

ND8080 ZB3DOS(CP/M互換DOS)操作説明書

(有)中日電工

目次

I 準備	1
1. はじめに	1
2. CDROMの内容	1
2.1 CPPフォルダ	1
2.2 N8ZB3DOSフォルダ	1
2.3 N8ZB3DOS説明書PDFフォルダ	1
2.4 ASM80フォルダ	1
3. N8ZB3DOS. EXEが動作するWindowsOSについて	2
4. CDROMのコピー	2
4.1 Windows7の場合	2
4.2 WindowsXPの場合	3
4.3 Windows2000の場合	3
4.4 Windows98、SE、Meの場合	4
5. USBケーブルを使ってND8080とDOS/Vパソコンを接続する	5
6. N8ZB3DOS. EXEの起動	5
6.1 コマンドプロンプト(MSDOSプロンプト)を開く	5
6.2 [NumRock]の確認	5
6.3 キー入力モードの確認	5
6.4 ND8080とパソコンをUSBケーブルで接続します	5
6.5 ND8080の電源を入れる	5
6.6 DOSプロンプト(コマンドプロンプト)でN8ZB3DOS. exeを実行する	6
6.7 ND8080のリモート機能をONにします [* (I/O)][8 VF]	6
7. ZB3BASICの起動	6
8.1 ログファイルについて	7
8.2 ログファイルのクローズ	7
9. N8ZB3DOSの終了	7
10. N8ZB3DOSの強制終了	7
II N8ZB3DOSで拡張された機能、コマンド	8
1. LLIST	8
2. LPRINT	8
3. DM@	8
4. CM@	8
5. BP@	8
6. RT@	9
7. BP@0	9
8. JP@	9
9. /LD	10
10. /SV	10
11. /BAT	10
11.1 バッチファイルの中で使える特殊記号	11
1) 'またはREM (注釈行)	11
2) p (コメントの表示)	11
3) \$\$1~\$\$5 (文字変数)	11
11.2 制限事項	11
11.3 データの入力	11
11.4 エラーメッセージ	11
12. /CPM	11
13. /CPM, D	11
14. ZB3BASIC用STARTREK	12
III CP/M互換DOSの基本情報	13
1. CP/M互換DOSのバージョン	13
2. CP/M互換DOSの起動	13
3. 仮想フロッピーディスクの構造	13
4. VFDUMP. EXE	14
5. Zドライブ	14
6. CP/M互換DOSの構造	14

1) M CCP. BIN (LOADアドレスCC00)	14	
2) MBDOS. BIN (LOADアドレスD406)	15	
3) N8BIOS. BIN (LOADアドレスE200)	15	
7. CP/M互換DOSのRAMアドレス	15	
8. システムブレイク(リブート)	15	
9. 0000~00FFの内容	15	
IV CP/M互換DOSのビルトインコマンドとトランザクションファイル		17
1. DIR	17	
2. ERA	17	
3. TYPE	18	
4. SAVE	18	
5. REN	18	
6. COPY	18	
6. 1 COPY ファイル名1. 拡張子a ファイル名2. 拡張子b	18	
6. 2 COPY ファイル名1. 拡張子a x:ファイル名2. 拡張子b	19	
6. 3 COPY ファイル名1. 拡張子a x:	19	
6. 4 COPY x:ファイル名1. 拡張子a	19	
7. ZB3	19	
8. チェンジドライブ	19	
9. リードオンリーディスク	19	
10. トランジェントファイルの実行	20	
V ファンクションコール	21	
0. ファンクション00 システムリセット	23	
1. ファンクション01 コンソール入力	23	
2. ファンクション02 コンソール出力	23	
3. ファンクション03 リーダー(RDR)入力(RS232C入力)	24	
4. ファンクション04パンチ(PUN)出力(RS232C出力)	24	
5. ファンクション05 リスト(LST)出力	24	
6. ファンクション06 コンソール入出力	24	
7. ファンクション07 IOバイト取り出し	25	
8. ファンクション08 IOバイトセット	25	
9. ファンクション09 文字列出力	25	
10. ファンクション0A コンソールバッファ入力	26	
11. ファンクション0B コンソールステータスチェック	26	
12. ファンクション0C バージョン№取り出し	27	
13. ファンクション0D ディスクリセット	27	
14. ファンクション0E ディスクドライブセレクト	27	
15. ファンクション0F ファイルオープン	28	
16. ファンクション10 ファイルクローズ	28	
17. ファンクション11 最初のファイルサーチ	28	
18. ファンクション12 次のファイルサーチ	29	
19. ファンクション13 ファイル削除	29	
20. ファンクション14 シーンシャルリード	29	
21. ファンクション15 シーケンシャルライト	30	
22. ファンクション16 新規ファイル作成(新規ファイルのオープン)	30	
23. ファンクション17 ファイル名の変更	31	
24. ファンクション18 ログインベクトルの取り出し	31	
25. ファンクション19 ログインディスク№の取り出し	32	
26. ファンクション1A DMAアドレスのセット	32	
27. ファンクション1B アローケーションアドレスの取り出し	32	
28. ファンクション1C ライトプロテクトセット	33	
29. ファンクション1D R/Oベクトルの取り出し	33	
30. ファンクション1E ファイルアトリビュートセット	33	
31. ファンクション1F ディスクパラメータアドレスのセット	34	
32. ファンクション20 ユーザーコードのセット	35	
33. ファンクション21 ランダムリード	35	
34. ファンクション22 ランダムライト	35	
35. ファンクション23 ファイルサイズの計算	36	

36.	ファンクション24	ランダムレコードアクセスポインタの更新	36
37.	ファンクション25	ディスクドライブのリセット	36
38.	ファンクション26	未定義	37
39.	ファンクション27	未定義	37
40.	ファンクション28	ゼロファイルを伴うランダムライト	37
VI	エラーコード		38
VII	N8BIOSサブルーチン		39
0)	E200	コールドブート	39
1)	E203	ウォームブート	39
2)	E206	コンソールステータス	39
3)	E209	コンソール入力	39
4)	E20C	コンソール出力	39
5)	E20F	リスト出力	39
6)	E212	232C出力	39
7)	E215	232C入力	39
8)	E218	ホームポジションシーク	39
9)	E21B	セレクトドライブ	40
10)	E21E	セットトラック	40
11)	E221	セットセクタ	40
12)	E224	DMAアドレスセット	40
13)	E227	セクタリード	40
14)	E22A	セクタライト	40
15)	E22D	プリンタステータス	40
16)	E230	セクタトランスレート	40
VIII	CP/Mアプリケーションソフト(参考)		41
1)	MBASIC	(BASIC-80)	41
2)	STARTREK		41
3)	FORTRAN80		42
4)	MACRO80		42
5)	WordStar		42
6)	COBOL-80		43
IX	エスケープシーケンス		44

〒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
TEL052-791-6254 Fax052-791-1391
E-mail thisida@alles.or.jp
Homepage <http://www.tyunitidenko.x0.com/>

2016. 5. 4 Rev. 1. 0

ND8080用ZB3DOS(CP/M互換DOS)取扱説明書

I 準備

1. はじめに

ND8080用ZB3DOSはND8080に付属のN8ZB3. EXEを機能拡張してCP/M互換DOSの機能やその他の機能を付加したものです。

ファイル名はN8ZB3DOS. EXEです。

N8ZB3. EXEを機能拡張したものですから、N8ZB3. EXEに含まれていた機能はそのまま使えます。

また操作方法等もN8ZB3. EXEと同じです。

この説明書では重複を避けるために、もともとN8ZB3. EXEに含まれている機能については説明を省いてあります。それらの機能については、ND8080に付属の各説明書を参照してください。

N8ZB3DOS. EXEはND8080に32KBのRAMを増設して、64KBRAM+システムROMの構成で使うことを前提としています。

N8ZB3DOS. EXEで追加された拡張機能を使うためには、ND8080のIC16(オプション)にICソケットを実装してそこに62256RAMを実装してRAM64K+ROM32Kの構成にする必要があります。

2. CDROMの内容

附属CDROMには、下記の各フォルダがあります。

CPP
N8ZB3DOS
N8ZB3DOS説明書PDF
ASM80

2.1 CPPフォルダ

「CPP」フォルダにはN8ZB3DOS. EXEのソースプログラムN8ZB3DOS. CPPが入っています。

またN8ZB3DOS. CPPをBorland C++コンパイラVersion5. 5でコンパイルして生成された各ファイルも入っています。

コンパイル後に作られたN8ZB3DOS. exeは「N8ZB3DOS」フォルダにもコピーしてあります。

なお、CPPフォルダのN8ZB3DOS. CPP、N8ZB3DOS. EXEはファイル名にバージョンを示す英数字が付加されています。

2016年5月現在はN8ZB3DOS1E. CPP、N8ZB3DOS1E. EXEです。

2.2 N8ZB3DOSフォルダ

「N8ZB3DOS」フォルダには、ND8080をUSBコネクタでWindowsパソコンと接続して、すぐにN8ZB3DOS. EXEを実行することができる全てのプログラムやデータファイルが含まれています。

またCP/M互換機能のトレーニングや理解を助けるために、ホームページ上で公開したサンプルプログラム(. bin)も入っています。

「N8ZB3DOS」フォルダをそのままハードディスクにコピーして、コマンドプロンプトの作業フォルダをコピー後の「N8ZB3DOS」フォルダにすれば、この説明書にある作業は全て、そのコマンドプロンプト上で行うことができます。

N8ZB3DOSフォルダにはユーザーの便を考慮してN8ZB3DOS. EXEを最初に実行したときに生成されるZフォルダおよびA. VFD~D. VFDファイルも作成してあります。

ZフォルダにはBDOSファンクションコールのテストプログラムが拡張子 . COMのファイル形式で入れてあります。

2.3 N8ZB3DOS説明書PDFフォルダ

この説明書のPDFファイルがあります。

Adobe Acrobat ReaderなどのPDF閲覧ソフトウェアで閲覧したり印刷することができます。

2.4 ASM80フォルダ

N8ZB3DOSのCP/M互換プログラム部分(N8BIOS. BIN、MBDOS. BIN、MCCP. BIN)のソースプログラムやホームページ上で説明をしたBDOSファンクションコールのテストプログラムのソースリストなどが入っています。

またそれらのソースプログラムを8080アセンブラZASM. COMでアセンブルして生成された各ファイルも入っています。

そのファイルのうち、BDOSファンクションコールのテストプログラムのバイナリファイルは拡張子を . COMに変えて¥N8ZB3DOS¥Zフォルダにコピーしてあります。

なおN8BIOS. BIN、MBDOS. BIN、MCCP. BINの3つのプログラムはBINレベルで合わせてN8CPM. BINというファイルにしてあります。

3. N8ZB3DOS. EXEが動作するWindowsOSについて

USB接続が必要なため、Windows98、98SE、98Me、Windows2000、WindowsXP、Windows7には接続して実行できますがWindowsNTには接続できません。

Windows7はHome Premium32ビット版と64ビット版で動作確認をしました。WindowsXPはHome Edition(32ビット)での確認です。WindowsXPの64ビット版は確認していません。

Vistaは未確認ですが、WindowsXPとWindows7で動作することは確認できていますから、おそらく問題はないと思います。

4. CDROMのコピー

ND8080の付属CDROMから、ND8080フォルダをCDドライブにコピーしたのと同じ要領で、N8ZB3DOS付属CDROMからN8ZB3DOSフォルダをCDドライブにコピーします。

以下OSごとに説明します。

付属CDROMをCDROMドライブ(DVDドライブ)にセットします。ドライブ番号はパソコンのハードウェア構成によって異なりますが、以下の説明ではD:にCDROM(DVD)ドライブが割り当てられているものとします。

またローカルディスクがCDドライブとして説明をします。

4.1 Windows7の場合

「N8ZB3DOS」フォルダをローカルディスク(C:)にコピーします。コピーの方法はいろいろあります。慣れている方法でコピーしてください。

操作例:「スタート」→「コンピュータ」→「DVDドライブ(D:)」の順にダブルクリックして開き「N8ZB3DOS」フォルダを右クリックしてメニューを開き、「コピー」をクリックします。次に「ローカルディスク(C:)」を右クリックしてメニューを開き、「貼り付け」をクリックします。

「ローカルディスク(C:)」に「N8ZB3DOS」フォルダができたことを確認してください。

次にコマンドプロンプトをすぐに使えるようにします。

「スタート」→「すべてのプログラム」→「アクセサリ」→「コマンドプロンプト」の順にマウスで選択し、「コマンドプロンプト」を右クリックします。

メニューが開くのでその中の「コピー」をクリックします。

次にデスクトップ(起動後のショートカットアイコンが並んでいる画面)のアイコンが無い地の部分にマウスを持って行って、そこで右クリックします。ここでもメニューが開くので「貼り付け」をクリックします。

デスクトップに「コマンドプロンプト」のアイコンができたことを確認してください。

デスクトップにできたコマンドプロンプトのアイコンを右クリックします。

メニューの中の「プロパティ」をクリックします。

「コマンドプロンプトのプロパティ」の「ショートカット」タブをクリックします。

「作業フォルダ」を書き換えます。

初期状態では「%HOMEDRIVE%%HOMEPATH%」になっているはずですが、ここを「%HOMEDRIVE%¥N8ZB3DOS」に書き換えてください(かならず半角で入れてください。半角なら大文字でも小文字でもよいのですが、全角では正しく実行されません)。

次に「レイアウト」タブをクリックします。

画面バッファのサイズとウィンドウのサイズを、幅(W):80、高さ(H):25に設定してください。

ウィンドウのサイズは、初期状態では幅:80、高さ:25になっていますから、変更の必要はないはずですが、確認はしておいてください。画面バッファサイズの高さは大きい値になっていますから、▼ボタンで25まで下げてください。

このとき25より小さい値まで下げると、ウィンドウのサイズの高さも連動して小さくなってしまいますが、画面バッファのサイズの高さを25まで上げても、ウィンドウのサイズは大きくなりません。それぞれ25になるように操作してください。

最後に「OK」をクリックします。

コマンドプロンプトアイコンの下に表示されている名前も変更しておきます。さきほどと同じようにコマンドプロンプトアイコンを右クリックしてメニューを開き、「名前の変更」をクリックします。名前(コマンドプロンプト)が白抜き文字で表示されるので、適当な名前に変更します。「N8ZB3DOS」でよいでしょう(ここは全角でも漢字でも構いません)。

4.2 WindowsXPの場合

「N8ZB3DOS」フォルダをローカルディスク(C:)にコピーします。コピーの方法はいろいろあります。慣れている方法でコピーしてください。

操作例: 「マイコンピュータ」→「Dドライブ」の順にダブルクリックして開き「N8ZB3DOS」フォルダを右クリックしてメニューを開き、「コピー」をクリックします。次に「マイコンピュータ」をダブルクリックして開き「ローカルディスク」(C:)を右クリックしてメニューを開き、「貼り付け」をクリックします。

「ローカルディスク」(C:)に「N8ZB3DOS」フォルダができたことを確認してください。

次にコマンドプロンプトをすぐに使えるようにします。

「スタート」→「すべてのプログラム」→「アクセサリ」→「コマンドプロンプト」の順にマウスで選択し、「コマンドプロンプト」を右クリックします。

メニューが開くのでその中の「コピー」をクリックします。

次にデスクトップ(起動後のショートカットアイコンが並んでいる画面)のアイコンが無い地の部分にマウスを持っていて、そこで右クリックします。ここでもメニューが開くので「貼り付け」をクリックします。

