

## エポック社 スーパーカセットビジョン ハードウェア解析資料

### 1) ハードウェア構成

スーパーカセットビジョンのハードウェアは、主に以下のチップから構成されます。

CPU      uPD7801G (4MHz)  
ビデオ    EPOCH TV-1 (14.31818MHz)  
サウンド uPD1771C (6MHz)  
メモリ    uPD4016C-2 (2KB SRAM) \*2

uPD7801G は、NEC 製の組み込み用 1 チップ CPU である uCOM87 シリーズの 1 つです。128B の RAM および 4KB の ROM を内蔵しており、また PA, PB, PC の 3 種類の 8 ビット I/O およびシリアル I/O を持っています。

EPOCH TV-1 は、おそらくスーパーカセットビジョン専用に開発された、NEC 製のカスタムチップです。TMS9918 など他のビデオコントローラとは異なる、独自仕様のチップとなっています。

UPD1771C は、NEC 製の組み込み用 1 チップ CPU かと思われますが、詳細は不明です。スーパーカセットビジョンでは、これをサウンド生成用に使用しています。

2KB SRAM \*2 は、CPU ではなく EPOCH TV-1 に接続されています。

### 2) メモリマップ

\$0000-\$0FFF : CPU 内蔵 ROM  
\$2000-\$2FFF : スプライトパターン定義  
\$3000-\$31FF : テキスト/BG 画面 VRAM  
\$3200-\$33FF : スプライト表示情報  
\$3400-\$3403 : EPOCH TV-1 制御レジスタ  
\$3600-        : uPD1771C コマンド送出  
\$8000-\$FF7F : ROM カートリッジ (バンク切り替え可能)  
\$FF80-\$FFFF : CPU 内蔵 RAM

\$8000-\$FF7F の領域については、PC5, PC6 によってバンク切り替えを行うことが可能です。

また BASIC 入門など一部のソフトについては、\$E000-\$FF7F にカートリッジ上の SRAM がマッピングされ、セーブデータの保存やワーク RAM として使用されます。

### 3) I/O

uPD7801G は、PA, PB, PC の 3 種類の 8bit I/O ポートおよびシリアル I/O が存在します。

スーパーカセットビジョンでは、PA (出力) をキーマトリックスのカラム指定に、PB (入力) をキー状態の取得に使用しています。

	PA bit0	bit1	bit2	bit3	bit4	bit5	bit6	bit7
PB bit0	P1-L	P1-D						
PB bit1	P1-U	P1-R						
PB bit2	P1-T1	P1-T2						
PB bit3	P2-L	P2-D						
PB bit4	P2-U	P2-R						
PB bit5	P2-T1	P2-T2						
PB bit6			0	2	4	6	8	CL
PB bit7			1	3	5	7	9	EN

また、PC はビット毎に入出力ポートとして使用しています。

PC bit0	IN	PAUSE
PC bit1	IN	N.C
PC bit2	IN	GND
PC bit3	OUT	uPD1771C リセット?
PC bit4	OUT	N.C
PC bit5	OUT	カートリッジバンク切り替え
PC bit6	OUT	カートリッジバンク切り替え
PC bit7	IN	カートリッジ拡張用 (未使用)

シリアル I/O はカートリッジに接続されていますが、実際には使用されていません。

#### 4) VDC (EPOCH TV-1)

NEC 製のカスタムチップで、64pin の DIP パッケージになっています。パッケージには 8432KX EPOCH TV-1 という型番が印刷されています。ワークとして SRAM 2KB\*2 (uPD4016C-2) が VDC に直結されていますが、VDC 自体にも更に 2KB+ のメモリが内蔵されているようです。

VDC の VBLANK 信号は CPU の INT2 に接続されており、信号の立ち上がり、立ち下り毎に CPU に対して割り込みを発生させます。

##### 4-1) 画面構成

スーパーカセットビジョンの画面は、テキスト/BG 画面とスプライト画面の 2 画面から構成されます。描画の優先順位は、常にスプライト画面 > テキスト/BG 画面のようです。

