To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

SH7764 Group

Video Display Controller TFT-LCD Interfacing Example 2

Introduction

This application note shows the TFT-LCD interfacing example using the SH7764 Microcontrollers (MCUs) on-chip Video Display Controller (VDC2). Overlay processing is executed for four layers, and includes α-blending and chroma-keying.

Target Device

SH7764 (R0K507764E001BR from Renesas Technology Corp.)

Contents

1. Preface

1.1 Specifications

The SH7764 MCU on-chip video display controller (VDC2) is connected with a TFT-LCD panel to display the graphic image.

1.2 Module Used

- Video display controller (VDC2)
- General-purpose I/O ports (GPIO)

1.3 Applicable Conditions

- MCU: SH7764
- Operating frequency:
  - CPU clock: 324 MHz
  - SuperHyway clock: 108 MHz
  - Peripheral clock: 54 MHz
  - Bus clock: 108 MHz
- Integrated development environment
  from Renesas Technology Corp.
- C compiler: SuperH RISC Engine Family C/C++ Compiler Package Ver.9.03 Release00
  from Renesas Technology Corp.
- Compiler options: Default settings of the High-performance Embedded Workshop
  -cpu=sh4a -endian=little -include="$(WORKSPDIR)\inc"
  -object="$(CONFIGDIR)\$(FILELEAF).obj" -debug -optimize=0
  -gbr=auto -chgincpath -errorpath
  -global_volatile=0 -opt_range=all -infinite_loop=0
  -del_vacant_loop=0 -struct_alloc=1 –nologo

1.4 Related Application Note

Refer to the related application notes as follows:

- SH7764 Group Application Note: SH7764 Example of Initialization (REJ06B0919)
- SH7764 Group Application Note: Video Display Controller TFT-LCD Interfacing Example 1 (REJ06B0920)

2. Description of the Sample Application

This application note shows the pin connection example and configuration example to display the graphic image by the VDC2. The specifications of the TFT-LCD panel used in this application note are shown in 2.2.

2.1 VDC2 Operation

2.1.1 Overview

The video display controller (VDC2) provides functions for reading four planes of graphic images (layers 1 to 4) stored in the external memory and overlaying them. It outputs 18-bit RGB video (each color is represented by six bits) and digital video data conforming to BTA T-1004.

2.1.2 Features

Table 1 lists the VDC2 features.

Table 1 VDC2 Features

Item	Function
Operating frequency	6.0 MHz to 36.0 MHz (depends on the display panel size)
Input image format	16-bit RGB565 progressive
Display size	18-bit progressive RGB output 720 × 480 (NTSC) 720 × 576 (PAL) 320 × 240 (QVGA) 640 × 480 (VGA) 800 × 480 (WVGA)
Display planes	Up to four planes (layers 1 to 4)
α blending	Mixes layers 1 to 4 according to the transparency
Chroma-keying	Applies chroma-key processing to the specified RGB color
Output video format	RGB666 progressive video output
Sync signal output	Either a combination of Vsync, Hsync, data enable, and COM/CDE signals
External sync mode	The VDC2 can operate with external sync signals (EX-VSYNC and EX-HSYNC) and the panel clock

2.1.3 I/O Pins

Table 2 lists the VDC2 I/O pins.

Table 2 VDC2 I/O Pins

Symbol	I/O	Pin Name	Function
DR[5:0]	Output	Digital red data	Video data output pins
DG[5:0]	Output	Digital green data	Video data output pins
DB[5:0]	Output	Digital blue data	Video data output pins
VSYNC	Output	Vertical sync signal	Vertical sync signal
HSYNC	Output	Horizontal sync signal	Horizontal sync signal
DE_V	Output	Vertical data enable signal	Vertical data enable signal
DE_H/DE_C	Output	Horizontal data enable signal/display enable signal	Horizontal data enable signal/display enable signal
COM/CDE	Output	Gate control signal/chroma data enable signal	Gate control signal/display enable signal (asserted when the data matches the chroma-key color specified in the register).
BT_DATA[7:0]	Output	BTA-T1004 display data	BTA-T1004 display data output pins
BT_VSYNC	Output	BTA-T1004 vertical sync	BTA-T1004 vertical sync signal
BT_DE_C	Output	BTA-T1004 display enable	BTA-T1004 display enable signal
EX_VSYNC	Input	VSYNC input	VSYNC input pin used in external sync mode
EX_HSYNC	Input	HSYNC input	HSYNC input pin used in external sync mode
DCLKIN	Input	Panel source clock input	Display source clock input pin. Input an appropriate frequency depending on the display panel.
DCLKOUT	Output	Panel clock output	Panel clock output pin

2.1.4 VDC2 Configuration

Table 3 lists the each functional block. Figure 1 shows the VDC2 block diagram.

Table 3 VDC2 Functional Blocks

Block Name	Overview of Functions
Graphics block 1	Reads a graphic image (RGB565: layer 1) stored in the external memory through the pixel bus and outputs it to graphics block 2.
Graphics block 2	Reads a graphic image (RGB565: layer 2) stored in the external memory through the pixel bus, overlays it on the output from graphics block 1, and outputs the result to graphics block 3.
Graphics block 3	Reads a graphic image (RGB565: layer 3) stored in the external memory through the pixel bus, overlays it on the output from graphics block 2, and outputs the result to graphics block 4.
Graphics block 4	Reads a graphic image (RGB565: layer 4) stored in the external memory through the pixel bus, overlays it on the output from graphics block 3, and outputs the resultant image data.
Display control block	Converts the output (RGB) from graphics block 4 into the YCbCr (4:2:2) format and outputs the data in the 8:4:4 parallel format conforming to the BTA T-1004 standard. It also outputs the control signals for the TFT-LCD panel.
Input timing control block	Selects the timing of the external sync signal input with respect to the clock rising or falling edge and selects the sync signal polarity.
Output timing control block	Controls the timing of the sync signal output with respect to the clock rising or falling edge and controls the sync signal polarity. It also controls the timing of the RGB666 video output signals with respect to the clock rising or falling edge.

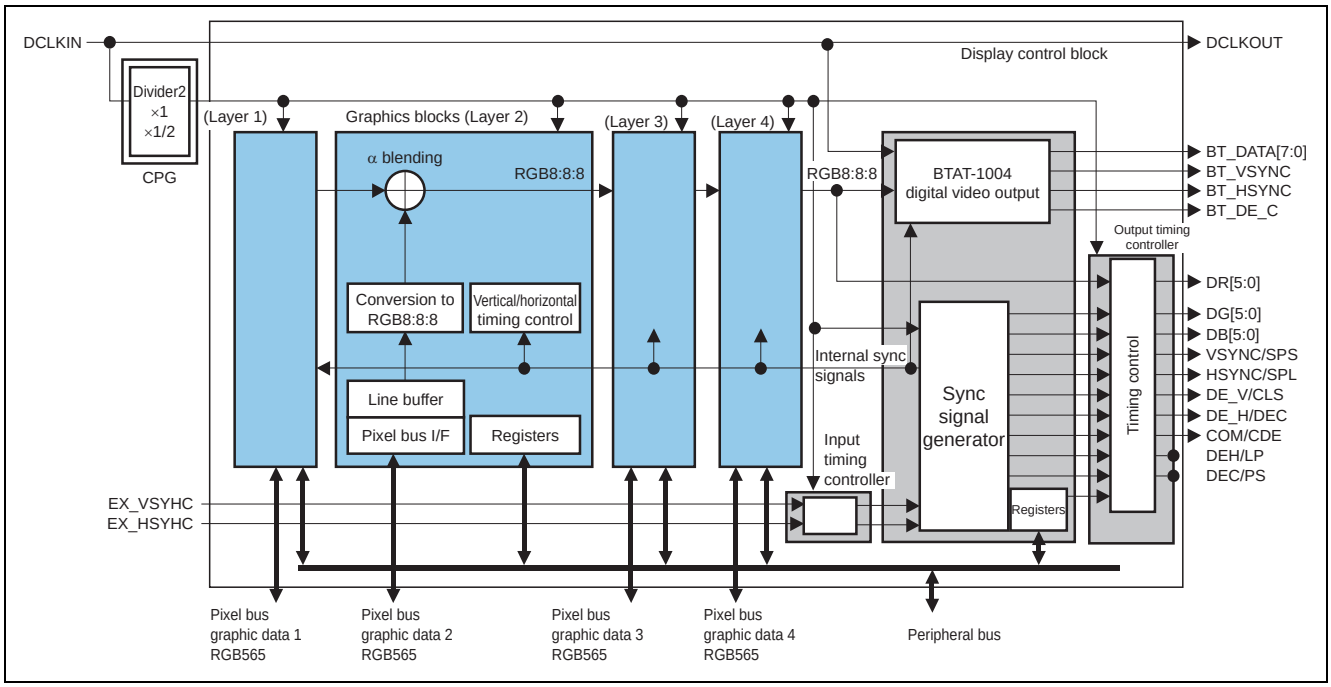


Figure 1 VDC2 Block Diagram

2.1.5 VDC2 Functions

The graphics blocks display in the RGB565 (16-bit) format the image data stored in the memory area. The graphics blocks control display by using the external input sync signals or internally generated sync signals. A single plane of an image can be displayed, and two to four planes of images can also be displayed through overlay processing. In overlaid display, the lower-layer images can be displayed through the current image (current layer) by specifying the α control area for the current layer (transparent processing) and chroma-keying.

(1) α-blending

— The current layer has an α-controlled area and is blended with the lower layer according to the α value.
— The α value can be increased or decreased in synchronization with the frame rate (for fade-in and fade-out).
— The degree of transparency is specified in units of 1/256 × 100%.

Figure 2 shows an example where the display includes α blending. Table 4 lists the blending ratios of the α-blending function according to the α value.

