SH7262/SH7264 Group

Video Display Controller 3,

How to Use the α (Alpha) Blending Window Function

R01AN0613EJ0103
Rev. 1.03
Mar. 23, 2011

Summary

This application note describes the α blending window function of the SH7262/SH7264 Group Microcomputers (MCUs) Video Display Controller 3 (VDC3).

Target Device

SH7264 MCU

Contents

1. Introduction

1.1 Specifications

The α blending window function of the SH7264 MCU Video Display Controller (VDC3) overlays the input video and graphics image (two images) into a single combined image to display on TFT-LCD panel.

1.2 Modules Used

- Video Display Controller 3 (VDC3)
- General-purpose I/O Ports

1.3 Applicable Conditions

MCU	SH7262/SH7264
Operating Frequency	Internal clock: 144 MHz Bus clock: 72 MHz Peripheral clock: 36 MHz
Integrated Development Environment	Renesas Electronics Corporation High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Electronics SuperH RISC engine Family C/C++ compiler package Ver.9.03 Release 00
Compiler Options	Default setting in the High-performance Embedded Workshop (-cpu=sh2afpu -fpu=single -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

1.4 Related Application Note

Refer to the related application notes as follows:

- SH7262/SH7264 Group Example of Initialization
- SH7262/SH7264 Group Video Display Controller 3 TFT-LCD Interfacing Example
- SH/7262/SH7264 Group Video Display Controller 3 Video Display Example
- SH/7262/SH7264 Group Video Display Controller 3 Video Recording Example

1.5 About Active-low Pins (Signals)

The symbol "#" suffixed to the pin (or signal) names indicates that the pins (or signals) are active-low.

2. Applications

This application note shows the pin connection example and configuration example to use the VDC3 α blending window function.

2.1 VDC3 Operation

This section describes an overview of the VDC3 and α blending window function.

2.1.1 Overview

The VDC3 provides the following four functions. The video display function and video recording function cannot be used at the same time.

1. Video display function: Reduces the size of the input video, buffers the resultant video data in memory, and then displays the video on the panel
2. Video recording function: Stores a specified number of fields of the input video in SDRAM
3. Function for overlaying graphics images (two planes) on the input video
4. Function for outputting the control signals for the TFT-LCD panel

2.1.2 Features

The following table lists the VDC3 features.

Table 1 VDC3 Features

Item	Description
Operating frequency	Video input clock: 27 MHz Panel clock: 4 to 36 MHz (depends on the panel specifications)
Input video standard	8-bit input compliant to the ITU-R BT.656 standard (27 MHz) 8-bit serial input compliant to the ITU-R BT.601 standard (27 MHz)
Video recording function	Stores the video data in the RGB565 format at a rate of the 1/2 field (NTSC: 30 fps, PAL: 25 fps)
Video quality adjustment function	Contrast adjustment and brightness adjustment
Video scaling processing	Vertical: x 1/2, x 1/3, x 1/4 Horizontal: x 2/3, x 1/2, x 1/3, x 1/4 Each scaled value can be further multiplied by 6/7 to support PAL.
Graphics images	Two planes (layers 1 and 2) RGB565 progressive format (α: none, R: 5 bits, G: 6 bits, B: 5 bits; 16 bits in total) αRGB4444 progressive format (α: 4 bits, R: 4 bits, G: 4 bits, B: 4 bits; 16 bits in total)
Graphics functions	• α blending window function: Mixes the input video and layers 1 and 2 according to the transparency rate α in the specified region (fade-in and fade-out functions are available) • Chroma-keying function: Mixes the images with applying the specified RGB color according to the transparency rate α • Dot α function: Mixes the images according to the transparency rate α when the target image is in α RGB4444 • For each dot, the priority among the α values of the above functions is as follows: α blending window > chroma-keying > dot α
Output video size	640 pixels x 480 lines (VGA size) 480 pixels x 240 lines (WQVGA size) 320 pixels x 240 lines (QVGA, landscape-mode) 240 pixels x 320 lines (QVGA, portrait-mode) Note: The maximum viewable area for the input image is 480 pixels x 240 lines (NTSC), and 480 pixels x 288 lines (PAL).
Output video format	RGB565 progressive video output (16-bit parallel output)
Sync signal output	Outputs the control signals for the TFT-LCD panel
Interrupt output	Line interrupt output (this can be output on a desired line) VSYNC cycle fluctuation detection signal for the BT.601 and BT.656 Field write completion signal Overflow/underflow detection signal for the internal buffer

2.1.3 I/O Pins

The following table lists the VDC3 I/O pins.

Table 2 VDC3 I/O Pins

Symbol	I/O	Pin Name	Description
DV_CLK	Input	Video input clock	BT.601, BT.656 clock input pin
DV_VSYNC	Input	VSYNC input	BT.601 VSYNC signal input pin
DV_HSYNC	Input	HSYNC input	BT.601 HSYNC signal input pin
DV_DATA7 to 0	Input	BT.601 or BT.656 input	BT.601 or BT.656 data signal input pins
LCD_CLK	Output	Panel clock	Panel clock output pin
LCD_EXTCLK	Input	Panel clock source	Panel clock source input pin
LCD_VSYNC	Output	Panel VSYNC output	Vertical sync signal output pin for the panel
LCD_HSYNC	Output	Panel HSYNC output	Horizontal sync signal output pin for the panel
LCD_DE	Output	Panel data enable output	Data enable signal or data start position pulse signal output pin for the panel
LCD_DATA15 to 0	Output	Panel data output	Data output pins for the panel MSB LSB MSB LSB [15 : 11]: Red [4 : 0] [10 : 5]: Green[5 : 0] [4 : 0]: Blue [4 : 0]
LCD_M_DISP	Output	Panel control signal	Alternating signal for the panel

2.1.4 Configuration

The figure below shows the VDC3 block diagram. Refer to Table 3 for each block.

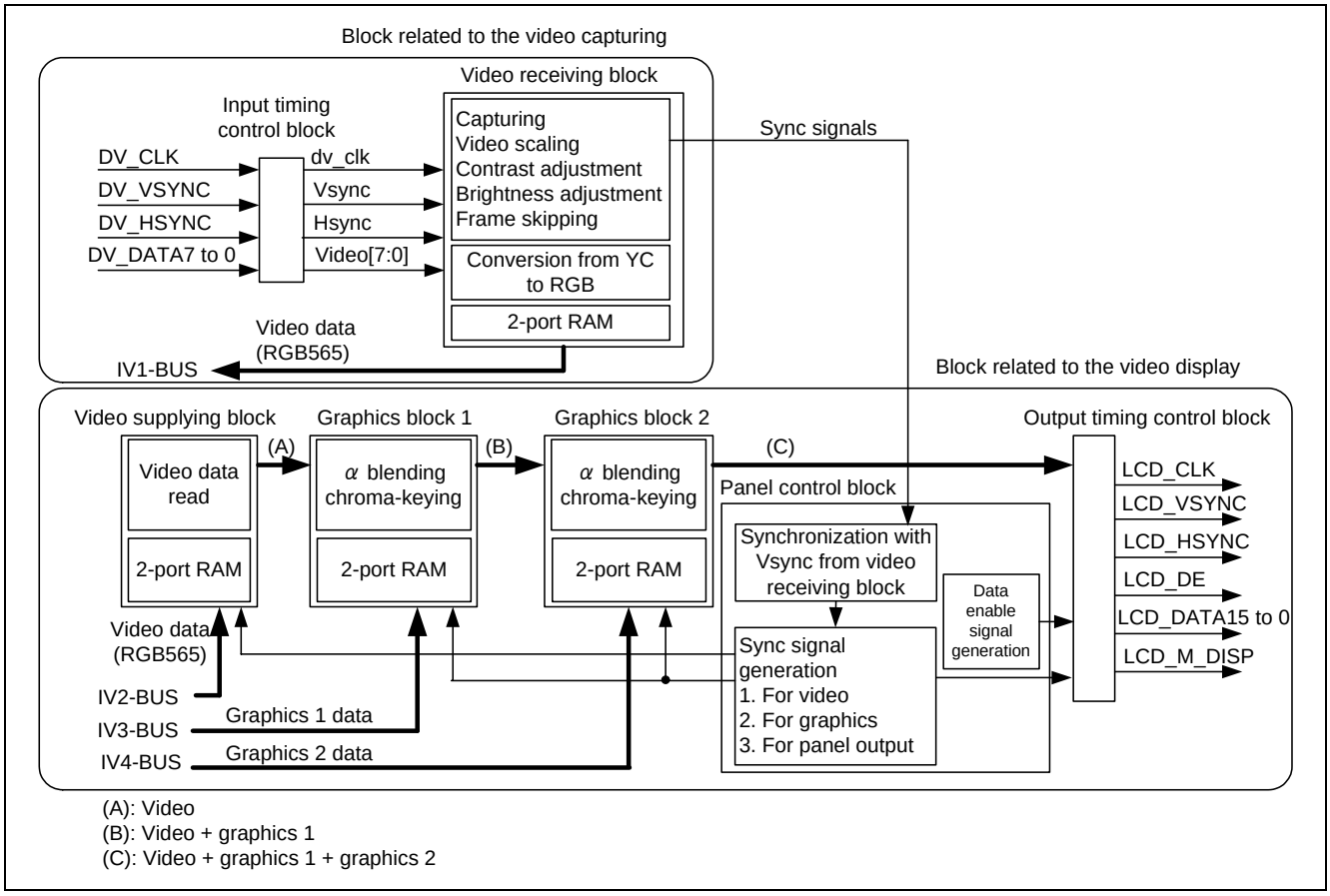


Figure 1 VDC3 Block Diagram

Table 3 VDC3 Functional Blocks

Block Name	Overview
Input timing control block	Controls the timing of the input sync signal clock rising or falling edge, and the sync polarity. It also controls the timing of the BT.601 and BT.656 video input signals clock rising or falling edge.
Video receiving block	(1) Captures the input video, scales, adjusts the contrast, and the brightness. (2) Converts the YC format to the RGB565 format, and stores the data via the IV1-BUS. (3) Skips the field, and stores the data in the RGB565 format via the IV1-BUS.
Video supplying block	Reads the video data via the IV2-BUS.
Graphics block 1	Reads a graphics image (layer 1) from the memory via the IV3-BUS, overlays it on the video sent from the video supplying block, and outputs the result to graphics block 2.
Graphics block 2	Reads a graphics image (layer 2) from the memory via the IV4-BUS, overlays it on the output from graphics block 1, and outputs the result to the output timing control block.
Panel control block	Generates the sync signals to output to the panel
Output timing control block	Controls the timing of the output sync signal clock rising-edge or falling-edge, and the sync signal polarity. It also controls the timing of the RGB565 video output signals clock rising or falling edge.

2.1.5 VDC3 Graphics Function

This section describes the VDC3 graphics functions, α blending window function, Chroma-keying function and Dot α function. Each function has a single alpha value, the transparency rate. The current layer and the lower layer can be mixed by controlling the α value.

The priority of alpha values among these functions is in the order of the α blending window function, Chroma-keying function, and Dot α function. When either one of these functions is enabled, its α value with the highest-priority is only valid; other alpha values are invalid.

(1) α blending window function

- Combines the current layer and lower layer located in alpha-controlled area by controlling the α value
- Synchronizes with the field rate of the input video to add to or subtract from α value (fade-in, fade-out)

The figure below shows an image of the α blending window function. Table 4 lists the blending ratio by the α value of the α blending window function.

