

NECパーソナルコンピュータ  
PC-9800シリーズ

**NEC**

# Software Library

N<sub>88</sub>-日本語BASIC(86)(Ver 6.1)  
ユーザーズマニュアル

PC-9801-BU04



# Software Library

N88-日本語BASIC(86)(Ver 6.1)  
ユーザーズマニュアル

**ご注意**

- (1) 本書の内容の一部または全部を無断転載することは禁止されています。
- (2) 本書の内容に関しては将来予告なしに変更することがあります。
- (3) 本書は内容について万全を期して作成いたしましたが、万一御不審な点や誤り、記載もれなどお気づきのことがありましたら御連絡下さい。
- (4) 運用した結果の影響について(3)項にかかわらず責任を負いかねますので御了承下さい。

© 1988, 89 NEC Corporation

MS-DOS は米国マイクロソフト社の商標です。

386は米国インテル社の商標です。

**輸出する際の注意事項**

本製品（ソフトウェアを含む）は、外国為替および外国貿易管理法の規定により、戦略物資等輸出規制品に該当します。従って、日本国外に持出す際には日本国政府の輸出許可申請等必要な手続きをお取り下さい。

日本電気株式会社の許可なく複製・改変などを行うことはできません。

## はじめに

PC-9800シリーズパーソナルコンピュータには標準プログラム言語としてN<sub>88</sub>-BASIC(86)が用意されています。

このBASICを使ってプログラミングすることにより、手軽にコンピュータの機能を利用することができます。

本書は、N<sub>88</sub>-BASIC(86)に用意されている機能を詳しく解説したものです。

必要に応じて該当項目をお読みください。

なお、BASICの個々の命令・関数の詳細についてはこのマニュアルでは解説していません。『N<sub>88</sub>-日本語BASIC(86)リファレンスマニュアル』(以下『BASICリファレンスマニュアル』と略します)に詳しく説明されていますので併読されることをおすすめします。

また、基本的なプログラミングの方法については『N<sub>88</sub>-日本語BASIC(86)入門』(以下『BASIC入門』と略します)をご覧ください。

また、日本語入力の具体的な方法については、『日本語入力ガイド』を参照してください。

## このマニュアルの構成

本書は次のような構成になっています。

### 目的別索引

N<sub>88</sub>-日本語 BASIC(86)を使って何かを行いたいときに本書のどの章,どの項目を読めばよいかを示しています。

### 第1章 N<sub>88</sub>-BASIC(86)の起動

N<sub>88</sub>-BASIC(86)システムの種類とそれぞれの起動方法,およびDISKモードBASICにおいて選択可能な機能について解説しています。

### 第2章 キーボードとスクリーンエディタ

PC-9800シリーズのキーの働きとスクリーンエディタを使ったプログラム編集の方法を解説しています。

### 第3章 画面

テキスト画面とグラフィック画面の種類とその選択方法,および画面ハードコピーについて解説しています。

### 第4章 グラフィックス

グラフィック画面の座標系,パレット機能,各種描画命令,タイリング,画像の転送など,グラフィックスの制御方法を解説しています。

### 第5章 日本語処理

BASICの日本語処理機能について解説しています。

### 第6章 ファイル

ファイルの概念,ディスクファイルの扱い方,ディスクファイルの操作法など,ファイルの取り扱いについて解説しています。

### 第7章 入力装置

マウスおよびライトペンをプログラム上で使う方法を解説しています。

### 第8章 機械語プログラム

機械語プログラムをBASICから実行させる場合の準備と実行方法について解説しています。

## 第9章 拡張機能

電話制御機能, サウンド制御機能, GP-IB (IEEE-488) インタフェース制御機能, RS-232C インタフェース (2 回線 / 3 回線) 制御機能などの拡張機能について解説しています。

## 第10章 ユーティリティプログラム

BASIC上で重要な作業を行うユーティリティプログラムの使用方法を解説しています。

## 第11章 ターミナルモードとモニタモード

PC-9800シリーズを他のコンピュータの端末にするターミナルモードと, 機械語プログラム作成の手助けとなるモニタモードについて解説しています。

## 付 録

BASICを使用する上で必要な情報を図や表を使ってまとめて解説しています。



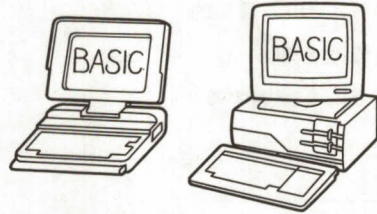
# 目的別索引

この目的別索引は、BASIC を使って何かを行いたいときに、どの章、どの項目を読めばよいか分かるようになっています。目的に応じて参照してください。

参照箇所中、二重カギカッコ(『 』)で囲まれているのは、他のマニュアルを示します。

なお、初めて BASIC を利用される方は、『BASIC 入門』をまずお読みください。

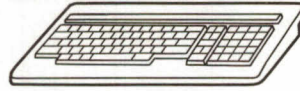
## BASIC を起動する



ROM モード BASIC と DISK モード BASIC のどちらかを起動させることができます。DISK モード BASIC の起動方法はフロッピーディスクのみのマシン構成の場合と固定ディスクのあるマシン構成の場合では違いがあります。

| 目 的                  | 参 照 箇 所                                |
|----------------------|--|
| ROM モード BASIC を起動する  | → 「1.1 ROM モード BASIC の起動」              |
| DISK モード BASIC を起動する | → 「1.2 DISK モード BASIC の起動」             |
| フロッピーディスクから起動する      | → 1.2 の「システムディスクからの起動」                 |
| 固定ディスクから起動する         | → 1.2 の「固定ディスクからの起動」                   |
| プログラムを自動的に実行させる      | → 1.2 の「オートスタート」<br>「10.5 ID セクタの書き換え」 |
| リセットする               | → 「1.3 リセットとウォームリスタート」                 |
| BASIC を終了する          | → 『BASIC 入門』                           |

## キーボードを使用する



PC-9800 シリーズのキーボードには文字キーのほか特殊な用途を持つキーがあり、BASIC 上で利用できます。また、このキーを使用して BASIC プログラムを編集するスクリーンエディタ機能が用意されています。

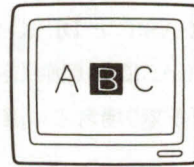
### 目的

キーの機能を知る  
プログラムを編集する

### 参照箇所

→ 「第2章 キーボードとスクリーンエディタ」  
→ 「2.4 スクリーンエディタ」  
『BASIC 入門』

## テキスト画面の状態を設定する



文字を表示するための画面がテキスト画面です。表示する文字とテキスト画面を制御するさまざまな機能が用意されています。

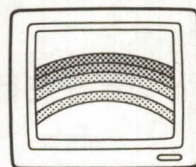
### 目的

表示文字の行数、桁数を設定する  
スクロールする範囲を設定する  
ファンクションキーを表示する  
文字の色を変える  
文字を点滅、反転表示する

### 参照箇所

→ 3.1 の「桁数と行数」  
→ 3.1 の「スクロール画面の範囲」  
→ 3.1 の「ファンクションキーの表示」  
→ 3.1 の「テキスト画面の文字の色と表示モード」  
3.1 の「カラーモード／白黒モード」  
→ 3.1 の「テキスト画面の文字の色と表示モード」

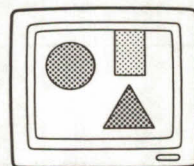
## グラフィック画面の状態を設定する



グラフィックスを描く画面がグラフィック画面です。カラーのモード／白黒のモードを選択したり、分解能の違うモードを選択することができます。

| 目的                 | 参照箇所            |
|--------------------|-----------------|
| カラー／白黒のモードを選択する    | → 3.2の「画面モード」   |
| 高分解能／低分解能のモードを選択する | → 3.2の「画面モード」   |
| 4096色の中から16色を使う    | → 4.3の「パレットモード」 |
| 4096色の中から8色を使う     | → 4.3の「パレットモード」 |
| 8色の中から8色を使う        | → 4.3の「パレットモード」 |
| 描画する画面、表示する画面を設定する | → 3.2の「ページ」     |

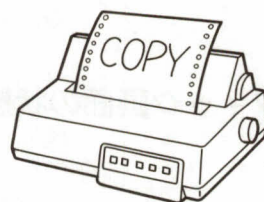
## グラフィック画面を制御する



グラフィック画面を制御するためにさまざまな機能が用意されています。

| 目的                  | 参照箇所                 |
|---------------------|----------------------|
| 座標上の表示する範囲を設定する     | → 「4.1 グラフィック画面の座標系」 |
| ディスプレイ上に表示する位置を設定する | → 「4.1 グラフィック画面の座標系」 |
| グラフィック画面の色を変える      | → 「4.3 グラフィック画面の色」   |
| 描画する色を選択する          | → 4.3の「パレットの色の変更」    |
| モザイク模様で塗りつぶす        | → 4.3の「タイリング」        |
| 図形をコピーして他の位置に描く     | → 「4.4 図形データの転送」     |

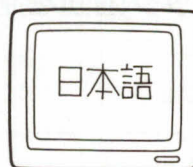
## 画面ハードコピーを行う



ディスプレイに表示した画面をそのままプリンタに印字したものが画面ハードコピーです。PC-PR201V系のカラープリンタを使用した場合、カラーコピーをとることができます。PC-PR601系のページプリンタを使用した場合、画面ハードコピーの倍率、印字方向を変えることができます。

| 目的                         | 参照箇所   |
|----------------------------|--|
| PC-PR201系プリンタを使用する         | → 3.3の「ハードコピー機能の設定方法」                        |
| PC-PR201系以外のプリンタを使用する      | → 3.3の「ハードコピー機能の設定方法」                        |
| カラーコピーをとる(PC-PR201V系プリンタ)  | → 3.3の「拡張画面ハードコピー機能」                         |
| 倍率、印字方向を変える(PC-PR601系プリンタ) | → 3.3の「拡張画面ハードコピー機能」                         |
| 画面ハードコピー機能の設定              | → 「10.14 メモリスイッチの設定」<br>「10.8 システムディスク属性の設定」 |

## 日本語文字を使用する

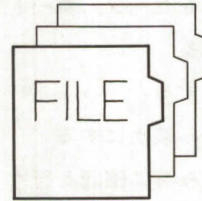


BASIC上で日本語文字を使用し、画面表示、プリンタ印字、ディスクファイルへの書き出し、ディスクファイルからの読み込みなどを行うことができます。

| 目的                 | 参照箇所   |
|--------------------|--|
| 日本語文字を使用するための準備をする | → 「5.2 日本語文字の使用準備と保守」<br>「10.13 利用者定義文字格納ファイルの作成・更新」<br>『日本語入力ガイド』 |

- 辞書ファイルへ単語の登録, 削除を行う → 5.2の「辞書ファイルの使用準備と保守」  
「10.15 辞書ファイルの保守」
- 辞書ファイルのドライブ番号を設定する → 5.2の「辞書ファイルの使用準備と保守」
- 日本語文字に変換する方式を設定する → 5.2の「日本語変換方式の選択」  
「10.8 システムディスク属性の設定」
- 文字を作成する → 5.2の「利用者定義文字」
- 日本語文字を入力する → 5.3の「日本語文字の入力」
- 日本語文字を画面表示する → 5.3の「日本語文字の出力」  
『日本語入力ガイド』
- 日本語文字をプリンタで印字する → 5.3の「日本語文字の出力」
- 日本語文字列を操作する → 「5.4 日本語文字列操作関数」

## ファイル进行操作する

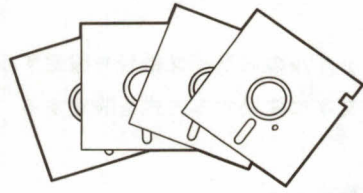


ディスクファイルだけでなく、キーボード、ディスプレイ、プリンタ、RS-232C 回線などのデバイスもファイルとして扱い、ファイル入出力命令を使って入出力を行うことができます。

| 目 的                  | 参照箇所                          |
|----------------------|-------------------------------|
| ファイル名(デバイス名)を指定する    | → 6.1の「デバイス(入出力装置)とファイル」      |
| ファイルをオープンする          | → 6.1の「ファイルのオープン/クローズ」        |
| ファイルをクローズする          | → 6.1の「ファイルのオープン/クローズ」        |
| デバイスファイル进行操作する       | → 「6.9 その他のデバイスファイルの使い方」      |
| RS-232C 回線ファイル进行操作する | → 「6.8 RS-232C 回線ファイルの扱い方」    |
| ディスクファイル进行操作する       | → 「6.2 ディスクファイル」              |
| ファイルを同時に3つ以上オープンする   | → 6.1の「ファイルバッファとファイルの同時オープン数」 |

## ディスクファイル进行操作する

(フロッピーディスクおよび固定ディスク)



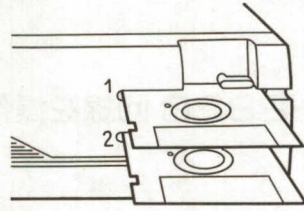
外部記憶装置としてフロッピーディスク，固定ディスクを使用することができます。ディスクファイルの操作は BASIC の命令，関数とユーティリティプログラムを使って行います。

### 目的

### 参照箇所

|                    |                           |
|--------------------|---------------------------|
| ファイルの種類やサイズをみる     | → 6.6の「ディスク中のファイル情報のチェック」 |
| プログラムをセーブ，ロードする    | → 6.6の「プログラムファイルのセーブとロード」 |
| 機械語プログラムをセーブ，ロードする | → 6.6の「機械語ファイルのセーブとロード」   |
| ファイルを削除する          | → 6.6の「ファイルの削除」           |
| ファイル名を変更する         | → 6.6の「ファイル名の変更」          |
| ファイルを書き込み禁止にする     | → 6.6の「ファイル属性の設定」         |
| ファイルの書き込み時に確認を行う   | → 6.6の「ファイル属性の設定」         |
| ファイルの属性を調べる        | → 6.6の「ファイル属性のチェック」       |
| ディスクを直接アクセスする      | → 6.6の「ディスクのセクタへの直接アクセス」  |
| ファイルを複写する          | → 「10.6 ファイル転送」           |
| シーケンシャルファイル进行操作する  | → 6.7の「シーケンシャルファイルの操作」    |
| ランダムファイル进行操作する     | → 6.7の「ランダムファイルの操作」       |

## フロッピーディスクの操作を行う



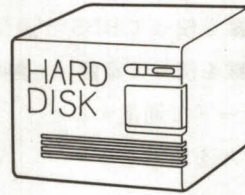
フロッピーディスクの使用準備、バックアップなどの操作法は固定ディスクの場合と異なります。

### 目的

### 参照箇所

- フロッピーディスクのフォーマットをする → 6.3の「フロッピーディスクのフォーマット」  
 システムディスクを作成する → 6.3の「システムディスクの作成」  
 フロッピーディスクのバックアップをとる → 6.3の「フロッピーディスクの管理」

## 固定ディスクの操作を行う

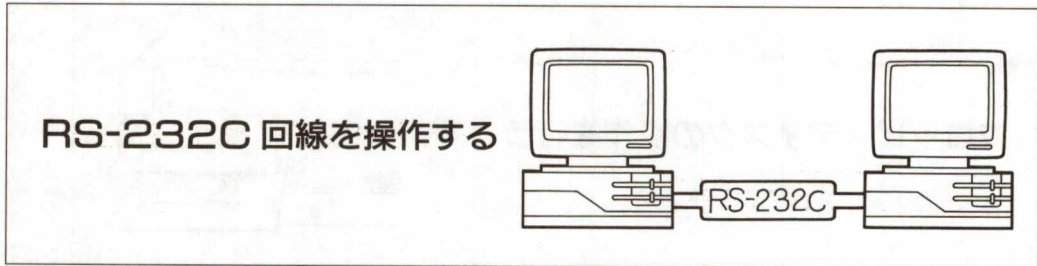


固定ディスクの使用準備、バックアップなどの操作法はフロッピーディスクの場合と異なります。

### 目的

### 参照箇所

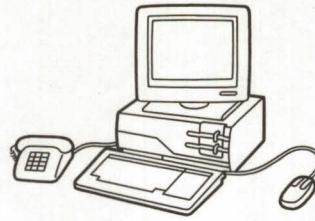
- 固定ディスクのフォーマットをする → 6.4の「固定ディスクのフォーマット」  
 『BASIC 入門』  
 固定ディスクの領域を異なった OS で分け  
 て使用する → 6.4の「固定ディスクに固有な機能」  
 固定ディスクに複数のシステムを登録する → 6.4の「固定ディスクに固有な機能」  
 固定ディスクを複数のユーザーで使用する → 6.4の「固定ディスクに固有な機能」  
 固定ディスクを複数のドライブに分割して  
 使用する → 6.4の「固定ディスクに固有な機能」  
 固定ディスクのバックアップを作る → 「10.12 固定ディスクファイル退避／復旧処  
 理」



BASIC では RS-232C インタフェースを介してコンピュータ本体と周辺機器の間でデータのやり取りを行い、機器をコントロールすることができます。また、コンピュータ本体どうしをつないでデータ通信を行ったりすることができます。

| 目 的                           | 参照箇所                       |
|-------------------------------|----------------------------|
| RS-232C 回線を使って周辺機器を操作する →     | 「6.8 RS-232C 回線ファイルの扱い方」   |
| RS-232C 回線を使ってコンピュータどうし →     | 「6.8 RS-232C 回線ファイルの扱い方」   |
| の通信を行う                        |                            |
| RS-232C 回線の 2 回線, 3 回線を使用する → | 6.8の「BASICによるRS-232C回線の制御」 |
| RS-232C 回線を使って BBS と通信する →    | 「9.1 電話制御機能」               |
| RS-232C 回線を使って電話機を制御する →      | 「9.1 電話制御機能」               |
| ターミナルモードで通信する →               | 「11.1 ターミナルモード」            |
| 通信プロトコルを設定する →                | 「11.1 ターミナルモード」            |
|                               | 「9.1 電話制御機能」               |
|                               | 6.8の「BASICによるRS-232C回線の制御」 |
|                               | 「10.14 メモリスイッチの設定」         |

その他



| 目 的                  | 参 照 箇 所              |
|----------------------|----------------------|
| マウスを操作する             | → 「7.1 マウスの使い方」      |
| ライトペンを操作する           | → 「7.2 ライトペンの使い方」    |
| 機械語プログラムを使用する        | → 「第8章 機械語プログラム」     |
| 電話機を制御する             | → 「9.1 電話制御機能」       |
| サウンド機能を使用する          | → 「9.2 サウンド制御機能」     |
| GP-IB(IEEE-488)を制御する | → 「第9章 拡張機能」         |
| メモリスイッチを変更する         | → 「10.14 メモリスイッチの設定」 |
| メモリサイズを設定する          | → 「10.14 メモリスイッチの設定」 |
| 数値演算プロセッサを使用する       | → 「10.14 メモリスイッチの設定」 |
| ターミナルモードを使用する        | → 「11.1 ターミナルモード」    |
| モニタモードを使用する          | → 「11.2 モニタモード」      |



# 目次

|                  |     |
|------------------|-----|
| はじめに .....       | (3) |
| このマニュアルの構成 ..... | (4) |
| 目的別索引 .....      | (7) |

---

## 第1章 N<sub>86</sub>-BASIC(86)の起動 1

---

|                           |    |
|---------------------------|----|
| 1.1 ROMモードBASICの起動.....   | 2  |
| 1.2 DISKモードBASICの起動 ..... | 3  |
| システムディスクからの起動 .....       | 4  |
| 固定ディスクからの起動 .....         | 6  |
| オートスタート .....             | 7  |
| DISKモードBASICの機能選択 .....   | 9  |
| 1.3 リセットとウォームリスタート .....  | 11 |
| リセット .....                | 11 |
| ウォームリスタート .....           | 11 |

---

## 第2章 キーボードとスクリーンエディタ 13

---

|                               |    |
|-------------------------------|----|
| 2.1 キーの配置 .....               | 13 |
| 2.2 文字キー .....                | 14 |
| 2.3 特殊キー .....                | 15 |
| 文字とグラフィックシンボルのタイプに関するキー ..... | 15 |
| 文字編集に関するキー .....              | 17 |
| その他のキー .....                  | 18 |
| テンキーについて .....                | 20 |
| 2.4 スクリーンエディタ .....           | 21 |
| BASICプログラムの入力 .....           | 21 |
| プログラムの修正 .....                | 21 |
| 便利な編集コマンドとキー .....            | 23 |
| CTRL キーを使った便利な編集機能.....       | 24 |
| 日本語文字使用時の注意 .....             | 25 |

---

**第3章 画面**

---

27

|     |                   |    |
|-----|-------------------|----|
| 3.1 | テキスト画面            | 27 |
|     | 桁数と行数             | 28 |
|     | スクロール画面の範囲        | 29 |
|     | ファンクションキーの表示      | 30 |
|     | カラーモード／白黒モード      | 30 |
|     | テキスト画面の文字の色と表示モード | 31 |
| 3.2 | グラフィック画面          | 32 |
|     | 画面モード             | 32 |
|     | ページ               | 33 |
|     | 画面スイッチ            | 38 |
| 3.3 | 画面ハードコピー          | 38 |
|     | 画面ハードコピー機能の種類と機能  | 39 |
|     | 通常の画面ハードコピー機能     | 39 |
|     | 拡張画面ハードコピー機能      | 40 |
|     | ハードコピー機能の設定方法     | 42 |

---

**第4章 グラフィックス**

---

43

|     |                      |    |
|-----|----------------------|----|
| 4.1 | グラフィック画面の座標系         | 43 |
|     | オリジナルスクリーン座標系        | 44 |
|     | ビューポートとスクリーン座標系      | 44 |
|     | ワールド座標系              | 46 |
|     | ウィンドウ                | 47 |
|     | 3つの座標系の関係            | 49 |
| 4.2 | WINDOW命令とVIEW命令の例    | 50 |
|     | ワールド座標に図形を描く         | 50 |
|     | WINDOW命令を使ったウィンドウの設定 | 51 |
|     | VIEW命令を使ったビューポートの設定  | 52 |
|     | ワールド座標値の制限           | 53 |

|     |                         |    |
|-----|-------------------------|----|
| 4.3 | グラフィック画面の色              | 54 |
|     | グラフィック画面の色指定            | 54 |
|     | パレットモード                 | 55 |
|     | 基本グラフィックモードと拡張グラフィックモード | 56 |
|     | パレットの色の変更               | 57 |
|     | 色の仕組み(カラーコードの指定方法)      | 59 |
|     | タイリング                   | 60 |
| 4.4 | 図形データの転送                | 64 |

---

## 第5章 日本語処理

---

67

|     |                      |    |
|-----|----------------------|----|
| 5.1 | 日本語文字とは              | 67 |
|     | 2バイトで扱われる日本語文字       | 67 |
|     | 日本語文字の種類             | 68 |
|     | 日本語文字の使い方            | 69 |
|     | 日本語文字列の内部形式          | 70 |
| 5.2 | 日本語文字の使用準備と保守        | 72 |
|     | 漢字ROMの装備             | 72 |
|     | ディスプレイ               | 72 |
|     | 日本語変換方式の選択           | 72 |
|     | 辞書ファイルの使用準備と保守       | 73 |
|     | 利用者定義文字              | 74 |
| 5.3 | 日本語文字の入力と出力          | 76 |
|     | 日本語文字の入力             | 76 |
|     | 日本語文字の出力             | 77 |
| 5.4 | 日本語文字列操作関数           | 79 |
|     | 日本語文字あるいは1バイト文字の取り出し | 79 |
|     | 日本語文字と1バイト文字の相互変換    | 80 |
|     | 日本語文字列と漢字コードの相互変換    | 80 |
|     | その他の関数               | 81 |
| 5.5 | 日本語処理機能を使ったプログラム例    | 82 |
|     | 住所録プログラムの使い方と解説      | 82 |

---

**第6章 ファイル**

---

89

|     |                            |     |
|-----|----------------------------|-----|
| 6.1 | ファイルとは                     | 89  |
|     | ファイルの種類                    | 89  |
|     | デバイス(入出力装置)とファイル           | 90  |
|     | ファイルディスクリプタ                | 92  |
|     | ファイルバッファ                   | 92  |
|     | ファイルのオープン/クローズ             | 93  |
|     | ファイルバッファとファイルの同時オープン数      | 95  |
| 6.2 | ディスクファイル                   | 96  |
|     | 外部記憶装置としてのディスク装置           | 96  |
|     | フロッピーディスクと固定ディスク           | 96  |
| 6.3 | フロッピーディスクの使用準備と管理          | 97  |
|     | フロッピーディスクのフォーマット           | 97  |
|     | システムディスクの作成                | 98  |
|     | フロッピーディスクの管理               | 99  |
| 6.4 | 固定ディスクの使用                  | 100 |
|     | 固定ディスクに固有な機能               | 100 |
|     | 固定ディスクのフォーマット              | 104 |
|     | 固定ディスクからのBASICの起動          | 106 |
|     | 固定ディスクの管理                  | 108 |
| 6.5 | ファイルディスクリプタとドライブ番号         | 109 |
|     | ディスクファイルにおけるファイルディスクリプタの形式 | 109 |
|     | ドライブ番号の割り当てられ方             | 110 |
| 6.6 | BASIC命令によるディスクファイルの操作      | 112 |
|     | ディスク中のファイル情報のチェック          | 112 |
|     | プログラムファイルのセーブとロード          | 113 |
|     | プログラムのマージ                  | 115 |
|     | プログラムの連結                   | 115 |
|     | 機械語ファイルのセーブとロード            | 117 |
|     | ファイルの削除                    | 118 |
|     | ファイル名の変更                   | 118 |
|     | ファイル属性の設定                  | 118 |
|     | ファイル属性のチェック                | 119 |
|     | ディスクのセクタへの直接アクセス           | 120 |

|     |                            |     |
|-----|----------------------------|-----|
| 6.7 | データファイルの扱い方                | 121 |
|     | シーケンシャルファイルとランダムファイル       | 121 |
|     | シーケンシャルファイルの操作             | 122 |
|     | シーケンシャルファイルの例              | 125 |
|     | ランダムファイルの操作                | 126 |
|     | ランダムファイルの例                 | 131 |
| 6.8 | RS-232C回線ファイルの扱い方          | 132 |
|     | RS-232C回線について              | 132 |
|     | BASICによるRS-232C回線の制御       | 135 |
|     | RS-232C回線ファイルを使った基本プログラム   | 144 |
|     | RS-232C回線ファイルの割り込み処理       | 146 |
|     | RS-232C回線ファイルを使ったサンプルプログラム | 147 |
|     | RS-232C回線ファイルを使用する場合の注意    | 153 |
| 6.9 | その他のデバイスファイルの使い方           | 154 |
|     | デバイスファイルの使用例               | 154 |

---

## 第7章 入力装置

---

157

|     |                     |     |
|-----|---------------------|-----|
| 7.1 | マウスの使い方             | 157 |
|     | マウス用ソフトウェアドライバの使用方法 | 157 |
|     | ソフトウェアドライバのファンクション  | 159 |
|     | マウス環境の初期設定          | 168 |
|     | ソフトウェアドライバ使用上の注意    | 168 |
|     | マウスを使ったプログラム例       | 169 |
| 7.2 | ライトペンの使い方           | 172 |
|     | ライトペン制御命令           | 172 |
|     | ライトペン制御命令を使ったプログラム例 | 173 |

---

**第8章 機械語プログラム**


---

175

|     |                       |     |
|-----|-----------------------|-----|
| 8.1 | メモリレイアウトとセグメント        | 176 |
|     | セグメントベース              | 176 |
| 8.2 | 機械語プログラムの準備           | 177 |
|     | 機械語プログラムが使用するメモリ領域の確保 | 177 |
|     | 使用するセグメントの宣言          | 178 |
|     | 機械語プログラムをメモリに置く方法     | 178 |
| 8.3 | 機械語プログラムの実行           | 180 |
|     | USR関数を使う方法            | 180 |
|     | CALL命令を使う方法           | 185 |
|     | BLOAD命令を使う方法          | 187 |

---

**第9章 拡張機能**


---

189

|     |                                      |     |
|-----|--------------------------------------|-----|
| 9.1 | 電話制御機能                               | 190 |
|     | 電話制御のための拡張命令                         | 191 |
|     | 電話制御命令の使用に際して                        | 191 |
|     | 4つのモード                               | 193 |
|     | 電話制御命令を使ったプログラム例                     | 196 |
|     | 電話制御命令使用時の注意                         | 199 |
|     | 電話管理ユーティリティ "tele.n88" の使い方          | 199 |
|     | 電話管理ユーティリティの使用に際して                   | 200 |
|     | 電話管理ユーティリティの操作                       | 201 |
| 9.2 | サウンド制御機能                             | 209 |
|     | サウンド制御機能の特徴                          | 209 |
|     | サウンド制御命令の使用準備                        | 210 |
|     | サウンド制御命令の概要                          | 210 |
|     | サウンドバッファについて                         | 211 |
|     | FM音源について                             | 213 |
|     | SSG音源について                            | 221 |
|     | FM/SSG音源内部レジスタマップ                    | 224 |
|     | ミュージックジェネレータボード(PC-9801-14)用プログラムの移植 | 226 |
|     | サウンド制御命令を使ったプログラム例                   | 228 |
|     | 外部オーディオ機器の使用                         | 230 |

---

**第10章 ユーティリティプログラム**

---

231

- 10.1 ユーティリティプログラムを使う前に .....231
- 10.2 メニュープログラム .....233
- 10.3 フロッピィディスクのフォーマット .....234
  - 物理フォーマットと論理フォーマット .....234
  - “format.nip” の使い方 .....235
- 10.4 フロッピィディスクのバックアップ .....238
  - “backup.n88” の使い方 .....238
- 10.5 IDセクタの書き換え .....240
  - “setinf.n88” の使い方 .....240
- 10.6 ファイル転送 .....242
  - “xfiles.n88” の使い方 .....242
- 10.7 システム(DISK CODEおよびIPL)のコピー .....244
  - “sysgen.nip” の使い方 .....244
- 10.8 システムディスク属性の設定 .....246
  - “setup.n88” を使った日本語入力の属性設定 .....246
  - “setup.n88” を使った拡張画面ハードコピーの機能設定 .....248
  - “setup.n88” を使った演算モードの設定 .....250
- 10.9 固定ディスクボリューム管理 .....252
  - “format.hd” の使い方 .....253
- 10.10 固定ディスク障害ボリューム・ファイル復旧 .....262
  - “recov.hd” の使い方 .....262
- 10.11 固定ディスクファイルディレクトリ表示 .....263
  - “dir.hd” の使い方 .....263
- 10.12 固定ディスクファイル退避/復旧処理 .....265
  - “backup.hd” の使い方 .....265
- 10.13 利用者定義文字格納ファイルの作成・更新 .....267
  - “mkfont.n88” の使い方 .....267
- 10.14 メモリスイッチの設定 .....273
  - “switch.n88” の使い方 .....273
- 10.15 辞書ファイルの保守 .....278
  - “dicmen.n88” の使い方 .....278
- 10.16 ファイル変換 .....286
  - “DDconv.n88” の使い方 .....286
- 10.17 ユーティリティプログラムの目的別分類 .....288

(23)

---

**第11章 ターミナルモードとモニタモード**


---

289

|      |                         |     |
|------|-------------------------|-----|
| 11.1 | ターミナルモード                | 289 |
|      | ターミナルモードとBASICモードとの切り換え | 289 |
|      | ターミナルの仕様                | 290 |
|      | ターミナルの仕様の設定方法           | 296 |
|      | ターミナルの使い方               | 298 |
| 11.2 | モニタモード                  | 303 |
|      | モニタモードの使用準備             | 303 |
|      | モニタモードとBASICモードとの切り換え   | 303 |
|      | 機械語プログラムセグメントのメモリ配置     | 304 |
|      | モニタモードのコマンド一覧           | 305 |
|      | コマンド使用上の規則              | 306 |
|      | コマンド解説                  | 308 |
| 付録A  | メモリマップ                  | 321 |
| 付録B  | データの内部形式                | 327 |
| 付録C  | ディスクの内部構造               | 333 |
| 付録D  | キーセンス                   | 347 |
| 付録E  | メモリスイッチ                 | 349 |
| 付録F  | 数値演算プロセッサの使用            | 353 |
| 付録G  | 日本語コード表                 | 355 |
| 付録H  | キャラクタコード表               | 359 |
| 索引   |                         | 361 |

# 第 1 章

## N<sub>88</sub>-BASIC (86)の起動

PC-9800 シリーズの各機種の本体内には N<sub>88</sub>-BASIC (86) インタプリタの ROM (96KB) が内蔵されています。この ROM には N<sub>88</sub>-BASIC (86) の基本機能が組み込まれており、電源を入れただけで BASIC を使うことができます。この BASIC を ROM モード BASIC と呼びます。

また、N<sub>88</sub>-日本語 BASIC (86) のシステムディスクをフロッピーディスク装置にセットして BASIC を起動させると、BASIC の機能が拡張し、フロッピーディスク装置の使用、日本語の使用など様々な機能が使えるようになります。この BASIC を DISK モード BASIC と呼びます。

### ● DISK モード BASIC によって拡張される機能

- ・ ディスク装置の制御機能
- ・ 倍精度数値データのための関数 (SIN, COS, TAN などの関数)
- ・ AI 逐次変換入力方式、AI 連文節変換入力方式あるいは単文節変換入力方式による日本語入力機能\*
- ・ 日本語文字列操作のための命令や関数
- ・ DRAW 命令 (複雑な図形や描画を行うのに便利な機能)
- ・ 拡張グラフィック機能 (中間色の表示が可能)\*
- ・ 電話制御機能\*
- ・ モニタ機能\*
- ・ 拡張画面ハードコピー機能\*
- ・ マウス用ソフトウェアドライバ (マウスを制御するためのプログラム) の使用


\*印の付いた機能は使用するかどうかを選択することができます。

▶ 参照 機能の選択 → 1.2 の「DISK モード BASIC の機能選択」

## 1.1 ROMモード BASIC の起動

- ①本体にフロッピーディスク装置が内蔵されている機種の場合には、フロッピーディスクがセットされていないことを確認します。
- ②本体に外付け型のフロッピーディスク装置や固定ディスク装置が接続されている場合、装置の電源を OFF にしておきます。
- ③本体の電源を ON にします。すでに電源が ON になっている場合にはリセットスイッチを押します。
- ④しばらくすると ROM モード BASIC が起動し次のように表示されます。

How many files(0-15)? ■

これは一度にオープンできるファイルの数を聞いているものです。ここでは  キーを押しておきます。

- ⑤ ROM モード BASIC が起動し、次のように表示されます。

How many files(0-15)?

NEC N-88 BASIC(86) version X.X

Copyright (C) 1983 by NEC Corporation/Microsoft Corp.

XXXXXXXX Bytes free

Ok

■

- ▶参照 ファイルの数→6.1の「ファイルバッファとファイルの同時オープン数」  
リセットスイッチ→『ガイドブック』

### ● “Ok” が表示されない場合の原因

“Ok” が出ない場合には、次のような原因が考えられます。

#### ターミナルモードが起動しているとき

この場合 “Terminal mode” と表示されます。

ディップスイッチ SW2 の 2 番を OFF にしてからリセットスイッチを押して再起動させてください。

#### 拡張インタフェースボードのセット状態とシステムの設定が不一致のとき

GP-IB ボード、RS-232C(第2回線/第3回線)などの拡張インタフェースボードが実装されていないにもかかわらず、ユーティリティ “switch.n88” で拡張インタフェースボードの使用を宣言しているような場合です。

この場合“×××××× Bytes free”までは画面に表示されますが、“Ok”が表示されません。次の手順で再設定してください。

- ①ディップスイッチ SW2 の 5 番を OFF にしてからリセットスイッチを押してシステムを再起動します。メモリスイッチの値が初期化されます。
- ②ユーティリティ “switch.n88” を使用して、必要なメモリスイッチを設定し直してください。
- ③メモリスイッチを変更したら、必ずディップスイッチ SW2 の 5 番を ON にしてからシステムを再起動します。

▶参照 ターミナルモード→「11.1 ターミナルモード」  
ディップスイッチ→『ガイドブック』  
switch.n88 →「10.14 メモリスイッチの設定」

---

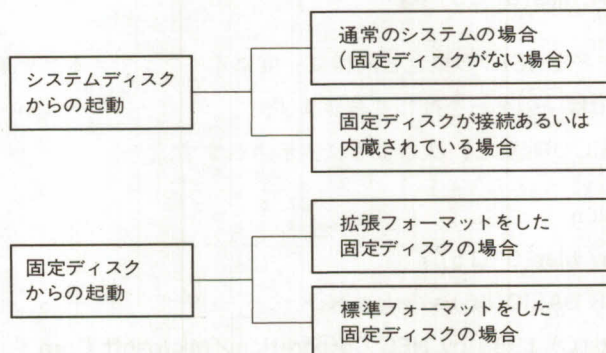
注意：“×××××× Bytes free” というのは使用できるメモリの量を示しています。

---

## 1.2 DISKモードBASICの起動

DISKモードBASICを起動させるにはN<sub>88</sub>-日本語BASIC(86)のシステムディスクを使います。また、固定ディスク装置が接続(あるいは内蔵)されている場合には固定ディスク装置からDISKモードBASICを起動させることもできます。

### 種々の起動方法



## ■システムディスクからの起動

### ●通常の装置構成の場合

固定ディスク装置が接続(あるいは内蔵)されていない場合です。

- ①本体の電源を ON にします。
- ②システムディスクをセットします。

これまでの操作の途中で ROM モード BASIC が起動してしまうことがありますが、かまわずに先に進んでください。

システムディスクは、本体に接続、あるいは内蔵されているフロッピーディスク装置のうちいずれのドライブにセットしても起動させることができます。ただし、ドライブ番号 1 以外のドライブにセットする場合はそれより優先順位の高いドライブにはフロッピーディスクをセットしないようにしてください。

▶参照 ディスクの優先順位→「6.5 ファイルディスクリプタとドライブ番号」

- ③リセットスイッチを押します。②までの操作で ROM モード BASIC が起動していなければ、リセットスイッチを押す必要はありません。
- ④しばらくするとシステムディスクをセットしたドライブのアクセス表示用の LED が点灯します。システムディスクに記録された DISK CODE(DISK モード BASIC 用のシステムプログラム部分)の読み込みが行われます。
- ⑤さらにしばらくすると LED が消え、ディスプレイに次のメッセージが現れます。

```
Disk version
How many files(0-15)? ■
```

これは、ROM モード BASIC の場合と同様に一度にオープンできるファイルの数を聞いているものです。ここでは  キーを押しておきます。

- ⑥ DISK モード BASIC が起動し、次のように表示されます。

```
Disk version
How many files(0-15)?
NEC N-88 BASIC(86) version X.X
Copyright (C) 1983 by NEC Corporation/Microsoft Corp.
XXXXXXXX Bytes free
Ok
■
```

▶参照 ファイルの数→6.1の「ファイルバッファとファイルの同時オープン数」

● 固定ディスク装置が接続あるいは内蔵されている場合

通常の装置構成に、固定ディスクが接続あるいは内蔵されている場合です。

固定ディスク装置は複数のユーザーで使うことができます。固定ディスクを使うユーザーは自分のユーザー識別名を持ち、BASICを起動する時にユーザー識別名によってユーザーが誰であるかを宣言します。BASICが起動したあとは通常どおりBASICを使うことができます。ただし、基本的にユーザーは自分の識別名で起動したときに作成したファイル以外は扱うことはできません。したがって、自分のファイルを他のユーザーから保護することができます。なお、ユーザー識別名が異なれば同じファイル名であっても別のファイルを作成することができます。実際の起動の仕方は次のとおりです。

① “How many files (0-15) ?” までの操作は固定ディスク装置が接続されていない場合と同じです。

②ここで  キーを押すと次のように表示されます。

User identifier ? ■

これは利用者が誰であるかの宣言(ユーザー識別名の入力)を求めるものです。ここでは  キーだけを入力してください。

③次のようにDISKモードBASICが起動します。

```
Disk version
How many files(0-15) ?
User identifier ?
NEC N-88 BASIC(86) version X.X
Copyright (C) 1983 by NEC Corporation/Microsoft Corp.
XXXXXXXX Bytes free
Ok
■
```

---

注意：ユーザー識別名は固定ディスク上のファイルを対象としているものです。フロッピーディスク上のファイルには関係ありません。

---

▶ 参照 ユーザー識別名→6.4の「固定ディスクに固有な機能」

## ■固定ディスクからの起動

### ●固定ディスクから DISK モード BASIC を起動するための準備

新しい固定ディスクから DISK モード BASIC を起動するには次のような準備が必要です。

#### ①固定ディスクをフォーマットする。

ユーティリティ “format.hd” を使います。

ディスクの種類によって、標準フォーマットと拡張フォーマットのどちらかを選択することもできます。

#### ▶参照 format.hd →「10.9 固定ディスクボリューム管理」

標準フォーマットと拡張フォーマット → 6.4 の「固定ディスクのフォーマット」

#### ② DISK CODE, IPL および日本語辞書ファイルを固定ディスクへ転送する。

ユーティリティ “sysgen.nip”, “xfiles.n88” を使って行います。

#### ▶参照 sysgen.nip →「10.7 システム (DISK CODE および IPL) のコピー」

xfiles.n88 →「10.6 ファイル転送」

### ●拡張フォーマットをした固定ディスクから起動させる場合

#### ①本体の電源を ON にします。

②拡張フォーマットをした固定ディスクに複数のシステム(N<sub>88</sub>-日本語 BASIC(86)あるいは MS-DOS など)が登録されている場合は、起動するシステムを選択するメニューが表示されますので、BASIC を選択します。

③ “How many files(0-15) ?” が表示されますので、システムディスクから起動させた場合と同様にここでは  キーだけを押してください。

④次に、“User identifier?” の問いが行われます。固定ディスクが接続あるいは内蔵されている装置構成でシステムディスクからの起動を行った場合と同様にここでは  キーだけを押してください。

### ●標準フォーマットをした固定ディスクから起動させる場合

標準フォーマットをしたディスクでは、N<sub>88</sub>-日本語 BASIC(86)以外を起動させることはできませんので、「拡張フォーマットをした固定ディスクから起動させる場合」のようにメニューの表示はありません。その他の手順は同じです。

---

**注意：**フロッピーディスク装置にシステム(他の OS の場合も含む)の起動が可能なディスクが入っていると、そのシステムが起動してしまうことがありますので、フロッピーディスクは、ディスク装置から抜いておいてください。

---

### ●固定ディスクだけを起動装置にする方法

出荷時の状態ではシステム(N<sub>88</sub>-日本語 BASIC (86)あるいはMS-DOSなど)の起動が可能なフロッピーディスクがディスク装置にセットされているとそのシステムが起動してしまいます。これを避けて、固定ディスクだけを起動装置にするように設定することができます。

それには、ユーティリティ "switch.n88" を使用します。

ユーティリティ "switch.n88" の "立ち上げ装置" の選択画面で、"BOOT on hard disk 1"、"BOOT on hard disk 2" または、"BOOT on SCSI hard disk" を選択してください。

なお、SCSI固定ディスクを起動装置に設定した場合、SCSI固定ディスクのうち1台目のみが対象となります。2台目以降を起動装置に設定することはできません。

この設定を解除するには同じ選択画面で "フロッピーディスク→hard disk" を選択してください。

▶参照 switch.n88 →「10.14 メモリスイッチの設定」

### ■オートスタート

これまで説明した通常の BASIC の起動方法では、BASIC の起動時に "How many files?" の情報やユーザー識別名、実行する命令などを入力しなければプログラムを実行することができません。

そこで、これらの情報をあらかじめシステムディスクや固定ディスクに記憶しておくことにより、電源を入れると同時に BASIC を起動させてプログラムを実行することが可能です。これをオートスタートといいます。

#### ●通常の(固定ディスクを使用していない)場合

① "How many files?" の情報をシステムディスクに書き込みます。

ユーティリティ "setinf.n88" を使います。

②起動直後に実行する命令をシステムディスクに書き込みます。

ユーティリティ "setinf.n88" を使います。

#### ●固定ディスクを使用している場合

固定ディスクを使用していない場合の①、②と同様に、ユーティリティ "setinf.n88" で "How many files?" の情報と起動直後に実行する命令をシステムディスクあるいは、起動可能な固定ディスクの ID に書き込んでおきます。

さらに固定ディスクを使用している場合には起動時にユーザー識別名の入力が必要になりますので、これを不要にする必要があります。それには次の方法があります。

#### ユーザー識別名を使用しない方法

ユーティリティ "switch.n88" を使用して "固定ディスクユーザー識別名" の "使わない" を選択してください。この設定を行うとユーザー識別名として "999" が入力されたものとして扱われます。

### ユーザー識別名をIDに設定する方法

ユーザー識別名を使いたい場合は次のように行います。

まず、ユーティリティ "switch.n88" を使用して "固定ディスクユーザー識別名" の "使わない" を選択します。

次にユーティリティ "setinf.n88" でIDに設定する BASIC の命令の先頭に次のように命令を書き込みます。

```
DEF SEG=&H60:POKE &H510, n1:POKE &H511, n2:POKE &H512, n3  
n1:ユーザー識別名の1文字目のキャラクタコードの整数表記  
n2:ユーザー識別名の2文字目のキャラクタコードの整数表記  
n3:ユーザー識別名の3文字目のキャラクタコードの整数表記
```

たとえばユーザー識別名が "A01" で "SAMPLE.BAS" というプログラムをオートスタートで実行したいときには、キャラクタコードを得るASC関数を使って、



```
DEF SEG=&H60:POKE &H510,ASC('A'):POKE &H511,ASC('0'):POKE &  
H512,ASC('1'):RUN"SAMPLE.BAS"
```

とIDに書き込みます。

▶参照 setinf.n88 →「10.5 IDセクタの書き換え」

### ●拡張フォーマットをした固定ディスクから起動する場合

拡張フォーマットをした固定ディスクに、複数のシステム(N<sub>88</sub>-日本語 BASIC(86)とMS-DOSなどが登録されている場合、電源をONにするとどのシステムを起動させるかをメニューで聞いてきます。メニューの表示をやめ、BASICをオートスタートさせるためには次のようにします。

- ①電源を入れシステムを選択するメニューが表示されたら、BASICの領域と"自動起動に設定する"を選択して  キーを押します。
- ②次に、"起動する"を選択して  キーを押すと BASIC が起動します。
- ③ユーティリティ "switch.n88" を使用して、"立ち上げ装置"の選択画面で "BOOT on hard disk 1", "BOOT on hard disk 2" あるいは、"BOOT on SCSI hard disk" を選択してください。
- ④リセットするとメニュー画面が現れず、自動的に DISK モード BASIC が起動します。

その後、リセットあるいは電源ONのたびに、DISKモードBASICがオートスタートします。

オートスタートを解除するには、ユーティリティ "switch.n88" を使って "立ち上げ装置" の選択画面で、"フロッピーディスク → hard disk" を選択してください。起動前にシステム選択のメニューが現れるようになります。その後、フロッピーディスク装置にディスクを入れないで再起動すると、メニュー画面が表示されますので、メニュー画面で自動起動を解除してください。

```

NEC PC-9800 シリーズ
固定ディスク起動メニュープログラム バージョン x.xx
Copyright (C) NEC Corporation xxxx,xxxx

```

固定ディスクドライブ#1

処 理： 起動する 自動起動に設定する 次のドライブ

領域①： MS-BASIC  
 ②： MS-DOS x.xx  
 ③：  
 ④：

説明：カーソル移動キー=処理/領域の選択 リターンキー=実行

▶参照 switch.n88 →「10.14 メモリスイッチの設定」

## ■ DISK モード BASIC の機能選択

DISK モード BASIC では、次に示す機能を使用するかどうかを選択することができます。

- ・グラフィック機能の選択
- ・日本語変換方式の選択
- ・画面ハードコピー機能の選択
- ・モニターモードの使用の有無
- ・電話制御機能の使用の有無

これらの選択はディップスイッチの設定、あるいはユーティリティプログラムを使用しています。

### ● グラフィック機能の選択

基本グラフィックモードと 4096 色表示のできる拡張グラフィックモードのいずれかを選択することができます。

本体のディップスイッチの設定で選択を行います。

|             |                        |
|-------------|------------------------|
| 基本グラフィックモード | ディップスイッチ SW1 の 8 番 OFF |
| 拡張グラフィックモード | ディップスイッチ SW1 の 8 番 ON  |

▶参照 基本グラフィックモードと拡張グラフィックモード → 4.3 の「基本グラフィックモードと拡張グラフィックモード」

ディップスイッチ → 『ガイドブック』

### ●日本語変換方式の選択

AI 逐次変換方式, AI 連文節変換方式, 逐次変換方式, 連文節変換方式, 単文節変換方式, JIS16 進コード変換方式のいずれかを選択することができます。

ユーティリティ “setup.n88” を使用します。

▶ 参照 setup.n88 → 「10.8 システムディスク属性の設定」

日本語変換方式 → 5.2 の「日本語変換方式の選択」

『日本語入力ガイド』

### ●画面ハードコピー機能の選択

従来 PC-9800 シリーズ各機種に備わっていた画面ハードコピー機能と同等のものと、カラー／縮小／拡大のできる拡張画面ハードコピー機能のいずれかを選択することができます。

ユーティリティ “switch.n88”, “setup.n88” を使用します。

▶ 参照 setup.n88 → 「10.8 システムディスク属性の設定」

switch.n88 → 「10.14 メモリスイッチの設定」

画面ハードコピー → 「3.3 画面ハードコピー」

### ●モニタモードの使用の有無

機械語プログラムの作成／修正のための環境（モニタモード）の使用の有無を選択することができます。

ユーティリティ “switch.n88” を使用します。

▶ 参照 機械語プログラム → 「第8章 機械語プログラム」

switch.n88 → 「10.14 メモリスイッチの設定」

モニタモード → 「11.2 モニタモード」

### ●電話制御機能の使用の有無

モデム-NCU 内蔵電話機, モデムボード+ハンドセットあるいはインテリジェントモデムなどが接続されている場合にはその制御機能の有無を選択することができます。

ユーティリティ “switch.n88” を使用します。

▶ 参照 switch.n88 → 「10.14 メモリスイッチの設定」

電話制御機能 → 「9.1 電話制御機能」

## 1.3 リセットとウォームリスタート

### ■リセット

本体についているリセットスイッチを押すと、電源を OFF にしたときと同じようにハードウェアの機能が初期化(リセット)されます。リセットスイッチは次のような場合に使います。

- 何かの理由でコンピュータがハングアップして、キーボードからの入力に反応がなくなった場合。
- システムを新たに起動させたい場合。

たとえば BASIC のモードを変えたいときとか、BASIC から MS-DOS にシステムを変えたいときなどです。リセットスイッチを押すと、利用者が使っていたメモリの内容などはすべて初期化されてしまいます。入力されていたメモリ上のプログラムやデータはすべて消失してしまいますから十分注意してください。

### ■ウォームリスタート

何かの理由でコンピュータがハングアップしてキーボードからの入力に反応がなくなったときにリセットを行うと、入力したメモリ上のプログラムは失われてしまいます。

プログラムを失っては困る場合には、リセットの代わりにウォームリスタートを使います。

ウォームリスタートは、**STOP** キーを押しながらリセットスイッチを押します。このとき **STOP** キーはリセットスイッチを離したあとで離すようにしなければなりません。すると、画面がクリアされ左上隅のホームポジションに "Ok" と表示されます。ウォームリスタートは入力されたメモリ上のプログラムやデータが保存される以外はほぼリセットと同じ働きをします。ただし、テキスト画面のモードはハングアップ前の状態に関係なく 80 桁×25 行モードになります。

ウォームリスタートした場合、ハングアップの原因となったプログラムに何らかの問題があると考えられますので、ひとまずそのプログラムをディスクにセーブしてください。さらに BASIC の作業領域などが破壊されていることがありますので、そのあと必ずリセットしてください。

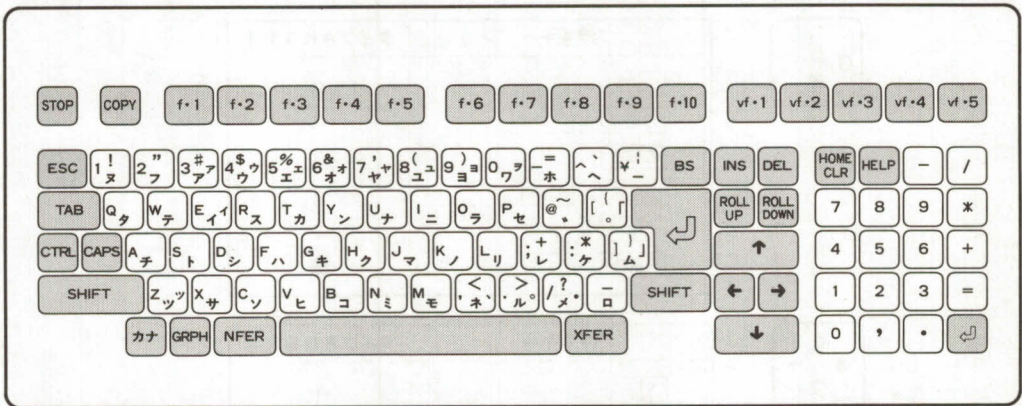
- ▶ 参照 セーブ→6.6の「プログラムファイルのセーブとロード」  
テキスト画面→「3.1 テキスト画面」



# 第2章

## キーボードとスクリーンエディタ

### 2.1 キーの配置



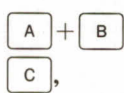
キーボードには英文字、カタカナ、数字、記号などの文字をタイプするための文字キーと、その他特殊な役割を持つ特殊キーがあります。図の□で示したキーが特殊キーです。

注意：キーの種類および配置は、機種によって異なります。

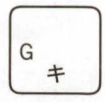
## 2.2 文字キー

1つの文字キーで数種類の文字をタイプすることができます。タイプする文字の種類は特殊キーとの組合せで選択することができます。

3つのキーを例にして文字をタイプする方法を説明します。“+”と“,”は次のことを意味します。



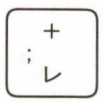
A キーを押しながら  B キーを押す。  
 C キーを押してロックさせておく。



| 押すキー                                   | タイプされる文字 |
|--|----------|
| <input type="checkbox"/>               | g        |
| SHIFT + <input type="checkbox"/>       | G        |
| CAPS, <input type="checkbox"/>         | G        |
| CAPS, SHIFT + <input type="checkbox"/> | g        |
| カナ, <input type="checkbox"/>           | キ        |
| GRPH + <input type="checkbox"/>        | ●        |



| 押すキー                                 | タイプされる文字 |
|--------------------------------------|----------|
| <input type="checkbox"/>             | 4        |
| SHIFT + <input type="checkbox"/>     | \$       |
| カナ, <input type="checkbox"/>         | ウ        |
| カナ, SHIFT + <input type="checkbox"/> | ウ        |



| 押すキー                             | タイプされる文字 |
|----------------------------------|----------|
| <input type="checkbox"/>         | ;        |
| SHIFT + <input type="checkbox"/> | +        |
| カナ, <input type="checkbox"/>     | レ        |
| GRPH + <input type="checkbox"/>  | !        |

## 2.3 特殊キー

### ■文字とグラフィックシンボルのタイプに関するキー

#### SHIFT (シフトキー)

このキーを押しながら文字キーを押すと、英大文字(ただし、CAPS キーがロックされているときは英小文字)、あるいは各文字キーの上部に刻印されている文字をタイプすることができます。このキーはキーボードの左右両側にありますがどちらを使ってもかまいません。

#### CAPS (キャピタルロックキー)

英大文字をタイプするのに使います。一度押すとこのキーはロックされた状態になり、英大文字をタイプできるようになります。もう一度押すとロックは解除されます。ロックが解除された状態では英小文字をタイプできます。ただし、SHIFT キーを使用した場合には、ここで説明した英大文字と英小文字の関係は逆になります。

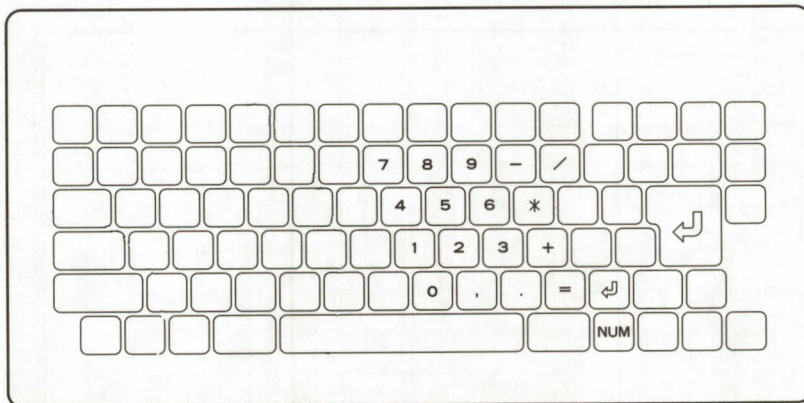
#### カナ (カナキー)

カナ文字をタイプするのに使います。CAPS と同様に一度押すとこのキーはロックされた状態になります。ロックされるとカタカナをタイプできるようになります。もう一度押すとロックは解除され英文字がタイプできる状態にもどります。SHIFT キーを押した状態では ッ、ア、イ、ウ、エ、オ (カナ小文字) など各キーの右側に刻印された文字をタイプできます。

#### NUM (ニューメリックロックキー)

テンキーのない機種で数字を続けて入力するときに使います。

たとえば、NUM キーがロックされた状態で、L<sub>1</sub>を押すと“3”が入力できます。



▶参照 テンキー→2.3の「テンキーについて」

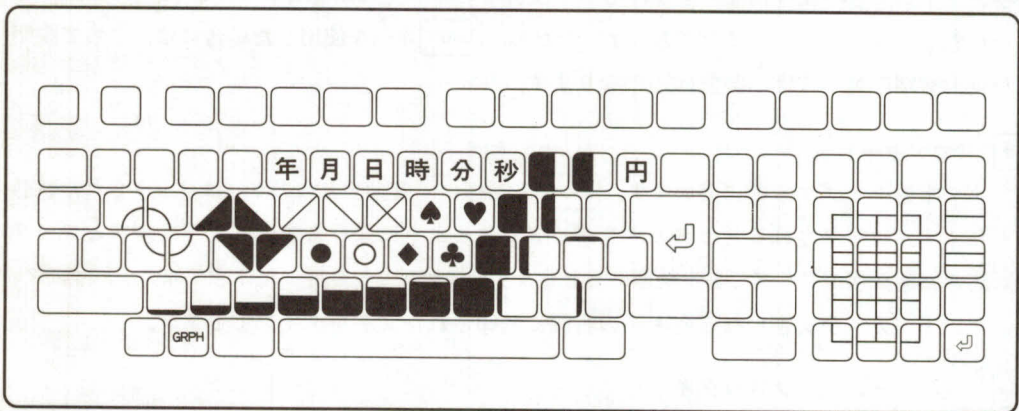
注意・**NUM**キーはテンキーのない機種にのみあります。

- ロックの可能なキーのロック状態は、機種により次の2種類があります。
  - ロック状態で、そのキーが沈み込むもの
  - ロック状態で、キーボード上のLEDが点灯するもの

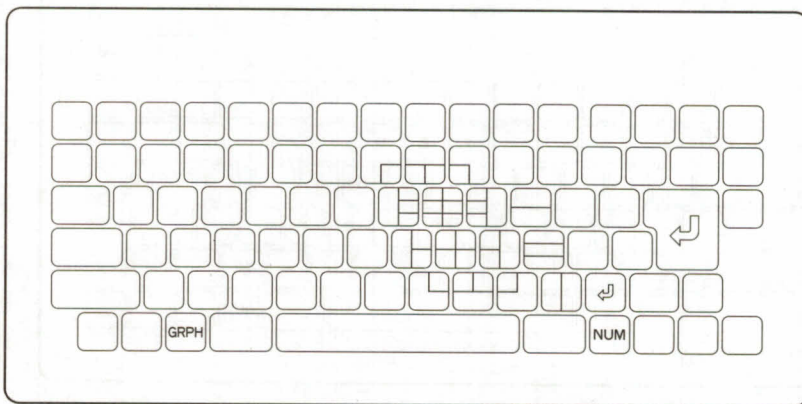
**GRPH** (グラフキー)

このキーを押しながら文字キーを押すとグラフィックシンボルをタイプすることができます。

グラフィックシンボルのキー配列




テンキーのない機種でも、**NUM**キーをロックし、**GRPH**キーを押しながら文字キーを押すと、タイプできます。



## ■文字編集に関係するキー


 (スペースキー)

空白をタイプするときに使います。


 (リターンキー)

ダイレクトモードではこのキーが押されると命令が実行されます。プログラムモード(行番号を付けて入力するとき)ではこのキーを押すことによって1行の入力が終わり、カーソルは次の行の先頭に移動します。


▶参照 ダイレクトモード、プログラムモード→『BASIC入門』

 (タブキー)


行の頭をそろえるときに使います。このキーを押すとカーソルを8桁単位で右へ移動させることができます。

 (デリートキー)

カーソルの左の1文字を削除するときに使います。カーソルより右にある文字は左に詰まります。

 (バックスペースキー)


バックスペースキーは、キーと同じ動作をします。


 (インサートキー)


このキーを押すと挿入モードになります。挿入モードで文字をタイプするとカーソルの位置に文字が挿入されていきます。カーソルより右にある文字は順に右に送られます。

 (カーソル移動キー)


カーソルを上下左右に動かすのに使います。

 (ホーム・クリアキー)

このキーを押すとテキスト画面のスクロール画面が消されカーソルはホームポジション(スクロール画面の左上隅)に移ります。キーを押しながらこのキーを押すと、画面はそのまま、カーソルだけがホームポジションに移動します。

 (クリアキー) (XLシリーズのみ)

このキーを押すと、スクロール画面が消されカーソルはホームポジションにもどります。

 (ホームキー) (XLシリーズのみ)

このキーを押すと、画面はそのままカーソルだけがホームポジションに移動します。

▶参照 スクロール画面→3.1の「スクロール画面の範囲」

## ■その他のキー

**STOP** (ストップキー)

ダイレクトモードの命令やプログラムの実行を強制的に中止します。

**COPY** (コピーキー)

このキーを押すとプリンタで画面のハードコピーをとることができます。

▶参照 ハードコピー→「3.3 画面ハードコピー」

**HELP** (ヘルプキー)

プログラムの実行中にエラーが起きた場合、その命令と位置を表示します。

▶参照 エラー位置の検出→2.4の「便利な編集コマンドとキー」

**ESC** (エスケープキー)

エスケープコード(&H1B)を入力することができます。

**ROLL DOWN** (ロールダウンキー)

**ROLL UP** (ロールアップキー)

DISK モード BASIC と ROM モード BASIC で働きが異なります。

DISK モード BASIC では、EDIT 命令の実行後、プログラム全体を上下にスクロールさせることができます。これはスクリーンエディタでプログラムの修正を行うのにたいへん便利です。

ROM モード BASIC では、テキスト画面を上下にスクロールさせるだけで、プログラム全体のスクロールを行うことはできません。

▶参照 EDIT 命令→2.4の「便利な編集コマンドとキー」、『BASICリファレンスマニュアル』スクリーンエディタ→「2.4 スクリーンエディタ」

**XFER** (変換キー)

日本語入力制御キーとして使います。

▶参照 日本語入力→『日本語入力ガイド』

**NFER** (無変換キー)

日本語入力制御キーとして使います。

---

注意： **NFER** キーのない機種もあります。











---

## f.1 ~ f.10 (ファンクションキー)

10種類のファンクションキーが使用できます。

BASICの起動時には各ファンクションキーに次に示す文字列が割り当てられています。C<sub>R</sub>は、



キーを押すのと同じ働きがあります。

|   |                    |  |                       |
|---|--------------------|--|-----------------------|
|  f.1 | load "             |  f.6  | save "                |
|  f.2 | auto               |  f.7  | key                   |
|  f.3 | go to              |  f.8  | print                 |
|  f.4 | list               |  f.9  | edit . C <sub>R</sub> |
|  f.5 | run C <sub>R</sub> |  f.10 | cont C <sub>R</sub>   |

これらのファンクションキーを使えば文字列のタイプを短い時間で行うことができます。さらにこのファンクションキーには、利用者が自由に文字列を定義することができます。

それには、BASICのKEY命令を使います。

たとえば、

KEY 3, "files"+CHR\$(13) (はリターンキーを押すことを表す)

とするとキーに、"files C<sub>R</sub>"が定義されます。

▶ 参照 KEY命令 → 『BASICリファレンスマニュアル』



定義されているファンクションキーの内容は画面の最下行に表示されるようになっています。表示内容は、40桁モードのときと80桁モードのときで異なります。表示される文字数は各キーごとに6文字です。

## 40桁モード

キーを押さない場合 ...  ~ が表示される

キーを押した場合 ...  ~ が表示される

## 80桁モード

 ~ がすべて表示される

定義されているファンクションキーの文字列をすべてテキスト画面に表示させるには、KEY LIST命令を実行します。

KEY LIST 

**CTRL** (コントロールキー)

他のキーと組み合わせて、特殊な機能を実行することができます。

▶ 参照 **CTRL** キーの機能→2.4の「**CTRL** キーを使った便利な編集機能」

---

参考：**STOP**、**HELP**、**f・1**～**f・10** キーは、それぞれ **ON STOP GOSUB**、**ON HELP GOSUB**、**ON KEY GOSUB** 命令を使って、キー割り込みの処理を行うことができます。

---

▶ 参照 **ON STOP GOSUB** 命令、**ON HELP GOSUB** 命令、**ON KEY GOSUB** 命令→  
『BASICリファレンスマニュアル』

---

注意：**^** キーの **,** (**SHFT** + **^**) は BASIC において取り扱うことはできません。  
**vf・1**～**vf・5** は BASIC では使用できません。

---

### ■テンキーについて

キーボードの右には演算子などのキーとともに数字のキーが並んでいます。これをテンキーといいます。数字を続けて入力する場合などに便利です。

---

注意：キーの配置は、機種によって異なります。

---

## 2.4 スクリーンエディタ



N<sub>88</sub>-BASIC(86)にはプログラムの作成および修正が簡単に行えるように、画面編集機能が用意されています。これをスクリーンエディタといいます。

この節では文字キー、特殊キーを使って BASIC プログラムを編集する方法の概略を説明します。

### ■ BASIC プログラムの入力

BASIC が起動するとスクリーンエディタの機能が使えるようになり、文字キーを押すと対応する文字が画面のカーソルの位置に表示されます。




BASIC のプログラムは必ず命令の前に 1 から 65529 の番号をつけて入力しなければなりません。これを行番号といいます。次のように入力してみてください。これを元にしてスクリーンエディタの使い方を説明します。

```
10 PRINT " Hardware " 
20 PRINT " Software " 
```


### ■ プログラムの修正

#### ● 文字の削除、変更、挿入







##### 文字の削除

10 行目の Hardware を削除するには、まずカーソル移動キーを使って行末の”の位置にカーソルを移動します。この位置で  キーを押すと e が削除されます。続けて  キーを 7 度押すと Hardware がすべて削除されます。削除したら  キーを押します。

##### 文字の変更

20 行目の Software を Computer に変更するには、カーソル移動キーを使ってカーソルを S の位置まで移動し Computer とタイプします。タイプしたら  キーを押します。




##### 文字の挿入

10 行目の””の間に Super と入れるには、まずカーソル移動キーを使ってカーソルを行末の”の位置まで移動し  キーを押します。  キーを押すと挿入モードになります。ここで Super とタイプします。挿入モードでは、タイプされた文字はカーソルの前に挿入されていきます。挿入モードを抜けるにはもう一度  キーを押すかカーソル移動キーまたは  キーを押します。ここでは、  キーを押してください。タイプしたら  キーを押します。

修正の結果は次のようになります。

```
10 PRINT "Super"  
20 PRINT "Computer"
```

---

**注意：**1つの行のプログラムのタイプ・修正が終わったら、必ず  キーを押してください。  キーを押すことによって初めて、タイプ・修正した内容がコンピュータのメモリに格納されます。これを入力するといいます。また、  キーを押すときにはカーソルがその行にあればどの位置にあってもかまいません。カーソルをいちいち行末に移動する必要はありません。

---

### オートリピート機能について

同じキーを約 0.5 秒押し続けると同じ文字が連続して表示されていきます。これをオートリピート機能といいます。この機能は文字キーばかりでなく特殊キー(ファンクションキーを除く)にもあります。同じ文字を連続してタイプしたいときに便利です。

### ●行の挿入、削除

#### 行の挿入

行を増やしたいときには、現在ある行番号の間の番号、あるいは前後の行番号をつけた命令を入力します。

たとえば、前の例で 10 行目と 20 行目の間に他の命令を挿入したい場合には、

```
15 PRINT "Personal" 
```

と入力します。

#### 行の削除

行を削除したいときには、削除したい行の行番号だけを入力します。


たとえば、10 行を削除したいときには、

```
10 
```

と入力します。

### ●修正結果の確認

修正結果を確認するには LIST 命令を使います。LIST はプログラムを画面表示させる命令です。今までの修正結果を表示してみます。

```
list 
15 PRINT "Personal"
20 PRINT "Computer"
Ok
■
```

正しく修正している場合にはこのようになります。このようにならない場合には、今まで使ったキー操作法を使って修正してください。BASIC のスクリーンエディタ機能は画面表示されている文字に対して有効ですから、LIST 命令で画面表示させた文字の位置にカーソルを移動して修正することができます。


▶参照 LIST 命令→『BASICリファレンスマニュアル』

### ■便利な編集コマンドとキー

#### HELP キー

プログラムを実行したときプログラム中にエラーがあると、エラーのある命令のところで実行が中断されます。このときエラーの種類とそのエラーの起きた行番号を表示するエラーメッセージが表れます。ここで、**HELP** キーを押すとその行が表示され、エラーのある命令かその次のワードの先頭でカーソルが点滅します。なお、ここでワードとは、空白、コンマ、セミコロン、コロンのいずれかで区切られた文字の並びのことです。

```
20 PRINT A : PLINT B : GOTO 10
```

この場合エラーは "PLINT" の "L" で、カーソルは B にあります。カーソル移動キーを L に移動し R をタイプし直して  キーを押します。

このとき、LIST. を使うと、その行だけを表示させることができます。

#### EDIT 命令

プログラムを全体に渡って修正したい場合があります。そのようなときには EDIT 命令を使います。

```
EDIT <エラーが起きた行番号>
```

指定した行が画面の最上位に表示されエディットモードに入ります。このモードでは **ROLL UP**, **ROLL DOWN** キーを使うことにより、プログラム全体をスクロールしながら表示させ、訂正することができます。この場合も各行の訂正が終わったら必ず **↵** キーを押さなければなりません。

エディットモードは、RUN、GOTOなどでプログラムを実行すると解除されます。

### LIST 命令

プログラムを画面表示します。画面表示は **CTRL** + **S** で一時停止でき、**STOP**, **COPY**, **SHIFT**, **カナ**, **GRPH**, **CAPS**, **CTRL**, **CTRL** + **C** 以外のキーを押すと表示が再開されます。**CTRL** + **C** を実行すると画面表示は中止され、コマンドモードにもどります。

### RENUM 命令

プログラムの行番号をつけかえます。行番号の間隔や行番号を変更することができます。なお、この命令でプログラムの各行の順番が変わることはありません。

### NEW 命令

メモリ上のプログラムを消去します。新規にプログラムを編集するときメモリをクリアしておく場合に使います。

## ■ **CTRL** キーを使った便利な編集機能

プログラムを編集するときに文字キーと **CTRL** キーを組み合わせることで使うことによって編集を効率よく行うことができます。

|                        |  |
|------------------------|--|
| <b>CTRL</b> + <b>A</b> | <b>HELP</b> キーと同じ機能を果たします。                                 |
| <b>CTRL</b> + <b>B</b> | カーソルを1つ左のワードの先頭に位置づけます。                                    |
| <b>CTRL</b> + <b>C</b> | <b>STOP</b> キーと同じ。プログラムの実行を強制的に中止します。                      |
| <b>CTRL</b> + <b>D</b> | カーソルのある位置からそのワードの終わりまでを削除します。                              |
| <b>CTRL</b> + <b>E</b> | カーソルのある位置からその行の終わりまでを消去します。                                |
| <b>CTRL</b> + <b>F</b> | カーソルを1つ右のワードの先頭に位置づけます。                                    |
| <b>CTRL</b> + <b>G</b> | スピーカ(ビーブ音)を鳴らします。  |
| <b>CTRL</b> + <b>H</b> | <b>BS</b> キーと同じ。カーソルの左の1文字を削除します。                          |
| <b>CTRL</b> + <b>I</b> | <b>TAB</b> キーと同じ。タブ位置までカーソルを移動させます。                        |
| <b>CTRL</b> + <b>J</b> | ラインフィードを挿入します。挿入モードのときはカーソルのある位置からその行の終わりまでを分割して次の行に移動します。 |
| <b>CTRL</b> + <b>K</b> | <b>SHIFT</b> + <b>HOME CLR</b> キーと同じ。カーソルをホームポジションに移動します。  |
| <b>CTRL</b> + <b>L</b> | <b>HOME CLR</b> キーと同じ。テキスト画面をクリアします。                       |
| <b>CTRL</b> + <b>M</b> | <b>↵</b> キーと同じです。  |

|      |   |   |   |
|------|---|---|---|
| CTRL | + | O | テキスト画面の表示を中止します。再び <b>CTRL</b> + <b>O</b> を入力すると再開されます。   |
| CTRL | + | R | <b>INS</b> キーと同じ。挿入モードになります。  |
| CTRL | + | S | プログラムの実行を一時停止します。 <b>STOP</b> , <b>COPY</b> , <b>SHIFT</b> , <b>カナ</b> , <b>GRPH</b> , <b>CAPS</b> , <b>CTRL</b> , <b>CTRL</b> + <b>C</b> 以外のキーを入力すれば実行が再開されます。 |
| CTRL | + | U | カーソルのある行を消去します。   |
| CTRL | + | X | カーソルのある行の行末にカーソルを移動させます。  |

**注意：**ワードとは、空白、コンマ(,)、コロン(:)などの記号で区切られた文字の並びのことです。

### ■日本語文字使用時の注意

N<sub>88</sub>-日本語 BASIC(86)では、2バイトの日本語文字を使用することができます。

スクリーンエディタ機能を使って日本語文字を入力したり、表示したりする際には次のような点に注意してください。

- ・1文字の日本語文字が2行にまたがると表示が乱れます。
- ・日本語文字の左右どちらか半分に1バイト文字を重ねるような入力をする则表示が乱れます。
- ・日本語文字の上にあるカーソルの横幅は1バイト文字の2倍の大きさになります。

日本語文字を使用する方法は、「第5章 日本語処理」に詳しく書かれてあります。

▶参照 日本語文字の使用→「第5章 日本語処理」



# 第3章

## 画面

N<sub>88</sub>-BASIC(86)の画面には大きく分けてテキスト画面とグラフィック画面の2種類があります。テキスト画面には文字を表示させることができます。いっぽう、グラフィック画面には点や線や絵などを表示させることができます。

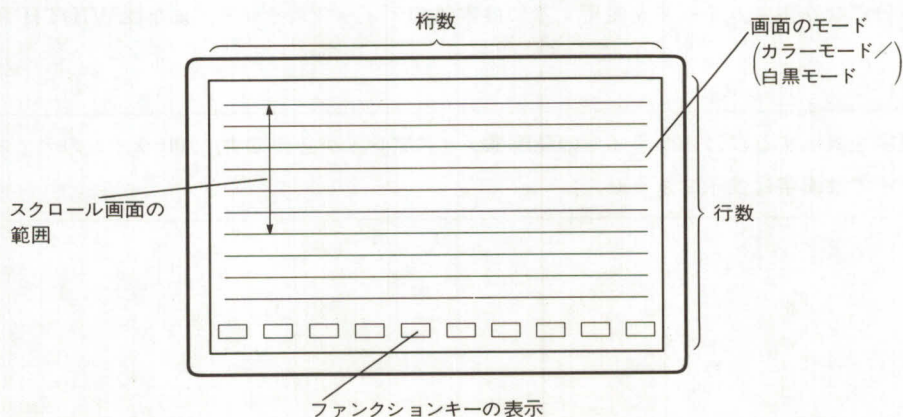
テキスト画面とグラフィック画面は独立したものです。ディスプレイにどちらか一方だけを表示したり、同時に重ねて表示することが自由にできます。

### 3.1 テキスト画面

テキスト画面は文字(キャラクタ)を表示するための画面で、1バイト文字で最大80桁×25行表示することができます。また、文字の色は8色の中から選択することができます。この画面を利用してBASICのプログラムを作成したり、漢字を含む文字を表示することができます。

テキスト画面は次のような項目を設定することが可能です。

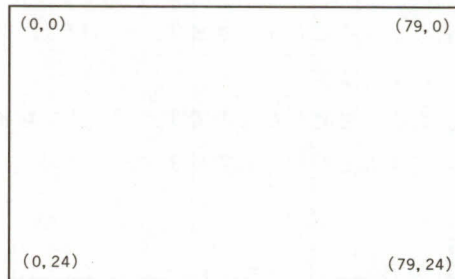
- 桁数と行数
- スクロール画面の範囲
- ファンクションキーの表示
- カラーモード/白黒モード
- テキスト画面の文字の色と表示モード



### ■桁数と行数

キャラクタを画面に出力するためには、行(row)および桁(column)の概念を用います。1つの桁には1バイト文字1文字が入ります。テキスト画面の原点は左上隅の桁で、(0, 0)で表されます。LOCATE 命令や COLOR@命令で参照されるテキスト画面の座標は(桁, 行)、つまり(X, Y)の順で表されます。テキスト画面の座標をキャラクタ座標と呼びます。

テキスト画面の座標の例(80桁×25行)



テキスト画面には桁数、行数の異なる4つのモードがあります。各モードの桁数、行数はそれぞれ次の表のとおりです。

| モードの名称    | 表示文字数(横×縦) |
|-----------|------------|
| 40桁20行モード | 40×20      |
| 80桁20行モード | 80×20      |
| 40桁25行モード | 40×25      |
| 80桁25行モード | 80×25      |

1つの桁には1バイト文字が1つ入ります。2バイトの日本語文字の場合は表示可能な文字数はこの半分になります。モードを変更するには本体のディップスイッチ、または WIDTH 命令を使います。

---

**注意：**漢字を表示するには400ラインの高解像ディスプレイが必要です。200ラインのディスプレイでは漢字は表示できません。

---

ディップスイッチでモードを設定するには本体の電源を OFF にした状態で行います。

| モード          | ディップスイッチ SW2 による設定 |     |
|--------------|--------------------|-----|
|              | 3 番                | 4 番 |
| 40 桁 20 行モード | OFF                | OFF |
| 80 桁 20 行モード | ON                 | OFF |
| 40 桁 25 行モード | OFF                | ON  |
| 80 桁 25 行モード | ON                 | ON  |

ディップスイッチによる指定は BASIC が起動した初期状態を設定するもので、WIDTH 命令が実行されるまで有効です。

▶参照 ディップスイッチ→『ガイドブック』

#### WIDTH 命令

WIDTH <桁数>[, <行数>]

例) 40 桁×20 行にする。

WIDTH 40, 20

WIDTH 命令による指定はリセットするか、次の WIDTH 命令が実行されるまで有効です。

#### ■スクロール画面の範囲

テキスト画面にデータを表示していくとき、表示データが画面いっぱいになるまでは画面の最上行から画面の下へ向かって 1 行ずつ順々に表示されていきます。表示されるデータが画面いっぱいになると、最上行にある一番古いデータが消え、残りの画面データは 1 行上へ移動し、空白になった最下行には最新の表示データが入ります。これが続けて行われると、データが次々に上方へ移動して行くようにみえます。これを画面の“スクロール”といいます。

BASIC が起動した初期状態では画面全体がスクロールしますが、スクロールする範囲を限定し、スクロールしない部分には同じ内容を常に表示させることができます。この限定されたスクロール範囲のことをスクロール画面と呼びます。

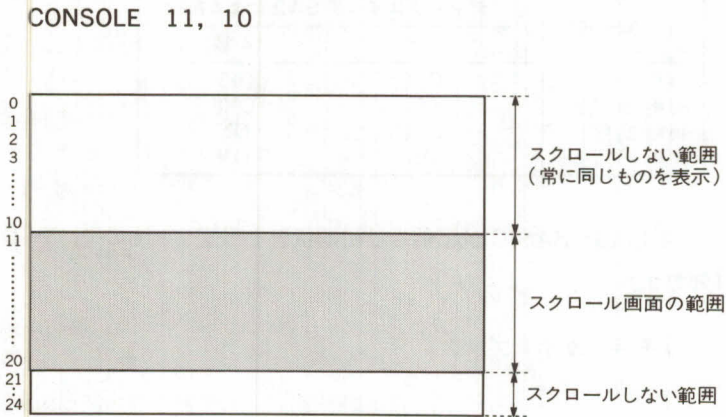
スクロール画面の範囲の指定は CONSOLE 命令で行います。CONSOLE 命令の 1 番目のパラメータにスクロール開始行を、2 番目のパラメータにスクロール行数を指定します。

#### CONSOLE 命令

CONSOLE [**<スクロール開始行>**][, **<スクロール行数>**][, **<ファンクションキー表示スイッチ>**][, **<カラー／白黒スイッチ>**]

スクロール開始行の指定は画面の最上行から 0, 1, 2, 3, 4……と数えます。20 行のモードの場合は 19 まで、25 行モードの場合は 24 までの数値を指定できます。

例) 25行のモードで最上行から10行目までをスクロールさせないで11行目から20行目までをスクロールさせる。



### ■ファンクションキーの表示

BASICが起動した初期状態では画面の最下行にはファンクションキーに登録された文字列の一部が表示されていますが、CONSOLE命令を使って表示させないようにすることができます。ファンクションキーの表示をする場合にはCONSOLE命令の3番目のパラメータに1を指定し、表示しない場合には0を指定します。

|               |       |
|---------------|-------|
| CONSOLE , , 0 | 表示しない |
| CONSOLE , , 1 | 表示する  |

### ■カラーモード/白黒モード

テキスト画面に表示する文字の色をカラーにするか、白黒にするかを設定することができます。BASIC起動直後は白黒モードになっています。

カラーモード/白黒モードの設定はCONSOLE命令で行います。4番目のパラメータに1を指定するとカラーモードに、0を指定すると白黒モードになります。

|                 |        |
|-----------------|--------|
| CONSOLE , , , 1 | カラーモード |
| CONSOLE , , , 0 | 白黒モード  |

ここでいうカラーモード、白黒モードはカラーディスプレイを使うかモノクロディスプレイを使うかということとは別の問題です。たとえば、カラーモードでモノクロディスプレイを使った場合は色は濃淡で表されます。カラーモード、白黒モードとディスプレイの関係は次のようになります。

| テキスト画面のモード | カラーディスプレイ | モノクロディスプレイ |
|------------|-----------|------------|
| カラーモード     | ○         | △          |
| 白黒モード      | ○         | ○          |

○：正常に表示される △：濃淡で表示される

## ■テキスト画面の文字の色と表示モード

テキスト画面に表示する文字の色や表示モードを〔1〕COLOR 命令を使って変えることができます。〔1〕COLOR 命令の書式は次のとおりです

### 〔1〕COLOR 命令

COLOR [〈ファンクションコード〉][, 〈バックグラウンドカラー〉][, 〈ボーダーカラー〉][, 〈フォアグラウンドカラー〉][, 〈パレットモード〉]

このなかで、テキストの文字の色を変えるパラメータは1番目の〈ファンクションコード〉です。他のパラメータはテキスト画面には関係なく、グラフィック画面の色の指定に関するものです。

### ▶参照 グラフィック画面の色の指定→「4.3 グラフィック画面の色」

〈ファンクションコード〉の働きはテキスト画面がカラーモードになっているか、白黒モードになっているかによって異なります。カラーモードの場合は8色の中から選択できます。白黒の場合は表示のモードを選択できます。

| ファンクションコード | カラーモードの場合<br>(色) | 白黒モードの場合<br>(表示モード) |
|------------|------------------|---------------------|
| 0          | 黒                | ノーマル                |
| 1          | 青                | シークレット              |
| 2          | 赤                | ブリンク                |
| 3          | 紫                | シークレット(1と同じ)        |
| 4          | 緑                | リバース                |
| 5          | 水色               | リバースシークレット          |
| 6          | 黄                | リバースブリンク            |
| 7          | 白                | リバースシークレット(5と同じ)    |

ノーマルとは通常の表示モードです。リバースにすると背景と文字の色が逆転します。シークレットは入力された文字を表示せず、文字数分だけカーソルが移動していきます。ブリンクは文字を点滅させます。BASIC 起動直後はノーマルになっています。

例1) テキスト画面の文字を紫にする。

```
CONSOLE , , , 1 ..... カラーモードに設定
COLOR 3 ..... 文字の色を紫にする
```

例 2) 文字を反転して表示する。

```
CONSOLE , , , 0 ..... 白黒モードに設定
COLOR 4 ..... 反転の表示
```

**範囲を指定する場合**

[1]COLOR 命令では、テキスト画面に表示する文字の色あるいは表示モードの設定はこれから表示させる文字を対象としていました。それに対してCOLOR@ 命令を使うと、すでに表示されている画面上の任意の範囲の文字に対して、この設定を行うことができます。

**COLOR@命令**

```
COLOR@ (X1, Y1)-(X2, Y2)[, <ファンクションコード>]
```

ファンクションコードについては[1]COLOR 命令の場合と同様です。また、座標の指定はキャラクタ座標で行います。

例) 80 桁×20 行モードで画面の右上の部分の文字を赤にする。

```
CONSOLE , , , 1
COLOR@ (40, 0)-(79, 11), 2
```

## 3.2 グラフィック画面

グラフィック画面に絵や画像などを描く際には、場合に応じて最も適した画面を設定しておかなければなりません。N<sub>88</sub>-BASIC(86)のグラフィック画面には、分解能と色の種類(カラー/白黒)の違う4つのモードがあり、その中から適当なものを選択して使用することができます。

**■画面モード**

グラフィック画面には次の4つの画面モードがあります。

| 画面モード      | 画面のタイプ | 分解能(横×縦) |
|------------|--------|----------|
| カラーモード     | カラー    | 640×200  |
| 白黒モード      | 白黒     | 640×200  |
| 高分解能白黒モード  | 白黒     | 640×400  |
| 高分解能カラーモード | カラー    | 640×400  |

---

注意：高分解能のモードにするには400ラインの高解像度ディスプレイが必要です。200ラインのディスプレイでは表示できません。

---

カラーの細かいドットで美しい画像を描きたい場合には高分解能カラーモードを、白黒の粗いドットで画像を描きたい場合には白黒モードに設定するというように使い分けることができます。BASIC起動直後はカラーモードになっています。

なお、カラーモードおよび白黒モードの2つを合わせて低分解能モード、他の2つを合わせて高分解能モードと呼ぶこともあります。

画面モードの選択は SCREEN 命令の1番目のパラメータで行います。

### SCREEN 命令

SCREEN [〈画面モード〉][, 〈画面スイッチ〉][, 〈アクティブページ〉][, 〈ディスプレイページ〉]

画面モードの指定値とモードの関係は次のとおりです。

| 画面モードの指定値 | 画面モード      |
|-----------|------------|
| 0         | カラーモード     |
| 1         | 白黒モード      |
| 2         | 高分解能白黒モード  |
| 3         | 高分解能カラーモード |

例) 高分解能カラーモードを使う場合。

SCREEN 3

### ■ ページ

N<sub>88</sub>-BASIC(86)の各画面モードでは、絵や図形を描いた画面を同時に何枚も保持しておき、そのなかから何枚かを表示することができるようになっています。各モードにおける複数の画面のことをページといいます。

各画面モードにおけるページ数は次のとおりです。

| 画面モード      | ページ数    |
|------------|---------|
| カラーモード     | 4       |
| 白黒モード      | 12(16*) |
| 高分解能白黒モード  | 6(8*)   |
| 高分解能カラーモード | 2       |

\*のついている数値は4096色中・16色モード時のページ数を表します。

▶ 参照 4096色中・16色モード→「4.3 グラフィック画面の色」

たとえば、カラーモードでは4つのページを使えますので、4ページ分の絵や図形を描いておいてそのうちの1枚を選択して次々に表示することができます。

つまり、絵や図形を描く(書き込む)ページと画面に表示するページを別に設定できるのです。描くページのことを「アクティブページ」、表示するページのことを「ディスプレイページ」といいます。

アクティブページとディスプレイページの指定は SCREEN 命令の3番目と4番目のパラメータで行います。

使用可能なページ数はパレットモードによって異なります。したがって、〈アクティブページ〉と〈ディスプレイページ〉の指定値とその意味もパレットモードによって異なりますので、ここでは区別して記述します。BASIC起動直後はアクティブページ、ディスプレイページともページ1となっています。

▶ 参照 パレットモード→「4.3 グラフィック画面の色」

〈アクティブページ〉に指定できる値と書き込まれるページのページ番号との対応関係は次のとおりです。

8色中・8色モードあるいは4096色中・8色モードの場合

| 指定値  | 画面モード      | 書き込まれるページ番号 |
|------|------------|-------------|
| 0~3  | カラーモード     | 1~4         |
| 0~11 | 白黒モード      | 1~12        |
| 0~5  | 高分解能白黒モード  | 1~6         |
| 0, 1 | 高分解能カラーモード | 1, 2        |

4096色中・16色モードの場合

| 指定値  | 画面モード      | 書き込まれるページ番号 |
|------|------------|-------------|
| 0~3  | カラーモード     | 1~4         |
| 0~15 | 白黒モード      | 1~16        |
| 0~7  | 高分解能白黒モード  | 1~8         |
| 0, 1 | 高分解能カラーモード | 1, 2        |

〈アクティブページ〉の指定値と書き込まれるページとは必ず1対1に対応しています。したがって、同時に複数のページに書き込むことはできません。

〈ディスプレイページ〉に指定できる値と表示されるページのページ番号との対応関係は次のとおりです。

## 8色中・8色モードあるいは4096色中・8色モードの場合

| 指定値 | カラーモード<br>(0) | 白黒モード<br>(1)     | 高分解能白黒モード<br>(2) | 高分解能カラーモード<br>(3) |
|-----|---------------|------------------|------------------|-------------------|
| 0   | 全ページ表示しない     | 全ページ表示しない        | 全ページ表示しない        | 全ページ表示しない         |
| 1   | ページ1のみ表示      | ページ1のみ表示         | ページ1のみ表示         | ページ1のみ表示          |
| 2   | ページ2のみ表示      | ページ2のみ表示         | ページ2のみ表示         | ×                 |
| 3   | ×             | ページ1,2を合成表示      | ページ1,2を合成表示      | ×                 |
| 4   | ×             | ページ3のみ表示         | ページ3のみ表示         | ×                 |
| 5   | ×             | ページ1,3を合成表示      | ページ1,3を合成表示      | ×                 |
| 6   | ×             | ページ2,3を合成表示      | ページ2,3を合成表示      | ×                 |
| 7   | ×             | ページ1,2,3を合成表示    | ページ1,2,3を合成表示    | ×                 |
| 8   | 全ページ表示しない     | 全ページ表示しない        | 全ページ表示しない        | 全ページ表示しない         |
| 9   | ×             | ページ4のみ表示         | ×                | ×                 |
| 10  | ×             | ページ5のみ表示         | ×                | ×                 |
| 11  | ×             | ページ4,5を合成表示      | ×                | ×                 |
| 12  | ×             | ページ6のみ表示         | ×                | ×                 |
| 13  | ×             | ページ4,6を合成表示      | ×                | ×                 |
| 14  | ×             | ページ5,6を合成表示      | ×                | ×                 |
| 15  | ×             | ページ4,5,6を合成表示    | ×                | ×                 |
| 16  | 全ページ表示しない     | 全ページ表示しない        | 全ページ表示しない        | 全ページ表示しない         |
| 17  | ページ3のみ表示      | ページ7のみ表示         | ページ4のみ表示         | ページ2のみ表示          |
| 18  | ページ4のみ表示      | ページ8のみ表示         | ページ5のみ表示         | ×                 |
| 19  | ×             | ページ7,8を合成表示      | ページ4,5を合成表示      | ×                 |
| 20  | ×             | ページ9のみ表示         | ページ6のみ表示         | ×                 |
| 21  | ×             | ページ7,9を合成表示      | ページ4,6を合成表示      | ×                 |
| 22  | ×             | ページ8,9を合成表示      | ページ5,6を合成表示      | ×                 |
| 23  | ×             | ページ7,8,9を合成表示    | ページ4,5,6を合成表示    | ×                 |
| 24  | 全ページ表示しない     | 全ページ表示しない        | 全ページ表示しない        | 全ページ表示しない         |
| 25  | ×             | ページ10のみ表示        | ×                | ×                 |
| 26  | ×             | ページ11のみ表示        | ×                | ×                 |
| 27  | ×             | ページ10,11を合成表示    | ×                | ×                 |
| 28  | ×             | ページ12のみ表示        | ×                | ×                 |
| 29  | ×             | ページ10,12を合成表示    | ×                | ×                 |
| 30  | ×             | ページ11,12を合成表示    | ×                | ×                 |
| 31  | ×             | ページ10,11,12を合成表示 | ×                | ×                 |

