

TN00021

LPC54608 LCD Dual Frame Buffer with Images loaded from Quad SPI flash

Rev. 1.0 — 14 August 2017

Technical Note

Document information

Info	Content
Keywords	LPC54608, LCD, Dual frame Buffer, Quad SPI, SPIFI, 16bpp, SDRAM
Abstract	This technical note gives an overview of examples that uses LCD with images loaded from Quad SPI flash.



Revision history

Rev	Date	Description
1.0	20170814	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

The LPC5460x is a family of ARM Cortex-M4 based microcontrollers for embedded applications. The LPCXpresso development board for LPC5460x MCUs is used in this technical note. See the following link for details of the board:

<http://www.nxp.com/products/microcontrollers-and-processors/arm-processors/lpc-cortex-m-mcus/lpc54000-series-cortex-m4-mcus/lpcxpresso-development-board-for-lpc5460x-mcus:OM13092>

[Figure 1](#) shows the LPC54608 LPCXpresso development board.



Fig 1. LPC54608 LPCXpresso development board

This technical note gives an overview of the example developed using the on-board LCD and Quad SPI flash via SPIFI.

2. Description

[Figure 1](#) shows the LCD with images loaded from Quad SPI through SPIFI direct memory access. The program code resides in LPC54608 internal flash and data images reside in the 16 MB Quad SPI flash.

2.1 LCD in 16BPP RGB565 with images data located in SPIFI

The example uses the on board 16MB Quad SPI flash and a 480 x 272 LCD screen. The LCD controller is configured for 16 bits per pixel (BPP) RGB565 TFT mode and implements dual frame buffers residing in on-board SDRAM. When the example is executed, a 480 x 272 image is loaded from SPI flash via SPIFI's direct access at 0x1000_0000 memory region. When a touch is detected on LCD panel, another 480 x 272 image is displayed.

The example is available in three tool chains:

- MCUXpresso IDE v10.0
- Keil MDK v5.24
- IAR Workbench v8.11

The Keil and IAR examples are located in:

lpc54608_lcd_spifi_keil_iar\boards\lpcxpresso54608\demo_apps\lcd_spifi_tft16bpp

The MCUXpresso example is in the zip file:

lpc54608_lcd_spifi_16bpp_mcux.zip

2.2 Converting Bitmap file to image file in C

Download the Bitmap converter from:

<https://www.segger.com/downloads/emwin/>

Extract and execute BmpCvt.exe. From the File menu, Open to select a BMP file. See [Figure 2](#). The LCD uses 16-bit RGB (5-bit red 6-bit green 5-bit blue) format. Converting the image to 565 reduces the real-time MCU processing time to scale down or compress the image(s).

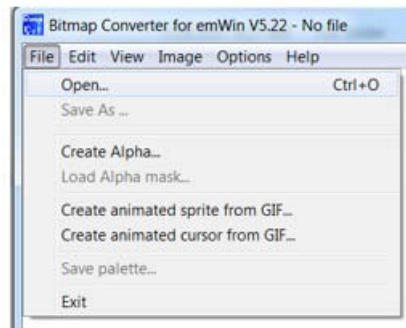


Fig 2. Selecting a BMP file

Click on the Image menu and select Scale if the resolution is larger than 480 x 272. See [Figure 4](#). Enter Width of 480 Pixels and Height of 272 Pixels. Uncheck the Keep Proportion box. See [Figure 4](#).

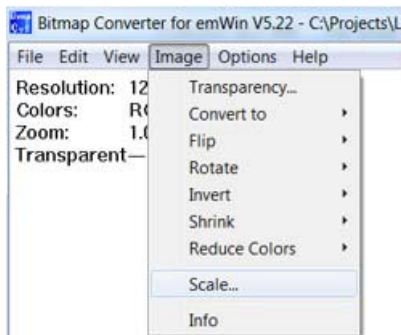


Fig 3. Selecting scale

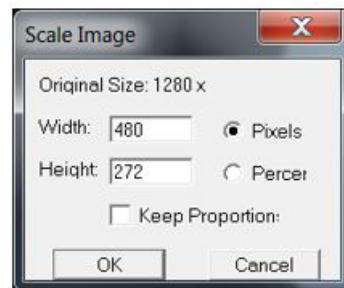


Fig 4. Scaling an image

From the File menu, select Save as menu "C" bitmap file (*.c). Click Save button and a Format specification dialog window appears. See [Figure 5](#).

