AN11315

Virtual screen implementation using emWin

Rev. 1 — 17 January 2013 Application note

Document information

Info	Content
Keywords	emWin, virtual screen, LPC4088FBD208, LPC4088FET208, LPC4088FET180, LPC4088FBD144, SPIFI, SPI Flash Interface, LCD, Q-SPI, Quad-SPI, Panning, smooth scrolling.
Abstract	This document describes the virtual-screen feature of Segger's emWin graphics library and how to implement this feature. It also demonstrates how external Quad SPI Flash memory is used to store image using LPC4088's SPIFI peripheral.

Revision history

Rev	Date	Description
1	20130117	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

This application note describes how to implement virtual screen using Segger's emWin graphics library on NXP's LPC4088 ARM Cortex-M4 microcontroller.

1.1 Project overview

The virtual-screen support of emWin can be used for panning or for switching between different video pages. If the application uses a screen which is larger than the display, the virtual screen API functions can be used to make the desired area visible.

The LPC4088 microcontroller contains the SPIFI peripheral, which can be connected to external SPI Flash or Quad SPI Flash memories. With the SPIFI peripheral, the external serial flash memory appears in the microcontroller's memory map and can be read like other on-chip memories. In the example project accompanying this application note, a large image is stored in external Quad serial flash memory instead of using precious internal flash memory space. emWin renders this image in the SDRAM as background image in advance, and displays different parts of the image at different times using the virtual-screen feature. It also displays the NXP logo in another layer in the foreground as well as a button which can be pressed to scroll the image.

1.2 System components and system setup

Fig 1 depicts the main system components required by this application. This application is developed on the Embedded Artists (EA) LPC4088 Developer's Kit. It has a Quad SPI Flash chip which stores the image data of a "rocket launch image". The SDRAM is used to store the virtual screen which contains multiple layers of images. The LCD controller loads the visible part of the virtual screen from the SDRAM and sends it to the 7 inch TFT LCD.

The software included in this application note was developed and built with Keil µVision4 IDE and Microsoft Visual C++ IDE. To make development easy, the application can be simulated on a Windows PC by compiling with Microsoft Visual C++ IDE, before building on Keil IDE and flashing into the real target board.

In Keil µVision4 IDE, this application is managed under multi-project workspace. This multi-project workspace has four projects:

- **CDL**. Contains the LPC4088 drivers.
- **BSP**. Contains EA's board specific firmware.
- **emWinlib**. Contains emWin library and configuration for virtual-screen feature and driver for EA's 7 inch LCD.
- **Project_emWinVirtualScreen**. Contains the virtual screen rocket launch example.

The final target is "LPC4088VirtualScreen". This final target needs to be flashed into EA's LPC4088 board. This above mentioned emWinlib project contains emWin library version 5.18 which includes the display driver for LPC4088 and EA's 7 inch TFT color LCD. The application loads the image data from the Quad SPI Flash and draws the entire image (which is 1400x480 pixels in size) in the SDRAM. emWin also draws another image containing the NXP logo and a button widget in the SDRAM on top of the "rocket launch image". These two are separate objects and not part of the "rocket launch image".

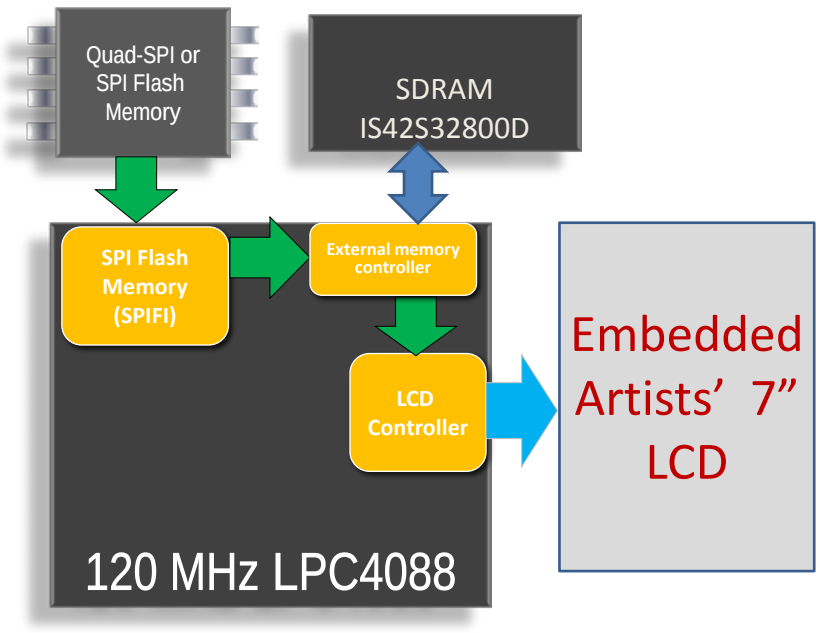

Fig 1. System components

Fig 2 shows the system setup. The LPC4088 board can be powered by connecting a mini USB connector J25 to a PC or any USB hub. Alternatively a 5 V DC adapter can be used to power EA's board and LCD after plugging it to connector J24. A Keil uLink-2, uLink-Pro or uLink ME will be needed with Keil µVision4 IDE to flash and debug the software. EA's 7 inch LCD is connected to connector J26 of the LPC4088 base board. For detail connection please check EA's LPC1788 Developer's Kit User's Guide and LCD Board - User's Guide.

