

# 腕コンピュータ100%活用法

リス+



はじめに

1159

14.2

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、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

- 1.1 時刻表示・カレンダー機能として使う
- 1.2 コンピュータとして使う

### 2 RC-20にマシン語プログラムを転送する

- 2.1 RS-232Cによる転送の規約
- 2.2 転送プログラムのフォーマット
- 2.3 プログラムの転送例
  - 1. QC-10の場合
  - 2. PC-8001/8001mk IIの場合
  - 3. PC-8801/8801mk IIの場合
  - 4. FM-7の場合
  - 5. HC-20の場合
  - 6. HC-40/80/88の場合

### 3 RC-20からのパソコンへのデータの転送

- 3.1 QC-10の場合
- 3.2 PC-8001/8001mk IIの場合
  - 1. CP/Mの場合
  - 2. DUADPCの場合
- 3.3 PC-8801/8801mk IIの場合
  - 1. CP/Mの場合
  - 2. DUAD88の場合
- 3.4 HC-40/80/88の場合

### 4 ハード構成とメモリマップ

### 5 ROM内サブルーチンの使い方

- 5.1 割込みおよびシステム処理サブルーチン
- 5.2 キー関係サブルーチン
- 5.3 入出力ポート制御サブルーチン
- 5.4 キャラクタ表示関係サブルーチン
- 5.5 演算データの型式と演算サブルーチン

## 目次

<b>6 表示機能</b>	102
<b>7 入出力ポート機能</b>	104
<b>8 割込み機能</b>	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
<b>9 浮動小数点演算における注意</b>	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
<b>10 スケジューラ、メモ、ワールドタイムのデータの構造</b>	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
<b>11 ダンプリストのプログラムの入力・転送</b>	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
<b>12 実用マシン語プログラム集</b>	147
卷末付録 ● RC-20とパソコンの接続ケーブル	202

**RC-20**

**NUMBER**

**1**

# こんな使い方 ができる

図1-1

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、一般的なパソコンと同様な 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

```

graph TD
    RAM[RAM] --- 4bitCPU[4ビットCPU]
    ROM[ROM] --- 4bitCPU
    4bitCPU --- 8bitCPU[8ビットCPU]
    4bitCPU --- LCDDriver[LCD ドライバ]
    LCDDriver --- LCD[LCD表示]
    KEY[KEY] --- 4bitCPU
    8bitCPU --- RS232C[RS-232C]
    RS232C --- PC[パソコン]
    
```

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自身のキーを使って、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ものですが、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ます。

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ます。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。



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のように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ポーネ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)、エラーアリーバーがアリーバーであれば、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

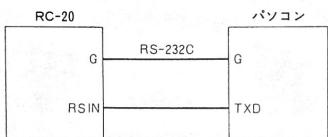


表2-1 転送データ

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ポーネ

## 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

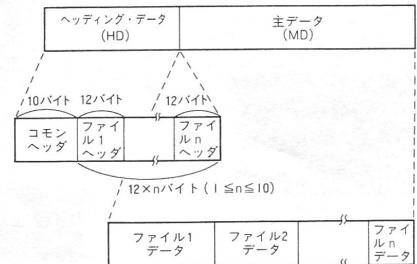
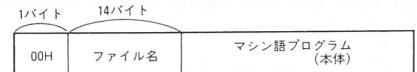


図2-3 ファイルデータの構造



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に、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)となります。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。

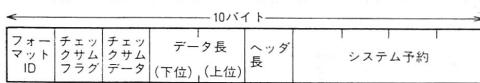
ファイルヘッダは1個が12バイトの大きさで、ファイルの数だけ作成され、RC-20のRAM

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。

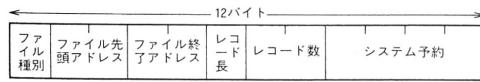
先頭の1バイトはプログラム種別を表わし、00Hが入ります。続く14バイトは当該プログラムのプログラム名で左につめて文字データが書かれ、データのないところには空白(20H)が書かれます。マシン語プログラム自体はその後ろに配置されます。

図2-4 コモンヘッダの詳細



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	01Hの時は、チェックサムデータ無効 01Hの時は、チェックサムデータ有効
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



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

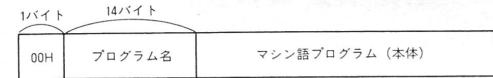


図2-7

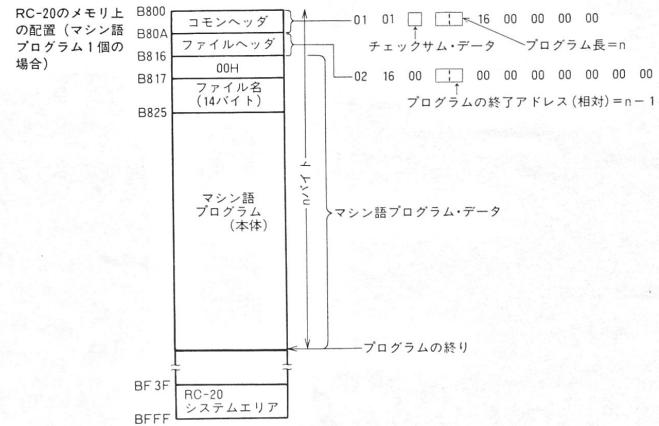


図2-8 マシン語プログラムの構造(1)

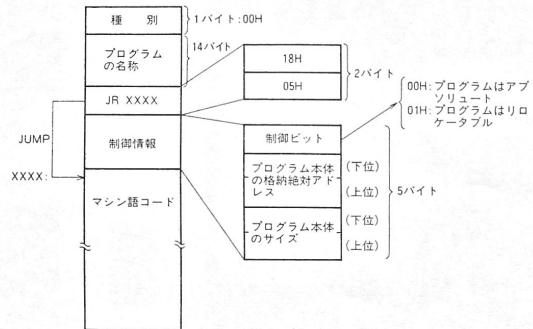
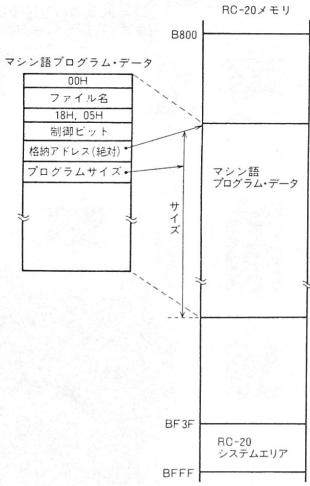


図2-9 マシン語プログラムの構造(2)



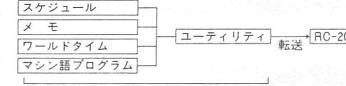
先にも述べたようにマシン語プログラム本体の最大は、 $1856 - 22 - 15 = 1819$  バイトですから、少しばかり複雑なプログラムを作成すると、プログラム1個でユーザ用RAMエリアを占めてしまうでしょう。

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、制御情報の後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。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)を実行せますと“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”的表示が出て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??”のモードになります。「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」キーを押すと、今転送したプログラムが実行され、画面に絵がで出來ます。

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    0BB16H
BFBA =    VRAMP  EQU    0BFB4H
002A =    PAGEL  EQU    2AH
411C =    DISP    EQU    411CH
41CB =    STBY   EQU    41CBH
409A =    ZBOEND EQU    409AH
```

```
;      ;          ORG    PORG
;          ;          MAIN: DB     0
B816 00      DB     'GRAPH PROGRAM '
B817 4752415048  DB     1BH,ENTRY-$-1 ;JR ENTRY
B825 1805      DB     0
B827 00      DB     0
B828 16BB      DW     MAIN
B82A 1601      DW     PGEND-MAIN
;
;          ;          ENTRY: LXI   H,MDATA
B82C 21B3BB  LXI   D,VRAMP
B82F 116ABF  LXI   B,PAGEL
B832 012A00  LXI   B,PAGE1
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 CD78B8  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 C378B8  JMP   LDIR
;
MDATA:
B883 000000000 DB    0,0,0,0,0,0,0,0,0,0,0,0
B88D 0102070A0E DB    1,2,7,0AH,0EH,1BH,23H
B894 343A232020 DB    34H,34H,23H,20H,20H,23H,34H,34H
B89C 23180E0A07 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
B8B2 0F3A6AAA8F DB    0FH,3AH,6AH,0AAH,0BFH,0AOH,0EOH
B8B9 000000415CE DB    0,0,4,15H,0CEH,20H,0A4H,0CEH
B8C1 0E0ECEE4A20 DB    0EH,0EH,0CEH,0A4H,20H,0CEH,15H
B8C8 0400000E0 DB    4,0,0,0EOH
B8CC A08FAA6A3A DB    0AOH,0BFH,0AAH,6AH,03AH,0FH
B8D2 0000000000 DB    0,0,0,0,0,0,0,0,0,0,0,0
B8DE A0A0B07008 DB    0AOH,0AOH,0BOH,70H,8,8,4,4,4
B8E7 82FB848C8C DB    82H,0FBH,84H,8CH,8CH,8CH,8CH,8AH,0FBH,82H
B8F1 0404040808 DB    4,4,4,B,8,70H,0BOH,0AOH,0AOH,0COH
B8FB 8000000000 DB    80H,0,0,0,0,0,0,0,0,0,0,1CH,22H
B908 26404464046 DB    26H,40H,46H,40H,46H,42H,42H,0BAH
B910 82B2020404 DB    82H,82H,2,4,4,4,4,4,2,92H,0BAH
B91C 9E7A524640 DB    9EH,7AH,52H,46H,40H,46H,40H,26H,22H
B925 1C00000000 DB    1CH,0,0,0,0,0
B92A 0000  DB    0,0
;
;
;
;
;
PRGEND:
B92C          END

```

リスト 2 転送プログラム(TRANS.BAS)

```

5 REM #transf
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 "#",#1,F$
150 IF EOF(1) THEN GOTO 1000
200 INPUT# 1,GPRO$
350 S$=MID$(GPRO$,2,2)
360 LNE=VAL(*$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
610 K=K+1
620 CHEE$(K)=CHE$ 
700 CHKS=CHKS+TXT0
900 GOTO 200
1000 CLOSE
1100 GOSUB 4000
1500 AX=CHKS-INT(CHKS/256)*256
1510 A$=MKI$(0-AZ)
1520 MID$(CHEE$(0),3,1)=LEFT$(A$,1)
1900 INPUT "src-20 ready ok";P$
2000 OPEN "#3,COM0:(CBNSF)"
2100 FOR S=0 TO K
2200 PRINT #3,CHEE$(S);
2300 NEXT
3000 CLOSE
3100 END
4000 CHD$=" "
4010 CHD$=CHD$+CHR$(KHI)

```

```

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=CHK$+ASC(TXT$)
4340 NEXT
4400 RETURN
9999 END

```

リスト3 RC-20用マシン語プログラム(例2)

```

; $KOHAN$
;
B816          ORG 0BB16H
;
409A =        ZBOEND EQU 409AH
4086 =        SAVER EQU 408AH
40C6 =        LOADR EQU 40C6H
42AE =        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 2144BB   LXI H,DATA
LOOP:
B82F 7E       MOV A,M
B830 FEFF     CPI OFFH
B832 CAA4A0   JZ ZBOEND
B835 23       INX H
B836 46       MOV B,M
B837 23       INX H
B838 CDB640   CALL SAVER
B83B C0AE42   CALL MUSIC
B83E CDC640   CALL LOADR
B841 C32FB88  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 1008190213 DB 16,11,25,2,19,2,16,11,25,2,19,2
B886 1008190213 DB 16,11,25,2,19,2,16,6,19,2,16,6
B892 1302100B19 DB 19,2,16,11,25,4,0FFH
;
PRGEND:
;
B899          END

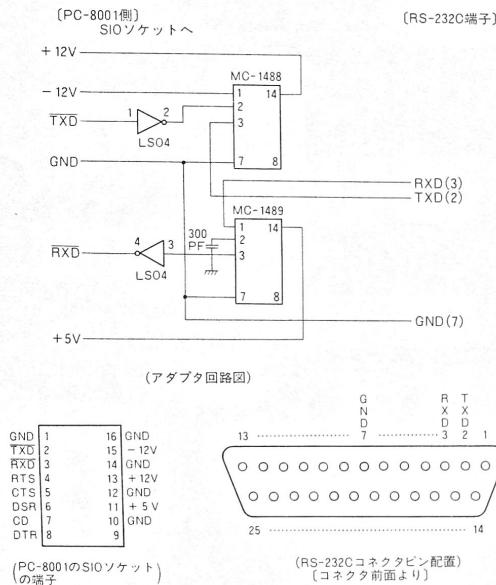
```

## 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分周で2400ポートになるように、PC-8001では内部のジャンパーで、8001mkIIでは後面のジャンパ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



### 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)

```
; !TRANS!
; FOR PC-8001/PC-8001 MK2
;
0100    ORG 100H
;
0005 =   B005 EQU 5
0000 =   B000 EQU 0
005C =   FCB EQU 005CH
0080 =   BUFFR EQU 0080H
00CE =   MODE EQU 0CEH ;$16
0020 =   PORT0 EQU 20H
0021 =   PORT11 EQU PORT0+1
;
;
0100 21CF12      LXI H,AREA
0103 22CB02      SHLD BPOINT
0106 0DA402      CALL TDATA
;
0109 3AB000      LDA BUFFR
010C 11B101      LXI D,MERR0
010F B7          ORA A
0110 CAC501      JZ EXIT
;
0113 0E0F        OPEN FILE
0115 115C00      MVI C,15
0115 115C00      LXI D,FCB
```

0118 CD0500	CALL	B00S		0170 E1	POP	H				
011B FEFF	CPI	0FFH		0171 010600	LXI	B,B				
011D 113902	LXI	D,MSOPNER		0174 09	DAD	B				
0120 CAC501	JZ	EXIT		0175 C1	POP	B				
;										
LOOP0:										
0123 CD5202	CALL	READF		0176 CD0502	CALL	THEX				
0126 C24701	JNZ	EOF		0179 12	STAX	D				
0129 2ACB02	LHLD	BPOINT		017A 13	INX	D				
012C EB	XCHG			017B 05	DCR	B				
012D 218000	LXI	H,BUFFR		017C C27601	JNZ	TLOOP1				
0130 0E80	MVI	C,128		017F 23	INX	H				
LOOP1:										
0132 7E	MOV	A,M		0180 23	INX	H				
0133 12	STAX	D		0181 23	INX	H				
0134 23	INX	H		0182 23	INX	H				
0135 13	INX	D		0183 C35501	JMP	TLOOP0				
0136 0D	DCR	C		CNVEND:						
0137 C23201	JNZ	LOOP1		0186 2ACD02	LHLD	COUNT				
013A 018000	LXI	B,128		0189 011600	LXI	B,22				
013D 2ACB02	LHLD	BPOINT		018C 09	DAD	B				
0140 09	DAD	B		018D 22CD02	SHLD	COUNT				
0141 22CB02	SHLD	BPOINT		0190 22D202	SHLD	AREA2+3				
0144 C32301	JMP	LOOP0		0193 22DC02	SHLD	AREA2+13				
;										
EDF:										
0147 0E10	MVI	C,16 ;FILE CLOSE		0196 1E00	MVI	E,0				
0149 115C00	LXI	D,FCB		0198 2ACD02	LHLD	COUNT				
014C CD0500	CALL	B00S		0198 E5	PUSH	H				
;										
014F 21CF12	LXI	H,AREA		019C C1	POP	B				
0152 11E502	LXI	D,AREA2+22		019D 21CF02	LXI	H,AREA2				
TLOOP0:										
0155 7E	MOV	A,M		SLOOP:						
0156 FE3A	CPI	'.'		01A0 7B	MOV	A,E				
0158 C2CB01	JNZ	ERROR1		01A1 86	ADD	M				
0158 23	INX	H		01A2 23	INX	H				
015C CD0502	CALL	THEX		01A3 0B	DCX	B				
015F 47	MOV	B,A		01A4 5F	MOV	E,A				
0160 B7	ORA	A		01A5 7B	MOV	A,B				
0161 CAB601	JZ	CNVEND		01A6 B1	ORA	C				
0164 C5	PUSH	B		01A7 C2A001	JNZ	SLOOP				
0165 E5	PUSH	H		01AA 7B	MOV	A,E				
0166 2ACD02	LHLD	COUNT		01AB 2F	CMA					
0169 0600	MVI	B,0		01AC C601	ADI	I				
016B 4F	MOV	C,A		;						
016C 09	DAD	B		01AE 32D102	STA	AREA2+2				
016D 22CD02	SHLD	COUNT		;						
;										
01B1 119102	LXI	D,MS000		01B4 CD2202	CALL	PRMSG				
01B7 0E01	MVI	C,1								

```

01B9 CD0500    CALL  BDOS
01BC CD5C02    ;CALL  SINIT
01BF CD7C02    ;CALL  SIO
01C2 112702    ;LXI   D,DONMSG
01C5 CD2202    ;CALL  PRMSG
01CB C30000    ;JMP   BOOT
;ERROR1:
01CB 11E701    LXI   D,MSERR1
01CE C3C501    ;JMP   EXIT
;MSERRO:
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   OFOH
020F 4F        MOV   C,A
0210 23        INX   H
0211 7E        MOV   A,M
0212 CD1A02    CALL  HEX0
0215 E60F    ANI   OFH
0217 B1        ORA   C
0218 23        INX   H
0219 C9        RET
;HEX0:
021A D630    SUI   '0'
021C FE0A    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 027202    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

```

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/Aセントラルを含んでいます。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。

```

MS000:   ; *RC-20 READY OK? ', '$'
DB      ; DATA:    H,DATA
02A4 21B502 LXI D,AREA2
02A7 11CF02 LXI C,22
02AA 0E16 MWI
; TLOOP:    MOV A,M
02AC 7E MOV D
02AD 12 STAX D
02AE 23 INX H
02AF 13 INX D
02B0 0D DCR C
02B1 C2A102 TLOOP
02B4 C9 RET
; DATA:    DB   I
02B5 01 DB   1
02B6 01 DB   1
02B7 00 DB   0
02B8 0000 DW   0
02B9 16 DB   22
02B8 00000000 DB   0,0,0,0
;           ; DB   2
02Bf 02 DB   22
02C0 1600 DW   0
02C2 0000 DW   0
02C4 00 DB   0
02C5 0000 DW   0
02C7 00000000 DB   0,0,0,0
; BPOINT:  DW   0
02C8 0000 COUNT: DW   0
02CD 0000 ; AREA2: DS   0
; AREA:    DS   EEU AREA2+1000H
02CF = 12CF ; END
02CF

```

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なります。

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と内容は同一です。

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。  
転送すべきデータのほうに行かないときは

- 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?"の表示が出来ます。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”的表示が出て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キーとハネルキーを使つて"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 100,1H8FFF
210 DEF USR=1H9000
800 FOR AD=$H9000 TO $H9087
810 READ DA$
820 POKA,AD,VAL(*&H*+DA$)
830 NEXT
900 INPUT "r=c-20 ready ok";X$
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 21,03,21,03,21,3E,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 21,00,BB,7E,CD,22,90,23,08,78,B1,20,F6,C9,1E,00
1104 DATA ED,4B,03,BB,21,00,BB,78,B6,23,0B,5F,78,B1,20,F7
1105 DATA 7B,2F,C6,01,32,02,BB,C9,2A,24,BB,01,16,00,09,22
1106 DATA 75,90,2B,22,7F,90,21,72,90,11,00,BB,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
BB00 AREA2: EQU 0B800H
00CE MODE: EQU 0CEH
0020 PORT0: EQU 20H
0021 PORT1: EQU PORT0+1
BB03 COUNT: EQU AREA2+3
BB2A 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,15H
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 ED4B03B8 S10: LD BC,(COUNT)
9030 2100BB LD HL,AREA2
9033 7E S102: 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,S102
903D C9 RET
;
903E 1E00 SUMC: LD E,0
9040 ED4B03B8 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 78 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 2A2AB8   ; TDATA: LD HL, (HADDR)
905B 011600   LD BC,22
               ADD HL,BC
               LD (DATA1),HL
               DEC HL
               LD (DATA2),HL
               LD HL,DATA
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9066 217290   ;
               LD HL,DATA
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9072 01      ; DATA: DB 1
9073 01      DB 1
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9074 00      ; DATA: DB 0
9075 0000    ; DATA: DW 0
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9077 16      ; DATA: DB 22
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9078 00000000; DATA: DB 0,0,0,0
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
907C 02      ; DATA: DB 2
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
907F 0000    ; DATA: DW 0
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9081 00      ; DATA: DW 0
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9082 0000    ; DATA: DW 0
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9084 00000000; DATA: DW 0,0,0,0
               LD DE,AREA2
               LD BC,22
               LDIR
               RET
9088 00      ; END

```

リスト 7 RC-20用マシン語プログラム(例 1)

```

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

PORG: EQU 08816H  

ZBOEND: EQU 409AH  

SAVER: EQU 4086H  

LOADR: EQU 40C6H  

MUSIC: EQU 424EH  

; ORG PORG
B816 00      ; MAIN: DB 0
               DB 'MUSIC'
               DB 'PLAY'
B817 4D555349
B818 43202020
B81F 2020504C

```

```

BB23 4159      ENTRY
BB25 1805      JR 0
BB27 00      DB 0
BB28 1688      DB MAIN
BB2A 8200      DB PRGEND-MAIN
BB2C          ; ENTRY:
BB2C 2143BB      LD HL,DATA
BB2F 7E      LOOP: LD A,(HL)
BB30 FFFF      CP OFFH
BB32 C9A40     JP Z,780END
BB35 23      INC HL
BB36 46      LD B,(HL)
BB37 23      INC HL
BB38 CDB640     CALL SAVER
BB3B CDAE42     CALL MUSIC
BB3E CD1640     CALL LOADR
BB41 18EC     JR LOOP
BB43          ; DATA: DS 0
BB43 0C060C02     DB 12,6,12,2,12,6,14,2,16,6,16,2
BB47 0C060E02     DB 12,6,12,2,12,6,14,2,16,6,16,2
BB4B 10061002     DB 16,6,16,2,14,6,12,2,14,6,16,2
BB53 0E160002     DB 12,6,12,2,7,8,16,6,16,2,16,6,17,2,19,6,19,2
BB57 0E161002     DB 12,6,12,2,7,8,16,6,16,2,16,6,17,2,19,6,19,2
BB5B 0C060C02     DB 12,6,12,2,7,8,16,6,16,2,16,6,17,2,19,6,19,2
BB5F 07081006     DB 12,6,12,2,7,8,16,6,16,2,16,6,17,2,19,6,19,2
BB63 10021006     DB 12,6,12,2,7,8,16,6,16,2,16,6,17,2,19,6,19,2
BB67 11021306     DB 12,6,12,2,7,8,16,6,16,2,17,6,19,2
BB6B 1302        DB 12,6,12,2,7,8,16,6,16,2,17,6,19,2
BB6B 13061302     DB 12,6,12,2,7,8,16,6,16,2,17,6,19,2
BB71 11061002     DB 12,6,12,2,7,8,16,6,16,2,17,6,19,2
BB75 11061302     DB 12,6,12,2,7,8,16,6,16,2,17,6,19,2
BB79 100B1902     DB 16,11,25,2,19,2,16,11,25,2,19,2
BB7D 1302100B     DB 16,11,25,2,19,2,16,11,25,2,19,2
BB81 19021302     DB 16,11,25,2,19,2,16,11,25,2,19,2
BB85 10091902     DB 16,11,25,2,19,2,16,11,25,2,19,2
BB89 13021006     DB 16,11,25,2,19,2,16,11,25,2,19,2
BB9D 13021006     DB 16,11,25,2,19,2,16,11,25,2,19,2
BB91 13021008     DB 16,11,25,2,19,2,16,11,25,2,19,2
BB95 1904FF     DB 16,11,25,2,19,2,16,11,25,2,19,2

```

## リスト 8 RC-20マシン語プログラム(例 2)

### 3. PC-8801/8801mkIIの場合

```
B8D3 0B060404          DB  82H,0FBH,84H,8CH,8CH,8CH,84H,0FBH,82H
B8D7 04          DB  4,4,4,8,8,70H,0B0H,0A0H,0A0H,0C0H
B8D8 B2FB848C          DB  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-8801/8801mkII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**

```
B8E0 FB82          DB  80H,0,0,0,0,0,0,0,1CH,22H
B8E2 04040408          DB  N88-BASICモードで立て上げて実行するCP/Mでも、先の「2.PC-8801/8801mkII」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**

```
B8F0 00000000          DB  転送プログラムは、N88-BASICで作ります(リスト9)。
B8F4 0000001C          DB  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)
B8F8 22          DB  と"GRAPH"(リスト8)を、そのまま使用します。
B8F9 2B404640          DB  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
B8FD 44472428A         DB  GRAPH.HEXを作成します。
B901 B2820204          DB  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
B905 04040404          DB  のローダではB16H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、OFF
B909 0402928A          DB  SETとして1000Hを指定し、C816H～C897Hにロードします。
B90D 9E705246          DB  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N
B911 40464026          DB  88-BASICにもどるので、
B915 22          DB  BSAVE "KOHAN.BIN",&HC816,&H82
B916 1C000000          DB  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。
B91A 00          DB  "GRAPH.BIN"を作成します。
B91B 0000          DB  RC-20への転送は、N88-BASICモードで、
B91D          PRGEND:          END
B91D          END
```

