SH7268/SH7269 Group

Video Display Controller 4 Driver User's Manual

R01AN0778JJ0100
Rev.1.00
2013.4.18

要旨

本アプリケーションノートは、SH7268/S7269 のビデオディスプレイコントローラ 4(VDC4)ドライバの仕様について説明するものです。

動作確認デバイス

SH7268/SH7269

目次

1. 概要

1.1 環境

本ドライバの開発および動作確認環境を以下に示します。

- CPU
  - SH7269
- 開発環境
  - HEW (SuperH RISC engine microcomputer software integrated development environment) Version 4.09.00.007
  - Renesas SuperH RISC engine Standard Toolchain Version 9.4.1.0
    - SH C/C++ Compiler Version 9.04.01
    - SH Assembler Version 7.01.02
    - SH C/C++ Standard Library Generator Version 3.00.03
    - Optimizing Linkage Editor Version 10.01.00
- 評価ボード
  - SH7269 CPU board (ボード型名: R0K572690C000BR)
  - Optional board for graphic display (ボード型名:M3A-HS64G02)
  - SH7269 VDC4 board (ボード型名: R0K572690B000BR)

1.2 機能

本ドライバのサポートする機能を以下に示します。

Table 1 VDC4 ドライバ機能

入力映像	入力映像規格	• ITU-R BT.656 規格準拠 8bit(27MHz) • ITU-R BT.601 規格準拠 8bit(27MHz,インタレース信号) • ITU-R BT.601 規格準拠 8bit(54MHz,プログレッシブ信号) • RGB888, RGB666 , RGB565 映像
	画質調整	• 水平ノイズリダクション • 黒伸張 • シャープネス / LTI
	録画機能	• YCbCr422, RGB565, RGB888 形式 1/1, 1/2, 1/4, 1/8 フィールドのレートで映像を保存
	スケーリング//回転	• 入力映像をスケーリング/回転 垂直/ 水平 : x1/8 - x8 • 回転：0 度, 90 度, 180 度, 270 度, 水平鏡像
グラフィックス	レイヤ	• 3 面 グラフィックス(1) / ビデオ [1] グラフィックス(2) グラフィックス(3)
	画像形式	• プログレッシブ形式 RGB565, RGB888, ARGB1555, ARGB4444, ARGB8888 CLUT8, CLUT4, CLUT1, YCC422[2]
	重畳機能	• 矩形領域アルファブレンド(フェードイン、フェードアウト) • クロマキー • 画素単位アルファブレンド
パネル出力	サイズ	• QVGA, WQVGA, VGA, WVGA, SVGA
	形式	• プログレッシブ RGB565 / RGB666 / RGB888 • プログレッシブ RGB888(シリアル出力形式)
	補正	• ブライトネス/ コントラスト/ パネルディザ • ガンマ補正

【注】 1. グラフィックス(1) とビデオは排他
2. グラフィックス(1)のみ.

1.3 ファイル構成

本ドライバのファイル構成を以下に示します。

Table 2 ファイル構成

ファイル名	説明
vdc4_api.c	VDC4 ドライバコールを含むソースファイル
vdc4_api.h	VDC4 ドライバコールのプロトタイプ宣言及び定数の定義を含むヘッダファイル
vdc4_clk.c	VDC4 のパネルクロックチェック処理とソフトウェアリセット処理を定義するソースファイル
vdc4_int.c	VDC4 ドライバの割り込みハンドラを含むソースファイル
vdc4_local.h	VDC4 ドライバのみが使用する定義を含むヘッダファイル
vdc4_para.c	VDC4 ドライバコールのパラメータチェック処理を含むソースファイル
vdc4_reg.c	VDC4 モジュールのレジスタ設定処理を含むソースファイル
vdc4_user.h	コンパイルオプション用ヘッダファイル

また、本ドライバを使用する際には下表に示す外部ヘッダファイルが必要となります。

Table 3 外部ファイルの依存関係

ファイル名	説明
typedefine.h	基本型の typedef 宣言定義を含むヘッダファイル
iodefine.h	IO 定義を含むヘッダファイル

1.4 プログラムサイズとセクション

VDC4 ドライバで使用するプログラムサイズとセクションを Table 4 に記載する。

Table 4 プログラムサイズとセクション

"Renesas SuperH RISC engine Standard Toolchain 9.4.1.0"
"Speed & size optimization enabled"

Type	セクション	サイズ[byte]	説明
ROM	P_VDC4	10.3K (14.4K)	プログラム領域
	C_VDC4	0.9K (1.0K)	定数領域
	D_VDC4	40 (40)	初期化データ領域
RAM	B_VDC4	204 (204)	未初期化データ領域

【注】 プログラムサイズは、VRAM と入力ビデオバッファのサイズは含みません。
パラメータのチェックが定義されている場合、括弧内のサイズになります。

1.5 レイヤ構成

本ドライバは VDC4 の持つ 3 つのレイヤをリソースとして 3 つのグラフィックスサーフェスと 1 つのビデオサーフェスを生成することができます。グラフィックス(1)サーフェスとビデオサーフェスはリソースとして共にレイヤ 1 を使用するため、同時に生成することはできません。

本ドライバはサーフェスを管理することで、VDC4 モジュールの機能を実装しています。

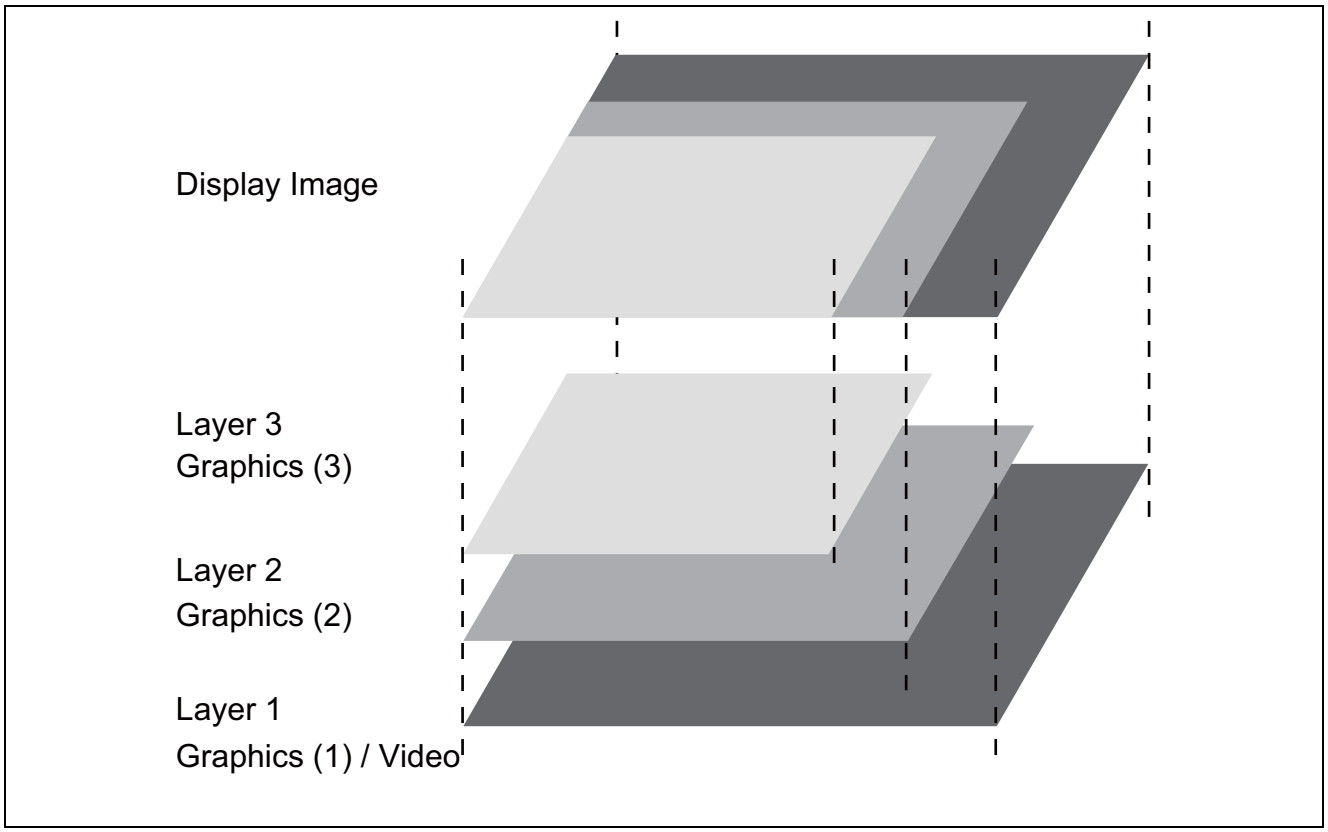
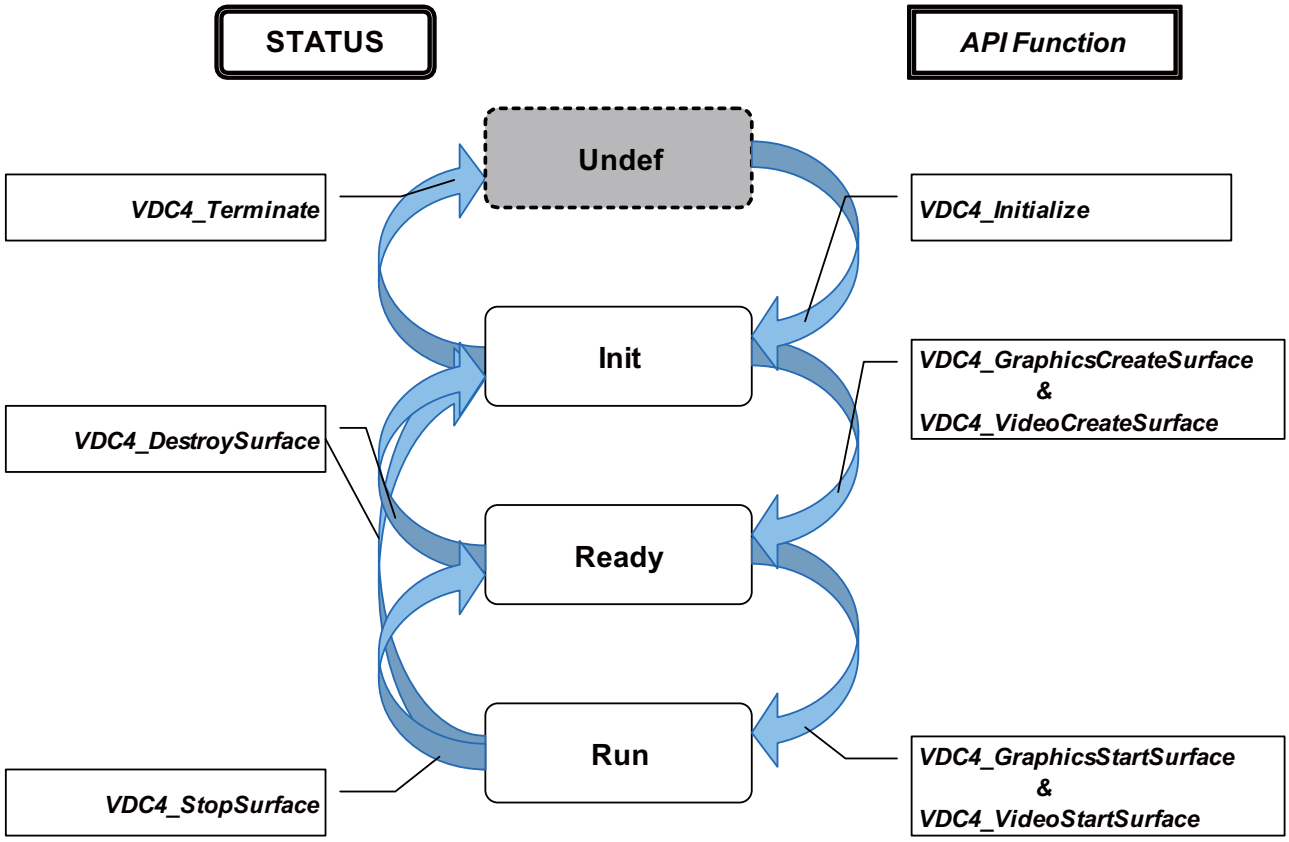


Figure 1 レイヤ構成

1.6 状態遷移

本ドライバではサーフェス毎に状態を定義します。各サーフェスの状態は特定の API を使用することで変化します。API の使用とサーフェス状態遷移の関係を以下に示します。



【注】 各サーフェスの状態は独立していますが、VDC4_Initialize と VDC4_Terminate の呼び出しは全てのサーフェスに対して影響します。

Figure 2 VDC4 ドライバ状態遷移

1.7 割り込みハンドラ

本ドライバは割り込みハンドラを実装しています。割り込みハンドラを以下に示します。

Table 5 割り込みハンドラ

割り込み要因	割り込みベクタ		割り込みハンドラ
	番号	アドレス	
VI_VSYNC スケーリング前の垂直同期信号	171	0x000002AC ~ 0x000002AF	void vdc4_Int_VI_VSYNC(void) ;
LO_VSYNC スケーリング後の垂直同期信号	172	0x000002B0 ~ 0x000002B3	void vdc4_Int_LO_VSYNC(void) ;
VSYNCERR スケーリングの垂直同期信号の欠落信号	173	0x000002B4 ~ 0x000002B7	void vdc4_Int_VSYNCERR(void) ;
VLINE パネル出力指定ライン信号	174	0x000002B8 ~ 0x000002BB	void vdc4_Int_VLINE(void) ;
VFIELD 録画機能のフィールド終了信号	175	0x000002BC ~ 0x000002BF	void vdc4_Int_VFIELD(void) ;
VBUFERR1 フレームバッファ 書き込みオーバフロー信号	176	0x000002C0 ~ 0x000002C3	void vdc4_Int_VBUFERR1(void) ;
VBUFERR2 グラフィックス(1)フレームバッファ 読み出しアンダフロー信号	177	0x000002C4 ~ 0x000002C7	void vdc4_Int_VBUFERR2(void) ;
VBUFERR3 グラフィックス(2)フレームバッファ 読み出しアンダフロー信号	178	0x000002C8 ~ 0x000002CB	void vdc4_Int_VBUFERR3(void) ;
VBUFERR4 グラフィックス(3)フレームバッファ 読み出しアンダフロー信号	179	0x000002CC ~ 0x000002CF	void vdc4_Int_VBUFERR4(void) ;

ユーザがVDC4 の割り込み処理を利用する場合、Table 5の関数を割り込みハンドラとして登録する必要があります。割り込みハンドラの登録機能を実装しているOS等を使用する場合は、その機能を利用することでTable 5の関数を登録することができます。そうでない場合は、ユーザが Table 5に示された関数をベクタテーブルへ登録してください。

```
1  /* Vector talbe example */
2  /* Register the interrupt handler in the vector table */
3  void (*VectorTable[])( void ) = {
4     …
5     vdc4_Int_VI_VSYNC,
6     vdc4_Int_LO_VSYNC,
7     vdc4_Int_VSYNCERR,
8     vdc4_Int_VLINE,
9     vdc4_Int_VFIELD,
10    vdc4_Int_VBUFERR1,
11    vdc4_Int_VBUFERR2,
12    vdc4_Int_VBUFERR3,
13    vdc4_Int_VBUFERR4,
14    …
15 } ;
```

1.8 コンパイラスイッチ

本ドライバでは"vdc4_user.h"ファイルにおいてコンパイラスイッチが定義されています。

1.8.1 パラメータチェック

"_VDC4_PARAMETER_CHECK"の定義を有効にすると、APIのコール時に引数のチェックを行います。パラメータチェックの結果、エラーがある場合はエラーコードを返します。エラーコードについては「2.2エラー」と「2.3API 関数」の項を参照してください。

1.8.2 割り込みハンドラの定義

本ドライバには割り込みハンドラ用の関数が用意されています(「1.7割り込みハンドラ」及び Table 5参照)。このハンドラ用の関数を、レジスタの保証などを行う割り込み関数とするためには"_VDC4_DEFINE_INTHDL"の定義を有効にする必要があります。"_VDC4_DEFINE_INTHDL"の定義を有効にすると、"#pragma interrupt"宣言により、ハンドラ用の関数が、割り込み関数としてコンパイラに処理されます。

OS 等の機能を利用して割り込みハンドラ用の関数を登録する場合には、"_VDC4_DEFINE_INTHDL"の定義を無効にしてください。

1.9 制限事項

1.9.1 予約語

本ドライバでは他のプログラムと区別する為、関数や変数名などのシンボル名称にプレフィックス"VDC4"を付加しています。大文字、小文字を問わず"VDC4"から始まるシンボルは使用しないでください。

1.9.2 レジスタ更新

VDC4 では多くのレジスタが、Vsync の立ち上がりのタイミングで設定の更新を反映します。よって、値が設定されてから反映までに最大で 1Vsync 時間掛かります。

1.9.3 再入可能性

本ドライバの API は再入可能ではありません。本ドライバの API を複数のタスクや割り込み処理から非同期に呼び出した場合、予期せぬ動作をする可能性があります。ドライバコールの呼び出し元やタイミングについては注意してください。

2. API

2.1 共通定義

2.1.1 Typedef

本ドライバでは以下に示したtypedef宣言を使用します。これらのtypedef宣言は"typedefine.h"にて定義されています(「1.3ファイル構成」参照)。

Table 6 typedef 宣言

Typedef	Type
_SBYTE	signed char
_UBYTE	unsigned char
_SWORD	signed short
_UWORD	unsigned short
_SINT	signed int
_UINT	unsigned int
_SDWORD	signed long
_UDWORD	unsigned long
_SQWORD	signed long long
_UQWORD	unsigned long long

2.1.2 列挙型の定義

vdc4_OnOff は ON と OFF を表す列挙型です。

```
typedef enum {
    VDC4_OFF = 0,
    VDC4_ON = 1
} vdc4_OnOff ;
```

列挙定数	値	説明
VDC4_OFF	0	OFF
VDC4_ON	1	ON

vdc4_Edge は信号のエッジを表す列挙型です。

```
typedef enum {
    VDC4_EDGE_RISING = 0,
    VDC4_EDGE_FALLING = 1
} vdc4_Edge ;
```

列挙定数	値	説明
VDC4_EDGE_RISING	0	立ち上がりエッジ
VDC4_EDGE_FALLING	1	立ち下がりエッジ

vdc4_LayerID はサーフェスの ID 番号や数を表す列挙型です。

```
typedef enum {
    VDC4_SURFACE_GRAPHICS_1 = 0,
    VDC4_SURFACE_GRAPHICS_2 = 1,
    VDC4_SURFACE_GRAPHICS_3 = 2,
    VDC4_SURFACE_GRAPHICS_NUM = 3,
```

```
    VDC4_SURFACE_VIDEO_1 = 3,
    VDC4_SURFACE_VIDEO_NUM = 1,
    VDC4_SURFACE_NUM = 4
} vdc4_LayerID ;
```

列挙定数	値	説明
VDC4_SURFACE_GRAPHICS_1	0	グラフィックス(1) サーフェス ID
VDC4_SURFACE_GRAPHICS_2	1	グラフィックス(2) サーフェス ID
VDC4_SURFACE_GRAPHICS_3	2	グラフィックス(3) サーフェス ID
VDC4_SURFACE_GRAPHICS_NUM	3	グラフィックスサーフェス数
VDC4_SURFACE_VIDEO_1	3	ビデオサーフェス ID
VDC4_SURFACE_VIDEO_NUM	1	ビデオサーフェス数
VDC4_SURFACE_NUM	4	サーフェス数

2.1.3 矩形構造体

本ドライバには様々な矩形で表されるタイミングデータが存在します。そこで Figure 3のような矩形を表すデータ型を構造体「vdc4_AreaRect」として定義します。

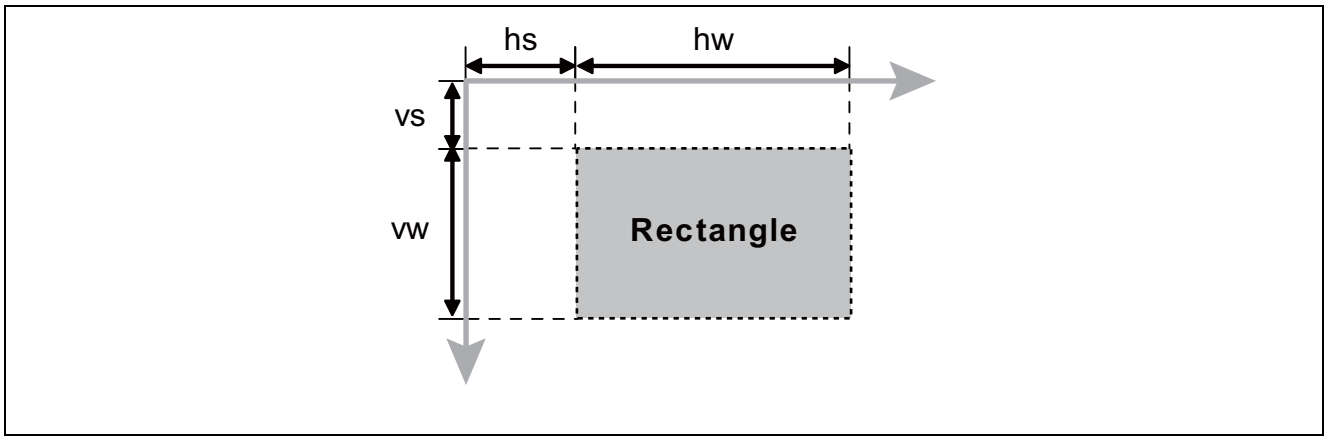


Figure 3 矩形データ

vdc4_AreaRect 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD hs ;
    _UWORD hw ;
    _UWORD vs ;
    _UWORD vw ;
} vdc4_AreaRect ;
```

型 メンバ名	説明
_UWORD hs	水平開始位置 [pixel clock cycles]
_UWORD hw	水平幅 [pixel clock cycles]
_UWORD vs	垂直開始位置 [lines]
_UWORD vw	垂直幅(高さ) [lines]

2.2 エラー

API 関数は、戻り値としてエラーコードを返します。エラーコードの一覧を以下に示します。

Table 7 エラーコード一覧

エラーコード	値	説明
VDC4_ERR_NONE	0	正常終了
VDC4_ERR_SURFACE_BAD	0x2000	不正なサーフェスに対して処理を実施
VDC4_ERR_SURFACE_STATUS	0x2001	サーフェスの状態に合わない処理を実施
VDC4_ERR_PARAM_RANGE	0x4001	パラメータに範囲外の値を設定
VDC4_ERR_PARAM_UNDEF	0x4002	必要なパラメータに NULL を設定
VDC4_ERR_PARAM_INVALID	0x4003	無効なパラメータを設定
VDC4_ERR_PARAM_OTHERS	0x4000	その他のパラメータエラー
VDC4_ERR_SYSTEM_PNLCLK	0x8000	パネルクロック異常検出

パラメータエラーはパラメータチェックのコンパイルスイッチを有効にした場合のみ発生します。

2.3 API 関数

Table 8 は、各API関数が呼び出し可能なサーフェスの状態を表します。

Table 8 状態表

API 関数	サーフェス	状態			
		Undef	Init	Ready	Run
VDC4_Initialize	すべてのサーフェス	OK	NG	NG	NG
VDC4_Terminate	すべてのサーフェス	NG	OK	NG	NG
VDC4_GraphicsCreateSurface	ビデオ	NG	NG	NG	NG
	グラフィックス(1)/(2)/(3)		OK		
VDC4_VideoCreateSurface	ビデオ	NG	OK	NG	NG
	グラフィックス(1)/(2)/(3)		NG		
VDC4_DestroySurface	すべてのサーフェス	NG	NG	OK	OK
VDC4_RegistCallbackFunc	すべてのサーフェス	NG	OK		
VDC4_GraphicsStartSurface	ビデオ	NG	NG	NG	NG
	グラフィックス(1)/(2)/(3)			OK	
VDC4_VideoStartSurface	ビデオ	NG	NG	OK	NG
	グラフィックス(1)/(2)/(3)			NG	
VDC4_GraphicsChangeParam	ビデオ	NG	NG	NG	NG
	グラフィックス(1)/(2)/(3)				OK
VDC4_VideoChangeParam	ビデオ	NG	NG	NG	OK
	グラフィックス(1)/(2)/(3)				NG
VDC4_StopSurface	すべてのサーフェス	NG	NG	NG	OK
VDC4_ImageColorMatrix	ビデオ	NG	NG	OK	NG
	グラフィックス(1)			OK	
	グラフィックス(2)/(3)			NG	
VDC4_VideoNoiseReduction	ビデオ	NG	NG	OK	NG
	グラフィックス(1)/(2)/(3)			NG	
VDC4_ImageEnhancement	ビデオ	NG	NG	OK	OK
	グラフィックス(1)			OK	OK
	グラフィックス(2)/(3)			NG	NG
VDC4_GraphicsAlphaBlending	ビデオ	NG	NG	NG	NG
	グラフィックス(1)			NG	NG
	グラフィックス(2)/(3)			OK	OK
VDC4_GraphicsChromakey	ビデオ	NG	NG	NG	NG
	グラフィックス(1)			NG	NG
	グラフィックス(2)/(3)			OK	OK
VDC4_GraphicsSetCLUT	ビデオ	NG	NG	NG	NG
	グラフィックス(1)/(2)/(3)			OK	OK
VDC4_DisplayCalibration	すべてのサーフェス	NG	OK		
VDC4_DisplaySetGamma CorrectionTable	すべてのサーフェス	NG	OK		
VDC4_SwitchVsync	すべてのサーフェス	NG	OK		
VDC4_CheckClock	すべてのサーフェス	OK	NG		
VDC4_Reset	すべてのサーフェス	OK	NG		

2.3.1 VDC4_Initialize

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_Initialize(const vdc4_InitAttr *attr, void (*init_func)(_UDWORD), _UDWORD user_num) ;	
引数	• [in]const vdc4_InitAttr *attr	初期化パラメータ
	• [in]void (*init_func)(_UDWORD)	ユーザ定義関数のポインタ
	• [in]_UDWORD user_num	ユーザ定義番号
戻り値	• vdc4_ErrorCode	エラーコード
	VDC4_ERR_NONE	正常終了
	VDC4_ERR_PARAM_RANGE	パラメータに範囲外の値を設定
	VDC4_ERR_PARAM_UNDEF	必要なパラメータに NULL を設定
	VDC4_ERR_PARAM_INVALID	無効なパラメータを設定
	VDC4_ERR_PARAM_OTHERS	その他のエラー

概要

本関数では以下の処理を行います。

- ユーザ定義関数の実行
- ドライバの RAM の初期化
- VDC4 モジュールの初期化(レジスタの初期設定)
- LCD パネル制御の設定
- パネルクロックの動作許可設定
- VDC4 モジュールの割り込みの許可設定
- 全てのサーフェスの状態を"Init"に遷移

ドライバの初期化処理に先立ち、`init_func`で指定されたユーザ定義関数を呼び出します。ユーザ定義関数の処理内容については「3.1VDC4_Initializeのユーザ定義関数例」を参考にしてください。

ユーザ定義関数は必ずしも指定する必要はありませんが、その場合本関数の呼び出し前に以下の処理を行ってください。

- VDC4 モジュールへのクロック供給
- VDC4 に関係する割り込みの優先度の設定
- VDC4 に関係するポートの設定
- その他、LCD 表示やビデオ入力に必要な環境固有の設定

引数の設定

型 引数名	入出力	説明
vdc4_InitAttr * attr	in	初期化パラメータ NULL を設定しないで下さい。
void (*init_func)(_UDWORD)	in	ユーザ定義関数のポインタ ユーザ定義関数が設定された場合、user_num が引数として渡されます。必要な場合、ユーザ定義関数を実装して下さい。 書式 void Init_Func(_UDWORD User_Num) ; 引数 • [in]_UDWORD ユーザ定義番号 User_Num 戻り値 • void 概要 ユーザによって実装された処理が実行されます。
_UDWORD user_num	in	ユーザ定義番号 このパラメータは、ユーザ定義関数の引数として使用されます。ユーザ定義関数ポインタが NULL の場合使用しません。

vdc4_InitAttr 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_VsyncSigSel Vsel ;
    vdc4_SyncCtrl *sync ;
    vdc4_PanelClockSel panel_icksel ;
    vdc4_PanelClkDCDR panel_dcdr ;
    _UWORD tcon_half ;
    _UWORD tcon_offset ;
    vdc4_Edge outcnt_lcd_edge ;
    vdc4_LcdTconTim *outctrl[ VDC4_LCD_TCONSIG_NUM ] ;
    vdc4_LcdOutput *settings ;
    _UDWORD IntEnable_1 ;
    _UDWORD IntEnable_2 ;
} vdc4_InitAttr ;
```

