

# JRシリーズにエース

昨年JR-100を出した松下電器が今度、JR-200を発売しました。さっそく使ってみたのですが、もうこれはJR-100とはまったく別のマシンだ、というのが実感です。JR-200は、①このクラスでは最高の32KバイトのRAMを実装し、フルにプログラムエリアに使える ②8色カラーで大文字はもちろん小文字やカタカナ、グラフィックパターン、ユーザー定義文字が表示できる ③9桁の浮動小数点の計算ができ、2400ボートの高速カセットインタフェイスでデータファイルも作れる ④さらにプログラムと3つの音が同時進行できる三重和音機能、マシン語に便利なモニタ機能までついて、なんと79,800円。思わず「やったね！」の本格派マシンの登場です。

## 1 外観・キーボード

ボディはシルバー、キーボードはグレーと青の精悍なデザイン、JR-200は低価格機にありがちなオモチャ的デザインから完全に脱却しています。JR-100で、消しゴムキーと陰口をたたかれたゴム接点一体型キーも、JR-200ではストロークが大きくなって非常に押しやすくなっています。

キーの配列は標準JISキーボードとほとんど同じで、カーソルコントロールキーは右側にまとめられています。

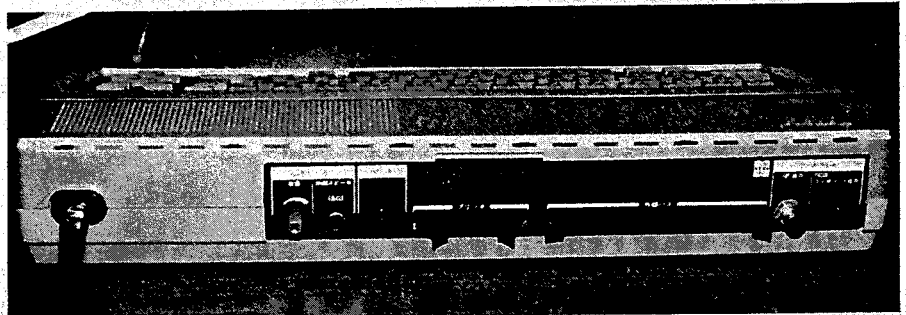


写真1 JR-200の背面部

プログラムの修正に便利です。リピート機能も付いていて、カーソルを動かした時、PC-6001のようにカーソルが消えてしまうことがなく、操作性は抜群です。

キーボードはサブCPU転送方式で、112文字分の文字バッファを備えているので、テープをロードしている最中でもRUNなどがキーインできます。ですから、テープをロードし終わったとたんに、プログラムをRUNできます。もちろんRUN“ファイルネーム”というのがありますが……。

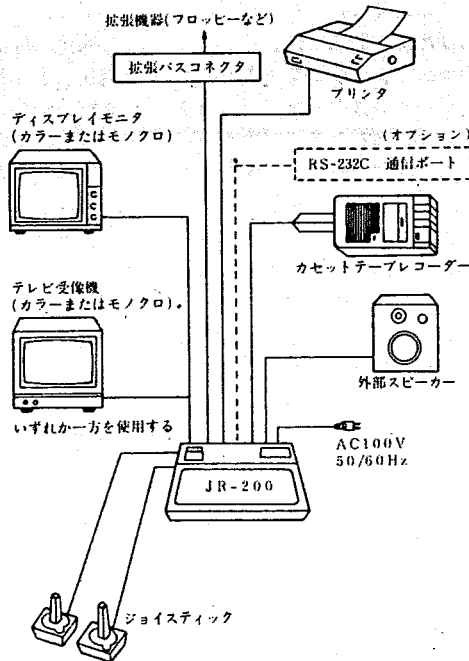


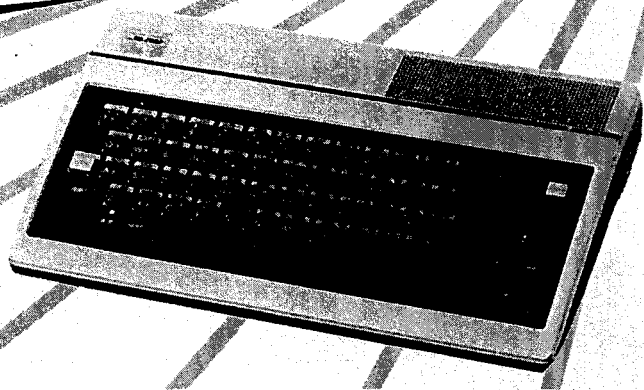
図1 JR-200のシステム構成

## 2 システム構成

図1にシステム構成を示します。モニタTVは、RGBでもコンジットビデオでもRFでもOKなので、たいのTVが使えます。ただし、RGBとコンジット

# JR-200 登場!

79,800円



山西一啓

の場合は、別売のケーブルが必要です。ジョイスティックも2個付けられます。拡張バスコネクタから拡張ユニットを介してフロッピーディスクを接続することもできます。

## 3 ハードウェア

表1にJR-200の仕様を示します。RAMは32KBも実装されていて、そのほとんどをプログラムエリアとして使えます。写真2(次頁)でもわかるように、この部分はたった4個の64KダイナミックRAMで構成されています。4ビット×64Kをうまく8ビット×32Kとして使っているわけです。また、ICの数が極端に少ないことも見逃せません。表2に使用されているLSIを示しますが、どれも最新のLSI技術を駆使したものばかりです。このほかにはTTLを20個ばかり使っているだけで、きわめてシンプルな構成となっています。