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。転送が始まつて、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」のモードに入ります。

#### リスト 9 転送プログラム (TRANS)

```

100 REM $trans†
110 CLEAR ,&H000
120 INPUT "file name":F$
130 BLOAD F$*
140 FOR I=0 TO 21
150 POKE &H000+I,0
160 NEXT I
170 POKE &H004,1
180 POKE &H001,1
190 POKE &H005,22
200 POKE &H004,2
210 POKE &H005,22
220 I=PEEK(&H002B)
230 I=I+256
240 J=PEEK(&H002A)
250 I=I+J+22
260 J=I
280 ZL=J MOD 256
290 ZH=INT(J/256)
300 POKE &H003,ZL
310 POKE &H004,ZH
320 J=J-1
330 ZL=J MOD 256
340 ZH=INT(J/256)
350 POKE &H003,ZL
360 POKE &H004,ZH
370 CHK=0
380 FOR J=0 TO I
390   CHR$=CHK+PEEK(&H000+I)
400 NEXT J
410 A$=CHR$-INT(CHR$/256)*256
420 A$=Z55-A$#
430 POKE &H002,A$#
440 REM
450 OPEN "COM1:N82N" FOR OUTPUT AS #1
460 INPUT "rc-20 ready ok":X$#
470 FOR K=0 TO 1
480   A=PEEK(&H000+K)
490   PRINT #1,CHR$(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(ボーレート)は、カード上のディップ・スイッチで指定します。slowロック(64分周)で、2400ボート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。

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、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モードに切り替えて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。

```

{(一) SAVEM "CAS0:KOHAN",&H5016,&H5097,&H0000
{(一) SAVEM "O:KOHAN",&H5016,&H5097,&H0000
{(一) SAVEM "CAS0:GRAPH",&H5016,&H511C,&H0000
{(一) SAVEM "O: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

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:"**

```
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 B8 82 00 21 A3 B8 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 40 20 18 05 00 16 B8 07 01 21 74 B8 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 01 02
5080 07 04 0E 18 23 34 23 20 23 34 23 18 0E
5090 04 07 02 01 00 00 00 00 00 00 00 00 00 00 00
50A0 00 00 00 00 3A 6A AA BF AO EO 00 04 15 CE 20
50B0 A4 CE 0E CE AA 20 CE 15 04 00 00 EO AO BF AA
50C0 6A 34 0F 00 00 00 00 00 00 00 00 00 00 00 00
50D0 A0 B0 70 08 08 04 04 B2 FB 84 BC BC B6 B4
50E0 FB 82 04 04 04 08 08 70 B0 AO CO 80 00 00 00
50F0 00 00 00 00 00 00 00 00 00 00 00 00 46 42 42
5100 BA B2 B2 02 04 04 04 04 04 04 02 92 BA 9E 74 52
```

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11 転送プログラム (TRANS)

```
10 REM trans†
20 CLEAR $HFFFF
30 INPUT "file name"; F$
40 LOAD F$
50 GOSUB 180
60 GOSUB 270
70 OPEN "U", #1, "D00: ($R2)"†
80 INPUT "$rc=20 ready ok"; X$†
90 FOR K=0 TO 1
100 A=FEEK($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
200 NEXT I
210 POKE $H5000, 1
220 POKE $H5001, 1
230 POKE $H5005, 22
```

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になっています）、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、行番号70のOPEN文のオプション (S8N2) は、slowロック (64分周)、データ長

```

240 POKE $4500A,2
250 POKE $H500B,22
260 RETURN
270 I=FEK &H502B)
280 I=1736
290 J=FEK &H502A)
300 I=I+1-I+22
310 J=I+1
320 L=j MOD 256
330 TH=INT(J/256)
340 POKE $H5003,7L
350 POKE $H5004,7H
360 J=J-1
370 L=j MOD 256
380 TH=INT(J/256)
390 POKE $H5000,7L
400 POKE $H500E,7H
410 CHK=0
420 FOR J=0 TO 1
430 CHK=CHK+FEER (&H5000+J)
440 NEXT J
450 A$=CHK-INT(CHK/256)*256
460 A$=A$-A$%
470 POKE $H5002,A%
480 RETURN

```

## 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&OC16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”的なプリリストを示します。  
 “KOHAN”的場合は&H0C16~0C97に、“GRAPH”的場合は&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”的例で説明すると、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、データ上の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に転送するデーターファイル名を指定します。“KOHAN”的場合なら、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”的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”的メッセージを出すようになります。  
 そこで、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。

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します。

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2 K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、

## 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ます。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のモニタにダイレクトに入力することにします。

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2 K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。



6. HC-40/80/880



```

3080 020A 0E09 LD C,9
3100 020C C0500 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 TOP 4016 MOVEF 0140
TDATA 01A5 SUNC 018C MS001 012C
MESS 020A SID 01D5 MSEND 011D
EXIT 01F2 MSOPNE 01FB LOOP0 0158
READF 0182 EOF 0179 BPOINT 0210
ASMB80(Z80) 0: 0: 0/ 2: 1:50 4

LOOP1 0166 SLOOP 0195 DATA1 01C2
DATA2 01CC DATA 01BF SID2 01DC
OUTPUT 01E7

```

リスト15

```

D>b:asmb80 kohan.asm
ASMB80(Z80) 0: 1: 2/ 1:19:14 1
100 0000 ; *KOHAN*
120 0000 ;
140 B816 MAIN EQU 0B816H
160 409A Z80END EQU 409AH
180 4086 SAVER EQU 4086H
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 60757369 DB 'music play'
380 010F 1B05 JR ENTRY
400 0111 00 DB 0
420 0112 16BB DW MAIN
440 0114 8200 DW PRGEND-START
460 0116 ;
480 0116 ENTRY:DS 0
500 0116 2143B8 LD HL,DATA-START+MAIN
520 0119 7E LOOP:LD A,(HL)
540 011A FEFF CP OFFH
560 011C CA9A40 JP Z,Z80END
580 011F 23 INC HL
600 0120 46 LD B,(HL)
620 0121 23 INC HL

```

```

640 0122 CDB640 CALL SAVER
660 0125 CDAE42 CALL MUSIC
680 0128 CDC640 CALL LOADR
700 012B 18EC JR LOOP
720 012D ;
740 012D DATA: DS 0
760 012D 0C060C02 DB 12,b,12,2,12,b,14,2,16,b,16,2
780 0139 10061002 DB 16,b,16,2,14,b,12,2,14,b,16,2
800 0145 0C060C02 DB 12,b,12,2,7,B,16,b,16,2,16,b,17,2,19,b,19,2
820 0157 13061302 DB 19,b,19,2,17,b,16,2,17,b,19,2
840 0163 10081902 DB 16,11,25,2,19,2,16,11,25,2,19,2
860 016F 100B1902 DB 16,11,25,2,19,2,16,b,19,2,16,b
880 017B 1302100B DB 19,2,16,11,25,4,0FFH
900 0182 ;
920 0182 PRGEND: DS 0
940 0182 ;
960 0182 END
MAIN 8816 Z80END 409A SAVER 4086
LOADR 40C6 MUSIC 42AE START 0100
ENTRY 0116 PRGEND 0182 DATA 012D
LOOP 0119

```

リスト16

```

A>b:asmb80 graph.asm,graph.com
ASMB80(Z80) 0: 0: 0/ 1:10:42 1
100 0000 ; $GRAPH$
110 0000 ;
120 B816 MAIN EQU 0B816H
130 BFBA VRAMP EQU 0BFBAAH
140 002A PAGEL EQU 2AH
150 411C DISP EQU 411CH
160 41CB STBY EQU 41CBH
170 409A Z80END EQU 409AH
180 0000 ;
190 0100 ORG 100H
200 0100 ;
210 0100 ;MAIN:
215 0100 START: DS 0
220 0100 00 DB 0
230 0101 47524150 DB 'GRAPH PROGRAM '
240 010F 1B05 JR ENTRY
250 0111 00 DB 0
260 0112 16BB DW MAIN
270 0114 0701 DW PRGEND-START
280 0116 ;
290 0116 2174B8 ENTRY: LD HL,MDATA-START+MAIN
300 0119 11BABF LD DE,VRAMP
310 011C 012A00 LD BC,PAGEL

```



```

340 0119 0E01 LD C,1
810 018E 115C00 LD DE,FCB8
350 0118 C0D500 CALL 800S
820 0191 C00500 CALL 800S
830 0194 B7 OR A
840 0195 C9 RET
850 0196
860 0196
870 0196 LD E,0
880 0198 ED4B0340 LD BC,(COUNT)
890 019C
900 019C
910 019C 210040 LD HL,ARE
920 019F
930 019F TB LD A,E
940 01A0 86 ADD A,(HL)
950 01A1 23 INC HL
960 01A2 88 DEC BC
970 01A3 5F LD A,B
980 01A4 78 LD A,B
990 01A5 B1 OR C
1000 01A6 20F7 LD NZ,SL0OP
1020 01A9 ED4A LD A,E
1030 01AB NE6
1050 01AC C9 RET
1060 01AF
1070 01AF TDATA:D5 0
1080 01AF 2A2A40 LD HL,(HADDR)
1090 01B2 011600 LD BC,22
1100 01B5 99 ADD HL,BC
1110 01B6 22CC01 LD (DATA1),HL
1120 01B9 28 DEC HL
1130 01B9 22D601 LD (DATA2),HL
1140 01B9 LD
1150 01B9 21C901 LD HL,DATA
1160 01C0 110040 LD DE,ARE
1170 01C3 011600 LD BC,22
1180 01C6 ED80 LDTR
1190 01C8 99 RET
1200 01C9
1210 01C9 DATA:D5 0
1220 01C9 01 DB 1
1230 01CA 01 DB 1
1240 01CB 00 DB 0
1250 01CC DATA:D5 0
1260 01CC 0000 DM 0
1270 01CE 16 DB 22
790 018E LD C,16 ,CLOSE.
780 0185 115C00 LD DE,FCB8
CALL 800S
740 0183 0E10 LD C,16
750 0185 115C00 LD DE,FCB8
760 0188 C0D500 LD DE,FCB8
770 0188 99 RET
780 018E LD C,16 ,CLOSE.
790 018C 0E14 LD C,20
READF:D5 0
790 018C LD (PPPOINT),HL
800 018C 0E14 LD C,20

```

```

1280 01CF 00000000      DB    0,0,0,0
1290 01D3      ;      DB    2
1300 01D3 02      DB    22
1310 01D4 1600      DM    0
1320 01D6      DATA2:DS 0
1330 01D6 0000      DW    0
1340 01D8 00      DB    0
1350 01D9 0000      DW    0
1360 01DB 00000000      DB    0,0,0,0
1370 01DF      ;      ST0:DS 0
1380 01DF 0610      LD    B,10H ;RS232C OPEN
1390 01DF 215702      LD    HL,PAR1ST
1400 01E1 CD6202      CALL  RS10X
1410 01E4 CD6202      CALL  JRR
1420 01E7 2047      NZ,RERR
1430 01E9      ;      LD    RC,(COUNT)
1440 01E9 ED4B0340      LD    HL,AREA
1450 01ED 210940      LD    0
1460 01F0      ST02:DS 0
1470 01F0 7E      LD    A,(HL)
1480 01F1 CD0002      CALL  OUTPUT
1490 01F4 23      INC   HL
1500 01F5 08      DEC   BC
1510 01F6 78      LD    A,B
1520 01F7 B1      ORR  C
1530 01FB 20F6      JR    N7,S102
1540 01FA 0420      LD    B,20H ;RS232C CLOSE
1550 01FC CD6202      CALL  RS10X
1560 01FF      ;      RET
1570 01FF C9      ;      RET
1580 0200      ;      OUTPUT:DS 0
1590 0200      ;      PUSH  HL
1600 0200 E5      PUSH  BC
1610 0201 C5      PUSH  AF
1620 0202 F5      LD    HL,PAR1ST
1630 0203 215702      LD    B,40H ;CHECK SEND STATUS
1640 0206 0640      LD    HL,PAR1ST
1650 0208 CD6202      CALL  RS10X
1660 0208 2023      JR    NZ,RERR
1670 0200 B7      ORR  A
1680 020E 2BF6      JR    Z,S103
1690 0210 F1      POP   AF
1700 0211 4F      LD    C,A
1710 0212 0660      LD    B,60H ;SEND DATA
1720 0214 CD6202      CALL  RS10X
1730 0217 2017      JR    NZ,RERR
1740 0219 C1      POP   BC
1750 021A E1      POP   HL
1760 021B C9      RET
1770 021C      ;      MSERR:DB 13,10,'#rs232c error$',13,10,'$,
1780 021C 000A0202A      ;      1790 0230
1800 0230 111C02      RERR:LD DE,MSERR
1810 0233 C05102      CALL  MESS
1820 0236 C30000      JP    BOOT
1830 0239      ;      EXIT:DS 0
1840 0239 CD5102      CALL  MESS
1850 0239 000A0202A      ;      1860 023C C30000
1870 023F      ;      MSOPNER:DS 0
1880 023F 000A0202A      ;      1890 0251
1900 0251 0E09      DB    13,10,'open error$',13,10,'$,
1910 0253 C00300      LD    C,9
1920 0256 C9      CALL  BDOS
1930 0257      ;      RET
1940 0257 6602      DW    BUFFR2
1950 0259 00008      DW    2048
1960 025B 0C      DB    OCH
1970 025C 03      DB    3
1980 0250 00      DB    0
1990 025E 03      DB    3
2000 025F FF      DB    OFFH
2010 0260      ;      2020 0260
2030 0260 1640      DW    TOP
2040 0262 CD0000      RS10X:CALL 0000H
2050 0265 C9      RET
2060 0266      ;      BUFFR2:DS 2048
2070 0266      ;      END
2080 0A66      BDD5 0005 BOOT 0000 FCB 005C
2090 0A66      BDF5 0005 BOOT 4000 AREA 4000 COUNT 4003
2100 0A66      BDF5 0005 AREA 4000 COUNT 4003
2110 0A66      BDF5 0005 HADR 402A TOP 4016 RS10X 0262
2120 0A66      BDF5 0005 MOVEF 014A TDAT 01AF SUMC 0196
2130 0A66      BDF5 0005 HSEND 0127 EXIT 0239 NSOPNE 023F
2140 0A66      BDF5 0005 LOOPQ 0162 READF 018C EOF 0183
2150 0A66      BDF5 0005 BPOINT 0260 LOOP1 0170 SLOOP 019F
2160 0A66      BDF5 0005 DATA1 01CC DATA2 0106 DATA 01C9
2170 0A66      BDF5 0005 PARLIS 0257 REERR 0230 S102 01F0
2180 0A66      BDF5 0005 OUTPUT 0200 S103 0206 MSERR 021C
2190 0A66      BDF5 0005 BUFFR2 0266

```

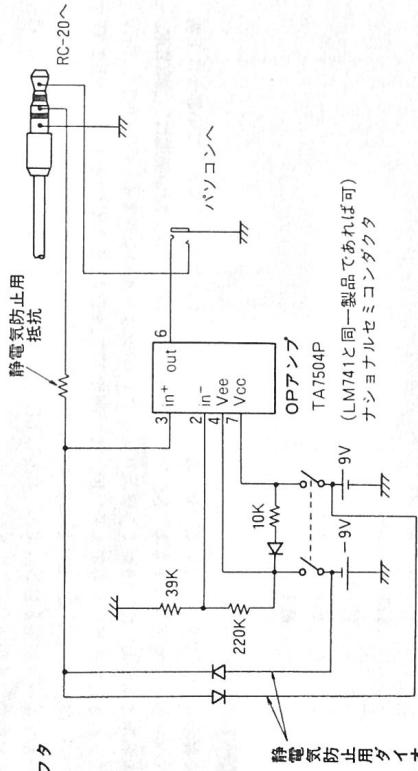
RC-20

NUMBER  
3

# RC-20からパソコンへデータの転送

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トは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。レベルシフトを自作される方のために参考例を図3-1に示します。

図3-1  
レベルシフト



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	ボーレート
	2400ボードときは08H

ボーレートは、2400、1200、600、300ボードの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ボード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内サブルーンRSOUTを使用しますが、このサブルーンは、パ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、パラメータによつ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、次の転送規約に対応しています。

- ① 出力データの前に「2 AH」(\*)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、データの先頭から2AH('\*)をあらかじめ除去おけば有効ですが、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、十分な対策とはいえません。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 OC-10の場合

CP/M上で実行することになります。  
まず、サンア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1KB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。

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。

RC-20のROM内サブルーチンRSOUTを使ってますが、このRSOUTで直接転送できるのは、  
出力がボリュームとRC-20は内部メッセージを出してへりつけになります。RC-20からの転送は

RAM上のデータに反映されています。このため、ROMの内容を1Kバイトだけ、いったんRAM

に転送しながら、それを RSOU II と RS 232C 間にしているのである。また、RSOU II は IK-BAT のデータを RS-232C に出力するとき、この IK-BAT の後に ±22.5° のアングルを加算する。

で出力しているので、ROM の内容を、連続して上記の方法で転送すると、IK バイトの ROM を得られる。

情報と次の1KB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に出力されていきます。そのためサンプルプログラム一時的にモードをRS-232Cに切り替えて、

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

各納します。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f1.Q-C-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'の表示が出来ますので、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「CRJKEYを押すと、RC-20の画面に、「MODE」を指定する数字のKEY表示がるので、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

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

		JN2	LDP1
B83B C238B8			
B83E 21EE88	LXI	H,DATA2	
B841 11EF8F	LXI	0,DISPB	
B844 010700	LXI	B,7	
	LDIR		
B847+EDB0	DB	.0ETH,0B0H	
B849 CD1A42	CALL	HY00	
	;		
	LOOPY:		
	CALL	KD1SP3	
B84F CD9040	CALL	GETKEY	
	;	LOOPY	
B852+28F8	DB	28H,LOOPY-\$-1	
B854 FE01	CPI	1	
B856+281E	DB	LN00	
B858 FE02	CPI	2	
	;	28H,LN00-\$-1	
B85A+2821	DB	LN01	
B85C FE03	CPI	3	
	;	28H,LN01-\$-1	
B85E+2824	DB	LN02	
B860 FE04	CPI	4	
B862+2827	DB	LN03	
B864 FE05	CPI	5	
B866+282A	DB	LN04	
B868 FE06	CPI	6	
	;	28H,LN04-\$-1	
B86A+282D	DB	LN05	
B86C FE11	CPI	7	
B86E+2830	DB	LN06	
B870 FE12	CPI	1H	
B872+2833	DB	LN07	
	;	28H,LN06-\$-1	
B874+18D6	DB	LN08:	
B876 3E30	MVI	A,'0'	
B878 210040	LXI	H,4000H	
	;	LOOP2	
B87D 3E31	JR	DB	18H,LOOP2-\$-1
	;		
	;		
B882+1828	LN02:	DB	
B884 3E32	MVI	A,'2'	
B886 210048	LXI	H,4000H	
	;	LOOP2	
B889+1821	LN03:	MVI	A,'3'
B88D 21004C	LXI	H,4000H	
	;	LOOP2	
B890+181A	LN04:	MVI	A,'4'
B892 3E34	LXI	H,5000H	
B894 210050	JR	LOOP2	
B897+1813	LN05:	DB	18H,LOOP2-\$-1
B899 3E35	MVI	A,'5'	
B89B 210054	LXI	H,5000H	
	;	LOOP2	
B89C+180C	LN06:	MVI	A,'7'
B8A0 3E36	LXI	H,5800H	
B8A2 210058	JR	LOOP2	
B8A5+1805	LN07:	DB	18H,LOOP2-\$-1
B8A7 3E37	MVI	A,'8'	
B8A9 21005C	LXI	H,5000H	
	;		
B8AC 22F5B8	LOOP2:	SHLD	WORK
B8AF 32EABF		STA	DISPB+b
B8B2 C01A42		CALL	HY06
B8B5 24F5B8		LHLD	WORK
B8B8 11F7B8		LXI	D,BUFF
B8BB 010004		LXI	B,0400H
	;	LDIR	
B8BE+EDB0	DB	0EDH,0B0H	
B8C0 C0C6B8	CALL	OUTPUT	
B8C3 C25B8	;	JMP	START
	;	OUTPUT:	
B8C6 3E08	MVI	A,BAUD	
B8C8 21F7B8	LXI	H,BUFF	

```

B8CB 11F6BC    LXI   D,BUFF+0400H-1
B8CE CD2043    CALL  RSOUT
B8D1 C9        RET

; DATA1:
B8D2 2A44415441 DB    '*DATA TRANSFER START OK? '
DATA2:
B8EE 2A4D4F4445 DB    '*MODE= '
; WORK: DW 0
BUFF: DS 0
; B8F7
; B8F7 END

```

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

```

; *INRC*
; FOR OC-10.
;
0100    ORG 100H
;
F654 =    RSIOX EQU 0F654H
0005 =    BDOS EQU 5
0000 =    BOOT EQU 0
9000 =    AREA EQU 9000H
;
0100 118C01  LXI D,MSXXX
0103 CD6301  CALL MESS
;
; OPEN RS232C.
0106 0610    MVI B,10H
0108 21EE01  LXI H,PARLST
0108 CD54F6    CALL RSIOX
010E C25A01  JNZ RERR
;
0111 110090  LXI D,AREA
;
SIOIN:
; *CHECK RECEIVER.
0114 D5    PUSH D
0115 E5    PUSH H
0116 0630  MVI B,30H
0118 CD54F6  CALL RSIOX
011B E1    POP H
011C D1    POP D

```

```

011D 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 7B    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 CA1401  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 PSW
0152 0E02  MVI C,2
0154 5F    MOV E,A
0155 CD0500  CALL BDOS
0158 F1    POP PSW
0159 C9    RET
RERR:
015A 116901  LXI D,MSERR
MSEND:
015D CD6301  CALL MESS
0160 C30000  JMP BOOT

```

420 MUSIC书711-手ノ(42AEH)⑨進丁也ニテルノ

0163 0E09	MVI	C,9	MESS:	
0168 C9	CALL	RDT		
0163 CD0500	BDS			
0169 0D0A20A52	DB	13,10,*	READ ERROR#	,13,10,*
0178 0D20	MVI	B,020H	CLSE	R5322C.
017D CD054F6	CALL	RSTDX		
0180 11C501	I1X1	d,wsunnn	MESSXX:	
0183 C6D301	C211	MESS		
0186 11B061	LXI	D,M5000		
0189 C23D01	JMP	MSEN0		
0196 0D0A20A52	DB	13,10,*	R5322C READER START#	,13,10,*
01A6 0D0A20A72	DB	13,10,*	READY OK#	,13,10
01B5 24	DB	*,*		
01B6 0D0A20A441	DB	13,10,*	DATA IS LOADED IN MEMORY FROM 9000#	
01C5 202020441	DB	13,10,*	MSNNN:	
01E6 0D0A20A44	DB	13,10,*	PARLSI#	
01F0 00	DM	2048	BUFER	BUFER ADDRESS.
01F2 03	DB	3	8 BIT/CHAR.	
01F4 00	DB	0	NON PRACTY.	
01F5 03	DB	3	STOP BIT=2.	
01F6 FF	DB	0FFH		
01F7	DS	2048	BUFR#:	
01F8	;			
01F9	;			
01FA	;			
01FB	;			
01FC	;			
01FD	;			
01FE	;			
01FF	;			



54

5 HC

06.2

4

四

12

四

08-2

三

•

## 2. DUADPCO概念

1) A F 23 RC-204b-5의 車輛考 - 3) 車輛考 3) 日 3) A (INRC)

BBUD4 ZAR444154 DARA1; DB DURAIA TRANSFER STAR! UK? PDRP A130545Z

14

73



```

100 REM $inrc8
110 CLECR, $HC000
120 PRINT "Prebler start."
130 LOAD COM1:N8Z2*, $HC000
140 PRINT "Data is loaded in $HC000-$HC3FFF."
150 PRINT "Job end."
160 END

88C5 C32C8 ENTRY jp ENTRY
88C8 3E08 OUTPUT:L0 A,BAUD
88C9 21F8B LD H,L,UFF-4
88CD 11FABC LD DE,UFF+0400H-1
88D0 CD2043 CALL RSOUT
88D3 C9 RET
88D4 41205528 DDATA1: DB "DATA TRANSFER START OK?"
88D5 45522535 DDAT2: DB "MODE=",
88E0 45415254 DDAT3: DB 0,0,0,0
88E8 204FAFB3P
88F0 20202020P
88F4 453020P
88F7 00000004 XXXX: DB 0,0,0,0
88FB 88FB: DS 0
88FB PRGEND:
88FB END
;
```