×印指定不可

## 4096色中・16色モードの場合

| 指定値 | カラーモード<br>(0) | 白黒モード<br>(1)    | 高分解能白黒モード<br>(2) | 高分解能カラーモード<br>(3) |
|-----|---------------|-----------------|------------------|-------------------|
| 0   | 全ページ表示しない     | 全ページ表示しない       | 全ページ表示しない        | 全ページ表示しない         |
| 1   | ページ1のみ表示      | ページ1のみ表示        | ページ1のみ表示         | ページ1のみ表示          |
| 2   | ページ2のみ表示      | ページ2のみ表示        | ページ2のみ表示         | ×                 |
| 3   | ×             | ページ1,2を合成表示     | ページ1,2を合成表示      | ×                 |
| 4   | ×             | ページ3のみ表示        | ページ3のみ表示         | ×                 |
| 5   | ×             | ページ1,3を合成表示     | ページ1,3を合成表示      | ×                 |
| 6   | ×             | ページ2,3を合成表示     | ページ2,3を合成表示      | ×                 |
| 7   | ×             | ページ1,2,3を合成表示   | ページ1,2,3を合成表示    | ×                 |
| 8   | ×             | ページ4のみ表示        | ページ4のみ表示         | ×                 |
| 9   | ×             | ページ1,4を合成表示     | ページ1,4を合成表示      | ×                 |
| 10  | ×             | ページ2,4を合成表示     | ページ2,4を合成表示      | ×                 |
| 11  | ×             | ページ1,2,4を合成表示   | ページ1,2,4を合成表示    | ×                 |
| 12  | ×             | ページ3,4を合成表示     | ページ3,4を合成表示      | ×                 |
| 13  | ×             | ページ1,3,4を合成表示   | ページ1,3,4を合成表示    | ×                 |
| 14  | ×             | ページ2,3,4を合成表示   | ページ2,3,4を合成表示    | ×                 |
| 15  | ×             | ページ1,2,3,4を合成表示 | ページ1,2,3,4を合成表示  | ×                 |
| 16  | 全ページ表示しない     | 全ページ表示しない       | 全ページ表示しない        | 全ページ表示しない         |
| 17  | ×             | ページ5のみ表示        | ×                | ×                 |
| 18  | ×             | ページ6のみ表示        | ×                | ×                 |
| 19  | ×             | ページ5,6を合成表示     | ×                | ×                 |
| 20  | ×             | ページ7のみ表示        | ×                | ×                 |
| 21  | ×             | ページ5,7を合成表示     | ×                | ×                 |
| 22  | ×             | ページ6,7を合成表示     | ×                | ×                 |
| 23  | ×             | ページ5,6,7を合成表示   | ×                | ×                 |
| 24  | ×             | ページ8のみ表示        | ×                | ×                 |
| 25  | ×             | ページ5,8を合成表示     | ×                | ×                 |
| 26  | ×             | ページ6,8を合成表示     | ×                | ×                 |
| 27  | ×             | ページ5,6,8を合成表示   | ×                | ×                 |
| 28  | ×             | ページ7,8を合成表示     | ×                | ×                 |
| 29  | ×             | ページ5,7,8を合成表示   | ×                | ×                 |
| 30  | ×             | ページ6,7,8を合成表示   | ×                | ×                 |
| 31  | ×             | ページ5,6,7,8を合成表示 | ×                | ×                 |
| 32  | 全ページ表示しない     | 全ページ表示しない       | 全ページ表示しない        | 全ページ表示しない         |
| 33  | ページ3のみ表示      | ページ9のみ表示        | ページ5のみ表示         | ページ2のみ表示          |
| 34  | ページ4のみ表示      | ページ10のみ表示       | ページ6のみ表示         | ×                 |
| 35  | ×             | ページ9,10を合成表示    | ページ5,6を合成表示      | ×                 |
| 36  | ×             | ページ11のみ表示       | ページ7のみ表示         | ×                 |

| 指定表 | カラーモード<br>(0) | 白黒モード<br>(1)          | 高分解能白黒モード<br>(2)  | 高分解能カラーモード<br>(3) |
|-----|---------------|-----------------------|-------------------|-------------------|
| 37  | ×             | ページ 9,11 を合成表示        | ページ 5,7 を合成表示     | ×                 |
| 38  | ×             | ページ 10,11 を合成表示       | ページ 6,7 を合成表示     | ×                 |
| 39  | ×             | ページ 9,10,11 を合成表示     | ページ 5,6,7 を合成表示   | ×                 |
| 40  | ×             | ページ 12 のみ表示           | ページ 8 のみ表示        | ×                 |
| 41  | ×             | ページ 9,12 を合成表示        | ページ 5,8 を合成表示     | ×                 |
| 42  | ×             | ページ 10,12 を合成表示       | ページ 6,8 を合成表示     | ×                 |
| 43  | ×             | ページ 9,10,12 を合成表示     | ページ 5,6,8 を合成表示   | ×                 |
| 44  | ×             | ページ 11,12 を合成表示       | ページ 7,8 を合成表示     | ×                 |
| 45  | ×             | ページ 9,11,12 を合成表示     | ページ 5,7,8 を合成表示   | ×                 |
| 46  | ×             | ページ 10,11,12 を合成表示    | ページ 6,7,8 を合成表示   | ×                 |
| 47  | ×             | ページ 9,10,11,12 を合成表示  | ページ 5,6,7,8 を合成表示 | ×                 |
| 48  | 全ページ表示しない     | 全ページ表示しない             | 全ページ表示しない         | 全ページ表示しない         |
| 49  | ×             | ページ 13 のみ表示           | ×                 | ×                 |
| 50  | ×             | ページ 14 のみ表示           | ×                 | ×                 |
| 51  | ×             | ページ 13,14 を合成表示       | ×                 | ×                 |
| 52  | ×             | ページ 15 のみ表示           | ×                 | ×                 |
| 53  | ×             | ページ 13,15 を合成表示       | ×                 | ×                 |
| 54  | ×             | ページ 14,15 を合成表示       | ×                 | ×                 |
| 55  | ×             | ページ 13,14,15 を合成表示    | ×                 | ×                 |
| 56  | ×             | ページ 16 のみ表示           | ×                 | ×                 |
| 57  | ×             | ページ 13,16 を合成表示       | ×                 | ×                 |
| 58  | ×             | ページ 14,16 を合成表示       | ×                 | ×                 |
| 59  | ×             | ページ 13,14,16 を合成表示    | ×                 | ×                 |
| 60  | ×             | ページ 15,16 を合成表示       | ×                 | ×                 |
| 61  | ×             | ページ 13,15,16 を合成表示    | ×                 | ×                 |
| 62  | ×             | ページ 14,15,16 を合成表示    | ×                 | ×                 |
| 63  | ×             | ページ 13,14,15,16 を合成表示 | ×                 | ×                 |

×印指定不可

この表のように<ディスプレイページ>の指定値は表示されるページと1対1に対応しているわけではありませんので注意してください。

なお、白黒モードでは複数のページを同時に表示することができます。

例1) 高分解能カラーモードでページ1にグラフィック命令を実行しながら表示させる場合。

SCREEN 3, , 0, 1

例2) 白黒モードでページ3にグラフィック命令を実行しながらページ1, 2を表示させる場合。

SCREEN 1, , 2, 3

この場合、実行されているグラフィック命令の結果は画面には現れない。例3)で初めて現れる。

例3) ページ3を表示する場合。

SCREEN , , , 4

### ■画面スイッチ

画面スイッチとは、グラフィック画面を一時的に消去するスイッチです。SCREEN命令の2番目のパラメータによって設定します。画面スイッチに指定する値とグラフィック画面の表示の関係は次のとおりです。BASIC起動直後は表示を行うようになっています。

| 指定値     | グラフィック画面の表示    |
|---------|----------------|
| 0 または 1 | 表示を行う          |
| 2 または 3 | 現在の表示を一時的に消去する |

これはプログラムの作成中にグラフィック画面を一時的に消してプログラムのテキストを見やすくする場合などに便利です。

## 3.3 画面ハードコピー

ディスプレイに表示された画面のイメージをそのままプリンタで印刷することをハードコピーといいます。

N<sub>88</sub>-BASIC(86)には、次のようなハードコピー機能があります。

- テキスト画面、あるいはグラフィック画面だけのハードコピーをとる。
- テキスト画面とグラフィック画面を合成したハードコピーをとる。

### ■画面ハードコピー機能の種類と機能

通常の画面ハードコピー機能とカラーコピーや縮小/拡大のできる拡張画面ハードコピー機能の2種類があり、どちらかを選択することができます。ROMモードBASICでは、通常の画面ハードコピー機能のみとなります。

| BASICの種類     | 画面ハードコピー機能    | 機能   |
|--------------|---------------|--|
| ROMモードBASIC  | 通常の画面ハードコピー機能 | テキスト画面はプリンタの書体で印字される。<br>モノクロコピーのみ。  |
| DISKモードBASIC | 通常の画面ハードコピー機能 |  |
|              | 拡張画面ハードコピー機能  | テキスト画面だけを出力した場合はプリンタの印字体で出力され、グラフィック画面と合成して出力した場合は画面のドットイメージで印字される。<br>PC-PR201V系プリンタを使ってカラーコピーをとることができる。<br>PC-PR601系プリンタを使って縮小/拡大コピーをとることができる。 |

どの機能を使用するかは、ユーティリティ“switch.n88”を使って選択します。

また、使用するプリンタの種類によってシステムの設定を変更しなければなりません。これには、ユーティリティ“switch.n88”と“setup.n88”を使います。

詳しいハードコピー機能のユーティリティによる設定の方法については、「ハードコピー機能の設定方法」を参照してください。

### ■通常の画面ハードコピー機能

通常の画面ハードコピー機能は次のとおりです。

| キー操作        | COPY命令 | 機能  |
|-------------|--------|---|
| CTRL + COPY | COPY 1 | テキスト画面のみを出力。<br>書体はプリンタの印字体。  |
| GRPH + COPY | COPY 2 | グラフィック画面のみを出力。  |
| COPY        | COPY 3 | テキスト画面とグラフィック画面を合成して出力。<br>(COPY 1とCOPY 2の結果を重ねたイメージ)。<br>テキストの書体はプリンタの印字体。 |
| なし          | COPY 4 | グラフィック画面を縦方向に縮小して出力(640×200のモードのみ)。   |
| なし          | COPY 5 | テキスト画面とグラフィック画面を縦方向に縮小して出力。<br>テキストの書体はプリンタの印字体(640×200のモードのみ)。             |

### ■ 拡張画面ハードコピー機能

拡張画面ハードコピー機能は、使用するプリンタによって機能が異なりますので分けて説明します。

#### ● PC-PR201V 系のカラープリンタの場合

##### カラーコピーをとる場合

| キー操作        | COPY 命令 | 機 能  |
|-------------|---------|--|
| CTRL + COPY | COPY 1  | テキスト画面のみをカラーで出力。<br>書体はプリンタの印字体。                                 |
| GRPH + COPY | COPY 2  | グラフィック画面のみをカラーで出力。<br>白と黒は反転する。                                  |
| COPY        | COPY 3  | テキスト画面とグラフィック画面を合成してカラーで出力。<br>白と黒は反転する。<br>テキストの書体は画面のドットイメージ。  |
| なし          | COPY 4  | グラフィック画面のみをカラーで出力。<br>白と黒は反転しない。                                 |
| なし          | COPY 5  | テキスト画面とグラフィック画面を合成してカラーで出力。<br>白と黒は反転しない。<br>テキストの書体は画面のドットイメージ。 |

注意：カラーコピーは 8 色中・8 色モードでのみ使用可能です。4096 色中・8 色、4096 色中・16 色モードではモノクロのハードコピーになります。

##### モノクロコピーをとる場合

| キー操作        | COPY 命令 | 機 能  |
|-------------|---------|--|
| CTRL + COPY | COPY 1  | テキスト画面のみを出力。<br>書体はプリンタの印字体。                   |
| GRPH + COPY | COPY 2  | グラフィック画面のみを出力。                                 |
| COPY        | COPY 3  | テキスト画面とグラフィック画面を合成して出力。<br>テキストの書体は画面のドットイメージ。 |

## ● PC-PR601 系のページプリンタの場合

PC-PR601 系のページプリンタを使用する場合には、印字方向や印字する倍率を指定することができます。

| キー操作        | COPY 命令 | 機 能  | 印字方向     | 印字倍率 |
|-------------|---------|--|----------|------|
| CTRL + COPY | COPY 1  | テキスト画面のみを出力。<br>書体はプリンタの印字体。                   | 水平または垂直。 | —    |
| GRPH + COPY | COPY 2  | グラフィック画面のみを出力。                                 | 水平または垂直。 | *    |
| COPY        | COPY 3  | テキスト画面とグラフィック画面を合成して出力。<br>テキストの書体は画面のドットイメージ。 | 水平または垂直。 | *    |

\*：標準、1/3 倍、2/3 倍、3/3 倍、4/3 倍のうちいずれか。

印字方向の欄の“水平”は用紙の走行方向が画面の上下に対応していることを意味します。また、“垂直”は用紙の走行方向が画面の左右に対応していることを意味します。

印字倍率の欄の“標準”とは PC-PR201 系のプリンタで出力される大きさを表します。

## ● PC-PR201V 系、PC-PR601 系以外のプリンタの場合

| キー操作        | COPY 命令 | 機 能  |
|-------------|---------|--|
| CTRL + COPY | COPY 1  | テキスト画面のみを出力。<br>書体はプリンタの印字体。                   |
| GRPH + COPY | COPY 2  | グラフィック画面のみを出力。                                 |
| COPY        | COPY 3  | テキスト画面とグラフィック画面を合成して出力。<br>テキストの書体は画面のドットイメージ。 |

■ハードコピー機能の設定方法

各ハードコピー機能の設定に使用するユーティリティは次のとおりです。

| 機能                   |                    |             | ユーティリティの設定            |                      |        |                           |       |
|----------------------|--------------------|-------------|-----------------------|----------------------|--------|---------------------------|-------|
|                      |                    |             | switch.n88 (画面ハードコピー) |                      |        | setup.n88 (画面ハードコピーの機能設定) |       |
|                      |                    |             | PC-PR201系<br>プリンタ     | 拡張画面<br>ハードコピー<br>機能 | カラーコピー | 方向                        | 倍率    |
| 通常の画面ハードコピー機能        |                    |             | 使う/使わない               | 使わない                 | —      | —                         | —     |
| 拡張画面<br>ハードコピー<br>機能 | PC-PR201V<br>系プリンタ | カラー<br>コピー  | 使う                    | 使う                   | カラー    | 水平                        | 標準    |
|                      |                    | モノクロ<br>コピー | 使う                    | 使う                   | モノクロ   | 水平                        | 標準    |
|                      | PC-PR601系<br>プリンタ  |             | 使う                    | 使う                   | モノクロ   | それぞれ可                     | それぞれ可 |
|                      | その他のプリンタ           |             | 使う/使わない               | 使う                   | モノクロ   | 水平                        | 標準    |

▶ 参照 switch.n88 →「10.14 メモリスイッチの設定」

setup.n88 →「10.8 システムディスク属性の設定」

# 第4章

## グラフィックス

N<sub>88</sub>-BASIC(86)にはグラフィックスを扱うための豊富な命令が用意されています。これらの命令を組み合わせることで、さまざまな線、図形などを描くことができます。この章ではグラフィックスを扱う上で知っておかなければならない次のような重要な概念を説明します。

- グラフィック画面の座標系
- パレット機能
- タイリング
- 画像の転送

なお、個々のグラフィック命令についての詳細は『BASICリファレンスマニュアル』を参照してください。

### 4.1 グラフィック画面の座標系

グラフィック画面に図形を表示する際には、画面のどの位置に出力するかを指定するために“座標”という概念を用います。この座標は数学などで用いられる座標と同じ考え方のもので、2つの直交する座標軸の原点からの距離を指定することによって平面上の1点を決めるものです。

N<sub>88</sub>-BASIC(86)のグラフィック画面には、オリジナルスクリーン座標系、スクリーン座標系、ワールド座標系の3つの座標系が用意されています。この座標系をうまく使うことによって複雑な図形を描いたり、図形の移動、拡大、縮小などを用意に行うことができます。

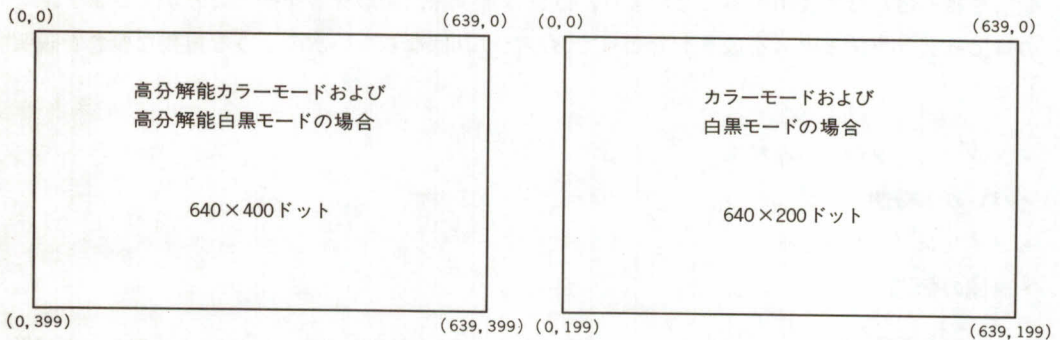
#### グラフィック画面の3つの座標系




## ■オリジナルスクリーン座標系

オリジナルスクリーン座標系はディスプレイ装置と結びついた座標系です。この座標系の大きさはディスプレイ装置と対応しており、座標上の各点が画面を分解するドット数と一致しています。つまり、高分解能カラーモードまたは高分解能白黒モードでは左上隅を(0, 0)とし右下隅を(639, 399)とする座標系です。カラーモードまたは白黒モードでは左上隅を(0, 0)とし右下の隅を(639, 199)とする座標系です。

### オリジナルスクリーン座標系



なお、この章のグラフィックスの説明は、特に断わり書きがないかぎり高分解能カラーモードの場合について説明します。本文中にある例を実行する場合には事前に SCREEN 命令で高分解能カラーモードの設定を行ってください。ダイレクトモードで“SCREEN 3 

▶参照 ダイレクトモード→『BASIC入門』

## ■ビューポートとスクリーン座標系

BASICの命令を使って描いた図形を表示する際には、ディスプレイ画面の中で表示領域を限定することができます。限定した表示領域をビューポートといいます。ビューポートの設定には VIEW 命令を使います。

### VIEW 命令

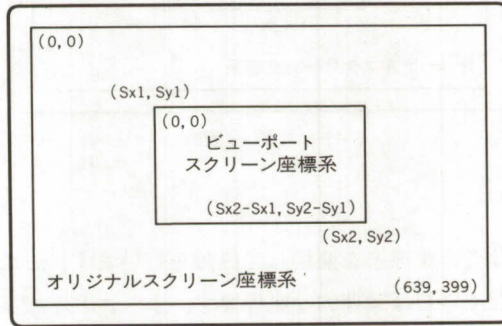
VIEW (Sx1, Sy1)-(Sx2, Sy2)[, <領域色>][, <境界色>]

ここでの座標指定はディスプレイの画面に対応したオリジナルスクリーン座標系で行います。オリジナルスクリーン座標系の(Sx1, Sy1)を左上隅、(Sx2, Sy2)を右下隅とする区画が図形を表示する範囲になります。なお、Sx1<Sx2, Sy1<Sy2が成り立たない場合、あるいはこれらの座標がディスプレイ画面から外れている場合にはエラーになります。

ビューポート内部の座標系をスクリーン座標系といいます。スクリーン座標系は、ビューポートの左上隅を原点(0, 0)としてドット単位に座標をとり直したものです。

領域色にパレット番号を指定するとビューポートの内部がその色でぬりつぶされます。境界色にパレット番号を指定するとビューポートの枠がその色で描かれます。

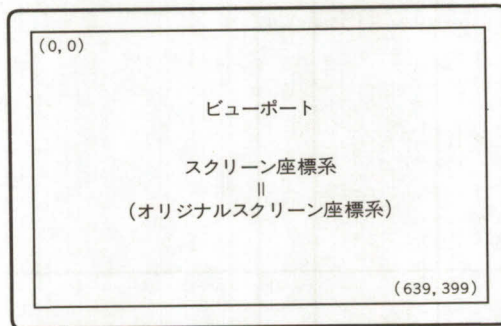
### VIEW 命令の図解



ディスプレイ画面

VIEW 命令を実行しない状態では、ディスプレイの画面全体がビューポートになっています。したがって、スクリーン座標系=オリジナルスクリーン座標系となります。

### VIEW を実行しない初期状態のビューポート

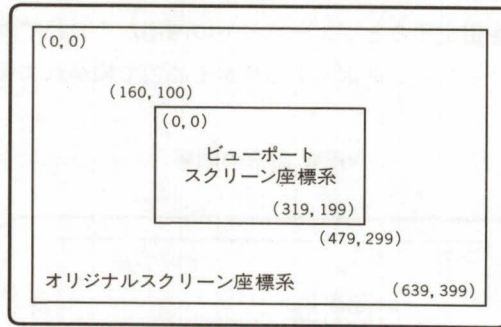


ディスプレイ画面

VIEW (160, 100) - (479, 299) を実行した場合、ビューポートは画面の中央になります。スクリーン座標系は(0, 0) - (319, 199)の範囲に新しく設定されます。

▶ 参照 パレット番号→4.3の「パレットモード」

VIEW (160, 100)–(479, 299)の実行結果

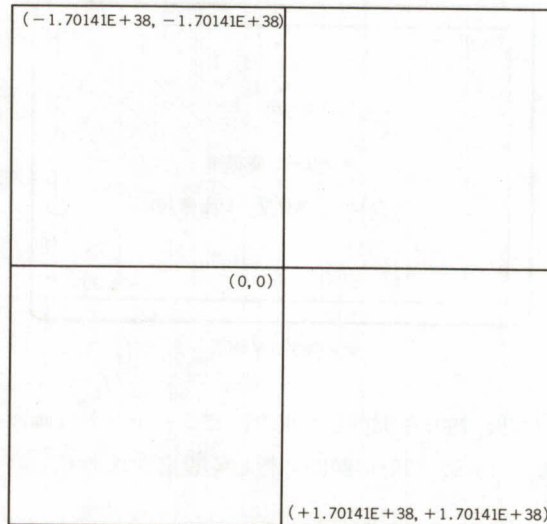


ディスプレイ画面

■ワールド座標系

大部分の描画コマンドはこの座標系を使用して描画を行います。この座標系はディスプレイ画面に対応したものではなくいわば論理的な座標です。ワールド座標系の大きさは縦横とも  $-1.70141E+38 \sim +1.70141E+38$  の範囲まで許されます。ユーザーはこの範囲の領域に PSET, DRAW, CIRCLE, LINE などのグラフィック命令を使って絵を描くことができます。ワールド座標系のおかげでスクリーンの物理的な大きさに制限されない広いワールド座標に、線を引いたりグラフや図を描いたりすることができます。

ワールド座標系



▶ 参照 DRAW, CIRCLE, LINE命令 → 『BASICリファレンスマニュアル』

## ■ウィンドウ

BASIC が設定した論理上の座標系であるワールド座標系は、範囲がたいへん広いのでワールド座標系のすべての1ドット1ドットをディスプレイのドットに対応させて表示することはできません。そこで、ワールド座標に描かれた図形を画面に表示させるためには、ワールド座標上の任意の範囲を、実際のディスプレイ上の表示領域(ビューポート)に割り当てます。この任意の領域のことをウィンドウと呼びます。

このウィンドウの設定を行うのが WINDOW 命令です。

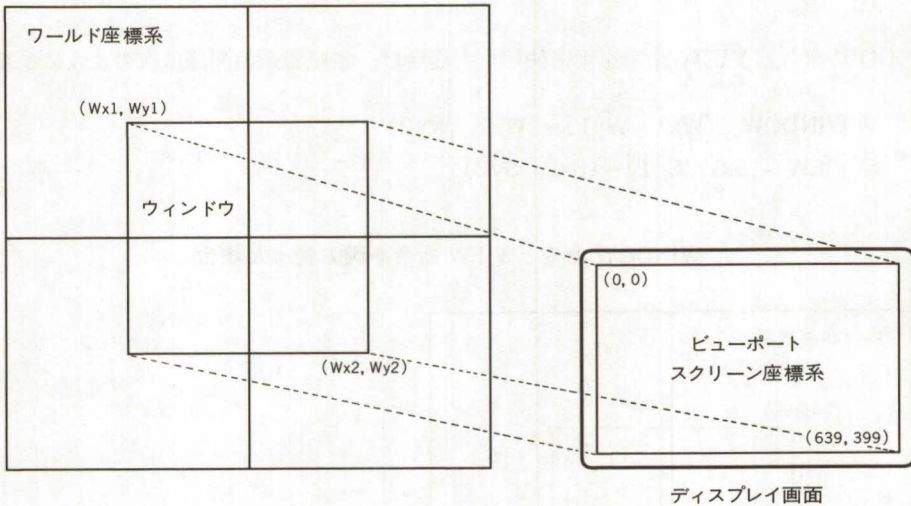
### WINDOW 命令

WINDOW (Wx1, Wy1)-(Wx2, Wy2)

ワールド座標上の(Wx1, Wy1)を左上隅、(Wx2, Wy2)を右下隅とする範囲がウィンドウになります。座標の指定はワールド座標で行います。Wx1<Wx2, Wy1<Wy2 が成り立たない場合、あるいはこれらの座標がワールド座標系から外れている場合にはエラーになります。

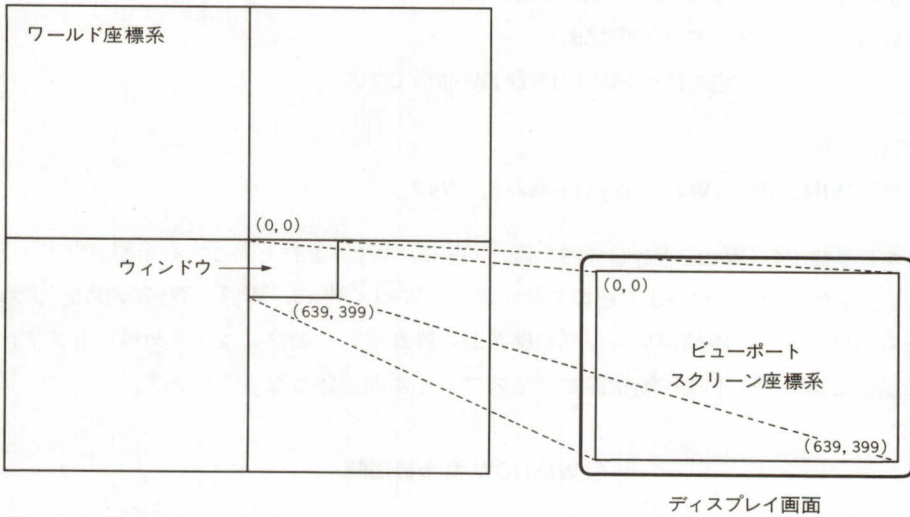
この例はビューポートは初期状態でディスプレイ画面全体となっています。

### WINDOW 命令の図解



これは WINDOW 命令を実行する前の初期状態です。ワールド座標系の  $(0, 0) - (639, 399)$  がスクリーン座標系の  $(0, 0) - (639, 399)$  に対応しています。

WINDOW 命令を実行しない初期状態のウィンドウ

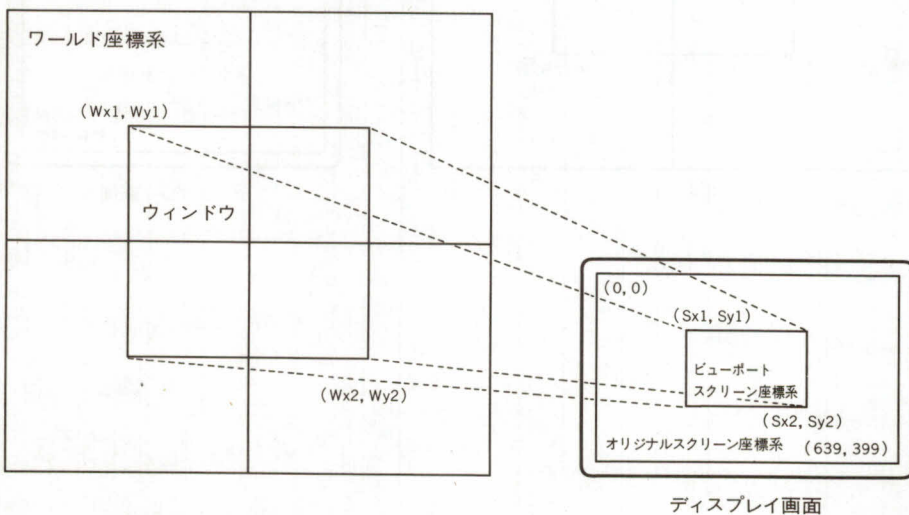


WINDOW 命令と VIEW 命令を両方使用した場合は、各座標系の関係は次のようになります。

WINDOW  $(Wx1, Wy1) - (Wx2, Wy2)$

VIEW  $(Sx1, Sy1) - (Sx2, Sy2)$

WINDOW 命令と VIEW 命令を両方使った場合



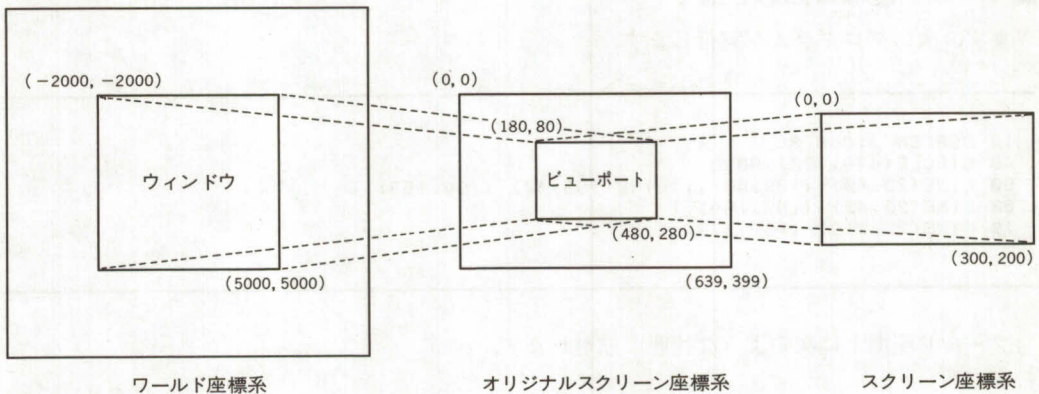
### ■ 3つの座標系の関係

オリジナルスクリーン座標系、スクリーン座標系、ワールド座標系の3つの座標系とウィンドウ、ビューポートの関係を見てみましょう。次のような命令を実行した場合にはこうなります。

SCREEN 3

WINDOW (-2000, -2000)-(5000, 5000)

VIEW (180, 80)-(480, 280)



ワールド座標系の中での  $(-2000, -2000) - (5000, 5000)$  をウィンドウとして取り出します。オリジナルスクリーン座標系の  $(180, 80) - (480, 280)$  をビューポートとします。これでウィンドウ内の図形がビューポート内に表示されます。

### ● 3つの座標系と BASIC のグラフィック関連の命令で使う座標

グラフィックに関係する命令の大部分は座標の指定が必要です。次の表はそれぞれの命令がどの座標系を使うのかということを示しています。

| 座標系                                  | グラフィック関連の命令・関数  |
|--------------------------------------|---|
| ワールド座標系<br>オリジナルスクリーン座標系<br>スクリーン座標系 | CIRCLE, DRAW, LINE, PAINT, POINT, PRESET, PSET, WINDOW<br>VIEW<br>GET@, PUT@, POINT(関数) |

各命令についての詳細は『BASICリファレンスマニュアル』を参照してください。

## 4.2 WINDOW 命令と VIEW 命令の例

ワールド座標に描いた図形は、ウィンドウやビューポートの設定によってディスプレイ画面内の位置や大きさが変わってきます。

ここでは実際にワールド座標に図形を描いた上で WINDOW 命令と VIEW 命令を使って画面の変化を見てみます。

### ■ワールド座標に図形を描く

まず、次のプログラムを実行します。

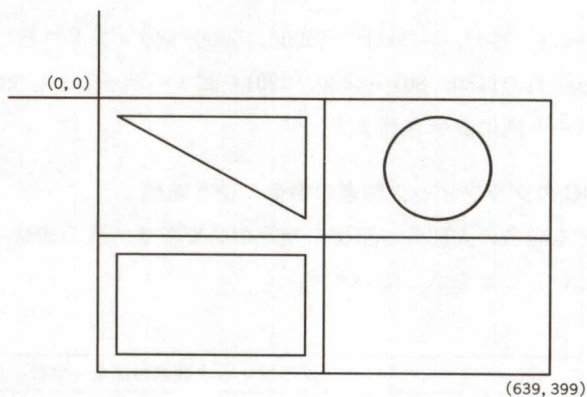
---

```

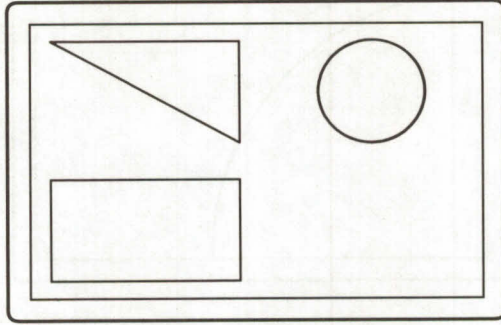
10 SCREEN 3:CLS 3
40 CIRCLE(479,100),90,6
50 LINE(20,40)-(199,40),1:LINE(199,40)-(199,159),1
60 LINE(20,40)-(199,159),1
70 LINE(20,219)-(199,379),2,B
    
```

---

ワールド座標上に次のような状態で描かれます。



ディスプレイ画面でもこのように見えます。初期状態ではウィンドウはワールド座標系の(0, 0) - (639, 399)であり、スクリーン座標系の(0, 0) - (639, 399)に対応しています。



### ■ WINDOW 命令を使ったウィンドウの設定

画面の右上の円の描かれている部分だけを表示するために、始めのプログラムの前に、WINDOW 命令を入れてみます。

---

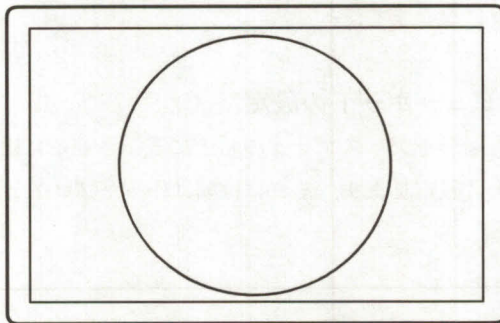
```

10 SCREEN 3:CLS 3
20 WINDOW(319,0)-(639,199)
40 CIRCLE(479,100),90,6
50 LINE(20,40)-(199,40),1:LINE(199,40)-(199,159),1
60 LINE(20,40)-(199,159),1
70 LINE(20,219)-(199,379),2,B

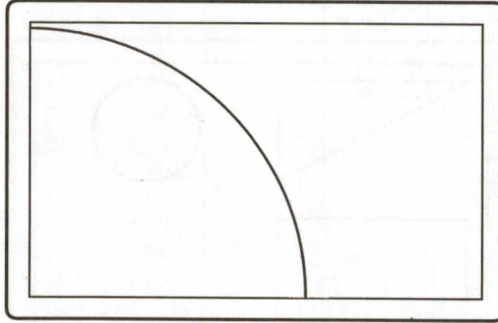
```

---

このプログラムを実行すると画面は次のようになります。

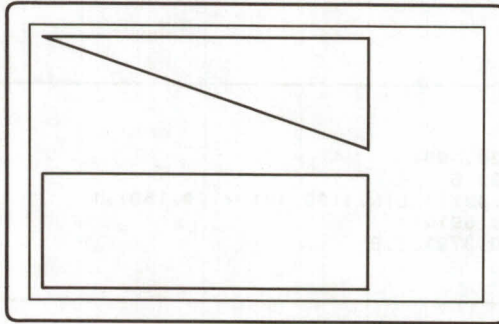


次に 20 行目の WINDOW 命令を “20 WINDOW (479, 0) - (639, 100)” に変更してみます。



次の三角形と四角形の部分を画面全体に表示してみます。

20 行目の WINDOW 命令を “20 WINDOW (0, 0) - (319, 399)” に変えて実行します。



このように、WINDOW 命令を使うとワールド座標に描かれた図形を任意に拡大、縮小したり移動させたりできるようになります。

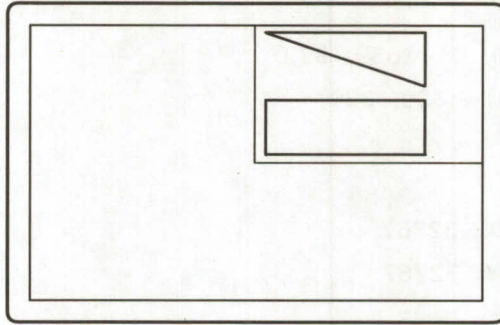
### ■ VIEW 命令を使ったビューポートの設定

ワールド座標に描かれた図形をディスプレイの右上に表示させるには、VIEW 命令を使ってビューポートを画面の右上に設定します。それには前に作ったプログラムに VIEW 命令を追加します。

---

```
10 SCREEN 3:CLS 3
20 WINDOW(0,0)-(319,399)
30 VIEW(319,0)-(639,199),,4
40 CIRCLE(479,100),90,6
50 LINE(20,40)-(199,40),1:LINE(199,40)-(199,159),1
60 LINE(20,40)-(199,159),1
70 LINE(20,219)-(199,379),2,B
```

---



### ■ワールド座標値の制限

CIRCLE 命令や LINE 命令などのグラフィック命令に使用するワールド座標の座標値には、次のような最小値と最大値の制限があります。最小値以下あるいは最大値以上の値を指定するとオーバーフローのエラーが起こります。

この最小値と最大値はいつも一定ではありません。WINDOW 命令と VIEW 命令で指定した座標値によって変化します。

WINDOW (a1, b1)–(a2, b2)

VIEW (x1, y1)–(x2, y2)

各座標にこのような値を指定して WINDOW 命令と VIEW 命令を実行した場合には、各種グラフィック命令で使用できるワールド座標値の範囲は、次のような式で表すことができます。

WX はワールド座標系の X 座標の値を、WY ワールド座標系の Y 座標の値を表します。

WX

$$(-32768-x_1) \frac{a_2-a_1}{x_2-x_1} + a_1 \leq WX \leq (32767-x_1) \frac{a_2-a_1}{x_2-x_1} + a_1$$

WY

$$(-32768-y_1) \frac{b_2-b_1}{y_2-y_1} + b_1 \leq WY \leq (32767-y_1) \frac{b_2-b_1}{y_2-y_1} + b_1$$

たとえば、

WINDOW (0, 0)-(639, 399)

VIEW (0, 0)-(639, 399)

の場合を計算してみると、

$-32768 \leq WX \leq 32767$

$-32768 \leq WY \leq 32767$

となります。

**注意：**VIEW 命令を使う前には、必ず WINDOW 命令を使ってウィンドウを設定してください。

VIEW 命令だけを単独で使うとウィンドウが変化してしまいます。

また、SCREEN 命令は WINDOW 命令と VIEW 命令の前に実行してください。

SCREEN 命令を実行すると、ビューポートとウィンドウの設定は初期化されます。

## 4.3 グラフィック画面の色

### ■グラフィック画面の色指定

グラフィック画面ではバックグラウンドカラー(背景色)、ボーダーカラー(周辺色)、フォアグラウンドカラー(前景色)を指定することができます。

グラフィック画面の色指定には [1] COLOR 命令を使います。

#### [1]COLOR 命令

COLOR [〈ファンクションコード〉][, 〈バックグラウンドカラー〉][, 〈ボーダーカラー〉][, 〈フォアグラウンドカラー〉][, 〈パレットモード〉]

この中でファンクションコードはテキスト画面の文字にいろいろな機能を与えるものです。

#### ▶参照 ファンクションコード→3.1の「テキスト画面の文字の色と表示モード」

〈ボーダーカラー〉はバックグラウンドの周辺の色を表します。BASIC 起動直後は黒になっています。

| 指定値 | 色  |
|-----|----|
| 0   | 黒  |
| 1   | 青  |
| 2   | 赤  |
| 3   | 紫  |
| 4   | 緑  |
| 5   | 水色 |
| 6   | 黄色 |
| 7   | 白  |

ただし、〈ボーダーカラー〉は、400ラインの高解像度ディスプレイの使用時には意味がありません。

〈バックグラウンドカラー〉はグラフィック画面の地の色を表します。

〈フォアグラウンドカラー〉はグラフィック画面に点や線を表示したりするときに使われる色を表します。CIRCLE、LINE、PSET 命令などの描画命令で色の指定を省略するとこの色が採用されます。

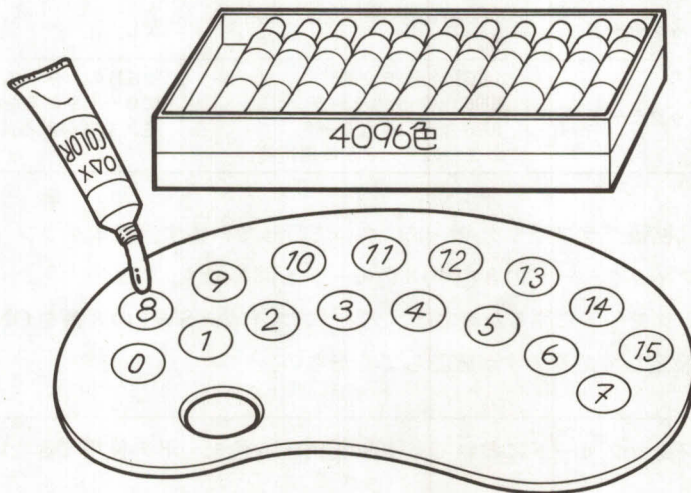
なお、〈バックグラウンドカラー〉と〈フォアグラウンドカラー〉の指定は色番号で指定するわけではありません。後で説明する“パレット番号”で指定します。

パレット番号、〈パレットモード〉については次の項を参照してください。

### ■パレットモード

N<sub>88</sub>-BASIC(86)でカラーのモード(カラーモード、高分解能カラーモード)を選択すると8色あるいは16色の色を一度に使用することができます。

各命令の色指定の際には固定した色を指定するのではなく、パレットと呼ばれる“色の入れ物”の番号を指定します。パレット番号にはユーザーが自由に色を割り当てることができます。各命令で表示色にパレット番号を指定するとそのパレット番号に割り当てられている色が使用されるというわけです。これをパレット機能といいます。パレット機能の利点は、いったん描いてしまった図形などでも、後からパレット番号に割り当てられた色(カラーコード)を変更するだけで図形の色を変えることが可能な点です。



N<sub>88</sub>-BASIC(86)には次のような3つのパレットモードがあり、使用できる色の数が異なります。

**8色中・8色モード**

黒, 明るい青, 明るい赤, 明るい紫, 明るい緑, 明るい水色, 明るい黄, 白の8色を使用することができます。

**4096色中・8色モード**

4096色の中から8色を使用することができます。

**4096色中・16色モード**

4096色の中から16色を使用することができます。

この3つのパレットモードのどれを選択するかは [1] COLOR 命令の5番目のパラメータで設定します。BASIC起動直後は8色中・8色モードになっています。

| 指定値 | パレットモード       |
|-----|---------------|
| 0   | 8色中・8色モード     |
| 1   | 4096色中・8色モード  |
| 2   | 4096色中・16色モード |

**■基本グラフィックモードと拡張グラフィックモード**

N<sub>88</sub>-BASIC(86)のグラフィックには2つのグラフィックモードがあります。これは主にパレットモードの選択に関わります。

|             | グラフィック機能の差  | 利用者メモリの圧迫                               |
|-------------|---|---|
| 基本グラフィックモード | 8色中・8色モードだけを使用できる。  | なし                                      |
| 拡張グラフィックモード | 8色中・8色モード<br>4096色中・8色モード<br>4096色中・16色モード<br>の3つのモードが使用可能。 | 22KBのシステムコードがメモリにロードされるため、その分利用者メモリが減る。 |

この章の解説は拡張グラフィックモードについて行っています。基本グラフィックモードの機能は拡張グラフィックモードの8色中・8色モードと同等です。

拡張グラフィックモードにするためには、ディップスイッチSW1の8番をONにします。出荷時にはONに設定してありますが確認してください。

---

**注意：**拡張グラフィックモードにおいても4096色中・16色モードの使用できない機種もあります。

---

▶ 参照 ディップスイッチ→『ガイドブック』

## ■パレットの色の変更

パレットの色の変更は(2)COLOR 命令で行います。

### (2)COLOR 命令

COLOR [(`<パレット番号>`), (`<カラーコード>`)]

カラーコードとは各色につけられたコード番号です。この命令ではパレット番号とカラーコードを対応づけることによって、どのパレットにどの色を対応させるか任意に決めることができます。その際、同じカラーコードの色を複数のパレット番号に対応づけてもかまいません。

各モードにおけるパレット番号とカラーコードの関係は次のとおりです。

| パレット番号 | カラーコード        |                  |                   |
|--------|---------------|------------------|-------------------|
|        | 8色中・<br>8色モード | 4096色中・<br>8色モード | 4096色中・<br>16色モード |
| 0      | 0 (黒)         | &H000            | &H000             |
| 1      | 1 (明るい青)      | )                | )                 |
| 2      | 2 (明るい赤)      |                  |                   |
| 3      | 3 (明るい紫)      |                  |                   |
| 4      | 4 (明るい緑)      |                  |                   |
| 5      | 5 (明るい水色)     |                  |                   |
| 6      | 6 (明るい黄)      |                  |                   |
| 7      | 7 (白)         |                  |                   |
| 8      | /             | /                | )                 |
| 9      |               |                  |                   |
| 10     |               |                  |                   |
| 11     |               |                  |                   |
| 12     |               |                  |                   |
| 13     |               |                  |                   |
| 14     |               |                  |                   |
| 15     |               |                  |                   |

### 8色中・8色モード

0~7のパレット番号に8個のカラーコードを任意に指定します。

### 4096色・8色モード

0~7のパレット番号に&H000~&HFFFの4096個のカラーコードから任意に指定します。


### 4096色・16色モード

0~15のパレット番号に&H000~&HFFFの4096個のカラーコードから任意に指定します。

たとえば、4096色中・16色モードで

COLOR=(0, &H777)

とすると、パレット番号0が灰色となります。

パレットを BASIC が起動した時の初期状態にもどすには、"COLOR  "とだけ入力します。この方法は DISK モード BASIC でのみ使用できます。

初期状態のパレット番号とカラーコードの関係は次のとおりです。

| パレット番号 | カラーコード    |              |               | 色      |
|--------|-----------|--------------|---------------|--------|
|        | 8色中・8色モード | 4096色中・8色モード | 4096色中・16色モード |        |
| 0      | 0         | &H000        | &H000         | 黒      |
| 1      | 1         | &H00F        | &H00F         | 明るい青   |
| 2      | 2         | &H0F0        | &H0F0         | 明るい赤   |
| 3      | 3         | &H0FF        | &H0FF         | 明るい紫   |
| 4      | 4         | &HF00        | &HF00         | 明るい緑   |
| 5      | 5         | &HFF0        | &HFF0         | 明るい水色  |
| 6      | 6         | &HFF0        | &HFF0         | 明るい黄   |
| 7      | 7         | &HFFF        | &HFFF         | 白      |
| 8      |           |              | &H777         | 灰色     |
| 9      |           |              | &H00A         | 少し暗い青  |
| 10     |           |              | &H0A0         | 少し暗い赤  |
| 11     |           |              | &H0AA         | 少し暗い紫  |
| 12     |           |              | &HA00         | 少し暗い緑  |
| 13     |           |              | &HA0A         | 少し暗い水色 |
| 14     |           |              | &HAA0         | 少し暗い黄  |
| 15     |           |              | &HAAA         | 少し暗い白  |

▶ 参照 カラーコードと色の関係→4.3の「色の仕組み(カラーコードの指定方法)」

注意：3つのパレットモードによって各画面モードでの使用可能なページ数も変わってきます。ページ数に関しては3.2の「ページ」を参考にしてください。なお、4096色中・8色モードと4096色中・16色モードはアナログRGB対応ディスプレイに対応したモードです。このモードでアナログRGBディスプレイ以外のディスプレイを使うこともできますが、その場合には指定どおりの色を出すことはできません。

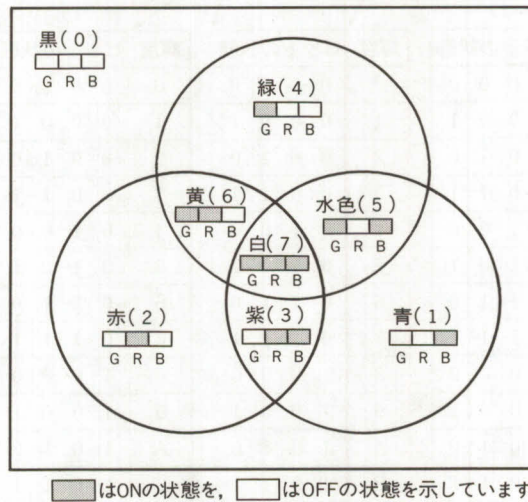
### ■色の仕組み(カラーコードの指定方法)

グラフィック画面に表示できる色はすべて基本色である緑(G)、赤(R)、青(B)の組み合わせで表現されています。

#### ●8色中・8色モードの場合

緑(G)、赤(R)、青(B)のそれぞれがONまたはOFFの2とおりを選択できます。したがって、 $2 \times 2 \times 2 = 8$ 色が表現できます。

R, G, B各色のON/OFFとカラーコードの関係は次のようになります。



#### ●4096色中・8色モードと4096色中・16色モードの場合

緑(G)、赤(R)、青(B)のそれぞれを16段階の輝度(光の強さ)で表現することができます。したがって、 $16 \times 16 \times 16 = 4096$ 色が使用可能です。

R, G, B各色の輝度は4ビットの情報を持っています。これを1桁の16進数にすると, R, G, B各色の輝度は0~Fまでの16段階の数値で表すことができます。

カラーコードは16進表記で3桁の数値で表されていますが, R, G, B各色の輝度がこの数値に対応しているわけです。なお, カラーコードの順番はG, R, Bの順になっています。

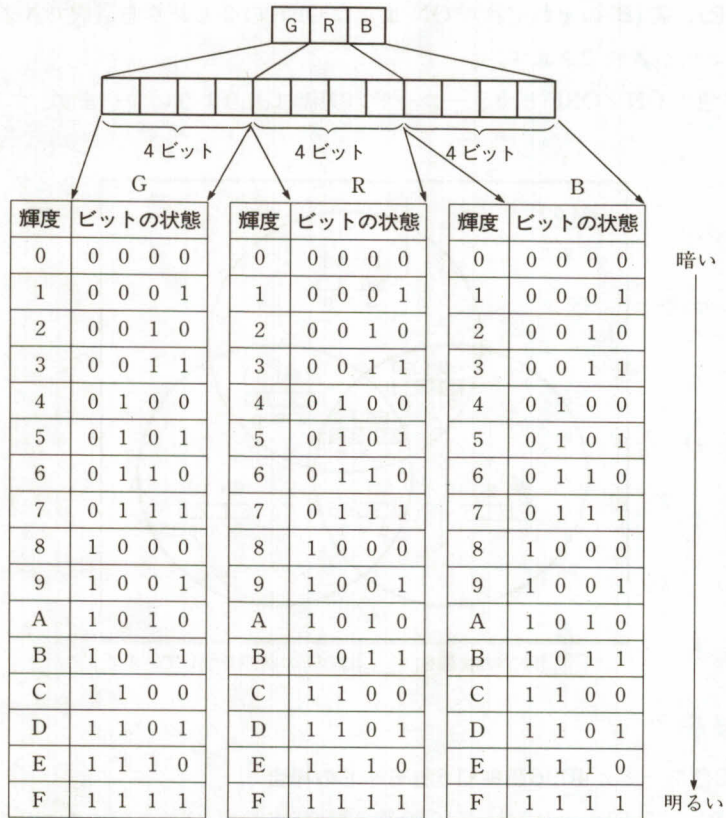
| R, G, Bの輝度 |
|------------|
| Gの輝度が&H9   |
| Rの輝度が&HF   |
| Bの輝度が&H0   |



| カラーコード | 色     |
|--------|-------|
| &H9F0  | オレンジ色 |

たとえば、Gが&H9(暗い緑)でRが&HF(最も明るい赤)の場合にはオレンジ色になります。このような方法で4096色をカラーコードで表現しているのです。

R, G, B各色の輝度とカラーコードの関係は次のようになります。



輝度0は各色が表示されないことを表す。

### ■タイリング

BASICの描画命令を使用して任意の範囲をぬりつぶすことができます。この場合、パレット番号を使用して単色でぬりつぶすのが一般的ですが、複数の色を使ったモザイク模様(タイルパターン)でぬりつぶすことも可能です。これを「タイリング」と呼びます。

白黒モードや8色中・8色モードで中間色を表現するためのテクニックとして特に効果的です。また、4096色中・8色モードや4096色中・16色モードでも使用できます。

この機能は(2) PAINT命令やCIRCLE命令、LINE命令などで使うことができます。パレット番号を指定する代わりにタイルパターンを表す文字列を指定します。この文字列を「タイリング文字列」といいます。

タイルパターンの大きさは横方向は8ドット分と決められていますが、縦方向の長さはタイルストリングの文字列の長さで決まります。縦方向がnドットのタイルを作るためには、タイルストリングとして、白黒モードでn文字、8色中・8色モード、4096色中・8色モードで3×n文字、4096色中・16色モードで4×n文字の長さがそれぞれ必要です。

タイルパターンは、タイルストリングに指定する文字列のキャラクタコードを2進数で表現したもの(ビットパターン)によって決定されます。タイルストリングの指定方法は白黒モード、8色モード、16色モードによって異なります。

#### ●白黒モードの場合

タイルストリング中の各1文字(バイト)のコードがタイルの横8ドットに対応します。文字コードの2進数表現で、タイルの横8ドットのうち、1に対応するドットはセット(白)、0に対応するドットはリセット(黒)として表されます。〈タイルストリング〉がn文字の長さであれば、その文字列の表す模様は、このようにして決定される横8ドットのパターンを縦にn列分だけ並べたパターンになります。

例) 白黒モードの場合

`CHR$(&HAA)+CHR$(&H55)`

|       |                 |
|-------|-----------------|
| ドット   | 1 2 3 4 5 6 7 8 |
| &HAA  | 1 0 1 0 1 0 1 0 |
| 対応する色 | 白黒白黒白黒白黒        |
| &H55  | 0 1 0 1 0 1 0 1 |
| 対応する色 | 黒白黒白黒白黒白        |

この例では、タイルストリングが2文字ですから横8ドットのタイルパターンが2列分設定されます。この2列のタイルパターンは1であるビットと0であるビットが互いに逆になっています。このパターンを基本タイルとして指定された領域を埋めれば、細かい市松模様になりますが、スクリーンのドットがたいへん細かいため、灰色のように見えます。

●カラーモードの場合

白黒モードと同様に、タイルパターンはタイルストリングに対応するドットパターンによって決定されます。白黒モードと異なる点は、8色中・8色モード、4096色中・8色モードの場合は3文字分ごとに、また4096色中・16色モードの場合は4文字分ごとに、タイルの横8ドット(1列分)の色がパレット番号として設定されることです。タイルパターンの例をみてみましょう。

例) 8色中・8色モード、4096色中・8色モードの場合

CHR\$(&HAA)+CHR\$(&H55)+CHR\$(&HFF)+CHR\$(&H55)+CHR\$(&HAA)+CHR\$(&HFF)

| ドット            | 1 2 3 4 5 6 7 8 |                     |
|----------------|-----------------|---------------------|
| &HAA           | 1 0 1 0 1 0 1 0 | ← 2 <sup>0</sup> の桁 |
| &H55           | 0 1 0 1 0 1 0 1 | ← 2 <sup>1</sup> の桁 |
| &HFF           | 1 1 1 1 1 1 1 1 | ← 2 <sup>2</sup> の桁 |
| 対応する<br>パレット番号 | 5 6 5 6 5 6 5 6 |                     |
| &H55           | 0 1 0 1 0 1 0 1 | ← 2 <sup>0</sup> の桁 |
| &HAA           | 1 0 1 0 1 0 1 0 | ← 2 <sup>1</sup> の桁 |
| &HFF           | 1 1 1 1 1 1 1 1 | ← 2 <sup>2</sup> の桁 |
| 対応する<br>パレット番号 | 6 5 6 5 6 5 6 5 |                     |

各ドットについての3ビット分のデータがパレット番号に読み換えられます。この場合、指定するタイルストリングのうちの最初の文字が最下位ビット(2<sup>0</sup>)の桁を表し、以後、2<sup>1</sup>の桁、2<sup>2</sup>の桁となります。

この例ではタイルストリングが6文字ですから横8ドットのタイルパターン2列分が設定されていることとなります。この2列のタイルパターンは両方ともパレット番号5とパレット番号6が交互に横8ドット並びますが、色の順序が1ドットずれているのでパレットが初期状態であれば水色と黄の市松模様になります。実際にはドットが細かいので淡い黄緑に見えます。

例) 4096色中・16色モードの場合

CHR\$(&HCE)+CHR\$(&H8C)+CHR\$(&HD0)+CHR\$(&H22)

| ドット            | 1 2 3 4 5 6 7 8 |                     |
|----------------|-----------------|---------------------|
| &HCE           | 1 1 0 0 1 1 1 0 | ← 2 <sup>0</sup> の桁 |
| &H8C           | 1 0 0 0 1 1 0 0 | ← 2 <sup>1</sup> の桁 |
| &HD0           | 1 1 0 1 0 0 0 0 | ← 2 <sup>2</sup> の桁 |
| &H22           | 0 0 1 0 0 0 1 0 | ← 2 <sup>3</sup> の桁 |
| 対応する<br>パレット番号 | 7 5 8 4 3 3 9 0 |                     |

この例ではタイルSTRINGが4文字ですから横8ドットのタイルパターンが1列分設定されているわけです。各ドットについての4ビット分のデータがパレット番号に読み換えられることにより、このような結果となります。

パレットが初期状態の場合には、横8ドットが白、明るい水色、灰色、明るい緑、明るい紫、明るい紫、少し暗い青、黒と並ぶタイルパターンになります。

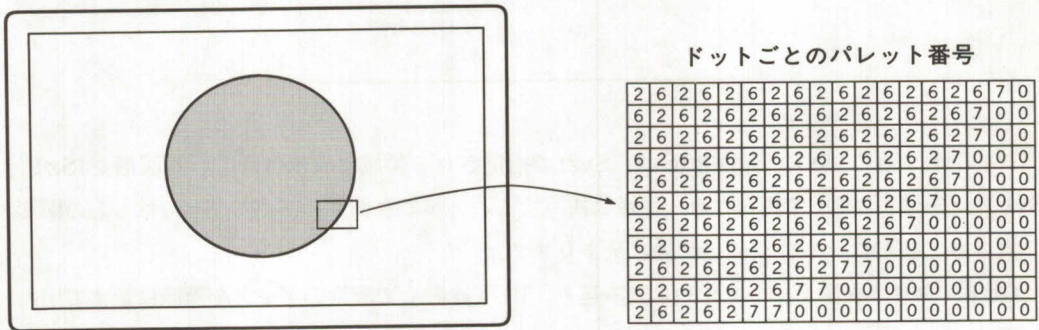
実際に8色中・8色モードでタイリングを使った描画例を示します。

```

10 SCREEN 3
20 CLS 3
30 TILE1$=CHR$(&H0)+CHR$(&HFF)+CHR$(&H55)
40 TILE2$=CHR$(&H0)+CHR$(&HFF)+CHR$(&HAA)
50 CIRCLE(320,200),100,7
60 PAINT(320,200),TILE1$,TILE2$,7

```

TILE1\$がタイルパターンの1列目をTILE2\$が2列目を表しています。この2列のタイルパターンは両方ともパレット番号2とパレット番号6が交互に横8ドット並びますが、色の順序が1ドットずれているのでパレットが初期状態であれば赤と黄の市松模様になります。実際にはドットが細かいので白で描かれた円の中はオレンジ色に見えます。



このように、タイリングを行うとタイルSTRINGで設定されたタイルパターンを単位として、領域がぬりつぶされます。

**注意：**カラーモードの場合、〈タイルSTRING〉の文字数が3(4096色中・16色モードの場合には4)の倍数でないときには、文字列の後ろから3(あるいは4)で割った余りの数だけ無視されます。また、3文字(あるいは4文字)に満たない場合には“Illegal function call”エラーとなります。

## 4.4 図形データの転送

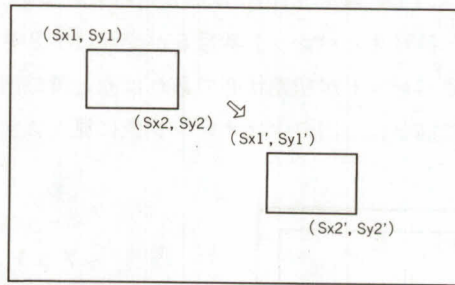
N<sub>88</sub>-BASIC(86)では画面上のある大きさの領域をそっくりそのまま他の場所へ複写させることができます。これを画像の転送と呼びます。これを行うためにはGET@命令とPUT@命令を組み合わせ使用します。

PUT@命令とGET@命令

GET[@] (Sx1, Sy1)–(Sx2, Sy2), <配列変数名>[[<添字>]]

PUT[@] (Sx1, Sy1), <配列変数名>[[<添字>]][, <条件>][, <フォアグラウンドカラー>, <バックグラウンドカラー>]

▶参照 GET@, PUT@→『BASICリファレンスマニュアル』



この例のように(Sx1, Sy1)と(Sx2, Sy2)で指定される領域に表示されている図形を(Sx1', Sy1')と(Sx2', Sy2')で指定される領域に複写することができます。(Sx2', Sy2')はもとの図形の大きさと同じになるように自動的に決まります。

GET@命令は画像データを配列に読み込み、PUT@命令で配列のデータを画像に変えて出力します。

使用する配列変数はGET@命令を使う前にDIM命令を使って定義しておかなくてはなりません。確保すべき配列変数の大きさは、次のようにして求めることができます。

①まず、読み込みたい図形の縦横のドット数を数え、必要なバイト数を算出します。

必要なバイト数=((横のドット数+7)¥8)\*縦のドット数\*M+4

ただし、白黒モードのとき…………… M=1

8色モードのとき…………… M=3

16色モードのとき…………… M=4

②次に、必要なバイト数から実際に確保する配列変数の大きさ(添字の値)を決めます。

$$\text{添字の値} = (\text{必要なバイト数} + (N-1)) \div N - 1$$

ただし、N は配列変数の型によって変わります。

整数型配列…………… N=2

単精度型配列…………… N=4

倍精度型配列…………… N=8

③求めた添字の値をもとに、配列変数(例：A%)を宣言します。

**DIM A%(添字の値)**

なお、配列変数(ここではA%)にはOPTION BASE(添字の最小値)が0の1次元配列を使用するようにしてください。

PUT@命令の〈条件〉は転送先に表示する際のいろいろな条件を指定するものです。そのまま表示するにはPSETを使います。

また、GET@命令とPUT@命令を使う際にアクティブページを変えると別のページに転送することも可能です。

```

10 COLOR , , , 0
20 I = (((100+7) \ 8) * 100 * 3 + 4 + 1) \ 2 - 1
30 DIM A%(I)
40 SCREEN 0 , , 0
50 CIRCLE (50, 50), 20, 3
60 GET@(0, 0) - (99, 99), A%
70 SCREEN , , 1
80 PUT@ (100, 100), A%, PSET
90 SCREEN , , 2

```

この場合には1ページの(0, 0)と(99, 99)で指定された範囲が2ページの(100, 100)を左上の頂点とする位置へ転送されます。



# 第5章

## 日本語処理

N<sub>88</sub>-日本語 BASIC(86)には日本語処理機能が用意されています。この機能を使用することにより、日本語文字列(2バイト系文字列)を英数カナ文字列(1バイト系文字列)同様にテキスト画面やグラフィック画面に表示したり、漢字プリンタで印刷することができます。

ここでは、このような BASIC の日本語処理機能について述べています。日本語入力や日本語変換の具体的な方法については、『日本語入力ガイド』に詳しく解説しています。

### 5.1 日本語文字とは

#### ■ 2バイトで扱われる日本語文字

コンピュータでは文字をコードとして扱います。

英数文字やカナ文字は1バイトの文字コードとして扱われています。1バイトでは最大で、 $2^8=256$ 種類の文字(制御コードを含む)だけしか表すことはできません。そこで、日本語文字を表す場合には、2バイトで1文字を表す方法がとられています。

1文字を2バイトで表すと文字の種類は最大 $2^{16}=65536$ 種類となりますので、何千種類の文字種のある日本語文字を表現することもできるわけです。

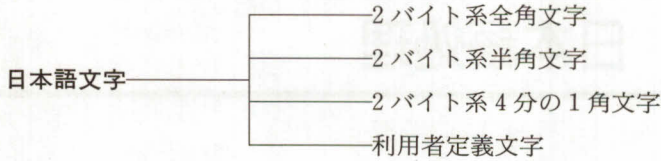
2バイトで表される文字には、漢字、ひらがな、カタカナ、英数文字、ギリシャ文字、ロシア文字、記号類、グラフィックキャラクタなどがあります。

N<sub>88</sub>-日本語 BASIC(86)では、これらの2バイト文字全体を総称して“日本語文字”，あるいは“2バイト系日本語文字”と表現します。

なお、N<sub>88</sub>-日本語 BASIC で使われている日本語文字のコードは JIS 漢字コードに準拠したものです。

## ■日本語文字の種類

BASIC で扱う日本語文字には次のようなものがあります。



### ●2バイト系全角文字

テキスト画面上に縦16ドット×横16ドット(最大)で出力される文字です。グラフィック画面にも表示できます。

### ●2バイト系半角文字

テキスト画面上に縦16ドット×横8ドット(全角文字の横半分の大きさ)で出力される文字です。文字の大きさは1バイト文字と同じですが、書体が異なります。

なお、2バイト系半角文字の一部(&H0080～&H009F, &H00E0～&H00FF)はテキスト画面には表示できません。

グラフィック画面にはすべての2バイト系半角文字を表示できます。

### ●2バイト系4分の1角文字

グラフィック画面上に縦8ドット×横8ドット(全角文字の縦横半分の大きさ)で出力される文字です。テキスト画面には表示できません。

### ●利用者定義文字

ユーザーが自由に定義することができる文字です。他の2バイト全角文字と同様の方法でテキスト画面やグラフィック画面に表示できます。ただし、プリンタへの出力はできません。

▶ 参照 2バイト系全角文字→『日本語入力ガイド』

2バイト系半角文字→「付録G 日本語コード表」

2バイト系4分の1角文字→「付録G 日本語コード表」

利用者定義文字→5.2の「利用者定義文字」

---

注意：2バイト系半角文字の形は1バイトの英数カナ文字に似ていますが、BASICは別の文字と解釈します。したがって2バイト系半角文字を命令や変数名に使用することはできません。

---

## ■日本語文字の使い方

### ●日本語文字が使える場合

BASIC で日本語文字が使えるのは次のような場合です。

- ・文字型定数として扱う場合
- ・INPUT 命令, KINPUT 命令などで文字型変数に文字列を代入する場合
- ・REM 命令中の注釈として使用する場合
- ・PUT@命令でグラフィック画面に表示させる場合

### 文字型定数として扱う場合

1バイト文字の場合と同様に、ダブルクォーテーション(”)で囲んで使います。1バイト文字と混在させて使用することもできます。また、文字列演算も可能です。

例) 画面表示, ファイルへの書き出し, 文字型変数への代入, 文字列演算など

```
PRINT "漢字"
PRINT # 1, "ABC 漢字"
A$ = "漢字"
C$ = "漢字" + "ひらがな"
```

---

注意: SCRN: (スクリーンファイル)への日本語の書き出しはできません。

---

INPUT 命令, KINPUT 命令などで文字型変数に文字列を代入する場合

例) キーボードから入力された日本語文字列を変数 A\$ に代入する。

```
KINPUT A$
```

REM 命令中の注釈として使用する場合

例) サブルーチンの注釈など

```
1000 ' 日本語表示ルーチン
```

PUT@命令で画面に表示させる場合

例) 「漢」という文字をグラフィック画面に表示させる。

```
PUT(100, 100), KANJI(&H3441)
```

●日本語文字が使用できない場合

- ・ 命令名、関数名、変数名、ファイル名、ラベル名などには日本語文字は使用できません。
- ・ KEY 命令で日本語文字列をファンクションキーに登録することはできません。
- ・ TAB 関数で日本語文字列を制御することはできません。
- ・ PRINT USING/LPRINT USING 命令で日本語文字を書式制御文字列に指定したり、日本語文字を含む文字列を編集したりすることはできません。

▶参照 命令・関数→『BASIC リファレンスマニュアル』

■日本語文字列の内部形式

●日本語シフトコード

日本語文字は2バイトで表されていますが、日本語文字中の上位1バイトと下位1バイトを取り出してみると、それぞれが1バイトコードであることにかわりありません。

たとえば、&H34, &H41 という2バイトのコードは1バイト系英数カナ文字2つであると考えれば、&H34 と &H41 はそれぞれ "4" と "A" ということになります。それに対して2バイトの日本語文字であると考えれば「漢」という文字になります。

例) 1バイト文字と2バイト文字の比較

|               | コード          | 表す文字   |
|---------------|--------------|--------|
| 1バイト系英数カナ文字   | &H34<br>&H41 | 4<br>A |
| 日本語文字(2バイト文字) | &H34+&H41    | 漢      |

したがって、1バイト系英数カナ文字と日本語文字を混合させて使用するには、1バイトコード2つを1バイト文字2つと見るか、2バイト文字と見るかを判別する必要があります。

このために使われているコードが日本語シフトコードです。

日本語文字の前にはKI(&H1B4B)コードを、日本語文字の後にはKO(&H1B48)コードを入れることにより、1バイト文字と2バイト文字の判別を行います。前者を漢字インコード(以下、KIコード)、後者を漢字アウトコード(以下、KOコード)とといいます。KIコードとKOコードには含まれたものは日本語、この外にあるデータは1バイト文字と解釈されるわけです。

なお、日本語シフトコードは日本語の1文字(2バイト)ごとにその前後に付ける必要はありません。連続した日本語文字列の前後にそれぞれ1つずつ付けばよいのです。

日本語シフトコードを使用することで、日本語文字列と1バイト文字を混在して使用することができます。

---

注意：日本語文字列の桁数を数えたり比較したりする場合には、前後に日本語シフトコードが存在していることに注意してください。

---

### ●日本語文字列の内部形式

日本語文字は2バイトで表されますので、1文字につき1バイト文字の2文字分のメモリを必要とします。また、日本語文字列の前後は日本語シフトコードで囲まれています。

例) “ABC 漢字 DE” の内部形式

|    |    |    |    |    |    |    |    |    |      |    |    |    |        |
|----|----|----|----|----|----|----|----|----|------|----|----|----|--------|
| A  | B  | C  | KI | 漢  | 字  | KO | D  | E  | 内部形式 |    |    |    |        |
| 41 | 42 | 43 | 1B | 4B | 34 | 41 | 3B | 7A | 1B   | 48 | 44 | 45 | 16進コード |

### ●日本語文字の入力と日本語シフトコード

KINPUT 命令あるいは **CTRL** + **XFER** キーの入力によってキーボードから日本語文字を入力すると、日本語文字の前に KI コードが、日本語文字の後ろに KO コードが自動的に挿入されます。したがって、ユーザーは KI コードと KO コードを意識して入れる必要はありません。

▶参照 KINPUT 命令, **CTRL** + **XFER** キー→5.3の「日本語文字の入力」

---

注意：BASIC で扱う文字型定数の大きさは最大 255 バイトですから、すべてが英数カナ文字の場合は 255 文字が最大となります。文字列が日本語文字を含む場合は、日本語文字が1文字2バイトであり、また前後に付加される KI コード、KO コードも各2バイトであることに注意してください。

---

## 5.2 日本語文字の使用準備と保守

BASIC 上で日本語文字を使用するには次のような準備が必要です。

- ・漢字 ROM の装備
- ・400 ラインの高解像度ディスプレイの接続
- ・日本語変換方式の選択
- ・日本語辞書ファイルの使用準備
- ・利用者定義文字格納ファイルの作成と更新

### ■漢字 ROM の装備

日本語文字を画面に出力させるためには、コンピュータ本体に漢字 ROM (漢字のフォントパターンが記憶されているメモリチップ) が実装されていなければなりません。漢字 ROM には、JIS 第一水準および第二水準の2種類があります。

PC-9800 シリーズはこれらの漢字 ROM を標準実装しています。ただし、次の機種を御使用の場合は別売の漢字 ROM が必要です。

PC-9801/E……第一水準、第二水準とも実装していない

F/M……第一水準のみ標準実装

▶参照 JIS 第一水準、第二水準→『日本語入力ガイド』

### ■ディスプレイ

日本語文字を画面表示するためには400ラインの高解像度のディスプレイが必要です。200ラインのディスプレイでは日本語文字を表示することはできません。

### ■日本語変換方式の選択

次の6種類の日本語変換方式(日本語文字を入力する際の変換の方法)の中から選択することができます。

- ・AI 逐次変換方式
- ・AI 連文節変換方式
- ・逐次変換方式
- ・連文節変換方式
- ・単文節変換方式
- ・JIS16 進コード変換方式

選択した変換方式はシステムディスクに属性として設定しておくことができます。一度設定を行うと、そのシステムディスクから DISK モード BASIC を起動するたびに選択した変換方式が採用されます。

変換方式の設定には、ユーティリティ “setup.n88” を使用します。

- ▶ 参照 setup.n88 → 「10.8 システムディスク属性の設定」  
日本語変換方式 → 『日本語入力ガイド』

注意：ROM モード BASIC では、JIS16 進コード変換方式のみが使用できます。

### ■辞書ファイルの使用準備と保守

キーボードには2バイトの文字を表すキーはありません。そのため、日本語を入力するには、まずカナまたは英文字で日本語の読みをタイプし、日本語変換機能を利用して日本語文字に変換します。このとき、BASIC はタイプされた文字の読みをもとに辞書ファイル(たとえば、KNJAI.DIC など)を参照して日本語文字に変換し、画面などに出力します。

このため、日本語変換を行う場合には辞書ファイルが必要です。それには次のような準備を行ってください。

- 辞書ファイルの選択
- 辞書ファイルのドライブ番号の設定
- 辞書ファイルの保守

注意：辞書ファイルを使わない場合には、JIS16 進コード変換方式のみが使用できます。

### ●辞書ファイルの選択

選択する日本語変換方式によって使用する辞書ファイルが異なります。変換方式と辞書ファイルの関係は次のようになります。

| 変換方式           | 辞書ファイル    |
|----------------|-----------|
| AI 逐次変換方式      | KNJAI.DIC |
| AI 連文節変換方式     | KNJAI.DIC |
| 逐次変換方式         | BUNSETSU  |
| 連文節変換方式        | BUNSETSU  |
| 単文節変換方式        | BUNSETSU  |
| JIS16 進コード変換方式 | なし        |

辞書ファイルは BASIC を起動させるディスク(フロッピーディスクまたは固定ディスク)に入れて使用することも、別のディスクに入れて使用することもできます。

ただし、"KNJAI.DIC"はフロッピィのシステムディスクに入れることができませんので、別のディスクに入れて使用してください。

▶ 参照 辞書ファイルの選択→『日本語入力ガイド』

辞書ファイルのコピー→「10.6 ファイル転送」

### ●辞書ファイルのドライブ番号の設定

辞書ファイルをどのディスク装置に入れて使うかが決まったら、辞書ファイルを入れておくドライブ番号をBASICシステムに属性として設定しておく必要があります。ドライブ番号を間違えてセットすると辞書ファイルの参照ができなくなり、日本語変換ができませんので注意してください。

辞書ファイルのドライブ番号の設定にはユーティリティ "setup.n88" を使用します。

▶ 参照 setup.n88 →「10.8 システムディスク属性の設定」

### ●辞書ファイルの保守

辞書ファイルには、通常日本語変換に必要な単語が登録されており、日本語変換機能を使って利用することができますが、辞書ファイルにない単語とその読みをユーザーが自由に登録することもできます。こうして登録された単語は一般の単語と同様に変換することができます。これを"利用者登録単語"といいます。

一度登録した利用者登録単語はあとで削除することも可能です。また、辞書ファイルの一覧を表示することもできます。

単語の登録、削除、表示はユーティリティ "dicmen.n88" を使って行います。

▶ 参照 dicmen.n88 →「10.15 辞書ファイルの保守」

### ■利用者定義文字

漢字ROMに記憶されている日本語のフォント(文字の形のデータ)とは別に、ユーザーが自由に文字を作成し使用することができます。これを利用者定義文字といいます。

利用者定義文字の定義は、2バイトの漢字コードと定義する文字の形を対応づけることによって行います。使用できる漢字コードは16進形式で&H7621~&H767E、および&H7721~&H777Eの範囲で、188個を定義できます。

利用者定義文字はJIS第一水準、第二水準の漢字と同様に画面に表示したり、ファイルに書き出ししたりすることができます。ただし、この場合ファイルに書き出されるのは、漢字コードですので、この文字をファイルから読み込んで表示させるには、読み込むシステムにこの文字が登録されていなければなりません。また、プリンタへの出力はできません。

---

注意：機種によっては利用者定義文字が使用できない機種、定義できる文字数が少ない機種もあります。

---

### ● KPLOAD 命令による利用者定義文字の登録

KPLOAD 命令を使用して利用者定義文字を定義することができます。KPLOAD 命令は、利用者定義文字をコンピュータ本体内の専用 RAM(メモリ)に登録します。

#### KPLOAD 命令

KPLOAD <漢字コード>, <整数型配列名>

文字を<漢字コード>に指定した2バイトの漢字コードで登録します。

文字パターンを整数型の配列変数に格納しておき、その配列変数名を<整数型配列名>に指定します。

ただし、KPLOAD 命令でシステムに登録した文字は電源を切るか、リセットすると初期化されて失われてしまいますので保存することはできません。

整数型配列変数への具体的なデータの格納のしかたについては、『BASIC リファレンスマニュアル』を参照してください。

▶参照 KPLOAD 命令→『BASIC リファレンスマニュアル』

### ●利用者定義文字の保存

KPLOAD 命令で利用者定義文字をシステムに登録しても電源を切るかリセットをすると登録していたデータは失われてしまいます。利用者定義文字を保存するには、ユーティリティ`mkfont.n88`を使用します。

`mkfont.n88`には、利用者定義文字を作成し利用者定義文字格納ファイル(usfont.n88)に保存する機能があります。

利用者定義文字格納ファイルをシステムディスク(BASICの起動が可能な固定ディスクも含む)に入れておくと、DISKモードBASICの起動時に利用者定義文字が自動的にシステムに登録されます。これで利用者定義文字が使用できるようになります。

なお、`mkfont.n88`を使って、利用者定義文字の更新、削除、表示をすることも可能です。

▶参照 mkfont.n88 →「10.13 利用者定義文字格納ファイルの作成・更新」

## 5.3 日本語文字の入力と出力

日本語文字を入力あるいは出力する方法について解説します。日本語文字の入力・出力の方法は1バイト文字とほぼ同様です。

### ■日本語文字の入力

キーボードからの入力、ファイルからの入力について説明します。

#### ●キーボードからの日本語文字の入力


画面上でプログラムを作成・編集している時に、AI 逐次変換方式、AI 連文節変換方式などの日本語変換機能を利用して、キーボードから日本語文字を入力することができます。

また、プログラムの実行中に INPUT 命令を使って入力待ちとなった場合にも、キーボードから日本語文字を入力をすることができます。

日本語文字を入力するには、まず、日本語入力モードへ入ります。日本語入力モードに移行するには2とおりの方法があります。

#### CTRL + XFER キーによる日本語入力モードへの移行


CTRL キーを押しながら XFER キーを押すと日本語入力モードとなります。

1バイト文字を入力するモードにもどるには、再び CTRL キーを押しながら XFER キーを押すか、STOP キーまたは  キーを押します。

#### KINPUT 命令による日本語入力モードへの移行

INPUT 命令の代わりに KINPUT 命令を実行すると自動的に日本語入力モードになります。

したがって、この場合には CTRL + XFER キーを押す必要はありません。そのままキーをタイプすると日本語の入力ができます。

日本語入力モードを解除するには、 キーか、STOP キーを押します。KINPUT 命令の場合、CTRL キーを押しながら XFER キーを押しても日本語入力モードは解除されません。

▶参照 日本語文字の入力→『日本語入力ガイド』

#### ●ファイルからの日本語文字の入力

1バイト文字の場合と同様に、INPUT #命令、INPUT \$関数などを使用して日本語文字を含むシーケンシャルファイルから日本語文字を入力することができます。また、日本語文字を含むランダムファイルから GET 命令を使用して日本語文字を入力することができます。ただし、バイト数をチェックする際には、日本語文字列の前後の KI コードと KO コードの存在に注意してください。

▶参照 シーケンシャルファイル→6.7の「シーケンシャルファイルの操作」  
ランダムファイル→6.7の「ランダムファイルの操作」

## ■日本語文字の出力

### ●テキスト画面への表示

日本語文字を画面に表示する方法は、1バイト文字と同様です。

日本語文字を文字型定数としてそのまま表示させる方法と、文字型変数に代入して表示させる方法があります。

例 1) 「日本語」という文字列を文字型定数として扱い、画面に表示する。

```
PRINT "日本語"
```

例 2) 「日本語」という文字列を文字型変数 A\$ に代入し、A\$ の内容を表示する。

```
A$="日本語"
PRINT A$
```

例 3) 「日本語」を表すキャラクタコードで画面に表示する。

```
KI$=CHR$(&H1B)+CHR$(&H4B)
S$=CHR$(&H46)+CHR$(&H7C)+CHR$(&H4B)+CHR$(&H5C)+CHR
$(&H38)+CHR$(&H6C)
KO$=CHR$(&H1B)+CHR$(&H48)
PRINT KI$+S$+KO$
```

---

注意：1文字の日本語文字が2行にまたがると表示が乱れます。また、日本語の左右どちらかに1バイトの文字を重ねるような表示あるいは入力をするとう日本語表示が乱れます。また、テキスト画面上の文字パターンは16×16となっており、画面が25行モードの場合、各行の文字と文字がくっついて表示されます。不都合な場合は20行モードで表示してください。

---

### ●グラフィック画面への表示

グラフィック画面へ表示させるにはPUT@命令を使います。この命令は日本語文字を1文字単位で出力します。

#### PUT@命令

```
PUT@[@] (Sx, Sy), KANJI(<漢字コード>)[, <条件>][, <フォアグラウンドカラー>]
[, <バックグラウンドカラー>]
```

スクリーン座標(Sx, Sy)にJIS第一水準、第二水準漢字および利用者定義文字を表示します。

例) 「漢」(JIS コードで&H3441)という文字を表示する。

```
PUT@ (100, 100), KANJI(&H3441)
```

▶参照 PUT@命令→『BASIC リファレンスマニュアル』

### ●プリンタへの出力

日本語文字列を漢字プリンタに出力すれば、日本語文字を印字することができます。

例 1) 「日本語」という文字をプリンタで印字する。

```
LPRINT "日本語"
```

例 2) プリンタファイルをオープンし、文字型変数 A\$ に代入した「日本語」という文字を印字する。

```
A$="日本語"  
OPEN "LPT:" FOR OUTPUT AS #1  
PRINT #1, A$
```

普通、プリンタは初期状態では1バイト文字を印字するモードとなっていますが、プリンタに KI コード(&H1B4B)が送られると日本語文字を横書きで印字するモードとなります。この状態ではデータは2バイトの漢字コードと解釈され、漢字コードに対応する2バイト文字が印字されます。

KO コード(&H1B48)を送った後は、1バイト文字を印字するモードにもどり、コンピュータから送られてきたデータは1バイト文字と解釈されて印字されます。

上の2つの例では「日本語」という文字列の前後に KI コードと KO コードが入っていますので、「日本語」と印字されます。

プリンタのなかには、縦書き、倍角文字の印字や、一定のレイアウトによる印字などが可能なものもあります。プリンタの制御方法はプリンタによって違いがあります。詳しくはプリンタのマニュアルを参照してください。

---

注意：日本語文字を印字させるには PC-9800 シリーズコンピュータに対応したプリンタを使用してください。それ以外のプリンタでは正常に印字できない場合があります。また、印字される文字の書体はプリンタの種類によって異なります。

---

### ●ファイルへの出力

1バイト文字の場合と同様に PRINT #命令, WRITE #命令などを使用して日本語文字を含むデータをシーケンシャルファイルに出力することができます。また、日本語文字を含むデータを PUT 命令を使用して、ランダムファイルに出力することができます。ただし、プログラム中でバイト数をチェックする際には、日本語文字列の前後の KI コードと KO コードの存在に注意してください。

注意：SCRN：(スクリーンファイル)に日本語文字を出力することはできません。

## 5.4 日本語文字列操作関数

1バイト文字を扱う一般の文字列操作関数を使って日本語文字列を操作することができますが、とくに日本語文字列を操作するのに便利な専用の関数が用意されています。

これらの関数を日本語文字列操作関数と呼びます。これを使用することによって、たとえば、日本語を含む文字列の中から文字列の一部を取り出したり、1バイト系の英数カナ文字を日本語に変換したりといったことができます。

### ■日本語文字あるいは1バイト文字の取り出し

#### KEXT \$ 関数

KEXT \$( <文字列>, <機能> )

日本語文字を含む<文字列>の中から、<機能>に0を指定した場合は1バイト文字だけを、<機能>に1を指定した場合には2バイト系日本語文字だけを抜き出します。抜き出された2バイト系日本語文字にはKI/KOコードは付加されません。

例) 文字列から取り出した日本語文字をグラフィック画面に表示する。

|                                  |                                |
|----------------------------------|--------------------------------|
| 10 A\$=KEXT\$( "1234漢5768" , 1 ) | ——「漢」を取り出す。(KI/KOコードはなし)       |
| 20 B=ASC(MID\$(A\$,1,1))         | ——「漢」の漢字コードの1バイト目のキャラクタコードを得る。 |
| 30 C=ASC(MID\$(A\$,2,1))         | ——「漢」の漢字コードの2バイト目のキャラクタコードを得る。 |
| 40 D=256*B+C                     | ——漢字コードを計算する。                  |
| 50 PUT@(10,10),KANJI(D)          | ——グラフィック画面に表示する。               |

注意：日本語文字列操作関数で2バイト系半角文字として扱うのは付録Gの「文字列として扱う(テキスト画面に表示できる)2バイト系半角文字一覧表」に記載されている文字のみです。

### KMID \$ 関数

KMID \$( <文字列>, <式 1>[, <式 2>])

日本語文字を含む<文字列>の中の<式 1>番目の文字から始まる<式 2>の長さの文字列を抜き出します。

例) A \$ に「日本電気」という文字列を抜き出す。

A \$ = KMID \$( "日本電気 COMPUTER", 1, 6)

### ■日本語文字と 1 バイト文字の相互変換

#### AKCNV \$ 関数

AKCNV \$( <文字列>)

<文字列>中の 1 バイト系英数カナ文字を、対応する 2 バイト系全角文字に変換します。

例) A \$ に「日本電気COMPUTER」という 2 バイト系文字列を代入する。

A \$ = AKCNV \$( "日本電気 COMPUTER")

#### KACNV \$ 関数

KACNV \$( <文字列>)

<文字列>中の 2 バイト系全角文字を、対応する 1 バイト系英数カナ文字に変換します。

例) A \$ に「コンピュータ」という 1 バイト系文字列を代入する。

A \$ = KACNV \$( "コンピュータ")

### ■日本語文字列と漢字コードの相互変換

#### JIS \$ 関数

JIS \$( <文字列>)

<文字列>の最初の 2 バイトの日本語文字の漢字コードを得ます。

例) A \$ に「漢」という文字の漢字コード (16 進 4 桁の文字列) を代入する。

A \$ = JIS \$( KMID \$( "漢", 2, 1))

日本語文字列の先頭には KI コードが入っているので、「漢」の漢字コードを得るにはこの例のように KMID \$ などを併用する必要があります。

**KNJ\$ 関数**

KNJ\$(**<文字列>**)

**<文字列>**に指定した漢字コード(16進4桁の文字列)を日本語文字に変換します。

例) &H3441の漢字コードの文字(「漢」)を画面表示する。

```
PRINT KNJ$("1B4B")+KNJ$("3441")+KNJ$("1B48")
```

**■その他の関数****KINSTR 関数**

KINSTR(**[<位置>]**, **<文字列 1>**, **<文字列 2>**)

日本語文字を含む**<文字列 1>**の中から**<文字列 2>**を捜してその文字の位置を得ます。

例) B\$中に「日本語」という文字があれば、Aにその位置(何文字目か)が代入される。

```
A=KINSTR("日本語", B$)
```

**KLEN 関数**

KLEN(**<文字列>**[, **<機能>**])

日本語文字を含む**<文字列>**の中から、1バイト文字、あるいは2バイト系日本語文字などの特定のタイプだけの文字数を得ます。**<機能>**に指定する値によって対象となる文字のタイプが変わります。

例) AにB\$中の2バイト系日本語文字(KI/KOコードは含まない)の文字数を代入する。

```
A=KLEN(B$, 2)
```

**KTYPE 関数**

KTYPE(**<文字列>**, **<式>**)

日本語文字を含む**<文字列>**の中から、**<式>**番目の文字のタイプを得ます。

例) Aに文字列中の5番目の文字のタイプを代入する。

```
A=KTYPE("1234 漢 678", 5)
```

## 5.5 日本語処理機能を使ったプログラム例

ここで紹介する住所録プログラムは、住所録の作成、更新、住所録表示、住所録印刷、宛て名印刷などを行うことができます。記憶することができるデータは999件までです。

プログラムでは画面、キーボード、ディスクファイル、プリンタなどの各デバイス(入出力装置)に対して、頻繁に日本語文字列の入出力を行っています。また、日本語文字列操作関数、およびKINPUT 命令などの日本語処理に特有な命令を使用していますので、プログラムを作成する際の参考にしてください。

以下、プログラムの使い方と内容の解説を行います。

### ■住所録プログラムの使い方と解説

プログラムを実行させるとメニューが表示されます。 、キーで行いたい作業を選び、キーを押すと、その作業が行われます。

#### ● “データ入力” を選択した場合

このプログラムは、新規の登録とすでにあるデータの変更を1つの流れの中で処理しています。

ユーザーが文字列を入力するとその文字列を含む名前で登録されているデータを検索します。このデータがある場合には “データの変更を行う” と解釈して作業を進めます。いっぽう、該当する名前がない場合には、 “新規の登録を行う” と解釈し、新規の入力を求めます。

メニューで “データ入力” を選択すると、まず、

名前を検索する文字列を入力してください

と表示されるので、住所録に登録したい人の名前を検索する文字列を入力します。

該当する名前がディスクファイルにすでに登録されているかどうか検索します。

#### ・該当する名前が見つかった場合

(1) 入力した文字列を含む名前で登録されているデータが表示されます。

たとえば、 “鈴木” という文字列を入力したときには、ディスクファイル中に “鈴木太郎”、 “鈴木商店”、 “株式会社鈴木” などの名前で登録されているデータの登録番号、郵便番号、住所、名前が1件ずつ表示されます。

(2) 次に、

これを変更しますか(Y/N)?

