■解説 (VISION Vol. 8, 1996)

視覚研究のための実験環境：IBM-PC互換機における VESA拡張BIOSを利用したグラフィックスプログラミング

蘆田 宏

ATR人間情報通信研究所

〒619-02 京都府相楽郡精華町光台 2-2

1．はじめに

1.1 視覚実験におけるIBM-PC互換機の利用

近年，いわゆるIBM-PC互換機は日本でも安価に入手できるようになった．これらのコンピュータは8 bitから24 bit (32 bit) の色数を表示できる上かなり高い解像度の表示まで行えるため，視覚実験用の提示装置としては最も手軽なものの一つと言える．また，日本で大きなシェアを持つNEC製のマシンでのプログラム環境に慣れている人であれば，比較的スムースに移行できる．しかしながら，実際にはプログラミングにおける情報，特に視覚研究者が気軽に読めるような資料は十分とは言えず，PCは実験においては必ずしも活用されていないかも知れない．本稿では，現在のIBM-PC互換機に標準となっているSVGAグラフィック機構について，視覚実験に必要と思われる最低限の機能の解説を行うとともに，助けになると思われる文献およびソフトウェアを紹介したい．

本稿はMS-DOSおよび類似OSにおけるプログラミングの経験を前提としている．特に，読者はC言語および8086アセンブラについて，熟練している必要はないが，ひととおりの知識があるものと期待される．蛇足ながら，そうでない方々には，むしろMacintoshにおけるShell/Macglib環境も検討されることをお勧めしたい．著者がPCを使い始めた頃と違い，Macintoshは価格，性能両面で大きく向上した．Shellはきわめて易しいプログラミング環境を提供してくれる上，複数モニタの同時利用など，実験環境としてMacintoshの方が優れている面も多い．

1.2 VESA拡張BIOSとは

IBM-PCの基本設計はオープンな仕様のもとにほとんどの機能を部品化するという思想に基づいており，CRT提示機構も，本体とは別に独自の進化をとげてきた．現在主流のSVGAは，IBMによるVGA (Video Graphics Array) 規格をサードパーティ各社が独自に拡張した結果生まれたものであり，当然，拡張部分におけるハードウェア的な互換性はない．VESA拡張BIOS (VBE) はこの点を補うために標準化団体VESAによって策定されたBIOSレベルのAPIであり，SVGAの機能のうち，ごく基本的な部分は共通のプログラムで扱えるようになった．現状では描画アクセラレータなどの高度な機能は使えないが，視覚実験において実際に必要な機能はごく限られており，VBEの範囲内だけでもかなりのことができる．なお，VBEは通常SVGAボード上にROMとして供給されるが，場合によっては専用のTSRやデバイスドライバの組み込みが必要かもしれない．

VBEの規格は徐々に進化しており，現段階ではバージョン1.2が主流である．1.1では後述の関数のうち6までしか使えないが，基本的な描画は同様にできる．現在32ビット対応を含むバージョン2が策定されている他，アクセラレータ機能についての拡張も検討されている様子であるが，Windows95と共にDOS上でのプログラミングの意味そのものが変化するた

め，今後それが普及する保証はない．本稿でも，バージョン1.2までの機能についてのみ扱う．

2．基本的なスペック

種類： ハード／ソフト

目的： SVGA機能を備えたIBM-PC互換機上でのグラフィックプログラム作成．

使用環境： MS-DOS (PC-DOS) が動作するIBM-PC互換機＋SVGAグラフィック装置（現在入手可能なIBM-PC互換機の大半が該当すると思われる），DOSプログラム作成用のC言語（Microsoft, Borland等）．

著者の環境： ATTLA AS100/VLB (Micronics JX30WB mother, DX4-100 CPU, 8MB RAM) + NANAO HA50VL (SVGA), PC-DOS6.3/V, Borland C++ 3.1J.