### 3.4 HC-40/80/88の場合

40ページ「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「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\ss\B0 r=232c.asm
A\$B0(ZB0) 0: 1:17 1:10:14 1
;
; RS232C
;
ORG 100H
;
160 0100
;
180 8816
MAIN EQU 08816H
W1AL EQU 426AH
D1SP3 EQU 0BF4AH
HY061 EQU 4106H
HY00 EQU 421AH
K01SP3 EQU 4169H
KEY EQU 4090H
Z80END EQU 409AH
BAUD EQU 0BH ;2400 BAUD
RSOUT EQU 4320H
;
;MAIN:
;
420 0100
440 0100 00
460 0101 2A727337
480 010F 1805
500 0111 00
520 0112 16B8
540 0114 E100
560 0116
580 0118
600 0116 CD6A42
620 0119 21D48B
640 011C 11E4BF
660 011F 01C00
680 0122 ED80
700 0124 CD06A1
720 0127 CD9040
740 012A FE34
760 012C 20F9
780 012E
800 012E Z1F08B
820 0131 11E4BF
840 0134 010700
860 0137 EB80
880 0139 CD1A42
900 013C
920 013C CD6941
940 013C CD9040
960 0142 2BF8
980 0144 FE01
1000 0146 281E
1020 0148 FE02
1040 014A 2821
1060 014C FE03
1080 014E 2824
1100 0150 FE04
1120 0152 2827
1140 0154 FE05
1160 0156 282A
1180 0158 FE06
1200 015A 282D
1220 015C FE11
1240 015E 2830
1260 0160 FE12
1280 0162 2833
1300 0164 18D6
1320 0166 SE30
LN00:LD
;
```

```
420 0100
440 0100 00
460 0101 2A727337
480 010F 1805
500 0111 00
520 0112 16B8
540 0114 E100
560 0116
580 0118
600 0116 CD6A42
620 0119 21D48B
640 011C 11E4BF
660 011F 01C00
680 0122 ED80
700 0124 CD06A1
720 0127 CD9040
740 012A FE34
760 012C 20F9
780 012E
800 012E Z1F08B
820 0131 11E4BF
840 0134 010700
860 0137 EB80
880 0139 CD1A42
900 013C
920 013C CD6941
940 013C CD9040
960 0142 2BF8
980 0144 FE01
1000 0146 281E
1020 0148 FE02
1040 014A 2821
1060 014C FE03
1080 014E 2824
1100 0150 FE04
1120 0152 2827
1140 0154 FE05
1160 0156 282A
1180 0158 FE06
1200 015A 282D
1220 015C FE11
1240 015E 2830
1260 0160 FE12
1280 0162 2833
1300 0164 18D6
1320 0166 SE30
LN00:LD
;
```

```
START:DS 0
DB 0
;`rs232c`*
ENTRY 0
MAIN
PRGEND-START
;
;WH1AL
HL,DATA1-START+MAIN
DE,DISPB
BC,28
LDIR
HY061
L00P1:CALL GETKEY
CP 34H
JR NL,L00P1
;
LD HL,DATA2-START+MAIN
DE,DISPB
BC,7
LDIR
HY00
CALL HY00
;
;L00PY:CALL K01SP3
CALL GETKEY
JR L,LO0PY
L,L00PY
;
CP 1
L,LN00
;
CP 2
L,LN01
;
CP 3
L,LN02
;
CP 4
L,LN03
;
CP 5
L,LN04
;
CP 6
L,LN05
;
CP 11H
L,LN06
;
L,LN07
;
L,LN08
;
A,'0'
```

```

MAIN    BB16   WH1AL  426A  015PB  BFE4
HWG1   4106   HY00  421A  K015F3  4159
GETKEY 4090   ZB0END 409A  BAUD  0008
RSOUT  4320   START 0100  ENTRY  0116
PREEND 01E1  DATA1 01BE  LOOP1 0127
DATA2 01DA  LOOPY 013C  LN00  0166
LN01  016D  LN02  0174  LN03  017B
LN04  0182  LN05  0189  LN06  0190
LN07  0197  LOOP2 019C  BUFF  01E1
OUTPUT 0182

LD     HL,4000H
JR     LOOP2
LN01:LD
A,'1'
LD     HL,4400H
JR     LOOP2
LN02:LD
A,'2'
HL,4800H
LD     HL,5000H
JR     LOOP2
LN03:LD
A,'3'
HL,4C00H
LD     LOOP2
LN04:LD
A,'4'
HL,5400H
LD     LOOP2
LN05:LD
A,'5'
HL,5800H
JR     LOOP2
LN07:LD
A,'7'
HL,5C00H
LD     PUSH HL
LD     (DISPB+6),A
CALL   HY00
POP    HL
DE,BUFF-START+MAIN
LD     BC,0400H
LD     DIR
CALL   OUTPUT-START+MAIN
;      ENTRY-START+MAIN
;      OUTPUT;LD A,BAUD
LD     HL,BUFF-START+MAIN
DE,BUFF-START+MAIN+0400H-1
CALL   RSOUT
RET
;      RET
LD     HL,4000H
DATA1:DB  'data transfer start ok?'
DATA2:DB  ',baud='
BUFF:DS  0
;      PROEND:DS 0
;      END

```

リスト27 RCC-20からの転送データを受信するプログラム(INRC)

```

0:0:455E80 10FC.353
ASMBL(Z80) 0: 1: 2: 1:10:18
;
```

```

100 0000 ;$INRC
110 0000 ;
120 0100 ;ORG 100H
130 0100 ;
140 0005 ;BD05
150 0000 ;BOOT EQU 0
160 4000 ;AREA EQU 4000H
170 0100 ;
180 0100 118501 ;LD DE,MSXX
190 0103 C05D01 ;CALL HESS
200 0106 ;LD HL,(B00T+1)
210 0106 2A0100 ;LD BC,51H
220 0109 015100 ;ADD HL,BC
230 010C 09 ;(RSJ0X+1),HL
240 0100 22EE01 ;LD
250 0110 ;LD B,10H ;OPEN RS232C
260 0110 0610 ;LD HL,PARLLIST
270 0112 21E401 ;CALL RSJ0X
280 0115 C0ED01 ;JR NL,RERR
290 0118 203A ;CALL
300 011A ;LD DE,AREA
310 011A 110040

```

```

320 011D D5
330 011D DE
340 011E E5
350 011F 0630
360 0121 CED01
370 0124 E1
380 0125 D1
390 0126 202C
400 0128 B7
410 0129 2B14
420 012B
430 012B C5
440 012C DS
450 012D E5
460 012E 0650
470 0130 CED01
480 0133 E1
490 0134 D1
500 0135 C1
510 0136 201C
520 0138 12
530 0139 13
540 013A 0B
550 013B 7B
560 013C B1
570 013D 20EC
580 013F
590 013F C5
600 0140 D5
610 0141 E5
620 0142 C04001
630 0145 E1
640 0146 D1
650 0147 C1
660 0148 B7
670 0149 2BD2
680 014B 1828
690 014D
700 014D 0E06
710 014F 1EFF
720 0151 C30500
730 0154
740 0154 116301
750 0157 C05001
760 0158 C30000
770 0159
780 0150 0E09

; RERR:LD DE, M$ERR
MESS:LD C,9

```

```

790 015F CD0500 CALL BDOS
800 0162 C9 RET
810 0163 000202A MSERR,DB 13,10, 'Read error', 13,10, '$'
820 0175 0620 ENDL:LD 2,20H ;RS232C CLOSE
830 0177 CDE001 CALL RS10X
840 017A ; LD DE,MSNNN
850 017A 11B601 LD MESS
860 017D CD5001 CALL DE,MS000
870 0180 11AF01 LD MSEND
880 0183 18D2 JR MSXX,DB
890 0185 0004202A DB 13,10, 'It's 232C reader start.', 13,10
900 019F 0004202A DB , 'Ready ok', 13,10, '$'
910 01AF 202B646F MS0000:DB , '$job end$', 13,10, '$'
920 01BC 202B6461 MSNNN:DB , 'data is loaded in memory from 40000H
930 01E4 PARLIST:DS 0
940 01E4 F101 DM BUFR
950 01EB 000B DB 2048
960 01EB OC DB 0CH
970 01E9 03 DB 3
980 01EA 00 DB 0
990 01EB 03 DB 3
1000 01EC FF DB 0FFH

```

# ハード構成 とモジュツ



図4-1

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個の固定キーのほかに液晶表示部に、 $6 \times 3 = 18$ 個の文字表示位置がタッチキー(ペネルスイッチ)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BFAAH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。

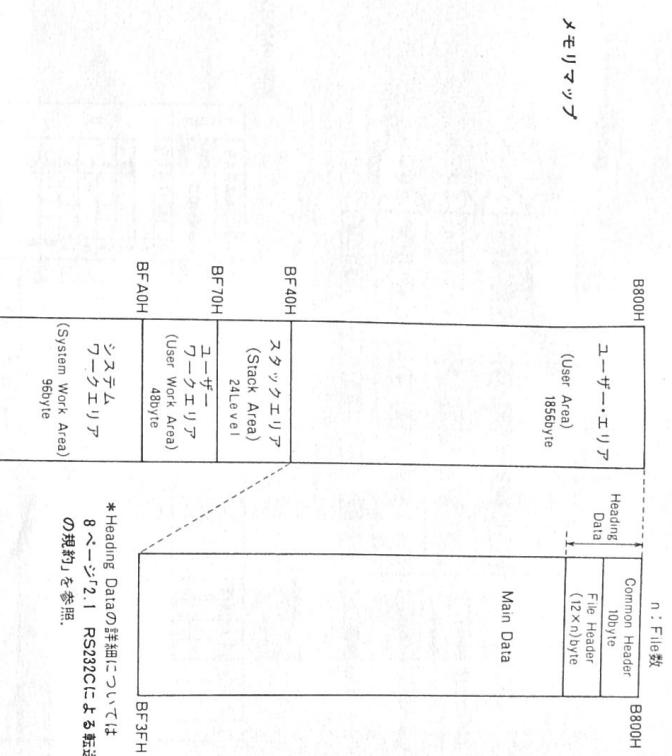
4ビット CPU	Time Standard CPU * 計時機能 * I/O処理機能
SMC84C00	Zilog Z-80 <sup>®</sup> コンパチ 8ビット CPU C-MOS版
2KB RAM	16Kbit C-MOS Static RAM (低電圧バージョン)
8KB ROM	64Kbit C-MOS Mask ROM (低電圧バージョン)
ドットドライバ A, B	Dot Driver (低電圧バージョン)

表4-1 RC-20機能概要

項目	内 容
1. コンピュータ	SMC84C00 (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 スケジューラ・メモ・ワールドタイム マシン語プログラム A/D・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）
アラーム機能	時分設定型 鳴鐘時間：20sec 鳴り止め機能 休止・復元機能 アラーム試験機能 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

図4-2

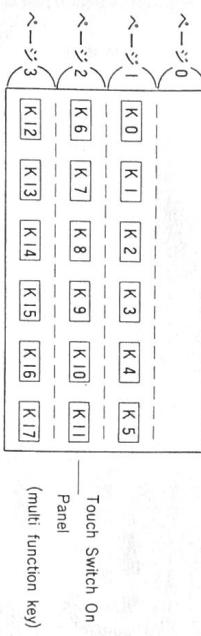
メモリマップ



## システムワークエリアの詳細

4. 作動温度範囲	0 ℃ ~ +50℃
5. 操作方式	タッチキー方式 ・固定キー • Touch Switch On Panel 5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	約1年 (コンピュータ20秒/1日鳴鐘 30分/1日使用)
使用電池	リチウム電池BR2325
9. スピーカ	圧電スピーカ
10. 表示	F/E型 白色液晶 表示方式 表示内容 表示寿命 42×32全面ドット (7文字×4行) 5年以上

図4-4 ドットとVRAMの対応



<sup>注</sup>) 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

Level	機能	Level	機能
FPPAD	File Data TOP ADDRESS	SAHL	Hlreg のスタックエリア INTSET
FEDAD	File Data END ADDRESS	KEYBU	KEY データ レジスター
FLNGT	Length of Record (スケジュールデータ & メモータファイル)	ALFLAG	I/60HZ JUMP TABLE
FRDCT	Numbers of Record (スケジュールデータ & メモータファイル)	INTFL	割り込み flag
		VRAM	仮想 VRAM
		DBUFF	DISPLAY BUFFER (キャラクタ表示用)
		MODE	Touch Switch On Panel を文章中 バニアルスイッチ と称する。
		←	Touch Switch Off Panel
		CR	(multi function key)
		→	固定 KEY
		MENU	

図4-3 表示位置とデータの関係

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	BFFE4H	DB 21	BFFF2H
02	E5	22	F3
03	E6	23	F4
04	E7	24	F5
05	E8	25	F6
06	E9	26	F7
07	EA	27	F8
11	EB	31	F9
I2	EC	32	FA
I3	ED	33	FB
I4	EE	34	FC
I5	EF	35	FD
I6	FO	36	FE
I7	FI	37	FF

図4-6 キャラクタコード一覧表

	上位4ビット→								下位4ビット→							
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	#	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	#	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
3	#	3	4	5	6	7	8	9	A	B	C	D	E	F		
4	#	4	5	6	7	8	9	A	B	C	D	E	F			
5	#	5	6	7	8	9	A	B	C	D	E	F				
6	#	6	7	8	9	A	B	C	D	E	F					
7	#	7	8	9	A	B	C	D	E	F						
8	C	O	H	X	T	X	X	Y	Z	A	B	C	D	E	F	
9	J	R	I	V	T	I	N	G	H	K	L	M	N	O	P	
A	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
B	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
C	^	^	^	^	^	^	^	^	^	^	^	^	^	^	^	
D	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
E	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
F	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	

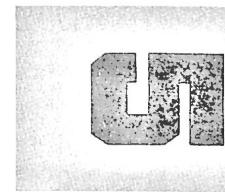
未定義

未定義

## NUMBER

RC-20

# ROM内サブルーチン の使い方



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	I	R	Z	F
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。	• STBY命令後、約1ms以内にH ALT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。	1
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。 1 H 1Hz割り込み 2 H 10Hz割り込み 3 H 1 Hzおよび10Hz割り込み 4 H 1/60Hz割り込み 0 H 上記割り込み解除	3~7ビットのデータは無効とする。	1
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。	• 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。	1
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。	1	
SARE	40B6	レジスタをスタックへ戻させる。 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。	1	
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(ホール命令は不可)。本サブルーチンでは、'MODE'キーを押すと、計時機能もどる。	エラーコード1~2 (はシステムエラー。 I:(①RAMソフトが入っていない。(②RG-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、	

図5-2 KEYコード表					
01	02	03	04	05	06
07	08	09	0A	0B	0C
0D	0E	0F	10	11	12
14	15	16	17	18	

図5-3 DEKEYによるKEYコード表の変換

ラベル	アドレス	機能	コメント	レジスタ 保護
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。	上記のKEYのデコード値を、左図の値に変換する。(HEXコード)	3

図5-2 キー関係サブルーチン					
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、このループ中では、10Hz割り込み同期して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、本ループの前に処理しておくこと。	3	
GETKEY	4090	PUTKEYと以降の次の点が異なる。 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へジャンプする。(それ以外はPUTKEYと同じ)			

## 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	レジスタ保護
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。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
KDISP1	413B	1～3ページに次のKEY表示を行う。	1～3ページの消去は不要。	
		A B C D E F G H I J K L M N O U P T		
KDISP2	4152	1～3ページに次のKEY表示を行う。 0 P Q R S T U V W X Y Z A E I □ 1 A	1～3ページの消去は不要。	7
		0 1 2 3 4 5 6 7 8 9 ; * / - . [ A ] P		
KDISP3	4169	1～3ページに次のKEY表示を行う。	1～3ページの消去は不要。	7
		RSIN		
KALDIS	4112	1～3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。	1～3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
KHYP1	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
KHYP2	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
KHYP3	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
		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からデータが取出されている時、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、音時モードに移る。	
			(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波特 4H : 1200波特 2H : 600ポート 1H : 300ポート	
			• スタートアドレスが8H以上になると、パラメータで指定したバイト数に統計して、最大22バイトの余分な情報(コミ)を送出する。	
			• H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。	
			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800H～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(音時モードになる)。	6

表5-4 音階データ

ラベル	アドレス	機 能	コ メ ン ト	レ ジ ス タ 音 階 保 持
RSIN	4310	(パラメータ) ・ HLレジスタ：格納スタート・アド (註)		6
		・ Aレジスタ：ボーレート   8H : 2400ボー   4H : 1200ボー   2H : 600ボー <sup>一</sup>   1H : 300ボー <sup>一</sup>		
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。	3	
		・ Aレジスタ {01H:エラーなし   FFH:エラー <sup>一</sup>		
RSVER 1	42E5	転送規約(2.1)にもとづいて、RS-232Cのインバ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 <sup>一</sup>	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状態になつてている。 (3)音長コード n=00H 1.6秒 (4)音階データを表5-4に示す。 (パラメータ) ・ 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、その後に22ハイフ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を超える場合は、スタッフエリアの22ハイフが補助エリアとなるためUSH\_CALL命令のネストイングは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。

B80H

BF40H BF5FH

	受信データ エリア	補助 エリア	ス タ ッ ク エ リ ア
ユーザーエリア			

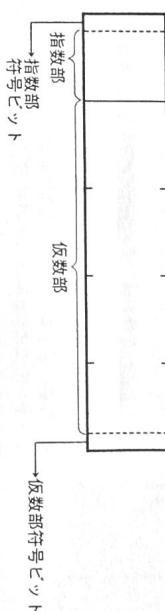
## 5.4 キャラクタ表示関係サブルーチン

表5-5

ラベル	アドレス	機 能	コ メ ン ト	レ ジ ス タ 音 階 保 持
HYOGI	4106	ディスプレイ・パック(BFFE4H~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、ディスプレイパック内に表示すべき ASCII コードを入れておく。	ディスプレイ・パック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ページに対しで行う。	//	7
HYO 23	410C	HYOGIと同様な機能を2~3ページに対しで行う。	//	7
HYO 0	421A	HYOGIと同様な機能を0ページに対して行う。	//	7
HYP I	4221	HYOGIと同様な機能を1ページに対しで行う。	//	7

## 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 :  $\pm 0.99999999 \times 10^{\pm 100}$

図5-5 演算サブルーチンのワークエリア

ラベル	アドレス	機能	能	コマンド	レジスタ 保護
HYP2	4228	HYOGIと同様な機能を2ページに対し て行う。	//	//	7
HYP3	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ヶタの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	5	
ACIBC	4081	Aレジスタに与えられた2ヶタの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。			
BUFC1	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。	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 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	7		

FPACC(フローティング・アキュムレータ)

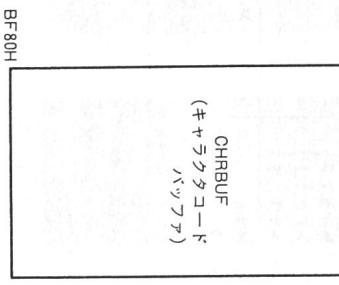
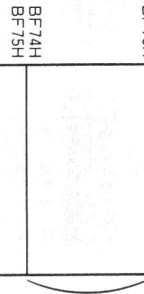


表5-6 演算サブルーチン

ラベル	アドレス	機能	コメント	レジスタ保護
CONVT 1	5A31	キャラクタコードハッファ(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。	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ヶタにまでセッタする。また、CHRBUF内のデータの終りには、00Hを符号、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。 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。)	レジスタ保護
CONVT 2	5BC7	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、キャラクタコードハッファ(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。キャラクタコードハッファ未満の時は、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。変換時のエラーは、Aレジスタに返される。	数値の指数部が10以上、または、-3未満の時は、指標表示に変換する。	
ADDNUM	589F	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。 Aレジスター : 正常終了 1 : エラー(数値に異常あり)	演算結果が0.9999999999999999×10 <sup>100</sup> 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。エラーの有無は、Aレジスタに返される。	同上	
ラベル	アドレス	機能	コメント	レジスタ保護
SUBNUM	58A8	・ Aレジスター 0 : 正常終了 1 : エラー(オーバーフロー)	同上	
MULTPL	572B	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レジスターに返される。 Aレジスター 0 : 正常終了 1 : エラー(オーバーフロー)	同上 また、0.1×10 <sup>-99</sup> よりも小さい時は、0とする。	
DIVIDE	57C6	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レジスターに返される。 Aレジスター 0 : 正常終了 1 : 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、エラーの有無は、Aレジスターに返される。 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反数部の符号を反転する。Aレジスターには、0が入る。		
INVERT	5F66	FPACCに格納されたデータの逆数を求める。結果をFPACCに格納する。エラーの有無は、Aレジスターに返される。 Aレジスター 0 : 正常終了 1 : 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	5F70	0以上1未満の乱数(ヶタ数は小数以下9ヶタで0~0.999999999)を1個だけ発生し、FPACCに格納する。Aレジスターには、0が入る。		

## RC-20

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≤BFFE 3Hとなること)、
- 3) CHARA(404H)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。

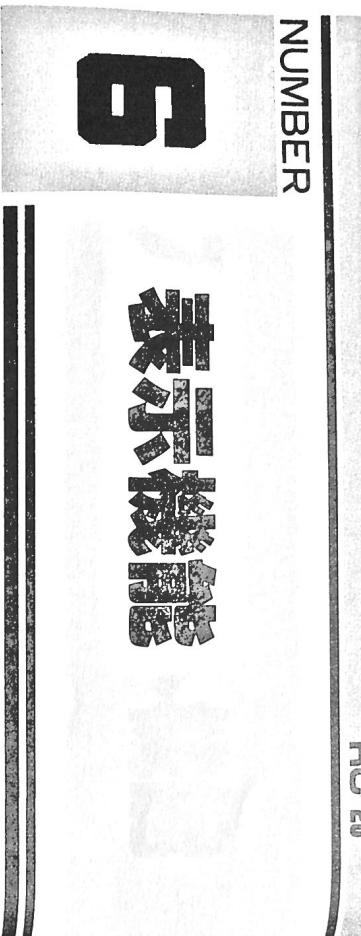


表6-1 グラフィックパラメータ

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

### キャラクタ表示

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レジスターとEレジスターにセットします。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。

# 7 入出力ポート機能

## 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か1/100秒カウンタの値を、D4～D7か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, (WA<sub>n</sub>) [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 時計データ入力

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

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に入れます。

## 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。

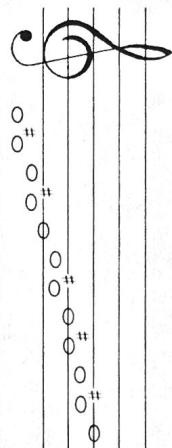
IN A, (60H) .....INKEY

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。

7

	C1	C0	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		

図7-2 音階コードと音階対応表



C = B B = E F F = G G = A A = E

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-  
 LD B,n2 ← 音長  
 LD C,50H ← MUSIC  
 OUT (C),A

Aレジスターには、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。

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

Z80エーアンドEND

この命令により、Z80モードからタイム・スタンダー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。

音階	平均半音	分周近似	音階	音階	音階	音階	音階
(H2)	(H2)	(H2)	(H2)	(H2)	(H2)	(H2)	(H2)
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	

# 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	1HZ	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。
	1HZ & 10HZ	計時機能の1HZと10HZに同期して、割込みをかける。
待機割込み	1/60HZ割込み	時刻表示モードにおいて、「一分ごとに割込みをかける」にシヤンブし、プログラムを実行する。
特殊割込み	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。

- 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。
- ユーザーの割込み処理は、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のです。

図8-1 INTFLの内容

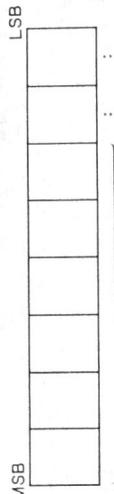
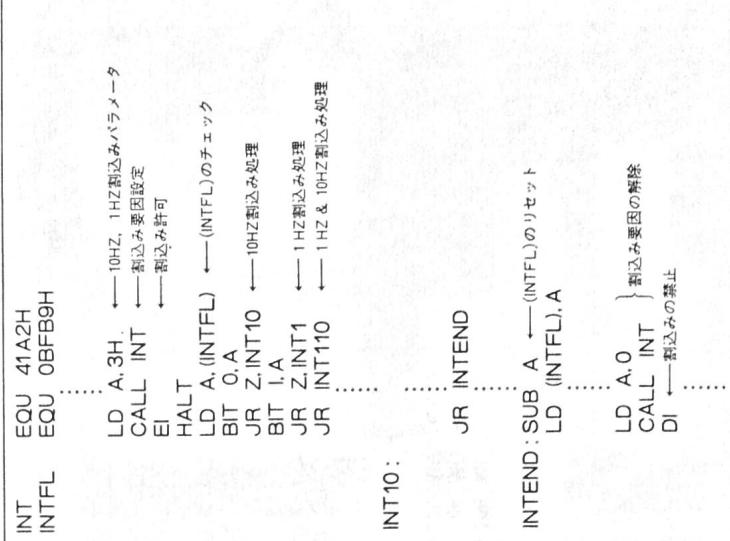


図8-2 割込み処理の例



## 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なります。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命令を行なう必要はありません。

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がセットされている場合、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。

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 l,A
LD A,3H
CALL INT
EI
HALT
B900H : DW INT1
B902H : DW INT10
B904H : DW INT110
INT1 :
INT10 :
INT110 :

