ND80ZⅢ　ND80Zモニタプログラム操作説明書

(有)中日電工

目次

〒463-0067　名古屋市守山区守山２－８－１４
パレス守山３０５
有限会社中日電工
℡052-791-6254　Fax052-791-1391
E-mail　thisida@alles.or.jp
Homepage　http://www.alles.or.jp/~thisida/

2010. 9. 25　Rev.　1. 0

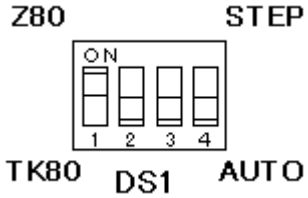
1章　基本操作

1．　はじめに

この説明書はND80Zモニタプログラムの操作説明書です。
ND80Zモニタプログラムの基本的な機能はTK80モニタプログラムと変わりません。
そこで、ND80ZモニタプログラムとTK80モニタプログラムとで共通している事柄については、TK80モニタプログラム操作説明書でまとめて説明することにしました。ですから、まず先にTK80モニタプログラム操作説明書を読んで、基本的な操作や基礎知識について理解してから、本書を読むようにしてください。

モニタプログラムは、ディップスイッチDS1で選択するようになっています。
ND80ZモニタプログラムはディップスイッチDS1のNo.1がON、No.2がOFFのときに選択されます。
ディップスイッチの設定は、電源投入時かまたはリセットスイッチを押したときに検出されます。
それ以外のときに設定を変更しても、状態は変化しませんから、電源が入っているときに設定を変更した場合、その設定を有効にするためには、リセットスイッチ（MONキー）を押す必要があります。

電源を入れる前に、図のようにディップスイッチDS1のNo.1がON、No.2がOFFになっていることを確認してください。
もし図のようになっていないビットがあれば、小型のマイナスドライバなどで、図と同じになるようにしてください。
電源ON後に設定を変更した場合には、変更後にリセットスイッチ（MONキー）を押すと、設定が有効になります。
No.3、No.4はRS232C通信のボーレート設定用です。通常はOFFにしておいてください。



2．　キーボード

ND80Zモニタプログラムに色々な指示を与えたり、メモリの中身を読んだり書いたりするときには、それらの作業は全てキーボードからの入力によって行います。
ND80ZⅢのキーは、５×５配列のキー25個です。この25個のキーは、その働きによって、次の3つのグループに分けられます。

2．1　データキー

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

メモリアドレスを指定して、データを書き込んだり、プログラムを入力するときの、16進数のキーです。
0～Fのほかにも、色々な記号がついていますが、この記号は［＊（I／O）］キーや［REG］キーを使うときのものです。
それについては2章以降で説明します。ここでは16進数の入力キーとして、覚えて下さい。

2．2　ファンクションキー

［CONT］［RUN］［＊（I／O）］［REG］［ADRSSET］［READINC］［READDEC］［WRITEINC］

［＊（I／O）］キーや［REG］キー以外のキーの操作方法や、キーの機能はTK80モニタプログラムと同じです。
基本的なキー操作については、「TK80モニタプログラム操作説明書」を参照してください。

2．3　リセット（モニタエントリ）キー

［MON］

実行中のプログラムを中止してモニタプログラムに戻ります。
もう少し詳しい説明が「TK80モニタプログラム操作説明書」にありますから参照してください。

2章　プログラムデバッグの仕方

1．ステップ動作

ND80Zモニタプログラムのステップ動作は、ワークアドレスが異なっていることおよび退避されるレジスタがZ80の全レジスタであることを除いてはTK80モニタプログラムと同じです。
ワークアドレスや対象になるレジスタについては、2．ブレイク動作で説明をします。
そのほかの基本的な操作については、ここでは省略しますから、具体的な操作については「TK80モニタプログラム操作説明書」を参照してください。

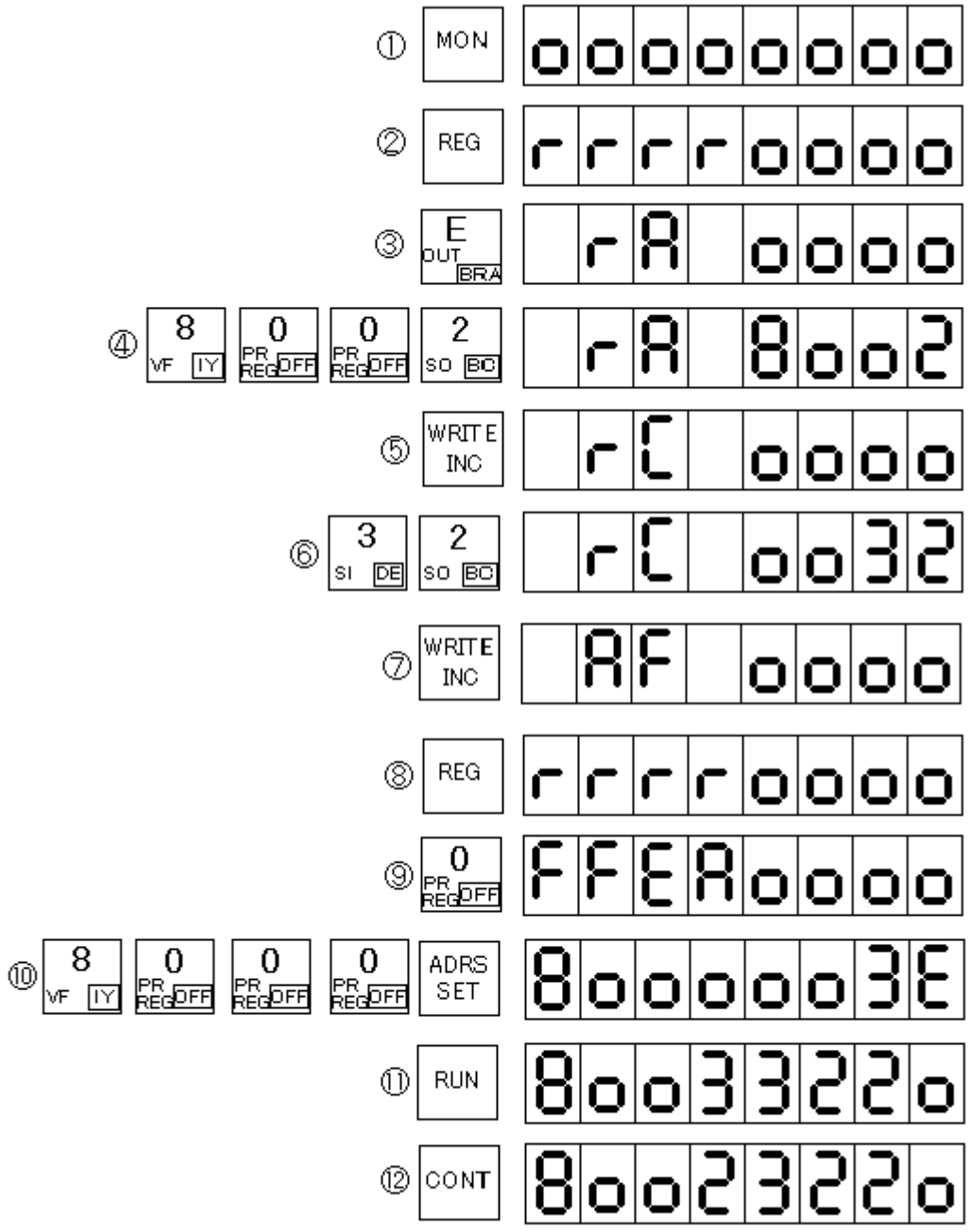
2．　ブレイク動作

ND80Zモニタプログラムのブレイク動作についても、基本的な機能はTK80モニタプログラムと同じです。
ただワークアドレスが異なること、退避されるレジスタがZ80の全レジスタであることが異なっているほか、ブレイクカウンタやブレイクアドレスの設定、退避されたレジスタの値の確認などがTK80モニタプログラムに比べて簡単なキー操作で行うことができます。
ここではTK80モニタプログラムの操作例と同じように、「TK80モニタプログラム操作説明書」の2章で作ったプログラムをブレイクさせてみます。
8002番地を50回実行したあとステップ動作に移る(ブレイクする)ようにセットします。
図2－1を参照しながら、以下の説明を読んで下さい。

①まずリセットします。(ブレイクの時は必ずしもリセットから始めなくてもよいのですが、この方が確実です)
②ブレイクアドレスをセットします。
まず[REG]キーを押します。
このキーについては後で詳しく説明します。いまは図2－1 の通りに操作して下さい。
[REG]キーを押すと、アドレス表示部にはrrrrが表示されます。
③データキーの[E](白抜き文字の[BR A])を押します。するとアドレス表示部に rA と表示され、データ表示部にはブレイクアドレスが表示されます(リセツト後は0000になっています)。
④ブレイクしたいアドレスを入れます。今回は8002をセットします。
⑤[WRITE INC]キーを押します。
これでブレイクアドレスがセットされ、アドレス表示部には rC (ブレイクカウンタ)が表示されます。
このときデータ表示部にはブレイクカウンタの値が表示されます(リセット後は0000になっています) 。
⑥ブレイクカウンタに値をセットします。今回は50回にします。10進の50は16進では32になります。そこで[3][2]と入力します。
ブレイクカウンタは16進2桁です。LEDのデータ表示部は4桁ありますが、上位2桁に何が表示されていても、その上位2桁の値は無視されます。
ブレイクカウンタにセットできる値は01～FF(十進数の0～255)です。
⑦[WRITE INC]キーを押します。
これでブレイクカウンタのセットができました。
⑧もう一度[REG]キーを押します。
⑨最後にデータキーの[0](白抜き文字の[OFF])を押します。
以上で準備完了です。このときアドレスレジスタにはFFEAが表示されますが、気にしないで下さい。
⑩プログラムの開始アドレスをセットして下さい。[8][0][0][0][ADRSSET]と入力します。
⑪ディップスイッチDS1の№4がON(STEP側)になっていることを確認して下さい。
OFF(AUTO側)になっていると、ブレイクできません。
[RUN]キーを押すと瞬間にプログラムが50回実行されて、ブレイクします。50回実行した証拠に、データ表示部の上2桁にはAレジスタの値、32(十進数の50)が表示されています。
これ以後はステップ操作と同じです。[CONT]キーを押すと1ステップずつ進みます。
ブレイクカウンタはブレイク時点で0になります。ブレイクアドレスは[MON](RESET)キーを押さない限りクリアされないでそのまま残ります。

〔注意1〕ステップ動作は、モニタROM内のプログラムには働きません。
〔注意2〕ステップ動作は割り込み(RST7)を利用しています。したがってもしユーザープログラムの中で、割り込み禁止命令(DI)を使うと、それ以後は割り込みが禁止されるため、ステップ動作ができなくなります。
〔注意3〕ブレイクカウンタが0になっている時は、ブレイク動作ではなくてステップ動作になります。
〔注意4〕ブレイクアドレスは各命令の1バイト目でなければいけません。今回の例では8000、8002、8003は指定できますが、8001、8004、8005を指定してはいけません。

〔注意5〕ブレイクアドレスを設定した場合、ブレイクするまでの間のプログラム実行時間は、通常処理の場合の数十倍かかります。これはブレイクアドレス以外のプログラム部分でも1ステップ実行する毎に、ブレイク処理プログラムが実行されているためです。



（図2－1）

3.　レジスタモード

Z80は他の8ビットCPUに比べて、非常に豊富なレジスタをもっています。

レジスタモードの機能を使うことにより、ステップ動作やブレイク時に、その時点でのレジスタの値を確認したり、変更したりすることが簡単にできます。

またレジスタに特定の値をセットしてから、プログラムをスタートさせることもできます。

具体的な操作方法について説明する前に、まずZ80Aのレジスタについて簡単に説明しておきます。

3．1　Z80のレジスタ

Z80は内部に下記のレジスタを持っていて、これらのレジスタはプログラムの中で色々な処理に利用されます。

← 8ビット→ ← 8ビット→

A	F
B	C
D	E
H	L

← 8ビット→ ← 8ビット→

A’	F’
B’	C’
D’	E’
H’	L’

← 16ビット →

IX	
IY	
SP	
PC	
I	R

（裏レジスタ）

A～Lは全く同じレジスタがもう1組あります。
通常裏レジスタと呼んでいます。EXX’ やEX AF，AF’ 命令で表裏を切り換えることができますが厳密な意味での表裏の区別はありません（裏も表に呼び出せばその時点から表レジスタになってしまいます）。

［A］
一般にアキュムレータ（加算器）と呼ばれているように、演算命令はこのレジスタを中心に行われます。
［F］
フラグレジスタ。命令の実行により現れる色々な状態を1ビットずつに記録して保持します。各ビットの意味は3．2で説明します。
［B］［C］
共に 8 ビットのレジスタとして独立して使うこともできますが、つないで 16 ビットのレジスタ［BC］として使うこともできます。その場合には［B］が上位8ビット、［C］が下位8ビットになります。［B］または［BC］をカウンタとして使っている命令がいくつかあります。
［D］［E］
B、Cと同じですがカウンタとして使う命令はありません。
［H］［L］
B、Cと同じですがカウンタとして使う命令はありません。16ビットレジスタ［HL］はメモリアドレスを入れて、メモリを間接的に示すときにも使われます（間接アドレッシングモード）。また［HL］は16ビットの加減算命令で加算器（アキュムレータ）としても使われます。
［IX］［IY］
メモリアドレスを間接的に示すインデックスレジスタとして使いますが、16ビットのワークレジスタとして使うこともできます。［HL］と同様に16ビットの加減算命令で加算器（アキュムレータ）としても使われます。
［SP］
スタックポインタ。現在のスタックのトップ・アドレスを示しています。スタックについては、3．3で説明します。
［PC］
プログラムカウンタ。現在実行中のアドレス(正しくは、次のアドレス)を管理しています。
［I］
インタラプトベクタレジスタ。モード2の割り込みで使用されます。
［R］
リフレッシュレジスタ。ダイナミックRAMのリフレッシュアドレス出力用カウンタ。普通のプログラムでは使いませんが、乱数発生カウンタとして利用する、という使われ方をされることもあります。

3．2 Z80のフラグ

フラグは8ビットのフラグレジスタに、下のように割りつけられています。

S	Z		H		P／V	N	C
7	6	5	4	3	2	1	0

（ビット位置）

各記号の意味は下の通りです。（ビット3とビット5は使用されません）
なお、フラグがセットされたときは、そのビットが1になり、リセットされたときは0になります。

C
キャリーフラグ。計算結果がオーバーフローしたときなどにセットされます。ローテイト命令でもセット、リセットされます。
N
加減算フラグ。加算命令のときリセット、減算命令のときセットされます。
P／V

パリティ・オーバーフロー・フラグ。論理演算のときはパリティ・フラグとして用いられ、演算の結果1のビットが偶数個あるときにセットされ、奇数個のときはリセットされます。算術演算のときはオーバーフロー・フラグとして用いられ結果がオーバーフローしたときセットされます。

