

Configuring and using the DCU2 on the MPC5606S MCU

by: **Steve McAslan**
Microcontroller Solutions Group

Contents

1 Introduction

The Display Control Unit features a rich set of capabilities that allow users to build engaging and dynamic graphic content from pre-rendered images and display this content on a wide range of TFT LCD panels. This application note explains the steps to configure and use the DCU with TFT LCD panels and how to display and blend graphics on the panel. See www.freescale.com for more details on how a user can use these features in a typical application.

2 Hardware Interface

In most cases the DCU2 connects directly to a TFT LCD panel through the MCU I/O pins. The DCU requires that the panel includes an on-chip timing controller (TCON) function which accepts a parallel RGB interface consisting of up to 24-bits of pixel data (RGB), a pixel clock, an optional data active (enable) line, optional horizontal sync (HSYNC) and optional vertical sync (VSYNC) timing signals.

It is possible to use a panel that accepts fewer than 24-bit data input and in this case it is normal to connect the most significant bits from the DCU2 to the panel data pins. For example, in the case of an 18-bit panel (RGB each with 6-bits) it is normal to connect as shown in [Table 1](#).

1	Introduction.....	1
2	Hardware Interface.....	1
3	Enabling the DCU2.....	2
3.1	Choosing a suitable clock source.....	2
3.2	Making the DCU2 active on the MPC5606S.....	2
4	Configuring the DCU2.....	3
4.1	Configuring the DCU for a TFT LCD panel.....	3
4.2	Testing the hardware and software configuration.....	7
5	Using the DCU2.....	8
5.1	Using the graphic layers.....	9
5.2	Using the cursor.....	11
5.3	Using the PDI.....	12
6	Summary.....	13

NOTE

The unused pins may be used as GPIO or other function where available.

Table 1. Recommended connection for 18-bit panels

MCU = Panel	MCU = Panel	MCU = Panel
R7 = R5	G7 = G5	B7 = B5
R6 = R4	G6 = G4	B6 = B4
R5 = R3	G5 = G3	B5 = B3
R4 = R2	G4 = G2	B4 = B2
R3 = R1	G3 = G1	B3 = B1
R2 = R0	G2 = G0	B2 = B0
R1 unused	G1 unused	B1 unused
R0 unused	G0 unused	B0 unused

Other timing signals are connected to the panel as required. For example, some panels do not require a horizontal sync signal and in this case the pin can be used as GPIO.

3 Enabling the DCU2

Enabling the DCU2 on the MPC5606S microcontroller requires two main steps: the DCU2 must be made active in the current MCU mode (mode entry control) and an appropriate clock must be supplied to the DCU2.

3.1 Choosing a suitable clock source

The DCU2 takes an input clock and divides it inside the module to produce the pixel clock required by the connected TFT LCD panel. The DCU2 can choose from the fast internal IRC clock (FIRC) which operates at approximately 16 MHz; the system clock FMPLL0; the secondary PLL FMPLL1; and the fast external crystal oscillator FXOSC (4-16MHz). A typical TFT LCD panel needs to be refreshed at approximately 60 Hz. The greater the number of pixels on the panel, the faster the pixel clock has to be, in order to transfer each pixel in the available time.

Pick a clock source that can give the best match to the required pixel clock when divided by an integer value. For example, a QVGA panel requires a pixel clock of approximately 5.33 MHz. If the system clock is 64 MHz then a divider of 12 gives a near-exact match to this value. A wide QVGA panel (480 x 272) requires a pixel clock of approximately 9 MHz. If the system clock is 64 MHz then a divider of 7 gives a close match of 9.11 MHz. If the system clock is not 64 MHz or a more exact match is required, then it may be required to use an alternative clock source such as the FMPLL1.

The clock provided to the DCU2 to generate the pixel clock is one of the MCU auxiliary clock sources. Select the clock source required by the DCU2 by writing to the Auxiliary Clock 0 Select register (CGM_AC0_SC).

3.2 Making the DCU2 active on the MPC5606S

It is possible to have the DCU2 operating in all MCU RUN and Low Power modes except STANDBY. In a typical application the DCU2 will only be active in RUN modes (DRUN, RUN0..3) and so only these will be discussed here.

