EMWIN: DESIGN AND DEVELOP A FULLY FEATURED GUI FOR FREE

NXP SEMICONDUCTORS & SEGGER

APRIL 6, 2017

AGENDA

- Introduction to SEGGER
- emWin graphics
  - Background
  - Examples of emWin HMI
  - Features
  - emWin utilities
- Overview of LPC MCUs for embedded graphics
- Where to find more information
- Q & A

Introduction to SEGGER

- 20 years of experience with embedded systems
- Customers world wide
- Debugging and production tools
- Embedded software
- One stop shopping – all out of one hand

Development Tools | Debug Probes | RTOS | File System | Compression | Connectivity | Security & IoT | User Interface | In-Field-Upgrades | Production

Complete R&D and production chain out of one hand!

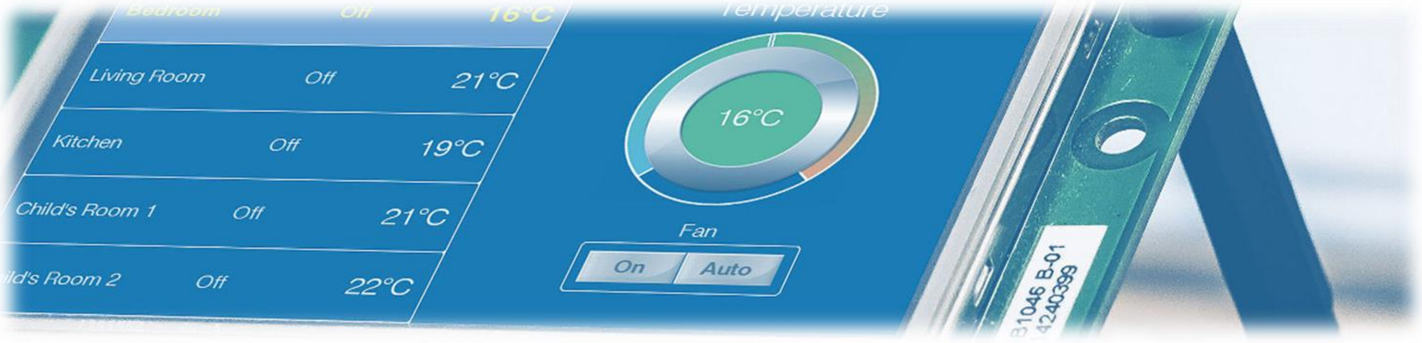
SEGGER
It simply works!

What is emWin?

- A library for developing graphical user interfaces
- Key attributes:
  - Written in ANSI C
  - Supports any CPU
  - No RTOS required
  - Low memory footprint
    - Less than 60KB of ROM and RAM
  - Highly configurable
  - Runtime configurable driver

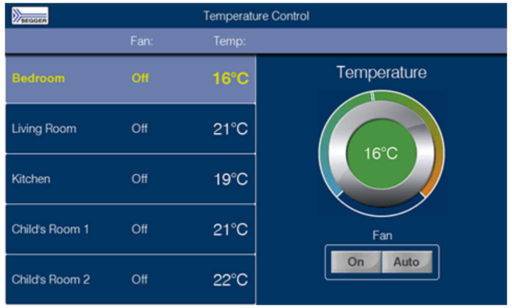
Key features

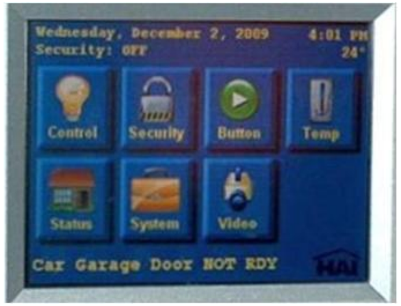
- Graphics and fonts can be placed anywhere (internal/external memory)
- Small footprint (single chip)
- VNC support with file transfer
- Can be used with any IDE
- Runs on any CPU
- Also available as source code
- Default look and feel for widgets
- Supports common image formats at runtime (BMP, JPEG, PNG, GIF)
- Supports direct and indirect LCD interface
- Window-Manager

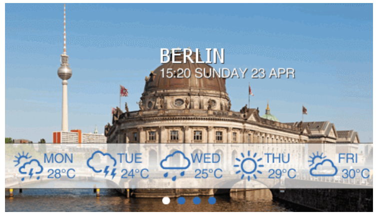


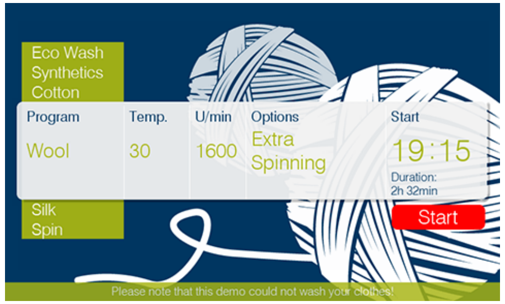
It simply works!

Example emWin HMI’s









What customers have used emWin for on LPC MCUs:

- Paper money counter (LPC4300, 3.1” LCD)
- ATM (LPC1800,14” 1024x768 LCD)
- Industrial touch panel (LPC1788, 10.1” 640x480 LCD)
- Washing machine (LPC3000)
- Elevator control with LCD (LPC1788)
- High accuracy scales with LCD (LPC1788)
- Security Panel (LPC2132)

It simply works!

Using emWin on NXP devices

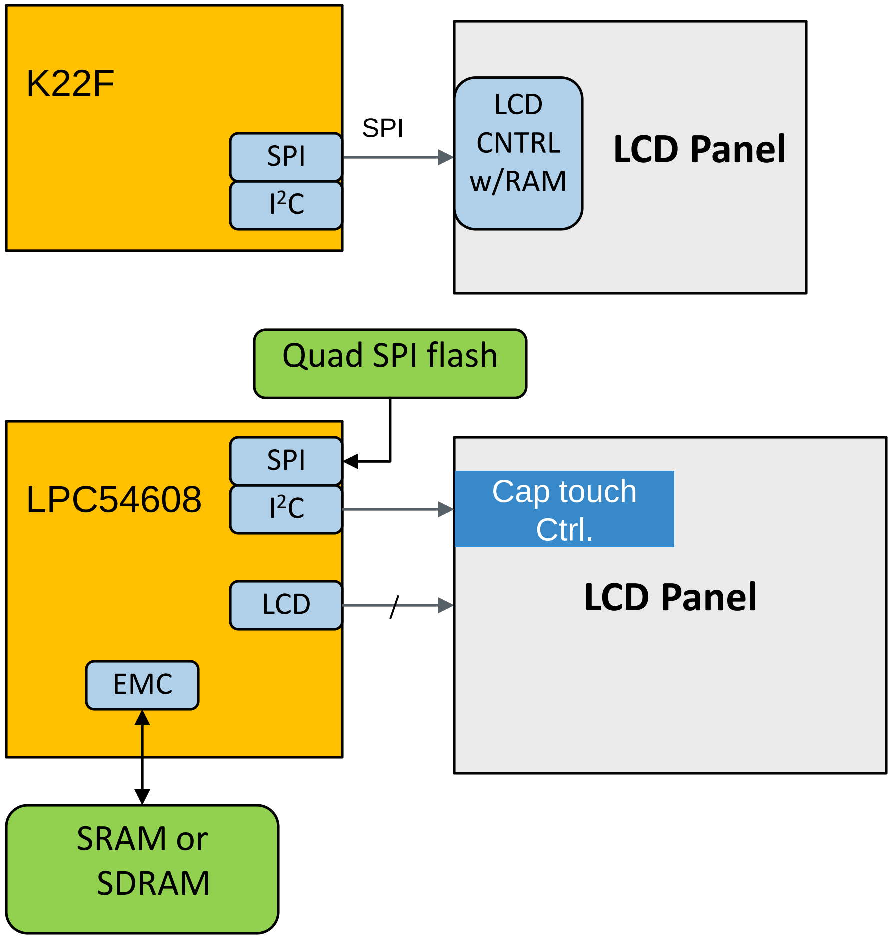
- Free to use on NXP MCUs (ARM7, ARM9, M0/0+, M3 or M4)
- No royalties or licensing fees when used with NXP Cortex-M MCUs (Kinetis or LPC)
- Provided as a pre-compiled library right from NXP website
  - Including runtime configurable drivers
- No limitations on the use of emWin with NXP MCUs
- Source code upgrade with a discount of 50% for NXP users
- Product trainings tailored to your needs
- Support from engineer to engineer via SEGGER forum

MCU with external or internal LCD controller

Example interfaces MCU to LCD

- Serial interfaces to reduce pin count to the LCD
  - Lower resolution due to limited SPI bandwidth
  - GUIDRV_FlexColor

