

Nextor 2.0 Programmers Reference

By Konamiman, 10/2014

目次

1. はじめに.....	2
2. 既存のファンクションコールの変更.....	3
2.1. _STROUT (09h).....	3
2.2. _ALLOC (1Bh).....	4
2.3. _RDABS (2Fh) / _WRABS (30h).....	5
2.4. _DPARM (31h).....	6
2.5. _DEFER (64h).....	8
2.6. _EXPLAIN (66h).....	9
2.7. _FORMAT (67h).....	10
2.8. _DOSVER (6Fh).....	12
2.8.1. Nextor の MSX-DOS 1 モードの検出.....	13
2.8.2. 各モードでの _DOSVER の戻り値の例.....	14
3. 新規のファンクションコール.....	15
3.1. Get/set fast STROUT mode (_FOUT, 71h).....	16
3.2. Print a zero-terminated string (_ZSTROUT, 72h).....	16
3.3. Read absolute sectors from drive (_RDDRIV, 73h).....	17
3.4. Write absolute sectors from drive (_WRDRV, 74h).....	17
3.5. Get/set reduced allocation information mode vector (_RALLOC, 75h).....	18
3.6. Get drive space information (_DSPACE, 76h).....	19
3.7. Lock/unlock a drive, or get lock state for a drive (_LOCK, 77h).....	20
3.8. Get information about a device driver (_GDRVR, 78h).....	21
3.9. Get information about a drive letter (_GDLI, 79h).....	23
3.10. Get information about a device partition (_GPART, 7Ah).....	24
3.11. Call a routine in a device driver (_CDRVR, 7Bh).....	26
3.12. Map a drive letter to a drive and device (_MAPDRV, 7Ch).....	27
3.13. Enable or disable the Z80 access mode for a driver (_Z80MODE, 7Dh).....	29
4. 新しいエラーコード.....	30
5. 拡張マッパーサポートルーチン.....	31
5.1. BLK_ALLOC : Allocate a memory block (+30h).....	31
5.2. BLK_FREE : Free a memory block (+33h).....	32
6. その他の特徴.....	33
6.1. _STROUT (09h) におけるエスケープシーケンス “ESC-Y” の不具合を修正.....	33
6.2. Nextor のバージョン変更.....	33
7. 変更履歴.....	35
7.1. v2.0.4.....	35
7.2. v2.0.3.....	35
7.3. v2.0 final.....	35
7.4. v2.0 Beta 2.....	35
7.5. v2.0 Beta 1.....	35
7.6. v2.0 Alpha 2b.....	35
7.7. v2.0 Alpha 2.....	35
8. 問い合わせ.....	36

1. はじめに

Nextor は MSX 用ディスクオペレーションシステムである MSX-DOS 2 の強化バージョンです。MSX-DOS 2.31 をベースにしており、100%の互換性があります。

本ドキュメントでは開発者視点で、Nextor が MSX-DOS 2 に追加した特徴の解説を行います。基本的には新しいファンクションコールについて解説しますが、その他の有用な情報も含んでいます。Nextor のデバイスドライバー開発については本ドキュメントに含まれていません。デバイスドライバー開発については、“*Nextor 2.0 Driver Development Guide*”を参照してください。

このドキュメントの読者には一般的な MSX、特に MSX-DOS 2 のアプリケーションを開発した経験がある方を想定しています。特に “*MSX2 テクニカル・ハンドブック*” の「第 3 部 MSX-DOS」、 “*MSX-Datapack Volume 3 turboR 版*” の「第 2 部 MSX-DOS 2」に記載されている内容を理解されている方を想定しています。また、事前に “*Nextor 2.0 User Manual*” を読んで Nextor についての理解を深めることをお奨めします。

2. 既存のファンクションコールの変更

この章では既存の MSX-DOS 2 のファンクションコールに対する Nextor の変更点を解説します。すべての変更は `_DPARM (31h)` の FAT16 対応を除いて互換性を保っています。

Nextor で変更されたファンクションコールについてすべての解説はせず、変更部分についてのみ解説します。変更部分以外の解説については、“MSX-Datapack Volume 3 turboR 版”の17章「ファンクションコール」を参照してください。

2.1. `_STROUT (09h)`

Fast STROUT モードが有効になっているとき、表示可能な文字列が最大で 511byte となります。文字列は '\$' (24h) で終了しますが、文字列が 511byte より長いときは、最初の 511byte 分のみ表示されます。Fast STROUT モードはデフォルトでは無効になっており、追加されたファンクションコール `_FOUT (71h)` で有効にします。

<code>_STROUT</code>	Nextor	MSX-DOS 2
コール手順	DE : 文字列のアドレス	
戻り値	なし	
解説	Fast STROUT モード 無効 : '\$' (24h) で終了 有効 : '\$' または 511byte の短い方で終了	'\$' (ASCII 24h) で終了

2.2. _ALLOC (1Bh)

対象ドライブに対する Reduced Allocation Information モードが有効になっているとき、クラスタ総数と未使用クラスタ数について、1 クラスタあたりの論理セクタ数を掛け合わせて得られる容量が 32MB を超えるときに、32MB 未満になるように偽の値を返します。Reduced Allocation Information モードはデフォルトではすべてのドライブに対して無効になっており、追加されたファンクションコール _RALLOC (75h) で指定したいドライブに対して有効にします。

さらに Nextor 起動時に環境変数 "ZALLOC" に "ON" を設定することにより、Reduced Allocation Information モードは Zero Allocation Information モードになります。このとき、_ALLOC は Reduced Allocation Information に指定されているドライブについて、未使用クラスタ数 0 を返します。

32MB を超えるドライブにおいて、COMMAND2.COM 2.20 や 2.30 等の FAT16 に対応していないプログラムがドライブの総容量や空き容量を間違っで計算してしまうため、約 32MB と計算するようにします。

_ALLOC	Nextor	MSX-DOS 2
コール手順	E : 論理ドライブ番号 (00h = カレント, 01h = A:など)	
戻り値	A : 1 クラスタあたりのセクタ数 BC : セクタサイズ (常に 512 = 0200h) DE : ディスク上のクラスタの総数 HL : ディスク上の未使用クラスタ数 IX : PB へのポインタ IY : 最初の FAT セクタへのポインタ	
解説	Reduced Allocation Information モード 無効 : 通常の戻り値 有効 : $DE \times A$ 、 $HL \times A$ それぞれについて値が FFFFh 以下のときは通常の戻り値 FFFFh より大きいときは FFFFh 以下になるまで DE や HL を小さくする 環境変数 ZALLOC = ON のときは HL = 0000h	

2.3. _RDABS (2Fh) / _WRABS (30h)

これらのファンクションコールは FAT12 のドライブに対してのみ有効となります。FAT16 や不明なファイルシステムのドライブに対しては “Not a DOS disk” エラーを返します。

従来の MSX-DOS でも FAT12 のみサポートしており、厳密に言うなら仕様は変わっていません。しかしながら、MSX-DOS では FAT16 のドライブをアクセスしたときでも FAT12 として認識して動作し、エラーを返しません。Nextor では FAT16 のドライブは扱えないようにします。これは、CHKDSK や IMPROVE のような、Low Level でのディスクアクセスするプログラムが実行されたときに FAT12 と認識してデータ破壊を引き起こすことを防ぐためです。

Nextor 対応アプリケーションはこれらのファンクションコールの代わりに、追加されたファンクションコール _RDDRIV (73h) と _WRDRV (74h) を使って下さい。追加されたファンクションコールでは 32bit のセクタ番号を使用し、ファイルシステムにかかわらずディスクのアクセスが可能になります。

_RDABS	Nextor	MSX-DOS 2
コール手順	DE : セクタ番号 L : 物理ドライブ番号 (00h = A:, 01h = B: など) H : 読み出すセクタ数	
戻り値	A : エラーコード (00h = エラーなし)	
解説	ファンクションコールの使い方に変更はないが、FAT16 のドライブのときはエラーが返る	FAT16 のドライブのときは FAT12 としてアクセスしてしまう

_WRABS	Nextor	MSX-DOS 2
コール手順	DE : セクタ番号 L : 物理ドライブ番号 (00h = A:, 01h = B: など) H : 書き込むセクタ数	
戻り値	A : エラーコード (00h = エラーなし)	
解説	ファンクションコールの使い方に変更はないが、FAT16 のドライブのときはエラーが返る	FAT16 のドライブのときは FAT12 としてアクセスしてしまう

