

リスト

腕コンピュータ100%活用法

M159
10.2



はじめに

近年、ユーザのニーズの多様化に伴い、パソコンも多様な形態で開発が行われてきました。ディスクトップタイプのパソコンも、8ビット機から16ビット機へと、大容量化と高速化がはかられ、かつては大型計算機がこなしていたような仕事を遂行できるような機能の強化がはかられると共に、他方では、ハンドヘルドタイプ、ポケコンタイプのような、パソコンの小型化とその機能の充実がはかられてきました。

EPSON社のRC-20は、後者のひとつの典型といえるでしょう。RC-20は、腕時計に似た小さなボディの中に、8ビットCPU、ROM、RAM、KEY、液晶表示等を備えた立派なコンピュータです。

RC-20は、スケジュール・アラーム、メモバンク、ワールドタイム、簡易電卓を標準機能として備え、そのまで、パーソナル情報機器として使用することができます。しかし、RC-20の一番の特徴は、RS-232Cを介して、ユーザが作成したプログラムやデータを、パソコンとやりとりできることです。このことにより、ユーザは、RC-20に、自分のニーズに合わせて、様々な機能を持たせることができます。

本書は、ユーザが、RC-20をパソコンと接続して、RC-20に、自分の作ったプログラムを自由に実行させるための基礎的な手法を解説し、さらにRC-20で実行できる実用プログラム例を紹介しました。

筆者の浅学のために、舌足らずのところも多々あるかと思いますが、読者の皆さんのが、RC-20の様々な活用法を独自に開発されるのに、本書がお役に立てば幸いです。

第14章実用プログラム集では、清水和文、小野洋一各氏の御協力をあおぎました。また、様々な面で、EPSON社の御助力をいただきました。さらに、出版にあたり、技術評論社の足立幸雄氏、スタッフの皆さんには、大変お世話になりました。ここに記して、深甚の謝意を表します。

1985年3月

著者

85W56458

目次

1 こんな使い方ができる	5
1.1 時刻表示・カレンダー機能として使う	6
1.2 コンピュータとして使う	6
2 RC-20にマシン語プログラムを転送する	8
2.1 RS-232Cによる転送の規約	8
2.2 転送プログラムのフォーマット	9
2.3 プログラムの転送例	13
1. QC-10の場合	13
2. PC-8001/8001mk IIの場合	20
3. PC-8801/8801mk IIの場合	35
4. FM-7の場合	37
5. HC-20の場合	40
6. HC-40/80/88の場合	44
3 RC-20からのパソコンへのデータの転送	56
3.1 QC-10の場合	57
3.2 PC-8001/8001mk IIの場合	66
1. CP/Mの場合	66
2. DUADPCの場合	68
3.3 PC-8801/8801mk IIの場合	74
1. CP/Mの場合	74
2. DUAD88の場合	74
3.4 HC-40/80/88の場合	78
4 ハード構成とメモリマップ	84
5 ROM内サブルーチンの使い方	91
5.1 割込みおよびシステム処理サブルーチン	92
5.2 キー関係サブルーチン	93
5.3 入出力ポート制御サブルーチン	95
5.4 キャラクタ表示関係サブルーチン	97
5.5 演算データの型式と演算サブルーチン	99

目次

6 表示機能	102
7 入出力ポート機能	104
8 割込み機能	108
8.1 割込み要因	108
8.2 割込みの方法 (10HZ, 1HZ割込み)	108
8.3 特殊割込みの方法	111
9 浮動小数点演算における注意	113
9.1 データ補正サブルーチン (CORRECT)	113
9.2 データ補正サブルーチンの使用例	125
10 スケジューラ, メモ, ワールドタイムのデータの構造	131
10.1 ヘッダ・データの構造	131
10.2 スケジューラ・データの構造	132
10.3 メモ・データの構造	134
10.4 ワールドタイム・データの構造	135
10.5 転送上での注意	135
11 ダンプリストのプログラムの入力・転送	136
11.1 QC-10の場合	136
11.2 PC-8001/8001mkII, PC-8801/8801mkIIのCP/Mの場合	137
11.3 PC-8001/8001mkIIのDUADPCの場合 (N-BASIC)	140
11.4 PC-8801/8801mkII (N88-BASIC) の場合	142
11.5 HC-20の場合	143
11.6 HC-40/80/88のCP/Mの場合	143
11.7 FM-7の場合	146
12 実用マシン語プログラム集	147
卷末付録 ● RC-20とパソコンの接続ケーブル	202

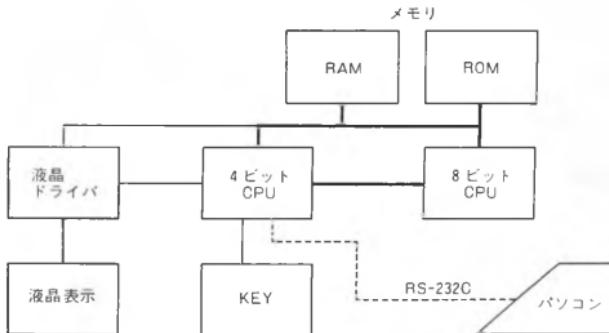
NUMBER

1

こんな使い方 ができる

図1-1
RC-20

EPSON のリストコンピュータ RC-20 は、図1-1にある縦42mm、横40mmの一見、腕時計のようなコンピュータです。外見は腕時計とさしてかわらぬこの内部には、一般的パソコンと同様なCPUを含むコンピュータ機能が組み込まれています。リストコンピュータ RC-20 の内部構成の概要を図1-2に、RC-20の機能概要を図1-3に示します。

図1-2
RC-20の内部構成図1-3
RC-20の機能概要

- (1) 時刻表示・カレンダー機能
 - ① 計時機能
 - ② 時報/アラーム機能
 - ③ オート・カレンダー機能
- (2) コンピュータ機能
 - ① 内蔵プログラムによるもの
 - 1) スケジュール・登録機能
 - 2) メモ機能

- 3) ワールドタイム機能
 ②ユーザプログラムによるもの
 マシン語ソフト機能
 (パソコンとの通信を利用)

1.1 時刻表示・カレンダー機能として使う

時刻表示・カレンダー機能としては、時間、分、秒、年、月、日、曜日の表示が行なえるほか、任意の時刻に時報やアラームを設定することができます。アラームは20秒間鳴り、スイッチによって鳴り止めを行なうことができます。

また、カレンダーは、月末日の設定が自動的に行なわれます。

1.2 コンピュータとして使う

リストコンピュータRC-20の一番の特長は、このコンピュータ機能にあるといえます。図1-2からもわかるようにZ-80とコンパチブルなCPUのほかに4ビットのサブCPUをもち、メモリもROM 8Kバイト、RAM2Kバイトをもち、外部のパソコンともRS-232Cを介してプログラムやデータをやりとりすることのできる立派なコンピュータです。

このコンピュータ機能を大別すると、内蔵プログラムによる機能とユーザプログラムによる機能とに分けることができます。

内蔵プログラムによる機能

これはプログラム自体は、あらかじめRC-20のROM内部に書き込まれていて、データは外部のパソコンを使って作成し、RC-20にRS-232Cを介して転送し記憶させておくほか、RC-20自体のキーを使って、編集、入力、削除等を行なうことができます。

内蔵プログラムには、次のものがあります。

①スケジュール・アラーム機能

年、月、日、曜日、時間、分等を指定して、用件の表示とアラームの種別を設定することができます。アラームの種別には、アラームを鳴らさない場合、毎日セットした時刻にアラームを鳴らす場合、毎曜セットした時刻にアラームを鳴らす場合等の機能が含まれています。コメント文の長さは外部のパソコンを使って任意に設定することができます。

②メモバンク機能

電話番号や時刻表、住所録等いろいろなデータを記憶させておいて表示させることができます。一度に表示できるのは28文字以内ですが、1個のレコードの大きさはパソコンにより255バイトまで任意の大きさにセットすることができ、記憶できるデータ全部の大きさは最大1856(740H)バイ

トまでです。これらのメモはRC-20のキーを使って自由に編集、削除、入力することができます。

③ワールドタイム機能

世界のいろいろな地点の時刻を瞬時に表示させることができます。

基本時計をベースにして、それに対するロンドン、ニューヨーク等いろいろな地点の時差データを、時間・分の指定で地名と一緒に書き込んでおくと、それらの地点の時刻を表示させることができます。地名は14文字まで設定できます。時差計算には、さらにサマータイムの計算機能も付加されています。

出荷時には、世界の主要8都市の時差データが入っています。

ユーザ・プログラムによる機能

「内蔵プログラムによる機能」で述べた3つの機能は、すべてRC-20のROMの中に組み込まれたプログラムによって行なわれるのですが、RC-20にはこのほかに、ユーザが自由に自分で作ったプログラムをRC-20上で実行させる機能があります。これによってRC-20の機能を、ユーザが自分のニーズに合わせて自由に拡張することができるのです。

とりわけ、RC-20の8ビットCPUはZ-80とコンパチブルなので、Z-80のマシン語で書いたプログラムを直接実行することができます。Z-80CPUとRS-232Cコネクタを持つパソコンさえあれば、そのパソコンでマシン語プログラムを作成して、これをRS-232Cを介してRC-20に転送すれば、ユーザが自分で作ったプログラムをRC-20で実行することができるのです。また、ユーザがRC-20用のプログラムを作成するのに有効なROM内のサブルーチンの多くが後述のようにユーザに公開されています。

さらにRC-20内のプログラムやデータを、RS-232Cを介してパソコンに転送することもできます。

出荷時にはこのユーザプログラムとしては、簡易電卓機能プログラムが入っています。

NUMBER

2

RC-20にマシン語 プログラムを転送する

パソコンで、マシン語プログラムを作成し、それを RC-20 に転送する方法について説明します。

2.1 RS-232Cによる転送の規約

パソコンから RC-20 にプログラムを転送するには、別売の RS-232C 用のケーブルで両者を接続します。パソコン側には、当然 RS-232C の端子がついている必要があります。使用する信号線は図2-1のよう TXD と GND だけです。

転送データは表2-1のようにパリティなしの 8 ビット、ストップビットは 2 ビット、転送速度 2400 ポーの非同期通信によります。

データの転送は、ハンドシェイクなしのいわゆる垂れ流しですので、転送の終了は①RC-20 が受信できる最大長のデータを受け取った時、あるいは②送信データが 0.6 秒以上ない場合としています。

転送エラーのチェックは、RC-20 側でチェックサムを行ない(チェックサムを指定した時)、エラーがあれば、エラーメッセージが表示されます。

なお、RC-20 側ではデータ転送中は 2 秒ごとに、「ピッ」という音を鳴らします。

図2-1 RS-232Cによるパソコンとの接続

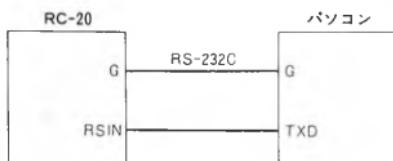


表2-1 転送データ

スタートビット	1 ビット
データビット	8 ビット
ストップビット	2 ビット
パリティビット	なし
転送速度	2400 ポー

2.2

転送プログラムのフォーマット

転送すべきマシン語プログラムは、図2-2のフォーマットを持ち、RC-20のRAMのB800Hより書き込まれます。書き込み可能な最大アドレスはBF3FHですので、最大長は1856バイト(740Hバイト)となります。

プログラムは1個のファイルとみなされ、1856バイト以内であれば、複数個のファイルを(最大10ファイルまで)転送することができます。プログラム・ファイルは図2-2に示すように先頭にヘッディング・データが配置され、その後にマシン語プログラムよりなる主データが配置されます。

図2-2 プログラムファイルの構造

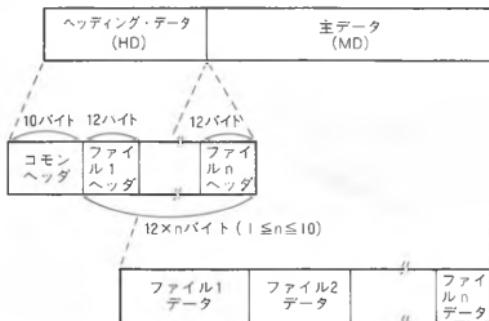


図2-3 ファイルデータの構造



ヘッディング・データは、さらにコモンヘッダ(10バイト)と各ファイルごとのファイルヘッダ(12バイト)に分かれています。コモンヘッダとファイルヘッダの仕様を図2-4に示します。

転送エラーをチェックするため、チェックサムフラグのセットとチェックサムデータの記入が必要です。チェックサムデータは、ヘッダ・データと主データ全部を加算した時にゼロとなるような値(1バイト)をとります。コモンヘッダ内のデータ長は、ヘッダ・データと主データ全部のバイト数を表わしています。コモンヘッダは、RC-20のRAMのアドレスB800H～B809Hにロードされます。

次にファイルヘッダの構造を図2-5に示します。

ファイルヘッダは1個が12バイトの大きさで、ファイルの数だけ作成され、RC-20のRAM

のアドレスB80AH以降に配置されます。ファイルの先頭アドレスは、ヘッダの先頭アドレス(B800H)から、当該のマシン語プログラムデータの先頭までのバイト数を表わし、ファイル終了アドレスはヘッダの先頭アドレス(B800H)から当該のマシン語プログラムデータの終りまでのバイト数を表わします。

マシン語プログラムデータの構造を図2-6に示します。

先頭の1バイトはプログラム種別を表わし、00Hが入ります。続く14バイトは当該プログラムのプログラム名で左につめて文字データが書かれ、データのないところには空白(20H)が書かれます。マシン語プログラム自体はその後ろに配置されます。

図2-4 コモンヘッダの詳細

← 10バイト →					
フォーマットID	チェックサムフラグ	チェックサムデータ	データ長 (下位)	ヘッダ長 (上位)	システム予約

位置 (バイト)	名 称	内 容			RC-20での対応		
0	フォーマットID	01H			01H以外を検出するとエラーメッセージを表示する。		
1	チェックサムフラグ	00H 01H	チェックサムデータ無効 チェックサムデータ有効		01Hの時は、チェックサムを行なう。		
2	チェックサムデータ	チェックサムデータ(00H~FFH), HD, MDすべてのデータを加算して、00Hになるように設定する。					
3, 4	データ長	HD, MD を含めた全データ長のバイト数を示す。2バイトで、3バイト目は、下位部分を、4バイト目は、上位部分を表わす。			最大740Hとする。		
5	ヘッダ長	HDのバイト長で、コモンヘッダとファイルヘッダの長さを加えたもの(最大130バイト)。					
6 ~ 9	システム拡張用	00, 00, 00, 00H					

図2-5 ファイルヘッダの構造

← 12バイト →					
ファイル種別	ファイル先頭アドレス	ファイル終了アドレス	レコード長	レコード数	システム予約

位 置	名 称	内 容
0	ファイル種別	02H マシン語ソフトを表わす。
1, 2	ファイル先頭アドレス	各ファイルのデータの先頭アドレス(HDの先頭からの相対アドレス)、1バイト目は下位部分、2バイト目は、上位部分を表わす。
3, 4	ファイル終了アドレス	各ファイルデータの終了アドレス(HDの先頭からの相対アドレス)、3バイト目は下位部分、4バイト目は、上位部分を表わす。
5	レコード長	マシン語ソフトでは何でもよい。
6 ~ 7	レコード数	マシン語ソフトでは何でもよい。
8 ~ 11	システム予約	00, 00, 00, 00H

図2-6 マシン語プログラムデータの構造



図2-7

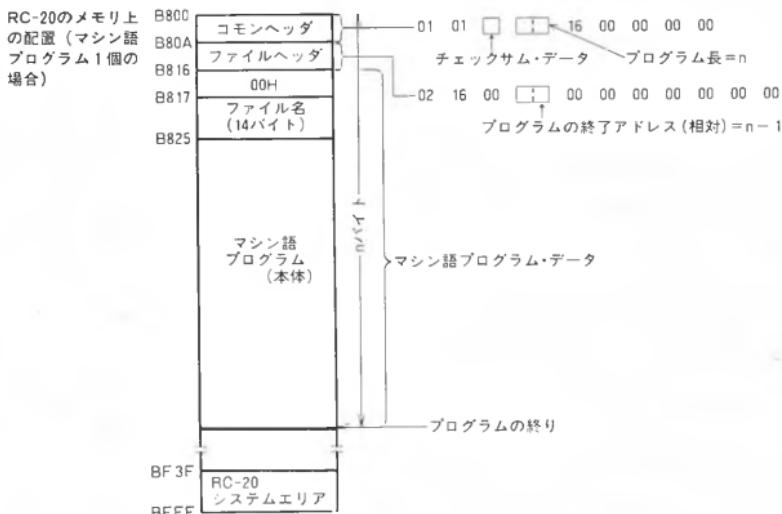


図2-8 マシン語プログラムの構造(1)

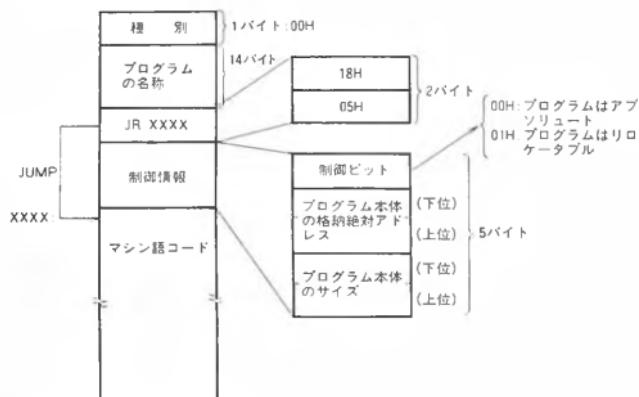
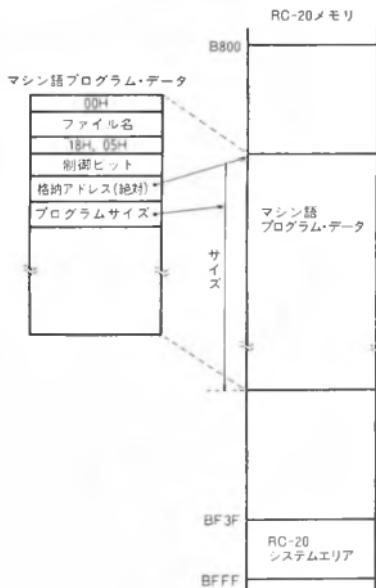


図2-9 マシン語プログラムの構造(2)



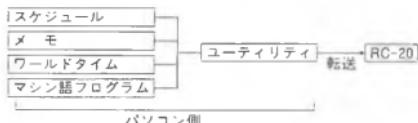
先にも述べたようにマシン語プログラム本体の最大は、 $1856 - 22 - 15 = 1819$ バイトですから、少しばかり複雑なプログラムを作成すると、プログラム1個でユーザ用RAMエリアを占拠してしまうでしょう。

簡単な例として、マシン語プログラムが1個だけの場合のファイル構造とメモリ上への配置を図2-7に示します(注)。

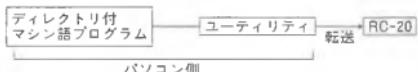
なお、EPSON社では今後、提供する予定のユーティリティ等でも処理できるようにするために、ユーザのマシン語プログラム本体に、さらに図2-8、2-9のような5バイトの制御情報を設定することを推めています。図2-8のマシン語プログラム本体の先頭の2バイト(18H, 05H)はプログラムの制御を、制御情報の次に位置するマシン語コードに移すためのジャンプ命令です。本書でもRC-20用のマシン語プログラムは特にことわらない限り図2-8、2-9の形式に従うこととします。

注) マシン語プログラムをRC-20に転送する方法としては、次のようなものが考えられますが、本書では②の方法を採用しています。

- ① ユーティリティにより、転送用データをフォーマットしてから転送を行う。



- ② ディレクトリを含んだ型でデータを作成しておき、そのまま転送する。スケジュールなど、他のデータとの混在はしない。



2.3 プログラムの転送例

それでは、実際にパソコンで作成したプログラムをパソコンからRS-232Cを介して、RC-20に転送して実行させる例をいくつか紹介しましょう。わかりやすくするために、転送するファイルはマシン語のプログラムファイル1個だけとします。

1. QC-10の場合

QC-10のCP/M上で作成したRC-20用のマシン語プログラムをJBASIC上で作成した転送プログラム(TRANS.BAS)によってRC-20に転送します。

初めに簡単なグラフィック表示のプログラムを作成して転送しましょう。このプログラムは犬の顔をRC-20上にグラフィック表示するもので、任意のキーを押すと計時モードにもどります。ブ

ログラム(GRAPH)はリスト1に示します。

このソースリストをエディタ(ED)で作成(GRAPH.ASM)し、アセンブラー(ASM)でアセンブルしてHEX型式のファイル(GRAPH.HEX)を作成します。

次にJBASICを実行させます。そして転送プログラム(TRANS.BAS)を実行させます。転送プログラムをリスト2に示します。このプログラムはHEX型式のマシン語ファイルをディスクから入力して、これにヘッダ部分を付け加え、チェックサムを計算した後、RS-232Cを介してRC-20に転送するプログラムです。

転送プログラムを実行する前に、QC-10とRC-20とを付属のRS-232C用のケーブルで接続しておきます。

さて、JBASIC上で転送プログラム(TRANS)を実行させると"filename?"と入力ファイル名を聞いてきますので、転送すべきHEX型式のマシン語プログラムファイルの名前を入力します。ここでは"GRAPH"を入力します。転送プログラムは必要な計算を行なった後、"RC-20 READY OK?"と問い合わせてきますので、RC-20のMODEスイッチを押して「LOAD MODE」にし、「CR」キーを押して"READY OK?"を表示させます。次いで「OK?」の部分を押すと、"LOADING NOW"の表示が出てRC-20の入力準備は完了しますので、QC-10のリターンキーを押します。QC-10からRC-20への転送が行なわれ、RC-20から「ピッ」という音が発生します。

転送が終了するとRC-20はチェックサムを行ない、エラーがなければ「PROGRAM RUN?」のモードに入ります。「CR」キーを押すとプログラムの番号が表示されます、1個しか転送していませんので、プログラムの番号は1だけです。もう一度「CR」キーを押しますと、今転送したプログラムが実行され、画面に絵が出てきます。

転送時にチェックサム等のエラーがあると、RC-20に「ERROR」が表示されますので、転送プログラムに間違いがないか、チェックしてください。

転送プログラム(TRANS)で取り扱うマシン語プログラムはヘッダ部を除いた本体部分だけを、先頭がB816Hからはじまるように作成しておきます。コモンヘッダ部とファイルヘッダ部は転送プログラムが自動的に作成します。

RC-20用のマシン語プログラムの例として音楽演奏プログラム(KOHAN)をリスト3に示します。使用方法は、先の"GRAPH"と同様です。

リスト1 RC-20用マシン語プログラム(例1)

```
; $GRAPHIC$  
;  
;  
;  
B816 =      PORG   EQU    0B816H  
BFBA =      VRAMP   EQU    0BFBABH  
002A =      PAGEL   EQU    2AH  
411C =      DISP    EQU    411CH  
41CB =      STBY    EQU    41CBH  
409A =      Z80END  EQU    409AH
```

```

;          ;      ORG      PORG
;
MAIN:
B816 00      DB      0
B817 4752415048   DB      'GRAPH PROGRAM '
B825 1805     DB      18H,ENTRY-$-1 ;JR ENTRY
B827 00      DB      0
B828 1688     DW      MAIN
B82A 1601     DW      PRGEND-MAIN
;
ENTRY:
B82C 218388   LXI    H,MDATA
B82F 11BABF   LXI    D,VRAMP
B832 012A00   LXI    B,PAGEL
B835 E5      PUSH   H
B836 D5      PUSH   D
B837 C5      PUSH   B
B838 CD7888   CALL   LDIR
;
B83B 3E01     MVI    A,1
B83D 1E40     MVI    E,40H
B83F CD1C41   CALL   DISP
B842 C1      POP    B
B843 D1      POP    D
B844 E1      POP    H
B845 09      DAD    B
B846 E5      PUSH   H
B847 D5      PUSH   D
B848 C5      PUSH   B
B849 CD7888   CALL   LDIR
B84C 3E01     MVI    A,1
B84E 1E00     MVI    E,0
B850 CD1C41   CALL   DISP
B853 C1      POP    B
B854 D1      POP    D
B855 E1      POP    H
B856 09      DAD    B
B857 E5      PUSH   H
B858 D5      PUSH   D
B859 C5      PUSH   B
B85A CD7888   CALL   LDIR
B85D 3E02     MVI    A,2
B85F 1E00     MVI    E,0
B861 CD1C41   CALL   DISP
B864 C1      POP    B
B865 D1      POP    D

```

```

B866 E1      POP   H
B867 09      DAD   B
B868 C07888    CALL  LDIR
B86B 3E02      MVI   A,2
B86D 1E40      MVI   E,40H
B86F CD1C41    CALL  DISP
B872 CDCB41    CALL  STBY
B875 C39A40    JMP   Z80END
;
B878 7E      LDIR: MOV   A,M
B879 12      STAX  D
B87A 23      INX   H
B87B 13      INX   D
B87C 0B      DCX   B
B87D 78      MOV   A,B
B87E B1      ORA   C
B87F C8      RZ
B880 C37888    JMP   LDIR
;
MDATA:
B883 0000000000 DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
B88D 0102070A0E DB  1,2,7,0AH,0EH,18H,23H
B894 3434232020 DB  34H,34H,23H,20H,20H,23H,34H,34H
B89C 231B0E0A07 DB  23H,18H,0EH,0AH,07H,02H,1,0,0,0
B8A6 0000000000 DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
B8B2 0F3A6AAABF DB  0FH,3AH,6AH,0AAH,0BFH,0AOH,0EOH
B8B9 00000415CE DB  0,0,4,15H,0CEH,20H,04H,0CEH
BBC1 0E0ECEA420 DB  0EH,0EH,0CEH,0A4H,20H,0CEH,15H
BBCB 040000E0 DB  4,0,0,0E0H
BBC2 A0BFAA6A3A DB  0A0H,0BFH,0AAH,6AH,03AH,0FH
BBD2 0000000000 DB  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
BBD6 A0A0B07008 DB  0AOH,0AOH,0BOH,70H,8,8,4,4,4
BBE7 82FB848CBC DB  82H,0FBH,84H,BCH,BCH,BCH,BCH,84H,0FBH,82H
BBF1 040404080B DB  4,4,4,B,8,70H,0BOH,0AOH,0AOH,0COH
BBFB 8000000000 DB  80H,0,0,0,0,0,0,0,0,0,0,1CH,22H
B908 2640464046 DB  26H,40H,46H,40H,46H,42H,42H,0BAH
B910 82B2020404 DB  82H,82H,2,4,4,4,4,4,4,2,92H,0BAH
B91C 9E7A524640 DB  9EH,7AH,52H,46H,40H,46H,40H,26H,22H
B925 1C00000000 DB  1CH,0,0,0,0
B92A 0000      DB  0,0
;
;
;
;
PRGEND:
B92C          END

```

リスト2 転送プログラム(TRANS.BAS)

```
5 REM $trans$
10 REM transfer program(QC-10)
20 DIM TXT(30)
25 DIM CHEE$(100)
30 CHKS=0
50 K=0
90 INPUT "file name";F$
91 LNF=LEN(F$):IF LNF=0 THEN GOTO 90
93 IF LNF > 8 THEN F$=MID$(F$,1,8)
95 F#=F$+".hex"
98 PRINT "file name=";F$
100 OPEN "I",#1,F$
150 IF EOF(1) THEN GOTO 1000
200 INPUT# 1,GPRO$
350 S$=MID$(GPRO$,2,2)
360 LN=VAL("H"+S$")
365 IF LN=0 THEN GOTO 1000
390 TXT0=0
395 J=0
396 CHE$=""
400 FOR I=1 TO LN
410 TXT$=MID$(GPRO$,10+J,2)
420 J=J+2
440 TXT(I)=VAL("H"+TXT$)
450 CHE$=CHE$+CHR$(TXT(I))
460 TXT0=TXT0+TXT(I)
490 NEXT
510 K=K+1
520 CHEE$(K)=CHE$
530 CHKS=CHKS+TXT0
540 GOTO 200
550 CLOSE
560 GOSUB 4000
570 A$=CHKS-INT(CHKS/256)*256
580 A$=MKI$(0-A$)
590 MID$(CHEE$(0),3,1)=LEFT$(A$,1)
1900 INPUT "#rc-20 ready ok";P$
2000 OPEN "O",#3,"COM0:(CBN3F)"
2100 FOR S=0 TO K
2200 PRINT #3,CHEE$(S);
2300 NEXT
2300 CLOSE
3100 END
4000 CHD$=""
4010 CHD$=CHD$+CHR$(H1)
```

```
4020 CHD$=CHD$+CHR$(&H1)
4030 CHD$=CHD$+CHR$(&H0)
4040 CHD$=CHD$+CHR$(&H0)+CHR$(&H0)
4050 CHD$=CHD$+CHR$(&H16)
4060 CHD$=CHD$+CHR$(&H0)+CHR$(&H0)+CHR$(&H0)+CHR$(&H0)
4070 REM
4100 FHD$=**
4110 FHD$=FHD$+CHR$(&H2)
4120 FHD$=FHD$+CHR$(&H16)+CHR$(&H0)
4130 FHD$=FHD$+CHR$(&H0)+CHR$(&H0)
4140 FHD$=FHD$+CHR$(&H0)
4150 FHD$=FHD$+CHR$(&H0)+CHR$(&H0)
4160 FHD$=FHD$+CHR$(&H0)+CHR$(&H0)+CHR$(&H0)+CHR$(&H0)
4170 CHEE$(0)=CHD$+FHD$
4200 REM
4210 ZL=0
4220 FOR S=0 TO K
4230 ZL=ZL+LEN(CHEE$(S))
4240 NEXT
4250 A=ZL:ZL=ZL MOD 256:ZH=INT(A/256)
4260 MID$(CHEE$(0),4,1)=CHR$(ZL):MID$(CHEE$(0),5,1)=CHR$(ZH)
4270 MID$(CHEE$(0),14,1)=CHR$(ZL):MID$(CHEE$(0),15,1)=CHR$(ZH)
4300 REM
4310 FOR I=1 TO 22
4320 TXT$=MID$(CHEE$(0),I,1)
4330 CHKS=CHKS+ASC(TXT$)
4340 NEXT
4400 RETURN
9999 END
```

リスト3 RC-20用マシン語プログラム(例2)

```

;      $KOHAN$
;
ORG    0B816H
;
ZBOEND EQU    409AH
SAVER  EQU    40B6H
LOADR  EQU    40C6H
MUSIC   EQU    42AEH
;
MAIN:
B816 00      DB     0
B817 4D55534943 DB     'MUSIC     PLAY'
;
B825 1805    DB     1BH,ENTRY-$-1 ;JR ENTRY
B827 00      DB     0
B828 1688    DW     MAIN
B82A 8300    DW     PRGEND-MAIN
;
ENTRY:
B82C 2144B8    LXI   H,DATA
LOOP:
B82F 7E      MOV   A,M
B830 FEPF    CPI   OFFH
B832 CA9A40    JZ    ZBOEND
B835 23      INX   H
B836 46      MOV   B,M
B837 23      INX   H
B838 CDB640    CALL  SAVER
B839 C0AE42    CALL  MUSIC
B83E CDC640    CALL  LOADR
B841 C32FB8    JMP   LOOP
;
DATA:
B844 0C060C020C DB    12,6,12,2,12,6,14,2,16,6,16,2
B850 100610020E DB    16,6,16,2,14,6,12,2,14,6,16,2
B85C 0C060C0207 DB    12,6,12,2,7,8,16,6,16,2,16,6,17,2,19,6,19,2
B86E 1306130211 DB    19,6,19,2,17,6,16,2,17,6,19,2
B87A 100B190213 DB    16,11,25,2,19,2,16,11,25,2,19,2
B886 100B190213 DB    16,11,25,2,19,2,16,6,19,2,16,6
B892 1302100B19 DB    19,2,16,11,25,4,0FFH
;
PRGEND:
;
END
B899

```

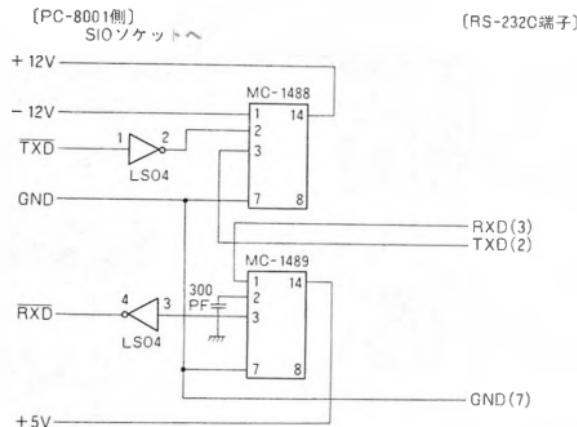
2. PC-8001/8001mk IIの場合

PC-8001にはRS-232Cがオプションになっていますので、オプションパーツ(PC-8062)を取り付ける必要がありますが、簡単に自作することもできます。回路図を図2-10に示します。

PC-8001mkIIにはRS-232C用端子がついていますので、直接これを使用します。

ポートレイトは16分間で2400ポーになるように、PC-8001では内部のジャンパーで、8001mkIIでは後面のジャンバースイッチでセットします。転送方法としては、CP/Mを使用する方法とそれ以外の方法を考えます。なお、PC-8001mkIIは、N-BASICモードで使用します。

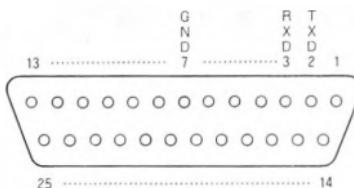
図2-10 PC-8001用RS-232Cケーブルユニット自作例



(アダプタ回路図)

GND	1	16	GND
TXD	2	15	-12V
RXD	3	14	GND
RTS	4	13	+12V
CTS	5	12	GND
DSR	6	11	+5V
CD	7	10	GND
DTR	8	9	

(PC-8001のSIOソケットの端子)



(RS-232Cコネクタピン配置)
(コネクタ前面より)

CP/Mを使う

PC-8001では、PC-8011かPC-8012またはマルチカード等が必要になります。

転送プログラムはマシン語で作成します。QC-10の場合と同様にHEX型式のRC-20用マシン語プログラムファイルを入力し、ヘッダ部等を作成し、RS-232Cを介してRC-20へ転送を行ないます。転送プログラムをリスト4に示します(TRANS)。転送すべきRC-20用のマシン語プログラムの作り方はQC-10の場合と同じです。ここでは例としてリスト1の"GRAPH"を使用してみます。転送プログラムの実行は次のように行ないます。

A>TRANS GRAPH.HEX

パラメータとして、転送すべきRC-20用マシン語プログラムのHEX型式ファイル名を指定します。".HEX"を必ず入れてください。

転送プログラムが実行されると"RC-20 READY OK?"とメッセージが表示されますので、RC-20を「LOAD」モードにし"LOADING NOW"を表示させたら、PC-8001/8001mkIIのリターンキーを押してください。「ピッ」という音がRC-20から発生してプログラムの転送が行なわれます。

リスト4 PC-8001/8001mkII用転送プログラム(TRANS)

```
;          tTRANS$  
;          FOR PC-8001/PC-8001 MK2  
;  
0100      ORG    100H  
;  
0005 =    B905  EQU    5  
0000 =    B00T  EQU    0  
005C =    FCB   EQU    005CH  
0080 =    BUFR  EQU    0080H  
00CE =    MODE  EQU    0CEH  ;$16  
0020 =    PORT0 EQU    20H  
0021 =    PORT1 EQU    PORT0+1  
;  
;  
0100 21CF12      LXI    H,AREA  
0103 22CB02      SHLD   BPOINT  
0106 CDA402      CALL   TDATA  
;  
0109 3A8000      LDA    BUFR  
010C 11D101      LXI    D,MSERRO  
010F B7          ORA    A  
0110 CAC501      JZ     EXIT  
;  
OPEN FILE  
0113 0EOF        MVI    C,15  
0115 115C00      LXI    D,FCB
```

```

0118 CD0500      CALL    BDOS
011B FEFF      CPI     OFFH
011D 113902      LXI     D,MSOPNER
0120 CAC501      JZ      EXIT
;
LOOP0:
0123 CD5202      CALL    READF
0126 C24701      JNZ    EOF
0129 2ACB02      LHLD   BPOINT
012C EB          XCHG
012D 218000      LXI    H,BUFFR
0130 0EB0          MVI   C,128
LOOP1:
0132 7E          MOV    A,M
0133 12          STAX   D
0134 23          INX    H
0135 13          INX    D
0136 0D          DCR    C
0137 C23201      JNZ    LOOP1
013A 018000      LXI    B,128
013D 2ACB02      LHLD   BPOINT
0140 09          DAD    B
0141 22CB02      SHLD   BPOINT
0144 C32301      JMP    LOOP0
;
EOF:
0147 0E10          MVI   C,16 ;FILE CLOSE
0149 115C00      LXI    D,FCB
014C CD0500      CALL   BDOS
;
014F 21CF12      LXI    H,AREA
0152 11E502      LXI    D,AREA2+22
TLOOP0:
0155 7E          MOV    A,M
0156 FE3A          CPI   ';'
0158 C2CB01      JNZ    ERROR1
015B 23          INX    H
015C CD0502      CALL   THEX
015F 47          MOV    B,A
0160 87          ORA    A
0161 CA8601      JZ     CNVEND
0164 C5          PUSH   B
0165 E5          PUSH   H
0166 2ACD02      LHLD   COUNT
0169 0600          MVI   B,0
016B 4F          MOV    C,A
016C 09          DAD    B
016D 22CD02      SHLD   COUNT

```

0170 E1	POP	H
0171 010600	LXI	B,B
0174 09	DAD	B
0175 C1	POP	B
TLOOP1:		
0176 CD0502	CALL	THEX
0179 12	STAX	D
017A 13	INX	D
017B 05	DCR	B
017C C27601	JNZ	TLOOP1
017F 23	INX	H
0180 23	INX	H
0181 23	INX	H
0182 23	INX	H
0183 C35501	JMP	TLOOP0
CNVEND:		
;		
0186 2ACD02	LHLD	COUNT
0189 011600	LXI	B,22
018C 09	DAD	B
018D 22CD02	SHLD	COUNT
0190 22D202	SHLD	AREA2+3
0193 22D02	SHLD	AREA2+13
;		
0196 1E00	MVI	E,0
0198 2ACD02	LHLD	COUNT
019B E5	PUSH	H
019C C1	POP	B
019D 21CF02	LXI	H,AREA2
SLOOP:		
01A0 7B	MOV	A,E
01A1 B6	ADD	M
01A2 23	INX	H
01A3 0B	DCX	B
01A4 5F	MOV	E,A
01A5 7B	MOV	A,B
01A6 B1	ORA	C
01A7 C2A001	JNZ	SLOOP
01AA 7B	MOV	A,E
01AB 2F	CMA	
01AC C601	ADI	I
;		
01AE 32D102	STA	AREA2+2
;		
;		
01B1 119102	LXI	D,MS000
01B4 CD2202	CALL	PRMSG
01B7 0E01	MVI	C,I

```

01B9 CD0500      CALL    BDOS
;
01BC CD5C02      CALL    SINIT
01BF CD7C02      CALL    SIO
;
01C2 112702      LXI    D,DONMSG
EXIT:
01C5 CD2202      CALL    PRMSG
01CB C30000      JMP    BOOT
;
ERROR1:
01CB 11E701      LXI    D,MSERR1
01CE C3C501      JMP    EXIT
MSERR0:
01D1 202A4E4F20  DB     'NO INPUT FILE NAME$'
MSERR1:
01E7 202A494E50  DB     'INPUT FILE IS NOT HEX FILE.', '$'
;
THEX:
0205 7E          MOV    A,M
0206 CD1A02      CALL   HEX0
0209 17          RAL
020A 17          RAL
020B 17          RAL
020C 17          RAL
020D E6F0          ANI   0FOH
020F 4F          MOV    C,A
0210 23          INX
0211 7E          MOV    A,M
0212 CD1A02      CALL   HEX0
0215 E60F          ANI   OFH
0217 B1          ORA    C
0218 23          INX
0219 C9          RET
;
HEX0:
021A D630      SUI    '0'
021C FEOA      CPI    10
021E D8          RC
021F D607      SUI    7
0221 C9          RET
;
;
PRMSG:
0222 0E09      MVI    C,9
0224 C30500      JMP    BDOS
;
DONMSG:
0227 0D0A202A4A  DB     13,10,'JOB COMPLETE$', '$'

```

```

;
MSOPNER:
0239 202A494E50      DB      ' $INPUT FILE OPEN ERROR$', '$'
;
READF:
0252 0E14      MVI    C,20
0254 115C00    LXI    D,FCB
0257 CD0500    CALL   BDOS
025A B7      ORA    A
025B C9      RET

;
;
;
SINIT:
025C 3E00      MVI    A,00H
025E D321    OUT    PORT1
0260 D321    OUT    PORT1
0262 D321    OUT    PORT1
0264 3E40      MVI    A,40H
0266 D321    OUT    PORT1
0268 3ECE      MVI    A,MODE
026A D321    OUT    PORT1
026C 3E15      MVI    A,15H
026E D321    OUT    PORT1
0270 C9      RET

;
OUTPUT:
0271 F5      PUSH   PSW
WAIT:
0272 DB21      IN     PORT1
0274 1F      RAR
0275 D27202   JNC    WAIT
0278 F1      POP    PSW
0279 D320    OUT    PORT0
027B C9      RET

;
;
;
SIO:
027C 2ACD02   LHLD   COUNT
027F E5      PUSH   H
0280 C1      POP    B
0281 21CF02   LXI    H,AREA2
SIO2:
0284 7E      MOV    A,M
0285 CD7102   CALL   OUTPUT
0288 23      INX    H

```

```

0289 0B      DCX    B
028A 78      MOV    A,B
028B B1      ORA    C
028C C28402   JNZ    $102
028F C9      RET

;
;

0290 C9      RET

;
;

MS000:
0291 202A52432D DB     * #RC-20 READY OK? *,*$*
;
TDATA:
02A4 21B502   LXI    H,DATA
02A7 11CF02   LXI    D,AREA2
02AA 0E16   MVI    C,22
TLOOP:
02AC 7E      MOV    A,M
02AD 12      STAX   D
02AE 23      INX    H
02AF 13      INX    D
02B0 0D      DCR    C
02B1 C2AC02   JNZ    TLOOP
02B4 C9      RET

;
DATA:
02B5 01      DB     1
02B6 01      DB     1
02B7 00      DB     0
02BB 0000    DW     0
02BA 16      DB     22
02BB 00000000 DB     0,0,0,0
;
02BF 02      DB     2
02C0 1600    DW     22
02C2 0000    DW     0
02C4 00      DB     0
02C5 0000    DW     0
02C7 00000000 DB     0,0,0,0
;
BPOINT:
02CB 0000    DW     0
COUNT:
02CD 0000    DW     0
;
AREA2:
02CF          DS     0
12CF =       AREA   EQU     AREA2+1000H
;
02CF          END

```

DUAD-PCを使う

CP/Mによらない方法の一つとして DUAD-PC の場合をとりあげてみます。DUAD-PC は使いやすいディスク使用のエディタ/アセンブラーを含んでいます。PC-8001 でも RAM32K の拡張を必要としません。ディスクがつながれば、OK です。

DUAD-PC では、マシン語ファイルを N-BASIC のファイルと一緒にディスク上に格納できます。したがって、転送プログラムはできるだけ BASIC(RS-232C 転送の部分はマシン語)で記述し、RC-20 に転送するプログラムはマシン語で作成してディスクに格納することにします。

転送プログラムをリスト 5 に示します。DATA 文はマシン語の部分で、そのソースリストをリスト 6 に示します。転送プログラムは、転送すべきマシン語プログラムが PC-8001/8001mkII のメモリの B816H から始まるエリアに置かれているものとして、それを RC-20 に転送します。

転送するマシン語プログラム例をリスト 7 とリスト 8 に示します。これらは先に述べたリスト 1 およびリスト 3 の内容は同一です。

転送すべきプログラムは DUAD-PC のエディタでソースプログラムを作成し、アセンブラーでアセンブルして HEX 型式(オブジェクト・プログラム)でディスクに格納します。

転送は次のように行ないます。

- ① DUAD-PC を起動すると、画面にメニューがあらわれます。「5」の LOADER を指定すると LOADER がロードされ、画面に "filename?" の表示が出ます。転送すべきマシン語プログラム(HEX 型式)のファイル名を指定します。"KOHAN"(リスト 7)を例にすれば、KOHAN.HEX を指定します。LOADER は KOHAN.HEX をディスクからメモリ上にロードし、"B816-B897" という表示が画面に表示され、次いで "OK" の表示が出て BASIC モードにもどります。KOHAN.HEX は B816H~B897H にロードされたわけです。
- ② 転送プログラム TRANS(リスト 5)を RUN "TRANS" で実行します。転送プログラム(TRANS)は、先に LOADER でロードした RC-20 用プログラム(たとえば KOHAN.HEX)にコモンヘッダとファイルヘッダを添付し、チェックサムを計算すると "RC-20 READY OK?" のメッセージを出します。
- ③ RC-20 の MODE スイッチで、RC-20 を「LOAD」モードにし、C/R キーとパネルキーを使って「LOADING NOW」の状態にします。次いで、PC のリターンキーを押すと、転送が始まります。
- ④ 転送が終了すると、PC 側では "*JOB COMPLETE*" のメッセージが出力され、RC-20 側はエラーがなければ、自動的に「PROGRAM RUN?」のモードに入ります。

リスト 5 PC-8001/8001mkII用転送プログラム(N-BASIC)

```
10 REM ***trans***  
200 CLEAR 100,&HFFFF  
210 DEF USR=&H9000  
300 FOR AD=&H9000 TO &H90B7  
810 READ DA$  
820 POKE AD,VAL("&h"+DA$)  
830 NEXT  
900 INPUT "#rc-20 ready ok";X$  
920 A=USR(1)  
950 PRINT "job complete!"  
1000 END  
1100 DATA CD,5B,90,CD,3E,90,CD,0D,90,CD,2C,90,C9,3E,00,D3  
1101 DATA 21,D3,21,D3,21,3E,40,D3,21,3E,CE,D3,21,3E,15,D3  
1102 DATA 21,C9,F5,DB,21,1F,30,FB,F1,D3,20,C9,ED,4B,03,B8  
1103 DATA 21,06,B8,7E,CD,22,90,23,08,78,B1,20,F6,C9,1E,00  
1104 DATA ED,B8,21,00,B8,7B,B6,23,0B,5F,7B,B1,20,F7  
1105 DATA 7B,2F,C6,01,32,02,B8,C9,2A,2A,B8,01,16,00,09,22  
1106 DATA 75,90,2B,22,7F,90,21,72,90,11,00,B8,01,16,00,ED  
1107 DATA B0,C9,01,01,00,00,00,16,00,00,00,00,02,16,00,00  
1108 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
```

リスト 6

```
;***SIO***  
;  
ORG 9000H  
0000 AREA2: EQU 0B800H  
00CE MODE: EQU OC EH  
0020 PORT0: EQU 20H  
0021 PORT1: EQU PORT0+1  
0003 COUNT: EQU AREA2+3  
002A HADDR: EQU AREA2+2AH  
;  
9000 CD5B90 CALL TDATA  
9003 CD3E90 CALL SUMC  
9006 CD0D90 CALL SINIT  
9009 CD2C90 CALL SIO  
900C C9 RET  
;  
;  
900D 3E00 SINIT: LD A,0  
900F D321 OUT (PORT1),A
```

9011 D321 OUT (PORT1),A
9013 D321 OUT (PORT1),A
9015 3E40 LD A,40H
9017 D321 OUT (PORT1),A
9019 3ECE LD A,MODE
901B D321 OUT (PORT1),A
901D 3E15 LD A,1SH
901F D321 OUT (PORT1),A
9021 C9 RET

;
9022 F5 OUTPUT:PUSH AF
9023 DB21 WAIT: IN A,(PORT1)
9025 1F RRA
9026 30FB JR NC,WAIT
9028 F1 POP AF
9029 D320 OUT (PORT0),A
902B C9 RET

;
902C ED4B03BB SIO: LD BC,(COUNT)
9030 2100BB LD HL,AREA2
9033 7E SIO2: LD A,(HL)
9034 CD2290 CALL OUTPUT
9037 23 INC HL
9038 0B DEC BC
9039 78 LD A,B
903A B1 OR C
903B 20F6 JR NZ,SIO2
903D C9 RET