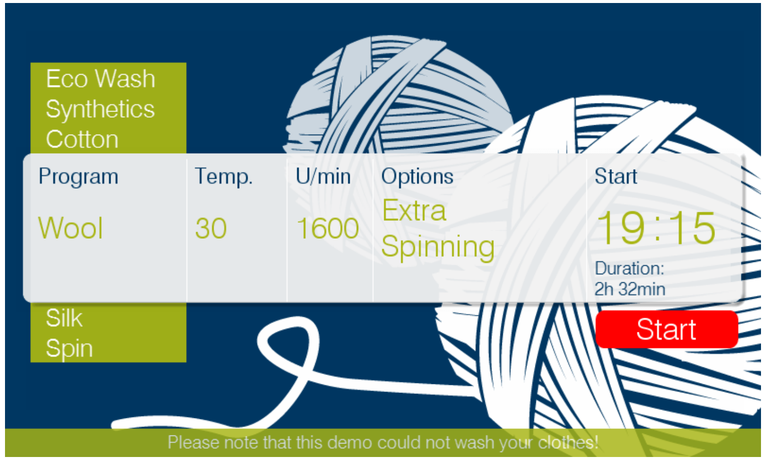
- MCU with parallel LCD controller on-chip
  - Can support mid-range resolutions
  - Quad SPIFI used for fast access to pre-defined graphics resources
  - GUIDRV_Lin



SEGGER
It simply works!

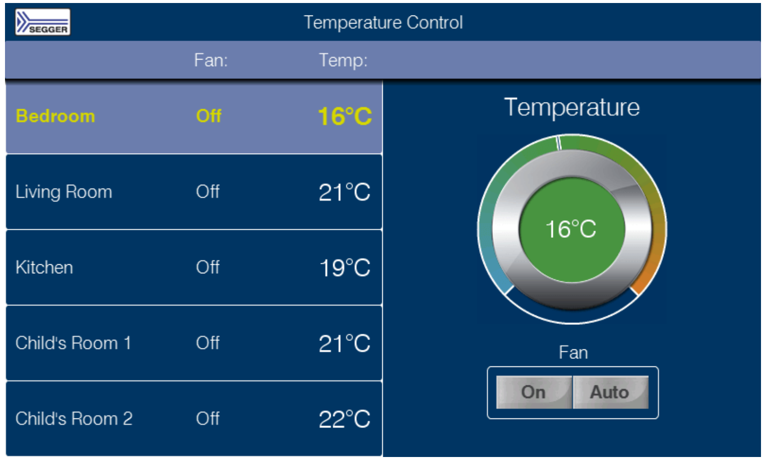
EMWIN FEATURES

Typical emWin HMI’s

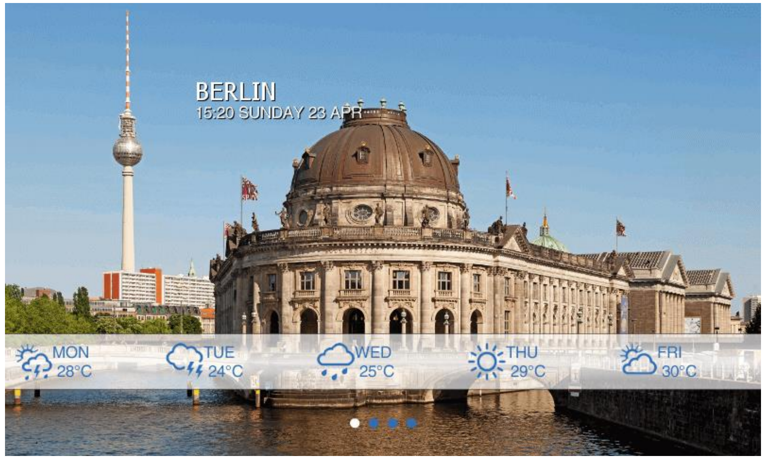


Program	Temp.	U/min	Options	Start
Wool	30	1600	Extra Spinning	19:15 Duration: 2h 32min

Please note that this demo could not wash your clothes!


Temperature Control

	Fan:	Temp:
Bedroom	Off	16°C
Living Room	Off	21°C
Kitchen	Off	19°C
Child's Room 1	Off	21°C
Child's Room 2	Off	22°C


BERLIN
15:20 SUNDAY 23 APR

MON	TUE	WED	THU	FRI
28°C	24°C	25°C	29°C	30°C

Tools overview

Bitmap converter – Convert any image

emWinView – View screen while debugging in the simulation

emWinSPY – Runtime analysis of your GUI application

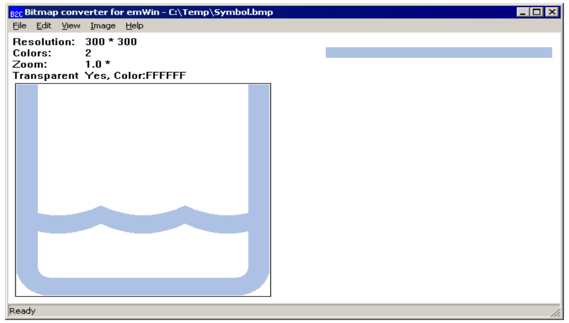
Bin2C – Convert Binary to C-code

U2C – Convert UTF8-text into C-code.

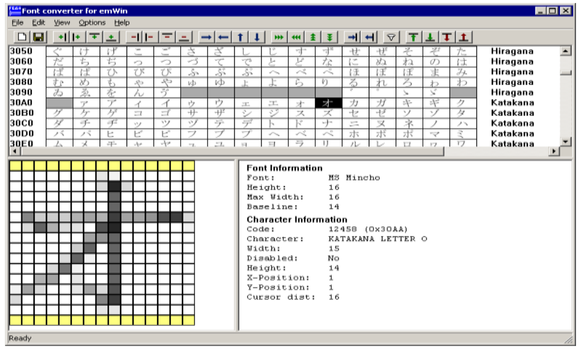
<u>Optional tools:</u>

Font converter – Convert and modify fonts for use with emWin

GUIBuilder – Create dialog based interfaces without writing code

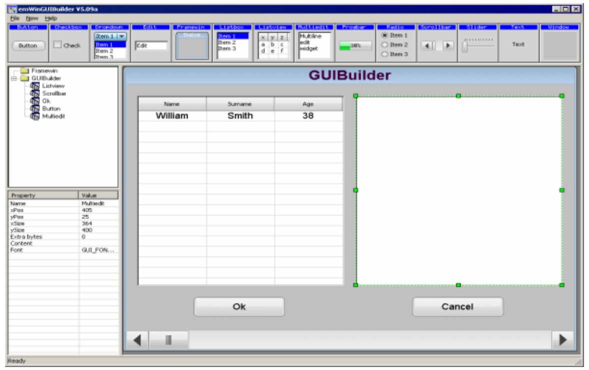


Resolution: 300 * 300
Colors: 2
Zoom: 1.0 *
Transparent Yes, Color:FFFFFF



Font Information
Font: MS Mincho
Height: 16
Max Width: 16
Baseline: 14

Character Information
Code: 12458 (0x30AA)
Character: KATAKANA LETTER O
Width: 15
Disabled: No
Height: 14
X-Position: 1
Y-Position: 1
Cursor dist: 16



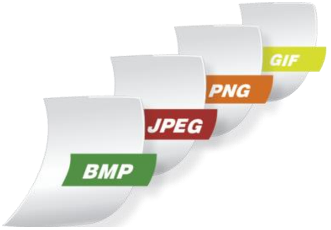
Name	Surname	Age
William	Smith	38

Property	Value
Name	Multiedit
xPos	405
yPos	25
xSize	364
ySize	400
Extra bytes	0
Content	
Font	GUI_FON...

Core functions

Features covered by the base package:

- Image file support for BMP, GIF, PNG and JPEG
- Drawing of images from non addressable media
- Language support (BiDi, use of resource files)
- Alpha blending
- Animations
- Sprites and cursors, also animated
- Drawing of different kind of shapes
- Displaying text and values
- Multiple buffering
- Virtual screens
- Touch screen support
- Multitasking support
- Standard font package



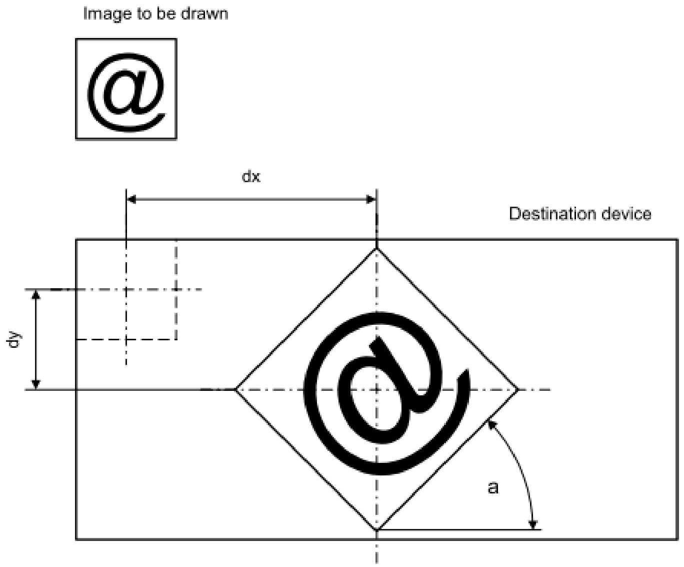
Memory devices

How do they work?