デスクトップに「コマンドプロンプト」のアイコンができたことを確認してください。

デスクトップにできたコマンドプロンプトのアイコンを右クリックします。

メニューの中の「プロパティ」をクリックします。

「コマンドプロンプトのプロパティ」の「ショートカット」タブをクリックします。

「作業フォルダ」を書き換えます。

初期状態では「%HOMEDRIVE%\HOMEPATH%」になっているはずですが、ここを「%HOMEDRIVE%\N8ZB3DOS」に書き換えてください(かならず半角で入れてください。半角なら大文字でも小文字でもよいのですが、全角では正しく実行されません)。

次に「レイアウト」タブをクリックします。

画面バッファのサイズとウィンドウのサイズを、幅(W):80、高さ(H):25に設定してください。

ウィンドウのサイズは、初期状態では幅:80、高さ:25になっていますから、変更の必要はないはずですが、確認はしておいてください。画面バッファサイズの高さは大きい値になっていますから、▼ボタンで25まで下げてください。

このとき25より小さい値まで下げると、ウィンドウのサイズの高さも連動して小さくなってしまいますが、画面バッファのサイズの高さを25まで上げても、ウィンドウのサイズは大きくなりません。それぞれ25になるように操作してください。

最後に「OK」をクリックします。

コマンドプロンプトアイコンの下に表示されている名前も変更しておきます。さきほどと同じようにコマンドプロンプトアイコンを右クリックしてメニューを開き、「名前の変更」をクリックします。名前(コマンドプロンプト)が白抜き文字で表示されるので、適当な名前に変更します。「N8ZB3DOS」でよいでしょう(ここは全角でも漢字でも構いません)。

4.3 Windows2000の場合

「N8ZB3DOS」フォルダをローカルディスク(C:)にコピーします。コピーの方法はいろいろあります。慣れている方法でコピーしてください。

操作例: 「マイコンピュータ」→「Dドライブ」の順にダブルクリックして開き「N8ZB3DOS」フォルダを右クリックしてメニューを開き、「コピー」をクリックします。次に「マイコンピュータ」をダブルクリックして開きローカルディスク(C:)を右クリッ

くして

メニューを開き、「貼り付け」をクリックします。

ローカルディスク(C:)に「N8ZB3DOS」フォルダができたことを確認してください。

次にコマンドプロンプトをすぐに使えるようにします。

「スタート」→「プログラム」→「アクセサリ」→「コマンドプロンプト」の順にマウスで選択し、「コマンドプロンプト」を右クリックします。

メニューが開くのでその中の「コピー」をクリックします。

次にデスクトップ(起動後のショートカットアイコンが並んでいる画面)のアイコンが無い地の部分にマウスを持って行って、そこで右クリックします。ここでもメニューが開くので「貼り付け」をクリックします。

デスクトップに「コマンドプロンプト」のアイコンができたことを確認してください。

デスクトップにできたコマンドプロンプトのアイコンを右クリックします。

メニューの中の「プロパティ」をクリックします。

「コマンドプロンプトのプロパティ」の「ショートカット」タブをクリックします。

「作業フォルダ」を書き換えます。

初期状態では「%HOMEDRIVE%\%HOMEPATH%」になっているはずですが、ここを「%HOMEDRIVE%\N8ZB3DOS」に書き換えてください(かならず半角で入れてください。半角なら大文字でも小文字でもよいのですが、全角では正しく実行されません)。

次に「レイアウト」タブをクリックします。

画面バッファのサイズとウィンドウのサイズを、幅(W):80、高さ(H):25に設定してください。

ウィンドウのサイズは、初期状態では幅:80、高さ:25になっていますから、変更の必要はないはずですが、確認はしておいてください。画面バッファサイズの高さは大きい値になっていますから、▼ボタンで25まで下げてください。

このとき25より小さい値まで下げると、ウィンドウのサイズの高さも連動して小さくなってしまいますが、画面バッファのサイズの高さを25まで上げても、ウィンドウのサイズは大きくなりません。それぞれ25になるように操作してください。

最後に「OK」をクリックします。

コマンドプロンプトアイコンの下に表示されている名前も変更しておきます。さきほどと同じようにコマンドプロンプトアイコンを右クリックしてメニューを開き、「名前の変更」をクリックします。名前(コマンドプロンプト)が白抜き文字で表示されるので、適当な名前に変更します。「N8ZB3DOS」でよいでしょう(こは全角でも漢字でも構いません)。

4. 4 Windows98、SE、Meの場合

「N8ZB3DOS」フォルダをCドライブ(C:)にコピーします。コピーの方法はいろいろあります。慣れている方法でコピーしてください。

操作例:「マイコンピュータ」→「Dドライブ」の順にダブルクリックして開き、「N8ZB3DOS」フォルダを右クリックしてメニューを開き、「コピー」をクリックします。次にマイコンピュータをダブルクリックして開きCドライブを右クリックしてメニューを開き、「貼り付け」をクリックします。

Cドライブに「N8ZB3DOS」フォルダができたことを確認してください。

次にDOSプロンプトをすぐに使えるようにします。

「スタート」→「プログラム」→「MSDOSプロンプト」の順にマウスで選択し、「MSDOSプロンプト」を右クリックします。

メニューが開くのでその中の「コピー」をクリックします。

次にデスクトップ(起動後のマイコンピュータなどのショートカットアイコンが並んでいる画面)のアイコンなどが無い地の部分にマウスを持って行って、そこで右クリックします。ここでもメニューが開くので「貼り付け」をクリックします。

デスクトップに「MSDOSプロンプト」のアイコンができたことを確認してください。

デスクトップにできたアイコンをマウスで右クリックします。

メニューの中の「プロパティ」をクリックします。

「MSDOSプロンプトのプロパティ」の「プログラム」タブをクリックします。「作業ディレクトリ」を書き換えます。初期状態では「C: ¥ WINDOWS」になっているはずですが、ここを「C: ¥ N8ZB3DOS」に書き換えてください(かならず半角で入れてください。半角なら大文字でも小文字でもよいのですが、全角では正しく実行されません)。書き換えたら「OK」をクリックします。

最後にMSDOSプロンプトアイコンの下に表示されている名前も変更しておきます。

さきほどと同じようにMSDOSプロンプトアイコンを右クリックしてメニューを開き、「名前の変更」をクリックします。名前(MSDOSプロンプト)が白抜き文字で表示されるので、てきとうな名前に変更します。「N8ZB3DOS」でよいでしょう(ここは全角でも漢字でも構いません)。

5. USBケーブルを使ってND8080とDOS/Vパソコンを接続する

USBケーブルを使って、ND8080基板のUSBコネクタと、DOS/VパソコンのUSBコネクタを接続し、ND8080の電源を入れます。

USB接続の方法および基本的な操作方法については、ND8080付属の「USB接続操作説明書」を参照してください。

6. N8ZB3DOS. EXEの起動

N8ZB3DOS. EXEはN8ZB3. EXEを機能拡張したものですから、N8ZB3. EXEに含まれている機能(リモート機能など)も同じように使えます。

しかしこの説明書はN8ZB3DOSで拡張された機能を中心に説明するものですから、N8ZB3. EXEに含まれている機能については、特に必要な場合以外はあらためて説明はしません。

それらについてはND8080の付属説明書を参照してください。

6.1 コマンドプロンプト(MSDOSプロンプト)を開く

「USB接続説明書」の説明にしたがってデスクトップに作成したコマンドプロンプト(N8ZB3DOS)アイコンをクリックして、コマンドプロンプトを開始します。

コマンドプロンプト画面に、

```
C: ¥ N8ZB3DOS >
```

と表示されていることを確認してください。

6.2 [NumLock]の確認

パソコンのフルキーボードの右側部分にある数字キーを有効にするために、NumLockランプがついていることを確認してください。

NumLockが有効になっていると、キーボード右側の数字キーからも数字の入力ができます。

もし数字の入力ができない状態でしたら、[NumLock]キーを押してください。

NumLockランプが点灯し、キーボード右側の数字キーからも入力ができるようになります。

6.3 キー入力モードの確認

パソコンのキーボードの入力モードが「直接入力」または「半角英数」になっていることを確認してください(「直接入力」を推奨)。

キーボード右上のCapsLock表示が消灯していることを確認してください。

もしCapsLockが点灯していたら、[Shift]キーを押しながら[Caps Lock]キーを押して、CapsLock表示を消灯させてください。

6.4 ND8080とパソコンをUSBケーブルで接続します

接続については「USB接続説明書」を参照してください。

6.5 ND8080の電源を入れる

ND8080をND80Zモニタモードで立ち上げます(電源を入れます)。

[注意1]Windowsパソコンが起動するよりも前にND8080の電源が入っているとUSB接続が確立できないことがあります。

Windowsの起動が完了したあとでND8080の電源を入れるようにしてください。

[注意2]TK80モニタモードではリモート機能を使うことはできません。

ND8080のディップスイッチが下図の通りになっていることを確認してください。

ディップスイッチを切り換えたあとはND8080をリセットしてください。



6. 6 DOSプロンプト(コマンドプロンプト)でN8ZB3DOS. exeを実行する

コマンドプロンプト画面で、N8ZB3DOS[Enter]と入力すると、下記のように表示されます。

```
C:\N8ZB3DOS>N8ZB3DOS[Enter]
*** ND8080 remote keyin ***
logfile nd8080log¥mmdhmm.txt open
```

ND8080に接続しました

このとき、

ND80Z3が接続されていないか電源が入っていません

と表示されたら、USB接続がうまく行われていません。

ND8080のリセットキーを押してから、もう一度、N8ZB3DOS. exeを実行してみてください。

6. 7 ND8080のリモート機能をONにします [* (I/O)][8 VF]

ND8080で、[* (I/O)]キーに続けて[8 VF]キーを押します。
すると、コマンドプロンプト画面に下のように表示されます。

```
ND8080に接続しました
0001 0000 -
```

【注記】上の表示はND8080をリセットまたは電源ON直後の状態から、リモート機能をONにしたときの表示です。

リモートONにする前のND8080の7セグメントLEDに表示されていた、アドレス表示部の値+1が、7セグメントLEDと、コマンドプロンプト画面の両方に表示されます。

7. ZB3BASICの起動

N8ZB3DOSのCP/M互換機能はZB3BASICを起動したあとで/CPMコマンドを入力することで使うことができます。
ようになります。

そのため、まずはZB3BASICを起動して、それからCP/M互換モードにエントリし、CP/M互換モードを終了したときも、ZB3BASICに戻るようにします。

リモート接続プログラムからZB3BASICを起動するには、[z]キーを入力します。

```
logfile nd8080log¥05021137.txt open
```

```
ND8080に接続しました
0001 0000 - z            [z]キーを入力します
```

すると画面がクリアされて、下の表示になります。

```
*** nd8080 zb3basic ****
>
```

これ以後は、BASICのコマンドとマシン語モニタコマンドが入力できます。

8.1 ログファイルについて

N8ZB3DOS. EXEの実行を開始すると、実行中のキー操作や画面表示を記録するためのログファイルがOPENされます。

ログファイルはN8ZB3DOS. EXEの実行が開始されたときの月日時分を8桁の数値で表現したファイル名で拡張子はTXTです。

N8ZB3DOS. EXEがあるフォルダN8ZB3DOSに新しいフォルダND8080LOGが作られてそこに保存されます。

TXTファイルですから、NOTEPAD(メモ帳)などで開くことができます。

ログファイルはN8ZB3DOS. EXEが正しく終了したときに、保存されます。

8.2 ログファイルのクローズ

ログファイルはN8ZB3DOS. EXEを正しく終了したときにクローズされますが、N8ZB3DOS. EXEを実行中に、それまでのログファイルを参照したい場合があります。

画面に今まで表示されたものをもう一度確認するために「ページモード」を利用することができます。

「ページモード」については「ZB3BASIC操作説明書」を参照してください。

もうひとつの方法がログファイルクローズコマンド/CLOSEを使って、それまでのログファイルをクローズする方法です。

／CLOSE[Enter]

と入力すると、それまでのログファイルがクローズされて、その時点から新しいログファイルがオープンされます。

[注記]ある程度の内容がログファイルに入っていない状態(数画面分以下)で／CLOSEを実行すると、空のログファイルが作成されることがあります。

9 N8ZB3DOSの終了

N8ZB3DOS. EXEを正しく終了するには[Ctrl]キーを押しながら[E]キーを押します。

または終了コマンド/EXITを使って

／EXIT[Enter]

と入力します。

強制終了のために使われる[Ctrl]+[C]で終了すると、ログファイルは破棄されるか一部しか保存されません。

10 N8ZB3DOSの強制終了

何らかの理由でプログラムが応答しなくなり、[Ctrl]+[C]も利かなくなったようなときは、先にND8080をリセットしてください。

そのあと[Ctrl]+[C]を入力するとエラーブレイクします。

[Ctrl]+[C]で終了すると、ログファイルは破棄されるか一部しか保存されません。

II N8ZB3DOSで拡張された機能、コマンド

N8ZB3DOSはN8ZB3. EXEにCP/M互換機能を付加したものですが、CP/M互換機能のほかにも拡張された機能があります。

ここではそれらについて説明をします。

[注記]以下のコマンドは見易さを考えて英大文字で表記していますが、小文字の入力でも実行できます。

1. LLIST

LISTコマンドで画面に表示されるBASICのプログラムリストと同じものが、ND8080のI/O入出力コネクタに接続したプリンタに出力されます。

ファンクション05は最近のWindowsプリンタでは働きません。

セントロニクスインターフェイスプリンタが必要です。

詳しくは下記のホームページ記事を参照してください。

[注記]以下の説明文中で「第xx回」という表示で紹介しているのは、当社ホームページの「ワンボードマイコンでCP/Mを！」の[第xx回]です。

[参考記事]第242回～第256回

2. LPRINT

BASICプログラムのPRINT命令で画面に表示する代わりに、ND8080のI/O入出力コネクタに接続したプリンタに出力します。

ファンクション05は最近のWindowsプリンタでは働きません。

セントロニクスインターフェイスプリンタが必要です。

詳しくは下記のホームページ記事を参照してください。

[参考記事]第242回～第256回

3. DM@

DMコマンドはZB3BASICにある機能ですが、その機能を拡張しました。

ZB3BASICのもとではアドレス0000～7FFFはND8080モニターROMが選択されていて、そのアドレスに増設されたRAMには通常はアクセスすることはできません。

DM@コマンドを使うと、裏に置かれているRAMの内容を表示させることができます。

DM@0000, 00FF[Enter]

のように入力します。

RAMの内容が表示されたあとは、0000～7FFFはもと通りモニターROMが選択されます。

DM@コマンドの用法はDMコマンドと同じです。

「ZB3BASIC操作説明書10章」を参照してください。

4. CM@

DM@コマンドと同様の機能です。

CM@コマンドを使うと、裏に置かれているRAMの内容を書き換えることができます。

CM@1000[Enter]

のように入力します。

CM@コマンドを終了したあとは、0000～7FFFはもと通りモニターROMが選択されます。

CM@コマンドの用法はCMコマンドと同じです。

「ZB3BASIC操作説明書10章」を参照してください。

5. BP@

DM@コマンドやCM@コマンドと同様の機能です。

このコマンドはZB3BASICのモードでは普通は使うことはありませんが、CP/M互換モードでのマシン語のデバッグには必須の機能です。

通常はモニタROMが選択されている0000～7FFFの範囲の増設RAM上に書かれたマシン語プログラムにブレークポイントを設定します。

BP@0100[Enter]