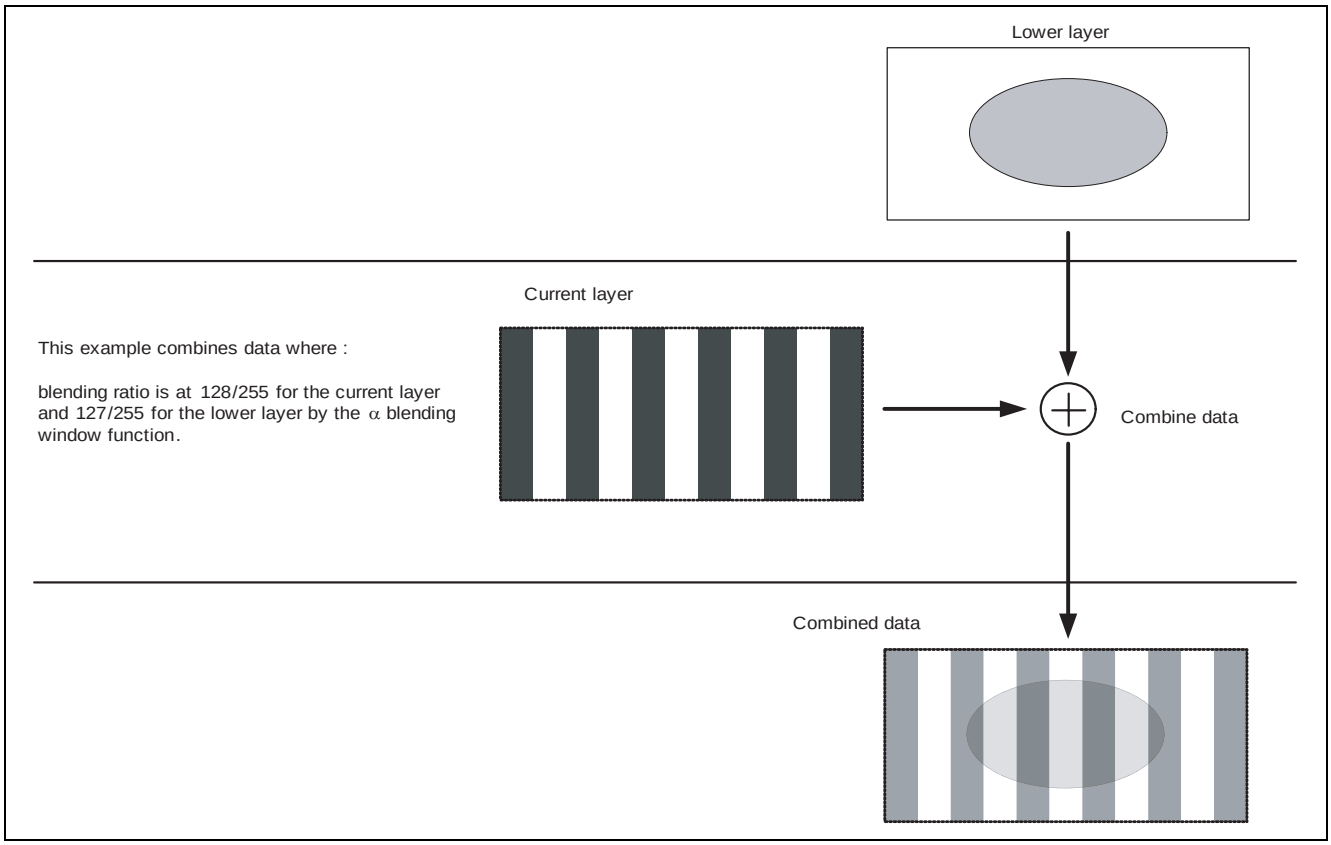


Figure 2 α Blending Window Function Image

Table 4 Blending Ratio by the α Value of the α Blending Window Function

α value	Current layer	Lower layer	Percentage of each layer to display
255	255/255	0/255	• Current layer: 100 (%) • Lower layer: 0 (%)
254	254/255	1/255	• Current layer: 254/255 x 100 (%) • Lower layer: 1/255 x 100 (%)
253	253/255	2/255	• Current layer: 253/255 x 100 (%) • Lower layer: 2/255 x 100 (%)
…	…	…	…
1	1/255	254/255	• Current layer: 1/255 x 100 (%) • Lower layer: 254/255 x 100 (%)
0	0/255	255/255	• Current layer: 0 (%) • Lower layer: 100 (%)

Note: "··· (ellipsis dots) " indicate the parameter is omitted.

(2) Chroma-keying function

If any pixel on the graphics image that has the same color as the color specified by the chroma-keying, following operations are processed;

- Replaces the color according to the color information set in the Chroma-Key color register
- After replacing the color, ALPHA [7:0] bits in the Chroma-Key color register controls the α value

The figure below shows an image of the Chroma-Keying function. Table 5 lists the blending ratio by the α value of the Chroma-keying function.

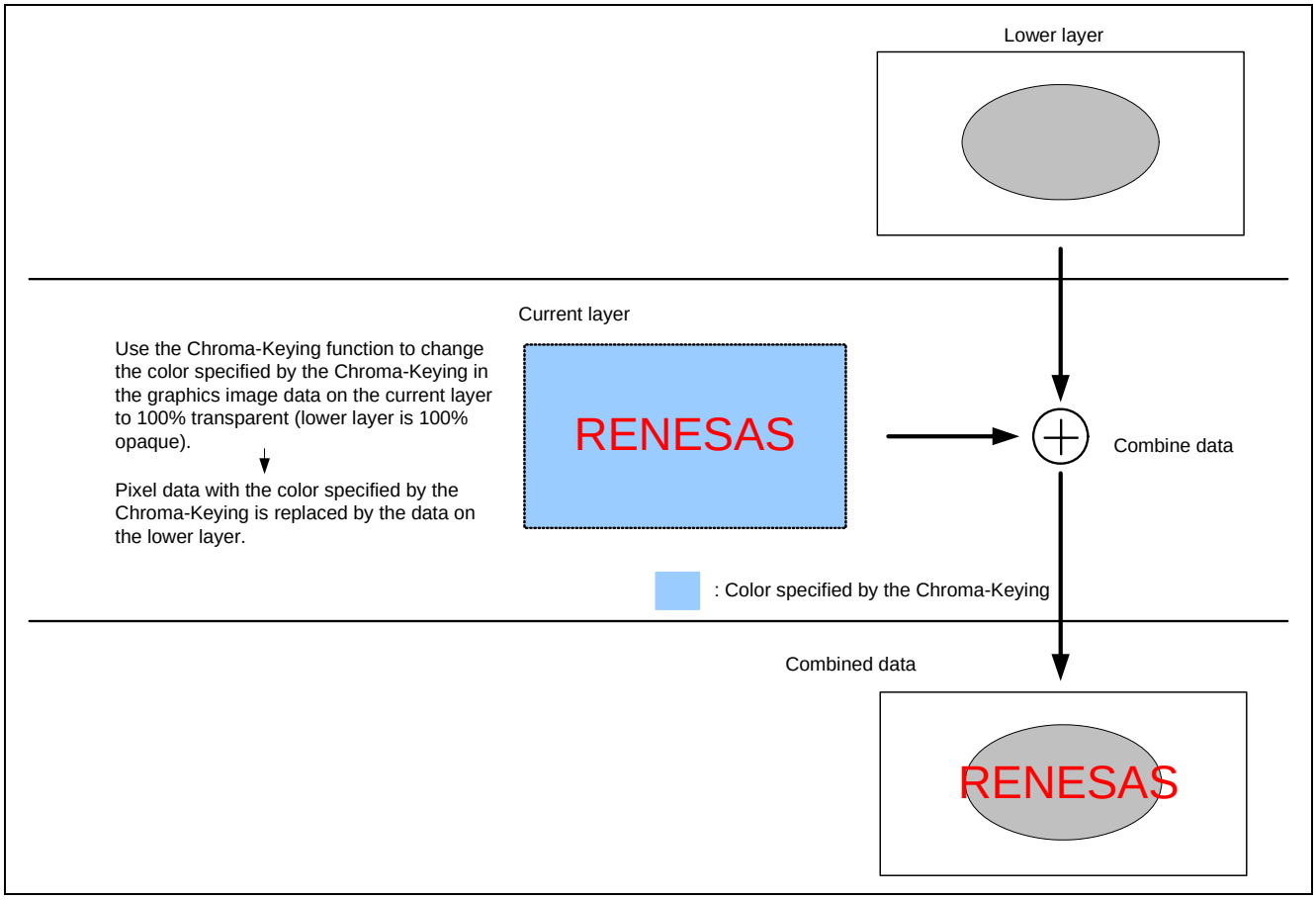


Figure 3 Chroma-Keying Function Image

Table 5 Blending Ratio by the α Value of the Chroma-Keying Function

α value	Current layer	Lower layer	Percentage of each layer to display
255	255/255	0/255	• Current layer: 100 (%) • Lower layer: 0(%)
254	254/255	1/255	• Current layer: 254/255 x 100(%) • Lower layer: 1/255 x 100 (%)
253	253/255	2/255	• Current layer: 253/255 x 100(%) • Lower layer: 2/255 x 100(%)
…	…	…	…
1	1/255	254/255	• Current layer: 1/255 x 100 (%) • Lower layer: 254/255 x 100 (%)
0	0/255	255/255	• Current layer: 0 (%) • Lower layer: 100 (%)

Note: "··· (ellipsis dots) " indicate the parameter is omitted.

(3) Dot α Function

When selecting αRGB4444 as the graphics image format, combines the graphics image with the lower layer based on the transparency rate α. The figure below shows the αRGB4444 graphics image format. Use upper 4 bits of the graphics image data in the αRGB4444 format to set the α value. Table 6 lists the blending ratio by the α value of the Dot α function.

When handling the graphics data in the αRGB4444 format, specify the BUS_FORMAT bit in the Bus control register as 1.

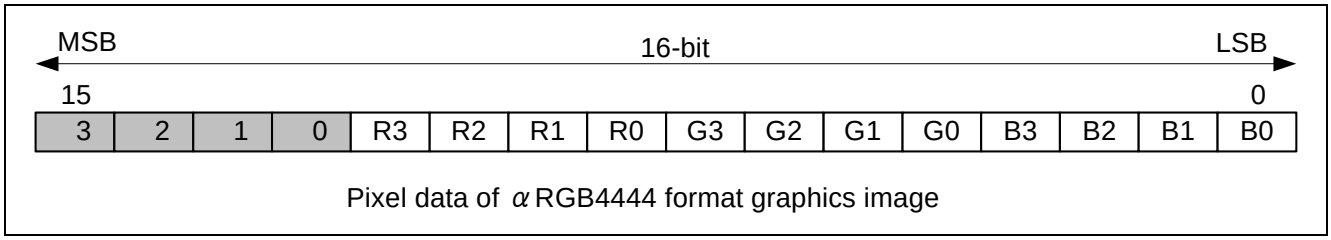


Figure 4 αRGB4444 Graphics Image Format

Table 6 Blending Ratio by the α Value of the Dot α Function

α value	Current layer	Lower layer	Percentage of each layer to display
15	15/15	0/15	• Current layer: 100 (%) • Lower layer: 0(%)
14	14/15	1/15	• Current layer: 14/15 x 100 (%) • Lower layer: 1/15 x 100 (%)
13 … 2	13/15 … 2/15	2/15 … 13/15	…
1	1/15	14/15	• Current layer: 1/15 x 100 (%) • Lower layer: 14/15 x 100 (%)
0	0/15	15/15	• Current layer: 0 (%) • Lower layer: 100 (%)

Note: "··· (ellipsis dots) " indicate the parameter is omitted.

2.2 Application Circuit with the SH7264 MCU

This application combines the input image and the graphics image by the α blending window function, and outputs the combined image on the TFT-LCD panel. Figure 5 shows the circuit diagram between the SH7264 and the digital video decoder. Figure 6 shows the circuit diagram between the SH7264 and the TFT-LCD panel.

