロット8のI/Oに異常発生時ON
S0509	I/O エラーマップ#4-9	ユニット4-スロット9のI/Oに異常発生時ON
S050A	I/O エラーマップ#4-10	ユニット4-スロット10のI/Oに異常発生時ON
S050B		リザーブ(将来用)
S050C		
S050D		
S050E		
S050F		
S0510	I/O エラーマップ#5-0	ユニット5-スロット0のI/Oに異常発生時ON
S0511	I/O エラーマップ#5-1	ユニット5-スロット1のI/Oに異常発生時ON
S0512	I/O エラーマップ#5-2	ユニット5-スロット2のI/Oに異常発生時ON
S0513	I/O エラーマップ#5-3	ユニット5-スロット3のI/Oに異常発生時ON
S0514	I/O エラーマップ#5-4	ユニット5-スロット4のI/Oに異常発生時ON
S0515	I/O エラーマップ#5-5	ユニット5-スロット5のI/Oに異常発生時ON
S0516	I/O エラーマップ#5-6	ユニット5-スロット6のI/Oに異常発生時ON
S0517	I/O エラーマップ#5-7	ユニット5-スロット7のI/Oに異常発生時ON
S0518	I/O エラーマップ#5-8	ユニット5-スロット8のI/Oに異常発生時ON
S0519	I/O エラーマップ#5-9	ユニット5-スロット9のI/Oに異常発生時ON
S051A	I/O エラーマップ#5-10	ユニット5-スロット10のI/Oに異常発生時ON
S051B		リザーブ(将来用)
S051C		
S051D		
S051E		
S051F		

*) 本ページの機能はT3Hのみ

特殊リレー	名 称	機 能
S0520	I/O エラーマップ#6-0	ユニット 6-スロット0の I/O に異常発生時 ON
S0521	I/O エラーマップ#6-1	ユニット 6-スロット1の I/O に異常発生時 ON
S0522	I/O エラーマップ#6-2	ユニット 6-スロット2の I/O に異常発生時 ON
S0523	I/O エラーマップ#6-3	ユニット 6-スロット3の I/O に異常発生時 ON
S0524	I/O エラーマップ#6-4	ユニット 6-スロット4の I/O に異常発生時 ON
S0525	I/O エラーマップ#6-5	ユニット 6-スロット5の I/O に異常発生時 ON
S0526	I/O エラーマップ#6-6	ユニット 6-スロット6の I/O に異常発生時 ON
S0527	I/O エラーマップ#6-7	ユニット 6-スロット7の I/O に異常発生時 ON
S0528	I/O エラーマップ#6-8	ユニット 6-スロット8の I/O に異常発生時 ON
S0529	I/O エラーマップ#6-9	ユニット 6-スロット9の I/O に異常発生時 ON
S052A	I/O エラーマップ#6-10	ユニット 6-スロット10の I/O に異常発生時 ON
S052B		リザーブ(将来用)
S052C		
S052D		
S052E		
S052F		

特殊 レジスタ	名 称	機 能
SW061	周辺サポート処理応答性の高速化指定	<ul style="list-style-type: none"> 周辺装置(プログラマ、コンピュータリンク、伝送装置)からの応答性を高速化するため、メイン CPU の空き時間を強制的に設定可能。 本体スキャンがフローティング・スキャンモード時にご使用下さい。 設定値は 0~10 まで、1ms 単位に設定可能。 本体の第2スキャン時にこの設定値を内部に取込むため、それ以降、設定値を変更しても無効。

*) 本機能は T3H(ver1.2 以降)でサポート

特殊リレー	名 称	機 能
S0620	サンプリングトレースコピー	サンプリングトレースデータを IC メモリカードにコピー。(ON でコピー)
S0621		
S0622		
S0623		
S0624		
S0625		
S0626		
S0627		
S0628		リザーブ(将来用)
S0629		
S062A		
S062B		
S062C		
S062D		
S062E		
S062F		

*) 本ページの機能は T3H のみ

特殊リレー	名 称	機 能
S0630	テスト機能実行フラグ	テスト機能実行中 ON
S0631		
S0632		リザーブ(将来用)
S0633		
S0634	ソケット I/F 伝送許可フラグ	ソケット I/F 伝送許可状態 ON
S0635	メッセージ伝送許可フラグ	メッセージ伝送許可状態 ON
S0636		
S0637		
S0638		
S0639		リザーブ(将来用)
S063A		
S063B		
S063C	ラン	ラン状態のとき ON
S063D	スタンバイ	スタンバイ状態のとき ON
S063E	イニシャライズ	イニシャライズ状態のとき ON
S063F	ダウソ	ダウソ状態のとき ON

*) 本機能は T3H (ver1.13 以降) のみサポート。詳細はイーサネットモジュール (EN311) 取扱説明書参照

特殊レジスタ	名 称	機 能
SW064	EN311 CH2 ステーションステータス	レジスタ内各ビットの内容は、上記(SW063)と同様
SW065	EN311 CH3 ステーションステータス	
SW066	EN311 CH4 ステーションステータス	

*) 本機能は T3H(ver1.13 以降)のみサポート。詳細はイーサネットモジュール(EN311)取扱説明書参照

特殊リレー	名 称	機 能
S0670	XW/YW レジスタ・プロテクト	XW/YW レジスタをプロテクト時 ON に設定する
S0671	W レジスタ・プロテクト	W レジスタをプロテクト時 ON に設定する
S0672	LW レジスタ・プロテクト	LW レジスタをプロテクト時 ON に設定する
S0673	RW レジスタ・プロテクト	RW レジスタをプロテクト時 ON に設定する
S0674	D レジスタ・プロテクト	D レジスタをプロテクト時 ON に設定する
S0675	F レジスタ・プロテクト	F レジスタをプロテクト時 ON に設定する
S0676		リザーブ(将来用)
S0677	T レジスタ・プロテクト	T レジスタをプロテクト時 ON に設定する
S0678	C レジスタ・プロテクト	C レジスタをプロテクト時 ON に設定する
S0679	SW レジスタ・プロテクト	SW レジスタをプロテクト時 ON に設定する
S067A		
S067B		
S067C		
S067D		
S067E		
S067F		リザーブ(将来用)

*) 本機能は SEND/RECV 命令を用いた PC リンク機能で、相手局からのレジスタ書き込みをプロテクトするもの。

詳細は命令語説明書参照。

*) 本機能は T3H(ver1.13 以降)のみサポート。

特殊リレー	名 称	機 能
S0780	TOSLINE-F10 CH1 コマンド	通信動作状態
S0781		出力禁止動作完了フラグ
S0782		再構成中フラグ
S0783		リザーブ(将来用)
S0784		スキャン伝送エラーフラグ
S0785		リザーブ(将来用)
S0786		リザーブ(将来用)
S0787		通信停止要求
S0788		ON にて通信停止要求
S0789		出力禁止要求
S078A		リザーブ(将来用)
S078B		
S078C		
S078D		
S078E		
S078F		
S0790	TOSLINE-F10 CH1 ステータス	通信動作状態
S0791		スキャン伝送状態
S0792		リザーブ(将来用)
S0793		
S0794		
S0795		MS 動作モード
S0796		OFF:通常モード ON:テストモード
S0797		ステーション種別
S0798		OFF:マスター ON:リモート
S0799		リザーブ(将来用)
S079A		
S079B		
S079C		
S079D		
S079E		
S079F		

*) 詳細はデータ伝送装置 TOSLINE-F10 概説書参照

特殊レジスタ	名 称	機 能
SW080	TOSLINE-F10 CH2 コマンド	
SW081	TOSLINE-F10 CH2 ステータス	
SW082	TOSLINE-F10 CH3 コマンド	
SW083	TOSLINE-F10 CH3 ステータス	
SW084	TOSLINE-F10 CH4 コマンド	
SW085	TOSLINE-F10 CH4 ステータス	
SW086	TOSLINE-F10 CH5 コマンド	
SW087	TOSLINE-F10 CH5 ステータス	
SW088	TOSLINE-F10 CH6 コマンド	
SW089	TOSLINE-F10 CH6 ステータス	
SW090	TOSLINE-F10 CH7 コマンド	
SW091	TOSLINE-F10 CH7 ステータス	
SW092	TOSLINE-F10 CH8 コマンド	
SW093	TOSLINE-F10 CH8 ステータス	

特殊レジスタ	名 称	機 能
SW094	LW000～LW015	
SW095	LW016～LW031	
SW096	LW032～LW047	
SW097	LW048～LW063	
SW098	LW064～LW079	
SW099	LW080～LW095	
SW100	LW096～LW111	
SW101	LW112～LW127	
SW102	LW128～LW143	
SW103	LW144～LW159	
SW104	LW160～LW175	
SW105	LW176～LW191	
SW106	LW192～LW207	
SW107	LW208～LW223	
SW108	LW224～LW239	
SW109	LW240～LW255	

- レジスタ内各ビットの内容は、前ページ(SW078, SW079)と同様

- TOSLINE-F10 のスキャン伝送において、リンクリレーレジスタ(LW)に対応するスキャンデータが正常に更新されていないとき対応するビットが ON となる。

リンクリレーレジスタ(LW)と特殊レジスタのビットとの対応は、リンクリレーレジスタ LWn に対応する特殊レジスタを SWm、対応するビット位置を P とすれば、次式により決まる。

$$\begin{cases} m=94+(n/16 \text{ の商}) \\ P=(n/16 \text{ の余り}) \end{cases}$$

(例)

$$\begin{aligned} & \text{LW100 のとき、} 100/16=6 \text{ 余り } 4 \text{ より} \\ & \left. \begin{aligned} & m=94+6=100 (\text{SW100}) \\ & P=4(\text{ビット } 4) \end{aligned} \right\} \rightarrow S1004 \end{aligned}$$

特殊リレー	名 称	機 能
S1100	テストモード	テストモード実行中 ON
S1101	TOSLINE-S20 CH1 ステーション ステータス	リザーブ(将来用)
S1102		
S1103		
S1104		
S1105	親局/子局	親局のとき ON
S1106	スキャン伝送禁止/許可	スキャン伝送禁止のとき ON
S1107		
S1108		
S1109	リザーブ(将来用)	
S110A		
S110B		
S110C	オンライン	オンライン状態のとき ON
S110D	スタンバイ	スタンバイ状態のとき ON
S110E	オフライン	オフライン状態のとき ON
S110F	ダウン	ダウン状態のとき ON
S1110	テストモード	テストモード実行中 ON
S1111	TOSLINE-S20 CH2 ステーション ステータス	リザーブ(将来用)
S1112		
S1113		
S1114		親局/子局
S1115		親局のとき ON
S1116		スキャン伝送禁止/許可
S1117		スキャン伝送禁止のとき ON
S1118		
S1119		リザーブ(将来用)
S111A		
S111B		
S111C	オンライン	オンライン状態のとき ON
S111D	スタンバイ	スタンバイ状態のとき ON
S111E	オフライン	オフライン状態のとき ON
S111F	ダウン	ダウン状態のとき ON

*) 詳細はデータ伝送装置 TOSLINE-S20 概説書参照

特殊レジスタ	名 称	機 能
SW112	TOSLINE-S20 CH1 オンラインマップ	ステーション No. 1～No. 16
SW113		ステーション No. 17～No. 32
SW114		ステーション No. 33～No. 48
SW115		ステーション No. 49～No. 64
SW116	TOSLINE-S20 CH2 オンラインマップ	ステーション No. 1～No. 16
SW117		ステーション No. 17～No. 32
SW118		ステーション No. 33～No. 48
SW119		ステーション No. 49～No. 64
SW120	TOSLINE-S20 CH1 スタンバイマップ	ステーション No. 1～No. 16
SW121		ステーション No. 17～No. 32
SW122		ステーション No. 33～No. 48
SW123		ステーション No. 49～No. 64
SW124	TOSLINE-S20 CH2 スタンバイマップ	ステーション No. 1～No. 16
SW125		ステーション No. 17～No. 32
SW126		ステーション No. 33～No. 48
SW127		ステーション No. 49～No. 64

特殊レジスタ	名 称	機 能
SW128	TOSLINE-S20 スキャンヘルシーマップ	W0000～W0015
SW129		W0016～W0031
SW130		W0032～W0047
SW131		W0048～W0063
SW132		W0064～W0079
SW133		W0080～W0095
SW134		W0096～W0111
SW135		W0112～W0127
SW136		W0128～W0143
SW137		W0144～W0159
SW138		W0160～W0175
SW139		W0176～W0191
SW140		W0192～W0207
SW141		W0208～W0223
SW142		W0224～W0239
SW143		W0240～W0255

*) TOSLINE-S20 が CH1, CH2 の2台存在しているときには、スキャンヘルシーマップはリンクレジスタ (W) に割り付けられている CH のスキャンデータの状態が反映される(本体スキャンの第2スキャン目から有効)

特殊レジスタ	名 称	機 能
SW144	W0256～W0271	
SW145	W0272～W0287	
SW146	W0288～W0303	
SW147	W0304～W0319	
SW148	W0320～W0335	
SW149	W0336～W0351	
SW150	W0352～W0367	
SW151	W0368～W0383	
SW152	W0384～W0399	
SW153	W0400～W0415	
SW154	W0416～W0431	
SW155	W0432～W0447	
SW156	W0448～W0463	
SW157	W0464～W0479	
SW158	W0480～W0495	
SW159	W0496～W0511	
SW160	W0512～W0527	
SW161	W0528～W0543	
SW162	W0544～W0559	
SW163	W0560～W0575	
SW164	W0576～W0591	
SW165	W0592～W0607	
SW166	W0608～W0623	
SW167	W0624～W0639	
SW168	W0640～W0655	
SW169	W0656～W0671	
SW170	W0672～W0687	
SW171	W0688～W0703	
SW172	W0704～W0719	
SW173	W0720～W0735	
SW174	W0736～W0751	
SW175	W0752～W0767	

TOSLINE-S20
スキャンヘルシー
マップ

- TOSLINE-S20 のスキャン伝送において、リンクレジスタ (W) に対応するスキャンデータが正常に更新されているとき、対応するビットが ON となる。

- リンクレジスタ (W) と特殊レジスタのビットとの対応は、リンクレジスタ Wn に対応する特殊レジスタを SWm、対応するビット位置を P とすれば、次式により決まる。

$$\begin{cases} m=128+(n/16 \text{ の商}) \\ P=(n/16 \text{ の余り}) \end{cases}$$

(例)

$$\begin{aligned} & W0500 \text{ のとき、 } 500/16=31 \text{ 余り } 4 \text{ より、} \\ & m=128+31=159 (\text{SW159}) \\ & P=4(\text{ビット } 4) \end{aligned} \quad \rightarrow S1594$$

*) TOSLINE-S20 が CH1, CH2 の2台存在しているときには、リンクレジスタ (W) に割り付けられている CH のスキャンデータの状態が反映される(本体スキャンの第2スキャン目から有効)

特殊レジスタ	名 称	機 能
SW176	TOSLINE-S20 スキャンヘルシー マップ	W0768～W0783
SW177		W0784～W0799
SW178		W0800～W0815
SW179		W0816～W0831
SW180		W0832～W0847
SW181		W0848～W0863
SW182		W0864～W0879
SW183		W0880～W0895
SW184		W0896～W0911
SW185		W0912～W0927
SW186		W0928～W0943
SW187		W0944～W0959
SW188		W0960～W0975
SW189		W0976～W0991
SW190		W0992～W1007
SW191		W1008～W1023

- TOSLINE-S20 のスキャン伝送において、リンクレジスタ (W) に対応するスキャンデータが正常に更新されているとき、対応するビットが ON となる。

- リンクレジスタ (W) と特殊レジスタのビットとの対応は、リンクレジスタ W_n に対応する特殊レジスタを SW_m 、対応するビット位置を P とすれば、次式により決まる。

$$\begin{cases} m=128+(n/16 \text{ の商}) \\ P=(n/16 \text{ の余り}) \end{cases}$$

(例)

$$\begin{aligned} & W0990 \text{ のとき}, 990/16=61 \text{ 余り } 14 \text{ より,} \\ & m=128+61=189 (SW189) \\ & P=14 (\text{ビット } E) \end{aligned} \quad \} \rightarrow S189E$$

*) TOSLINE-S20 が CH1, CH2 の2台存在しているときには、リンクレジスタ (W) に割り付けられている CH のスキャンデータの状態が反映される(本体スキャンの第2スキャン目から有効)

特殊レジスタ	名 称	機 能
SW192	W01024～W1039	<ul style="list-style-type: none"> TOSLINE-S20 のスキャン伝送において、リンクレジスタ (W) に対応するスキャンデータが正常に更新されているとき、対応するビットが ON となる。
SW193	W1040～W1055	
SW194	W1056～W1071	
SW195	W1072～W1087	
SW196	W1088～W1103	
SW197	W1104～W1119	
SW198	W1120～W1135	
SW199	W1136～W1151	
SW200	W1152～W1167	
SW201	W1168～W1183	
SW202	W1184～W1199	
SW203	W1200～W1215	
SW204	W1216～W1231	
SW205	W1232～W1247	
SW206	W1248～W1263	
SW207	W1264～W1279	
SW208	W1280～W1295	
SW209	W1296～W1311	
SW210	W1312～W1327	
SW211	W1328～W1343	
SW212	W1344～W1359	
SW213	W1360～W1375	
SW214	W1376～W1391	
SW215	W1392～W1407	
SW216	W1408～W1423	
SW217	W1424～W1439	
SW218	W1440～W1455	
SW219	W1456～W1471	
SW220	W1472～W1487	
SW221	W1488～W1503	
SW222	W1504～W1519	
SW223	W1520～W1535	

- リンクレジスタ (W) に対応するスキャンデータが正常に更新されているとき、対応するビットが ON となる。
- リンクレジスタ (W) と特殊レジスタのビットとの対応は、リンクレジスタ W_n に対応する特殊レジスタを SW_m 、対応するビット位置を P とすれば、次式により決まる。

$$\begin{cases} m=128+(n/16 \text{ の商}) \\ P=(n/16 \text{ の余り}) \end{cases}$$

(例)

$$\begin{aligned} & W1300 \text{ のとき}, 1300/16=81 \text{ 余り } 4 \text{ より}, \\ & \left. \begin{aligned} m &= 128+81=209 (\text{SW209}) \\ P &= (n/16 \text{ の余り}) \end{aligned} \right\} \rightarrow S2094 \\ & P=4(\text{ビット } 4) \end{aligned}$$

*) TOSLINE-S20 が CH1, CH2 の2台存在しているときには、リンクレジスタ (W) に割り付けられている CH のスキャンデータの状態が反映される(本体スキャンの第2スキャン目から有効)

**) 本ページの機能は T3H のみ

特殊レジスタ	名 称	機 能
SW224	W1536～W1551	
SW225	W1552～W1567	
SW226	W1568～W1583	
SW227	W1584～W1599	
SW228	W1600～W1615	
SW229	W1616～W1631	
SW230	W1632～W1647	
SW231	W1648～W1663	
SW232	W1664～W1679	
SW233	W1680～W1695	
SW234	W1696～W1711	
SW235	W1712～W1727	
SW236	W1728～W1743	
SW237	W1744～W1759	
SW238	W1760～W1775	
SW239	TOSLINE-S20 スキャンヘルシー マップ	
SW240	W1776～W1791	
SW241	W1792～W1807	
SW242	W1808～W1823	
SW243	W1824～W1839	
SW244	W1840～W1855	
SW245	W1856～W1871	
SW246	W1872～W1887	
SW447	W1888～W1903	
SW248	W1904～W1919	
SW249	W1920～W1935	
SW250	W1936～W1951	
SW251	W1952～W1967	
SW252	W1968～W1983	
SW253	W1984～W1999	
SW254	W2000～W2015	
SW255	W2016～W2031	
	W2032～W2047	

- TOSLINE-S20 のスキャン伝送において、リンクレジスタ (W) に対応するスキャンデータが正常に更新されているとき、対応するビットが ON となる。

- リンクレジスタ (W) と特殊レジスタのビットとの対応は、リンクレジスタ W_n に対応する特殊レジスタを SW_m 、対応するビット位置を P とすれば、次式により決まる。

$$\begin{cases} m=128+(n/16 \text{ の商}) \\ P=(n/16 \text{ の余り}) \end{cases}$$

(例)

$$\begin{cases} W1900 \text{ のとき、 } 1900/16=118 \text{ 余り } 2 \text{ より、} \\ m=128+118=246 \text{ (SW246)} \\ P=2 \text{ (ビット } 2) \end{cases} \rightarrow S2462$$

*) TOSLINE-S20 が CH1, CH2 の2台存在しているときには、リンクレジスタ (W) に割り付けられている CH のスキャンデータの状態が反映される(本体スキャンの第2スキャン目から有効)

**) 本ページの機能は T3H のみ

3.3

レジスタデータの
扱い

レジスタとは「16 ビットのデータを格納する場所」ということは前にも説明しましたが、PC の命令語では、単独のレジスタまたは連続する複数のレジスタを用いて次の種類のデータを扱うことができます。

- 正整数(0~65535 の範囲の整数)
- 整数(-32768~32767 の範囲の整数)
- BCD(BCD コードで表現された 0~9999 の範囲の整数)
- 正の倍長整数(0~4294967295 の範囲の整数)
- 倍長整数(-2147483648~2147483647 の範囲の整数)
- 倍長 BCD(BCD コードで表現された 0~99999999 の範囲の整数)
- 実数(-3.40282×10³⁸~3.40282×10³⁸ の範囲の実数)

ただし、これらの種類のデータを扱うために、種類に応じた専用のレジスタがある訳ではなく、どの命令語を使用するかによってレジスタデータの扱いが変わってきます。

つまり、下記の例のように、同じレジスタを用いた場合でも、命令語のデータタイプが異なれば、レジスタデータの扱いも変わります。

例)

D0005 の値が HFFF(16 進数の FFFF) のとき、

- ① 符号なし比較命令(より大きい)では、

—[D0005 U > 100]— 判定出力(真のとき ON)

D0005 の値は 65535(正整数)と見なされ、従って比較値(100)よりも大きいと判断されて、命令の出力は ON となる。

- ② 符号付き比較命令(より大きい)では、

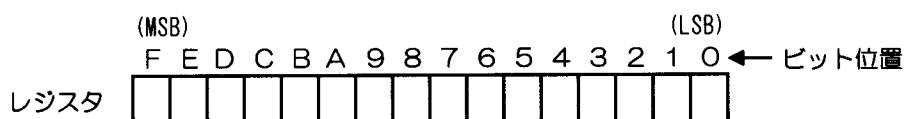
—[D0005 > 100]— 判定出力(真のとき ON)

D0005 の値は -1(整数)と見なされ、従って比較値(100)よりも大きくないと判断されて、命令の出力は OFF となる。

このように、レジスタにはデータタイプによる区別はないため、よく理解して使用すれば複雑なデータ操作を行うことも可能です。ただし、プログラムを見やすくするために、できるだけデータタイプ別にレジスタを割り当てて(1つのレジスタは1つのデータタイプで扱う)使用することをお勧めします。

(1) 正整数

レジスタ1個で表現される16ビットの正の整数です。レジスタ内部のビット構成は下図のようになります。



ビット0が最下位ビット、ビットFが最上位ビットとなります。扱える数値範囲は下表の通りです。

数値(10進)	2進数表現	16進数表現
65535	1111 1111 1111 1111	FFFF
65534	1111 1111 1111 1110	FFFE
5	5	5
1	0000 0000 0000 0001	0001
0	0000 0000 0000 0000	0000

3

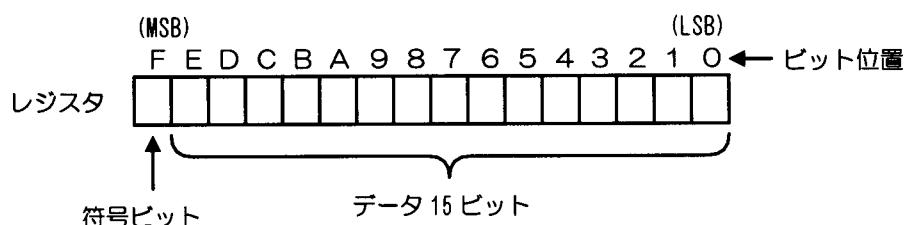
補足

プログラミング時及びプログラムモニタ時、レジスタデータの表示/設定は、10進数と16進数の切換えが可能です。16進数表示のときは、数値の前に“H”が付きます。

例) H89AB(16進数の89AB)

(2) 整数

レジスタ1個で表現される16ビットの整数です。負の数は2の補数表現となります。



ビット0からビットEまでの15ビットで数値を表現し、ビットFで符号(正のとき0, 負のとき1)を表わします。

扱える数値範囲及び表現形式は下表の通りです。

数値(10進)	2進数表現				16進数表現
32767	0111	1111	1111	1111	7FFF
32766	0111	1111	1111	1110	7FFE
5					5
1	0000	0000	0000	0001	0001
0	0000	0000	0000	0000	0000
-1	1111	1111	1111	1111	FFFF
5					5
-32767	1000	0000	0000	0001	8001
-32768	1000	0000	0000	0000	8000

2の補数表現とは、互いに足し合わせると下位16ビットが全て0となる数値のことです。

例)

$$\begin{array}{r}
 & 0111 & 1111 & 1111 & 1111 & \text{(2進)} = 32767 \\
 + \} & 1000 & 0000 & 0000 & 0001 & \text{(2進)} = -32767 \\
 \hline
 & 1\ 0000 & 0000 & 0000 & 0000 &
 \end{array}$$

計算上は、数値の各ビットを反転し、1を加えるという操作によって、その数値の2の補数を求めることができます。

例)

$$0111\ 1111\ 1111\ 1111 \text{ (2進)} = 32767$$

(ビット反転)

$$1000\ 0000\ 0000\ 0000 \text{ (2進)}$$

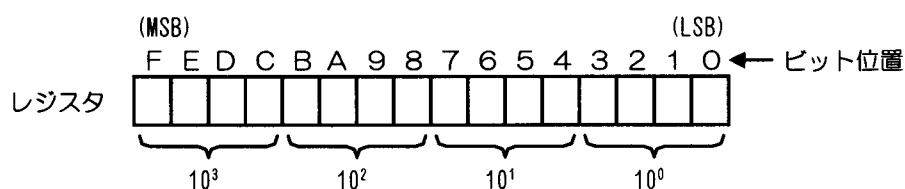
(1を加える)

$$1000\ 0000\ 0000\ 0001 \text{ (2進)} = -32767$$

(3) BCD

BCDとは、Binary Coded Decimalの略で2進化10進数とも呼びます。

2進数の4ビットで10進数の1桁(0~9)を表現します。従って1個のレジスタで4桁の10進数の数値を表すことができます。



扱える数値範囲及び表現形式は下表の通りです。

数 値	2進数表現	16進数表現
9999	1001 1001 1001 1001	9999
9998	1001 1001 1001 1000	9998
‐	‐	‐
10	0000 0000 0001 0000	0010
9	0000 0000 0000 1001	0009
‐	‐	‐
1	0000 0000 0000 0001	0001
0	0000 0000 0000 0000	0000

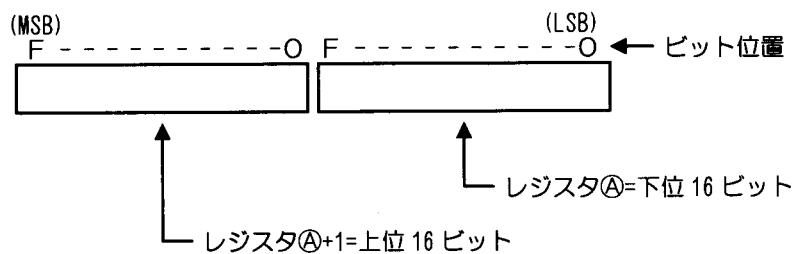
一 補足

BCD データは基本的には、BCD 出力形の数値設定器等からのデータ入力、及び BCD 入力形の数値表示器等へのデータ出力などに使用するデータ形式ですが、T3/T3H では、BCD データのまま四則演算を行う専用命令も備えています。

3

(4) 正の倍長整数

連続する2つのレジスタを用いて表現される 32 ビットの正の整数です。倍長データの場合、レジスタは $\textcircled{A}+1 \cdot \textcircled{A}$ のように指定し、 \textcircled{A} が下位 16 ビットを、 $\textcircled{A}+1$ が上位 16 ビットを示します。 $(\textcircled{A}+1$ はレジスタ \textcircled{A} の次のレジスタ)



例) レジスタ D0201・D0200 で正の倍長整数を扱うとすれば、D0200 が \textcircled{A} に、D0201 が $\textcircled{A}+1$ になり、D0200 側が下位、D0201 側が上位となります。