Following steps are needed to enable a module:

1. The first step is to identify which Peripheral Control Register controls its operating modes; for the DCU this is PCTL63.
2. Next define the RUN modes in which the DCU is active. This may depend on the clocks that are available in the modes (FIRC is always available in all RUN modes). Choose or configure the modes so that the selected clock source is active.
3. Next choose a Run Peripheral Configuration Register (RUN_PC 0..7) that has the required active mode and set the RUN_CFG bit field of PCTL63 to select that register.

See for [Table 2](#) some examples of suitable values.

Table 2. Example values for RUN_PC and PCTL63 registers

DCU active in:	RUN_PC configured	Required RUN_PC setting	PCTL63 RUN_CFG value
DRUN, RUN2	1	0x48	1
RUN0, RUN1, RUN2	6	0x70	6
RUN3	0	0x80	0

It is possible to verify that the DCU2 is active by polling the S_DCU bit in the ME_PS1 register. Since the DCU also controls the I/O pins that are connected to the panel the SIUL module must be enabled in a similar way. The SIUL is normally enabled early in the initialization process since it is required before the MCU can manipulate any I/O pins. The SIUL on the MPC5606S is enabled using PCTL68.

4 Configuring the DCU2

Once the DCU is enabled and has a suitable clock, it is possible to begin configuring the module for the TFT LCD panel being used and verify that the connections are operating as expected.

4.1 Configuring the DCU for a TFT LCD panel

The first step is to configure the SIUL pins so that the DCU function is selected. This will be done by modifying the PCR register associated with each pin.

The relevant bit-fields in the PCR register are the alternate function (PA) field, the output buffer enable (OBE) field and possibly the slew rate control (SRC) field. Not all pins have an active SRC bit-field. Refer to the MPC5606S signal description chapter for details of the functions available on each of the pins.

To enable the DCU signals, configure the MPC5606S SIUL PCR registers as given in [Table 3](#). The SMC function relates to SAFE mode recovery and is application dependent. The configuration values for input and pull resistors are disabled but can be enabled if required for some reason.

Table 3. SIUL module PCR register recommended values

Signal	PCR no	SMC	PA[1:0]	OBE	IBE	ODE	SRC[1:0]	WPE	WPS
R0	0	x	1	1	0	0	0	0	0

Table continues on the next page...

Table 3. SIUL module PCR register recommended values (continued)

R1	1	x	1	1	0	0	0	0	0
R2	2	x	1	1	0	0	0	0	0
R3	3	x	1	1	0	0	0	0	0
R4	4	x	1	1	0	0	0	0	0
R5	5	x	1	1	0	0	0	0	0
R6	6	x	1	1	0	0	0	0	0
R7	7	x	1	1	0	0	0	0	0
G0	8	x	1	1	0	0	0	0	0
G1	9	x	1	1	0	0	0	0	0
G2	10	x	1	1	0	0	0	0	0
G3	11	x	1	1	0	0	0	0	0
G4	12	x	1	1	0	0	0	0	0
G5	13	x	1	1	0	0	0	0	0
G6	14	x	1	1	0	0	1	0	0
G7	15	x	1	1	0	0	0	0	0
B0	86	x	1	1	0	0	1	0	0
B1	87	x	1	1	0	0	0	0	0
B2	88	x	1	1	0	0	1	0	0
B3	89	x	1	1	0	0	0	0	0
B4	90	x	1	1	0	0	1	0	0
B5	91	x	1	1	0	0	0	0	0
B6	92	x	1	1	0	0	1	0	0
B7	93	x	1	1	0	0	0	0	0
VSYNC	94	x	1	1	0	0	1	0	0
HSYNC	95	x	1	1	0	0	0	0	0
DE	96	x	1	1	0	0	1	0	0
PCLK	97	x	1	1	0	0	0	0	0

The second step is to configure the DCU2 operating parameters to match the specification of the panel being used. The relevant configuration registers are given in [Table 4](#).

Table 4. DCU2 Panel configuration registers

Register	Function
DIV_RATIO	Divides the selected DCU2 auxiliary clock down to the required pixel clock value

Table continues on the next page...

Table 4. DCU2 Panel configuration registers (continued)

DISP_SIZE	Defines the number of horizontal and vertical pixels on the panel
HSYN_PARA	Defines the horizontal (line) timing parameters
VSYN_PARA	Defines the vertical (whole frame) timing parameters
SYN_POL	Defines the polarity of the timing signals

Start by providing the correct division ratio for the auxiliary clock provided to the DCU. This is a simple modulus division with the value in DIV_RATIO being one less than the divider required, for example a value of 0 gives a division of 1.