H

ハーフ・キャリーフラグ。算術演算でビット3からビット4へのキャリーや、ビット4からビット3へのボローがあったときセットされます。Nフラグとともに、BCD演算後のDAA命令で利用されます。

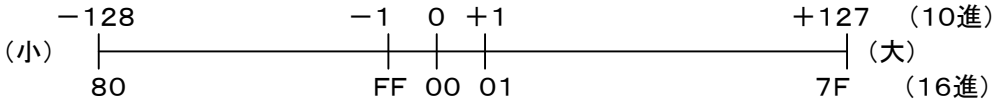
Z

ゼロ・フラグ。結果がゼロのときセットされます。

S

サイン・フラグ。結果が負のときセット、正またはゼロのときリセットされます。8ビットの数00～FFは符号なしでは10進の0～255として扱われますが、符号付の数として扱ったときには、－128～＋127の数になり、これは16進では80～7Fになります（ビット7が0のときはその数は正で、ビット7が1のときは負になります）。

●符号付8ビットの数の大小



3．3 スタック

大きなプログラムになると、レジスタもたくさん必要で、とても3．1で説明した数では足りません。そこでレジスタの値をひとまずメモリのワークエリアにしまっておいて、そのレジスタを次の用途に使う、ということが簡単にできると便利になります。

ところがレジスタの値をしまうときに、一々異なるメモリアドレスに割りつけていくのでは大変です。

そんなときにこのスタックを使えば、メモリアドレスを指定しなくても簡単な操作でレジスタの値を保存することができます。

スタックとは積み重ねるという意味です。

ちょうど本などを積み重ねるように、メモリの中にレジスタの値を順番にしまうことができます（PUSH命令を使います）。

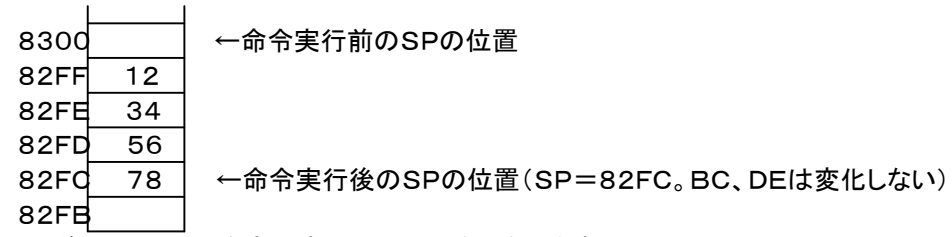
取り出すときは、入れたときと逆の順番で取り出します（POP命令を使います）。

そしてそのスタックの現在の位置を管理しているのがSP（スタックポインタ）です。

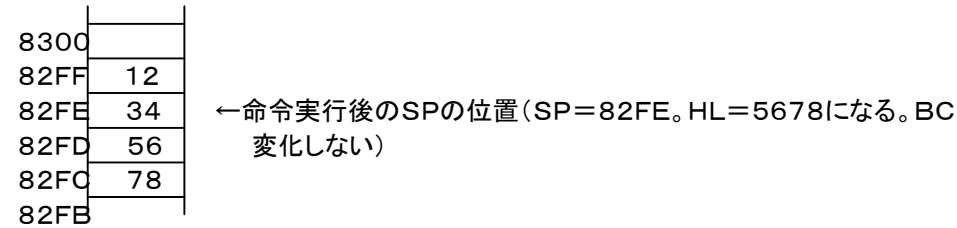
（例）SP＝8300、BC＝1234、DE＝5678のとき、

```
PUSH  BC
PUSH  DE
```

を実行すると、メモリ内容は下のようになります。



この後でPOP　HLを実行すると、下のようになります。



こうなってしまった後で、POP　DEを実行してもDEにはもとの値は戻りません（1234が入る）。

ここではスタックの特殊な使い方として説明しました。

一般的な使い方としては、一時的に退避しておきたいレジスタを、PUSH命令でスタックに保存しておいて、あとでそれをもとのレジスタに戻すときは、PUSHとは「逆の順序」でPOPを実行します。

PUSH　BC、PUSH　DE　の順で実行したら、それをもとに戻すときはPOP　DE、POP　BCのようにPUSHのときの逆の順序にします。

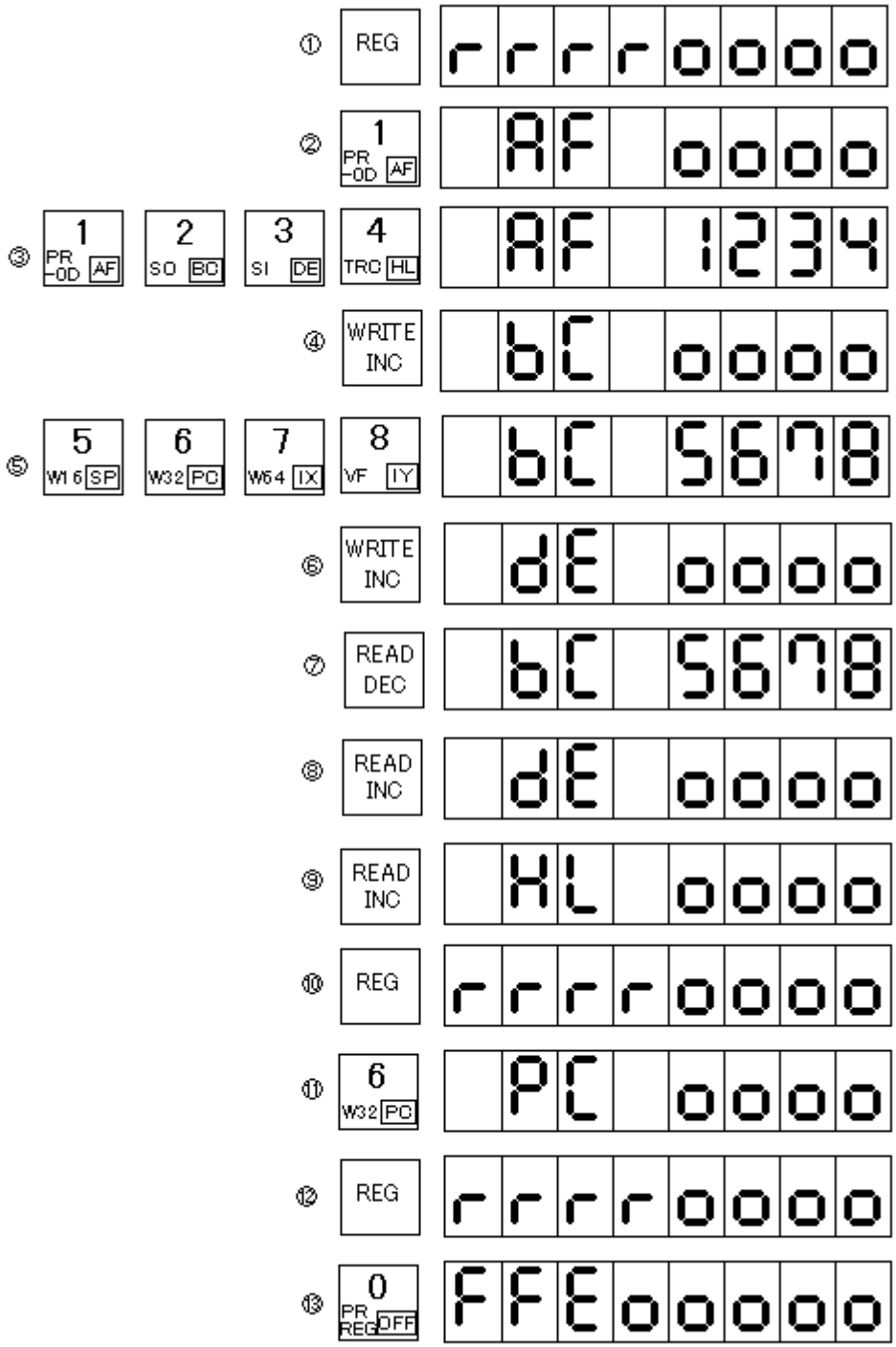
PUSH、POPは常に順番を覚えておいて、間違わないように使う必要があります。

〔注意1〕スタックの操作はPUSH、POPだけではなくCALL、RET命令や割り込み処理でも使用されます（アドレスがスタックに入れられる）。

〔注意2〕スタックはメモリ上のどこにでも設定することができます（LD　SP命令を使う）。
　しかし指定場所によってはプログラムやデータの入っている領域と重なってしまい、その結果プログラムやデータが壊されてしまうことがあるので、充分注意が必要です。
　マシン語プログラムでは、普通はその先頭部分でスタックポインタのセットが必要ですが、ND80Zモニタプログラムではリセット後はSP＝F800にセットされるのでユーザーがあらためてスタックポインタをセットする必要はありません。

3．4　レジスタモードの操作方法

　レジスタモードは［REG］キーとデータキーを組み合わせて使います（ 以下の説明は次頁の図2－2を参照しながら読んで下さい）。
①［REG］キーを押すとアドレス表示部にrrrrが表示され、レジスタモードであることを示します。
rrrrが表示されているときには、データキー以外のキーは押してはいけません。
　処理を中止するためにリセット（［MON］）キーを押すことはできます。
②データキーはZ80のレジスタに対応しています。
　レジスタモードで、［REG］キーを押して、rrrrが表示されているときは、データキーの白抜き文字が有効になっています。
　データキーの［1］（白抜き文字の［AF］）を押して下さい。アドレス表示部にAFと表示され、データ表示部にはステップ動作やブレイク動作で中断した時点のAレジスタとFレジスタの内容が表示されます。（リセット後はSP＝F800以外は全レジスタ＝0000になっています）
　この状態のときは、データ入力キーとして［0］～［F］キーと、［READ　INC］、［READ　DEC］、［WRITE　INC］が使用できます。
③レジスタ内容を変更したい場合は、データキーを使って、値をセットします。例として［1］［2］［3］［4］とキー入力してみます。
④最後に［WRITE　INC］を押します。これでAレジスタに12が入り、Fレジスタに34が入れられました。
　正確に言うと、この時点ではまだメモリ上のレジスタセーブエリアが書き換えられただけで、CPUレジスタに書き込まれた訳ではありません。［CONT］または［RUN］キーを押したときに、レジスタセーブエリアからCPUレジスタに値がセットされます。
　アドレスレジスタには次のレジスタペアBCが表示されます。
⑤これも変更したければ、同じように操作します。［5］［6］［7］［8］と入れてみます。
⑥［WRITE　INC］を押すと、さらに次のレジスタペアDEが表示されます。
⑦今書き込んだ内容を確認することもできます。
　［READ　DEC］キーを押して下さい。
　表示がBCに戻って、データ表示部には今書き込んだ内容が表示されます。
⑧逆に書き込みをしないで、先に進んでレジスタ内容を見たいときは、［READ　INC］キーを押します。
⑨続けて押せば、一つずつ表示が進んでいきます（［READ　DEC］も同様です）。
⑩順番に確認するのではなくて、必要なレジスタだけ見たい場合には、もう一度［REG］キーを押します。
⑪続けて確認したいレジスタのキーを押します。たとえば［6］（［PC］）を押すと、PCが表示されます。
⑫レジスタ設定モードを終了するときは、先に［REG］キーを押し、
⑬続いて、データキーの［0］（［OFF］）を押します。これで普通のモニタの状態に戻ります。
アドレス表示部には、OFFになった位置によって異なるアドレスが表示されますが気にしないでください。
以上の作業がブレイク後またはトレース後に行われたときは、このあとで［CONT］キーを押すことによって、上の作業で最終的に書き換えられたデータを各レジスタにセットしたあと、ユーザープログラムに復帰します。
　またブレイク動作やトレース動作以外の普通の処理でも、まずこのレジスタモードで必要な値をセットしたあと、RUNすることにより、CPUレジスタに特定の初期値を持たせてユーザープログラムを開始させることができます。
　以上のようにこの機能によって、実行途中のCPUレジスタの値を簡単な操作で参照できるばかりではなく、必要ならば途中で各レジスタの値を簡単に変更することもできるため、非常にきめ細かなデバッグ作業が行えます。



（図2－2）

〔注意1〕rrrr表示のときは、データキー以外のキーのうち、[CONT][RUN][＊(I／O)][REG][ADRS　SET]の各キーは押しても無視されますが、[READ　INC][READ　DEC][WRITE　INC]の各キーを押すと、異常表示やレジスタセーブエリアの値が書き換えられたりします。

〔注意2〕プログラムカウンタ(PC)やスタックポインタ(SP)の値を不用意に変更すると、以後のユーザープログラムがまともに実行されなくなることがあります。

〔注意3〕ブレイクアドレス(BR　A)やブレイクカウンタ(BR　C)はCPUレジスタではありませんが、機能から考えてこのレジスタモードに入れてあります。

〔注意4〕USBコネクタにUSBケーブルを接続して、Windowsパソコンと接続することにより、ブレイク時やトレース時に全レジスタの内容を一度にWindowsパソコンのディスプレイに表示させることができます(USB接続については5章で説明します)。

3章　I／Oモード（IN／OUT）

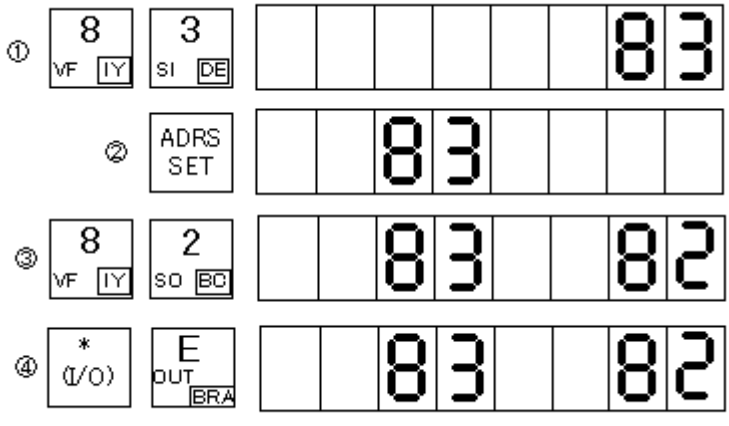
1．　IN、OUTキーの使い方

　2章では[REG]キーを使って、レジスタセーブエリアの参照や値の変更を行う操作について説明をしました。
　3章では、もうひとつの特殊機能キーの[＊（I／O）]キーの基本的な操作について説明をします。
　[＊（I／O]キーはもともとは、メモリアクセスと同じようにI／Oアクセスも簡単なキー操作で行えるように、という目的で、用意したものなのですが、ND80Zモニタの多様な機能を簡単なキー操作で行うときにも、[＊（I／O）]キーが利用されます。そのために、キーシールには[I／O]だけではなくて、その他のいろいろな機能を意味する記号として[＊]が付け加えられています。
　その他のいろいろな機能については4章以降で説明を行います。
　3章では、[＊（I／O）]キーの基本的な使い方として、I／O回路に対する入出力操作（IN／OUT）について説明をします。