図2にメモリマップを示します。キャラクタRAMというのがありますが、ROMの間違いではありません。普通これはROMなのですが、JR-200では電源投入時、サブCPUからキャラクタパターンを転送するのです。つまり、プログラムによって文字の字体を変えたりできるのです。これはひょっとしてディスクシステムにした時、ひらがな表示にしたり、専用システムとして特殊文字や、ある

|                        |  |
|------------------------|--|
| C P U                  | <ul style="list-style-type: none"> <li>●メインCPU MN1800A (6802相当, 6802の外部クロック版)</li> <li>●サブCPU MN1544 (4ビット, 1チップ)</li> </ul>   |
| メモリ                    | <ul style="list-style-type: none"> <li>●ROM 16Kバイト実装</li> <li>●RAM 32Kバイト実装(プログラムエリア30716バイト)</li> <li>●ビデオRAM 2Kバイト実装</li> <li>●キャラクタRAM 2Kバイト実装(サブCPUからパターンを転送)</li> </ul>   |
| キーボード                  | <ul style="list-style-type: none"> <li>●方式 サブCPU転送</li> <li>●キー 64キー, 英数・グラフィック・カナ・ファンクションモード切り換え</li> </ul>   |
| ディスプレイ<br>インタフェイス      | <ul style="list-style-type: none"> <li>●方式 RGB同期分離方式<br/>コンポジットビデオ方式(カラー・モノクロ切り換え)<br/>RF変調方式(Ch-1または2切り換え, カラー・モノクロ切り換え)</li> <li>●画面構成 32文字×24行</li> <li>●色指定 文字・バックともにキャラクタ単位で任意に色指定可能<br/>指定色は, 黒・青・赤・マゼンタ(紫)・緑・シアン(水色)・黄・白の8色</li> <li>●文字構成 6×7ドットマトリクス文字—192種<br/>8×8ドットセミグラフィック文字—64種<br/>8×8ドットユーザー定義文字—64種</li> <li>●グラフィック 64×48ドット, ドットごとに上記8色で任意に色指定可能</li> </ul> |
| ミュージック                 | <ul style="list-style-type: none"> <li>●仕様 5オクターブの音域で, 3種のメロディーを同時演奏できる<br/>プログラムと同時進行可能</li> <li>●出力 内蔵スピーカー(音量調節付き), および外部スピーカー端子</li> </ul>   |
| カセット<br>インタフェイス        | <ul style="list-style-type: none"> <li>●方式 FSK方式, 1200Hz(スペース), 2400Hz(マーク)</li> <li>●ボーレート 600ボー, 2400ボー切り換え, ソフトウェアで切り換え可能. LOAD時は自動切り換え</li> <li>●その他 リモート端子付き, DATAファイル作成可能</li> </ul>   |
| プリンタ<br>インタフェイス        | <ul style="list-style-type: none"> <li>●端子 セントロニクス標準<br/>専用プリンタJR-P01, またはEPSON MP-80IIIシリーズ使用可</li> </ul>   |
| ジョイスティック               | <ul style="list-style-type: none"> <li>●アドコム電子特製ジョイスティックに適合する端子2個(9ピン)</li> </ul>  |
| RS-232Cユニット<br>(オプション) | <ul style="list-style-type: none"> <li>●ボーレート 9600ボーまで可能</li> <li>●その他 本体内部に実装</li> </ul>  |
| 外部拡張                   | <ul style="list-style-type: none"> <li>●50ピン拡張バスコネクタ付き. 拡張ユニットを介してフロッピーディスクユニットを接続可能</li> </ul>  |
| 電源                     | AC100V, 50/60Hz, 約8W   |
| 寸法・重量                  | <ul style="list-style-type: none"> <li>●寸法(高さ×幅×奥行) 56×348×208mm</li> <li>●重量 約1.7kg(本体)</li> </ul>  |
| 付属品                    | <ul style="list-style-type: none"> <li>●RFケーブル</li> <li>●録音ケーブル</li> </ul>   |

表1 JR-200の仕様

## RAM

いは漢字の表示のためかもしれません。本当のところは、筆者にもわかりません。

ビデオRAMは、図3のようにアトリビュートエリアで文字単位の色指定と、ユーザー定義文字とグラフィックモードを切り換えています。図4にコード表を示します。

英数字、グラフィック、カナ、およびインサートのモードはカーソルの形や色を変えて区別しているので、非常にわかりやすくなりました。一部特殊コードと重なっている部分はPRINT文で表示できませんが、たとえばCTRL + SHIFT + Aで「年」が表示できたりします。試してみてください。

グラフィックは64×48ドットで、PLOT文を用いて1ドットずつ表示できます。もっと細かいパターンを表示したい場合は、ユーザー定義文字を使って、256×192ドットの分解能で表示できます。もちろん、一度に表示できるのは64文字分ですから、限度があります。ラムちゃんやメーテルを描こうとするなら、フルグラフィックにはかないません。しかし、パターンの移動をやろうとすると、動作が遅く、マシン語に頼らざるを得なくなってきます。その点ユーザー定義文字なら、PRINT文で手軽に表示でき、BASICでも割と速い移動ができます。