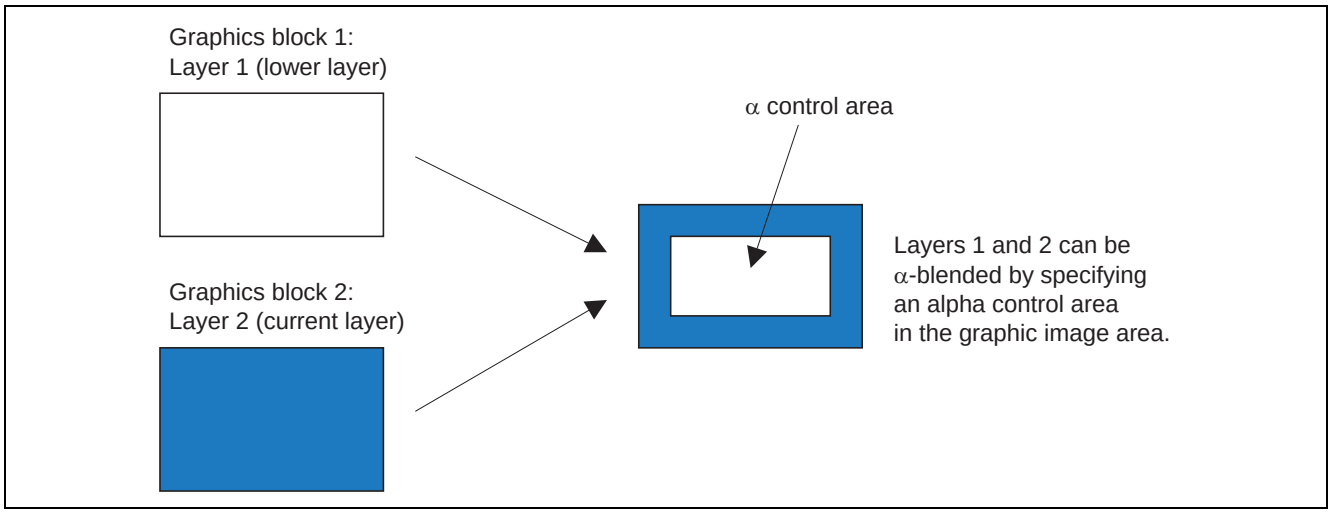


Figure 2 Display Includes α Blending Example (Overlaid Display of Layers 1 and 2)

Table 4 Blending Ratios of the α-Blending Function According to the α Value

α Value	Current Layer	Lower Layer	Percentage Contributions to the Display
255	255/255	0/255	• Current layer: 100 [%] • Lower layer: 0 [%]
254	254/255	1/255	• Current layer: 254/255 × 100 [%] • Lower layer: 1/255 × 100 [%]
. . .	. . .	. . .	.
1	1/255	254/255	• Current layer: 1/255 × 100 [%] • Lower layer: 254/255 × 100 [%]
0	0/255	255/255	• Current layer: 0 [%] • Lower layer: 100 [%]

(2) Chroma-keying

In chroma-keying, any pixel in the image with color matching the specified chroma key is processed as follows.

— The color is replaced in accord with the color information set in the chroma-key control register and chroma-key color register.
— After color replacement, the ALPHA [23:16] bits in the chroma-key color register control the α value.

Figure 3 shows an example where the display includes chroma-keying. Table 5 lists the blending ratios of the chroma-keying function according to the α value.

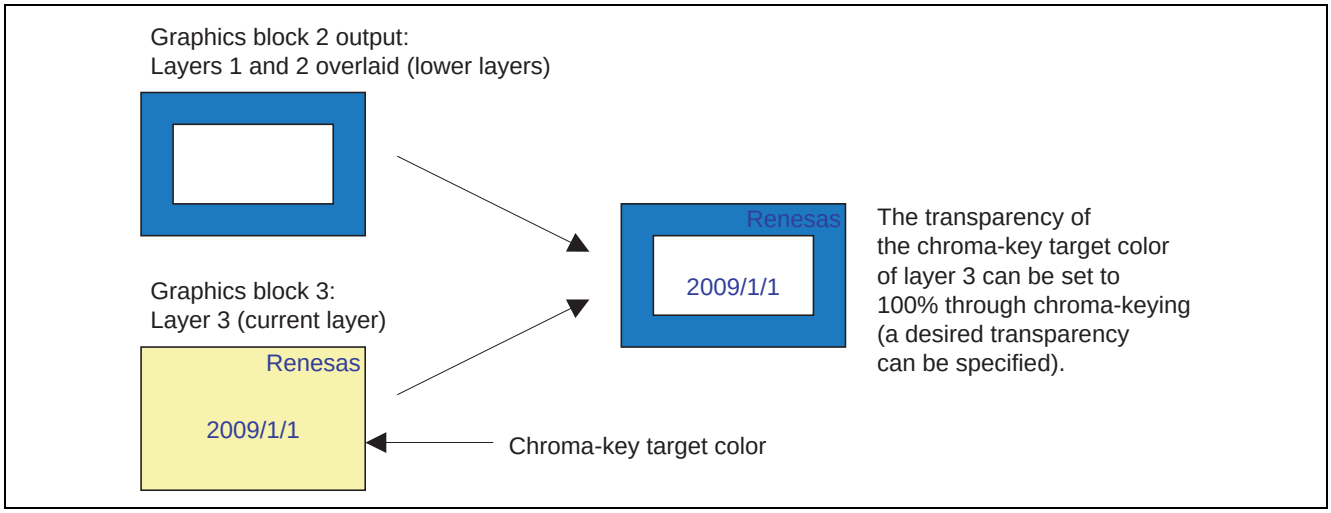


Figure 3 Display Includes Chroma-Keying Example (Overlaid Display of Layers 2 and 3)

Table 5 Blending Ratios of the Chroma-Keying Function According to the α Value

α Value	Current Layer	Lower Layer	Percentage Contributions to the Display
255	255/255	0/255	• Current layer: 100 [%] • Lower layer: 0 [%]
254	254/255	1/255	• Current layer: 254/255 × 100 [%] • Lower layer: 1/255 × 100 [%]
. . .	. . .	. . .	.
1	1/255	254/255	• Current layer: 1/255 × 100 [%] • Lower layer: 254/255 × 100 [%]
0	0/255	255/255	• Current layer: 0 [%] • Lower layer: 100 [%]

2.1.6 Sync Signal Output Format

Figure 4 shows an example of the format for the output of synchronizing signals to the TFT-LCD panel. The figure illustrates the timing of synchronizing signals that are set in the display control and graphics blocks and the relationship among the synchronous signal area including the blanking interval, the data enable area, and the graphic image area. Make settings that suit the TFT-LCD panel you will be using.

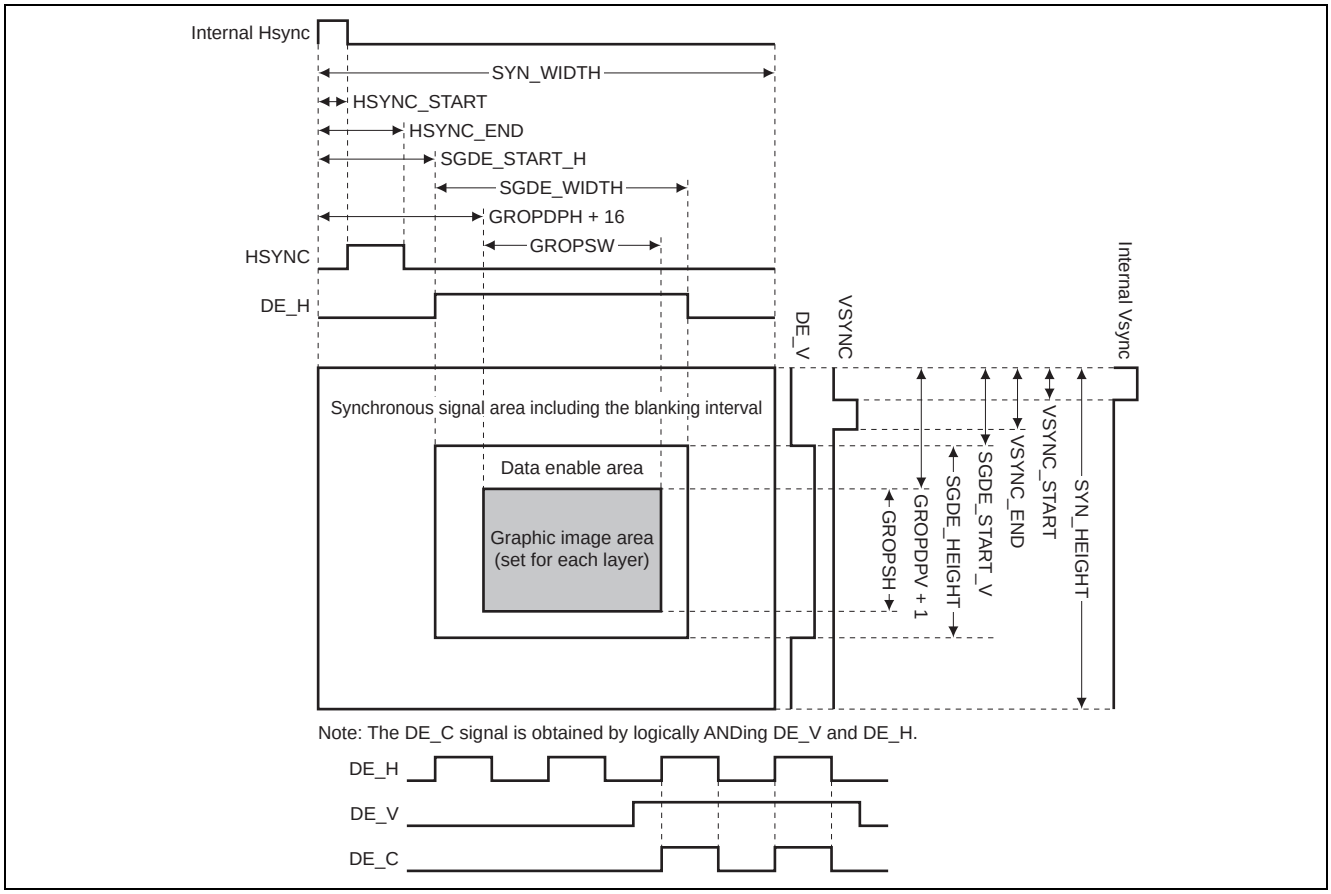


Figure 4 Screen Format

2.2 TFT-LCD Panel Specifications

Table 6 lists the specification of the TFT-LCD panel to use in this application. The specifications of the TFT-LCD panel used for this application note (TX18D55VM1CAA, manufactured by Hitachi Displays, Ltd.) are listed in the table below. As detailed specifications differ with the TFT-LCD panel, be sure to check the data sheet for the product you will be using.

2.2.1 General Specifications

Table 6 lists the general specifications of the TFT-LCD panel to use in this application.