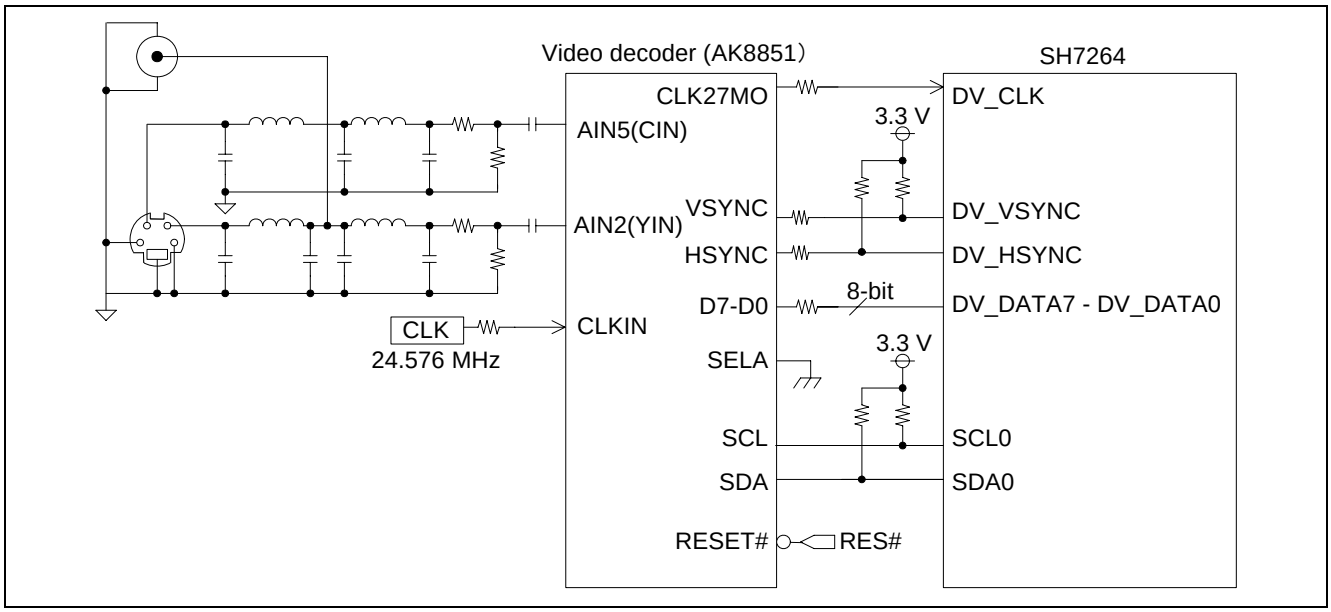


Figure 5 Digital Video Decoder Circuit

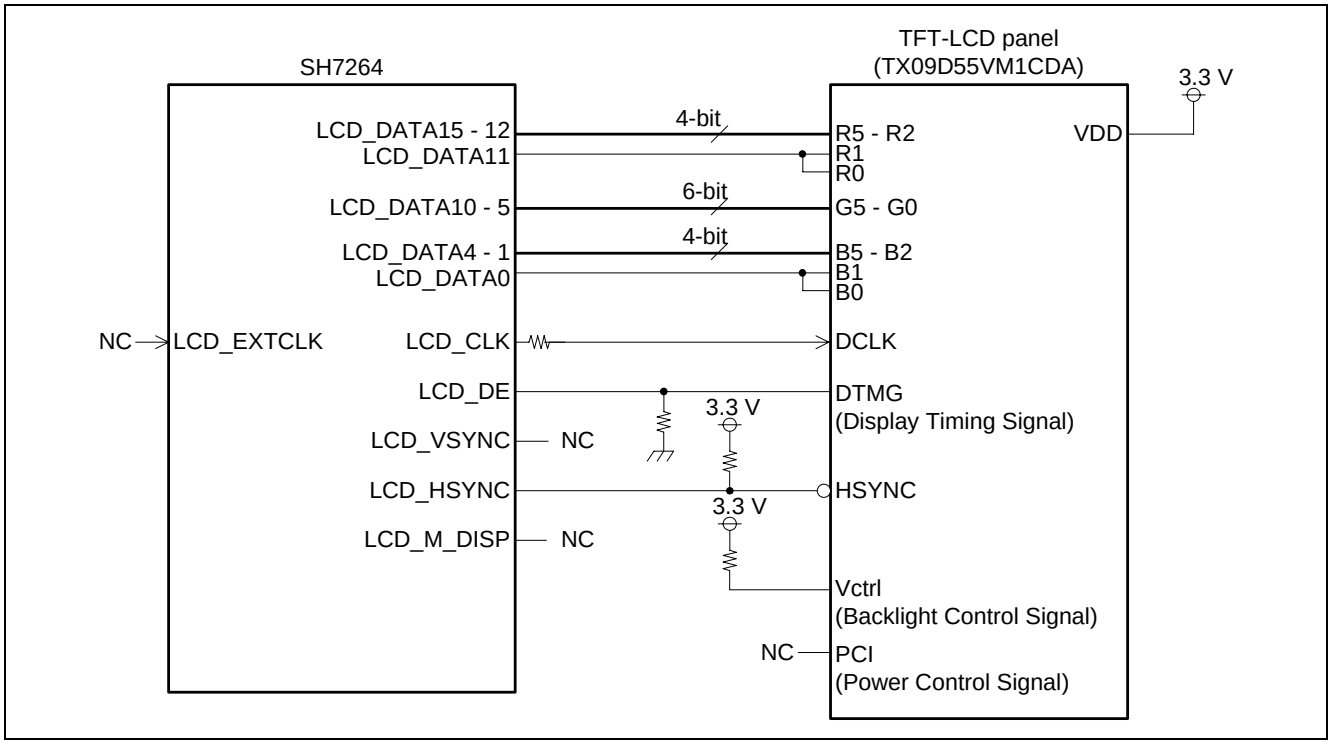


Figure 6 TFT-LCD Panel Circuit

2.3 Sample Program Specifications

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

2.3.1 Specifications

- Overlaps the video data, graphics image 1, and graphics image 2 by the α blending window function, and displays the combined image on the QVGA size (V 320 x H 240) TFT-LCD panel
- Inputs the BT.656 (NTSC) format video, scales down the video to 1/3 (horizontal), 1/3 (vertical), to output the video in the center of the panel in V 160 x H 240
- Graphics image 1 is 80 x 80, the start position to display is at (60, 100), in black
- Graphics image 2 is 80 x 80, the start position to display is at (100, 140), in white
- α control to the graphics image 1: Specifies the default α value as 255, subtracts 1 from the α value per frame rate to fade-in
- α control to the graphics image 2: Specifies the default α value as 0, adds 1 to the α value per frame rate to fade-out
- Chroma-keying function and Dot α function are disabled

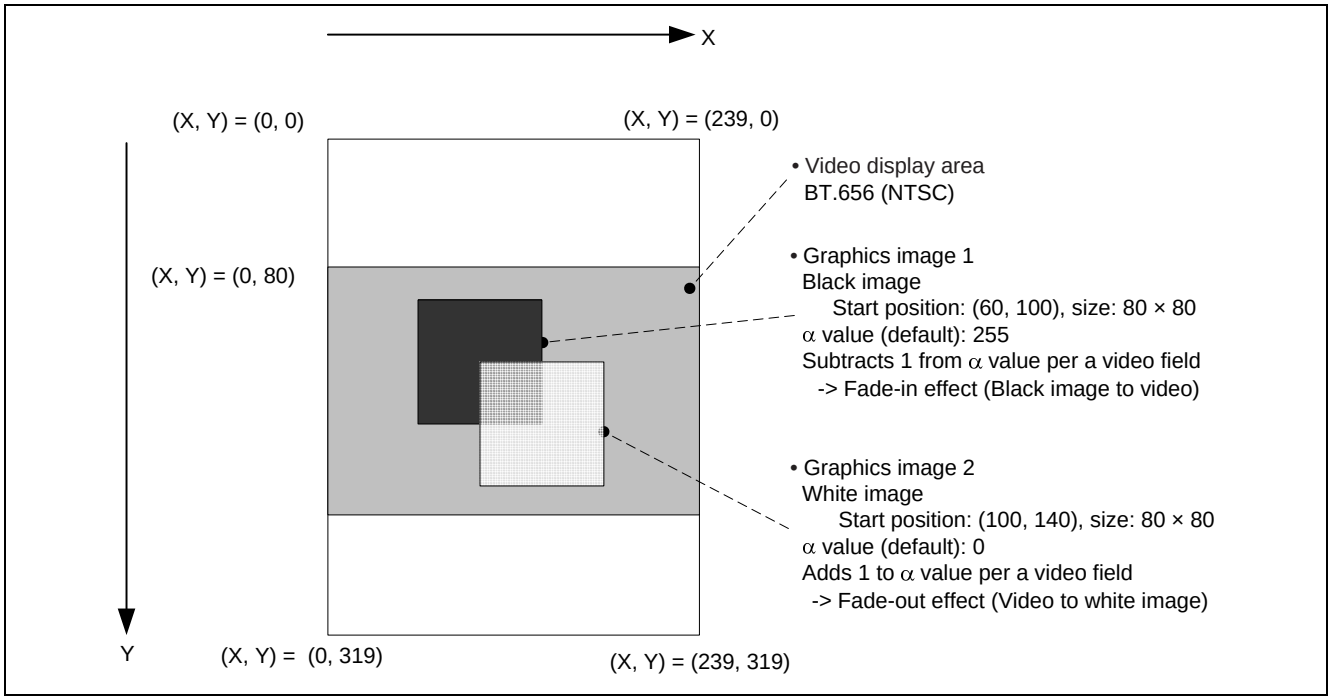


Figure 7 Sample Program Operation Image

2.3.2 Main Flow Chart of the Sample Program

The figure below shows the main flow chart of the sample program. The sample program executes a series of the initializing processing as shown in Figure 9 to Figure 17 to display the alpha-blended video and image on the TFT-LCD panel.

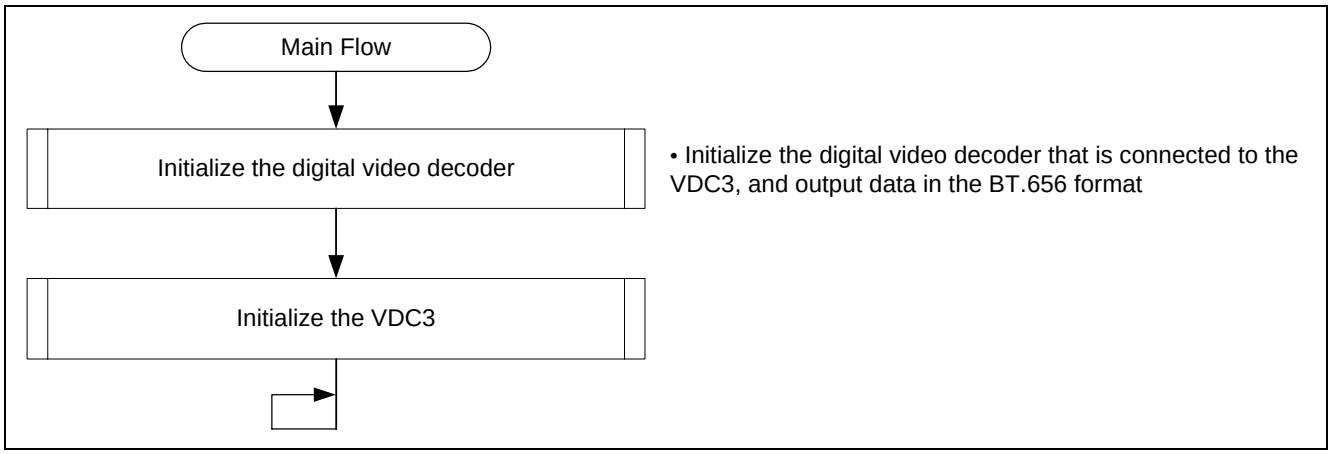


Figure 8 Sample Program Main Flow Chart

2.3.3 Initializing the VDC3

The following figure shows the procedure to initialize the VDC3.

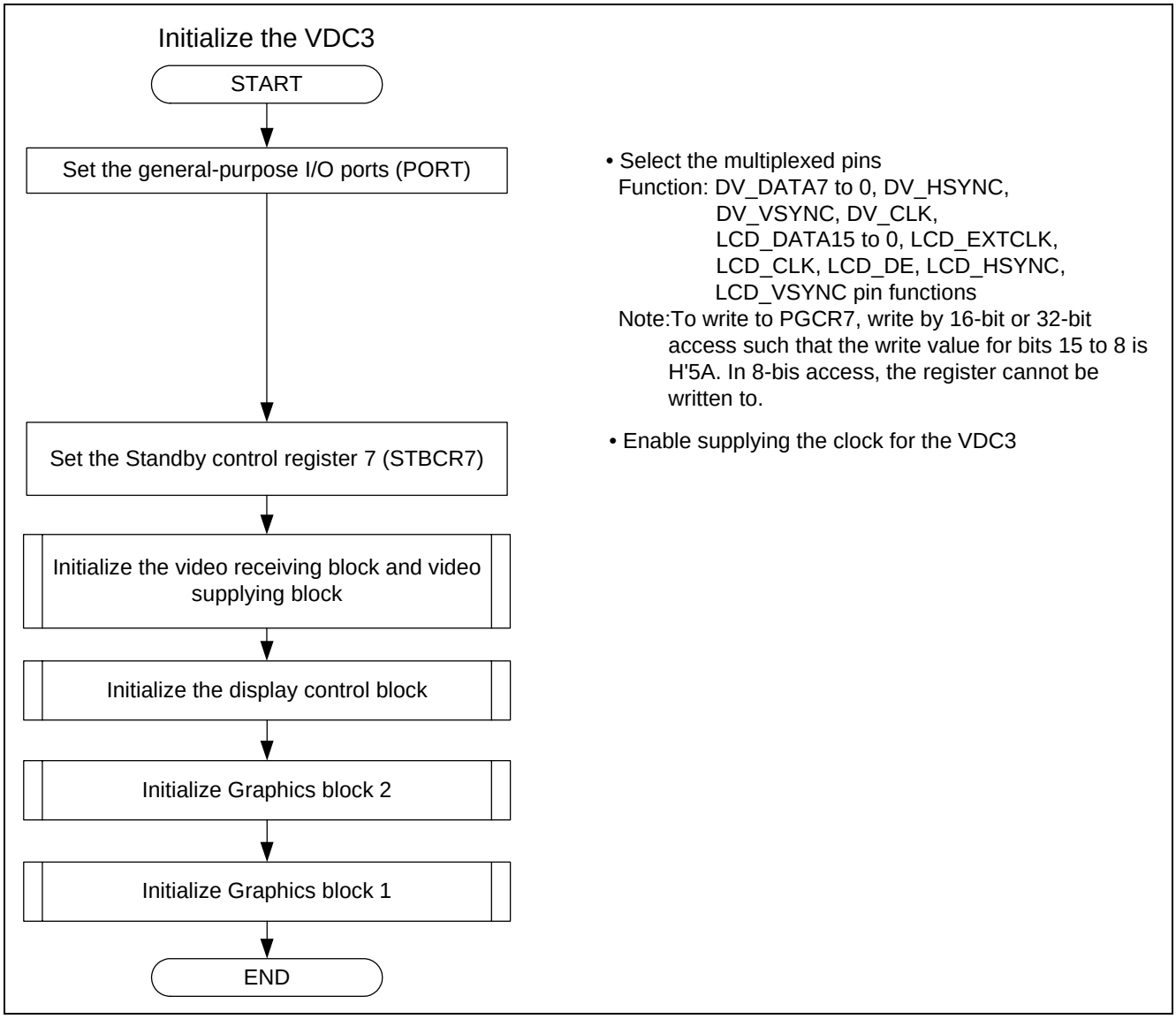
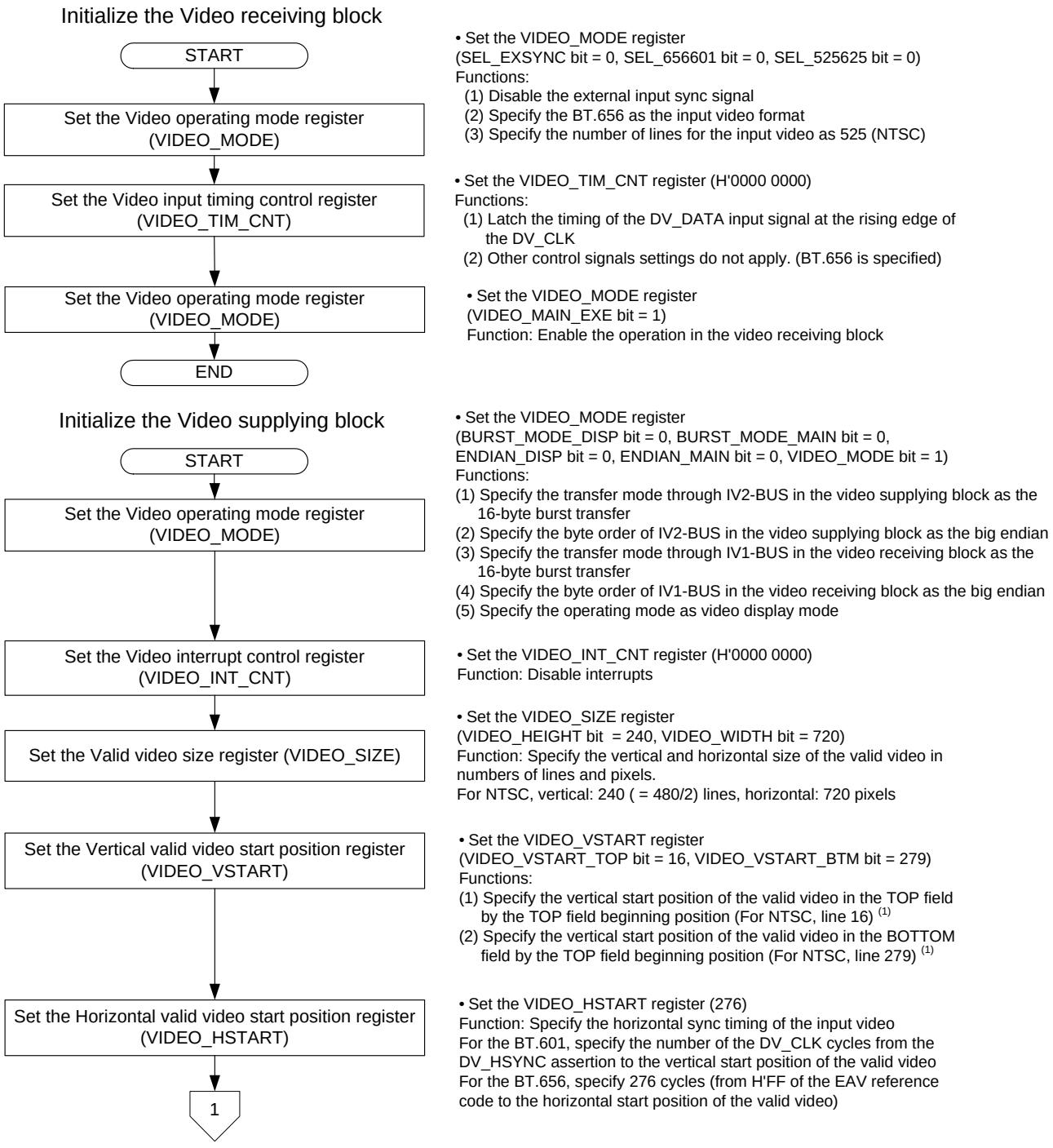