2.4. _DPARAM (31h)

このファンクションではディスクパラメータのオフセット 18 ~ 1Bh に論理セクタの総数を 32bit で保存します。さらに論理セクタの総数が FFFFh を超えるとき、オフセット 09 ~ 0Ah は 0000h となります。

また、オフセット+1Ch にファイルシステムの種類を保存します。

00h : FAT12

01h : FAT16

FFh : その他

注意: 現在、Nextor で扱えるのは FAT12 と FAT16 のみです。

_DARM	Nextor	MSX-DOS 2
コール手順	DE : ディスクパラメータ (32byte のバッファ) へのポインタ L : 論理ドライブ番号 (00h = カレントドライブ, 01h = A:など)	
戻り値	A : エラーコード (00h = エラーなし) DE : 保存される	
解説	オフセット 09 ~ 0Ah : 論理セクタの総数 (16bit) ただし、FFFFh を超えるときは 0000h オフセット 18 ~ 1Bh : 論理セクタの総数 (32bit) オフセット 1Ch : ファイルシステムの種類 00h : FAT12, 01h : FAT16, FFh : その他	オフセット 18 ~ 1Fh はシステム予約 (常に 00h)

ディスクパラメータブロックフォーマット (オフセットは DE+xxh を示す)

オフセット	Nextor	MSX-DOS 2
01 ~ 02h	セクタのサイズ (バイト単位)	
03h	クラスタのサイズ (セクタ単位、2 の n 乗で指定)	
04 ~ 05h	予約セクタ数 (ブートセクタで使用するセクタ数)	
06h	FAT の数	
07 ~ 08h	ルートディレクトリエントリの数 (作成可能なエントリの数)	
09 ~ 0Ah	総セクタ数 (16bit、FFFFh を超えるときは 0000h)	総セクタ数 (16bit)
0Bh	メディア ID	
0Ch	FAT サイズ (セクタ単位)	
0D ~ 0Eh	ルートディレクトリの先頭セクタ番号	
0F ~ 10h	最初のデータのセクタ番号	
11 ~ 12h	最大クラスタ番号	
13h	ダーティディスクフラグ (00h 以外で可能)	
14 ~ 17h	ボリューム ID (00 ~ 7Fh で構成、FFFF FFFFh のとき、ボリューム ID なし)	
18 ~ 1Bh	総セクタ数 (32bit)	システム予約 (現在は常に 00h)
1Ch	ファイルシステムの種類 00h = FAT12, 01h = FAT16, FFh = その他	
1Dh ~ 1Fh	システム予約	

ダーティディスクフラグは、ディスク中に UNDEL コマンドで復活できるファイルがあることを示すフラグです。したがって、ファイルあるいはディレクトリにクラスタが割り当てられるとこのフラグはリセットされます。

2.5. _DEFER (64h)

ディスクエラーが発生したときにコールされるユーザーのルーチンに渡すパラメータを 32bit セクタ番号対応にします。

MSX-DOS2 でのパラメータ仕様

- C : bit 3 - セクタ番号が有効のときにセット
- DE : セクタ番号 (C の bit 3 がセットされているとき)

Nextor でのパラメータ仕様

- C : bit 3 - セクタ番号が有効で FFFFh以下のときにセット (bit 4 もセットされ、HL=0 となる)
- C : bit 4 - セクタ番号が有効のときにセット
- HL:DE : セクタ番号 (C の bit 4 がセットされているとき)

_DEFER	Nextor	MSX-DOS 2
コール手順	DE : ディスクエラールーチンのアドレス、0000h で定義解除	
戻り値	A : 00h (エラーは発生しない)	
解説	<p>ファンクションコールの使い方に変更は無く、ユーザールーチンに渡すパラメータが変更される</p> <p>パラメータ</p> <p>A : エラーの原因となったエラーコード</p> <p>B : 物理ドライブ番号 (00h = A:, 01h = B: など)</p> <p>C : bit 0 - 書き込みでセット</p> <p>bit 1 - 無視の処理が望ましくないときセット</p> <p>bit 2 - オートアボートを指示するときセット</p> <p>bit 3 - セクタ番号が有効で FFFFh以下のときにセット</p> <p>(bit 4 もセット、HL は 0000h となる)</p> <p>bit 4 - セクタ番号が有効のときにセット</p> <p>HL:DE : セクタ番号 (32bit)</p> <p>C の b4 がセットされているとき</p> <p>結果</p> <p>A : 00h = システムエラールーチンのコール</p> <p>01h = アボート</p> <p>02h = 再試行</p> <p>03h = 無視</p>	<p>A : エラーの原因となったエラーコード</p> <p>B : 物理ドライブ番号 (00h = A:, 01h = B: など)</p> <p>C : bit 0 - 書き込みでセット</p> <p>bit 1 - 無視の処理が望ましくないときセット</p> <p>bit 2 - オートアボートを指示するときセット</p> <p>bit 3 - セクタ番号が有効のときにセット</p> <p>DE : セクタ番号 (16bit)</p> <p>C の b3 がセットされているとき</p> <p>結果</p> <p>A : 00h = システムエラールーチンのコール</p> <p>01h = アボート</p> <p>02h = 再試行</p> <p>03h = 無視</p>

2.6. _EXPLAIN (66h)

環境変数 “ERRLANG” が “EN” に設定されているとき、漢字モードであっても英語のエラーメッセージを返します。

_EPLAIN	Nextor	MSX-DOS 2
コール手順	B : 説明すべきエラーコード DE : 64 バイトの文字列バッファへのポインタ	
戻り値	A : 00h B : 00h あるいは変更なし DE : エラーメッセージが入る	
解説	ファンクションコールの使い方に変更はなく、環境変数 ERRLANG = EN のときは漢字モードであっても英語でエラーメッセージを表示	エラーメッセージで表示される言語はエラー発生時の言語モードによる

2.7. _FORMAT (67h)

MSX-DOS 2 ではパラメータ FEh、FFh によって実際はフォーマットせずにブートセクタのみ MSX-DOS 2 用に更新するオプションがあります。この機能は FIXDISK で MSX-DOS 1 用ディスクを MSX-DOS 2 用ディスクに更新するために使われます。

Nextor ではさらに 3 つのパラメータを追加します。

- FDh : FAT12 もしくは FAT16 のブートセクタ情報を Nextor の標準ブートセクタ情報に更新します。
それ以外のディスクでは “Not a DOS disk” エラーとなります。
- ① ブートセクタのオフセット 03h (OEM Name) に “NEXTOR” をセット
- ② ブートセクタのオフセット 27h にボリューム ID (4byte) 、2Bh にボリューム名 (11byte) 、36h にファイルシステムタイプ (32byte の文字列 “FAT12” または “FAT16”、残りは 20h) をセット上記以外のディスクパラメータは変更しません。

Nextor 標準の 1C ~ 3Dh は一般的な拡張 BPB と同じ仕様となっており、既に拡張 BPB が存在しているときは、OEM Name のセットのみを行います。このオプションは他のシステムでフォーマットされたディスクを Nextor の標準ブートセクタ情報に修正したいときに使います。

- FCh : FAT12 のディスクに対しては MSX-DOS 2 の標準ブートセクタ情報に更新します。
FAT16 のディスクに対しては FDh と同じです。
- FBh : FAT とルートディレクトリの情報のみ消去するクイックフォーマットを行います。
FAT12 もしくは FAT16 のディスクでないときは “Not a DOS disk” エラーとなります。