ユーザー定義文字を表示する時は、COLOR—文字色、バックの色、1で行ないます。

## 4 JR-200 BASIC

JR-200 BASICでは中間言語を使っているため、1行入力するたびにエラーチェックが行なわれます。小文字でプログラムを打ち込んでも、リストを取ると大文字に変わります。

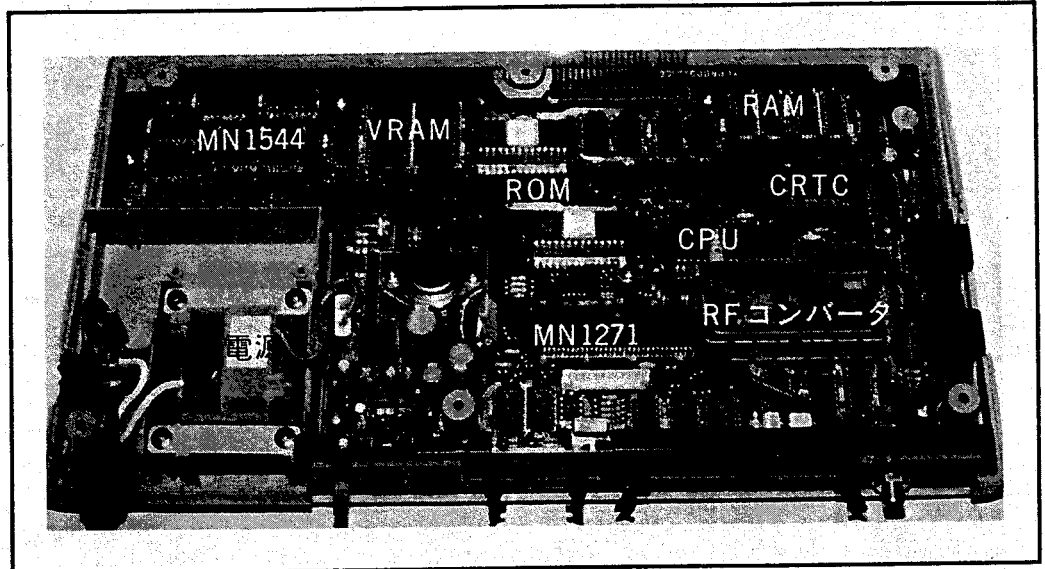


写真2 JR-200の内部

| IC         | 個数 | 内 容  |
|------------|----|--|
| MN1800A    | 1  | NMOS 8ビット CPU 6802の外部クロック版   |
| MN1544CJR  | 1  | NMOS 4ビット 1チップマイコン 40Pシュリンク・パッケージ  |
| MN1271     | 1  | CMOS パラレルI/O 8ビット×4, 3ビット×1<br>シリアルI/O×1<br>プログラム・タイマー 16ビット×2, 8ビット×2<br>64Pシュリンク・パッケージ |
| HM6116P-3  | 2  | CMOS RAM 8ビット×2K   |
| M5L2764K   | 2  | NMOS EPROM 8ビット×8K   |
| μPD4164C-2 | 4  | NMOS DRAM 1ビット×64K   |
| HD61K201F  | 1  | CMOS セミカスタム? (CRTコントローラと思われる)<br>80Pフラット・パッケージ   |

表2 JR-200に使用されているLSI

|                 |   |   |
|-----------------|---|---|
| 数 値 変 数         | 英字または英字+英数字。<br>配列は後ろにカッコを付ける。  | A, AB, X 2<br>A(I), AB(5, 0)  |
| 文 字 列 変 数       | 英字+\$または英字+英数字+\$。<br>配列は後ろにカッコを付ける。<br>最大255文字まで(上限はDIMで宣言する, 宣言しなければ32文字)                         | A\$, AB\$, A2\$<br>A\$(3), X2\$(I, J)   |
| 数 値 定 数         | 有効数字9桁 10 <sup>-39</sup> ~10 <sup>38</sup> , 16進数は\$0~\$FFFF<br>整数形式<br>固定小数点形式<br>浮動小数点形式<br>16進形式 | 1, -21<br>3.14, .255<br>6E23, 1.6E-19<br>\$41, \$C100   |
| 文 字 列 定 数       | 「”」または「'」で囲む  | “JR-200—BASIC”<br>'ABC'   |
| 算 術 演 算         | 加算(+), 減算(-), 乗算(*), 除算(/), べき乗(^), 余り(MOD)   |   |
| 比 較 演 算         | =, <, >, <=, >=   |   |
| 論 理 演 算         | NOT, AND, OR  |   |
| 文 字 列 演 算       | 「+」による結合, 比較演算子による比較  |   |
| コ マ ン ド と 文     |   | AUTO, BEEP, CLEAR, CLOSE, CLS, COLOR, CONT, DATA<br>DELETE, DIM, END, FIND, FOR~NEXT, GOSUB~RETURN<br>GOTO, HCOPY, IF~THEN, INITP, INPUT, INPUT#, LET<br>LFIND, LIST, LLIST, LOAD, LOCATE, LPRINT, MERGE<br>MLOAD, MON, MSAVE, NEW, ON~GOSUB, ON~GOTO<br>OPEN, PICK, PLAY, PLOT, POKE, PRINT, PRINT#<br>RANDOMIZE, READ, REM, RESTORE, RUN, SAVE,<br>SOUND, STOP, TEMPO, VERIFY |
| 関 数             |   | ABS, ASC, CHR\$, COS, EXP, FRE, HEX\$, HPOS, VPOS<br>INT, LEFT\$, LEN, LOG, MID\$, MOD, PEEK, RIGHT\$<br>RND, SGN, SIN, SPC, SQR, STICK, STR\$, TAB, USR<br>VAL, VARPTR   |
| モ ニ タ の コ マ ン ド |   | D 128バイトのメモリ内容の表示<br>M メモリ内容の参照, 変更<br>G 指定アドレスからプログラムを実行   |

