

Panasonic[®]

プログラマブルコントローラ
MEWNET FP3
FP-BASIC編集ソフト
Ver. 2
ユーザーズマニュアル

MEWNET FP3 FP-BASIC編集ソフト ユーザーズマニュアル
FAF-44③ '93・7月

松下電工

安全に関するご注意

ケガや事故防止のため、以下のことを必ずお守りください。

据付、運転、保守、点検の前に、必ずこのマニュアルをお読みいただき、正しくご使用下さい。
機器の知識、安全の情報、その他注意事項のすべてを習熟してからご使用下さい。

このマニュアルでは、安全注意事項のレベルを「警告」と「注意」に区分しています。



警告

取扱いを誤った場合に、使用者が死亡または重傷を負う危険の状態が生じることが想定される場合

本製品の故障や外部要因による異常が発生しても、システム全体が安全側に働くように本製品の外部で安全対策を行ってください。

可燃性ガスの雰囲気では使用しないでください。

爆発の原因となります。

本製品を火中に投棄しないでください。

電池や電子部品などが破裂する原因となります。



注意

取扱いを誤った場合に、使用者が傷害を負うかまたは物的損害のみが発生する危険の状態が生じることが想定される場合

異常発熱や発煙を防止するため、本製品の保証特性・性能の数値に対し余裕をもたせて使用してください。
分解、改造はしないでください。

異常発熱や発煙の原因となります。

通電中は端子に触れないでください。

感電のおそれがあります。

非常停止、インターロック回路は外部で構成してください。

電線やコネクタは確実に接続してください。

接続不十分な場合は、異常発熱や発煙の原因となります。

製品内部に液体、可燃物、金属などの異物を入れないでください。

異常発熱や発煙の原因となります。

電源を入れた状態では施工(接続、取り外しなど)しないでください。

感電のおそれがあります。

著作権および商標に関する記述

このマニュアルの著作権は、松下電工株式会社が所有しています。

本書からの無断複製は、かたくお断りします。

Windows および WindowsNT は米国 Microsoft Corporation の米国およびその他の国における登録商標です。

その他の会社および製品名は、各社の商標または登録商標です。

商品改良のため、仕様、外観およびマニュアルの内容を予告なく変更することがありますので、ご了承ください。

／は／じ／め／に／

このたびは、FP-BASICをご購入いただきまして、まことにありがとうございます。

FP-BASICは、市販パソコン（NEC・9801シリーズおよびEPSON・PC286/386シリーズ）を使用して、弊社プログラマブルコントローラFP3(H)-BASICタイプCPUのプログラミングを行うための専用ソフトウェアです。①BASICライクな親しみやすい操作性と構造化による効率良いプログラミング、②多彩な高機能ユニット専用制御命令のサポート、③高度なモニタ機能の装備により、FP3(H)-BASICタイプの高度な機能を十二分に活用することができます。

この『FP-BASIC ユーザーズマニュアル』は、FP-BASICを使用して初めてFP3(H)-BASICタイプのプログラミングをされる方のために、ソフトウェアの導入方法と基本的な操作方法、およびFP-BASICの主な機能について説明しています。

なお、ご使用にあたっては、このマニュアルの他に、以下のマニュアルを参照してください。

- プログラマブルコントローラについて
『FP3(H)-BASICタイプ ハード導入マニュアル』
- プログラミングについて
『FP-BASIC リファレンスマニュアル』

制限事項

- (1) FP-BASICのすべての機能が使用できるのは、FP3H-BASICタイプおよびVer.2.0以降のFP3-BASICタイプを使用した場合に限定されます。
- (2) Ver.2.0以前のFP3-BASICタイプを使用する場合、使用できるコマンド、命令、関数が制限されます。このマニュアル中の「使用するコマンド・命令・関数」の記述の中で、◆が付いているものは使用できません。
- (3) FP-BASICには、MS-DOSのシステムは含まれていません。別途「NEC日本語MS-DOS」(Ver.3.10以降)を用意してください。

目次

はじめに
制限事項
目次
このソフトをご利用いただく前に
梱包品一覧
保証について
お問い合わせについて
関連マニュアル
このマニュアルの構成
このマニュアルの記述上の注意

第1章 特長とシステム構成

1-1. FP - BASICのシステム構成と特長	12
1-1-1. システム構成	12
1-1-2. 特長	13
1-1-3. 動作環境	14
1-2. FP - BASICの機能概要	16
1-2-1. プログラミング機能	16
1-2-2. モニタ機能	17
1-2-3. マルチタスクプログラミング	18

第2章 準備

2-1. ハードウェアの準備	20
2-1-1. 機器の接続方法	20
2-1-2. FP3(H)-BASICタイプCPUの設定	21
2-1-3. パソコンの設定	21
2-2. ソフトウェアの準備 (インストール)	22
2-2-1. インストールの概要	22
2-2-2. MS-DOSの知識のある方のために	23
2-2-3. フロッピーディスクへのインストール (起動ディスクの作成)	24
2-2-4. ノートパソコンへのインストール (起動ディスクの作成)	28
2-2-5. ハードディスクへのインストール	33
2-2-6. フロッピーディスクの初期化	36
2-2-7. フロッピーディスクの複写 (バックアップ)	38
2-2-8. ワープのかな漢字変換の組み込み	40

第3章 FP - BASICのプログラミング

3-1. FP - BASICの起動と終了	46
3-1-1. 起動画面	46
3-1-2. FP - BASICの終了方法	46
3-1-3. キー操作の概要	47
3-1-4. ファンクションキーへのコマンド割り付け	48
3-2. CPUの初期設定	50
3-2-1. CPUのバージョン表示	50
3-2-2. CPUのメモリの初期化	50
3-2-3. パラメータメモリの設定	51
3-2-4. 変数領域の割り付け	53
3-2-5. スロット割り付け	54
3-3. プログラム編集の基本操作	56
3-3-1. プログラム入力と編集の基本	56
3-3-2. 行の移動	58
3-3-3. プログラム中の文字列検索	59
3-4. ファイルのセーブとロード	60
3-4-1. プログラムのセーブ (保存)	60
3-4-2. プログラムのロード (読み出し)	60
3-4-3. ファイル管理	61
3-4-4. ファイル選択ウィンドウの使用法	62
3-5. オブジェクトプログラムの作成と転送	64
3-5-1. コンパイル	64
3-5-2. ダウンロード	65
3-5-3. アップロード	65
3-6. プログラムの実行	66
3-6-1. プログラムの実行	66
3-6-2. トレースモード	66
3-6-3. テストモード実行	67
3-7. 実行中のプログラムのモニタ	68
3-7-1. CPUの動作モードの表示	68
3-7-2. タスクごとのステータス表示	68
3-7-3. タスクごとのステータスのリアルタイム表示	69
3-7-4. I/O、メモリおよび変数のモニタ	70
3-7-5. I/Oおよびメモリのマルチポイントモニタ	71
3-7-6. I/Oおよびメモリのダンプ (状態表示)	72
3-7-7. タイマ経過値の表示および設定	73
3-7-8. 高機能ユニットの割り込み設定の表示	73
3-7-9. I/Oおよびメモリその他の状態の 一括ダンプ (状態表示) と編集	74
3-7-10. 強制入出力	76
3-7-11. I/O、メモリ、変数のセーブとロード	76

第4章 FP - BASICの基礎知識

4-1. BASICプログラムの基本ルール	78	4-7. 一般命令と組み込み関数	106
4-1-1. 直接実行と間接実行	78	4-7-1. N88BASIC一般命令	106
4-1-2. プログラムの例	78	4-7-2. ディスクファイルアクセス命令	106
4-1-3. 命令文	78	4-7-3. 変数操作命令	106
4-1-4. コメント(注釈文)	79	4-7-4. I/Oポートアクセス命令	107
4-1-5. 行と行番号	79	4-7-5. タイマ命令と関数	107
4-1-6. ラベルとラベル名	79	4-7-6. システム変数とシステム関数	107
4-7-7. 組み込み関数	108		
4-2. 使用できる文字と記号	80	4-8. リモートI/Oシステムのスロット割り付け命令	108
4-2-1. 使用できる文字と記号	80	4-9. MEWNET データ転送命令	112
4-2-2. FP - BASICの予約語	81	4-10. 高機能ユニット専用命令	114
4-3. 定数	82	4-10-1. 通常スロットの高機能ユニット入出力命令	114
4-3-1. 定数の種類	82	4-10-2. リモートI/Oシステムの高機能ユニット入出力命令	114
4-3-2. 数値定数	83	4-10-3. D/A変換ユニット専用制御命令	115
4-3-3. 文字定数	83	4-10-4. A/D変換ユニット専用制御命令	115
4-4. 変数	84	4-10-5. シリアルデータユニット専用制御命令	116
4-4-1. 変数名	84	4-10-6. パルス出力ユニット専用制御命令	116
4-4-2. 変数の型と値	84	4-10-7. 高速カウンタユニット専用制御命令	117
4-4-3. グローバルな変数とローカルな変数	96	4-10-8. 位置決めユニット専用制御命令	117
4-4-4. 配列	97		
4-4-5. 型変換	98		
4-5. 式と演算子	89		
4-5-1. 式の種類	89		
4-5-2. 演算子の優先順位	89		
4-6. FP - BASICの制御構文	90		
4-6-1. マルチタスクとFUNCTION文	90		
4-6-2. WAIT命令によるインターロック	92		
4-6-3. 分岐命令 GOTO	94		
4-6-4. サブルーチン呼び出し命令 GOSUB	95		
4-6-5. ON GOTO/GOSUB文	96		
4-6-6. SELECT ~ CASE文	97		
4-6-7. IF/IFB文	98		
4-6-8. ループ文	100		
4-6-9. CALL ~ FUNCTION文	102		
4-6-10. 割り込み制御	103		

第5章 分割コンパイル

5-1. 分割コンパイル	120
5-1-1. ソースプログラムの作成	120
5-1-2. 分割コンパイル	121
5-1-3. オブジェクトリンク	122

第6章 付録

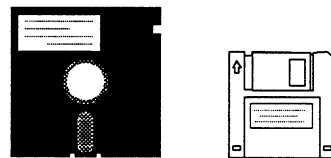
6-1. I/O・メモリー	124
6-1-1. I/O・メモリー構成	124
6-1-2. 特殊メモリー	125
6-1-3. 特殊データメモリー	131
6-2. パラメータメモリー設定	138
6-3. エラーコード	140
6-3-1. 自己診断エラーコード	140
6-3-2. FP-BASICエラーコード	141
6-4. スロット割り付けの基礎知識	146
6-4-1. フリーロケーション機能を使用するスロット割り付け	146
6-4-2. スロットコマンド機能を使用するスロット割り付け	147
6-4-3. ユニットのI/O点数と機能番号	148
6-5. プログラム実行時間	150
6-6. ASCII/JISキャラクタコード	152
6-7. 索引	154

このソフトウェアをご利用いただく前に

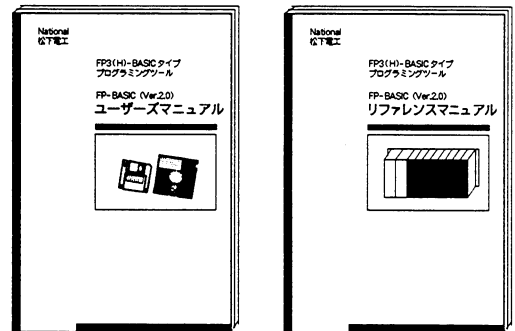
■梱包品一覧

- FP-BASICシステムディスク

5.25インチフロッピーディスクと3.5インチフロッピーディスクが同梱されています。このマニュアルでは、「オリジナルディスク」と呼びます。なお、MS-DOSのシステムは含まれていませんので、別途「NEC日本語MS-DOS」(Ver.3.10以降)を用意してください。



- ユーザーズマニュアル
- リファレンスマニュアル



■保証について

- 保証期間

この製品についての保証期間は、納入後1ヵ年とします。

- 保証範囲

保証期間中に納入者側の責任により故障が生じた場合は、その機器の故障部分の交換、または修理を納入者側により無償で行います。

ただし、次に該当する場合は、この保証の対象範囲から除外させていただきます。

- (1) マニュアルと異なった取り扱い、ならびに使用による場合。
- (2) 故障の原因が納入品以外の事由による場合。
- (3) 納入者以外の改造、または修理による場合。
- (4) その他、天災、災害などで納入者側の責任ではない場合。

なおここでいう保証は、納入品単体の保証を意味するもので、納入品の故障により誘発される損害はご容赦いただきます。

■お問い合わせについて

FP - BASICについてのお問い合わせは、ご購入いただきました販売代理店または、下記営業所までご連絡ください。

販売代理店

松下制御機器株式会社

本社	☎ 03-3454-6181	中部	
東部		名古屋機器営業所	☎ 052-581-8863
東北営業所	☎ 022-223-8163	名古屋FA営業所	☎ 052-581-8864
北関東営業所	☎ 0273-63-2033	名古屋北FA営業所	☎ 0568-33-5111
関東営業所	☎ 048-665-7991	豊田FA営業所	☎ 0565-35-2181
宇都宮FA営業所	☎ 0286-34-0161	三重FA営業所	☎ 0593-53-9041
茨城FA営業所	☎ 0292-43-8868	静岡機器営業所	☎ 054-261-7711
長野営業所	☎ 0263-28-0790	静岡FA営業所	☎ 054-261-7711
西東京機器営業所	☎ 0425-28-2241	浜松FA営業所	☎ 053-442-0531
東京機器営業所	☎ 03-3454-6186	北陸営業所	☎ 0762-37-3663
東部車載営業所	☎ 03-3454-6065	中部FAセンター	☎ 052-581-8861
東部システム営業所	☎ 03-3454-6190	西部	
東京FA営業所	☎ 03-3454-6188	京滋FA営業所	☎ 075-681-0237
千葉FA営業所	☎ 0471-63-6161	京滋機器営業所	☎ 0775-64-9371
横浜機器営業所	☎ 045-321-1235	松下グループ営業所	☎ 06-943-1082
横浜FA営業所	☎ 045-321-1621	大阪機器営業所	☎ 06-943-1082
神奈川西営業所	☎ 0462-28-9533	西部システム営業所	☎ 06-354-0531
東部FAセンター	☎ 03-3454-6190	大阪FA営業所	☎ 06-943-9680
		堺FA営業所	☎ 0722-70-2631
		兵庫営業所	☎ 078-735-8601
		岡山営業所	☎ 0862-45-3701
		高松販売課	☎ 0878-41-3231
		広島営業所	☎ 082-247-9084
		九州営業所	☎ 092-522-5545
		西部FAセンター	☎ 06-354-0531

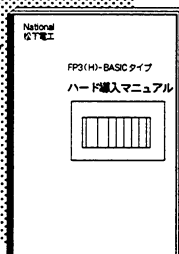
お問い合わせの際には、以下の点をご確認の上ご連絡くださいますよう、お願いいたします。

1. お客様の氏名、住所、郵便番号、電話番号
2. 製品のロット番号（登録番号）
3. 周辺機器の構成

関連マニュアル

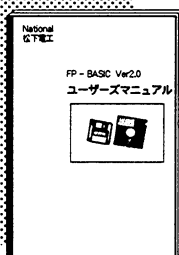
/各/マ/ニ/ュ/ア/ル/の/内/容/

FP3(H)-BASICタイプ ハード導入 マニュアル



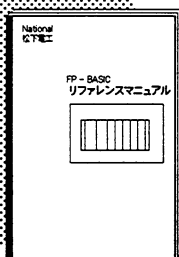
FP3(H)-BASICタイプの特長、システム概要、機能説明、仕様、設置・配線、運転、および保守について説明しています。このマニュアルの中では、『ハード導入マニュアル』と記述します。

FP-BASIC ユーザズ マニュアル



このマニュアルです。

FP-BASIC リファレンス マニュアル



FP-BASICのソフトウェア仕様、およびコマンド、命令、関数の詳細について説明しています。このマニュアルの中では、『リファレンスマニュアル』と記述します。

このマニュアルの構成

/各/章/の/内/容/

特長とシステム構成

第 1 章

FP-BASICを使用してBASICタイプCPUを使用する際のシステム構成の概要、およびFP-BASICによるプログラミングの特長について、説明します。

準 備

第 2 章

FP-BASICを使用する際の、ハードウェアおよびソフトウェアの準備について、説明します。とくに、初めてFP-BASICを使用する場合は、この章をお読みにになり、正しくインストールしていただきますようお願い致します。

FP-BASIC の プログラミング

第 3 章

FP-BASICの基本的な操作方法を説明します。ダイレクトコマンドの使い方の説明を中心に、CPUの環境設定、プログラムの作成方法、デバッグの方法などについて、概説します。とくに、BASICのプログラミングを初めて経験する場合は、必ずこの章の内容を十分にお読みください。

FP-BASIC の 基礎知識

第 4 章

FP-BASICの文法の概要を説明します。構文、変数、演算子、命令語、関数などの使い方について、概説します。とくに、BASICのプログラミングを初めて経験する場合は、必ずこの章の内容を十分にお読みください。

分割コンパイル

第 5 章

FP-BASICの分割コンパイルについて説明します。分割コンパイルは、比較的大きなプログラムの作成とメンテナンスに有効な手法です。

付 録

第 6 章

I/O・メモリー一覧、パラメータメモリー設定一覧、エラーコード一覧、スロット割り付けの基礎知識、プログラム実行時間、ASCII/JISキャラクタコード表、BIN/BCDコード対応表、索引

このマニュアルの記述上の注意

- 本文中で「FP3(H)-BASICタイプ」「BASICタイプCPU」と記述している場合、または単に「CPU」と記述している場合、特に断りのない限り、弊社プログラマブルコントローラFP3H-BASICタイプおよびFP3-BASICタイプを意味します。
- 本文中で「MS-DOS」と記述している場合は、特に断りのない限り、「NEC日本語MS-DOS」(Ver.3.10以降)を意味します。
- 本文中で「かな漢字変換」と記述している場合、MS-DOSの日本語入力機能、または日本語入力のためのシステム(デバイスドライバ)を意味します。FP-BASICを標準的なインストール方法で使用する場合、「NEC日本語MS-DOS」の単文節変換システム(NECDIC.DRV)が組み込まれます。なお、かな漢字変換の操作方法については、MS-DOS付属のマニュアルをご覧ください。
- 本文中の「使用するコマンド・命令・関数」の記述の中で、◆印がついているコマンド・命令・関数を使用できるのは、FP3H-BASICタイプおよびVer.2.0以降のFP3-BASICタイプに限定されます。Ver.2.0未満のFP3-BASICタイプでは、◆印つきのコマンド・命令・関数は使用できません。

例

- 使用するコマンドと関数

◆MAP	タスクごとにローカル変数領域の割り付けをします
◆FRE ()	タスクごとのローカル変数領域の空きエリアサイズを返します(16バイト単位)
◆FRE V	グローバル変数の空きエリアサイズを返します。(kバイト単位)

* FREはプログラム中でステートメントとしても記述できます。

- 本文中で「コマンド」と記述している場合、特に断らない限り、ベーシックのキーコマンドを意味します。ただし、コマンドの中には、プログラム中にステートメントとして記述することができるものもあります。詳しくは、『リファレンスマニュアル』をご覧ください。

第 1 章

特長とシステム構成

こ の 章 の 内 容

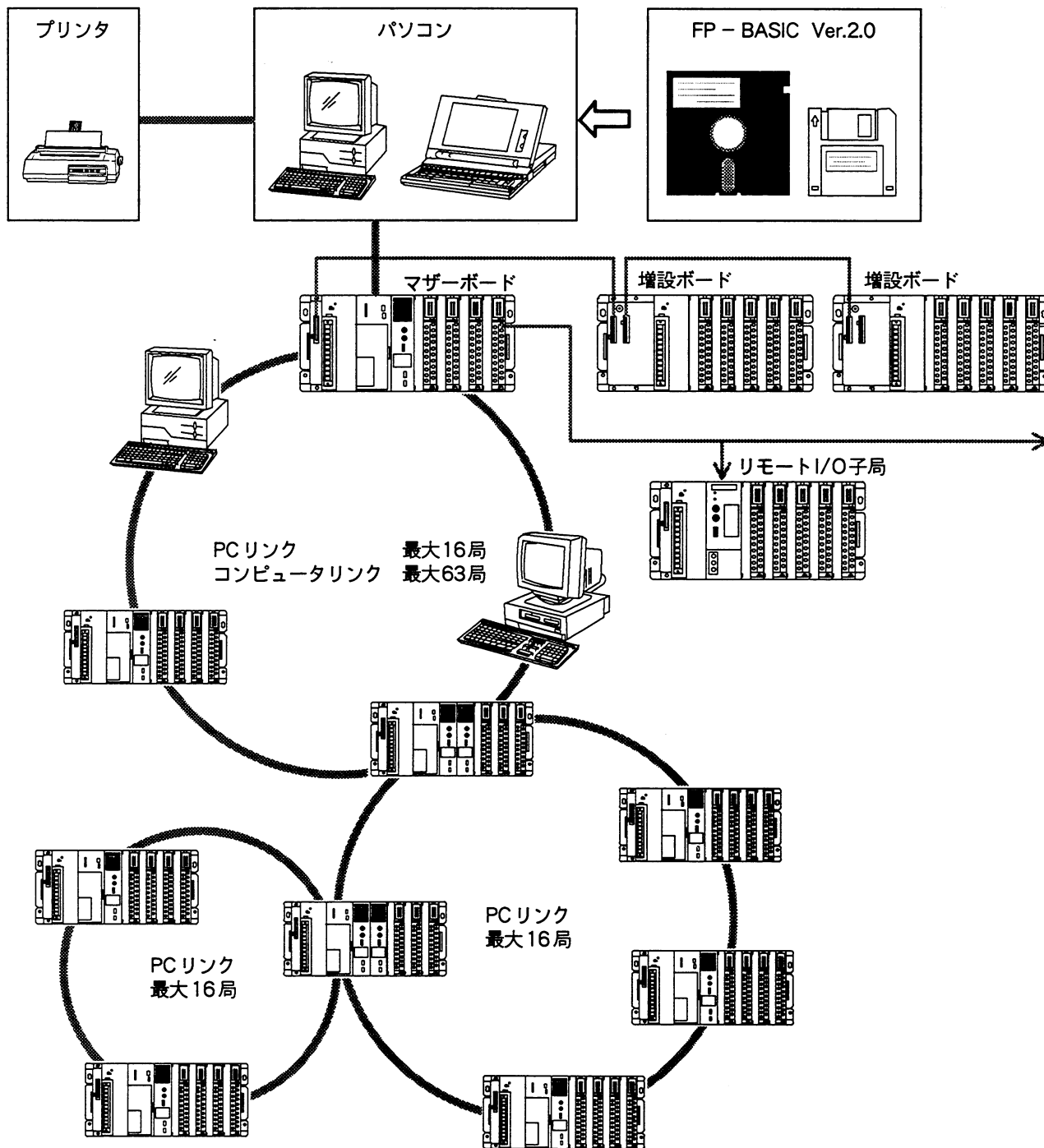
1-1. FP - BASIC のシステム構成と特長	12
1-1-1. システム構成	12
1-1-2. 特長	13
1-1-3. 動作環境	14
1-2. FP - BASIC の機能概要	16
1-2-1. プログラミング機能	16
1-2-2. モニタ機能	17
1-2-3. マルチタスクプログラミング	18

FP - BASIC のシステム構成と特長

1-1-1 システム構成

FP - BASICは、市販パソコンを使用して、弊社プログラマブルコントローラFP3(H)-BASICタイプのプログラミングをするための専用ソフトウェアです。

使用できるパソコンは、NEC・9801シリーズおよびEPSON・PC286/386シリーズです。



注意

BASICタイプCPUについての詳細は、『ハード導入マニュアル』をお読みください。

1-1-2 特長

① BASIC ライクな親しみやすい操作性

PC専用のプログラミング言語を学習する必要がなく、親しみやすいBASIC言語でプログラマブルコントローラ（PC）のプログラミングができます。したがって、迅速かつ低コストに、PCが導入できます。

② 能率的なフルスクリーンエディタ機能

フルスクリーンエディタの採用と充実したエディットコマンド群が、能率的なプログラミング環境を提供します。また、作成したプログラムのディスクへのセーブや、ASCIIファイル（MS-DOSテキストファイル）の読み込みも可能ですので、プログラム資産を有効に活用できます。

③ パソコンをターミナルデバイスとして利用可能

FP-BASICを起動しているパソコンは、作動中のBASICタイプCPUのターミナルとしての機能を果たします。さらに、ファイルアクセス命令を使用すれば、パソコンのディスクドライブからBASICタイプCPUにデータを読み出すことができます。

④ PC専用マルチタスク言語

プログラマブルコントローラ（PC）のマルチタスク動作（最大16タスク）に対応するプログラムを、「FUNCTION ~ FEND」構文を使用して、容易に記述できます。なお、変数は、タスク間でグローバルにもローカルにも任意に宣言できます。

⑤ 先進のサブルーチンコール構文を採用

プログラムの中の1つのタスクから、他のタスクをサブルーチンとして呼び出すことができる「CALL ~ SUB」構文を採用。もちろん、このサブルーチンコールでは、変数の値の受け渡しも可能ですので、サブルーチンの独立性を完全に確保でき、構造化によるプログラム作成の効率化が期待できます。

⑥ 分割コンパイルをサポート

大きなプログラムを開発する場合、分割コンパイルを使用すれば、すっきりと見通しの良いソースプログラムが作成でき、プログラムのメンテナンスも容易になります。

⑦ 高機能ユニット専用制御命令の充実

BASICタイプCPUの高度な機能を十二分に活用するために、豊富な制御命令、関数を装備しています。とくに、FP3シリーズの高機能ユニット群の利用を助ける高機能ユニット専用制御命令の充実により、高度なシステムを比較的容易に構築することができます。

⑧ 高度なモニタ機能の装備

モニタ機能の充実により、作動中のプログラマブルコントローラ（PC）のI/O、メモリおよび変数の状態を容易に認識することができます。これにより、作成したプログラムのデバッグ、および現場調整が迅速に行えます。

⑨ 簡単なPC動作環境の設定

パソコンのベーシック画面へのコマンド入力、またはプログラムへの記述により、プログラマブルコントローラ（PC）の各種動作環境の設定と確認ができます。リモートI/Oシステムを含むスロット割り付け、PCリンクの設定など、ネットワークを含む複雑なシステムであっても、迅速かつ容易にPCの動作環境が設定できます。

注意

上記特長は、FP3H-BASICタイプおよびVer.2.0以降のFP3-BASICタイプを使用する場合に実現されます。Ver.2.0未満のFP3-BASICタイプを使用する場合、一部の機能が使用できません（このマニュアル中の「使用するコマンド・命令・関数」の記述の中で、◆が付いているものは使用できません）。

1-1-3

動作環境

FP-BASICが使用できるハードウェア環境、ソフトウェア環境は以下のとおりです。

■使用できるCPUについて

FP3H-BASICタイプ……FP-BASICのすべてのコマンド、命令、関数が使用できます
Ver.2.0以降の
FP3-BASICタイプ……FP-BASICのすべてのコマンド、命令、関数が使用できます
Ver.2.0未満の
FP3-BASICタイプ……使用できるコマンド、命令、関数が限定されます

*詳しくは、『ハード導入マニュアル』を参照してください。

■使用できるI/Oユニットについて

高機能ユニットを含むすべてのI/Oユニットが使用できます

■使用できるパソコンについて

日本電気株式会社	NEC PC9801 シリーズ
	*初代PC9801、PC9801E/F/U2、PC9801XA、PC9801LTは使用できません
	*PC9801M/VF/VM/UVは、メモリの増設が必要です(640KB実装)
セイコーエプソン株式会社	PC286 シリーズ/386 シリーズ

注意

FP-BASICはノーマル画面モード専用ソフトですので、一部機種(PC-98XL/XL2など)をハイレゾリューションモードで使用することはできません。ノーマル画面モードに設定して使用してください。

■使用できるオペレーションシステム

日本電気株式会社	NEC日本語MS-DOS	Ver.3.10以降
----------	--------------	------------

■使用するケーブルについて

(FP3(H)-BASICタイプとパソコンとを接続)

品名	仕様/その他	品番
RS232Cケーブル	ケーブル長3m	AFB85813

*ケーブルを自作する場合は、『ハード導入マニュアル』を参照してください。

1 - 2

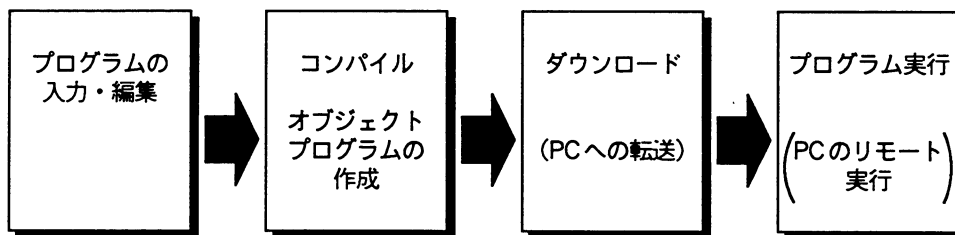
FP - BASIC の機能概要

FP - BASICには、BASICタイプCPUのためのプログラムを作成する機能と、CPUで作動中のプログラムをモニタする機能があります。

1-2-1 プログラミング機能

FP - BASICを使用して、BASICタイプCPUのプログラムを作成したり、作成したプログラムをCPUに転送したりします。また、FP - BASICは、CPUの各種動作環境を設定することもでき、BASICタイプCPUの<統合プログラミング環境>としての機能を備えています。

```
>LIST ..... プログラムリストを表示する
10 FUNCTION MAIN
20 XQT 13 T_B
30 XQT 12 T_A
:
:
80 END
90 FEND
>COMPILE ..... コンパイルする
COMPILE END
>DWNLD ..... CPUにダウンロード (転送) する
>XQT■
```



注意

- ダウンロード時には、あらかじめBASICタイプCPUのモードスイッチを「PROG.」にしてください。モードスイッチが「REMOTE」に設定されている場合は、CPUがRUN状態でないことを確認してください。
- プログラム実行時（リモート実行）には、あらかじめBASICタイプCPUのモードスイッチを「REMOTE」にしてください。

1-2-2 モニタ機能

FP-BASICを使用して、BASICタイプCPUで作動中のプログラムの実行状態をモニタします。モニタ機能を使用することによって、作成したプログラムのデバッグが簡単にでき、現場調整を円滑に進めることができます。

■ I/O、メモリおよび変数のモニタ

```

>MON SW(Y_&M10) ..... ビット単位で指定
MONITOR START
0 ..... 表示はリアルタイムに変化する
MONITOR END ..... [ESC] キーまたはリターンキーを押してモニタ終了
>MON INW(WY_1) ..... ワード単位で指定
MONITOR START
0000 H ..... 「H」キーを押すと16進表示
MONITOR END
>MON IND(WY_1) ..... ダブルワード単位で指定
MONITOR START
00000000 H
MONITOR ND
>■

```

■ I/O、メモリおよび変数のマルチポイントモニタ

```

COMMAND.COM
CURRENT DIRECTORY = A:¥
PATH = A:¥
COMMAND.COM  ESPEL.EXE  FPB.EXE  TEST.PRG
SYSTMP.BIN   SYSTMP.PRM  FKEY.DEF  CONFIG.SYS
AUTOEXEC.BAT  FP.EXE     PRINT.SYS  TEST_2.PRG
TEST_3.PRG

```

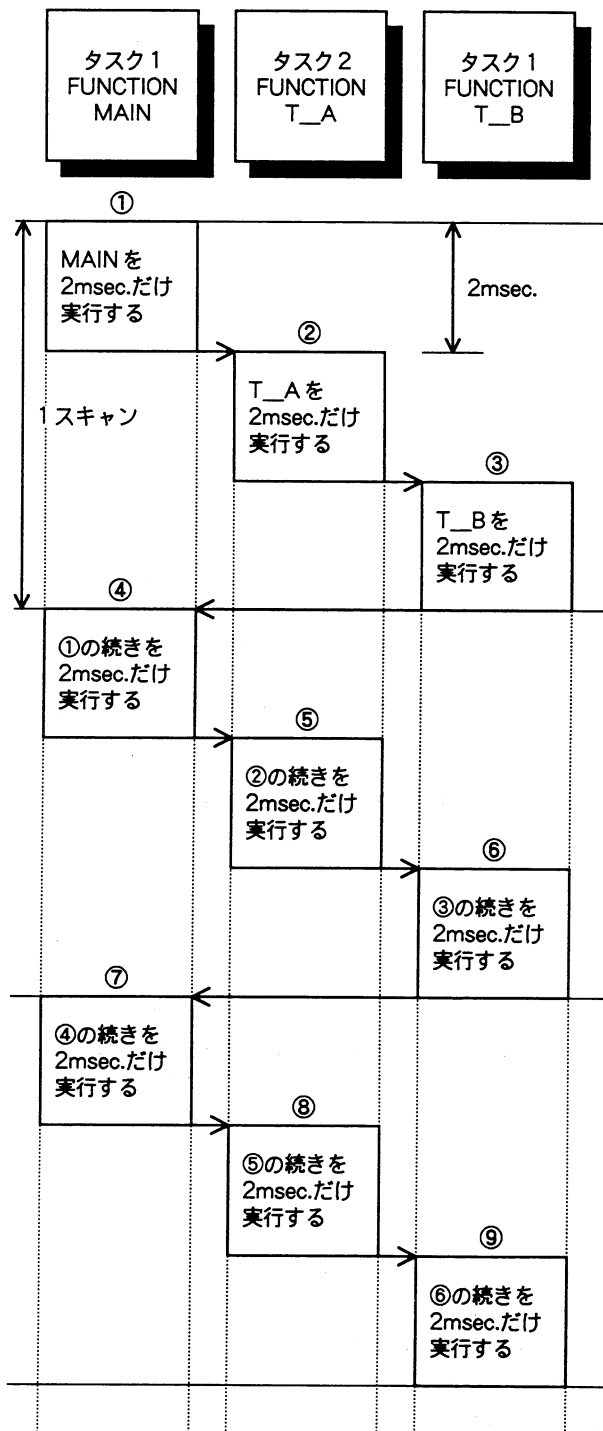
● モニタ可能な内容

- CPUの動作モード
- タスクごとのステータス表示
- タスクごとのステータスのリアルタイム表示
- I/O、メモリ、および変数のモニタ
- I/Oおよびメモリのマルチポイントモニタ
- I/Oおよびメモリのダンプ (状態表示)
- タイマ経過値の設定、および表示
- 高機能ユニットからの割込未処理の設定の一覧表示
- I/Oおよびメモリ、その他の状態の一括ダンプ (状態表示) と編集

1-2-3 マルチタスクプログラミング

BASICタイプCPUは、同時に16タスクまでの複数のプログラムを実行できます。このため、FP-BASICは、1つのプログラムで能率よくマルチタスクを記述できる「FUNCTION ~ FEND」構文を採用しています（「4-6-1. マルチタスクとFUNCTION文 (P.90)」参照）。

なお、FP-BASICで使用する変数には、タスク間で共有されるグローバルな変数と、タスク間で共有されないローカルな変数を任意に宣言できます。



第 2 章

準 備

こ の 章 の 内 容

2-1. ハードウェアの準備	20
2-1-1. 機器の接続方法	20
2-1-2. FP3(H)-BASICタイプCPUの設定	21
2-1-3. パソコンの設定	21
2-2. ソフトウェアの準備 (インストール)	22
2-2-1. インストールの概要	22
2-2-2. MS-DOSの知識のある方のために	23
2-2-3. フロッピーディスクへのインストール (起動ディスクの作成)	24
2-2-4. ノートパソコンへのインストール (起動ディスクの作成)	28
2-2-5. ハードディスクへのインストール	33
2-2-6. フロッピーディスクの初期化	36
2-2-7. フロッピーディスクの複写 (バックアップ)	38
2-2-8. ワープロのかな漢字変換の組み込み	40