Table 5 shows some example DIV_RATIO values for different panel sizes. The table assumes the following pixel clock requirements for the example panels: QVGA (240 x320) = 5.33 MHz; WQVGA (480x272) = 9 MHz. Note that the chosen DIV_RATIO value will typically not be able to match the target pixel clock frequency exactly and so judgement must be used to verify that the value is suitable for the panel.

Table 5. Example DIV_RATIO values for different panels

Auxiliary clock	QVGA	WQVGA
64 MHz	11	6
56 MHz	9	5
48 MHz	8	4
32 MHz	5	3

Next configure the DISP_SIZE to match the panel size. The vertical dimension of the panel is entered directly in the DELTA_Y bit field. The horizontal dimension is configured differently. The DCU2 requires the width of the panel to be a multiple of 16 and the DELTA_X bit field indicates the number of multiples in the horizontal dimension. In other words the DELTA_X field contains the width of the panel divided by 16. See Table 6 for some typical examples.

Table 6. Example DISP_SIZE values for different panels

Panel size	DELTA_Y	DELTA_X
320H x 240V	240	20
240H x 320V	320	12
480H x 272V	272	24
640H x 240V	240	32

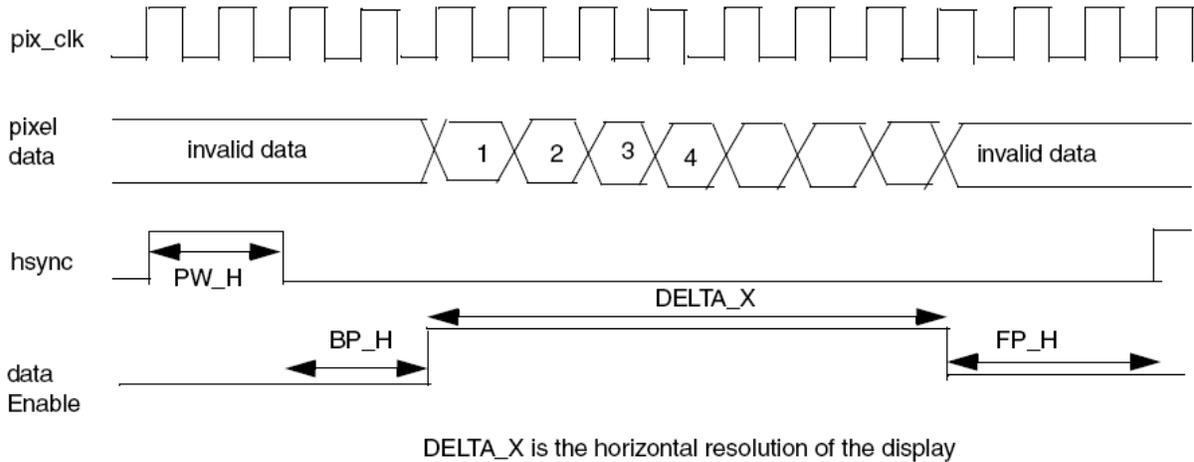
Now the synchronization signals must be configured for the panel. Typically each panel has unique timing requirements and not all timing signals are required for all panels, however manufacturers tend to specify the timing configuration in a consistent way that matches the parameter requirements for the DCU2. In most cases it is possible to read the timing parameter from a TFT LCD specification and enter that number directly into the relevant DCU2 configuration register.

Begin the configuration with the HSYN_PARA register that configures the horizontal signals. Three values are required as illustrated in Figure 1 and all are defined in multiples of the pixel clock:

Configuring the DCU2

- Horizontal pulse width which defines the number of pixel clocks for which the horizontal timing pulse is active (HSYNC signal).
- Horizontal back porch width which defines the number of pixel clocks after the horizontal pulse before the start of pixel data. Some manufacturers define the back porch starting from the start of the horizontal pulse rather than the end.
- Horizontal front porch width which defines the number of pixel clocks after the pixel data before the start of the horizontal pulse.