- Drawing operations passed to a memory device instead to the display
- Memory device is hardware independent destination device for drawing operations.

What can they be used for?

- Container for decompressed images
- Scaling and rotating
- Fading operations
- Window animations
- Transparency effects
- Preventing flickering

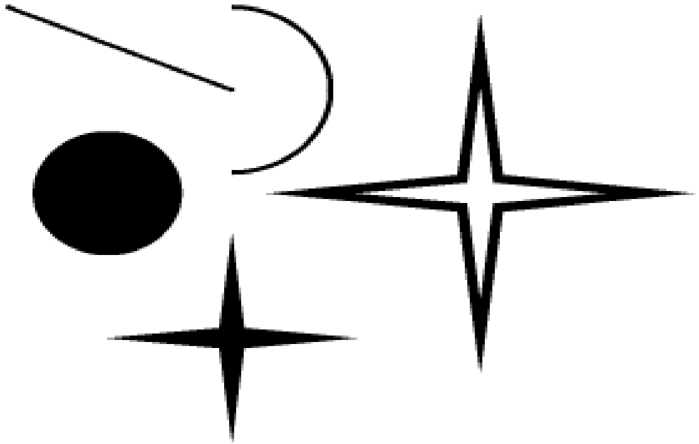


Antialiasing

Antialiasing smoothes curves and diagonal lines by „blending" the background color with that of the foreground.

emWin supports antialiased drawing of:

- Text
- Arcs
- Circles
- Lines
- Polygons



Window manager

What is the **W**indow **M**anager (WM)?

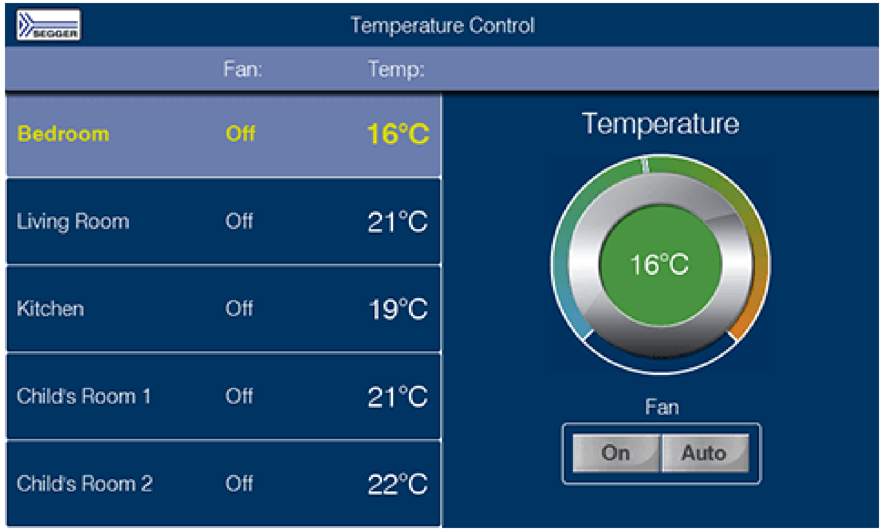
- Management system for a hierarchic window structure
- Callback mechanism based system

Foundation of widget library

All widgets are based on the functions of the WM.

Basic capabilities:

- Automatic clipping
- Automatic use of multiple buffers
- Automatic use of memory devices
- Automatic use of display driver cache
- Motion support



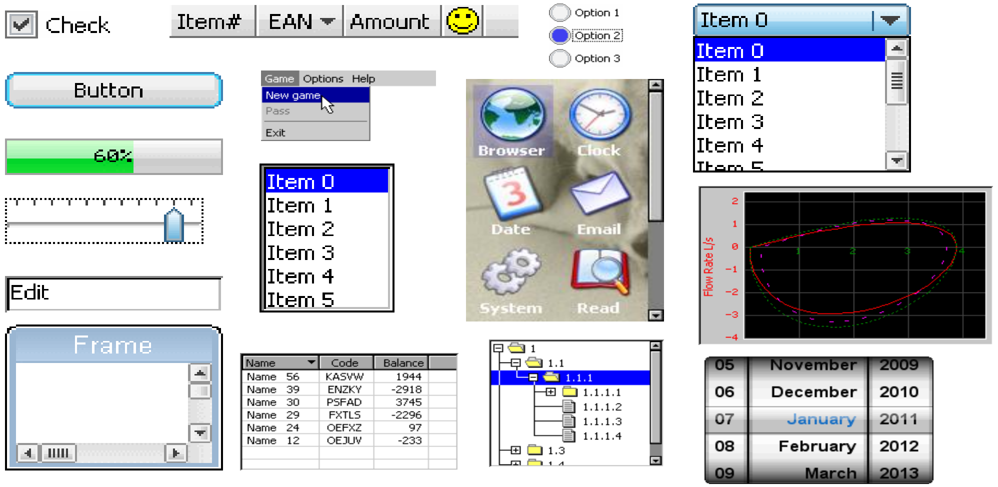
Temperature Control

	Fan:	Temp:
Bedroom	Off	16°C
Living Room	Off	21°C
Kitchen	Off	19°C
Child's Room 1	Off	21°C
Child's Room 2	Off	22°C

Widget library

Widget = **Wi**ndow + Ga**dget**

- Wide range of predefined widgets (like buttons, Slider, progressbars)
- Standard look and feel
- Creation of custom widgets
- Appearence completely configurable



Name	Code	Balance
Name 56	KASVW	1944
Name 39	ENZKY	-2918
Name 30	PSFAD	3745
Name 29	FXTLS	-2296
Name 24	OEFXZ	97
Name 12	OEJUV	-233

Widget customizing

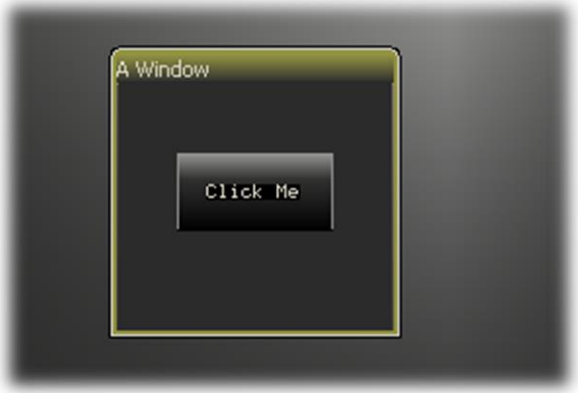
There are different ways to customize widgets and windows.

Overwriting the callback:

- Full contol
- React on different WM message
  - Change behavior
  - Change appearance

Set a custom skin or owner draw function:

- Change appearance
- Customizing only parts of the widget

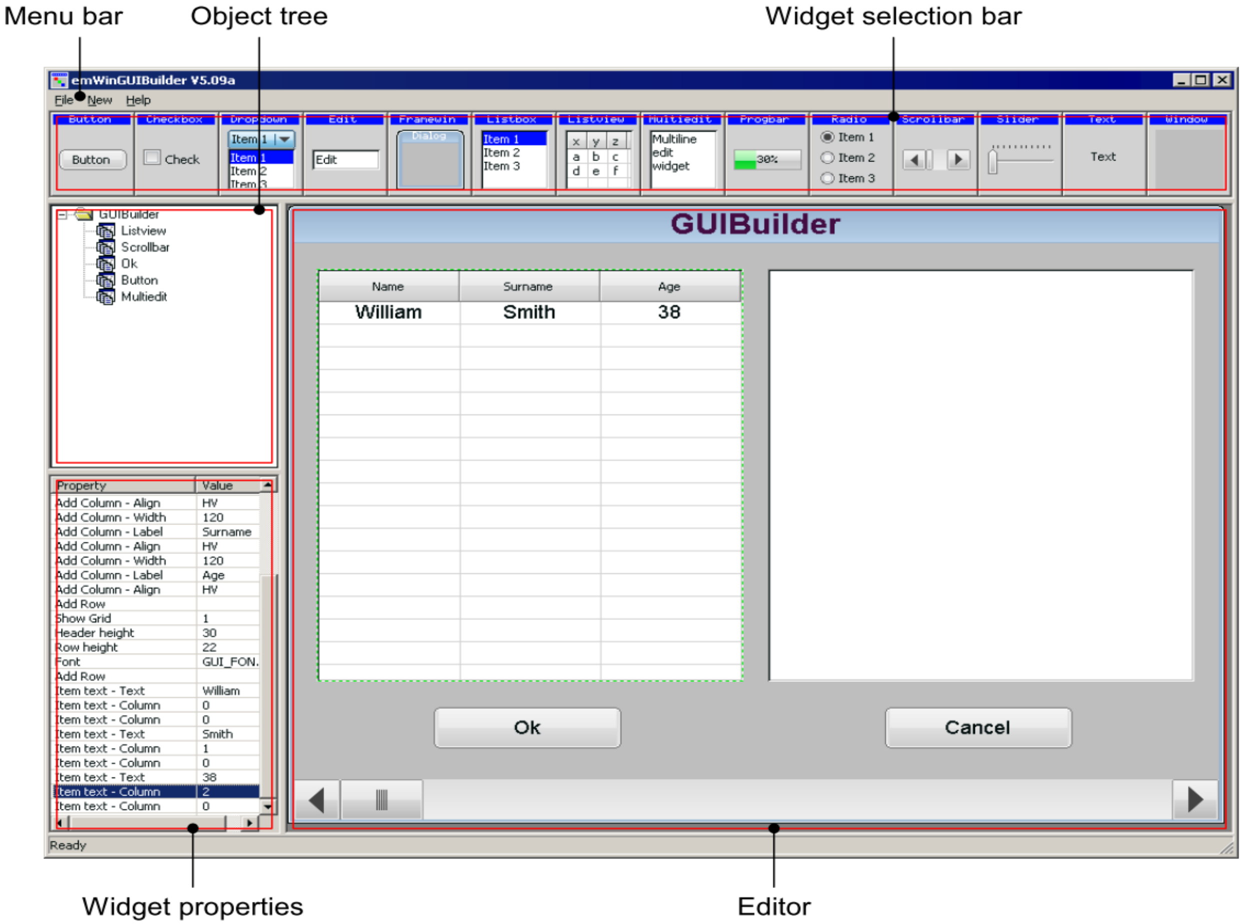


GUI-Builder

The GUI-Builder is a tool for creating dialogs without a line of code.

Basic usage of the GUI-Builder:

- Create a basic layout of your interface
- Easy arrangement of widgets
- Pre-configure widgets
- Save as C-code ready to use with emWin
- Adapt C-code for more sophisticated application
- Have full control in C-code



Name	Surname	Age
William	Smith	38

Property	Value
Add Column - Align	HV
Add Column - Width	120
Add Column - Label	Surname
Add Column - Align	HV
Add Column - Width	120
Add Column - Label	Age
Add Column - Align	HV
Add Row	
Show Grid	1
Header height	30
Row height	22
Font	GUI_FON.
Add Row	
Item text - Text	William
Item text - Column	0
Item text - Column	0
Item text - Text	Smith
Item text - Column	1
Item text - Column	0
Item text - Text	38
Item text - Column	2
Item text - Column	0

SEGGER
It simply works!

Common dialogs

Common dialogs are available for

- Message boxes
- Color selection
- Exploring a file system

MESSAGEBOX

- Created with one line of code

CHOOSEFILE

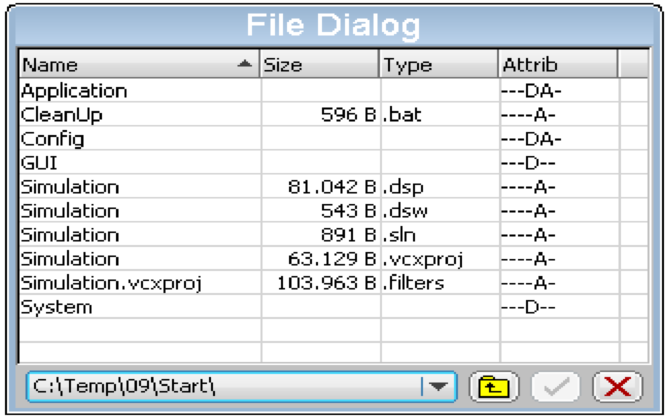
- Embedded file system explorer.
- Simple callback mechanism used to get file data from application.
- Ready to use sample for emFile available

CHOOSECOLOR

- Selection from an application defined array of colors.

CALENDAR

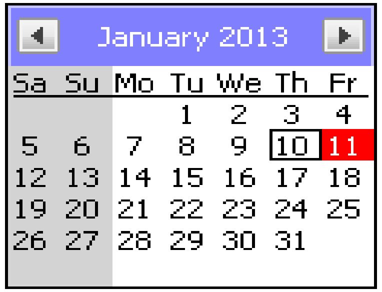
- Selection of a date of the gregorian calendar.



File Dialog

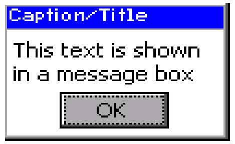
Name	Size	Type	Attrib
Application			---DA-
CleanUp	596 B	.bat	----A-
Config			---DA-
GUI			---D--
Simulation	81.042 B	.dsp	----A-
Simulation	543 B	.dsw	----A-
Simulation	891 B	.sln	----A-
Simulation	63.129 B	.vcxproj	----A-
Simulation.vcxproj	103.963 B	.filters	----A-
System			---D--