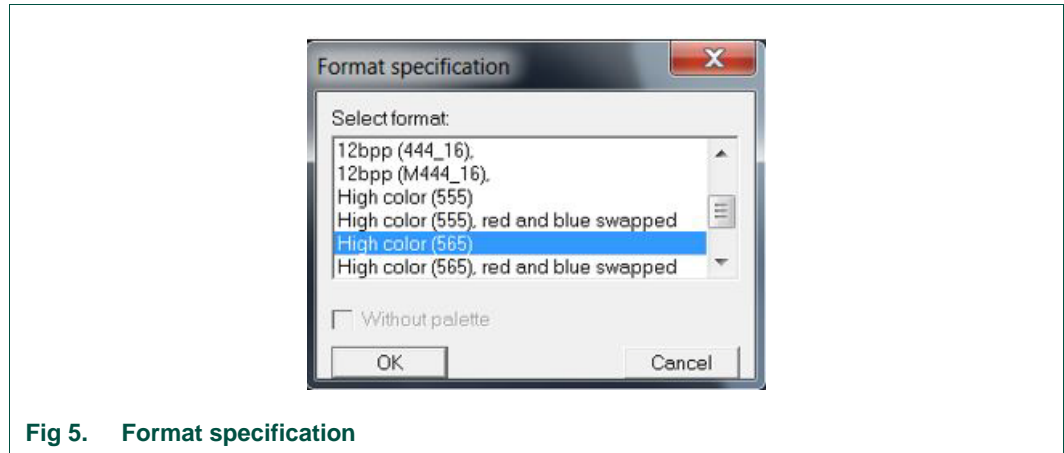


Fig 5. Format specification

2.3 Locating data in Quad SPI flash

There are multiple ways to populate data in Quad SPI flash at location 0x1000_0000. This demo uses the preprocessor directives:

- `_Pragma(address)` or `#pragma location=address` to allocate functions or variables in named section for IAR.
- For Keil `__attribute__((at(address)))` is used.
- For MCUXpresso `__RODATA(address)` is used.

The same address defined in MCUXpresso can be used several times across multiple variables, the compiler calculates and assigns the next offset. IAR and KEIL must define each address location for each variable. The example of this setup is in `images.c` and `images.h`.

Note: Preprocessor Symbol (GCC/IAR/MDK) is defined in C/C++ project option.

IAR:

```
_Pragma(location=0x10000000)

GCC_IMGIN_SPIFI static GUI_CONST_STORAGE unsigned short image1[] = {...};

_Pragma(location=0x10040000)

GCC_IMGIN_SPIFI static GUI_CONST_STORAGE unsigned short image2[] = {...};

...
```

Keil:

```
__attribute__((at(0x10000000)))

GCC_IMGIN_SPIFI static GUI_CONST_STORAGE unsigned short image1[] = {...};

__attribute__((at(0x10040000)))

GCC_IMGIN_SPIFI static GUI_CONST_STORAGE unsigned short image2[] = {...};

...
```

MCUXpresso:

```
__RODATA(0x10000000) GCC_IMGIN_SPIFI static GUI_CONST_STORAGE unsigned short image1[]
    = { ... };

__RODATA(0x10000000) GCC_IMGIN_SPIFI static GUI_CONST_STORAGE unsigned short image2[]
    = { ... };
```

2.4 Notes

2.4.1 IAR reset type and macro setup:

When “Verify download” is checked, change reset type and provide macro file (FlashLPC546x_spifi.mac must be used for Quad SPI and SPIFI to verify correctly). [Figure 6](#) shows the macro file provided with “Use macro file(s)” checkbox checked.