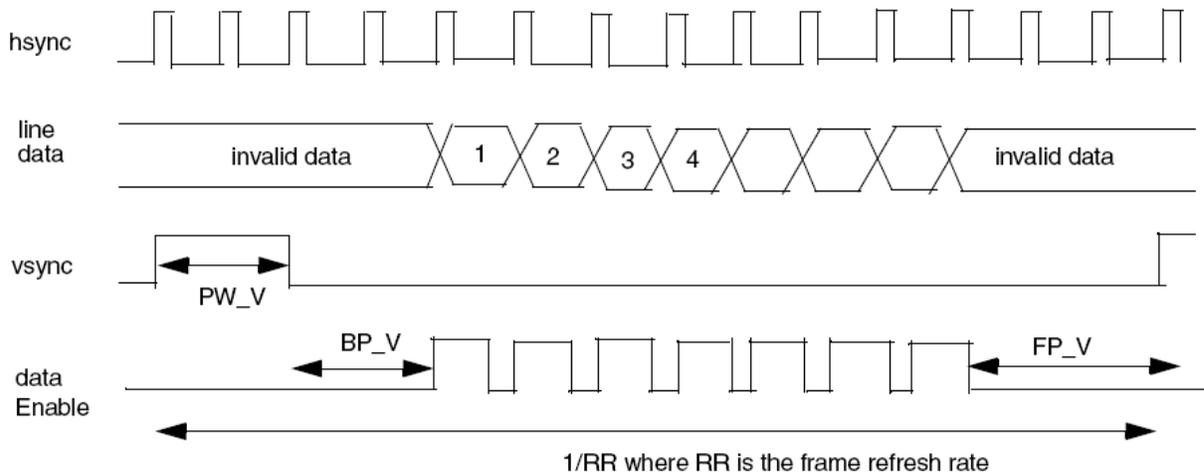
Figure 1. Horizontal timing diagram



Next configure the vertical timing signals using the VSYN_PARA register. Three values are required as illustrated in [Figure 2](#) and this time all are defined in multiples of horizontal lines:

- Vertical pulse width which defines the number of horizontal lines for which the vertical timing pulse is active (VSYNC signal).
- Vertical back porch width which defines the number of horizontal lines after the vertical pulse before the start of pixel data. Some manufacturers define the back porch starting from the start of the vertical pulse rather than the end.
- Vertical front porch width which defines the number of horizontal lines after the pixel data before the start of the vertical pulse.

Figure 2. Vertical timing diagram



The polarity of the pixel clock (PCLK), the horizontal timing signal (HSYNC) and the vertical timing signal (VSYNC) may be inverted using the SYN_POL register.

4.2 Testing the hardware and software configuration

After the DCU2 is enabled and configured, it is possible to display graphics on the panel or perform verification tests. [Using the DCU2](#) describes the normal use of the panel and this section provides guidance on how to verify the hardware and software configuration of the system.

The DCU2 includes a special test card mode that displays bands of known colors on the panel. This mode allows verification so that the connection to the panel and the DCU2 timing configuration is correct.

At this stage the panel is connected and the DCU2 is configured to drive the panel but no timing signals are being driven by the MPC5606S. It is possible to start the pixel clock before HSYNC, VSYNC and the pixel data; this is a requirement for some panels. It is also possible to start all signals at the same time.

To start the pixel clock before other timing signals, follow the given steps:

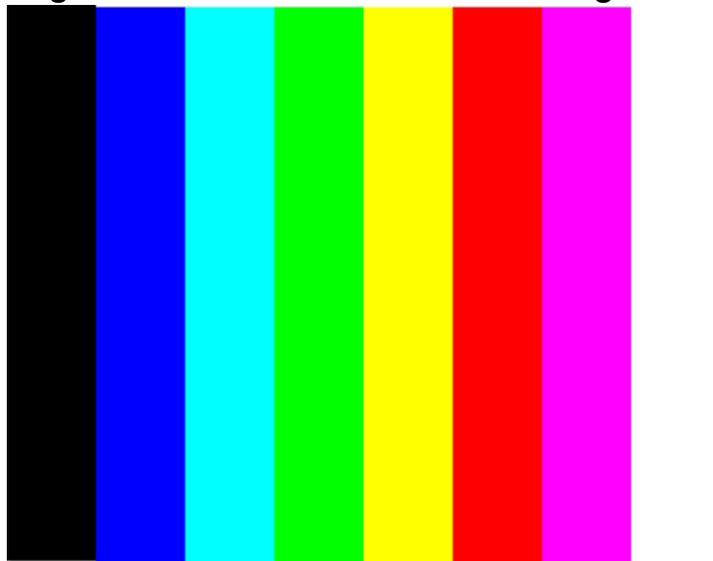
- In the DCU_MODE register, set the DCU_MODE bit-field to 3 which selects color bar mode
- In the DCU_MODE register set the RASTER_EN bit-field to 1 which enables the raster signals (HSYNC, VSYNC, pixel data)

To start all the signals together, follow the given step:

- In the DCU_MODE register set the DCU_MODE bit-field to 3 and the RASTER_EN bit-field to 1

The DCU2 will begin sending pixel data which divides the panel into 8 color bars horizontally. By default, each color bar is assigned a color as shown in [Figure 3](#).