と聞いてきます。

ここで、Yを入力すると(4)へ進みます。

Nを入力すると、ディスク内に該当する名前の他のデータがあるかどうかを捜し、あればこれを表示し再び(2)にもどります。

(3) 他にデータがない場合には、

その名前は登録されていません。

<〇〇>さんを新規に登録しますか(Y/N)?


と表示されます。登録する場合にはYを入力してください。(4)に進みます。

Nを入力すると

メニューにもどります。ESCキーを押してください

と表示されます。ESCキーを押すと始めのメニューにもどります。

(4) 郵便番号、住所、名前を入力してください。郵便番号は1バイト文字で6文字以内、住所は2バイト系日本語文字で21文字以内、名前は2バイト系日本語文字で8文字以内です。それ以上の文字を入力しようとすると、繰り返し入力を求めてきます。

データ変更の場合に何も文字を入力せずに  キーを押すと、そのデータは変更されません。

(5) 入力が終わると新しく入力されたデータが表示され、

これでよろしいですか(Y/N)?

と確認を求めてきます。よいならYを、間違っているならNを入力します。Yを入力するとデータをディスクに書き込み、

ディスクにデータを書き込みました。

と表示されます。

Nを入力すると、再び入力を求めてきますので正しく入力しなおしてください。

データの書き込みが終わると、

メニューにもどります。ESCキーを押してください

と表示されます。ESCキーを押すと始めのメニューにもどります。

・入力した文字列を含むデータがない場合

該当する名前が見つかった場合の(3)以降と同様です。

● “住所録表示” を選択した場合

ディスクファイル中に登録されている住所録のデータを画面に表示します。1画面に16件のデータを表示することができます。データ件数が16を超える場合には、次の画面に表示します。スペースキーを押すと次の画面へ進みます。まえの画面にもどるには **BS** キーを押します。**ESC** キーを押すと始めのメニューにもどります。

● “住所録印刷” を選択した場合

プリンタの用意をしてください。

用意ができたらスペースキーを押してください。

と表示されます。プリンタの電源を入れてコンピュータ本体との接続を確認してください。用紙をセットし、オンライン(SEL)ボタンを押します。オンライン(SEL)のLEDが点灯していれば印刷が可能です。

スペースキーを押すと登録されているすべての住所録データが印字されます。

● “宛て名印刷” を選択した場合

登録番号(1-999)?

と表示されるので宛て名を印刷したい相手の登録番号を入力してください。

入力された登録番号のデータが未登録の場合は、

このデータは登録されていません

と表示され、再び入力を求めてきますので正しい番号を入力してください。

郵便番号、住所、名前が印字されます。郵便番号の前には“〒”のマークが印字されます。

また、名前は横倍角で印字され、名前の後には“様”という字が印字されます。

登録番号の入力で、0または **↵** キーを入力すると、

メニューにもどります。ESC キーを押してください

と表示されるので、始めのメニューにもどることができます。

● “終了” を選択した場合

終了しました

と表示してプログラムを終了します。

---

注意：このサンプルプログラムの対応プリンタはPC-PR201系のプリンタです。他のプリンタでは正常な印字ができない場合があります。

---

## ●プログラムの解説

## メインルーチン(100-340行)

おもに、住所録プログラムのメニューの表示を行っています。

メニューの項目のうち選択している部分を反転表示しています。、キーを押すと反転表示の部分が変わります。

## データ入力のルーチン(1000-1650行)

1000-1070行 名前を検索する文字列を入力します。

1090-1250行 変更の場合は現在のデータを表示し、新しいデータ入力を行います。  
新規の場合は新しいデータの入力を行います。

1270-1320行 変更の許可を選択するルーチンです。

1340-1550行 キーボードからのデータの入力を行います。

1570-1640行 データをディスクへ書き出します。

## 住所録表示のルーチン(2000-2240行)

ディスクファイルから住所録のデータを16件ずつ入力し、画面に出力します。

## 住所録印刷のルーチン(3000-3220行)

ディスクファイルから住所録のデータを入力し、プリンタに出力します。

## 宛て名印刷のルーチン(4000-4350行)

入力された登録番号のデータを検索し、見つかった場合には一定の書式でプリンタに出力します。

## 終了のルーチン(5000-5010行)

プログラムを終了し、BASICのコマンドモードにもどります。

## その他のルーチン(6000-6190行)

6000-6050行 項目を表示します。

6070-6120行 データを表示します。

6140-6190行 メニューにもどるルーチンです。メニューにもどる前には必ずこのルーチンを実行します。このときファイルはクローズされます。

---

注意：“宛て名印刷”部分にプリンタの紙送りを行う命令を付け加えることにより、市販のラベル紙に宛て名を印字することも可能です。紙を送る命令について詳しくはプリンタに添付されているマニュアルを参照してください。

---

```

100 '
110 '=====  

120 '
130 KI$=CHR$(&H1B)+CHR$(&H4B):KO$=CHR$(&H1B)+CHR$(&H48):M=0
140 CONSOLE 0,20,0,0:WIDTH 80,20
150 *MENU:CLS
160 LOCATE 16,0:PRINT "***** 住 所 録 プ ロ グ ラ ム *****"
170 LOCATE 30,4:PRINT "デ ー タ 入 力"
180 LOCATE 30,6:PRINT "住 所 録 表 示"
190 LOCATE 30,8:PRINT "住 所 録 印 刷"
200 LOCATE 30,10:PRINT "宛 て 名 印 刷"
210 LOCATE 30,12:PRINT "終 了"
220 COLOR @ (29,4+2*M)-(29+19,4+2*M),4
230 I$=INKEY$
240 IF I$="" THEN 230
250 IF I$=CHR$(13) THEN GOTO 320
260 IF I$=CHR$(30) AND M>0 THEN M1=M:M=M-1:GOTO 290
270 IF I$=CHR$(31) AND M<4 THEN M1=M:M=M+1:GOTO 290
280 GOTO 230
290 COLOR @ (29,4+2*M1)-(29+19,4+2*M1),0
300 COLOR @ (29,4+2*M)-(29+19,4+2*M),4
310 GOTO 230
320 ON M+1 GOSUB 1000,2000,3000,4000,5000
330 GOTO *MENU
340 '
1000 '=====  

1010 I=0:CLS
1020 LOCATE 0,0:PRINT" [ デ ー タ 入 力 ] "
1030 LOCATE 0,2:PRINT"名前を検索する文字列を入力してください。";
1040 KINPUT NAME$:CLS 3
1050 IF NAME$="" THEN LOCATE 0,2:PRINT STRING$(40,CHR$(32)):GOTO 1020
1060 OPEN "ADDRES.DAT" AS #1
1070 FIELD #1,3 AS AA$,6 AS BB$,42 AS CC$,16 AS DD$
1080 '
1090 *KENSAKU '検索
1100 IF LOC(1)=LOF(1) THEN 1180
1110 GET #1,I+1
1120 IF KINSTR(1,KI$+DD$+KO$,NAME$)=0 THEN I=I+1:GOTO *KENSAKU
1130 TA$=AA$:TB$=BB$
1140 TC$=CC$:TD$=DD$
1150 GOSUB *HYOUJ11
1160 GOSUB *HYOUJ12
1170 GOTO *HENKOUKYOKA
1180 CLS 3
1190 PRINT "その名前は登録されていません。"
1200 PRINT "<" + NAME$ + ">"; "さんを新規に登録しますか (Y/N) ?";
1210 I$=INKEY$
1220 IF I$="" GOTO 1210
1230 IF I$="Y" OR I$="y" THEN I=LOF(1):TB$="":TC$="":TD$="":GOTO *NYURYOKU
1240 IF I$="N" OR I$="n" THEN GOTO *MENUHEKAERU
1250 GOTO 1210
1260 '
1270 *HENKOUKYOKA '変更許可
1280 LOCATE 0,4:INPUT"これを変更しますか (Y/N)";I$
1290 IF I$="" THEN 1280
1300 IF I$="Y" OR I$="y" THEN *NYURYOKU
1310 IF I$="N" OR I$="n" THEN LOCATE 25,4:PRINT STRING$(1,CHR$(32)):I=I+1:GOTO *KENSAKU
1320 GOTO *HENKOUKYOKA
1330 '
1340 *NYURYOKU '入力
1350 LOCATE 0,6:INPUT"郵便番号 (xxx-xx)";B$
1360 IF B$="" THEN B$=TB$
1370 IF LEN(B$)>6 THEN LOCATE 0,6:PRINT STRING$(80,CHR$(32)):GOTO 1350
1380 LOCATE 0,8:PRINT "住所 (21文字以内)";:KINPUT C$
1390 IF C$="" THEN C$=KI$+TC$+KO$
1400 IF KLEN(C$,2)>21 THEN LOCATE 0,8:PRINT STRING$(160,CHR$(32)):GOTO 1380
1410 C$=KEXT$(C$,1)+STRING$(42-KLEN(C$,2)*2,CHR$(&H21))
1420 LOCATE 0,10:PRINT "名前 (8文字以内)";:KINPUT D$

```

```

1430 IF D$="" THEN D$=K1$+TD$+K0$
1440 IF KLEN(D$,2)>8 THEN LOCATE 0,10:PRINT STRING$(80,CHR$(32)):GOTO 1420
1450 D$=KEXT$(D$,1)+STRING$(16-KLEN(D$,2)*2,CHR$(&H21))
1460 TB$=B$:TC$=C$:TD$=D$
1470 CLS
1480 GOSUB *HYOUJ11
1490 GOSUB *HYOUJ12
1500 LOCATE 0,5:PRINT "これでよろしいですか(Y/N)?"
1510 I$=INKEY$
1520 IF I$="" THEN 1510
1530 IF I$="Y" OR I$="y" THEN *KAKIKOMI
1540 IF I$="N" OR I$="n" THEN *NYURYOKU
1550 GOTO 1510
1560 '
1570 *KAKIKOMI '書き込み
1580 LSET AA$=MID$(STR$(1+1),2)
1590 LSET BB$=B$
1600 LSET CC$=C$
1610 LSET DD$=D$
1620 PUT #1,1+1
1630 LOCATE 0,5:PRINT "ディスクにデータを書き込みました。"
1640 GOTO *MENUHEKAERU
1650 '
2000 '===== 住所録表示 =====
2010 T=0:CH=0:CLS
2020 OPEN "ADDRES.DAT" AS #1
2030 FIELD #1,3 AS AA$,6 AS BB$,42 AS CC$,16 AS DD$
2040 CLS:LOCATE 0,0:PRINT" [住所録表示] "
2050 GOSUB *HYOUJ11
2060 I=1
2070 IF LOF(1)=0 THEN CH=1:GOTO 2160
2080 GET #1,1+16*T
2090 IF LOC(1)=LOF(1) THEN CH=1
2100 LOCATE 0,1+1:PRINT AA$
2110 LOCATE 10,1+1:PRINT BB$
2120 LOCATE 20,1+1:PRINT K1$+CC$
2130 LOCATE 64,1+1:PRINT K1$+DD$
2140 IF CH=1 THEN GOTO 2160
2150 IF I<16 THEN I=I+1:GOTO 2080
2160 LOCATE 0,19
2170 PRINT T+1;+"/";(LOF(1)-1)¥16+1;SPC(2);"E S Cキーでメニューへ。";
2180 IF CH<>1 THEN PRINT SPC(2);"スペースキーで次へ。";
2190 IF T>0 THEN PRINT SPC(2);"B Sキーで前へ。";
2200 I$=INKEY$
2210 IF I$=CHR$(32) AND CH<>1 THEN T=T+1:GOTO 2040
2220 IF I$=CHR$(8) AND T>0 THEN T=T-1:CH=0:GOTO 2040
2230 IF I$=CHR$(27) THEN CLOSE :RETURN
2240 GOTO 2200
2250 '
3000 '===== 住所録印刷 =====
3010 CLS
3020 LOCATE 0,0:PRINT" [住所録印刷] "
3030 LOCATE 0,2:PRINT"プリンタの用意をしてください。"
3040 PRINT "用意ができたならスペースキーを押してください。"
3050 I$=INKEY$
3060 IF I$="" THEN 3050
3070 IF I$<>CHR$(32) GOTO 3050
3080 LPRINT " [登録番号] "; " [郵便番号] ";SPC(6);
3090 LPRINT " [住所] ";SPC(33);
3100 LPRINT " [名前] "
3110 LPRINT
3120 OPEN "ADDRES.DAT" AS #1
3130 FIELD #1,3 AS AA$,6 AS BB$,42 AS CC$,16 AS DD$
3140 IF LOF(1)=LOC(1) THEN GOTO 3210
3150 GET #1
3160 LPRINT SPC(2);AKCNV$(AA$);SPC(3);
3170 LPRINT AKCNV$(BB$);SPC(6);
3180 LPRINT K1$:CC$:K0$:SPC(10);
3190 LPRINT K1$:DD$:K0$

```

```

3200 GOTO 3140
3210 LOCATE 0,5:PRINT "印刷が終わりました。"
3220 GOTO *MENUHEKAERU
3230 '
4000 '===== 宛て名印刷 =====
4010 OPEN "ADDRES.DAT" AS #1
4020 FIELD #1,3 AS AA$,6 AS BB$,42 AS CC$,16 AS DD$
4030 CLS:LOCATE 0,0:PRINT "[宛て名印刷]"
4040 LOCATE 0,2:INPUT "登録番号(1-999)";I
4050 IF I=0 THEN *MENUHEKAERU
4060 IF I>999 OR I<0 THEN LOCATE 0,2:PRINT STRING$(80,CHR$(32)):GOTO 4040
4070 IF I>LOF(1) THEN PRINT "このデータは登録されていません。":GOTO 4040
4080 GET #1,I:I=I-1
4090 TB$=BB$:TC$=CC$:TD$=DD$
4100 CLS
4110 GOSUB *HYOUJ11
4120 GOSUB *HYOUJ12
4130 LOCATE 0,4:PRINT "これを印字しますか(Y/N)";
4140 I$=INKEY$
4150 IF I$="" THEN 4130
4160 IF I$="Y" OR I$="y" THEN *INJI
4170 IF I$="N" OR I$="n" THEN LOCATE 0,4:PRINT STRING$(1,CHR$(32)):I=I+1:GOTO
4030
4180 GOTO 4130
4190 '
4200 *INJI '印刷
4210 LOCATE 0,6:PRINT"プリンタの用意をしてください。"
4220 PRINT "用意ができたらスペースキーを押してください。"
4230 I$=INKEY$
4240 IF I$="" THEN 4230
4250 FOR K=16 TO 1 STEP -1
4260 IF MID$(TD$,K,1)<>CHR$(&H21) THEN 4280
4270 NEXT K
4280 TD$=MID$(TD$,1,K)
4290 LPRINT "〒";
4300 LPRINT TB$:LPRINT
4310 LPRINT K I$:TC$:LPRINT
4320 LPRINT STRING$(10,CHR$(&H21))+CHR$(&HE)+TD$+K0$;
4330 LPRINT "様";CHR$(&HF)
4340 LOCATE 0,8:PRINT "印刷が終わりました"
4350 GOTO *MENUHEKAERU
4360 '
5000 '===== 終了 =====
5010 CLS 3:PRINT "終了しました":END
5020 '
6000 *HYOUJ11 '項目表示
6010 LOCATE 0,1:COLOR 4:PRINT "登録番号"
6020 LOCATE 10,1:PRINT "郵便番号"
6030 LOCATE 20,1:PRINT "住 所"
6040 LOCATE 64,1:PRINT "名 前":COLOR 0
6050 RETURN
6060 '
6070 *HYOUJ12 'データ表示
6080 LOCATE 0,2:PRINT MID$(STR$(I+1),2)
6090 LOCATE 10,2:PRINT TB$
6100 LOCATE 20,2:PRINT K I$+TC$
6110 LOCATE 64,2:PRINT K I$+TD$
6120 RETURN
6130 '
6140 *MENUHEKAERU 'メニューへ帰る
6150 LOCATE 0,9:PRINT "メニューにもどります。ESCキーを押してください。"
6160 I$=INKEY$
6170 IF I$="" THEN 6160
6180 IF I$=CHR$(&H1B) THEN CLOSE :RETURN
6190 GOTO 6160

```

# 第6章

## ファイル

### 6.1 ファイルとは

ファイルとは、ある一定の決まり、あるいは体系に従ってまとめられた、情報やデータのかたまり(あるいはその入れ物)のことです。たとえば、会社などで社員の個人的なデータ(人事データ)をバイндаなどにとじたものならば“人事ファイル”,あるいは、売り上げの品目や金額などのデータをキャビネットに取めたものならば“売り上げファイル”,といったぐあいです。

コンピュータにおいても、このファイルの概念はまったく同じで、あるまとまりをもったデータ(あるいはその入れ物)をすべてファイルとして取り扱います。一般的なファイルとの大きな違いは、そのデータのかたまりがバイндаやキャビネットではなく、フロッピィディスクやハードディスクなどに電氣的に格納されることです。電氣的に記録されたファイルの中のデータを取り出したり、逆に格納したりするには、コンピュータに対して手順を指示することにより行います。

BASICには、コンピュータでファイルを取り扱うために必要な命令や関数が各種用意されており、これらを組み合わせて使うことによって、データをファイルの形で自由に操作することができます。

#### ■ファイルの種類

ファイルはいろいろな観点から分類することができますが、BASICで一般に使用される分類法に従うと、次のようになります。

##### 1. 内容による分類

プログラムファイル…… BASICで記述されたプログラムを記録したファイル。

データファイル…… BASICプログラムで利用するデータを記録したファイル。

##### 2. 記録形式による分類

アスキーファイル…… 人間が判読できる文字(アスキーキャラクタおよび日本語キャラクタ)から成るファイル。テキストファイルともいう。

バイナリファイル…… コンピュータで利用されるコード(機械語など)から成るファイル。人間がふつうには判読できないものを指す。

### 3. データファイルの形式による分類

シーケンシャルファイル

……………行単位から成るファイル。順次ファイル、逐次ファイルとも呼ばれ、アクセス(読み書き)はそのファイルの先頭部分から順に行う。

ランダムファイル……………定義可能な区分(フィールド)をもつレコード単位のファイル。高速なアクセス(読み書き)が可能。

### 4. 格納される装置による分類

ディスクファイル……………フロッピィディスク装置または固定ディスク装置に記録されるファイル。たんに“ファイル”と呼ぶ場合、このディスクファイルを指すことが多い。

デバイスファイル……………いろいろな入出力装置(デバイス)に対するアクセスでやりとりされるデータを論理的に扱ったファイル。キーボードファイル、スクリーンファイル、プリンタファイル、RS-232C 回線ファイルなどがある。広い意味ではディスクファイルも含めてデバイスファイルと呼ぶことがある。

このようにさまざまな分類が可能です。これらの分類はそれぞれの観点から分けており、実際のファイルはこれらの組み合わせで表されます。

#### 例 1) RS-232C 回線から送られてくるアスキー形式の BASIC プログラムのファイル

→ デバイスファイル(RS-232C 回線ファイル)でシーケンシャルファイルでアスキーファイルで、かつプログラムファイル

#### 例 2) ディスクにランダムファイルの形式で格納された売り上げの記録ファイル

→ ディスクファイルでランダムファイルでバイナリファイルで、かつデータファイル

### ■ デバイス(入出力装置)とファイル

BASIC では、あるまとまりをもったデータのかたまり(あるいはその入れ物)をすべてファイルとして取り扱うことについては前に説明したとおりです。この概念により、ディスクに記録されたデータをファイルとして操作できるわけですが、BASIC ではこれをさらに一般化して、たとえばプリンタやディスプレイ画面(スクリーン)、さらにはキーボードなどもファイルとして取り扱うことができるようにしています。つまり、こうした各種の入出力装置(デバイス)を、出力データや入力データのかたまりを格納する入れ物としてみなすことにより、ファイルとして扱うようにしているわけです。

実際に BASIC でデバイスに対してデータの入出力を行う場合には、そのデバイスに対応する

デバイス名を指定して操作を行います。次に、デバイスと、それに対応するデバイス名を示します。“入力/出力”の欄には、そのデバイスに対する入力/出力の命令が使用可能であるかどうかを示してあります。

| デバイス       | デバイス名 | 入力 | 出力 |
|------------|-------|----|----|
| ディスク       | 1:    | ○  | ○  |
|            | 2:    | ○  | ○  |
|            | 3:    | ○  | ○  |
|            | }     |    |    |
|            | 30:   | ○  | ○  |
| キーボード      | KYBD: | ○  | ×  |
| スクリーン      | SCRN: | ×  | ○  |
| プリンタ       | LPT:  | ×  | ○  |
| RS-232C 回線 | COM1: | ○  | ○  |
|            | COM2: | ○  | ○  |
|            | COM3: | ○  | ○  |

注意：ROM モード BASIC ではプリンタと RS-232C 回線のみが使用可能です。

また、各デバイスに対して使用できる入出力命令はデバイスの種類によって異なります。

| 入出力命令 \ デバイス                            | ディスク n: | COM: | KYBD: | SCRN: | LPT: |
|---|---------|------|-------|-------|------|
| BLOAD                                   | ○       | ○    | ×     | ×     | ×    |
| BSAVE                                   | ○       | ○    | ×     | ○     | ○    |
| CLOSE                                   | ○       | ○    | ○     | ○     | ○    |
| EOF                                     | ○       | ○    | ×     | ×     | ×    |
| FPOS                                    | ○       | ×    | ×     | ×     | ○    |
| GET                                     | ○       | ×    | ○     | ×     | ×    |
| INPUT #                                 | ○       | ○    | ○     | ×     | ×    |
| INPUT\$                                 | ○       | ○    | ○     | ×     | ×    |
| LINE INPUT #                            | ○       | ○    | ○     | ×     | ×    |
| LOAD                                    | ○       | ○    | ○     | ×     | ×    |
| LOC                                     | ○       | ○    | ○     | ×     | ×    |
| LOF                                     | ○       | ○    | ×     | ×     | ×    |
| OPEN                                    | ○       | ○    | ○     | ○     | ○    |
| PRINT #<br>(PRINT # USING)<br>(WRITE #) | ○       | ○    | ×     | ○     | ○    |
| PUT                                     | ○       | ×    | ×     | ○     | ○    |
| SAVE                                    | ○       | ○    | ×     | ○     | ○    |
| WIDTH                                   | ×       | ○    | ×     | ×     | ○    |
| CHAIN/MERGE                             | ○       | ×    | ×     | ×     | ×    |

○：使用できる    ×：使用できない

## ■ファイルディスクリプタ

BASICでファイルを操作する際には、コンピュータがファイルを捜したり特定したりすることが可能なように、個々のファイルに名前をつけて取り扱います。これにより、ユーザーがその名前を指定するだけで、そのファイルに対してアクセス(読み書き)を行うことができます。この名前のことを「ファイルディスクリプタ」といいます。その形式は次のとおりです。

[<デバイス名>][<ファイル名>]

<デバイス名>は各入出力装置につけられた名前のもので、前出の「デバイス(入出力装置)とファイル」の項の表に示したもののうちのいずれかを指定します。

ディスクファイルの場合、<ファイル名>は次の形式で指定します。

<主ファイル名> [. <拡張子>]

▶参照 ファイル名→6.5の「ディスクファイルにおけるファイルディスクリプタの形式」

例) ディスク装置のドライブ2にセットされたディスク上の「TEST」というファイル。  
2:TEST

---

注意:<デバイス名>がキーボード(KYBD:), スクリーン(SCRN:)およびプリンタ(LPT:)である場合は<ファイル名>を指定しても意味がありません。

<デバイス名>がRS-232C回線(COM1:~COM3:)の場合は、<ファイル名>に相当する場所に回線の各種パラメータなどを指定します(「6.8 RS-232C回線ファイルの扱い方」参照)。

---

## ■ファイルバッファ

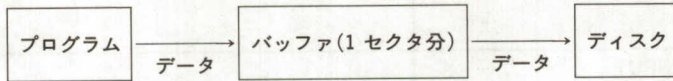
BASICでファイルを扱う際は、ファイルバッファと呼ばれるメモリの領域を使用します。ファイルバッファはプログラムとファイルとの間で一時的にデータを蓄えるものです。

プログラムからファイルに対してデータを出力する、あるいはファイルからプログラムへデータを読み込むといった場合には、必ずファイルバッファを仲介します。たとえばディスクファイルにデータを書き込む場合は、いったんファイルバッファにデータを蓄え、蓄えたデータが256バイト(1セクタ分)になったら、ファイルに書き出すという動作を行っています。

つまり、ファイルに対して入出力命令を実行しても実際にはバッファに対する入出力を行っているわけで、ファイルに対する実際の入出力はバッファがいっぱい(あるいは空)になったかどうかをBASICが自動的に判別して行っています。

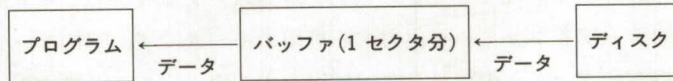
▶参照 セクタ→「付録C ディスクの内部構造」

## プログラムからディスクへ書き込む場合



データをバッファに書き出す。  
 バッファが1セクタ分たると BASIC が自動的にディスクに書き出す。

## ディスクからプログラムへデータを読み込む場合



ディスクから1セクタ分のデータをバッファへ読み込む。  
 プログラムはバッファから必要なだけデータを読み込む。  
 バッファが空になったら、BASIC が自動的にディスクから1セクタ分読み込む。

ファイルの入出力を行う際には必ずファイルごとに専用のファイルバッファが必要です。さもないといろんなファイルのデータがファイルバッファでまじりあってしまうことになります。

ファイルとファイルバッファの対応関係は始めから決まっているものではありません。ファイルの入出力を行う前には必ず、そのファイルに専用のファイルバッファを割り当てなければなりません。

## ■ファイルのオープン/クローズ

ファイルの入出力を行う前には、前項で述べたように、使用するファイルにファイルバッファを割り当てて入出力操作が可能な状態にしなければなりません。これをファイルをオープンするといいます。また、ファイルの入出力が終了したらファイルに割り当てられていたファイルバッファを解放する必要があります。これをファイルのクローズといいます。

## ●ファイルのオープン(OPEN)

ファイルのオープンは OPEN 命令を使って行います。

## OPEN 命令

OPEN <ファイルディスクリプタ> [FOR <モード>] AS [#] <ファイル番号>

OPEN 命令は、<ファイルディスクリプタ>で指定されるファイルに<ファイル番号>で表される専用バッファを割り当てます。

この命令が実行された後は、ファイルに対する入出力は<ファイル番号>を指定して行います。ファイルの入出力命令は実際にはバッファへ入出力を行っているわけですから、<ファイル番号>はファイルバッファ番号と考えてさしつかえありません。

〈モード〉はファイルへのアクセスの仕方を指定するものです。次の4つから選択することができます。

|        |                              |
|--------|------------------------------|
| INPUT  | 入力モード                        |
| OUTPUT | 出力モード                        |
| APPEND | 追加モード                        |
| 省略時    | ランダムモード、またはRS-232C回線ファイルの入出力 |

入力モードは既存のファイルのデータをプログラムへ入力する専用のモードです。出力モードはプログラムからデータをファイルへ出力する専用のモードです。追加モードはプログラムからデータを送り既存のファイルの終わりに追加出力する専用のモードです。

省略時はランダムモード、あるいはRS-232C回線ファイルの入出力を意味し、入力、出力の両方が可能なモードです。

---

注意：入力(INPUT)モード、追加(APPEND)モードでオープンした場合、指定したファイルがディスク上にないとエラーとなります。

---

例1) ドライブ2の"TEXT"という名前のファイルに1番のファイルバッファを割り当て、入力モードで使用可能な状態にする。

```
OPEN "2:TEXT" FOR INPUT AS #1
```

例2) スクリーンファイルに2番のファイルバッファを割り当て、出力モードで使用可能な状態にする。

```
OPEN "SCRN:" FOR OUTPUT AS #2
```

例3) ドライブ1の"DATA"という名前のファイルに3番のファイルバッファを割り当て、ランダムモードで使用可能な状態にする。

```
OPEN "DATA" AS #3
```

### ●ファイルのクローズ(CLOSE)

ファイルの入出力が終了すると、OPEN命令で確保したファイルバッファは不要になります。そこでCLOSE命令を実行すると、ファイルに割り当てられていたファイルバッファが解放されます。

これで、解放されたファイルバッファは別のファイルのために使用することができるようになります。

また、前に述べたようにデータの出力命令は実際にはファイルバッファへのデータ出力を行うだけであり、それが実際にファイルに出力されるのはファイルバッファがいっぱいになって

からです。したがって、最後にファイルバッファに出力されたデータがファイルに出力されずにファイルバッファに残ってしまう可能性があります。CLOSE 命令は、このようにしてファイルバッファに残ってしまったデータを、最後に(ファイルバッファを解放する前に)ファイルに出力するという役割も持っています。

### CLOSE 命令

CLOSE [(#)〈ファイル番号〉[, (#)〈ファイル番号〉]…]

CLOSE 命令では〈ファイル番号〉を指定してファイルのクローズを行います。〈ファイル番号〉を省略した場合にはそのときオープンしているファイルはすべてクローズされます。

ファイルをクローズしない状態でプログラムを終了した場合には、ファイルバッファに残されたデータは失われてしまいます。特にディスクファイルの時には最悪の場合、ディスクを管理するための情報が不十分なままで処理が終了してしまい、ディスクが破壊されてファイルやディスクの内容が読めなくなる、という事態をひき起こすことがあります。このような事態を防ぐために、入出力の終わったファイルは必ずクローズするようにしてください。

なお、END 命令、NEW 命令、RUN 命令などの実行によってもファイルがクローズされるようになっています。


### ■ファイルバッファとファイルの同時オープン数

ファイルバッファはメモリ上に確保されるものです。N<sub>88</sub>-日本語 BASIC(86)では最大 15 個のファイルバッファが確保されます。ただし、0 番のファイルバッファは特殊なバッファで普通ユーザーは使用できませんから、実際には最大 14 個のファイルバッファが使用可能です。これは、同時に 14 個のファイルを使用することができるということを意味します。

BASIC の起動時に、

How many files (0-15)?

と聞いてくるのはファイルバッファを何個確保するかを設定するものです。ここで 1~15 までの数値を入力します。ここでの指定は、BASIC が起動した後では変更することができません。変更するには BASIC を再起動する必要があります。そこで、できるだけたくさんのファイルバッファを使える状態にしておけばよいのかというそういうわけではありません。ファイルバッファの領域はメモリ上に確保されますので、ここで大きな数を指定するとその分ユーザーが使用できるメモリが減ってしまいます。特別に多くのファイルを同時オープンするのでなければ、あまり多く設定しておいても無意味です。

なお、起動時の“How many files?”に対して  キーのみを押すと、そのとき接続(内蔵も含む)されているドライブ数と同数のファイルバッファが設定されたものとみなされます。

ただし、最大値は 15 です。

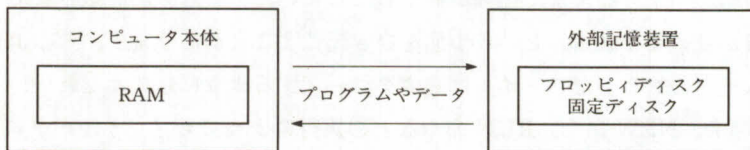
## 6.2 ディスクファイル

この節では外部記憶装置として一般的なフロッピーディスクと固定ディスクの種類について解説します。

### ■外部記憶装置としてのディスク装置

コンピュータには読み書き可能なメモリ (RAM) があり、データやプログラムはここに一時的に置かれて処理されます。しかし、電源を切るとこれらのデータは失われてしまいます。

そこで作成したプログラムやデータは外部記憶装置に保存します。外部記憶装置を使うことによって、プログラムやデータを保存し必要なときに取り出して使用することができます。



BASIC では、外部記憶装置としてフロッピーディスク装置と固定ディスク装置が使用できます。

### ■フロッピーディスクと固定ディスク

フロッピーディスクは、容易に持ち運びができます。ただし、記憶容量は固定ディスクより小さくなっています。

いっぽう、固定ディスクは重く持ち運びには不向きですが、フロッピーディスクに比べて容量がたいへん大きくアクセスの速度も高速です。特に多量のデータを扱う場合にはたいへん便利です。

#### 使用できるフロッピーディスクの種類と容量

| フロッピーディスクの種類 |                | 容量    |
|--------------|----------------|-------|
| 3.5 インチディスク  | 2DD (倍密度倍トラック) | 640KB |
|              | 2HD (高密度倍トラック) | 1MB   |
| 5 インチディスク    | 2DD (倍密度倍トラック) | 640KB |
|              | 2HD (高密度倍トラック) | 1MB   |
| 8 インチディスク    | 2D (倍密度)       | 1MB   |

\* これらのディスクのうち容量が同じディスクは同一の論理構造を持ちます。

▶ 参照 論理構造→「付録 C ディスクの内部構造」

### 使用できる固定ディスクの種類と容量

固定ディスクは、固定ディスクインタフェースを介してコンピュータ本体と接続されていますが、この固定ディスクインタフェースには2つの種類があります。

1つは、従来からあるもので1つのインタフェースで2台の固定ディスクを接続することができます。もう1つはSCSIと呼ばれるもので、1つのインタフェースで4台までの固定ディスクを接続することができます。従来のインタフェースで接続できる固定ディスクを従来インタフェースの固定ディスク、SCSIインタフェースに接続できる固定ディスクをSCSI固定ディスクといいます。

| 固定ディスクの種類            | 容量    |
|----------------------|-------|
| 従来インタフェースの<br>固定ディスク | 5MB   |
|                      | 10MB  |
|                      | 20MB  |
|                      | 40MB  |
| SCSI 固定ディスク          | 20MB  |
|                      | 40MB  |
|                      | 130MB |

## 6.3 フロッピーディスクの使用準備と管理

ここでは、フロッピーディスクを使用する上での準備と管理の方法について解説します。

### ■フロッピーディスクのフォーマット

買ったばかりの新しいフロッピーディスクはそのままでは使うことはできません。一定の手順を踏んで使用できる状態にしなければなりません。これをフォーマットといいます。フォーマットはディスクのデータ格納方式に従ってディスクの領域を分けし、ディスクにデータを書き込むことができるようにするものです。

フォーマットには2つのレベルがあり、それぞれ物理フォーマット、論理フォーマットといいます。

**物理フォーマット** ディスクにデータを記録するための磁気的な溝(トラック)を作成する。

**論理フォーマット** ディスクの記録面を論理的に分けし、BASICでデータを管理できる状態にする。

フロッピーディスクのフォーマットをするにはユーティリティ「format.nip」を使用します。

▶参照 「format.nip」 →「10.3 フロッピーディスクのフォーマット」

## ■システムディスクの作成

システムディスクとは、DISK モード BASIC を起動させることのできるディスクです。システムディスクは間違った使い方をしたり、不慮の事態が起こった場合には、内容が破壊されてしまうことがあります。そこで、システムディスクの複製を作っておく必要があります。

### ●システムディスクの複製の作成

システムディスクには、システム(DISK CODE と IPL)が記録されています。

DISK CODE とは、ROM モード BASIC に DISK モード BASIC の機能を付加するために必要な BASIC システム本体のことです。また、IPL は DISK モード BASIC を起動させるために必要なシステムプログラムのことです。

システムディスクにはこの2種類のデータが記録されているために、DISK モード BASIC が起動できます。したがって、システムディスク中に記録されている DISK CODE と IPL を別のフロッピーディスクにコピーすればシステムディスクを作成することができます。

システムディスクの複製を作るのにいちばん簡単なのは、ユーティリティ "backup.n88" を使う方法です。この方法では元のシステムディスクとまったく同一のディスク(バックアップディスクという)を作成することができます。つまり、DISK CODE、および IPL 以外のファイルもコピーされます。

また、ユーティリティ "format.nip" や "sysgen.nip" を使ってシステムのコピーをすることもできます。この2つの方法では他のファイルの転送は行われません。

システムディスクの具体的な作成方法は「第10章 ユーティリティプログラム」の該当の項目を参照してください。

- ▶参照 format.nip →「10.3 フロッピーディスクのフォーマット」
- sysgen.nip →「10.7 システム(DISK CODE および IPL)のコピー」
- backup.n88 →「10.4 フロッピーディスクのバックアップ」

---

注意：システムディスクの複製を作るには、元になるシステムディスクと新しくコピーするディスクが同じ記憶容量をもつものでなければなりません。

---

### ●システムディスクの属性の設定

システムをコピーしてシステムディスクを作成すると、システムディスクの属性は元のディスクと同じになりますが、この属性を "setup.n88" を使って変更することができます。

システムディスクには次のような属性が設定されています。

- ・日本語変換方式
- ・辞書ファイルのドライブ番号

- ・画面ハードコピーの機能
- ・演算モード

必要に応じて変更を行ってください。

▶参照 setup.n88 →「10.8 システムディスク属性の設定」

## ■フロッピーディスクの管理

フロッピーディスクを管理するために次のようなユーティリティプログラムが用意されています。

| ユーティリティプログラム | プログラムの内容                          |
|--------------|-----------------------------------|
| setinf.n88   | オートスタートのための情報をシステムディスクに記録する       |
| DDconv.n88   | 1D/2Dのディスクのファイルを2DD/2HDのディスクに転送する |
| xfiles.n88   | ディスク間でファイルを転送する                   |
| backup.n88   | フロッピーディスクのバックアップの作成               |

なお、ファイルの削除やファイル名の変更などは BASIC の命令を用いて行います。

▶参照 フロッピーディスクを管理する BASIC の命令 →「6.6 BASIC 命令によるディスクファイルの操作」

setinf.n88 →「10.5 IDセクタの書き換え」

DDconv.n88 →「10.16 ファイル変換」

xfiles.n88 →「10.6 ファイル転送」

backup.n88 →「10.4 フロッピーディスクのバックアップ」

## 6.4 固定ディスクの使用

固定ディスクには、フロッピーディスクにはない固有の機能があります。この機能は固定ディスクの大きな記憶容量を効果的に使うために用意されているものです。

このため、固定ディスクの扱いはフロッピーディスクの場合と異なり、フォーマット、システムディスクの作成、ディスクの管理などに使用するユーティリティプログラムも別に用意されています。

ここでは固定ディスクに固有な機能、使用するための準備、管理の方法について解説します。

### ■固定ディスクに固有な機能

固定ディスクにはフロッピーディスクと異なる次のような機能があります。

| 機能                                 | 従来インタフェースの<br>固定ディスク |              | SCSI<br>固定ディスク |
|------------------------------------|----------------------|--------------|----------------|
|                                    | 標準<br>フォーマット         | 拡張<br>フォーマット | 拡張<br>フォーマット   |
| BASIC と他の OS で領域を分割して使用する機能        | ○                    | ○            | ○              |
| BASIC と他の OS などの起動可能なシステムを複数登録する機能 | ×                    | ○            | ○              |
| BASIC 上で複数のユーザーで使用する(マルチユーザー)機能    | ○                    | ○            | ○              |
| BASIC 上で複数のドライブに分割して同時に使用する機能      | ×                    | ×            | ○              |

○：使用できる，×：使用できない。

次に、これらの機能について解説します。

#### ● BASIC と他の OS で領域を分割して使用する

固定ディスクは、その記憶領域を BASIC 用とその他の OS(MS-DOS など)用に分割し、各々独立して使用することができます。BASIC が使用する領域の設定には、ユーティリティ“format .hd”を使用します。

▶参照 format.hd →「10.9 固定ディスクボリューム管理」

#### ● BASIC と他の OS などの起動可能なシステムを複数登録する

フロッピーディスクに登録できるシステム(起動可能なシステム)は1つだけです。このため、BASIC のシステムの入ったフロッピーディスクからは BASIC だけしか起動することができません。

これに対して、固定ディスクの場合には BASIC 以外にたとえば、MS-DOS などのシステムを登録しておき、コンピュータの起動時にどちらかを選択して使用することができます。

複数のシステムを選択して起動させるには、必ず固定ディスクを拡張フォーマットしておかなければなりません。その後、BASIC のシステムとその他のシステムの両方を登録しておけばどちらかを選択して起動させることができます。

---

**注意：**拡張フォーマットを行ったディスクに BASIC 以外のシステム(たとえば MS-DOS など)を登録する場合は、該当するシステムのマニュアルなどを参照して操作を行ってください。

---

▶参照 拡張フォーマット→「10.9 固定ディスクボリューム管理」

● BASIC 上で複数のユーザーで使用する(マルチユーザー機能)

固定ディスク内にユーザーが作成するファイルにはユーザーを識別する識別子をつけることができます。識別子を使用することにより、他のユーザーのファイルと区別することができます。これを“ユーザー識別名”といいます。

各々のユーザーは基本的に自分のユーザー識別名をもつファイル以外は扱うことはできません。こうして、1台の固定ディスクをあたかも複数の独立した記憶媒体の集まりであるかのように利用することができます。

この機能を“マルチユーザー機能”といいます。マルチユーザー機能は標準フォーマットをした固定ディスクでも拡張フォーマットをした固定ディスクでも使用することができます。

● BASIC 上で複数のドライブに分割して同時に使用する

従来インタフェースの固定ディスク(拡張フォーマット)では複数の BASIC 領域を確保して、1つの領域のみを有効(アクティブ)にすることができましたが、SCSI 固定ディスクは1台の固定ディスクに BASIC の領域を複数確保し、最大4つの領域を同時に有効にすることができます。

このとき、各領域は個々の独立したドライブとしてみなされます。たとえば、1台の固定ディスクに3~6までのドライブを割り当てて使用するということが可能です。1つの BASIC 領域について120MBまでの容量を確保することができます。

BASIC の領域確保や、領域の有効、無効の設定にはユーティリティ“format. hd”を使用します。

▶参照 format. hd →「10.9 固定ディスクボリューム管理」

### ユーザー識別名の宣言方法

ユーザー識別名の宣言は BASIC の起動時に行われます。


固定ディスクが接続(あるいは内蔵)されている機種で BASIC を起動させると、“How many files?”に続いて、“User identifier?”と聞いてきます。これは BASIC を起動させるユーザーが誰であるか(ユーザー識別名)を宣言するものです。

こうして起動させた BASIC 上で作成されたファイルには、そのユーザー識別名が付けられます。また、アクセスが可能なファイルは、基本的に自分のユーザー識別名で作成したファイルだけとなります。もちろん、他のユーザー識別名で作られたファイルにはアクセスできません。

### ユーザー識別名の条件

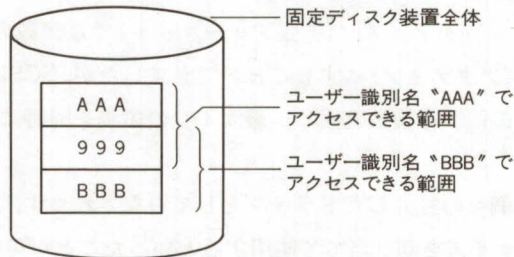
ユーザー識別名は1バイト英数字3文字以内で宣言します。

ただし、ユーザー識別名“999”で作られたファイルはすべてのユーザー共通のファイルとなり、すべてのユーザー識別名で読むことができます。

また、“User identifier?”の問いに  キーだけを押しすと、ユーザー識別名として、“□□□”(半角スペース3つ)が入力されたとみなされます。

ユーザー識別名が異なれば、同じ名前で作成することができます。ただし、“999”のユーザー識別名で作成されたファイル名と重複することはできません。ユーザー識別名とアクセス可能な範囲を模式的に表すと次のようになります。

### ユーザー識別名とアクセス可能範囲



- AAA : ユーザー識別名“AAA”のファイル群
- BBB : ユーザー識別名“BBB”のファイル群
- 999 : ユーザー識別名“999”のファイル群

また、ある特定のユーザー識別名でアクセスできるファイルは次のようになります。次の表はユーザー識別名“AAA”で BASIC を起動させた状態でどのファイルにアクセスできるかを示したものです。

## ユーザー識別名“AAA”からアクセスできるファイル

| 対象ファイル名                  |          | “AAA”の下のファイル<br>(同じユーザー識別名) | “BBB”の下のファイル<br>(他のユーザー識別名) | “999”の下のファイル<br>(共通ユーザー識別名) |
|--------------------------|----------|-----------------------------|-----------------------------|-----------------------------|
| ファイルに対する操作               |          |                             |                             |                             |
| OPEN” ~” FOR APPEND      |          | ○                           | ×                           | ○                           |
| OPEN” ~” FOR INPUT       |          | ○                           | ×                           | ○                           |
| OPEN” ~” FOR OUTPUT      |          | ○                           | ×                           | ×                           |
| OPEN” ~”<br>(ランダムモードの場合) | PUT (更新) | ○                           | ×                           | ○                           |
|                          | PUT (作成) | ○                           | ×                           | ×                           |
|                          | GET      | ○                           | ×                           | ○                           |
| KILL                     |          | ○                           | ×                           | ×                           |

\*” ~” はファイルディスクリプタを表します。

なお、ユーザー識別名“999”を持つファイルの作成・削除はユーザー識別名“999”で起動した BASIC 上でしかできません。

## ユーザー識別名を使わない場合

固定ディスクを1人で使用するときなど、ユーザー識別名が不必要なことがあります。このような場合には、ユーザー識別名を使用しない状態に設定することができます。

ユーティリティ“switch.n88”を使って、立ち上げ装置の選択画面で設定します。

▶参照 ユーザー識別名→1.2の「システムディスクからの起動」

switch.n88 →「10.14 メモリスイッチの設定」

---

注意：ユーザー識別名“999”を持つファイルは、誤って更新されることがないように、個人用のファイルと区別して一意な名前(たとえば、“～.SYS”など)を付けておくとよいでしょう。

---

## ■固定ディスクのフォーマット

固定ディスクを使用する場合には、フロッピーディスクの場合と同様にフォーマットをする必要があります。フォーマットの方法は、固定ディスクの種類や機種によって異なります。

### ●物理フォーマットと論理フォーマット

固定ディスクのフォーマットにはフロッピーディスクの場合と同様に物理フォーマットと論理フォーマットという2つのレベルがあります。新しいディスクを使用する場合には両方のフォーマットを続けて行わなければなりません。

物理フォーマットを行うとディスクの全領域はすべて初期化されてしまいます。したがって、それまで固定ディスクを使用していた場合にはデータはすべて失われてしまいます。たとえば、BASICとMS-DOSで使用していた場合にはその両方のデータが失われてしまいます。

また、論理フォーマットを行った場合にはBASICで確保されていた領域のみがフォーマットされます。MS-DOSなどの他のシステムで使用されている領域はそのままとなります。

---

参考：BASICでは固定ディスクを1セクタ256バイトで物理フォーマットします。

---

### ●標準フォーマットと拡張フォーマット

固定ディスクのフォーマット形式には標準フォーマットと拡張フォーマットの2つの形式があります。

#### 標準フォーマット

PC-9800シリーズで当初から採用されている形式です。ディスクをBASICで使用する領域とMS-DOSなどの他のシステムで使用する領域に分けることができます。ただし、標準フォーマットをしたディスクから起動できるシステムはBASICだけとなります。20MBまでの固定ディスクを管理することができます。

#### 拡張フォーマット

機能の拡張されたフォーマット形式です。ディスクの領域をBASICだけで使用する領域とMS-DOSなどの他のシステムで使用する領域に分けて使用することができただけでなく、複数のシステムのいずれかを任意に選択して起動させることができます。

また、20MBを超える大容量の固定ディスクも管理することができます。

さらに、SCSI固定ディスクのフォーマットが可能です。

## ● 固定ディスクの種類と可能なフォーマット形式

固定ディスクの種類によって、可能なフォーマット形式は次のようになります。

## 従来インタフェースの固定ディスクの場合

| 本体に従来インタフェースの固定ディスクが内蔵されているかどうか | 内蔵ディスク①         | 内蔵ディスク②         | 外付けディスク①                    |          | 外付けディスク②                    |                       |
|---------------------------------|-----------------|-----------------|-----------------------------|----------|-----------------------------|-----------------------|
|                                 |                 |                 | 5MB<br>10MB<br>20MB<br>40MB | 標準<br>拡張 | 5MB<br>10MB<br>20MB<br>40MB | 標準<br>拡張              |
| 1台内蔵の場合                         | 標準<br>または<br>拡張 | —               | —                           |          | 5MB<br>10MB<br>20MB<br>40MB | 標準<br>または<br>拡張<br>拡張 |
| 2台内蔵の場合                         | 標準<br>または<br>拡張 | 標準<br>または<br>拡張 | —                           |          | —                           | —                     |
| 内蔵していない場合                       | —               | —               | 5MB<br>10MB<br>20MB<br>40MB | 標準<br>拡張 | 5MB<br>10MB<br>20MB<br>40MB | 標準<br>拡張              |
|                                 |                 |                 | —                           |          | —                           |                       |

表中の①、②は従来インタフェースの固定ディスクの順番を示します。なお、従来インタフェースの固定ディスクは、コンピュータ1台について2台しか接続できません。

PC-9801F3/M3/VM4の場合、内蔵の固定ディスクに対して拡張フォーマットを行うことはできません。

## SCSI 固定ディスクの場合

20MB、40MB、130MBとも拡張フォーマットのみが可能です。

**注意：**ユーティリティ“format.hd”を使用して固定ディスクの拡張フォーマットをした場合は、フォーマット終了後リセットスイッチを押してBASICを再起動してください。  
拡張フォーマットの固定ディスクを使用する場合、標準フォーマットのディスクを使用する場合に比べ、従来インタフェースの固定ディスクで約18KB、SCSI固定ディスクでは約35KBのメモリを余分に必要とします。その分だけユーザーのメモリが減りますので注意してください。

### ●フォーマットをする方法

固定ディスクのフォーマットには、ユーティリティ `format.hd` を使用します。

ユーティリティ `format.hd` では標準フォーマットと拡張フォーマットのどちらかのフォーマットをすることができます。フォーマットを行う場合には「10.9 固定ディスクボリューム管理」を参照して行ってください。

▶参照 format.hd →「10.9 固定ディスクボリューム管理」

### ■固定ディスクからの BASIC の起動

DISK モード BASIC は、固定ディスクから起動させることもできます。

### ●システム(DISK CODE, IPL)のコピー

固定ディスクから BASIC を起動させるには、まずフォーマットを行った固定ディスクにシステム(DISK CODE, IPL)のコピーをします。これにはユーティリティ `sysgen.nip` を使用します。

▶参照 sysgen.nip →「10.7 システム(DISK CODE および IPL)のコピー」

---

注意：標準フォーマットをした固定ディスクにシステムをコピーする場合には、他のシステム(MS-DOS などの)の標準フォーマットより先に BASIC の標準フォーマットを行っておい  
てください。他のシステムの標準フォーマットの後に BASIC の標準フォーマットを行っ  
た固定ディスクに BASIC のシステムをコピーすると他のシステムの領域のデータが読め  
なくなります。

---

### ●システムの属性の設定

ユーティリティ `sysgen.nip` を使用して固定ディスクにシステムをコピーした場合、シ  
ステムの属性は元のシステムディスクの設定と同じになっています。そこで、必要があればユー  
ティリティ `setup.n88` を使用して、固定ディスクにコピーされたシステムの属性を変更す  
ることができます。

システムの属性には次のようなものがあります。

- 日本語変換方式
- 辞書ファイルのドライブ番号
- 画面ハードコピーの機能
- 演算モード

必要に応じて変更を行ってください。

---

注意：辞書ファイルのドライブ番号に設定したドライブに辞書ファイルがない場合には、日本語の変換ができません。

---

▶ 参照 setup.n88 →「10.8 システムディスク属性の設定」

### ●起動装置の選択

コンピュータ本体の出荷時の状態では、システムの起動装置の順番は「フロッピーディスク装置→固定ディスク装置」の順となっています。

この場合、固定ディスクに BASIC のシステムがコピーしてあったとしても、フロッピーディスク装置になんらかのシステムディスクが入っている場合には、コンピュータ本体の電源を ON またはリセットすると、フロッピーディスク装置に入っているシステムが起動してしまいます。もちろん、フロッピーディスク装置に起動可能なフロッピーディスクが入っていない場合には、固定ディスクから BASIC が起動します。

そこで、無条件に固定ディスクから起動したい場合には、起動装置の設定を固定ディスクのみに変更します。起動装置の選択は、ユーティリティ “switch.n88” の立ち上げ装置の選択画面で立ち上げ装置を「BOOT on hard disk 1」、「BOOT on hard disk 2」または、「BOOT on SCSI hard disk」に設定します。

なお、SCSI 固定ディスクを起動装置に設定した場合は、1 台目の固定ディスクのみが起動装置となります。2 台目以降を起動装置にすることはできません。

▶ 参照 switch.n88 →「10.14 メモリスイッチの設定」

### ●ドライブ番号の優先順位の選択

コンピュータ本体の出荷時の状態ではドライブ番号割付の優先順位は「フロッピーディスク装置→固定ディスク装置」の順になっています。「固定ディスク装置→フロッピーディスク装置」の順にしたい場合には、ユーティリティ “switch.n88” の立ち上げ装置の選択画面で固定ディスク優先割付けを「hard disk → floppy disk」に設定します。

▶ 参照 switch.n88 →「10.14 メモリスイッチの設定」

### ●辞書ファイルのコピー

キーボードから日本語を入力するには辞書ファイルが必要です。辞書ファイルは辞書ディスクから固定ディスクにコピーしておくのが便利です。辞書ファイルの固定ディスクへのコピーには “xfiles.n88” を使用します。

なお、辞書ファイルは必ずしも固定ディスクにコピーする必要はありません。辞書ファイルの入ったフロッピーディスクをフロッピーディスク装置に入れておけば、日本語入力を行うことができます。

また、辞書ファイルを入れるドライブ番号をシステムに登録しておく必要があります。

ユーティリティ `setup.n88` の日本語入力属性の設定の選択画面で、BASIC システムが参照する辞書ファイルのドライブ番号を指定してください。

- ▶ 参照 xfiles.n88 →「10.6 ファイル転送」
- setup.n88 →「10.8 システムディスク属性の設定」

### ■ 固定ディスクの管理

次のようなユーティリティプログラムを使って固定ディスク、およびそのファイルの管理をすることができます。

| ユーティリティプログラム | プログラムの内容  |
|--------------|---|
| setinf.n88   | オートスタートのための情報をシステムディスクに記録する                       |
| xfiles.n88   | ディスク間でファイルを転送する                                   |
| recov.hd     | 固定ディスクの障害のある領域(クラスタ)の中から、正常な部分(セクタ)を見つけてデータの復旧を行う |
| dir.hd       | 固定ディスクのファイルディレクトリ(一覧表)の表示およびファイルの削除               |
| backup.hd    | 固定ディスクファイルの内容をフロッピーディスクにバックアップをとる。また、元の固定ディスクにもどす |

なお、ファイルの削除、ファイル名の変更などはフロッピーディスクの場合と同様に BASIC の命令を使って行います。

- ▶ 参照 ファイルを管理する BASIC の命令 →「6.6 BASIC 命令によるディスクファイルの操作」
- setinf.n88 →「10.5 ID セクタの書き換え」
- xfiles.n88 →「10.6 ファイル転送」
- recov.hd →「10.10 固定ディスク障害ボリューム・ファイル復旧」
- dir.hd →「10.11 固定ディスクファイルディレクトリ表示」
- backup.hd →「10.12 固定ディスクファイル退避/復旧処理」

## 6.5 ファイルディスクリプタとドライブ番号

BASIC 命令でディスクファイルを作成したり、更新したりといったファイル操作を行う場合には、ファイルをファイルディスクリプタという概念で扱います。

### ■ ディスクファイルにおけるファイルディスクリプタの形式

ディスクのファイルディスクリプタの形式は次のようになります。

[<ドライブ番号>][<ファイル名>]

<ドライブ番号>は BASIC が起動したときにディスク装置の各ドライブに割り当てられる一連の番号です。本体に接続しているドライブの数に応じたドライブ番号にコロン(:)をつけて指定します。ドライブは最大 30 まで使用することができますので、指定できるドライブ番号も 30 以内となります。

<ドライブ番号>を省略したときは、“1:” が指定されたものとみなされます。

<ファイル名>は、次の形式で指定します。

<主ファイル名>[.<拡張子>]

<主ファイル名>は 1~6 文字の 1 バイト系英数文字から成る文字列です。

<拡張子>は 1~3 文字の 1 バイト系英数文字から成る文字列で、一般にファイルの内容を識別するのに使用します。省略可能ですが、指定する場合には必ずその前にピリオド(.)をつけなければなりません。

なお、<拡張子>をつけないときには、<ファイル名>として 1~9 文字の 1 バイト系英数文字から成る文字列を指定することができます。

いずれの場合も、英文字は大文字/小文字が区別されます。

なお、“主ファイル名”をたんに“ファイル名”と表現する場合があります。

例) 1 : TEST.TXT  
 ABC.BAS  
 2 : store

## ■ドライブ番号の割り当てられ方

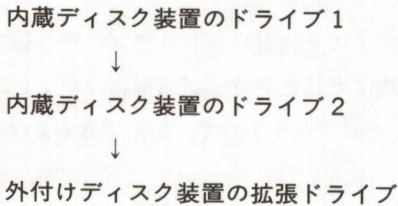
フロッピーディスク装置だけを接続(あるいは内蔵)している場合と、固定ディスクが接続(あるいは内蔵)されている場合とではドライブ番号の割り当てられ方が異なります。

### ●フロッピーディスク装置だけを接続(あるいは内蔵)している場合

この場合、本体内にディスク装置を内蔵しているかないかによって割り当てられ方がさらに2とおりに分かれます。

#### 本体にディスク装置が内蔵されている場合

次の順で番号が割り当てられます。

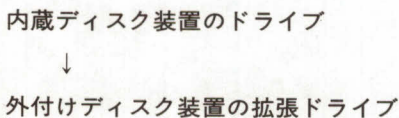


外付けの拡張ドライブとして、異なった種類のディスク装置がいくつか使用されているときには、システムディスクと同じ種類のディスクを使用するディスク装置のドライブが優先されます。

例) 5インチ2HDのドライブ(2台)を内蔵している機種に、5インチ2HDの拡張ドライブ(2台)と3.5インチ2DDの拡張ドライブ(2台)を外付けして、本体のドライブからBASICを起動させたとき。

| ドライブ             | ドライブ番号 |
|------------------|--------|
| 5インチ2HD内蔵ドライブ1   | 1:     |
| 5インチ2HD内蔵ドライブ2   | 2:     |
| 5インチ2HD拡張ドライブ1   | 3:     |
| 5インチ2HD拡張ドライブ2   | 4:     |
| 3.5インチ2DD拡張ドライブ1 | 5:     |
| 3.5インチ2DD拡張ドライブ2 | 6:     |

また、本体にディスク装置が内蔵されており、さらに同型タイプの外付けディスク装置が接続されている場合、初期状態では、



の順でドライブ番号が割り当てられますが、ディップスイッチSW1の4番をONにしますと、拡張ドライブが優先されます。

本体にディスク装置が内蔵されていない場合

次の順で番号が割り当てられます。

システムディスクと同じ種類のディスクを使用する  
外付けディスク装置の拡張ドライブ



システムと別の種類のディスクを使用する  
外付けディスク装置の拡張ドライブ

例) 5 インチ 2HD の拡張ドライブ(2 台)と 3.5 インチ 2DD の拡張ドライブ(2 台)を外付けして、3.5 インチ 2DD ドライブから BASIC を起動させたとき。

| ドライブ                 | ドライブ番号 |
|----------------------|--------|
| 3.5 インチ 2DD 拡張ドライブ 1 | 1:     |
| 3.5 インチ 2DD 拡張ドライブ 2 | 2:     |
| 5 インチ 2HD 拡張ドライブ 1   | 3:     |
| 5 インチ 2HD 拡張ドライブ 2   | 4:     |

**注意：**拡張ドライブが同じ種類の場合には、接続時にドライブ番号の優先順位が決まります。

なお、このルールはシステムディスクの標準として定められているもので、ユーザーは必要に応じてフロッピーディスクのドライブ番号の割り振り順を変更することが可能です。このためには、ユーティリティ "format.nip", "sysgen.nip" を使ってシステムディスクを作成する場合にドライブアロケーションタイプの設定を行います。

▶ 参照 ユーティリティ→「第 10 章 ユーティリティプログラム」  
ドライブアロケーションタイプ→10.3の「"format.nip" の使い方」

#### ●固定ディスク装置が接続(内蔵)されている場合

固定ディスクが接続されている場合、固定ディスクを効率よく使用するために2とおりのドライブ番号の割り当て方が用意されています。この設定はユーティリティ "switch.n88" で行います。

これにより、フロッピーディスク装置と固定ディスク装置のうち、どちらのドライブが優先指定されるのかを選ぶことができます。

▶ 参照 ドライブ番号の優先指定→6.4の「固定ディスクからの BASIC の起動」  
switch.n88 →「10.14 メモリスイッチの設定」

なお、初期状態(出荷時)には「フロッピーディスク→固定ディスク」に設定されています。

## 6.6 BASIC 命令によるディスクファイルの操作

N<sub>88</sub>-日本語 BASIC(86)にはディスクファイルを扱うための命令が数多く用意されています。これらの命令を使うことによって、作成したプログラムや文書データなどをファイルとして保存し、必要に応じて呼び出して使用するといったことが自由にできるようになります。

ここでは BASIC の命令を使ったファイル操作の方法について解説します。

### ■ ディスク中のファイル情報のチェック

ディスクにはたくさんのファイルを記録しておくことができます。ファイルの操作をする場合にはディスクにどのようなファイルが入っているかを常に知っておく必要があります。

ディスクの中に入っているファイルの情報を調べるには FILES 命令を使います。

#### FILES 命令

FILES [〈ドライブ番号〉]

〈ドライブ番号〉のうしろにコロン(:)をつける必要はありません。

〈ドライブ番号〉を省略した場合はドライブ1が対象となります。

例) ドライブ2の中に入っているファイルの情報を調べる。

```
FILES 2 
SAMP1 * BIN 10      SAMP2 .BAS 2      text      txt
```

FILES 命令を実行すると、ドライブ中のすべてのファイルについて、主ファイル名と拡張子、およびそのサイズ(大きさ)が画面に表示されます。サイズはクラスタ単位です。主ファイル名と拡張子の間の区切り記号は、次のようにファイルの種類を表しています。

空白               : アスキーセーブされたプログラムファイルまたは OPEN 命令によって作られたデータファイル

ピリオド(. )       : バイナリセーブされたプログラムファイル

アスタリスク(\*) : BSAVE された機械語ファイル

また、FILES の代わりに LFILES を使用すると、情報はプリンタに出力されます。

▶ 参照 クラスタ→「付録C ディスクの内部構造」

## ■プログラムファイルのセーブとロード

BASIC を使ってプログラムを作成しても本体の電源を OFF にしたり、リセットしたりするとプログラムは失われます。そこで作成したプログラムをディスクに保存します。こうすれば、必要な時にディスクから取り出して使用することができます。プログラムをディスクに保存(セーブ)するための命令が SAVE 命令、ディスクから取り出す(ロード)ための命令を LOAD 命令といます。また、RUN 命令を使用してプログラムをロードして直ちに実行することもできます。

### ●セーブ(SAVE)

BASIC のエディタ機能を使用してプログラムを作成した状態では、プログラムはメモリの BASIC プログラム領域にデータとして存在しています。SAVE 命令を使うと、このメモリ上のデータをディスクに書き込むことができます。

#### アスキー形式とバイナリ形式

SAVE 命令ではプログラムをアスキー形式とバイナリ形式という 2 とおりの形式でセーブすることができます。

アスキー形式とはプログラムを LIST 命令で表示したときと同じイメージでテキストとしてセーブするものです。

バイナリ形式とはプログラムをメモリにある状態のままにディスクにセーブするものです。プログラムはメモリ上ではメモリの節約をするために中間コードという圧縮された状態で存在しており、バイナリ形式によるセーブを行うとメモリの内容がそのままの状態にディスクにセーブされることになります。

#### アスキー形式の利点

アスキー形式でセーブされたプログラムファイルはテキストファイルとして扱うことができます。したがって、OPEN 命令などを使ったファイルの入出力を行うことができます。また、MERGE 命令を使って 2 つのプログラムを合成することも可能になります。

#### バイナリ形式の利点

バイナリ形式でセーブすると、通常、アスキー形式でセーブしたものに比べてプログラムファイルの大きさが小さくなりますので、ディスクを無駄なく使用することができ、また、セーブ、ロードの時間も短縮されます。

#### SAVE 命令

```
SAVE <ファイルディスクリプタ> [ , | A | ]
                               | P | ]
```

A オプションも P オプションもつけない場合はバイナリ形式でセーブされます。

A オプションをつけるとアスキー形式でセーブされます。P オプションをつけるとプログラムは暗号化されたバイナリ形式でセーブされます。この形式でセーブされたプログラムファイルは、ロードしても LIST 命令や EDIT 命令を実行することはできません。

---

**注意：**SAVE 命令を実行する際にすでにディスク上にあるファイルと同じファイルディレクトリを指定すると、ディスク上のファイルは失われてしまいますので注意してください。特に SCSI 固定ディスクで複数ドライブを使用している場合は、ドライブ番号の指定には注意が必要です。

---

### ●ロード (LOAD)

ディスク上のプログラムファイルを BASIC のメモリ領域にロードするには LOAD 命令を使用します。この命令を実行するとメモリ上にプログラムが読み込まれ、そのプログラムが実行可能な状態になります。

#### LOAD 命令

LOAD <ファイルディレクトリ> [, R]

R オプションをつけるとロード後ただちにプログラムを実行します。この場合、そのときオープンされていたファイルは閉じられず、オープンされたままで実行されます。

---

**注意：**LOAD 命令を実行すると、それまで BASIC のメモリ領域にあったプログラムは消滅してしまいます。LOAD 命令を実行する前にはメモリ上に重要なプログラムがないことを確認してください。

---

### ●ロード後実行 (RUN)

普通プログラムを実行するには RUN 命令を使用しますが、RUN 命令にはディスクからプログラムをロードして、直ちに実行させるという機能もあります。

#### RUN 命令

RUN [<ファイルディレクトリ>][, R]

R オプションをつけると、そのときオープンされていたファイルは閉じられず、オープンされたままで実行されます。

## ■プログラムのマージ

メモリ上にあるプログラムとディスクの中にあるプログラムを行単位に合成して、1つのプログラムにすることを「マージ」といいます。そのための命令がMERGE命令です。プログラムの改造を行う場合などに有効です。

### MERGE 命令

MERGE <ファイルディスクリプタ>

マージする際、2つのプログラムに同じ行番号があるとディスク上のプログラムが優先されて、メモリ上のプログラムのその行と置き変わります。

---

**注意：**MERGE命令でディスクから読み込むプログラムファイルは必ずアスキー形式でセーブされたものでなければなりません。

---

## ■プログラムの連結

機能が豊富なプログラムを作ろうとするとどうしても大きなものになります。その結果、大きくなりすぎてメモリに入りきれないとか、プログラムのバグが発生しやすいといったことがおこりがちです。

このような問題を解決するための方法としてプログラムのモジュール化があります。プログラムを単純な機能の基本単位(モジュール)にわけ、それを連結(チェーン)して実行させるのです。

モジュール化は次のような手順で行います。

- ①プログラムをいくつかの単純な機能を持ったモジュールに分けて作成する。
- ②それぞれのプログラムをディスクにセーブする。
- ③最初のモジュールを実行させる。
- ④その後、自動的に必要なモジュールを次々にメモリにロードして実行する。

④の自動実行は、前に説明したRUN命令やLOAD命令のRオプションを使って行うことができますが、それよりもっと便利な命令としてCHAIN命令が用意されています。

### CHAIN 命令

CHAIN [MERGE] <ファイルディスクリプタ> [, <行番号>] [, ALL] [, DELETE<範囲>]

CHAIN命令の各パラメータは次のような働きがあります。

### 〈ファイルディスクリプタ〉

〈ファイルディスクリプタ〉で指定したプログラムファイルをメモリにロードしてから実行します。

### 〈行番号〉

プログラムをメモリにロードした後、〈行番号〉で指定した行から実行します。〈行番号〉の指定を省略すると先頭の実行されます。

### ALL オプション

プログラムをメモリにロードした後、実行します。CHAIN 命令を実行する前のすべての変数の値はクリアされずそのまま新しいプログラムで使用できます。

ALL オプションを省略した場合には CHAIN 命令を実行する前のすべての変数の値はクリアされます。

なお、特定の変数のみを使えるようにするには、COMMON 命令を使用します。

▶参照 COMMON命令→『BASICリファレンスマニュアル』

### MERGE オプション

メモリ上のプログラムとディスク上のプログラムを合成し、実行します。合成する際、2つのプログラムに同じ行番号があるとディスク上のプログラムが優先されて、メモリ上のプログラムのその行と置き変わります。

---

注意：MERGE オプションを使ってディスクから読み込むプログラムファイルは必ずアスキー形式でセーブされたものでなければなりません。

---

### DELETE オプション

MERGE オプションを指定したときのみ指定することができます。これを指定すると、メモリ上のプログラムの中から DELETE オプションで指定した範囲を削除した後で合成を行い、実行します。

例 1) ドライブ 2 のディスクから "SAMPLE.BAS" をロードし、300 行から実行する。

```
CHAIN " 2 : SAMPLE.BAS", 300
```

例 2) メモリ上のプログラムの 200 行から 500 行を削除し、ドライブ 2 のディスクの "SAMPLE.BAS" をマージした後で実行する。CHAIN 命令を実行する前の変数の値はクリアされない。

```
CHAIN MERGE " 2 : SAMPLE.BAS", ALL, DELETE 200-500
```

## ■機械語ファイルのセーブとロード

コンピュータが動作しているとき、メモリはデータを一時的に保存する場所となります。BASICのテキストはメモリのテキストエリアに、機械語のプログラムは機械語プログラムエリアに、また画面の情報はVRAM(画面専用のメモリ)に保存されています。これらのデータは電源を切れば失われてしまいます。

BASICには、このメモリのデータをそのままのイメージでディスクファイルとして保存(セーブ)したり、ディスクから逆に取り出し(ロード)たりする命令が用意されています。

これがBSAVE命令とBLOAD命令です。この命令は特に機械語プログラムのセーブ、ロードによく使われます。

- ▶参照 機械語プログラム→「第8章 機械語プログラム」  
 テキストエリア、機械語プログラムエリア→「付録A メモリマップ」

### ●セーブ(BSAVE)

#### BSAVE 命令

**BSAVE** <ファイルディスクリプタ>, <開始アドレス>, <長さ>

<開始アドレス>を先頭として<長さ>で表される大きさのメモリのデータを<ファイルディスクリプタ>で指定されたファイル名で、バイナリファイルとしてセーブします。

このとき指定する<開始アドレス>はセグメント内の相対アドレスです。したがって、この命令を実行する前にはDEF SEG命令でセグメントベースを設定しなければなりません。

- ▶参照 セグメント、DEF SEG命令→「8.1 メモリレイアウトとセグメント」

### ●ロード(BLOAD)

#### BLOAD 命令

**BLOAD** <ファイルディスクリプタ>[, <ロードアドレス>][, R]

<ファイルディスクリプタ>で指定されたファイルをメモリ上にロードします。

<ロードアドレス>の指定を省略した場合は、0番地(セグメントベースの先頭アドレス)からロードが行われます。<ロードアドレス>を指定すると、ロードはそのアドレスから行われます。BLOAD命令のアドレス指定はセグメント内の相対アドレスです。したがって、この命令を実行する前にはDEF SEG命令でセグメントベースを設定しなければなりません。

なお、機械語ファイルをロードするメモリの領域がBASICの領域と重なっていると、ロードしたデータが壊されてしまったり、逆にBASICの領域を乱してしまったりするおそれがあります。したがって、事前にCLEAR命令を実行してロードする領域を機械語プログラム用に確保しておかなければなりません。

- ▶参照 CLEAR命令→「8.2 機械語プログラムの準備」



```
3)SET #<ファイル番号>, | "P" |
                        | "R" |
```

1)は指定したドライブに対して属性が付けられます。2)は指定したファイルに対してだけ属性が付けられます。また、3)は指定したファイル番号に対応するファイルバッファに対して属性が付けられます。

属性の指定には、“P”と“R”のオプションを使います。“P”を指定すると「書き込み禁止」の属性が付けられます。また、“R”を指定すると「書き込み確認」の属性が付けられます。“p”、“r”と小文字で指定してもかまいません。

「書き込み禁止」の属性がつけられるとそのドライブ、ファイル、ファイルバッファに対して出力モードでのOPENやランダムモード時のPUTなどの書き込み動作が禁止されるとともに、ファイルに対するKILL命令も実行できなくなります。

「書き込み確認」の属性がつけられると以後書き込みを行うたびに元のデータと書き込まれたデータのチェックを行い、間違いなく書き込みが行われているかどうかを確認します。この場合、書き込みに要する時間は長くなりますがデータの信頼性は高くなります。

属性を解除するには、“P”と“R”以外の文字をオプションとしてSET命令を実行してください。

### ■ファイル属性のチェック

ATTR\$関数を使うと、SET命令で付けられたファイルやドライブの属性を調べることができます。

#### ATTR\$関数

- 1)ATTR\$(<ドライブ番号>)
- 2)ATTR\$(<ファイルディスクリプタ>)
- 3)ATTR\$(# <ファイル番号>)

1)はドライブに付けられている属性を、2)はファイルに付けられている属性を、3)はファイルバッファの属性を得ることができます。

得られる値とその意味は次のとおりです。□は空白を意味します。

- “□□□” 属性が解除された状態です。
- “R□□” 「書き込み確認」の属性が付けられています。
- “□□P” 「書き込み禁止」の属性が付けられています。
- “□E□” PオプションつきでSAVEされたファイルです。

## ■ディスクのセクタへの直接アクセス

一般のファイル入出力命令では、ディスクの内容を読み込むのにファイル名やファイル番号を指定しますが、次の2つの関数・命令では、ディスクのアドレスを指定します。アドレスを間違えて指定するとディスクを壊してしまう危険があります。したがって、これを実行する場合は使用するディスクのバックアップをとっておくようにしてください。

### DSKI\$関数

DSKI\$ (〈ドライブ番号〉, 〈サーフェス番号〉, 〈トラック番号〉, 〈セクタ番号〉)

このようにアドレスを指定すると指定した1セクタ256バイトのデータが0番のファイルバッファ(#0)に読み込まれます。バッファを使用するため、FIELD命令を併せて実行しなければなりません。なお、この場合OPEN命令を実行する必要はありません。

```
例) 10 FIELD #0, 128 AS A$, 128 AS B$
     20 D$=DSKI$(2, 1, 39, 16)
```

この例ではドライブ番号2, サーフェス番号1, トラック番号39, セクタ番号16のデータの前半の128バイトがA\$に, 後半の128バイトがB\$に代入されます。

文字変数の最大の長さは255バイトであるため、D\$にはセクタの情報が完全に代入されませんが、10行と20行を合わせて実行することによって、1セクタ分のデータをA\$, B\$の2つの文字変数に分けて得ることができます。

### DSKO\$命令

DSKO\$ (〈ドライブ番号〉, 〈サーフェス番号〉, 〈トラック番号〉, 〈セクタ番号〉)

DSKI\$関数と反対に256バイトの値を指定するセクタに書き出します。

この場合も、ファイルバッファ(#0)を使用し、LSET/RSET命令を使ってバッファにデータを書き込んだ後、セクタに書き出します。

```
例) 10 FIELD #0 128 AS C$, 128 AS D$
     20 LSET C$=SAMPLE1$
     30 RSET D$=SAMPLE2$
     40 DSKO$ 2, 1, 39, 16
```

この例では、セクタの前半にSAMPLE1\$, 後半にSAMPLE2\$を書き出します。

▶ 参照 FIELD, LSET/RSET命令 → 『BASIC リファレンスマニュアル』

DSKI\$ 関数, DSKO\$ 命令, サーフェス, トラック, セクタ → 「付録C ディスクの内部構造」

注意：SCSI 固定ディスクの場合、DSKI \$ 関数、DSKO \$ 命令で指定するトラック番号は各ドライブの相対的なトラック番号です。

## 6.7 データファイルの扱い方

データファイルにはシーケンシャルファイルとランダムファイルの2種類があります。

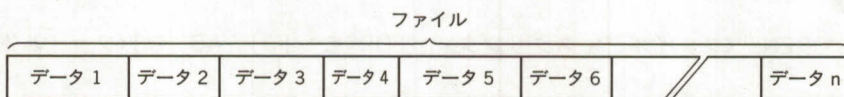
### ■シーケンシャルファイルとランダムファイル

#### ●シーケンシャルファイル

シーケンシャルファイルは連続して書かれたデータの集まりで次のような特徴があります。

- 読み書きするデータの長さは自由に変更できる。
- ファイル操作が簡単である。
- データの読み書きは必ず先頭からしかできない。データを捜すには先頭から最後まで捜さなければならないので時間がかかる。
- ファイル内の一部のデータだけを変更することができない。変更するときはファイル全部を更新するほかない。
- データの追加はファイルの終わりに書き込むことしかできない。

#### シーケンシャルファイルの構造



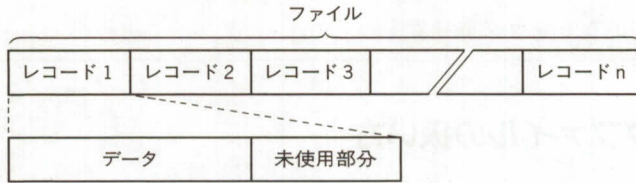
→ データを読んで行く方向

#### ●ランダムファイル

いっぽう、ランダムファイルはある一定の大きさのカードにデータを書き込んだものを並べたようなもので、次のような特徴があります。

- ファイル内のデータを自由に何番目のものでも自由に読み書きすることができるので、必要なデータだけを即座に取り出すことができる。
- ファイル内の特定のデータだけを変更することができる。
- シーケンシャルファイルに比べて、プログラムがやや複雑になる。
- 読み書きするデータの大きさは固定されている。

### ランダムファイルの構造



データはレコード(256バイト)を単位にして読み書きされる。

シーケンシャルファイルもランダムファイルもそれぞれ長所と短所がありますので、その違いを十分理解して処理の内容にあったファイルを作成する必要があります。

### ■シーケンシャルファイルの操作

まず比較的操作が簡単なシーケンシャルファイルから解説します。

シーケンシャルファイルの操作手順には基本的な形式があります。それは次のとおりです。

- ① OPEN 命令によってシーケンシャルファイルをオープンする。
- ② WRITE #, PRINT #などを使ってデータをファイルに書き出す。または INPUT #などを使用してデータをファイルから読み込む。
- ③ ファイルをクローズする。

各段階において使われる命令を簡単に説明します。

### ●ファイルをオープンする

#### OPEN 命令

OPEN <ファイルディスクリプタ> [FOR<モード>] AS [#]<ファイル番号>

OPEN 命令の機能として次の2つがあります。

#### ファイルにファイルバッファを割り当てる

<ファイルディスクリプタ>で指定されたファイルに<ファイル番号>で表される専用ファイルバッファを割り当てます。以後入出力操作を行うときにはこの<ファイル番号>を指定します。

#### ファイルのアクセスモードを設定する

<モード>はファイルのアクセスの仕方を指定するものです。シーケンシャルファイルの場合は次の3種類があります。

- INPUT .....入力モード。既存のファイルよりデータを読み込むことを示す。
- OUTPUT .....出力モード。新規にファイルを作り、書き込みを行うことを示す。
- APPEND .....追加モード。既存のファイルに追加書き込みを行うことを示す。

OPEN 命令を実行することによってファイルが入出力可能となります。

---

注意：入力モード、追加モードでオープンした場合はファイルがないとエラーになります。また、既にあるファイルを出力モードでオープンするとそのファイルは失われてしまいます。

---

●データを読み書きする

PRINT #, PRINT # USING, WRITE #命令

```
PRINT #<ファイル番号>, [<式>] [ | , | <式> ... ] [ | , | ]
                ; | ;
```

```
PRINT #<ファイル番号>, USING<書式制御文字列>; <式> [ | , | <式> ... ] [ | , | ]
                ; | ;
```

```
WRITE #<ファイル番号>, <式> [ | , | <式> ... ] [ | , | ]
                ; | ;
```

これらはファイルバッファへの書き出しを行う命令です。これらの命令を使って <ファイル番号> で指定されるファイルバッファへ <式> の値を書き込みます。ファイルバッファがいっぱいになったときにファイルへの書き込みが行われます。

基本的な機能は3つとも同じです。それぞれの命令の違いについては「BASIC リファレンスマニュアル」を参考にしてください。なお、これらの命令はファイルが出力モードあるいは追加モードでオープンされていなければ使用できません。

INPUT #, LINE INPUT #命令, および INPUT \$ 関数

```
INPUT #<ファイル番号>, <変数名>[, <変数名>...]
```

```
LINE INPUT #<ファイル番号>, <文字型変数名>
```

```
INPUT $ (<文字数>[, [#]<ファイル番号>))
```

ファイルバッファからの読み込みを行う命令・関数です。

これらの命令・関数を使って、ファイルバッファからデータを読み込むことができます。

基本的な機能は3つとも同じです。それぞれの命令の違いについては『BASICリファレンスマニュアル』を参考にしてください。なお、これらの命令はファイルが入力モードでオープンされていなければ使用できません。

そのほかにファイルの状態を調べるための命令・関数が用意されています。

## EOF 関数

### EOF(<ファイル番号>)

<ファイル番号>で指定されたファイルのデータを最後まで読み込んだら真(-1), そうでなくまだ残っているなら偽(0)を与えます。ファイルの読み込みが終わっているかどうかの判定に使用します。なお, この関数はファイルが入力モードでオープンされていなければ使用できません。

## LOC 関数

### LOC(<ファイル番号>)

<ファイル番号>で指定されたファイルをオープンしてから現在までにファイルの入出力をした回数を得られます。

## LOF 関数

### LOF(<ファイル番号>)

<ファイル番号>で指定されたファイルの大きさをセクタ数で得ます。

## FPOS 関数

### FPOS(<ファイル番号>)

<ファイル番号>で指定されたファイルの最後に読み書きしたデータのセクタ番号を得られます。

## ●ファイルをクローズする

### CLOSE 命令

CLOSE [[#]<ファイル番号>[, [#]<ファイル番号>]…]

CLOSE 命令には次の2つの機能があります。

#### ファイルバッファを解放する

ファイルに割り当てられていたファイルバッファを解放し, 別のファイルで使用できるようにします。

最後にファイルバッファに残ったデータをファイルに出力する(出力モードおよび追加モードのとき)

データの出力命令は実際にはファイルバッファへデータ出力を行い, ファイルバッファがいっぱいになったらファイルに書き出すようにしています。したがって, ファイルバッファにデータが残っている可能性があります。このデータをファイルに出力します。

## ■シーケンシャルファイルの例

シーケンシャルファイルの例として住所録を取り上げます。

### ●住所録の作成

キーボードから名前と住所と電話番号を入力してファイルを作成します。

---

```

100 OPEN "ADDRES.DAT" FOR OUTPUT AS #1 —— "ADDRES.DAT" を出力モードでオープンする。
110 INPUT "名前";A$ —— A$に名前を入力する。
120 IF A$="" THEN CLOSE : END —— 110行で[Enter]キーのみを押したら終了する。
130 INPUT "住所";B$ —— B$に住所を入力する。
140 INPUT "電話番号";C$ —— C$に電話番号を入力する。
150 PRINT #1,A$;",";B$;",";C$ —— データをファイルに書き出す。
160 PRINT —— 改行する。
170 GOTO 110 —— 110行へもどる。

```

---

この場合、漢字でデータを入力するとファイルには次のような形式でデータが出力されます。

### 書き出されるデータの例

|                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 文字および<br>制御コード | KI | 鈴  | 木  | 太  | 郎  | KO | ,  | KI | 東  | 京  | 都  | KO | ,  | 0  | 3  | -  | 1  | 1  | 7  | -  | 4  | 3  | 2  | 1  | CR | LF |    |    |    |    |    |    |    |    |    |    |    |
| 16進表記          | 1B | 4B | 4E | 6B | 4C | 5A | 42 | 40 | 4F | 3A | 1B | 4B | 2C | 1B | 4B | 45 | 6C | 35 | 7E | 45 | 54 | 1B | 4B | 2C | 30 | 33 | 2D | 31 | 31 | 37 | 2D | 34 | 33 | 32 | 31 | 0D | 0A |

### ●住所録の画面表示

住所録の中身をすべて画面表示します。

---

```

100 OPEN "ADDRES.DAT" FOR INPUT AS #1 —— "ADDRES.DAT" を入力モードでオープンする。
110 IF EOF(1) THEN CLOSE : END —— ファイルの中身がなくなったら終了する。
120 INPUT #1,A$,B$,C$ —— ファイルからデータを読み込む。
130 PRINT "名前:";A$ —— 名前を表示する。
140 PRINT "住所:";B$ —— 住所を表示する。
150 PRINT "電話:";C$ —— 電話番号を表示する。
160 PRINT —— 改行する。
170 GOTO 110 —— 110行へもどる。

```

---

### ●住所録の検索

住所録の中から特定の文字、数字を含む人のデータを表示します。

---

```

100 INPUT " 捜す文字は ";F$: PRINT _____ 検索する文字を入力する。
110 OPEN "ADDRES.DAT" FOR INPUT AS #1 _____ "ADDRES.DAT"を入力モードでオープンする。
120 IF EOF(1) THEN CLOSE : END _____ ファイルの中身がなくなったら終了する。
130 INPUT #1,A$,B$,C$ _____ データをファイルから読み込む。
140 IF KINSTR(A$+B$+C$,F$)=0 THEN 120 _____ データの中に検索する文字がなかったら
150 PRINT " 名前 ";A$ _____ 名前を表示する。 120行へもどる。
160 PRINT " 住所 ";B$ _____ 住所を表示する。
170 PRINT " 電話 ";C$ _____ 電話番号を表示する。
180 PRINT _____ 改行する。
190 GOTO 120 _____ 120行へもどる。
    
```

---

### ●住所録の追加

住所録にデータを追加します。

---

```

100 OPEN "ADDRES.DAT" FOR APPEND AS #1 _____ "ADDRES.DAT"を追加モードでオープンする。
110 INPUT " 名前 ";A$ _____ A$に名前を入力する。
120 IF A$="" THEN CLOSE : END _____ 110行で[Enter]キーのみを押したら終了する。
130 INPUT " 住所 ";B$ _____ B$に住所を入力する。
140 INPUT " 電話番号 ";C$ _____ C$に電話番号を入力する。
150 PRINT #1,A$," ";B$," ";C$ _____ データをファイルに書き出す。
160 PRINT _____ 改行する。
170 GOTO 110 _____ 110行へもどる。
    
```

---

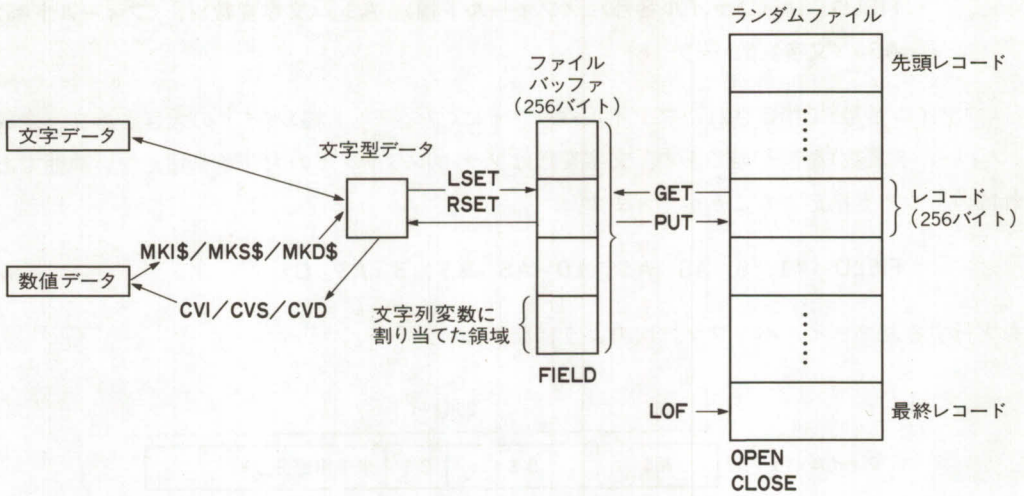
### ■ランダムファイルの操作

ランダムファイルを操作する手順は次のようになります。

- ① OPEN 命令を使ってランダムファイルをオープンする。
- ② FIELD 命令を使ってファイルバッファに文字変数(フィールド変数)を割り当てる。
- ③ ファイルへの書き出しの場合は、LSET/RSET 命令を使ってバッファにデータをセットし、PUT 命令によってファイルへ書き出す。
- ④ ファイルからの読み込みの場合は、GET 命令によってデータをバッファに読み込み、バッファに割り当てられた文字変数からデータを取り出す。
- ⑤ ファイルをクローズする。

なお、③の操作において数値データを扱う場合は、前もって数値データを文字型データに変換(MKI\$, MKS\$, MKD\$関数)してからバッファにセットします。またこのような変換を行ったデータに対しては、④の操作においてデータを取り出したあとで文字型データを数値データにもどします(CVI, CVS, CVD関数)。

## ランダムファイルのデータの受け渡し



各段階において使われる命令・関数を簡単に解説します。

### ● ファイルをオープンする

#### OPEN 命令

OPEN <ファイルディスクリプタ> AS [#]<ファイル番号>

ランダムファイルではシーケンシャルファイルのようなモードの指定はありません。<ファイルディスクリプタ>で指定したファイルの入出力のために<ファイル番号>のファイルバッファが割り当てられ、以後、<ファイル番号>を指定してデータの入力・出力の両方ができるようになります。

### ● フィールド変数を割り当てる

シーケンシャルファイルでは読み書きするデータの長さが自由に設定できましたが、ランダムファイルではデータの大きさは256バイトに固定されています。これを1レコードといいます。

1レコードの大きさはファイルバッファと同じですから、一度にファイルバッファの大きさの分だけデータの受け渡しができるということになります。

また、ファイルバッファの領域をいくつかの領域に分けることにより、複数のデータの受け渡しを一度に行うことができます。

FIELD 命令はファイルバッファに複数のフィールド変数の領域を割り当てます。その結果この変数を使って、プログラムとファイルバッファ間のデータの入出力を行うことができます。

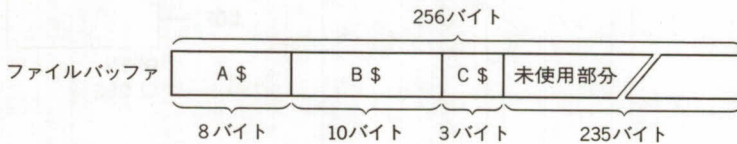
FIELD 命令

FIELD [#]〈ファイル番号〉, 〈フィールド幅〉 AS 〈文字変数〉[, 〈フィールド幅〉 AS 〈文字変数〉…]

〈ファイル番号〉で指定されたファイルバッファに〈フィールド幅〉バイトの大きさの文字変数(フィールド変数)を割り当てます。文字変数はファイルバッファの大きさを超えない範囲であればいくつでも指定することができます。

FIELD #1, 8 AS A\$, 10 AS B\$, 3 AS C\$

を実行するとファイルバッファは次のようになります。



FIELD 命令によるフィールド変数の割り当ては1つのファイルバッファに対して何回でも行うことができます。これらの定義はすべてが有効になります。

●ファイルバッファにデータをセットしてファイルへ書き出す

ファイルに対する出力を行うには、まず、ファイルバッファにデータをセットします。

ファイルバッファへのデータのセットは、LSET/RSET 命令を使って、ファイルバッファに割り当てたフィールド変数に文字列データを代入するという方法で行います。

LSET/RSET 命令

LSET 〈文字変数〉=〈文字列〉

RSET 〈文字変数〉=〈文字列〉

〈文字列〉の文字数が〈文字変数〉(FIELD 命令で割り当てられたフィールド変数)の長さより短い場合は、LSET では左詰め、RSET では右詰めとなります。

PUT 命令

PUT [#]〈ファイル番号〉[, 〈数式〉]

〈ファイル番号〉で指定されたファイルバッファの内容を対応するファイルに書き出します。ここで初めてディスクファイルへの書き出しが行われます。〈数式〉にはファイル中のレコード番号を指定します。レコード番号の最小値は1, 最大値は65000です。PUT 命令を実行するとディスク上のあるセクタ(256 バイト)の領域にデータが書き出され、各々にレコード番号がふられ

ることになります。この番号はデータを GET 命令で参照したり、PUT 命令で変更したりする場合の目印となるものです。レコード番号に大きな数を指定するとその数の分だけのレコードが設定されます。たとえば最初に 300 と指定して PUT 命令を実行すると、ディスクに 1~300 までのレコード番号が割り振られ、その大きさの領域がランダムファイルとして確保されます。この場合 1~299 までのレコードにはデータは書き込まれていませんので、1~299 までのレコードを GET 命令で参照した場合のデータは意味のないものとなります。