表3 JR-200 BASICの概要

コマンドとステートメントは1バイトのコードに、関数は\$FF+1バイトのコードに、数値は数によって3~6バイトに、GOTOなどの飛び先行番号は3バイトに変換されます。そのため、実行速度が速くなりました。たとえば、某誌におけるベンチマークテストでは101秒をマークしました。JR-100で153秒ですから、1.5倍は速いということになります。実数BASICでは最も速いと言われているパソピアT-BASICで92秒ですか

ら、それに次ぐ速さと言うことができるでしょう(ただし、パソピアの有効桁は7桁です)。

精度については、有効桁は9桁ですが、10桁目まで考慮しているみたいで、

PRINT 1 / 3 - .333333333

を実行すると、

3.49245965E-10

と表示し、10桁目まで正しいことがわかります。計算

桁数と表示桁数が同じで、下の方はまったくあてにならないパソコンもある中で、非常に良心的です。JR-200は電卓事業部で作られているので、電卓のノウハウを活かしたものだと思います。

変数は英字+英数字となり、文字列配列も使えるようになりました。文字列変数はDIM文で最大文字数を宣言できるので、必要に応じたメモリの使用ができ、メモリの節約になります。宣言しなければ、最大文字数は32文字です。配列も宣言しないで使うと、添字の上限が自動的に10に設定されます。下限は0になったので、いっそう使いやすくなったと言えるでしょう。