テキスト/BG 画面は、領域を指定することでテキスト画面とグラフィック画面に分割することができます。またグラフィック画面は、8x8 ドットを 1 画素として画素単位に色を指定することができる低解像度モードと、4x4 ドットを 1 画素とした単色の高解像度モードを排他的に選択することができます。VRAM の容量から判断すると 256x256 の解像度をもちますが、TV に表示される実効解像度は 192x222 程度のようです。

スプライト画面は、16x16 の解像度の単色スプライトを最大で 128 個まで表示することが可能です。(横方向の表示制限は 20 個までとの情報がありますが、自分では検証できておりません) スプライトを X, Y 方向に連結することで 32x16, 16x32, 32x32 の解像度とする機能と、2 つのスプライトを同じ位置に重ねて表示することで 2 色表示する機能を排他利用することができます。また、16x16 のスプライトを上下または左右に分割して片側だけ表示する機能を利用することも可能です。

##### 4-2) テキスト/BG 画面

テキスト/BG 画面の VRAM は、CPU から見ると \$3000-\$31FF の領域に存在します。これは 32x16 文字に相当しますが、1 バイトあたり 8x16 ドットの画素の表示内容が指定されますので、全体としては 256x256 の解像度となります。ただし、TV に表示される実効解像度は 192x222 程度となります。

テキスト/BG 画面は、領域を指定することでテキスト画面とグラフィック画面に分割することができます。領域の指定は、

$XMAX = \$3400 \text{ の bit0-bit3 の値} * 2$

$YMAX = \$3400 \text{ の bit4-bit7 の値}$

として、(0,0)-(XMAX-1,YMAX-1)の領域(文字単位)がテキスト画面、残りがグラフィック画面となります。ただし、\$3400 の bit7 が 1 の場合は、テキスト画面とグラフィック画面の領域が

逆になります。

テキスト画面の領域では、VRAM の内容は文字コードとなります。文字パターンは 8x8 ドットで、パターン情報は VDC に内蔵されているようです。これが 8x16 の領域の上半分に描画されます。表示色は全文字共通で、

表示色 = \$3403 の bit4-bit7 の値

背景色 = \$3403 の bit0-bit3 の値

の単色表示となります。

グラフィック画面は、8x8 ドットを 1 画素として画素単位に色を指定することができる低解像度モードと、4x4 ドットを 1 画素とした単色の高解像度モードを排他的に選択することができます。モードの選択は、

\$3400 の bit0-bit1 の値 = 1 : 高解像度

= 3 : 低解像度

となります。

低解像度モードでは、VRAM の内容は画素の色番号となります。8x16 の領域のうち、上半分の色番号を bit4-7、下半分の色番号を bit0-bit3 で表します。

高解像度モードでは、VRAM の内容は表示パターンを表します。1bit あたり 4x4 の画素に相当するとして、8x16 の領域は以下のように表現されます。

b7,b6

b5,b4

b3,b2

b1,b0

高解像度モードではグラフィック画面全体が単色表示となり、色番号は

表示色 = \$3401 の bit4-bit7

背景色 = \$3401 の bit0-bit3

で指定されます。

#### 4-3) スプライト画面

スプライトは、1 パターンが単色で 16x16 のサイズとなります。これを最大 128 パターン定義することができ、また任意のパターンを最大 128 個まで表示することができます。パターン個数と表示個数が同じになっていますが、各パターン 1 個ずつではなく、同じパターンを複数個表示することも可能です。

なお BIOS 中には、BIOS 内部に記録されている文字パターンをスプライトのパターンにコピーす

るルーチンが用意されていますので、これを用いることで、スプライトによって文字を表示することが容易に可能となっています。

スプライトのパターン定義は、\$2000-\$2FFF の領域で行います。1 パターンあたり 32 バイトを使用しますので、4096 バイトで 128 パターン定義できることになります。

まず 1 バイトが、以下のように 4x2 ドットに相当します。

b7, b6, b5, b4

b3, b2, b1, b0

これを以下のような配列で 32 バイト並べて、16x16 ドットのパターンとなります。

P00, P01, P02, P03

P04, P05, P06, P07

P08, P09, P10, P11

P12, P13, P14, P15

P16, P17, P18, P19

P20, P21, P22, P23

P24, P25, P26, P27

P28, P29, P30, P31

表示するスプライトの情報（表示位置、パターン番号など）は、\$3200-\$33FF の領域で指定します。1 個のスプライトあたり 4 バイトで指定を行いますので、512 バイトで 128 個のスプライトの情報を指定することができます。