レコード番号の指定を省略した場合は、直前に行われた PUT 命令あるいは GET 命令で使われたレコードの次のレコードに書き込みが行われます。

---

**注意：**レコード番号の最大値は 65000 ですが、ランダムファイル以外にディスクを使用している場合には、その領域にはレコードを作ることができませんので、実際にレコード番号として指定できる値はより小さくなります。レコード番号として指定できる限界値を超えて指定を行うとエラーが出ます。

---

#### ●ファイルからデータをバッファに読み込み、バッファからデータを取り出す

##### GET 命令

GET [#]〈ファイル番号〉[, 〈数値〉]

ファイルに書き込まれているデータを〈ファイル番号〉で指定されたファイルバッファに読み込みます。〈数値〉には、読み込みたいレコード番号を指定します。レコード番号の指定を省略した場合は直前に行われた PUT 命令あるいは GET 命令で使われたレコードの次のレコードから読み込みが行われます。

バッファに割り当てられたフィールド変数にデータが代入されます。

#### ●ファイルをクローズする

ランダムファイルのクローズも CLOSE 命令を使用します。

##### CLOSE 命令

CLOSE [[#]〈ファイル番号〉[, [#]〈ファイル番号〉]…]

CLOSE 命令を実行するとファイルに割り当てられていたファイルバッファが解放され、別のファイルで使用できるようになります。

●その他の命令, 関数

LOF 関数

LOF(<ファイル番号>)

<ファイル番号>で指定されるファイルの最大のレコード番号, つまりファイルの大きさをレコード数で得ることができる関数です。LOC 関数と併用してファイルが最後まで読み込まれているかの判定を行うことができます。

LOC 関数

LOC(<ファイル番号>)

<ファイル番号>で指定されるファイルにおいて, そのファイルを最後に読み書きしたレコード番号を得ることができます。前の例の LOF 関数と併用してファイルが最後まで読み込まれたかどうかの判定をすることができます。

例) 最後に読み込まれたレコード番号が最後のレコード番号と同じであればファイルをクローズする。

```
IF LOC(1)=LOF(1) THEN CLOSE #1
```

FPOS 関数

FPOS 関数(<ファイル番号>)

<ファイル番号>で指定されたファイルの最後に読み書きしたデータのセクタ番号が得られます。

MKI \$ /MKS \$ /MKD \$ 関数

MKI \$ (<整数値>) ..... 整数値を 2 文字(2 バイト)の文字列に変換

MKS \$ (<単精度実数値>) ..... 単精度実数値を 4 文字(4 バイト)の文字列に変換

MKD \$ (<倍精度実数値>) ..... 倍精度実数値を 8 文字(8 バイト)の文字列に変換

ランダムファイルへの書き込みは, 文字列で行わなければなりません。したがって数値を書き込みたいときには文字列に変換してから LSET/RSET 命令でフィールド変数に代入するようにします。数値の種類によって文字列の長さが変わります。

CVI/CVS/CVD 関数

CVI(<2 文字の文字列>) ..... 2 文字の文字列を整数値に変換

CVS(<4 文字の文字列>) ..... 4 文字の文字列を単精度実数値に変換

CVD(<8 文字の文字列>) ..... 8 文字の文字列を倍精度実数値に変換

ランダムファイルの書き込みの際に MKI \$ /MKS \$ /MKD \$ 関数で変換された文字列を, 読み込みの際に元の数値にもどすために使用します。

## ■ランダムファイルの例

ランダムファイルの例として名簿を取り上げます。

### ●名簿の作成

キーボードから名前, 出身地, 年齢, 登録番号を入力してランダムファイルを作成します。この例では登録番号がレコード番号と同じ数値になっています。

---

```

100 OPEN "MEIBO.DAT" AS #1 —— "MEIBO.DAT" をオープンする。
110 FIELD #1,30 AS AA$,16 AS BB$,2 AS CC$,4 AS NOS —— ファイルバッファにフィールド
120 *NEXTP —— 変数を割り当てる。
130 PRINT —— 改行する。
140 INPUT "名前" ;AS —— 名前を入力する。
150 IF AS="" THEN CLOSE : END —— 140行で[Enter]キーだけが押されたら終了する。
160 INPUT "出身地" ;B$ —— 出身地を入力する。
170 INPUT "年齢" ;C% —— 年齢を入力する。
180 INPUT "登録番号";N —— 登録番号を入力する。
190 LSET AA$=AS —— データをバッファにセットする。
200 LSET BB$=B$
210 LSET CC$=MKI$(C%)
220 LSET NOS=MKS$(N)
230 PUT #1,N —— データをファイルに書き出す。
240 GOTO *NEXTP —— 120行へもどる。

```

---

### ●名簿の表示

登録番号を指定してデータを画面に表示させます。

---

```

100 OPEN "MEIBO.DAT" AS #1 —— "MEIBO.DAT" をオープンする。
110 FIELD #1,30 AS AA$,16 AS BB$,2 AS CC$,4 AS NOS —— ファイルバッファにフィールド
120 *NEXTP —— 変数を割り当てる。
130 PRINT —— 改行する。
140 INPUT "登録番号は";N$ —— 登録番号を入力する。
150 IF N$="" THEN CLOSE : END —— 140行で[Enter]キーのみが押された場合には終了する。
160 IF VAL(N$)>LOF(1) THEN *NEXTP —— 登録番号が登録されているデータの最大レコード番号
170 GET #1,VAL(N$) —— データをファイルから より大きい場合は 120行へもどる
180 PRINT —— ファイルバッファへ読み込む。
190 PRINT " 登録番号:";CVS(NOS) —— データを表示する。
200 PRINT " 名前" ;AA$
210 PRINT " 出身地" ;BB$
220 PRINT " 年齢" ;CVI(CC$)
230 GOTO *NEXTP —— 120行へもどる。

```

---

## 6.8 RS-232C 回線ファイルの扱い方

パーソナルコンピュータは、通常ではスタンドアローン(単独)の状態で使用しますが、他のコンピュータと接続してコンピュータ間でデータ交換を行ったり、相互に制御しあったりすることができます。また、外部機器を接続してその機器を制御するといったこともできます。

コンピュータを他のコンピュータや外部装置と接続することにより、単独では実現できなかったことが可能となるわけです。

コンピュータ同士あるいはコンピュータと外部機器と接続してデータ交換を行う際に、一般的に利用されているのがRS-232C回線です。PC-9800シリーズのすべての機種には、このRS-232C回線を利用するためのRS-232Cインタフェースが内蔵されています。

RS-232C回線はBASICを使って制御することができます。そしてこの場合、BASICはRS-232C回線をファイルとして扱います。このファイルを“RS-232C回線ファイル”と呼びます。

ここではBASICによるRS-232C回線ファイルの基本的な扱い方について解説します。

### ■ RS-232C 回線について

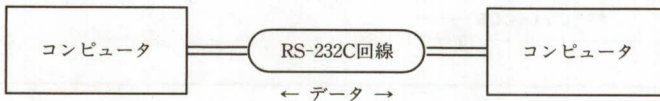
#### ● RS-232C インタフェース

RS-232CインタフェースはRS-232C規格に準拠したコネクタです。RS-232Cインタフェースは大部分のパーソナルコンピュータに内蔵されており、コンピュータと他のコンピュータや周辺機器をつないでデータ通信を行うための標準的なインタフェースとなっています。

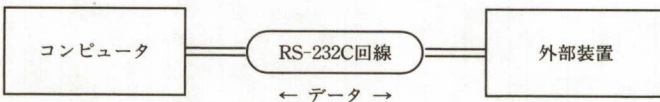
#### ● RS-232C 回線を使ったデータ通信

RS-232C回線を使ったデータ通信には、次の2つのパターンがあります。

#### コンピュータとコンピュータの間のコミュニケーション



#### コンピュータによる機器の制御



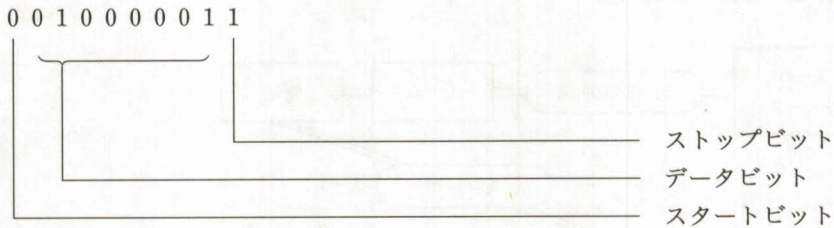
### ● RS-232C 回線のデータ形式

RS-232C インタフェースで接続した回線を RS-232C 回線といいます。

RS-232C 回線ではデータは1ビットずつ送られます。このようなデータ伝送形式をシリアル方式といいます。たとえば“A”というキャラクタを送る場合には、“A”のキャラクタコード(“01000001”)を1ビットずつ順に送ります。

さらに、データを送る場合にはデータの区切りを示すために、データの前にスタートビット、あとにストップビットを入れます。また、データビット長を7ビットにした場合にはデータ伝送の誤りをチェックするためにパリティビットなどを使用することもあります。

データ形式の例 (“A”というキャラクタをデータビット長8ビットで送る場合の例)



#### スタートビット

データの開始を知らせるビットです。普通は0になっています。

#### データビット

キャラクタのデータを表すビットです。

#### ストップビット

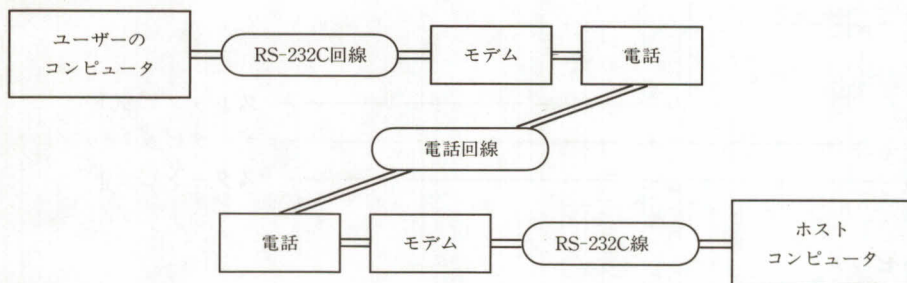
ストップビットは1つのキャラクタデータの最後に付けてデータの終わりを示すものです。

●コンピュータとコンピュータの間のコミュニケーション

RS-232C 回線では異なった機種種のコンピュータ同士を接続してデータの伝送を行うことができます。このとき、いっぽうのコンピュータをホストコンピュータ、もういっぽうをターミナル(端末)として使用することもできます。たとえば、端末側で命令を入力し、その命令をホスト側で実行して、その結果をターミナル側に表示するといったことが可能です。

また、モデムと電話回線を介することによりコンピュータと PC-VAN などのネットワークのホストコンピュータを接続して通信が行えます。この場合、RS-232C 回線でコンピュータに直接接続されているのは外部機器(モデム)ですが、コンピュータ間でデータ通信が行われていることになります。

パソコン通信の場合の接続状態



なお、モデムを介したデータ通信を BASIC で行うために、モデムや電話を制御するための拡張 BASIC 命令が用意されています。この拡張命令を“電話制御命令”といいます。

電話制御命令の使用方法については、「9.1 電話制御機能」を参照してください。

---

注意：RS-232C 回線を使用したデータ通信はターミナルモードを使っても行うことができます。

---

▶参照 ターミナルモード→「11.1 ターミナルモード」

電話制御命令→「9.1 電話制御機能」

### ●コンピュータによる機器の制御

RS-232C 回線を使ってさまざまな機器をコンピュータで制御することができます。

一般によく使用される機器にイメージスキャナ、プロッタなどがあります。

機器の制御は、普通、RS-232C 回線を通じてコンピュータと機器の間でデータを取り交わすことによって行います。

なお、制御方法は、周辺機器の種類によって千差万別です。したがって、ここですべてを解説することはできません。実際に使用する機器の仕様にしたがってプログラムを作成してください。なお、RS-232C 回線を利用した機器の制御の一例として、この項の終わりにイメージスキャナを制御するプログラムがありますので参考にしてください。

### ■ BASIC による RS-232C 回線の制御

RS-232C 回線ファイルの入出力を行うときには、他のデバイスファイルの場合と同じ命令を使用します。しかし、同じ命令であっても他のデバイスファイルの場合と比べて、引数のとり方、意味などが異なることがありますので注意してください。また、RS-232C 回線ファイルの入出力だけに使われる命令もあります。

#### ●データ通信

次のような順序でデータ通信を行います。

- ① OPEN 命令で RS-232C 回線ファイルをオープンし、通信プロトコルの設定を行う。
- ② PRINT #命令、WRITE #命令などで、データをファイルに書き出す(送信)。または、INPUT #命令などを使用してデータを読み込む(受信)。
- ③ CLOSE 命令で、RS-232C 回線ファイルをクローズする。

#### ●RS-232C 回線ファイルのオープンと通信プロトコルの設定

RS-232C 回線ファイルを使用する場合には、まず、RS-232C 回線ファイルをオープンします。

RS-232C 回線ファイルは同時に入出力が可能ですから、入出力のモードを指定する必要はありません。

OPEN 命令ではファイルのオープンと同時に通信プロトコルの設定を行います。通信プロトコルとは、通信する際の手順(規約)のことです。通信を行う機器は必ず通信プロトコルを同じに設定しておかなければ、正しくデータを送受信することができません。

通信プロトコルには、通信速度、データの形式、データチェックの方法などがあります。

### OPEN 命令

```
OPEN "COM[<回線番号>]:[<パリティチェック>[<データビット長>[<ストップ  
ビット長>[<Xパラメータ>[<Sパラメータ>]]]]]" AS [#]<ファイル番号>
```

指定する各パラメータの値とその意味は次のとおりです。

| パラメータ    | 指定値               | 意 味  |
|----------|-------------------|--|
| 回線番号     | 1 または省略<br>2<br>3 | RS-232C 第 1 回線<br>RS-232C 第 2 回線 (RS-232C 拡張インタフェースボードが必要)<br>RS-232C 第 3 回線 (RS-232C 拡張インタフェースボードが必要) |
| パリティチェック | E<br>O<br>N       | 偶数<br>奇数<br>使わない   |
| データビット長  | 7<br>8            | 7 ビット<br>8 ビット   |
| ストップビット長 | 1<br>2<br>3       | 1 ビット<br>1.5 ビット<br>2 ビット  |
| X パラメータ  | X<br>N            | 有効 (受信データのフロー制御を行う)<br>無効 (受信データのフロー制御を行わない)   |
| S パラメータ  | S<br>N            | 有効 (データビット長 7 でカナを扱う)<br>無効 (データビット長 7 でカナを扱わない)   |

回線番号以外のパラメータは通信プロトコルを設定するものです。

それぞれのパラメータについて解説します。

### 回線番号

回線番号には使用する RS-232C インタフェースの番号を指定します。コンピュータ本体に内蔵されているインタフェースを使用する場合は回線番号に 1 を指定 (または省略) します。

回線番号 2 あるいは 3 を使用する場合は、RS-232C 拡張インタフェースボードが必要です。この場合さらに、ユーティリティ "switch.n88" を使用し、拡張ボードの選択画面で RS-232C 拡張ボードを使用するように設定してください。

▶ 参照 switch.n88 → 「10.14 メモリスイッチの設定」

### パリティチェック

パリティチェックはデータビット長が7の場合に通信の誤り発見するために使われるものです。

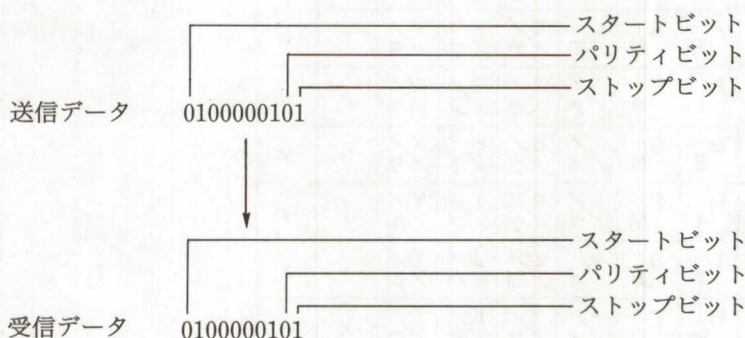
パリティチェックを有効にした場合は、送信側は1キャラクタごとに1ビットのパリティビットを付け加えて送信します。

パリティビットは間違いチェック用のビットで、“偶数パリティ”と“奇数パリティ”の2つの方式があります。

偶数パリティの場合はキャラクタコードを2進数表記したときの1のビットが偶数個になるようにパリティビットが付けられます。奇数パリティの場合は1のビットが奇数個になるようにパリティビットが付けられます。

データビットとパリティビットを含めた各ビットの1の数が偶数個、あるいは、奇数個あるかどうかでデータが正しく送られているかどうかをチェックします。

“A” (キャラクタコードの2進表記で“1000001”)を偶数パリティで送信する場合



この例では、1の数が偶数なのでデータが正しく受信できたことが分かります。もし、データのビットの1つが通信の途中で反転した(1が0に、あるいは0が1になった)場合には、1の数は奇数になるので、受信側はデータが正しく伝送されていないことが分かります。これをパリティエラーといいます。

### データビット長

データビット長とは伝送するキャラクタのビット数のことです。つまりキャラクタとして7ビットコードを使用するか、8ビットコードを使用するかを選択するものです。

8ビットコードはBASICで普通に使われているコードで、英数文字、カタカナなどを含む $2^8=256$ 種類のキャラクタを扱うことができます。BASICで通信を行う場合には普通8ビットコードを使用します。

7ビットコードは普通、次の表のように $2^7=128$ 種類のキャラクタしか扱うことができません。

7ビットコード表

上位4ビット→

|         | 0        | 1              | 2                   | 3   | 4   | 5   | 6   | 7 |   |
|---------|----------|----------------|---------------------|-----|-----|-----|-----|---|---|
| 下位4ビット↓ | <b>0</b> |                | D <sub>E</sub> SP   | 0/_ | @/タ | P/ミ |     | p |   |
|         | <b>1</b> | S <sub>H</sub> | D <sub>1</sub> !/。  | 1/ア | A/チ | Q/ム | a   | q |   |
|         | <b>2</b> | S <sub>X</sub> | D <sub>2</sub> "/   | 2/イ | B/ツ | R/メ | b   | r |   |
|         | <b>3</b> | E <sub>X</sub> | D <sub>3</sub> #/   | 3/ウ | C/テ | S/モ | c   | s |   |
|         | <b>4</b> | E <sub>T</sub> | D <sub>4</sub> \$/、 | 4/エ | D/ト | T/ヤ | d   | t |   |
|         | <b>5</b> | E <sub>Q</sub> | N <sub>K</sub> %/.  | 5/オ | E/ナ | U/ユ | e   | u |   |
|         | <b>6</b> | A <sub>K</sub> | S <sub>N</sub> &/ラ  | 6/カ | F/ニ | V/ヨ | f   | v |   |
|         | <b>7</b> | B <sub>L</sub> | E <sub>B</sub> '/ア  | 7/キ | G/ヌ | W/ラ | g   | w |   |
|         | <b>8</b> | B <sub>S</sub> | C <sub>N</sub> (/イ  | 8/ク | H/ネ | X/リ | h   | x |   |
|         | <b>9</b> | H <sub>T</sub> | E <sub>M</sub> )/ウ  | 9/ケ | I/ノ | Y/ル | i   | y |   |
|         | <b>A</b> | L <sub>F</sub> | S <sub>B</sub> */エ  | :/コ | J/ハ | Z/レ | j   | z |   |
|         | <b>B</b> | H <sub>M</sub> | E <sub>C</sub> +/オ  | ;/サ | K/ヒ | [/ロ | k   | { |   |
|         | <b>C</b> | C <sub>L</sub> | →                   | /ヤ  | </シ | L/フ | ¥/ワ | l |   |
|         | <b>D</b> | C <sub>R</sub> | ←                   | /ユ  | =/ス | M/ヘ | ]/ン | m | } |
|         | <b>E</b> | S <sub>O</sub> | ↑                   | /ヨ  | >/セ | N/ホ | ^/  | n | ~ |
|         | <b>F</b> | S <sub>I</sub> | ↓                   | /ツ  | ?/ソ | O/マ | /。  | o |   |

(注) SP は空白(スペース)コードを示します。

7ビットコードはおもに英数文字だけの通信に使われます。

ただし、7ビットコードでカナ文字を扱うことも可能です。それにはSパラメータを使用します。Sパラメータを使用した場合は、&H20~&H5Fまでのコードは英数文字とカナ文字の両方に割り当てられます。7ビットコードでSパラメータを“無効”にした場合にはカナ文字は使用できません。詳しくはSパラメータの項を参照してください。

## データビット長7ビットおよび8ビットで使用できるキャラクタ

| データビット長 | Sパラメータ | 英数 | カナ |
|---------|--------|----|----|
| 8ビット    | —      | ○  | ○  |
| 7ビット    | S(有効)  | ○  | ○  |
|         | N(無効)  | ○  | ×  |

○は使用可能, ×は使用不可能を表す

## ストップビット長

ストップビット長はストップビットの時間的長さを示すものです。ストップビットの時間的長さは、“1ビット”、“1.5ビット”、“2ビット”の中から選択できます。“1ビット”に設定するとデータビットの1ビット分の時間的長さと同じになります。

## Xパラメータ

データを送信する場合も受信する場合もデータはいったんバッファというメモリに貯められて処理されます。バッファの容量は一定量であるため、データがバッファに入る速度がバッファから出る(データを処理する)速度より速い場合には、データがバッファをあふれて失われてしまいます。これをデータの“オーバーフロー”といいます。

Xパラメータは受信バッファのオーバーフローを防ぐためのパラメータです。

Xパラメータを“有効”に設定すると、受信バッファ中のデータがバッファ域の3/4を超えると、受信側はデータの送信側に対して一時的にデータ送信をやめるようにD3(&H13)コードを送り、送信側はこれを受け取るとデータの送信を中断します。そして、受信バッファ中のデータ処理が進み、データがバッファ域の1/4以下になるとデータ送信を再開するようにD1(&H11)コードを送ります。このコードを受け取った送信側は送信を再開します。このような方法をとることにより、データのオーバーフローを防ぐことができます。これを“フロー制御”といいます。

Xパラメータを無効に設定すると受信バッファのフロー制御は行われません。

なお、BASICでは送信側はXパラメータの指定にかかわらずフロー制御を行いませんが、受信側からのD3コード(送信中止)あるいはD1コード(送信再開)を常に受け取ることができますので、必要に応じてプログラム内で制御してください。

## Sパラメータ

データビット長が7ビットの場合でも、英数文字とカナ文字の両方を送ることができるようにするためのパラメータがSパラメータです。

7ビットコードは英数文字のみを扱うコードであり、普通はカナ文字を表すことはできません。

そこで、英数文字とカナ文字の両方を使用するために、&H20～&H5F までのコードに英数文字とカナ文字を割り付け、シフトインコード(以下 SI コード)とシフトアウトコード(以下 SO コード)によって英数文字とカナ文字の区別をしています。

S パラメータを有効にした場合には、SI コードが送られた後の7ビットコードはカナ文字となり、SO コードに続く7ビットコードは英数文字となります。

なお、BASIC ではキャラクタを8ビットで扱っていますが、データビット長を"7"、S パラメータを"有効"に設定した場合には、送信側の BASIC が自動的に8ビットデータを7ビットデータに変換し、英数文字とカナ文字を区別できるように SI コードと SO コードを付加してデータを送信します。いっぽう、受信側では BASIC が送られてきた7ビットデータをもとの8ビットデータに自動的にもどします。

### OPEN 命令のパラメータを省略した場合

パラメータを省略して OPEN 命令を実行した場合は、メモリスイッチに設定されているプロトコルが有効になります。個々のパラメータの指定を省略するには指定する文字の代わりにスペースを入れてください。スペースを入れないで次のパラメータを詰めて記述すると間違いになります。

メモリスイッチによるプロトコルの設定は、ユーティリティ "switch.n88" を使って行うことができます。メモリスイッチの変更を一度行うと、BASIC を起動する度(ディップスイッチ SW2 の5番が ON のときのみ)にその状態に設定されます。

また、OPEN 命令による設定は、命令の実行後電源を切るか、リセットするか、OPEN 命令で設定をしないおすまで有効です。

### ● OPEN 命令で指定できない通信プロトコル

通信プロトコルのうち OPEN 命令では設定のできないものもあります。

これらについては、プログラムを実行する前に必ず、ユーティリティ "switch.n88" および、ディップスイッチで設定を行ってください。

### 通信プロトコルの設定項目と設定方法

| 設定項目        | 設定方法    |              |          |
|-------------|---------|--------------|----------|
|             | OPEN 命令 | "switch.n88" | ディップスイッチ |
| パリティチェック    | ○       | ○            | ×        |
| データビット長     | ○       | ○            | ×        |
| ストップビット長    | ○       | ○            | ×        |
| X パラメータ     | ○       | ○            | ×        |
| S パラメータ     | ○       | ○            | ×        |
| DEL コード受信処理 | ×       | ○            | ×        |
| ボーレート       | ×       | ○            | ×        |
| 同期クロックの設定   | ×       | ×            | ○        |

○は設定可能, ×は設定不可能を表す

### DEL コード受信処理

受信側が DEL コード (&H7F または &HFF) を受信した場合に、これを DEL コードとして処理するか NULL コード (&H00) として処理するかのどちらかを設定するものです。NULL コードとして処理するように設定すると、DEL コードが送られてきたときに受信側の BASIC はそのコードを自動的に NULL コードに変換します。

DEL コード受信処理の設定は、ユーティリティ "switch.n88" を使用して行います。

### ボーレート

ボーレートとは、通信速度のことです。1 秒当たり最大何ビットのデータを伝送できるかを表します。

75 ボーから 9600 ボーまでの中から選択することができます。

ボーレートの設定は、ユーティリティ "switch.n88" を使用して行います。

---

**注意：**ボーレートは 9600 ボーまで指定することができますが、通信相手のシステムの処理速度が遅い場合には、受信バッファが溢れてデータが失われてしまう場合があります。これは X パラメータを有効にした状態でもまれに起こる可能性があります。したがって、データがこぼれてしまうような場合には、ボーレートを下げるか、または、データの処理速度を速くするようにプログラム上で工夫を行ってください。

---

▶参照 switch.n88 →「10.14 メモリスイッチの設定」

### 同期クロックの設定

同期クロックとは、RS-232C インタフェースで接続した機器とコンピュータ本体との間でデータ通信を行う際のタイミングのとり方を、コンピュータ本体のタイミング(内部同期)で行うか、接続した機器のタイミング(外部同期)に合わせるかを定めるものです。

普通は内部同期にします。この選択にはディップスイッチ SW1 の 5, 6 を使用しますが、BASIC 使用時は常に OFF(内部同期)にしておいてください。

なお、外部同期を使用するためには、RS-232C シンクロナスモードで使用しなければなりません。しかし、BASIC では外部同期の通信機能を提供していませんので、これを行うには機械語の IN または OUT 命令を使用して RS-232C インタフェースの初期化を行い、さらに新しい手順、コマンドなども作成しなければなりません。

---

**注意：**通信プロトコルは、通信を行う相手と必ず同じに設定してください。さもないと正常なデータ伝送ができません。

---

●ファイルの入出力とファイルのクローズ

RS-232C 回線ファイルをオープンしたあとは、RS-232C 回線への入出力は以下の命令を使用して行います。

ここでは各命令の機能の概略を示します。詳しくは『BASICリファレンスマニュアル』を参照してください。

RS-232C 回線ファイルへデータを書き出す(送信する)命令

|                  |   |
|------------------|---|
| PRINT #命令        | データを出力(送信)します。  |
| PRINT # USING 命令 | データを編集して出力(送信)します。                                      |
| WRITE # 命令       | PRINT #命令と同様です。ただし、区切り記号としてコンマ(,)を必ず出力し、不要な空白の出力は行いません。 |

RS-232C 回線ファイルからデータを読み込む(受信する)命令

|                |   |
|----------------|---|
| INPUT #命令      | データを入力(受信)します。データの区切りは、読み込む変数が文字型の場合はコンマ(,), 改行コード(C <sub>R</sub> ), ダブルクォーテーション(" ")のいずれかです。非文字型の場合はコンマ(,), 改行コード(C <sub>R</sub> ), スペース(空白文字)のいずれかです。これらの区切りを受信するまでデータの入力を待ちます。 |
| LINE INPUT #命令 | データを入力(受信)します。データの区切りは改行コード(C <sub>R</sub> )だけです。改行コードを受信するまでデータの入力を待ちます。   |
| INPUT \$ 関数    | 指定されたバイト数だけデータを入力(受信)します。   |

RS-232C 回線ファイルの状態を調べる関数

|        |   |
|--------|---|
| EOF 関数 | 入力(受信)バッファが空であるかどうかを調べます。空の場合はその値は真(-1)となります。 |
| LOC 関数 | 入力(受信)バッファに貯っている文字数をバイト単位で得ます。                |
| LOF 関数 | 入力(受信)バッファの残りバイト数(入力可能な残りバイト数)を得ます。           |

**BASIC プログラムを送受信する命令**

- LOAD 命令**                    ファイル名に (COM1:~COM3:) を指定してこの命令を実行すると、RS-232C 回線から送られてきたプログラムを受信し、メモリにロードします。このとき、送信側のコンピュータでは SAVE 命令を実行しなければなりません。
- SAVE 命令**                    ファイル名に (COM1:~COM3:) を指定してこの命令を実行すると、RS-232C 回線にメモリ上のプログラムを出力(送信)します。

**機械語イメージのデータを送受信する命令**

- BLOAD 命令**                    ファイル名に (COM1:~COM3:) を指定してこの命令を実行すると、RS-232C 回線から送られてきた機械語イメージのデータを受信しメモリにロードします。
- BSAVE 命令**                    ファイル名に (COM1:~COM3:) を指定してこの命令を実行すると、RS-232C 回線にメモリ上の機械語イメージのデータを出力(送信)します。

---

**注意** : LOAD, SAVE, BLOAD, BSAVE の各命令の前に、RS-232C 回線ファイルをオープンする必要はありません。ただし、各命令のファイルディスクリプタに OPEN 命令と同じように回線番号や通信プロトコルを設定します。各パラメータを省略した場合はメモリスイッチに設定された値となります。

---

**ファイルをクローズする命令**

- CLOSE 命令**                    ファイルをクローズします。

**その他**

- WIDTH 命令**                    出力するデータの 1 行の桁数を決めます。ファイルに対するデータの出力(送信)時、指定された桁数の位置ごとに改行コード (C<sub>R</sub>) を出力(送信)することになります。

## ■ RS-232C 回線ファイルを使った基本プログラム

ここでは、BASIC で RS-232C 回線を使用して通信を行うプログラムを紹介します。

### ● キーボードから入力したデータを RS-232C 回線ファイルに出力する - [送信 1]

---

```
100 'SEND KYBD
110 OPEN "COM:" AS #1
120 LINE INPUT A$
130 IF A$="END" THEN PRINT #1,A$:CLOSE:END
140 PRINT #1,A$
150 GOTO 120
```

---

110 行では、RS-232C 回線ファイルをファイル番号 1 としてオープンしています。

120 行ではキーボードからの入力を待ち、入力されたデータを RS-232C 回線ファイルに出力 (送信) します。120 行で "END" という文字列が入力された場合には、130 行で "END" を RS-232C 回線ファイルに書き出し、その後、ファイルをクローズしてプログラムを終了させます。

### ● ディスクから入力したデータを RS-232C 回線ファイルに出力する - [送信 2]

---

```
100 'SEND DISK FILE
110 OPEN "COM:" AS #1
120 OPEN "TEST.DAT" FOR INPUT AS #2
130 IF EOF(2) THEN 170
140 LINE INPUT #2,A$
150 PRINT #1,A$
160 GOTO 130
170 PRINT #1,"END":CLOSE #1,#2:END
```

---

110 行で RS-232C 回線をファイル番号 1 としてオープンしています。

120 行では "TEST.DAT" というディスクファイルを入力モードでファイル番号 2 としてオープンしています。

130 行で "TEST.DAT" が空になった場合は 170 行へジャンプし、RS-232C 回線に "END" という文字を送った後で、2つのファイルをクローズして終了します。

140-150 行では "TEST.DAT" から 1 行ずつデータを入力し、RS-232C ファイルに書き出しています。

## ● RS-232C 回線ファイルから送られてきたデータを入力し、画面に表示する-[受信]

---

```

100 'RECEIVE
110 OPEN "COM:" AS #1
120 IF LOC(1)=0 THEN GOTO 120
130 LINE INPUT #1,A$
140 IF A$="END" THEN PRINT A$:GOTO 170
150 PRINT A$
160 GOTO 120
170 CLOSE:END

```

---

110行でRS-232C回線ファイルをオープンしています。

120行では受信バッファにデータがあるかどうかをチェックしています。データがない場合にはここで待ちます。

130行ではRS-232C回線ファイルからデータを受取り、150行でそのデータを画面に表示しています。読み込んだデータが“END”の場合にはそれを表示して、170行へジャンプします。

170行ですべてのファイルをクローズしプログラムを終了します。

120行で受信バッファにデータがあることをチェックしているのは、130行で空のバッファから入力を行わないようにするためです。空のバッファに対してLINE INPUT #命令などの入力命令を実行すると、“Input past end”のエラーが起これプログラムが停止してしまいます。

---

参考：[送信1]、および[送信2]のプログラムで送ったデータは、相手のコンピュータで[受信]のプログラムが実行中であれば、相手のコンピュータの画面に表示されます。

---

注意：この3つのサンプルプログラムはファイルの入力にLINE INPUT #命令とLINE INPUT 命令を使用しています。したがって、1行が256バイト以上になった場合にはエラーが起これますので注意してください。

---

### ■ RS-232C 回線ファイルの割り込み処理

RS-232C回線ファイルを通じてデータ通信を行うプログラム中で、INPUT#, LINE INPUT #命令、INPUT\$関数が実行されると、外部からのデータの入力が完了する(区切り記号を受け取る)までプログラムは停止します。相手からのデータが来ないうちは、こちらからデータを送ったり、他の作業を行ったりすることがまったくできなくなります。

この問題を解決するために用意されている命令がON COM GOSUB 命令とCOM ON/OFF/STOP 命令です。この命令を使用すると、データが受信されるとあらかじめ指定した行から始まるサブルーチンへジャンプすることができますから、データが受信されない間は必要な作業を行い、データを受信した場合だけ入力処理のサブルーチンへジャンプしてINPUT\$関数を実行するといったことが可能になります。

注意：INPUT #, LINE INPUT #は割り込みサブルーチン内では使用しないでください。正しくデータを受信できない場合があります。

#### ON COM GOSUB 命令

```
ON COM [(回線番号)] GOSUB <行番号>
```

<回線番号>に指定されたRS-232C回線ファイルでデータを受信すると、<行番号>で始まるサブルーチンにジャンプします。

#### COM ON/OFF/STOP 命令

```
COM [(回線番号)] | ON  
                  | OFF  
                  | STOP
```

<回線番号>で指定されたRS-232C回線ファイルからの割り込みを許可/禁止/停止します。

```
10 OPEN "COM:ET2XS" AS #1
20 ON COM GOSUB *REC
30 COM ON
40 OPEN "TEST" FOR INPUT AS #2
50 IF EOF(2) THEN *WAITING
60 LINE INPUT #2, A$
70 PRINT A$:
80 GOTO 50
90 *WAITING
100 CLOSE #2
110 GOTO 110
120 *REC
130 IF LOC(#1)=0 THEN RETURN
140 B$=INPUT$(LOC(1), #1)
150 PRINT B$:
160 GOTO *REC
```

データの受信がない間は、“TEST”というディスクファイルの内容を画面に表示します。データの受信があるとそれまでの作業を中止して\*REC行にジャンプし、送られてきたデータを画面に表示します。データの受信が終わるとメインルーチンにもどります。ディスクファイル“TEST”の内容の画面表示が先に終わってしまった場合、メインルーチンは\* WAITING にジャンプしてディスクファイルをクローズし、110行でRS-232C回線からのデータの受信を待ちます。

### ■ RS-232C 回線ファイルを使ったサンプルプログラム

ここでは、RS-232C回線ファイルを使ってデータ転送を行うプログラムを2つ紹介します。

1. コンピュータ同士で会話を行うプログラム
2. イメージスキャナから画像データをコンピュータに取り込み、画面に表示し、そのデータをディスクに書き込むプログラム

#### ●コンピュータ同士で会話を行うプログラム

キーボードから入力したデータの送信と受信を同時に行い、送信データと受信データの両方を画面に表示するプログラムです。文字による会話ができます。

##### ・プログラムの使い方

2台のPC-9800シリーズをRS-232C回線で接続し、次のような設定を行います。

|             |         |
|-------------|---------|
| ボーレート       | 9600 ボー |
| DEL コード受信処理 | 有効      |
| クロック同期      | 内部同期    |

ボーレートとDELコード受信処理の設定はユーティリティ“swich.n88”を使用して行ってください。クロック同期はディップスイッチによって設定します。

2台のPC-9800シリーズで同時にこのプログラムを実行します。

画面が上下2つのエリアに分離します。キーボードから入力したデータは画面の上半分に表示され、同時に相手に送られます。相手から送られてきたデータは画面の下半分に表示されます。

・プログラムの解説

初期設定(100-150行)

画面の設定, RS-232C 回線ファイルのオープン, 変数の初期化などを行っています。

100 行では画面からデータを消去し, 送信データ表示エリアとしてスクロールする範囲を 1~11 行に設定します。

110 行では RS-232C 回線ファイルをオープンします。設定される通信プロトコルは次のとおりです。

|          |       |
|----------|-------|
| パリティ     | なし    |
| データビット長  | 8 ビット |
| ストップビット長 | 1 ビット |
| X パラメータ  | 有効    |
| S パラメータ  | なし    |


120 行では変数の初期化, 130-140 行は送信データと受信データの画面表示エリアを分離しています。

150 行では通信割り込みがあった場合の飛び先を 270 行に宣言しています。

送信ルーチン(160-250行)

キーボードから入力したデータを画面表示し, 送信するルーチンです。なお, 180-240 行の処理中に通信割り込みが発生すると送信側と受信側の文字が画面上で混じるなどの不都合が生じるため, 190 行の COM STOP と 240 行の COM ON でこの間を割り込み停止状態にしています。

170-190 行ではキーボードから文字が入力されるのを待ちます。

何かキー入力があれば, その文字を RS-232C 回線ファイルに書き出し(送信)(200行), さらに画面の送信データ表示エリアに表示します(210-220行)。170 行で  キーが押された(&H0D が入力された)場合には, それを &H0D + &H0A (改行コード) に変換します。

230-250 行では送信データの画面の表示位置を記憶して次のキー入力待ちのために, 170 行へジャンプします。

受信ルーチン(260-350行)

送信中やキー入力待ちで受信による通信割り込みがあった場合には, このルーチンに実行が移ります。

270-280 行で受信バッファ中のデータがない場合は, 350 行を経由してメインルーチンにもどります。

290-320 行では受信バッファ中のデータを入力し、受信データ表示エリアとしてスクロール画面を13-23行に設定してそのデータを画面に表示します。このとき&H0Dだけが送られてきたら&H0D+&H0A(改行コード)に変換します。

330-340 行では受信データの画面表示位置を記憶し、スクロール画面を1-11行にもどして、メインルーチンにもどります。

**注意:** このプログラムを利用して、PC-9800 シリーズ以外のコンピュータとの会話を行うこともできます。ただし、相手のコンピュータ側で実行するプログラムはそのコンピュータのBASIC言語の仕様に合わせて修正してください。

```

100 CONSOLE 0,24,1:CLS:CONSOLE 1,11
110 OPEN "COM:N81XN" AS #1                :'RS-232C回線ファイルのオープン
120 XA=0:YA=1:XB=0:YB=13
130 LOCATE 0,0:PRINT STRING$(66,"-");"[送信データ]";
140 LOCATE 0,12:PRINT STRING$(66,"-");"[受信データ]";
150 ON COM GOSUB 270:COM ON                :'割り込み処理ルーチンの定義
160 'メインルーチン(送信)
170 A$=INKEY$
180 IF A$="" THEN 170
190 COM STOP                               :'割り込みの停止
200 PRINT #1,A$;                           :'データ送信
210 IF A$=CHR$(13) THEN A$=A$+CHR$(10)    :'CRをCR+LFに変換
220 LOCATE XA,YA:PRINT A$;                :'送信データの画面表示
230 XA=POS(0):YA=CSRLIN                    :'次のカーソル位置を記憶
240 COM ON                                 :'割り込みの許可
250 GOTO 170
260 '受信処理ルーチン
270 BUF=LOC(1)
280 IF BUF=0 THEN 350
290 B$=INPUT$(BUF,#1)                       :'バッファ中の受信文字列の取り込み
300 IF B$=CHR$(13) THEN B$=B$+CHR$(10)    :'CRをCR+LFに変換
310 CONSOLE 13,11
320 LOCATE XB,YB:PRINT B$;                 :'受信データの画面表示
330 XB=POS(0):YB=CSRLIN                    :'次のカーソル位置を記憶
340 CONSOLE 1,11
350 RETURN

```

### ● イメージスキャナを使用するプログラム

イメージスキャナは写真や印刷物の色(濃淡)をデジタルデータに変換してRS-232C回線を通じてコンピュータに送ります。コンピュータはこのデータを受け取って画面表示します。

ここでは、RS-232C回線を使った周辺機器の操作の一例としてイメージスキャナを取り上げています。

プログラム1は原稿を8階調で読み取って画面に表示し画像をディスクにセーブするものです。

プログラム2は、プログラム1でディスクにセーブした画像データをロードして画面に表示するものです。

・プログラム 1 の使い方

PC-9800 シリーズとイメージスキャナ(PC-IN503G)を RS-232C 回線で接続し、次のような設定を行ってください。

|             |         |
|-------------|---------|
| ボーレート       | 9600 ボー |
| DEL コード受信処理 | 有効      |
| クロック同期      | 内部同期    |

ボーレートと DEL コード受信処理の設定はユーティリティ "swich.n88" を使用して行ってください。クロック同期はディップスイッチによって設定します。

イメージスキャナに原稿を置きます。

プログラムを実行すると、

**画像をセーブするファイル名は？**

と聞いてきますので、ファイル名を入力してください。

イメージスキャナは画像を読み取り、コンピュータがそれを受信してデータ(画像)がディスプレイに表示されます。

データの読み込みが終了すると、画像がディスクにセーブされます。

・プログラム 1 の解説

**初期設定(100-170 行)**

高分解能カラーモードに設定し、画面をクリアし、変数を初期化します。

**ファイル名入力(180 行)**

画像をセーブするファイル名を入力します。

**RS-232C 回線ファイルのオープン(220 行)**

通信プロトコルを次のように設定します。

|          |       |
|----------|-------|
| パリティ     | なし    |
| データビット長  | 8 ビット |
| ストップビット長 | 1 ビット |
| X パラメータ  | 無効    |
| S パラメータ  | なし    |

**イメージスキャナの準備(240-340行)**

ESC\$ (&H1B)に続くエスケープシーケンスや、ENQ\$ (&H05), STT\$ (&H12)などの制御コードをRS-232C回線ファイルとして接続されたイメージスキャナに書き出(送信)して、スキャナの準備をします。

**画像データの転送と描画(360-570行)**

1ラインずつデータをいったん配列に読み込み、画面に描画しています。

画面上のデータライン1本がここで送られてくるデータライン3本分にあたります。

スキャナから送られてくる画像データの先頭および末尾にはコントロールデータなどがありますが、ここでは画像の読み込みに差し支えないので370行と410行でスキップさせています。

430行で必要な画像データを1バイトずつ読み込みます。

480-540行では画面への描画を行っています。

**イメージスキャナの停止(590-620行)**

STP\$ (&H14), ACK\$ (&H06)などの制御コードをRS-232C回線ファイルに書き出(送信)し、スキャナを停止させます。

**画像のセーブ(640-700行)**

画像データの格納されているメモリからメモリーメージの画像データをファイルにセーブします。

**RS-232C回線ファイルのクローズと終了(720-730行)**

END命令で、すべてのファイルはクローズされます。

**・プログラム2の使い方**

このプログラムを実行すると、

ロードするファイル名は？

と聞いてくるので、プログラム1で画像をセーブしたファイル名を入力してください。

**注意：**このプログラムはイメージスキャナ(PC-IN503G)を使う場合を想定してあります。また、これはBASICによるRS-232C回線ファイル操作(機械制御)の意味と内容を理解するためのサンプルです。このため、機械語を利用した場合にくらべると速度も高速ではありません。

実務レベルで高速なデータ転送が必要な場合には、オプションとして用意されている「イメージスキャナ対応 N<sub>88</sub>-日本語 BASIC(86)ソフトウェアドライバ」を使用してください。

## プログラム 1

```

100 SCREEN 3,0:CLS 3:COLOR ,,,,2
110 DIM S%(119,2)
120 ENQ$=CHR$(&H5)
130 ACK$=CHR$(&H6)
140 STT$=CHR$(&H12)
150 STP$=CHR$(&H14)
160 ESC$=CHR$(&H1B)
170 Y=0
180 INPUT "画像をセーブするファイル名は":F$
190 CLS:C=&HFFF:FOR I=0 TO 7:COLOR=(I,C):C=C-&H111:NEXT
200 '
210 '=== RS-232C回線ファイルのオープン ===
220 OPEN "COM:N81NN" AS #1
230 '
240 '=== スキャナの準備 ===
250 PRINT #1,ESC$;"I":      :'初期化
260 PRINT #1,ENQ$:          :'スキャナの状態の問い合わせ
270 IF LOC(1)=0 THEN 260
280 IF INPUT$(1,#1)<>ACK$ THEN 260
290 IF LOC(1)<>0 THEN A$=INPUT$(1,#1):GOTO 290
300 PRINT #1,ESC$;"H3":    :'8階調モード
310 PRINT #1,ESC$;"R4":    :'読み取り線密度 120本/インチ
320 PRINT #1,ESC$;"T3":    :'ラインハンドシェイクモード
330 '
340 PRINT #1,STT$:         :'読み取り開始
350 '
360 '=== 画像データの転送と描画 ===
370 A$=INPUT$(24,#1)      :'コントロールデータをスキップ
380 '1ライン分の読み取り
390 FOR I=0 TO 2          :'データ8行で1ライン分
400   PRINT #1,ACK$:     :'応答コードを送信
410   A$=INPUT$(4,#1)    :'ヘッダ情報をスキップ
420   FOR X=0 TO 120-1
430     A$=INPUT$(1,#1)
440     S%(X,I)=ASC(A$)
450   NEXT X
460   A$=INPUT$(1,#1)    :'トレーラ情報をスキップ
470 NEXT I
480 '1ライン分の描画
490 FOR I=0 TO 2
500   DEF SEG=&HA800+((I+2) MOD 3)*&H800
510   FOR X=0 TO 79
520     POKE X+Y,S%(X,I)
530   NEXT X
540 NEXT I
550 Y=Y+80
560 IF Y=400*80 THEN 600
570 GOTO 390
580 '
590 '=== スキャナの停止 ===
600 PRINT #1,STP$:       :'読み取り停止
610 PRINT #1,ACK$:       :'応答コードを送信
620 PRINT #1,ACK$:       :'応答コードを送信 (最大2行分をスキップ)
630 '
640 '=== 画像のセーブ ===
650 DEF SEG=&HA800
660 BSAVE F$+".1",0,&H8000
670 DEF SEG=&HB000
680 BSAVE F$+".2",0,&H8000
690 DEF SEG=&HB800
700 BSAVE F$+".3",0,&H8000
710 '
720 '=== 終了 (RS-232C回線ファイルのクローズ) ===
730 END

```

## プログラム 2

---

```

100 SCREEN 3:CLS 3
110 COLOR ,,,,2
120 INPUT "ロードするファイル名は":F$
130 CLS
140 C=&HFFF:FOR I=0 TO 7:COLOR=(I,C):C=C-&H111:NEXT
150 DEF SEG=&HA800
160 BLOAD F$+".1"
170 DEF SEG=&HB000
180 BLOAD F$+".2"
190 DEF SEG=&HB800
200 BLOAD F$+".3"
210 END

```

---

### ■RS-232C 回線ファイルを使用する場合の注意

#### 遠距離で通信を行う場合

離れているコンピュータとコンピュータ、あるいはコンピュータと機器で通信を行う場合(たとえば、電話回線を経由している場合など)、通信の環境条件が悪いと、データの一部が化けてしまったり、失われてしまったり、あるいは雑音によって余計なデータを拾ってしまうことがあります。

このようなときには、送られてきたデータをもう一度送り返して内容が同一であるかをチェックするといった操作を行うと、データの信頼性を確保することができます。

#### 一方の通信不能に対する処置

また、通信中になんらかの原因(たとえば停電、機器のハングアップなど)で一方がプログラムの実行をやめてしまったような場合に、もう一方がそのことを認識できないで、相手からのデータ送信を永遠に待ち続ける状態におちいる可能性があります。これは、無人運転を行っている場合にはとくに重要な問題です。

これを避けるには、「時間待ちをして一定の時間内に相手からの応答がなければ通信を終了する」といったようなアルゴリズムを使う方法があります。

こうした例以外にも、正常な通信をさまたげる原因はいろいろ存在します。プログラムを作成したあとは、あらゆる可能性を想定して何度か実験を行い、正しくコミュニケーションが行われるようにしてください。

## 6.9 その他のデバイスファイルの使い方

### ■ デバイスファイルの使用例

ここでは、RS-232C 回線ファイル以外のデバイスファイルの使用例を解説します。

#### ● キーボードファイル(KYBD:)

キーボードファイルは入力でのみ使えます。

例) キーボードファイルを入力モードでオープンし、タイプした文字を A\$ に代入する。

```
OPEN " KYBD:" FOR INPUT AS #1
INPUT #1, A$
CLOSE #1
```

#### ● スクリーンファイル(SCRN:)

スクリーンファイルは出力でのみ使えます。

例) スクリーンファイルを出力モードでオープンし、“ABC” という文字を出力する。

```
OPEN " SCRN:" FOR OUTPUT AS #1
WRITE #1, " ABC"
CLOSE #1
```

---

注意：スクリーンファイルに対しては日本語表示は行えません。

---

#### ● プリンタファイル(LPT:)

プリンタファイルは出力でのみ使えます。

例) プリンタファイルを出力モードでオープンし、“XYZ” という文字を出力する。

```
OPEN " LPT:" FOR OUTPUT AS #1
PRINT #1, " XYZ"
CLOSE #1
```

#### ● デバイスファイルの有効利用

入出力装置をファイルとして扱うことの利点は、異なった装置に対しても、同じ命令で入出力が可能になるということです。

たとえば、PRINT#命令はキーボードファイル以外のすべてのファイルに対して出力を行うことができます。また、INPUT#命令はスクリーンファイルとプリンタファイル以外のファイルからの入力を行うことができます。この場合、入出力先の指定さえ変更すればよいことになります。

たとえば、画面に文字を表示する場合に使用する一般的な命令である PRINT 命令を数多く使ったプログラムを作ったとします。このプログラムを画面表示するかわりにプリンタに出力するように改造したい場合には、プログラム中のすべての PRINT 命令を LPRINT 命令に変更しなければなりません。

それに対して、画面をスクリーンファイルとして扱い、OPEN 命令で "SCRN:" をオープンして PRINT #命令で出力するというプログラムを作っておけば、OPEN 命令の "SCRN:" を "LPT:" に変更するだけでプリンタに印字させることができます。さらに、出力先をディスクファイルに変えればディスクに書き込むこともできますし、"COM:" を指定すれば RS-232C 回線に出力することができます。

この機能は特にプログラムのデバッグの際に有効です。次にその一例をあげます。

- RS-232C 回線ファイルから送られてきたテキストファイルを入力してディスクに出力するプログラムを作ったが、書き込みがうまくいかない場合。

出力先を "SCRN:" に変更して、どんな文字が RS-232C 回線から送られてきているか調べる。

入力先を "KYBD:" に変えてみて、キーボードから入力したデータが、うまくディスクに書き込まれるかどうか調べる。

---

**注意：**ファイルの入出力に使用可能な BASIC の命令はファイルの種類によって異なりますので注意してください。

---

▶参照 ファイルの入出力に使用可能な BASIC の命令→6.1の「デバイス(入出力装置)とファイル」



# 第7章

## 入力装置

N<sub>88</sub>-日本語 BASIC(86)ではキーボードの他にマウス、ライトペンといった入力装置を使用することができます。この章ではこの2つの装置の使い方を解説します。

### 7.1 マウスの使い方

マウスはディスプレイ上のポインティングデバイス(指示装置)として、強力な働きをします。マウスの動きに合わせてディスプレイ上でカーソルを移動させることができますし、マウスについている2つのボタンを押したり離したりすることにより、ソフトウェアとのやり取りが可能になります。

N<sub>88</sub>-日本語 BASIC(86)にはマウスを使用するための命令や関数は用意されていませんが、それに代わるものとしてマウス用のソフトウェアドライバ“mouse.cod”(マウスを使用するための機械語プログラム)が提供されています。このプログラムはマウスの動き、ボタンが押されているかどうかなどを調べるもので、BASIC上からも機械語プログラムからも利用できるようになっています。

#### ■マウス用ソフトウェアドライバの使用方法

マウス用ソフトウェアドライバは機械語プログラムですから、BASICのCALL命令を使って呼び出すことができます。

BASICでマウスを使うときには、マウス用のソフトウェアドライバをメモリ上にロードしておき、マウスの状態を知りたいときにBASICのCALL命令を使ってソフトウェアドライバを呼び出して実行させます。

#### ●ソフトウェアドライバを呼び出すための準備

ソフトウェアドライバを利用するためには、まずその準備を行わなければなりません。次に、準備のためのサンプルプログラムを示し、各部分について説明します。

---

```
10 'load mouse driver.
20 CLEAR ,&H7F00
30 DEF SEG=&H7F00
40 BLOAD "mouse.cod"
50 '
60 'check environment.
70 MOUSE.INI=&H100
80 FLAG%=3
90 CALL MOUSE.INI(FLAG%)
100 IF FLAG%=0 THEN PRINT "BAD ENVIRONMENT." : END
110 '
120 'set mouse driver offset.
130 DEF SEG=0
140 INT33=PEEK(&H33*4)+PEEK(&H33*4+1)*256
150 MOUSE=INT33+3
160 DEF SEG=&H7F00
170 '
```

---

### ①ソフトウェアドライバのロード

まず、最初にソフトウェアドライバ"mouse.cod"をメモリの機械語領域にロードします。ソフトウェアドライバは4KBの領域を必要としますので、あらかじめこれを格納するのに十分な機械語領域をCLEAR命令で確保しておきます。

### ②マウスハードウェア環境のテストとソフトウェアドライバの初期化

次に、マウスハードウェア環境が整っているかどうかのテストを行うために、ソフトウェアドライバの相対アドレスの&H100から始まる機械語サブルーチンをCALL命令で呼び出して実行します。CALLする際にはグラフィック画面の解像度によって次の値を引数として指定します。

高分解能モード(640×400ドット)の場合：3

低分解能モード(640×200ドット)の場合：0

---

注意：グラフィック画面の解像度はSCREEN命令で設定できます。

上の例の「高分解能モード」とは高分解能カラーモードと高分解能白黒モードのことを、「低分解能モード」とはカラーモードと白黒モードのことを表します。

---

マウスハードウェア環境に問題がある場合、引数には0が返されます。この場合は、サンプルプログラムのように、エラーメッセージを表示してプログラムの実行を中止するなどの措置をとるようにしてください。

マウスハードウェア環境に問題がない場合、ソフトウェアドライバの実行開始アドレス(相対アドレス)が割り込みベクタテーブルの33H番目のエン트리(登録場所)に置かれます。

### ③ソフトウェアドライバのファンクションエントリアドレスの設定

最後に、割り込みベクタテーブルの 33H 番目から値を取り出し、これに 3 を加えると、ソフトウェアドライバを呼び出す場合のファンクションエントリアドレス(実行開始アドレス、相対アドレスで表される)が得られます。

このサンプルプログラムでは、変数 MOUSE にファンクションエントリアドレスをセットしています。

#### ●ソフトウェアドライバの呼び出し

ソフトウェアドライバの呼び出しには BASIC の CALL 命令を使います。次のように、引数に 5 つの整数型の変数を指定します。

```
CALL MOUSE(AX, BX, CX, DX, ES)
```

MOUSE は数値変数であり、前項で述べた準備の方法により、まえもってソフトウェアドライバのファンクションエントリアドレス(相対アドレス)が代入されていなければなりません。

引数(AX~ES)はすべて整数型の変数でなくてはなりません。あらかじめ DEFINT A-E などと宣言しておくか、あるいは各変数の後ろに % をつけるようにしてください。なお、ここではファンクションのパラメータの説明と対応がとりやすいように AX, BX, CX, DX, ES を使用していますが、順番さえ間違えなければ別の名前を使用してもかまいません。

なお引数として直接数値を指定することはできません。必ず変数を介して指定します。また引数は必ず 5 つ指定しなければなりません。使用しないパラメータがある場合にはダミーの変数名を指定してください。

#### ■ソフトウェアドライバのファンクション

ソフトウェアドライバには次のような 16 個のファンクション(機能)が用意されています。各ファンクションは、ソフトウェアドライバを呼び出すとき、CALL 命令の最初の引数 AX に指定する値(ファンクションコード)によって決定されます。また、各ファンクションの実行結果は、呼び出し時に指定した CALL 命令の各引数に値がセットされることによって返されます。

16 種類のファンクションは次のとおりです。

| ファンクションコード | 機 能                     |
|------------|-------------------------|
| 0          | マウス環境の初期化               |
| 1          | カーソル表示                  |
| 2          | カーソル消去                  |
| 3          | カーソル位置の取得               |
| 4          | カーソル位置の設定               |
| 5          | 左ボタンの押下情報の取得            |
| 6          | 左ボタンの解放情報の取得            |
| 7          | 右ボタンの押下情報の取得            |
| 8          | 右ボタンの解放情報の取得            |
| 9          | カーソルの形の設定               |
| 11         | マウスの移動距離の取得             |
| 12         | ユーザー定義機械語プログラムのコール条件の設定 |
| 15         | ミッキー／ドット比の設定            |
| 16         | 水平方向のカーソル移動範囲の設定        |
| 17         | 垂直方向のカーソル移動範囲の設定        |
| 18         | カーソルの表示画面の設定            |

注意：ファンクションコード 10, 13, 14 は存在しません。

次にファンクションコード別に、ソフトウェアドライバに渡す引数の値と、実行後に引数に返される値について説明します。渡す値を[入力]、返される値を[出力]と表しています。

### ファンクション 0 (マウス環境の初期化)

[入力] AX=0

[出力] AX=環境状態

0：マウスを使用できない環境

-1：マウスを使用できる環境

[説明] カーソルの表示、カーソルの形、カーソルの中心点、ミッキー／ドット比、マウスの割り込み周期などの環境を初期化します。

### ファンクション 1 (カーソル表示)

[入力] AX=1

[出力] なし

[説明] カーソルをスクリーン上に表示させるためのファンクションです。一度このファンクションをコールすると、ファンクション 2(カーソル消去)あるいはファンクション 0(マウス環境の初期化)をコールするまで、カーソルがマウスの動きに従ってスクリーン上を動きます。

## ファンクション 2 (カーソル消去)

---

[入力] AX=2

[出力] なし

[説明] スクリーン上に表示されているカーソルを消去するためのファンクションです。一度このファンクションをコールすると、ファンクション1(カーソル表示)をコールするまで、スクリーンには表示されません。しかし、カーソルは表示されていなくても、カーソルの位置はマウスの移動に応じてスクリーン上で移動しています。

## ファンクション 3(カーソル位置の取得)

---

[入力] AX=3

[出力] AX=左ボタンの状態

0: 離されている

-1: 押されている

BX=右ボタンの状態

0: 離されている

-1: 押されている

CX=カーソル位置の水平座標

0~639

DX=カーソル位置の垂直座標

0~199: 低分解能モード

0~399: 高分解能モード

[説明] 現在のカーソルの位置を取得するためのファンクションです。カーソルの位置は、オリジナルスクリーン座標系の水平座標、垂直座標で得られます。それはカーソルの移動範囲内の値です。またこのときのマウスの右ボタンおよび左ボタンの状態(離されているか、押されているか)も得ることができます。

## ファンクション 4 (カーソル位置の設定)

---

[入力] AX=4

CX=カーソルの新しい位置の水平座標

0~639

DX=カーソルの新しい位置の垂直座標

0~199: 低分解能モード

0~399: 高分解能モード

[出力] なし

[説明] カーソルを任意の位置に移動するためのファンクションです。オリジナルスクリーン座標系の水平座標、垂直座標を引数に指定してこのファンクションをコールすると、カーソルはその位置に移動します。指定の位置がカーソル移動範囲外の場合には、移動範囲内の端にカーソルを移動します。

### ファンクション 5 (左ボタンの押下情報の取得)

---

[入力] AX=5

[出力] AX=左ボタンの状態

0: 離されている

-1: 押されている

BX=このファンクションが最後にコールされてから、今回コールされるまでに左ボタンが押された回数

CX=最後に左ボタンが押された時のカーソル位置の水平座標

DX=最後に左ボタンが押された時のカーソル位置の垂直座標

[説明] マウスの左ボタンの押下(押されること)に関する各種の情報を得るためのファンクションです。

### ファンクション 6 (左ボタンの解放情報の取得)

---

[入力] AX=6

[出力] AX=左ボタンの状態

0: 離されている

-1: 押されている

BX=このファンクションが最後にコールされてから、今回コールされるまでに左ボタンが離された回数

CX=最後に左ボタンが離された時のカーソル位置の水平座標

DX=最後に左ボタンが離された時のカーソル位置の垂直座標

[説明] マウスの左ボタンの解放(離されること)に関する各種の情報を得るためのファンクションです。

## ファンクション 7 (右ボタンの押下情報の取得)

---

[入力] AX=7

[出力] AX=右ボタンの状態

0: 離されている

-1: 押されている

BX=このファンクションが最後にコールされてから、今回コールされるまでに右ボタンが押された回数

CX=最後に右ボタンが押された時のカーソル位置の水平座標

DX=最後に右ボタンが押された時のカーソル位置の垂直座標

[説明] マウスの右ボタンの押下(押されること)に関する各種の情報を得るためのファンクションです。

## ファンクション 8 (右ボタンの解放情報の取得)

---

[入力] AX=8

[出力] AX=右ボタンの状態

0: 離されている

-1: 押されている

BX=このファンクションが最後にコールされてから、今回コールされるまでに右ボタンが離された回数

CX=最後に右ボタンが離された時のカーソル位置の水平座標

DX=最後に右ボタンが離された時のカーソル位置の垂直座標

[説明] マウスの右ボタンの解放(離されること)に関する各種の情報を得るためのファンクションです。

## ファンクション 9 (カーソルの形の設定)

---

[入力] AX=9

BX=カーソルの中心点の水平座標(0~15)

CX=カーソルの中心点の垂直座標

0~15: 低分解能モード

0~31: 高分解能モード

ES: DX=カーソルの形を決定するデータを格納した領域の先頭アドレス

ES: セグメントベース

DX: 相対アドレス

データの必要量

低分解能モード → 16×16 ビット=32 バイト

高分解能モード → 16×32 ビット=64 バイト

[出力] なし

[説明] カーソルの形や中心点を設定するためのファンクションです。カーソルの形は、16×16(または16×32)ドットの四角形のうち、どのドットを表示するかによって決まります。つまり、表示されたドットの集合がカーソルとして見えるわけです。たとえば、四角形のカーソルのドットをすべて表示するように指定すれば、カーソルの形は最大の四角形になります。

カーソルの形を決定するデータの形式は次のようになります。

|          |          |   |                            |
|----------|----------|---|----------------------------|
| 11000000 | 00000000 | ↑ |                            |
| 11100000 | 00000000 |   | 低分解能モード→16行<br>高分解能モード→32行 |
|          | )        |   |                            |
| 00000000 | 00000000 | ↓ |                            |

また、BX, CX に指定するカーソルの中心点は、カーソルの左上角のドットを(0, 0)とした座標系の相対位置で与えます。

### ファンクション 11 (マウスの移動距離の取得)

[入力] AX=11

[出力] CX=マウスの水平方向の移動距離

-32768~32767

DX=マウスの垂直方向の移動距離

-32768~32767

[説明] マウスの移動距離を取得するためのファンクションです。このファンクションを最後にコールした時のマウスの位置から、今回コールされた時点のマウスの位置までの水平方向および、垂直方向の相対的な距離を返します。水平方向では右の向きが正であり、垂直方向では手前の向きが正となります。また距離の単位はミッキー(約1/100インチ, 0.25 ミリ)で、その範囲は-32768 から 32767 です。

### ファンクション 12 (ユーザー定義機械語プログラムのコール条件の設定)

[入力] AX=12

CX=コールの条件(ビットが1のときはコールし、0のときはコールしない)

ビット0:カーソル位置の変化

ビット1:左ボタンが押される

ビット 2：左ボタンが離される

ビット 3：右ボタンが押される

ビット 4：右ボタンが離される

ビット 5～15：未使用

以上、最下位ビットをビット 0 とする。

ES：DX＝ユーザー定義機械語プログラムの実行開始アドレス

[出力] なし

[説明] マウスを使って機械語プログラムをコールする条件と、その実行開始アドレスを設定するためのファンクションです。このファンクションをあらかじめ実行しておくことにより、ソフトウェアドライバは、ここに示した 5 つのコールの条件のどれかが満たされていると次のように機械語プログラムをコールします。

①ソフトウェアドライバは、マウスから割り込みが発生すると、まずコール条件が満たされているかどうかを調べます。

②条件が満たされていない場合にはそのまま次の動作(命令)に移ります。

③条件が満たされている場合には CALL FAR-PROC 命令によってソフトウェアドライバから機械語プログラムに制御が移ります。したがって、機械語プログラムからソフトウェアドライバに制御をもどすためには、RET FAR-PROC 命令を使用しなければなりません。

なお、ソフトウェアドライバが機械語プログラムをコールするときには、各レジスタには次の情報が格納されます。

AX＝コールの原因となった現象

- 1：カーソルの位置が変わった
- 2：左ボタンが押された
- 4：左ボタンが離された
- 8：右ボタンが押された
- 16：右ボタンが離された

BL＝左ボタンの状態

- 0：離されている
- 1：押されている

BH＝右ボタンの状態

- 0：離されている
- 1：押されている

CX＝カーソルの位置の水平座標

DX＝カーソルの位置の垂直座標

## ファンクション 15 (ミッキー／ドット比の設定)

---

[入力] AX=15

CX=水平方向のミッキー／ドット比

DX=垂直方向のミッキー／ドット比

[出力] なし

[説明] マウスの移動距離とそれに対応するカーソルの移動距離との比を設定するためのファンクションです。水平方向および垂直方向にカーソルが8ドット移動するのに要するマウスの水平方向および垂直方向の移動距離(ミッキー／ドット比)を設定します。この設定により、マウスを少し動かしただけでカーソルが大きく移動したり、逆にマウスを大きく動かしてもカーソルは少ししか移動しないというように、マウスの移動感度を変えることができます。

## ファンクション 16 (水平方向のカーソル移動範囲の設定)

---

[入力] AX=16

CX=カーソルの水平方向の移動範囲の最小値(0~639)

DX=カーソルの水平方向の移動範囲の最大値(0~639)

[出力] なし

[説明] カーソルの水平方向の移動範囲を設定するためのファンクションです。移動範囲は、その最小値および最大値を設定することにより決まり、カーソルの中心点がこの範囲内を移動することができます。CXの値の方がDXの値より大きい場合には、DXの値が最小値、CXの値が最大値となります。

## ファンクション 17 (垂直方向のカーソル移動範囲の設定)

---

[入力] AX=17

CX=カーソルの垂直方向の移動範囲の最小値

0~199:低分解能モード

0~399:高分解能モード

DX=カーソルの垂直方向の移動範囲の最大値

0~199:低分解能モード

0~399:高分解能モード

[出力] なし

[説明] カーソルの垂直方向の移動範囲を設定するためのファンクションです。移動範囲は、その最小値および最大値を設定することにより決まり、カーソルの中心点がこの範囲内

を移動することができます。CX の値の方が DX の値より大きい場合には、DX の値が最小値、CX の値が最大値となります。

---

**注意：**水平方向あるいは垂直方向のカーソル移動範囲の設定(ファンクション 16, 17)によってカーソルの位置が移動範囲外になった場合には、ソフトウェアドライバが自動的にカーソルを移動範囲内の端に移動させます。

---

## ファンクション 18 (カーソルの表示画面の設定)

---

[入力] AX=18

BX=カーソルの表示画面

0: プレーン 0 へ表示

1: プレーン 1 へ表示

2: プレーン 2 へ表示

3: プレーン 3 へ表示

[出力] なし

[説明] カーソルの表示画面を設定するファンクションです。ソフトウェアドライバによって画面にマウスカーソルを表示すると、このファンクションで設定したグラフィック用 VRAM のプレーン 0~3 に直接カーソルが書き込まれます。

パレットが初期状態であれば、使用するプレーンとカーソルの色の関係は次のようになります。

プレーン 0: 青

プレーン 1: 赤

プレーン 2: 緑

プレーン 3: 灰

パレットを変更すれば、カーソルの色も変化します。

なお、プレーン 3 が実装されていない場合にカーソルの表示画面をプレーン 3 に設定しても無効です。

---

**注意：**8 色中・8 色モードのときにプレーン 3 にカーソルを表示しようとしても、その動作は保証されません。

高分解能白黒モードおよび白黒モードの場合に、グラフィックのディスプレイページのプレーンと、マウスカーソルを表示するプレーンとが一致していないと、マウスカーソルを見ることができなくなります。

---

## ■マウス環境の初期設定

マウス用ソフトウェアドライバは、ファンクション0のコールにより次のようにマウス環境を初期設定します。

カーソル表示           : 表示しない  
カーソルの形           : 左上向きの矢印  
カーソルの中心点       : (0, 0)  
ミッキー／ドット比     : 水平方向, 垂直方向とも 8

## ■ソフトウェアドライバ使用上の注意

### ●マウスの画面座標系

マウス用ソフトウェアドライバを使用して座標を指定する場合(ファンクション3, 4, 16, 17), 画面のモードによって座標の指定値が異なりますので注意が必要です。

#### 高分解能モード(640×400)の場合

水平方向: 0~639

垂直方向: 0~399

#### 低分解能モード(640×200)の場合

水平方向: 0~639

垂直方向: 0~199

なお、画面モードは SCREEN 命令で設定することができます。

### ●カーソルの形と中心点

カーソルは画面モードによって大きさが異なります。

高分解能モード   …… 16×32 ドット(最大)

低分解能モード   …… 16×16 ドット(最大)

カーソルのスクリーン上の位置を決める基準となる点をカーソルの中心点といいます。カーソル内の座標中の任意の1ドットを中心点とすることができます。カーソルの中心点はカーソル内の左上隅のドットを(0, 0)とする相対的な座標系で設定します。

### ●ミッキー

マウスを動かした時の移動距離はミッキーという単位で表します。1ミッキーは約100分の1インチ(0.25ミリ)です。ファンクション15を使って、カーソルを8ドット移動するのに必要なマウスの移動距離をミッキーの単位で指定することにより、マウスの移動感度を変えることができます。

## ■マウスを使ったプログラム例

マウスのソフトウェアドライバを使ったサンプルプログラムを示します。

このプログラムは次の6種類の機能を持っています。

- 画面のクリア(CLS)
- 2点間を結ぶ線を引く(LINE)
- 2点間を結ぶ箱を描く(BOX)
- 2点間を結ぶ円を描く(CIRCLE)
- ペイントする(PAINT)
- プログラムを終了する(EXIT)

これらの機能は、画面最上段に表示されたCLS、LINE、BOX、CIRCLE、PAINT、EXITの任意の場所にカーソルを位置づけ、左あるいは右のボタンを押すことにより選択します。

また、各機能を実行する際の色の選択も可能です。これを行うには画面2行目に表示されたBLACK、BLUE、RED、MAGENTA、GREEN、CYAN、YELLOW、WHITEの中で任意の場所にカーソルを位置づけ、左あるいは右ボタンを押すことにより選択します。

たとえば、赤で線を引く場合、まずカーソルをLINEに位置づけてボタンを押し、次にカーソルをREDに位置づけてボタンを押します。

機能を実行する場所も左あるいは右ボタンで指示します。たとえば、赤で線を引く場合は、機能と色の選択をした後、任意の2点でボタンを押します。

このプログラムは次の5種のファンクションを使用しています。

- マウス環境の初期化(ファンクション0)
- カーソルの表示(ファンクション1)
- カーソルの消去(ファンクション2)
- カーソルの取得(ファンクション3)
- カーソルの位置の設定(ファンクション4)

なお、このサンプルプログラムではマウス用ソフトウェアドライバをロードするための機械語領域の先頭番地を7F000H番地(物理アドレス)としています。

実際にこのプログラムを動作させる場合、お手持ちのシステムのメモリサイズに合わせて調整してください。

```
1000 CLEAR,&H7F00
1010 MOUSE.INI=&H100
1020 MSEG=&H7F00
1030 GOSUB *SET.SEG
1040 CONSOLE 0,25,0,0
1050 WIDTH 80,25
1060 SCREEN 3,1,0,1
1070 CLS 3
1080 DIM COMSTR$(6),COLSTR$(7)
1090 GOSUB *LOADCOM
1100 WUP=32
1110 FALSE=(1=0)
1120 TRUE=(1=1)
1130 GOSUB *NORMSCRN
1140 IF PEEK(&H100)<>233 THEN GOSUB *LOAD.MOUSE ELSE 1180
1150 FLAG%=3
1160 CALL MOUSE.INI(FLAG%)
1170 IF FLAG%=0 THEN PRINT "MOUSE:Interface card not found or illegal pa
rameter" : BEEP : END
1180 ON STOP GOSUB *MEXIT
1190 STOP ON
1200 DEF SEG=0
1210 INT33=PEEK(&H33*4)+PEEK(&H33*4+1)*256
1220 MOUSE=INT33+3
1230 GOSUB *SET.SEG
1240 AX%=0 : GOSUB *MOUSE
1250 IF AX%=0 THEN PRINT "MOUSE:Interface card not found" : BEEP : END
1260 COMD=2 : SWFLAG=FALSE : COL=7 : CANF=FALSE
1270 GOSUB *PRCOM
1280 X=320 : Y=200 : GOSUB *SETXY
1290 GOSUB *SHOWCU
1300 *MLOOP
1310 IF CANF THEN *COMMAND
1320 GOSUB *WAIT.SWITCH
1330 *COMMAND
1340 CANF=FALSE
1350 IF Y>=16 THEN *SKIPCOM
1360 SAVED.COM=COMD
1370 COMD=INT(X/8/8)+1
1380 IF COMD>6 THEN COMD=SAVED.COM : GOTO *MLOOP
1390 GOSUB *PRCOM
1400 ON COMD GOSUB *MCLS,*MLIN1,*MBOX1,*MCIRCLE1,*NUL.PROC,*MEXIT
1410 IF COMD=1 THEN COMD=SAVED.COM : GOSUB *PRCOM
1420 GOTO *MLOOP
1430 *SKIPCOM
1440 SAVED.COL=COL
1450 IF Y>=WUP THEN *SKIPCOL
1460 COL=INT(X/8/8)
1470 IF COL>7 THEN COL=SAVED.COL : GOTO *MLOOP
1480 GOSUB *PRCOM
1490 GOTO *MLOOP
1500 *SKIPCOL
1510 ON COMD GOSUB *MCLS,*MLIN2,*MBOX2,*MCIRCLE2,*MPAINT,*MEXIT
1520 IF CANF THEN *COMMAND
1530 GOTO *MLOOP
1540 *MCLS
1550 GOSUB *HIDEUC
1560 SCREEN ,3 : CLS 2 : GOSUB *NORMSCRN
1570 GOSUB *SHOWCU
1580 RETURN
1590 *MLIN1
1600 GOSUB *WAIT.SWITCH
1610 *MLIN2
1620 X1=X : Y1=Y
1630 IF Y1<WUP THEN CANF=TRUE : RETURN
1640 GOSUB *WAIT.SWITCH
```

```

1650 X2=X : Y2=Y
1660 IF Y2<WUP THEN CANF=TRUE : RETURN
1670 GOSUB *HIDECU
1680 LINE (X1,Y1-WUP)-(X2,Y2-WUP),COL
1690 GOSUB *SHOWCU
1700 CANF=FALSE : RETURN
1710 *MBOX1
1720 GOSUB *WAIT.SWITCH
1730 *MBOX2
1740 X1=X : Y1=Y
1750 IF Y1<WUP THEN CANF=TRUE : RETURN
1760 GOSUB *WAIT.SWITCH
1770 X2=X : Y2=Y
1780 IF Y2<WUP THEN CANF=TRUE : RETURN
1790 GOSUB *HIDECU
1800 LINE (X1,Y1-WUP)-(X2,Y2-WUP),COL,B
1810 GOSUB *SHOWCU
1820 CANF=FALSE : RETURN
1830 *MCIRCLE1
1840 GOSUB *WAIT.SWITCH
1850 *MCIRCLE2
1860 X1=X : Y1=Y
1870 IF Y1<WUP THEN CANF=TRUE : RETURN
1880 GOSUB *WAIT.SWITCH
1890 X2=X : Y2=Y
1900 IF Y2<WUP THEN CANF=TRUE : RETURN
1910 R=SQR((X2-X1)^2+(Y2-Y1)^2)
1920 GOSUB *HIDECU
1930 CIRCLE(X1,Y1-WUP),R,COL
1940 GOSUB *SHOWCU
1950 CANF=FALSE : RETURN
1960 *MPAINT
1970 GOSUB *HIDECU
1980 PAINT (X,Y-WUP),COL
1990 GOSUB *SHOWCU
2000 *NUL.PROC
2010 CANF=FALSE : RETURN
2020 *MEXIT
2030 GOSUB *HIDECU
2040 COLOR 0 : CLS : LOCATE 0,0,1
2050 STOP OFF
2060 END
2070 *MOUSE
2080 CALL MOUSE (AX%,BX%,CX%,DX%,ES%)
2090 RETURN
2100 *LOAD.MOUSE
2110 GOSUB *SET.SEG
2120 BLOAD "mouse.cod"
2130 RETURN
2140 *SET.SEG
2150 DEF SEG=MSEG
2160 RETURN
2170 *SHOWCU
2180 AX%=1 : GOSUB *MOUSE
2190 RETURN
2200 *HIDECU
2210 AX%=2 : GOSUB *MOUSE
2220 RETURN
2230 *SETXY
2240 AX%=4 : CX%=X : DX%=Y : GOSUB *MOUSE
2250 RETURN
2260 *GETXY
2270 AX%=3 : GOSUB *MOUSE
2280 SW1=AX% : SW2=BX% : X=CX% : Y=DX%
2290 RETURN
2300 *WAIT.SWITCH : GOSUB *GETXY
2310 IF SW1=0 AND SW2=0 THEN SWFLAG=FALSE : GOTO *WAIT.SWITCH

```

```

2320 IF SWFLAG THEN GOTO *WAIT.SWITCH
2330 SWFLAG=TRUE
2340 RETURN
2350 *NORMSCRN
2360 SCREEN 3,0
2370 VIEW (0,WUP)-(639,399)
2380 RETURN
2390 *PRCOM
2400 COLOR 0
2410 FOR I=1 TO 6
2420 IF I=COMD THEN COLOR 4
2430 LOCATE (I-1)*8,0,0
2440 PRINT COMSTR$(I);
2450 COLOR 0
2460 NEXT
2470 FOR I=0 TO 7
2480 IF I=COL THEN COLOR 4
2490 LOCATE I*8,1,0
2500 PRINT COLSTR$(I);
2510 COLOR 0
2520 NEXT
2530 RETURN
2540 *LOADCOM
2550 FOR I=1 TO 6
2560 READ COMSTR$(I)
2570 NEXT
2580 FOR I=0 TO 7
2590 READ COLSTR$(I)
2600 NEXT
2610 RETURN
2620 DATA "CLS","LINE","BOX","CIRCLE","PAINT","EXIT"
2630 DATA "BLACK","BLUE","RED","MAGENTA","GREEN","CYAN","YELLOW","WHITE"

```

---

## 7.2 ライトペンの使い方

N<sub>88</sub>-BASIC(86)にはライトペンを BASIC で制御する命令が用意されています。この命令を使用すればライトペンを利用するプログラムを簡単に作成することができます。

---

注意：デジタル CRT ディスプレイ用インタフェースが実装されていない機種では、ライトペンを利用することはできません。

---

### ■ライトペン制御命令

#### ON PEN GOSUB 命令

ライトペン割り込みによる処理ルーチンの開始行を定義します。

#### PEN ON/OFF/STOP 命令

ライトペンによる割り込みの許可、禁止、停止を制御します。

#### PEN 関数

ライトペンを画面に押しつけたときのペンの座標を与えます。

## ■ライトペン制御命令を使ったプログラム例

ライトペン制御命令の個々の詳しい解説は『BASICリファレンスマニュアル』を参照してください。ここでは、実際にプログラムのなかで各命令がどのように使用されるかを解説します。

---

```

100 DATA triangle,box,box fill,circle,CRT clear,end
110 SCREEN 0,0 : COLOR 7,0,0,7
120 CONSOLE 0,25,0,1 : WIDTH 80,25 : CLS 3
130 DEFINT A-Z
140 XX=500 : YY=300 : X=500 : Y=300
150 DEF FNX=RND*X : DEF FNY=RND*Y
160 DEF FNC=INT(RND*7+1) : DEF FNS=RND*100-50
170 DEF FNR=RND*100
180 P$=CHR$(&H87)+CHR$(&H87) : P$=P$+P$
190 ON PEN GOSUB *PENSUB
200 PEN ON
210 '
220 LOCATE 18,0 : PRINT "LIGHT PEN demonstration"
230 LOCATE 66,4 : PRINT "< MENU >"
240 FOR I=0 TO 5
250   READ MENU$(I)
260   LOCATE 63,I*3+6 : PRINT P$
270   LOCATE 63,I*3+7 : PRINT P$
280   LOCATE 68,I*3+7 : PRINT MENU$(I)
290 NEXT I
300 VIEW (1,10)-(451,198),,7
310 WINDOW(0,0)-(XX,YY)
320 GOTO 320
330 '
340 *PENSUB
350 PX=PEN(1) : PY=PEN(2) : PEN OFF
360 IF PX<64 OR PX>67 THEN *EXIT
370 IF 5<PY AND PY<8 THEN GOSUB *TRIANGLE : GOTO *EXIT
380 IF 8<PY AND PY<11 THEN GOSUB *BOX : GOTO *EXIT
390 IF 11<PY AND PY<14 THEN GOSUB *BOXFILL : GOTO *EXIT
400 IF 14<PY AND PY<17 THEN GOSUB *CIRCL : GOTO *EXIT
410 IF 17<PY AND PY<20 THEN GOSUB *CRTCLS : GOTO *EXIT
420 IF 20<PY AND PY<23 THEN GOSUB *ENDER
430 *EXIT : PEN ON : RETURN
440 '
450 *TRIANGLE
460 SX=FNX : SY=FNY : C=FNC
470 LINE (SX,SY)-STEP(FNS*2,FNS),C
480 LINE -STEP(FNS*2,FNS),C
490 LINE -(SX,SY),C
500 RETURN
510 *BOX
520 LINE (FNX,FNY)-STEP(FNS*2,FNS),FNC,B
530 RETURN
540 *BOXFILL
550 LINE (FNX,FNY)-STEP(FNS*2,FNS),FNC,BF
560 RETURN
570 *CIRCL
580 CIRCLE (FNX,FNY),FNR,FNC
590 RETURN
600 *CRTCLS
610 CLS 2 : RETURN
620 *ENDER
630 PEN OFF : END

```

---

### プログラムの動作

このプログラムを実行すると、画面メニューとして、白でぬられた枠(図形の名前が表示されている)が表示されます。任意の図形の枠にライトペンを押しつけると、対応する図形が画面に描かれます。

“CRT clear” を選択すると、それまでに描かれた図形がクリアされます。

“end” を選ぶとプログラムは自動的に終了します。

### プログラムの説明

190 行でライトペンによる割り込み先を定義し、200 行で割り込みの許可を行っています。240 行から 310 行でメニューとライトペンでタッチする部分の表示を行った後、320 行でループに入って、割り込みを待っています。

画面のどこかがライトペンで押されると 340 行から始まるサブルーチンへジャンプします。350 行ではペンが押されている座標をチェックし、360 行から 420 行ではその座標値からどのメニューが選択されたかを判定しています。

図形が描かれると再び 320 行へもどり割り込みを待ちます。

### ライトペンを扱う上の注意

ライトペンはディスプレイの光を読み取っていますので、ライトペンでタッチする部分は色(黒以外)がついていなければなりません。また、赤に対する感度が非常に弱いので、赤は使用しないほうがよいでしょう。

▶ 参照 ライトペン制御命令 → 『BASICリファレンスマニュアル』

# 第8章

## 機械語プログラム

N<sub>88</sub>-日本語 BASIC(86)では機械語プログラムをメモリ上に書いておき、それを BASIC から呼び出して実行させることができます。機械語プログラムを実行するためには BASIC の命令を使って次に示すような準備をしなければなりません。この章では機械語プログラムを BASIC から使う場合の準備と実行方法の流れを解説します。

なお、アセンブラなどを使って機械語プログラムそのものを作成する方法はここでは解説しませんので、機械語の入門書などを参考にしてください。

機械語プログラムが使用するメモリ領域の確保

CLEAR 命令

↓

使用するセグメントの宣言

DEF SEG 命令

↓

機械語プログラムをメモリに置く ↔ 機械語プログラムをディスクにセーブする

POKE 命令

BSAVE 命令

BLOAD 命令

MON 命令

↓

BASIC から機械語プログラムを呼び実行させる

DEF USR 命令と USR 関数

CALL 命令

BLOAD 命令

## 8.1 メモリレイアウトとセグメント

機械語では00000H 番地から FFFFFH 番地までの最大 1M バイトのメモリを扱うことができます。

しかし、番地の指定の際には 00000H 番地から FFFFFH 番地までの絶対アドレスを指定することはできません。メモリを 64K バイト単位のセグメントという単位に分けて扱います。番地の指定の前には、あらかじめセグメントの先頭番地を宣言し、このセグメントの先頭番地(セグメントベース)からの相対アドレスで指定します。

### ■セグメントベース

セグメントの先頭番地を宣言するには DEF SEG 命令を使います。

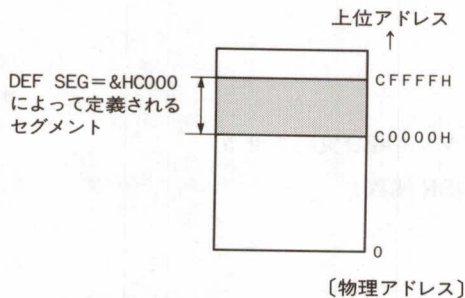
#### DEF SEG 命令

```
DEF SEG=<セグメントベース>
```

セグメントの先頭番地の指定は実際の物理アドレス(絶対アドレス)を 16 で割った値で行います。これをセグメントベースといいます。たとえば、セグメントの先頭番地を物理アドレスの C0000H に設定したいときには、

```
DEF SEG=&HC000
```

と指定します。この宣言をすると、セグメントは物理アドレスの C0000H 番地から始まる 64K バイト分のメモリということになります。



---

**注意：**セグメントベースは物理アドレスを 16 で割った値ですから、セグメントの先頭番地の物理アドレスは必ず 16 の倍数になります。

---

セグメントの設定後、そのセグメント内の特定番地を指定するには相対アドレスを使用します。相対アドレスの指定可能範囲は 0000H~FFFFH です。これにより、セグメントベースにより決定されたセグメントの先頭番地から 64K バイト分のメモリ中の任意のアドレスをアクセスすることができます。前の例でいえば、C0000H~CFFFFH 番地が指定できることとなります。

次の命令は、相対アドレスを利用するため、実行前に DEF SEG 命令によるセグメントの指定が必要です。

CALL, BLOAD, BSAVE, DEF USR, PEEK, POKE

セグメントの概念は機械語を使う上でしっかり理解する必要があります。以下の解説では実際の命令のなかでセグメントがどのように使われているのかに注意してください。

## 8.2 機械語プログラムの準備

### ■機械語プログラムが使用するメモリ領域の確保

機械語プログラムを使う場合にまず最初にやらなければならないことは、機械語プログラム用のメモリ領域と BASIC プログラム用のメモリ領域を別に確保することです。もし両者のメモリ領域が重複すると、BASIC の動作が異常になったり、機械語プログラムが失われてしまうこととなります。

機械語プログラムのメモリ領域と BASIC のメモリ領域の境界の宣言、つまり機械語プログラムのメモリ領域の確保は CLEAR 命令で行います。この命令は DEF SEG 命令より先に実行しなければなりません。

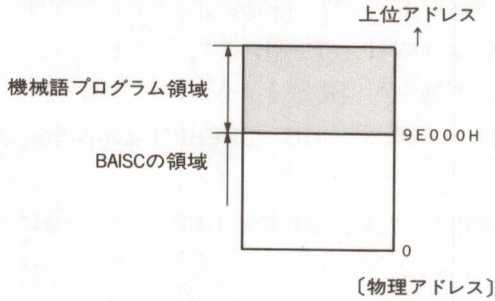
#### CLEAR 命令

CLEAR [**<ダミーパラメータ>**][**<メモリの上限>**][**<スタックの大きさ>**][**<配列データ領域の大きさ>**]

2 番目のパラメータ<メモリの上限>で、機械語プログラム用のメモリ領域の確保を行います。<メモリの上限>はセグメントベース(物理アドレスを 16 で割った値)で指定します。たとえば、

CLEAR , &H9E00

とすると BASIC のメモリ領域は物理アドレスの &H9DFFF 番地までとなり、&H9E000 以降が機械語プログラムで使用できるメモリ領域となります。



### ■使用するセグメントの宣言

CLEAR 命令による機械語のメモリ領域の確保が終わったら、実際に機械語プログラムをメモリ上に書き込むための準備として、DEF SEG 命令を使ってセグメントを宣言します。

CLEAR 命令の例に合わせた例をあげると、次のようになります。

```
DEF SEG=&H9E00
```

この命令の実行によって、セグメントは物理アドレスの9E000Hから9EFFFHまでの64Kバイトの範囲になります。この命令を実行するとDEF SEG 命令を使ってセグメントを変更しない限り、メモリへの読み書きはこのセグメントが対象になります。

このセグメントベースは、機械語プログラム実行時にCSレジスタに入ります。

### ■機械語プログラムをメモリに置く方法

機械語プログラムを実行するには、まずプログラムをメモリ上に置かなければなりません。そのための方法として次の3つが考えられます。

#### ● MON 命令を使う方法

MON 命令を使ってモニタモードに移り、機械語モニタの機能を利用してプログラムを書き込むものです。この方法についてはモニタモードの章を参照してください。

▶参照 モニタモード→「11.2 モニタモード」

#### ● POKE 命令を使う方法

POKE 命令でプログラムを書き込む方法です。

POKE 命令は、データを直接メモリ上に書き込むための命令です。

#### POKE 命令

```
POKE <アドレス>, <式>
```

POKE 命令は、<アドレス>に指定した番地に<式>の値を書き込みます。<アドレス>には、直前に実行されたDEF SEG 命令で宣言されたセグメントベースからの相対アドレスを指

定します。比較的短いプログラムなら、DATA 行で機械語プログラムを用意し、READ 命令で読み取り、POKE 命令でメモリに書き込むことができます。

### ● BLOAD 命令を使う方法

BLOAD 命令は、ディスクにすでにファイルとしてセーブされた機械語プログラムをメモリ上にロードするときに使います。

#### BLOAD 命令

**BLOAD** <ファイルディスクリプタ> [, <ロードアドレス>][, R]

BLOAD 命令は、<ファイルディスクリプタ> で指定されたファイル名の機械語プログラムをメモリ上にロードします。<ロードアドレス> には、機械語ファイルを読み込む先頭の番地を、直前に実行された DEF SEG 命令で宣言されたセグメントベースからの相対アドレスで指定します。<ロードアドレス> を省略した場合には、直前に実行された DEF SEG 命令で宣言されたセグメントベースに、機械語プログラムをセーブする際に BSAVE 命令で指定した開始アドレス(相対アドレス)を加えた番地からロードが行われます。

