

## WZPC/ST BIOS 仕様(作成中)

## BIOS コール

ソフトウェア割込み INT 命令とハードマクロ命令の 0 番を BIOS コールとして使う。  
アセンブラでは BIOSCALL として定義される。

^BIOSCALL n
-------------

n: BIOS ファンクション番号

## BIOS ファンクション

番号	内容
0x00	BIOS Info
0x01	BIOS Display Mode
0x02	BIOS Text
0x03	BIOS Timer
0x04	BIOS KeyInput
0x05	BIOS Random
0x06	BIOS Graphic

## BIOSCALL 0x00(BIOS Info)

A レジスタ	内容
0x00	レジスタ AB に 16bit 数字のバージョンが入る。必ず成功。

## BIOSCALL 0x01(BIOS Display Mode)

## ^BIOSCALL 0x01

画面モード。0xF0~0xFF は画面モードの設定変更

A レジスタ	内容
0x00	画面 OFF
0x01	画面 ON
0x04	表示画面を 0 面に切替

0x05	表示画面を 1 面に切替
0x06	バンクメモリを 0 面に切替
0x07	バンクメモリを 1 面に切替
0x08	バンクメモリと表示画面の 0,1 を入替
0x21	256x32 モノクロ VRAM:0x7400 (TINY 32KB) (暫定)
0x22	240x32 モノクロ VRAM:0xFA00,256x48 (HALF 128KB or TINY 64KB)
0x52	320x240、16 色、VRAM 先頭アドレス 0x6A00

戻り値: A レジスタ 成功 0、失敗 0 以外

### BIOSCALL 0x02(BIOS Text)

BIOS コールはハードマクロ 0 番の命令を実行することでできます。

テキスト表示のオペランドは 0x02

テキスト表示のパラメータを格納したメモリのアドレスをレジスタ A,B に設定。

LD A, &bios_param.H
LD B, &bios_param.L
^BIOSCALL 0x02

### UTF-8 のコード

00 00 00 ~ 00 00 7F	0xxx xxxx
00 00 80 ~ 00 07 FF	110x xxxx 10xx xxxx
00 08 00 ~ 00 FF FF	1110 xxxx 10xx xxxx 10xx xxxx
01 00 00 ~ 10 FF FF	1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx

テキスト表示 BIOS のパラメータ

先頭アドレスから次のように順番で格納される。1 番以降はタイプ 1 のフォーマット

0	<p>ファンクション番号</p> <p>0: SET CURSOR、カーソルを移動</p> <p>1: SET COLOR、色を設定</p> <p>2: PUTC、UTF-8 の一文字表示</p> <p>3: PUTS、UTF-8 の文字列表示</p> <p>4: SET FONT(0 : 5x7 1: 半角 8x16、全角 16x16)</p> <p>5: SET VRAM、VRAM アドレス</p>
---	---

戻り値: A レジスタ 成功 0、失敗 0 以外

### SET CURSOR

カーソル位置の設定、原点(0,0)

1	X 値
2	Y 値

#### SET COLOR

1	フォアグラウンド COLOR 値(0～15)
2	バックグラウンド COLOR 値(0～15)

#### PUTC

カーソル位置に UTF-8 の 1 文字を表示

1～3	UTF-8 の文字コード
-----	--------------

#### PUTS

カーソル位置から UTF-8 の 0x00 で終端された文字列の表示

1～	UTF-8 の文字列。終端は 0x00
----	---------------------

#### SET FONT

1	フォント番号 0 or 1
---	---------------

#### SET VRAM

1	VRAM アドレス上位バイト
---	----------------

#### BIOSCALL 0x03(BIOS Timer)

1ms 単位でタイマーはインクリメントされる。実装によって 2 バイト～8 バイトがある

レジスタ	動作内容
A=0x00	タイマーの値をリセット
A=0x01	タイマーの値をレジスタに取得、メモリ[B]に 2 バイトのタイマーを書き込む成功した場合 A レジスタは 0。失敗は 0 以外。
A=0x02	タイマーの値をレジスタに取得、メモリ[B]に 4 バイトのタイマーを書き込む成功した場合 A レジスタは 0。失敗は 0 以外。
A=0x03	タイマーの値をレジスタに取得、メモリ[B]に 8 バイトのタイマーを書き込む成功した場合 A レジスタは 0。失敗は 0 以外。(実装依存)

BIOSCALL 0x04(BIOS KeyInput)

A レジスタはキーインプットのモード。A=0x01 は同時キー入力不可。

レジスタ	動作内容
A=0x01 B=0x01	C 言語の kbhit() と同等の機能 A レジスタ 0: キー無し 1: キー有り 0xFF: 失敗
A=0x01 B=0x02	C 言語の getch() と同等の機能 キューから 1 バイトを取得