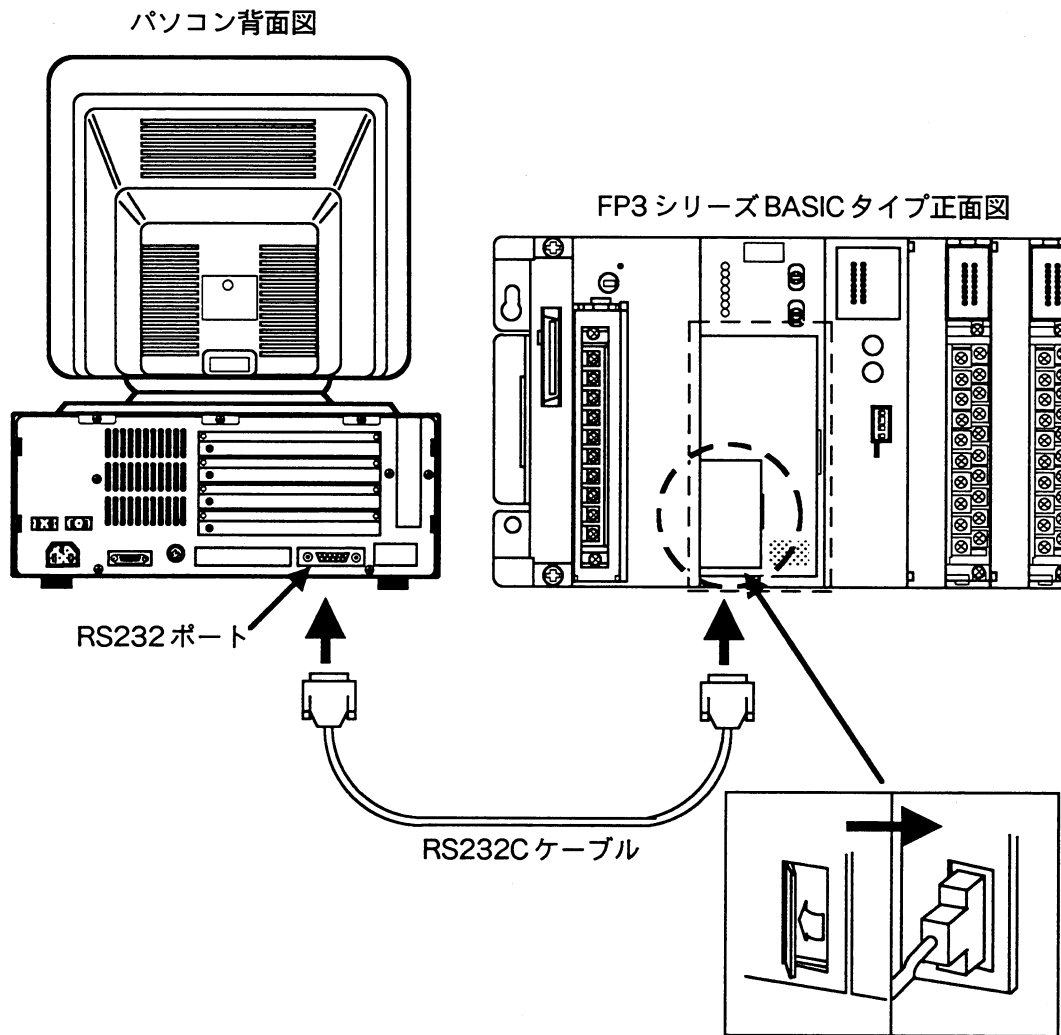
2-1

ハードウェアの準備

2-1-1 機器の接続方法

パソコンと BASICタイプCPUの接続には、RS232Cの通信ケーブル（専用品：弊社品番 AFB8513）を使用します。

RS813



*パソコン、FP3シリーズBASICタイプ、ケーブルの大きさは実際の比率ではありません

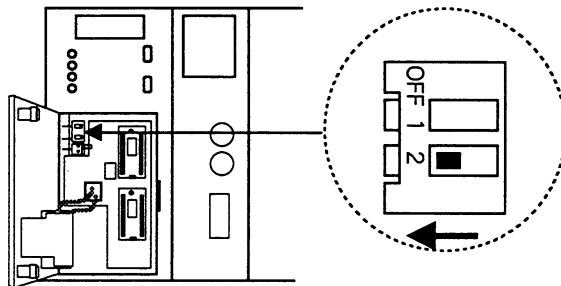
ユニットカバー内のコネクタにRS232Cケーブルを接続する

注意

接続時には、パソコンとFP3 (H) - BASICタイプCPUの電源を必ずOFFにしてください。

2-1-2 FP3H – BASIC タイプ CPU の設定

BASICタイプCPUの通信ボーレート切り換えスイッチを、OFF側に設定してください。通信ボーレートは、19200bpsです。

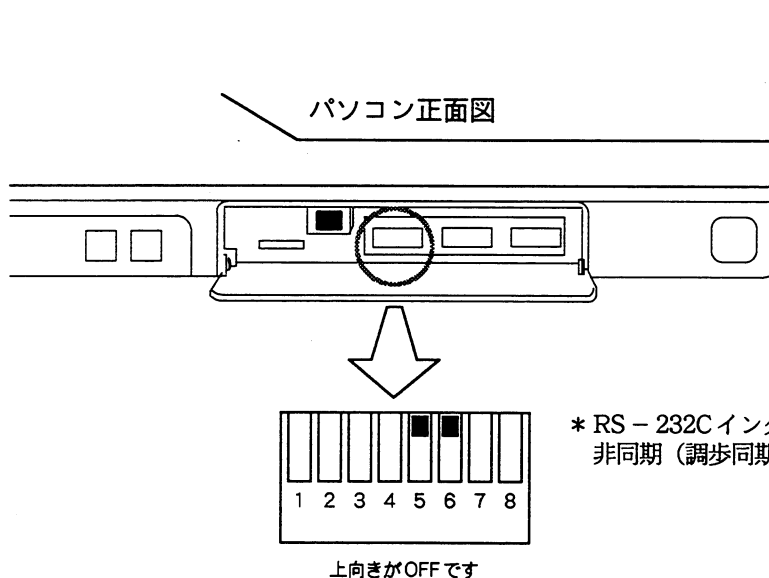


注意

- RS232Cケーブルは、専用品 (3mケーブル:弊社品番AFB⁸⁵⁸¹³) または相当品を使用してください。ケーブルを自作する場合は、『ハード導入マニュアル』を参照してください。
- パソコンの機種によっては、19200bpsでは動作しないことがあります。その場合は9600bpsに設定してください。(「2-2-2. 起動時のオプションコマンドについて (P.23)」参照)

2-1-3 パソコンの設定

パソコン (PC9801 シリーズ、PC286/386 シリーズ) のパネルカバーを開き、ディップスイッチ1番の「5」と「6」をOFF側にします。これにより、パソコンのRS232C通信ポートを非同期 (調歩同期) に設定できます。



* RS - 232C インターフェースの設定
非同期 (調歩同期)

注意

- ディップスイッチが無いパソコン (ノートパソコンなど) では、ソフトウェアにてRS232C通信ポートの設定をします。詳しくは、パソコン付属のマニュアルをご覧ください。
- PC9801F/Mをご使用される場合は、クロック周波数を5MHzに設定してください。クロック周波数が8MHzの場合、RS232Cを介しての通信ができない場合があります。このクロック周波数の設定はディップスイッチで行います。

2 - 2

ソフトウェアの準備（インストール）

2-2-1

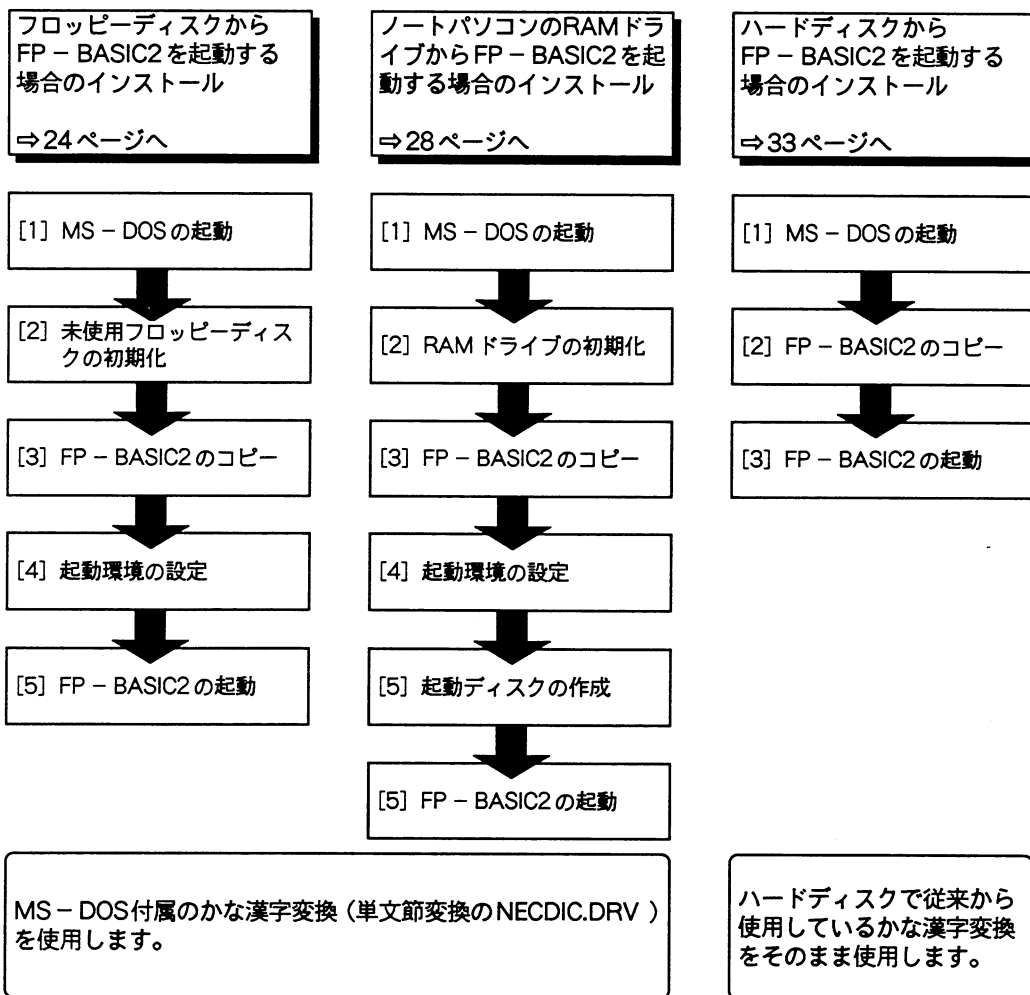
インストールの概要

FP - BASICを使用するには、MS - DOSとFP - BASICの入った起動ディスク（実用システムディスク）を作成するか、ハードディスクへFP - BASICを複写する作業が必要です。これらソフトを準備する作業をインストールと呼びます。

FP - BASICのオリジナルディスクの中には、インストールを簡単に行うためのプログラムが含まれていますので、このマニュアルに記述されているとおりに操作してください。なお、MS - DOSのシステムディスクは、お買い上げいただいた商品には含まれていませんので、「NEC日本語MS - DOS」（Ver.3.10以降）をお客様でご用意ください。

■インストールの方法

3種類のインストールがあります。使用するハードウェア環境によってインストールの方法が異なりますので、下記に指示されているページからお読みください。



注意

- MS - DOSは、必ず「NEC日本語MS - DOS」のVer.3.10以降のバージョンをご用意ください。
- ワープロなどで使用しているかな漢字変換を使用する場合は、「2-2-8. ワープロのかな漢字変換の組み込み（P.40）」を参照してください。

2-2-2 MS - DOS の知識のある方のために

以下の記述は、とくに MS - DOS の知識のある方以外は、読み流してください。

● CONFIG.SYS について

- (1) BUFFERSは1以上、FILESは8以上に設定してください(使用するFEPに合わせて設定します)。
- (2) プリンタドライバ PRINT.SYS の組み込みが必要です。 RSDRV.SYS の組み込みは不要です。
標準的なインストールでは、かな漢字変換に NECDIC.DRV が組み込まれます。

例：

```
A>TYPE CONFIG.SYS
BUFFERS=xx.....使用するFEPに合わせて設定してください
FILES=xx.....使用するFEPに合わせて設定してください
DEVICE=PRINT.SYS
DEVICE=NECDIC.DRV A:¥NECDIC.SYS
A>
```

● AUTOEXEC.BAT について

FP - BASICを起動する前に、RS232C通信ポートの設定は必要ありません (SPEED.COM SWITCH.COM等は不要です)。

例：

```
A>TYPE AUTOEXEC.BAT
FPB
A>
```


● 起動時のコマンドオプションについて

FP - BASICは、起動時にコマンドオプションを付けることで、[INS] キーの動作の変更、およびプログラムを読み出ししての起動ができます。

- (1) FPB で起動 (オプション無し) した場合

キーを押すことにより、カーソル位置にスペースを挿入します。

- (2) FPB/i で起動した場合

キーを押すことにより、上書きモードと挿入モードの切り替えができます。
挿入モード時には、カーソルの形状が  に変わります。

- (3) FPB/I で起動した場合

FPBASIC/i で起動した場合と同じ動作をしますが、カーソルの形状は変わりません。

- (4) FPB/S で起動した場合

起動時に、「 SYSTMP.PRG 」をメモリに読み出します。前回の終了時の状態で起動されます。

- (5) FPB/H で起動した場合

Ver.2.0未満のFP - BASICタイプCPUに対応します (AFP3251等)。

● メモリ不足について

FP - BASICは、約520Kバイト (532480バイト) 以上のフリーエリア (空きメモリ) を必要とします。メモリ不足が発生した場合は、FEPを取り外すか、EMS対応FEPに変更してください。後者の場合、当然EMSメモリおよびEMSドライバが実装されていなければなりません。
なお、FEPにNECDIC.DRVを使用する場合、EMSは必要ありません。

2-2-3 フロッピーディスクへのインストール（起動ディスクの作成）

フロッピードライブを2台持つパソコンのために、FP-BASICの起動ディスク（実用システムディスク）を作成する方法について説明します。

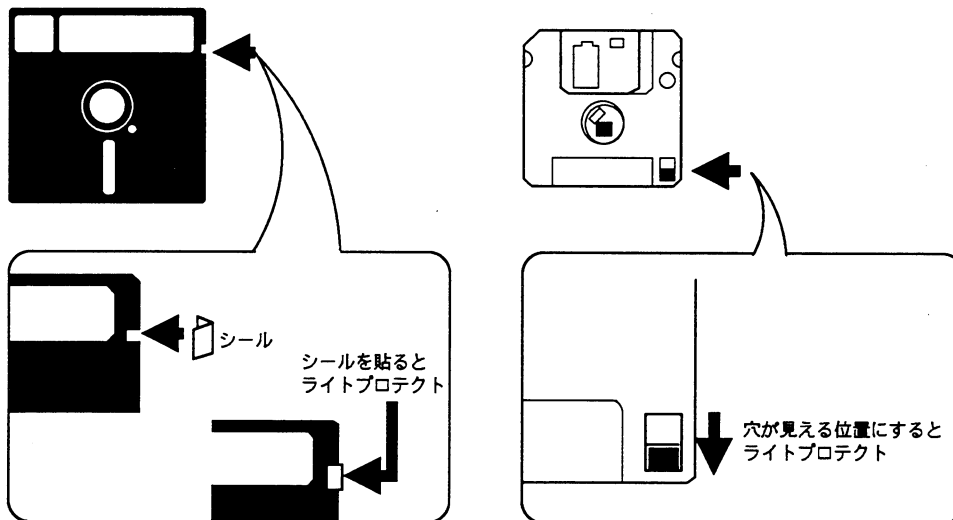
- 用意するもの：①FP-BASICのオリジナルディスク
 ②MS-DOSのシステムディスク（Ver.3.10以降）
 ③未使用のフロッピーディスク（2HDタイプ）

注意

FP-BASICのオリジナルディスクとMS-DOSのシステムディスクにはプロテクトを施してください。

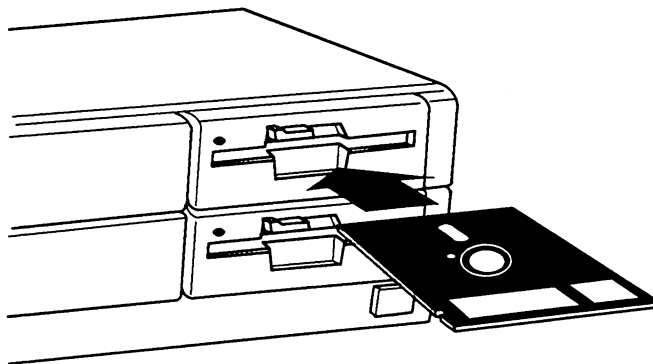
●5.25インチディスクの場合

●3.5インチディスクの場合



[1] MS-DOSの起動

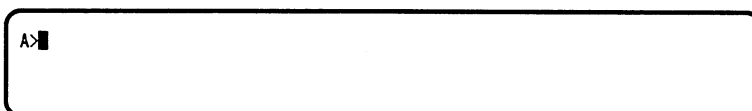
ドライブAに、MS-DOSの#1ディスクを入れて、パソコンを起動してください。



ドライブA：MS-DOS #1

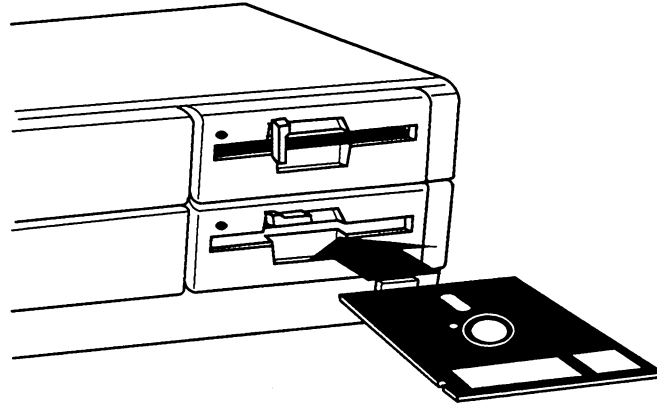
「メニュー」が表示される場合は、**STOP** キーを押して、MS-DOSのコマンド入力待ち状態（プロンプト「A>」が表示されている状態）にしてください。また、MS-DOSのバージョンによっては、「MS-DOSインストールプログラム」の画面が表示されますので、**ESC** キーを押してから、**↵** キーを押してください。

MS-DOSのコマンド入力待ち状態（■はカーソルです）



[2] 未使用フロッピーディスクの初期化

ドライブBに、未使用のフロッピーディスクを入れてください。



ドライブ A : MS - DOS # 1

ドライブ B : 未使用フロッピー

フォーマットコマンド「`FORMAT B: /S`」を以下のように入力して下さい。

```
A>FORMAT B: /S
```

下線部を入力し、**Enter** キーを押してください

以下が表示されます。

```
新しいディスクをドライブ B: に挿入し  
どれかキーを押してください
```

Enter キーまたは **Space** キーを押すと、以下が表示されます。

```
ディスクのタイプは 1 : 640(KB) 2 : 1(MB) =
```

ここで、**2**を入力し、**Enter** キーを押すと、次が表示されます。

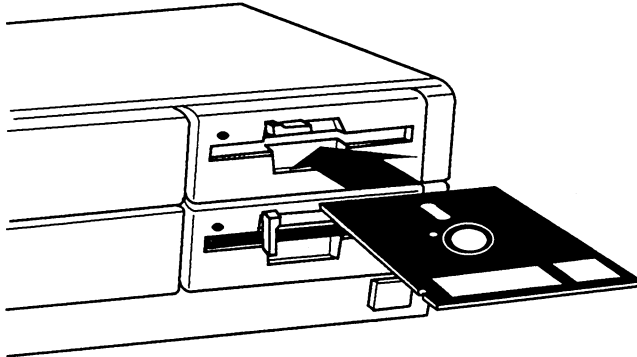
```
目的のディスクは 1MB FD です  
フォーマットが終了しました  
0 10 20 30 40 50 60 70 80 90 100  
████████████████████ (％)  
システムを転送しました  
***** バイト 全ディスク容量  
***** バイト システム領域  
***** バイト 使用可能ディスク容量  
別のディスクをフォーマットしますか (Y/N)
```

ここで、**N** キーを押し、つづけて **Enter** キーを押してください。

以上の操作で、起動ディスク用のフロッピーディスクが初期化されました。

[3] FP - BASICのコピー

ドライブAから、MS - DOSの#1ディスクを取り出し、かわりにFP - BASICのオリジナルディスクを入れてください。




ドライブA : FP - BASIC
オリジナルディスク

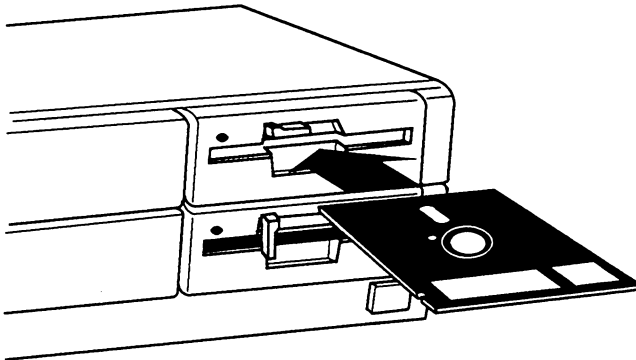
ドライブB : 未使用フロッピー
(そのままにしておきます)

コピーコマンド「COPY *.* B:」を実行します。ドライブBの起動ディスクに、FP - BASICのオリジナルディスク内のすべてのファイルがコピーされます。

```
A>COPY *.* B:
```

下線部を入力し、キーを押してください。

コピーが終了したら、ドライブAから、FP - BASICのオリジナルディスクを取り出し、かわりにMS - DOSの#1ディスクを入れてください。



ドライブA : MS - DOS #1

ドライブB : 未使用フロッピー
(そのままにしておきます)

[4] 起動環境の設定


カレントドライブを、ドライブBに変更します。「B:」を以下のように入力してください。

```
B>
```

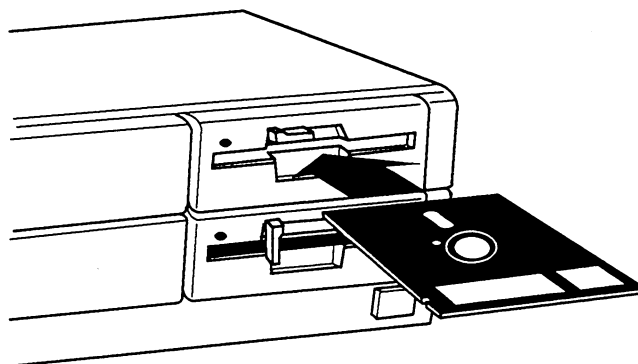
下線部を入力し、キーを押してください。

起動環境設定プログラムを、実行します。「SETUP A:」を以下のように入力してください。

```
B>SETUP A:
```

下線部を入力し、キーを押してください。

起動環境設定プログラムが実行されますので、画面の指示にしたがって **[スペース]** キーまたは **[↵]** キーを押してください。MS-DOSのバージョンによっては、画面の指示にしたがって、途中でMS-DOSの#1ディスクを、#2または#3ディスクに入れ換えてください。



ドライブA : MS-DOS
#1~#3

ドライブB : 未使用フロッピー
(そのまましておきます)

途中で、「かな漢字変換にNECDIC.DRV (単文節変換)を組み込みますか?」と表示されますので、**[Y]** または **[N]** キーを入力して、**[↵]** キーを押してください。

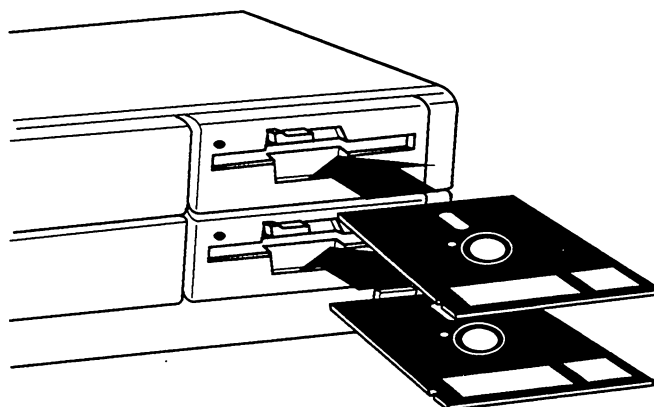
以上の操作で、ドライブBの起動ディスクに CONFIG.SYS と AUTOEXEC.BAT が作成され、PRINT.SYS と NECDIC.DRV が自動的にコピーされます。
MS-DOSの各ファイルの機能については、「NEC日本語MS-DOS」のマニュアルをご覧ください。

注意

かな漢字変換を組み込んだ場合、「NEC日本語MS-DOS」に付属している NECDIC.DRV (単文節変換)が、組み込まれます。他のかな漢字変換を組み込まれる方は、「2-2-8. ワープロのかな漢字変換の組み込み (P.40)」をお読みください。

[5] FP-BASICの起動

作成した起動ディスクをドライブAに入れて、パソコンのリセットボタンを押すと、FP-BASICが起動します。



ドライブA : FP-BASIC
起動ディスク

ドライブB : 辞書ディスク

注意

かな漢字変換を使用する場合は、ドライブBにかな漢字変換の辞書ディスクを入れてください。NECDIC.DRVを使用する場合、Ver.3.10のMS-DOSではMS-DOSの#2ディスクを、Ver.3.30以降ではMS-DOSの#3ディスクを使用します。辞書ディスクは、破損に備えて複写して使用することをお勧めします(「2-2-7. フロッピーディスクの複写 (バックアップ) (P.38)」参照)。なお、かな漢字変換の操作方法については、MS-DOSに付属のマニュアルまたはワープロ等のマニュアルをご覧ください。

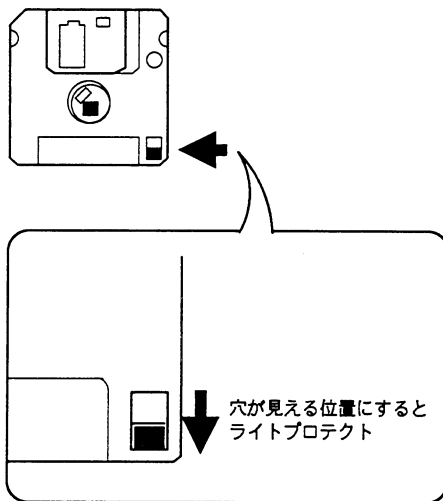
2-2-4 ノートパソコンへのインストール (起動ディスクの作成)

RAMドライブ付きのノートパソコンのために、FP-BASICの起動ディスク (実用システムディスク) を作成する方法について説明します。以下の作業を開始する前に、パソコン付属のマニュアルを参照し、起動ドライブをFD優先に、内蔵RAMドライブを使用するに、またレジューム機能を使用しないに設定してください。

- 用意するもの：①FP-BASICのオリジナルディスク
 ②MS-DOSのシステムディスク (Ver.3.10以降)
 ③未使用のフロッピーディスク (2HDタイプ)

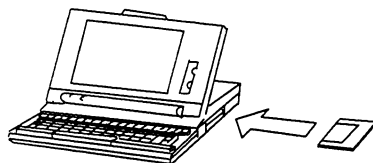
注意

FP-BASICのオリジナルディスクとMS-DOSのシステムディスクにはプロテクトを施してください。
 ● 3.5インチディスク



[1] MS-DOSの起動

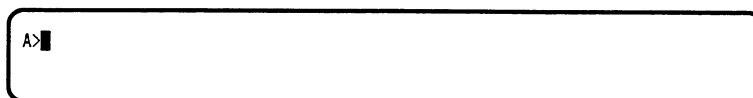
フロッピードライブに、MS-DOSの#1ディスクを入れて、パソコンを起動してください。



ドライブA : MS-DOS #1

「メニュー」が表示される場合は、**[STOP]** キーを押して、MS-DOSのコマンド入力待ち状態 (プロンプト「A>」が表示されている状態) にしてください。また、MS-DOSのバージョンによっては、「MS-DOSインストールプログラム」の画面が表示されますので、**[ESC]** キーを押してから、**[↵]** キーを押してください。

MS-DOSのコマンド入力待ち状態 (■はカーソルです)



[2] RAMドライブの初期化

フロッピードライブからMS-DOSを起動した場合、RAMドライブはドライブBになりますので、フォーマットコマンド「FORMAT B:/S」を以下のように入力してください。

```
A>FORMAT B:/S
```

下線部を入力し、キーを押してください

以下が表示されます。

```
新しいディスクをドライブ B: に挿入し  
どれかキーを押してください
```

ここで、キーまたは キーを押すと、以下が表示されます。

```
ディスクのタイプは 1 : 640(KB) 2 : 1(MB) =
```

ここで、を入力し、キーを押すと、次が表示されます。

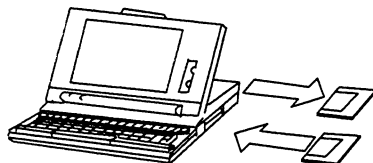
```
目的のディスクは 1MB FD です  
フォーマットが終了しました  
0 10 20 30 40 50 60 70 80 90 100  
██████████████████████ (%)  
  
システムを転送しました  
***** バイト 全ディスク容量  
***** バイト システム領域  
***** バイト 使用可能ディスク容量  
  
別のディスクをフォーマットしますか(Y/N)
```

ここで、キーを押し、つづけて キーを押してください。

以上の操作で、RAMドライブは初期化されました。

[3] FP-BASICのコピー

フロッピードライブ（ドライブA）から、MS-DOSの#1ディスクを取り出し、かわりにFP-BASICのオリジナルディスクを入れてください。



ドライブA : FP-BASIC
オリジナルディスク

コピーコマンド「COPY *.* B:」を実行してください。RAMディスク（ドライブB）に、FP-BASICのオリジナルディスク内のすべてのファイルがコピーされます。

```
A>COPY *.* B:
```

下線部を入力し、キーを押してください。

以上の操作で、FP-BASICのすべてのファイルがRAMドライブ（ドライブB）にコピーされます。

コピーが終了したら、フロッピードライブ（ドライブA）から、FP-BASICのオリジナルディスクを取り出し、かわりにMS-DOSの#1ディスクを入れてください。



ドライブA：MS-DOS #1

[4] 起動環境の設定

カレントドライブを、ドライブBに変更します。「B:」を以下のように入力してください。

```
A>B:
```

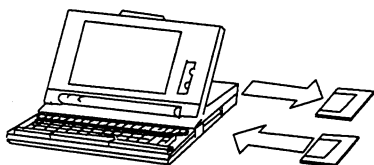
下線部を入力し、キーを押してください。

起動環境設定プログラムを、実行します。「SETUP A:」を以下のように入力してください。

```
B>SETUP A:
```

下線部を入力し、キーを押してください。

起動環境設定プログラムが実行されますので、画面の指示にしたがって キーまたは キーを押してください。MS-DOSのバージョンによっては、画面の指示にしたがって、途中でMS-DOSの#1ディスクを、#2または#3ディスクに入れ換えてください。



ドライブA：MS-DOS
#1～#3

途中で、「かな漢字変換にNECDIC.DRV（単文節変換）を組み込みますか?」と表示されますので、または キーを入力して、キーを押してください。

以上の操作で、RAMドライブに CONFIG.SYS と AUTOEXEC.BAT が作成され、PRINT.SYSと NECDIC.DRV が自動的にコピーされます。

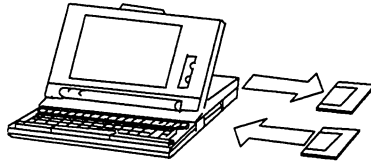
MS-DOSの各ファイルの機能については、「NEC日本語MS-DOS」のマニュアルをご覧ください。

注意

かな漢字変換を組み込んだ場合、「NEC日本語MS-DOS」に付属している NECDIC.DRV（単文節変換）が、組み込まれます。他のかな漢字変換を組み込まれる方は、「2-2-8. ワープロのかな漢字変換の組み込み（P.40）」をお読みください。

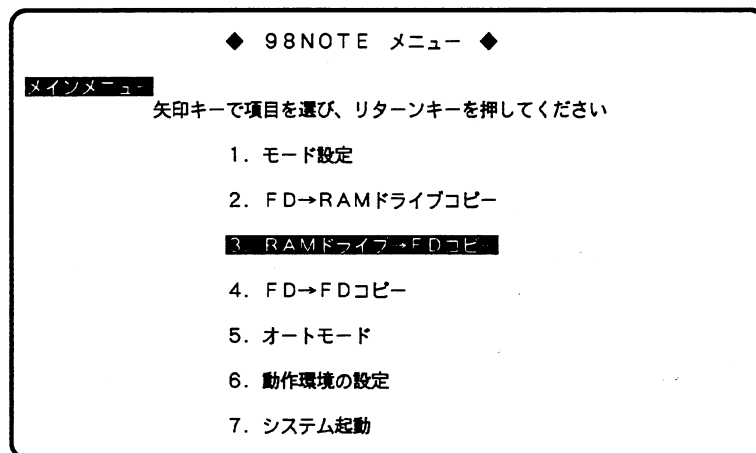
[5] 起動ディスクの作成

フロッピードライブから、MS-DOSのディスクを取り出し、未使用のフロッピーディスクを入れてください。



ドライブA：未使用フロッピー

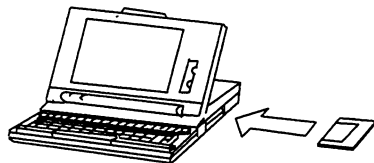
CTRL **GRPH** **HELP** の3つのキーを同時に押すことにより「ノートメニュー」が表示されますので、「RAMドライブ→FDコピー」の機能を選択して、RAMドライブの内容をフロッピーディスクにコピーしてください。



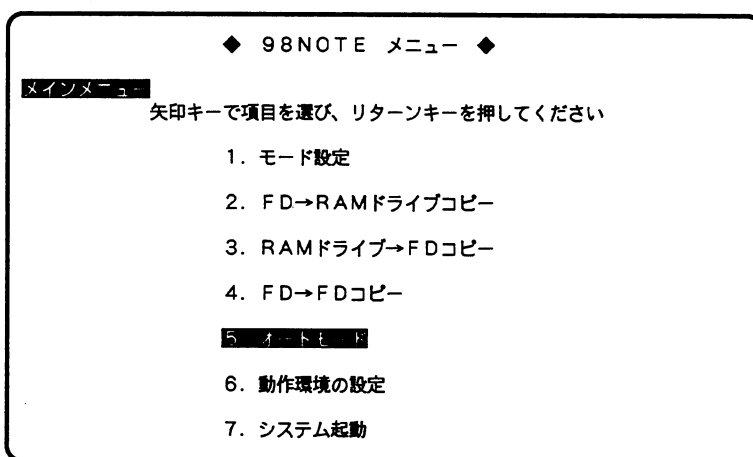
以上で、ノートパソコン用の起動ディスクの作成が終わります。フロッピーディスクを取り出し、電源を切ってください。

[6] FP - BASICの起動

作成した起動ディスクをフロッピードライブに入れて、「ノートメニュー」の「オートモード」を使用して、FP - BASICを起動してください。



ドライブ A : FP - BASIC
起動ディスク



注意

- 以降、RAMドライブの内容が保持されている限り、起動ディスクがなくても、FP - BASICは起動できます。ただし、作成したプログラムをRAMドライブ上に置くと、バッテリー切れなどで消失することがあります。したがって、必要に応じて「ノートメニュー」の「RAMドライブ→FDコピー」を使用して、RAMドライブ上のプログラムを起動ディスクに移動させるなどの対策が必要です。
 - かな漢字変換を使用する場合は、フロッピードライブに、かな漢字変換の辞書ディスクを入れてください。NECDIC.DRVを使用する場合は、Ver.3.10のMS - DOSではMS - DOSの#2ディスクを、Ver.3.30以降ではMS - DOSの#3ディスクを使用します。辞書ディスクは、破損に備えて複製して使用することをお勧めします（「2-2-7. フロッピーディスクの複製（バックアップ）」（P.38）」参照）。
- なお、かな漢字変換の操作方法については、MS - DOSに付属のマニュアルまたはワープロ等のマニュアルをご覧ください。

2-2-5 ハードディスクへのインストール

FP-BASICをハードディスクにインストールする方法を説明します。

ここでは、あらかじめハードディスクにMS-DOS (Ver.3.10以降) のシステムと、ワープロなどの他のアプリケーションが既にインストールされているものとします。ハードディスクにMS-DOSがインストールされていない場合は、MS-DOS付属のマニュアルをお読みになり、先にMS-DOSのインストールを行ってください。

注意

ハードディスクから起動したとき、CONFIG.SYS ファイルにプリンタドライバ PRINT.SYSが登録されている必要があります。

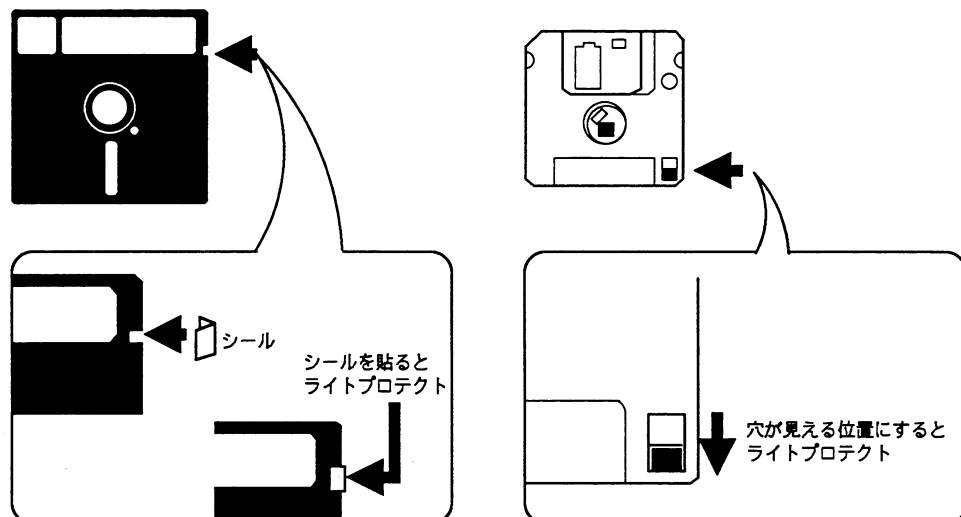
用意するもの：FP-BASICのオリジナルディスク

注意

FP-BASICのオリジナルディスクにはプロテクトを施してください。

● 5.25インチディスクの場合

● 3.5インチディスクの場合



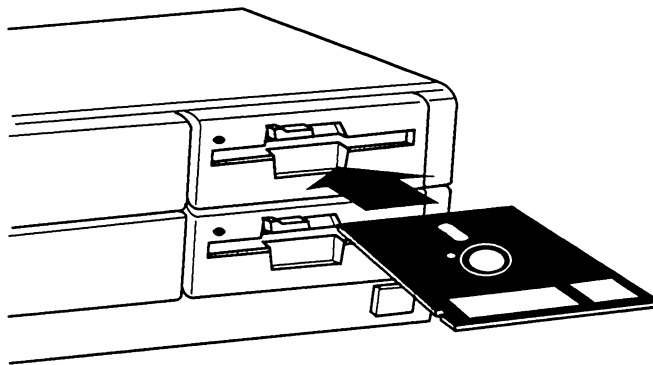
[1] MS-DOSの起動

ハードディスクからパソコンを起動します。他のアプリケーションソフト、または「メニュー」が表示される場合は、それらを終了して、MS-DOSのコマンド入力待ち（プロンプト「A>」が表示されている状態）にしてください。