#### ▶ 参照 R オプション→8.3の「BLOAD 命令を使う方法」

BLOAD 命令を使うには、あらかじめ機械語プログラムがディスクにセーブされていなければなりません。セーブには基本的に2つの方法があります。

- アセンブラなどのソフトを使う。
- MON 命令や POKE 命令で機械語プログラムをメモリに書き込んだ後、BSAVE 命令でセーブする。

アセンブラを使用する方法については、そのアセンブラの説明に従ってください。

BSAVE 命令による方法は、次のとおりです。

#### BSAVE 命令

**BSAVE** <ファイルディスクリプタ>, <開始アドレス>, <長さ>

BSAVE 命令は、メモリ上の機械語プログラムを<ファイルディスクリプタ>で指定されたファイル名でディスクにセーブします。<開始アドレス> には、セーブを開始する番地を、直前に実行された DEF SEG 命令で宣言されたセグメントベースからの相対アドレスで指定します。<長さ> にはセーブする機械語プログラムのバイト数を指定します。

## 8.3 機械語プログラムの実行

BASIC から機械語プログラムを使う方法は3つあります。

- USR 関数を使う
- CALL 命令を使う
- BLOAD 命令を使う

これらの方法について、以降の項でそれぞれ別に解説します。

### ■ USR 関数を使う方法

USR 関数は、メモリ上に置かれた機械語プログラムを、関数の形で実行するものです。

USR 関数を使用するには、まず機械語プログラムの実行開始アドレスを USR 関数そのものにセットしなくてはなりません。これを機械語関数の定義といい、DEF USR 命令を用いて行います。

#### DEF USR 命令

```
DEF USR[<番号>]=<開始アドレス>
```

<番号>には0から9までの数値を指定することができます。<開始アドレス>には機械語プログラムの開始アドレスを相対アドレス(直前のDEF SEG 命令で定義したセグメントベースからの相対アドレス)で指定します。

たとえば次のようにすると、

```
DEF USR0=&HD000
```

セグメントベースからの相対アドレス D000H のところから始まる機械語プログラムが0番の機械語関数、すなわちUSR0として定義されます。このようにして、最大10個の機械語関数(USR0~USR9)を定義して実行することができます。

なお、機械語関数の定義は、機械語プログラムをメモリへロードする前でも、また後でもかまいません。ロード前の場合には機械語プログラムをロードする番地を想定して、定義を行います。

DEF USR 命令は機械語関数の定義を行うだけです。実際に機械語プログラムを実行させるにはUSR関数を使います。

USR関数を次の書式で用いると機械語プログラムに制御が移ります。

## USR 関数

〈変数〉=USR〔〈番号〉〕(〈引数〉)

〈番号〉には DEF USR 命令で定義された機械語関数の番号を指定します。省略すると 0 を指定したのと同様になります。

〈引数〉には機械語プログラムに引き渡す変数、定数、式を指定します。〈引数〉は省略できませんので、機械語プログラムに値を引き渡す必要がない場合には何らかの値をダミーとして指定します。

〈変数〉には 〈引数〉と同じ型の変数を指定してください。〈変数〉には機械語プログラムの実行後その実行結果が格納されます。

なお、USR 関数は、BASIC 上で実行されるだけでメモリ上の機械語プログラムに制御が移ります。すなわち、必ずしも“〈変数〉=USRn(x)”の形で実行する必要はなく、たとえば“PRINT USRn(x)”としたり、他の関数や命令のパラメータとして使用したりしてもかまいません。

機械語プログラム側では、その実行結果を特定のメモリ領域に入れておくようにします。こうすると、機械語プログラムの実行が終了した際、〈変数〉に結果の値が入れられた状態で引き続き BASIC のプログラムが実行されます。この特定のメモリ領域については次の項で説明します。

---

**注意：**機械語プログラムを終了したあとで制御を BASIC にもどすためには、機械語プログラムの最後で必ず機械語の IRET 命令を用いなければなりません。

---

### ●機械語プログラムと BASIC との間でデータを交換するためのメモリ領域のアドレス

BASIC から渡された 〈引数〉を機械語プログラム上で使用するには、〈引数〉の値がどこのメモリ領域に置かれているかを知る必要があります。また、機械語プログラムが実行されたあとで BASIC に値を返すには、機械語側でどこのメモリ領域に値を入れればよいかを知る必要があります。

実際には、どちらの場合も同じアドレスを使用します。次にこのアドレスを実際に求める方法について説明します。

#### 〈引数〉の内容を機械語プログラムで使う場合

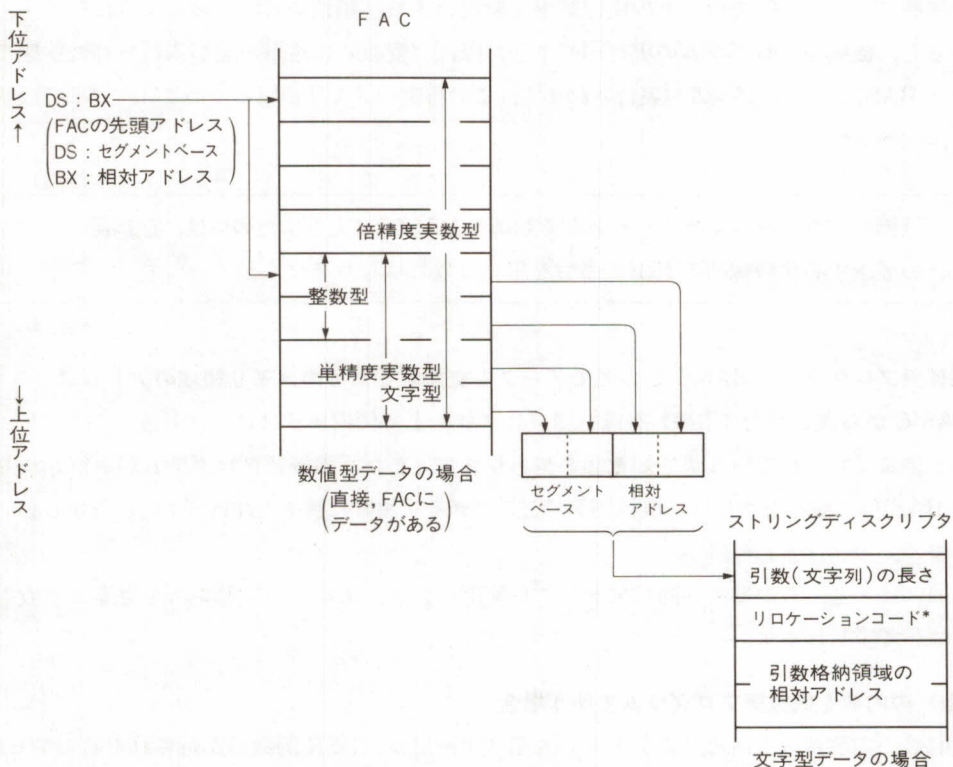
〈引数〉の内容が入るメモリのアドレスを示すデータは、USR 関数が実行された時点で CPU の AL レジスタ、DS レジスタ、BX レジスタに入れられます。したがって機械語プログラムでは、これらのレジスタから得たアドレスによって示されるメモリの内容を読み出すことにより、〈引数〉の値を得ることができます。

〈引数〉のアドレス情報を表す各レジスタの値

| 引数の型   | AL レジスタ | DS レジスタ  | BX レジスタ   |
|--------|---------|--|---|
| 整数型    | 2       | FACがあるテキストセグメントのセグメントベース<br>(引数の内容が入っている領域のセグメントベース) | FACの5バイト目の相対アドレス<br>(引数の内容が入っている領域のセグメントベースからの相対アドレス) |
| 単精度実数型 | 4       |  |   |
| 文字型    | 3       |  |   |
| 倍精度実数型 | 8       |  | FACの1バイト目の相対アドレス<br>(引数の内容が入っている領域のセグメントベースからの相対アドレス) |

表中、FACとは“浮動小数点アキュムレータ”のことで、BASICが算術演算などを行う際に使用する8バイトの作業領域を表します。

〈引数〉の内容は、その型によって次のようにFACへの格納のされ方が異なります。



\*リロケーションコードは文字列が格納されているセグメントの情報を表す。  
 &H00 : ストリング格納エリア(シンボルテーブルセグメント内)……DXレジスタの値がセグメントベース  
 &H02 : テキストセグメント  
 &H04 : I/Oファイルバッファ(テキストセグメント内) } ……&H0060がセグメントベース

▶参照 データの形式→「付録B データの内部形式」

#### 機械語プログラムで BASIC に値を返す場合

機械語プログラムから BASIC に値を返す際に使用されるメモリ領域は、USR 関数の実行時に〈引数〉が格納された領域とまったく同じです。したがって、BASIC にもどる際には、機械語プログラムで求められた値を FAC 中に設定し、さらに AL レジスタにその値の型を設定することが必要です。なお、返す値の型は、〈引数〉と同じでなければなりません。

#### ●機械語関数を使ったプログラム例

機械語関数の使用例として、文字列中に含まれる小文字を大文字に変換して返す機械語プログラムを含んだ BASIC プログラムを示します。DATA 行で機械語プログラムを用意しておき、POKE 命令を使ってメモリに書き込んでいます。

機械語プログラムは、次に示すような手順で、与えられた引数自身を直接変更します。

- ① AL レジスタの値を調べて与えられた引数が文字型かどうかを調べ、文字型でない場合にはそのまま BASIC にもどる。
- ② 引数が文字型なら、BX レジスタと DS レジスタで与えられた FAC の内容からストリングディスクリプタの番地を求める。
- ③ ストリングディスクリプタの番地を見て文字列の長さや文字列の置かれている番地を求める。
- ④ 文字列を最初から1文字ずつ調べ、小文字の時には5ビット目を0にすることにより大文字に変換する。

```

100 CLEAR,&H7F00
110 MACHINE=&H0
120 DEF SEG=&H7F00
130 DEF USR0=MACHINE
140 '
150 FOR ADR=MACHINE TO MACHINE+49
160 READ BYTE:POKE ADR,BYTE 'poke ML program
170 NEXT ADR
180 '
190 INPUT A$
200 B$=USR0(A$)
210 PRINT B$
220 GOTO 190
230 END
240 '
250 ' machine language program data
260 '
270 DATA &H3C,&H03          : ' CMP    AL,03H      ; string?
280 DATA &H75,&H2D          : ' JNZ    STRCNV90   ; jump no
290 DATA &H53                : ' PUSH   BX
300 DATA &H51                : ' PUSH   CX
310 DATA &H52                : ' PUSH   DX
320 DATA &H56                : ' PUSH   SI
330 DATA &HC5,&H1F          : ' LDS    BX,[BX]
340 DATA &H8B,&H0F          : ' MOV    CX,[BX]    ; CH:rel no. CL:data le
neth
350 DATA &H80,&HFD,&H00      : ' CMP    CH,00H
360 DATA &H74,&H05          : ' JZ     STRCNV10   ; jump yes
370 DATA &H32,&HED          : ' XOR    CH,CH
380 DATA &HBA,&H60,&H00      : ' MOV    DX,0060H
390 ' STRCNV10 : '
400 DATA &H8B,&H77,&H02      : ' MOV    SI,2[BX]   ; get offset
410 DATA &H8E,&HDA          : ' MOV    DS,DX
420 ' STRCNV20 : '
430 DATA &H8A,&H24          : ' MOV    AH,[SI]
440 DATA &H80,&HFC,&H61      : ' CMP    AH,61H    ; cmp. small "a"
450 DATA &H72,&H08          : ' JC     STRCNV30   ; jump less
460 DATA &H80,&HFC,&H7B      : ' CMP    AH,7BH    ; cmp. small "z"
470 DATA &H73,&H03          : ' JNC    STRCNV30   ; jump greater or equal
480 DATA &H80,&H24,&HDF      : ' AND    BYTE PTR [SI],0DFH ;convert
490 ' STRCNV30 : '
500 DATA &H46                : ' INC    SI
510 DATA &HE2,&HEE          : ' LOOP   STRCNV20
520 DATA &H5E                : ' POP    SI
530 DATA &H5A                : ' POP    DX
540 DATA &H59                : ' POP    CX
550 DATA &H5B                : ' POP    BX
560 ' STRCNV90 : '
570 DATA &HCF                : ' IRET

run
? a
A
? cdefg
CDEFG
?
Break in 190
Ok

```



機械語プログラムから BASIC に値を返す場合にもこのアドレスを使用します。すなわち、引数のデータを機械語で書き換えることにより、BASIC にもどったときの引数(変数)の内容を変えるわけです。ただし、引数の型は、CALL 命令を実行する前と後で同じでなければなりません。したがって、機械語プログラムに引き渡し引数と BASIC に返す(受け渡し)引数の個数や型が異なる場合には、引き渡し専用または受け渡し専用の引数を指定するなどの対策を講じるようにしてください。

CALL 命令による〈引数〉のやりとりで気をつけなければならないことは、引数の型とその個数が機械語プログラムに渡されないことです。機械語プログラムを作るときには BASIC との整合性に注意してください。

---

**注意：**機械語プログラムを終了したあとで制御を BASIC にもどすためには、機械語プログラムの最後で必ず機械語の IRET 命令を用いなければなりません。

---

### ● CALL 命令を使ったプログラム例

CALL 命令の使用例として、2つの数のうち最大値を与える機械語プログラムを含んだ BASIC プログラムを示します。

機械語プログラムは、BX レジスタの示す引数テーブルから A % と B % の内容を取り出し、大きな方を C % の番地に格納します。

---

```

100 CLEAR ,&H7D00
110 DEF SEG=&H7D00
120 MAX=&H0
130 '
140 FOR ADR=MAX TO MAX+47
150   READ BYTE:POKE ADR,BYTE 'poke ML language
160 NEXT ADR
170 '
180 C%=0
190 INPUT A%,B%
200   CALL MAX(A%,B%,C%)
210   PRINT C%
220 GOTO 190
230 END
240 '
250 ' machine language program data
260 '
270 '
280 DATA &H50           MAX:  :' PUSH  AX
290 DATA &H53           :' PUSH  BX
300 DATA &H51           :' PUSH  CX
310 DATA &H56           :' PUSH  SI
320 DATA &H06           :' PUSH  ES
330 DATA &H8B,&H4F,&H0A  :' MOV   CX,10[BX]   ; get 1st base addr.
340 DATA &H8E,&HC1      :' MOV   ES,CX
350 DATA &H8B,&H77,&H08  :' MOV   SI,8[BX]    ; get 1st offset addr.
360 DATA &H26,&H8B,&H04  :' MOV   AX,ES:[SI]  ; get 1st data
370 DATA &H8B,&H4F,&H06  :' MOV   CX,6[BX]   ; get 2nd base addr.
380 DATA &H8E,&HC1      :' MOV   ES,CX

```

```

390 DATA &H8B,&H77,&H04      : ' MOV    SI,4[BX]      ; get 2nd offset addr.
400 DATA &H26,&H3B,&H04      : ' CMP    AX,ES:[SI]   ; compare data
410 DATA &H7D,&H03          : ' JGE    CMP10        ; jump 1st >=2nd
420 DATA &H26,&H8B,&H04      : ' MOV    AX,ES:[SI]   ; move 2nd to AX
430 '                          CMP10: '
440 DATA &H8B,&H4F,&H02      : ' MOV    CX,2[BX]     ; get 3rd base addr.
450 DATA &H8E,&HC1          : ' MOV    ES,CX
460 DATA &H8B,&H37          : ' MOV    SI,0[BX]     ; get 3rd offset addr.
470 DATA &H26,&H89,&H04      : ' MOV    ES:[SI],AX   ; move AX to 3rd area
480 DATA &H07              : ' POP    ES
490 DATA &H5E              : ' POP    SI
500 DATA &H59              : ' POP    CX
510 DATA &H5B              : ' POP    BX
520 DATA &H58              : ' POP    AX
530 DATA &HCF              : ' IRET

```

```

run
? 100,200
200
? 100,-200
100
?
Break in 190
Ok

```

### ■ BLOAD 命令を使う方法

BLOAD 命令に R オプションを指定して実行すると、機械語プログラムのロード後ただちにその先頭番地から実行が開始されます。したがって指定する機械語プログラムのファイルは実行可能なプログラムでなければなりません。この場合には BASIC から引数を渡すことや BASIC に実行結果を返すことはできません。



# 第9章

## 拡張機能

N<sub>88</sub>-BASIC(86)には、次のような拡張機能が用意されています。これらの機能は機械語を使って制御できるだけでなく、BASICの拡張命令を使って取り扱うことができます。

- 電話制御機能
- サウンド制御機能
- GP-IB(IEEE-488)インタフェース制御機能
- RS-232Cインタフェース(2回線/3回線)制御機能

これらの拡張機能を使用する上での条件は次のとおりです。

| 拡張機能                                    | 必要な機器および<br>インタフェースボード  | 利用者メモリの圧迫                  |
|---|---|----------------------------|
| 電話制御機能                                  | PC-TL101, PC-TL102L,<br>PC-9863N, PC-9865,<br>PC-9801-57のいずれか | 利用者領域が<br>20KB減少           |
| サウンド制御機能                                | PC-9801-26K<br>サウンドボード  | プログラム領域(テキスト<br>エリア)が2KB減少 |
| GP-IB(IEEE-488)<br>インタフェース<br>制御機能      | PC-9801-29N<br>GP-IB(IEEE-488)<br>インタフェースボード                  | なし                         |
| RS-232C<br>インタフェース<br>(2回線/3回線)<br>制御機能 | PC-9861K RS-232C<br>(2回線/3回線)<br>インタフェースボード                   | プログラム領域(テキスト<br>エリア)が1KB減少 |

それぞれの機能を使うには、ユーティリティ `switch.n88` を使用して拡張機能の設定をする必要があります。

この4つの機能のうち、サウンド制御機能、GP-IB(IEEE-488)インタフェース制御機能とRS-232Cインタフェース(2回線/3回線)制御機能を使用するには、別売のインタフェースボードを装着する必要があります。なお、機種によってはサウンドボードを内蔵しているものがあります。その場合は、インタフェースボードを取り付ける必要はありません。

電話制御機能を使用するには、指定されたモデムまたは電話機が接続されていなければなりません。

この章では、電話制御機能とサウンド制御機能についての解説を行います。

GP-IB(IEEE-488)インタフェース制御機能とRS-232Cインタフェース(2回線/3回線)制御機能の取り扱いについてはそれぞれのインタフェースボードに添付されている取扱説明書をご覧ください。

▶参照 switch.n88 →「10.14 メモリスイッチの設定」

## 9.1 電話制御機能

N<sub>88</sub>-BASIC(86)には、次のようなモデム-NCU内蔵電話機やモデムなどを扱うための、拡張機能が用意されています。

N<sub>88</sub>-日本語 BASIC(86)で制御のできるモデム-NCU内蔵電話機・モデム

|               |  |   |
|---------------|--|---|
| モデム-NCU内蔵電話機  | PC-TL101 オートホン<br>PC-TL102Lオートホン                             | 1200ボー(半二重), 300ボー(全二重)<br>1200ボー(全二重), 300ボー(全二重)  |
| モデムボード+ハンドセット | PC-9863N+PC-TL901<br>PC-9865+PC-TL901<br>PC-9801-57+PC-TL901 | 1200ボー(半二重), 300ボー(全二重)<br>1200ボー(全二重), 300ボー(全二重)<br>2400ボー(全二重), 1200ボー(全二重),<br>300ボー(全二重) |

これらのモデム-NCU内蔵電話機やモデムをコンピュータ本体に接続すれば、N<sub>88</sub>-BASIC(86)の拡張命令を利用して、オートダイヤル、自動着信、モデムを利用したデータ通信などを行うことができます。

また、ユーティリティ“tele.n88”(電話管理ユーティリティ)を使用すれば次のようなことができます。

- ・電話帳の作成や保守
- ・電話の自動発信
- ・BBSへの接続
- ・ファイル転送

各電話機やモデムについての説明は、それぞれの取扱説明書をご覧ください。

## ■電話制御のための拡張命令

N<sub>88</sub>-BASIC(86)には次のような拡張命令および関数が用意されています。

| 分類            | 命令・関数  | 機能   |
|---------------|--|--|
| 電話機の<br>制御命令  | CMD LINE OPEN<br>CMD LINE CLOSE<br>CMD DIAL<br>CMD STORE DIAL*<br>CMD ON LINE GOSUB<br>CMD LINE ON/OFF/STOP<br>CMD RECEIVE<br>CMD MODE CUT | 電話機と BASIC を論理的に接続する<br>電話機と BASIC を論理的に切り離す<br>電話をかける<br>電話機に電話番号を記憶させる<br>着信があった時の割り込み先を定義する<br>着信割り込みを制御する<br>着信があった場合の動作を指定する<br>電話を切る |
| 電話機に<br>対する関数 | STATUS LINE<br>STATUS DIAL \$*<br>STATUS DIAL*<br>STATUS MODE  | 着信があったかどうかを調べる<br>電話機に記憶されている電話番号を調べる<br>電話機に記憶されている電話番号の機能を調べる<br>現在の電話機のモードを調べる  |
| 通信関係<br>の命令   | CMD CHANGE DUPLEX<br>CMD BREAK<br>CMD ERROR ON/OFF/STOP<br>CMD ON ERROR GOSUB  | 通信方式を切り換える<br>ブレイク信号を送出する<br>通信エラー割り込みを制御する<br>通信エラーがあった時の割り込み先を定義する   |
| 通信関係<br>の関数   | STATUS ERROR   | 通信エラーの有無あるいはエラーの種類を調べる   |

\*の付いている命令および関数はモデムボードにおいては使用できません。

▶参照 電話制御命令→『BASICリファレンスマニュアル』

## ■電話制御命令の使用に際して

### ●電話機のスイッチ設定

各モデム-NCU 内蔵電話機、モデムボード、ハンドセットのユーザーズマニュアルを参照しながら、ボーレート(転送速度)、発信/受信モードなどを設定してください。

### ●電話機とコンピュータ本体の接続

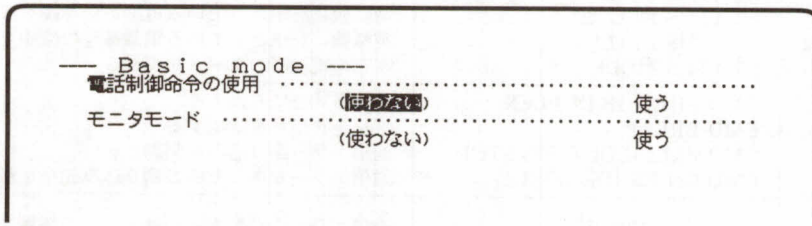
電話機とコンピュータ本体の RS-232C ポートを、電話機に付属している RS-232C ケーブルで確実に接続してください。RS-232C ポートは標準ポート(第1回線)および拡張ポート(第2回線/第3回線)のいずれも使用できますが、拡張ポートを使用するには、RS-232C 拡張インタフェースボードが必要です。

次に、モデム-NCU 内蔵電話機の場合、電話機の自動/手動ボタンを押して“自動”のランプが点灯するようにセットしてください。手動の場合、オートダイヤル、自動着信などの制御は行えません。

●電話制御命令の使用宣言

ユーティリティ "switch.n88" を使って電話制御命令を使用するための設定を行います。この設定を行わないと、電話制御命令は使用できません。実行しようとするとき "Feature not available" のエラーメッセージが表示されます。

- ①まず、本体のディップスイッチ SW2 の 5 番を ON にした状態で BASIC を起動させます。
- ②システムディスクをドライブ 1 にセットします。
- ③"menu" プログラムを起動させ "メモリスイッチの設定 (switch.n88)" を選択します。
- ④"switch.n88" が起動しますので、"Basic mode" を選択します。



- ⑤"電話制御命令の使用" で "使う" にカーソルを位置づけて  キーを押します。
- ⑥ キーを押すと初期画面にもどりますので "終了" を選択します。ここで、リセットスイッチを押して、BASIC を再起動すると電話制御命令が使用できる状態になります。

▶参照 switch.n88 →「10.14 メモリスイッチの設定」

---

注意：電話制御命令の使用を宣言すると、BASIC のプログラム領域が約 20KB 減少します。

---

●ボーレートの指定

RS-232C 標準ポート (第 1 回線) 使用の場合



第 1 回線を使用する場合、ユーティリティ "switch.n88" を使用して、電話機のボーレートと同じ値に設定します。

- ①まず、本体のディップスイッチ SW2 の 5 番を ON にした状態で BASIC を起動させます。
- ②システムディスクをドライブ 1 にセットします。
- ③"menu" プログラムを起動させ "メモリスイッチの設定" を選択します。
- ④"switch.n88" が起動しますので、"RS-232C (初期設定)" を選択します。すると画面に次のように表示されます。

| — RS-232C (初期設定) — |               |             |                     |
|--------------------|---------------|-------------|---------------------|
| Xパラメータ             | .....         |             |                     |
| 通信方式               | ( <u>二重</u> ) | 有効          |                     |
| データbit長            | (全二重)         | 半二重         |                     |
| パリティ チェック          | (7 bit)       | 8 bit       |                     |
| ストップbit長           | (使わない)        | 奇数          | 偶数                  |
| ボーレート              | (1 bit)       | 1.5 bit     | 2 bit               |
|                    | 75<br>(1200)  | 300<br>2400 | 600<br>4800<br>9600 |

|       |       |            |
|-------|-------|------------|
| ( )   | ..... | 現在の値を示します  |
| ESC   | ---   | 最初の画面に戻ります |
| HELP  | ---   | 説明をします     |
| ↑ ↓ ↔ | ---   | 他を選択します    |
| ←     | ---   | 決定します      |

- ⑤ ボーレートの“300”、“1200”または“2400”にカーソルを位置づけて  キーを押します。
- ⑥  キーを押すと初期画面にもどりますので“終了”を選択します。ここで、リセットスイッチを押して、BASICを再起動すると電話制御命令が使用できる状態になります。

注意：ここではボーレート以外の設定はしなくてかまいません。その他のプロトコルの設定は OPEN 命令でコミュニケーションファイルをオープンする際に設定してください。

#### RS-232C 拡張ポート (第2回線あるいは第3回線) を使用する場合

第2回線あるいは第3回線を使用する場合、RS-232C 拡張インタフェースボード上の転送速度設定スイッチの値を電話機のボーレートと同じ値に設定します。

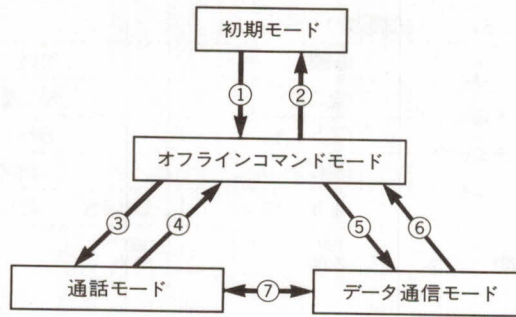
#### ■ 4つのモード

電話制御命令を使い、電話回線を通じてデータ通信などを行う場合には、BASIC、電話機、電話回線を論理的に接続します。この論理的な接続の状態には段階的な4つのレベルがあり、次に説明する4つのモードに分けて考えることができます。

たとえば、電話回線を通じてデータ通信を行う場合には、電話制御命令を使って、「初期モード→オフラインコマンドモード→データ通信モード」へと進んでいきます。

それぞれのモードでは実行できる電話制御命令が異なりますので注意してください。

4つのモードの関係



- ① : CMD LINE OPENの実行
- ② : CMD LINE CLOSEの実行
- ③ : CMD DIALの実行, 手動ダイヤルまたは手動着信の操作
- ④ : 受話器を置く, フックボタンを押す
- ⑤ : CMD DIALの実行, 自動着信の操作
- ⑥ : CMD MODE CUTの実行
- ⑦ : 通話/データ切り換えボタンによる操作

●初期モード

BASIC と電話機が論理的に接続されていない状態です。この状態ではオートダイヤル、自動着信などの機能を利用することができません。

手動による電話回線の接続あるいはモデム機能のみを利用したデータ通信が可能です。

●オフラインコマンドモード

CMD LINE OPEN 命令を実行すると電話機と BASIC が論理的に接続され、オートダイヤル、自動着信などの機能が利用できるようになります。このモードを“オフラインコマンドモード”といいます。この状態では、電話機と電話回線は切り離されていますが、BASIC と電話機との間で命令の受け渡しができます。

CMD LINE CLOSE 命令を実行すると初期モードにもどります。

●データ通信モード

オフラインコマンドモードから、データ通信モード指定の CMD DIAL 命令を実行するかあるいは自動着信が行われると電話回線と BASIC がモデムを介して接続された状態になります。これを“データ通信モード”といいます。この状態ではコミュニケーションファイルを通じて電話回線上の通信相手とデータの受け渡しをすることができます。このモードのときには電話機のフックランプとデータランプがともに点灯しています。

CMD MODE CUT 命令を実行するとオフラインコマンドモードにもどります。

また、データ通信モードで受話器を外し、通話/データ切り替えボタンを押すことにより通話モードに移行することができます。

### ●通話モード

オフラインコマンドモードから、通話モード指定の CMD DIAL 命令を実行するかあるいは手動ダイヤル、手動着信を行うと、電話回線と電話機とが接続され通話相手と話すことができます。この状態を“通話モード”といいます。このモードのときには電話機のフックランプが点灯し、データランプが消灯しています。

このモードでは BASIC から電話機を制御することはできません。受話器を置くかフックボタンを押すことによって電話を切るとオフラインコマンドモードにもどります。

また、このモードで通話/データ切り替えボタンを押すことによりデータ通信モードに移行することができます。

| モード              | 使用できる命令・関数   |
|------------------|--|
| 初期モード            | CMD LINE OPEN 命令<br>STATUS MODE 関数<br>CMD CHANGE DUPLEX 命令<br>コミュニケーションファイルに対する入出力命令<br>(OPEN/CLOSE/PRINT #/INPUT #など)   |
| オフライン<br>コマンドモード | CMD LINE CLOSE 命令<br>CMD STORE DIAL 命令*<br>CMD RECEIVE 命令<br>CMD LINE ON/OFF/STOP 命令<br>CMD MODE CUT 命令<br>CMD ON LINE GOSUB 命令<br>STATUS LINE 関数<br>STATUS DIAL\$関数*<br>STATUS DIAL 関数*<br>STATUS MODE 関数 |
| データ通信モード         | CMD CHANGE DUPLEX 命令<br>CMD BREAK 命令<br>CMD MODE CUT 命令<br>CMD LINE CLOSE 命令<br>STATUS ERROR 関数<br>STATUS MODE 関数<br>コミュニケーションファイルに対する入出力命令<br>(OPEN/CLOSE/PRINT #/INPUT #など)                              |
| 通話モード            | STATUS MODE 関数   |

\*の付いている命令あるいは関数はモデムボードでは使用できません。

## ■電話制御命令を使ったプログラム例

### ●電話をかけて通話を行う

自動ダイヤルで相手に電話をかけるプログラムです。

---

```


100 ' AUTO DIAL
110 CMD LINE OPEN "COM:"
120 *MAIN
130 INPUT "Telephone number";DL$
140 IF DL$="" THEN CMD LINE CLOSE : END
150 CMD DIAL DL$
160 GOTO *MAIN

```

---

110行で電話機と BASIC を論理的に接続し、初期モードからオフラインコマンドモードに切り換えます。`COM:`は、コンピュータ本体と電話機が RS-232C 標準ポートで接続されていることを意味します。

130行で相手の電話番号を入力します。内線電話から外線にかける場合に電話番号の前に`0:`などをつける必要がある場合には`0:XXXX`のように入力します。

電話番号の入力の際に  キーだけを押し、140行で電話機と BASIC を論理的に切り離し、初期モードにもどしてプログラムを終了します。

150行でダイヤル後、通話モード指定で電話をかけます。ダイヤルが終わると次の命令に進みます。この場合再び電話番号入力待ちとなります。

オートダイヤルを行うと電話器のフックランプが自動的に点灯し、相手呼び出し始めます。この間、内蔵スピーカを通して呼び出し音が確認できます。相手が出たら受話器をとって会話をします。通話が終わった場合、または話し中あるいは相手が出ないなどの場合には受話器を置くかフックボタンを押して電話を切ってください。

### ●電話をかけてデータを送信する

自動ダイヤルで電話をかけファイルの内容を転送するプログラムです。このプログラムは、受信側が次の「自動着信後データを受信する」プログラムを起動させていることを想定しています。

---

```

100 ' AUTO DIAL & SEND FILE
110 CMD LINE OPEN "COM:"
120 *MAIN
130 INPUT "Telephone number";DL$
140 IF DL$="" THEN CMD LINE CLOSE : END
150 INPUT "Send file name";FLN$
160 OPEN FLN$ FOR INPUT AS #1
170 CMD DIAL DL$,1
180 OPEN "COM:E71N" AS #2
190 PRINT #2,FLN$
200 WHILE (NOT EOF(1))
210 PRINT #2,"1"
220 LINE INPUT #1,A$

```


```

230 PRINT #2,A$
240 WEND
250 PRINT #2,"0"
260 CLOSE
270 CMD MODE CUT
280 GOTO *MAIN

```

---

まず 110 行で電話機と BASIC を論理的に接続し、初期モードからオフラインコマンドモードに切り換えます。

次に 130 行で相手の電話番号を入力します。電話番号の入力の際に  キーだけを押し、電話機と BASIC を論理的に切り離し初期モードにもどしてプログラムを終了します。

150, 160 行では、ディスク中の転送するファイル名を入力し、そのファイルをオープンします。

170 行で電話をかけますが、電話番号の次に "1" が指定されているのでダイアル後相手のアンサートーンを検出するまで待っています。この命令は送信先の相手が「自動着信後データを受信する」のプログラムを起動して受信待ちの状態になっていなければなりません。そうでない場合には規定時間内にアンサートーンが検出できないのでエラーとなります。

電話回線が接続され、アンサートーンを検出したら、180 行で電話機に接続されているコミュニケーションファイルをオープンします。この命令を実行することにより以降、モデムを介して PRINT # , INPUT # などの入出力命令を実行することが可能となります。

なお、半二重による通信を行う場合、OPEN 命令の前に次の命令を実行してください。

#### CMD CHANGE DUPLEX 1, 1

この命令は通信方式を全二重通信方式から半二重通信方式に切り換える命令です。

190 行～250 行では、転送するファイル名に続いてファイルのデータを順次相手に送信します。このプログラムでは、始め "1" を送り、続けてファイルの内容を 1 行分送る、ということを繰り返しています。ファイルの中身の転送が終わると "0" を送ります。これはデータの転送が終わったことを知らせる意味を持っています。

データの転送が終わると、260 行で電話機に接続しているコミュニケーションファイルと転送したデータファイルをクローズします。

最後に 270 行で電話を切りオフラインコマンドモードに移行させます。

#### ●自動着信後データを受信する

自動着信を行い、受信したデータをディスクファイルに書き出すプログラムです。このプログラムは、送信側が「電話をかけてデータを送信する」プログラムを起動させて電話をかけることを想定しています。

---

```
100 ' RECEIVE FILE
110 CMD LINE OPEN "COM:"
120 CMD ON LINE GOSUB *REC
130 CMD LINE ON
140 *LOOP
150 GOTO *LOOP
160 *REC
170 CMD RECEIVE 1
180 IF STATUS MODE<>2 THEN 180
190 OPEN "COM:E71N" AS #1
200 LINE INPUT #1,FLN$
210 PRINT "RECEIVE FILE=";FLN$
220 PRINT "START"
230 OPEN FLN$ FOR OUTPUT AS #2
240 *REC10
250 INPUT #1,A$
260 IF A$<>"1" THEN *REC20
270 LINE INPUT #1,A$
280 PRINT #2,A$
290 GOTO *REC10
300 *REC20
310 CLOSE
320 CMD MODE CUT
330 CMD RECEIVE 0
340 PRINT "END"
350 RETURN
```

---

まず、110行で電話機と BASIC を論理的に接続し、初期モードからオフラインコマンドモードにします。

120, 130 行では着信があった場合の分岐先を定義し、着信割り込みを許可します。これ以降着信があるとすぐ \* REC ルーチンへ分岐します。

その後、着信があるまで待っています。もちろんこの間に別の処理を実行させることも可能です。

170 行では着信があるとアンサートーンを返し、その後データ通信モードに移行します。このとき、180 行の STATUS MODE 関数によってデータ通信モードであることを確認してから、190 行で電話機に接続されているコミュニケーションファイルを開きます。以降、モデムを介して PRINT#, INPUT # などの入出力命令が実行可能となります。

なお、半二重で通信を行う場合、OPEN 命令の前に次の命令を実行してください。

#### CMD CHANGE DUPLEX 1, 1

この命令は通信方式を全二重通信方式から半二重通信方式に切り換える命令です。

まず、200 行でコミュニケーションファイルから、受信するファイルの名前を受け取ります。その後、210 行から 230 行でファイル名と受信を始める合図として、“START” と表示し、そのファイル名のディスクファイルを出力モードでオープンします。

240 行から 290 行ではコミュニケーションファイルから順次データを受信し、ファイルに書き込みます。

この場合、“1”の次にはファイルの内容が1行送られてくるので、これをファイルに書き込むということを繰り返します。250行で“1”以外を受信した場合には電話を切る処理の\*REC 20ルーチンにジャンプします。

310行～330行はファイルをクローズし、電話を切り、自動着信を禁止にするサブルーチンです。つまり、再び着信割り込み処理ルーチンに分岐するまではデータ通信モードに切り換わらないようにします。そして“END”という受信終了のメッセージを表示して割り込み前の処理(着信待ち)にもどします。

### ■電話制御命令使用時の注意

- PC-TL101, PC-TL102Lオートホーン, PC-9863N, PC-9865, PC-9801-57モデムボード以外の電話機を接続した場合の動作は保証できません。
- データ通信モードのとき BASIC のエラーが発生したり **STOP** キーを押してプログラムの実行が中断した場合には自動的に電話が切れます。
- データ通信中になんらかの事情により通信できなくなった場合、永久に相手からのデータを待っている状態に陥る可能性があります。無人運転を行う際は時間監視を行い、一定の時間内にデータが送られてこない場合には電話を切るというようなプログラムを作成してください。
- 電話制御命令を使用しているときにはコミュニケーションポートに対して OUT 命令を実行しないでください。動作を保証できません。
- CMD LINE OPEN命令でコミュニケーションポートを接続した場合、そのポートに対応するコミュニケーションファイルは必ずデータ通信モード内で、OPEN命令とCLOSE命令を実行してください。オープンしたままデータ通信モードを抜け出し再度データ通信モードとなると、コミュニケーションファイルに対する入出力命令が誤動作する可能性があります。
- 異なった種類の電話機を接続してデータ通信を行う場合、正常動作しなければ受信側のモード設定スイッチは ANS モードに設定してください。また、送信側は AUTO あるいは CALL にセッティングします。同一機種を接続する場合、両者とも AUTO モードでかまいません。

### ■電話管理ユーティリティ “tele.n88”の使い方

システムディスクには電話管理ユーティリティ “tele.n88”が入っています。

電話管理ユーティリティは次の4つの機能を持っています。

- 電話帳管理機能
- オートダイアル通話機能
- ファイル転送機能
- ファイル受信機能

それぞれについて解説します。

### ●電話帳管理機能

オートダイヤルを行うために通信先の電話番号や名前を電話帳ファイルに登録します。

電話帳ファイルには氏名、電話番号をはじめ住所、所属等のコメント情報を登録しておくことができます。

また、通信先がBBS(掲示板サービス)や各種のデータベースである場合には、接続のための通信条件を設定することもできます。

電話帳管理機能には、電話帳ファイルの作成および電話帳ファイルの保守を行うための4つの機能が用意されています。

- ・発信相手の新規登録
- ・氏名、電話番号、属性などの変更
- ・発信相手の削除
- ・電話帳の表示

### ●オートダイヤル通話機能

電話帳ファイルに登録されている通話先を選択するだけで、自動的に電話をかけることができます。通話先がBBS(掲示板サービス)や各種データベースの場合には、オートダイヤル後、あらかじめ設定されている通信条件に従ってターミナルモードへ移行します。ターミナルモードにした後は、相手サービスとの規約に従って通信を行ってください。

### ●ファイル転送機能

ファイルを通話先に転送します。

エラーチェック付きのファイル転送機能ですから、確実にデータを送ることができます。なお、ファイル受信側はこのユーティリティプログラムの「ファイル受信機能」を使用していなければなりません。

### ●ファイル受信機能

前の項の「ファイル転送機能」を使って送られてくるファイルデータを受信し、ファイルに格納します。

## ■電話管理ユーティリティの使用に際して

### ●電話機の準備

「電話制御命令の使用に際して」に従って電話機の準備を行ってください。なお、電話管理ユーティリティはRS-232C標準ポートを使用しますので、コンピュータ本体と電話機の接続はRS-232C標準ポートを使用してください。

### ●電話制御命令の使用宣言

電話管理ユーティリティは電話制御命令を使用してオートダイヤルの制御を行います。このため、電話制御命令の使用宣言が必要です。前出の「電話制御命令の使用に際して」に従って準備してください。

---

**注意：**電話管理ユーティリティは日本語表示を行いますので、使用の際には 400 ラインの高解像度ディスプレイが必要です。


---

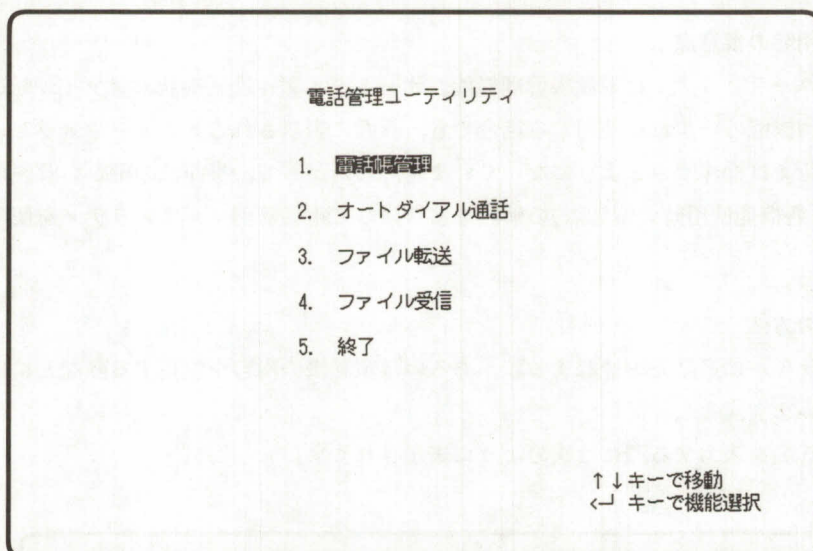
## ■電話管理ユーティリティの操作

### ●電話管理ユーティリティの起動

電話管理ユーティリティはシステムディスク内の “tele.n88” というプログラムです。

メニュープログラム “menu” を起動させ、“電話管理ユーティリティ (tele.n88)” を選択します。

電話管理ユーティリティが起動されると次のメニュー画面が表示されます。カーソルを選択する機能に位置づけて  キーを押してください。



### ●電話帳ファイルの作成と電話帳ファイルの指定

ファイル受信機能以外の機能を使用する場合、電話帳ファイルが必要です。電話帳ファイルは“TBOOK.SYS”というファイル名で作成されます。電話帳ファイルは52セクタ分の容量を持ち、104人分の情報を登録することができます。

電話帳管理機能を選択した場合、処理に入る前に次のようなメッセージが表示されます。

#### 電話管理ユーティリティ

電話帳ファイル(TBOOK.SYS)が存在するドライブのドライブ番号を入力して下さい。  
なお電話帳ファイルを新規に作成したい場合、HELPキーを押して下さい。電話帳ファイル  
の新規作成が行えます。  
?

はじめて電話管理ユーティリティを使用する場合や、なんらかの理由で電話帳ファイルの内容が破壊されてしまった場合には、**HELP** キーを押してください。電話帳ファイルの作成に進みます。

電話帳を作成ずみの場合は、ドライブ番号を指定するとそのドライブに存在する電話帳ファイルが保守の対象となります。

### ●各機能使用時の留意点

電話管理ユーティリティは電話帳管理機能、オートダイアル通話機能、ファイル転送機能、ファイル受信機能のいずれを使用する場合でも、画面に表示されるメニューやメッセージに従って作業を行えば操作できるようになっています。したがって、操作法の細かい解説は行いません。この「各機能使用時の留意点」の解説を参考にして電話管理ユーティリティを使用してください。

### 日本語入力の方法

電話帳ファイルに通話先を登録する際、あるいは電話帳の内容を変更する際などには、日本語入力を行います。

通話先の氏名を入力する際には次のように表示されます。

#### 発信相手の新規登録

氏名を入力して下さい。氏名は日本語(12文字以内)で入力します。  
?

このような場合は、自動的に日本語入力モードになります。日本語入力の方法はそのときの BASIC の日本語入力環境に依存します。

▶参照 日本語入力の方法→5.3の「日本語文字の入力」

### 電話番号

電話帳に登録する電話番号は16桁以内の数字、コロン(:)およびハイフン(-)で指定します。

コロンを使うと約2秒待ってから残りの番号をダイヤルします。これは構内交換機(PBX)から外線への接続要求を行う必要がある場合などに使用します。

ハイフンは形式的な区切り記号です。省略してもかまいません。

例1) 次の3つの電話番号はすべて同じ意味に解釈される。

XX-XXX-XXXX      XXXXXXXXX      XX-XXXXXXX

例2) 次の3つの電話番号はすべて同じ意味に解釈される。

X:XX-XXX-XXXX    X:XXXXXXXXXX    X:XX-XXXXXXX

### オートダイヤル

オートダイヤル時には、特に次の点に注意してください。

- 電話機の自動/手動スイッチを“自動”にしておきます。
- 受話器は電話機に置いた状態にしておきます。
- 電話機のモードスイッチは、AUTO あるいは CALL にセッティングしておきます。
- コンピュータ本体のポーレート指定と電話機の手動スイッチの値を一致させます。

### ● BBS/データベースなどへのアクセス法

BBS にアクセスする方法を概説します。BBS アクセスには「電話帳管理機能」と「オートダイヤル通話機能」を使います。

#### 電話帳管理機能で行うこと

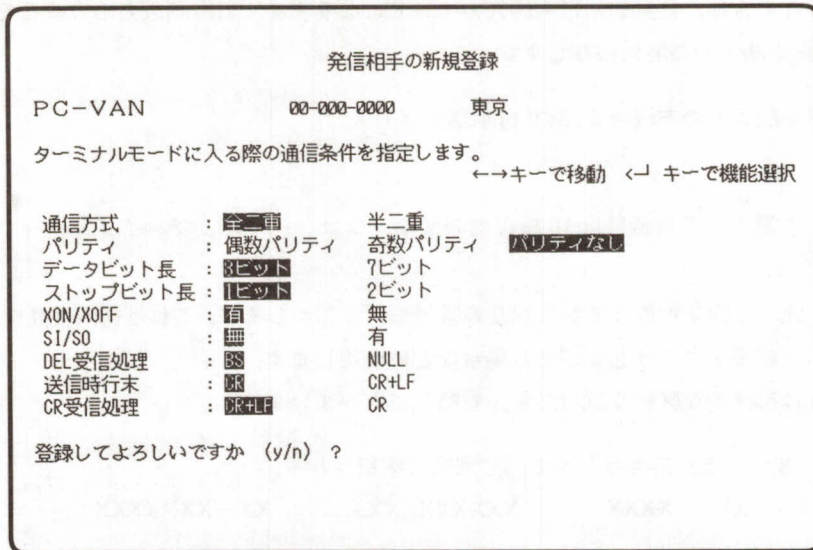
① BBS へアクセスすることを宣言する。

電話帳ファイルに通話先を登録する際、“発信相手は BBS ですか(y/n)?” の問いに対して、“y”を入力してください。

② BBS アクセスのための通信条件を登録しておく。

通信条件の指定は次のようなメニューから選ぶだけの簡単な操作で行えます。

BBS アクセス時の通信条件は各サービスによって異なりますので、対象とする BBS の通信条件をよく調べた上で指定してください。



オートダイアル通話機能で行うこと

①オートダイアル通話機能を使用してオートダイアルします。

BBSの電話番号が通話先に選ばれると、まず、次のように転送速度(ボーレート)の問い合わせが行われます。

BBSにアクセスします。転送速度を指定して下さい。

- 1. 300 ボー    2. 1200 ボー    3. 2400ボー

アクセスするBBSサービスの規約に従って転送速度を指定します。このとき電話機の色速度スイッチで指定した転送速度(ボーレート)と同じに設定されていることを確認してください。

②転送速度の設定が終わったら、次のように表示されます。

オートダイアルします。呼び出し音に続き「ピー」という連続音が確認できたら、電話機をデータモードにしてください。

オートダイアルが始まったら受話器を取り、呼び出し音を確認します。相手が話し中の場合は受話器を置き **STOP** キーを押してください。プログラムが終了します。

呼び出し音に続く「ピー」という連続音が確認できたら、接続成功です。ここで電話機の通話／データスイッチを押して通話モードからデータ通信モードにします。それまでは、次のように表示されています。

ターミナルモードへ移行します。ターミナルモードへの移行後は、対象 BBS との規約に従って会話を行って下さい。

データ通信モードにすると、画面がクリアされてターミナルモードへ入り、BBS との取り決めにしたがって通信を開始します。通常、ユーザー ID やパスワードなどの問い合わせが行われますので、順次、応答してください。

▶参照 ターミナルモード→「11.1 ターミナルモード」

③通信を終えたいときには **SHIFT** + **STOP** キーを押します。

ターミナルモードから抜けオートダイアルの画面にもどります。

### ●ファイルを転送する方法

ファイルを転送するには、送信側が「ファイル転送機能」を、受信側が「ファイル受信機能」を実行させなければなりません。

プログラムファイル(アスキー形式およびバイナリ形式)、機械語ファイル、データファイルのすべてを転送することができます。

ファイル転送の手順は次のとおりです。

#### 送信側の手順

- ①電話管理ユーティリティの初期メニュー画面において、「ファイル転送」を選択します。
- ②次のようなガイドメッセージに従って、データの転送速度(ボーレート)、通信方式、転送するファイル名を指定します。

ファイル転送

転送速度を指定して下さい。  
 1. 300\* - 2. 1200\* - 3. 2400\* -  
 ? 2

通信方式を指定して下さい。  
 1. 全二重 2. 半二重  
 ? 1

転送するファイルのドライブ番号とファイル名を指定して下さい。  
 ドライブ番号 : 1  
 ファイル名 :

データの転送速度は300ボー、1200ボーおよび2400ボーのいずれかを選択します。300ボー以外を選択した場合、通信方式をたずねてきます。

PC-TL102L, PC-9865, PC-9867のいずれかを使用している場合、全二重方式を指定してください。

PC-TL101, PC-9863Nのいずれかを使用している場合、半二重方式を指定してください。

このとき、電話機の速度スイッチで指定した転送速度(ボーレート)と同じに設定されていることを確認してください。

③この後オートダイヤルの場合と同様に、次のように電話帳が表示されますので、ファイルを送る相手側の電話番号を選択してください。オートダイヤルで相手を呼び出します。

|               |             |         |     |
|---------------|-------------|---------|-----|
| <b>PC-VAN</b> | 00-000-0000 | 東京      | BBS |
| 日本電気株式会社      | 00-000-0000 | 本社      |     |
| 北海道支社         | 00-000-0000 | 北海道ブロック |     |
| 東北支社          | 00-000-0000 | 東北ブロック  |     |
| 東京支社          | 00-000-0000 | 首都圏ブロック |     |
| 中部支社          | 00-000-0000 | 中部ブロック  |     |
| 北陸支社          | 00-000-0000 | 北陸ブロック  |     |
| 関西支社          | 00-000-0000 | 関西ブロック  |     |

↑↓キーで通話先のエントリにカーサを位置付けて下さい。通話先にカーサを位置付けたら←キーを押して下さい。↑↓←以外のキーを押すと次の画面に進みます。

④電話が通じたら相手にファイルを受け取る準備をさせます。


⑤受信側の①～⑤までの準備ができて、受信側がデータ通信モードになったら、送信側は受信側のキャリア音(ピーという連続音)が鳴るので、送信側の通話/データスイッチを押してデータ通信モードにし、受話器を置きます。するとファイルの転送が始まります。必ず、受信側のキャリア音を確認してから電話機をデータ通信モードにしてください。

#### 受信側の準備

送信側から電話がかかったら次のようにファイルを受け取る準備をします。

①電話を受け取ったらすぐに電話管理ユーティリティを起動し、「ファイル受信」を選択してください。以下の操作の間は受話器を電話機から上げて置き、切らないようにしてください。

②発信側の転送速度(ボーレート)および通信方式を電話で確認し、同じ条件を指定します。このとき電話機の速度スイッチも同じ速度に設定します。

- ③電話機のモードスイッチを AUTO にセッティングします。ただし、送信側と受信側の電話機の種類が異なる場合、受信側は ANS モードにセッティングしてください。
- ④ファイルデータを格納する際のファイル名を指定します。発信側が送ってくるファイル名と同じ名前登録してよければ  キーを押します。

#### ファイル受信

発信側の転送速度を確認し、それと同じ転送速度を選んで下さい。

1. 300\* - 2. 1200\* - 3. 2400\* -

? 2

通信方式を指定して下さい。


1. 全二重 2. 半二重

? 1

受信ファイルを格納するドライブを指定して下さい。

? 1

受信ファイルをドライブに格納する際にファイル名を変更したい場合、そのファイル名を指定して下さい。

発信側から送られてくるファイル名と同じ名前登録してよければ、 キーのみ応答して下さい。


?

- ⑤発信側に受信の準備ができたことを電話で伝え、通話/データスイッチを押して電話機を通話モードからデータ通信モードにした後、受話器を置きます。

#### ●通話異常時の原因と処置

電話管理ユーティリティを使用している際に異常が起きた場合には、次のような原因が考えられますのでチェックしてください。

[状況] オートダイヤルができないとき。

[対策] オートダイヤル時、無応答になったり、通信異常エラーが発生した場合、 キーを押して、オートダイヤル処理を中断した後、もう一度オートダイヤルをやり直します。それでもうまくいかない場合、次の点を確認してください。

- ・電話機が本体に正しく接続されているかどうかを確認します。特に電話機のモードスイッチが AUTO あるいは CALL にセットされているかどうかを確かめてください。
- ・電話制御命令の使用宣言が行われているかどうか、ユーティリティ "switch.n88" で確認します。
- ・電話機が自動モード(自動ランプが点灯)になっているかどうかを確認します。手動モードではオートダイヤルはできません。

[状況] オートダイヤルはできるが通話できないとき。

[対策] 次の点を確認します。

- 電話機が通話モード(通話ランプが点灯)になっているかどうかを確認します。データ通信モード(データランプが点灯)では通話はできません。

[状況] オートダイヤル後、ターミナルモードにはなるがBBSとの通信がうまくいかないとき。

[対策] 次の点を確認します。

- 電話機がデータ通信モードになっているかどうかを確認します。
- 対象BBSのキャリア音を確認した後、データ通信モードにしたかどうかを確認めます。
- 電話機の世界速度スイッチが、BBSアクセス時に指定した転送速度(ボーレート)と同じ速度にセッティングされているかどうかを確認します。
- 通信方式(全二重/半二重)、データビット長(8ビット/7ビット)、パリティチェックの有無などの通信条件が、対象BBSの規約どおりになっているかを確認します。

[状況] オートダイヤル後、通話ではうまくいったがファイル転送がうまくいかないとき。

[対策] 次の点を確認します。

- 受信側の電話機が正しくコンピュータ本体に接続されているかどうかを確認します。
- 送信側/受信側とも、電話機の世界速度スイッチがユーティリティで指定した値と同じ速度にセッティングされているかどうかを確認します。
- 受信側のキャリア音を確認してから送信側の電話機をデータ通信モードにしたかどうかを確認めます。
- 電話機のモードスイッチがAUTOになっているかどうかを確認します。AUTOでうまくいかない場合、送信側をCALL、受信側をANSにしてみてください。

▶ 参照 switch.n88 →「10.14 メモリスイッチの設定」

## 9.2 サウンド制御機能

サウンド発生機能は、FM音源とSSG音源を利用していろいろな音を発生することのできるものです。N<sub>88</sub>-BASIC(86)では、このサウンド発生機能を制御することのできる拡張機能が用意されています。ここではサウンド制御機能を使うために準備すべきこと、サウンド制御命令の概要、FM音源とSSG音源の仕組みなどについて解説しています。

▶参照 サウンド制御命令→『BASICリファレンスマニュアル』

### ■サウンド制御機能の特徴

#### ●基本機能

- 6重和音(FM音源：3声、SSG音源：3声)による音楽演奏が可能。
- 8オクターブに渡る音域の発生が可能。
- ビブラート、トレモロなどの効果音制御が可能。
- オーディオ出力端子をオーディオアンプ、ラジオカセットなど外部オーディオ機器に接続することができる。外部オーディオ機器を接続しない場合、本体内のアンプ、スピーカが使用できる。

#### ●FM音源

- サウンド発生部にFM(Frequency Moduration)方式音源のLSIを使用。
- FM音源3声に対する多種多様な基本音色の提供により、音色番号を指定するだけで音楽演奏が可能。さらに独自に音色を作ることも可能。種々の効果音も用意されている。
- FM音源3声に対してそれぞれ別の音色を割り当てることができる。たとえばストリング系音色(バイオリン、チェロなど)による三重奏などもできる。

#### ●SSG音源

- SSG音源3声に対してエンベロープ形状とエンベロープ周期を設定することにより、音に音量の時間的变化をつけることができる。
- シンセサイザレジスタに直接値を指定することにより音色を変えることができる。

#### ●BASICによる音楽演奏

- N<sub>88</sub>-BASIC(86)のサウンド制御命令を使った音楽演奏が可能。
- バックグラウンド演奏が可能。たとえば、グラフィック画面に絵を描きながら音楽演奏を行うというような並列処理が可能。

## ■サウンド制御命令の使用準備

サウンド制御命令を使用するにはサウンドボードが実装されていなければなりません。また、ユーティリティ "switch.n88" を使ってサウンド制御命令を使用することを設定しなければなりません。こうすることによって、N<sub>88</sub>-BASIC(86)本体と、サウンドインタフェースボード上のROMに格納されているサウンド制御命令実行部がリンク(連結)され、PLAY命令、VOICE命令などのサウンド制御命令を使用することができます。サウンド機能が使用可能な状態でBASICを起動すると、使用しない状態でBASICを起動させたときに比べ、プログラム領域が約2KB減少します。また、サウンド機能を使用すると、インタプリタの処理速度が多少低下します。

---

注意：ユーティリティによる設定をしないでサウンド制御命令を実行すると、"Syntax error" が発生します。

---

## ■サウンド制御命令の概要

### ●初期設定…PLAY ALLOC 命令

サウンド制御命令では、演奏のための情報の大部分をPLAY命令で指定します。PLAY命令の情報はいったんサウンドバッファと呼ばれる領域に格納され、この領域から情報が引き出されながら演奏が行われます。

サウンドバッファは、ユーザーが必要な大きさだけ確保する必要があります。サウンドバッファの確保を行うのがPLAY ALLOC命令です。なお、PLAY ALLOC命令は音色、音量、テンポ、音の高さなどの初期化も行います。

### ●音楽データの表現…PLAY 命令

音楽演奏に必要な音程、長さ、テンポなどの情報を指示する命令としてPLAY命令があります。PLAY命令では6チャンネルの同時発音が可能です。また、MML(ミュージックマクロランゲージ) という表現を用いて、先に述べた基本的な情報の他に特殊効果や音色の変化など、多くの情報を与えることができます。

### ●音色の設定や変更…VOICE, VOICE COPY, VOICE LFO, PLAY, VOICE REGの各命令

#### FM音源の音色

69種82個の音色データが用意されています。また、ユーザーが自分の好みに合わせて音色を変化させたり、まったく最初から自分で新しく音色を作り出したりできるように音色操作のための機能も用意されています。VOICE, VOICE COPY, VOICE LFO命令などを使って、音色データの設定、読み出し、一時変更などが可能です。また、PLAY命令のYコマンドやVOICE REG命令で音色を変えることもできます。

### SSG 音源の音色

SSG 音源では VOICE, VOICE COPY 命令は使用できません。音色の変更は VOICE LFO 命令や PLAY 命令の Y コマンドや、VOICE REG 命令を使用します。

#### ●特殊効果の付加…VOICE LFO, VOICE REG の各命令

FM 音源, SSG 音源ともに特殊な演奏効果として、音量や音程にゆらぎを与える効果(LFO 効果)が用意されています。FM 音源ではこの効果のためのデータを音色データの一部として設定可能です。また、VOICE LFO 命令で一時的に変更することも可能です。

SSG 音源のみに使用できる演奏効果として、音量を時間変化させるエンベロープコントロールがあります。これには、PLAY 命令の S コマンド、M コマンドや、VOICE REG 命令を使用します。

#### ●サウンドハードウェア(シンセサイザ LSI)の直接制御…VOICE REG 命令

サウンドハードウェアに使用されているシンセサイザ LSI(YM-2203)の内部レジスタを操作してより細かな音創りや演奏を可能にする命令として、VOICE REG 命令が用意されています。FM 音源, SSG 音源の両方に使用できます。

#### ●他の BASIC 命令との並行動作…ON PLAY GOSUB, PLAY ON/OFF/STOP の各命令

サウンド制御命令を N<sub>88</sub>-BASIC(86)の他の命令と並行して実行させることができます。たとえば、音楽を演奏させながら絵を描くといったことが可能です。また、長い曲を複数の PLAY 命令に分けて演奏させる時など、音楽が途切れないようになめらかに演奏させることができます。このような場合、ON PLAY GOSUB, PLAY ON/OFF/STOP 命令などを使用します。

### ■サウンドバッファについて

#### ●サウンドバッファサイズの決定

PLAY 命令を含んだプログラムを実行すると、PLAY 命令ごとに、指定された MML 文字列がサウンドバッファに格納され、演奏が行われます。

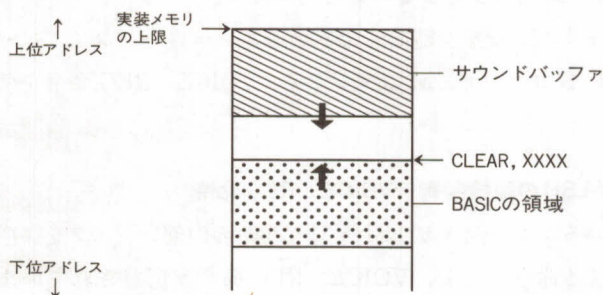
したがって、確保するサウンドバッファのサイズは、メモリに余裕があれば、プログラム中に存在する PLAY 命令のうち最も情報(MML 文字列)の多いものにそのサイズを合わせておくと、演奏がなめらかに行われます。

しかし、メモリに余裕がない場合には、サウンドバッファサイズを小さくして、一つの曲のデータをいくつもの部分(PLAY 命令)に分割して表現する必要があります。

なお、一つの曲を複数の部分に分割した場合、このプログラムをフォアグラウンドで実行すると、途中でとぎれたりしてなめらかに演奏できないことがあります。このような場合には、ON PLAY(c, n) GOSUB, STATUS PLAY などの命令を使って、サウンドバッファに残っている情報の量を監視しながら PLAY 命令を実行するようにしてください。

● BASICのメモリ領域とサウンドバッファの確保される領域

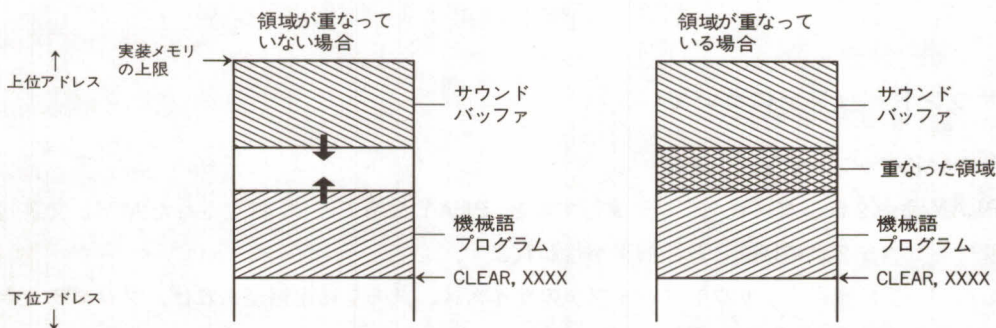
サウンドバッファは実装メモリの上限から下位アドレスに向かって確保されていきます。一方、BASICの使用する領域は CLEAR 命令によって上限を設定しない限り、実装メモリの上限にまでおよぶ可能性があります。したがって、確保したいサウンドバッファの量に応じて、BASICの使用するメモリの上限アドレスを CLEAR 命令の第2パラメータで宣言し、BASICの領域とサウンドバッファの領域が重ならないようにしてください。



▶ 参照 CLEAR命令→『BASICリファレンスマニュアル』

● 機械語プログラム領域とサウンドバッファの確保

サウンド制御命令と機械語を同時に使用する場合には、サウンドバッファと機械語プログラム領域が重ならないように注意してください。



注意：実装メモリの上限は、ユーティリティ "switch.n88" を使って設定してください。たとえば、メモリが 640KB 実装されていてもこの設定が 384KB になっている場合には実装メモリサイズは 384KB となります。

### ● サウンドバッファの変更

いったん確保したサウンドバッファは新たに PLAY ALLOC 命令が実行されるまでは有効です。変更後のサイズが CLEAR 命令により確保されているサイズを超えるような場合は再び CLEAR 命令を実行しなおす必要があります。

なお、PLAY ALLOC 命令でサウンドバッファのサイズを変更すると、演奏は中断され、初期状態に再設定されます。

### ● サウンドバッファと他の命令との関連

#### ERROR, PLAY ALLOC の各命令

バックグラウンドで演奏が行われているときに、これらの命令が実行されるとサウンドバッファが初期化されますので、演奏は中断されます。

#### STOP 命令, STOP キー

バックグラウンドでの演奏はいったん停止します。その後、バックグラウンドでの演奏に影響を与えるような命令が実行されず、かつ再実行可能であれば、CONT 命令でバックグラウンドでの演奏も再開されます。また、バックグラウンドでの演奏をいったん停止しているときに、ダイレクトモードで PLAY 命令を実行すると、サウンドバッファに新しいデータが格納されてから、演奏が再開されます。このデータを初期化するためには、PLAY ALLOC 命令を実行してください。

#### サウンドバッファの取扱いに関して

バックグラウンドによる演奏が行われている際に、BLOAD 命令あるいは POKE 命令によりサウンドバッファが乱されますと正しい演奏が行えませんので、アドレス指定には注意が必要です。

#### エラーが起きたとき

現在設定中の音色が初期化され演奏が中断されます。

## ■ FM 音源について

### ● FM 音源の原理

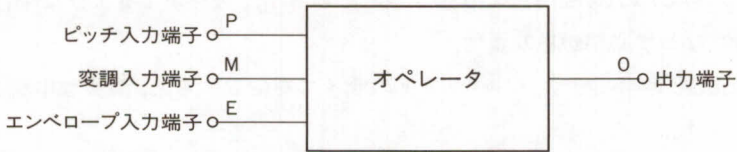
チャンネル 1~3 は FM 音源となっています。ここでは FM 音源についてその原理を簡単に説明します。

#### オペレータとアルゴリズム

FM 音源の基本単位となるものが“オペレータ”と呼ばれる回路です。

オペレータは、3 個の入力端子に入れられる信号を電氣的に加算したり乗算したりすることによって、波形を合成し、出力端子に出すしくみになっています。

次にオペレータを簡略化して示します。



ピッチ入力端子は、波形の周波数(高さ)を決めるための信号を入力するところです。

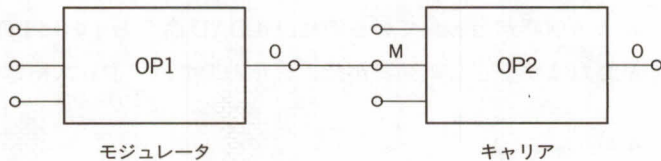
変調入力端子は、波形そのもの(形状)を変化させるための信号を入力するところです。

エンベロープ入力端子は、波形の振幅(強さ)を変化させるための信号を入力するところです。

出力端子は、得られた波形をとりだすところです。

3種類の入力信号をオペレータに入れてやると、出力端子からその入力信号に応じた波形が得られます。これらの入力信号のうち、エンベロープ入力だけはオペレータ外部の別の回路でつくってやります。ピッチ入力と変調入力は、とくに別の回路を使わず、同様な構成をもつ別のオペレータからの出力信号を用います。つまり、オペレータを組み合わせて使うわけです。これにより、ひとつのオペレータではつくることができない複雑な波形を得ることができるようになります。

たとえば、次のような組み合わせを考えてみます。



オペレータ OP1 の出力端子がオペレータ OP2 の変調入力端子に接続されています。これにより、オペレータ OP1 の出力信号は、オペレータ OP2 でつくられる波形そのもの(形状)を変化させるための変調信号となります。このような組み合わせの場合、オペレータ OP1 を“モジュレータ”、オペレータ OP2 を“キャリア”と呼びます。

このシンセサイザ LSI には、オペレータが各チャンネルごとに4個ずつ用意されており、これらの4個の組み合わせ方をいろいろ変えることができるようになっています。つまり、どのオペレータも、キャリアとしてあるいはモジュレータとして使えるようになっているわけです。この組み合わせ方のことを、“アルゴリズム”と呼び、全部で8種類用意されています。

FM 音源の最大の特徴はこのアルゴリズムの自由さにあり、これによって非常に複雑な波形をつくりだすことができるのです。

BASIC では、アルゴリズムを VOICE 命令で指定することができます。

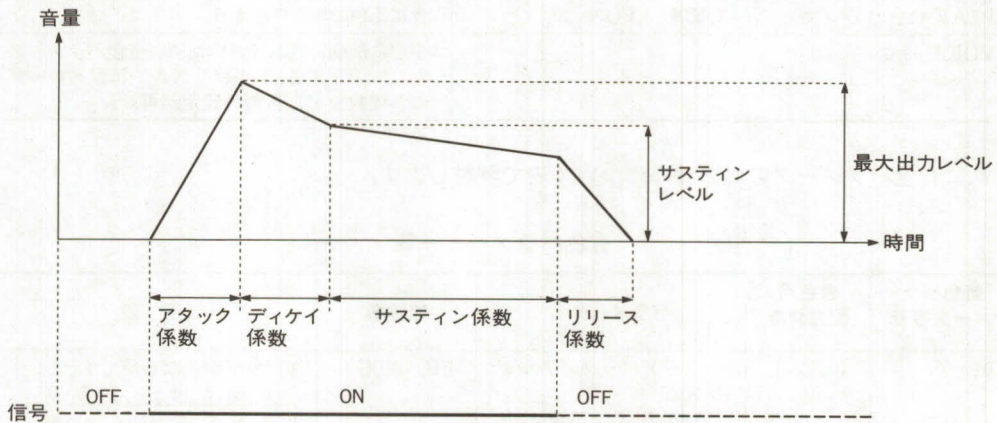
参考：FMとはFrequency Modulation(周波数変調)のことです。FM音源は、オペレータ中で変調を行う際にこの方式を用いることから、こう呼ばれます。

### エンベロープジェネレータ

前の項で、“エンベロープ入力だけは別の回路を使って信号をつくる”と説明しましたが、これを行う回路がエンベロープジェネレータ(EG)で、シンセサイザLSIに内蔵されています。

エンベロープジェネレータは音量の時間的変化を表す次のようなパラメータを持っており、音にさまざまな表情を付け加えることができます。

BASICで具体的にこれらのパラメータを指定するには、VOICE命令を使用します。



#### アタック係数(Attack Rate : AR)

発音してからEGの出力レベルが最大になるまでの時間を定めます。

最短：0，最長：31

#### ディケイ係数(Decay Rate : DR)

EGの出力レベルが最大からサスティンレベルになるまでの時間を定めます。

最短：0，最長：31

#### サスティン係数(Sustain Rate : SR)

サスティンレベルからEGの出力レベルが0になるまでの時間を定めます。

最短：0，最長：31

#### リリース係数(Release Rate : RR)

発音が終わってからEGの出力レベルが0になるまでの時間を定めます。

最短：0，最長：15

#### 最大出力レベル(Total Level : TL)

EGの最大出力レベルを定めます。

最小：0，最大：127

サスティンレベル(Sustain Level : SL)

Decay 状態から Sustain 状態に移るときのレベルを定めます。

最小 : 0, 最大 : 15

●音色パラメータの設定

サウンド制御命令には, FM 音源での音創りを容易に行うための命令が用意されています。これらの命令を使って, “音色パラメータ”(音色を決定する要素)を値として設定することにより音色を変更したり創り出したりすることができます。

| 音色を設定する命令                     | 機能  |
|-------------------------------|---|
| PLAY 命令の @X コマンド              | 各チャンネルに音色を設定する。   |
| PLAY 命令の Z コマンド, VOICE LFO 命令 | 音に LFO 効果を与える。  |
| VOICE 命令                      | 事前に配列に入れておいた値を音色パラメータとして設定する。アルゴリズムやエンベロープの各係数など, 各種の設定が可能。 |

次に音色パラメータのおのおのについて表で解説します。

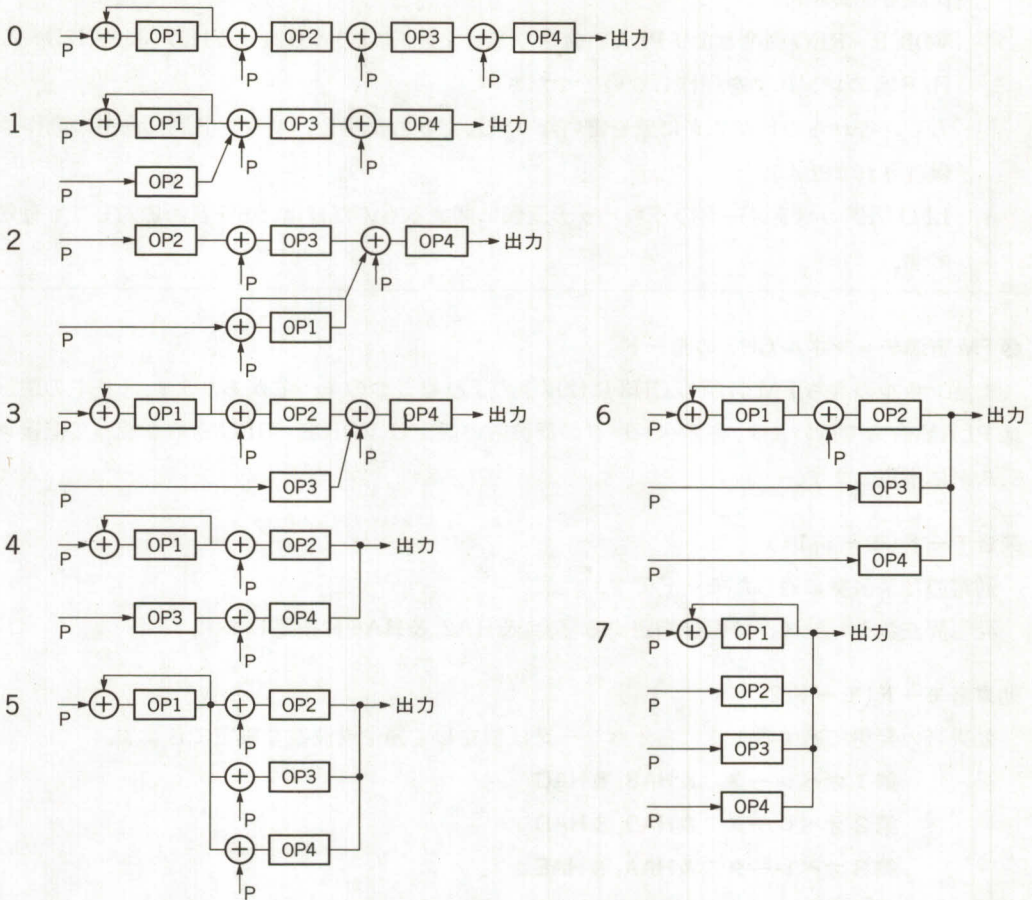
音色パラメーター一覧

| 音色パラメータ番号                      | 音色用配列要素 | パラメータ名                                  | 略称名            | 解説   |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
|--------------------------------|---------|---|----------------|--|----------------|----------------|----------------|---|---|---|---|---|---|---|----------------|----------------|----------------|----------------|----------------|----------------|--------------------------------|-----|---|-----|---|------|---|-----|---|-----|---|-----|---|---|---|----|---|----|
| 0                              | (0,0)   | フィードバック/アルゴリズム<br>(Feed Back/Algorithm) | FB/ALG         | <p>6ビットからなる値です。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>×</td><td>×</td><td>F<sub>2</sub></td><td>F<sub>1</sub></td><td>F<sub>0</sub></td><td>A<sub>2</sub></td><td>A<sub>1</sub></td><td>A<sub>0</sub></td> </tr> </table> <p style="text-align: center;"> <span style="margin-right: 40px;">フィードバック</span> <span>アルゴリズム</span> </p> <ul style="list-style-type: none"> <li>・フィードバック<br/>F<sub>2</sub>~F<sub>0</sub>の3ビットを使用します。第1オペレータは自分の出力を変調入力としていますが, その変調度を決定します。</li> </ul> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>F<sub>2</sub>~F<sub>0</sub></th> <th>変調度</th> </tr> </thead> <tbody> <tr><td>0</td><td>OFF</td></tr> <tr><td>1</td><td>π/16</td></tr> <tr><td>2</td><td>π/8</td></tr> <tr><td>3</td><td>π/4</td></tr> <tr><td>4</td><td>π/2</td></tr> <tr><td>5</td><td>π</td></tr> <tr><td>6</td><td>2π</td></tr> <tr><td>7</td><td>4π</td></tr> </tbody> </table> <ul style="list-style-type: none"> <li>・アルゴリズム<br/>A<sub>2</sub>~A<sub>0</sub>の3ビットを使用します。4つのオペレータのアルゴリズムを決定します。<br/>A<sub>2</sub>~A<sub>0</sub>の値とアルゴリズムの関係は「アルゴリズムの種類」の図参照。</li> </ul> | 7              | 6              | 5              | 4 | 3 | 2 | 1 | 0 | × | × | F <sub>2</sub> | F <sub>1</sub> | F <sub>0</sub> | A <sub>2</sub> | A <sub>1</sub> | A <sub>0</sub> | F <sub>2</sub> ~F <sub>0</sub> | 変調度 | 0 | OFF | 1 | π/16 | 2 | π/8 | 3 | π/4 | 4 | π/2 | 5 | π | 6 | 2π | 7 | 4π |
| 7                              | 6       | 5                                       | 4              | 3  | 2              | 1              | 0              |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| ×                              | ×       | F <sub>2</sub>                          | F <sub>1</sub> | F <sub>0</sub>   | A <sub>2</sub> | A <sub>1</sub> | A <sub>0</sub> |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| F <sub>2</sub> ~F <sub>0</sub> | 変調度     |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| 0                              | OFF     |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| 1                              | π/16    |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| 2                              | π/8     |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| 3                              | π/4     |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| 4                              | π/2     |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| 5                              | π       |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| 6                              | 2π      |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |
| 7                              | 4π      |   |                |  |                |                |                |   |   |   |   |   |   |   |                |                |                |                |                |                |                                |     |   |     |   |      |   |     |   |     |   |     |   |   |   |    |   |    |

| 音色パラメータ番号 | 音色用配列要素   | パラメータ名                            | 略称名   | 解説  |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
|-----------|-----------|-----------------------------------|-------|---|----------------|----------------|----------------|-------|---|-----|---|-----|---|-----------|---|-----------|----------------|----------------|----------------|----------------|
| 5         | (0,1)     | オペレータマスク<br>(Operater Mask)       | OPMSK | 4ビットからなる値です。<br>指定値：0~15<br><table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>×</td><td>×</td><td>×</td><td>×</td><td>O<sub>4</sub></td><td>O<sub>3</sub></td><td>O<sub>2</sub></td><td>O<sub>1</sub></td> </tr> </table> オペレータ ON/OFF<br>O <sub>4</sub> ~O <sub>1</sub> の4ビットを使用します。<br>各オペレータの使用/非使用を決定します。<br>O <sub>4</sub> ~O <sub>1</sub> の各ビットが"1"のとき使用,<br>"0"のとき非使用となります。 | 7              | 6              | 5              | 4     | 3 | 2   | 1 | 0   | × | ×         | × | ×         | O <sub>4</sub> | O <sub>3</sub> | O <sub>2</sub> | O <sub>1</sub> |
| 7         | 6         | 5                                 | 4     | 3   | 2              | 1              | 0              |       |   |     |   |     |   |           |   |           |                |                |                |                |
| ×         | ×         | ×                                 | ×     | O <sub>4</sub>  | O <sub>3</sub> | O <sub>2</sub> | O <sub>1</sub> |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 10        | (0,2)     | LFO 変調波形<br>(LFO Wave Form)       | WF    | LFO の変調波形を決定します。<br>指定値：0~5<br><table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>指定値</th> <th>波形</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>ノコギリ波</td> </tr> <tr> <td>1</td> <td>矩形波</td> </tr> <tr> <td>2</td> <td>三角波</td> </tr> <tr> <td>3</td> <td>サンプル&amp;ホールド</td> </tr> <tr> <td>4</td> <td>ノコギリ波ショット</td> </tr> <tr> <td>5</td> <td>三角波ワンショット</td> </tr> </tbody> </table>   | 指定値            | 波形             | 0              | ノコギリ波 | 1 | 矩形波 | 2 | 三角波 | 3 | サンプル&ホールド | 4 | ノコギリ波ショット | 5              | 三角波ワンショット      |                |                |
| 指定値       | 波形        |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 0         | ノコギリ波     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 1         | 矩形波       |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 2         | 三角波       |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 3         | サンプル&ホールド |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 4         | ノコギリ波ショット |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 5         | 三角波ワンショット |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 15        | (0,3)     | LFO Sync デイレイ<br>(LFO Sync Delay) | SYNC  | LFO の Sync デイレイタイムを決定します。<br>指定値：0~255  |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 20        | (0,4)     | LFO 速度<br>(LFO Speed)             | SPEED | LFO の速度を設定します。<br>指定値：0~16383   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 25        | (0,5)     | LFO ピッチ変調深さ<br>(LFO Pitch Depth)  | PD    | LFO ピッチ変調のかかる深さを細かく調整します。<br>指定値：-127~127   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 30        | (0,6)     | LFO 振幅変調の深さ<br>(LFO Amp Depth)    | AD    | LFO 振幅変調のかかる深さを細かく調整します。<br>指定値：-127~127  |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 35        | (0,7)     | LFO ピッチ変調深さ<br>(LFO Pitch Sense)  | PS    | LFO ピッチ変調のかかる深さを大まかに調整します。<br>指定値：0~15  |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 40        | (0,8)     | 未使用                               |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 45        | (0,9)     | 未使用                               |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 1         | (1,0)     | アタック係数<br>(Attack Rate)           | AR    | 各オペレータのアタック係数を設定します。<br>指定値：0~31  |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 2         | (2,0)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 3         | (3,0)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 4         | (4,0)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 6         | (1,1)     | ディケイ係数<br>(Decay Rate)            | DR    | 各オペレータのディケイ係数を設定します。<br>指定値：0~31  |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 7         | (2,1)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 8         | (3,1)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 9         | (4,1)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 11        | (1,2)     | サステイン係数<br>(Sustain Rate)         | SR    | 各オペレータのサステイン係数を設定します。<br>指定値：0~31   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 12        | (2,2)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 13        | (3,2)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |
| 14        | (4,2)     |                                   |       |   |                |                |                |       |   |     |   |     |   |           |   |           |                |                |                |                |

| 音色パラメータ番号            | 音色用配列要素                          | パラメータ名   | 略称名 | 解説   |
|----------------------|----------------------------------|--|-----|--|
| 16<br>17<br>18<br>19 | (1,3)<br>(2,3)<br>(3,3)<br>(4,3) | リリース係数<br>(Release Rate)                           | RR  | 各オベレータのリリース係数を設定します。<br>指定値：0～15   |
| 21<br>22<br>23<br>24 | (1,4)<br>(2,4)<br>(3,4)<br>(4,4) | サスティンレベル<br>(Sustain Level)                        | SL  | 各オベレータのサスティンレベルを設定します。<br>指定値：0～15   |
| 26<br>27<br>28<br>29 | (1,5)<br>(2,5)<br>(3,5)<br>(4,5) | 最大出力レベル<br>(Total Level)                           | TL  | 各オベレータの出力レベルを設定します。<br>指定値：0～127   |
| 31<br>32<br>33<br>34 | (1,6)<br>(2,6)<br>(3,6)<br>(4,6) | キーボードレイトスケーリング深さ<br>(Key-board Rate Scaling Depth) | KS  | 各オベレータのキーボードレイトスケーリング深さを設定します。<br>このパラメータは音に表情を与えるため、高い音になるほどエンベロープの各レイトを短くします。<br>指定値：0～3<br>0で最小、3で最大です。 |
| 36<br>37<br>38<br>39 | (1,7)<br>(2,7)<br>(3,7)<br>(4,7) | マルチプル<br>(Multiple)                                | ML  | 各オベレータに対して与える周波数の比を設定します。<br>FM音源の各オベレータには、基本となる周波数整数比倍の値を設定することができます。<br>指定値：0～15<br>0が1/2倍、15が15倍の周波数です。 |
| 41<br>42<br>43<br>44 | (1,8)<br>(2,8)<br>(3,8)<br>(4,8) | デチューンレイト<br>(Detune Rate)                          | DT  | 各オベレータのデチューンレイトを設定します。<br>このパラメータは、コーラス効果や微妙なゆらぎ効果を出すために、各オベレータの周波数をわずかにずらします。<br>指定値：-4～3                 |
| 46<br>47<br>48<br>49 | (1,9)<br>(2,9)<br>(3,9)<br>(4,9) | LFO 振幅変調深さ<br>(LFO Amp Sense)                      | AS  | 各オベレータのLFO 振幅変調深さを、大まかに設定します。<br>指定値：0～15  |

## アルゴリズムの種類



0～7は音色パラメータ番号0の $A_2$ ～ $A_0$ の値を示す。Pはピッチ入力を示す。

音色パラメータは、VOICE LOF 命令では有効範囲のチェックが行われますが、VOICE 命令やPLAY 命令のZ コマンドではチェックされません。そのため思ったような音が出ないことがありますので注意してください。

音色パラメータはシンセサイザ LSI の内部レジスタと密接に関係しており、LFO 関連のパラメータ以外是对应するレジスタが必ず1つは存在します。たとえば、パラメータ ML(マルチプル)とDT(デチューンレイト)は、内部レジスタの&H30～&H3E 番の下位4ビットと上位3ビットに対応します。その対応は「FM/SSG 音源内部レジスタマップ」の表に示してあります。VOICE REG 命令やPLAY 命令のY コマンドで直接内部レジスタを操作する場合には、この表を参考にしてください。

---

**注意：**OUT 命令によりシンセサイザ LSI レジスタを操作した場合、サウンド制御命令の動作は保証できません。

VOICE REG 命令および PLAY 命令の Y コマンドにより &H21～&H27 番、&H2D～&H2F 番のレジスタを操作しないでください。

もし、これらのレジスタに値を書いたり読んだりした場合、サウンド制御命令の動作は保証されません。

LFO 関係のパラメータのうちピッチ変調に関するものだけは、SSG 音源に対しても有効です。

---

### ● FM 音源チャンネル CH3 のモード

チャンネルのうち FM 音源の CH3 には以下のような 3 つのモードがあります。モードの選択は PLAY 命令で行います。各オペレータの周波数の設定は VOICE REG 命令を使って直接レジスタに指定します。

#### 楽音モード(モード 1)

通常の音楽演奏に適したモードです。

発生周波数をレジスタで直接指定する場合、&HA2、&HA6 に指定します。

#### 効果音モード(モード 2)

効果音の発生に適するように各オペレータは独立して発生周波数を設定できます。

第 1 オペレータ：&HA8、&HAC

第 2 オペレータ：&HA9、&HAD

第 3 オペレータ：&HAA、&HAE

第 4 オペレータ：&HA2、&HA6

#### CSM モード(モード 3)

発生周波数の設定は効果音モードと同様です。ただし、このモードの時には内部タイマレジスタ A のカウンタがオーバーフローしたときに音を発生します。

### ● FM 音源チャンネルの発生周波数の求め方

FM 音源チャンネル CH1～3 に与える周波数データ(Block および F-number)は次の式で計算します。発生する周波数と、その周波数のオクターブをもとにして、Block 値と F-number 値を計算します。

Block = <オクターブデータ>

F-number = (<発生する周波数(Hz)> \* 2<sup>20</sup> / 55556) / 2<sup>(Block-1)</sup>

例) PLAY 命令における O4A の音(440Hz)の場合、

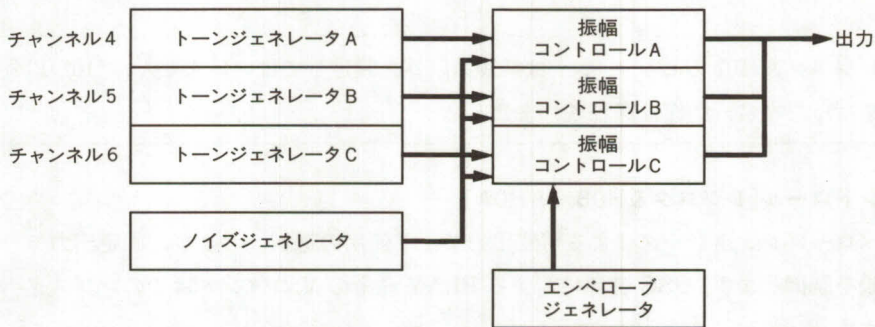
Block=4

$F\text{-number} = (440 * 2^{20} / 55556) / 2^{(4-1)} = 1038$

### ■ SSG 音源について

チャンネル 4~6 は SSG 音源となっています。

SSG 音源部の構成は次のようになっています。レジスタの値を設定することにより音質が変化します。VOICE REG 命令、PLAY 命令の Y コマンドで直接設定を行うことができます。「FM/SSG 音源内部レジスタマップ」を参照してください。なお、エンベロープ形状とエンベロープ周期は PLAY 命令の S コマンド、M コマンドで設定することもできます。



トーンジェネレータ：[レジスタ &H00~&H05]

デューティ比1:1の方形波を発生する発振器です。

各チャンネルの発振周波数は次の式により求められます。

$$F_{note} = 667 * 1000 / (32 * T_o)$$

$F_{note}$ ：発振周波数(Hz)

$T_o$ ：トーンジェネレータ設定値(0~4095)

ノイズジェネレータ：[レジスタ &H06]

類似ランダムノイズを発生します。レジスタ &H06 に設定される値によりノイズ周波数を可変できます。

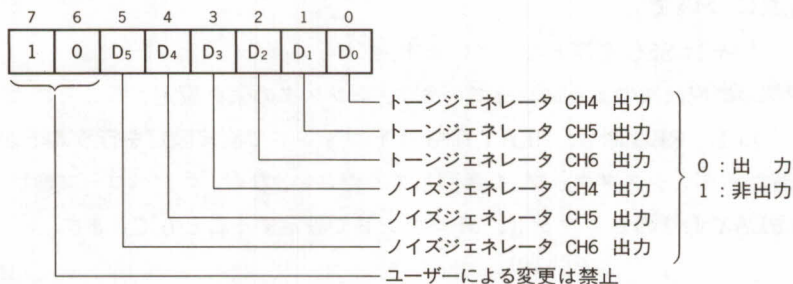
$$F_{nois} = 667 * 1000 / (32 * T_n)$$

$F_{nois}$ ：ノイズ周波数(乱数系列発生周期)

$T_n$ ：ノイズジェネレータ設定値(0~31)

ミキサコントロール：[レジスタ &H07]

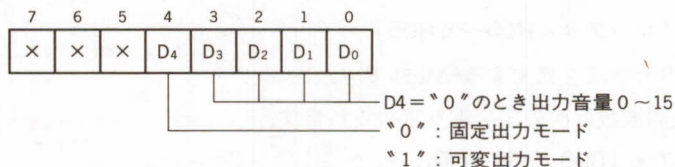
トーンジェネレータとノイズジェネレータを出力チャンネルに出すか否かを設定します。  
このレジスタは次のようなビット構成になっています。