Figure 9 VDC3 Initialization Flow Chart

2.3.4 Initializing the Video Receiving Block and Video Supplying Block

Figure 10 and Figure 11 show the flow charts of initializing the video receiving block and video supplying block.



Note 1: As the number of lines of the valid video in the BT.656 (NTSC) are 244 for TOP and 243 for BOTTOM, 1 to 3 lines can be moved according to the size of the input video.

Figure 10 Initializing the Video Receiving Block and Video Supplying Block (1/2)

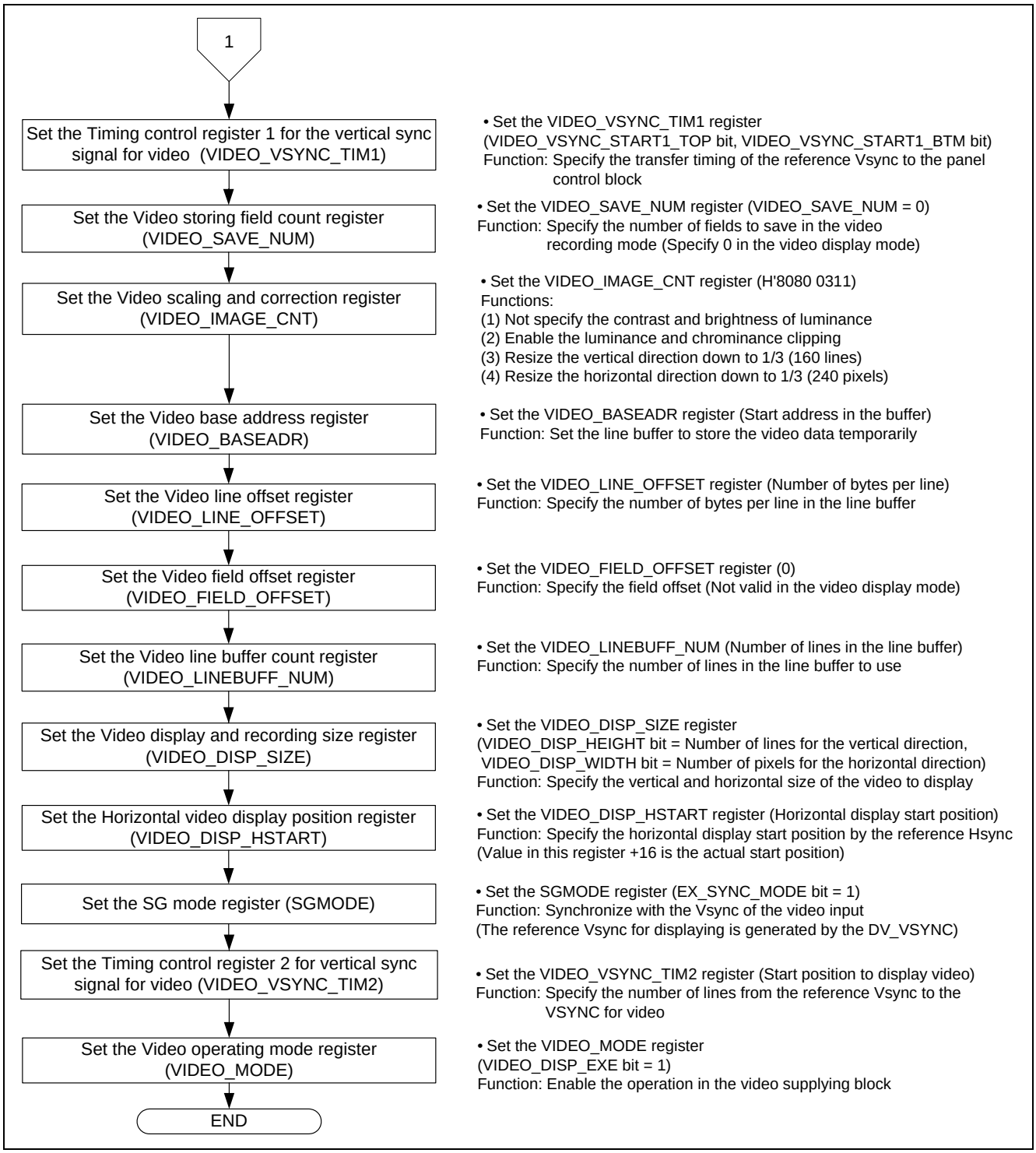


Figure 11 Initializing the Video Receiving Block and Video Supplying Block (2/2)

2.3.5 Initializing the Panel Control Block

Figure 12 and Figure 13 show examples of initializing the panel control block.

Values listed in Figure 12 and Figure 13 are set according to the specifications of the TFT-LCD panel used in this application. Alter the setting according to the type of the TFT-LCD panel.

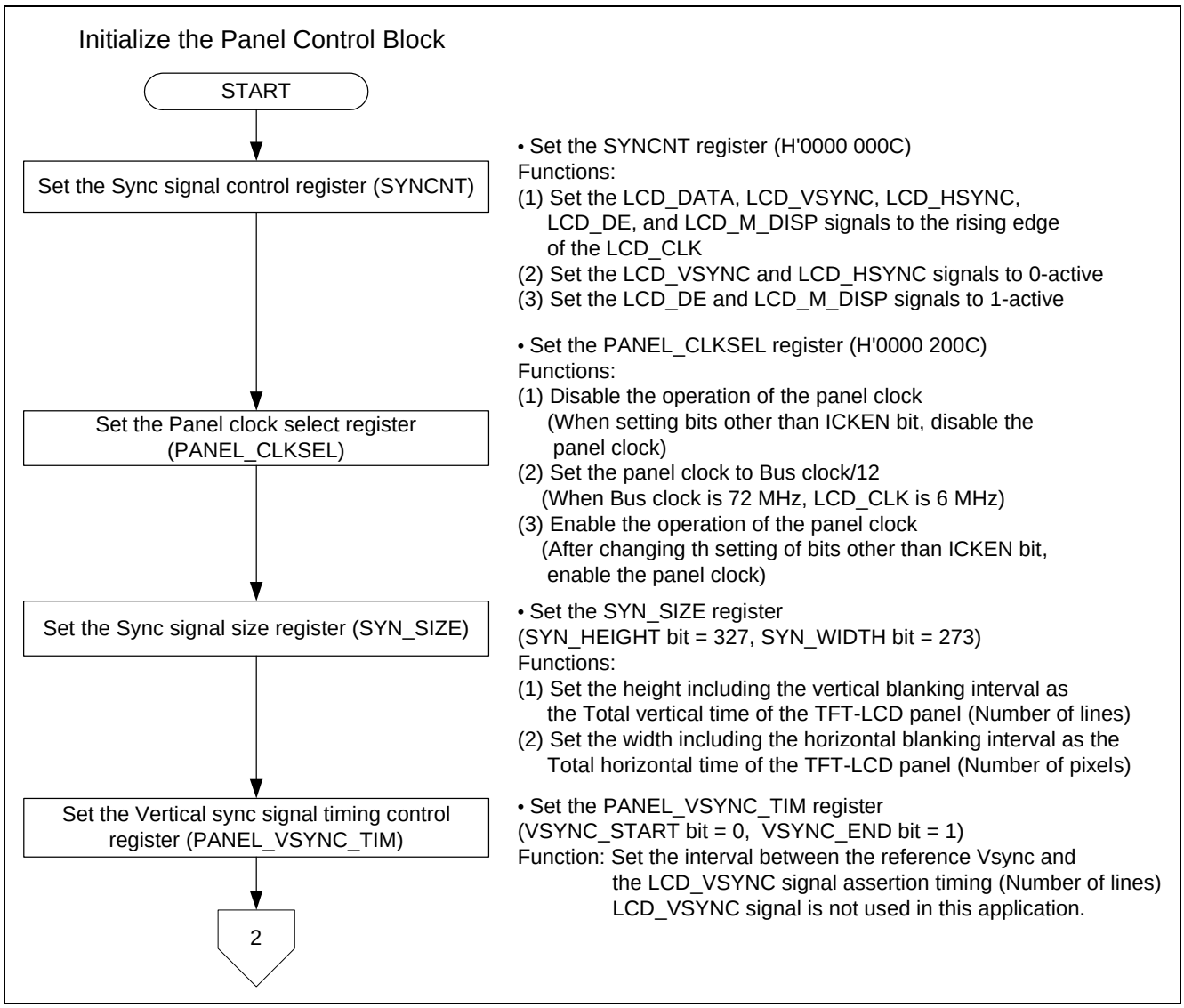


Figure 12 Initializing the Panel Control Block (1/2)

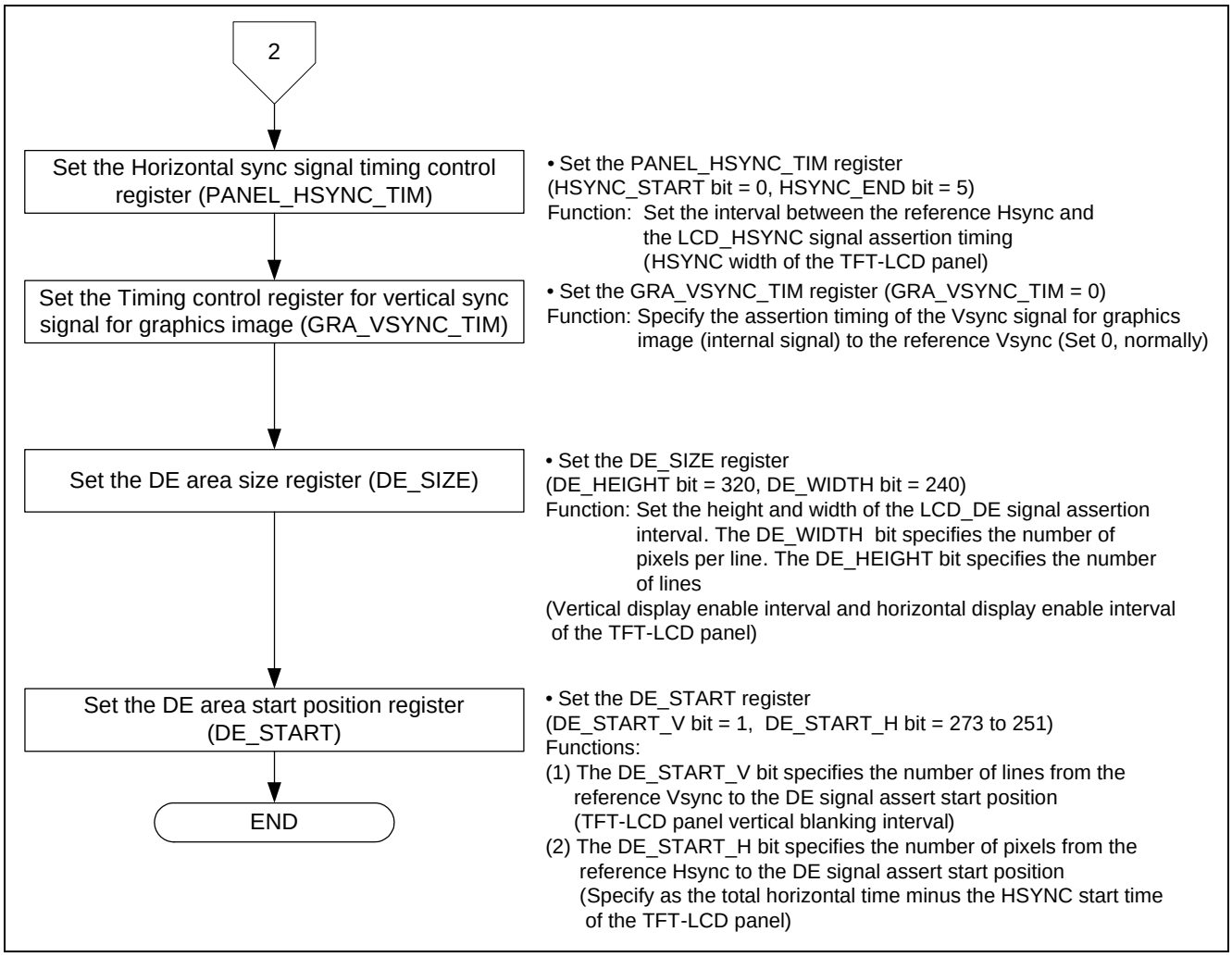


Figure 13 Initializing the Panel Control Block (2/2)

2.3.6 Initializing Graphics Block 1

Figure 14 and Figure 15 show examples of initializing Graphics block 1.

