

FA-3Sシリーズプログラマブルコントローラ

伊 沢 克 己*¹⁾ 米 田 親 男*²⁾ 大 橋 昭 博*³⁾
吉 田 功 二*⁴⁾ 高 見 浩 志*⁵⁾

1. はじめに

1970年代初頭にリレーなどを使用したワイヤロジックな制御装置に代わるコントローラとして登場したPC(プログラマブルコントローラ)は、プログラマビリティと高速性が市場に受け入れられて、各種機器、装置のコントローラとして普及してきた。そして、年々進歩するマイクロエレクトロニクスをはじめとする技術を取り入れ、小形化・低価格化による底辺の拡大と高性能化・高機能化による用途の拡大を図り、今日ではFAをはじめとする自動化システムの汎用コントローラとして産業分野全般にわたって使用されるに至っている。

一方、市場ニーズの多様化・製品の短納期化に対応してより生産性を上げるために、処理速度への高速化要求や、よりフレキシブルな生産システムが求められている。また、従来PC、NCなどを用いた自動化装置はそれぞれに独立したシステムであったが、通信機能やデータ加工機能を高めて、それぞれの装置が有機的に結合できるシステムへと発展した。そして、受注・製造、販売に関わる情報・物の流れを一元化する企業全体のシステムの統合化(CIM化)へと展開されつつある。

このような背景の中、小・中規模用プログラマブルコントローラFA-3Sシリーズを開発した。図1に外観を示す。表1に仕様の概要を示す。

2. 開発思想

PCはFA化推進には不可欠なコンポーネントとなり、工場設備の自動化を考えた場合、中核コンポーネントとして位置付けされてくる。また、使用される分野が広がって、PCに対する制御ニーズはますます多様化しており、個々の状況に応じたシステムの提供が必要である。

FA-3Sの開発にあたっては、これらの要求に応えるべ

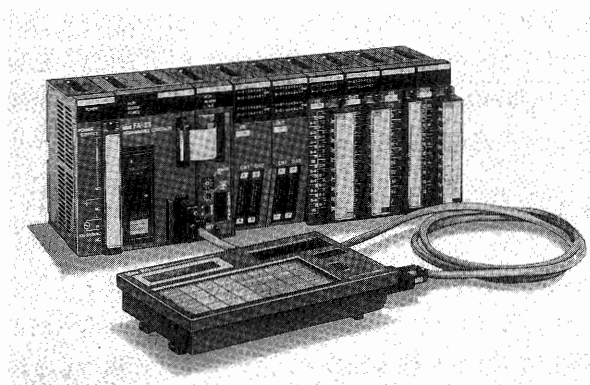


図 1.

表 1. FA-3S仕様

項目	仕様
制御方式	ストアードプログラム方式
プログラム方式	リレー接点式(論理シンボル)
命令語	基本命令 15種 応用命令 122種
プログラム容量	4Kステップ(4036ステップ)
メモリバックの種類	CMOS-RAM(バッテリー付) 1K, 4Kステップ E ² PROM 4Kステップ EPROM 4Kステップ
スキャンタイム	基本命令・平均約0.3μsec/1命令
入力・出力点数	入力・出力最大合計256点 (入力・出力割付比率フリー)
内部リレー	1024点(240点キープ可)
特殊内部リレー	32点
シフトレジスタ	224点(全点キープ可)
シングルアウトプット	256点
タイマ	256点(減算式 0~999.9sec) 80点(減算式 0~99.99sec)
カウンタ	100点(加算式 0~9999 全点キープ可) 2点(可逆カウンタ、クロック切換式、ゲート切換式)
データレジスタ	1000点(全点キープ可)
コントロールレジスタ	72点
自己診断機能	CPUエラー(W.D.T)、通信エラー、ユーザプログラムのCRCエラー、オペコードエラー

く、『使用目的に最適なシステムをむだなく構築できるPC』を目指して、下記方針で開発に臨んだ。

- (1) 取付スペースを有効利用できるコンパクトサイズ。
- (2) 状況に応じたフレキシビリティ。システム構成の自在性。システムの追加・変更が容易。
- (3) ラダープログラム開発効率を上げる、プログラム

* 1) 製品開発センターソフト開発グループ 課長
* 2) 製品開発センターソフト開発グループ
* 3) 製品開発センターソフト開発グループ
* 4) 研究部
* 5) 製品開発センター第7開発グループ

作成・編集機能, デバッグ機能などの向上。

- (4) 周辺機器との接続の汎用性。
- (5) 高速化, 高機能化。
- (6) 現場の環境条件に耐える信頼性の確保。
- (7) 従来資産の継承

3. システム構成

図2に基本システム例を示す。システムは、2スロット用、3スロット用の2機種種のベースモジュールにCPUモジュール、電源モジュール、各種入出力モジュール、高機能モジュールを組合せて、最大入出力256点までむだなく構成する。各ベースモジュール間は、増設ケーブルにより接続されている。

4. 特長と機能説明

FA-3Sの主な特長を以下に述べる。

(1) クラス最高の高速処理

シーケンス処理専用プロセッサの開発により、シーケンス基本命令が0.3μsec/1命令と、入出力256点規模のPCとして、トップレベルの処理速度を実現した。

(2) 省スペース

表面実装技術(SMT)により小形化を図り、ベースモジュールの増設時には密着取付が可能な構造にしたため、盤取付面積を大幅に縮小した。2スロットお

よび3スロットのベースモジュールの組合せと電源モジュールの増設位置をフリー(容量上必要な所へ配置)にしたことにより、むだのないきめ細かなシステムの構築が行なえる。

(3) フレキシブルな分散配置

2モジュールまたは3モジュールの単位で最大延長10mまでの分散配置が行えるので、盤内のレイアウトが容易である。入出力の増設など、システム拡張時にも空スペースを有効利用できる。

リモートI/Oシステムでは、最大延長1kmの分散配置が可能である。

(4) ローコストアナログ処理

マルチプレクサモジュールを用意することにより、経済的価格のアナログ入力処理を実現した。

(5) 高機能制御

簡単な命令語で動作するアナンシェータ、ドラムシーケンスなどの機能を搭載した。

(6) デバック機能

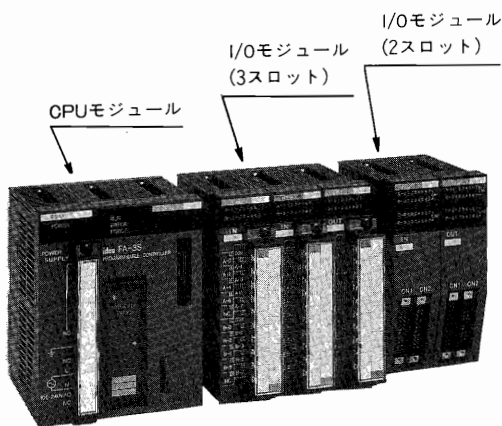
運転中のプログラム変更機能や強制入出力(フォース)機能などを用意することにより、システム立上げやシステム変更を容易にした。

(7) ラダー図入力用ソフトウェア

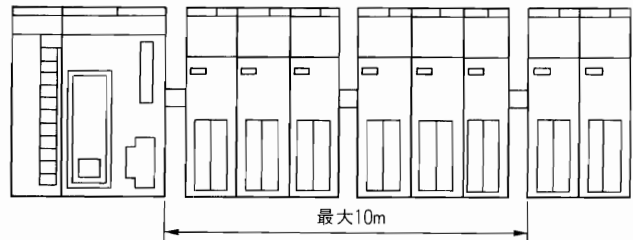
高価な専用プログラマの購入負担を軽減するために、市販のパソコンとアプリケーションソフトウェアで高機能グラフィックローダを実現した。

(8) 資産継承

最大I/O 256点(入力、出力割付比率フリー)まで、自在にシステム構成できます。

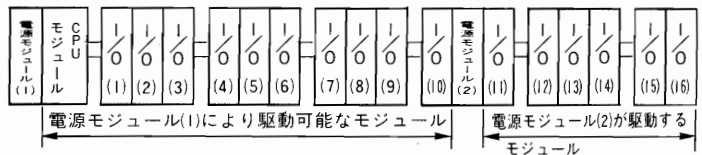


(最大I/O 256点を全モジュール32点モジュールで構成した場合の例)



CPUモジュールからI/Oモジュールは最大10mの延長が可能です。

(増設用の増設モジュールを使用する場合の例)



