AN11365

SCT camera interface design with LPC1800 and LPC4300

Rev. 1.2 — 3 June 2014 Application note

Document information

Info	Content
Keywords	SCT SGPIO camera module LPC1800 LPC4300 DMA Cortex-M3 Cortex-M0 Cortex-M4
Abstract	This application note demonstrates the camera interface design with the SCT peripheral on the LPC1800 and LPC4300.

Revision history

Rev	Date	Description
1.2	20140603	• Added a separate project for lpc1857 with its own header and drivers • Added the hardware design doc to zipfile • Added recommendation to use debugger “with Pre-reset”
1.1	20130711	• Added LPC4300 to title.
1	20130625	• Initial version.

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

LPC1800 is the Cortex-M3 series MCU from NXP which runs up to 180 MHz and has a wide selection of advanced peripherals, like the State Configurable Timer (SCT), SPI Flash Interface (SPIFI), LCD Controller, EMC and high-speed USB. LPC4300 is the dual core series MCU from NXP which includes an industry leading Cortex-M4 core and a Cortex-M0 coprocessor both running up to 204 MHz. The State Configurable Timer (SCT) is available on the LPC1800 and the LPC4300 series MCUs. All discussions about the SCT implementation in this application note apply to both LPC4300 and the LPC1800. To simplify this discussion, we will only mention LPC1800 unless it is a LPC4300 unique feature. The sample project has been tested with both MCB1800 and MCB4300 development boards.

The SCT interface is a powerful digital configurable interface on the LPC1800 series MCU. With 16 events, 32 states, 8 inputs, and 16 outputs plus the match/capture capability, the SCT interface can easily input or output complex waveforms. In this application, we create the states and events based on the camera’s timing signals such that the SCT can sample the camera output at the correct timing.

The demonstration hardware is a Keil MCB1800 board with an add-on daughter board available from NXP. The picture of an assembled board is shown in Fig 1.

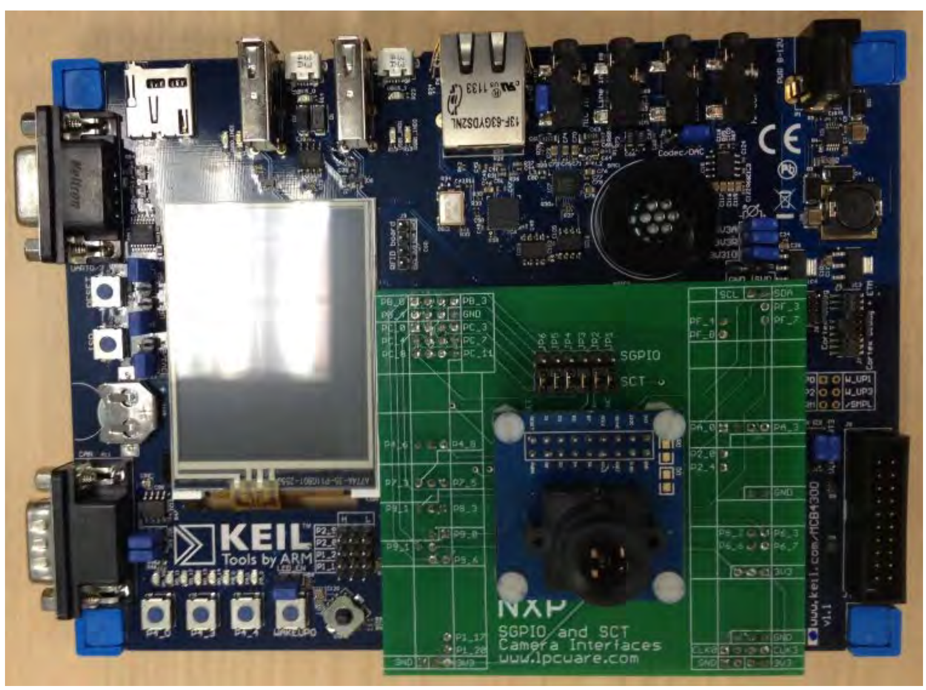

Fig 1. NXP camera module daughter board plugged on MCB1800

Some of the highlights in this camera module design include a demonstration of the flexibility of the SCT system as well as a very low bandwidth usage of the LPC1800 MCU. This application note is related to AN11343 (SGPIO camera interface design with LPC4300) in that the same camera sensor is used in both application notes with a similar CPU bandwidth usage.