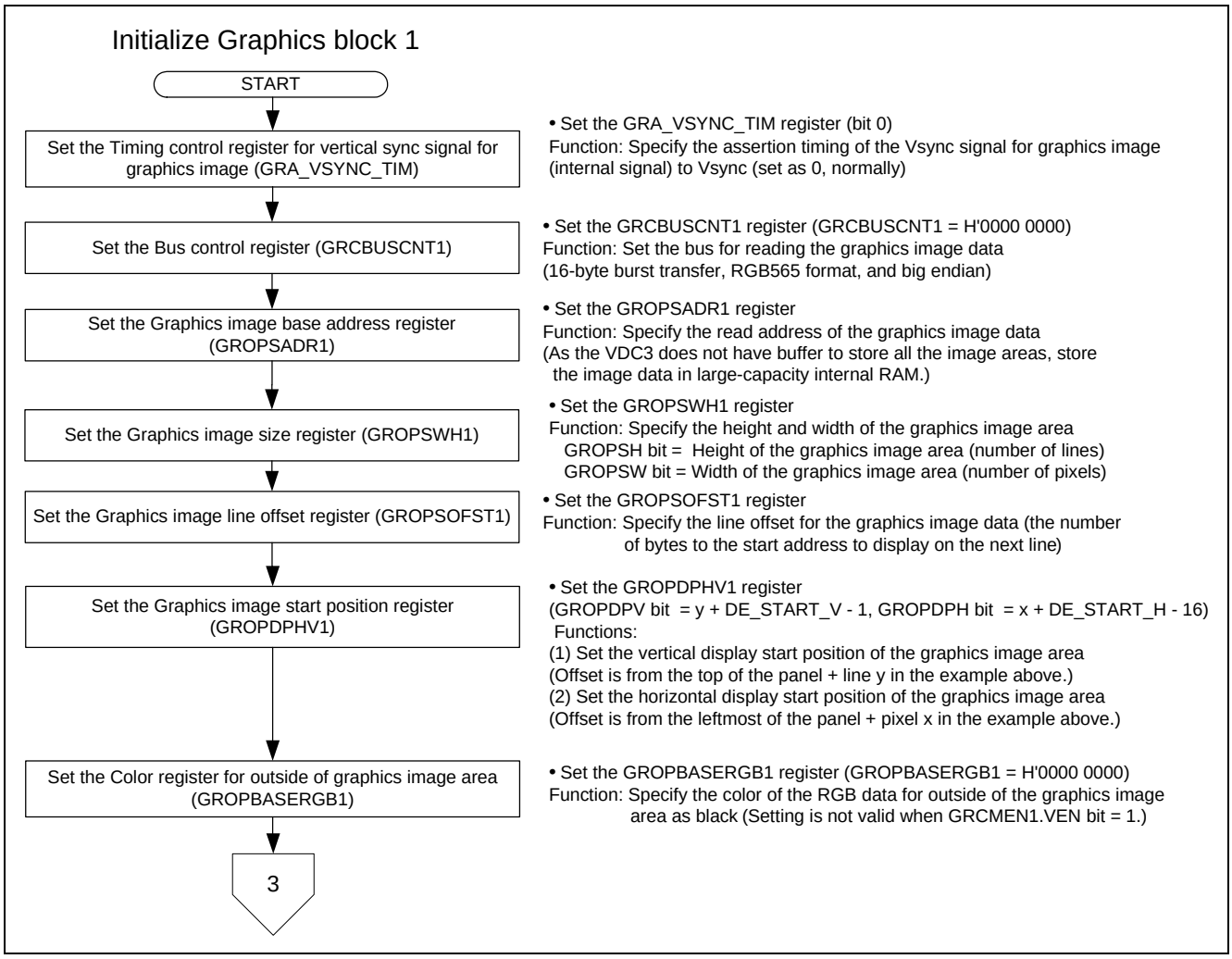


Figure 14 Initializing Graphics Block 1 (1/2)

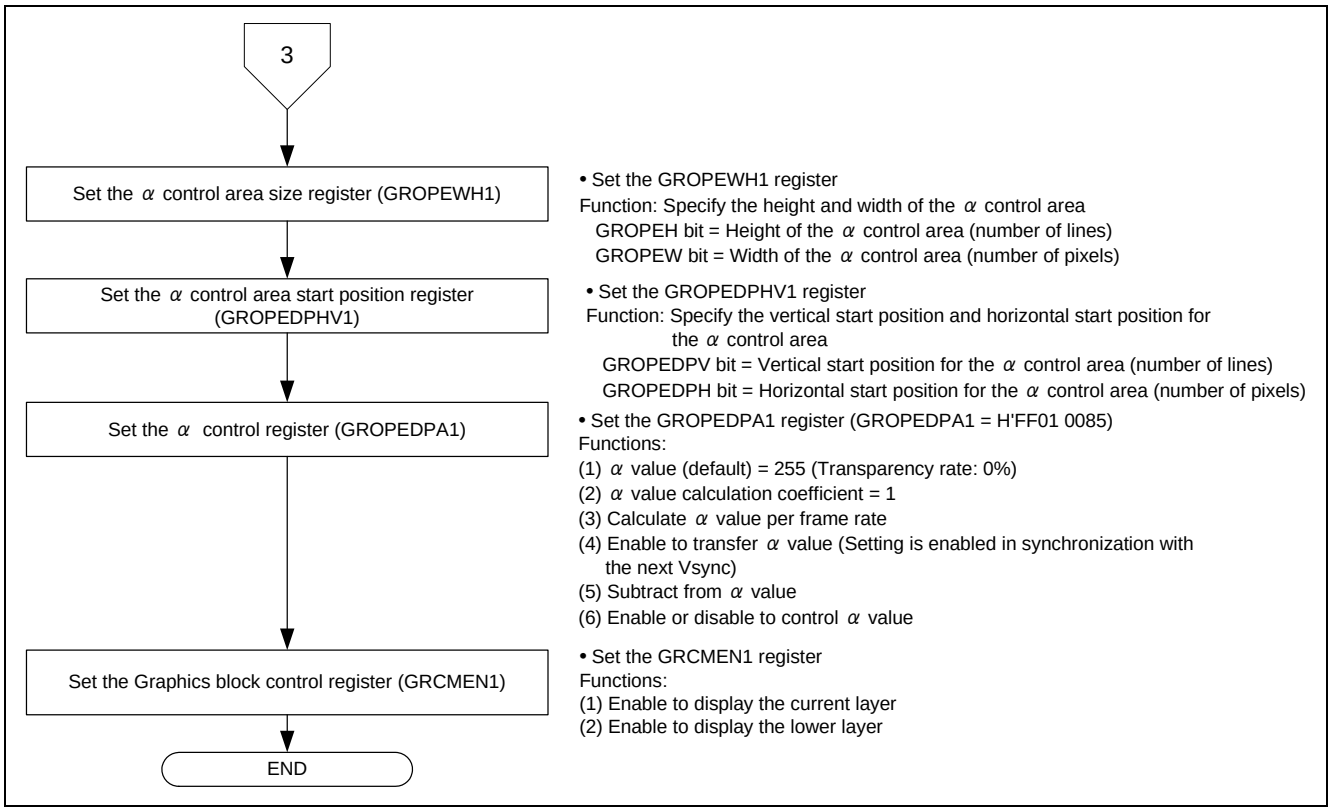


Figure 15 Initializing Graphics Block 1 (2/2)

2.3.7 Initializing Graphics block 2

Figure 16 and Figure 17 show examples of initializing Graphics block 2.

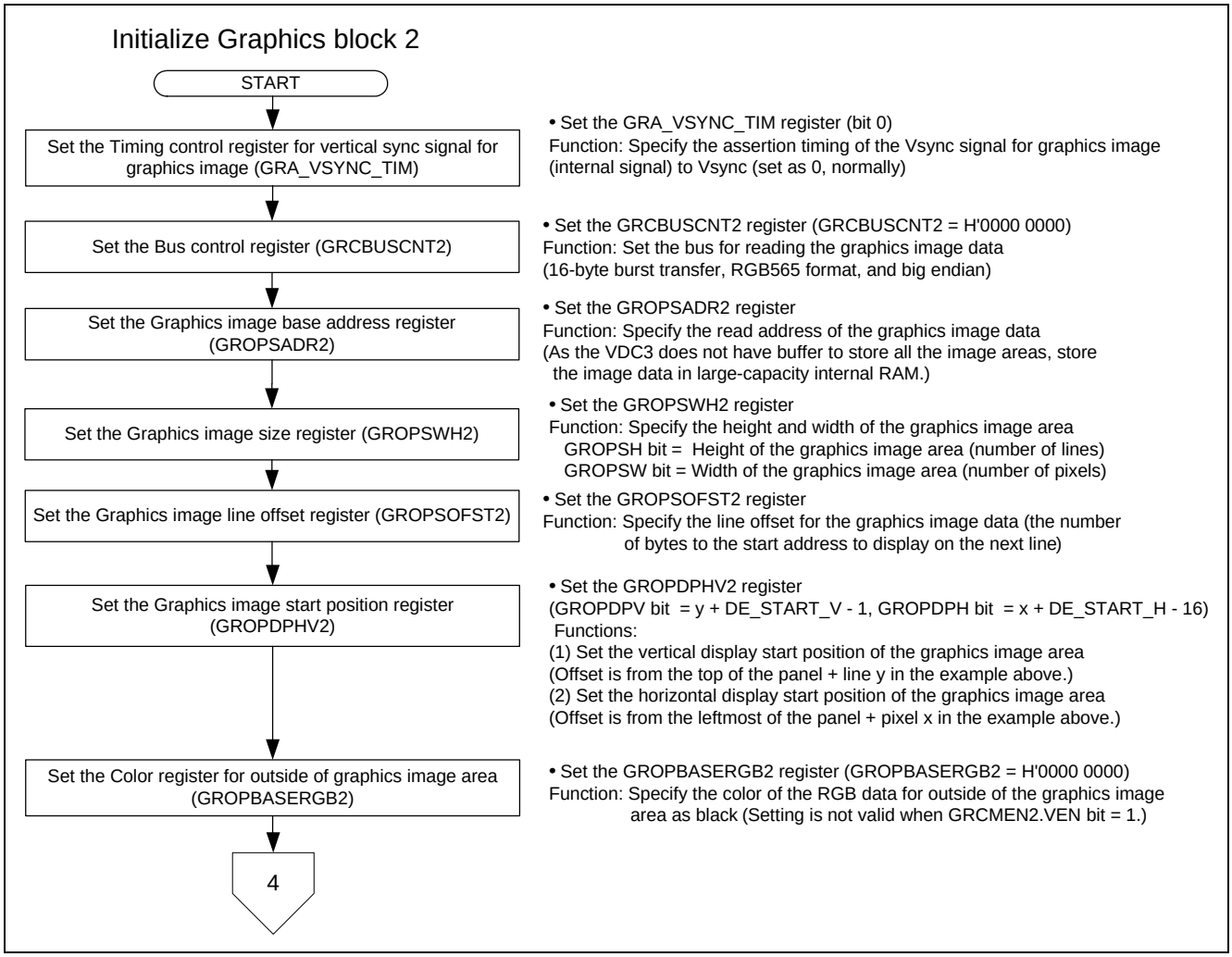


Figure 16 Initializing Graphics Block 2 (1/2)