のように入力します(この例では増設RAMのアドレス0100にブレークポイントが設定されます)。

BP@の使い方については[第392回]を参考にしてください。

BP@の機能については増設RAM上にブレークポイントが設定されることを除いては、BPコマンドと同じです。

BPコマンドについては「ZB3BASIC操作説明書10章」を参照してください。

BP@コマンドを終了したあとは、0000～7FFFはもと通りモニタROMが選択されます。

6. RT@

BP@コマンドと対で使います。

このコマンドはZB3BASICのモードでは普通は使うことはありませんが、CP/M互換モードでのマシン語のデバッグには必須の機能です。

BP@の設定によってブレークしても、その後はアドレス0000～7FFFはモニタROMが選択された状態になります。

増設RAMに設定したブレークポイントによってブレークしたポイントからプログラムの実行を再開するためには、増設RAMが選択された状態に戻す必要があります。

RT@コマンドを使うと、増設RAMを有効にしたうえでブレークポイントに戻ります。

[注記]BPコマンドによって、RAMアドレスの8000～のポイントでブレークした場合でも、その後の実行では増設RAMが選択されていなければならない場合にはRT@コマンドでリターンしてください。

RT@[Enter]

と入力します。

RT@の使い方については[第392回]を参考にしてください。

RT@の機能については増設RAMを選択したうえでブレークポイントに戻ることを除いては、RTコマンドと同じです。

RTコマンドについては「ZB3BASIC操作説明書10章」を参照してください。

7. BP@0

BP 0 は、設定されているブレークポイントをクリアします。

BP@0 は、その機能を0000～7FFFのRAMに対して行ないます。

BPの設定は指定したアドレスのマシン語コードをFFに置き換えます。

ブレークが実行されると、FFが置かれていたアドレスにはもとのマシン語コードが戻されます。

しかし何らかの理由でブレークしなかった場合には、FFに置き換えられたままになってしまいます。

BP 0(およびBP@0)はそのように未処理のままになってしまったFFをもとのマシン語コードに戻します。

BPが設定されていることは、BP D で確認することができます。

8. JP@

JPコマンドはマシン語プログラムに直接ジャンプするためのコマンドです。

JP 8100[Enter]

と入力すると8100番地から始まるマシン語プログラムにジャンプします。

従来の機能では

JP 1033[Enter]

のように0000～7FFFの範囲を指定するとROM内のプログラムにジャンプしますが、そのアドレスにRAMを選択していてもRAM内のアドレスにはジャンプできませんでした。

そのような場合に、

JP@0100[Enter]

のように@をつけて入力すると、0000～7FFFのアドレスにRAMを選択して、そこに書かれたのマシン語プログラムにジャンプできます。

[注記]CP/M互換DOSのトランジェントプログラムにこの方法でジャンプするためには、さきに/CPMでCP/M互換DOSを実行し、さらに目的のトランジェントプログラムを仮想FDDからRAMの0100番地にロードしておく必要があります。

コマンド入力待ちのときに、拡張子をつけないでファイル名(カレントディスク以外なら、ドライブ名:ファイル名)を入力すると、指定したファイルが0100番地にロードされたあと、実行されます(このようにできるのは拡張子がCOMのファイルだけです)。

その実行を終了後(または中止後)、ZB3コマンドでZB3BASICに戻ったあとでなければ、そのプログラムにJP@0100でジャンプすることはできません。

9. /LD

/LDコマンドの機能を拡張しました。

アドレス0000～7FFFの範囲を指定して/LDを実行した場合には無条件で、増設RAMの指定アドレスにそのファイルの内容がロードされます。

/LDコマンドの実行後は0000～7FFFはもと通りモニターROMが選択されます。

[注記]このコマンドと次に説明する/SVコマンドに限り@記号は使いません。指定アドレスが0000～7FFFの範囲にあることを判別して、自動的に増設RAMが選択されます。

ロードした結果ロード最終アドレスが8000よりも後ろになる場合でも正しく実行されます。

10. /SV

/SVコマンドの機能を拡張しました。

上で説明した/LD機能と同様の拡張機能です。

指定したアドレス範囲の一部または全てがアドレス0000～7FFFにある場合には、無条件で、増設RAMが選択されて、その範囲を含む指定アドレスのデータがセーブされます。

/SVコマンドの実行後は0000～7FFFはもと通りモニターROMが選択されます。

[注記]このコマンドも上の/LDコマンドと同様@記号は使いません。指定アドレスの一部または全部が0000～7FFFの範囲であることを判別して、自動的に増設RAMが選択されます。

指定アドレス範囲が8000よりも後ろまでを含む場合でも正しく実行されます。

この機能拡張によって、アドレス0000～7FFFの範囲のROMの内容を直接セーブすることはできなくなりました。

11. /BAT

バッチファイルを実行します。

バッチファイルは通常のテキストファイルと同じ内容で、拡張子は自由です(.BATにする必要はありません)。

バッチ機能は、ZB3BASICを実行しているときに入力するコマンドや命令を、キーボードから入力する代わりにあらかじめ作成したテキストファイルから入力して実行させるものです。

バッチファイルはテキストエディタ(メモ帳)などを使って、普通にキーボードから入力する通りに記述します。

ただし、バッチモードでは小文字から大文字に変換する機能は働かないので、命令やコマンド、ファイル名は全て大文字で記述します。

バッチ機能はバッチファイルの終わりに来るまで終了しません。

処理を打ち切るためには[Ctrl]+[C]でZB3BASICを強制終了するしかありません。

バッチ機能はZB3BASICのときに実行した後は、CP/Mモードでもそのまま実行されます。

CP/Mモードではコマンドやファイル名を小文字で記述することができます。

/BATの使い方については[第390回]を参考にしてください。

11.1 バッチファイルの中で使える特殊記号

バッチファイルの中では以下の特殊記号が使えます。

1) ' または REM (注釈行)

行の先頭に ' または REM を書くとその行は実行されません。

2) p (コメントの表示)

行の先頭に小文字の p と1字以上の空白を置くと、その行が(pを空白に置き換えて)画面に表示されます。

3) \$\$1～\$\$5 (文字変数)

バッチファイル中に記述された \$\$1～\$\$5が、/BAT実行時にファイル名とともに入力したパラメータに置き換えて実行されます。

パラメータは , (カンマ) またはスペースで区切って入力します。

11.2 制限事項

バッチファイルの1行の長さは255字以内です。

ただし \$\$1～\$\$5がパラメータによって置き換えられた結果255字を超えるとエラーになることに注意してください。

またZB3BASICでは、もともとプログラム行の入力は80字以内、文字変数、文字定数は40字以内であることに注意してください。

パラメータの個数は5個までです。

\$\$1～\$\$5をバッチファイルの中で繰り返し使っても構いません。

パラメータの長さは12文字以内です。

11.3 データの入力

バッチファイルの実行中は、キーボードから入力される代わりにバッチファイルに書かれた文字が順に読み込まれて実行されます(バッチファイルの実行中はキーボードからの入力はできません)。

11.4 エラーメッセージ

batch line is too long 1行の長さが255文字を越えている

パラメータ数が制限を越えている パラメータが6個以上ある

パラメータの文字数が制限を越えている パラメータの文字数が12文字を越えている

バッチファイルの実行時に発生したエラーは通常実行時と同じように表示されますが、バッチ処理はそのまま続けて実行されます。

12. /CPM

CP/M互換DOSモードにエンタリします。

/CPM[Enter]

と入力するとCP/M互換DOSシステムプログラム(N8BIOS、MBDOS、MCCP)がアドレスCC00からのRAMエリアにロードされ、CP/M互換DOSが起動します。

13. /CPM, D

／CPMコマンドはシステムプログラムをロードしてからCP/M互換DOSを起動しますが、ユーザープログラムにブレークポイントを設定してデバッグを行なうときなど、システムが再ロードされてはデバッグが行なえない場合が出てきます。

／CPMコマンドの後ろに , D をつけて入力すると、システムプログラムを再ロードしないでCP/M互換DOSを起動します。

この機能を使うためには一度は／CPMコマンドが実行されている必要があります。

／CPM, Dの使い方については[第392回]を参考にしてください。

14. ZB3BASIC用STARTREK

MBASIC用STARTREKプログラム(STARTREK. BAS)とは別に、ZB3BASICで実行できるSTARTREKプログラムが付属CDROMに入れてあります。

ZB3用STARTREK(STARTREK. TXT)はマシン語サブルーチン(MSTRTRKS. BIN)とともに使います。

ZB3BASICを起動したあと、

／LD MSTRTRKS. BIN, 8004[Enter]

／LOAD STARTREK. TXT, 8100[Enter]

と入力すると使えるようになります。

RUN[Enter]

でゲームが開始されます。

[第367回]で紹介していますが、具体的な操作については[第376回]～[第389回]を参照してください。

プログラムを中止したいときは

[Ctrl]+[B]

を入力します。

[注記]CP/M互換DOSのファンクションコールのテストプログラムはZフォルダに入っていますが、MSTRTRKS. BINとSTARTREK. TXTはN8ZB3DOSフォルダに入っています。

ZB3BASICを起動したあと、上記のように／LD、／LOADコマンドでプログラムをND8080のRAMにすぐにロードできます。

Ⅲ CP/M互換DOSの基本情報

1. CP/M互換DOSのバージョン

CP/M互換DOSはCP/M2. 2(58KB版)と基本的な互換性があります。

[注記]58KB版CP/MとはCCP~BIOSの各プログラムの開始アドレスは合わせてありますが、未使用エリアには相違があります。

58KB版CP/MはメモリアドレスのE800~FFFFの6KBは未使用領域とされています。

これに対してCP/M互換DOSではアドレスE800以後の領域も作業エリアとして使用しています。

CP/Mではこの領域はシステムが使う領域として、通常は使用しないことになっています。

2. CP/M互換DOSの起動

ZB3BASICが起動している状態で

／CPM[Enter]

と入力するとCP/M互換DOSが起動します。

```
>/cpm
loading mcpm5g.bin ...1a0d(6669)bytes loaded,from CC00 to E60C
drive D .....
drive C .....
drive B .....
drive A .....
```

A>

CP/M互換DOSが起動すると、最初に仮想フロッピーディスクドライブA~Dの存在がチェックされ、存在しないドライブについては新しく作成されます。

仮想フロッピーディスクドライブはN8ZB3DOS. EXEと同じフォルダに作成されます。

A. VFD、B. VFD、C. VFD、D. VFDの4ファイル各2MBです。

またWindowsとの共通認識フォルダ(Z)の存在もチェックされ、存在しない場合には自動的に作成されます(Zフォルダについては下記 5. Zドライブ を参照してください)。

その後CP/M互換DOSプログラム本体がRAMにロードされ、CCP(コマンドエンドプロセッサ)に制御がわたされます。

画面には

A>

と表示されて、これ以後ビルトインコマンドまたはユーザプログラム(トランジェントプログラム)の実行ができるようになります。

3. 仮想フロッピーディスクの構造

仮想フロッピーディスクは A. VFD、B. VFD、C. VFD、D. VFDの4ファイル各2MBのバイナリファイルとしてN8ZB3DOS. EXEと同じフォルダに作成されます。

CP/M互換DOSから見たディスクの構造は全体が1024ブロック(トラックと同じ)で1ブロック(トラック)は16セクタで構成されます。

1セクタは128バイトです。

したがって1トラックは2KBになります。

トラックNo.は先頭から0~1023までで、各トラック内のセクタNo.は0~15になります。

詳細は[第219回]を参照してください。

このうちファイルの情報(FCB)をまとめて管理するディレクトリエリアは先頭の2ブロック32セクタです。

ユーザーのデータトラックはNo.2トラックから始まります。

[参考]

各ディスクの内容はVFDUMP. EXEで確認することができます。

4. VFDUMP. EXE

N8ZB3DOS. EXEが起動していない状態のとき、

VFDUMP[Enter]

と入力すると、

drive no. ? (input a-d):

と表示されますから、a、b、c、dのいずれかを入力して[Enter]を押します。

sector no. ? (0-16383):

と表示されるので、内容を確認したいセクターNo.を入力して[Enter]を押します。
セクターNo.は先頭を0とする10進数の連番です。

VFDUMP. EXEを終了したいときは、-1を入力するか、[Ctrl][C]を入力します。
[第167回]にVFDUMPの実行例があります。

[注意]／CPMコマンドを実行したあとはN8ZB3DOSを終了するまでVFDUMP. EXEは実行できません。

5. Zドライブ

テキストファイルやバイナリファイルをWindowsのN8ZB3DOS. EXEのあるフォルダからCP/M互換DOSの仮想フロッピーディスクドライブA～Dにコピーしたり、またその逆を行なうためのドライブ(フォルダ)です。

仮想フロッピーディスクドライブA～DにWindows上から直接アクセスすることはできませんが、Zドライブ(Zフォルダ)はWindowsの通常のフォルダですからWindowsから自由にアクセスすることができます。

逆にCP/M互換DOSからはWindowsのフォルダをアクセスすることはできませんが、このZドライブ(Zフォルダ)だけは仮想フロッピーディスクドライブA～Dと同じタイプのドライブとして扱えます。

そこでWindows上で作成したテキストファイルやバイナリファイルをCP/M互換DOSで扱いたいという場合には、まずそのファイルをZフォルダにコピーします。

その上でCP/M互換DOSを実行すると、DIRコマンドでそのファイルが見えますから、そのままそこでそのファイルを扱うか、または他のA～Dドライブにコピーします。

Zドライブについてはホームページでの説明も参照してください([第380回]～[第386回])。

[注意1] CP/M互換DOSを実行中にZドライブ(Zフォルダ)にWindows上からファイルをコピーしても、そのファイルはCP/M互換DOSからは見えません。

CP/M互換DOSを一旦終了してから、CP/M互換DOSを再実行すると見えるようになります。

[注意2] Zドライブ(Zフォルダ)にコピーするファイルはCP/M互換DOSで扱えるファイルに限ります。

ファイル名はアルファベットか数字および-(マイナス記号)8文字以内で拡張子は英数字3文字以内です。

ひとつのファイルのサイズには制限はありませんが、シーケンシャルファイルに限られます。

またZドライブ(Zフォルダ)に置くことのできるファイルは最大128個です。

[注記]／CPMコマンドの実行時にZドライブの有無がチェックされ、もし存在しない場合にはZドライブ(Zフォルダ)が新規作成されますが、すでに存在する場合にはZドライブ(Zフォルダ)の内容がチェックされ、ファイル名が長すぎるものがみつかったときや、ファイル数が128を超えている場合にはそれぞれエラーメッセージが表示されます。

6. CP/M互換DOSの構造

CP/M互換DOSは以下の3つのプログラムから構成されています。

1) MCCP. BIN (LOADアドレスCC00)

DIR、ERAなどのコマンドを実行するためのプログラムです。

CP/MのCCPと同等の互換機能にいくつかの機能を追加したものです。

CCPはConsole Command Processorの略称です。

CP/Mのコマンドはこのプログラムによって解釈され実行されます。

2) MBDOS. BIN (LOADアドレスD406)

「DOS」の名前がついていることからわかるように、CP/M互換DOSの中核になるプログラムです。

CP/MのBDOSと同等の互換機能にいくつかの機能を追加したものです。

BDOSはBasic Disk Operating Systemの略称です。

ディスクオペレーティングシステムの名前の通り、主にディスクに対するセーブ、ロードや、ファイルオープン、クローズなどの基本的なサブルーチンの集合体です。

またディスクに限らず、キーボードからの入力や画面、プリンタへの出力などの基本的な入出力関数もここにありません。

ユーザーが作る簡単なプログラムからMBASIC.COMのような大きなプログラムまで、ファイルのセーブ、ロードや入出力を行なうときはこのDOSプログラムの各種関数サブルーチンをコールすることで、その機能を実現するというのがCP/Mプログラムのいわば約束事です。