型 メンバ名	入出力	説明
vdc4_VsyncSigSel Vsel	in	出力する垂直同期信号の選択
vdc4_SyncCtrl * sync	in	同期制御 NULL を設定した場合、垂直同期信号の多発マスク及び垂直同期信号の欠落補償制御はオフに設定されます。
vdc4_PanelClockSel panel_icksel	in	パネルクロック供給源選択 • VDC4_LCDPANEL_CLKSEL_IMG : 映像クロック選択(VIDEO_X1) • VDC4_LCDPANEL_CLKSEL_EXT : 外部クロック選択(LCDEXT_CLK) • VDC4_LCDPANEL_CLKSEL_PERI :

		周辺クロック 1(P1φ) • VDC4_LCDPANEL_CLKSEL_IMG_DV : 映像クロック選択(DV_CLK)
vdc4_PanelClkDCDR panel_dcdr	in	クロック分周比設定 • VDC4_LCDPANEL_CLKDIV_1_1 : 1/1 • VDC4_LCDPANEL_CLKDIV_1_2 : 1/2 • VDC4_LCDPANEL_CLKDIV_1_3 : 1/3 • VDC4_LCDPANEL_CLKDIV_1_4 : 1/4 • VDC4_LCDPANEL_CLKDIV_1_5 : 1/5 • VDC4_LCDPANEL_CLKDIV_1_6 : 1/6 • VDC4_LCDPANEL_CLKDIV_1_7 : 1/7 • VDC4_LCDPANEL_CLKDIV_1_8 : 1/8 • VDC4_LCDPANEL_CLKDIV_1_9 : 1/9 • VDC4_LCDPANEL_CLKDIV_1_12 : 1/12 • VDC4_LCDPANEL_CLKDIV_1_16 : 1/16 • VDC4_LCDPANEL_CLKDIV_1_24 : 1/24 • VDC4_LCDPANEL_CLKDIV_1_32 : 1/32
_UWORD tcon_half	in	TCON 1/2fH タイミング設定 水平同期信号の立ち上がりからのクロック数を指定 0x0000 ~ 0x07FF
_UWORD tcon_offset	in	オフセット付き水平同期人号のタイミング設定 水平同期信号の立ち上がりからのクロック数を指定 0x0000 ~ 0x07FF
vdc4_Edge outcnt_lcd_edge	in	LCD_DATA23〜0 端子の出力位相制御 • VDC4_EDGE_RISING : 立ち上がりエッジ • VDC4_EDGE_FALLING : 立ち下がりエッジ
vdc4_LcdTconTim * outctrl[VDC4_LCD_TCONSIG_NUM]	in	LCD TCON のタイミング設定 • outctrl[VDC4_LCD_TCONSIG_STVA_VS] : STVA / VS • outctrl[VDC4_LCD_TCONSIG_STVB_VE] : STVB / VE • outctrl[VDC4_LCD_TCONSIG_STH_SP_HS] : STH / SP / HS • outctrl[VDC4_LCD_TCONSIG_STB_LP_HE] : STB / LP / HE • outctrl[VDC4_LCD_TCONSIG_CPV_GCK] : CPV / GCK • outctrl[VDC4_LCD_TCONSIG_POLA] : POLA • outctrl[VDC4_LCD_TCONSIG_POLB] : POLB • outctrl[VDC4_LCD_TCONSIG_DE] : DE 設定が不要な端子については、NULL を設定。
vdc4_LcdOutput * settings	in	LCD 出力制御 NULL を設定しないで下さい。
_UDWORD IntEnable_1	in	割り込み許可ビット 1 (INT 0 - INT 7) 指定した割り込みが使用されます。複数の割り込みを使用する場合は、OR で設定して下さい。 • VDC4_INT_BIT_NONE : 割り込み無し • VDC4_INT_BIT_VI_VSYNC : VI_VSYNC • VDC4_INT_BIT_LO_VSYNC : LO_VSYNC • VDC4_INT_BIT_VSYNCERR : VSYNCERR • VDC4_INT_BIT_VLINE : VLINE • VDC4_INT_BIT_VFIELD : VFIELD

		• VDC4_INT_BIT_VBUFERR1 : VBUFFER1 • VDC4_INT_BIT_VBUFERR2 : VBUFFER2 • VDC4_INT_BIT_VBUFERR3 : VBUFFER3
_UDWORD IntEnable_2	in	割り込み許可ビット 2 (INT 8) • VDC4_INT_BIT_NONE : 割り込み無し • VDC4_INT_BIT_VBUFERR4 : VBUFFER4

vdc4_VsyncSigSel 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_OnOff FreeRunVsync ;
    _UWORD res_fv ;
    _UWORD res_fh ;
} vdc4_VsyncSigSel ;
```

型 メンバ名	入出力	説明
vdc4_OnOff FreeRunVsync	in	自走用垂直同期信号の ON/OFF • VDC4_OFF • VDC4_ON
_UWORD res_fv	in	自走用垂直同期信号の周期設定 Vsync 周期 = (res_fv + 1) x Hsync 周期 0x0000 ~ 0x07FF [lines] FreeRunVsync が VDC4_OFF に設定された時、参照されません。
_UWORD res_fh	in	水平同期信号の周期設定 Hsync 周期 = (res_fh + 1) / ピクセルクロック周波数 0x0000 ~ 0x07FF [pixel clock cycles]

vdc4_SyncCtrl 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_OnOff res_vmask_on ;
    _UWORD res_vmask ;
    vdc4_OnOff res_vlack_on ;
    _UWORD res_vlack ;
} vdc4_SyncCtrl ;
```

型 メンバ名	入出力	説明
vdc4_OnOff res_vmask_on	in	垂直同期信号の多発マスク制御 • VDC4_OFF • VDC4_ON
_UWORD res_vmask	in	垂直同期信号の多発マスク期間設定 128 ピクセルクロック周期で指定
vdc4_OnOff res_vlack_on	in	垂直同期信号の欠落補償制御 • VDC4_OFF • VDC4_ON
_UWORD res_vlack	in	垂直同期信号の欠落補償期間設定 128 ピクセルクロック周期で指定

vdc4_LcdTconTim 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD tcon_hsvs ;
    _UWORD tcon_hwvw ;
    vdc4_LcdTconPolMode tcon_md ;
    _UWORD tcon_hs_sel ;
    _UWORD tcon_inv ;
    vdc4_LcdTconPin tcon_pin ;
    vdc4_Edge outcnt_edge ;
} vdc4_LcdTconTim ;
```

型 メンバ名	入出力	説明
_UWORD tcon_hsvs	in	信号のパルス開始位置(第 1 の変化タイミング)を設定 同期信号の立ち上がりから tcon_hsvs 後にパルス出力開始 0x0000 ~ 0x07FF [pixel clock cycles or 1/2fH cycles] POLA / POLB 信号使用時は tcon_md がノーマルモードの時以外は 1 以上を設定してください。
_UWORD tcon_hwvw	in	信号のパルス幅 tcon_hwvw 期間パルス出力 0x0000 ~ 0x07FF [pixel clock cycles or 1/2fH cycles]
vdc4_LcdTconPolMode tcon_md	in	POLA / POLB 信号の生成モード • VDC4_LCD_TCON_POLMD_NORMAL : ノーマルモード • VDC4_LCD_TCON_POLMD_1X1REV : 1x1 リバースモード • VDC4_LCD_TCON_POLMD_1X2REV : 1x2 リバースモード • VDC4_LCD_TCON_POLMD_2X2REV : 2x2 リバースモード
_UWORD tcon_hs_sel	in	水平信号の動作基準選択 • 0 : 水平同期信号基準 • 1 : オフセット後の水平同期信号基準 オフセット値は、vdc4_InitAttr 構造体の tcon_offset になります。
_UWORD tcon_inv	in	信号の極性反転制御 • 0 : 非反転 • 1 : 反転
vdc4_LcdTconPin tcon_pin	in	出力端子選択 • VDC4_LCD_TCON_PIN_NON : 無し • VDC4_LCD_TCON_PIN_0 : LCD_TCON0 • VDC4_LCD_TCON_PIN_1 : LCD_TCON1 • VDC4_LCD_TCON_PIN_2 : LCD_TCON2 • VDC4_LCD_TCON_PIN_3 : LCD_TCON3 • VDC4_LCD_TCON_PIN_4 : LCD_TCON4 • VDC4_LCD_TCON_PIN_5 : LCD_TCON5 • VDC4_LCD_TCON_PIN_6 : LCD_TCON6
vdc4_Edge outcnt_edge	in	信号の出力位相制御 • VDC4_EDGE_RISING : 立ち上がりエッジ • VDC4_EDGE_FALLING : 立ち下がりエッジ

		このパラメータは信号出力しない(tcon_pin が VDC4_LCD_TCON_PIN_NON)場合、参照されません。

【注】 tcon_hsvs と tcon_hwvw は水平信号の時はピクセルクロック単位で、垂直信号の時は 1/2fH 単位で指定してください。

構造体「vdc4_LcdTconTim」のメンバは必ずしも全て設定する必要はありません。使用する信号によって、設定が必要なメンバは異なります。

Table 9 vdc4_LcdTconTim 構造体とタイミング信号

Signal / Member	STVA	STVB	STH	STB	CPV	POLA	POLB	DE
tcon_hsvs	R	R	R	R	R	R	R	I
tcon_hwvw	R	R	R	R	R	R	R	I
tcon_md	I	I	I	I	I	R	R	I
tcon_hs_sel	I	I	R	R	R	R	R	I
tcon_inv	R	R	R	R	R	R	R	R
tcon_pin	R	R	R	R	R	R	R	R
outcnt_edge	R	R	R	R	R	R	R	R

【注】 "R":参照します。
"I" :参照されません。

vdc4_LcdOutput 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_AreaRect res_f ;
    vdc4_OnOff out_endian_on ;
    vdc4_OnOff out_swap_on ;
    vdc4_LcdOutFormat out_format ;
    vdc4_LcdClkFrqSel out_frq_sel ;
    _UWORD out_dir_sel ;
    vdc4_LcdClkPhase out_phase ;
} vdc4_LcdOutput ;
```

型 メンバ名	入出力	説明
vdc4_AreaRect res_f	in	フル画面イネーブル制御 2.1.3章のvdc4_AreaRect構造体を参照下さい。 res_f.hs は 16 クロック以上、res_f.hs + res_f.hw は 2015 クロック以内になるように設定してください。 res_f.vs は 4 ライン以上、res_f.vs + res_f.vw は 2039 ライン以内になるように設定してください。
vdc4_OnOff out_endian_on	in	ビットエンディアン変更オン/オフ制御 • VDC4_OFF • VDC4_ON
vdc4_OnOff out_swap_on	in	B/R 信号入れ替えオン/オフ制御 • VDC4_OFF • VDC4_ON
vdc4_LcdOutFormat out_format	in	出力フォーマット選択 • VDC4_LCD_OUTFORMAT_RGB888 : RGB888 • VDC4_LCD_OUTFORMAT_RGB666 : RGB666 • VDC4_LCD_OUTFORMAT_RGB565 : RGB565 • VDC4_LCD_OUTFORMAT_SERIAL_RGB :

		serial RGB
vdc4_LcdClkFrqSel out_frq_sel	in	クロック周波数制御 ● VDC4_LCD_SERIAL_CLKFRQ_3 : x 3, serial RGB ● VDC4_LCD_SERIAL_CLKFRQ_4 : x 4, serial RGB LCD の出力フォーマット(out_format)がシリアル RGB の時のみ参照します。
_UWORD out_dir_sel	in	スキャン方向選択 ● 0 : 正スキャン ● 1 : 逆スキャン LCD の出力フォーマット(out_format)がシリアル RGB の時のみ参照します。
vdc4_LcdClkPhase out_phase	in	シリアル RGB 出力時のクロック位相調整 3 倍速の時 ● VDC4_LCD_SERIAL_CLKPHASE_0 : 0[clk] ● VDC4_LCD_SERIAL_CLKPHASE_1 : 1[clk] ● VDC4_LCD_SERIAL_CLKPHASE_2 : 2[clk] 4 倍速の時 ● VDC4_LCD_SERIAL_CLKPHASE_0 : 0[clk] ● VDC4_LCD_SERIAL_CLKPHASE_1 : 1[clk] ● VDC4_LCD_SERIAL_CLKPHASE_2 : 2[clk] ● VDC4_LCD_SERIAL_CLKPHASE_3 : 3[clk] LCD の出力フォーマット(out_format)がシリアル RGB の時のみ参照します。

2.3.2 VDC4_Terminate

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_Terminate(void (*quit_func)(_UDWORD), _UDWORD user_num) ;	
引数	• [in]void (*quit_func)(_UDWORD)	ユーザ定義関数のポインタ
	• [in]_UDWORD user_num	ユーザ定義番号
戻り値	• vdc4_ErrorCode	エラーコード
	VDC4_ERR_NONE	正常終了
	VDC4_ERR_SURFACE_STATUS	サーフェスの状態に合わない処理を実施

概要

本関数では以下の処理を行います。

- ユーザ定義関数の実行
- パネルクロックの動作禁止設定
- VDC4 モジュールの割り込みの禁止設定
- 全てのサーフェスの状態を"Undef"に遷移

本関数では終了処理の最後に`quit_func`で指定されたユーザ定義関数を呼び出します。ユーザ定義関数の処理内容については「3.2VDC4_Terminateのユーザ定義関数例」を参考にしてください。

ユーザ定義関数は必ずしも指定する必要はありません。ユーザ定義関数では以下のような処理を行う事が想定されています。

- VDC4 モジュールへのクロック供給停止
- 割り込み優先度の設定クリア
- その他、LCD 表示やビデオ等の環境固有の設定

引数の設定

型 引数名	入出力	説明
void (*quit_func)(_UDWORD)	in	ユーザ定義関数のポインタ ユーザ定義関数が設定された場合、user_num が引数として渡されます。必要な場合、ユーザ定義関数を実装して下さい。 書式 void Quit_Func(_UDWORD User_Num) ; 引数 • [in]_UDWORD ユーザ定義番号 User_Num 戻り値 • void 概要 ユーザによって実装された処理が実行されます。
_UDWORD user_num	in	ユーザ定義番号 このパラメータは、ユーザ定義関数の引数として使用されます。ユーザ定義関数ポインタが NULL の場合使用しません。

2.3.3 VDC4_GraphicsCreateSurface

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsCreateSruface(vdc4_LayerID id, const vdc4_GraphicsAttr *attr) ;	
引数	• [in]vdc4_LyerID id • [in]const vdc4_GraphicsAttr *attr	レイヤ ID グラフィックスパラメータ
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_UNDEF VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定 必要なパラメータに NULL を設定 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- ID で指定されたグラフィックスサーフェスの作成
- カラーフォーマットの指定
- フレームバッファの設定
- ID で指定されたグラフィックスサーフェスの状態を"Init"から"Ready"に遷移
- 拡大処理の設定(グラフィックス(1)サーフェス生成時のみ)

カラーフォーマットがCLUT1, CLUT4, CLUT8 の場合、本関数による設定とは別にCLUTを設定する必要があります。また、カラーフォーマットがARGB1555 の場合は、α値を設定する必要があります。CLUTの設定とARGB1555 のα値の設定には関数「VDC4_GraphicsSetCLUT (「2.3.17」参照)」を使用します。

ビデオサーフェスとグラフィックス(1)サーフェスは排他動作となります。

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_GRAPHICS_1 :グラフィックス(1) • VDC4_SURFACE_GRAPHICS_2 :グラフィックス(2) • VDC4_SURFACE_GRAPHICS_3 :グラフィックス(3)
vdc4_GraphicsAttr * attr	in	グラフィックスパラメータ NULL を設定しないで下さい。

vdc4_GraphicsAttr 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_AreaRect gr_grc ;
    vdc4_FrameBuff buff ;
} vdc4_GraphicsAttr ;
```

型 メンバ名	入出力	説明
vdc4_AreaRect gr_grc	in	表示エリア 2.1.3章のvdc4_AreaRect構造体を参照下さい。 gr_grc.hs は 16 クロック以上、gr_grc.hs + gr_grc.hw は 2015 クロック以内になるように設定してください。また、gr_grc.hw は 3 クロック以上を設定してください。 gr_grc.vs は 4 ライン以上、gr_grc.vs + gr_grc.vw は 2039 ライン以内になるように設定してください。
vdc4_FrameBuff buff	in	フレームバッファパラメータ

vdc4_FrameBuff 構造体のメンバは以下の通りです。

```
typedef struct {
    _SINT gr_bst_md ;
    void *gr_base ;
    void *buff_2nd ;
    _UWORD gr_ln_off ;
    _SINT gr_ln_off_dir ;
    _UWORD gr_flm_loop ;
    vdc4_GrFormat gr_format ;
    vdc4_OnOff gr_endian_on ;
    _UDWORD bg_color ;
    vdc4_GraphicsExt *GraEx ;
} vdc4_FrameBuff ;
```

型 メンバ名	入出力	説明
_SINT gr_bst_md	in	フレームバッファバースト転送モード • 0 : 32 バイト • 1 : 128 バイト
void *	in	フレームバッファのベースアドレス

gr_base		NULL を設定することはできません。
void * buff_2nd	in	第２フレームバッファアドレス(ダブルバッファのみ) シングルバッファの場合は､NULL を設定して下さい。 第２フレームバッファのアドレスに関する制限事項については、Figure 4とその説明を参照ください。
_UWORD gr_ln_off	in	フレームバッファのオフセットアドレス [bytes] gr_bst_md が 0 の時は 32 の倍数、gr_bst_md が 1 の時は 128 の倍数を設定して下さい。
_SINT gr_ln_off_dir	in	フレームバッファのラインオフセットアドレスの方向 • 0 : ラインオフセットアドレス分をインクリメント • 1 : ラインオフセットアドレス分をデクリメント gr_ln_off_dir を 1 に設定すると、垂直方向に上下反転した画像が表示されます。
_UWORD gr_flm_loop	in	アドレスをリング状に読み出す場合のライン数 ライン数は gr_flm_loop + 1 になります。 0x0000(0) ~ 0x03FF(1023) この機能が必要でない場合、gr_flm_loop に十分大きな値を設定することを推奨します。(例 1023)
vdc4_GrFormat gr_format	in	フレームバッファ読み出し信号のフォーマット設定 • VDC4_FORMAT_RGB565 : RGB565 • VDC4_FORMAT_RGB888 : RGB888 • VDC4_FORMAT_ARGB1555 : ARGB1555 • VDC4_FORMAT_ARGB4444 : ARGB4444 • VDC4_FORMAT_ARGB8888 : ARGB8888 • VDC4_FORMAT_CLUT8 : CLUT8 • VDC4_FORMAT_CLUT4 : CLUT4 • VDC4_FORMAT_CLUT1 : CLUT1 • VDC4_FORMAT_YCC422 : YCbCr422, グラフィックス(1)のみ
vdc4_OnOff gr_endian_on	in	バッファ読み出しデータのエンディアン制御オン/オフ設定 • VDC4_OFF • VDC4_ON
_UDWORD bg_color	in	背景色 gr_formatがCLUT1､CLUT4､CLUT8 又は､YCbCr422 の場合､RGB888 形式で値を指定します。それ以外の場合はgr_formatで指定したカラーフォーマットで値を指定します (LSB詰め) 。 詳細は Figure 5を参照ください。
vdc4_GraphicsExt * GraEx	in	グラフィックス拡張パラメータ グラフィックス(1)を指定している場合のみ有効です。必要がない場合は NULL を設定して下さい。

ダブルバッファ使用時に指定する第 2 バッファ(buff_2nd)のメモリ配置に関する注意点を以下に示します (Figure 4参照)。

- 第 2 バッファの幅(gr_arc.hw)、高さ(gr_arc.vw)、ラインオフセット(gr_ln_off)は第 1 バッファと同じでなければならない。
- 第 2 バッファを確保する位置は第 1 バッファより後のアドレスで無ければならない。
- 第 1 バッファと第 2 バッファ間のアドレスオフセット(Frame offset)は 0x00800000 (8.0Mbyte)より小さくなければならない。
- 第 1 バッファと第 2 バッファ間のアドレスオフセット(Frame offset)は、フレームバッファバースト転送モード(gr_bst_md)が 0 の時は 32 の倍数、1 の時は 128 の倍数でなければならない。

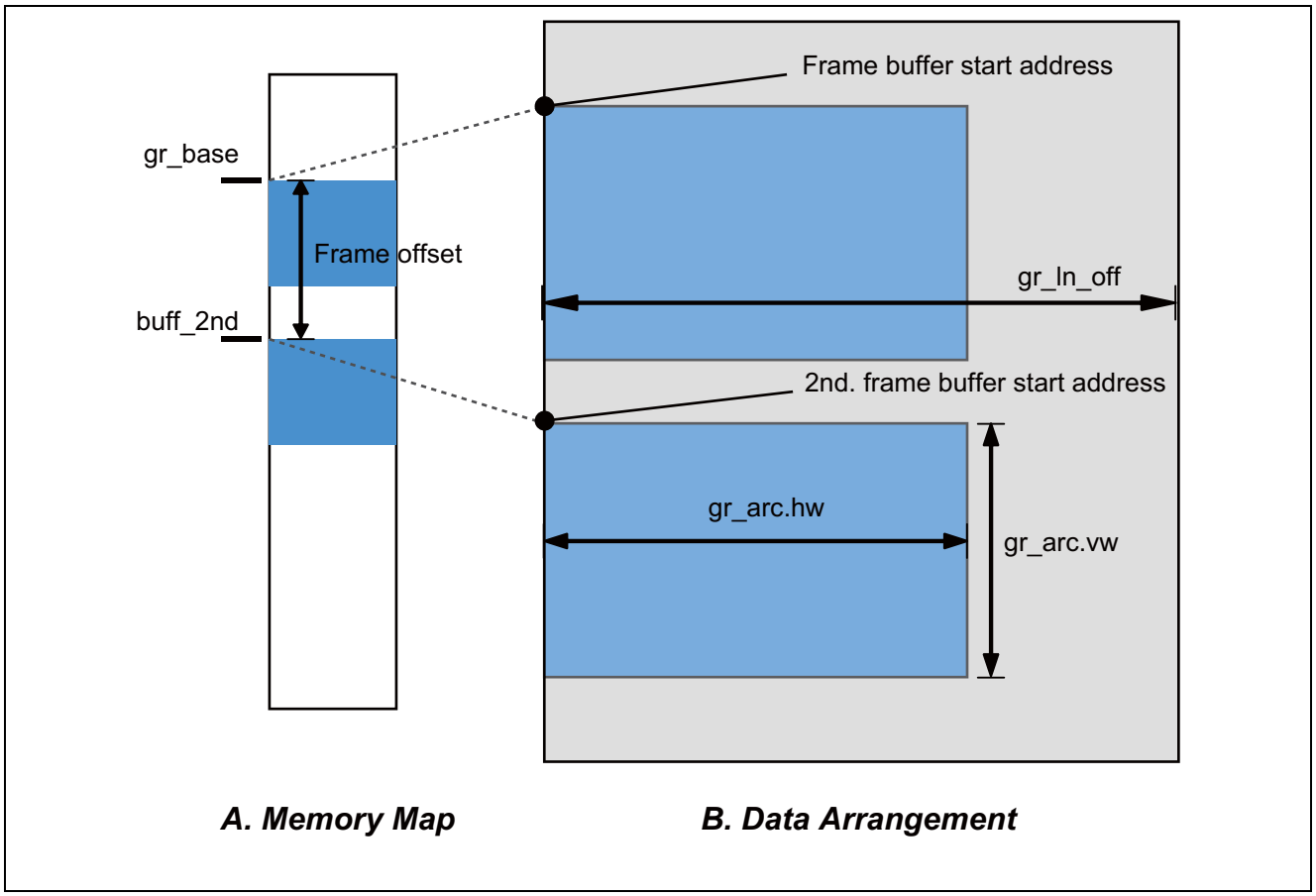
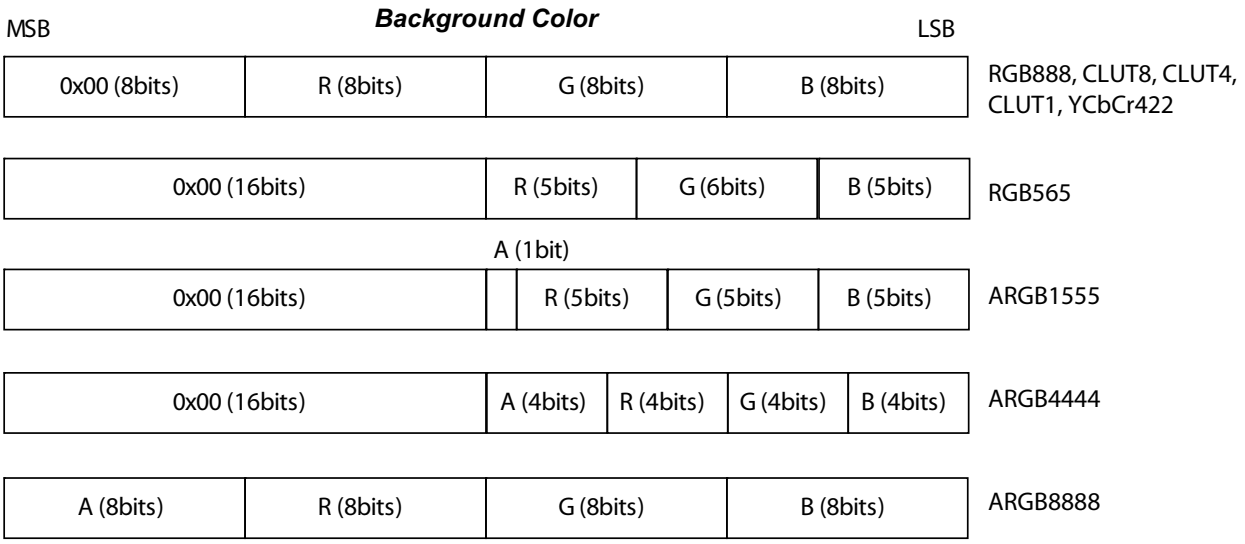


Figure 4　メモリの割り当てと第２フレームバッファの配置



【注】 ARGB1555、ARGB4444、ARGB8888 のα値は無視されます。背景色に何の影響も与えません。

Figure 5 背景色の設定

vdc4_GraphicsExt 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_YccSwapFormat gr_ycc_swap ;
    vdc4_YccExchgMode gr_cnv444_md ;
    _UWORD gr_in_hw ;
    _UWORD gr_in_vw ;
    vdc4_Interpolation res_interpotyp ;
    vdc4_OnOff adj_sel ;
} vdc4_GraphicsExt ;
```