電源モジュール(1)の電源容量を超えるモジュールを実装する場合、上図のように増設用として電源モジュール(2)が必要です。電源モジュール(2)はこの例の場合、I/Oモジュール(11)~(16)に電源を供給します。

図2. 基本システム例

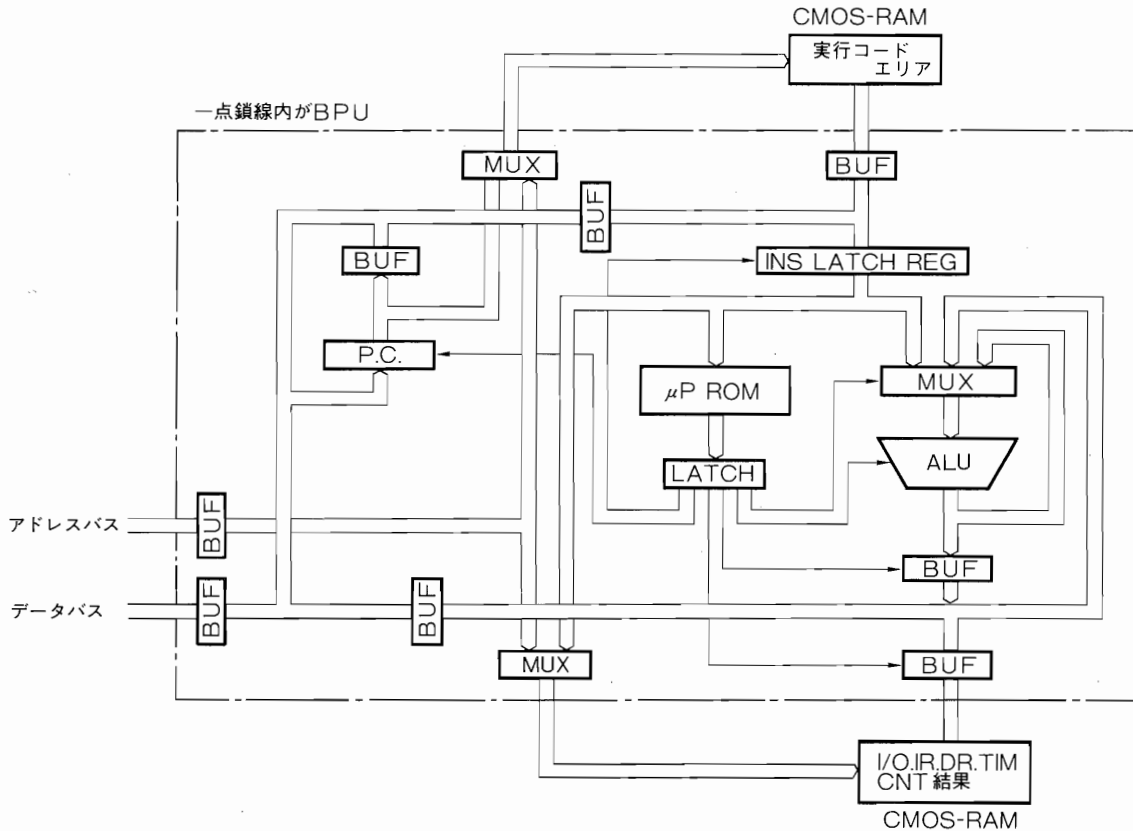


図3. BPU概略ブロック図

従来のFAシリーズプログラミングが継続して使用できるのでシステムのレベルアップが容易である。増設I/OバスラインはFAシリーズ(FA-1J, FA-2J)との相互接続を可能にした。

4.1 CPUモジュール動作

CPUモジュールは、2個のプロセッサを使用してPCに必要な処理を実行している。PCの処理は大きく分けると、先頭ステップより各種命令を実行する部分(シーケンス処理)と、外部のI/Oモジュールへの処理・エラーチェックなどのツールサービスを行う部分(END処理)の2つである。また、シーケンス処理は、基本命令処理(ビット演算処理)と応用命令処理(ワード演算処理)とに分けられる。BPU(ビット・プロセッシング・ユニット)が基本命令と応用命令の一部を実行し、その他の処理をP80C31BH-1が実行している。

専用プロセッサ(BPU)を開発することにより、ロード、アンドなどのシーケンス基本処理の高速化を図った。BPUは、基本命令を $0.3\mu\text{sec}/1$ 命令で処理する。また、タイマ・カウンタ・ワード加減算・データ比較・データタムプの命令も $0.8\sim 1.9\mu\text{sec}$ と高速で処理する。

応用命令、ツールサービス部は、ユーザープログラムにより処理時間を短縮させる機能を持たせている。

4.1.1 BPU動作

BPUは、4000ゲートのゲートアレイを用いて開発した。

汎用マイコンにより処理されていたときには、マイコンに備えられた命令を組合せてPCの1つの命令を処理していたが、専用プロセッサにより1つの命令で処理できるため、高速化が行える。

BPUの概略ブロック図を図3に示す。ユーザープログラムをコンパイルした命令コードが、実行コードエリアに格納されている。BPUは、その実行コードエリアから命令をフェッチ、デコードしたときに応用命令と判断すると、P80C31BH-1に処理を実行させている。BPUの動作として、命令の取り込み、命令のデコードおよびオペランドの読み出し、演算処理の3つの動作があるが、これらの動作を同時に行なうためパイプライン処理を行っている。

4.1.2 8031CPUの動作

8031CPUでは、I/Oドライブを含めたEND処理、通信処理、大半の応用命令処理、BPUのコントロールやデータアクセス等を実行している。BPUによる処理は大変高速であるが、ソフトウェアによって実行される8031の処理スピードは、かなり遅い。しかしながら、処理の複雑さ、柔軟さにおいては圧倒的にBPUに優っているため、両者の機能を分担させて全体的に高機能化と高速化を計っている。またソフトウェア処理自身においても、“実装I/O数の設定”、“各種ENDサービスの使用/不使用の設定”、“複合の演算応用命令”、“特殊内部リレーやコント

ロール・データレジスタの追加”等の機能を加え、総合的なフレキシビリティと高速化を実現させている。以下にそれぞれの機能について概略の説明を行う。

(1) 実装I/O数の設定

FA-3Sは、FAシリーズPCとのコンパチビリティを重視しているため、I/Oも従来のシフトI/Oを採用している。したがって、I/Oドライブに要する時間は1点あたり約23 μ secとあまり速くはない。そこで、実際に使用する入力点数および出力点数を設定することにより実使用時のI/Oドライブ時間を短縮させられる機能を設けた。

(2) 各種ENDサービスの使用/不使用の設定

高機能化に伴いEND処理中にいろいろなサービスをもりこんだため、当然END処理時間が長くなり、高速化の要求とは相反するので、使用しないサービスに対してはその旨設定して処理時間を速められるようにした。これは処理時間の短縮と同時に、ユーザシステムのデバッグ中にだけ、あるサービスを停止しておくというような目的にも利用できる。例えばリモートI/Oを使用するシステムで、リモートI/O用プログラムは既にダウンロードしているがリモートI/O親局モジュールがまだ未実装なので、そのサービスを停止させたいという場合に効果がある。設定可能なENDサービスには、“スキャンタイム測定とプログラムエリアのコードチェック”、“フォース”、“リモートI/O”、“通信モード変更やユーザ定義通信”等の各種サービスがある。

(3) 複合の演算応用命令

従来の演算命令を組み合わせて新たに1ヶの複合的な演算応用命令を作り、プログラムステップ数の節減と命令実行時間の短縮を計った。例えば、配列データ処理(連続したタイマ・カウンタ・データレジスタを配列とみなし、データ検索・置換・加算を行う命令)、ビットデータ処理(データレジスタ内の任意の1ビットのセット・リセットやリードアウト等の命令)、プログラム分岐処理(間接タグジャンプ命令、間接タグコール命令、プログラムnヶ分岐用のジャンプ/コール命令)等の演算応用命令を用意した。

(4) 特殊内部リレーやコントロールDR

システムで予め定義した内部リレーやデータレジスタをいくつか設けフレキシビリティを高めた。例えば、特殊内部リレーとしては、“ユーザ割込スピードオーバ”、“ラン中のプログラム書込許可”、“フォース設定中”、“連続データのシフト中や連続したSET/RST命令実行中のパワーダウンに対する、復電再スタート時のホットスタート可能性判定”等のフラグがあり、また、コントロールDRとしては、“プロテ