　Z80はメモリについては0000～FFFFの64KBのアドレス空間をアクセスすることができますが、これとは全く独立して82C55などのI／Oデバイスについても00～FFの256バイトのアドレス空間をアクセスできます。
　メモリに対しては、キーボードからアドレスを入力するだけで、簡単にREAD、WTRITEすることができました。
　ND80Zモニタプログラムではメモリに対しても、同様の簡単な操作でIN、OUTができます。
　次にその操作例を示します。
　ND80ZⅢに実装されている82C55に対してIN、OUTの操作をしてみます。

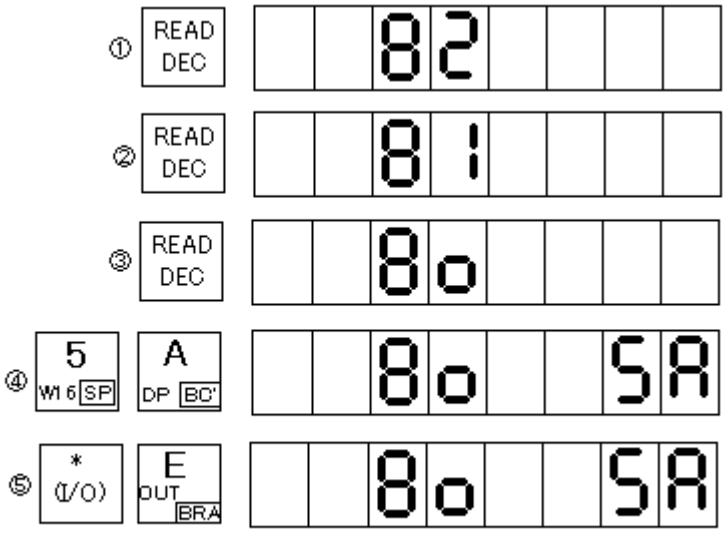
　82C55は、最初にAポート～Cポートの入出力の向き決めるためのコントロールワードをコントロールアドレスに与える必要があります。
　82C55については、「TK80モニタプログラム操作説明書」の　5章　I／O制御　を参照してください。
　ND80ZⅢに実装されている82C55のI／Oアドレスは80～83です。
　80がAポート、81がBポート、82がCポート、83がコントロールワードアドレスです。
　例としてAポートとCポートを出力に、Bポートを入力に設定してみます。
　コントロールワードは82（10000010）です。

①まずI／Oアドレスをセットします。コントロールワートアドレスは83です。
　[8][3]と入力します。メモリと違って下2桁だけが有効です（上2桁は何になっていても構いません）。
　LED表示の空白部分には何が表示されていても構いません。誤解を避けるために空白にしました。
②[ADRS　SET]キーを押すとそのアドレスの「メモリ」の値が、データ表示部に表示されます（このときはまだ「I／Oの値ではありません」）。このとき何が表示されても、気にしないで下さい。
③コントロールワード82を入力します（[WRITE　INC]キーを押してしまわないように注意してください）。
④[＊（I／O）]キーに続いて、データキーのE（[OUT]）を押します。
　このとき表示は変化しませんが指定したI／Oアドレスに対してデータの出力が行われます。
　82C55のコントロールワードアドレスにコントロールワードの82が与えられましたから、82C55のAポートとCポートが出力に、Bポートが入力に設定されました。
　82C55の入出力コネクタCN1の各端子をテスターで測定してみてください（CN1の端子接続図は「ND80ZⅢ取扱説明書」にあります）。
　82C55の、出力に設定されたポートは、設定直後は全ビットが0になります。
　入力に設定されたポートはハイインピーダンスになりますが、ND80ZⅢに実装してある82C55は、各ポートの全ビットが抵抗（10KΩ）で＋5Vにプルアップしてありますから、入力に設定されたポートのビットは＋5Vになります。



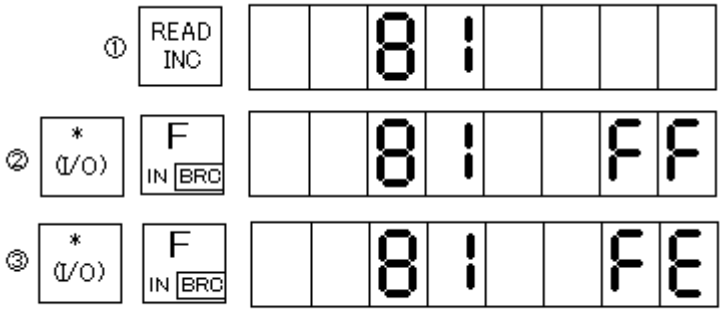
（図3－1）

Aポートにデータを出力してみます。
①Aポートのアドレスは80です。I／Oアドレスの80を、さきほどの例のように[ADRS SET]キーでセットしてもよいのですが、せっかくI／Oアドレスの83が表示されているのですから、[READ DEC]キーを使ってみます。
アドレスが－1されて82が表示されます。
②もう一度[READ DEC]キーを押すと表示が81になります。
③もう一度[READ DEC]キーを押すと表示が80になります。
これでAポートのアドレスが選択されました。
④出力データをセットします。データ5Aをセットしてみます（ここで[WRITE INC]キーを押してしまわないように注意してください）。
⑤[＊(I／O)]キーに続いてデータキーの[E]([OUT])を押すと、表示は変化しませんがこのときAポートから5A(01011010)が出力されています。
Aポートの出力をテスターで確認してみて下さい。1のビットは＋5V、0のビットは0Vになっています。



（図3－2）

Bポートが入力に設定されているので、このBポートからデータを入力してみます。
①Bポートのアドレスは81です。今はI／Oアドレスの80が表示されていますから、[READ INC]を押して、アドレスを81にします。
②[＊(I／O)]キーに続いて、データキーの[F]([IN])を押します。するとデータ表示部の下2桁にFFが表示されます。Bポートは抵抗(10KΩ)で＋5Vにプルアップしてありますから、入力がない場合には、全ビットが1になります。
③附属品の26pinフラットケーブルを利用して、pin№.25(PB0)とpin№.10(GND)とをつないでください。そうしておいて、もう一度[＊(I／O)]キーに続いて、データキーの[F]([IN])を押します。今度はデータ表示部の下2桁がFEに変わります。これは PB0 だけが0になったためです(11111110＝FE)。



（図3－3）

4章　プログラム、データのSAVE、LOAD

1. USB接続

ND80ZⅢのRAMはボタン電池でバックアップをしていますから、RAMに書き込んだプログラムやデータは電源を切っても消えずにそのまま残っています。
しかし複数のプログラムをRAMに常駐させて保存するというのはあまり感心できる方法ではありません。プログラムが暴走したりすれば、プログラムもデータも一瞬で破壊されてしまいます。
ND80Zモニタにはプログラムやデータを SAVE、LOADする機能があります。
USBコネクタにUSBケーブルをつないで、パソコンと接続することによって、RAMにあるプログラムやデータをパソコンに送り、ファイルとして保存することができます。
逆にパソコン上で作成したマシン語のプログラムをUSB接続でND80ZⅢのRAMに送ることもできます。
そのためには、パソコン側でもUSB（HID）READ、WRITEをするプログラムを用意する必要があります。
そのようなプログラムを自作することも可能ですが、ND80ZⅢ組立キットには附属ソフトウェアとして、ND80ZⅢをUSB（HID）で接続して、パソコンのキーボードからND80ZⅢを操作する、リモートリモート＋Z80BASICプログラムがついていますから、それを使った方が簡単です。
しかしここでは、ND80Zの基本的な機能として、リモートプログラムではなくて、ただのUSB（HID）送信、受信プログラムをパソコン側で実行する場合について、説明をします。

ここではパソコン側のハードディスクにHID受信プログラム（HIDRD. EXE）、HID送信プログラム（HIDWR. EXE）がND80ZⅢ附属CDROMからCOPY済みで、ND80ZⅢとWindowsパソコンがUSBケーブルで接続されているものとして説明します。
それらのプログラムをCDROMからハードディスクにCOPYする作業については、「USB接続説明書」を参照してください。

1－1. プログラムのSAVEの仕方

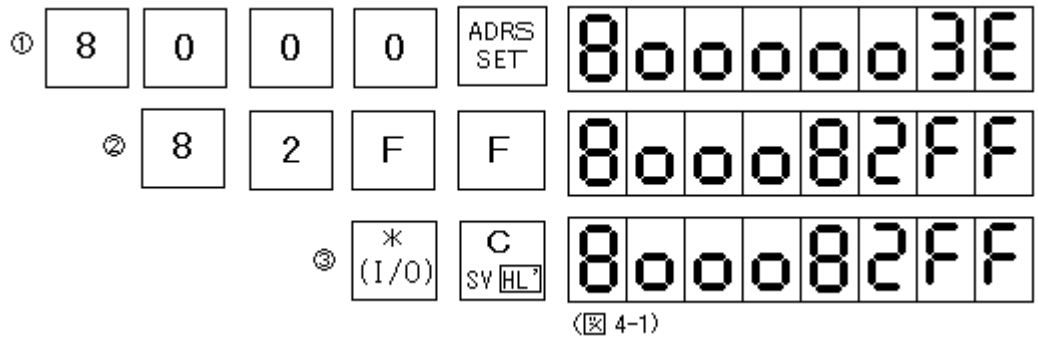
［Windowsパソコン側の操作］
ND80ZⅢの操作をする前に、Windowsパソコン側でHID受信プログラムを実行しておいてください。
コマンドプロンプト画面で、

hidrd　xxxxx. btk［Entr］

と入力します。xxxxx. btkはプログラムを保存したい任意のファイルネームにします。
. 以下の拡張子はbtkでなければならない、ということではありませんが、TK80の場合、データの先頭に4バイトの開始アドレス、終了アドレスが置かれますから、一般的なバイナリファイル（拡張子bin）と区別する意味で、btkを使うことをおすすめします。btkは、binary file　for　TK80の意味です。
画面は以下の表示になって、データ受信待ちになります。

c:￥nd80z3＞hidrd　test. btk
送信を開始してください

［ND80ZⅢの操作］
図 4-1 の通りに操作して下さい。ここでは例として8000～82FFの内容をSAVEすることにします。



（図 4-1）

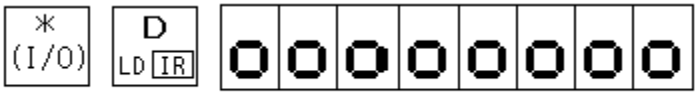
①SAVE開始アドレス（この例では8000）をアドレス表示部にセットします。データ表示部には8000番地の内容が表示されます（ここでは3Eになっています）。
②続いてデータ表示部に、SAVE終了アドレスを入力します（［WRINC］キーは絶対に押さないように!!）。

③最後に[＊(I／O)]に続けて[C SV]を押すと送信が開始されます。
送信中や送信が終了しても7セグメントLEDの表示は変化しませんが、Windowsパソコンのコマンドプロンプト画面には、受信が完了する以下のように表示されます。

```
s=8000,e=82ff,datan=304
received data=772(=304) bytes
```

1－2. プログラムのLOADの仕方

[ND80ZⅢの操作]
LOADの操作は簡単で、[＊(I／O)]に続けて[D LD]を押すだけです。
7セグメントLEDの表示は変化しませんが、[RESET]キー以外は受け付けなくなります([D LD]キーはすぐに離してください)。



(図4－2)

[Windowsパソコン側の操作]

hidwr xxxxx. btk[Entr]

と入力します。xxxxx. btkはND80ZⅢに送りたいプログラム、データのファイルです。
受信が完了すると、ND80ZⅢの7セグメントLEDには、メモリに格納された先頭のアドレスと終りのアドレスが表示されます。

(図4－3)

[注記1]
SAVE、LOAD終了後、アドレスが表示されている状態では普通のキー入力モードになっています。したがってこの状態ですぐにRUNキーを押せば、今SAVE(またはLOAD)したプログラムをただちに実行させることができます。またRDINCキーなど他のキーを使うこともできます(キー入力に先立ってリセットする必要はありません)。
なおSAVE、LOADの説明では、「プログラム」のSAVE、LOADということで説明してきましたが、プログラムでも単なるデータでも、扱いは全く同じです。
[注記2]
ND80ZモニタプログラムのSAVEプログラムは、送信データの先頭に2バイトの送信開始アドレスと同じく2バイトの送信終了アドレスを送ります。
[注記3]
LOADは受信したアドレス情報に従って、指定されたメモリアドレスに受信したデータを書き込みます。
指定されたアドレスがROMのアドレス(0000～7FFF)であってもエラーにはなりませんが、結果としては受信しなかったのと同じことになります。
また受信したアドレス情報が、モニタプログラムのワークエリア(83xxまたはF800～FFFF)を示していた場合にはデータの受信によってモニタプログラムが暴走してしまうことがあります。
暴走したときはリセットすれば初期状態に戻ります。

2. RS232C接続

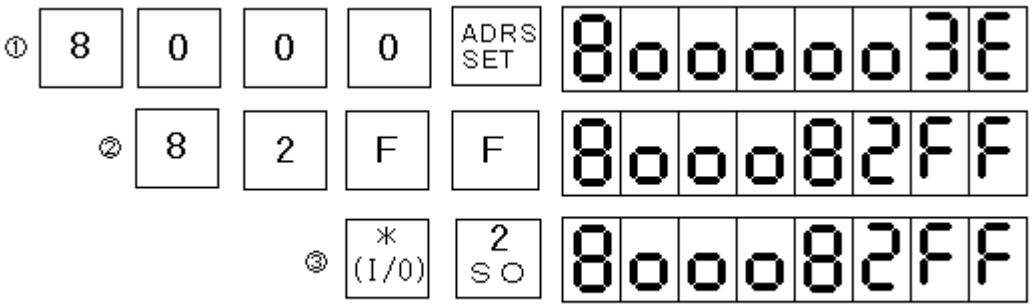
ND80ZⅢにはUSBインターフェースのほかに、RS232Cインターフェースも内蔵していますから、RS232Cケーブルで他のパソコンやRS232Cインターフェースを持った機器との間で、プログラムやデータの送受信を行うことができます。

2－1. RS232Cの送信

RS232Cの送信は、[＊(I／O)]キーに続いて、データキーの[2(SO)]を押すことで行われます。
SOはSerial Outです。

送信前に、送信したいデータ（またはプログラム）が格納されているメモリエリアの先頭アドレスを7セグメントLEDのアドレス表示部に表示し、終了アドレスをデータ表示部に表示しておくことで、その指定した範囲のデータが送信されます。
USBを経由してプログラム、データを送信するときと違い、先頭アドレス、終了アドレスそのものは送信されません。
送信データはASCIIコードに変換することなく、バイナリデータのまま送信されます。