```

図8-3 ユーザ割込み処理の例

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。その時の割込みハラメータは07Hです。

図8-4は、使用例を示したもののです。

### Z80 STAND-BY & KEY RESTART割込み

この割込みは、Z80をSLEEP状態にして、KEYが押されるたび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のです。

図8-4  
1/60HZ割込み  
の使用例

```

INT EQU 41A2H
INTBF EQU 0BFB7H
INT A,4H ← 1/60HZ割込み設定
CALL INT
LD A,00H
LD (INTBF),A
LD A,OBAB
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 &amp; 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

```

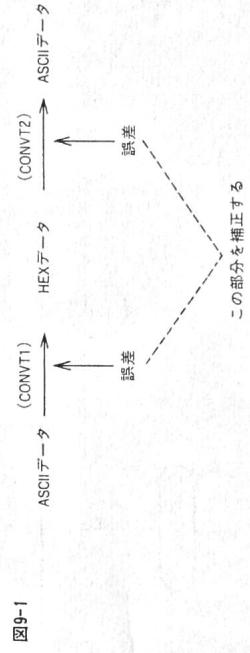
注) 割込み請求中(割込み要因をセットし、割込み許可中)には、RS232CおよびMUSIC関係のI/Oポート(C0H, 50H)を使用する命令の直後において、I/O命令を行わないようにしてください。

# NUMBER 漸動小数点演算 における注意

## 9

### 9.1 データ補正サブルーチン(CORRECT)

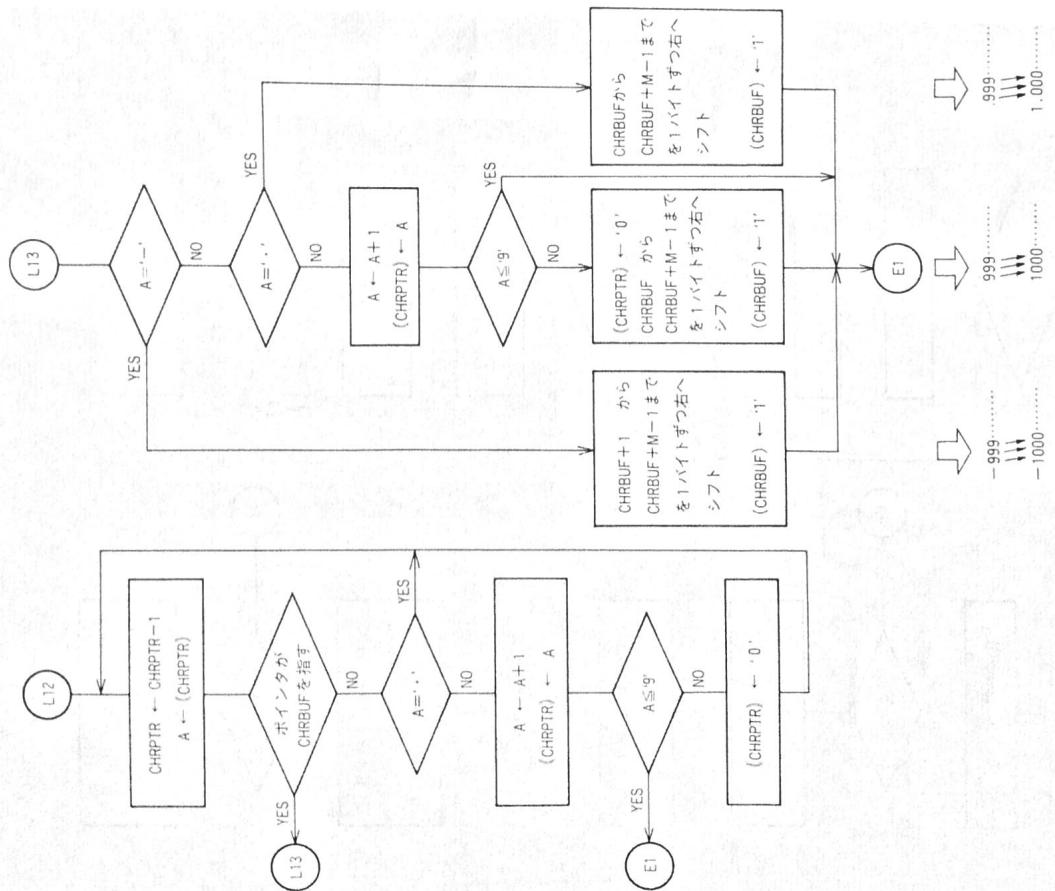
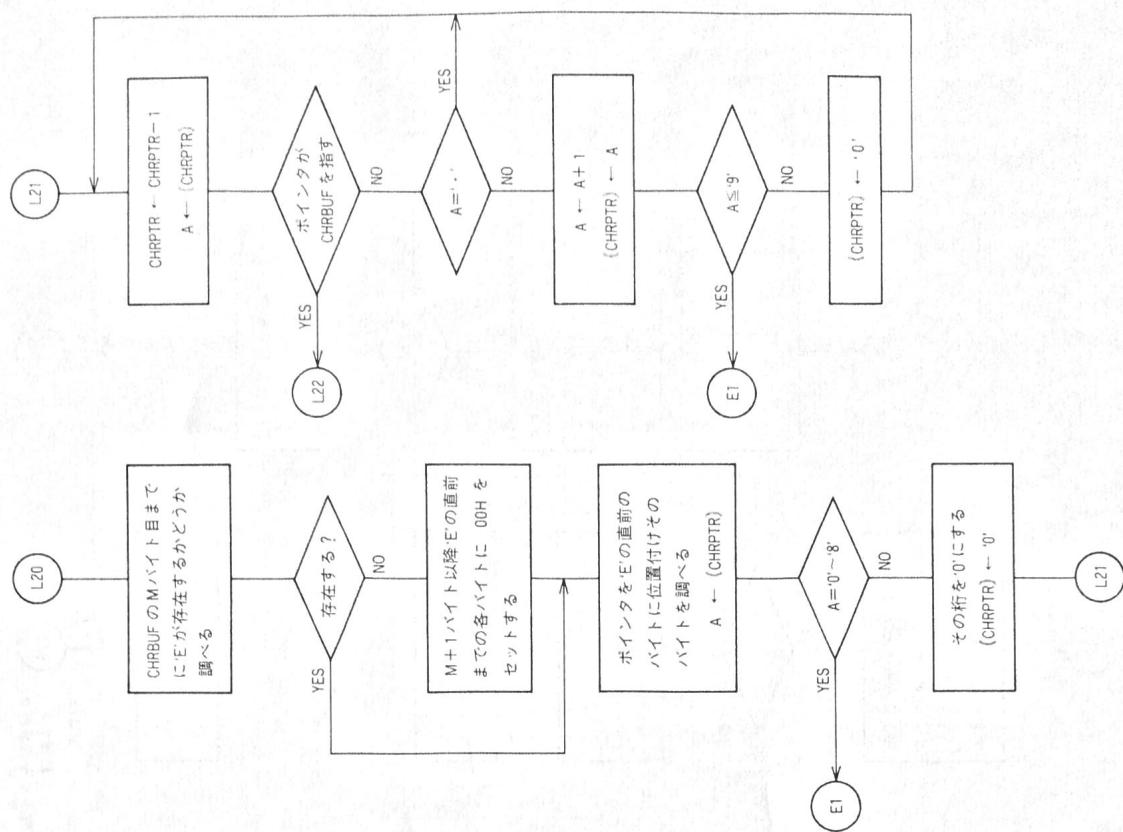
演算サブルーチンで演算を行う場合、一般にキャラクタバッファ内の数値の文字列をCONVT 1(数値変換)によって、浮動小数点数(HEX型式)に変換した後、演算を行い、さらにその結果をCONVT 2(キャラクタ変換)によって、再びキャラクタバッファ内の数値の文字列に変換します。この際、図9-1に示すように、2カ所で誤差が生じます。

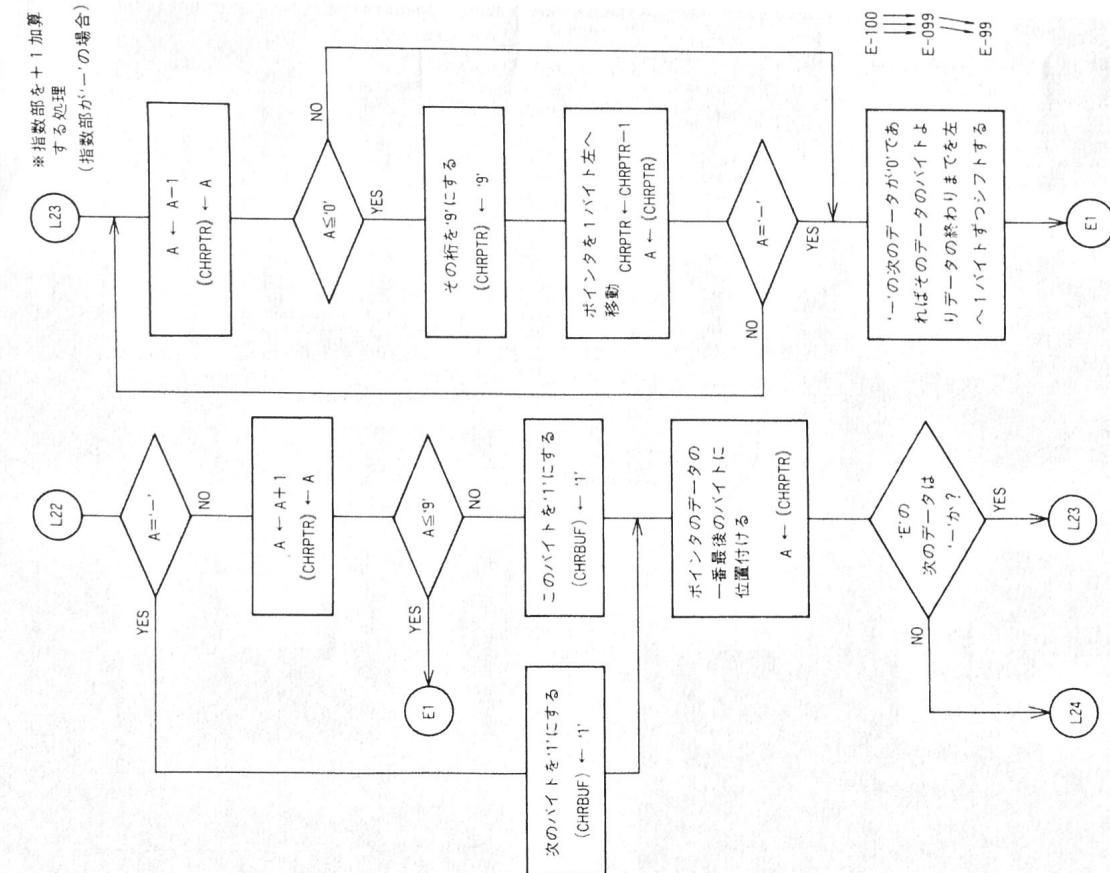
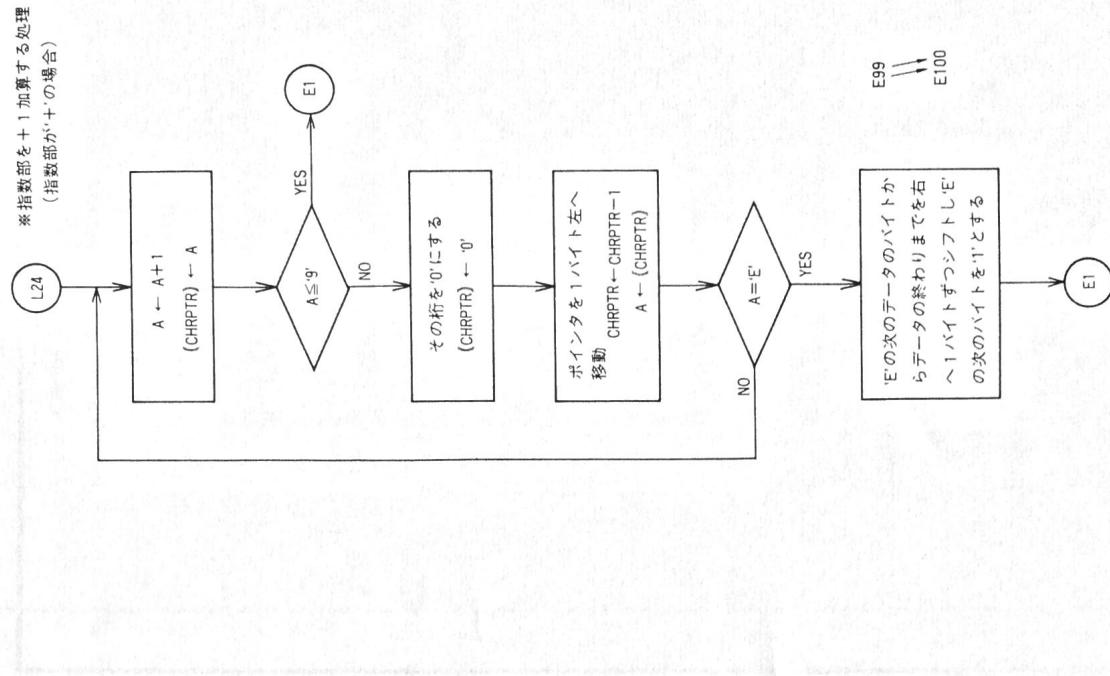


その結果、これらの誤差を補正する必要が生じます。たとえば3.99999は4に、9.99999E-99は1E-98に補正する必要があります。このデータ補正を行うサブルーチン(CORRECT)は、キャラクタコードバッファに格納されているデータを、先頭から9桁目でまるめ込み、結果をキャラクタコードバッファに格納します。このCORRECTサブルーチンは、RC-20のROMには含まれていないので、ユーザが各自で作成して使用してください。

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







## リスト28 CORRECTサブルーチンのソースリスト

```

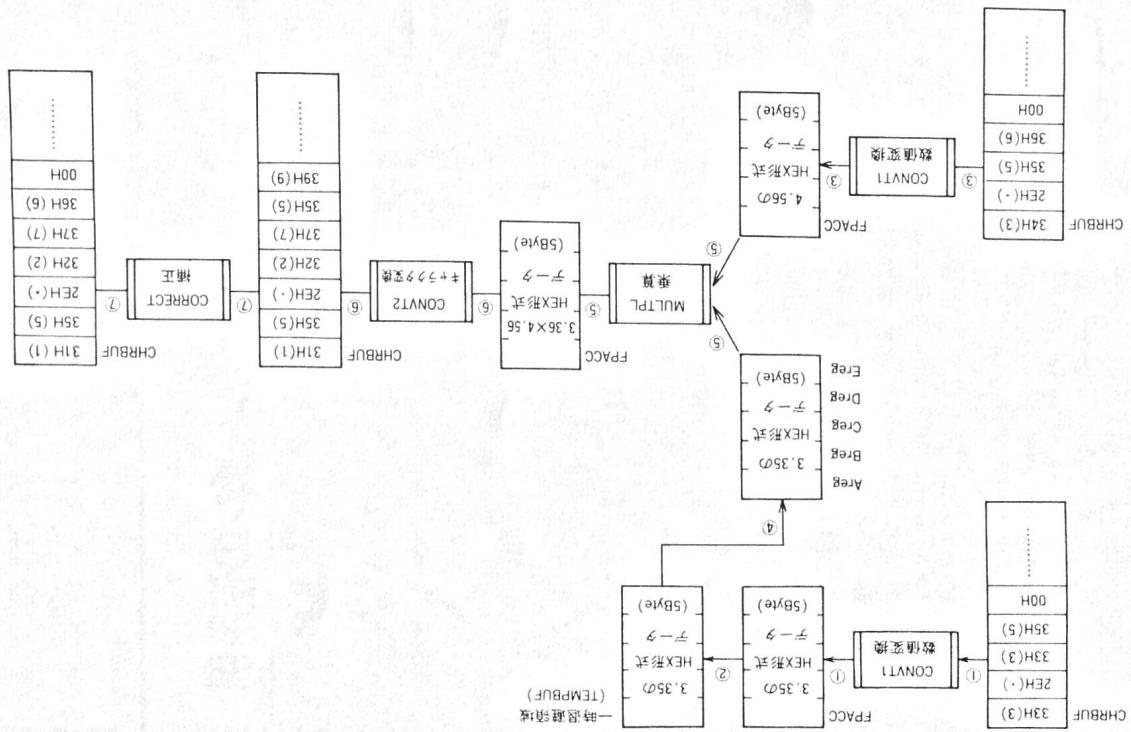
0034' 36 00 LD (HL),0
      C3 01C' JP ENDCRT
;
0035' 21 BF7C LD HL,CHARBUF+H-1
      JP
;
0039' 0039' 7E LD A,(HL)
      JP
003C' 003C' FE 2E LD A,(HL)
      JP
003D' 0046' 2B 04 LD Z,NEXT000
      JP
003F' CA 0034' LD I,CORCT2
      JP
0042' E5 PUSH HL
      JP
;
0043' 0043' 23 INC HL
      JP
0044' 7E LD A,(HL)
      JP
0045' B7 LD A,(HL)
      JP
0046' 2B 04 LD Z,NEXT000
      JP
0048' 36 00 LD (HL),0
      JP
004A' 1B F7 LD NEXT001
      JP
004C' E1 PUP HL
      JP
004D' 7E LD A,(HL)
      JP
004E' FE 39 LD CP,9
      JP
004F' C2 01AC' LD NZ,ENDRCT
      JP
0050' 0053' 36 30 LD (HL),0
      JP
;
0055' 2B DEC HL
      JP
0056' E5 PUSH HL
      JP
0057' 11 BF75 LD DE,CHARBUF
      XOR A
0058' AF LD HL,DE
      SBC HL,DE
0059' ED 52 POP HL
      JR Z,NEXT01
005D' E1 LD A,(HL)
      JP
005E' 28 10 LD A,(HL)
      JP
0060' 7E LD A,(HL)
      JP
0061' FE 2E LD CP,9
      JP
0063' 2B F0 LD JR,Z,LOOPNE1
      INC A
0065' 3C LD (HL),A
      OUTNUM
0066' 77 LD JP NZ,ENDRCT
      LD (HL),0
      JR LOOPNE1
0067' FE 3A LD CP,9
      JP
0068' C2 01AC' LD NZ,ENDRCT
      JP
0069' 36 30 LD (HL),0
      JR
006E' 1B E5 ;NEXT01:
      JP
;
0070' 0070' 7E LD A,(HL)
      JP
0071' FE 2D LD CP,9
      JP
0073' 0073' 2B 1E LD JR,Z,MINSHT
      INC A
0075' FE 2E LD CP,9
      JP
0077' 2B 09 LD JR,Z,PROSHT
      INC A
0079' 0079' 3C LD A,(HL),A
      OUTNUM
007A' 77 LD JR,NZ,ENDRCT
      LD (HL),0
      JR
007B' FE 3A LD CP,9
      JP
007D' C2 01AC' LD NZ,ENDRCT
      JP
0080' 36 30 LD (HL),0
      JR
;
0082' 0082' 01 0008 PRSHT:
      LD MNSH1:
      LD BC,M
;
```



9.2

LOGCS 0106	LOGDCX 0106	LOGDCX 012A	LOGDCX 012B	LOGDCX 012C	LOGDCX 012D	LOGDCX 012E	LOGDCX 012F
PCDCX 0089	PCDCX 0089	MNSH1 0093	MNSH1 0094	MNSH1 0095	MNSH1 0096	MNSH1 0097	MNSH1 0098
MEIX101 0020	MEIX101 0020	NETICX 0070	NETICX 0071	NETICX 0072	NETICX 0073	NETICX 0074	NETICX 0075
MEIX101 0020	MEIX101 0020	NETICX 0070	NETICX 0071	NETICX 0072	NETICX 0073	NETICX 0074	NETICX 0075
MINWSN 0010	MINWSN 0010	MINWSN 0011	MINWSN 0012	MINWSN 0013	MINWSN 0014	MINWSN 0015	MINWSN 0016
NITICX 0049	NITICX 0049	NITICX 0050	NITICX 0051	NITICX 0052	NITICX 0053	NITICX 0054	NITICX 0055
MINWSN 0062	MINWSN 0062	MINWSN 0063	MINWSN 0064	MINWSN 0065	MINWSN 0066	MINWSN 0067	MINWSN 0068
MINWSN 0067	MINWSN 0067	MINWSN 0068	MINWSN 0069	MINWSN 0070	MINWSN 0071	MINWSN 0072	MINWSN 0073
MINWSN 0074	MINWSN 0074	MINWSN 0075	MINWSN 0076	MINWSN 0077	MINWSN 0078	MINWSN 0079	MINWSN 0080
MINWSN 0082	MINWSN 0082	MINWSN 0083	MINWSN 0084	MINWSN 0085	MINWSN 0086	MINWSN 0087	MINWSN 0088

リスト29 「 $3.35 \times 4.56$ 」の演算を行うプログラム (FPTEST)



```

0124 CD 5A31          CALL    C0NVT1
                                ; (TEMP BUFFR) -> A-FREG

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    MULPL
                                ; HEX CODE -> ASCII CODE
0136 CD 5FC7          CALL    C0NVT2
                                ; DATA CORRECTION
0139 CD 000000          CALL    CORRECT
                                ; RET
013C C9                RET
                                ; N3.35: DB '3.35',0
013D 33 2E 33 35        DB      '3.35'
0141 00                DB      ,0
                                ; N4.56: DB '4.56',0
0142 34 2E 35 36        DB      '4.56'
0146 00                DB      ,0
0147 TEMPBUF: DS 5      ; END

```

図9-4 FPTESTプログラムの実行の中途および最終結果

# スケジューラ、メモ、ワールドタイムのデータ構造

NUMBER

9

BF70: [01 F4 6A 7E E6 01] F4 6A 7E 4D D2 F2 00 35 99  
 BF80: 99 98 00 00 00 00 00 00 00 00 00 00 00 00 00 75 BF

BF8F: 01 F4 6A 7E E6 01 F4 6A 7E 4D D2 F2 00 35 99  
 BF90: 99 98 00 00 00 00 00 00 00 00 00 00 00 00 00 75 BF

BF9F: 01 F4 6A 7E E6 01 F4 6A 7E 4D D2 F2 00 35 99  
 BFA0: 99 98 00 00 00 00 00 00 00 00 00 00 00 00 00 75 BF

BFBF: 01 F4 6A 7E E6 01 F4 6A 7E 4D D2 F2 00 35 99  
 BFC0: 99 98 00 00 00 00 00 00 00 00 00 00 00 00 00 75 BF

BFDF: 01 F4 6A 7E E6 01 F4 6A 7E 4D D2 F2 00 35 99  
 BFE0: 99 98 00 00 00 00 00 00 00 00 00 00 00 00 00 75 BF

BFEF: 01 F4 6A 7E E6 01 F4 6A 7E 4D D2 F2 00 35 99  
 BFF0: 99 98 00 00 00 00 00 00 00 00 00 00 00 00 00 75 BF

BFFF: 01 F4 6A 7E E6 01 F4 6A 7E 4D D2 F2 00 35 99  
 BFFC: 99 98 00 00 00 00 00 00 00 00 00 00 00 00 00 75 BF

#0136  
#DBF70, BFBF  
BF70: [01 F4]  
BF80: 99 98  
#6.139

## 10.1 ベッダ・データの構造

表10-1 スケジューラ、メモ、ワールドタイマーのファイル・ヘッダの構造

位置	名 称	内 容	密
0	ファイル種別	00H スケジュール・データ 01H メモ・データ 03H フールド・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	ファイルのレコード長のバイト 数を示す。	
6, 7	レコード数	ファイル種別 00H ファイル中のレコード長 01H のバイト数(最大:FFH) 03H 17バイト固定  ファイル中のレコード数を示す ファイル種別 00H ファイル中のレコード数 01H (最大:FFFFH) 03H ファイル中のレコード数 (最大:FFFFH)	内 容 0000-7FFFまでの 範囲に入るレコード数とすること。
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
-------	-------	-------	-------	-------------	-------------

- 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	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	9バイト.....分データ
-------	-----------------	---------------	---------------	---------------	-----------------	---------------	---------------

スケジュール・アラーム 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

スケジュール・アラーム 00H : AM

08H : PM

01~12 (BCD)

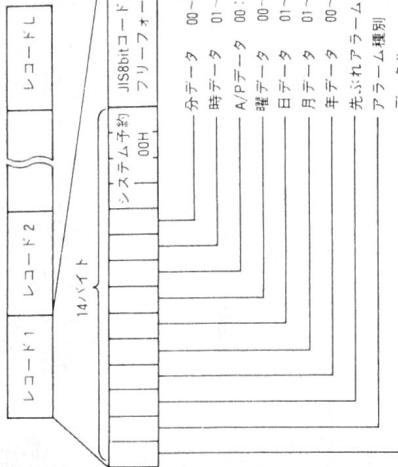
00~59 (BCD)