2. Camera interface overview

An OmniVision OV7670 camera module is used in this demonstration but any camera with an 8-bit parallel output in RGB565 output and QVGA mode support should fit in this design. User can easily adapt the camera pin-out to reuse the hardware and software provided in this application note.

The OV7670 is controlled via I2C interface taking commands from the LPC1800. Its output is a byte-wide interface that sends out one byte per pixel clock. As each pixel is represented by 16 bits of data in an RGB565 format, it takes 2 pixel clocks from the camera to send out one pixel of image data. The camera also sends the horizontal and vertical sync pulses which determine the position of the pixel. The following figure depicts the signals sent out from the camera to the SCT interface.

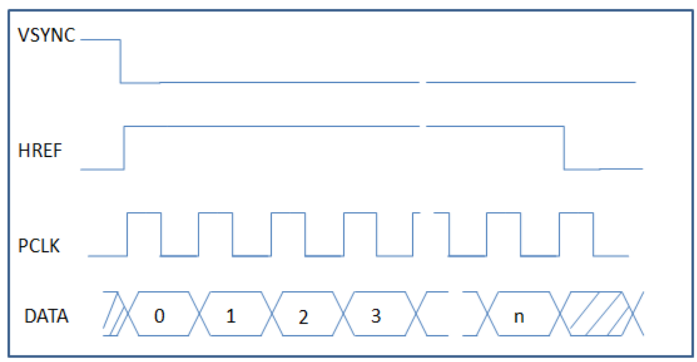

Fig 2. Camera output signals

3. SCT camera module overview

Fig 3 shows the interactions of the major system blocks in the SCT camera module. After the camera is configured through the I2C interface, the SCT block samples the video data and then transfers the data using GPDMA to the external SDRAM available on the MCB1800 development board. The LCD controller on LPC1800 then uses its dedicated DMA controller to pull the frame buffer data from SDRAM through EMC for displaying on the LCD panel available on the MCB1800 development board.