プログラミング上は、命令語の倍長オペランドを指定する位置で、D0200 と入力すると自動的に D0201・D0200 と表示されます。

正の倍長整数が扱える数値範囲を次ページの表に示します。

数 値	16進数表現	
	レジスタⒶ+1	レジスタⒶ
4294967295	FFFF	FFFF
\$	\$	\$
65536	0001	0000
65535	0000	FFFF
\$	\$	\$
0	0000	0000

補足

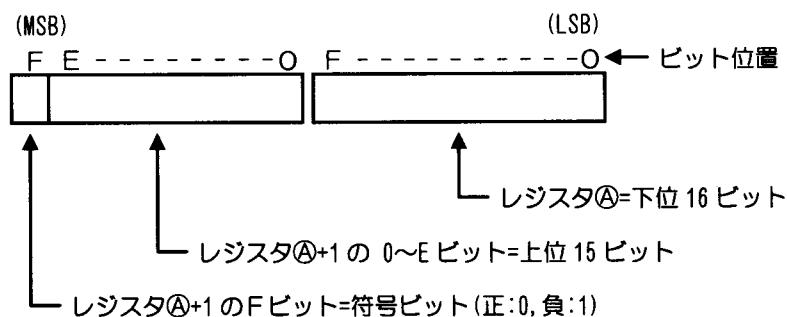
レジスタⒶとしては奇数アドレスでも偶数アドレスでもかまいません。

(5) 倍長整数

連続する2つのレジスタを用いて表現される32ビットの整数です。

負の数は2の補数表現となります。((2)整数の項参照)

レジスタはⒶ+1・Ⓐのように指定し、Ⓐが下位、Ⓐ+1が上位となります。



レジスタⒶのOビットからレジスタⒶ+1のEビットまでの31ビットで数値を表わし、レジスタⒶ+1のFビットで符号(正のとき0、負のとき1)を表わします。

例) レジスタ D1002・D1001 で倍長整数を扱うとすれば、D1001 がⒶに、D1002 がⒶ+1 になり、D1001 が下位、D1002 が上位となります。また符号は D1002 のFビットで表わされます。

プログラミング上は、命令語の倍長オペランドを指定する位置で、D1001 と入力すると自動的に D1002・D1001 と表示されます。

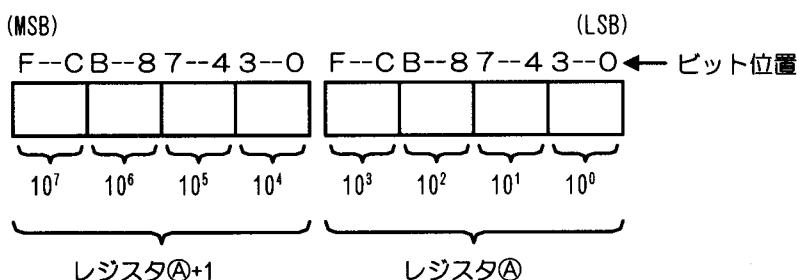
倍長整数の扱える数値範囲を次ページの表に示します。

数 値	16進数表現	
	レジスタⒶ+1	レジスタⒶ
2147483647	7FFF	FFFF
\$	\$	\$
65536	0001	0000
65535	0000	FFFF
\$	\$	\$
0	0000	0000
-1	FFFF	FFFF
\$	\$	\$
-65536	FFFF	0000
-65537	FFFE	FFFF
\$	\$	\$
-2147483648	8000	0000

3

(6) 倍長 BCD

連続する2つのレジスタを用いて表現される8桁のBCDデータです。



レジスタは④+1・④のように指定し、④が下位4桁、④+1 が上位4桁となります。

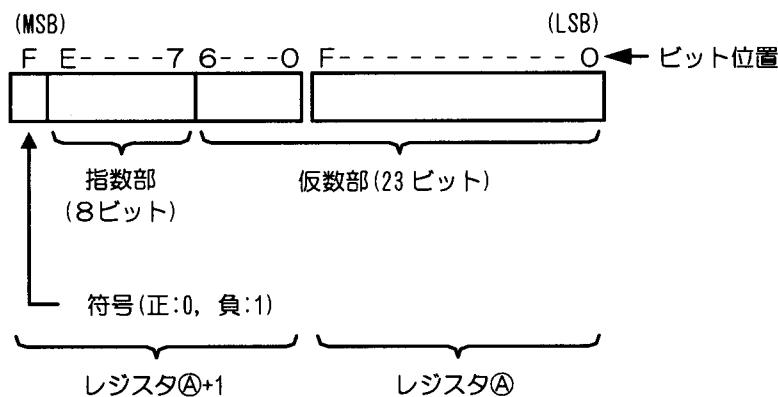
例) レジスタ XW001・XW000 で倍長 BCD を扱うとすれば、XW000 がⒶ、
XW001 がⒶ+1 となり、XW000 が下位4桁、XW001 が上位4桁となります。

倍長 BCD データが扱える数値範囲及び表現形式を下表に示します。

数 値	16進数表現	
	レジスタⒶ+1	レジスタⒶ
99999999	9999	9999
\$	\$	\$
1	0000	0001
0	0000	0000

(7) 実数

連続する2つのレジスタを用いて表現される実数（浮動小数点データ）です。レジスタは $\textcircled{A}+1 \cdot \textcircled{A}$ のように指定し、内部的には次のようなフォーマットでデータが格納されます。（IEEE754に準拠）



$$\text{値} = (\text{符号}) 1.(\text{仮数部}) \times 2^{(\text{指数部}-127)}$$

実数（浮動小数点データ）は、下記の浮動小数点命令においてのみ使用されます。従って、上記の浮動小数点データ格納フォーマットを意識する必要はありません。

- 変換命令（浮動小数点 \leftrightarrow 倍長整数）
- 浮動小数点四則演算命令
- 浮動小数点比較命令
- 浮動小数点関数命令（三角関数、指数関数、対数関数、平方根など）
- 浮動小数点プロセス演算（リミット、デッドバンド、PIDなど）

実数（浮動小数点データ）で扱える数値範囲は以下の通りです。

数 値	表 現	備 考
3.40282×10^{38}	$3.40282E38$	最大値
∞	∞	
1.17549×10^{-38}	$1.17549E-38$	0に近い数
0	0	
-1.17549×10^{-38}	$-1.17549E-38$	0に近い数
$-\infty$	$-\infty$	
-3.40282×10^{38}	$-3.40282E38$	最小値

3.4

インデックス修飾

命令語にてレジスタを使用する場合、下記例 1) のように直接レジスタを指定する方法を直接アドレス指定といいます。

これに対し、下記例 2) のようにインデックスレジスタ (I, J, K) の内容との組み合わせで、レジスタを間接的に指定する方法を間接アドレス指定といいます。特にこの場合は、インデックスレジスタを用いてアドレスの修飾を行いますので、インデックス修飾と呼びます。

また、T3H ではプログラム中で直接アドレス指定が行えるファイルレジスタは F0000～F9999 です。

このため、F10000～F32767 を指定する場合にはインデックス修飾によってアドレスを指定します。

例 1)

[RW100 MOV D3500]

データ転送命令

RW100 の内容を D3500 に転送する

3

例 2)

[RW100 MOV D3500]^I^J

データ転送命令(インデックス修飾付き)

RW(100+I) の内容を D(3500+J) に転送する

(I=3, J=200 なら、RW103 の内容を D3700 に転送することになる)

インデックスレジスタとしては、I, J, K の3種類があり、各々16 ビットの整数(-32768～32767)を扱います。これら3種類のインデックスレジスタには特に機能上の違いはありません。

これらのインデックス命令に値を代入するための特別な命令はありません。通常の転送命令や演算命令の転送先として指定します。

例 1) インデックスレジスタに定数を代入する

[64 MOV I] (インデックスレジスタ I に 64 を代入)

[-2 MOV J] (インデックスレジスタ J に -2 を代入)

例 2) インデックスレジスタにレジスタデータを代入する

[D0035 MOV K] (インデックスレジスタ K に D0035 の値を代入)

[RW078 MOV I] (インデックスレジスタ I に RW078 の値を代入)

例 3) インデックスレジスタに演算の結果を代入する

$\boxed{RW200 - 30 \rightarrow I}$

(RW200 から 30 を引いた結果を I に代入)

$\boxed{XW004 ENC (4) J}$

(XW004 の最上位 ON ビットの位置を J に代入 (エンコード))

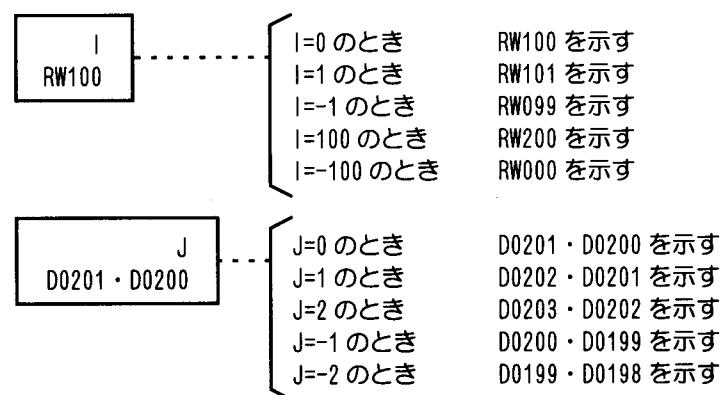
補足

インデックスレジスタは単長 (16 ビット) として扱うことが基本ですが、例えば乗算命令などのように結果が倍長となる命令の結果格納先としてインデックスレジスタを使用する場合には、J・I または K・J という組み合わせのみ有効です。この場合、倍長オペランドの位置で I と指定することにより、J・I となり、J が上位 I が下位となります。同様に J と指定することにより K・J となり、K が上位 J が下位となります。

例)

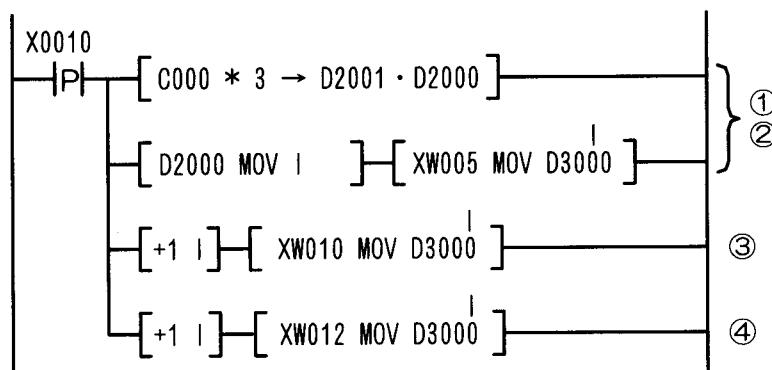
$\boxed{D1357 * 10 \rightarrow J \cdot I}$

レジスタにインデックス修飾を行った場合の例を下に示します。



インデックス修飾を命令語に適用した場合の動作例を以下に示します。

例)



X0010 が OFF から ON に変化したときに以下の処理を行う

- ① C000 の内容を 3 倍した値をインデックスレジスタ I に代入する
- ② XW005 の内容を D(3000+I) に格納する
- ③ I の内容に 1 を加え、XW010 の内容を D(3000+I) に格納する
- ④ さらに I の内容に 1 を加え、XW012 の内容を D(3000+I) に格納する

3

なお、

は、デバイス④の OFF→ON の立ち上がりに 1 回(次スキャンのこの命令実行までの間)だけ ON となるポジティブパルス接点

[④ * ④ → ④+1 · ④] は、④に④を掛けて④+1 · ④の倍長レジスタに格納する乗算命令

[+1 ④] は、④の内容に 1 を加え④に格納するインクリメント命令

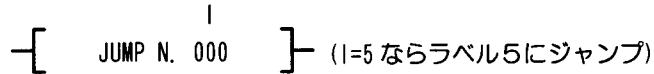
[④ MOV ④] は、④の内容を④に代入するデータ転送命令

補足

- (1) インデックスレジスタへの値の代入及びインデックス修飾は、プログラム中何回行ってもかまいません。従って通常は、インデックス修飾を行う直前にインデックスレジスタへの値の代入を行った方がプログラムが見やすくなります。
- (2) インデックス修飾によってレジスタがアドレス範囲を超えないように注意して下さい。インデックス修飾の結果がアドレス範囲を超えてしまう場合には、その命令は不実行となり、バウンダリエラー発生を示す特殊リレー(S0051 と S0064)が ON となります。

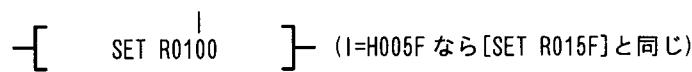
前述のように、レジスタに対してインデックス修飾を行い、間接アドレス指定として使用するのが通常のインデックス修飾の使用方法ですが、特殊な例として下記の使い方も可能です。

- サブルーチンコール命令(CALL)及び条件ジャンプ命令(JUMP)の飛び先指定にインデックス修飾を行う。


[JUMP N. 000] (I=5ならラベル5にジャンプ)

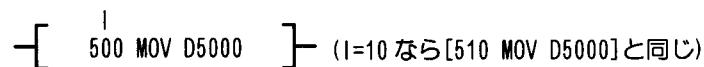
なお、インデックス修飾後の飛び先が存在しない場合には、特殊リレーS0051とS0060が、飛び先が設定範囲を超える場合には、S0051とS0065がONとなり、いずれの場合も命令不実行となります。

- セット命令(SET),リセット命令(RST)等、デバイスオペランドを持つファンクション命令については、デバイスのインデックス修飾が可能です。


[SET R0100] (I=H005Fなら[SET R015F]と同じ)

基本命令のデバイスに対するインデックス修飾はできません。

- 定数オペランドを持つファンクション命令については、定数のインデックス修飾が可能です。(定数に対するオフセットとなります。)


[500 MOV D5000] (I=10なら[510 MOV D5000]と同じ)

補足

- (1) 基本命令(タイマ命令、カウンタ命令含め)については、インデックス修飾はできません。
- (2) 個々の命令の各オペランドについて、インデックス修飾が可能かどうかについては、別冊の命令語説明書をご覧下さい。

3.5

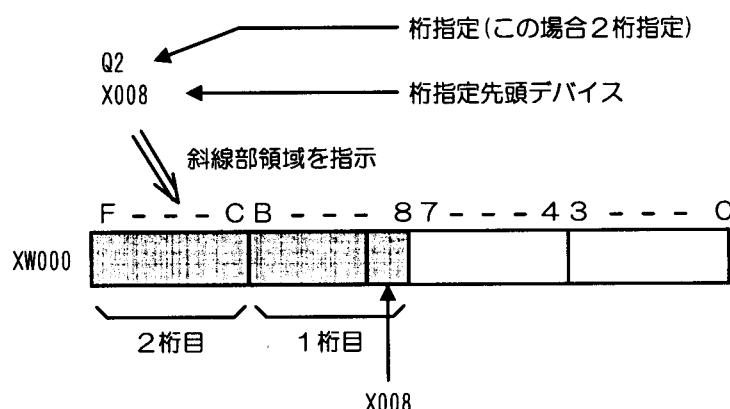
桁指定

レジスタデータの特殊な指定方法として桁指定という方法があります。

桁指定とは、16進数の1桁(4ビット)をデータの単位として扱い、指定したデバイス(ビット位置)から何桁分をデータ操作の対象とするか、という指定の方法となります。

具体的には、下の例の場合は、X0008 から 2 桁分(つまり XW000 の上位 8 ビット)がデータ操作の対象となります。

例)



3

桁指定には Q0, Q1, …, Q8 の 9 種類があり、各々以下の意味となります。

Q0 … 指定デバイス 1 ビットをデータ操作の対象とする

Q1 … 指定デバイスを先頭とする 1 桁(4 ビット)をデータ操作の対象とする

Q2 … 指定デバイスを先頭とする 2 桁(8 ビット)をデータ操作の対象とする

Q3 … 指定デバイスを先頭とする 3 桁(12 ビット)をデータ操作の対象とする

Q4 … 指定デバイスを先頭とする 4 桁(16 ビット)をデータ操作の対象とする

Q5 … 指定デバイスを先頭とする 5 桁(20 ビット)をデータ操作の対象とする

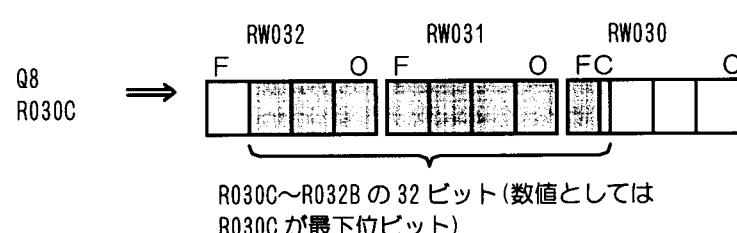
Q6 … 指定デバイスを先頭とする 6 桁(24 ビット)をデータ操作の対象とする

Q7 … 指定デバイスを先頭とする 7 桁(28 ビット)をデータ操作の対象とする

Q8 … 指定デバイスを先頭とする 8 桁(32 ビット)をデータ操作の対象とする

なお、桁指定で指定した領域が複数のレジスタに渡る場合には、下の例のように、アドレスの小さい方から大きい方に渡って領域が指定されます。

例)



以下に、参照オペランド(そのデータを使用して命令を実行するためのレジスタ)として桁指定を行う場合、及び転送先オペランド(命令実行の結果を格納するレジスタ)として桁指定を行う場合の各々について、桁指定の働きを説明します。

なお、1つの命令で参照オペランドと転送先オペランドの両方に桁指定をかけることも可能です。

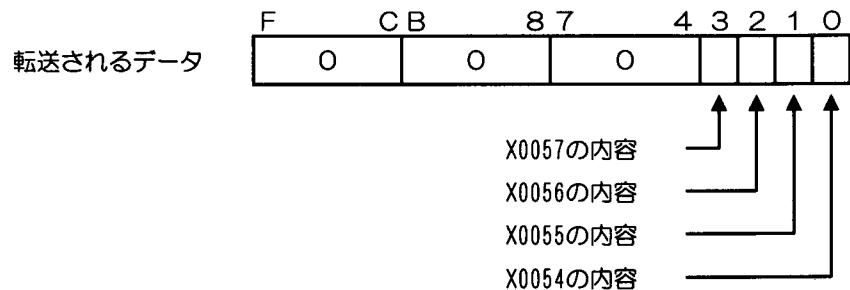
(1) 参照オペランドに対する桁指定

単長(16ビット)オペランドに対しては、Q0～Q4の指定が有効で(Q5～Q8は指定不可)、データが足りないビット分(2桁を指定した場合の上位8ビット分など)は0として、16ビット分のデータが参照されます。

例 1)

$\left[\begin{array}{c} Q1 \\ X0054 \text{ MOV D1000} \end{array} \right]$ (データ転送命令)

X0054からの1桁(4ビット)分のデータを下位4ビットの内容とし、上位12ビットは0であるデータ(下記)をD1000に転送する。

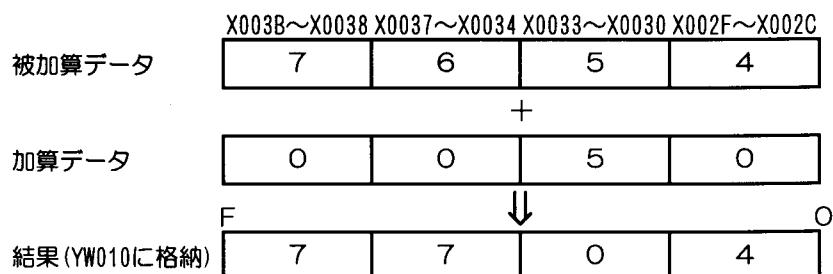


例 2)

$\left[\begin{array}{c} Q4 \\ X002C \text{ B+ H0050 } \rightarrow YW010 \end{array} \right]$ (BCD加算命令)

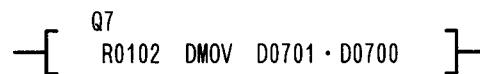
X002Cから始まる4桁(16ビット)分のBCDデータにBCDの50を加え、結果をYW010にBCDコードで格納する。

(下記は、XW003=H8765, XW002=H4321の場合)

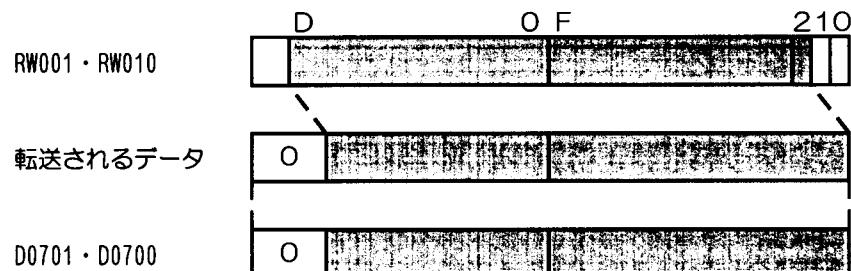


倍長(32ビット)オペランドに対しては、Q0～Q8の指定が全て有効で、データが足りないビット分は0として、32ビット分のデータが参照されます。

例)

 (倍長データ転送)

R0102から始まる7桁(28ビット)分のデータを下位28ビットの内容とし、上位4ビットは0である32ビットデータをD0701・D0700の倍長レジスタに転送する。



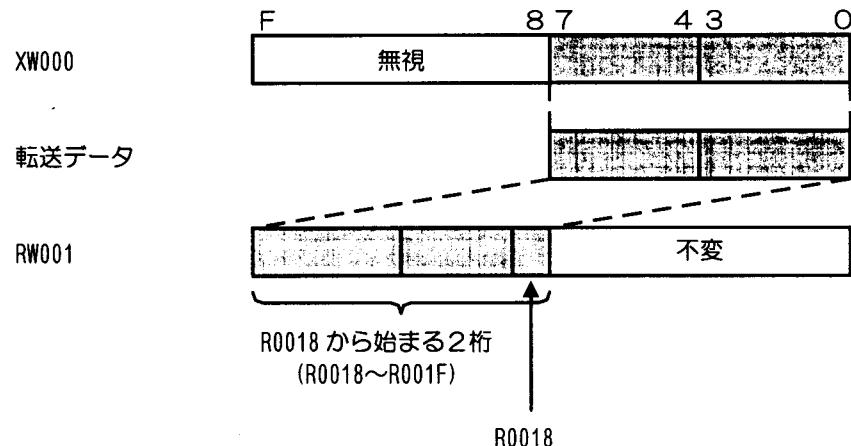
(2) 転送先オペランドに対する桁指定

単長(16ビット)オペランドに対しては、Q0～Q4の指定が有効で(Q5～Q8は指定不可)、指定された桁数分に命令実行の結果が格納されます。指定桁数以上のデータ(16ビットデータを2桁の転送先に転送する場合の上位8ビットデータなど)は無視されます。

例 1)

 (データ転送命令)

XW000の下位2桁(8ビット)のデータをR0018から始まる2桁(8ビット)に転送する。

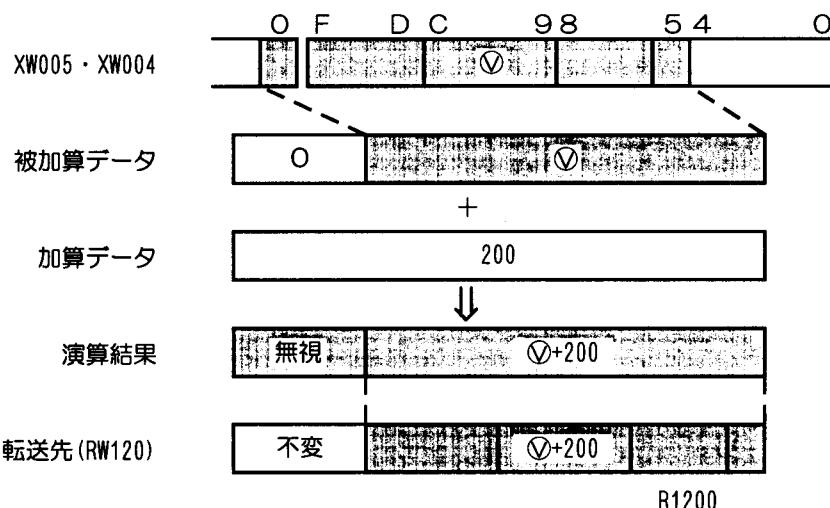


例 2)

$\left[\begin{array}{c} Q3 \\ X0045 + 200 \rightarrow R1200 \end{array} \right] \quad \text{(加算命令)}$

X0045 から始まる 3 衔(12 ビット)のデータに 200 を加え、R1200 から始まる 3 衔(12 ビット)に格納する。

(参照オペランドと転送先オペランドの両方に桁指定を行った例)



例えば、XW005=H0077=0000 0000 0111 0111(2進)

XW004=H182A=0001 1000 0010 1010(2進)

のとき、被加算データは、

0000 1000 1100 0001(2進)=H08C1=2241(10進)

これに 200 を加えると、

0000 1001 1000 1001(2進)=H0989=2441(10進)

従って、RW120 のビット 0 (R1200) からビット Bまでの3桁(12 ビット)に

1001 1000 1001(2進)=H989=2441(10進)

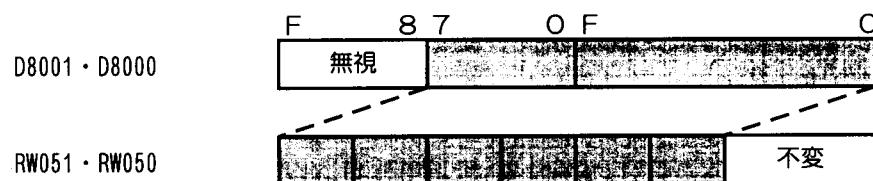
というデータが格納される。

倍長(32 ビット)オペランドに対しては、Q0～Q8 の指定が全て有効で、指定された桁数分の領域に命令実行の結果が格納されます。指定桁数以上のデータ(倍長データを 7 衔の転送先に転送する場合の上位 4 ビットデータなど)は無視されます。

例)

$\left[\begin{array}{c} Q6 \\ D8001 \cdot D8000 \quad DMOV \quad R0508 \end{array} \right] \quad \text{(倍長データ転送命令)}$

D8001・D8000 の倍長データの下位6桁(24ビット)を R0508 を先頭とする6桁(24ビット)の領域に転送する。



補足

- (1) 桁指定の結果がレジスタのアドレス範囲を超えないように注意して下さい。なお、アドレス範囲を超えて桁指定がかった場合には、超えた部分は無視されます。
- (2) 桁指定とインデックス修飾を組み合わせて使用することも可能です。
例)

Q1 I	<u>I=H001C であれば</u>	Q1	と同じ意味
R0000		R001C	

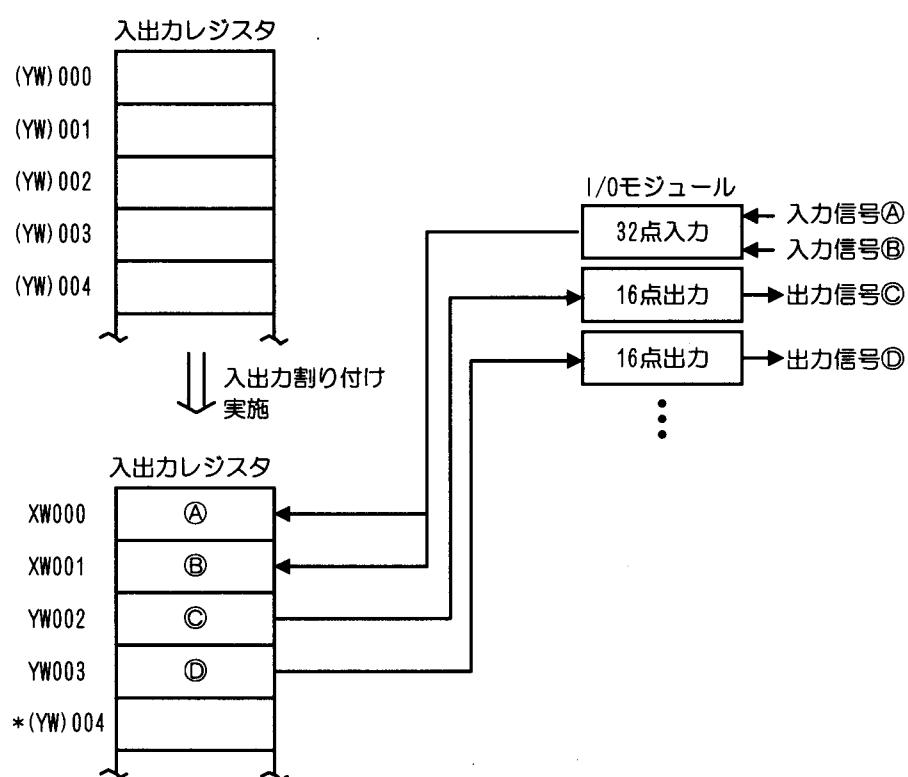
4.1 入出力割り付けの 概要

PC の入力モジュールに入力される外部入力信号の状態は、スキャン制御実行時に、入力レジスタ/デバイス(XW/X または IW/I)を介して読み込まれます。一方、ユーザプログラム実行において決定された出力データは、出力レジスタ/デバイス(YW/Y または OW/O)を介して出力モジュールに出力され、このデータに基づいて出力モジュールから外部負荷に出力されます。