型 メンバ名	入出力	説明
vdc4_YccSwapFormat gr_ycc_swap	in	YCbCr422 フォーマット時バッファ読み出しデータのスワップ制御 • VDC4_YCCSWAP_CBY0CRY1 : CbY0CrY1 • VDC4_YCCSWAP_Y0CBY1CR : Y0CbY1Cr • VDC4_YCCSWAP_CRY0CBY1 : CrY0CbY1 • VDC4_YCCSWAP_Y0CRY1CB : Y0CrY1Cb • VDC4_YCCSWAP_Y1CRY0CB : Y1CrY0Cb • VDC4_YCCSWAP_CRY1CBY0 : CrY1CbY0 • VDC4_YCCSWAP_Y1CBY0CR : Y1CbY0Cr • VDC4_YCCSWAP_CBY1CRY0 : CbY1CrY0 表示形式(gr_format)が YCbCr422 の時のみ参照します。また、この設定はエンディアン制御(gr_endian_on)が ON の時のみ有効となります。
vdc4_YccExchgMode gr_cnv444_md	in	YCbCr422→YCbCr444 変換時の補間モード設定 • VDC4_YCC_444_HOLD : ホールド補間 • VDC4_YCC_444_AVERAGE : 平均値補間 表示形式(gr_format)が YCbCr422 の時のみ参照します。
_UWORD	in	グラフィックス(1)入力の水平幅 [pixel clock cycles]

gr_in_hw		gr_in_hw が表示エリア(gr_grc)で設定される水平幅より小さい場合でかつ、表示形式(gr_format)が RGB565, RGB888, YCbCr422 の時拡大処理が行われます。
_UWORD gr_in_vw	in	グラフィックス(1)入力の垂直幅 [lines] gr_in_vw が表示エリア(gr_grc)で設定される垂直幅より小さい場合でかつ、表示形式(gr_format)が RGB565, RGB888, YCbCr422 の時拡大処理が行われます。
vdc4_Interpolation res_interpotyp	in	補間方法選択 • VDC4_INTERPOLATION_HOLD :ホールド補間 • VDC4_INTERPOLATION_LINEAR :リニア補間 拡大処理が行われる場合のみ参照されます。
vdc4_OnOff adj_sel	in	折り返し対策オン/オフ制御 • VDC4_OFF • VDC4_ON 拡大処理が行われる場合のみ参照されます。

2.3.4 VDC4_VideoCreateSurface

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_VideoCreateSruface(vdc4_LayerID id, const vdc4_VideoAttr *attr) ;	
引数	• [in]vdc4_LyerID id • [in]const vdc4_VideoAttr *attr	レイヤ ID ビデオサーフェスパラメータ
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_UNDEF VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定 必要なパラメータに NULL を設定 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- ID で指定されたビデオサーフェスを作成
- スケーリング／回転の設定
- IP 変換、フィールド判別信号(RES_FLD_DLY_SEL)の制御
- ID で指定されたビデオスサーフェスの状態を"Init"から"Ready"に遷移

入力ビデオ信号に対するスケーリング及び回転処理は以下のような順序で処理されます(Figure 6参照)。

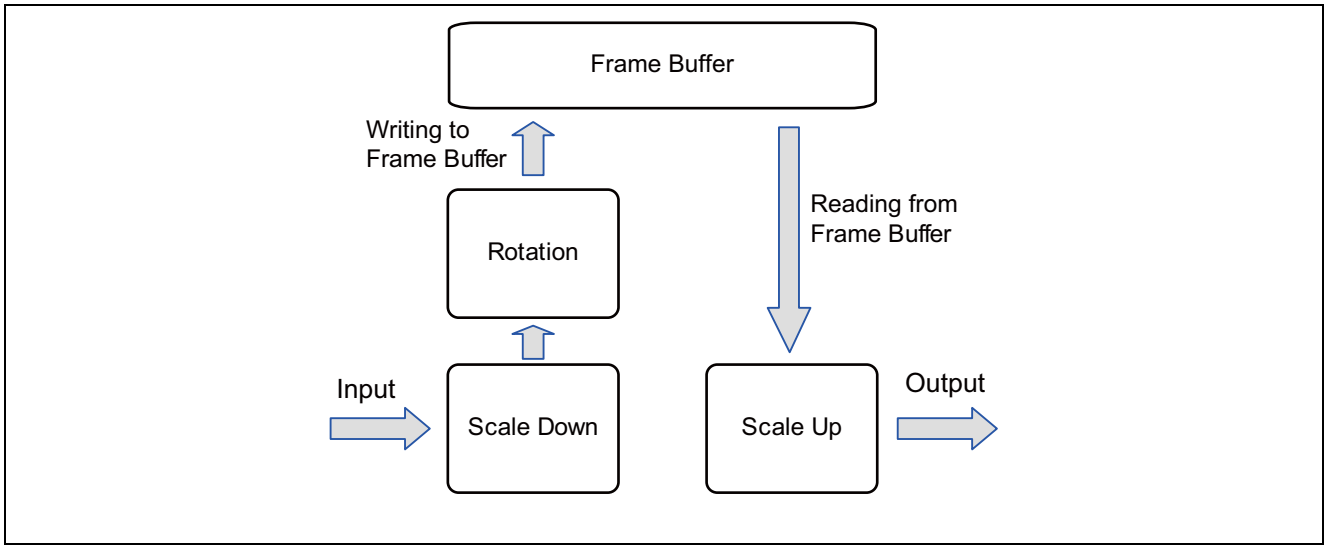
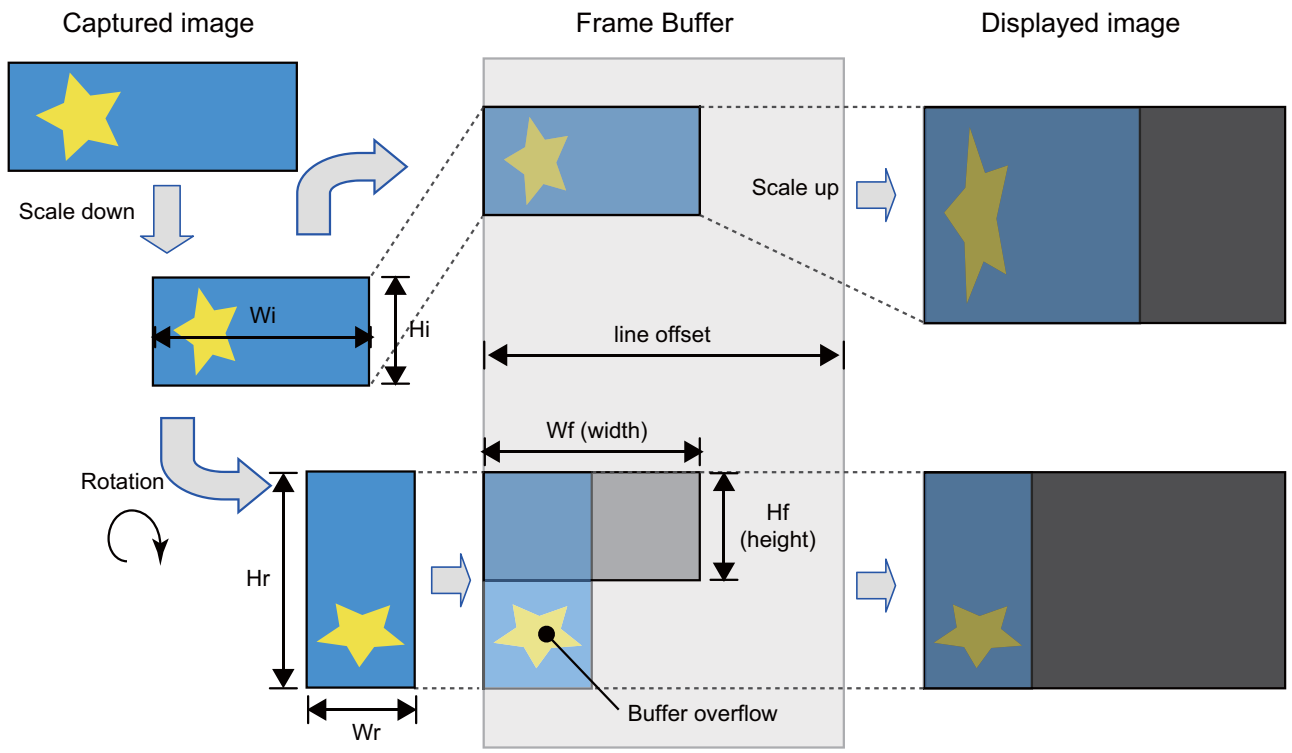


Figure 6 拡大縮小と回転処理フロー

Figure 6に示されているように、フレームバッファへの書き込みは拡大処理の前に行われます。従って、フレームバッファのサイズを決めるにあたり、拡大後のサイズを考慮する必要はありません。しかし、フレームバッファへの書き込みは、回転処理後に行われます。90 度や 270 度の回転時には、取り込み画像の幅と高さが入れ替わるので注意が必要です。

Figure 7は取り込み画像に対するスケーリングと回転処理により、必要なフレームバッファサイズが変わる事を示したものです。図の上段は、取り込み画像を縮小スケーリング後にフレームバッファへ書き込みます。フレームバッファは、書き込まれるデータに十分なサイズがあります。図の下段の場合、縮小スケーリング後に回転処理が行われることで幅と高さが入れ替わります。この為、フレームバッファの高さを超えてしまいオーバフローが発生します。

ビデオ入力映像のフレームバッファを確保する際には、縮小スケーリングと回転処理の事を考慮に入れてメモリを確保してください。縮小スケーリングや回転処理により、フレームバッファのラインオフセットアドレス(`res_ln_off`)やフレームオフセットアドレス(`res_flm_off`)の設定値が影響を受けます。これらのパラメータについては構造体 `vdc4_Res_FrameBuff` の説明を参照してください。



【注】 Wi, Hi: 取り込み画像の縮小後の幅と高さ。
Wr, Hr: 取り込み画像の回転後の幅と高さ。Wr と Hi、Hr と Wi は同じサイズ。
Wf, Hf: フレームバッファの幅と高さ。Hf は Hi と等しく、Hr より小さいサイズ。

Figure 7 ビデオフレームバッファ

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_VIDEO_1 : ビデオサーフェス
vdc4_VideoAttr * attr	in	ビデオサーフェスパラメータ NULL を設定しないで下さい。

vdc4_VideoAttr 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_VideoType type ;
    _SINT inp_sel ;
    _UWORD inp_fh50 ;
    _UWORD inp_fh25 ;
    vdc4_ExtInSig *ExtSig ;
    vdc4_InpDlay *dly ;
    vdc4_AreaRect res ;
    vdc4_AreaRect res_p ;
    vdc4_ScalingRot *ScaleRot ;
    vdc4_ResFrameBuff frame ;
    vdc4_VideoDisplay *Disp ;
} vdc4_VideoAttr ;
```

型 メンバ名	入出力	説明
vdc4_VideoType type	in	ビデオサーフェスのタイプ • VDC4_VIDEO_TYPE_DISPLAY :表示 • VDC4_VIDEO_TYPE_RECORD :録画
_SINT inp_sel	in	入力選択 • 0 : ビデオデコーダ出力 • 1 : 外部入力端子 関数「VDC4_Initialize」でパネルクロック供給源選択(panel_icksel)にVDC4_LCDPANEL_CLKSEL_IMGを指定した場合 0 を、VDC4_LCDPANEL_CLKSEL_IMG_DVを指定した場合 1 を設定してください。
_UWORD inp_fh50	in	垂直同期の 1/2fH 位相タイミング設定 0x0000 ~ 0x03FF [clock cycles]
_UWORD inp_fh25	in	垂直同期の 1/4fH 位相タイミング設定 0x0000 ~ 0x03FF [clock cycles]
vdc4_ExtInSig * ExtSig	in	外部入力信号パラメータ inp_sel に 1 を設定した場合、NULL を設定しないで下さい。inp_sel に 0 を設定した場合、ExtSig は、参照されません。
vdc4_InpDlay * dly	in	同期信号遅延調整パラメータ 必要がない場合、NULL を設定して下さい。
vdc4_AreaRect res	in	画像取り込み範囲設定 2.1.3章のvdc4_AreaRect構造体を参照下さい。 res.hs は 16 クロック以上、res.hs + res.hw は 2015 クロック以内になるように設定してください。また、

		res.hw は 4 クロックアライメントで設定してください。 res.vs は 4 ライン以上、res.vs + res.vw は 2039 ライン以内になるように設定してください。また、res.vw は 4 ラインアライメントで設定してください。
vdc4_AreaRect res_p	in	画像出力イネーブル範囲設定 2.1.3章のvdc4_AreaRect構造体を参照下さい。 res_p.hs は 16 クロック以上、res_p.hs + res_p.hw は 2015 クロック以内になるように設定してください。また、res_p.hw は 4 クロックアライメントで設定してください。 res_p.vs は 4 ライン以上、res_p.vs + res_p.vw は 2039 ライン以内になるように設定してください。また、res_p.vw は 4 ラインアライメントで設定してください。
vdc4_ScalingRot * ScaleRot	in	スケーリング／回転パラメータ 必要がない場合、NULL を設定して下さい。 ScaleRotがNULLに設定された場合、スケーリング及び回転関連パラメータは、Table 10の値に設定されます。
vdc4_ResFrameBuff frame	in	ビデオサーフェスのフレームバッファパラメータ
vdc4_VideoDisplay * Disp	in	ビデオサーフェスの表示パラメータ ビデオサーフェスのタイプ(type)が表示の場合、NULL を設定しないで下さい。

vdc4_ExtInSig 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_InpFormat format ;
    vdc4_OnOff inp_endian_on ;
    vdc4_OnOff inp_swap_on ;
    vdc4_Edge inp_pxd_edge ;
    vdc4_Edge inp_vs_edge ;
    vdc4_Edge inp_hs_edge ;
    _SINT inp_vs_inv ;
    _SINT inp_hs_inv ;
    _SINT inp_f525_625 ;
    _SINT inp_h_edge_sel ;
    vdc4_InpHpos inp_h_pos ;
} vdc4_ExtInSig ;
```

型 メンバ名	入出力	説明
vdc4_InpFormat format	in	外部入力のフォーマット選択 • VDC4_IN_FORMAT_RGB888 : RGB888 • VDC4_IN_FORMAT_RGB666 : RGB666 • VDC4_IN_FORMAT_RGB565 : RGB565 • VDC4_IN_FORMAT_BT656 : BT656 • VDC4_IN_FORMAT_BT601 : BT601
vdc4_OnOff inp_endian_on	in	外部入力のビットエンディアン変更オン/オフ制御 • VDC4_OFF • VDC4_ON
vdc4_OnOff inp_swap_on	in	外部入力の B/R 信号入れ替えオン/オフ制御 • VDC4_OFF

		• VDC4_ON
vdc4_Edge inp_pxd_edge	in	外部入力の映像信号の入力段取り込みクロックのエッジ選択 • VDC4_EDGE_RISING : 立ち上がりエッジ • VDC4_EDGE_FALLING : 立ち下がりエッジ
vdc4_Edge inp_vs_edge	in	外部入力の垂直同期信号の入力段取り込みクロックのエッジ選択 • VDC4_EDGE_RISING : 立ち上がりエッジ • VDC4_EDGE_FALLING : 立ち下がりエッジ
vdc4_Edge inp_hs_edge	in	外部入力の水平同期信号の入力段取り込みクロックのエッジ選択 • VDC4_EDGE_RISING : 立ち上がりエッジ • VDC4_EDGE_FALLING : 立ち下がりエッジ
_SINT inp_vs_inv	in	外部入力の垂直同期信号の反転制御 • 0 : 非反転(正極性) • 1 : 反転(負極性)
_SINT inp_hs_inv	in	外部入力の水平同期信号の反転制御 • 0 : 非反転(正極性) • 1 : 反転(負極性)
_SINT inp_f525_625	in	外部入力系統の BT656 入力時のライン数設定 • 0 : 525 [lines] • 1 : 625 [lines]
_SINT inp_h_edge_sel	in	外部入力系統の BT656 水平同期信号の基準選択 • 0 : EAV 基準 • 1 : SAV 基準
vdc4_InpHpos inp_h_pos	in	水平同期基準に対する Y/Cb/Y/Cr のデータ列の開始タイミング設定 • VDC4_INP_H_POS_CBYCRY : Cb/Y/Cr/Y • VDC4_INP_H_POS_YCRYCB : Y/Cr/Y/Cb • VDC4_INP_H_POS_CRYCBY : Cr/Y/Cb/Y • VDC4_INP_H_POS_YCBYCR : Y/Cb/Y/Cr

vdc4_InpDlay 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD inp_vs_dly_l ;
    _UWORD inp_vs_dly ;
    _UWORD inp_hs_dly ;
    _UWORD inp_fld_dly ;
} vdc4_InpDlay ;
```

型 メンバ名	入出力	説明
_UWORD inp_vs_dly_l	in	垂直同期信号、フィールド判別のライン遅延量 0 ~ 7 [lines]
_UWORD inp_vs_dly	in	垂直同期信号の遅延量 0 ~ 254 [clock cycles]
_UWORD inp_hs_dly	in	水平同期信号の遅延量 0 ~ 254 [clock cycles]
_UWORD inp_fld_dly	in	フィールド判別信号の遅延量 0 ~ 254 [clock cycles]

vdc4_ScalingRot 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_OnOff res_pfil_sel ;
    vdc4_Interpolation res_interpotyp ;
    vdc4_RotationType rot ;
    vdc4_OnOff adj_sel ;
} vdc4_ScalingRot ;
```

型 メンバ名	入出力	説明
vdc4_OnOff res_pfil_sel	in	輝度信号プリフィルタモード選択 • VDC4_OFF • VDC4_ON
vdc4_Interpolation res_interpotyp	in	補間方法選択 • VDC4_INTERPOLATION_HOLD :ホールド補間 • VDC4_INTERPOLATION_LINEAR :リニア補間
vdc4_RotationType rot	in	回転及び水平鏡像設定 • VDC4_ROT_NORMAL :無し • VDC4_ROT_MIRROR :水平鏡像 • VDC4_ROT_90_DEG : 90 度回転 • VDC4_ROT_180_DEG :180 度回転 • VDC4_ROT_270_DEG :270 度回転
vdc4_OnOff adj_sel	in	折り返し及び入力最終ライン欠落対策オン/オフ制御 • VDC4_OFF • VDC4_ON この設定は水平／垂直拡大処理の拡大率と垂直縮小処理の縮小率に影響します。水平縮小時の入力最終画素欠落対策は常に行われます。

このスケーリングと回転処理のパラメータは不要であれば設定をする必要はありません。ポインタ ScaleRot へ NULL を設定することで設定を省略できます。この場合、スケーリングと回転処理のパラメータとして以下の値が適用されます。

Table 10 スケーリングと回転処理に適用される初期値

パラメータ	初期値
res_pfil_sel	VDC4_OFF
res_interpotyp	VDC4_INTERPOLATION_HOLD
rot	VDC4_ROT_NORMAL
adj_sel	VDC4_OFF

本ドライバでは、取り込み画像サイズ(res)と出力画像サイズ(res_p)の差分から自動的に判定してスケーリング処理を行います。サイズの判定では、回転処理による水平方向と垂直方向のサイズの入れ替わりも考慮されます(Figure 8参照)。

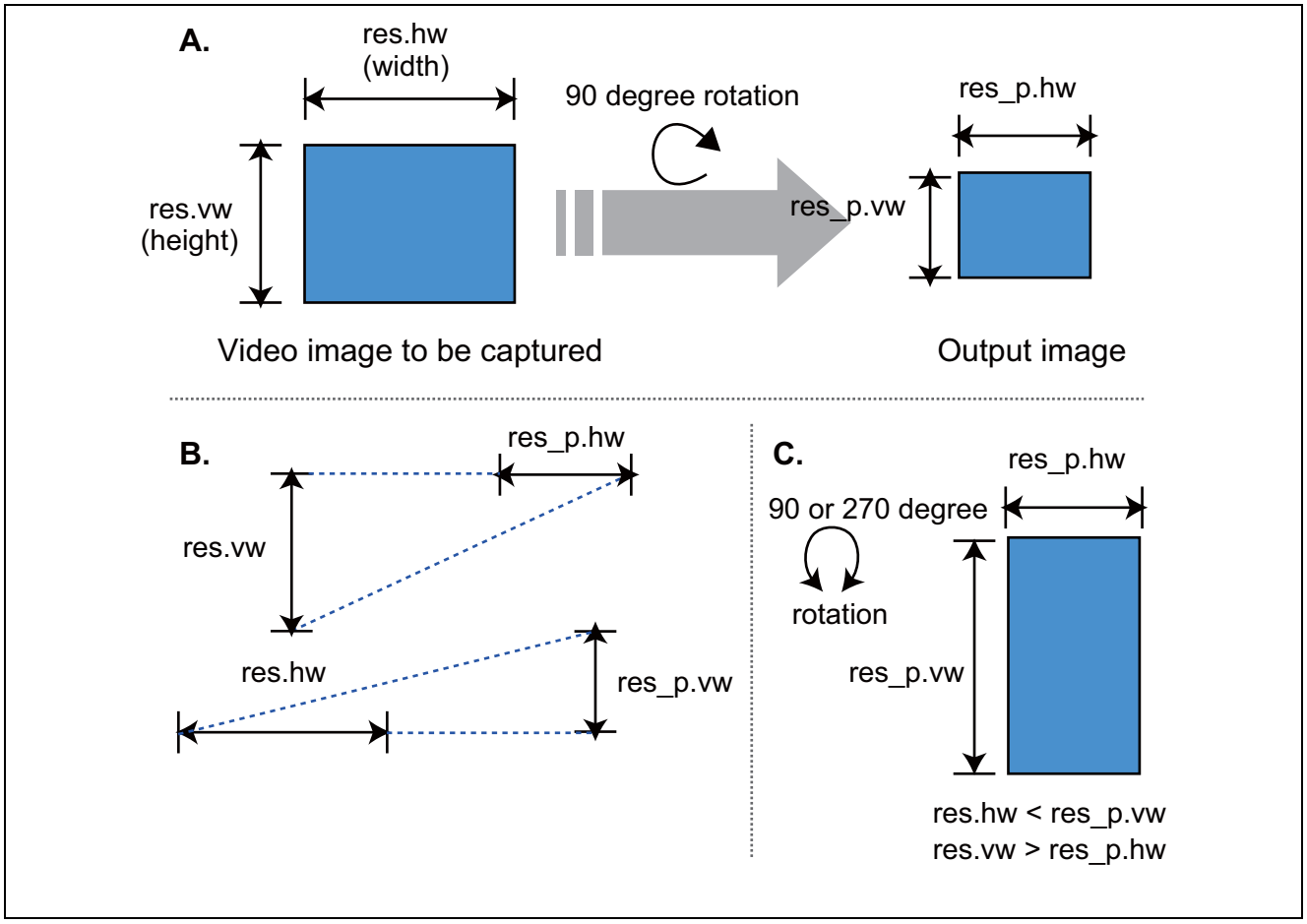


Figure 8　拡大縮小と回転

Figure 8はスケーリングと回転処理の例です。図Aは、取り込み画像と出力画像のスケーリングと 90 度回転している様子を表しています。Bは 90 度の回転により、比較する水平サイズと垂直サイズが入れ替わった事を表しています。Cは禁止されている処理を表します。Cのように 90 か 270 度回転する時、取り込み画像の幅が出力画像の高さより小さく、取り込み画像の高さが出力画像の幅より大きいような変換は禁止です。

また、本ドライバではスケーリング、回転処理においてIP変換のための初期位相制御を行います。この設定は、ユーザが行う必要はありません。初期位相制御で設定されるレジスタとパラメータは以下のとおりです(Table 11参照)。

Table 11　IP 変換時のスケーリング初期位相設定

回転制御	水平 スケーリング	垂直 スケーリング	レジスタ	設定値
通常	拡大/縮小	拡大/縮小	RES_TOP_INIPASE	2048
水平鏡像	拡大/縮小	拡大/縮小	RES_TOP_INIPASE	2048
90 度回転	縮小	縮小	RES_TOP_INIPASE	2048
	拡大と縮小	-	RES_TOP_INIPASE	2048
	拡大	拡大	RES_US_HB_INIPHASE	2048
180 度回転	拡大/縮小	縮小	RES_TOP_INIPASE	2048
	拡大/縮小	拡大	RES_BTM_INIPASE	2048
270 度回転	縮小	縮小	RES_TOP_INIPASE	2048
	拡大と縮小	-	RES_TOP_INIPASE	2048
	拡大/縮小	拡大	RES_US_HT_INIPHASE	2048

vdc4_ResFrameBuff 構造体のメンバは以下の通りです。

```
typedef struct {
    _SINT res_bst_md ;
    void *res_base ;
    _UWORD res_ln_off ;
    _UDWORD res_flm_off ;
    vdc4_ResFormat res_md ;
    vdc4_OnOff res_dth_on ;
    vdc4_ResFsRate res_fs_rate ;
    _SINT res_fld_sel ;
    _SINT res_inter ;
    _UWORD res_flm_num ;
    _SINT res_loop ;
    _UDWORD bg_color ;
} vdc4_ResFrameBuff ;
```