00, 00, 00H (将来の拡張に従う)

JIS 8bitコードに準ずる(コメント文)

- 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年以内となります。

図10-1  
スケジュール・データの構造



(※1)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

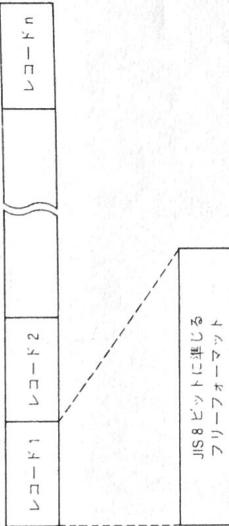
表10-2

位置	名 称	内 容	容	RC-20での対応
0	データFlag	00H 01H	データ無効 データ有効	
1	アラーム種別	01H 02H 03H	日アラーム 曜アラーム ヶ月アラーム	アラームはOFFされている 毎日セトした時刻にアラームをならす 毎曜セトした時刻にアラームをならす セトした年・月・日・時・分にアラームをならす
2	先ぶれアラーム	00	:先ぶれアラーム時刻	0 0 (BCD)
3	年データ	キデータ 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 01 02 03 04 05 06	曜データ SUN MON TUE WED THU FRI 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	システム予約	00, 00, 00H	(将来の拡張に従う)	
11	コメント文	JIS 8bitコードに準ずる(コメント文)		
12		フリーフォーマット	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10-21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

ワードタイム・データの構造を図10-3に示します。

20日（アランタ）を書き込んでおきます

卷之三

サマー・タイムを設定した時は、サマータイムの時差データを記入します。

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。

十一、日本語をベースとする言語

小一公時刻は、全半時日と、八二七六；

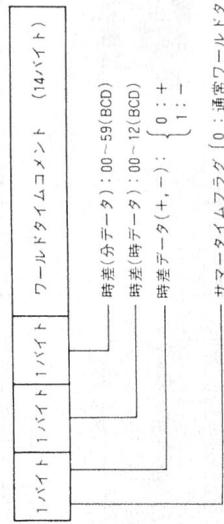
図10-3 ワールドタイム・データの構造

ANSWER

ワールドタイムメント  
(ワールドタイムメント)

卷之三

図10-3 フニルドタイン： $\pi-\text{タ}\text{イ}\text{ン}$ の構造



134

## 10.5 輸送上での注意

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・データの場合には、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、空きエリヤのレコードは、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、および先述アラーム時刻(00Hデータを設定)は、必ず設定してください。

图 10-4

135

# NUMBER 11 ダンプリストのリスト用プログラム

NUMBER  
11

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ッダとプログラム本体の全部を含んでいますか、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、&H100₈リ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ハイトで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となりますが、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";Fs
15 INPUT "rc-20 ready ok";X$  

20 OPEN "R";#1,F$  

25 OPEN "0";#2, COMG: (C8N3F)  

30 FIELD #1,128 AS A$  

40 N=1  

50 GET #1,N  

60 PRINT #2,A$;
```

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

## 11.2 PC-8801/8801mkII, 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&H100₈リモリに書き込み、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&gt;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-8801/8801mkII, PC-8801/8801mkIIのCP/M用転送プログラム

```
; TRANS
; TRANS.COM FILE TO RC-20,
; FOR FC-8801/PC-8801 MK2
; & PC-8801/PC-8801 MK2

0100      ; ORG 100H
0005 =    ; B005 EQU 5
0000 =    ; B000 EQU 0
005C =    ; FCB EQU 005CH
0080 =    ; BUFR EQU 0080H
00CE =    ; MODE EQU 0CEH
0020 =    ; PORT0 EQU 20H
0021 =    ; PORT1 EQU PORT0+1

; LXI SHLD BPOINT
0100 210802 ; LXI H, AREA
0103 220402 ; SHLD BPOINT
0106 3AB600 ; LDA BUFR
0109 119C01 ; LXI D, MSERR
010C B7 ; ORA A
010D CAA601 ; JZ EXIT
; OPEN FILE
0110 0EOF ; MN1 C,15
```

```

; PRMSG:          ; $115000
0112 115000 LXI D,FCB      ; CALL B005
0115 C05000 CALL CPI      ; OFFH
0118 FEFF LXI D,MSOPNER   ; EXIT
011A 11B301 JZ
011D CA6601 ;               ; $115000

; LOOP1:          ; $115000
0120 CDB201 CALL READF    ; EOF
0122 C24401 JNZ EOF      ; BPOINT
0126 2A0402 LHD BPOINT
0128 EB XCHG
012A 21B000 H,BUFR
012D 0EB0 MV1 LXI C,128
012E 0EB0 MV1 ;               ; $115000

; LOOP1:          ; $115000
012F 7E MOV A,M      ; STAX D
0130 12 D
0131 23 INX H      ; INX D
0132 13 INX D      ; DCR C
0133 0D DCR C      ; LDDP1
0134 C22F01 JNZ LXI B,128
0137 01B000 LHD B
013A 2A0402 BPOINT
013D 09 DAD B      ; BPOINT
013E 220402 SHLD JMP LOOP1
0141 C32001 ;               ; $115000

; EOF:            ; $115000
0144 0E10 MV1 C,16      ; FILE CLOSE
0146 115C00 LXI D,FCB
0149 C05000 CALL B005
014C 2A0B02 LHLD AREA+3
014F 220602 SHLD COUNT
0152 11F101 LXI D,MS000
0155 C06C01 CALL PRMSG
0158 0E01 MV1 C,1
015A C05000 CALL B005
015D C0BC01 CALL SINIT
0160 C0D6C01 CALL SIO
0163 117101 LXI D,DOMMSG
0166 C06601 CALL PRMSG
0169 C30000 JMP EXIT
;               ; $115000

; MSOPNER:        ; $115000
016C 0E09 MV1 DB      ; INPUT FILE OPEN ERROR!, '$'
016E C30500 JMP B005
;               ; $115000

; DOMMSG:         ; $115000
0171 0D0A0204AA DB 13,10, '$JOB COMPLETE!', '$'
;               ; $115000

; MSOPNER:        ; $115000
0183 202A4A4E50 DB , '$INPUT FILE OPEN ERROR!', '$'
;               ; $115000

; MSERR:          ; $115000
019C 202A4E4F20 DB , '$NO INPUT FILE NAME!', '$'
;               ; $115000

; READF:          ; $115000
01B2 0E14 MV1 C,20
01B4 115C00 LXI D,FCB
01B7 C0D500 CALL B005
01B8 B7 ORA A
01BB C9 RET
;               ; $115000

; SINIT:          ; $115000
01BC SE00 MV1 A,00H
01BE D521 OUT PORT1
01C0 D521 OUT PORT1
01C2 D521 OUT PORT1
01C4 SE40 MV1 A,40H
01C6 D521 OUT PORT1
01C8 SECE MV1 A,MODE
01CA D521 OUT PORT1
01CC SE15 MV1 A,15H
01CE D521 OUT PORT1
01D0 C9 RET
;               ; $115000

; OUTPUT:         ; $115000
01D1 F5 PUSH PSW
;               ; $115000

; WAIT:           ; $115000
01D2 DB21 IN PORT1
01D4 1F RAR
01D5 D2D201 JNC WAIT
01D8 F1 POP PSW
01D9 D220 OUT PORT0
01DB C9 RET
;               ; $115000

```

```

S10:      LHLD COUNT
          PUSH H
          POP B
          LXI H, AREA
          MOV A, M
          CALL OUTPUT
          INX H
          DCA B
          MOV A, B
          ORA C
          JNZ S102
          RET
;
01F0 C9      ;           RET
;
MS000:      DB    'IIC-20 READY OK?', '$'
;
01F1 202452432D ;           BPOINT:
;
0204 0000:   COUNT: DW 0
0206 0000:   AREA EQU $
;
0208 =       ;           END
;
0208

```

```

50 READ DA$          ;           RET
60 POKE AD,VAL("ah+DA$")
70 NEAT
80 INPUT "IIC-20 ready ok";$
90 A=USR(1)
100 PRINT "1.job [com]etet"
110 END
120 DATA CD,07,90,CD,26,90,09
130 DATA JE,09,03
140 DA@A21,D3,21,3E,49,03,21,3E,CE,03,21,3E,15,03
150 DATA@21,C9,F,0B,21,1F,30,FB,F1,D3,20,C9,ED,4B,03,BB
160 DATA@21,00,89,7E,CD,1C,90,23,08,78,B1,20,F6,C9
;
```

リスト 67 マシン語部分のソースリスト

```

;           RET
;
01F0 C9      ;           RET
;
MS000:      DB    'IIC-20 READY OK?', '$'
;
01F1 202452432D ;           BPOINT:
;
0204 0000:   COUNT: DW 0
0206 0000:   AREA EQU $
;
0208 =       ;           END
;
0208

```

```

;           RET
;
01F0 C9      ;           RET
;
MS000:      DB    'IIC-20 READY OK?', '$'
;
01F1 202452432D ;           BPOINT:
;
0204 0000:   COUNT: DW 0
0206 0000:   AREA EQU $
;
0208 =       ;           END
;
0208

```

## 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HB800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ソースリストを、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 Ittrans11
20 CLEAR 100, tr8FFF
30 DEF USE=H9000 TO trH037

```

```

;           RET
;
01F0 C9      ;           RET
;
MS000:      DB    'IIC-20 READY OK?', '$'
;
01F1 202452432D ;           BPOINT:
;
0204 0000:   COUNT: DW 0
0206 0000:   AREA EQU $
;
0208 =       ;           END
;
0208

```

```

;           RET
;
01F0 C9      ;           RET
;
MS000:      DB    'IIC-20 READY OK?', '$'
;
01F1 202452432D ;           BPOINT:
;
0204 0000:   COUNT: DW 0
0206 0000:   AREA EQU $
;
0208 =       ;           END
;
0208

```

```

9022 F1      POP AF
9023 0320    OUT (PORT0),A
9025 C9      RET

9026 ED4B03B8 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 H1,S102
9037 C9      RET
;          ENO
9038

```

## 11.5 HC-20の場合

モニタのSコマンドで、ダンプリストを&H000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コマンドを使つか、BASICのSAVEMコマンドを使って、それをテーブ上に格納します。

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、テーブ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 MENSET &H1400
30 WIND
40 INPUT "FILE NAME";F$
50 LOADM "CAS0:+F$"
60 OPEN "0",#1,"COM0:(58
H2F)"
70 WIDTH "COM0:255
80 INPUT "*RC-20 READY 0
K";:$
90 ZL=PEEK(&H0003)
100 ZH=PEEK(&H0004)
110 I=ZH*256+ZL
120 FOR K=0 TO I
130 A=PEEK(&H0C08+K)
140 PRINT #1,CHR$(A);
150 NEXT K
160 PRINT "*JOB END*"
170 CLOSE #1
180 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 *transnt
110 CLEAR ,&H000
120 INPUT "file name":F$
130 BLOAD F$
400 LL=PEEK (&HCB03)
410 IH=PEEK (&HCB04)
420 I=IH*256+LL
450 OPEN "COM1:N88N" FOR OUTPUT AS #1
460 INPUT "rc-20 ready ok":$1
470 FOR K=0 TO 1
480 A=PEEK (&HCB00+K)
490 PRINT #1,CHR$(A);
500 NEXT K
510 PRINT "#1 job complete"
520 CLOSE #1
530 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

# 実用マシン言語 プログラム集

NUMBER

12

```

1040 01C7      EXIT:DS    0          MESS
1050 01C7      CODE$01    CALL       JP
1060 01CA      C30000    BOOT       MSOPNER:DS 0
1070 01CD      0          13,10, 'open error',13,10,'$'
1080 01CD      0004020A   DB
1090 01DF      MESS:DS   0          LD
1100 01DF      0E09      C,9
1110 01E1      C00300    CALL       B005
1120 01E4      C9      RET
1130 01E5      BPOINT:DS 0
1140 01E5      0040     DW
1160 01E7      AREA
1160 01E7      END
B005 0005      BOOT     0000 FCB 005C
BUFFR 0080      AREA
MSERR 0144      E11T    0107 NOVEF 015E
MESS 0130      MESS    010F S10 01AA
MSEND 0121      MSOPNE 01CD LOOP0 0176
READF 0140      EOF     0197 BPOINT 01E5
LOOP1 0184      S102 01B1 OUTPUT

```

## 11.7 FM-7の場合

モニタのMコマンド、&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 ,#4FFF
30 INPUT "file name";FS
40 LOADM FS
50 J-PEEK #H5003
60 I=PEEK #H5004
65 I=I+1256
70 OPEN "U",41, "COM0:(ISBN2)"
80 INPUT "rc-20 ready ok";X$
90 FOR I=0 TO 1
100 A=PEEK #H5000+K)
110 PRINT #1,CHR$(A);
120 NEXT K
130 PRINT "job end"
140 CLOSE #1
150 END

```

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。**[タブ]**または**[フタリ]**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]**キーを押します。最後に**[CR]**キーを押して終ります。

金利計算

=MENU =  
・タブ  
・フタリ

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。

2 ガンキン?  
¥20000  
56789C  
01234.

3 イリツ?  
6. 5 %  
56 7 8 9 C  
0 1 2 3 4.

Add +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum  
B900 CD A5 BA CD DB BA CD C1 43 3E 07 11 75 BF CD 90 : 46  
B910 CD 21 42 0E 03 CD B7 BA CD AC 30 E8 CD 30 E8 CD 90 : 46  
B920 B9 38 D0 CD 31 5A C3 6A BB B7 BD DD 3F 00 CD C1 : 18  
B930 43 AF 11 70 B9 CD 90 BA CD 1A 42 21 3F 25 22 EB : FE  
B940 BF CD A5 BA CD DB BA 18 BD CD 3E 07 11 75 : 7A  
B950 BF CD 90 BA 36 25 CD 21 42 0E 06 CD B7 BA CD AC : 2C  
B960 BB 30 E6 CD 31 5A CD 49 BB CD 28 57 CD 65 BB C9 : FF  
B970 D8 D8 E2 3F 00 CD 00 CD 7E CD 53 BB CD 2B 57 B7 C2 : 5A  
B980 2D BB CD 44 BB CD 9F 58 CD 4E BB C3 2B 57 CD 6A : CA  
B990 42 3E 07 11 C9 B9 CD 90 BA CD 21 42 CD 7E BB CD 34 : B>  
B9A0 44 BB CD 9F 58 CD 6A BB 3A AC BD 01 32 AC BD 01 : 2A  
B9B0 28 0E 38 12 CD 53 BB CD 2B 57 EB C3 2B BB 1F : 1F  
B9C0 CD 4E BB C3 2B 57 C3 79 BB B9 B2 BB DD C1 AD B3 : 36  
B9D0 CD C1 43 AF 11 EE B9 CD 98 BA AF CD 88 BB B7 : C5  
B9E0 C2 2D BB 3E 0F 11 75 CD 98 BA CD 06 41 B6 : F1  
B9F0 DD D8 20 20 BA DE B3 B9 B2 20 5C 00 21 75 : FD  
Sum DB DD C9 36 B9 4A 30 18 50 87 74 C6 00 9D 79 52 : 7B

Add +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum  
BA00 BF 01 00 03 2E B7 20 0D 78 2B FE 82 28 45 2B FE : 5E  
BA10 01 28 33 37 C9 FE 32 20 10 23 7E FE 36 2F D8 : C7  
BA20 FE 33 3F D8 06 23 7E FE 36 2F D8 06 23 7E FE 35 : C7  
BA30 29 06 23 7E FE 36 2F D8 06 23 7E FE 35 : C7  
BA40 C6 64 4F 10 FA 23 7E D6 30 28 07 47 79 C6 8A : 4F  
BA50 18 FA 23 7E D6 30 81 32 AC BD B7 C9 CD BB 3E : CF  
BA60 31 32 75 BF CD 31 5A C3 74 BB CD 91 43 FE : D3  
BA70 30 31 22 76 BF CD 31 5A C3 74 BB CD 91 43 FE : D3  
BA80 28 09 FE 34 C9 FE 36 20 F2 E1 C9 4A : 4A  
BA90 FE 28 D0 06 00 4F 21 E4 FE 09 7D B7 F0 1A : C8  
BAA0 77 13 23 18 F5 21 AB BA 13 15 41 35 36 37 38 : 39  
BAB0 43 30 31 32 33 34 2C CD 43 31 32 33 34 2C CD 43 : 43  
BAC0 FE 16 38 3F 15 15 FE 26 DA 07 BB 28 19 FE : 34  
BAD0 D9 FE 36 20 E1 E1 C9 37 C9 CD BC AF 32 : 78  
BAE0 A8 BD 32 A9 BD C9 3A A9 BD 32 A9 BD 3A : EB  
BAF0 A8 BD B7 20 0A 3E 30 32 75 BF 3E 01 32 AB BD 3E : 2E  
Sum IC E1 DF 15 9E 0D B0 C8 6C 4B 42 A6 B8 F4 4E 79 : 26

4 キカシ?  
期間(月数、年数など)を入力。  
1 2  
56 7 8 9 C  
0 1 2 3 4.

Add +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum  
ガソリ  
ゴウケイ  
¥35 600

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

Add +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum  
B880 01 00 00 A8 05 16 00 00 00 02 16 00 A8 05 00 : 89  
B881 00 00 00 00 00 00 00 00 00 00 00 00 00 00 : 00  
B882 45 49 53 41 4E 18 05 00 16 00 00 00 00 00 : 00  
B883 34 B8 1B FB CD C1 43 AF 11 62 B8 CD 1A : A8  
B884 42 CD 5A 42 21 69 B8 CD 15 41 CD 91 43 FE : 87  
B885 09 FE 17 38 20 FE 27 DA 99 FE 32 20 EC E1 : C3  
B886 09 40 3D 4D 45 4E 55 3D 00 20 A5 C0 DD 08 : 26  
B887 05 CC BB D8 20 CD A9 B8 CD 2E B9 CD F3 : B7  
B888 B9 CD D1 B9 CD 7B BA 18 EC CD A9 B8 CD 2E : B9  
B889 03 B8 CD 86 B9 CD AC BB CD D1 B9 CD 7B BA 18 : E9  
B890 F3 B8 CD 86 B9 CD DB BA CD B5 BB 3A AB BD : B7  
B891 00 AF 11 EC BB CD 90 BA CD A5 BA F1 0E 07 : FE  
B892 05 1E 00 CD C1 43 7B 16 00 21 E4 BF 19 36 5C : 30  
B893 11 75 BE CD 90 BA CD 1A 42 0E 09 CD B7 : 50  
B894 BA CD AC BB 36 BD CD 31 5A C3 60 BB DE DD : B7  
B895 DD 3F 00 CD C1 43 AF 11 29 B9 CD 90 BA CD 1A : 42  
Sum AF 1D 83 3E 6E 4A 78 CE D5 61 B0 7B IC 2A 5F 38 : C9

## プログラムリスト

ダブルリスト

Add +0 +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 0F 57 3A A8 BD B9 00 : 47  
BB10 5F B7 20 09 CD BD 7A FE 30 28 06 BD 7B 3C 32 : AB  
BB20 BD 7A 00 21 75 BF 19 77 23 AF 77 C9 CD 6A : 42  
BB30 21 3E BB CD F3 41 CD 91 33 28 C5 BD 18 02 : 52  
BB40 52 4F 52 21 C0 BD 18 02 21 C5 BD 18 02 : 52  
BB50 BD 18 03 21 BB BD 56 28 4E 46 2B 7E C9 : AC  
BB60 11 AD BD 18 3E 11 B2 BD 18 39 11 B7 BD 18 : 34  
BB70 BC BD 18 2F 11 C1 BD 18 2A 21 AD BD 18 20 : 21  
BB80 BD 18 1B 21 B7 BD 18 16 F5 CD BB 17 5F C7 : BF  
BB90 C3 CA BB 21 75 BF 11 F2 BF 01 0C 00 18 BB 11 : 70  
BB99 BF 18 03 21 70 BF 01 05 00 ED BB C9 3E 80 06 : 01  
BB99 0E 50 ED 79 C9 21 EB 06 07 18 08 21 75 BF 36 : 10  
BBC0 30 23 06 00 00 23 10 FB C9 F5 21 75 BF E5 : 3E  
BB99 B7 28 08 FE 45 CA 67 BC 23 18 F4 AF 32 AA BD E1 : 6F  
BBE0 06 08 27 FE 2E 28 21 23 10 FB 7E B7 CA 23 BD FE : 09  
BBF0 2E 28 10 36 30 23 E5 11 81 BF AF ED 52 E1 CA 23 : E1  
Sum AF 1D 83 3E 6E 4A 78 CE D5 61 B0 7B IC 2A 5F 38 : C9

ASCII																	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
BCB0	BD	18	E7	36	00	C3	23	BD	21	7C	BF	7E	FE	2E	39	C3	:68
BCB10	BC	23	E5	28	04	36	00	AA	7E	7F	E1	28	10	7E	:28		
BCB20	BC	33	70	28	05	11	75	BF	AF	ED	52	E1	28	10	7E	:28	
BCB30	FE	28	E0	37	07	FE	3A	23	BD	30	18	E5	7E	:1B			
BCB40	FE	2D	28	1E	FE	2E	28	09	3C	77	FE	3A	23	BD	30	:91	
BCB50	39	01	08	09	21	7C	BF	11	7D	BF	E8	23	36	11	01	:01	
BCB60	BD	79	FE	08	07	00	20	F4	E5	7E	FE	45	28	05	36	:01	
BCB70	BC	86	02	01	00	01	01	01	01	01	01	01	01	01	01	:01	
BCB80	FE	22	AA	BD	21	7E	BF	18	63	22	AA	BD	21	7E	BF	:C9	
BCB90	BD	36	30	2B	EE	15	75	BF	ED	52	E1	28	10	7E	:FB		
BCBA0	BD	36	30	2B	EE	15	75	BF	ED	52	E1	28	10	7E	:FB		
BCBB0	BD	28	0B	3C	77	FE	3A	C2	23	BD	36	30	18	E5	7E	:B2	
BCBC0	BD	31	21	0B	7F	77	FE	3A	C2	23	BD	36	31	18	E5	7E	:B2
BCBD0	45	20	FA	23	54	7D	7E	FE	2D	20	24	E1	3D	20	F2	:A6	
BCBE0	BD	30	08	36	08	23	BD	21	7E	BF	ED	52	E1	28	10	7E	:A6
BCBF0	BD	23	BD	62	BD	23	7E	12	B7	CA	23	BD	13	23	18	F6	:E1
Sum	CE	22	50	E4	3F	DF	09	36	55	81	A6	98	56	9C	1D	E9	:8D
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
BD00	F1	3C	77	FE	3A	C2	23	BD	36	30	2B	7E	FE	45	20	F1	:E1
BD010	21	7F	BF	11	00	01	01	01	01	01	01	01	01	01	01	01	:03
BD020	3E	31	12	3A	BA	2B	46	2A	AA	BD	23	7E	FE	03	38	64	:C6
BD030	BD	28	06	01	00	01	01	01	01	01	01	01	01	01	01	01	:00
BD040	FE	08	08	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BD050	23	36	2E	23	01	05	36	30	23	10	01	01	01	01	01	01	:00
BD060	7F	06	01	02	01	01	01	01	01	01	01	01	01	01	01	01	:00
BD070	7F	05	00	06	01	01	01	01	01	01	01	01	01	01	01	01	:00
BD080	BD	54	05	00	06	01	01	01	01	01	01	01	01	01	01	01	:00
BD090	BD	04	2B	7E	FE	28	08	01	01	01	01	01	01	01	01	01	:00
BD0A0	13	04	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:00
BD0B0	36	00	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:00
BD0C0	BD	32	30	13	04	01	01	01	01	01	01	01	01	01	01	01	:00
BD0D0	3A	36	32	01	01	01	01	01	01	01	01	01	01	01	01	01	:00
BD0E0	23	36	2E	23	01	05	36	30	23	10	01	01	01	01	01	01	:00
BD0F0	28	BD	01	01	01	01	01	01	01	01	01	01	01	01	01	01	:00
BD100	75	BF	06	0C	7E	23	BD	28	04	19	F8	18	24	21	75	19	:00

```

***** LITHINE *****

; Z-80 Standby & Key Restart Set
; Interrupt Code Set
; Z-80 Control End From Stack
; Load All Register Into Stack
; Save All Register Into Stack
; Display Of Error Code