注意：レジスタ &H07 のビット 6, 7 は "10" 以外を設定してはいけません, "10" 以外の設定を行った場合, 動作は保証されません。

振幅コントロール[レジスタ &H08～&H0A]

エンベロープジェネレータによる可変出力モード使用の選択, および, 固定出力モード時の出力音量を制御します。SSG 音源に対する PLAY 命令の V コマンドはこのレジスタへの設定を行います。



エンベロープジェネレータコントロール[レジスタ &H0B～&H0D]

変化のあるエンベロープを作り出すために 2 つの値を設定します。

1 つはレジスタ &H0B, &H0C に設定するエンベロープ周期で, PLAY 命令の Mx コマンドに対応します。

エンベロープ周波数は次の式で求められます。

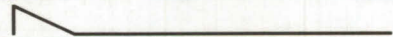







$$F_{env} = 667 * 1000 / (512 * T_e)$$

F<sub>env</sub> : エンベロープ周波数(Hz)

T<sub>e</sub> : エンベロープ周期設定値(0～65535)

もう1つは、レジスタ&H0Dに設定するエンベロープ形状でこれはPLAY命令のSxコマンドに対応します。

レジスタ&H0Dへの設定値とエンベロープの形状の関係は次のとおりです。

| 設定値       | 対応するエンベロープ   |
|-----------|--|
| 0 ~ 3, 9  |   |
| 4 ~ 7, 15 |  |
| 8         |  |
| 10        |  |
| 11        |  |
| 12        |  |
| 13        |  |
| 14        |  |

注意：タイマレジスタ(レジスタ番号&H24～&H27)はサウンド制御命令が内部で使用します。書き込みを行った場合は動作の保証はありません。

■ FM/SSG音源内部レジスタマップ

| レジスタ<br>番号*          | レジスタのビット構成      |                 |                |                |                |                 |                 |                     | 対応<br>チャンネル                         | 説明                   | 備考             |
|----------------------|-----------------|-----------------|----------------|----------------|----------------|-----------------|-----------------|---------------------|-------------------------------------|----------------------|----------------|
|                      | D <sub>7</sub>  | D <sub>6</sub>  | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub>  | D <sub>1</sub>  | D <sub>0</sub>      |                                     |                      |                |
| &H21                 | TEST            |                 |                |                |                |                 |                 |                     | ×                                   | テスト用レジスタ             | ユーザー<br>アクセス禁止 |
| &H24                 | TIMER-A(I)      |                 |                |                |                |                 |                 |                     | ×                                   | タイマAレジスタ(上位)         | ユーザー<br>アクセス禁止 |
| &H25                 | —————           |                 |                |                |                |                 |                 | TIMER<br>-A(II)     | ×                                   | タイマAレジスタ(下位)         | ユーザー<br>アクセス禁止 |
| &H26                 | TIMER-B         |                 |                |                |                |                 |                 |                     | ×                                   | タイマBレジスタ             | ユーザー<br>アクセス禁止 |
| &H27                 | MODE            | RESET<br>B A    | ENABL<br>B A   | LOAD<br>B A    |                |                 |                 |                     | ×                                   | (CH3) CH3モードおよびタイマ制御 | ユーザー<br>アクセス禁止 |
| &H28                 | OPERATOR        |                 |                |                |                |                 |                 |                     | CH                                  | 1~3                  | 発音/非発音         |
|                      | 4               | 3               | 2              | 1              |                |                 |                 |                     |                                     |                      |                |
| &H30<br>&H31<br>&H32 | —               | DT <sub>1</sub> |                |                |                |                 | ML <sub>1</sub> | 1<br>2<br>3         | デチューン/マルチプライ<br>(オペレータ1)            |                      |                |
| &H34<br>&H35<br>&H36 | —               | DT <sub>3</sub> |                |                |                |                 | ML <sub>3</sub> | 1<br>2<br>3         | デチューン/マルチプライ<br>(オペレータ3)            |                      |                |
| &H38<br>&H39<br>&H3A | —               | DT <sub>2</sub> |                |                |                |                 | ML <sub>2</sub> | 1<br>2<br>3         | デチューン/マルチプライ<br>(オペレータ2)            |                      |                |
| &H3C<br>&H3D<br>&H3E | —               | DT <sub>4</sub> |                |                |                |                 | ML <sub>4</sub> | 1<br>2<br>3         | デチューン/マルチプライ<br>(オペレータ4)            |                      |                |
| &H40<br>&H41<br>&H42 | —               |                 |                |                |                | TL <sub>1</sub> | 1<br>2<br>3     | 最大出力レベル<br>(オペレータ1) | **                                  |                      |                |
| &H44<br>&H45<br>&H46 | —               |                 |                |                |                | TL <sub>3</sub> | 1<br>2<br>3     | 最大出力レベル<br>(オペレータ3) | **                                  |                      |                |
| &H48<br>&H49<br>&H4A | —               |                 |                |                |                | TL <sub>2</sub> | 1<br>2<br>3     | 最大出力レベル<br>(オペレータ2) | **                                  |                      |                |
| &H4C<br>&H4D<br>&H4E | —               |                 |                |                |                | TL <sub>4</sub> | 1<br>2<br>3     | 最大出力レベル<br>(オペレータ4) | **                                  |                      |                |
| &H50<br>&H51<br>&H52 | KS <sub>1</sub> | —               |                |                |                |                 | AR <sub>1</sub> | 1<br>2<br>3         | キーボードレートスケーリング深さ/<br>アタック係数(オペレータ1) | **                   |                |
| &H54<br>&H55<br>&H56 | KS <sub>3</sub> | —               |                |                |                |                 | AR <sub>3</sub> | 1<br>2<br>3         | キーボードレートスケーリング深さ/<br>アタック係数(オペレータ3) | **                   |                |
| &H58<br>&H59<br>&H5A | KS <sub>2</sub> | —               |                |                |                |                 | AR <sub>2</sub> | 1<br>2<br>3         | キーボードレートスケーリング深さ/<br>アタック係数(オペレータ2) | **                   |                |
| &H5C<br>&H5D<br>&H5E | KS <sub>4</sub> | —               |                |                |                |                 | AR <sub>4</sub> | 1<br>2<br>3         | キーボードレートスケーリング深さ/<br>アタック係数(オペレータ4) | **                   |                |

| レジスタ<br>番号(注1)       | レジスタのビット構成        |                |                 |                  |                |                |                |                | 対応<br>チャンネル | 説 明  | 備考                   |
|----------------------|-------------------|----------------|-----------------|------------------|----------------|----------------|----------------|----------------|-------------|--|----------------------|
|                      | D <sub>7</sub>    | D <sub>6</sub> | D <sub>5</sub>  | D <sub>4</sub>   | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |             |  |                      |
| &H60<br>&H61<br>&H62 | —                 |                | DR <sub>1</sub> |                  |                |                |                |                | 1<br>2<br>3 | ディケイ係数<br>(オペレータ1)   | **                   |
| &H64<br>&H65<br>&H66 | —                 |                | DR <sub>3</sub> |                  |                |                |                |                | 1<br>2<br>3 | ディケイ係数<br>(オペレータ3)   | **                   |
| &H68<br>&H69<br>&H6A | —                 |                | DR <sub>2</sub> |                  |                |                |                |                | 1<br>2<br>3 | ディケイ係数<br>(オペレータ2)   | **                   |
| &H6C<br>&H6D<br>&H6E | —                 |                | DR <sub>4</sub> |                  |                |                |                |                | 1<br>2<br>3 | ディケイ係数<br>(オペレータ4)   | **                   |
| &H70<br>&H71<br>&H72 | —                 |                | SR <sub>1</sub> |                  |                |                |                |                | 1<br>2<br>3 | サステイン係数<br>(オペレータ1)  | **                   |
| &H74<br>&H75<br>&H76 | —                 |                | SR <sub>3</sub> |                  |                |                |                |                | 1<br>2<br>3 | サステイン係数<br>(オペレータ3)  | **                   |
| &H78<br>&H79<br>&H7A | —                 |                | SR <sub>2</sub> |                  |                |                |                |                | 1<br>2<br>3 | サステイン係数<br>(オペレータ2)  | **                   |
| &H7C<br>&H7D<br>&H7E | —                 |                | SR <sub>4</sub> |                  |                |                |                |                | 1<br>2<br>3 | サステイン係数<br>(オペレータ4)  | **                   |
| &H80<br>&H81<br>&H82 | SL <sub>1</sub>   |                | RR <sub>1</sub> |                  |                |                |                |                | 1<br>2<br>3 | サステインレベル/リリース係数<br>(オペレータ1)  | **                   |
| &H84<br>&H85<br>&H86 | SL <sub>3</sub>   |                | RR <sub>3</sub> |                  |                |                |                |                | 1<br>2<br>3 | サステインレベル/リリース係数<br>(オペレータ3)  | **                   |
| &H88<br>&H89<br>&H8A | SL <sub>2</sub>   |                | RR <sub>2</sub> |                  |                |                |                |                | 1<br>2<br>3 | サステインレベル/リリース係数<br>(オペレータ2)  | **                   |
| &H8C<br>&H8D<br>&H8E | SL <sub>4</sub>   |                | RR <sub>4</sub> |                  |                |                |                |                | 1<br>2<br>3 | サステインレベル/リリース係数<br>(オペレータ4)  | **                   |
| &HA0<br>&HA1<br>&HA2 | F-number 1        |                |                 |                  |                |                |                |                | 1<br>2<br>3 | F-number 下位  | ***                  |
| &HA4<br>&HA5<br>&HA6 | —                 | Block          |                 | F-number2        |                |                |                |                | 1<br>2<br>3 | Block/F-number 上位  | ***                  |
| &HA8<br>&HA9<br>&HAA | CH3<br>F-number 1 |                |                 |                  |                |                |                |                | 3           | F-number 下位(オペレータ1)<br>F-number 下位(オペレータ2)<br>F-number 下位(オペレータ3)                | MODE<br>2,3の<br>ときのみ |
| &HAC<br>&HAD<br>&HAE | —                 | CH3<br>Block   |                 | CH3<br>F-number2 |                |                |                |                | 3           | Block/F-number上位(オペレータ1)<br>Block/F-number上位(オペレータ2)<br>Block/F-number上位(オペレータ3) | MODE<br>2,3の<br>ときのみ |
| &HB0<br>&HB1<br>&HB2 | —                 |                | FB              |                  | ALG            |                |                |                | 1<br>2<br>3 | フィードバック/アルゴリズム   |                      |

| レジスタ番号*              | レジスタのビット構成               |                |                |                |                |                |                |                | 対応チャンネル                | 説明          | 備考 |
|----------------------|--------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|------------------------|-------------|----|
|                      | D <sub>7</sub>           | D <sub>6</sub> | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |                        |             |    |
| &H00                 | TUNE11                   |                |                |                |                |                |                |                | 4                      | 周波数レジスタ下位   |    |
| &H01                 | —                        |                |                |                | TUNE12         |                |                |                | 4                      | 周波数レジスタ上位   |    |
| &H02                 | TUNE21                   |                |                |                |                |                |                |                | 5                      | 周波数レジスタ下位   |    |
| &H03                 | —                        |                |                |                | TUNE22         |                |                |                | 5                      | 周波数レジスタ上位   |    |
| &H04                 | TUNE31                   |                |                |                |                |                |                |                | 6                      | 周波数レジスタ下位   |    |
| &H05                 | —                        |                |                |                | TUNE32         |                |                |                | 6                      | 周波数レジスタ上位   |    |
| &H06                 | —                        |                |                |                | NOISE          |                |                |                | 4~6                    | ランダムノイズ周波数  |    |
| &H07                 | 1                        | 0              | NOISE MIX      |                | TONE MIX       |                |                | 4~6            | ノイズ/楽音ミキサーコントロール       | 上位2bit 変更禁止 |    |
| &H08<br>&H09<br>&H0A | —                        |                | F/V            |                | LEVEL          |                |                | 4<br>5<br>6    | 固定量/可変音量コントロール<br>出力音量 |             |    |
| &H0B                 | ENVELOP PER <sub>1</sub> |                |                |                |                |                |                |                | 4~6                    | エンベロープ周期下位  |    |
| &H0C                 | ENVELOP PER <sub>2</sub> |                |                |                |                |                |                |                | 4~6                    | エンベロープ周期上位  |    |
| &H0D                 | —                        |                |                |                | SHAPE          |                |                |                | 4~6                    | エンベロープ形状    |    |

\* レジスタ番号&H00~&H0DはSSG用, &H21~&HB2はFM用となっています。また, 内部レジスタのうち未定義のものやユーザーアクセスが禁止されているものをアクセスした場合, その動作は保証されません。

\*\* 内部レジスタの設定値のうち, TL, AR, DR, SR, SL, RRは, VOICE命令における音色パラメータの設定値とは大小/長短関係が逆になっています。

例) TLの場合

大 ↔ 小  
内部レジスタの設定値 0 ~127  
音色パラメータの設定値 127 ~ 0

\*\*\* MODE 2,3のときは, &HA2がCH3のオペレータ4のF-number 1(下位)となり, &HA6がCH3のオペレータ4のBlock/F-number 2(上位)となります。

### ■ ミュージックジェネレータボード(PC-9801-14)用プログラムの移植

サウンド制御命令はミュージックジェネレータボードで使用するミュージック制御命令と高い互換性があります。したがってミュージックジェネレータ用のプログラムをサウンド制御命令用に移植することも, PLAY命令のMML(ミュージックマクロランゲージ)単位では大きな変更なしに行うことができます。移植の際に注意すべき点は次のとおりです。

サウンド制御命令では6声までの出力ですが, ミュージック制御命令では8声までが使えます。移植の際は2チャンネル分を削ってください。

PLAY命令では次のような変更が必要です。

## ● S コマンド

## ミュージック制御命令

チャンネル：全チャンネルで使用可能

書式：S<アタック係数>，<ディケイ係数>，<サスティンレベル>，<リリース係数>，<サスティン係数>

## サウンド制御命令

チャンネル：チャンネル 4～6 でのみ有効

書式：S <エンベロープ形状>

チャンネル 1～3 に対するエンベロープは音色を切り換えるかユーザーが音色を作成するかしてください。なお、音色パラメータのエンベロープパラメータの数値とミュージック制御命令の S コマンドの数値には互換性はありません。

## ● O コマンド

## ミュージック制御命令

O4～O7 まで可能

## サウンド制御命令

O1～O8(O9) まで可能

## ● K コマンド

## ミュージック制御命令

K1～K49 まで可能。O4C に対応するものは K1

## サウンド制御命令

K0～K96 まで可能。O4C に対応するものは K36

## ● T コマンド

## ミュージック制御命令

各チャンネルごとにテンポを設定可能。

## サウンド制御命令

全チャンネルを通じて 1 つ。

PLAY 命令の文字列では、記述されている文字列の最後のテンポ指定が有効。

たとえば次の場合、

PLAY "T96CDE", "T120ABC", "T80FGA"

チャンネル 3 の文字列中にある "T80" が有効となる。

●その他

ミュージック制御命令は1小節を内部で128分割していますが、サウンド制御命令は1小節を192分割しますので、“L12”や“L24”などの音長も正確に表現できます。

なお、音長が256を超えた場合は、その値を256で割った余りが有効な値となります。長い音符を表現する際には注意が必要です。

例1) 2分音符の場合。

$$2 \text{ 分音符} = 192 / 2 = 96$$

96の長さ、つまり半小節分が発音される。

例2) 符点全音符の場合。

$$\text{符点全音符} = 192 / 2 + 192 = 288 \rightarrow 288 \div 256 = 1 \dots 32$$

実際には288を256で割った余りの長さ=32(2拍3連符の1音分)が発音されるため、符点全音符とはならない。

■サウンド制御命令を使ったプログラム例

●演奏プログラム

“バッハのインベンション”を次に示します。

```

1000 '*****
1010 '
1020 '          BACH
1030 '          Invention
1040 '
1050 '*****
1060 CLEAR ,&H9E00
1070 PLAY ALLOC 500,500,500
1080 PLAY "@46@V95MFZ0,3","@46@V95MFZ0,3","@46@V95MFZ0,3"
1090 '
1100 PLAY "T70"
1110 PLAY "L16R05DC+D04A805F8.EDE04A805G8",,"O4L8DEFDAB05C+O4A"
1120 PLAY "L16RFEDA4RG+F+G+G+8.A","L16R04AG+AE805C8.O4BABE805D8","L1605D4RCO4B
A05E404E4"
1130 PLAY "O5L16A4RDC+DG4RC+O4B05C+",,"O3A4","L16RCO4B05CF4R04BAB05E4"
1140 PLAY "O4RDC+D03A804F8.EDE03A804G8",,"O5F4.A8G4RFEF","L16R804A805DEDCO4B8.O
5C+C+4"
1150 PLAY "RFEDB-4REDC4","D4RB-AB-E4RAGA",,"RDC+D04G4R05C04B-O5C04F4"
1160 PLAY "RDCD03G804F8.EDEC8B-8",,"D4RDEFG4RFED",,"RB-A05C04B-4R05C04B-AB-AGF"
1170 PLAY "RAGAL8F05E-DC04B-A",,"C2C04B-AB-G05DCD",,"E8B-8.AGAF2"
1180 PLAY "B-G05C04CL16RFEFC8A8",,"O4GB-AB-805C04B-O5C04A4R805F8",,"R404E403L8FG
AF"
1190 PLAY "RGFGC8B-8.O5FEFC8A8",,"RE8.R8G8.O4A8.R805F8",,"O4CDECFGAF"
1200 PLAY "RGFGC8B-8.O5C04B-O5C04F805F8",,"RE8.R8G8.R4..",,"O5CDECF4L16REDC"
1210 PLAY "R04B-AB-E805E8.O4AGAD805D8",,"RDC+DG4RFEFG4",,"O4B-4RAB-G05C+4D04B-AB
-"
1220 PLAY "R8C8.O4BABG+8A8.G+F+G",,"REDE04A805D8.CO4B05C04F+8B8",,"L8G+AF+FE4D+D
"
1230 PLAY "L8AD+E03EL1604RAGAE805C8",,"RAG+A8GF+G03A4R4",,"L8C+CO3B4ABO4CO3A"
1240 PLAY "R04BABE805D8.O4ED+E03A804G8",,"R2O5RCO4B05CE-4",,"O4EF+G+EA4L16RGF+G"
1250 PLAY "RF+EFD805C8.GF+GD8B-8",,"RD8.R8F+8G2",,"D8C803B-8A8G04B-AB-O5CDDC"

```

```

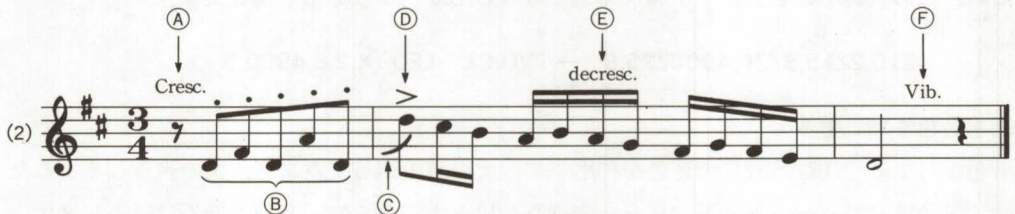
1260 PLAY "RAGAD805C8.DCD04G805F8","RF+B-AF4G","E-4RDE-CD04B-AB-05CO4B-AG"
1270 PLAY "REDEC8B-8.O4FEFC8A8","R2RAGB-AGFE","O5C8D8E8C8O4F4R8F8"
1280 PLAY "RGFG03B-8O4G8.FEFO3A8O4F8","D4RFEDC+4RO4AO5DC","L8B-AGB-AGFD"
1290 PLAY "FEDFE-DC+DC+8D8E4","O4B-2RAG+BAGFG","GFGAEO5C+O4A"
1300 PLAY "RDC+DO3A8O4F8.EDEO3A8O4G8","F4RAGAB-888O5C8C+8","RO3AO4D4L16GFGEB-A
G"
1310 PLAY "RF+EF+D8O5C8O4B-8A8B-AGA","D2.R8D8","AD05C8.O4B-AB-AGF+GD8B-8"
1320 PLAY "RAGAD8G8.FEFO3B8O4E8","L8C+CO4BB-A4G+G","GF+EF+GFE-DC+8D8.CO3B04C"
1330 PLAY "RDC+D8C+O3B04C+","F+FE4D1.,"D8O3G+8A4D1."
    
```

●演奏に表情をつける



譜面(1)は、表情を特につけずに演奏する例です。これを PLAY 命令の MML で示すと次のようになります。

PLAY "@2R8L16D8F+8D8A8D8>D8C+<BABAGF+GF+EL2DR4"

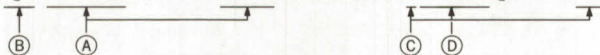


譜面(1)に表情をつけてみたのが譜面(2)です。これをサウンド命令を使って表現してみると、次のようになります。

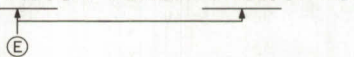
VOICE LFO 1,4,1,3500,-127,,5

← ③のための前設定

PLAY "@2R8 \*L16Q1 @V115D8F+8 @V120D8A8D8 !@V127Q2>D8 \*



Q6 @V120C+<BABA @V115GF+G



F+E Z10,2 Z15,8 Z20,4900 Z25,9 T80Q7 L2D."

↑ ⑥ビブラートをわかりやすくするために最後はテンポをおそくしています。

④クレッシェンド

徐々に音量が増す効果を付加します。実際には聴感上不自然でない程度に分割して音量を増加させます。

⑤スタカート

Q コマンドにより、音を短く区切ります。

⑥ピッチベンド

LFO 変調効果によりワンショット波形でピッチ変調効果を付加します。効果が不必要な部分では\*コマンドで効果を停止させておき、!コマンドで、必要なところには効果を付加します。

⑦アクセント

アクセントは音量を一時的に増すことで表現します。

⑧デクレッシェンド

徐々に音量が減る効果を付加します。実際には聴感上不自然でない程度に分割して音量を減少させます。

⑨ビブラート、トレモロなど

LFO 変調効果の繰り返し波形(三角波形)で、変調効果を付加します。

なお、この例の中の Z コマンドは次のような VOICE LFO 命令と同等です。

Z10,Z215,8Z20,4900Z25,9 → VOICE LFO X,2,8,4900,9

●基本音色を加工する

音色番号1の“BRASS2”の音色を利用して、犬の声を作ってみます。犬の声らしくするために、三角波ワンショット波形でピッチ変調をかけます。さらに、音程を低めに設定します。このように内蔵音色を加工して新しい音を創ることができます。

PLAY"@1": VOICE LFO 1,5,1,4500,127,,15

PLAY"O3L8Q4CR{CCC}4"

■外部オーディオ機器の使用

サウンド用外部オーディオ機器接続端子を使用することにより、外部オーディオ機器を接続することができます。

オーディオ機器への接続にはミニプラグを使用してください。ミニプラグを接続しますと、サウンド音は内部スピーカーから出力されなくなりますが、BEEP音は内部スピーカーから出力されます。

# 第10章

## ユーティリティプログラム

N<sub>88</sub>-日本語 BASIC(86)のシステムディスクには、BASIC を使う上で重要な作業を行うプログラムが用意されています。

### 10.1 ユーティリティプログラムを使う前に

#### ●ユーティリティプログラム一覧

次の表は、ユーティリティプログラムの名前と、その機能の概要を示したものです。

| プログラム名       | 節     | 機能                    | 内容   |
|--------------|-------|-----------------------|--|
| menu         | 10.2  | ユーティリティプログラムのメニュー     | ユーティリティプログラムをメニュー形式で選択、実行させる。  |
| format . nip | 10.3  | フロッピーディスクのフォーマット      | フロッピーディスクのフォーマットを行う。   |
| backup . n88 | 10.4  | フロッピーディスクのバックアップ      | 同じ容量のフロッピーディスク間でコピーを行う。すべてのファイルがコピーされる。                                |
| setinf . n88 | 10.5  | ID セクタの書き換え           | オートスタートのために、使用するファイル数、BASIC の命令を ID セクタへ書き込む。                          |
| xfiles . n88 | 10.6  | ファイル転送                | フロッピーディスクおよび固定ディスク間でのファイルのコピーを行う。                                      |
| sysgen . nip | 10.7  | システムのコピー              | システム(DISK CODE, IPL)のコピーを行う。   |
| setup . n88  | 10.8  | システムディスク属性の設定         | システムディスクの日本語入力環境、拡張画面ハードコピーの制御情報、および演算モードの設定を行う。                       |
| format . hd  | 10.9  | 固定ディスクのボリューム管理        | 固定ディスクの物理フォーマットと論理フォーマットを行う。   |
| recov . hd   | 10.10 | 固定ディスクの障害ボリューム/ファイル復旧 | 固定ディスクのエラー発生クラスタを正常な未使用クラスタで代替する。                                      |
| dir . hd     | 10.11 | 固定ディスクのファイルディレクトリ表示   | 固定ディスクに登録されているファイルのディレクトリを表示したり、ファイルを削除したりする。                          |
| backup . hd  | 10.12 | 固定ディスクファイル退避/復旧処理     | 固定ディスクとフロッピーディスク間でファイルの退避/復旧を行う。大きな1つのファイルを複数のフロッピーディスクに分割して退避することが可能。 |

| プログラム名       | 節     | 機能                 | 内容   |
|--------------|-------|--------------------|--|
| mkfont . n88 | 10.13 | 利用者定義文字格納ファイル作成／更新 | 利用者定義文字格納ファイル(usfontn88)の作成および更新を行う。                               |
| switch . n88 | 10.14 | メモリスイッチの設定         | メモリスイッチの設定を行う。   |
| dicmen . n88 | 10.15 | 辞書ファイルの保守          | 辞書ファイルへの単語登録・更新などを行う。  |
| DDconv . n88 | 10.16 | ファイル変換             | 5インチ 1D, 5インチ 2Dのフロッピーディスク上のファイルを, 1MB および 640KB のディスク上のファイルに変換する。 |
| tele . n88   | 9.2   | 電話管理               | 電話制御命令を使用して電話帳管理, オートダイヤル通話, ファイル転送などを行う。                          |

▶参照 tele . n88→9 . 1の「電話管理ユーティリティ “tele . n88” の使い方」

この章では次のことを前提にして, 各ユーティリティプログラムの解説を行います。

- 本体に内蔵されているディスクドライブのうち, 1の番号が付いているものをドライブ1, 2の番号が付いているものをドライブ2とします。
- 各種ユーティリティプログラムの入ったシステムディスクが本体内蔵のドライブ1から起動されたものとします。
- メニュープログラム(menu)を用いるか, またはRUN 命令を用いて, 各ユーティリティプログラムが起動している状態であるとしてします。

●フロッピーディスク装置が1台の機種の場合

フロッピーディスク装置が1台の機種でも format . nip, backup . n88, xfiles . n88, sygen . nip, DDconv . n88などのユーティリティプログラムを使ってフロッピーディスク間のデータ転送を行うことができます。

この場合, 各ユーティリティプログラムで送り側と受け側のドライブ番号を指定する際には同じドライブ番号を入力してください。すると,

送り側ディスクをドライブ番号×にセットして下さい  
受け側ディスクをドライブ番号×にセットして下さい

といった表示が繰り返されますので, 表示に従って送り側のディスクと受け側のディスクを交互に入れ換えてください。データ転送が行われます。

---


**注意:** フロッピーディスク装置が1台の機種でフロッピーディスク間のデータ転送を行う際にディスクの入れ換えを誤ると, 両方のディスクのデータを壊してしまう場合がありますので十分注意してください。

---

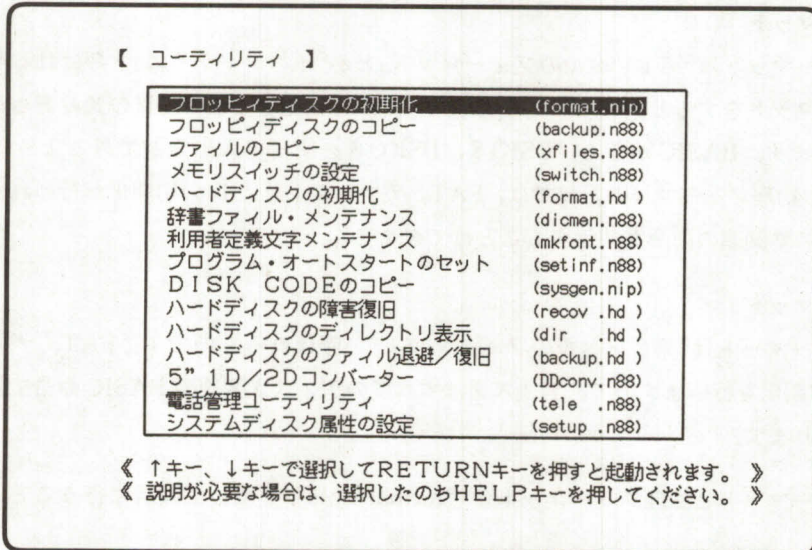
## 10.2 メニュープログラム




メニュープログラム(menu)は各ユーティリティプログラムを呼び出すプログラムです。このプログラムを使うと、各ユーティリティプログラムをメニュー形式で選んで実行させることができます。画面に表示されるメニューの中から行いたい作業を選択してください。メニュープログラムの実行手順は次のとおりです。

(1)メニュープログラムを実行する。


run " menu" 

(2)メニューが表示される。



これがメニュー画面です。カーソル移動キー(, )を使ってプログラムを選び(反転表示させる)、キーを押すとそのプログラムがロードされて起動されます。

なお、各ユーティリティプログラムは、メニュープログラムを使わないで直接実行することもできます。それには、

run "プログラムファイル名" 

とします。これでメニューを使った場合と同様にプログラムが起動します。

## 10.3 フロッピィディスクのフォーマット

ディスクを BASIC で使用可能な状態にすることを、初期化(フォーマット)といいます。新しいディスクを使うときには必ずフォーマットを行わなければなりません。すでにデータの入っているディスクのデータをすべて消し、改めて使用するときにもフォーマットを行います。

フロッピィディスクのフォーマットはユーティリティ "format . nip" を使って行います。なお、このユーティリティプログラムではシステム(DISK CODE および IPL)をコピーすることにより、ユーザー用のシステムディスクを作ることもできます。

### ■物理フォーマットと論理フォーマット

フォーマットには物理フォーマットと論理フォーマットの2つのレベルがあります。

#### 物理フォーマット

物理フォーマットは「第1レベルのフォーマット」とか「イニシャライズ」と呼ばれるものです。このフォーマットをすると、ディスクの物理的な位置を指定してディスクの読み書きができるようになります。BASIC から、DSKI \$、DSKO \$ を使って入出力ができるようになります。しかし、物理フォーマットだけでは、FAT、ディレクトリ、ID の初期化が行われませんので他のディスク関連の命令を使用することはできません。

#### 論理フォーマット

論理フォーマットは「第2レベルのフォーマット」と呼ばれるものです。FAT、ディレクトリ、ID の初期化を行いますので、ディスクはすべてのディスク関連の BASIC 命令で使用可能な状態になります。

物理フォーマットと論理フォーマットは "format . nip" を使って同時に行うことができます。

フォーマットは BASIC で新しいディスクを使うときには必ず実行してください。また、フォーマットは BASIC 以外のシステム(たとえば MS-DOS など)でフォーマットされたディスクを改めて使用するときにも必ず行ってください。

一度、BASIC 用に物理フォーマットを行ったディスクをフォーマットする場合には、物理フォーマットを繰り返す必要はありません。

▶参照 FAT、ディレクトリ、ID →「付録 C ディスクの内部構造」

## ■ “format . nip” の使い方

このユーティリティプログラムでフォーマットを行ったディスクは、データディスクとして使用することができますが、フォーマットに続けてシステムをコピーしシステムディスクとすることもできます。

画面に表示されるメッセージに従って作業を進めてください。

(1) “format . nip” を起動させると次のような画面が表示されます。

新しいディスクをドライブにセットして下さい

フォーマットするドライブは？

システムディスクをドライブ装置から外して、どちらかのドライブにフォーマットするフロッピーディスクをセットし、ドライブ番号を入力してください。すると、

物理フォーマットを行いますか(y/n) ?

と表示されますので、物理フォーマットが必要な場合はyを入力し、物理フォーマットの必要がない場合はnを入力してください。ここでnを選択した場合は(2)に進みます。

この後、ディスク装置が1MB/640KB 両用タイプの場合にはフロッピーディスクのタイプをきいてきます。

ディスクタイプ(1. 1MB 2. 640KB) ?

フロッピーディスクが1MBタイプの場合は1を入力し、640KBタイプの場合は2を入力してください。物理フォーマットが行われる間は、次のようなメッセージが表示されています。

ドライブXを物理フォーマット中です

Xは物理フォーマット中のドライブ番号です。

(2)物理フォーマットが終了すると、システムディスクを作成するかどうかを聞いてきます。

システムディスクを作成しますか(y/n) ?

データディスクとして使用する場合にはnを入力してください。システムディスクを作成する場合にはyを入力してください。yと答えたかnと答えたかにかかわらず、論理フォーマットが行われます。論理フォーマットが行われる間は、次のようなメッセージが表示されています。

処理中です

- (3)論理フォーマットが終了すると、システムディスクを作成しない場合には次のように表示されてプログラムを終了します。

終了しました

- (4)システムディスクを作成する場合には次のように表示されます。

システムディスクのドライブ番号は？

システムディスクを、現在フォーマットしているディスクのドライブとは別のドライブに入れ、そのドライブ番号を入力してください。

確認しましたか (y/n) ?


と表示されたら、ディスクが正しくドライブにセットされていることを確認し、まちがいでなければyを入力してください。システム(DISK CODEとIPL)のコピーを行います。ディスクのセットがまちがっている場合には、nを入力してやりなおしてください。システムのコピーが行われている間は次のように表示されます。

システムをコピーしています

- (5)次に、システムディスクを起動したときに各種のフロッピーディスク装置のデバイス番号をどのような順番で割り当てるか(ドライブアロケーションタイプ)を設定します。

システムディスクのドライブアロケーションタイプ：X

新しいディスクのドライブアロケーションタイプは？

と表示されるので、システムディスクと同じ場合には、だけを押しします。システムディスクと異なるドライブアロケーションタイプを新しいディスクにセットしたい場合は、そのタイプに応じて1または2を入力してください。

入力する数値とドライブアロケーションタイプの関係は次のとおりです。

1：640KB フロッピーディスク→1MB フロッピーディスク

2：1MB フロッピーディスク→640KB フロッピーディスク

たとえば、1MBのフロッピーディスクドライブ装置を内蔵し、640KBのフロッピーディスクドライブ装置を外部に取り付けている場合に1を選択すると、640KBのドライブがデバイス番号1、2となり、1MBのドライブがデバイス番号3、4となります。

---

注意：PC-9801, PC-9801E/F/M では、フロッピーディスクのドライブアロケーションタイプが次のようになります。

- 1 : 640KB → 1MB → 160/320KB
  - 2 : 640KB → 160/320KB → 1MB
  - 3 : 1MB → 160/320KB → 640KB
  - 4 : 1MB → 640KB → 160/320KB
  - 5 : 160/320KB → 640KB → 1MB
  - 6 : 160/320KB → 1MB → 640KB
- 

(6) ドライブアロケーションタイプの設定が終わると画面には次のように表示されてプログラムが終了します。

終了しました

●注意すべきこと

- ・システムディスクを作成する際には、システムディスクとフォーマットするディスクが同じ種類(同じ容量)のフロッピーディスクでなければなりません。
- ・フォーマットを行うと、そのディスクに書き込まれていたファイルはすべて消去されます。誤って必要なファイルを消さないように注意してください。
- ・属性の宣言で書き込み禁止を宣言していても、フォーマットを行うと、ファイルは失われます。
- ・フロッピーディスクへの書き込みを禁止するには、ライトプロテクトを行ってください。

- ▶参照 属性の宣言で書き込み禁止→6.6の「ファイル属性の設定」  
ライトプロテクト→『BASIC入門』

## 10.4 フロッピーディスクのバックアップ

ディスクに書き込まれたファイルを、誤って KILL 命令で削除してしまうことがあります。また、OPEN 命令や PRINT #などでファイルの読み書きを行う際に命令の使い方を誤って、ファイルを消してしまうこともあります。

また、DSKO \$ 命令でディスクの物理アドレスに対して直接書き込みを行った場合や、機械語プログラムが暴走した場合などにファイルが壊れてしまうこともあります。

そこで、このような不測の事態に備えて、重要な情報が記録されているディスクは定期的に複写をとっておく必要があります。

ディスクを複写することを“バックアップ”をとるといい、複写したディスクのことを“バックアップディスク”、元になるディスクを“マスターディスク”と呼びます。フロッピーディスクのバックアップはユーティリティ “backup.n88” を使って行います。

### ■ “backup.n88” の使い方

(1) “backup.n88” を起動させると次のように表示されます。

```
ディスクのバックアップを行います
送り側ディスクをドライブにセットして下さい
ドライブ番号は？
```

マスターディスクをどちらかのドライブに入れて、ドライブ番号を入力してください。すると、

```
受け側のディスクをドライブにセットして下さい
ドライブ番号は？
```

と表示されるので、バックアップ用のディスクをもう一方のドライブに入れて、ドライブ番号を入力してください。

(2) ディスクのセットが終わったら新しいディスク(受け側ディスク)の物理フォーマットを行います。

```
ディスクのバックアップを行います
物理フォーマットを行いますか (y/n) ?
```

と表示されたら、物理フォーマットが不要のディスクを使う場合は n を入力してください。その後(3)に進みます。物理フォーマットが必要な場合には y を入力してください。y を

入力した場合には、さらに次のように表示されます。

**ドライブXのディスクを物理フォーマットします  
確認しましたか(y/n)?**

表示されたドライブが指定したドライブであるか、そのドライブのディスクがフォーマットしたいディスクであるかを確認したらYを入力してください。直ちに物理フォーマットが行われます。nを入力した場合には(1)にもどり、ドライブを指定し直すことができます。

- (3)続いてバックアップ処理が始まります。マスターディスクが入っているドライブとバックアップ用のディスクが入っているドライブのアクセス表示用LEDが交互に点灯します。バックアップが行われている間は次のように表示されます。

**XXトラックコピー中です**

XXはコピー中のトラックの番号です。

なお、マスターディスクとバックアップ用のディスクの容量は同じでなければなりません。ディスクの容量が異なると、ここで次のようなメッセージが表示されます。

**バックアップできません**

この場合にはディスクを確認してください。

- (4)バックアップが終わると次のように表示されてプログラムが終了します。

**終了しました**

## 10.5 IDセクタの書き換え

IDセクタとは、システムディスク(フロッピーディスク、固定ディスク)上に設けられた特別の領域で、これを書き換えることにより、BASICで使用するファイルの同時オープン数や、オートスタートで実行するプログラムを設定することができます。IDセクタの書き換えはユーティリティ“setinf.n88”で行います。

### ■ “setinf.n88” の使い方

(1) “setinf.n88” を起動させると次のように表示されます。

オートスタート情報をセットします

システムディスクがセットされているドライブ番号は？

IDを書き換えたいシステムディスクをドライブにセットし、そのドライブ番号を入力します。

確認しましたか (y/n)？

ドライブ番号とディスクを確認し、正しければyを入力してください。nを入力するとはじめにもどります。

How many files (0-15) ?

BASIC上で同時にオープンしたいファイル数を入力します。

▶参照 同時にオープンするファイル数→6.1の「ファイルバッファとファイルの同時オープン数」

(2)ファイル数の指定を行うと、次のように聞いてきます。

テキストを入力してください

BASICの起動後、自動的に実行したいBASICの命令を入力します。たとえば、“test.n88”というプログラムをスタートさせたい場合には、

```
run " test.n88"
```

と入力してください。このほかにも、任意の命令を入力しておくことができます。テキストは253文字までしか入力できませんので、それを超える長さのテキストを入力すると、

テキストは 253 文字までです

と表示してスピーカを鳴らし、もう一度テキストを入力するように指示してきます。


(3)テキストの指定が終わると次のように ID セクタに書き込む内容が表示されます。

ファイルの同時オープン数は **XX** です

テキストは : **run " test.n88"**

システムディスクをドライブ番号 XX にセットして下さい

準備ができたならリターンキーを押して下さい

(1)で指定したドライブにディスクが入っていることを確認してから  キーを押してください。

終了しました

終了メッセージが表示されたら ID セクタの書き込みの完了です。このシステムディスクを使用して BASIC を起動すると自動的に、プログラム "test.n88" が実行されます。もちろん、このとき、ドライブ番号1となるディスクには "test.n88" が入っていなければなりません。

---

**注意：**同時オープンするファイル数の入力時に "1" を入力すると ID セクタには 255 (16 進で FFH) がセットされ、以降 BASIC をスタートするたびにファイル数を尋ねるようになります。この場合、自動実行したい BASIC テキストを入力しても無効となります。

---

## 10.6 ファイル転送

ユーティリティ“xfiles.n88”は、ディスク間でファイル単位に情報を転送することによってコピーを行います。したがってディスク全体の複写を行うだけでなく、特定のファイルだけを転送してファイルのバックアップを作ることができます。また、フロッピーディスクと固定ディスクなど異なる容量のディスクの間でもファイルのバックアップをとることができます。

---

注意：「10.4 フロッピーディスクのバックアップ」で説明した“backup.n88”は、同じ容量のディスク間で、ディスク単位にバックアップを作るものです。

---

### ■ “xfiles.n88” の使い方

(1) “xfiles.n88” を起動すると、次のように画面に表示されます。

ファイル転送を行います  
送り側のドライブ番号は？

転送するファイルの入っているディスクをドライブに入れ、ドライブ番号を入力してください。すると次のように聞いてきます。

受け側のドライブ番号は？

ファイルの転送先となるディスクを別のドライブに入れ、そのドライブ番号を入力してください。

バックアップ用のディスクとして新しいディスクを使う場合には、前もって物理フォーマットと論理フォーマットを行っておいてください。


(2) ドライブの指定が終わると転送するファイルの単位を聞いてきます。

ファイル単位ですか、ボリューム単位ですか(f/v)？

特定のファイルだけを転送する場合は f を入力します。すると(3)に進みます。ディスク中のすべてのファイルを転送したいときには v を入力します。v を選択した場合には(5)へ進みます。

(3)(2)で f を選択した場合には次のように表示されます。

転送するファイルを選択して下さい。矢印キーでカーサ移動後リターンキーを押して下さい。ROLL UP キーを押すと次の画面になります。

カーソルを移動して転送するファイルの名前の位置で  キーを押してください。

#### 転送中です

転送しているファイル名の文字が反転し、ファイルの転送が行われます。

(4) 転送が終わると続けて別のファイルを転送するかどうかを聞いてきます。

#### 別のファイルを転送しますか(y/n)?

別のファイルがあればYを入力します。すると(3)からの繰り返しになります。

nを入力すると次のように表示されプログラムが終了します。

#### 終了しました

(5)(2)でvを選択した場合には確認のメッセージが表示されます。

#### ボリューム単位でよろしいですか(y/n)?

ドライブにディスクが正しくセットされていることを確認したらYを入力してください。

次々にファイルが転送されます。コピーが行われている間は、コピー中のファイル名が反転表示されます。

#### 転送中です

(6) ボリューム内のすべてのファイルがコピーされると終了します。

#### 終了しました

**注意：**コピーの際に受け手側のディスクの容量が足りなくなった場合には“disk full”のエラーメッセージが表示され、ファイルのコピーが中断します。このときにはディスク容量の十分なディスクを用意してやりなおしてください。

また、固定ディスク内のファイルをフロッピーディスクにコピーする場合には、一枚のフロッピーディスクの容量を超えたファイルをコピーすることはできません。その場合には“backup.hd”を使用してください。

▶参照 固定ディスクのバックアップ→「10.12 固定ディスクファイル退避／復旧処理」

## 10.7 システム(DISK CODE および IPL)のコピー

システムディスクには、DISK モード BASIC を起動させるために不可欠なシステム(DISK CODE および IPL)が格納されています。

“sysgen.nip”は、システムをフロッピーディスクあるいは固定ディスクにコピーし、BASIC の起動が可能なディスクを作成するためのユーティリティです。

フロッピーディスクの場合は“format.nip”あるいは“backup.n88”を使ってもシステムディスクを作ることができます。

固定ディスクの場合は、必ずこの“sysgen.nip”を使用してください。

---

注意：DISK CODE は、ROM モード BASIC に DISK モード BASIC の機能を付加するために必要な BASIC システム本体のことです。また IPL は、DISK モード BASIC を起動するために必要なシステムプログラムのことです。

---

### ■ “sysgen.nip” の使い方

(1) “sysgen.nip” を起動すると次のように聞いてきます。

システムをコピーします  
送り側のドライブ番号は？



システムディスクをセットし、そのドライブ番号を入力してください。すると、

受け側のドライブ番号は？

と表示されますので、システムをコピーしたいフロッピーディスクがセットされているドライブのドライブ番号、または固定ディスクのドライブ番号を入力します。

ただし、このときフロッピーディスクや固定ディスクは前もってフォーマットされていなければなりません。また、フロッピーディスク間で処理を行う場合には、両方のフロッピーディスクは同じ容量のものでなければなりません。

準備ができたらリターンキーを押して下さい

このようなメッセージが表示されたら、ディスクが正しくセットされているかどうかを確認してください。まちがいがなければ、キーを押してください。まちがっている場合は キーを押してプログラムを止め、RUN 命令を実行して始めからやりなおしてください。システムのコピーが行われている間は次のようなメッセージが表示されます。

## コピー中です

(2)次に、システムディスクを起動したときに各種のフロッピーディスク装置のデバイス番号をどのような順番で割り当てるか(ドライブアロケーションタイプ)を設定します。

システムディスクのドライブアロケーションタイプ：X

新しいディスクのドライブアロケーションタイプは？

もとのシステムディスクと同じにする場合には  キーを押します。

もとのディスクとは異なるドライブアロケーションタイプにしたい場合には、そのタイプに応じて、1または2を入力してください。

1：640KB フロッピーディスク→1MB フロッピーディスク

2：1MB フロッピーディスク→640KB フロッピーディスク

▶参照 PC-9801, PC-9801E/F/M の場合のドライブアロケーションタイプ→10.3の「format .nip」の使い方」

たとえば、1MBのフロッピーディスクドライブ装置を内蔵し、640KBのフロッピーディスクドライブ装置を外部に取り付けている場合に1を選択すると、640KBのドライブがドライブ番号1, 2となり、1MBのドライブがドライブ番号3, 4となります。

(3)次のように表示されてプログラムが終了します。

終了しました

---

**注意**・標準フォーマットの固定ディスクから BASIC を起動させるためには、必ず、他の OS (MS-DOS など) のフォーマットの前には BASIC 上で標準フォーマットを行っておき、その後で、システムのコピーを行います。

システムは固定ディスクの先頭の領域にコピーされますので、BASIC より先に他の OS の標準フォーマットを行っている場合には、他の OS の領域を破壊してしまいます。

・受け側ディスクにファイルが存在するとシステムをコピーできない場合があります。

---

▶参照 標準フォーマット、拡張フォーマット→「10.9 固定ディスクボリューム管理」

## 10.8 システムディスク属性の設定

“setup.n88”は、システムディスクに対して次のような諸属性の設定を行うユーティリティプログラムです。

- 日本語変換方式と辞書ファイルのドライブ番号の設定
- 拡張画面ハードコピー(PC-PR601系)の機能設定
- 演算モード(基本/拡張)の設定

### ■ “setup.n88”を使った日本語入力の属性設定

N<sub>88</sub>-日本語 BASIC(86)では、次の6種類の日本語入力方式を使うことができます。

- AI 逐次変換方式
- AI 連文節変換方式
- 逐次変換方式
- 連文節変換方式
- 単文節変換方式
- JIS16 進コード変換方式

どれを使用するかは、システムディスクごとに決めることができます。

システムディスクは出荷時に AI 逐次変換方式の状態にセットされています。他の日本語変換方式に変更するにはユーティリティ“setup.n88”を使います。また、これによって何度でも設定し直すことができます。

日本語変換の際に参照される辞書ファイルは出荷時にはドライブ2に入れて使用するよう設定してありますが、これを別のドライブに入れて使うこともできます。この場合、辞書ファイルが入ったディスクをどのドライブに入れるかということをシステムディスクの属性として設定しなければなりません。この設定には“setup.n88”を使って行います。

(1) “setup.n88”を起動させると、次のように表示されます。

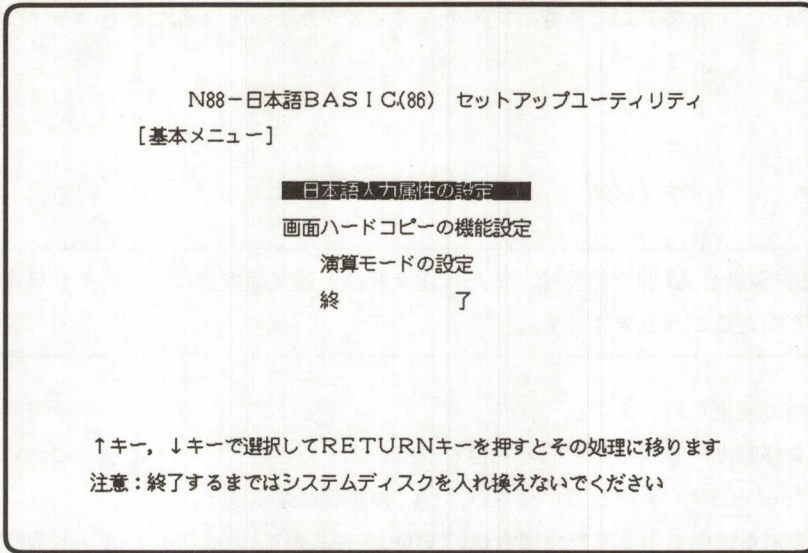
システムディスクがセットされているドライブは？


日本語入力の属性を変更したいシステムディスクを適当なドライブにセットし、そのドライブ番号を入力します。すると次のようなメニューが現れます。

---

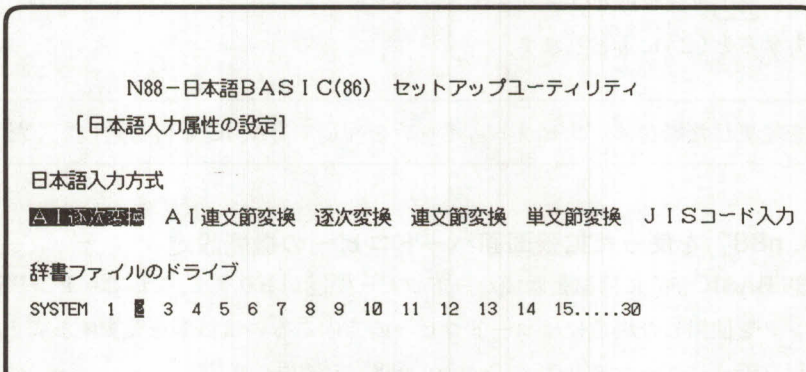
**注意：**以後、ユーティリティプログラムの実行が終了するまで、システムディスクは入れ換えないでください。

---



カーソル移動キーで“日本語入力属性の設定”を選択して、キーを押してください。

- (2)現在の日本語入力方式と、辞書ファイルのドライブの設定状態が次のようなメニューで表示されます。



反転表示されているところは、現在設定されている属性を表しています。

“日本語入力方式”は、そのディスクが現在どの変換方式に設定されているかを示しています。

“辞書ファイルのドライブ”は、日本語変換の際にBASICシステムが参照する辞書ファイルのドライブ番号を示します。日本語変換方式がJIS16進コード変換方式の場合にはこの選択はできません。

ドライブ番号は次のように示されます。

| SYSTEM | システム起動時にシステムディスクがセットされていたドライブ |
|--------|-------------------------------|
| 1      | ドライブ 1                        |
| 2      | ドライブ 2                        |
| }      | }                             |
| 30     | ドライブ 30                       |

参考：AI 逐次変換と AI 連文節変換，または逐次変換と連文節変換は，それぞれ日本語入力中に切り換えることもできます。

(3) 設定状態の変更を行います。

カーソル移動キーを用いて目的の項目を選択してください。すべての項目について選択が終わったら  キーを押してください。(4)の確認に進みます。

なお，設定を途中で中止したい場合は， キーまたは  +  キーを押すと，(1)の基本メニューにもどります。

(4) 日本語入力属性を変更していかどうか聞いてきます。

変更してよろしいですか(y/n) ?

y を入力すると変更が行われ，(1)の基本メニューにもどります。基本メニューで“終了”を選択し， キーを押せばプログラムは終了します。

n を入力すると(2)にもどります。

注意：属性を変更した場合は，リセットスイッチを押して BASIC を再起動してください。

### ■ “setup.n88” を使った拡張画面ハードコピーの機能設定

N<sub>88</sub>-日本語 BASIC(86)には拡張画面ハードコピー機能があります。なかでも PC-PR601 系のページプリンタを使用した場合にはハードコピーの方向あるいは倍率を変更することができます。これを行う際にはユーティリティ “setup.n88” を使用します。

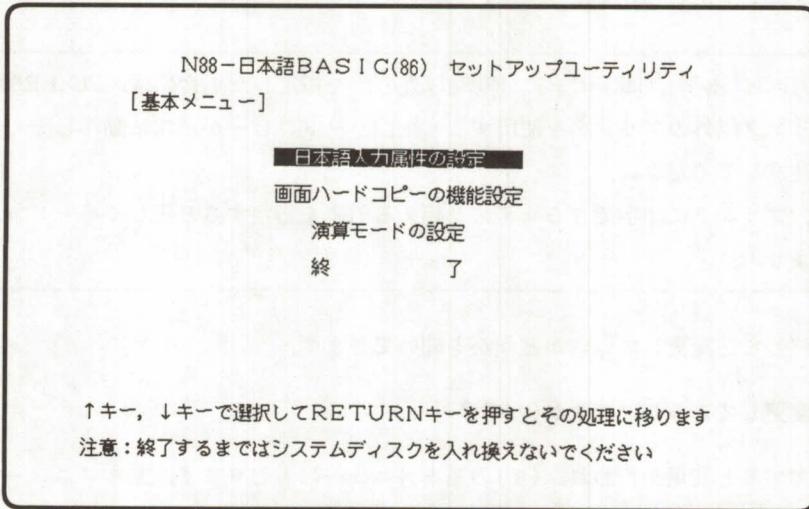
▶ 参照 ハードコピー→「3.3 画面ハードコピー」


(1) “setup.n88” を起動させると，次のように表示されます。

システムディスクがセットされているドライブは？

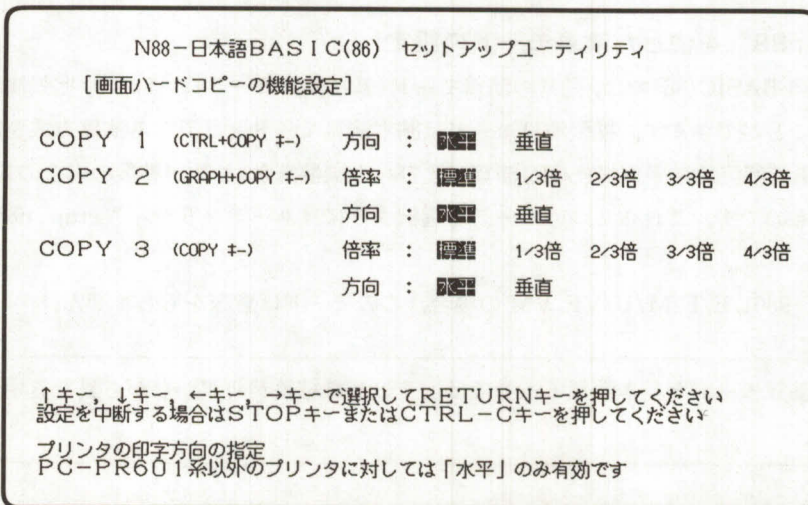
拡張画面ハードコピー機能の設定の変更を行いたいシステムディスクを適当なドライブにセットし，そのドライブ番号を入力します。すると次のようなメニューが現れます。

注意：以後、ユーティリティプログラムの実行が終了するまでシステムディスクは入れ換えな  
いでください。



カーソル移動キーで“画面ハードコピーの機能設定”を選択して、キーを押してください。

(2) 次のようなメニューが表示されます。



反転しているところは現在設定されている機能を表しています。カーソル移動キーを使って機能を選択してください。選択が完了した時点で $\leftarrow$ キーを押すと(3)の確認に進みます。

なお、設定を途中で中止したい場合は、 $\square$ STOPキーまたは $\square$ CTRL +  $\square$ Cキーを押すと、(1)の基本メニューにもどります。

---

**注意：**印字方向に「水平」、印字倍率に「標準」以外の値を指定した場合には、PC-PR601系ページプリンタ以外のプリンタを使用すると画面ハードコピーが正常に動作しなくなりますので注意してください。

また、プリンタに印字できるサイズを超える倍率を指定すると正しくハードコピーが行われません。

---

(3) 選択が終わると変更していいかどうかを聞いてきます。

変更してよろしいですか(y/n)?

yを入力すると変更が行われ、(1)の基本メニューにもどります。基本メニューで“終了”を選択し、 $\leftarrow$ キーを押せばプログラムは終了します。

nを入力すると(2)にもどります。

---

**注意：**機能の設定を変更した場合は、ユーティリティ“switch.n88”で拡張画面ハードコピーの使用を宣言し、ディップスイッチSW2の5番をONにした後、リセットスイッチを押してBASICを再起動してください。

---

## ■ “setup.n88”を使った演算モードの設定

N<sub>88</sub>-日本語 BASIC(86)では、通常の演算モード(基本演算モード)のほかに、拡張演算モードを使用することができます。拡張演算モードは基本演算モードとは異なる演算方式を用いることにより、単精度実数の平方根(SQR)演算およびべき乗演算(べき数が整数の場合のみ)の高速化を図ったものです。これら2つのモードを選択するにはユーティリティ“setup.n88”を使用します。

なお、PC-9801/E/F/M/U/VF/VMの場合、このモードは意味をもちません。

---

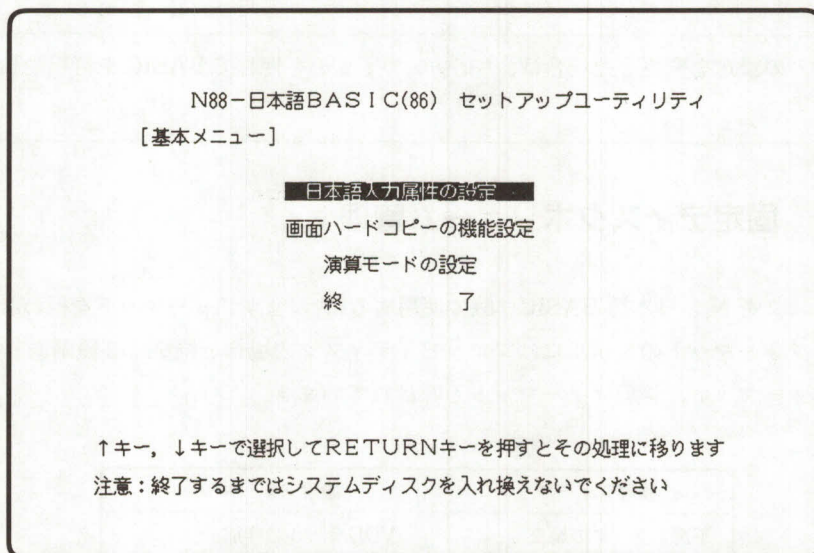
**注意：**拡張演算モードと基本演算モードでは、その演算結果がわずかながら異なる場合があります。


---

(1) "setup.n88" を起動させると、次のように表示されます。

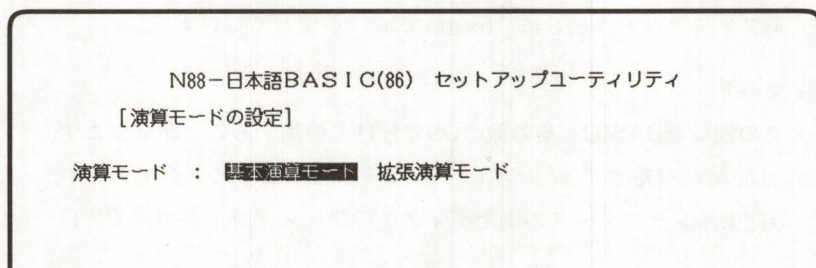
システムディスクがセットされているドライブは？


演算モードの設定の変更を行いたいシステムディスクを適当なドライブにセットし、そのドライブ番号を入力します。すると次のようなメニューが現れます。


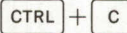


カーソル移動キーで“演算モードの設定”を選択して、キーを押してください。

(2) 次のようなメニューが表示されます。



反転しているほうが現在設定されているモードを表しています。カーソル移動キーを使って機能を選択し、キーを押すと(3)の確認に進みます。

なお、設定を途中で中止したい場合は、キーまたはキーを押すと、(1)の基本メニューにもどります。

(3) 選択が終わると変更していいかどうかを聞いてきます。

変更してよろしいですか (y/n) ?

y を入力すると変更が行われ、(1) の基本メニューにもどります。基本メニューで“終了”を選択し、 キーを押せばプログラムは終了します。

n を入力すると(2)にもどります。

**注意：**モードの設定を変更した場合は、リセットスイッチを押して BASIC を再起動してください。

## 10.9 固定ディスクボリューム管理

固定ディスクを N<sub>88</sub>-日本語 BASIC (86) で使用するには、まずフォーマットを行わなければなりません。フォーマットのレベルにはフロッピーディスクの場合と同様に2段階あり、それぞれは物理フォーマット、論理フォーマットと呼ばれています。

| 物理フォーマット    | 論理フォーマット      |
|-------------|---------------|
| 不良トラックの検査   | VOL ラベルの作成    |
| ディレクトリの作成   | ボリューム ID 部の設定 |
| データ部の初期化    | ID の作成        |
| トラックの読み書き検査 | FAT の作成       |

また、固定ディスクのフォーマット方式には標準フォーマットと拡張フォーマットの2種類があります。両フォーマットはともに“format .hd”を使って行います。

### ●標準フォーマット

- ・固定ディスクの領域をBASICとその他のOSで分けて使用することができます。
- ・BASICまたは、他のOSのうち1つのシステムを起動させることができます。
- ・20MBまでの従来インタフェースの固定ディスクのフォーマットが可能です。

### ●拡張フォーマット

- ・固定ディスクの領域をBASICとその他のOSで分けて使用することができます。
- ・複数のOSのシステムを登録しておき、任意のOSを起動することができます。
- ・20MBを超える固定ディスクのフォーマットができます。
- ・SCSI固定ディスクのフォーマットができ、SCSI固定ディスク1台を複数のドライブに分けて使用することができます。

---

**注意：**標準フォーマットの固定ディスクの領域を複数の OS で分けて使用するときには次のような点に注意してください。

- 複数の OS で使用する場合  
それぞれの OS (BASIC や MS-DOS など) で標準フォーマットを行い、使用する領域を確保します。
  - BASIC を起動させる場合  
標準フォーマットの固定ディスクから BASIC を起動させる場合は、他の OS の標準フォーマットの前に BASIC のための標準フォーマットを行ってからシステムをコピーするようにしてください。  
なお、他の OS の標準フォーマットを先に行った固定ディスクを BASIC で標準フォーマットし、BASIC のシステムをコピーした場合は、他の OS の領域のデータが破壊されてしまいます。
- 

## ■ “format .hd” の使い方

“format .hd” を起動させると次のように画面表示が行われます。

フォーマットする装置番号は？

フォーマットを行う固定ディスクの装置番号を指定してください。指定可能な装置番号が表示されますので、その中から選んで入力します。接続している固定ディスクが両フォーマット可能なときは、次のように表示され、標準フォーマットするか拡張フォーマットするかを聞いてきます。なお、標準フォーマットまたは拡張フォーマットのみが使用可能な固定ディスクの場合はこの表示はありません。

**拡張フォーマット**

標準フォーマット


以降、標準フォーマットと拡張フォーマットの場合で使い方が異なります。

---

**注意：**SCSI 固定ディスクの場合、装置番号は SCSI 固定ディスクに設定されている ID 番号順になります。

---

## ● 標準フォーマットを行う場合

(1) カーソル移動キーで標準フォーマットを選択して  キーを押してください。すると、現在のフォーマットの状態が表示されます。

ディスクタイプ = mMB n1 (n2) メガバイトが使用可能です

mは指定した固定ディスクのサイズを表します。

n1はBASICで指定可能な領域のサイズを表します。つまり、固定ディスクの全体のサイズから、BASIC以外のOSで使用している領域のサイズを引いた値です。

n2はn1のうちすでにBASICで使用可能となっている領域のサイズを意味します。

例1) 10MBの固定ディスクを初めてフォーマットする場合。

ディスクタイプ=10MB 10(0)メガバイトが使用可能です

例2) 20MBの固定ディスクをBASIC用に3MB、MS-DOS用に5MB分すでにフォーマットしている場合。この場合どのOSでも使用されていない未使用領域のサイズは12MBとなる。

ディスクタイプ=20MB 15(3)メガバイトが使用可能です

(2)続けてBASICのために確保するサイズをたずねてきます。

フォーマットするサイズは(メガバイト)?

BASIC用にフォーマットしたいサイズをメガバイト単位で入力します。

他のOSで使用している領域もBASICで使用したい場合には、BASICで指定可能な領域のサイズ(n1)より大きな値を指定します。この場合は、物理フォーマットが必ず必要です。また、他のOSで使用している領域をそのままにしておくには、フォーマットするサイズにはn1を超えない値を指定します。

(3)次に物理フォーマットを行うかどうかを聞いてきます。

物理フォーマットを行いますか(y/n)?

物理フォーマットを行う場合はyを入力してください。(4)へ進みます。

(2)でBASICで指定可能な領域のサイズ(n1)より大きな値を指定した場合にnを入力すると(1)にもどります。(2)でn1を超えないサイズを指定した場合にnを入力すると物理フォーマットを行わないで(5)へ進みます。

(4)物理フォーマットを選択した場合は、次のように表示されます。

ディスクの全領域をフォーマットします

確認しましたか(y/n)?

yを入力すると物理フォーマットが行われます。物理フォーマットが行われている間は次のように表示されます。Xは固定ディスクのドライブ番号を表します。

装置Xをフォーマット中です

固定ディスクのフォーマットには時間がかかります。横棒グラフで処理の進行状況が表示されます。

(5)次は論理フォーマットが行われます。論理フォーマットの間は次のように表示されます。

処理中です

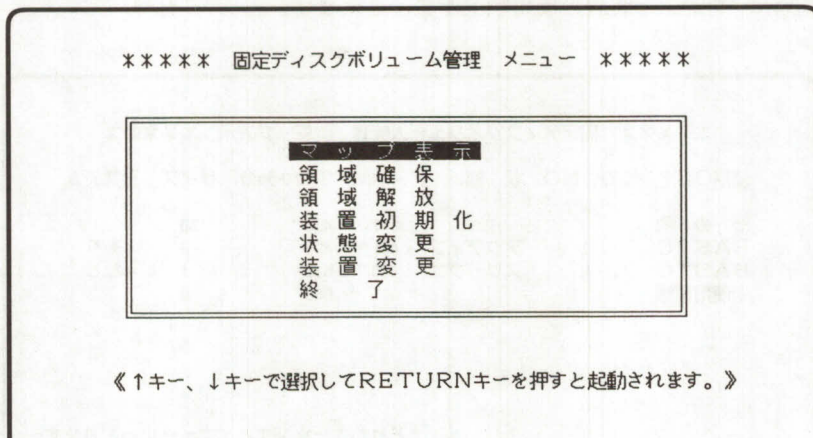
(6)処理が終了すると次のように表示されます。

終了しました

**注意：**固定ディスクの論理フォーマットを行うと、今まで BASIC で使用していた領域はすべて初期化されます。また、固定ディスクの物理フォーマットを行うと、今まで BASIC で使用していた領域だけでなく、他の OS で使用していた領域もすべて初期化されます。

#### ●拡張フォーマットを行う場合

最初に次のようなメニューが表示されます。



各々の項目を選択することにより、次のようなことを行うことができます。

- マップ表示 : 現在の固定ディスクの使用状況を示します。
- 領域確保 : 固定ディスクの中に BASIC の領域を確保します。
- 領域解放 : 現在 BASIC で使っている領域を解放します。
- 装置初期化 : 固定ディスクの全領域を物理フォーマットします。
- 状態変更 : 確保された領域を使用可能な状態(アクティブ)にするか、使用できない状態(スリープ)にしておくかを設定します。
- 装置変更 : 対象とする固定ディスクを変更します。

目的に応じて各々の機能を組み合わせて使用してください。  
いくつかの例を挙げます。


- 新しい固定ディスクをフォーマットし、BASIC で使用可能な状態にする場合

**装置初期化→領域確保→状態変更**

- BASIC で使用する領域を増減する場合(他の OS の領域はそのまま)

**マップ表示→領域解放→領域確保→状態変更**

この他にさまざまな場合が考えられます。各機能の解説をよく読んでから作業を行ってください。

最初のメニューで目的の項目をカーソル移動キーで選択し、 キーを押してください。

**注意：**領域確保、領域解放は他の OS の領域に影響はありません。

• **マップ表示**

“マップ表示” を選ぶと装置の使用状況が表示されます。

| ***** 固定ディスクボリューム管理    マップ表示 ***** |       |       |      |           |     |      |
|------------------------------------|-------|-------|------|-----------|-----|------|
| 使用OS名                              | ID_NO | 状態    | FROM | TO(トラック)  | サイズ | システム |
| その他                                |       |       |      | 1 ~ 450   | 30  |      |
| BASIC                              | 1     | アクティブ |      | 451 ~ 495 | 3   | あり   |
| BASIC                              | 2     | スリープ  |      | 496 ~ 510 | 1   | なし   |
| 未使用領域                              |       |       |      | 511 ~ 613 | 6   |      |

どれかキーを押すとメニューにもどります

この例では、40MB 固定ディスク内に BASIC 領域が2つ、他の OS の領域が1つ存在する状態を示しています。

各項目の意味を説明します。

**使用 OS 名**                   : 現在その領域を使用しているオペレーティングシステムが BASIC かその他の OS かを示します。

**ID\_NO**                       : BASIC で使用している領域の識別番号を示します。

- 状態 : BASIC で使用している領域の状態を示します。  
 “アクティブ” … ドライブとして認識されている状態(BASIC で使用可能な状態)。  
 “スリープ” … 領域としては確保されているが、ドライブとして認識されていない状態。
- FROM TO(トラック) : 領域が占有する範囲を示します。
- サイズ : 領域の大きさをメガバイト(MB)で示します。
- システム : BASIC のシステムの登録状態を示します。  
 “あり” … システムが登録されている状態。  
 “なし” … システムが登録されていない状態。  
 システムが登録されているというのは、この固定ディスクから起動することができるということを意味します。

マップ表示の状態で適当なキーを押すと始めのメニューにもどります。

#### ・領域確保

固定ディスクの未使用領域を BASIC の領域として確保するには、メニューで領域確保を選択します。画面には次のように表示されます。

| ***** 固定ディスクボリューム管理 領域確保 ***** |       |       |           |          |     |      |
|--------------------------------|-------|-------|-----------|----------|-----|------|
| 使用OS名                          | ID_NO | 状態    | FROM      | TO(トラック) | サイズ | システム |
| その他                            |       |       |           | 1 ~ 450  | 30  |      |
| BASIC                          | 1     | アクティブ | 451 ~ 495 |          | 3   | あり   |
| BASIC                          | 2     | スリープ  | 496 ~ 510 |          | 1   | なし   |
| 未使用領域                          |       |       | 511 ~ 613 |          | 6   |      |


---

何メガバイト確保しますか?  
 FROMトラック番号=  
 実行しますか(y/n)?

---

ここで、確保したいサイズをメガバイト単位で入力してください。このとき、未使用領域のサイズを超えた値を指定すると、エラーとなります。

すると、確保する領域の先頭トラック番号を聞いてきます。

FROM トラック番号=

トラック番号を入力します。キーのみを入力すると空き領域の先頭から領域を割り当てます。

実行しますか(y/n) ?

今までに入力したパラメータを確認して正しい場合はyを入力してください。領域の確保が行われます。もし、パラメータに誤りがある場合は、nを入力してください。領域の確保は行われず始めのメニューにもどります。

**注意**・“領域確保”で確保されただけの領域はドライブとして認識されません。領域をアクティブな状態に変更しなければなりません。「状態変更」の項を参照してください。

- ・BASICの領域として確保できるのは、1つの領域につき120MB以内です。120MB以上の固定ディスクを使用する場合は注意してください。

・領域解放

BASICで使用している領域を解放するには、メニューで“領域解放”を選択します。画面には次のように表示されます。

| ***** 固定ディスクボリューム管理 領域解放 ***** |       |       |      |           |     |      |
|--------------------------------|-------|-------|------|-----------|-----|------|
| 使用OS名                          | ID_NO | 状態    | FROM | TO(トラック)  | サイズ | システム |
| その他                            |       |       |      | 1 ~ 450   | 30  |      |
| BASIC                          | 1     | アクティブ |      | 451 ~ 495 | 3   | あり   |
| BASIC                          | 2     | スリープ  |      | 496 ~ 510 | 1   | なし   |
| 未使用領域                          |       |       |      | 511 ~ 613 | 6   |      |


---

解放するID\_NO=  
解放してもいいですか(y/n)?

---

解放する ID\_NO=

解放する領域のID\_NOを入力してください。次に確認の要求があります。

解放してもいいですか(y/n) ?

解放する領域の指定が正しいかどうかを確認して、正しい場合はyを入力してください。領域の解放が行われます。

nを入力すると、解放は行われずに始めのメニューにもどります。

### ・装置初期化

装置全体の物理フォーマットを行うには、“装置初期化”を選択します。画面には次のように表示されます。

```

***** 固定ディスクボリューム管理   装置初期化 *****
使用OS名 ID_NO 状態 FROM TO(トラック) サイズ システム
その他
BASIC 1 アクティブ 451 ~ 495 3 あり
BASIC 2 スリープ 496 ~ 510 1 なし
未使用領域 511 ~ 613 6

          ディスクの全領域をフォーマットします
          確認しましたか (y/n)?
  
```

ディスクの全領域をフォーマットします  
確認しましたか(y/n) ?

フォーマットしてよければyを入力します。nを入力するとフォーマットが行われなくて、始めのメニューにもどります。yを入力した場合は次のように表示され、物理フォーマットが行われます。

装置Xをフォーマット中です

従来インタフェースの固定ディスクでは、物理フォーマットを行っている間、横棒グラフで処理の進行状況が表示されます。

---

**注意：**装置初期化ではその他のOSで使用している領域も含めて、すべてが物理フォーマットされますのでよく確認してください。

---

・状態変更

指定された領域の状態(アクティブ/スリープ)を変更します。

領域確保をした直後の領域の状態は“スリープ”であり、このままではドライブとして使用できません。領域を“スリープ”から“アクティブ”に変更するには状態変更を選択します。

```

          **** 固定ディスクボリューム管理   状態変更 ****
使用OS名  ID_NO  状態   FROM TO(トラック)  サイズ システム
その他
BASIC    1      アクティブ   1 ~ 450          30
BASIC    2      スリープ   451 ~ 495         3   あり
未使用領域  511 ~ 613         1   なし
          6

=====
変更する ID_NOを入力してください ?
変更してもいいですか(y/n)?
=====
    
```

変更する ID\_NO を入力してください？

と表示されたら、状態を変更したい領域のID\_NOを入力してください。

変更してもいいですか (y/n) ?

yを入力すると指定された領域がそれまでとは逆の状態に変更されます。なお、アクティブにできるBASIC領域の個数は固定ディスクによって異なります。

従来インタフェースの固定ディスク……………最大1つ

SCSI固定ディスク ……………最大4つ


状態をアクティブにするには、制限数を越えないように他のBASIC領域をスリープにしてから行ってください。複数のアクティブなBASIC領域はそれぞれ別のドライブとして認識されます。

nを入力すると、状態の変更は行われず始めのメニューにもどります。

**● 拡張フォーマットの固定ディスクから起動させる場合の注意点**

拡張フォーマットの固定ディスクからシステムを起動すると、固定ディスク内のどの領域のOSを起動するかを指定するメニュー画面が表示されます。

メニュー画面の表示にしたがって、以下の操作を行ってください。

- (1)カーソル移動キーで処理、領域の選択をし、キーで実行します。
- (2)カーソルを“起動する”にして実行すると、領域で指定したOSが起動します。

なお、固定ディスクからオートスタートする場合、カーソルを“自動起動に設定する”にして起動した後、“switch.n88”を実行し、立ち上げ装置の選択画面で、“BOOT on hard disk1”、“BOOT on hard disk 2”または、“BOOT on SCSI hard disk”に設定してください。以後、固定ディスクから起動させると、メニュー表示は行われず、固定ディスクからオートスタートすることができます。

この設定を行った場合、フロッピーディスク装置などの他のディスク装置から起動することはできなくなります。

オートスタートを解除するには、“switch.n88”を使って、立ち上げ装置の選択画面で、起動装置を“フロッピーディスク→hard disk”に設定します。その後、フロッピーディスク装置にディスクを入れないで再起動すると、メニュー画面が表示されるようになります。メニュー画面で自動起動を解除してください。

---

**注意**・標準フォーマット↔拡張フォーマットの変更を行った場合は、必ずシステムを再起動してください。

- ・SCSIインタフェースの固定ディスクは4台接続できますが、オートスタートが可能なのは最初の1台だけです。
-

## 10.10 固定ディスク障害ボリューム・ファイル復旧

固定ディスクの読み書きを行ったとき、読み書きしようとした部分に障害があると、“Disk I/O error” というメッセージが表示されます。

障害の検出は1セクタ(256バイト)単位に行われますが、使用/未使用の管理単位は1クラスタ単位になっています。このため、障害があるのが1つのセクタだけであってもそのセクタを含むクラスタ全体の読み書きが不可能になります。正常なセクタが残っているはずですから、正常なセクタをできるだけ復旧することができれば、障害によるデータの喪失を最小限にすることができます。そのためのユーティリティとして固定ディスク障害ボリューム・ファイル復旧プログラム“`recov.hd`”が用意されています。

このプログラムを使うと、障害を起こしたセクタのあるクラスタの中から、正常なセクタのデータを未使用の正常なセクタだけからなるクラスタにコピーすることができます。

復旧する対象をボリューム(ドライブ)単位とするものを「ボリューム復旧」、ファイル単位とするものを「ファイル復旧」とよびます。

▶参照 セクタ、クラスタ→「付録C ディスクの内部構造」

### ■ “`recov.hd`” の使い方

(1) “`recov.hd`” を起動すると、次のように表示されます。

ファイル単位ですか、ボリューム単位ですか(f/v)?

ファイル単位で復旧を行う場合にはfを入力してください。(2)へ進みます。ボリューム(ドライブ)単位の復旧を行う場合はvを入力してください。(3)へ進みます。

(2)ファイル単位を選択すると、障害を起こして読み書きできなかったファイルを問い合わせてきます。

ファイル名は?

ドライブ番号、ファイル名を入力してください。復旧処理が始まり、終了すると、

処理を繰り返しますか(y/n)?

と表示されます。yを入力すると改めてファイル名を聞いてきます。nを入力すると終了します。

(3) ボリューム単位を選択すると、障害を起こして読み書きできなかったボリュームのドライブ番号を聞いてきます。ドライブ番号を入力してください。

なお、障害を起こしたセクタのあるクラスタ中の正常なセクタのデータを、未使用の正常なセクタだけからなるクラスタにコピーする際に、未使用の正常なクラスタがないと、次のように表示されます。

**復旧処理ができません**

この場合、復旧処理は不可能です。

## 10.11 固定ディスクファイルディレクトリ表示

固定ディスクに格納されているファイルを管理するためユーティリティとして“dir.hd”が用意されています。このプログラムは、次の2つの機能を持っています。

### ディレクトリ表示機能

ディレクトリとはディスクに格納されたファイルの一覧表のようなものです。ドライブ中のファイルをユーザー識別名ごとに整理してディレクトリを出力します。

### ファイル削除機能

ユーザー識別名とファイル名を指定してファイルを削除します。

### ■ “dir.hd” の使い方

(1) “dir.hd” を起動させると、次のメッセージが表示されます。

ファイルディレクトリ表示  
ドライブ番号は？

ファイルディレクトリを表示したい固定ディスクのドライブ番号を入力します。

(2) 次にディレクトリの表示を行うか、ファイルの削除を行うかを聞いてきます。

機能(1. 表示 2. 削除)？

表示を選択する場合は1を、削除を選択する場合は2を入力します。

(3) 続けてユーザー識別名の入力を行います。

ユーザー識別名(XXX/999/000) ?

次のようにユーザー識別名を入力してください。

XXX： 特定ユーザーのファイルを対象にするときは、そのユーザー識別名を入力します。

999： 共通ファイルを対象とするときは、999を入力します。

000： 全利用者のファイルを対象とするときは、000を入力します。

キーだけを押した場合は、"" (スペース3つ)のユーザー識別名になります。

▶ 参照 ユーザー識別名，共通ファイル→6.4の「固定ディスクに固有な機能」

(4)(2)で表示を選択した場合は画面に出力するか，プリンタに出力するかを聞いてきます。

出力装置は(1. CRT 2. PRINTER) ?

1または2を選択してください。

(5)(3)で指定したユーザー識別名のディレクトリが出力されます。出力が終わり、キーを押すと、次のメッセージが表示されます。

\*\*\* 出力終了 \*\*\*

次の処理をしますか(y/n) ?

yを入力すると(2)から繰り返します。nを入力すると終了します。

(6)(2)で削除を選択した場合は，削除するファイル名を聞いてきます。

削除したいファイル名は？

ファイル名を入力してください。処理が終了すると(5)へもどります。

## 10.12 固定ディスクファイル退避／復旧処理

固定ディスクは容量が大きいため、そのバックアップをとる際には、複数のフロッピーディスクに分けて退避(転送)しなければならないことがあります。

そのためのユーティリティが“backup.hd”です。このプログラムを使うとフロッピーディスクに退避させたファイルを、再び固定ディスクに復旧させることもできます。

なお、“backup.hd”ではフロッピーディスク間でファイルの転送を行うことはできません。

### ■ “backup.hd”の使い方

(1) “backup.hd”を実行すると次のメッセージが表示されます。

ファイルのバックアップを行います  
送り側のドライブ番号は？

固定ディスクのファイルをフロッピーディスクに転送したい場合(退避)、固定ディスクのドライブ番号を入力してください。

フロッピーディスクのファイルを固定ディスクに転送したい場合(復旧)、フロッピーディスクのドライブ番号を入力してください。

(2)次に受け側のドライブ番号を聞いてきます。

受け側のドライブ番号は？

受け側となるドライブのドライブ番号を入力してください。

(3)次に対象となるファイル名を聞いてきます。


ファイル名は？

ファイル名を入力してください。

(4)すると、フロッピーディスクをドライブ装置にセットするように要求してきます。

ドライブXXにnn枚目のフロッピーディスクをセットして下さい

XXにはフロッピーディスクをセットするドライブ番号が表示されます。nnは何枚目のフロッピーディスクであるかを示す数値です。最初は01になっています。


1枚目のフロッピーディスクをセットし、キーを押してください。  
次のメッセージが表示されファイルの転送が始まります。

ffffffff m->n

ffffffff は処理中のファイル名です。m は送り側のドライブ名、n は受け側のドライブ名です。

- (5) 固定ディスク内のファイル容量が1枚のフロッピーディスクの容量を超える場合は、再び次のメッセージが表示されます。

ドライブ XX に nn 枚目のフロッピーディスクをセットして下さい

フロッピーディスクを入れ換えて、キーを押してください。2枚目のフロッピーディスクと固定ディスク間でのファイル転送が始まります。

このように、指定したファイルが完全に転送されるまでフロッピーディスクを入れ換えながらファイル転送を行ってください。

- (6) ファイルの転送が終わると次のメッセージが表示されます。

別のファイルを転送しますか(y/n) ?

さらに処理すべきファイルがある場合には Y を入力してください。(3) から再び始まります。n を入力すると、

終了しました

と表示して実行を終わります。

- 
- 注意**・退避するフロッピーディスクはあらかじめフォーマットされていなければなりません。
- ・複数のフロッピーディスクに退避する場合は、フロッピーディスクの種類がすべて同じでなければなりません。
  - ・複数のフロッピーディスクに退避したファイルを固定ディスクに復旧する際は、退避したときの順番と同順にフロッピーディスクを入れ換えるようにしてください。
  - ・“backup.hd” で作成した複数のフロッピーディスクにまたがるファイルを BASIC の通常のディスク関連命令で扱うことはできません。
-

## 10.13 利用者定義文字格納ファイルの作成・更新

DISK モード BASIC を起動したときシステムディスク内に "usfontn88" というファイルがあると、自動的にそのファイルから利用者定義文字パターンが取り出され、本体内の専用 RAM に設定されます。その結果、利用者定義文字を使用することができるようになります。

この "usfontn88" という利用者定義文字の格納ファイルを作成したり、更新したりするには、"mkfont.n88" というユーティリティプログラムを使います。

---

**注意：**機種によっては、利用者定義文字の利用できないもの、登録可能な文字数の異なるものがあります。

---

### ■ "mkfont.n88" の使い方

(1) "mkfont.n88" を起動させると、タイトルが表示されたあとで、次のメッセージが表示されます。

ドライブ番号は(1-X) :

X は使用可能なドライブの最大値です。

利用者定義文字格納ファイルの作成・更新を行うシステムディスクをセットし、そのドライブ番号を入力してください。

(2) 続いて次のうちのいずれかのメニューが出力されます。

- ・システムディスク中に "usfontn88" がない場合、または、"usfontn88" に利用者定義文字が登録されていない場合

登録済み漢字コード : ありません

機能(0. 終了 1. 追加)

- ・システムディスク中の "usfontn88" に利用者定義文字が登録されており、さらに新たな登録が可能な場合

登録済み漢字コード :

XXXX XXXX...XXXX

機能(0. 終了 1. 追加 2. 更新 3. 削除 4. 表示)

- ・システムディスクのファイル`usfontn88`に利用者定義文字がすべて登録されている場合

登録済み漢字コード：

XXXX XXXX…XXXX

機能(0. 終了 1. 更新 2. 削除 3. 表示)

利用者文字が定義されている場合のXXXXは登録済みの漢字コード(16進4桁)です。

- (3)(2)のうちのいずれの場合も、“終了”、“追加”、“更新”、“削除”、“表示”の中から選択可能な動作モードを番号で選択し、キーを押してください。

- ・“終了”を選択した場合

システムディスクに`usfontn88`を作成して終了します。追加、更新、削除などを行った場合には最後に必ずこの終了を選択してください。新しく追加、更新、削除された内容がセーブされます。

終了する前にキーを押すと、

処理が中断されました。処理を中止しますか(y/n)

とメッセージが表示されます。yを入力すると、システムディスクの`usfontn88`はそのまま書き換えられないで終了します。nを入力すると、再び(1)からやり直せます。ただし、今までの追加、更新、削除などはすべて無効となります。

- ・“追加”を選択した場合

①次のメッセージが表示されます。

漢字コードは(7621-767E hex OR 7721-777E hex)：

利用者定義文字に対応する漢字コードは、16進形式で7621~767E および7721~777E までですから、追加する利用者定義文字の漢字コードを16進形式で入力してください。

たとえば`762A`の漢字コードで文字を定義したい場合には、

762A

と入力してください。

②次のように表示されます。

| 利用者定義文字メンテナンスユーティリティ |       | Version 1.0 |
|----------------------|-------|-------------|
| 利用者定義文字を作成します        |       |             |
| 漢字コード=762A           |       |             |
| (line)               | (hex) | (dot image) |
| 1                    | 0000  | .....       |
| 2                    | 0000  | .....       |
| 3                    | 0000  | .....       |
| 4                    | 0000  | .....       |
| 5                    | 0000  | .....       |
| 6                    | 0000  | .....       |
| 7                    | 0000  | .....       |
| 8                    | 0000  | .....       |
| 9                    | 0000  | .....       |
| 10                   | 0000  | .....       |
| 11                   | 0000  | .....       |
| 12                   | 0000  | .....       |
| 13                   | 0000  | .....       |
| 14                   | 0000  | .....       |
| 15                   | 0000  | .....       |
| 16                   | 0000  | .....       |

この画面が表示されると、16進形式の表示の左上の文字のところにカーソルが表示されます。文字パターンの作成には、16進形式の数値を変える方法と、文字パターンを直接変える方法の2種類があります。いずれを用いてもかまいません。

次のように操作して文字パターンを作成してください。なお、カーソルは、カーソル移動キーを使って移動させることができます。

- 16進形式の数値変更** : 変更する数値の位置にカーソルを移動させて、0~9, A~Fのキーを押してください。同時に文字パターンの部分も変化します。
- 文字パターンの直接変更** : スペースキーと **\*** キーを使います。パターンを変更したい位置にカーソルを移動させてスペースキーを押すとその位置は "・" (0) に、**\*** キーを押すとその位置は "\*" (1) になります。同時に16進形式の部分も変化します。
- 文字パターン作成の終了** : カーソルが一番下の行にあるときに **↵** キーを押すと文字パターンの作成を終了し次に進みます。また、**ESC** キーを押すとカーソルの位置とは関係なく、文字パターンの作成、変更が終了したのものとして次へ進みます。

③今作成した文字を表示し、よいかどうか確認を求めてきます。

文字パターン ■

よろしいですか(y/n)

と表示されます。■は日本語文字として実際に使用される場合の文字イメージです。文字が正しい場合にはYを入力してください。(2)のメニューにもどります。文字がまちがっているときにはnを押して、再びやり直してください。

・“更新”を選択した場合

①次のように表示されます。

更新モード(0. 1 文字 1. 全部)

登録されている利用者定義文字の特定の1文字を更新する場合には、0を入力してください。すると②に進みます。登録されている利用者定義文字のすべてを更新する場合は1を入力してください。⑤に進みます。

②追加の場合と同様に次のように表示されます。

漢字コードは(7621-767E hex OR 7721-777E hex) :

更新する利用者定義文字の漢字コードを16進形式で入力してください。  
たとえば“762A”の漢字コードの文字を更新したい場合には、

762A

と入力してください。

③文字パターンが表示されるので、“追加”の場合と同様にキーを使って文字パターンを更新してください。文字パターンの作成方法は“追加”を選択した場合を参考にしてください。

④文字パターンの更新が終わるとそれでよいかどうか確認を求めてきます。

文字パターン ■

よろしいですか(y/n)

と表示されます。■は日本語文字として実際に使用される場合の文字イメージです。文字が正しい場合にはYを入力してください。(2)のメニューにもどります。文字がまちがっているときにはnを押して、再びやり直してください。

⑤登録されている文字すべてを更新する場合には、登録されているすべての文字に対して、③と④を繰り返して更新を行うことができます。すべての文字の更新が終わると(2)のメニューにもどります。

・“削除”を選択した場合

①次のように表示されます。

**削除モード(0. 1文字 1. 全部)**

登録されている利用者定義文字の特定の1文字を削除する場合には、0を入力してください。すると②に進みます。登録されている利用者定義文字のすべてを削除する場合は1を入力してください。④に進みます。

②1文字を削除する場合には次のように表示されます。

**漢字コードは(7621-767E hex OR 7721-777E hex) :**

削除する利用者定義文字の漢字コードを16進形式で入力してください。たとえば“762A”の漢字コードの文字を削除したい場合には、

**762A**

と入力してください。

③次に、削除する文字の確認を要求してきます。

**利用者定義文字 762A を削除しますか(y/n) :**

yを入力すると、削除が行われ、nを入力すると削除されません。その後(2)のメニューにもどります。

④利用者定義文字をすべて削除する場合には、次のように表示されます。

**利用者定義文字全部を削除しますか(y/n) :**

yを入力すると、すべての利用者定義文字について削除が行われ、nを入力すると削除されません。その後始めのメニューにもどります。

・“表示”を選択した場合

①次のように表示されます。

**表示モード(0. 1文字 1. 全部)**

登録されている利用者定義文字の特定の1文字を表示する場合には、0を入力してください。すると②に進みます。登録されている利用者定義文字のすべてを表示する場合は1を入力してください。④に進みます。

②1文字を表示する場合には次のように表示されます。

**漢字コードは(7621-767E hex OR 7721-777E hex) :**

表示する利用者定義文字の漢字コードを16進形式で入力してください。  
たとえば“762A”の漢字コードの文字を表示したい場合には、

**762A**

と入力してください。

③文字パターンを表示します。

**文字パターン ■**

**リターンキーを押して下さい**

■は日本語文字として実際に使用される文字イメージです。ここで $\boxed{\text{↵}}$ キーを押すと(2)のメニューにもどります。

④すべての文字の表示を選択した場合、 $\boxed{\text{↵}}$ キーを押すと登録されている漢字コードの順で次々と表示します。

---

**注意：**このプログラムでは利用者が作成した文字パターンを表示するために、高分解能のグラフィック画面を使用します。このため、400ラインの高解像度ディスプレイが必要です。

---

## 10.14 メモリスイッチの設定

PC-9800 シリーズの本体には 8 バイト分の不揮発性メモリがあります。このメモリは本体の電源が OFF になってもその状態を保持しています。

メモリスイッチはユーザーのシステム的环境を決定するさまざまな情報を持っています。

このスイッチを変更することによって、システムの起動時の状態をあらかじめ設定することができます。メモリスイッチの設定を行うユーティリティとして、`switch.n88` が用意されています。

なお、メモリスイッチは、モニタモードの ssw コマンドにより直接変更することもできます。

▶参照 モニタモード→「11.2 モニタモード」

### ■ `switch.n88` の使い方

`switch.n88` を起動させると次のような初期メニューが表示されます。

```


+++++ switch set.MENU +++++
RS-232C (初期設定)
RS-232C (送受信コード)
メモリスイッチ
数値演算プロセッサ
CRT 初期画面 色指定 (TEXT)
拡張ボード
画面ハードコピー
立ち上げ装置
Basic mode
Network Basic
終了

↑↓ ←→ --- 他を選択します
← --- 決定します

```

このメニューに表示されているのが、設定の可能な項目です。

**注意：**機種により、表示される項目が異なります。ここに挙げた項目が表示されない場合、その機種にはその項目に該当する機能がありません。

カーソル移動キーを押してカーソルを移動させると、選択する項目が反転表示されますので、  
 選択する項目のところで  キーを押してください。

すると、各項目ごとに次のような選択画面が現れます。

RS-232C(初期設定)の選択画面

```

--- RS-232C (初期設定) ---
Xパラメータ .....
通信方式 ..... (無効) ..... 有効
データbit長 ..... (全二重) ..... 半二重
パリティ チェック ..... (7 bit) ..... 8 bit
ストップbit長 ..... (使わない) 奇数 ..... 偶数
ボーレート ..... (1 bit) 1.5 bit 2 bit
          75          150          300          600
          (1200)       2400       4800       9600
    
```

RS-232C(送受信コード)の選択画面

```

--- RS-232C (送受信コード) ---
日本語ソフト コード .....
          (KI=1B4Bh KO=1B48h)
          KI=1A70h KO=1A71h
CR受信処理 .....
          (CR 復帰+改行)
          CR・LF 復帰+改行 CR 復帰
リターンキー押下時の送信処理 .....
          (CR) ..... CR・LF
S パラメータ .....
          (無効) ..... 有効
DELコード受信時動作(ターミナルモード/入出力モード) .....
          (BS/DEL) ..... NUL/NUL
    
```

メモリサイズの選択画面

```

--- メモリサイズ ---
サイズ ..... byte .....
  128K      256K      384K      512K      (640K)
    
```

## 数値演算プロセッサの選択画面

| — 数値演算プロセッサ —                  |       |    |
|--------------------------------|-------|----|
| V30用数値演算プロセッサ                  | ..... | 使う |
| 8086/80286/386/386SX用数値演算プロセッサ | ..... | 使う |

(使わない) (使わない)

## CRT 初期画面 色指定(TEXT)の選択画面

| — CRT 初期画面 色指定 (TEXT) — |       |   |
|-------------------------|-------|---|
| 色                       | ..... | 緑 |

(青)

## 拡張ボードの選択画面

| — 拡張 ボード —              |       |    |
|-------------------------|-------|----|
| サウンドボード                 | ..... | 使う |
| RS-232C (第2回線/第3回線) ボード | ..... | 使う |
| GP-IBボード                | ..... | 使う |

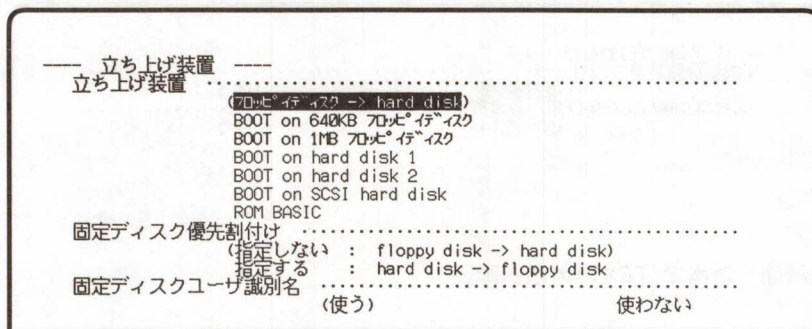
(使わない) (使わない) (使わない)

## 画面ハードコピーの選択画面

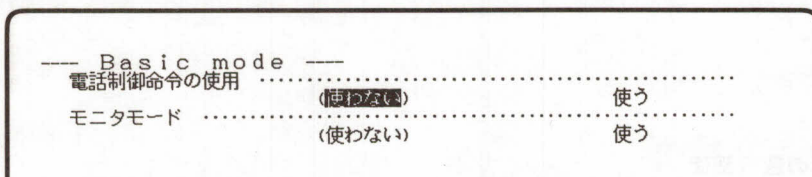
| — 画面ハードコピー —  |       |           |
|---------------|-------|-----------|
| PC-PR201系プリンタ | ..... | (使わない)    |
| 拡張画面ハードコピー機能  | ..... | 使う        |
| カラーコピー        | ..... | カラーコピーモード |

(使わない) (モノクロコピーモード)

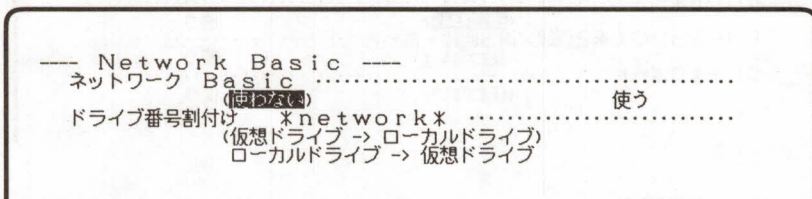
立ち上げ装置の選択画面




Basic mode の選択画面



Network Basic の選択画面



どの項目の設定も同様の手順で行うことができます。ここでは、例として“メモリサイズ”を選択した場合を説明します。

(1)初期メニューの画面で“メモリサイズ”を選択し、キーを押してください。

(2)メモリサイズを選択すると次のように表示されます。

```

.....メモリサイズ.....
サイズ・・・byte.....
128K   256K   384K   512K   ( 640K )
    
```

640K には ( ) が付いています。これは、現在のメモリの設定が、640Kbyte になっていることを示しています。ここでカーソル移動キーを押すと、反転表示されている項目が変わります。512K のところを選択してみます。

……メモリサイズ……

サイズ・・・byte.....