クトコード”、“FA-3Sエラー/アラームコード”、“ユーザプログラムエラーコードとステップ番号”、“1スキャンタイム測定値”、“FA-3SシステムPROMバージョン番号”、“リモートI/O関係の設定/読出”、“通信関係の設定/読出”等々の各情報を読出、あるいは登録できるレジスタを用意した。

4.2 リモートI/Oについて

○概要

本リモートI/Oシステムは各種I/OモジュールをCPUモジュールより離して設置したい場合や、I/O点数を基本システム(最大I/O256点)より増やしたい場合などに構築するシステムである。

リモート親局モジュールは、ユーザプログラムにより、リモートI/Oケーブルで接続された各リモート子局モジュールに対して出力データの送信、入力データの収集を行わない、システム全体を監視、制御する。

リモート子局モジュールは、リモート親局からの出力データを出力モジュールに出力し、入力モジュールからの入力データをリモート親局に送信する。

○システム構成、及び仕様

リモートI/Oシステムは、CPUモジュールとデータバスを介して接続されたりリモート親局モジュールと、分散配置された各種I/Oモジュールを接続したりリモート子局モジュール、及びそれらの間を光ファイバで結ぶリモートI/Oケーブルとから構成される。本リモートI/Oシステムでは、リモート子局間の距離に応じて親局、子局モジュール、それぞれに短距離型と中距離型の2種類のモジュールと布設条件によって4種類のリモートI/Oケーブルを用意している。リモートI/Oシステム構成例を図4に、またそれぞれの仕様を表2に示す。

○リモートI/Oシステム動作

リモートI/Oシステムは多様化するユーザニーズに対応して、柔軟なシステムが構築できるように種々の機能を備えている。その一つとして、リモートI/Oスキャン同期モード、非同期モードがある。同期モードでは、リモートI/Oリフレッシュサイクル(図5)とプログラムのスキャンを同期させるためにプログラムのEND処理を待機させている(図6)。非同期モードではEND処理を待機せず、プログラムのスキャンを優先させている(図7)。同期モードは、リモート子局の入出力に表示器やデジタルスイッチ等入出力データとラッチデータが同期しないと正常動作しない機器を接続する場合などに設定するモードである。非同期モードは、入出力データのオン、オフが比較的ゆっくりしていて、プログラムのスキャンタイムを短くしたい場合に設定するモードである。また、リモートI/Oリフレッシュサイクルでの親局、子局間の通信形態は親局が子局にコマンドを与えることにより子局が

- 子局接続台数：1系列最大8局、システム2系列接続可能。
- 最大I/O点数：1子局合計64点(入力32点、出力32点)
1系列合計512点(入力256点、出力256点)
- 通信路：光コネクタ付光ファイバ
- 通信時間：約38.5msec(リモート子局8局構成時32点モード)
約10.6msec(リモート子局1局構成時16点モード)

- 通信距離
- (1)短距離形(親局モジュール：PF3S-RMP1形、子局モジュール：PF3S-RTP1形)の場合
 - ①APF(プラスチック光ファイバ)使用時(PF3S-KP1、KP2形)使用時
局間最大25m、総延長最大200m
 - ②PCF(プラスチックラッド石英光ファイバ)(PF3S-KP3、KP4形)使用時
局間最大50m、総延長最大400m
- (2)中距離形(親局モジュール：PF3S-RMQ1形、子局モジュール：PF3S-RTQ1形)の場合
PCF(プラスチックラッド石英光ファイバ)(PF3S-KP3、KP4形)使用時
局間最大400m、総延長最大1km

	リモート親局モジュール	リモート子局モジュール	リモートI/Oケーブル	
			PF3S-KP1、KP2	PF3S-KP3、KP4
短距離形	PF3S-RMP1	PF3S-RTP1	使用可	使用可
中距離形	PF3S-RMQ1	PF3S-RTQ1	使用不可	使用可

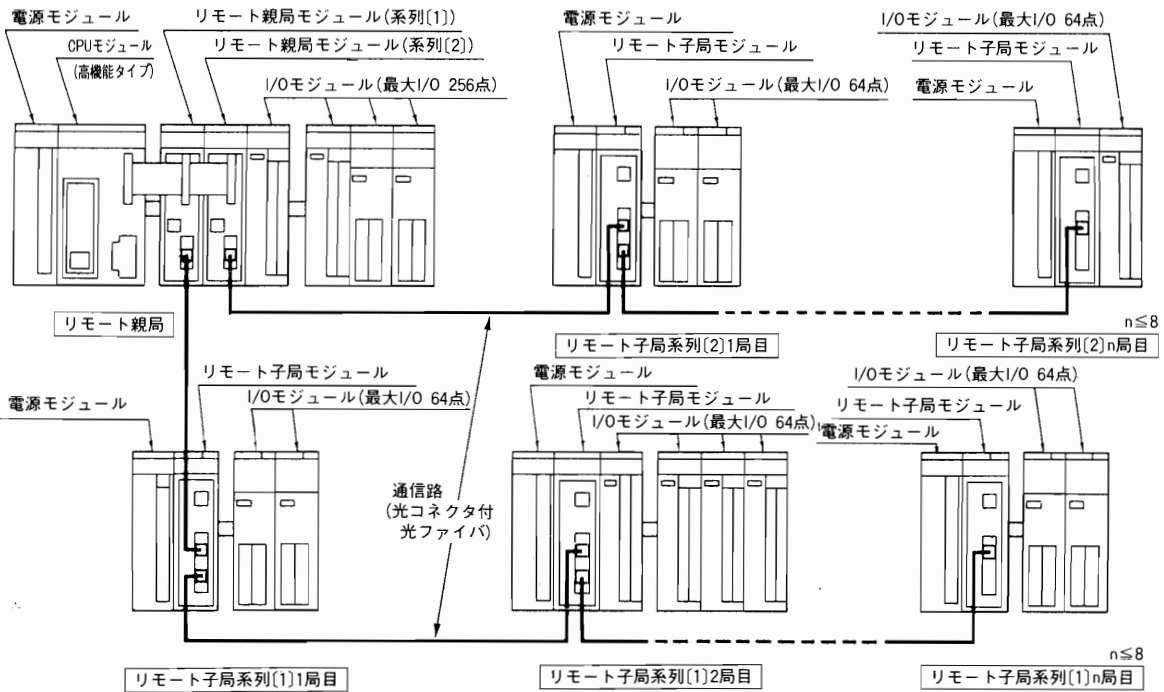


図 4. リモートI/Oシステム構成例

表 2. 仕様

●リモートI/Oシステム

	モジュール	形番	通信距離
短距離形	リモート親局モジュール	PF3S-RMP1	●局間最大 25m 総延長最大 200m 使用光ファイバ:PF3S-KP1、KP2
	リモート子局モジュール	PF3S-RTP1	
中距離形	リモート親局モジュール	PF3S-RMQ1	●局間最大 400m 総延長最大 1000m 使用光ファイバ:PF3S-KP3、KP4
	リモート子局モジュール	PF3S-RTQ1	

●性能仕様

システム形態	1:N形
通信路形態	バス型
通信線	光コネクタ付光ファイバ
通信距離	上記参照
通信時間	図4参照
通信方式	半二重
同期方式	調歩同期
通信プロトコル	ポーリングセレクション方式
通信速度	38.4kbps
符号形態	NRZ、ベースバンド
誤り制御	サム、オーバーラン、フレーミング、通信タイマ
リモート親局接続数	PF3S-CP12形(高性能タイプ)CPUモジュール1台当り最大2局
リモート子局接続数	リモート親局モジュール1台当り最大8局

●リモート親局モジュール (1局当り)

形番	PF3S-RMP1(短距離形) PF3S-RMQ1(中距離形)
割当 入力点数	最大 256点
割当 出力点数	最大 256点
コントロールデータレジスタ	読み専用 2レジスタ 書き専用 1レジスタ
リモート子局接続数	最大8局
消費電流	180mA
重量(約)	250g

●リモート子局モジュール (1局当り)

形番	PF3S-RTP1(短距離形) PF3S-RTQ1(中距離形)
割当 入力点数	最大 32点
割当 出力点数	最大 32点
I/O増設ケーブル	総延長3m以内
消費電流	250mA
重量(約)	200g