入出力割り付けとは、入力レジスタ/デバイスと入力モジュールとの対応付け、及び出力レジスタ/デバイスと出力モジュールとの対応付けを行うことです。言い換えると、レジスタ/デバイスという論理装置に、I/O モジュールという物理装置を割り当てると言えます。

入力レジスタ/デバイスと出力レジスタ/デバイスは、各々独立したメモリ領域を使用する訳ではなく、入出力レジスタ/デバイスと言うべき一連のメモリ領域を使用します(レジスタアドレス範囲:T3=000 から 255 までの 256 ワード/T3H=000 から 511 までの 512 ワード)。

入出力割り付けを行うことによって、入力モジュールに割り当てられたアドレスは入力レジスタ/デバイスとして、出力モジュールに割り当てられたアドレスは出力レジスタ/デバイスとして、機能種別の決定が行われることになります。



*) I/O モジュールに割り付けられていないアドレスは、内部的には出力(YW)となります。

4.2

入出力割り付けの方法

入出力割り付けを行うということは、システム情報内に入出力割り付け情報を登録すること言い換えることができます。PC の CPU は RUN 起動時に、この入出力割り付け情報に基づいて、I/O モジュールが正常に実装されているかどうかをチェックします。また、同時に、この入出力割り付け情報を基に入出力レジスタ (XW/YW) と I/O モジュールとの対応を決定します。一方プログラマは PC との交信確立時に、この入出力割り付け情報を読み出し、入出力レジスタのアドレス毎に入力 (XW) か出力 (YW) かの区別を認識します。

システム情報内に入出力割り付け情報を登録するために 2 つの方法があります。自動割り付けと個別割り付けの 2 つです。

なお、入出力割り付け情報の登録は、PC が HALT モードでかつメモリプロテクト状態でないとき (モードスイッチ P-RUN 以外) のみ可能です。

自動割り付け

入出力割り付け情報の作成/登録を PC 本体に行わせる方法です。プログラマの一般 I/O 割り付け情報画面にて、自動割り付けコマンドを選択し実行することによって行われます。

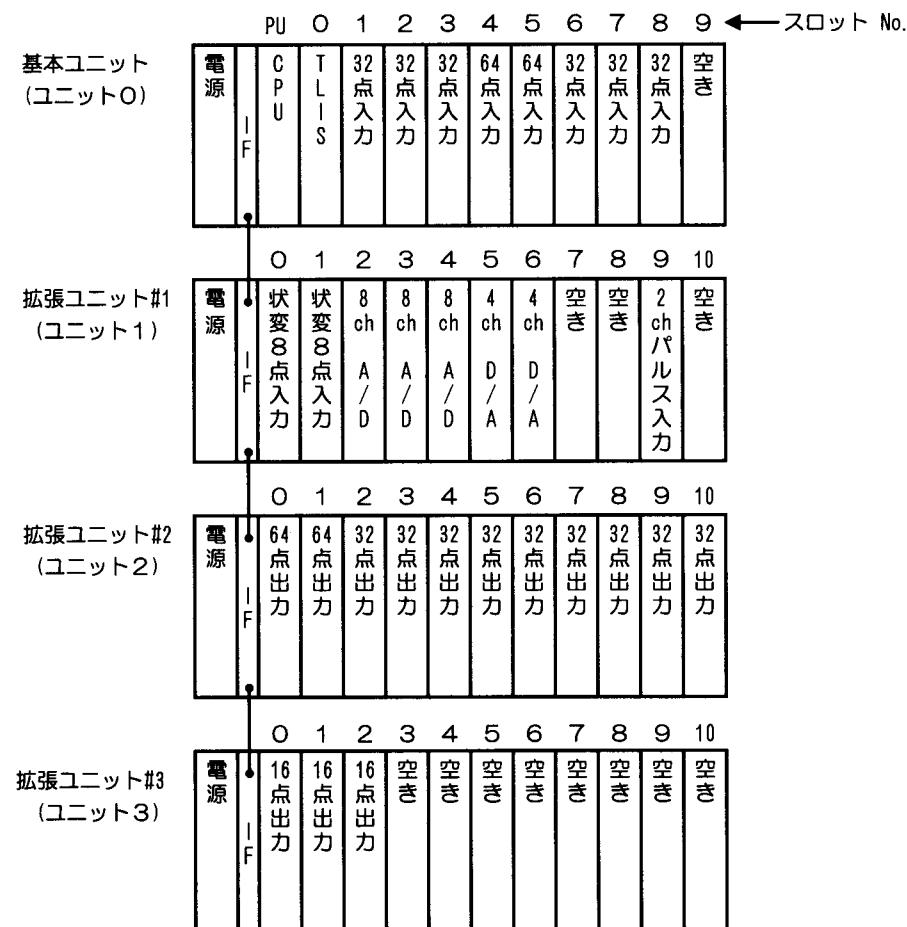
自動割り付けコマンドが実行されると、PC の CPU は実装されている I/O モジュールの状態 (どの位置にどんな種別のモジュールが実装されているか) を読み取り、入出力割り付け情報を作成/登録します。

各々の I/O モジュールは下表に示すモジュール種別を持っています。

形 式	概 要	種 別
DI334/334H	32 点 DC 入力	X 2W
DI335/335H	64 点 DC 入力	X 4W
IN354/364	32 点 AC 入力	X 2W
D0333	16 点 DC 出力	Y 1W
D0334	32 点 DC 出力	Y 2W
D0335	64 点 DC 出力	Y 4W
AC363	16 点 AC 出力	Y 1W
AC364	32 点 AC 出力	Y 2W
R0364	32 点接点出力	Y 2W
R0363S	16 点独立接点出力	Y 1W
AD368	8 チャネルアナログ入力	X 8W
DA364/374	4 チャネルアナログ出力	Y 4W
CD332	状態変化検出付き 8 点 DC 入力	iX 1W
P1312	2 チャネルパルス入力	iX+Y 2W
AS311	ASCII モジュール	iX+Y 4W
MC352	2 軸位置決めモジュール	X+Y 4W
SN321/322/323 /325	TOSLINE-S20 データ伝送装置 (T3/T3H ステーション)	TL-S
MS311/RS311	TOSLINE-F10 データ伝送装置 (T3/T3H ステーション)	TL-F
EN311	イーサネットモジュール	OPT

例えば、下図の I/O モジュール実装状態のときに自動割り付けを実行すると、CPU は実装されている I/O モジュールの種別を読み出して、入出力割り付け情報を作成し、システム情報内に登録します。

- モジュール実装状態



- 入出力割り付け情報

ユニットO		ユニット1		ユニット2		ユニット3	
スロット	モジュール種別	スロット	モジュール種別	スロット	モジュール種別	スロット	モジュール種別
PU		O	iX 1W	O	Y 4W	O	Y 1W
O	TL-S	1	iX 1W	1	Y 4W	1	Y 1W
1	X 2W	2	X 8W	2	Y 2W	2	Y 1W
2	X 2W	3	X 8W	3	Y 2W	3	
3	X 2W	4	X 8W	4	Y 2W	4	
4	X 4W	5	Y 4W	5	Y 2W	5	
5	X 4W	6	Y 4W	6	Y 2W	6	
6	X 2W	7		7	Y 2W	7	
7	X 2W	8		8	Y 2W	8	
8	X 2W	9	iX+Y 2W	9	Y 2W	9	
9		10		10	Y 2W	10	

個別割り付け プログラマの一般 I/O 割り付け情報画面にて、ユーザが入出力割り付け情報を作成し、PC 本体に書き込む方法です。

個別割り付けは次のような場合に使用します。

- I/O モジュールがフル実装されていない状態でプログラミングを行うとき
- 特定のモジュールを一括入出力の対象から外したいとき
- ユニット先頭アドレス設定を行うとき
- 将来の追加のため、空きスロットに所定のレジスタ数を割り当てておくとき
- オフラインプログラミングを行うとき

個別割り付けでは、スロット毎にモジュール種別を設定していきます。このとき設定できるモジュール種別は下表の通りです。なおモジュール種別は、機能区分と占有レジスタ数の組み合わせで表現します。(MMR, TL-S, TL-F を除く)

機能区分	占有レジスタ数	備 考
X	01, 02, 04, 08, 16, 32	入力(一括入出力対象)
Y	01, 02, 04, 08, 16, 32	出力(一括入出力対象)
X+Y	02, 04, 08, 16, 32, 64, 128	入力+出力(一括入出力対象)
iX	01, 02, 04, 08, 16, 32	入力(一括入出力対象外)
iY	01, 02, 04, 08, 16, 32	出力(一括入出力対象外)
iX+Y	02, 04, 08, 16, 32, 64, 128	入力+出力(一括入出力対象外)
Z	08, 16, 32	
SP	01, 02, 04, 08, 16, 32, 64, 128	スペース
MMR	—	メモリタイプ
TL-S	—	TOSLINE-S20用
TL-F	—	TOSLINE-F10用

- (1) X及びiXは入力モジュールに、Y及びiYは出力モジュールに、X+Y及びiX+Yは入出力混合モジュールに各自割り付けます。なお、i指定を付けたモジュールに対応する入出力レジスタは、一括入出力の対象外となります。
- (2) SPは空きスロットに任意のレジスタ数を割り付けておくときに使用します。出力レジスタ(YW)が割り当てられます。
- (3) MMRはI/Oメモリカードを拡張メモリとして使用するときにCPUスロットに設定します。
- (4) TL-Sはデータ伝送装置TOSLINE-S20に割り付けます。
- (5) TL-Fはデータ伝送装置TOSLINE-F10に割り付けます。
- (6) ZはT3/T3Hシステムでは不使用です。

注意

個別割り付けを行うことによって、入出力割り付け情報を自由に作成/登録することができますが、RUN 起動させるためには登録された入出力割り付け情報と I/O モジュールの実装状態が一致していることが必要です。なお、強制運転コマンドを実行したときには、割り付け情報に登録されたモジュールが未実装の状態でも運転(RUN-F モード)は可能です。ただしこの場合も、登録されたモジュールと異なる種別のモジュールが実装されているときには運転は行えません(I/O 照合チェック異常)。

ユニット先頭アドレス設定 個別割り付けにおいて、各ユニットの先頭レジスタアドレス(入出力レジスタ)を設定し登録することができます。

この機能を使用することによって、ユニット毎にレジスタアドレスを整理することができます。また、将来空きスロットに I/O モジュールを追加する場合にも、他のユニットのレジスタアドレスには影響を及ぼさないようにすることができます。

(ユニット先頭アドレス設定/表示画面)

ユニット0～3の例

ユニット#0	ユニット#1	ユニット#2	ユニット#3
先頭レジスタNo.	先頭レジスタNo.	先頭レジスタNo.	先頭レジスタNo.
[0]	[50]	[100]	[150]

この画面例の場合、

基本ユニットは XW/YW000 から、

拡張ユニット#1 は XW/YW050 から、

拡張ユニット#2 は XW/YW100 から、

拡張ユニット#3 は XW/YW150 から、

アドレスの割り付けが行われます。

補足

後段のユニットが若いレジスタアドレスとなるような設定はできません。

4.3 レジスタとモジュールの対応

自動割り付けまたは個別割り付けを行うことによって入出力割り付け情報が登録されると、レジスタとモジュールの対応は以下のルールによって自動的に決まります。

- (1) ユニット上では、左端のモジュールから順に若いアドレスのレジスタが割り付けられます。
- (2) ユニット先頭アドレスが設定されていない場合は(自動割り付けでは常に設定なし)、前段ユニットに続けてレジスタが割り付けられます。
- (3) モジュール種別が設定されていないスロットは(自動割り付けにおける空きスロットも同様)、レジスタを占有しません。
- (4) I/O 5スロットの基本ベース(BU315)及び I/O 6スロットの拡張ベース(BU356)の場合も、入出力割り付け上は標準サイズのベースと同じ扱いとなり、ユニット後半に、設定なしのスロットがあると見なされます。従ってこの部分はレジスタを占有しません。
- (5) SP(スペース)設定されたスロットには、内部的に出力レジスタが、設定ワード数分割り当てられます。
- (6) Z, MMR, TL-S, TL-F の設定がされたモジュールは、入出力レジスタ(XW/YW)を占有しません。
- (7) I/O モジュールに割り付けられていない入出力レジスタは、プログラミング上は出力レジスタ(YW)となりますので、補助レジスタ/リレー(RW/R)と同様の使い方ができます。

入出力割り付け情報が登録された場合のレジスタの割り付けを以下の例に示します。

例1)

- 入出力割り付け情報

ユニット0		ユニット1		ユニット2		ユニット3	
先頭アドレス[]	モジュール種別						
スロット		スロット		スロット		スロット	
PU		O	iX 1W	O	Y 4W	O	Y 1W
O	X 2W	1	iX 1W	1	Y 4W	1	Y 1W
1	X 2W	2	X 8W	2	Y 2W	2	Y 1W
2	X 2W	3	X 8W	3	Y 2W	3	
3	X 2W	4	X 8W	4	Y 2W	4	
4	X 4W	5	Y 4W	5	Y 2W	5	
5	X 4W	6	Y 4W	6	Y 2W	6	
6	X 2W	7		7	Y 2W	7	
7	X 2W	8		8	Y 2W	8	
8	X 2W	9	iX+Y 2W	9	Y 2W	9	
9		10		10	Y 2W	10	

- レジスタ割り付け

ユニット0		ユニット1		ユニット2		ユニット3	
スロット	レジスタ	スロット	レジスタ	スロット	レジスタ	スロット	レジスタ
PU		O	XW022	O	YW058～YW061	O	YW084
O	XW000, XW001	1	XW023	1	YW062～YW065	1	YW085
1	XW002, XW003	2	XW024～XW031	2	YW066, YW067	2	YW086
2	XW004, XW005	3	XW032～XW039	3	YW068, YW069	3	
3	XW006, XW007	4	XW040～XW047	4	YW070, YW071	4	
4	XW008～XW011	5	YW048～YW051	5	YW072, YW073	5	
5	XW012～XW015	6	YW052～YW055	6	YW074, YW075	6	
6	XW016, XW017	7		7	YW076, YW077	7	
7	XW018, XW019	8		8	YW078, YW079	8	
8	XW020, XW021	9	XW056, YW057	9	YW080, YW081	9	
9		10		10	YW082, YW083	10	

例2)

- 入出力割り付け情報

ユニット0		ユニット1		ユニット2		ユニット3	
先頭アドレス[]		先頭アドレス[]		先頭アドレス[100]		先頭アドレス[150]	
スロット	モジュール種別	スロット	モジュール種別	スロット	モジュール種別	スロット	モジュール種別
PU		0	X 8W	0	X 4W	0	Y 2W
0	iX 2W	1	X 8W	1	X 4W	1	Y 2W
1	iX 2W	2	X 8W	2	Y 4W	2	Y 2W
2	iX 2W	3	SP 8W	3	Y 4W	3	Y 2W
3	SP 4W	4	Y 4W	4	SP 8W	4	Y 1W
4		5	Y 4W	5	X 2W	5	Y 1W
5	iY 2W	6	Y 4W	6	X 2W	6	Y 1W
6	iY 2W	7	Y 4W	7	X 2W	7	Y 1W
7	iY 2W	8	Y 4W	8	X 2W	8	
8	SP 4W	9		9	X 2W	9	
9		10		10		10	

- レジスタ割り付け

ユニット0		ユニット1		ユニット2		ユニット3	
スロット	レジスタ	スロット	レジスタ	スロット	レジスタ	スロット	レジスタ
PU		0	XW020～XW027	0	XW100～XW103	0	YW150, YW151
0	XW000, XW001	1	XW028～XW035	1	XW104～XW107	1	YW152, YW153
1	XW002, XW003	2	XW036～XW043	2	YW108～YW111	2	YW154, YW155
2	XW004, XW005	3	YW044～YW051	3	YW112～YW115	3	YW156, YW157
3	YW006～YW009	4	YW052～YW055	4	YW116～YW123	4	YW158
4		5	YW056～YW059	5	XW124, XW125	5	YW159
5	YW010, YW011	6	YW060～YW063	6	XW126, XW127	6	YW160
6	YW012, YW013	7	YW064～YW067	7	XW128, XW129	7	YW161
7	YW014, YW015	8	YW068～YW071	8	XW130, XW131	8	
8	YW016～YW019	9		9	XW132, XW133	9	
9		10		10		10	

T3Hは拡張ユニットを2系統(最大6ユニット)接続することができます。

例3) 拡張が2系統の場合(T3Hのみ)

- 入出力割り付け情報

ユーニット0		ユーニット1		ユーニット2		ユーニット3		ユーニット4		ユーニット5		ユーニット6	
先頭アドレス []		先頭アドレス [200]		先頭アドレス [300]									
スロット	モジュール種別	スロット	モジュール種別	スロット	モジュール種別								
PU		0	iX 1W	0	Y 4W	0	Y 1W	0	X 8W	0	X 4W	0	Y 2W
0	X 2W	1	iX 1W	1	Y 4W	1	Y 1W	1	X 8W	1	X 4W	1	Y 2W
1	X 2W	2	X 8W	2	Y 2W	2	Y 1W	2	X 8W	2	Y 4W	2	Y 2W
2	X 2W	3	X 8W	3	Y 2W	3		3	SP 8W	3	Y 4W	3	Y 2W
3	X 2W	4	X 8W	4	Y 2W	4		4	Y 4W	4	SP 8W	4	Y 1W
4	X 4W	5	Y 4W	5	Y 2W	5		5	Y 4W	5	X 2W	5	Y 1W
5	X 4W	6	Y 4W	6	Y 2W	6		6	Y 4W	6	X 2W	6	Y 1W
6	X 2W	7		7	Y 2W	7		7	Y 4W	7	X 2W	7	Y 1W
7	X 2W	8		8	Y 2W	8		8	Y 4W	8	X 2W	8	
8	X 2W	9	iX+Y 2W	9	Y 2W	9		9		9	X 2W	9	
9		10		10	Y 2W	10		10		10		10	

- レジスタ割り付け

ユーニット0		ユーニット1		ユーニット2		ユーニット3	
スロット	レジスタ	スロット	レジスタ	スロット	レジスタ	スロット	レジスタ
PU		0	XW022	0	YW058～YW061	0	YW084
0	XW000, XW001	1	XW023	1	YW062～YW065	1	YW085
1	XW002, XW003	2	XW024～XW031	2	YW066, YW067	2	YW086
2	XW004, XW005	3	XW032～XW039	3	YW068, YW069	3	
3	XW006, XW007	4	XW040～XW047	4	YW070, YW071	4	
4	XW008～XW011	5	YW048～YW051	5	YW072, YW073	5	
5	XW012～XW015	6	YW052～YW055	6	YW074, YW075	6	
6	XW016, XW017	7		7	YW076, YW077	7	
7	XW018, XW019	8		8	YW078, YW079	8	
8	XW020, XW021	9	XW056, YW057	9	YW080, YW081	9	
9		10		10	YW082, YW083	10	

ユーニット4		ユーニット5		ユーニット6	
スロット	レジスタ	スロット	レジスタ	スロット	レジスタ
0	XW087～XW094	0	XW200～XW203	0	YW300, YW301
1	XW095～XW102	1	XW204～XW207	1	YW302, YW303
2	XW103～XW110	2	YW208～YW211	2	YW304, YW305
3	YW111～YW118	3	YW212～YW215	3	YW306, YW307
4	YW119～YW122	4	YW216～YW223	4	YW308
5	YW123～YW126	5	XW224, XW225	5	YW309
6	YW127～YW130	6	XW226, XW227	6	YW310
7	YW131～YW134	7	XW228, XW229	7	YW311
8	YW135～YW138	8	XW230, XW231	8	
9		9	XW232, XW233	9	
10		10		10	

4.4

伝送入出力割り付け

データ伝送装置(TOSLINE-S20, TOSLINE-F10)以外のI/Oモジュールについては、前述の入出力割り付けを行うだけで入出力レジスタの割り付けが決まり、一括入出力処理時にI/Oモジュールと入出力レジスタとの間でデータの交換が行われます。

一方、データ伝送装置(TOSLINE-S20, TOSLINE-F10)については、前述の入出力割り付けを行うことによってモジュールの登録は行われますが、モジュール内のメモリとリンクレジスタとの間のデータ交換は行われません。これを行わせるためには、さらに以下の述べる伝送入出力割り付けの設定が必要です。

入出力割り付けと伝送入出力割り付けの両方が正常に設定されているときに、一括入出力処理において、データ伝送装置内のスキャンデータエリアとPC内のリンクレジスタの間でデータ交換が行われます。

TOSLINE-S20 TOSLINE-S20は1024ワードのスキャンデータエリアを持っています。このスキャンデータエリアをPCのリンクレジスタ(W)に割り付けるかどうかの設定を行います。リンクレジスタ(W)は64ワード単位の16ブロックに分割されており、このブロック単位で割り付けの設定を行います。

なお、この伝送入出力割り付け上のブロックは、TOSLINE-S20の送信/受信エリアとは関係ありません。リンクレジスタ(W)とスキャンデータエリア間のデータ転送の方向は、TOSLINE-S20の送信/受信エリアの設定に応じて、PCのCPUが自動的に決定します。

4

割り付けの設定は次の3種類があります。

設 定	内 容
設定なし (プランク)	割り付けなし
LINK	リンクレジスタ(W)とスキャンデータエリアとの間でデータ転送を行う。(データ転送の方向は自動判別)
GLOBAL	1台のPCに2枚のTOSLINE-S20を装着し、2系統間の中継機能をPCに持たせるときこの設定を行う。

補足

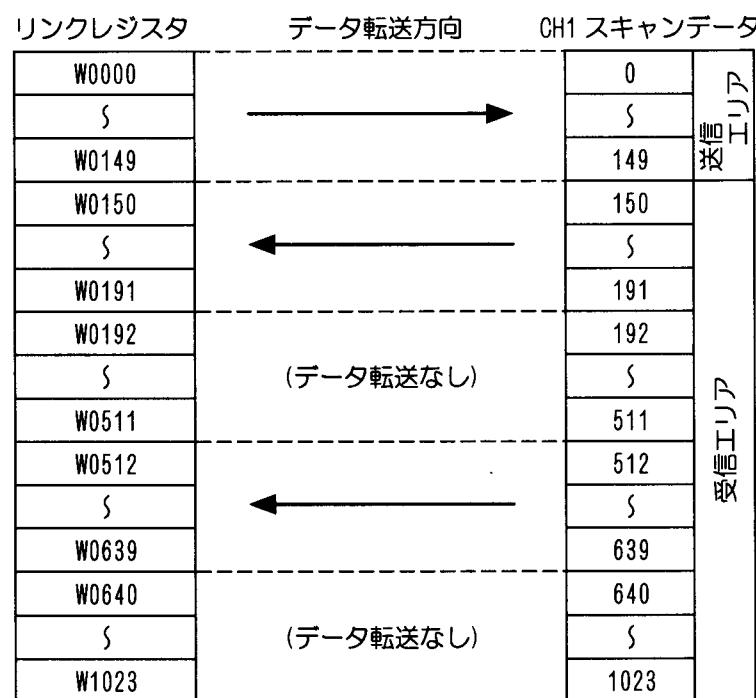
1台のPCにはTOSLINE-S20を2枚装着することができます。このとき、PCのCPUに近い方のTOSLINE-S20がCH1となり、他方がCH2となります。

(1) TOSLINE-S20 が1枚のとき(CH1のみ存在)

- 伝送入出力割り付け側

ブロック	対応リンクレジスタ	CH1	CH2
1	W0000～W0063	LINK	
2	W0064～W0127	LINK	
3	W0128～W0191	LINK	
4	W0192～W0255		
5	W0256～W0319		
6	W0320～W0383		
7	W0384～W0447		
8	W0448～W0511		
9	W0512～W0575	LINK	
10	W0576～W0639	LINK	
11	W0640～W0703		
12	W0704～W0767		
13	W0768～W0831		
14	W0832～W0895		
15	W0896～W0959		
16	W0960～W1023		

- 送信/受信エリアの設定例と上記割り付けとの関連

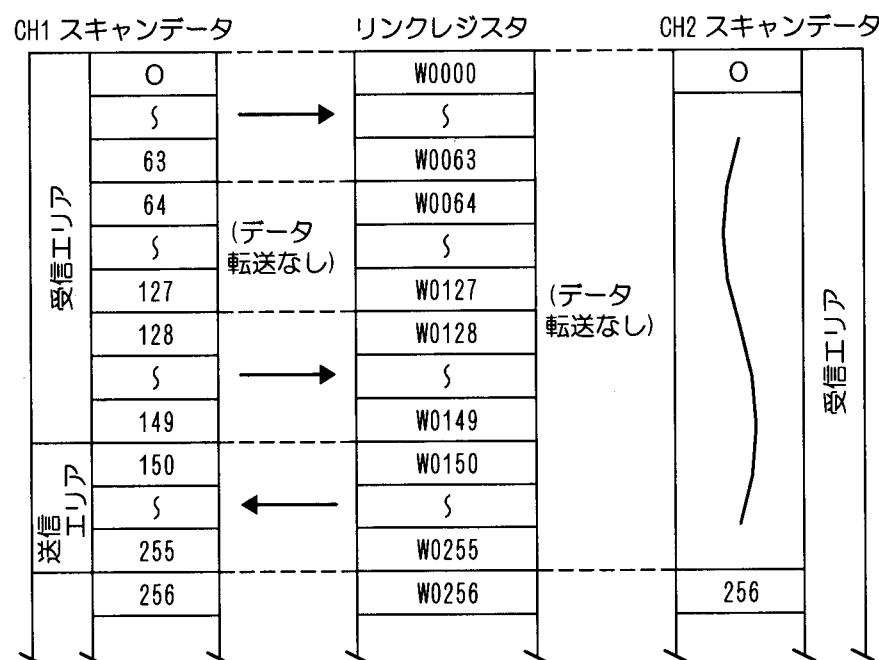


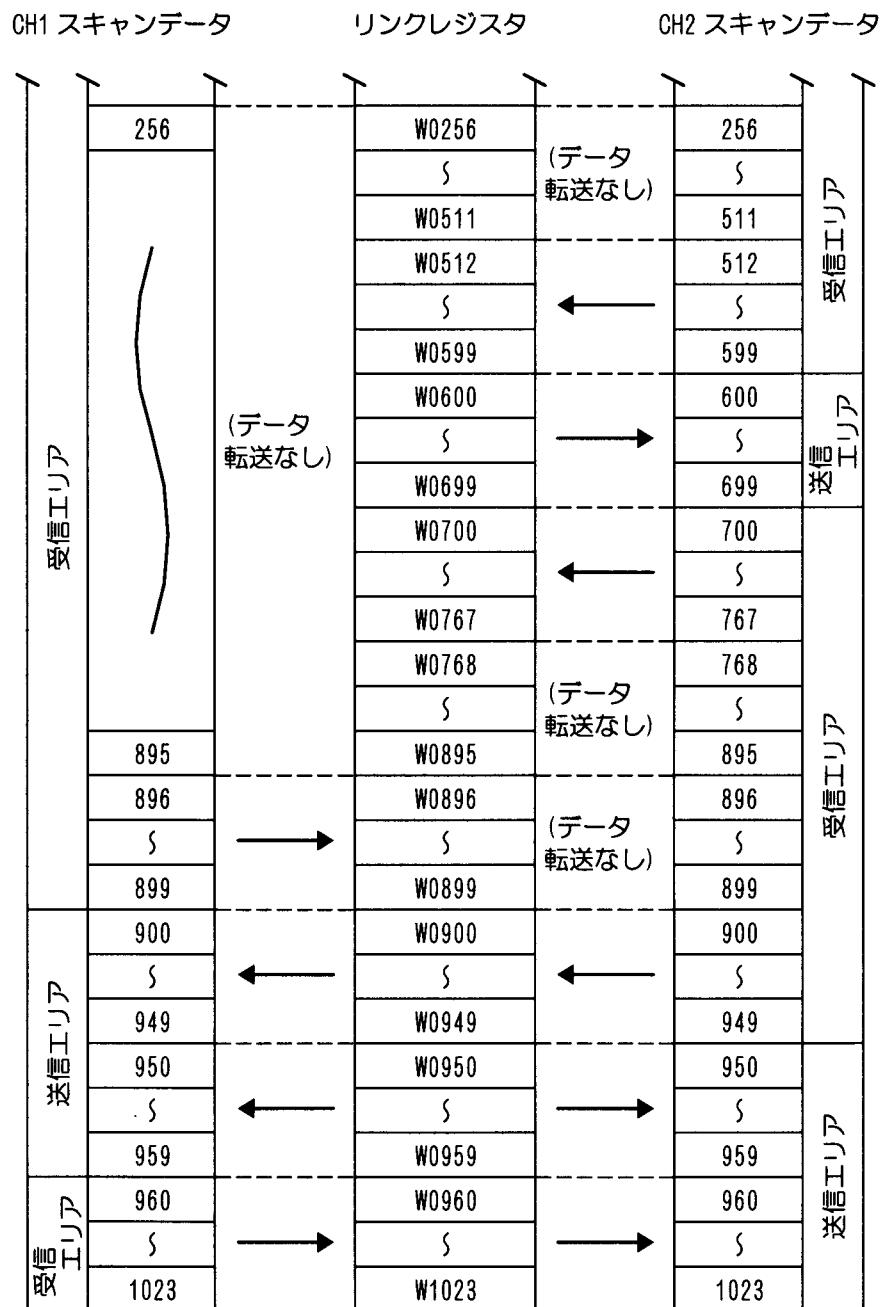
(2) T3においてTOSLINE-S20が2枚のとき(CH1, CH2)