;
903E 1E00 SUMC: LD E,0
9040 E04B03BB LD BC,(COUNT)
9044 2100BB LD HL,AREA2
9047 7B SLOOP: LD A,E
9048 B6 ADD A,(HL)
9049 23 INC HL
904A 0B DEC BC
904B 5F LD E,A
904C 7B LD A,B
904D B1 OR C
904E 20F7 JR NZ,SLOOP
9050 7B LD A,E
9051 2F CPL
9052 C601 ADD A,1
9054 3202BB LD (AREA2+2),A

```

9057 C9          RET
;
9058 2A2ABB  TDATA: LD   HL,(HADDR)
905B 011600      LD   BC,22
905E 09          ADD  HL,BC
905F 227590      LD   (DATA1),HL
9062 2B          DEC  HL
9063 227F90      LD   (DATA2),HL
;
9066 217290      LD   HL,DATA
9069 1100B8      LD   DE,AREA2
906C 011600      LD   BC,22
906F EDB0          LDIR
9071 C9          RET
;
;
9072 01          DATA: DB   1
9073 01          DB   1
9074 00          DB   0
9075 0000  DATA1: DW   0
9077 16          DB   22
9078 00000000      DB   0,0,0,0
;
907C 02          DB   2
907D 1600      DW   22
907F 0000  DATA2: DW   0
9081 00          DB   0
9082 0000      DW   0
9084 00000000      DB   0,0,0,0      ;
;
9088          END

```

リスト7 RC-20用マシン語プログラム(例1)

```

;*****KOHAN*****  

BB16    PORG: EQU 08816H  

409A    ZBOEND: EQU 409AH  

40B6    SAVER: EQU 40B6H  

40C6    LOADR: EQU 40C6H  

42AE    MUSIC: EQU 42AEH  

;  

        ORG  PORG  

;  

BB16 00    MAIN: DB   0  

BB17 40555349      DB   'MUSIC'     PLAY'  

BB1B 43202020  

BB1F 2020504C

```

```

B823 4159
B825 1805      JR  ENTRY
B827 00          DB  0
B828 1688      DW  MAIN
B82A B200      DW  PRGEND-MAIN
;
B82C           ENTRY:
;
B82C 214388      LD  HL,DATA
B82F 7E      LOOP: LD  A,(HL)
B830 FEFF      CP  OFFH
B832 CA9A40      JP  Z,ZBOEND
B835 23          INC HL
B836 46          LD  B,(HL)
B837 23          INC HL
B838 CDB640      CALL SAVER
B83B CDAE42      CALL MUSIC
B83E CDC640      CALL LOADR
B841 18EC      JR  LOOP
;
B843           DATA: DS  0
B843 0C060C02      DB  12,6,12,2,12,6,14,2,16,6,16,2
B847 0C060E02
B848 10061002
B84F 10061002      DB  16,6,16,2,14,6,12,2,14,6,16,2
B853 0E060C02
B857 0E061002
B858 0C060C02      DB  12,6,12,2,7,8,16,6,16,2,16,6,17,2,19,6,19,2
B85F 07081006
B863 10021006
B867 11021306
B86B 1302
B86D 13061302      DB  19,6,19,2,17,6,16,2,17,6,19,2
B871 11061002
B875 11061302
B879 10081902      DB  16,11,25,2,19,2,16,11,25,2,19,2
B87D 1302100B
B881 19021302
B885 10081902      DB  16,11,25,2,19,2,16,6,19,2,16,6
B889 13021006
B88D 13021006
B891 1302100B      DB  19,2,16,11,25,4,OFFH
B895 1904FF

```

リスト 8 RC-20マシン語プログラム(例 2)

```
;*GRAPH*
BB16    PORG: EQU 0B816H
BFBA    VRAMP: EQU 0BFBAH
002A    PAGEL: EQU 2AH
411C    DISP: EQU 411CH
41CB    STBY: EQU 41CBH
409A    ZBOEND: EQU 409AH
;
;          ORG PORG
;
BB16 00   MAIN: DB 0
BB17 47524150 DB 'GRAPH PROGRAM '
BB18 48205052
BB1F 4F475241
BB23 4D20
BB25 1B05      JR ENTRY
BB27 00      DB 0
BB28 1B88      DW MAIN
BB2A 0701      DW PRGEND-MAIN
;
BB2C 2174BB  ENTRY: LD HL,MDATA
BB2F 11BABF  LD DE,VRAMP
BB32 012A00  LD BC,PAGEL
BB35 E5      PUSH HL
BB36 D5      PUSH DE
BB37 C5      PUSH BC
BB38 EDB0    LDIR
BB3A 3E01    LD A,1
BB3C 1E40    LD E,40H
BB3E CD1C41  CALL DISP
BB41 C1      POP BC
BB42 D1      POP DE
BB43 E1      POP HL
BB44 09      ADD HL,BC
BB45 E5      PUSH HL
BB46 D5      PUSH DE
BB47 C5      PUSH BC
BB48 EDB0    LDIR
BB4A 3E01    LD A,1
BB4C 1E00    LD E,0
BB4E CD1C41  CALL DISP
BB51 C1      POP BC
BB52 D1      POP DE
BB53 E1      POP HL
BB54 09      ADD HL,BC
```

B855 E5	PUSH HL
B856 D5	PUSH DE
B857 C5	PUSH BC
B858 EDB0	LDIR
B85A 3E02	LD A,2
B85C 1E00	LD E,0
B85E CD1C41	CALL DISP
B861 C1	POP BC
B862 D1	POP DE
B863 E1	POP HL
B864 09	ADD HL,BC
B865 EDB0	LDIR
B867 3E02	LD A,2
B869 1E40	LD E,40H
B86B CD1C41	CALL DISP
B86E CDCB41	CALL STBY
B871 C39AA0	JP ZBOEND

B874	MDATA: DS 0
B874 00000000	DB 0,0,0,0,0,0,0,0,0,0
B878 00000000	
B87C 0000	
B87E 0102070A	DB 1,2,7,0AH,0EH,18H,23H
B882 0E1823	
B885 34342320	DB 34H,34H,23H,20H,20H,23H,34H,34H
B889 20233434	
B88D 23180E0A	DB 23H,18H,0EH,0AH,7,2,1,0,0,0
B891 07020100	
B895 0000	
B897 00000000	DB 0,0,0,0,0,0,0,0,0,0
B89B 00000000	
B89F 00000000	
B8A3 0F3A6AAA	DB 0FH,3AH,6AH,0AAH,0BFH,0AOH,0EOH
B8A7 BF4AE0	
B8A8 00000415	DB 0,0,4,15H,0CEH,20H,0A4H,0CEH
B8AE CE20A4CE	
B8B2 0E0ECEA4	DB 0EH,0EH,0CEH,0A4H,20H,0CEH,15H
B8B6 20CE15	
B8B9 040000E0	DB 4,0,0,0EH
B8BD A0BFAA6A	DB 0AOH,0BFH,0AAH,6AH,03AH,0FH
B8C1 3A0F	
B8C3 00000000	DB 0,0,0,0,0,0,0,0,0,0
B8C7 00000000	
B8CB 000080C0	
B8CF A0A0B070	DB 0AOH,0AOH,0BOH,70H,B,B,4,4,4

BBD3	08080404	
BBD7	04	
BBD8	82FB848C	DB 82H,0FBH,84H,8CH,8CH,8CH,8CH,84H,0FBH,82H
BBD9	BC8C8C84	
BBE0	FBB2	
BBE2	04040408	DB 4,4,4,8,8,70H,0B0H,0A0H,0A0H,0C0H
BBE6	08708080	
BBEA	A0C0	
BBEC	80000000	DB 80H,0,0,0,0,0,0,0,0,0,0,0,1CH,22H
BBF0	00000000	
BBF4	0000001C	
BBF8	22	
BBF9	26404640	DB 26H,40H,46H,40H,46H,42H,42H,0BAH
BBF0	464242BA	
B901	82820204	DB 82H,82H,2,4,4,4,4,4,4,2,92H,0BAH
B905	04040404	
B909	040292BA	
B90D	9E7A5246	DB 9EH,7AH,52H,46H,40H,46H,40H,26H,22H
B911	40464026	
B915	22	
B916	1C000000	DB 1CH,0,0,0,0
B91A	00	
B91B	0000	DB 0,0
	;	
B91D		PRGEND:
B91D		END

3. PC-8801/8801mkIIの場合

PC-8801/8801mkIIには、N-BASICモードとN88-BASICモードの2つがありますが、N-BASICモードについては、先に述べたPC-8001/8001mkIIの方法がそのまま使用できます。そこで、ここではN-88BASICモードでの使用方法を説明します。

CP/Mによる場合

N88-BASICモードで立上げて実行するCP/Mでも、先の「2. PC-8001/8001mkII」のところで述べた方法が、すべて使用できます。

DUAD88による場合

転送プログラムは、N88-BASICで作ります(リスト9)。

RC-20に転送するプログラムとしてはPC-8001のDUADPCで使用した“KOHAN”(リスト7)と“GRAPH”(リスト8)を、そのまま使用します。

これらをDUAD88のエディタで作成し、アセンブルして、HEX型ファイル KOHAN.HEX と GRAPH.HEXを作成します。

次に、これらのHEX型ファイルをマシン語のバイナリ型ファイルに変換するのですが、DUAD88のローダではB816Hからのエリアに、これらのプログラムをロードすることができないので、OFFSETとして1000Hを指定し、C816H～にロードします。

たとえば、KOHAN.HEXの場合は、C816H～C897Hにロードします。同時に、DUAD88からN88-BASICにもどるので、

BSAVE "KOHAN.BIN", &HC816, &H82

として、マシン語のバイナリ型式のファイルを作成します。

“GRAPH”についても同様に、“GRAPH.BIN”を作成します。

RC-20への転送は、N88-BASICモードで、

RUN "TRANS"

を実行します。“file name?”と問合せてくるので、たとえば、KOHAN.BINを指定します。次いで、“RC-20 READY OK?”と問合せてくるので、RC-20を「LOAD」モードにし、「LOADING NOW」にセットしたら、PC側のRETURN KEYを押します。転送が始まって、終了すると、“* JOB COMPLETE*”のメッセージが出力され、RC-20側は自動的に、「PROGRAM RUN」のモードになります。

リスト9 転送プログラム (TRANS)

```
100 REM $transn$  
110 CLEAR ,&HC000  
120 INPUT "file name";F$  
130 BLOAD F$  
140 FOR I=0 TO 21  
150 POKE &HC800+I,0  
160 NEXT I  
170 POKE &HC800,1  
180 POKE &HC801,1  
190 POKE &HC805,22  
200 POKE &HC80A,2  
210 POKE &HC80B,22  
220 I=PEEK(&HC82B)  
230 I=I*256  
240 J=PEEK(&HC82A)  
250 I=I+J*22  
260 J=I  
280 ZL=J MOD 256  
290 ZH=INT(J/256)  
300 POKE &HC803,ZL  
310 POKE &HC804,ZH  
320 J=J-1  
330 ZL=J MOD 256  
340 ZH=INT(J/256)  
350 POKE &HC809,ZL  
360 POKE &HC80E,ZH  
370 CHK=0  
380 FOR J=0 TO I  
390     CHK=CHK+PEEK(&HC800+J)  
400 NEXT J  
410 AX=CHK-INT(CHK/256)*256  
420 AL=256-AX  
430 POKE &HC802,AX  
440 REM  
450 OPEN "COM1:N82H" FOR OUTPUT AS #1  
460 INPUT "rc-20 ready ok";X$  
470 FOR K=0 TO I  
480     A=PEEK(&HC800+K)  
490     PRINT #1,CHR$(A);  
500 NEXT K  
510 PRINT "job complete"  
520 CLOSE #1  
530 END
```

4. FM-7の場合

FM-7では、RS-232Cの機能は標準装備されていないので、オプションのRS-232Cカードが必要となります。

転送速度(ポーレート)は、カード上のディップ・スイッチで指定します。slowクロック(64分周)で、2400ポーに設定することとし、表2-2に示したようにセットします。

表2-2
RS-232Cカードの
ディップスイッチの設定

ディップSWの番号	ON/OFF
1	OFF
2	OFF
3	OFF
4	ON
5	OFF
6	ON
7	OFF
8	OFF

FM-7は、CPUに6809を使用していますが、オプションとしてZ-80カードがあるので、これを装備すると、Z-80マシンとしてRC-20用のマシン語作成に便利ですが、ここでは簡単にするために、Z-80カードなしで行なってみることにします。そしてRC-20用のマシン語プログラムは、FM-7のモニタを使ってマシン語コードを直接メモリに書き込み、ファイルとしてディスクかテープにSAVEすることにします。

RC-20に転送するマシン語コードを格納するエリアを&H5000から確保することとし、

CLEAR, &H4FFF

で、そのエリアを最初に確保します。

RC-20用のマシン語プログラムの例として、リスト7とリスト8の"KOHAN"と"GRAPH"を用います。これらのダンプリストを、次ページリスト10に示します。

MONコマンドでモニタ・モードに入り、Mコマンドで&H5016より、これらのRC-20用マシン語プログラムをメモリに書き込みます。次いで、BASICモードにもどり、これらをSAVEMコマンドで以下のように、テープまたはディスクに書き込みます。

{(テ - ブ) SAVEM "CAS0 : KOHAN", &H5016, &H5097, &H0000
(ディスク) SAVEM "0 : KOHAN", &H5016, &H5097, &H0000

{(テ - ブ) SAVEM "CAS0 : GRAPH", &H5016, &H511C, &H0000
(ディスク) SAVEM "0 : GRAPH", &H5016, &H511C, &H0000

リスト10 RC-20用マシン語プログラム・ダンプリスト

0 1 2 3 4 5 6 7 8 9 A B C D E F

```

5010          00 4D 55 53 49 43 20 20 20 20
5020 20 50 4C 41 59 18 05 00 16 BB B2 00 21 43 BB 7E
5030 FE FF CA 9A 40 23 46 23 CD B6 40 CD AE 42 CD C6
5040 40 18 EC 0C 06 0C 02 0C 06 0E 02 10 06 10 02 10
5050 06 10 02 0E 06 0C 02 0E 06 10 02 0C 06 0C 02 07
5060 08 10 06 10 02 10 06 11 02 13 06 13 02 13 06 13
5070 02 11 06 10 02 11 06 13 02 10 08 19 02 13 02 10
5080 08 19 02 13 02 10 08 19 02 13 02 10 06 13 02 10
5090 06 13 02 10 0B 19 04 FF

```

● KOHAN ●

0 1 2 3 4 5 6 7 8 9 A B C D E F

```

5010          00 47 52 41 50 48 20 50 52 4F
5020 47 52 41 4D 20 18 05 00 16 BB 07 01 21 74 BB 11
5030 BA 5F 01 2A 00 E5 D5 C5 ED B0 3E 01 1E 40 CD 1C
5040 41 C1 D1 E1 09 E5 D5 C5 ED B0 3E 01 1E 00 CD 1C
5050 41 C1 D1 E1 09 E5 D5 C5 ED B0 3E 02 1E 00 CD 1C
5060 41 C1 D1 E1 09 ED B0 3E 02 1E 40 CD 1C 41 CD C8
5070 41 C3 9A 40 00 00 00 00 00 00 00 00 00 00 01 02
5080 07 0A 0E 18 23 34 34 23 20 20 23 34 34 23 18 0E
5090 0A 07 02 01 00 00 00 00 00 00 00 00 00 00 00 00
50A0 00 00 00 0F 3A 6A AA BF A0 E0 00 00 04 15 CE 20
50B0 A4 CE 0E 0E CE A4 20 CE 15 04 00 00 E0 A0 BF AA
50C0 6A 3A 0F 00 00 00 00 00 00 00 00 00 00 B0 C0 A0
50D0 A0 B0 70 08 08 04 04 04 B2 FB 84 BC BC BC BC B4
50E0 FB B2 04 04 08 08 70 B0 A0 A0 C0 B0 00 00 00
50F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5100 BA B2 B2 02 04 04 04 04 04 04 04 04 04 04 04 04 04

```

0 1 2 3 4 5 6 7 8 9 A B C D E F

```
5110 46 40 46 40 26 22 1C 00 00 00 00 00 00 00
```

● GRAPH ●

3番目のパラメータの入口番地は、実際には使用しないので（上記の例では&H0000になっている）、適当なアドレスを入れておきます。

RC-20への転送を行うプログラムは、これらのファイルをデータとして読み込んで、コモンヘッダとファイルヘッダを作成し、RC-20に転送します。リスト11は、転送プログラム（TRANS）を示したものです。行番号70のOPEN文のオプション（S8N2）は、slowクロック（64分周）、データ長

8ビット、パリティなし、2ストップビットを指定します。FM-7では、このオプションを省略したとき、自動的に(S8N2)が仮定されることになっているので、行番号70は、

OPEN "O", #1, "COM0 : "

でもかまいません。

転送プログラム(TRANS)を実行させると、"FILE NAME?"と問合せてくるので、テーブまたはディスク上のRC-20へ転送するマシン語ファイル名を指定します。この場合、ファイル名の前にデバイス名をつけて下さい。(ディスクのドライブ0では、デバイス名の省略ができます)。

次いで、"RC-20 READY OK?"と問合せてくるので、RC-20を「LOAD」モードにして、"LOADING NOW"の表示が出るようにセットします。そして、FM-7のリターンキーを押すとプログラムの転送が始まり、終了すると"*JOB END*"とメッセージが出来されます。

RC-20方は、自動的に転送されたデータのチェックを行い、エラーがなければ、"PROGRAM RUN"のモードに入ります。

リストII 転送プログラム(TRANS)

```

10 REM !trans!
20 CLEAR ,&H4FFF
30 INPUT "file name";F$
40 LOADM F$
50 GOSUB 180
60 GOSUB 270
70 OPEN "O",#1,"COM0:(S8N2)"
80 INPUT "rc=20 ready ok ";X$ 
90 FOR K=0 TO I
100 A=PEEK(&H5000+K)
110 PRINT #1,CHR$(A);
120 NEXT K
130 PRINT "#job end"
140 CLOSE #1
150 END
160 FOR I=0 TO 21
170 POKE &H5000,I,0
180 NEXT I
190 POKE &H5000,1
220 POKE &H5001,1
230 POKE &H5005,22

```

```

240 POKE &H500A,2
250 POKE &H500B,22
260 RETURN
270 I=FEEK(&H502B)
280 I=I+256
290 J=FEEK(&H502A)
300 I=I+J-1+22
310 J=I+1
320 ZL=J MOD 256
330 ZH=INT(J/256)
340 POKE &H5003,ZL
350 POKE &H5004,ZH
360 J=J-1
370 ZL=J MOD 256
380 ZH=INT(J/256)
390 POKE &H500B,ZL
400 POKE &H500E,ZH
410 CHK=0
420 FOR J=0 TO I
430 CHK=CHK+PEEK(&H5000+J)
440 NEXT J
450 AX=CHK-INT(CHK/256)*256
460 AX=256-AX
470 POKE &H5002,AX
480 RETURN

```

5. HC-20の場合

ハンドヘルドコンピュータHC-20には、RS-232Cが標準装備されているので、HC-20からRC-20にプログラムを転送することができます。問題は、HC-20のCPUが6800系の6301であるため、Z-80系のCPUをもつRC-20用のマシン語プログラムを、HC-20でソース・プログラムからアセンブルして作成することができないという点です。HC-20上で走るZ-80用のクロス・アセンブラーでもあればよいのですが、今のところはないようです。

そこで、RC-20用のプログラムは、直接ハンド・アセンブルした機械語か、他のZ-80系のパソコンでアセンブルした機械語を、HC-20のモニタを使って機械語エリアにダイレクトに入力することになります。

HC-20では、機械語エリアは&H0A40から後方へMEMSET文で指定するアドレスまで確保することができます。ここでは、&H0C00～&H13FFの2Kバイトのエリアを、RC-20用のプログラムを格納する場所として使用します。

RC-20用のマシン語の例としてはリスト7の“KOHAN”，リスト8の“GRAPH”を使用します。最初にHC-20のBASICを動作させたら、

MEMSET &H1400

で、先に述べた機械語エリアを確保します。

次に、MONでモニタモードに入り、Sコマンドで&0C16からRC-20用のマシン語プログラムをメモリに書き込みます。リスト12に、"KOHAN"と"GRAPH"のダンプリストを示します。

"KOHAN"の場合は&H0C16~0C97に、"GRAPH"の場合は&H0C16~0D1Cに書き込みます。

次いで、これらをAコマンドとWコマンドを使って、マイクロカセットテープにファイルとして書き込みます。"KOHAN"の例で説明すると、AコマンドではT=0C16, L=0C97, O=0000, E=0000を指定し、Wコマンドによって

WM. KOHAN✓

で、テープに書き込みます。

RC-20に転送するプログラムは、すべてこのようにしてテープ上にマシン語ファイルとして格納しておきます。

リスト13は、HC-20用の転送プログラム(TRANS)を示したもので、このプログラムは、テープ上のRC-20用機械語ファイルをHC-20の機械語エリアに読み込んだ上、コモンヘッダとファイルヘッダを作成し、チェックサムを計算した上、RC-20に転送するものです。

まず、TRANSを実行させると、"FILE NAME?"を問合せるので、RC-20に転送するテープ上のファイル名を指定します。"KOHAN"の場合なら、KOHANと入力します。

次いで、"*RC-20 READY OK?"と問合せてくるので、RC-20を「LOAD」モードにし"LOADING NOW"の表示が出るようにします。その後でHC-20のリターンキーを入力すると、プログラムの転送が始まり、転送が終了するとメッセージが出て、BASICのプロンプトにもどります。

ここで注意することは、"*RC-20 READY OK"が表示されてから、RC-20を「LOAD」モードの受信可能状態にすることです。HC-20では、RS-232Cインターフェースへの電流はOPEN "COM0:"命令の実行で、はじめて流れますが、この電流の流れはじめに、信号線に不安定な電圧を発生します。このため、OPEN "COM0:"の実行時点でRC-20が受信可能状態になっていると、データマニアデータがRC-20に受信されてしまって、LOADエラーが生じてしまいます。したがってTRANSではQOPEN "COM0:"の実行後、"*RC-20 READY OK"のメッセージを出すようにしてあり、この後でRC-20を受信可能状態にすれば、問題は解決します。

リスト12 ダンプリスト

KOHAN

GRAPHÉ

リスト13 HC-20用転送プログラム (TRANS)

```
10 REM *TRANS*
20 MEMSET &H1400
30 WIND
40 INPUT "*FILE NAME";F$
50 LOADM "CAS0:";+F$
60 GOSUB 180
70 GOSUB 270
80 OPEN "0",#1,"COM0:(58
N2F)"
90 WIDTH "COM0:",255
100 INPUT "*RC-20 READY
OK ";X$
110 FOR K=0 TO I
120 A=PEEK(&H0C00+K)
130 PRINT #1,CHR$(A);
140 NEXT K
150 PRINT "*JOB END*"
160 CLOSE #1
170 END
180 FOR I=0 TO 21
190 POKE &H0C00+I,0
200 NEXT I
210 POKE &H0C00,1
220 POKE &H0C01,1
230 POKE &H0C05,22
240 POKE &H0C0A,2
250 POKE &H0C0B,22
260 RETURN
270 I=PEEK(&H0C2B)
280 I=I*256
290 J=PEEK(&H0C2A)
300 I=I+J-1+22
310 J=I+1
320 ZL=J MOD 256
330 ZH=INT(J/256)
340 POKE &H0C03,ZL
350 POKE &H0C04,ZH
360 J=J-1
370 ZL=J MOD 256
380 ZH=INT(J/256)
390 POKE &H0C00,ZL
400 POKE &H0C0E,ZH
410 CHK=0
420 FOR J=0 TO I
430   CHK=CHK+PEEK(&H0C00
+J)
440 NEXT J
450 A%=CHK-INT(CHK/256)*
256
460 A%=256-A%
470 POKE &H0C02,A%
480 RETURN
```

6. HC-40/80/88の場合

HC-40/80/88は、標準のCP/Mを装備しているので、CP/Mの下で転送プログラムとRC-20用マシン語プログラムを作成することにします。そのためには、開発ツールとして、エディタとアセンブラーが必要となります。

ディスクが使用できれば、ディスク版の機械語ツールに含まれているエディタやアセンブラーが使用できますが、ディスクのない場合でも機械語開発ツールのROMがあれば、その中のアセンブラー(ASM80)で作業を進めることができます。ここでは、ASM80を使用した例をあげます。

転送プログラム(TRANS)をリスト14に、RC-20用の機械語プログラム例を、リスト15(KOHAN)とリスト16(GRAPH)にそれぞれ示します。これらは先に示した他の機種用のプログラムと、機能的にはまったく同一です。

ASM80では、エディタとしてBASICのエディタ機能を前提としているので、ソース・プログラムはBASICで入力し、これをRAMディスクまたはマイクロカセットにファイルとして格納します。

ASM80では、アセンブルの結果、オブジェクトプログラムをHEX型式ではなく、COM型式で出力します。そのため、RC-20用のプログラムも、先頭アドレスをB816Hではなく100Hにリロケートした型で作成する必要があります。

転送プログラム(TRANS)では、RS-232Cへデータを出力するのに、CP/MのBIOSコールのPUNCH出力を利用しているので、実行に際しては、あらかじめトランジメント・コマンドのCONFIGで、RS-232Cの規格をRC-20に合うように設定(2400ポー、8ビット、パリティなし、2ストップビット)して置く必要があります。

転送プログラムの起動は、

A>TRANS [D:]XXX.COM

で行ないます。なお、Dはドライブ名、XXX.COMはRC-20用のプログラムです。

転送プログラムのRS-232C出力の部分を、BIOSコールのPUNCH出力ではなく、BIOS内のRS-232Cインターフェース制御ルーチンRSIOXを使用した場合(TRANS2)のプログラムを、リスト17に示します。後者の場合はプログラム内で、HC-40/80/88のRS-232C規格をRC-20に合わせているので、あらかじめCONFIGを使用する必要はありません。RSIOXルーチンの詳細については、HC-40/80/88の「オペレーション・マニュアル」を参照してください。

リスト14 HC-40/80/88用転送プログラム

```
A:b:asm80 trans.asm,trans.com
ASM80(Z80) 0: 0: 0/ 2: 0:14      1

100 0000      ; $TRANS$
120 0000      ;
140 0100      ORG    100H
160 0005      EQU    5
```

180 0000	BOOT	EQU	0
200 005C	FCB	EQU	005CH
220 0080	BUFFR	EQU	0080H
240 4000	AREA	EQU	4000H
260 4003	COUNT	EQU	4000H+3
280 4024	HADDR	EQU	4000H+2AH
300 4016	TOP	EQU	4000H+16H
320 0100	;		
340 0100	;		
360 0100	CD4001	CALL	MOVEF
380 0103	CD4501	CALL	TDATA
400 0106	CD8C01	CALL	SUMC
420 0109	I12C01	LD	DE,MS001
440 010C	CD0A02	CALL	MESS
460 010F	OE01	LD	C,1
480 0111	CD0500	CALL	BDS
500 0114	CD0501	CALL	SIO
520 0117	I11D01	LD	DE,MSEND
540 011A	C3F201	JP	EXIT
560 011D	OD0A202A	MSEND:DB	I3,10,'\$job end\$',I3,10,"\$"
580 012C	OD0A202A	MS001:DB	I3,10,'\$RC-20 READY OK?\$','\$'
600 0140	;		
620 0140	MOVEF:DS	O	
640 0140	I18000	LD	DE,BUFFR
660 0143	OE1A	LD	C,26
680 0145	CD0500	CALL	BDS
700 0148	;	FILE OPEN.	
720 0148	OE0F	LD	C,15
740 014A	I15C00	LD	DE,FCB
760 014D	CD0500	CALL	BDS
780 0150	FEFF	CP	OFFH
800 0152	I1FB01	LD	DE,MSOPNER
820 0155	CAF201	JP	Z,EXIT
840 0158	LOOP0:DS	O	
860 015B	CDB201	CALL	READF
880 015B	201C	JR	NZ,EOF
900 015D	ED5B1002	LD	DE,(BPOINT)
920 0161	;		
940 0161	I2B000	LD	HL,BUFFR
960 0164	OEBO	LD	C,128
980 0166	LOOP1:DS	O	
1000 0166	7E	LD	A,(HL)
1020 0167	I2	LD	(DE),A
1040 0168	23	INC	HL
1060 0169	I3	INC	DE
1080 016A	0D	DEC	C
1100 016B	20F9	JR	NZ,LOOP1
1120 016D	01B000	LD	BC,I28
1140 0170	2A1002	LD	HL,(BPOINT)
1160 0173	09	ADD	HL,BC
1180 0174	221002	LD	(BPOINT),HL

ASMB0(Z80) 0: 0: 0/ 2: 0:47 2

```
1200 0177 1BDF      JR     LOOP0
1220 0179      ; 
1240 0179      EOF:DS  0
1260 0179 0E10      LD     C,16   ;CLOSE.
1280 017B 115C00      LD     DE,FCB
1300 017E CD0500      CALL   B00S
1320 0181 C9      RET
1340 0182      ;
1360 0182      READF:DS  0
1380 0182 0E14      LD     C,20
1400 0184 115C00      LD     DE,FCB
1420 0187 CD0500      CALL   B00S
1440 018A B7      OR    A
1460 018B C9      RET
1480 01BC      %
1500 01BC      SUMC:DS  0
1520 01BC 1E00      LD     E,0
1540 01BE ED4B0340      LD     BC,(COUNT)
1560 0192      ;
1580 0192      ;
1600 0192 210040      LD     HL,AREA
1620 0195      SLOOP:DS  0
1640 0195 7B      LD     A,E
1660 0196 86      ADD   A,(HL)
1680 0197 23      INC   HL
1700 0198 0B      DEC   BC
1720 0199 5F      LD     E,A
1740 019A 7B      LD     A,B
1760 019B B1      OR    C
1780 019C 20F7      JR     NZ,SLOOP
1800 019E 7B      LD     A,E
1820 019F ED44      NEG
1840 01A1      ;
1860 01A1 320240      LD     (AREA+2),A
1880 01A4 C9      RET
1900 01A5      ;
1920 01A5      TDATA:DS  0
1940 01A5 2A2A40      LD     HL,(HADDR)
1960 01AB 011600      LD     BC,22
1980 01AB 09      ADD   HL,BC
2000 01AC 22C201      LD     (DATA1),HL
2020 01AF 2B      DEC   HL
2040 01B0 22CC01      LD     (DATA2),HL
2060 01B3      ;
2080 01B3 21BF01      LD     HL,DATA
2100 01B6 110040      LD     DE,AREA
2120 01B9 011600      LD     BC,22
2140 01BC EDB0      LDIR
```

```

2160 01BE C9      RET
2180 01BF          ;
2200 01BF          DATA:DS 0
2220 01BF 01       DB   1
2240 01C0 01       DB   1
2260 01C1 00       DB   0
2280 01C2          DATA1:DS 0
ASMB80(ZBO) 0: 0: 0/ 2: 1:18    3

2300 01C2 0000      DW   0
2320 01C4 16      DB   22
2340 01C5 00000000  DB   0,0,0,0
2360 01C9          ;
2380 01C9 02      DB   2
2400 01CA 1600      DW   22
2420 01CC          DATA2:DS 0
2440 01CC 0000      DW   0
2460 01CE 00      DB   0
2480 01CF 0000      DW   0
2500 01D1 00000000  DB   0,0,0,0
2520 01D5          ;
2540 01D5          $10:DS 0
2560 01D5 ED4B0340  LD   BC,(COUNT)
2580 01D9 210040  LD   HL,AREA
2600 01DC          $102:DS 0
2620 01DC 7E      LD   A,(HL)
2640 01D0 CDE701  CALL  OUTPUT
2660 01E0 23      INC  HL
2680 01E1 0B      DEC  BC
2700 01E2 78      LD   A,B
2720 01E3 B1      OR   C
2740 01E4 20F6  JR   NZ,$102
2760 01E6 C9      RET
2780 01E7          ;
2800 01E7          OUTPUT:DS 0
2820 01E7 E5      PUSH  HL
2830 01EB C5      PUSH  BC
2840 01E9 0E04  LD   C,A
2860 01EB SF      LD   E,A
2880 01EC CD0500  CALL  BDOS
2890 01EF C1      POP   BC
2900 01F0 E1      POP   HL
2920 01F1 C9      RET
2940 01F2          ;
2960 01F2          EXIT:DS 0
2980 01F2 CD0A02  CALL  MESS
3000 01F5 C30000  JP   BOOT
3020 01F8          MSOPNER:DS 0
3040 01FB 0D0A202A  DB   13,10,'$open error$',13,10,'$'
3060 020A          MESS:DS 0

```

```

3080 020A 0E09      LD    C,9
3100 020C CD0500      CALL  BDOS
3120 020F C9      RET
3140 0210      BPOINT:DS 0
3160 0210 1640      DW    TOP
3200 0212      END
BDOS 0005 BOOT 0000 FCB 005C
BUFFR 0080 AREA 4000 COUNT 4003
HADDR 402A TDP 4016 MOVEF 0140
TDATA 01A5 SUMC 01BC MS001 012C
MESS 020A SIO 01D5 MSEND 011D
EXIT 01F2 MSOPNE 01F8 LOOP0 015B
READF 01B2 EOF 0179 BPOINT 0210
ASMB01(Z80) 0: 0: 0/ 2: 1:50      4

LOOP1 0166 SLOOP 0195 DATA1 01C2
DATA2 01CC DATA 01BF SIO2 01DC
OUTPUT 01E7

```

リスト15

```

D:\b\asm80 kohan.asm
ASMB01(Z80) 0: 1: 2/ 1:19:14      1
100 0000      ; $KOHAN$
120 0000      ;
140 B816      MAIN    EQU    0BB16H
160 409A      ZBOEND  EQU    409AH
180 40B6      SAVER   EQU    40B6H
200 40C6      LOADR   EQU    40C6H
220 42AE      MUSIC   EQU    42AEH
240 0000      ;
260 0100      ORG    100H
280 0100      ;
300 0100      :MAIN
320 0100      START: DS    0
340 0100 00      DB    0
360 0101 6D757369      DB    'music     play'
380 010F 1805      JR    ENTRY
400 0111 00      DB    0
420 0112 16BB      DW    MAIN
440 0114 B200      DW    PRGEND-START
460 0116      ;
480 0116      ENTRY:DS  0
500 0116 2143BB      LD    HL,DATA-START+MAIN
520 0119 7E      LOOP:LD  A,(HL)
540 011A FEFF      CP    OFFH
560 011C CA9A40      JP    Z,ZBOEND
580 011F 23      INC   HL
600 0120 46      LD    B,(HL)
620 0121 23      INC   HL

```

49

918 (1)

320 011F E5	PUSH	HL
330 0120 D5	PUSH	DE
340 0121 C5	PUSH	BC
350 0122 EDB0	LDIR	
360 0124 3E01	LD	A,1
370 0126 1E40	LD	E,40H
380 0128 CD1C41	CALL	DISP
390 012B C1	POP	BC
400 012C D1	POP	DE
410 012D E1	POP	HL
420 012E 09	ADD	HL,BC
430 012F E5	PUSH	HL
440 0130 D5	PUSH	DE
450 0131 C5	PUSH	BC
460 0132 EDB0	LDIR	
470 0134 3E01	LD	A,1
480 0136 1E00	LD	E,0
490 0138 CD1C41	CALL	DISP
500 013B C1	POP	BC
510 013C D1	POP	DE
520 013D E1	POP	HL
530 013E 09	ADD	HL,BC
540 013F E5	PUSH	HL
550 0140 D5	PUSH	DE
560 0141 C5	PUSH	BC
570 0142 EDB0	LDIR	
580 0144 3E02	LD	A,2
590 0146 1E00	LD	E,0
600 0148 CD1C41	CALL	DISP
610 014B C1	POP	BC
620 014C D1	POP	DE
630 014D E1	POP	HL
640 014E 09	ADD	HL,BC
650 014F EDB0	LDIR	
660 0151 3E02	LD	A,2
670 0153 1E40	LD	E,40H
680 0155 CD1C41	CALL	DISP
690 0158 CDCB41	CALL	STBY
700 015B C39A40	JP	Z80END
710 015E	;	
720 015E	M DATA:	DS 0
730 015E 00000000	DB	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
740 0168 0102070A	DB	1,2,7,0AH,0EH,18H,23H
750 016F 34342320	DB	34H,34H,23H,20H,20H,23H,34H,34H
760 0177 23180E0A	DB	23H,1BH,0EH,0AH,7,2,1,0,0,0
770 01B1 00000000	DB	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
780 01BD 0F3A6AAA	DB	0FH,3AH,6AH,0AAH,0BFH,0AOH,0EOH
790 0194 00000415	DB	0,0,4,15H,0CEH,20H,0A4H,0CEH

4/24/17 HC-40/80用耗送窗口为之(TRANS2)

800 019C 0E0CECA4	DB	0EH,0EH,0CH,0AH,20H,0CH,15H
810 01A3 040000E0	DB	4,0,0,0,E0H
820 01A7 A0BFA6A	DB	0AH,0FH,0AH,6AH,03AH,EFH
830 01AD 00000000	DB	0,0,0,0,0,0,0,0,B0H,0C0H
840 01B9 0A0B070	DB	0AH,0AH,0B0H,0B0H,70H,8,4,4,4
850 01C2 B2FB8ABC	DB	B2H,0FH,8AH,0CH,8CH,0CH,8AH,0FH,82H
860 01CC 0400408	DB	4,A,4,B,8,70H,0B0H,0AH,0AH,0COH
870 01D6 80000000	DB	80H,0,0,0,0,0,0,0,0,1CH,22H
880 01E3 2644630	DB	26H,40H,46H,40H,46H,42H,08AH
890 01EB 28B20204	DB	82H,82H,2,4,4,4,4,4,2,92H,08AH
900 01F7 9E7A5246	DB	9EH,7AH,52H,46H,0AH,46H,10H,26H,22H
910 0200 1C000000	DB	1CH,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
920 0205 0000	DB	920 0207
940 0207 PRGEND: DS 0	;	PRGEND: DS 0
950 0207 END	;	END
MAIN 8816 VRAMP BFB4 PABEL 002A	DISP 411C STBY 4118 Z80END 409A	
DATA 015E ENTRY 0116 PRGEND 0207	DISP 411C STBY 4118 Z80END 409A	
100 0000 ; TRANS	100 0000 ; TRANS	
110 0000 ;	110 0000 ;	
120 0100 DB6 100H	120 0100 DB6 100H	
130 0005 B005 E8U 5	130 0005 B005 E8U 5	
140 0000 B007 E8U 0	140 0000 B007 E8U 0	
150 005C FCB E8U 005CH	150 005C FCB E8U 005CH	
160 0080 BUFR E8U 0080H	160 0080 BUFR E8U 0080H	
170 4000 AREA E8U 4000H	170 4000 AREA E8U 4000H	
180 4003 COUNT E8U 4000H+3	180 4003 COUNT E8U 4000H+3	
190 402A HADR E8U 4000H+2AH	190 402A HADR E8U 4000H+2AH	
200 4016 TOP E8U 4000H+16H	200 4016 TOP E8U 4000H+16H	
210 0100 ;	210 0100 ;	
220 0100 ;	220 0100 ;	
230 0100 ;	230 0100 ;	
240 0100 2A0100 LD HL,(B00T+1)	240 0100 2A0100 LD HL,(B00T+1)	
250 0103 015100 LD BC,51H	250 0103 015100 LD BC,51H	
260 0106 09 ADD HL,BC	260 0106 09 ADD HL,BC	
270 0107 226302 LD (RSIXD+1),HL	270 0107 226302 LD (RSIXD+1),HL	
280 010A ;	280 010A ;	
300 010A C0A01	300 010A C0A01	
310 0110 CD9601 CALL TDATA	310 0110 CD9601 CALL TDATA	
320 0113 113601 LD DE,MS001	320 0113 113601 LD DE,MS001	
330 0116 C05102 CALL MESS	330 0116 C05102 CALL MESS	

DB:35M80 TRANS2,AS8,TRANS2,COM
AM801ZB0) 0: 1: 2: 2:35:41 1

340 0119 0E01 LD C,1
350 011B CD0500 CALL BDOS
360 011E CDDF01 CALL SIO
370 0121 112701 LD DE,MSEND
380 0124 C33902 JP EXIT
.390 0127 0D0A202A MSEND:DB 13,10,'\$job end\$',13,10,'\$'
400 0136 0D0A202A MS001:DB 13,10,'\$RC-20 READY OK?','\$'
410 014A ;
420 014A MOVEF:DS 0
430 014A 11B000 LD DE,BUFFR
440 014D 0E1A LD C,26
450 014F CD0500 CALL BDOS
460 0152 ; FILE OPEN.
470 0152 0EOF LD C,15
480 0154 115C00 LD DE,FCB
490 0157 CD0500 CALL BDOS
500 015A FEFF CP OFFH
510 015C 113F02 LD DE,MSOPNER
520 015F CA3902 JP Z,EXIT
530 0162 LOOP0:DS 0
540 0162 CDB801 CALL READF
550 0165 201C JR NZ,EOF
560 0167 ED5B6002 LD DE,(BPOINT)
570 016B ;
580 016B 218000 LD HL,BUFFR
590 016E 0E80 LD C,128
600 0170 LOOP1:DS 0
610 0170 7E LD A,(HL)
620 0171 12 LD (DE),A
630 0172 23 INC HL
640 0173 13 INC DE
650 0174 0D DEC C
660 0175 20F9 JR NZ,LOOP1
670 0177 018000 LD BC,128
680 017A 2A6002 LD HL,(BPOINT)
690 017D 09 ADD HL,BC
700 017E 226002 LD (BPOINT),HL
710 0181 1BDF JR LOOP0
720 0183 ;
730 0183 EOF:DS 0
740 0183 0E10 LD C,16 ;CLOSE.
750 0185 115C00 LD DE,FCB
760 0188 CD0500 CALL BDOS
770 018B C9 RET
780 018C ;
790 018C READF:DS 0
800 018C 0E14 LD C,20

810 018E 115C00	LD	DE,FCB
820 0191 C00500	CALL	B005
830 0194 B7	OR	A
840 0195 C9	RET	
850 0196 ;		
860 0196 SUMC:DS 0		
870 0196 1E00 LD E,0		
880 0198 ED4B0340 LD BC,(COUNT)		
890 019C ;		
900 019C ;		
910 019C 210040 LD HL,AREA		
920 019F SLOOP:DS 0		
930 019F 7B LD A,E		
940 01A0 86 ADD A,(HL)		
950 01A1 23 INC HL		
960 01A2 0B DEC BC		
970 01A3 5F LD E,A		
980 01A4 7B LD A,B		
990 01A5 B1 OR C		
1000 01A6 20F7 JR NZ,SLOOP		
1010 01A8 7B LD A,E		
1020 01A9 ED44 NEG		
1030 01AB ;		
1040 01AB 320240 LD (AREA+2),A		
1050 01AE C9 RET		
1060 01AF ;		
1070 01AF TDATA:DS 0		
1080 01AF 2A2A40 LD HL,(HADDR)		
1090 01B2 011600 LD BC,22		
1100 01B5 09 ADD HL,BC		
1110 01B6 22CC01 LD (DATA1),HL		
1120 01B9 2B DEC HL		
1130 01BA,22D601 LD (DATA2),HL		
1140 01BD ;		
1150 01BD 21C901 LD HL,DATA		
1160 01C0 110040 LD DE,AREA		
1170 01C3 011600 LD BC,22		
1180 01C6 EDB0 LDIR		
1190 01C8 C9 RET		
1200 01C9 ;		
1210 01C9 DATA:DS 0		
1220 01C9 01 DB 1		
1230 01CA 01 DB 1		
1240 01CB 00 DB 0		
1250 01CC DATA1:DS 0		
1260 01CC 0000 DW 0		
1270 01CE 16 DB 22		

1280 01CF 00000000	DB	0,0,0,0
1290 01D3	;	
1300 01D3 02	DB	2
1310 01D4 1600	DW	22
1320 01D6	DATA2:DS	0
1330 01D6 0000	DW	0
1340 01DB 00	DB	0
1350 01D9 0000	DW	0
1360 01DB 00000000	DB	0,0,0,0
1370 01DF	;	
1380 01DF	SIO:DS	0
1390 01DF 0610	LD	B,10H ;RS232C OPEN
1400 01E1 215702	LD	HL,PARLIST
1410 01E4 CD6202	CALL	RSIOX
1420 01E7 2047	JR	NZ,RERR
1430 01E9	;	
1440 01E9 ED4B0340	LD	RC,(COUNT)
1450 01ED 210040	LD	HL,AREA
1460 01F0	SIO2:DS	0
1470 01F0 7E	LD	A,(HL)
1480 01F1 CD0002	CALL	OUTPUT
1490 01F4 23	INC	HL
1500 01F5 0B	DEC	BC
1510 01F6 7B	LD	A,B
1520 01F7 B1	OR	C
1530 01FB 20F6	JR	NZ,SIO2
1540 01FA 0620	LD	B,20H ;RS232C CLOSE
1550 01FC CD6202	CALL	RSIOX
1560 01FF	;	
1570 01FF C9	RET	
1580 0200	;	
1590 0200	OUTPUT:DS	0
1600 0200 E5	PUSH	HL
1610 0201 C5	PUSH	BC
1620 0202 F5	PUSH	AF
1630 0203 215702	LD	HL,PARLIST
1640 0206 0640	SIO3:LD	B,40H ;CHECK SEND STATUS
1650 0208 CD6202	CALL	RSIOX
1660 020B 2023	JR	NZ,RERR
1670 020D B7	OR	A
1680 020E 28F6	JR	Z,SIO3
1690 0210 F1	POP	AF
1700 0211 4F	LD	C,A
1710 0212 0660	LD	B,60H ;SEND DATA
1720 0214 CD6202	CALL	RSIOX
1730 0217 2017	JR	NZ,RERR
1740 0219 C1	POP	BC

```

1750 021A E1      POP    HL
1760 021B C9      RET
1770 021C          ;
1780 021C 0D0A202A MSERR:DB 13,10,' $rs232c error',13,10,''
1790 0230          ;
1800 0230 111C02 RERR:LD  DE,MSERR
1810 0233 CD5102   CALL   MESS
1820 0236 C30000   JP     BOOT
1830 0239          ;
1840 0239 EXIT:DS  0
1850 0239 CD5102   CALL   MESS
1860 023C C30000   JP     BOOT
1870 023F          MSOPNER:DS 0
1880 023F 0D0A202A DB    13,10,' $open error',13,10,''
1890 0251          MESS:DS 0
1900 0251 0E09      LD    C,9
1910 0253 CD0500   CALL   BDOS
1920 0256 C9      RET
1930 0257 PARLIST:DS 0
1940 0257 6602      DW    BUFR2
1950 0259 0008      DW    2048
1960 025B 0C      DB    0CH
1970 025C 03      DB    3
1980 025D 00      DB    0
1990 025E 03      DB    3
2000 025F FF      DB    OFFH
2010 0260          ;
2020 0260 BPOINT:DS 0
2030 0260 1640      DW    TOP
2040 0262 CD0000   RSIOX:CALL 0000H
2050 0265 C9      RET
2060 0266          ;
2070 0266 BUFR2:DS 2048
2080 0466 END
BDOS 0005 BOOT 0000 FCB 005C
BUFR 0080 AREA 4000 COUNT 4003
HADDR 402A TOP 4016 RSIOX 0262
MOVEF 014A TDATA 01AF SUMC 0196
MS001 0136 MESS 0251 SIO 01DF
MSEND 0127 EXIT 0239 MSOPNE 023F
LOOP0 0162 READF 018C EOF 0183
BPOINT 0260 LOOP1 0170 SLOOP 019F
DATA1 01CC DATA2 01D6 DATA 01C9
PARLIS 0257 RERR 0230 SIO2 01F0
DPUTPUT 0200 SIO3 0206 MSERR 021C
BUFR2 0266

```

NUMBER

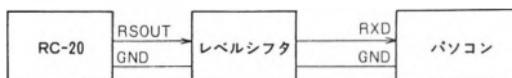
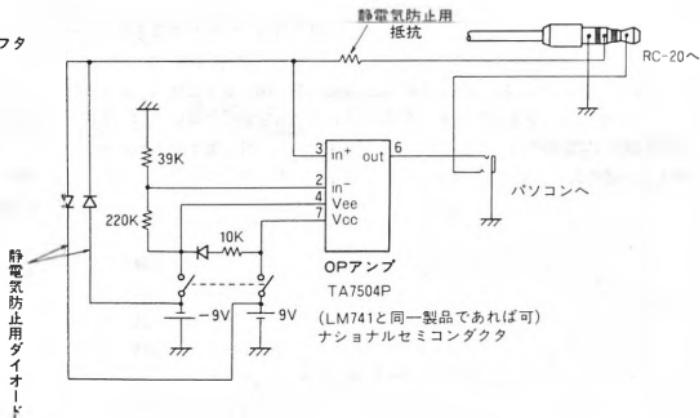
3