3．VESA拡張BIOS

3.1 概要

VESA拡張BIOSの一覧を表1に示す．これらを使うには，AXレジスタに04Fhを，ALレジスタに機能番号を入れてInt 10hを呼び出す．特に重要な関数の使用法は付録にあげるが，他については参考文献を参照していただきたい．

3.2 モードの設定

画面の解像度と色数をSVGAモードとして設定する．実験に有用と思われる主なモードを表2にあげる．他に16bit色モードもある．

モードの設定には，AXレジスタに4F01hを入れ，Int 10hを呼び出す．実際にはあらかじめ機能番号0により，そのモードが使用可能か調べる必要があるが，自明な場合は省略できる．使用可能なモードはほぼVRAM容量とDACの性能で決定される．モード101hはほとんどの場合に使用可能と思われる．

表1　VESA拡張BIOS (VBE) 一覧

機能番号	機能
0	SVGA情報の取得
1	SVGAモード情報の取得
2	ビデオモードの設定
3	現在のモードを取得
4	ビデオ状態の保存／復帰
5	ビデオメモリウインドウ制御
6	スキャンライン長設定
7	表示開始アドレス設定
8	DAC情報の設定／取得

3.3 点の描画

点の描画は全ての描画の基本であり，あとは一般的なアルゴリズムの適用で実現できる．

16色以下のモードでは，ビデオメモリは複雑なビットプレーン構造を持つが，256色 (8 bit) 以上のモードでは，VRAMはリニアなアドレスを持つ1枚のプレーンとして扱われる．8bit色の場合，1byteデータをそのまま書き込めばよいが，24bit色の場合1点に3byte要し，それらはアドレス上に順に書き込めばよい．なお，RGBの並び順はグラフィック制御チップによって異なる可能性がある．

VRAMは，主メモリアドレス上の0A0000hから0AFFFFhまでの64 KBの空間に表れるが，当然これだけでは全てのVRAMにアクセスできないため，バンク切り替えを行う．メモリウインドウの開始アドレスは通常64 KB単位で指定する（アドレス設定の単位はSVGA情報のWinGranularityフィールドで取得）．

点描画のプロセスをここで詳しく説明する余裕はないが，概要としては，(1) 描画位置と解像度，色数から点のリニアアドレスを計算，(2) バンク位置を計算し，機能番号5でウインドウに表れるメモリバンクを設定，(3) メモリウインドウ上に点データを書き込む，という順になる．付録2を参照していただきたい．

3.4 DAC情報の設定

SVGAにおける8bitカラーモードはイン

表2　主なSVGAモード

モード番号	色数 (bit)	解像度
101h	8	640x480
103h	8	800x600
105h	8	1024x768
112h	24	640x480
115h	24	800x600
118h	24	1024x768

デックスカラーモードであり，実際の色はパレットで設定する．オリジナルのVGAではパレットの値はRGB各下位6 bitのみが有効であった．この制約は，グレースケールを多用する視覚実験ににおいてはしばしば問題となる．しかし，SVGAにおいてはこれが各8 bitに拡張されている場合があり，VBEの機能番号8で設定する．SVGA情報のcapabilitiesフィールドのビット0が1になっていれば拡張可能である．8 bit拡張が使えるグラフィックスボードについては情報が不足しており，現時点で著者が把握しているのはNANAO社のHA50 (VL)のみであるが，後述のUNIVBEを用いると多くの製品で8 bit拡張ができる．細かな注意点として，HA50はcapabilityフィールドの1になっているバイトがUNIVBEと逆である．複数バイトの場合エンディアンの問題も絡んで，どちらが正しいのか，著者には判断がつきかねている．

なお，6 bitパレットの制約はDACの入力が24 bit対応であるかどうかによるものではない．そのため，ダイレクトカラーモードである24 bitモードが使えるのに8 bit拡張パレットは使えないという場合は多い．S3社のSDACはその例であり，結果としてDiamond社のStealth ProやCanopus社のPower Window 864では8 bit拡張が使えない（現在のSDACは変更されている可能性もある）．