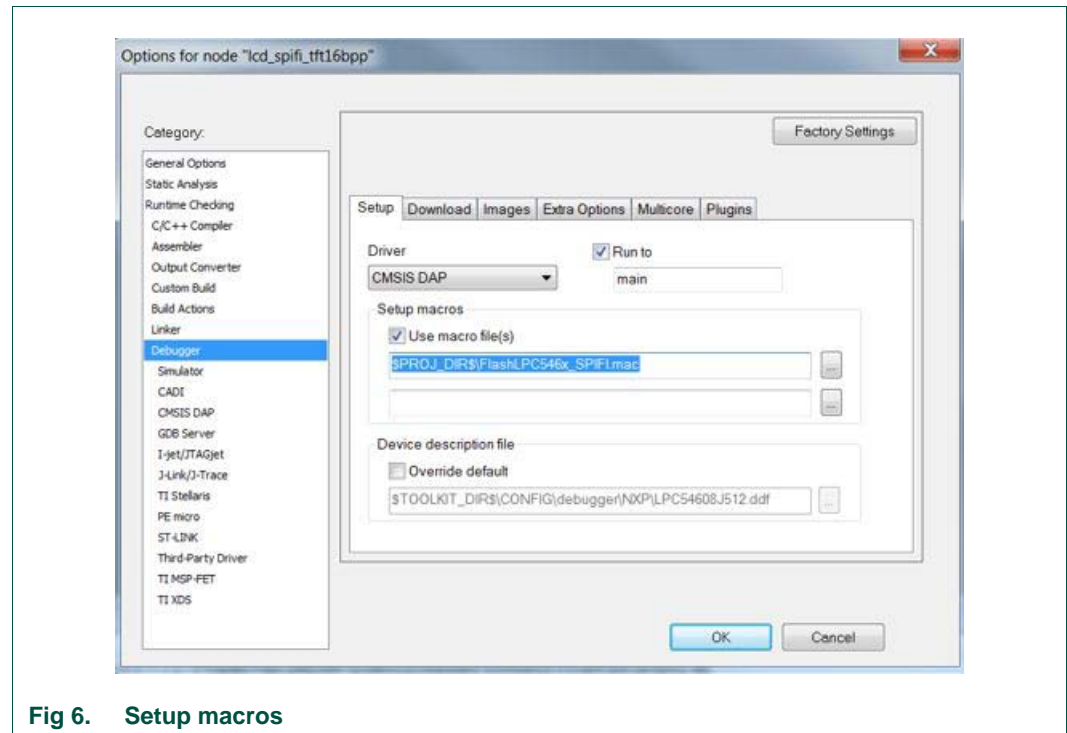


Fig 6. Setup macros

For CMSIS-DAP, change the reset type to Hardware. See [Figure 6](#).