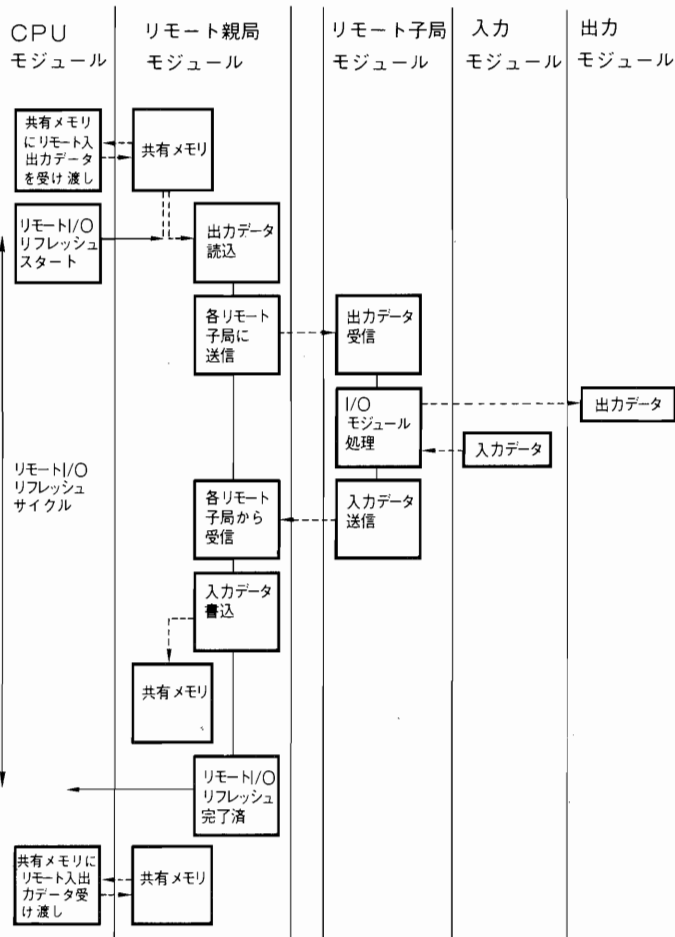
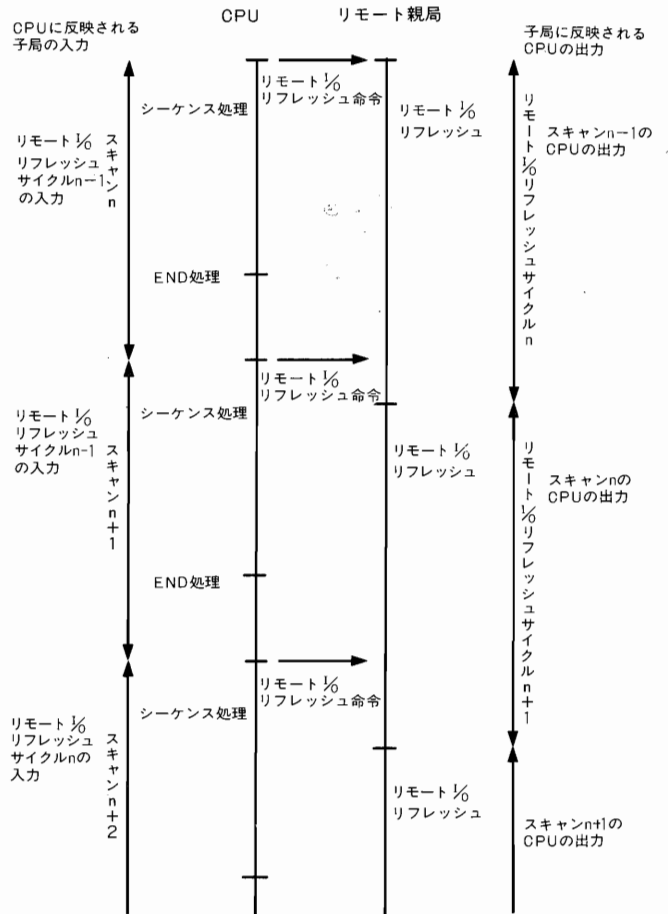


図5. リモートI/Oリフレッシュサイクル



(注) シーケンス処理時間 \geq リモート $\frac{1}{6}$ リフレッシュ時間の場合、非同期モードでも同期モードと同様の処理の流れになる。

図7. 非同期モードの処理の流れ

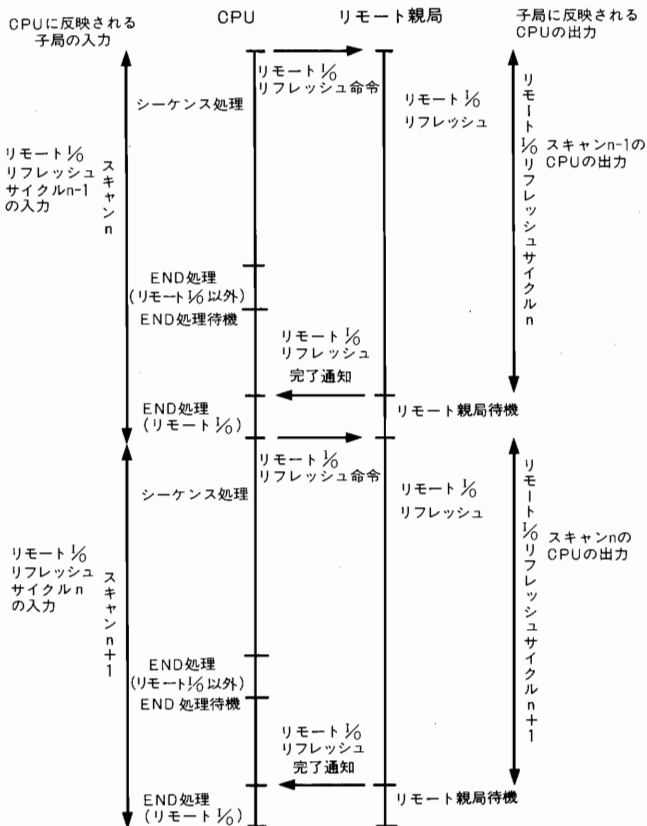


図6. 同期モードの処理の流れ

動作するという形態をとっている(図8)。特に各子局が入力データを送信する際、まず一番大きい子局番号の局が送信し、他の子局は、そのデータの受信確認後、順次送信を行なうアルゴリズムを採用し、データの衝突等を回避して確実な通信を行なえるようにした。それでもなお、何らかの障害が発生した際のために障害からの復帰モードを2種類(ポーズ復帰モード、オート復帰モード)を用意した。例えば、リモートI/Oケーブル断線等の障害が発生した際、ポーズ復帰モードではCPUからの指令があるまでシステムを停止して待機する。また、オート復帰モードでは障害の要因が取り除かれた時、自動的に再スタートする。ポーズ復帰モードは、障害要因の特定が必要な場合や障害によって処理を変えたい場合に設定する。オート復帰モードは、障害復旧時すぐにシステム動作が必要な場合等に設定する。

