AN11343

SGPIO camera module design using LPC4300

Rev. 1 — 6 June 2013 Application note

Document information

Info	Content
Keywords	SGPIO Camera Module LPC4300 DMA Cortex-M0 Cortex-M4
Abstract	This application note demonstrates the camera module design using the SGPIO interface on LPC4300. Several unique implementations in this application note highlight the advanced features on LPC4300.

Revision history

Rev	Date	Description
1	20130606	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

LPC4300 is the dual core MCU series from NXP which has an industry leading 204 MHz Cortex-M4 core with a Cortex-M0 coprocessor which also runs up to 204 MHz. With the Cortex-M0 capable of handling all the peripherals, this dual core architecture makes handling complicated systems much easier and more efficient.

SGPIO is a digitally configurable interface on the LPC4300. It is perfect for emulating serial or parallel interfaces to connect to external devices. In this application note, the SGPIO is used to emulate an 8-bit parallel interface to communicate with an external camera for video streaming.

The camera used in this application note is an OmniVision OV7670. The demonstration hardware includes a Keil MCB4300 development board as well as an NXP daughter board to plug onto the MCB4300. See Fig 1 for the assembled system. The daughter board's schematic and design files are included in this application note.

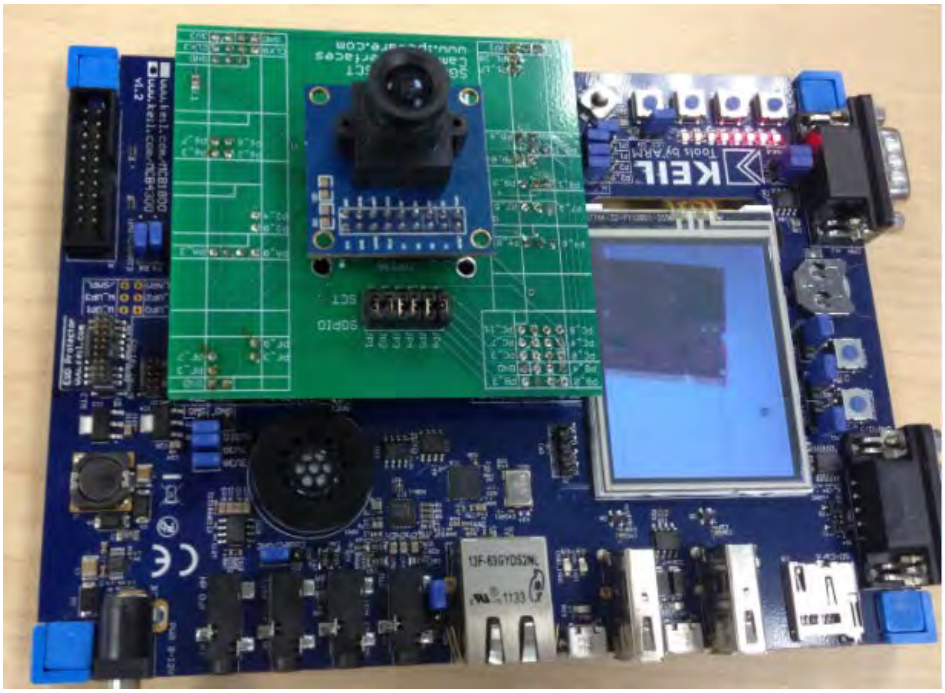

Fig 1. NXP camera module daughter plugged on MCB4300

Some of the highlights in this camera module design include a very low bandwidth usage of the LPC4300 MCU as well as serving as a good example of utilizing the dual core feature of the LPC4300. In addition, a comprehensive LCD driver is included in this application note that outputs the camera video to the LCD panel on the Keil MCB4300 development board.

The SGPIO video streaming is handled by the M0 core while the LCD driver is handled by the M4 core. The output format from the camera is 320x240 QVGA which is then rotated 90 degrees to display on the 240x320 LCD panel on the MCB4300 development board.