表3にBASICの概要を示しますが、その中で特徴のあるものを挙げてみます。

まずLOCATEは、N-BASICと同じX,Yの順になったので、移植がしやすくなりました。

CLEAR文は、変数エリアをクリアするとともに、BASICの使用上限を設定できるので、マシン語や音譜データの保護ができます。

PLAYは、メモリに格納されている音譜データをもとに自動的に曲を演奏します。音譜データは、1つの音について音符と音階の2

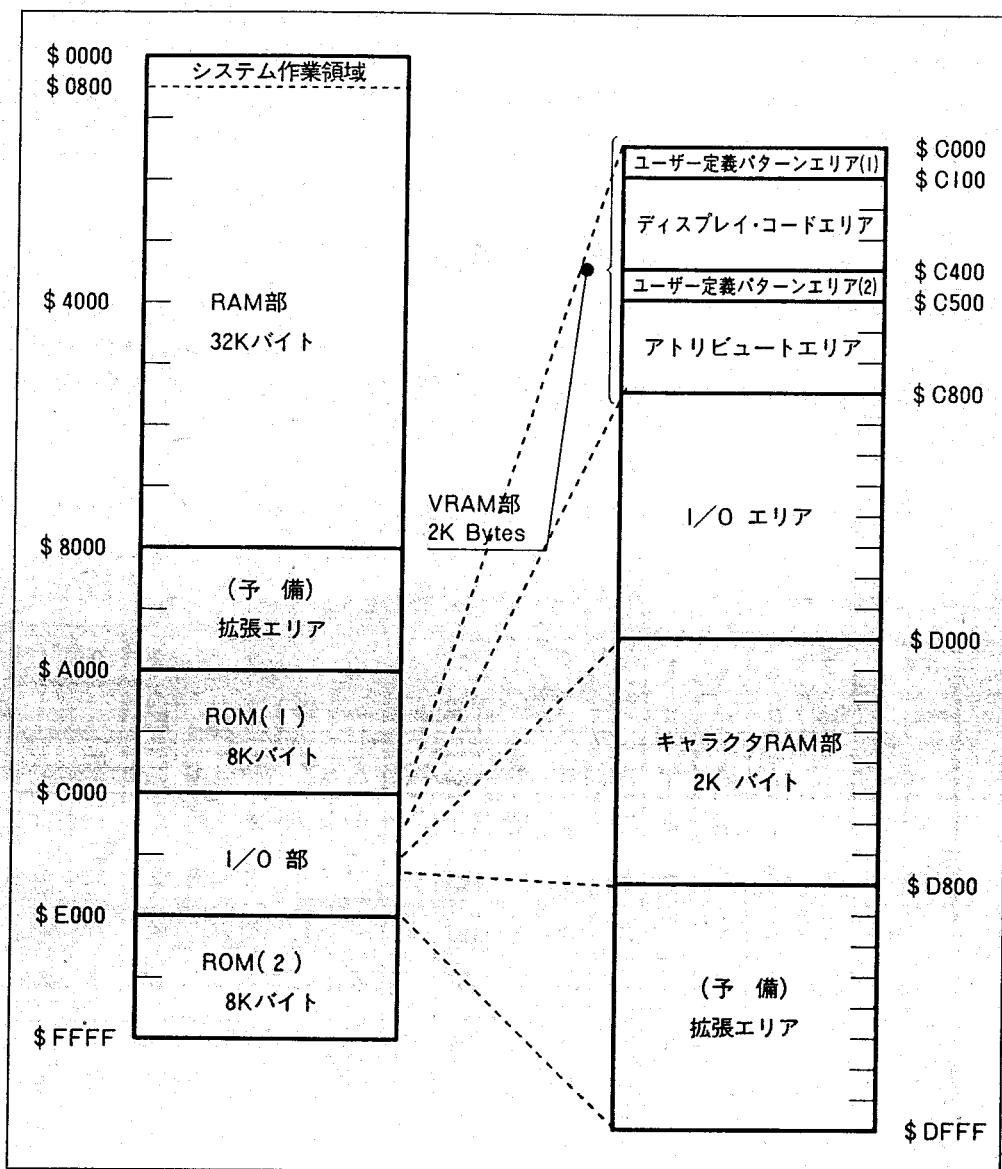


図2 JR-200のメモリマップ

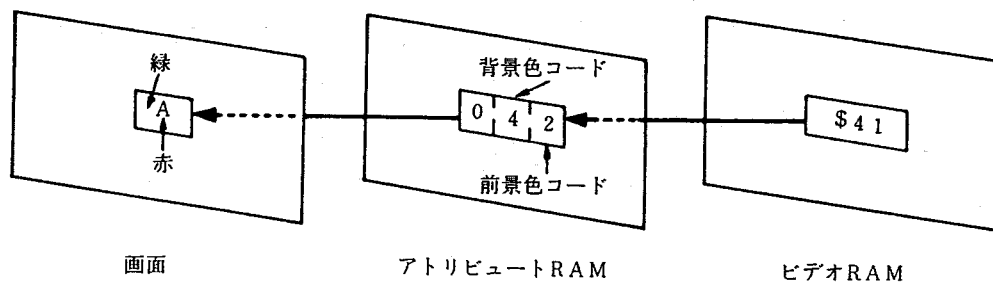


図3 ビデオRAMの仕組

# RAM

バイトで構成され、モニタモードでメモリに直接書き込んでいきます。フォアグラウンドとバックグラウンドのモードがあり、フォアグラウンドでは演奏が終わるまでプログラムは止まっていますが、バックグラウンドの場合プログラムと同時進行できます。ラリーXなどのゲームにはもってこいの機能ですね。なお、PLAYは3つのパートを同時に演奏できます。

ON~GOTO, ON~GOSUB文が新たに加わりましたが、これは中間言語になったので、

GOTO\_A

などが使えなくなったためです。しかし、その方が読みやすいプログラムが作れるので大歓迎です。読みやすいと言えば、IF~THEN~ELSEが使えれば申し分なかったのですが。

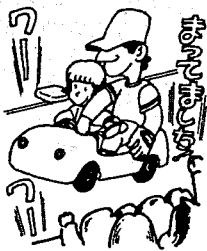
## 5 スクリーンエディット機能

中間言語に変換しているためか、2行以上の文も、1行目でRETURNを押しても、うまく入力されます。

インサートモードでは、カーソルが紫色になり、インサートした文字も紫色になるので、非常にわかりやすくなりました。

リストも、キーを押すと途中で止まり、もう一度押すと続きをリストするというふうになり、修正箇所を見つけやすくなりました。

FINDは、見つけたい文字列を含む行をすべて表示するという、他のパソコンには見られないユニークな機能です。JR-100でも重宝していました



▶図4 JR-200のキャラクタコード表

| 下位4ビット \ 上位4ビット | 0          | 1          | 2    | 3 | 4 | 5 | 6 | 7   | 8    | 9   | A | B | C | D | E | F | 0 | 1 |
|-----------------|------------|------------|------|---|---|---|---|-----|------|-----|---|---|---|---|---|---|---|---|
| 0               | NULL       |            | (sp) | 0 | @ | P | ` | p   | (??) | (カ) | 一 | タ | ミ | ○ | □ |   |   | ~ |
| 1               |            |            | !    | 1 | A | Q | a | q   | ♠    | □   | 。 | ア | チ | ム | □ | □ | 年 | ÷ |
| 2               |            |            | ▼    | 2 | B | R | b | r   | ♥    | □   | 「 | イ | ツ | メ | □ | □ | 月 | α |
| 3               | BRE<br>AK  | INS        | #    | 3 | C | S | c | s   | ♦    | □   | 」 | ウ | テ | モ | □ | □ | 日 | β |
| 4               |            | (英数)       | \$   | 4 | D | T | d | t   | ♣    | □   | 、 | エ | ト | ヤ | □ | □ | 火 | γ |
| 5               |            |            | %    | 5 | E | U | e | u   | ★    | □   |   | オ | ナ | ユ | □ | □ | 水 | Σ |
| 6               |            | HCO<br>PY  | &    | 6 | F | V | f | v   | ←    | □   | ヲ | カ | ニ | ヨ | □ | □ | 木 | θ |
| 7               | BEEP       |            | ▼    | 7 | G | W | g | w   | ↓    | □   | ア | キ | ヌ | ラ | □ | □ | 金 | 目 |
| 8               | RUB<br>OUT | CAN<br>SEL | (    | 8 | H | X | h | x   | ↑    | □   | イ | ク | ネ | リ | □ | □ | 土 | 目 |
| 9               |            |            | )    | 9 | I | Y | i | y   | →    | □   | ウ | ケ | ノ | ル | □ | □ | 時 | 目 |
| A               |            | LINS       | *    | : | J | Z | j | z   | 人    | □   | エ | コ | ハ | レ | □ | □ | 分 | 目 |
| B               | HOME       |            | +    | ; | K | [ | k | {   | ☺    | □   | オ | サ | ヒ | ロ | □ | □ | 秒 | ♪ |
| C               | CLS        | →          | .    | < | L | ¥ | l |     | ■    | □   | ヤ | シ | フ | ワ | □ | □ | 円 | ◎ |
| D               | RET<br>URN | ←          | -    | = | M | ] | m | }   | □    | □   | ユ | ス | ヘ | ン | □ | □ | 生 | 〒 |
| E               |            | ↑          | .    | > | N | ^ | n | -   | ■    | □   | ヨ | セ | ホ | ッ | □ | □ | ☎ | ◎ |
| F               |            | ↓          | /    | ? | O | _ | o | DEL | □    | □   | ツ | ソ | マ | ° | □ | □ | π |   |