• 伝送入出力割り付け側

ブロック	対応リンクレジスタ	CH1	CH2
1	W0000～W0063	LINK	
2	W0067～W0127		
3	W0128～W0191	LINK	
4	W0192～W0255	LINK	
5	W0256～W0319		
6	W0320～W0383		
7	W0384～W0447		
8	W0448～W0511		
9	W0512～W0575		LINK
10	W0576～W0639		LINK
11	W0640～W0703		LINK
12	W0704～W0767		LINK
13	W0768～W0831		
14	W0832～W0895		
15	W0896～W0959	GLOBAL	
16	W0960～W1023	GLOBAL	

• 送信/受信エリアの設定例と上記割り付けとの関連





補足

- (1) GLOBAL 設定の領域では、受信側の CH からスキャンデータが読み出され、リンクレジスタ及び他方送信側の CH に転送されます。
- (2) GLOBAL 設定の領域で、CH1, CH2 共送信エリアの場合は、リンクレジスタのデータが両 CH に転送されます。
- (3) GLOBAL 設定の領域で、CH1, CH2 共受信エリアの場合は、GLOBAL の設定が行われている CH のデータがリンクレジスタに転送されます。
- (4) 1 つのブロックに対して、CH1 と CH2 の設定が重ならないように注意する必要がありますが、もし重なった場合には CH1 が優先します。

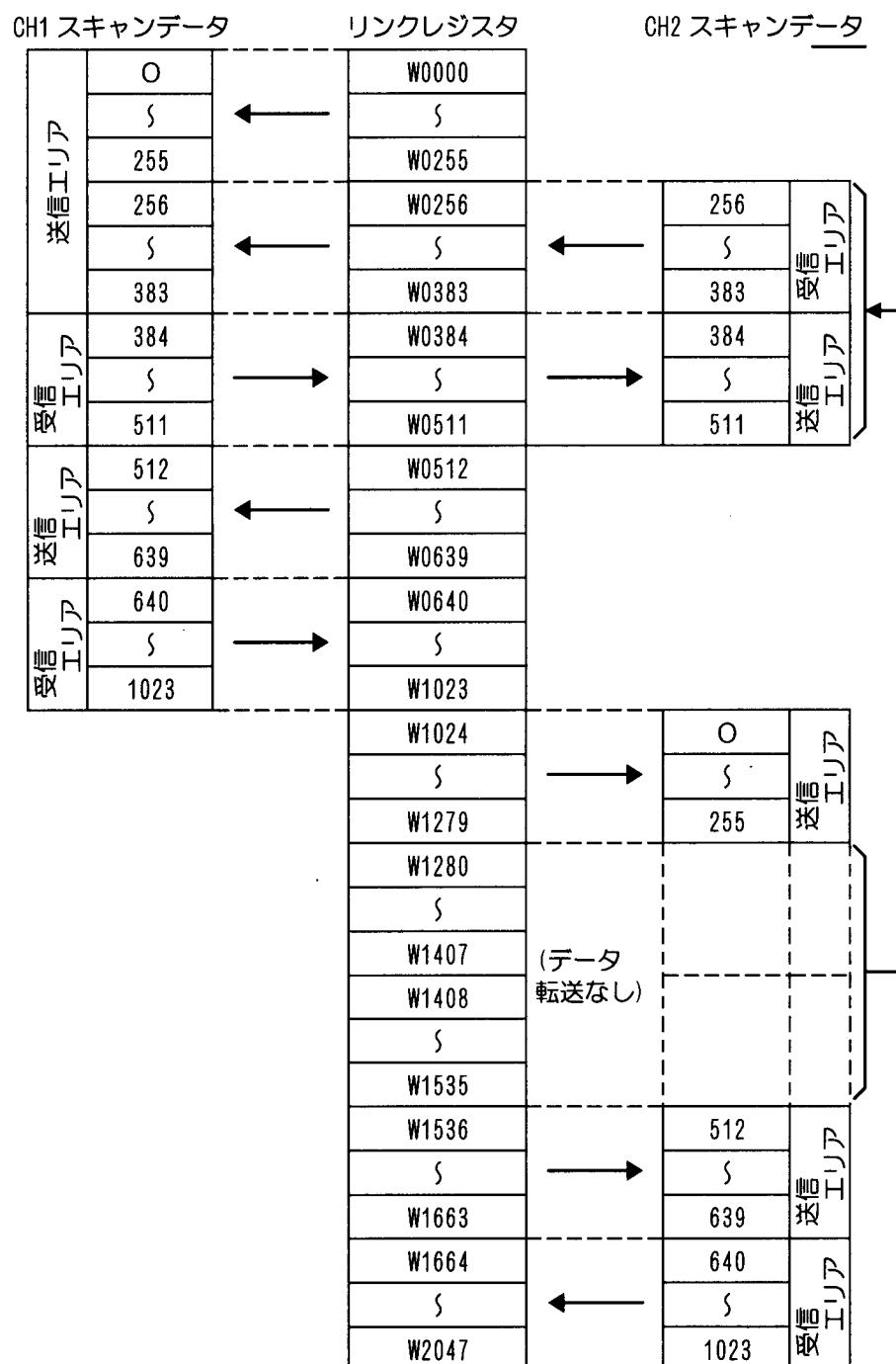
(3) T3HにおいてTOSLINE-S20が2枚のとき(CH1, CH2)

T3Hにおいては、リンクレジスタWは2048ワードあります。2枚のTOSLINE-S20に2048ワードのリンクレジスタを割り当てる場合は、次の例のように割付けます。

• 伝送入出力割り付け例

ブロック	対応リンクレジスタ	CH1	CH2
1	W0000～W0063	LINK	
2	W0064～W0127	LINK	
3	W0128～W0191	LINK	
4	W0192～W0255	LINK	
5	W0256～W0319	GLOBAL	
6	W0320～W0383	GLOBAL	
7	W0384～W0447	GLOBAL	
8	W0448～W0511	GLOBAL	
9	W0512～W0575	LINK	
10	W0576～W0639	LINK	
11	W0640～W0703	LINK	
12	W0704～W0767	LINK	
13	W0768～W0831	LINK	
14	W0832～W0895	LINK	
15	W0896～W0959	LINK	
16	W0960～W1023	LINK	
17	W1024～W1087		LINK
18	W1088～W1151		LINK
19	W1152～W1215		LINK
20	W1216～W1279		LINK
21	W1280～W1343		
22	W1344～W1407		
23	W1408～W1471		
24	W1472～W1535		
25	W1536～W1599		LINK
26	W1600～W1663		LINK
27	W1664～W1727		LINK
28	W1728～W1791		LINK
29	W1792～W1855		LINK
30	W1856～W1919		LINK
31	W1920～W1983		LINK
32	W1984～W2047		LINK

- 送信/受信エリアの設定例と割り付けとの関連



補足

- (1) 1枚目(CH1)のTOSLINE-S20に対応する割付けはブロック1～16(W0000～W1023)、2枚目(CH2)に対応する割付けはブロック17～32(W1024～W2047)です。
CH2に対してブロック1～16を、CH1に対してブロック17～32を割付けることはできません。
- (2) GLOBAL 設定の領域では、受信側のCHからスキャンデータが読み出され、リンクレジスタ及び他方送信側のCHに転送されます。
- (3) GLOBAL 設定を行った場合、GLOBAL 設定を行っていないCHのリンクレジスタとGLOBAL領域ブロックとのデータのやりとりは行いません。
- (4) GLOBAL 設定の領域で、CH1, CH2共送信エリアの場合は、リンクレジスタのデータが両CHに転送されます。
- (5) GLOBAL 設定の領域で、CH1, CH2共受信エリアの場合は、GLOBALの設定が行われているCHのデータがリンクレジスタに転送されます。

TOSLINE-F10 TOSLINE-F10 は 32 ワード(512 点)のスキャンデータエリアを持っています。また 1 台の PC には 8 枚まで TOSLINE-F10 を装着することができます。この場合、PC の CPU に近いものから順に CH1, CH2, …CH8 となります。TOSLINE-F10 に対する伝送入出力割り付けは、存在する CH 全てについて、LINK の設定を行います。これにより、リンクリレーレジスタ(LW)の若いアドレスから 32 ワード単位で、CH1, CH2, … の順に割り付けられます。

- TOSLINE-F10 が 4 枚のときの伝送入出力割り付け例

CH	設 定	リンクリレーレジスタ割り付け
1	LINK	LW000～LW031
2	LINK	LW032～LW063
3	LINK	LW064～LW095
4	LINK	LW096～LW127
5		
6		
7		
8		

リンクリレーレジスタ(LW)と TOSLINE-F10 のスキャンデータとの間のデータ転送の方向は、TOSLINE-F10 の接続ステーション構成に応じて PC の CPU が自動的に決定します。

つまり、LW000 に対応する接続ステーションが例えばリモート入力ユニットであるときには、データ転送の方向は、スキャンデータ(0)→LW000 となり、リモート出力ユニットであるときには、LW000→スキャンデータ(0)となります。

■ 補足

データ伝送装置(TOSLINE-S20, TOSLINE-F10)の使用方法については、各々別冊の説明書をご覧下さい。

5.1

プログラム言語の
概要

PCは、ユーザプログラムを構成するプログラム言語として、ラダー図、SFCの2種類をサポートしています。プログラムのブロックを切り換えることにより、1つのユーザプログラムで複数のプログラム言語を混在使用することができますので、制御機能に最適のプログラム構成をとることが可能です。

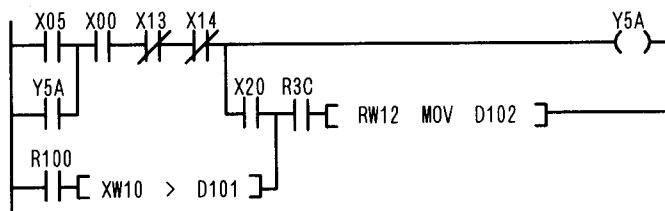
(1) ラダー図

PCのプログラム言語の中心となる言語です。リレーシンボルとファンクションブロックの組み合わせによってプログラムを構成します。条件制御、時間制御に適した言語です。

リレーシンボル …… a接点、b接点、コイルなどがあります。

ファンクション …… 1つの機能を表す箱型の命令です。リレーの接点のような扱いでラダー図のネットワーク中に自由に配置し、ファンクションブロックの出力を他のファンクションブロックの入力とすることができます。

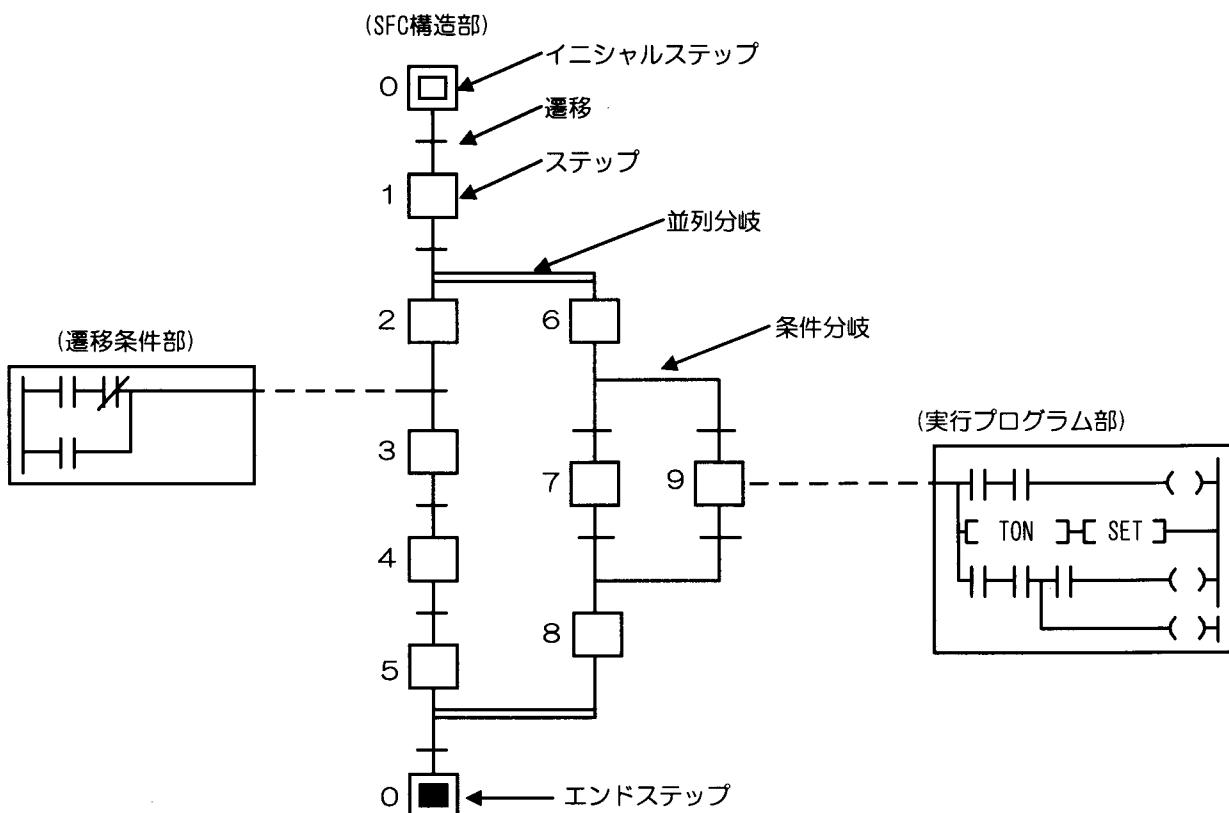
例)



(2) SFC(シーケンシャルファンクションチャート)

工程歩進制御(順序制御)に適したプログラム言語です。また制御の流れが見やすい言語ですので、プログラムのメンテナンス及び標準化にも有効です。SFCは、制御の流れを示すSFC構造部、各工程での動作を示す実行プログラム部、及び工程を進める条件を示す遷移条件部から成ります。実行プログラム部と遷移条件部はラダー図で作成します。

SFCとは、単独のプログラム言語というよりも、むしろ制御工程や条件を整理し、プログラムを見やすくするための実行制御要素と考えることができます。



制御の流れは、イニシャルステップから下方に向かって進み、エンドステップまで到達すると再びイニシャルステップに戻ります。

ステップは動作工程に対応し、各々のステップに対応する実行プログラム部があります。あるステップから次のステップに移る条件を遷移と呼び、各々の遷移に対応する遷移条件部があります。直前のステップが実行状態でかつ遷移条件がONしたときに、直前のステップは実行停止となり、次のステップが実行状態となります。

各プログラム種別/部分で使用可能なプログラム言語を下表に示します。

プログラム種別/部分	ラダー図	SFC
メインプログラム	○	○
サブプログラム	○	△
割り込みプログラム	○	×
サブルーチン	○	×*
SFC 実行プログラム部	○	×*
SFC 遷移条件部	○	×

○ : 使用可

× : 使用不可

△ : T3H のみ使用

- *) SFC を階層構造とすること (SFC の1つのステップに他の SFC を対応付けること) は可能です。この場合、マクロステップ (SFC のサブルーチンに相当) を使用します。

5.2

ラダー図

PCでは、ラダー図、SFCの2種類のプログラム言語を混在使用することができますが、このうちラダー図は、ユーザプログラムの中に必ず存在しなければならない基本言語です。

ここでは、ラダー図のプログラムについて、構成、実行順序、ラダー図命令の一般的な事項について説明します。

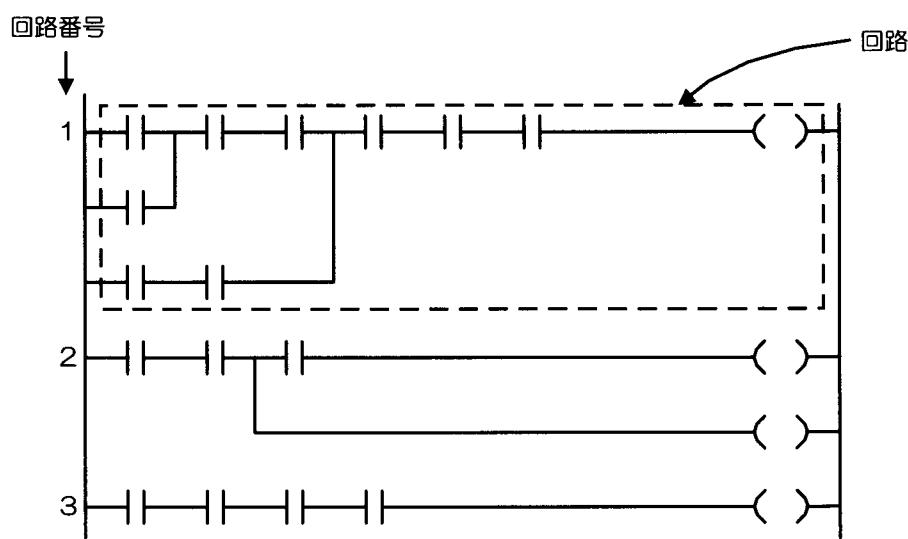
前に説明したように、ユーザプログラムはプログラム種別という機能的な種別毎に登録されます。さらに各プログラム種別の中では、ユーザプログラムは、1つまたは複数のブロックという単位で登録されます。

プログラム種別 ...	メインプログラム、サブプログラム #1～#4、定期割り込みプログラム、I/O 割り込みプログラム#1～#8、サブルーチン
ブロック	ブロック 1～256(1 言語/1 ブロック)

新しく登録するブロックのプログラミングを始める際には、そのプログラムをどの言語で作成するかを指定します（言語指定と呼びます）。

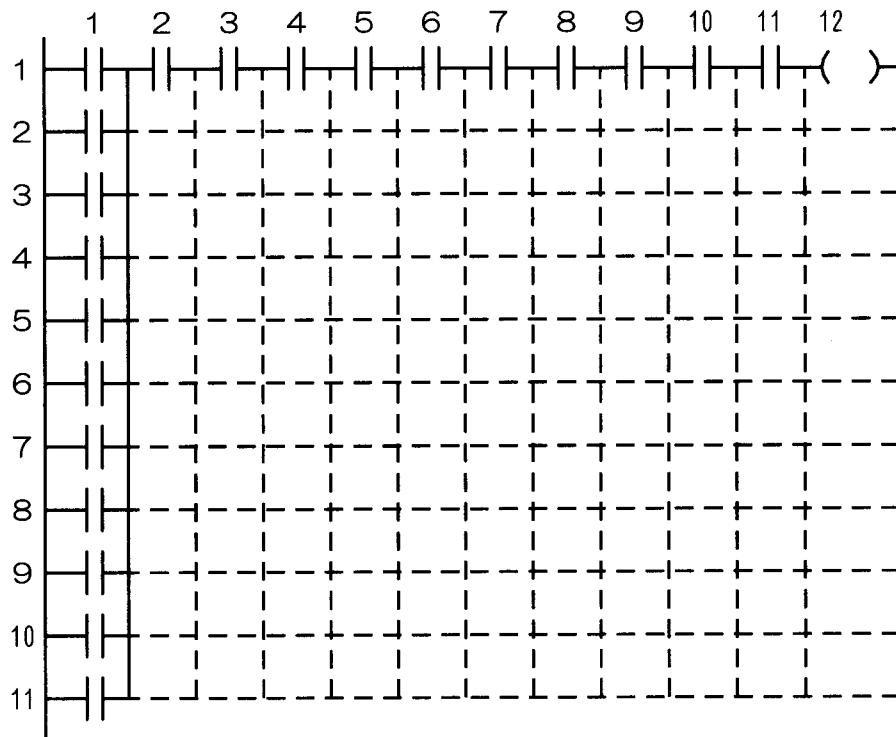
ただし、ラダー図の場合には言語指定の操作は不要です（デフォルトはラダー図となっています）。

1つのブロックの中では、ラダー図のプログラムは回路という単位で登録/管理されます。回路とは、下図のように、互いに接続線によって結合された1つのネットワークとして定義します。



回路番号は1から始まる一連の番号(10進数)であり、途中で回路番号を飛ばすことはできません。(最大回路番号 65024←実質無制限)

1つの回路の大きさは、下図のように11行×12列(最大132ステップ)に制限されます。

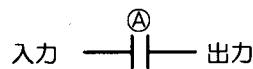


ラダー図は、ハードワイヤードのリレーシーケンスと同様のイメージで、リレーシンボルをベースとしてプログラムを作成する言語です。T3では効率的なデータ処理プログラムを可能とするために、リレーシンボルとファンクションブロックを組み合わせたラダー図を採用しています。

リレーシンボル……a接点、b接点、コイル、及び特別の機能を持たせた接点とコイルです。これらの1つ1つを命令と呼びます。

5

例) a接点



デバイスⒶがONのとき入力側と出力側が導通となる。

プログラム実行の面からみると、入力がONでかつデバイスⒶの内容がONのとき出力をONにする、という動作となる。

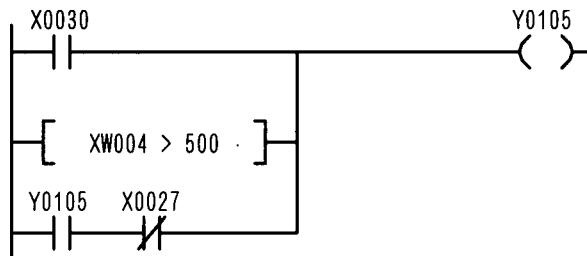
ファンクションブロック …… 1つの機能を示す箱として表現します。機能の種類としては、データ転送、四則演算、論理演算、比較判定、各種関数などがあります。これらの1つ1つを命令語と呼びます。ファンクションブロックには、1つまたは複数の入力と1つの出力があり、入力の状態がある条件を満たしたときに所定の機能を実行し、実行の結果によって出力のON/OFFを決定します。

例1) 加算命令

入力 $\rightarrow [\textcircled{A} + \textcircled{B} \rightarrow \textcircled{C}] \leftarrow$ 出力

入力がONのとき、レジスタ \textcircled{A} の内容と \textcircled{B} の内容を加算し、結果をレジスタ \textcircled{C} に格納する。加算の結果、オーバフローまたはアンダーフローが発生したら出力をONにする。

例2) リレーションボルとファンクションブロックの組み合わせ



X0030がONかまたはXW004の内容が500を超えるときY0105がONとなる。X0030がOFFかつXW004の内容が500以下となってもY0105はONのままで、X0027がONになったときにY0105はOFFとなる。

5

補足

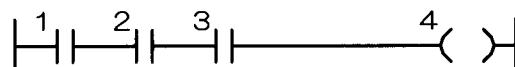
- (1) ファンクションブロックは、ラダー図の動作として考えると特別な機能を持った接点のように考えることができます。命令の実行順序に注意してファンクションブロックを配置することにより、複雑な制御機能を分かりやすいプログラムで実現することができます。
- (2) ラダー図の命令語一覧を5.5節に示します。なお各命令語の詳細仕様については、別冊の命令語説明書をご覧下さい。

命令実行順序 ラダー図によって構成されたブロック内での命令語実行順序を以下に示します。

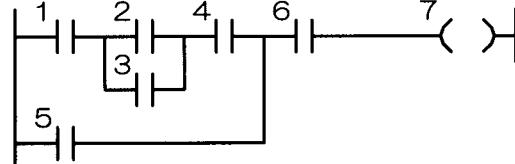
① 回路1, 回路2, 回路3, … の順にブロック内最終回路まで実行されます
(END 命令のあるブロックの場合は END 命令のある回路まで)。

② 1つの回路上では次のルールに従って実行されます。

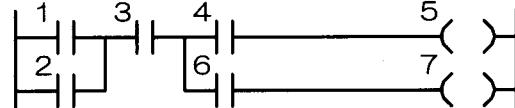
(1) 上下接続がないときは
左から右へ実行されます。



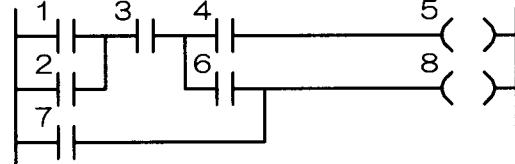
(2) OR 接続があるときは、OR
論理部が先に実行され
ます。



(3) 分流があるときは、上の
行から下の行の順に実
行されます。



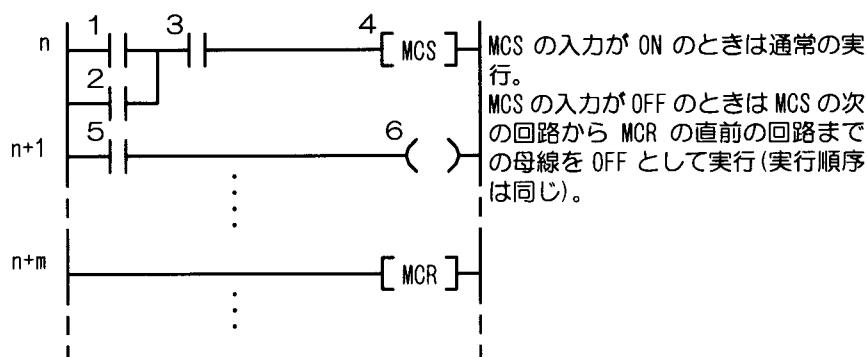
(4) 上記(2)と(3)の組み合
わせ



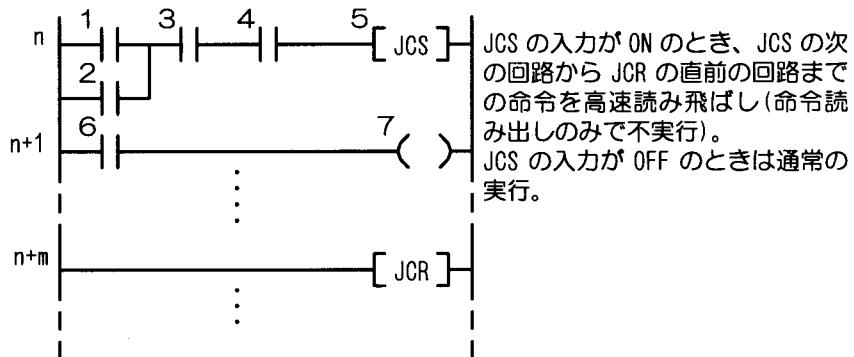
ファンクション命令を含んだ場合の命令実行順序も基本的に上記のルールに従います。ただし、プログラム実行制御命令については、各命令の仕様によります。

主なプログラム実行制御命令を組み合わせた場合の実行順序を以下に示します。

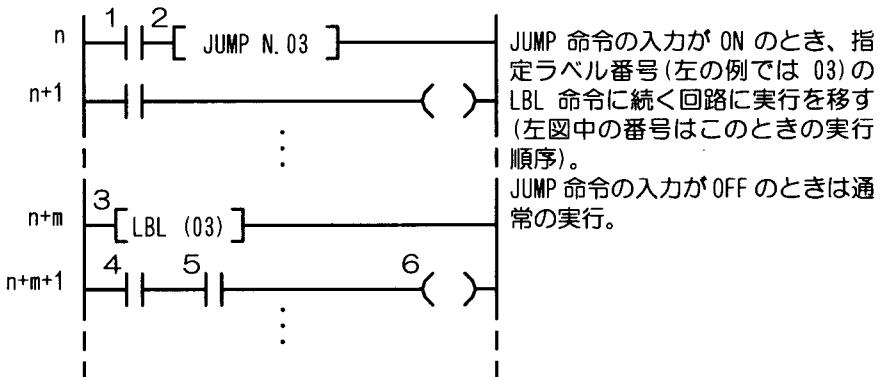
- マスターントロール(MCS/MCR, MCSn/MCRn)