C:\Temp\09\Start\



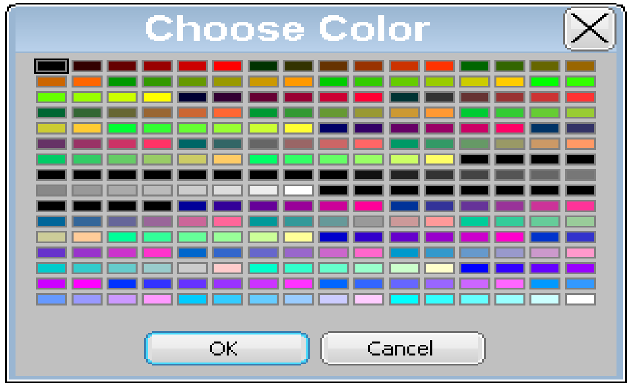
January 2013

Sa	Su	Mo	Tu	We	Th	Fr
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	



Caption/Title

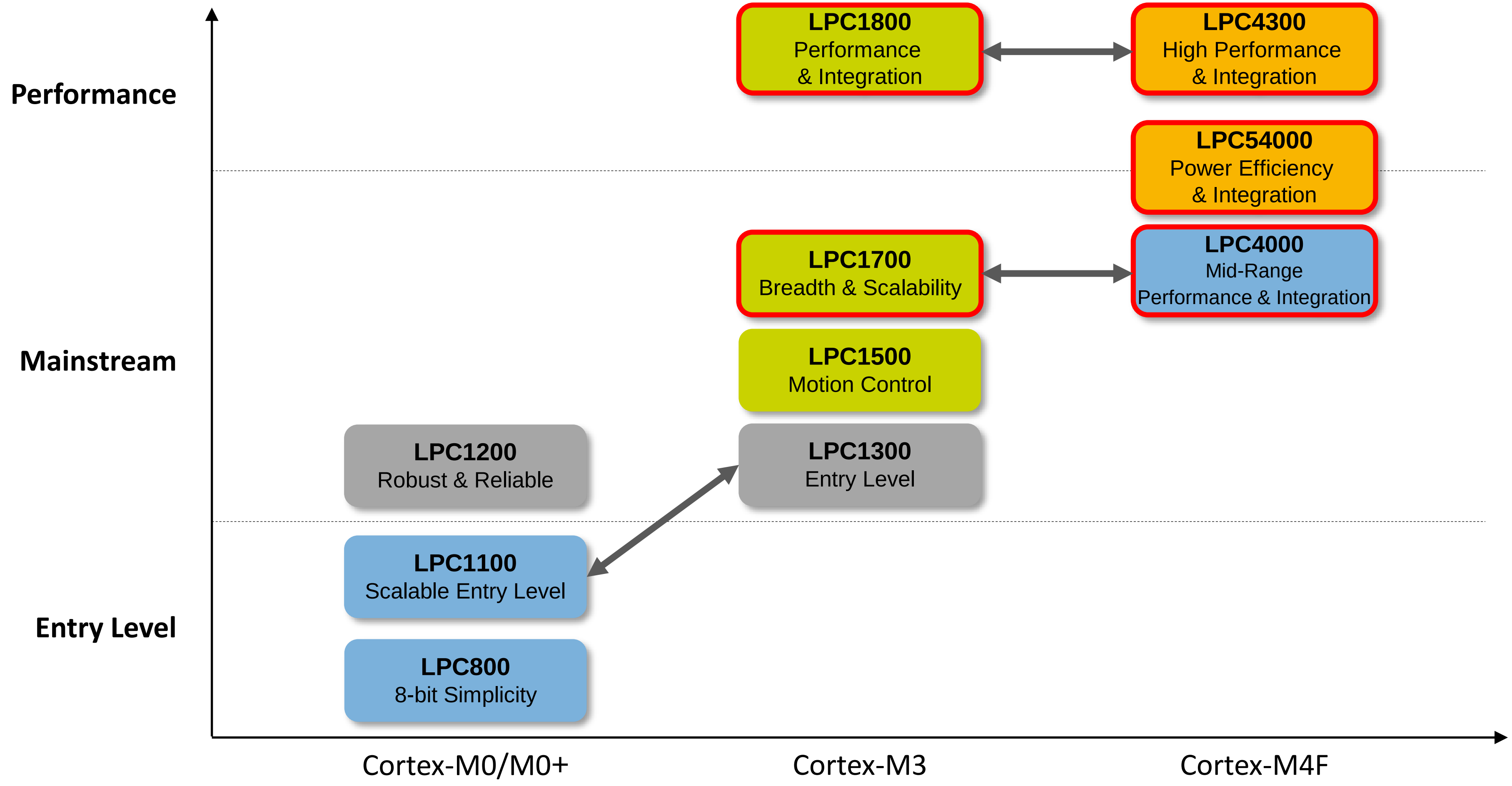
This text is shown in a message box



Choose Color

LPC FAMILY FOR EMBEDDED GRAPHICS

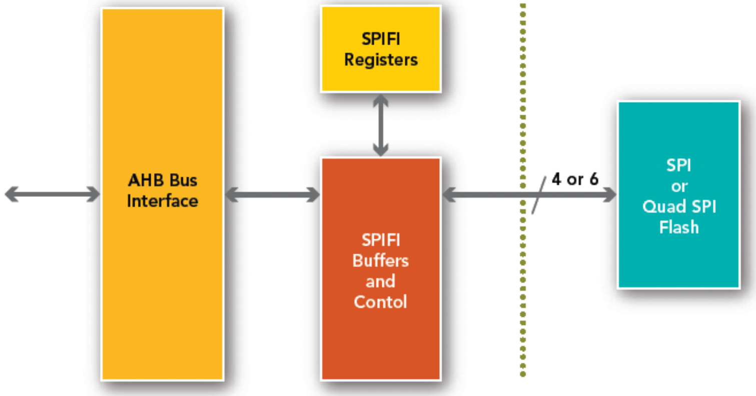
LPC Microcontroller Portfolio



SEGGER
It simply works!

SPIFI (SPI Flash Interface)

- Enables Flash to appear in MCU memory map and be read like other on-chip memory (incl. DMA)
- Why use SPIFI?
  - Cost: small, inexpensive serial Flash
  - Performance: ~70% of internal Flash
  - Space: Saves board space and pins
  - App size: Ideal for storing image/data



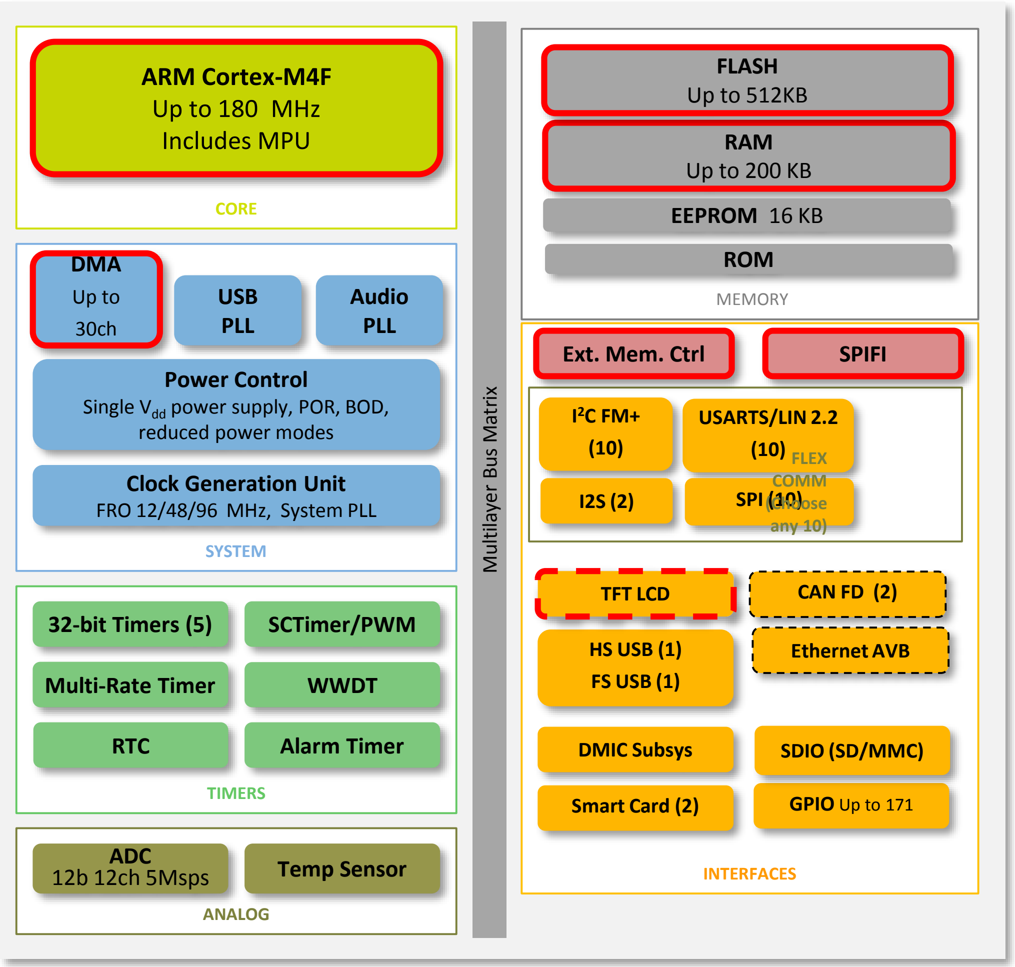
Graphic LCD Interface

Key features

- Support for STN and TFT panels
- Up to 1024x768 resolution
- 24-bit LCD interface supports 24bpp (16M colors)
- Palette table allows display of up to 256 of 64K colors
- Adjustable LCD bus size supports various panel bus configurations
- Dedicated LCD DMA controller
  Hardware cursor support

Full supported by MCUXpresso SDK / LPCOpen Board Support Packages

LPC546xx Block Diagram



CPU

- 180MHz Cortex-M4F

Memory

- Up to 512 KB Flash, Up to 200 KB RAM
- 16 KB EEPROM

Interfaces for connectivity & sensors

- Stereo DMIC subsystem
- 1x HS USB (H/D) w/ on-chip HS PHY, XTAL-less FS USB (H/D)
- 10 SPI, 10 I2C, 10 UART, 2 I2S channels (max 10 channels total)
- Graphic LCD with resolutions up to 1024x768
- Ethernet AVB
- 2 x CAN-FD controller
- Quad SPI flash interface
- External Memory interface (up to 32 bits)