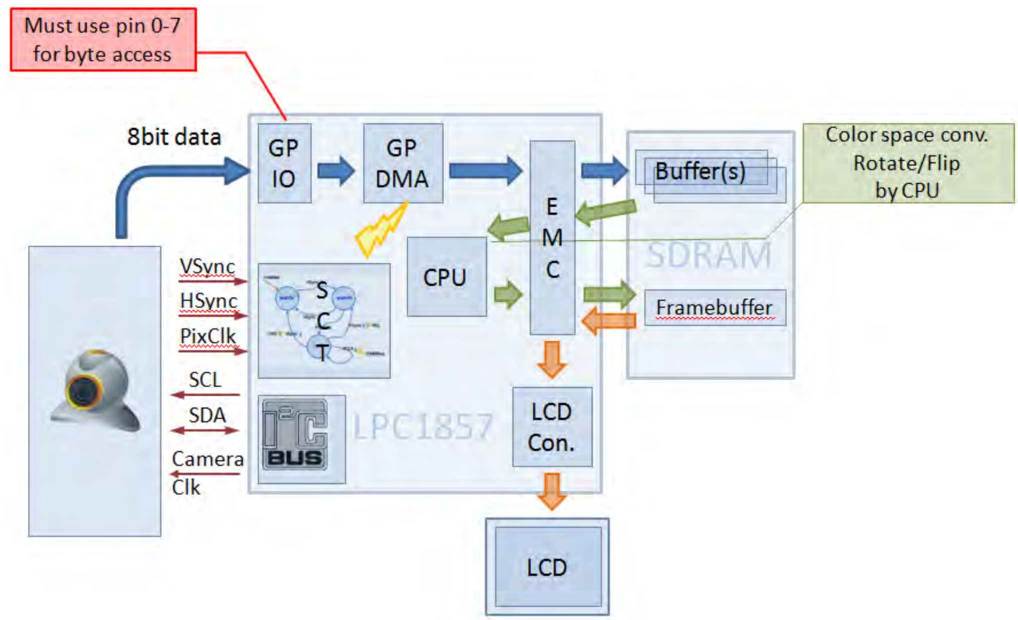

Fig 3. SCT camera interface block diagram

3.1 Hardware connections

The hardware connections between the camera module daughter board and the LPC1800 on the MCB1800 development board are shown in Fig 4.

- DATA0 to DATA7 are the 8 bit OV7670 camera data
- XCLK is the camera input clock sent from LPC1800 to OV7670
- PCLK is the output pixel clock sent from OV7670 to the SCT interface
- HREF is the horizontal reference signal output sent from OV7670 to the SCT interface
- VSYNC is the vertical synchronization signal output sent from OV7670 to the SCT interface
- RESET and PWRD are control signals sent from LPC1800 to OV7670
- SCL and SDA are the I2C command connections between the LPC1800 and OV7670

Camera pin	Video Signal	Port	Function
1	**3V3**		**3V3**
2	**GND**		**GND**
3	**SCL**	**I2C0_SCL**	**I2C0_SCL**
4	**SDA**	**I2C0_SDA**	**I2C0_SDA**
5	VSYNC	P2_4	CTIN0
6	HREF	P7_3	CTIN3
7	PCLK	P4_8	CTIN5
8	**XCLK**	**PA_0**	**CGU_OUT1**
9	DATA7	PC_8	GPIO6[7]
10	DATA6	PC_7	GPIO6[6]
11	DATA5	PC_6	GPIO6[5]
12	DATA4	PC_5	GPIO6[4]
13	DATA3	PC_4	GPIO6[3]
14	DATA2	PC_3	GPIO6[2]
15	DATA1	PC_2	GPIO6[1]
16	DATA0	PC_1	GPIO6[0]
17	**RESET**	**P8_1**	**GPIO4[1]**
18	**PWDN**	**PC_10**	**GPIO6[9]**

Fig 4. SCT camera interface connections

3.2 High level flow chart

The flow chart in Fig 5 shows the major initialization of the system and the while (1) loop in main(). Notice that during the SCT initialization, the GPDMA is configured to respond to several SCT events and thus take care of the image data transfer from the SCT inputs 0-7 to the SDRAM. Section 4 in this application note provides details on the SCT and GPDMA configuration and control.