4.3 高機能命令について

FA-3Sの高機能タイプには、今回新たに、便利で強力な高機能命令(演算命令)が装備された。以下にそのうちのいくつかを紹介する。

(1) 連続四則演算命令

四則演算を連続して実行させる場合、従来の命令

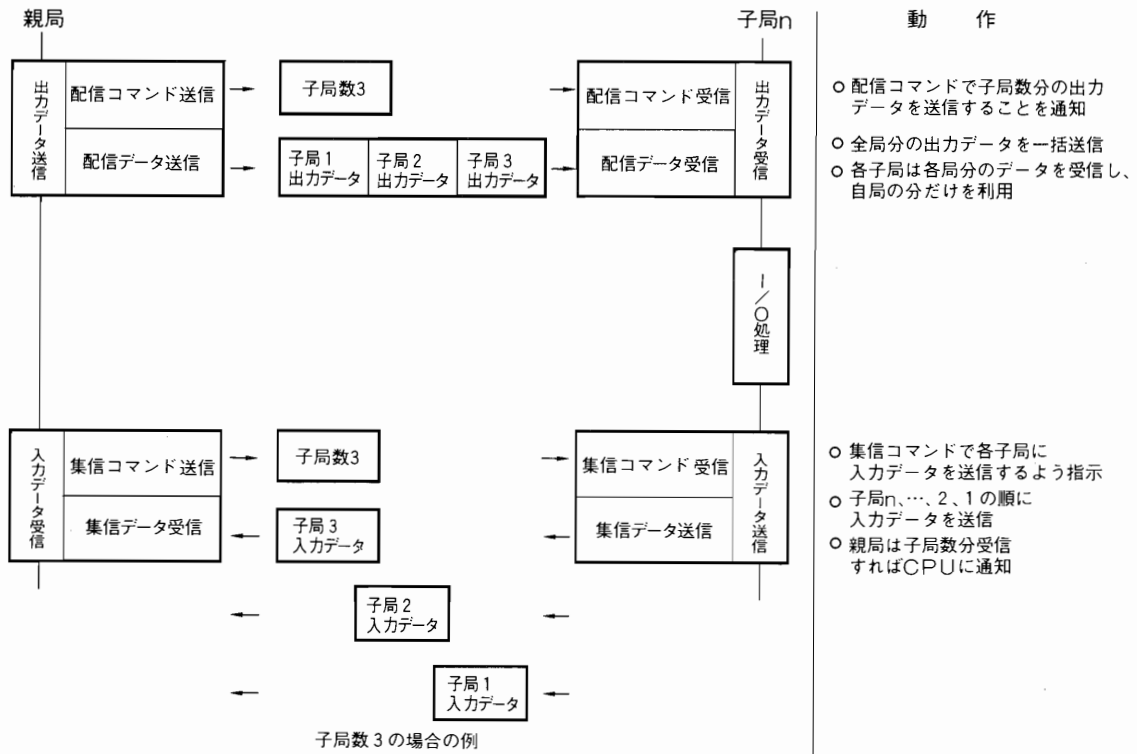


図 8. 親局、子局間の通信形体

では、電卓のような便利な使い方ができなかった。1つはプログラミングツールにおける表現/キー入力の問題、いま1つは演算途中におけるデータのオーバーフロー/アンダーフロー(精度落ち)の問題のため、プログラマーは、PC独自の操作と工夫を余技なくされた。今回の連続四則演算命令は、我々のプログラミングツールである“ラダー入力プログラム”と組み合わせて、以下に述べるような改善がなされた。まず、四則エディタ時の表示/キー入力については、第②項のように、ラダーエディター中に別モードを設け、四則計算式そのままの形で表示/キー入力できるようにした。次に第⑤項で説明しているように、プログラマーが演算途中の数値をあまり意識しなくても四則計算できるように配慮した。

では、第①項の計算式を例にとって概説する。

① 実行したい四則演算の1例

$$A = B - \#1250 + C / D$$

A, B, C, Dは変数。#1250は定数。

② 上の例を“ラダー入力プログラム”でエディットする時の入力フォーム。

A, B, C, DをD50~D53の各データレジスタに割当てると、

$$D50 = D51 - \#1250 + D52 / D53$$

③ FA3Sでは、この式を逆ポーランド記法のルールに従って実行するので、“ラダー入力プログラム”

上の“ラダー→ニーモ変換”時、②の式は、下の意味を持つ内部コードに変換される。

a) ソースオペランド	D51
b) ソースオペランド	#1250
c) 演算子	-
d) ソースオペランド	D52
e) ソースオペランド	D53
f) 演算子	/
g) 演算子	+
h) ディスティネーション	D50

④ FA-3Sでの演算実行の概略

③の意味を持つ内部コードを、逆ポーランド記法にのっとって実行するため、FA-3Sは、LIFO(ラストイン、ファーストアウト)動作をするワークを持つ。それをW₀~W₂とすると、

a) (D51)	→ (W ₀)
b) #1250	→ (W ₁)
c) (W ₀) - (W ₁)	→ (W ₀)
d) (D52)	→ (W ₁)
e) (D53)	→ (W ₂)
f) (W ₁) / (W ₂)	→ (W ₁)
g) (W ₀) + (W ₁)	→ (W ₀)
h) (W ₀)	→ (D ₅₀)

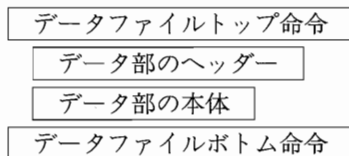
最終結果が、D50にストアされる。

⑤ オーバフロー／アンダーフローについて

この連続四則演算命令は、ソースオペランドとディスティネーションオペランドの各々に対して、データ処理単位(データタイプ)の設定ができるようになっている。データ処理単位としては、ワード、インテジャー、ダブル、ロング等種々のタイプが指定できる。しかし、演算途中の処理単位を、ソースまたはディスティネーションのそれと合わせると、乗除算の実行時、すぐにオーバ／アンダーフローが発生し、必要な精度が確保できない。この問題を解決するため、本命令は演算途中には、フローティングナンバ(浮動小数点)を採用した。その仮数部は32ビット、指数部の基数は処理単位により2または10であり、扱える数の桁範囲は、 $2^{-63} \sim 2^{63}$ (10進数では約 $10^{-19} \sim 10^{19}$)となっている。従ってその範囲内では、オーバフロー／アンダーフローは発生しなくなった。

(2) データファイル命令

PCで実行すべき処理の高度化に伴い、取扱うデータ量が増え、数値データあるいは文字データを定数の配列として自由にアクセスしたいという要求が強くなっている。例えば、「外部表示・印字命令」や「ユーザ定義通信命令」、「ドラムシーケンサ命令」、「アナンシェータ命令におけるユーザ定義のシーケンスパターン」あるいは「連続したnヶのタイマの設定値登録」等々、従来は各演算応用命令の中に、必要な定数をプログラミングしていたところを、定数部だけまとめて配列化し、プログラム部と切り離すことによって、可視性やメンテ性の向上が期待される。データファイル命令は、そのために用意されたもので、下図のような構造になっている。



① データファイルトップ命令

データファイルの先頭を示す命令で、データファイルのファイル番号と長さ(ワード数)を持っている。通常のラダープログラムにて、そのファイル番号を設定することにより、簡単にアクセスできる。

② データ部のヘッダー

データファイルは、複数のレコードから成り、1レコードは何バイトかのデータを含んでいる。このヘッダーには、その1レコード当りのバイト数(1~16)が登録されている。さらに、ローダやラダー入力パ

ログラムで、データファイルを入力する際、キー入力／表示をわかり易くするために、1レコードを複数のフィールドにわけ、フィールド毎にデータの設定／読出ができるので、各フィールドが何バイトのデータで構成されるかの情報もこのヘッダーに含まれる。

③ データ部の本体

「データファイルトップ命令」および、「データ部のヘッダー」で設定された数のデータが、そのままの形(バイナリイメージ)でストアされている。

④ データファイルボトム命令

データファイルの最終を示す命令である。

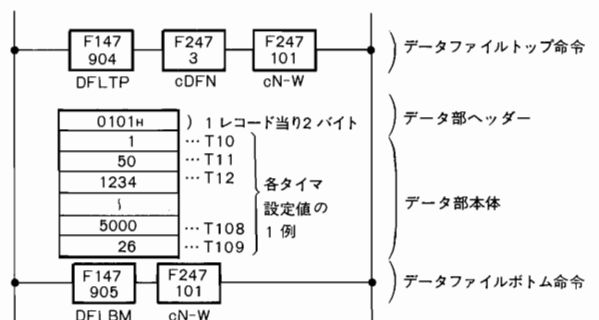
ここで、データファイルの簡単な例として、タイマ10~タイマ109までの100ヶのタイマに設定する設定値を、ファイル番号3に作成し、入力X1の立上り時に、それらを、タイマ設定値に転送するプログラム例を示す。

○ラダープログラム部の設定値転送



この命令にて、ファイル番号3のデータファイルの、第0レコードから100ヶのレコードデータが、T10~T109なる100ヶのタイマの設定値エリアへ転送される。

○データファイル命令(ファイル番号3)



データファイルを使う利点としては次のような点があげられる。

- ・プログラムステップ総数の短縮。
- ・プログラム部とデータ部の分離によるメンテ性、移植性の向上。

例えば、「プログラム部はセットメーカーがメンテするため半固定とし、エンドユーザのアプリケーションバラエティーは、データファイル内のデータ部のみ変更すればよい」というシステムが考えられる。