Packages

- LQFP208 (28 x 28 mm), TFBGA180 (12 x 12 mm)
- LQFP100, TFBGA100

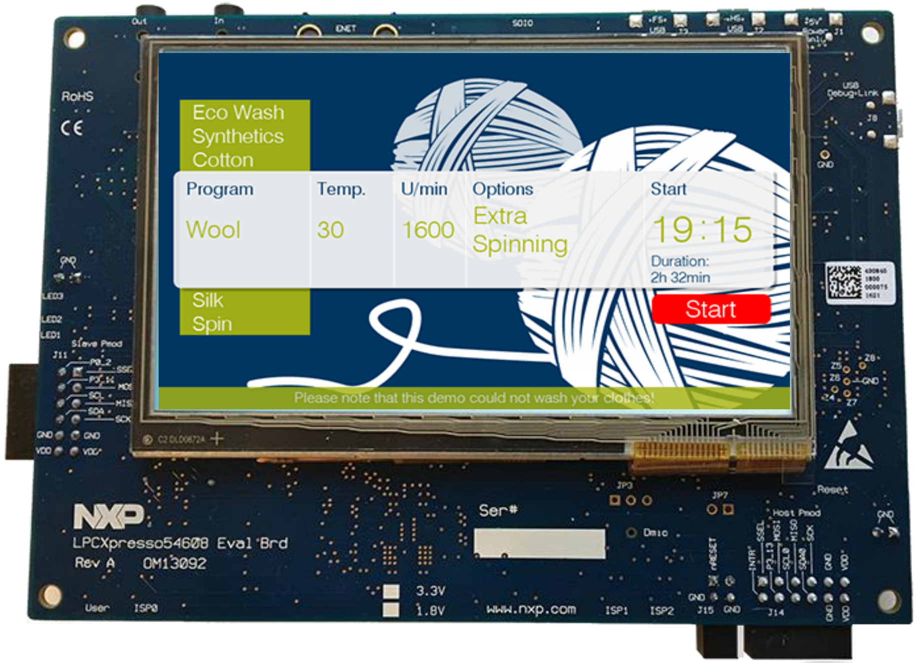
Other

- Operating voltage: 1.71 to 3.6V
- Temperature range: -40 to 105 °C

LPCXpresso54608

Embedded Graphics Development platform for LPC546xx Series

- LPC54608 MCU running at 180MHz
- 4.3” TFT LCD (272x480) cap touch display
- 128Mb Micron SDRAM
- 128Mb Micron quad SPI flash
- Built-in CMSIS-DAP/J-link debug probe
- Ethernet, DMIC, SD card, USB HS/FS ports
- Stereo audio codec
- Arduino UNO R3 compatible expansion ports
- Supported by SDK with emWIN examples



It simply works!

MCUXpresso Software and Tools

for LPC & Kinetis Microcontrollers

MCUXpresso free Software and Tools

- IDE
- SDK
- Config Tools

NXP Cortex-M Microcontrollers

- LPC + Kinetis

MCUXpresso IDE

Edit, compile, debug and optimize in an intuitive and powerful IDE

MCUXpresso SDK

Runtime software including peripheral drivers, middleware, RTOS, demos and more

MCUXpresso Config Tools

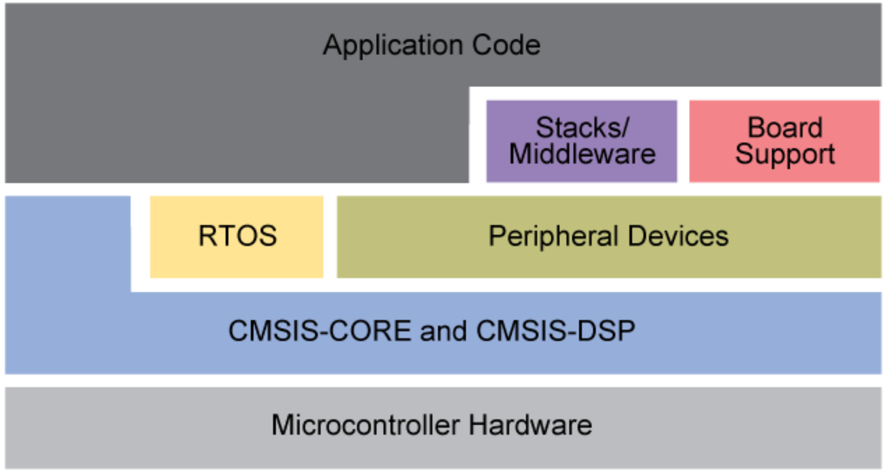
Online and desktop tool suite for system configuration and optimization

Visit http://mcuxpresso.nxp.com for Config tools & SDK builder and http://nxp.com/mcuxpresso to download MCUXpresso IDE

MCUXpresso SDK

Software framework and reference for LPC & Kinetis MCU application development

Learn more at: www.nxp.com/mcuxpresso/sdk



Features

Architecture:

- CMSIS-CORE compatible
- Single driver for each peripheral
- Transactional APIs w/ optional DMA support for communication peripherals

Integrated RTOS support (optional):

- FreeRTOS
- RTOS-native driver wrappers

Integrated Stacks and Middleware

- USB Host, Device and OTG, lwIP, FatFS
- Crypto acceleration wolfSSL & mbedTLS
- SD card support
- **emWIN graphics**

Reference Software:

- Peripheral driver usage examples
- Application demos
- FreeRTOS usage demos

License:

- BSD 3-clause for startup/drivers/USB

Toolchains:

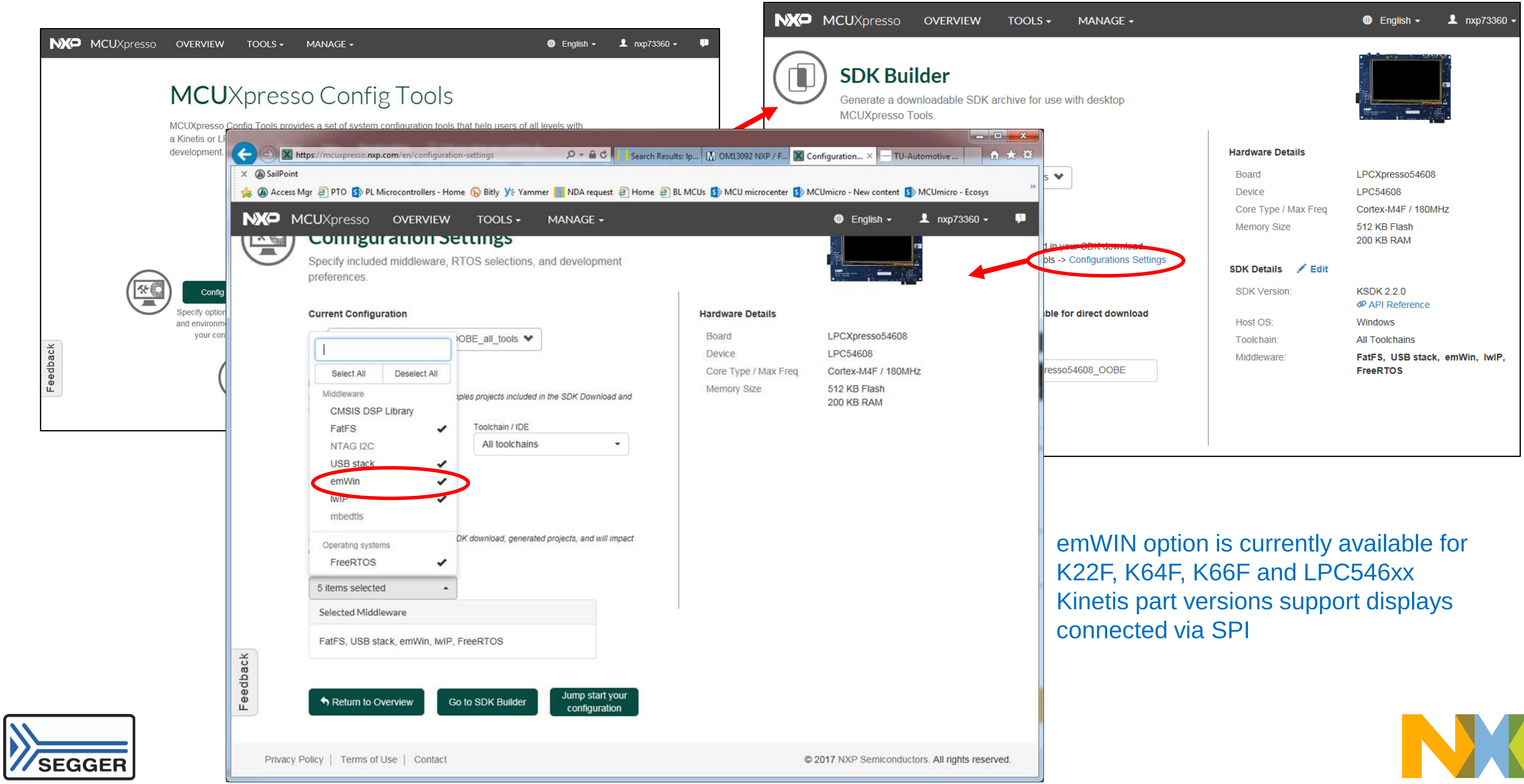
- IAR®, Keil®, MCUXpresso IDE*

Quality

- Production-grade software
- MISRA 2004 compliance

Visit http://mcuxpresso.nxp.com to build and download SDKs

Building an SDK with emWIN



MCUXpresso Config Tools

MCUXpresso Config Tools provides a set of system configuration tools that help users of all levels with a Kinetis or LF... development...