RC-20からパソコン へのデータの転送

RC-20 からパソコンへ RS-232C を介してデータを送ることができます。この場合、RC-20 からの出力電圧のレベル($-3V$ ~ $0V$)を RS-232C の規格の電圧レベル($-10V$ ~ $+10V$)に変換する必要があります。このために必要なレベルシフタは EPSON から販売されており、これを RC-20 とパソコンとの間に接続します。レベルシフタを自作される方のために参考例を図3-1に示します。

FIG 3-1

レベルシフタ



さて、実際にRC-20からパソコンへデータを転送するには、この作業を行なうプログラムを作る必要があります。

幸いにも、RC-20のROM内には後述のように、RC-20内のRAM上のデータをRS-232Cを介して出力するサブルーチン(RSOUT)がありますので、これを使用すると便利でしょう。

RSOUTの使用条件は次の通りです。

RSOUT	4320H
HLレジスタ	: 転送すべきメモリ・エリアの先頭アドレス
DEレジスタ	: 転送すべきメモリ・エリアの終了アドレス
Aレジスタ	: ポーレート
2400ポーのときは08H	

ポーレートは、2400, 1200, 600, 300ポーの4種類が使用できますが、パソコンからRC-20への転送には2400ポーを使用していますので、RC-20からパソコンへの転送もこれに合わせておいた方がよいでしょう。RC-20からパソコンへの転送例として、RC-20のROM(8K:4000H-5FFFH)の内容をパソコンへ転送する例を紹介します。

注1: RC-20からデータを出力する場合、RC-20のROM内サブルーチンRSOUTを使用しますが、このサブルーチンは、バラメータで指定したデータの後に続けて、最大22バイトの余分な情報(ゴミ)を出力するので、受信側のパソコンでは、このことを配慮する必要があります。

注2: RC-20からRSOUTでデータを出力するとき、約2.6%の確率で、バラメータによってセットしたアドレスの次のアドレスからデータを出力することがあります。これはRC-20の計時用ICのBUGですが、EPSON社では、次の転送規約で対処しています。

- ① 出力データの前に「2AH」('*')を2個加え、最初の2AHのある番地を、RSOUTでの転送開始アドレスとする。



- ② 受信側では、送られてきたデータの先頭から2AHを除き、必要データを取出す。ただし、受信した2AHは2バイトとは限らない(転送開始アドレスが1バイトずれると1個になる)ので、注意が必要である。

ただ、この方法だと転送データがASCIIコードの場合は、データの先頭から2AH('*')をあらかじめ除いておけば有効ですが、任意のバイナリ型式のデータを送る場合は「2AH」が転送データの先頭にくる可能性もあり、十分な対策とはいません。以下の例では、バイナリ型式のデータを転送するため、上記の方法はとらず、複数回の転送の結果を比較して、先頭の1バイト落ちのチェックをすることとします。

3.1 QC-10の場合

CP/M上で実行することにします。

まず、サンプルプログラムとしてRC-20のROMの内容を、RS-232Cを介して出力するRC-20用のプログラム(RS-232C)を作成します。このプログラムは、ROM内の内容を1Kバイト単位でRAM

上に転送してから、その部分を RS-232C を介して出力します。RC-20 の ROM は 8K バイトありますが、そのうちのどの 1K バイトを転送するかはメニューで指定します。

出力が終了すると、RC-20 は再びメッセージを出して入力待ちになります。RC-20 からの転送は、RC-20 の ROM 内サブルーチン RSOUT を使っていますが、この RSOUT で直接転送できるのは、RAM 上のデータに限られています。このため、ROM の内容を 1K バイトだけ、いったん RAM に転送してから、それを RSOUT で RS-232C 出力にしているのです。また、RSOUT は 1K バイトのデータを RS-232C に出力するとき、この 1K バイトの後に最大 22 バイトの不要なデータを付加えて出力しているので、ROM の内容を、連続して上記の方法で転送すると、1K バイトの ROM の情報と次の 1K バイトの ROM の情報との間に、余分なデータが入った型で RS-232C に出力されてしまします。そのため、サンプルプログラムでは一度の転送は 1K バイトに止めその都度パソコンの受信プログラムの方で適当に処理できるようにしています。リストをリスト 18 に示します。

次に RC-20 からの RS-232C の出力データを受信して、RAM 上に格納するプログラム (INRC) を作成します。このプログラムは、RC-20 から 2400 ポートで出力されるデータを RAM の 9000H 以降に格納します。キーボード上の任意の KEY を押すと作業を終了します。リストをリスト 19 に示します。

このプログラムでは RS-232C 用のポートの処理に、BIOS 内のサブルーチン RSIOX を使用しています。このサブルーチンの詳細については「QC-10 日本語処理 CP/M BIOS リファレンス」(技術評論社刊) の第 10 章を参照してください。

さて、実際のデータ転送を行ないます。初めに RC-20 用のプログラム RS232C を 13 ページ「1. QC-10 の場合」で述べた方法で、RC-20 に転送します。次いで、RC-20 を「PROGRAM RUN」のモードにしてプログラムを実行させますと、RC-20 の画面上に「*DATA TRANSFER START OK?」の表示が出ますので、QC-10 側で受信プログラム (INRC) を実行させます。QC-10 側で「READY OK」のメッセージが画面に出ます。RC-20 の「CR」KEY を押すと、RC-20 の画面に、「MODE」を指定する数字の KEY 表示が出るので、0 ~ 7 の数字のうちの 1 つを指定します。この数字は下記のように RC-20 の ROM の転送エリアを指定します。

数字 KEY	転送エリア
0	4000H ~ 43FFH
1	4400H ~ 47FFH
2	4800H ~ 4BFFH
3	4C00H ~ 4FFFH
4	5000H ~ 53FFH
5	5400H ~ 57FFH
6	5800H ~ 5BFFH
7	5C00H ~ 5FFFH

0 ~ 7 の数字を指定すると、データの転送が始まり、RC-20 側からは 2 秒おきに「ピッ」という音

が出来ます。転送が終了すると、RC-20の画面に再び“*DATA TRANSFER START OK?”の表示が出ます。それを確認したら、QC-10の任意のKEYを押して、QC-10側のプログラムを終了させます。この結果、RC-20のROM内の1Kバイトのデータが、QC-10の9000H-93FFHに転送されています。ZSIDなどを使って、逆アセンブルしてください。

RC-20の方は、「CR」KEYを押すと再び転送を行うことができます。転送を再度行わないときは、「MODE」スイッチを押すと、計時モードにもどります。

9000H～93FFH 上に転送したROMの内容のうち、MUSICサブルーチン(42AEH)の部分をZSIDで逆アセンブルしたリストを、リスト20に示します。91ページ「ROM内サブルーチンの使い方」も参照してください。

リスト18 RC-20よりROMの内容を転送するプログラム (RS232C)

```
;      $RS232C$  
;      TRANSFER ROM CONTENT BY RS232C.  
;  
;      MACLIB Z80  
;  
B816 =      PORG    EQU    0BB16H  
426A =      WHIAL   EQU    426AH  
BFE4 =      DISP8   EQU    0BFE4H  
4106 =      HYGGI   EQU    4106H  
421A =      HYOO    EQU    421AH  
4090 =      GETKEY  EQU    4090H  
409A =      Z80END  EQU    409AH  
0008 =      BAUD    EQU    08H      ;2400  
4320 =      RSOUT   EQU    4320H  
4169 =      KDISP3  EQU    4169H  
;  
B816          ORG    PORG  
;  
MAIN:  
B816 00        DB     0  
B817 2A52533233  DB     '$RS232C'$  
;  
START:  
B825 CD6A42  CALL    WHIAL  
B828 21D2B8  LXI    H,DATA1  
B82B 11E4BF  LXI    D,DISPB  
B82E 011C00  LXI    B,28  
LDIR  
B831+EDB0  DB     0EDH,0B0H  
B833 CD0641  CALL    HYGGI  
;  
LOOP1:  
B836 CD9040  CALL    GETKEY  
B839 FE34  CPI    34H
```

B83B C23688	JNZ	LOOP1
;		
B83E 21EE88	LXI	H,DATA2
B841 11E4BF	LXI	D,DISPB
B844 010700	LXI	B,7
	LDIR	
B847+EDB0	DB	.OEDH,0BOH
B849 CD1A42	CALL	HY00
;		
LOOPY:		
B84C CD6941	CALL	KDISP3
B84F CD9040	CALL	GETKEY
	JRZ	LOOPY
B852+2BF8	DB	2BH,LOOPY-\$-1
B854 FE01	CPI	1
	JRZ	LN00
B856+2B1E	DB	2BH,LN00-\$-1
B858 FE02	CPI	2
	JRZ	LN01
B85A+2B21	DB	2BH,LN01-\$-1
B85C FE03	CPI	3
	JRZ	LN02
B85E+2B24	DB	2BH,LN02-\$-1
B860 FE04	CPI	4
	JRZ	LN03
B862+2B27	DB	2BH,LN03-\$-1
B864 FE05	CPI	5
	JRZ	LN04
B866+2B2A	DB	2BH,LN04-\$-1
B868 FE06	CPI	6
	JRZ	LN05
B86A+2B2D	DB	2BH,LN05-\$-1
B86C FE11	CPI	11H
	JRZ	LN06
B86E+2B30	DB	2BH,LN06-\$-1
B870 FE12	CPI	12H
	JRZ	LN07
B872+2B33	DB	2BH,LN07-\$-1
	JR	LOOPY
B874+18D6	DB	18H,LOOPY-\$-1
LN00:		
B876 3E30	MVI	A,'0'
B878 210040	LXI	H,4000H
	JR	LOOP2
B87B+182F	DB	18H,LOOP2-\$-1
LN01:		
B87D 3E31	MVI	A,'1'

B87F 210044	LXI	H,4400H
	JR	LOOP2
B882+1828	DB	18H,LOOP2-\$-1
LN02:		
B884 3E32	MVI	A,'2'
B886 210048	LXI	H,4800H
	JR	LOOP2
B889+1821	DB	18H,LOOP2-\$-1
LN03:		
B888 3E33	MVI	A,'3'
B88D 21004C	LXI	H,4C00H
	JR	LOOP2
B890+181A	DB	18H,LOOP2-\$-1
LN04:		
B892 3E34	MVI	A,'4'
B894 210050	LXI	H,5000H
	JR	LOOP2
B897+1813	DB	18H,LOOP2-\$-1
LN05:		
B899 3E35	MVI	A,'5'
B89B 210054	LXI	H,5400H
	JR	LOOP2
B89E+180C	DB	18H,LOOP2-\$-1
LN06:		
B8A0 3E36	MVI	A,'6'
B8A2 210058	LXI	H,5800H
	JR	LOOP2
B8A5+1805	DB	18H,LOOP2-\$-1
LN07:		
B8A7 3E37	MVI	A,'7'
B8A9 21005C	LXI	H,5C00H
;		
LOOP2:		
B8AC 22F5B8	SHLD	WORK
B8AF 32EABF	STA	DISPB+6
B8B2 CD1A42	CALL	HYDO
B8B5 2AF5B8	LHLD	WORK
B8B8 11F7B8	LXI	D,BUFF
B8BB 010004	LXI	B,0400H
	LDIR	
B8BE+E0B0	DB	0EDH,0B0H
B8C0 CDC6B8	CALL	OUTPUT
;		
B8C3 C325B8	JMP	START
;		
OUTPUT:		
B8C6 3E08	MVI	A,BAUD
B8C8 21F7B8	LXI	H,BUFF

```

BBCB 11F6BC      LXI    D,BUFF+0400H-1
BBCD C02043      CALL   RSOUT
BB01 C9          RET

;
; DATA1:
B802 2A44415441 DB     '$DATA TRANSFER START OK? '
;
; DATA2:
B8EE 2A404F4445 DB     '$MODE= '
;
; WORK:
BBF5 0000        DW     0
;
; BUFF:
BBF7             DS     0
;
; END
BBF7

```

リスト19 QC-10とRC-20からの転送データを受信するプログラム(INRC)

```

;
; *INRC1
; FOR QC-10.
;

0100           ORG    100H
;
F654 =         RSIOX EQU    0F654H
0005 =         B00S  EQU    5
0000 =         BOOT  EQU    0
9000 =         AREA  EQU    9000H
;
0100 11B8C01   LXI    D,MSXXX
0103 CD6301   CALL   MESS
;
; OPEN RS232C.
0106 0610       MVI    B,10H
0108 21EE01   LXI    H,PARLST
010B C054F6   CALL   RSIOX
010E C25A01   JNZ    REAR
;
0111 110090   LXI    D,AREA
;
SIOIN:
;
; *CHECK RECEIVER.
0114 D5         PUSH   D
0115 E5         PUSH   H
0116 0630       MVI    B,30H
0118 C054F6   CALL   RSIOX
011B E1         POP    H
011C D1         POP    D

```

```

0110 C25A01      JNZ    RERR
0120 B7          ORA    A
0121 CA3A01      JZ     SIOIN2
GETD:
;      *GET DATA.
0124 C5          PUSH   B
0125 D5          PUSH   D
0126 E5          PUSH   H
0127 0650        MVI    B,050H
0129 CD54F6        CALL   RSIOX
012C E1          POP    H
012D D1          POP    D
012E C1          POP    B
012F C25A01      JNZ    RERR
0132 12          STAX   D
0133 13          INX    D
0134 0B          DCX    B
0135 78          MOV    A,B
0136 B1          ORA    C
0137 C22401      JNZ    GETD
SIOIN2:
013A C5          PUSH   B
013B D5          PUSH   D
013C E5          PUSH   H
013D CD4A01      CALL   KEYIN
0140 E1          POP    H
0141 D1          POP    D
0142 C1          POP    B
0143 B7          ORA    A
0144 CA1A01      JZ     SIOIN
0147 C37B01      JMP    ENDL
;
014A 0E06        KEYIN  MVI    C,6
014C 1EFF        MVI    E,0FFH
014E C30500      JMP    BDOS
;
0151 F5          CONOUT PUSH   PSM
0152 0E02        MVI    C,2
0154 5F          MOV    E,A
0155 CD0500      CALL   BDOS
0158 F1          POP    PSM
0159 C9          RET
RERR:
015A 116901      LXI    D,MSERR
MSEND:
015D CD6301      CALL   MESS
0160 C30000      JMP    BOOT

```

```

;
MESS:
0163 0E09    MVI    C,9
0165 CD0500    CALL   BDOS
0168 C9        RET
MSERR:
0169 000A202A52    DB     13,10,'$READ ERROR$',13,10,'$'
ENDL:
;
CLOSE RS232C.
017B 0620    MVI    B,020H
017D CD54F6    CALL   RSI0X
;
0180 11C501    Lxi    d,sssss
0183 CD6301    call   merr
0186 11B601    LXI    D,MS000
0189 C35D01    JMP    MSEND
MSXXX:
018C 000A202A52    DB     13,10,'$RS232C READER START.',13,10
01A6 000A202A72    db     13,10,'$ready ok',13,10
01B5 24        db     '$'
MS000:
01B6 000A202A4A    DB     13,10,'$JOB END$',13,10,'$'
MSNNN:
01C5 202020441    DB     'DATA IS LOADED IN MEMORY FROM 9000H'
01EB 000A24    DB     13,10,'$'
PARLST:
01EE F701    DW     BUFR ;BUFFER ADDRESS.
01F0 0008    DW     2048 ;BUFFER SIZE.
01F2 0C        DB     0CH  ;2400 BPS.
01F3 03        DB     3    ;8 BIT/CHAR.
01FA 00        DB     0    ;NON PARITY.
01F5 03        DB     3    ;STOP BIT=2.
01F6 FF        DB     OFFH
;
BUFR:
01F7        DS     2048
;
09F7        END

```

リスト20 MUSICサブルーチン(42AEH)の逆アセンブルリスト
(4000Hは9000Hに対応している)

```
#L92AE
92AE CP 20
92B0 RET NC
92B1 CALL 40B6 .....レジスタSAVE
92B4 LD HL,42C6
92B7 OR A
92B8 JR Z,04
92B9 INC HL
92B8 DEC A
92BC JR F9
92BE LD A,(HL)
92BF LD C,50
#L
92C1 OUT (C),A } .....MUSICポートへの出力 { レジスタA: 音階コード
92C3 JP 43CF
92C6: 82 B9 BF 96 9B A1 A7 AB B0 B5
        . . . . .
92D0: B9 BD C0 C4 C7 CA CD D0 D3 D5 D8 DA DC DE E0 00
        . . . . .
        :
#L93CF
93CF CALL 40C6 .....レジスタLOAD
93D2 RET
```

3.2 PC-8001/8001mkIIの場合

20ページ2.3-2の場合と同様にCP/Mを使用する場合とDUADPCを使用する場合について説明します。

1. CP/Mの場合

RC-20で使用するROM内容の転送プログラムとしては、QC-10の場合と同様、リスト18を使用します。このプログラムのRC-20への転送は20ページ2.3-2で使用したTRANSを使って、次のように行ないます。

A>TRANS RS232C. HEX

次にPC-8001/8001mkII側での受信プログラムとしては、INRCを使用しています。このプログラムの機能は、QC-10の場合のINRCと同じです。リストをリスト21に示します。

転送方法もQC-10の場合と同様にして行ないます。ただし、INRCプログラムを終了させる時、QC-10の場合は任意のキーを押しましたが、PC-8001の場合はRETURNキーを押すことにします。

リスト21 RC-20からの転送データを受信するプログラム(INRC)

```
;          ;*INRC*  
;          FOR PC-8001.  
;  
0100      ORG    100H  
;  
0005 =    B005 EQU    5  
0000 =    B000 EQU    0  
00CE =    MODE   EQU    0CEH  
0020 =    PORT0  EQU    20H  
0021 =    PORT1  EQU    PORT0+1  
9000 =    AREA   EQU    9000H  
;  
0100 111401    LXI    D,MSXXX  
0103 C09D01    CALL   MESS  
;  
0106 C06501    CALL   INIT  
0109 210090    LXI    H,AREA  
LOOP:  
010C C07A01    CALL   SIOIN  
010F 77        MOV    M,A  
0110 23        INX    H  
0111 C30C01    JMP    LOOP  
;
```

```

MSXXX:
0114 0D0A202A52    DB    13,10,'#RS232C READER START.',13,10
012E 202A524541    DB    ' #READY OK!',13,10,'$'
MSNNN:
013C 2020204441    DB    ' DATA IS LOADED IN MEMORY FROM 9000H'
0162 0D0A24        DB    13,10,'$'
;
INIT:
0165 3E00          MVI A,0
0167 D321          OUT PORT1
0169 D321          OUT PORT1
016B D321          OUT PORT1
016D 3E40          MVI A,40H
016F D321          OUT PORT1
0171 3ECE          MVI A,MODE
0173 D321          OUT PORT1
0175 3E15          MVI A,15H
0177 D321          OUT PORT1
0179 C9            RET
;
;
017A DB21          SIOIN IN PORT1
017C 1F            RAR
017D 1F            RAR
017E D28901         JNC   SIOIN2
0181 E60E          ANI   OEH
0183 C29301         JNZ   RERR
0186 DB20          IN    PORT0
0188 C9            RET
SIOIN2:
0189 DB01          IN    IH
018B FE7F          CPI   7FH    :RET KEY CHECK
018D C27A01         JNZ   SIOIN
0190 C3B401         JMP   ENOL
;
;
RERR:
0193 11A201         LXI   D,MSERR
MSEND:
0196 CD9C01         CALL  MESS
0199 C30000         JMP   BOOT
;
MESS:
019C 0E09          MVI   C,9
019E CD0500         CALL  BDOS
01A1 C9            RET
MSERR:
01A2 0D0A202A52    DB    13,10,'#READ ERROR!',13,10,'$'

```

```

        ENDL:
01B4 113C01    LXI    D,MSNNN
01B7 C09C01    CALL   MESS
01BA 11C001    LXI    D,MS000
01BD C39601    JMP    MSEND
        MS000:
01C0 0D0A202A4A    DB     13,10,'$JOB END$',13,10,'$'
01CF          END

```

2. DUADPCの場合

RC-20で使用するROM内容の転送プログラムは、CP/Mの場合と同様です。リストをリスト22に示します。このプログラムを28ページで使用した転送プログラム（リスト5）でRC-20に転送します。

PC-8001側の受信プログラム（INRC）のリストをリスト23に示します。このプログラムをアセンブルして、INRC.HEXを作成します。転送手順は次の通りです。

- ① RC-20を「PROGRAM RUN?」モードにして、先に転送しておいたRS232Cプログラムを実行させます。
画面に“* DATA TRANSFER START OK?”の表示が出ます。
- ② DUADPCのLOADERで、INRC.HEXをメモリ上にロードします。画面に“8500—85E4 OK”的表示が出て、BASICモードになります。
- ③ MONコマンドでモニタモードに入り *G8500でINRCを実行させます。
“* RS232C READER START *READY OK!”の表示が出ますので、RC-20の「CR」キーを押すと、MODEの指定になります。0～7のうちの数字を指定すると、転送が始まります。
- ④ 転送が終了すると、RC-20の画面に再び“*DATA TRANSFER START OK?”の表示が出るので、PC側では任意のKEYを押すとプログラムは終了し、モニタモードにもどります。モニタのロコマンドで9000H～93FFHをダンプしてみると、RC-20のROMの内容が1Kバイト転送されているのがわかります。
- ⑤ RC-20側では、「CR」KEYを押すと再びモード指定に入り、ROMの内容の転送を始めることができます。「MODE」KEYを押すと計時モードにもどります。

リスト22 RC-20からROMの内容を送信するプログラム(RS232C)

```

;***RS232C***  

; FOR RC-20.  

B816    PORG: EQU 0B816H  

426A    WHIAL: EQU 426AH  

BFE4    DISPB: EQU 0BFE4H  

4106    HY06I: EQU 4106H

```

```

421A      HYDO: EQU 421AH
4090      GETKEY: EQU 4090H
409A      Z00END: EQU 409AH
0008      BAUD: EQU 0BH
4320      RSOUT: EQU 4320H
4169      KDISP3: EQU 4169H
;
;          ORG PORG
;
8816 00    MAIN: DB 0
8817 2A525332   DB '$RS232C$'
8818 3332432A
881F 20202020
8823 2020
8825 1805      JR ENTRY
8827 00        DB 0
882B 1688      DW MAIN
882A E100      DW PRGEND-MAIN
;
882C      ENTRY:
882C CD6A42      CALL WHIAL
882F 21D4BB      LD HL, DATA1
8832 11E4BF      LD DE, DISPB
8835 011C00      LD BC, 28
8838 ED80      LDIR
883A CD0641      CALL HYOGI
883D CD9040      LOOP1: CALL GETKEY
8840 FE34      CP 34H
8842 20F9      JR NZ,LOOP1
;
8844 21F0BB      LD HL, DATA2
8847 11E4BF      LD DE, DISPB
884A 010700      LD BC, 7
884D ED80      LDIR
884F CD1A42      CALL HYDO
;
8852 CD6941      LOOPY: CALL KDISP3
8855 CD9040      CALL GETKEY
8858 28FB      JR Z, LOOPY
885A FE01      CP 1
885C 281E      JR Z, LN00
885E FE02      CP 2
8860 2821      JR Z, LN01
8862 FE03      CP 3
8864 2824      JR Z, LN02
8866 FE04      CP 4
8868 2827      JR Z, LN03

```

B86A FE05 CP 5
B86C 282A JR Z,LN04
B86E FE06 CP 6
B870 282D JR Z,LN05
B872 FE11 CP 11H
B874 2830 JR Z,LN06
B876 FE12 CP 12H
B878 2833 JR Z,LN07
B87A 18D6 JR LOOPY
B87C 3E30 LN00: LD A,'0'
B87E 210040 LD HL,4000H
B881 182F JR LOOP2
B883 3E31 LN01: LD A,'1'
B885 210044 LD HL,4400H
B888 1828 JR LOOP2
B88A 3E32 LN02: LD A,'2'
B88C 210048 LD HL,4800H
B88F 1821 JR LOOP2
B891 3E33 LN03: LD A,'3'
B893 21004C LD HL,4C00H
B896 181A JR LOOP2
B898 3E34 LN04: LD A,'4'
B89A 210050 LD HL,5000H
B89D 1813 JR LOOP2
B89F 3E35 LN05: LD A,'5'
B8A1 210054 LD HL,5400H
B8A4 180C JR LOOP2
B8A6 3E36 LN06: LD A,'6'
B8A8 210058 LD HL,5800H
B8AB 1805 JR LOOP2
B8AD 3E37 LN07: LD A,'7'
B8AF 21005C LD HL,5C00H
B8B2 E5 LOOP2: PUSH HL
B8B3 32EABF LD (DISPB+6),A
B8B6 CD1A42 CALL HYOO
B8B9 E1 POP HL
B8BA 11F7BB LD DE,BUFF
B8BD 010004 LD BC,0400H
BBC0 ED80 LDIR
BBC2 CDC888 CALL OUTPUT
BBC5 C32C88 JP ENTRY
;
;
BBC8 3E0B OUTPUT:LD A,BAUD
BBCA 21F7BB LD HL,BUFF
BBCD 11F6BC LD DE,BUFF+0400H-1
BBD0 CD2043 CALL RSOUT
BBD3 C9 RET

```

;          ;
B804 2A444154 DATA1: DB    '$DATA TRANSFER START OK?    '
B808 41205452
B80C 414E5346
B80E 45522053
B814 54415254
B818 204F4B3F
B81C 20202020
B8F0 2A4D4F44 DATA2: DB    '$MODE='
B8F4 453D20
B8F7      BUFF: DS    0
;
B8F7      PRGEND:
;
B8F7      END

```

リスト23 RC-20からの転送データを受信するプログラム(INRC)

```

;***INRC***
;
ORG 8500H
;
SC66    MON: EQU SC66H
OFAC    KEYIN: EQU OFACH
00CE    MODE: EQU OCEH
0020    PORT0: EQU 20H
0021    PORT1: EQU PORT0+1
9000    AREA: EQU 9000H
0257    DISP: EQU 0257H
EA66    PORT30: EQU 0EA66H
0030    PORT0: EQU 30H
;
8500 211FB5      LD   HL,MSXXX
8503 CDABBB      CALL MESS
;
;#SID SELECT
8506 3A66EA      LD   A,(PORT30)
8509 E60F         AND  0FH
850B F620         OR   20H
850D 3266EA      LD   (PORT30),A
8510 D330         OUT  (PORT0),A
;
8512 CD7085      CALL INIT
8515 210090      LD   HL,AREA
8518 CD8585      LOOP: CALL SIOIN
851B 77           LD   (HL),A

```

851C 23 INC HL
851D 18F9 JR LOOP
:
851F 000A202A MSXXX: DB '13,10,'#RS232C READER START.',13,10
8523 52533233
8527 32432052
852B 45414445
852F 52205354
8533 4152542E
8537 000A
8539 202A5245 DB 'READY OK!',13,10,'\$'
853D 41445920
8541 4F4B210D
8545 0A24
8547 20204441 MSNNH: DB 'DATA IS LOADED'
854B 54412049
854F 53204C4F
8553 41444544
8557 20494E20 DB 'IN MEMORY FROM 9000H.',13,10,'\$'
855B 4D454D4F
855F 52592046
8563 524F4D20
8567 39303030
856B 4B2E0D0A
856F 24
:
8570 3E00 INIT: LD A,0
8572 D321 OUT (PORT1),A
8574 D321 OUT (PORT1),A
8576 D321 OUT (PORT1),A
8578 3E40 LD A,40H
857A D321 OUT (PORT1),A
857C 3ECE LD A,MODE
857E D321 OUT (PORT1),A

8580 3E15 LD A,15H
8582 D321 OUT (PORT1),A
8584 C9 RET
:
8585 DB21 S10IN: IN A,(PORT1)
8587 1F RRA
8588 1F RRA
8589 3007 JR NC,S10IN2
858B E60E AND 0EH
858D 2013 JR NZ,RERR

```
858F DB20      IN A,(PORT0)
8591 C9      RET
;
8592 C5      SIDIN2: PUSH BC
8593 D5      PUSH DE
8594 E5      PUSH HL
8595 CDACOF   CALL KEYIN
8598 E1      POP HL
8599 D1      POP DE
859A C1      POP BC
859B 28E8    JR Z,SIDIN
859D 38E6    JR C,SIDIN
859F F1      POP AF
85A0 1B25    JR ENDL
;
;
85A2 RERR: LD HL,MSERR
85A2 21B585   MSEND: CALL MESS
85A5 CDAB85   JP MON
;
85A8 C3665C   MESS: LD A,(HL)
85AB 7E      CP '$'
85AC FE24   RET Z
85AE CB      CALL DISP
85AF CD5702   INC HL
85B2 23      JR MESS
85B3 1BF6    LD A,(HL)
;
85B5 000A202A MSERR: DB 13,10,'$READ ERROR$',13,10,'$'
85B9 52454144
85BD 20455252
85C1 4F522A0D
85C5 0A24
85C7 214785   ENDL: LD HL,MSNNN
85CA CDAB85   CALL MESS
85CD 216685   LD HL,MS000
85D0 CDAB85   CALL MESS
85D3 C3665C   JP MON
85D6 000A202A MS000: DB 13,10,'$JOB END$',13,10,'$'
85DA 4A4F4220
85DE 454E442A
85E2 000A24
;
85E5 END
```

3.3 PC-8801/8801mk IIの場合

N-BASICモードに設定した場合は、66ページ「PC-8001/8001mk II」のところで述べた方法が、そのまま使用できます。

ここでは、N88-BASICモードに設定した場合の方法について説明します。

1. CP/Mの場合

N88-BASICモードで走るCP/Mの場合も、66ページ「1.CP/Mの場合」のところで述べた方法が、そのまま使用できます。

2. DUAD88の場合

RC-20からROMの内容を、一度に1Kバイト転送するプログラムを、リスト24に示します。先述のRS-232Cプログラム（リスト22）とはほとんど同じですが、2か所ほど異なっています。これは、OUTPUTサブルーチンで、1Kバイトのデータを転送する前に、4バイトのデータを転送するようにした点です。この4バイトは、受信プログラムのBLOAD文で使用するもので、先の2バイトは0000H、次の2バイトは転送データの長さ1Kバイトを示す0400Hが入っています。

このプログラムを、35ページ「DUAD88による場合」で示した方法によってアセンブルし、RS232C HEXを作り、さらにRS232C BINを作ります（必ずOFFSETとして1000を指定してください）。

RC-20への転送は、前と同様に、「TRANS」プログラム（リスト14）で行います。

受信プログラムを、リスト25に示します。RC-20からの転送データをマシン語ファイルとみなし、先に述べたように4バイトのデータを先頭に付けて送っているので、受信プログラムは非常に簡単になります。受信したデータは、C000H～C3FFHにロードされ、1Kバイトデータを受信するとプログラムは終了します。

転送の手順は、以下のようになります。

- ① RC-20に転送しておいたプログラム（RS232C）を、RUNさせます。"DATA TRANSFER START OK?"と、RC-20の画面に表示されます。
- ② PC側でINRC（リスト25）を実行させます。"*READER START"のメッセージが出ます。
- ③ RC-20の「CR」キーを押し、続いて0～7のうちの1数字を押すと転送が始まります。
- ④ 転送が終わると、RC-20は再び"DATA TRANSFER START OK?"と表示されて、PC側はプログラムが終了します。RC-20の「MODE」キーを押すと、RC-20は計時モードとなります。

```

;***RS232C***
; FOR RC-20.
B816    PORG: EQU 08B16H
426A    WHIAL: EQU 426AH
BFE4    DISP8: EQU 0BFE4H
4106    HYOGI: EQU 4106H
421A    HY00: EQU 421AH
4090    GETKEY: EQU 4090H
409A    Z80END: EQU 409AH
0008    BAUD: EQU 08H
4320    RSOUT: EQU 4320H
4169    KDISP3: EQU 4169H
;
;      ORG PORG
;
B816 00    MAIN: DB 0
B817 2A525332   DB '#RS232C$'
B818 3332432A
B81F 20202020
B823 2020
B825 1805      JR ENTRY
B827 00          DB 0
B828 16BB      DW MAIN
B82A E500      DW PRGEND-MAIN
;
B82C    ENTRY:
B82C C06A42      CALL WHIAL
B82F 21D4BB      LD HL,DATA1
B832 11E4BF      LD DE,DISPB
B835 011C00      LD BC,28
B838 EDB0      LDIR
B83A C00641      CALL HYOGI
B83D CD9040      LOOP1: CALL GETKEY
B840 FE34      CP 34H
B842 20F9      JR NZ,LOOP1
;
B844 21FOBB      LD HL,DATA2
B847 11E4BF      LD DE,DISPB
B84A 010700      LD BC,7
B84D EDB0      LDIR
B84F C01A42      CALL HY00
;
B852 C06941      LOOPY: CALL KDISP3
B855 CD9040      CALL GETKEY
B858 28FB      JR Z,LOOPY

```

B85A FE01	CP	1
B85C 281E	JR	Z,LN00
B85E FE02	CP	2
B860 2821	JR	Z,LN01
B862 FE03	CP	3
B864 2824	JR	Z,LN02
B866 FE04	CP	4
B868 2827	JR	Z,LN03
B86A FE05	CP	5
B86C 282A	JR	Z,LN04
B86E FE06	CP	6
B870 282D	JR	Z,LN05
B872 FE11	CP	11H
B874 2830	JR	Z,LN06
B876 FE12	CP	12H
B878 2833	JR	Z,LN07
B87A 1806	JR	LOOPY
B87C 3E30	LN00:	LD A,'0'
B87E 210040	LD	HL,4000H
B881 182F	JR	LOOP2
B883 3E31	LN01:	LD A,'1'
B885 210044	LD	HL,4400H
B888 182B	JR	LOOP2
B88A 3E32	LN02:	LD A,'2'
B88C 210048	LD	HL,4800H
B88F 1821	JR	LOOP2
B891 3E33	LN03:	LD A,'3'
B893 21004C	LD	HL,4C00H
B896 181A	JR	LOOP2
B898 3E34	LN04:	LD A,'4'
B89A 210050	LD	HL,5000H
B89D 1813	JR	LOOP2
B89F 3E35	LN05:	LD A,'5'
B8A1 210054	LD	HL,5400H
B8A4 180C	JR	LOOP2
B8A6 3E36	LN06:	LD A,'6'
B8A8 210058	LD	HL,5800H
B8AB 1805	JR	LOOP2
B8AD 3E37	LN07:	LD A,'7'

```
BBC5 C32CB8      JP ENTRY
;
;
BBC8 3E08      OUTPUT:LD A,BAUD
BBCA 21F7B8      LD HL,BUFF-4
BBCD 11FABC      LD DE,BUFF+0400H-1
BB00 CD2043      CALL R$OUT
BB03 C9          RET
;
BBC4 2A444154 DATA1: DB    '$DATA TRANSFER START OK? '
BBC8 41205452
BBCD 414E5346
BBE0 45522053
BBE4 54415254
BBE8 204F4B3F
BBEC 20202020
BBF0 2A4D4F44 DATA2: DB    '$MODE='
BBF4 453D20
BBF7 00000004 XXXX: DB    0,0,0,4
BBFB           BUFF: DS    0
;
BBFB           PRSEND:
;
BBFB           END
```

リスト25

```
100 REM $inrc$1
110 CLEAR ,&HC000
120 PRINT "$reader start."
130 BLOAD "COM1:N82N",&HC000
140 PRINT "$data is loaded in &hc000-&hc3ff."
150 PRINT "$job end."
160 END
```

3.4 HC-40/80/88の場合

40ページ「5.HC-20の場合」と同様に、CP/Mの下で行うこととします。RC-20で使用するROM内容の転送プログラム(RS232C)と受信用のプログラム(INRC)は、他の機種の場合と機能的にはまったく同一です。

RS232Cをリスト26に、INRCをリスト27にそれぞれ示します。

INRCプログラムでは、RS232Cの入力処理にTRANS2(リスト17)と同様、BIOS内のRS-232C制御ルーチンRSIOXを使用しています。

RS232CプログラムのRC-20への転送は、40ページ「5.HC-20の場合」で述べた転送プログラム(TRANS/TRANS2)を使って、次のように行います。

A>TRANS RS232C.COM

HC-40/80/88側での受信プログラムINRCの動作は、QC-10の場合と同じです。転送方法も、QC-10の場合と同様に行います。

転送されたRC-20から1Kバイトのデータは、HC-40/80/88のメモリの4000H～43FFHに格納されます。

リスト26 RC-20からROMの内容を送信するプログラム

```
D>b:asm80 rs232c.asm
ASM80(Z80) 0: 1:17/ 1:10:14      1
100 0000      ; *RS232C*
120 0000      ;
140 0100      ORG 100H
160 0100      ;
180 8816      MAIN EQU 0BB16H
200 426A      WHIAL EQU 426AH
220 BFE4      DISP8 EQU 0BFE4H
240 4106      HYOGI EQU 4106H
260 421A      HYODI EQU 421AH
280 4169      KDISP3 EQU 4169H
300 4090      GETKEY EQU 4090H
320 409A      Z80END EQU 409AH
340 0008      BAUD EQU 08H ;2400 BAUD
360 4320      RSOUT EQU 4320H
380 0100      ;
400 0100      ;MAIN:
```

420 0100	START:DS	0
440 0100 00	DB	0
460 0101 2A727332	DB	'#rs232c'
480 010F 1805	JR	ENTRY
500 0111 00	DB	0
520 0112 168B	DW	MAIN
540 0114 E100	DW	PRGEND-START
560 0116	;	
580 0116	ENTRY:DS	0
600 0116 CD6A42	CALL	WHIAL
620 0119 2104BB	LD	HL,DATA1-START+MAIN
640 011C 11E4BF	LD	DE,DISPB
660 011F 011C00	LD	BC,28
680 0122 E0B0	LDIR	
700 0124 CD0641	CALL	HYOGI
720 0127 CD9040	LOOP1:CALL	GETKEY
740 012A FE34	CP	34H
760 012C 20F9	JR	NZ,LOOP1
780 012E	;	
800 012E 21FOBB	LD	HL,DATA2-START+MAIN
820 0131 11E4BF	LD	DE,DISPB
840 0134 010700	LD	BC,7
860 0137 E0B0	LDIR	
880 0139 CD1A42	CALL	HYDO
900 013C	;	
920 013C CD6941	LOOPY:CALL	KDISP3
940 013F CD9040	CALL	GETKEY
960 0142 28FB	JR	Z,LOOPY
980 0144 FE01	CP	1
1000 0146 261E	JR	Z,LN00
1020 0148 FE02	CP	2
1040 014A 2821	JR	Z,LN01
1060 014C FE03	CP	3
1080 014E 2824	JR	Z,LN02
1100 0150 FE04	CP	4
1120 0152 2827	JR	Z,LN03
1140 0154 FE05	CP	5
1160 0156 282A	JR	Z,LN04
1180 0158 FE06	CP	6
1200 015A 282D	JR	Z,LN05
1220 015C FE11	CP	11H
1240 015E 2830	JR	Z,LN06
1260 0160 FE12	CP	12H
1280 0162 2833	JR	Z,LN07
1300 0164 18D6	JR	LOOPY
1320 0166 3E30	LN00:LD	A,'0'

1340 0168 210040	LD	HL,4000H
1360 0169 182F	JR	LOOP2
1380 016D 3E31	LN01:LD	A,'1'
1400 016F 210044	LD	HL,4400H
1420 0172 1828	JR	LOOP2
1440 0174 3E32	LN02:LD	A,'2'
1460 0176 21004B	LD	HL,4800H
1480 0179 1821	JR	LOOP2
1500 017B 3E33	LN03:LD	A,'3'
1520 017D 21004C	LD	HL,4C00H
1540 0180 181A	JR	LOOP2
1560 0182 3E34	LN04:LD	A,'4'
1580 0184 210050	LD	HL,5000H
1600 0187 1813	JR	LOOP2
1620 0189 3E35	LN05:LD	A,'5'
1640 018B 210054	LD	HL,5400H
1660 018E 180C	JR	LOOP2
1680 0190 3E36	LN06:LD	A,'6'
1700 0192 210058	LD	HL,5800H
1720 0195 1805	JR	LOOP2
1740 0197 3E37	LN07:LD	A,'7'
1760 0199 21005C	LD	HL,5C00H
1780 019C E5	LOOP2:PUSH HL	
1800 019D 32EABF	LD	(DISPB+6),A
1820 01A0 CD1A42	CALL	HY00
1840 01A3 E1	POP	HL
1860 01A4 11F7B8	LD	DE,BUFF-START+MAIN
1880 01A7 010004	LD	BC,0400H
1900 01AA EDB0	LDIR	
1920 01AC CDCB88	CALL	OUTPUT-START+MAIN
1940 01AF ;		
1960 01AF C32CB8	JP	ENTRY-START+MAIN
1980 01B2 ;		
2000 01B2 3E08	OUTPUT:LD	A,BAUD
2020 01B4 21F7B8	LD	HL,BUFF-START+MAIN
2040 01B7 11F6BC	LD	DE,BUFF-START+MAIN+0400H-1
2060 01BA CD2043	CALL	RSOUT
2080 01BD C9	RET	
2100 01BE ;		
2120 01BE 2A646174	DATA1:DB	'#data transfer start ok?
2140 01DA 2A6D6F64	DATA2:DB	'%node= '
2160 01E1	BUFF:DS	0
2180 01E1	;	
2200 01E1	PRGEND:DS	0
2220 01E1	;	
2240 01E1	END	

```

MAIN 8B16 WHIAL 426A DISPB BFE4
HYOGI 4106 HYDO 421A KOISP3 4159
GETKEY 4090 Z80END 409A BAUD 0008
RSOUT 4320 START 0100 ENTRY 0116
PRGEND 01E1 DATA1 01BE LOOP1 0127
DATA2 01DA LOOPY 013C LN00 0166
LN01 0160 LN02 0174 LN03 017B
LN04 0182 LN05 0189 LN06 0190
LN07 0197 LOOP2 019C BUFF 01E1
OUTPUT 01B2

```

リスト27 RC-20からの転送データを受信するプログラム(INRC)

```

D:\brass\80\inrc.asm
ASMB0(Z80) 0: 1: 2/ 1:10:18      1
100 0000      ; $INRC$
110 0000      ;
120 0100      ORG 100H
130 0100      ;
140 0005      BDOS EQU 5
150 0000      BOOT EQU 0
160 4000      AREA EQU 4000H
170 0100      ;
180 0100 118501 LD DE,MSXXX
190 0103 CD5001 CALL MESS
200 0106      ;
210 0106 2A0100 LD HL,(BOOT+1)
220 0109 015100 LD BC,51H
230 010C 09    ADD HL,BC
240 010D 22EE01 LD (RS10X+1),HL
250 0110      ;
260 0110 0610 LD B,10H ;OPEN RS232C
270 0112 21E401 LD HL,PARLIST
280 0115 CDE001 CALL RS10X
290 0118 203A JR NZ,RERR
300 011A      ;
310 011A 110040 LD DE,AREA

```

320 0110	SIOIN:DS	0
330 011D D5	PUSH	DE
340 011E E5	PUSH	HL
350 011F 0630	LD	B,30H ;CHECK RECEIVER
360 0121 CDED01	CALL	RSIOX
370 0124 E1	POP	HL
380 0125 D1	POP	DE
390 0126 202C	JR	NZ,RERR
400 012B B7	OR	A
410 0129 2814	JR	Z,SIOIN2
420 012B	GETD:DS	0
430 012B C5	PUSH	BC
440 012C D5	PUSH	DE
450 012D E5	PUSH	HL
460 012E 0650	LD	B,50H ;GET DATA
470 0130 CDED01	CALL	RSIOX
480 0133 E1	POP	HL
490 0134 D1	POP	DE
500 0135 C1	POP	BC
510 0136 201C	JR	NZ,RERR
520 0138 12	LD	(DE),A
530 0139 13	INC	DE
540 013A 0B	DEC	BC
550 013B 7B	LD	A,B
560 013C B1	OR	C
570 013D 20EC	JR	NZ,GETD
580 013F	SIOIN2:DS	0
590 013F C5	PUSH	BC
600 0140 D5	PUSH	DE
610 0141 E5	PUSH	HL
620 0142 CD4D01	CALL	KEYIN
630 0145 E1	POP	HL
640 0146 D1	POP	DE
650 0147 C1	POP	BC
660 0148 B7	OR	A
670 0149 28D2	JR	Z,SIOIN
680 014B 1828	JR	ENDL
690 014D	;	
700 014D 0E06	KEYIN:LD	C,6
710 014F 1EFF	LD	E,OFFH
720 0151 C30500	JP	BDOS
730 0154	;	
740 0154 116301	RERR:LD	DE,MSERR
750 0157 C05101	MSEND:CALL	MESS
760 015A C30000	JP	BOOT
770 015B	;	
780 015D 0E09	MESS:LD	C,9

```

790 015F CD0500      CALL    BDOS
800 0162 C9          RET
810 0163 0D0A202A    MSERR:DB 13,10,'$read error$',13,10,'$'
820 0175 0620        ENDL:LD  B,20H ;RS232C CLOSE
830 0177 CDE001      CALL    RSIOX
840 017A              ;
850 017A 11BC01      LD      DE,MSNNN
860 017D CD5D01      CALL    MESS
870 0180 11AF01      LD      DE,MS000
880 0183 1BD2        JR      MSEND
890 0185 0D0A202A    MSXXX:DB 13,10,'$rs232c reader start.',13,10
900 019F 0D0A202A    DB      13,10,'$ready ok!',13,10,'$'
910 01AF 202A6A6F    MS000:DB '$job end$',13,10,'$'
920 01BC 20206461    MSNNN:DB '$ data is loaded in memory from 4000h',13,10,
930 01E4              PARLIST:DS 0
940 01E4 F101        DW      BUFFR
950 01E6 000B        DW      2048
960 01E8 0C          DB      0CH
970 01E9 03          DB      3
980 01EA 00          DB      0
990 01EB 03          DB      3
1000 01EC FF         DB      OFFH
1010 01ED              ;
1020 01ED CD0000      RSIOX:CALL 0000H
1030 01FO C9          RET
1040 01F1              BUFFR:DS 2048
1050 09F1              ;
1060 09F1              END
BDOS 0005 BOOT 0000 AREA 4000
MSXXX 0185 MESS 015D RSIOX 01ED
PARLIS 01E4 RERR 0154 SIOIN 011D
SIOIN2 013F GETD 012B KEYIN 014B
ENDL 0175 MSERR 0163 MSEND 0157
MSNNN 01BC MS000 01AF BUFFR 01F1

```

NUMBER

4

ハード構成
とメモリマップ

RC-20のハードウェアのブロック図を図4-1に、機能概要を表4-1に示します。SMC 84C00はZ-80相当のC-MOS CPUです。4ビットCPUは主として、キー入力、液晶表示、タイマー等の周辺の制御の機能を受け持っています。

RC-20のメモリマップを図4-2に示します。ユーザの使用できるエリアはB800H～BF3FHの1856(740H)バイトです。

文字表示用のデータ・バッファはBFE4Hから始まる28バイトで、液晶上の表示位置にそれぞれ対応しています。液晶表示面はページ0からページ3の4つのページから構成され、表示はページ単位で行なわれます。

この関係を図4-3に示します。

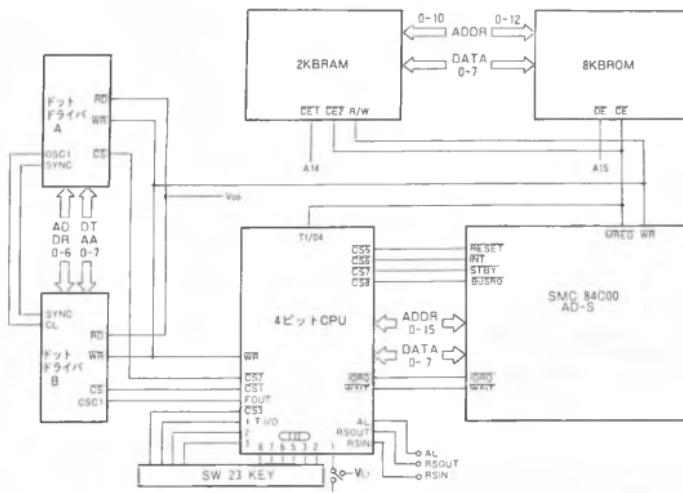
RC-20には下部の5個の固定キーのはかに液晶表示部に、 $6 \times 3 = 18$ 個の文字表示位置がタッチキー(パネルスイッチ)となっています。この関係を図4-4に示します。