MS-DOSのコマンド入力待ち状態（■はカーソルです）



フロッピードライブ（ドライブB）にFP-BASICのオリジナルディスクを入れてください。



ドライブB：FP-BASIC
オリジナルディスク


注意

ハードディスクからMS-DOSを起動すると、1番目のフロッピードライブはドライブBになります。ただし、ハードディスクが2台接続されている場合は、1番目のフロッピーディスクはドライブCになりますので、以下の説明の「B:」は「C:」と読み替えてください。

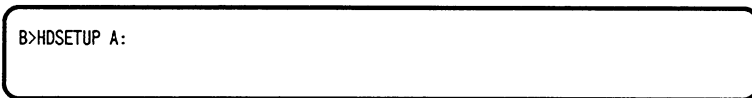
[2] FP-BASICのコピー

カレントドライブを、ドライブBに変更します。「B:」を以下のように入力してください。



下線部を入力し、キーを押してください。

インストールプログラムを、実行します。「HDSETUP A:」を以下のように入力してください。



下線部を入力し、キーを押してください。


以上の操作で、ハードディスク（ドライブA）にサブディレクトリFPBASICが作成され、FP-BASICの必要なファイルがコピーされます。また、ハードディスクのルートディレクトリに、FPB.BATが作成されます。

[3] FP - BASICの起動


インストールが終了しましたので、ドライブBからFP - BASICのオリジナルディスクを取り出して、パソコンのリセットボタンを押してください。自動起動アプリケーション、または「メニュー」が表示される場合は、それらを終了させてMS - DOSのコマンド入力待ち（プロンプト「A>」が表示されている状態）にしてください。

MS - DOSのコマンド入力待ち状態（■はカーソルです）

```
A>■
```

「A>」の後に、「FPB」と入力し、キーを押します。

```
A>FPB
```

下線部を入力し、キーを押してください。

以上の操作で、FP - BASICが起動されます

かな漢字変換を使用する場合、ハードディスクで従来から使用しているかな漢字変換がそのまま使用できます。

注意

メモリ不足によりハードディスクからFP - BASICが正常に起動できない場合は、使用中のかな漢字変換デバイスドライバ（FEP）を外すか、NECDIC.DRVに変更してください。どちらの場合も、CONFIG.SYSの内容を変更しなければなりません。詳しくは、MS - DOSおよび使用中のFEPのマニュアルをご覧ください。

2-2-6 フロッピーディスクの初期化

フロッピーディスクは、初期化をしてからでないと、データの読み書きやファイルのコピーに使用できません。以下の手順にしたがえば、フロッピードライブが1台しかないパソコン（ノートパソコンなど）の場合も、フロッピーディスクの初期化が簡単に行えます。

なお、初期化済みで販売されているフロッピーディスクの場合はこの作業は不要です

注意

ノートパソコンの場合は、「起動ドライブ」を「FD優先」に、またレジューム機能を「使用しない」に設定してください。

用意するもの：①MS-DOSのシステムディスク（Ver.3.10以降）

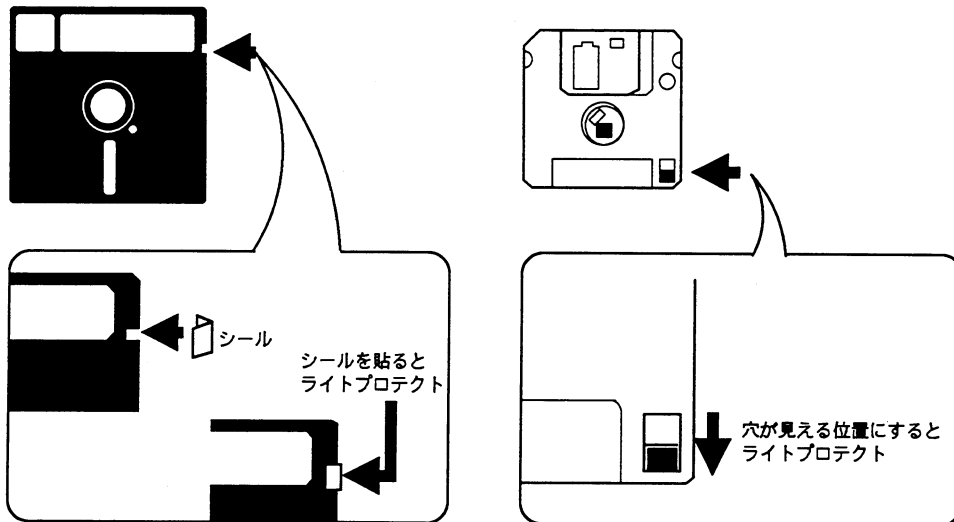
②未使用のフロッピーディスク（2HDタイプ）

注意

MS-DOSのシステムディスクにはプロテクトを施してください。

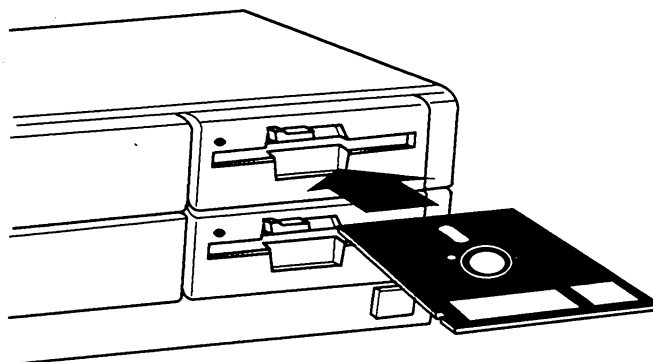
●5.25インチディスクの場合

●3.5インチディスクの場合



[1] MS-DOSの起動

フロッピードライブに、MS-DOSの#1ディスクを入れて、パソコンを起動してください。



ドライブA：MS-DOS #1

「メニュー」が表示される場合は、**[STOP]** キーを押して、MS-DOSのコマンド入力待ち状態（プロンプト「A>」が表示されている状態）にしてください。また、MS-DOSのバージョンによっては、「MS-DOSインストールプログラム」の画面が表示されますので、**[ESC]** キーを押してから、**[↵]** キーを押してください。

MS-DOSのコマンド入力待ち状態 (■はカーソルです)

A>■

[2] 未使用フロッピーディスクの初期化

「A>」の後に、以下のように入力してください。

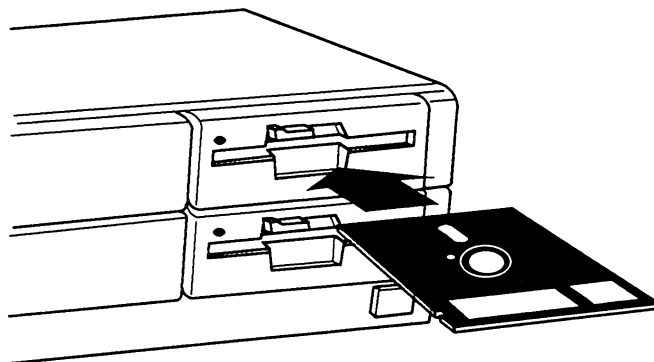
A>FORMAT A:

下線部を入力し、キーを押してください。

以下が表示されます。

新しいディスクをドライブ B: に挿入し
どれかキーを押してください

ここで、フロッピードライブから、MS-DOSの#1ディスクを取り出し、未使用のフロッピーディスクを入れてください。未使用フロッピーディスクにはプロテクトを施さないでください。



ドライブA: 未使用フロッピー

つぎに、キーまたは キーを押すと、以下が表示されます。

ディスクのタイプは 1 : 640(KB) 2 : 1(MB) =

ここで、を入力し、キーを押すと、次が表示されます。

目的のディスクは 1MB FD です

フォーマットが終了しました
0 10 20 30 40 50 60 70 80 90 100
██████████████████████ (%)

システムを転送しました
***** バイト 全ディスク容量
***** バイト システム領域
***** バイト 使用可能ディスク容量

別のディスクをフォーマットしますか(Y/N)

ここで、キーを押し、つづけて キーを押してください。以上の操作で、フロッピーディスクは初期化され、データの読み書きやファイルのコピーに使用できます。

2-2-7 フロッピーディスクの複写 (バックアップ)

FP-BASICのオリジナルディスクやかな漢字変換の辞書ディスクは、破損に備えて複写 (バックアップ) されることをお勧めします。

ノートパソコンの場合は、「ノートメニュー」の機能を使用して、簡単にフロッピーディスクのコピーができます (「FD→RAMドライブコピー」「RAMドライブ→FDコピー」を繰り返します)。「ノートメニュー」のない一般のパソコンでは、以下の手順にしたがいます。

用意するもの：①MS-DOSのシステムディスク (Ver.3.10以降)

②FP-BASICのオリジナルディスクやかな漢字変換の辞書ディスクなど複写しておきたいフロッピーディスク

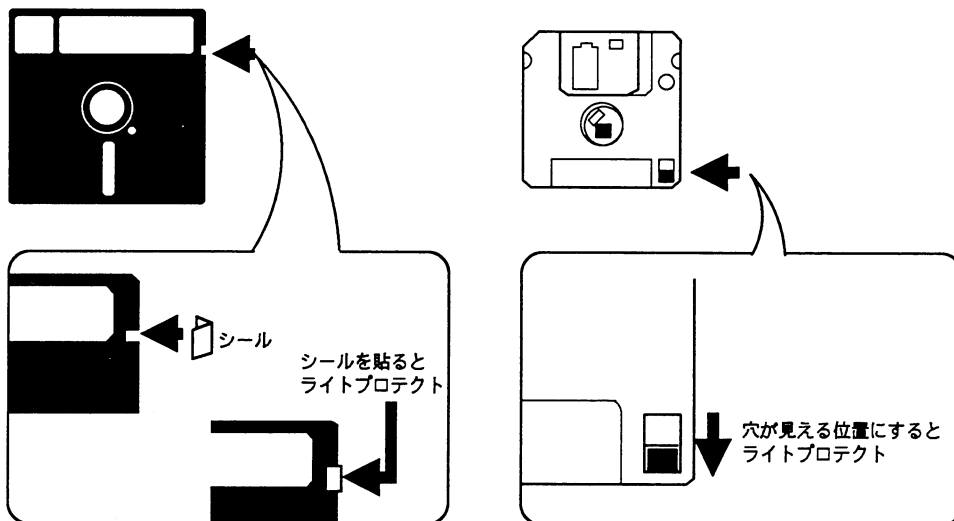
③初期化済みのフロッピーディスク (「2-2-6. フロッピーディスクの初期化 (P.36)」参照)

注意

FP-BASICのオリジナルディスク、かな漢字変換の辞書ディスクなどの複写元のフロッピーディスク、およびMS-DOSのシステムディスクにはプロテクトを施してください。

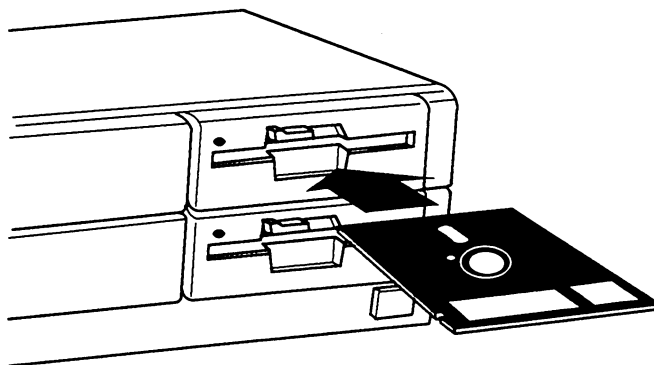
●5.25インチディスクの場合

●3.5インチディスクの場合



[1] MS-DOSの初期化

ドライブAに、MS-DOSの#1ディスクを入れて、パソコンを起動してください。



ドライブA : MS-DOS #1

「メニュー」が表示される場合は、**[STOP]** キーを押して、MS-DOSのコマンド入力待ち状態 (プロンプト「A>」が表示されている状態) にしてください。また、MS-DOSのバージョンによっては、「MS-DOSインストールプログラム」の画面が表示されますので、**[ESC]** キーを押してから、**[↵]** キーを押してください。

MS-DOSのコマンド入力待ち状態 (■はカーソルです)

A>■

[2] フロッピーディスクのバックアップ (ディスクコピー)

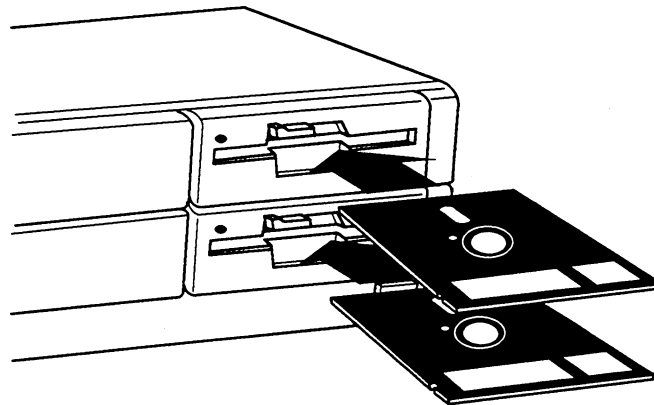
ディスクコピーコマンド「DISKCOPY A: B:」を入力します。

A>DISKCOPY A: B:

下線部を入力し、キーを押してください。



以下の画面が表示されるので、MS-DOSの#1ディスクをドライブAから取り出し、FP-BASICのオリジナルディスクやかな漢字変換の辞書ディスクなどの複写元のフロッピーディスクをドライブAに入れてください。次に、初期化済みのディスク(「2-2-6. フロッピーディスクの初期化 (P.36)」参照)をドライブBに入れてください。

送り側のディスクをドライブA:に挿入してください
受け側のディスクをドライブB:に挿入してください
準備ができましたらどれかのキーを押してください



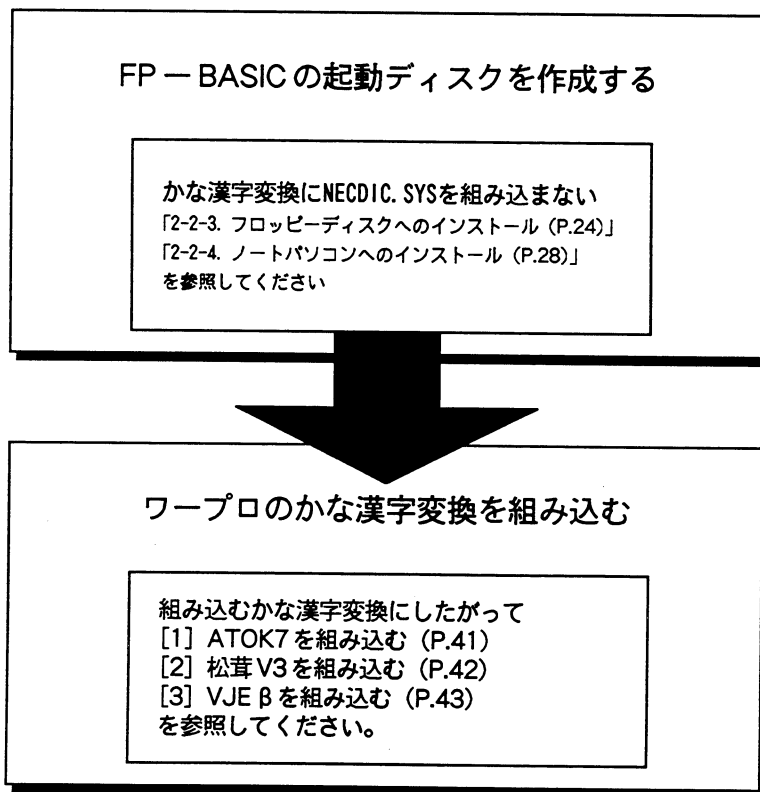
ドライブA: 複写元フロッピー

ドライブB: 初期化済み
フロッピー

ドライブAとドライブBへのフロッピーのセットが終わってから、キーまたは キーを押します。複写作業が始まり、ドライブAの複写元のフロッピーディスクの内容はすべてドライブBのフロッピーディスクに複写されます。

2-2-8 ワープロのかな漢字変換の組み込み

起動ディスクに、使いなれたワープロのかな漢字変換を組み込む方法について説明します。
ただし、NECDIC.DRV以外のかな漢字変換を組み込む場合は、拡張メモリ (EMS) が必要です。また、組み込むかな漢字変換ドライバ (FEP) も、EMS対応でなければなりません。
ここでは、一太郎 Ver.3のATOK7、松 Ver.5の松茸 V2、およびVJE βをFP - BASICの起動ディスクに組み込む方法について説明します。



[1] ATOK7 (一太郎 Ver.4 のかな漢字変換) の組み込み

ATOK7を組み込むには、一太郎 Ver.4の「Large/Small辞書+ユーティリティ」ディスクの ATUT.EXE を使用します。以下の手順で作業してください。

注意

起動ディスク作成時に、「かな漢字変換にNECDIC.DRV (単文節変換)を組み込みますか?」の表示に対しては、**[N]**を入力してください。

- ① FP - BASICの起動ディスクを使用して、FP - BASICを起動し、すぐに終了します。
- ② 画面に「A>」が表示されていることを確認します。
以下、キーボードからの入力、すべて「A:」の後に続けて入力されます。
- ③ 現在組み込まれているかな漢字変換システム (デバイスドライバ) を削除します。
「DEL NEC*. *」と入力し、**[↵]**キーを押します。

```
A>DEL NEC*. *
```

- ④ ドライブBに、一太郎の「Large/Small辞書+ユーティリティ」ディスクを入れます。
- ⑤ 「B:ATUT」と入力し、**[↵]**キーを押します。

```
A>B:ATUT
```

- ⑥ ATUT.EXE が起動し、環境設定の画面が表示されますので、自分の好みに合わせて設定し、最後に**[↵]**キーを押します。
ただし、このとき必ず、組み込むドライブ名にAドライブを指定してください。

- ⑦ ドライブBを一太郎 Ver.4の「起動」ディスクに交換してください。

- ⑧ 画面に「A>」が表示されていることを確認し、「COPY B:ATOK7*. SYS」と入力し、**[↵]**キーを押します。

```
A>COPY B:ATOK7*. SYS
```

以上で、ドライブAのFP - BASIC起動ディスクにATOK7が組み込まれました。

以後、FP - BASICを使用するときに、ドライブBに一太郎 Ver.4の「辞書[標準]+システム2」ディスクを入れることにより、一太郎 Ver.4と同様のかな漢字変換が使用できます。

一太郎 Ver.4の「辞書[標準]+システム2」ディスクは、破損事故に備えて複製したものを使用することをお薦めします。フロッピーディスクの複製については、「2-2-7. フロッピーディスクの複製 (バックアップ) (P.38)」を参照してください。

[2] 松茸 V3 (松 Ver.5 のかな漢字変換) の組み込み

松茸 V2 はワープロ「松 Ver.5」のかな漢字変換システムです。ここでは、松 Ver.5 の実用システムを使用して、松茸 V2 を組み込む方法を説明します。以下の手順で作業してください。

注意

起動ディスク作成時に、「かな漢字変換に NECDIC.DRV (単文節変換) を組み込みますか?」の表示に対しては、**[N]** を入力してください。

① FP - BASIC の起動ディスクを使用して、FP - BASIC を起動し、すぐに終了します。

② 画面に「A>」が表示されていることを確認します。

以下、キーボードからの入力は、すべて「A:」の後に続けて入力されます。

```
A>■
```

③ 現在組み込まれているかな漢字変換システム (デバイスドライバ) を削除します。

「DEL NEC*. *」と入力し、**[↵]** キーを押します。

```
A>DEL NEC*. *
```

④ ドライブ B に、松の「辞書ターボ」ディスクを入れます。

⑤ 「B:SETMTTK A:」と入力し、**[↵]** キーを押します。

```
A>B:SETMTTK A:
```

⑥ 環境設定の画面が表示されますので、自分の好みに合わせて設定し、最後に **[ESC]** キーを押します。

⑦ ドライブ B を松の「松システム1」ディスクに交換します。

⑧ 画面に「A>」が表示されていることを確認し、「COPY B:MTTK3.*」と入力し、**[↵]** キーを押します。

```
A>COPY B:MTTK3.*
```

以上で、ドライブ A の FP - BASIC 起動ディスクに松茸 V3 が組み込まれました。

以後、FP - BASIC を使用するとき、ドライブ B に松の「松システム1」ディスクを入れることにより、松茸 V3 のかな漢字変換が使用できます。

松の「松システム1」ディスクは、破損事故に備えて複写したものを使用することをお勧めします。フロッピーディスクの複写については、「2-2-7. フロッピーディスクの複写 (バックアップ) (P.38)」を参照してください。

[3] VJE β の組み込み

VJE - β は市販アプリケーション各種に幅広く採用されています。

VJE - β を採用しているアプリケーションには、「VJE - β 」ディスクが付属していますので、これを使用します。組み込む VJE - β は、EMS 対応バージョンでなければなりません。

注意 起動ディスク作成時に、「かな漢字変換に NECDIC.DRV (単文節変換) を組み込みますか?」の表示に対しては、**[N]** を入力してください。

① FP - BASIC の起動ディスクを使用して、FP - BASIC を起動し、すぐに終了します。

② 画面に「A>」が表示されていることを確認します。

以下、キーボードからの入力は、すべて「A:」の後に続けて入力されます。

```
A>■
```

③ 現在組み込まれているかな漢字変換システム (デバイスドライバ) を削除します。

「DEL NEC*. *」と入力し、**[↵]** キーを押します。

```
A>DEL NEC*. *
```

④ ドライブ B に、「VJE - β 」ディスクを入れます。

⑤ 「B:SETVJE A:」と入力し、**[↵]** キーを押します。

```
A>B:SETVJE A:
```

⑥ 環境設定の画面が表示されますので、自分の好みに合わせて設定し、最後に **[ESC]** キーを押して、メニューから「登録」を選びます。

⑦ 画面に「A>」が表示されていることを確認し、「COPY B:VJEB.SYS」と入力し、**[↵]** キーを押します。

```
A>COPY B:VJEB.SYS
```

以上で、ドライブ A の FP - BASIC 起動ディスクに VJE - β が組み込まれました。

以後、FP - BASIC を使用するとき、ドライブ B に「VJE - β 」ディスクを入れることにより、VJE - β のかな漢字変換が使用できます。

「VJE - β 」ディスクは、破損事故に備えてコピーしたものを使用することをお勧めします。フロッピーディスクの複写については、「2-2-7. フロッピーディスクの複写 (バックアップ) (P.38)」を参照してください。

第 3 章

3

FP – BASIC の プログラミング

こ の 章 の 内 容

3-1. FP – BASICの起動と終了——46	3-6. プログラムの実行——66
3-1-1. 起動画面……………46	3-6-1. プログラムの実行……………66
3-1-2. FP – BASICの終了方法……………46	3-6-2. トレースモード……………66
3-1-3. キー操作の概要……………47	3-6-3. テストモード実行……………67
3-1-4. ファンクションキーへの コマンド割り付け……………48	
3-2. CPUの初期設定——50	3-7. 実行中のプログラムのモニター—68
3-2-1. CPUのバージョン表示……………50	3-7-1. CPUの動作モードの表示……………68
3-2-2. CPUのメモリの初期化……………50	3-7-2. タスクごとのステータス表示……………68
3-2-3. パラメータメモリの設定……………51	3-7-3. タスクごとのステータスの リアルタイム表示……………69
3-2-4. 変数領域の割り付け……………53	3-7-4. I/O、メモリおよび変数のモニター ……70
3-2-5. スロット割り付け……………54	3-7-5. I/Oおよびメモリの マルチポイントモニター……………71
3-3. プログラム編集の基本操作——56	3-7-6. I/Oおよびメモリの ダンプ (状態表示)……………72
3-3-1. プログラム入力と編集の基本……………56	3-7-7. タイマ経過値の表示および設定……………73
3-3-2. 行の移動……………58	3-7-8. 高機能ユニットの割り込み設定の表示73
3-3-3. プログラム中の文字列検索……………59	3-7-9. I/Oおよびメモリその他の状態の 一括ダンプ (状態表示)と編集……………74
3-4. ファイルのセーブとロード——60	3-7-10. 強制入出力……………76
3-4-1. プログラムのセーブ (保存)……………60	3-7-11. I/O、メモリ、変数の セーブとロード……………76
3-4-2. プログラムのロード (読み出し)……………60	
3-4-3. ファイル管理……………61	
3-4-4. ファイル選択ウィンドウの使用方法…62	
3-5. オブジェクトプログラムの 作成と転送——64	
3-5-1. コンパイル……………64	
3-5-2. ダウンロード……………65	
3-5-3. アップロード……………65	

3 - 1

FP - BASIC の起動と終了

FP - BASICの起動画面と、終了方法について説明します。起動方法については、「2-2. ソフトウェアの準備 (インストール)」の中からご自分のインストールと同じ項の「FP - BASICの起動」をご覧ください。また、必ず BASIC タイプ CPU を接続してください。

3-1-1 起動画面

FP - BASICの起動画面は、以下のとおりです。

```
FP-BASIC Version 2.xx by Matsushita Electric Works, Ltd
```

「>」は、プロンプトと呼び、コンピュータがユーザからの命令入力を待っていることを示します。「■」は、カーソルといい、キーボードから入力された文字がここに表示されます。

3-1-2 FP - BASIC の終了方法

FP - BASICを終了させるには、EXIT 命令を入力します。確認のメッセージが表示されますので、終了の場合は キーを、終了せず FP - BASICに戻る場合は キーを押してください。

```
>EXIT  
Sure (Y/N):Y.....プログラム編集中の場合
```

FP - BASICが終了し、MS - DOSのプロンプトが表示されます。

```
A>.....MS - DOSのコマンドレベルに移ります
```

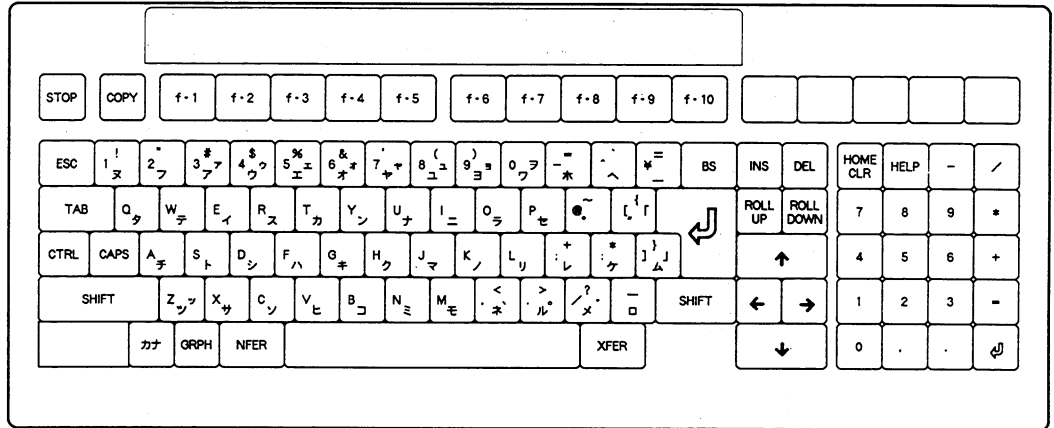
注意















MS - DOSの入力待ち (「A>」が表示されている状態) で、「FPB」と入力して、 キーを押すと、再び FP - BASICが起動されます。

●使用するコマンド

EXIT	FP - BASICを終了します
------	------------------

3-1-3 キー操作の概要

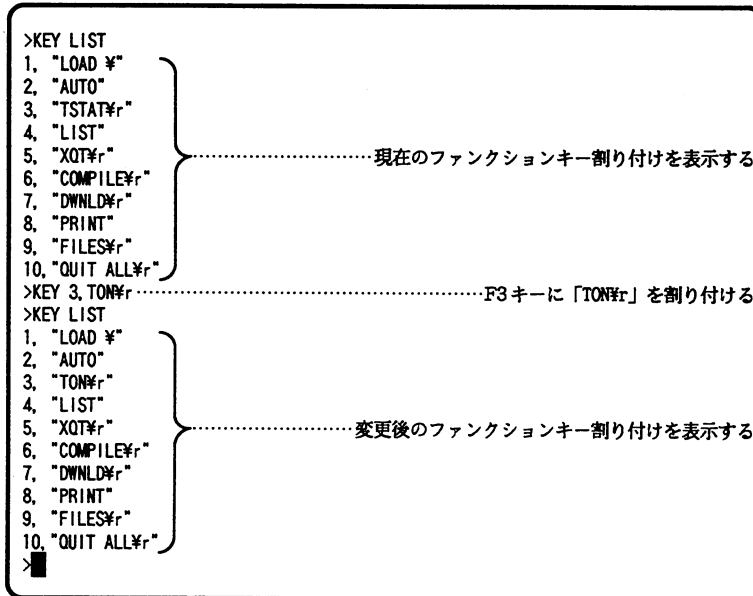



キー	キーの名称	機能
   	カーソルキー	カーソルが上下左右に1文字分移動します 行頭の1桁を除いて、文字が表示されていない位置には移動しません 最上行・最下行で   を押すと画面がスクロールします
	スクロールキー	カーソルが最上行に移動し、画面がスクロールします
		カーソルが最下行に移動し、画面がスクロールします
	ホームクリアキー	エディット画面を消去します
	タブキー	カーソルを8桁右に移動します
	インサートキー	カーソル位置に空白（スペース）を1文字挿入します *起動時のオプションで機能が変化します (「P.23 ●起動時のコマンドオプションについて」を参照してください)
	デリートキー	カーソルの左隣の文字を消去し、その位置にカーソルを移動します
	バックスペースキー	カーソルを左に1文字分移動します (そこにあった文字は消去しません)
	ストップキー	実行中のプログラムを停止します エラー解除の際などに使用しますが、変数の内容は保持されています (ストップキーを押した際のBASICタイプCPUの状態はパラメータメモリの設定内容により異なります)

3-1-4 ファンクションキーへのコマンド割り付け

FP - BASICの画面の下部には、ファンクションキーのコマンド割り付けの内容が表示されます。この機能を利用すれば、少ないキータッチで能率的に入力できます。

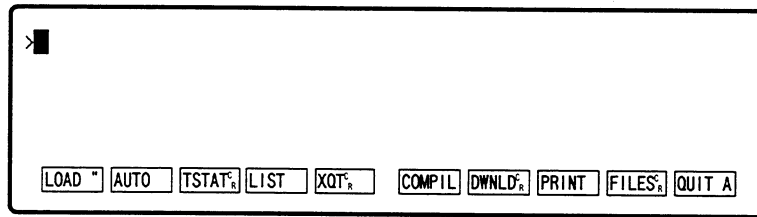
KEYコマンドおよびKEY LISTコマンドを使用して、ファンクションキーに任意のコマンドを割り付けることができます。



* 「TON¥r」は、キーボードから「TON」と入力して、キーを押すことを意味します。

●使用するコマンド

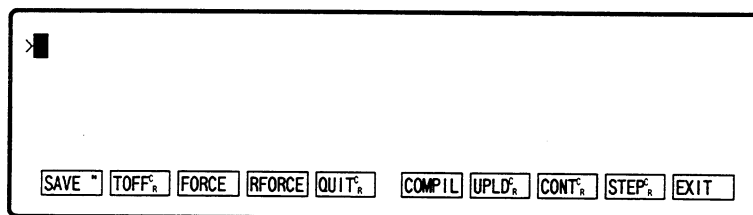
KEY	ファンクションキーにコマンドを割り付けます
KEY LIST	ファンクションキーのコマンド割り付けの内容を表示します



コマンド割り付け

ファンクションキー	割り付けられた内容
f・1	「LOAD "」を入力する
f・2	「AUTO」を入力する
f・3	「TSTAT」を入力して、リターンキーを押す
f・4	「LIST」を入力する
f・5	「XQT」を入力して、リターンキーを押す
f・6	「COMPILE」を入力して、リターンキーを押す
f・7	「DWNLD」を入力して、リターンキーを押す
f・8	「PRINT」を入力する
f・9	「FILES」を入力して、リターンキーを押す
f・10	「QUIT ALL」を入力して、リターンキーを押す

● (SHIFT) キーを押したとき



コマンド割り付け

ファンクションキー	割り付けられた内容
f・1	「SAVE "」を入力する
f・2	「TOFF」を入力して、リターンキーを押す
f・3	「FORCE」を入力する
f・4	「RFORCE」を入力する
SHIFT + f・5	「QUIT」を入力して、リターンキーを押す
f・6	「COMPILE "」を入力する
f・7	「UPLD」を入力して、リターンキーを押す
f・8	「CONT」を入力して、リターンキーを押す
f・9	「STEP」を入力して、リターンキーを押す
f・10	「EXIT」を入力する

3 - 2

CPUの初期設定

FP-BASICの画面からCPUの初期設定を行うことができます。このとき、パソコンとCPUは、正しく接続されて、CPUのモードスイッチは「PROG.」に設定されていなければなりません（モードスイッチが「REMOTE」に設定されている場合は、CPUがRUN状態でないことを確認してください）。

3-2-1 CPUのバージョン表示

VERコマンドを使用して、接続されているFP3 (H) - BASICタイプCPUのバージョンを確認します。

```
>VER  
FP3 BASIC TYPE(V53/V50)V2_x:File name xxx  
>■
```

注意

「File name xxx」には、ROM運転時のROMに書き込まれているプログラム名が表示されます。

●使用するコマンド

VER	BASICタイプCPUのバージョン名を表示します
-----	--------------------------

3-2-2 CPUのメモリの初期化

接続されているBASICタイプCPUのメモリを初期化します。

CPUメモリの初期化には、VERINIT、CLRVARの2つのコマンドがあり、別にCPUのリセット用にRESETコマンドがあります。なお、VERINITで初期化した場合、パラメータメモリ、I/Oマップ、プログラム、変数、およびパラメータメモリで設定されたI/O、メモリI/Oおよびデータメモリの保持・非保持の属性も初期化されますので、注意が必要です。

```
A>VERINIT  
A>■
```

●使用するコマンド

VERINIT	BASICタイプCPUのすべての状態を初期化します
CLRVER	BASICタイプCPUのプログラムと変数を初期化します
RESET	BASICタイプCPUをリセットします
◆CLRVAL	BASICタイプCPUのグローバル変数を初期化します

3-2-3 パラメータメモリの設定

BASICタイプCPUのシステム構成に応じて、プログラムエリアのサイズ、I/Oおよびメモリの保持・非保持の属性の設定、異常運転時の運転モードの設定、他のPCとのリンク状態の設定などを、パラメータメモリに設定します（「5-2. パラメータメモリ設定一覧 (P.138)」参照）。

パラメータメモリの設定用に、PRM、PRMCLRの2つのコマンドと、PRM () 関数が用意されています。

```
>PRINT PRM(0).....ユーザープログラムの容量
128
>PRM 0,64
>PRINT PRM(0)
64
>PRMCLR
>PRINT PRM(0)
128
>■
```

注意

パラメータメモリを設定するコマンド、関数は、FP - BASICの画面から直接実行できる他、実行プログラムの中に記述することもできます。実行プログラム中に記述すれば、複数のシステム開発におけるプログラムの管理が容易になります。

●使用するコマンドと関数

PRMCLR	パラメータメモリの設定を初期化します
PRM	パラメータメモリを設定します
PRM ()	パラメータメモリの現在の設定値を返します

*PRM、PRMCLRは、プログラム中のステートメントとして記述可能です。

■ BASIC タイプ CPU のイニシャライズ機能

BASICタイプCPUは、動作モードスイッチの操作、およびコマンドの実行により、以下の表のとおり初期化されます。I/Oおよびメモリの内容は、パラメータメモリで、保持・非保持の属性を設定します。