オプション 01~09h でフォーマットしたときは MSX-DOS 2 の標準ブートセクタ情報となります。

_FORMAT	Nextor	MSX-DOS 2
コール手順	B : 論理ドライブ番号 (00h = カレント, 01h = A:) A : 00h 選択文字列を返す 01 ~ 09h この選択をフォーマットする 0A ~ FAh 不正 FBh クイックフォーマット 以下はブートセクタの更新のみ実施 FCh FAT12 : MSX-DOS 2 標準で更新 FAT16 : Nextor 標準で更新 FDh Nextor 標準で更新 FEh MSX-DOS 標準で更新 オフセット 1Eh 以降は更新しない FFh MSX-DOS 2 標準で更新 HL : バッファへのポインタ (A = 01 ~ 09h のとき) DE : バッファのサイズ (A = 01 ~ 09h のとき)	B : 論理ドライブ番号 (00h = カレント, 01h = A:) A : 00h 選択文字列を返す 01 ~ 09h この選択をフォーマットする 0A ~ FDh 不正 以下はブートセクタの更新のみ実施 FEh MSX-DOS 標準で更新 オフセット 1Eh 以降は更新しない FFh MSX-DOS 2 標準で更新 HL : バッファへのポインタ (A = 01 ~ 09h のとき) DE : バッファのサイズ (A = 01 ~ 09h のとき)
戻り値	A : エラーコード (00h = エラーなし) B : 選択文字列のスロット (エントリで A = 00h のとき) HL : 選択文字列のアドレス (エントリで A = 00h のとき)	

ブートセクタの構造

オフセット	Nextor	MSX-DOS 2	
00 ~ 02h	8086 のジャンプ命令 (MSX では無視される)		
03 ~ 0Ah	文字列 “NEXTOR”	未使用	
0B ~ 0Ch	セクタのサイズ (バイト単位、通常 0200h)		
0Dh	クラスタのサイズ (セクタ単位、2 の n 乗で指定)		
0E ~ 0Fh	予約セクタ数 (ブートセクタで使用するセクタ数、通常 0001h)		
10h	FAT の数 (通常 02h)		
11 ~ 12h	ルートディレクトリエントリの数 (作成可能なエントリの数)		
13 ~ 14h	総セクタ数 (16bit) ただし、FFFFh を超えるときは 0000h	総セクタ数 (16bit)	
15h	メディア ID		
16 ~ 17h	FAT サイズ (セクタ単位)		
18 ~ 19h	トラックあたりのセクタ数		
1A ~ 1Bh	ディスクの面数 (片面/両面)		
1C ~ 1Fh	隠されたセクタ数 (32bit) Nextor ではこの値は変更しません	1C ~ 1Dh	隠されたセクタ数
		1E ~ 1Fh	ブートプログラムの先頭 (30h) へのジャンプ命令
20 ~ 23h	総セクタ数 (32bit) ただし、0000 FFFFh 以下の時は 0000 0000h	20 ~ 25h	文字列 “VOL_ID”
24h	物理ドライブ番号 (00h)		
25h	ダーティディスクフラグ (00h 以外で可能)		
26h	拡張ブートレコーディングネチャ (29h のときボリューム名、ファイルシステムタイプが有効)	26h	ダーティディスクフラグ (00h 以外で可能)
27 ~ 2Ah	ボリューム ID (フォーマット時に乱数で決定される)		
2B ~ 35h	ボリューム名	2B ~ 2Fh	将来のための予約 (00h で埋める)
		30h ~	ブートプログラム
36 ~ 3Dh	ファイルシステムタイプ 文字列 “FAT12” または “FAT16”		

- 0B ~ 1Dh は FAT の BPB (BIOS Parameter Block) 仕様通りです。
MSX-DOS 1 では、1Eh からブートプログラムが始まります。
MSX-DOS 2 の 1C ~ 26h は MSX-DOS 2 独自仕様で、30h からブートプログラムが始まります。
Nextor 標準の 1C ~ 3Dh は 25h を除き FAT の拡張 BPB の仕様通りです。ブートプログラムは規定されていません。
- オフセット 26h が 28h または 29h のとき、Nextor はオフセット 25h をダーティディスクフラグとして参照します。
- 総セクタ数が 65535 (0000 FFFFh) 以下のとき、オフセット 20 ~ 23h の総セクタ数 (32bit) が 0000 0000h となるのは FAT の仕様通りです。_DPARM (31h) のオフセット 18 ~ 1Bh の総セクタ数 (32bit) には 32bit の正しい値が入ります。これは Nextor の仕様です。

2.8. _DOSVER (6Fh)

MSX-DOS 2 または Nextor (通常モードもしくは MSX-DOS 1 mode) が動作しているかをアプリケーションから判別するために使用します。

Nextor のバージョンを判別するには下記のパラメータ (ここでは Magic Number と呼びます) をセットしてコールする必要があります。

B = 5Ah
HL = 1234h
DE = ABCDh
IX = 00h

MSX-DOS 2 上でこの Magic Number をセットしてコールしたとき、IX = 00h が返ります。MSX-DOS 1 のファンクションコールでは裏レジスタやインデックスレジスタを含むすべてのレジスタの値が破壊されますが、MSX-DOS 2 では裏レジスタとインデックスレジスタについては、そのレジスタが値を返すときを除いて保存されるためです。Nextor 上で Magic Number をセットせずにコールしたとき、IX, IY の値は変更されません。MSX-DOS 2 との互換性を保つためです。Nextor 上で Magic Number をセットせずにコールしたときも IX, IY の値は変更されません。これは MSX-DOS 2 との互換性を保つためです。

Nextor 上で Magic Number をセットしてコールしたとき、下記の情報が返ります。

B = MSX-DOS カーネルのメジャーバージョン番号のエミュレート (常に 02h)
C = MSX-DOS カーネルのマイナーバージョン番号のエミュレート (常に 31h)
D = NEXTOR.SYS のメジャーバージョン番号 (BCD フォーマット)
E = NEXTOR.SYS のマイナーバージョン番号 (BCD フォーマット)
HL = カーネル内にある OS を示す文字列のアドレス
IXH = 01h
IXL = Nextor カーネルのメジャーバージョン番号 (02 ~ 0Fh)
IYH = Nextor カーネルのセカンダリバージョン番号 (00h ~ 0Fh)
IYL = Nextor カーネルのレビジョン番号 (00 ~ FFh)

アプリケーションからオペレーティングシステムを判別するには下記の手順で行って下さい。

1. Magic Number をセットして _DOSVER をコールする
2. A = 00h でないとき (エラーのとき)、MSX-DOS でも Nextor でもない
3. B < 02h なら MSX-DOS 1
4. IX = 00h なら MSX-DOS 2 なので B, C を参照する
5. IXH = 01h なら Nextor なので、IXL, IYH, IYL を参照する
6. IXH が 00h でも 01h でもないとき、MSX-DOS でも Nextor でもない

HL は 00h で終わる OS の名前を示す文字列のアドレスです。例えば “Nextor kernel version 2.0” というような文字列が格納されています。この文字列はマスターとなっているカーネルのスロット (スロット番号は F348h に格納) にあり、_RDSLTL で読むことができます。

_DOSVER	Nextor	MSX-DOS 2
コール手順	B : 5Ah HL : 1234h DE : ABCDh IX : 00h	なし
戻り値	A : エラーコード (00h = エラーなし) B : 02h C : 31h D : NEXTOR.SYS のメジャーバージョン E : NEXTOR.SYS のマイナーバージョン (D, E はいずれも BCD フォーマット) HL : カーネル内の OS を示す文字列のアドレス IXH : 01h IXL : Nextor カーネルのメジャーバージョン IYH : Nextor カーネルのセカンダリバージョン IYL : Nextor カーネルのレビジョン	A : エラーコード (00h = エラーなし) B : MSX-DOS カーネルのメジャーバージョン C : MSX-DOS カーネルのマイナーバージョン D : MSXDOS2.SYS のメジャーバージョン E : MSXDOS2.SYS のマイナーバージョン (B ~ E はいずれも BCD フォーマット)

2.8.1. Nextor の MSX-DOS 1 モードの検出

_DOSVER は Nextor の MSX-DOS 1 モードでも使用可能で、Nextor を判別することができます。

Nextor の MSX-DOS 1 モードで _DOSVER をコールしたときは、BC = 0100h となり、HL の値は意味を持ちません。MSX-DOS 1 では _DOSVER は F37Dh からしかコールできず (3章を参照)、IX 及び IY の内容は破壊されます。その為、IX で Nextor かどうかの判定ができません。その代わりに、_DOSVER は A = 01h を返します。

もし MSX-DOS 1 モードで動くアプリケーションで、Nextor カーネルか MSX-DOS 1 カーネルのどちらで動作しているかを知りたいときは、下記の手順となります。