データの送信開始アドレスが8000、終了アドレスが82FFであるときの、キー操作です。

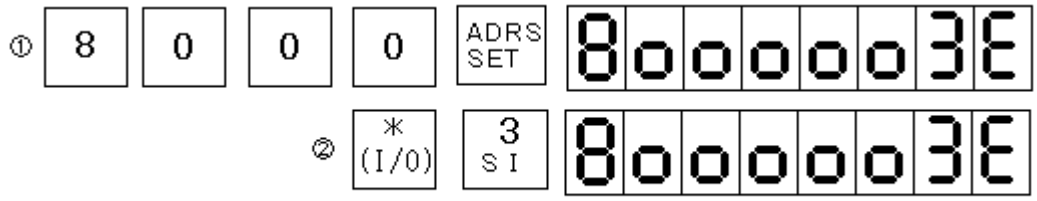


（図4－2）

送信中は7セグメントLEDのアドレス表示部には送信開始アドレスが表示され、データ表示部には現在送信中のアドレスがモニタ表示されます。
送信が完了すると、送信開始アドレスと終了アドレスが7セグメントLEDに表示された状態で静止しますが、この時点でRS232C送信プログラムは終了してTK80モニタのエントリルーチンに戻っていますから、普通にキー操作をすることができます。

2－2. RS232Cの受信

RS232Cの受信は、[＊(I／O)]キーに続いて、データキーの[3(SI)]を押すことで行われます。
SI はSerial Inです。
受信前に、受信データを格納したいメモリエリアの先頭アドレスを7セグメントLEDのアドレス表示部に表示しておくことで、受信データが指定したメモリアドレスから格納されます。



（図4－3）

受信中は7セグメントLEDのアドレス表示部には送信開始アドレスが表示され、データ表示部には現在受信中のアドレスがモニタ表示されます。
受信が終了（送信データが途切れる）しても、そのまま受信待ちで停止しています。ふたたび送信が開始されると、受信動作が再開されます。
送信プログラムと違い、受信の場合には受信データが完了したことを知ることができません。
RS232Cの送信、受信では、[＊(I／O)][C(SV)]、[＊(I／O)][D(LD)]キーを使ってUSB通信を行うときのように、データの先頭に開始アドレスと終了アドレスを置くことはしません。
実際に送信、受信するデータのみを送受信しますから、受信プログラムでは送信が完了したのか、それとも送信側の都合で一時的に送信データが途切れているのかを判断することができません。
もし送信が完了していて、これ以上受信を待つ必要がない状態になったら、リセットすることで、通常のモニタ操作に戻ることができます。

5章　USB接続

1．リモートプログラム

ND80ZⅢはWindowsパソコンとUSBケーブルで接続して、データやプログラムをWindowsパソコンのハードディスクにSAVEしたり、逆にあらかじめ作成してハードディスクに保存しておいた、ND80ZⅢ用のマシン語のプログラムファイルやデータファイルをUSB経由でND80ZⅢにLOADすることができます。
そのほかデバッグのためのモニタプログラムの機能のうちのいくつかは、USB接続によって、Windowsパソコンのディスプレイに表示させることで、より効果的に使うこともできます。

USBケーブルでWindowsパソコンと接続して、それによって拡張されたモニタプログラムの機能を使うためには、Windowsパソコン側にもそのためのプログラムをロードして実行させる必要があります。
そのプログラムはND80ZⅢ組立キットに附属しているCDROMにあります。
ND80ZⅢリモート＋Z80BASICプログラムです。

ND80ZⅢのモニタプログラムの通常の機能は、ND80ZⅢのキーボードを操作して使いますが、WindowsパソコンとUSBで接続して、ND80ZⅢモニタの拡張機能を使うためには、キー操作はND80ZⅢのキーで行うのではなくて、Windowsパソコンのキーボードから行います。

ND80ZⅢリモート＋Z80BASICプログラムの準備や起動方法、使い方などについては、ND80ZⅢリモート＋Z80BASICプログラム操作説明書を参照してください。

2．レジスタダンプ

2章で説明したステップ動作やブレイク動作を、WindowsパソコンとUSBで接続して、Windowsパソコン側でND80ZⅢリモート＋Z80BASICプログラムを起動させることによって、[REG]キー＋データキーの操作で一つずつレジスタの中身を確認しなくても、全レジスタの値を一度にWindowsパソコンのコマンドプロンプト画面に表示させることができます。
USB接続でこの機能を使うときは、ND80ZⅢのキーから操作をするのではなくて、Windowsパソコンのキーボードからキー入力操作を行います。
具体的な操作の仕方については、ND80ZⅢリモート＋Z80BASICプログラム操作説明書を参照してください。

3．トレース（TRACE）機能

ステップ動作は、[CON]キーを繰り返し必要なだけ押す必要がありましたが、トレース機能を使うと[CON]キーを押さなくても、レジスタダンプをUSBで接続したWindowsパソコンのコマンドプロンプト画面に表示しながら自動的に10ステップ分のステップ動作が行われます。
トレース機能の具体的な操作の仕方については、ND80ZⅢリモート＋Z80BASICプログラム操作説明書を参照してください。

4．メモリダンプ

USBで接続したWindowsパソコンのコマンドプロンプト画面に、指定範囲のメモリ内容を、16進で表示するとともに、JIS（ASCII）コードとみなして、キャラクタ（文字） でも表示します。
メモリダンプ機能の具体的な操作の仕方については、ND80ZⅢリモート＋Z80BASICプログラム操作説明書を参照してください。

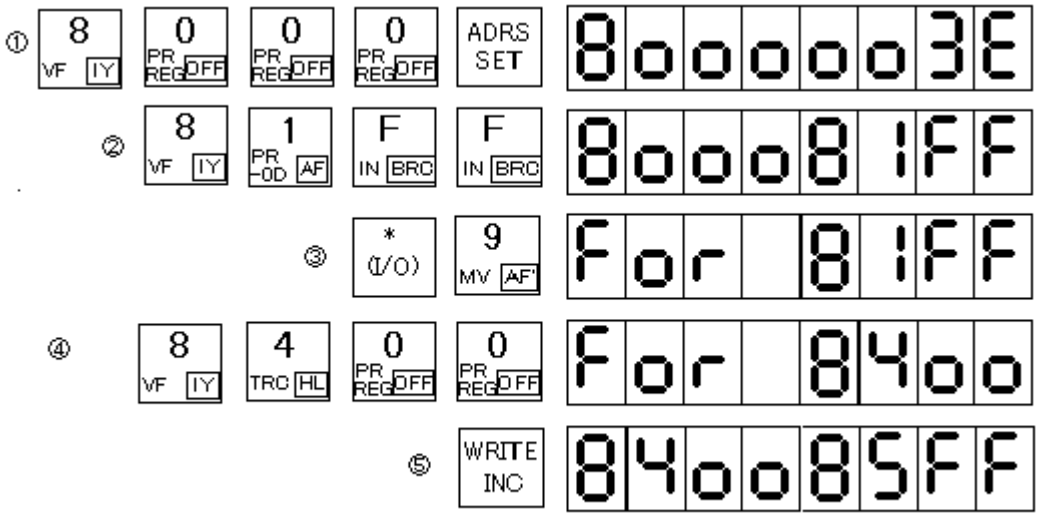
6章　その他の機能

1．MOVE MEMORY

プログラムの変更や追加のために、ある範囲のプログラムを少し移動させたり、ある範囲のデータをそっくりそのまま、別のアドレスに転送したい場合があります。

そんなときにこの機能を使えば、簡単にメモリ内容の転送（COPY）ができます。

例として、8000～81FFのデータを8400からはじまるメモリエリアに転送する仕方を説明します。

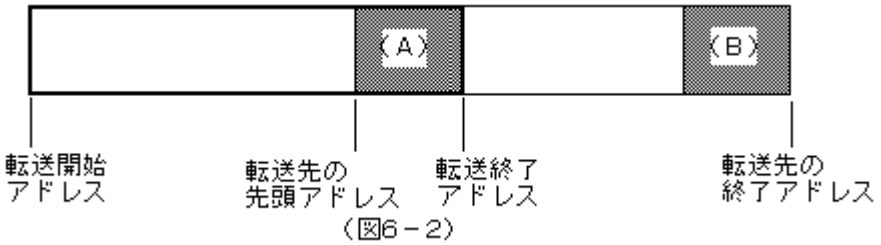


（図6－1）

①転送開始アドレス（ここでは8000）をアドレス表示部に入れます。
②続いて転送終了アドレス（81FF）を入力します（[WRITE　INC]キーを押してしまわないように注意して下さい）。
③[＊（I／O）]キーに続いてデータキーの[9]（[MV]）を押します。するとアドレス表示部にForの表示が出ます（このときデータ表示部は、変化しません。転送終了アドレスを表示したままです）。
④今度は転送先の先頭アドレスを入力します。
⑤[WRITE　INC]キーを押すと、瞬時に転送が完了して、転送後の先頭アドレスと終わりのアドレスを表示します。

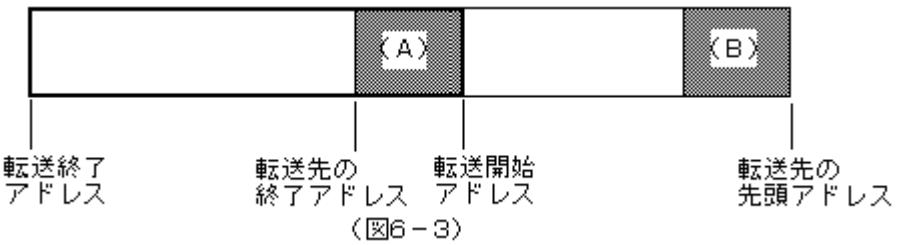
●転送元のメモリ範囲と転送先のメモリ範囲が重なっていても転送は正しく行われます。

普通は小さいアドレスから大きいアドレスの順に転送しますが、図6－2のように転送元と転送先のアドレスに重なりがあると、転送によって重なった部分のデータが壊されてしまうため、正しく転送が行われません。
転送によって（A）が先に壊されてしまうので（B）の部分には壊されたデータが転送されてしまいます。



（図6－2）

そこでこの転送プログラムは、与えられたアドレスをチェックして、上図のような重なりがある場合には、逆に大きいアドレスから小さいアドレスの順に転送するように考えてあります。



（図6－3）

図6－3 のようにすると、A部分は転送の終わりの方で壊されますがそのときにはBへの転送は完了しているため、正しく転送されます。

〔注意1〕転送部分の重なりの判断およびその処理は全てプログラムが行います。従ってアドレスの入力については必ず、転送開始アドレス＜転送終了アドレスという条件で入れて下さい。例えば開始アドレス＝81FF、終了アドレス＝8000と指定すると7セグメントLEDにエラーの表示が出されます。

　転送先の先頭アドレスについては制限はありません。例えば開始アドレス＝8400終了アドレス＝85FFのとき、転送先の先頭アドレス＝8000であっても問題はありません。
〔注意2〕メモリのF800～FFFFの部分はモニタプログラムが使用しています。もし転送先のメモリ範囲に、このF800～FFFFの一部でも含むような指定をすると、モニタプログラムが暴走することがあります。
〔注意3〕この転送機能は転送先でのチェックは行いません。転送先として指定した範囲がROMアドレス（0000～7FFF）の場合、転送は結果的に無意味ですが、特にエラー表示は出ません（転送元がROMのアドレスであることは支障ありません）。
〔注意4〕この転送機能はデータをそのままのかたちで移動するものです。もし移動の対象がプログラムである場合、転送元ではうまく動作していても転送先では正しく動作しなくなる可能性があります（特にジャンプやCALL命令に注意して下さい）。

```
8000  3E00    LD A,00
8002  3C       INC A
8003  C30280  JP $8002
```

このプログラムを8150に転送した場合

```
8150  3E00
8152  3C
8153  C30280  →C35281  に直さなければ、このアドレスでは正しく動作しません。
```

7章　モニタサブルーチン

ND80Zモニタプログラムには、幾つかのサブルーチンが含まれており、この中にはユーザーが利用すると便利なものもあります。

なお、ND80Zモニタプログラムはキー入力やLED表示などの基本的な処理については0400～07FFにある、TK80モニタプログラムのサブルーチンをCALLしています。

0400～07FFにあるサブルーチンは、0000スタートのTK80モニタプログラムのサブルーチンと同じ機能です（ワークエリアだけが83xxではなくてFFxxになっています）。

0400～07FFにあるサブルーチンのエントリアドレスは、0000スタートのTK80モニタプログラムのサブルーチンのエントリアドレスに0400を加算したアドレスになります。

たとえばキー入力ルーチン①のエントリアドレスは0000スタートのTK80モニタプログラムでは0216ですが、0400スタートのTK80モニタプログラムでは0616です。

これらのサブルーチンはエントリアドレスとワークエリアのアドレスが異なるだけですから、ここでの説明は省略します（「TK80モニタプログラム操作説明書」を参照してください）。