ユーザー定義エリア  
に対応

ディスプレイでは  
一部異なる

だが、JR-200では、さらに行番号のエリア指定ができるようになりました。また、LFINDもあるので、プリンタにも打ち出せます。

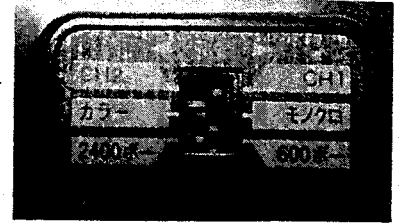


写真3 JR-200のボーレート切り換え部

複数行を一度に消してしまえるDELETE文と、カセットからプログラムを追加できるMERGE文が加わって、エディットの機能はかなり強力になったと思うのですが、RENUMがないのが気になりました。MERGEがあつてRENUMがないのは、片手落ちではないでしょうか。

## 6 カセットインタフェイス

カセットインタフェイスは従来の600ボーに加え、4倍の2400ボーが可能になりました。これは、数あるパソコンの中でも最高速ではないでしょうか。ただし、テレコによっては使用できない場合もあります。筆者のテレコは専用テレコではないのですが、十分使えました。でも、デモテープはロードできなかったのも、互換性がシビアになるようです。だから、友達とのテ

ープのやりとりは600ボーで、自分だけの場合は2400ボーで行なえば良いと思います。

ケースの裏側に、スピードの切り換えスイッチがあって電源を入れた時のスピードを決めます(写真3)。もちろん、ソフトで切り換えができるので、いちいち裏のスイッチを切り換える必要はありません。しかも、ロードする時は、自動的に切り換わるようになっています。このため、テープフォーマットが異なるので、JR-100用のテープはロードできません。もしロードできたとしても、中間言語に変換してやらないとダメです。

また、JR-200では新たにリモート端子が付きまして。そして、OPEN, CLOSE, PRINT#, INPUT#で、テープにデータファイルを作ることができます。マニュアルには、さらにテープの始めにディレクトリ(カセットのインデックスのように、内容を書いたもの)を作るプログラムが載っています。

ファイルネームは、カナやグラフィック文字も使えますが、スペースや記号が使えません。スペースの代わりにアンダーラインを使えば良いでしょう。

MLOADは、ロードする場所を指定することもできるようにになりました。

また、2つのプログラムを結合するMERGEができます。しかし、追加するプログラムの行番号が、元のプログラムの行番号より大きくなければならないという条件が付きま。

よく使うサブルーチンは、大きめの行番号を付けてテープにストックしておいて、結合してプログラムを作ることができます。

## 7 マニュアル

JR-200は本格的ホームコンピュータを目指しているため、マニュアルが非常にていねいに書かれています。

JR-100の時もそうだったのですが、初心者の方でも読み進んでいくうちにBASICが十分理解できるようになっています。そのつど短いプログラム例が書いてあるので、それを入れていくうちに実力が付いてくるというわけです。中にはモルス符号を発するプログラムもあつたりして、楽しく取り組みます。

## 8 モニタ

マシン語や音譜データの入力のために、モニタ機能があります。決して十分とは言えませんが、D, M, Gの3つのサブコマンドがあって、一応メモリの内容を書き換えたり、マシン語の実行ができます。

サブコマンドは大文字でないとダメなので、少しめんどくさいです。Dコマンドは、チェックサムも表示してくれるればもっと良かったと思います。

☆ ☆ ☆

JR-200はとにかく機能がたくさんあって、特徴をすべて書ききれないくらいです。低価格機もここまで機能が充実してくると、もはや初心者向けとは呼べなくなってしまった感があります。さらに、RS-232Cやフロッピーディスクを装備すれば、JR-200の応用は大きく広がることでしょう。3インチのコンパクトフロッピーも開発されていることだし、これが接続される日が楽しみです。

(Kazuhiro Yamanishi)

1月号は、12月21日(火)発売 

●やさしく楽しいコンピュータバビコン 後藤 猛 著 定価2300円

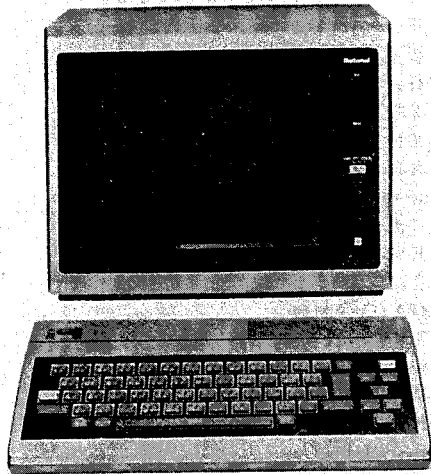
# NEC PC-6001 プログラミング教本



TEL 03-453-1201  
〒100 東京都港区芝2-23の13  
廣濟堂出版



# ナショナル JR-200



このBASICもナショナル独自のものです。何年間かマイクロソフトBASICのシェアが大きかっただけに、ここに来て国産のものが増えてきたことは注目し値するでしょう。