Figure 3. Default color bar mode configuration



This standard configuration can be used to verify that the connection to the panel and the timing is correct. Verify the timing by measuring the MPC5606S signals and comparing them to the values expected by the panel. The pixel data connection integrity can be verified by examining the content of the panel. This setting also allows testing and configuration of the panel backlight control.

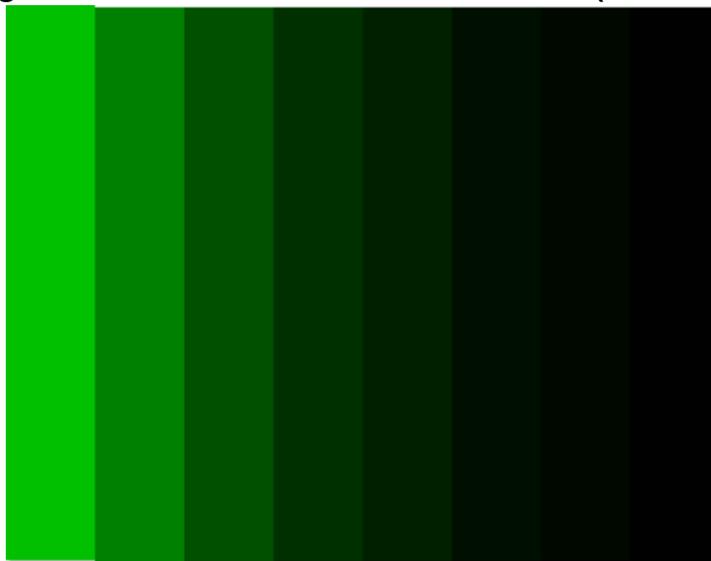
using the DCU2

It is possible to verify the connection of each of the pixel data lines by modifying the color bar color settings. This is done using the COLBAR_1 to COLBAR_8 registers. The registers are numbered to represent the bars from left to right so that the COLBAR_1 contains the color setting of the left-hand bar (in the default case this is black) and COLBAR_8 contains the color settings of the right-hand bar (white). Each COLBAR_n register contains an RGB888 value with 8 bits to define each of red, green and blue content of the bar. It is possible to change these registers to any other value. As an example it is possible to verify that each green data line is connected and that there are no short circuits by changing the COLBAR_n registers to contain only a single green data line value as follows:

- Set COLBAR_1 to 0x00008000
- Set COLBAR_2 to 0x00004000
- Set COLBAR_3 to 0x00002000
- Set COLBAR_4 to 0x00001000
- Set COLBAR_5 to 0x00000800
- Set COLBAR_6 to 0x00000400
- Set COLBAR_7 to 0x00000200
- Set COLBAR_8 to 0x00000100

The resulting panel content is shown in [Figure 4](#) . Each color bar should be a different shade of green with no red or blue visible. This is not a comprehensive test on its own but combined with other combinations and test approaches may be used to visually and electrically verify correct configuration.

Figure 4. Green data line connection test (simulated)



5 Using the DCU2

Once the panel hardware and software configuration is verified it is possible to begin using the DCU2 in its functional mode. To enable the panel for normal use:

- In the DCU_MODE register set the DCU_MODE bit-field to 1 which selects normal mode
- In the DCU_MODE register set the RASTER_EN bit-field to 1 which enables the raster signals (HSYNC, VSYNC, pixel data)
- OR to start all signals together, in the DCU_MODE register set the DCU_MODE bit-field to 1 and the RASTER_EN bit-field to 1

At this point the panel should be active and displaying a single color. This color is configured in the BGND register and is black after reset. Like the COLBAR_n registers this register can be modified to display any single color by modifying its value to another RGB888 value.