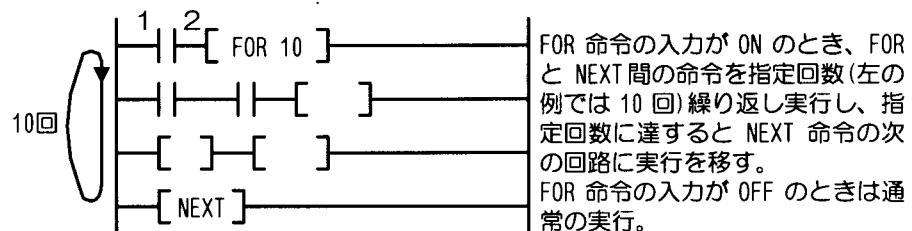
- ジャンプコントロール (JCS/JCR)



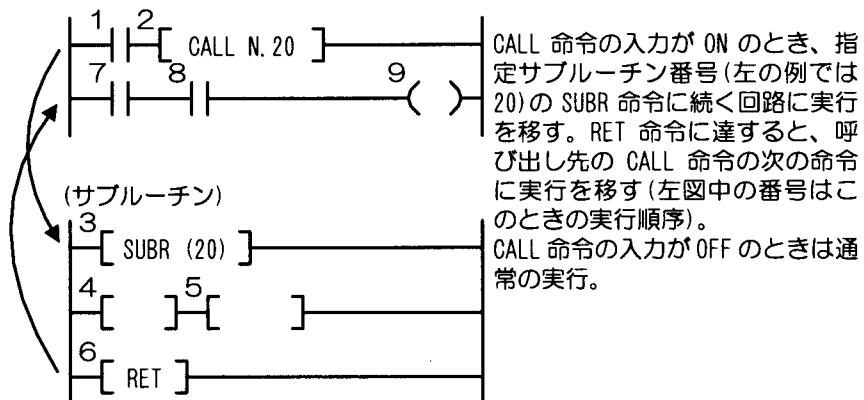
• 条件ジャンプ(JUMP/LBL)



• 繰り返し(FOR/NEXT)



- サブルーチン(CALL/SUBR/RET)



ラダー図命令全般事項 ラダー図でプログラムを作成するために、全般的に必要と思われる事項について以下に列記します。

- (1) 全てのプログラム種別において、少なくとも1つのブロックはラダー図で作成する必要があります。つまりメインプログラム及び各サブプログラムの最終はラダー図の END 命令によって判断されます。また、各割り込みプログラムの最終はラダー図の IRET 命令によって判断されます。また、サブルーチンの入口と出口はラダー図の SUBR 命令と RET 命令によって構成する必要があります。
- (2) リレーシンボル形の各命令にタイマ命令(4種)、カウンタ命令、ジャンプ命令、マスターコントロール命令及び END 命令を加えた命令群を基本命令と称します。
- (3) 基本命令以外の命令をファンクション命令と称します。ファンクション命令は各々個別のファンクション番号(FUN No.)を持っています。なお、同じファンクション番号の命令でも、下記のように実行条件の選択が可能です。(選択できない命令も1部あります)

通常 命令の入力が ON の間毎スキャン実行します。

パルス化 命令の入力が OFF から ON に変化したスキャンだけ実行します。

例) データ転送命令



R0000 が ON の間毎スキャン MOV 命令(D1000 に 10 を代入)を実行する。

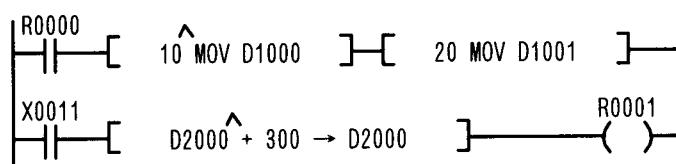


R0000 が OFF から ON に変化したスキャンだけ MOV 命令(D1000 に 10 を代入)を実行する。

5

なお、パルス化したファンクション命令の後(右側)には命令語を配置することはできません。

例)



これらの2つの回路はいずれも作成不可

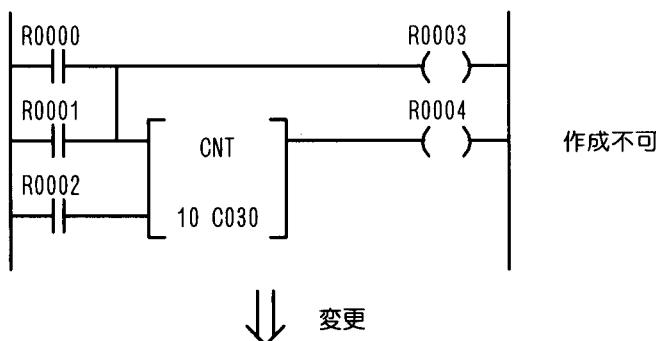
(4) 1つの命令が必要とするステップ数は命令の種類によって異なります。

また、同じ命令でもオペランドに桁指定を使用するかどうか、倍長オペランドに定数を使用するカレジスタを使用するか、などによって占有ステップ数は変化します(1~10ステップ/1命令)。

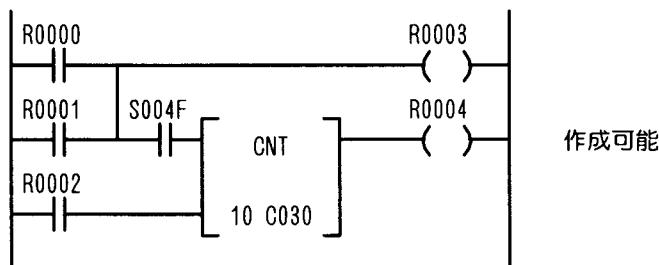
なお、縦接続線と横接続線は基本的にステップ数を必要としません。

(5) 複数の入力を持つ命令において、入力の直前に合流/分流の縦接続線を置くことはできません。この場合には、ダミー接点(常時ONの特殊リレーS004Fのa接点など)を入力の直前に入れて下さい。

例)

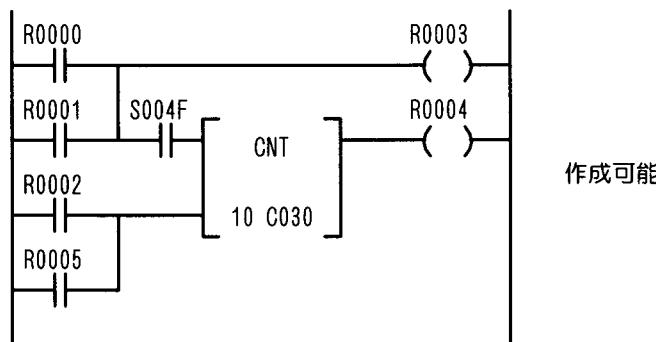


↓ 変更



なお、複数入力のうち、最下段の入力については上記の配慮は不要です。

例)



5.3

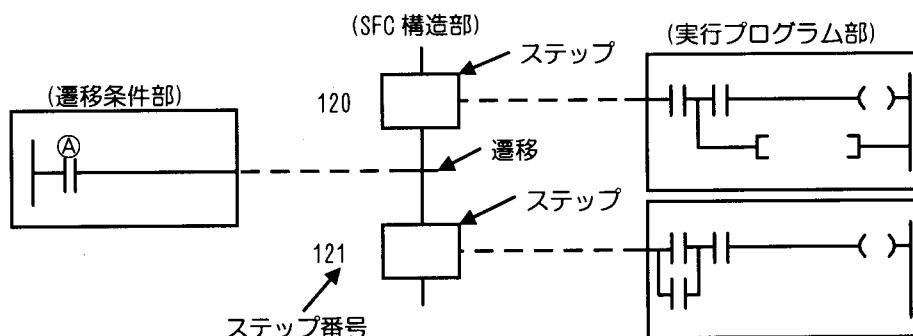
SFC SFC は、シーケンシャルファンクションチャートの略で、工程歩進制御(順序制御)に適したプログラム言語です。PC では一般の SFC に次の機能を付加した SFC を採用しています。

- ジャンプ …… 条件成立時任意のステップに実行を移す
- 待ち時間付きステップ …… 遷移条件が成立しても設定時間経過まではステップ遷移しない
- 渋滞検出付きステップ …… 設定時間が経過しても次ステップに遷移しないとき指定デバイスを ON する

SFC はメインプログラム及びサブプログラムに使用することができます。

ここでは、SFC の全体構成、SFC の構成要素、プログラム作成上の注意事項などについて説明します。

SFC プログラムは、SFC 構造部、実行プログラム部、遷移条件部から構成されます。



SFC 構造部は、制御動作の流れを規定するもので、ステップと遷移を基本要素とします。ステップは上記のように 1 つの箱で表現され、各々のステップは固有のステップ番号を持ちます。また、ステップには 1 対 1 に対応する実行プログラムが付随します。

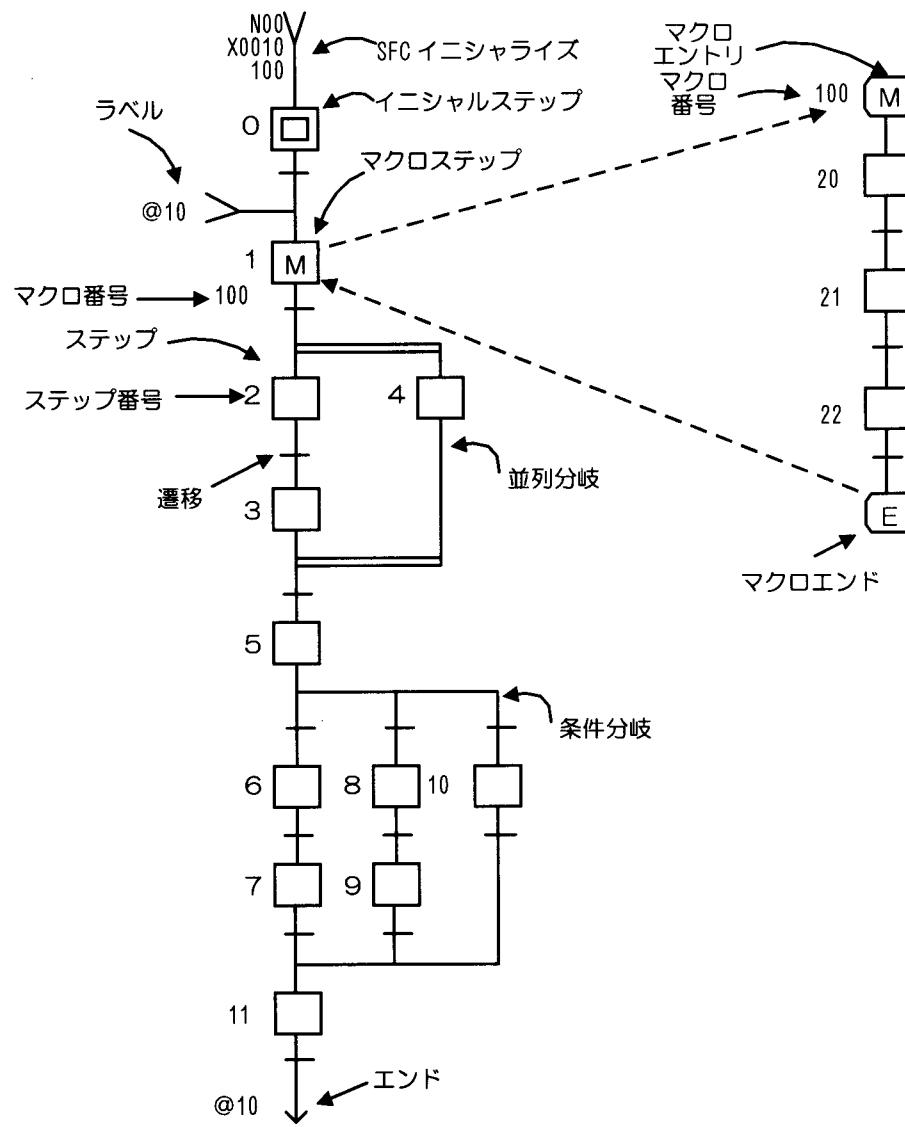
ステップはアクティブ(活性)とインアクティブ(不活性)の 2 つの状態を持ち、ステップがアクティブなときに、対応する実行プログラムの母線が活きた状態(母線 ON)となります。ステップがインアクティブなときには、対応する実行プログラムの母線が切れた状態(母線 OFF)となります。

一方遷移は、ステップとステップの間に位置し、直前(上側)のステップから次(下側)のステップへアクティブ状態を遷移させるための条件を表わします。遷移には、1 対 1 に対応する遷移条件部が付随します。

例えば上の図で、ステップ 120 がアクティブなとき、ステップ 120 に対応する実行プログラムの母線が ON となります。この状態でデバイスⒶが ON すると遷移条件成立となり、ステップ 120 がインアクティブ、ステップ 121 がアクティブとなります。これに伴い、ステップ 120 に対応する実行プログラムの母線は OFF となり(母線 OFF で実行されます)、ステップ 121 に対応する実行プログラムの母線が ON となります。

全体構成 PC に作成することのできる SFC プログラムの全体構成を以下に説明します。

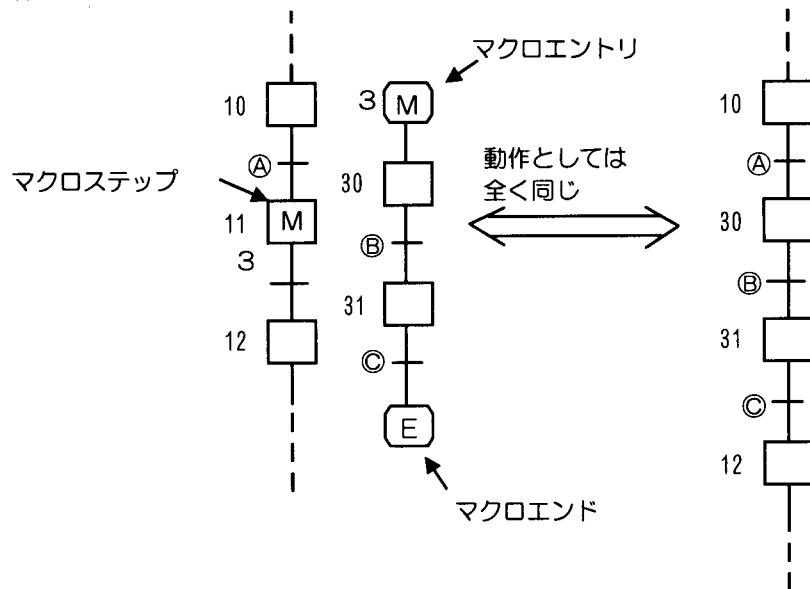
(SFC メインプログラム) (マクロプログラム)



SFC プログラム全体は、SFC メインプログラムとマクロプログラムに分けて考えることができます。

SFC メインプログラムとは、イニシャルステップを持ち、エンドまたはエンドステップで終わる一連の SFC プログラムです。PC では、最大 64 本の SFC メインプログラムを作成することができます。(プログラム種別メインプログラム、サブプログラム、割込プログラム(T3H のみ)合計)

一方、マクロプログラムは、マクロエントリから始まりマクロエンドで終わる一連の SFC プログラムです。マクロプログラムは各々個別のマクロ番号を持ち、SFC メインプログラムまたは他のマクロプログラムに存在するマクロステップに、1 対 1 に対応します。マクロプログラムは、SFC プログラムを階層構造にしてプログラムを見やすくするために使用します。マクロプログラムは全体で 128 本まで作成することができます。



補足

- (1) マクロプログラム内にマクロステップを作成すること (SFC多段階層) も可能です。段数の制限はありません。
- (2) マクロプログラムとマクロステップは 1 対 1 に対応している必要があります。つまり、同一のマクロ番号を指定したマクロステップを複数箇所で使用することはできません。

SFC のプログラミングは、ブロックを指定し、さらに言語指定にて SFC を選択することによって可能となります。

1 つのブロックには、SFC メインプログラムまたはマクロプログラムが 1 本だけ作成できます。 (SFC1 本/ブロック)

また、1 ブロック当りの SFC ステップの個数は最大 128 個です。言い換えると、SFC メインプログラム及びマクロプログラムの 1 本当りのステップの個数は 128 個までということになります。 (命令語ステップ数によっても制限されます。後述)

5

補足

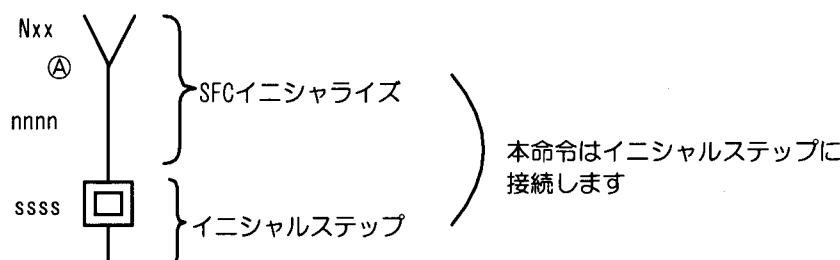
マクロプログラムは、呼び出し元の SFC メインプログラムより後方 (ブロック番号の大きい方) に配置して下さい。

SFC 構成要素 SFC プログラムを構成する要素について以下に説明します。

(1) SFC イニシャライズ

指定範囲のステップをインアクティブにして、指定のイニシャルステップに起動をかける(アクティブにする)機能です。SFC 命令とラダー図命令の2つの方法があり、どちらかを使用します。SFC メインプログラム1本に1つのSFC イニシャライズが必要です。

① SFC 命令



オペランド: xx =プログラム番号(0~63)

\textcircled{A} =起動デバイス(T., C. 以外)

$nnnn$ =初期化ステップ個数(1~4096)

機能: \textcircled{A} で指定されるデバイス(タイマリレー、カウンタリレーを除く)が OFF から ON に変化したとき、下に続くイニシャルステップ(ssss)から $nnnn$ で指定される個数分のステップ(ステップ番号 ssss から $ssss+nnnn-1$)をインアクティブにして、イニシャルステップ(ssss)をアクティブにします。

② ラダー図命令

入力——[SFIZ(nnnn) ssss]——出力

オペランド: $nnnn$ =初期化ステップ個数(1~4096)

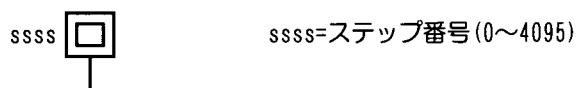
$ssss$ =イニシャルステップのステップ番号(0~4095)

機能: 入力が OFF から ON に変化したとき、 $ssss$ で指定されるステップ番号から $nnnn$ で指定される個数分のステップ(ステップ番号 ssss から $ssss+nnnn-1$)をインアクティブにして、 $ssss$ で指定されるイニシャルステップをアクティブにします。

(2) イニシャルステップ

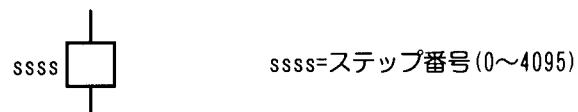
SFC メインプログラムの開始を示すステップです。固有のステップ番号を持ち、1対1に対応する実行プログラムを持つことができます。

イニシャルステップは1つのブロックに1個だけ作成することができます。



(3) ステップ

制御の1単位です。固有のステップ番号と、1対1に対応する実行プログラムを持ちます。



(4) 遷移

あるステップから次のステップへ、アクティブ状態を遷移させるための条件を表わします。遷移は、1対1に対応する遷移条件部を持ちます。



(5) エンド

SFC メインプログラムの終了を表わします。SFC メインプログラムには、このエンドかまたは(6)のエンドステップのいずれか1つが必要です。エンドは、1対1に対応する遷移条件部と戻り先ラベル番号を持ちます。直前のステップがアクティブな状態で遷移条件が成立すると、直前のステップをインアクティブにして、指定ラベルに続くステップをアクティブにします。(後に述べるジャンプと同様の動作です)



(6) エンドステップ

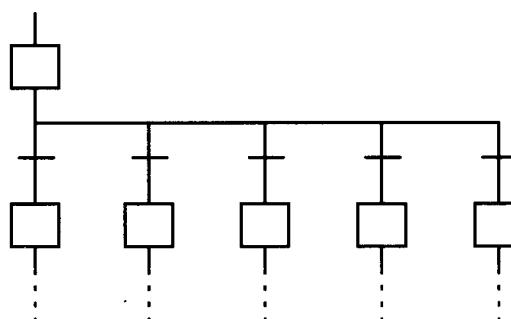
SFC メインプログラムの終了を表わします。SFC メインプログラムにはエンドステップかまたは(5)のエンドのいずれか1つが必要です。

エンドステップは、イニシャルステップと同じステップ番号を持ち、直前の遷移条件が成立するとイニシャルステップにアクティブ状態を戻します。



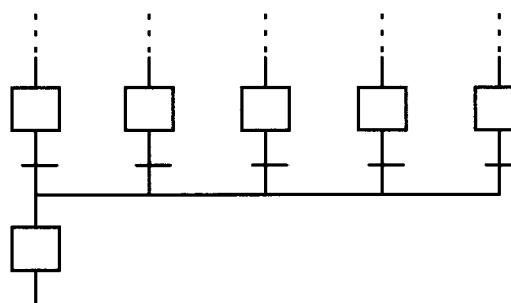
(7) 条件分岐(分流)

接続されている複数のステップのうち、遷移条件が成立している1つのステップにアクティブ状態を移します。遷移条件が同時に成立した場合には左側のステップが優先です。(分岐数は最大5列)



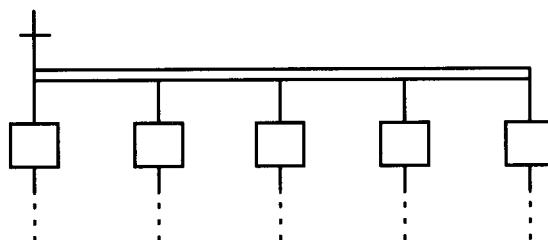
(8) 条件分岐(合流)

条件分岐で分かれた経路を1つのステップにまとめます。



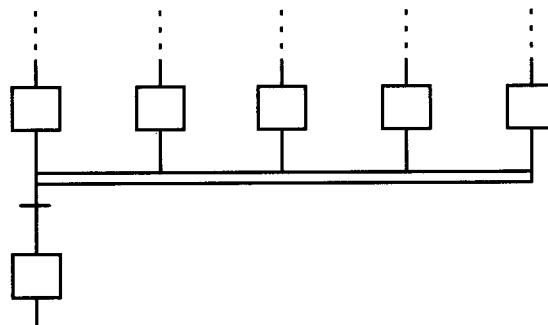
(9) 並列分岐(分流)

直前の遷移条件が成立したら、接続されている全てのステップをアクティブにします。(分岐数は最大5列)



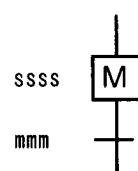
(10) 並列分岐(合流)

直前のステップが全てアクティブで、かつ遷移条件が成立したときに、次の1つのステップにアクティブ状態を移します。



(11) マクロステップ

マクロステップは1つのマクロプログラムに対応します。直前の遷移条件が成立すると、指定したマクロ番号のマクロプログラムにアクティブ状態を移します。マクロプログラム内で遷移が進み、マクロエンドまで達すると、マクロステップに続くステップにアクティブ状態を移します。マクロステップは遷移条件部を持たない(常にON)のダミー遷移を伴います。



(12) マクロエントリ

マクロプログラムの開始を表わします。マクロエントリは実行プログラムを持ちません。マクロエントリの下にはステップがつながります。マクロエントリは1つのブロックに1個だけ作成することができます。



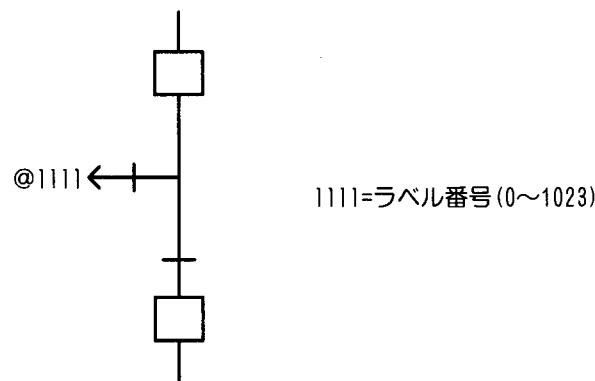
(13) マクロエンド

マクロプログラムの終了を表わします。マクロエンドは、1対1に対応する遷移条件部を持ち、この遷移条件が成立したとき対応するマクロステップに戻ります。



(14) ジャンプ

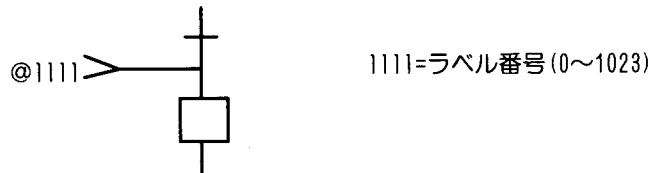
任意のステップへのジャンプを表わします。ジャンプは1対1に対応する遷移条件部とジャンプ先ラベル番号を持ち、遷移条件が成立したときに指定のラベルへジャンプします。ジャンプの遷移条件と下に続くステップへの遷移条件が同時に成立したときにはジャンプが優先します。



ジャンプはステップの直後に配置します。なお、同じラベル番号を持ったジャンプが複数箇所に存在してもかまいません。

(15) ラベル

エンドからの戻り先及びジャンプからの飛び先を表わします。
ラベルは遷移の直後に配置します。

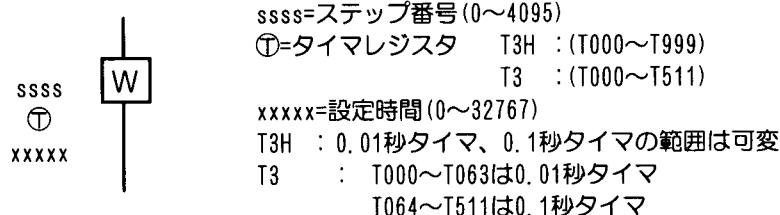


補足

ジャンプ及びエンドに対応するラベルが存在しないとき、及び同一のラベル番号を持ったラベルが複数箇所に存在するときは、RUN 起動時にエラーとなりますのでご注意下さい。

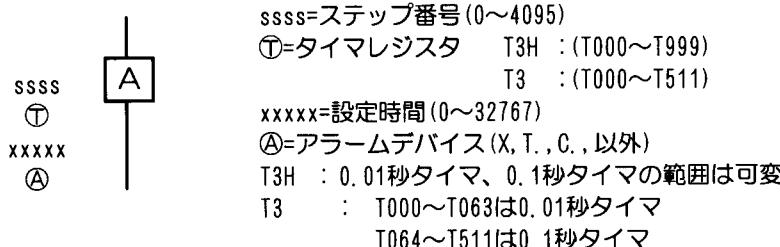
(16) 待ち時間付きステップ

アクティブになってからの時間を計測し、設定時間が経過するまでは、遷移条件が成立しても遷移を行わないステップです。1対1に対応する実行プログラムを持ちます。



(17) 渋滞検出付きステップ

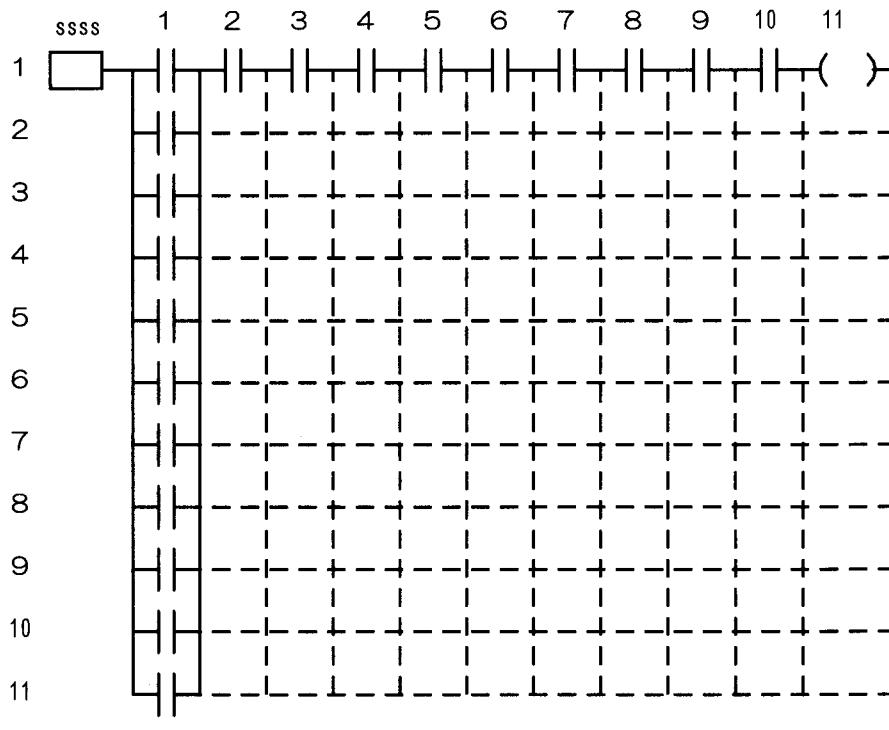
アクティブになってからの時間を計測し、設定時間内に遷移条件が成立しないときに、指定されたアラームデバイスを ON にするステップです。1対1に対応する実行プログラムを持ちます。なお、遷移条件が成立し、渋滞検出付きステップがインアクティブになると、アラームデバイスも OFF となります。