まず中間コードですが、第3表のようになっています。変数はキャラクタコードのまま、定数(数字)は\$F2+2バイト、\$F3+3バイト、\$F5+5バイト、\$F1+2バイト(16進数。\$12A0など)のように内部表現になっています。変数名は2文字の制限があるため、とても高速に動き同社の16ビットマイコンJB-3000(マイブレイン3000)とほとんどかわ

らぬ速さです。また、JR-200より速いマイコンもいくつかありますが、全体的に見るとやはり上方に位置しそうです。

また、予約語が128個以下しかないのに関数が\$FF+\$nnとなっているのは、将来、このBASICを拡張していわゆるレベル3BASICにする予定があるからかもしれません。代表的なレベル2BASICというのは、80H~FFHでステートメント(文)、コマンド、演算子、関数のすべてを表わすのが一般的だからです(メモリの節約から)。

変数の内部表現ですが、5バイトで表わされ、マイクロソフト社の80系(PC-6001など)と順序が逆、68系のもと同じ順序で並んでいます。5バイト、とは精度の問題だけですから、他の大半のものと似たようなものと考えられます。まず1バイト目が指数部で、最上位ビットを反転させた2の補数表現(2<sup>n</sup>のn)、2バイト目との間に仮想小数点があり、2バイト目以降が仮数部がきて、最上位に1が来るように最適化され、実質的に余ってしまう最上位ビット(2バイト目の最上位ビット)が正=0、負=1の仮数部符号を持たせ、最適化できない0だけは指数部を0にする、というものです。100などの小さな整数では2バイトか3バイトで済んでしまうものもあるため、プログラム中では\$F2+2バイト、または\$F3+3バイトという形も存在させることで省メモリ化しています。F1H+2バイトの16進定数は、H、Lの順になっています。

〈第3表〉National JR-200 BASIC中間コード表

| L/H | 8×     | 9×     | A×      | B×        | C×  | FF+8×  | FF+9×     | FF+A× |
|-----|--------|--------|---------|-----------|-----|--------|-----------|-------|
| ×0  |        | ON     | TEMPO   | CLS       | =   |        | VAL (     |       |
| ×1  | NEW    | IF     | CLEAR   | STOP      | >   | FRE (  | USR (     |       |
| ×2  | MON    | READ   | SAVE    | END       | +   | HPOS ( | MOD (     |       |
| ×3  | CONT   | PICK   | LOAD    | RETURN    | -   | VPOS ( | VARPTR (  |       |
| ×4  | FIND   | INPUT  | VERIFY  | RANDOMIZE | *   | RND (  | STICK (   |       |
| ×5  | LFIND  | OPEN   | MLOAD   | INITP     | /   | PEEK ( | CHR\$ (   |       |
| ×6  | DELETE | LET    | RUN     | SPC (     | ^   | ABS (  | HEX\$ (   |       |
| ×7  | AUTO   | COLOR  | BEEP    | TAB (     | NOT | INT (  | STR\$ (   |       |
| ×8  | LIST   | PLOT   | NEXT    | TO        | AND | SGN (  | LEFT\$ (  |       |
| ×9  | LLIST  | LOCATE | RESTORE | STEP      | OR  | SQR (  | RIGHT\$ ( |       |
| ×A  | MERGE  | SOUND  | PRINT   | THEN      |     | SIN (  | MID\$ (   |       |
| ×B  |        | POKE   | LPRINT  | EOF       |     | COS (  |           |       |
| ×C  | MSAVE  | GOTO   | CLOSE   | <>        |     | EXP (  |           |       |
| ×D  | PLAY   | GOSUB  | REM     | <=        |     | LOG (  |           |       |
| ×E  | DIM    | INPUT# | DATA    | >=        |     | ASC (  |           |       |
| ×F  | FOR    | PRINT# | HCOPY   | <         |     | LEN (  |           |       |