1. Magic Number をセットして _DOSVER (F37Dh) をコールする
2. A = 01h, B = 01h, IXH = 01h なら Nextor の MSX-DOS 1 モードなので、IXL, IYH, IYL を参照する
3. A = 00h でないとき (エラーのとき)、MSX-DOS でも Nextor でもない
4. B < 02h なら MSX-DOS 1
5. IX = 00h なら MSX-DOS 2 なので B, C を参照する
6. IXH = 01h なら Nextor なので、IXL, IYH, IYL を参照する
7. IXH が 00h でも 01h でもないとき、MSX-DOS でも Nextor でもない

2.8.2. 各モードでの_DOSVERの戻り値の例

Kernel	Magic Number をセットしたとき	Magic Number をセットしなかったとき
Nextor MSX-DOS 1 mode	<u>A = 01h</u> B = 01h, C = 00h (MSX-DOS kernel 1.00) HL = カーネル内の OS を示す文字列のアドレス <u>IXH = 01h, IXL = 02h</u> (Nextor kernel 2.0.1) IYH = 00h, IYL = 01h	A = 00h B < 02h
Other OS	<u>A <> 00h</u> (Nextor の MSX-DOS1 kernel のときの A = 01h を先に判別する必要がある)	
MSX-DOS 1	A = 00h <u>B < 02h</u>	
MSX-DOS 2	A = 00h B = 02h, C = 31h (MSX-DOS2 kernel 2.31) D = 02h, E = 30h (MSX-DOS2.SYS 2.30) HL = カーネル内の OS を示す文字列のアドレス <u>IX = 00h</u>	A = 00h B = 02h, C = 31h (MSX-DOS2 kernel 2.31) D = 02h, E = 30h (MSX-DOS2.SYS 2.30) HL = カーネル内の OS を示す文字列のアドレス
Nextor	A = 00h B = 02h, C = 31h (MSX-DOS2 kernel 2.31) D = 02h, E = 10h (NEXTOR.SYS 2.10) HL = カーネル内の OS を示す文字列のアドレス <u>IXH = 01h, IXL = 02h</u> (Nextor kernel 2.0.1) IYH = 00h, IYL = 01h	A = 00h B = 02h, C = 31h (MSX-DOS2 kernel 2.31) D = 02h, E = 10h (NEXTOR.SYS 2.10) HL = カーネル内の OS を示す文字列のアドレス

下線部が判別に用いる戻り値です。

通常は、0005h または F37Dh を使い、下記の順番でチェックします。

“Other OS” → “MSX-DOS 1” → “MSX-DOS2” → “Nextor” (→ “Other OS”)

Nextor の MSX-DOS 1 モードの判別が必要なときは F37Dh を使い、下記の順番でチェックします。

“Nextor MSX-DOS 1 mode” → “Other OS” → “MSX-DOS 1” → “MSX-DOS2” → “Nextor” (→ “Other OS”)

3. 新規のファンクションコール

この章では Nextor で追加された新しいファンクションコールの解説をします。使い方は既存の MSX-DOS のファンクションコールと同様、C レジスタにファンクション番号をセットして 0005h または F37Dh をコールします。ソースコードではファンクションのショートネーム (例えば “_FOUT”) を使用することを推奨します。また、それらの名前はこのマニュアルの機能相互参照表でも使われています。

いくつかの新しいファンクションコールは MSX-DOS 1 モードでも使用できます。現時点では、_GDRVR, _GPART, _CDRVR 及び _GDLI のみです。新しいファンクションコールを MSX-DOS 1 モードで使用するとき、以下の制限があります。

- F37Dh をコールしなければなりません。0005h は使用できません。MSX-DOS.SYS が Nextor 用に作られていないためです。
- RAM をバッファとして使うとき、バッファはページ 1 (4000~7FFFh) にはおけません。ファンクション実行中は kernel がページ 1 にあるためです。

3.1. Get/set fast STROUT mode (_FOUT, 71h)

fast STROUT モードの有効／無効を切り替えます。

_FOUT	C : 71h
コール手順	A : 動作モード 00h = fast STROUT モードの状態取得 01h = fast STROUT モードをセット B : 有効／無効の切り替え 00h = 無効にする (A = 01h のときのみ) FFh = 有効にする (A = 01h のときのみ)
戻り値	A : エラーコード (00h = エラーなし) B : fast STROUT モードの状態 00h = 無効 FFh = 有効

有効にしたとき、_STROUT (09h) と_ZSTROUT (72h) の動作が速くなります。ただし、表示可能な文字列が最大で 511byte となります。文字列は 24h (_STROUT のとき) または 00h (_ZSTROUT のとき) で終了しますが、文字列が 511byte より長いときは先頭からの 511byte 分のみ表示します。

3.2. Print a zero-terminated string (_ZSTROUT, 72h)

DEレジスタで指される文字列の文字が通常の「コンソール出力」ファンクションを使用して出力されます。文字列は 00h で終了します。

_ZSTROUT	C : 72h
コール手順	DE : 文字列のアドレス
戻り値	A : 00h (エラーは返さない)

_STROUT (09h) とほぼ同じファンクションですが、文字列の終了は '\$' (24h) ではなく 00h で指定します。Fast STROUT モードが有効になっているとき、表示可能な文字列が最大で 511byte となります。文字列が 511byte より長いときは、最初の 511byte 分のみ表示されます。Fast STROUT モードはデフォルトでは無効になっており、追加されたファンクションコール _FOUT (71h) で有効にします。

3.3. Read absolute sectors from drive (_RDDR, 73h)

セクタをファイルとして解釈することなしに、ディスクからセクタを直接読み出します。セクタ番号をディスク上の物理的な位置に変換するため、ディスクは有効な DOS のディスクでなければなりません。セクタは現在のディスク転送アドレスに読み出されます。ディスクエラーは通常の方法で知らされます。

_RDDR	C : 73h
コール手順	A : 物理ドライブ番号 (00h = A:, 01h=B: など) B : 読み出すセクタ数 HL:DE : セクタ番号 (32bit)
戻り値	A : エラーコード (00h = エラーなし)

_RDABS (2Fh) とほぼ同じファンクションですが、FAT12 だけでなく FAT16 やその他のファイルシステム、ファイルシステムが存在しない状態であっても読み出し可能です。

3.4. Write absolute sectors from drive (_WRDR, 74h)

セクタをファイルとして解釈することなしに、ディスクへセクタを直接書き込みます。セクタ番号をディスク上の物理的な位置に変換するため、ディスクは有効な DOS のディスクでなければなりません。セクタは現在のディスク転送アドレスから書き込まれます。ディスクエラーは通常の方法で知らされます。

_WRDR	C : 74h
コール手順	A : 物理ドライブ番号 (00h = A:, 01h = B: など) B : 書き込むセクタ数 HL:DE : セクタ番号 (32bit)
戻り値	A : エラーコード (00h = エラーなし)

_WRABS (30h) とほぼ同じファンクションですが、FAT12 だけでなく FAT16 やその他のファイルシステム、ファイルシステムが存在しない状態であっても書き込み可能です。

3.5. Get/set reduced allocation information mode vector (_RALLOC, 75h)

reduced allocation information モードの有効／無効を切り替えます。各 bit はそれぞれのドライブに割り当てられます。L レジスタの bit 0 が A:ドライブ、bit 1 が B:ドライブ・・・となります。この bit が 1 のときは reduced allocation information モードが有効となり、_ALLOC (1Bh) コールはクラスタ総数と未使用クラスタ数が 32MB を超えるときに 32MB 未満になるように偽の値を返すようになります。各 bit のデフォルトは 0b です。

_RALLOC	C : 75h
コール手順	A : 動作モード 00h = reduced allocation information モードの状態取得 01h = reduced allocation information モードをセット HL : 有効／無効の切り替え 0b = 無効にする (A = 01h のときのみ) 1b = 有効にする (A = 01h のときのみ) (bit 0 = A:, bit 1 = B: など)
戻り値	A : 00h (エラーは返さない) HL : 物理ドライブの reduced allocation information モードの状態 (bit 0 = A:, bit 1 = B: など)

さらに Nextor 起動時に環境変数 "ZALLOC" に "ON" を設定することにより、Reduced Allocation Information モードは Zero Allocation Information モードになります。このとき、_ALLOC (1Bh) は Reduced Allocation Information に指定されているドライブについて、未使用クラスタ数 0 を返します。