型 メンバ名	入出力	説明
_SINT res_bst_md	in	フレームバッファ書き込み転送のバースト長 • 0 : 32 バイト転送 • 1 : 128 バイト転送
void * res_base	in	フレームバッファのベースアドレス NULL を設定しないで下さい。 res_bst_md が 0 の時、32 バイト境界のアドレスを設定してください。res_bst_md が 1 の時、128 バイト境界のアドレスを設定してください。
_UWORD res_ln_off	in	フレームバッファのラインオフセットアドレス [byte] res_bst_md が 0 の時は 32 の倍数、res_bst_md が 1 の時は 128 の倍数を設定して下さい。
_UDWORD res_flm_off	in	フレームバッファのフレームオフセットアドレス [byte] res_bst_md が 0 の時は 32 の倍数、res_bst_md が 1

		の時は 128 の倍数を設定して下さい。
vdc4_ResFormat res_md	in	フレームバッファ書き込み映像フォーマット • VDC4_RES_MD_YCC422 : YCbCr422 • VDC4_RES_MD_RGB565 : RGB565 • VDC4_RES_MD_RGB888 : RGB888
vdc4_OnOff res_dth_on	in	ディザ補正オン/オフ設定 • VDC4_OFF • VDC4_ON
vdc4_ResFsRate res_fs_rate	in	書き込み間隔 • VDC4_RES_FS_RATE_PER1 : 1/1 • VDC4_RES_FS_RATE_PER2 : 1/2 • VDC4_RES_FS_RATE_PER4 : 1/4 • VDC4_RES_FS_RATE_PER8 : 1/8
_SINT res_fld_sel	in	書き込みフィールド選択 • 0 : Top フィールド • 1 : Bottom フィールド res_fs_rate が VDC4_RES_FS_RATE_PER1 以外の時のみ、このパラメータは有効となります。
_SINT res_inter	in	フィールド動作モード設定 • 0 : プログレッシブ • 1 : インタレース
_UWORD res_flm_num	in	書き込みフレームバッファのフレーム数 (res_flm_num+1) ビデオサーフェスのタイプ(type) 記録：0x0000～0x3FF 表示：0 or 1 (1 面 or 2 面) 画像を回転させる時、0 (1 面)は設定禁止です。
_SINT res_loop	in	フレームバッファ書き込みモード選択 • 0 : フレーム書き込みモード • 1 : ライン書き込みモード (リング状読み出し)
_UDWORD bg_color	in	背景色 res_md がYCbCr422 の場合、CrYCb形式で値を設定します。但し、上記以外の場合、RGB888 形式で値を設定します(LSB詰め)。Figure 9の背景色の設定を参照ください。

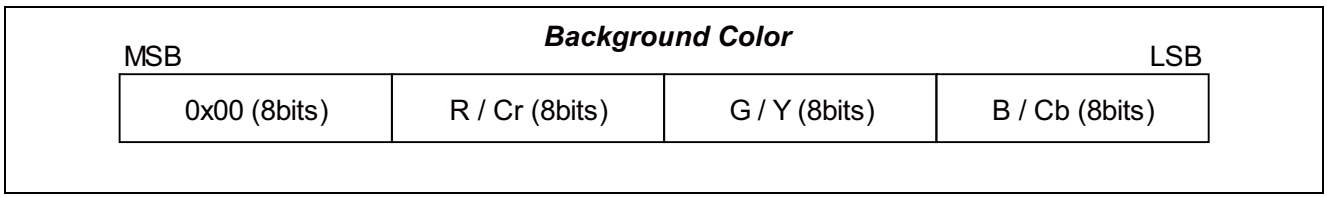


Figure 9 背景色の設定

色データの RGB888 から YCbCr への変換は以下の式で計算できます。(SMPTE 293M 規格による標準値)

Y = 0.299R + 0.587G + 0.114B
Cb = -0.169R – 0.331G + 0.500B + 128.0
Cr = 0.500R – 0.419G – 0.081B + 128.0

`vdc4_VideoDisplay` 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD res_vsdly ;
    vdc4_OnOff gr_endian_on ;
    vdc4_YccSwapFormat gr_ycc_swap ;
    vdc4_YccExchgMode gr_cnv444_md ;
} vdc4_VideoDisplay ;
```

型 メンバ名	入出力	説明
_UWORD res_vsdly	in	垂直同期信号遅延制御 0x0000 ~ 0x00FF 遅延量[usec] : res_vsdly x 出力水平周期[usec]
vdc4_OnOff gr_endian_on	in	バッファ読み出しデータのエンディアン制御オン/オフ設定 • VDC4_OFF • VDC4_ON
vdc4_YccSwapFormat gr_ycc_swap	in	YCbCr422 フォーマット時バッファ読み出しデータのスワップ制御 • VDC4_YCCSWAP_CBY0CRY1 : CbY0CrY1 • VDC4_YCCSWAP_Y0CBY1CR : Y0CbY1Cr • VDC4_YCCSWAP_CRY0CBY1 : CrY0CbY1 • VDC4_YCCSWAP_Y0CRY1CB : Y0CrY1Cb • VDC4_YCCSWAP_Y1CRY0CB : Y1CrY0Cb • VDC4_YCCSWAP_CRY1CBY0 : CrY1CbY0 • VDC4_YCCSWAP_Y1CBY0CR : Y1CbY0Cr • VDC4_YCCSWAP_CBY1CRY0 : CbY1CrY0 本パラメータは、res_md が YCbCr422 の場合のみ参照されます。また、この設定はエンディアン制御(gr_endian_on)が ON の時のみ有効となります。
vdc4_YccExchgMode gr_cnv444_md	in	YCbCr422→YCbCr444 変換時の補間モード設定 • VDC4_YCC_444_HOLD : ホールド補間 • VDC4_YCC_444_AVERAGE : 平均値補間 本パラメータは、res_md が YCbCr422 の場合のみ参照されます。

2.3.5 VDC4_DestroySurface

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_DestroySruface(vdc4_LayerID id) ;	
引数	• [in]vdc4_LyerID id	レイヤ ID
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施

概要

本関数では以下の処理を行います。

- ID で指定されたサーフェスを終了
- 該当サーフェスが動作中(状態が"Run")の場合は、停止後に終了
- 該当サーフェスの状態を"Init"に遷移

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_GRAPHICS_1 :グラフィックス(1) • VDC4_SURFACE_GRAPHICS_2 :グラフィックス(2) • VDC4_SURFACE_GRAPHICS_3 :グラフィックス(3) • VDC4_SURFACE_VIDEO_1 :ビデオ

2.3.6 VDC4_RegistCallbackFunc

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_RegistCallbackFunc(vdc4_IntType type, void (*callback)(vdc4_IntType), _UWORD line_num) ;	
引数	• [in]vdc4_IntType type • [in]void (*callback)(vdc4_IntType) • [in]_UWORD line_num	割り込みタイプ ユーザ定義コールバック関数ポインタ VLINE 割り込みのライン数
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- 指定された割り込みのコールバック関数を登録

コールバック関数は 1 つの割り込みタイプに対して 1 つだけ登録が可能です。既に登録済みの割り込みタイプが指定された場合、コールバック関数は上書き登録されます。また、コールバック関数ポインタに'0'が指定された場合は、登録が抹消されます。パネル出力ライン番号 `line_num` は、割り込みタイプが「パネル出力の指定ライン信号(VLINE)」の時のみ参照、設定されます。

登録されたコールバック関数は、割り込みハンドラから割り込みタイプを引数として呼び出されます。割り込みハンドラが登録されていない場合、本関数で登録したコールバック関数は呼び出されません。割り込みハンドラの登録については「1.7割り込みハンドラ」を参照してください。また、割り込みを発生させる為には、予め関数「VDC4_Initialize」にて使用する割り込みを有効にしておく必要があります。

登録されたコールバック関数はその割り込みタイプによって、どのようなサーフェスの状態で呼び出されるかが変化します(Table 12参照)。Table 12に示されていないサーフェスについては、コールバック関数の呼び出しが無いという事を表しています(例：グラフィックス(1)サーフェスが生成された時、VFIELDの割り込みコールバック関数は呼ばれません)。

Table 12 割り込みタイプとサーフェス状態の関係

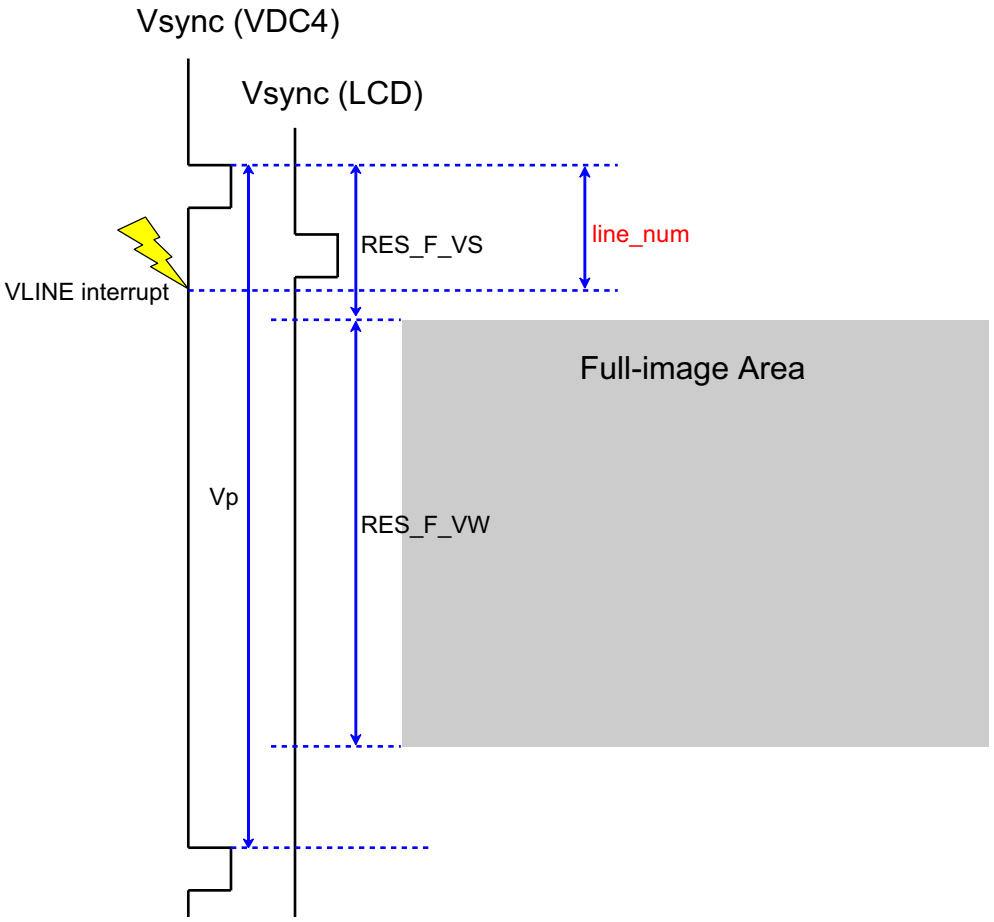
割り込み要因	サーフェス	サーフェスの状態		
		Init	Ready	Run
VI_VSYNC	ビデオ	x	o	o
LO_VSYNC	すべてのサーフェス	x	o	o
VSYNCERR	すべてのサーフェス	x	o	o
VLINE	すべてのサーフェス	x	x	o
VFIELD	ビデオ	x	x	o
VBUFERR1	ビデオ	x	x	o
VBUFERR2	グラフィックス(1)	x	x	o
VBUFERR3	グラフィックス(2)	x	x	o
VBUFERR4	グラフィックス(3)	x	x	o

【注】 "o":コールバック関数がコールされる。
"x":コールバック関数がコールされない。

引数の設定

型 引数名	入出力	説明
vdc4_IntType type	in	割り込みタイプ • VDC4_INT_TYPE_VI_VSYNC : VI_VSYNC • VDC4_INT_TYPE_LO_VSYNC : LO_VSYNC • VDC4_INT_TYPE_VSYNCERR : VSYNCERR • VDC4_INT_TYPE_VLINE : VLINE • VDC4_INT_TYPE_VFIELD : VFIELD • VDC4_INT_TYPE_VBUFERR1 : VBUFERR1 • VDC4_INT_TYPE_VBUFERR2 : VBUFERR2 • VDC4_INT_TYPE_VBUFERR3 : VBUFERR3 • VDC4_INT_TYPE_VBUFERR4 : VBUFERR4
void (*callback)(vdc4_IntType)	in	ユーザ定義コールバック関数ポインタ 必要でない場合、NULL を設定して下さい。 書式 void CallbackFunc(vdc4_IntType cb_type) ; 引数 • [in]vdc4_IntType 割り込みタイプ cb_type 戻り値 • void 概要 このコールバック関数は、指定された割り込みが発生した時に呼び出されます。この関数は引数に割り込みタイプを返します。
_UWORD line_num	in	VLINE 割り込みのライン数 割り込みタイプが VDC4_INT_TYPE_VLINE でない時、参照されません。

VLINE割り込みが発生するタイミングは、line_numによってライン数で指定されます。このライン数は、VDC4 内部の垂直同期信号を基準とします(Figure 10参照)。VDC4 内部の垂直同期信号とLCDの垂直同期信号や画像表示タイミングとの関係は、LCDパネル駆動用信号の設定や、フル画面イネーブル制御の設定によって決まります。これらの設定については「2.3.1 VDC4_Initialize」や「4.3 LCDの設定例」を参照ください。



Vsync (VDC4) : VDC4 内部垂直同期信号
Vsync (LCD) : LCD 垂直同期信号
RES_F_VS : フル画面の垂直イネーブル信号開始位置
RES_F_VW : フル画面の垂直イネーブル信号幅
Vp : 垂直同期信号周期
line_num : VLINE 割り込み発生タイミング

Figure 10 VLINE 割り込みの発生タイミング

2.3.7 VDC4_GraphicsStartSurface

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsStartSurface(vdc4_LayerID id, const vdc4_ShowGraphics *Graphics) ;	
引数	• [in]vdc4_LyerID id • [in]const vdc4_ShowGraphics * Graphics	レイヤ ID グラフィックスの表示設定
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- ID で指定されたグラフィックスサーフェスの表示を開始
- 指定されたパラメータを表示開始時の設定に反映
- ID で指定されたグラフィックスサーフェスの状態を"Ready"から"Run"に遷移

本関数を呼び出してから実際に表示が開始されるまで、最大で 1Vsync 時間掛かります。

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_GRAPHICS_1 : Graphics surface 1 • VDC4_SURFACE_GRAPHICS_2 : Graphics surface 2 • VDC4_SURFACE_GRAPHICS_3 : Graphics surface 3
vdc4_ShowGraphics * Graphics	in	グラフィックスの表示設定 NULL を設定しないで下さい。

vdc4_ShowGraphics 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_CtrlFrameBuff *CtrlFrameBuff ;
    vdc4_CtrlDispSel *CtrlDispSel ;
    vdc4_AreaRect *Gr_AreaRect ;
} vdc4_ShowGraphics ;
```

型 メンバ名	入出力	説明
vdc4_CtrlFrameBuff * CtrlFrameBuff	in	グラフィックスフレームバッファ制御 NULL が設定された場合、設定は変更されません。
vdc4_CtrlDispSel * CtrlDispSel	in	グラフィックス表示選択 NULL が設定された場合、設定は変更されません。
vdc4_AreaRect * Gr_AreaRect	in	表示領域 NULL が設定された場合、設定は変更されません。 2.1.3章のvdc4_AreaRect構造体を参照下さい。 Gr_AreaRect->hs は 16 クロック以上、Gr_AreaRect->hs + Gr_AreaRect->hw は 2015 クロック以内になるように設定してください。また、Gr_AreaRect->hw は 3 クロック以上を設定してください。 Gr_AreaRect->vs は 4 ライン以上、Gr_AreaRect->vs + Gr_AreaRect->vw は 2039 ライン以内になるように設定してください。

vdc4_CtrlFrameBuff 構造体のメンバは以下の通りです。

```
typedef struct {
    _SINT ChgFrameNum ;
    void *Buffer ;
} vdc4_CtrlFrameBuff ;
```

型 メンバ名	入出力	説明
_SINT ChgFrameNum	in	フレームバッファのフレーム番号 0 or 1 ダブルバッファを使用する場合のみ有効となります。グラフィックスサーフェスが生成される時に、フ

		レーム番号は、0 に初期化されます。
void * Buffer	in	フレームバッファのベースアドレス この API では、参照されません。

vdc4_CtrlDispSel 構造体のメンバは以下の通りです。

```
typedef struct {
   vdc4_DispSel gr_disp_sel ;
} vdc4_CtrlDispSel ;
```

型 メンバ名	入出力	説明
vdc4_DispSel gr_disp_sel	in	グラフィックスの表示設定 • VDC4_DISPSEL_BACK :背景色表示 • VDC4_DISPSEL_LOWER :下層グラフィックス表示 • VDC4_DISPSEL_CURRENT:カレントグラフィックス表示 • VDC4_DISPSEL_BLEND:下層グラフィックスとカレントグラフィックスのブレンド グラフィックス(1)では、下層グラフィックスとカレントグラフィックスのブレンド設定は禁止です。また、グラフィックスの表示設定は、各サーフェスの状態により Table 13のように変化します。

グラフィックスの表示設定は各サーフェスの状態により以下のように変化します。

Table 13　グラフィックス表示モード

State / Surface	Init	Ready	Run
Video	背景色表示	背景色表示	下層グラフィックス表示
Graphics (1)	背景色表示	背景色表示	ユーザ設定
Graphics (2)	下層グラフィックス表示	下層グラフィックス表示	ユーザ設定
Graphics (3)	下層グラフィックス表示	下層グラフィックス表示	ユーザ設定

グラフィックス(1)で拡大処理を行う場合は、"下層グラフィックス表示 (VDC4_DISPSEL_LOWER)"を設定する必要があります。

2.3.8 VDC4_VideoStartSurface

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_VideoStartSurface(vdc4_LayerID id) ;	
引数	• [in]vdc4_LyerID id	レイヤ ID
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施

概要

本関数では以下の処理を行います。

- ID で指定されたビデオサーフェスの表示/録画を開始
- ID で指定されたビデオサーフェスの状態を"Ready"から"Run"に遷移

本関数を呼び出してから実際に動作が開始されるまで、最大で 1Vsync 時間掛かります。

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_VIDEO_1 : ビデオサーフェス

2.3.9 VDC4_GraphicsChangeParam

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsChangeParam(vdc4_LayerID id, const vdc4_ShowGraphics *Graphics) ;	
引数	• [in]vdc4_LyerID id • [in]const vdc4_ShowGraphics * Graphics	レイヤ ID グラフィックスの表示設定
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- フレームバッファの変更 (アドレス若しくは番号による指定)
- 表示選択の変更
- 表示矩形領域の変更

本関数を呼び出してから実際に設定が反映されるまで、最大で 1Vsync 時間掛かります。

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_GRAPHICS_1 :グラフィックス(1) • VDC4_SURFACE_GRAPHICS_2 :グラフィックス(2) • VDC4_SURFACE_GRAPHICS_3 :グラフィックス(3)
vdc4_ShowGraphics * Graphics	in	グラフィックスの表示設定 NULL を設定しないで下さい。

vdc4_ShowGraphics 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_CtrlFrameBuff *CtrlFrameBuff ;
    vdc4_CtrlDispSel *CtrlDispSel ;
    vdc4_AreaRect *Gr_AreaRect ;
} vdc4_ShowGraphics ;
```

型 メンバ名	入出力	説明
vdc4_CtrlFrameBuff * CtrlFrameBuff	in	グラフィックスフレームバッファ制御 NULL が設定された場合、設定は変更されません。
vdc4_CtrlDispSel * CtrlDispSel	in	グラフィックスの表示設定 NULL が設定された場合、設定は変更されません。 2.3.7章のvdc4_CtrlDispSel構造体を参照下さい。
vdc4_AreaRect * Gr_AreaRect	in	表示領域 NULL が設定された場合、設定は変更されません。 2.1.3章のvdc4_AreaRect構造体を参照下さい。 Gr_AreaRect->hs は 16 クロック以上、Gr_AreaRect->hs + Gr_AreaRect->hw は 2015 クロック以内になるように設定してください。また、Gr_AreaRect->hw は 3 クロック以上を設定してください。 Gr_AreaRect->vs は 4 ライン以上、Gr_AreaRect->vs + Gr_AreaRect->vw は 2039 ライン以内になるように設定してください。

vdc4_CtrlFrameBuff 構造体のメンバは以下の通りです。

```
typedef struct {
    _SINT ChgFrameNum ;
    void *Buffer ;
} vdc4_CtrlFrameBuff ;
```

型 メンバ名	入出力	説明
_SINT ChgFrameNum	in	フレームバッファのフレーム番号 0 or 1 ダブルバッファを使用する場合のみ有効となります。

void * Buffer	in	フレームバッファのベースアドレス シングルバッファ時のみ有効となります。

2.3.10 VDC4_VideoChangeParam

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_VideoChangeParam(vdc4_LayerID id, const vdc4_ChangeVideo *ChgVideo) ;	
引数	• [in]vdc4_LyerID id • [in]const vdc4_ChangeVideo * ChgVideo	レイヤ ID ビデオ設定の変更
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- 画像取り込み範囲の変更
- 画像出力イネーブルの変更
- スケーリング／回転処理のパラメータ変更

スケーリングや回転処理の設定を変更する際には、フレームバッファのオーバフローに注意する必要があります。ビデオサーフェス生成時に十分なサイズのメモリが確保されていない場合、本関数の設定により入力映像が正しく取り込めなくなる可能性があります。ビデオ入力映像のフレームバッファについては「2.3.4 VDC4_VideoCreateSurface」を参照ください。

本関数を呼び出してから実際に設定が反映されるまで、最大で 1Vsync 時間掛かります。

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_VIDEO_1 :ビデオ
vdc4_ChangeVideo * ChgVideo	in	ビデオ設定の変更 NULL を設定しないで下さい。

vdc4_ChangeVideo 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_AreaRect *res ;
    vdc4_AreaRect *res_p ;
    vdc4_ScalingRot *ScaleRot ;
} vdc4_ChangeVideo ;
```

型 メンバ名	入出力	説明
vdc4_AreaRect * res	in	画像取り込み範囲設定 NULL が設定された場合、設定は変更されません。 2.1.3章のvdc4_AreaRect構造体を参照下さい。 res->hs は 16 クロック以上、res->hs + res->hw は 2015 クロック以内になるように設定してください。また、res->hw は 4 クロックアライメントで設定してください。 res->vs は 4 ライン以上、res->vs + res->vw は 2039 ライン以内になるように設定してください。また、res->vw は 4 ラインアライメントで設定してください。
vdc4_AreaRect * res_p	in	画像出力イネーブル範囲設定 NULL が設定された場合、設定は変更されません。 2.1.3章のvdc4_AreaRect構造体を参照下さい。 res_p->hs は 16 クロック以上、res_p->hs + res_p->hw は 2015 クロック以内になるように設定してください。また、res_p->hw は 4 クロックアライメントで設定してください。 res_p->vs は 4 ライン以上、res_p->vs + res_p->vw は 2039 ライン以内になるように設定してください。また、res_p->vw は 4 ラインアライメントで設定してください。
vdc4_ScalingRot * ScaleRot	in	スケーリング／回転パラメータ NULL が設定された場合、設定は変更されません。 2.3.4章のvdc4_ScalingRot構造体を参照下さい。

2.3.11 VDC4_StopSurface

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_StopSurface(vdc4_LayerID id) ;	
引数	• [in]vdc4_LyerID id	レイヤ ID
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施

概要

本関数では以下の処理を行います。

- ID で指定されたサーフェスの動作停止
- ID で指定されたサーフェスの表示選択を切り替え
- ID で指定されたサーフェスの状態を"Run"から"Ready"に遷移

ビデオサーフェスの動作停止は、フレームバッファに対する書き込みと読み出し処理を停止することで行われます。グラフィックスサーフェスでは、フレームバッファの読み出し処理の停止が行われます。停止処理で行われる表示選択の切り替えについては Table 13にて示される、サーフェスの状態と表示選択の関係を参照してください。

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_GRAPHICS_1 :グラフィックス(1) • VDC4_SURFACE_GRAPHICS_2 :グラフィックス(2) • VDC4_SURFACE_GRAPHICS_3 :グラフィックス(3) • VDC4_SURFACE_VIDEO_1 : ビデオ

2.3.12 VDC4_ImageColorMatrix

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_ImageColorMatrix(const vdc4_ColorMatrix *mtx) ;	
引数	• [in]const vdc4_ColorMatrix *mtx	カラーマトリクスパラメータ
戻り値	• vdc4_ErrorCode	エラーコード
	VDC4_ERR_NONE	正常終了
	VDC4_ERR_SURFACE_STATUS	サーフェスの状態に合わない処理を実施
	VDC4_ERR_PARAM_RANGE	パラメータに範囲外の値を設定
	VDC4_ERR_PARAM_OTHERS	その他のエラー

概要

本関数では以下の処理を行います。

- 指定されたカラーマトリクスの設定

ユーザが設定できるカラーマトリクスは入力制御部の出力部とグラフィックス(1)出力後の画質改善部の 2 つあります(Figure 11参照)。

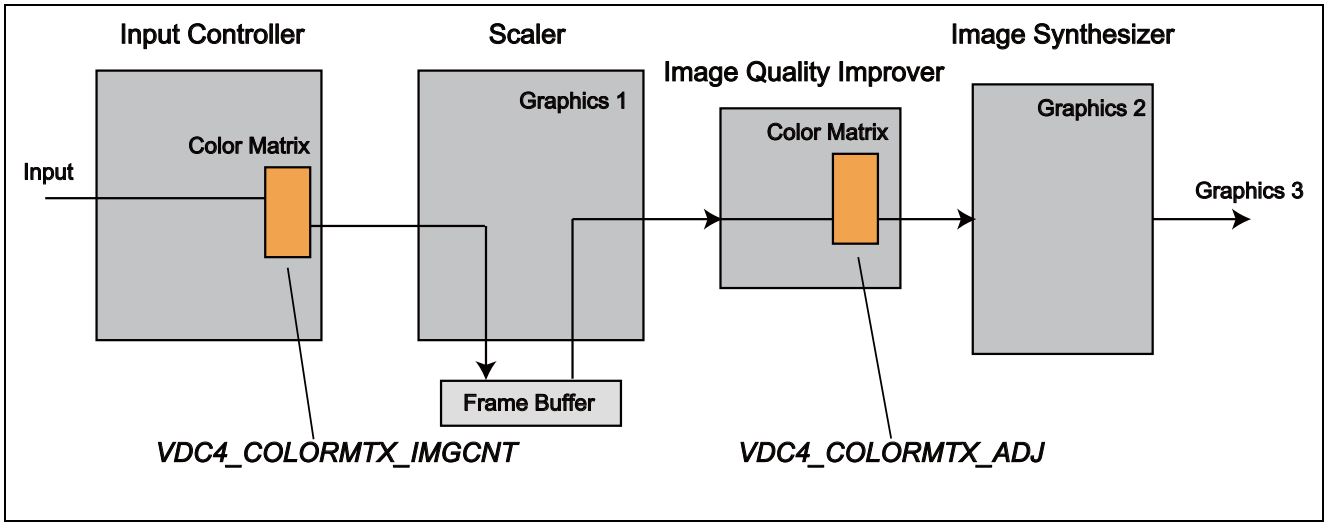


Figure 11 カラーマトリクス

Figure 11のように、マトリクス"VDC4_COLORMTX_IMGCNT"はビデオ入力映像に対して動作し、マトリクス"VDC4_COLORMTX_ADJ"はビデオ入力映像とグラフィックス(1)出力に対して動作します。