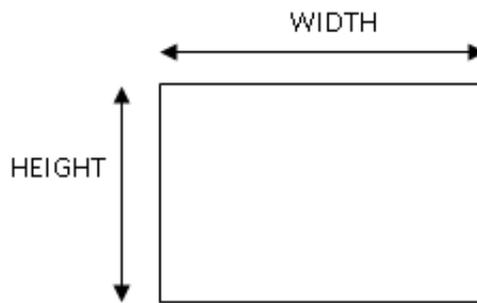
Functional content is placed on the panel by enabling graphic layers, the cursor or the Parallel Data Interface (PDI) input or any combination of those. These elements are discussed in the following sections with a focus on their basic properties. To see examples of how an application may use the different features please refer to demonstration software, application notes and video examples available at www.freescale.com.

5.1 Using the graphic layers

The most common way of presenting content on the panel is the layer. By configuring a layer it is possible to fetch a graphic directly from any memory mapped module and display it on the panel. The graphic data may be encoded in many different formats. There are 16 layers in total and each is configured using 7 registers. The layers are independent of each other and placed in a fixed priority on the panel. The steps to display a graphic on a layer are as follows:

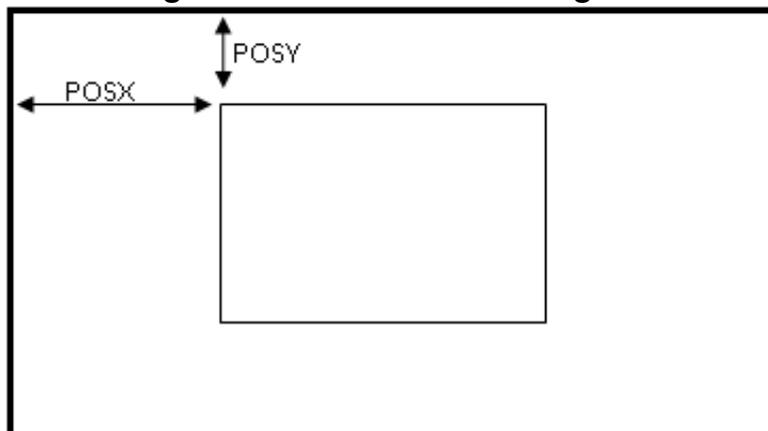
1. Configure the size of the layer using CtrlDescLn_1. The width of the graphic must conform to configuration rules related to the graphic encoding as described in the [MPC5606S Reference Manual](#).

Figure 5. CTRLDESCLn_1 Register



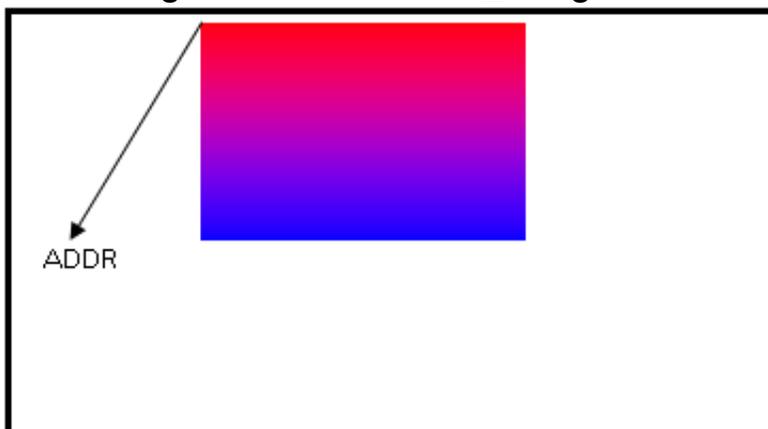
2. Place the layer on the panel using CtrlDescLn_2

Figure 6. CTRLDESCLn_2 Register



- Point to the top left-hand pixel of the graphic in memory using CtrlDescLn_3

Figure 7. CTRLDESCLn_3 Register



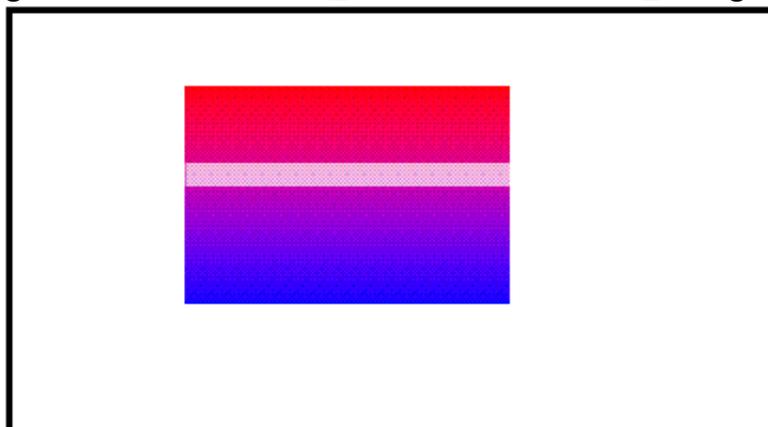
- Choose the graphic encoding, blending, and tiling and then enable it using CtrlDescLn_4