3.6. Get drive space information (_DSPACE, 76h)

ドライブの総容量または空き容量を返します。容量情報はドライブのファイルシステムやクラスタサイズにかかわらず kB 単位で返します。kB 未満の余り容量も別のレジスタで返します。

_DSPACE	C : 76h
コール手順	E : 論理ドライブ番号 (00h = カレント, 01h = A: など) A : 動作モード 00h = ドライブの空き容量情報取得 01h = ドライブの総容量情報取得
戻り値	A : エラーコード (00h = エラーなし) HL:DE : 容量 (kB 単位) BC : 余り容量 (byte 単位)

余り容量はドライブの最小割り当て単位が kB 未満のときに 0 以外の値となります。FAT ドライブでは、1 クラスタあたりのセクタ数が 1 で、クラスタ数が奇数のときに 0 以外の値 (特に 512) となります。例えば、あるドライブにおいて 1 クラスタあたりのセクタ数が 0 で未使用クラスタ数が 15 (=7.5kB) のとき、HL = 0000h, DE = 0007h (=7kB) , BC=0100h (512byte) となります。

この容量情報は reduced allocation information モードの影響は受けず、常に正しい値となります。

3.7. Lock/unlock a drive, or get lock state for a drive (_LOCK, 77h)

ドライブのロック及びロックの解除、またはロック状態の確認を行います。ドライブがロックされているとき、ドライブのメディアは交換されないと判断し、メディア交換のステータスを確認しません。それによりメディアへのアクセス速度を改善します。マルチメディアカード等のリムーバブルメディアをメインストレージとして使用するときには有用です。

_LOCK	C : 77h
コール手順	E : 物理ドライブ番号 (00h = A:, 01h = B: など) A : 動作モード 00h = ロック状態の取得 01h = ロック状態の設定 B : ロック/ロック解除の切り替え 00h = ロックを解除する (A = 01h のときのみ) FFh = ロックする (A = 01h のときのみ)
戻り値	A : エラーコード (00h = エラーなし) B : ロック状態 00h = ロックされていない FFh = ロックされている

ロックするためには、有効なファイルシステムがドライブとして割り当てられていなければなりません。つまり、ドライブがアクセス可能でなければなりません。

ドライブがロックされた状態でも、ディスクエラーが発生したときにはロックが解除されます。

ロック及びロックの解除を行ったとき、バッファ内のデータはドライブに書き込まれ、無効となります。また、ディスクパラメータブロックの情報もクリアされ、次のアクセス時に再度読み込みます。

デバイスドライバ上でノンリムーバブルとして扱っているデバイスに対してもロックを設定することは可能ですが、Nextor はそれらのデバイスに対してメディアの交換を確認することはないので意味がありません。

ロック機能は注意して使う必要があります。ロック状態のデバイスに対してロック解除をせずにメディア交換をすると交換前のメディアと交換後のメディアの両方でデータを破壊する恐れがあります。

3.8. Get information about a device driver (_GDRVR, 78h)

デバイスドライバーの情報を返します。このファンクションは MSX-DOS 1 モードでも動作します。

_GDRVR	C : 78h
コール手順	A : インデックス番号 (01h ~) もしくは 00h (スロット番号とセグメント番号のペアで指定するとき) D : スロット番号 (A = 00h のときのみ) E : セグメント番号もしくは FFh (Nextor の MSX-DOS 1 カーネル内のとき) (A = 00h のときのみ) HL : 64byte のバッファのアドレス
戻り値	A : エラーコード (00h = エラーなし) HL で指定したアドレスにある 64byte のバッファにドライバー情報が入る

デバイスドライバーはインデックス番号もしくはスロット番号+セグメント番号のペアで指定します。デバイスドライバーをインデックス番号で指定するときは、Aレジスタにインデックス番号をセットします。インデックス番号は1から始まります。デバイスのあるスロット番号とセグメント番号はその他のデバイスドライバー情報と一緒にデータバッファに入れられて返ってきます。これはどのデバイスドライバーが使われているかを確認するのに有用です。

もし確認したいデバイスのスロット番号とセグメント番号を既に知っているときは、それらの番号をDレジスタとEレジスタにセットし、Aレジスタには00hをセットします。このときもスロット番号とセグメント番号はその他のデバイスドライバー情報と一緒にデータバッファに入れられて返ってきます。

指定したインデックス番号にデバイスドライバーが割り当てられていなかったとき、指定したスロット番号とセグメント番号のペアにデバイスドライバーがないときには、IDRVR エラー (0B6h) が返ります。事前にくいつのデバイスドライバーがシステムにあるかを知ることはできません。そのため、すべてのデバイスドライバーを見つけるためにはこのファンクションをインデックス番号1から数を増やしながら、IDRVR エラー (0B6h) が返るまで繰り返し実行する必要があります。

デバイスドライバー情報フォーマット (オフセットは HL+xxh を示す)

オフセット	内容
00h	デバイスドライバーのロット番号
01h	デバイスドライバーのセグメント番号 (Nextor や Nextor の MSX-DOS 1 カーネル内のデバイスドライバーは FFh)
02h	起動時に割り当てられた論理ドライブ数
03h	起動時に割り当てられたドライブの一番若い番号 (A: = 00h, B: = 01h など) 割り当てられていないときは使いません。
04h	bit 7 : 0b = Nextor の MSX-DOS 1 モード用デバイスドライバー (Nextor の MSX-DOS 1 カーネル内のとき) 1b = Nextor 用ドライバー bit 6 - 0 : 未使用、常に 0b bit 0 : 0b = Device-based のデバイスドライバー 1b = Drive-based のデバイスドライバー
05h	デバイスドライバーのメジャーバージョン番号
06h	デバイスドライバーのセカンダリバージョン番号
07h	デバイスドライバーのレビジョン番号
08 ~ 27h	デバイスドライバー名、左詰で残りは 20h で埋める
28 ~ 3Fh	予約 (現在は 00h)

Drive-based のデバイスドライバー / MSX-DOS 1 モード用デバイスドライバー

MSX-DOS のドライバー仕様に沿っており、1つのドライブレターに対して1つのデバイスドライバーが対応しています。デバイスドライバーがドライブレターの割り当てやパーティション管理を行います。

Device-based のデバイスドライバー

Device-based のデバイスドライバーはドライブ単位ではなく、デバイス単位で動きます。デバイスドライバーの中には“Read sector X of unit N” というようなルーチンはなく、“Return information of device X” や “Read raw data from device X” というようなデバイス単位で動くルーチンがあります。Device-based のデバイスドライバーは最大7個のデバイスを管理でき、それぞれのデバイスは1~7個の論理ユニットを持つことができます。

3.9. Get information about a drive letter (_GDLI, 79h)

ドライブレターの情報を返します。このファンクションは MSX-DOS 1 モードでも動作します。

_GDLI	C : 79h
コール手順	A : 物理ドライブ番号 (00h = A:, 01h = B: など) HL : 64byte のバッファのアドレス
戻り値	A : エラーコード (00h = エラーなし) HL で指定したアドレスにある 64byte のバッファに、ドライブレターの情報が入る

もし物理ドライブ番号 (= ドライブレター数) がシステムでサポートしている数を超えているときは、.IDRV エラー (0DBh) が返ります。ドライブレターが Nextor で指定されていて、かつデバイスドライバーが割り当てられていないときはエラーとなりませんが、ドライブレターの情報は空白となります。ドライブステータスをチェックする必要があります。

“First device sector number” は、そのドライブにおける先頭セクタ番号です。通常はパーティションの先頭セクタで、パーティションが区切られていないときはセクタ番号 0 となります。この値が 0 かどうかで Device-based のデバイスかどうかを判断してはいけません。+01h のスロット番号と+02h のセグメント番号を使って _GDRVR (78h) をコールして得られる戻り値で判断してください。

ドライブレター情報フォーマット (オフセットは HL+xxh を示す)