さらにRC-20には42×32ドットのグラフィック表示の機能がありますが、このグラフィック・データを書き込むVRAMがBFBAHからBFE3Hの42バイトに配置されています。ドットとVRAMの対応を図4-5に示します。

RC-20では表示できる文字が $12 \times 16 = 192$ 種類ありますが、それらのキャラクタ・コードの一覧を図4-6に示します。

RC-20のユーザのワークエリアはBF70HからBF9FHまで、ユーザが自由に使うことができます。ただし、演算サブルーチンでは一部のユーザワークエリアを使用します。

図4-1



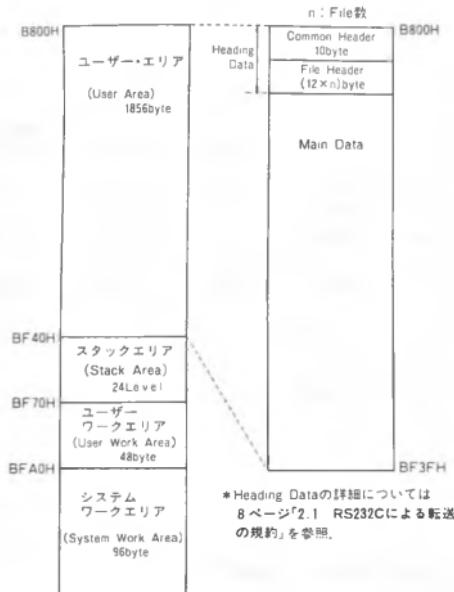
4ビット CPU	Time Standard CPU * 計時機能 * I/O処理機能
SMC84C00	Zilog Z-80® コンバチ 8ビット CPU C-MOS版
2KB RAM	16Kbit C-MOS Static RAM (低電圧バージョン)
8KB ROM	64Kbit C-MOS Mask ROM (低電圧バージョン)
ドットドライバ A, B	Dot Driver (低電圧バージョン)

表4-1 RC-20機能概要

項目	内 容
1. コンピュータ	SMC84CC0 (Z-80コンパチブル) 8ビットCPU
2. メモリ容量	ROM 8Kバイト RAM 2Kバイト
3. 機能	コンピュータ機能 マシン語プログラム A/P・時・分・秒・年・月・日・曜 月末自動修正（2月28日制オートカレンダー） 時分設定型 鳴鐘時間：20sec 鳴り止め機能 休止・復元機能 アラーム試聴機能 報時時間：1/32sec 休止・復元機能 RS-232C 双方向 ただし RC-20→パソコン：レベルシフト必要 転送速度：2400bps ソフトにより1200bps・600bps・300bpsも可能
4. 効作温度範囲	0°C ~ +50°C
5. 操作方式	タッチキー方式 • 固定キー 5キー • Touch Switch On Panel 18キー
6. 時間精度	±20sec以内/月
7. 振動子	音叉型水晶振動子 (32,768Hz)
8. 電池	電池寿命 約1年 (アラーム20秒/1日鳴鐘) (コンピュータ30分/1日使用) 使用電池 リチウム電池BR2325
9. スピーカ	庄電スピーカ
10. 表示	表示方式 FE型 白色液晶 表示内容 42×32全面ドット (7文字×4行) 表示寿命 5年以上

图 4-2

メモリマップ



システムワークエリアの詳細

(a) マップ

	BFA0H	BFB0H	BFC0H	BFD0H	BFE0H	BFF0H
0	システム 予約 (1byte)	FRDCT				
1		SAHL				
2						
3						
4		INTSET				DBUFF
5		KEYBU				(28byte)
6		ALFLAG				
7		INTBF				
8						
9						
A		INTFL				
B	FTPAD	VRAM				
C		(42byte)				
D	FEDAD					
E						
F	FLNGT					

(b) 詳細		Level	機能	Level	機能
		FTPAD	File Data TOP ADDRESS		SAHL HReg のスタックエリア
		FEDAD	File Data END ADDRESS		INTSET 割り込み要因設定エリア
		FLNGT	Length of Record (スケジュールデータ & メモーダタファイル)		KEYBU KEY データ レジスター
		FRDCT	Numbers of Record (スケジュールデータ & メモーダタファイル)		ALFLAG 1/60HZ JUMP TABLE
					INTFL 割り込み flag
					VRAM 仮想 VRAM
					DBUFF DISPLAY BUFFER (キャラクタ表示用)

図4-3 表示位置およびディスプレイバッファとの関係

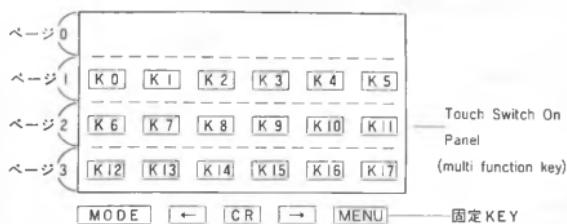
〈表示位置〉

0 page	DB01	02	03	04	05	06	07
1 page	DB11	12	13	14	15	16	17
2 page	DB21	22	23	24	25	26	27
3 page	DB31	32	33	34	35	36	37

〈表示位置と DISPLAY—BUFFER の関係〉

表示位置	DB ADDRESS	表示位置	DB ADDRESS
DB 01	BFE4H	DB 21	BFF2H
02	E5	22	F3
03	E6	23	F4
04	E7	24	F5
05	E8	25	F6
06	E9	26	F7
07	EA	27	F8
DB 11	EB	DB 31	F9
12	EC	32	FA
13	ED	33	FB
14	EE	34	FC
15	EF	35	FD
16	F0	36	FE
17	F1	37	FF

図4-4 ドットとVRAMの対応



注) Touch Switch On Panel を文章中 パネルスイッチ と称する。

図4-5 VRAMの構成

図4-6 キャラクタコード一覧表

上位4ビット→

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
T 値4ビット←																
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

未定義

未定義

NUMBER



ROM内サブルーチン の値い方

RC-20 の ROM の中にはユーザプログラムの作成に便利なサブルーチンが多く含まれています。これらは大別すると次の 5 種類になります。

- ①割り込み、およびシステム処理サブルーチン
- ②キー関係サブルーチン
- ③入出力ポート制御サブルーチン
- ④キャラクタ表示関係サブルーチン
- ⑤演算サブルーチン

以下、これらのサブルーチンについて順を追って説明します。

これらのサブルーチン内でのレジスタの保護には図5-1に示すように 7 種のパターンがあり、各サブルーチンごとに番号で示すことにします。この番号のついていないサブルーチンでは保護されるレジスタはないものとします。

図5-1 ROM内サブルーチンでのレジスタの保護

種別番号	A	B	C	D	E	H	L	IY	ZF	CF
1	○	○	○	○	○	○	○	○	○	○
2	○	○	○	○	○	○	○	○	×	×
3	×	○	○	○	○	○	○	○	×	×
4	×	○	○	×	×	○	○	○	×	×
5	×	×	×	○	○	○	○	○	×	×
6	×	×	×	○	○	×	×	○	○	×
7	×	×	×	×	×	×	×	○	○	×

5.1 割り込みおよびシステム処理サブルーチン

表5-1

ラベル	アドレス	機能	コメント	レジスタ 保 護
STBY	41C8	KEYが押されるまでZ-80は発振が停止して、スタンバイ状態に入る。KEYが押されると、Z-80の発振が再開し、KEYが押されてから2~3秒後、スタンバイ状態から脱出し、次のステップに制御を移す。	<ul style="list-style-type: none"> STBY命令後、約1msec以内にHALT命令を実行する。 STBY命令を行う前に、割り込みを不許可にする。 	I
INT	41A2	割り込みの設定を行う。 入力パラメータは、Aレジスタに設定する。 1H 1Hz割り込み 2H 10Hz割り込み 3H 1Hzおよび10Hz割り込み 4H 1/60Hz割り込み 0H 上記割り込み解除	3~7ビットのデータは無効とする。	I
Z80END	409A	マスターCPUがZ-80からサブCPUに移行する。Z-80プログラムを終了したとき、このエントリーへジャンプする。	<ul style="list-style-type: none"> この命令直後にHALT命令を実行する。 この命令を行う前に割り込みを不許可にする。 	I
LORE	40C6	レジスタをスタックより回復させる。 SAREと対で使用する。対象となるレジスタは、A, B, C, D, E, H, L, F, IX, IYレジスタである。		I
SARE	40B6	レジスタをスタックへ退避させる。 LOREと対で使用する。対象となるレジスタは、A, B, C, D, E, H, L, F, IX, IYレジスタ。		I
ERROR	428D	'ERROR'とエラーコードを表示する。 Aレジスタに1~9のエラーコードを入れて、本エントリにジャンプする(コール命令は不可)。本サブルーチンでは、'MODE'キーを押すと、計時機能にもどる。	エラーコード1~2はシステムエラー。 1:(①RAMソフトが入っていない、②RC-20の転送規約にあっていない、 2:ファイルの数が10個以上あるとき。	

5.2 キー関係サブルーチン

図5-2 KEYコード表

01	02	03	04	05	06
11	12	13	14	15	16
21	22	23	24	25	26
32	33	34	35	36	

①KEY 入力なし → 00H

②KEY デコード値(HEX コード)
→ 左図

図5-3 DEKEYによるKEYコード表の変換

01	02	03	04	05	06
07	08	09	0A	0B	0C
0D	0E	0F	10	11	12
14	15	16	17	18	

上記の KEY のデコード値を、左図の値に変換する。(HEX コード)

表5-2

ラベル	アドレス	機能	コメント	レジスタ 保 持
INKEY	43A3	KEY のチェックを行い、KEY の値(図5-2を参照)を A レジスタに返す。KEY が押されていないときは、0 を返す。スピーディな KEY 入力処理に適している。		3
PUTKEY	4391	INKEY と同様に、KEY のチェックを行い、値を A レジスタに返す。ただし、このルーチンでは、10Hz 割り込みに同期して KEY をチェックしている。通常の KEY 処理に適当。	割り込みフラグをリセットしているので、10Hz 割り込み処理をしている時は、本ルーチンの前に処理しておくこと。	3
GETKEY	4090	PUTKEY と似ているが次の点が異なる。 ①KEY 入力のないときは、ZERO フラグを ON にする。 ②モードスイッチが押された時は、Z-80END ヘジャンプする。 (それ以外は PUTKEY と同じ)		

ラベル	アドレス	機能	コメント	レジスタ保種																		
KEYIN	4352	STBYのCALL時に、どのKEYが押されたかをチェックするのに用いる。 AレジスタにKEYデータを返す。STBYに続いてCALLする。	STBY状態を解除した時のKEYの値を入力する。																			
DEKEY	40A1	KEYの値の変換を行なう。図5-2の値を図5-3の値に変換する。PUTKEY等の後でCALLする。		3																		
KDISP 1	413B	~ 3ページに次のKEY表示を行う。 <table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr><tr><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td></tr><tr><td>M</td><td>N</td><td>O</td><td>U</td><td>[P]</td><td>[I]</td></tr></table>	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	U	[P]	[I]	~ 3ページの消去は不要。	7
A	B	C	D	E	F																	
G	H	I	J	K	L																	
M	N	O	U	[P]	[I]																	
KDISP 2	4152	~ 3ページに次のKEY表示を行う。 <table border="1"><tr><td>O</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td></tr><tr><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td></tr><tr><td>A</td><td>E</td><td>I</td><td>[]</td><td>[I]</td><td>[A]</td></tr></table>	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	E	I	[]	[I]	[A]	~ 3ページの消去は不要。	7
O	P	Q	R	S	T																	
U	V	W	X	Y	Z																	
A	E	I	[]	[I]	[A]																	
KDISP 3	4169	~ 3ページに次のKEY表示を行う。 <table border="1"><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td><td>9</td><td>:</td><td>*</td></tr><tr><td>/</td><td>-</td><td>.</td><td>*</td><td>[A]</td><td>[P]</td></tr></table>	0	1	2	3	4	5	6	7	8	9	:	*	/	-	.	*	[A]	[P]	~ 3ページの消去は不要。	7
0	1	2	3	4	5																	
6	7	8	9	:	*																	
/	-	.	*	[A]	[P]																	
KALDIS	4112	~ 3ページに、HLレジスタで指定された先頭アドレスから、ASCIIコードの列を、3段×6文字で、KEY表示する。	~ 3ページの消去は不要	7																		
KHYP23	4115	2~3ページに、2段×6文字の型式で、HLレジスタで先頭アドレスを指定されたASCII文字列をKEY表示する。	2~3ページの消去は不要	7																		
KHYP 1	41F3	1ページにKEY表示を行う。表示するASCII文字列の先頭アドレスをHLレジスタで指定する。	1ページの消去は不要	7																		
KHYP 2	41F7	2ページに、KHYP1と同様なKEY表示を行う。	2ページの消去は不要	7																		
KHYP 3	41FB	3ページに、KHYP1と同様なKEY表示を行う。	3ページの消去は不要	7																		

5.3 入出力ポート制御サブルーチン

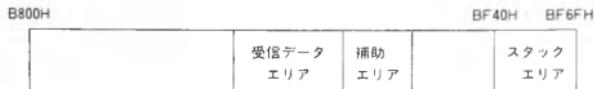
表5-3

ラベル	アドレス	機能	コメント	レジスタ 保 證
TIMER	5FE8	Bレジスタの引数に従ってタイマを作り(0.1~1.6秒)。タイマの値=0.1×(Bレジスタ)秒、引数が0のときは1.6秒Bレジスタの上位4ビットは無視される。	タイマの誤差は0.1秒以内。	1
RSOUT	4320	RS-232Cにより、パラメータで指定したメモリ上のデータを出力する。RS-232Cからデータが送出されている時、2秒おきに鳴鐘する。RS-232C出力を強制的に止めるには、「MODE」KEYを1.5秒以上押す。このとき制御は、計時モードに移る。 (パラメータ) ・HL：スタート・アドレス ・DE：エンド・アドレス ・A：ポーレート 8H:2400ポー 4H:1200ポー 2H:600ポー ^ー 1H:300ポー ^ー	・スタートアドレスとエンドアドレスがB800H～BF3FHの間に入っていないと、何もせずにリターンする。 ・パラメータで指定したバイト数に統けて、最大22バイトの余分な情報(ゴミ)を送出する。 ・HLで指定したアドレスの次のアドレスよりデータを出力することがある。	7
RSIN	4310	RS-232Cから、データが、パラメータで指定したRAM上に格納される。RS-232Cよりデータを入力している時、2秒おきに鳴鐘する。RS-232Cからの入力は次の条件の時終了する。 ①データがB800H～BF3FHの範囲をこえたとき。 ②送信データが、0.6秒以上の間こないとき。 ③「MODE」スイッチを1.6秒以上押したとき(計時モードになる)。		6

ラベル	アドレス	機能	コメント	レジスタ 保 護
RSIN	4310	(バラメータ) ・ HLレジスタ：格納スタート・アドレス ・ Aレジスタ：ポーレート 8H : 2400ポー 4H : 1200ポー 2H : 600ポー ¹ H : 300ポー	(註)	6
RSVER 0	42E0H	HLレジスタで指定されるアドレスの値からDEで指定されるアドレスまでの総和を計算し合計が0となれば, VERIFY OKとする。結果はAレジスタに返される。 ・ Aレジスタ (0IH:エラーなし FFH:エラー)		3
RSVER 1	42E5	転送規約(2.1)にもとづいて, RS-232Cのインプットデータをチェックする。結果はAレジスタに返される。 ・ Aレジスタ (00H:VERIFYデータなし 01H:エラーなし FFH:エラー)		3
MUSIC	42AEH	(1)音階と音長のデータより音を発生する。 (2)音発生中は, Z-80は, WAIT状態になっている。 (3)音長コード $n = 00H \quad 1.6\text{秒}$ $n = 1H - FH \quad (0.1 \times n)\text{秒}$ (4)音階データを表5-4に示す。 (バラメータ) ・ Aレジスタ : 音階コード00H~19H ・ Bレジスタ : 音長コード00H~0FH		2

(註) i) 受信側は、送信側の設定したスタートアドレスからエンドアドレスまでのデータ長と同等の受信データエリアを用意し、その直後に22バイト以上の補助エリアを設ける必要がある。

ii) 受信データエリアがユーザーエリアの最終アドレスBF3FHを越える場合は、スタックエリアの22バイトが補助エリアとなるためPUSH, CALL命令のネスティングはCALL RS INを含めて、5以内（サブルーチンRSINの内部にCALL命令有り）とする。



ユーザーエリア

表5-4 音階データ

音階コード	音 階	周 波 数	音階コード	音 階	周 波 数
0H	C 5	532Hz	CH	C 6	1040Hz
I	C 5♯	555	D	C 6♯	1110
2	D 5	585	E	D 6	1170
3	D 5♯	624	F	D 6♯	1237
4	E 5	655	I0	E 6	1311
5	F 5	697	I1	F 6	1395
6	F 5♯	745	I2	F 6♯	1490
7	G 5	780	I3	G 6	1560
8	G 5♯	830	I4	G 6♯	1667
9	A 5	886	I5	A 6	1771
A	A 5♯	936	I6	A 6♯	1864
B	B 5	993	I7	B 6	1975
19H	休 止		I8	C 7	2093

5.4 キャラクタ表示関係サブルーチン

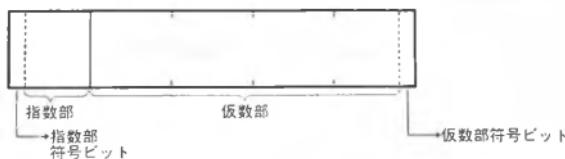
表5-5

ラベル	アドレス	機 能	コ メ ン ト	レジス タ 保 護
HYOGI	4106	ディスプレイ・バッファ(BFE4H～BFFFH)の値に従って、0～3ページにキャラクタを表示する。前もって、ディスプレイバッファ内に表示すべきASCIIコードを入れておく。	ディスプレイ・バッファと表示位置との関係は、図4-3を参照。	7
HYO123	4109	HYOGIと同様な機能を1～3ページに対して行う。	//	7
HYO23	410C	HYOGIと同様な機能を2～3ページに対して行う。	//	7
HYO0	411A	HYOGIと同様な機能を0ページに対して行う。	//	7
HYP1	4221	HYOGIと同様な機能を1ページに対して行う。	//	7

ラベル	アドレス	機能	コメント	レジスタ 保 護
HYP 2	4228	HYOGIと同様な機能を2ページに対し て行う。	//	7
HYP 3	422F	HYOGIと同様な機能を3ページに対し て行う。	//	7
WHIAL	426A	0~3ページの全画面をすべて消去する。 (VRAMに00Hをつめる。)		1
WHI01	4257	0~1ページの画面をすべて消去する。		1
WHI23	4281	2~3ページの画面をすべて消去する。		1
WHI0	4247	0ページの画面をすべて消去する。		1
WHI1	425A	1ページの画面をすべて消去する。		1
WHI2	4263	2ページの画面をすべて消去する。		1
WHI3	4284	3ページの画面をすべて消去する。		1
ASCII	40D0	Aレジスタで与えられた1ケタの0~ 9, A~Fの16進コードをASCIIコー ドに変換する。結果は、Aレジスタに 返される。		3
ACIBC	4081	Aレジスタに与えられた2ケタの16進 コードを2個のASCIIコードに変換する。 上位コードはBレジスタ、下位コードはCレジスタに返される。		5
BUFCL	43C1	キャラクタバッファ(CHRBUF)内に、 すべて20Hを書き込む。		1
DISPCL	43B6	VRAM内に、すべて00Hを書き込む。		2
CHARA	4041	Aレジスタに文字コード、DEレジス タにVRAMのアドレスを指定すると、 文字データをVRAM上に書き込む。DE レジスタは、+5値が増加する。	Aレジスタに文字 コード以外の値が 入ると、空白(20 H)を表示する。	4
DISP	411C	VRAMデータをドット・ドライバに書 き込む。 (パラメータ) Aレジスタ Eレジスタ 表示		7
		1H 40H 0ページ		
		1H 00H 1ページ		
		2H 00H 2ページ		
		2H 40H 3ページ		
		3H 40H 0,3ページ		
		3H 00H 1,2ページ		

5.5 演算データの型式と演算サブルーチン

図5-4 データの型式(FPACCの構成)
浮動小数点型式をとり、次の表現を持つ。



数値範囲 : $\pm 0.999999999 \times 10^{100}$

図5-5 演算サブルーチンのワークエリア

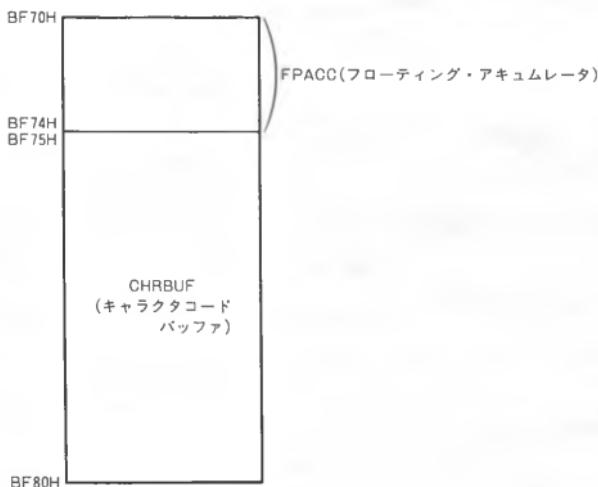


表5-6 演算サブルーチン

ラベル	アドレス	機能	コメント	レジスタ 保譲
CONVT 1	5A31	<p>キャラクタコードバッファ(CHRBUF)に格納されている数値の文字列(ASCII型式)を浮動小数点数(HEX型式)に変換し、正規化して、フローティングアキュムレータ(FPACC)に結果を入れる。CHRBUF内のデータの終りには、00Hをセットすること。また、CHRBUF内には、符号、コロン(.)、「E」も含む。変換のエラーの有無は、Aレジスタに返される。</p> <ul style="list-style-type: none"> A : 0 : 正常終了 1 : エラー(文字列に異常) 2 : エラー(数値範囲をこえている。) 	HEX 変換により小数点以下10ケタ以下については四捨五入により、9ケタにまるめられる。	
CONVT 2	5BC7	<p>FPACCに格納されている数値(HEX型式)を文字列(ASCII型式)に変換して、キャラクタコードバッファ(CHRBUF)に結果を格納してもどる。なお、FPACCには、正規化された形でデータを格納しておくこと。キャラクタコードが12ケタ未満の時は、キャラクタコードの最後に、00Hをつけてデータの終りを示す。</p> <p>変換時のエラーは、Aレジスタに返される。</p> <ul style="list-style-type: none"> ・ Aレジスタ 0 : 正常終了 1 : エラー(数値に異常あり) 	数値の指数部が10以上、または、-3未満の時は、指数表示に変換する。	
ADDDNUM	589F	<p>レジスタABCDEに格納したデータとFPACCに格納したデータの和を計算し、FPACCに結果を格納する。エラーの有無は、Aレジスタに格納される。</p> <ul style="list-style-type: none"> ・ Aレジスタ 0 : 正常終了 1 : エラー(オーバーフロー) 	演算結果が $0.999999999 \times 10^{100}$ をこえるときは、オーバーフローとしてエラー扱いとなる。	
SUBNUM	58A8	レジスタABCDEに格納されたデータから、FPACCに格納されているデータを引き、結果をFPACCに格納する。	同上	

ラベル	アドレス	機能	コメント	レジスタ保護
SUBNUM	58 A8	· A レジスタ { 0 : 正常終了 : エラー(オーバーフロー)	同上	
MULTPL	572 B	レジスタ ABCDE に格納されたデータに、 FPACC に格納されたデータを乗じ、結果を FPACC に格納する。エラーの有無は、 A レジスタに返される。 · A レジスタ { 0 : 正常終了 : エラー(オーバーフロー)	同上 また、 0.1×10^{-99} より小さい時は、 0 とする。	
DIVIDE	57 C6	レジスタ ABCDE に格納されているデータを、 FPACC に格納されたデータで除算し、結果を FPACC に格納する。エラーの有無は、 A レジスタに返される。 · A レジスタ { 0 : 正常終了 : エラー(オーバーフロー) 2 : エラー(除数が 0)	同上 また、除数が 0 の場合は、エラーとなりレジスタ A = 2 とし計算結果は保証されない。	
NEGSIN	5720	FPACC に格納されている数値の仮数部の符号を反転する。 A レジスタには、 0 が入る。		
INVERT	5F 66	FPACC に格納されたデータの逆数を求め、結果を FPACC に格納する。エラーの有無は、 A レジスタに返される。 · A レジスタ { 0 : 正常終了 : エラー(オーバーフロー) 2 : エラー(データが 0)	DIVIDE の場合と同じ。	
RANDOM	5F 70	0 以上 1 未満の乱数(ケタ数は小数以下 9 ケタで 0 ~ 0.999999999)を 1 個だけ発生し、 FPACC に格納する。 A レジスタには、 0 が入る。		

NUMBER

6

表示機能

リストコンピュータ RC-20 では、キャラクタ表示とグラフィック表示が可能であり、また、両者を併用することもできます。キャラクタ表示は 7 文字 × 4 行、グラフィック表示は 32 × 42 ドットとなります。

キャラクタ表示

7 文字 4 行の任意の場所にキャラクタを表示することができます。このためには、ディスプレイバッファ (BFE4H ~ BFFFH) にキャラクタコードを格納しておき、キャラクタ表示関係サブルーチンを CALL します。キャラクタ表示関係サブルーチンについては表 5-5 を参照してください。

7 文字 4 行以外の任意の場所へキャラクタ表示する場合は、グラフィック表示を使用します。

グラフィック表示

一般的なグラフィック表示の手順について説明します。

① VRAM 上に 1 ページ分グラフィックデータを書き込みます。すなわち表示すべき 1 ページ分の VRAM 全部にデータを書き込みます。ドットと VRAM のアドレスの対応は図 4-5 を参照してください。

② 液晶上の表示はページ単位ですので、どのページに表示するかを指定するグラフィックパラメータを A レジスタと E レジスタにセットします。パラメータの一覧表を表 6-1 に示します。

③ サブルーチン DISP(411CH) を CALL します。

この応用例として、先のリスト 1 に示した「*GRAPHIC*」がありますので参照してください。

次にグラフィック表示で、任意の場所へ文字を表示する方法について説明します。

① 1 ページ分の VRAM の任意のエリアにキャラクタコードをセットします。

すなわち

- 1) A レジスタにキャラクタコードを設定し。
- 2) DE レジスタに VRAM のアドレスを設定し (1 文字は 5 バイトで表示される。それ故(DE) + 5 ≤ BFE
3H となること)。
- 3) CHARA(4041H) を CALL します。DE の値は +5 された値になります。
②VRAM 内の上記のキャラクタコード以外の部分のデータを書き込みます。
- ③グラフィックパラメータをセットします。
- ④サブルーチン DISP(411CH) を CALL します。

表6-1 グラフィックパラメータ

ページ	A レジスタ	E レジスタ
0	1 H	40H
1	1 H	00H
2	2 H	00H
3	2 H	40H
0, 3	3 H	40H
1, 2	3 H	00H

NUMBER

7 入出力ポート機能

RC-20には、ユーザ・プログラムを直接アクセスできる入出力ポートがいくつかあります。これらの入出力ポートの一覧表を表7-1に示します。I/OポートへのIN、OUTは、逆にしないように注意してください。

以下、各入出力ポートについて詳しく説明します。

表7-1 入出力ポート一覧表

Port	Function	Name
10H	1/10 1/100sec DATA INPUT	STW
30H	計時 DATA INPUT	sec WA 1
31H		min WA 2
32H		hour WA 3
33H		A/P & Week WA 4
34H		day WA 5
35H		month WA 6
36H		year WA 7
40H	KEY DATA INPUT	KEYIN
50H	MUSIC DATA INPUT	MUSIC
60H	INKEY(SCAN & DATA)	INKEY
E0H	Z80 MODE END	END

カウンタ・データ入力

IN A,(10H)

この命令で、データバスを経て、1/100秒、1/10秒のカウンタの値がAレジスタにロードされます。データは、データバスのD0～D3が1/100秒カウンタの値を、D4～D7が1/10秒のカウンタの値を表します。これらのカウンタには、リセット機能はありません。

時計データ入力

IN A,(WAn)

[n = 1 ~ 7]

この命令で、時計データをAレジスタにロードします。ポートは30H～36Hまであり、それぞれの場合のデータは、表7-2に示すようになります。

表7-2

ポート	D0～D3	D4～D7
WA 1 (30H)	1 sec	10 sec
WA 2 (31H)	1 min	10 min
WA 3 (32H)	1 hour	10 hour
WA 4 (33H)	A/P	Week
WA 5 (34H)	1 day	10 day
WA 6 (35H)	1 month	10 month
WA 7 (36H)	1 year	10 year

表7-3

	D7	D6	D5	D4
SUN	0	0	0	0
MON	0	0	0	1
TUS	0	0	1	0
WED	0	0	1	1
THU	0	1	0	0
FRI	0	1	0	1
SAT	0	1	1	0

A/Pは、AM/PMの表示であり、AMの場合は0000、PMの場合は1000のビットパターンが、D3、D2、D1、D0に入ります。

また、Weekの場合は、表7-3に示すように、日曜(SUN)の0000～土曜(SAT)の0110というビットパターンが、D7、D6、D5、D4に入ります。

キーデータの入力(KEYIN/INKEY)

IN A,(40H)KEYIN

この命令は、最新のキーデータを入力します。

IN A,(60H)INKEY

この命令は、キースキャンを行い、キーが押されているかどうかをチェックします。

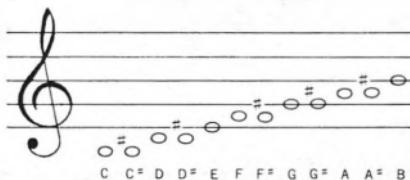
なお、キーの値と優先順位を、次ページ図7-1に示します。

図7-1

	C1	C0	C0	C0	C0	C1
R3	01	02	03	04	05	06
R2	11	12	13	14	15	16
R1	21	22	23	24	25	26
R0	32		33	34	35	36

$R0 > R3 > R2 > R1$
 $C0 > C1$

図7-2 音階コードと音階対応表



音名	平均音高 基準(Hz)	分周近似 音高(Hz)	音階コード	音階	平均 音高 (Hz)	分周近似 音高(Hz)	音階コード
C5	523.25	524.3	82	C6	1046.5	1040.3	C0
C5#	555.4	555.4	89	C6#	1108.7	1110.8	C4
D5	587.3	585.2	8F	D6	1174.6	1170.3	C7
D5#	622.3	624.2	96	D6#	1244.5	1236.5	CA
E5	659.3	655.4	9B	E6	1312.5	1310.7	CD
F5	698.5	697.2	A1	F6	1395.9	1394.4	D0
F5#	740.0	744.7	A7	F6#	1480.0	1489.5	D3
G5	784.0	780.2	AB	G6	1568.0	1560.4	D5
G5#	830.6	829.6	B0	G6#	1661.2	1657.1	D8
A5	880.0	885.6	B5	A6	1760.0	1771.2	DA
A5#	932.3	936.2	B9	A6#	1864.6	1872.5	DC
B5	987.8	993.0	BD	B6	1975.6	1985.9	DE
00	休止	00	C7	2093.0	2114.1	E0	

音楽(MUSIC)ポート

このポートへのアクセスは、次のように行います。

```
LD A,n1 ←————音階コード  
LD B,n2 ←————音長  
LD C,50H ←————MUSIC  
OUT (C),A
```

Aレジスタには、音階コードを入れます。音階コードと音階の対応表を図7-2に示します。音階コードは、80H~FFHまでセットできます。

Bレジスタには、音長(00H~0FH)を入れます。音長は、次のようにになります。

00H	1.6秒
01~0FH	($0.1 \times n$)秒

Z80モードEND

OUT (EOH),A

この命令により、Z80モードからタイム・スタンダードCPU(サブCPU)に制御が移ります。

この命令の直後に、HALT命令を入れてください(入れないと暴走します)。また、この命令を行う前に、Z80のINT割り込みを無効にしてください。

D1 ←————割り込み禁止
OUT (EOH),A
HALT

なお、この命令でZ80を終了した場合は、再びZ80を起動させると、HALT命令以降の命令から実行を始めます。

通常は、サブルーチン Z80END(409AH)を使用してください。

NUMBER

8

割込み機能

8.1 割込み要因

RC-20における割込み要因を、表8-1に示します。

表8-1

割 込 み	IHZ	計時機能の1HZに同期して、割込みをかける。
	10HZ	計時機能の10HZに同期して、割込みをかける。
	IHZ & 10HZ	計時機能の1HZと10HZに同期して、割込みをかける。
	1/60HZ割込み	時刻表示モードにおいて、「分ごとに割込みをかける」、INTBF (BFB7H～BFB8H) に格納されているアドレスにジャンプし、プログラムを実行する。
	Z-80 STAND BY & KEY RESTART	Z-80がHALT状態になり、KEYが押されると、再びHALTの次のステップから実行する。

8.2 割込みの方法(10HZ, 1HZ割り込み)

システム・サブルーチンを用いて割込み処理を行う場合

システム・サブルーチンを用いて割込み処理を行う場合は、次のような手順で行います。

- ① INTERRUPT CODE SET(INT:41A2H) サブルーチンによって、割込み要因をセットします。
- ② 通常、割込み禁止にしてあるので、EI命令で割込み許可状態にします。割込み処理中は、割込み処理ルーチンで割込み許可を行なっているので、その都度 EI命令を行う必要はありません。

- ③ 割込みが発生すると、INTFL (BFB9H) の割込み要因に対応するビットがセットされます。ユーザの割込み処理は、INTFLのビットをチェックして、割込みがあれば、その処理を行います。**図 8-1** は、INTFLの内容を示したものです。
- ④ 割込みを終了させたい時は、INTサブルーチンで割込みを解除して、DI命令を実行します。**図 8-2** は、その一例を示したものです。

図 8-1 INTFL の内容

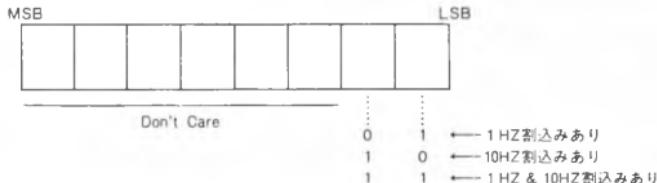


図 8-2 割込み処理の例

```

INT EQU 41A2H
INTFL EQU 0BFB9H

LD A, 3H   ← 10HZ, 1HZ割込みパラメータ
CALL INT   ← 割込み要因設定
EI         ← 割込み許可
HALT
LD A, (INTFL) ← (INTFL)のチェック
BIT 0, A
JR Z, INT10 ← 10HZ割込み処理
BIT 1, A
JR Z, INT11 ← 1HZ割込み処理
JR INT10   ← 1HZ & 10HZ割込み処理

INT10 :

JR INTEND

INTEND : SUB A ← (INTFL)のリセット
LD (INTFL), A

LD A, 0
CALL INT } 割込み要因の解除
DI   ← 割込みの禁止

```

なお、ユーザプログラムにおいて、PUTKEY, GETKEYと割込みを併用して用いる場合、次の注意が必要です。

- ① PUTKEY, GETKEYは、10HZ割込みを使用しています。
- ② サブルーチンをコールすると、内部でINTFLをリセットしています。
- ③ そのため、サブルーチンを呼ぶ前にINTFLがセレクトされている場合、その処理をしておく必要があります。ただし、KEY処理後のINTFLの内容は、保護されています。

ユーザ割込み処理を行う場合

システム・サブルーチンでZ80の割込みモード2を使用しているので、ユーザもモード2を使用し、IM0, IM1命令を使用することはできません。

Iレジスタの値を変えることによって、ユーザは割込みテーブルを設定することができます。この場合、有効でない番地を設定すると、Z80は暴走する可能性があるので、RAM上の有効な番地を設定してください。割込み要因に基づくシステムのベクトル値は、以下の通りです。

1HZ割込み - 00H
10HZ割込み - 02H
1HZ&10HZ割込み - 04H

図8-3 ユーザ割込み処理の例

```
INT EQU 41A2H
.
.
.
LD A, 0B9H
LD I, A
.
.
.
LD A, 3H
CALL INT
EI
HALT
.
.
.
B900H: DW INT1
B902H: DW INT10
B904H: DW INT110 } 割込みテーブル
.
.
.
INT1:
INT10:
.
.
.
INT110:
```

ユーザ割込みを使用する場合、割込みがかかり、割込みが終了すると、Z80は割込み禁止状態になっているので、EI命令で割込み許可状態にする必要があります。割込み終了は、システム・サブルーチンと同様です。

図8-3は、その使用例を示したものです。

8.3 特殊割込みの方法

1/60HZ割込み

時刻表示モードで、1分ごとにZ80に起動をかけます。ただし、スケジューラ・アラームがセットされ、アラーム許可の場合は、1/60HZ割込みは無視されます。

1/60HZ割込みを許可する時は、INTBF (BFB7H~BFB8H) に、1/60HZ割込み処理用プログラムへのジャンプ・アドレスを入れておく必要があります。割込み許可(EI命令)の必要はありません。もし、1/60HZ割込み処理用プログラムが入っていないと、基本時計モードの分析上りで、“ERROR”表示が出ます。これを直すには、新たにRAM上で正規なデータを入力する必要があります。1HZと10HZの割込みとの併用も可能です。その時の割込みパラメータは07Hです。

図8-4は、使用例を示したものです。

Z80 STAND-BY & KEY RESTART割込み

この割込みは、Z80をSLEEP状態にして、KEYが押されると次のアドレスから実行します。この割込みを許可すると、10HZと1HZの割込みはZ80がSLEEP状態の間無効となり、KEYが押されて再度Z80が実行を始めると、有効となります。

どのKEYが押されてリスタートしたかは、KEYINサブルーチン(4352H)を用いるとチェックすることができます。KEYが押されてからZ80がリスタートするまで2~3秒かかります。図8-5は、使用例を示したものです。

図8-4
1/60HZ割込み
の使用例

```
INT EQU 41A2H
INTBF EQU 0BFB7H
...
LD A,4H ← 1/60HZ割込み設定
CALL INT
LD A,00H
LD (INTBF),A
LD A,0BAH
LD (INTBF+1),A
...

```

1/60HZ割込み時に、BA00Hにジャンプするように設定(BA00Hは割込み処理プログラムの先頭アドレス)。

図8-5 Z80-STAND-BY & KEY RESTART割込みの使用例

```
INT    EQU 41A2H
STBY  EQU 41C8H
KEYIN EQU 4352H
:
LD A,3H
CALL INT
:
CALL STBY ←この期間中10HZ, 1HZ割込みは不可
CALL KEYIN
{
    KEY処理
}
```

注) 割込み請求中(割込み要因をセットし、割込み許可中)には、RS232CおよびMUSIC関係のI/Oポート(C0H, 50H)を使用する命令の直後において、I/O命令を行わないようしてください。

NUMBER

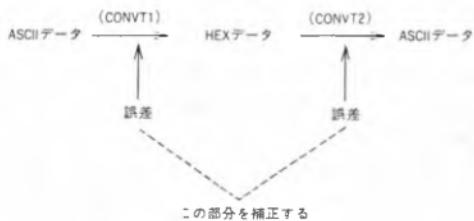
9

浮動小数点演算 における注意

9.1 データ補正サブルーチン(CORRECT)

演算サブルーチンで演算を行う場合、一般にキャラクタバッファ内の数値の文字列をCONVT 1(数値変換)によって、浮動小数点数(HEX型式)に変換した後、演算を行い、さらにその結果をCONVT 2(キャラクタ変換)によって、再びキャラクタバッファ内の数値の文字列に変換します。この際、図9-1に示すように、2カ所で誤差が生じます。

図9-1



その結果、これらの誤差を補正する必要が生じます。

たとえば3.99999は4に、99.99999は100に、9.99999E-99は1E-98に補正する必要があります。

このデータ補正を行うサブルーチン(CORRECT)は、キャラクタコードバッファに格納されているデータを、先頭から9桁目でまるめ込み、結果をキャラクタコードバッファに格納します。このCORRECTサブルーチンは、RC-20のROMには含まれていないので、ユーザが各自で作成して使用してください。

図9-2にCORRECTサブルーチンのフローチャートを、リスト28にアセンブリしたソースリスト

をそれぞれ示します。

なお、パラメータは次のように略記してあります。

CHR PTR : キャラクタバッファのポインタ

CHR BUF : キャラクタバッファの先頭アドレス

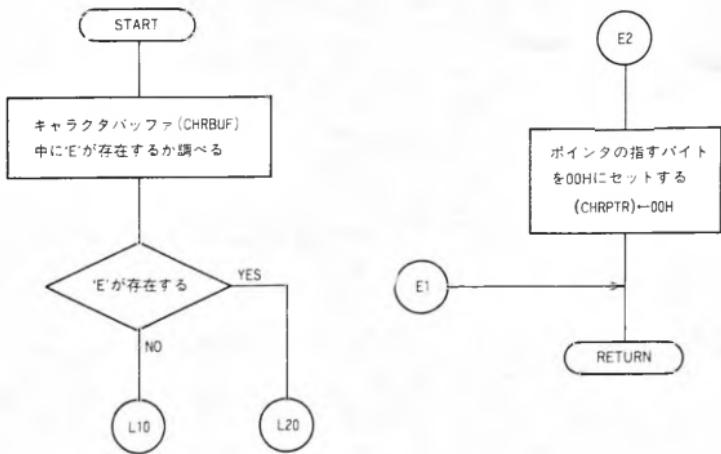
CHR BUFE : キャラクタバッファの最終アドレス

M : 補正の精度（有効桁数）

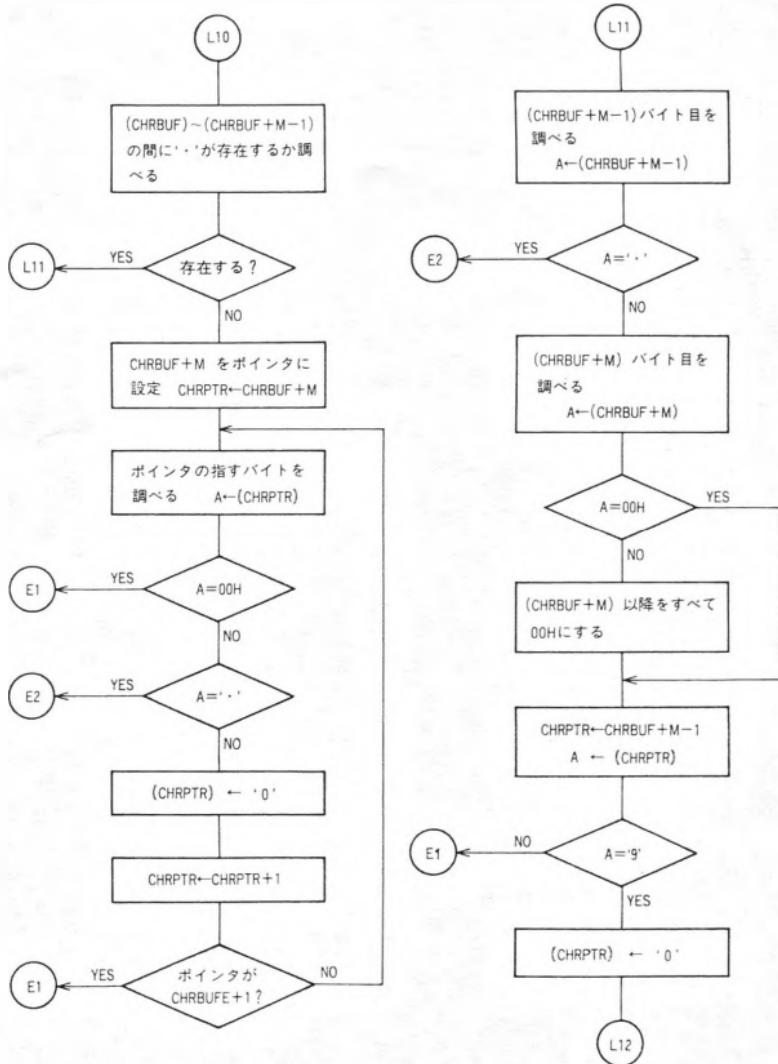
[注] A : レジスタ A とは限らず、ある作業領域
に付けた名前です。

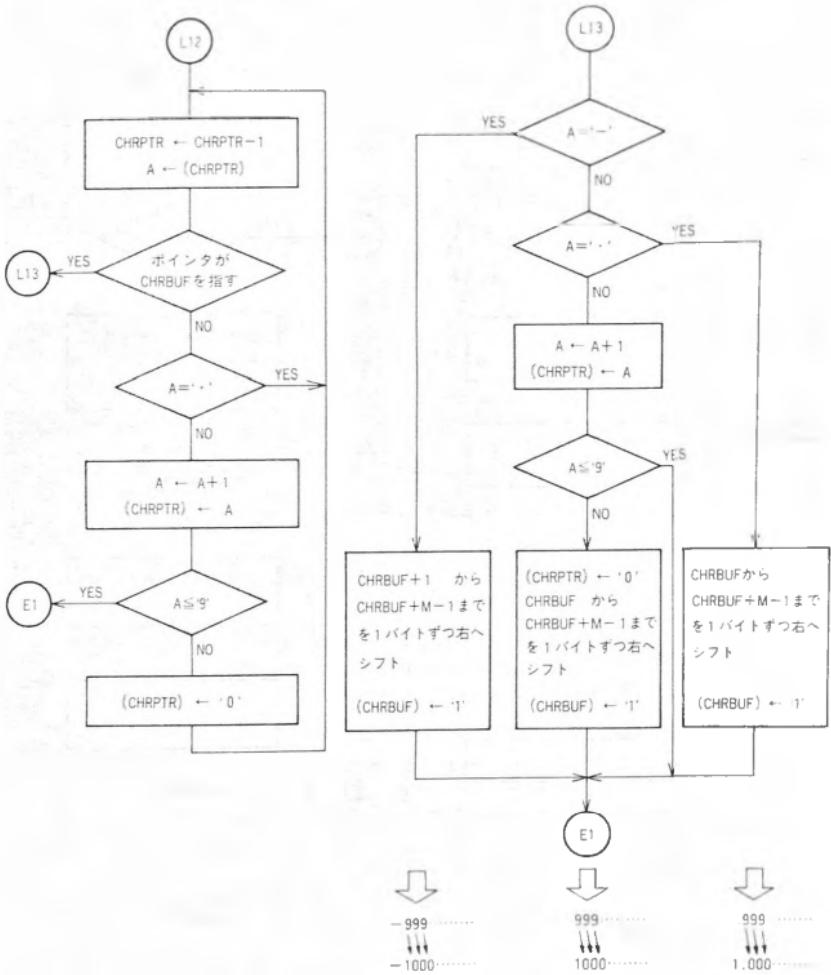
図9-2 CORRECTサブルーチンのフローチャート

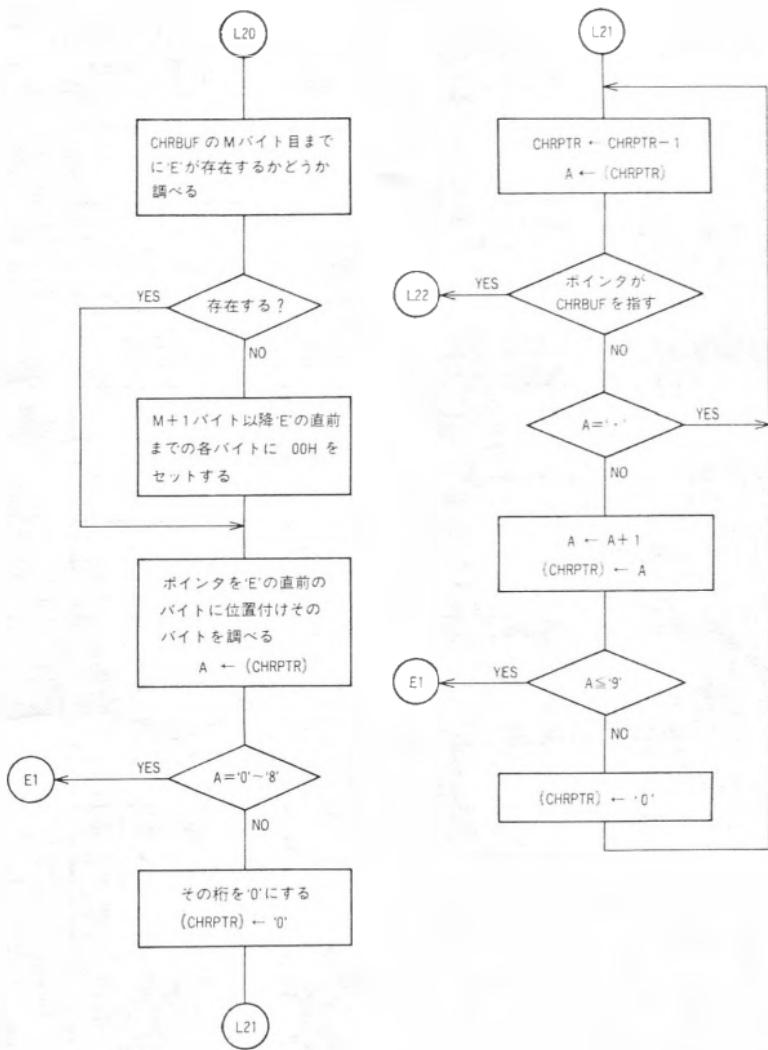
CORRECT :