This application note is related to AN11196 (Camera interface design using SGPIO) in that the same SGPIO interface and capture swap interrupt is used in the video capture, however, the OV7670 camera module used in this application note uses a different camera sensor. More importantly, there are two major improvements in the demonstration software listed below:

- An improved LCD driver
- SGPIO image collection is handled by the Cortex-M0 in this implementation, whereas AN11196 uses the Cortex-M4 to handle SGPIO image collection

2. Camera interface overview

Camera module OV7670 from OmniVision is used in this demonstration but any camera with an 8-bit parallel output in RGB656 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 an I2C interface taking commands from LPC4300. 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 SGPIO interface.

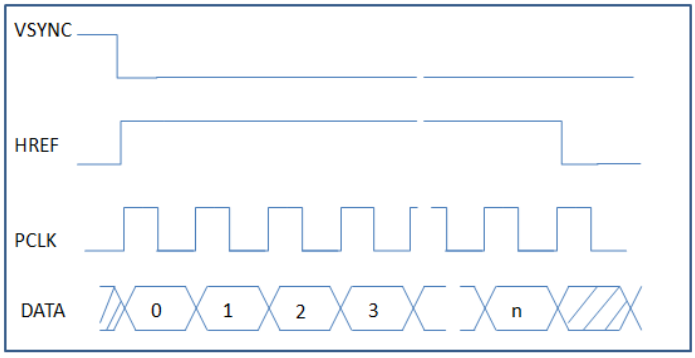

Fig 2. Camera output signals

3. SGPIO camera module overview

Fig 2 shows the interactions of the major system blocks in the SGPIO camera module. After the camera is configured through the I2C interface, the SGPIO block samples the video data and stores them in the SDRAM available on the MCB4300 development board. The LCD controller on LPC4300 has built in DMA controller to pull the frame buffer data from SDARM through EMC for displaying on the LCD panel available on the MCB4300 development board.

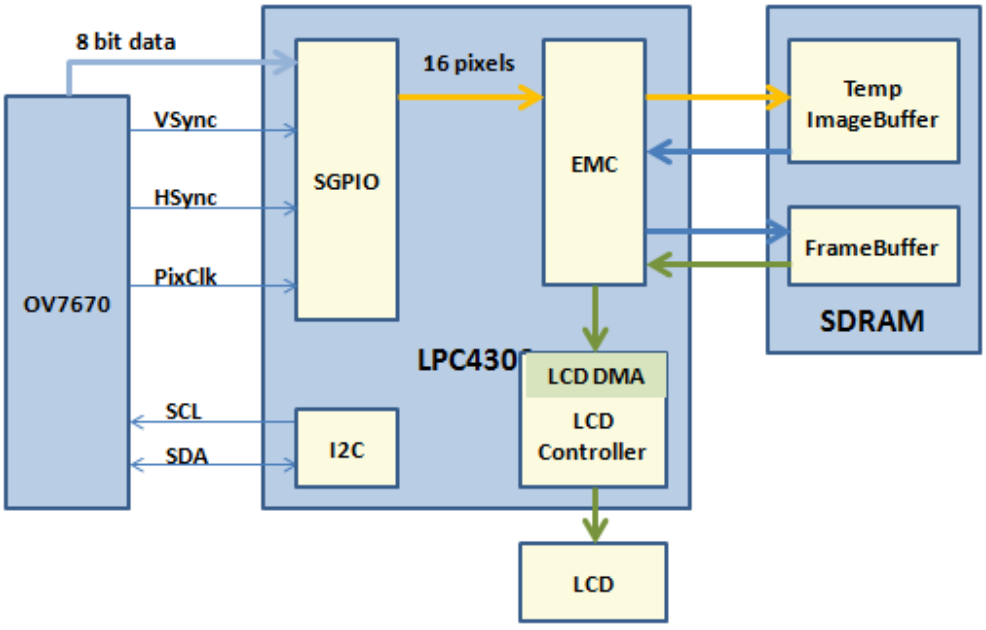

Fig 3. SGPIO camera interface block diagram

3.1 Hardware connections

The hardware connection between the camera module daughter board and LPC4300 on the MCB4300 development board is shown in Fig 4.