8章　モニタプログラムリスト

　ND80Zモニタプログラムは、0400番地台のTK80モニタプログラム内のサブルーチンをCALLしています。0400番地台のTK80モニタプログラム内のサブルーチンは、0000番地台のTK80モニタプログラム内のサブルーチンアドレスに0400を加算したアドレスになっていて、機能もワークアドレスが83xxの代わりにFFxxになっているだけですが、ごく一部のみ、ND80Zモニタの機能のために変更しています。

```
2010/8/24  8:13  NDMON4P.TXT
END=07CE
                ;;; TK80 MONITOR PROGRAM FOR ND80Z(WORK AREAR FFxx)
                ; 09/5/28 09/6/1 6/3 6/5
                ; 10/2/25 3/5 3/6 3/19 3/22 3/23
                ; 10/5/5 SERIAL IN/OUT
                ;10/5/9 PIC INTERFACE 4BITS
                ;5/31SAVE/LOAD ENTRY TO ND3MON FROM NDMON4_G
                ;NDMON4J for ND80Z3 10/6/15 6/16
                ;10/8/21 NDMON4M
                ;8/22 NDMON4N
                ;8/23 NDMON4O
                ;8/24 NDMON4P
                ;
                ;
                ; FFFF-FFF8 7SEGMENT DISPLAY ADDRESS
                ;
                DIG=$FFF8
                ;
                DISP=$FFF4
                KFLAG=$FFF3
                BRKCT=$FFF2
                BRKAD=$FFF0
                ADRES1=$FFEF
                ADRES=$FFEE
                DATA1=$FFED
                DATA=$FFEC
                FSAVE=$FFEA
                BSAVE=$FFE9
                CSAVE=$FFE8
                DSAVE=$FFE7
                ESAVE=$FFE6
                HSAVE=$FFE5
                LSAVE=$FFE4
                SSAVE=$FFE2
                PSAVE=$FFE0
                ;
                ;FFDF-FFCF are used by ND80Z MONITOR
                ;
                RST7=$FFCC
                RST6=$FFC9
                RST5=$FFC6
                RST4=$FFC3
                RST3=$FFC0
                RST2=$FFBD
                RST1=$FFBA
                ;REMOTEWK=$FFB9
                SINERMK=$FFB8
                ;
                MONSP=$FFB8
                USRSP=$F800
```

```
                ;
                NDZENT1=$080F
                NDZENT2=$081B
                ;
                ;
                ORG $0400
                ;
0400 C33B04     JMP MONST
                ;
                ;       ORG $0008
                ;       JMP START
                ;
                ;       ORG $0010
                ;       JMP RST2
                ;
                ;       ORG $0018
                ;       JMP RST3
                ;
                ;       ORG $0020
                ;       JMP RST4
                ;
                ;       ORG $0028
                ;       JMP RST5
                ;
                ;       ORG $0030
                ;       JMP RST6
                ;
                ;       ORG $0038
                ;       JMP RST7
                ;
                ; INITIALIZE ROUTINE
                ;
                ORG $043B
                ;
043B 3EFF       MONST:MVI A,FF
043D D398       OUT 98
043F 21ECFF     LXI H,DATA
0442 060C       MVI B,0C
0444 AF         XRA A
0445 77         MONST2:MOV M,A
0446 23         INX H
0447 05         DCR B
0448 C24504     JNZ MONST2
044B 2100F8     LXI H,USRSP
044E 22E2FF     SHLD SSAVE
                ;
                ; MONITOR START
                ;
0451 3EFF       START:MVI A,FF
0453 D398       OUT 98
0455 31B8FF     LXI SP,MONSP
0458 CDC005     CALL SEGCG
045B CD1606     START2:CALL KEYIN
045E 47         MOV B,A
045F E610       ANI 10
0461 CA8404     JZ DIGIT
0464 78         MOV A,B
0465 E60F       ANI 0F
```

```
0467 0600      MVI B,00
0469 87        ADD A
046A 4F        MOV C,A
046B 217404    LXI H,TABL
046E 09        DAD B
046F 7E        MOV A,M
0470 23        INX H
0471 66        MOV H,M
0472 6F        MOV L,A
0473 E9        PCHL
               ;
0474 CC04      TABL:DW GOTO
0476 F905      DW RESRG
0478 9404      DW ADSET
047A B804      DW ADDCX
047C 9D04      DW ADINX
047E C204      DW MEMW
0480 D504      DW SDATA
0482 0705      DW LDATA
               ;
0484 CDB505    DIGIT:CALL SHIFT
0487 3AECFF    LDA DATA
048A B0        ORA B
048B 32ECFF    STA DATA
048E CDA105    CALL RGDSP
0491 C35104    JMP START
               ;
               ; ADDRESS SET
               ;
0494 2AECFF    ADSET:LHLD DATA
0497 22EEFF    SHLD ADRES
049A C3A104    JMP ADINX2
               ;
               ; MEMORY READ & ADDRESS INCREMENT
               ;
049D 2AEEFF    ADINX:LHLD ADRES
04A0 23        INX H
04A1 CDAD04    ADINX2:CALL MEMR
04A4 22EEFF    ADSTR:SHLD ADRES
04A7 CDA105    CALL RGDSP
04AA C35104    JMP START
               ;
04AD 3AECFF    MEMR:LDA DATA
04B0 32EDFF    STA DATA1
04B3 7E        MOV A,M
04B4 32ECFF    STA DATA
04B7 C9        RET
               ;
               ; MEMORY READ & ADDRESS DECREMENT
               ;
04B8 2AEEFF    ADDCX:LHLD ADRES
04BB 2B        DCX H
04BC CDAD04    CALL MEMR
04BF C3A404    JMP ADSTR
               ;
               ; MEMORY WRITE
               ;
04C2 2AEEFF    MEMW:LHLD ADRES
```

```
04C5 3AECFF    LDA DATA
04C8 77        MOV M,A
04C9 C39D04    JMP ADINX
               ;
               ; MONITOR TO USER CONTROL ROUTINE
               ;
04CC 2AEEFF    GOTO:LHLD ADRES
04CF 22E0FF    SHLD PSAVE
04D2 C3F905    JMP RESRG
               ;
               ; STORE DATA
               ;
04D5 06F7      SDATA:MVI B,F7
04D7 2AECFF    SDATA1:LHLD DATA
04DA EB        XCHG
04DB 2AEEFF    LHLD ADRES
04DE 78        MOV A,B
04DF D398      OUT 98;SET 94=OUT
04E1 7C        MOV A,H
04E2 CD7C06    CALL SOUT
04E5 7D        MOV A,L
04E6 CD7C06    CALL SOUT
04E9 7A        MOV A,D
04EA CD7C06    CALL SOUT
04ED 7B        MOV A,E
04EE CD7C06    CALL SOUT
04F1 2B        DCX H
04F2 23        SDATA2:INX H
04F3 7E        MOV A,M
04F4 CD7C06    CALL SOUT
04F7 CD0107    CALL HDCMP
04FA C2F204    JNZ SDATA2
04FD C30F08    JMP NDZENT1
0500 00        NOP
0501 00        NOP
0502 00        NOP
0503 00        NOP
0504 00        NOP
0505 00        NOP
0506 00        NOP
               ;
               ;LOAD DATA
               ;
0507 06FF      LDATA:MVI B,FF
0509 78        LDATA1:MOV A,B;need this!
050A D398      OUT 98;need this!
050C CDA006    CALL SIN
050F 67        MOV H,A
0510 CDA006    CALL SIN
0513 6F        MOV L,A
0514 CDA006    CALL SIN
0517 57        MOV D,A
0518 CDA006    CALL SIN
051B 5F        MOV E,A
051C 22EEFF    SHLD ADRES
051F EB        XCHG
0520 22ECFF    SHLD DATA
0523 EB        XCHG
```

```
0524 2B       DCX H
0525 23       LDATA2:INX H
0526 CDA006   CALL SIN
0529 77       MOV M,A
052A CD0107   CALL HDCMP
052D C22505   JNZ LDATA2
0530 CDA105   CALL RGDSP
0533 C30F08   JMP NDZENT1
              ;
              ; BREAK ENTRY
              ; BREAK & ONE STEP OPERATION
              ;
              ORG $0551
              ;
0551 E3       BRENT:XTHL
0552 22E0FF   SHLD PSAVE
0555 F5       PUSH PSW
0556 210400   LXI H,$0004
0559 39       DAD SP
055A F1       POP PSW
055B 22E2FF   SHLD SSAVE
055E E1       POP H
055F 31ECFF   LXI SP,DATA
0562 F5       PUSH PSW
0563 C5       PUSH B
0564 D5       PUSH D
0565 E5       PUSH H
0566 31B8FF   LXI SP,MONSP
0569 3AF2FF   LDA BRKCT
056C A7       ANA A
056D CA8B05   JZ BSTOP
0570 2AF0FF   LHLD BRKAD
0573 EB       XCHG
0574 2AE0FF   LHLD PSAVE
0577 7D       MOV A,L
0578 BB       CMP E
0579 C28505   JNZ NOBRK
057C 7C       MOV A,H
057D BA       CMP D
057E C28505   JNZ NOBRK
0581 21F2FF   LXI H,BRKCT
0584 35       DCR M
0585 CD9105   NOBRK:CALL ADDSP
0588 C3F905   JMP RESRG
058B CD9105   BSTOP:CALL ADDSP
058E C35104   JMP START
0591 2AEAFF   ADDSP:LHLD FSAVE
0594 22ECFF   SHLD DATA
0597 2AE0FF   LHLD PSAVE
059A 22EEFF   SHLD ADRES
059D CDA105   CALL RGDSP
05A0 C9       RET
              ;
              ;
              ;;; SUBROUTINE
              ;
05A1 21EFFF   RGDSP:LXI H,ADRES1
05A4 11F4FF   LXI D,DISP
```

```
05A7 0604      MVI B,04
05A9 7E        RGDSP2:MOV A,M
05AA 12        STAX D
05AB 2B        DCX H
05AC 13        INX D
05AD 05        DCR B
05AE C2A905    JNZ RGDSP2
05B1 CDC005    CALL SEGCG
05B4 C9        RET
               ;
               ;DATA REG SHIFT(4 BITS)
               ;
05B5 2AECFF    SHIFT:LHLD DATA
05B8 29        DAD H
05B9 29        DAD H
05BA 29        DAD H
05BB 29        DAD H
05BC 22ECFF    SHLD DATA
05BF C9        RET
               ;
               ; SEGMENT CONVERT SUB
               ;
05C0 21F4FF    SEGCG:LXI H,DISP
05C3 11F8FF    LXI D,DIG
05C6 01E905    LXI B,SEGD
05C9 7E        SEGCG2:MOV A,M
05CA 23        INX H
05CB E5        PUSH H
05CC F5        PUSH PSW
05CD E6F0      ANI F0
05CF 0F        RRC
05D0 0F        RRC
05D1 0F        RRC
05D2 0F        RRC
05D3 2600      MVI H,00
05D5 6F        MOV L,A
05D6 09        DAD B
05D7 7E        MOV A,M
05D8 12        STAX D
05D9 13        INX D
05DA F1        POP PSW
05DB E60F      ANI 0F
05DD 2600      MVI H,00
05DF 6F        MOV L,A
05E0 09        DAD B
05E1 7E        MOV A,M
05E2 12        STAX D
05E3 E1        POP H
05E4 1C        INR E
05E5 C2C905    JNZ SEGCG2
05E8 C9        RET
               ;
               ; SEGMENT DATA
               ;
05E9 5C        SEGD:DB 5C
05EA 06        DB 06
05EB 5B        DB 5B
05EC 4F        DB 4F
```

```
05ED 66       DB 66
05EE 6D       DB 6D
05EF 7D       DB 7D
05F0 27       DB 27
05F1 7F       DB 7F
05F2 6F       DB 6F
05F3 77       DB 77
05F4 7C       DB 7C
05F5 39       DB 39
05F6 5E       DB 5E
05F7 79       DB 79
05F8 71       DB 71
              ;
              ; REGISTER RESTORE
              ;
05F9 2AE2FF   RESRG:LHLD SSAVE
05FC F9       SPHL
05FD 2AE0FF   LHLD PSAVE
0600 E5       PUSH H
0601 2AE4FF   LHLD LSAVE
0604 E5       PUSH H
0605 2AEAFF   LHLD FSAVE
0608 E5       PUSH H
0609 2AE8FF   LHLD CSAVE
060C 4D       MOV C,L
060D 44       MOV B,H
060E 2AE6FF   LHLD ESAVE
0611 EB       XCHG
0612 F1       POP PSW
0613 E1       POP H
0614 FB       EI
0615 C9       RET
              ;
              ; KEY INPUT
              ;
0616 CD2306   KEYIN:CALL INPUT
0619 47       MOV B,A
061A 3AF3FF   LDA KFLAG
061D A7       ANA A
061E CA1606   JZ KEYIN
0621 78       MOV A,B
0622 C9       RET
              ;
              ; KEY INPUT SUB
              ;
0623 CD4706   INPUT:CALL KEY
0626 3C       INR A
0627 CA4206   JZ NOKEY
062A CDEA06   INPUT2:CALL D2
062D CD4706   CALL KEY
0630 47       MOV B,A
0631 3C       INR A
0632 CA4206   JZ NOKEY
0635 3AF3FF   LDA KFLAG
0638 A7       ANA A
0639 C22A06   JNZ INPUT2
063C 3D       DCR A
063D 32F3FF   INPUT3:STA KFLAG
```

```
0640 78       MOV A,B
0641 C9       RET
0642 06FF     NOKEY:MVI B,FF
0644 C33D06   JMP INPUT3
              ;
              ; KEY SCAN & CONVERT HEX DATA SUB
              ;
0647 1600     KEY:MVI D,00
0649 42       MOV B,D
064A 3EFE     MVI A,FE
064C D39C     OUT 9C
064E DB9C     IN 9C
0650 EEFF     XRI FF
0652 C27106   JNZ KEYI
0655 0608     MVI B,08
0657 3EFD     MVI A,FD
0659 D39C     OUT 9C
065B DB9C     IN 9C
065D EEFF     XRI FF
065F C27106   JNZ KEYI
0662 0610     MVI B,10
0664 3EFB     MVI A,FB
0666 D39C     OUT 9C
0668 DB9C     IN 9C
066A EEFF     XRI FF
066C C27106   JNZ KEYI
066F 3D       DCR A
0670 C9       RET
0671 0F       KEYI:RRC
0672 DA7906   JC KEYI2
0675 14       INR D
0676 C37106   JMP KEYI
0679 7A       KEYI2:MOV A,D
067A B0       ORA B
067B C9       RET
              ;
              ;SERIAL OUTPUT ROUTINE
              ;
067C 4F       SOUT:MOV C,A
067D DB94     SOUT2:IN 94
067F E640     ANI 40
0681 CA7D06   JZ SOUT2
0684 CD2607   CALL SOUTSB
0687 78       MOV A,B; I/Oaddress 94 "out" set & STROBE ON
0688 E6FD     ANI FD;bit1=0
068A D398     OUT 98
068C DB94     SOUT3:IN 94
068E E640     ANI 40
0690 C28C06   JNZ SOUT3
0693 78       MOV A,B; I/Oaddress 94 "out" set & STROBE OFF
0694 D398     OUT 98
0696 C9       RET
              ;
              ;SERIAL INPUT ROUTINE
              ;
              ORG $06A0
              ;
06A0 78       SIN:MOV A,B
```

```
06A1 D398     OUT 98
06A3 CD8D07   SIN2:CALL SINSB
06A6 CAA306   JZ SIN2
06A9 79       MOV A,C
06AA C9       RET
              ;
              ;CHATTERING TIMER
              ;
              ORG $06DD
06DD 1624     D1:MVI D,24;=36 ck=7 124.83*36+(7+10)/6=4496.71microsec
06DF 1E34     D1_2:MVI E,34;=52 ck=7  7+14*52+14=749 749/6=124.83microsec
06E1 1D       D1_3:DCR E;       ck=4
06E2 C2E106   JNZ D1_3; ck=10
06E5 15       DCR D;    ck=4
06E6 C2DF06   JNZ D1_2; ck=10
06E9 C9       RET;      ck=10
06EA 1648     D2:MVI D,48;=72  124.83*72+(7+10+10)/6=8992.26microsec
06EC C3DF06   JMP D1_2
06EF 16D8     D3:MVI D,D8;=216 124.83*216+27/6=26967.78microsec
06F1 C3DF06   JMP D1_2
              ;
06F4 22ECFF   HLDTDP:SHLD DATA
06F7 C5       PSHRGDP:PUSH B
06F8 D5       PUSH D
06F9 E5       PUSH H
06FA CDA105   CALL RGDSP
06FD E1       POP H
06FE D1       POP D
06FF C1       POP B
0700 C9       RET
              ;
0701 7D       HDCMP:MOV A,L
0702 BB       CMP E
0703 C0       RNZ
0704 7C       MOV A,H
0705 BA       CMP D
0706 C9       RET
              ;
0707 32B8FF   LDERR:STA SINERMK
070A 211E07   LXI H,ERRT
070D CD1307   CALL SEGDP
0710 C31B08   JMP NDZENT2
              ;
0713 11F8FF   SEGDP:LXI D,DIG
0716 7E       SEGDP2:MOV A,M
0717 12       STAX D
0718 23       INX H
0719 1C       INR E
071A C21607   JNZ SEGDP2
071D C9       RET
              ;
071E 79       ERRT:DB 79;E
071F 50       DB 50;r
0720 50       DB 50;r
0721 5C       DB 5C;o
0722 50       DB 50;r
0723 80       DB 80
0724 80       DB 80
```

```
0725 80       DB 80
              ;
0726 79       SOUTSB:MOV A,C
0727 D394     OUT 94
0729 78       MOV A,B
072A E6FD     ANI FD
072C D398     OUT 98
072E DB94     SOUTSB2:IN 94
0730 E640     ANI 40
0732 C22E07   JNZ SOUTSB2
0735 78       MOV A,B
0736 D398     OUT 98
0738 DB94     SOUTSB3:IN 94
073A E640     ANI 40
073C CA3807   JZ SOUTSB3
073F 79       MOV A,C
0740 0F       RRC
0741 0F       RRC
0742 0F       RRC
0743 0F       RRC
0744 D394     OUT 94
0746 C9       RET
              ;
              ; RS232C SAVE & LOAD
0747 2AECFF   RSAVE:LHLD DATA
074A EB       XCHG
074B 2AEEFF   LHLD ADRES
074E 06F3     MVI B,F3
0750 2B       DCX H
0751 23       RSAVE2:INX H
0752 7E       MOV A,M
0753 CD7C06   CALL SOUT
0756 3EFF     MVI A,FF
0758 D398     OUT 98
075A 22ECFF   SHLD DATA
075D CDF706   CALL PSHRGDP
0760 CD0107   CALL HDCMP
0763 C25107   JNZ RSAVE2
0766 C30F08   JMP NDZENT1
              ;
0769 2AEEFF   RLOAD:LHLD ADRES
076C 06FB     MVI B,FB
076E 2B       DCX H
076F 23       RLOAD2:INX H
0770 CD7D07   CALL RSIN
0773 77       MOV M,A
0774 22ECFF   SHLD DATA