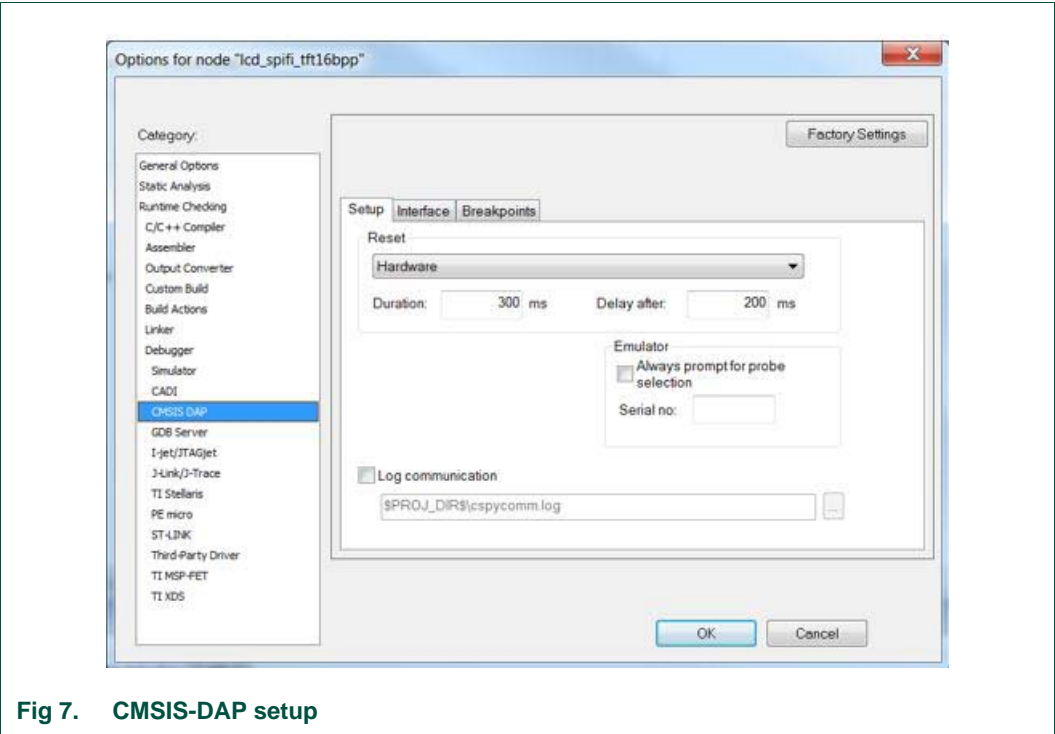


Fig 7. CMSIS-DAP setup

For JLink, change the reset type to Connect during reset. See [Figure 8](#).

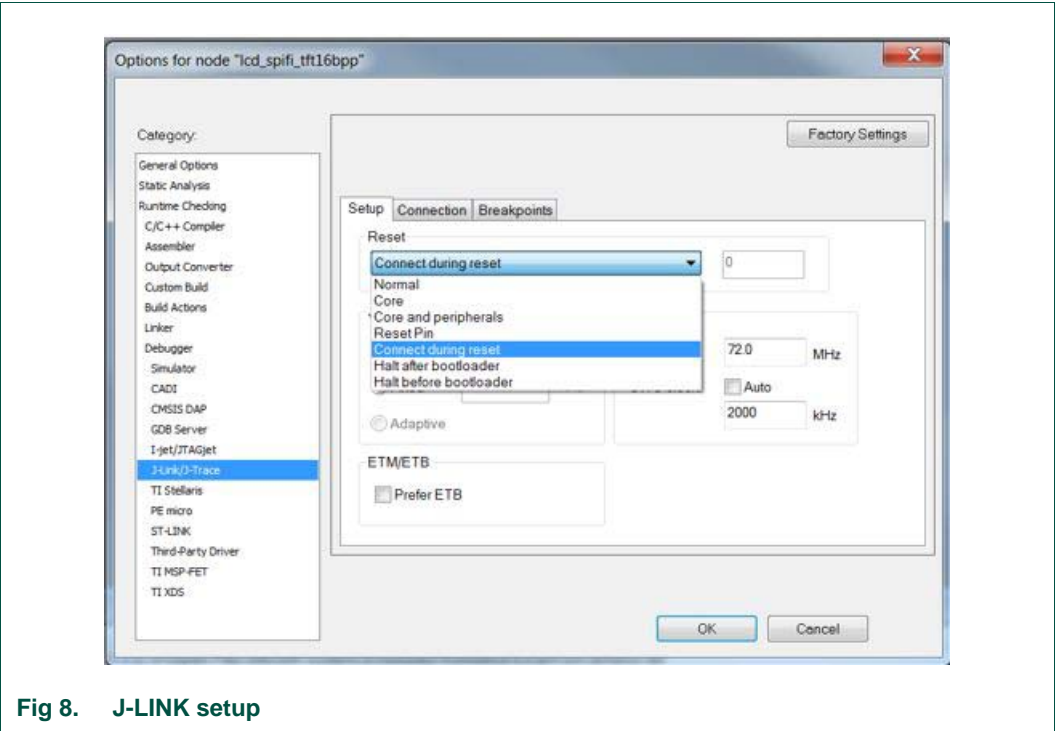


Fig 8. J-LINK setup

2.4.2 Adding 16MB Quad SPI flash to Flash programming in Keil

From the Flash menu, select Configure Flash Tools to bring up flash configuration. See [Figure 9](#).



Fig 9. Flash configuration

Under the Utilities tab, click Settings button. See [Figure 10](#). This brings up the Cortex-M Target Driver Setup dialog box. See [Figure 11](#).