オフセット	内容
00h	Drive status 0 = 未割り当て 1 = Nextor または Nextor の MSX-DOS 1 モードのストレージデバイスとして割り当て 2 = 未使用 3 = 未使用 4 = RAM ディスクに割り当て (他のフィールドは 00h)
01h	デバイスドライバーのスロット番号
02h	デバイスドライバーのセグメント番号 (Nextor や Nextor の MSX-DOS 1 カーネル内のデバイスドライバーは FFh)
03h	このデバイスドライバーで扱うドライブ数 (Drive-based のデバイスドライバーのみ、Device-based のデバイスドライバーでは FFh)
04h	デバイス番号 (Device-based のデバイスドライバーのみ、Drive-based のデバイスドライバーもしくは MSX-DOS1 モード用のデバイスドライバーでは 00h)
05h	論理ユニット番号 (Device-based のデバイスドライバーのみ、Drive-based のデバイスドライバーもしくは MSX-DOS1 モード用のデバイスドライバーでは 00h)
06 ~ 09h	First device sector number (Device-based のデバイスのみ、Drive-based のデバイスもしくは MSX-DOS1 モード用のデバイスでは 00h)
0A ~ 3Fh	予約 (現在は 00h)

3.10. Get information about a device partition (_GPART, 7Ah)

パーティションの情報を返します。このファンクションは MSX-DOS 1 モードでも動作します。

_GPART	C : 7Ah
コール手順	A : スロット番号 B : セグメント番号もしくは FFh (Nextor の MSX-DOS 1 カーネル内のとき) D : デバイス番号 E : 論理ユニット番号 H : プライマリパーティション番号 (01 ~ 04h) L : 拡張パーティション番号 (プライマリパーティションでは 00h)
戻り値	A : エラーコード (00h = エラーなし) B : パーティションタイプ (指定されたパーティションが存在しないときは 00h) HL:DE : パーティションの先頭セクタ番号

このファンクションは Device-based のデバイスドライバーに対してのみ使用できます。存在しないデバイスドライバー、Drive-based のデバイスドライバーもしくは MSX-DOS 1 モード用のデバイスドライバーが A レジスタと B レジスタで指定されたときは、IDRVR エラー (0B6h) が返ります。もし指定したデバイスとロジカルユニットの両方もしくは一方がデバイスドライバーに存在しないときは、IDEVL エラー (0B5h) が返ります。

ストレージデバイスは通常、パーティションに分割され、それぞれの独立したパーティション内では連続したセクタを使用しています。このファンクションは各パーティションの先頭セクタ番号を見つけることができます。_MAPDRV (7Ch) でそれを使ってドライブレターを割り当て、ファイルシステムにアクセスできるようにします。

パーティションタイプはそのパーティションで使われているファイルシステムの情報を示します。パーティションタイプは以下のいずれかになります。

- 00h : None (パーティションタイプが存在しない)
- 01h : FAT12
- 04h : FAT16 (32MB 以下のとき、廃止)
- 05h : Extended (下記参照)
- 06h : FAT16
- 0Eh : FAT16X (現在の Nextor では未対応です)

実際にはもっと多くのパーティションタイプが存在していますが、Nextor では扱えないためリストから除外しています。

1 つのデバイスは 1~4 の 4 個までのプライマリパーティションを持つことができます。4 個より多くのパーティションを持つためには、プライマリパーティション 2 のパーティションタイプを 05h (Extended) にします。パーティションタイプ 05h (Extended) のパーティションは中に多くの拡張パーティションを持つことができ、その数は無制限です。拡張パーティションのパーティションタイプも 05h (Extended) となります。プライマリパーティション 2 のパーティションタイプが 05h (Extended) のとき、プライマリパーティション 3 及び 4 は存在しません。

デバイス内に存在するすべてのパーティションを数え上げるためには、下記の手順が必要となります。

1. パーティション 1-0 をチェック (プライマリパーティション 1 の拡張パーティション番号 0)
2. パーティション 2-0 をチェックし、存在してパーティションタイプが 05h (Extended) のとき、パーティション 2-1, 2-2, 2-3, ... とパーティションタイプが 00h で返ってくるまで続けます。
3. パーティション 2-0 が存在しないか、パーティションタイプが 05h (Extended) でなかったとき、パーティション 3-0 及び 4-0 をチェックします。

デバイス内にパーティションが存在しないこともあります。そのときでも有効なファイルシステムを持ち、セクタ 0 にドライブレターを割り当てることが可能なものがあります。フロッピーディスクや小容量のデバイスがそれにあたります。

パーティションがドライブレターに割り当てられたとき、パーティションの先頭セクタから実際のファイルシステムを確認します。Nextor がファイルシステムを確認するのにパーティションタイプ情報は使いません。

Nextor がパーティションを確認するときにデバイスを読みます。エラー (例えば “Not Ready” 等) が発生したときは、エラーコードを返します。標準のエラー処理ルーチン (DISKVE (F323h)) やユーザー定義 (_DEFER (64h)) のディスクエラー処理ルーチンは呼び出されません。

指定したパーティションがデバイスに存在しないとき (例えばプライマリパーティション番号に 5 以上を指定したり、拡張パーティションが存在しないパーティションで 1 以上の拡張パーティション番号を指定したりしたとき)、B = 00h, A = .IPART (B4h) が返ります。

3.11. Call a routine in a device driver (_CDRVR, 7Bh)

デバイスドライバー内のルーチンを直接呼び出します。このファンクションは MSX-DOS 1 モードでも動作します。

_CDRVR	C : 7Bh
コール手順	A : スロット番号 B : セグメント番号もしくは FFh (Nextor の MSX-DOS 1 カーネル内のとき) DE : ルーチンのアドレス HL : ルーチンに渡す引数 (8byte) のあるバッファのアドレス
戻り値	A : エラーコード (00h = エラーなし) BC, DE, HL : ルーチンの戻り値 IX : ルーチンの戻り値 AF

Device-based, Drive-based や MSX-DOS 1 モード用かに関わらず、すべてのデバイスドライバーのルーチン呼び出すことができますが、Device-based のドライバーがデバイスを数えたり (DEV_INFO (4163h) や LUN_INFO (4169h))、デバイスのセクタに直接アクセスしたり (DEV_RW (4160h)) することを意図しています。例えば、デバイスのパーティションツールを作るときに使います。呼び出し可能なルーチンの一覧やその詳細については、“Nextor 2.0 Driver Development Guide” を参照してください。

ルーチンへの引数として AF, BC, DE, HL レジスタの値を 8byte のバッファに用意し、HL レジスタにバッファのアドレスを入れます。バッファには、F, A, C, B, E, D, L, H の順で並べます。ルーチンの戻り値としてのレジスタの値は、AF レジスタの値が IX レジスタに入る以外は、各々のレジスタに直接入れられます。

いくつかのルーチンではユーザーが用意したバッファに対して読み書きすることができます。ただし、この方法では二つの制限があります。

- ① バッファはプライマリマッパースロットになければならない
- ② バッファの一部またはすべてがページ1にあってはならない

これらの制限はルーチンに渡す引数をおくバッファには適用されません。ただし、F37Dh を用いて呼び出すときにはいかなるパラメータもページ 1 にはおけません。また、引数をおくバッファはルーチンを実行する前にしか使いませんので、ルーチンからの戻り値をおくためのバッファとして使ってもかまいません。

A レジスタ及び B レジスタで指定されたデバイスドライバーが存在しないときは、IDRVR エラー (0B6h) が返ります。Nextor で追加されたファンクションコール_GDRVR (78h) を使って、存在しているドライバーを確認してください。

3.12. Map a drive letter to a drive and device (_MAPDRV, 7Ch)

Device-based デバイスドライバーのデバイス番号、論理ユニット番号及びその先頭セクタの組み合わせ (mapping data) で指定するパーティションに対してドライブレターを割り当てます。また、起動時の状態に戻したり、すべてのドライブの割り当てを解除したりすることも出来ます。このファンクションは MSX-DOS 1 モードでも動作しますが、後述するいくつかの制限があります。

_MAPDRV	C : 7Ch
コール手順	A : 物理ドライブ番号 (00h = A:, 01h = B: など) B : 動作モード 00h = ドライブレターの割り当てを解除 01h = ドライブレターの割り当てを起動時の状態に戻す 02h = ドライブレターを mapping data で指定するパーティションに割り当てる HL : mapping data (8byte) のあるバッファのアドレス (B = 02h のときのみ)
戻り値	A : エラーコード (00h = エラーなし)