0777 CDF706   CALL PSHRGDP
077A C36F07   JMP RLOAD2
              ;
077D 78       RSIN:MOV A,B
077E D398     OUT 98
0780 CD8D07   RSIN1:CALL SINSB
0783 79       MOV A,C
0784 C0       RNZ
0785 FEFF     CPI FF
0787 C20707   JNZ LDERR
078A C38007   JMP RSIN1
```

```
                ;
078D DB94       SINSB:IN 94
078F E620       ANI 20
0791 CA8D07     JZ SINSB
0794 78         MOV A,B;BUSY
0795 E6FE       ANI FE;bit0=0
0797 D398       OUT 98
0799 DB94       SINSB2:IN 94
079B E620       ANI 20
079D C29907     JNZ SINSB2
07A0 DB94       IN 94
07A2 E610       ANI 10
07A4 F5         PUSH PSW
07A5 DB94       IN 94
07A7 E60F       ANI 0F
07A9 4F         MOV C,A
07AA 78         MOV A,B;READY
07AB D398       OUT 98
07AD DB94       SINSB3:IN 94
07AF E620       ANI 20
07B1 CAAD07     JZ SINSB3
07B4 78         MOV A,B
07B5 E6FE       ANI FE
07B7 D398       OUT 98
07B9 DB94       SINSB4:IN 94
07BB E620       ANI 20
07BD C2B907     JNZ SINSB4
07C0 DB94       IN 94
07C2 07         RLC
07C3 07         RLC
07C4 07         RLC
07C5 07         RLC
07C6 E6F0       ANI F0
07C8 B1         ORA C
07C9 4F         MOV C,A
07CA 78         MOV A,B
07CB D398       OUT 98
07CD F1         POP PSW
07CE C9         RET
                ;END
ADDCX       =04B8  ADDSP       =0591  ADINX       =049D
ADINX2      =04A1  ADRES       =FFEE  ADRES1      =FFEF
ADSET       =0494  ADSTR       =04A4  BRENT       =0551
BRKAD       =FFF0  BRKCT       =FFF2  BSAVE       =FFE9
BSTOP       =058B  CSAVE       =FFE8  D1          =06DD
D1_2        =06DF  D1_3        =06E1  D2          =06EA
D3          =06EF  DATA        =FFEC  DATA1       =FFED
DIG         =FFF8  DIGIT       =0484  DISP        =FFF4
DSAVE       =FFE7  ERRT        =071E  ESAVE       =FFE6
FSAVE       =FFEA  GOTO        =04CC  HDCMP       =0701
HLDTDP      =06F4  HSAVE       =FFE5  INPUT       =0623
INPUT2      =062A  INPUT3      =063D  KEY         =0647
KEYI        =0671  KEYI2       =0679  KEYIN       =0616
KFLAG       =FFF3  LDATA       =0507  LDATA1      =0509
LDATA2      =0525  LDERR       =0707  LSAVE       =FFE4
MEMR        =04AD  MEMW        =04C2  MONSP       =FFB8
MONST       =043B  MONST2      =0445  NDZENT1     =080F
NDZENT2     =081B  NOBRK       =0585  NOKEY       =0642
```

```
PSAVE       =FFE0  PSHRGDP     =06F7  RESRG       =05F9
RGDSP       =05A1  RGDSP2      =05A9  RLOAD       =0769
RLOAD2      =076F  RSAVE       =0747  RSAVE2      =0751
RSIN        =077D  RSIN1       =0780  RST1        =FFBA
RST2        =FFBD  RST3        =FFC0  RST4        =FFC3
RST5        =FFC6  RST6        =FFC9  RST7        =FFCC
SDATA       =04D5  SDATA1      =04D7  SDATA2      =04F2
SEGCG       =05C0  SEGCG2      =05C9  SEGD        =05E9
SEGDP       =0713  SEGDP2      =0716  SHIFT       =05B5
SIN         =06A0  SIN2        =06A3  SINERMK     =FFB8
SINSB       =078D  SINSB2      =0799  SINSB3      =07AD
SINSB4      =07B9  SOUT        =067C  SOUT2       =067D
SOUT3       =068C  SOUTSB      =0726  SOUTSB2     =072E
SOUTSB3     =0738  SSAVE       =FFE2  START       =0451
START2      =045B  TABL        =0474  USRSP       =F800

2010/9/9  6:53  ND3MON2O.TXT
END=0DE3
              ;;; ND80Z MONITOR FOR ND80Z3
              ;;; ND3MON
              ;2010/02/24 2/25 2/26 3/2 3/5 3/6 3/15 3/22 3/23
              ;;;5/5
              ;; 5/24 for remote
              ;5/25 5/26 5/27 5/29 5/30 5/31 6/10 6/13 6/14
              ;6/15 for ND80Z3 6/16 6/17 8/20
              ;10/8/21 ND3MON2M
              ;9/5 nd3mon2n
              ;9/9 nd3mon2o
              ;
                    LSEG8=$FFFF
                    LSEG7=$FFFE
                    LSEG5=$FFFC
                    LSEG4=$FFFB
                    LSEG3=$FFFA
                    LSEG2=$FFF9
                    LSEG1=$FFF8
                    DISP3=$FFF6
                    DISP1=$FFF4
                    ;KFLAG=$FFF3
                    BRKC=$FFF2
                    BRKA=$FFF0
                    ADRSH=$FFEF
                    ADRSL=$FFEE
                    DATAH=$FFED
                    DATAL=$FFEC
                    AREG=$FFEB
                    FREG=$FFEA
                    CREG=$FFE8
                    EREG=$FFE6
                    LREG=$FFE4
                    SPL=$FFE2
                    PCL=$FFE0
                    LDSH=$FFD4
                    IREG=$FFD3
                    TRSW=$FFD1
                    IOREG=$FFD0
              RMODE=$FFCF
                    RST7JA=$FFCD
```

```
                RST7JC=$FFCC
              RST6JA=$FFCA
              RST6JC=$FFC9
              RST5JA=$FFC7
              RST5JC=$FFC6
              RST4JA=$FFC4
              RST4JC=$FFC3
              RST3JA=$FFC1
              RST3JC=$FFC0
              RST2JA=$FFBE
              RST2JC=$FFBD
              RST1JA=$FFBB
              RST1JC=$FFBA
              REMOTEMK=$FFB9
              SINERMK=$FFB8
              ;
                SPTOP=$FFB8
              PRCNT=$FF50
              PRBF=$FF00
                USTOP=$F800
              ;
              ;
              TKRST6=$83DD
              TKRST5=$83DA
              TKRST4=$83D7
              TKRST3=$83D4
              TKRST2=$83D1
              TKRST1=$0051
              ;
              TKMON=$003B
              TKBREAK=$0151
              D2=$02EA
              ;
              TK2MON=$043B
              TK2RST1=$0451
              TK2BREAK=$0551
              ;
              LEDDPA=$05C0;SEGCG
              LEDDP=$FFC9;=RST6JC
              RGDSP=$05A1
              SEGCG1=$05C6
              KEYINA=$0616
              KEYIN=$FFC6;=RST5JC
              INPUT=$0623
              KEY=$0647
              ;
              LOAD=$0507
              SOUTSB=$067C
              SINSB=$06A0
              SOUT=$0747
              SIN=$0769
              ;
              ;     ND80Z MONITOR
              ;
                ORG $0800
              ;
0800 C32408           JP NDMON
0803 C3FF0A           JP SEGDP
```

```
0806 C3080B         JP LPRTSB
0809 C3150B         JP LPRTSB2
080C C35D0B         JP CRLF
080F C38E08    JP START1
0812 C3D30B    JP PRHX4
0815 C3D70B    JP PRHX2
0818 C3E00B    JP PRHX1
081B C39808    JP START2
081E C31D0B    JP LPRTSB22
0821 C34B0D    JP REMOTE
               ;
0824 3EFF      NDMON:LD A,FF
0826 ED47      LD I,A
0828 D398      OUT (98),A
082A 31B8FF    LD SP,SPTOP
082D AF        XOR A
082E 32B9FF    LD (REMOTEMK),A
0831 0607      LD B,07
0833 21CCFF    LD HL,RST7JC
0836 3EC3      LD A,C3
0838 77        NDMON2:LD (HL),A
0839 2B        DEC HL
083A 2B        DEC HL
083B 2B        DEC HL
083C 10FA      DJNZ NDMON2
083E DB94      IN A,(94)
0840 E680      AND 80
0842 CA6608    JP Z,NDMON3;No.1=ON
               ;TK80 MONITOR
0845 215101    LD HL,TKBREAK
0848 22CDFF    LD (RST7JA),HL
084B 215100    LD HL,TKRST1
084E 22BBFF    LD (RST1JA),HL
0851 11DD83    LD DE,TKRST6
0854 21CAFF    LD HL,RST6JA
0857 0605      LD B,05
0859 36DE      NDMON2_2:LD (HL),DE
085B 2B        DEC HL
085C 2B        DEC HL
085D 2B        DEC HL
085E 1B        DEC DE
085F 1B        DEC DE
0860 1B        DEC DE
0861 10F6      DJNZ NDMON2_2
0863 C33B00    JP TKMON
               ;
0866 21C005    NDMON3:LD HL,LEDDPA
0869 22CAFF         LD (RST6JA),HL
086C 211606    LD HL,KEYINA
086F 22C7FF    LD (RST5JA),HL
0872 215104    LD HL,TK2RST1
0875 22BBFF    LD (RST1JA),HL
               ;
0878 21D209    LD HL,BREAK
087B 22CDFF    LD (RST7JA),HL
087E 21CFFF    LD HL,RMODE
0881 0629      LD B,29
0883 AF        XOR A
```

```
0884 77       LD (HL),A
0885 23       INC HL
0886 10FC     DJNZ FC
0888 2100F8   LD HL,USTOP
088B 22E2FF   LD (SPL),HL
              ;
088E 3EFF     START1:LD A,FF
0890 D398     OUT (98),A
0892 31B8FF   LD SP,SPTOP
0895 CDC9FF         CALL LEDDP
0898 CDC6FF   START2:CALL KEYIN
089B 47             LD B,A
089C E610           AND 10
089E CADE08         JP Z,KDIN
08A1 3ACFFF         LD A,(RMODE)
08A4 B7             OR A
08A5 CABE08         JP Z,KCIN
08A8 21AE08         LD HL,JPT1
08AB C3C108         JP KCIN2
              ;
              ;;; KEY JUMP TABLE (1)
08AE 9808     JPT1:DW START2
08B0 9808           DW START2
08B2 9808           DW START2
08B4 D70A           DW RRDEC
08B6 C80A           DW RRINC
08B8 E40A           DW RWINC
08BA 9808           DW START2
08BC 650A           DW RMOD
              ;
              ;
              ; COMMAND KEYIN
08BE 21CE08   KCIN:LD HL,JPT2
08C1 78       KCIN2:LD A,B
08C2 E607           AND 07
08C4 0600     KCIN3:LD B,00
08C6 87             ADD A,A
08C7 4F             LD C,A
08C8 09             ADD HL,BC
08C9 7E             LD A,(HL)
08CA 23             INC HL
08CB 66             LD H,(HL)
08CC 6F             LD L,A
08CD E9             JP (HL)
              ;
              ;;; KEY JUMP TABLE (2)
              ;
08CE A609     JPT2:DW RUN
08D0 B209           DW CONT
08D2 7409           DW ADST
08D4 9509           DW RDEC
08D6 7A09           DW RINC
08D8 9C09           DW WINC
08DA 8D0A           DW IOMOD
08DC 650A           DW RMOD
                    ;;;
              ;
              ;0-F KEY INPUT
```

```
08DE 3AD0FF   KDIN:LD A,(IOREG)
08E1 B7             OR A
08E2 CA6409         JP Z,KINDP
08E5 F21509         JP P,RGDIN; =01
              ; =80 or 81
08E8 AF             XOR A
08E9 32D0FF         LD (IOREG),A
08EC 21F508         LD HL,JPT3
08EF 78             LD A,B
08F0 E60F           AND 0F
08F2 C3C408         JP KCIN3
              ;
              ;;; KEY JUMP TABLE (3)
              ;
08F5 5B0A     JPT3:DW RPRT
08F7 9808           DW START2;LPRT
08F9 4707           DW SOUT
08FB 6907           DW SIN
08FD 560A           DW TRON
08FF 9808           DW START2;W256
0901 9808           DW START2;R256
0903 9808           DW START2;START2
0905 4B0D           DW REMOTE
0907 3C0C           DW MMEM
0909 DF0C           DW DMEM
090B 9808           DW START2;DAS
090D B40C           DW LSAVE
090F 0705           DW LOAD
0911 BC0A           DW OUT
0913 950A           DW IN
              ;;;
              ;
              ;0-F KEY INPUT (REG MODE)
0915 AF       RGDIN:XOR A
0916 32D0FF         LD (IOREG),A
0919 32F8FF         LD (LSEG1),A
091C 32FBFF         LD (LSEG4),A
091F 78             LD A,B
0920 87             ADD A,A
0921 CA5E09         JP Z,RGMDE
0924 32CFFF   RGDI2:LD (RMODE),A
0927 4F             LD C,A
0928 0600           LD B,00
092A 21BC0D         LD HL,RSGDT
092D 09             ADD HL,BC
092E 5E             LD E,(HL)
092F 23             INC HL
0930 56             LD D,(HL)
0931 EB             EX DE,HL
0932 22F9FF         LD (LSEG2),HL
0935 21ECFF         LD HL,DATAL
0938 ED42           SBC HL,BC
093A FE1C           CP 1C
093C DA5109         JP C,RGDI3
093F 21F0FF         LD HL,BRKA
0942 CA5109         JP Z,RGDI3
0945 21F2FF         LD HL,BRKC
0948 22EEFF   LD (ADRSL),HL
```

```
094B 6E       LD L,(HL)
094C 2600     LD H,00
094E C35809   JP RGDI4
0951 22EEFF   RGDI3:LD (ADRSL),HL
0954 5E             LD E,(HL)
0955 23             INC HL
0956 56             LD D,(HL)
0957 EB             EX DE,HL
0958 22ECFF   RGDI4:LD (DATAL),HL
095B C39E0A         JP IN1
              ;
              ; REG MODE END
095E 32CFFF   RGMDE:LD (RMODE),A
0961 C38409         JP RINC3
              ;
              ;KEY INPUT DATA DISP
0964 CD4B0A   KINDP:CALL DSFTL
0967 3AECFF         LD A,(DATAL)
096A B0             OR B
096B 32ECFF   KIDP1:LD (DATAL),A
096E CD770A   KIDP2:CALL DRDSP
0971 C39808         JP START2;cannot to IN2!!! 10/5/27
              ;
              ; ADDRESS SET
0974 2AECFF    ADST:LD HL,(DATAL)
0977 C37E09         JP RINC2
              ;
              ; READ INC
097A 2AEEFF    RINC:LD HL,(ADRSL)
097D 23             INC HL
097E CD8A09   RINC2:CALL DTSET
0981 22EEFF         LD (ADRSL),HL
0984 CDA80D   RINC3:CALL ADDSP
0987 C39808         JP START2
              ;
098A 3AECFF   DTSET:LD A,(DATAL)
098D 32EDFF         LD (DATAH),A
0990 7E             LD A,(HL)
0991 32ECFF         LD (DATAL),A
0994 C9             RET
              ;
              ; READ DEC
0995 2AEEFF    RDEC:LD HL,(ADRSL)
0998 2B             DEC HL
0999 C37E09         JP RINC2
              ;
              ; WRITE INC
099C 2AEEFF    WINC:LD HL,(ADRSL)
099F 3AECFF         LD A,(DATAL)
09A2 77             LD (HL),A
09A3 C37A09         JP RINC
              ;
              ; RUN
09A6 2AEEFF     RUN:LD HL,(ADRSL)
09A9 22E0FF         LD (PCL),HL
09AC 7C       LD A,H
09AD B5       OR L
09AE F3       DI
```

```
09AF CA0000    JP Z,$0000
               ;
               ; CONTINUE
09B2 31D4FF     CONT:LD SP,LDSH
09B5 E1              POP HL
09B6 D1              POP DE
09B7 C1              POP BC
09B8 D9              EXX
09B9 F1              POP AF
09BA 08              EX AF,AF'
09BB FDE1            POP IY
09BD DDE1            POP IX
09BF 31E6FF          LD SP,EREG
09C2 D1              POP DE
09C3 C1              POP BC
09C4 F1              POP AF
09C5 ED7BE2FF        LD SP,(SPL)
09C9 2AE0FF          LD HL,(PCL)
09CC E5              PUSH HL
09CD 2AE4FF          LD HL,(LREG)
09D0 FB              EI
09D1 C9              RET
               ;
               ; BREAK
09D2 22E4FF    BREAK:LD (LREG),HL
09D5 E1              POP HL;=PC
09D6 ED53E6FF        LD (EREG),DE
09DA ED43E8FF        LD (CREG),BC
09DE F5              PUSH AF
09DF 22E0FF    LD (PCL),HL
09E2 22EEFF          LD (ADRSL),HL
09E5 E1              POP HL;=AF
09E6 22EAFF          LD (FREG),HL
09E9 22ECFF          LD (DATAL),HL
09EC ED73E2FF        LD (SPL),SP
09F0 31E0FF          LD SP,PCL
09F3 DDE5            PUSH IX
09F5 FDE5            PUSH IY
09F7 08              EX AF,AF'
09F8 F5              PUSH AF
09F9 D9              EXX
09FA C5              PUSH BC
09FB D5              PUSH DE
09FC E5              PUSH HL
09FD ED57            LD A,I
09FF 67              LD H,A
0A00 ED5F            LD A,R
0A02 6F              LD L,A
0A03 E5              PUSH HL
               ;
0A04 31B8FF    LD SP,SPTOP
0A07 3AB9FF    LD A,(REMOTEMK)
0A0A B7        OR A
0A0B CCA80D    CALL Z,ADDSP
0A0E 3AF2FF    LD A,(BRKC)
0A11 B7        OR A
0A12 CA360A    JP Z,BRKON
0A15 2AE0FF    LD HL,(PCL)
```

```
0A18 ED5BF0FF LD DE,(BRKA)
0A1C B7       OR A
0A1D ED52     SBC HL,DE
0A1F C2260A   JP NZ,NOBRK
0A22 3D       DEC A
0A23 32F2FF   LD (BRKC),A