各関数はCレジスタにその関数を示す番号をセットしてアドレス0005をコールすることで呼び出すことができます。

3) N8BIOS. BIN (LOADアドレスE200)

CP/MのBIOSと同等の互換機能にいくつかの機能を追加したものです。

BIOSはBasic Input/Output Systemの略称です。

上記MBDOSプログラムは入出力やファイル操作のための各種関数プログラムの集合体ですが、実際にI/O装置や仮想フロッピーディスクドライブなどに直接アクセスする部分は全てこのN8BIOS内のサブルーチンをコールすることで行なわれています。

より基本的なアクセスルーチンの集合体です。

MBDOSプログラムよりも原始的なプログラムですから、普通はMBDOS内の関数サブルーチンをコールすることが安全ですが、プログラムによっては、より細かなアクセスのために、直接このN8BIOS内のサブルーチンをコールすることもあります。

MBDOS内の関数サブルーチンをコールするときはCレジスタに関数No.を入れてアドレス0005をコールしますが、N8BIOS内のサブルーチンをコールするときは、直接N8BIOS先頭のジャンプテーブルアドレスをコールします。

7. CP/M互換DOSのRAMアドレス

RAMエリアの0000~00FFはシステムが特別の目的としてリザーブされています。

ユーザーがそのエリアを使うこともできますが、もともとシステムに割り当てられた用途、使い方を外れて使用することはできません。

アドレス0100~CBFFは、トランジェントエリアとよばれていて、ユーザーが自由に使うことができます。

拡張子がCOMのマシン語プログラムファイルはこのトランジェントエリアの先頭アドレス0100にロードされたあと、0100番地から実行されます。

つまりユーザープログラムは常にアドレス0100が先頭アドレスで、かつ0100からスタートするようにプログラムを書かなければなりません。

8. システムブレイク(リポート)

CP/Mではコマンド入力待ちのとき(A:~D:、Z:が表示されてキー入力待ちのとき)に[Ctrl]+[C]を入力するとシステムブレイクし、アドレス0000番地のウォームブートが実行されます。

CP/M互換DOSでは当初は[Ctrl]+[B]を使うように設計していましたが、2013年7月修正版では[Ctrl]+[C]を使うように改めました。

9. 0000~00FFの内容

CP/M互換DOSはRAMアドレス0000~00FFを以下のように割り当てて使用します。

- | | |
|-----------|--|
| 0000-0002 | CP/Mシステムへのリエントリアドレス。
ここにはBIOSのウォームブートルーチンへのジャンプ命令(C303E2)が書かれている。 |
| 0003 | I/Oバイト。CP/Mが外部I/O装置をアクセスするときの物理装置と論理装置をむすびつけるための作業エリア。CP/M互換DOSでは使っていない。 |
| 0004 | ログインディスクNo. ログインしているディスクのNo.が記入される。たとえばA>と表示されているときは00、B>と表示されているときは01が書き込まれている。 |
| 0005-0007 | BDOSのファンクションコールのときにこのアドレスがコールされる。ここにはBDOSへのジャンプ命 |

0008-0037 令(C306D4)が書かれている。
 0038-003A 8080の割込みベクトル用としてリザーブされている。CP/M互換DOSでは使用していない。
 003B-005B 未使用
 005C 第1パラメータのドライブNo.。ここから007FまでがFCBエリアとしてファイルアクセスの際に使用される。
 005D-0064 プライマリファイルネームエリア(8バイト)。第1パラメータのファイル名がここに入る。
 0065-0067 同上拡張子エリア(3バイト)。第1パラメータのファイル名の拡張子部分がここに入る。
 0068 エクステントNo.。ディレクトリのファイルのエクステントNo.が入る。エクステントは[第148回]参照。
 0069-006A システムが内部で使用。
 006B このエクステントのレコード数。
 006C-007B ディスクアローケーションマップ。第2パラメータが存在するときには、第2FCBのファイルネーム以下のエリアとしても使用される(その場合のマップは下を参照)。
 007C シーケンシャルファイルアクセスのnextレコードNo.。[第49回]参照。
 007D-007F ランダムアクセスレコードNo.。[第102回]参照。
 0080-00FF DMAデータバッファ。ファイルデータをアクセスするときに1セクタ分のデータ(128バイト)がここに入れられる。

[第2パラメータが存在する場合の006C-007B]

006C 第2パラメータのドライブNo.
 006D-0074 セカンダリファイルネームエリア(8バイト)。第2パラメータのファイル名がここに入る。
 0075-0077 同上拡張子エリア(3バイト)。第2パラメータのファイル名の拡張子部分がここに入る。
 0078-007B 使用されない。

IV CP/M互換DOSのビルトインコマンドとトランザクションファイル

CP/Mではダイレクトにキーインして実行できるコマンドとして、ビルトインコマンドとトランジェントコマンドの2種類のコマンドがあります。

ビルトインコマンドとは最初からシステムプログラムに組み込まれているコマンドで、CP/M互換DOSでは以下のコマンドが使えます。

DIR、ERA、TYPE、SAVE、REN、COPY、ZB3

このうちDIR、ERA、TYPE、REN、SAVEはCP/Mとほぼ同じ機能です。

COPY、ZB3はCP/M互換DOS独自のコマンドです。

CP/Mにはオプションとしていろいろなトランジェントコマンドがあります。

トランジェントコマンドとはCP/M本体のプログラムには組み込まれていなくて、そのプログラムがセーブされているディスクから読み出して実行されるコマンドです。

拡張子がCOMのユーザープログラムも、ファイル名だけをキーインするとディスクからロードされて実行されますから、ユーザーが作成したトランジェントコマンドということができます。

[注記]CP/M互換DOSでは、特にトランジェントコマンドは用意していません。

以下ビルトインコマンドについて説明をします。

1. DIR

仮想フロッピーディスクおよびZドライブにセーブされているファイル名を表示します。

DIR[Enter]

と入力すると、そのときのカレントドライブ(A>と表示されていればAドライブがカレントドライブです。B>と表示されていればBドライブがカレントドライブです)にセーブされているファイル名が表示されます。

DIR C:[Enter]

のように後ろにドライブ名とコロン(:)をつけて入力すると、そこで指定したドライブにセーブされているファイル名が表示されます。

このコマンドを実行してもカレントドライブは変更されません。

2. ERA

ERA ファイル名. 拡張子[Enter]

と入力するとカレントドライブの指定したファイルが削除されます。

一度削除されたファイルはもとに戻すことができません。

ファイル名にはワイルドカードが使えます。

ERA C:ファイル名. 拡張子[Enter]

のように後ろにドライブ名とコロン(:)をつけて入力すると、そこで指定したドライブにセーブされているファイルが削除されます。

このコマンドを実行してもカレントドライブは変更されません。

[参考]ワイルドカード

ワイルドカードとは*や?を使ってファイル名や拡張子の一部を代用させるものです。

[例1]

ABC*. *

ABCではじまるファイル名は拡張子が何であっても選択されます。

[例2]

AB??C.X?Y

?は1文字を代用していて、そこにはどんな文字があっても選択されます。

3. TYPE

TYPE ファイル名. 拡張子[Enter]

と入力すると、指定したファイルの内容が画面に表示されます。

[注意] 指定するファイルはテキストファイルでなければなりません(拡張子がTXTである必要はありません)。

[注意] ファイル名や拡張子にワイルドカードは使えません。

TYPE C:ファイル名. 拡張子[Enter]

のようにファイル名の前にドライブ名とコロン(:)をつけて入力すると、そこで指定したドライブにセーブされているファイルの内容が画面に表示されます(ドライブ名はA:~D:およびZ:)。

このコマンドを実行してもカレントドライブは変更されません。

[注記] CP/Mのテキストファイルはファイルの終わりにEOFコード(1AH)を置く決まりになっていました。

TYPEコマンドもファイルの終わりをEOFコードで識別しています。

しかし現在のWindowsシステムではテキストファイルを作成してもファイルの終わりにEOFコードは付加されません。

もしもWindowsシステムで作成したファイルをそのままZドライブにコピーしても、そのままではファイルの終わりにEOFコードがないためにTYPEコマンドをそのファイルに対して実行するとファイルの最後にゴミまで表示されてしまいます。

そのようなことがおこらないようにCP/M互換DOSではZドライブから仮想フロッピーディスクドライブ(A~D)にファイルをコピーする時点で、機械的にファイルの終わりの余白をEOFコード(1AH)で埋めるようにしてあります。

4. SAVE

SAVE n ファイル名. 拡張子[Enter]

と入力すると、カレントドライブに、現在メモリの0100~にあるプログラム(データ)が0100~n×256バイト分が指定したファイル名でセーブされます。

SAVE n B:ファイル名. 拡張子[Enter]

のようにファイル名の前にドライブ名とコロン(:)をつけて入力すると、指定したドライブにファイルがセーブされます。

このコマンドを実行してもカレントドライブは変更されません。

5. REN

ファイル名を変更します。

REN ファイル名1. 拡張子a ファイル名2. 拡張子b[Enter]

と入力するとカレントドライブにあるファイル名1. 拡張子がファイル名2. 拡張子bに名前変更されます。

このコマンドではワイルドカード(*) (?)が使えます。

[注意] RENはカレントドライブのファイルのみ有効です。

ファイル名の前にドライブ名をつけても無視されます。

変更後のファイル名と同じファイル名がすでに存在する場合には、そのファイルに対してはこの命令は実行されません。

6. COPY

COPYはファイルをコピーする機能です。同時にREN(ファイル名変更)機能も含んでいます。

このコマンドではワイルドカード(*) (?)が使えます。

[注意] 名前変更後のファイル名と同じファイル名がコピー先ドライブにすでに存在する場合には、そのファイルに対してはこの命令は実行されません。

6.1 COPY ファイル名1. 拡張子a ファイル名2. 拡張子b

カレントディスク内で ファイル名1. 拡張子a で指定したファイルを ファイル名2. 拡張子b としてコピーします。

6. 2 COPY ファイル名1. 拡張子a x:ファイル名2. 拡張子b

カレントディスクの ファイル名1. 拡張子a で指定したファイルを 別のディスク(x:はA:~D:およびZ:)に ファイル名2. 拡張子b としてコピーします。

[注意]6. 1、6. 2でファイル名2と同じファイル名がコピー先ドライブにすでに存在する場合には、そのファイルに対してはこの命令は実行されません。

6. 3 COPY ファイル名1. 拡張子a x:

カレントディスクの ファイル名1. 拡張子a で指定したファイルを 別のディスク(x:はA:~D:およびZ:)に 同じ名前でコピーします。

6. 4 COPY x:ファイル名1. 拡張子a

別のディスク(x:はA:~D:およびZ:)にあるファイル名1. 拡張子a で指定したファイルを、カレントディスクに同じ名前でコピーします。

[注記]6. 3、6. 4でファイル名1と同じファイル名がコピー先ドライブにすでに存在する場合には、そのファイルはコピー元のファイルで上書きされます。

7. ZB3

CP/M互換モードを終了してZB3BASICに戻ります。
CP/M互換モードを終了して操作を終わりたいときには、まず

ZB3[Enter]

でCP/M互換モードを終了してZB3BASICに戻ってから、そこで/EXITで終了するようにします。
そのようにしないでいきなり[Ctrl]+[C]で強制終了するとND8080がハングアップします。
またログファイルも正しく作成保存されません。

8. チェンジドライブ

これはコマンドではありませんが、これもやはりMCCP. BINの機能ですからここで説明をします。
カレントドライブを変更したいときはそのドライブ名を示すアルファベットに続けてコロン(:)をつけて[Enter]を入力します。

A>
と表示されているときに、
A>b:[Enter]
と入力すると
B>
と表示されて、カレントドライブがBドライブになります。

9. リードオンリーディスク

次章で説明するMBDOSのファンクションの中には仮想ディスクドライブを書き込み禁止にしたり、逆にそれを解除する機能が含まれています。

それはフロッピーディスクでは必要な機能でしたが、ファイルを簡単にバックアップコピーできるWindowsシステムでは余り意味のない機能ですから、ファンクションとしてはオリジナルのままに機能しますが、COPY、REN、ERAノ各コマンドでは、リードオンリーになっているドライブに対してもそれを無視して働くようにしてあります。

誤操作によって消去、書き換えされると困るようなファイルについてはあらかじめバックアップを取っておくようにしてください。

なお、これとは逆にコピー先やRENによって変更されるファイル名がすでに存在するときには、上書きされないことに注意してください。

10. トランジェントファイルの実行

ユーザープログラムをディスクにセーブしたあと、そのプログラム名をダイレクトに入力すると、そのプログラムがディスクから0100番地にロードされたあと実行することができます。

その場合ユーザープログラムの拡張子は**COM**でなければなりません。

いわばユーザープログラムがそのままコマンドとして実行されることになります。

CP/Mの大きな特徴のひとつで、この機能はそのままMSDOSにも引き継がれています。

またそのときに次のようにパラメータをつけて入力すると、それぞれのパラメータがデフォルトのFCBエリアにセットされます。

プログラムファイル名 x:第1パラメータ y:第2パラメータ

x:, y: はドライブ名です(省略可)。

また第1パラメータ、第2パラメータは省略可能です。

パラメータを全て省略した場合にはデフォルトのFCBエリアは書き換えられません。

上記パラメータはいつもそのようにしてセットしなければならないものではありません。

プログラムによって上記FCBアドレスに任意のファイル名を書き込んだ上でファンクションをコールすることもできます(ドライブNo.はAドライブ=01、Bドライブ=02、…というように指定します。00を指定するとカレントディスクが選択されます)。

また任意のRAMエリアに上記のルールにしたがってファイル名等のデータを書き込んで、その先頭アドレスをDEレジスタにセットして、ファンクションをコールすることもできます。

V ファンクションコール

ファンクションコールはCP/Mの中核となる機能です。

キー入力や画面表示の機能、ファイルのオープン、クローズやファイルリード、ライトなどいろいろな機能が簡単に利用できます。

ファンクションコールは機能を示す番号をCレジスタに入れてアドレス0005をコールすることで使うことができます。

ZB3DOSはCP/Mのファンクションコールと同じ機能を同じ手続きで使うことができます。

以下にZB3DOSに含まれるファンクションの一覧を示します。

[注記1]説明文中「第XX回」(ワンボード)とあるのは、当社ホームページ連載記事「復活！ CP/M ワンボードマイコンでCP/Mを！」の「第XX回」です。

また「第XX回」(MYCPU80)とあるのは、当社ホームページ連載記事「MYCPU80でCP/Mを！」の「第XX回」です。

当初はND80ZⅢ(ND80Z3. 5)での実行を前提に開発したため、参考プログラム等はザイログニーモニック(Z80)で表記しました。

その後にMYCPU80用を開発したため、その時点で参考プログラムの表記をインテルニーモニック(8080)に書き改めました。

ND8080は8080CPUですから、連載記事「MYCPU80でCP/Mを！」を参照していただいたほうがわかりやすいと思います。

[注記2]CP/M互換DOSではコマンドやファイル名をキー入力するときにMCCPプログラムによって英小文字→英大文字の変換が行なわれます。そのため通常の入力では小文字のまま構いませんが、小文字は印刷書体としては見難いところがありますので、当説明書では特に必要のない限り大文字で説明をしています。