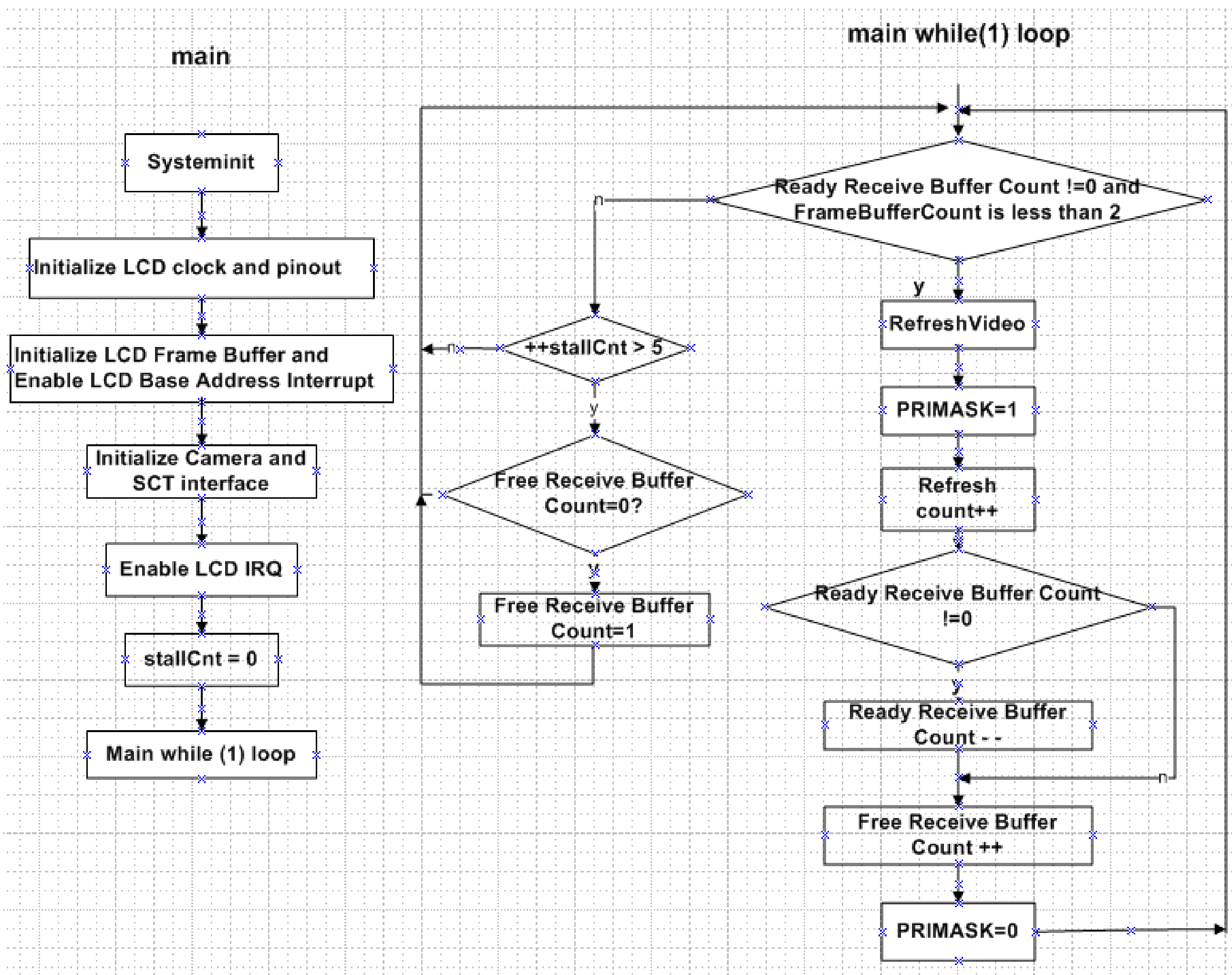

Fig 5. System initialization and main while(1) loop

3.3 Description of the contents included in this application note

In this section, the various contents included in this application note are discussed such that users can easily find their way through the project.

At the very top level, Fig 6 shows the contents included in this application note.

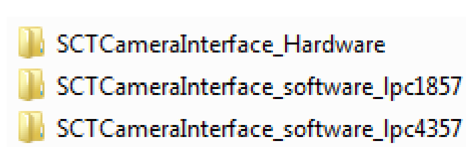

Fig 6. Top level folder contents

Subfolder \SCTCameraInterface_Hardware\ holds the schematic, Gerber file as well as the Eagle design file of the daughter board as shown in Fig 7. The MCB1800 information can be found from the Keil website: http://www.keil.com/mcb1800/

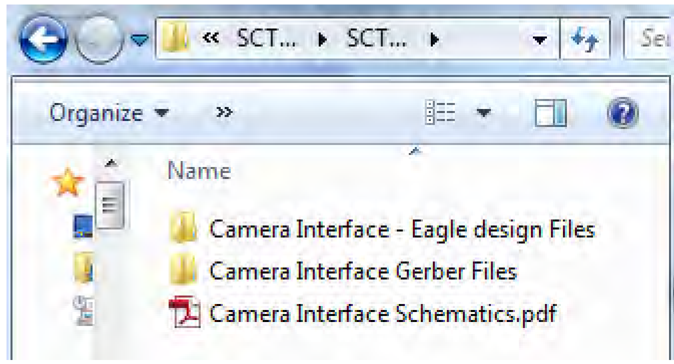

Fig 7. Hardware folder contents

Folder \SCTCameraInterface_software_lpc1857 and \SCTCameraInterface_software_lpc4357 contains the Keil project and source code for the two devices.