Fig 2. System Setup

2. Theory of operation

A virtual screen means a display area greater than the physical size of the display. It requires additional video memory and allows instantaneous switching between different screens even on slow CPUs. If a virtual display area is configured, the visible part of the display can be changed by setting the origin.

The virtual screen support of emWin can be used for panning or for switching between different video pages.

2.1 Virtual pages

Virtual pages are a way to use the display RAM as multiple pages. If an application needs three different screens as shown in the left hand part of Fig 3, each screen can use its own page in the display RAM. In this case, the application can draw the second and the third page before they are used. After that, the application can quickly switch between the different pages using the virtual screen API functions of emWin. The right display start address needs to be passed to these functions to show the desired screen. When showing multiple screens, the virtual Y-size typically is a multiple of the display's Y-size.

2.2 Panning

If the application uses one screen which is larger than the display, the virtual screen API functions can be used to make the desired area visible. The right-hand side image of Fig 3 shows that the emWin graphics library can be used to display one part of the virtual

area on the display at a time by scrolling from one end to another end of the virtual area in X direction as well as in the Y direction.

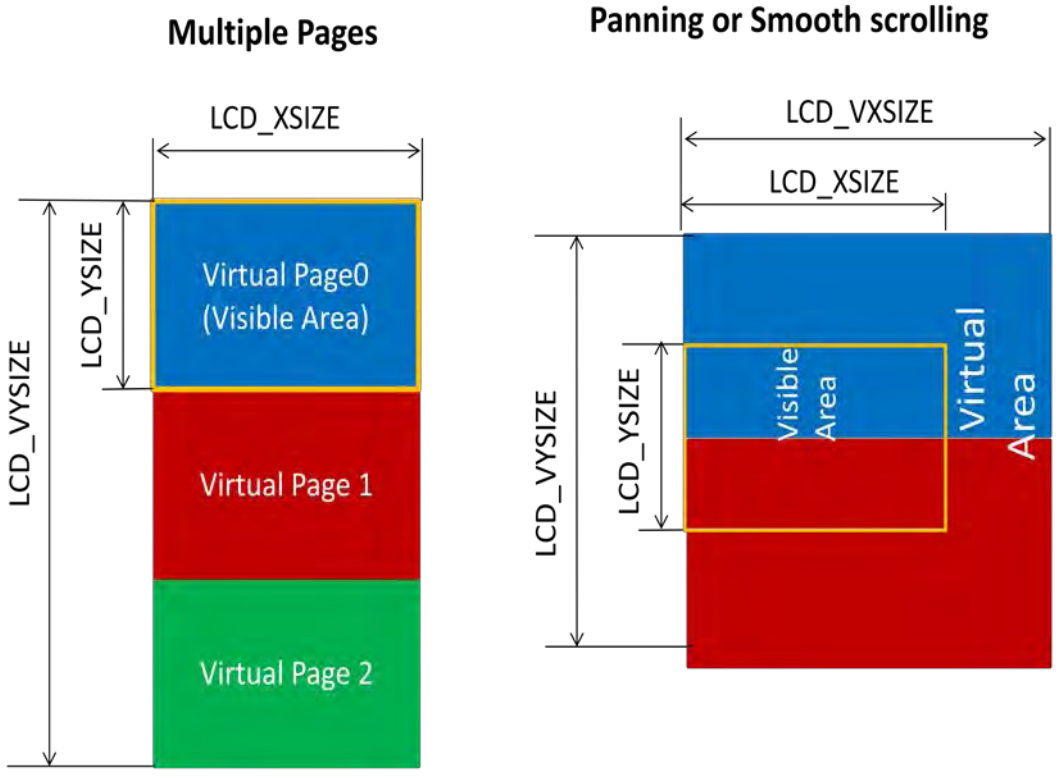

Fig 3. Virtual pages and panning

2.3 Requirements

The virtual-screen feature requires hardware with more display RAM than required for a single screen, and the ability of the hardware to change the start position of the display output. The display controller should support video RAM for the virtual area. Thankfully the LPC4088 satisfies both requirements in its hardware. On one hand, the LPC4088 has an external memory controller which supports external SRAM as well as SDRAM; on the other hand it has an LCD controller with DMA capability. The LCD controller's upper panel DMA base address register just needs to point to the correct location of the video RAM to display the desired portion of the virtual screen.