Figure 8. CTRLDESCLn_4 Register



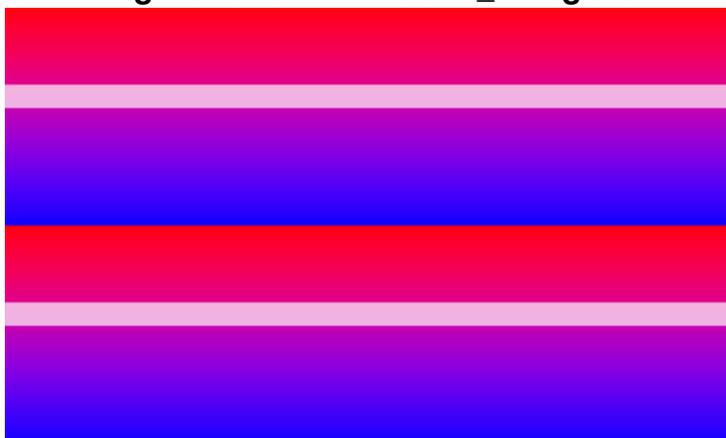
- Configure the chroma-keying ranges using CtrlDescLn_5 and CtrlDescLn_6

Figure 9. CTRLDESCLn_5 and CTRLDESCLn_6 Register



6. Configure the tile size using CtrlDescLn_7

Figure 10. CTRLDESCLn_7 Register



CtrlDescLn_4 makes multiple blending and encoding combinations available which are explored in more detail in video material available at <http://www.youtube.com/freescale>.

Errors in configuring the layers are reported in the PARR_ERR_STATUS register. Note that an error in the layer configuration registers will prevent the layer being visible. If a layer is not visible as expected check the flags in the PARR_ERR_STATUS register to see if an error has been detected.

The priority of the layers is fixed so that layer 0 is always higher priority than layer 1, which is higher than layer 2 and so on. This arrangement influences the effect of blending.

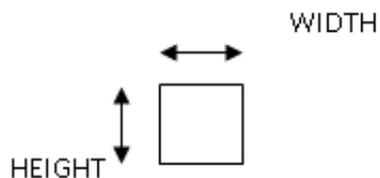
5.2 Using the cursor

The cursor is a graphic element independent from the graphic layers. It allows a single graphic to be placed on top of all the layers and does not support any blending or choice of graphic encoding. The cursor is always stored in a special area of RAM with the DCU2 module.

The steps to display the cursor are as follows:

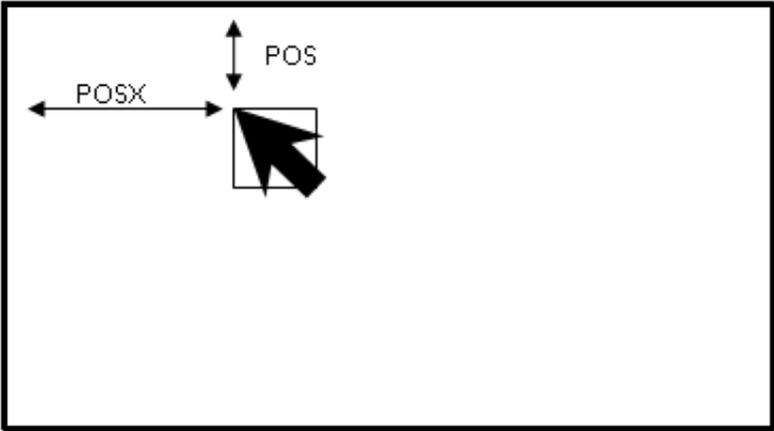
1. Configure the size of the cursor using CtrlDescCursor_1. The maximum size of the cursor is determined by the size of the internal RAM.