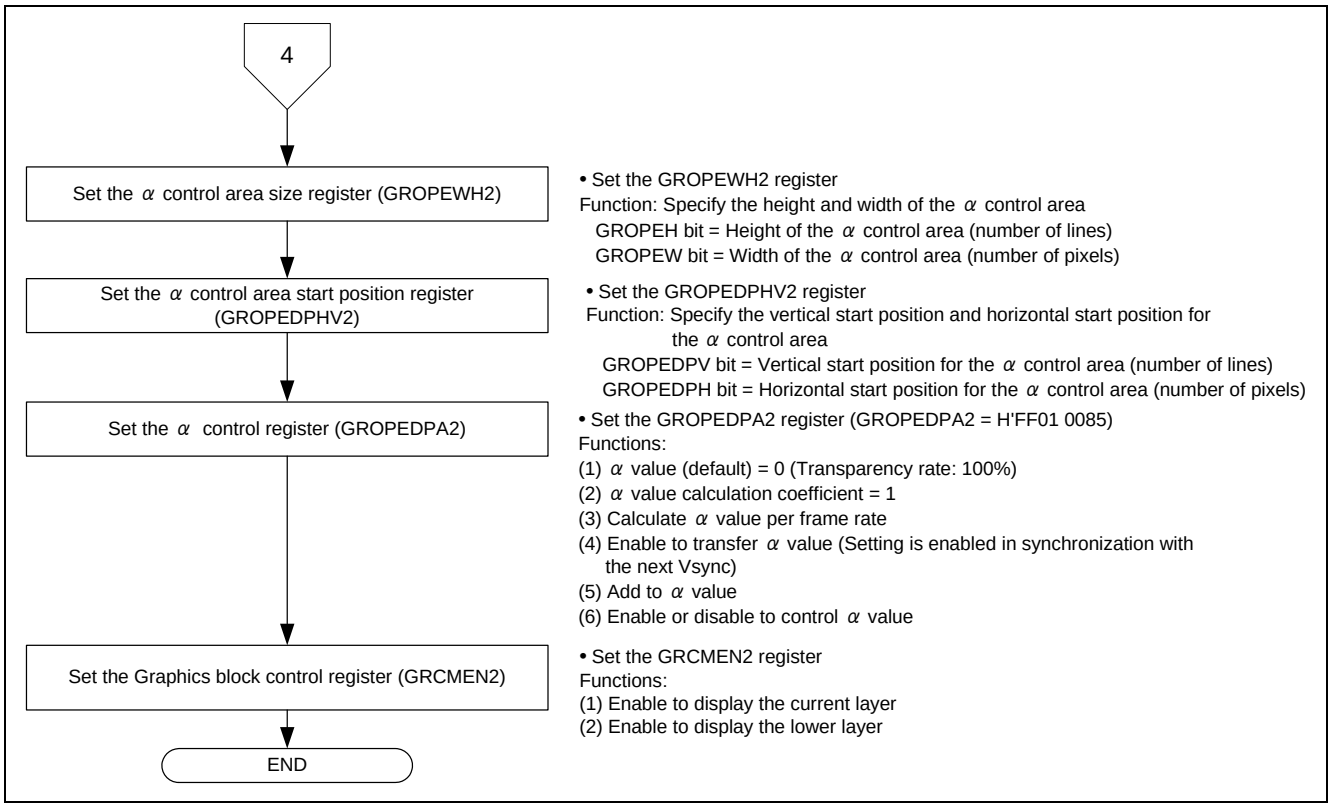


Figure 17 Initializing Graphics Block 2 (2/2)

3. Sample Program Listing

3.1 Supplement to the Sample Program

As the capacity of the SH7264 large-capacity internal RAM varies as 1 MB or 640 KB, depending on the MCU type, the section alignment and register setting must be partly altered. To support both MCU types, this application note provides two types of sample programs (workspaces) for 1-MB RAM and 640-KB RAM.

As the MCU with 640-KB RAM must be write-enabled before writing data in the data-retention RAM, the System control register 5 (SYSCR5) is set to write-enable the RAM in the sample program for 640-KB RAM.

Review your product and use the appropriate workspace.

3.2 Sample Program Listing "main.c" (1/2)

```
1       /*********************************************************************************
2       *   DISCLAIMER
3       *
4       *   This software is supplied by Renesas Electronics Corporation and is only
5       *   intended for use with Renesas products. No other uses are authorized.
6       *
7       *   This software is owned by Renesas Electronics Corporation and is protected under
8       *   all applicable laws, including copyright laws.
9       *
10      *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11      *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12      *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13      *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14      *   DISCLAIMED.
15      *
16      *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17      *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18      *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19      *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20      *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21      *
22      *   Renesas reserves the right, without notice, to make changes to this
23      *   software and to discontinue the availability of this software.
24      *   By using this software, you agree to the additional terms and
25      *   conditions found by accessing the following link:
26      *   http://www.renesas.com/disclaimer
27      *********************************************************************************/
28      /* Copyright (C) 2009(2010,2011) Renesas Electronics Corporation. All Rights Reserved.*/
29      /*********************************************************************************
30      *   System Name : SH7264 Sample Program
31      *   File Name   : main.c
32      *   Abstract    : VDC3, How to use the alpha-blending window function
33      *   Version     : 2.00.00
34      *   Device      : SH7264
35      *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36      *               : C/C++ compiler package for the SuperH RISC engine family
37      *               :                               (Ver.9.03 Release00).
38      *   OS          : None
39      *   H/W Platform: M3A-HS64G50(CPU board), M3A-HS64G02(Option board)
40      *   Description :
41      *********************************************************************************
42      *   History     : Mar.12,2009 Ver.1.00.00
43      *               : Jan.14,2010 Ver.1.01.00
44      *               : Feb.28,2011 Ver.2.00.00
45      *********************************************************************************/
46
47
```

3.3 Sample Program Listing "main.c" (2/2)

```
48    /***********************************************************************
49    Includes  <System Includes> , "Project Includes"
50    ***********************************************************************/
51    #include <stdio.h>
52    #include "io_vdc3_alpha_blend.h"
53
54    /***********************************************************************
55    Exported global variables and functions (to be accessed by other files)
56    ***********************************************************************/
57    /* ==== Global functions ==== */
58    void main(void);
59
60    /***********************************************************************
61     * ID           :
62     * Outline      : Alpha blending window function main
63     * Include      :
64     * Declaration  : void main(void);
65     * Description  : This function sets the alpha blending window function.
66     * Argument     : void
67     * Return Value : void
68     ***********************************************************************/
69    void main(void)
70    {
71      /* ==== Initializes the digital video decoder ==== */
72      init_video_decoder();
73
74      /* ==== Initializes the VDC3 ==== */
75      io_vdc3_init();
76
77      while(1){
78        /* loop */
79        }
80
81    }
82
83    /* End of File */
84
```

3.4 Sample Program Listing "io_vdc3_alpha_blend.c" (1/11)

```
1     /***********************************************************************
2     *   DISCLAIMER
3     *
4     *   This software is supplied by Renesas Electronics Corporation and is only
5     *   intended for use with Renesas products. No other uses are authorized.
6     *
7     *   This software is owned by Renesas Electronics Corporation and is protected under
8     *   all applicable laws, including copyright laws.
9     *
10    *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11    *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12    *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13    *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14    *   DISCLAIMED.
15    *
16    *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17    *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18    *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19    *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20    *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21    *
22    *   Renesas reserves the right, without notice, to make changes to this
23    *   software and to discontinue the availability of this software.
24    *   By using this software, you agree to the additional terms and
25    *   conditions found by accessing the following link:
26    *   http://www.renesas.com/disclaimer
27    ***********************************************************************/
28    /*   Copyright (C) 2011 Renesas Electronics Corporation. All Rights Reserved.*/
29    /***********************************************************************
30    *   System Name : SH7264 Sample Program
31    *   File Name   : io_vdc3_alpha_blend.c
32    *   Abstract    : VDC3, How to use the alpha-blending window function
33    *   Version     : 1.00.00
34    *   Device      : SH7264
35    *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36    *               : C/C++ compiler package for the SuperH RISC engine family
37    *               :                              (Ver.9.03 Release00).
38    *   OS          : None
39    *   H/W Platform: M3A-HS64G50(CPU board), M3A-HS64G02(Option board)
40    *   Description :
41    ************************************************************************
42    *   History     : Feb.28,2011 Ver.1.00.00
43    ************************************************************************/
44
45
```

3.5 Sample Program Listing "io_vdc3_alpha_blend.c" (2/11)

```
46  /***********************************************************************
47  Includes  <System Includes> , "Project Includes"
48  ***********************************************************************/
49  #include "iodefine.h"
50  #include "io_vdc3_alpha_blend.h"
51
52  /***********************************************************************
53  Exported global variables and functions (to be accessed by other files)
54  ***********************************************************************/
55  /* ==== Global functions ==== */
56  void io_vdc3_init(void);
57  void io_vdc3_change_buffer( int grphcs_no, unsigned short *buffer );
58
59  /* ==== Global variables ==== */
60  #pragma section GRPH1_BUFF /* Allocates the buffer at the 128-byte or 16-byte boundary
61                                in cache-disabled space */
62  unsigned short grph_buffer1[2][GRPHCS1_Y_SIZE][(GRPHCS1_LINE_OFFSET / BYTES_PER_PIXEL)];
63  #pragma section
64
65  #pragma section GRPH2_BUFF /* Allocates the buffer at the 128-byte or 16-byte boundary
66                                in cache-disabled space */
67  unsigned short grph_buffer2[2][GRPHCS2_Y_SIZE][(GRPHCS2_LINE_OFFSET / BYTES_PER_PIXEL)];
68  #pragma section
69
70  #pragma section VLINE_BUFF /* Allocates the buffer at the 128-byte or 16-byte boundary
71                                in cache-disabled space */
72  unsigned short video_line_buffer[VOUT_LINEBUF_NUM][(VOUT_LINE_OFFSET / BYTES_PER_PIXEL)];
73  #pragma section
74
75
76  /***********************************************************************
77  Private global variables and functions
78  ***********************************************************************/
79  /* ==== Private fuctions ==== */
80  static void io_vdc3_init_video_in(void);
81  static void io_vdc3_init_video_out(void);
82  static void io_vdc3_init_grphcs1(void);
83  static void io_vdc3_init_grphcs2(void);
84  static void io_vdc3_init_disp(void);
85  static void io_vdc3_start(void);
86
```

3.6 Sample Program Listing "io_vdc3_alpha_blend.c" (3/11)

```
87    /***********************************************************************************
88     * ID           :
89     * Outline      : Initializes the VDC3
90     * Include      :
91     * Declaration  : void io_vdc3_init(void);
92     * Description  : Initializes the VDC3 to use the alpha blending window function.
93     *              : Input video format is the BT.656. TFT-LCD panel TX09D55VM1CDA
94     *              : (Hitachi Displays) is used in this application.
95     * Argument     : void
96     * Return Value : void
97     ***********************************************************************************/
98    void io_vdc3_init(void)
99    {
100     int i, j;
101
102     /* ==== Initializes the data ==== */
103     /* ---- The graphics image 1 ---- */
104     for( i = 0; i < GRPHCS1_Y_SIZE; i++){
105       for( j = 0 ; j < GRPHCS1_X_SIZE; j++){
106           grph_buffer1[0][i][j] = RGB565_BLACK;
107           grph_buffer1[1][i][j] = RGB565_BLACK;
108       }
109     }
110     /* ---- The graphics image 2 ---- */
111     for( i = 0; i < GRPHCS2_Y_SIZE; i++){
112       for( j = 0 ; j < GRPHCS2_X_SIZE; j++){
113           grph_buffer2[0][i][j] = RGB565_WHITE;
114           grph_buffer2[1][i][j] = RGB565_WHITE;
115       }
116     }
117
118     /* ==== PORT ==== */
119     /* ---- Video (in) ---- */
120     PORT.PFCR1.BIT.PF7MD = 3;       /* DV_DATA7 */
121     PORT.PFCR1.BIT.PF6MD = 3;       /* DV_DATA6 */
122     PORT.PFCR1.BIT.PF5MD = 3;       /* DV_DATA5 */
123     PORT.PFCR1.BIT.PF4MD = 3;       /* DV_DATA4 */
124     PORT.PFCR0.BIT.PF3MD = 3;       /* DV_DATA3 */
125     PORT.PFCR0.BIT.PF2MD = 3;       /* DV_DATA2 */
126     PORT.PFCR0.BIT.PF1MD = 3;       /* DV_DATA1 */
127     PORT.PFCR0.BIT.PF0MD = 3;       /* DV_DATA0 */
128     PORT.PECR1.BIT.PE5MD = 3;       /* DV_HSYNC */
129     PORT.PECR1.BIT.PE4MD = 3;       /* DV_VSYNC */
130     PORT.PFCR2.BIT.PF8MD = 3;       /* DV_CLK */
131
```

3.7 Sample Program Listing "io_vdc3_alpha_blend.c" (4/11)

```
132     /* ---- Display (out) ---- */
133     PORT.PGCR7.WORD = 0x5A01u;      /* LCD_DATA0 ( Bits 15 to 8 is H'5A. )*/
134     PORT.PGCR5.BIT.PG20MD= 1;       /* LCD_EXTCLK */
135     PORT.PGCR4.WORD = 0x1111u;      /* LCD_CLK, LCD_DE, LCD_HSYNC, LCD_VSYNC */
136     PORT.PGCR3.WORD = 0x1111u;      /* LCD_DATA15-12 */
137     PORT.PGCR2.WORD = 0x1111u;      /* LCD_DATA11-08 */
138     PORT.PGCR1.WORD = 0x1111u;      /* LCD_DATA07-04 */
139     PORT.PGCR0.BIT.PG3MD = 1;       /* LCD_DATA03 */
140     PORT.PGCR0.BIT.PG2MD = 1;       /* LCD_DATA02 */
141     PORT.PGCR0.BIT.PG1MD = 1;       /* LCD_DATA01 */
142
143     /* ==== CPG ==== */
144     CPG.STBCR7.BIT.MSTP74 = 0;      /* VDC3 */
145
146     /* ==== VDC3 ==== */
147     /* ---- Initializes the video receiving block ---- */
148     io_vdc3_init_video_in();
149
150     /* ---- Initializes the video supplying block ---- */
151     io_vdc3_init_video_out();
152
153     /* ---- Initializes the graphics block 1 ----  */
154     io_vdc3_init_grphcs1();
155
156     /* ---- Initializes the graphics block 2 ----  */
157     io_vdc3_init_grphcs2();
158
159     /* ---- Initializes the panel control block and output timing control block ---- */
160     io_vdc3_init_disp();
161
162     /* ---- Enables the operation ---- */
163     io_vdc3_start();
164   }
165
```