Table 6 TFT-LCD Panel General Specifications (Excerpt from Datasheet)

Item	Specifications
Resolution	Wide-VGA
Number of pixels	H 800 × V 480 (Number of dots: H (800 × 3) × V 480)
Pixel configuration	R, G, B vertical stripes
Number of colors	260,000 colors
Input signal	CMOS RGB (6 bits each digital)

2.2.2 Pin Functions

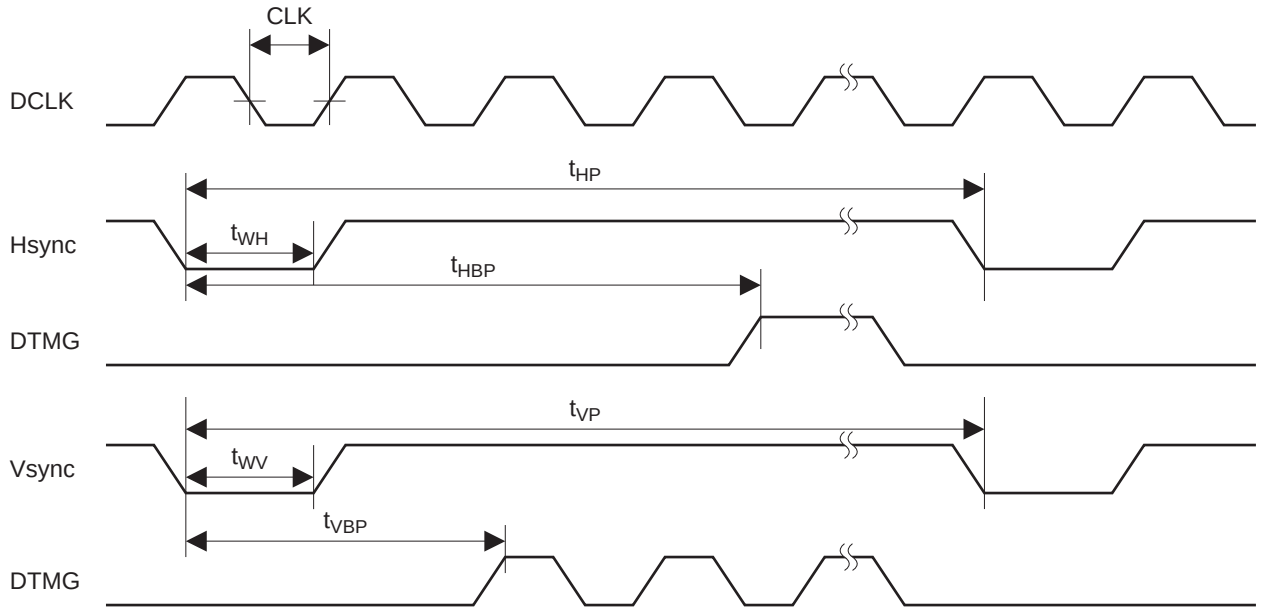
Table 7 lists the pin functions of the TFT-LCD panel used in this application.

Table 7 TFT-LCD Panel Pin Functions (Excerpt from Datasheet)

Symbol	Description
DCLK	Dot clock
DTMG	Display timing signal
R5-0	Red data signal (MSB: R5)
G5-0	Green data signal (MSB: G5)
B5-0	Blue data signal (MSB: B5)

2.2.3 Interface Timing

Figure 5 shows the interface timing of the TFT-LCD panel used in this application. Table 8 lists the timing characteristics.



Note: Data are latched on falling edges of DCLK.
Operation is with synchronization obtained by using the DTMG signal only.
While no valid data are being input, the DTMG signal must be at the low level.

Figure 5 TFT-LCD Panel Interface Timing Example (Excerpt from Datasheet)

Table 8 TFT-LCD Panel Timing Characteristics (Excerpt from Datasheet)

Item		Symbol	Min	Typ	Max	Unit
DCLK	Cycle time	t_{CLK}	25	30	33	ns
Hsync	Cycle time	t_{HP}	944	1056	1088	t_{CLK}
	Valid width	t_{WH}	4	128	—	
Vsync	Cycle time	t_{VP}	515	525	610	t_{HP}
	Valid width	t_{WV}	1	2	—	
DTMG	Horizontal back porch time	t_{HBP}	7	88	—	t_{CLK}
	Vertical back porch time	t_{VBP}	4	32	—	t_{HP}

2.3 TFT-LCD Panel Circuit Example

2.3.1 Pin Connection Example

Figure 6 shows the TFT-LCD panel hardware connection in this application.

Operation of the TX18D55VM1CAA is with synchronization obtained by using the display-timing signal (DTMG) only, so the HSYNC and VSYNC signals need not be connected. The display-enable signal (DE_C) is connected to the DTMG input. The source clock signal for the panel (input to DCLKIN) is output through DCLKOUT with no frequency multiplication.

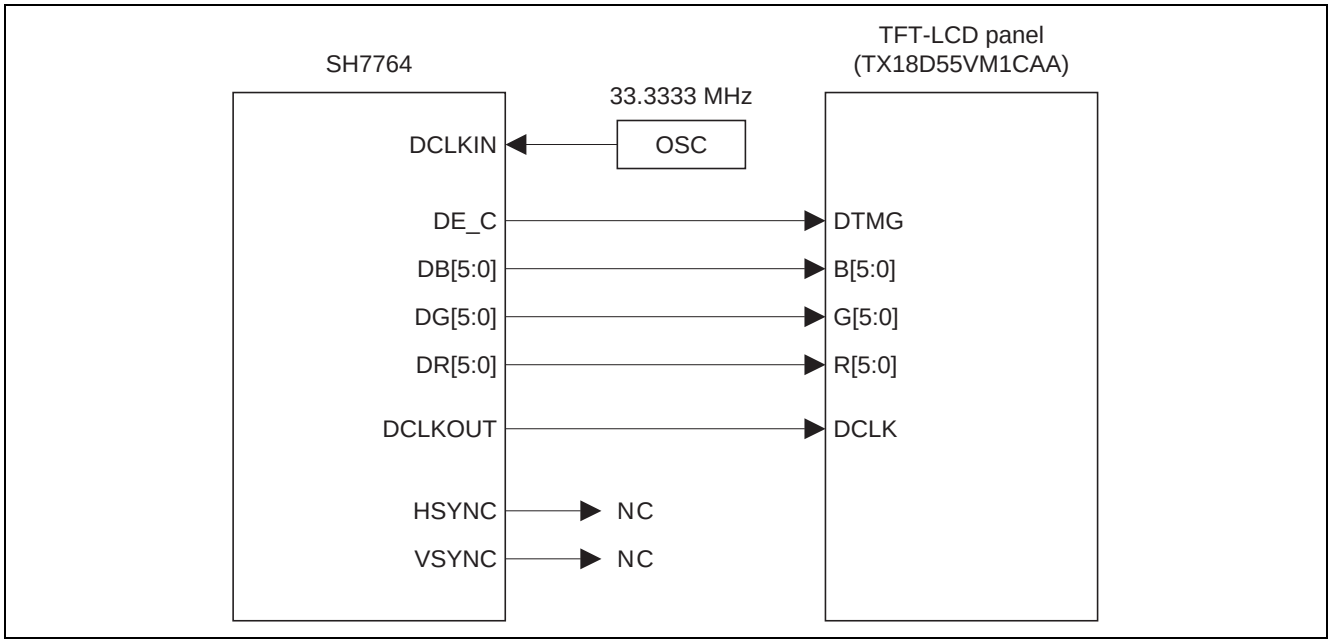


Figure 6 TFT-LCD Panel Hardware Connection

2.4 How to Calculate the Bus Occupancy Rate

Adding a display layer increases the amount of data transfer from external memory to the VDC2 module, raising the load on the internal bus. The bus occupancy rate for VDC2 is the sum of the bus occupancy rates calculated for the individual layers; as a rough guide, adjust the composition so that the rate is no greater than 40%. Since acceptable bus occupancy rates for display processing vary with the system, be sure to select the TFT-LCD panel and consider the layer configuration accordingly.

$$\text{Bus occupation ratio (\%)} = \frac{\text{Overhead coefficient} \times \text{Total number of display pixels} \times \text{Frame rate (Hz)} \times \text{Number of colors (bpp)}}{\text{Bus clock} \times \text{Bus width (= 32 bits)}} \times 100$$

Example) Total number of display pixels = H800 × V480, Frame rate = 60 Hz,
Number of colors 16 bits, Overhead coefficient = 2.00, Bus clock = 108 MHz

$$\text{Bus occupation ratio} = \frac{2.00 \times 800 \times 480 \times 60 \times 16}{108 \times 10^6 \times 32} \times 100 = 21.3\%$$

2.5 Sample Program Specifications

This section describes the specifications of the sample program and shows the flow chart of each processing.

2.5.1 Specifications

- An image with four superposed layers is displayed on a TFT-LCD panel with WVGA resolution (H 800 × V 480).
- Layers 1 to 4 are superposed in order at a fixed interval. In the finished display, layer 3 controls the chroma-keying function and layer 4 controls the α value.
- In layer 1, the image is white and takes up 800 × 480 pixels.
- In layer 2, the image is a set of colored bars and takes up 400 × 240 pixels. The display start position is 200 × 120, so the bars are displayed in the center of the screen.
- In layer 3, the image is yellow in the lower-right quadrant of the screen and black in other areas, and takes up 800 x 480 pixels. This layer also controls the chroma-keying function. Black is the transparent color, so the lower layers are displayed in the black portion.
- For chroma-key control in layer 3, the target color is black and the α value is 0.
- In layer 4, the image is black and takes up 800 × 480 pixels. Alpha control for fade-in processing is also applied in this layer so that the lower layers are displayed.
- For α control in layer 4, the size of the α-controlled area is 720 × 480, the start position for α control is 40 × 0, the default α value is specified as 255, and five is subtracted from the α value per frame to execute fade-in processing.

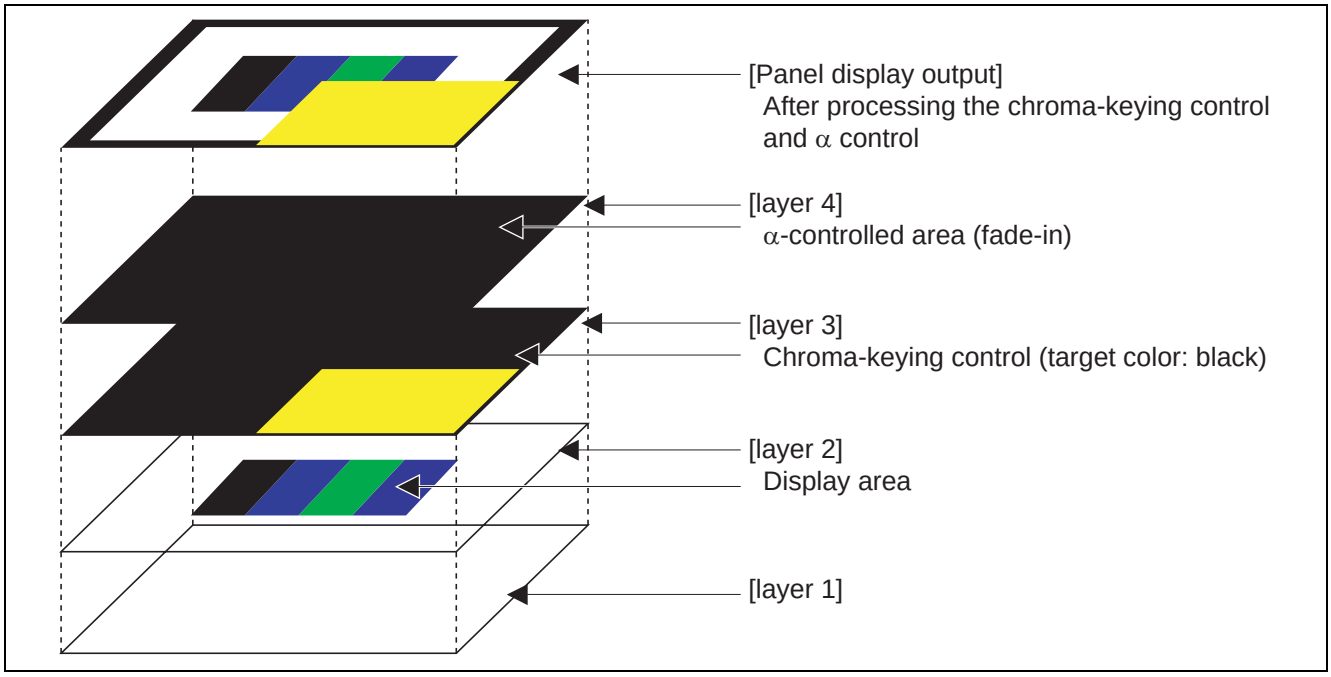


Figure 7 Display Image of a Sample Program

2.5.2 Main Flow Chart of the Sample Program

Figure 8 shows the main flow chart of the sample program. Initialization by the sample program is shown in figures 9 to 11, and the image overlapping layers 1 to 4 is displayed on the TFT-LCD panel after controlling the chroma-key and α values as shown in figures 12 and 13.