- ・アプリケーション毎のユーザフレンドリーなCRTエディタ&コードコンバータ。

例えば、アナンシェータのシーケンスパターン登録・読出・修正用のエディターをパソコン上

で作る場合、PCで実行可能なオブジェクトを生成するコードコンバータが必要になるが、その時、ラダーの基本命令／応用命令を含んだコードに変換するよりは、定形的なデータファイル内のコードに変換する方が、はるかにたやすく、開発工程の短縮が計れる。エディタ自身はロータス1-2-3のような汎用ツールを利用することもできる。そういう意味で、データファイルを活用すると、アプリケーション毎のユーザフレンドリーなマンマシンインタフェイスツールが開発しやすくなる。

(3) ユーザ定義通信命令

近年、RS232C等のシリアルポートを持ち、通信にてコントロールできる機器が大変増え、PCの周辺にも多く使われるようになってきている。しかも、各シリアルデバイスの多くは独自のコマンド、プロトコルを持っており、PC側でプログラマブルに対応しなければならない。そこでFA-3Sでは、従来との互換性も考慮し、ローダまたはパソコンと接続していたシリアルポート(信号はオリジナル)のハード的仕様は変えず、ソフト的にも、従来の機能は全てそのままにして、別なモードにて、自由な送信／受信コマンドを定義できる“ユーザ定義通信命令”を設けた。

本命令は、次の個別機能命令より成る。

- ・ユーザ定義通信 許可／禁止
- ・受信フォーマット
- ・受信後照合
- ・送信フォーマット
- ・送信実行

以下に各命令の概略を説明する。

① ユーザ定義通信“許可／禁止”命令

1ヶの入力または内部リレーを設定する。その入力／内部リレーがONの間だけ、ユーザ定義通信モードとなり、OFFすると、従来の標準通信モード(ローダやパソコンと通信可能なモード)にもどる。

② 受信フォーマット

ユーザ定義通信モードに入ると、常時受信可能となるが、この命令でその受信条件の設定を行う。設定項目は、

- a) ポート番号(シリアルポートは0)
- b) 通信ステータスDR (OK/NG)
- c) 受信完了IR
- d) スタートデリミタの最大コード
- e) スタートデリミタの最小コード
- f) エンドデリミタのコード
- g) 受信データを格納するDR

h) 受信データリミット数

となっており、これらの設定を行うとシステムソフトは、受信文字列を1バイトづつチェックして、d)とe)の間に入っている受信コードを最初の受信として識別し、f)の“エンドデリミタ”が来るまで全受信コードをバッファにストアし、f)がくれば、g)のDRへ転送するとともにb)“ステータスDR”、c)“受信完了IR”をセットする。f)がくる前に、受信文字数が、h)の“リミット数”をオーバーすればそこで受信が打ち切られる。

③ 受信後照合命令

受信が正しく行われると、②c)の“受信完了IR”がセットされ、受信データが②g)“受信データを格納するDR”に入るので、この命令を用いて、固定文字列の照合と、変数部の変換と格納を行うことができる。

④ 送信フォーマット命令

送信処理に必要な初期設定を行うための命令で、次の項目を含む。

- a) ポート番号(シリアルポートは0)
- b) 通信ステータスDR(OK/NG)
- c) 送信完了IR

⑤ 送信実行命令

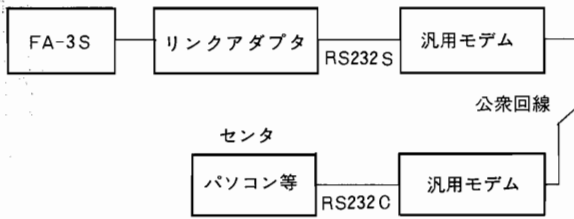
この命令の条件入力の立上りで、命令オペランドで指定した文字列が、送信処理用のワークに作成され、④b)の“通信ステータスDR”に、「送信文字列作成完了」のステータスがセットされる。実際の送信開始は、その後、10msec以内に、システムソフトにより実行され、その送信が完了すると、④c)“送信完了IR”がONになる。

送信文字コードには制限はなく、1メッセージの文字数は、MAX255文字(255バイト)である。

FA-3Sでドライブできるシリアル機器の通信コマンドには、次の制約がある。

- ・1メッセージが255バイト以下
- ・受信メッセージは、いつも一定のエンドターミネータ(たとえば、ODH)で終了するか、または、いつも一定個数の文字列で終了するかのいずれか一方に限る。

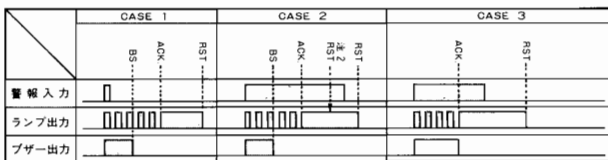
ユーザ定義通信命令を使ってラダープログラムを作ると、FA-3Sからモデムを通してセンタのコンピュータにつなぎ、FA-3Sのタイミングによる発信(ダイヤリングを含む)や、常時着信することができる。そのシステム構成の簡単な概略図を次に示す。



(4) アナシエータ命令

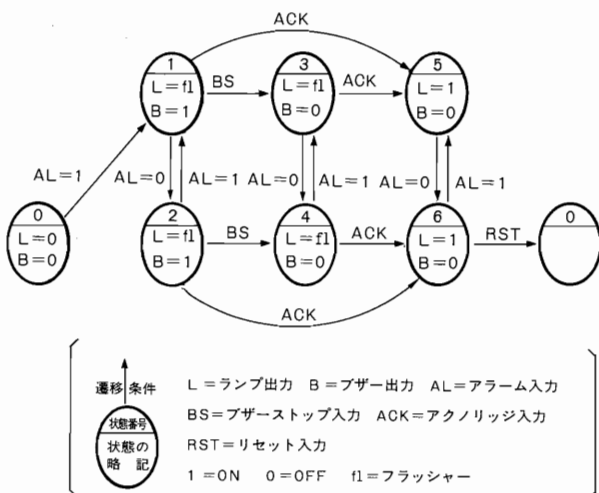
FA-3Sでは、初めて、本格的なアナシエータ命令を採用した。アナシエータのシーケンスパターンには、種々のパターンがあり、あまり多くは網羅できないので、“ANIC-15S形ワンボードアナシエータ”の、パターン番号1～5までの5パターンだけ、標準として装備した。それ以外のパターンは、シーケンスの状態遷移等を一定の約束に従ってコード化し、データファイルを作成することにより可能とした。即ち、ユーザ定義可能な、プログラマブルシーケンスパターンを実現した。以下に、プログラマブルシーケンスパターンの作成について、そのアウトラインを示す。この例は、“ANIC-15S”のパターン番号3（記憶動作 ISA規格では、M-1）の場合であり、このパターンはFA-3Sに標準として入っているが、もしこれをデータファイルに作る場合という前提で説明する。

① “記憶動作”形パターンの動作チャート



(注) 1. BSよりもACKが優先されます。また、ACKとRSTの間にはインターロックあり。
2. 警報入力が残っている状態でRSTを行っても状態の変化は起こりません。

② 動作チャートを基にして状態遷移図を作る。(上記チャートに表現されていない状態遷移についても考慮する必要がある。)



遷移条件 L=ランプ出力 B=ブザー出力 AL=アラーム入力
BS=ブザーストップ入力 ACK=アクトリッジ入力
RST=リセット入力
状態の略記 1=ON 0=OFF f1=フラッシャー

③ ②の状態遷移図に従って、全状態遷移を表すテーブルを作成する。②の図では、1ヶの状態遷移は、1ヶの→で表現されており、この例では、計14ヶの遷移がある。1ヶの遷移について必要な情報は下記の3項目となる。

- a) 遷移前の状態番号……(ex) 2
- b) 遷移条件 ……(ex) AL=1
- c) 遷移後の状態番号……(ex) 1

この3項目の情報を1組とし、全ての遷移について、全14組の遷移テーブルを作る。

④ ②の状態遷移図に従って、各状態番号毎に、出力信号の動作を定義するためのテーブルを作成する。

例えば、L即ちランプ出力については、

状態番号	1	……f1	(フラッシャー)
"	2	……f1	(")
"	3	……f1	(")
"	4	……f1	(")
"	5	……1	(点灯)
"	6	……1	(点灯)

のような動作テーブルが定義される。

⑤ ③、④で作成したテーブルを、コード化し、遷移や状態の総数等、いくつかの必要な設定値を付け加えて、1ヶのデータファイルを作成する。

⑥ ⑤のデータファイルのファイル番号を、アナシエータ命令の中に設定する。

以上の手順により、かなり短い工数で、任意のシーケンスパターンが設計でき、従来のリレーロジックを組み合わせて実現するよりも、設計/検査/メンテ時の大いなる“省”につながるものと期待している。