以下、1 個あたりの情報を、以下のように ATR0-ATR3 で表します。

$ATR0 = \$3200 + (\text{index} * 4)$

$ATR1 = \$3201 + (\text{index} * 4)$

$ATR2 = \$3202 + (\text{index} * 4)$

$ATR3 = \$3203 + (\text{index} * 4)$

また、スプライトの情報の番号をインデックス番号とします。

スプライトのパターン番号は、ATR3 の bit0-bit6 で指定します。

表示位置は、以下のように 2 ドット単位で指定します。

$X = \text{ATR2 の bit1-bit7}$

$Y = \text{ATR0 の bit1-bit7}$

表示色は、ATR1 の bit0-bit3 で指定します。

複数のスプライトを、1 つのスプライト情報だけで纏めて表示する機能が用意されています。これには、X, Y 方向に各々別のスプライトを連結することで、32x16, 16x32, 32x32 の大きなスプ

ライトとする機能と、2つのスプライトを同じ位置に重ねて表示することで、2色のスプライトとする機能があります。

\$3400のbit5が1のとき、スプライト情報のインデックス番号が32-63, 96-127のものは、2色表示機能を使用することができます。それ以外の場合については、スプライト連結機能を使用することができます。

スプライト連結機能では、ATR2のbit0が1の場合はX方向に、ATR0のbit0が1の場合はY方向にスプライトが連結されます。連結されるスプライトのパターン番号は、ATR3のbit0-bit6で指定される元のパターン番号をN0とすると、右側は(N0 or 1)、下側は(N0 or 8)、右下は(N0 or 9)となります。従って、元のパターン番号のbit0, bit3が0の場合は異なるパターンが連結されますし、1の場合は同じパターンが連結されることとなります。

2色表示機能は、ATR0およびATR2のbit0が共に0の場合は無効、それ以外の場合は有効となります。重ねて表示するスプライトのパターン番号は、ATR3のbit0-bit6で指定される元のパターン番号をN0とすると、以下の式で表されます。

$$N0 \text{ xor } ((\text{ATR2のbit0が1の場合は}8) \text{ or } (\text{ATR0のbit0が1の場合は}1))$$