3.8 Sample Program Listing "io_vdc3_alpha_blend.c" (5/11)

```
/*****************************************************************************
 * ID           :
 * Outline      : Update the graphics image data
 * Include      :
 * Declaration  : void io_vdc3_change_buffer( int grphcs_no, unsigned short *buffer);
 * Description  : Switches to another buffer specified by an argument to read the
 *              : graphics image data. Update the image data before executing
 *              : this function.
 * Argument     : int grphcs_no ; I : Index number of the buffer to display
 *              :                   : (1 : graphics 1, 2 : graphics 2)
 *              : unsigned short * buffer ; I : buffer address
 * Return Value : void
 *****************************************************************************/
void io_vdc3_change_buffer( int grphcs_no, unsigned short *buffer )
{
  if( grphcs_no == 1 ){
    VDC3.GROPSADR1.LONG = (0x1FFFFFFFul & (unsigned long)buffer);
    VDC3.GRCMEN1.BIT.WE = 1;      /* Transfers the graphics setting
                                          (Enabled from next Vsync) */
  }
  else{
    VDC3.GROPSADR2.LONG = (0x1FFFFFFFul & (unsigned long)buffer);
    VDC3.GRCMEN2.BIT.WE = 1;      /* Transfers the graphics setting
                                          (Enabled from next Vsync) */
  }
}

/*****************************************************************************
 * ID           :
 * Outline      : Initializes the video receiving block
 * Include      : iodefine.h
 * Declaration  : static void io_vdc3_init_video_in(void);
 * Description  : Initializes the video receiving block.
 *              : BT.656 is used as the input video format.
 * Argument     : void
 * Return Value : void
 *****************************************************************************/
static void io_vdc3_init_video_in(void)
{
  /* ----Input video format setting ---- */
  VDC3.VIDEO_MODE.BIT.SEL_EXSYNC = 0;           /* Disables the external input
                                                    sync signal */
  VDC3.VIDEO_MODE.BIT.SEL_656601 = 0;           /* Specifies the BT.656 input */
  VDC3.VIDEO_MODE.BIT.SEL_525625 = 0;           /* Number of lines for the
                                                    input video: 525 (NTSC) */
  VDC3.VIDEO_TIM_CNT.LONG    = 0x00000000ul;    /* Latches the DV_DATA input
                                                    signal at the rising edge */
                                                /* Other control signals settings
                                                    are not required for the BT656) */
}
```

3.9 Sample Program Listing "io_vdc3_alpha_blend.c" (6/11)

```
217  /***********************************************************************
218   * ID           :
219   * Outline      : Initializes the video supplying block
220   * Include      : iodefine.h
221   * Declaration  : static void io_vdc3_init_video_out(void);
222   * Description  : Initializes the video supplying block.
223   *              : BT.656 is used as the input video format.
224   * Argument     : void
225   * Return Value : void
226   ***********************************************************************/
227  static void io_vdc3_init_video_out(void)
228  {
229    /* ---- Video display function setting (NTSC, BT.656) ---- */
230    VDC3.VIDEO_MODE.BIT.BURST_MODE_DISP = 0;       /* Bus in the video supplying block:
231                                                       16-byte burst transfer */
232    VDC3.VIDEO_MODE.BIT.BURST_MODE_MAIN = 0;       /* Bus in the video receiving block:
233                                                       16-byte burst transfer */
234    VDC3.VIDEO_MODE.BIT.ENDIAN_DISP     = 0;       /* Bus in the video supplying block:
235                                                       big endian */
236    VDC3.VIDEO_MODE.BIT.ENDIAN_MAIN     = 0;       /* Bus in the video receiving block:
237                                                       big endian */
238    VDC3.VIDEO_MODE.BIT.VIDEO_MODE      = 1;       /* Specifies the video display function */
239    VDC3.VIDEO_INT_CNT.LONG             = 0x00000000ul;/* Disables video interrupts */
240    VDC3.VIDEO_SIZE.BIT.VIDEO_HEIGHT    = VIN_INPUT_HEIGHT;
241    VDC3.VIDEO_SIZE.BIT.VIDEO_WIDTH     = VIN_INPUT_WIDTH;
242                                                 /* Specifies the number of lines
243                                                     and pixels of the valid video */
244    VDC3.VIDEO_VSTART.BIT.VIDEO_VSTART_TOP = VIN_VSTART_VALIDDATA_TOP;
245    VDC3.VIDEO_VSTART.BIT.VIDEO_VSTART_BTM = VIN_VSTART_VALIDDATA_BTM;
246    VDC3.VIDEO_HSTART.BIT.VIDEO_HSTART     = VIN_HSTART_VALIDDATA;
247                                                 /* Vertical start position of the
248                                                     valid video in the TOP field
249                                                     Vertical start position of the
250                                                     valid video in the BOTTOM field
251                                                     Horizontal start position of the
252                                                     valid video */
253    VDC3.VIDEO_VSYNC_TIM1.BIT.VIDEO_VSYNC_START1_TOP
254     = VIN_VSTART_VALIDDATA_TOP+VOUT_IO_DFLINE-TFT_DE_START_V+(VOUT_BUF_MARGIN/2)-
     VOUT_DISP_POS_IO_DF;
255    VDC3.VIDEO_VSYNC_TIM1.BIT.VIDEO_VSYNC_START1_BTM
256     = VIN_VSTART_VALIDDATA_BTM+VOUT_IO_DFLINE-TFT_DE_START_V+(VOUT_BUF_MARGIN/2)-
     VOUT_DISP_POS_IO_DF;
257                                                 /* Specifies the reference Vsync position
258                                                     in the TOP field and BOTTOM field */
259    VDC3.VIDEO_SAVE_NUM.BIT.FIELD_SAVE_NUM  = 0;/* Number of fields to store (Set to 0
260                                                     in the video display function)*/
261    VDC3.VIDEO_IMAGE_CNT.LONG = 0x80800311ul;  /* Luminance contrast not adjusted */
262                                                 /* Luminance brightness not adjusted */
263                                                 /* Luminance clipping is valid */
264                                                 /* Chrominance clipping is valid */
265                                                 /* Scales down vertically to 1/3 */
266                                                 /* Scales down horizontally to 1/3 */
```

3.10 Sample Program Listing "io_vdc3_alpha_blend.c" (7/11)

```
267      VDC3.VIDEO_BASEADR.LONG         = (unsigned long)video_line_buffer;
268      VDC3.VIDEO_LINE_OFFSET.LONG     = VOUT_LINE_OFFSET;
269      VDC3.VIDEO_FIELD_OFFSET.LONG        = 0;    /* Field offset (Disabled in the video
270                                                     disable function) */
271      VDC3.VIDEO_LINEBUFF_NUM.BIT.VIDEO_LINEBUFF_NUM = VOUT_LINEBUF_NUM;
272                                                  /* Specifies the line buffer address,
273                                                     line offset (Number of bytes per line),
274                                                     and the number of lines to use */
275      VDC3.VIDEO_DISP_SIZE.BIT.VIDEO_DISP_HEIGHT   = VOUT_DISP_SZ_Y;
276      VDC3.VIDEO_DISP_SIZE.BIT.VIDEO_DISP_WIDTH    = VOUT_DISP_SZ_X;
277      VDC3.VIDEO_DISP_HSTART.BIT.VIDEO_DISP_HSTART = (TFT_DE_START_H-16);
278                                                  /* Specifies the vertical and
279                                                      horizontal size, horizontal start
280                                                      position of the video */
281      VDC3.SGMODE.BIT.EX_SYNC_MODE= 1;            /* Synchronizes the reference Vsync
282                                                      with the Vsync for video input */
283      VDC3.VIDEO_VSYNC_TIM2.LONG      = (TFT_DE_START_V + VOUT_DISP_POS_Y - 1);
284                                                  /* Specifies the timing of the Vsync
285                                                      for video display */
286    }
287
```

3.11 Sample Program Listing "io_vdc3_alpha_blend.c" (8/11)

```
288  /***********************************************************************************
289   * ID           :
290   * Outline      : Initializes the graphics block 1
291   * Include      : iodefine.h
292   * Declaration  : static void io_vdc3_init_grphcs1(void);
293   * Description  : Initializes the graphics block 1.
294   * Argument     : void
295   * Return Value : void
296   ***********************************************************************************/
297  static void io_vdc3_init_grphcs1(void)
298  {
299    /* ---- Graphics output setting ---- */
300    VDC3.GRA_VSYNC_TIM.LONG = 0;           /* Sets the timing for
301                                              the graphics image VSYNC */
302    VDC3.GRCBUSCNT1.LONG = 0;              /* Sets the bus between the graphics
303                                              image data and the VDC3 */
304                                           /* (16-byte burst transfer,
305                                              RGB565 format, and big endian) */
306    VDC3.GROPSADR1.BIT.GROPSADR = (0x1FFFFFFFul & (unsigned long)grph_buffer1[0]);
307                                           /* Sets the start address of
308                                              the graphics image data */
309    VDC3.GROPSWH1.BIT.GROPSH = GRPHCS1_Y_SIZE; /* Number of lines for
310                                              the graphics image area */
311    VDC3.GROPSWH1.BIT.GROPSW = GRPHCS1_X_SIZE; /* Number of pixels for
312                                              the graphics image area */
313    VDC3.GROPSOFST1.BIT.GROPSOFST = GRPHCS1_LINE_OFFSET;
314                                           /* Sets the number of bytes per line */
315    VDC3.GROPDPHV1.BIT.GROPDPV = GRPHCS1_POS_Y + TFT_DE_START_V - 1;
316                                           /* Graphics image area start position in
317                                              the vertical direction */
318    VDC3.GROPDPHV1.BIT.GROPDPH = GRPHCS1_POS_X + TFT_DE_START_H -16;
319                                           /* Graphics image area start position in
320                                              the horizontal direction */
321    VDC3.GROPBASERGB1.LONG   = 0x00000000ul;    /* Specifies the color in the
322                                              blanking area (RGB565, black) */
323
324    /* ---- Alpha control area setting ---- */
325    VDC3.GROPEWH1.BIT.GROPEH    = GRPHCS1_ALPHA_Y_SIZE;
326                                           /* Height of the alpha control area */
327    VDC3.GROPEWH1.BIT.GROPEW    = GRPHCS1_ALPHA_X_SIZE;
328                                           /* Width of the alpha control area */
329    VDC3.GROPEDPHV1.BIT.GROPEDPV= GRPHCS1_ALPHA_POS_Y + TFT_DE_START_V - 1;
330                                           /* Vertical start position */
331    VDC3.GROPEDPHV1.BIT.GROPEDPH= GRPHCS1_ALPHA_POS_X + TFT_DE_START_H -16;
332                                           /* Horizontal start position */
333    VDC3.GROPEDPA1.BIT.DEFA     = 255;  /* Alpha value (default) */
334    VDC3.GROPEDPA1.BIT.ACOEF    = 1;    /* Alpha value calculation coefficient */
335    VDC3.GROPEDPA1.BIT.ARATE    = 0;    /* Calculates alpha value per frame rate */
336    VDC3.GROPEDPA1.BIT.WE       = 1;    /* Enables to transfer alpha value */
337    VDC3.GROPEDPA1.BIT.AMOD     = 2;    /* Adds to alpha value */
338    VDC3.GROPEDPA1.BIT.AEN      = 1;    /* Enables or disables to control alpha value */
339  }
340
```

3.12 Sample Program Listing "io_vdc3_alpha_blend.c" (9/11)

```
341  /***********************************************************************
342   * ID           :
343   * Outline      : Initializes the graphics block 2
344   * Include      : iodefine.h
345   * Declaration  : static void io_vdc3_init_grphcs2(void);
346   * Description  : Initializes the graphics block 2.
347   * Argument     : void
348   * Return Value : void
349   ***********************************************************************/
350  static void io_vdc3_init_grphcs2(void)
351  {
352    /* ---- Graphics output setting ---- */
353    VDC3.GRA_VSYNC_TIM.LONG = 0;           /* Sets the timing for
354                                              the graphics image VSYNC */
355    VDC3.GRCBUSCNT2.LONG = 0;              /* Sets the bus between the graphics
356                                              image data and the VDC3 */
357                                           /* (16-byte burst transfer,
358                                              RGB565 format, and big endian) */
359    VDC3.GROPSADR2.BIT.GROPSADR = (0x1FFFFFFFul & (unsigned long)grph_buffer2[0]);
360                                           /* Sets the start address of
361                                              the graphics image data */
362    VDC3.GROPSWH2.BIT.GROPSH = GRPHCS2_Y_SIZE; /* Number of lines for
363                                              the graphics image area */
364    VDC3.GROPSWH2.BIT.GROPSW = GRPHCS2_X_SIZE; /* Number of pixels for
365                                              the graphics image area */
366    VDC3.GROPSOFST2.BIT.GROPSOFST = GRPHCS2_LINE_OFFSET;
367                                           /* Sets the number of bytes per line */
368    VDC3.GROPDPHV2.BIT.GROPDPV = GRPHCS2_POS_Y + TFT_DE_START_V - 1;
369                                           /* Graphics image area start position in
370                                              the vertical direction */
371    VDC3.GROPDPHV2.BIT.GROPDPH = GRPHCS2_POS_X + TFT_DE_START_H -16;
372                                           /* Graphics image area start position in
373                                              the horizontal direction */
374    VDC3.GROPBASERGB2.LONG   = 0x00000000ul;    /* Specifies the color in the
375                                              blanking area (RGB565, black) */
376    /* ---- Alpha control area setting ---- */
377    VDC3.GROPEWH2.BIT.GROPEH    = GRPHCS2_ALPHA_Y_SIZE;
378                                           /* Height of the alpha control area */
379    VDC3.GROPEWH2.BIT.GROPEW    = GRPHCS2_ALPHA_X_SIZE;
380                                           /* Width of the alpha control area */
381    VDC3.GROPEDPHV2.BIT.GROPEDPV= GRPHCS2_ALPHA_POS_Y + TFT_DE_START_V - 1;
382                                           /* Vertical start position */
383    VDC3.GROPEDPHV2.BIT.GROPEDPH= GRPHCS2_ALPHA_POS_X + TFT_DE_START_H -16;
384                                           /* Horizontal start position */
385    VDC3.GROPEDPA2.BIT.DEFA     = 0;    /* Alpha value (default) */
386    VDC3.GROPEDPA2.BIT.ACOEF    = 1;    /* Alpha value calculation coefficient */
387    VDC3.GROPEDPA2.BIT.ARATE    = 0;    /* Calculates alpha value per frame rate */
388    VDC3.GROPEDPA2.BIT.WE       = 1;    /* Enables to transfer alpha value */
389    VDC3.GROPEDPA2.BIT.AMOD     = 1;    /* Adds to alpha value */
390    VDC3.GROPEDPA2.BIT.AEN      = 1;    /* Enables or disables to control alpha value */
391  }
```