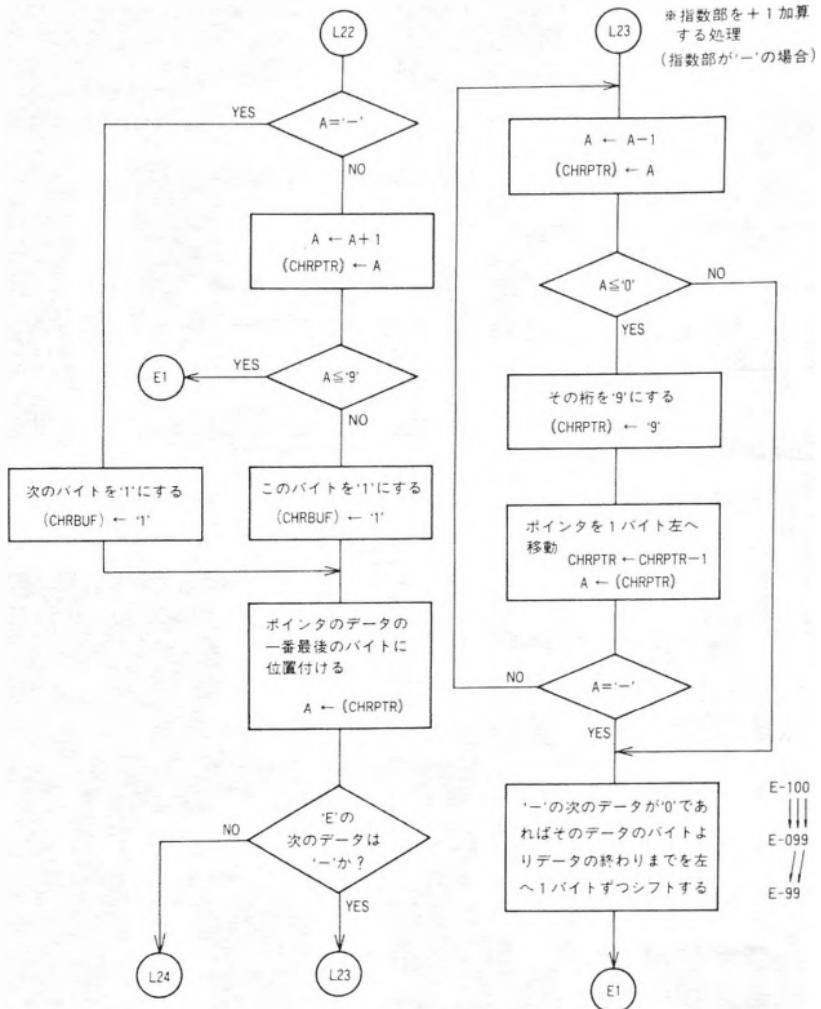


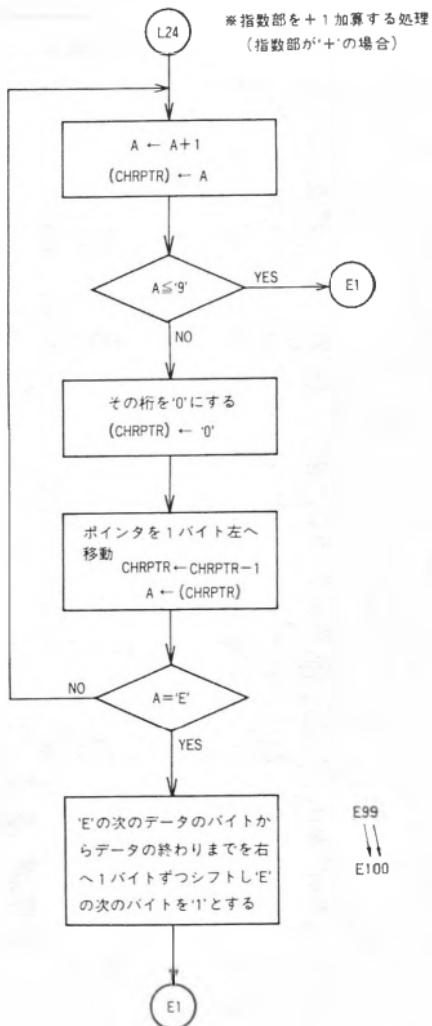
※データ中に'E'を含まない場合の処理











リスト28 CORRECTサブルーチンのソースリスト

MACRO-B0 J.36 17-Mar-80		PAGE	1
; *CORRECT*			
; CORRECTION OF NUMBER			
; .780			
; ENTRY CORRECT			
BF75	CHRBUF EQU	0BF75H ;CHARACTER DATA BUFFER	
0008	M EQU	B	
003A	OUTNUM EQU	3AH	
;			
0000'	CORRECT:		
0000'	; ** CHECK CHARACTER 'E' **		
21 BF75	LD	HL,CHRBUF	
0003' E5	PUSH	HL	
0004' 7E	LD	A,(HL)	
0005' B7	OR	A	
0006' 28 0B	JR	Z,NOEXIST	
0008' FE 45	CP	'E'	
000A' CA 0098'	JP	Z,EXIST	
000D' 23	INC	HL	
000E' 1B F4	JR	LOOPCQ	
;	; CASE OF DATA NOT INCLUDE 'E' :		
;	; ** CHECK CHARACTER ',' **		
0010'	NOEXIST:		
0010' E1	POP	HL	
0011' 06 08	LD	B,M	
0013' 7E	LD	A,(HL)	
0014' FE 2E	CP	'.'	
0016' 28 21	JR	Z,NEITQQ	
0018' 23	INC	HL	
0019' 10 FB	DJNZ	LOOPCQ	
;	CORCT1:		
001B' 7E	LD	A,(HL)	
001C' B7	OR	A	
001D' CA 014C'	JP	Z,ENDCRCT	
0020' FE 2E	CP	'.'	
0022' 2B 10	JR	Z,CORCT2	
0024' 36 30	LD	(HL),'0'	
0026' 23	INC	HL	
0027' E5	PUSH	HL	
0028' 11 BF81	LD	DE,CHRBUF+12	
002B' AF	IOR	A	
002C' ED 52	SBC	HL,DE	
002E' E1	POP	HL	
002F' CA 014C'	JP	Z,ENDCRCT	
0032' 1B E7	JR	CORCT1	
0034'	CORCT2:		

0034'	36 00	LD	(HL),0
0036'	C3 014C'	JP	ENDCRT
;			
0039'		NEXT00:	
0039'	21 BF7C	LD	HL,CHRBUF+M-1
003C'	7E	LD	A,(HL)
003D'	FE 2E	CP	"."
003F'	CA 0034'	JP	Z,CRCT2
0042'	E5	PUSH	HL
0043'		NXT001:	
0043'	23	INC	HL
0044'	7E	LD	A,(HL)
0045'	87	OR	A
0046'	28 04	JR	Z,NXT000
0048'	36 00	LD	(HL),0
004A'	1B F7	JR	NXT001
004C'		NXT000:	
004C'	E1	POP	HL
004D'	7E	LD	A,(HL)
004E'	FE 39	CP	'9'
0050'	C2 014C'	JP	NZ,ENDCRT
0053'	36 30	LD	(HL),'0'
0055'		LOOPNE1:	
0055'	28	DEC	HL
0056'	E5	PUSH	HL
0057'	11 BF75	LD	DE,CHRBUF
0058'	AF	XOR	A
0058'	E0 52	SBC	HL,DE
005D'	E1	POP	HL
005E'	28 10	JR	Z,NEXT01
0060'	7E	LD	A,(HL)
0061'	FE 2E	CP	"."
0063'	28 F0	JR	Z,LOOPNE1
0065'	3C	INC	A
0066'	77	LD	(HL),A
0067'	FE 3A	CP	OUTNUM
0069'	C2 014C'	JP	NZ,ENDCRT
006C'	36 30	LD	(HL),'0'
006E'	1B E5	JR	LOOPNE1
;			
0070'		NEXT01:	
0070'	7E	LD	A,(HL)
0071'	FE 20	CP	"-x"
0073'	28 1E	JR	Z,MNSHT
0075'	FE 2E	CP	"."
0077'	28 09	JR	Z,PROSH
0079'	3C	INC	A
007A'	77	LD	(HL),A
007B'	FE 3A	CP	OUTNUM
007D'	C2 014C'	JP	NZ,ENDCRT
0080'	36 30	LD	(HL),'0'
0082'		PROSH:	
0082'	01 0008	LD	BC,M
0085'		MNSHT1:	

0000'	CA 014C'		JP	Z,ENDCRCT
0003'	36 30		LD	(HL), '0'
0005'	18 E5		JR	LOPCK2
0007'				NXTCK1:
0007'	7E		LD	A,(HL)
0008'	FE 2D		CP	'-'
000A'	2B 0B		JR	Z,RESET1
000C'	3C		INC	A
000D'	77		LD	(HL), A
000E'	FE 3A		CP	OUTNUM
000E'	C2 014C'		JP	nz,ENDCRCT
0003'	36 31		LD	(HL), '1'
0005'	18 03		JR	NITCK2
0007'				RESET1:
0007'	23		INC	HL
0008'	36 31		LD	(HL), '1'
000A'				NITCK2:
000A'	21 8FB0		LD	HL,CHRBUF+11
000D'				LOPCK3:
000D'	7E		LD	A,(HL)
000E'	87		OR	A
000F'	20 03		JR	nz,NITCK3
0011'	2B		DEC	HL
0012'	1B F9		JR	LOPCK3
				;
00F4'				NITCK3:
00F4'	F5		PUSH	AF
00F5'	E5		PUSH	HL
00F6'				NITCK4:
00F6'	2B		DEC	HL
00F7'	7E		LD	A,(HL)
00FB'	FE 45		CP	'E'
00FA'	20 FA		JR	nz,NITCK4
00FC'	23		INC	HL
00FD'	54		LD	D,H
00FE'	5D		LD	E,L
00FF'	7E		LD	A,(HL)
0100'	FE 2D		CP	'-'
0102'	20 24		JR	nz,NOMINS
				;
0104'	E1		POP	HL
0105'	F1		POP	AF
0106'				LOPCK5:
0106'	3D		DEC	A
0107'	77		LD	(HL), A
0108'	FE 30		CP	'0'
010A'	30 0B		JR	nc,NITCK5
010C'	36 39		LD	(HL), '9'
010E'	2B		DEC	HL
010F'	7E		LD	A,(HL)
0110'	FE 2D		CP	'-'
0112'	20 F2		JR	nz,LOPCK5
				NITCK5:
0114'			INC	DE
0114'	13			

0085'	21 BF7C	LD	HL,CHRBUF+M-1
0088'	11 BF7D	LD	DE,CHRBUF+M
008B'	ED BB	LDOR	
008D'	23	INC	HL
008E'	36 31	LD	(HL), '1'
0090'	C3 014C'	JP	ENDCRCT
;			
0093'		MNSHT:	
0093'	01 0007	LD	BC,M-1
0094'	1B ED	JR	MNSHT1
;			
;# CASE OF DATA INCLUDE 'E' #			
0098'		EXIST:	
0098'	E1	POP	HL
0099'	0E 01	LD	C,I
0099'		LOOPCK1:	
009B'	7E	LD	A,(HL)
009C'	FE 45	CP	'E'
009E'	28 I3	JR	Z,PMTSET
00A0'	23	INC	HL
00A1'	0C	INC	C
00A2'	79	LD	A,C
00A3'	FE 0B	CP	M
00A5'	20 F4	JR	MZ,LOOPCK1
00A7'	ES	PUSH	HL
00AB'		LPCOK1:	
00AB'	7E	LD	A,(HL)
00AB'	FE 45	CP	'E'
00AB'	28 05	JR	Z,PMTST1
00AD'	36 00	LD	(HL),0
00AF'	23	INC	HL
00B0'	1B F6	JR	LPCOK1
00B2'		PMTST1:	
00B2'	E1	POP	HL
;			
00B3'		PWTSET:	
00B3'	2B	DEC	HL
00B4'	7E	LD	A,(HL)
00B5'	FE 39	CP	'9'
00B7'	C2 014C'	JP	MZ,ENDCRCT
00BA'	36 30	LD	(HL), '0'
00BC'		LOOPCK2:	
00BC'	2B	DEC	HL
00BD'	ES	PUSH	HL
00BE'	11 BF75	LD	DE,CHRBUF
00C1'	AF	JOR	A
00C2'	ED 52	SBC	HL,DE
00CA'	E1	POP	HL
00CS'	2B 10	JR	Z,NEITCK1
00C7'	7E	LD	A,(HL)
00CB'	FE 2E	CP	'.'
00CA'	2B F0	JR	Z,LOOPCK2
00CC'	3C	INC	A
00CD'	77	LD	(HL),A
00CE'	FE 3A	CP	OUTNUM

LOPCK5	0106'	LOPCK6	011E'	LOPCK7	012A'	LOPCK8	013F'
LPCK01	00AB'	M	000B	MNSHT1	0093'	MNSHT1	00B5'
MEXT01	0070'	MEITCK	00D7'	MEITQQ	0039'	NGEISI	0010'
NOMINS	0128'	NIIT000	004C'	NIIT001	0043'	NIITCK2	00EA'
NIITCK3	00F4'	NIITCK4	00F6'	NIITCK5	0114'	NIITCK7	0149'
OUTNUM	003A	PNTSET	00B3'	PNTST1	00B2'	PRDSHT	00B2'
RESET1	00E7'						

9.2

データ補正サブルーチンの使用例

演算サブルーチンの使用方法について、「 3.35×4.56 」という演算を例にとって説明します。

- ① CHRBUF (BF75H~BF80H) にアスキー型式で 3.35 をセットします。なお、データの後に 00H をセットし、データの終わりを示します。
次に、数値変換ルーチン (CONVT1) を CALL し、アスキー型式を HEX 型式に変換します。HEX 型式のデータは、FPACC (BF70H~BF74H) にセットされます。
- ② FPACC にセットされた HEX 型式のデータを、ユーザの定義する一時退避領域に保持しておきます。
- ③ ①と同様に、'4.56'+00H を CHRBUF にセットし、CONVT1 を CALL すると、FPACC に HEX 型式のデータがセットされます。
- ④ ②で退避させたデータを、A~E レジスタにセットします。
- ⑤ A~E レジスタおよび FPACC のデータを引数として、乗算のルーチン (MULTPL) を CALL します。結果は、FPACC に HEX 型式でセットされます。
- ⑥ キャラクタ変換ルーチン (CONVT2) を CALL すると、FPACC にセットされた HEX 型式のデータがアスキー型式に変換され、CHRBUF にセットされます。データの終りは、00H で識別されます。
- ⑦ 数値変換、キャラクタ変換により、若干の誤差が生じる可能性があるので、先に述べたデータ補正サブルーチン (CORRECT) を CALL します。
最終的な結果は、CHRBUF にアスキー型式でセットされます。

以上の手順を、次ページ 図 9-3 に示します。

また、上記の手順を実行するプログラム (FPTEST) のリストをリスト 29 に、その実行の中途結果の表示を図 9-4 に示します。

注) 図 9-4 は CP/M の下で FPTEST.MAC と CORRECT.MAC を M80 でアセンブルし、L80 でリンクして FPTEST.COM を作成し、RC-20 の ROM の内容を RAM の 4000H~5FFFH に配置した後、ZSID の下で FPTEST.COM を実行して得られたものです。

0115'	1A	LD	A, ^(DE)
0116'	FE 30	CP	'0'
0118'	C2 014C'	JP	NZ,ENDRCT
0119'	62	LD	H,D
011C'	6B	LD	L,E
011D'	23	INC	HL
011E'		LOPCK6:	
011E'	7E	LD	A, _(HL)
011F'	12	LD	(DE),A
0120'	87	OR	A
0121'	CA 014C'	JP	Z,ENDRCT
0124'	13	INC	DE
0125'	23	INC	HL
0126'	1B F6	JR	LOPCK6
		;	
0128'		NOMINS:	
0128'	E1	POP	HL
0129'	F1	POP	AF
		;	
012A'		LOPCK7:	
012A'	3C	INC	A
012B'	77	LD	(HL),A
012C'	FE 3A	CP	OUTNUM
012E'	C2 014C'	JP	NZ,ENDRCT
0131'	36 30	LD	(HL),'0'
0132'	2B	DEC	HL
0134'	7E	LD	A, _(HL)
0135'	FE 45	CP	'E'
0137'	20 F1	JR	NZ,LOPCK7
		;	
0139'	21 BF7F	LD	HL,CHRBUF+10
013C'	11 BF80	LD	DE,CHRBUF+11
013F'		LOPCK8:	
013F'	7E	LD	A, _(HL)
0140'	FE 45	CP	'E'
0142'	2B 05	JR	Z,NXTCK7
0144'	12	LD	(DE),A
0145'	2B	DEC	HL
0146'	1B	DEC	DE
0147'	1B F6	JR	LOPCK8
0149'		NXTCK7:	
0149'	3E 31	LD	A,'1'
014B'	12	LD	(DE),A
		;	
014C'		ENDRCT:	
014C'	C9	RET	
		;	
		END	

Macros:

Symbols:

CHRBUF	BF75	CORCT1	001B'	CORCT2	0034'	CORREC	00001'
ENDRCT	014C'	EXSIST	0098'	LOPCKX	009B'	LOPCKQ	0013'
LOOPEQ	0004'	LOOPME	0055'	LOPCK2	00BC'	LOPCK3	00ED'

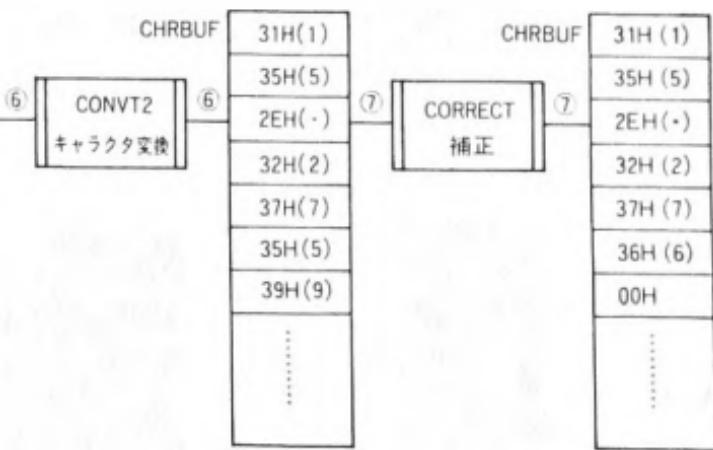
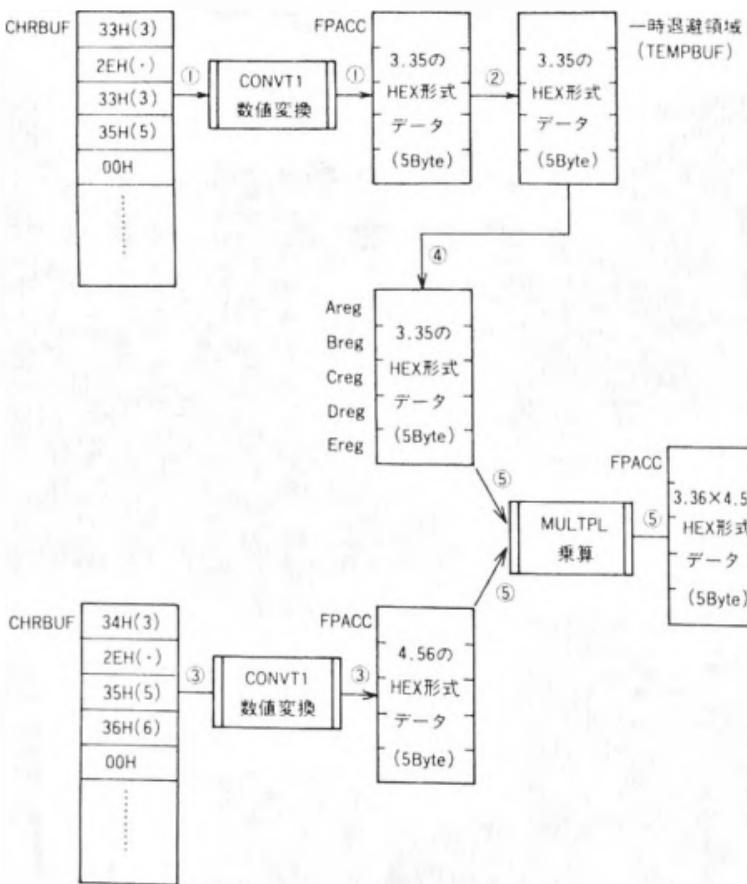


図9-3



リスト29 「3.35×4.56」の演算を行うプログラム (FPTEST)

MACRO-80 3.36 17-Mar-80 PAGE 1

```

;      #FPTEST#
.Z80
0000'      ASE6      ORG    100H
;      EXTRN  CORRECT
;
5A31      CONVT1 EQU     5A31H
5BC7      CONVT2 EQU     5BC7H
572B      MULTPL EQU     572BH
;
BF70      FPACC  EQU     0BF70H
BF75      CHRBUF EQU     FPACC+5
;
0005      COUNTER EQU     5
;
0100      START:
;
;
;      '3.35'->CHRBUF
;
;      ASCII CODE->HEX CODE
;
0100  21 013D      LD      HL,N3.35
0103  11 BF75      LD      DE,CHRBUF
0106  01 0005      LD      BC,COUNTER
0109  ED B0        LDIR
010B  CD 5A31      CALL   CONVT1
;
;      (FPACC)->TEMP BUFFR
;
010E  21 BF70      LD      HL,FPACC
0111  11 0147      LD      DE,TEMPBUF
0114  01 0005      LD      BC,COUNTER
0117  ED B0        LDIR
;
;      '4.56'->CHRBUF
;
;      ASCII CODE ->HEX CODE
;
0119  21 0142      LD      HL,N4.56
011C  11 BF75      LD      DE,CHRBUF
011F  01 0005      LD      BC,COUNTER
0122  ED B0        LDIR

```

0124	CD 5A31	CALL CONVT1
		; (TEMP BUFFER) -> A-FREQ
		;
0127	21 0147	LD HL,TEMPBUF
012A	7E	LD A,(HL)
012B	23	INC HL
012C	46	LD B,(HL)
012D	23	INC HL
012E	4E	LD C,(HL)
012F	23	INC HL
0130	56	LD D,(HL)
0131	23	INC HL
0132	5E	LD E,(HL)
		;
		;
		MULTIPLICATION
		;
0133	CD 572B	CALL MULTPL
		;
		HEX CODE -> ASCII CODE
		;
0136	CD 5BC7	CALL CONVT2
		;
		DATA CORRECTION
		;
0139	CD 0000\$	CALL CORRECT
		;
		;
013C	C9	RET
		;
013D	33 2E 33 35	N3.35:
0141	00	DB '3.35',0
0142		N4.56:
0142	34 2E 35 36	DB '4.56',0
0146	00	
0147		TEMPBUF:
0147		DS 5
		;
		END

Macros:

Symbols:

CHRBUF	BF75	CONVT1	5A31	CONVT2	5BC7	CORREC	013A#
COUNTE	0005	FPACC	BF70	MULTPL	572B	M3.35	013D
N4.56	0142	START	0100	TEMPBU	0147		

図9-4 FPTESTプログラムの実行の中途および最終結果

```
#6100,10B  
  
$010B  
#DBF70,BFBF  
BF70: 00 00 00 00 00 [33 2E 33 35 00] 82 57 78 E1 C1 C9 ← CHRBUFに '3.35' が  
. . . . . 3 . 3 5 . . M x . . . .  
BF80: 21 00 AA CD C4 BD DA 90 BF 3A 4D CB B7 CC A0 B9  
! . . . . . . . . . M . . . .  
#6,10E ← CALL CONVT1  
  
$010E  
#DBF70,BFBF  
BF70: [01 35 99 99 98] 33 2E 33 35 00 82 57 78 E1 C1 C9 ← FPACCに R3.35 の HEX  
. 5 . . . 3 . 3 5 . . M x . . .  
BF80: 21 00 00 00 05 60 00 00 00 00 00 00 00 00 00 70 BF  
! . . . . . . . . . P .  
#6,119  
  
$0119  
#D147,14A  
0147: 01 35 99 99 ← FPACC の内容が 退避  
. 5 . .  
#6,124 ← 領域(147~14A) に  
← コピーされた  
  
$0124  
#DBF70,BFBF  
BF70: 01 35 99 99 98 [34 2E 35 36 00] 82 57 78 E1 C1 C9 ← CHNBUF に '4.56' が  
. 5 . . . 4 . 5 6 . . M x . . .  
BF80: 21 00 00 00 05 60 00 00 00 00 00 00 00 00 00 70 BF  
! . . . . . . . . . P .  
#6,127 ← CALL CONVT1  
  
$0127
```

#DBF70,BFBF
 BF70: 01 4B F5 C2 8C 34 2E 35 36 00 82 57 78 E1 C1 C9 ← FPACCIに4,56のHEX
 + H . . . 4 . 5 6 . M * * * *
 BFB0: 21 00 00 00 06 B0 00 00 00 00 00 00 00 00 70 BF ← 型式データがセット
 P .
 #6,133

#0133
 #X
 -I-E [A=01 B=3599 D=9998] H=014B S=0100 P=0133 ← A～Eレジスタに,
 -Z-E A'00 B'0000 D'0000 H'0000 X=BF70 Y=BF71 CALL 572B 3,35のHEX型式データ
 #6,134
 #0136
 #DBF70,BFBF ← CALL MULTPL
 BF70: 01 F4 6A 7E E6 01 F4 6A 7E E7 4D D2 F2 00 35 99 ← 先豆の結果(HEX型
 + + j ^ + + + j ^ . M . . . 5 . 式)が, FPACCIにセ
 BFB0: 99 98 00 00 00 00 00 00 00 00 00 00 00 00 75 BF ットされた

#6,139
 ← CALL CONVT2
 #DBF70,BFBF
 BF70: 01 0E EB A4 00 31 35 2E 32 37 35 39 39 39 39 34 ← CHRBUFに, アスキ
 1 5 . 2 7 5 9 9 9 9 4 一型式でセットされ
 BFB0: 30 00 00 00 01 08 ED 50 00 00 F7 00 00 00 00 00 00 た
 0 P
 #6,13C
 ← CALL CORRECT
 #DBF70,BFBF
 BF70: 01 0E EB A4 00 31 35 2E 32 37 36 30 30 00 00 00 ← (データの補正)
 1 5 . 2 7 6 0 0 . . 最終的な答えが,
 BFB0: 00 00 00 00 01 08 ED 50 00 00 F7 00 00 00 00 00 00 00 CHRBUFにセットさ
 P れた

NUMBER

10

スケジューラ, メモ, ワールド タイムのデータの構造

10.1 ヘッダ・データの構造

表10-1 スケジューラ, メモ, ワールドタイムのファイル・ヘッダの構造

位置	名 称	内 容		
0	ファイル種別	00H	スケジュール・データ	
		01H	メモ・データ	
		03H	ワールド・タイム・データ	
1, 2	ファイル先頭アドレス	各ファイルの(HDの先頭からの)データの先頭アドレス		1バイト: 下位部分 2バイト: 上位部分
3, 4	ファイル終了アドレス	各ファイルの(HDの先頭からの)データの終了アドレス		3バイト: 下位部分 4バイト: 上位部分
5	レコード長	ファイルの1レコード長のバイト数を示す。		0000~73FHまでの範囲に入るレコード数とすること。
		ファイル種別	内 容	
		00H	ファイル中の1レコード長のバイト数(最大:FFH)	
		01H		
		03H	17バイト固定	
6, 7	レコード数	ファイル中のレコード数を示す。		0000~73FHまでの範囲に入るレコード数とすること。
		ファイル種別	内 容	
		00H	ファイル中のレコード数(最大:FFFFH)	
		01H		
		03H	ファイル中のレコード数(最大:FFFFH)	
8~11	システム予約	00, 00, 00, 00H		

スケジューラ、メモ、ワールドタイムのファイルの場合、ヘッダ・データ(MD)の構成、コモンヘッダの構造は、マシン語プログラムファイルの場合と同じです。(2.2「転送プログラムのフォーマット」を参照)

表10-1に、ファイル・ヘッダの構造をまとめました。

10.2 スケジューラ・データの構造

スケジューラ・データの構造の詳細を、次ページ図10-1、表10-2にそれぞれ示します。

レコードを配列するにあたっては、次の点について注意してください。

- ① アラーム種別について、以下の順にレコードを並べてください。

日アラーム	日アラーム	曜アラーム	曜アラーム	スケジュール・アラーム	スケジュール・アラーム	スケジュール・アラーム
-------	-------	-------	-------	-------------	-------------	-------------

- ② レコードは、それぞれ以下のように、時刻データをソートしておいてください。

日アラーム………A/P・時・分

曜アラーム………曜・A/P・時・分 (曜は日～土の順とする)

スケジュール・アラーム………年・月・日・A/P・時・分

- ③ レコードの数は、ファイルヘッダ中のレコード数と一致させてください。

- ④ 各アラームについて、以下の値をRC-20内で判断します。

日アラーム	7バイト	………A/Pデータ
	8バイト	………時データ
	9バイト	………分データ
曜アラーム	6バイト	………曜データ
	7バイト	………A/Pデータ
	8バイト	………時データ
スケジュール・アラーム	9バイト	………分データ
	3バイト	………年データ
	4バイト	………月データ
	5バイト	………日データ
	7バイト	………A/Pデータ
	8バイト	………時データ
	9バイト	………分データ

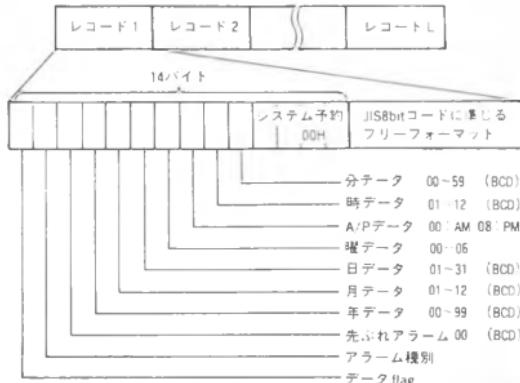
- ⑤ アラーム種別のアラームOFFは、データを表示する機能しかありません。

- ⑥ レコードのすべてがアラームOFFというデータは許されません。

- ⑦ 時刻データには、存在しない値は入力しないでください。

- ⑧ スケジュール・アラームデータは、1年以内とします。

図10-1
スケジュール・データの構造



(※)コメント文の空きエリアはすべて20H(Blank)を書き込んでおく

表10-2

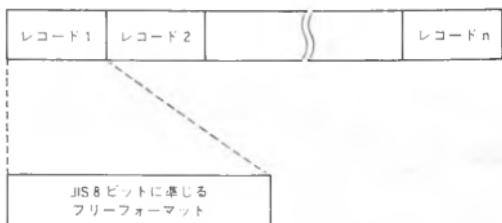
位置	名 称	内 容		BCD-20での対応	
0	データFlag	00H	データ無効		
		01H	データ有効		
1	アラーム種別	00H	アラームOFF	アラームはOFFされている	
		01H	日アラーム	毎日セットした時刻にアラームをならす	
		02H	曜アラーム	毎曜セットした時刻にアラームをならす	
		03H	スケジュール・アラーム	セットした年・月・日・時・分にアラームをならす	
2	先づれアラーム	00	先づれアラーム時刻	0 0	(BCD)
3	年データ	年データ	00~99 (BCD) 西暦の2桁		
4	月データ	月データ	01~12 (BCD)		
5	日データ	日データ	01~31 (BCD)		
6	曜データ	00	曜データ	SUN	
		01		MON	
		02		TUE	
		03		WED	
		04		THU	
		05		FRI	
		06		SAT	
7	A/Pデータ	AM/PM データ	00H:AM 08H:PM		
8	時データ	時データ	01~12 (BCD)		
9	分データ	分データ	00~59 (BCD)		
10 11	システム予約	00, 00, 00, 00H (将来の拡張に従う)			
12 13	コメント文	JIS 8bitコードに準ずる フリーフォーマット (コメント文) データのないところはすべて20H Blankを入れておくこと			

10.3 メモ・データの構造

メモ・データの構造を図10-2に示します。

レコードの最初に00Hがあると、空きレコードとなります。コメント文のデータのないところは、すべて20H（ブランク）してください。また、レコードの最初が00Hの場合も残りのデータはすべて20H（ブランク）にしてください。

図10-2 メモ・データの構造



10.4 ワールドタイム・データの構造

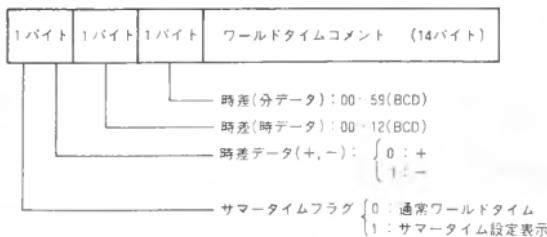
ワールドタイム・データの構造を図10-3に示します。

ワールドタイム・コメントのデータのないところは、20H（ブランク）を書き込んでおきます。

サマータイムを設定した時は、サマータイム時の時差データを記入します。サマータイムフラグは、サマータイム設定表示を行うだけで、時差計算は、時差データより求められます。

ホーム時刻は、基本時計をベースとします。

図10-3 ワールドタイム・データの構造



10.5 転送上での注意

スケジュール・データ、およびメモ・データには、空きレコードエリアを1レコード以上入れてください。

スケジュール・データの場合は、日アラーム、曜アラーム、スケジュール・アラームの各データの最後に、空きレコードエリアを1レコード以上入れておきます。

メモ・データの場合は、データの最後に空きレコードエリアを、1レコード以上入れておきます。これらを、図10-4に示します。

なお、空きエリアのレコードは、スケジュール・データ、メモ・データの場合、共に先頭の1バイトを00Hに設定します。

ただし、スケジュール・データの場合は、スケジュールのアラーム種別のデータ、および先ぶれアラーム時刻(00Hデータを設定)は、必ず設定してください。

図10-4

日アラ ーム データ	ヒアラ ーム データ	日アラ ーム 空	曜アラ ーム データ	曜アラ ーム 空	曜アラ ーム 空	スケ データ	スケ データ	スケ データ	スケ 空	スケ 空	スケ 空	スケ 空	スケ 空
メモ データ	メモ データ	メモ データ	メモ データ	メモ 空	メモ 空	メモ 空	メモ 空						

NUMBER

11

ダンプリストのプロ グラムの入力・転送

次節で提供する実用プログラムは、2つのソースリストを除き他はすべてダンプリストです。

ダンプリストはプログラムのヘーダとプログラム本体の全部を含んでいますが、これをすべてバイナリ・データとして、RC-20へ転送します。

実用プログラムを紹介する前に、ここでダンプリストのプログラムの入力方法と転送方法について説明しておきます。

11.1 QC-10の場合

CP/M上で実行します。まず、ダンプリストのプログラムを、DDTのSコマンドで、&H100よりメモリに書き込みます。次いで、SAVEコマンドによって、ディスクに格納します。

SAVE n ファイル名

ここでnはページ数であり、256バイトで1ページとなります。したがって、たとえば&H100～450に書き込んだ場合、プログラムの長さは&H350となりますから、n=4となります。

RC-20への転送プログラムをリスト64に示しますが、これはJBASIC上で走ります。

リスト64 QC-10用転送プログラム

```

10 INPUT "file name";F$
15 INPUT "rc-20 ready ok";X$
20 OPEN "R",#1,F$
25 OPEN "O",#2,"COM0:ICBN3F"
30 FIELD #1,128 AS A$
40 N=
50 GET #1,N
60 PRINT #2,A$;

```

```
70 N=N+1  
80 IF EOF() THEN 100  
90 GOTO 50  
100 CLOSE  
1010 END
```

11.2 PC-8001/8001mkII, PC-8801/8801mkIIのCP/Mの場合

ダンプリストのプログラムは、QC-10の場合と同様に、DDTのSコマンドで&H100よりメモリに書き込み、SAVEコマンドでディスクに格納します。

転送プログラム(TRANS)を、リスト65に示します。実行は、

A>TRANS ファイル名

で行います。ファイル名で、ダンプリストから作成したファイルの名前を指定します。

リスト65 PC-8001/8001mkII, PC-8801/8801mkIIのCP/M用転送プログラム

```
;      *TRANS*
;      TRANS COM FILE TO RC-20.
;      FOR PC-8001/PC-8001 MK2
;      & PC-8801/PC-8801 MK2
;
0100      ORG    100H
;
0005 =    B005  EQU    5
0000 =    B000  EQU    0
005C =    FCB   EQU    005CH
0080 =    BUFRR EQU    00B0H
00CE =    MODE  EQU    0CEH ;$16
0020 =    PORT0 EQU    20H
0021 =    PORT1 EQU    PORT0+1
;
;
0100 210B02    LXT    H,AREA
0103 220402    SHLD   BPOINT
;
0106 3A8000    LDA    BUFFR
0109 119C01    LXI    D,MSERR
010C B7        ORA    A
010D CA6601    JZ     EXIT
;
0110 0EOF      OPEN   FILE
                  MVI    C,15
```

```

0112 115C00    LXI   D,FCB
0115 CD0500    CALL  BDOS
0118 FEFF     CPI   OFFH
011A 11B301    LXI   D,MSOPNER
011D CA6601    JZ    EXIT
;
LOOP0:
0120 CD8201    CALL  READF
0123 C24401    JNZ   EOF
0126 2A0402    LHLD  BPOINT
0129 EB        XCHG
012A 218000    LXI   H,BUFFR
012D 0EB0     MVI   C,128
LOOP1:
012F 7E        MOV   A,M
0130 12        STAX  D
0131 23        INX   H
0132 13        INX   D
0133 0D        DCR   C
0134 C22F01    JNZ   LOOP1
0137 018000    LXI   B,128
013A 2A0402    LHLD  BPOINT
013D 09        DAD   B
013E 220402    SHLD  BPOINT
0141 C32001    JMP   LOOP0
;
EOF:
0144 0E10     MVI   C,16 ;FILE CLOSE
0146 115C00    LXI   D,FCB
0149 CD0500    CALL  BDOS
;
014C 2A0B02    LHLD  AREA+3
014F 220602    SHLD  COUNT
;
;
0152 11F101    LXI   D,MS000
0155 CD6C01    CALL  PRMSG
0158 0E01     MVI   C,1
015A CD0500    CALL  BDOS
;
015D CD8C01    CALL  SINIT
0160 CDDE01    CALL  SIO
;
0163 117101    LXI   D,DONMSG
EXIT:
0166 CD6C01    CALL  PRMSG
0169 C30000    JMP   BOOT
;

```

```

;
PRMSG:
016C 0E09      MVI    C,9
016E C30500    JMP    B005
;
DONMSG:
0171 0D0A202A4A DB     13,10.'$JOB COMPLETE$','$'
;
MSOPNER:
0183 202A494E50 DB     '$INPUT FILE OPEN ERROR$','$'
MSERR:
019C 202A4E4F20 DB     '$NO INPUT FILE NAME$$'
;
READF:
01B2 0E14      MVI    C,20
01B4 115C00    LXI    D,FCB
01B7 C00500    CALL   B005
01BA B7       ORA    A
01BB C9       RET
;
;
;
SINIT:
01BC 3E00      MVI    A,00H
01BE D321      OUT   PORT1
01C0 D321      OUT   PORT1
01C2 D321      OUT   PORT1
01C4 3E40      MVI    A,40H
01C6 D321      OUT   PORT1
01C8 3ECE      MVI    A,MODE
01CA D321      OUT   PORT1
01CC 3E15      MVI    A,15H
01CE D321      OUT   PORT1
01D0 C9       RET
;
OUTPUT:
01D1 F5       PUSH   PSW
WAIT:
01D2 DB21      IN    PORT1
01D4 1F       RAR
01D5 D2D201    JNC   WAIT
01D8 F1       POP   PSW
01D9 D320      OUT   PORT0
01DB C9       RET
;
;
;
```

```

S10:
01DC 2A0602 LHLD COUNT
01DF E5 PUSH H
01E0 C1 POP B
01E1 210802 LXI H,AREA
S102:
01E4 7E MOV A,M
01E5 CDD101 CALL OUTPUT
01E8 23 INX H
01E9 0B DCX B
01EA 7B MOV A,B
01EB B1 ORA C
01EC C2E401 JNZ S102
01EF C9 RET
;
;
01F0 C9 RET
;
;
MS000:
01F1 202A52432D DB 'SRC-20 READY OK? ', '$'
;
;
BPOINT:
0204 0000 DW 0
COUNT:
0206 0000 DW 0
0208 = AREA EQU $
;
;
020B END

```

11.3 PC-8001,8001mkIIのDUADPCの場合(N-BASIC)

ダンプリストのプログラムは、モニタのSコマンドで&H800より入力します。次いで、それをDUADPCのSAVERでディスクに格納します。

転送プログラムと、そのマシン語部分のリースリストを、リスト66、リスト67にそれぞれ示します。実行にあたっては、DUADPCを実行した後、LOADERでダンプリストから作成したファイルを、メモリ上にロードします。次いで、転送プログラムを実行します。

リスト66 転送プログラム

```

10 REM *****trans*****
20 CLEAR 100,6H0FFF
30 DEF USR=&H9000
40 FOR AD=&H9000 TO &H9037

```

```

50 READ DA$
60 POKE AD,VAL("&h"+DA$)
70 NEXT
80 INPUT "Irc-20 ready ok":I$
90 A=USR(I)
100 PRINT "#job complete"
110 END
120 DATA CD,07,90,CD,26,90,C9
130 DATA 3E,00,03
140 DATA 21,03,21,D3,21,3E,A6,03,21,3E,CE,03,21,3E,15,03
150 DATA 21,C9,F5,0B,21,1F,30,FB,F1,03,20,C9,E0,48,03,BB
160 DATA 21,00,98,7E,CD,1C,90,23,0B,7B,B1,20,F6,C9

```

リスト67 マシン語部分のソースリスト

```

;***S10***  

;  

    ORG 9000H  

BB00  AREA2: EQU 0B00H  

00CE  MODE: EQU 0CEH  

0020  PORT0: EQU ZOH  

0021  PORT1: EQU PORT0+1  

BB03  COUNT: EQU AREA2+3  

;  

9000 CD0790      CALL SINIT  

9003 CD2690      CALL SIO  

9006 C9          RET  

;  

;  

;  

9007 3E00        SINIT: LD A,0  

9009 D321        OUT (PORT1),A  

900B D321        OUT (PORT1),A  

900D D321        OUT (PORT1),A  

900F 3E40        LD A,40H  

9011 D321        OUT (PORT1),A  

9013 3ECE        LD A,MODE  

9015 D321        OUT (PORT1),A  

9017 3E15        LD A,15H  

9019 D321        OUT (PORT1),A  

901B C9          RET  

;  

901C F5          OUTPUT:PUSH AF  

901D DB21        WAIT: IN A,(PORT1)  

901F 1F          RRA  

9020 30FB        JR NC,WAIT

```

```

9022 F1      POP AF
9023 D320    OUT (PORT01,A
9025 C9      RET
;
9026 ED4B03BB S10: LD BC,(COUNT)
902A 2100BB   LD HL,AREA2
902D 7E      S102: LD A,(HL)
902E CD1C90   CALL OUTPUT
9031 23      INC HL
9032 0B      DEC BC
9033 7B      LD A,B
9034 B1      OR C
9035 20F6    JR NZ,S102
9037 C9      RET
;
9038          END

```

11.4 PC-8801/8801mkII(N88-BASIC)の場合

ダンプリストのプログラムを、モニタのSコマンドで&HC800より入力します。次いでそれを、BSAVEコマンドによってディスクに格納します。

転送プログラムを、リスト68に示します。

リスト68 PC-8801(DUAD)用転送プログラム

```

100 REM #transn#
110 CLEAR ,&HC000
120 INPUT "file name";F$
130 BLOAD F$
400 ZL=PEEK(&HC803)
410 ZH=PEEK(&HC804)
420 I=ZH*256+ZL
450 OPEN "COM1:N82N" FOR OUTPUT AS #1
460 INPUT "rc-20 ready ok";X$
470 FOR K=0 TO I
480   A=PEEK(&HC800+K)
490   PRINT #1,CHR$(A);
500 NEXT K
510 PRINT "#job complete#"
520 CLOSE #1
530 END

```

11.5 HC-20の場合

モニタのSコマンドで、ダンプリストを&H0C00よりメモリ上に書き込みます。次いで、モニタのAコマンドとWMコマンドを使うか、BASICのSAVEMコマンドを使って、それをテープ上に格納します。

転送プログラムを、リスト69に示します。転送プログラムは、テープより指定されたファイルを読み出し、それをRC-20に転送します。

リスト69 HC-20用転送プログラム

```
10 REM *TRANS*
20 MEMSET &H1400
30 WIND
40 INPUT "*FILE NAME";F$
50 LOADM "CAS0:";+F$
60 OPEN "0",#1,"COM0:(58
N2F)"
70 WIDTH "COM0:",255
80 INPUT "*RC-20 READY 0
K ";X$
90 ZL=PEEK(&H0C03)
100 ZH=PEEK(&H0C04)
110 I=ZH*256+ZL
120 FOR K=0 TO I
130 A=PEEK(&H0C00+K)
140 PRINT #1,CHR$(A);
150 NEXT K
160 PRINT "*JOB END"
170 CLOSE #1
180 END
```

11.6 HC-40/80/88のCP/Mの場合

QC-10などのCP/Mの場合と同様に、ダンプリストのプログラムをDBG80またはDDTで、&H100よりメモリ上に書き込み、SAVEコマンドで、ディスクに格納します。

転送プログラムを、リスト70に示します。実行は、

A>TRANS ファイル名

で行います。ファイル名でダンプリストから作成したファイルの名前を指定します。

リスト70 HC-40/80/88用転送プログラム

```

100 0000      ; $TRANS4
110 0000      ;
120 0100      ORG    100H
130 0005      B005  EQU    5
140 0000      BOOT   EQU    0
150 005C      FCB    EQU    005CH
160 0080      BUFFR  EQU    0080H
170 4000      AREA   EQU    4000H
180 4003      COUNT  EQU    4000H+3
190 0100      ;
200 0100      ;
210 0100 3A8000 LD     A,(BUFFR)
220 0103 114401 LD     DE,MSERR
230 0106 B7    OR     A
240 0107 CAC701 JP     Z,EXIT
250 010A CD5E01 CALL   MOVEF
260 010D 113001 LD     DE,MS001
270 0110 CDDF01 CALL   MESS
280 0113 0E01  LD     C,I
290 0115 CD0500 CALL   B005
300 0118 CDA001 CALL   S10
310 011B 112101 LD     DE,MSEND
320 011E CSC701 JP     EXIT
330 0121 0D0A202A MSEND:DB 13,10,'$job end$',13,10,'$'
340 0130 0D0A202A MS001:DB 13,10,'$RC-20 READY OK$',13,10,'$'
350 0144 UD0A202A MSERR:DB 13,10,'$no input file name$',13,10,'$'
360 015E          ;
370 015E          MOVEF:DS 0
380 015E 118000 LD     DE,BUFFR
390 0161 0E1A  LD     C,26
400 0163 CD0500 CALL   B005
410 0166          ; FILE OPEN.
420 0166 0EOF    LD     C,15
430 0168 115C00 LD     DE,FCB
440 016B CD0500 CALL   B005
450 016E FEFF    CP     OFFH
460 0170 11CD01 LD     DE,MSOPENR
470 0173 CAC701 JP     Z,EXIT
480 0176          LOOP0:DS 0
490 0176 CDA001 CALL   READF
500 0179 201C    JR     NZ,EOF
510 017B ED5BE501 LD     DE,(BPOINT)
520 017F          ;
530 017F 21B000 LD     HL,BUFFR
540 0182 0E80    LD     C,128
550 0184          LOOP1:DS 0