以上、いくつかの高機能命令について説明したが、これらの命令は全てソフト的に処理しており、その実行時間には不満が残る。複雑な高機能命令は、どうしても、処理速度が遅くなるが、一方、アプリケーションとしては高速化要求がますます高まりつつあり、高機能命令のスピードアップが大きな課題である。

4.4 ラダー図入力用ソフトウェア

汎用パソコンによるラダー図作成用CADソフトウェアとして、従来のFAシリーズPC対応であるラダー入力プログラムを、今回FA-3Sの開発にともない、次の事項を考慮して、機能アップを行った。

- 操作性の向上
 - ポップアップメニューなど。
- ソフトウェア資産の継承
 - 上位互換性あり。
- FAシリーズ全機種に対応

MICROシリーズにも対応。

○プログラム機能の強化

修正部分の強調表示, マップ表示, undo機能など。

○デバック機能の強化

集合モニタ, フォース操作, 画面分割表示など。

この章では, 機能アップにおいて, ソフトウェア上,

工夫した点を, いくつか紹介する。

なお, 従来のソフトを旧版, 今回のソフトを新版と呼ぶことにする。

4.4.1 ポップアップメニュー

本プログラムは, メインメニューからの階層メニュー構造である。そのため, 通常操作では, 一度メインメニューに戻らないと, 次の処理項目を選択できない場合がある。しかし, ポップアップメニューを使用すると, 一回の操作で, 次の処理項目を選択することが可能である。以下, ポップアップメニューについて, 簡単に説明する。

ポップアップメニューの主な特徴(機能)を, 次に示す。

○自社開発したウィンド使用

○常時, 呼出し可能

○操作が容易

○一回の操作で次の処理項目を選択

○動作状態表示

○子プロセス起動可能

○連続呼出し設定機能

〔最大4項目まで, 連続して行う処理項目を登録できる。ひとつの処理項目終了後, 自動的に, 次の処理項目を実行する。〕

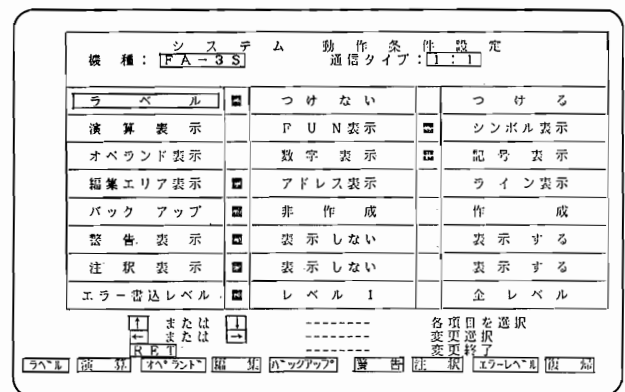
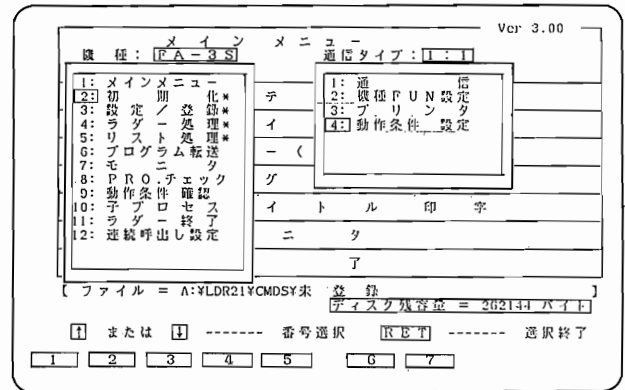
図9は, 処理項目選択例である。この例において, 内部では, メインメニューからのパス情報を設定する。(図10参照) そのパス情報を追っていくことにより, 目的の処理項目へ到達することが可能である。

図11は, 連続呼出し設定例である。この例における処理の流れは, 図12のとおりである。制御情報には, 実行順に処理項目を設定する。ひとつの処理項目終了後は, 制御情報から終了した処理項目を取除く。そして, 次の処理項目を認識し, 自動的に実行する。最後の処理項目終了後は, 制御情報を未登録とする。以後は, 通常メニュー選択方式に戻る。

4.4.2 バックアップファイル

ファイル更新時に, バックアップファイル作成機能を追加した。旧版では, バックアップファイルを作成しないので, キー操作を誤った場合, 元に復帰することができない。今回の新版では, 元に復帰することが可能である。また, ディスクを有効に使用するため, この機能を使用するか使用しないか, 選択することができる。

図13にバックアップファイル名と, そのファイル更新



キー操作

- ① SHIFT + f.10 : ポップアップメニュー呼出し
- ② 2 : 初期化選択 (サブメニュー) オープン
- ③ 4 : システム動作条件設定メニューへ

図9. ポップアップメニューによる処理項目選択例

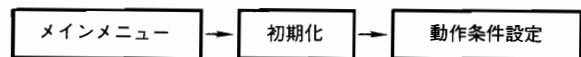


図10. 図9の例におけるパス情報

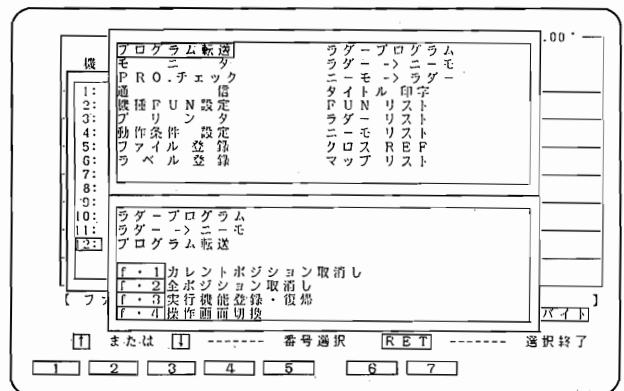


図11. 連続呼出し設定例

処理対象モードを示す。バックアップファイルを復帰する場合は、ファイル名を変更する。

図14にラダーファイルの更新処理手順を示す。次に、各手順について、簡単に説明する。

- ① プログラミング中は、ワークファイルに、その内容を保持する。
(現在のラダーファイルは、更新されない。)
- ② 古いバックアップファイルを削除する。
(バックアップファイルは、常に、直前のファイルである。)
- ③ 現在のラダーファイルをバックアップファイルに変更する。
(ファイル名を変更する。)
- ④ ワークファイルより、新たに、ラダーファイルを作成する。
(ラダーファイルの更新。)
- ⑤ ワークファイルを削除する。
(ワークファイルは、一時的に使用する。)

4.4.3 ワークファイル

プログラム編集用のワーク領域として、ワークファイルを導入する。

旧版では、プログラム編集は、すべて、メモリ上で行う。そのため、初めに、メモリ領域を確保する必要がある。この場合、確保した領域より大きなプログラムの編集は、不可能である。(図15①参照)

新版では、プログラム編集のデータは、すべて、ワークファイルに保持する。そして、メモリは、編集部分のデータのみを持つ。これにより、メモリ領域は、常に、一定の領域を確保できればよいことになる。(図15②参照)

ワークファイルの導入により、最大プログラム容量は、メモリ依存からディスク依存へと変る。したがって、ディスク容量が大きくなれば、これまでより、大きなプログラムが編集可能である。

4.4.4 undo機能

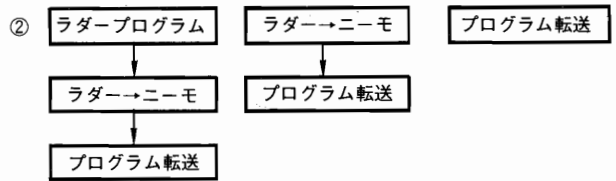
undo機能は、誤操作によりデータを削除した場合などに、元の状態へ復帰する機能である。ここでは、undo機能について説明する。説明上必要となるので、まず、ワークファイルの構造について簡単に述べる。

ワークファイルは、ある一定のデータをまとめてレコードとし、複数のレコードからなる。各レコードは、制御情報として、リンク情報をもつ。リンク情報は、レコード間のつながりを示す。