[注記3]ホームページで紹介しているテストプログラムのうちおもなものはZドライブ(Zフォルダ)に入れてあります。

そのうちの多くのプログラムはZドライブのままでも実行できますが、中にはA～Dドライブでなければ正しく実行できないものもあるかも知れません。

Zドライブの本来の目的はWindowsとCP/M互換DOSとの間でのファイルのやり取りをスムーズに行なうためのもので、その上でプログラムを実行することを考慮して設計を行なったものではありません。

基本的な使い方としては、まずCOPYコマンドでZドライブからA～Dドライブにプログラムをコピーしてから使うようにしてください。

ファンクション No.(Cレジスタにセ ットする値)	機能	パラメータセット DEまたはEに値をセットする	結果
0 00(16進)	システムリセット	なし	システムがリセットされる
1 01	コンソール入力	なし	A=文字コード
2 02	コンソール出力	E=文字コード	文字が表示される
3 03	リーダー(RDR)入力	なし	A=文字コード
4 04	パンチ(PUN)出力	E=文字コード	PUN:に出力される
5 05	リスト(LST)出力	E=文字コード	LST:(プリンタ)に出力される
6 06	コンソール入出力	入力:E=FF 出力:E=文字 コード	入力:A=文字コード 出力:文字 が表示される
7 07	IOバイトの取出し	なし	A=IOバイト
8 08	IOバイトセット	E=IOバイト	IOバイトがセットされる
9 09	文字列出力	DE=文字列アドレス	文字列が表示される
10 0A	コンソールバッファ入力	DE=バッファアドレス	バッファに入力される
11 0B	コンソールステータスチェ ック	なし	入力あり:A=01 入力なし:A=0 0
12 0C	バージョンNo.の取出し	なし	H=00 L=バージョンNo.
13 0D	ディスクリセット	なし	すべてのディスクがリセットされる
14 0E	ディスクドライブセレクト	E=ドライブNo.	デフォルトディスクに指定される
15 0F	ファイルオープン	DE=FCBアドレス	正常:A=ディレクトリコード 異常: A=FF

ファンクション No.(Cレジスタにセ ットする値)	機能	パラメータセット DEまたはEに値をセットする	結果
16 10	ファイルクローズ	DE=FCBアドレス	正常:A=ディレクトリコード 異常: A=FF
17 11	最初のファイルサーチ	DE=FCBアドレス	正常:A=ディレクトリコード 異常: A=FF
18 12	次のファイルサーチ	なし	正常:A=ディレクトリコード 異常: A=FF
19 13	ファイル削除	DE=FCBアドレス	正常:A=ディレクトリコード 異常: A=FF
20 14	シーンシャルリード	DE=FCBアドレス	正常:A=00 異常:A=エラーコ ード
21 15	シーケンシャルライト	DE=FCBアドレス	正常:A=00 異常:A=エラーコ ード
22 16	新規ファイル作成	DE=FCBアドレス	正常:A=00 異常:A=エラーコ ード
23 17	ファイル名の変更	DE=FCBアドレス	正常:A=00 異常:A=FF
24 18	ログインベクトルの取出し	なし	HL=ログインベクトル
25 19	ログインディスクNo.の取出 し	なし	A=ディスクNo.
26 1A	DMAアドレスのセット	DE=DMAアドレス	DMAアドレスがセットされる
27 1B	アローケーションアドレス の取出し	なし	HL=アローケーションベクトルアド レス
28 1C	ライトプロテクトセット	なし	ログインディスクがR/O(リードオ ンリー)になる
29 1D	R/Oベクトルの取出し	なし	HL=I/Oベクトル
30 1E	ファイルアトリビュートセッ ト	DE=FCBアドレス	正常:A=ディレクトリコード 異常: A=FF
31 1F	ディスクパラメータアドレス のセット	なし	HL=ディスクパラメータアドレス
32 20	ユーザーコードのセット	ZB3DOSにはありません	
33 21	ランダムリード	DE=FCBアドレス	正常:A=00 エラー:A=エラーコ ード
34 22	ランダムライト	DE=FCBアドレス	正常:A=00 エラー:A=エラーコ ード
35 23	ファイルサイズの計算	DE=FCBアドレス	FCB末尾3バイトにファイルサイズ が入る
36 24	ランダムレコードアクセス ポインタの更新	DE=FCBアドレス	FCB末尾3バイトにレコードNo.が入 る
37 25	ディスクドライブのリセット	DE=ドライブベクトル	ベクトルで指定したドライブがリセッ トされる
38 26	未定義		
39 27	未定義		
40 28	ゼロファイルを伴うランダ ムライト	DE=FCBアドレス	正常:A=00 エラー:A=エラーコ ード

0. ファンクション00 システムリセット

```
MVI C, 00  
CALL $0005
```

システムがリセットされます。
MCCP. BINとMBDOS. BINがメモリに再ロードされます。
ディスクドライブの書き込み禁止が解除されます。
見た目には何も変わりません。
普通のプログラムではこの機能は使うことがありません。

[参考記事]第391回(ワンボード)、第100回(MYCPU80)、第102回(MYCPU80)

[サンプルプログラム]MVFTS0-2

* MVFTS0-2. COMの使用例については第102回(MYCPU80)を参照してください。

1. ファンクション01 コンソール入力

```
MVI C, 01  
CALL $0005
```

キーボードから1文字がAレジスタに入力されます。
キーボードから入力された文字は画面にそのまま表示されます。
[BackSpace][Enter][Tab]以外の特殊機能キーも入力できますが機能しません。
キーが押されるまでリターンしません。
[Enter]キーまたは[Ctrl]+[M]の入力でカーソルが行の先頭に復帰します。このときAレジスタには0Dが入ります。
[Ctrl]+[J]の入力でカーソルが行の先頭に復帰するとともに1行改行されます。このときAレジスタには0Aが入ります。
[TAB]キーまたは[Ctrl]+[I]の入力でカーソルがタブ位置に移動します。このときAレジスタには09が入ります。
[BackSpace]キーまたは[Ctrl]+[H]の入力でカーソルが1字前に戻ります(文字は消去されません)。このときAレジスタには08が入ります。
このファンクションは[Ctrl]+[C]を入力してもブレイクしません。

[参考記事]第39回(ワンボード)、第122回(ワンボード)、第197回(ワンボード)、第391回(ワンボード)、第102回(MYCPU80)

[サンプルプログラム]MVFTST1、MVFTS0-2

* MVFTST1. COM、MVFTS0-2. COMの使用例については第99回(MYCPU80)、第100回(MYCPU80)を参照してください。

2. ファンクション02 コンソール出力

```
MVI E, XX  
MVI C, 02  
CALL $0005
```

Eレジスタにセットした文字コード(ASCIIコード)に対応する文字が画面に表示されます。
E=0Dのときカーソルが行の先頭に復帰します。
E=0Aのときカーソルが行の先頭に復帰するとともに1行改行されます。
E=09のときカーソルがタブ位置に移動します。
E=08のときカーソルが1字前に戻ります。

画面表示中にキーボードから[Ctrl]+[P]を入力すると表示が一時停止します。
表示停止中に[Ctrl]+[Q]の入力で実行が再開されます。
[Ctrl]+[P]を入力して一時停止中に[Ctrl]+[C]を入力すると、プログラムの実行が中止されてCP/M互換DOSに戻ります。
しかし表示実行中にいきなり[Ctrl]+[C]を入力するとシステムがブレイクしてZB3DOSの実行が打ち切られてコマンドプロンプトに戻ってしまうので注意が必要です(この場合にはMYCPU80がハングアップするのでMYCPU80をリセットする必要があります)。

[参考記事]第39回(ワンボード)、第122回(ワンボード)、第197回(ワンボード)、第391回(ワンボード)、第400回(ワンボード)、第102回(MYCPU80)

[サンプルプログラム]MVFTST1、MVFTS0-2

* MVFTST1. COM、MVFTS0-2. COMの使用例については第99回(MYCPU80)、第100回(MYCPU80)を参照してください。

3. ファンクション03 リーダー(RDR)入力(RS232C入力)

```
LD C, 03  
CALL $0005
```

ファンクション03はCP/Mでは紙テープリーダーからの入力機能でした。

N8ZB3DOSではRS232C入力にしました。

ND8080のRS232Cコネクタに接続した外部機器からRS232C送信が行なわれると、受信バッファから1文字がAレジスタに入ってリターンします。

[参考記事]第78回、第139回、第205回

[サンプルプログラム]FNC03T-2

* FNC03T-2. COMの使用例は[第205回]にあります。

4. ファンクション04パンチ(PUN)出力(RS232C出力)

```
MVI E, XX  
MVI C, 04  
CALL $0005
```

ファンクション04はCP/Mでは紙テープパンチャーへ出力機能でした。

N8ZB3DOSではRS232C出力にしました。

Eレジスタにセットされた文字コードが、ND8080のRS232Cコネクタに接続した外部機器にRS232Cで送信されません。

[参考記事]第78回、第139回、第205回

[サンプルプログラム]FNC03T-2

* FNC03T-2. COMの使用例は[第205回]にあります。

5. ファンクション05 リスト(LST)出力

```
MVI E, XX  
MVI C, 05  
CALL $0005
```

Eレジスタにセットされた文字コードが、ND8080のI/O入出力コネクタに接続したプリンタに出力されます。

ファンクション05は最近のWindowsプリンタでは働きません。

セントロニクスインターフェイスプリンタが必要です。

詳しくは下記のホームページ記事を参照してください。

[参考記事]第242回～第256回

[サンプルプログラム]FNC05T2

* FNC05T2. COMの使用例は[第249回]にあります。

6. ファンクション06 コンソール入出力

(入力の場合)

```
MVI E, FF  
MVI C, 06  
CALL $0005
```

(出力の場合)

```
MVI E, XX(FF以外)
MVI C, 06
CALL $0005
```

ファンクション06はファンクション01とファンクション02を合わせたような機能ですが、それよりも動作はシンプルです。

EレジスタにFFを入れてコールすると、そのときキーボードが押されていていればそのキーの文字コードがAレジスタにセットされてリターンします。

キーが押されていない場合はAレジスタに00が入ってリターンします。

EレジスタにFF以外の文字コードを入れてコールすると画面にそのコードに対応する文字が表示されます。

このファンクションはファンクション02と異なり、[Ctrl]+[P]、[Ctrl]+[Q]、[Ctrl]+[C]には応答しません。

E=0D([Ctrl]+[M])のときカーソルが行の先頭に復帰します。

E=0A([Ctrl]+[J])のときカーソルが行の先頭に復帰するとともに1行改行されます。

E=08([Ctrl]+[H])のときカーソルが1字前に戻ります。

E=09([Ctrl]+[I])のときカーソルがタブ位置に移動します。

[参考記事]第79回(ワンボード)、第139回(ワンボード)、第205回(ワンボード)、第105回(MYCPU80)

[サンプルプログラム]MF06T、MF06T-2

* MF06T.COM、MF06T-2.COMの使用例は[第105回](MYCPU80)にあります。

7. ファンクション07 IOバイト取り出し

I/OバイトはI/O装置の物理デバイスと論理デバイスを結びつけるためにアドレス0003に置かれています。

ファンクション07はそのI/Oバイトの値を取り出す機能です。

I/OバイトはZB3DOSでは使っていないから、このファンクションを使う意味はありません。

```
MVI C, 07
CALL $0005
```

I/Oバイト(アドレス0003)の値(常に00)がAレジスタに入力されます。

[参考記事]第62回(ワンボード)、第137回(ワンボード)、第203回(ワンボード)、第106回(MYCPU80)

[サンプルプログラム]MFTST7

* MFTST7.COMの使用例は[第106回](MYCPU80)にあります。

8. ファンクション08 IOバイトセット

I/OバイトはI/O装置の物理デバイスと論理デバイスを結びつけるためにアドレス0003に置かれています。

ファンクション08はそのI/Oバイトに値をセットする機能です。

I/OバイトはZB3DOSでは使っていないから、このファンクションを使う意味はありません。

```
MVI E, XX
MVI C, 08
CALL $0005
```

Eレジスタにセットした値がI/Oバイト(アドレス0003)に書き込まれます。

[参考記事]第62回(ワンボード)、第137回(ワンボード)、第203回(ワンボード)、第106回(MYCPU80)

[サンプルプログラム]MFTST8

* MFTST8.COMの使用例は[第106回](MYCPU80)にあります。

9. ファンクション09 文字列出力

ファンクション02は画面に1文字を表示しますが、ファンクション09は文字列を表示します。

```
LXI D, $XXXX (文字列のあるアドレス)
MVI C, 09
CALL $0005
```

DEレジスタに文字列の先頭アドレスをセットしてコールするとその文字列が画面に表示されます。
文字列の最後には必ず\$ (コード24H)がなければなりません。
画面表示中にキーボードから[Ctrl]+[S]を入力すると表示が一時停止します。
表示停止中に[Ctrl]+[Q]の入力で実行が再開され、[Ctrl]+[C]を入力すると、プログラムの実行が中止されま
す。

[参考記事]第41回(ワンボード)、第134回(ワンボード)、第199回(ワンボード)

[サンプルプログラム]MFTST2

* MFTST2. COMの使用例は[第106回](MYCPU80)にあります。

10. ファンクション0A コンソールバッファ入力

キーボードから入力される文字があらかじめ指定した文字数に達するか、[Enter]キーが押されるまで順次コンソールバッファ(キー入力バッファ)に蓄えます。

キー入力するごとに入力した文字が画面にエコー表示されます。

指定した文字数に達するか、[Enter]キーが押されるとそこでリターンします。

キー入力バッファの最大値は255バイトで、プログラムで使っていないアドレスならば任意の場所に置くことができます。

```
LXI D, $XXXX (キー入力バッファの先頭アドレス)
MVI C, 0A
CALL $0005
```

[キー入力バッファの構造]

(バッファをC000に置いたときの例。DE=\$C000とする)

C000 FF ここに入力を許可する文字数を入れておく(最大値はFF=255バイト)

C001 XX リターンしたときに実際に入力された文字数がここに入れられる(0D0Aは入らない)

C002 ここから後ろがキー入力コードが入る入力バッファ

|

C101 入力バッファの終わりのアドレス

ファンクション0Aの入力ではキー入力中に下記の機能が使えます。

[BackSpace]キーまたは[Ctrl]+[H]を入力すると最後に入力した文字が削除されてカーソルが1文字分前に戻ります。

[Tab]キーまたは[Ctrl]+[I]を入力するとカーソルが次のTAB位置に進みます。

[Enter]キーまたは[Ctrl]+[J]または[Ctrl]+[M]を入力するとキー入力が確定してメインルーチンにリターンします。

[Ctrl]+[R]を入力すると、入力した文字列が下に再表示されて、そこから入力が続けられます。

[Ctrl]+[X]を入力すると、入力文字がクリアされてカーソルが先頭に戻って入力スタンバイになります。

最初の1文字目の入力で/Eを入力するとプログラムを終了します(2文字目以後では受け付けられません)。

[参考記事]第64回(ワンボード)、第128回(ワンボード)、第198回(ワンボード)、第108回(MYCPU80)

[サンプルプログラム]MFTST9、MFTST92

* MFTST9. COM、MFTST92. COMの使用例は[第108回](MYCPU80)にあります。

11. ファンクション0B コンソールステータスチェック

キーボードの状態をチェックします。

```
MVI C, 0B
CALL $0005
```

キーボードの入力をチェックしてキーが押されていないときはAレジスタに00を入れてリターンします。キーが押されていたらAレジスタに00以外を入れてリターンします。

[参考記事]第68回(ワンボード)、第138回(ワンボード)、第204回(ワンボード)、第363回(ワンボード)、第109回(MYCPU80)