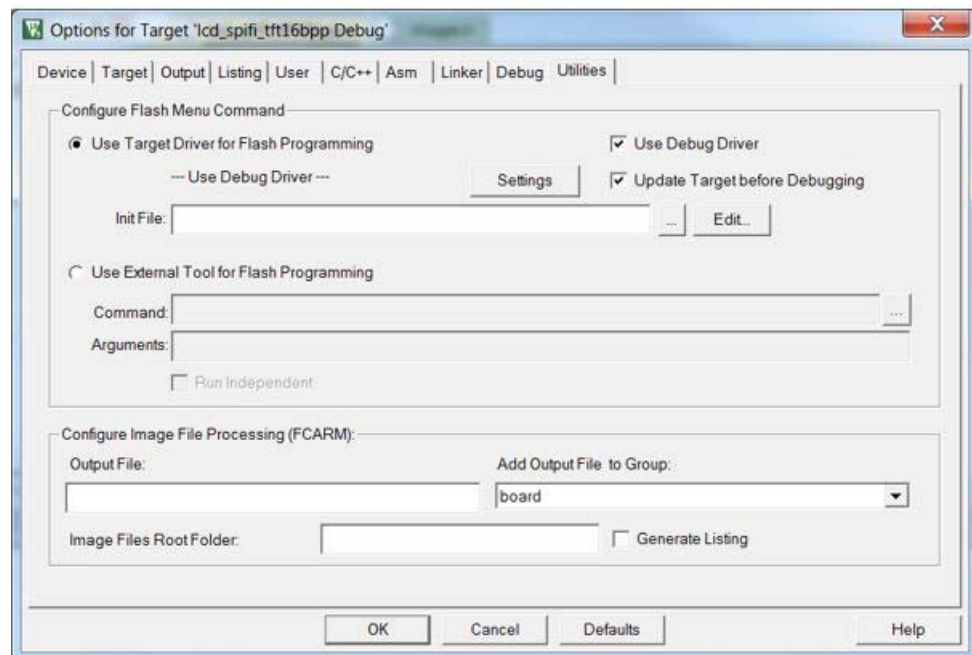


Fig 10. Debugger settings

Click on Add button to bring up flash programming algorithm selection.

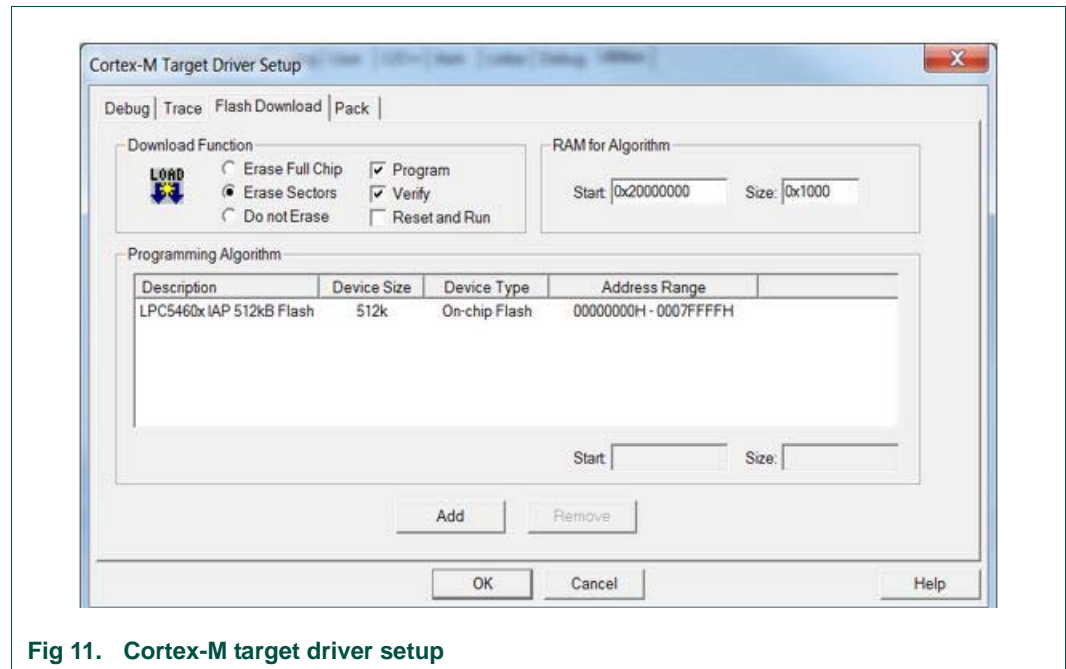


Fig 11. Cortex-M target driver setup

Select LPC5460x MT25QL128 SPIFI. Click Add button to continue. See [Figure 12](#). MT25QL128 SPIFI is added, click OK to continue. See