動作モード	RUN	—	○	○		○	○		○	○		○	
	REMOTE	RUN	—	○	○		○	○		○	○		○
		PROG.	—	○		○	○	○	○	○	○	○	○
	PROG.	—	○		○	○	○	○	○	○	○	○	
操作内容／コマンド実行		CPUの電源投入	CPUのイニシャライズスイッチ	PROG.モードへの切り替え	RUNモードへの切り替え	VERINITコマンド	RESETコマンド	CLRVARコマンド	PRMCLRコマンド	SLOTCLRコマンド	DWNLDコマンド	RFFORCE ALLコマンド	
初期化される内容	パラメータメモリ					○			○				
	I/Oマップ					○				○			
	プログラム					○		○			○		
	I/O (出力)	○	○	△	△	○	○						
	メモリI/O	△	△			○	△						
	データメモリ	△	△			○	△						
	リンクメモリI/O	△	△			○	△						
	リンクデータメモリ	△	△			○	△						
	変数	*			*	○		○			○		
	強制入出力	△	○			○	○					○	

○：初期化されます。

△：初期化されます。ただしパラメータメモリで「保持」に設定した領域は初期化されません。

*印は、Ver.2.0以前のFP3 - BASICタイプCPUのみ初期化されます。

3-2-4 変数領域の割り付け

MAPコマンドとFRE () 関数を使用して、タスクごとの変数領域 (メモリサイズ) の割り付けをします。

```
>PRINT FRE(11)
128
>MAP 11,1 .....タスク1の変数領域に1キロバイトを割り付ける
>PRINT FRE(11)
64
>■
```

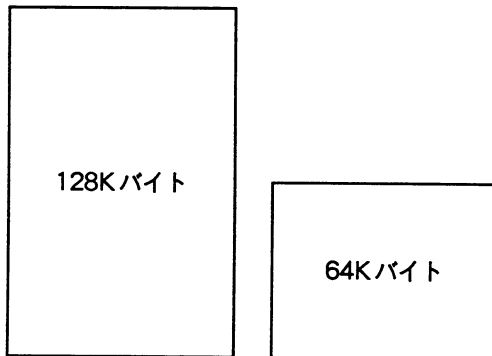
●使用するコマンドと関数

◆ MAP	タスクごとにローカル変数領域を割り付けます
◆ FRE ()	タスクごとのローカル変数領域の空きエリアサイズを返します (16バイト単位)
◆ FREV	グローバル変数の空きエリアサイズを返します (Kバイト単位)

*FREはプログラム中のステートメントとして記述することもできます。

■ BASIC タイプ CPU のメモリ構成とプログラム領域

プログラムエリア

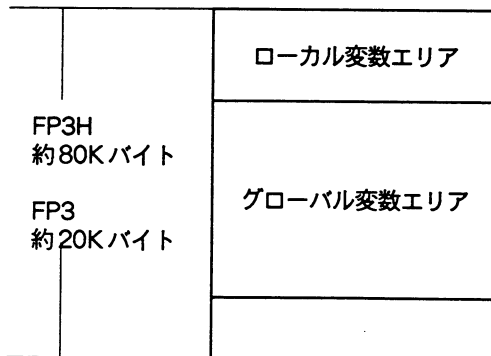


プログラムエリアにはユーザープログラムが格納されます。

- ・コンパイルされた*.OBJファイル
- *.SYMファイル

したがって、*.OBJと*.SYMのファイルがプログラムエリアの容量を越えると、DWNLDでエラーが発生します。また、パラメータメモリのPRM (0) でプログラムエリアとして使用するエリアを小さくすると、残りの部分をファイルメモリ (FL_) として使用できます。

変数エリア



変数エリアは、ローカル変数エリアとグローバル変数エリアにあらかじめ分割されて、割り付けられています。

現在の割り付け状態は、MAP 命令を使用して確認できます。

ローカル変数エリアは、MAP コマンドを使用して、タスクごとに1Kバイト単位で任意に割り付けることができます。

注意

Ver.2.0以降のFP3 - BASICタイプCPUはVer.2.0以前と比べて、グローバル変数エリアが狭くなっています。(約5Kバイト)

3-2-5 スロット割り付け

BASICタイプCPUに接続されているユニット構成に応じて、スロット番号の割り付けをします。スロット割り付けには、フリーロケーション、任意の割り付けがあります。

スロット割り付けには、SLOTCLR、SLOTコマンドとSLOT () 関数が用意されています。

●フリーロケーションの場合

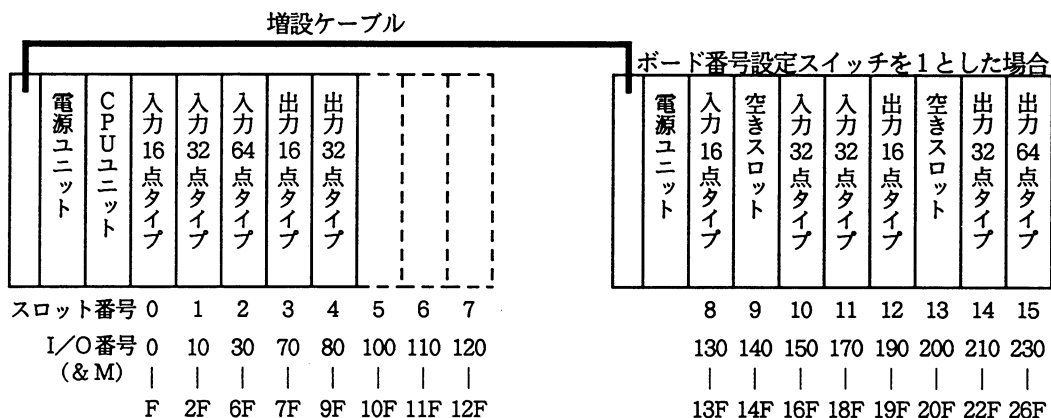
```
>SLOTCLR
>PRINT SLOT(1)
102.....8、16点出力ユニットを意味する
>■
```

- 基本マザーボードおよび増設マザーボードは3スロットタイプ、5スロットタイプも含めすべて、1枚につき8スロット分のI/O番号が設定されています。
- 5スロットタイプの場合、実際には後3スロット分を空きスロットとして扱います。同様に3スロットタイプは後5スロット分を空きスロットとして扱います。
- 空きスロットには16点分のI/O番号が割り付けられます。
- 増設マザーボードも設定番号順にI/O番号が割り当てられます。
- リンクユニット等のI/O番号に関係のないユニットの場合にも空きスロットと同様に16点分I/O番号が割り当てられます。
- 増設マザーボードのボード番号設定スイッチは1~3の範囲で重複のないように設定してください。

例

基本マザーボード (5スロットタイプ)

増設マザーボード



●任意割り付けの場合

```

>SLOT 1,103 ..... スロット番号と機能番号を指定する
>PRINT SLOT(1)
103 ..... 32点出力ユニットを意味する
>■

```

注意

- スロット番号は、CPUに近いものから「0」「1」「2」・・・と付けられます。スロット番号の最大値は23です (No.0～No.23)。ユニットの機能番号については「6-4-3. ユニットのI/O点数と機能番号 (P.152)」を参照してください。
- リモートI/Oシステムのスロット割り付けについては、「4-8. リモートI/Oシステムのスロット割り付け命令 (P.110)」を参照してください。
- スロット割り付けを設定するコマンド、関数は、FP - BASIC IIの画面から直接実行できるほか、実行プログラムの中に記述することもできます。実行プログラム中に記述すれば、複数のシステム開発におけるプログラムの管理が容易になります。

●使用するコマンドと関数

SLOTCLR	スロット割り付けを初期化します (フリーロケーション)
SLOT	スロット割り付けをします
SLOT ()	スロット割り付けの設定値を返します

*SLOTCLR、SLOTは、プログラム中のステートメントとして記述することもできます。

3 - 3

プログラム編集の基本操作

3-3-1 プログラム入力と編集の基本

FP-BASICの編集画面は、スクロール可能なフルスクリーンエディタ仕様になっています。キーボードからの入力は、コンピュータのメモリに書き込まれて、画面に表示されます。プログラム編集時は、CPUを接続しない状態で行うことができます。

新しくプログラムを作成する手順は、以下のとおりです。

- 1) プログラムを作成する前に、NEWコマンドを使用して、パソコン上のプログラムメモリの内容を初期化します。プログラムの編集中に、NEWコマンドを実行すると、メモリ上のプログラムは消去されますので、注意してください。

```
>NEW  
>
```

- 2) AUTOコマンドで行番号の自動発生モードに入ります。AUTOモードは、**CTRL** キーを押しながら **C** キーを押すと解除されます。行番号の割り付けを変更する場合は、RENUMコマンドを使用します。

```
>NEW  
>AUTO  
>10 FUNCTION MAIN.....この行からキー入力を開始  
>20 ST:  
>30 INTEGER A  
>40 FOR A=1 TO 3 STEP 1  
>50     ON Y_&M10  
>60     WAIT 1  
>70     OFF Y_&M10  
>80     WAIT 1  
>90 NEXT A  
>100 END  
>110 FEND  
>120.....この行で「CTRL」+「C」キーを入力する  
>■
```


●使用するコマンド


NEW	パソコンのプログラムメモリを初期化します
AUTO	行番号を自動的に発生します
RENUM	行番号をつけ変えます
CLS	パソコンの画面を初期化（表示を消去）します
LIST	編集中のプログラムを画面に表示します
LLIST	編集中のプログラムをプリンタに印字します












注意

FP-BASICのスクリーンエディタは最大64Kバイトまで編集可能です。それ以上のサイズのプログラムの編集には、市販のテキストエディタを使用しますが、コンパイルによるオブジェクトプログラムにもサイズの制限がありますので注意してください（「3-5. オブジェクトプログラムの作成と転送（P.64）」参照）。

なお、1行のサイズは255文字（バイト）以内です。

- 3) プログラムを入力します。1文字ずつキーボードから入力し、1行の入力が終了したら、 キーを押します。

以下の方法で、入力した文字を修正することができます(「3-1-3. キー操作の概要 (P.47)」参照)。修正後、必ず  キーを押します。


- ・文字修正(重ね書き)の方法
修正する文字の上に、  キーを使用してカーソルを移動させ、正しい文字を入力します。
- ・文字削除の方法
削除する文字の右に、  キーを使用してカーソルを移動させ、 キーを押します。また、 +  キーで、カーソル位置から右側の文字が削除されます。
- ・文字挿入の方法
挿入する個所に、  キーを使用してカーソルを移動させ、 キーを押すと1文字分のスペースが挿入されます。FP - BASICの起動時に「/i」オプションを付けた場合は、 キーは上書き/挿入モードの切り替えになります(「2-2-2. MS - DOSの知識のある方のために、●起動時のコマンドオプションについて (P.23)」を参照してください)。

- 4) LISTコマンドで入力したプログラムを表示させます。パソコンのプログラムメモリに記憶されているプログラムが表示されます。

```

>LIST
10 FUNCTION MAIN
20 ST:
30 INTEGER A
40 FOR A=1 TO 3 STEP 1
50     ON Y_&M10
60     WAIT 1
70     OFF Y_&M10
80     WAIT 1
90 NEXT A
100 END
110 FEND
>

```

- 5) 行の挿入と削除
- ・たとえば、プログラムの行番号20と30の間に行を追加したければ、10と20の間の任意の行番号で新しい行を入力します。
 - ・たとえば、行番号15の行を削除したければ、行番号15だけを入力し、 キーを押します。

3-3-2 行の移動

LINEMV コマンドを使用すれば、行の範囲を指定しての、複写と削除ができます。

この場合、行番号は、自動的に振り直されます（新しい行番号の増分を指定することもできます）。

```

>LIST
10 FUNCTION MAIN
20 ST:
30 INTEGER A
40 FOR A=1 TO 3 STEP 1
50 ON Y_8M10
60 WAIT 1
70 OFF Y_8M10
80 WAIT 1
90 NEXT A
100 END
110 FEND
>LINEMV 30,30,11 .....30行目を11行目に移動
>LIST
10 FUNCTION MAIN
11 INTEGER A
20 ST:
40 FOR A=1 TO 3 STEP 1
50 ON Y_8M10
60 WAIT 1
70 OFF Y_8M10
80 WAIT 1
90 NEXT A
100 END
110 FEND
>■

```

●使用するコマンド

◆LINEMV 行を移動します

3-3-3 プログラム中の文字列検索

FINDコマンドを使用すれば、プログラム中の任意の文字列を検索できます。検索文字列には、ワイルドカード（「*」「?」）が使用できます。

```

>LIST
10 FUNCTION MAIN
20   INTEGER A
30   ST:
40   FOR A=1 TO 3 STEP 1
50     ON Y_&M10
60     WAIT 1
70     OFF Y_&M10
80     WAIT 1
90   NEXT A
100 END
110 FEND
>FIND WAIT
60     WAIT 1 ..... マッチした行を表示する
80     WAIT 1
>■

```

●使用するコマンド

◆FIND 編集中のプログラムから文字列を検索します

3 - 4

ファイルのセーブとロード

FP-BASICでは、プログラムファイルをセーブまたはロードすることができ、これにより能率的なプログラム管理が可能です。プログラムファイルのセーブまたはロードは、CPUを接続しない状態で行うことができます。

3-4-1

プログラムのセーブ (保存)

DSAVEコマンドで、編集したプログラムをフロッピーディスク上のファイルにセーブ (保存) します。DSAVEコマンドは、省略してSAVEと入力することもできます。

```
>LIST
10 FUNCTION MAIN
20   INTEGER A
30   ST:
40   FOR A=1 TO 20 STEP 1
50     ON Y_&M10
60     WAIT 1
70     OFF Y_&M10
80     WAIT 1
90   NEXT A
100 END
110 FEND
>DSAVE "TEST".....「TEST.PRG」というファイル名でプログラムをセーブする
Over Write(Y/N) :Y.....すでに同名ファイルが存在するときに表示されます
>■
```

*セーブされたファイルは、拡張子「.PRG」のASCIIファイルです。

MS-DOSのTYPEコマンドや市販のテキストエディタ (Mifesなど) で読めます。

3-4-2

プログラムのロード (読み出し)

DLOADコマンドで、ディスク上のファイルからプログラムを読み出し、メモリにロードします。DLOADコマンドは、省略してLOADと入力することもできます。

```
>NEW
>DLOAD "TEST".....「TEST.PRG」のプログラムをロードする
>LIST
10 FUNCTION MAIN
20   INTEGER A
30   ST:
40   FOR A=1 TO 20 STEP 1
50     ON Y_&M10
60     WAIT 1
70     OFF Y_&M10
80     WAIT 1
90   NEXT A
100 END
110 FEND
>■
```

*ロード可能なファイルは、拡張子「.PRG」のASCIIファイル (テキストファイル) です。これは、市販のテキストエディタ (Mifesなど) で作成できます。

3-4-3

ファイル管理

FP - BASICには、ディスク上のプログラムファイルを管理するために、FILES、NAME、KILL コマンドを用意しています。

● FILES

ディスク上のファイルの一覧を表示します。

```
>FILES
FPB. EXE      SPEED. COM    RSDRV. SYS    PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT TEST. PRG
```

● NAME

ファイル名を変更します。

```
>NAME "TEST" AS "TEST_2"
>FILES
FPB. EXE      SPEED. COM    RSDRV. SYS    PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT TEST_2. PRG
```

● KILL

ディスク上の指定したファイルを消去します。

```
>KILL "TEST_2"
Sure(Y/N) :Y.....確認メッセージ
>FILES
FPB. EXE      SPEED. COM    RSDRV. SYS    PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT
```


● 注意

ファイル名は、「"」（ダブルクォーテーション）」で囲んでください（最後の「"」（ダブルクォーテーション）」は省略することも可能です）。

●使用するコマンド

DSAVE	パソコンのプログラムメモリからディスクにプログラムを書き込みます
DLOAD	ディスクからパソコンのプログラムメモリにプログラムを読み出します
FILES	ディスクに書き込まれているファイルを一覧表示します
NAME	ディスクに書き込まれているファイルの名前を変更します
KILL	ディスクに書き込まれているファイルを削除します

3-4-4 ファイル選択ウィンドウの使用方法

ディスクファイルのセーブ、ロードを行うコマンドを使用するとき、ファイル名を入力しないで  キーだけを押ししたり、ディレクトリパス名やワイルドカード「*」「?」を入力した場合、ファイル選択ウィンドウが表示されます。

1) コマンドの入力

```
>FILES
FPB. EXE      SPEED. COM    RSDRV. SYS    PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT TEST. PRG
>DLOAD
```


ファイル名に、ワイルドカード「」「?」、ディレクトリパス名を入力することもできます。

2) ファイル名入力ウィンドウの表示


ファイル名を省略してコマンドを入力すると、ファイル名入力ウィンドウが表示されます。

```
INPUT LOAD FILE NAME
_____

*.* *.*.PRG *.*.PRM _____
```

ファイル名の他に、ワイルドカード「」「?」、ディレクトリパス名が入力できます。
入力なしに  キーを押すと、「.PRG」が入力されたものとみなします。





3) ファイル選択ウィンドウの表示

さらに、ファイル名を省略して  キーを押すと、ファイル選択ウィンドウが表示されます。

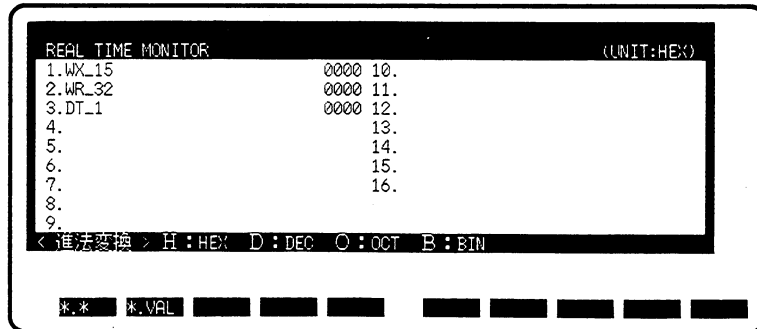
・ DSAVE、DLOADなどの場合

```
TEST.PRG
CURRENT DIRECTRY = A:¥
PATH = A:¥
TEST.PRG 10 'TEST PROGRAM No.1
TEST_2.PRG 10 'TEST PROGRAM No.2
TEST_3.PRG 10 'TEST PROGRAM No.2

*.* *.*.PRG *.*.PRM _____
```

*   キーで選択し、 キーを押すとファイル名が入力されます。
* 拡張子が「*.PRG」の場合、ソースファイルの先頭行の内容が表示されます。
* 直接入力する場合は、ファイル名の他に、ワイルドカード「*」「?」、ディレクトリパス名が入力できます。
* 入力なしに  キーを押すと、「*.PRG」が入力されたものとみなします。

・SVIO、LDIO、SVVAL、LDVAL、SVCR、LDCRなどの場合



* キーで選択し、 キーを押すとファイル名が入力されます。

* 直接入力する場合は、ファイル名の他に、ワイルドカード (「*」「?」)、ディレクトリパス名が入力できます。

* 入力なしに キーを押すと、「*.IO」「*.PRG」「*.CR」が入力されたものとみなします。

注意

キー入力の代わりに、以下のファンクションキーが使用できます。

- DSAVE、DLOADの場合 : 「*. *」 「*. PRG」 「*. PRM」
- SVIO、LDIOの場合 : 「*. *」 「*. X」 「*. Y」
 「*. R」 「*. DT」 「*. L」
- SVVAL、LDVALの場合 : 「*. *」 「*. VAL」
- SVCR、LDCRの場合 : 「*. *」 「*. CMN」

3 - 5

オブジェクトプログラムの作成と転送

BASICタイプCPUは、FP - BASICで編集したプログラム（ソースプログラム）をそのままの形で理解し実行することはできません。ソースプログラムをコンパイルして、CPUが理解できるオブジェクトプログラムを生成し、それをCPUにダウンロード（転送）します。このとき、CPUのモードスイッチは、「PROG.」に設定されていなければなりません（モードスイッチが「REMOTE」に設定されている場合は、CPUがRUN状態でないことを確認してください）。ただし、コンパイルは、CPUと接続しなくても可能です。

3-5-1 コンパイル

COMPILEコマンドで、FP - BASICのメモリ上のプログラムから、CPUで実行できるオブジェクトプログラムを生成し、ディスク上のファイルにセーブします。COMPILEコマンドは、フロッピーディスク上のプログラムファイルから直接オブジェクトファイルを生成することもできます。また、ROMライターで使用するインテルHEXフォーマットでオブジェクトプログラムを作成することもできます。

```
>FILES
FPB. EXE      SPEED. COM    RSDRV. SYS    PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT TEST. PRG
>NEW
>DLOAD "TEST"..... プログラムをメモリにロードする
>LIST
10 FUNCTION MAIN
20   INTEGER A
30   ST:
40   FOR A=1 TO 3 STEP 1
50     ON Y_&M10
60     WAIT 1
70     OFF Y_&M10
80     WAIT 1
90   NEXT A
100 END
110 FEND
>COMPILE..... コンパイルする : ディスク上に TEST. OBJ と TEST. SYM が作成される。
COMPILE END
>FILES
FPB. BAT      SPEED. COM    RSDRV. SYS    PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT TEST. PRG      TEST. OBJ
TEST. SYM
>■
```

●使用するコマンド

COMPILE	ソースプログラムをオブジェクトプログラムに変換します
DWNLD	オブジェクトプログラムをパソコンのプログラムメモリからCPUに転送します
UPLD	オブジェクトプログラムをCPUからパソコンのプログラムメモリに転送します

注意

FP - BASICの最大プログラムサイズは128Kバイト（*.OBJ+*.SYM）です。したがって、*.OBJファイルと*.SYMファイルの合計が128Kバイトを越えるようなソースファイルはコンパイルできません。

3-5-2 ダウンロード

DWNLDコマンドを使用して、オブジェクトプログラムをCPUに転送します。

```
>DWNLD.....ディスク上の TEST .OBJと TEST.SYMをCPUに転送する
>
```

3-5-3 アップロード

UPLDコマンドを使用して、BASICタイプCPUのメモリ上のオブジェクトプログラムを読み出し、さらに逆コンパイルしてソースプログラムを再現します。ファイル名を指定することにより、再現したソースプログラムを直接ディスクにセーブすることもできます。なお、アップロードにより得られるファイルは、拡張子「.PRM」のASCIIファイルと拡張子「.BIN」のバイナリファイルです（それぞれ*.PRGと*.OBJと対応します）。

```
>NEW.....メモリをクリア
>UPLD "TEST_1".....アップロード
>FILES
FPB. BAT      SPEED. COM      RSDRV. SYS      PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT   TEST. PRG       TEST. OBJ       3つのファイル
TEST. SYM     TEST_1. PRM     TEST_1. BIN     TEST_1. SYM.....が作成される
>LIST
10 FUNCTION MAIN
20 INTEGER A
30 '
40 FOR A=1 TO 3 STEP 1
50 ON Y_&M10
60 WAIT 1.
70 OFF Y_&M10
80 WAIT 1.
90 NEXT A
100 END
110 FEND
>■
```

注意

- コンパイルして得られるファイルは、拡張子「*.OBJ」のオブジェクトファイルと拡張子「*.SYM」のシンボリックファイルです。
- アップロードして得られるファイルは、拡張子「*.PRM」のソースファイルと「*.BIN」のオブジェクトファイルです。シンボリックファイルがディスクのカレントディレクトリに存在しない場合は、拡張子「*.SYM」のシンボリックファイルを生成します。
- 分割コンパイルについては、「第5章 分割コンパイル」を参照してください。

3 - 6

プログラムの実行

BASICタイプCPUをリモートモードに設定すると、FP-BASICからCPUの動作を制御することができます (CPUのリモート実行)。このとき、CPUはリモートモード (REMOTE) に設定しておきます。

3-6-1 プログラムの実行

プログラムの実行、停止、再開、終了には、XQT、HALT、RESUME、QUITコマンドを使用します。

```
>XQT.....プログラムを実行 (タスク1)
>HALT.....プログラムの実行を一時停止する
>RESUME.....プログラムの実行を再開する
>QUIT.....プログラムの実行を終了する
>■
```

●使用するコマンド

XQT	BASICタイプCPUのプログラムを実行します
HALT	実行中のタスクを一時停止します
RESUME	一時停止中タスクの実行を再開します
QUIT	実行中または一時停止中のプログラムを終了します

*XQT、HALT、RESUME、QUITは、プログラム中でステートメントとしても記述できます。

3-6-2 トレースモード

TON、TOFFコマンドを使用して、実行中の行番号をリアルタイムで画面に表示するモード (トレースモード) に設定します。

```
>TON 11.....タスク1をトレースする
>XQT
[1:00010].....タスク番号と実行中の行番号を表示
[1:00020]
[1:00030]
[1:00040]
[1:00050]
:
:
:
>TOFF.....トレースモードを解除する
>■
```

●使用するコマンド

TON	実行中の行番号を画面に表示します (トレースモード)
TOFF	TONコマンドで設定したトレースモードを解除します

3-6-3 テストモード実行

BASICタイプCPUをテストモードに設定すれば、プログラム中にPAUSE命令を記述することによって、テストモードでの実行ができます。CPUのイニシャライズ/テストスイッチをテスト側に倒します。テストモードでプログラム実行を一時停止するためには、プログラム中にPAUSE命令を記述します。TESTコマンドでテストモードの内容を設定し、一時停止したタスクの再開にはCONTおよびSTEPコマンドを使用します。

```

>LIST
10 FUNCTION MAIN
20   PAUSE
30   LOOP:
40     ON Y_&M10
50     WAIT 1
60     OFF Y_&M10
70     WAIT 1
80   GOTO LOOP
90 FEND
>COMPILE
>DWNLD
>TEST
0.....現在の実行モードがシュミレーションモードであることを表示します
>TEST 1.....実行モードをテストモードに設定
>XQT.....実行する (タスク 1)
>CONT.....PAUSE 命令実行でプログラム実行が一時停止するので
>■

```

●使用するコマンド

TEST	テストモードの設定、および表示をします
CONT	PAUSE 命令により一時停止しているタスクの実行を再開します
STEP	PAUSE 命令により一時停止しているタスクの実行をステップモードで再開します

●使用する命令

PAUSE	テストモード時、実行中のタスクを一時停止します
-------	-------------------------

*PAUSE命令は、プログラム中に記述します。

注意

タスク番号には、1~16の値を指定します。

3 - 7

実行中のプログラムのモニタ

FP - BASICには、BASICタイプCPUの動作状態をモニタする機能と、CPUのI/O、メモリ、および実行中のプログラムの変数値を強制的にセットする機能が用意されています。これらのモニタ機能を使用すれば、作成したプログラムのデバッグ、現場調整が能率的に行えます。モニタ機能を使用するとき、CPUはRUN状態にあります（CPUのモードスイッチは「RUN」または「REMOTE」に設定されます）。

3-7-1 CPUの動作モードの表示

MODEコマンドを使用して、CPUの現在の動作モードを確認します。

```
>MODE  
TEST RUN(RAM).....テストランモード (RAM上のプログラムが実行対象)  
>
```

●使用するコマンド

MODE	BASICタイプCPUの動作モードを画面に表示します
------	----------------------------

3-7-2 タスクごとのステータス表示

TSTATコマンドを使用することにより、タスクごとの現在のステータス（状態）を表示します。

```
>TSTAT  
task1 quit 0010  
task2 wait 0050  
task3 run 0100  
task4 halt 0150  
:  
:  
:  
task14 quit 0200  
task15 quit 0250  
task16 quit 0300  
>■
```

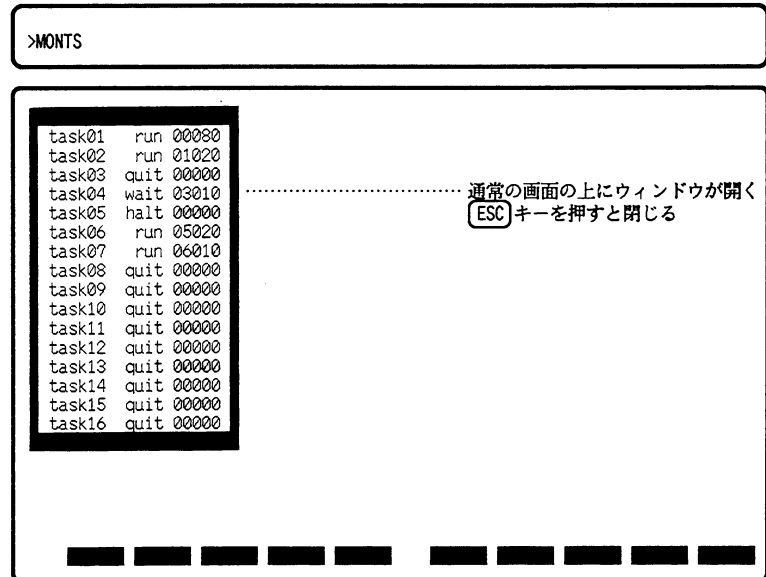
●使用するコマンド

TSTAT	タスクごとのステータスを表示します
-------	-------------------

3-7-3

タスクごとのステータスのリアルタイム表示

MONTSコマンドを使用することにより、タスクごとの現在のステータス（状態）をウィンドウ形式で表示します。



●使用するコマンド

◆MONTS タスクごとのステータスをリアルタイムに表示します

3-7-4 I/O、メモリおよび変数のモニタ

MONコマンドを使用して、I/O、メモリの状態および変数の値をリアルタイムでモニタします。MONコマンドでモニタできるのは、I/O、メモリI/O、データメモリ、リンクメモリI/O、リンクデータメモリ、およびグローバル変数です。

● I/Oのモニタ

```

>MON A
MINITOR START
0
MONITOR END ..... (ESC)キーまたはリターンキーを押してモニタ終了
>■

>MON SW(Y_8M10)..... ビット単位で指定
MINITOR START
0 ..... 表示はリアルタイムに変化する
MONITOR END
>MON INW(WY_1)..... ワード単位で指定
MONITOR START
0000 H ..... (H)キーを押すと16進表示
MONITOR END
>MON IND(WY_1)..... ダブルワード単位で指定
MONITOR START
00000000 H
MONITOR END
>■
    
```

* モニタ中に (B) (O) (H) (D) キーを押すと、それぞれ2進、8進、16進、10進表示に切り替わります。

● グローバル変数のモニタ

```

>MON A..... 変数Aを指定
100 ..... 表示はリアルタイムに変化する
>■
    
```

* モニタ中に (B) (O) (H) (D) キーを押すと、それぞれ2進、8進、16進、10進表示に切り替わります。

注意

モニタできる変数は、タスク間でグローバルな変数に限られます (タスクの中でだけ有効なローカル変数はモニタできません)。詳しくは、「4-4-3. グローバル変数とローカル変数 (P.86)」を参照してください。

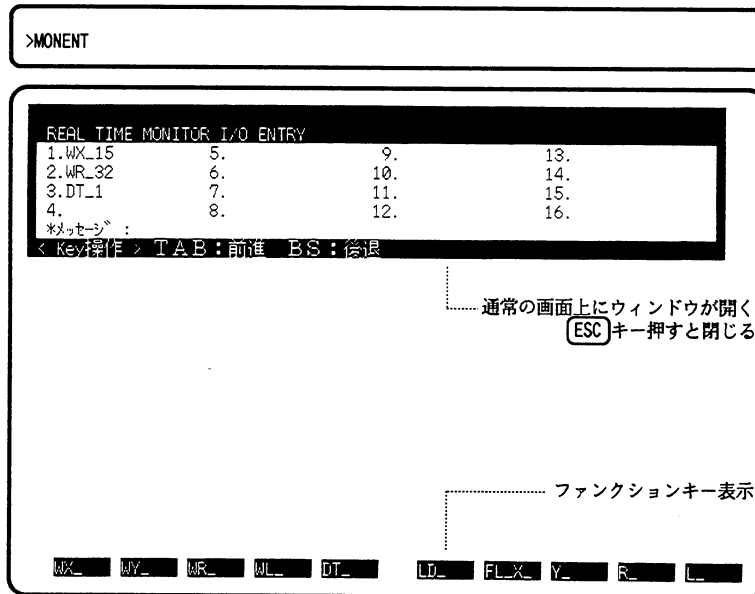
● 使用するコマンド

MON	I/O、メモリおよび変数の値をリアルタイムに表示します
-----	-----------------------------

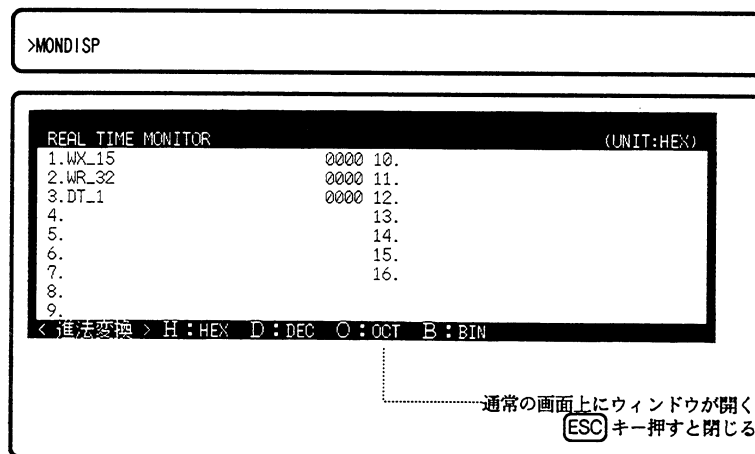
3-7-5 I/Oおよびメモリのマルチポイントモニタ

MONENT、MONDISPコマンドを使用して、複数のI/Oおよびメモリの状態を同時にリアルタイムでモニタします。

■登録する



■モニタの開始



*モニタ中に [B] [O] [H] [D] キーを押すと、それぞれ2進、8進、16進、10進表示に切り替わります。

●使用するコマンド

- ◆MONENT リアルタイムでモニタする複数のI/Oおよびメモリを登録します
- ◆MONDISP 複数のI/Oおよびメモリをリアルタイムでモニタします

3-7-6 I/Oおよびメモリのダンプ (状態表示)

DUMPコマンドを使用して、範囲指定したI/Oおよびメモリの内容を表示します (ダンプリストを表示します)。DUMPコマンドで内容を表示できるのは、I/O、メモリI/O、データメモリ、リンクメモリI/O、リンクデータメモリです。

- ビット単位の表示X_Y_R_L_ を指定

```
>DUMP X_6,X_20
I/O      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
X_800000          1 0 1 0 0 0 0 0 0 1
X_800010 1 0 1 1 0
>
```

- バイト単位の表示WX_WY_WR_WL_DT_LD_ を指定

```
>DUMP WX_6,WX_7,B
I/O      +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F [ A S C I I ]
X_800000:          4C 4D 4E 4F          LMNO
>
```

- ワード単位の表示WX_WY_WR_WL_DT_LD_ を指定

```
>DUMP WX_6,WX_7,W
I/O      +0 +1 +2 +3 +4 +5 +6 +7
X_800000:          4D4C 4F4E
>
```

- 使用するコマンド

DUMP	I/Oおよびメモリの内容を読み出して、表示します
------	--------------------------

3-7-7 タイマ経過値の表示および設定

TIME コマンド、TIME () 関数を使用すれば、WAIT 命令で使用するタイマの経過値を表示、設定することができます。経過値を任意に設定することにより、WAIT 命令の動作確認などプログラムの調整が円滑になります。

なお、タイマは、各タスクに1つずつ使用できます（各タスクで同時に実行できる WAIT 文は1つだけです）。

```
>TIME 11,200..... タイマ経過値を200秒に設定する
>PRINT TIME(11)..... タイマ経過値を確認する
200
>■
```

●使用するコマンドと関数

TIME	タイマ経過値を設定します
TIME ()	タイマ経過値を返します (秒単位)

3-7-8 高機能ユニットの割り込み設定の表示

DUMPINT コマンドを使用すれば、高機能ユニットで使用している割り込みの割り込み番号を表示することができます。

```
>DUMPINT
Interrupt No. 0 Disable No req.
>■

>DUMPINT
Interrupt No. 0 Enable No req.
>■
```

●使用するコマンド

◆ DUMPINT	割り込み設定状況の一覧を表示します
-----------	-------------------

3-7-9 I/Oおよびメモリその他の状態の一括ダンプ (状態表示) と編集

DUMPCおよびDUMPPコマンドを使用して、I/O、メモリ、その他の内容を読み出して、表示および内容の変更をします。I/O、メモリ、その他の内容は、種類別に一括して読み出し、スクロールが可能なスクリーン表示をします。

■コマンドの入力

```
>DUMPC
```

■ダンプ対象の選択

```

1. 入力                10. 強制出力
2. 出力                11. 強制メモリ/I/O
3. メモリ/I/O         12. 強制リンク・メモリ/I/O
4. リンク・メモリ/I/O 13. パラメータ・メモリ
5. データ・メモリ     14. 共有RAM
6. リンク・データ・メモリ 15. 知っ設定
7. 特殊メモリ・I/O   16. 割り込み設定
8. 特殊データ・メモリ 17. ID番号設定
9. 強制入力
   処理番号を入力して下さい ==>
*メッセージ :
```

* DUMPP コマンドでは、「7. SPECIAL MEMORY I/O」「8. SPECIAL DATA MEMORY」は実行できません。

●使用するコマンド

◆DUMPC	I/O、メモリ、その他の内容を一括して読み出して、スクリーン表示します
◆DUMPP	I/O、メモリ、その他の内容を一括して読み出して、スクリーン表示および編集をします

■ウィンドウダンプ

ウィンドウダンプ画面では、↑ ↓ キーでスクロールが可能です。また、DUMPP コマンドでは、← → TAB BS キーで訂正個所にカーソルを移動し、任意のキー入力ができます。ウィンドウダンプを終了させるには、ESC キーを押します。