重ねて表示するスプライトの色番号は、インデックス番号が0-63の場合は、

$$\text{PAIR}[ ] = \{ 0, 15, 12, 13, 10, 11, 8, 9, 6, 7, 4, 5, 2, 3, 1, 1 \}$$

インデックス番号が64-127の場合は、

$$\text{PAIR}[ ] = \{ 0, 1, 8, 11, 2, 3, 10, 9, 4, 5, 12, 13, 6, 7, 14, 15 \}$$

として、ATR1のbit0-bit3で指定される元の色番号がCのとき、PAIR[C]となります。

ATR3のbit7が1の場合、16x16のスプライトを左右に分割して表示する機能が有効となります。

このとき、ATR2のbit0が0の場合は左半分が、1の場合は右半分が表示されます。

更にATR3のbit6が1の場合は、上下にも分割することができます。ATR0のbit0が0の場合は上半分が、1の場合は下半分が表示されます。

なお、右半分、下半分を表示する場合は、表示位置をX,Y方向に-8することで、同じ位置に表示されるようになっています。

ATR1のbit4-bit7で、スプライトを何ライン目から描画を行うかを指定することができます。

指定は2ドット単位で、0-30ラインの範囲で指定することになります。

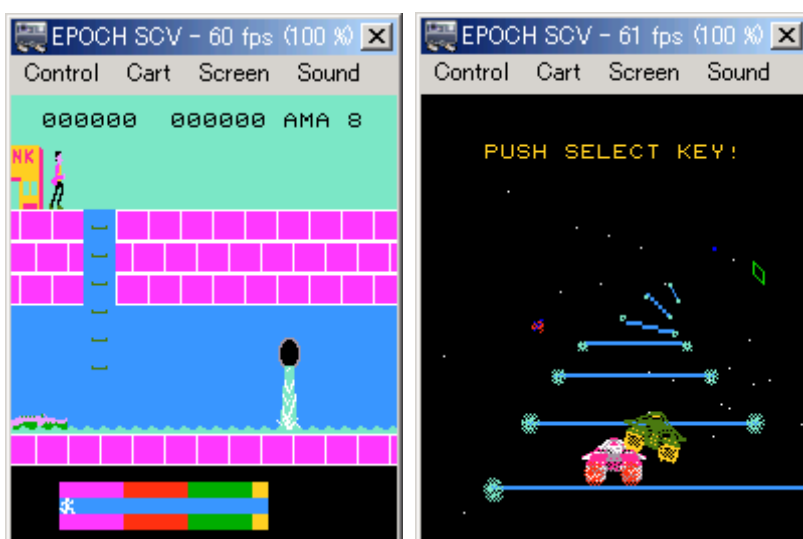
なお、スプライトは1パターンあたり縦16ドットですが、Y方向にスプライトを連結した場合は下側のスプライトまでこの設定が影響されます。

スプライトの情報データを整理すると、以下のようになります。

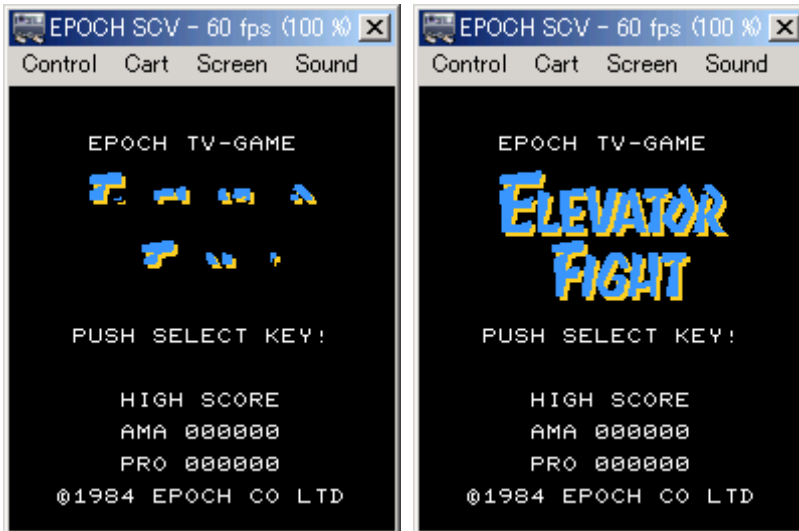
ATR0 : bit0	Y 方向連結/右半分表示
bit1-bit7	表示位置 Y
ATR1 : bit0-bit3	表示色
bit4-bit7	描画開始ライン
ATR2 : bit0	X 方向連結/下半分表示
bit1-bit7	表示位置 X
ATR3 : bit0-bit6	パターン番号
bit6	上下分割有効
bit7	左右分割有効

ATR3 の bit6 はパターン番号と上下分割有効・無効の双方に渡るため、上下分割機能の使用はパターン番号によって制限されます。また 2 色表機能の有効・無効の指定や重ねて表示する色番号の指定は、インデックス番号によって制限されます。このため、スプライト番号の割り振りは、その使い方を意識して行う必要があります。

#### 4-4) 表示サンプル



テキスト/BG 画面の例：ルパン 3 世の得点部分はテキスト、背景の水色（空）や青色（下水道の中）は、低解像度グラフィックによって表示されています。テキスト領域の背景色は空の部分と同じ水色が指定されています。またスタースピーダーのコース部分は、高解像度グラフィックによってラインが描画されています。