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	P1_20	SGPIO13
6	HREF	P8_1	SGPIO9
7	PCLK	P9_6	SGPIO8
8	XCLK	PA_0	CGU_OUT1
9	DATA7	PF_8	SGPIO7
10	DATA6	PF_7	SGPIO6
11	DATA5	P6_6	SGPIO5
12	DATA4	P6_3	SGPIO4
13	DATA3	P9_5	SGPIO3
14	DATA2	P9_2	SGPIO2
15	DATA1	P9_1	SGPIO1
16	DATA0	P9_0	SGPIO0
17	RESET	P1_17	GPIO0[12]
18	PWDN	PC_10	GPIO6[9]

Fig 4. SGPIO camera interface connection

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 the SGPIO initialization is done by the M4 in the camera initialization routine. The M0 only needs to respond to the SGPIO capture swap interrupt routine. After the M0 is released from reset, the SGPIO is ready for video collection. Section 4 of this application note contains more details about the SGPIO camera interface design.

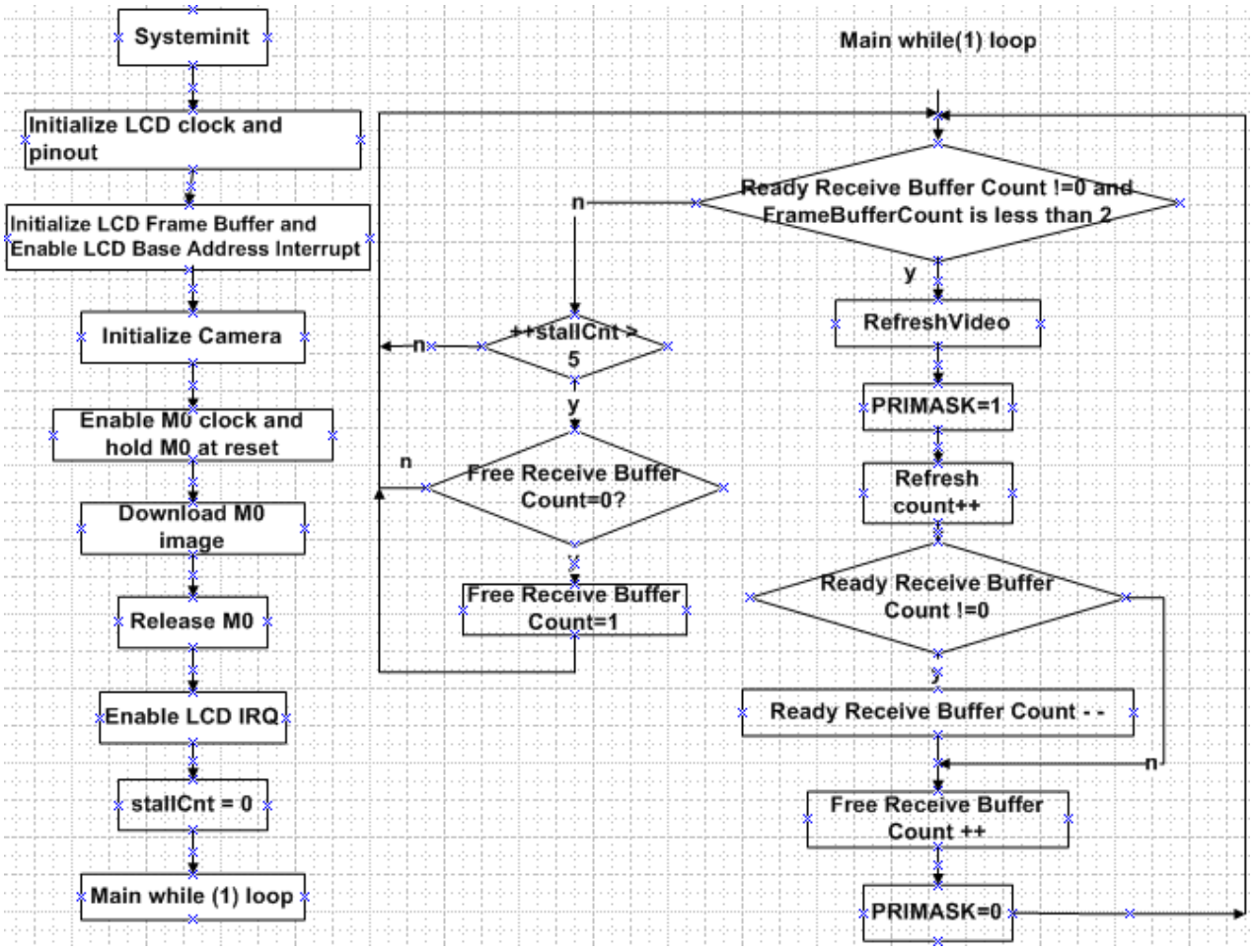

Fig 5. System initialization and main while(1) loop

3.3 Double buffering feature for image collection and LCD frame rendering

The image collection and the LCD display utilizes double buffering scheme in this application note. The double buffering in image collection avoids data corruption of received image being over written 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.