ビデオサーフェスやグラフィックス(1)サーフェスを使用するとき、必ずしもユーザがカラーマトリクスを設定する必要はありません。カラーマトリクスはビデオサーフェスやグラフィックス(1)サーフェスの生成時に、初期化されます。カラーマトリクスの初期化に関しては以下の Table 14と Table 15を参照ください。

引数の設定

型 引数名	入出力	説明
vdc4_ColorMatrix * mtx	in	カラーマトリクスパラメータ NULL を設定しないで下さい。

vdc4_ColorMatrix 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_ColorMtxModule module ;
    vdc4_ColorMtxMode mode ;
    _UWORD DcOffset[ VDC4_COLORMTX_OFS_NUM ] ;
    _UWORD matrix[ VDC4_COLORMTX_GAIN_NUM ] ;
} vdc4_ColorMatrix ;
```

型 メンバ名	入出力	説明
vdc4_ColorMtxModule module	in	カラーマトリクスモジュール選択(Figure 11) • VDC4_COLORMTX_IMGCNT : 入力制御部の画質調整部カラーマトリクス • VDC4_COLORMTX_ADJ : スケーリング(グラフィックス(1))出力に対する画質改善部のカラーマトリクス
vdc4_ColorMtxMode mode	in	カラーマトリクス動作モード • VDC4_COLORMTX_GBR_GBR : GBR => GBR • VDC4_COLORMTX_GBR_YCBCR : GBR => YCbCr • VDC4_COLORMTX_YCBCR_GBR : YCbCr => GBR • VDC4_COLORMTX_YCBCR_YCBCR : YCbCr => YCbCr module に VDC4_COLORMTX_ADJ を指定する場合、VDC4_COLORMTX_GBR_YCBCR と VDC4_COLORMTX_YCBCR_YCBCR は設定しないで下さい。
_UWORD DcOffset[VDC4_COLORMTX_OFS_NUM]	in	Y/G,B,R 信号のオフセット調整 符号無し (0 (-128) ~ 0x00FF (+127))
_UWORD matrix[VDC4_COLORMTX_GAIN_NUM]	in	GG, GB, GR, BG, BB, BR, RG, RB , RR のゲイン調整 符号付(2 の歩数) (-1024 ~ +1023, 256 = 1.0 倍)

Table 14は 2 つのカラーマトリクスが、ビデオサーフェスやグラフィックス(1)サーフェスの生成時にどのような動作モードで初期化されるのかを示したものです。カラーマトリクスのYCbCrとRGB変換の動作モードは、ビデオサーフェスやグラフィックス(1)サーフェスの信号フォーマットに従い決定されます。

Table 15はカラーマトリクスの各動作モードに対して設定するカラーマトリクスのパラメータです。YCbCrとRGBの変換はSMPTE 293Mの仕様に準拠しています。

Table 14　カラーマトリクス初期設定

Surface	入力選択	外部入力フォーマットの選択	フレームバッファ書き込みフォーマット	フレームバッファ読み込みフォーマット	VDC4_COLOR-MTX_IMGCNT	VDC4_COLOR-MTX_ADJ
Video	ビデオデコーダ出力	-	YCbCr (YCbCr422)	-*	YCbCr to YCbCr	YCbCr to RGB
			RGB (RGB565, RGB888)	-*	YCbCr to RGB	RGB to RGB
	外部入力	YCbCr (BT656, BT601)	YCbCr (YCbCr422)	-*	YCbCr to YCbCr	YCbCr to RGB
			RGB (RGB565, RGB888)	-*	YCbCr to RGB	RGB to RGB
		RGB (RGB888, RGB666, RGB565)	YCbCr (YCbCr422)	-*	RGB to YCbCr	YCbCr to RGB
			RGB (RGB565, RGB888)	-*	RGB to RGB	RGB to RGB
Graphics 1	-	-	-	YCbCr (YCbCr422)	-	YCbCr to RGB
				RGB (RGB565, RGB888, ARGB1555, ARGB4444, ARGB8888, CLUT8, CLUT4, CLUT1)	-	RGB to RGB

【注】 * ビデオサーフェスのフレームバッファ書き込み/ 読み出しフォーマットは同じ形式です。

Table 15　カラーマトリクス行列 (初期値)

カラーマトリクスパラメータ	GBR to GBR	GBR to YCbCr	YCbCr to GBR	YCbCr to YCbCr
Y/G 信号のオフセット調整	128	128	128	128
B 信号のオフセット調整	128	128	128	128
R 信号のオフセット調整	128	128	128	128
Y/G 信号出力の Y/G 信号のゲイン調整	256	150	256	256
Y/G 信号出力の Cb/B 信号のゲイン調整	0	29	1960	0
Y/G 信号出力の Cr/R 信号のゲイン調整	0	77	1865	0
Cb/B 信号出力の Y/G 信号のゲイン調整	0	1963	256	0
Cb/B 信号出力の Cb/B 信号のゲイン調整	256	128	454	256
Cb/B 信号出力の Cr/R 信号のゲイン調整	0	2005	0	0
Cr/R 信号出力の Y/G 信号のゲイン調整	0	1941	256	0
Cr/R 信号出力の Cb/B 信号のゲイン調整	0	2027	0	0
Cr/R 信号出力の Cr/R 信号のゲイン調整	256	128	359	256

2.3.13 VDC4_VideoNoiseReduction

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_VideoNoiseReduction(vdc4_OnOff nr1d_on, const vdc4_NoiseReduction *nr) ;	
引数	• [in]vdc4_OnOff nr1d_on • [in]const vdc4_NoiseReduction *nr	ノイズリダクションのオン/オフ制御 ノイズリダクションのパラメータ設定
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- 水平ノイズリダクションの設定

水平ノイズリダクションの「動作モード(G/B/R mode or Y/Cb/Cr mode)」は、ビデオサーフェス生成時に関数「VDC4_VideoCreateSurface」で指定した入力信号の情報を元に、VDC4 ドライバ内部で自動的に判定及び設定されます。それ以外の情報は本関数を利用して設定する必要があります。本関数を使用しない場合、水平ノイズリダクションのパラメータはリセット時の初期値のままとなります。初期値については「SH7268 Group, SH7269 Group User's Manual: Hardware (R01UH0048JJ)」を参照ください。

引数の設定

型 引数名	入出力	説明
vdc4_OnOff nr1d_on	in	ノイズリダクションのオン/オフ制御 • VDC4_OFF • VDC4_ON
vdc4_NoiseReduction * nr	in	ノイズリダクションのパラメータ設定 NULL が設定された場合、設定は変更されません。

vdc4_NoiseReduction 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_NRparam y ;
    vdc4_NRparam cb ;
    vdc4_NRparam cr ;
} vdc4_NoiseReduction ;
```

型 メンバ名	入出力	説明
vdc4_NRparam y	in	Y/G 信号のノイズリダクションパラメータ
vdc4_NRparam cb	in	Cb/B 信号のノイズリダクションパラメータ
vdc4_NRparam cr	in	Cr/R 信号のノイズリダクションパラメータ

vdc4_NRparam 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_NRTap nr1d_tap ;
    _UWORD nr1d_th ;
    vdc4_NRGain nr1d_gain ;
} vdc4_NRparam ;
```

型 メンバ名	入出力	説明
vdc4_NRTap nr1d_tap	in	TAP 選択 • VDC4_NR_TAPSEL_1 : 1 画素隣接 • VDC4_NR_TAPSEL_2 : 2 画素隣接 • VDC4_NR_TAPSEL_3 : 3 画素隣接 • VDC4_NR_TAPSEL_4 : 4 画素隣接
_UWORD nr1d_th	in	コアリングの最大値(絶対値) 0 ~ 0x007F
vdc4_NRGain nr1d_gain	in	ノイズリダクションのゲイン調整 • VDC4_NR_GAIN_1_2 : 1/2 • VDC4_NR_GAIN_1_4 : 1/4 • VDC4_NR_GAIN_1_8 : 1/8 • VDC4_NR_GAIN_1_16 : 1/16

2.3.14 VDC4_ImageEnhancement

書式

```
#include "vdc4_api.h"
vdc4_ErrorCode VDC4_ImageEnhancement( vdc4_OnOff bkstr_on,
                                      const vdc4_Black *black,
                                      vdc4_OnOff shp_h_on,
                                      const vdc4_EnhanceSharp *sharp,
                                      vdc4_OnOff lti_h_on,
                                      const vdc4_EnhanceLTI *lti,
                                      const vdc4_AreaRect *EnhArea ) ;
```

引数

• [in]vdc4_OnOff bkstr_on	黒伸張のオン/オフ制御
• [in]const vdc4_Black *black	黒伸張パラメータ
• [in]vdc4_OnOff shp_h_on	シャープネスのオン/オフ制御
• [in]const vdc4_EnhanceSharp *sharp	シャープネスパラーメータ
• [in]vdc4_OnOff lti_h_on	LTI のオン/オフ制御
• [in]const vdc4_EnhanceLTI *lti	LTI パラメータ
• [in]const vdc4_AreaRect *EnhArea	エンハンサの有効領域

戻り値

• vdc4_ErrorCode	エラーコード
VDC4_ERR_NONE	正常終了
VDC4_ERR_SURFACE_STATUS	サーフェスの状態に合わない処理を実施
VDC4_ERR_PARAM_RANGE	パラメータに範囲外の値を設定
VDC4_ERR_PARAM_INVALID	無効なパラメータを設定
VDC4_ERR_PARAM_OTHERS	その他のエラー

概要

本関数では以下の処理を行います。

- 黒伸張 ON/OFF 設定
- 黒伸張のパラメータ設定
- シャープネス ON/OFF 設定
- シャープネスのパラメータ設定
- LTI 制御 ON/OFF 設定
- LTI のパラメータ設定
- エンハンサ領域指定

本関数では、各調整処理の ON/OFF 制御とパラメータ設定を分けて設定できます。パラメータ設定用のポインタに NULL を指定すると、調整用パラメータは変更されません。ハードウェアリセット後、一度もパラメータ設定が行われていない場合は、ハードウェアマニュアルに定められた初期値が適用されます。初期値については「SH7268 Group, SH7269 Group User's Manual: Hardware (R01UH0048JJ)」を参照ください。シャープネス処理や LTI のエンハンサ機能を利用する時は、必ず一度はエンハンサ領域を指定して下さい。

また、RGB 信号入力時に本関数を呼び出した場合はパラメータエラー(VDC4_ERR_PARAM_OTHERS)となります。

引数の設定

型 引数名	入出力	説明
vdc4_OnOff bkstr_on	in	黒伸張のオン/オフ制御 • VDC4_OFF • VDC4_ON
vdc4_Black * black	in	黒伸張パラメータ NULL が設定された場合、設定は変更されません。
vdc4_OnOff shp_h_on	in	シャープネスのオン/オフ制御 • VDC4_OFF • VDC4_ON
vdc4_EnhanceSharp * sharp	in	シャープネスパラーメータ NULL が設定された場合、設定は変更されません。
vdc4_OnOff lti_h_on	in	LTI のオン/オフ制御 • VDC4_OFF • VDC4_ON
vdc4_EnhanceLTI * lti	in	LTI パラメータ NULL が設定された場合、設定は変更されません。
vdc4_AreaRect * EnhArea	in	エンハンサの有効領域 NULL が設定された場合、設定は変更されません。 2.1.3章のvdc4_AreaRect構造体を参照下さい。 EnhArea->hs は 4 クロック以上になるように設定してください。 EnhArea->vs は 2 ライン以上になるように設定してください。

vdc4_Black 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD bkstr_st ;
    _UWORD bkstr_t1 ;
    _UWORD bkstr_t2 ;
    _UWORD bkstr_d ;
} vdc4_Black ;
```

型 メンバ名	入出力	説明
_UWORD bkstr_st	in	黒伸張の開始点指定 0(低) ~ 15(高)
_UWORD bkstr_t1	in	黒伸張の時定数(T1) 0(小) ~ 31(大)
_UWORD bkstr_t2	in	黒伸張の時定数(T2) 0(小) ~ 30(大)
_UWORD bkstr_d	in	黒伸張の深さ 0(浅) ~ 15(深)

vdc4_EnhanceSharp 構造体のメンバは以下の通りです。

```
typedef struct {
    _SINT shp_h2_lpf_sel ;
    vdc4_SharpnessCtrl HrzSharp[ VDC4_IMGENH_SHARP_NUM ] ;
} vdc4_EnhanceSharp ;
```

型 メンバ名	入出力	説明
_SINT shp_h2_lpf_sel	in	H2 エッジ検出前の折り返し除去用 LPF 選択 • 0 : LPF なし • 1 : LPF あり
vdc4_SharpnessCtrl HrzSharp[VDC4_IMGENH_SHARP_NUM]	in	水平シャープネス HrzSharp[VDC4_IMGENH_SHARP_H1] : H1 HrzSharp[VDC4_IMGENH_SHARP_H2] : H2 HrzSharp[VDC4_IMGENH_SHARP_H3] : H3

vdc4_SharpnessCtrl 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD shp_clip_o ;
    _UWORD shp_clip_u ;
    _UWORD shp_gain_o ;
    _UWORD shp_gain_u ;
    _UWORD shp_core ;
} vdc4_SharpnessCtrl ;
```

型 メンバ名	入出力	説明
_UWORD shp_clip_o	in	シャープネスの補正値クリップ(オーバーシュート側) 0x0000 ~ 0x00FF
_UWORD shp_clip_u	in	シャープネスの補正値クリップ(アンダーシュート側) 0x0000 ~ 0x00FF
_UWORD shp_gain_o	in	シャープネスのエッジ振幅値に対するゲイン設定(オーバーシュート側) 0x0000 (0 倍) ~ 0x0040 (1 倍) ~ 0x00FF (約 4 倍)
_UWORD shp_gain_u	in	シャープネスのエッジ振幅値に対するゲイン設定(アンダーシュート側) 0x0000 (0 倍) ~ 0x0040 (1 倍) ~ 0x00FF (約 4 倍)
_UWORD shp_core	in	シャープネスの能動範囲の指定 0x0000 ~ 0x007F

vdc4_EnhanceLTI 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD lti_h2_inc_zero ;
    _SINT lti_h2_lpf_sel ;
    _UWORD lti_h2_gain ;
    _UWORD lti_h2_core ;
    _UWORD lti_h4_inc_zero ;
    _SINT lti_h4_median_tap_sel ;
    _UWORD lti_h4_gain ;
    _UWORD lti_h4_core ;
} vdc4_EnhanceLTI ;
```

型 メンバ名	入出力	説明
_UWORD lti_h2_inc_zero	in	メディアンフィルタの LTI 補正スレッシュ設定 (H2) 0x0000 ~ 0x00FF
_SINT lti_h2_lpf_sel	in	H2 エッジ検出前の折り返し除去用 LPF 選択 • 0 : LPF なし • 1 : LPF あり
_UWORD lti_h2_gain	in	LTI のエッジ振幅値に対するゲイン設定 (H2) 0x0000 (0 倍) ~ 0x0040 (1 倍) ~ 0x00FF (約 4 倍)
_UWORD lti_h2_core	in	LTI のコアリング(コアリング量は最大 255) (H2) 0x0000 ~ 0x00FF
_UWORD lti_h4_inc_zero	in	メディアンフィルタの LTI 補正スレッシュ設定 (H4) 0x0000 ~ 0x00FF
_SINT lti_h4_median_tap_sel	in	メディアンフィルタの参照画素選択 • 0 : 隣接 2 画素目参照 • 1 : 隣接 1 画素目参照
_UWORD lti_h4_gain	in	LTI のエッジ振幅値に対するゲイン設定 (H4) 0x0000 (0 倍) ~ 0x0040 (1 倍) ~ 0x00FF (約 4 倍)
_UWORD lti_h4_core	in	LTI のコアリング(コアリング量は最大 255) (H4) 0x0000 ~ 0x00FF

2.3.15 VDC4_GraphicsAlphaBlending

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsAlphaBlending(vdc4_LayerID id, vdc4_OnOff gr_arc_on, const vdc4_AlphaAttr *attr) ;	
引数	• [in]vdc4_LyerID id • [in]vdc4_OnOff gr_arc_on • [in]const vdc4_AlphaAttr *attr	レイヤ ID 矩形領域アルファブレンドオン/オフ制御 アルファブレンドパラメータ
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- 矩形領域アルファブレンドの設定
- 矩形領域アルファブレンドの領域設定

矩形領域アルファブレンド表示処理の矩形領域は、グラフィックスサーフェス生成時に表示領域と同じ領域で初期化されます。

矩形領域アルファブレンド表示処理で、フェードイン／フェードアウトを使用しない場合、αブレンディング ON 中にα値の変更はできません。α値を変更する際には一度 OFF にする必要があります。ブレンディングの ON と OFF の切り替えや、α値が設定してから反映されるまでには 1Vsync 時間掛かります。

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_GRAPHICS_2 :グラフィックス(2) • VDC4_SURFACE_GRAPHICS_3 :グラフィックス(3)
vdc4_OnOff gr_arc_on	in	矩形領域アルファブレンドオン/オフ設定 • VDC4_OFF • VDC4_ON
vdc4_AlphaAttr * attr	in	矩形領域アルファブレンドパラメータ NULL が設定された場合、設定は変更されません。

vdc4_AlphaAttr 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_AreaRect *gr_arc ;
    vdc4_AlphaFade *fade ;
} vdc4_AlphaAttr ;
```

型 メンバ名	入出力	説明
vdc4_AreaRect * gr_arc	in	矩形領域アルファブレンド処理の領域設定 NULL が設定された場合、設定は変更されません。 2.1.3章のvdc4_AreaRect構造体を参照下さい。
vdc4_AlphaFade * fade	in	矩形領域アルファブレンドパラメータ NULL が設定された場合、設定は変更されません。

vdc4_AlphaFade 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD gr_arc_def ;
    _SWORD gr_arc_coef ;
    _UWORD gr_arc_rate ;
} vdc4_AlphaFade ;
```

型 メンバ名	入出力	説明
_UWORD gr_arc_def	in	矩形領域アルファブレンド処理のアルファ初期値 0 ~ 255
_SWORD gr_arc_coef	in	矩形領域アルファブレンド処理のアルファ係数 -255 ~ 255
_UWORD gr_arc_rate	in	矩形領域アルファブレンド処理のフレームレート Vsync の立ち上がりエッジ数 Vsync が gr_arc_rate+1 回立ち上がるたびに gr_arc_coef の値がアルファ値に加算されます。 0 ~ 255

2.3.16 VDC4_GraphicsChromakey

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsChromakey(vdc4_LayerID id, vdc4_OnOff gr_ck_on, const vdc4_ChromaKeyAttr *attr) ;	
引数	• [in]vdc4_LyerID id • [in]vdc4_OnOff gr_ck_on • [in]const vdc4_ChromaKeyAttr *attr	レイヤ ID CLUT 参照/RGB 参照クロマキー処理オン/オフ設定 クロマキーパラメータ
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_BAD VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 不正なサーフェスに対して処理を実施 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- クロマキーの設定

クロマキーの設定は、置換対象色と置換後の色指定から成ります。本ドライバでは、対象となるグラフィックスサーフェスと同じカラーフォーマットで指定することとなっていますが、以下の例外が在ります。

- CLUT4、CLUT8 形式の時は、置換後の色指定を ARGB8888 形式とする。
- RGB565、RGB888 形式の時は、置換後の色指定のうちα値を別に指定する。
- ARGB1555 形式の時は、α値のクロマキーによる置換はサポートしない。

本ドライバでは、同じグラフィックスサーフェスに対して矩形領域アルファブレンド表示処理が行われている場合、矩形領域のアルファブレンド処理が優先されます。アルファブレンド処理で指定した矩形領域以外の場所では、クロマキー処理が行われます。

また、カラーフォーマットが CLUT1 のグラフィックスサーフェスに対するクロマキー処理は禁止です。

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_GRAPHICS_2 :グラフィックス(2) • VDC4_SURFACE_GRAPHICS_3 :グラフィックス(3)
vdc4_OnOff gr_ck_on	in	CLUT 参照/RGB 参照クロマキー処理オン/オフ設定 • VDC4_OFF • VDC4_ON
vdc4_ChromaKeyAttr * attr	in	クロマキーパラメータ NULL が設定された場合、設定は変更されません。

vdc4_ChromaKeyAttr 構造体のメンバは以下の通りです。

```
typedef struct {
    _UDWORD ck_color ;
    _UDWORD rep_color ;
    _UDWORD rep_alpha ;
} vdc4_ChromaKeyAttr ;
```

型 メンバ名	入出力	説明
_UDWORD ck_color	in	クロマキー処理対象 RGB/CLUT 信号 ターゲットとなるグラフィックスサーフェスと同じカラーフォーマットで指定して下さい(LSB詰め)。カラーフォーマットの詳細については、Figure 12を参照して下さい。
_UDWORD rep_color	in	クロマキー処理置換後 ARGB 信号 ターゲットとなるグラフィックスサーフェスのカラーフォーマットが CLUT4 か CLUT8 の場合、ARGB8888 フォーマットで値を指定します。それ以外の場合、グラフィックスサーフェスと同じカラーフォーマットで値を指定します。詳細は、Figure 12を参照して下さい。
_UDWORD rep_alpha	in	クロマキー処理置換後アルファ信号 ターゲットとなるグラフィックスサーフェスのカラーフォーマットがアルファ値を持たないRGB565 もしくはRGB888 の場合、LSB詰めでアルファ値を指定して下さい。詳細は、Figure 12を参照して下さい。

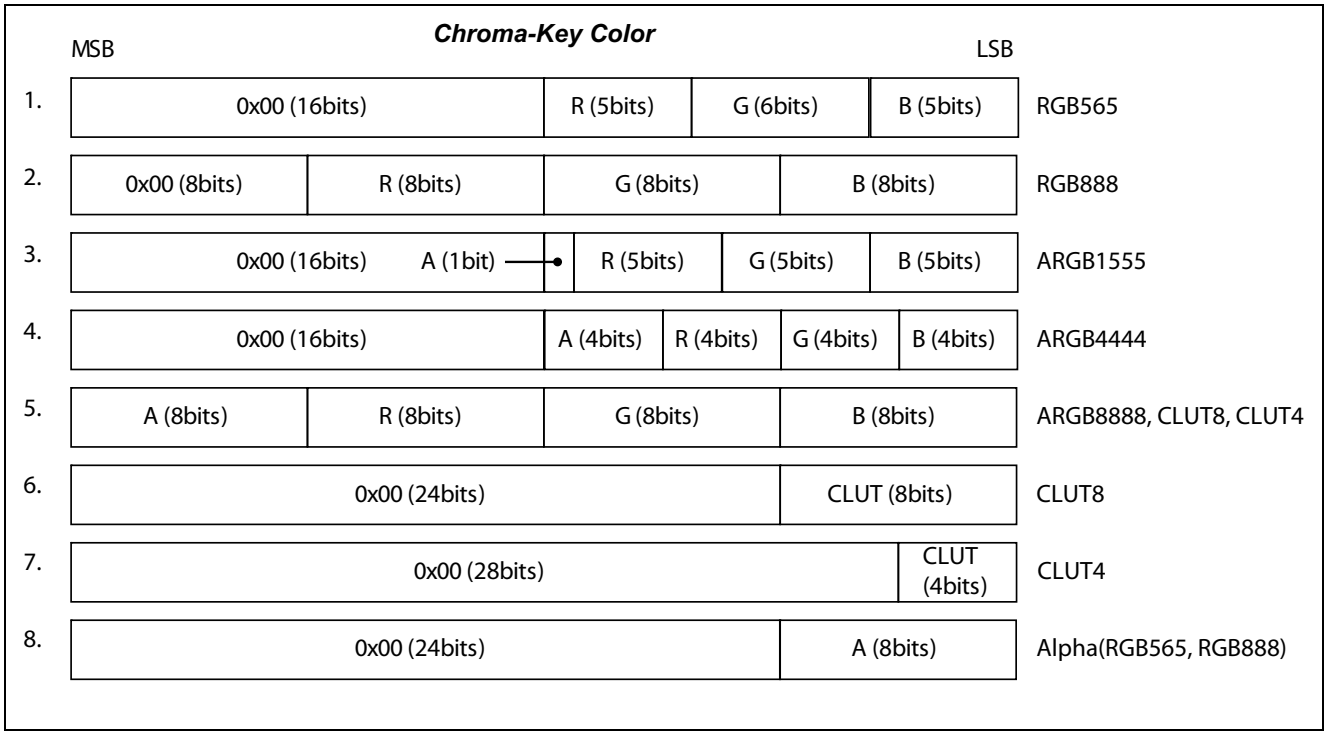


Figure 12　クロマキー処理の色指定

Figure 12はクロマキー処理の色指定データフォーマットについて表したものです。RGB565、RGB888、ARGB1555、ARGB4444、ARGB8888 形式の場合は`ck_color`と`rep_color`共に図のように指定します(図中 1 から 5 参照)。ただし、RGB565 とRGB888 にはα値が無いので、置換後の色のα値を別に`rep_alpha`で図中8のように指定します。CLUT8 とCLUT4形式では`ck_color`をそれぞれ図中6と7のように指定しますが、`rep_color`の指定は図中 5 のようにARGB8888 形式で指定します。

2.3.17 VDC4_GraphicsSetCLUT

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_GraphicsSetCLUT(vdc4_LayerID id, const vdc4_CLUT *table) ;	
引数	• [in]vdc4_LyerID id	レイヤ ID
	• [in]const vdc4_CLUT *table	CLUT パラメータ
戻り値	• vdc4_ErrorCode	エラーコード
	VDC4_ERR_NONE	正常終了
	VDC4_ERR_SURFACE_BAD	不正なサーフェスに対して処理を実施
	VDC4_ERR_SURFACE_STATUS	サーフェスの状態に合わない処理を実施
	VDC4_ERR_PARAM_RANGE	パラメータに範囲外の値を設定
	VDC4_ERR_PARAM_UNDEF	必要なパラメータに NULL を設定
	VDC4_ERR_PARAM_OTHERS	その他のエラー