3.5 ページ切り替え

視覚実験ではあらかじめ複数の画面に書き込みを行っておき，ページ切り替えで高速に提示する必要がある場合が多い．SVGAにおいても，機能番号7のVBE関数で表示開始アドレスを設定することにより，搭載しているVRAMの範囲内で複数のページを扱うことができる．

ここで，グラフィックスボードによるVESA関数の実装の違いが問題になる．著者が確認した範囲では，S3社のチップを使った製品では表示開始アドレスの設定がうまくいかない．これはバグとも思われるが，むしろ関数の仕様の解釈の違いかもしれない．対策としては，後述のUNIVBEを用いるのが最も簡単である．

3.6 パレットの設定

8 bitカラーモードではパレットを動的に設定することができ，これを用いて運動刺激を作ることができる．パレットの設定には標準のVGA BIOSを用いてもよいが，その実装には各社で微妙な違いがある．特に，パレット設定を自動的に垂直同期信号にあわせる場合とそうでない場合とがある．以下の方法で直接パレットを設定することをお勧めする．

パレットを設定するには，ポート3C8hにパレット番号（色番号）を1 byteで出力した後，ポート3C9hにR, G, Bの順に3 byte出力する．速いマシンではウエイトが必要かもしれない．また，標準では出力値の下位6 bitのみが有効である．

3.7 表示フレーム制御

垂直同期に関しては標準VGAの機能を使う．VGAカラーモードの場合（ほとんどの場合そうであろう），ポート3DAhのビット3（下から4番目）が1になっていると垂直帰線期間中である．

4．関連ソフトウェア

4.1 UNIVBE

入手先：ftp.scitechsoft.com (shareware, $28 + shipping)．simtel20.msdosのミラーサイトやCD-ROMなどからも入手できる．

最新バージョン：5.1a

ファイル名：UVBE51A.ZIP

先にも述べたように，VESA規格に沿っているはずのSVGA BIOSにも各社で実装にばらつきがある．これを解決してくれるのがSciTech Software社のUNIVBEである．UNIVBEはTSRとして既存のVESA BIOSを補完あるいは置換する．このプログラムはVESA関数の仕様にかなり忠実に作られているように思えるため，我々が実験プログラムを作成する際には大きな助けになる．

UNIVBEを用いるもう一つの利点が，8 bit

パレットのサポートである．市販のグラフィックスボードには，ハードウェア的に能力があるにも関わらずVBE関数の8番をきちんと実装していないものが多い．そうした場合でもUNIVBEを用いると8bitグレースケールを実現できる．著者が作者に確認したところでは，UNIVBEが8 bitパレットをサポートできるのは以下のDACを搭載したボードである．例えば，#9社のGXE64, GXE64Proなどが該当するが，これらは既に前世代の製品であり，最新の製品については個別に確認していただきたい．

. AT&T ATT 20c490/1, 20c498
. SGS Thompson STG 1701/2/3
. Texas Intruments TVP 3020, 3025
. BrookTree Bt485
. Sierra SC15021/25/26

4.2 SuperVGA Kit

入手先：ftp.scitechsoft.com (freeware) の他simtel20.msdosミラーサイトやCD-ROM等．
最新バージョン：5.1
ファイル名：SVGAKT51.ZIP

SciTech SoftwareからUNIVBEのテスト用にフリーで配布されているライブラリ＋サンプルプログラムであるが，line等の基本的な描画やページ切り替え，パレット設定の関数があり，視覚実験には十分な機能を持っている．SVGAKitを使うと，これまでに述べてきたVBE自体をほとんど意識することなくプログラミングすることができる．UNIVBEは必須ではないが，使った方が動作が安定する．バージョンはUNIVBEと対応しており，違うバージョンで使うと若干問題が出るかもしれない．最新版では使用法がドキュメント化されている．バージョン4代の旧版でも視覚実験には十分使えるが，メジャーバージョンが変わると関数仕様が大きく変わるため，注意が必要である．

バージョン5.1では，UNIVBE 5.1が持つVBE 2.0の機能に対応して，32 bitのプログラミングが可能である．ドキュメントによるとコンパイラは以下に対応しているが，著者がテストしたのはBC++ 3.1のみである．