```

560 0184 7E LD A,(HL)
570 0185 12 LD (DE),A
580 0186 23 INC HL
590 0187 13 INC DE
600 0188 0D DEC C
610 0189 20F9 JR NZ,LOOP1
620 01BB 01B000 LD BC,128
630 01BE 2AE501 LD HL,(BPOINT)
640 0191 09 ADD HL,BC
650 0192 22E501 LD (BPOINT),HL
660 0195 180F JR LOOP0
670 0197 ;
680 0197 EDF:DS 0
690 0197 0E10 LD C,16 ;CLOSE.
700 0199 115C00 LD DE,FCB
710 019C CD0500 CALL BDOS
720 019F C9 RET
730 01A0 ;
740 01A0 READF:DS 0
750 01A0 0E14 LD C,20
760 01A2 115C00 LD DE,FCB
770 01A5 CD0500 CALL BDOS
780 01AB 87 OR A
790 01A9 C9 RET
800 01AA ;
810 01AA SIO:DS 0
820 01AA ED4B0340 LD BC,(COUNT)
830 01AE 210040 LD HL,AREA
840 01B1 SIO2:DS 0
850 01B1 7E LD A,(HL)
860 01B2 CDBC01 CALL OUTPUT
870 01B5 23 INC HL
880 01B6 0B DEC BC
890 01B7 7B LD A,B
900 01BB B1 OR C
910 01B9 20F6 JR NZ,SIO2
920 01BB C9 RET
930 01BC ;
940 01BC OUTPUT:DS 0
950 01BC E5 PUSH HL
960 01BD C5 PUSH BC
970 01BE 0E04 LD C,4
980 01C0 5F LD E,A
990 01C1 CD0500 CALL BDOS
1000 01C4 C1 POP BC
1010 01C5 E1 POP HL
1020 01C6 C9 RET
1030 01C7 ;

```

1040 01C7      EXIT:DS  0
1050 01C7 CDDF01  CALL    MESS
1060 01CA C30000  JP     BOOT
1070 01CD      MSOPNER:DS 0
1080 01CD 0D0A202A  DB     13,10,"open error",13,10,""
1090 01DF      MESS:DS  0
1100 01DF 0E09  LD     C,9
1110 01E1 CD0500  CALL    BDOS
1120 01E4 C9  RET
1130 01E5      BPOINT:DS 0
1140 01E5 0040  DW     AREA
1160 01E7      END
BDOS 0005 BOOT 0000 FCB 005C
BUFR 0080 AREA 4000 COUNT 4003
MSERR 0144 EXIT 01C7 MOVEF 015E
MS001 0130 MESS 01DF SIO 01AA
MSEND 0121 MSOPENE 01CD LOOP0 0176
READF 01A0 EOF 0197 BPOINT 01E5
LOOP1 01B4 S102 01B1 OUTPUT 01BC

```

11.7 FM-7の場合

モニタのMコマンドで、&H5000よりダンプリストのプログラムを、メモリに書き込みます。次いでそれを、BASICのSAVEMコマンドでディスク、またはテープに書き込みます。

転送プログラムを、リスト71に示します。転送プログラムは、指定されたファイルを、ディスクまたはテープより読み出し、RC-20に転送します。

リスト71 FM-7用転送プログラム

```

10 REM $trans$
20 CLEAR ,&H4FFF
30 INPUT "file name";F$
40 LOADM F$
50 J=PEEK(&H5003)
60 I=PEEK(&H5004)
65 I=J+I*256
70 OPEN "0",#1,"COM0:(SBN2)"
80 INPUT "#rc-20 ready ok ";I$
90 FOR K=0 TO I
100 A=PEEK(&H5000+K)
110 PRINT #1,CHR$(A);
120 NEXT K
130 PRINT "#job end"
140 CLOSE #1
150 END

```

NUMBER

12

実用マシン語 プログラム集

金利計算

ローンなどの単利、複利の金利計算を行うプログラムです。

まず、メニュー画面で単利か複利かを選びます。[タンリ] または [フクリ] のキーを押してください。

次に元金を聞いてきますので、画面の下半分に表示されている数字のキーを押して金額を入力します。間違えたときは [C] を押します。最後に [CR] キーを押して終ります。

次に利率を聞いてきますので、月利や年利などを入力します。元金の場合と同様に入力してください。単位はパーセントです。

次は期間です。月数や年数などを入力します。

ここまで入力が終了すると、計算結果が元利合計として表示されます。

[CR] キーを押すと次の計算へ進みます。また、[MENU] キーでメニュー画面へ、[MODE] キーで時計モードにもどります。

1

=M E N U =
· タンリ
· フクリ

メニュー画面。
単利計算か複利計算かを選ぶ

2

ガンキン?
¥ 20000
5 6 7 8 9 C
0 1 2 3 4

元金を画面の下半分に表示されている
キーで入力し、'CR' キーを押す。
'C' はクリアキー。

リリツ?
6. 5%
56789C
01234

利率(月利、年利など)を入力。

4 キカン?
12
567890
01234

期間（日数、年数など）を入力。

5 ガンリ
ゴウケイ
¥35600

計算結果が表示される。
'CR' キーで次の計算へ。
'MENU' キーでメニュー画面へ。
'MODE' キーで時計モードへ。

単利、複利とも同じ手順

プログラムリスト

ダンブリスト

Add +8 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum

ASCI

B988	CD	A5	BA	CD	DB	BA	CD	C1	43	3E	07	11	75	BF	CD	9B	:46
B918	BA	CD	21	42	0E	03	CD	B7	BA	CD	AC	BB	30	E8	CD	FE	:46
B920	B9	38	DB	CD	31	5A	CS	6A	B8	B7	BD	DD	3F	00	CD	C1	:18
B930	43	AF	II	70	B9	CD	9B	BA	CD	IA	42	21	3F	25	EB	EB	:FB
B948	BF	CD	A5	BA	CD	DB	BA	1B	8D	CD	BB	BB	3E	07	II	75	:46
B958	BF	CD	9B	36	25	CD	21	42	0E	06	CD	BT	BA	CD	AC	0D	:46
B969	BH	30	ED	C6	31	5A	CD	49	BB	CD	28	57	CD	65	B8	C9	:FB
B978	DB	DB	C2	3F	00	CD	7E	BB	CD	53	BB	CD	2B	57	B7	C2	:54
B980	2D	BB	CD	44	BB	CD	9F	58	CD	4E	BB	C3	2B	57	CD	6A	:46
B999	42	3E	07	11	C9	B9	CD	90	BA	CD	21	42	CD	7E	BB	CD	:34
B9A0	44	BB	CD	9F	58	CD	6A	BB	3A	AC	BD	D6	81	32	AC	BD	:46
B9B8	2B	0E	38	12	CD	53	BB	CD	28	57	B7	2B	EB	C3	2D	BB	:1F
B9C0	CD	4B	BB	CC	28	57	C7	79	BB	B9	BB	DD	CD	11	AB	B3	:36
B9D9	00	CD	C1	43	AF	II	EE	BB	CD	9B	BA	AF	CD	88	BB	BB	:C5
B9E8	C2	2D	BB	3E	0F	11	75	BF	CD	9B	BB	CD	05	41	BD	FE	:46
B9F9	DD	DB	28	2B	2B	2B	BA	DE	BS	B9	B2	2B	5C	00	21	75	:46

Sum DB DD C9 36 B9 4A 30 18 50 87 74 C6 00 9D 79 52 :7E

ASCI

Add	+8	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
BA80	BF	81	80	83	7E	B7	28	8D	78	2B	FE	82	28	45	2B	FE	:5E
BA10	81	28	33	37	C9	FE	2E	37	C8	23	18	E8	21	75	BF	TE	:75
BA20	FE	33	37	D8	FE	32	2B	18	23	7E	FE	36	3F	D8	FE	35	:C7
BA30	29	86	23	7E	FE	36	3F	D8	21	75	BF	7E	D8	36	47	79	:AB
BA40	C6	64	4F	18	FA	23	7E	D6	30	2B	87	47	79	C6	8A	4F	:38
BA50	10	FA	23	7E	D6	30	81	32	AC	BD	87	C9	CD	BC	BB	3E	:CF
BA60	31	32	75	BF	CD	31	5A	CD	6F	BB	3E	2E	32	75	BF	21	:D9
BA70	39	31	22	75	BF	CD	31	5A	C3	74	BB	CD	91	43	FE	32	:D3
BA80	28	89	FE	34	C8	FE	36	2B	F2	E1	C9	E1	E1	C3	9A	40	:7A
BA90	FE	28	D8	86	80	4F	21	E4	BF	89	7D	B7	F0	IA	B7	C8	:D5
BA00	77	13	23	18	F5	21	AB	8A	C3	15	41	35	35	37	38	39	:6C
BA10	43	30	31	32	33	34	32	CE	CD	91	43	B7	FE	11	DA	B7	:1D
BA20	FE	16	38	28	15	FE	26	DA	07	BB	28	19	FE	34	CA	:C5	
BA30	D9	BA	FE	36	20	E1	E1	C9	37	9C	CD	BC	BF	AD	32	:78	
BA40	A8	AB	SD	32	A9	B9	C9	3A	9A	BD	70	C0	32	32	AB	3D	:EA
BA50	AB	BD	77	28	0A	3E	30	32	75	BF	3E	81	32	AB	BD	3E	:2E

Sum 1C E1 DF 15 9E 0D B8 C8 6C 4B 42 A6 B8 F4 4E 79 :26

ASCI

Add	#B	+B	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
BB00	2E	18	06	C6	24	18	02	C6	BF	57	3A	A8	BD	B9	38	A7	:AB	
BB18	5F	B7	20	09	CD	BC	BB	7A	FE	38	28	B6	78	3C	32	A8	:EA	
BB20	BD	7A	16	08	21	75	BF	19	77	23	A7	79	C9	CD	6A	42	:BD	
BB30	21	3E	BD	CF	03	41	CD	91	43	B7	28	FA	E1	C9	45	52	:D6	
BB40	52	4F	52	21	21	C0	BD	18	00	21	C5	BD	18	08	21	BI	:6C	
BB50	BD	18	03	21	BB	BD	SE	5B	56	2B	4E	2B	46	2B	7E	C9	:AC	
BB60	11	AD	BD	18	3E	11	II	BD	18	39	11	B7	BD	18	34	11	:84	
BB70	BC	BD	18	2F	11	C1	BD	18	2A	21	AD	BD	18	20	21	B2	:27	
BB80	BD	18	1B	21	B7	BD	18	16	F5	CD	BC	BD	CD	C7	5B	F1	:CC	
BB90	C3	CA	BD	21	75	BF	II	FD	BF	01	0C	08	18	08	11	70	:18	
BA00	FB	18	03	21	7B	BF	01	B5	00	ED	BB	C9	3E	8B	06	01	:5B	
BBB0	EE	50	0E	FD	79	C9	21	EB	BF	06	07	18	08	21	75	BF	36	:18
BBC0	30	23	B6	08	36	00	23	10	FB	C9	F5	21	75	BF	E5	7E	:3E	
BBD0	B7	28	88	FE	45	CA	67	BC	23	I8	F4	AF	32	AA	BD	E1	:6F	
BBE0	05	08	7E	FE	28	21	23	18	F0	7E	BF	23	BD	FE	23	BF	:89	
BBF0	2E	18	10	36	30	23	E5	11	81	BF	AF	ED	52	E1	CA	23	:E1	

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
BC00	BD	18	E7	36	00	C3	23	BD	21	7C	BF	7E	FE	2E	C4	B3	:68	ス#6,テ#スリソ-,ル,
BC10	BC	E5	23	7E	B7	28	04	36	00	18	F7	E1	7E	FE	39	C2	:C2	ル#6,テ#スリソ-,ル,
BC20	23	BD	36	00	28	E5	11	75	BF	AF	ED	52	E1	28	10	7E	:28	ル#6,テ#スリソ-,ル,
BC30	FE	2E	28	F0	3C	77	FE	3A	C2	23	BD	36	00	18	E5	7E	:B2	ル#6,テ#スリソ-,ル,
BC40	FE	2D	28	1E	FE	2E	28	09	3C	77	FE	3A	C2	23	BD	36	:91	-L,-L,-L,-W:#26
BC50	30	01	00	21	7C	BF	11	7D	BF	ED	B8	23	BD	36	31	C3	:D4	0...111,1)ス#61テ
BC60	23	BD	01	00	18	ED	E1	0E	81	7E	FE	45	28	10	18	23	:61	0...111,1)ス#61テ
BC70	BC	79	FE	00	20	F4	E5	7E	FE	45	28	05	36	00	23	18	:E3	0...111,1)ス#61テ
BC80	F6	22	AA	BD	E1	18	03	22	AA	BD	28	7E	FE	39	C2	:C9	ス#6,テ#スリソ-,ル,	
BC90	BD	36	00	28	E5	11	75	BF	AF	ED	52	E1	28	10	7E	FE	:FB	ス#6,テ#スリソ-,ル,
BCA0	2E	28	F0	3C	77	FE	3A	C2	23	BD	36	00	18	E5	7E	FE	:B2	ス#6,テ#スリソ-,ル,
BCB0	2D	28	00	3C	77	FE	3A	C2	23	BD	36	31	18	03	23	36	:C8	-L,-W:#26
BCC0	31	21	8B	BF	7E	B7	28	03	28	1B	F9	F5	E5	2B	7E	FE	:A6	111...#,*,*,*,*+*
BCD0	45	28	FA	23	54	5D	7E	FE	2D	28	24	E1	F1	3D	77	FE	:A4	E:#TJ,-S:#W=
BCE0	38	00	36	39	28	7E	FE	2D	28	F2	I3	1A	FE	3C	22	DA	:00	00.69+-S:#W=
BCF0	23	BD	62	6B	23	7E	I2	B7	CA	23	BD	I3	23	1B	F6	E1	:E6	#Sbk#,キガネ#,#分
Sum	CE	22	50	E4	3F	DF	09	36	55	81	A6	98	56	9C	ID	E9	:8D	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
BD00	F1	3C	77	FE	3A	C2	23	BD	36	00	2B	7E	FE	45	28	F1	:E1	円くW:#ツス#68+-E 円
BD10	21	7F	BF	11	80	BF	7E	FE	45	28	05	12	28	1B	18	F6	:03	!ツ,-"! E(..,+..,分
BD20	3E	31	12	3A	AA	BD	E1	7B	28	46	2A	AA	BD	23	7E	FE	:A4	>1.:エキ(Feエラフ,-
BD30	28	05	21	00	01	E3	18	37	23	7E	D6	30	FE	03	38	64	:C6	(.,.1.7#~#)B,-8d
BD40	FE	88	30	60	F5	CD	93	BB	CD	BB	21	75	BF	36	30	:A5	,@#ツカシカシカ#68	
BD50	23	36	2E	23	C1	05	36	30	23	18	FB	11	F2	BF	IA	B7	:97	#6.#チ,60#,年ソ,キ
BD60	28	8D	FE	2E	28	01	77	13	23	B4	78	FE	09	38	EF	21	:02	(.,.1.-W,.X,-8!
BD70	75	BF	06	0C	7E	23	2E	28	04	I0	F8	18	24	21	75	:19	Uソ,.-#,(.,.1.-\$!u	
BD80	BF	54	50	06	0C	7E	B7	28	03	23	I0	F9	78	FE	0C	30	:C8	ソTJ,-#チ(.#,-X,-8
BD90	I3	84	2B	7E	FE	2E	28	08	FE	30	28	06	36	00	I8	EC	:AA	..,+..,(.,.0,-G,-8
BDA0	36	00	F1	C9	F1	C3	BC	BB	1B	C2	62	30	31	F7	22	FC	:3A	6..R/モテシサ,ツB#68
BDB0	C2	62	30	3A	I3	40	87	B2	62	30	F1	37	C9	22	FC	:3A	ツB#68,ツB#70,"?	
BDC0	3A	FE	3F	3D	32	FE	93	BD	CB	BB	21	75	BF	36	30	:31	: ?=2 Hヘシサ!U#68	
BDD0	23	36	2E	23	C1	05	36	30	23	18	FB	11	F2	BF	IA	B7	:97	#6.#チ,60#,年ソ,キ
BDE0	28	BD	FE	2E	28	01	77	13	23	84	78	FE	09	38	EF	21	:02	(.,.1.-W,.X,-8!
BDF0	75	BF	06	0C	7E	23	FE	2E	28	04	I0	F8	18	24	21	75	:19	Uソ,.-#,(.,.1.-\$!u
Sum	FA	B6	E5	27	68	ED	3E	1F	D8	ED	AF	33	0B	BE	8E	84	:F0	

ソースリスト

```

=====
RC-20 KINRI KEISAN by Y.ONO
=====
9/5 1984

ORG 0B816H
***** SYSTEM SUBROUTINE *****
STBY EQU 41C8H : Z-80 Standby & Key Restart Set
INT EQU 41A2H : Interrupt Code Set
ZEND EQU 409AH : Z-80 Control End
LORD EQU 40C6H : Load All Register From Stack
SARE EQU 4086H : Save All Register Into Stack
ERROR EQU 428DH : Display Of Error Code

INKEY EQU 43A3H : Inkey (Scan & Key Data Input)
PUTKEY EQU 4391H : Put Key
GETKEY EQU 409AH : Get Key
KEYIN EQU 4352H : Key In (Load Of Input Key Data)
DECKEY EQU 40A1H : Decode Of Key Data
KD1SP1 EQU 413BH : Key Display1
KD1SP2 EQU 4152H : Key Display2

```

```

4169      KDISP3 EQU 4169H      : Key Display3
4112      KALDIS EQU 4112H      : Key Of All Display (1-3 Page)
4115      KHYP23 EQU 4115H      : Key Of 2-3 Page Display
41F3      KHYP1 EQU 41F3H      : Key Of 1 Page Display
41F7      KHYP2 EQU 41F7H      : Key Of 2 Page Display
41FB      KHYP3 EQU 41FBH      : Key Of 3 Page Display
;
MUSIC     EQU 42AEH      : Music
5FE8      TIMER EQU 5FE8H      : Soft Timer (CS ADD ---> 411BH)
4320      RSOUT EQU 4320H      : RS-232C Output
4310      RSIN  EQU 4310H      : RS-232C Input
42E0      RSVERD EQU 42E0H      : Verify Of RS-232C Input Data(1)
42E5      RSVERI EQU 42E5H      : Verify Of RS-232C Input Data(2)
;
4106      HY0G1 EQU 4106H      : Display Of All Page
4109      HY123 EQU 4109H      : Display Of 1-3 Page
410C      HY023 EQU 410CH      : Display Of 2-3 Page
421A      HY00  EQU 421AH      : Display Of 0 Page
4221      HY01  EQU 4221H      : Display Of 1 Page
4228      HY02  EQU 4228H      : Display Of 2 Page
422F      HY03  EQU 422FH      : Display Of 3 Page
426A      WH1AL EQU 426AH      : Display Clear Of All Page
4257      WH101 EQU 4257H      : Display Clear Of 0 Page
4281      WH123 EQU 4281H      : Display Clear Of 1-3 Page
4247      WH10  EQU 427FH      : Display Clear Of 0 Page
425A      WH11  EQU 425AH      : Display Clear Of 1 Page
4263      WH12  EQU 4263H      : Display Clear Of 2 Page
4284      WH13  EQU 4284H      : Display Clear Of 3 Page
40D0      ASCII  EQU 40DH      : ASCII Convert (0-9)
40B1      ACIBC EQU 4081H      : ASCII Convert By BCD
43C1      BUFLC EQU 43C1H      : Character Buffer Clear
43B6      DISPLC EQU 43B6H      : Display Buffer Clear
4041      CHARA  EQU 4041H      : Character Of Display
411C      DISP   EQU 411CH      : Transfer Dot RAM <- V RAM
;
5A31      CONVT1 EQU 5A31H      : Number Convert (ASCII-->HEX)
5B27      CONVT2 EQU 5B27H      : Character Convert (ASCII-->HEX)
589F      ADDNUM EQU 589FH      : FPACC<--ABCDE:FPACC
58A8      SUBNUM EQU 58AH      : FPACC<--ABCDE:FPACC
572B      MULT   EQU 572BH      : FPACC<--ABCDE:FPACC
572C      DIVIDE EQU 572CH      : FPACC<--ABCDE:FPACC
572D      NEGSDN EQU 572DH      : FPACC<--0:FPACC
5F66      INVERT  EQU 5F66H      : FPACC<--1:FPACC
5F70      RANDOM EQU 5F70H      : FPACC<--Random Number
;
***** SYSTEM WORK AREA *****
;
BFAB      FTPAD EQU 0BFABH      : File Data Top Address
BFBD      FEDAD EQU 0BFBDH      : File Data End Address
BFAF      FLDAT EQU 0BFAFH      : Length Of Record
BF80      FRDCT EQU 0BF80H      : Number Of Record
BF82      SAHL  EQU 0BF82H      : Stack Area HL reg.
BF84      INTSET EQU 0BF84H      : Interrupt Set Area
BF85      KEYBU EQU 0BF85H      : Key Data Buffer reg.
BF86      ALFLAG EQU 0BF86H      : Key Data Buffer reg.
BF87      INTBL  EQU 0BF87H      : 1/64Hz Jump Table
BF89      INTFL  EQU 0BF89H      : Interrupt flag
BF8A      VPMEM  EQU 0BF8AH      : VRAM Memory
BF84      DRUFF  EQU 0BFE4H      : Graphic Display Buffer Address
BF84      DBUFF0 EQU 0BFF0H      : Page 0 Address
BF84      DBUFF1 EQU 0BFF0+7    : Page 1 Address
BF82      DBUFF2 EQU 0BFF1+7    : Page 2 Address
BF83      DBUFF3 EQU 0BFF2+7    : Page 3 Address
;
***** USER WORK AREA *****
;
BF70      FPACC EQU 0BF70H      : Floating Acc. Top Address
BF75      CHRBUF EQU 0BF75H      : Character Code Buffer Top Address
;
***** I/O ADDRESS *****
;
0D10      STW   EQU 0010H      : I/O 10 I/O00 sec Data Input
0030      WAI   EQU 0030H      : Watch Data Input sec
0031      WAD   EQU 0031H      : Watch Data Input min
0032      WA3   EQU 0032H      : Watch Data Input hour
0033      WA4   EQU 0033H      : Watch Data Input A/P & week
0034      WAS   EQU 0034H      : Watch Data Input day
0035      WA6   EQU 0035H      : Watch Data Input month
0036      WA7   EQU 0036H      : Watch Data Input year
0040      PKEYIN EQU 0040H      : Key Data Input
0050      PMUSIC EQU 0050H      : Music data Output
0060      PINKEY EQU 0060H      : Inkey (Scan & Data)
00E0      PEND  EQU 00E0H      : Z-80 Mode End
;
;
;
```


B888	1E 07	GAN2:	LD E.7
B88D	FE 07		CP 7
B88F	38 05		JR C.GAN3
B8C1	1E 00		LD E.0
B8C2	CD 43C1		CALL BUFCL
B8C6	CD 43C1	GAN3:	LD A-E
B8C7	16 00		LD D.0
B8C9	21 BFE4		LD HL,DRUFF
B8CC	19		ADD HL,DE
B8CD	36 5C		LD (HL),“*”
B8CF	3C		INC A
B8D0	11 BF75		LD DE,CHRBUF
B8D3	CD BA90		CALL PUTCHR
B8D5	CD 421A		CALL HYD0
B8D9	CD 4221		CALL HYD1
B8DC	0F 09		LD C.9
B8DE	CD BAB7		CALL INPUT
B8E1	CD BBAC		CALL SOUND
B8E4	30 BD		JR NC.GAN1
B8E6	CD 5A31	:	CALL CONVT1
B8E9	C3 B860	:	JP TFRRA1
B8EC	B6 DE DD B7	M2:	DB 0B6H.0DEH.0DDH.0B7H.0DDH.“?”,0
B8F0	0D 3F 00	:	
B8F3	CD 43C1	KAIISU:	LD LEVEL 2
B8F6	AF		CALL BUFCL
B8F7	11 B929		XOR A
B8F8	CD B900		LD DE,M3
B8FD	CD 421A		CALL PUTCHR
B900	CD BA55		CALL HYD0
B903	CD BADB		CALL KEYDSP
B906	CD 43C1		CALL CLK
B909	3E 07	KAIISI:	CALL BUFL
B90B	11 BF75		LD A.7
B90C	CD B900		LD DE,CHRBUF
B911	CD 4221		CALL PUTCHR
B914	0E 83		LD HYD1
B916	CD BAB7		LD C.3
B919	CD BBAC		CALL INPUT
B91C	30 E8		CALL SOUND
B91E	CD B9FE		JR NC.KAIISI
B921	38 00		CALL CNUBIN
B923	CD 5A31		JR : for FUKUR
B926	C3 B86A		CALL CONVT1
B929	B7 B6 DD 3F		JP TFRRA3
B92D	00	:	
B90E	CD 43C1	RIRITU:	LD LEVEL 2
B92E	AF		CALL BUFCL
B931	11 B970		XOR A
B932	CD BA90		LD DE,M4
B938	CD 421A		CALL PUTCHR
B93B	21 253F		CALL HYD0
B93E	22 BFE8		LD HL,253FH : %
B941	CD BA55		LD (DRUFF)).HL
B944	CD BADB		CALL KEYDSP
B947	18 BD		CALL CLK
B949	CD BB85	RIRIT1:	LD RIR12
B94C	3E 07		CALL BUFL
B94E	11 BF75		LD A.7
B951	CD BA90		LD DE,CHRBUF
B954	30 25		CALL PUTCHR
B956	CD 4221		LD (HL),“*”
B959	0E 86		LD HYD1
B95B	CD BAB7		LD C.6
B95E	CD BBAC		CALL INPUT
B961	30 E6		CALL SOUND
B963	CD 5A31		JR NC.RIRIT1
B966	CD BB49		CALL CONVT1 : ACC=RIRIT1
B969	CD 572B		CALL TFRPR : ACC=RIRIT1(%)
B96C	CD B865		CALL MULTPL
B96F	C9		CALL TFRAA2
B970	DB D8 C2 3F	M4:	DB 0DBH.0DH.0C2H.“?”,0
B974	00	:	


```

; BAI.FC
; 21 BYT$5
; HL.CHRBUF
; A : HL.J
; -3*
; x100

BA23: DB FE 32 RET C
; HL.CHRBUF
; A : HL.J
; -3*
; x100

BA22: DB 20 10 RET CP CCF
; HL.CHRBUF
; A : HL.J
; -6*
; x100

BA21: FE 36 RET CCF
; HL.CHRBUF
; A : HL.J
; -6*
; x100

BA20: DB 35 RET RET
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA19: DB 06 20 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA18: DB 23 BA32 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA17: DB 33 BA33 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA16: DB 36 BA36 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA15: DB 37 BA37 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA14: DB 21 BA45 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA13: DB 30 BA47 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA12: DB 30 BA48 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA11: DB 30 BA49 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA10: DB 47 BA4A RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA09: DB 79 BA4B RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA08: DB 64 BA4C RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA07: DB 4F BA4D RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA06: DB 10 BA4E RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA05: DB 23 BA4F RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA04: DB 30 BA49 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA03: DB 30 BA50 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA02: DB 30 BA51 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA01: DB 30 BA52 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA00: DB 30 BA53 RET CP C
; HL.CHRBUF
; A : HL.J
; -5*
; x100

BA5C: CD 3E 31 CALL CLRBUF
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA5F: CD 52 9A CALL CONVT1
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA60: CD 5A 95 CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA61: CD BFFF CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA62: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA63: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA64: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA65: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA66: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA67: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA68: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA69: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA70: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA71: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA72: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA73: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA74: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA75: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA76: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA77: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA78: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA79: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA7A: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA7B: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA7C: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA7D: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA7E: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA7F: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA80: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA81: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA82: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA83: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA84: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA85: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA86: CD 3E 3E CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA87: CD 20 F2 CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA88: E1 CB CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA89: E1 CB CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA8A: E1 CB CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA8B: E1 CB CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA8C: E1 CB CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA8D: E1 CB CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA8E: E1 CB CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA8F: E1 CB CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA90: F0 28 CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA91: 06 00 CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA92: 4F 00 CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

BA93: 21 BFE4 CALL TFFM
; HL.CHRBUF
; A : HL.J
; -1*
; x100

```


BB1D	3C	NUM4:	INC	A
BB1E	32	BDA8	LD	(CBP),A
BB21	7A		LD	A,D
BB22	16	00	NUM5:	D,0
BB23	21	BF75	LD	HL,CHRBUF
BB27	19		ADD	HL,DE
BB28	77		LD	(HL),A
BB29	23		INC	HL
BB2A	AF		XOR	A
BB2B	77		LD	(HL),A
BB2C	C9		RET	
;				
BB2D	CD	426A	ERRDSP:	: LEVEL 2
BB2D	21	BB3E	CALL	WHIAL
BB30	CD	41F3	LD	HL,EDAT
BB33	CD	4391	ERR1:	CALL KHYPI
BB36	CD	41F1	CALL	PUTKEY
BB39	B7		OR	A
BB3A	28	FA	JR	Z,ERR1
BB3C	E1		POP	HL : LEVEL1
BB3D	C9		RET	
;				
BB3E	45	52	EDAT:	DB "ERROR!"
BB42	52	21	;	
;				
BB44	21	BDCC	TFRIR:	LD HL,1CHI+4
BB47	18	0D	JR	TFR
BB49	21	BDCC	TFRPR:	LD HL,PCNT+4
BB4C	18	08	JR	TFR
BB4E	21	BDB1	TFRA1R:	LD HL,ACC1+4
BB51	18	03	JR	TFR
BB53	21	BDBB	TFRA3R:	LD HL,ACC3+4
BB56	5E		TFRH:	LD E,(HL)
BB57	2B		DEC	HL
BB58	56		LD	D,(HL)
BB59	2B		DEC	HL
BB5A	42		LD	C,(HL)
BB5B	2B		DEC	HL
BB5C	46		LD	B,(HL)
BB5D	2B		DEC	HL
BB5E	7E		LD	A,(HL)
BB5F	C9		RET	
;				
BB60	11	B0AD	TFRA1S:	LD DE,ACC1
BB63	18	3E	JR	TFRSAX
BB65	11	B0B2	TFRA2S:	LD DE,ACC2
BB68	18	39	JR	TFRSAX
BB6A	11	B0B7	TFRA3S:	LD DE,ACC3
BB6A	18	34	JR	TFRSAX
BB6F	11	BDBC	TFRA1:	LD DE,1CH1
BB72	18	2F	JR	TFRSAX
BB74	11	BDC1	TFRAP:	LD DE,PCNT
BB77	18	2A	JR	TFRSAX
BB79	21	B0AD	TFRA1A:	LD HL,ACC1
BB79	18	20	JR	TFRSKA
BB7E	21	B0B2	TFRA2A:	LD HL,ACC2
BB81	18	1B	JR	TFRSKA
BB83	21	B0B7	TFRA3A:	LD HL,ACC3
BB86	18	16	JR	TFRSKA
;				
BB88	F5		TFRAC:	PUSH AF
BB89	CD	BBBC		CALL CLRBCF
BB8C	CD	5BC7		CALL CONVT2
BB8F	F1			POP AF
BB90	C3	BBCA		JP CONNECT
;				
BB93	21	BF75	TFRC0:	LD HL,CHRBUF
BB96	11	BPF2	LD	DE,OBUFF2
BB99	01	0B0C	LD	BC,12
BB9C	18	0B	JR	TFR

BC0E:	CA	BC03 [*]				
BC11:	E5					
BC12:	23					
BC13:	7E					
BC14:	B7					
BC15:	28	84				
BC17:	36	00				
BC19:	18	F7				
BC1B:						
BC1C:	E1					
BC1D:	7E					
BC1F:	C2	BD23 [*]				
BC22:	36	30				
BC24:						
BC2A:	28					
BC2B:	E5					
BC2C:	A7	BF75				
BC2D:	ED	52				
BC2E:	E1					
BC2F:	28	10				
BC30:	FE	2E				
BC31:	28	F0				
BC34:	3C					
BC35:	FE					
BC38:	36	30	23 [*]			
BC3B:	BC2D:	18	E5			
BC3F:						
BC40:	FE	2D				
BC41:	FE	1E				
BC42:	FE	1E				
BC43:	28	99				
BC44:	BC48:	3C				
BC45:	28	99				
BC46:	BC48:	77				
BC47:	FE	3A				
BC48:	IC4C:	C2	BD23 [*]			
BC49:	IC4F:	36	31			
BC51:	01	00	08			
BC52:	BC52:	01	00	07		
BC54:	21	BF7C				
BC57:	11	BF7D				
BC5A:	ED	BB				
BC5D:	23					
BC5F:	36	31				
BC62:	C3	BD23 [*]				
BC65:	18	ED				
BC67:	E1					
BC68:	0E	91				
BC6A:						
BC6B:	7E					
BC6B:	FE	45				
BC6D:	28	18				
BC6F:	23					
BC70:	OC					
BC71:	79					
BC72:	FE	9B				
BC74:	20	F4				
BC76:	E5					
BC77:	7E					
BC78:	FE	45				
BC79:	FE	45				
BC7C:	28	05				
BC7C:	36	00				
BC7E:	23					
BC7F:	18	F6				
BC81:						
BC81:	22	BD2A [*]				
BC84:	E1					
BC85:	18	83				
BC87:	22	BD2A [*]				
BC8A:	2B					

NEXT001:	INC					
	LD	A,(HL)				
	OR	A				
	JR	Z,NEXT000				
	LD	(HL),0				
NEXT001:	JR	NEXT001				
PDP:						
LOOPNE1:	DEC					
	PUSH					
	LD	HL				
	XOR	A				
	LD	DE,CHRRBUF				
	DE	HL,DE				
	POP					
	JR	Z,NEXT01				
	LD	A,(HL)				
	CP	"_"				
	JR	Z,LOOPNE1				
	A	INC				
	LD	HL,1,A				
	CP	OUTNUM				
	LD	NP,ENDCFT				
	JP	(HL),*_2				
	LOOPNE1					
	LD	JR				
	CP	A,(HL)				
	JR	Z,MINSHT				
	LD	INC				
	LD	Z,MINSHT				
	LD	DE,CHRRBUF+4 [*]				
	LD	LD				
	LD	LDRR				
	INC	HL				
	LD	(HL),*_1				
	JP	NZ,ENDCFT				
	INC	HL				
	INC	C				
	LD	A,C				
	CP	M				
	JR	NZ,LOOPCK1				
	PUSH	HL				
	LOOPCK1:					
	LD	A,(HL)				
	CP	"_*				
	JR	Z,PO1NTSET1				
	LD	(HL),0				
	INC	HL				
	JR	PO1NTSET1				
	LD	(EE),HL				
	RD	HL				
	JP	_7BR				
	JR					
	PO1NTSET1:					
	LD	(EE),HL				
	RD	HL				

BC88'	7E	LD	A. (HL)
BC8C'	FE 35	CP	"9"
BC8E'	C2 BD23'	JP	NZ, ENDRCRT
BC91'	36 30	LD	(HL), "0"
BC93'		LOOPCK2:	
BC94'	28	DEC	HL
	E5	PUSH	HL
BC95'	11 BF75	LD	DE, CHRBUF
BC98'	AF	XOR	A
BC99'	ED 52	SBC	HL, DE
BC9B'	E1	POP	HL
BC9C'	28 10	JR	Z, NEXTCK1
BC9E'	7E	LD	A. (HL)
BC9F'	F1 2E	CP	"= "
BCA1'	28 F0	JR	Z, LOOPCK2
BCA3'	3C	INC	A
BCA4'	77	LD	(HL), A
BCA5'	FE 3A	CP	OUTNUM
BCA7'	C2 BD23'	JP	NZ, ENDRCRT
BCAA'	36 30	LD	(HL), "0"
BCAC'	18 E5	JR	LOOPCK2
BCAE'		NEXTCK1:	
BCAF'	7E	LD	A. (HL)
BCB1'	FE 2D	CP	"= "
BCB1'	28 0B	JR	Z, RESET
BCB3'	3C	INC	A
BCB4'	77	LD	(HL), A
BCB5'	FE 3A	CP	OUTNUM
BCB7'	C2 BD23'	JP	NZ, ENDRCRT
BCBA'	36 31	LD	(HL), "1"
BCBC'	18 03	JR	NEXTCK2
BCBE'		RESET1:	
BCBE'	23	INC	HL
BCBF'	36 31	LD	(HL), "1"
BCC1'		NEXTCK2:	
BCC1'	21 BF80	LD	HL, CHRBUF+11
BCC4'		LOOPCK3:	
BCC4'	7E	LD	A. (HL)
BCC5'	B7	OR	A
BCC6'	20 03	JR	NZ, NEXTCK3
BCCB'	28	DEC	HL
BCCB'	18 F9	JR	LOOPCK3
BCCB'		NEXTCK3:	
BCCB'	F5	PUSH	AF
BCCD'	E5	PUSH	HL
BCCD'		LOOPCK4:	
BCCD'	2B	DEC	HL
BCCF'	FE 45	LD	A. (HL)
BCD1'	20 FA	CP	"E"
BCD3'	23	JR	NZ, LOOPCK4
BCD4'	54	INC	HL
BCD5'	5D	LD	D, H
BCD6'	7E	LD	E, L
BCD7'	FE 2D	CP	A. (HL)
BCD9'	20 24	JR	"= "
BCDB'	E1	POP	HL
BCDC'	F1	POP	AF
BCDD'		LOOPCK5:	
BCDD'	3D	DEC	A
BCDE'	77	LD	(HL), A
BCDF'	FE 30	CP	"0"
BCE1'	30 08	JR	NC, NEXTCK5
BCE3'	36 39	LD	(HL), "9"
BCE5'	2B	DEC	HL
BCE6'	7E	LD	A. (HL)
BCE7'	FE 2D	CP	"= "
BCE9'	20 F2	JR	NZ, LOOPCK5
BCEB'	13	INC	DE
BCEC'	1A	LD	A. (DE)
BCED'	FE 3B	CP	"0"
BCE7'	C2 BD23'	JP	NZ, ENDRCRT
BCF2'	62	LD	H, D
BCF3'	6B	LD	L, E
BCF4'	23	INC	HL
BCF5'		LOOPCK6:	
BCF5'	7E	LD	A. (HL)
BCF6'	12	LD	(DE), A
BCF7'	B7	OR	A
BCF8'	CA BD23'	JP	Z, ENDRCRT
BCF9'	11	INC	DE
BCFC'	23	INC	HL
BCFD'	18 F6	JR	LOOPCK6

		NOMINS:	HL
		POP	AF
EBCFF·	E1		
E0D0·	F1		
E0D1·	3C	LOOPCK7;	INC
E0D2·	71		A (HL), A
E0D3·	FF 3A	LD	OUTA
E0D4·	E0D3·	CP	OUTA
E0D5·	C2 E0D3·	JP	NEOKRCT
E0D6·	36 30	LD	(HL), "0"
E0D7·	28	DBE;	HL
E0D8·	7E	LD	A (HL)
E0D9·	FE 45	CP	"E"
E0D0·	20 F1	JR	NZ, LOOPCK7
E0D1·	21 BF7F		
E0D2·	11 BF80	LOOPCK8;	LD
E0D3·	7E	LD	HL, CRBUF+10
E0D4·	0016·	DE;	CRBUF+11
E0D5·	0017·		
E0D6·	FE 45	CP	A (HL)
E0D7·	28 05	JR	2, NEXTCK7
E0D8·	E0D9·	LD	(GP), A
E0D9·	12	DBE;	HL
E0D0·	2B	DE;	DE
E0D1·	E0D2·	JR	LOOPCK8
E0D2·	1B		A, "1"
E0D3·	1B F6	LD	(DE), A
E0D4·	0020·		
E0D5·	3E 31	NEXTCK7;	
E0D6·	0021·		
E0D7·	12	EN0C7;	
E0D8·			ENDC7;
E0D9·		LD	A, (EE)
E0D0·	3A BDAA·	OR	A
E0D1·	E0D2·	JR	Z, END4
E0D2·	2B 46	LD	HL, (EE)
E0D3·	2A BDAA·	INC	
E0D4·	23	LD	A (HL)
E0D5·	E0D2·	CP	Z, END0
E0D6·	7E	JR	HL, 010H
E0D7·	0022·	LD	(SP), HL
E0D8·	FE 05	EX	END4
E0D9·	0019·	JR	HL
E0D0·	2B 06	LD	A (HL)
E0D1·	0020·	SUB	3DH
E0D2·	3E 01 00	CP	3
E0D3·	E0D4·	JR	C, ZERO
E0D4·	18 37	LD	HL
E0D5·	0023·	PUSH	HL, ZERO
E0D6·	1B 37	CALL	TRCD
E0D7·	0024·	CD	CLRCBF
E0D8·	FS 60	BBR3·	HL, CRBUF
E0D9·	0025·	CD	(HL), "0"
E0D0·	BBR3·	BBR3·	
E0D1·	CD BBRC·	LD	
E0D2·	21 BF75	INC	
E0D3·	0026·	INC	
E0D4·	3E 30	INC	
E0D5·	0027·	INC	
E0D6·	2B 06	INC	
E0D7·	0028·	INC	
E0D8·	3E 2E	INC	
E0D9·	0029·	INC	
E0D0·	2B 00	INC	
E0D1·	002A·	INC	
E0D2·	2B 01	INC	
E0D3·	002B·	INC	
E0D4·	2B 02	INC	
E0D5·	002C·	INC	
E0D6·	3E 30	INC	
E0D7·	002D·	INC	
E0D8·	2B 03	INC	
E0D9·	002E·	INC	
E0D0·	1B 0F	INC	
E0D1·	11 BF72	INC	
E0D2·	1A 0F	INC	
E0D3·	002F·	INC	
E0D4·	1A 0F	INC	
E0D5·	0030·	INC	
E0D6·	1B 0F	INC	
E0D7·	0031·	INC	
E0D8·	2B 01	INC	
E0D9·	0032·	INC	
E0D0·	1B 0F	INC	
E0D1·	0033·	INC	
E0D2·	2B 01	INC	
E0D3·	0034·	INC	
E0D4·	2B 02	INC	
E0D5·	0035·	INC	
E0D6·	3E 30	INC	
E0D7·	0036·	INC	
E0D8·	2B 03	INC	
E0D9·	0037·	INC	
E0D0·	1B 0F	INC	
E0D1·	0038·	INC	
E0D2·	2B 01	INC	
E0D3·	0039·	INC	
E0D4·	2B 02	INC	
E0D5·	003A·	INC	
E0D6·	3E 30	INC	
E0D7·	003B·	INC	
E0D8·	2B 03	INC	
E0D9·	003C·	INC	
E0D0·	1B 0F	INC	
E0D1·	003D·	INC	
E0D2·	2B 01	INC	
E0D3·	003E·	INC	
E0D4·	2B 02	INC	
E0D5·	003F·	INC	
E0D6·	3E 30	INC	
E0D7·	0040·	INC	
E0D8·	2B 03	INC	
E0D9·	0041·	INC	
E0D0·	1B 0F	INC	
E0D1·	0042·	INC	
E0D2·	2B 01	INC	
E0D3·	0043·	INC	
E0D4·	2B 02	INC	
E0D5·	0044·	INC	
E0D6·	3E 30	INC	
E0D7·	0045·	INC	
E0D8·	2B 03	INC	
E0D9·	0046·	INC	
E0D0·	1B 0F	INC	
E0D1·	0047·	INC	
E0D2·	2B 01	INC	
E0D3·	0048·	INC	
E0D4·	2B 02	INC	
E0D5·	0049·	INC	
E0D6·	3E 30	INC	
E0D7·	004A·	INC	
E0D8·	2B 03	INC	
E0D9·	004B·	INC	
E0D0·	1B 0F	INC	
E0D1·	004C·	INC	
E0D2·	2B 01	INC	
E0D3·	004D·	INC	
E0D4·	2B 02	INC	
E0D5·	004E·	INC	
E0D6·	3E 30	INC	
E0D7·	004F·	INC	
E0D8·	2B 03	INC	
E0D9·	0050·	INC	
E0D0·	1B 0F	INC	
E0D1·	0051·	INC	
E0D2·	2B 01	INC	
E0D3·	0052·	INC	
E0D4·	2B 02	INC	
E0D5·	0053·	INC	
E0D6·	3E 30	INC	
E0D7·	0054·	INC	
E0D8·	2B 03	INC	
E0D9·	0055·	INC	
E0D0·	1B 0F	INC	
E0D1·	0056·	INC	
E0D2·	2B 01	INC	
E0D3·	0057·	INC	
E0D4·	2B 02	INC	
E0D5·	0058·	INC	
E0D6·	3E 30	INC	
E0D7·	0059·	INC	
E0D8·	2B 03	INC	
E0D9·	005A·	INC	
E0D0·	1B 0F	INC	
E0D1·	005B·	INC	
E0D2·	2B 01	INC	
E0D3·	005C·	INC	
E0D4·	2B 02	INC	
E0D5·	005D·	INC	
E0D6·	3E 30	INC	
E0D7·	005E·	INC	
E0D8·	2B 03	INC	
E0D9·	005F·	INC	
E0D0·	1B 0F	INC	
E0D1·	0060·	INC	
E0D2·	2B 01	INC	
E0D3·	0061·	INC	
E0D4·	2B 02	INC	
E0D5·	0062·	INC	
E0D6·	3E 30	INC	
E0D7·	0063·	INC	
E0D8·	2B 03	INC	
E0D9·	0064·	INC	
E0D0·	1B 0F	INC	
E0D1·	0065·	INC	
E0D2·	2B 01	INC	
E0D3·	0066·	INC	
E0D4·	2B 02	INC	
E0D5·	0067·	INC	
E0D6·	3E 30	INC	
E0D7·	0068·	INC	
E0D8·	2B 03	INC	
E0D9·	0069·	INC	
E0D0·	1B 0F	INC	
E0D1·	006A·	INC	
E0D2·	2B 01	INC	
E0D3·	006B·	INC	
E0D4·	2B 02	INC	
E0D5·	006C·	INC	
E0D6·	3E 30	INC	
E0D7·	006D·	INC	
E0D8·	2B 03	INC	
E0D9·	006E·	INC	
E0D0·	1B 0F	INC	
E0D1·	006F·	INC	
E0D2·	2B 01	INC	
E0D3·	0070·	INC	
E0D4·	2B 02	INC	
E0D5·	0071·	INC	
E0D6·	3E 30	INC	
E0D7·	0072·	INC	
E0D8·	2B 03	INC	
E0D9·	0073·	INC	
E0D0·	1B 0F	INC	
E0D1·	0074·	INC	
E0D2·	2B 01	INC	
E0D3·	0075·	INC	
E0D4·	2B 02	INC	
E0D5·	0076·	INC	
E0D6·	3E 30	INC	
E0D7·	0077·	INC	
E0D8·	2B 03	INC	
E0D9·	0078·	INC	
E0D0·	1B 0F	INC	
E0D1·	0079·	INC	
E0D2·	2B 01	INC	
E0D3·	007A·	INC	
E0D4·	2B 02	INC	
E0D5·	007B·	INC	
E0D6·	3E 30	INC	
E0D7·	007C·	INC	
E0D8·	2B 03	INC	
E0D9·	007D·	INC	
E0D0·	1B 0F	INC	
E0D1·	007E·	INC	
E0D2·	2B 01	INC	
E0D3·	007F·	INC	
E0D4·	2B 02	INC	
E0D5·	0080·	INC	
E0D6·	3E 30	INC	
E0D7·	0081·	INC	
E0D8·	2B 03	INC	
E0D9·	0082·	INC	
E0D0·	1B 0F	INC	
E0D1·	0083·	INC	
E0D2·	2B 01	INC	
E0D3·	0084·	INC	
E0D4·	2B 02	INC	
E0D5·	0085·	INC	
E0D6·	3E 30	INC	
E0D7·	0086·	INC	
E0D8·	2B 03	INC	
E0D9·	0087·	INC	
E0D0·	1B 0F	INC	
E0D1·	0088·	INC	
E0D2·	2B 01	INC	
E0D3·	0089·	INC	
E0D4·	2B 02	INC	
E0D5·	008A·	INC	
E0D6·	3E 30	INC	
E0D7·	008B·	INC	
E0D8·	2B 03	INC	
E0D9·	008C·	INC	
E0D0·	1B 0F	INC	
E0D1·	008D·	INC	
E0D2·	2B 01	INC	
E0D3·	008E·	INC	
E0D4·	2B 02	INC	
E0D5·	008F·	INC	
E0D6·	3E 30	INC	
E0D7·	0090·	INC	
E0D8·	2B 03	INC	
E0D9·	0091·	INC	
E0D0·	1B 0F	INC	
E0D1·	0092·	INC	
E0D2·	2B 01	INC	
E0D3·	0093·	INC	
E0D4·	2B 02	INC	
E0D5·	0094·	INC	
E0D6·	3E 30	INC	
E0D7·	0095·	INC	
E0D8·	2B 03	INC	
E0D9·	0096·	INC	
E0D0·	1B 0F	INC	
E0D1·	0097·	INC	
E0D2·	2B 01	INC	
E0D3·	0098·	INC	
E0D4·	2B 02	INC	
E0D5·	0099·	INC	
E0D6·	3E 30	INC	
E0D7·	009A·	INC	
E0D8·	2B 03	INC	
E0D9·	009B·	INC	
E0D0·	1B 0F	INC	
E0D1·	009C·	INC	
E0D2·	2B 01	INC	
E0D3·	009D·	INC	
E0D4·	2B 02	INC	
E0D5·	009E·	INC	
E0D6·	3E 30	INC	
E0D7·	009F·	INC	
E0D8·	2B 03	INC	
E0D9·	00A0·	INC	
E0D0·	1B 0F	INC	
E0D1·	00A1·	INC	
E0D2·	2B 01	INC	
E0D3·	00A2·	INC	
E0D4·	2B 02	INC	
E0D5·	00A3·	INC	
E0D6·	3E 30	INC	
E0D7·	00A4·	INC	
E0D8·	2B 03	INC	
E0D9·	00A5·	INC	
E0D0·	1B 0F	INC	
E0D1·	00A6·	INC	
E0D2·	2B 01	INC	
E0D3·	00A7·	INC	
E0D4·	2B 02	INC	
E0D5·	00A8·	INC	
E0D6·	3E 30	INC	
E0D7·	00A9·	INC	
E0D8·	2B 03	INC	
E0D9·	00AA·	INC	
E0D0·	1B 0F	INC	
E0D1·	00AB·	INC	
E0D2·	2B 01	INC	
E0D3·	00AC·	INC	
E0D4·	2B 02	INC	
E0D5·	00AD·	INC	
E0D6·	3E 30	INC	
E0D7·	00AE·	INC	
E0D8·	2B 03	INC	
E0D9·	00AF·	INC	
E0D0·	1B 0F	INC	
E0D1·	00B0·	INC	
E0D2·	2B 01	INC	
E0D3·	00B1·	INC	
E0D4·	2B 02	INC	
E0D5·	00B2·	INC	
E0D6·	3E 30	INC	
E0D7·	00B3·	INC	
E0D8·	2B 03	INC	
E0D9·	00B4·	INC	
E0D0·	1B 0F	INC	
E0D1·	00B5·	INC</td	