スプライトの連結機能の例：スプライトを X, Y 方向に連結することで、32x32 のスプライトでタイトルを表示しています。

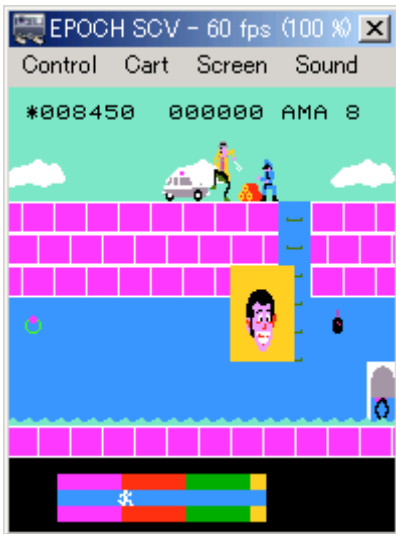


2色表示の例：自機や敵などのキャラクタについては、2色表示のスプライトを更に2つ重ねることによって、各々4色で表示しています。





分割表示の例：熱血カンフーロードでは、左右を向いた顔を1つのスプライトに纏めており、それを左右半分に分割して一方のみを表示しています。ルパン3世では、蝙蝠の羽ばたくアニメーションを、1つのスプライトを上下に分割することで実現しています。



描画開始ラインの例：上方向に閉まる扉に応じて、ルパンが上から消えていきます。

## 5) サウンド (uPD1771C)

NEC 製の汎用 1 チップ CPU ではないかと思われませんが、詳細は不明です。8 種類の音色によるトーン 1 音、ホワイトノイズ風またはノコギリ波風の周期ノイズ 1 音、1bit PCM のうち、任意の 1 音を出力することができます。詳細は不明な点が多いため、以下の資料は仮のものです。

### 5-1) コマンド送付

CPU からはメモリ空間上の\$3600 からアクセスされます。可変長のコマンドを書き込むことによって、それに応じた音が発音されます。コマンド送付の際、最後の 1 バイトを除いてコマンドを書き込む毎に CPU に対して INT1 割り込みを発生します。

### 5-1) リセットコマンド

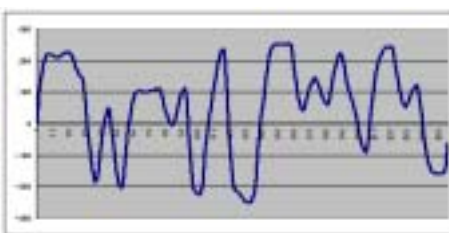
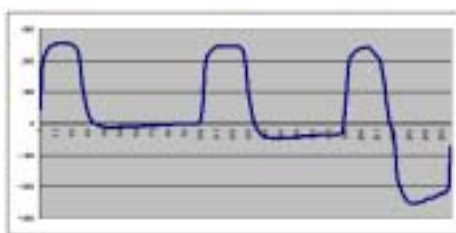
コマンドとして\$00 が送付されると、サウンド出力がリセットされます。

### 5-2) トーンコマンド

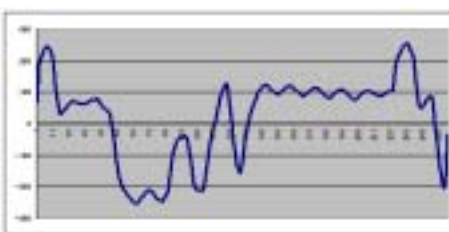
コマンドとして\$02 から始まる 4 バイトが送付されると、トーンが出力されます。

コマンドフォーマットは「\$02, Timbre, Period, Volume」です。

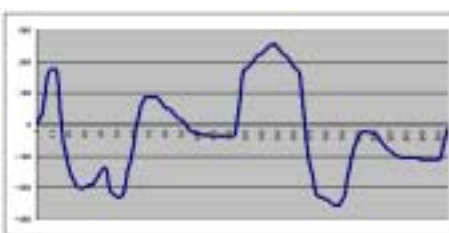
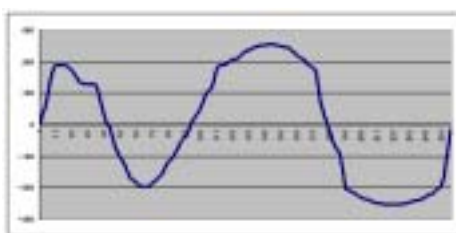
Timbre の bit5-bit7 で、8 種類のプリセット波形から 1 音色を選択します。