When using the Keil IDE for debugging the board, please select “with Pre-reset” under the “Connect & Reset Options”. This will ensure the clock signal will be reinitialized to allow proper operation.

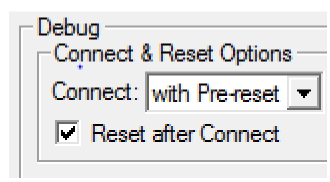

Fig 8. Debugger setup

4. SCT camera interface design

4.1 General introduction

This section details the SCT interface design for the camera interface. The SCT on LPC1800 and LPC4300 supports 16 events and 32 states (when using 32-bit timer) or 64 states (when using two 16-bit timers). The events can be generated through 8 input pins, or 16 output pins or 16 capture/match registers.

Fig 9 is the basic block diagram of the SCT interface. Please refer to the user manual for detailed description of the SCT interface.

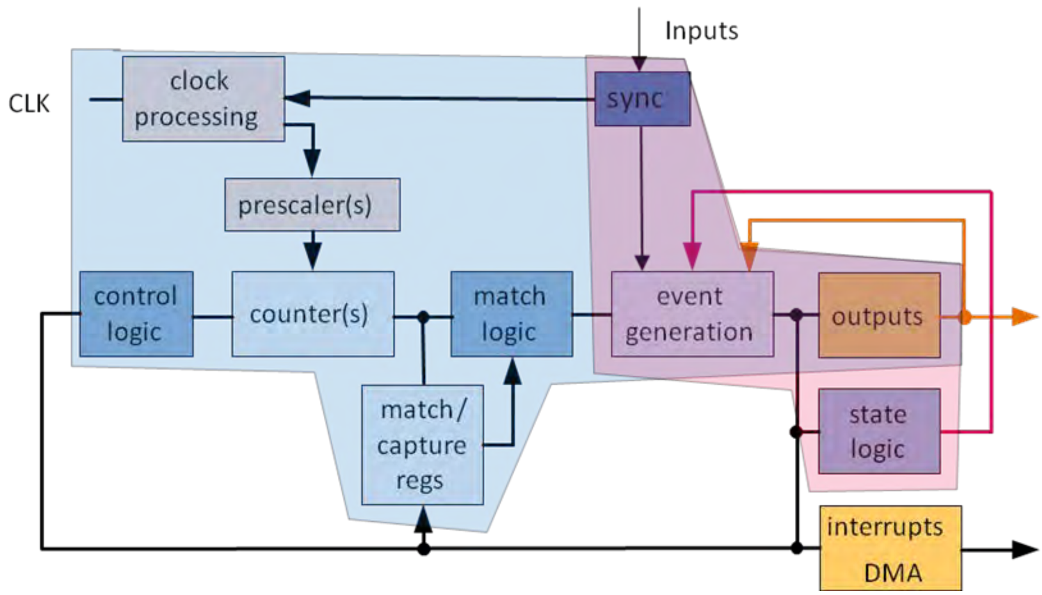

Fig 9. Architecture of the SCT interface

In this implementation, the SCT is configured to be a single 32-bit timer with 8 non-synchronized inputs for the SCT from the camera interface. The bus clock is used as the SCT and prescaler clock. Below is a snippet of the code used for SCT general configuration in function SCTInit() from source code Cam_OV7670.c.

```
LPC_SCT_Type *pSCT = LPC_SCT;
pSCT->CONFIG = 1UL<<0 | 0UL<< 1 | 0x00UL<<9;  // unify| busClk | InSync:All8
```