At each vertical sync interrupt, 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 as indicated in the process block “Initialize LCD Frame Buffer and Enable LCD Base Address Interrupt” in the initialization flow chart. Please refer to the LCD interrupt routine “LCD_IRQHandler() “ for details on this implementation.

3.4 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.

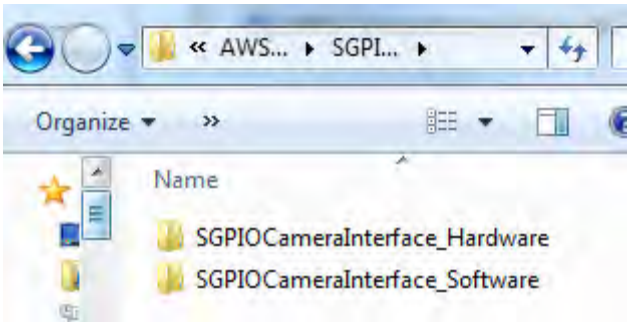

Fig 6. Top level folder contents

Subfolder \SGPIOCameraInterface_Hardware\ holds the schematic, Gerber file, as well as the Eagle design file of the daughter board as shown in Fig 7.

For more information on the MCB4300, refer to the Keil website: http://www.keil.com/mcb4300/

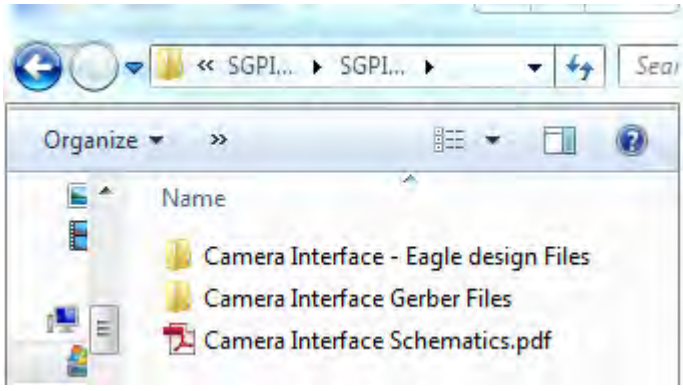

Fig 7. Hardware folder contents

Folder \SGPIOCameraInterface_Software holds the Keil projects for the Cortex-M0 core and Cortex-M4 core that demonstrate the camera interface based on the hardware described above. The M0 project should be compiled first as it generates the binary for M0 which is loaded by the M4 code into RAM.

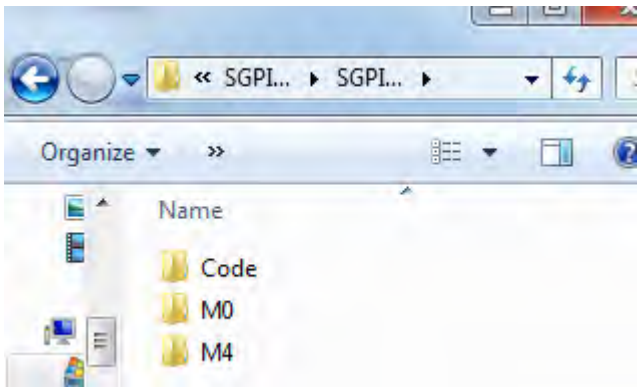

Fig 8. Software folder contents

4. SGPIO camera interface design

This section details the SGPIO design for the camera interface. The SGPIO interface is made of 16 slices with 16 SGPIO pins. The architecture within each slice is shown in Fig 9.