; Load Key (Scan & Key Data Input)
; Put Key
; Get Key
; Key In (Load Of Input Key Data)
; Decode Of Key Data
; Key Display
; Key Display

```









	NOMINS:	POP	HL
	POP	AF	HL
BD011'	3C	INC	A (HL) A
BD012'	77	BD	C. OUTNUK
BD013'	36	CD	NZ ENDRECT
BD005'	30	JP	(HL), 0"
BD008'	2B	LD	DEC
BD00B'	7E	UD	A. (HL)
BD00C'	CP	CP	"E"
BD0E'	20	JR	NZ, 1. LOOPK7
BD10'	21	LD	HL CHRBIF+1
BD13'	11	BD	DE, CHRBIF+1
BD16'	LD	DE	HL
BD17'	7E	CP	Z, NEXTCK?
BD19'	FE	JR	(DE), A
BD1B'	28	LD	DEC
BD1C'	05	DE	HL
BD1D'	1B	JR	LOOPK8
BD1E'	18	FE	
BD20'	3B	31	
BD22'	12	LD	A, "1" (DE), A
;			
ENDRCT:	LD	LD	A. (EE)
;	OR	A	Z, END4
BD23'	3A	BDAA'	JR
BD26'	B7	LD	LD
BD27'	2B	46	INC
BD29'	2A	BDAA'	LD
BD2C'	23	LD	HL (EE)
BD2D'	7E	LD	A. (HL)
BD2E'	FE	2D	CP
BD2F'	2B	05	JR
BD30'	21	01	LD
BD32'	CD	BB93'	EX
BD35'	BD	66	END4
BD36'	1B	37	JR
BD38'	2B	23	INC
BD39'	7E	7E	HL (EE)
BD40'	FE	03	SUB
BD41'	2B	03	30H
BD42'	FF	64	JR
BD43'	0B	08	CP
BD44'	30	60	JR
BD45'	F5	08	NC, ZERO
BD46'	AF	08	PUSH
BD47'	CALL	TRCD	CALL
BD48'	CD	BBBF	CLRCBF
BD49'	BB	75	HL CHRBIF
BD50'	2B	3B	(HL), 0"
BD51'	LD	INC	HL
BD53'	3B	2E	(HL), ..
BD54'	LD	INC	HL
BD55'	05	05	DEC
BD56'	3D	30	BC
BD57'	3B	23	(HL), 0"-
BD58'	10	FB	HL
BD59'	11	BFF2	DNZ
BD5B'	LD	LD	DE, DBUFF2
BD5E'	1A	LD	A. (DE)
BD5F'	B7	OR	A
BD60'	2B	0D	Z, END4
BD62'	FE	2E	CP
BD64'	2B	01	JR
BD66'	77	77	Z, END3
BD67'	13	INC	(HL), A
BD68'	23	INC	DNZ
BD69'	04	HL	DE, DBUFF2
BD6A'	7B	04	A, B
BD6B'	FE	09	CP
BD6C'	0B	EF	9
BD6D'	2B	75	C, END2
BD6F'	2B	DC	HL CHRBIF
BD72'	2B	0C	(HL), 0"
BD74'	7E	HL	END5:
;			HL
END3:	INC	CP	"
;	LD	JR	Z, END6
END4:	LD	HL	DNZ
;	LD	HL	END5:
END5:	LD	HL	END1
;	LD	HL	HL CHRBIF
END6:	LD	HL	D.H.
;	LD	HL	E.L.
END7:	LD	HL	B, 12

日數計算

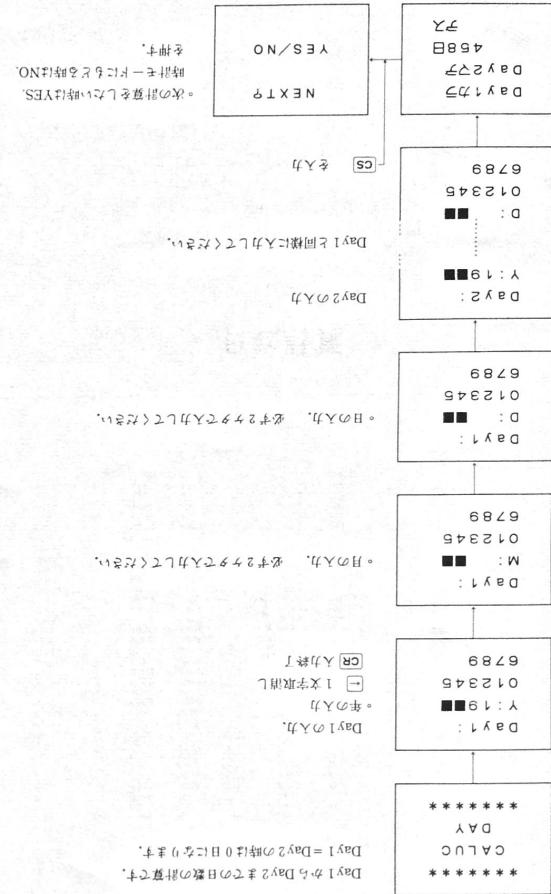
両方の日付は1900年から1999年までに入っていなければなりません。

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。

次は日の入力です。入力方法は年月の入力と同じです。入力は必ず2桁となるようにしてください。

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

時は、**YES** を押します、**NO** を押すと時計モードにもどります。















## 相性診断

:uシク->θ2ウソ>、2ウシ  
>...=&\$.-.ニウシケル

B9E0 3A 75 BC B7 20 0A 3E 30 32 75 BF 3E 01 32 75 BC :C2  
B9F0 3E 2E 18 06 C6 24 18 02 C6 0F 57 3A 75 BC B9 30 :0E

:uシク->θ2ウソ>、2ウシ  
>...=&\$.-.ニウシケル

Sum B4 29 48 90 F5 04 16 DB 00 90 92 4B 38 A7 86 8F :00

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

ASCII

BA00 A8 5F B7 20 05 CD 89 BA 7A FE 30 28 06 7B 3C 32 :B6  
BA10 75 BC 7A 16 00 21 75 BF 19 77 23 AF 77 C9 21 7D :56

BA20 BC 5E 2B 56 2B 4E 2B 46 2B 7E C9 CD 81 42 21 46 :EE  
BA30 BA CD FB 41 CD 91 43 FE 34 28 05 FE 36 20 F5 E1 :ED  
BA40 CD CC B9 C3 97 B9 45 52 52 4F 52 21 21 70 BF 11 :71  
BA50 79 BC 18 26 21 7E BC 11 70 BF 10 21 70 BF 11 :A5  
BA60 7E BC 18 16 F5 CD 89 BA CD C7 5B F1 C3 97 BA 21 :82  
BA70 75 BF 11 F2 01 0C 00 18 03 01 05 ED B0 C9 :8A  
BA80 89 06 01 0E 50 ED' F5 21 75 BF E5 7E B7 28 08 :CC  
BA90 0B 36 00 23 10 FB C9 F5 21 75 BC E6 86 7E :AC  
BAA0 FE 45 CA 34 BB 23 18 F4 32 77 FB 00 BB FE 2E 28 10 :AE  
BAC0 36 30 23 E5 11 81 BF AF ED 52 E1 CA F0 BB 18 E7 :02  
BAD0 36 00 C3 00 21 7C BB 21 7E FE 39 C2 F0 BB 36 :DF  
BAE0 7E B7 28 04 16 F7 E1 7E FE 39 C2 F0 BB 36 :DF  
BAF0 3B 2B E5 11 75 BF AF ED 52 E1 28 10 FE 2E 28 :5E

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

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

ASCII

BB00 F0 3C 77 FE 3A C2 F0 BB 36 30 18 E5 7E FE 2D 28 :7C  
BB10 1E FE 2E 28 09 3C 77 FE 3A C2 F0 BB 36 30 01 08 :42  
BB20 00 21 7C BF 11 FD ED BA 23 01 31 C3 F0 BB 36 30 01 :47  
BB30 00 18 E1 0E 01 7E FE 45 28 18 23 0C 79 FE :A3  
BB40 08 20 F4 05 7E FE 45 28 05 36 00 23 18 F6 22 77 :EF  
BB50 8C BC E1 18 03 22 77 BC 2B FE 39 C2 F0 BB 36 30 :C0  
BB60 2B E5 11 75 BF AF ED 52 E1 28 10 7E FE 2E 28 F0 :1E  
BB70 77 FE 3A C2 F0 BB 36 30 18 03 23 36 31 21 00 :85  
BB80 3C 77 FE 3A C2 F0 BB 36 31 18 03 23 36 31 21 00 :85  
BB90 BF 7E B7 20 03 2B 18 F9 F5 E5 2B 7E FE 45 28 10 :E3  
BBB0 23 54 5D 7D 20 24 21 03 2B 18 F9 F5 E5 2B 7E FE 45 28 10 :E3  
BBB1 36 39 2B 7E FE 2D 20 24 21 03 2B 18 F9 F5 E5 2B 7E FE 45 28 10 :E3  
BBC0 6B 23 12 B7 CA F0 BB 13 23 04 18 F9 F5 E5 2B 7E FE 45 28 10 :E3  
BBC1 3C 22 F0 BB 36 30 18 03 23 36 31 21 00 :85  
BBC2 11 80 BF 7E FE 45 28 05 12 2B 1B 18 F6 3E 31 12 :25  
BBC3 3A 77 BC B7 28 46 21 7C FE 2D 28 06 21 :0A  
BBC4 3A 77 BC B7 28 46 21 7C FE 2D 28 06 21 :0A

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

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

ASCII

BC00 00 01 E3 18 37 23 7E D6 30 FE 03 38 64 FE 08 30 :AD  
BC10 60 F5 CD 6F BA CD 89 BA 21 75 FB 1A B7 28 0D FE :43  
BC20 23 C1 05 36 30 23 00 11 F2 BF 1A B7 28 0D FE :43  
BC30 2E 28 01 77 13 23 04 78 FE 09 38 FB 21 75 BF 54 5D :49  
BC40 BC 7E 23 FE 28 04 18 F8 18 24 21 75 BF 54 5D :49  
BC50 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BC60 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BC70 C9 F1 C3 89 04 FC CD 4C C3 00 00 :FE  
BC80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :FE  
BC90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :FE  
BCA0 60 F5 CD 89 BA 21 75 FB 1A B7 28 0D FE :43  
BCB0 23 C1 05 36 30 23 00 11 F2 BF 1A B7 28 0D FE :43  
BCC0 2E 28 01 77 13 23 04 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCD0 86 BC 7E 23 FE 28 04 18 F8 18 24 21 75 BF 54 5D :49  
BCE0 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCE1 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCE2 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCE3 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCE4 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCE5 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCE6 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCE7 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCE8 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCE9 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEA 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEB 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEC 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCED 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEF 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEG 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEH 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEI 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEJ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEK 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEL 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEM 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEN 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEO 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEP 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEQ 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCER 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCES 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCET 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEU 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEV 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEW 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEX 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEY 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEZ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEA 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEB 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEC 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCED 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEE 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEF 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEG 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEH 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEI 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEJ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEK 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEL 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEM 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEO 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEP 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEQ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCER 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCES 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCET 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEU 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEV 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEW 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEX 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEY 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEZ 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEA 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEB 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEC 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCED 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEE 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEF 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEG 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEH 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEI 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEJ 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEK 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEL 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEM 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEO 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEP 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEQ 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCER 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCES 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCET 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEU 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEV 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEW 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEX 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEY 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEZ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEA 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEB 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEC 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCED 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEE 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEF 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEG 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEH 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEI 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEJ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEK 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEL 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEM 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEO 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEP 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEQ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCER 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCES 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCET 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEU 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEV 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEW 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEX 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEY 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEZ 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEA 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEB 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEC 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCED 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEE 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEF 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEG 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEH 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEI 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEJ 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEK 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEL 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEM 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEO 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEP 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEQ 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCER 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCES 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCET 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEU 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEV 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEW 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEX 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEY 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEZ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEA 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEB 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEC 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCED 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEE 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEF 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEG 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEH 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEI 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEJ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEK 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEL 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEM 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEO 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEP 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEQ 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCER 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCES 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCET 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB 21 75 BF 54 5D :49  
BCEU 86 BC 7E 28 08 FB 30 06 36 00 18 EC 0C 30 13 04 2B :92  
BCEV 86 BC 7E 28 03 23 10 F9 78 FE 09 38 FB

18

## バイオリズム

アシノヲハアシヨウハアシヲ  
ノ...▲.W...■  
ノ...!フ.▲<G>...  
.1.Y...Q...  
..C.S...4.e110  
1011011.0.BB1

BCA8	B1	BC	2F	A6	77	C9	CD	B1	BC	AE	77	C9	CD	B1	BC	AG	:8A
BCC8	C9	CD	6F	8F	8F	E6	B3	57	87	87	82	87	82	87	82	87	:5E
BCC8	16	00	5F	7D	21	A6	B0	19	E6	87	3C	47	3E	01	01	:5D	
BCC8	FD	C9	00	00	00	31	00	59	00	90	01	20	01	51	01	:D5	
BCF8	82	12	02	43	82	73	83	04	03	34	03	65	31	29	31	:2F	
BCF8	31	31	31	30	31	31	00	23	00	28	00	33	42	69	:AE		
Sum	D9	E3	98	11	B5	48	43	64	27	E1	C6	9F	CA	3A	97	:7A	

Sum

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

ASCII

BD08	72	74	68	20	31	42	69	72	74	68	20	32	30	31	32	:B8
BD18	34	35	36	37	38	39	20	20	4E	45	58	54	20	3F	59	:9E
BD28	45	53	2F	4E	59	3A	31	39	20	20	4D	3A	20	20	:88	
BD38	20	20	28	44	3A	20	20	20	20	14	1F	13	1E	12	:11	
BD48	11	1C	06	19	0C	16	01	19	0C	16	0B	16	0A	15	:30	
BD58	89	14	08	13	07	12	06	11	05	10	05	0F	04	0E	:B4	
BD68	03	03	0B	03	0A	03	09	03	08	03	07	03	06	04	:5D	
BD78	84	05	03	06	02	07	01	08	01	09	0A	00	0B	00	:47	
BD88	6C	00	0D	0E	00	05	01	10	01	11	02	12	03	13	:87	
BD98	14	05	00	00	00	00	00	00	00	00	00	00	00	00	:19	
BDA8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
BDB8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
BDC8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
BDD8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
BDE8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	
BDF8	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	:0F

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。  
人間の身体、感性、知性は日々、その調子を変えてます。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案されたのがバイオリズムです。  
「恥はぬ先の杖」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。

誕生日の入力

年(西暦)の入力

月の入力

日(西暦)の入力

1文字削除

CR 入力終了

Birth:  
M: ■■  
012345  
6789

Birth:

M:

012345

6789

○月の入力 必ず2ヶタで入力してください。  
○日(西暦)の入力 必ず2ヶタで入力してください。

1文字削除

CR 入力終了

Birth:  
M: ■■  
012345  
6789

Birth:

M:

012345

6789

○月の入力 必ず2ヶタで入力してください。  
○日(西暦)の入力 必ず2ヶタで入力してください。

1文字削除

CR 入力終了

Today ?

Birth:

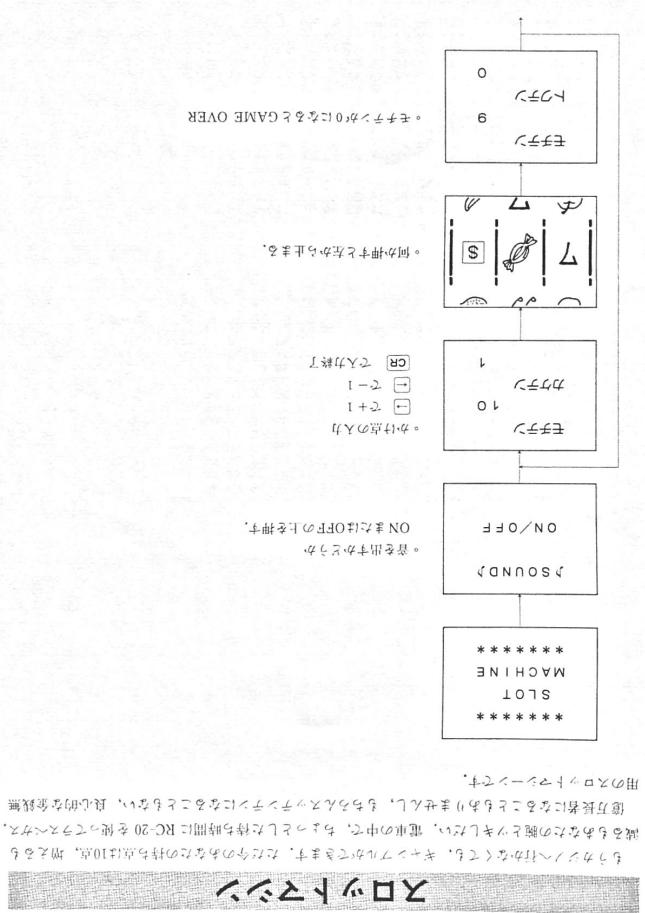
M:

012345

6789

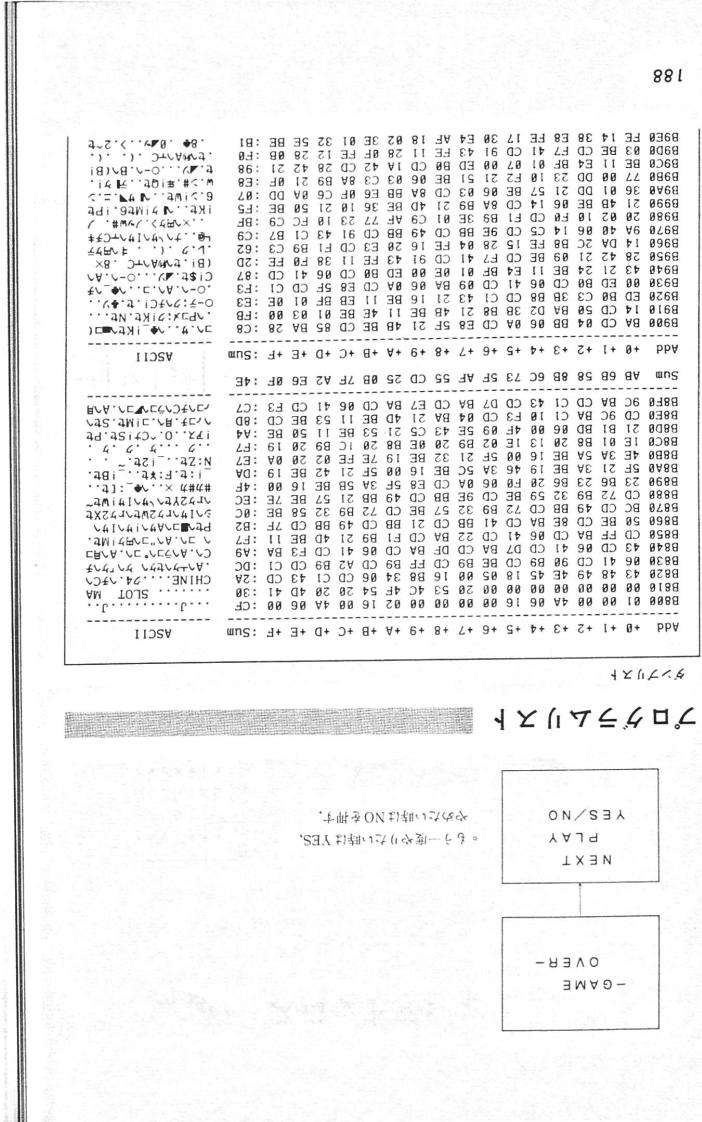
○今日のバイオリズムを知りたい時はYES  
何年何月何日のバイオリズムを知りたい、  
時はNOを押してください。  
YESを押すとその日のバイオリズムを





88

681



## もぐらたたき

さて、RC-20の中に居てもぐらとあなたと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。