概要

本関数では以下の処理を行います。

- ID で指定されたグラフィックスサーフェスの CLUT の設定
- ID で指定されたグラフィックスサーフェスが CLUT 形式の時、CLUT データの設定
- ID で指定されたグラフィックスサーフェスが ARGB1555 の時、α 値の設定

CLUT 形式や ARGB1555 形式のグラフィックスサーフェスを使用する場合には、グラフィックスサーフェス生成後に必ず 1 度は本関数を使用する必要があります。

CLUTの設定後に、関数「VDC4_DestroySurface」によりサーフェスが破棄された場合、設定されたCLUTの値は保証されません。同じグラフィックスサーフェスを生成した後に、再び本関数でCLUTを設定してください。

CLUT形式でグラフィックスサーフェスの表示を行っている状態から、関数「VDC4_StopSurface」によりグラフィックスサーフェスの表示を停止した場合はCLUTの設定は保持されています。その後、本関数を呼び出さずに関数「VDC4_GraphicsStartSurface」を呼び出して再表示をすると、表示停止前のCLUTの設定で色が表示されます。

CLUT の設定が反映されるには最大で 1Vsync 時間掛かります。特に表示中の CLUT 設定には注意してください。以下に示される関数の呼出し後に本関数を呼び出す場合には、1Vsync 時間程度あける必要があります。

- 「VDC4_GraphicsStartSurface」グラフィックスサーフェス表示開始処理
- 「VDC4_StopSurface」サーフェス停止処理
- 「VDC4_GraphicsSetCLUT」本関数、CLUT設定処理

引数の設定

型 引数名	入出力	説明
vdc4_LayerID id	in	レイヤ ID • VDC4_SURFACE_GRAPHICS_1 :グラフィックス(1) • VDC4_SURFACE_GRAPHICS_2 :グラフィックス(2) • VDC4_SURFACE_GRAPHICS_3 :グラフィックス(3)
vdc4_CLUT * table	in	CLUT パラメータ NULL を設定しないで下さい。

vdc4_CLUT 構造体のメンバは以下の通りです。

```
typedef struct {
    _SINT color_num ;
    _UDWORD *clut ;
} vdc4_CLUT ;
```

型 メンバ名	入出力	説明
_SINT color_num	in	CLUT の色の数 CLUT4 0 ~ 16 CLUT8 0 ~ 256 CLUT1 / ARGB1555 参照しません。CLUT1 と ARGB1555 フォーマットの時は、clut に 2 色のデータ指定してください。
_UDWORD * clut	in	ARGB8888 フォーマットの CLUT データポインタ NULLを設定しないでください(Figure 13参照)。

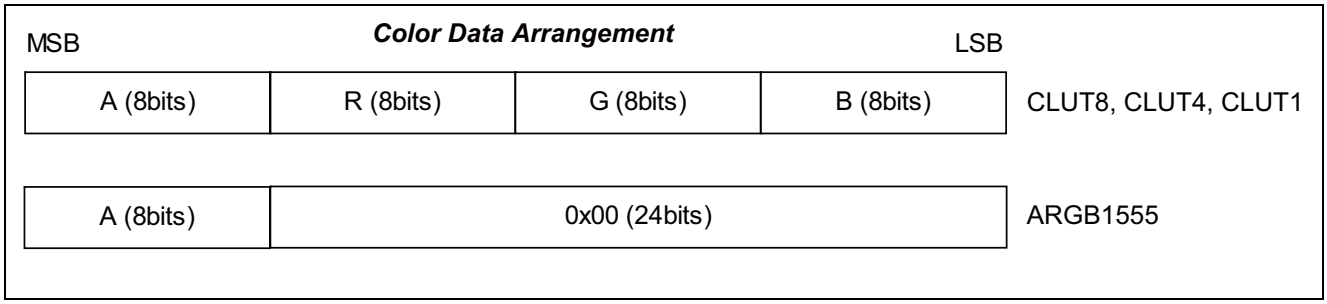


Figure 13　CLUT のデータ形式

Figure 13はCLUTデータの形式を表したものです。ARGB1555 形式の時は上位 8bitにα値を指定します。

CLUT1 と ARGB1555 形式の時、clut へ 2 色のデータが格納されたポインタを設定します。この 2 色の色は格納されている順に、CLUT1 の'0'と'1'に対応します。ARGB1555 形式の時は、格納されている順に A が'0'と'1'の時のα値となります。

2.3.18 VDC4_DisplayCalibration

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_DisplayCalibration(const vdc4_DispCalibration *param) ;	
引数	• [in]const vdc4_DispCalibration *param	表示キャリブレーションパラメータ
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE VDC4_ERR_PARAM_INVALID	エラーコード 正常終了 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定 無効なパラメータを設定

概要

本関数では以下の処理を行います。

- ガンマ補正の ON/OFF 設定
- パネルブライト調整のパラメータ設定
- コントラスト調整のパラメータ設定
- パネルディザ補正のパラメータ設定
- 補正回路の順番制御設定

本関数ではガンマ補正のON/OFF制御のみが可能です。ガンマ補正処理のパラメータ設定には関数「2.3.19VDC4_DisplaySetGammaCorrectionTable」を呼び出す必要があります。

「パネルディザ出力フォーマット選択」の設定は関数「VDC4_Initialize」にて指定した出力フォーマットに従い、VDC4 ドライバが自動的に設定します。

引数の設定

型 引数名	入出力	説明
vdc4_DispCalibration * param	in	表示キャリブレーションパラメータ NULL を設定しないで下さい。

vdc4_DispCalibration 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_CalibrationPath path ;
    vdc4_OnOff gam_on ;
    vdc4_Calibr_Bright *bright ;
    vdc4_Calibr_Contrast *contrast ;
    vdc4_Calibr_Dither *panel_dither ;
} vdc4_DispCalibration ;
```

型 メンバ名	入出力	説明
vdc4_CalibrationPath path	in	補正回路の順番の制御 • VDC4_CALIBR_BCG : ブライトネス->コントラスト->ガンマ補正 • VDC4_CALIBR_GBC : ガンマ補正->ブライトネス->コントラスト
vdc4_OnOff gam_on	in	ガンマ補正オン/オフ制御 • VDC4_OFF • VDC4_ON
vdc4_Calibr_Bright * bright	in	ブライトネス DC パラメータ NULL が設定された場合、設定は変更されません。
vdc4_Calibr_Contrast * contrast	in	コントラストゲインパラメータ NULL が設定された場合、設定は変更されません。
vdc4_Calibr_Dither * panel_dither	in	パネルディザパラメータ NULL が設定された場合、設定は変更されません。

vdc4_Calibr_Bright 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD pbrt_g ;
    _UWORD pbrt_b ;
    _UWORD pbrt_r ;
} vdc4_Calibr_Bright ;
```

型 メンバ名	入出力	説明
_UWORD pbrt_g	in	G 信号のブライト(DC)調整 0x0000(-512) ~ 0x03FF(+511)
_UWORD pbrt_b	in	B 信号のブライト(DC)調整 0x0000(-512) ~ 0x03FF(+511)
_UWORD pbrt_r	in	R 信号のブライト(DC)調整 0x0000(-512) ~ 0x03FF(+511)

vdc4_Calibr_Contrast 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD cont_g ;
    _UWORD cont_b ;
    _UWORD cont_r ;
} vdc4_Calibr_Contrast ;
```

型 メンバ名	入出力	説明
_UWORD cont_g	in	G 信号のコントラスト(ゲイン)調整 0x0000(0/128 倍) ~ 0x00FF(255/128 倍)
_UWORD cont_b	in	B 信号のコントラスト(ゲイン)調整 0x0000(0/128 倍) ~ 0x00FF(255/128 倍)
_UWORD cont_r	in	R 信号のコントラスト(ゲイン)調整 0x0000(0/128 倍) ~ 0x00FF(255/128 倍)

vdc4_Calibr_Contrast 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_PDthMode pdth_sel ;
    _UWORD pdth_pa ;
    _UWORD pdth_pb ;
    _UWORD pdth_pc ;
    _UWORD pdth_pd ;
} vdc4_Calibr_Contrast ;
```

型 メンバ名	入出力	説明
vdc4_PDthMode pdth_sel	in	パネルディザ動作モード • VDC4_PDTH_MD_TRU : 切り捨て • VDC4_PDTH_MD_RDOF : 四捨五入 • VDC4_PDTH_MD_2X2 : 2 x 2 パターンディザ • VDC4_PDTH_MD_RAND :ランダムパターンディザ
_UWORD pdth_pa	in	2x2 パターンディザのパターン値(A) 0 ~ 3

		pdth_sel が VDC4_PDTH_MD_2X2 の時参照されます。
_UWORD pdth_pb	in	2x2 パターンディザのパターン値(B) 0 ~ 3 pdth_sel が VDC4_PDTH_MD_2X2 の時参照されます。
_UWORD pdth_pc	in	2x2 パターンディザのパターン値(C) 0 ~ 3 pdth_sel が VDC4_PDTH_MD_2X2 の時参照されます。
_UWORD pdth_pd	in	2x2 パターンディザのパターン値(D) 0 ~ 3 pdth_sel が VDC4_PDTH_MD_2X2 の時参照されます。

2.3.19 VDC4_DisplaySetGammaCorrectionTable

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_DisplaySetGammaCorrectionTable(const vdc4_GammaCorrection *table) ;	
引数	• [in]const vdc4_GammaCorrection * table	ガンマ補正パラメータ
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE	エラーコード 正常終了 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定

概要

本関数では以下の処理を行います。

- ガンマ補正用のパラメータを G/B/R 毎に設定します。

本関数はガンマ補正で参照されるパラメータの設定のみを行います。ガンマ補正処理のON/OFFを切り替える為には関数「2.3.18VDC4_DisplayCalibration」を呼び出す必要があります。

本関数を呼び出さない場合、ガンマ補正パラメータはリセット時の初期値のままとなります。初期値については「SH7268 Group, SH7269 Group User's Manual: Hardware (R01UH0048JJ)」を参照ください。

引数の設定

型 引数名	入出力	説明
vdc4_GammaCorrection * table	in	ガンマ補正パラメータ NULL を設定しないで下さい。

vdc4_GammaCorrection 構造体のメンバは以下の通りです。

```
typedef struct {
    vdc4_GamCrrct *gam_g ;
    vdc4_GamCrrct *gam_b ;
    vdc4_GamCrrct *gam_r ;
} vdc4_GammaCorrection ;
```

型 メンバ名	入出力	説明
vdc4_GamCrrct * gam_g	in	G 信号のガンマ補正パラメータ NULL が設定された場合、設定は変更されません。
vdc4_GamCrrct * gam_b	in	B 信号のガンマ補正パラメータ NULL が設定された場合、設定は変更されません。
vdc4_GamCrrct * gam_r	in	R 信号のガンマ補正パラメータ NULL が設定された場合、設定は変更されません。

vdc4_GamCrrct 構造体のメンバは以下の通りです。

```
typedef struct {
    _UWORD gam_th[ VDC4_GAM_TH_NUM ] ;
    _UWORD gam_gain[ VDC4_GAM_GAIN_NUM ] ;
} vdc4_GamCrrct ;
```

型 メンバ名	入出力	説明
_UWORD gam_th[VDC4_GAM_TH_NUM]	in	GBR 信号の開始閾値テーブル 0 ~ 255 VDC4_GAM_TH_NUM は 31 です。
_UWORD gam_gain[VDC4_GAM_GAIN_NUM]	in	GBR 信号のゲイン調整テーブル 0x0000 ~ 0x07FF (0x0400 = 1.0 倍) VDC4_GAM_GAIN_NUM は 32 です。

2.3.20 VDC4_SwitchVsync

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_SwitchVsync(vdc4_OnOff FreeRunVsync, const vdc4_SyncPeriod *SyncPeriod) ;	
引数	• [in]vdc4_OnOff FreeRunVsync • [in]const vdc4_SyncPeriod * SyncPeriod	自走用垂直同期信号のオン/オフ設定 同期信号の周期設定パラメータ
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_STATUS VDC4_ERR_PARAM_RANGE	エラーコード 正常終了 サーフェスの状態に合わない処理を実施 パラメータに範囲外の値を設定

概要

本関数では以下の処理を行います。

- スケーリング部より出力する垂直同期信号を切り替えます(自走／外部入力)。
- 同期信号の周期を設定します。

本関数ではスケーリング部から出力する垂直同期信号を選択し、その周期を設定することができます。この設定は、関数「VDC4_Initialize (2.3.1参照)」にて初期設定されるので、必ずしも行う必要はありません。

本関数の呼び出しにより表示中に垂直同期信号を切り替えると表示が乱れる可能性がありますが、本ドライバではこの表示の乱れを補償しません。垂直同期信号切り替えのタイミングについては、ユーザが考慮して行う必要があります。

引数の設定

型 引数名	入出力	説明
vdc4_OnOff FreeRunVsync	in	自走用垂直同期信号のオン/オフ設定 • VDC4_OFF : 外部入力垂直同期信号 • VDC4_ON : 内部生成した自走用垂直同期信号
vdc4_SyncPeriod * SyncPeriod	in	同期信号の周期設定パラメータ NULL が設定された場合、設定は変更されません。

vdc4_SyncPeriod 構造体のメンバは以下の通りです。

```
typedef struct {
    __UWORD res_fv ;
    __UWORD res_fh ;
} vdc4_SyncPeriod ;
```

型 メンバ名	入出力	説明
_UWORD res_fv	in	自走用垂直同期信号の周期設定 Vsync 周期 = (res_fv + 1) x Hsync 周期 0x0000 ~ 0x07FF [lines] FreeRunVsync が VDC4_OFF に設定された時は、参照されません。
_UWORD res_fh	in	水平同期信号の周期設定 Hsync 周期 = (res_fh + 1) / ピクセルクロック周波数 0x0000 ~ 0x07FF [pixel clock cycles]

2.3.21 VDC4_CheckClock

書式

```
#include "vdc4_api.h"
vdc4_ErrorCode VDC4_CheckClock( vdc4_PanelClockSel PanelClockSel,
                                vdc4_PanelClkDCDR Dcdr,
                                _UDWORD SrcClkFreq,
                                void (*WaitMsec)( _UDWORD )) ;
```

引数	• [in] vdc4_PanelClockSel PanelClockSel	パネルクロック供給源選択
	• [in]vdc4_PanelClkDCDR Dcdr	クロック分周比設定
	• [in]_UDWORD SrcClkFreq	ソースクロックの周波数
	• [in]void (*WaitMsec)(_UDWORD)	ユーザ定義 wait 関数のポインタ
戻り値	• vdc4_ErrorCode	エラーコード
	VDC4_ERR_NONE	正常終了
	VDC4_ERR_SURFACE_STATUS	サーフェスの状態に合わない処理を実施
	VDC4_ERR_PARAM_UNDEF	必要なパラメータに NULL を設定
	VDC4_ERR_PARAM_INVALID	無効なパラメータを設定
	VDC4_ERR_SYSTEM_PNLCLK	パネルクロック異常検出

概要

本関数では以下の処理を行います。

- パネルクロック供給源選択とクロック分周比の設定
- 自走用同期信号の設定
- 垂直同期信号の多発マスクと欠落補償を OFF に設定
- TCON レジスタ更新制御レジスタ(TCON_UPDATE)の設定
- 自走周期とソースクロックの周波数から wait 時間を算出し、ユーザ定義 wait 関数を利用して wait
- wait 後の TCON レジスタ更新制御レジスタの状態からパネルクロックの異常検出

本関数はVDC4 モジュールにてパネルクロックが正しく動作しているかを確認する為のAPIです。パネルクロックの動作確認と、異常検出時のリトライ処理については「4.2.2パネルクロックの異常検出とリトライ処理」を参考にしてください。また、以下に示した条件ではパネルクロックの異常動作が発生しない為、動作確認の必要はありません。

Table 16 パネルクロック正常動作条件

クロック分周比設定	VIDEO_X1 端子	パネルクロック供給源選択
1, 2, 5, 9 分周	任意	任意
7 分周	入力なし(端子固定)	映像クロック選択(VIDEO_X1)
		周辺クロック 1(P1φ)
その他(1, 2, 5, 9, 7 分周以外)	入力あり	映像クロック選択(VIDEO_X1)
	入力なし(端子固定)	映像クロック選択(VIDEO_X1)
		周辺クロック 1(P1φ)

本関数は、ハードウェアリセット後最初に呼び出す必要があります。その為、本関数を呼び出す際には必要に応じて予め以下のようなクロックやポートの設定を行っておく必要があります。ただし、LCD の同期信号やイネーブル信号のポート出力設定は行わない事を推奨します。本関数ではパネルクロックの動作確認を行う為に、自走同期信号の設定を行います。ポートを出力設定にしていると、LCD に対して不正な信号が出力される可能性があります。

- スタンバイコントロールレジスタの設定により VDC4 モジュールへクロック供給
- スタンバイコントロールレジスタの設定によりビデオデコーダモジュールへクロック供給
- 汎用 I/O ポートレジスタの設定により外部クロック(LCD_EXTCLK)や映像クロック(DV_CLK)を入力

本関数を利用したパネルクロックの動作確認のためには、wait処理が必要です。wait処理は`WaitMsec`で指定されたユーザ定義関数により行われます。ユーザ定義関数の実装については「3.3VDC4_CheckClockのユーザ定義関数例」を参考にしてください。

引数の設定

型 引数名	入出力	説明
vdc4_PanelClockSel PanelClockSel	in	パネルクロック供給源選択 • VDC4_LCDPANEL_CLKSEL_IMG : 映像クロック選択(VIDEO_X1) • VDC4_LCDPANEL_CLKSEL_EXT : 外部クロック選択(LCDEXT_CLK) • VDC4_LCDPANEL_CLKSEL_PERI : 周辺クロック 1(P1φ) • VDC4_LCDPANEL_CLKSEL_IMG_DV : 映像クロック選択(DV_CLK)
vdc4_PanelClkDCDR Dcdr	in	クロック分周比設定 • VDC4_LCDPANEL_CLKDIV_1_1 : 1/1 • VDC4_LCDPANEL_CLKDIV_1_2 : 1/2 • VDC4_LCDPANEL_CLKDIV_1_3 : 1/3 • VDC4_LCDPANEL_CLKDIV_1_4 : 1/4 • VDC4_LCDPANEL_CLKDIV_1_5 : 1/5 • VDC4_LCDPANEL_CLKDIV_1_6 : 1/6 • VDC4_LCDPANEL_CLKDIV_1_7 : 1/7 • VDC4_LCDPANEL_CLKDIV_1_8 : 1/8 • VDC4_LCDPANEL_CLKDIV_1_9 : 1/9 • VDC4_LCDPANEL_CLKDIV_1_12 : 1/12 • VDC4_LCDPANEL_CLKDIV_1_16 : 1/16 • VDC4_LCDPANEL_CLKDIV_1_24 : 1/24 • VDC4_LCDPANEL_CLKDIV_1_32 : 1/32
_UDWORD SrcClkFreq	in	ソースクロックの周波数 [Hz] 例) 40MHz の時‘40000000’を設定
void (*WaitMsec)(_UDWORD)	in	ユーザ定義 wait 関数のポインタ ユーザにより実装された wait 関数のポインタです。引数で指定された時間[msec]の wait 処理を行います。 書式 void Wait_Msec(_UDWORD wait) ; 引数 • [in]_UDWORD wait　wait 時間 [msec] 戻り値 • void 概要 ユーザによって実装された wait 処理が実行されます。

2.3.22 VDC4_Reset

書式	#include "vdc4_api.h" vdc4_ErrorCode VDC4_Reset(void) ;	
引数	• [in]void	なし
戻り値	• vdc4_ErrorCode VDC4_ERR_NONE VDC4_ERR_SURFACE_STATUS	エラーコード 正常終了 サーフェスの状態に合わない処理を実施

概要

本関数では以下の処理を行います。

- VDC4 モジュールのソフトウェアリセット
- VDC4 モジュールのソフトウェアリセット解除

本関数を使用することにより、VDC4 モジュールはハードウェアリセット後と同じ状態になります。

3. ユーザ定義関数

本ドライバの「VDC4_Initialize」と「VDC4_Terminate」では、ユーザ定義関数を実行することができます。また、「VDC4_CheckClock」ではユーザ定義のwait関数を指定する必要があります。これらのユーザ定義関数の作成例を以下に示します。

3.1 VDC4_Initializeのユーザ定義関数例

```
/***********************************************************************//**
 * Function Name : Init_VDC4_CallBack
 * @brief
 * @param     [in]_UDWORD mode
 * @retval    void
 ***********************************************************************/
void Init_VDC4_CallBack( _UDWORD mode )
{
    /* VDC4 Interrupt Level
        b11:b8  ----0101 -------- ; VI_VSYNC, LO_VSYNC, VSYNCERR
        b7:b4   -------- 0101---- ; VLINE, VFIELD
        b3:b0   -------- ----0101 ; VBUFERR1, 2, 3, 4 */
    INTC.IPR10.WORD &= ~0x0FFFu ;
    INTC.IPR10.WORD |= 0x0555u ;

    /* standby control register 7 (STBCR7)
        b3      ----0--- ; MSTP73 : 0 : VDC4 enable */
    CPG.STBCR7.BYTE &= ~0x0008u ;

    /* port G control register 0
        b14:b12 -010---- -------- ; PG3MD[2:0] : 010 : LCD_DATA3
        b10:b8  -----010 -------- ; PG2MD[2:0] : 010 : LCD_DATA2
        b6:b4   -------- -010---- ; PG1MD[2:0] : 010 : LCD_DATA1
        b2:b0   -------- -----010 ; PG0MD[2:0] : 010 : LCD_DATA0 */
    PORT.PGCR0.WORD = 0x2222u ;
    /* port G control register 1
        b14:b12 -010---- -------- ; PG7MD[2:0] : 010 : LCD_DATA7
        b10:b8  -----010 -------- ; PG6MD[2:0] : 010 : LCD_DATA6
        b6:b4   -------- -010---- ; PG5MD[2:0] : 010 : LCD_DATA5
        b2:b0   -------- -----010 ; PG4MD[2:0] : 010 : LCD_DATA4 */
    PORT.PGCR1.WORD = 0x2222u ;
    /* port G control register 2
        b14:b12 -010---- -------- ; PG11MD[2:0] : 010 : LCD_DATA11
        b10:b8  -----010 -------- ; PG10MD[2:0] : 010 : LCD_DATA10
        b6:b4   -------- -010---- ; PG9MD[2:0]  : 010 : LCD_DATA9
        b2:b0   -------- -----010 ; PG8MD[2:0]  : 010 : LCD_DATA8 */
    PORT.PGCR2.WORD = 0x2222u ;
    /* port G control register 3
        b13:b12 --10---- -------- ; PG15MD[1:0] : 10 : LCD_DATA15
        b9:b8   ------10 -------- ; PG14MD[1:0] : 10 : LCD_DATA14
        b5:b4   -------- --10---- ; PG13MD[1:0] : 10 : LCD_DATA13
        b1:b0   -------- ------10 ; PG12MD[1:0] : 10 : LCD_DATA12 */
    PORT.PGCR3.WORD = 0x2222u ;
    /* port G control register 4
        b14:b12 -010---- -------- ; PG19MD[2:0] : 010 : LCD_DATA19
        b10:b8  -----010 -------- ; PG18MD[2:0] : 010 : LCD_DATA18
        b5:b4   -------- --10---- ; PG17MD[1:0] : 10 : LCD_DATA17
        b1:b0   -------- ------10 ; PG16MD[1:0] : 10 : LCD_DATA16 */
```

```
49        PORT.PGCR4.WORD = 0x2222u ;
50        /* port G control register 5
51           b14:b12 -010---- -------- ; PG23MD[2:0] : 010 : LCD_DATA23
52           b10:b8  -----010 -------- ; PG22MD[2:0] : 010 : LCD_DATA22
53           b6:b4   -------- -010---- ; PG21MD[2:0] : 010 : LCD_DATA21
54           b2:b0   -------- -----010 ; PG20MD[2:0] : 010 : LCD_DATA20 */
55        PORT.PGCR5.WORD = 0x2222u ;
56        /* port G control register 6
57           b13:b12 --11---- -------- ; PG27MD[1:0] : 11 : LCD_EXTCLK
58           b9:b8   ------10 -------- ; PG26MD[1:0] : 10 LCD_TCON1
59           b5:b4   -------- --10---- ; PG25MD[1:0] : 10 LCD_TCON0
60           b1:b0   -------- ------10 ; PG24MD[1:0] : 10 LCD_CLK */
61        PORT.PGCR6.WORD = 0x3222u ;
62        /* port J control register 5
63           b14:b12 -011---- -------- ; PJ23MD[2:0] : 011 : LCD_TCON6
64           b10:b8  -----011 -------- ; PJ22MD[2:0] : 011 : LCD_TCON5
65        */
66        PORT.PJCR5.WORD &= ~0x7700u ;
67        PORT.PJCR5.WORD |= 0x3300u ;
68
69        /* port J control register 7
70           b12    ---1---- -------- ; PJ31MD : 1 : DV_CLK
71        */
72        PORT.PJCR7.WORD |= 0x1000u ;
73        /* port E control register 1
74           b5:b4   -------- --11---- ; PE5MD[1:0] : 11 : DV_HSYNC
75           b1:b0   -------- ------11 ; PE4MD[1:0] : 11 : DV_VSYNC */
76        PORT.PECR1.WORD |= 0x0033u ;
77        /* port J control register 0
78           b14:b12 -001---- -------- ; PJ3MD[2:0] : 001 : DV_DATA3
79           b10:b8  -----001 -------- ; PJ2MD[2:0] : 001 : DV_DATA2
80           b6:b4   -------- -001---- ; PJ1MD[2:0] : 001 : DV_DATA1
81           b2:b0   -------- -----001 ; PJ0MD[2:0] : 001 : DV_DATA0
82        */
83        PORT.PJCR0.WORD &= ~0x7777u ;
84        PORT.PJCR0.WORD |= 0x1111u ;
85        /* port J control register 1
86           b14:b12 -001---- -------- ; PJ7MD[2:0] : 001 : DV_DATA7
87           b10:b8  -----001 -------- ; PJ6MD[2:0] : 001 : DV_DATA6
88           b6:b4   -------- -001---- ; PJ5MD[2:0] : 001 : DV_DATA5
89           b2:b0   -------- -----001 ; PJ4MD[2:0] : 001 : DV_DATA4
90        */
91        PORT.PJCR1.WORD &= ~0x7777u ;
92        PORT.PJCR1.WORD |= 0x1111u ;
93    }
```