B = 00h ではドライブの割り当てを解除します。その時点からドライブへのアクセスが不可能になり、すべてのアクセスは “Invalid drive” エラーとなります。既に割り当てが解除されているときは何も起きず、エラーにはなりません。

B = 01h では指定したドライブレターの割り当ては起動時の状態に戻ります。起動時に割り当てられていなかったり、Drive-based のデバイスドライバーで割り当てられていたり、MSX-DOS1 モード用のドライバーで割り当てられていたときは、その状態に戻ります。起動時に Device-based のデバイスドライバーに割り当てられていたときは、起動時の割り当てルールに従って再割り当てされますが、ルートディレクトリに仮想 FD 機能用のファイル “NEXTOR.DAT” が存在するときは仮想 FD 機能が優先されます。割り当て機能については、“Nextor 2.0 User Manual” に記載されています。再割り当ての際に起動時と同じ割り当て状態に戻るかは、同じデバイスドライバーで管理するリムーバルデバイスの状態や、他のドライブの状態によります。再割り当てが失敗したとき (例えばデバイスやパーティションがなかったとき)、実行前の状態にかかわらず、ドライブレターの割り当ては解除され、.IDRVR エラー (0B6h) が返ります。

B = 02h では HL レジスタで指定したアドレスにあるバッファ内の mapping data にしたがって、ドライブレターが割り当てられます。この方法では Device-based のデバイスドライバーに対してどのドライブレターでも自由に割り当てられます。起動時にドライブレターが割り当てられていなかったり、異なるドライブレターに割り当てられていたりしたときでも可能です。バッファ内の mapping data の構造は以下の通りです。

オフセット	内容
00h	スロット番号
01h	デバイスドライバーのセグメント番号 (Nextor や Nextor の MSX-DOS 1 カーネル内のデバイスドライバーは FFh)
02h	デバイス番号
03h	論理ユニット番号
04 ~ 07h	先頭セクタの番号

指定したデバイスドライバーが存在しない、または Device-based のデバイスドライバーでないときは、.IDRVR エラー

(0B6h) が返ります。デバイスドライバー内にデバイスや論理ユニットが存在しないときは IDEVL エラー (0B5h) が返ります。このとき、実行前の状態から変更されません。

H:ドライブを指定したときに RAM ディスクが存在しているときは、.RAMDX エラー (0BCh) が返ります。

先頭セクタの番号はデバイスにおける物理セクタで、ドライブの論理セクタ 0 として使われます。通常、パーティションの先頭セクタで、_GPART (7Ah) で確認できます。ただし、指定したセクタにファイルシステムがあって、かつ Nextor で認識できるかどうかはチェックしません。有効なファイルシステムが存在しなくても _MAPDRV は成功しますが、次にアクセスしたときに “Not a DOS disk” エラーとなります。

また、メディアが入っていない状態のリムーバブルデバイスにドライブレターを割り当てることも可能です。この場合でも _MAPDRV は成功しますが、次にアクセスしたときに “Disk offline” エラーとなります。

同じ組み合わせ (ドライバー、デバイス、論理ユニット、先頭セクタの番号) に対して二つのドライブレターを割り当てることはできません。各ドライブのセクタバッファが同期せず、データの不整合が起きることを防ぐためです。このとき、.IDEVL エラー (0B5h) が返ります。すでに割り当てられているドライブレターを変更したいときは、先に古いドライブレターの割り当てを解除する必要があります。

また、Drive-based のデバイスドライバーや MSX-DOS 1 モード用のデバイスドライバーに対して、ドライブレターを指定して割り当てることは出来ません。

ドライブレターの割り当てを変更する前に、オープンされているすべてのファイルハンドルはこのファンクションによって閉じられます。これは _CLOSE (45h) と同じです。そのため、バッファに問題があって、デバイスへの書き込み時にエラーが発生したときは、ディスクエラーとなります。

MSX-DOS 1 モードで使った時、下記の制限が発生します。

- ① 指定するドライブレターは起動時に Device-based のデバイスドライバーに割り当てられていなければなりません。起動時に割り当てられていない、または Drive-based のデバイスドライバーや MSX-DOS 1 モード用のデバイスドライバーに割り当てられているドライブレターは変更できません。
- ② 新しいドライブレターの割り当てはパーティションとデバイスの両方もしくは片方が異なるものが指定できますが、スロットは起動時に割り当てられたドライブと同じでなければなりません。Nextor の kernel が 1 個だけ存在しているときには問題になりません。

これらの制限は Nextor のアーキテクチャによるものです。

3.13. Enable or disable the Z80 access mode for a driver (`_Z80MODE`, 7Dh)

このファンクションは MSX-DOS 用デバイスドライバー (MSX-DOS カーネル内のデバイスドライバー) に対して Z80 アクセスモードの有効/無効を設定します。このファンクションは MSX turbo R でのみ有効です。MSX, MSX2, MSX2+ では .IDRVR エラー (0B6h) となります。

<code>_Z80MODE</code>	C: 7Dh
コール手順	A: デバイスドライバーのロット番号 B: 動作モード 00h = 現在の Z80 アクセスモードの状態をとる 01h = Z80 アクセスモードを設定する D: 有効/無効の切り替え 00h = Z80 アクセスモードを無効にする (B = 01h のときのみ) FFh = Z80 アクセスモードを設定する (B = 01h のときのみ)
戻り値	A: エラーコード (00h = エラーなし) D: 現在の Z80 アクセスモードの状態 00h = Z80 アクセスモードが無効 FFh = Z80 アクセスモードが有効

Z80 アクセスモードが対象のデバイスドライバーに対して有効になっているとき、そのデバイスドライバーで管理するドライブにアクセスする前に Nextor が CPU を Z80 に切り替えます。無効のときは CPU の切り替えを行いませんので、R800 のままドライブにアクセスします。

起動時にはすべての MSX-DOS 用デバイスドライバーに対して Z80 アクセスモードが有効になっています。いくつかの古いデバイスは R800 モードではアクセスできず、それらのデバイスから起動出来ないためです。起動後にはこのファンクションを使って Z80 アクセスモードを無効にしたり、後で再度有効にしたりすることができます。

Z80 アクセスモードは設定するデバイスドライバーで管理しているすべてのドライブに対して有効/無効を切り替えます。個々のドライブに対して、Z80 アクセスモードの有効/無効を設定することはできません。

Z80 アクセスモードは MSX-DOS 用デバイスドライバーに対してのみ有効です。Nextor は Drive-based のデバイスドライバーや Device-based のデバイスドライバーに対しては CPU を切り替えません。

4. 新しいエラーコード

Nextor で追加された新しい機能を扱うときのエラーをハンドルするため、新しいエラーコードを定義しました。MSX-DOS 1 モードにおいても Nextor の新しい機能がサポートされている場合は、同じエラーコードが返ります。

新しいエラーコードの一覧と詳細は下記の通りです。

- Invalid device driver (.IDRVR, 0B6h)

Device-based のデバイスドライバーに対する操作がリクエストされたときに、指定されたデバイスドライバーが存在しないか、タイプが異なる。(例えば、MSX-DOS または Driver-based のデバイスドライバーに対して Device-based のデバイスドライバー用の操作がリクエストされたような場合。)

- Invalid device or LUN (.IDEVL, 0B5h)

Device-based のデバイスドライバーを使うデバイスに対する操作がリクエストされたときに、指定されたデバイスがデバイスドライバーに存在しない、または指定された論理ユニットが指定されたデバイスに存在しない。

- Invalid partition number (.IPART, 0B4h)

デバイスのパーティション情報が要求されたが、指定されたパーティションがデバイスに存在しない。

- Partition is already in use (.PUSED, 0B3h)

デバイスドライバー、デバイス、論理ユニット、先頭セクタの組み合わせに対してドライブレターの割り当てを試みたが、同じ組み合わせに対して別のドライブレターが既に割り当てられているとき。

5. 拡張マッパーサポートルーチン

MSX-DOS 2 のマッパーサポートルーチンにセグメントより小さい容量 (1~16,378byte) のメモリブロックを TPA セグメント (起動時に割り当てられたセグメント) もしくは新しく割り当てたセグメントに確保するルーチンを追加しました。これらのルーチンは、マッパーサポートルーチンのジャンプテーブルを拡張して使えるようにしています。ジャンプテーブルのアドレスはマッパーサポートの拡張 BIOS コールを使用して取得します。新しいルーチンの名前とアドレスは以下の通りです。