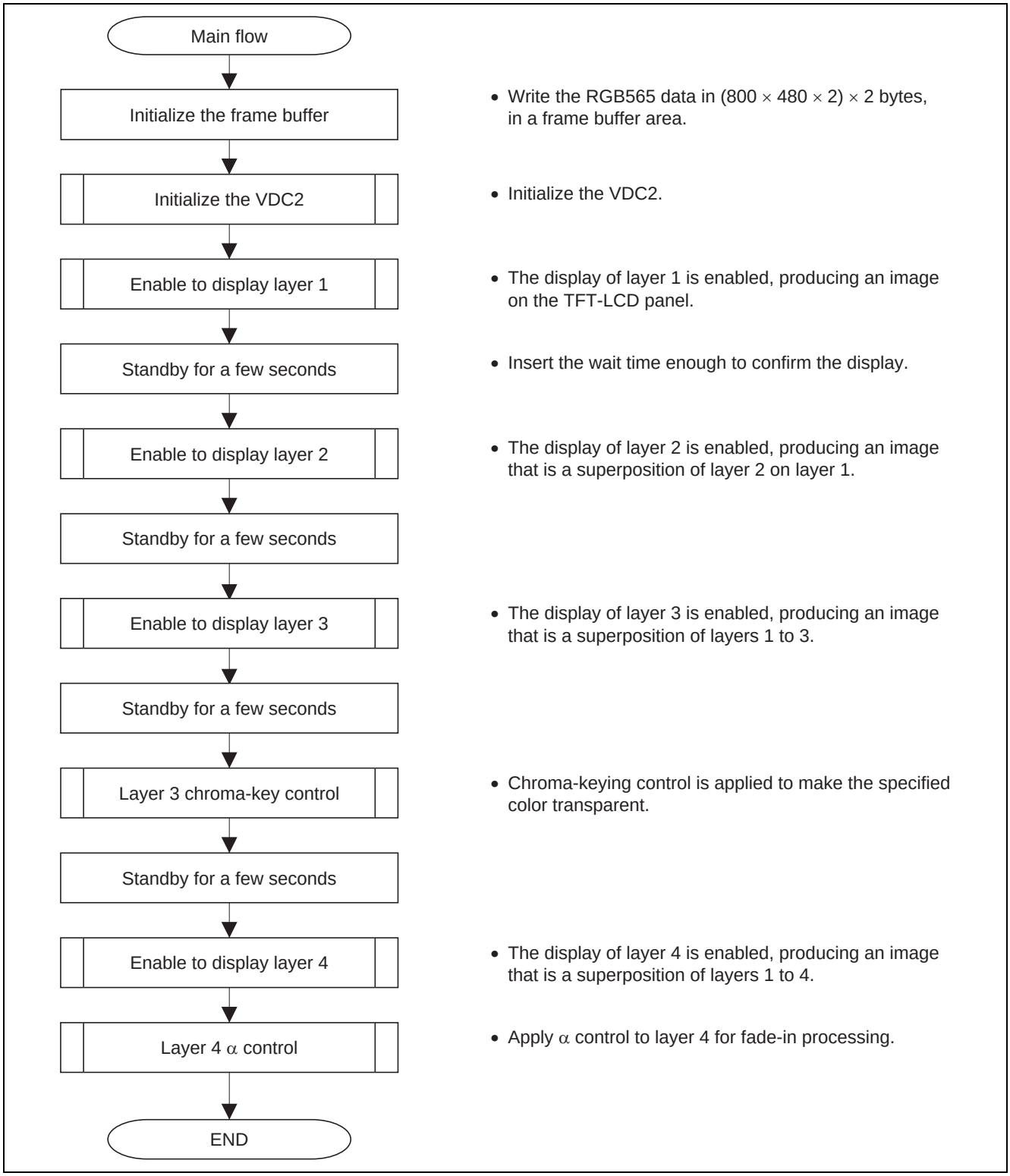


Figure 8 Sample Program Main Flow

2.5.3 Initialization of the VDC2

Figure 9 shows the flow for initialization of the VDC2.

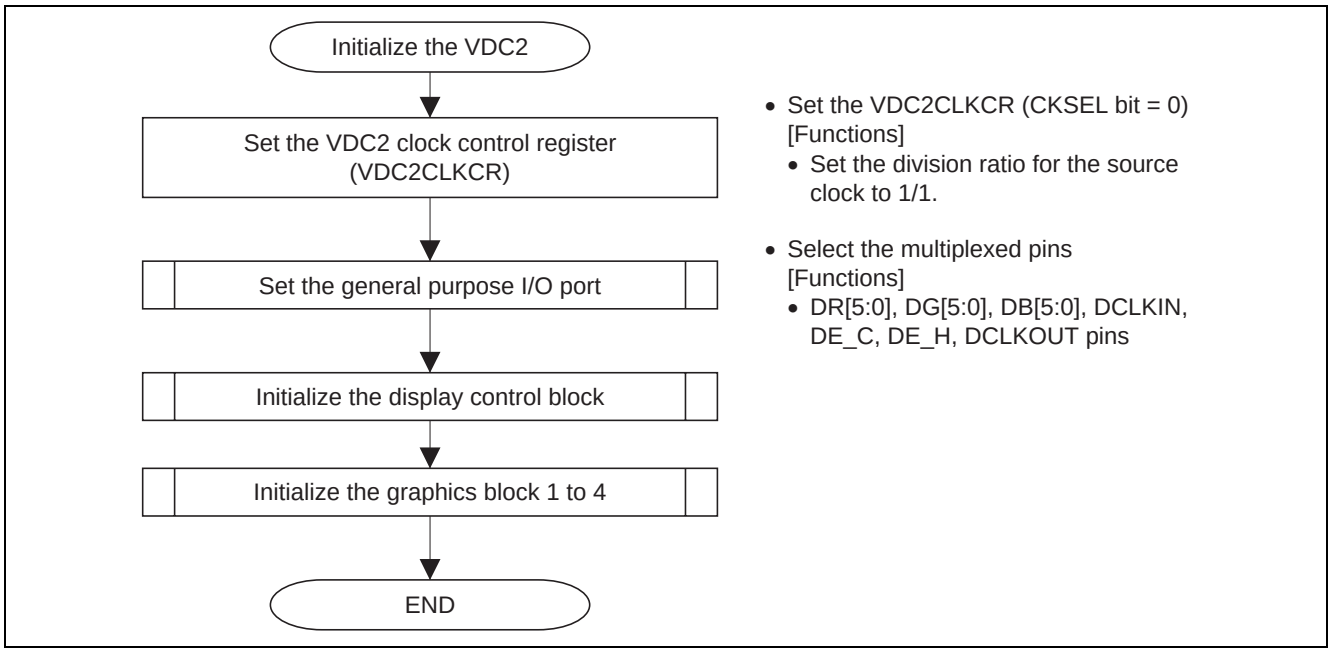


Figure 9 Flow for Initialization of the VDC2

2.5.4 Setting the Display Control Block

Figure 10 shows the setting examples of the display control block. Follow section 2.5.5 and this procedure to set the control signal output for the TFT-LCD panel. Values listed in figure 10 are set according to the TFT-LCD panel specifications described in section 2.2.