●X(WX_) Y(WY_) R(WR_) L(WL_) DT_ のダンプ (状態表示)

```

          入力I/O 状態表示
          HEX      DEC          BIN      ASC
0000      0000      00000      0000 0000 0000 0000      .
0001      0000      00000      0000 0000 0000 0000      .
0002      0000      00000      0000 0000 0000 0000      .
0003      0000      00000      0000 0000 0000 0000      .
```

● X(WX_) Y(WY_) R(WR_) L(WL_) の強制入出力の状態表示

フォース	入力I/O 状態表示															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
&M000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
&M000010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
&M000020	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
&M000030	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

● パラメータメモリの状態表示

パラメータメモリ 状態表示			
	DATA	MIN	MAX
00007	00060	(0 ~ 98)	
00008	00000	(0 ~ 2048)	
00009	00000	(0 ~ 24576)	
00010	00000	(0 ~ 64)	

● 共有メモリのダンプ (状態表示)

共有RAM 状態表示																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	00	00	00	00	00	00	00	00	00	FF	FF	00	FF	00	FF	00
0010	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

● スロット割り付けの状態表示

スロット割り付け 状態表示				
	I/O種類	入力点数	出力点数	共有RAM
000000	汎用	00016	00000	無
000001	汎用	00000	00016	無
000002	汎用	00016	00016	有
000003	汎用	00000	00016	無

● 割り込み番号割り付けの状態表示

割り込みユニット 割り付け状態 表示			
割込番号	スロット番号	割込番号	スロット番号
0 ~ 7	—	19	—
8 ~ 15	—	20	—
16	—	21	—
17	—	22	—
18	—	23	—

3-7-10 強制入出力

モニタ機能を使用してプログラムのデバッグを行う場合、一般に外部の入出力機器との接続は困難です。このようなとき、FORCE、RFORCEコマンドを使用して、I/Oを強制的にON/OFFします。

```

>PRINT SW(Y_&M10)
0
>FORCE Y_&M10=1.....Y_&M10を強制的にONする
>FORCE
Y_:17.....Y_&M10 が強制ON/OFF状態にあることを表示する
>PRINT SW(Y_&M10)
1
>RFORCE Y_&M10.....Y_&M10の強制モードを解除する
>PRINT SW(Y_&M10)
1.....Y_&M10はONのまま
>FORCE.....表示なし
>■
    
```

●使用するコマンド

FORCE	I/O、メモリの強制セットおよびリセットを設定します
RFORCE	I/O、メモリの強制セットおよびリセットを解除します

3-7-11 I/O、メモリ、変数のセーブとロード

I/O、メモリ、変数、共有メモリの内容をファイルにセーブ、ロードすることができます。これらのコマンドとI/Oまたはメモリへの一括書き込み、I/O間のデータ転送を使用すれば、より効率的なデバッグ作業が可能です（「4-7-4. II/Oポートアクセス命令 (P.107)」参照）。

```

>SVIO 0,2,"TEST.Y".....「TEST.Y」というファイル名で出力I/Oをセーブする
Over Write(Y/N) :Y.....すでに同名ファイルが存在するときに表示されます
>FILES
FPB.EXE      SPEED.COM    RSDRV.SYS    PRINT.SYS
CONFIG.SYS   AUTOEXEC.BAT TEST.PRG      TEST.OBJ
TEST.SYM     TEST.Y
>■
    
```

●使用するコマンド

◆SVIO	I/Oをセーブします
◆LDIO	I/Oをロードします
◆SVVAL	グローバル変数をセーブします
◆LDVAL	グローバル変数をロードします
◆SVCR	共有メモリをセーブします
◆LDCR	共有メモリをロードします

*ファイル選択ウィンドウについては、「3-4-4. ファイル選択ウィンドウの使用方法 (P.62)」を参照してください。

第 4 章

4

FP - BASIC の基礎知識

こ の 章 の 内 容

4-1. BASICプログラムの基本ルール	78	4-7. 一般命令と組み込み関数	106
4-1-1. 直接実行と間接実行	78	4-7-1. N88BASIC一般命令	106
4-1-2. プログラムの例	78	4-7-2. ディスクファイルアクセス命令	106
4-1-3. 命令文	78	4-7-3. 変数操作命令	106
4-1-4. コメント(注釈文)	79	4-7-4. I/Oポートアクセス命令	107
4-1-5. 行と行番号	79	4-7-5. タイマ命令と関数	107
4-1-6. ラベルとラベル名	79	4-7-6. システム変数とシステム関数	107
4-2. 使用できる文字と記号	80	4-7-7. 組み込み関数	108
4-2-1. 使用できる文字と記号	80	4-8. リモートI/Oシステムの	
4-2-2. FP - BASICの予約語	81	スロット割り付け命令	108
4-3. 定数	82	4-9. MEWNET データ転送命令	112
4-3-1. 定数の種類	82	4-10. 高機能ユニット専用命令	114
4-3-2. 数値定数	83	4-10-1. 通常スロットの	
4-3-3. 文字定数	83	高機能ユニット入出力命令	114
4-4. 変数	84	4-10-2. リモートI/Oシステムの	
4-4-1. 変数名	84	高機能ユニット入出力命令	114
4-4-2. 変数の型と値	84	4-10-3. D/A変換ユニット専用制御命令	115
4-4-3. グローバル変数とローカル変数	96	4-10-4. A/D変換ユニット専用制御命令	115
4-4-4. 配列	97	4-10-5. シリアルデータユニット	
4-4-5. 型変換	98	専用制御命令	116
4-5. 式と演算子	89	4-10-6. パルス出力ユニット専用制御命令	116
4-5-1. 式の種類	89	4-10-7. 高速カウンタユニット専用制御命令	117
4-5-2. 演算子の優先順位	89	4-10-8. 位置決めユニット専用制御命令	117
4-6. FP - BASIC の制御構文	90		
4-6-1. マルチタスクとFUNCTION文	90		
4-6-2. WAIT 命令によるインターロック	92		
4-6-3. 分岐命令 GOTO	94		
4-6-4. サブルーチン呼び出し命令 GOSUB	95		
4-6-5. ON GOTO/GOSUB文	96		
4-6-6. SELECT ~ CASE文	97		
4-6-7. IF/IFB文	98		
4-6-8. ループ文	100		
4-6-9. CALL ~ FUNCTION文	102		
4-6-10. 割り込み制御	103		

4 - 1

BASIC プログラムの基本ルール

4-1-1 直接実行と間接実行

コマンドを直接キーボードから入力して実行することを「直接実行」と呼び、行番号を使用して記述された一連のプログラムを実行することを「間接実行」と呼びます。

このマニュアルでは、直接実行で使用する命令群を「コマンド」という呼び方で統一しています。また、これと区別するため、間接実行で使用する命令群を「命令」と呼んでいます。

なお、間接実行されるプログラム中で使用される命令や関数、式などを合わせて、命令語（ステートメント）と呼びます。

4-1-2 プログラムの例

間接実行されるプログラムの例を示します。プログラムには、行番号が付けられています。

```
10 FUNCTION MAIN
20   INTEGER A
30   ST:
40   FOR A=1 TO 3 STEP 1
50     ON Y_3M10
60     WAIT 1
70     OFF Y_3M10
80     WAIT 1
90   NEXT A
100  END
110  FEND
```

4-1-3 命令文

命令文は、プログラムの中で実行される最小単位の記述で、以下の種類があります。

- 1) 式または関数 例：A = 3 B = SW (X_0)
- 2) 命令語 例：END
- 3) 命令語 + オペランド 例：ON Y_32

命令語（ステートメント）は、単独で使用される場合と、命令語の後にオペランドを伴う場合があります。オペランドには、定数、変数、演算子、関数およびそれらの組み合わせを用います。

4-1-4 コメント（注釈文）

命令文以外の文もあります。REM文はプログラムの記述における注記・注釈のためのコメントであり、何も動作しません。REMの代わりに、「'」（シングルクォーテーション）」を使用することもできます。

```
10 FUNCTION MAIN
20 REM 'テストプログラム
30   INTEGER A
40   ST:
50   FOR A=1 TO 3 STEP 1
:
```

```
10 FUNCTION MAIN
20 'テストプログラム
30   INTEGER A
40   ST:
50   FOR A=1 TO 3 STEP 1
:
```

4-1-5 行と行番号

行とは、プログラムが間接実行される場合に、順次実行される命令文の集まりのことです。プログラムの1行中には、1つあるいは複数の命令文が記述されます（1行に複数の文が記述されることを複文またはマルチステートメント呼びます）。

- 1) 行番号と文との間には必ず1文字以上のスペースを入れます。
- 2) マルチステートメントでは命令文を「;（セミコロン）」を使用して区切ります。
- 3) 行番号は、1～32767までの整数で記述します。
- 4) プログラムは、任意の行番号で開始することができます。
- 5) 行番号は行ごとに大きな値になりますが、これを行番号の増分と呼びます。

注意

- FP – BASICで扱える1行の長さは255バイトまで、編集可能なソースファイルのサイズは64Kバイトまでです。
- FP – BASICで編集したソースプログラムは、コンパイルされてオブジェクトプログラムになってから、CPUにダウンロードされます。したがって、実際にCPUのメモリ内に記憶され、実行されるプログラムは、ソースプログラムとは異なります。

4-1-6 ラベルとラベル名

GOTO文、GOSUB文、その他の制御文を使用してプログラムの実行順序を変更するとき、直接行番号を指定する代わりに、ラベルを指定することができます。行番号ではなくラベルを使用する方が、可読性の高いプログラムになるとともに、後で修正しやすいプログラムになります。

FP – BASICのラベル名には、8文字までの英数字が使用でき、先頭の1文字は必ず英字で書き始めなければならないという約束があります（変数名と同じ規則）。なお、FP – BASICでは、英字の大文字と小文字は区別されませんので注意してください。

また、分岐先のラベル（ラベル行）ではラベル名の後に必ず「:（コロン）」を付けます（GOTOやGOSUBの後にラベル名を記述する場合など、分岐先を指示するときは、「:」は不要です）。

注意

FP – BASICのラベル名には、予約語を使用しないでください。また、予約語の直後が数字または「_」（アンダースコア）」の文字列も、ラベル名に使用できません（「4-2-2. FP – BASICの予約語（P.81）」参照）。

4 - 2

使用できる文字と記号

4-2-1 使用できる文字と記号

FP-BASICでは、1バイト文字の「英字 (大文字・小文字)」「数字」「カタカナ」と、「特殊記号」が使用できます。1バイト文字の「英字」と「記号」には、特別の意味をもつものや、あらかじめ決められた意味を持つ予約語がありますので、使用にあたっては十分に注意してください。

また、FP-BASICでは、2バイト文字の「英字 (大文字・小文字)」「数字」「カタカナ」、「特殊記号」、「ひらがな」「漢字」が使用できます。ただし、2バイト文字で命令語や変数を記述することはできません。2バイト文字の使用は、文字定数とコメントに限られます。

■文字の種類

文字の種類	例
英字	A B C D E ... a b c d e ...
数字	0 1 2 3 4 5 6 7 8 9
特殊記号	! " # \$ % & ' () , . / ; : [] ¥ ^ _
カタカナ	ア イ ウ エ オ ...
ひらがな	あ い う え お ...
漢字	松下電工株式会社

*特殊記号のうち「" (ダブルクォーテーション)」は文字定数として使用できません。

■特殊記号

記号	読み方	特別の用途 (使用時に注意が必要です)
-	マイナス	演算、範囲指定
:	コロン	ラベル行
;	セミコロン	マルチステートメント (複文) の区切り記号
'	シングルクォーテーション	REM (コメント) の省略型
"	ダブルクォーテーション	文字定数
,	コンマ	パラメータの区切り
	スペース	行番号、命令語、パラメータの区切り
.	ピリオド	ファイル名と拡張子の区切り
_	アンダースコア	なし
*	アスタリスク	ファイル名、拡張子のワイルドカード指定
?	クエスション	なし

*特殊記号のうち「" (ダブルクォーテーション)」は文字定数として使用できません。

注意

- ひらがな・漢字などの2バイト文字の入力方法については、使用するかな漢字変換のマニュアルをご覧ください。
- 「_ (アンダースコア)」は、変数名、ラベル名の先頭には使用できません。

4-2-2 FP – BASICの予約語

FP – BASICの予約語は、以下のとおりです。

FP – BASICの変数名、ラベル名には、予約語を使用しないでください。また、予約語の直後が数字または「_ (アンダースコア)」の文字列も、変数名、ラベル名に使用できません。なお、FP – BASICでは、英字の大文字と小文字の区別をしませんので、変数名、ラベル名の重複に注意してください。

()	*	+	-	/	<	<=
<>	=	=<	=>	>	>=	X_	Y_
R_	L_	WX_	WY_	WR_	WL_	DT_	LD_
FL_	ABS()	ACCEL	ADALM()	ADASET	ADAVRG	ADDATA	ADDATA()
ADERR	ADLIMIT	ADLSET	ADRDY()	ADSCALE	ADRSET	AND	ASC()
ATAN()	ATN()	BCD()	BIN\$()	BIN()	BLKMOV	BLKOUT	BYTE
CALL	CASE	CHR\$()	CLEAR	CLOSE	CLRLIB	CLRVAL	CLRVAR
CLS	COLOR	COMPILE	CONSOLE	CONT	COS()	CVB()	CVL()
CVS()	CVW()	CX()	CY()	DAALM()	DAERR()	DALIMIT	DALSET
DARDY()	DASET	DATA	DATA\$	DATE\$	DEC	DECO()	DEFAULT
DIM	DIST()	DLOAD	DO	DREAD	DRV	DUMP	DUMPC
DUMPINT	DUMPP	DUMPR	DWNL	DECLR	ELSE	ELSEIF	ENCO()
END	ENDIF	ENTRY	EOF()	ERR()	EXIT	EXP()	EXT
ERN	FEED	FEND	FERROR()	FILES	FLOAT	FOR	FORCE
FRE	FRE()	FREV	FUNCTION	GO	GOSUB	GOTO	HALT
HCDATA()	HCFSET	HCISSET	HCOUTEN	HCPSET	HCRESET	HCSTAT()	HEX\$()
HOME	HORDR	HTEST	IDAC	IDBC	IDCAN	IDCLR	IDCRSP()
ID	IDBRID	IDBWID	IDRID	IDW	IDPRR	IDPRW	IDRI
IDREADY()	IDRECV	IDRV	IDRVA	IDRVH	IDRVHA	IDRVHAID	IDRXACK
IDRXDT	IDSEND	IDTCHR	IDTXACK()	IDTXDT	IDTXRDY()	IDTYPE	IDWI
IDWRX()	IDWV	IDWVA	IDWVH	IDWVHA	IDWVHAID	IF	IFBIN()
INCIND()	INH()	INL()	INPUT	INPUT\$()	INPUTR	INSTR()	INT
INT()	INTEGER	INV()	INW()	KILL	LDCR	LDIO	LDVAL
LEFT\$()	LEN()	LINE	LIST	LOC()	LOCATE	LOF()	LOG()
LONG	LOOP	LROLL()	LSHIFT()	MAP	MAXDEV	MEW\$()	MID\$
MID\$()	MKB\$()	MKL\$()	MKS\$()	MKW\$()	MOD	MODEMON	MONCLR
MONDISP	MONENT	MONTS	MOVE	MYINT	MYTASK()	NAMENEG()	NEXT
NGX	NGY	NOP	NOTOCT\$()	OFF	ON	ONERR	ONSW
OPEN	OR	OUT	OUTD	OUTH	OUTL	OUTW	PALET
PAUSE	PDEL	PEEK()	PODATA()	POFCHG	PIOSET	POKE	POORG
POPSET	POREOS	PORESET	POSTART	POSTAT()	PRGSIZE	PRINT	PRINTR
PRIVATE	PRMPRM()	PRMCLR	PRMR	PRMR()	PSADRS()	PSADSET	PSAID()
PSBUSY()	PSCTRL	PSDSET	PSECLR	PSERR()	PSJOBNO	PSLOAD	PSORGH
PSORGS	PSPRM	PSPRM()	PSREADY()	PSSAVE	PSSTART	PSSTAT()	PSSTOP
PSTYPE	PSX	PSY	PULSE	QUIT	RANDOMIZE	RANGE	READ
READR	REAL	RECV	RECVB	REM	RENUM	RESET	RESTORE
RESUME	RETRN	RETURN	RFORCE	RIGHT\$()	RIOCLR	RIOSET	RND()
RROLL()	RSHIFT()	RSLOTRUN	SDERR()	SDRDY()	SDRECV	SDRESET	SDRXACK
SDRCDT	SDSEND	SDTCHR	SDTERM	SDTXACK()	SDTXDT	SDTXRDY()	SDWRX()
SEGT()	SEL	SELECT	SELEND	SEND	SENDB	SET	SGN()
SIN()	SLOT()	SLOTCLR	SLOTI	SLOTICLR	SLOTR()	SPACE\$()	SPEED
SQR()	STEP	STR\$()	STRING	SVCR	SVIO	SVVAL	SW()
SWAP	SWITCH	TAN()	TEST	THEN	TIME	TIME\$	TIME\$()
TIME()	TIMER	TIMER()	TMOUT	TO	TOFF	TON	TSTAT
TSTAT	TW()	TWAIT	UNIT()	UPLDVAL()	VARIABLE	VER	VERINIT
WAIT	WEND	WHILE	WRITE	WRITE	RXO	RXOTZE	ROFLG()
¥	_ACLMAX	_ENCDIV	_HDIR	_HSENSE	_HSPEED	_HTELIM	_MNAME
_PMATCH	_SELAXIS	_SPDMAX	_ZPHASE				

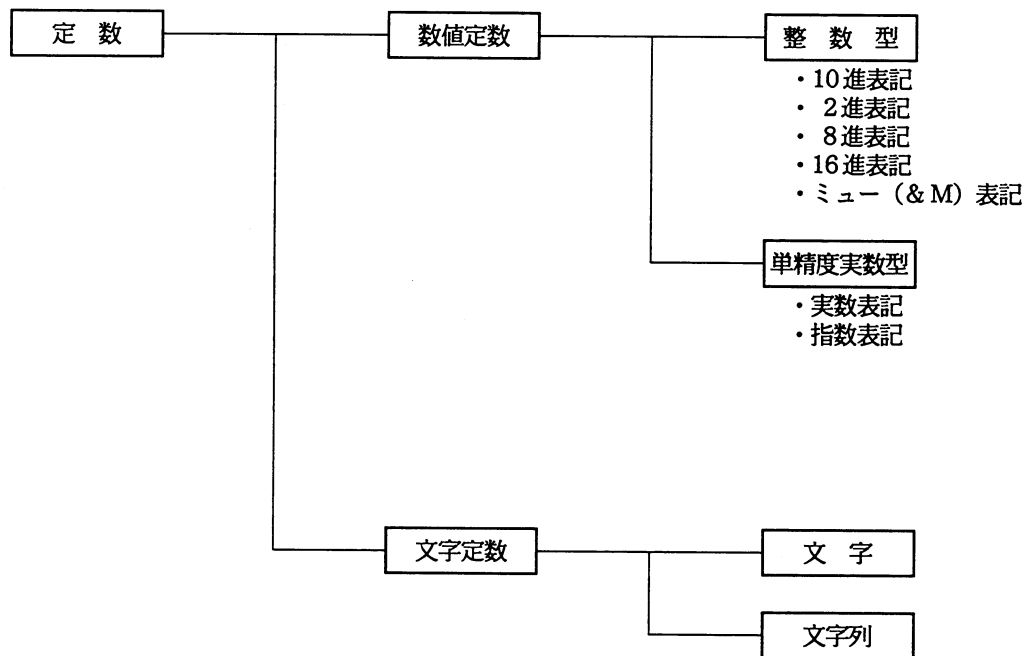
4 - 3

定 数

4-3-1 定数の種類

定数は、値の決まっているデータです。たとえば、「1」、「32」という数字や「A」、「ABC」という文字は、それ自身の値を持つ定数です。

定数の種類は、大きく分けて、数値定数と、文字定数（文字列）があります。数値定数には、データの「型」による分類と、「表記」上の分類があります。



4-3-2

数値定数

数値定数は、通常の10進数での表記の他、2進数、8進数、16進数での表記で記述することができます。さらに、FP - BASICでは、「&M」による表記を使用して数値を記述することができます。

- ・10進表記の例 1 123 6210
- ・2進表記の例 &B1 &b1011011
- ・8進表記の例 &O123 &o722
- ・16進表記の例 &H123 &h12F
- ・ミュー表記の例 &M123 &m12F

注意

ミュー表記 (&M表記) は、末尾1桁だけが16進数表記 (0~F) で、その他は10進数表記です。主に、I/O番号の指定に使用します (「6-1-1. I/O・メモリ構成一覧 (P.124)」参照)。

4-3-3

文字定数

FP - BASICでは、80バイトまでの文字列を文字定数として使用できます。

プログラム中に文字定数を記述する場合、必ず「"」(ダブルクォーテーション)で囲みます。なお、「"」(ダブルクォーテーション)そのものは、文字定数には使用できません。

- ・文字定数の例 "ABC" "123" "FP - BASIC" "1+2=3" "ｱｲｴ" "松下電工"

4 - 4

変 数

4-4-1

変数名

FP - BASICでは、配列を含む任意の変数を使用することができます。

変数名には、8文字までの英数字と「_ (アンダースコア)」が使用できますが、先頭の1文字は必ず英字で書き始めなければならないという約束があります。なお、FP - BASICでは、英字の大文字と小文字を区別しませんので、注意してください。

また、文字型の変数には、変数名の末尾に「\$」を付けなければなりません。さらに、文字変数では、「\$」の後に「*」と0~255数値を記述することにより、変数のサイズを任意に決めることができます。

・ 数値変数の例	A	B1	C20	FP3	FPBASIC
・ 文字変数の例	A\$	B1\$	C20\$	FP3\$	FPBASIC\$
	A\$*4	B1\$*8	C20\$*16	FP\$*255	

注意

- FP - BASICの変数名には、「P」で始まる文字列および予約語を使用しないでください。また、予約語の直後が数字または「_ (アンダースコア)」の文字列も、変数名に使用できません(「4-2-2. FP - BASICの予約語 (P.81)」参照)。

4-4-2

変数の型と値

変数は、代入される値の種類により、数値型と文字型に分けられます。さらに、数値型の変数の場合、1バイト整数型、2バイト整数型、4バイト整数型、4バイト実数型に分けられます。

4バイト整数型 (LONG型) 以外では、使用時に変数宣言 (型宣言) をしなければなりません。なお、変数宣言をせずに変数を使用した場合、タスク間でグローバルな変数として定義されます。

注意

- 文字型の変数 (例 : A\$, DATA1\$) では変数の末尾に「\$」記号を付けますが、数値型の変数の場合末尾に記号は付けません。
- 値が代入される前の変数は、数値変数では「0」として、文字変数では「空の文字列 (“” ヌルストリング)」として扱われます。
- 変数宣言は、プログラム中の各タスクブロックの (FUNCTION ~ FEND) の最初にまとめて記述します (タスクブロックの途中で変数宣言をすると正しく解釈されない場合があります)。「4-6-3.」 ~ 「4-6-7.」のプログラムリストを参照してください。

変数宣言	型宣言の意味	値	メモリ格納形
BYTE A	1バイト整数型	-128～+127 &B0～&B01111111 &00～&0177 &H0～&H7F &M0～&M7F	1バイト
INTEGER A	2バイト整数型	-32,768～+32,767 &B0～&B01111111 11111111 &00～&0177777 &H0～&H7FFF &M0～&M2047F	2バイト
LONG A	4バイト整数型	-2,147,483,648～+2,147,483,647 &B0～&B01111111 11111111 11111111 11111111 &00～&017777777777 &H0～&H7FFFFFFF &M0～&M1342177272F	4バイト
REAL A	4バイト実数型	精度が保証される0に最も遠い値： -1.2E+38 ～ +1.2E+38 精度が保証される0に最も近い値： -1.2E-38 ～ +1.2E-38	4バイト
STRING A\$	文字型	0～80バイトの文字列	80バイト
STRING A\$n	文字型	nバイトの文字列 (1 ≤ n ≤ 255)	nバイト

注意

- FP – BASICの整数は、符号付き整数です。ただし、2・8・16進数表記のバイナリデータの先頭1ビットは符号として解釈されません。

たとえば、INL ()、INH () 関数でI/Oの状態を取り出して1バイト型変数に代入すると、最上位ビットが1の場合エラーになります。INW () 関数の値を2バイト整数型変数に代入する場合、IND () 関数の値を4バイト型整数に代入する場合も、同じことが起こります。

したがって、整数型の変数に2・8・16進数表記のバイナリデータを代入する場合は、十分に注意してください。

●使用する命令

BYTE	変数を1バイト整数として使用するよう宣言します
INTEGER	変数を2バイト整数として使用するよう宣言します
LONG	変数を4バイト整数として使用するよう宣言します
REAL	変数を1バイト実数として使用するよう宣言します
STRING	変数を文字変数として使用するよう宣言します

4-4-3 グローバルな変数とローカルな変数

FP – BASICの変数は、タスク間でグローバルな変数として使用することも、タスク間でローカルな変数として使用することもできます。

- ①何も付けずに型宣言した場合は、タスク間でグローバルな変数として使用されます。

```
210 BYTE A
```

- ②PRIVATE型の型宣言命令を使用した場合は、タスク間でローカルな変数として使用されます。

```
210 PRIVATE BYTE A
```

注意

- 分割コンパイル時の変数の内部定義と外部参照については、「第5章 分割コンパイル」を参照してください。
- ローカル変数に使用した変数名とグローバル変数に使用した変数名が重複しないように、注意してください。なお、ローカル変数は、異なるタスクで同一の変数名が使用できます。

●使用する命令

-
- ◆PRIVATE 変数をローカル変数として宣言します
-

4-4-4 配列

FP – BASICでは、任意の配列を変数として使用することができます。配列の次元は、文法的には256次元まで可能ですが、実際にはメモリ容量とエディタが1行に記述できる文字数（255文字まで）により制限されます。

配列名には、8文字までの英数字が使用でき、先頭の1文字は必ず英字で書き始めなければならないという約束があります（変数名と同じ規則）。また、文字列配列は各要素のサイズ（バイト長）を指定できます。

● 数値配列

```
210 PRIVATE BYTE A(10)
```

```
210 PRIVATE BYTE A(5,7)
```

● 文字列配列

```
210 PRIVATE STRING A$(10)
```

```
210 PRIVATE STRING A$*8(4,2)
```

注意

- 同じような変数をいくつも使用する場合、同じ変数名を付け、たとえば「A(10)」のように配列中の要素の番号を添え字として指示するようにします。このような変数を配列変数と呼びます。
- 配列変数の添え字は、「0」から始まります。したがって、「BYTE A(10)」で宣言した配列変数の添え字は「0」～「10」までです。

4-4-5 型変換

FP – BASICでは、異なった型の数値を代入または演算するとき、次の法則にしたがって型変換を行います。

①代入文による型変換

代入される変数の型に変換されます。ただし、文字型と数値型の変換はできません。

・整数型の変数に実数型の値を代入
(小数点以下を四捨五入します)

・実数型の変数に有効桁数以上の実数を代入
(有効桁数の範囲に丸められます)

```
>LIST
10 FUNCTION MAIN
20 BYTE A
30 A=24.51
40 PRINT A
50 FEND
>DWNLD
>XQT
25
>■
```

```
>LIST
10 FUNCTION MAIN
20 REAL B
30 B=13.45789123456
40 PRINT B
50 FEND
>DWNLD
>XQT
13.45789
>■
```

②算術演算式による型変換

精度の異なる数値演算は、精度の低い値を精度の高い方に変換します。

```
>LIST
10 FUNCTION MAIN
20 BYTE A
30 REAL B
40 A=23
50 B=2.34
60 PRINT (A*B)
70 FEND
>DWNLD
>XQT
53.82
>■
```

③論理演算式（ビット演算式）による型変換

すべて整数に変換（小数点を四捨五入）して演算します。

```
>PRINT (2 OR 2)
2
PRINT (2 OR 2.499)
2
PRINT (2 OR 2.5)
3
>■
```

4 - 5

式と演算子

4-5-1 式の種類

FP - BASICでは、算術演算、論理演算（ビット演算）、比較演算ができます。

式は、定数、変数、関数および式自体を演算子で結合したものであり、演算子には以下のような特別の記号または文字列を使用します。

	演算子		例	変数Aに代入される結果
四則演算	+	加算	$A = 1 + 1$	2
	-	減算	$A = 2 - 1$	1
	*	乗算	$A = 3 * 2$	6
	/	除算	$A = 4 / 2$	2
	MOD	除算の余り	$A = 5 \text{MOD} 3$	2
論理演算 (ビット演算)	OR	論理和	$A = 1 \text{ OR } 2$	3
	AND	論理積	$A = 1 \text{ AND } 2$	0
	XOR	排他的論理和	$A = 1 \text{ XOR } 2$	2
比較演算	=	等しい	$A = (B = C)$	(真のとき)
	<>	等しくない	$A = (B <> C)$	-1
	<	より大きい	$A = (B < C)$	
	>	より小さい	$A = (B > C)$	(偽のとき)
	<=	以上	$A = (B <= C)$	0
	>=	以下	$A = (B >= C)$	

注意

- FP - BASICの論理演算式（ビット演算式）は、2つの数値の対応するビットごとに、指定された論理演算を実行します。
- FP - BASICの比較演算式は、2つの数値の大きさを比較して、式が真であるとき「-1」を、式が偽であるとき「0」を、それぞれ式の値として返します。

4-5-2 演算子の優先順位

演算子の優先順位は、以下のとおりです。「() カッコ」でくくられた式および関数は最優先で演算を行います。

優先順位

1	()	「() カッコ」で囲まれた式
2	関数	数値関数・文字関数
3	-	マイナス記号
4	* /	乗算・除算
5	MOD	除算の余り
6	+ -	加算・減算
7	= < >	比較演算
8	NOT	否定
9	AND	論理積
10	OR	論理和
11	XOR	排他的論理和

4 - 6

FP - BASIC の制御構文

4-6-1

マルチタスクと FUNCTION 文

FP - BASIC のマルチタスクは、1 つのプログラムで制御されます。

- ① プログラム中で、それぞれのタスクは「FUNCTION タスク名」と「FEND」ではさんで記述します。このプログラムの 1 タスク分をタスクブロックまたは FUNCTION ブロックと呼びます。
- ② タスク名には、8 文字までの英数字が使用でき、先頭の 1 文字は必ず英字で書き始めなければならないという約束があります (変数名と同じ規則)。先頭が「P」で始まる文字列および予約語は使用できません。(「4-2-2. FP - BASIC の予約語 (P.81)」参照)
- ③ プログラム中で最初に記述されたブロックがタスク 1 に割り当てられ、実行されます。また、タスク 1 以外のタスクは、他のタスクから起動されないと、実行されません。

```
10 FUNCTION MAIN
20 INTEGER A, B, C, D
30 FOR A=1 TO 7 STEP 1
40 OUTH WY_1, A
50 WAIT 0.3
60 NEXT
70 XQT I2, T_A
80 END
90 FEND
110 FUNCTION T_A
120 FOR B=1 TO 7 STEP 1
130 OUTL WY_1, B
140 WAIT 0.3
150 NEXT
160 XQT I3, T_B
170 END
180 FEND
190 FUNCTION T_B
200 FOR C=1 TO 7 STEP 1
210 OUTL WY_1, C
220 WAIT 0.3
230 NEXT
240 XQT I4, T_C
250 END
260 FEND
270 FUNCTION T_C
280 FOR D=1 TO 7 STEP 1
290 OUTL WY_1, D
300 WAIT 0.3
310 NEXT
320 XQT I1, MAIN
330 END
340 FEND
```

タスク名 MAIN
(タスク 1 で実行)

タスク名 T_A
(タスク 2 で実行)

タスク名 T_B
(タスク 3 で実行)

タスク名 T_C
(タスク 4 で実行)

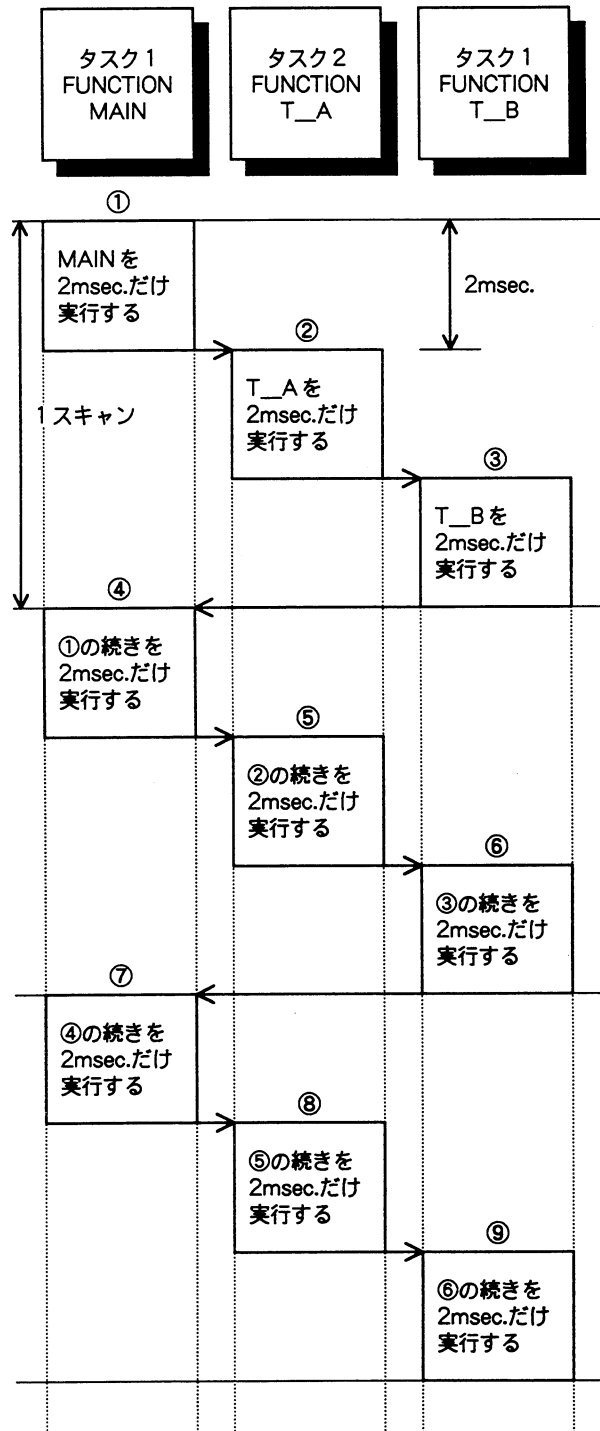
●使用する命令

FUNCTION~FEND プログラムの開始および終了を宣言します

END プログラムを終了します

注意

- FP - BASIC は、最大 16 タスクのマルチタスクで動作します (タスク番号は 1~16 で指定します)。
- 複数プログラムの分割コンパイルについては、「第 5 章 分割コンパイル」を参照してください。



4-6-2 WAIT 命令によるインターロック

WAIT 命令は、通常時間を計るタイマとして機能しますが、入出力命令と組み合わせてインターロックのための命令として使用できます。

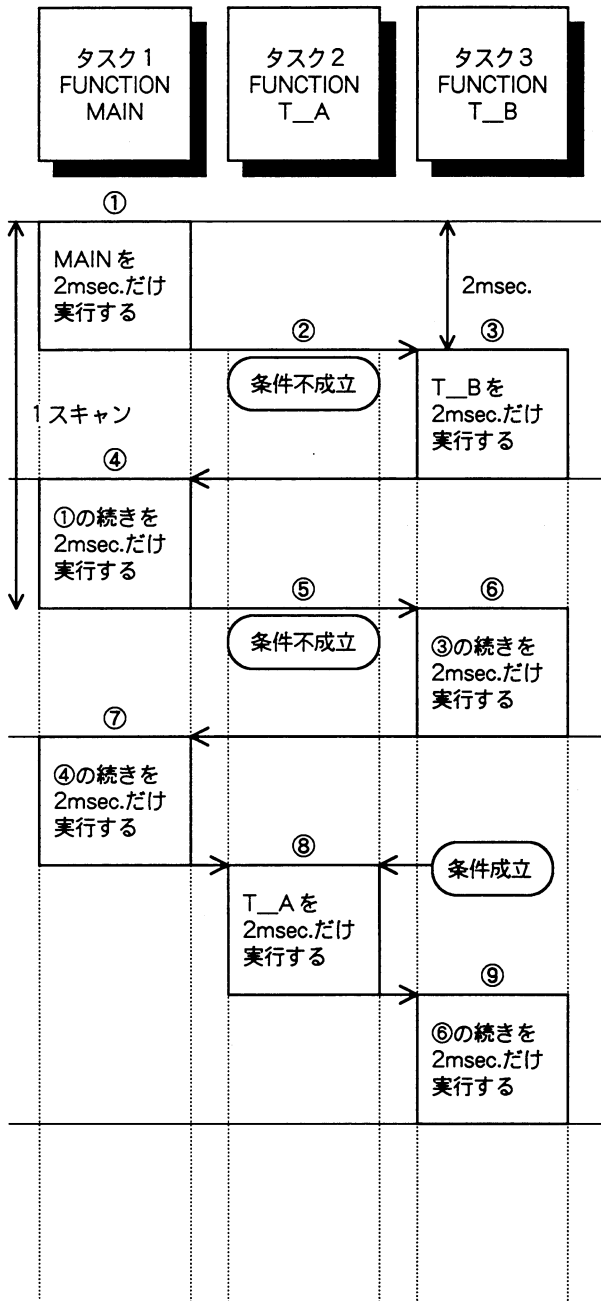
タスク 1	タスク 2	
60 OFF(R_0)		
:	400 FUNCTION TASK_2	
110 XQT !2, TASK_2	:	1
:	<処理 1 >	2
<処理 2 >	:	3
:	500 WAIT SW(R_0)=1;OFF(R_0)	4
200 ON(R_0)	:	5
:	<処理 3 >	6
300 WAIT SW(R_0)=1, OFF(R_0)	:	7
:	600 ON(R_0)	8
<処理 4 >	610 FEND	9