128K 256K 384K **512K** (640K)

(3)ここで $\leftarrow$ キーを押すと、

……メモリサイズ……

サイズ・・・byte.....

128K 256K 384K (**512K**) 640K

( )は512Kに移動します。これでメモリサイズが決定されます。

(4) $\leftarrow$ キーを押して最初のメニュー画面にもどり、“終了”を選択して $\leftarrow$ キーを押してください。プログラムが終了し、最後に次のメッセージが表示されます。

メモリスイッチを設定し終了しました

---

**注意：**メモリスイッチの内容を変更したときは、必ずリセットスイッチを押してBASICを再起動してください。再起動を行わないと、変更したメモリスイッチの内容は有効となりません。

また、再起動する前にディップスイッチSW2の5番がONになっていることを確認してください。OFFのまま再起動すると全メモリスイッチはシステムの規定値で初期化されてしまいます。

---

各項目の表示の意味とキー操作は、次のようにどの項目を設定する際も同様です。

- このユーティリティを実行中に $\leftarrow$ キーを押すことにより、作業を中止することができますが、その場合メモリスイッチの内容の変更は行われません。
- このユーティリティ実行中の表示状態および各キーの機能は次のようになっています。

|                 |                  |
|-----------------|------------------|
| ( )             | 現在設定されている値を示します。 |
| $\leftarrow$ キー | 初めの画面にもどります。     |
| HELPキー          | 説明を表示します。        |
| カーソル移動キー        | 項目を選択します。        |
| $\leftarrow$ キー | 決定を行います。         |

▶参照 メモリスイッチ→「付録 E メモリスイッチ」

## 10.15 辞書ファイルの保守

辞書ファイル保守のためのユーティリティとして“dicmen.n88”が用意されています。

“dicmen.n88”には次の3つの機能があります。

- 辞書ファイルへの単語の登録
- 辞書ファイルに登録されている利用者登録単語の削除
- 辞書ファイルに登録されている単語の一覧表出力

### ■ “dicmen.n88”の使い方

(1) “dicmen.n88”を起動させると次のように表示されます。

辞書ファイルのドライブ番号を入力して下さい。

辞書ファイルが格納されているディスクのドライブ番号を入力してください。

(2)次に、辞書ファイル名を聞いてきます。

辞書ファイルの名前を入力して下さい。


ディスクに格納されている辞書ファイルの名前を入力してください。

たとえば、“KNJAI.DIC”と入力します。

(3)次のようなメニューが表示されます。

↑、↓でカーソルを移動して機能を選択して下さい。

1. **単語の登録**
2. 単語の削除
3. 単語の一覧
4. 終了

選択する項目にカーソルを位置づけて  キーを押してください。

## ●単語の登録を選択した場合

①単語の登録を選択すると次のように表示されます。

## 〈単語の登録〉

カナ見出しを入力して下さい。

カナ見出し

登録したい単語のカナ見出し(単語の読み)を入力してください。日本語入力の方法はそのとき BASIC で設定している日本語入力環境によって異なりますので、適合した方法で入力するようにしてください。

カナ見出しは 16 文字以内の英数、カナ、記号(“\*”, “!” を除く)の組合せです。カナ見出しの入力に誤りがあるときには、スピーカが鳴り、再びカナ見出しの入力にもどります。

キーだけを入力すると(3)のメニュー画面にもどります。

ここでは例として“NEC”と入力したことにします。

②カナ見出しの入力が終わると次のように表示され、登録する単語の入力を求めてきます。

## 〈単語の登録〉

単語を入力して下さい。

カナ見出し NEC

単語

登録する単語を入力してください。単語は 16 文字以内の日本語で入力してください。単語入力に誤りがあるとスピーカが鳴り、再び単語入力にもどります。キーだけを入力すると①のカナ見出しの入力にもどります。

ここでは例として“日本電気株式会社”と入力したことにします。

③次に品詞の指定を求めてきます。

## 〈単語の登録〉

品詞の指定をしてください。

カナ見出し NEC

単語 日本電気株式会社

品詞の指定をしますか(Y/N)?

品詞の指定をしない場合 n と入力してください。⑥に進みます。

品詞の指定を行う場合 y と入力してください。

④品詞の一覧が表示されます。

〈単語の登録〉

品詞の指定をしてください。

カナ見出し NEC

単語 日本電気株式会社

品詞 **基本語** 動詞 固有名詞

カーソルを移動し、選択する品詞に位置づけ、キーを押してください。選んだ品詞に応じて次のように表示されますので、指示に従って操作します。

・基本語を選択した場合

品詞の詳細が次のように表示されます。この中から選択してください。

**名詞** サ変名詞 形容詞  
形容動詞 副詞 接続詞

サ変名詞とは“する”をつけることができる名詞のことです。解析する、感動するといった場合の“解析”、“感動”がサ変名詞です。

・動詞を選択した場合

品詞の詳細が次のように表示されます。動詞を活用形で分類しています。この中から選択してください。

**カ行5段** ガ行5段 サ行5段 タ行5段  
ナ行5段 バ行5段 マ行5段 ラ行5段  
アワ行5段 1段 サ行変格 カ行変格

各々の活用形は次のように活用します。

5段活用…口語で語尾が五十音図のあ段、い段、う段、え段、お段の5段に活用するものです。

| (例)  | 基本形 | 未然   | 連用 | 終止 | 連体 | 仮定(已然) | 命令 |
|------|-----|------|----|----|----|--------|----|
| カ行5段 | 行く  | か(こ) | き  | く  | く  | け      | け  |
| ガ行5段 | 泳ぐ  | が(ご) | ぎ  | ぐ  | ぐ  | げ      | げ  |
| サ行5段 | 押す  | さ(そ) | し  | す  | す  | せ      | せ  |
| タ行5段 | 打つ  | た(と) | ち  | つ  | つ  | て      | て  |
| ナ行5段 | 死ぬ  | な(の) | に  | ぬ  | ぬ  | ね      | ね  |
| バ行5段 | 飛ぶ  | ば(ぼ) | び  | ぶ  | ぶ  | べ      | べ  |
| マ行5段 | 飲む  | ま(も) | み  | む  | む  | め      | め  |
| ラ行5段 | 乗る  | ら(ろ) | り  | る  | る  | れ      | れ  |
| アワ5段 | 思う  | わ(お) | い  | う  | う  | え      | え  |

カ変活用…口語の“来(く)る”と文語の“来(く)”の各一語です。

サ変活用…語尾が、口語では“～する”と文語ではの“～す”となるものです。

| (例) | 基本形 | 未然 | 連用 | 終止 | 連体 | 仮定(已然) | 命令     |
|-----|-----|----|----|----|----|--------|--------|
| 文語  | す   | せ  | し  | す  | する | すれ     | せよ     |
| 口語  | する  | し  | し  | する | する | すれ     | せよ(しろ) |

一段活用…語尾が五十音図のい段(上一段)、またはえ段(下一段)の一段だけに活用するものです。


| (例)   | 基本形 | 未然 | 連用 | 終止 | 連体 | 仮定(已然) | 命令     |
|-------|-----|----|----|----|----|--------|--------|
| 上一段活用 | 着る  | き  | き  | きる | きる | きれ     | きろ(きよ) |
| 下一段活用 | 得る  | え  | え  | える | える | えれ     | えろ(えよ) |

活用の種類をまちがえて登録すると、日本語変換がうまくいかなくなるので正しく設定してください。

#### ・固有名詞を選択した場合

品詞の詳細が次のように表示されます。この中から選択してください。

|           |     |     |     |
|-----------|-----|-----|-----|
| <b>苗字</b> | 名前  | 地名  | 団体名 |
| 会社名       | 建物名 | 商品名 |     |

カーソルを移動し、選択するものに位置づけ、キーを押してください。

**注意：**固有名詞およびサ変名詞に接続する語句(接尾語)の関係。

固有名詞やサ変名詞には次に示すような語句(接尾語)をつなげて使うことがよくあります。

“dicmen.n88”を使って固有名詞やサ変名詞を登録すると、日本語変換の際に登録した単語の読みをタイプしたあとで続けてこれらの語句の読みをタイプすると、1つの文節として扱われるようになります。

固有名詞、サ変名詞に続けることができる語句(接尾語)の種類は限られており、次に示す表のとおりです。

| 固有名詞, サ変名詞の種類 | 語句(接尾語)の種類 |   |   |   |   |   |   |
|---------------|------------|---|---|---|---|---|---|
|               | A          | B | C | D | E | F | G |
| 名 詞           | ○          | ○ | ○ | ○ | ○ | × | ○ |
| 団 体 名         | ×          | × | × | ○ | ○ | × | ○ |
| 商 品 名         | ×          | × | × | × | ○ | × | × |
| 地 名           | ×          | × | ○ | × | ○ | × | × |
| サ変名詞          | ×          | × | × | × | × | × | × |
| 会 社 名         | ×          | × | × | ○ | ○ | × | ○ |
| 苗 字           | ×          | ○ | × | ○ | ○ | ○ | ○ |
| 名 前           | ×          | ○ | × | ○ | ○ | ○ | ○ |
| 建 物 名         | ×          | × | × | × | ○ | × | × |

A：論，率，力，料，流，名，法，別，文，物，部，病，日，品，表，費，内，等，的，片，中，値，高，代，説，性，上，所，時，室，書，者，策，後，権，組，業，金，局，級，型，額，後，会，化，員，案

B：用

C：風，港，駅

D：殿

E：式

F：氏，君

G：様

○は接続されるもの、×は接続されないものを表します。

たとえば、商品名に接続することができるのは式だけです。

“NEC”というカナ見出しに対し“日本電気”を商品名として登録しておきますと、日本語入力の際に“NECしき”という読みを入力すると“日本電気式”という変換が可能になります。

⑥最後に登録単語の確認を行います。

よろしいですか(Y/N) ?

yを入力すると単語の登録が行われます。nを入力するとこの単語の登録を取りやめて、①のカナ見出しの入力にもどります。

なお、カナ見出し、単語、品詞のすべてが同じ単語を登録することはできません。この場合、スピーカが鳴るとともに次のメッセージが表示され、①のカナ見出しの入力にもどります。

すでに登録されています。

⑦単語の登録が完了すると、終了するかどうか聞いてきます。

終わりますか(Y/N) ?

yを入力すると(3)の機能選択のメニュー画面にもどります。

nを入力すると、①のカナ見出しの入力にもどります。

#### ●単語の削除を選択した場合

①単語の削除を選択すると次のように表示されます。

〈単語の削除〉

カナ見出しを入力して下さい

カナ見出し

カナ見出しを単語の登録の場合と同じ方法で入力してください。

もし入力したカナ見出しに対応する利用者登録単語が存在しない場合は、次のメッセージが表示されるとともにスピーカが鳴り、再びカナ見出しの入力にもどります。

指定された利用者登録単語は存在しません。

また、利用者登録単語以外の削除を行おうとすると、次のメッセージが表示されます。


削除できません。

②指定された利用者登録単語があった場合には、指定した単語が次のように表示されます。同じカナ見出しで複数の単語が登録してあった場合には、、キーを押すと他の単語が表示されます。

登録単語 XXXXX/YYYY・ZZ

XXXX は登録されている単語です。/YYYY・ZZ は品詞情報で、次のように表示されます。

|       |             |                 |
|-------|-------------|-----------------|
| /     | ……品詞指定無し    | /基・接続……基本語/接続詞  |
| /動・カ5 | ……動詞/カ行5段活用 | /基・副詞……基本語/副詞   |
| /動・ガ5 | ……動詞/ガ行5段活用 | /基・形動……基本語/形容動詞 |
| /動・サ5 | ……動詞/サ行5段活用 | /基・形容……基本語/形容詞  |
| /動・タ5 | ……動詞/タ行5段活用 | /基・サ変……基本語/サ変名詞 |
| /動・ナ5 | ……動詞/ナ行5段活用 | /基・名詞……基本語/名詞   |
| /動・バ5 | ……動詞/バ行5段活用 | /固・商品……固有名詞/商品名 |
| /動・マ5 | ……動詞/マ行5段活用 | /固・建物……固有名詞/建物名 |
| /動・ラ5 | ……動詞/ラ行5段活用 | /固・会社……固有名詞/会社名 |
| /動・ア5 | ……動詞/ア行5段活用 | /固・団体……固有名詞/団体名 |
| /動・1段 | ……動詞/1段活用   | /固・地名……固有名詞/地名  |
| /動・サ変 | ……動詞/サ行変格活用 | /固・名前……固有名詞/名前  |
| /動・カ変 | ……動詞/カ行変格活用 | /固・苗字……固有名詞/苗字  |

削除したい単語を選択して  キーを押してください。

③すると、次のように確認メッセージが表示されます。

よろしいですか(Y/N) ?

削除する場合はYを入力してください。④に進みます。nを入力すると削除は中止されカナ見出しの入力にもどります。

④単語の削除が完了すると次のメッセージが表示されます。

終わりますか(Y/N) ?

Yを入力すると(3)の機能選択の画面にもどります。nを入力するとカナ見出しの入力にもどります。

### ●単語の一覧を選択した場合

①登録されている単語はカナ見出しを指定して画面に表示させることができます。登録されている単語がたくさんある場合には、一覧として続けて表示されます。

#### <単語の一覧>

出力開始位置のカナ見出しを入力して下さい。

出力開始カナ見出し

 キーだけを入力すると辞書の先頭から出力されます。

また、出力する範囲を設定するには、出力させたい先頭のカナ見出しを入力してください。



## 10.16 ファイル変換

5 インチ 2HD(1MB)または、5 インチ 2DD(640KB)のフロッピーディスク用のディスク装置では、通常、5 インチ 1D(160KB)および5 インチ 2D(320KB)のフロッピーディスクを読み書きすることはできません。

そこで、ファイル変換のユーティリティとして、“DDconv.n88”が用意されています。このプログラムを使用すると、5 インチ 2DD の読み書きが可能なフロッピーディスク装置で5 インチ 1D および 2D のフロッピーディスク中にあるファイルを読み出し、1MB または 640KB のフロッピーディスクに転送することが可能になります。

### ■ “DDconv.n88” の使い方

(1) “DDconv.n88” を起動させると、次のメッセージが表示されます。

1D / 2D 媒体のファイルを 2HD / 2DD 媒体に転送します  
送り側のドライブ番号は？

送り側となる 5 インチ 1D および 2D のディスクをセットするドライブの番号を入力してください。

続いて、次のメッセージが表示されます。

ディスクタイプは(1. 1D 2. 2D)？

送り側のディスクが 1D の場合は 1 を 2D の場合は 2 を入力してください。

(2) 次のメッセージが表示されます。

受け側のドライブ番号は？

受け側となる 1MB(3.5 インチ 2HD, 5 インチ 2HD, 8 インチ 2D)または 640KB(3.5 インチ 2DD, 5 インチ 2DD)のディスクをセットするドライブの番号を入力してください。次のように確認を求めてきます。

確認しましたか(y/n)？


まちがいなければ y を入力してください。n を入力した場合は (1) にもどって再びドライブ番号の入力になります。

(3) ディスクのセットを行います。

送り側ディスクをドライブ X にセットして下さい

受け側ディスクをドライブ Y にセットして下さい

X, Y は(1), (2)で指定したドライブ番号です。(1), (2)で指定したドライブに各々のディスクを入れてください。

セットが終わったら  キーを入力してください。次のように表示され、データの転送が始まります。

ffffff ファイル転送中です X->Y

ffffff は処理中のファイル名です。このようにしてすべてのファイルの転送を行います。

(4) 転送が終了すると、次のメッセージが表示されプログラムが終了します。

終了しました

---

**注意：**受け側のディスクは必ずフォーマットし、書き込み可能な状態にしておいてください。  
受け側にあるファイルと同じ名前のファイルを送り側から転送すると、受け側のファイルは失われます。  
このプログラムでは DISK CODE の転送はできません。

---

## 10.17 ユーティリティプログラムの目的別分類

## ● ディスク管理のためのユーティリティ

| 目 的                         | ユーティリティ名                     |              |
|-----------------------------|------------------------------|--------------|
|                             | フロッピーディスク                    | 固定ディスク       |
| フォーマット                      | format . nip                 | format . hd  |
| 起動可能なディスクの作成                | format . nip<br>sysgen . nip | sysgen . nip |
| システムの ID セクタの書き換え           | setinf . n88                 | setinf . n88 |
| ファイルの転送                     | xfiles . n88                 | xfiles . n88 |
| バックアップの作成                   | backup . n88                 | backup . hd  |
| 5 インチ 1D/2D ディスクの<br>ファイル変換 | DDconv . n88                 | -----        |
| 障害の復旧(ディスクの保守)              | -----                        | recov . hd   |
| ディレクトリ表示,<br>ファイル削除         | -----                        | dir . hd     |

## ● 日本語処理

| 目 的                     | ユーティリティ名     |
|-------------------------|--------------|
| 利用者定義文字格納ファイル<br>の作成・更新 | mkfont . n88 |
| 辞書ファイルの保守               | dicmen . n88 |
| 日本語入力方式の設定              | setup . n88  |
| 辞書ファイルのドライブ番号<br>指定     | setup . n88  |

## ● その他

| 目 的                                       | ユーティリティ名      |
|---|---------------|
| 拡張画面ハードコピー(PC-<br>PR601系)における倍率と方向<br>の設定 | setup . n88   |
| 演算モードの設定                                  | setup . n88   |
| メモリスイッチの設定                                | switch . n88* |

\* "switch . n88" では、PC-9800 シリーズ本体に対して次のような様々な設定を行うことができます。

- ・ RS-232C の初期設定
- ・ RS-232C の送受信コード処理の選択
- ・ メモリサイズの選択
- ・ 数値演算プロセッサの使用/非使用
- ・ 各種拡張ボードの使用/非使用
- ・ モニタモードの使用/非使用
- ・ 拡張画面ハードコピー機能の使用/非使用および機能の選択
- ・ 電話制御機能の使用/非使用
- ・ その他(立ち上げ装置の選択、ディスプレイ画面の色の指定など)

# 第11章

## ターミナルモードとモニタモード

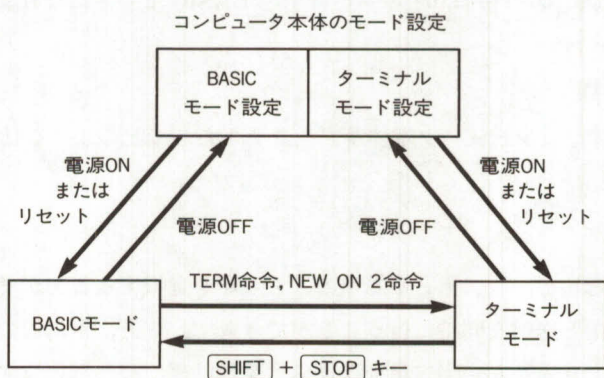
PC-9800 シリーズ本体には、BASIC モードの他にターミナルモードとモニタモードの2つのモードが用意されています。

ターミナルモードはPC-9800 シリーズを他のコンピュータの端末装置とするためのモードです。また、モニタモードは機械語モニタとして機能するモードです。

### 11.1 ターミナルモード

#### ■ターミナルモードと BASIC モードとの切り換え

ターミナルモードと BASIC のモードの切り換え関係は次のようになります。



#### ●ターミナルモードとしてシステムを起動させる方法

本体の電源を入れた直後にターミナルモードを起動させるには、次の2つの方法があります。

##### ディップスイッチによる方法

本体の電源を OFF にしてからディップスイッチ SW2 の 2 番を ON にし、再度本体の電源を ON にします。するとターミナルモードが起動します。以降電源を入れるたびに、またはリセットボタンを押すたびにターミナルモードが起動されます。

### システムディスクの ID セクタに BASIC の命令を書き込む方法

システムディスクの ID セクタに BASIC の命令を書き込んでおくと、本体に電源を入れて BASIC が起動した直後にその命令を自動的に実行します。

ターミナルモードにするには、ID セクタに TERM 命令あるいは NEW ON 2 命令を書き込んでおきます。

TERM 命令には通信の諸元を設定する多くのパラメータがあります。これについては「ターミナルの仕様の設定方法」に解説してあります。NEW ON 2 はコンピュータ本体のディップスイッチ SW2 の 2 番を一時的に ON と同様の状態にするものです。この設定は本体の電源を OFF にするかりセットボタンを押すともにもどります。

▶参照 TERM 命令→11.1 の「ターミナルの仕様の設定方法」

ID セクタへの BASIC 命令の書き込み→「10.5 ID セクタの書き換え」

### ●BASIC モードとターミナルモードの切り換え方法

#### BASIC モードからターミナルモードへ

BASIC モードで TERM 命令、あるいは NEW ON 2 命令を実行すると、ターミナルモードになります。

#### ターミナルモードから BASIC モードへ

ターミナルモードで、**SHIFT** + **STOP** キーを押すと BASIC モードになります。

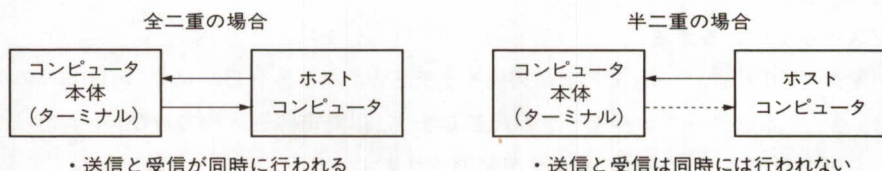
### ■ターミナルの仕様

ターミナルモードで、コンピュータ本体がターミナルとしてどのような仕様になるのかをおおまかに解説します。

### ●全二重と半二重

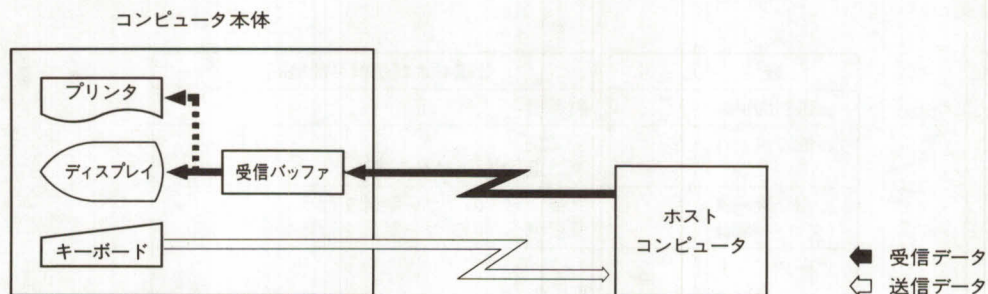
ターミナルの設定として、全二重と半二重のどちらかを選択することができます。

全二重の場合は受信と送信を同時に行うことができます。一方、半二重の場合は受信と送信を同時に行うことはできません。半二重では通常は受信状態になっています。キーボードからキーをタイプした後、**↵** キーを押すか、**CTRL** + **M** キーを押すと送信状態となりそれまでタイプしたデータが送信されます。



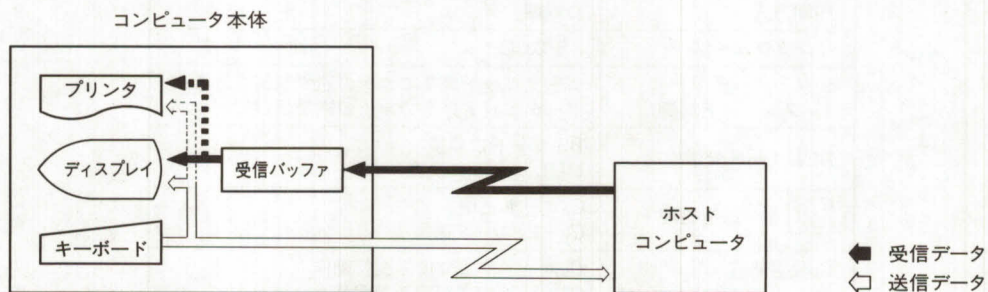
## ●ターミナルとしてのデータの流れ

## 全二重モードの設定の場合



- 1) タイプされたデータは、ただちに回線に送出されます。このときディスプレイ、プリンタへは出力されません。
- 2) 受信データはディスプレイに表示されます。受信データをプリンタに出力することもできます。
- 3) ホストコンピュータからのデータの受信と送信は、それぞれ独立して行われます。

## 半二重を設定した場合





- 1) タイプされたデータは、そのまま、ディスプレイに表示されます。このときプリンタへ出力させることもできます。
- 2) タイプされたデータは **[Enter]** キー、あるいは **[CTRL] + [M]** キーを押すことにより、ホストコンピュータへ送信されます。
- 3) 送信が終わるとホストコンピュータからの受信データをディスプレイに出力します。このときプリンタに出力することもできます。

全二重、半二重ともにプリンタに出力するには **[F·8]** キーを押します。

●ターミナルの諸元

次の表はターミナルモードでの諸元を示しています。固定的なものと選択できるものがあります。

| 諸元   | 仕様および選択可能範囲  |
|--|--|
| 伝送制御手順   | 無手順  |
| 通信方式   | 全二重<br>半二重(回線上是全二重)  |
| Xパラメータ<br>(フロー制御)  | 受信データのフロー制御を行う<br>受信データのフロー制御を行わない*  |
| データビット長  | 7ビット<br>8ビット   |
| ストップビット長   | 1ビット<br>1.5ビット<br>2ビット   |
| パリティチェック   | 偶数<br>奇数<br>なし   |
| ボーレート  | 75, 150, 300, 600, 1200, 2400, 4800, 9600  |
| 文字コード  | JIS7(ASCII), JIS8  |
| 日本語コード   | JIS C6226 コード  |
| 日本語シフトコード  | KI=ESC+K, KO=ESC+H<br>KI=SUB+p, KO=SUB+q   |
| 使用モデム  | 全二重  |
| 同期方式   | 非同期  |
| インタフェース  | RS-232C  |
| Sパラメータ<br>(シフトコード制御)   | データビット長7でカナの送受信ができる<br>データビット長7でカナの送受信ができない  |
| DELコード受信処理   | BSコードに変換<br>NULコードに変換  |
|  キー送信処理 | C <sub>R</sub> コード送信<br>C <sub>R</sub> +L <sub>F</sub> コード送信   |
| C <sub>R</sub> 受信処理  | C <sub>R</sub> コード単独で復帰改行動作<br>C <sub>R</sub> +L <sub>F</sub> 連続で復帰改行動作  |
| ブレイク信号処理   |  キーを押すとホストへ送信される。<br>ホストからのブレイク信号は受け付けない。 |

\*送信データに関しては無条件にフロー制御を行う。

日本語シフトコードを16進表記で表すと次のようになります。

ESC+K → 1B4B, ESC+H → 1B48, SUB+p → 1A70, SUB+q → 1A71

データビット長7ビットのモードでのカナの送受信は、シフトコードSO(&H0E)があると、それ以後のデータはカナと見なし、シフトコードSI(&H0F)がくるとそれ以後のデータを英数字と見なします。初期状態ではSIになっています。

ボーレートは、RS-232Cのハードウェア上の性能を示します。実際に通信を行う場合、実行するソフトウェアによっては、記述された速度による通信ができないことがあります。ソフトウェア的要因には次のようなものがあります。

- 一度に転送するデータの量
- 周辺装置に対する入出力の有無
- アプリケーションプログラム内での通信データの入出力間隔

これらの要因による、データの損失を防止するため、フロー制御という手法が用いられます。

#### ●フロー制御と装置制御コードD1/D3の処理

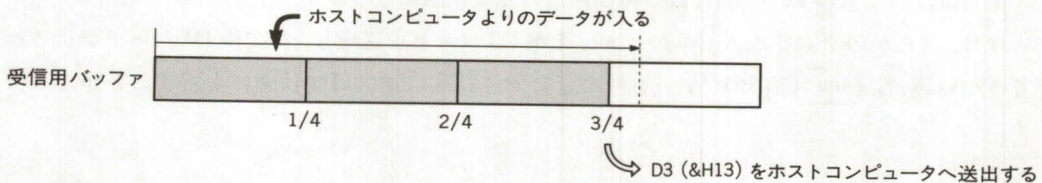
##### •ターミナル側の受信データのフロー制御

ターミナルでは、ホストコンピュータから送られてきたデータは受信用バッファと呼ばれるメモリにいったん蓄えられ、その後処理されます。このとき、ターミナルのデータ処理速度がホストコンピュータからのデータ転送速度より遅いと、受信用バッファが溢れてしまい、データが失われてしまうことになります。

このようなデータの損失を防ぐためにターミナルは次のような制御コードを送信します。

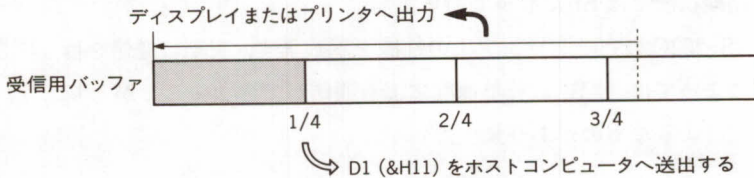
##### D3(&H13)コード

受信用バッファ中のデータがバッファ域の3/4を超えると、ターミナルは自動的にホストコンピュータへ装置制御コードD3を送信します。このコードはホストコンピュータからの送信を一時停止することを要求するものです。



### D1(&H11)コード

装置制御コード D3 を送信した後、受信バッファ中のデータ処理が進みデータ量がバッファ域の 1/4 以下になると、ターミナルは装置制御コード D1 を送信します。このコードはホストコンピュータの送信再開を要求するものです。



#### ・ターミナル側の送信データのフロー制御

ターミナル側で起こる受信バッファのオーバーフローはホストコンピュータ側でも同様に起こる可能性があります。


ターミナルからの送信を一時的に止めたり再開させたりするためには、ホストコンピュータはターミナルに対して装置制御コードを送ります。制御コードを受け取ったターミナルは送信を一時的に停止したり再開したりします。

### D3(&H13)コード

#### 全二重の場合



D3 コードを受信すると直ちにデータの送出を停止し、その後にタイプされたデータは無効になります。

#### 半二重の場合

D3 コードを受信したときにタイプしているデータは次に  キーが押されるところまでが送信されます。それ以降タイプされたデータは無効になります。

### D1(&H11)コード

全二重、半二重いずれの場合にも再び送信可能な状態になります。

なお、何らかの原因でホストコンピュータが D3 コードを送信し、その後 D1 コードが送られてこない場合、 +  キーを押すことで強制的に送信可能状態にもどすことができます。

### ●スクロール用メモリについて

ターミナルモードでは、受信したデータやキーボードからタイプしたデータがディスプレイに出力されますが、カーソルが最終行右端になると自動的にスクロールアップします。このとき、スクロールアップによって画面から見えなくなったデータはスクロール用メモリに蓄えられます。このデータは、ファンクションキー(F9)を押すことによってプリンタへ出力することができます。

### ●日本語コードの送受信について

#### ・日本語コードの入力

日本語コードを送信するためにキーボードから日本語を入力する方法は次のとおりです。

- ① CTRL + XFER キーを押します。すると、日本語コードを入力できるモードになります。
- ② 16進表記4桁のJISコードを入力します。
- ③ 日本語の入力を終わるときには CTRL + XFER キーを押します。

#### ▶参照 JISコード→『日本語入力ガイド』

日本語入力するときには、日本語文字列の前後に日本語シフトコード(KI/KO)が自動的に入力されます。回線では2バイトの日本語コードと1バイト系英数カナコードはこの日本語シフトコードの有無で区別されます。

|              |    |        |    |              |
|--------------|----|--------|----|--------------|
| 1バイト系英数カナ文字列 | KI | 日本語文字列 | KO | 1バイト系英数カナ文字列 |
|--------------|----|--------|----|--------------|

送/受信方向(順)→

#### ・日本語シフトコード KI/KO の選択

2種類のシフトコードがあります。ユーティリティ"switch.n88"またはTERM命令を使って選択することができます。

#### ▶参照 日本語シフトコード→5.1の「日本語文字列の内部形式」

#### ・日本語データ送受信の注意

##### 送信の場合

1) Sパラメータが有効のときには、日本語文字列はSI側(1バイト系英数文字側)の文字列の一部として送信されます。

例)

|    |    |    |    |    |   |    |    |    |    |    |   |    |    |   |    |    |   |                |                |
|----|----|----|----|----|---|----|----|----|----|----|---|----|----|---|----|----|---|----------------|----------------|
| SI | KI | 動物 | KO | SO | ト | SI | KI | 植物 | KO | SO | ノ | SI | KI | 差 | KO | SO | ハ | C <sub>R</sub> | L <sub>F</sub> |
|----|----|----|----|----|---|----|----|----|----|----|---|----|----|---|----|----|---|----------------|----------------|

- 2) 日本語文字列中に制御コード(C<sub>R</sub>など)を挿入するときは直前に KO が入っていなければなりません。日本語文字列の入力の途中で制御コードを入力するには、**CTRL** + **XFER** キーを押して英数カナの入力モードにしてから制御コードの入力をしなければなりません。このようにすると、制御コードの直前には自動的に KO コードが入ります。
- 3) 日本語 1 文字を表す第 1 バイトと第 2 バイトの間に制御コード(&H00~&H1F)が入っているはいけません。**CTRL** + **XFER** キーを使って日本語入力をした場合には第 1 バイトと第 2 バイトの間には制御コードは入りません。

### 受信の場合

- 1) 送られてくるデータには日本語コードの第 1 バイトと第 2 バイトの間に他のコードが挿入されていないものとして処理します。挿入されていると正常に文字が読めなくなる状態になります。
- 2) 送られてくるデータの日本語文字の第 2 バイトと次の日本語文字の第 1 バイトの間には制御コードが入っていてもかまいません。
- 3) 第 1 バイトが日本語コードでないときには、制御コードとして処理します。

## ■ターミナルの仕様の設定方法

ターミナルの仕様を細かく設定するには、ユーティリティ "switch.n88" による方法と TERM 命令を使う方法の 2 つがあります。

メモリスイッチによってターミナルの仕様設定を一度行くと、電源を切ってもその後ターミナルモードを起動するたびにいつもその設定で通信を始めることができるようになります。これを変更するには再びユーティリティ "switch.n88" を使って設定をやり直します。

なお、一時的にターミナルの仕様を変更したい場合には、BASIC モードで TERM 命令を使います。ただし、このときの設定が有効なのは、この TERM 命令の実行によって起動されたターミナルモードである間だけです。

最も使用頻度の高いターミナル仕様をユーティリティプログラムで設定し、異なった仕様のターミナルとして使いたい場合には TERM 命令で一時的に変更するとよいでしょう。

▶参照 switch.n88 →「10.14 メモリスイッチの設定」

### ●TERM 命令による方法

TERM 命令の書式は次のとおりです。

```
TERM "[COM:][<パリティチェック> [<データビット長> [<ストップビット長> [<Xパラメータ> [<Sパラメータ> [<DELコード受信処理> [<リターンキー送信処理> [<CRコード受信処理> [<日本語シフトコード>]]]]]]]]]" [, [<通信方式>][, <変数領域の大きさ>]]
```

〈変数領域の大きさ〉を除くすべてのパラメータは、省略するとメモリスイッチ(ユーティリティ"switch.n88"を使って設定する)で設定されている値になります。したがって、メモリスイッチの設定を変更する必要がないパラメータについては省略してかまいません。ただし、途中のパラメータを省略して後のパラメータを指定する場合は、必ず、省略記号としてブランク(空白)を入れてください。〈変数領域の大きさ〉の設定を省略すると 1024 バイトが確保されます。

また、パラメータには必ず大文字を使ってください。

各パラメータ指定値とその意味を示します。

| パラメータの種類   | 指定値 | 意味                                       |
|------------|-----|--|
| パリティチェック   | E   | 偶数パリティ                                   |
|            | O   | 奇数パリティ                                   |
|            | N   | パリティなし                                   |
| データビット長    | 7   | 7ビット                                     |
|            | 8   | 8ビット                                     |
| ストップビット長   | 1   | 1ビット                                     |
|            | 2   | 1.5ビット                                   |
|            | 3   | 2ビット                                     |
| Xパラメータ     | X   | 有効(受信データのフロー制御を行う)                       |
|            | N   | 無効(受信データのフロー制御を行わない)                     |
| Sパラメータ     | S   | 有効(データビット長7でカナを扱う)                       |
|            | N   | 無効(データビット長7でカナを扱わない)                     |
| DELコード受信処理 | B   | BSコードとして処理                               |
|            | N   | NULコードとして処理                              |
| リターンキー送信処理 | C   | C <sub>R</sub> コードを送信                    |
|            | L   | C <sub>R</sub> +L <sub>F</sub> コードを送信    |
| CRコード受信処理  | C   | C <sub>R</sub> コードで復帰と改行                 |
|            | L   | C <sub>R</sub> +L <sub>F</sub> コードで復帰と改行 |
| 日本語シフトコード  | P   | KI=ESC・K, KO=ESC・H                       |
|            | I   | KI=SUB・p, KO=SUB・q                       |
| 通信方式       | F   | 全二重方式                                    |
|            | H   | 半二重方式                                    |
| 変数領域の大きさ   | 数値  | リモート BASIC プロトコルを利用する際に用いられる変数領域の大きさを指定。 |

▶参照 TERM命令→『BASICリファレンスマニュアル』

### ●ボーレート(転送速度)と同期の設定

#### ボーレート(転送速度)

ボーレートの設定はホストコンピュータとの調整を行った後、ユーティリティ"switch.n88"を使って行ってください。TERM命令では設定することはできません。

## 同期クロックの設定

PC-9800 シリーズ本体でモデムを介して通信を行う際には普通 RS-232C のボーレートをカウントするためのクロックを PC-9800 シリーズ本体からとる (内部同期) か、モデムからとる (外部同期) を選択することができます。しかし、ターミナルモードでは内部同期しか使用できません。この選択にはディップスイッチ SW1 の 5 番と 6 番を使用しますが、ターミナルモードでは両方とも OFF にしておいてください。

## ■ターミナルの使い方

ターミナルモードで操作するために必要な事項を説明します。

### ●ファンクションキー

10 種類のファンクションキーを使用することができます。通常、ファンクションキーに登録されている機能 (ファンクションキーリスト) がディスプレイの最下行に表示されます。表示しないようにするには、ターミナルモードを起動させる前に BASIC 上で CONSOLE 命令を使いファンクション表示スイッチを OFF にしておいてください。

#### • f.1 ~ f.5 までの機能

ファンクションキーの内容はターミナルモードになる直前の状態が引き継がれます。電源 ON で直接ターミナルモードを起動させた場合は BASIC の初期状態と同じです。

#### • f.6 ~ f.10 までの機能

f.6 : 制御コードの扱い方を選択するために使用します。

制御コードとは文字以外の情報を持つ、&H00 ~ &H1F のコードです。次のどちらかを選択できます。

“N-LTRL” : 制御コードを制御コードとして処理します。

“LTRL” : 制御コードを文字として処理します。ただし、この場合は C<sub>R</sub>、L<sub>F</sub> だけは制御コードとして機能します。

f.7 : 通信方式を定義するために使用します。次のどちらかを選択できます。

“HALF” : 半二重通信方式

“FULL” : 全二重通信方式

f.8 : 画面に表示する内容をプリンタにも出力するかどうかを選択できます。

“LPT on” : プリンタにも出力します。

“LPT off” : プリンタには出力しません。

f.9 : 画面スクロール用メモリの内容をプリンタに出力することを要求します。

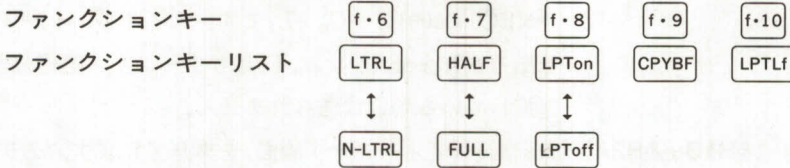
“CPY BF” : プリンタに出力します。f.8 が “LPT on” のときに有効です。

**f.10** : プリンタを1行改行することを要求します。

"LPT Lf" : プリンタを1行改行します。 **f.8**が"LPT on"のときに有効です。

### ●ファンクションキーリストの表示状態

ディスプレイの最下行に表示されるファンクションキーリストは次のようになっています。



**f.6**~**f.8**までは、選択できる機能を表示しています。**f.7**の場合を例にとると、全二重を選択している状態では **HALF** が、半二重が選択されている場合には **FULL** が表示されます。

また、**f.9**、**f.10**については機能名が常に表示されています。

### ●特殊キーの働き

**BS** (バックスペースキー)

**BS** キーのコード(&H08)を送信します。半二重モードでは同時にカーソルの左の1文字を削除します。

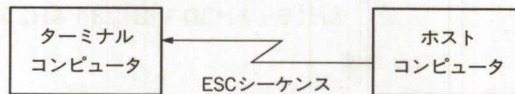
**STOP** (ストップキー)

ブレーク信号を送出します。

### ●スペシャル ESC シーケンス

ホストコンピュータが端末装置を制御する手段として"スペシャル ESC シーケンス"という特別な制御キャラクタが取り決められています。

これは ESC(エスケープ)コード(&H1B)で始まる一連の文字列です。ホストコンピュータは端末装置を制御するために ESC シーケンスを送信します。これを受信した端末装置は各々のエスケープシーケンスに定義されている機能を実行します。



定義されているスペシャル ESC シーケンスのコードとその機能、動作は次のとおりです。

- 1) ESC@ (&H1B+&H40) ディスプレイへの出力は、以後プリンタへも出力されます。
- 2) ESCA (&H1B+&H41) ディスプレイへの出力は、以後プリンタへは出力されません。1)の状態を解消するものです。

- 3) ESC" (&H1B+&H22) キーボードから入力したデータを有効にします。
- 4) ESC# (&H1B+&H23) キーボードから入力したデータを無効にします。3)の状態を解消するものです。以後 ESC" を受信するまでタイプされたデータは捨てられます。
- 5) ESC! (&H1B+&H21) ホストコンピュータへ ID コードを送信します。ID コードは "NEC00000010"+C<sub>R</sub>+L<sub>F</sub>です。ディスプレイやプリンタには何も出力されません。なお、L<sub>F</sub>はリターンキー送信処理で C<sub>R</sub>のみを選択している場合は送られません。
- 6) ESC\* (&H1B+&H2A) 表示画面(ディスプレイ画面)をクリアします。クリア後、カーソルは表示画面(ディスプレイ画面)の左上角にセットされます。
- 7) ESCY (&H1B+&H59) 現在のカーソル位置から、スクロールエリアの最下位行までをクリアします。カーソル位置は変化しません。
- 8) ESC△ (&H1B+&H20) 端末の状態をターミナルモードになった直後の状態にもどします。ただし、画面はクリアされません。"△"は半角スペースを表します。
- 9) ESC=ij (&H1B+&H3D) カーソルの位置を移動させます。移動する位置のキャラクタ座標を(列, 行)=(p, q)とすると、指定する i, j の値は次の式で求められます。なお、座標はテキスト画面のキャラクタ座標です。

$$i=p+\&H20$$

$$j=q+\&H20$$

&H20 だけ値をシフトしているのは、送るコードが&H00~&H1Fまでの制御コードとして扱われることを避けるためです。

例)カーソルの位置をキャラクタ座標の(3, 5)に移動させる場合

$$i=3+\&H20 \rightarrow i=\&H23$$

$$j=5+\&H20 \rightarrow j=\&H25$$

という計算を行い、ESC=#% となります。したがって、ホストが送信すべきコードは、

$$\&H1B+\&H3D+\&H23+\&H25$$

になります。

- 10) ESCT (&H1B+&H54) 現在カーソルがある行を1行消去します。カーソルはその行の先頭(左端)へ移動します。
- 11) ESC>BASIC 命令 (&H1B+&H3E+BASIC 命令)
- 12) ESC.BASIC 命令 (&H1B+&H2E+BASIC 命令)
- 13) ESC<BASIC 命令 (&H1B+&H3C+BASIC 命令)

11), 12), 13) はリモート BASIC プロトコルと呼ばれます。次の項で詳しく解説します。

---

**注意：**ターミナルが受信したスペシャル ESC シーケンスの内容はディスプレイ、プリンタのいずれにも出力されません。

---

### ●リモート BASIC プロトコル

前の項で説明したスペシャル ESC シーケンスの中の 11), 12), 13) は「リモート BASIC プロトコル」といいます。これを使うと BASIC プログラムをホストコンピュータから送り、ターミナル側で実行することができます。

なお、ホストコンピュータも PC-9800 シリーズを使用した場合には、BASIC プログラムの相互のやり取りが可能になります。

この3つのリモート BASIC プロトコルについて解説します。

それぞれホスト側から、&H1B+&H3E, &H1B+&H2E, &H1B+&H3C に続けて BASIC の命令を送ると、ターミナル側がそれを受信して命令を実行します。

送る命令はダイレクトで実行させる命令でも、プログラムでもかまいません。

プログラムを送る場合は1行ごとに C<sub>R</sub>を送る必要があります。また、行の始めには必ず ESC シーケンスを送ってください。

### ・実行結果の表示

3つのリモート BASIC プロトコルの違いは、PRINT 命令や PRINT USING 命令などテキスト画面に実行結果を出力させる命令において、表示をターミナル側で行うか、ホストコンピュータ側で行うかという点です。

ただし、ホストコンピュータ側への表示を選択した場合でもホストコンピュータ側のディスプレイに実行結果が表示されているとはかぎりません。この場合ターミナル側はディスプレイへの表示情報をホストへ送っているだけであり、それをホストコンピュータのディスプレイに表示するかどうかはホスト側の受信データの処理方法によります。

**ESC>BASIC 命令** 実行結果(表示情報)をターミナル側に表示する。

**ESC. BASIC 命令** 実行結果(表示情報)を捨てる。

**ESC<BASIC 命令** 実行結果(表示情報)をホストに送る。

---

**注意：**グラフィック画面に表示される実行結果は ESC シーケンスの指定に関わりなく、必ずターミナル側のディスプレイに表示されます。

---

・リモート BASIC プロトコルの利用例

例1) ターミナルモードの PC-9800 シリーズにグラフを表示させるためにホストからデータを送る。

```
ESC>SCREEN 0, 0
ESC>CIRCLE (100, 50), 100, 7
ESC>a=100 : b=100 : c=200 : d=50
ESC>LINE (100, 50)-(a, b), 7
ESC>LINE (100, 50)-(c, d), 7
ESC>PAINT (120, 70), 1, 7
```

例2) ターミナルモードの PC-9800 シリーズのディスクに“100”という数値を書き出させるためにホスト側からデータを送る。

```
ESC>OPEN "2:TEST1" FOR OUTPUT AS #1
ESC>PRINT #1, 100
```

---

注意：リモート BASIC プロトコルの各行の終端記号(デリミタ)には  $C_R$  を使用します。 $C_R + L_F$  の場合は  $L_F$  の制御が個別に働き、各行で 2 行改行してしまいます。

---

・リモート BASIC プロトコル使用時の注意

BASIC モードからターミナルモードに切り換えて、リモート BASIC プロトコルを使用する場合には次の点に注意してください。

リモート BASIC モードとその前の BASIC モードでは同じファイルバッファが使用されます。したがって、以前の BASIC モードで使用していたファイルバッファのデータを失わないようにするためには、リモート BASIC プロトコルで使用するファイル番号には BASIC モードで使用していたファイル番号と異なったものを定義する必要があります。

リモート BASIC プロトコル中においては、以前の BASIC モードで未使用のメモリ領域が使用されます。CLEAR 命令でメモリ使用条件を定義したり、FRE 関数で状態表示をしたりする場合には注意してください。

## 11.2 モニタモード

モニタモードを使用することにより、機械語プログラムの作成を手助けすることができます。

### ■ モニタモードの使用準備

モニタモードは通常のプログラム実行のためには不必要なため、初期状態では使用できません。モニタモードを使用する場合にはユーティリティ "switch.n88" を使ってモニタモードの使用を宣言し、DISKモードBASICを再起動してください。この方法でモニタモードの使用準備をすると、17KBのモニタ用プログラム部分が利用者メモリ上に追加ロードされます。その分、利用者メモリが減りますので注意してください。

---

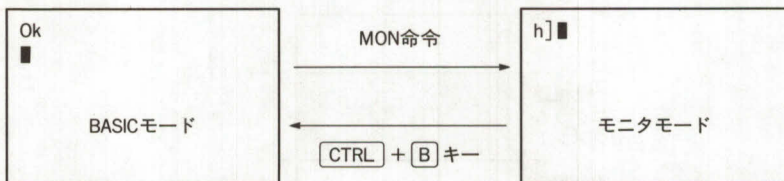
**注意：**モニタモードの使用準備を行わなくてもモニタモードの機能の一部または全部が使用できる機種もあります。

---

▶参照 switch.n88 →「10.14 メモリスイッチの設定」

### ■ モニタモードと BASIC モードとの切り換え

モニタモードと BASIC モードの切り換え関係は次のようになっています。



#### ● BASIC モードからモニタモードへの切り換え

BASIC のコマンドモードで MON 命令を実行します。

すると、"h]" が表示されます。"h]" はモニタモードのプロンプトです。これが表示されているときにはモニタモードのコマンドを実行することができます。

---

**注意：**モニタモードのプロンプトには "q]" というものもあります。モニタモード上での数値入力を 16 進数形式で行うモードの場合には "h]" が、8 進数形式の場合には "q]" が使われます。この設定は B コマンドで行います。初期状態は 16 進数形式です。

---

●モニタモードから BASIC への切り換え

モニタモードのプロンプトが出ている状態で、**CTRL** + **B** キーをタイプします。  
すると、BASIC のプロンプト “Ok” が表示され、次の行にカーソルが表示されます。

■機械語プログラムセグメントのメモリ配置

モニタモードで処理の対象とするメモリ領域のことを“機械語プログラムセグメント”といいます。モニタモードには 21 種類のコマンドが用意されており、これらを使ってメモリの読み書きを行うことができます。コマンド中で使用するアドレスは、前もって決めた機械語プログラムセグメントのセグメントベースからの相対アドレスです。

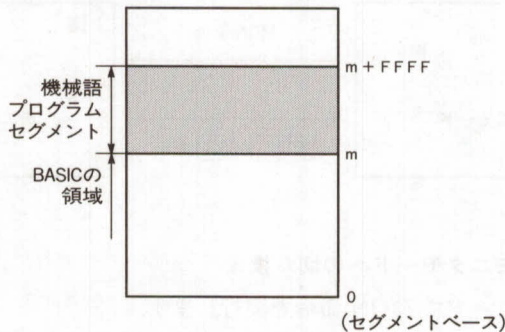
モニタモードを使用する前には機械語プログラムセグメントをメモリ上のどこに置くかを定める必要があります。

▶参照 セグメント→「第 8 章 機械語プログラム」

●モニタモードへ移行する直前に、BASIC の命令で機械語プログラムセグメントの配置を決める方法

DEF SEG 命令で機械語プログラムセグメントを定義します。

- ①CLEAR , m …… BASIC が使用するメモリ領域の上限を  $m * 16 - 1$  番地までとします。
- ②DEF SEG=m ……セグメントベースを  $m * 16$  とします。
- ③MON ……モニタモードに移行します。



注意：DEF SEG 命令を省略した場合、BASIC の上限はユーザーの使用可能な RAM メモリの上限となっているので、機械語プログラムセグメントもこの上限から始まる 64KB の範囲となります。したがってこの場合にはユーザーが使用可能なメモリ領域以外のメモリに対してアクセスを行うこととなりますので、メモリアクセス関係のモニタモードの命令を実行しても無意味となります。

### ●モニタモードで、機械語プログラムセグメントを決める方法

モニタモードのCコマンド(チェンジセグメントベースコマンド)を実行することにより、セグメントベースを変更することができます。この方法はBASICの上限が決められていないので、モニタモードのコマンドでBASICの領域を参照する場合は問題ありませんが、書き込む場合はBASICの領域を壊すおそれがありますので注意してください。

### ■モニタモードのコマンド一覧

モニタモードのコマンドは次の21種です。

| コマンド               | 意 味          | 機 能                          |
|--------------------|--------------|------------------------------|
| A                  | アセンブル        | 入力した1行をアセンブルする。              |
| B                  | ベース          | 数値の表現形式を変える(8進→16進)。         |
| C                  | チェンジセグメントベース | 機械語プログラムセグメントのセグメントベースを設定する。 |
| D                  | ダンプ          | メモリの内容をディスプレイに表示する。          |
| E                  | エディット        | スクリーンエディタの機能を用いてメモリの内容を変更する。 |
| F                  | フィル          | メモリの内容を定数で埋める。               |
| G                  | ゴー           | 機械語プログラムを実行する。               |
| I                  | インプット        | I/Oポートの値を読み込む。               |
| L                  | ディスアセンブル     | 機械語を逆アセンブルする。                |
| M                  | ムーブ          | ある範囲のメモリの内容を他のアドレスのメモリ領域へ移す。 |
| O                  | アウトプット       | I/Oポートヘデータを出力する。             |
| P                  | プリンタスイッチ     | プリンタへの出力をコントロールする。           |
| S                  | セット          | メモリにデータをセットする。               |
| SSW                | セットメモリスイッチ   | メモリスイッチにデータをセットする。           |
| TM                 | テストメモリ       | メモリをテストする。                   |
| X                  | イグザミンレジスタ    | CPUのレジスタの値を調べ、変更する。          |
| HELP<br>(CTRL + A) | ヘルプ          | コマンドとそのパラメータの形式をディスプレイに表示する。 |
| CTRL + B           | リターン         | BASICモードへ復帰する。               |
| CTRL + D           | ダンプディスク      | ディスクの内容をディスプレイに表示する。         |
| CTRL + R           | リードディスク      | ディスクからデータをロードする。             |
| CTRL + W           | ライトディスク      | ディスクヘデータをセーブする。              |

## ■コマンド使用上の規則

### ●数値の入力

モニタモードで扱う数値形式には、16進数形式と8進数形式があります。この切り換えはBコマンドで行います。初期状態では16進数形式になっています。使用できる数値の範囲は次のとおりです。

#### 16進数形式(プロンプトh]表示時)


アドレス : 4桁 0~FFFF  
 メモリの内容 : 2桁 0~FF

#### 8進数形式(プロンプトq]表示時)

アドレス : 6桁 0~177777  
 メモリの内容 : 3桁 0~377

アドレスの指定が8進数0~177777, 16進数0~FFFFの範囲内となっているのは、モニタモードでは械語プログラムセグメント内の64KBのメモリしかアクセスできないからです。

また、各コマンドでアドレスやメモリの内容を入力する場合、指定する数値を加減の算術演算子を使った式で表すことも可能です。

例) D1111+1111 






この場合、2222HがDコマンドの先頭の番地となります。

### ●文字列の入力

次の各コマンドで入力する文字列には有効桁数があります。先頭の文字からこの有効桁数分のみが有効となります。

| コマンド名 | パラメータ     | 有効桁数 |
|-------|-----------|------|
| B     | HまたはQ     | 1    |
| SSW   | メモリスイッチ番号 | 1    |
| X     | CPUレジスタ名  | 2    |
| X     | CPUフラグ名   | 1    |

### ●入力する文字の修正

コマンドの入力は  キーが押されることによって行われます。  キーを押す前に間違いに気づいた場合は、  ,  , カーソル移動キーを使って修正してください。  キーは使えません。

### ●入力時のエラー

コマンドの入力時にコマンドに間違った表現があった場合、次の行の先頭に「？」が表示されるので、もう一度入力し直してください。

### ●ディスクアクセスについての注意

複数サーフェスをもつディスクユニットを使用する場合は、サーフェスをまたがってアクセスすることはできません。また、サーフェス番号は必ず指定しなければなりません。

1MB フロッピィディスク(3.5 インチ, 5 インチ 2HD および, 8 インチ 2D)の0トラックをアクセスすることはできません。

### ●モニタモードにおける画面情報の表示形式

BASIC モードから切り換える直前の画面モードが引き継がれます。

A(アセンブル), L(ディスアセンブル)コマンド処理時の表示形式は、行当たりの文字数(40文字/行, 80文字/行)と数値形式(16進, 8進)によって異なります。アドレス, 機械語イメージ, ニーモニックに使用されるそれぞれの表示域の桁数は次のとおりです。なお, それぞれの情報の間には1桁のスペース(空白)が入ります。

|                  | アドレス |   | 機械語イメージ |   | ニーモニック |
|------------------|------|---|---------|---|--------|
| 80 文字/行, 16 進数表現 | 4    | △ | 14      | △ | 60     |
| 80 文字/行, 8 進数表現  | 6    | △ | 21      | △ | 51     |
| 40 文字/行, 16 進数表現 | 4    | △ | 14      |   | 40     |
| 40 文字/行, 8 進数表現  | 6    | △ | 21      |   | 40     |

△は1桁のスペース(空白)を表しています。40文字/行の場合は1つの処理単位が2行で表示されます。

## ■コマンド解説

### A(アセンブル)コマンド

---

書式 A [**<格納開始アドレス>**]

機能 i-8086 相当のニーモニックで入力したテキストを1行ずつアセンブルし、できた機械語を**<格納開始アドレス>**で始まるメモリに格納していきます。

解説 **<格納開始アドレス>**の指定を省略した場合には、その前に実行した A コマンドの実行終了番地の次の番地となります。  
また、前に A コマンドが実行されていない場合は0番地が格納開始アドレスになります。

①コマンドと格納開始番地を入力します。この例では 9000H 番地です。

```
h]A9000 
```

すると、次の行に格納開始アドレスが表示されます。カーソルは機械語イメージの表示域を空けてニーモニックの表示域の先頭に移動します。

```
h]A9000
```

```
9000 機械語イメージの表示域 ■
```

②ここで、ニーモニックをタイプし  キーを押します。

```
h]A9000
```

```
9000 機械語イメージの表示域 mov si, di 
```

機械語イメージの表示域にアセンブルされた結果が表示されます。そして、次のアドレスのニーモニックの入力を求めてきます。

```
h]A9000
```

```
9000 89FE mov si, di
```

```
9002 ■
```

③このようにして、ニーモニックを画面の右側に入力すると、対応する機械語の割り付けられたアドレスと機械語の内容が左側に表示されていきます。

④ A コマンドの状態から抜け出すには、アドレスが表示されたのち、 キーのみを押すか、 キー、または  +  を入力してください。モニタモードのプロンプトにもどります。

---

注意：A コマンドでは 64KB を超えるような動作をする命令(たとえば、セグメント間の CALL/JUMP/RET 命令など)をアセンブルすることはできません。

---

## B(ベース)コマンド

---

書式 B( $\left. \begin{array}{|c|} \hline H \\ \hline Q \\ \hline \end{array} \right\}$ )

機能 モニタモードで取り扱う数値の形式(16進または8進)を設定します。

解説 hを指定すると16進数形式を採用し、qを指定すると8進数形式となります。モニタのプロンプトは16進数形式の場合には“h”に、8進数形式の場合には“q”となります。

## C(チェンジセグメントベース)コマンド

---

書式 C(<セグメントベース>)

機能 機械語プログラムセグメントのセグメントベースを変更します。

解説 <セグメントベース>に指定した値を16倍した値が実際の(物理アドレスでの)ベースアドレスとなります。<セグメントベース>の指定を省略した場合には現在のセグメントベースを表示します。

Cコマンドでセグメントベースを設定した後、A、D、E、L、Sコマンドにおいてアドレス指定を省略した場合には、新しく設定されたセグメント内の0番地がコマンドの対象番地となります。

## D(ダンプ)コマンド

---

書式 D(<表示開始アドレス>)[, <表示終了アドレス>]

機能 メモリの内容をディスプレイに表示します。

解説 16進数形式になっている場合にはメモリの内容に対応する1バイト文字も表示されます。さらに、P(プリンタスイッチ)コマンドでプリンタモードが指定されていれば、結果がプリンタへも出力されます。

<表示開始アドレス>から<表示終了アドレス>までのメモリの内容が表示されます。  
<表示終了アドレス>を省略すると<表示開始アドレス>から16バイト分のメモリの内容が表示されます。

〈表示開始アドレス〉を省略して、`,` (コンマ)と〈表示終了アドレス〉を指定した場合は、その前に実行したDコマンドで表示した最後のアドレスの次のアドレスから〈表示終了アドレス〉までのメモリの内容を表示します。前にDコマンドを使っていない場合は、0番地から表示します。

〈表示開始アドレス〉も〈表示終了アドレス〉も省略した場合には、その前に実行したDコマンドで表示した最後のアドレスの次のアドレスから16バイト分のメモリの内容を表示します。前にDコマンドを使っていない場合は、0番地から表示します。

Dコマンドの実行を中止したい場合は、**STOP**キー、または**CTRL**+**C**キーを押します。また、コマンドの実行を一時停止させるには、**CTRL**+**S**キーを押します。表示を再開させるには**STOP**キー、**CTRL**+**C**キー以外を押します。

Dコマンド終了後に**↵**キーのみを押すと、パラメータを省略してDコマンドを実行したのと同じになります。

---

**注意**：Dコマンドで内容を表示させることができる連続したメモリ空間は64KBです。

---

## E(エディット)コマンド

---

**書式** E[〈開始アドレス〉]

**機能** メモリの内容をディスプレイに表示します。必要ならば、表示されている内容を画面上で変更できます。画面上でデータを変更するとそのままメモリの内容も変更されます。

**解説** 〈開始アドレス〉を省略した場合には、前に行ったEコマンドの終了したときにカーソルがあったアドレスが〈開始アドレス〉となります。

カーソル移動キー(**←** **→** **↑** **↓**)と**ROLL UP**キー、**ROLL DOWN**キーを使って変更したいメモリの内容のところへカーソルを移動し、新しい数値をタイプしてください。メモリの内容も変更されます。

エディットを終了するときは**STOP**キー、**CTRL**+**C**キー、または**ESC**キーを押してください。

---

**注意**：Eコマンドを使用する場合、モニタモードに入る直前のBASICのテキスト画面のスクロール開始行が0、スクロール行数が19以上でなければなりません。スクロール行の設定は**CONSOLE**命令で行います。

---

## F(フィル)コマンド

---

**書式** F <開始アドレス>, <終了アドレス>, <定数>

**機能** アドレスで指定した範囲のメモリの内容を指定した定数の値で置き換えます。

**解説** パラメータはいずれも省略できません。<定数>に設定する値は16進数で0~FF, 8進数で0から377までの値です。

---

**注意:** F コマンドで扱える連続したメモリ空間は最大 64KB です。

---

## G(ゴー)コマンド

---

**書式** G[<実行開始アドレス>][, <ブレイクポイントアドレス 1>][, <ブレイクポイントアドレス 2>]

**機能** 指定したアドレスにジャンプして、そこから機械語プログラムの実行を開始します。

**解説** <ブレイクポイントアドレス 1>および<ブレイクポイントアドレス 2>により、2つまでブレイクポイントを指定できます。機械語プログラムの実行中、実行アドレスがこのブレイクポイントに来ると、実行は中断され、モニタモードのプロンプトにもどってきます。

<実行開始アドレス>を省略した場合、最後に実行したGコマンドのブレイクポイントが実行開始アドレスとなります。また、前にGコマンドを実行していない場合には<実行開始アドレス>は省略できません。

ブレイクポイントで実行が中断された場合は、CPUのレジスタの値が保存されます。

ブレイクポイントに達したときにはそのアドレスが次のように表示されます。

16進形式の場合      \* xxxx : yyyy

8進形式の場合      \* xxxxxx : yyyyyy

ともに、x~x はセグメントベースを、y~y はオフセットアドレスを示します。

## I(インプット)コマンド

---

書式 I<ポートアドレス>

機能 入力ポートの値(入力データ)を読み画面に表示します。

解説 パラメータは省略できません。<ポートアドレス>には、16進形式のモードならば0～FF、8進形式のモードならば0～377の値を指定します。

Iコマンドを実行すると入力ポートの値を表示します。ここでスペースキーを押すと同じ入力ポートの次の値を表示します。キーを押すと処理は終了してモニタモードのプロンプトが表示されます。

## L(ディスアセンブル)コマンド

---

書式 L[<逆アセンブル開始アドレス>][, <逆アセンブル終了アドレス>]

機能 メモリの内容をi-8086相当のニーモニックコードに逆アセンブルして表示します。

解説 <逆アセンブル開始アドレス>から<逆アセンブル終了アドレス>までのメモリの内容がニーモニックに逆アセンブルされます。

<逆アセンブル終了アドレス>を省略すると<逆アセンブル開始アドレス>から12行分が逆アセンブルされます。

<逆アセンブル開始アドレス>を省略して、","(コンマ)と<逆アセンブル終了アドレス>を指定した場合は、その前に実行したLコマンドで逆アセンブルした最後のアドレスの次のアドレスから<逆アセンブル終了アドレス>までのメモリの内容を逆アセンブルします。前にLコマンドを使っていない場合は、0番地から逆アセンブルします。

<逆アセンブル開始アドレス>も<逆アセンブル終了アドレス>も省略した場合には、その前に実行したLコマンドで逆アセンブルした最後のアドレスの次のアドレスから12行分を逆アセンブルして表示します。前にLコマンドを使っていない場合は、0番地から表示します。

Lコマンド実行中に、実行を中断したい場合は+キーを入力してください。実行を再開するためには、キー、+キー以外のキーを押してください。

ニーモニックのオペランドは、現在指定されている数値形式で表示されます。

Lコマンド終了後にキーのみを押すと、パラメータを省略してLコマンドを実行したのと同じになります。

---

**注意：**L コマンドで逆アセンブル可能な連続したメモリ空間は 64KB です。

したがって、これを超えるような動作をする命令、アドレス(たとえば、セグメント間 CALL/JUMP/RET 命令など)を逆アセンブルすることはできません。

---

## M(ムーブ)コマンド

---

**書式** M<転送するメモリ領域の先頭アドレス>, <転送するメモリ領域の最終アドレス>, <転送先の先頭アドレス>

**機能** メモリ内で、ある位置から別の位置へメモリ領域のブロック転送を行います。

**解説** パラメータはいずれも省略できません。転送の確認は D コマンドで行ってください。

---

**注意：**M コマンドでは機械語プログラムセグメントの最大メモリ (64KB) を超えるような転送はできません。

---

## O(アウトプット)コマンド

---

**書式** O <ポートアドレス>, <データ>

**機能** 指定したポートアドレスに、データを出力します。

**解説** <ポートアドレス>, <データ>は、16 進形式のモードならば 0~FF, 8 進形式のモードならば 0~377 の値でなければなりません。

## P(プリンタスイッチ)コマンド

書式 P

機能 D(ダンプ), L(ディスアセンブル), **CTRL**+**D** (ダンプディスク)コマンドの実行結果をプリンタに出力するかどうかを設定します。

解説 モニタモードに切り換わった当初は、プリンタスイッチが“OFF”の状態となっており、実行結果はプリンタに出力されません。P コマンドを実行するたびにプリンタスイッチの“ON”, “OFF” が切り換わります。

プリンタスイッチの状態に応じて、モニタモードのプロンプトが次のように変わります。

| プリンタスイッチ | OFF | ON |
|----------|-----|----|
| 16進形式    | h]  | h) |
| 8進形式     | q]  | q) |

なお、D, L, **CTRL**+**D**以外のコマンドの実行結果はプリンタへは出力されません。

注意：モニタモードを使用しているときも、**COPY**キーを押すことによって画面のハードコピーがとれます。

## S(セット)コマンド

書式 S[<開始アドレス>]

機能 メモリの内容をバイトごとに表示します。これを確認して内容を変更することができます。

解説 ①<開始アドレス>を指定した後、**↵**キーを入力すると、次の行にアドレスと内容を表示し、新しい内容の入力待ちになります。

```
h]S9000
9000 D3-■
```

②メモリの内容を変更する場合は数値をタイプします。すると、メモリの値が変更されます。このあと、スペースキーを押すと次のアドレスの内容が表示されるので続けてメモリの内容を変更することができます。変更をやめるには**↵**キーを押してください。モニタモードのプロンプトにもどります。

```
h]S9000          h]S9000
9000 D3-A3 D4-■  9000 D3-A4 ↵
h]■
```

③メモリの内容を変更しない場合は、数値をタイプせずに、スペースキーを押してください。内容は変更されずにカーソルは次のアドレスに移動します。

④カーソルを1つ前にもどりたい場合は、**CTRL** + **B** キーを押してください。

## SSW(セットメモリスイッチ)コマンド

書式 SSW[<スイッチ番号>]

機能 メモリスイッチの内容を表示します。これを確認して内容を変更することができます。

解説 ①スイッチ番号を入力しなかった場合は、すべてのメモリスイッチの内容を16進表記2桁で表示し、モニタモードのプロンプトにもどります。

h]SSW

| SW1 | SW2 | SW3 | SW4 | SW5 | SW6 | SW7 |
|-----|-----|-----|-----|-----|-----|-----|
| 48  | 05  | 04  | 00  | 00  | 00  | 00  |

h]■

②スイッチ番号(1~7)を指定した後、**↵**キーを押すと次の行に指定したスイッチの内容を表示し、数値の入力待ちになります。

h]SSW3

00-■

③メモリスイッチの内容を変更する場合は数値をタイプします。すると、メモリスイッチの値が変更されます。このあと、スペースキーを押すと次のスイッチ番号の内容が表示されるので続けてメモリスイッチの内容を変更することができます。変更をやめるには**↵**キーを押してください。モニタモードのプロンプトにもどります。

h]SSW3

00-07 00-■

h]SSW3

00-07 **↵**

h]

④メモリの内容を変更しない場合は、数値をタイプせずに、スペースキーを押してください。内容は変更されずにカーソルは次のスイッチの処理に移ります。

⑤スペースキーを利用しながら、1行で全部のスイッチの値をセットすることができます。最後のスイッチの値のセットが終わると自動的にモニタモードのプロンプトにもどります

---

**注意：**メモリスイッチの変更を行った後はディップスイッチ SW2 の 5 番が ON であることを確認して、必ず一度リセットボタンを押して再起動させてください。これでメモリスイッチの変更が有効となります。

---

## TM(テストメモリ)

---

書式 TM

機能 本体に実装されている RAM メモリをテストします。

解説 テスト対象メモリ領域はユーティリティ "switch.n88" の "メモリサイズ" で設定されている利用者メモリ領域と、システム領域(テキスト VRAM, グラフィック VRAM 領域)です。

- ①テストメモリを行う前には、次の命令を実行して、グラフィック画面の状態を初期化し、テキスト画面状態を 80 桁モードにしておいてください。

```
SCREEN 0, 2
WIDTH 80, 25
```

- ②ユーティリティ "switch.n88" で設定されている利用者メモリ領域のサイズ(Real memory)と BASIC 起動時のマイクロ診断によって正常と確認されたメモリサイズ(Active memory)とが一致しない場合には、メモリをテストする前に次のメッセージを表示します。

```
Real memory : xxxKB
Active memory : xxxKB
```

この場合にはなぜ相違しているかを確認した後、ファンクションキー以外のキーを押すとメモリテストを開始します。

- ③メモリに異常がなければ、テスト終了後、

```
Test complete !
```

と表示されます。

テストメモリ終了までの経過時間は機種および環境設定により、多少異なりますが、およそ次のようになります。

| 利用者メモリ領域  | 経過時間   |
|-----------|--------|
| 384KB の場合 | 約 15 分 |
| 512KB の場合 | 約 20 分 |
| 640KB の場合 | 約 25 分 |

異常アドレスを検出すると、スピーカが鳴り異常発生アドレスを 16 進数 5 桁(物理アドレス)で表示します。

- ④テストが終了しても、モニタモード(または BASIC モード)にはもどりません。再起動の必要があるときは、リセットスイッチを押してください。

## X(イグザミンレジスタ)コマンド

---

書式 X{ | <CPU レジスタ名> | }  
           | <CPU フラグ名> | }

機能 CPU の全レジスタと CPU フラグの内容の表示および変更を行います。

解説 ① <CPU レジスタ名> には次のようなレジスタの名前を指定します。

AX, BX, CX, DX, SP, BP, SI, DI, CS, DS, SS, ES, IP

レジスタ名が指定できるのは 16 ビットのペアレジスタのみです。AH, AL…などは指定できません。

② <CPU フラグ名> には次のようなフラグの頭文字を指定します。

|                      |                     |
|----------------------|---------------------|
| O : Overflow         | Z : Zero            |
| D : Direction        | A : Auxiliary Carry |
| I : Interrupt Enable | P : Parity          |
| T : Trap             | C : Carry           |
| S : Sign             |                     |

③<CPU レジスタ名>, <CPU フラグ名>を省略した場合はフラグの全ステータスと全レジスタの内容を表示します。


```
FL  -----
Ax  0000
BX  0000
CX  0000
DX  0000
SP  F7F6
BP  0000
SI  0000
DI  0000
CS  0000
DS  0000
SS  0060
ES  0000
IP  0000
h]■
```

FL は 9 個のフラグのステータスを示します。“ON”の場合にはそのフラグの頭文字が表示されます。“OFF”の場合にはこの例のように “-” が表示されます。

- ④ <CPUレジスタ名>を指定した場合は、そのCPUレジスタ名と内容を表示し新しい値の入力待ちになります。

h]XSP

SP F7F6 ■

- ⑤ CPUレジスタの内容を変更する場合は数値をタイプします。すると、レジスタの値が変更されます。このあと、スペースキーを押すと次のCPUレジスタ名と内容が表示されるので続けてCPUレジスタの値を変更することができます。変更を終えるには  キーを押してください。モニタモードのプロンプトにもどります。

h]XSP


SP F7F6 0000



BP 0000 ■

h]XSP

SP F7F6 0000 

h]■

- ⑥ 内容を変更しない場合は、数値をタイプしないで  キーを押してください。モニタモードのプロンプトにもどります。また、スペースキーを押すと次のレジスタの処理へ移ります。

- ⑦ <CPUフラグ名>を指定した場合は、そのフラグの現在のステータスが表示されます。“ON”の場合は1、“OFF”の場合は0が表示され、新しいステータスの入力待ちとなります。内容を変更する場合には新しい値をタイプして  キーを押してください。フラグの内容を変更したのち、モニタモードのプロンプトにもどります。変更しない場合は、新しい値をタイプしないで  キーを押してください。フラグの内容は変更されずモニタモードのプロンプトにもどります。

- ⑧ X(イグザミンレジスタ)コマンドでレジスタやフラグの値を変更した場合、有効となるのはG(ゴー)コマンドで機械語プログラムを実行する場合だけです。BASICにもどるときは有効ではありません。

**HELP, CTRL + A (コマンドの使い方の表示)コマンド**

---

**書式** HELP キーまたは CTRL + A キーをタイプします。

**機能** モニタモードで使用できるコマンドの種類とそのコマンドのパラメータが画面に表示されます。

**CTRL + B (BASIC へもどる)コマンド**

---

**書式** CTRL + B キーをタイプします。

**機能** モニタモードから BASIC モードにもどります。

**CTRL + D (ダンプディスク)コマンド**

---

**書式** CTRL + D <ドライブ番号> [, <サーフェス番号>], <表示開始セクタのトラック番号>, <表示開始セクタのセクタ番号> [, <表示終了セクタのトラック番号>, <表示終了セクタのセクタ番号>]

**機能** ディスクの内容を画面に表示します。

**解説** <ドライブ番号> にはダンプを行うドライブの番号を指定します。

<サーフェス番号> にはダンプを行うディスクのサーフェス番号を指定します。複数サーフェスをもつドライブの場合は必ず指定してください。

<表示終了セクタのトラック番号> と <表示終了セクタのセクタ番号> の指定を省略した場合は表示開始セクタだけ表示します。

実行を中止したい場合は、STOP キーまたは、CTRL + C キーをタイプしてください。

また、一時中断したい場合には CTRL + S キーをタイプし、再開するときは STOP, CTRL + C キー以外のキーを押してください。

---

**注意:** CTRL + D コマンドでは異なったトラック間にわたってダンプを行うことはできません。

---

---

**CTRL + R (リードディスク) コマンド**

---

**書式** CTRL + R <ドライブ番号> [, <サーフェス番号>], <トラック番号>, <セクタ番号>, <開始アドレス>, <終了アドレス>

**機能** ディスクからメモリにデータをロードします。

**解説** <ドライブ番号>, <サーフェス番号>, <トラック番号>, <セクタ番号>で表されるディスクのアドレスからメモリ上の<開始アドレス>～<終了アドレス>へデータを読み込みます。

---

**CTRL + W (ライトディスク) コマンド**

---

**書式** CTRL + W <ドライブ番号> [, <サーフェス番号>], <トラック番号>, <セクタ番号>, <開始アドレス>, <終了アドレス>

**機能** メモリの内容をディスクにセーブします。

**解説** メモリ上の<開始アドレス>～<終了アドレス>で表されるメモリの内容を, <ドライブ番号>, <サーフェス番号>, <トラック番号>, <セクタ番号>で表されるディスクのアドレスを先頭セクタとして, 連続したディスクのアドレスへ書き出します。

---

**注意:** このコマンドはディスクに直接書き込みを行うので注意してください。

特にSCSI固定ディスクで複数ドライブを使用している場合は, ドライブ番号の指定に注意してください。

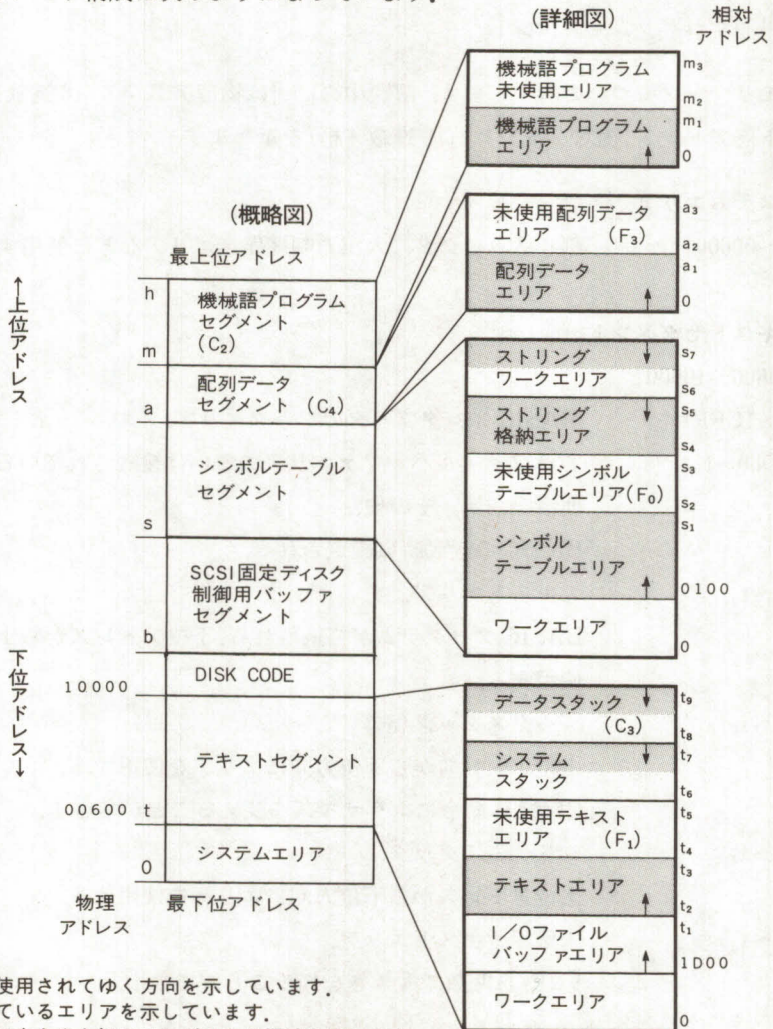
---

# 付録 A

## メモリマップ

### ■ N<sub>88</sub>-BASIC(86)のメモリ構成

N<sub>88</sub>-BASIC(86)のメモリ構成は次のようになっています。



↑または↓: メモリが使用されてゆく方向を示しています。

■: 使用されているエリアを示しています。

F<sub>n</sub>: FRE関数で大きさを知ることができる領域を示します。  
たとえば、F<sub>0</sub>は、fre(0)で知ることができます。

C<sub>n</sub>: CLEAR命令により確保できる領域を示しています。  
たとえば、C<sub>2</sub>はCLEAR命令の第2パラメータで確保します。

概略図左側のアドレス: 物理アドレスを16進数で示しています。

詳細図右側のアドレス: セグメント内アドレス(相対アドレス)を16進数で示しています。

## ■システムエリアと5つのセグメント

メモリ全体は、システムエリアと DISK CODE と 5 つのセグメントに区分されています。

- システムエリア
- テキストセグメント
- DISK CODE
- SCSI固定ディスク制御用バッファセグメント
- シンボルテーブルセグメント
- 配列データセグメント
- 機械語プログラムセグメント

メモリマップについて解説します。解説中の [ ] は物理アドレス (16 進数 5 桁)、( ) はセグメント内アドレス (相対アドレス) (16 進数 4 桁) を示します。

### ●システムエリア

[0~00600] 割り込みベクタ, 入出力制御作業エリアなどに使用する。

### ●テキストセグメント

[00600~10000]

- |                                   |  |
|-----------------------------------|--|
| (0~1CFF)                          | BASIC インタプリタのワークエリア。   |
| (1D00~t <sub>1</sub> )            | I/O ファイルバッファエリアあるいは接続されているディスク装置の管理情報領域, その他。<br>BASIC の起動時に確定される。       |
| (t <sub>2</sub> ~t <sub>3</sub> ) | テキストエリア。<br>BASIC プログラムが格納される。下位アドレスから上位方向にエリアを使用する。                     |
| (t <sub>8</sub> ~t <sub>9</sub> ) | データスタック (演算スタック)。<br>上位アドレスから下位方向にエリアを使用する。<br>CLEAR 命令によりサイズを変えることができる。 |
| (t <sub>6</sub> ~t <sub>7</sub> ) | システムスタック。<br>上位アドレスから下位方向にエリアを使用する。                                      |
| (t <sub>4</sub> ~t <sub>5</sub> ) | 未使用テキストエリア。<br>FRE(1)関数で大きさを知ることができる。                                    |

---

注意: サウンド, GP-IB, RS-232C (第 2 回線 / 第 3 回線) などの拡張機能使用時は t<sub>1</sub>~t<sub>2</sub> の間システムにより使用されます。この分だけテキストエリアが少なくなります。

---

## ● DISK CODE

[10000~b] DISK モード BASIC 起動時に DISK CODE がロードされる。

## ・ DISK CODE のサイズについて

DISK CODE のサイズはシステムの起動形態によって異なります。メモリにロードされるサイズは次のとおりです。

| 機 能                                   | サイズ    |
|---------------------------------------|--------|
| 標 準                                   | 24KB   |
| AI逐次変換, AI連文節変換, 逐次変換, 連文節変換入力を選択した場合 | +121KB |
| 単文節変換入力機能の場合                          | +42KB  |
| 拡張グラフィックモードでシステムが起動された場合              | +22KB  |
| 拡張画面ハードコピー機能が選択された場合                  | +8KB   |
| モニタモードを使用する場合                         | +17KB  |
| 電話制御機能を使用する場合                         | +20KB  |
| 拡張フォーマットの固定ディスクを使用する場合                | +18KB  |
| SCSI固定ディスクを使用する場合                     | +17KB  |

例 1) AI 逐次変換, AI 連文節変換, 逐次変換, 連文節変換入力機能を使用する場合

標準 DISK CODE (24KB) + AI 逐次変換, AI 連文節変換, 逐次変換, 連文節変換  
入力手続き (121KB) → 145KB

例 2) 単文節変換入力機能を使用する場合

標準 DISK CODE (24KB) + 単文節変換入力手続き (42KB) → 66KB

例 3) AI 逐次変換, AI 連文節変換, 逐次変換, 連文節変換入力機能に加え拡張グラフィック  
モードを使用する場合

標準 DISK CODE (24KB) + AI 逐次変換, AI 連文節変換, 逐次変換, 連文節変換  
入力手続き (121KB) + 拡張グラフィック手続き (22KB) → 167KB

例 4) 日本語入力機能は使用せず, 拡張グラフィックモードを使用する場合

標準 DISK CODE (24KB) + 拡張グラフィック手続き (22KB) → 46KB

ROM モード BASIC の場合 DISK CODE は必要としないので, [10000] からシンボルテール  
セグメントがとられます。

## ● SCSI 固定ディスク制御用バッファセグメント

[b~s] SCSI 固定ディスクの入出力制御を行うためのエリア。アクティブな BASIC  
領域のある SCSI 固定ディスクが接続されていない場合には確保されな  
い。

●シンボルテーブルセグメント

|                                   |   |
|-----------------------------------|---|
| [s~a]                             | シンボルテーブルセグメント全体のサイズは CLEAR 命令による配列データセグメントサイズの増減により変化する。  |
| (0100~s <sub>1</sub> )            | シンボルテーブルエリア。<br>ラベル、変数名、関数名とその属性、数値型の変数データの値を格納する。  |
| (s <sub>6</sub> ~s <sub>7</sub> ) | 下位アドレスから、上位方向にエリアを使用する。<br>文字型変数のストリングディスクリプタもこのエリアに格納される。<br>ストリングワークエリア。                                |
| (s <sub>4</sub> ~s <sub>5</sub> ) | 文字列の演算のための作業エリア。2K バイトの領域。<br>ストリング格納エリア。   |
| (s <sub>2</sub> ~s <sub>3</sub> ) | 上位アドレスから下位方向にエリアを使用する。文字型変数あるいは文字型配列変数に代入される文字列はこのエリアに格納される。<br>未使用シンボルテーブルエリア。<br>FRE(0)関数で大きさを知ることができる。 |