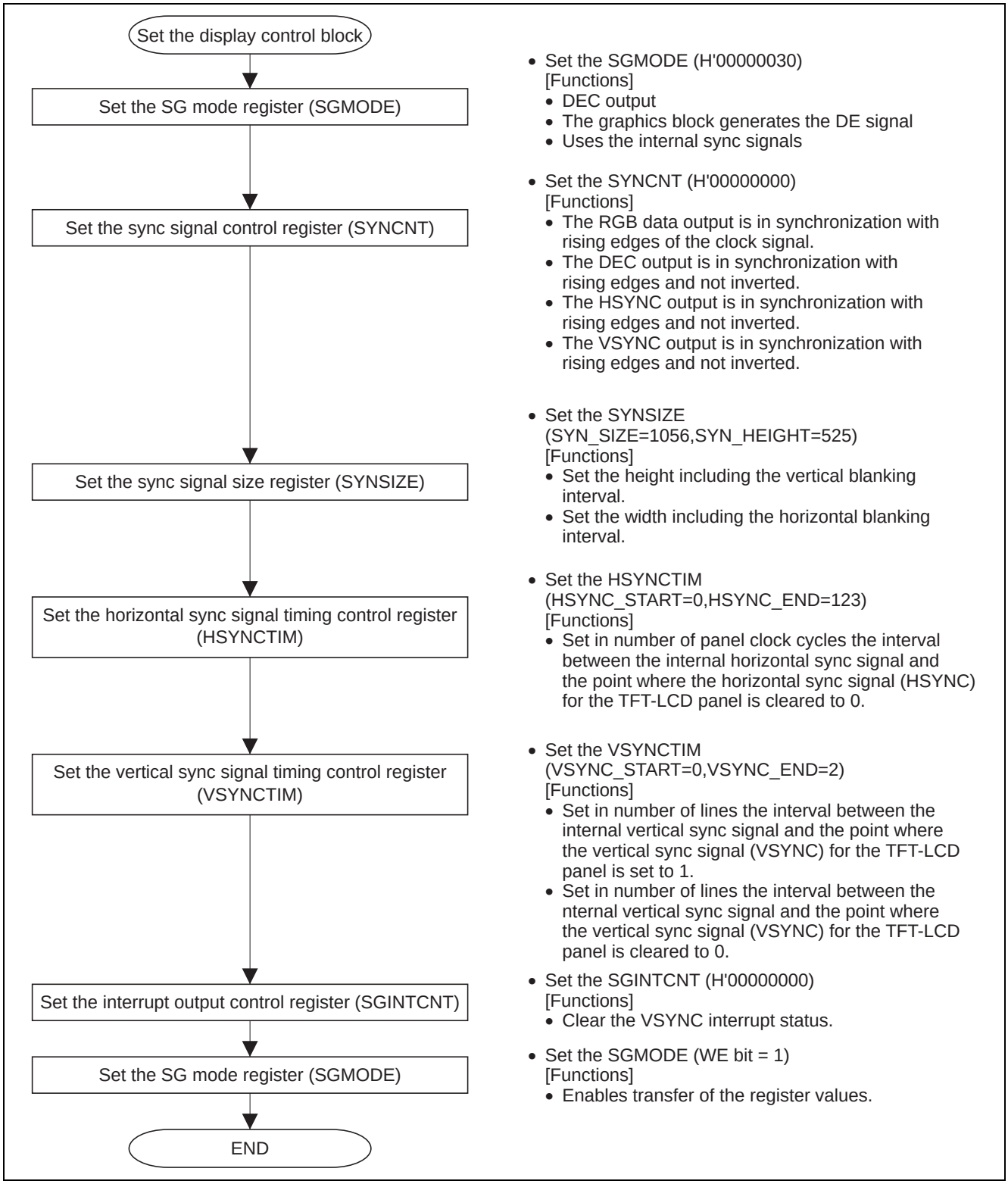


Figure 10 Setting Examples of the Graphics Control Block

2.5.5 Setting the Graphics Blocks

Figure 11 shows the setting example of the graphics block. Follow section 2.5.4 and this procedure to display the graphics image data in a specified area of the panel. The same setting procedure applies to graphics blocks 1 to 4.

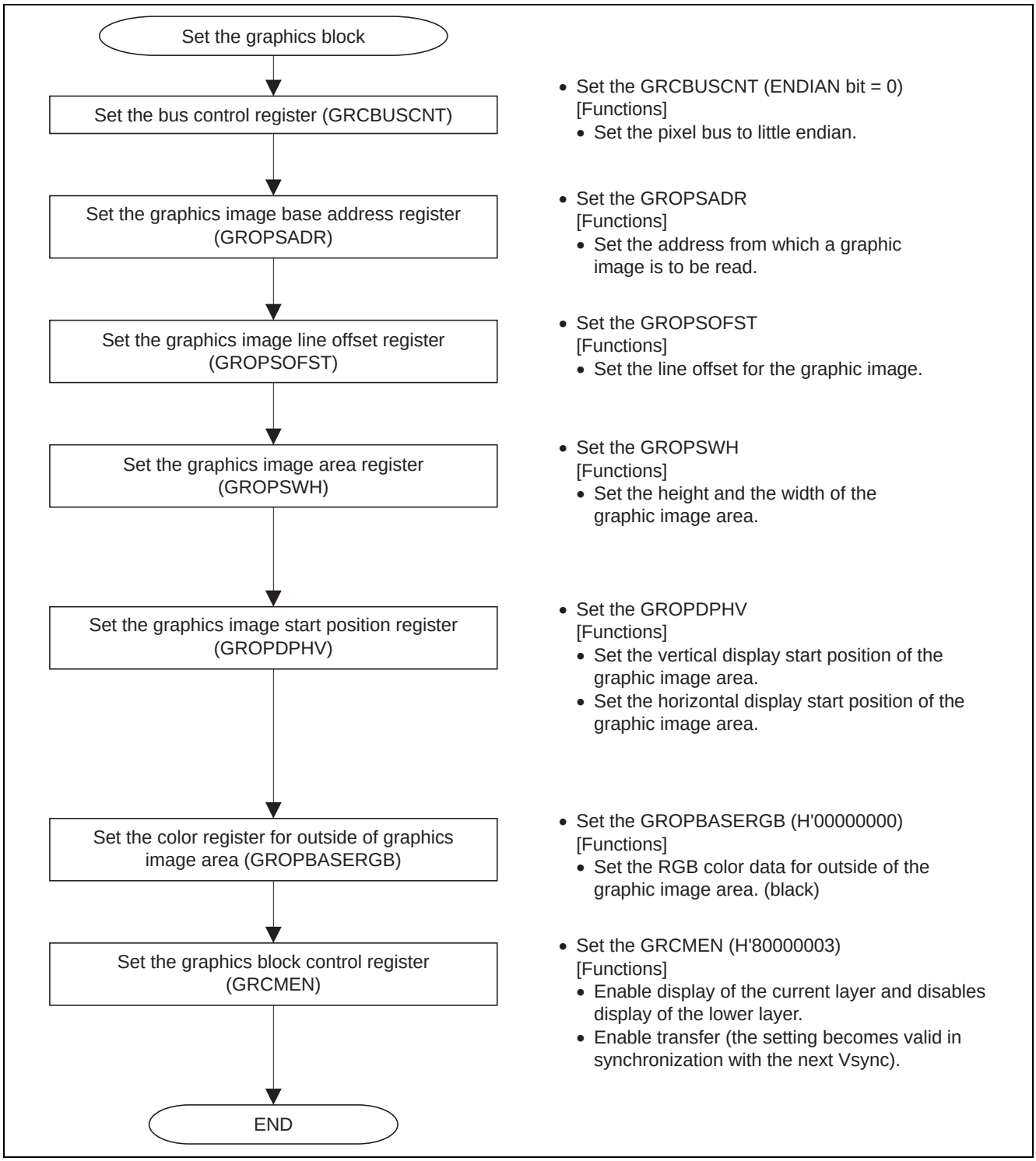


Figure 11 Setting Examples of the Graphics Block

2.5.6 Chroma-Keying Control

Figure 12 shows an example of the procedure for setting up chroma-keying control for a graphics block. This procedure causes the specified RGB color to be replaced (become transparent). Execute this procedure in combination with that in section 2.5.4, Setting the Display Control Blocks, and that in section 2.5.5, Setting the Graphics Blocks. Only layer 3 is used for chroma-keying control.

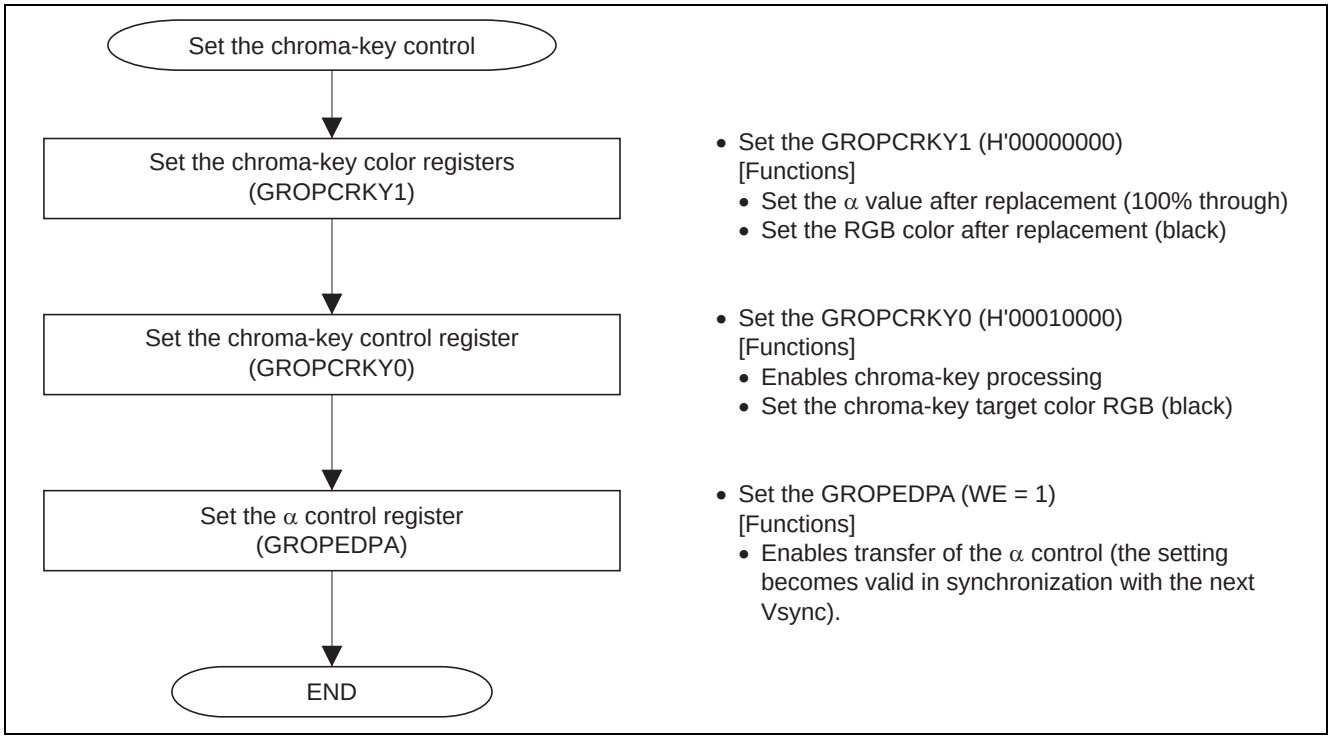


Figure 12 Setting Examples of the Chroma-Key Control

2.5.7 α Control

Figure 13 shows an example for setting up α control for a graphics block. This procedure leads to α-blending of the current and lower layers. Execute this procedure in combination with that in section 2.5.4, Setting the Display Control Blocks, and that in section 2.5.5, Setting the Graphics Blocks. Only layer 4 is used for α-blend processing.