- 1 [タスク 1] TASK_2をタスク 2で実行 (スタート) させます。
- 2 [タスク 2] <処理 1 >を行います。
- 3 [タスク 1] タスク 2が<処理 1 >を行っている間に<処理 2 >を行います。
- 4 [タスク 2] <処理 2 >が終わり、メモリ I/O の R_0 が 1 になるのを待ちます。
- 5 [タスク 1] <処理 2 >が終了したので、タスク 2 に知らせます (R_0 を ON)。
- 6 [タスク 2] タスク 1 が<処理 2 >を終了した (R_0 が ON) ので、<処理 3 >を行います。
- 7 [タスク 1] メモリ I/O の R_0 が 1 になるのを待ちます。
- 8 [タスク 2] <処理 3 >が終了したのでタスク 1 に知らせます (R_0 を ON)。
- 9 [タスク 1] タスク 2 が<処理 3 >を終了した (R_0 が ON) ので、<処理 4 >を行います。

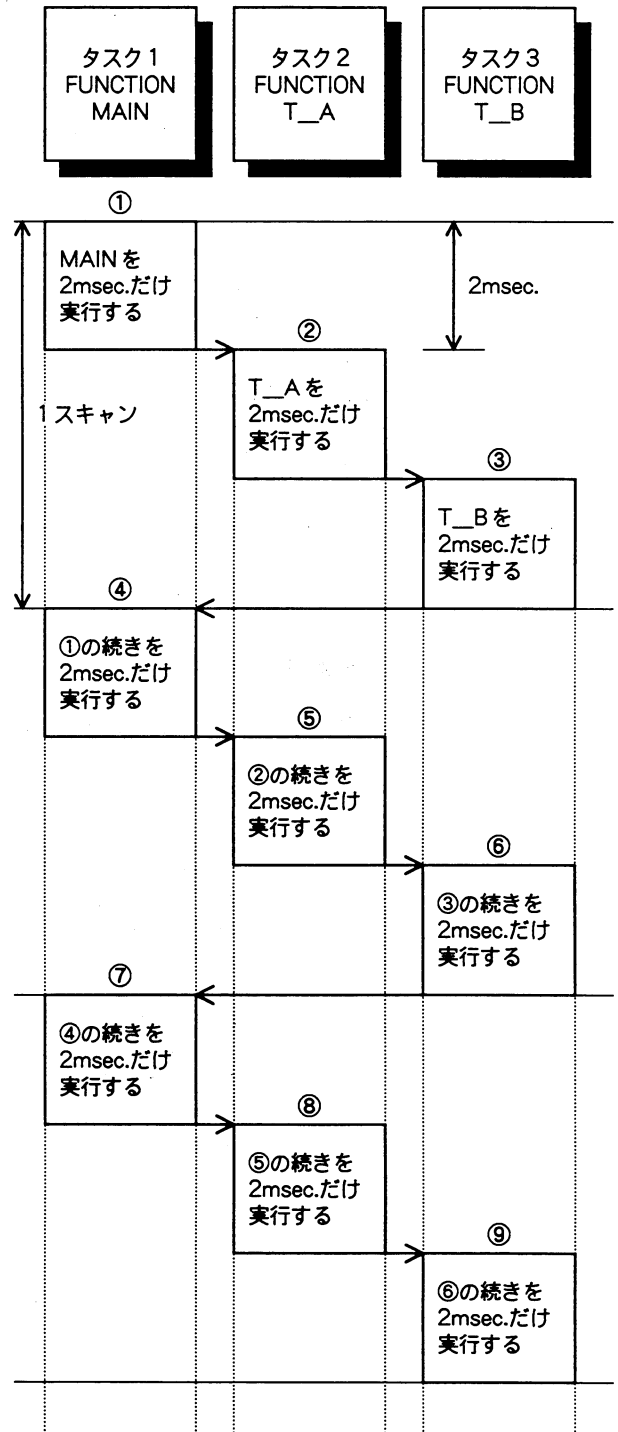
注意

- WAIT 命令を使用せずに、たとえば上記でタスク 2 が
500 IF IN(R_0)=0 THEN GOTO 500
という方法 (入力のポーリング) でインターロックした場合、CPU はタスク 2 のこの行を実行するために多くの時間を費やします。WAIT 命令では、入力状態が変化するまでこのタスクを停止状態 (WAIT 状態) にするので、他のタスクの応答速度への影響が少なくなります。
- WAIT 命令はタイマ命令と呼ばれます (「4-7-5. タイマ命令と関数」を参照してください)。

■ WAIT 命令を使用した場合



■ IF 文を使用した場合



前ページのプログラムの例で、タスク2がWAIT状態になった場合、WAIT条件のみを実行して結果が否なら次のタスクに実行を移します(条件待ちタスクは停止状態= WAIT状態になります)。したがって、条件待ちは、IF文で待つよりもWAIT文で待つ方が、プログラム実行時間を1スキャン分だけ短くすることができます。

4-6-3 分岐命令 GOTO

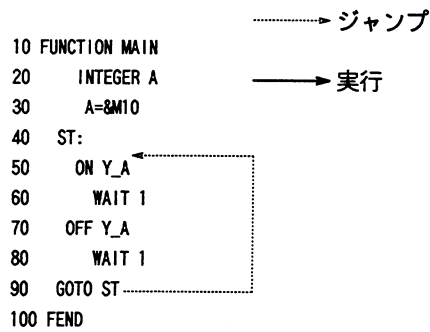
プログラムおよびタスクブロックの中では、命令文は行番号の順序で実行されます。
 GOTO 命令は、行番号またはラベルを指定することにより、次に実行する命令文を任意の行に分岐することができます。ただし、GOTO文により分岐できる分岐先は、同一のタスクブロック内でなければなりません（タスクブロックを越える分岐は不可）。
 分岐先にラベルを使用する場合、必ずラベルの後に「: (コロン)」を付けて記述します。

```

10 FUNCTION MAIN
20     INTEGER A
30     A=&M10
40     ST:
50     ON Y_A
60     WAIT 1
70     OFF Y_A
80     WAIT 1
90     GOTO ST
100 FEND
    
```

●使用する命令

GOTO プログラムの実行を分岐します



4-6-4

サブルーチン呼び出し命令 GOSUB

実行する命令文を任意の行に移す点ではGOTO命令と同じですが、復帰命令RETURNによって、分岐元の次の命令文に実行処理が戻ります。分岐先は、行番号またはラベルで指定しますが、同一タスクブロック内でなければなりません（タスクブロックを越える分岐は不可）。

GOTOでなくGOSUBを使用することにより、何度も行う一連の実行処理の記述を1回だけで済ませることができます。なお、GOSUB ~ RETURNで記述されるプログラムを一般にサブルーチンと呼び、GOSUB命令の実行をサブルーチンの呼び出しと呼びます。

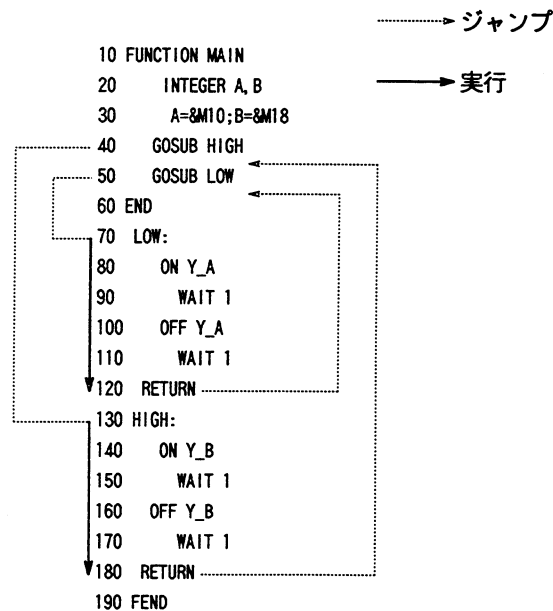
```

10 FUNCTION MAIN
20   INTEGER A, B
30   A=&M10;B=&M18
40   GOSUB HIGH
50   GOSUB LOW
60 END
70 LOW:
80   ON Y_A
90   WAIT 1
100  OFF Y_A
110  WAIT 1
120  RETURN
130 HIGH:
140  ON Y_B
150  WAIT 1
160  OFF Y_B
170  WAIT 1
180  RETURN
190 FEND

```

上記のプログラムの場合、制御の流れは以下ようになります（数字は行番号です）。

なお、このようにサブルーチンの中でさらにサブルーチンを呼び出すことを、ネスティング（入れ子）と呼びます。FP - BASICの場合、GOSUB文のネスティングは、最高10段までです。



●使用する命令

GOSUB	サブルーチンに実行を移します
RETURN	サブルーチンから復帰します

4-6-5 ON GOTO/GOSUB文

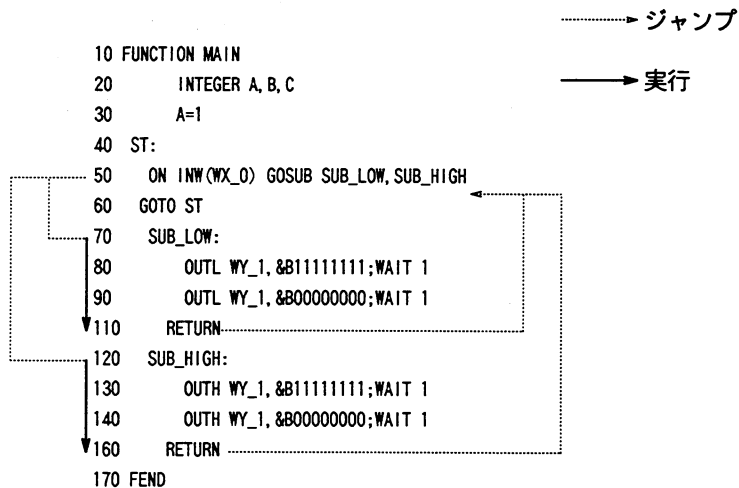
プログラム中の式の値によって、指定した分岐先への分岐またはサブルーチンの呼び出しをします。分岐先は行番号またはラベルで指定しますが、同一タスク内でなければなりません（タスクブロックを越える分岐は不可）。

```

10 FUNCTION MAIN
20     INTEGER A, B, C
30     A=1
40 ST:
50     ON INW(WX_0) GOSUB SUB_LOW, SUB_HIGH
60     GOTO ST
70     SUB_LOW:
80         OUTL WY_1, &B11111111;WAIT 1
90         OUTL WY_1, &B00000000;WAIT 1
110    RETURN
120    SUB_HIGH:
130        OUTH WY_1, &B11111111;WAIT 1
140        OUTH WY_1, &B00000000;WAIT 1
160    RETURN
170 FEND
    
```

●使用する命令

- ◆ON ~ GOTO 式の値に応じて分岐します
- ◆ON ~ GOSUB 式の値に応じてサブルーチンと呼出します



注意

- FP – BASICの場合、分岐先の行番号またはラベルの記述の順序は、最も大きな式の値に対応する行番号またはラベルから記述します。
- 式の値が不になったときエラーが起きます。値が「0」または分岐先の数より大きいときは、次のステートメントに実行が移りエラーになりません。

4-6-6

SELECT ~ CASE文

プログラム中の式の値によって、実行する命令を選択します。

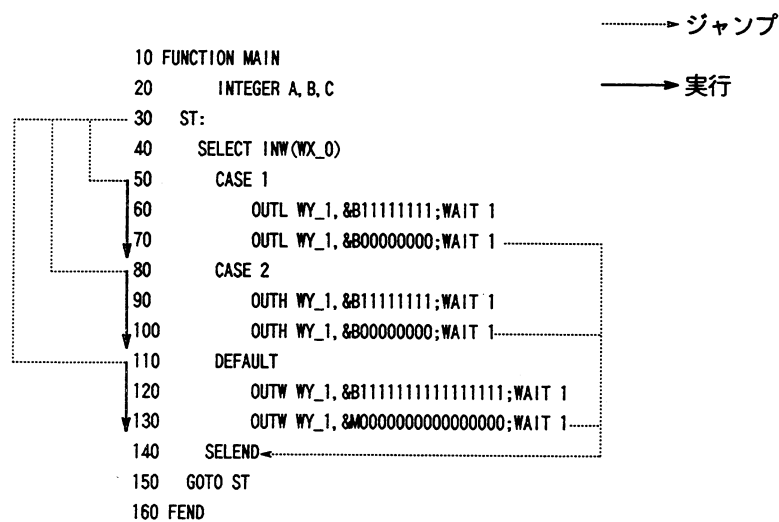
```

10 FUNCTION MAIN
20   INTEGER A, B, C
30   ST:
40   SELECT INW(WX_0)
50     CASE 1
60       OUTL WY_1, &B11111111;WAIT 1
70       OUTL WY_1, &B00000000;WAIT 1
80     CASE 2
90       OUTH WY_1, &B11111111;WAIT 1
100      OUTH WY_1, &B00000000;WAIT 1
110    DEFAULT
120      OUTW WY_1, &B1111111111111111;WAIT 1
130      OUTW WY_1, &M0000000000000000;WAIT 1
140    SELEND
150    GOTO ST
160 FEND

```

●使用する命令

◆SELECT~
CASE 式の値と一致するCASE文を実行します



4-6-7 IF/IFB文

■ IF条件文

条件分岐命令IFは、「ある条件がAの場合はある処理をして、B場合は別の処理をする」というように、条件によってその後の処理を変える必要があるときに使用します。

記述は、「IF 条件式 THEN 命令文① ELSE 命令文②」です。条件式が満たされた場合（条件式の値が0以外）に限って、命令文①を実行します。条件文が満たされなかった場合（条件式の値が0）は、命令文②（ELSEを省略した場合は次の命令）を実行します。

IF条件文は、必ず1行で記述しなければなりません。

```
10 FUNCTION MAIN
20   INTEGER A
30   ST:
40     A=INW(WX_0)
50   JUDGE:
60     IF A=1 THEN OUTW WY_1,1 ELSE OUTW WY_1,2
70   GOTO ST
80 FEND
```

注意

- 条件文および実構文は、複数行にわたって記述することは出来ません。
- IF条件文はネスティング（入れ子）できません。
- 条件式の値について

条件式は、関係演算式、またはブール代数とも呼ばれ、その値が「真」か「偽」かだけを判断材料に使います。

また、算術演算、論理演算（ビット演算）、比較演算などのすべての式は、条件式として使用される場合、式の値が0以外の場合「真」として、式の値が0の場合は「偽」として、評価されます。

式については、「4-5-1. 式の種類 (P.89)」を参照してください。

● 使用する命令

```
IF THEN~
   ELSE プログラムの実行を分岐します
```

■ IFB 条件文

条件分岐命令IFBは、THEN 実行文に複数の行を記述することができるため、ブロック形式のIF文ともいわれます。通常のIF条件分岐では複雑なネスティング（入れ子）になるような場合でも、すっきりとしたプログラムが記述できます。

各々の実行文は、複数行での記述、およびマルチステートメントでの記述が可能です。

```

10 FUNCTION MAIN
20 INTEGER A
30 ST:
40 A=INW(WX_0)
50 JUDGE:
60 IFB A=1 THEN
70     OUTW WY_1,1;WAIT 1;OUTW WY_1,0;WAIT 1
80     OUTW WY_1,0;WAIT 1
90 ELSEIF A=2 THEN
100    OUTW WY_1,2;WAIT 1;OUTW WY_1,0;WAIT 1
110    OUTW WY_1,0;WAIT 1
120 ELSEIF A=3 THEN
130    OUTW WY_1,3;WAIT 1;OUTW WY_1,0;WAIT 1
140    OUTW WY_1,0;WAIT 1
150 ELSE
160    OUTW WY_1,&B11111111111111111111
170 ENDIF
180 GOTO ST
190 FEND

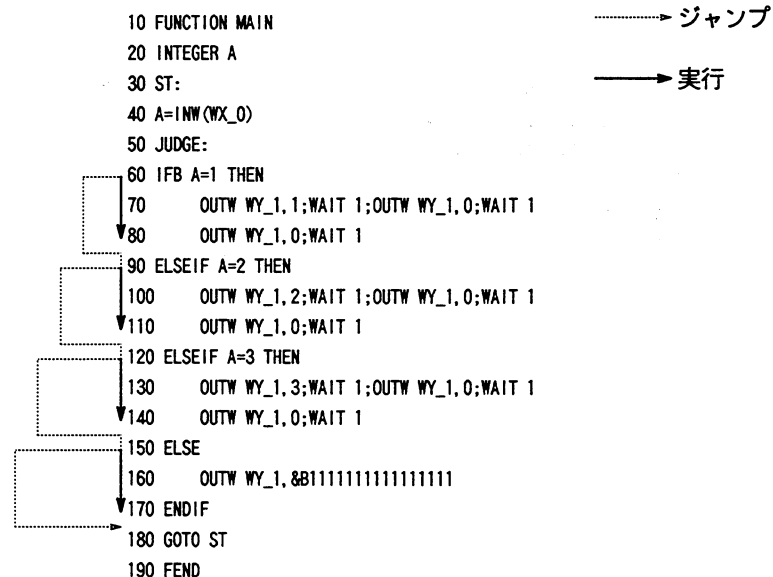
```

● 使用する命令

◆ IFB THEN~

ELSEIF~/ 条件式の結果により、プログラムの実行を制御します

ELSE~ENDIF



4-6-8 ループ文

命令文を、指定回数分、あるいはある条件が成立するまで、繰り返し実行するには、ループ文を使用します。

■ FOR ループ文

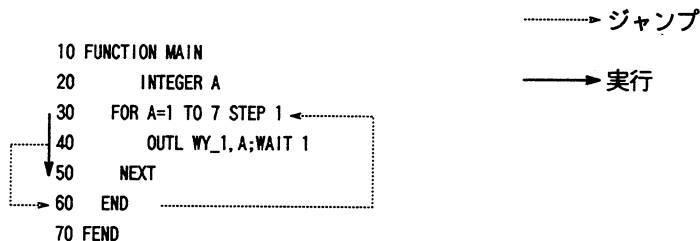
「FOR 変数=初期値 TO 最終値 STEP 差分」と「NEXT」で、繰り返す命令文の行を囲んで記述します。変数は繰り返しのカウンタとして使用され、差分は省略すると1に自動設定されます。

```

10 FUNCTION MAIN
20     INTEGER A
30     FOR A=1 TO 7 STEP 1
40         OUTL WY_1,A;WAIT 1
50     NEXT
60     END
70 FEND
    
```

●使用する命令

FOR~NEXT	命令文を繰り返し実行します
◆DO~LOOP	条件式が満たされている間、命令文を繰り返し実行します (DO条件ループ)
◆WHILE~ WEND	条件式が満たされている間、命令文を繰り返し実行します (WHILE条件ループ)
◆EXIT	ループから強制的に脱出します



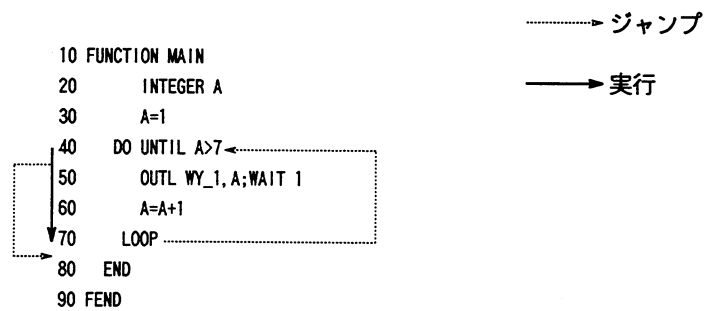
■ DO ループ文

前ページの FOR ループ文と同じ内容を、DO ループ文で記述すると次のようになります。

```

10 FUNCTION MAIN
20   INTEGER A
30   A=1
40   DO UNTIL A>7
50     OUTL WY_1,A;WAIT 1
60     A=A+1
70   LOOP
80 END
90 FEND

```



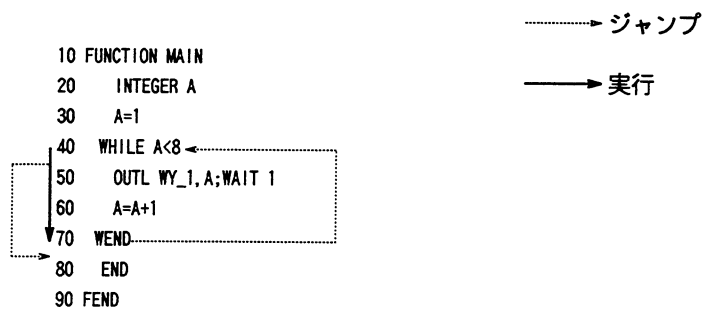
■ WHILE ループ文

前ページの FOR ループ文と同じ内容を、WHILE ループ文で記述すると次のようになります。

```

10 FUNCTION MAIN
20   INTEGER A
30   A=1
40   WHILE A<8
50     OUTL WY_1,A;WAIT 1
60     A=A+1
70   WEND
80 END
90 FEND

```



4-6-9

CALL ~ FUNCTION文

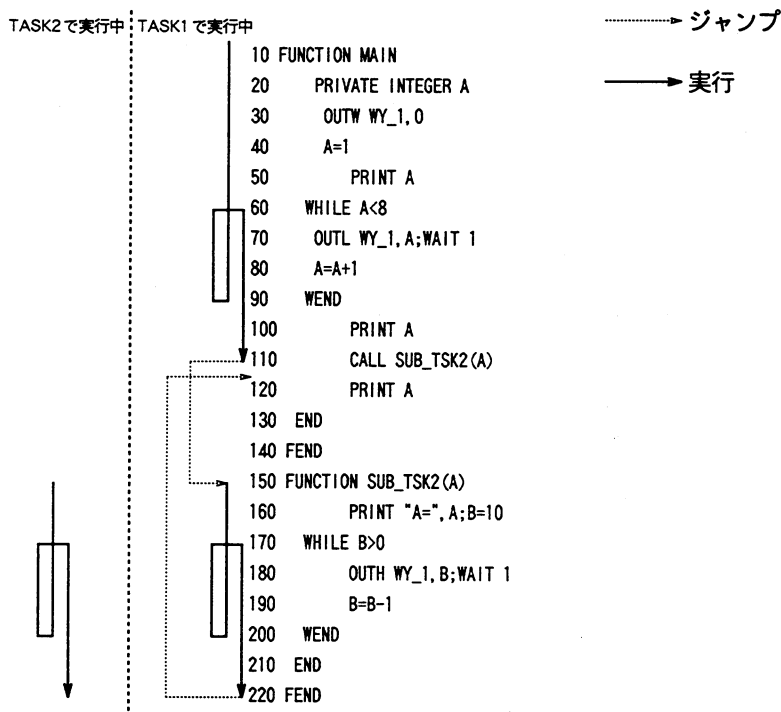
CALL文を使用すれば、他のタスクをサブルーチンとして呼び出せ、さらにサブルーチンとの間で変数の値の受け渡しができます。これにより、モジュール間の独立性の高い、いわゆる構造化プログラミングが可能になります。

```

10 FUNCTION MAIN
20   PRIVATE INTEGER A
30   OUTW WY_1,0
40   A=1
50   PRINT A
60   WHILE A<8
70     OUTL WY_1,A;WAIT 1
80     A=A+1
90   WEND
100  PRINT A
110  CALL SUB_TSK2(A)
120  PRINT A
130  END
140 FEND
150 FUNCTION SUB_TSK2(A)
160  PRINT "A=",A;B=10
170  WHILE B>0
180    OUTH WY_1,B;WAIT 1
190    B=B-1
200  WEND
210  END
220 FEND
    
```

●使用する命令

◆CALL~
FUNCTION 他タスクをサブルーチンとして呼び出します



4-6-10 割り込み制御

割り込みとは、あらかじめ設定した特定の条件によって、即時に実行の流れを制御することです。割り込みが発生すると、通常の処理（その時点で実行しつつある処理）を中断して、指定したサブルーチンを実行し、その後通常の処理に戻ります。

■入力割り込み ONSW () GOSUB

入力割り込み処理プログラムを定義します。

指定した入力ポートが指定した状態に変化した時に、割り込み処理サブルーチンに実行を移します。

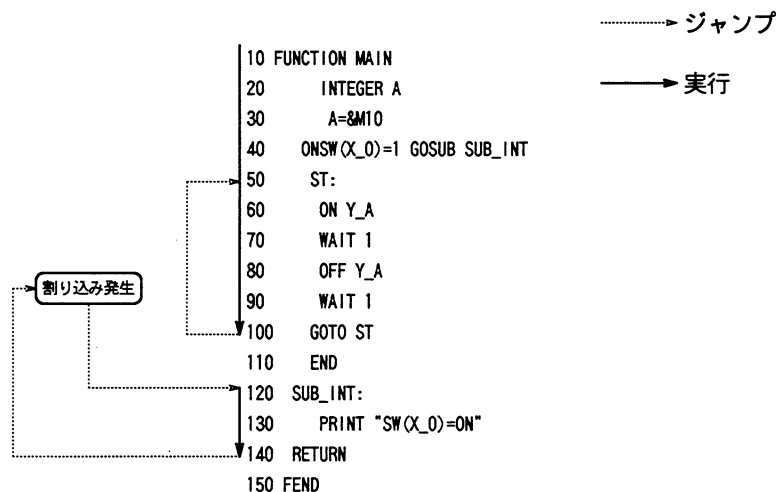
```

10 FUNCTION MAIN
20   INTEGER A
30   A=8M10
40   ONSW(X_0)=1 GOSUB SUB_INT
50   ST:
60   ON Y_A
70   WAIT 1
80   OFF Y_A
90   WAIT 1
100  GOTO ST
110  END
120 SUB_INT:
130  PRINT "SW(X_0)=ON"
140  RETURN
150 FEND

```

●使用する命令

ONSW () 入力割り込み処理を定義します



注意

- WAIT 命令実行中は割り込みルーチンは作動しません。

■エラー割り込み ONERR <行番号/ラベル>

エラー割り込み処理プログラムを定義します。

通常、エラー発生時には、エラーが発生したタスクが終了し、エラーの状態を表示します。エラー割り込みを使用すれば、プログラムの実行を停止せずに、エラー処理サブルーチンに実行を移すことができます。

```

10 FUNCTION MAIN
20     INTEGER A
30     A=8M10
40     TMOUT 5
50     ONERR ER_SUB
60     WAIT SW(X_8M10)
70     IF SW(X_8M10)=0 THEN GOTO 60
80     ST:
90     ON Y_A
100    WAIT 1
110    OFF Y_A
120    WAIT 1
130    GOTO ST
140    END
150    ER_SUB:
160    IF ERR(0)<>1004 THEN END
170    PRINT "TIME OUT ERROR"
180    ECLR
190    RETURN
200    FEND
    
```

●使用する命令

ONERR	エラー割り込み処理を定義します
ECLR	エラーステータスを初期化(クリア)します
ERR ()	エラー番号を返します

■割り込みユニットからの割り込み処理

割り込みユニットを使用して入力割り込みをする場合、ON INT () 命令を使用します。

通常、ON INT () は、タスク1 (FUNCTION MAIN) に記述することにより、割り込みユニットに入力割り込みが発生したとき、他のタスクに記述した割り込み処理サブルーチンに実行を移します。実行中の他のタスクは、停止状態 (WAIT 状態) になります。

```

10 FUNCTION MAIN
20   INTEGER A
30   ON INT(0) SUB_INT
40   INT(0) ON
50   FOR A=1 TO 7 STEP 1
60     OUTW WY_1, A
70     WAIT 1
80     OUTW WY_1, 0
90     WAIT 1
100  NEXT
110 FEND
120 FUNCTION SUB_INT
130   PRINT MYINT
140   OUTW WY_1, &B11111111111111111111
150   INT(0) CLR
160 FEND

```

注意

- 割り込み処理サブルーチンの中では、WAIT 命令は使用できません。
- 割り込み番号は、0～23まで使用できます。詳細は、割り込みユニットのハードウェア導入マニュアルをご覧ください。

●使用する命令

◆ ON INT ()	高機能ユニットからの割り込み処理を定義します
◆ INT () ON	割り込みを許可します
◆ INT () OFF	割り込みを禁止します
◆ INT () CLR	割り込み処理を解除します
◆ MYINT	割り込み処理ルーチンで、実行中の割り込み番号を返します

4 - 7

一般命令と組み込み関数

4-7-1

N88BASIC 一般命令

FP - BASICでは、N88BASICライクなBASIC一般命令が使用できます。なお、BASICタイプCPUでプログラムが実行されているとき、FP - BASICが起動されているパソコンはCPUのターミナルとして動作します。

●使用する命令

PRINT	パソコンの画面にデータを表示します
◆PRINT USING	指定した書式（フォーマット）で、画面にデータを表示します
◆COLOR	テキスト画面に表示される文字の色を指定します
◆LOCATE	テキスト画面のカーソル位置を制御します
◆DATA	DREAD文で読み出されるデータを定義します
◆DREAD	DATA文で定義したデータを読み出し、変数に代入します
◆RESTORE	DREAD文で読み出すDATA文を指定します

4-7-2

ディスクファイルアクセス命令

BASICタイプCPUでは、パソコンのディスクドライブのデータを読み書きすることができます。BASICタイプCPUでプログラムが実行されているとき、FP - BASICを起動しているパソコンはCPUの補助記憶装置として動作します。

●使用する命令と関数

◆OPEN	ファイルを開きます
◆CLOSE	ファイルを閉じます
◆PRINT #	ファイルにデータを出力します。
◆INPUT	ファイルからデータを入力し、変数に代入します
◆LINE INPUT	ファイルから入力されるデータを区切らず一括して文字変数に代入します
◆INPUT \$ ()	ファイルから指定された長さの文字列を取り出して、返します
◆LOC ()	ファイル中の論理的な現在位置を返します
◆LOF ()	ファイルの大きさを返します
◆EOF ()	ファイルの終了コードを返します

4-7-3

変数操作命令

●使用する命令と関数

◆INC	変数をインクリメントします（カウントUP）
◆DEC	変数をデクリメントします（カウントDOWN）
◆SWAP	2つの変数の値を入れ替えます

4-7-4 I/Oポートアクセス命令

FP – BASICでは、BASICタイプCPUのプログラマブルコントローラとしての機能をサポートするため、I/Oポートのアクセスに関する豊富な命令と関数を用意しています。

●使用する命令と関数

ON	I/Oを1ビット単位でONします
OFF	I/Oを1ビット単位でOFFします
OUTL	ワード指定したI/O、メモリの下位8ビットに、データを出力します
OUTH	ワード指定したI/O、メモリの上位8ビットに、データを出力します
OUTW	ワード指定したI/O、メモリに16ビットデータを出力します
OUTD	ワード指定したI/O、メモリに32ビットデータを出力します
SW ()	I/Oの状態を1ビット単位で返します
INL ()	ワード指定したI/O、メモリの下位8ビットの内容を返します
INH ()	ワード指定したI/O、メモリの上位8ビットの内容を返します
INW ()	ワード指定したI/O、メモリの内容を16ビット単位で返します
IND ()	ワード指定したI/O、メモリの内容を32ビット単位で返します
◆BLKOUT	I/Oへの一括書き込みをします
◆BLKMOV	I/O間のデータ転送をします

4-7-5 タイマ命令と関数

FP – BASICでは、タスクごとに1つのタイマが使用できます。

●使用する命令

WAIT	プログラムを一時的に停止状態にします
TMOUT	WAIT文のタイムアウト時間を設定します

4-7-6 システム変数とシステム関数

FP – BASICでは、いくつかのシステム変数とシステム関数が使用できます。これらのシステム関数とシステム変数は、BASICタイプCPUのシステム関数、システム変数であることに注意してください (CPUと接続されていない状態では使用できません)。

DATE \$, TIME \$, DATE \$ (), TIME \$ () はFP3Hのみ使用できます。

●使用する変数と関数

◆DATE \$	日付を設定します
◆DATE \$ ()	日付を返します
◆TIME \$	時刻を設定します
◆TIME \$ ()	時刻を返します
◆TIMER	システム用タイマの経過値を初期化します
◆TIMER ()	システム用タイマの経過値を返します (10ms単位)
MYTASK ()	タスク番号を返します

4-7-7 組み込み関数

FP - BASICには、豊富な組み込み関数が用意されています。これにより、BASICタイプCPU単体で高度な演算を行うことができ、シンプルで高機能なシステムを構築することができます。なお、CPUと接続されていない状態では、これらの組み込み関数は使用できません。

■数値関数と命令

ABS ()	数値の絶対値を返します
INT ()	数値の小数点以下を切り捨てます
LSHIFT ()	数値をビット単位で左シフトします
RSHIFT ()	数値をビット単位で右シフトします
LROLL ()	数値をビット単位で左ローテート (回転) します
RROLL ()	数値をビット単位で右ローテート (回転) します
BCD ()	バイナリコードを2進化10進コードに変換します
BIN ()	2進化10進コードをバイナリコードに変換します
DECO ()	数値をデコード変換します
ENCO ()	数値をエンコード変換します
INV ()	数値をビット反転します
NEG ()	数値の2の補数を返します
SEGT ()	数値を7セグメント表示用データに変換します
UNIT ()	複数の数値の下位4ビットを結合します
DIST ()	数値を4ビット単位で分離します
◆RND ()	乱数を返します
◆RANDMIZE	新しい乱数系を設定します

■三角関数

◆ATN ()	数値の逆正接 (アークタンジェント) を返します
◆COS ()	数値の余弦 (コサイン) を返します
◆SIN ()	数値の正弦 (サイン) を返します
◆TAN ()	数値の正接 (タンジェント) を返します

■対数関数

LOG ()	数値の自然対数を返します
EXP ()	eに対する指数関数の値を返します
SQR ()	数値の平方根 (スクエアルート) を返します

■文字関数

ASC ()	1バイトの文字に対応するキャラクタコードを返します
CHR \$ ()	キャラクタコードに対応する1バイトの文字を返します
LEN ()	文字列の文字数を返します
RIGHT \$ ()	文字列の右端から指定した文字数分の文字列を返します
LEFT \$ ()	文字列の左端から指定した文字数分の文字列を返します
MID \$ ()	文字列の指定した位置から指定した文字数分の文字列を返します
◆MID \$ ()	文字列の指定した部分を別の文字列で置き換えます
SPACE \$ ()	指定した文字数分の空白（スペース）を返します
◆INSTR ()	文字列中の何文字目に指定した文字列があるかを返します

■数値文字変換関数

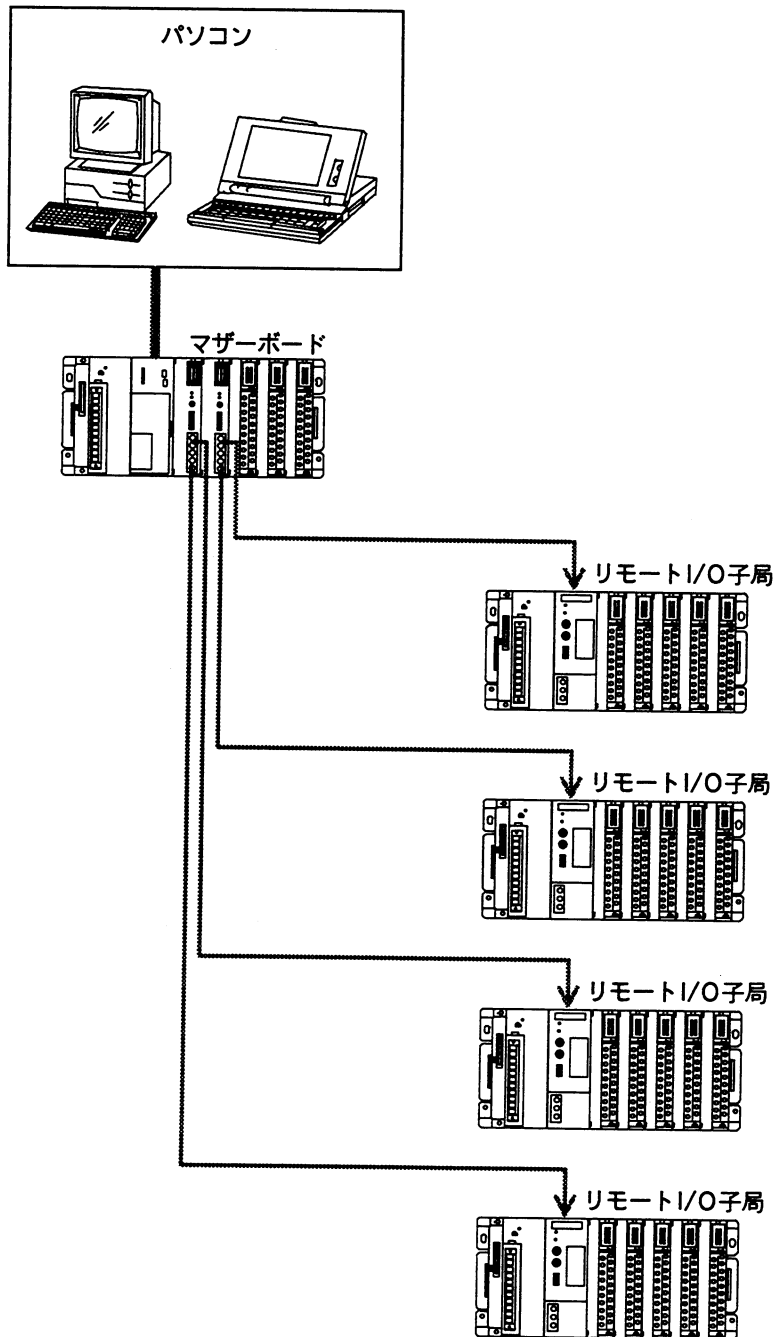
BIN \$ ()	数値を2進表記の文字列に変換します
OCT \$ ()	数値を8進表記の文字列に変換します
HEX \$ ()	数値を16進表記の文字列に変換します
MEW \$ ()	数値をMEW表記の文字列に変換します
STR \$ ()	数値を文字列に変換します
VAL ()	文字列を数値に変換します
◆MKB \$ ()	1バイト整数を1バイトの文字に変換します
◆MKW \$ ()	2バイト整数を2バイトの文字列に変換します
◆MKL \$ ()	4バイト整数を4バイトの文字列に変換します
◆MKS \$ ()	4バイト実数を4バイトの文字列に変換します
◆CVB ()	1バイトの文字を1バイト整数に変換します
◆CVW ()	2バイトの文字列を2バイト整数に変換します
◆CVL ()	4バイトの文字列を4バイト整数に変換します
◆CVS ()	4バイトの文字列を4バイト実数に変換します

4 - 8

リモートI/Oシステムのスロット割り付け命令

BASICタイプCPUには、1台あたり4台 (No.1~4) のリモートI/Oマスタユニット (親局) を接続することができ、1台のリモートI/Oマスタユニットからは32台 (No.1~32) までのリモートI/Oスレーブユニット (子局) を接続することができます。1台のスレーブユニットには、32までのスロット数を割り当てることができます。

このようなリモートI/Oシステムの初期設定には、RIOCLR、RPRM、SLOTR、PIOSET、DUMPRのコマンドとSLOTR () 関数を使用します。




```

>PRMR 1, 1, 3
>SLOTR 1, 1, 0, 120
>RIOSET
>PRINT PRMR(1, 1)
3
>PRINT SROTR(1, 1, 0)
120
>■

```

```

>PRINT SLOTR(1, 1, 0)
120
>SLOTR 1, 1, 0, 220
>RIOSET
>PRINT SLOTR(1, 1, 0)
220
>RIOCLR
>PRINT SROTR(1, 1, 0)
120
>■

```

```

>DUMPR 1, 1
  SLOT ATTR  SLOT ATTR  SLOT ATTR  SLOT ATTR
  00 120    01 100    02 100    03 ...
  04 ...    05 ...    06 ...    07 ...
  08 ...    09 ...    10 ...    11 ...
  12 ...    13 ...    14 ...    15 ...
  16 ...    17 ...    18 ...    19 ...
  20 ...    21 ...    22 ...    23 ...
  24 ...    25 ...    26 ...    27 ...
  28 ...    29 ...    30 ...    31 ...