3.13 Sample Program Listing "io_vdc3_alpha_blend.c" (10/11)

```
392    /***********************************************************************
393     * ID           :
394     * Outline      : Initializes the panel control block and output timing control block
395     * Include      : iodefine.h
396     * Declaration  : static void io_vdc3_init_disp(void);
397     * Description  : Initializes the panel control block and output timing control block.
398     * Argument     : void
399     * Return Value : void
400     ***********************************************************************/
401    static void io_vdc3_init_disp(void)
402    {
403      /* ---- TFT-LCD panel control signal output setting ---- */
404      VDC3.SYNCNT.LONG = 0x0000000Cul;               /* Outputs all signals at the
405                                                         rising edge */
406                                                     /* LCD_VSYNC/LCD_HSYNC signal:
407                                                         output is inverted */
408      VDC3.PANEL_CLKSEL.BIT.ICKEN = 0;               /* Disables the operation of
409                                                         the panel clock */
410      VDC3.PANEL_CLKSEL.LONG      = 0x0000200Cul;    /* Clock source: Bφ (72 MHz) */
411                                                     /* Clock frequency: 6 MHz */
412      VDC3.PANEL_CLKSEL.BIT.ICKEN = 1;               /* Enables the operation of
413                                                         the panel clock */
414      VDC3.SYN_SIZE.BIT.SYN_HEIGHT= TFT_TOTAL_SZ_V;  /* Number of lines including the
415                                                         vertical blanking interval */
416      VDC3.SYN_SIZE.BIT.SYN_WIDTH = TFT_TOTAL_SZ_H;  /* Number of pixels including the
417                                                         horizontal blanking interval */
418      VDC3.PANEL_VSYNC_TIM.LONG   = TFT_VSYNC_WDTH;  /* Sets the timing for the
419                                                         panel output VSYNC */
420      VDC3.PANEL_HSYNC_TIM.LONG   = TFT_HSYNC_WDTH;  /* Sets the timing for the
421                                                         panel output HSYNC */
422      VDC3.DE_SIZE.BIT.DE_HEIGHT  = TFT_DISP_SZ_V;   /* Number of lines for the DE area */
423      VDC3.DE_SIZE.BIT.DE_WIDTH   = TFT_DISP_SZ_H;   /* Number of pixels for the DE area */
424      VDC3.DE_START.BIT.DE_START_V= TFT_DE_START_V;  /* DE area start position in
425                                                         the vertical direction */
426      VDC3.DE_START.BIT.DE_START_H= TFT_DE_START_H;  /* DE area start position in
427                                                         the horizontal direction */
428    }
429
```

3.14 Sample Program Listing "io_vdc3_alpha_blend.c" (11/11)

```
430    /*****************************************************************************
431     * ID           :
432     * Outline      : Enables the operation
433     * Include      : iodefine.h
434     * Declaration  : static void io_vdc3_start(void);
435     * Description  : Enables the operation.(Enabled from the next Vsync)
436     * Argument     : void
437     * Return Value : void
438     *****************************************************************************/
439    static void io_vdc3_start(void)
440    {
441      /*  ---- Enables the video receiving block  ---- */
442      VDC3.VIDEO_MODE.BIT.VIDEO_MAIN_EXE  = 1;
443
444      /*  ---- Enables the video supplying block  ---- */
445      VDC3.VIDEO_MODE.BIT.VIDEO_DISP_EXE  = 1;
446
447      /*  ---- Enables the graphics block 2  ---- */
448      VDC3.GRCMEN2.LONG = 0x80000003ul;   /* Enables to display the current graphics,
449                                             enables to display the lower-layer graphics */
450
451      /*  ---- Enables the graphics block 1  ---- */
452      VDC3.GRCMEN1.LONG = 0x80000003ul;   /* Enables to display the current graphics,
453                                             enables to display the lower-layer graphics */
454    }
455    /* End of File */
456
```

3.15 Sample Program Listing "io_vdc3_alpha_blend.h" (1/3)

```
1       /***********************************************************************
2       *   DISCLAIMER
3       *
4       *   This software is supplied by Renesas Electronics Corporation and is only
5       *   intended for use with Renesas products. No other uses are authorized.
6       *
7       *   This software is owned by Renesas Electronics Corporation and is protected under
8       *   all applicable laws, including copyright laws.
9       *
10      *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11      *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12      *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13      *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14      *   DISCLAIMED.
15      *
16      *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17      *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18      *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19      *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20      *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21      *
22      *   Renesas reserves the right, without notice, to make changes to this
23      *   software and to discontinue the availability of this software.
24      *   By using this software, you agree to the additional terms and
25      *   conditions found by accessing the following link:
26      *   http://www.renesas.com/disclaimer
27      ***********************************************************************/
28      /*   Copyright (C) 2011 Renesas Electronics Corporation. All Rights Reserved.*/
29      /***********************************************************************
30      *   System Name : SH7264 Sample Program
31      *   File Name   : io_vdc3_alpha_blend.h
32      *   Abstract    : VDC3, How to use the alpha-blending window function
33      *   Version     : 1.00.00
34      *   Device      : SH7264
35      *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36      *               : C/C++ compiler package for the SuperH RISC engine family
37      *               :                              (Ver.9.03 Release00).
38      *   OS          : None
39      *   H/W Platform: M3A-HS64G50(CPU board), M3A-HS64G02(Option board)
40      *   Description :
41      ************************************************************************
42      *   History     : Feb.28,2011 Ver.1.00.00
43      ************************************************************************/
44
45
```

3.16 Sample Program Listing "io_vdc3_alpha_blend.h" (2/3)

```
46      /***********************************************************************
47      Macro definitions
48      ***********************************************************************/
49      #define BYTES_PER_PIXEL       2       /* Number of bytes per pixel */
50      #define RGB565_BLACK          0x0000u /* Black */
51      #define RGB565_WHITE          0xFFFFu /* White */
52      #define RGB565_GREEN          0x07E0u /* Green */
53      #define RGB565_BLUE           0x001Fu /* Blue */
54
55      /* ---- Video input parameters ---- */
56      #define VIN_VSTART_VALIDDATA_TOP 16   /* Vertical capture timing in the TOP field */
57      #define VIN_VSTART_VALIDDATA_BTM 279  /* Vertical capture timing in the BOTTOM field */
58      #define VIN_HSTART_VALIDDATA  276     /* Horizontal capture timing */
59      #define VIN_INPUT_HEIGHT      240     /* Number of lines of the input valid video */
60      #define VIN_INPUT_WIDTH       720     /* Number of pixels of the input valid video */
61
62      /* ---- Video output parameters ---- */
63      #define VOUT_DISP_SZ_Y        160     /* Video display area height */
64      #define VOUT_DISP_SZ_X        240     /* Video display area width */
65      #define VOUT_BUF_MARGIN       6       /* Line buffer margin (specify 6 or greater) */
66      #define VOUT_IO_DFLINE        127     /* Difference in number of lines between
67                                               the input cycle and display cycle */
68                                    /*  = VIN_INPUT_HEIGHT -(VOUT_DISP_SZ_Y * (0.045/0.064))*/
69      #define VOUT_LINEBUF_NUM      (VOUT_IO_DFLINE + VOUT_BUF_MARGIN)
70                                           /* Number of lines of line buffer */
71      #define VOUT_LINE_OFFSET      (((VOUT_DISP_SZ_X * BYTES_PER_PIXEL) + 15 ) & 0xFFFFFFF0ul)
72                                           /* Number of bytes per line */
73      #define VOUT_DISP_POS_Y       80      /* Vertical start position of the video
74                                               (from the top of the panel) */
75      #define VOUT_DISP_POS_X       0       /* Horizontal start position of the video
76                                               (from the leftmost of the panel) */
77      #define VOUT_DISP_POS_IO_DF   56      /* Converted value of the VD_DISP_POS_Y
78                                                to the number of lines of the input video */
79                                           /*  = VOUT_DISP_POS_Y * (0.045/0.064) */
80
81      /* ---- Graphics image parameters ---- */
82      #define GRPHCS1_Y_SIZE        80      /* Height of Graphics image 1 */
83      #define GRPHCS1_X_SIZE        80      /* Width of Graphics image 1 */
84      #define GRPHCS1_LINE_OFFSET   (((GRPHCS1_X_SIZE * BYTES_PER_PIXEL) + 15 ) & 0xFFFFFFF0ul)
85                                           /* Line offset of Graphics image 1 */
86      #define GRPHCS1_POS_Y         100     /* Vertical display start position
87                                               (0: top of the panel) */
88      #define GRPHCS1_POS_X         60      /* Horizontal display start position
89                                               (0: leftmost of the panel) */
90      #define GRPHCS1_ALPHA_Y_SIZE  320     /* Height of the alpha control area */
91      #define GRPHCS1_ALPHA_X_SIZE  240     /* Width of the alpha control area */
92      #define GRPHCS1_ALPHA_POS_Y   0       /* Vertical start position of
93                                               the alpha control area */
94      #define GRPHCS1_ALPHA_POS_X   0       /* Horizontal start position of
95                                               the alpha control area */
```

3.17 Sample Program Listing "io_vdc3_alpha_blend.h" (3/3)

```
96
97      #define GRPHCS2_Y_SIZE         80      /* Height of Graphics image 2 */
98      #define GRPHCS2_X_SIZE         80      /* Width of Graphics image 2 */
99      #define GRPHCS2_LINE_OFFSET    (((GRPHCS2_X_SIZE * BYTES_PER_PIXEL) + 15 ) & 0xFFFFFFF0ul)
100                                            /* Line offset of Graphics image 2 */
101     #define GRPHCS2_POS_Y          140     /* Vertical display start position
102                                               (0: top of the panel) */
103     #define GRPHCS2_POS_X          100     /* Horizontal display start position
104                                               (0: leftmost of the panel) */
105     #define GRPHCS2_ALPHA_Y_SIZE   320     /* Height of the alpha control area */
106     #define GRPHCS2_ALPHA_X_SIZE   240     /* Width of the alpha control area */
107     #define GRPHCS2_ALPHA_POS_Y    0       /* Vertical start position of
108                                               the alpha control area */
109     #define GRPHCS2_ALPHA_POS_X    0       /* Horizontal start position of
110                                               the alpha control area */
111
112     /* ---- TFT-LCD panel parameters ---- */
113     #define TFT_TOTAL_SZ_V         327     /* Number of lines including the vertical
114                                               blanking interval */
115     #define TFT_TOTAL_SZ_H         273     /* Number of pixels including the horizontal
116                                               blanking interval */
117     #define TFT_DISP_SZ_V          320     /* Vertical display enable interval */
118     #define TFT_DISP_SZ_H          240     /* Horizontal display enable interval */
119     #define TFT_VSYNC_WDTH         1       /* LCD_VSYNC pulse width (number of lines) */
120     #define TFT_HSYNC_WDTH         5       /* LCD_HSYNC pulse width (number of pixels) */
121     #define TFT_DE_START_V         1       /* Number of lines between the reference Vsync
122                                               and the enable interval */
123     #define TFT_DE_START_H         (TFT_TOTAL_SZ_H - 251)
124                                            /* Number of pixels between the reference
125                                               Hsync and the enable interval */
126
127     /*****************************************************************************
128     Imported global variables and functions (from other files)
129     *****************************************************************************/
130     /* ==== Global functions ==== */
131     extern void io_vdc3_init(void);
132     extern void io_vdc3_change_buffer( int grphcs_no, unsigned short *buffer );
133
134     /* ==== Global variables ==== */
135     extern unsigned short grph_buffer1[2][GRPHCS1_Y_SIZE][(GRPHCS1_LINE_OFFSET /
        BYTES_PER_PIXEL)];
136     extern unsigned short grph_buffer2[2][GRPHCS2_Y_SIZE][(GRPHCS2_LINE_OFFSET /
        BYTES_PER_PIXEL)];
137     extern unsigned short video_line_buffer[VOUT_LINEBUF_NUM][(VOUT_LINE_OFFSET /
        BYTES_PER_PIXEL)];
138
139     /* End of File */
140
```

4. References

- Software Manual
  SH-2A/SH2A-FPU Software Manual Rev.3.00
  The latest version of the software manual can be downloaded from the Renesas Electronics website.

- Hardware Manual
  SH7262 Group, SH7264 Group Hardware manual Rev.2.00
  The latest version of the hardware manual can be downloaded from the Renesas Electronics website.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	May 19.09	—	First edition issued
1.01	Jan.19.10	16	Figure 9, Note added
		25	Supplement to the sample program added for 640-KB RAM
		31	Writing procedure to the PGCR7 updated
		37, 39	Start position of the alpha control area updated
		26 to 39	Format for header comments updated
1.02	Mar.23.11	—	—
1.03	Mar.23.11	15 to 41	Changed the configuration of the source code

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0