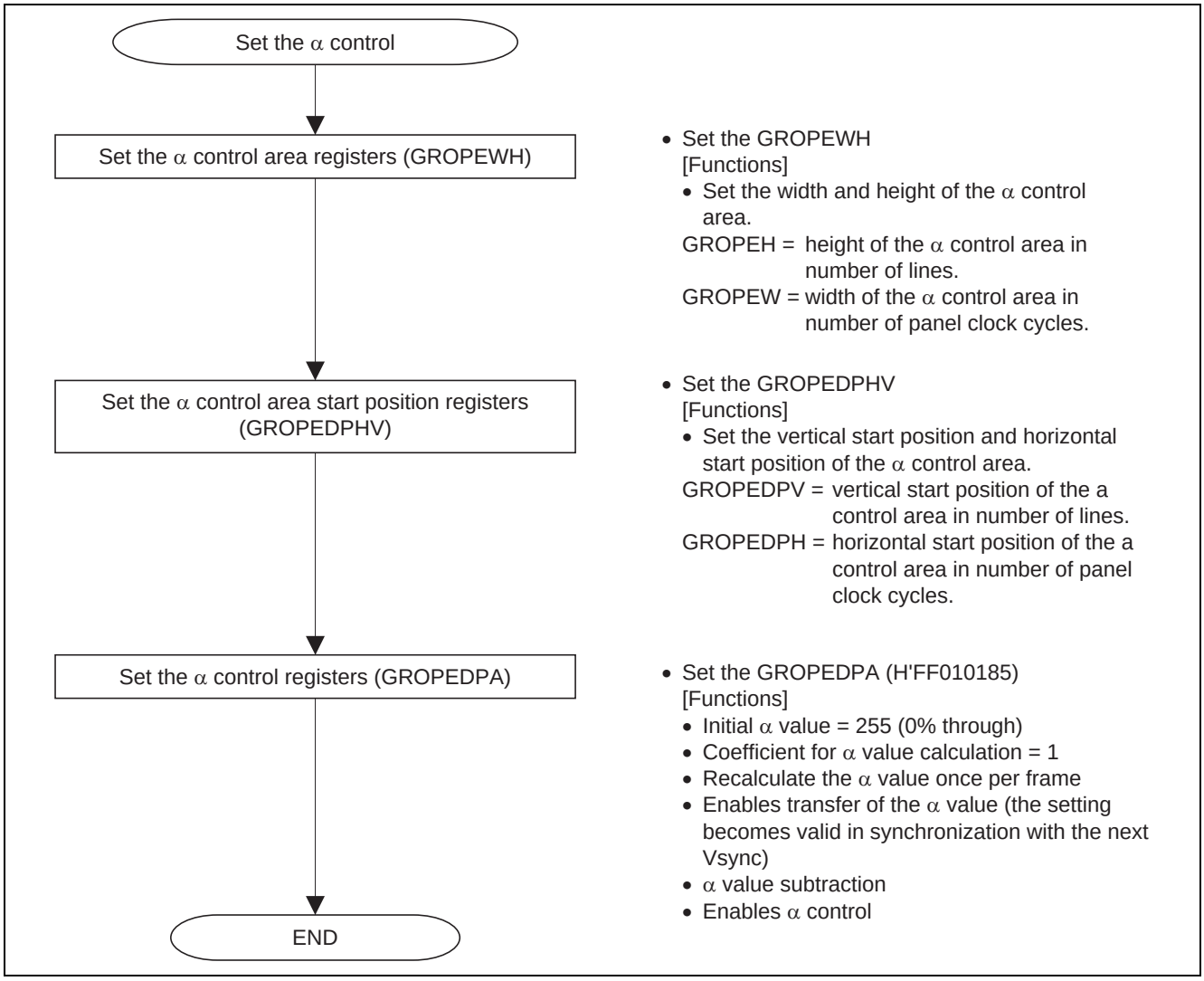


Figure 13 Setting Examples of the α Control

3. Sample Program "vdc2.c"

3.1 Listings of Sample Program "Macro definition"

```
/*""FILE COMMENT""*********** Technical reference data ************************
*
*       System Name : SH7764 Sample Program
*       File Name   : vdc2.c
*       Abstract    : VDC2 TFT-LCD Panel Display Example
*       Version     : 1.00.00
*       Device      : SH7764
*       Tool-Chain  : High-performance Embedded Workshop (Ver.4.05.01).
*                   : C/C++ compiler package for the SuperH RISC engine family
*                   :                               (Ver.9.03 Release00).
*       OS          : none
*       H/W Platform: R0K507764E001BR
*       Disclaimer  :
*                     <Note>
*                     This sample program is provided only as a reference and
*                     its operation is not guaranteed.
*                     Use this sample program as a technical reference when
*                     developing software.
*
*       The information described here may contain technical inaccuracies or
*       typographical errors. Renesas Technology Corporation and Renesas Solutions
*       assume no responsibility for any damage, liability, or other loss rising
*       from these inaccuracies or errors.
*
*       Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
*       AND Renesas Solutions Corp. All Rights Reserved
*
*       History     : June.01,2009 Ver.1.00.00
*""FILE COMMENT END""*************************************************************/
#include "iodefine.h"

/* ==== Macro definition ==== */
/* ---- TFT panel display module ---- */
#define TFT_TOTAL_CLOCK     1056    /* Width including the blanking interval    */
#define TFT_TOTAL_LINE      525     /* Height including the blanking interval   */
#define TFT_PANEL_CLOCK     800     /* Number of pixels in horizontal direction */
#define TFT_PANEL_LINE      480     /* Number of pixels in vertical direction   */
#define TFT_DTMG_START_H 88     /* Horizontal display start position */
#define TFT_DTMG_START_V 32     /* Vertical display start position   */
#define TFT_HSYNC_START     0       /* Hsync pulse width start position  */
#define TFT_HSYNC_END       123     /* Hsync pulse width end position    */
#define TFT_VSYNC_START     0       /* Vsync pulse width start position  */
#define TFT_VSYNC_END       2       /* Vsync pulse width end position    */
/* ---- Graphics block parameter ---- */
#define GRAPHICS1_Y_SIZE    TFT_PANEL_LINE
          /* Height of graphics block 1 */
#define GRAPHICS1_X_SIZE    TFT_PANEL_CLOCK
          /* Width of graphics block 1 */
#define GRAPHICS1_OFFSET    TFT_PANEL_CLOCK
          /* Line offset of graphics block 1 */
#define GRAPHICS1_POS_Y         0
          /* Display start position in vertical direction (0: Top of the panel) */
#define GRAPHICS1_POS_X         0
          /* Display start position in horizontal direction (0: Leftmost of the panel) */
#define GRAPHICS1_BG_COLOR      0x0000
          /* Color of graphics block 1: Black */

#define GRAPHICS2_Y_SIZE    (TFT_PANEL_LINE / 2)
          /* Height of graphics block 2 */
#define GRAPHICS2_X_SIZE    (TFT_PANEL_CLOCK / 2)
          /* Width of graphics block 2 */
#define GRAPHICS2_OFFSET    TFT_PANEL_CLOCK
          /* Line offset of graphics block 2 */
#define GRAPHICS2_POS_Y         (TFT_PANEL_LINE / 4)
          /* Display start position in vertical direction (0: Top of the panel) */
```

3.2 Listings of Sample Program "Function prototype declaration, Variable definition"

```
66    #define GRAPHICS2_POS_X         (TFT_PANEL_CLOCK / 4)
67              /* Display start position in horizontal direction (0: Leftmost of the panel) */
68    #define GRAPHICS2_BG_COLOR      0x0000
69              /* Color of graphics block 2: Black */
70
71    #define GRAPHICS3_Y_SIZE    TFT_PANEL_LINE
72              /* Height of graphics block 3 */
73    #define GRAPHICS3_X_SIZE    TFT_PANEL_CLOCK
74              /* Width of graphics block 3 */
75    #define GRAPHICS3_OFFSET    TFT_PANEL_CLOCK
76              /* Line offset of graphics block 3 */
77    #define GRAPHICS3_POS_Y         0
78              /* Display start position in vertical direction (0: Top of the panel) */
79    #define GRAPHICS3_POS_X         0
80              /* Display start position in horizontal direction (0: Leftmost of the panel) */
81    #define GRAPHICS3_BG_COLOR      0x0000
82              /* Color of graphics block 3: Black */
83    #define GRAPHICS3_CRKY_COLOR 0x0000
84              /* Specified chroma-key color: Black */
85
86    #define GRAPHICS4_Y_SIZE    TFT_PANEL_LINE
87              /* Height of graphics block 4 */
88    #define GRAPHICS4_X_SIZE    TFT_PANEL_CLOCK
89              /* Width of graphics block 4 */
90    #define GRAPHICS4_LINE_OFFSET   TFT_PANEL_CLOCK
91              /* Line offset of graphics block 4 */
92    #define GRAPHICS4_POS_Y         0
93              /* Display start position in vertical direction (0: Top of the panel) */
94    #define GRAPHICS4_POS_X         0
95              /* Display start position in horizontal direction (0: Leftmost of the panel) */
96    #define GRAPHICS4_BG_COLOR      0x0000
97              /* Color of graphics block 4: Black */
98    #define GRAPHICS4_ALPHA_Y_SIZE  TFT_PANEL_LINE
99              /* Height of the alpha control area */
100   #define GRAPHICS4_ALPHA_X_SIZE  (TFT_PANEL_CLOCK - 80)
101             /* Width of the alpha control area */
102   #define GRAPHICS4_ALPHA_POS_Y   0
103             /* Vertical start position of the alpha control area */
104   #define GRAPHICS4_ALPHA_POS_X   ((TFT_PANEL_CLOCK - GRAPHICS4_ALPHA_X_SIZE) / 2)
105             /* Horizontal start position of the alpha control area */
106
107   /* ==== Function prototype declaration ==== */
108   void vdc_main(void);
109   void vdc2_initial(void);
110   void vdc2_port_set(void);
111   void vdc2_display_control_initial(void);
112   void vdc2_graphic_layer1_initial(void);
113   void vdc2_graphic_layer2_initial(void);
114   void vdc2_graphic_layer3_initial(void);
115   void vdc2_graphic_layer4_initial(void);
116   void vdc2_graphic_layer1_enable(void);
117   void vdc2_graphic_layer2_enable(void);
118   void vdc2_graphic_layer3_enable(void);
119   void vdc2_graphic_layer4_enable(void);
120   void vdc2_graphic_layer3_chromakey(void);
121   void vdc2_graphic_layer4_alpha(void);
122   void fill_rect(unsigned int x, unsigned int y,
123       unsigned int w, unsigned int h, unsigned short color,
124       unsigned int base_address, unsigned int line_offset);
125   void delay(void);
126
127   /* ==== Variable definition ==== */
128   #pragma section _VDC2_FRAME_BUFFER /* Places on a 16-byte boundary in the cache disabled area */
129   unsigned short frame_buffer[4][TFT_PANEL_LINE][TFT_PANEL_CLOCK];
130   #pragma section
```