```

```

>DUMPR 1
  SLAVE CNT  SLAVE CNT  SLAVE CNT  SLAVE CNT
  01 03     02 00     03 00     04 00
  05 00     06 00     07 00     08 00
  09 00     10 00     11 00     12 00
  13 00     14 00     15 00     16 00
  17 00     18 00     19 00     20 00
  21 00     22 00     23 00     24 00
  25 00     26 00     27 00     28 00
  29 00     30 00     31 00     32 00
>■

```

*ユニットの機能番号については「5-3-3.」を参照してください。

●使用するコマンドと関数

◆ PRMR	リモートI/O子局の占有スロット数を設定します
◆ SLOTR	リモートI/O子局のスロット割り付けをします
◆ RIOSET	リモートI/Oシステムに子局の占有スロット数とスロット割り付けの設定を転送しますのsロット割り付けをします
◆ RIOCLR	リモートI/Oシステムのスロット割り付けを初期化します (フリーロケーション)
◆ DUMPR	リモートI/O子局の占有スロット数とスロット割り付けの設定値を表示します
◆ PRMR ()	リモートI/O子局の占有スロット数の設定値を返します
◆ SLOTR ()	リモートI/O子局のスロット割り付けの設定値を返します

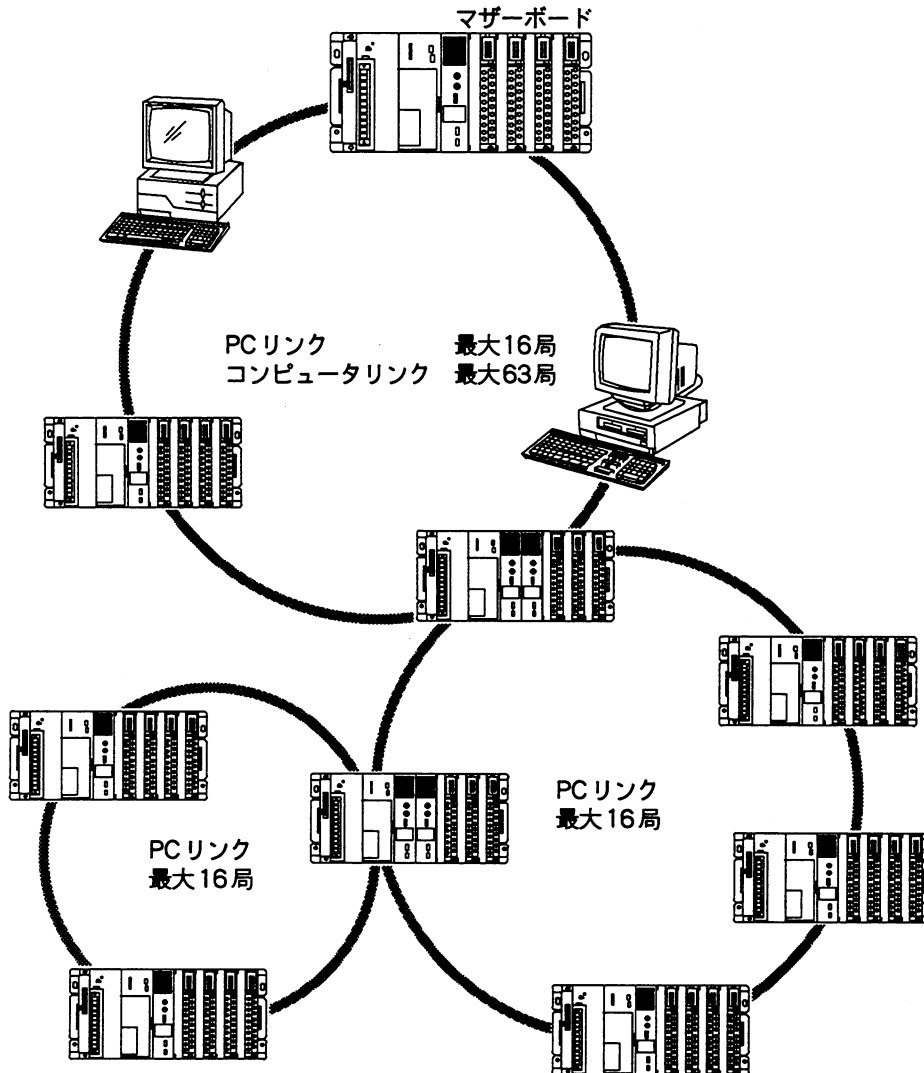
* PRMR、RIOSET、RIOCLR、SLOT、DUMPRは、プログラム中のステートメントとして記述することもできます。

4 - 9

MEWNET データ転送命令

MEWNET (PC リンクの他局) との通信を制御する命令です。

MEWNET では、1台の PC (CPU) 当り 3 台のリンクユニットが接続でき、3つのネットワーク上の他の CPU (他局) を通信相手局に特定して、データの送受信ができます。なお、データ転送命令では、相手局側でのプログラムは不用です。



```

10 FUNCTION MAIN
20   XQT 12, SUB_SEND
30   XQT 13, SUB_RECV
40 END
50 FEND
100 FUNCTION SUB_SEND
110 PRINT "送信開始 DT_100から27-Fを#1の第2局のDT_100以降に書き込む"
120 SEND 1, 2, DT_100, 2, DT100
130 PRINT "送信終了"
140 END
150 FEND
200 FUNCTION SUB_RECV
210 PRINT "受信開始 #1の第2局のDT_100から27-FをDT_100以降に格納"
220 RECV 1, 2, DT_200, 2, DT_200
230 PRINT "受信終了"
240 END
250 FEND

```

●使用する命令

SEND	ワードデータをMEWNETに送信します
RECV	ワードデータをMEWNETから受信します
SENDB	ビットデータをMEWNETに送信します
RECVB	ビットデータをMEWNETから受信します

4-10

高機能ユニット専用命令

BASICタイプCPUは、FP3シリーズのすべての高機能ユニットが使用できます。このため、FP-BASICには、高機能ユニット入出力命令をはじめとする多くの制御命令と関数が用意されています。とくに、各ユニット別の高機能ユニット専用命令を使用すれば、共有メモリの読み書きやI/Oを使用したハンドシェイクを意識することなく、高機能ユニットが使用できます。

4-10-1 通常スロットの高機能ユニット入出力命令

高機能ユニットへのデータの入出力を制御する命令です。

●使用する命令

PRINT %	高機能ユニットにデータを出力します
INPUT %	高機能ユニットからデータを入力します
WRITE	I/Oまたはメモリから、高機能ユニットにデータを書き込みます
READ	高機能ユニットから、I/Oまたはメモリにデータを読み出します

4-10-2 リモートI/Oシステムの高機能ユニット入出力命令

リモートI/Oの子局の高機能ユニットへのデータの入出力を制御する命令と関数です。

●使用する命令と関数

◆PRINTR	リモートI/O子局の高機能ユニットにデータを出力します
◆INPUTR	リモートI/O子局の高機能ユニットからデータを入力します
◆WRITER	I/Oまたはメモリから、リモートI/O子局の高機能ユニットにデータを書き込みます
◆READR	リモートI/O子局の高機能ユニットから、I/Oまたはメモリにデータを読み出します

4-10-3 D/A変換ユニット専用制御命令

D/A変換ユニット専用の制御命令と関数です。

●使用する命令と関数

◆DALIMIT	出力の制限処理を設定します
◆DALSET	出力の上限値、下限値を設定します
◆DASET	入力値を設定します
◆DARDY ()	設定完了ステータスを返します
◆DAERR ()	エラーステータスを返します
◆DAALM ()	アラームステータスを返します

4-10-4 A/D変換ユニット専用制御命令

A/D変換ユニット専用の制御命令と関数です。

●使用する命令と関数

◆ADAVRG	平均処理の指定を設定します
◆ADASET	サンプリング平均回数を設定します
◆ADLIMIT	出力の制限処理を設定します
◆ADLSET	出力の上限値、下限値を設定します
◆ADSCALE	スケーリング処理をするチャンネルを設定します
◆ADSSET	スケーリング値を設定します
◆ADDATA	変換データを読み出します
◆ADRDY ()	変換準備完了ステータスを返します
◆ADERR ()	エラーステータスを返します
◆ADALM ()	アラームステータスを返します
◆ADDATA ()	変換データを返します

4-10-5 シリアルデータユニット専用制御命令

シリアルデータユニット専用の制御命令と関数です。

●使用する命令と関数

◆SDRESET	シリアルデータユニットをリセットします
◆SDTCHR	終端コードを設定します
◆SDTERM	終端コードの送信指定を設定します
◆SDTXDT	送信データを書き込みます
◆SDRXDT	受信データを読み出します
◆SDSEND	データを送信します (送信可能チェックから送信完了確認まで)
◆SDRECV	データを受信します (受信データの有無チェックから受信完了確認まで)
◆SDRXACK	受信データ読み出し完了を通知します
◆SDRDY ()	動作可能確認ステータスを返します
◆SDTXRDY ()	データ送信可能ステータスを返します
◆SDTXACK ()	データ送信完了ステータスを返します
◆SDWRX ()	データ受信待ち状態で待機します (受信データがあるまでWAITする)
◆SDERR ()	通信エラーステータスを返します

4-10-6 パルス出力ユニット専用制御命令

パルス出力ユニット専用の制御命令と関数です。

●使用する命令と関数

◆POISET	初期値を書き込みます
◆POPSET	目標値を書き込みます
◆PORESET	カウンタをリセットします
◆POSTART	パルス出力をスタートします
◆POREOS	パルス出力方向を切り替えます
◆POFCHG	パルス周波数を切り替えます
◆POORG	原点復帰スタートします
◆HCSTAT	ユニットの動作状態を読み出します
◆PODATA ()	経過値を返します

4-10-7 高速カウンタユニット専用制御命令

高速カウンタユニット専用の制御命令と関数です。

●使用する命令と関数

◆ HCISSET	初期値を書き込みます
◆ HCPSET	目標値を書き込みます
◆ HCFSET	入力フィルタ時定数を設定します
◆ HCRESET	カウンタリセットをします
◆ HCOUTEN	出力を許可します
◆ HCSTAT	動作状態を読み出します
◆ HCDATA ()	経過値を返します

4-10-8 位置決めユニット専用制御命令

位置決めユニット専用の制御命令と関数です。

●使用する命令と関数

◆ PSTYPE	ユニットのタイプを宣言します
◆ PSCTRL	稼働モードを設定します
◆ PSSTART	ジョブを始動します
◆ PSJOBNO	ジョブ始動するデータ No.を設定します
◆ PSORGH	機械原点復帰します
◆ PSORGS	ソフト原点復帰します
◆ PSSTOP	ジョブ停止を設定します
◆ PSPRM	位置決めユニットのパラメータを設定します
◆ PSPRM ()	パラメータの設定値を返します
◆ PSPCLR	パラメータを共有メモリに転送します
◆ PSPSET	パラメータをシステムメモリに転送します
◆ PSSAVE	位置データおよびパラメータをセーブします
◆ PSLOAD	位置データおよびパラメータをロードします
◆ PSDSET	位置決めユニットの位置データを設定します
◆ PSADSET	現在位置を変更します
◆ PSREADY ()	準備完了ステータスを返します
◆ PSSTAT ()	動作状態を返します
◆ PSADRS ()	現在値を返します
◆ PSAID	補助出力を読み出します
◆ PSBUSY	動作状態を返します
◆ PSERR ()	エラーコードを返します

第 5 章

5

分割コンパイル

この章の内容

5-1. 分割コンパイル	120
5-1-1. ソースプログラムの作成	120
5-1-2. 分割コンパイル	121
5-1-3. オブジェクトリンク	122

5 - 1

分割コンパイル

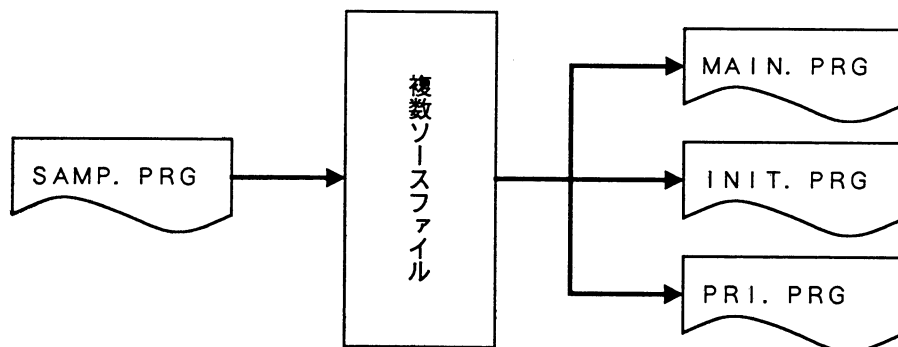
FP-BASICでは、プログラムの分割コンパイルができます。比較的大規模なプログラムは、複数のソースプログラムに分けて作成することにより、効率のよい開発と管理ができます。

5-1-1 ソースプログラムの作成

複数のソースプログラムを作成する場合、変数の内部定義にはENTRY命令を、変数の外部参照にはEXTERN命令を使用します。

```
>LOAD "MAIN. PRG"
>LIST
10 FUNCTION MAIN
11 EXTERN FUNCTION INI '参照ファンクションの宣言
12 INTEGER I
13 ENTRY INTEGER WORK(100) 'グローバル変数定義
14 ENTRY DOUBLE A_SIN, A_COS 'グローバル変数定義
15 XQT 12, INI
16 FEND
>LOAD "INIT. PRG"
LIST
100 FUNCTION INI
101 INTEGER I
102 EXTERN INTEGER WORK(100) '参照変数の宣言
103 EXTERN DOUBLE A_SIN, A_COS '参照変数の宣言
104 EXTERN FUNCTION PRI '参照ファンクションの宣言
105 FOR I=0 TO 100
106 WORK(I)=I
107 NEXT
108 A_SIN=SIN(I);A_COS=COS(I)
109 XQT 13, PRI
110 FEND
>LOAD "PRI. PRG"
>LIST
200 FUNCTION PRI
201 INTEGER I
202 EXTERN INTEGER WORK(100) '参照変数の宣言
203 EXTERN DOUBLE A_SIN, A_COS '参照変数の宣言
204 FOR I=0 TO 100
205 PRINT WORK(I)
206 NEXT
207 PRINT A_SIN, A_COS
208 FEND
>■
```

```
分割前のプログラム
>LOAD "SAMP"
>LIST
10 FUNCTION MAIN
20 INTEGER I
30 DIM WORK(100)
40 DOUBLE A_SIN, A_COS
50 XQT 12, INI
60 FEND
70 '
100 FUNCTION INI
110 FOR I=1 TO 100
120 WORK(I)=I
130 NEXT
140 A_SIN=SIN(I);A_COS=COS(I)
150 XQT 13, PRI
160 FEND
170 '
200 FUNCTION PRI
210 FOR I=0 TO 100
220 PRINT WORK(I)
230 NEXT
210 PRIVATE BYTE A(10)
```



●使用する命令

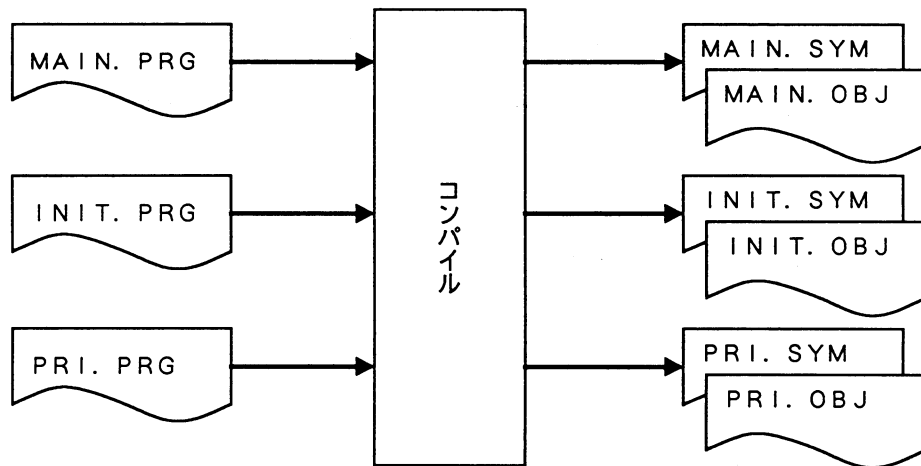
- ◆ENTRY 分割コンパイル時のソースファイル間グローバル変数を宣言します
- ◆EXTERN 分割コンパイル時のソースファイル間グローバル変数またはファンクション名の参照を宣言します

5-1-2 分割コンパイル

複数のソースプログラムプログラムをコンパイルします。

```

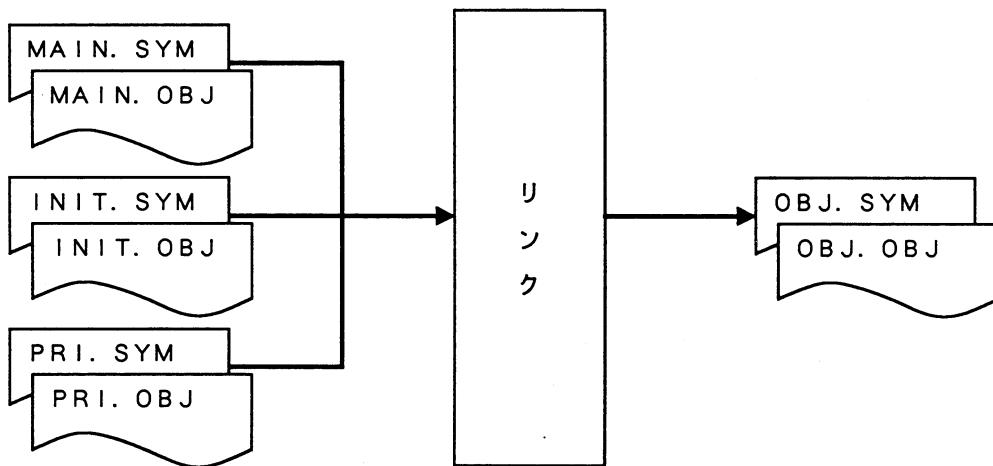
>COMPILE "MAIN"
COMPILE END
>COMPILE "INIT"
COMPILE END
>COMPILE "PRI"
COMPILE END
>FILES
FPB. BAT      SPEED. COM      RSDRV. SYS      PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT  MAIN. PRG        MAIN. OBJ
MAIN. SYM     INIT. PRG      INIT. OBJ        INIT. SYM
PRI. PRG      PRI. OBJ       PRI. SYM
>■
    
```



5-1-3 オブジェクトリンク

複数のオブジェクトプログラムをリンクして、1つのプログラムを作成します。

```
>LINK MAIN+INIT+PRI, OBJ
>FILES
FPB. BAT      SPEED. COM      RSDRV. SYS      PRINT. SYS
CONFIG. SYS   AUTOEXEC. BAT  MAIN. PRG       MAIN. OBJ
MAIN. SYM     INIT. PRG      INIT. OBJ       INIT. SYM
PRI. PRG      PRI. OBJ       PRI. SYM
OBJ. OBJ      OBJ. SYM
```



●使用するコマンド

◆LINK 分割コンパイルのオブジェクトをリンクし、実行オブジェクトを作成します

注意

リンク数の実行ファイル (*.OBJ, *.SYM) の最大サイズは128Kバイトです。
したがってリンク後の*.OBJファイルと*.SYMファイルの合計が128Kバイトを越えるオブジェクトリンクはできません。

第 6 章

6

付 録

こ の 章 の 内 容

6-1. I/O・メモリー一覧	124
6-1-1. I/O・メモリー構成一覧	124
6-1-2. 特殊メモリーI/O一覧	125
6-1-3. 特殊データメモリー一覧	131
6-2. パラメータメモリー設定一覧	138
6-3. エラーコード一覧	140
6-3-1. 自己診断エラーコード一覧	140
6-3-2. FP-BASICエラーコード一覧	141
6-4. スロット割り付けの基礎知識	146
6-4-1. フリーロケーション機能を使用するスロット割り付け	146
6-4-2. スロットコマンド機能を使用するスロット割り付け	147
6-4-3. ユニットのI/O点数と機能番号	148
6-5. プログラム実行時間	150
6-6. ASCII/JIS キャラクターコード一覧	152
6-7. 索引	154

6 - 1

I/O・メモリー一覧

6-1-1 I/O・メモリ構成一覧

BASICタイプCPUのI/O、メモリの構成一覧です。なお、1ワードは、16点（16ビット）です。

名称	点数/ サイズ	アドレス 指定	機能	
外部入力 (X_) (WX_)	2048点 (128ワード)	0~4027 &M0~127F	入力ユニットに対応するI/Oです。	* 3
外部出力 (Y_) (WY_)	2048点 (128ワード)	0~4027 &M0~127F	出力ユニットに対応するI/Oです。 外部出力ユニットで使用しない部分は、メモリI/Oとして使用できます。	* 1 * 2 * 3
メモリI/O (R_) (WR_)	1568点 (98ワード)	0~1567 &M0~97F	外部へ出力できず、CPU内部だけ使用できるI/Oです。	* 1 * 3
特殊メモリI/O (R_) (WR_)	176点 (11ワード)	14400~14575 &M9000~910F	あらかじめ用途が決まっているメモリI/Oです。 読み出し専用です。	* 3
リンクメモリI/O (L_) (WL_)	2048点 (128ワード)	0~4027 &M0~127F	PCリンク使用時のデータ受け渡し用のメモリI/Oです。 リンク用に使用しない部分は、メモリI/Oとして使用できます。	* 1 * 2 * 3
データメモリ (DT_)	2048ワード	0~2047	CPU内部で使用できるメモリです。	* 1
特殊データメモリ (DT_)	256ワード	9000~9255	あらかじめ用途が決まっているデータメモリです。 読み出し専用です。	
リンクデータメモリ (LD_)	256ワード	0~255	PCリンク使用時のデータ受け渡し用のメモリです。	* 1 * 2
ファイルメモリ	* 4	* 4	ユーザープログラムエリアの未使用領域を利用する拡張メモリです。 データメモリとして使用します。	

* 1: パラメータメモリにより、保持/非保持の設定ができます。

* 2: 使用しないY_、L_はRとして、LD_はD_として、それぞれ使用できます。

* 3: X_、Y_、R_、L_のビット単位でのアドレス指定で、&M表記（ミュー表記）を使用する場合、最下位の桁は16進表記、それ以外の桁は10進表記です。

したがって、ワード単位のアドレス指定では、&M表記の最下位桁を除いた数値を指定します。

* 4: ユーザープログラム領域に設定値により異なります（「3-2-4.」参照）。

6-1-2 特殊メモリ/O一覧

BASICタイプCPUの内部で、あらかじめ用途が決まっているメモリ/Oです。読み出し専用なので、ON、OFF、OUT命令の対象にはなりません。SW、IN命令でのみ使用できます。

ワード No.	番 号	名 称	内 容	備 考
900	R_14400 R_&M9000	自己診断エラー	正常時:0 検出時:1	自己診断の結果はDT_9000に格納します。 同コードは、ERRに反映します。
	R_14401 R_&M9001	瞬停検知	正常時:0 検出時:1	DT_9002,9003に検出したスロット番号を格納します。
	R_14402 R_&M9002	ヒューズ断検知	正常時:0 検出時:1	DT_9001に検出した瞬停回数を格納します。
	R_14403 R_&M9003	特殊ユニット 異常	正常時:0 検出時:1	DT_9006,9007に検出したスロット番号を格納します。
	R_14404 R_&M9004	I/O照合異常	正常時:0 検出時:1	DT_9010,9011に検出したスロット番号を格納します。
	R_14405 R_&M9005	電池異常	正常時:0 検出時:1	
	R_14406 R_&M9006	電池異常保持	正常時:0 検出時:1	1度電池異常を検出すると復帰後も保持します。 電源OFFかイニシャライズSW操作で0にすることが出来ます。
	R_14407 R_&M9007	演算異常	正常時:0 検出時:1	発生した行番号をDT_9017に格納します。
R_14408 R_&M9008	リモートI/O R/Wエラー	正常時:0 検出時:1	コントロールデータが指定外するとき マスタユニットがないとき アドレス修飾エラーが発生しているとき 書き込みデータが指定範囲を越えているとき	

ワード No.	番 号	名 称	内 容
901	R_14416 R_&M9010	常時ON	
	R_14417 R_&M9011	常時OFF	
	R_14425 R_&M9018	未使用	
	R_14426 R_&M9019	未使用	
	R_14427 R_&M901A	0.1秒 クロックパルス	
	R_14428 R_&M901B	0.2秒 クロックパルス	
	R_14429 R_&M901C	1秒 クロックパルス	
	R_14430 R_&M901D	2秒 クロックパルス	
	R_14430 R_&M901E	1分 クロックパルス	
902	R_14432 R_&M9020	RUNモード	PROG.モード時：0 RUNモード時：1
	R_14433 R_&M9021	テストモード	通常モード時：0 テストモード時：1
	R_14434 R_&M9022	ポーズ	通常実行中：0 PAUSE命令またはSTEP命令で一時停止中：1
	R_14435 R_&M9023	ポーズモード	テストRUN時のポーズ命令無効：0 テストRUN時のポーズ命令有効：1
	R_14436 R_&M9024	シミュレーション モード	テストRUN時にも通常出力をする：0 テストRUN時に実出力をしない：1
	R_14439 R_&M9027	リモートモード	RUN \leftrightarrow PROG.の遠隔切り替えが可能なモード：1
	R_14440 R_&M9028	強制モード	強制出力をしていないとき（解除時）：0 I/O、メモリI/Oの強制入出力しているとき：1
R_14443 R_&M902B	割込異常フラグ	割り込み異常発生時：1	
903	R_14442 R_&M9030	MEWNET送受信 命令実行可	実行不可（実行中）：0 実行可：1
	R_14443 R_&M9031	MEWNET送受信 命令実行完了	正常終了：0 異常終了：1 異常コードはDT_9039にセットされる。
	R_14447 R_&M9035	メモリアクセス 命令実行可能	実行不可（実行中）：0 実行可：1
	R_14448 R_&M9036	メモリアクセス 命令完了	正常終了：0 異常終了：1

ワード No.	番 号	名 称	内 容
905	R_14480 R_&M9050	MEWNET 伝送 異常 (リンク1)	正常時 : 0 伝送異常または設定異常 : 1
	R_14481 R_&M9051	MEWNET 伝送 異常 (リンク2)	正常時 : 0 伝送異常または設定異常 : 1
	R_14482 R_&M9052	MEWNET 伝送 異常 (リンク3)	正常時 : 0 伝送異常または設定異常 : 1

ワード No.	番 号	名 称	内 容
906	R_14496 R_&M9060	P C リ ン ク 0 用 伝 送 保 証 メ モ リ ー 0	ユニット No. 1 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14497 R_&M9061		ユニット No. 2 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14498 R_&M9062		ユニット No. 3 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14499 R_&M9063		ユニット No. 4 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14500 R_&M9064		ユニット No. 5 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14501 R_&M9065		ユニット No. 6 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14502 R_&M9066		ユニット No. 7 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14503 R_&M9067		ユニット No. 8 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14504 R_&M9068		ユニット No. 9 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14505 R_&M9069		ユニット No.10 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14506 R_&M906A		ユニット No.11 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14507 R_&M906B		ユニット No.12 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14508 R_&M906C		ユニット No.13 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14509 R_&M906D		ユニット No.14 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14510 R_&M906E		ユニット No.15 の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14511 R_&M906F		ユニット No.16 の伝送正常時 : ON 伝送異常時または停止時 : OFF

ワード No.	番 号	名 称	内 容
907	R_14512 R_&M9070	P C リ ン ク 0 用 動 作 モ ー ド メ モ リ ー ／ 0	ユニットNo. 1のRUN時 : ON PROG.時 : OFF
	R_14513 R_&M9071		ユニットNo. 2のRUN時 : ON PROG.時 : OFF
	R_14514 R_&M9072		ユニットNo. 3のRUN時 : ON PROG.時 : OFF
	R_14515 R_&M9073		ユニットNo. 4のRUN時 : ON PROG.時 : OFF
	R_14516 R_&M9074		ユニットNo. 5のRUN時 : ON PROG.時 : OFF
	R_14517 R_&M9075		ユニットNo. 6のRUN時 : ON PROG.時 : OFF
	R_14518 R_&M9076		ユニットNo. 7のRUN時 : ON PROG.時 : OFF
	R_14519 R_&M9077		ユニットNo. 8のRUN時 : ON PROG.時 : OFF
	R_14520 R_&M9078		ユニットNo. 9のRUN時 : ON PROG.時 : OFF
	R_14521 R_&M9079		ユニットNo.10のRUN時 : ON PROG.時 : OFF
	R_14522 R_&M907A		ユニットNo.11のRUN時 : ON PROG.時 : OFF
	R_14523 R_&M907B		ユニットNo.12のRUN時 : ON PROG.時 : OFF
	R_14524 R_&M907C		ユニットNo.13のRUN時 : ON PROG.時 : OFF
	R_14525 R_&M907D		ユニットNo.14のRUN時 : ON PROG.時 : OFF
	R_14526 R_&M907E		ユニットNo.15のRUN時 : ON PROG.時 : OFF
	R_14527 R_&M907F		ユニットNo.16のRUN時 : ON PROG.時 : OFF

ワード No.	番 号	名 称	内 容
908	R_14528 R_&M9080	P C リ ン ク 1 用 伝 送 保 証 メ モ リ ー ／ O	ユニットNo. 1の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14529 R_&M9081		ユニットNo. 2の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14530 R_&M9082		ユニットNo. 3の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14531 R_&M9083		ユニットNo. 4の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14532 R_&M9084		ユニットNo. 5の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14533 R_&M9085		ユニットNo. 6の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14534 R_&M9086		ユニットNo. 7の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14535 R_&M9087		ユニットNo. 8の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14536 R_&M9088		ユニットNo. 9の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14537 R_&M9089		ユニットNo.10の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14538 R_&M908A		ユニットNo.11の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14539 R_&M908B		ユニットNo.12の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14540 R_&M908C		ユニットNo.13の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14541 R_&M908D		ユニットNo.14の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14542 R_&M908E		ユニットNo.15の伝送正常時 : ON 伝送異常時または停止時 : OFF
	R_14543 R_&M908F		ユニットNo.16の伝送正常時 : ON 伝送異常時または停止時 : OFF

ワード No.	番 号	名 称	内 容
909	R_14544 R_&M9090	P C リ ン ク 1 用 動 作 モ ー ド メ モ リ ー ／ O	ユニットNo. 1のRUN時 : ON PROG.時 : OFF
	R_14545 R_&M9091		ユニットNo. 2のRUN時 : ON PROG.時 : OFF
	R_14546 R_&M9092		ユニットNo. 3のRUN時 : ON PROG.時 : OFF
	R_14547 R_&M9093		ユニットNo. 4のRUN時 : ON PROG.時 : OFF
	R_14548 R_&M9094		ユニットNo. 5のRUN時 : ON PROG.時 : OFF
	R_14549 R_&M9095		ユニットNo. 6のRUN時 : ON PROG.時 : OFF
	R_14550 R_&M9096		ユニットNo. 7のRUN時 : ON PROG.時 : OFF
	R_14551 R_&M9097		ユニットNo. 8のRUN時 : ON PROG.時 : OFF
	R_14552 R_&M9098		ユニットNo. 9のRUN時 : ON PROG.時 : OFF
	R_14553 R_&M9099		ユニットNo.10のRUN時 : ON PROG.時 : OFF
	R_14554 R_&M909A		ユニットNo.11のRUN時 : ON PROG.時 : OFF
	R_14555 R_&M909B		ユニットNo.12のRUN時 : ON PROG.時 : OFF
	R_14556 R_&M909C		ユニットNo.13のRUN時 : ON PROG.時 : OFF
	R_14557 R_&M909D		ユニットNo.14のRUN時 : ON PROG.時 : OFF
	R_14558 R_&M909E		ユニットNo.15のRUN時 : ON PROG.時 : OFF
	R_14559 R_&M909F		ユニットNo.16のRUN時 : ON PROG.時 : OFF

6-1-3 特殊データメモリー一覧

BASICタイプCPUの内部で、あらかじめ用途が決まっているデータメモリです。このデータメモリは、読み出し専用なので、書き込みはできません。

番 号	名 称	内 容	備 考
DT_9000	自己診断コード		自己診断の結果が格納されます。
DT_9001	瞬停回数		瞬停の回数が格納されます。
DT_9002	ヒューズ断 ユニット	正常：0 異常：1	bit0～bit7～bit15 →スロット0→スロット7→スロット15
DT_9003	ヒューズ断 ユニット	正常：0 異常：1	bit0～bit7～bit15 →スロット16→スロット23→スロット31
DT_9004	異常高機能 ユニット	正常：0 異常：1	bit0～bit7～bit15 →スロット0→スロット7→スロット15
DT_9005	異常高機能 ユニット	正常：0 異常：1	bit0～bit7～bit15 →スロット16→スロット23→スロット31
DT_9010	照合異常 ユニット	正常：0 異常：1	bit0～bit7～bit15 →スロット0→スロット7→スロット15
DT_9011	照合異常 ユニット	正常：0 異常：1	bit0～bit7～bit15 →スロット16→スロット23→スロット31
DT_9017	演算エラー 行番号		最初の演算エラーが発生した行番号が格納されます。
DT_9020	ユーザー プログラム容量		PRM (0) に設定されたユーザープログラムの容量が格納されます。 単位：ワード。
DT_9021	ファイルメモリ 最大値		ファイルメモリ (FL) の最大番号が格納されます。
DT_9022	スキャンタイム		スキャンタイム(マルチタスクの1周期)の現在の値が格納されます。 例) 200→20ms以内225→22.5ms以内
DT_9023	スキャンタイム		スキャンタイム (マルチタスクの1周期) の最小値が格納されます。
DT_9024	スキャンタイム		スキャンタイム (マルチタスクの1周期) の最大値が格納されます。
DT_9029	ポーズ番号		テストRUN時のポーズしている行番号が格納されます。