BDS5	7E		END7:	LD	A . (HL)
BDS6	B7			OR	A
BDS7	28 03			JR	Z . END8
BDS8	23			INC	HL
BDS9	10 F9			DUNZ	END7
BDS10	78		END8:	LD	A . B
BDS11	F1 0C			CP	12
BDS12	30 13			JR	NC . ZERC
BDS13	04			INC	B
BDS14	2B			DEC	HL
BDS15	7E			LD	A . (HL)
BDS16	FE 2E			CP	*
BDS17	2B 08			JR	Z . END10
BDS18	FE 3D			CP	* 0
BDS19	2B 05		END9:	LD	(HL) . 0
BDS20	36 00			JR	END8
BDS21	18 EC		END10:	LD	(HL) . 0
BDS22	3B 00			POP	AF
BDS23	F1 C9		END11:	RET	

BDA4			ZERO:	POP	AF
BDA4	F1			JP	CLEARF
BDA5	C3 BBB0				

BDA8			PRGEND:		
				----- WORK AREA -----	
BDA8				CBP:	DS
BDA9				GMP:	DS
BDA10				EE:	DS
BDA11				KAI:	DS
BDA12				ACC1:	DS
BDA13				ACC2:	DS
BDA14				ACC3:	DS
BDA15				CHM:	DS
BDA16				PCNT:	DS
				END	

Macros:

BGA4	.7BRR	BDA0	ACCI	BDB2	ACC2
BDF7	ACC3	40B1	ATBC	5A9F	ADBNUM
BFI6	ALPLAG	40B0	ASHT	43C1	BUFL
BFI7	BUFFL	B5A6	CALF1	B9C0	CALF2
BFI8	CALJ3	B9AE	CALFUN	B975	CALTAN
BFI9	CRP	4041	CHMRA	BF75	CHRMUF
BFI10	CLR0MF	BAD0	CLRK	BDA1	CMD
BFI11	CNVB3	BA15	CNVB2	BAA0	CNVB3
BFI12	CNVB4	BA49	CNVB5	BAA4	CNVB6
BFI13	CNVB7	BA71	CNVB7	B9F6	CNVB1N
BFI14	COMAK	BAFF	COMK1	5A31	CONVT1
BFI15	CORRECT	BBEA	CORECT1	B0C0	CORECT2
BFI16	DEKEY	BFE4	DBUFF	BF1E4	DBUFF0
BFI17	DBFFF1	BFF2	DBUFF2	BF1F9	DBUFF3
BFI18	D1510E	411C	D15P	419E	DISPCL
BFI19	D1510E	B5E9	DISPAT	B0AA	EE
BFI20	END0	B5E6	END1	B0A0	END10
BFI21	END11	B5E7	END2	B067	END3
BFI22	END4	B074	END5	B07E	END6
BFI23	END7	BDC1	END8	B09C	END9
BFI24	ENDCHCT	B82C	ENTHY	B0B0	EMR
BFI25	ENDCHSP	42BD	ENTHR	B567	EXS1ST
BFI26	FEDAD	H4C4	F1J1	BFAF	FLNGT
BFI27	FPACT	BFB0	FRXT	BFAB	FTPAD
BFI28	FUKURJ	B8A3	GAN1	B8BB	GAN2
BFI29	GAN3	B8A0	GANKIN	4990	GSTKEY
BFI30	GOKE1	421A	HY00	4221	HY01
BFI31	HY0123	4228	HY02	410C	HY023
BFI32	HY03	410D	HY03G	B0B0	ICHI
BFI33	INKEY	B429	INPUT1	BAB7	INPUT
BFI34	INT	BFB7	INTBF	BFB9	INTFL
BFI35	INTSET	5F66	INVIRT	B669	K1
BFA0	K2	BDA0	KAI	B906	KAS1
BFI37	KAIISU	4112	KALDIS	413B	KDISPI
BFI38	KD1SP2	4169	KD1SP3	BF85	KEYBU

BAAS'	KEYDSP	4352	KEYIN	BA7B'	KEYW
BABB'	KEYWI	41F3	KHYP1	41F7'	KHYP2
4115	KHYP23	41FB	KHYP3	B82F'	LOOP
BCC0'	LOOPC10	BC77'	LOOPC10	BC6A'	LOOPC1
BC93'	LOOPC12	BC7C	LOOPC13	BC6C'	LOOPC4
BCDD'	LOOPC15	BC7F'	LOOPC15	BD01'	LOOPC7
BD16'	LOOPC18	BB0F'	LOOPC18	BC24'	LOOPNE1
40C6	LORE	0008	M	B862'	M1
B8EC'	M2	B929	M3	B970'	M4
B8C9'	M5	B91E	M6	B815'	MAIN
B834'	ME1	B84A	MENU1	BC62'	M1NSHT
BC54'	M1NSHT1	572B	MULTPL	42AE'	M1S1C
571A'	NE1	B818	NE1P0	BC63'	NE1NSHT
BC12'	NE1T01	BC7F'	NE1T01	BCAE'	NE1T01
BC11'	NE1T02	BCCB	NE1T03	BCEB'	NE1T05
BD20'	NE1T07	BB0B	NOEX1ST	BCFF'	NONINS
B803'	NUM1	BB07'	NUM2	B809'	NUM3
BB1D'	NUM4	BB22'	NUM5	003A'	GLUTNUM
BCDC'	PCTN	00E0	PEND	0060'	PINKEY
0040'	PKEY1N	0050	PMUS1C	BC87'	POINTSET
BC81'	POINTSET1	BC51	PROSH1	BDA8'	PRGEND
BA0A'	PRGEND	BA10	PRGEND	4391'	PRGEND
5F70'	RANDOM	BC8E'	RESET1	B849'	R1R1
B856'	R1R12	B92E	R1R1TU	4310'	RSIN
4320	RSOUT	42E5	RSVER1	42E0'	RSVERD
BFB2	SAHL	40B6	SARE	B45C'	SETPRA
BBAC'	SOUND	41C8	STBY	0010'	STW
58A8	SUBLNUM	8757	TANR1	B8A5'	TRF
BBAC'	TRFS	BB43	TRFSX4	B8B5'	TRFSXA
BB89'	TRFA1A	B84E	TRFA1B	B87E'	TRFA2A
BB83'	TRFA3A	B852	TRFA3B	B86C'	TRFAA1
BB65'	TRFAA2	BB6A	TRFAA3	B888'	TRFAC
BB6F'	TRFA1	BB74'	TRFRP	B893'	TRFCD
BB44'	TRF1R	BB49'	TRFRP	BB56'	TRFRR
5FE8	TIMER	BBFA	VRAM	0030'	WA1
0031	WA2	0032	WA3	0033'	WA4
0038	WA5	0035	WA6	0036'	WA7
4247	WH10	4280	WH101	425A'	WH11
4263	WH12	4281	WH123	4284'	WH13
426A	WH1AL	409A	Z80END	BD44'	ZERO

日数計算

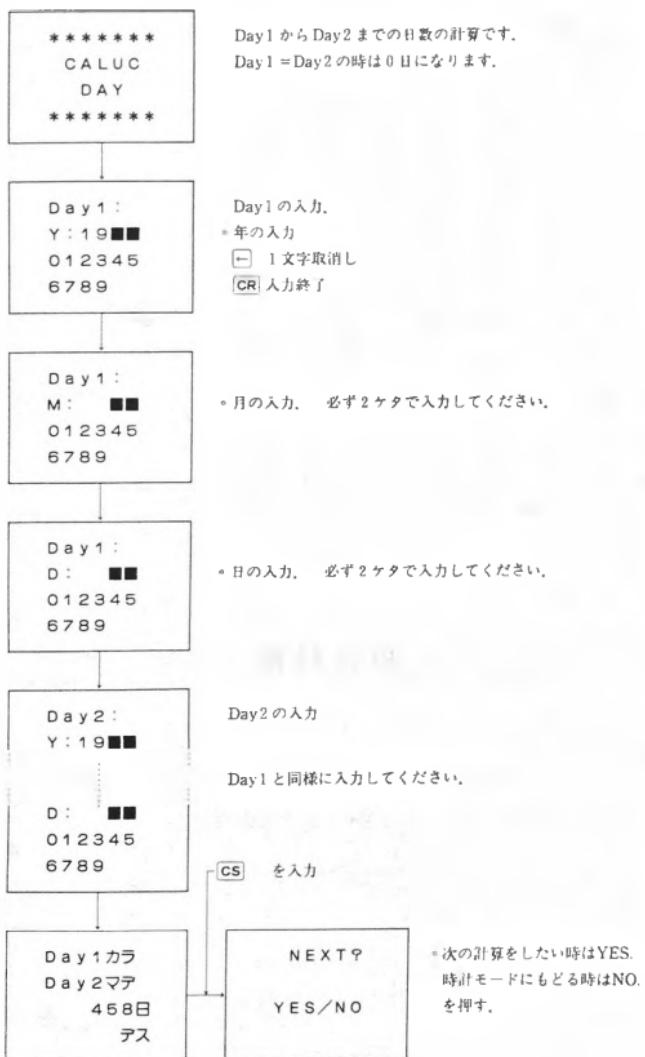
ある日からある日までの日数を計算します。同じ日を入力した時を0日として計算しています。
両方の日付は1900年から1999年までに入っています。

最初は年の入力です。

下二段の[0]～[9]までのボタンで入力してください。誤って入力した場合は、[←]ボタンで1桁もどすことができます。2桁の入力が終ったら[CR]ボタンを押してください。

次に月の入力です。入力方法は年の入力と同じです。入力は必ず2桁となるようにしてください。
次は日の入力です。入力方法は年の入力と同じです。入力は必ず2桁となるようにしてください。
以上で一方の日付の入力は終ります。

もう一方の日付の入力を同じように入力します。そうすると、最初に入れた日から次に入力した日までの日数を表示します。[CR]を入力すると、次の計算をするか聞いてきますので、計算したい時は、[YES]を押します。[NO]を押すと時計モードにもどります。



プロゲラムリスト

サンプリスト

BAB0 01 28 02 06 00 78 B7 C1 C9 C5 47 E6 F0 0F 4F 0F :39
BAC0 0F B1 4F 78 E6 0F B1 C1 C9 21 D0 BB 06 03 37 TE :C1
BAD0 CE 00 27 77 2B 10 F8 C9 11 D0 BB 23 IA 86 27 12 :00
BAE0 2B 1A 8E 27 12 1B 1A CE 00 27 12 C9 11 D0 BB :C8
BAF0 23 1A 98 27 12 1B 2B IA 9E 27 12 1B IA DE 00 27 :D0

Sum B3 BF F7 89 F2 BE C8 29 6A 53 95 EB 6C 3D 6F C6 :AE

Add +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum

ASCII

```

BB60 12 C9 3E 06 32 D1 BB CD 0D BB CD 0D BB 7E 23 CD :75
BB10 B1 40 CD 16 BB 41 3A D1 BB 3D 28 11 32 D1 BB 78 12
BB20 FE 30 28 07 3E D1 32 BB 1B 02 06 20 7B 12 13 :37
BB30 C5 3E 02 1E 0C C3 1C 41 06 07 11 BA BB 7F CD 41 :64
BB40 40 13 23 18 F8 C9 00 00 00 31 00 59 00 90 D1 20 :82
BB50 01 51 01 81 02 12 02 43 02 73 03 04 03 34 03 65 :48
BB60 31 29 31 30 31 30 31 31 30 31 30 31 00 FE 92 92 :62
BB70 FE 44 61 79 31 3A 44 61 79 32 3A 30 31 32 33 34 :08
BB80 35 36 37 38 39 20 20 4E 45 58 54 3F 59 45 53 2F :F1
BB90 4E 4F 44 61 79 31 86 D7 20 44 61 79 32 CF CE DE :59
BBA0 20 20 20 20 20 20 20 20 20 20 C3 DE BD 59 3A 51
BBB0 31 39 20 20 4D 3A 20 20 20 20 20 44 3A 20 20 AF :AF
BBC0 20 20 00 00 D4 74 1A 21 37 3D 34 34 21 ED 2F CD :C9
BBD0 9D 1A 00 81 02 12 02 43 02 73 03 04 03 34 03 65 :EB
BBE0 31 29 31 30 31 30 31 30 31 30 31 00 FE 92 92 :62
BBF0 FE 44 61 79 31 3A 44 61 79 32 3A 30 31 32 33 34 :08

```

Sum C6 CD 58 7E B1 C9 7B E0 BB 0D 0B D0 D2 95 0C 43 :C7

ソースリスト

```

***** CALLUCRATE OF DAY ***** by K Shizuzu ****
***** 9/4 1984 *****

ORG 0B815H
;
; *** SYSTEM SUBROUTINE ***
;
STBY EQU 41CBH : Z-80 Standby & Key Restart Set
INT EQU 41A2H : Interrupt Code Set
ZB0END EQU 409AH : Z-80 Control End
LORE EQU 40C6H : Load All Register From Stack
SARE EQU 4086H : Save All Register Into Stack
ERROR EQU 42D8H : Display Of Error Code
;
INKEY EQU 43A0H : Inkey (Scan & Key Data Input)
PUTKEY EQU 4391H : Put Key
GETKEY EQU 4090H : Get Key
KEYIN EQU 4175H : Key In & Load Of Input Key Data
DESI EQU 40A1H : Decode Of Key Data
KD1SP1 EQU 413BH : Key Display1
KD1SP2 EQU 4152H : Key Display2
KD1SP3 EQU 4169H : Key Display3
KALD15 EQU 4112H : Key Of All Display (I-3 Page)
KHYP1 EQU 4115H : Key Of 2-3 Page Display
KHYP2 EQU 41F7H : Key Of 1 Page Display
KHYP3 EQU 41FBH : Key Of 2 Page Display
;
MUSIC EQU 42AEH : Music
TIMER EQU 5FEBH : Soft Timer (CS ADD --> 411BH)
REOUT EQU 4220H : RS-232C Output
RSIN EQU 4310H : RS-232C Input
RSVERD EQU 42E0H : Verify Of RS-232C Input Data(1)
RSVER1 EQU 42E5H : Verify Of RS-232C Input Data(2)
;
HY0G1 EQU 4106H : Display Of All Page
HY0123 EQU 4109H : Display Of 1-3 Page
HY023 EQU 410CH : Display Of 2-3 Page
HY010 EQU 421AH : Display Of 0 Page
HY011 EQU 4224H : Display Of 0 Page
HY02 EQU 423BH : Display Of 2 Page
HY03 EQU 425FH : Display Of 3 Page
WH1AL EQU 4256H : Display Clear Of All Page
WH101 EQU 4257H : Display Clear Of 0-1 Page
WH123 EQU 4281H : Display Clear Of 2-3 Page
WH10 EQU 4247H : Display Clear Of 0 Page
WH11 EQU 425AH : Display Clear Of 1 Page
WH12 EQU 4263H : Display Clear Of 2 Page
WH13 EQU 4264H : Display Clear Of 3 Page
ASCII EQU 4000H : ASCII Convert (0-8)
AC1BC EQU 401RH : ASCII Convert By BCD
BURCL EQU 43C1H : Character Buffer Clear
DISPCL EQU 4386H : Display Buffer Clear
CHARA EQU 4641H : Character Of Display
DISP EQU 411CH : Transfer Dot RAM (--> V RAM)
;
CONVT1 EQU 5A31H : Number Convert (ASCII-->HEX)
CONVT2 EQU 5B27H : Character Convert (ASCII-->HEX)
ADMINH EQU 5A89H : FPACC-->ABCDE+FPACC
SUBRIM EQU 5A8AH : FPACC-->ABCDE-FPACC
WLTTP EQU 572BH : FPACC-->ABCDE*FPACC
DIVIDE EQU 5756H : FPACC-->ABCDE/FPACC
NEGSIN EQU 5728H : FPACC-->-FPACC
INVERT EQU 5F66H : FPACC-->!/FPACC
RANDOM EQU 5F78H : FPACC-->Random Number
;
; *** SYSTEM WORK AREA ***
;
FTPID EQU 0BFA8H : File Data Top Address
FEDAD EQU 0BFADH : File Data End Address
FLNGT EQU 0BFAFH : Length Of Record
FRDCT EQU 0BFB0H : Number Of Record
SP40H EQU 0BF52H : Start Area HL Reg.
INTSET EQU 0BF54H : Interrupt So Area
KEYBUV EQU 0BF55H : Key Data Buffer Reg.
SP45H EQU 0BF56H : Key Data Buffer Reg.
BPLAG EQU 0BF64H : Alarm Flag
INTBF EQU 0BF67H : 1/64Hz Jump Table
;
```


B88F'	FE #2		CP	#2H	: Month 2 ?
B891'	2# BF		JR	NZ.DINDN	:
B893'	3A BBC4'		LD	A.(DAY1YE)	:
B896'	CD B89B'		CALL	URCK	:
B899'	2# 87		JR	NZ.DINDN	:
B89B'	DD 7E #8		LD	A.(IX)	:
B89E'	FE 29		CP	29H	:
B8A0'	2# D5		JR	Z.DINDD	:
B8A2'	21 B876'	DINDN:	LD	HL.MSGTB2	"Day2:"
B8A5'	CD BA16'		CALL	DDMOV	:
B8A8'	21 BBAE'		LD	HL.MSGTBY	:
B8A9'	CD BA21'		CALL	DDMOV	:
B8AE'	CD 21 BBC9'		LD	IX.DAY2YE	:
B8B2'	CD BA4D'	DINDM2:	CALL	INDLP1	:
B8B5'	21 BB85'		LD	HL.MSGTBM	:
B8B8'	CD BA21'		CALL	DDMOV	:
B8B9'	DD 21 BBCA'		LD	IX.DAY2MN	:
B8BF'	CD BA4D'	DINDM2:	CALL	INDLP1	:
B8C2'	DD 7E #8		LD	A.(IX)	:
B8C5'	FE 13		CP	13H	:
B8C7'	3# F6		JR	NC.DINDM2	:
B8C9'	FE #1		CP	01H	:
B8C8'	3# F2		JR	C.DINDD2	:
B8CD'	21 BBCB'		LD	HL.MSGTBD	:
B8D8'	CD BA21'		CALL	DDMOV	:
B8D3'	DD 21 BBCB'		LD	IX.DAY2DY	:
B8D7'	CD BA4D'	DINDD2:	CALL	INDLP1	:
B8D8'	DD 7E #8		LD	A.(IX)	:
B8DD'	FE #8		CP	88H	:
B8DF'	2# F6		JR	Z.DINDD2	:
B8E1'	3A BBCA'		LD	A.(DAY2MN)	
B8E4'	CD B89F'		CALL	GETMM	Max Day Data
B8E7'	DD BE #8		CP	(IX)	
B8EA'	3# EB		JR	C.DINDD2	
B8EC'	3A BBCA'		LD	A.(DAY2MN)	
B8EF'	FE #2		CP	82H	
B8F1'	2# BF		JR	NZ.DINDN2	
B8F3'	3A BBC9'		LD	A.(DAY2YE)	
B8F6'	CD B89B'		CALL	URCK	
B8F9'	2# 87		JR	NZ.DINDN2	
B8FB'	DD 7E #8		LD	A.(IX)	
B8FE'	FE 29		CP	29H	
B900'	2# D5		JR	Z.DINDD2	
B902'	21 BBC4'	DINDN2:	LD	HL.DAY1YE	
B905'	11 BBC9'		LD	DE.DAY2YE	
B908'	8# 83		LD	DE	
B90A'	1A	LCHECK:	LD	A.(DE)	
B90B'	BE		CP	(HL)	
B90C'	3# 88		JR	C.SWAPOK	
B90E'	2# 11		JR	NZ.NOSWAP	
B910'	13		INC	DE	
B911'	23		INC	HL	
B912'	1# F6		DJNZ	LCHECK	
B914'	1# 8B		JR	NOSWAP	
B916'	21 BBC4'	SWAPOK:	LD	HL.DAY1YE	
B919'	11 BBC9'		LD	DE.DAY2YE	
B91C'	8# 83		LD	B.3	
B91E'	CD B8E9'		CALL	SWAP	
B921'	DD 21 BBC4'	NOSWAP:	LD	IX.DAY1YE	
B925'	CD B8C3'		CALL	STOTAL	DAY1 Total
B928'	DD 21 BBC9'		LD	IX.DAY2YE	
B92C'	CD BBC3'		CALL	STOTAL	DAY2 Total
B92F'	21 BBC4'		YADDLP:	LD	
B932'	3A BBC9'			HL.DAY1YE	
B935'	BE		CP	A.(DAY2YE)	
B936'	2# 19		JR	(HL)	
				Z.NOYINC	DAY1YE = DAY2YE ?

B938	21 BB5E	LD CALL	HL_WDAYTA	:
B93B	CD BAD8		NXDADD	: NXDAY <- NXDAY + 365
B93E	3A BBC4	LD CALL	A.(DAYYE)	:
B941	CD BA98	CALL JR	UR_N	: URUDOSHI Check
B944	28 #3		Z.DIYINC	:
B946	CD BAC9	CALL	NXDINC	: NXDAY Increment
B949	21 BBC4	DIYINC LD	HL_DAYYE	: DAYYE increment
B94C	CD BBF3	CALL JR	BCDINC	:
B94F	18 DE		YADDLP	:
B951	21 BBCC	NOYINC LD	HL_DAYS2	:
B954	CD BAD8	CALL	NXDADD	:
B957	21 BBC7	LD CALL	HL_DAYS1	:
B95A	CD BAED		NXDISUB	:
B95D	21 BB92	LD	HL_MSGTB6	:
B960	11 BFE4	LD	DE_DBUFF	:
B963	B1 #E1C	LD	BC_28	:
B966	ED B8	LDIR		:
B968	CD 41B6	CALL	HYOGI	:
B96B	21 BBCE	LD	HL_NXDAY	:
B96E	11 BFF2	LD	DE_DBUFF2	:
B971	CD BB92	CALL	DDISP	:
B974	CD 40B6	CALL	DISPL	:
B977	21 BFF2	LD	HL_DBUFF2	:
B97A	CD BB92	CALL	CHARAS	: Graphic Display
B97D	21 BBC7	LD	HL_DADAY	:
B980	11 BFD6	LD	DE_VRAM+36	:
B983	B1 #E85	LD	BC_5	:
B986	ED B8	LDIR		: Character DAY
B988	CD BB91	CALL	DISP2	:
B98B	CD BB2C	CALL	MAIT	: Wait 2 Second
B98E	CD 4391			:
B991	B7	NEXTIN: CALL OR	PUTKEY A	:
B992	28 FA		JR Z.NEXTIN	:
B994	CD 43C1	CALL BUFL		:
B997	CD B877	LD HL_MSGTB4		"NEXT?"
B99A	11 BFF5	LD DE_DBUFF+1		:
B99D	B1 #E85	LD BC_5		:
B9A0	ED B8	LDIR		:
B9A2	CD 41B6	CALL	HYOGI	:
B9A5	CD 4228	NEXTGM: CALL	HYO2	:
B9A8	21 BBC7	LD HL_MSGTB5		"YES/NO"
B9AB	CD 4377	CALL KHYO2		:
B9AE	CD 4391	CALL PUTKEY		:
B9B1	FE 11	CP 11H		:
B9B3	38 F8	JR C.NEXTGM		:
B9B5	FE 14	CP 14H		:
B9B7	DA B82C	JP C.MAIN		:
B9B8	28 E9	JR Z.NEXTGM		:
B9B9	FE 17	CP 17H		:
B9B6	38 E5	JR NC.NEXTGM		:
B9C0	C3 489A	JP Z#END		: Z-B8 Control End
B9C3	DD 7E #1			:
B9C6	CD BAE6	STOTAL: LD CALL	A.(IX+1)	: Get Month Data
B9C9	DD 7E #8		GETMD	:
B9CC	CD BAB8	LD CALL	A.IIX1	: Get Year Data
B9CF	28 #8		URCK JR	:
B9D1	DD 7E #1		Z.STOTAN	:
B9D4	FE #3	LD	A.(IX+1)	: Get Month Data
B9D6	38 #1	CP #3H		:
B9D8	BC	JR C.STOTAN		:
B9D9	DD 7E #2	INC	C	:
B9D9	DD 7E #2	STOTAN: LD	A.(IX+2)	: Get Day Data


```

B9C- 81 ADD A,C
B9D- 27 ADD (1X4) .A
B9E- DD 77 64 LD A,B
B9F- 78 LD A,B
B9G- CE 88 ADC A,B
B9H- 27 DAA
B9I- DD 77 63 LD (IX+3) .A
B9J- C9 RET BCD Data

; SWAP:
B9K- 1A LD A,(DE)
B9L- 4E LD C,(HL)
B9M- 77 LD (HL),A
B9N- 79 LD A,C
B9O- 12 LD I(DEF),A
B9P- 13 INC HL
B9Q- 23 LD DE,HL
B9R- 1B F7 INC SWAP
B9S- C9 RET

; BCDINC:
B9T- 7E LD A,(HL)
B9U- C6 #1 ADD A,I
B9V- 27 LD DATA BCD Data
B9W- 77 LD (HL),A
B9X- C9 RET

; GETMAX:
B9Y- 11 B96B- LD DE,MDATA Max Day Data Of Month
B9Z- CD BAB9- CALL BCDEX Convert From BCD To HEX
B9F- 3D DEC A
B9G- 6F LD L,A
B9H- 88 LD H,B
B9I- 26 ADD HL,DE
B9J- 19 ADD A,(HL)
B9K- 7E LD DE <-- A#2
B9L- C9 RET Get Max Data

; GETMDT:
B9M- CD BAB9- CALL BCDEX Convert From BCD To HEX
B9N- 3D DEC A
B9O- 21 B946- LD HL,MDATA Month Data Table
B9P- 16 88 LD D-B
B9Q- 87 ADD A,A
B9R- 5F ADD E,A
B9S- 19 ADD HL,DE
B9T- 46 ADD B,(HL)
B9U- 23 ADD INC HL
B9V- 4E LD C,(HL)
B9W- C9 RET B: High BCD Data
C: Low BCD Data

```


BAC8'	C9		RET	
BAC9'	21 BBD8	NXDINC:	LD HL,NXDAY+2 B,3	: NXDAY increment
BACC'	#6 #8		SCF	
BACE'	37	NXDINL:	LD A,(HL)	
BACF'	7E		ADC A,B	
BAD0'	CE #8		DAA	
BAD2'	77		LD (HL),A	
BAD3'	77		DEC HL	
BAD4'	28		DJN2 NXDINL	
BAD5'	18 F8		RET	
BAD7'	C9			
BAD8'	11 BBD#	NXDADD:	LD DE,NXDAY+2 INC HL	
BADB'	23		LD A,(DE)	
BADC'	1A		ADD A,(HL)	
BADD'	86		DAA	: BCD Data
BADE'	27		LD (DE),A	
BADF'	2B		DEC HL	
BAE0'	12		DEC DE	
BAE1'	2B		DEC DE	
BAE2'	1B			
BAE3'	1A		LD A,(DE)	
BAE4'	8E		ADC A,(HL)	
BAE5'	27		DAA	: BCD Data
BAE6'	12		LD (DE),A	
BAE7'	1B		DEC DE	
BAE8'	CD #8		LD A,(DE)	
BAE9'	27		ADC A,B	
BAE10'	12		DAA	: BCD Data
BAE11'	C9		LD (DE),A	
BAE12'			RET	
BAED'	11 BBD#	NXDSUB:	LD DE,NXDAY+2 INC HL	
BAF0'	23		LD A,(DE)	
BAF1'	1A		SUB (HL)	
BAF2'	96		DAA	: BCD Data
BAF3'	27		LD (DE),A	
BAF4'	12		DEC DE	
BAF5'	1B		DEC HL	
BAF6'	2B			
BAF7'	1A		LD A,(DE)	
BAF8'	9E		SBC A,(HL)	
BAF9'	27		DAA	: BCD Data
BAFA'	12		LD (DE),A	
BAFB'	1B		DEC DE	
BAFC'	1A		LD A,(DE)	
BAFD'	DE #8		SBC A,B	
BAFF'	27		DAA	: BCD Data
BB#0'	12		LD (DE),A	
BB#1'	C9		RET	
BB#2'	3E #6	DBDISP1:	LD A,6 (ZEROSP).A	: 6 Digit Zero Suplies Flag Clear
BB#4'	32 BBD1'		LD	
BB#7'	CD BBD#	CALL	DSCORZ	
BB#8'	CD BBD#	CALL	DSCORZ	
BB#D'	7E	DSCORZ:	LD A,(HL)	: BCD Data
BB#E'	23		INC HL	
BB#F'	CD 48#1		CALL ACTBC	: Decord B,C--A
BB#12'	CD BB#16'		CALL DSCORL	
BB#15'	41		LD B,C	
BB#16'	3A BBD1'	DSCORL:	LD A,(ZEROSP)	
BB#19'	3D		DEC A	: A*: Display
BB#1A'	28 11		JR Z,DSCOZ1	
BB#1B'	32 BBD1'		LD (ZEROSP).A	
BB#1F'	78		A,B	: B: ASCII Character
BB#20'	FE 28		CP *1	Character "g"?
BB#22'	28 87		JR Z,DSCOZ#	
BB#24'	3E #1		LD A,1	
BB#26'	32 BBD1'		LD (ZEROSP).A	: Set Flag (Display On)
BB#29'	18 #2		JR DSCOZ1	
BB#B'	#6 28	DSCOZ#:	LD B,2BH	: Space Character
BB#D'	78		A,B	
BB#E'	12		LD (DE).A	: DE: DBUFF Address

BB2F	I3		INC	DE	:
BB30	C9		RET		
		:			
BB31	3E 82	DISP2:	LD	A.2	: Page 2 Graphic Display
BB32	1E B8		LD	E.8	
BB33	C3 411C		JP	DISP	
		:			
BB34	66 87	CHARAS:	LD	B.7	: Convert Graphic <-- Character
BB35	11 BFBA		LD	DE, VRAM	
BB36	7E	CHARAL:	LD	A. (HL)	
BB37	CD 4B41		CALL	CHARA	: Set Character Pattern
BB41	I3		INC	DE	
BB42	23		INC	HL	
BB43	18 F8		DJNZ	CHARAL	
BB45	C9		RET		
		:			
		:			
BB46	BB BB BB 31	MIDATA:	DB	BBH.BBH.BBH.31H.BBH.59H.BBH.9BH	
BB4A	BB 59 BB 9B		DB	B1H.B2H.B1H.51H.B1H.B1H.02H.12H	
BB4E	B1 B2 B1 51		DB		
BB52	B1 B2 B2 12		DB	B2H.43H.B2H.73H.B3H.B4H.B3H.34H	
BB56	B2 43 B2 73		DB		
BB5A	B3 B4 B3 34		DB		
		:			
BB5E	B3 65	MDATA:	DB	B3H.65H	
		:			
BB5F	31 29 31 38	MDATA:	DB	31H.29H.31H.38H.31H.38H	
BB64	31 38		DB		
BB66	31 31 38 31		DB	31H.31H.38H.31H.38H.31H	
BB6A	38 31		DB		
		:			
BB6C	BB FE 92 92	DADAY:	DB	BBH.0FEH.92H.92H.#FED	
BB70	FE		DB		
		:			
BB71	44 61 79 31	MSGTB1:	DB	"Day1:"	
BB75	3A	MSGTB2:	DB	"Day2:"	
BB76	44 61 79 32		DB		
BB77	3A	MSGTB3:	DB	"#123456789 "	
BB78	38 31 32 33		DB		
BB7F	34 35 36 37	MSGTB4:	DB	"NEXT?"	
BB83	38 39 28 28		DB		
BB87	4E 45 58 54	MSGTB5:	DB	"YES/NO"	
BB88	3F		DB		
BB8C	59 45 53 2F	MSGTB6:	DB	"Day1", #B6H.#D7H," Day2", #C3H.#DEH	
BB89	4E 4F		DB		
		:			
BB92	44 61 79 31	MSGTB7:	DB	"Day1", #B6H.#D7H," Day2", #C3H.#DEH	
BB96	B6 D7 28 44		DB		
BB9A	61 79 32 CF		DB		
BB9E	C3 DE		DB		
BBAA	28 28 28 28		DB	*	" , #C3H.#DEH. #BDH
BB44	28 28 28 28		DB		
BB48	28 28 28 C3		DB		
BBC4	DE BD		DB		
		:			
BBAE	59 3A 31 39	MSGTBY:	DB	"Y:19 "	
BBB2	28 28 28	MSGTBM:	DB	"M: "	
BBB5	4D 3A 28 28	MSGTBD:	DB	"D: "	
BBB9	28 28 28		DB		
BBBC	44 3A 28 28		DB		
BBC8	28 28 28		DB		
		:			
		:			
BBC3	BB	PRGEND:	DB	B	
		:			
		***** WORK AREA *****			
		:			
BBC4		DAY1YE:	DS	1	: Day1 Year Buffer
BBC5		DAY1MN:	DS	1	: Day1 Month Buffer
BBC6		DAY1DY:	DS	1	: Day1 Day Buffer
BBC7		DAY51:	DS	2	: Total Days 2
		:			
BBC9		DAY2YE:	DS	1	: Day2 Year Buffer
BBCA		DAY2MN:	DS	1	: Day2 Month Buffer
BBCB		DAY2DY:	DS	1	: Day2 Day Buffer
BBCC		DAYS2:	DS	2	: Total Days 2
		:			
BBCD		NXDAY:	DS	3	: Answer Day Number
		:			

```

BB01'           ZEROSP: DS      I : Zero Suples Flag
;
;
END

```

Macros:

ACINC	4881'	ADNUM	589F'	ALFLAG	BFB6	ASCII	4B08
BCDINEX	RAB05'	BCDINC	89F3'	BUFLCL	43C1	CHARA	4B41
CHANAL	BB3D'	CHARAS	BB28'	CONVT1	5A31	CONVT2	5B27
DIYINC	BB49'	DADAY	BB6C'	DAY1DY	BB6C'	DAY1MN	BB5C'
DAY1YE	BBC4'	DAY2DY	BBCB'	DAY2WN	BBCA'	DAY2YE	BBC9'
DAY51	BBC7'	DAY52	BBCF'	DBUFFT	BFE4	DBUF#B	BFE4
DBUFPI	BFFB	DBUFF2	BFF2	DBUFFF3	BFF9	DD1SP	BB#2'
DD2SP	BFFC	DBUFF4	BFF4	DD2SPD	BFFA'	DD2SPD2	BB#2'
DDNDM	BB5F'	DINMD2	BB8F'	DINND	BB8A'	DINND2	BB#2'
DINDY	BB52'	DINDY2	BBH2'	DISP	41C1	DISP2	BB31'
DISPCL	438E	DIVID1	57C6	DAMCV	B421'	DSCORL	BB16
DSCORZ	BB#D'	DSCOZ1	BB2B	DSCOZ1	BB2D'	ERROR	428D
FEDAD	BFA0	FLNGT	BFAF	FRDCT	BFB6	FTPAD	BFAB
GETKEY	4898	GETMD	BAB6	GETM#	BB9F'	HY08	421A
HY01	4221	HYO123	4189	HYO2	4228	HYO23	418C
HYD0	422F	HYOCD	41B6	INDLP1	B44D	INDLP2	BB48
INOLP3	BB#3'	INOLP4	BB#4P	INPS	BB45	INPS	BB43
INPUT	BB36	INT	41A2	INTBF	BFB7	INTFL	BBB8
INTSET	BBF4	INVERT	5F66	KALD15	4112	KDISP1	413B
KD1SP2	4152	KD1SP3	4169	KEYB#	BBF5	KEYIN	4352
KHYP1	41F3	KHYP2	41F7	KHYP23	4115	KHYP3	41FB
LCHECK	BB9A'	LORE	4#C6	MAIN	B816'	MAIN1	BB2C'
MDATA	BB66'	MIDATA	BB46'	MSCTB1	BB77'	MSCTB2	BB76'
MSCTB3	BB78'	MSCTB4	BB77'	MSCTB5	BB80'	MSCTB6	BB72'
MSCTBD	BB89'	MSCTBM	BB89'	MSCTAE	BB78	MULTFL	BB78
MUSIC	42AE	MYDATA	BB5E	NEGSIN	5728	NEXTGM	BB45'
NEXTIN	BB8E	NGSWAP	BB2I	NOYINC	BB51'	NXDAODD	BA0D'
NXDAY	BBC5'	NXDINC	BAC9'	NXDINL	BACF'	NXDSIB	BA0D'
PEND	BBEB	PINKER	BB68	PKEYIN	BB48	PMUSIC	BB58
PRGEND	BBC3'	PUTKEY	4391	RANDOM	5F78	RSIN	4310
RSOUT	432E	RSVER1	42E9	RSVERD	42E8	SALH	BBF2
SARF	41B6	STBL	BB43	SBALH	BB43	STOTAN	BB09'
STW	BB10	SUBNM	58A8	SWAP	B8E9	SWAPOK	B316
TIMER	5FF8	URCK	BB9B	URYE	BAB5'	VRAM	BBFA
VAI	BB32	WA2	BB31	WA3	BB32	VA4	BB33
WAS	BB34	WA6	BB35	WA7	BB36	WAIT	BA2C'
WH18	4247	WH181	4257	WH11	425A	WH12	4263
WH123	4281	WH13	4284	WH1AL	426A	WKCLR	BB38'
YADDLP	BB92F'	ZBBEND	4B9A	ZEROSP	BB01'		

円・ドル換算

円からドル、またはドルから円への換算を行うプログラムです。

まず、レートを入力します。メニュー画面の [Rate] を押します。1ドルは何円かと聞いてきますので、画面の下半分に表示されている数字のキーを押してその日のレートを入力します。間違えたときは、[C] を押します。最後に [CR] キーを押して終ります。なお、このレートは次に入力されるまで保存されます。

次に、円からドルへの換算かドルから円への換算かを、メニュー画面で [¥→\$] または [\$→¥] を押して選びます。すると、金額を聞いてきますので、換算する金額を入力します。レートの場合と同様に入力してください。[CR] キーを押すと、画面の下半分に換算された金額が表示されます。ここで、[CR] キーを押すと次の計算へ進みます。また、[MENU] キーでメニュー画面へ、[MODE] キーで時計モードにもどります。

なお、レートがゼロの状態で計算したり金額が大きすぎると、エラーが表示されますのでやり直してください。

1

= M E N U =
· R a t e
· ¥ --> \$
· \$ --> ¥

メニュー画面。

まず、レートを入力する。

'Rate' を押す。

2

1 \$ = ¥ ?
¥ 215
5 6 7 8 9 C
0 1 2 3 4 .

下半分に表示されているキーを

押してレートを入力し 'CR' を押す。

'C' はクリアキー。

次に入力されるまでこのレートは保存される。

3

= M E N U =
· R a t e
· ¥ --> \$
· \$ --> ¥

メニュー画面にもどる。

'¥→\$' または '\$→¥' を選ぶ。

4

¥ ?
5 6 7 8 9 C
0 1 2 3 4 .

'¥→\$' を押した場合

5

*1000

金額を入力し「CR」キーを押す

56789c

01234

6

¥1000

画面の下にドルに換算された金額が表示される。

4 65116

2 \$

「MENU」キーをタップして画面へ

「MODE」キーで時計を一回合

注) レートが0の状態で計算したり、金額が大きすぎるとERRORが表示される。
なお、以上の手順はS→Y計算の場合も同じである。

プログラムリスト

ダンブリスト

B9E0 3A 75 BC B7 28 0A 3E 30 32 75 BF 3E 01 32 75 BC :C2 :シテ .>02Uソ>.2ウ
B9F0 3E 2E 18 06 C6 24 18 02 C6 0F 57 3A 75 BC B9 38 :0E >...ニ.ニ.W.ウシテ

Sum B4 29 48 98 F5 04 16 DB 00 98 92 4B 38 A7 86 8F :00

Add +8 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum

ASCII

BA00 AB 5F B7 28 09 CD 89 BA 7A FE 38 28 06 7B 3C 32 :B6 イ.キ .ルコZ 0.((2
BA10 75 BC 7A 16 08 21 75 BF 19 77 23 AF 77 C9 21 7D :56 ルZ..ルW.ウシテ!
BA20 BC 5E 28 56 2B 4E 28 46 2B 7E C9 CD 81 42 21 46 :EE シ+V+N+F+~/*~B!F
BA30 BA CD FB 41 CD 91 43 FE 34 28 05 FE 36 20 F5 E1 :ED コハ A+C C 4(. 6.キ
BA40 CD CC B9 93 97 89 45 52 52 4F 52 21 21 70 BF 11 :71 ハヘク! ERROR!! ピ.
BA50 79 BC 18 26 21 7E BC 11 70 BF 18 1E 21 70 BF 11 :A5 ヨ.シ.&シ.シ.ソ...ソ.
BA60 7E BC 18 16 F5 CD 89 BA CD C7 5B FI C3 97 BA 21 :82 ヲ.シ.ヨウコスアドロ
BA70 75 BF 11 F2 BF 01 8C 88 18 03 81 05 00 ED 88 C9 :8A ュ.ソ.ソ.ソ.ソ.ソ.ソ.
BA80 3E 88 06 01 BE 50 ED 79 C9 21 75 BF 36 30 23 86 :36 ュ.ソ. POY!@J#6B#. -
BA90 0B 36 08 23 18 FB C9 21 75 BF 05 2E 7B 28 08 :CC .6. #. ルW.ウシテ!
BAAE FE 45 CA 34 BB 23 18 F4 AF 32 77 BC E1 06 08 :AC En4#(. ルZ.ウシテ.. -
BAB0 FE 2E 28 21 23 10 F8 TE 07 C9 FA FF 2E 28 18 :AE .(. !. - キバザ. .ル.
BAC0 36 38 23 E5 11 81 BF AF ED 52 E1 CA F0 BB 18 E7 :82 6B#. - ツコ#ORF!X#
BAD0 36 08 C3 FB BB 21 7C BF 7E FE 2E CA D0 EA 05 23 :06 6.テウ!11#. - ハミコ#
BAE0 7E B7 28 04 36 00 18 F7 E1 7E FE 39 C2 F0 BB 36 :DF "キ.6. .ドド" 9マケ6
BAF0 3E 2B E5 11 75 BF AF ED 52 E1 28 10 FE 2E 28 :SE 04.ル.ソ.ORF!. - .ル.

Sum 2B 84 3C 21 E0 B1 CA 0C 87 34 B7 CF CC 88 BC E6 :AA

Add +8 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum

ASCII

BB00 F0 3C 77 FE 3A C2 F8 BB 36 38 18 E5 7E FE 2D 28 :7C X<W :ツク#6B.▲ - (-
BB10 1E FE 2E 28 09 3C 77 FE 3A C2 F8 BB 36 38 01 08 :42 . .(.(.W :ツク#6B. -
BB20 00 21 7C B1 11 7D BF ED BB 23 36 31 C3 F9 BB 01 :47 . !.!)#(OK#61ルテ.
BB30 07 00 18 ED E1 0E 01 7E FE 45 28 18 23 0C 79 FE :A3 ..OF. ~ E1. 6. #. y
BB40 08 20 F4 E5 7E FE 45 28 05 36 00 23 18 FE 22 77 :EF . ■. E1. 6. #. y
BB50 BC E1 18 03 22 77 BC 2B 7E FE 39 C2 F0 BB 36 38 :C0 シ. ~.W#4. 9マク#6B
BB60 2B E5 11 75 BF AF ED 52 E1 28 10 FE 2E 07 FE 2B :F8 .(X
BB70 3C 77 FE 3A C2 F0 BB 36 38 18 E5 7E FE 2D 28 :85 <W :ツク#6B.▲ - (-
BB80 3C 77 FE 3A C2 F0 BB 36 31 18 03 23 36 31 21 80 :85 <W :ツク#61ルテ.
BB90 B7 2B 03 28 18 F9 F5 E5 28 26 FE 45 28 2A FE :33 ソ.キ. ~.E
BBAA 23 54 5D 7E FE 2D 20 24 E1 FI 3D 77 FE 30 08 :AD #T]Z - . \$H#W. 00.
BBBA 38 39 2B 7E FE 2D 20 F2 13 IA FE 30 C2 FB BB 62 :7F 69+~ - .#. ルW.ウシテ
BBC0 6B 23 7E 12 B7 CA FB BB 13 23 18 FE E1 FI 3C 77 :13 K#~.キバ. #. 分H#<W
BBBD FE 3A C2 F0 BB 36 38 2B 7E FE 45 20 FI 21 7F BF :67 :ツク#6B. ~ E1! リ
BBE0 11 88 BF TE FE 45 28 05 12 2B 1B 18 FE 3E 31 12 :25 .ソ.ソ. E1(+. .分)I
BBF0 3A 77 BC B7 28 46 2A 77 BC 23 7E FE 2D 28 06 21 :0A :Wシキ(FW#シ#) - (-!

Sum 48 8E 4C F6 AF 9D 55 A6 33 45 F3 3E 87 44 28 1E :19

Add +8 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum

ASCII

BC00 08 01 E3 18 37 23 7E D6 30 FE 03 38 64 FE 88 38 :AD ..4.7#~30 .8d .B
BC10 60 F5 CD 6F BA CD 89 BA 21 75 BF 36 38 23 36 2E :9D .(W#コ.コ!コ!コ!60#6. -
BC20 23 C1 05 36 38 23 10 FB 11 F2 BF IA B7 28 0D FE :43 #.6# .ル.キ. #.
BC30 28 28 81 77 13 23 04 07 FE 09 38 EF 21 75 BF 06 :89 .(.W. #. X. 8\!W.
BC40 07 E2 23 FE 2E 28 04 10 F8 18 24 21 75 BF 54 5D :4F .~#(. #. X. 0. .+
BC50 06 0C 7E B7 28 03 23 10 F9 78 FE 0C 30 13 04 2B :92 .~(. (. 0. .+ .●6.△
BC60 7E FE 2E 28 08 FE 30 20 06 36 00 18 EC 36 00 F1 :8F ノルコ.コ.ノルコ.ルテ...
BC70 C9 F1 C3 89 BA 4C CD DF 4B CD 1F 4C C3 00 00 :FE7#~30 .8d .B
BC80 08 00 08 18 37 23 7E D6 30 FE 03 38 64 FE 88 38 :C9 ..ル.ノルコ.コ!コ!コ!60#6. -
BC90 60 F5 CD 6F BA CD 89 BA 21 75 BF 36 38 23 36 2E :9D #.6# .ル.キ. #.
BCA0 23 C1 05 36 38 23 10 FB 11 F2 BF IA B7 28 0D FE :43 .(.W. #. X. 8\!W.
BCB0 2E 28 01 77 13 23 04 07 FE 09 38 EF 21 75 BF 06 :89 .~#(. (. .+ \$!W!T]
BCC0 0C 7E 23 FE 2E 28 04 10 F8 18 24 21 75 BF 54 5D :4F .~(. (. 0. .+ .●6.△
BCD0 06 0C 7E B7 28 03 23 10 F9 78 FE 0C 30 13 04 2B :92 ノルコ.コ.ノルコ.ルテ...
BCE0 7E FE 2E 28 08 FE 30 20 06 36 00 18 EC 36 00 F1 :8F7#~30 .8d .B
BCF0 C9 F1 C3 89 BA 4C CD DF 4B CD 1F 4C C3 00 00 :FE ノルコ.コ.ノルコ.ルテ...