The other timer control registers are disabled as they are not used in this implementation.

```
pSCT->CTRL_U = 1UL<<1 | 1UL<<2 | 1UL<<3 | 0UL<<4 | 0x01<<5;
//    Stop    | Halt    | ClrCnt | !BiDir | Prsc
pSCT->LIMIT_L = 0xFFFF;     // EvtClrCnt
// event doesn't affect cnt
pSCT->HALT_L = 0, pSCT->STOP_L = 0, pSCT->START_L = 0;
```

4.2 Camera interface SCT states and transition analysis

Based on the timing diagram of the camera (Fig 2), the video signal starts with an Vertical Sync signal and sends out the image data line by line during the horizantal reference signal high period. When all the lines are transmitted, there is a vertical blanking period. Following the vertical blanking period, a new Vertical Sync signal can start a new frame.

The timing of the image tranmission and the signaling of the camera interface gives us three states to monitor in the SCT module:

1. Waiting for the next frame
   This is named “wantV” as inidicated in below state transition diagram in Fig 10. The rising edge of VSync serves as the event to trigger the SCT transit to “wantV”. This state is the initial state of the camera image collection after reset.
2. Waiting for the next line
   This is named “wantH” as indicated in below state transition diagram in Fig 10.

The falling edge of the VSync signal as well as the falling edge of the Href signal serve as the event to trigger the SCT transit to “wantH”.

3. Receiving a line of data
This is named “inH” as indicated in below state transition diagram in Fig 10. The rising edge of the Href signal triggers the SCT to transition to “inH”. Within the “inH” state, each PCLK triggers a SCT event to request the DMA transfer. At the rising edge of the VSync, the DMA is initialized to transfer the first line of the image to the SDRAM. At each falling edge of Href, the DMA transfer is initialized to transfer a new line of data to the SDRAM. Please refer to the SCT interrupt service routine SCT_IRQHandler() in Cam_OV7670.c for details of this implementation.

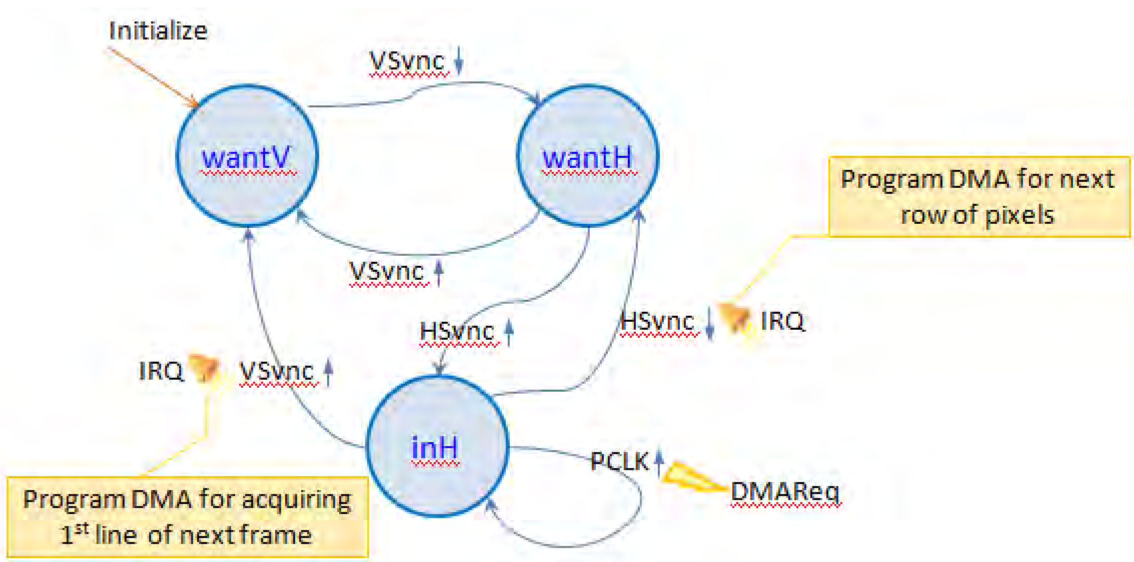

Fig 10. Camera SCT module states and events

Below is a snippet of the code used for SCT events and states definition and configuration in function SCTInit().

```
typedef enum
{
        ctst_wantV = 0,  // wait for VSYNC
        ctst_wantH = 1,  // wait for HSYNC
        ctst_inH = 2,      // in a HSYNC pulse ctst_skipV = 3,
}enum_CamTimSt;
```