+30h : BKL_ALLOC

+33h : BLK_FREE

どちらのルーチンもその時点でページ 2 に割り当てられた RAM に対して実行されます。TPA セグメントやメモリマッパーで割り当てたセグメント、あるいはメモリマッパーでない RAM であっても、読み書きできるメモリがページ 2 にあれば使用可能です。しかしながら、ここでは説明の都合上、ページ 2 に割り当てたセグメントと仮定しています。

5.1. BLK_ALLOC : Allocate a memory block (+30h)

BLK_ALLOC	ジャンプテーブルの+30h
コール手順	HL : 確保したい容量 (0001 ~ 3FFAh)
戻り値	成功時 HL : 確保したメモリブロックの先頭アドレス (常にページ 2 のアドレス) A : 00h, Z フラグをセット 失敗時 HL : 0000h A : .NORAM (0DEh) エラー, Z フラグをリセット

このルーチンは指定されたサイズのメモリブロックをその時点でページ 2 に割り当てられているセグメントから確保します。成功したときは確保したメモリブロックの先頭アドレスを HL に入れて返します。失敗したときは、“Not enough memory” エラーとなります。複数回実行することによって複数のメモリブロックを確保することができますが、使用するページ 2 のセグメントは最初に BLK_FREE ルーチンを HL = 0000h で実行して初期化していなければなりません。そうでないとき、動作は保証出来ません。

5.2. BLK_FREE : Free a memory block (+33h)

BLK_FREE	ジャンプテーブルの+33h
コール手順	HL : BLK_ALLOC ルーチンで得られたメモリブロックの先頭アドレス (解放するメモリブロックの先頭アドレス) 0000h 指定時はセグメントを初期化

このルーチンはその時点でページ 2 に割り当てられているセグメントにあるメモリブロックを解放します。HL で指定するアドレスは、同じセグメントで実行した BLK_ALLOC ルーチン実行時に得られたアドレスでなければなりません。そうでないとき、動作は保証できません。解放されたメモリは、次の割り当てに利用可能です。

メモリブロックの確保及び解放の状態は、セグメント内に保存されています。Nextor のワークエリアにメモリ割り当ての状態を保存することはありません。そのため、割り当てたすべてのメモリブロックが不要となったときには、それぞれのメモリブロックを個別に解放する必要はありません。メモリブロックを解放せずに、セグメント内のメモリを別のデータで書き換えたり、セグメントを解放したり、TPA セグメントであればアプリケーションを直接終了したりしてもかまいません。

HL = 0000h でコールしたときは、その時点でページ 2 に割り当てられているメモリをブロックメモリ確保のために初期化します。そのセグメントでメモリブロックを確保する前に必ず一度実行しなければなりません。また、既にメモリブロックを複数確保しているときに、すべてのメモリブロックを解放する最速の方法です。

6. その他の特徴

この章では Nextor がサポートするその他の特長について解説します。

6.1. `_STROUT (09h)` におけるエスケープシーケンス “ESC-Y” の不具合を修正

`_STROUT (09h)` は文字列の文字が通常の「コンソール出力」ファンクションを使用して出力され、`'$'` (`24h`) で終了します。エスケープシーケンスの中には、画面上の任意の位置にカーソルを移動させるものがあります。例えば、カーソルを $(x, y) = (10, 5)$ に移動したいときは `ESC Y 5h+20h Ah(=10)+20h`、つまり `1Bh 59h 25h 2Ah` となります。 x は横方向、 y は縦方向のカーソル位置で、オフセットとして `20h` を加算した値を使用します。

バグはこのエスケープシーケンスで x または y に `4` が指定されたときに発生します。 x または y が `4` の場合、オフセットの `20h` が加算された値が `24h` となり、`_STROUT (09h)` における文字列出力の終了条件に一致してしまいます。そのため、MSX-DOS が文字列の終了マークと誤認識して出力を打ち切ります。

このバグは Nextor で修正され、ESC-Y は問題なく使用できます。

6.2. Nextor のバージョン変更

いくつかの MSX-DOS アプリケーションは `MSX-DOS2.SYS` のバージョンをチェックしていることはよく知られています。(Nextor では `NEXTOR.SYS` が `MSX-DOS2.SYS` にあたります。) そして、もし `MSX-DOS2.SYS` のバージョンがアプリケーションが必要とするバージョンより小さいとき、特に `2.20` より小さいときに動作を中断することがあります。このようなアプリケーションは、Nextor でも `NEXTOR.SYS` のバージョンが `2.0` のため問題となります。

この問題の回避方法として、`NEXTOR.SYS` の `2.0 beta 2` からは、`_DOSVER (6Fh)` の戻り値を RAM に保管しておく、簡単に書き換えられるようにしています。`NSYSVER.COM` を使用することによって簡単に書き換えられます。`NSYSVER.COM` の詳細については “*Nextor 2.0 User Manual*” を参照してください。もし、プログラムで書き換えたいときは下記の手順となります。

1. MSX-DOS 上で、ページ 0 が TPA セグメントのときに、`0001h` に保存してあるアドレス (ジャンプテーブルへのアドレス) を読み込みます。
2. 1. で得られた値に `32h` を加算します。(ジャンプテーブルのサイズが `32h` なので、ジャンプテーブルの次のアドレスになります。)
3. 2. で得られたアドレスが 16bit の数字を保管する場所です。`_DOSVER (6Fh)` 実行時に DE レジスタに入れて返す値が保管されています。

例えば、`NEXTOR.SYS` のバージョンを `2.31` に書き換えるには下記のコードとなります。

```
LD IX, (0001h)
LD BC, 32h
ADD IX, BC
LD (IX), 31h
LD (IX+1), 02h
```

MSX-DOS2.SYS や NEXTOR.SYS のバージョンは BCD コードになっていることに注意してください。

もちろん、この変更は一時的なものです。書き換え後であっても、BASIC に行って “CALL SYSTEM” 等で Nextor に戻ったり、リセットしたりして NEXTOR.SYS が再度読み込まれると元の正しい値に戻ってしまいます。

この機能は NEXTOR.SYS の 2.0 beta 2 以降においてのみ有効です。

7. 変更履歴

この章では、Nextor のバージョン毎の変更履歴について解説しますが、アプリケーション開発の観点から意味のある変更点についてのみとなります。一般的な仕様の変更点については “Nextor 2.0 User Manual” を参照してください。また、デバイスドライバー開発に関する変更点については “Nextor 2.0 Driver Development Guide” を参照してください。

7.1. v2.0.4

- 環境変数 “ERRLANG” が存在し、値が “EN” のとき、ファンクションコール_EXPLAIN (66h) は漢字モードかどうかにかかわらず、英語のエラーメッセージを表示します。

7.2. v2.0.3

- ファンクションコール_DPARM (31h) のバグを修正。クラスターカウントが 4095 ~ 4080 のとき、実際のファイルシステムをチェックすることなくオフセット+1Ch に 01h (FAT16) を返してしまう不具合を修正。
- ファンクションコール_ALLOC (1Bh) に zero allocation information モードの説明を追加。

7.3. v2.0 final

このバージョンではアプリケーション開発に関わる変更はありません。

7.4. v2.0 Beta 2

- ファンクションコール_DOSVER (6Fh) で得られる NEXTOR.SYS のバージョンを書き換えられるようにしました。

7.5. v2.0 Beta 1

このバージョンではアプリケーション開発に関わる変更はありません。

7.6. v2.0 Alpha 2b

このバージョンではアプリケーション開発に関わる変更はありません。

7.7. v2.0 Alpha 2

- ファンクションコール_DOSVER (6Fh) を拡張し、MSX-DOS 1 モードでも Nextor を判別できるようにしました。
- _GDRVR (78h) , _GDLI (79h) , _GPART (7Ah) , _CDRVR (7Bh) を MSX-DOS 1 モードに対応させました。

8. 問い合わせ

Nextor 及び関連ツールの最新版は Konamiman's MSX page からダウンロード出来ます。

<http://www.konamiman.com>

バグ報告や提案については下記宛先までご連絡ください。

konamiman@konamiman.com