[サンプルプログラム]MFOBT-3

* MFOBT-3. COMの使用例は[第109回](MYCPU80)にあります。

12. ファンクション0C バージョンNo.取り出し

ファンクション0CはCP/Mのバージョンナンバーを取り出します。

ZB3DOSはCP/M2.2互換です。

このファンクションをコールするとHLレジスタに0022がセットされてリターンします。

```
MVI C, 0C  
CALL $0005
```

HLレジスタに0022が入ってリターンします。

[参考記事]第62回(ワンボード)、第137回(ワンボード)、第203回(ワンボード)、第106回(MYCPU80)

[サンプルプログラム]MFTST7

* MFTST7. COMの使用例は[第106回](MYCPU80)にあります。

13. ファンクション0D ディスクリセット

全てのドライブのライトプロテクトを解除します。

```
MVI C, 0D  
CALL $0005
```

A~Dドライブが初期状態にセットされ、ライトプロテクトが解除されます。

[注記]CP/M互換DOSではライトプロテクトは働きません。詳しくは第394回(ワンボード)を参照してください。

[参考記事]第92回(ワンボード)、第142回(ワンボード)、第208回(ワンボード)、第394回(ワンボード)、第112回(MYCPU80)

[サンプルプログラム]MFTST13

* MFTST13. COMの使用例は[第112回](MYCPU80)にあります。

14. ファンクション0E ディスクドライブセレクト

カレントドライブを変更します。

```
MVI E, XX (ドライブNo.を示す数値)  
MVI C, 0E  
CALL $0005
```

Eレジスタに入れた値(Aドライブ=00~Dドライブ=03)によってそれ以後は指定したドライブNo.のディスクドライブがカレントドライブになります。

ただしこの指定はユーザープログラムの中だけで有効でプログラムを終了するとカレントディスクに戻ります。

[参考記事]第84回(ワンボード)、第85回(ワンボード)、第140回(ワンボード)、第206回(ワンボード)、第113回(MYCPU80)

[サンプルプログラム]MFTST102

* MFTST102. COMの使用例は[第113回](MYCPU80)にあります。

ファンクション0EでEレジスタに設定するドライブNo.はAドライブ=00、Bドライブ=01…、ですがVFTST102では0を入力するとドライブNo.を変更しない、という設定にしているため、数値を入力する場合にはAドライブ=1、Bドライブ=2、…というようにしています。

MFTST102はファイル名サーチプログラムの変形ですからプログラム実行時にワイルドカード(*)を使って

```
MFTST102 *.* [Enter]
```

と入力しなければホームページ記事と同じ結果は得られません。

15. ファンクション0F ファイルオープン

ファイルを開きます。

```
LXI D, $XXXX (FCBアドレス)
MVI C, 0F
CALL $0005
```

DEレジスタにファイル名情報のあるFCBアドレスをセットしてコールすると、そのファイルが見つかった場合にはFCBのディレクトリエリアに書き込んでリターンします。

ファイルが見つかったときはAレジスタに00~03が、見つからなかったときはFFが入ってリターンします。

FCBエリアについては[第47回]で説明をしています。

またⅢ CP/M互換DOSの基本情報 9. 0000~00FFの内容 でも説明をしています。

Ⅳ CP/M互換DOSのビルトインコマンドとトランザクションファイル 11. トランザクションファイルの実行 も参照してください。

ファンクション0Fをコールする前にセットする必要があるのはFCB先頭のドライブNo.~拡張子までです。

[参考記事]第46回(ワンボード)~第49回(ワンボード)、第134回(ワンボード)、第200回(ワンボード)、第115回(MYCPU80)

[サンプルプログラム]MFTST4

* MFTST4. COMの使用例は[第115回](MYCPU80)にあります。

16. ファンクション10 ファイルクローズ

```
LXI D, $XXXX (FCBアドレス)
MVI C, 10
CALL $0005
```

ファンクション0Fやファンクション16でファイルを開いて、シーケンシャルライト(ファンクション15)でそこに書き込みを行っても、まだ仮に書き込みが行なわれただけで、それだけでは実際にファイルの更新や新規作成は行なわれません。

最後にファイルをクローズしたときに初めて更新情報がディスクに書き込まれて、データの更新が確定されます。

なおファイルクローズはシーケンシャルライト(ファンクション15)やランダムライト(ファンクション22、28)を完結させるために必要な作業で、シーケンシャルリードやランダムリードでただ読み出しを行なうためだけにオープンした場合にはクローズする必要はありません。

ファイルオープン(ファンクション0F、ファンクション16)で指定したのと同じFCBアドレスをDEレジスタにセットしてコールすることで、更新されたFCBのディレクトリ情報がディスクに書き込まれます。

ファイルクローズが正しく行なわれた場合には、Aレジスタには00~03のいずれかの値が入ります。

ファイルクローズに失敗した場合にはAレジスタにFFが入ります。

FCBエリアについては[第47回](ワンボード)で説明をしています。

[参考記事]第51回(ワンボード)、第135回(ワンボード)、第201回(ワンボード)、第116回(MYCPU80)

[サンプルプログラム]MFTST5

* MFTST5. COMの使用例は[第116回](MYCPU80)にあります。

17. ファンクション11 最初のファイルサーチ

最初のファイル名をサーチします。

```
LXI D, $XXXX (FCBアドレス)
MVI C, 11
CALL $0005
```

DEレジスタにファイル名情報のあるFCBアドレスをセットしてコールすると、そのファイルが見つかった場合には、そのファイル名情報が含まれるディスクのディレクトリエリア(128バイト)がDMAバッファにコピーされます。

同時にAレジスタには00~03の値が入られます。

ファイル名が見つからなかった場合にはAレジスタにはFFが入れられます。

FCBのディレクトリエリアに書き込んでリターンします。

ファイルが見つかったときはAレジスタに00~03が、見つからなかったときはFFが入ってリターンします。

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 9. 0000~00FFの内容 でも説明をしています。

Ⅳ CP/M互換DOSのビルトインコマンドとトランザクションファイル 11. トランザクションファイルの実行 も参照してください。

ファンクション11をコールする前にセットする必要があるのはFCB先頭のドライブNo.~拡張子までです。

ファイルが見つかった場合にAレジスタに入れられる00~03は、そのファイル名情報が、DMAバッファにコピーされたディレクトリエリア(128バイト)の何番目にあるかを示しています。

128バイトのディレクトリエリア(1セクタ分)は32バイトずつ4つに区切られていて、4つのファイル名情報が入れられます。

[参考記事]第65回(ワンボード)、第138回(ワンボード)、第204回(ワンボード)、第117回(MYCPU80)

[サンプルプログラム]MFTST10

* MFTST10. COMの使用例は[第117回](MYCPU80)にあります。

18. ファンクション12 次のファイルサーチ

次のファイル名をサーチします。

```
MVI C, 12
CALL $0005
```

ファンクション12はファンクション11または、ファンクション12を実行して指定したファイル名が見つかったときのディレクトリエリアのそのときの位置からスタートして、次の同じファイル名を探します。

そのこと以外はファンクション11と同じです。

ファンクション12はそれ以前にファンクション11が実行されていることが前提になります。

ですからFCBアドレスの指定もファンクション11で行なったものがそのまま引き継がれますから、DEレジスタにFCBアドレスの指定は行ないません。

[参考記事]第65回(ワンボード)、第138回(ワンボード)、第204回(ワンボード)、第117回(MYCPU80)

[サンプルプログラム]MFTST10

* MFTST10. COMの使用例は[第117回](MYCPU80)にあります。

19. ファンクション13 ファイル削除

ファイルを削除します。

```
LXI D, $XXXX (FCBアドレス)
MVI C, 13
CALL $0005
```

DEレジスタにファイル名情報のあるFCBアドレスをセットしてコールすると、そのファイルが見つかった場合にはFCBのディレクトリエリアに書き込んでリターンします。

ファイルが見つかったときはAレジスタに00~03が、見つからなかったときはFFが入ってリターンします。

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 9. 0000~00FFの内容 でも説明をしています。

Ⅳ CP/M互換DOSのビルトインコマンドとトランザクションファイル 11. トランザクションファイルの実行 も参照してください。

ファンクション13をコールする前にセットする必要があるのはFCB先頭のドライブNo.~拡張子までです。

[参考記事]第51回(ワンボード)、第135回(ワンボード)、第201回(ワンボード)、第116回(MYCPU80)

[サンプルプログラム]MFTST5

* MFTST5. COMの使用例は[第116回](MYCPU80)にあります。

20. ファンクション14 シーンシャルリード

```
LXI D, $XXXX (FCBアドレス)
MVI C, 14
CALL $0005
```

ファイルオープン(ファンクション0F)で指定したのと同じFCBアドレスをDEレジスタにセットしてコールすることで、ディスクから1レコード(128バイト)をDMAバッファに読み込みます。

最初のコールのときにはFCBエリアの33バイト目に00をセットします。

FCBエリアの33バイト目はシーケンシャルリードのときに次に読み込むレコードNo.をセットします。

1度このファンクションがコールされると、FCBエリアの33バイト目の値は+1されます。

レコードデータが正常に読み込まれた場合にはAレジスタに00が入ります。

エンドオブファイルの場合にはAレジスタには00以外の値がセットされます。

FCBエリアについては[第47回](ワンボード)で説明をしています。

FCBエリアの33バイト目の値については[第49回](ワンボード)で説明をしています。

[参考記事]第46回(ワンボード)～第49回(ワンボード)、第134回(ワンボード)、第200回(ワンボード)、第115回(MYCPU80)

[サンプルプログラム]MFTST4

* MFTST4. COMの使用例は[第115回](MYCPU80)にあります。

21. ファンクション15 シーケンシャルライト

```
LXI D, $XXXX (FCBアドレス)
MVI C, 15
CALL $0005
```

ファイルオープン(ファンクション0F、ファンクション16)で指定したのと同じFCBアドレスをDEレジスタにセットしてコールすることで、DMAバッファの1レコード分のデータ(128バイト)がディスクに書き込まれます。

新規作成をするファイルで、このファンクションを最初にコールするときにはFCBエリアの33バイト目に00をセットします。

FCBエリアの33バイト目はシーケンシャルリードライトのときに次に書き込むレコードNo.をセットします。

1度このファンクションがコールされると、FCBエリアの33バイト目の値は+1されます。

すでに存在するファイルをファンクション0Fでオープンしてこのファンクションを使うと、FCBエリアの33バイト目で指定したレコードが書き換えられます(この場合にはクローズしなくてもディスクの内容は書き換えられてしまうことに注意してください)。

レコードデータが正常に書き込まれた場合にはAレジスタに00が入ります。

書き込まれなかった場合にはAレジスタには00以外の値がセットされます。

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 9. 0000～00FFの内容 でも説明をしています。

FCBエリアの33バイト目の値については[第49回](ワンボード)で説明をしています。

[参考記事]第51回(ワンボード)、第135回(ワンボード)、第201回(ワンボード)、第116回(MYCPU80)

[サンプルプログラム]MFTST5

* MFTST5. COMの使用例は[第116回](MYCPU80)にあります。

22. ファンクション16 新規ファイル作成(新規ファイルのオープン)

新規ファイルをオープンします。

```
LXI D, $XXXX (FCBアドレス)
MVI C, 16
CALL $0005
```

DEレジスタにファイル名情報のあるFCBアドレスをセットしてコールすると、そのファイル名情報がシステム内のFCBエリアに作成されます(まだディスクには書き込まれません)。

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 9. 0000～00FFの内容 でも説明をしています。

Ⅳ CP/M互換DOSのビルトインコマンドとランザクションファイル 11. ランザクションファイルの実行 も参照し

てください。

ファンクション16をコールする前にセットする必要があるのはFCB先頭のドライブNo.～拡張子までです。

指定したディスクがディスクフルかどうかを確認して、ディスクにまだ作成する余地がある場合にはAレジスタに00～03が、ディスクフルのときはFFが入ってリターンします。

ファンクション16は指定したファイル名がすでにディスクに存在するかどうかは確認しません。

すでに存在するファイル名を指定して、このファンクションでファイルを新規オープンして、それをファンクション10でクローズすると同一ディスク上に同じファイル名のファイルが複数作成されてしまいます。

そのようなことを避けるためにはこのファンクションを実行する前にファイル削除を行なうかファイル名が存在するかどうかをサーチして確認しておく必要があります。

[参考記事]第51回(ワンボード)、第135回(ワンボード)、第201回(ワンボード)、第116回(MYCPU80)

[サンプルプログラム]MFTST5

* MFTST5. COMの使用例は[第116回](MYCPU80)にあります。

23. ファンクション17 ファイル名の変更

ファイル名を変更します。

```
LXI D, $XXXX (FCBアドレス)
MVI C, 17
CALL $0005
```

DEレジスタにファイル名情報のあるFCBアドレスをセットしてコールすると、FCBの第1ファイルネームエリアにあるファイル名でサーチをして、そのファイル名が見つかった場合には、FCBの第2ファイルネームエリアの名前で置き換えます。

ファイル名が見つかった場合にはAレジスタに00～03のいずれかが入ってリターンします。

みつからなかった場合にはAレジスタにFFが入ってリターンします。

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 9. 0000～00FFの内容 でも説明をしています。

Ⅳ CP/M互換DOSのビルトインコマンドとランザクションファイル 11. ランザクションファイルの実行 も参照してください。

ファンクション17では第2パラメータにつけたドライブ名は利用されません(無視されます)。

ファンクション17は変更後のファイル名がすでにディスクに存在するかどうかは確認しません。

すでに存在するファイル名を第2パラメータに指定して、このファンクションをコールすると同一ディスク上に同じファイル名のファイルが複数作成されてしまいます。

そのようなことを避けるためにはこのファンクションを実行する前にファイル削除を行なうかファイル名が存在するかどうかをサーチして確認しておく必要があります。

[参考記事]第93回(ワンボード)、第144回(ワンボード)、第213回(ワンボード)、[第122回](MYCPU80)

[サンプルプログラム]MFTST14

* MFTST14. COMの使用例は[第122回](MYCPU80)にあります。

24. ファンクション18 ログインベクトルの取り出し

ログインベクトルの値(16ビット)を取り出します。

```
MVI C, 18
CALL $0005
```

Hレジスタにログインベクトルの値が入ります。

ログインベクトルはログインディスクの履歴のようなものです。

ログインディスクはカレントドライブのことです。

CP/M2. 2では最大16台のディスクを接続することができます。

ディスクドライブはA～Pの名前をつけて識別します。

そしてそのAドライブ～Pドライブをビット0～ビット15までの各ビットに順に割り当てた16ビットの値がログインベクトルです。

あるドライブに1回でもアクセスすると、ログインベクトルの、そのドライブに対応するビットが1になります。

bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

[注記]CP/M互換DOSでは仮想ディスクドライブA～Dの4ドライブしか実装していません。またZドライブにはログインベクトルはありません。

[参考記事]第90回(ワンボード)、第141回(ワンボード)、第142回(ワンボード)、第207回(ワンボード)、第209回(ワンボード)、第118回(MYCPU80)

[サンプルプログラム]MFTST105

* MFTST105.COMの使用例は[第118回](MYCPU80)にあります。

25. ファンクション19 ログインディスクNo.の取り出し

現在のカレントドライブのNo.を取り出します。

```
MVI C, 19  
CALL $0005
```

ALレジスタにカレントドライブのNo.(A=01、B=02、…)が入ります。
ログインディスクはカレントドライブのことです。

[参考記事]第90回(MYCPU80)、第141回(MYCPU80)、第207回(MYCPU80)、第118回(MYCPU80)