それでは、undo機能について、図16を用いて説明する。図16は、編集後、さらに、undo実行後のワークファイルの状態を、概念的に示したものである。

- ① 編集部分を新しく確保し、リンク情報をつけかえる。元の部分は、オリジナル部分に残っているが、

① ラダープログラム ⇔ ラダー→ニーモ ⇔ プログラム転送



①実行中の処理項目
②制御情報

図12. 図11の例における処理項目と制御情報の流れ

ファイル名および拡張子	バックアップファイルの拡張子	バックアップ作成対象モード
ラダーファイル(.LDR)	.LD	プログラミング ニーモニック→ラダー変換
コードファイル(.COD)	.CD	ラダー→ニーモニック変換 プログラム転送
ラベル登録ファイル(.LAB)	.LB	プログラミング、ラベル登録
マップ情報ファイル(.LMP)	.LP	プログラミング

図13. バックアップファイル名および対象モード

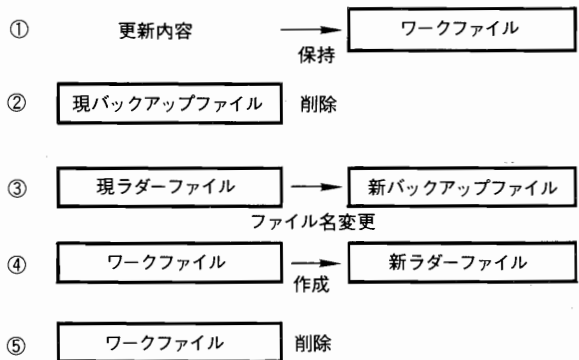
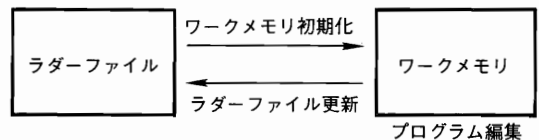
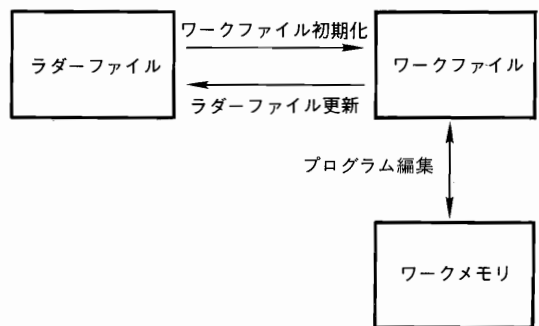


図14. ファイル更新処理手順 (ラダーファイルの例)



①旧版



②新版

図15. プログラム編集時のワーク領域

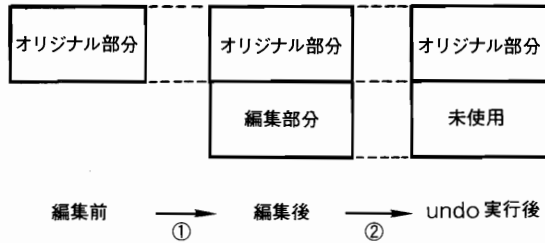


図16. undo実行時のワークファイル状態遷移

リンクからは、はずれている。

- ② リンク情報を元に戻すことにより復帰する。編集部分は未使用となり、次の編集時に再使用される。したがって、undo機能は、直前の編集操作についてのみ、有効となる。

4.5 FAプログラムローダ

FAプログラムローダは、数字方式と記号方式の2種類の表示モードが有り、それぞれ独立したプログラミング体系を採る。

数字方式表示モードは、図17のように従来のFA-2シリーズプログラムローダの機能を装備したモードであり、プログラミング等のオペランド番号を物理的番号として処理する。

記号方式表示モードは、FA-3Sシリーズから導入されたもので、オペランド番号を記号化し、論理的番号として処理する。

図18は、記号方式表示モードの表示例を示したものである。

各モードの切り替えは、容易であり、ファンクション(FUN90)設定のソフトスイッチで切り替える。

数字方式、記号方式の両モードを有することは、ユーザーソフト資産の有効利用を目的としたものである。

また、FAプログラムローダは、FA-3Sシリーズ対応は元より、FAシリーズ全機種に対応し、FAシリーズの周辺ツールの共通化が成されている。

FAプログラムローダのハードは、ROM60Kバイト、RAM34Kバイトの容量を持ち、CPU-8085と周辺LSI-8155を使用して下記の項目を実現している。

1. インターバル・タイマー
2. LCD表示コントロール
3. キースキャン処理
4. PROMライター
5. シリアル通信
6. メモリバンク切り換え
7. ブザー出力
8. プログラムスイッチ

CPU-8085のメモリ管理は、64Kバイトであるためメモリバンク切り換えを採用し、システムプログラム容量60

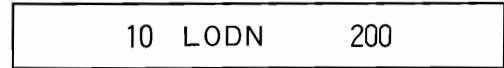


図 17. ローダ表示例 (数字方式)

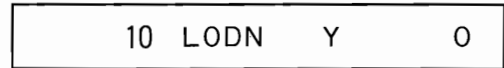


図 18. ローダ表示例 (記号方式)

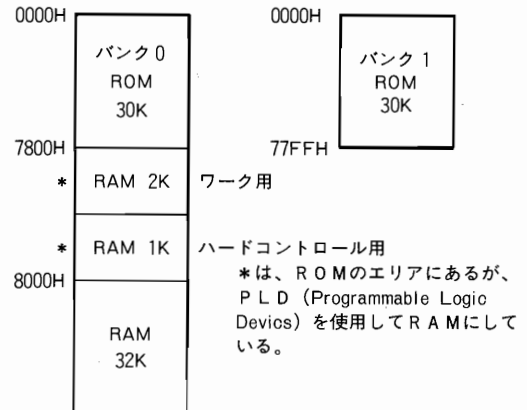


図 19. ローダのメモリマップ

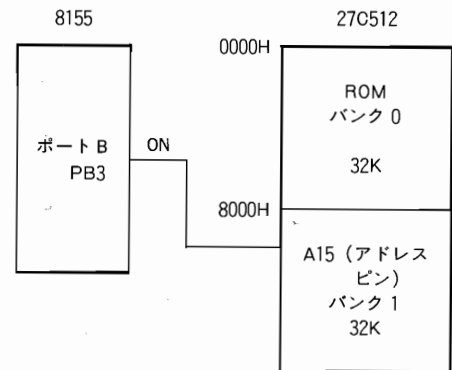


図 20. バンク切り換え略図

Kバイト (EPROM 27C512) をそれぞれ30Kバイトに分割し、バンク0 (0000H~77FFH)、バンク1 (8000H~F7FFH)として、ソフトにて切り換えを行っている。(図19と図20を参照)

切り換えの方法は、8155のBポートを出力とし、PB3をONするとROMのアドレス8000番地となり、バンク1を選択することが可能となる。(CPU側から見た場合、バンク1のアドレス範囲は、0000H~77FFHである。)

このバンク切り換え方式を使用すると2つの独立した30Kのシステムプログラムが存在することができる。

FAプログラムローダは、45Kバイト以上のシステムプログラムなので、バンク0の追加プログラムがバンク1に納められている。

5. 今後の展開, 動向

総合化システムを実現するために、ネットワークの標準化・統一化が進められ、現場での使用例も出て来ている。しかし、まだ高価なものである。当社では、小規模システムにも導入できる低価格で容易なネットワークを開発中である。自由なプロトコルでの通信をサポートし、ネットワーク資源の共有化を図っている。FA-3Sは前述したユーザ定義通信モードや標準装備の通信モードで、このネットを利用することにより、複数の周辺機器と接続することが可能になる。

プログラミングツールや表示装置などのマンマシンインターフェイスに目を向けてみると、制御用途の拡がりに伴いオペレータが処理する情報は多様化し高度になる。そのため、より使い易さを追求したオペレータI/F機能が今後必要となる。

PCの用途が拡大していくと、ユーザの専用システム化、

高機能化への要求に対して、メーカーが供給する機能だけでは充分に応えられなくなるだろう。これに応える手段として、PCの機能(システムソフト)をユーザに開放することが考えられる。PCの機能をユーザが作成することにより、ユーザ独自の機能、命令をもったコントローラが実現できる。

6. おわりに

以上述べた様に制御ニーズにフレキシブルに対応する機能を備えたFA-3Sを開発した。装置や機能がいかに自動化、省力化されてもこれらを使用するのは人間であるという考えにたって、ヒューマンインタフェイスが重要視されている。高機能化に伴い処理が複雑化するPCにおいても重要な課題と考え現場に密着した御意見御要望を取り入れた製品を開発していく所存である。