```
*****  
RC-20  
ATTACK  
*****
```

```
) SOUND)  
ON/OFF
```

。音を出すかどうか。  
ONまたはOFFの上を押す。

```
TIME  
POINT
```

50 0

左がTIME  
右がその面でのPOINT

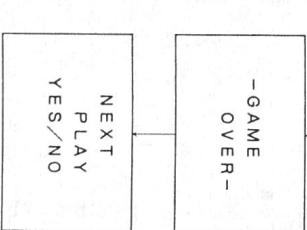
。TIMEの上を押す。TIME 0で終り。

SCORE  
40

。POINT 20以上で次の面へ行く。  
POINT 19以下でGAME OVER

Add	+θ	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
BD08	01	0A	00	ED	B0	13	13	13	13	DD	23	01	08	00	FD	09	:03	..O...#...
BD18	C9	00	7C	60	60	61	63	7E	7C	60	00	00	00	00	00	7E	:BD	J,1,1...act...;
BD28	FE	00	00	00	00	03	04	08	08	78	44	2A	15	8C	30	:CC	xD*...0	
BD38	A8	54	22	1E	10	10	20	C0	00	00	00	00	00	00	00	:3F	<T".....;	
BD48	4C	70	1F	60	90	90	60	98	24	24	18	50	80	01	02	:9A	L.P. ....;	
BD58	0A	09	00	19	24	42	41	E0	10	08	04	04	04	04	08	:0A	\$BA=....	
BD68	E0	03	04	08	10	19	14	14	08	04	03	E0	10	48	04	:24	=H.\$	
BD78	04	84	08	10	E0	00	00	00	00	19	39	50	0F	00	00	:2E	=0P....	
BD88	38	C4	02	02	02	02	C4	38	03	05	0A	15	1A	1A	15	:72	81....;	
BD98	0A	05	03	E0	50	A8	54	2C	2C	54	A8	50	00	3F	3C	:3D	=PT..,TP=?K	
BDA8	3B	28	3B	3D	3F	00	FE	DE	6E	02	6E	02	9E	35	:3D	;:;?..,n,n,n,		
BDB8	FE	1E	05	07	11	6A	0F	14	15	C0	02	D0	02	C7	:A3	:\$V=,v=,		
BDC8	02	00	01	D0	02	DA	02	E0	02	DA	02	E0	02	DE	03	:32	....=,=,=,	
BDD8	FF	0E	01	E0	01	E0	01	:32	♦..,=,+1,J,-,I									
BDE8	01	E8	02	00	FF	82	02	8F	02	89	62	9B	02	82	02	:41	♦..,=,+1,J,-,I	
BDF8	02	9B	03	00	FF	D3	C1	C3	DD	B6	C4	:38	♦..,=,+1,J,-,I	♦..,=,+1,J,-,I				
Sum	29	46	22	60	67	A3	F4	EA	25	9C	E9	15	F8	4F	C4	:E1	:84	ASCII

Add	+θ	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
BEB8	B8	C3	DD	4F	4E	2F	4F	46	46	59	45	53	2F	4E	4F	9A	:56	←,ON/OFF,FES,NOL
BEC8	53	4F	55	4E	44	9A	4D	47	41	4D	45	28	20	20	20	:39	SOUND, GAME	
BED8	56	45	52	2D	45	54	20	45	28	20	20	20	4C	41	:D6	NEXT PLA		
BEE8	59	20	01	02	03	04	05	06	07	00	04	06	01	07	02	:A9	Y.....	
BEF8	05	03	00	06	05	02	03	01	07	04	09	00	00	00	00	:24	.....	
BEG8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEH8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00	.....	
BEE9	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	ASCII	



。もう一度やりたい時はYESを押す。  
NOで時計モードになる。

プログラムリスト

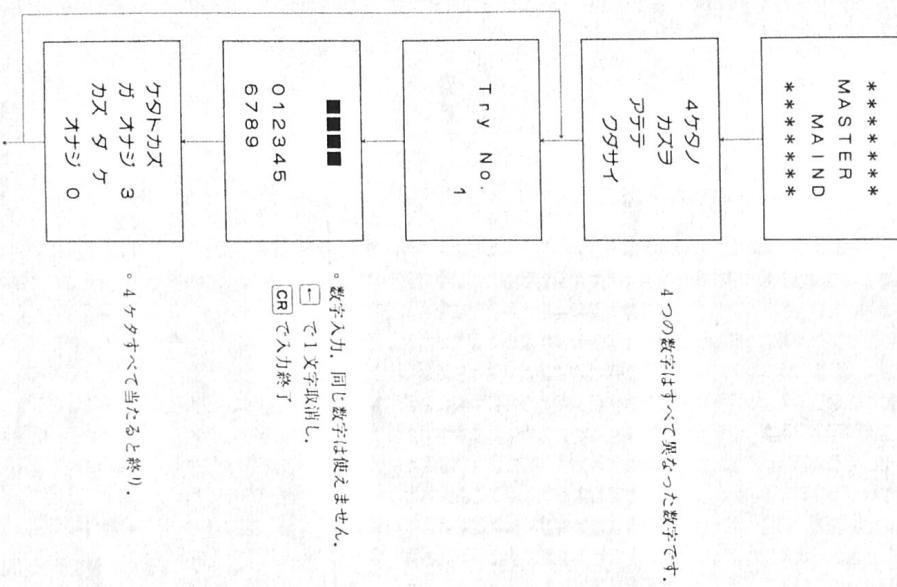
ダンプリスト

Add	+g	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	ASCII
B800	01	00	00	00	16	04	16	00	00	00	00	02	16	00	16	04	00	63
B810	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	RC-20 AT
B820	54	41	43	46	20	18	05	00	00	00	00	20	41	54	F9	..	AT	
B830	C1	A3	43	46	41	CD	B6	43	CD	D6	00	00	00	00	00	00	00	TICK ..
B840	CD	A6	B9	CD	I1	43	CD	06	41	CD	59	B9	CD	C1	43	CD	8E	
B850	BD	46	43	CD	D1	B9	06	00	00	00	00	00	00	00	00	00	GC#(+) ..	
B860	3A	19	19	PC	F5	CD	8F	88	CD	88	CD	D1	B9	F1	47	3A	A5	
B870	19	BC	B8	28	0E	FE	28	00	00	00	00	00	00	00	00	00	..	
B880	5F	18	DA	BC	B7	20	D4	21	D2	00	00	00	00	00	00	00	00	..
B890	FE	50	28	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B8A0	80	ED	B9	21	D9	11	BC	00	00	00	00	00	00	00	00	00	..	
B8B0	41	CD	B9	21	17	BC	00	00	00	00	00	00	00	00	00	00	..	
B8C0	5F	3A	1D	BC	3D	28	00	32	00	00	00	00	00	00	00	00	..	
B8D0	32	1D	BC	CD	I1	43	21	CD	B9	11	ED	00	00	00	00	00	00	
B8E0	21	17	BC	11	F5	CD	ED	B9	11	ED	00	00	00	00	00	00	..	
B8F0	E8	5F	3A	19	BC	FE	28	D2	43	B8	CD	C1	43	21	FA	BB	E8	
Sum	71	E9	C3	B4	C5	8A	BD	94	25	29	01	20	C8	D5	1E	AE	:49	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	
B900	11	EB	BF	01	0E	00	ED	B0	CD	B6	41	CD	BA	B9	00	0A	:CB	
B910	CD	E8	5F	CD	I1	43	21	08	BC	11	E4	BF	01	0E	00	ED	:7A	
B920	B0	CD	00	00	41	CD	28	42	00	00	00	00	00	00	00	00	..	
B930	FE	11	30	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B940	96	B9	C3	9A	40	21	17	BC	00	00	00	00	00	00	00	00	..	
B950	1D	BC	C9	A9	77	23	00	FB	CF	00	32	19	BC	30	50	32	..	
B960	91	BC	C9	21	F3	BC	11	E4	BF	01	07	00	ED	B0	CD	1A	AE	
B970	42	CD	24	21	E7	BB	CD	F7	41	CD	91	43	FE	11	28	19	..	
B980	00	12	28	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B990	3E	01	32	1F	BC	C9	3E	00	00	00	00	00	00	00	00	00	..	
B9A0	00	00	00	00	50	79	C9	21	6B	18	12	21	BB	18	00	00	..	
B9B0	21	99	BB	18	00	00	00	00	00	00	00	00	00	00	00	00	..	
B9C0	B7	C8	4E	23	45	23	78	FF	EE	00	00	00	00	00	00	00	..	
B9D0	FC	21	11	E4	BF	CD	B6	13	13	13	21	19	BC	37	00	00	..	
B9E0	CD	E6	B9	C3	1A	42	00	00	00	00	00	00	00	00	00	00	..	
B9F0	1E	BC	CD	F5	B9	7E	23	CD	B1	32	1E	BC	18	00	00	00	..	
Sum	9D	E0	16	8E	D7	DD	CD	39	55	26	2B	7B	BB	E7	4E	A2	:8E	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	
B900	11	EB	BF	01	0E	00	ED	B0	CD	B6	41	CD	BA	B9	00	0A	:CB	
B910	CD	E8	5F	CD	I1	43	21	08	BC	11	E4	BF	01	0E	00	ED	:7A	
B920	B0	CD	00	00	41	CD	28	42	00	00	00	00	00	00	00	00	..	
B930	FE	11	30	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B940	96	B9	C3	9A	40	21	17	BC	00	00	00	00	00	00	00	00	..	
B950	1D	BC	C9	A9	77	23	00	FB	CF	00	32	19	BC	30	50	32	..	
B960	91	BC	C9	21	F3	BC	11	E4	BF	01	07	00	ED	B0	CD	1A	AE	
B970	42	CD	24	21	E7	BB	CD	F7	41	CD	91	43	FE	11	28	19	..	
B980	00	12	28	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B990	3E	01	32	1F	BC	C9	3E	00	00	00	00	00	00	00	00	00	..	
B9A0	00	00	00	00	50	79	C9	21	6B	18	12	21	BB	18	00	00	..	
B9B0	21	99	BB	18	00	00	00	00	00	00	00	00	00	00	00	00	..	
B9C0	B7	C8	4E	23	45	23	78	FF	EE	00	00	00	00	00	00	00	..	
B9D0	FC	21	11	E4	BF	CD	B6	13	13	13	21	19	BC	37	00	00	..	
B9E0	CD	E6	B9	C3	1A	42	00	00	00	00	00	00	00	00	00	00	..	
B9F0	1E	BC	CD	F5	B9	7E	23	CD	B1	32	1E	BC	18	00	00	00	..	
Sum	9D	E0	16	8E	D7	DD	CD	39	55	26	2B	7B	BB	E7	4E	A2	:8E	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	
B900	11	EB	BF	01	0E	00	ED	B0	CD	B6	41	CD	BA	B9	00	0A	:CB	
B910	CD	E8	5F	CD	I1	43	21	08	BC	11	E4	BF	01	0E	00	ED	:7A	
B920	B0	CD	00	00	41	CD	28	42	00	00	00	00	00	00	00	00	..	
B930	FE	11	30	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B940	96	B9	C3	9A	40	21	17	BC	00	00	00	00	00	00	00	00	..	
B950	1D	BC	C9	A9	77	23	00	FB	CF	00	32	19	BC	30	50	32	..	
B960	91	BC	C9	21	F3	BC	11	E4	BF	01	07	00	ED	B0	CD	1A	AE	
B970	42	CD	24	21	E7	BB	CD	F7	41	CD	91	43	FE	11	28	19	..	
B980	00	12	28	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B990	3E	01	32	1F	BC	C9	3E	00	00	00	00	00	00	00	00	00	..	
B9A0	00	00	00	00	50	79	C9	21	6B	18	12	21	BB	18	00	00	..	
B9B0	21	99	BB	18	00	00	00	00	00	00	00	00	00	00	00	00	..	
B9C0	B7	C8	4E	23	45	23	78	FF	EE	00	00	00	00	00	00	00	..	
B9D0	FC	21	11	E4	BF	CD	B6	13	13	13	21	19	BC	37	00	00	..	
B9E0	CD	E6	B9	C3	1A	42	00	00	00	00	00	00	00	00	00	00	..	
B9F0	1E	BC	CD	F5	B9	7E	23	CD	B1	32	1E	BC	18	00	00	00	..	
Sum	9D	E0	16	8E	D7	DD	CD	39	55	26	2B	7B	BB	E7	4E	A2	:8E	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	
B900	11	EB	BF	01	0E	00	ED	B0	CD	B6	41	CD	BA	B9	00	0A	:CB	
B910	CD	E8	5F	CD	I1	43	21	08	BC	11	E4	BF	01	0E	00	ED	:7A	
B920	B0	CD	00	00	41	CD	28	42	00	00	00	00	00	00	00	00	..	
B930	FE	11	30	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B940	96	B9	C3	9A	40	21	17	BC	00	00	00	00	00	00	00	00	..	
B950	1D	BC	C9	A9	77	23	00	FB	CF	00	32	19	BC	30	50	32	..	
B960	91	BC	C9	21	F3	BC	11	E4	BF	01	07	00	ED	B0	CD	1A	AE	
B970	42	CD	24	21	E7	BB	CD	F7	41	CD	91	43	FE	11	28	19	..	
B980	00	12	28	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B990	3E	01	32	1F	BC	C9	3E	00	00	00	00	00	00	00	00	00	..	
B9A0	00	00	00	00	50	79	C9	21	6B	18	12	21	BB	18	00	00	..	
B9B0	21	99	BB	18	00	00	00	00	00	00	00	00	00	00	00	00	..	
B9C0	B7	C8	4E	23	45	23	78	FF	EE	00	00	00	00	00	00	00	..	
B9D0	FC	21	11	E4	BF	CD	B6	13	13	13	21	19	BC	37	00	00	..	
B9E0	CD	E6	B9	C3	1A	42	00	00	00	00	00	00	00	00	00	00	..	
B9F0	1E	BC	CD	F5	B9	7E	23	CD	B1	32	1E	BC	18	00	00	00	..	
Sum	9D	E0	16	8E	D7	DD	CD	39	55	26	2B	7B	BB	E7	4E	A2	:8E	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	
B900	11	EB	BF	01	0E	00	ED	B0	CD	B6	41	CD	BA	B9	00	0A	:CB	
B910	CD	E8	5F	CD	I1	43	21	08	BC	11	E4	BF	01	0E	00	ED	:7A	
B920	B0	CD	00	00	41	CD	28	42	00	00	00	00	00	00	00	00	..	
B930	FE	11	30	00	00	00	00	00	00	00	00	00	00	00	00	00	..	
B940	96	B9	C3	9A	40	21	17	BC	00	00	00	00	00	00	00	00	..	
B950	1D	BC	C9	A9	77</td													

## 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た数をかべ、何度かあなたが質問し、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。

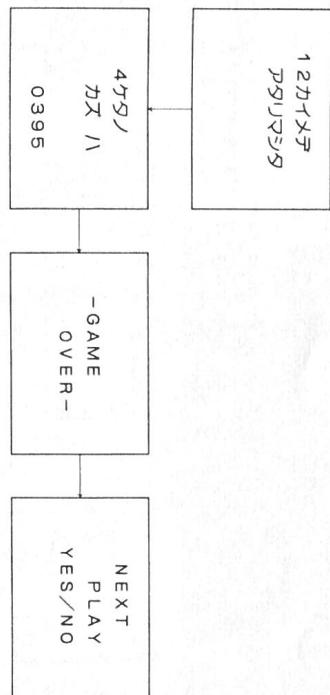
できるだけ少ない回数で当ててください。RC-20は決して嘘つきません。頭をとびきり回転させて、RC-20の爆をあかしてください。



12カイメテ アタマシタ	
Add +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F : Sum	ASCII1
B800 01 00 00 00 91 03 16 00 00 00 00 00 02 16 00 91 03 00 :57	.....T.....T..
B810 00 00 00 00 00 00 00 00 4D 41 53 54 45 52 20 20 20 :C	.....MASTER
B820 4D 41 45 44 44 48 85 00 16 BB 7B 83 CD C1 43 CD :70	MAIND .. .21.1.7C>
B830 06 41 CD 4C BA 21 F6 BA 11 E4 BF 01 IC 00 ED B0 :59	.A\N\G\H\T\A\O\O
B840 CD 06 41 3E 01 32 9A BB CD GF BA CD 4C BA CD C1 :31	\.A>.2\N\G\H\T\A\O\O
B850 43 CD 06 41 EF BA 11 EB BF 01 07 00 ED B0 21 :A2	C\A\I\N\G\H\T\A\O\O
B860 9A BB 11 F7 BF CD B2 BA CD 06 41 CD 4C BA CD C1 :CA	\.P\N\G\H\T\A\O\O
B870 43 CD 06 41 21 12 BB CD 15 41 21 E5 BF 3E A 06 :11	C\A\I\N\G\H\T\A\O\O
B880 04 77 23 10 FC DD 21 E5 BF DD 36 00 A0 CD 1A 42 :28	\.W#. .\N\G\H\T\A\O\O
B890 CD 58 BA FE 15 28 F9 C6 30 DD 77 00 CD 1A 42 DD :63	\.A\J. (\.B\O\W\A\B\Y
B8A0 36 01 A0 CD 1A 42 CD 01 CD 1A 42 DD 36 02 A0 CD 1A :F2	6. \.B\N\X\O. (\.B\O\W\A\B\Y
B8B0 BE 00 28 F2 DD 77 01 CD 1A 42 DD 36 02 A0 CD 1A :F2	\.B\N\X\O. (\.B\O\W\A\B\Y
B8C0 42 CD 55 BA FE 15 28 D9 C6 30 DD 77 00 CD 1A 42 DD :BB	\.B\N\X\O. (\.B\O\W\A\B\Y
B8D0 BE 02 28 ED DD 77 02 CD 1A 42 DD 36 03 A0 CD 1A :F1	\.B\N\X\O. (\.B\O\W\A\B\Y
B8E0 42 CD 58 BA FE 15 28 D2 C6 30 DD 77 03 CD 1A 42 CD :F6	\.B\N\X\O. (\.B\O\W\A\B\Y
B8F0 BE 01 28 ED DD 77 03 CD 1A 42 CD :CE	\.B\N\X\O. (\.B\O\W\A\B\Y
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	ASCII1
B900 91 43 B7 28 FA FE 33 28 D1 FE 34 28 F2 21 E5 BF :E0	\.C\#( 3 (L 4 \.K\Y
B910 11 96 BB 01 04 ED B0 21 4D BB 11 E4 BF 01 :FE	\.P...O-1M\#.\A\Y
B920 08 ED B0 CD 06 41 06 05 04 ED B0 21 96 BB 11 92 BB :C9	O-\A..@!1944
B930 1A BE CC 56 BA 23 13 10 F7 41 79 32 F1 BF DD 21 :8B	\.T\#V\#..#AY2B\#!
B940 96 BB 21 92 BB 08 00 92 BB 08 00 92 BB 08 00 92 BB :8E	\.1\+4\#..#AY2B\#!
B950 02 CC 56 BB DD BE 03 CC 56 BA 23 7E DD BE 03 CC :60	\.T\#V\#..#AY2B\#~\#.
B960 56 BA DD BE 02 CC 56 BA DD BE 03 CC 56 BA 23 TE :A4	\.T\#V\#..#AY2B\#~\#.
B970 DD BE 00 CC 56 BA DD BE 01 CC 56 BA DD BE 01 CC :59	\.T\#V\#..#AY2B\#~\#.
B980 56 BA 23 7E DD BE 00 CC 56 BA DD BE 01 CC 56 BA :A0	\.T\#V\#..#AY2B\#~\#.
B990 DD BE 02 CC 56 BA DD BE 01 CC 56 BA DD BE 01 CC :D4	\.T\#V\#..#AY2B\#~\#.
B9A0 BA C1 78 FE 34 26 3A 9A BB FE 99 28 09 C6 01 :91	\.T\#V\#..#AY2B\#~\#.
B9B0 27 32 9A BB C3 4E BB CD C1 43 21 69 BB 11 EB BF :48	\.T\#V\#..#AY2B\#~\#.
B9C0 01 0E 00 ED B0 CD 06 41 CD 4C BA 18 1D CD C1 :43	\.T\#V\#..#AY2B\#~\#.

## プログラムリスト

ダンプリスト



ボーカーゲーム

アプローチ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]キーを押して暗け点(BET)を決めます。

なお、**MODE**キーでりにもどります。  
**[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。カードが決まつたら**[CR]**キーを押してください。Hマークのないカードは

ここで、役ができて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。  
勝手点に倍率を掛けた点数がクリエットに加算され、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

—

卷之三

ounds

ON OFF

3 CREDIT

B E T

カードが配られる。  
残しておきたいカードを、下の「○」を押して進む。  
カードが決まつたらCRキーを押す。  
また、「MODE」キーでキャンセルできる。

The diagram illustrates the correspondence between the four cardinal directions and the four main gates:

- North (N):** Associated with Gate 1 (Top Gate).
- South (S):** Associated with Gate 2 (Bottom Gate).
- East (E):** Associated with Gate 3 (Right Gate).
- West (W):** Associated with Gate 4 (Left Gate).

B9D0	21	85	BB	11	E6	BF	01	0C	00	ED	B0	21	9A	BB	11	E4	:2C
B9E0	BF	CD	B2	BA	CD	06	41	CD	4C	BA	CD	C1	43	21	77	BB	:03
B9F0	11	E4	BF	01	0E	00	ED	B0	21	92	BB	11	FB	BF	01	04	:9E
Sum	8D	32	A5	DE	49	34	2B	7D	F2	BA	B9	61	C7	2F	76	37	:D0
Add	#0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
BAA0	00	ED	BB	CD	06	41	CD	4C	BA	CD	C1	43	21	31	BB	11	:73
BA10	EB	BF	01	0E	00	ED	B0	CD	06	41	CD	4C	BA	CD	C1	43	:8E
BA20	21	3F	BB	11	E4	BF	01	0E	00	ED	B0	CD	06	41	CD	2B	:84
BA30	42	21	BB	CD	F7	41	CD	91	43	FE	11	38	0F	FE	14	:31	
BA40	DA	2C	BB	28	E9	FE	17	30	E5	C3	9A	40	06	9A	CD	E9	:5B
BA50	F6	06	EA	C3	8F	5F	0C	91	43	7B	28	FA	CD	A1	36	:7A	
BA60	40	FE	07	38	F3	EE	91	30	EC	07	97	D9	DD	FF	09	@..8H	:J
BA70	21	92	BB	CD	A3	BA	DD	77	00	47	CD	A3	BA	BB	8A	:37	
BA80	77	01	4F	CD	A3	BA	BB	29	8B	28	77	02	0D	66	W..	:J	
BA90	CD	A3	BA	BB	28	FA	B9	28	F7	DD	46	02	B8	28	F1	:DD	
BAA0	77	03	C9	CD	92	DB	BB	0F	0F	ED	06	81	00	30	F4	C6	:B9
BAB0	3A	98	BB	3D	28	11	32	9B	BB	78	FE	30	2B	87	3E	01	:A1
BA20	32	9B	BB	18	02	64	67	11	81	78	12	13	09	9A	BB	54	:61
BA30	29	BB	19	7D	BB	8A	66	11	81	19	22	9D	BB	7C	93	54	:91
BA40	72	79	28	4E	6F	2E	20	34	B9	C0	C9	20	20	20	B6	:C2	J
Sum	4B	8C	85	8D	3D	97	94	61	D8	D8	57	72	F0	45	63	3E	:B3
Add	#0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
BB00	BD	DE	A6	20	20	B1	C3	C3	20	20	20	B8	C0	DE	:AE	X..Y	:J
BB10	BB	BB	B2	30	31	32	33	34	35	36	37	38	39	20	4F	4E	:57
BB20	2F	4F	46	46	46	59	53	52	4F	4E	4F	53	54	4E	44	:EA	
BB30	9A	20	47	41	49	20	20	20	20	4F	56	45	52	2D	20	:EA	
BB40	4E	45	58	54	28	20	20	20	4C	41	59	20	B9	CD	C4	:52	
BB50	BD	20	B6	DE	B5	CD	B5	CD	20	B6	DE	C0	CD	C4	:28	NEXT,	
BB60	B9	20	20	20	B5	BC	DE	30	B0	B1	B7	D7	D2	20	20	:4E	
BB70	29	20	29	CF	BC	AE	B3	B2	20	34	B9	C0	20	20	20	:F2	
BB80	BB	BB	BB	DE	CA	AE	B3	B2	20	B2	CD	20	BB	CB	0A	:KA	
BB90	C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:KA	
BB00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:KA	
BBF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:KA	
Sum	94	0B	66	F2	51	9A	5D	FC	E7	37	33	42	68	E0	19	E2	:5D

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum
BBB0	BD	DE	A6	20	20	B1	C3	C3	20	20	20	20	20	B8	C0	DE	:AE
BBB1	BF	BB	30	31	32	33	34	35	36	37	38	39	39	20	4F	55	:F57
BBB2	FB	FF	45	46	50	53	52	4F	4E	54	53	4F	55	4F	4E	44	:EA
BBB3	FA	FB	47	49	47	41	45	20	20	20	20	4F	56	45	52	2D	:EA
BBB4	AE	45	58	54	20	20	20	20	20	20	4C	41	59	20	B9	C0	:C4
BBB5	BD	BB	20	20	20	20	20	20	20	20	20	20	20	20	DE	C0	:D28
BBB6	B9	20	20	20	20	20	20	20	20	20	20	20	20	20	B0	B1	:B7
BBB7	20	20	CF	BC	AE	B3	B0	20	34	B9	C0	20	20	D2	20	20	:F2
BBB8	B6	20	CA	20	B6	B2	D2	C3	20	20	B1	C0	20	D8	GF	BC	:DA
BBB9	C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:C0
BBB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB7	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
BBB9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
Sum	94	0B	66	F2	51	9A	5D	FC	E7	37	33	42	68	E0	19	2E	:5D

67



カードを引いてくる。

この場合、2ペアの役ができるとしている。また1ペアは役と見なされない。

7  
2 PAIR  
X2 役の表示。

8 賭け点に倍率を掛けたものが

スコアとなる。

1

CREDIT  
22  
スコアがクレジットに加算される。  
WIN!

GAME  
OVER

クリエイティブがなくなると  
ゲームオーバー。

Replay?

プログラムリスト

۲۷۴

Sum		BF	AB	D3	FA	3D	1A	EE	B3	B4	2D	BA	C5	01	BD	BE	CD	3F	BA	C1	DD	:75		
Sum	BF0	05	21	38	BE	CD	2D	BA	C5	01	BD	BE	CD	3F	BA	:3D	FF	BA	:3D	FF	BA	:3D		
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum							
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum							
BB00	7E	0A	21	79	BE	CD	2D	BA	78	C5	01	E7	BE	CD	3F	BA	:3D	FF	BA	:3D	FF	BA	:3D	
BA10	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	07	
BA20	CD	BC	C1	DD	BC	A5	CD	BA	F5	CD	E5	CD	7B	39	CD	EE	1F	51	F1	06	01	BC	:27	
BA30	CB	27	63	5F	16	00	19	EB	68	26	00	29	29	29	29	29	29	29	29	29	29	29	29	
BA40	EB	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	
BA50	CD	FB	41	CD	47	43	43	43	43	43	43	43	43	43	43	43	43	43	43	43	43	43	43	
BA60	38	09	FE	32	28	E1	FE	34	28	ED	C9	D6	21	5F	16	00	EE	EE	EE	EE	EE	EE	EE	
BA70	21	93	BE	19	CB	46	20	DF	36	01	EB	29	29	29	29	29	29	29	29	29	29	29	29	
BA80	BC	11	BD	BE	CD	3F	3E	30	31	1E	SD	01	1E	SD	01	1E	SD	01	1E	SD	01	1E	SD	
BA90	BF	18	C4	6F	6F	6F	6F	6F	20	11	98	BE	66	04	C5	62	7B	62	7B	62	7B	62	7B	
BAAB	GB	11	23	4E	B9	38	03	77	79	12	10	F6	13	C1	10	EE	EE	EE	EE	EE	EE	EE	EE	
BABA	C9	21	9E	BE	7E	CD	D9	BA	C8	21	98	BE	7E	BE	7E	BE	7E	BE	7E	BE	7E	BE	7E	
BAC0	E6	G9	21	98	BE	7E	CD	D9	BA	C8	21	98	BE	7E	BE	7E	BE	7E	BE	7E	BE	7E	BE	
BADD	3E	08	CD	D9	BA	C8	CB	FB	06	04	3C	23	BE	CD	06	01	1E	23	BE	CD	06	01	1E	
BAE0	CD	21	B7	BE	CB	EE	C9	21	98	BE	0E	0E	0E	0E	0E	0E	0E	0E	0E	0E	0E	0E	0E	
BAF0	BE	20	63	79	B8	4F	CB	60	28	F2	3A	B7	BE	21	98	BE	7E	BE	7E	BE	7E	BE	7E	
Sum	A6	E6	DC	CC	6F	6A	AC	C8	14	40	B5	D5	B8	FA	31	22	FB	:AA						
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum							
BB00	32	B7	BE	C9	CD	C1	43	3E	08	11	73	BD	CD	D2	B8	CD	:EC							
BB10	E2	BC	C9	CD	CD	E8	5F	CD	85	CB	21	43	79	BD	CD	08	1E	00	EE	EE	EE	EE	EE	
BB20	3A	BE	47	E7	E7	7D	7D	CB	82	BB	28	09	23	78	E6	0F	CD	:E4						
BB30	82	BB	20	DB	DB	23	16	00	21	AC	BB	19	11	B5	BE	:2B								