詳細部 1つのステップに対応する実行プログラム部を実行詳細部と称し、1つの遷移に対応する遷移条件部を遷移詳細部と称します。
これらの詳細部はラダー図で作成します。

(1) 実行詳細部

1つの実行詳細部の大きさは下記の通り 11 行×11 列、命令語ステップ数最大 121 ステップです。



実行詳細部として上記以上の大きさが必要な場合にはサブルーチンを利用します。(CALL 命令)

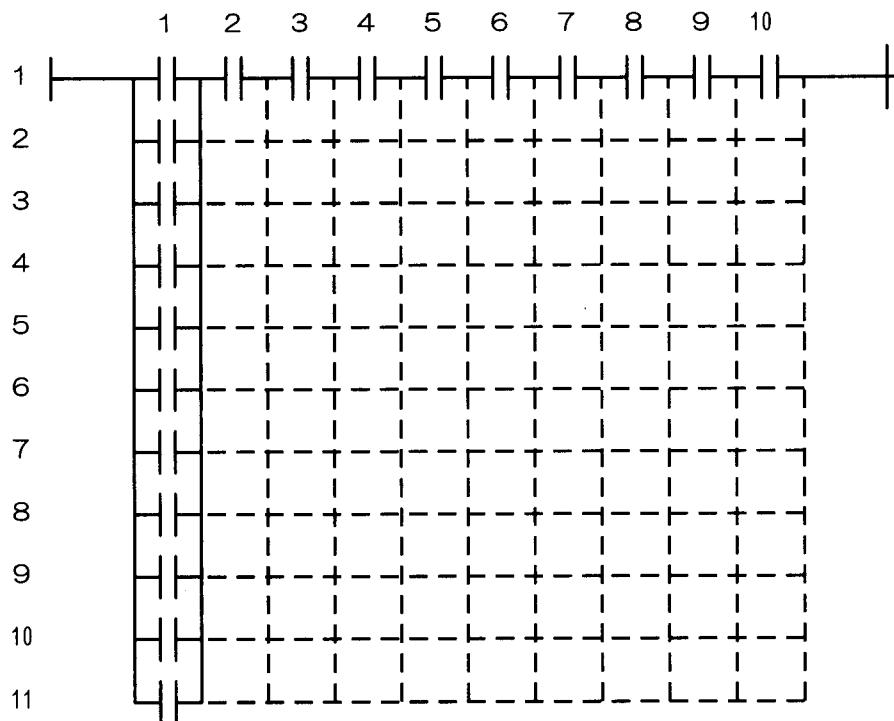
なお、ステップに対応する実行詳細部が無くても SFC の動作には影響はありません。この場合、そのステップはダミーステップ(次の遷移条件成立を待つだけのステップ)となります。

プログラミングにおいては、SFC 画面でステップを指定して詳細部表示モードを選択することにより、そのステップに対応する実行詳細部のモニタ/編集画面となります。

なお、実行詳細部の内容が、SET 命令、RST 命令、コイル、インバートコイル、ポジティブパルスコイル、ネガティブパルスコイルの中の1命令のみの場合には、詳細部表示画面とすることなく、直接編集が行えます。操作については別冊のプログラマの説明書をご覧下さい。

(2) 遷移詳細部

1つの遷移詳細部の大きさは、11行×10列、命令語ステップ数最大110ステップです。



ある遷移に対応する遷移詳細部がない場合には、その遷移条件は常に成立と見なされます。(ダミー遷移)

プログラミングにおいては、SFC画面で遷移を指定して詳細部表示モードを選択することにより、その遷移に対応する遷移詳細部のモニタ/編集画面となります。

なお、遷移詳細部の内容が、a接点またはb接点の1命令のみの場合には、詳細部表示画面とすることなく直接編集が行えます。操作については別冊のプログラマの説明書をご覧下さい。

補足

実行詳細部及び遷移詳細部には、下記の実行制御命令は使用できません。

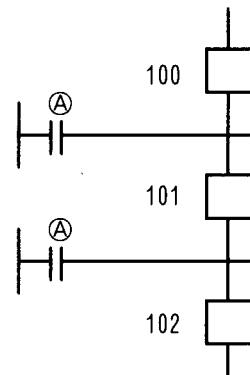
- ジャンプ(JSC/JCR, JUMP/LBL)
- マスター制御(MCS/MCR, MCSn/MCRn)
- エンド(END)
- FOR～NEXT(FOR/NEXT)

また遷移詳細部には、インバート接点及び各種コイル命令は使用できません。

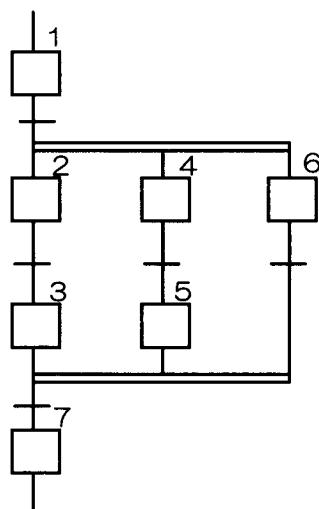
実行方式 1つのSFCプログラムにおける実行方式の考え方を以下に示します。

- (1) 1スキャン内では、遷移条件の評価、ステップ遷移処理、実行プログラム部の実行の順に動作します。
- (2) 遷移条件の評価とは、アクティブなステップにつながる遷移条件部を実行し、遷移条件成立のチェックを行うことです。このとき、アクティブなステップについてのみ評価するため、連続的につながるステップにおいて1スキャンで複数ステップ遷移することはできません。

例えば右図のように、ステップ100から101の遷移条件とステップ101から102の遷移条件が同一であるプログラムにおいて、前回スキャン時にステップ100がアクティブとなっていて、今回のスキャンでデバイスⒶがONしていたとき、今回のスキャンではステップ101に遷移します。(ステップ102への遷移は次回スキャン以降となります)



- (3) ステップ遷移処理とは、遷移条件の評価結果に基づいて、遷移条件成立なら前ステップをインアクティブにして次ステップをアクティブにすることです。
- (4) 実行プログラム部の実行は、アクティブなステップに対応する実行プログラム部は母線ONとして実行し、インアクティブなステップに対応する実行プログラム部は母線OFFとして実行します。このときの実行順序は下図のように、上から下へ、分岐では左から右への順に実行します。



図中の数字は実行プログラム部の実行順序を示しています

注意事項 SFC プログラムを作る上での注意事項について以下に列記します。

- (1) SFC プログラムの容量制限を下表にまとめます。この容量を越えないよう注意して下さい。

- ・ 全体容量(PC に作成することのできる最大数)

SFCemainプログラム本数	64本(0~63)
マクロプログラム本数	128本(0~127)
SFCステップ個数	4096個(0~4095)
SFCラベル数	1024個(0~1023)

- ・ SFC メインプログラム/マクロプログラム 1 本当りの容量

SFCステップ個数	128個
命令語ステップ数(詳細部合計)	765ステップ*
命令語ステップ数(SFC, 詳細部合計)	1024ステップ*
同時分岐数	5
SFC編集画面容量	128行×5列

- ・ 詳細部容量(ステップ/遷移 1 個当たり)

実行詳細部容量	121ステップ*
遷移詳細部容量	110ステップ*

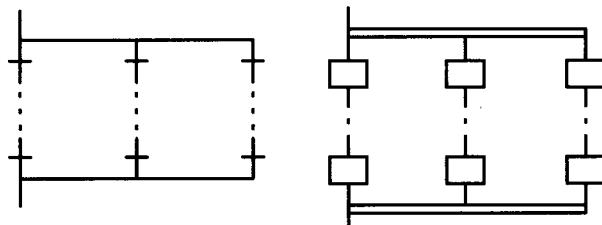
*) SFC 命令、ラダー図命令の所要ステップ数については、5.5 命令語一覧を参照して下さい。

- (2) SFC プログラムの起動及びリセットは、SFC イニシャライズ命令(SFC 命令/ラダー図命令)によって行われます。SFC イニシャライズは、指定範囲のステップをインアクティブにし、イニシャルルステップをアクティブにするという動作を行います。従って、SFC イニシャライズで指定するステップの範囲(初期化ステップ個数)は、その SFC プログラム(マクロプログラムも含む)で使用しているステップ番号を全て含み、他の SFC プログラムで使用しているステップ番号にはかかるないように考慮して下さい。

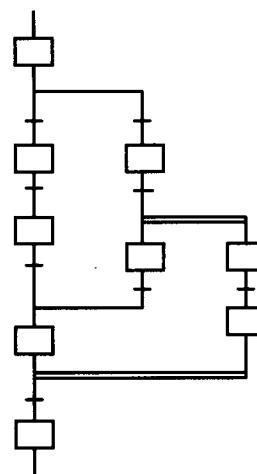
例えば、SFC イニシャライズの指定がステップ番号 0 から 50 個で、その SFC プログラムの中で 50 というステップを使用していたとすると、ステップ 50 がアクティブの状態で SFC イニシャライズを実行したときには、ステップ 50 はアクティブのままとなってしまいます。

一方、SFC イニシャライズの指定がステップ番号 100 から 201 個で、他の SFC プログラムで 300 というステップを使用していたとすると、ステップ 300 がアクティブの状態で SFC イニシャライズを実行したときには、ステップ 300 は無条件でインアクティブとなってしまいます。

- (3) 1つの SFC プログラム(マクロプログラム含む)の中で使用するステップ番号の順序には制約はありません。ただし、イニシャルステップはその中で最も若いステップ番号とする必要があります。(前述(2)参照)
 - (4) 条件分岐は遷移の上で分流し、遷移の下で合流します。また並列分岐はステップの上で分流し、ステップの下で合流します。



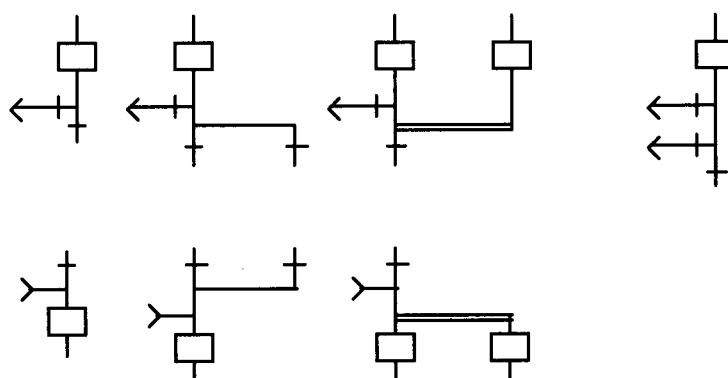
ただし、条件分岐、並列分岐共、対応する合流で終結する必要があります。従って次のようなプログラムは作成できません。



- (5) シャンプの飛び先は、上方向でも下方向でも、あるいは同一プログラム種別内の他の SFC プログラムでもかまいません。また、分岐の中から外へジャンプすることもできます。

このように、ジャンプはきわめて自由な使い方でできますので、ジャンプすることによって SFC の論理が異常とならないように(一連の SFC の中で関連のない複数のステップがアクティブになったりしないように)十分注意して下さい。

なお、ジャンプは常にステップの直後に配置します。また、ラベルは遷移の直後に配置するのが基本ですが、条件分岐(合流)の場合は合流線とステップの間に配置します。



- (6) SFC ステップの状態(アクティブ/インアクティブ)は停電保持されません。起動時には全てのステップはインアクティブとなります。(瞬停継続機能使用時の瞬停再始動時を除く)

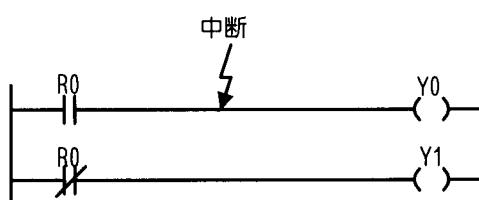
- (7) SFC プログラムブロックをラダー図のマスターントロール(MCS/MCR)で挟むことによって、SFC ステップの出力をコントロールすることができます。MCS の入力が OFF のとき、SFC のステップは遷移しますが、アクティブなステップの実行部も母線 OFF となります。

5.4 プログラム作成上 の補足

PC はマルチタスク機能をサポートしています。マルチクスク機能を使用した場合には、メインプログラムは割り込みプログラムによって、サブプログラムはメインプログラムと割り込みプログラムによって、中断される可能性があります。これに起因する留意点を以下に述べますので、プログラム作成時に考慮して下さい。

- (1) メインプログラム、サブプログラム、及び割り込みプログラムで同一のサブルーチンを使用することは避けて下さい。サブルーチン実行中に中断され、この状態で同じサブルーチンが実行された場合、再開後の結果が期待通りとならない場合があります。
- (2) ユーザデータ(レジスタ/デバイス)にはプログラム種別による区分はありません。従って、各々のデータについて、プログラム種別間での混用がないように十分注意して下さい。

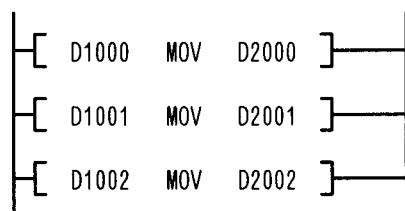
例)



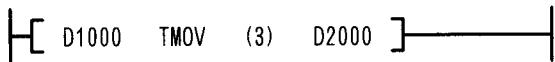
上図のタイミングで中断が発生し、中断中に R0 の内容が変更された場合には、本来起こり得ない Y0 と Y1 の同時 ON(または同時 OFF)が発生します。

- (3) 異なるプログラム種別間でのデータの受け渡しは、データ転送命令(MOV)やテーブル転送命令(TMOV)など1つの命令で行うようにして下さい。そうしないと上記(2)と同じことが起こり得ます。

例) 割り込みプログラムからメインプログラムへ D1000, D1001, D1002 の3つのデータを渡す場合のメインプログラムの構成



前記のプログラムでは、命令間で割り込みが発生した場合、D2000, D2001, D2002 の同期が保証できなくなります。この場合以下のよう にテーブル転送命令を使用して 1 命令にします。

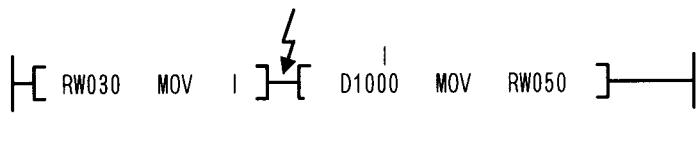


1 つの命令にすることが難しい場合は、メインプログラムと割り込みプログラム間であれば、割り込み禁止命令(DI)/割り込み許可命令(EI)を使用することも可能です。(DI 命令と EI 命令は必ずペアで使用して下さい)

- (4) 異なるプログラム種別間で同じインデックスレジスタを使用する場合には特に注意して下さい。場合によっては下の例のようにインデックスレジスタの退避が必要となります。

例)

ここで中断され I の内容が変わってしまう可能性がある

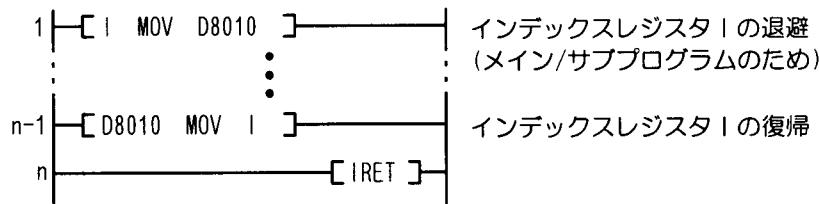


(メインプログラム)



5

(割り込みプログラム)



5.5 命令語一覧

次ページ以降にラダー図命令、SFC 命令の順で命令語リストを示します。

リスト中のグループは、プログラム開発システム(T-PDS)で採用しているファンクション命令のグループ分けに対応します。(SFC を除く)

所要ステップ数は、その命令を記憶するために必要とするメモリの大きさを意味します。所要ステップ数を4~7のように範囲で示してあるのは、同じ命令でも次の条件によってステップ数が変わることによります。

- 桁指定を使用した場合にはオペランド1個当たり1ステップ増加します。
- 倍長オペランドに定数を使用した場合には1ステップ増加します。
- 定数にインデックス修飾を行った場合には1ステップ増加します。

所要実行時間は、その命令をメモリから読み出し、解読し、必要な処理を行い、処理の結果を所定のメモリに書き込むという1サイクルに要する時間を表わします。

所要実行時間の最小値は、インデックス修飾無し、桁指定無しでユーザデータメモリをアクセス対象とした条件での値であり、最大値は、インデックス修飾付き、桁指定付き、直接入力レジスタ/デバイス(IW/I)及び直接出力レジスタ/デバイス(OW/O)を対象とするなど、その命令にとっての最悪条件での実行時間です。また、不実行時とは、その命令の入力がOFFであるなどの条件で、処理を行わないとき(読み出しのみ)の所要時間です。(不実行時の値は所要ステップ数が最小のときの値)

補足

ここでは各命令の概要について記述します。詳細については別冊の命令語説明書をご覧下さい。

ラダーラン命令(基本命令)

グル ープ	FUN No.	名 称	表 現	概 要		所要 スルップ数	所要実行時間 (μs)
				T3	T3H		
基 本 命 令	a接点		① — —	デバイス④のa接点(常時閉接点)		1	0.15 0.09
	b接点		② — —	デバイス④のb接点(常時閉接点)		1	0.15 0.09
	ON時微分接点		— —	前回スキャン時の入力が OFF で今回の入力が ON のときのみ出力を ON する		1	0.6 0.36
	OFF時微分接点		— —	前回スキャン時の入力が ON で今回の入力が OFF のときのみ出力を ON する		1	0.6 0.36
	コイル		③ — —	入力が ON のときデバイス④を ON する		1	0.3 0.18
	フォーストコイル		④ *(— —)	入力の ON/OFF にかかわらず、フォース特のデバイス④の状態を保持する		1	0.15 0.09
	インバータ		— —	入力の状態を反転して出力する		1	0.15 0.09
	インバートコイル		⑤ — —	入力状態を反転してデバイス④に格納する		1	0.3 0.18
	ポジティブ/パルス接点		⑥ — P—	入力が ON で、デバイス④が OFF→ON に変化したとき1スキャンだけ出力を ON する		1	0.6 0.36
	ネガティブ/パルス接点		⑦ — N—	入力が ON で、デバイス④が ON→OFF に変化したとき1スキャンだけ出力を ON する		1	0.6 0.36
ポジティブ/パルスコイル	ポジティブ/パルスコイル		⑧ — P)—	入力が OFF→ON に変化したときデバイス④を 1 スキャンだけ ON する		1	0.6 0.36
	ネガティブ/パルスコイル		⑨ — N)—	入力が ON→OFF に変化したとき、デバイス④を 1 スキャンだけ ON する		1	0.6 0.36
	シンプコントロールセット		— JCS —	JCS の入力が ON のとき、JCS-JCR 間の命令を高速読み飛ばしを行う		1	0.15 0.09
	シンプコントロールリセット		— JCR —			1	0.15 0.09
終了			H END H	メインプログラム、サブプログラムの終了を示す		1	— —

ラダー図命令(基本命令)

グループ	FUN No.	名 称	表 現	概 要	所要実行時間(μs)		
					ステップ数	T3	T3H
基本命令	オンディレイタイマ	-[T @ T0N ④]-	入力が ON してから④で指定する設定時間経過後出力を ON する。④はタイマレジスタ	2	0.3	0.18	
	オフディレイタイマ	-[T @ T0F ④]-	入力が OFF してから④で指定する設定時間経過後出力を OFF する。④はタイマレジスタ	2	0.3	0.18	
	シンクルショット	-[T @ SS ④]-	入力が ON してから④で指定する設定時間だけ出力を ON する。④はタイマレジスタ	2	0.3	0.18	
	カウンタ	-[CNT ④ E ④]-	イネーブル入力(E)がONのとき、カウント入力(C)が ON する回数をカウントする。カウント値が④で指定する設定値と等しくなったら出力(Q)を ON する。④はカウンタレジスタ	2	0.3	0.18	
	マスターコントロールセット	-[MCS]-	MCS の入力が ON のとき、MCS-MCR 間の母線を ON する	1	0.15	0.09	
	マスターコントロールリセット	-[MCR]-	MCS の入力が ON のとき、対応する MCR までの母線を ON する。n はネスティング No. (1 ~ 7)	2	12.25	4.9	
	134 マスターコントロールセット	-[MCS n]-	入力が OFF から ON に変化したとき、④で指定するタイマレジスタをクリアしてタイマを起動する	2	6.74	2.89	
	135 マスターコントロールリセット	-[MCR n]-					
148 タイマトリガ	-[TRG ④]-						

ラダー図命令(ファンクション命令)

グループ	FUN No.	名 称	表 現	概 要	所要ステップ数	所要実行時間(μs)		
						T3	T3H	備考
転送命令	18	データ転送	-[A MOV B]-	Ⓐの内容をⒷに転送する	3～5	0.9	0.54	*1
	19	倍長データ転送	-[A+1 · A DMOV B+1 · B]-	Ⓐ+1 · Ⓐの内容をⒷ+1 · Ⓑに転送する	3～6	10.35	4.14	
	20	データ否定転送	-[A NOT B]-	Ⓐの内容のビット反転データをⒷに転送する	3～5	9.0	3.6	
	21	倍長データ否定転送	-[A+1 · A DNOT B+1 · B]-	Ⓐ+1 · Ⓐの内容のビット反転データをⒷ+1 · Ⓑに転送する	3～6	10.8	4.32	
	22	データ交換	-[A XCHG B]-	ⒶとⒷの内容を交換する	3～5	15.3	6.12	
	23	倍長データ交換	-[A+1 · A DXCH B+1 · B]-	Ⓐ+1 · ⒶとⒷ+1 · Ⓑの内容を交換する	3～5	18.9	7.56	
	24	テーブル初期化	-[A TINZ (n) B]-	Ⓐを先頭とするサイズnのテーブルの内容をⒷに以降に初期化する	4～6	38.75 +0.93n	15.5 +0.37n	
	25	テーブル転送	-[A TMOV (n) B]-	Ⓐを先頭とするサイズnのテーブルの内容をⒷ以降に転送する	4～6	60.8 +1.23n	24.32 +0.49n	
	26	テーブル否定転送	-[A TNOT (n) B]-	Ⓐを先頭とするサイズnのテーブルの内容の比特反転データをⒷ以降に転送する	4～6	61.1 +1.45n	24.44 +0.58n	
四則演算	27	加算	-[A + B → C]-	Ⓐの内容とⒷの内容を加算し、結果をⒸに格納する	4～7	1.5	0.9	*2
	28	減算	-[A - B → C]-	Ⓐの内容からⒷの内容を減算し、結果をⒸに格納する	4～7	1.5	0.9	*3
		乗算	-[A * B → C+1 · C]-	Ⓐの内容とⒷの内容を乗算し、結果をⒸ+1 · Ⓑに格納する	4～7	4.35	1.08	*4
	30	除算	-[A / B → C]-	Ⓐの内容をⒷの内容で除算し、商をⒸに、余りをⒸ+1に格納する	4～7	7.65	4.59	*5
	31	倍長加算	-[A+1 · A D+ B+1 · B → C+1 · C]-	Ⓐ+1 · Ⓐの内容とⒷ+1 · Ⓑの内容を加算し、結果をⒸ+1 · Ⓑに格納する	4～9	15.25	6.1	
	32	倍長減算	-[A+1 · A D- B+1 · B → C+1 · C]-	Ⓐ+1 · Ⓐの内容からⒷ+1 · Ⓑの内容を減算し、結果をⒸ+1 · Ⓑに格納する	4～9	15.25	6.1	
	33	倍長乗算	-[A+1 · A D* B+1 · B → C+1 · C]-	Ⓐ+1 · Ⓐの内容とⒷ+1 · Ⓑの内容を乗算し、結果をⒸ+3 · Ⓑに格納する	4～9	15.55	6.22	

ラダー図命令(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要実行時間(μs)			
					ステップ数	T3	T3H	備考
四 則 演 算	34	倍長除算	-[A+1 · A) D / B+1 · B → C+1 · C]-	Ⓐ+1 · Ⓑの内容をⒷ+1 · Ⓑの内容で除算し、商をⒸ+1 · Ⓒに格納する	4～9	24.63	9.85	
	35	キャリー付き加算	-[A +C B → C]-	Ⓐの内容とⒷの内容とキャリーフラグの内容を加算し、結果をⒸに格納する。演算結果によりキャリーフラグは変化	4～7	15.73	6.29	
	36	キャリー付き減算	-[A -C B → C]-	Ⓐの内容からⒷの内容とキャリーフラグの内容を減算し、結果をⒸに格納する。演算結果によりキャリーフラグは変化	4～7	15.73	6.29	
	37	キャリー付き倍長加算	-[A+1 · A) D+C B+1 · B → C+1 · C]-	Ⓐ+1 · Ⓑの内容とⒷ+1 · Ⓑの内容とキャリーフラグの内容を加算し、結果をⒸ+1 · Ⓒに格納する。演算結果によりキャリーフラグは変化	4～9	18.03	7.21	
	38	キャリー付き倍長減算	-[A+1 · A) D-C B+1 · B → C+1 · C]-	Ⓐ+1 · Ⓑの内容からⒷ+1 · Ⓑの内容とキャリーフラグの内容を減算し、結果をⒸ+1 · Ⓒに格納する。演算結果によりキャリーフラグは変化	4～9	18.03	7.21	
	39	符号無し乗算	-[A) U* B → C+1 · C]-	Ⓐの内容とⒷの内容を乗算し、結果をⒸ+1 · Ⓒに格納する(正の整数演算)	4～7	18.43	7.37	
	40	符号無し除算	-[A) U/ B → C]-	Ⓐの内容をⒷの内容で除算し、商をⒸに、余りをⒸ+1に格納する(正の整数演算)	4～7	19.43	7.77	
	41	符号無し倍長単長除算	-[A+1 · A) DIV B → C]-	Ⓐ+1 · Ⓑの内容をⒷの内容で除算し、商をⒸに、余りをⒸ+1に格納する(正の整数演算)	4～7	21.68	8.67	
	42	倍長乗除算	-[A+1 · A) D*/B+1 · B → C+1 · C]-	Ⓐ+1 · Ⓑの内容とⒷ+1 · Ⓑの内容を乗算し、それをⒸ+3 · Ⓒ+2 の内容で除算し、商をⒸ+1 · Ⓒに余りをⒸ+3 · Ⓒ+2 に格納する	4～8		61.07	
	43	インクリメント	-[+1 A]-	Ⓐの内容を1だけ増加させる	2～3	8.08	3.23	
	44	倍長インクリメント	-[D+1 A+1 · A]-	Ⓐ+1 · Ⓑの内容を1だけ増加させる	2～3	10.28	4.11	
	45	デクリメント	-[-1 A]-	Ⓐの内容を1だけ減少させる	2～3	8.08	3.23	
	46	倍長デクリメント	-[D-1 A+1 · A]-	Ⓐ+1 · Ⓑの内容を1だけ減少させる	2～3	10.28	4.11	