3.3 Listings of Sample Program "Display main"

```
/*""FUNC COMMENT""*************************************************************
 * ID          :
 * Outline     : Display main
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void main(void);
 *-----------------------------------------------------------------------------
 * Function    : Increases layers to display on the TFT-LCD panel in a certain period
 *             : of time. Chroma-key control is enabled for layer 3 and alpha-blending
 *             : control (fading in) is used for layer 4.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc_main(void)
{
    /* ---- Draws 4 frame buffer planes ---- */
    fill_rect(0,0,GRAPHICS1_X_SIZE,GRAPHICS1_Y_SIZE,0xFFFF,
     (unsigned int)frame_buffer[0],GRAPHICS1_OFFSET);
      /* Fills the frame buffer 1 plane with white */

    fill_rect(0,0, GRAPHICS2_X_SIZE/4,GRAPHICS2_Y_SIZE,0x0000,
     (unsigned int)frame_buffer[1],GRAPHICS2_OFFSET);
    fill_rect(GRAPHICS2_X_SIZE/4 ,0,(GRAPHICS2_X_SIZE / 4) * 2,
     GRAPHICS2_Y_SIZE,0xF800,(unsigned int)frame_buffer[1],GRAPHICS2_OFFSET);
    fill_rect((GRAPHICS2_X_SIZE/4) * 2,0,(GRAPHICS2_X_SIZE / 4) * 3,
     GRAPHICS2_Y_SIZE,0x07E0,(unsigned int)frame_buffer[1],GRAPHICS2_OFFSET);
    fill_rect((GRAPHICS2_X_SIZE/4) * 3,0,GRAPHICS2_X_SIZE,GRAPHICS2_Y_SIZE,0x001F,
     (unsigned int)frame_buffer[1],GRAPHICS2_OFFSET);
      /* Draws a color bar with the size of 400x240 on the frame buffer 2 plane */

    fill_rect(0,0,GRAPHICS3_X_SIZE,GRAPHICS3_Y_SIZE,0x0000,
     (unsigned int)frame_buffer[2],GRAPHICS3_OFFSET);
      /* Fills the frame buffer 3 plane with black */
    fill_rect(TFT_PANEL_CLOCK - GRAPHICS2_X_SIZE,TFT_PANEL_LINE - GRAPHICS2_Y_SIZE,
     GRAPHICS2_X_SIZE,GRAPHICS2_Y_SIZE,0xFFE0,(unsigned int)frame_buffer[2],GRAPHICS3_OFFSET);
      /* Draws a red rectangle with the size of 400x240 on the frame buffer 3 plane */

    fill_rect(0,0,GRAPHICS4_X_SIZE,GRAPHICS4_Y_SIZE,0x0000,
     (unsigned int)frame_buffer[3],GRAPHICS4_LINE_OFFSET);
      /* Fills the frame buffer 4 plane with black */

    /* ---- Initializes the VDC2 module ---- */
    vdc2_initial();

    /* ---- Outputs 4 frame buffer planes on the TFT-LCD ---- */
    vdc2_graphic_layer1_enable(); /* Full screen */
    delay(); /* Waits for several seconds */

    vdc2_graphic_layer2_enable(); /* Size of the center (400x240) */
    delay(); /* Waits for several seconds */

    vdc2_graphic_layer3_enable(); /* Full screen */
    delay(); /* Waits for several seconds */

    vdc2_graphic_layer3_chromakey(); /* Chroma-key (black) */
    delay(); /* Waits for several seconds */

    vdc2_graphic_layer4_enable(); /* Full screen */
    vdc2_graphic_layer4_alpha();  /* Alpha-blends the 720x480 size and fades in */

    while(1){};
}
```

3.4 Listings of Sample Program "VDC2 initialization"

```
/*""FUNC COMMENT""*************************************************************
 * ID          :
 * Outline     : VDC2 initialization
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_initial(void);
 *-----------------------------------------------------------------------------
 * Function    : Initializes the VDC2 for displaying. Layers 1 to 4 are used.
 *             : TFT-LCD panel TX09D55VM1CDA (Hitachi Displays) is used in this
 *             : application.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*********************************************************/
void vdc2_initial(void)
{
    CPG.VDC2CLKCR.BIT._CKSEL = 0;   /* Clock divider 1/1 */
    vdc2_port_set();            /* I/O pin setting */
    vdc2_display_control_initial(); /* Display control block setting */
    vdc2_graphic_layer1_initial();  /* Layer 1 setting */
    vdc2_graphic_layer2_initial();  /* Layer 2 setting */
    vdc2_graphic_layer3_initial();  /* Layer 3 setting */
    vdc2_graphic_layer4_initial();  /* Layer 4 setting */
}
```

3.5 Listings of Sample Program "I/O pin settings"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : I/O pin setting
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_port_set(void);
 *-----------------------------------------------------------------------------
 * Function    : Sets I/O pins for the VDC2.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_port_set(void)
{
    /* ---- DR3, DR2, DR1, DR0, DG5, DG4, DG3, DG2 ---- */
    GPIO.PTSEL_G.BIT._PTSEL_G7=GPIO.PTSEL_G.BIT._PTSEL_G6=
    GPIO.PTSEL_G.BIT._PTSEL_G5=GPIO.PTSEL_G.BIT._PTSEL_G4=
    GPIO.PTSEL_G.BIT._PTSEL_G3=GPIO.PTSEL_G.BIT._PTSEL_G2=
    GPIO.PTSEL_G.BIT._PTSEL_G1=GPIO.PTSEL_G.BIT._PTSEL_G0=1;
    /* ---- DB3, DB2, DB1, DG1, DG0, DB5, DB4, ---- */
    GPIO.PTSEL_I.BIT._PTSEL_I7=GPIO.PTSEL_I.BIT._PTSEL_I6=
    GPIO.PTSEL_I.BIT._PTSEL_I5=GPIO.PTSEL_I.BIT._PTSEL_I4=
    GPIO.PTSEL_I.BIT._PTSEL_I3=GPIO.PTSEL_I.BIT._PTSEL_I2=
    GPIO.PTSEL_I.BIT._PTSEL_I1 = 1;
    /* ---- DB0, DCLKIN, DE_C/DE_H ---- */
    GPIO.PTSEL_K.BIT._PTSEL_K4=GPIO.PTSEL_K.BIT._PTSEL_K2=
    GPIO.PTSEL_K.BIT._PTSEL_K0=1;
    /* ---- DCLKOUT, DR4, DR5 ---- */
    GPIO.PTSEL_H.BIT._PTSEL_H2=GPIO.PTSEL_H.BIT._PTSEL_H1=
    GPIO.PTSEL_H.BIT._PTSEL_H0=1;
    GPIO.PTIO_H.BIT._PTIO_H2=GPIO.PTIO_H.BIT._PTIO_H1=
    GPIO.PTIO_H.BIT._PTIO_H0=0;
}
```

3.6 Listings of Sample Program "Display control block initialization"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Display control block initialization
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_display_control_initial(void);
 *-----------------------------------------------------------------------------
 * Function    : Initializes the display control block.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_display_control_initial(void)
{
    /* ---- Selects the sync signal and outputs DE_C ---- */
    VDC2.SGMODE.LONG = 0x00000030;
      /* bit5 (DE_SEL)=1 outputs DE_C                        */
        /* bit4 (DEC_MODE)=1 generates the DE signal in the graphics block */
        /* bit1 (SYNC_SEL)=0 uses the internal sync signal           */

    /* ---- Sync signal timing setting ---- */
    VDC2.SYNCNT.LONG = 0x00000000;
        /* RGB data output is synchronized with the rising edge of the clock     */
        /* DE_C output is synchronized with the rising edge of the clock without inversion  */
        /* HSYNC output is synchronized with the rising edge of the clock without inversion */
        /* VSYNC output is synchronized with the rising edge of the clock without inversion */

    /* ---- Specifies the size of the sync signal (including the blanking interval) ---- */
    /* Number of pixels including the horizontal blanking interval and number of lines
including the vertical blanking interval */
    VDC2.SYNSIZE.BIT._SYN_WIDTH = TFT_TOTAL_CLOCK;
    VDC2.SYNSIZE.BIT._SYN_HEIGHT = TFT_TOTAL_LINE;

    /* ---- HSYNC pulse width ---- */
    VDC2.HSYNCTIM.BIT._HSYNC_START = TFT_HSYNC_START;
    VDC2.HSYNCTIM.BIT._HSYNC_END = TFT_HSYNC_END;

    /* ---- VSYNC pulse width ---- */
    VDC2.VSYNCTIM.BIT._VSYNC_START = TFT_VSYNC_START;
    VDC2.VSYNCTIM.BIT._VSYNC_END = TFT_VSYNC_END;

    /* ---- Clears the VSYNC interrupt status ---- */
    VDC2.SGINTCNT.LONG = 0x00000010;
    VDC2.SGMODE.BIT._WE = 1; /* Enables to transfer the register value */
}
```

3.7 Listings of Sample Program "Graphics block 1 initialization"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Graphics block 1 initialization
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer1_initial(void);
 *-----------------------------------------------------------------------------
 * Function    : Initializes graphics block 1.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer1_initial(void)
{
    /* ---- Disables to display (current and lower layers) ---- */
    VDC2_GR1.GRCMEN.LONG = 0x00000000;
        /* bit31 (WE)=0 disables transfer to register       */
        /* bit1 (DEN)=0 disables to display graphics     */
        /* bit0 (VEN)=0 disables to display lower-layer graphics */

    /* ---- Bus endianness ---- */
    VDC2_GR1.GRCBUSCNT.BIT._ENDIAN = 0; /* Little endian */

    /* ---- Setting for reading images from external memory ---- */
    VDC2_GR1.GROPSADR = (unsigned int *)frame_buffer[0];

    /* ---- Line offset setting ---- */
    VDC2_GR1.GROPSOFST = GRAPHICS1_OFFSET * sizeof(short);

    /* ---- Sets the output image area (LCD panel size) ---- */
    /* Width and height of the graphics image area */
    VDC2_GR1.GROPSWH.BIT._GROPSH = GRAPHICS1_Y_SIZE;
    VDC2_GR1.GROPSWH.BIT._GROPSW = GRAPHICS1_X_SIZE;

    /* ---- Vertical and horizontal display start positions of the image area ---- */
    VDC2_GR1.GROPDPHV.BIT._GROPDPV = TFT_DTMG_START_V - 1 + GRAPHICS1_POS_Y;
    VDC2_GR1.GROPDPHV.BIT._GROPDPH = TFT_DTMG_START_H - 16 + GRAPHICS1_POS_X;

    /* ---- Color for outside of the graphics image area ---- */
    VDC2_GR1.GROPBASERGB.LONG = GRAPHICS1_BG_COLOR;