The required size of video RAM for the left hand side virtual pages scheme of Fig 3 can be calculated as:

Video RAM Size = LCD_XSIZE * LCD_YSIZE* LCD_BITSPERPIXEL / 8 *NUMS_VIRTUAL_PAGES

Or

Video RAM Size = LCD_XSIZE * LCD_VYSIZE* LCD_BITSPERPIXEL / 8

The required size of video RAM for the right hand side panning scheme of Fig 3 can be calculated as:

Video RAM Size = LCD_VXSIZE * LCD_VYSIZE* LCD_BITSPERPIXEL / 8

3. Virtual screen demo firmware

The firmware presented in this application example gives a demo of emWin's virtual-screen feature and is used to scroll an 800x1400 pixel image on a display with a resolution of 800x480 pixels.

3.1 Firmware overview

Fig 4 shows one visible area (display screen) and one virtual area. This application uses panning feature for emWin to smoothly scroll the image in Y direction.

In this example, the scrolling is done only in Y-direction. The X-size of the display and of the virtual screen is the same and fixed at 800 pixels, although we can have virtual area in X-direction as well as shown in the Fig 3.

This application uses external SDRAM as video RAM. EA's LCD display used to demonstrate the panning feature has a resolution of 800x480 pixels whereas the resolution of actual image to be displayed is 800x1400 pixels with a color depth of 16 bits per pixel. The size of the image in Y should be same as the virtual screen size in Y to fit entire image. Therefore LCD_VYSIZE should be 1400 lines.

The size of the video RAM to support the required size of the image can be calculated as follows:

Video RAM Size = LCD_XSIZE * LCD_VYSIZE * LCD_BITSPERPIXEL / 8

LCD_XSIZE = 800

LCD_VYSIZE = 1400

LCD_BITSPERPIXEL = 16

Therefore:

Video RAM Size = 800 x 1400 x 16 / 8

Video RAM Size = 2240000 Bytes

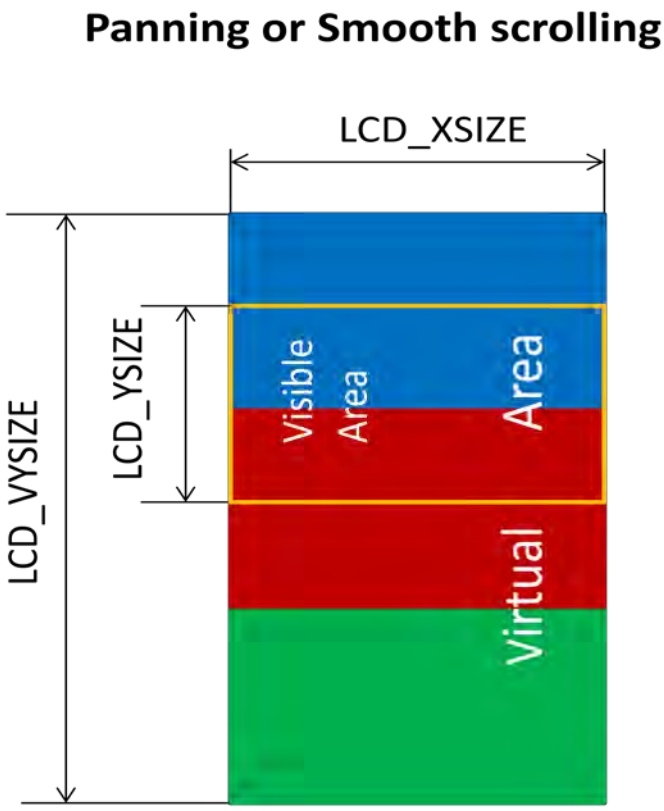

Fig 4. Panning

Another virtual screen feature requirement is display controller needs a configurable display start position. This means the display driver has a register for setting the frame buffer start address, or it has a command to set the upper left display start position. Virtual screens should be configured during the initialization. The function LCD_SetVSizeEx() is used to define the virtual screen size. It is also required to react on the command LCD_X_SETORG in the driver callback routine by setting the right frame buffer start address. In this application example display position is set using following formula:

LPC_LCD->UPBASE = VRAM_ADDR_PHYS + (y * XSIZE_PHYS * PIXEL_WIDTH);

Where:

UPBASE is the color LCD upper panel DMA base address register

VRAM_ADDR_PHYS is video RAM address

PIXEL_WIDTH = LCD_BITSPERPIXEL / 8 bytes

and *y* is the location in video ram which needs to be set in the emWin API. The value of *y* decides the frame buffer address for a particular frame which needs to be displayed on LCD display.