Borland C++ 3.1 16 bit
Borland C++ 4.0 16 bit
Borland C++ 4.0 32 bit
Microsoft Visual C++ 1.5 16 bit
Microsoft Visual C++ 1.0 32 bit
Symantec C++ 6.1 16 bit
Symantec C++ 6.1 32 bit
Watcom C++ 10.0 16 bit
Watcom C++ 10.0 32 bit
Metaware High C/C++ 3.2
DJGPP GNU C/C++ 32 bit

4.3 GLIB

入手先：著者に連絡（準フリーウェア）

SVGAKitのバージョン4を参考に著者が独自に開発したSVGAライブラリで，著者の実験では全てこれを用いている．基本的にはフリーウェアとしての公開をめざしているが，著者のテスト環境が限られること，また，線や円の描画ルーチンに市販の書物からの引用を含むことなどから現在のところ公開はしていない．ガンマ補正機能を備えている他，限定的ながらS3社のチップにおけるBitBltなどのアクセラレータ機能を使用できるようになり，運動刺激作成に使用している．

著者の実験計画の都合で現在開発は若干停滞しているが，もし，テストやプログラミング面での協力をしていただける方がおられたらご連絡をいただければ幸いである．特に，line等の描画関数のソースをフリーで書いていただける方がおられたらぜひご協力をお願いしたい．

5．おわりに

本稿では，IBM-PCとSVGAにおけるプログラミングについて簡単に説明してきた．限られたスペースではとても全ての機能を解説できなかったが，後述の参考文献をご覧いただきたい．DOS/V人気にあわせて，日本でもPCに関する技術解説書が数多く出版されるようになった．著者が初めてPCで実験プログラムを作成したのはたかだか3年前の事であるが，洋書を

求めて苦労した当時を思うと既に隔世の感がある．そうした解説書への入門編として本稿を利用していただければ幸いである．なお，著者はコンピュータの専門ではないため，内容に不適切な点もあるかと思われるが，ご容赦とともにご指摘をお願い申しあげたい．

最後に参考となる文献[1-5]を記す．VGAの基本構成を詳しく知りたい場合は文献[1,4]をお勧めする．VBE関数の仕様については文献[2-4]で知ることができるが，特に実際の使用については文献[2]が詳しい．著者はテストしていないが，文献[4]にはSVGA用ライブラリが付属する．文献[1]は線，円等の描画アルゴリズムも記載．キー，マウスなどについても知りたい場合，文献[3]が簡便である．

文　献

1) R. Wilton（SE編集部：訳編）：PC&PS/2ビデオシステムプログラマーズガイド．翔泳社，1989.
2) Bootstrap Project-2, No.5：PCビデオサブシステムの研究．CQ出版社，1993.
3) C. F. Computing：DOS/Vプログラマーズハンドブック．ソフトバンク，1993.
4) J. Sanchez and M. P. Canton（豊田　孝：監訳）：PCグラフィックプログラミング．アスキー出版局，1994.
5) Video Electronics Standards Association：VESA BIOS EXTENTION (VBE) Core Functions Version: 2.0.　SVGAKT51.ZIP付属，1995.

付録1 主なVESA関数の仕様

視覚実験で使うと思われる最小限について，関数の仕様を示す．

呼び出しレジスタをセットした上でINT 10hを呼び出す．全ての機能においてAXレジスタにステータスが返され，それは以下のような意味を持つ．また，リスト中BYTE型は1 byte，WORD型は2 byteを示す．フィールド名は参考文献[2]をもとに著者の判断でつけた．

共通

AXレジスタへの戻り値

AL = 4Fh　関数がサポートされている
AL!= 4Fh　サポートされていない
AH = 0　成功
AH!= 0　失敗

機能0 GET SuperVGA INFO.