Config

Specify option... and environm... your con...

https://mcuxpresso.nxp.com/en/configuration-settings

Configuration Settings

Specify included middleware, RTOS selections, and development preferences.

Current Configuration

...OBE_all_tools

Select All | Deselect All

Middleware

- CMSIS DSP Library
- FatFS ✓
- NTAG I2C
- USB stack ✓
- emWin ✓
- lwIP ✓
- mbedtls

Operating systems

- FreeRTOS ✓

5 items selected

...ples projects included in the SDK Download and

Toolchain / IDE: All toolchains

...DK download, generated projects, and will impact

Selected Middleware

FatFS, USB stack, emWin, lwIP, FreeRTOS

Hardware Details

Board	LPCXpresso54608
Device	LPC54608
Core Type / Max Freq	Cortex-M4F / 180MHz
Memory Size	512 KB Flash 200 KB RAM

Privacy Policy | Terms of Use | Contact

© 2017 NXP Semiconductors. All rights reserved.

SDK Builder

Generate a downloadable SDK archive for use with desktop MCUXpresso Tools.

...d in your SDK download ...ols -> Configurations Settings

...ble for direct download

...resso54608_OOBE

Hardware Details

Board	LPCXpresso54608
Device	LPC54608
Core Type / Max Freq	Cortex-M4F / 180MHz
Memory Size	512 KB Flash 200 KB RAM

SDK Details Edit

SDK Version:	KSDK 2.2.0 API Reference
Host OS:	Windows
Toolchain:	All Toolchains
Middleware:	**FatFS, USB stack, emWin, lwIP, FreeRTOS**

emWIN option is currently available for K22F, K64F, K66F and LPC546xx Kinetis part versions support displays connected via SPI

SEGGER It simply works!

Examples in MCUXpresso SDK for LPCXpresso54608

- Current SDK (version 2.2) includes multiple examples that do not require a frame buffer:
  - Simple touch and draw application
  - Serial terminal
  - Keyboard and mouse (USB host) application
- Examples to be available during April (available separately but based on SDK)
  - SEGGER application example, based on FreeRTOS
  - NXP dual frame SDRAM buffer example, using accelerometer

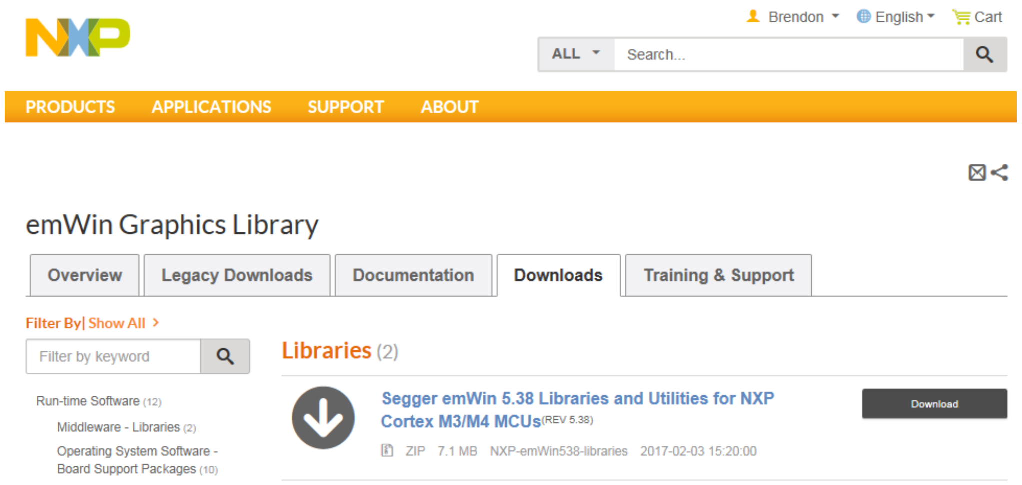
WHERE TO FIND MORE INFORMATION

emWin information

- emWin is supported by extensive information resources online
  - http://www.segger.com/emwin.html
- Information includes
  - In-depth user guide / reference manual https://www.segger.com/downloads/emwin/UM03001_emWin5.pdf
  - Full details on all utilities
  - Information of memory footprints for widgets and other resources
  - Performance benchmarks
  - Information about creating own widgets https://www.segger.com/downloads/appnotes/AN03002_Custom_Widget_Type.pdf

How to get emWIN utilities

- Free to download and use from nxp.com
  - Includes library and Windows utilities, GUI builder, bitmap and font converters
- emWIN included in MCUXpresso SDK for LPC546xx, K22F, K64F and K66F families
- Examples provided with LPCOpen (for previous LPC devices); available from nxp.com/lpcopen



emWin Graphics Library

Libraries (2)

Run-time Software (12)
Middleware - Libraries (2)
Operating System Software - Board Support Packages (10)

Segger emWin 5.38 Libraries and Utilities for NXP Cortex M3/M4 MCUs(REV 5.38)
ZIP 7.1 MB NXP-emWin538-libraries 2017-02-03 15:20:00

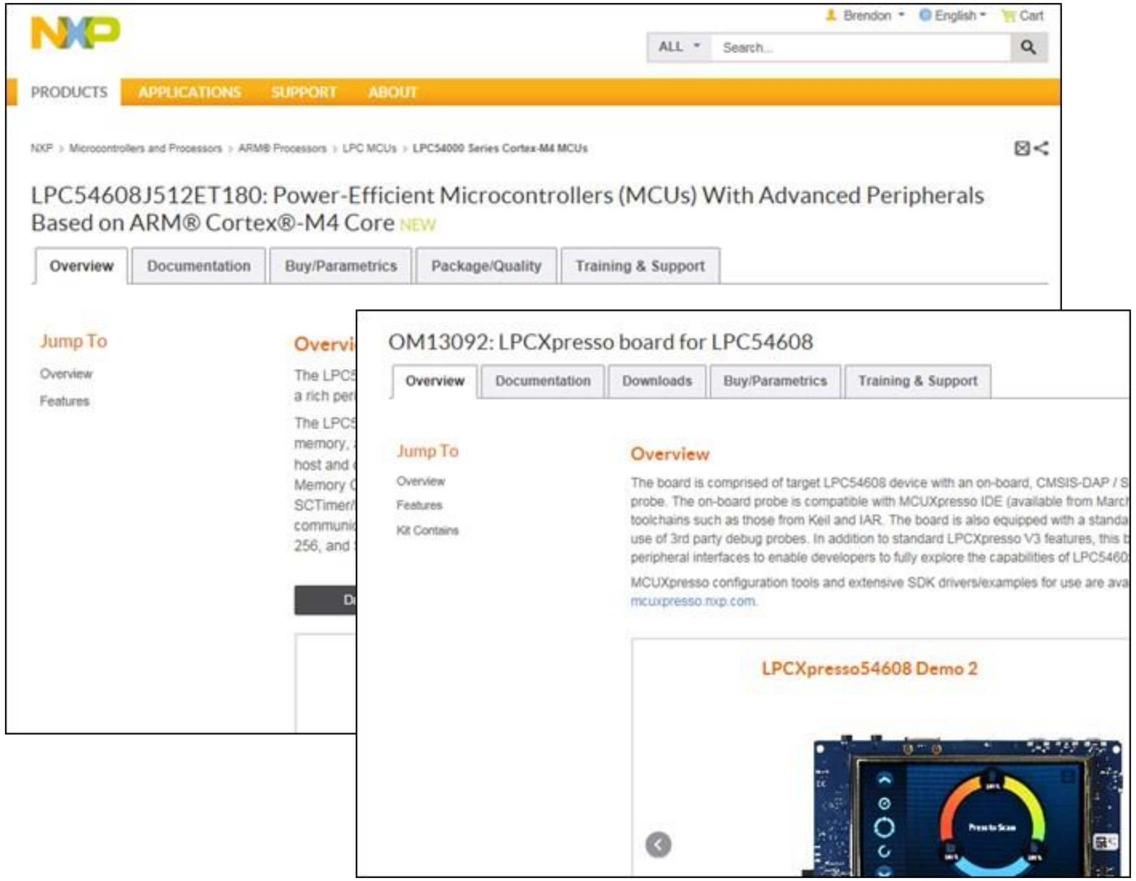
- Libraries can be used with MCUXpresso, Keil, IAR and other 3rd party ARM development tools

http://www.nxp.com/pages/emwin-graphics-library:EMWIN-GRAPHICS-LIBRARY
(or search for emwin in the search box!)

It simply works!