    /* ---- Graphics output setting transfer ---- */
    VDC2_GR1.GRCMEN.BIT._WE = 1; /* Enables to transfer the register value */
}
```

3.8 Listings of Sample Program "Graphics block 2 initialization"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Graphics block 2 initialization
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer2_initial(void);
 *-----------------------------------------------------------------------------
 * Function    : Initializes graphics block 2.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer2_initial(void)
{
    /* ---- Disables to display (current layer) ---- */
    VDC2_GR2.GRCMEN.LONG = 0x00000001;
        /* bit31 (WE)=0 disables transfer to register       */
        /* bit1 (DEN)=0 disables to display graphics     */
        /* bit0 (VEN)=0 enables to display lower-layer graphics */

    /* ---- Bus endianness ---- */
    VDC2_GR2.GRCBUSCNT.BIT._ENDIAN = 0;    /* Little endian */

    /* ---- Setting for reading images from external memory ---- */
    VDC2_GR2.GROPSADR = (unsigned int *)frame_buffer[1];

    /* ---- Line offset setting ---- */
    VDC2_GR2.GROPSOFST = GRAPHICS2_OFFSET * sizeof(short);

    /* ---- Sets the output image area (LCD panel size) ---- */
    /* Width and height of the graphics image area */
    VDC2_GR2.GROPSWH.BIT._GROPSH = GRAPHICS2_Y_SIZE;
    VDC2_GR2.GROPSWH.BIT._GROPSW = GRAPHICS2_X_SIZE;

    /* ---- Vertical and horizontal display start positions of the image area ---- */
    VDC2_GR2.GROPDPHV.BIT._GROPDPV = TFT_DTMG_START_V - 1 + GRAPHICS2_POS_Y;
    VDC2_GR2.GROPDPHV.BIT._GROPDPH = TFT_DTMG_START_H - 16 + GRAPHICS2_POS_X;

    /* ---- Color for outside of the graphics image area ---- */
    VDC2_GR2.GROPBASERGB.LONG = GRAPHICS2_BG_COLOR;    /* (RGB565, black)*/

    /* ---- Graphics output setting transfer ---- */
    VDC2_GR2.GRCMEN.BIT._WE = 1;    /* Enables to transfer the register value */
}
```

3.9 Listings of Sample Program "Graphics block 3 initialization"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Graphics block 3 initialization
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer3_initial(void);
 *-----------------------------------------------------------------------------
 * Function    : Initializes graphics block 3.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer3_initial(void)
{
    /* ---- Disables to display (current layer) ---- */
    VDC2_GR3.GRCMEN.LONG = 0x00000001;
        /* bit31 (WE)=0 disables transfer to register       */
        /* bit1 (DEN)=0 disables to display graphics     */
        /* bit0 (VEN)=0 enables to display lower-layer graphics */

    /* ---- Bus endianness ---- */
    VDC2_GR3.GRCBUSCNT.BIT._ENDIAN = 0; /* Little endian */

    /* ---- Setting for reading images from external memory ---- */
    VDC2_GR3.GROPSADR = (unsigned int *)frame_buffer[2];

    /* ---- Line offset setting ---- */
    VDC2_GR3.GROPSOFST = GRAPHICS3_OFFSET * sizeof(short);

    /* ---- Sets the output image area (LCD panel size) ---- */
    /* Width and height of the graphics image area */
    VDC2_GR3.GROPSWH.BIT._GROPSH = GRAPHICS3_Y_SIZE;
    VDC2_GR3.GROPSWH.BIT._GROPSW = GRAPHICS3_X_SIZE;

    /* ---- Vertical and horizontal display start positions of the image area ---- */
    VDC2_GR3.GROPDPHV.BIT._GROPDPV = TFT_DTMG_START_V - 1 + GRAPHICS3_POS_Y;
    VDC2_GR3.GROPDPHV.BIT._GROPDPH = TFT_DTMG_START_H - 16 + GRAPHICS3_POS_X;

    /* ---- Color for outside of the graphics image area ---- */
    VDC2_GR3.GROPBASERGB.LONG = GRAPHICS3_BG_COLOR;

    /* ---- Graphics output setting transfer ---- */
    VDC2_GR3.GRCMEN.BIT._WE = 1; /* Enables to transfer the register value */
}
```

3.10 Listings of Sample Program "Graphics block 4 initialization"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Graphics block 4 initialization
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer4_initial(void);
 *-----------------------------------------------------------------------------
 * Function    : Initializes graphics block 4.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer4_initial(void)
{
    /* ---- Disables to display (current layer) ---- */
    VDC2_GR4.GRCMEN.LONG = 0x00000001;
        /* bit31 (WE)=0 disables transfer to register       */
        /* bit1 (DEN)=0 disables to display graphics     */
        /* bit0 (VEN)=0 enables to display lower-layer graphics */

    /* ---- Bus endianness ---- */
    VDC2_GR4.GRCBUSCNT.BIT._ENDIAN = 0; /* Little endian */

    /* ---- Setting for reading images from external memory ---- */
    VDC2_GR4.GROPSADR = (unsigned int *)frame_buffer[3];

    /* ---- Line offset setting ---- */
    VDC2_GR4.GROPSOFST = GRAPHICS4_LINE_OFFSET * sizeof(short);

    /* ---- Sets the output image area (LCD panel size) ---- */
    /* Width and height of the graphics image area */
    VDC2_GR4.GROPSWH.BIT._GROPSH = GRAPHICS4_Y_SIZE;
    VDC2_GR4.GROPSWH.BIT._GROPSW = GRAPHICS4_X_SIZE;

    /* ---- Vertical and horizontal display start positions of the image area ---- */
    VDC2_GR4.GROPDPHV.BIT._GROPDPV = TFT_DTMG_START_V - 1 + GRAPHICS4_POS_Y;
    VDC2_GR4.GROPDPHV.BIT._GROPDPH = TFT_DTMG_START_H - 16 + GRAPHICS4_POS_X;

    /* ---- Color for outside of the graphics image area ---- */
    VDC2_GR4.GROPBASERGB.LONG = GRAPHICS4_BG_COLOR;

    /* ---- Graphics output setting transfer ---- */
    VDC2_GR4.GRCMEN.BIT._WE = 1; /* Enables to transfer the register value */
}
```

3.11 Listings of Sample Program "Enables to display graphics block 1 and 2"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Enable to display graphics block 1
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer1_enable(void);
 *-----------------------------------------------------------------------------
 * Function    : Enables to display graphics block 1.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer1_enable(void)
{
    VDC2_GR1.GRCMEN.BIT._DEN = 1;   /* Enables the current layer */

    /* ---- Graphics output setting transfer ---- */
    VDC2_GR1.GRCMEN.BIT._WE = 1;/* Enables to transfer the register value */
}

/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Enable to display graphics block 2
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer2_enable(void);
 *-----------------------------------------------------------------------------
 * Function    : Enables to display graphics block 2.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer2_enable(void)
{
    VDC2_GR2.GRCMEN.BIT._DEN = 1;   /* Enables the current layer */
    VDC2_GR2.GRCMEN.BIT._VEN = 1;   /* Enables the lower layer */

    /* ---- Graphics output setting transfer ---- */
    VDC2_GR2.GRCMEN.BIT._WE = 1;/* Enables to transfer the register value */
}
```

3.12 Listings of Sample Program "Enables to display graphics block 3 and 4"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Enable to display graphics block 3
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer3_enable(void);
 *-----------------------------------------------------------------------------
 * Function    : Enables to display graphics block 3.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer3_enable(void)
{
    VDC2_GR3.GRCMEN.BIT._DEN = 1;   /* Enables the current layer */
    VDC2_GR3.GRCMEN.BIT._VEN = 1;   /* Enables the lower layer *

    /* ---- Graphics output setting transfer ---- */
    VDC2_GR3.GRCMEN.BIT._WE = 1;/* Enables to transfer the register value */
}

/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Enable to display graphics block 4
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer4_enable(void);
 *-----------------------------------------------------------------------------
 * Function    : Enables to display graphics block 4.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer4_enable(void)
{
    VDC2_GR4.GRCMEN.BIT._DEN = 1; /* Enables the current layer */
    VDC2_GR4.GRCMEN.BIT._VEN = 1; /* Enables the lower layer */

    /* ---- Graphics output setting transfer ---- */
    VDC2_GR4.GRCMEN.BIT._WE = 1; /* Enables to transfer the register value */
}
```

3.13 Listings of Sample Program "chroma-key setting and alpha blending setting"

```
/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Graphics block 3 chroma-key setting
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer4_chromakey(void);
 *-----------------------------------------------------------------------------
 * Function    : Controls chroma-keying for graphics block 3.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer3_chromakey(void)
{
    VDC2_GR3.GROPCRKY1.BIT._ALPHA = 0;             /* Alpha value after replacing */
    VDC2_GR3.GROPCRKY0.LONG = GRAPHICS3_CRKY_COLOR;    /* Specified color */
    VDC2_GR3.GROPCRKY0.BIT._CKEN = 1;              /* Enables chroma-keying */
    VDC2_GR3.GROPEDPA.BIT._WE = 1;                 /* Enables to transfer the register value
*/
}

/*""FUNC COMMENT""*****************************************************************
 * ID          :
 * Outline     : Graphics block 4 alpha blending setting
 *-----------------------------------------------------------------------------
 * Include     :
 *-----------------------------------------------------------------------------
 * Declaration : void vdc2_graphic_layer4_alpha(void);
 *-----------------------------------------------------------------------------
 * Function    : Controls alpha blending for graphics block 4.
 *-----------------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void vdc2_graphic_layer4_alpha(void)
{
    /* ---- Alpha control area setting ---- */
    /* Height and width of the alpha control area */
    VDC2_GR4.GROPEWH.BIT._GROPEH = GRAPHICS4_ALPHA_Y_SIZE;
    VDC2_GR4.GROPEWH.BIT._GROPEW = GRAPHICS4_ALPHA_X_SIZE;

    /* ---- Vertical and horizontal display start positions of the alpha control area ---- */
    VDC2_GR4.GROPEDPHV.BIT._GROPEDPV = TFT_DTMG_START_V - 1 + GRAPHICS4_ALPHA_POS_Y;
    VDC2_GR4.GROPEDPHV.BIT._GROPEDPH = TFT_DTMG_START_H - 16 + GRAPHICS4_ALPHA_POS_X;

    VDC2_GR4.GRCMEN.BIT._WE = 1;/* Enables to transfer the register value */
    /* ---- Alpha control data setting ---- */
    VDC2_GR4.GROPEDPA.BIT._DEFA  = 0xff; /* Initial alpha value */
    VDC2_GR4.GROPEDPA.BIT._ACOEF = 5;    /* Alpha value calculation coefficient   */
    VDC2_GR4.GROPEDPA.BIT._ARATE = 1;    /* Frame rate */
    VDC2_GR4.GROPEDPA.BIT._AMOD  = 2;    /* Subtracts from alpha value */
    VDC2_GR4.GROPEDPA.BIT._AEN   = 1;    /* Enables or disables to control alpha value */

    VDC2_GR4.GROPEDPA.BIT._WE = 1;  /* Enables to transfer the register value */
}
```

4. Documents for Reference

- Hardware Manual
  SH7764 Group Hardware Manual (REJ09B0360)
  The most up-to-date version of this document is available on the Renesas Technology Website.

- Software Manual (REJ09B0003)
  SH-4A Software Manual
  The most up-to-date version of this document is available on the Renesas Technology Website.

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Sep.28.09	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.