The double buffering and concatenation capabilities available in the SGPIO interface are the main features that help to relieve CPU usage when inputting or outputting data through SGPIO. These two features are key to the success of this camera interface implementation. The double buffering for SGPIO slice is represented with the 32-bit REG Register (shift register) and its 32-bit REGSS (shadow register). The concatenation enables a slice to input data from other slices, namely the slice that is concatenated with this particular slice.

Section 4.1 through 4.3 details the SGPIO slice design, configuration and the interrupt handling.

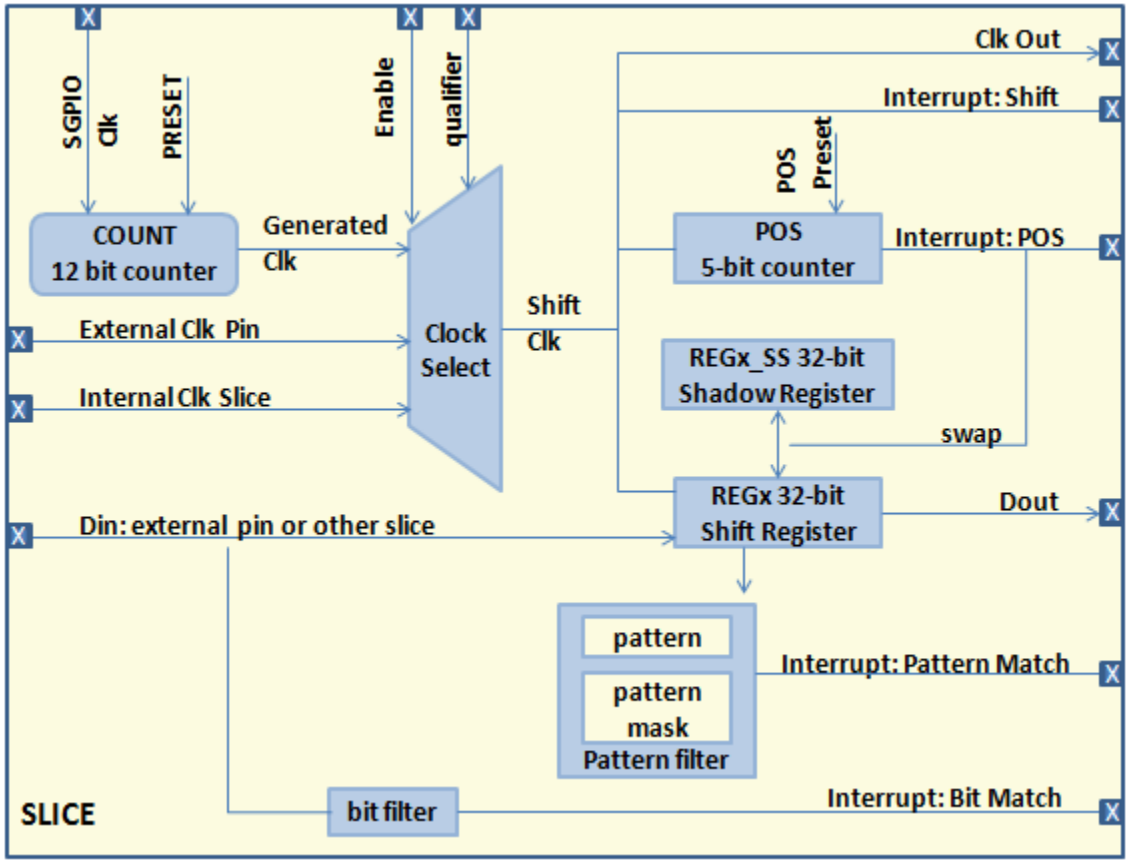

Fig 9. Architecture of a slice

For more information regarding the fundamental operation of SGPIO, please refer to LPC4300 user manual as well as AN11275 (SGPIO on the LPC4300). The link for the LPC4300 user manual is provided here for your reference:

http://www.lpcware.com/content/nxpfile/um10503-lpc43xx-user-manual

The link for AN11275 is provided here for your reference:

http://www.lpcware.com/content/nxpfile/an11275-sgpio-lpc4300-emulating-pwm-sgpio

4.1 SGPIO slice design and analysis for the camera interface

Based on Fig 2, we assign different SGPIO pins to take care of the different input signals from the camera.

As the data bus is byte width, we need to have an 8-bit parallel input bus from the SGPIO for the data input from the camera. As the data input is gated by the pixel clock, we will use the pixel clock from the camera as the shift clock for the data input.

In this example, 8 slices are concatenated to create a data buffer of 256 bits. As the data lane is 8 bits wide, we configure all 8 slices in parallel output byte lane mode. Since each pixel has 16 bits, these 8 slices can hold 16 pixels. This means 16 pixels can come in before the processor has to deal with the data, giving the CPU more time to copy data into SDRAM.

Since only SGPIO8 to 11 can work as shift clock from external, we cannot use these SGPIO in the 8 slice concatenation group. Therefore, the only choice in this design for the 8 slice concatenation is SGPIO0 to 7. In addition, only SGPIO8 to 11 can work as shift clock qualifier input. We choose SGPIO8 to input pixel clock as shift clock for the 8 concatenated slices. Horizontal sync pulse is taken in from SGPIO9 as a slice to qualify the pixel (shift) clock on SGPIO8.

The above analysis resulted in the following SGPIO slice and pin selection shown in Fig 10.