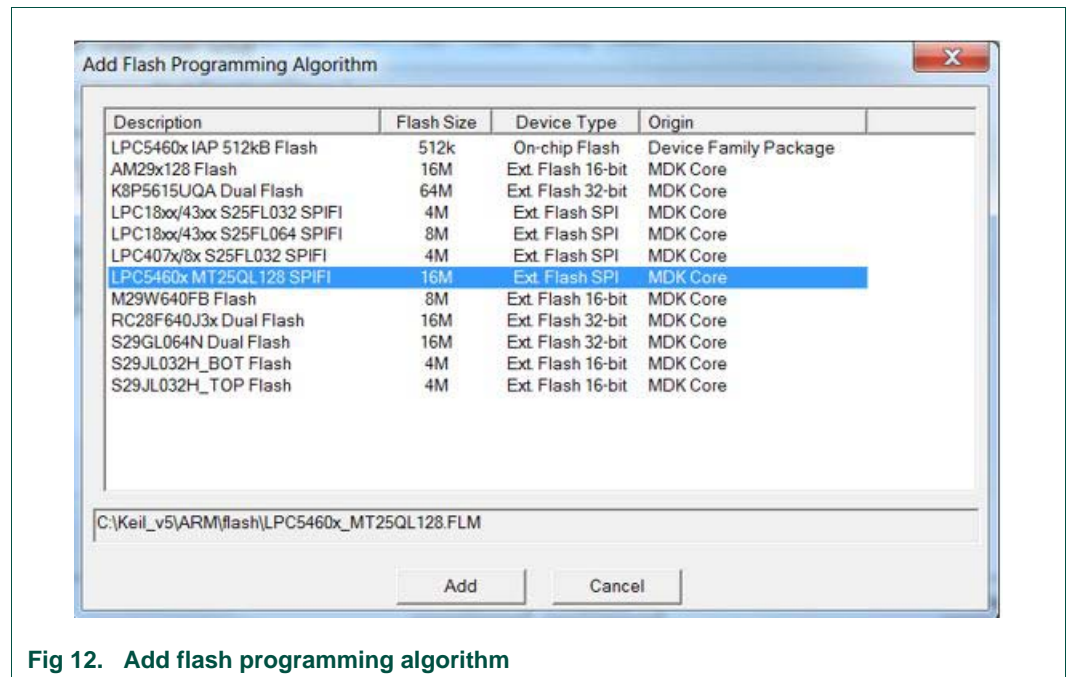


Fig 12. Add flash programming algorithm

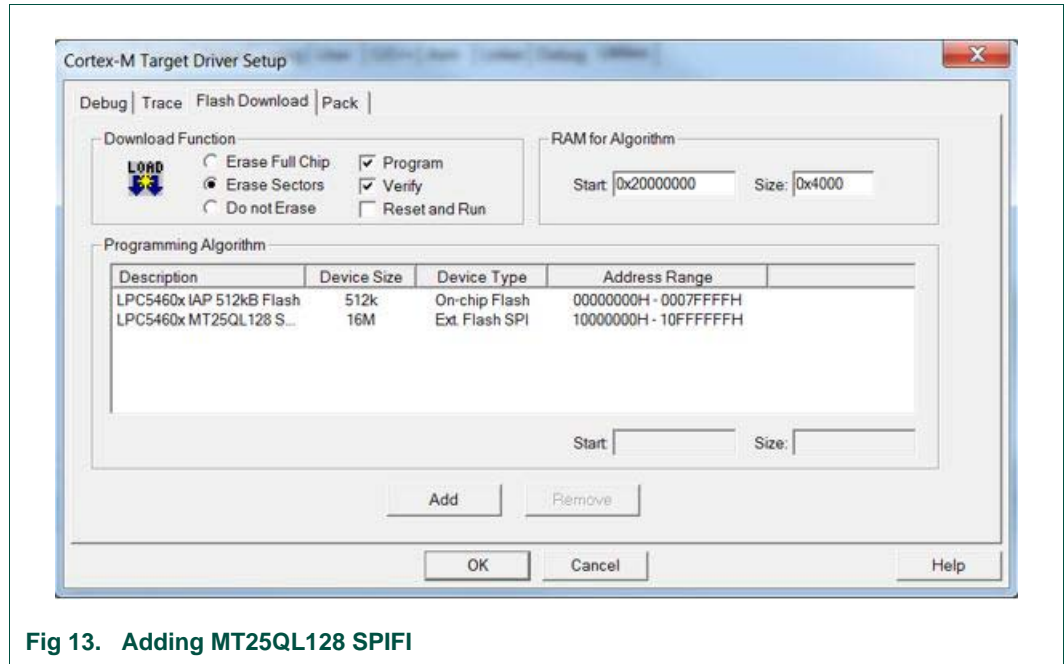


Fig 13. Adding MT25QL128 SPIFI

2.4.3 MCUXpresso project bring up

Create a new workspace by providing a name to MCUXpresso workspace.