This application is developed on Keil µVision4 version 4.6.0.0 IDE as well as Microsoft Visual C++ IDE. As shown in the Fig 5 this application has a multi-project workspace which contains four projects. The application can be built using batch build or by single

project at a time. A linker script is used to place the “rocket launch image” data into Q-SPI Flash memory. The original image in BMP format is stored inside the image folder. emWin’s Bin2C.exe conversion utility is used to convert this image into to a C array. File RocketImage.c in folder \emWinVirtualScreen\Data\ contains this array. The final target is LPC4088VirtualScreen.

emWin API LCD_SetVSizeEx(...) is called to set the virtual screen size whereas GUI_SetOrg(…) is called to scroll the virtual screen on the 7 inch display.

As this application is built to run from internal flash and the “rocket launch image” is stored in Q-SPI Flash, both drivers need to be installed in Keil IDE before flashing target with final executable.

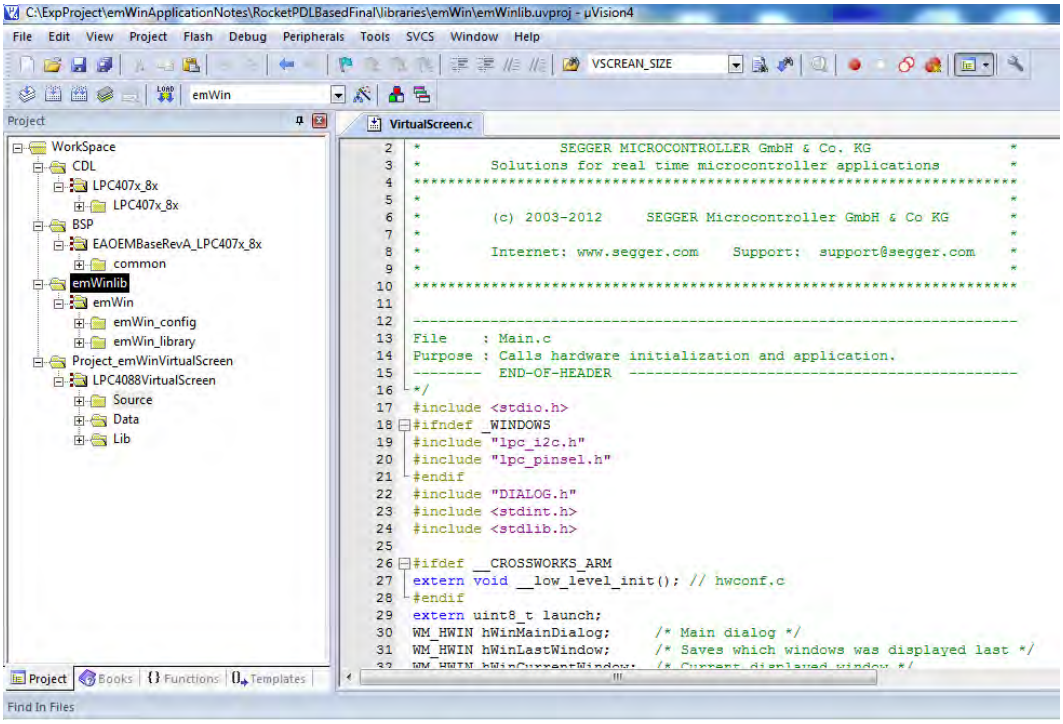

Fig 5. Multi Project WorkSpace

3.2 Expected output

In Keil IDE, build the final target *LPC4088VirtualScreen,* after setting project *Project_emWinVirtualScreen* as the active project. Programming the target board automatically programs the “rocket launch image” into the Q-SPI Flash and the firmware into the internal flash as corresponding drivers are already installed. After running this application example, a rocket image as shown in Fig 2 appears. If you press the button labeled “Press Here”, the “rocket launch image” will scroll upwards. After the entire image has been displayed, it will come back to its original position.

There is also a 10 second timer. Every 10 seconds the image scrolls automatically without having to press the button. This demo is also simulated using the Microsoft Visual C++ IDE. For the 10 second timer this application uses MSDN’s *GetTickCount(...)* function which may not be very accurate.

3.3 Conclusion

emWin's virtual-screen feature can be used to smoothly scroll an image on a display smaller than the size of the image. This feature eliminates any flickering effect as virtual screen or virtual pages are drawn before they are used. The LPC4088 SPIFI peripheral used in this application example is useful for storing images in Quad SPI Flash, hence saving precious internal flash memory.

4. Legal information

4.1 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.

4.2 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.

4.3 Trademarks

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

5. List of figures

6. Contents

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

Date of release: 17 January 2013
Document identifier: AN11315