[サンプルプログラム]MFTST105

* MFTST105.COMの使用例は[第118回](MYCPU80)にあります。

26. ファンクション1A DMAアドレスのセット

DMAアドレスを指定します。

```
LXI D, $XXXX (DMAアドレス)  
MVI C, 17  
CALL $0005
```

[参考記事]第56回(ワンボード)、第136回(ワンボード)、第202回(ワンボード)、第120回(MYCPU80)

[サンプルプログラム]MFTST6-3

* MFTST6-3.COMの使用例は[第120回](MYCPU80)にあります。

27. ファンクション1B アローケーションアドレスの取り出し

カレントドライブのアローケーションベクトル(ALV)の先頭アドレスを取得します(HLレジスタに入ります)。

```
MVI C, 1B  
CALL $0005
```

アローケーションベクトル(ALV)はディスク毎にメモリ上に確保されたエリアで1ビットが1ブロックに対応しています。
各ディスクは1024ブロックで構成されています。

1ビットが1ブロックに対応しますから、アローケーションベクトル(ALV)のエリアは1024/8=128バイト必要です。
このシステムではそのエリアは次のように割り当てられています。

```
EB00-EB7F Aドライブ  
EB80-EBFF Bドライブ  
EC00-EC7F Cドライブ  
EC80-ECFF Dドライブ
```

このアドレスはBIOSが管理しているエリアですから、システムによって異なっている可能性があります。

そこでこのファンクション1Bを使って、その先頭アドレスを取得するのです。

各ALVエリアはアドレスの若い方から順に、最初のバイトがブロック0-7、次のバイトがブロック8-15、…というように割り当てられています。

1バイトの中はビット7が若いブロックNo.で、ビット6がその次のNo.、…というように割り当てられています。

ちなみにこのシステムではファイル名とそのファイルのデータブロックを管理しているディレクトリエリアは各ディスクの先頭のブロックNo.0とブロックNo.1になっています。

したがってフォーマット直後のディスクのALVの先頭バイトの値はC0(11000000)になります。

[参考記事]第95回(ワンボード)、第143回(ワンボード)、第210回(ワンボード)、[第121回](MYCPU80)

[サンプルプログラム]MFTST15

* MFTST15. COMの使用例は[第121回](MYCPU80)にあります。

28. ファンクション1C ライトプロテクトセット

カレントドライブにライトプロテクトをかけてリードオンリーにします。

```
MVI C, 1C  
CALL $0005
```

[注記]CP/M互換DOSではライトプロテクトは働きません。詳しくは第394回(ワンボード)を参照してください。

[参考記事]第91回(ワンボード)、第141回(ワンボード)、第207回(ワンボード)、第394回(ワンボード)、第112回(MYCPU80)

[サンプルプログラム]MFTST11

* MFTST11. COMの使用例は[第112回](MYCPU80)にあります。

ライトプロテクトはMFTST13(ファンクション0D)の実行により解除されます。

29. ファンクション1D R/Oベクトルの取り出し

R/Oベクトル(リードオンリーベクトル)の値(16ビット)を取り出します。

```
MVI C, 1D  
CALL $0005
```

HLレジスタにR/Oベクトル(リードオンリーベクトル)の値が入ります。

R/Oベクトルはディスクドライブの状態(ライトプロテクトがかかっているかどうか)をドライブにつき1ビットを下のように割り当てたもので、リードオンリーになっているドライブのビットには1が入れられます。

bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

[注記1]CP/M互換DOSではライトプロテクトは働きません。詳しくは第394回(ワンボード)を参照してください。

[注記2]CP/M互換DOSでは仮想ディスクドライブA~Dの4ドライブしか実装していません。またZドライブにはROベクトルはありません。

[参考記事]第91回(ワンボード)、第141回(ワンボード)、第207回(ワンボード)、第394回(ワンボード)、第112回(MYCPU80)

[サンプルプログラム]MFTST12

* MFTST12. COMの使用例は[第112回](MYCPU80)にあります。

30. ファンクション1E ファイルアトリビュートセット

ファイルのアトリビュート(属性)を変更します。

```
LXI D, $XXXX (FCBアドレス)  
MVI C, 1E  
CALL $0005
```

ファイルの属性とは、CP/Mの場合、そのファイルがR/O(Read Only、更新削除禁止)かR/W(読み書き可能)か、またSYS(システムファイル)かDIR(それ以外のファイル)かという区別をファイルに付与することです。

通常の操作でファイルを作成したときは、そのファイルの属性はR/WかつDIRになります。

その後ファンクションコール1EHによって、それをR/OやSYSに変更したり、またはR/OやSYSをR/WやDIRに戻したりすることができます。

ファイルの属性(属性)はFCBエリアの拡張子(3バイト)のうちの第1バイトと第2バイトのビット7によって決定されます([第97回](ワンボード)で説明をしています)。

第1バイトのビット7を1にするとR/Oになります。

また第2バイトのビット7を1にするとSYSファイルになります。

[注記]CP/MではR/Oに指定したファイルは書き換え、削除が禁止されます。またSYSに指定した場合DIRコマンドで表示されなくなります。

しかしCP/M互換DOSでは利便性を優先したため、ファイル属性をどのように設定しても、書き換え、削除、DIR表示に影響はあてません([第394回](ワンボード)参照)。ただしCP/Mユーティリティ(たとえばPIP)では、ファイル属性の指定は本来のCP/Mの場合と同じように有効に働きます。

DEレジスタに上記のように拡張子部分を書き換えたファイル名情報のあるFCBアドレスをセットしてコールすると、そのファイル名でサーチをして、そのファイル名が見つかった場合には、指定した属性情報がディスクのそのファイル名のFCBエリアに書き込まれます。

ファイル名が見つかった場合にはAレジスタに00~03のいずれかが入ってリターンします。

みつからなかった場合にはAレジスタにFFが入ってリターンします。

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 9. 0000~00FFの内容 でも説明をしています。

Ⅳ CP/M互換DOSのビルトインコマンドとトランザクションファイル 11. トランザクションファイルの実行 も参照してください。

ファンクション1Eをコールする前にセットする必要があるのはFCB先頭のドライブNo.~拡張子までです。

[参考記事]第97回(ワンボード)、第99回(ワンボード)、第144回(ワンボード)、第214回(ワンボード)、第122回(MYCPU80)

[サンプルプログラム]MFTST17、MFTST107

* MFTST17.COM、MFTST107.COMの使用例は[第122回](MYCPU80)にあります。

31. ファンクション1F ディスクパラメータアドレスのセット

ディスクパラメータブロックの先頭アドレスを取得します(HLLレジスタに入ります)。

```
MVI C, 1F
CALL $0005
```

ディスクパラメータブロックは15バイトのエリアで、ディスクの基本的な情報(セクタ数/トラックや総ブロック数など)が書かれています。

ディスクパラメータはCP/M互換DOSの起動時にZBIOSが管理しているエリアからMBDOSの管理するエリアにコピーされます。

ファンクション1Fはコピー後のディスクパラメータブロックの先頭アドレスを取得してHLLレジスタに入ります。

参考までにMBDOSが管理するディスクパラメータブロックのアドレスと中身は次の通りです。

```
E920 1000 1トラック(ブロック)当たりのセクタ数。0010H=16
E922 04   ブロックシフトNo.(BIOS内部で計算に使われる)
E923 0F   セクタ/ブロック-1
E924 00   このシステムでは使いません
E925 FF03 ディスクサイズ(総ブロック数-1) 03FFH=1023
E927 7F00 ディレクトリサイズ(管理できる最大ファイル名数-1) 007FH=127
E929 C000 ALVの先頭2バイトに置かれているディレクトリブロックの配置
E92B 0000 このシステムでは使いません
E92D 0000 オフセット。使用可能な最初のデータブロックの位置
```

[参考記事]第96回(ワンボード)、第143回(ワンボード)、第212回(ワンボード)、第121回(MYCPU80)

[サンプルプログラム]MFTST16

* MFTST16. COMの使用例は[第121回](MYCPU80)にあります。

32. ファンクション20 ユーザーコードのセット

複数の人でディスクを共用するための機能です。
CP/M互換DOSでは扱いません。

33. ファンクション21 ランダムリード

```
LXI D, $XXXX (FCBアドレス)
MVI C, 21
CALL $0005
```

ファイルオープン(ファンクション0F、ファンクション16)で指定したのと同じFCBアドレスをDEレジスタにセットしてコールすることで、FCBの34、35バイト目で示されるランダムレコード番号のセクタのデータ(128バイト)がDMAバッファに読み出されます(FCBの36バイト目は必ず00にしておきます)。

シーケンシャルリード、シーケンシャルライトと異なり、FCBエリアの34、35バイト目はファンクションコール後も+1されません。

レコードデータが正常に読み出された場合にはAレジスタに00が入ります。
エラーが発生した場合にはAレジスタに下記のコードが入ります。

- 01 ディレクトリに登録されていないブロックを指定した
- 03 このエクステントをクローズできない(意味不明)
- 04 登録されていないエクステントを指定した
- 06 レコードNo.がアクセスできる範囲を超えている

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 9. 0000~00FFの内容でも説明をしています。

FCBエリアの34~36バイト目の値については[第102回](ワンボード)で説明をしています。

またランダムアクセスファイルのFCBエリアについては[第221回](ワンボード)でも説明をしています。

[参考記事]第108回(ワンボード)、第225回(ワンボード)、第394回(ワンボード)、第124回(MYCPU80)

[サンプルプログラム]MFTST21

* MFTST21. COMの使用例は[第124回](MYCPU80)にあります。

ユーザーの便を考えてレコード番号表ファイル作成プログラム(MRAFTDOT. COM)も作成しました。

このテストよりも前に次のファンクション22(ランダムライト)のテストを行なってランダムアクセスファイルを作成してください。

34. ファンクション22 ランダムライト

```
LXI D, $XXXX (FCBアドレス)
MVI C, 22
CALL $0005
```

ファイルオープン(ファンクション0F、ファンクション16)で指定したのと同じFCBアドレスをDEレジスタにセットしてコールすることで、DMAバッファの1レコード分のデータ(128バイト)が、FCBの34、35バイト目で示されるランダムレコード番号のセクタに書き込まれます(FCBの36バイト目は必ず00にしておきます)。

シーケンシャルリード、シーケンシャルライトと異なり、FCBエリアの34、35バイト目はファンクションコール後も+1されません。

すでにデータが存在するランダムレコードに対してこのファンクションを使うと、指定したレコードナンバーのセクタが書き換えられます(クローズしなくてもディスクの内容は書き換えられてしまうことに注意してください)。

しかし最後にクローズは必要です。

レコードデータが正常に読み込まれた場合にはAレジスタに00が入ります。

エラーが発生した場合にはAレジスタに下記のコードが入ります。

- 02 ディスクに空きがなくなった
- 03 このエクステントをクローズできない(意味不明)
- 05 ディレクトリに空きがなくなった

06 レコードNo.がアクセスできる範囲を超えている

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 7. 0000~00FFの内容 でも説明をしています。

FCBエリアの34~36バイト目の値については[第102回](ワンボード)で説明をしています。

またランダムアクセスファイルのFCBエリアについては[第221回](ワンボード)でも説明をしています。

[参考記事]第102回(ワンボード)、第145回(ワンボード)、第216回(ワンボード)~第224回(ワンボード)、第393回(ワンボード)、第123回(MYCPU80)

[サンプルプログラム]MFTST20B

* MFTST20B. COMの使用例は[第123回](MYCPU80)にあります。

ユーザーの便を考えてレコード番号表作成プログラム(RAFTDOUT. BIN)も作成しました。

35. ファンクション23 ファイルサイズの計算

```
LXI D, $XXXX (FCBアドレス)
```

```
MVI C, 22
```

```
CALL $0005
```

FCBに指定したファイルのサイズ(レコード数)がFCBエリアの第34、35バイト目に入れます。

ただしランダムアクセスファイルについてはそのファイルの現在の最大のレコードNo.+1が入れます。

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 7. 0000~00FFの内容 でも説明をしています。

[参考記事]第109回(ワンボード)、第148回(ワンボード)、第226回(ワンボード)、第125回(MYCPU80)

[サンプルプログラム]MFTST22

* MFTST22. COMの使用例は[第125回](MYCPU80)にあります。

36. ファンクション24 ランダムレコードアクセスポインタの更新

```
LXI D, $XXXX (FCBアドレス)
```

```
MVI C, 22
```

```
CALL $0005
```

シーケンシャルファイルをシーケンシャルリード、ライトしていて、その途中でファンクション24をコールすると、最後にシーケンシャルアクセスしたレコードNo.+1のレコードNo.がFCBの第34~36バイト目に入れます(第36バイト目は常に00)。

このファンクションを実行したあとでランダムリード(ファンクション21)やランダムライト(ファンクション22)を実行すると、そのレコードNo.にランダムアクセスすることができます。

FCBエリアについては[第47回](ワンボード)で説明をしています。

またⅢ CP/M互換DOSの基本情報 7. 0000~00FFの内容 でも説明をしています。

[参考記事]第111回(ワンボード)、第149回(ワンボード)、第233回(ワンボード)~第236回(ワンボード)、第129回(MYCPU80)

[サンプルプログラム]MFTST25

* MFTST25. COMの使用例は[第129回](MYCPU80)にあります。

37. ファンクション25 ディスクドライブのリセット

指定するディスクドライブをリセットします(複数指定可)。

```
LXI D, $xxxx (ディスクベクトルの値)
```

```
MVI C, 25
```

```
CALL $0005
```

A~Pの16台のディスクドライブを16ビットの値に1ドライブを1ビットとして割り当て(下の図)、そのうちリセットしたい

ドライブに対応するビットのみ1にします。

そのようにして作成した16ビットの値をDEレジスタにセットしてファンクション25をコールすると、指定したドライブがリセットされログインベクトル、R/Oベクトルのそれぞれ同じ位置のビットが0になります。

ログインディスクは、**24. ファンクション18 ログインベクトルの取り出し** を参照してください。またR/Oベクトルは、**29. ファンクション1D R/Oベクトルの取り出し** を参照してください。

bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

[注記1]CP/M互換DOSではライトプロテクトは働きません。詳しくは第394回(ワンボード)を参照してください。

[注記2]CP/M互換DOSでは仮想ディスクドライブA~Dの4ドライブしか実装していません。またZドライブにはログインベクトル、ROベクトルはありません。

[参考記事]第100回(ワンボード)、第142回(ワンボード)、第209回(ワンボード)、第394回(ワンボード)、第119回(MYCPU80)

[サンプルプログラム]MFTST18、MFTST19

* MFTST18. COM、MFTST19. COMの使用例は[第119回](MYCPU80)にあります。

38. ファンクション26 未定義

39. ファンクション27 未定義

40. ファンクション28 ゼロファイルを伴うランダムライト

基本的な動作は **ファンクション22(ランダムライト)** と同じです。

```
LXI D, $XXXX (FCBアドレス)
MVI C, 28
CALL $0005
```

ファンクション22は指定したレコードNo.によって示されるセクタにDMAバッファのデータを書き込むだけですが、ファンクション28は、さらにそのレコードが含まれるブロックに未使用のセクタが残っている場合にはそのセクタをすべて00で埋めます。

それ以外は **ファンクション22(ランダムライト)** と同じです。

[参考記事]第102回(ワンボード)、第110回(ワンボード)、第232回(ワンボード)、第395回(ワンボード)、第126回(MYCPU80)