BB40	7E	12	23	1B	7E	12	1D	64	01	IB	12	10	FC	3A	AC	BE	:F4							
BB50	47	CD	43	BC	C1	10	F9	DI	43	CD	43	44	11	7D	1A	1E	:57							
BB60	13	B7	20	FI	18	F9	CD	DI	B8	06	SD	0A	CD	E8	5F	FI	:F1							
BB70	04	28	91	CD	A4	BC	CD	63	42	21	BE	BB	CD	41	CD	41	CD	:C5						
BB80	BC	56	91	CD	8A	BB	84	CD	72	72	CB	23	BA	C8	18	F2	:44							
BB90	10	80	20	60	80	30	80	70	CB	01	02	64	08	80	05	00	09	09	09	09	09	09	09	
BBA0	0A	88	03	05	06	00	02	00	03	00	10	00	40	00	20	57	:D3							
BBC0	49	4E	21	28	CD	C1	43	AF	11	3F	BD	CD	D8	BB	3E	BE	:BE	:BE	:BE	:BE	:BE	:BE	:BE	
BBD0	11	46	BD	CD	D8	BA	FD	32	AC	72	BC	CD	72	BC	CD	8A	BC	:C4						
BBE0	CD	43	FE	32	28	20	FE	34	28	2B	FE	36	20	FI	3A	1C	:E3							
BBF0	AC	BE	FE	0A	30	EA	CD	2F	BC	DC	20	BC	38	E2	21	AC	:E3							
Sum	92	4D	9E	37	85	B4	E4	0F	1A	1B	BF	38	BB	66	51	F3	:71							
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum							

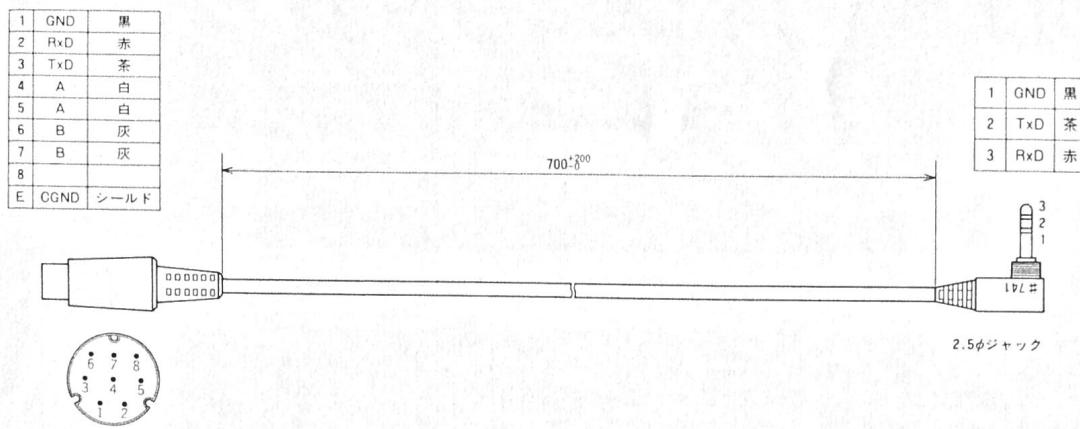
۱۰۳

ASCI

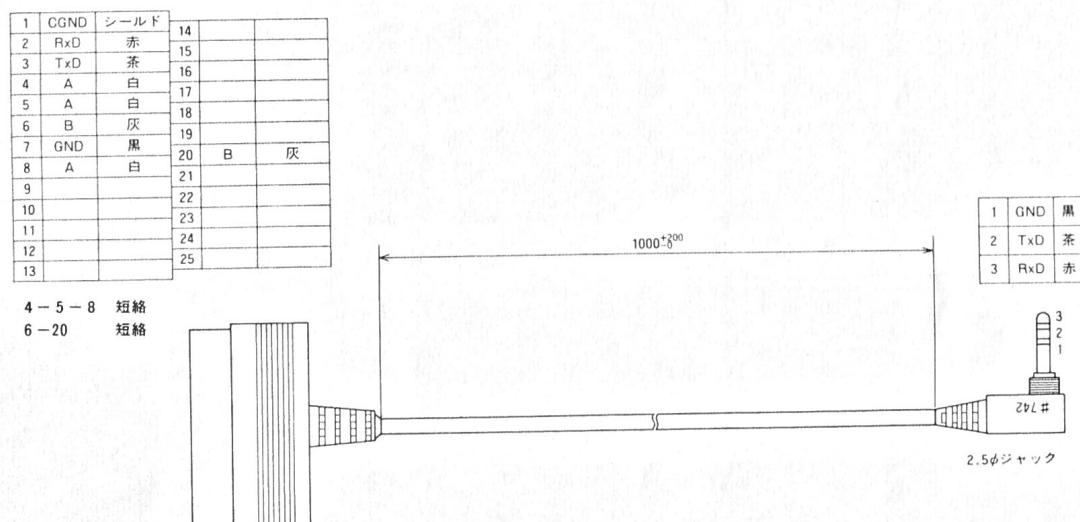
付録

RC-20とパソコンの接続ケーブル

#741(HC-80/88/40/41用)



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



付録

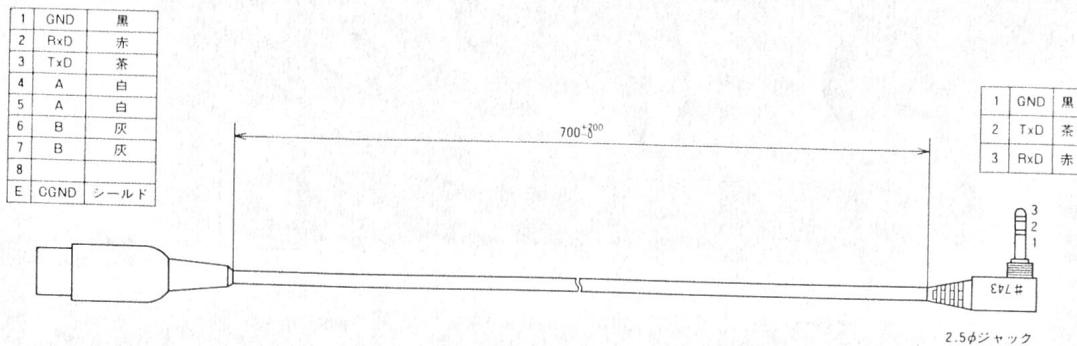
RC-20とパソコンの接続ケーブル



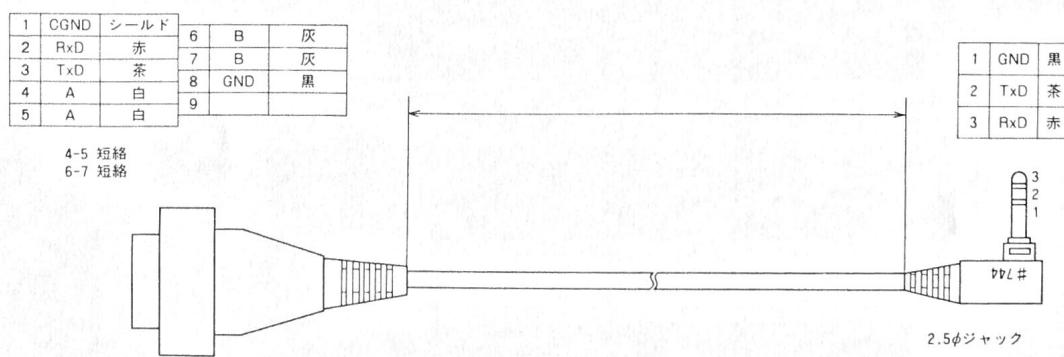
付録

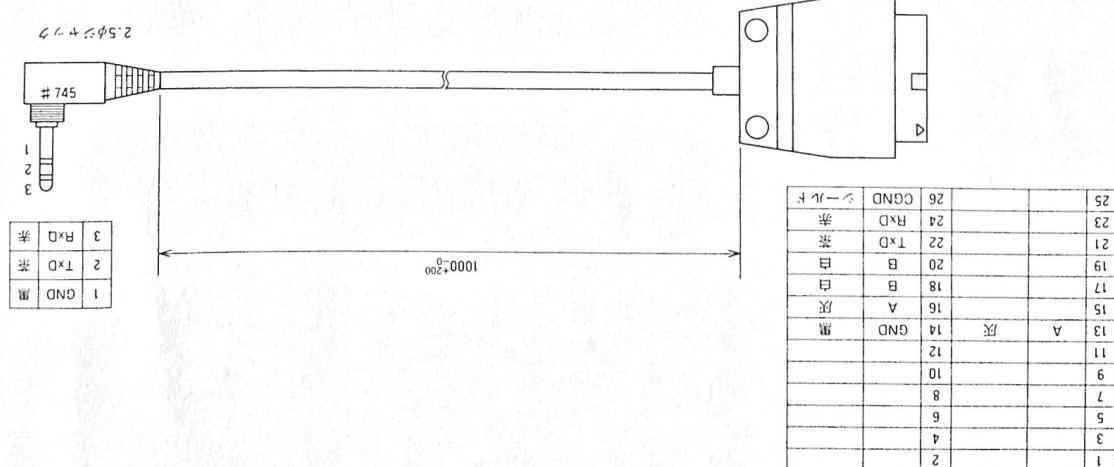
RC-20とパソコンの接続ケーブル

#743(HC-20用)



#744(MZ関係用)





#745(FM圖解用)

リスト 脳コンピュータ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 嶽  
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

■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

BF00	05	21	38	BE	CD	2D	BA	C5	01	BD	BE	CD	3F	BA	C1	DD	:75	:1874-3E, エルヘコロゴ
Sum	BF	AB	D3	FA	3D	1A	EE	B3	B4	20	83	4E	01	88	4D	5E	:08	
Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	:Sum	
BA00	7E	0A	21	79	BE	CD	2D	BA	78	C5	01	E7	BE	CD	3F	BA	:3D	
BA10	07	07	07	C6	AB	F5	CD	5D	7C	BD	99	DE	E1	F1	06	01	:F4	. ルバード・カーブ
BA20	CD	CE	BC	C1	DD	23	04	78	FE	05	38	BC	C9	45	FB	27	:A5	. ホシヂョ・X、8シノヒ:
BA30	CB	27	83	5F	16	60	19	EB	68	26	00	29	29	C9	09	:C9	. ヒー...+...#18.)...)	
BA40	EB	01	05	00	ED	00	BA	C9	27	BD	70	BF	79	21	93	:57	. +...+...#23+...#24)	
BA50	BA	00	CD	41	CD	47	42	CD	A3	43	7E	CD	3F	BA	26	:40	. ハグ・AGB...+...C 1.8 &	
BA60	38	09	FE	32	28	EL	FE	34	28	ED	C9	D6	21	5F	16	:00	. EE 8. 2.F4. #031) - .	
BA70	21	93	BE	19	CD	46	26	DE	61	EB	29	29	01	BF	F5	:P5	. ハグ・6.4) - .	
BA80	BF	11	8D	BE	CD	3F	BA	3E	01	1E	4C	0D	1C	41	CD	:51	. ユニコーン...+...AN	
BA90	BC	18	C4	6F	6F	20	11	98	BE	06	04	C5	62	7B	. ド...+...#1000...+...R...+...D			
BAB0	6B	1A	13	4E	B9	38	03	77	79	12	00	F6	13	1C	10	:EE	. C4. K.#NP28.WY...+...H.	
BAC0	C9	21	9D	BE	7E	66	4	23	BE	C0	10	FB	21	B7	BE	:DA	. ハグ・#17...+...#18...+...#19...+...#20...+...#21...+...#22...+...#23...+...#24...+...#25...+...#26...+...#27...+...#28...+...#29...+...#30...+...#31...+...#32...+...#33...+...#34...+...#35...+...#36...+...#37...+...#38...+...#39...+...#40...+...#41...+...#42...+...#43...+...#44...+...#45...+...#46...+...#47...+...#48...+...#49...+...#50...+...#51...+...#52...+...#53...+...#54...+...#55...+...#56...+...#57...+...#58...+...#59...+...#60...+...#61...+...#62...+...#63...+...#64...+...#65...+...#66...+...#67...+...#68...+...#69...+...#70...+...#71...+...#72...+...#73...+...#74...+...#75...+...#76...+...#77...+...#78...+...#79...+...#80...+...#81...+...#82...+...#83...+...#84...+...#85...+...#86...+...#87...+...#88...+...#89...+...#90...+...#91...+...#92...+...#93...+...#94...+...#95...+...#96...+...#97...+...#98...+...#99...+...#100...+...#101...+...#102...+...#103...+...#104...+...#105...+...#106...+...#107...+...#108...+...#109...+...#110...+...#111...+...#112...+...#113...+...#114...+...#115...+...#116...+...#117...+...#118...+...#119...+...#120...+...#121...+...#122...+...#123...+...#124...+...#125...+...#126...+...#127...+...#128...+...#129...+...#130...+...#131...+...#132...+...#133...+...#134...+...#135...+...#136...+...#137...+...#138...+...#139...+...#140...+...#141...+...#142...+...#143...+...#144...+...#145...+...#146...+...#147...+...#148...+...#149...+...#150...+...#151...+...#152...+...#153...+...#154...+...#155...+...#156...+...#157...+...#158...+...#159...+...#160...+...#161...+...#162...+...#163...+...#164...+...#165...+...#166...+...#167...+...#168...+...#169...+...#170...+...#171...+...#172...+...#173...+...#174...+...#175...+...#176...+...#177...+...#178...+...#179...+...#180...+...#181...+...#182...+...#183...+...#184...+...#185...+...#186...+...#187...+...#188...+...#189...+...#190...+...#191...+...#192...+...#193...+...#194...+...#195...+...#196...+...#197...+...#198...+...#199...+...#200...+...#201...+...#202...+...#203...+...#204...+...#205...+...#206...+...#207...+...#208...+...#209...+...#210...+...#211...+...#212...+...#213...+...#214...+...#215...+...#216...+...#217...+...#218...+...#219...+...#220...+...#221...+...#222...+...#223...+...#224...+...#225...+...#226...+...#227...+...#228...+...#229...+...#230...+...#231...+...#232...+...#233...+...#234...+...#235...+...#236...+...#237...+...#238...+...#239...+...#240...+...#241...+...#242...+...#243...+...#244...+...#245...+...#246...+...#247...+...#248...+...#249...+...#250...+...#251...+...#252...+...#253...+...#254...+...#255...+...#256...+...#257...+...#258...+...#259...+...#260...+...#261...+...#262...+...#263...+...#264...+...#265...+...#266...+...#267...+...#268...+...#269...+...#270...+...#271...+...#272...+...#273...+...#274...+...#275...+...#276...+...#277...+...#278...+...#279...+...#280...+...#281...+...#282...+...#283...+...#284...+...#285...+...#286...+...#287...+...#288...+...#289...+...#290...+...#291...+...#292...+...#293...+...#294...+...#295...+...#296...+...#297...+...#298...+...#299...+...#200...+...#201...+...#202...+...#203...+...#204...+...#205...+...#206...+...#207...+...#208...+...#209...+...#2010...+...#2011...+...#2012...+...#2013...+...#2014...+...#2015...+...#2016...+...#2017...+...#2018...+...#2019...+...#2020...+...#2021...+...#2022...+...#2023...+...#2024...+...#2025...+...#2026...+...#2027...+...#2028...+...#2029...+...#2030...+...#2031...+...#2032...+...#2033...+...#2034...+...#2035...+...#2036...+...#2037...+...#2038...+...#2039...+...#2040...+...#2041...+...#2042...+...#2043...+...#2044...+...#2045...+...#2046...+...#2047...+...#2048...+...#2049...+...#20410...+...#20411...+...#20412...+...#20413...+...#20414...+...#20415...+...#20416...+...#20417...+...#20418...+...#20419...+...#20420...+...#20421...+...#20422...+...#20423...+...#20424...+...#20425...+...#20426...+...#20427...+...#20428...+...#20429...+...#20430...+...#20431...+...#20432...+...#20433...+...#20434...+...#20435...+...#20436...+...#20437...+...#20438...+...#20439...+...#20440...+...#20441...+...#20442...+...#20443...+...#20444...+...#20445...+...#20446...+...#20447...+...#20448...+...#20449...+...#20450...+...#20451...+...#20452...+...#20453...+...#20454...+...#20455...+...#20456...+...#20457...+...#20458...+...#20459...+...#20460...+...#20461...+...#20462...+...#20463...+...#20464...+...#20465...+...#20466...+...#20467...+...#20468...+...#20469...+...#20470...+...#20471...+...#20472...+...#20473...+...#20474...+...#20475...+...#20476...+...#20477...+...#20478...+...#20479...+...#20480...+...#20481...+...#20482...+...#20483...+...#20484...+...#20485...+...#20486...+...#20487...+...#20488...+...#20489...+...#20490...+...#20491...+...#20492...+...#20493...+...#20494...+...#20495...+...#20496...+...#20497...+...#20498...+...#20499...+...#204100...+...#204101...+...#204102...+...#204103...+...#204104...+...#204105...+...#204106...+...#204107...+...#204108...+...#204109...+...#204110...+...#204111...+...#204112...+...#204113...+...#204114...+...#204115...+...#204116...+...#204117...+...#204118...+...#204119...+...#204120...+...#204121...+...#204122...+...#204123...+...#204124...+...#204125...+...#204126...+...#204127...+...#204128...+...#204129...+...#204130...+...#204131...+...#204132...+...#204133...+...#204134...+...#204135...+...#204136...+...#204137...+...#204138...+...#204139...+...#204140...+...#204141...+...#204142...+...#204143...+...#204144...+...#204145...+...#204146...+...#204147...+...#204148...+...#204149...+...#204150...+...#204151...+...#204152...+...#204153...+...#204154...+...#204155...+...#204156...+...#204157...+...#204158...+...#204159...+...#204160...+...#204161...+...#204162...+...#204163...+...#204164...+...#204165...+...#204166...+...#204167...+...#204168...+...#204169...+...#204170...+...#204171...+...#204172...+...#204173...+...#204174...+...#204175...+...#204176...+...#204177...+...#204178...+...#204179...+...#204180...+...#204181...+...#204182...+...#204183...+...#204184...+...#204185...+...#204186...+...#204187...+...#204188...+...#204189...+...#204190...+...#204191...+...#204192...+...#204193...+...#204194...+...#204195...+...#204196...+...#204197...+...#204198...+...#204199...+...#2041910...+...#2041911...+...#2041912...+...#2041913...+...#2041914...+...#2041915...+...#2041916...+...#2041917...+...#2041918...+...#2041919...+...#2041920...+...#2041921...+...#2041922...+...#2041923...+...#2041924...+...#2041925...+...#2041926...+...#2041927...+...#2041928...+...#2041929...+...#2041930...+...#2041931...+...#2041932...+...#2041933...+...#2041934...+...#2041935...+...#2041936...+...#2041937...+...#2041938...+...#2041939...+...#2041940...+...#2041941...+...#2041942...+...#2041943...+...#2041944...+...#2041945...+...#2041946...+...#2041947...+...#2041948...+...#2041949...+...#2041950...+...#2041951...+...#2041952...+...#2041953...+...#2041954...+...#2041955...+...#2041956...+...#2041957...+...#2041958...+...#2041959...+...#2041960...+...#2041961...+...#2041962...+...#2041963...+...#2041964...+...#2041965...+...#2041966...+...#2041967...+...#2041968...+...#2041969...+...#2041970...+...#2041971...+...#2041972...+...#2041973...+...#2041974...+...#2041975...+...#2041976...+...#2041977...+...#2041978...+...#2041979...+...#2041980...+...#2041981...+...#2041982...+...#2041983...+...#2041984...+...#2041985...+...#2041986...+...#2041987...+...#2041988...+...#2041989...+...#2041990...+...#2041991...+...#2041992...+...#2041993...+...#2041994...+...#2041995...+...#2041996...+...#2041997...+...#2041998...+...#2041999...+...#20419100...+...#20419101...+...#20419102...+...#20419103...+...#20419104...+...#20419105...+...#20419106...+...#20419107...+...#20419108...+...#20419109...+...#20419110...+...#20419111...+...#20419112...+...#20419113...+...#20419114...+...#20419115...+...#20419116...+...#20419117...+...#20419118...+...#20419119...+...#204191100...+...#204191101...+...#204191102...+...#204191103...+...#204191104...+...#204191105...+...#204191106...+...#204191107...+...#204191108...+...#204191109...+...#204191110...+...#204191111...+...#204191112...+...#204191113...+...#204191114...+...#204191115...+...#204191116...+...#204191117...+...#204191118...+...#204191119...+...#2041911100...+...#2041911101...+...#2041911102...+...#2041911103...+...#2041911104...+...#2041911105...+...#2041911106...+...#2041911107...+...#2041911108...+...#2041911109...+...#2041911110...+...#2041911111...+...#2041911112...+...#2041911113...+...#2041911114...+...#2041911115...+...#2041911116...+...#2041911117...+...#2041911118...+...#2041911119...+...#20419111100...+...#20419111101...+...#20419111102...+...#20419111103...+...#20419111104...+...#20419111105...+...#20419111106...+...#20419111107...+...#20419111108...+...#20419111109...+...#20419111110...+...#20419111111...+...#20419111112...+...#20419111113...+...#20419111114...+...#20419111115...+...#20419111116...+...#20419111117...+...#20419111118...+...#20419111119...+...#204191111100...+...#204191111101...+...#204191111102...+...#204191111103...+...#204191111104...+...#204191111105...+...#204191111106...+...#204191111107...+...#204191111108...+...#204191111109...+...#204191111110...+...#204191111111...+...#204191111112...+...#204191111113...+...#204191111114...+...#204191111115...+...#204191111116...+...#204191111117...+...#204191111118...+...#204191111119...+...#2041911111100...+...#2041911111101...+...#2041911111102...+...#2041911111103...+...#2041911111104...+...#2041911111105...+...#2041911111106...+...#2041911111107...+...#2041911111108...+...#2041911111109...+...#2041911111110...+...#2041911111111...+...#2041911111112...+...#2041911111113...+...#2041911111114...+...#2041911111115...+...#2041911111116...+...#2041911111117...+...#2041911111118...+...#2041911111119...+...#20419111111100...+...#20419111111101...+...#20419111111102...+...#20419111111103...+...#20419111111104...+...#20419111111105...+...#20419111111106...+...#20419111111107...+...#20419111111108...+...#20419111111109...+...#20419111111110...+...#20419111111111...+...#20419111111112...+...#20419111111113...+...#20419111111114...+...#20419111111115...+...#20419111111116...+...#20419111111117...+...#20419111111118...+...#20419111111119...+...#204191111111100...+...#204191111111101...+...#204191111111102...+...#204191111111103...+...#204191111111104...+...#204191111111105...+...#204191111111106...+...#204191111111107...+...#204191111111108...+...#204191111111109...+...#204191111111110...+...#204191111111111...+...#204191111111112...+...#204191111111113...+...#204191111111114...+...#204191111111115...+...#204191111111116...+...#204191111111117...+...#204191111111118...+...#204191111111119...+...#2041911111111100...+...#2041911111111101...+...#2041911111111102...+...#2041911111111103...+...#2041911111111104...+...#2041911111111105...+...#2041911111111106...+...#2041911111111107...+...#2041911111111108...+...#2041911111111109...+...#2041911111111110...+...#2041911111111111...+...#2041911111111112...+...#2041911111111113...+...#2041911111111114...+...#2041911111111115...+...#2041911111111116...+...#2041911111111117...+...#2041911111111118...+...#2041911111111119...+...#20419111111111100...+...#20419111111111101...+...#20419111111111102...+...#20419111111111103...+...#20419111111111104...+...#20419111111111105...+...#20419111111111106...+...#20419111111111107...+...#20419111111111108...+...#20419111111111109...+...#20419111111111110...+...#20419111111111111...+...#20419111111111112...+...#20419111111111113...+...#20419111111111114...+...#20419111111111115...+...#20419111111111116...+...#20419111111111117...+...#20419111111111118...+...#20419111111111119...+...#204191111111111100...+...#204191111111111101...+...#204191111111111102...+...#204191111111111103...+...#204191111111111104...+...#204191111111111105...+...#204191111111111106...+...#204191111111111107...+...#204191111111111108...+...#204191111111111109...+...#204191111111111110...+...#204191111111111111...+...#204191111111111112...+...#204191111111111113...+...#204191111111111114...+...#204191111111111115...+...#204191111111111116...+...#204191111111111117...+...#204191111111111118...+...#204191111111111119...+...#2041911111111111100...+...#2041911111111111101...+...#2041911111111111102...+...#2041911111111111103...+...#2041911111111111104...+...#2041911111111111105...+...#2041911111111111106...+...#2041911111111111107...+...#2041911111111111108...+...#2041911111111111109...+...#2041911111111111110...+...#2041911111111111111...+...#2041911111111111112...+...#2041911111111111113...+...#2041911111111111114...+...#2041911111111111115...+...#2041911111111111116...+...#2041911111111111117...+...#2041911111111111118...+...#2041911111111111119...+...#20419111111111111100...+...#20419111111111111101...+...#20419111111111111102...+...#20419111111111111103...+...#20419111111111111104...+...#20419111111111111105...+...#20419111111111111106...+...#20419111111111111107...+...#20419111111111111108...+...#20419111111111111109...+...#20419111111111111110...+...#20419111111111111111...+...#20419111111111111112...+...#20419111111111111113...+...#20419111111111111114...+...#20419111111111111115...+...#20419111111111111116...+...#20419111111111111117...+...#20419111111111111118...+...#20419111111111111119...+...#204191111111111111100...+...#204191111111111111101...+...#204191111111111111102...+...#204191111111111111103...+...#204191111111111111104...+...#204191111111111111105...+...#204191111111111111106...+...#204191111111111111107...+...#204191111111111111108...+...#204191111111111111109...+...#20419111111111	