3.2 VDC4_Terminateのユーザ定義関数例

```
1     /***********************************************************************//**
2      * Function Name : Quit_VDC4_CallBack
3      * @brief
4      * @param     [in]_UDWORD mode
5      * @retval    void
6      ***********************************************************************/
7     void Quit_VDC4_CallBack( _UDWORD mode )
8     {
9        /* VDC4 Interrupt Level
10          b11:b8  ----0000 -------- ; VI_VSYNC, LO_VSYNC, VSYNCERR
11          b7:b4   -------- 0000---- ; VLINE, VFIELD
12          b3:b0   -------- ----0000 ; VBUFERR1, 2, 3, 4 */
13       INTC.IPR10.WORD &= ~0x0FFFu ;
14
15       /* standby control register 7 (STBCR7)
16          b3            ----1--- ; MSTP73 : VDC4 disable */
17       CPG.STBCR7.BYTE |= 0x0008u ;
18    }
```

3.3 VDC4_CheckClockのユーザ定義関数例

関数「VDC4_CheckClock」を使用する際には、引数で指定された時間だけ処理を遅延させるwait処理をユーザが実装する必要があります。以下に、２つの実装例を示します。

3.3.1 コンペアマッチタイマを利用したwait処理

```
1     /**********************************************************************//**
2      * Function Name : Wait_Msec
3      * @brief      Compare Match Timer (1ch) to Wait a msec
4      * @n                Clock select
5      * @n                Peripheral 0 clock 33.33MHz / 512
6      * @param      [in]_UDWORD wait    : Wait time in msec
7      * @retval     void
8      **********************************************************************/
9     void Wait_Msec( _UDWORD wait )
10    {
11       UDWORD Compare ;
12       UWORD Reg ;
13       UDWORD PushReg ;
14       volatile _UBYTE DummyRead ;
15
16       PushReg = ( _UDWORD )CPG.STBCR7.BYTE & 0x0004ul ;
17       /* Standby control register 7 (STBCR7)
18          b3     -----0-- ; MSTP72 : 0 : Compare match timer enable */
19       CPG.STBCR7.BYTE &= ( _UBYTE )( ~0x0004u ) ;
20       /* Dummy read */
21       DummyRead = CPG.STBCR7.BYTE ;
22
23       /* Compare Match Timer Control/Status Register
24          b7     ---- ---- 0--- ---- CMF: Compare Match Flag; clearing
25          b6     ---- ---- -0-- ---- CMIE: Compare Match Interrupt Enable; disabled
26          b1:b0   ---- ---- ---- --11 CKS: Clock Select, P0/512
27        */
28       CMT.CMCSR1.WORD = ( _UWORD )0x0003u ;
29
30       /* 1000usec / 15.36usec = 65.1 */
31       Compare = wait * 66ul ;
32       if( Compare >= 0x00010000ul )
33       {
34          Compare = 0x0000FFFFul ;
35       }
36       /* Compare Match Constant Register */
37       CMT.CMCOR1.WORD = ( _UWORD )Compare ;
38       /* Compare Match Counter */
39       CMT.CMCNT1.WORD = 0 ;
40       /* Compare Match Timer Start Register
41          b1     ---- ---- ---- --1- STR1: Count Start 1; started
42        */
43       CMT.CMSTR.WORD |= ( _UWORD )0x0002u ;
44
45       do
46       {
47          Reg = ( _UWORD )( CMT.CMCSR1.WORD & 0x0080u ) ;
48       }
49       while( Reg == 0 ) ;
50       /* Compare Match Timer Start Register
```

```
51          b1     ---- ---- ---- --0- STR1: Count Start 1; stopped
52         */
53        CMT.CMSTR.WORD &= ( _UWORD )~0x0002u ;
54
55        /* Compare Match Timer Control/Status Register
56          b7     ---- ---- 0--- ---- CMF: Compare Match Flag; clearing
57         */
58        CMT.CMCSR1.WORD &= ( _UWORD )~0x0080u ;
59
60        if( PushReg != 0 )
61        {   /* Standby control register 7 (STBCR7)
62               b3     -----1-- ; MSTP72 : 1 : Compare match timer disable */
63           CPG.STBCR7.BYTE |= ( _UBYTE )0x0004u ;
64        }
65    }
```

この実装例では、コンペアマッチタイマの動作クロックとして、33.33MHz の周辺クロック 0(P0φ)を使用しています。

3.3.2 ビジーウェイトによるwait処理

```
1     /**********************************************************************//**
2      * Function Name : Wait_Msec
3      * @brief      Delya loop to Wait a msec
4      * @param      [in]_UDWORD wait    : Wait time in msec
5      * @retval     void
6      ***********************************************************************/
7     void Wait_Msec( _UDWORD wait )
8     {
9     _  UDWORD i, j ;
10       volatile _UDWORD count ;
11
12       for( i = 0 ; i < wait ; i++ )
13       {
14          count = 0 ;
15          /* The value '53368' is a number to wait a msec.
16             This value is depends on the environment (i.e. memory mapping,
17             memory caching, optimization, and etc.). */
18          for( j = 0 ; j < 53368ul ; j++ )
19          {
20             count++ ;
21          }
22       }
23    }
```

ビジーウェイトによる wait は、メモリマッピングやキャッシング、最適化の設定といった環境や、割り込みやタスク(スレッド)のディスパッチといった処理によって、ループ回数が変わってしまいます。OS を利用する場合には、OS で実装されている遅延用の関数を利用する事を推奨します。

4. 使用例

4.1 ダブルバッファの実装

本ドライバではグラフィックスのダブルバッファに関して 2 つの方法を提供します。

4.1.1 VDC4 のダブルバッファ

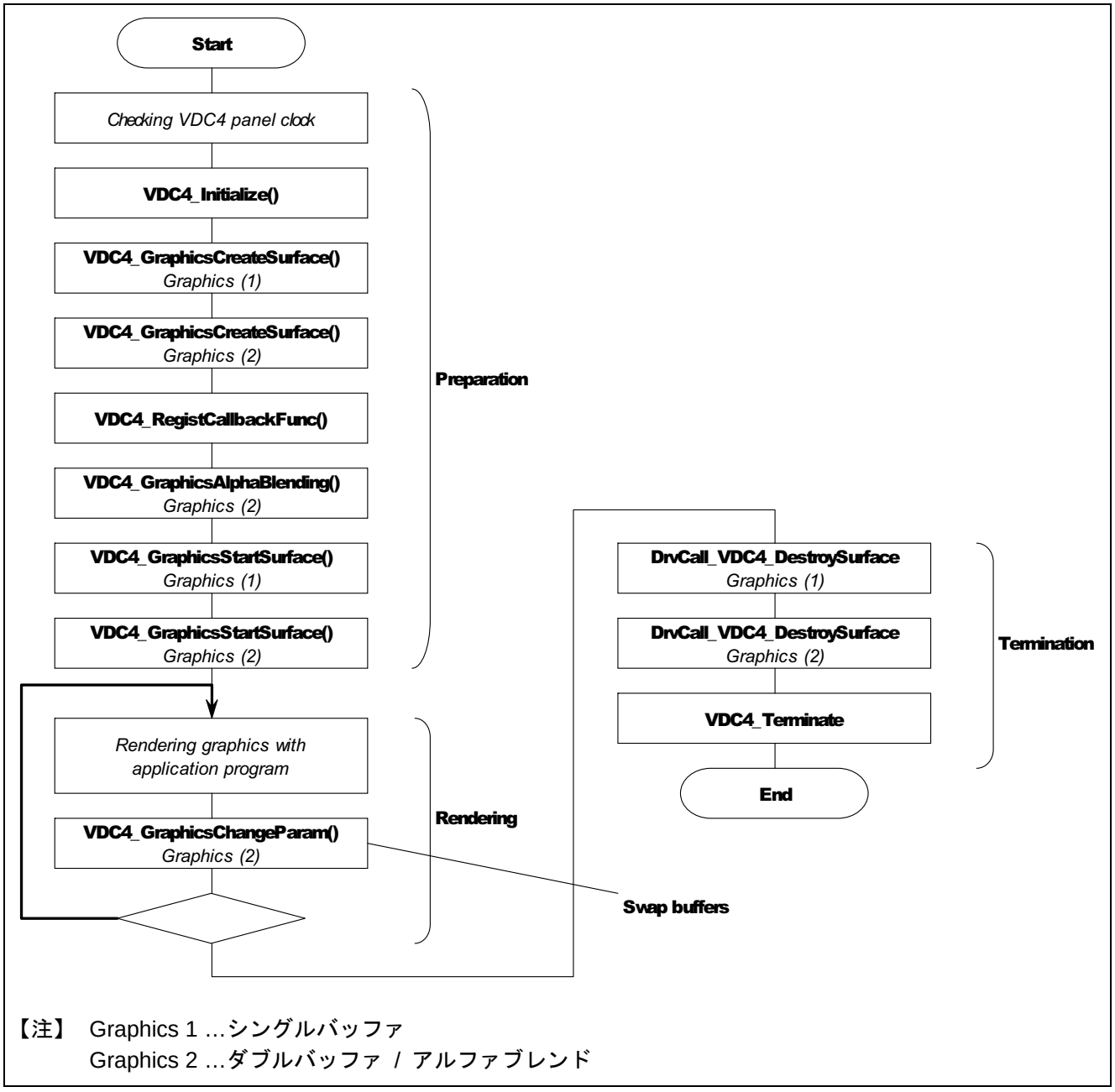
第 2 フレームバッファのアドレスを指定すると、VDC4 ドライバがダブルバッファリングを行います。表示開始時やバッファ切り替え時には、表示フレームをフレーム番号で指定します。

4.1.2 ユーザのダブルバッファ

フレームバッファのベースアドレスのみを指定します。この場合、VDC4 ドライバはシングルバッファの動作を行います。ユーザは、VDC4 ドライバの設定変更関数「VDC4_GraphicsChangeParam」を利用して切り替えるバッファのアドレスを指定します。

4.2 処理フロー例

4.2.1 グラフィックス(1)とグラフィックス(2)



【注】 Graphics 1 …シングルバッファ
Graphics 2 …ダブルバッファ / アルファブレンド

Figure 14 処理フロー例(グラフックス(1)とグラフィックス(2))

Figure 14において、最初に行われている処理であるVDC4 のパネルクロック確認処理については「4.2.2パネルクロックの異常検出とリトライ処理」を参照ください。

4.2.2 パネルクロックの異常検出とリトライ処理

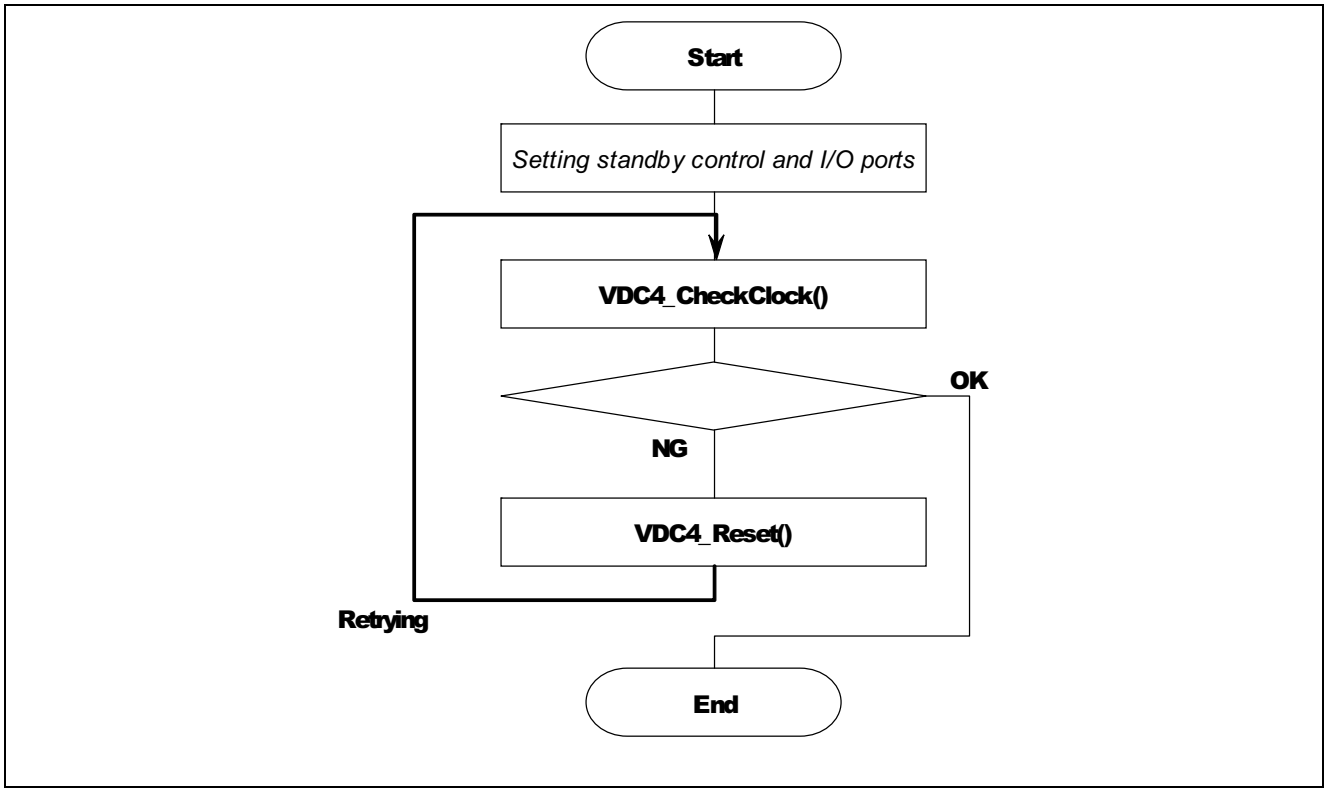


Figure 15 パネルクロック異常検出の処理フロー例

最初に、VDC4 モジュールのクロック供給設定や、I/Oポートの設定を行います。その後、関数「VDC4_CheckClock」によりパネルクロックのチェックを行います。関数「VDC4_CheckClock」の戻り値が"VDC4_ERR_NONE"だった場合、パネルクロックは正常動作しているので終了です。戻り値が"VDC4_ERR_SYSTEM_PNLCLK"だった場合は、パネルクロックの異常が検出されている為、リトライ処理を行います。リトライ処理は、関数「VDC4_Reset」によりVDC４モジュールを一旦リセットした後、関数「VDC4_CheckClock」により再度パネルクロックのチェックを行います。

4.3 LCDの設定例

VDC4 ドライバを利用したLCD設定処理は、API関数「VDC4_Initialize」によって行われます。以下に、LCD の仕様と関数「VDC4_Initialize」の設定値の関係を説明します。

4.3.1 水平同期信号と垂直同期信号を使用するLCDの設定例

水平同期信号(Hsync)と垂直同期信号(Vsync)を使用する LCD の設定について示します。例として、以下のような設定で動作させる LCD について説明します。

Table 17 LCD を制御する VDC4 の設定

LCD パネル駆動用信号出力	Hsync : LCD_TCON1 出力* Vsync : LCD_TCON0 出力*
使用する LCD TCON 信号	STVA/VS : 垂直同期信号(Vsync)として使用 STH/SP/HS : 水平同期信号(Hsync)として使用
パネルクロック供給源	周辺クロック 1(P1φ=66.67MHz)を使用
垂直同期信号	VDC4 の内部生成による自走用垂直同期信号
LCD 信号のビット割り付け	R[7:3]-G[7:2]-B[7:3]

【注】 * LCD_TCON1、LCD_TCON0 の出力ピンを使用するためには、VDC4 以外に I/O ポートの設定が必要です。

Table 18 使用する LCD の基本仕様

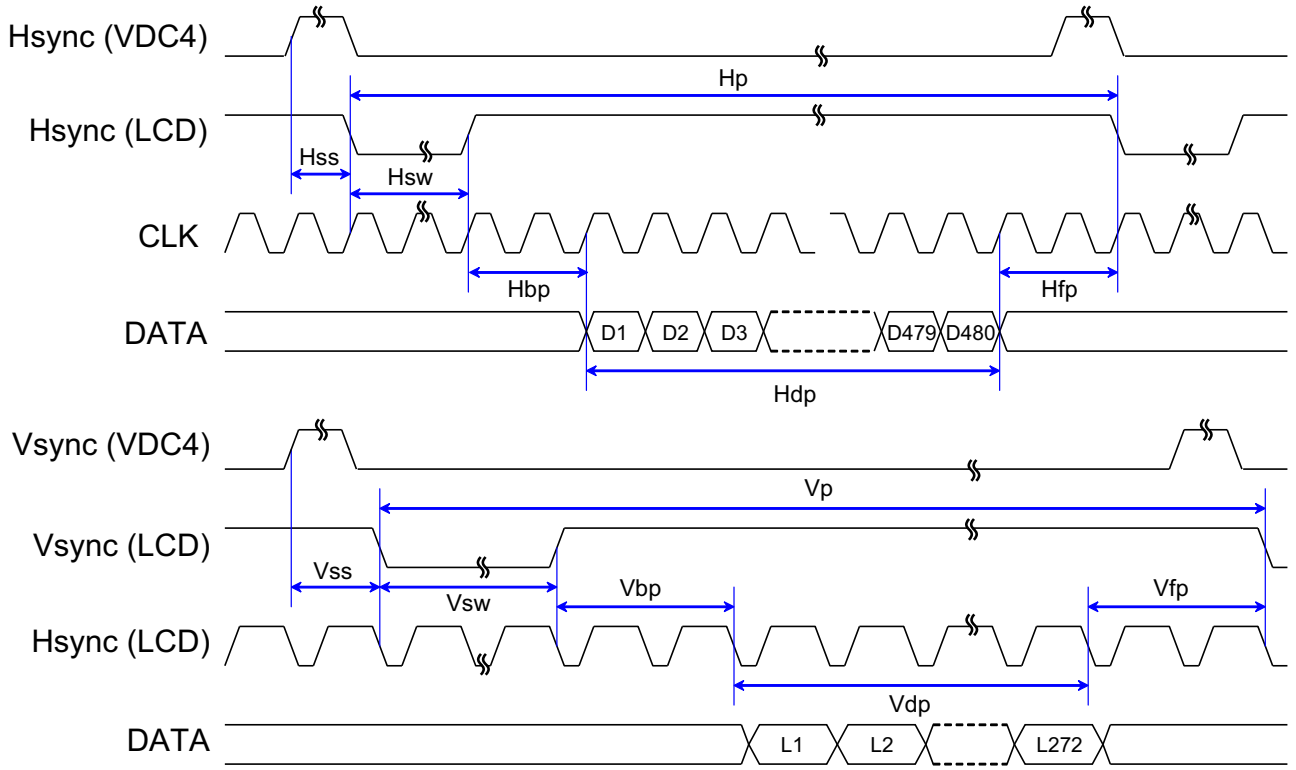
表示サイズ	480x272
表示形式	RGB565
データサンプリング	クロックの立ち下がりエッジ
同期信号の極性	Hsync : 負極性 Vsync : 負極性
パネルクロック周波数	8.0 ~ 9.0 MHz

Table 19 使用する LCD の信号制御タイミング

信号名	記号	設定値
水平同期信号周期	Hp	525 [clock]
水平表示幅	Hdp	480 [clock]
水平同期信号パルス出力開始位置*	Hss	0 [clock]
水平同期信号パルス出力幅	Hsw	41 [clock]
水平バックポーチ	Hbp	2 [clock]
水平フロントポーチ	Hfp	2 [clock]
垂直同期信号周期	Vp	286 [line]
垂直表示幅	Vdp	272 [line]
垂直同期信号パルス出力開始位置*	Vss	0 [line]
垂直同期信号パルス出力幅	Vsw	10 [line]
垂直バックポーチ	Vbp	2 [line]
垂直フロントポーチ	Vfp	2 [line]

【注】 * Hss と Vss 以外の設定値は LCD の仕様に従った値です。Hss と Vss の設定は VDC4 内部の同期信号に対する LCD の同期信号の信号制御タイミングを定義するものです。

LCDの信号制御タイミングの関係は以下のようになります(Figure 16参照)。



Hsync (VDC4) : VDC4 内部水平同期信号　　Vsync (VDC4) : VDC4 内部垂直同期信号
Hsync (LCD) : LCD 水平同期信号　　Vsync (LCD) : LCD 垂直同期信号
CLK : パネルクロック (ピクセルクロック)
DATA : データ (D1~D480 画素単位のデータ、L1~L272 : ライン単位のデータ)

Figure 16　Hsync / Vsync 駆動 LCD パネル制御信号のタイミング

このLCDの設定は、VDC4 ドライバの関数「VDC4_Initialize」の呼び出し時に指定する構造体「vdc4_InitAttr」に、以下のように設定されます。

```
1     vdc4_InitAttr InitAttr ;
2     vdc4_LcdTconTim LcdTconTim_STVA_VS ;
3     vdc4_LcdTconTim LcdTconTim_STH_SP_HS ;
4     vdc4_LcdOutput LcdOutput ;
5
6     /* Vsync */
7     LcdTconTim_STVA_VS.tcon_hsvs = ( _UWORD )( 0 * 2 ) ;
8     LcdTconTim_STVA_VS.tcon_hwvw = ( _UWORD )( 10 * 2 ) ;
9     LcdTconTim_STVA_VS.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
10    LcdTconTim_STVA_VS.tcon_hs_sel = ( _UWORD )0 ;
11    LcdTconTim_STVA_VS.tcon_inv = ( _UWORD )1 ;
12    LcdTconTim_STVA_VS.tcon_pin = VDC4_LCD_TCON_PIN_0 ;
13    LcdTconTim_STVA_VS.outcnt_edge = VDC4_EDGE_RISING ;
14    /* Hsync */
15    LcdTconTim_STH_SP_HS.tcon_hsvs = ( _UWORD )0 ;
16    LcdTconTim_STH_SP_HS.tcon_hwvw = ( _UWORD )41 ;
17    LcdTconTim_STH_SP_HS.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
18    LcdTconTim_STH_SP_HS.tcon_hs_sel = ( _UWORD )0 ;
19    LcdTconTim_STH_SP_HS.tcon_inv = ( _UWORD )1 ;
```

```
20    LcdTconTim_STH_SP_HS.tcon_pin = VDC4_LCD_TCON_PIN_1 ;
21    LcdTconTim_STH_SP_HS.outcnt_edge = VDC4_EDGE_RISING ;
22    /* LCD Output Control */
23    LcdOutput.res_f.hs = ( _UWORD )43 ;
24    LcdOutput.res_f.hw = ( _UWORD )480 ;
25    LcdOutput.res_f.vs = ( _UWORD )12 ;
26    LcdOutput.res_f.vw = ( _UWORD )272 ;
27    LcdOutput.out_endian_on = VDC4_OFF ;
28    LcdOutput.out_swap_on = VDC4_OFF ;
29    LcdOutput.out_format = VDC4_LCD_OUTFORMAT_RGB565 ;
30    LcdOutput.out_frq_sel = VDC4_LCD_PARALLEL_CLKFRQ_1 ;
31    LcdOutput.out_dir_sel = ( _UWORD )0 ;
32    LcdOutput.out_phase = VDC4_LCD_SERIAL_CLKPHASE_0 ;
33
34    /* Initialization parameter */
35    InitAttr.Vsel.FreeRunVsync = VDC4_ON ;
36    InitAttr.Vsel.res_fv = ( _UWORD )( 286 - 1 ) ;
37    InitAttr.Vsel.res_fh = ( _UWORD )( 525 - 1 ) ;
38    InitAttr.sync = NULL ;
39    InitAttr.panel_icksel = VDC4_LCDPANEL_CLKSEL_PERI ;
40    InitAttr.panel_dcdr = VDC4_LCDPANEL_CLKDIV_1_8 ;
41    InitAttr.tcon_half = ( _UWORD )262 ;
42    InitAttr.tcon_offset = ( _UWORD )0 ;
43    InitAttr.outcnt_lcd_edge = VDC4_EDGE_RISING ;
44    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STVA_VS ] = &LcdTconTim_STVA_VS ;
45    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STVB_VE ] = NULL ;
46    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STH_SP_HS ] = &LcdTconTim_STH_SP_HS ;
47    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STB_LP_HE ] = NULL ;
48    InitAttr.outctrl[ VDC4_LCD_TCONSIG_CPV_GCK ] = NULL ;
49    InitAttr.outctrl[ VDC4_LCD_TCONSIG_POLA ] = NULL ;
50    InitAttr.outctrl[ VDC4_LCD_TCONSIG_POLB ] = NULL ;
51    InitAttr.outctrl[ VDC4_LCD_TCONSIG_DE ] = NULL ;
52    InitAttr.settings = &LcdOutput ;
53    InitAttr.IntEnable_1 = VDC4_INT_BIT_VLINE ;
54    InitAttr.IntEnable_2 = VDC4_INT_BIT_NONE ;
```

上記サンプルコードについて以下に説明します。

1. Vsync の設定

Vsync は VDC4 の LCD TCON 生成信号 STVA/VS によって生成します。
STVA/VS の設定は、構造体「vdc4_InitAttr」のメンバ"outctrl"配列中の"VDC4_LCD_TCONSIG_STVA_VS"で参照される場所に、「vdc4_LcdTconTim」構造体データのポインタを設定することで行われます(44 行目)。「vdc4_LcdTconTim」構造体のデータには、Vsync のタイミング制御パラメータを設定します(7~13 行目)。

— 構造体「vdc4_LcdTconTim」の設定

- tcon_hsvs：Vsyncのパルス開始位置'Vss'(Table 19参照)を 1/2 水平周期単位で設定(7 行目)。
- tcon_hwvw：Vsyncのパルス幅'Vsw'(Table 19参照)を 1/2 水平周期単位で設定(8 行目)。
- tcon_md：STVA/VS 信号に対する設定では、本パラメータは参照されません。
- tcon_hs_sel：STVA/VS 信号に対する設定では、本パラメータは参照されません。
- tcon_inv：Vsyncの極性が負極性であるため(Table 18参照)、'1'を設定(11 行目)。
- tcon_pin：Vsyncの出力ピンがLCD_TCON0 なので(Table 17参照)、'VDC4_LCD_TCON_PIN_0'を設定(12 行目)。
- outcnt_edge：データ出力に合わせて立ち上がりエッジ'VDC4_EDGE_RISING'を設定(13 行目)。

2. Hsync の設定

Hsync は VDC4 の LCD TCON 生成信号 STH/SP/HS によって生成します。

STH/SP/HS の設定は、構造体「vdc4_InitAttr」のメンバ"outctrl"配列中の "VDC4_LCD_TCONSIG_STH_SP_HS "で参照される場所に、「vdc4_LcdTconTim」構造体データのポインタを設定することで行われます(46 行目)。「vdc4_LcdTconTim」構造体のデータには、Hsync のタイミング制御パラメータを設定します(15~21 行目)。