Figure 11. CtrlDescCursor_1 Register



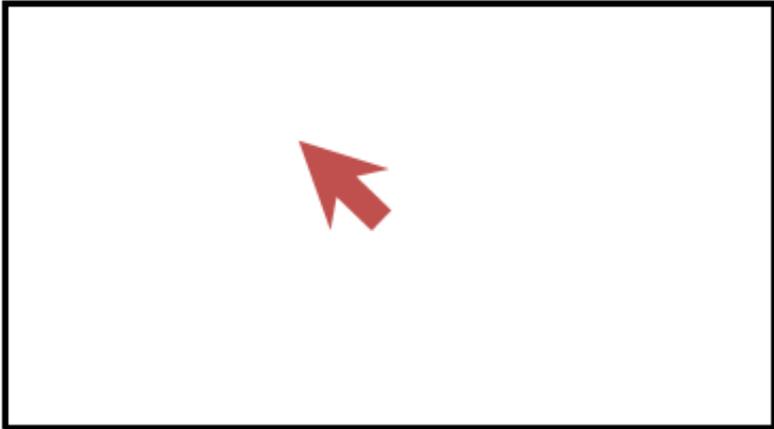
- 2. Place the layer on the panel using CtrlDescCursor_2

Figure 12. CtrlDescCursor_2 Register



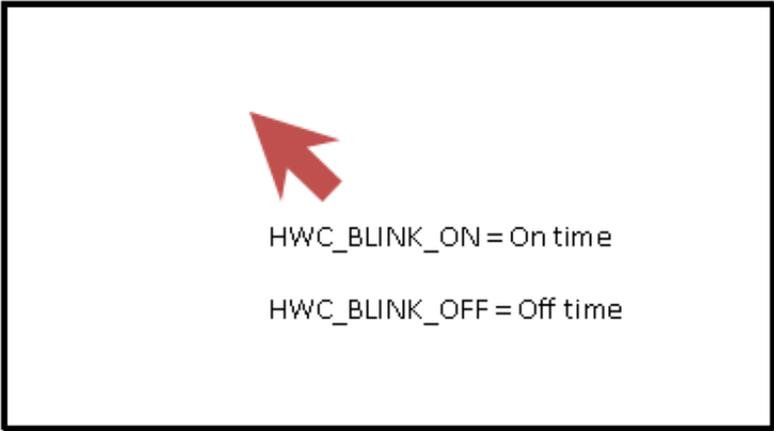
- 3. Choose the color of the cursor using CtrlDescCursor_3

Figure 13. CtrlDescCursor_3 Register



- 4. Choose the blinking options using CtrlDescCursor_4

Figure 14. CtrlDescCursor_4 Register



5.3 Using the PDI

The Parallel Data Interface is intended to accept a video stream and display it on the panel where it replaces the fixed background color. As such any video or image will always be the lowest priority content on the panel.

The PDI input has the following strict requirements for correct operation:

- Incoming video must be non-interlaced
- Incoming video must be formatted to match the panel size. For example, if the panel is QVGA (320 x 240) format then the incoming video must also be non-interlaced QVGA.
- The timing signals from the camera must match the requirements of the panel

The PDI also provides the possibility of synchronizing the DCU2 with an external reference which may be convenient if multiple video sources are in use or the DCU2 output is being post-processed. This configuration is known as PDI slave mode and uses timing signals provided by the PDI to display graphics stored internally in the DCU. Enable this mode using the PDI_SLAVE_MODE bit in the DCU_MODE register.

The PDI can accept either parallel data and timing signals similar to those produced by the DCU2 or encoded video in the format specified by the ITU in standard ITU-R BT.656. The incoming data can be up to 18 bits in width and for 16 bit input formats can be multiplexed onto 8 pins to reduce the number of pins required. This is described as narrow mode.

Configuration of the PDI input-format is performed by the PDI_MODE bitfield in the DCU_MODE register. Also this register has the bit-fields to enable narrow mode (PDI_NARROW_MODE), the order of the bytes in this mode (PDI_BYTE_REV), whether the data enable signal is used (PDI_DE_MODE) and the enable for the PDI itself (PDI_EN). DCU_MODE register also contains a control for the number of valid frames that must be received before incoming video is accepted as valid (PDI_SYNC_LOCK). The SYN_POL register also sets the expectations for the polarity of the PDI enable, clock, and synchronization signals. The status of the PDI is reported in the PDI_STATUS register.

6 Summary

This application note describes the steps to enable and use the DCU2 on the MPC5606S. The nature of graphical user interfaces is such that additional layers of graphic software and individual image designs will be required to make use of the DCU2 in a typical application. Examples of this software and design flow recommendations are available from the Freescale resource library available at www.freescale.com.

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2010 Freescale Semiconductor, Inc.