---

注意：シンボルテーブルセグメントの大きさは最大 64KB です。

---

●配列データセグメント

|                                   |   |
|-----------------------------------|---|
| [a~m]                             | CLEAR 命令によりサイズを変えることができる。                                 |
| (0~a <sub>1</sub> )               | 数値型配列のデータと文字型配列のストリングディスクリプタが格納される。下位アドレスから上位方向にエリアを使用する。 |
| (a <sub>2</sub> ~a <sub>3</sub> ) | 未使用配列データエリア。FRE(3)関数で大きさを知ることができる。                        |

---

注意：1つの数値型配列変数の大きさは最大 64KB です。

---

●機械語プログラムセグメント

|                                   |   |
|-----------------------------------|---|
| [m~h]                             | 機械語プログラムをロードするためのエリア。BASIC システムに干渉されない。<br>この m 番地は、CLEAR 命令の第 2 パラメータにより、セグメントベース値(物理アドレスを 16 で割った値)で定義する。 |
| (0~m <sub>1</sub> )               | 機械語プログラムで使用するエリア。下位アドレスから上位方向へエリアを使用する。   |
| (m <sub>1</sub> ~m <sub>2</sub> ) | 機械語プログラムの未使用エリア。  |

## ■メモリの使用状況

FRE 関数を使ってメモリの使用状況を知ることができます。

FRE(0) = 未使用シンボルテーブルエリア (F0) ( $=s_3 - s_2$ )

FRE(1) = 未使用テキストエリア (F1) ( $=t_5 - t_4$ )

FRE(2) = F0 + F1

FRE(3) = 未使用配列データエリア (F3) ( $=a_3 - a_2$ )

## ■メモリ領域の配分・確保

CLEAR 命令を使います。メモリマップと CLEAR 命令のパラメータとの関係を示します。

CLEAR     [〈ダミーパラメータ〉][, 〈メモリの上限〉][, 〈スタックの大きさ〉][, 〈配列データ領域の大きさ〉]

ダミーパラメータ     : 省略すること。ただし CLEAR 直後のコンマ(,)は必要。

メモリの上限         : m の値。BASIC が使用するメモリの上限値で、機械語プログラムセグメントの先頭番地に等しい。実装メモリの上限(h)とは違う点に注意。

スタックの大きさ     : C<sub>3</sub>(データスタック)の値。

配列データ領域の大きさ : C<sub>4</sub>(配列データセグメント)の値。



# 付録 B

## データの内部形式

### ■変数の型

次の 4 種類の変数の型があります。

| 変数の型   | 数値部分のメモリ           | 型宣言文*  | 型宣言文字** | 変数名の例    |
|--------|--------------------|--------|---------|----------|
| 整数型    | 2 バイト              | DEFINT | %       | AB %     |
| 単精度実数型 | 4 バイト              | DEFSNG | !       | AB, AB ! |
| 倍精度実数型 | 8 バイト              | DEFDBL | #       | AB #     |
| 文字型    | ストリングディスクリプタ 4 バイト | DEFSTR | \$      | AB \$    |

\* 型宣言文がないときは DEFSNG が実行されたものとして処理される。

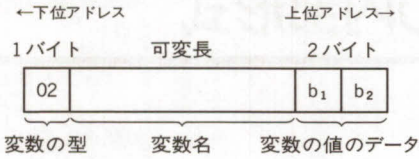
\*\* 変数に型宣言文字をつけないと型宣言文で指定した型となる。

### ■変数のメモリ表現とデータ形式

BASIC で扱う変数がメモリ上でどのように表現されているかを図解します。

● 整数型変数

メモリ表現



**変数の型** "02" は変数が整数型であることを表す。

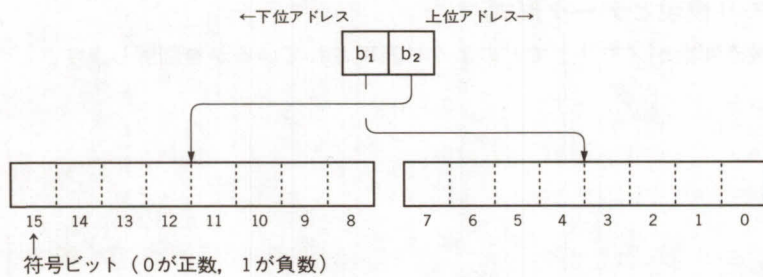
**変数名**

(変数名の長さ) - 1      先頭文字を除いた残りの変数名(可変長).  
偶数バイトに調整される。

例) 変数名が ABCDEFG% の場合  
06 42 43 45 46 47

**変数の値のデータ** b<sub>1</sub> に変数の値の下位バイトが格納され、b<sub>2</sub> に上位バイトが格納されている。

データ形式

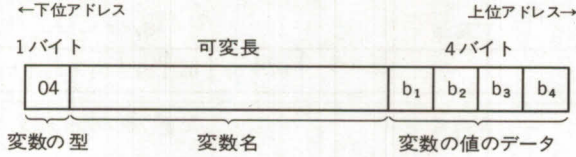


整数の表現範囲 16進数 7FFF ~ -8000 (2の補数表現)  
10進数 32767 ~ -32768  
このように、2の補数表現で表す。

|      |      |
|------|------|
| ∴    | ∴    |
| 0002 | → 2  |
| 0001 | → 1  |
| 0000 | → 0  |
| FFFF | → -1 |
| FFFE | → -2 |
| ∴    | ∴    |

●単精度実数型変数

メモリ表現

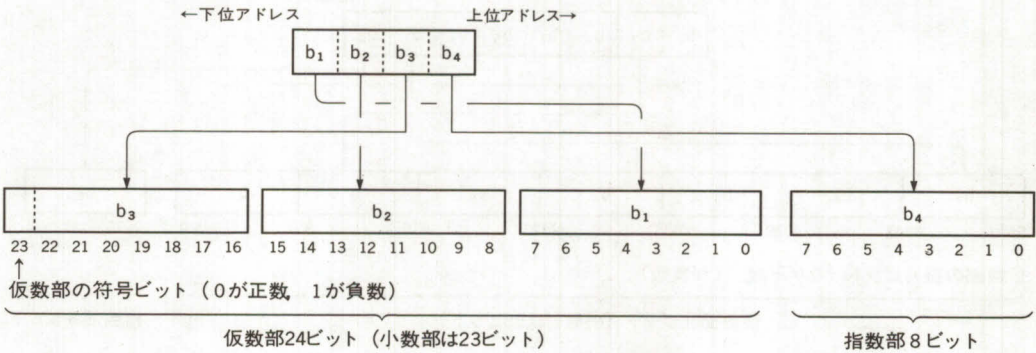


**変数の型** "04"は変数が単精度実数型であることを表す。

**変数名** 格納のされ方は整数型の場合と同じ。

**変数の値のデータ** b<sub>1</sub>, b<sub>2</sub>, b<sub>3</sub>, b<sub>4</sub>には変数の値のデータが格納されている。

データ形式

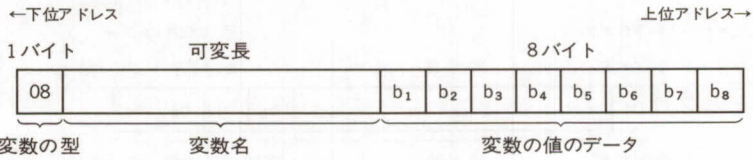


- ・仮数部は24ビットからなり、そのうち最上位の1ビットは符号表示、残り23ビットが小数部を表す。実際の仮数の値は、この23ビット分のデータの左側に小数点があり、さらにその左に1があるものとして扱われる。
- ・指数部は8ビットからなり、その表現形式は16進数で、81Hを0とし、82Hを1、80Hを-1として表現している。FFHは126、01Hは-128。  
00Hは仮数部にどんなデータが入っていても、この変数の値を0とする。

|    |              |
|----|--------------|
| FF | 126          |
| FE | 125          |
| ⋮  | ⋮            |
| 82 | 1            |
| 81 | 0            |
| 80 | -1           |
| ⋮  | ⋮            |
| 02 | -127         |
| 01 | -128         |
| 00 | この変数の値を0にする。 |

●倍精度実数型変数

メモリ表現

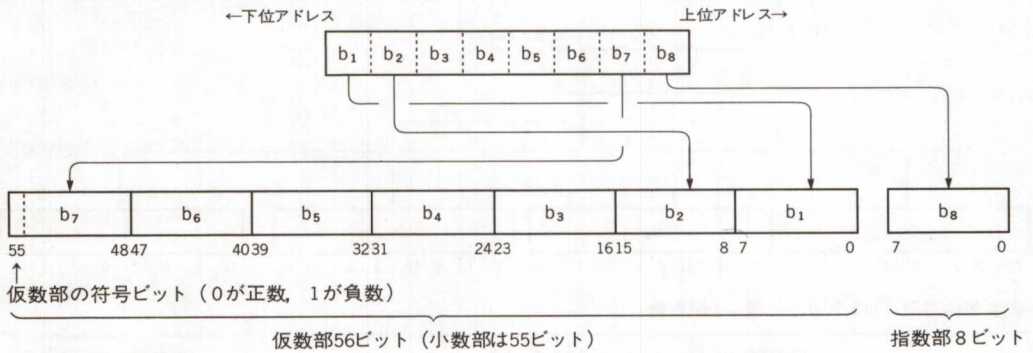


**変数の型**    ^08^ は変数が倍精度実数型であることを表す。

**変数名**    格納のされ方は整数型の場合と同じ

**変数の値のデータ**    b<sub>1</sub>~b<sub>8</sub>には変数の値のデータが格納されている。

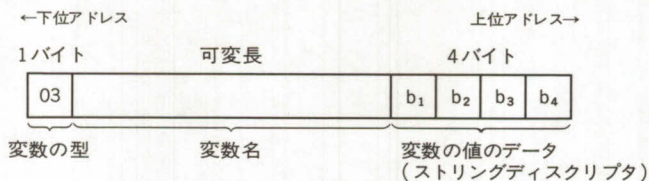
データ形式



- ・ 仮数部は56ビットからなり、そのうち最上位の1ビットは符号表示、残り55ビットが小数部を表す。実際の仮数の値は、この55ビット分のデータの左側に小数点があり、さらにその左に1があるものとして扱われる。
- ・ 指数部は8ビットからなり、その表現形式は単精度実数型と同じ。

## ●文字型変数

## メモリ表現

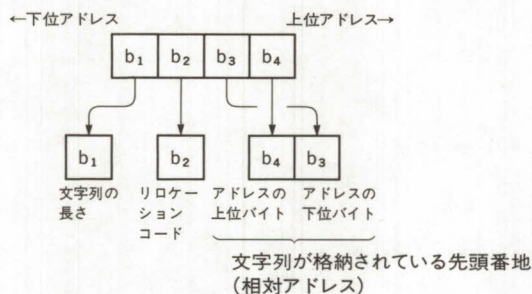


**変数の型** "03"は変数が文字型であることを表す。

**変数名** 格納のされ方は整値型の場合と同じ。

**変数の値のデータ** b<sub>1</sub>, b<sub>2</sub>, b<sub>3</sub>, b<sub>4</sub>には変数の値に関するデータが格納されている。

## データ形式



- ・リロケーションコードは文字列が格納されているセグメントを表す。
  - 00: ストリング格納エリア (シンボルテーブルセグメント内)
  - 02: テキストセグメント
  - 04: I/Oファイルバッファ (テキストセグメント内)



# 付録 C

## ディスクの内部構造

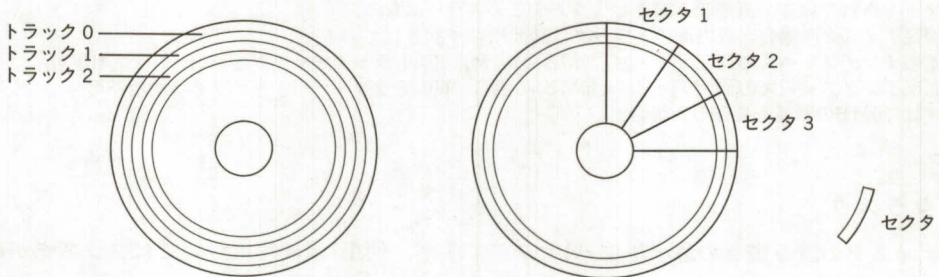
### ■トラックとセクタ

フォーマットされたディスクはトラックとセクタという単位に分割されます。

トラックとは、ディスクの外周から内周に向かって同心円上に並んだ部分のことで、外側から順にトラック 0, トラック 1, トラック 2……と番号が振られています。

セクタは各トラックを扇形に分割した部分で、セクタ 1, セクタ 2……と番号が振られています。ディスクの読み書きはこのセクタ単位に行われます。

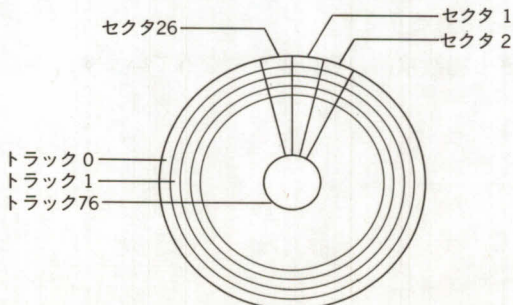
トラックとセクタの数はディスクの種類によって異なります。



また、ディスクの面のことを“サーフェス”といい、サーフェス 0, サーフェス 1……と番号が振られています。

ディスク内でサーフェス番号, トラック番号, セクタ番号を指定すれば固有のセクタを指し示すことができます。これをディスクのアドレスと呼びます。

ディスクの内部構造(3.5 インチ, 5 インチ 2HD および 8 インチ 2D の場合)



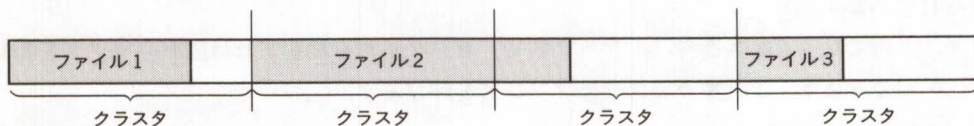
ディスクの物理的構造は次のとおりです

| ディスクタイプ  | サーフェス数 | サーフェス当りのトラック数 | トラック当りのセクタ数 | クラスタ当りのセクタ数 | クラスタ総数 | ディレクトリの大きさセクタ数 | 作成できるファイル数 | クラスタ当りのトラック数 |   |
|--|--------|---------------|-------------|-------------|--------|----------------|------------|--------------|---|
| 3.5インチ, 5インチ<br>2DD(両面倍密度倍トラック)                | 2      | 80            | 16          | 16          | 160    | 12             | 159        | 1            |   |
| 3.5インチ, 5インチ<br>2HD(両面高密度倍トラック), 8インチ2D(両面倍密度) | 2      | 77            | 26          | 26          | 154    | 22             | 151        | 1            |   |
| 従来インタフェースの<br>固定ディスク                           | 5MB    | 4             | 153         | 33          | 33     | 612            | 99         | 603          | 1 |
|  | 10MB   | 4             | 310         | 33          | 33     | 1240           | 99         | 1231         | 1 |
|  | 20MB   | 8             | 308         | 33          | 33     | 2460           | 231        | 2443         | 1 |
|  | 40MB   | 8             | 614         | 33          | 66     | 2404           | 231        | 2391         | 2 |
| SCSI固定<br>ディスク                                 | 20MB   | 4             | 440         | 47          | 47     | 1680           | 141        | 1675         | 1 |
|  | 40MB   | 8             | 440         | 47          | 47     | 3432           | 329        | 3423         | 1 |
|  | 130MB  | 10            | 823         | 64          | 64     | 7200*          | 576        | 7189*        | 1 |

- セクタの大きさは、一部の例外を除いて 256 バイトの共通の値になっている。3.5 インチ, 5 インチ 2HD および 8 インチ 2D ディスクの場合、サーフェス番号 0, トラック番号 0 は 1 セクタ 128 バイトでフォーマットされている。直接読み書きしようとするとうエラーになる。
- 固定ディスクの場合、最内周のトラックは診断用に予約されている。
- 従来インタフェースの固定ディスク (20MB) の場合、307 トラックのサーフェス 4~7 は使用不可。
- 従来インタフェースの固定ディスク (40MB) の場合、601 トラック~613 トラックは使用不可。
- \* は120MBの領域を確保した場合。

## ■クラスタ

ディスクの読み書きの最小単位は1セクタですが、使用・未使用についてはクラスタが最小単位になります。



クラスタは具体的には複数のセクタが集まったものです。クラスタの大きさ(クラスタ中のセクタ数)はディスクの種類によって異なります。

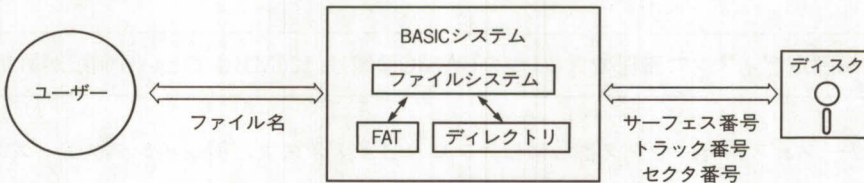
ディスクのクラスタの大きさと総数は前の表のようになっています。

## ■ディスクの読み書き

BASICにはディスクを読み書きする方法が2とおり用意されています。

### ●ファイル単位に読み書きする方法

通常、BASICで行うディスクの読み書きはファイル単位に行われます。このときにはユーザーはファイル名を指定すればよく、ディスクのアドレスを意識する必要はありません。これは、ユーザーがファイルを指定すると、BASICのファイルシステムがディレクトリやFATなどのファイルの管理情報を調べて、これをもとにそのファイルのディスク内でのアドレス(サーフェス番号,トラック番号,セクタ番号)を計算し、ディスクへのアクセスを行うからです。このとき、ディスクへの読み書きの最小単位はセクタですが、管理はクラスタ単位で行われます。



### ●ディスクのアドレスを直接読み書きする方法

DSKO\$命令, DSKI\$関数を使って行います。

#### フロッピーディスクおよび従来インタフェースの固定ディスクの場合

DSKO\$命令, DSKI\$関数を使って、ドライブ番号,サーフェス番号,トラック番号,セクタ番号でディスクのアドレス(物理的な位置)を指定することにより、ディスクに直接に読み書きします。

#### SCSI固定ディスクの場合

SCSI固定ディスクでは、ディスク内部を複数のBASIC領域に分割し、同時に最大4つのBASIC領域を使用することができます。それぞれの領域が別の独立したドライブとして割り当てられます。

このため、DSKO\$命令, DSKI\$関数においてディスクの直接読み書き時に指定するトラック番号は、各ドライブ内での相対的なトラック番号となります。

たとえば、20MBのSCSI固定ディスクに100トラックから始まる5MBの領域を確保すると、物理的な100~204まで(トラック数は105)のトラックが1つのドライブとして認識されるようになります。この場合、そのドライブには0~104番(合計105個)までの相対的なトラック番号が割り振られることになります。

この相対トラック数はDSKF関数を使って得ることができます。

$$\text{相対トラック数} = \text{DSKF}(n, 0) + 1$$

(nはドライブ番号を表します)

また、指定できるトラック番号の大きさは、固定ディスクの種類や各ドライブに割り当てたトラック数によって異なりますが、最大の大きさは次のようになります。

| SCSI固定ディスクの種類 | サーフェス当りのトラック数の最大値 |
|---------------|-------------------|
| 20MB          | 420               |
| 40MB          | 429               |
| 130MB         | 720               |

---

注意：SCSI固定ディスクに確保できる1つのBASIC領域は、120MBまでという制限があります。

---

なお、サーフェス番号、セクタ番号には、フロッピーディスク、従来インタフェースの固定ディスクの場合と同様にディスク装置のサーフェス番号、セクタ番号を指定します。

### ■クラスタ番号とディスクのアドレスとの関係

クラスタには固有の番号がつけられており、クラスタ0、クラスタ1、クラスタ2、……と番号が振られています。ディスクのアドレス(物理的な位置)との対応関係は次の式で知ることができます。ディスクの領域を物理的に読み書きする際の参考にしてください。

$$\text{クラスタ番号} = \frac{\text{トラック番号} \times \text{サーフェス数} + \text{サーフェス番号}}{\text{クラスタ当りのトラック数}}$$

(小数点以下は切り捨て)

なお、SCSI固定ディスクの場合、クラスタ番号とトラック番号はドライブ内での相対的な番号です。

## ■管理テーブル

ファイルの管理情報が書き込まれるディスクの領域を管理テーブルといいます。管理テーブルの中には、DIR(ディレクトリ)、FAT(ファイル領域管理テーブル)、IDの3つのテーブルが作られます。

## ●ディレクトリ

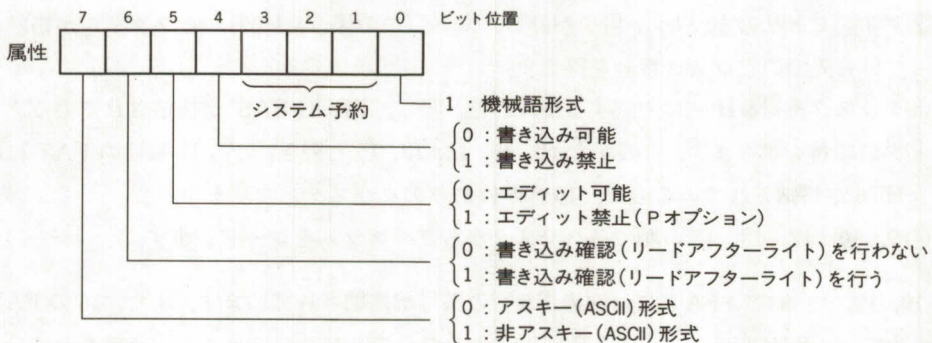
ディスク上のファイルを管理するための表です。格納されているすべてのファイルに対してファイルの名前、属性、格納されている場所(そのファイルが格納されている先頭のクラスタ番号)が16バイトの制御情報として記録されます。

なお、ファイルが格納されている先頭のクラスタ番号を示すエリアのバイト数は、固定ディスクでは2バイト、フロッピーディスクでは1バイトとなっています。

### ディレクトリの情報

| 内 容                     | フロッピーディスク    |                | 固定ディスク       |                |
|-------------------------|--------------|----------------|--------------|----------------|
|                         | 位 置<br>(バイト) | 大 き さ<br>(バイト) | 位 置<br>(バイト) | 大 き さ<br>(バイト) |
| ファイル名                   | 0～5          | 6              | 0～5          | 6              |
| 拡 張 子                   | 6～8          | 3              | 6～8          | 3              |
| 属 性                     | 9            | 1              | 9            | 1              |
| そのファイルが格納されている先頭のクラスタ番号 | 10           | 1              | 10～11        | 2              |
| システム予約                  | 11～15        | 5              | 12           | 1              |
| ユーザー識別名                 |              |                | 13～15        | 3              |

属性を表す1バイトは次のような意味を持っています。



ファイル名の先頭バイトの内容が&HFF のとき、そのディレクトリは未使用であることを示します。また、&H00 のときは削除(KILL)されたファイルであることを示します。

ディスクの種類により扱えるファイル数は異なります。各ディスクで扱えるファイルの数はディレクトリの大きさとクラスタの大きさにより決まります。

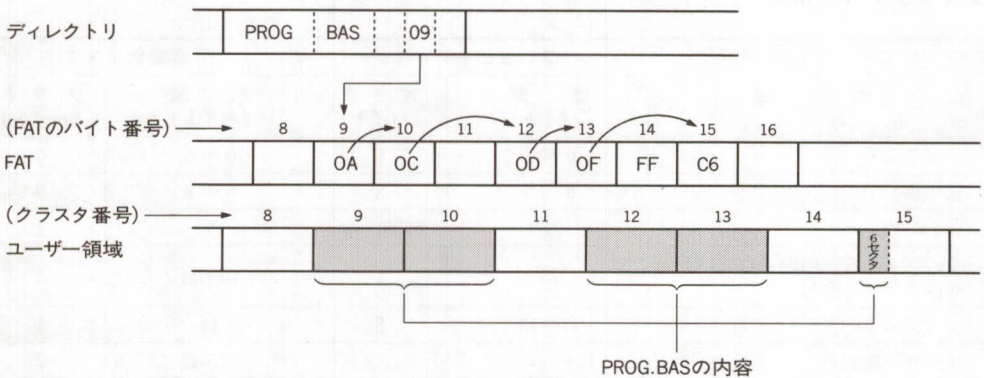
● FAT

FAT(ファイル領域管理テーブル)は、ファイルが使用するディスク領域を管理するための表です。FATにはクラスタごとに1バイト(フロッピーディスクの場合)、または2バイト(固定ディスクの場合)の領域(エントリ)が用意されており、そのクラスタが使用されているか、使用されていないかの状態を示しています。このFATの情報を見るとそれぞれのファイルがどのクラスタにセーブされているかを知ることができます。

BASICでファイルの読み書きを行った場合、BASICのファイルシステムがどのようにしてディスクへのアクセスを行うかをディレクトリとFATの関係で図解します。

ディレクトリ, および FAT 利用の一例

例) LOAD "PROG.BAS"を実行した場合



- ① BASICはディレクトリに“PROG.BAS”があるかどうかを捜します。
- ②ディレクトリの10バイト目を参照し、この値(この場合&H09)からファイルが格納されている先頭のクラスタ番号を得ます。
- ③クラスタ番号&H09に対応するFATを参照し、“PROG.BAS”が格納されているクラスタの順番を調べます。この場合、9、10、12、13、15と続き、この15番目のFATには&HC6が格納されているので、15番が終わりのクラスタとなります。
- ④9、10、12、13、15の順で各クラスタからプログラムをロードします。

9、10、12、13までのFATには次のFATの番号が格納されています。また、15のFATに格納されている&HC6は、このFATに対応するクラスタがこのファイルが格納されている最後のクラスタである、ということと、最後のクラスタの中で何セクタまでを使用しているかを示しています。

最後のクラスタで使用しているセクタ数は次の式で求められます。

### 最後の FAT に格納されている値—&HC0

この場合は最後のクラスタは6セクタまで使われていることになります。

なお、&HC0 というのはフロッピディスクの場合の値です。固定ディスクの場合は&H4000を使います。

そのほかにシステムで予約済みのクラスタに対応する FAT には&HFE が格納され、未使用のクラスタに対応する FAT には&HFF が格納されます。

FAT の値(16進表記)とそれに対応するクラスタの状態は次の表のようになっています。

| FATの値                      |                                | クラスタの使用状況                                      |
|----------------------------|--------------------------------|--|
| フロッピディスク                   | 固定ディスク                         |  |
| 0<br>} (クラスタ総数-1)          | 0<br>} (クラスタ総数-1)              | そのクラスタの全セクタを使用している。<br>FATの値は後に続くクラスタの番号を表す。   |
| &HC0<br>} クラスタ当りのセクタ数+&HC0 | &H4000<br>} クラスタ当りのセクタ数+&H4000 | あるファイルの最後のクラスタ。<br>FATの値はそのクラスタの使用しているセクタ数を表す。 |
| &HFE                       | &HFFFE                         | BASICで予約済みのクラスタ。                               |
| —                          | &HFFFD                         | 使用不可のセクタを含むクラスタ。                               |
| &HFF                       | &HFFFF                         | 未使用クラスタ。                                       |

BASICで予約済みのクラスタには、DISK CODE, IPL, ディレクトリ, FAT, IDなどが含まれます。

なお、フロッピディスクの場合、安全のために FAT の内容を3箇所に記録しています。

FAT は、その装置に対する最初のアクセスでメモリ上に読み込まれ、FAT を更新する際に、必要に応じてディスクへ書きもどされます。

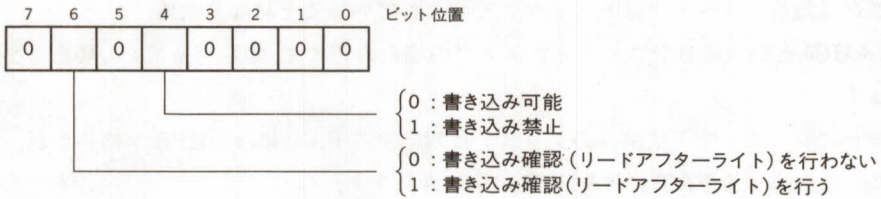
### ● ID

ディスクには、ディスクの種類を識別するために1セクタ分の領域が確保されています。この領域を ID とよびます。ID は次のような形式をしています。

|      |       |           |
|------|-------|-----------|
| 属性   | ファイル数 | BASICテキスト |
| 1バイト | 1バイト  | 254バイト    |

### 属性

属性はディスクの属性を示します。



属性は SET 命令で設定されます。

▶参照 SET 命令→6.6の「ファイル属性の設定」

### ファイル数(システムディスク用)

同時にオープンされるファイル数を設定するエリアです。

この<ファイル数>に0~15の数値を設定すると、BASICを起動したとき、`How many files (0-15)?`が表示されずに自動的にそのファイル数が設定されます。

初期状態のシステムディスクでは255(&HFF)が設定されており、BASIC起動時に同時にオープンするファイル数を聞いてきます。

### BASIC テキスト(システムディスク用)

オートスタート時に自動的に最初に行われる BASIC の命令を書き込んでおくエリアです。オートスタート時にはこの命令が実行されます。

### ID への書き込み方

ID への書き込みにはユーティリティ “setinf.n88” を使います。

▶参照 setinf.n88→「10.5 IDセクタの書き換え」

## ■トラックの割り当て

ディスク内のトラックの割り当ては次のようになっています。

### ●フロッピーディスクの場合

#### 3.5 インチ, 5 インチ 2DD(両面倍密度倍トラック)

##### システムディスク

| サーフェス番号 | トラック番号 | セクタ番号  | 内 容         |
|---------|--------|--------|-------------|
| 0       | 0      | 1～2    | IPL         |
|         | 0      | 3～16   | } DISK CODE |
|         | 1～35   | すべて    |             |
|         | 36～39  | すべて    | ユーザー領域      |
|         | 40     | 1～12   | ディレクトリ      |
|         | 40     | 13     | ID          |
|         | 40     | 14～16  | FAT         |
| 41～79   | すべて    | ユーザー領域 |             |
| 1       | 0～35   | すべて    | DISK CODE   |
|         | 36～79  | すべて    | ユーザー領域      |

##### ユーザーディスク

| サーフェス番号 | トラック番号 | セクタ番号 | 内 容    |
|---------|--------|-------|--------|
| 0       | 0～39   | すべて   | ユーザー領域 |
|         | 40     | 1～12  | ディレクトリ |
|         | 40     | 13    | ID     |
|         | 40     | 14～16 | FAT    |
|         | 41～79  | すべて   | ユーザー領域 |
| 1       | 0～79   | すべて   | ユーザー領域 |

3.5 インチ, 5 インチ 2HD(高密度倍トラック)および8 インチ 2D(両面倍密度)

システムディスク

| サーフェス番号 | トラック番号 | セクタ番号 | 内 容       |
|---------|--------|-------|-----------|
| 0       | 0*     | 1~4   | IPL       |
|         | 0*     | 5~26  | システム予約    |
|         | 1~23   | すべて   | DISK CODE |
|         | 24~34  | すべて   | ユーザー領域    |
|         | 35     | 1~22  | ディレクトリ    |
|         | 35     | 23    | ID        |
|         | 35     | 24~26 | FAT       |
|         | 36~76  | すべて   | ユーザー領域    |
| 1       | 0      | すべて   | システム予約    |
|         | 1~22   | すべて   | DISK CODE |
|         | 23~76  | すべて   | ユーザー領域    |

\*トラック番号0のトラックは1セクタ128バイトでフォーマットされている。  
他のトラックのセクタはすべて256バイトでフォーマットされている。

ユーザーディスク

| サーフェス番号 | トラック番号 | セクタ番号 | 内 容    |
|---------|--------|-------|--------|
| 0       | 0*     | 1~26  | システム予約 |
|         | 1~34   | すべて   | ユーザー領域 |
|         | 35     | 1~22  | ディレクトリ |
|         | 35     | 23    | ID     |
|         | 35     | 24~26 | FAT    |
|         | 36~76  | すべて   | ユーザー領域 |
| 1       | 0      | すべて   | システム予約 |
|         | 1~76   | すべて   | ユーザー領域 |

\*トラック番号0のトラックは1セクタ128バイトでフォーマットされている。  
他のトラックのセクタはすべて256バイトでフォーマットされている。

## ●固定ディスク(標準フォーマット)の場合

以下の表は媒体のすべてが BASIC で使用され、かつ不良トラックがない場合を示しています。したがって、IPL, DISK CODE およびシステム予約をのぞく部分はディスクフォーマット時のサイズ指定によりトラックの位置が異なってきます。

## 固定ディスク(5MB)

## システムディスク

| サーフェス番号 | トラック番号 | セクタ番号 | 内 容  |
|---------|--------|-------|--|
| 0       | 0      | すべて   | IPL およびシステム予約<br>DISK CODE<br>ユーザー領域<br>ディレクトリ<br>ユーザー領域<br>システム予約 |
|         | 1~8    | すべて   |  |
|         | 9~75   | すべて   |  |
|         | 76     | すべて   |  |
|         | 77~151 | すべて   |  |
|         | 152    | すべて   |  |
| 1~2     | 0~8    | すべて   | DISK CODE<br>ユーザー領域<br>ディレクトリ<br>ユーザー領域<br>システム予約                  |
|         | 9~75   | すべて   |  |
|         | 76     | すべて   |  |
|         | 77~151 | すべて   |  |
|         | 152    | すべて   |  |
|         | 3      | 0~8   |  |
| 9~75    | すべて    |       |  |
| 76      | 1      |       |  |
| 76      | 2~33   |       |  |
| 77~151  | すべて    |       |  |
| 152     | すべて    |       |  |

## ユーザーディスク

| サーフェス番号 | トラック番号 | セクタ番号 | 内 容  |
|---------|--------|-------|--|
| 0       | 0      | すべて   | システム予約<br>ユーザー領域<br>ディレクトリ<br>ユーザー領域<br>システム予約 |
|         | 1~75   | すべて   |  |
|         | 76     | すべて   |  |
|         | 77~151 | すべて   |  |
|         | 152    | すべて   |  |
|         | 1~2    | 0~75  |  |
| 76      | すべて    |       |  |
| 77~151  | すべて    |       |  |
| 152     | すべて    |       |  |
| 3       | 0~75   | すべて   | ユーザー領域<br>ID<br>FAT<br>ユーザー領域<br>システム予約        |
|         | 76     | 1     |  |
|         | 76     | 2~33  |  |
|         | 77~151 | すべて   |  |
|         | 152    | すべて   |  |
|         | 152    | すべて   |  |

固定ディスク(10MB)

システムディスク

| サーフェス番号 | トラック番号  | セクタ番号 | 内 容           |
|---------|---------|-------|---------------|
| 0       | 0       | すべて   | IPL およびシステム予約 |
|         | 1~8     | すべて   | DISK CODE     |
|         | 9~154   | すべて   | ユーザー領域        |
|         | 155     | すべて   | ディレクトリ        |
|         | 156~308 | すべて   | ユーザー領域        |
|         | 309     | すべて   | システム予約        |
| 1~2     | 0~8     | すべて   | DISK CODE     |
|         | 9~154   | すべて   | ユーザー領域        |
|         | 155     | すべて   | ディレクトリ        |
|         | 156~308 | すべて   | ユーザー領域        |
|         | 309     | すべて   | システム予約        |
| 3       | 0~8     | すべて   | DISK CODE     |
|         | 9~154   | すべて   | ユーザー領域        |
|         | 155     | 1     | ID            |
|         | 155     | 2~33  | FAT           |
|         | 156~308 | すべて   | ユーザー領域        |
|         | 309     | すべて   | システム予約        |

ユーザーディスク

| サーフェス番号 | トラック番号  | セクタ番号 | 内 容    |
|---------|---------|-------|--------|
| 0       | 0       | すべて   | システム予約 |
|         | 1~154   | すべて   | ユーザー領域 |
|         | 155     | すべて   | ディレクトリ |
|         | 156~308 | すべて   | ユーザー領域 |
|         | 309     | すべて   | システム予約 |
| 1~2     | 0~154   | すべて   | ユーザー領域 |
|         | 155     | すべて   | ディレクトリ |
|         | 156~308 | すべて   | ユーザー領域 |
|         | 309     | すべて   | システム予約 |
| 3       | 0~154   | すべて   | ユーザー領域 |
|         | 155     | 1     | ID     |
|         | 155     | 2~33  | FAT    |
|         | 156~308 | すべて   | ユーザー領域 |
|         | 309     | すべて   | システム予約 |

## 固定ディスク (20MB)

## システムディスク

| サーフェス番号 | トラック番号  | セクタ番号 | 内 容           |
|---------|---------|-------|---------------|
| 0       | 0       | すべて   | IPL およびシステム予約 |
|         | 1~4     | すべて   | DISK CODE     |
|         | 5~154   | すべて   | ユーザー領域        |
|         | 155     | すべて   | ディレクトリ        |
|         | 156~306 | すべて   | ユーザー領域        |
|         | 307     | すべて   | システム予約        |
| 1~3     | 0~4     | すべて   | DISK CODE     |
|         | 5~154   | すべて   | ユーザー領域        |
|         | 155     | すべて   | ディレクトリ        |
|         | 156~306 | すべて   | ユーザー領域        |
|         | 307     | すべて   | システム予約        |
| 4~6     | 0~3     | すべて   | DISK CODE     |
|         | 4~154   | すべて   | ユーザー領域        |
|         | 155     | すべて   | ディレクトリ        |
|         | 156~306 | すべて   | ユーザー領域        |
| 7       | 0~3     | すべて   | DISK CODE     |
|         | 4~154   | すべて   | ユーザー領域        |
|         | 155     | 1     | ID            |
|         | 155     | 2~33  | FAT           |
|         | 156~306 | すべて   | ユーザー領域        |

## ユーザーディスク

| サーフェス番号 | トラック番号  | セクタ番号 | 内 容    |
|---------|---------|-------|--------|
| 0       | 0       | すべて   | システム予約 |
|         | 1~154   | すべて   | ユーザー領域 |
|         | 155     | すべて   | ディレクトリ |
|         | 156~306 | すべて   | ユーザー領域 |
|         | 307     | すべて   | システム予約 |
| 1~3     | 0~154   | すべて   | ユーザー領域 |
|         | 155     | すべて   | ディレクトリ |
|         | 156~306 | すべて   | ユーザー領域 |
|         | 307     | すべて   | システム予約 |
| 4~6     | 0~154   | すべて   | ユーザー領域 |
|         | 155     | すべて   | ディレクトリ |
|         | 156~306 | すべて   | ユーザー領域 |
| 7       | 0~154   | すべて   | ユーザー領域 |
|         | 155     | 1     | ID     |
|         | 155     | 2~33  | FAT    |
|         | 156~306 | すべて   | ユーザー領域 |

●固定ディスク(拡張フォーマット)の場合

拡張フォーマットの場合、他のオペレーティングシステムとの領域の取り方により、BASIC内のディレクトリ、ID、FAT やユーザー領域の位置が大きく変化します。固定ディスクに対して DSKI \$ 関数や DSKO \$ 命令を用いて入出力操作を行う場合、各部分(たとえばディレクトリなど)がどの位置にあるかは DSKF 関数を用いて調べる必要があります。

| 内 容    | サーフェス番号       | トラック番号    | セクタ番号     |
|--------|---------------|-----------|-----------|
| ディレクトリ | 0~DSKF(n,2)-1 | DSKF(n,5) | すべて       |
| ID     | DSKF(n,2)     | DSKF(n,5) | 1         |
| FAT    | DSKF(n,2)     | DSKF(n,5) | 2~最大セクタ番号 |

(表中の n はドライブ番号)

SCSI固定ディスクの場合に得られるトラック番号は、各ドライブの相対的なトラック番号です。

# 付録 D

## キーセンス

BASIC の INP 関数を使用し、キーボードの各キーが押されているかどうかを知ることができます。これをキーセンスといいます。

キーセンスのためのポートアドレスは E0~EC で、ポートアドレスとキーの対応関係は次のとおりです。

```
例) 10 PRINT HEX$(INP(&HE3))
     20 GOTO 10
```

これを実行して「FF」が表示されたら H~O のキーは押されていません。もし「FE」が表示されていれば H が、「F7」が表示されていれば K が押されていることとなります。

(FF)<sub>16</sub> = (11111111)<sub>2</sub> ..... どのキーも押されていない  
(FE)<sub>16</sub> = (11111110)<sub>2</sub> ..... H のキーが押されている  
(FD)<sub>16</sub> = (11111101)<sub>2</sub> ..... I のキーが押されている  
(FB)<sub>16</sub> = (11111011)<sub>2</sub> ..... J のキーが押されている  
(F7)<sub>16</sub> = (11110111)<sub>2</sub> ..... K のキーが押されている  
(EF)<sub>16</sub> = (11101111)<sub>2</sub> ..... L のキーが押されている  
(DF)<sub>16</sub> = (11011111)<sub>2</sub> ..... M のキーが押されている  
(BF)<sub>16</sub> = (10111111)<sub>2</sub> ..... N のキーが押されている  
(7F)<sub>16</sub> = (01111111)<sub>2</sub> ..... O のキーが押されている

2 個のキーが同時に押されているときは、上記以外の値になります。たとえば I と M が押されていれば、(11011101)<sub>2</sub> = (DD)<sub>16</sub> が得られます。

| ポートアドレス | D7     | D6     | D5     | D4     | D3     | D2     | D1           | D0          |
|---------|--------|--------|--------|--------|--------|--------|--------------|-------------|
| E0      | 7      | 6      | 5      | 4      | 3      | 2      | 1            | 0           |
| E1      | ↵      | ・      | 、      | =      | +      | *      | 9            | 8           |
| E2      | Gキ     | Fハ     | Eイ     | Dシ     | Cソ     | Bコ     | Aチ           | @、          |
| E3      | Oラ     | Nミ     | Mモ     | Lリ     | Kノ     | Jマ     | Iニ           | Hク          |
| E4      | Wテ     | Vヒ     | Uナ     | Tカ     | Sト     | Rス     | Qタ           | Pセ          |
| E5      | =<br>ホ | ^<br>へ | }<br>ム | ¥<br>一 | [<br>レ | Z<br>ツ | Y<br>ン       | X<br>サ      |
| E6      | 7<br>ヤ | 6<br>オ | 5<br>エ | 4<br>ウ | 3<br>ア | 2<br>フ | 1<br>ヌ       | 0<br>ヲ      |
| E7      | -<br>ロ | ?<br>メ | ><br>ル | <<br>ネ | +<br>レ | *<br>ケ | 9<br>ヨ       | 8<br>ユ      |
| E8      | CTRL   | SHIFT  | カナ     | GRPH   | DEL    | →      | ↑            | HOME<br>CLR |
| E9      | ESC    | SPACE  | f・5    | f・4    | f・3    | f・2    | f・1          | STOP        |
| EA      | CAPS   | /      | -      | COPY   | HELP   | ←      | ↓            | TAB         |
| EB      |        |        |        |        |        |        | ROLL<br>DOWN | ROLL<br>UP  |
| EC      | INS    | f・10   | f・9    | f・8    | f・7    | f・6    | XFER         | BS          |

# 付録 E

## メモリスイッチ

PC-9800 シリーズの本体には不揮発性メモリが用意されており、本体の動作環境を規定しています。不揮発性メモリのそれぞれのビット状態「1」または「0」は、「スイッチ」の「ON」「OFF」の状態と考えられることから、「メモリスイッチ」と呼びます。

不揮発性メモリは電源が断たれても、メモリの状態を保持します。ただし、電源を入れない状態で、2ヶ月間以上動作させないと不定になります。

**注意：**次の場合には必ずディップスイッチを工場出荷時の状態にして電源投入状態を15時間以上続けてください。

- ・長期間(2ヶ月間以上)電源を入れなかった場合。
- ・本体を購入して初めて使用する場合(この場合、長時間は不要)。

不揮発性メモリのバックアップ用電池の充電が行われ、メモリスイッチの状態はクリアされ、システム既定値に設定されます。

### ●メモリスイッチのセット

#### システム既定値で使用する場合

- ①ディップスイッチ SW2 の 5 番スイッチを OFF にします。
- ②リセットスイッチを押してシステムを再起動します。

#### システム既定値以外で使用する場合

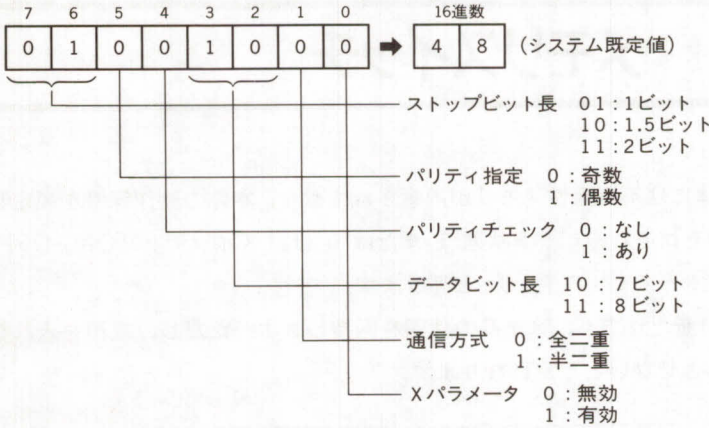
- ①ディップスイッチ SW2 の 5 番スイッチを ON にします。
- ②メモリスイッチを設定します。
- ③リセットスイッチを押してシステムを再起動します。

メモリスイッチの設定は、専用のユーティリティ “switch.n88” を使用して行います。

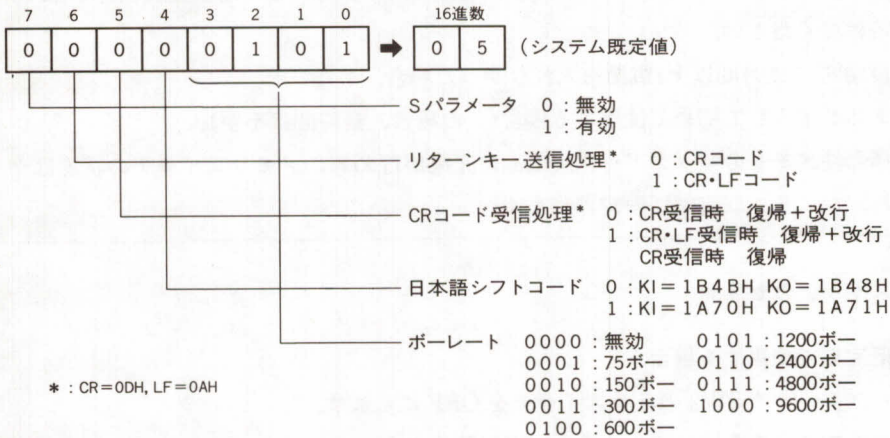
▶参照 switch.n88 →「10.14 メモリスイッチの設定」

●メモリスイッチの値と機能

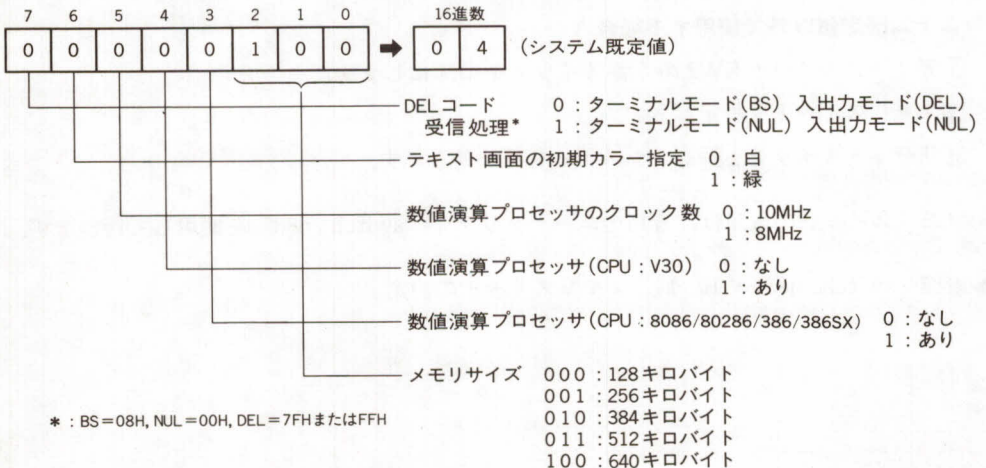
SW1(アドレスA3FE2H)



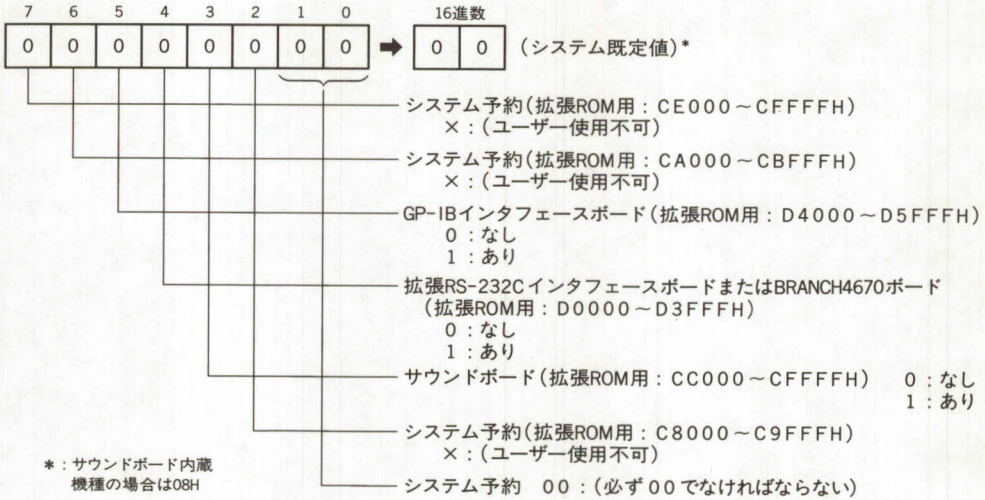
SW2(アドレスA3FE6H)



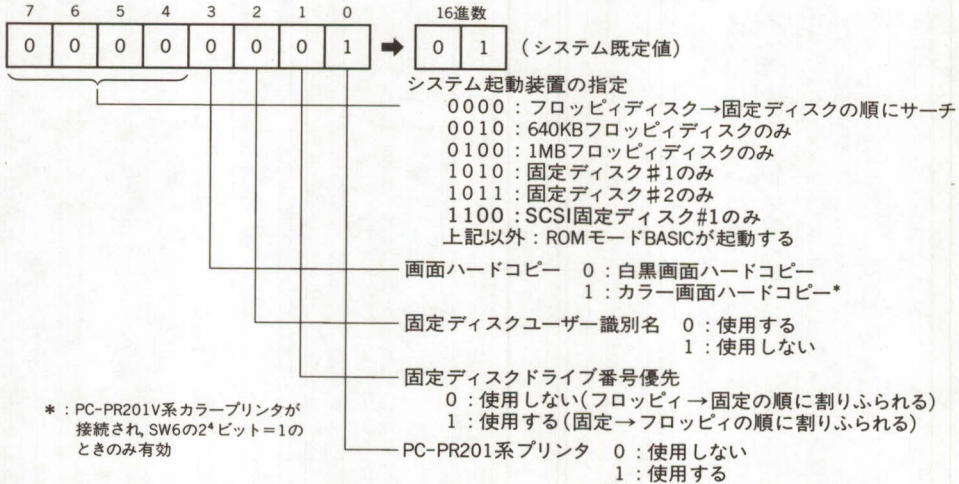
SW3(アドレスA3FEAH)



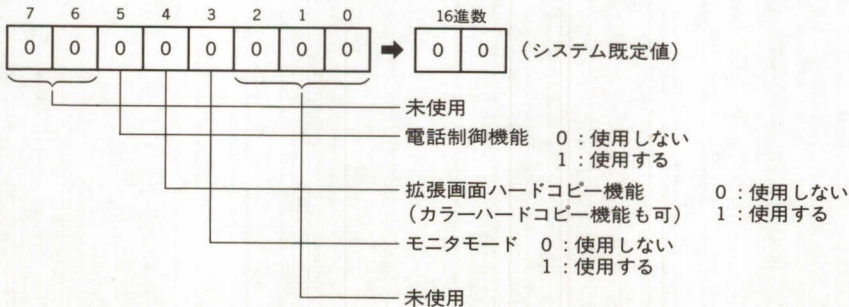
SW4(アドレスA3FEEH)



SW5(アドレスA3FF2H)



SW6(アドレスA3FF6H)





# 付録 F

---

## 数値演算プロセッサの使用

---

数値演算プロセッサは浮動小数点データによる演算を高速化するための付加プロセッサです。数値演算プロセッサがコンピュータ本体に実装されている場合、所定のメモリスイッチを設定することにより、BASICの数学関数の実行速度を速めることができます。

次に示す数学関数あるいは演算処理が高速化されます。

SIN, COS, TAN, ATN, EXP, SQR, べき乗演算

四則演算は高速化の対象となりません。また、数値演算プロセッサを適用した場合と適用しない場合とでは演算精度に違いが出ます。

### メモリスイッチの設定

V30用数値演算プロセッサと、8086/80286/386/386SX用数値演算プロセッサのどちらを実装したかによってメモリスイッチの設定が異なります。ユーティリティ“switch.n88”を使用してメモリスイッチの設定を行ってください。

▶参照 switch.n88 →「10.14 メモリスイッチの設定」

---

注意：数値プロセッサは使用できない機種もあります。

---



# 付録 G

## 日本語コード表

JIS第一水準および第二水準漢字コードは『日本語入力ガイド』に掲載されています。

●文字列として扱える(テキスト画面に表示できる)2バイト系半角文字一覧表

|          | 0020 | 0030 | 0040 | 0050 | 0060 | 0070 | 00A0 | 00B0 | 00C0 | 00D0 |
|----------|------|------|------|------|------|------|------|------|------|------|
| <b>0</b> | SP   | 0    | @    | P    |      | p    | SP   | 一    | タ    | ミ    |
| <b>1</b> | !    | 1    | A    | Q    | a    | q    | 。    | ア    | チ    | ム    |
| <b>2</b> | "    | 2    | B    | R    | b    | r    | 「    | イ    | ツ    | メ    |
| <b>3</b> | #    | 3    | C    | S    | c    | s    | 」    | ウ    | テ    | モ    |
| <b>4</b> | \$   | 4    | D    | T    | d    | t    | 、    | エ    | ト    | ヤ    |
| <b>5</b> | %    | 5    | E    | U    | e    | u    | ・    | オ    | ナ    | ユ    |
| <b>6</b> | &    | 6    | F    | V    | f    | v    | ヲ    | カ    | ニ    | ヨ    |
| <b>7</b> | '    | 7    | G    | W    | g    | w    | ア    | キ    | ヌ    | ラ    |
| <b>8</b> | (    | 8    | H    | X    | h    | x    | イ    | ク    | ネ    | リ    |
| <b>9</b> | )    | 9    | I    | Y    | i    | y    | ウ    | ケ    | ノ    | ル    |
| <b>A</b> | *    | :    | J    | Z    | j    | z    | エ    | コ    | ハ    | レ    |
| <b>B</b> | +    | ;    | K    | [    | k    | {    | オ    | サ    | ヒ    | ロ    |
| <b>C</b> | ,    | <    | L    | ¥    | l    |      | ヤ    | シ    | フ    | ワ    |
| <b>D</b> | -    | =    | M    | ]    | m    | }    | ユ    | ス    | ヘ    | ン    |
| <b>E</b> | .    | >    | N    | ^    | n    | ~    | ヨ    | セ    | ホ    | 〃    |
| <b>F</b> | /    | ?    | O    | _    | o    |      | ツ    | ソ    | マ    | °    |

(注) SP は空白(スペース)コードを示します。

なお、これらの半角文字はプリンタで印字することはできません。

●グラフィック画面に表示できる2バイト系半角文字一覧表

|   | 0020 | 0030 | 0040 | 0050 | 0060 | 0070 | 0080 | 0090 | 00A0 | 00B0 | 00C0 | 00D0 | 00E0 | 00F0 |
|---|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 0 | SP   | 0    | @    | P    |      | p    | SP   | 一    | SP   | 一    | タ    | ミ    | た    | み    |
| 1 | !    | 1    | A    | Q    | a    | q    | 。    | あ    | 。    | ア    | チ    | ム    | ち    | む    |
| 2 | "    | 2    | B    | R    | b    | r    | 「    | い    | 「    | イ    | ツ    | メ    | つ    | め    |
| 3 | #    | 3    | C    | S    | c    | s    | 」    | う    | 」    | ウ    | テ    | モ    | て    | も    |
| 4 | \$   | 4    | D    | T    | d    | t    | 、    | え    | 、    | エ    | ト    | ヤ    | と    | や    |
| 5 | %    | 5    | E    | U    | e    | u    | ・    | お    | ・    | オ    | ナ    | ユ    | な    | ゆ    |
| 6 | &    | 6    | F    | V    | f    | v    | を    | か    | ヲ    | カ    | ニ    | ヨ    | に    | よ    |
| 7 | '    | 7    | G    | W    | g    | w    | あ    | き    | ア    | キ    | ヌ    | ラ    | ぬ    | ら    |
| 8 | (    | 8    | H    | X    | h    | x    | い    | く    | イ    | ク    | ネ    | リ    | ね    | り    |
| 9 | )    | 9    | I    | Y    | i    | y    | う    | け    | ウ    | ケ    | ノ    | ル    | の    | る    |
| A | *    | :    | J    | Z    | j    | z    | え    | こ    | エ    | コ    | ハ    | レ    | は    | れ    |
| B | +    | ;    | K    | [    | k    | {    | お    | さ    | オ    | サ    | ヒ    | ロ    | ひ    | ろ    |
| C | ,    | <    | L    | ¥    | l    |      | や    | し    | ヤ    | シ    | フ    | ワ    | ふ    | わ    |
| D | -    | =    | M    | ]    | m    | }    | ゆ    | す    | ユ    | ス    | ヘ    | ン    | へ    | ん    |
| E | .    | >    | N    | ^    | n    | ~    | よ    | せ    | ヨ    | セ    | ホ    | ”    | ほ    | ”    |
| F | /    | ?    | O    | _    | o    |      | っ    | そ    | ツ    | ソ    | マ    | °    | ま    | °    |

(注) SPは空白(スペース)コードを示します。  
これらの文字を画面に出力するにはPUT@命令を使用します。

●グラフィック画面に表示できる2バイト系1/4角文字一覧表

|          | 0100           | 0110           | 0120 | 0130 | 0140 | 0150 | 0160 | 0170 | 0180 | 0190 | 01A0 | 01B0 | 01C0 | 01D0 | 01E0 | 01F0 |
|----------|----------------|----------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| <b>0</b> |                | <sup>D</sup> E | [SP] | 0    | @    | P    |      | p    |      |      | [SP] | 一    | タ    | ミ    | ≡    | ×    |
| <b>1</b> | <sup>S</sup> H | <sup>D</sup> 1 | !    | 1    | A    | Q    | a    | q    |      |      | 。    | ア    | チ    | ム    | ≡    | 円    |
| <b>2</b> | <sup>S</sup> X | <sup>D</sup> 2 | "    | 2    | B    | R    | b    | r    |      |      | 「    | イ    | ツ    | メ    | ≡    | 年    |
| <b>3</b> | <sup>E</sup> X | <sup>D</sup> 3 | #    | 3    | C    | S    | c    | s    |      |      | 」    | ウ    | テ    | モ    | ≡    | 月    |
| <b>4</b> | <sup>E</sup> T | <sup>D</sup> 4 | \$   | 4    | D    | T    | d    | t    |      |      | 、    | エ    | ト    | ヤ    | ▲    | 日    |
| <b>5</b> | <sup>E</sup> Q | <sup>N</sup> K | %    | 5    | E    | U    | e    | u    |      |      | ・    | オ    | ナ    | ユ    | ▲    | 時    |
| <b>6</b> | <sup>A</sup> K | <sup>S</sup> N | &    | 6    | F    | V    | f    | v    |      |      | ヲ    | カ    | ニ    | ヨ    | ▲    | 分    |
| <b>7</b> | <sup>B</sup> L | <sup>E</sup> B | '    | 7    | G    | W    | g    | w    |      |      | ア    | キ    | ヌ    | ラ    | ▲    | 秒    |
| <b>8</b> | <sup>B</sup> S | <sup>C</sup> N | (    | 8    | H    | X    | h    | x    |      |      | イ    | ク    | ネ    | リ    | ♠    |      |
| <b>9</b> | <sup>H</sup> T | <sup>E</sup> M | )    | 9    | I    | Y    | i    | y    |      |      | ウ    | ケ    | ノ    | ル    | ♥    |      |
| <b>A</b> | <sup>L</sup> F | <sup>S</sup> B | *    | :    | J    | Z    | j    | z    |      |      | エ    | コ    | ハ    | レ    | ♦    |      |
| <b>B</b> | <sup>H</sup> M | <sup>E</sup> C | +    | ;    | K    | [    | k    | {    |      |      | オ    | サ    | ヒ    | ロ    | ♣    |      |
| <b>C</b> | <sup>C</sup> L | →              | ,    | <    | L    | ¥    | l    |      |      |      | ヤ    | シ    | フ    | ワ    | ●    |      |
| <b>D</b> | <sup>C</sup> R | ←              | -    | =    | M    | ]    | m    | }    |      |      | ユ    | ス    | ヘ    | ン    | ○    |      |
| <b>E</b> | <sup>S</sup> O | ↑              | .    | >    | N    | ^    | n    | ~    |      |      | ヨ    | セ    | ホ    | ”    | △    |      |
| <b>F</b> | <sup>S</sup> I | ↓              | /    | ?    | O    | _    | o    |      |      |      | ツ    | ソ    | マ    | °    | ▽    |      |

(注) [SP] は空白(スペース)コードを示します。  
 これらの文字を画面に出力するには PUT@命令を使用します。



# 付録 H

## キャラクタコード表

上位4ビット→







|         | 0 | 1   | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A  | B | C | D | E | F |
|---------|---|---|----|---|---|---|---|---|---|---|----|---|---|---|---|---|
| 下位4ビット↓ | 0 | <sup>D<sub>E</sub></sup>                          | SP | 0 | @ | P |   | p |   |   | SP | 一 | タ | ミ |   | × |
|         | 1 | <sup>S<sub>H</sub></sup> <sub>D<sub>1</sub></sub> | !  | 1 | A | Q |   | a | q |   |    | 。 | ア | チ | ム | 円 |
|         | 2 | <sup>S<sub>X</sub></sup> <sub>D<sub>2</sub></sub> | "  | 2 | B | R |   | b | r |   |    | 「 | イ | ツ | メ | 年 |
|         | 3 | <sup>E<sub>X</sub></sup> <sub>D<sub>3</sub></sub> | #  | 3 | C | S |   | c | s |   |    | 」 | ウ | テ | モ | 月 |
|         | 4 | <sup>E<sub>T</sub></sup> <sub>D<sub>4</sub></sub> | \$ | 4 | D | T |   | d | t |   |    | 、 | エ | ト | ヤ | 日 |
|         | 5 | <sup>E<sub>Q</sub></sup> <sub>N<sub>K</sub></sub> | %  | 5 | E | U |   | e | u |   |    | ・ | オ | ナ | ユ | 時 |
|         | 6 | <sup>A<sub>K</sub></sup> <sub>S<sub>N</sub></sub> | &  | 6 | F | V |   | f | v |   |    | ヲ | カ | ニ | ヨ | 分 |
|         | 7 | <sup>B<sub>L</sub></sup> <sub>E<sub>B</sub></sub> | '  | 7 | G | W |   | g | w |   |    | ア | キ | ヌ | ラ | 秒 |
|         | 8 | <sup>B<sub>S</sub></sup> <sub>C<sub>N</sub></sub> | (  | 8 | H | X |   | h | x |   |    | イ | ク | ネ | リ | ♠ |
|         | 9 | <sup>H<sub>T</sub></sup> <sub>E<sub>M</sub></sub> | )  | 9 | I | Y |   | i | y |   |    | ウ | ケ | ノ | ル | ♥ |
|         | A | <sup>L<sub>F</sub></sup> <sub>S<sub>B</sub></sub> | *  | : | J | Z |   | j | z |   |    | エ | コ | ハ | レ | ♦ |
|         | B | <sup>H<sub>M</sub></sup> <sub>E<sub>C</sub></sub> | +  | ; | K | [ |   | k | { |   |    | オ | サ | ヒ | ロ | ♣ |
|         | C | <sup>C<sub>L</sub></sup>                          | →  | , | < | L |   | ¥ | ! |   |    | ヤ | シ | フ | ワ | ● |
|         | D | <sup>C<sub>R</sub></sup>                          | ←  | - | = | M |   | ] | m | } |    | ユ | ス | ヘ | ン | ○ |
|         | E | <sup>S<sub>O</sub></sup>                          | ↑  | . | > | N |   | ^ | n | ~ |    | ヨ | セ | ホ | " | ◁ |
|         | F | <sup>S<sub>I</sub></sup>                          | ↓  | / | ? | O |   | _ | o |   |    | ツ | ソ | マ | ° | ▷ |

(注) SPは空白(スペース)コードを示します。



# 索引

## 数字・記号

|  |        |
|--|--------|
|  (スペースキー) .....   | 17     |
|  (カーソル移動キー) ..... | 17     |
|  (カーソル移動キー) ..... | 17     |
|  (カーソル移動キー) ..... | 17     |
|  (カーソル移動キー) ..... | 17     |
|  (リターンキー) .....   | 17     |
| <b>BS</b> (バックスペースキー) .....  | 17,299 |
| <b>CAPS</b> (キャピタルロックキー) .....   | 15     |
| <b>CLR</b> (クリアキー) .....   | 17     |
| <b>COPY</b> (コピーキー) .....  | 18     |
| <b>CTRL</b> (コントロールキー) .....   | 20     |
| <b>CTRL</b> (コントロールキー)を使った便利な編集機能 .....  | 24     |
| <b>DEL</b> (デリートキー) .....  | 17     |
| <b>ESC</b> (エスケープキー) .....   | 18     |
| <b>f・1</b> ~ <b>f・10</b> (ファンクションキー) .....   | 19,298 |
| <b>GRPH</b> (グラフキー) .....  | 16     |
| <b>HELP</b> (ヘルプキー) .....  | 18,23  |
| <b>HOME</b> (ホームキー) .....  | 17     |
| <b>HOME CLR</b> (ホーム・クリアキー) .....  | 17     |
| <b>INS</b> (インサートキー) .....   | 17     |
| <b>NFER</b> (無変換キー) .....  | 18     |
| <b>NUM</b> (ニューメリックロックキー) .....  | 15     |
| <b>ROLL DOWN</b> (ロールダウンキー) .....  | 18     |
| <b>ROLL UP</b> (ロールアップキー) .....  | 18     |
| <b>SHIFT</b> (シフトキー) .....   | 15     |
| <b>STOP</b> (ストップキー) .....   | 18,299 |
| <b>TAB</b> (タブキー) .....  | 17     |
| <b>XFER</b> (変換キー) .....   | 18     |
| <b>カナ</b> (カナキー) .....   | 15     |
| <b>"</b> (ダブルクォーテーション) .....   | 69     |
| 1MBタイプ .....   | 235    |
| 2D .....   | 96,334 |
| 2DD .....  | 96,334 |

|  |        |
|--|--------|
| 2HD .....                              | 96,334 |
| 2バイト系4分の1角文字 .....                     | 68     |
| 2バイト系4分の1角文字<br>(グラフィック画面に表示できる) ..... | 357    |
| 2バイト系全角文字 .....                        | 68     |
| 2バイト系半角文字 .....                        | 68     |
| 2バイト系半角文字<br>(グラフィック画面に表示できる) .....    | 356    |
| 2バイト系半角文字(文字列として扱える) .....             | 355    |
| 4096色中・16色モード .....                    | 56     |
| 4096色中・8色モード .....                     | 56     |
| 40桁20行モード .....                        | 28     |
| 40桁25行モード .....                        | 28     |
| 5インチ1Dフロッピディスク .....                   | 286    |
| 5インチ2Dフロッピディスク .....                   | 286    |
| 640KBタイプ .....                         | 235    |
| 7ビットコード .....                          | 138    |
| 80桁20行モード .....                        | 28     |
| 80桁25行モード .....                        | 28     |
| 8色中・8色モード .....                        | 56     |

## A

|                |     |
|----------------|-----|
| ATTR\$関数 ..... | 119 |
|----------------|-----|

## B

|                                      |         |
|--------------------------------------|---------|
| Basic modeの選択画面 (switch . n88) ..... | 276     |
| backup . hd .....                    | 265     |
| backup . n88 .....                   | 238     |
| BLOAD命令 .....                        | 117,179 |
| <b>BS</b> (バックスペースキー) .....          | 17,299  |
| BSAVE命令 .....                        | 117,179 |

## C

|              |     |
|--------------|-----|
| CALL命令 ..... | 185 |
|--------------|-----|

|  |         |
|--|---------|
| <b>[CAPS]</b> (キャピタルロックキー) .....               | 15      |
| CHAIN命令 .....                                  | 115     |
| CLEAR命令 .....                                  | 177,325 |
| CLOSE命令 .....                                  | 95      |
| CLOSE命令(RS-232C回線ファイル) .....                   | 143     |
| CLOSE命令(シーケンシャルファイル) .....                     | 124     |
| CLOSE命令(デバイスファイル) .....                        | 154     |
| CLOSE命令(ランダムファイル) .....                        | 129     |
| <b>[CLR]</b> (クリアキー) .....                     | 17      |
| [1] COLOR命令 .....                              | 31,54   |
| [2] COLOR命令 .....                              | 57      |
| COM1: (RS-232C回線ファイル) .....                    | 91      |
| COM2: (RS-232C回線ファイル) .....                    | 91      |
| COM3: (RS-232C回線ファイル) .....                    | 91      |
| COM ON/OFF/STOP命令 .....                        | 146     |
| CONSOLE命令 .....                                | 29      |
| <b>[COPY]</b> (コピーキー) .....                    | 18      |
| CRT初期画面 色指定(TEXT)の選択画面<br>(switch . n88) ..... | 275     |
| C <sub>R</sub> コード受信処理 .....                   | 297     |
| <b>[CTRL]</b> (コントロールキー) .....                 | 20      |
| <b>[CTRL]</b> (コントロールキー)を使った便利な編集機能 .....      | 24      |
| CVI/CVS/CVD関数 .....                            | 130     |
| <b>D</b>                                       |         |
| DDconv . n88 .....                             | 286     |
| DEF SEG命令 .....                                | 176     |
| DEF USR命令 .....                                | 180     |
| DELコード受信処理 .....                               | 141,297 |
| <b>[DEL]</b> (デリートキー) .....                    | 17      |
| dicmen . n88 .....                             | 278     |
| dir . hd .....                                 | 263     |
| DISK CODE .....                                | 244,323 |
| Disk version .....                             | 4       |
| DSKI\$関数 .....                                 | 120     |
| DSKO\$命令 .....                                 | 120     |
| DISKモードBASIC .....                             | 1,3     |
| DISKモードBASICの画面ハードコピー .....                    | 39      |
| DISKモードBASICの機能選択 .....                        | 9       |

|  |         |
|--|---------|
| <b>E</b>                                     |         |
| EOF関数(シーケンシャルファイル) .....                     | 124     |
| <b>[ESC]</b> (エスケープキー) .....                 | 18      |
| <b>F</b>                                     |         |
| <b>[F1]</b> ~ <b>[F10]</b> (ファンクションキー) ..... | 19,298  |
| FAC(浮動小数点アキュムレータ) .....                      | 182     |
| FAT(ファイル領域管理テーブル) .....                      | 338     |
| FIELD命令 .....                                | 128     |
| FILES命令 .....                                | 112     |
| FM/SSG音源内部レジスタマップ .....                      | 224     |
| FM音源 .....                                   | 209,213 |
| FM音源の原理 .....                                | 213     |
| format . hd .....                            | 253     |
| format . nip .....                           | 235     |
| FPOS関数(シーケンシャルファイル) .....                    | 124     |
| FPOS関数(ランダムファイル) .....                       | 130     |

|                              |     |
|------------------------------|-----|
| <b>G</b>                     |     |
| GET命令 .....                  | 129 |
| GET@命令 .....                 | 64  |
| <b>[GRPH]</b> (グラフキー) .....  | 16  |
| GP-IB(IEEE-488)インタフェース ..... | 189 |

|                                     |          |
|-------------------------------------|----------|
| <b>H</b>                            |          |
| h] (モニタモードのプロンプト) .....             | 303      |
| <b>[HELP]</b> (ヘルプキー) .....         | 18,23    |
| <b>[HOME]</b> (ホームキー) .....         | 17       |
| <b>[HOME CLR]</b> (ホーム・クリアキー) ..... | 17       |
| How many files(0-15)? .....         | 2,95,240 |

|                              |     |
|------------------------------|-----|
| <b>I</b>                     |     |
| ID(ディスクの) .....              | 339 |
| ID_NO(領域の) .....             | 256 |
| IDセクタ .....                  | 240 |
| IDセクタの書き換え .....             | 240 |
| ID番号(SCSI固定ディスク) .....       | 253 |
| INPUT#命令 .....               | 123 |
| INPUT\$関数 .....              | 123 |
| INP関数 .....                  | 347 |
| <b>[INS]</b> (インサートキー) ..... | 17  |
| IPL .....                    | 244 |

- K**
- KI(漢字イン) .....70
- KILL命令 .....118
- KINPUT命令 .....76
- KO(漢字アウト) .....70
- KPLOAD命令 .....75
- KYBD:(キーボードファイル) .....91,154
- L**
- LFILES命令 .....112
- LINE INPUT#命令 .....123
- LOAD命令 .....114
- LOC関数(シーケンシャルファイル) .....124
- LOC関数(ランダムファイル) .....130
- LOF関数(シーケンシャルファイル) .....124
- LOF関数(ランダムファイル) .....130
- LPT:(プリンタファイル) .....91,154
- LSET命令 .....128
- M**
- menu .....233
- MERGE命令 .....115
- mkfont.n88 .....267
- MKI\$/MKS\$/MKD\$関数 .....130
- N**
- NAME命令 .....118
- Network Basicの選択画面(switch.n88) ...276
- NFER(無変換キー) .....18
- NUM(ニューメリックロックキー) .....15
- O**
- ON COM GOSUB命令 .....146
- ON PEN GOSUB命令 .....172
- OPEN命令 .....93
- OPEN命令(RS-232C回線ファイル) .....136
- OPEN命令(シーケンシャルファイル) .....122
- OPEN命令(デバイスファイル) .....154
- OPEN命令(ランダムファイル) .....127
- P**
- PC-PR201V系カラープリンタ .....40
- PC-PR601系ページプリンタ .....41
- PEN ON/OFF/STOP命令 .....172
- PEN関数 .....172
- POKE命令 .....178
- PRINT#命令 .....123
- PRINT# USING命令 .....123
- PUT命令 .....128
- PUT@命令 .....64
- Q**
- q] (モニタモードのプロンプト) .....303
- R**
- recov.hd .....262
- ROLL DOWN(ロールダウンキー) .....18
- ROLL UP(ロールアップキー) .....18
- ROMモードBASIC .....1
- ROMモードBASICの画面ハードコピー .....39
- ROMモードBASICの起動 .....2
- RS-232Cインタフェース .....132
- RS-232Cインタフェース(2回線/3回線) .....189
- RS-232C(初期設定)の選択画面(switch.n88) .....274
- RS-232C(送受信コード)の選択画面(switch.n88) .....274
- RS-232C回線ファイル .....91,132
- RS-232C回線ファイルのオープン .....136
- RS-232C回線ファイルの入出力 .....94
- RSET命令 .....128
- RUN命令 .....114
- S**
- SAVE命令 .....113
- SCREEN命令 .....33
- SCRN:(スクリーンファイル) .....91,154
- SCSI固定ディスク .....97
- SET命令 .....118
- setinf.n88 .....240
- setup.n88 .....246

- SHIFT (シフトキー) .....15  
SSG音源 .....209,221  
STOP (ストップキー) .....18,299  
switch . n88 .....273  
sysgen . nip .....244  
Sパラメータ (OPEN命令) .....136,139  
Sパラメータ (TERM命令) .....297
- T**  
TAB (タブキー) .....17  
tele . n88 .....190,199  
TERM命令 .....296
- U**  
User identifier ? .....5  
usfontn88 (利用者定義文字格納ファイル) ...267  
USR関数 .....181
- V**  
VIEW命令 .....44
- W**  
WIDTH命令 .....29  
WINDOW命令 .....47  
WRITE#命令 .....123
- X**  
XFER (変換キー) .....18  
xfiles . n88 .....242  
Xパラメータ (OPEN命令) .....136,139  
Xパラメータ (TERM命令) .....297
- あ**  
アクティブページ .....34  
アスキー形式 (セーブ) .....113  
アスキーファイル .....89  
アドレス (ディスクの) .....335  
アルゴリズム (FM音源) .....213  
アルゴリズムの種類 .....219
- い**  
色の仕組み .....59  
インサートキー .....17  
印字 (日本語文字列の) .....78  
印字倍率 .....41  
印字方向 .....41
- う**  
ウィンドウ .....47  
ウォームリスタート .....11
- え**  
エスケープキー .....18  
演算モードの設定 .....250  
エンベロープ (FM音源) .....215  
エンベロープ (SSG音源) .....223  
エンベロープジェネレータ (FM音源) .....215  
エンベロープジェネレータ (SSG音源) .....222
- お**  
オートスタート .....7  
オートスタート情報 .....240  
オートリピート機能 .....22  
オープン (RS-232C回線ファイルの) .....136  
オープン (シーケンシャルファイルの) .....122  
オープン (デバイスファイルの) .....154  
オープン (ファイルの) .....93  
オープン (ランダムファイルの) .....127  
オフラインコマンドモード (電話制御機能) ...194  
オペレータ (FM音源) .....213  
オリジナルスクリーン座標系 .....43,44  
音色パラメータ .....216
- か**  
カーソル (マウスの) .....168  
カーソル移動キー .....17  
カーソルの形と中心点 (マウスの) .....168  
回線番号 .....136  
書き込み確認属性 .....118  
書き込み禁止属性 .....118  
拡張演算モード .....250  
拡張画面ハードコピー機能 .....40

- 拡張画面ハードコピーの機能設定 .....248  
 拡張機能 .....189  
 拡張グラフィックモード .....9,56  
 拡張子 .....109  
 拡張フォーマット(固定ディスクの) .....104,252  
 拡張フォーマットをした固定ディスク .....6  
 拡張ボードの選択画面(switch . n88) .....275  
 カナキー .....15  
 画面スイッチ .....38  
 画面ハードコピー .....38  
 画面ハードコピー機能の種類と機能 .....39  
 画面ハードコピー機能の選択 .....10  
 画面ハードコピーの選択画面(switch . n88) 275  
 画面モード .....32  
 カラーコードの指定方法 .....59  
 カラーコピー .....40  
 カラーモード(グラフィック画面) .....32  
 カラーモード(テキスト画面) .....30  
 漢字アウト(KO) .....70  
 漢字イン(KI) .....70  
 管理テーブル(ディスクの) .....337
- き**  
 キーセンス .....347  
 キーの配置 .....13  
 キーボードファイル(KYBD:) .....91,154  
 機械語関数 .....180  
 機械語ファイルのセーブ .....117  
 機械語ファイルのロード .....117  
 機械語プログラム .....175  
 機械語プログラムセグメント .....304,324  
 機械語プログラムの実行 .....180  
 機械語プログラムの準備 .....177  
 機械語プログラム用メモリ領域 .....177  
 輝度 .....59  
 基本演算モード .....250  
 基本グラフィックモード .....9,56  
 キャピタルロックキー .....15  
 キャラクタコード表 .....359  
 キャリア(FM音源) .....214  
 行数 .....28  
 行の削除 .....22
- 行の挿入 .....22
- く**  
 クラスタ .....334  
 クラスタ番号とディスクのアドレスとの関係 336  
 グラフィックシンボル .....16  
 グラフィック画面 .....32  
 グラフィック画面の色 .....54  
 グラフィック画面の座標系 .....43  
 グラフィック画面への表示(日本語文字列の) 77  
 グラフィック機能の選択 .....9  
 グラフィックス .....43  
 グラフキー .....16  
 クリアキー .....17  
 クローズ(RS-232回線ファイルの) .....143  
 クローズ(シーケンシャルファイルの) .....124  
 クローズ(デバイスファイルの) .....154  
 クローズ(ファイルの) .....94  
 クローズ(ランダムファイルの) .....129
- け**  
 桁数 .....28
- こ**  
 高分解能カラーモード .....32  
 高分解能白黒モード .....32  
 固定ディスク .....96,100,252  
 固定ディスクからの起動 .....6  
 固定ディスク障害ボリューム・ファイル復旧 262  
 固定ディスク装置 .....5  
 固定ディスクのバックアップ .....265  
 固定ディスクのフォーマット .....252  
 固定ディスクのフォーマット(拡張フォーマット) .....255  
 固定ディスクのフォーマット(標準フォーマット) .....253  
 固定ディスクボリューム管理 .....252  
 固定ディスクファイル退避/復旧処理 .....265  
 固定ディスクファイルディレクトリ表示 .....263  
 コピーキー .....18  
 コントロールキー .....20  
 コントロールキーを使った便利な編集機能 .....24

- さ
- サーフェス .....333
  - サウンドバッファ .....211
  - サウンドバッファの確保 .....212
  - サウンドバッファの変更 .....213
  - サウンド制御機能 .....189,209
  - サウンド制御命令 .....210
  - サウンド制御命令の使用準備 .....210
  - 削除(ファイルの) .....118
  - 削除(行の) .....22
  - 削除(単語の) .....283
  - 削除(文字の) .....21
  - 座標系 .....43
- し
- シーケンシャルファイル .....90,121
  - シーケンシャルファイルのオープン .....122
  - 辞書ファイル .....73,278
  - 辞書ファイルの保守 .....73,278
  - システムエリア .....322
  - システムディスク .....98,235
  - システムディスクからの起動 .....4
  - システムディスク属性の設定 .....98,246
  - システムのコピー .....244
  - 実行(RUN) .....114
  - シフトキー .....15
  - 出力モード .....94
  - 主ファイル名 .....109
  - 順次ファイル .....90
  - 状態変更(format . hd) .....260
  - 初期モード(電話制御機能) .....194
  - 白黒モード(グラフィック画面) .....32
  - 白黒モード(テキスト画面) .....30
  - シンボルテーブルセグメント .....324
- す
- 数値演算プロセッサ .....353
  - 数値演算プロセッサの選択画面( switch . n88)  
.....275
  - スクリーンエディタ .....21
  - スクリーンファイル(SCRN : ) .....91,154
- スクリーン座標系 .....43,44
- スクロール .....29
- スクロール画面 .....29
- スクロール用メモリ .....295
- 図形データの転送 .....64
- ストップキー .....18,299
- ストップビット長(OPEN命令) .....136,139
- ストップビット長(TERM命令) .....297
- ストリングディスクリプタ .....182
- スペースキー .....17
- スペシャルESCシーケンス .....299
- せ
- セーブ(機械語ファイルの) .....117
  - セーブ(プログラムファイルの) .....113
  - 整数型変数 .....328
  - セクタ .....333
  - セグメント .....176
  - セグメントベース .....176
  - セグメントベースの宣言 .....176
  - 絶対アドレス .....176
  - 全二重 .....290
- そ
- 相対アドレス .....177
  - 装置初期化(format . hd) .....259
  - 装置制御コードD1/D3 .....293
  - 挿入(行の) .....22
  - 挿入(文字の) .....21
  - 属性(システムディスクの) .....246
  - 属性(日本語入力) .....246
  - 属性(ファイルの) .....118
  - ソフトウェアドライバ .....157
  - ソフトウェアドライバのファンクション .....159
  - ソフトウェアドライバの呼び出し .....159
- た
- ターミナルの仕様 .....290
  - ターミナルの仕様の設定 .....296
  - ターミナルの諸元 .....292
  - ターミナルの使い方 .....298
  - ターミナルモード .....289

- ターミナルモードの起動 .....289  
 タイリング .....60  
 タイルストリング .....60  
 タイルパターン .....60  
 立ち上げ装置の選択画面(switch . n88) .....276  
 タブキー .....17  
 ダブルクォーテーション .....69  
 単語の一覧(dicmen . n88) .....284  
 単語の削除(dicmen . n88) .....283  
 単語の登録(dicmen . n88) .....279  
 単精度実数型変数 .....329
- ち**  
 逐次ファイル .....90
- つ**  
 追加モード .....94  
 通常の画面ハードコピー機能 .....39  
 通信方式 .....297  
 通話異常時の原因と処置 .....207  
 通話モード(電話制御機能) .....195
- て**  
 データ通信モード(電話制御機能) .....194  
 データディスク .....235  
 データの内部形式 .....327  
 データビット長(OPEN命令) .....136,137  
 データビット長(TERM命令) .....297  
 データファイル .....89,121  
 ディスクのセクタへの直接アクセス .....120  
 ディスクの内部構造 .....333  
 ディスクのアドレス .....333  
 ディスクファイル .....90,91,96  
 ディスクの読み書き .....335  
 ディスプレイページ .....34  
 ディレクトリ .....337  
 ディレクトリ表示機能 .....263  
 テキスト画面 .....27  
 テキスト画面の座標 .....28  
 テキスト画面の表示モード .....31  
 テキスト画面の文字の色 .....31  
 テキスト画面への表示(日本語文字列の) .....77
- テキストセグメント .....322  
 テキストファイル .....89  
 デバイス .....90  
 デバイスファイル .....90,154  
 デバイスファイルのオープン .....154  
 デバイス名 .....91  
 デリートキー .....17  
 テンキー .....20  
 転送速度(電話機の) .....204  
 電話管理ユーティリティ .....190,199  
 電話管理ユーティリティの起動 .....201  
 電話制御機能 .....190  
 電話制御機能の使用の有無 .....10  
 電話制御命令 .....191  
 電話制御命令の使用宣言 .....192
- と**  
 同期クロック .....141,298  
 同時オープン数(ファイルの) .....95,240  
 特殊キー .....13,15  
 ドライブアロケーションタイプ .....236,245  
 ドライブ番号 .....107,109  
 トラック .....333  
 トラックの割り当て  
 (固定ディスク, 拡張フォーマットの場合) .....346  
 トラックの割り当て  
 (固定ディスク, 標準フォーマットの場合) .....343  
 トラックの割り当て  
 (フロッピーディスクの場合) .....341
- な**  
 内部形式(日本語文字列の) .....71  
 内部構造(ディスクの) .....333
- に**  
 日本語コードの送受信 .....295  
 日本語コード表 .....355  
 日本語シフトコード .....70,297  
 日本語入力の属性設定 .....246  
 日本語変換方式の選択 .....10  
 日本語文字 .....67  
 日本語文字の出力 .....77

- 日本語文字の種類 .....68  
 日本語文字の入力 .....76  
 日本語文字列操作関数 .....79  
 入出力装置 .....90  
 入力装置 .....157  
 入力モード .....94  
 ニューメリックロックキー .....15
- は**  
 倍精度実数型変数 .....330  
 バイナリ形式(セーブ) .....113  
 バイナリファイル .....89  
 配列データセグメント .....324  
 バックアップ(固定ディスクの) .....265  
 バックアップ(ファイルの) .....242  
 バックアップ(フロッピィディスクの) .....238  
 バックグラウンドカラー .....54  
 バックスペースキー .....17,299  
 パリティチェック(OPEN命令) .....136,137  
 パリティチェック(TERM命令) .....297  
 パレットの色の変更 .....57  
 パレット番号 .....57  
 パレットモード .....55  
 半二重 .....290
- ひ**  
 標準フォーマット(固定ディスクの) .....104,252  
 標準フォーマットをした固定ディスク .....6  
 ビューポート .....44
- ふ**  
 ファイル .....89  
 ファイル削除機能 .....263  
 ファイルディスクリプタ .....92,109  
 ファイル転送 .....242  
 ファイル出力(日本語文字列の) .....79  
 ファイル入力(日本語文字列の) .....76  
 ファイルのオープン .....93  
 ファイルのクローズ .....94  
 ファイルの削除 .....118  
 ファイルの種類 .....89  
 ファイルの属性 .....118
- ファイルの同時オープン数 .....95  
 ファイルのバックアップ .....242  
 ファイルバッファ .....92  
 ファイル復旧 .....262  
 ファイル変換 .....286  
 ファイル名 .....109  
 ファイル名の変更 .....118  
 ファンクションキー .....19,298  
 ファンクションキーの表示 .....30  
 ファンクション(マウスの) .....159  
 フォーマット(固定ディスクの) .....104,252  
 フォーマット(フロッピィディスクの) .....97,234  
 フォアグラウンドカラー .....54  
 不揮発性メモリ .....273  
 複写(図形データの) .....64  
 物理アドレス .....176  
 物理フォーマット(固定ディスクの) .....104,252  
 物理フォーマット(フロッピィディスクの) .....97,234,238  
 浮動小数点アキュムレータ(FAC) .....182  
 プリントファイル(LPT:) .....91,154  
 フロー制御 .....293  
 プログラムの修正 .....21  
 プログラムのマージ .....115  
 プログラムの連結 .....115  
 プログラムファイル .....89  
 プログラムファイルのセーブ .....113  
 プログラムファイルのロード .....114  
 フロッピィディスク .....96  
 フロッピィディスク装置が1台の場合 .....232  
 フロッピィディスクのバックアップ .....238  
 フロッピィディスクのフォーマット .....234  
 プロンプトh] (モニタモード) .....306  
 プロンプトq] (モニタモード) .....306
- へ**  
 ページ .....33  
 ヘルプキー .....18,23  
 変換キー .....18  
 変更(ファイル名の) .....118  
 編集コマンド .....23  
 変数領域の大きさ .....297

- ほ**
- ボーダーカラー .....54
  - ホームキー .....17
  - ホーム・クリアキー .....17
  - ポーレート .....141,192,297
  - ポインティングデバイス .....157
  - ボリューム復旧 .....262
- ま**
- マージ(プログラムの) .....115
  - マウス .....157
  - マウス環境の初期設定 .....168
  - マウスの画面座標系 .....168
  - マップ表示(format .hd) .....256
  - マルチユーザー機能 .....101
  - マウス用ソフトウェアドライバ .....157
- み**
- ミッキー .....168
  - ミュージックジェネレータボード .....226
- む**
- 無変換キー .....18
- め**
- メニュープログラム .....233
  - メモリサイズの選択画面( switch .n88) .....274
  - メモリスイッチ .....349
  - メモリスイッチのセット .....349
  - メモリスイッチの設定 .....273
  - メモリスイッチの値と機能 .....350
  - メモリマップ .....321
  - メモリ領域の配分・確保 .....325
  - メモリレイアウト .....176
- も**
- 文字型定数(日本語文字列の) .....69
  - 文字型変数 .....331
  - 文字キー .....14
  - 文字の削除 .....21
  - 文字の挿入 .....21
- 文字の変更 .....21
- 文字編集 .....17
- モジュレータ(FM音源) .....214
- モデム-NCU内蔵電話機 .....190
- モニタモード .....303
- モニタモードのコマンド一覧 .....305
- モニタモードの使用の有無 .....10
- モニタモードへの切り換え .....303
- モノクロコピー .....40
- ゆ**
- ユーザー識別名 .....5,101
  - ユーティリティプログラム .....231
  - ユーティリティプログラム一覧 .....231
  - 優先順位の選択(ドライブ番号の) .....107
- ら**
- ライトペン .....172
  - ランダムファイル .....90,121,126
  - ランダムモード .....94
- り**
- リセット .....11
  - リターンキー .....17
  - リターンキー送信処理 .....297
  - リモートBASICプロトコル .....301
  - 領域解放(format .hd) .....258
  - 領域確保(format .hd) .....257
  - 領域分割(固定ディスクの) .....100
  - 利用者定義文字 .....74,267
  - 利用者定義文字格納ファイルの作成・更新 .....267
  - リロケーションコード .....182
- れ**
- 連結(プログラムの) .....115
- ろ**
- ロード(機械語ファイルの) .....117
  - ロード(プログラムファイルの) .....114
  - ロールアップキー .....18
  - ロールダウンキー .....18
  - 論理フォーマット(固定ディスクの) .....104,252

論理フォーマット(フロッピーディスクの)  
.....97,234

わ

ワールド座標系 .....43,46  
ワールド座標値の制限 .....53  
割り込み処理(RS-232C回線ファイルの) .....146  
割り込み処理(キー) .....20  
割り込み処理(サウンド) .....211  
割り込み処理(電話機) .....191  
割り込み処理(マウス) .....165  
割り込み処理(ライトペン) .....172







**NEC**



PCN24  
59920301