— 構造体「vdc4_LcdTconTim」の設定

- tcon_hsvs：Hsyncのパルス開始位置'Hss'(Table 19参照)を設定(15 行目)。
- tcon_hwvw：Hsyncのパルス幅'Hsw'(Table 19参照)を設定(16 行目)。
- tcon_md：STH/SP/HS 信号に対する設定では、本パラメータは参照されません。
- tcon_hs_sel：オフセット基準信号のタイミングは利用しないので、'0'を設定(18 行目)。
- tcon_inv：Hsyncの極性が負極性であるため(Table 18参照)、'1'を設定(19 行目)。
- tcon_pin：Hsyncの出力ピンがLCD_TCON1 なので(Table 17参照)、'VDC4_LCD_TCON_PIN_1'を設定(20 行目)。
- outcnt_edge：データ出力に合わせて立ち上がりエッジ'VDC4_EDGE_RISING'を設定(21 行目)。

3. LCD 出力制御の設定

LCD 出力制御の設定は、構造体「vdc4_InitAttr」のメンバ"settings"に、「vdc4_LcdOutput」構造体データのポインタを設定することで行われます(52 行目)。

— 構造体「vdc4_LcdOutput」の設定

- res_f.hs：水平表示開始位置'Hss'+'Hsw'+'Hbp'(Table 19参照)を設定(23 行目)。
- res_f.hw：水平表示幅'Hdp'(Table 19参照)を設定(24 行目)。
- res_f.vs：垂直表示開始位置'Vss'+' Vsw'+' Vbp'(Table 19参照)を設定(25 行目)。
- res_f.vw：垂直表示幅' Vdp'(Table 19参照)を設定(26 行目)。
- out_endian_on：LCD信号のビット割り付けがMSB first(上位ビットが先)なので(Table 17参照)、'VDC4_OFF'を設定(27 行目)。
- out_swap_on：LCD信号のビット割り付けがRGB順の指定なので(Table 17参照)、'VDC4_OFF'を設定(28 行目)。
- out_format：表示形式がRGB565 なので(Table 18参照)、'VDC4_LCD_OUTFORMAT_RGB565'を設定(29 行目)。
- out_frq_sel：LCD の出力形式がシリアル RGB ではない時、本パラメータは参照されません。
- out_dir_sel：LCD の出力形式がシリアル RGB ではない時、本パラメータは参照されません。
- out_phase：LCD の出力形式がシリアル RGB ではない時、本パラメータは参照されません。

4. その他の設定

その他の LCD に関係する設定(35~43 行目)を以下に説明します。

— 構造体「vdc4_InitAttr」の設定

- Vsel.FreeRunVsync：垂直同期信号にVDC4 内部の自走信号を使用するので(Table 17参照)、'VDC4_ON'を設定(35 行目)。
- Vsel.res_fv：VDC4 内部の垂直同期信号をLCDに合わせる為に、垂直同期信号周期'Vp-1'(Table 19参照)を設定(36 行目)。
- Vsel.res_fh：VDC4 内部の水平同期信号をLCDに合わせる為に、水平同期信号周期'Hp-1'(Table 19参照)を設定(37 行目)。
- panel_icksel：パネルクロックの供給源として周辺バスクロックを使用するので(Table 17参照)、'VDC4_LCDPANEL_CLKSEL_PERI'を設定(39 行目)。
- panel_dcdr:パネルクロックの供給源の周辺バスクロックが 66.67MHzなので(Table 17参照)、LCDのパネルクロック(8.0 MHz ~ 9.0 MHz、Table 18参照)を生成する為に、クロック分周比'VDC4_LCDPANEL_CLKDIV_1_8'(66.67 / 8 = 8.33 MHz)を設定(40 行目)。
- tcon_half：1/2 水平周期タイミング'Hp / 2'(Table 19参照)を設定(41 行目)。
- tcon_offset：基準信号の遅延制御は行わないので、'0'を設定(42 行目)。
- outcnt_lcd_edge：LCDのデータサンプリングがクロックの立ち下がりエッジなので(Table 18参照)、データ出力には立ち上がりエッジ'VDC4_EDGE_RISING'を設定(43 行目)。

4.3.2 HsyncとDEを使用するLCDの設定例

水平同期信号(Hsync)とデータイネーブル信号(DE)を使用する LCD の設定について示します。例として、以下のような設定で動作させる LCD について説明します。

Table 20 LCD を制御する VDC4 の設定

LCD パネル駆動用信号出力	Hsync : LCD_TCON1 出力* DE : LCD_TCON5 出力*
使用する LCD TCON 信号	STH/SP/HS : 水平同期信号(Hsync)として使用 DE : データイネーブル信号(DE)として使用 STVB/VE : 垂直イネーブル信号(VE)として使用 STB/LP/HE : 水平イネーブル信号(HE)として使用
パネルクロック供給源	周辺クロック 1(P1φ=66.67MHz)を使用
垂直同期信号	VDC4 の内部生成による自走用垂直同期信号
LCD 信号のビット割り付け	R[7:2]-G[7:2]-B[7:2]

【注】 * LCD_TCON1、LCD_TCON5 の出力ピンを使用するためには、VDC4 以外に I/O ポートの設定が必要です。

Table 21 使用する LCD の基本仕様

表示サイズ	240x320
表示形式	RGB666
データサンプリング	クロックの立ち下がりエッジ
同期信号の極性	Hsync : 負極性 DE : 正極性
パネルクロック周波数	5.0 ~ 6.0 MHz

Table 22 使用する LCD の信号制御タイミング

信号名	記号	設定値
水平同期信号周期	Hp	273 [clock]
水平表示幅	Hdp	240 [clock]
水平同期信号パルス出力開始位置*	Hss	0 [clock]
水平同期信号パルス出力幅	Hsw	5 [clock]
データイネーブル信号開始位置	DEs	22 [clock]
垂直周期	Vp	327 [line]
垂直表示幅	Vdp	320 [line]

【注】 * Hss 以外の設定値は LCD の仕様に従った値です。Hss の設定は VDC4 内部の同期信号に対する LCD の同期信号の信号制御タイミングを定義するものです。

DE信号はVDC4 内部の水平イネーブル信号(HE)と垂直イネーブル信号(VE)の合成(論理積)により生成されます。DE信号を生成する為のHE信号とVE信号のタイミングを Table 23に示します。

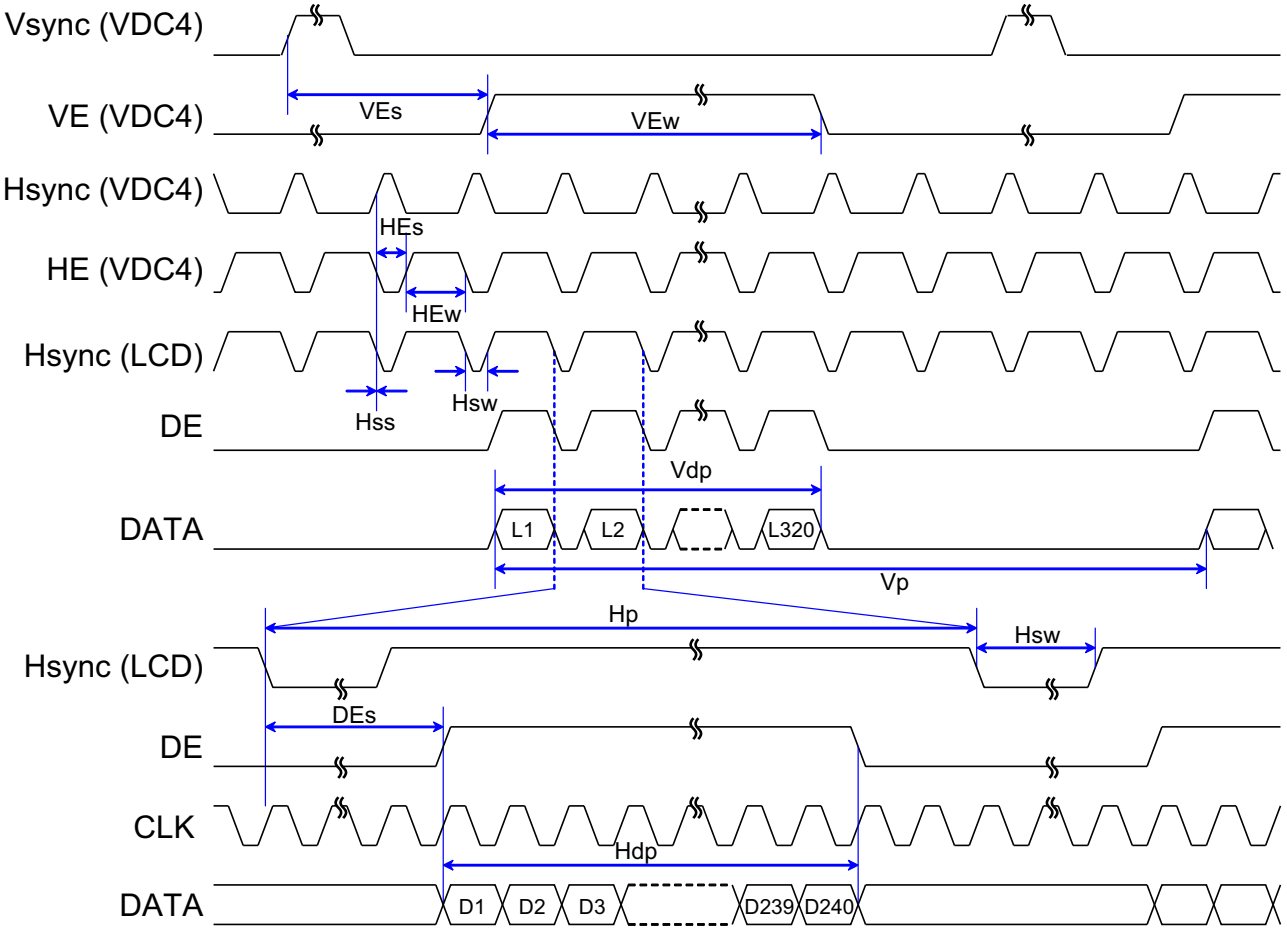
Table 23のVEsの設定値は垂直表示開始位置の設定でも参照されます。VDC4 では、垂直表示開始位置の設定は 4[line]以上を設定しなければならない制限事項がある為、VEsの設定値は 4～垂直ブランキング期間の間の値とする必要があります。

Table 23　DE 信号生成用の HE/VE 信号制御タイミング

信号名	記号	設定値
垂直イネーブル信号パルス出力開始位置	VEs	7 [line]* (= Vp - Vdp)
垂直イネーブル信号パルス出力幅	VEw	320 [line] (= Vdp)
水平イネーブル信号出力開始位置	HEs	22 [clock] (= Hss + DEs)
水平イネーブル信号パルス出力幅	HEw	240 [clock] (= Hdp)

【注】 * VEs の設定値'Vp' - 'Vdp'は垂直ブランキング期間です。

LCDの信号制御タイミングの関係は以下のようになります(Figure 17参照)。



Vsync (VDC4) : VDC4 内部垂直同期信号　　Hsync (VDC4) : VDC4 内部水平同期信号
VE (VDC4) : VDC4 内部垂直イネーブル信号　　HE (VDC4) : VDC4 内部水平イネーブル信号
Hsync (LCD) : LCD 水平同期信号　　DE : データイネーブル信号
CLK : パネルクロック (ピクセルクロック)
DATA : データ (D1~D240 画素単位のデータ、L1~L320 : ライン単位のデータ)

Figure 17　Hsync / DE 駆動 LCD パネル制御信号タイミング

このLCDの設定は、VDC4 ドライバの関数「VDC4_Initialize」の呼び出し時に指定する構造体「vdc4_InitAttr」に、以下のように設定されます。

```
1     vdc4_InitAttr InitAttr ;
2     vdc4_LcdTconTim LcdTconTim_STH_SP_HS ;
3     vdc4_LcdTconTim LcdTconTim_STVB_VE ;
4     vdc4_LcdTconTim LcdTconTim_STB_LP_HE ;
5     vdc4_LcdTconTim LcdTconTim_DE ;
6     vdc4_LcdOutput LcdOutput ;
7
8     /* VE */
9     LcdTconTim_STVB_VE.tcon_hsvs = ( _UWORD )( 7 * 2 ) ;
10    LcdTconTim_STVB_VE.tcon_hwvw = ( _UWORD )( 320 * 2 ) ;
11    LcdTconTim_STVB_VE.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
12    LcdTconTim_STVB_VE.tcon_hs_sel = ( _UWORD )0 ;
13    LcdTconTim_STVB_VE.tcon_inv = ( _UWORD )0 ;
14    LcdTconTim_STVB_VE.tcon_pin = VDC4_LCD_TCON_PIN_NON ;
15    LcdTconTim_STVB_VE.outcnt_edge = VDC4_EDGE_RISING ;
16    /* HE */
17    LcdTconTim_STB_LP_HE.tcon_hsvs = ( _UWORD )22 ;
18    LcdTconTim_STB_LP_HE.tcon_hwvw = ( _UWORD )240 ;
19    LcdTconTim_STB_LP_HE.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
20    LcdTconTim_STB_LP_HE.tcon_hs_sel = ( _UWORD )0 ;
21    LcdTconTim_STB_LP_HE.tcon_inv = ( _UWORD )0 ;
22    LcdTconTim_STB_LP_HE.tcon_pin = VDC4_LCD_TCON_PIN_NON ;
23    LcdTconTim_STB_LP_HE.outcnt_edge = VDC4_EDGE_RISING ;
24    /* DE */
25    LcdTconTim_DE.tcon_hsvs = ( _UWORD )0 ;
26    LcdTconTim_DE.tcon_hwvw = ( _UWORD )0 ;
27    LcdTconTim_DE.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
28    LcdTconTim_DE.tcon_hs_sel = ( _UWORD )0 ;
29    LcdTconTim_DE.tcon_inv = ( _UWORD )0 ;
30    LcdTconTim_DE.tcon_pin = VDC4_LCD_TCON_PIN_5 ;
31    LcdTconTim_DE.outcnt_edge = VDC4_EDGE_RISING ;
32    /* Hsync */
33    LcdTconTim_STH_SP_HS.tcon_hsvs = ( _UWORD )0 ;
34    LcdTconTim_STH_SP_HS.tcon_hwvw = ( _UWORD )5 ;
35    LcdTconTim_STH_SP_HS.tcon_md = VDC4_LCD_TCON_POLMD_NORMAL ;
36    LcdTconTim_STH_SP_HS.tcon_hs_sel = ( _UWORD )0 ;
37    LcdTconTim_STH_SP_HS.tcon_inv = ( _UWORD )1 ;
38    LcdTconTim_STH_SP_HS.tcon_pin = VDC4_LCD_TCON_PIN_1 ;
39    LcdTconTim_STH_SP_HS.outcnt_edge = VDC4_EDGE_RISING ;
40    /* LCD Output Control */
41    LcdOutput.res_f.hs = ( _UWORD )22 ;
42    LcdOutput.res_f.hw = ( _UWORD )240 ;
43    LcdOutput.res_f.vs = ( _UWORD )7 ;
44    LcdOutput.res_f.vw = ( _UWORD )320 ;
45    LcdOutput.out_endian_on = VDC4_OFF ;
46    LcdOutput.out_swap_on = VDC4_OFF ;
47    LcdOutput.out_format = VDC4_LCD_OUTFORMAT_RGB666 ;
48    LcdOutput.out_frq_sel = VDC4_LCD_PARALLEL_CLKFRQ_1 ;
49    LcdOutput.out_dir_sel = ( _UWORD )0 ;
50    LcdOutput.out_phase = VDC4_LCD_SERIAL_CLKPHASE_0 ;
51
52    /* Initialization parameter */
53    InitAttr.Vsel.FreeRunVsync = VDC4_ON ;
54    InitAttr.Vsel.res_fv = ( _UWORD )( 327 - 1 ) ;
```

```
55    InitAttr.Vsel.res_fh = ( _UWORD )( 273 - 1 ) ;
56    InitAttr.sync = NULL ;
57    InitAttr.panel_icksel = VDC4_LCDPANEL_CLKSEL_PERI ;
58    InitAttr.panel_dcdr = VDC4_LCDPANEL_CLKDIV_1_12 ;
59    InitAttr.tcon_half = ( _UWORD )136 ;
60    InitAttr.tcon_offset = ( _UWORD )0 ;
61    InitAttr.outcnt_lcd_edge = VDC4_EDGE_RISING ;
62    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STVA_VS ] = NULL ;
63    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STVB_VE ] = &LcdTconTim_STVB_VE ;
64    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STH_SP_HS ] = &LcdTconTim_STH_SP_HS ;
65    InitAttr.outctrl[ VDC4_LCD_TCONSIG_STB_LP_HE ] = &LcdTconTim_STB_LP_HE ;
66    InitAttr.outctrl[ VDC4_LCD_TCONSIG_CPV_GCK ] = NULL ;
67    InitAttr.outctrl[ VDC4_LCD_TCONSIG_POLA ] = NULL ;
68    InitAttr.outctrl[ VDC4_LCD_TCONSIG_POLB ] = NULL ;
69    InitAttr.outctrl[ VDC4_LCD_TCONSIG_DE ] = &LcdTconTim_DE ;
70    InitAttr.settings = &LcdOutput ;
71    InitAttr.IntEnable_1 = VDC4_INT_BIT_VLINE ;
72    InitAttr.IntEnable_2 = VDC4_INT_BIT_NONE ;
```

上記サンプルコードについて以下に説明します。

1. VE の設定

VE は VDC4 の LCD TCON 生成信号 STVB/VE によって生成します。
STVB/VE の設定は、構造体「vdc4_InitAttr」のメンバ"outctrl"配列中の"VDC4_LCD_TCONSIG_STVB_VE"で参照される場所に、「vdc4_LcdTconTim」構造体データのポインタを設定することで行われます(63 行目)。「vdc4_LcdTconTim」構造体のデータには、VE のタイミング制御パラメータを設定します(9~15 行目)。

— 構造体「vdc4_LcdTconTim」の設定

- tcon_hsvs：VEのパルス開始位置'VEs'(Table 23参照)を 1/2 水平周期単位で設定(9 行目)。
- tcon_hwvw：VEのパルス幅'VEw'(Table 23参照)を 1/2 水平周期単位で設定(10 行目)。
- tcon_md：STVB/VE 信号に対する設定では、本パラメータは参照されません。
- tcon_hs_sel：STVB/VE 信号に対する設定では、本パラメータは参照されません。
- tcon_inv：VE は外部出力しないので本設定は動作に影響しない。'0'を設定(13 行目)。
- tcon_pin：VE は外部出力しないので、'VDC4_LCD_TCON_PIN_NON'を設定(14 行目)。
- outcnt_edge：VE は外部出力しないので、本パラメータは参照されません。

2. HE の設定

HE は VDC4 の LCD TCON 生成信号 STB/LP/HE によって生成します。
STB/LP/HE の設定は、構造体「vdc4_InitAttr」のメンバ"outctrl"配列中の"VDC4_LCD_TCONSIG_STB_LP_HE"で参照される場所に、「vdc4_LcdTconTim」構造体データのポインタを設定することで行われます(65 行目)。「vdc4_LcdTconTim」構造体のデータには、HE のタイミング制御パラメータを設定します(17~23 行目)。

— 構造体「vdc4_LcdTconTim」の設定

- tcon_hsvs：HEのパルス開始位置'HEs'(Table 23参照)を設定(17 行目)。
- tcon_hwvw：HEのパルス幅'HEw'(Table 23参照)を設定(18 行目)。
- tcon_md：STB/LP/HE 信号に対する設定では、本パラメータは参照されません。
- tcon_hs_sel： オフセット基準信号のタイミングは利用しないので、'0'を設定(20 行目)。
- tcon_inv：HE は外部出力しないので本設定は動作に影響しない。'0'を設定(21 行目)。
- tcon_pin：HE は外部出力しないので、'VDC4_LCD_TCON_PIN_NON'を設定(22 行目)。
- outcnt_edge：HE は外部出力しないので、本パラメータは参照されません。

3. DE の設定

DE は VDC4 の LCD TCON 生成信号 DE によって生成します。

DE の設定は、構造体「vdc4_InitAttr」のメンバ"outctrl"配列中の"VDC4_LCD_TCONSIG_DE"で参照される場所に、「vdc4_LcdTconTim」構造体データのポインタを設定することで行われます(69 行目)。「vdc4_LcdTconTim」構造体のデータには、DE のタイミング制御パラメータを設定します(25~31 行目)。

— 構造体「vdc4_LcdTconTim」の設定

- tcon_hsvs：DE 信号に対する設定では、本パラメータは参照されません。
- tcon_hwvw：DE 信号に対する設定では、本パラメータは参照されません。
- tcon_md：DE 信号に対する設定では、本パラメータは参照されません。
- tcon_hs_sel：DE 信号に対する設定では、本パラメータは参照されません。
- tcon_inv：DEの極性が正極性であるため(Table 21参照)、'0'を設定(29 行目)。
- tcon_pin：DEの出力ピンがLCD_TCON5 なので(Table 20参照)、'VDC4_LCD_TCON_PIN_5'を設定(30 行目)。
- outcnt_edge：データ出力に合わせて立ち上がりエッジ'VDC4_EDGE_RISING'を設定(31 行目)。

4. Hsync の設定

Hsync は VDC4 の LCD TCON 生成信号 STH/SP/HS によって生成します。

STH/SP/HS の設定は、構造体「vdc4_InitAttr」のメンバ"outctrl"配列中の"VDC4_LCD_TCONSIG_STH_SP_HS"で参照される場所に、「vdc4_LcdTconTim」構造体データのポインタを設定することで行われます(64 行目)。「vdc4_LcdTconTim」構造体のデータには、Hsync のタイミング制御パラメータを設定します(33~39 行目)。

— 構造体「vdc4_LcdTconTim」の設定

- tcon_hsvs：Hsyncのパルス開始位置'Hss'(Table 22参照)を設定(33 行目)。
- tcon_hwvw：Hsyncのパルス幅'Hsw'(Table 22参照)を設定(34 行目)。
- tcon_md：STH/SP/HS 信号に対する設定では、本パラメータは参照されません。
- tcon_hs_sel：オフセット基準信号のタイミングは利用しないので、'0'を設定(36 行目)。
- tcon_inv：Hsyncの極性が負極性であるため(Table 21参照)、'1'を設定(37 行目)。
- tcon_pin：Hsyncの出力ピンがLCD_TCON1 なので(Table 20参照)、'VDC4_LCD_TCON_PIN_1'を設定(38 行目)。
- outcnt_edge：データ出力に合わせて立ち上がりエッジ'VDC4_EDGE_RISING'を設定(39 行目)。

5. LCD 出力制御の設定

LCD 出力制御の設定は、構造体「vdc4_InitAttr」のメンバ"settings"に、「vdc4_LcdOutput」構造体データのポインタを設定することで行われます(70 行目)。

— 構造体「vdc4_LcdOutput」の設定

- res_f.hs：水平表示開始位置'HEs'(Table 23参照)を設定(41 行目)。
- res_f.hw：水平表示幅'Hdp'(Table 22参照)を設定(42 行目)。
- res_f.vs：垂直表示開始位置'VEs'(Table 23参照)を設定(43 行目)。
- res_f.vw：垂直表示幅' Vdp'(Table 22参照)を設定(44 行目)。
- out_endian_on：LCD信号のビット割り付けがMSB first(上位ビットが先)なので(Table 20参照)、'VDC4_OFF'を設定(45 行目)。
- out_swap_on：LCD信号のビット割り付けがRGB順の指定なので(Table 20参照)、'VDC4_OFF'を設定(46 行目)。
- out_format：表示形式がRGB666 なので(Table 21参照)、'VDC4_LCD_OUTFORMAT_RGB666'を設定(47 行目)。
- out_frq_sel：LCD の出力形式がシリアル RGB ではない時、本パラメータは参照されません。
- out_dir_sel：LCD の出力形式がシリアル RGB ではない時、本パラメータは参照されません。
- out_phase：LCD の出力形式がシリアル RGB ではない時、本パラメータは参照されません。

6. その他の設定

その他の LCD に関係する設定(53~61 行目)を以下に説明します。

— 構造体「vdc4_InitAttr」の設定

- Vsel.FreeRunVsync：垂直同期信号にVDC4 内部の自走信号を使用するので(Table 20参照)、'VDC4_ON'を設定(53 行目)。
- Vsel.res_fv：VDC4 内部の垂直同期信号をLCDに合わせる為に、垂直同期信号周期'Vp-1'(Table 22参照)を設定(54 行目)。

- `Vsel.res_fh` : VDC4 内部の水平同期信号をLCDに合わせる為に、水平同期信号周期'Hp-1'(Table 22参照)を設定(55 行目)。
- `panel_icksel` : パネルクロックの供給源として周辺バスクロックを使用するので(Table 20参照)、'`VDC4_LCDPANEL_CLKSEL_PERI`'を設定(57 行目)。
- `panel_dcdr`:パネルクロックの供給源の周辺バスクロックが 66.67MHzなので(Table 20参照)、LCDのパネルクロック(5.0 MHz ~ 6.0 MHz、Table 21参照)を生成する為に、クロック分周比'`VDC4_LCDPANEL_CLKDIV_1_12`'(66.67 / 12 = 5.56 MHz)を設定(58 行目)。
- `tcon_half` : 1/2 水平周期タイミング'Hp / 2'(Table 22参照)を設定(59 行目)。
- `tcon_offset` : 基準信号の遅延制御は行わないので、'0'を設定(60 行目)。
- `outcnt_lcd_edge` : LCDのデータサンプリングがクロックの立ち下がりエッジなので(Table 21参照)、データ出力には立ち上がりエッジ'`VDC4_EDGE_RISING`'を設定(61 行目)。

ホームページとサポート窓口

- ルネサス エレクトロニクスホームページ
  http://japan.renesas.com/
- お問合せ先
  http://japan.renesas.com/inquiry

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2013. 04.18	—	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

   【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

   CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

   【注意】電源投入時は，製品の状態は不定です。

   電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

   外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

   同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

   【注意】リザーブアドレスのアクセスを禁止します。

   アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

   【注意】リセット時は、クロックが安定した後、リセットを解除してください。

   プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

   リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

   【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

   同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、
   各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
   家電、工作機械、パーソナル機器、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通用信号機器、
   防災・防犯装置、各種安全装置等
   当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。
注2. 本資料において使用されている「当社製品」とは、注１において定義された当社の開発、製造製品をいいます。

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口 http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社　〒100-0004 千代田区大手町 2-6-2（日本ビル）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：http://japan.renesas.com/contact/

Colophon 2.1