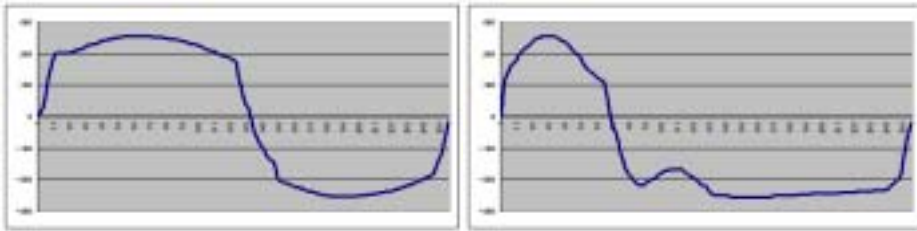
Timbre=0,1



Timbre=2, 3



Timbre=4, 5



Timbre=6, 7

Periodで、波形の出力周期を指定します。ただし、Timbreのbit0-bit4で指定される割合によって、この周期は減衰されます。減衰比は以下のテーブルによります。

```
Detune[32] = {
    100.0, 97.8, 91.7, 91.3, 88.1, 85.2, 82.1, 79.2,
    75.5, 72.4, 69.0, 66.3, 63.0, 60.0, 56.5, 53.6,
    50.7, 47.4, 44.3, 41.4, 38.0, 34.8, 31.6, 28.6,
    25.3, 22.3, 19.0, 15.9, 12.6, 9.4, 6.3, 3.2
}
```

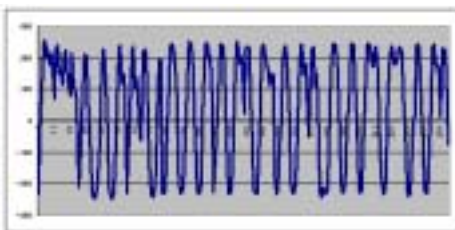
Volumeのbit0-bit4で音量を指定します。

### 5-3) ノイズコマンド

コマンドとして\$01から始まる10バイトが送出されると、ノイズが出力されます。ノイズとしては、ホワイトノイズ風、またはノコギリ波風の周期ノイズが出力されます。

コマンドフォーマットは「\$01, (\$80/\$e0), Period1, Volume1, Period2, Detune2, ?, Volume2, ?, ?」です。ただし、(\$80/\$e0)および?の意味は不明です。

ホワイトノイズ風の周期ノイズを出力する場合は、Period1, Volume1で周期、音量を指定します。波形はプリセットですが、複数のテーブルが用意されている可能性もあります。



Period1, Volume1が0の場合は、ノコギリ波風のノイズが出力されます。こちらの波形は矩形波を基本としていますが、1周期の間で音量を減衰させることで、ノコギリ波風の波形となります。この減衰比はDetune2で指定されます。またPeriod2, Volume2で周期、音量を指定します。



ただし、通常のノコギリ波風ノイズでは説明できない波形が出力される場合があります。例えば、ルパン3世のパトカーの濁ったサイレン音 (Period1 = Volume1 = Volume2 = 0, Detune2 = 5) では、以下のように異なる波形が交互に出力されています。



#### 5-4) 1bit PCM

コマンドとして\$1f から始まる可変長のコマンドを受け取ると、1bit PCM が出力されます。コマンドフォーマットは「\$1f, (\$04/\$06), \$64, \$01, \$00, (波形データ: 可変長), \$fe, \$00」です。波形データはMSB から LSB の順で 1bit ずつ出力され、\$fe, \$00 がくると末端データと見なされコマンドが終了します。

#### 6) 解析の担当について

メモリマップ、I/O などハードウェア全般については、主に Enri 氏に解析して頂きました。VDC の諸機能については、主に Enri 氏に解析して頂きました。ただし、スプライトの連結機能、2色表示機能におけるパターン番号の詳細、スプライトの分割表示機能については武田が解析を行いました。また、スプライトの横方向の描画数制限、スプライト連結機能のフラグの存在については、333 氏によってご示唆頂きました。サウンド機能については、主に武田が解析を行いました。なお資料中に提示した波形は、ノコギリ波を除いて実機よりサンプリングした波形となっています。