ラダー図命令(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間 (μs)		
						T3	T3H	備考
四 則 演 算	208	浮動小数点加算	-[Ⓛ+1 · Ⓛ F+ Ⓛ+1 · Ⓛ → Ⓛ+1 · Ⓛ]-	Ⓐ+1 · ⓁとⒷ+1 · Ⓛの浮動小数点データを加算し、結果をⒸ+1 · Ⓛに格納する	4	25.87	14.44	
	209	浮動小数点減算	-[Ⓛ+1 · Ⓛ F- Ⓛ+1 · Ⓛ → Ⓛ+1 · Ⓛ]-	Ⓐ+1 · Ⓛの浮動小数点データからⒷ+1 · Ⓛの浮動小数点データを減算し、結果をⒸ+1 · Ⓛに格納する	4	27.13	14.82	
	210	浮動小数点乗算	-[Ⓛ+1 · Ⓛ F* Ⓛ+1 · Ⓛ → Ⓛ+1 · Ⓛ]-	Ⓐ+1 · ⓁとⒷ+1 · Ⓛの浮動小数点データを乗算し、結果をⒸ+1 · Ⓛに格納する	4	22.59	12.08	
論 理 演 算	211	浮動小数点除算	-[Ⓛ+1 · Ⓛ F / Ⓛ+1 · Ⓛ → Ⓛ+1 · Ⓛ]-	Ⓐ+1 · Ⓛの浮動小数点データをⒷ+1 · Ⓛの浮動小数点データで除算し、結果をⒸ+1 · Ⓛに格納する	4	28.28	12.06	
	48	論理積	-[Ⓛ AND Ⓛ → Ⓛ]-	ⒶとⒷの論理積を求めⒸに格納する	4~7	12.1	4.84	
	49	倍長論理積	-[Ⓛ+1 · Ⓛ DAND Ⓛ+1 · Ⓛ → Ⓛ+1 · Ⓛ]-	Ⓐ+1 · ⓁとⒷ+1 · Ⓛの論理積を求めⒸ+1 · Ⓛに格納する	4~9	14.8	5.92	
論 理 演 算	50	論理和	-[Ⓛ OR Ⓛ → Ⓛ]-	ⒶとⒷの論理和を求めⒸに格納する	4~7	12.1	4.84	
	51	倍長論理和	-[Ⓛ+1 · Ⓛ DOR Ⓛ+1 · Ⓛ → Ⓛ+1 · Ⓛ]-	Ⓐ+1 · ⓁとⒷ+1 · Ⓛの論理和を求めⒸ+1 · Ⓛに格納する	4~9	14.8	5.92	
	52	排他的論理和	-[Ⓛ EOR Ⓛ → Ⓛ]-	ⒶとⒷの排他的論理和を求めⒸに格納する	4~7	12.1	4.84	
論 理 演 算	53	倍長排他的論理和	-[Ⓛ+1 · Ⓛ DEOR Ⓛ+1 · Ⓛ → Ⓛ+1 · Ⓛ]-	Ⓐ+1 · ⓁとⒷ+1 · Ⓛの排他的論理和を求めⒸ+1 · Ⓛに格納する	4~9	14.8	5.92	
	54	否定排他的論理和	-[Ⓛ ENR Ⓛ → Ⓛ]-	ⒶとⒷの否定排他的論理和を求めⒸに格納する	4~7	12.1	4.84	
	55	倍長否定排他的論理和	-[Ⓛ+1 · Ⓛ DENR Ⓛ+1 · Ⓛ → Ⓛ+1 · Ⓛ]-	Ⓐ+1 · ⓁとⒷ+1 · Ⓛの否定排他的論理和を求めⒸ+1 · Ⓛに格納する	4~9	14.8	5.92	
テ ー ブ ル	57	テーブル論理積	-[Ⓛ TAND (n) Ⓛ → Ⓛ]-	Ⓐを先頭とするサイズnのテーブルとⒷを先頭とするサイズnのテーブルの論理積を求め、Ⓒに格納する	5	52.28 +1.8n	23.31 +0.72n	
	58	テーブル論理和	-[Ⓛ TOR (n) Ⓛ → Ⓛ]-	Ⓐを先頭とするサイズnのテーブルとⒷを先頭とするサイズnのテーブルの論理和を求め、Ⓒに格納する	5	52.28 +1.8n	23.31 +0.72n	

ラダー図命令(ファンクション命令)

グループ	FUN No.	名 称	表 現	概 要	所要ステップ数	T3	T3H	所要実行時間(μs)	備考
論理演算	59	テーブル排他的論理和	-[④ TEOR (n) ⑤ → ⑥]-	④を先頭とするサイズnのテーブルと⑤を先頭とするサイズnのテーブルの排他的論理和を求める、⑥以降に格納する	5	52.28 +1.8n	23.31 +0.72n		
	60	テーブル否定排他的論理和	-[④ TENR (n) ⑤ → ⑥]-	④を先頭とするサイズnのテーブルと⑤を先頭とするサイズnのテーブルの否定排他的論理和を求める、⑥以降に格納する	5	52.28 +1.8n	23.31 +0.72n		
	64	テスト	-[④ TEST ⑤]-	④と⑤の論理積の結果が0以外のとき出力をONにする	3~5	9.4	3.76		
	65	倍長テスト	-[④+1・④ 0TST ③+1・④]-	④+1と④と③+1・④の論理積の結果が0以外のとき出力をONにする	3~7	11.7	4.68		
	66	ビットファイブルビットテスト	-[④ TTST (n) ⑤]-	④を先頭とするnワードのテーブル内の④番目のビットのON/OFF状態を判定する	4~5	22.45	8.98		
シフト	68	1ビット右シフト	-[④ SHR 1 ⑤]-	④のデータを1ビット右(MSB方向)にシフトし、結果を④に格納する。結果によりキャリーフラグは変化	2~3	10.3	4.12		
	69	1ビット左シフト	-[④ SHL 1 ⑤]-	④のデータを1ビット左(MSB方向)にシフトし、結果を④に格納する。結果によりキャリーフラグは変化	2~3	11.7	4.68		
	70	nビット右シフト	-[④ SHR n → ⑤]-	④のデータをnビット右(MSB方向)にシフトし、結果を④に格納する。結果によりキャリーフラグは変化	4~6	11.93 +0.68n	4.77 +0.27n		
	71	nビット左シフト	-[④ SHL n → ⑤]-	④のデータをnビット左(MSB方向)にシフトし、結果を④に格納する。結果によりキャリーフラグは変化	4~6	13.33 +0.68n	5.33 +0.27n		

ラダー図命令(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間 (μs)		
						T3	T3H	備考
シ フ ト	72	mビットファイル nビット右シフト	-[TSHR (m) → ③]-	⑥がレジスタの場合: ⑥を先頭とするmワードのテーブルを④で示すワード数だけ右(若いアドレス方向)にシフトする ⑥がデバイスの場合: ⑥を先頭とするmビットのビットファイルを④で示すビット数だけ右(MSB 方向)にシフトする。 結果によりキャリーフラグは変化	4~5	36.48 -0.2n +1.13m	14.59 -0.08n +0.45m	
				⑥がレジスタの場合: ⑥を先頭とするmワードのテーブルを④で示すワード数だけ左(大きいアドレス方向)にシフトする ⑥がデバイスの場合: ⑥を先頭とするmビットのビットファイルを④で示すビット数だけ左(MSB 方向)にシフトする。 結果によりキャリーフラグは変化	4~5	53.25 -0.05n +0.15m	21.3 -0.02n +0.05m	
73		mビットファイル nビット左シフト	-[TSHL (m) → ③]-	イネーブル入力(E)が ON のとき、シフト入力(S)が ON するとデバイス⑥を先頭とするn個のデバイスの内容を1ビット左にシフトする。結果によりキャリーフラグは変化	4~5	37.4 -0.23n +1.13m	14.96 -0.09n +0.45m	
74	シフトレジスタ		[D SR q] [S (n)] [E] ④	イネーブル入力(E)が ON のとき、シフト入力(S)が ON するヒデバイス⑥を先頭とするn個の内容を1ビット左または右にシフトする(シフト方向は方向入力(L)の状態による)。結果によりキャリーフラグは変化	3	40.53 +0.28n	16.21 +0.11n	
75	双方向シフトレジスタ		[D DSR Q] [S (n)] [E] [L] ④	デバイス⑥の1つ前のデバイス(④-1)の内容を④に格納し、④-1は0にする	2	32.05	12.82	
76	デバイスシフト		-[SFT ④]-					

ラダー図命令(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)	
						T3	T3H
78	1ビット右ローテート	-[RTR 1 ④]-	④のデータを1ビット右(MSB方向)にローテートする。結果によりキャリーフラグは変化	2～3	10.78	4.31	
79	1ビット左ローテート	-[RTL 1 ④]-	④のデータを1ビット左(MSB方向)にローテートする。結果によりキャリーフラグは変化	2～3	10.38	4.15	
80	nビット右ローテート	-[④ RTR n → ④]-	④のデータをnビット右(MSB方向)にローテートする。結果によりキャリーフラグは変化	4～6	13.73 +0.25n	5.49 +0.1n	
81	nビット左ローテート	-[④ RTL n → ④]-	④のデータをnビット左(MSB方向)にローテートする。結果によりキャリーフラグは変化	4～6	12.78 +0.25n	5.11 +0.1n	

82	mビットファイル nビット右ローテート	-[④ TRTR (m) ④]-	④がレジスタの場合: ④を先頭とするmワードのテーブルを④で示すワード数だけ右(若い)アドレス方向)にローテートする ④がデバイスの場合: ④を先頭とするmビットのビットファイルを④で示すビット数だけ右(MSB方向)にローテートする。結果によりキャリーフラグは変化	4～5	40.58 +1.13n +1.13m	16.23 +0.45n +0.45m	

83	mビットファイル nビット左ローテート	-[④ TRTL (m) ④]-	④がレジスタの場合: ④を先頭とするmワードのテーブルを④で示すワード数だけ左(大きい)アドレス方向)にローテートする ④がデバイスの場合: ④を先頭とするmビットのビットファイルを④で示すビット数だけ左(MSB方向)にローテートする。結果によりキャリーフラグは変化	4～5	40.53 +1.15n +1.13m	16.21 +0.46n +0.45m	

84	キャリー付き 1ビット右ローテート	-[RRC 1 ④]-	④のデータをキャリーフラグを含めて1ビット右(MSB方向)にローテートする。結果によりキャリーフラグは変化	2～3	11.73	4.69	
85	キャリー付き 1ビット左ローテート	-[RLC 1 ④]-	④のデータをキャリーフラグを含めて1ビット左(MSB方向)にローテートする。結果によりキャリーフラグは変化	2～3	11.25	4.15	

ラダーツクショニ命合(ファンクション命合)

グル ープ	FUN No.	名 称	表 現	概 要	所要 スルップ数		所要実行時間 (μs)
					T3	T3H	
ロ ー ト テ ー ト	86	キャリー付き nビット右ロードート	-[Ⓛ RRC n → Ⓛ]-	Ⓐのデータをキャリーフラグを含めてnビット右(MSB方向)にロードートし、結果を⑥に格納する。結果によりキャリーフラグは変化	4~6	11.48 +2.03n	4.59 +0.81n
	87	キャリー付き nビット左ロードート	-[Ⓛ RLC n → Ⓛ]-	Ⓐのデータをキャリーフラグを含めてnビット左(MSB方向)にロードートし、結果を⑥に格納する。結果によりキャリーフラグは変化	4~6	13.6 +1.8n	5.44 +0.72n
	88	キャリー付き mビットファイル nビット右ロードート	-[Ⓛ TRRC (m) → Ⓛ]-	⑥がレジスタの場合: ⑥を先頭とするmワードのテーブルを④で示すワード数だけ右(大きいアドレス方向)にロードする。(FUN82のレジスタ指定と同じ) ⑥がデバイスの場合: ⑥を先頭とするmビットのビットファイルをキャリーフラグを含めて④で示すビット数だけ右(MSB方向)にロードートする。結果によりキャリーフラグは変化	4~5	40.6 +1.08n +1.13m	16.24 +0.43n +0.45m
	89	キャリー付き mビットファイル nビット左ロードート	-[Ⓛ TRLC (m) Ⓛ]-	⑥がレジスタの場合: ⑥を先頭とするmワードのテーブルを④で示すワード数だけ左(大きいアドレス方向)にロードする。(FUN83のレジスタ指定と同じ) ⑥がデバイスの場合: ⑥を先頭とするmビットのビットファイルをキャリーフラグを含めて④で示すビット数だけ左(MSB方向)にロードートする。結果によりキャリーフラグは変化	4~5	40.53 +1.15n +1.13m	16.21 +0.46n +0.45m
	90	マルチプレクサ	-[Ⓛ MPX (n) Ⓛ → Ⓛ]-	レジスタ⑥を先頭としたサイズnのテーブルの⑨番目のレジスタの内容を⑩に格納する	5~6	71.38 +0.18n +0.13m	28.55 +0.07n +0.05m
	91	デマルチプレクサ	-[Ⓛ DPX (n) Ⓛ → Ⓛ]-	レジスタ⑥の内容をレジスタ⑩番目のレジスタに格納する	5~6	24.35	9.74
	92	テーブル→ビット転送	-[Ⓛ TBM (n) Ⓛ → Ⓛ]-	レジスタ⑥を先頭とするサイズnのテーブル内の先頭から⑩番目のビットの内容をデバイス⑩に格納する	5~6	22.15	8.86
					31.1	12.44	

ラダー図命令(ファンクション命令)

グループ	FUN No.	名 称	表 現	概 要	所要ステップ数	所要実行時間(μs)	
						T3	T3H
ローテート	93	ビット→テーブル転送	-[Ⓐ BTW (n) Ⓛ → Ⓜ]-	デバイスⒶの内容をレジスタⓂを先頭とするサイズnのテーブル内の②番目のビットに格納する	5~6	28.85	11.54

*) 開発中

ラダーランクション命令

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)		
						T3	T3H	備考
比 較	95	ビットファイル比較	-[Ⓐ TCMR [n] Ⓢ → Ⓣ]-	Ⓐ及びⒷから始まる2組のnワードテーブルを比較し、不一致ビット位置をⒸに格納する。	5	45.08	18.03	
96	より大きい		-[Ⓐ > Ⓢ]-	Ⓐ>Ⓑのとき出力をONする。(整数比較)	3~5	9.4	3.76	
97	より大きいまたは等しい		-[Ⓐ >= Ⓢ]-	Ⓐ≥Ⓑのとき出力をONする。(整数比較)	3~5	9.4	3.76	
98	等しい		-[Ⓐ = Ⓢ]-	Ⓐ=Ⓑのとき出力をONする。(整数比較)	3~5	9.4	3.76	
99	等しくない		-[Ⓐ ≠ Ⓢ]-	Ⓐ≠Ⓑのとき出力をONする。(整数比較)	3~5	9.4	3.76	
100	より小さい		-[Ⓐ < Ⓢ]-	Ⓐ<Ⓑのとき出力をONする。(整数比較)	3~5	9.4	3.76	
101	より小さいまたは等しい		-[Ⓐ <= Ⓢ]-	Ⓐ≤Ⓑのとき出力をONする。(整数比較)	3~5	9.4	3.76	
102	倍長 より大きい		-[Ⓐ+1・Ⓐ D> Ⓢ+1・Ⓓ]-	Ⓐ+1・Ⓐ>Ⓓ+1・Ⓓのとき出力をONする。 (倍長整数比較)	3~7	12.1	4.84	
103	倍長 より大きいまたは等しい		-[Ⓐ+1・Ⓐ D= Ⓢ+1・Ⓓ]-	Ⓐ+1・Ⓐ≥Ⓓ+1・Ⓓのとき出力をONする。 (倍長整数比較)	3~7	11.2	4.48	
104	倍長 等しい		-[Ⓐ+1・Ⓐ D< Ⓢ+1・Ⓓ]-	Ⓐ+1・Ⓐ≠Ⓓ+1・Ⓓのとき出力をONする。 (倍長整数比較)	3~7	11.2	4.48	
105	倍長 等しくない		-[Ⓐ+1・Ⓐ D<> Ⓢ+1・Ⓓ]-	Ⓐ+1・Ⓐ≠Ⓓ+1・Ⓓのとき出力をONする。 (倍長整数比較)	3~7	11.2	4.48	
106	倍長 より小さい		-[Ⓐ+1・Ⓐ D< Ⓢ+1・Ⓓ]-	Ⓐ+1・Ⓐ<Ⓓ+1・Ⓓのとき出力をONする。 (倍長整数比較)	3~7	12.1	4.84	
107	倍長 より小さいまたは等しい		-[Ⓐ+1・Ⓐ D<= Ⓢ+1・Ⓓ]-	Ⓐ+1・Ⓐ≤Ⓓ+1・Ⓓのとき出力をONする。 (倍長整数比較)	3~7	11.2	4.48	
108	符号無し より大きい		-[Ⓐ > Ⓣ]-	Ⓐ>Ⓓのとき出力をONする。(正整数比較)	3~5	9.4	3.76	
109	符号無し より大きいまたは等しい		-[Ⓐ >= Ⓣ]-	Ⓐ≥Ⓓのとき出力をONする。(正整数比較)	3~5	9.4	3.76	
110	符号無し 等しい		-[Ⓐ = Ⓣ]-	Ⓐ=Ⓓのとき出力をONする。(正整数比較)	3~5	9.4	3.76	

ラダー図命令(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)	
						T3	T3H
比 較	111	符号無し等しくない	-[A]< [B]-	Ⓐ≠Ⓑのとき出力をONする。(正整数比較)	3～5	9.4	3.76
	112	符号無しより小さい	-[A]< [B]-	Ⓐ<Ⓑのとき出力をONする。(正整数比較)	3～5	9.4	3.76
	113	符号無しより大きいまたは等しい	-[A]<= [B]-	Ⓐ≤Ⓑのとき出力をONする。(正整数比較)	3～5	9.4	3.76
	212	浮動小数点より大きい	-[A+1 · Ⓢ F]> [B+1 · Ⓢ F]-	Ⓐ+1 · ⓈとⒷ+1 · Ⓢの実数データを比較し、 Ⓐ+1 · Ⓢ>Ⓑ+1 · Ⓢのとき出力をONする。 (実数比較)	3	21.41	7.2
	213	浮動小数点より大きいまたは等しい	-[A+1 · Ⓢ F]= [B+1 · Ⓢ F]-	Ⓐ+1 · ⓈとⒷ+1 · Ⓢの実数データを比較し、 Ⓐ+1 · Ⓢ≥Ⓑ+1 · Ⓢのとき出力をONする。 (実数比較)	3	21.41	7.2
	214	浮動小数点等しい	-[A+1 · Ⓢ F]= [B+1 · Ⓢ F]-	Ⓐ+1 · ⓈとⒷ+1 · Ⓢの実数データを比較し、 Ⓐ+1 · Ⓢ=Ⓑ+1 · Ⓢのとき出力をONする。 (実数比較)	3	20.48	6.31
	215	浮動小数点等しくない	-[A+1 · Ⓢ F]< [B+1 · Ⓢ F]-	Ⓐ+1 · ⓈとⒷ+1 · Ⓢの実数データを比較し、 Ⓐ+1 · Ⓢ≠Ⓑ+1 · Ⓢのとき出力をONする。 (実数比較)	3	20.48	6.31
	216	浮動小数点より小さい	-[A+1 · Ⓢ F]< [B+1 · Ⓢ F]-	Ⓐ+1 · ⓈとⒷ+1 · Ⓢの実数データを比較し、 Ⓐ+1 · Ⓢ<Ⓑ+1 · Ⓢのとき出力をONする。 (実数比較)	3	21.47	7.22
	217	浮動小数点より小さいまたは等しい	-[A+1 · Ⓢ F]≤ [B+1 · Ⓢ F]-	Ⓐ+1 · ⓈとⒷ+1 · Ⓢの実数データを比較し、 Ⓐ+1 · Ⓢ≤Ⓑ+1 · Ⓢのとき出力をONする。 (実数比較)	3	21.47	7.18

ラダー図命令 (ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 スループ数	所要実行時間 (μs)		
						T3	T3H	備考
テ タ 処 理	114	デバイス/レジスタセット	-[SET ④]-	④がデバイスの場合: デバイス④を 0N にセットする ④がレジスタの場合: レジスタ④に HFFF を格納する	2~3	9.0	3.6	
	115	デバイス/レジスタリセット	-[RST ④]-	④がデバイスの場合: デバイス④を 0FF にリセットする ④がレジスタの場合: レジスタ④に 0 を格納する	2~3	5.8	2.32	
	116	テーブルビットセット	-[④ TSET (n) ④]-	レジスタ④を先頭とする n ワードのビットファ イルにて、④で示される位置のビットを 0N にセ ットする	4~5	23.55	9.42	
	117	テーブルビットリセット	-[④ TRST (n) ④]-	レジスタ④を先頭とする n ワードのビットファ イルにて、④で示される位置のビットを 0FF にリ セットする	4~5	24.05	9.62	
	118	キャリーセット	-[SETC]-	キャリーフラグをセットする	1	3.15	1.26	
	119	キャリーリセット	-[RSTC]-	キャリーフラグをリセットする	1	3.15	1.26	
	120	エンコード	-[④ ENC (n) ④]-	④を先頭とするサイズ 2^n ビットのビットファ イルにて、最上位 n ビット位置をレジスタ④に 格納する	3~4	48.88 +7.28n	19.55 +2.91n	
	121	テコード	-[④ DNC (n) ④]-	④を先頭とするサイズ 2^n ビットのビットファ イルに対して、レジスタ④の下位 n ビットで示され るビット位置を 0N とし、他は全て 0FF とする	3~4	26.7 +6.2n	10.68 +2.48n	
	122	ビットカウント	-[④ BC ④]-	④ のデータ中、0N しているビットの数をカウント し④に格納する	3~5	26.4	10.56	
	123	倍長ビットカウント	-[④ DBC ④]-	④ + ④ の倍長データ中、0N しているビットの数 をカウントし④に格納する	3~6	45.4	18.16	

ラダーノット(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要実行時間(μs)			
					ステップ数	T3	T3H	備考
データ 処理	124	データサーチ	-[④ SCH (n) ③ → ②]-	⑥を先頭とするnワードのデータテーブル中、④の内容と一致するものと検索し、一致した個数を⑤に、一致したレジスタのうち最も若いレジスタアドレスを⑥に格納する	5~6	39.52 +2.22n	12.47 +0.9n	
	125	プッシュ	-[④ PUSH (n) ③ → ②]-	⑥を先頭とするnワードのデータテーブルに④のデータをプッシュし、③の値を1増加させる	5~6	24.98 +1.18n	9.99 +0.47n	
	126	ポップラスト	-[④ POPL (n) ③ → ②]-	⑥を先頭とするnワードのデータテーブルから最後にプッシュしたデータを取り出して③に格納すると共に④の値を1減少させる	5~6	27.25 +1.15n	10.9 +0.46n	
	127	ホップファースト	-[④ POPF (n) ③ → ②]-	⑥を先頭とするnワードのデータテーブルから最初にプッシュしたデータを取り出して③に格納すると共に④の値を1減少させる	5~6	28.65	11.46	
	147	フリップフロップ	-[SF/F Q R ④]-	セット入力(S)がONのときデバイス④をONにしてセッティ、リセット入力(R)がONのときデバイス④をOFFにリセットする。(リセット優先)	2	9.45	3.78	
	149	アップダウンカウンタ	-[U/D Q C E ④]-	イネーブル入力(E)がONのとき、カウント入力(C)がONする回数をカウントし、カウンタレジスタ④に格納する。カウンタ方向(加算/減算)はUP/DOWN選択入力(U)の状態によって選択可能(下記) ON アップカウント(加算) OFF ダウンカウント(減算)	2	5.65	2.26	

ラダー図命令(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間 (μs)		
						T3	T3H	備考
P R	128	サブルーチンコール	-[CALL N. nn]-	入力が ON のとき、サブルーチン No. nn のサブルーンをコールする	2~3	20.1	9.24	
G	129	サブルーチンリターン	-[RET]-	サブルーチンの終了を示す	1			
制 御	130	条件ジャンプ	-[JUMP N. nn]-	入力が ON のとき、ラベル No. nn のラベルへ直接ジャンプする	2~3	8.1	3.24	
	136	ジャンプラベル	-[LBL (nn)]-	条件ジャンプの飛び先を示す	2			
	132	繰り返しFOR	-[FOR n]-	FOR～NEXT 間を n で指定される回数繰り返し実行する	2	15.43 +6.78n	6.17 +2.71n	
	133	繰り返しNEXT	-[NEXT]-	サブルーチン (No. nn) の入口を示す	1			
	137	サブルーチンエントリ	-[SUBR (nn)]-	サブルーチン (No. nn) の実行を許可する	2	0.45	0.18	
	140	割り込みプログラム許可	-[EI]-	割り込みプログラムの実行を禁止する	1	133.2	53.28	
	141	割り込みプログラム禁止	-[DI]-	割り込みプログラムの終了を示す	1	132.2	52.88	
	142	割り込みプログラム終了	-[RET]-	スキヤンタイムオーバーフローを伸ばす	1	—	—	
	143	ウォッチドッグタイマリセット	-[WD n]-	デバイス④を先頭とする n 個のデバイスを OFF し、④を ON する。(ステップシーケンスの起動)	2	156.95	62.78	
	144	ステップシーケンス イニシャライズ	-[STZ (n) ④]-	入力が ON でデバイス④が ON の一連のステップシーケンスを構成する	3	29.25 +0.09n	5.0 +0.02n	
	145	ステップシーケンス入力	-[STIN ④]-	入力が ON のとき、同一回路上のステップシーケンス入力命令のデバイスを OFF にしデバイス④を ON にする	2	9.04	3.22	
	146	ステップシーケンス出力	-[STOT ④]-	入力が OFF のとき、同一回路上のステップシーケンス入力命令のデバイスを ON にする	2	12.66 +0.78n	5.67 +2.44n	
	241	SFC イニシャライズ	-[SFIZ (n) ④]-	入力が OFF から ON に変化したとき、SFC ステップ④から n 個分のステップをリセットし、ステップ④をアクティビティにする。(SFC の起動)	3	17.58 +0.12n	6.95 +0.05n	

ラダー図命令(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)		備考
						T3	T3H	
R A S	150	診断表示	-[DIAG ④ ⑤]-	入力が OFF から ON に変化したとき、④で示すワードを特殊レジスタに登録し、対応するアナンシエータリレーを ON する。⑤を先頭とするレジスタテーブルに登録されたエラーメッセージ(最大12文字)は周辺装置にて確認可能	3~4	39.14 +0.09n	10.98 +0.02n	
151	診断表示リセット		-[DIAR ④]-	診断表示命令(FUN150)で登録されたエラーワードリスト及びアンシエータリレーからのワードコードを削除する	2~3	17.71 +1.43n	6.41 +1.31n	
152	スタートスラッシュセット		-[STLS]-	あらかじめプログラムで設定した最大32個のデバイス/レジスタの内容をラッチエリアに格納する	1	898.04 +25.32n	320.48 +12.94n	
153	スタートスラッシュセット		-[STLR]-	スタートスラッシュ状態を解除する	1	117.95	47.18	
154	カレンダ設定		-[④ CLND]-	レジスタ④を先頭とするワードのデータをカレンダLSIにセットする。(日付・時刻設定)	2	504.95	201.98	
155	カレンダ演算		-[④ CLDS ⑤]-	現在日時から④を先頭とするワードの日付・時刻データを減算し、結果を⑤以降のワードに格納する	3	956.2	382.48	

ラダーラ命令(ファンクション)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)		
						T3	T3H	備考
関 数			-[④ DRUM (n) ⑥ → ⑦ (m)]-	カウント値⑥をカウント直設定テーブル(⑧+2n ~)と比較して、ステップNo.を決定し、⑨+1に格納する。 データ出力/パターンテーブル⑩により、このステップNo.に対応する出力パターンを参照し、ビットテーブル⑪に出力する。	6	36.52 +0.18n	16.46 +0.02n	
158	ドラムシーケンサ		-[④ CAM (n) ⑥ → ⑦ (m)]-	レジスタ⑥をテーブル⑧のそれぞれの立ち上り/立ち下り設定値と比較して、対応するデバイスのON/OFF制御を行う。	5	24.7 +11.55n	9.88 +4.62n	
159	カムシーケンサ		-[④ UL ⑥ → ⑦ (m)]-	⑨の内容に対し⑧の値で上限リミットをかけ、結果を⑩に格納する	4~7	12.6	5.04	
160	上限リミット		-[④ LL ⑥ → ⑦ (m)]-	⑨の内容に対し⑧の値で下限リミットをかけ、結果を⑩に格納する	4~7	12.6	5.04	
161	下限リミット		-[④ MAX (n) ⑥ → ⑦ (m)]-	⑩を先頭とするワードのデータテーブルの中で、最大値を⑧に、最大値が存在するポインタを⑨+1に格納する	4	22.23 +1.8n	8.89 +0.72n	
162	最大値		-[④ MIN (n) ⑥ → ⑦ (m)]-	⑩を先頭とするワードのデータテーブルの中で、最小値を⑧に、最小値が存在するポインタを⑨+1に格納する	4	22.23 +2.03n	8.89 +0.81n	
163	最小値		-[④ AVE (n) ⑥ → ⑦ (m)]-	⑩を先頭とするワードのデータテーブルの中で、平均値を計算し⑧に格納する	4	24.48 +2.58n	9.79 +1.03n	
164	平均値		-[④ FG (n) ⑥ → ⑦ (m)]-	⑩を先頭とする2×n個のパラメータによって定義される関数によって、⑧の内容を引数とする閾値を求め⑩に格納する	5~7	25.23 +2.85n	10.09 +1.14n	
165	閾数発生器		-[④ DB ⑥ → ⑦ (m)]-	⑩の内容に対して、⑧で示すステップバンドを持たせた値を⑩に格納する	4~7	16.3	6.12	
166	ステップバンド		-[④ +1/(⑧ RT ⑨)]-	倍長データ⑩+1・⑧の平方根を求め⑩に格納する	3~6	200.65	80.26	
167	ルート		-[④ INTG ⑥ → ⑦ (m)]-	⑩の値に対し⑨+1・⑧の積分定数より演算し、結果を⑩に格納する	4~7	44.1	17.64	