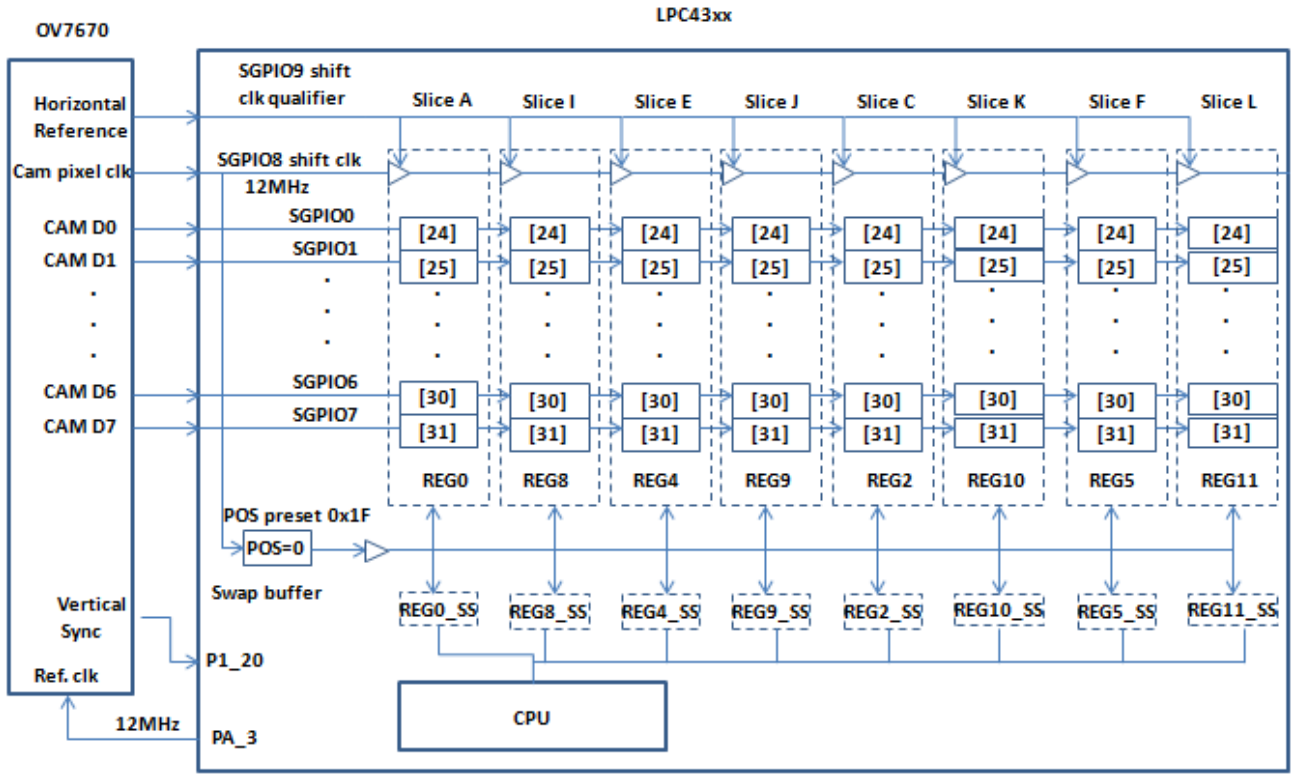

Fig 10. Operation of the SGPIO interface

Notice that a 12 MHz clock is provided from LPC4300 through PA_3 to the camera for its reference clock of the QVGA video.

4.2 SGPIO configuration

Table 1 and Table 2 summarize the SGPIO configuration of the 8 video data input slices. Notice that Slice A's input data is different from Slice I, E, J, C, K, F, L. Slice A takes input data from external pin whereas the other 7 data slices take input data from concatenated slices.

Table 1. Video data input slice I, E, J, C, K, F, L configuration

Register	Slice I, E, J, C, K, F, L	Function description
SLICE_MUX_CFG	1<<2 | 3<<6	(1)Use external clock (2)Shift 8 bits per clock
SGPIO_MUX_CFG	1<<0 | 3<<5 | 1<<7 | 1<<11 | 3<<12	(1)Use external clock from a pin (2)Use SGPIO as qualifier (3)Enable 8 slice concatenation (4)Input data from concatenated slice

Table 2. Video data input slice A configuration

Register	Slice A	Function description
SLICE_MUX_CFG	1<<2 | 3<<6	Same as Slice I, E, J, C, K, F, L (1)Use external clock (2)Shift 8 bits per clock
SGPIO_MUX_CFG	1<<0 | 3<<5 | 1<<7 | 0<<11 | 3<<12	(1)Use external clock from a pin (2)Use SGPIO as qualifier (3)Enable 8 slice concatenation (4)Input data from external pin

4.3 SGPIO interrupt handling with M0 core

The major involvement of the M0 core in this application is to handle the SGPIO interrupt. When the M0 core comes out of reset, it first initialize the inter-process communication (IPC) then it just seats idle when not handling the SGPIO interrupts.

The capture interrupt is enabled when the POS counter reaches 0 and swaps the shift register and the shadow register. This interrupt is used to collect every 16 pixels from the camera in each interrupt service routine. For detailed implementation, please refer to the SGPIO interrupt service routine SGPIO_IRQHandler() in M0Main.c.

A separate project under folder \SGPIOCameraInterface\M0 holds the project used to compile the code for M0. The 4k SRAM space starting from 0x10080000 is reserved to store the M0 core's binary code.

5. Conclusion

LPC4300 MCU series from NXP provide the best Cortex-M4 performance in the industry. The asymmetric duo core architecture allows versatile CPU load distribution within the ever increasingly complicated embedded systems. In this application note, we demonstrated the M0 core is capable of handling a camera interface with less than 5 % of CPU usage. This demonstration is not only a SGPIO camera interface design, but also a good example of distributing work load between M0 and M4. In addition, this application note showcases the LCD controller capability on the LPC4300 MCU series. The M4 core on LPC4300 MCU series has implemented an FPU unit for enhanced floating point handling and DSP functionality. Together with the digitally configurable interface like the SGPIO, the LCP4300 MCU series is ready for broad advanced embedded applications.

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. List of tables

9. Contents

For more information, visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 6 June 2013
Document identifier: AN11343