[サンプルプログラム]MFTST20C

* MFTST20C. COMの使用例は[第126回](MYCPU80)にあります。

VI エラーコード

コマンドの実行中にエラーが発生すると、エラーメッセージが表示されるほか、エラーによっては `err` に続けてアルファベット1文字が表示されることがあります。

その意味は下記の通りです。

- A ファイルが見つからない。指定されたディスクに指定した名前のファイルがない。
- B このコマンドではワイルドカードは使えない。
- C ファイルをREAD中にエラーが発生した。
- D SAVEコマンドでファイルサイズ(第1パラメータ)が指定されていない。
- E パラメータの区切りがスペースでない。
- F ディレクトリエリアに空きがない。ファイル名が127を越えた。またはファイルにプロテクトがかかっている削除できない。
- G ファイルをWRITE中にエラーが発生した。
- H ファイルがクローズできない。
- M ユーザープログラムをLOAD中にエラーが発生した。

Ⅶ N8BIOSサブルーチン

BIOSはBasic Input Output Systemの略です。

その名からもわかるように、基本的なI/Oアクセスのためのシステムプログラムです。

CP/Mは異なるマシンの上で同じソフトウェアを実行するためのOS(オペレーションシステム)ですが、BIOSはその下層にあって、異なる機能や働きをするマシンを共通の土台であるCP/M OSとつなぐ役割をしています。

CP/M互換DOSのN8BIOSはCP/MのBIOSとエントリアドレスおよび動作が同じになるようにした基本的なI/Oアクセスサブルーチンです。

N8BIOSのサブルーチンはファンクションコールと同じようにユーザープログラム内でコールして使うことができますが、ファンクションコールに比べると、よりマシンに近い原始的な働きをします。

いわば上級者向きで、通常は直接コールして使われることは少ないですが、参考までに以下に簡単に説明をします。

なおファンクションコールはCレジスタにファンクションNo.を入れて、0005番地をコールして使いますが、N8BIOSのサブルーチンはアドレスを直接コールします。

以下の説明では最初にサブルーチンのアドレス、次にその働きを、最後にコール前にセットするレジスタの値とコール後に得られるレジスタの値について示します。

0) E200 コールドブート

システムが再起動されます。ユーザーが直接使ってははいけません。

1) E203 ウォームブート

N8BIOS以外のシステム(MCCP、MBDOS)が再ロードされ、システムが初期リセットされます。

このルーチンもユーザーが直接使ってははいけません。

2) E206 コンソールステータス

キーボードの状態をセンスします。

キーが押されていないときはAレジスタに00が入ります。

押されていないときはAレジスタにFFが入ります。

3) E209 コンソール入力

キーボードが押されるまで待って、押されるとそのキーの文字コードをAレジスタに入れてリターンします。

4) E20C コンソール出力

Cレジスタに入れた文字コードに対応する文字が画面に表示されます。

5) E20F リスト出力

Cレジスタに入れた文字コードに対応する文字がプリンタに出力されます。

プリンタがノットレディのときはレディになるまでリターンしません。

6) E212 232C出力

Cレジスタに入れた文字コードに対応する文字がRS232Cポートに出力されます。

RS232Cポートがノットレディのときはレディになるまでリターンしません。

7) E215 232C入力

RS232Cポートから受信した文字コードをAレジスタに入れてリターンします。

受信データが無いか、またはノットレディのときは受信するまでリターンしません。

8) E218 ホームポジションシーク

カレントディスクドライブがホームポジション(トラック0セクタ0)にセットされます。

9) E21B セレクトドライブ

BCレジスタの値に対応するディスクドライブが選択されます(A=00、B=01、...)。

10) E21E セットトラック

BCレジスタの値によって示されるトラック(ブロック)が選択されます。

11) E221 セットセクタ

BCレジスタの値によって示されるセクタが選択されます。

12) E224 DMAアドレスセット

BCレジスタの値によって示されるアドレスがセクタがDMAバッファのアドレスになります。

13) E227 セクタリード

選択されているトラック(ブロック)、セクタのデータがDMAバッファに読み込まれます。
正常終了ならAレジスタに00が入ってリターンします。
異常終了ならAレジスタに01が入ってリターンします。

14) E22A セクタライト

選択されているトラック(ブロック)、セクタにDMAバッファのデータが書き込まれます。
正常終了ならAレジスタに00が入ってリターンします。
異常終了ならAレジスタに01が入ってリターンします。

15) E22D プリントステータス

プリンタノットレディならAレジスタに00が入ってリターンします。
レディならAレジスタにFFが入ってリターンします。

16) E230 セクタトランスレート

CP/Mではこのルーチンはフロッピーディスクのスキューを考慮した論理セクタNo.から物理セクタNo.への変換(BCレジスタ→HLレジスタ)ですが、CP/M互換DOSでは意味がありません。
HLにBCの値をそのまま入れてリターンします。

Ⅷ CP/Mアプリケーションソフト(参考)

CP/Mの上で動作するアプリケーションソフトは非常にたくさんあります。

CP/M互換DOSは基本的な部分ではCP/Mと同じ動作をするようになってはいるはずですから、それらのアプリケーションも原則的には、CP/M互換DOSの上でも動作するはずですが、

しかしソフトウェアの中には特定の機種や機能に依存するものなどもあって、全てのCP/Mソフトウェアが動作するという保証はありません。

またCP/Mにもいろいろなバージョンがあって、その特定のバージョンに依存するソフトウェアのなかにはCP/M互換DOSでは動かないものも出てきます。

CP/M互換DOSはCP/M2. 2(58KB版)に準拠しています。

参考までにこれまで当社がCP/M互換DOSの上で基本的な動作が確認できたアプリケーションを紹介します。

いずれも動作確認を行なった2012年時点でインターネットからダウンロードしたものです。

以下にそのダウンロードサイトの情報も紹介しますが、サイトによってはその後に閉鎖されたりダウンロードができなくなる可能性もあります。以下の情報は当社がダウンロードを行なった時点のものです。

[注記]著作権について

多くのソフトウェアには著作権があります。現在のところここで紹介するのはソフトウェアに関する著作権が確立する以前のもので、その権利はあいまいですが、著作権フリーということではありません。

ここで紹介するソフトについて当社はなんらの保証を与えるものではありません。

ダウンロードにあたってはあくまで個人の責任においておこなってください。

1) MBASIC(BASIC-80)

ダウンロードサイトは <http://www.retroarchive.org/cpm/lang/lang.htm> です。

[第157回](ワンボード)に上記サイトへのリンクがあります。

ページの中ほどの [Mbasic.com](http://www.mbasic.com) をクリックするとMbasic.com(V5. 21)がダウンロードできます。

[第139回](MYCPU80)[第140回](MYCPU80)を参考にしてください。

MBASIC[Enter]で起動できます。

MBASICを終了してCP/M互換DOSに戻るには

SYSTEM[Enter]

と入力します。

ユーザープログラムは

LOAD "ファイル名(. 拡張子)"[Enter]

で仮想FDDからLOADできます。

プログラムをSAVEするときは

SAVE "ファイル名(. 拡張子)", A[Enter]

と入力します。

拡張子を省略した場合はBASという拡張子が自動的に付加されます。

SAVEのときにパラメータ(, A)をつけると、拡張子はBASですが中身はTXTファイルになります。

, Aを省略するとMBASICの内部コードでセーブされるため、テキストエディタで開くことができません。

[注意]ZB3BASICやCP/M互換DOSはSAVE/LOAD時のファイル名は小文字でも通りますが、MBASICではファイル名に限り大文字を使います。

コマンドやプログラム中の命令は小文字でも構いません。

MBASICのユーザープログラムの実行を中止するときは[Ctrl]+[C]を入力します。

MBASIC MANUAL(英文)

<http://www.classiccmp.org/cini/pdf/Microsoft/mbasic.pdf>

[参考記事]第191回(ワンボード)(上記サイトへのリンクがあります)

2) STARTREK

MBASIC用STARTREKプログラムです。

[第10回](MYCPU80)でダウンロード先について説明をしています。

こちらでダウンロードしたものを解凍して付属CDROMのZフォルダに入れてあります。
STARTREK. BASとTREKINST. BASです。
MBASIC用のSTARTREKはMBASICがないと使えません。先にMBASICを入手してください。
ZフォルダのSTARTREK. BASとTREKINST. BASをMBASICと同じ仮想FDDにコピーしてください。
MBASICを起動後
LOAD "STARTREK"[Enter]
でロードします(ファイル名等の入力は小文字は不可)。
RUN[Enter]
でゲームが開始されます。
具体的な操作については[第376回](ワンボード)～[第389回](ワンボード)を参照してください。
ZB3BASIC用STARTREKの実行例ですが使い方は同じです。

3) FORTRAN80

MBASICと同じくMicrosoft社のCP/M用FORTRANです。
ダウンロードサイトは <http://www.retroarchive.org/lang/lang.htm> です。
Mbasic.comの5行くらい下のF80. ZIPをクリックするとFORTRAN80とその関連ファイルの圧縮ファイルがダウンロードできます。
Windows7ではダウンロード後自動的に解凍されますが、OSによっては解凍ツールなどで解凍する必要があります。
そのうちのF80. COM、L80. COM、FORLIB. RELをZB3DOSフォルダ内のZフォルダにコピーしてください。
Zフォルダは最初に/CPMコマンドを実行したときに自動で生成されます。
そのままZフォルダで使うかA～D仮想ドライブにコピーして使用します。
具体的な使い方については[第168回](ワンボード)～[第173回](ワンボード)を参考にしてください。

FORTRAN80 MANUAL(英文)

http://bitsavers.trailing-edge.com/pdf/microsoft/cpm/Microsoft_FORTRAN-80_Ver3.4_Users_Manual_Nov80.pdf

[参考記事]第191回(ワンボード)(上記サイトへのリンクがあります)

4) MACRO80

MBASICと同じくMicrosoft社のCP/M用アセンブラです。
FORTRAN80と同じ圧縮ファイルF80. ZIPに入っています。
M80. COMとL80. COMをZB3DOSフォルダ内のZフォルダにコピーしてください(もし先にFORTRAM80関係のファイルをコピーしていたら、追加でコピーするのはM80. COMだけです)。
Zフォルダは最初に/CPMコマンドを実行したときに自動で生成されます。
そのままZフォルダで使うかA～D仮想ドライブにコピーして使用します。
具体的な使い方については[第189回](ワンボード)～[第195回](ワンボード)を参考にしてください。

MACRO80 MANUAL(英文)

<http://www.retroarchive.org/cpm/lang/MACRO-80.PDF>

[参考記事]第191回(ワンボード)(上記サイトへのリンクがあります)

5) Wordstar

Micropro社のCP/M用ワープロソフトです。
ダウンロードサイトは <http://www.retroarchive.org/cpm/text/text.htm> です。
上から8行目、WordStar 3.3 for CP/M-80/Kayproの前のリンクをクリックすると圧縮ファイルをダウンロードできます(その何行か下にもWordStar 3.0がありますが、こちらはOVERLAY ERRORが発生します)。
ファイルを解凍して、その中のWS. COMとWSMSG. OVRとWSOVLY1. OVRをZフォルダにコピーします。
そのあとA～Dのいずれかの仮想ドライブにコピーします。
具体的な使い方については[第398回](ワンボード)を参考にしてください。

WordStar3.3 Reference Manual(英文)

http://bitsavers.trailing-edge.com/pdf/microPro/Wordstar_3.3/Wordstar_3.3_Reference_Manual_1983.pdf

[参考記事]第400回(ワンボード)(上記サイトへのリンクがあります)

6) COBOL-80

MBASICと同じくMicrosoft社のCP/M用COBOLです。

ダウンロードサイトは <http://www.retroarchive.org/lang/lang.htm> です。

F80.ZIPの下にMSCOBOL.ZIPをクリックするとCOBOL-80とその関連ファイルの圧縮ファイルがダウンロードできます。

Windows7ではダウンロード後自動的に解凍されますが、OSによっては解凍ツールなどで解凍する必要があります。

COBOL-80については使い方は詳しく調べていません。

[第429回](ワンボード)~[第431回](ワンボード)を参考にしてください。

COBOL-80 MANUAL(英文)

http://bitsavers.informatik.uni-stuttgart.de/pdf/microsoft/cpm/Microsoft_COBOL-80_1978.pdf

[参考記事]第429回(ワンボード)(上記サイトへのリンクがあります)

IX エスケープシーケンス

エスケープシーケンスはESCコード(1BH)に続けて1個以上のコードを記述することで、特殊な画面制御を行なうものです。

もともとディスプレイ装置などの端末側に実装されていた機能ですから、オリジナルのCP/Mにはエスケープシーケンスの機能はありません。

しかしCP/M用のアプリケーションソフトの中には特定の表示端末を想定したエスケープシーケンスを利用しているものがあって、それらのソフトは利用しているエスケープシーケンスがないと期待通りには動いてくれないことがわかってきました。

エスケープシーケンスは機種によって異なっている機能も多いようですが、標準的なものもいくつか知られています。

Windows98SEのMSDOSプロンプトでは標準的なエスケープシーケンスが使えることが確認できました。

しかしWindowsXP以後のWindowsOSではその機能はなくなってしまったようです。

そのあたりのことについては「ワンボードマイコンでCP/Mを！」[第425回]を参照願います。

WindowsXP以後のOSではエスケープシーケンスが使えないことがわかりましたので、このままではせっかくCP/M互換DOSを作っても、アプリケーションのなかにはエスケープシーケンスに対応していないために、使うことが出来ないというものがでてきてしまいます。

とはいえとてもその全てに対応することはできませんが、ND80ZⅢ用のZB3DOS(CP/M互換DOS)をご購入いただいたお客様から情報をいただいていくつかのエスケープシーケンスを、Windows側のプログラム(MZB3DOS.EXE)に組み込みました。

以下はその一覧です。

この機能はWindows側のプログラムに組み込んでありますから、ZB3DOS(CP/M互換DOS)だけではなくてZB3BASICでも使うことができます。

エスケープシーケンスの使用例は「MYCPU80でCP/Mを」[第130回][第131回]にありますので参照してください。

(1) ESC=YX

カーソルを左上隅を(0, 0)とするとき座標(y, x)に移動する。ただしY=y+20H、X=x+20Hの文字コードの文字で示す。

たとえば(5, 8)なら

ESC=%(

になる。%=25H、(=28H。

(2) ESC[2J 画面クリア

(3)

ESC[30m 文字色を黒に指定
ESC[31m 文字色を赤に指定
ESC[32m 文字色を緑に指定
ESC[33m 文字色を黄に指定
ESC[34m 文字色を青に指定
ESC[35m 文字色をマゼンタに指定
ESC[36m 文字色をシアンに指定
ESC[37m 文字色を白に指定
ESC[39m 文字色を標準色に戻す

ESC[40m 背景色を黒に指定
ESC[41m 背景色を赤に指定
ESC[42m 背景色を緑に指定
ESC[43m 背景色を黄に指定
ESC[44m 背景色を青に指定
ESC[45m 背景色をマゼンタに指定
ESC[46m 背景色をシアンに指定
ESC[47m 背景色を白に指定
ESC[49m 背景色を標準色に戻す

(4)

ESC[K カーソル位置から行末までをクリア

ESC[nA カーソルを上にな行移動

ESC[nB カーソルを下にn行移動
ESC[nC カーソルを右にn桁移動
ESC[nD カーソルを左にn桁移動

(5)

ESC[0m 文字修飾をもとに戻す
ESC[1m ~ ESC[7m 文字の色指定を明るい白にする

(6)

ESC[yy;xxH
yyは01~25で行位置を指定
xxは01~80で桁位置を指定

[memo]