《第4表》JR-200内部処理ルーチン

| アドレス | 名 称                     | アドレス | 名 称                          | アドレス | 名 称                  |
|------|-------------------------|------|------------------------------|------|----------------------|
| A02D | END                     | AD81 | GOSUB                        | E922 | INPUT用1行入力(モニタ用兼用)   |
| A064 | プログラム入力                 | ADB3 | STOP                         | EA91 | MON                  |
| A069 | ダイレクトモード実行              | ADC2 | GOTO                         | EAF2 | モニタ内Gコマンド            |
| A0E1 | LET                     | ADF1 | CLEAR                        | EAF9 | モニタ内Dコマンド            |
| A113 | DIM                     | AE19 | ON                           | EB18 | モニタ内Mコマンド            |
| A253 | 変数アドレスGET               | AE53 | RESTORE                      | EBA5 | DEL処理                |
| A3EF | 総合的演算ルーチン               | AE88 | RANDOMIZE                    | EBCD | →処理                  |
| A46A | ストリング定数                 | AE97 | REM, DATA                    | EBD4 | ←G(ベル)処理             |
| A5AD | VARPTR                  | AF24 | RUN                          | EBD7 | ←V↓(コピー)処理           |
| A5E9 | MOD                     | AF42 | MERGE                        | EBDA | 1字プリント               |
| A5F2 | COS                     | AF6E | LOAD                         | EC66 | CR                   |
| A5F9 | SIN                     | AF9D | MLOAD                        | EC69 | ↓処理                  |
| A600 | EXP                     | AFB0 | SAVE                         | EC6E | CLS, ←L処理            |
| A607 | LOG                     | AFC4 | MSAVE                        | EC7A | ←K(HOME)処理           |
| A611 | SQR                     | AFED | VERIFY                       | EC80 | ↑処理                  |
| A618 | RND                     | B059 | INPUT#                       | EC82 | ←Z処理                 |
| A61F | INT                     | B0EB | PRINT#                       | EC90 | ←処理                  |
| A626 | STICK                   | B19B | OPEN                         | EC9F | RUBOUT(BS)処理         |
| A650 | CHR\$                   | B1F6 | CLOSE                        | ECA9 | (↑処理)                |
| A678 | ASC                     | B219 | SOUND                        | EDB1 | コマンド待ち用1行入力          |
| A68A | USR                     | B269 | TEMPO                        | EE56 | #用Ret処理              |
| A6CF | PEEK                    | B26F | PLAY                         | EF0A | X+AB→X(レジスタ加算)       |
| A6DD | ABS                     | B2ED | INPUT                        | EF56 | }スペースをカットする          |
| A6E5 | SGN                     | B332 | READ                         | EF57 |                      |
| A700 | HPOS                    | B40F | PLOT                         | EF8E | 変数名(1文字目)チェック        |
| A705 | VPOS                    | B4A8 | COLOR                        | EFE6 | 1行プリント               |
| A709 | LEN                     | B517 | LOCATE                       | F007 | BEEP                 |
| A710 | FRE                     | B52B | LPRINT                       | F042 | (←G処理)               |
| A733 | VAL                     | B52F | PRINT                        | F059 | +実行                  |
| A758 | HEX\$                   | B610 | PICK                         | F189 | -実行                  |
| A769 | STR\$                   | B61F | POKE                         | F1A5 | *実行                  |
| A787 | LEFT\$                  | B630 | IF                           | F28A | /実行                  |
| A78D | RIGHT\$                 | B63C | IF成立                         | F2C6 | ←実行                  |
| A7A2 | MID\$                   | B645 | 文字列→中間コード                    | F3EA | MOD実行                |
| A7ED | 演算子・関係子ジャンプテーブル(A808まで) | BD4E | 予約語テーブル(BEDDまで)              | F416 | COS実行                |
| A81B | 演算子・関係子優先順位テーブル(A828)   | BF62 | ステートメント・コマンドジャンプテーブル(BFCBまで) | F42B | SIN実行                |
| A8E3 | ←                       | BFFC | エラー(FCW)                     | F501 | EXP実行                |
| A8EA | *                       | BFFE | ホットスタート(FCW)                 | F55C | LOG実行                |
| A8F1 | /                       | E000 | ファイル操作                       | F623 | SQR実行                |
| A8F8 | +                       | E00F | ファイル操作ジャンプテーブル(FCW)(E021まで)  | F6B3 | ストリング加算実行            |
| A90E | -                       | E025 | MSAVE実行                      | F73D | INT実行                |
| A93B | =                       | E026 | SAVE実行                       | F74A | RND実行                |
| A948 | >                       | E042 | OPEN "0" 実行                  | F888 | HEX\$実行              |
| A950 | <                       | E132 | 出力ファイルのCLOSE実行               | F979 | 文字列→内部表現数値           |
| A956 | >=                      | E144 | 入力ファイルのCLOSE実行               | FBD0 | 数値内部表現→文字列           |
| A95C | <=                      | E14D | PRINT#実行                     | FED6 | エラーメッセージ(FF8Aまで)     |
| A964 | <>                      | E24E | LOAD実行                       | FF8B | Break in line        |
| A97B | AND                     | E250 | MLOAD実行                      | FF99 | JR BASIC 5.0~        |
| A989 | OR                      | E27D | VERIFY実行                     | FFDC | モニタコマンドテーブル(FCB+FCW) |
| A991 | NOT                     | E281 | OPEN "I" 実行                  | FFE9 | ← (モニタ用)             |
| A9C6 | DELETE                  | E388 | INPUT#実行                     | FFF8 | FFFFまで, CPUのベクトル     |
| AA2A | NEW                     | E3A1 | カセット1バイト読み込み                 |      |                      |
| AA3A | AUTO                    | E5D5 | HCOPY, (←Vキー)                |      |                      |
| AA99 | LFIND                   | E6AF | INITP                        |      |                      |
| AB40 | LLIST                   | E6E4 | コールドスタート                     |      |                      |
| AB49 | LIST                    | E8F9 | INPUT用1字入力(キーボード)            |      |                      |
| AC0B | NEXT                    |      |                              |      |                      |
| AC8D | FOR                     |      |                              |      |                      |
| AD59 | CONT                    |      |                              |      |                      |
| AD5C | RETURN                  |      |                              |      |                      |