呼び出し

AX = 4F00h
ES:DI = VGAInfoBlock（256 byteのバッファ）へのポインタ

戻り値

VGAInfoBlockに以下の情報が入る．

```
typedef struct {
  char  VESAsignature[4];
  WORD  VESAversion;
  char *OEMStringPtr;
  BYTE  Capabilities[4];
  WORD *VideoModePtr;
  WORD  TotalMemory;
  BYTE  Reserved[242];
} VgaInfoBlock_t;
```

機能1 GET SuperVGA MODE INFO.

呼び出し

AX = 4F01h
ES:DI ModeInfoBlock（256 byteのバッファ）へのポインタ

戻り値

ModeInfoBlockに以下の情報が入る．

```
typedef struct {
  WORD ModeAttributes;
  BYTE WinAAttributes;
  BYTE WinBAttributes;
  WORD WinGranularity;
  WORD WinSize;
  WORD WinASegment;
  WORD WinBSegment;
  BYTE (*WinFuncPtr)();
  WORD BytesPerScanLine;
  WORD XResolution;
  WORD YResolution;
  BYTE XCharSize;
  BYTE YCharSize;
  BYTE NumberofPlanes;
  BYTE BitsPerPixel;
  BYTE NumberofBanks;
  BYTE MemoryModel;
  BYTE BankSize;
```

```
  BYTE NumberofImagePages;
  BYTE RedMaskSize;
  BYTE RedMaskPosition;
  BYTE GreenMaskSize;
  BYTE GreenMaskPosition;
  BYTE BlueMaskSize;
  BYTE BlueMaskPosition;
  BYTE ReservedMaskSize;
  BYTE ReservedMaskPosition;
  BYTE DirectScreenInfo;
  BYTE Reserved[216];
} ModeInfoBlock_t;
```

機能 2 GET SuperVGA MODE INFO.

呼び出し

AX = 4F02h

BX = video mode, MSB=1 にすると画面をクリアしない.

機能 5 CPU VIDEO MEMORY CONTROL

呼び出し

AX = 4F05h

BH = 0 : 設定　1 : 取得

BL = 0: Video window A

1: Video window B（通常0でよい）

DX = ウインドウ位置（設定／取得）

機能 7 GET/SET DISPLAY START

呼び出し

AX = 4F07h

BL = 0 : 設定　1 : 取得

BH = 0 (reserved)

CX = 左上の x 座標

DX = 左上の y 座標

機能 8 GET/SET DAC PALETTE CONTROL

呼び出し

AX = 4F08h

BL = 0 : 設定　1 : 取得

BH = 表示ビット数（6か8）

戻り値

BH = 現在の表示色ビット数

付録 2 VBE による点描画プログラム例

8 bit 色モード用点描画の例．画面に赤い水平線がひかれる．掲載のために簡略化してあるが，本来は bpl, pseg, yres を ModeInfoBlock から取得する他，返値のチェックが必要な場所もある．また，高速化の工夫も省略されている．Borland C++ 対応.

```
// sample program for drawing with
VESA BIOS
// (C) H.Ashida 1995/11
#include <conio.h>
typedef unsigned int WORD;
typedef unsigned char BYTE;
int ActivePage=0;

void SetSVGAMode(WORD mode)
// VESA function 2
{
  asm mov  ax, 4f02h
  asm mov  bx, mode
  asm int  10h
}
void putPixel256(WORD x,WORD y,
BYTE color)
{ // assuming WinGranularity=64k
  WORD bpl=640;  //bit per line
  WORD yres=480;
  WORD pseg=0xa000, poff, bank;
  unsigned long addr;

  addr=yres*bpl*ActivePage+y*bpl+x;
  poff=addr;
  bank=addr>>16;
// VESA function 5
  asm mov ax, 4f05h
  asm mov bx, 0
  asm mov dx, bank
  asm int 10h
// writing to VRAM
  asm mov ax, pseg;
  asm mov es, ax
  asm mov bx, poff
  asm mov al, color
  asm mov byte ptr es:[bx], al
}
main()
{
  int x;
  SetSVGAMode(0x101);
  for (x=100; x<540; x++)
    putPixel256(x,100,4);
  getch();
  SetSVGAMode(3);
  return 0;
}
```