ラダー図命令(ファンクション命令)

グループ No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)	備考
関数	FUN No.			T3	T3H	
169	ランプ関数	-[A RAMP ③ → C]-	Ⓐの値に対し③以降のパラメータによりランプ関数を発生させ④に格納する	4~7	30.6	12.24
170	速度型PID	-[A PID ③ → C]-	Ⓐの値に対し③以降のパラメータにより速度型PID演算を行い④に格納する	4	44.45	17.78
171	偏差2乗型PID	-[A PID2 ③ → C]-	Ⓐの値に対し③以降のパラメータにより偏差2乗速度型PID演算を行い④に格納する	4	63.2	25.28
156	本質継承型PID	-[A PID3 ③ → C]-	Ⓐの値に対し③以降のパラメータにより不完全微分先行型PID演算を行い④に格納する	4		
172	正弦関数(SIN)	-[A SIN ③]-	Ⓐの値を1/100 した角度(deg)の正弦値の10000倍の値を④に格納する	3~5	37.35	14.94
173	余弦関数(COS)	-[A COS ③]-	Ⓐの値を1/100 した角度(deg)の余弦値の10000倍の値を④に格納する	3~5	38.6	15.44
174	正接関数(TAN)	-[A TAN ③]-	Ⓐの値を1/10000 した角度(deg)の正接値の10000倍の値を④に格納する	3~5	10.6	4.24
175	逆正弦関数(SIN ⁻¹)	-[A ASIN ③]-	Ⓐの値を1/10000 し逆正弦値を100 倍後、④に格納する	3~5	11.6	4.64
176	逆余弦関数(COS ⁻¹)	-[A ACOS ③]-	Ⓐの値を1/10000 し逆余弦値を100 倍後、④に格納する	3~5	12.6	5.04
177	逆正接関数(TAN ⁻¹)	-[A ATAN ③]-	Ⓐの値を1/10000 し逆正接値を100 倍後、④に格納する	3~5	480.7	192.28
178	指數関数	-[A EXP ③+1 · ④]-	Ⓐの絶対値の1/1000 の指數関数を求め④+1 · ④に格納する	3~5	423.2	169.28
179	常用対数関数	-[A LOG ③]-	Ⓐの絶対値の常用対数を演算し 1000 倍した値を④に格納する	3~5	543.2	217.28

ラダーラン命令(ファンクション)

グル ープ	FUN No.	名 称	表 現	概 要	所要 スルツフ数	所要実行時間(μs)	
						T3	T3H
テ ー タ 変 換	180	絶対値	-[④ ABS ③]-	④の絶対値を③に格納する	3~5	9.4	3.76
	181	倍長絶対値	-[④+1 · ④ DABS ③+1 · ④]-	④+1 · Aの絶対値を③+1 · ④に格納する	3~6	10.8	4.32
	182	2の補数	-[④ NEG ③]-	④の2の補数を③に格納する	3~5	9.0	3.6
	183	倍長2の補数	-[④+1 · ④ DNNEG ③+1 · ④]-	④+1 · ④の2の補数を③+1 · ④に格納する	3~6	11.7	4.68
	184	ダブルワード変換	-[④ DW ③+1 · ④]-	④の符号付きデータを倍長に変換する	3~5	10.3	4.12
	185	7セグメントコード	-[④ 7 SEG ③]-	④の下位4ビットをセグメントコードに変換する	3~5	9.4	3.76
	186	ASCII変換	-[④ ASC ③]-	④で示される最大16文字の英数字を ASCIIコードに変換し③に格納する	3~10	23.23 +0.83n	9.29 +0.33n
	188	バイナリ変換	-[④ BIN ③]-	④のBCDデータをバイナリデータに変換し③に格納する	3~5	34.65	13.86
	189	倍長バイナリ変換	-[④+1 · ④ DBIN ③+1 · ④]-	④+1 · ④の倍長BCDデータをバイナリデータに変換し④+1 · ④に格納する	3~6	81.45	32.58
	190	BCD変換	-[④ BCD ③]-	④のバイナリデータをBCDデータに変換し③に格納する	3~5	34.65	13.86
浮動小数点変換	191	倍長BCD変換	-[④+1 · ④ DBCD ③+1 · ④]-	④+1 · ④のバイナリデータをBCDデータに変換し④+1 · ④に格納する	3~6	33.8	13.52
	204	浮動小数点変換	-[④+1 · ④ FLT ③+1 · ④]-	④+1 · ④の倍長整数データを浮動小数点データに変換し④+1 · ④に格納する	3~5	11.04	5.03
	205	固定小数点変換	-[④+1 · ④ FIX ③+1 · ④]-	④+1 · ④の浮動小数点データを倍長整数データに変換し④+1 · ④に格納する	3	12.33	5.03
	206	浮動小数点絶対値	-[④+1 · ④ FABS ③+1 · ④]-	④+1 · ④の浮動小数点データの絶対値を③+1 · ④に格納する	3	11.29	4.5
	207	浮動小数点符号反転	-[④+1 · ④ FNNEG ③+1 · ④]-	④+1 · ④の浮動小数点データの符号を反転し④+1 · ④に格納する	3	11.29	4.68

5. プログラム言語

ラダー図命令(ファンクション命令)

グループ	FUN No.	名 称	表 現	概 要	所要ステップ数	T3	T3H	所要実行時間(μs)
D 演 算	B 192	BCD 加算	-[A B+ ⑧ → ⑩]-	ⒶとⒷの内容をBCD加算し、結果を⑩に格納する	4~7	63.15	25.26	
	C 193	BCD 減算	-[A B- ⑧ → ⑩]-	ⒶからⒷの内容をBCD減算し、結果を⑩に格納する	4~7	63.15	25.26	
	194	BCD 乗算	-[A B* ⑧ → ⑩+1 · ⑩]-	ⒶとⒷの内容をBCD乗算し、結果を⑩+1・⑩に格納する	4~7	99.15	39.66	
	195	BCD 除算	-[A / ⑧ → ⑩]-	ⒶをⒷの内容でBCD除算し、商を⑩に、余りを⑩+1に格納する	4~7	87.15	34.86	
	196	倍長BCD加算	-[A+1 · ⑧ DB+ ⑨+1 · ⑨ → ⑩+1 · ⑩]-	Ⓐ+1・Ⓐと⑨+1・⑨の内容をBCD加算し、結果を⑩+1・⑩に格納する	4~9	117.15	46.86	
	197	倍長BCD減算	-[A+1 · ⑧ DB- ⑨+1 · ⑨ → ⑩+1 · ⑩]-	Ⓐ+1・Ⓐから⑨+1・⑨の内容をBCD減算し、結果を⑩+1・⑩に格納する	4~9	117.15	46.86	
	198	倍長BCD乗算	-[A+1 · ⑧ DB* ⑨+1 · ⑨ → ⑩+1 · ⑩]-	Ⓐ+1・Ⓐと⑨+1・⑨の内容をBCD乗算し、結果を⑩+3・⑩+2・⑩+1・⑩に格納する	4~9	267.2	106.88	
	199	倍長BCD除算	-[A+1 · ⑧ DB / ⑨+1 · ⑨ → ⑩+1 · ⑩]-	Ⓐ+1・Ⓐを⑨+1・⑨の内容でBCD除算し、商を⑩+1・⑩に、余りを⑩+3・⑩+2に格納する	4~9	215.3	86.12	
C キャリー付き	200	BCD加算	-[A B+C ⑧ → ⑩]-	ⒶとⒷとキャリーフラグの内容をBCD加算し、結果を⑩に格納する。演算結果によってキャリーフラグは変化	4~7	64.8	25.92	
	201	BCD減算	-[A B-C ⑧ → ⑩]-	ⒶからⒷとキャリーフラグの内容をBCD減算し、結果を⑩に格納する。演算結果によってキャリーフラグは変化	4~7	65.3	26.12	
	202	キャリー付き 倍長BCD加算	-[A+1 · ⑧ DB+C ⑨+1 · ⑨ → ⑩+1 · ⑩]-	Ⓐ+1・Ⓐと⑨+1・⑨とキャリーフラグの内容をBCD加算し、結果を⑩+1・⑩に格納する。演算結果によってキャリーフラグは変化	4~9	118.3	47.32	
	203	キャリー付き 倍長BCD減算	-[A+1 · ⑧ DB-C ⑨+1 · ⑨ → ⑩+1 · ⑩]-	Ⓐ+1・Ⓐから⑨+1・⑨とキャリーフラグの内容をBCD減算し、結果を⑩+1・⑩に格納する。演算結果によってキャリーフラグは変化	4~9	120.3	48.12	

ラダーノット(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)		
						T3	T3H	備考
実 数 関 数	218	浮動小数点 上限リミット	-[@+1 · @ FUL @+1 · @ → @+1 · @]-	④+1 · ④の浮動小数点データに@+1 · ④の浮動小数点データで上限リミットをかけ、結果を@+1 · ④に格納する	4	23.72	8.46	
	219	浮動小数点 下限リミット	-[@+1 · @ FLL @+1 · @ → @+1 · @]-	④+1 · ④の浮動小数点データに@+1 · ④の浮動小数点データで下限リミットをかけ、結果を@+1 · ④に格納する	4	22.5	8.5	
	220	浮動小数点 テッドバンド	-[@+1 · @ FDB @+1 · @ → @+1 · @]-	④+1 · ④の浮動小数点データに@+1 · ④の浮動小数点データのテッドバンドを持たせた値を@+1 · ④に格納する	4	38.35	20.68	
	221	浮動小数点 ルート	-[@+1 · @ FRT @+1 · @]-	④+1 · ④の浮動小数点データの平方根を求め、④+1 · ④に格納する	3	366.44	54.3	
	222	浮動小数点 速度型 PID	-[@+1 · @ FPID @+1 · @ → @+1 · @]-	④+1 · ④の浮動小数点データにより速度型 PID 演算を行い、④+1 · ④に格納する	4	848.14	201.98	
	223	浮動小数点 偏差2乗型 PID	-[@+1 · @ FPID2 @+1 · @ → @+1 · @]-	④+1 · ④の浮動小数点データにより偏差2乗速度型 PID 演算を行い、④+1 · ④に格納する	4	1068.14	217.48	
	232	浮動小数点 本質継承型 PID	-[@+1 · @ FPID3 @+1 · @ → @+1 · @]-	④+1 · ④の値に対し④+1 · ④以降のパラメータにより不完全微分先行型 PID 演算を行い④+1 · ④に格納する	4			
	224	浮動小数点 正弦関数(SIN)	-[@+1 · @ FSIN @+1 · @]-	浮動小数点データ④+1 · ④の正弦値を④+1 · ④に格納します	3	705.54	129.08	
	225	浮動小数点 余弦関数(COS)	-[@+1 · @ FCOS @+1 · @]-	浮動小数点データ④+1 · ④の余弦値を④+1 · ④に格納します	3	791.74	148.48	
	226	浮動小数点 正接関数(TAN)	-[@+1 · @ FTAN @+1 · @]-	浮動小数点データ④+1 · ④の正接値を④+1 · ④に格納します	3	1471.64	259.48	
	227	浮動小数点 逆正弦関数(SIN ⁻¹)	-[@+1 · @ FASIN @+1 · @]-	浮動小数点データ④+1 · ④の逆正弦値を④+1 · ④に格納します	3	1504.64	213.98	
	228	浮動小数点 逆余弦関数(COS ⁻¹)	-[@+1 · @ FACOS @+1 · @]-	浮動小数点データ④+1 · ④の逆余弦値を④+1 · ④に格納します	3	1540.14	221.98	

ラダーノット(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)	
						T3	T3H
実 数 関 数	229	浮動小数点 正接関数(TAN^{-1})	-[A+1 · A FATAN B+1 · B]-	浮動小数点データ④+1・Ⓐの逆正接値を③+1・Ⓑに格納する	3	1170.14	189.98
	230	浮動小数点 指数関数	-[A+1 · A FEXP B+1 · B]-	浮動小数点データ④+1・Ⓐの指數関数を求め、③+1・Ⓑに格納します	3	838.34	141.08
	231	浮動小数点 常用対数関数	-[A+1 · A FLG B+1 · B]-	浮動小数点データ④+1・Ⓐの常用対数值を③+1・Ⓑに格納します	3	1261.0	206.98

ラダー図命令(ファンクション命令)

グル ープ	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要実行時間(μs)	
						T3	T3H
入 出 力	235	直接入力	-[I /0 (n) ④]-	入出力レジスタ④からnワードのレジスタ範囲について、対応する I/O モジュールとの間でデータの入出力を行う	3	25.84	(基本) 10.26 (拡張) 14.38 *6
	236	拡張データ転送	-[④ XFER ⑥ → ④]-	レジスタ④により指定される転送元の連続したデータをレジスタ⑥により指定される転送先へレジスタ⑥で示されるサイズのブロック転送する。	4	転送対象による による	転送対象にによる による *7
237	特殊モジュールデータ入力	-[④ READ ⑧ → ④]-	レジスタ④の特殊モジュールのレジスタ⑧で指定されたメモリアドレスからレジスタ④を先頭としたエリアにレジスタ⑧+1で示されるサイズの転送を行う。	4~5	764.14	(基本) 270.98 (拡張) 293.48 *8	
	238	特殊モジュール出力	-[④ WRITE ⑨ → ④]-	レジスタ④を先頭としたレジスタ⑨+1で示されるサイズのデータをレジスタ⑨の特殊モジュールのレジスタ⑨で示されるモリアドレスにブロック転送を行う。	4~5	748.14	(基本) 261.98 (拡張) 291.48 *9
239	伝送送信	-[④ SEND ⑩]-	レジスタ④を先頭とする伝送パラメータに従つてネットワーク経由でデータを送信し、レジスタ⑩に完了ステータスを格納します。				
240	伝送受信	-[④ RECV ⑪]-	レジスタ④を先頭とする伝送パラメータに従つてネットワーク経由でデータを受信し、レジスタ⑪に完了ステータスを格納します。				

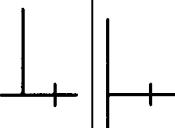
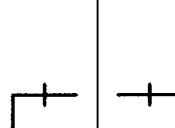
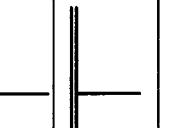
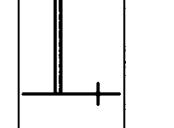
インテックス修飾、桁指定、直接 I/O 指定をする場合は、上表の処理時間が追加されます。

修飾形態	オペランド形態			T3 (μs)	T3H (μs)		
	単長	倍長	テーブル		単長	倍長	テーブル
インテックス	16.9	17.1	17.1	5.4	6.7	6.7	
桁指定	16.8	25.0	27.5+7.5 × (n+1)	6.0	10.0	11+3 × (n+1)	
桁指定 I/O	基本	38.0	57.9	45.5+16.4 × (n+1)	14.6	22.3	14+6.26 × (n+1)
直接 I/O	基本	11.0	21.0	7.5+9.5n	4.3	7.2	3+3.5n
	拡張				8.8	16.2	3+8n

n=テーブルサイズ

SFC命令	FUN No.	名 称	表 現	概 要	所要 ステップ数	所要 実行時間(μs)
ゲル ープ					T3	T3H
SFC イニシャルライズ		Nxx Ⓐ nnnn	Y	デバイスⒶがOFFからONに変化したとき、以降に続くSFCプログラムのnnn個のステップをイニシアルティブにし、イニシャルステップをアクティブにする(SFCの起動)	4	493.7 197.48
ステップ	イニシャルステップ	sss	□	SFCプログラムの開始を示す、1対1に対応する実行詳細部あり。sssはステップ番号	(2 〔詳細部 除く〕 1 〔詳細部 除く〕 1)	7.875 3.15
ステップ	エンドステップ	sss	□	SFCプログラムの1単位。1対1に対応する実行詳細部があり。sssはステップ番号	(1 〔詳細部 除く〕 1 〔詳細部 除く〕 1)	3.0 1.2
マクロステップ	マクロステップ	ssss mmmm	□	SFCプログラムの終了を示す。直前の遷移条件成立時に對応するイニシャルステップへ戻る。ssssはイニシャルステップ番号 mmmmで示されるマクロプログラムに1対1に対応する。sssはステップ番号、mmmはマクロ番号	2	3.15 1.26
待ち時間付きステップ	待ち時間付きステップ	ssss xxxxx	W +	直後の遷移条件が成立しても設定時間経過までは遷移を行わない。1対1に対応する実行詳細部あり。sssはステップ番号、①はタイマレジスタ、xxxxxは設定時間	(4 〔詳細部 除く〕 4)	9.9 3.96
渋滞検出付きステップ	渋滞検出付きステップ	ssss xxxxx	A ① Ⓐ	アクティブ時間監視し、設定時間内に遷移しないときⒶのアームデバイスをONにする。1対1に対応する実行詳細部あり。sssはステップ番号、①はタイマレジスタ、xxxxxは設定時間、Ⓐはアームデバイス	(5 〔詳細部 除く〕 5)	10.8 4.32

SFC 命令 グループ	FUN No.	名 称	表 現	概 要	所要実行時間 (μs)		
					ステップ数	T3	T3H
遷 移	遷移		+	ステップ間の遷移条件を示す。1 対 1 に対応する遷移詳細部あり	1 (詳細部) 除く	5. 6	2. 24
エンド	エンド		①↓↑	SFC プログラムの終了を示す。遷移条件成立時に 1111 で示すラベルへジャンプする。1 対 1 に対応する遷移詳細部あり	2 (詳細部) 除く	6. 53	2. 61
ジャンプ	ジャンプ		①↓←↑	任意のステップへのジャンプを示す。条件成立時に 1111 で示すラベルへジャンプする。1 対 1 に対応するジャンプ条件詳細部あり	5 (詳細部) 除く	8. 03	3. 21
マクロエンド	マクロエンド		↑ E	マクロプログラムの終了を示す。1 対 1 に対応する遷移詳細部あり	2 (詳細部) 除く	6. 53	2. 61
ラベル	ラベル		①↓↑	エンドからの戻り先またはジャンプからの飛び先を示す	2	11. 0	4. 4
	マクロエントリ		mmm M	マクロプログラムの開始を示す	1	3. 0	1. 2

SFC命令	グループ	名 称	表 現	概 要	所要ステップ数	所要実行時間(μs)
	GRU	UN No.			T3	T3H
条件 分岐	条件 分岐	条件分流(I)		接続されているステップ中、遷移条件が成立しているステップをアクティブにする(左方優先)	$2 \times n - 1$ nは分岐 列数	6.45 2.58
	条件 分岐	条件分流(II)			6.45	2.58
	条件 分岐	条件分流(III)			5.77	2.31
	条件 分岐	条件分流合流			0.15	0.09
	並列 分岐	並列分流(I)		接続されているステップ全てをアクティブにする	$n + 3$ nは分岐 列数	0.15 0.09
並列 分岐	並列 分岐	並列分流(II)			0.15	0.09
	並列 分岐	並列分流(III)			0.15	0.09
	並列 分岐	並列分流合流			5.18	2.07
	並列 分岐	並列分流(II)			8.79	3.52

(備考欄補足)

備考欄No.		処理速度(μs)	
		T3	T3H
*1	レジスタ → レジスタ	0.9	0.54
	定数 → レジスタ	7.65	3.06
*2	レジスタ + レジスタ	1.5	0.9
	定数 + 定数	10.85	4.34
*3	レジスタ - レジスタ	1.5	0.9
	定数 - 定数	10.9	4.36
*4	レジスタ × レジスタ	4.35	1.08
	定数 × 定数	15.35	6.14
*5	レジスタ / レジスタ	7.65	4.59
	定数 / 定数	22.7	9.08
*6	n:ワード数	基本ユニット	6.8+3.05n
		拡張ユニット	15.71+10.13n
*7	n:ワード数	D→S20(基本ユニット)	286.48+4.5n
		D→S20(拡張ユニット)	757.03+11.23n
		S20→D(基本ユニット)	302.46+9.02n
		S20→D(拡張ユニット)	394.69+7.49n
		D→EEPROM	1086.22+11.92n
		EEPROM→D	417.97+9.51n
		D→拡張レジスタ	871+5.24n
		拡張レジスタ→D	252.44+1.54n
		D→D	519.1+5.14n
*8	n:ワード数	基本ユニット	185.88+1.58n
		拡張ユニット	516.95+5.19n
*9	n:ワード数	基本ユニット	186.75+1.53n
		拡張ユニット	524.96+5.18n

D:本体レジスタ, S20:TOSLINE-S20

索引

[ア行]

アナンシェータリレー	128, 166, 167, 168
アプリケーションプログラム(ユーザプログラム)	22, 23, 137, 142
1回実行モード	85, 88, 169
一括出力	11, 77
一括入出力処理	75, 77
一括入出力方式	78
一括入力	11, 77
イニシャルステップ	231
イニシャルロード	10, 30, 67, 74, 94, 95
イベント履歴	99
イメージテーブル	11, 120
インデックス修飾	191
インデックスレジスタ(I, J, K)	27, 30, 68, 158, 191
ウォッチドックタイマ	98
運転コマンド	8, 70, 71
エンド(SFC)	231
エンドステップ	232
オンライントレース機能	108
オンラインプログラム変更	82, 101, 117

[カ行]

回路	25, 220
回路実行機能	123
回路番号	24, 220
カウンタリレー(C.)	28, 30, 67, 157
カウンタレジスタ(C)	27, 30, 67, 157
拡張ファイルレジスタ	95, 158
カレンダタイマ機能	97, 98, 164
機能仕様	64
基本命令	225
強制運転コマンド	8, 70, 71
桁指定	195
個別割り付け	13, 208
コメント	135, 151
コンピュータリンク伝送/パラメータ	140

[サ行]

サイクリック実行モード	85, 90, 169, 170
サブプログラム	84, 85, 144
サブプログラム実行時間	140
サブルーチン	149
サンプリングトレース機能	109
サンプリングバッファ容量	139
シーケンシャル・ファンクション・チャート (SFC)	218, 227
シーケンス渋滞検出機能	131
自己診断項目	96
システム構成	63
システムコメント	139
システム初期化処理	66
システム情報	23, 81, 137, 139
実行時間計測機能	108
実数	184, 190
自動割り付け	13, 81, 201
ジャンプ (SFC)	234
周辺サポート処理	80
渋滞検出付きステップ	235
出力デバイス (Y)	28, 30, 67, 154, 155
出力レジスタ (YW)	14, 27, 30, 67, 78, 154, 155
瞬停継続機能	66, 87, 104, 169
瞬停検出機能	102
瞬停時間設定機能	102, 170
瞬停判定	66
使用ステップ	139
詳細部 (SFC)	236
条件分岐 (SFC)	232
シングルステップ実行機能	119, 121
診断表示機能	128, 166
スキャン	11, 73
スキャン周期	11, 98, 108
スキャン時間	140
スキャンモード	74, 75
スキャンモードの選択方法	76, 140
スタートモード	8, 140
ステータスマニタ機能	108

索引

ステータスラッチ機能	115
ステップ(SFC)	227, 231
ステップ番号(SFC)	227
整数	184, 185
正整数	184, 185
正の倍長整数	184, 187
遷移(SFC)	227, 231

[夕行]

タイマ更新処理	75, 79
タイマリレー(T.)	28, 30, 67, 154, 157
タイマレジスタ(T)	27, 30, 67, 79, 154, 157
タイミングリレー	79, 80, 163
ダイレクト方式(直接入出力方式)	78
直接出力デバイス(O)	28, 29, 78, 154, 156
直接出力レジスタ(OW)	27, 29, 78, 154, 156
直接入出力方式	78
直接入力デバイス(I)	28, 29, 78, 154, 156
直接入力レジスタ(IW)	27, 29, 78, 154, 156
データ初期化	30, 67, 74
データレジスタ(D)	27, 30, 67, 154, 158
定刻スキャン	76, 87
停止コマンド	8, 70, 71
停止条件設定機能	119, 124
定周期割り込み	92, 146
定周期割り込み周期	140
停電保持範囲指定	140, 223
停電保持範囲情報	23
デバイス	28, 152
デバッグコマンド	70, 71
デバッグモード機能	119
伝送入出力割り付け情報	141, 209
動作モード	7, 68, 69
特殊リレー(S)	28, 30, 67, 154, 157, 159
特殊レジスタ(SW)	27, 30, 67, 154, 157, 159
時計機能(カレンダタイマ機能)	97, 98, 164

[ナ行]

入出力シミュレーション機能	120
入出力割り付け	13, 23, 200
入出力割り付け情報	23, 81, 141, 201
入出力割り付けのルール	16, 205
入力デバイス(X)	28, 30, 67, 154, 155
入力レジスタ(XW)	14, 27, 30, 67, 78, 154, 155

[ハ行]

倍長BCD	184, 189
倍長整数	184, 188
パスワード機能	132
パルス化	225
ビットパターンチェック機能	129, 165
ファイルレジスタ(F)	27, 30, 68, 154, 158
ファンクションブロック	222
ファンクション命令	225
フォース機能	117
浮動小数点データ(実数)	184, 190
ブレークポイント設定機能	119, 120
フローティングスキャン	75, 87
プログラマのPC制御コマンド	8, 68, 70, 71
プログラミング手順	4, 38
プログラムID	139
プログラム言語	217
プログラム種別	24, 25, 84, 138, 142
プログラムの実行順序	26, 223, 238
プログラム容量	139
ブロック	24, 25, 138, 142
並列分岐(SFC)	233
補助リレー(R)	28, 30, 67, 154, 157
補助レジスタ(RW)	27, 30, 67, 154, 157
本体動作フロー	10, 65

[マ行]

マクロエントリ	234
マクロエンド	234
マクロステップ	233

索引

マクロプログラム	229
待ち時間付きステップ	235
マルチタスク機能	84, 242
マルチタスクによる注意事項	242
メインプログラム	84, 143
メモリクリア	81, 101
メモリプロテクト機能	101
メモリ容量	139
モードスイッチ	8, 68, 70, 71
モード制御処理	68, 75
モード遷移条件	9, 68, 70, 71
モジュール種別	14, 201

[ヤ行]

ユーザデータ	27, 152
ユーザデータ初期化	11, 67, 74
ユーザプログラム	22, 24, 137, 142
ユーザプログラム実行	11, 75
ユーザプログラムチェック	74
ユーザプログラムメモリ	22, 137
ユニット先頭アドレス設定	20, 204

[ラ行]

ラダー図	217, 220
ラダー図命令の注意事項	225
ラベル(SFC)	235
リアルタイムクロック機能(カレンダタイマ機能)	97, 98, 164
リトライ	100
リフレッシュ方式(一括入出力方式)	78
リレーシンボル	221
リンクリレー(L)	28, 30, 68, 154, 158
リンクレジスタ(W)	27, 30, 68, 154, 158
リンクレジス்டリレー(Z)	28, 30, 68, 154, 158
レジスタ	27, 152
レジスタ/デバイスのアドレス範囲	27, 28, 154
レジスタ/デバイスの初期化(ユーザデータ初期化)	11, 67, 74
レジスタ値正当性チェック機能	128, 165

[ワ行]

割り込み機能付きI/Oモジュール	146
割り込みの許可/禁止	93, 243
割り込みプログラム	84, 92, 146
割り込みプログラムの優先度	93, 148
割り込み割り付け情報	141, 148

[B]

BASIC	219
BCD	184, 186

[D]

DEBUGモード	69, 119
----------------	---------

[E]

EEPROM	10, 67, 94
EEPROM書き込み	83, 94, 101
EEPROM読み出し	83, 94, 101
ERRORモード	7, 69

[H]

HALT(モードスイッチ)	8, 70, 71
HALTモード	7, 69
HOLDモード	69, 169

[I]

i 指定	77, 203
ICメモリカード	10, 67, 95
ICメモリカード書き込み	83, 95, 101
ICメモリカード初期化	83, 95
ICメモリカードバッテリ異常	162
ICメモリカード読み出し	83, 95, 101
I/O活線着脱機能	106
I/O実装状態チェック	74
I/O割り込み	92, 146
I/O割り込みプログラムの応答時間	93
I/O割り込みプログラムの割り付け	93, 147

索引

[N]

Nスキャン実行機能 119, 124

[P]

PCタイプ 139

P-RUN(モードスイッチ) 8, 70, 71, 101

[Q]

Q0～Q8(桁指定) 195

[R]

RAM(RAM/ROM切換スイッチ) 8, 70, 71

RAM/ROM切換スイッチ 8, 67, 70, 71, 74

RAS機能 96

ROM(RAM/ROM切換スイッチ) 8, 70, 71

RUN(モードスイッチ) 8, 70, 71

RUNモード 7, 69

RUN-Fモード 7, 69

[S]

SFC 218, 227

SFCイニシャライズ 230

SFCプログラムの注意事項 239

SFCプログラムの容量制限 239

SFCメインプログラム 228