Sum 14 AF AD 34 98 56 7E 44 44 02 F4 10 80 BC C4 B6 :24

相性診断

2人の誕生日から相性を出します。

最初は年の入力です。下二段の[0]～[9]のボタンを押して入力します。

2桁入力が終ったら[CR]を押してください。入力を誤った時は[←]ボタンを押すと1文字もどります。次に月の入力です。入力は必ず2桁で入力してください。入力方法は年の入力と同じです。

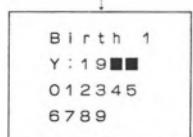
次は日の入力です。入力は必ず2桁で入力してください。入力方法は年の入力と同じです。これで1人目の生年月日が入力されます。

次にもう1人の生年月日の入力です。入力方法は1人目のときと同じく年、月、日を入力してください。入力が終るとハートが表示され、下からぬりつぶしていきます。右下に相性度が表示されます。最良が100です。

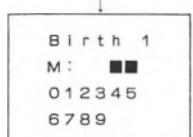
[CR]ボタンを押すと、もう一度相性診断をしたい時には[YES]を押します。[NO]を押すと時計モードにもどります。



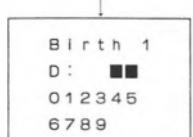
2人の生年月日から
相性を出します。



Birthday 1 の入力
• 年の入力
[←] 1 文字取消し
[CR] 入力終了



• 月の入力、必ず2ケタで入力してください。
(1ケタの場合は最初に0を)



• 日の入力、必ず2ケタで入力してください。

Birth 2
Y: 19■■

D: ■■
0 1 2 3 4 5
6 7 8 9



Birthday 2 の入力

Birthday 1 と同様に入力してください。

[CR] を入力

NEXT ♀
YES / NO

YESでもう一度。
NOで時計モード

プログラムリスト

ダンプリスト

Add	+B	+I	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
B800	01	00	00	92	05	16	00	00	00	02	16	00	92	05	00	:5D	...4.....4...	
B810	00	00	00	00	00	00	41	49	53	48	49	4F	55	53	48	:ADAISHIOUSH	
B820	49	4E	44	41	4E	18	05	00	16	B8	7C	05	CD	C1	43	CD	:74	INDAN...クリ.ヘチCヘ
B830	06	41	21	93	BD	AF	05	12	77	23	18	FC	21	FE	BC	CD	:CD	.AITスツ...ウB. ! シハ
B840	0C	BB	21	8C	BD	CD	15	41	21	25	BD	CD	17	BB	DD	21	:74	.サ! 2へ. A!エヌヘ. サン!
B850	93	BD	CD	43	BB	21	2C	BD	C0	I7	BD	DD	21	94	BD	CD	:E8	トスヘセ! .スヘ. サ! トスヘ
B860	43	BB	DD	7E	00	FE	13	38	F6	FE	01	38	F2	21	33	BD	:CA	Cサン! . .8年! 3ス
B870	CD	I7	BB	DD	21	95	BD	CD	43	BB	DD	7E	00	FE	00	28	:3B	ヘ. サン! -スヘCサン! . .(
B880	F6	3A	94	BD	CD	EF	BA	DD	BE	08	38	EB	3A	94	BD	FE	:3E	8: :スヘCセ. 8年! -ス
B890	02	28	8F	3A	93	BD	CD	91	BB	28	87	DD	7E	00	FE	29	:7D	., .:トスヘ. .シ. .)
B8A0	28	D5	CD	C1	43	CD	06	41	21	85	BD	CD	8C	BB	21	8C	:86	(エハCヘ. A!. スヘ. サ! .
B8B0	BD	CD	I5	41	21	25	BD	CD	17	BB	DD	21	98	BD	CD	43	:E5	スヘ. A!エヌヘ. サン! -スヘC
B8C0	BB	21	2C	BD	CD	I7	BB	DD	21	99	BD	CD	43	BB	DD	7E	:DE	サ! .スヘ. サン! -スヘCサン! . .
B8D0	00	FE	13	38	F6	FE	01	38	F2	21	33	BD	CD	17	BB	DD	:ED	. .8年! 3ス. サン!
B8E0	21	9A	BD	CD	43	BB	DD	7E	00	FE	00	28	F6	3A	99	BD	:4A	トスヘCサン! . .(分)ス
B8F0	CD	EF	BA	DD	BE	00	38	EB	3A	99	BD	FE	02	28	8F	3A	:2D	ハコセ. 8年! -ス . .:
<hr/>																		
Sum																		
Add	+B	+I	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
B900	98	BD	CD	91	BB	20	07	DD	7E	00	FE	29	28	D5	21	93	:C8	rスヘヤ. .ン. .(エ!ト
B910	BD	11	98	BD	06	03	1A	BE	38	08	20	11	13	23	10	F6	:B1	ス. フ. .セ8. . .#. 4
B920	18	0B	21	93	BD	11	98	BD	06	03	CD	B9	BA	DD	21	93	:D4	..!ス. .ス. .ヘクソ!ト
B930	BD	CD	C3	BA	DD	21	98	BD	CD	C3	BA	21	93	BD	3A	98	:E7	スヘデコ! !スヘコ! !トスヘア
B940	BD	BE	28	19	21	EA	BC	CD	C6	BB	3A	93	BD	CD	91	BB	:74	スセ! . !シヘニ! :トスヘア

BCA0 B1 BC 2F A6 77 C9 CD B1 BC AE 77 C9 CD B1 BC A6 :8A	アシ/ヲW/ヘシヨW/ヘシヲ
BCB0 C9 7D 0F 0F E6 03 57 07 07 82 07 07 82 07 84 :5E	ノ) ... ▲.W..---■
BCC0 16 00 5F 7D 21 A6 BD 19 E6 07 3C 47 3E 01 0F 18 :5D	...)!ヲス.▲.<G>..■
BCD0 FD C9 00 00 00 31 00 59 00 90 01 20 01 51 01 81 :D5	ノ. . . I.Y.▲. . Q.
BCE0 02 12 02 43 02 73 03 04 03 34 03 65 31 29 31 30 :2F	... C.s... 4.e110
BCF0 31 30 31 31 30 31 30 31 00 23 00 28 00 33 42 69 :AE	10110101. #(.3B1

Sum D8 E3 98 11 B5 40 43 64 27 E1 C6 9F CA 3A 97 7A :7A

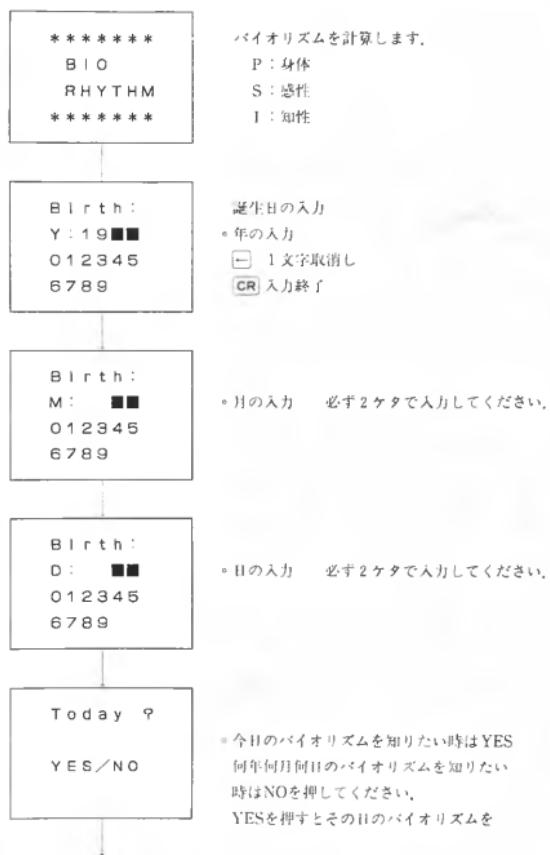
Add.	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum	ASCII
BD00	72	74	68	28	31	42	69	72	74	68	28	32	30	31	32	33	:B0	rth iBirth 20123
BD10	34	35	36	37	38	39	20	20	20	4E	45	58	54	20	3F	59	:9E	456789 NEXT ?Y
BD20	45	53	2F	4E	4F	59	3A	31	39	20	28	40	3A	20	20	:88	ES/NODY:19 M:	
BD30	28	28	28	44	3A	28	20	28	20	14	1F	13	1E	12	1D	:11	D:	
BD40	11	1C	18	18	0F	1A	0E	19	0D	18	BC	17	0B	16	0A	15	:30
BD50	89	14	08	13	87	12	06	11	05	18	05	0F	04	0E	04	0D	:B4
BD60	83	0C	03	0B	83	0A	83	09	03	08	03	87	03	06	04	85	:5D
BD70	04	04	05	03	06	02	07	01	06	01	09	00	0A	00	0B	00	:47
BD80	0C	00	0D	00	0E	00	0F	01	10	01	11	02	12	03	13	04	:87
BD90	14	05	08	00	00	00	00	00	00	00	00	00	00	00	00	00	:19
BDA0	00	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDB0	00	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00

Sum 4C 61 1A 25 1F 2C 10 18 1A 28 C7 F8 12 D6 D3 F4 :0F

バイオリズム

人間の身体、感性、知性は日々、その調子を変えていきます。今、自分はどういう状態にあるのか、それを知るために科学的に考察されたのがバイオリズムです。

「転ばぬ先の杖」当るも八卦当らぬも八卦、悪いと出れば慎重に、よいと出れば大胆な勝負に、そこであなたもバイオリズムを1つの行動の目安にしてはいかが、かな。



<p>Check</p> <p>Y: 19 ■■</p> <p>D: ■■</p> <p>0 1 2 3 4 5</p> <p>6 7 8 9</p>	<p>NOを押した場合は、バイオリズムを知りたい日の年月日を入力してください。</p> <table border="1" style="margin: auto;"> <tr> <td colspan="2">4 / 12</td> </tr> <tr> <td>P -</td> <td>■ +</td> </tr> <tr> <td>S -</td> <td>■ +</td> </tr> <tr> <td>I -</td> <td>■ +</td> </tr> </table> <p>→ で次の日 ← で前日を表示します。</p>	4 / 12		P -	■ +	S -	■ +	I -	■ +	<p><input type="button" value="CR"/> を入力</p> <p>NEXT ?</p> <p>YES / NO</p>
4 / 12										
P -	■ +									
S -	■ +									
I -	■ +									

プログラムリスト

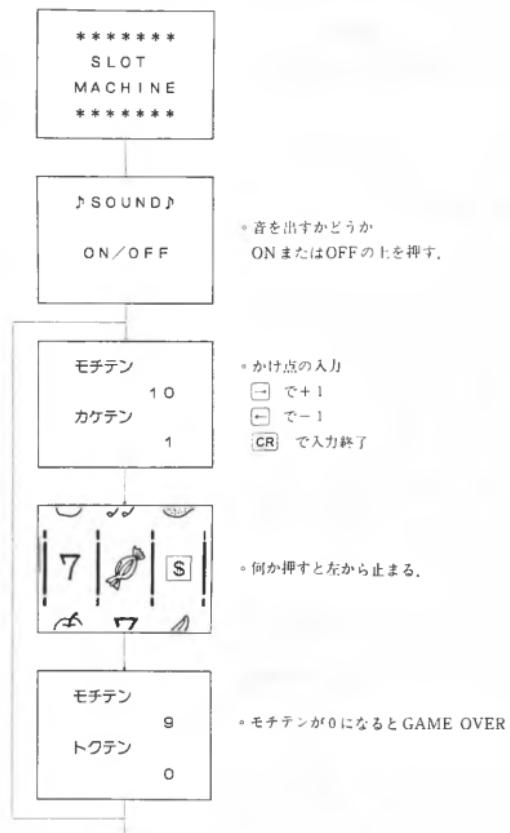
ダンプリスト

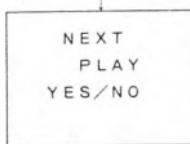
Add	+8	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
B880	01	00	00	21	05	16	00	00	00	02	16	00	21	05	00	:7B!.....!..	
B880	00	00	00	00	00	00	00	20	42	49	4F	20	20	20	52	:CC	BIO	
B820	45	59	54	48	4D	18	05	00	16	00	08	05	CD	01	43	:CD	23	
B830	05	41	21	22	BD	AF	06	10	77	23	18	FC	21	DE	BC	:3A		
B840	17	2B	21	EC	BC	ID	51	41	21	08	BD	23	BB	DD	21	:5B		
B850	22	BD	CD	4E	BB	21	13	BD	CD	22	BD	DD	21	23	BD	:FB		
B860	4E	BB	DD	7E	00	FE	13	30	F5	FE	01	38	F2	21	1A	:BC		
B870	CD	22	BB	DD	21	24	BD	CD	4E	BB	DD	7E	00	FE	00	:EB		
B880	F6	3A	23	BD	CD	FA	BD	DD	00	38	EB	3A	23	BD	FE	:67		
B890	02	28	0F	3A	22	BD	CD	9C	BB	20	07	DD	7E	00	FE	:29		
B8A0	28	05	CD	C1	43	CD	06	41	21	05	BD	CD	17	BB	CD	:DF		
B8B0	BA	BB	28	6B	21	05	BC	CD	17	BB	21	EC	BC	CD	15	:46		
B8C0	21	0C	BD	CD	22	BB	DD	21	27	BD	CD	4E	BB	21	13	:3D		
B8D0	CD	22	BB	DD	21	28	BD	CD	4E	BB	DD	7E	00	FE	13	:FF		
B8E0	F6	FE	01	38	F2	21	1A	BD	CD	22	BB	DD	21	29	BD	:72		
B8F0	4E	BB	DD	7E	00	FE	00	28	F6	3A	28	BD	CD	FA	BA	:FD		
Sum																		
AF	BC	70	A0	2F	58	00	85	EA	BF	6C	7E	77	CA	12	6C	:D9	ASCII	
Add	+8	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
B900	BE	00	38	EB	3A	28	BD	FE	02	28	1F	3A	27	BD	CD	9C	:C6	セ、84!(ス
B910	BB	20	17	DB	7E	00	FE	29	28	D5	18	0E	21	27	BD	DB	:77!...!...!
B920	35	77	23	DB	35	77	23	DB	34	77	CD	01	43	CD	06	:41	W#D5W#D4W#N!C..	
B930	21	28	BD	11	E5	BF	CD	28	BC	3E	2F	12	13	21	29	:B5	!(スル!シ!)	
B940	CD	28	BD	CD	IA	42	21	22	BD	11	27	BD	06	03	IA	:B8	!!!(スル!シ!	
B950	DA	2C	BB	20	04	13	23	10	F5	CD	21	22	BD	CD	CE	:4F	レグ、!!!(スル!シ!	
B960	DD	21	27	BD	CD	CE	BA	21	22	BD	3A	27	BD	BE	28	:54	!!!(スル!シ!	
B970	21	76	BC	CD	E0	BB	3A	22	BD	CD	9C	BB	20	03	CD	:C2	!!!(スル!シ!	
B980	HB	21	22	BD	CD	F4	BA	18	DE	21	2A	BD	CD	E0	BB	:B1	!!!(スル!シ!	
B990	25	BD	CD	F5	BB	21	2C	BD	11	2F	BD	01	83	08	ED	:B0	!!!(スル!シ!	
B9A0	CD	7B	BA	11	BB	3E	58	CD	41	48	21	84	BC	CD	0A	:AB	!!!(スル!シ!	
B9B0	BC	21	8A	BC	CD	61	BA	CD	69	BA	3E	01	IE	88	CD	:IC	!!!(スル!シ!	
B9C0	41	CD	7B	BA	11	BB	3E	53	CD	41	48	21	86	BC	CD	:DC	!!!(スル!シ!	
B9D0	00	BC	21	A1	BB	CD	61	BA	CD	69	BA	3E	02	IE	00	:D7	!!!(スル!シ!	

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII	ES/NO	Today	?Y:19
BD00	45	53	2F	4E	4F	54	6F	64	61	79	28	3F	59	3A	31	39	:C1
BD10	28	28	28	4D	3A	28	28	28	28	44	3A	28	28	28	28	28	:85	M:	D:
BD20	28	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:28
BD30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BD40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BD50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BD60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BD70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BD80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BD90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BDF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
Sum	85	73	4F	9B	89	74	8F	84	81	99	64	79	79	5A	51	59	:66

スロットマシン

もうカジノへ行かなくても、ギャンブルができます。ただ今のあなたの持ち点は10点、増えるも減るもあなたの腕とツキしだい。電車の中で、ちょっとした待ち時間にRC 20を使ってラスベガス、億万長者になることもありませんし、もちろんスッテンテンになることもない、良心的な金銭無用のスロットマシーンです。





。もう一度やりたい時はYES.

やめたい時はNOを押す。

プログラムリスト

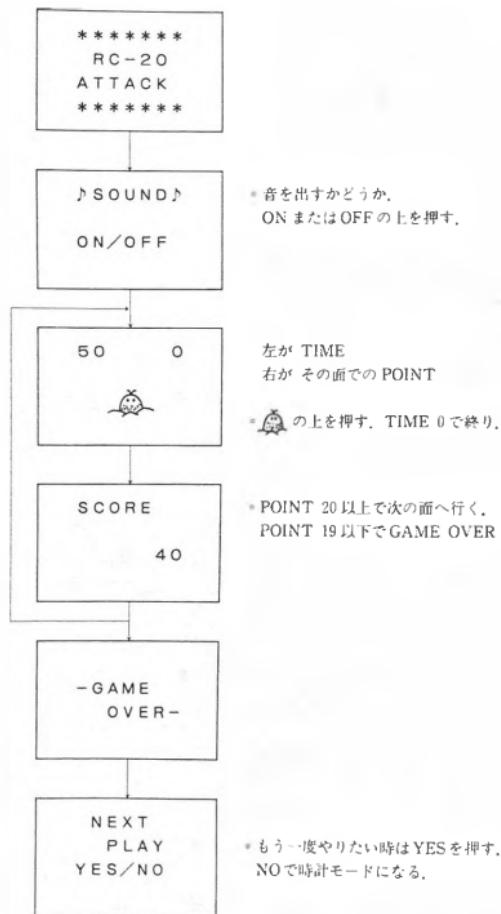
ダンプリスト

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
B800	81	00	00	4A	06	16	00	00	00	02	16	00	4A	06	00	:CF	...J.....J..	
B810	00	00	00	00	00	00	00	00	00	00	53	4C	4F	54	20	20	D4	:30
B820	43	48	49	4E	45	18	85	00	16	B8	34	00	CD	C1	43	CD	:2A	
B830	06	41	CD	98	99	CD	B9	CD	F9	CD	A2	B9	CD	C1	:DC	.Aヘテヘセガヘ ケヘツハヘ		
B840	43	CD	06	41	CD	D7	BA	CD	06	41	CD	F3	BA	:A9	Cヘ. Aヘラツベコヘ. Aヘヨコ			
B850	CD	FF	BA	CD	06	41	CD	22	BB	CD	F1	B9	21	4D	BE	11	:F7	
B860	58	BE	CD	8E	BA	CD	41	BB	CD	21	BB	CD	49	BB	CD	7F	:B2	
B870	BC	CD	49	BB	CD	72	B9	32	57	BE	CD	72	B9	32	58	BE	:0C	
B880	CD	72	B9	32	59	BE	CD	9E	BB	CD	49	BB	21	57	BE	7E	:EC	
B890	23	B6	23	B6	28	F8	66	0A	CD	E8	5F	3A	5B	BE	16	00	:4F	
B8A0	5F	21	3A	BE	19	46	3A	5C	BE	16	00	5F	21	42	BE	19	:DA	
B8B0	4E	3A	5A	BE	16	00	5F	21	32	BE	19	7E	FE	02	20	0A	:E7	
B8C0	1E	B1	88	28	13	IE	B2	B9	20	0E	BB	20	IC	B9	20	19	:F7	
B8D0	21	B1	BD	86	00	4F	09	5E	43	C5	21	53	BE	11	58	BE	:A4	
B8E0	CD	9C	BA	C1	18	F3	CD	84	BA	21	4D	BE	11	53	BE	CD	:8D	
B8F0	9C	BA	CD	C1	43	CD	D7	BA	CD	E7	BA	CD	06	41	CD	F3	:C7	
<hr/>																		
Sum																		
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
B900	BA	CD	04	BB	86	0A	CD	E8	5F	21	4B	BE	CD	85	BA	28	:C8	
B910	14	CD	58	BA	D2	3B	B8	21	4B	BE	11	4E	BE	01	03	00	:FB	
B920	ED	BB	C3	3B	BB	CD	C1	43	21	16	BE	11	EB	BF	01	BE	:E3	
B930	00	ED	BB	CD	06	41	CD	B9	BA	06	0A	CD	E8	5F	CD	C1	:F3	
B940	43	21	24	BE	11	E4	BF	01	BE	00	ED	BB	CD	B6	41	CD	:87	
B950	28	42	21	B9	BE	CD	F7	41	CD	91	43	FE	11	38	FB	FE	:2D	
B960	14	DA	2C	B8	FE	15	28	84	FE	16	20	E3	CD	F1	B9	C3	:62	
B970	9A	48	06	14	C5	CD	9E	BB	CD	49	BB	CD	91	43	C1	B7	:C9	
B980	20	02	18	F0	CD	F1	B9	3E	B1	C9	AF	77	23	1B	FC	C9	:BF	
B990	21	4B	BE	06	14	CD	8A	B9	21	4D	BE	36	10	21	58	BE	:F5	
B9A0	36	81	DD	21	57	BE	06	83	CD	8A	BB	E6	BF	C6	8A	DD	:87	
B9B0	77	00	DD	23	1B	F2	21	51	BE	85	83	C3	8A	B9	21	0F	:E8	
B9C0	BE	11	E4	B1	07	00	ED	BB	CD	1A	42	CD	28	42	21	:98		
B9D0	83	BE	CD	F7	41	CD	91	43	FE	11	28	FB	FE	12	28	0B	:FB	
B9E0	FE	14	3B	E8	FE	17	30	E4	AF	1B	02	3E	01	32	5E	BE	:B1	
<hr/>																		

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
BDB0	01	8A	00	ED	BB	13	13	13	13	DD	23	01	08	00	FD	09	:83	..O...ン#...
BD10	C9	00	7C	7C	60	60	61	63	7E	7C	00	00	00	00	7E	:BD	J..I``ac`l..`	
BD20	FE	00	00	00	00	00	03	04	00	05	78	44	2A	15	0C	30	:CCxD*.0
BD30	A8	54	22	1E	10	10	20	C0	00	00	00	00	00	01	02	:3F	4T`..9....	
BD40	4C	70	1F	60	90	90	60	98	24	24	18	50	80	01	02	:9A	Lp.``r\$..`..	
BD50	0A	09	00	19	24	42	41	E0	10	08	04	04	04	04	08	:00	...\$BA=...	
BD60	E8	03	04	00	10	19	14	14	08	04	03	E0	10	48	04	:24	=.....,=H,\$	
BD70	B4	84	00	10	E0	00	00	00	F0	10	30	50	0F	00	00	:2E	.m.=...0P...	
BD80	38	C4	02	02	02	02	02	C4	38	03	05	0A	15	1A	1A	:72	8t.=...f8....	
BD90	0A	05	03	E0	50	A8	54	20	20	54	A8	50	E0	00	3F	:3D	...=P4T..T4P=2?	
BDA0	3B	20	3B	20	3B	3D	3F	00	FE	DE	60	82	60	02	6E	:9E	:i:=?.`n.n`n	
BDB0	FE	IE	05	07	11	8A	0F	14	19	C0	02	C7	02	D0	02	:A3タヌミヌ	
BDC0	02	00	01	D0	02	DA	02	E0	02	DA	02	E0	02	DE	03	:32	..ミ.レ.=.レ.=..	
BDD0	FF	DE	01	E0	01	E8	01	DE	01	E8	01	E8	01	DE	01	E0	:10	..=..+I..`..=..
BDE0	01	E0	02	00	FF	80	02	BF	02	89	02	90	02	82	02	96	:41	◆..+I..`..=..
BDF0	02	90	03	00	FF	D3	C1	C3	DD	B6	B9	C3	DD	C4	:38	..+.モナテンカケント		
Sum	29	46	22	60	67	A3	F4	EA	25	9C	E9	15	F8	4F	C4	:E1	:84	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
BEE0	BB	C3	DD	4F	4E	2F	4F	46	46	59	45	53	2F	4E	4F	9A	:56	クリON/OFFYES/NO
BE10	53	4F	55	4E	44	9A	20	47	41	40	45	28	28	28	28	4F	:39	SOUND-L-GAME 0
BE20	56	45	52	20	20	4E	45	58	54	20	20	28	28	50	4C	:41	VER-NEXT PLA	
BE30	59	20	00	01	02	03	04	05	06	07	00	04	06	01	07	02	:A9	Y
BE40	05	03	00	06	05	02	03	01	07	04	00	00	00	00	00	00	:24
BE50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BE60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BE70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BE80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BE90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BEA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BEB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BEC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BED0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BEF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
Sum	BF	7A	84	D1	B9	1C	C8	EB	E8	D1	AA	97	75	BF	C2	:32		

もぐらたたき

さて、RC-20の中に潜むもぐらとあなたの反射神経が競い合う。限られた時間内で、何匹のもぐらをK.Oできるか。20匹以上もぐらを叩けば、次の面へ、19匹以下だとあなたの負けです。



プログラムリスト

ダンブリスト

BAF# 19 BB C9 11 BA BF 18 83 11 37 BB E6 87 6F 87 87 :AF .サノコ...7ツ.0

Sum 87 3D 6C 1A C9 B0 E2 D1 4E BE 14 92 6A 60 1F 33 :C4

Add +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum

ASCII

BB00 87 95 GF 26 88 19 C9 3E 81 1E 48 18 18 3E 81 1E :85
BB10 88 18 8A 3E 82 1E 88 18 84 3E 02 1E 48 E5 CD 1C :88
BB20 41 E1 C9 2A 2B BC 54 5D 29 29 19 7D 84 67 11 81 :87
BB30 00 19 22 2A B7 7C C9 0E 11 24 31 24 11 0E 8E 51 :F2
BB40 24 31 24 51 8E 91 64 31 64 91 8E 4E 91 64 31 :83
BB50 64 91 4E 00 00 18 3C 18 00 00 00 00 18 30 66 3C 18 :BD
BB60 00 18 42 00 81 00 42 18 00 00 00 00 00 00 00 00 :C7 :FC
BB70 02 D8 82 C7 83 00 81 BD 81 AB 81 B5 81 A1 82 9B :FD
BB80 C7 82 D5 82 C7 84 00 FF E8 81 DE 81 E8 81 DE :EB
BB90 81 E0 01 DE 81 E0 02 00 FF DE 81 00 81 DE 81 00 :61
BB00 81 DE 81 00 FF DE 81 E0 81 E8 81 DE 81 E8 01 E8 :30
BB10 81 DE 81 E0 81 E8 82 00 FF AB 02 B5 82 AB 02 8F :4A
BB20 82 B5 02 82 82 9B 02 A1 82 82 03 00 FF 53 43 4F :E6
BB30 52 45 58 45 52 46 45 43 54 2D 42 4F 4E 55 53 2D :81
BB40 20 20 31 30 30 20 4F 4E 2F 4F 46 46 59 45 53 :A9
BB50 F0 2F 4E 4F 9A 53 4F 55 4E 44 9A 2D 47 41 4D 45 20 :F0

Sum FA 1C E0 EB CA E2 BB 73 57 81 E4 7F 48 C7 34 FB :35

Add +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum

ASCII

BC00 20 20 20 4F 56 45 52 2D 20 4E 45 58 54 20 20 20 :88
BC10 20 50 4C 41 59 20 88 00 00 00 00 00 00 00 00 00 :76
BC20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BC30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BC40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BC50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BC60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BC70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BC80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BC90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BCA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BCB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BCC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BCD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BCE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
BCF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00

Sum 40 70 6C 98 AF 65 52 2D 20 4E 45 58 54 20 20 20 :FE

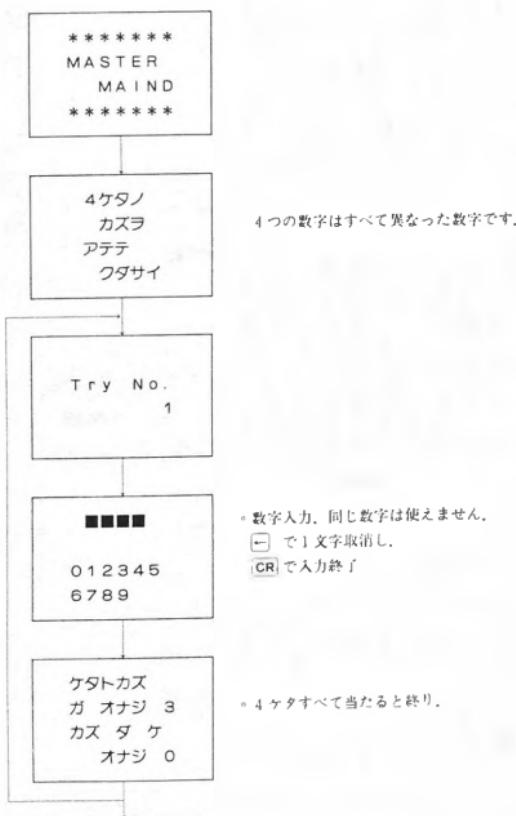
OVER-NEXT

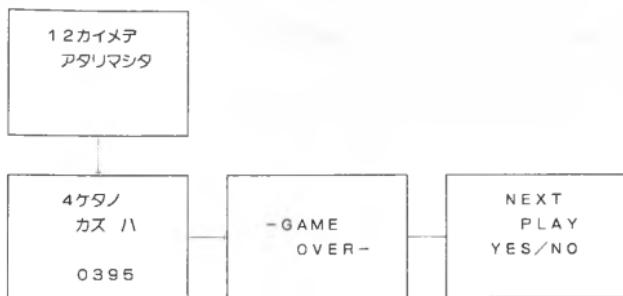
PLAY

数当てゲーム

RC-20があなたを試します。遊び方は簡単です。RC-20がその小さな頭の中に4桁の数を思い浮かべ、何度もあなたが質問し、それに対するRC-20からのヒントから、RC-20が思い浮かべた数を当てるというものです。

できるだけ少ない回数で当ててください。RC-20は決して嘘をつきません。頭をとびきり回転させて、RC-20の鼻をあかしてください。





プログラムリスト

ダンプリスト

Add	+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum	ASCII
B800	81 88 88 81 83 16 80 00 00 00 02 16 00 91 03 00 :57T.....T..
B810	88 88 88 88 00 00 4D 41 53 54 45 52 20 20 20 :2CMASTER
B820	4D 41 49 4E 44 19 05 88 16 88 7B 03 CD C1 43 CD :78	MAIND... ク. マチハ
B830	06 41 CD 4C BA 21 F6 BA 11 E4 BF 81 IC 00 ED B0 :59	.AヘLゴ1コ. ソー.
B840	CD 06 41 3E 01 32 9A BB CD 6F RA CD 4C BA CD C1 :31	ヘ.A>.2ウハゴルコヘチ
B850	43 CD 06 41 21 EF BA 11 EB BF 81 87 00 ED B0 21 :A2	C\..A\>コ. ソン. オ-イ
B860	9A BB 17 F7 BD CD B2 BA CD 06 41 CD 4C BA CD C1 :CA	ト. パソコバコ. A\>コハチ
B870	43 CD 06 41 21 12 BB CD 15 41 21 E5 BF 3E A0 06 :11	C\..A\>コ. A\>ソ.
B880	04 77 23 10 FC DD 21 E5 BF DD 36 00 A0 CD 1A 42 :28	.W#. .ン\>ソ. ヘ.B
B890	CD 58 BA FE 15 28 F9 C6 30 DD 77 00 CD 1A 42 DD :63	\X\> . .ニ\>W. ヘ.BN
B8A0	36 01 A0 CD 1A 42 CD 58 BA FE 15 28 DC C6 30 DD :C9	6. ヘ.BXコ. (ワニ\>0
B8B0	BE 00 28 F2 DD 77 01 CD IA 42 DD 36 02 A0 CD 1A :F2	セ. (ヨン. ヘ.BN. 6. ヘ.
B8C0	42 CD 58 BA FE 15 28 D7 C6 30 DD BE 00 28 F2 DD :BB	B\>Xコ. (ラコ\>セ. (ヨン
B8D0	BE 02 28 ED DD 77 02 CD IA 42 DD 36 03 A0 CD 1A :F1	セ. (ヨン. ヘ.BN. 6. ヘ.
B8E0	42 CD 58 BA FE 15 28 D2 C6 30 DD BE 00 28 F2 DD :B6	B\>Xコ. (メニ\>セ. (ヨン
B8F0	BE 01 28 ED DD BE 02 28 E8 DD 77 03 CD 1A 42 CD :CE	セ. (ヨン. ヘ.BN. 6. ヘ.
Sum 06 4A 19 FD C1 6C F8 C8 53 DD 5A F8 AD 68 89 FD :70		
Add	+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F :Sum	ASCII
B900	91 43 B7 28 FA FE 33 28 D1 FE 34 28 F2 21 E5 BF :E8	マC#1 3(ル 4 年\>ソ
B910	11 96 BB 81 04 00 ED B0 21 4D BB 11 E4 BF 81 IC :FE	.1サ. .O-IMサ. ソ.
B920	00 ED BB CD 86 41 06 04 0E 30 21 96 BB 11 92 BB :C9	.O-ヘ. A.. 0!1!9. イサ
B930	1A BE CC 56 BA 23 13 18 F7 41 79 32 F1 BF DD 21 :BB	.セフコ\>. ハヤ2円ソ!
B940	95 BB 21 92 BB BE 30 7E DD BE 01 CC 56 BA DD BE :BE	1サ!4サ. 0-セ. フコセ
B950	02 CC 56 BA DD BE 03 CC 56 BA 23 7E DD BE 00 CC :68	.フコセ. フコ\>セ. フコ#
B960	56 BA DD BE 02 CC 56 BA DD BE 03 CC 56 BA 23 7E :A4	Vコセ. フコセ. フコ#
B970	DD BE 00 CC 56 BA DD BE 01 CC 56 BA DD BE 03 CC :59	ンセ. フコセ. フコセ. フ
B980	56 BA 23 7E DD BE 00 CC 56 BA DD BE 01 CC 56 BA :A0	コ#セ. フコンセ. フコ
B990	DD BE 02 CC 56 BA 79 32 FF BF C5 CD 86 41 CD 4C :D4	ンセ. フココ2 ソヘ. ANL
B9A0	BA C1 78 FE 34 28 26 3A 9A BB FE 99 28 89 C6 01 :91	コチ. 4:& ナヘ. ニ.
B9B0	27 32 9A BB C3 4E BB CD C1 43 21 69 BB 11 EB BF :48	'2-リテNコヘチC!1サ. ソ
B9C0	81 0E 00 ED BB CD 06 41 CD 4C BA 18 1D CD C1 43 :99	...O-ヘ. ANLコ. ハチC

ポーカーゲーム

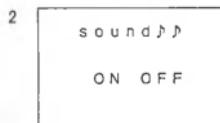
プログラムをスタートさせると、まず、音を出すかどうかを聞いてきますので画面上の[ON]または[OFF]を押して選んでください。

持ち点(CREDIT)は20点からスタートです。[MENU]キーを押して賭け点(BET)を決めます。なお、[MODE]キーで0にもどります。

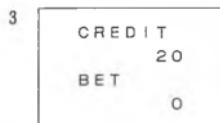
[CR]キーを押すと、カードが配られます。残しておきたいカードを画面の下に5つ並んでいる[○]を押して選びます。すると、そのカードの上に、Hマークが表示されます。取消したいときには、[MODE]キーを押します。カードが決まったら[CR]キーを押してください。Hマークのないカードは捨てられ、新しいカードを引いてきます。

ここで、役ができるいれば画面にその役と倍率が表示されますが、ワンペアは得点となりません。賭け点に倍率を掛けた点数がクレジットに加算され、1ゲーム終了です。

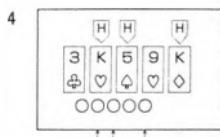
持ち点がなくなるとゲームオーバーです。もう一度やるかどうか聞いてきますので[YES]または[NO]を押してください。なお、ゲームを途中でやめるときには、[MODE]キーを1.5秒以上押し続けてください。



音を出すかどうか聞いてくる。
'ON' または 'OFF' の文字の上を押して選ぶ。



賭け点を入力する。
'MENU'キーで増加し、'MODE'キーで取消し。



カードが配られる。
残しておきたいカードを、下の'○'を押して選ぶ。
カードが決まったら'CR'キーを押す。
また、'MODE'キーでキャンセルできる。

- 5 カードを引いてくる。
- 6 この場合、2ペアの役ができるている。また1ペアは役と見なされない。
- 7 役の表示。
- 8 踏け点に倍率を掛けたものがスコアとなる。
- 9 スコアがクレジットに加算される。

.....

GAME
OVER

クレジットがなくなると
ゲームオーバー。

Replay?

YES NO

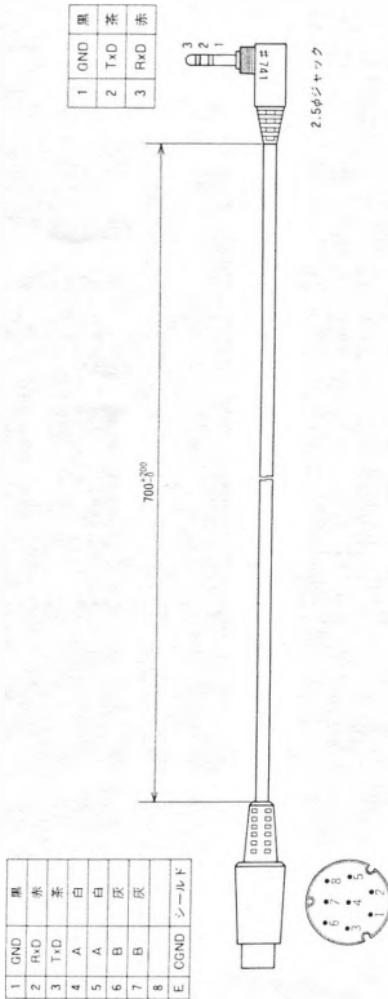
もう一度やるときは'YES'を押す。

プログラムリスト

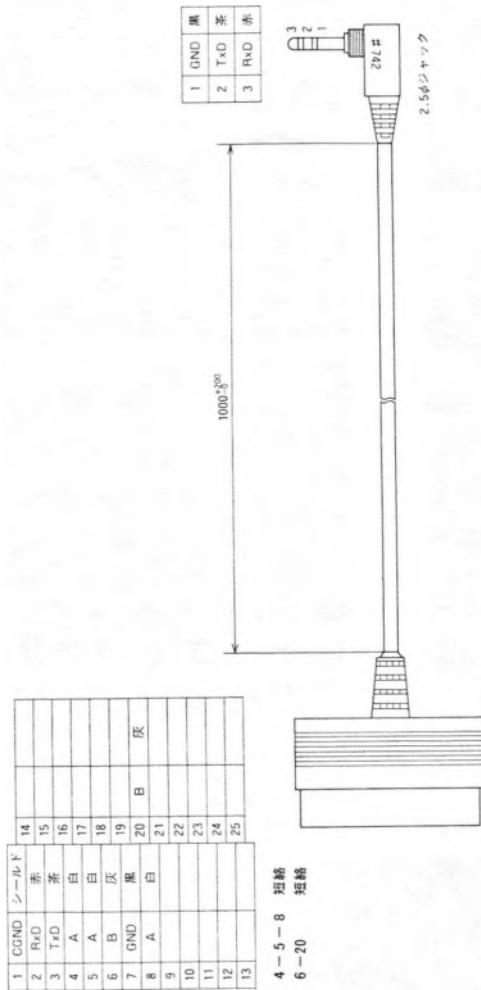
ダンプリスト

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII	
B800	01	00	00	92	06	16	00	00	00	00	02	16	00	92	06	00	:5F4.....4...	
B810	00	00	00	00	00	00	00	2A	:7A*****									
B820	2A	2A	2A	2A	18	05	00	16	B8	7C	06	CD	93	B8	CD	:24	*****.!21..ヘトクヘ		
B830	ED	B8	CD	C4	BB	CD	70	B9	CD	5B	B9	CD	32	B9	CD	:E2	□クヘタハケヘ[ケハ2ケヘー		
B840	B9	CD	DF	B9	CD	47	BA	CD	32	B9	CD	95	B9	CD	DF	B9	:24	ケハ!ケヘGハ2ヘ~ケ~ケ	
B850	CD	99	BA	CD	B1	BA	CD	C2	BA	CD	E7	BA	CD	I3	BB	CD	:77	△コアコツコヽコヽサハ	
B860	C5	BC	20	CB	I8	00	CD	DC	BC	CD	C1	43	3E	08	I1	4A	:5B	ナシヒ..ヘフシナヒC..J	
B870	BD	CD	D2	B8	06	00	CD	E8	5F	I1	57	BD	CD	D1	B8	CD	:76	スヌズメ..ヘ◆..Wスムクヘ	
B880	91	43	FE	I1	38	F9	FE	I4	38	A2	28	F3	FE	I1	DA	9A	:AE	△C..8..8'(月!レ	
B890	40	18	EC	I1	00	00	22	AD	BE	3E	20	32	AF	BE	CD	C1	:7D	ミ.●!..「ユセ」.2ヤセハチ	
B8A0	43	3E	07	I1	21	BD	CD	D2	B8	86	8A	CD	E8	5F	I1	29	:2C	C>..!スメク..ヘ◆..)	
B8B0	BD	CD	I1	B8	CD	91	43	FE	I1	38	F9	FE	I1	38	07	FE	:42	スヘムクヘC..8..8.	
B8C0	21	30	F1	AF	I8	02	3E	01	32	B8	BE	DC	DC	B3	C3	6A	:93	!Bリツ..>.2ケツワジツ	
B8D0	42	4F	CD	D8	B8	C3	86	41	FE	28	D0	06	00	4F	I2	E4	:A8	△マリクテ.A(ミ..01▲	
B8E0	BD	BF	89	7D	B7	FB	I4	B7	CB	77	I3	23	I8	F5	AF	32	B6	:D6	ソ.キX.キズW.#.時ツカ
B8F0	BE	CD	I4	B9	0E	01	21	A2	BE	77	23	E5	CD	I4	B9	:C6	セヘ.ケ..!「セW#.ハ.ケ		
Sum																			
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII	
B900	CI	21	A2	BE	41	BE	28	F4	23	I0	FA	E1	77	8C	79	FE	:65	チ!「Bアセ(日#, ハ.у.	
B910	0B	38	E7	C9	DD	18	47	2A	B9	BE	54	50	29	29	I9	7D	:5F	.8'ノロ.G*カツ丁).)	
B920	84	67	I1	81	00	19	22	B9	BE	7C	88	FE	D0	38	E5	E6	:F4	■q.!.「ケセ」.ミ!■	
B930	F3	C9	DD	21	93	BE	06	05	DD	CB	00	46	20	18	I2	B6	:13	月ソ!トセ..ヒビ..F..!カ	
B940	BE	5E	34	I6	00	21	A2	BE	I9	AF	ED	6F	DD	77	05	ED	:51	セ^4..!トセ..ヒロ!ロンw.○	
B950	6F	DD	77	0A	ED	6F	DD	23	I8	DE	C9	81	00	54	21	BB	:11	○ンw.○ン#.~ノ..T!サ	
B960	BE	I8	13	81	88	83	21	AD	BE	I8	0B	01	01	05	I8	03	:BE	セ.....!ユセ.!.~.	
B970	01	00	05	21	93	BE	I1	23	I0	FC	C9	3E	01	I1	BB	BE	:BA	...!トセ#..!トセ.!リセ	
B980	CD	88	B9	3E	02	21	E5	BE	I1	BA	BF	01	I2	AB	00	ED	B8	:64	△!グソ.△セ.コリ.*.○
B990	IE	00	C3	I1	C4	CD	84	42	I1	93	BE	21	BC	BE	06	05	:D9	..テ.A~B.ヒセシ..	
B9A0	C5	IA	B7	20	31	36	FF	23	3E	55	06	05	F6	80	77	2F	:F9	ナ.キ!6..#3J..9~W/	
B9B0	23	I0	F9	36	FF	01	24	00	09	36	FF	23	3E	55	06	05	:85	#. 6 ..\$. #JU..	
B9C0	F6	01	77	2F	23	I0	F9	36	FF	01	D8	FF	09	D5	ED	:66	△.L..W/#. 6 .リ..2Kヘ		
B9D0	7B	B9	E1	I1	I8	04	01	08	00	I3	C1	I8	C2	C9	CD	:50	(ケム!...ヒ.ツ.ツ/ハ		
B9E0	47	42	DD	21	93	BE	06	00	DD	CB	00	46	I0	36	DD	7E	:7D	GBN!ヒ..ンヒ.F 6~	

#741(HC-80/88/40/41用)



#742 (QC-10, PC-8001, PC-8001MKII, PC-8801, PC-8801MKII等)



付録

RC-20とパソコンの接続ケーブル

#743(HC-20用)



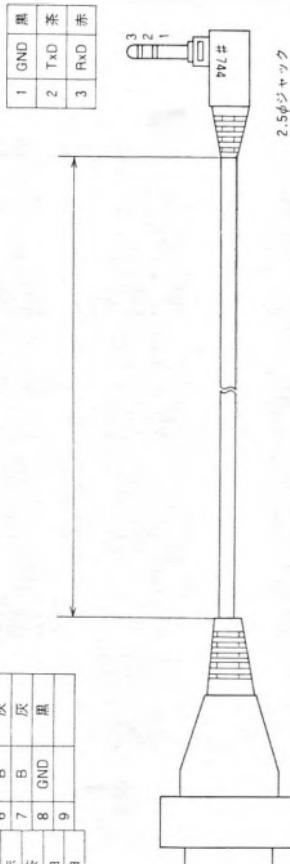
付録

RC-20とパソコンの接続ケーブル

#744 (MZ関係用)

1	C/GND	シールド	6	白	灰
2	RxD	赤	7	白	灰
3	TxD	茶	8	GND	黒
4	A	白	9		
5	A	白			

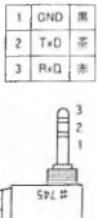
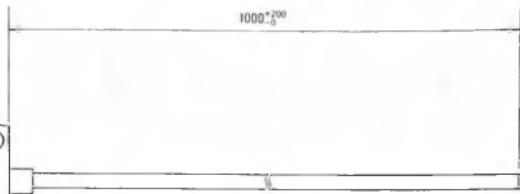
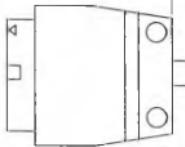
4-5 直結
6-7 直結



206

#745(FM関係用)

1		2	
3		4	
5		6	
7		8	
9		10	
11		12	
13	A 灰	14	GND 黒
15		16	A 灰
17		18	B 白
19		20	B 白
21		22	TxD 茶
23		24	RxD 赤
25		26	CGND シールド



2.5φジャック

「日本
腕コンピュータ100%活用法

昭和60年3月15日 初版第1刷発行

著者 岡田 ひろし

発行者 片岡 崑

発行所 株式会社技術評論社

東京都千代田区平河町1-4-12

電話 03(262)9351

印 刷 ㈱加藤文明社

製 本 村上製本所 1200円

■定価はカバーに表示しております。

本書の一部または全部を無断で複写、複製、
転載、テープ化、ファイルに落とすことを
禁じます。

■本書の内容に関するお問合せは、必ず文書
によるようお願い致します。

ISBN4-87408-277-7 C3055