番 号	名 称	内 容	備 考	
DT_9036	リモートI/O メモリアクセス 命令完了異常 コード	正常時 : 0		
		異常時	58H	タイムアウト : 相手先不在で送信不可状態
			68H	アクセスエリア無エラー : 指定したメモリアクセスエリアがスレーブユニット上に存在しない
			71H	送信アンサー待ちタイムアウトエラー
			72H	送信バッファアンサー空き待ちタイムアウトエラー
			73H	レスポンス待ちタイムアウトエラー
DT_9039	SEND/RECV 完了コード	正常終了 : 0	MEWNET データ転送命令 (SEND/RECIV)	

番 号	名 称	内 容	
DT_9131	異常子局の確認	上位8ビット：マスタユニット選択	0~3 (CPUに近い順)
		下位8ビット：表示選択	00H：異常子局表示 01H：I/O異常子局
DT_9132	異常子局 スロット	bit 0----- bit 15 スロット 0----- スロット 15	
		DT_9131：異常子局表示（現在値）	0：正常 1：異常
		DT_9131：I/O異常子局表示（I/O属性変化）	0：変化無 1：変化有
DT_9133	異常子局 スロット	bit 16----- bit 31 スロット 0----- スロット 15	
		DT_9131：異常子局表示（現在値）	0：正常 1：異常
		DT_9131：I/O異常子局表示（I/O属性変化）	0：変化無 1：変化有
DT_9134	異常子局 スロット	bit 0----- bit 15 スロット 0----- スロット 15	
		DT_9131：異常子局表示（累積値）	0：正常 1：異常
		DT_9131：I/O異常子局表示（瞬停止）	0：無 1：有
DT_9135	異常子局 スロット	bit 16----- bit 31 スロット 0----- スロット 15	
		DT_9131：異常子局表示（累積値）	0：正常 1：異常
		DT_9131：I/O異常子局表示（瞬停止）	0：無 1：有
DT_9136	リモートI/O 異常コード	ビット0~7：マスタユニット1	0：正常
		ビット8~15：マスタユニット2	1：異常
DT_9137	リモートI/O 異常コード	ビット0~7：マスタユニット3	0：正常
		ビット8~15：マスタユニット4	1：異常

番 号	名 称	内 容
DT_9140	P C リ ン ク ス テ ー タ ス	PCリンク0の受信回数RINGカウンタ
DT_9141		PCリンク0の受信間隔現在値 (* 2.5ms)
DT_9142		PCリンク0の受信間隔最小値 (* 2.5ms)
DT_9143		PCリンク0の受信間隔最大値 (* 2.5ms)
DT_9144		PCリンク0の送信回数RINGカウンタ
DT_9145		PCリンク0の送信間隔現在値 (* 2.5ms)
DT_9146		PCリンク0の送信間隔最小値 (* 2.5ms)
DT_9147		PCリンク0の送信間隔最大値 (* 2.5ms)
DT_9148		PCリンク1の受信回数RINGカウンタ
DT_9149		PCリンク1の受信間隔現在値 (* 2.5ms)
DT_9150		PCリンク1の受信間隔最小値 (* 2.5ms)
DT_9151		PCリンク1の受信間隔最大値 (* 2.5ms)
DT_9152		PCリンク1の送信回数RINGカウンタ
DT_9153		PCリンク1の送信間隔現在値 (* 2.5ms)
DT_9154		PCリンク1の送信間隔最小値 (* 2.5ms)
DT_9155	PCリンク1の送信間隔最大値 (* 2.5ms)	
DT_9160	リンク ユニットNo.	LINK1のユニットNo.が格納されます。
DT_9161	異常フラグ	LINK1の異常フラグが格納されます。
DT_9162	リンク ユニットNo.	LINK2のユニットNo.が格納されます。
DT_9163	異常フラグ	LINK2の異常フラグが格納されます。
DT_9164	リンク ユニットNo.	LINK3のユニットNo.が格納されます。
DT_9165	異常フラグ	LINK3の異常フラグが格納されます。

番 号	名 称	内 容
DT_9170		PC-LINKアドレス重複先
DT_9171		テストモード結果
DT_9172		トークン紛失回数
DT_9173		(2重トークン回数)
DT_9174		無信号状態回数
DT_9175		同期異常回数
DT_9176		送信NACK
DT_9177		送信NACK
DT_9178		送信WACK
DT_9179	L	送信WACK
DT_9180	I	送信アンサー
DT_9181	N	送信アンサー
DT_9182	K	未定義コマンド
DT_9183	1	パリティエラー回数
DT_9184	S	Endcode受信エラー
DT_9185	T	フォーマットエラー
DT_9186	エ	NOTサポート
DT_9187	ラ	自己診断結果
DT_9188	ス	ループ切り替え回数
DT_9189	テ	リンク不可状態発生回数
DT_9190	ー	主ルート入力断線回数
DT_9191	タ	副ルート入力断線回数
DT_9192	ス	ループ再構成処理中
DT_9193		ループ運転モード
DT_9194		ループ入力状態

番 号	名 称	内 容
DT_9200		PC-LINK アドレス重複先
DT_9201		テストモード結果
DT_9202		トークン紛失回数
DT_9203		(2重トークン回数)
DT_9204		無信号状態回数
DT_9205		同期異常回数
DT_9206		送信 NACK
DT_9207		送信 NACK
DT_9208		送信 WACK
DT_9209	L	送信 WACK
DT_9210	I	送信アンサー
DT_9211	N	送信アンサー
DT_9212	K	送信アンサー
DT_9213	2	未定義コマンド
DT_9214	S	パリティエラー回数
DT_9215	T	Endcode 受信エラー
DT_9216	E	フォーマットエラー
DT_9217	S	NOT サポート
DT_9218		自己診断結果
DT_9219		ループ切り替え回数
DT_9220		リンク不可状態発生回数
DT_9221		主ルート入力断線回数
DT_9222		副ルート入力断線回数
DT_9223		ループ再構成処理中
DT_9224		ループ運転モード
DT_9225		ループ入力状態

番 号	名 称	内 容
DT_9230		PC-LINKアドレス重複先
DT_9231		テストモード結果
DT_9232		トークン紛失回数
DT_9233		(2重トークン回数)
DT_9234		無信号状態回数
DT_9235		同期異常回数
DT_9236		送信 NACK
DT_9237		送信 NACK
DT_9238		送信 WACK
DT_9239	L	送信 WACK
DT_9240	I	送信アンサー
DT_9241	N	送信アンサー
DT_9242	K	未定義コマンド
DT_9243	3	パリティエラー回数
DT_9244	ス	Endcode受信エラー
DT_9245	テ	フォーマットエラー
DT_9246	丨	NOTサポート
DT_9247	タ	自己診断結果
DT_9248	ス	ループ切り替え回数
DT_9249		リンク不可状態発生回数
DT_9250		主ルート入力断線回数
DT_9251		副ルート入力断線回数
DT_9252		ループ再構成処理中
DT_9253		ループ運転モード
DT_9254		ループ入力状態

6 - 2

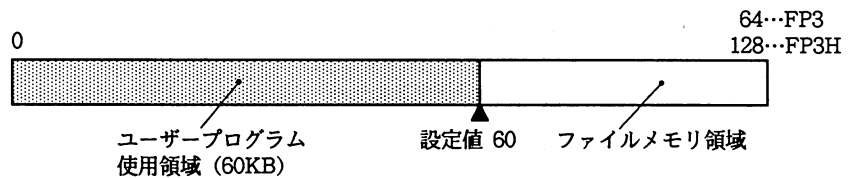
パラメータメモリ設定一覧

No.	説明	デフォルト値	表示設定範囲の説明
■プログラム容量の設定			
0	ユーザープログラム容量の設定	(FP3) 64 (FP3H) 128	(FP3) 16~64kバイト *1 (FP3H) 16~128kバイト *2
■I/O、メモリの保持・非保持の属性の設定			
7	メモリI/Oの保持エリアの設定	60	0~98 (ワード単位で設定可) *2
8	データメモリの保持エリアの設定	0	0~2048 (1点単位で設定可) *2
9	ファイルメモリの保持エリアの設定	0	(FP3) 0~24576 ワード *2 (FP3H) 0~57344 ワード *2
10	PCリンク0用リンクメモリI/O保持エリアの設定	0	0~64 (ワード単位で設定可) *2
11	PCリンク1用リンクメモリI/O保持エリアの設定	64	64~128 (ワード単位で設定可) *2
12	PCリンク0用リンクデータメモリ保持エリアの設定	0	0~128 (1点単位で設定可) *2
13	PCリンク1用リンクデータメモリ保持エリアの設定	128	128~256 (1点単位で設定可) *2
15	出力の保持・非保持の設定	1	非保持:0 保持:1 *2
16	プログラムモードでの出力の保持・非保持の設定	1	0:非保持 1:保持 *2
19	出力の保持・非保持の設定方向 設定値以降を保持	1	0:設定値未満を保持 *2 1:設定値以上を保持 *2
■異常時の運転モード			
21	出力ヒューズ断時	0	0:停止 1:運転
22	特殊ユニット異常時	0	0:停止 1:運転
23	I/O照合の異常時	0	0:停止 1:運転
26	エラーの発生時の処置	0	0:停止 1:運転
27	リモートI/O交信異常時の処置	0	0:停止 1:運転
28	リモート子局のユニット異常時の処置	0	0:停止 1:運転
32	MEWNET SEND/RECV 命令待ち時間	800	4~32760 (10msec~81.9sec) (2.5msec × 設定値)
35	リモートI/O子局接続待ちモード設定 (接続を待って運転開始)	1	0:解除 (接続を待たずに運転開始) 1:設定 (接続を待って運転開始)
■PCリンク0のリンクエリアの割り付け			
40	リンクメモリI/Oリンク範囲の設定	0	0~64 ワード *3
41	リンクデータメモリリンク範囲の設定	0	0~128 ワード *3
42	リンクメモリI/O送信開始ワードNo.の設定	0	0~63
43	リンクメモリI/O送信ワード数の設定	0	0~64
44	リンクデータメモリ 送信開始ワードNo.の設定	0	0~127
45	リンクデータメモリ 送信ワード数の設定	0	0~127

No.	説 明	デフォルト値	表示設定範囲の説明
■ PCリンク1のリンクエリアの割り付け			
50	リンクメモリI/Oリンク範囲の設定	0	0~64ワード * 3
51	リンクデータメモリリンク範囲の設定	0	0~128ワード * 3
52	リンクメモリI/O送信開始ワードNo.の設定	64	64~127
53	リンクメモリI/O送信ワード数の設定	0	0~64
54	リンクデータメモリ送信開始ワードNo.の設定	128	128~255
55	リンクデータメモリ送信ワード数の設定	0	0~127

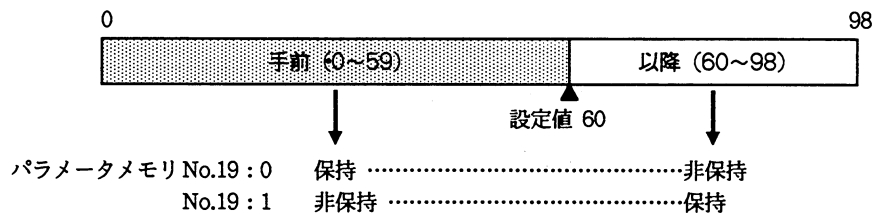
* 1 : 設定した使用領域以外は、ファイルメモリ (FL) として使用されます。

例



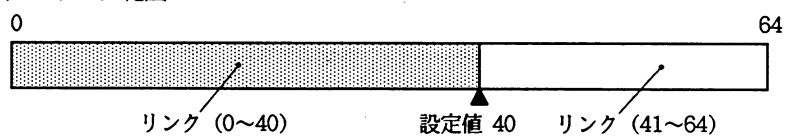
* 2 : 指定値の手前までを保持にするか、指定値以降を保持にするかを選択できます。パラメータメモリ No.19の値で設定します。

例 7.メモリI/Oの保持エリア



* 3 : 実際に設定されるリンクエリア範囲は、設定範囲の小さい方から設定値までがリンクエリアとなります。

例 40.リンクI/Oリンク範囲



6 - 3

エラーコード一覧

6-3-1

自己診断エラーコード一覧

PCのCPUユニット本体からのエラーメッセージです。

番号	内容
23	RAM異常1エラー
25	RAM異常2エラー
26	ユーザーROMサムチェックエラー
27	特殊ユニット装着制限エラー
28	パラメータメモリ異常エラー
29	メモリ管理テーブル異常エラー
30	割込異常1エラー
31	割込異常2エラー
33	マルチパラメータ照合チェックエラーコード
35	スレーブ上に禁止ユニットが実装
36	リモートI/O仕様制限エラー（重複あるいは範囲オーバー）
38	I/Oターミナルボード登録エラー
40	出力ユニットヒューズ断エラー
41	特殊ユニット暴走エラー
42	I/O照合エラー
45	演算エラー発生
46	リモートI/O交信エラー
47	スレーブ上のI/O属性エラー
50	電池電圧低下エラー
51	リモートI/O終端局エラー
53	ラダー用I/Oマップ照合チェックワーニングコード
60	ESB受信バッファオーバーフローエラー

6-3-2 FP - BASIC エラーコード一覧

FP - BASIC インタープリタからのエラーメッセージです。

エラーコード	内 容
1	NEXT 文に対応する FOR 文がない
2	文法エラーが発生した
3	GOSUB 文がないのに RETURN 文を実行しようとした
4	プログラム中に GOTO 文、GOSUB 文が合わせて 448 個以上ある または、ラベルが 410 個以上ある
5	関数のパラメータ値が規定範囲外である
6	数字や変数の演算オーバーフローが発生した
7	GOSUB 文のネスト・オーバーが発生した
8	GOTO、GOSUB で呼ばれた行番号がない
9	宣言していない配列変数を使おうとした または、配列の大きさが定義サイズ越えた (配列領域オーバー)
10	変数の 2 重定義をしようとした
11	0 で割った
12	コマンドで使用すべき命令をステートメントとして使用した
13	左括弧と右括弧の数があわない
14	パラメータの数があわない (配列で配列要素があわない)
15	DIM で、定義されていない関数、変数または、配列を使おうとした
16	FUNCTION ~ FEND の間以外にプログラムがある
17	FUNCTION ~ FEND の中で FUNCTION 宣言をした
18	変数宣言文が行の先頭がない
21	変数の型変換エラー
22	パラメータの設定テーブルが設定範囲を越えた
23	1 行の文字数が多すぎる
24	データタグエラー (未定義中間コード命令エラー) (未定義コマンド実行) (パラメータタイプ不相当)
25	数値変数エラー 桁が多い (INPUT 文による ASCII 文字が数値に正しく変換できない)
26	存在しない機器を使用しようとした (DEBUG # n を使用しない場合)

エラーコード	内 容
27	存在しない入出力番号を指定した
28	FOR ループに NEXT 文がない
30	INPUT 文によるデータの受信数があわない
31	通信タイムアウトエラー
32	SRC - 42 が立ち上がっているのに RIIOC の準備ができていない
33	バッファオーバーフロー
34	パリティオーバーランフレーミングエラー
36	受信時のバッファオーバーフロー (入力が 80 文字を越えた)
39	プログラム番号が見つからない
40	指定タスクが存在しない
41	タスクが起動できない (実行タスク)
45	タスクの実行中に許されないコマンドを実行した
53	ファイルが存在しない
57	同じ名前のファイルが既に存在している
60	ファイルメモリに空きスペースがない
62	ファイルの名前が正しくない
63	ディスクがレディでない
66	ディスクライトエラー
67	ドライブセレクトエラー
68	書き込みが禁止されているディスクに書き込もうとした
70	GOSUB 文がないのに RETURN 文を実行しようとした
71	数値オーバーフロー
73	パラメータ数値が範囲外
75	中間コードエラー (未定義タグデータエラー) (宣言または定義されていない関数、変数を使用しようとした)
79	ENDIF 文に対応する IF 文がない
80	ELSE 文に対応する IF 文がない
81	IF 文に対応する ENDIF 文がない

エラーコード	内 容
83	関数や変数名が長すぎる (中間コード定義領域がない)
84	中間コードの格納領域が足りない
85	EXITの出口がない
86	システムワークのメモリチェックエラー
87	ファイルメモリのメモリチェックエラー
88	DO文に対応するLOOP文がない
89	LOOP文に対応するDO文がない
93	関数に与えるパラメータの数があわない
100	デバイス通信エラー
135	タグが異常
136	与えられた関数が存在しない
258	オブジェクトが存在しない
259	ラベル情報が大きすぎる
260	FUNCTION文が大きすぎる
261	メモリーが足りない
1002	タスク起動中のXQT
1003	HALT中の再起動異常
1004	タイムアウト
1005	指定されたシンボルがない
1006	モードが違う
1007	スタックが空
1008	コマンドの使用方法が間違っている
1009	パラメータが間違っている (DT_、LD_のビット操作)
1010	未定値演算
1011	割り込み中のサブルーチンコール
1012	指定タスクがPAUSE状態ではない
1013	メモリーROMエラー

エラーコード	内 容
1014	メモリープロテクトエラー
1015	MEWNET データリンクエラー
1016	MEWNET データリンクがアクティブではない
1017	指定されたスロットに共有RAMはありません
1018	WAITで他のデバイスの入力指定された
1019	WAIT I/OのI/O数が多すぎる
1020	タスク指定が不適當
1098	I/O種別エラー
1099	メモリアロケーションエラー
1101	リモートI/Oの設定異常
1200	数値をあらわす文字列として正しくない
1201	変数のタイプが等しくない
1202	データを最後まで読み込んだ後、再びREAD文を実行した
1203	データ文が見つからない
1250	親局が見つからない
1251	親局が最大親局数 (4) を越えている
1252	子局が最大子局数 (32) を越えている
1253	スロットが最大スロット数を越えている
1254	スロット機能番号が正しくない
1255	親局の総スロット数が最大親局総スロット数 (64) を越えている
1256	全体の総スロット数が最大 (全体) 総スロット数 (128) を越えている
1257	リモートI/Oのメモリアクセスエラー
1300	ユニットIDが正しくない
1301	ユニット種別が正しくない
1302	チャンネル番号が正しくない
1303	平均回数が正しくない
1304	終端コードタイプが正しくない

エラーコード	内 容
1305	入力フィルタ定数が正しくない
1306	ユニットタイプが正しくない
1307	軸番号が正しくない
1308	JOB 番号が正しくない
1309	始動データ No.が正しくない
1310	データ長が正しくない
1311	ID アドレスが正しくない
1312	ID 番号が正しくない
1313	クリア ID 番号が正しくない
1314	エラーレスポンスを受け取った
1315	ハードウェアの制限 (定義) によりこのステートメント (パラメータ) ... ???
1316	高機能ユニットエラー
1400	ファイル名の指定が間違っている
1401	オープンしていないファイルを参照しようとした
1402	既にオープンされているファイルを再度オープンしようとした
1403	ファイル中のすべてのデータを読み尽くした後、更に読み込もうとした
1404	ファイルがオープンできない
2000	致命的エラー
2002	I/O1 度読み異常
2003	I/O 点数異常
2004	ベーシックスタックオーバー
2005	ONERR 処理中のエラー
2006	IF レベルオーバー
2007	内部タグコードエラー
3000	通信シーケンスエラー

6 - 4

スロット割り付けの基礎知識

各スロットのスロット番号およびI/O番号は、マザーボードに装着したユニットの位置により、自動的に設定されます。また、その設定を、SLOTコマンドを使用して、変更することもできます。

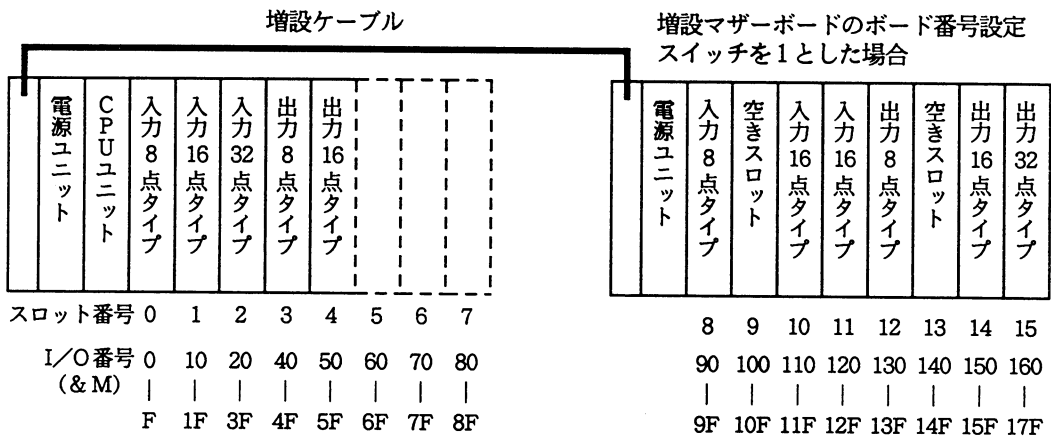
6-4-1

フリーロケーション機能を使用するスロット割り付け

FP-BASICからSLOTCLRコマンドを実行すると、各スロットのI/Oユニットのタイプを調べ、スロット番号およびI/O番号を自動的に割り付けます。これを、フリーロケーションと呼びます。

■フリーロケーションによるスロット割り付けの内容

1. 8点ユニットには16点分のI/O番号が割り付けられます。
2. 空きスロットには、16点分のI/O番号が割り付けられます。
3. リンクユニットなどのI/Oに関係のないユニットには、16点分のI/O番号が割り付けられます。
4. 内部的にI/O番号を持つユニットには、そのユニットが占有するI/O番号が割り付けられます。



電源ユニット	CPUユニット	リンクユニット	空きスロット	シリアルデータユニット	A/D変換ユニット	D/A変換ユニット	高速カウンタユニット		
スロット番号	0	1	2	3	4	5	6	7	
I/O番号 (&M)	0	10	20	40	50	60	70	80	
	F	1F	3F	4F	5F	6F	7F	8F	

6-4-2 スロットコマンド機能を使用するスロット割り付け

FP-BASICのSLOTコマンドを使用して、スロット番号およびI/O番号を自由に割り付けることができます(任意割り付け)。事前にフリーロケーション機能(SLOTCLRコマンド)で自動割り付けした場合は、SLOTコマンドで再割り付けしたスロット番号およびI/O番号以降の割り付け情報はシフトされます。

■ SLOTコマンドによるスロット割り付けの内容

1. 各スロットに装着されるユニットの種類を、汎用I/Oユニット、高機能ユニットの中から選択します。
2. 各スロットのI/O点数は、0点、8・16点、32点、64点、128点の中から選択します。リンクユニットなどのI/Oを持たないユニットには、0点を選択することでI/O番号の空きをなくすことができます。
3. I/O点数は、正しく設定してください。I/O点数を実際に装着されているユニットの点数よりも少なく設定すると、そのユニットで使用できるI/O点数が減少します。

注意

- ユニット交換時には、SLOTCLRコマンドまたはSLOTコマンドを使用して必ずスロット割り付けしてください。スロット割り付けを行わないと正常な動作が得られないことがあります。
- ユニット交換によるスロット番号の変更を最小限にするには、交換の可能性のあるユニットをマザーボードの外側(8スロットタイプの場合は8スロット目)に装着します。

6-4-3 ユニットのI/O点数と機能番号

ユニット品名	注文品番	入力点数	出力点数	機能番号
空ユニット		-	-	302
8・16点入力ユニット		8/16	-	120
32点入力ユニット		32	-	130
64点入力ユニット		64	-	140
128点入力ユニット		128	-	150
8・16点出力ユニット		-	8/16	102
32点出力ユニット		-	32	103
64点出力ユニット		-	64	104
128点出力ユニット		-	128	105
A/D変換ユニット	AFP3400	16	-	220
D/A変換ユニット	AFP3410/ 3411	16	-	220
シリアルデータユニット	AFP3460	16	16	222
パルス出力ユニット	AFP3480	16	16	222
高速カウンタユニット	2CHタイプ AFP3421 4CHタイプ AFP3422	16	16	222 222
位置決めユニット	1軸タイプ AFP3431 Fタイプ 2軸タイプ AFP3432	32	32	233 233
割り込みユニット	AFP3452	16	-	220
ID/Xコントロールユニット	1CHタイプ AFP3470 2CHタイプ AFP34701 1CHタイプ+RS232C AFP34702	16	16	222 222 222
マイクロ波I/Dコントロールユニット	AFP3471	16	16	222
バーコードリーダコントロールユニット	AFP3461	-	-	402
データプロセスユニット	AFP3461	16	16	222
リモートI/Oマスタユニット	AFP3740	-	-	402

*実際にI/Oを持たないユニットは、SLOTコマンドを使用して、I/O占有点数を0として割り付けることができます。

6 - 5

プログラム実行時間

プログラム中の各命令（行番号の処理も1命令として扱います）の実行時間は、命令により異なりますが、およそ0.1ms~2.0msです。ただし、マルチタスクプログラムの場合は、同時に実行するタスクの数が多くなるにしたがって、命令の実行時間は遅くなります（タスクが2倍になると、実行時間も約2倍になります）。以下、代表的な命令の実行時間（実測値）を示します。

実行時間による制限をシビアに受けるプログラムを開発する場合は、テスト実行を繰り返し、安全な時間配分を行ってください。

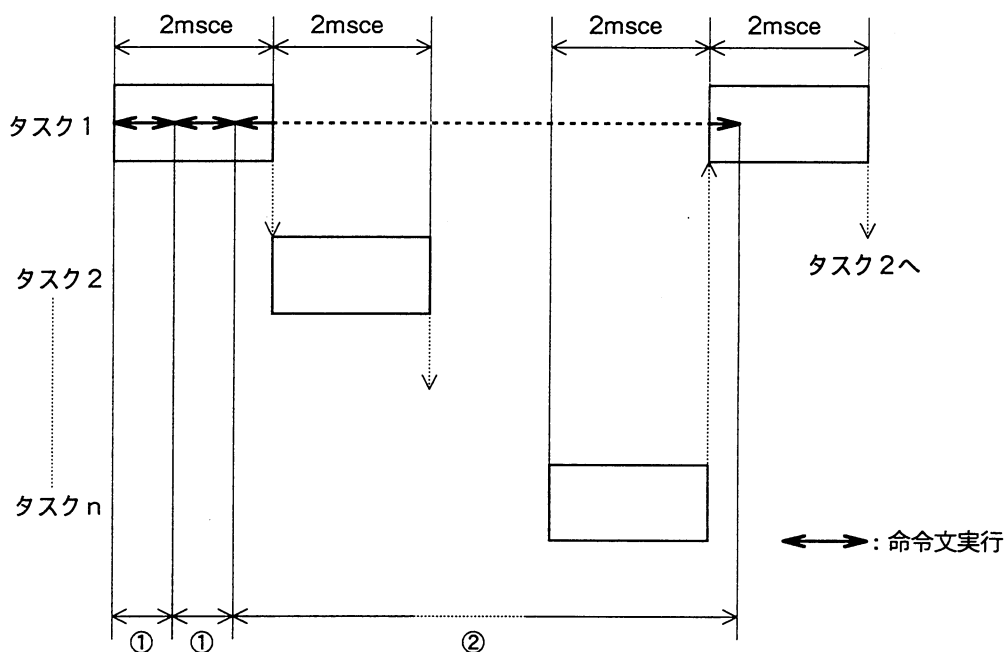
命令文	平均実行時間 *(ms)	
	FP3H BASIC (高速タイプ)	FP3 BASIC (従来タイプ)
ON Y_0	0.24	1.02
ON Y_0;OFF Y_0 (2命令でのパルス出力)	0.49	2.04
WAIT SW(X_0)=1 (WAIT条件文が既に成立していた場合)	1.23	4.63
WAIT SW(X_0)=1 AND SW(R_1)=1 (WAIT条件文が既に成立していた場合)	2.03	7.90
GOTO 100	0.13	0.56
FOR I=1 TO 10:NEXT I (FOR~NEXT 1回あたり)	0.98	4.02
IF SW(R_0)=1 THEN ON Y_0 (IF条件が成立した場合)	0.89	3.70
IF SW(R_0)=1 THEN ON Y_0 (IF条件が成立しなかった場合)	0.61	2.53
I=LSHIFT(I, 1) (I...2バイト整数)	0.41	1.72
A\$=RIGHT("ABCDEF6", 2)	0.47	1.93

* この平均実行時間はシングルタスク（同時に実行するタスクの数 = 1）の場合です。

マルチタスク使用時のプログラム実行時間

マルチタスク使用時は、命令の実行を開始するタイミングによって、ひとつの命令の実行を開始してから終了するまでの時間が変わります。

マルチタスク使用時は、各タスクを2msec間ずつ順次切り替えて実行していきます。ひとつのタスクについてみれば、2msec間実行～中断（他のタスクを実行）～2msec間実行の繰り返しになります。このため、ひとつの命令文を実行している途中で、次のタスクに切り替わる場合があります。この場合は、ひとつの命令文が、複数回の処理時間にまたがって処理されます。



①1回の処理時間内に実行が終了する命令文の場合（シングルタスクの場合の実行時間を参考にしてください）

②1回の処理時間内に命令文の実行が終了せず、途中で実行が中断し、次回の処理時間を待って続きを実行する場合（タスクの数が増えるにしたがって、ひとつの命令の実行を開始してから終了するまでの時間は長くなります）

6 - 6

ASCII/JIS キャラクターコード一覧

上位 下位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		DEL	スペース	0	@	P	'	p			スペース	ー	タ	ミ		
1	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
2	STX	DC2	"	2	B	R	b	r			「	イ	ツ	メ		
3	ETX	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
4	EOT	DC4	\$	4	D	T	d	t			、	エ	ト	ヤ		
5	ENT	NAK	%	5	E	U	e	u			・	オ	ナ	ユ		
6	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	BEL	ETB	'	7	G	W	g	w			ァ	キ	ヌ	ラ		
8	BS	CAN	(8	H	X	h	x			ィ	ク	ネ	リ		
9	HT	EM)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A	LF	SUB	*	:	J	Z	j	z			ェ	コ	ハ	レ		
B	VT	ESC	+	;	K	[k	{			ォ	サ	ヒ	ロ		
C	FF	→	,	<	L	¥	l	!			ャ	シ	フ	ワ		
D	CR	←	-	=	M]	m	}			ュ	ス	ヘ	ン		
E	SO	↑	.	>	N	^	n	~			ョ	セ	ホ	”		
F	SI	↓	/	?	O	_	o				ッ	ソ	マ	°		

6 - 7

索引

英数記号

2進表記	83
8進表記	83
10進表記	83
16進表記	83
& M表記	83
A/D変換ユニット	115
ASCII/JIS キャラクタコード一覧	152
ATOK7	41
AUTOEXEC.BAT	23
AUTOモード	56
CALL ~ FUNCTION文	102
CONFIG.SYS	23
CPUの設定	21
CPUの動作モード	68
D/A変換ユニット	115
DOループ文	101
FORループ文	100
FP - BASICの起動	27,33,35,46
FP - BASICの終了	46
FUNCTION文	90
GOSUB文	95
GOTO文	94
I/O・メモリ構成一覧	124
I/Oのセーブ	76
I/Oのロード	76
I/Oポートアクセス命令	107
I/O番号	146
IFB条件文	99
IF条件文	98
MEWNET データ転送命令	112
MS - DOS	22
N88BASIC一般命令	106
ON GOTO/GOSUB文	96
RAMドライブ	28
REM文	79
RS232C通信ケーブル	20
SELECT ~ CASE文	97
VJE β	43
WHILEループ文	101

あ

アップロード	65
異常時の運転モード	138
位置決めユニット	117
一太郎 Ver.4	41
一括ダンプ	74
イニシャライズ	52
インストール	22
インターロック	92
ウィンドウダンプ	74
エラーコード一覧	140
エラー割り込み	104
演算子	89
演算子の優先順位	89
オブジェクトファイル	65
オブジェクトプログラム	64
オブジェクトリンク	122
オペランド	78
オペレーションシステム	14

か

型変換	88
かな漢字変換	22,27,30,35,40,41
間接実行	78
キー操作	47
機器の接続	20
記号	80
起動ディスクの作成	24,28
起動時のコマンドオプション	23
機能番号	148
共有メモリ	75
共有メモリのセーブ	76
共有メモリのロード	76
強制入出力	75,76
行	79
行の移動	58
行の挿入と削除	57
行番号	56,78
組み込み関数	108
クロック周波数	21
グローバル変数	53,86
ケーブル	14

高機能ユニット入出力命令	114
高速カウンタユニット	117
コメント	79
コンパイル	64

さ

三角関数	108
算術演算	89
式	89
システム関数	107
システム変数	107
初期設定	50
シリアルデータユニット	116
シンボリックファイル	65
辞書ディスク	27,32
状態表示	74
数値関数	108
数値定数	83
数値配列	87
数値文字変換関数	109
ステータスのリアルタイム表示	69
ステータス表示	68
スロット番号	146
スロット割り付け	54,75,146
セーブ	60
制御構文	90
ソフトウェアの準備	22

た

対数関数	108
タイマ	107
タイマ経過値	73
タスク名	90
ダウンロード	65
ダンプ	72,74
注釈文	79
直接実行	78
通信ボーレート	21
定数	82
テストモード	67
ディップイッチ	21
ディスク	24
ディスクファイルアクセス命令	106

特殊データメモリー一覧	131
特殊メモリーI/O一覧	125
特殊記号	80
トレースモード	66
動作環境	14

な

入力割り込み	103
任意割り付け	55,146
ネスティング	99
ノートパソコンへのインストール	28

は

ハードウェアの準備	20
ハードディスクへのインストール	33
配列	87
バージョン表示	50
バックアップ	38
パソコンの設定	21
パラメータメモリ	51,75
パラメータメモリ設定一覧	138
パルス出力ユニット	116
比較演算	89
ビット演算	89
ファイルの一覧	61
ファイルの消去	61
ファイルメモリ	53,138
ファイル管理	61
ファイル選択ウィンドウ	62
ファイル名を変更	61
ファンクションキー	48
複文	79
フリーロケーション	54,146
フロッピーディスクの初期化	36
フロッピーディスクの複写	38
フロッピーディスクへのインストール	24
分割コンパイル	120
プログラミング機能	16
プログラムエリア	53
プログラム実行時間	150
プログラムの実行	66
プログラムメモリ	53,56
プログラム編集	56

変数	84
変数エリア	53
変数宣言	84
変数操作命令	106
変数の型	84
変数のセーブ	76
変数のロード	76
変数領域	53
変数領域の割り付け	53
保持・非保持	139
保存	60

ま

松 Ver.5	42
松茸 V3	42
マルチステートメント	79
マルチタスク	18,90
マルチポイントモニタ	71
ミュー表記	83
命令語	78
命令文	78
メモリの初期化	50
メモリ不足	23,35
文字	80
文字関数	109
文字削除	57
文字修正	57
文字挿入	57
文字定数	83
文字列検索	59
文字列配列	87
モニタ	17,68

や

ユーザプログラム容量	138
ユニットのI/O点数と機能番号	148
読み出し	60
予約語	81

ら

ラベル	79
リモートI/Oシステム	110,114
リモートモード	66
リンクエリアの割り付け	138
ローカル変数	53,86
ロード	60
論理演算	89

わ

割り込み制御	103
割り込み設定の表示	73
割り込み番号	75,105
割り込みユニットからの割り込み	105

マニュアル改訂履歴

マニュアル番号は、表紙下に記載されています。

発行日付	マニュアル番号	改訂内容
1990年11月	FAF - 44	初版
1991年 8月	FAF - 44①	2版 誤記訂正
1992年 7月	FAF - 44②	3版 バージョン・アップ(Ver.2.0)に対応
1993年 7月	FAF - 44③	4版

●このマニュアルに使われている用紙は古紙配合率100%の再生紙を使用しております。
●この印刷物は環境にやさしい植物性大豆油インキを使用しています。



古紙配合率100%再生紙を使用しています



大豆油を主成分としたインキで印刷しています

●在庫・納期・価格など販売に関するお問い合わせは

●技術に関するお問い合わせは

制御機器コールセンター

☎ 0120-101-550

※お問い合わせ商品 / リレー・機器用センサ・スイッチ・コネクタ・
プログラマブルコントローラ・プログラマブル表示器・
画像処理装置・タイマ・カウンタ・温度調節器

※サービス時間 / 9:00-17:00 (11:30-13:00、当社休業日除く)

●FAX 06-6904-1573 (24時間受付)

松下電工株式会社 制御機器本部
制御デバイス事業部

〒571-8686 大阪府門真市門真1048

TEL.(06)6908-1131<大代表>

©Matsushita Electric Works, Ltd. 2006
本書からの無断の複製はかたくお断りします。

このマニュアルの記載内容は平成5年7月現在のものです。