0A26 CD4706   NOBRK:CALL KEY
0A29 FEFF     CP FF
0A2B CAB209   JP Z,CONT
0A2E FE16     CP 16;*I/O
0A30 C2B209   JP NZ,CONT
0A33 C39808   JP START2
              ;
0A36 3AB9FF   BRKON:LD A,(REMOTEMK)
0A39 B7       OR A
0A3A C48D0B   CALL NZ,REGDP
0A3D 3AD1FF   LD A,(TRSW)
0A40 B7       OR A
0A41 CA9808   JP Z,START2
0A44 3D       DEC A
0A45 32D1FF   LD (TRSW),A
0A48 C3B209   JP CONT
              ;
0A4B 2AECFF   DSFTL:LD HL,(DATAL)
0A4E 29             ADD HL,HL
0A4F 29             ADD HL,HL
0A50 29             ADD HL,HL
0A51 29             ADD HL,HL
0A52 22ECFF         LD (DATAL),HL
0A55 C9             RET
              ;;;
0A56 3E10     TRON:LD A,10
0A58 C35D0A         JP RPRT2
              ;;;
0A5B 3E01     RPRT:LD A,01
0A5D 21D1FF   RPRT2:LD HL,TRSW
0A60 AE       XOR (HL)
0A61 77       LD (HL),A
0A62 C37E09         JP RINC2
              ;;;
0A65 210201   RMOD:LD HL,$0102
0A68 22CFFF         LD (RMODE),HL
0A6B 215050         LD HL,$5050
0A6E 22F8FF         LD (LSEG1),HL
0A71 22FAFF         LD (LSEG3),HL
0A74 C36E09         JP KIDP2
0A77 21EDFF   DRDSP:LD HL,DATAH
0A7A 11F6FF         LD DE,DISP3
0A7D D5             PUSH DE
0A7E 0602           LD B,02
0A80 7E             LD A,(HL)
0A81 12             LD (DE),A
0A82 2B             DEC HL
0A83 13             INC DE
0A84 10FA           DJNZ FA
0A86 E1             POP HL
0A87 11FCFF         LD DE,LSEG5
0A8A C3C605         JP SEGCG1
```

```
                ;
0A8D 3E80       IOMOD:LD A,80
0A8F 32D0FF           LD (IOREG),A
0A92 C39808           JP START2
                ;
                ;
0A95 3AEEFF     IN:LD A,(ADRSL)
0A98 4F               LD C,A
0A99 ED78             IN A,(C)
0A9B 32ECFF           LD (DATAL),A
0A9E CD770A     IN1:CALL DRDSP
0AA1 3AB9FF     IN2:LD A,(REMOTEMK)
0AA4 B7         OR A
0AA5 CA9808     JP Z,START2
0AA8 21EFFF     LD HL,ADRSH
0AAB 11F4FF     LD DE,DISP1
0AAE 0602       LD B,02
0AB0 7E         IN3:LD A,(HL)
0AB1 12         LD (DE),A
0AB2 2B         DEC HL
0AB3 13         INC DE
0AB4 10FA       DJNZ IN3
0AB6 CD890D     CALL RLEDOUT1
0AB9 C39808             JP START2
                ;;;
0ABC 3AEEFF       OUT:LD A,(ADRSL)
0ABF 4F               LD C,A
0AC0 3AECFF           LD A,(DATAL)
0AC3 ED79             OUT (C),A
0AC5 C3A10A           JP IN2
                ;
                ; READ INC(REG MODE)
0AC8 3ACFFF     RRINC:LD A,(RMODE)
0ACB 3C               INC A
0ACC 3C               INC A
0ACD FE20             CP 20
0ACF DA2409           JP C,RGDI2
0AD2 3E02       RRINC2:LD A,02
0AD4 C32409           JP RGDI2
                ;
                ; READ DEC(REG MODE)
0AD7 3ACFFF     RRDEC:LD A,(RMODE)
0ADA 3D               DEC A
0ADB 3D               DEC A
0ADC C22409           JP NZ,RGDI2
0ADF 3E1E             LD A,1E
0AE1 C32409           JP RGDI2
                ;
                ; WRITE INC(REG MODE)
0AE4 2AEEFF     RWINC:LD HL,(ADRSL)
0AE7 ED5BECFF         LD DE,(DATAL)
0AEB 73               LD (HL),E
0AEC 3ACFFF     LD A,(RMODE)
0AEF FE1E       CP 1E
0AF1 CAD20A     JP Z,RRINC2
0AF4 23               INC HL
0AF5 72               LD (HL),D
0AF6 C3C80A           JP RRINC
```

```
                ;
0AF9 7C         HDCMP:LD A,H
0AFA BA               CP D
0AFB C0               RET NZ
0AFC 7D               LD A,L
0AFD BB               CP E
0AFE C9               RET
                ;
0AFF 11F8FF     SEGDP:LD DE,LSEG1
0B02 010800           LD BC,$0008
0B05 EDB0             LDIR
0B07 C9               RET
                ;
0B08 1A         LPRTSB:LD A,(DE)
0B09 13         INC DE
0B0A F5         PUSH AF
0B0B CD150B           CALL LPRTSB2
0B0E F1         POP AF
0B0F FE0D       CP 0D
0B11 C2080B           JP NZ,LPRTSB
0B14 C9         RET
                ;
0B15 FE0D       LPRTSB2:CP 0D
0B17 CA5D0B     JP Z,CRLF
0B1A FE0A       CP 0A
0B1C C8         RET Z
0B1D E5         LPRTSB22:PUSH HL
0B1E D5         PUSH DE
0B1F 2150FF     LD HL,PRCNT
0B22 5E         LD E,(HL)
0B23 16FF       LD D,FF
0B25 12         LD (DE),A
0B26 13         INC DE
0B27 7B         LD A,E
0B28 FE50       CP 50
0B2A CA3B0B     JP Z,LPRTSB3
0B2D 77         LD (HL),A
0B2E D1         POP DE
0B2F E1         POP HL
0B30 C9               RET
                ;
0B31 E5         BFOUT:PUSH HL
0B32 D5         PUSH DE
0B33 3A50FF     LD A,(PRCNT)
0B36 5F         LD E,A
0B37 B7         OR A
0B38 CA5A0B     JP Z,LPRTSB6
                ;
                ; BUFFER FULL
0B3B 2100FF     LPRTSB3:LD HL,PRBF
0B3E C5         PUSH BC
0B3F 3EF7       LD A,F7
0B41 47         LD B,A
0B42 D398       OUT (98),A
0B44 7B         LD A,E
0B45 CD7C06     CALL SOUTSB
0B48 7E         LPRTSB4:LD A,(HL)
0B49 CD7C06     CALL SOUTSB
```

```
0B4C 23       INC HL
0B4D 1D       DEC E
0B4E C2480B   JP NZ,LPRTSB4
0B51 AF       LPRTSB5:XOR A
0B52 3250FF   LD (PRCNT),A
0B55 C1       POP BC
0B56 3EFF     LD A,FF
0B58 D398     OUT (98),A
0B5A D1       LPRTSB6:POP DE
0B5B E1       POP HL
0B5C C9       RET
              ;
0B5D E5       CRLF:PUSH HL
0B5E D5       PUSH DE
0B5F C5       PUSH BC
0B60 3EF7     LD A,F7
0B62 47       LD B,A
0B63 D398     OUT (98),A
0B65 3A50FF   LD A,(PRCNT)
0B68 B7       OR A
0B69 F5       PUSH AF
0B6A 5F       LD E,A
0B6B 3C       INC A;FOR 0D
0B6C 3C       INC A;FOR 0A
0B6D CD7C06   CALL SOUTSB
0B70 F1       POP AF
0B71 CA800B   JP Z,CRLF3
0B74 2100FF   LD HL,PRBF
0B77 7E       CRLF2:LD A,(HL)
0B78 CD7C06   CALL SOUTSB
0B7B 23       INC HL
0B7C 1D       DEC E
0B7D C2770B   JP NZ,CRLF2
0B80 3E0D     CRLF3:LD A,0D
0B82 CD7C06   CALL SOUTSB
0B85 3E0A     LD A,0A
0B87 CD7C06   CALL SOUTSB
0B8A C3510B   JP LPRTSB5
              ;
0B8D 11EE0B   REGDP:LD DE,REG_PR_T
0B90 CD080B         CALL LPRTSB
0B93 060D           LD B,0D
0B95 11EBFF         LD DE,AREG
0B98 CDCD0B   REGDP2:CALL PRDEHX4
0B9B 3E20           LD A,20
0B9D CD150B         CALL LPRTSB2
0BA0 05             DEC B
0BA1 C2980B         JP NZ,REGDP2
0BA4 3AEAFF         LD A,(FREG)
0BA7 4F             LD C,A
0BA8 0608           LD B,08
0BAA CB11     REGDP3:RL C
0BAC DAB40B         JP C,REGDP4
0BAF 3E30           LD A,30
0BB1 C3B60B         JP REGDP5
0BB4 3E31     REGDP4:LD A,31
0BB6 CD150B   REGDP5:CALL LPRTSB2
0BB9 05             DEC B
```

```
0BBA C2AA0B          JP NZ,REGDP3
0BBD 3E20            LD A,20
0BBF CD150B          CALL LPRTSB2
0BC2 3AF2FF          LD A,(BRKC)
0BC5 67        LD H,A
0BC6 CDD70B          CALL PRHX2
0BC9 CD5D0B          CALL CRLF
0BCC C9              RET
               ;
0BCD 1A        PRDEHX4:LD A,(DE)
0BCE 67              LD H,A
0BCF 1B              DEC DE
0BD0 1A              LD A,(DE)
0BD1 6F              LD L,A
0BD2 1B              DEC DE
0BD3 CDD70B    PRHX4:CALL PRHX2
0BD6 65              LD H,L
0BD7 7C        PRHX2:LD A,H
0BD8 0F              RRCA
0BD9 0F              RRCA
0BDA 0F              RRCA
0BDB 0F              RRCA
0BDC CDE00B          CALL PRHX1
0BDF 7C              LD A,H
0BE0 E60F      PRHX1:AND 0F
0BE2 C630            ADD A,30
0BE4 FE3A            CP 3A
0BE6 DA150B          JP C,LPRTSB2
0BE9 C607            ADD A,07
0BEB C3150B          JP LPRTSB2
               ;
               ; REGISTER DUMP TITLE(for PRINTER)
0BEE 41204620 REG_PR_T:"A F "
0BF2 20422043 " B C"
0BF6 20204420 "  D "
0BFA 45202048 "E  H"
0BFE 204C2020 " L  "
0C02 20535020 " SP "
0C06 20205043 "  PC"
0C0A 20202049 "   I"
0C0E 58202020 "X   "
0C12 49592020 "IY  "
0C16 41274627 "A'F'"
0C1A 20422743 " B'C"
0C1E 27204427 "' D'"
0C22 45272048 "E' H"
0C26 274C2720 "'L' "
0C2A 49205220 "I R "
0C2E 20535A20 " SZ "
0C32 4820504E "H PN"
0C36 43204252 "C BR"
0C3A 43        "C"
0C3B 0D        DB 0D
               ;;;
0C3C ED4BEEFF  MMEM:LD BC,(ADRSL)
0C40 2AECFF          LD HL,(DATAL)
0C43 C5              PUSH BC
0C44 E5              PUSH HL
```

```
0C45 DDE1           POP IX
0C47 FDE1           POP IY
0C49 B7             OR A
0C4A ED42           SBC HL,BC
0C4C DAAB0C         JP C,ERRDP
0C4F 23             INC HL
0C50 E5             PUSH HL
0C51 21715C         LD HL,$5C71;Fo
0C54 22F8FF         LD (LSEG1),HL
0C57 215000         LD HL,$0050;r
0C5A 22FAFF         LD (LSEG3),HL
0C5D CD770A   MMEM2:CALL DRDSP
0C60 CDC6FF   MMEM3:CALL KEYIN
0C63 FE10           CP 10
0C65 DA700C         JP C,MMEM4
0C68 FE15           CP 15
0C6A CA7E0C         JP Z,MMEM5
0C6D C3600C         JP MMEM3
0C70 47       MMEM4:LD B,A
0C71 CD4B0A         CALL DSFTL
0C74 3AECFF         LD A,(DATAL)
0C77 B0             OR B
0C78 32ECFF         LD (DATAL),A
0C7B C35D0C         JP MMEM2
0C7E ED5BECFF MMEM5:LD DE,(DATAL)
0C82 C1             POP BC
0C83 FDE5           PUSH IY
0C85 E1             POP HL
0C86 CDF90A         CALL HDCMP
0C89 DA940C         JP C,MMEM6
0C8C D5             PUSH DE
0C8D EDB0           LDIR
0C8F E1             POP HL
0C90 1B             DEC DE
0C91 C3A10C         JP MMEM7
0C94 EB       MMEM6:EX DE,HL
0C95 09             ADD HL,BC
0C96 EB             EX DE,HL
0C97 1B             DEC DE
0C98 DDE5           PUSH IX
0C9A E1             POP HL
0C9B D5             PUSH DE
0C9C EDB8           LDDR
0C9E E1             POP HL
0C9F 13             INC DE
0CA0 EB             EX DE,HL
0CA1 22EEFF   MMEM7:LD (ADRSL),HL
0CA4 ED53ECFF LD (DATAL),DE
0CA8 C38409         JP RINC3
              ;;;
0CAB 21DC0D   ERRDP:LD HL,ERRT
0CAE CDFF0A   CALL SEGDP
0CB1 C39808   JP START2
              ;;;
0CB4 2AEEFF   LSAVE:LD HL,(ADRSL)
0CB7 ED5BECFF LD DE,(DATAL)
0CBB 7C       LD A,H
0CBC CD1D0B   CALL LPRTSB22
```

```
0CBF 7D       LD A,L
0CC0 CD1D0B   CALL LPRTSB22
0CC3 7A       LD A,D
0CC4 CD1D0B   CALL LPRTSB22
0CC7 7B       LD A,E
0CC8 CD1D0B   CALL LPRTSB22
0CCB 7E       SAVE2:LD A,(HL)
0CCC CD1D0B   CALL LPRTSB22
0CCF CDF90A   CALL HDCMP
0CD2 CAD90C   JP Z,SAVE3
0CD5 23       INC HL
0CD6 C3CB0C   JP SAVE2
0CD9 CD310B   SAVE3:CALL BFOUT
0CDC C39808   JP START2
              ;;;
0CDF 2AEEFF    DMEM:LD HL,(ADRSL)
0CE2 ED5BECFF       LD DE,(DATAL)
0CE6 E5       DMEM2:PUSH HL
0CE7 E5             PUSH HL
0CE8 CDD30B         CALL PRHX4
0CEB E1             POP HL
0CEC CD460D         CALL PRSP
0CEF 0610           LD B,10
0CF1 CD460D   DMEM3:CALL PRSP
0CF4 E5             PUSH HL
0CF5 66             LD H,(HL)
0CF6 CDD70B         CALL PRHX2
0CF9 E1             POP HL
0CFA 23             INC HL
0CFB 05             DEC B
0CFC C2F10C         JP NZ,DMEM3
0CFF E1             POP HL
0D00 0610           LD B,10
0D02 CD460D         CALL PRSP
0D05 CD460D         CALL PRSP
0D08 7E             LD A,(HL)
0D09 CD2D0D         CALL ASCDUMP
0D0C 23             INC HL
0D0D 05             DEC B
0D0E 20F8           JR NZ,F8
0D10 CD5D0B         CALL CRLF
0D13 2B             DEC HL
0D14 CDF90A         CALL HDCMP
0D17 D22A0D         JP NC,DMEM5
0D1A 23             INC HL
0D1B D5       DMEM4:PUSH DE
0D1C CD4706   CALL KEY
0D1F D1       POP DE
0D20 FEFF     CP FF
0D22 CAE60C         JP Z,DMEM2
0D25 FE16           CP 16
0D27 C21B0D         JP NZ,DMEM4
0D2A C39808   DMEM5:JP START2
              ;
0D2D FE20     ASCDUMP:CP 20
0D2F DA410D         JP C,ASCDUMP2
0D32 FE7F           CP 7F
0D34 DA150B         JP C,LPRTSB2
```

```
0D37 FEA0           CP A0
0D39 DA410D         JP C,ASCDUMP2
0D3C FEE0           CP E0
0D3E DA150B         JP C,LPRTSB2
0D41 3E2E      ASCDUMP2:LD A,2E
0D43 C3150B         JP LPRTSB2
                    ;
0D46 3E20      PRSP:LD A,20
0D48 C3150B         JP LPRTSB2
                    ;
                    ; REMOTE
                    ;
0D4B 211234    REMOTE:LD HL,$3412
0D4E 22F4FF    LD (DISP1),HL
0D51 215678    LD HL,$7856
0D54 22F6FF    LD (DISP3),HL
0D57 CDC005    CALL LEDDPA;=SEGCG   not send data to HOST
0D5A 21860D    LD HL,RLEDOUT
0D5D 22CAFF    LD (RST6JA),HL
0D60 21720D    LD HL,RSIN
0D63 22C7FF    LD (RST5JA),HL
0D66 3EFF      LD A,FF
0D68 32B9FF    LD (REMOTEMK),A
0D6B AF        XOR A
0D6C 3250FF    LD (PRCNT),A
0D6F C39808    JP START2
                    ;
0D72 3EF7      RSIN:LD A,F7
0D74 C5        PUSH BC
0D75 47        LD B,A
0D76 D398      OUT (98),A
0D78 AF        XOR A
0D79 CD7C06    CALL SOUTSB
0D7C 3EFF      LD A,FF
0D7E 47        LD B,A
0D7F D398      OUT (98),A
0D81 CDA006    CALL SINSB
0D84 C1        POP BC
0D85 C9        RET
                    ;
0D86 CDC005    RLEDOUT:CALL LEDDPA
0D89 3EF7      RLEDOUT1:LD A,F7
0D8B C5        PUSH BC
0D8C 47        LD B,A
0D8D D398      OUT (98),A
0D8F 3E04      LD A,04
0D91 CD7C06    CALL SOUTSB
0D94 1E04      LD E,04
0D96 21F4FF    LD HL,DISP1
0D99 7E        RLEDOUT2:LD A,(HL)
0D9A CD7C06    CALL SOUTSB
0D9D 23        INC HL
0D9E 1D        DEC E
0D9F C2990D    JP NZ,RLEDOUT2
0DA2 C1        POP BC
0DA3 3EFF      LD A,FF
0DA5 D398      OUT (98),A
0DA7 C9        RET
```

```
                ;
0DA8 21EFFF     ADDSP:LD HL,ADRSH
0DAB 11F4FF     LD DE,DISP1
0DAE 0604       LD B,04
0DB0 7E         ADDSP2:LD A,(HL)
0DB1 12         LD (DE),A
0DB2 2B         DEC HL
0DB3 13         INC DE
0DB4 05         DEC B
0DB5 C2B00D     JP NZ,ADDSP2
0DB8 CDC9FF     CALL LEDDP
0DBB C9         RET
                ;
                ;;; SEGMENT DATA(REG MODE)
                ;
0DBC FF         RSGDT:RST 7;DUMMY
0DBD FF               RST 7;DUMMY
0DBE 77               DB 77;A
0DBF 71               DB 71;F
0DC0 7C               DB 7C;b
0DC1 39               DB 39;C
0DC2 5E               DB 5E;d
0DC3 79               DB 79;E
0DC4 76               DB 76;H
0DC5 38               DB 38;L
0DC6 6D               DB 6D;S
0DC7 73               DB 73;P
0DC8 73               DB 73;P
0DC9 39               DB 39;C
0DCA 06               DB 06;I
0DCB 76               DB 76;X
0DCC 06               DB 06;I
0DCD 6E               DB 6E;y
0DCE F7               DB F7;A.
0DCF F1               DB F1;F.
0DD0 FC               DB FC;b.
0DD1 B9               DB B9;C.
0DD2 DE               DB DE;d.
0DD3 F9               DB F9;E.
0DD4 F6               DB F6;H.
0DD5 B8               DB B8;L.
0DD6 06               DB 06;I
0DD7 50               DB 50;r
0DD8 50               DB 50;r
0DD9 77               DB 77;A
0DDA 50               DB 50;r
0DDB 39               DB 39;C
                ;
0DDC 79         ERRT:DB 79;E
0DDD 50         DB 50;r
0DDE 50         DB 50;r
0DDF 5C         DB 5C;o
0DE0 50         DB 50;r
0DE1 80         DB 80
0DE2 80         DB 80
0DE3 80         DB 80
                ;END
ADDSP          =0DA8  ADDSP2        =0DB0  ADRSH          =FFEF
```