As a single timer is used – UNIFY is 1 in the CONFIG register, only the _L bits in the SCT state register is used. Notice that every frame starts from state: wantV

```
pSCT->STATE_L = ctst_wantV;
```

Next the state transition events are defined.

```
typedef enum
{
        ctev_vsyncRise = 0,         // Rising edge of VSync
        ctev_vsyncRiseInSkip, // Used only if frame skip is enabled,
                               rising edge of VSync in skipped frame.
        ctev_vsyncFall,              // Falling edge of VSync
        ctev_hsyncRise,              // Rising edge of HSync
        ctev_hsyncFall,              // Falling edge of HSync
        ctev_pxclkRise,              // Rising edge of PCLK
}enum_ctev;
```

Subsequently, the event state mask register and control register of each event are configured. Please refer to SCTInit() for details on the configuration.

4.3 Specific notes for the SCT camera interface design

4.3.1 Notes on GPDMA usage

GPDMA usage is an important factor in the SCT camera interface design. It greatly relieves the CPU burden. Below are some key pointers to implement the GPDMA functionality within the SCT camera interface design.

- The GPDMA transfer type should be peripheral to memory transfer because only the GPIO (the SCT inputs) peripheral knows the amount of data generated.
- Because the video image is 8-bit parallel data, but the SDRAM is 32-bit wide, we choose GPDMA burst length as 1 for the source (SCT inputs) and 4 for the destination (SDRAM).
- As the source of the data is from the SCT inputs, the source address is not incremented after each GPDMA transfer. The destination address however needs to be incremented after each GPDMA transfer such that all image data are stored sequentially in the SDRAM.
- There are two GPDMA AHB masters. As master 1 can access memory and peripheral, but master 0 can only access memory, we use master 0 to access SDRAM and master 1 to access GPIO.

Please refer to the GPDMA initialization routine _prvInitCamDMAXferDir() in Cam_OV7670.c for details regarding the above discussion.

4.3.2 Notes on double buffering in video reception/transmission

The image collection and the LCD display utilize a double buffering scheme in this application note. The double buffering in image collection avoids data corruption of the received image being overwritten before being fully rendered on the LCD. This double buffering also provides optimized LCD display frame rate as the image collection and LCD frame rendering can happen in a parallel manner.

The SCT interrupt responds to every HRef falling edge as well as the VSync rising edge. When the SCT interrupt indicates a VSync rising edge, the image collection buffer is toggled between the two buffers. At the LCD frame rendering time, the image buffer that is not being currently updated is utilized such that LCD tearing can be avoided in case the image collection process is faster than LCD frame rendering. The LCD display loop is handled in the while(1) loop in main. The LCD base address interrupt is enabled to allow switching between the two LCD frame buffers. Please refer to the LCD interrupt routine “LCD_IRQHandler()” for details on this implementation.

5. Conclusion

The State Configurable Timer (SCT) interface on the LPC1800 and LPC4300 series MCUs features flexible timing configuration with states and events control. These features make the SCT a perfect choice to handle state controlled input data stream like the camera interface design demonstrated in this application note. With the help of the GPDMA, this camera interface uses only 8 % of the CPU bandwidth with a 180 MHz Cortex-M3 core in the LPC1800 series MCU. Users are encouraged to port this implementation to their custom application without the cost of a dedicated camera interface. The SCT interface is a truly value added peripheral to NXP’s MCU families. With more NXP parts providing the SCT interface, users can greatly simplify their complicated applications in power conversion, lighting, motor control, and audio applications, etc.

6. Legal information

6.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

6.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

7. List of figures

8. Contents

Date of release: 3 June 2014

Document identifier: AN11365