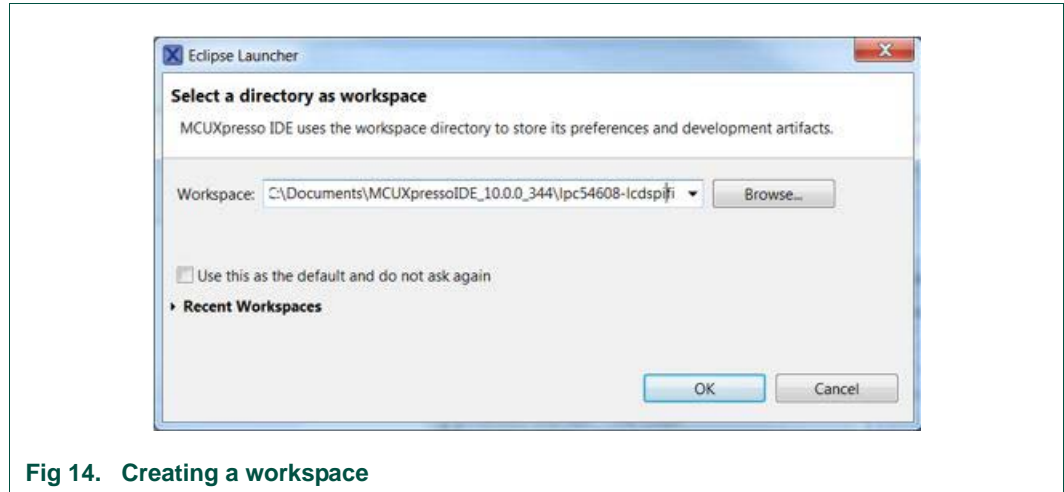
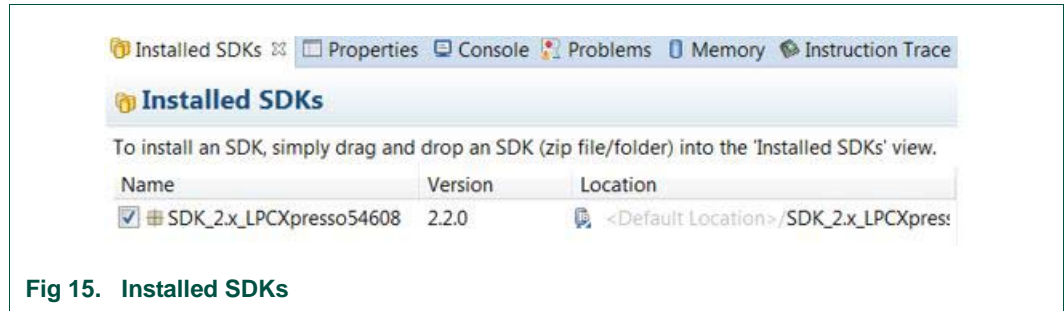


Fig 14. Creating a workspace

Download SDK_2.x_LPCXpresso54608 from:

<https://mcuxpresso.nxp.com/en/welcome>

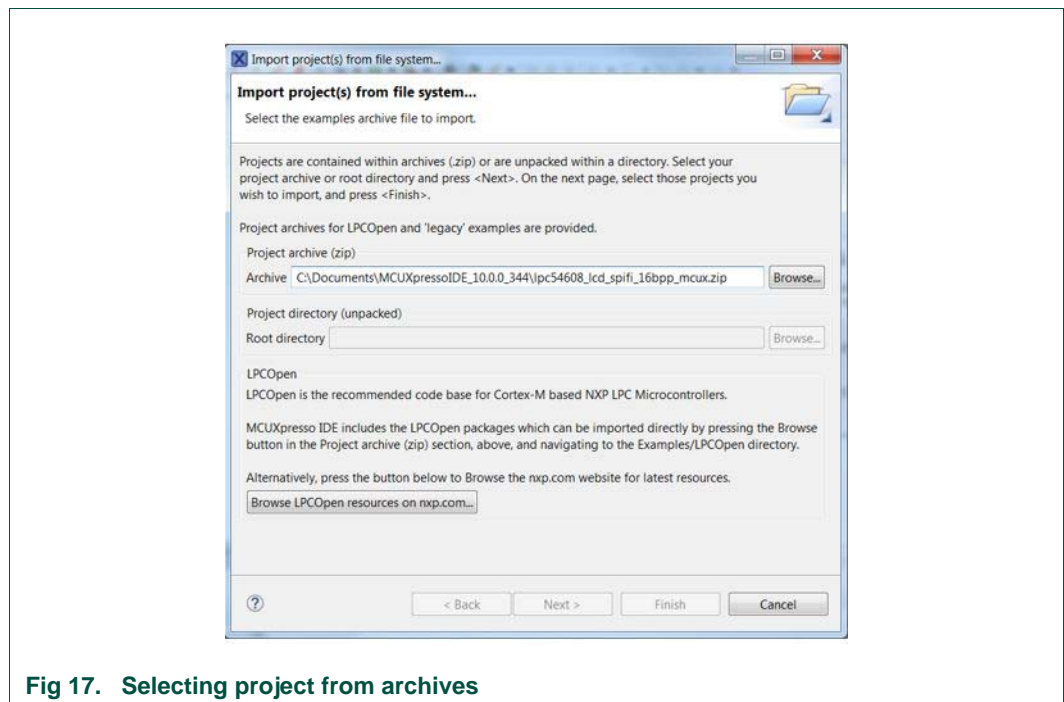
Select Installed SDKs tab in MCUXpresso to make the tab active. Drag and drop the SDK zip file to the tab. Make sure the check box in front of the SDK package is checked.



Click on Import project(s) from file system at bottom left panel.



Locate and select the lpc54608_lcd_spifi_16bpp_mcux.zip file.



2.5 Issues

2.5.1 IAR linker warning message

```
Warning[Lt009]: Inconsistent wchar_t size  
board.o and 36 other objects have wchar_t size 16 bits  
GUI_DrawBitmap_565.o(libemWin_M4F.a) and 128 other objects have wchar_t size 32 bit
```

There is an update to wchar_t size from 16 bits to 32 bits in IAR version 8.11. The library used to compile in version 8.10 or older sets wchar_t size to 16-bit. A release note from IAR can be found at the link:

<https://netstorage.iar.com/SuppDB/Public/UPDINFO/012120/arm/doc/infocenter/iccarm.ENU.html>

Remark: An update to the fsl_power_lib.a library to use 32 bits wchar_t size will be available at the next SDK release.

2.5.2 Keil flash programming progress bar delay

When downloading or debugging the code, the flash program for Quad SPI flash progress bar seems to stop around 40-50%. Wait for around 30 secs to 1 min. The progress bar will reach 100% and complete the programming.

2.5.3 Raw binary

Attempt to generate raw binary creates at least a 256 MB file due to internal flash region at 0x000_0000 and Quad SPI flash region at 0x1000_0000. Raw binary does not contain header to isolate the two offsets, ELF and SREC formats do.

3. Legal information

3.1 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.

3.2 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.

3.3 Trademarks

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

4. Contents

1 Introduction 3

2 Description 4

2.1 LCD in 16BPP RGB565 with images data located in SPIFI 4

2.2 Converting Bitmap file to image file in C 5

2.3 Locating data in Quad SPI flash 6

2.4 Notes 7

2.4.1 IAR reset type and macro setup: 7

2.4.2 Adding 16MB Quad SPI flash to Flash programming in Keil 9

2.4.3 MCUXpresso project bring up. 11

2.5 Issues 13

2.5.1 IAR linker warning message. 13

2.5.2 Keil flash programming progress bar delay .. 13

2.5.3 Raw binary 13

3 Legal information 14

3.1 Definitions 14

3.2 Disclaimers 14

3.3 Trademarks 14

4 Contents 15

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.