ADRSL	=FFEE	ADST	=0974	AREG	=FFEB
ASCDUMP	=0D2D	ASCDUMP2	=0D41	BFOUT	=0B31
BREAK	=09D2	BRKA	=FFF0	BRKC	=FFF2
BRKON	=0A36	CONT	=09B2	CREG	=FFE8
CRLF	=0B5D	CRLF2	=0B77	CRLF3	=0B80
D2	=02EA	DATAH	=FFED	DATAL	=FFEC
DISP1	=FFF4	DISP3	=FFF6	DMEM	=0CDF
DMEM2	=0CE6	DMEM3	=0CF1	DMEM4	=0D1B
DMEM5	=0D2A	DRDSP	=0A77	DSFTL	=0A4B
DTSET	=098A	EREG	=FFE6	ERRDP	=0CAB
ERRT	=0DDC	FREG	=FFEA	HDCMP	=0AF9
IN	=0A95	IN1	=0A9E	IN2	=0AA1
IN3	=0AB0	INPUT	=0623	IOMOD	=0A8D
IOREG	=FFD0	IREG	=FFD3	JPT1	=08AE
JPT2	=08CE	JPT3	=08F5	KCIN	=08BE
KCIN2	=08C1	KCIN3	=08C4	KDIN	=08DE
KEY	=0647	KEYIN	=FFC6	KEYINA	=0616
KIDP1	=096B	KIDP2	=096E	KINDP	=0964
LDSH	=FFD4	LEDDP	=FFC9	LEDDPA	=05C0
LOAD	=0507	LPRTSB	=0B08	LPRTSB2	=0B15
LPRTSB22	=0B1D	LPRTSB3	=0B3B	LPRTSB4	=0B48
LPRTSB5	=0B51	LPRTSB6	=0B5A	LREG	=FFE4
LSAVE	=0CB4	LSEG1	=FFF8	LSEG2	=FFF9
LSEG3	=FFFA	LSEG4	=FFFB	LSEG5	=FFFC
LSEG7	=FFFE	LSEG8	=FFFF	MMEM	=0C3C
MMEM2	=0C5D	MMEM3	=0C60	MMEM4	=0C70
MMEM5	=0C7E	MMEM6	=0C94	MMEM7	=0CA1
NDMON	=0824	NDMON2	=0838	NDMON2_2	=0859
NDMON3	=0866	NOBRK	=0A26	OUT	=0ABC
PCL	=FFE0	PRBF	=FF00	PRCNT	=FF50
PRDEHX4	=0BCD	PRHX1	=0BE0	PRHX2	=0BD7
PRHX4	=0BD3	PRSP	=0D46	RDEC	=0995
REGDP	=0B8D	REGDP2	=0B98	REGDP3	=0BAA
REGDP4	=0BB4	REGDP5	=0BB6	REG_PR_T	=0BEE
REMOTE	=0D4B	REMOTEMK	=FFB9	RGDI2	=0924
RGDI3	=0951	RGDI4	=0958	RGDIN	=0915
RGDSP	=05A1	RGMDE	=095E	RINC	=097A
RINC2	=097E	RINC3	=0984	RLEDOUT	=0D86
RLEDOUT1	=0D89	RLEDOUT2	=0D99	RMOD	=0A65
RMODE	=FFCF	RPRT	=0A5B	RPRT2	=0A5D
RRDEC	=0AD7	RRINC	=0AC8	RRINC2	=0AD2
RSGDT	=0DBC	RSIN	=0D72	RST1JA	=FFBB
RST1JC	=FFBA	RST2JA	=FFBE	RST2JC	=FFBD
RST3JA	=FFC1	RST3JC	=FFC0	RST4JA	=FFC4
RST4JC	=FFC3	RST5JA	=FFC7	RST5JC	=FFC6
RST6JA	=FFCA	RST6JC	=FFC9	RST7JA	=FFCD
RST7JC	=FFCC	RUN	=09A6	RWINC	=0AE4
SAVE2	=0CCB	SAVE3	=0CD9	SEGCG1	=05C6
SEGDP	=0AFF	SIN	=0769	SINERMK	=FFB8
SINSB	=06A0	SOUT	=0747	SOUTSB	=067C
SPL	=FFE2	SPTOP	=FFB8	START1	=088E
START2	=0898	TK2BREAK	=0551	TK2MON	=043B
TK2RST1	=0451	TKBREAK	=0151	TKMON	=003B
TKRST1	=0051	TKRST2	=83D1	TKRST3	=83D4
TKRST4	=83D7	TKRST5	=83DA	TKRST6	=83DD
TRON	=0A56	TRSW	=FFD1	USTOP	=F800
WINC	=099C				