BIOSCALL 0x05(BIOS Random)

高速な疑似乱数

レジスタ	動作内容
A=0x01	B レジスタに 1 バイトの乱数シードを設定する成功 A レジスタ 0x00、失敗はそれ以外
A=0x02	A レジスタに 0~255 までの乱数が返る

BIOSCALL 0x06(BIOS Graphic)

パラメータを格納したメモリのアドレスをレジスタ A,B に設定。パラメータの全バイトのアドレスの上位バイトは同一でなければならない。パラメータ中のポインタで指定されたビットマップデータは任意のアドレスで良い。

LD A, &bios_param.H LD B, &bios_param.L ^BIOSCALL 0x06
--

BIOS のパラメータ

先頭アドレスから次のように順番で格納される。1 番以降はタイプ 1 のフォーマット

0	ファンクション番号 0: PSET、C 色の点を座標(X,Y)に描く 1: HLINE、C 色の水平線を座標(X,Y)から Z ドット描く 2: VLINE、C 色の垂直線を座標(X,Y)から Z ドット描く 3: CLS、全画面クリア 4: BWRITE 座標(x,y)を左上とした 幅 W ビット高さ H ビットのビットマップを描く x,W は 2 ドット単位、y,H は 1 ドット単位
---	---

	5:BWRITEOR カラー番号 8 は透明であること以外は BITMAP と同じ 6:BREAD 指定領域のビットマップデータをメモリ上にリードする
--	--

戻り値: A レジスタ 成功 0、失敗 0 以外

#### PSET

1-2	X 値(リトルエンディアン)
3	Y 値
4	カラーC

#### LINEH

1-2	X 値(リトルエンディアン)
3	Y 値
4	カラーC
5	Z ドット

#### LINEV

1-2	X 値(リトルエンディアン)
3	Y 値
4	カラーC
5	Z ドット

#### CLS

パラメータ無し

#### BITMAP 系

1	x 値 (2 ドット単位)
2	y 値 (1 ドット単位)
3	W 幅(2 ドット単位)
4	H 縦(1 ドット単位)
5	ビットマップデータのアドレス下位バイト
6	ビットマップデータのアドレス上位バイト

BIOS メモリマップ

BIOS で使っているメモリ

プログラムメモリ

0000h～0005h	BIOS スタートルーチンへ分岐
0006h～001Fh	(空)
0020h～0023h	BIOS の INT0 ルーチン分岐(割込み 0 番)
0024h～0027h	BIOS の INT2 ルーチン分岐
0028h～002Bh	BIOS の INT4 ルーチン分岐
002Ch～002Fh	BIOS の INT6 ルーチン分岐
0030h～003Dh	(空) <b>アプリで利用可能なものは INT8～14</b>
003Eh～0041h	BIOS の INT15 ルーチン分岐(割込み 1 番)
0042h～01FFh	(空)
0200h～020Fh	BIOS コールのハードマクロ 0 番
0210h～DFFFh	(空)
E000h～FFFFh	BIOS 領域(ワードアドレスは 7000h～7FFFh) リセット後、BIOS 初期化終了後、OFFSET2 を実行してから 0x001C 番地に分岐

アプリで使える合計値: 約 55KB

WZ-660 のハードウェアは 320x240 16 色の画面モードのみなので画面モードの設定は必要ないが、シミュレータでは起動時の画面モードを設定する必要がある。画面モードの 1 バイトの値(0x52)を 0006h 番地のメモリに書き込んでおく。実際のメモリにはコピーされない。シミュレータのみが、その値を使える。

.DEAD
LD A,0x52     # 画面モード 320x240 16 色

データメモリ

0000h～007Fh	OFFSET0、割込み処理、BIOS 処理で使うゼロページ
0080h～00FFh	基本 OS のゼロページ
0100h～01FFh	OFFSET2、3。アプリやライブラリで使うゼロページ
0200h～XXXXh	BIOS ワークエリア(アプリが利用する BIOS コールで最大容量確保)
XXXXh～67FFh	アプリのデータ
6800h～FFFFh	VRAM、320x240 16 色

現時点の BIOS は BIOS ワークエリアは、存在しない。

BIOS 割込み

BIOS では次のように使用している

INT 命令	内容
0	外部割込み 0 番。優先外部割込み。通信処理など緊急を要する割込み。他の割込み処理をしている場合は、その処理が終了するまで待つため、割込み処理は可能な限り短い時間で終了させなければならない。
2	異常終了(ABEND)
4-6	フォント圧縮のため(暫定的に使用)
15	外部割込み 1 番。キーボードなど、ほとんどの外部割込みを処理する。