Where to find information on NXP MCUs

- Visit nxp.com for information
  - on the LPC range of MCUs nxp.com/lpc
  - on the LPCXpresso54608 board nxp.com/demoboard/om13092
- Site also provides contacts for your local distributors
- Buy the LPCXpresso54608 directly from the nxp.com site or from one of our distributors



LPC54608J512ET180: Power-Efficient Microcontrollers (MCUs) With Advanced Peripherals Based on ARM® Cortex®-M4 Core NEW

The LPC5
a rich per

The LPC5
memory,
host and
Memory C
SCTimer/
communic
256, and

OM13092: LPCXpresso board for LPC54608

Overview

The board is comprised of target LPC54608 device with an on-board, CMSIS-DAP / S
probe. The on-board probe is compatible with MCUXpresso IDE (available from Marc
toolchains such as those from Keil and IAR. The board is also equipped with a standa
use of 3rd party debug probes. In addition to standard LPCXpresso V3 features, this b
peripheral interfaces to enable developers to fully explore the capabilities of LPC5460

MCUXpresso configuration tools and extensive SDK drivers/examples for use are ava
mcuxpresso.nxp.com.

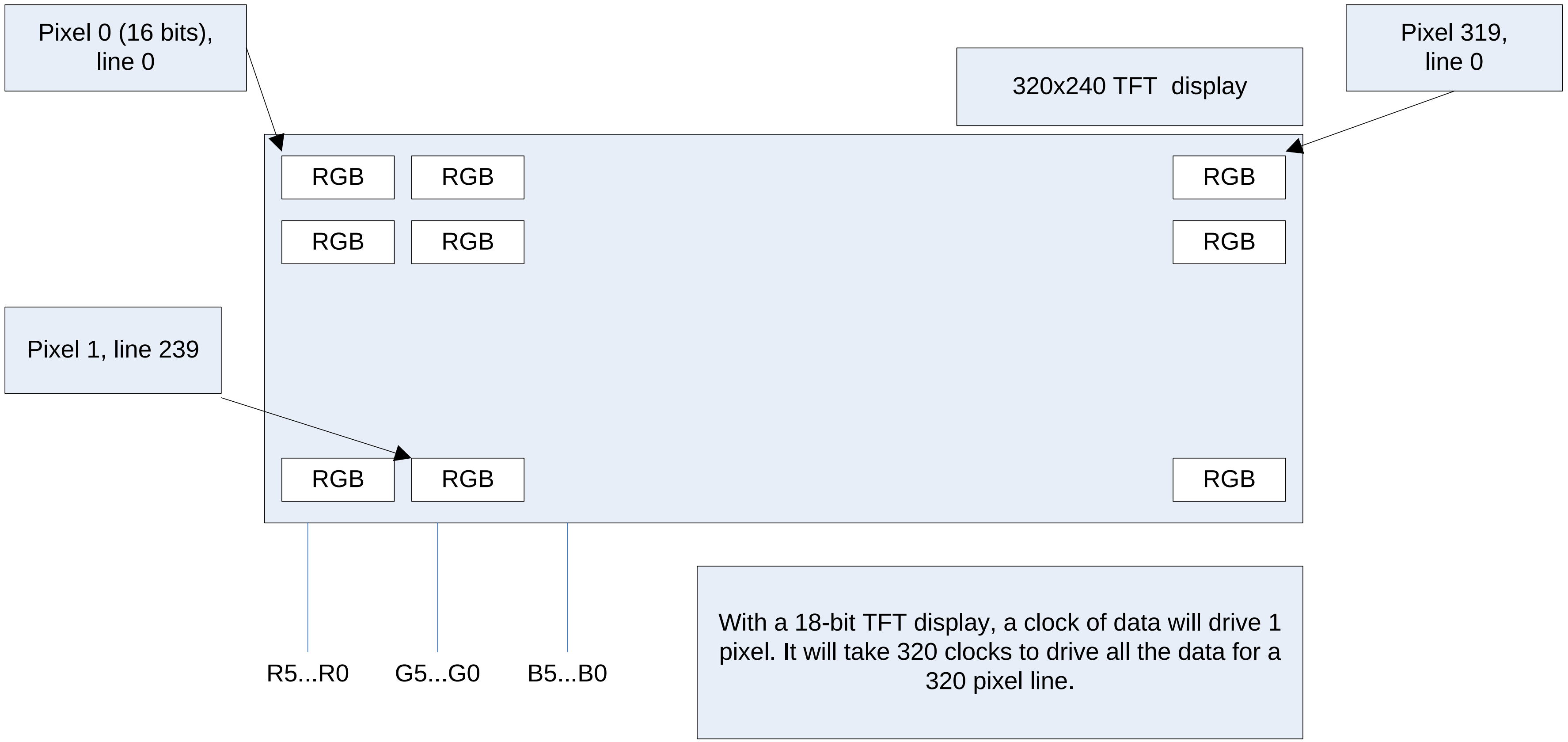
SEGGER
It simply works!

THANK YOU FOR LISTENING!

SECURE CONNECTIONS
FOR A SMARTER WORLD

ADDITIONAL INFORMATION

LCD Clocked TFT



Pixel 0 (16 bits), line 0

320x240 TFT display

Pixel 319, line 0

Pixel 1, line 239

RGB RGB RGB

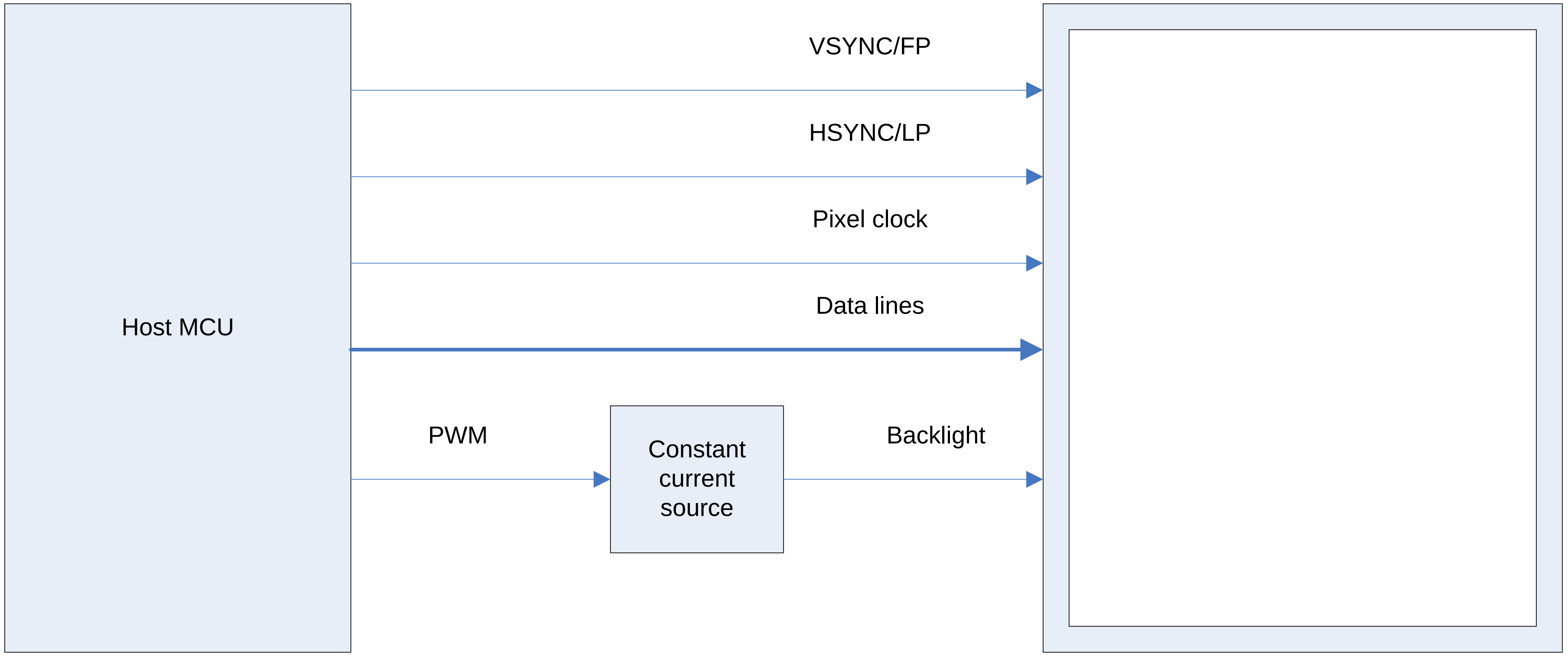
RGB RGB RGB

RGB RGB RGB

R5...R0 G5...G0 B5...B0

With a 18-bit TFT display, a clock of data will drive 1 pixel. It will take 320 clocks to drive all the data for a 320 pixel line.

Driving a clocked LCD bus



Refresh Rate

- REFRESH_RATE (Hz) =

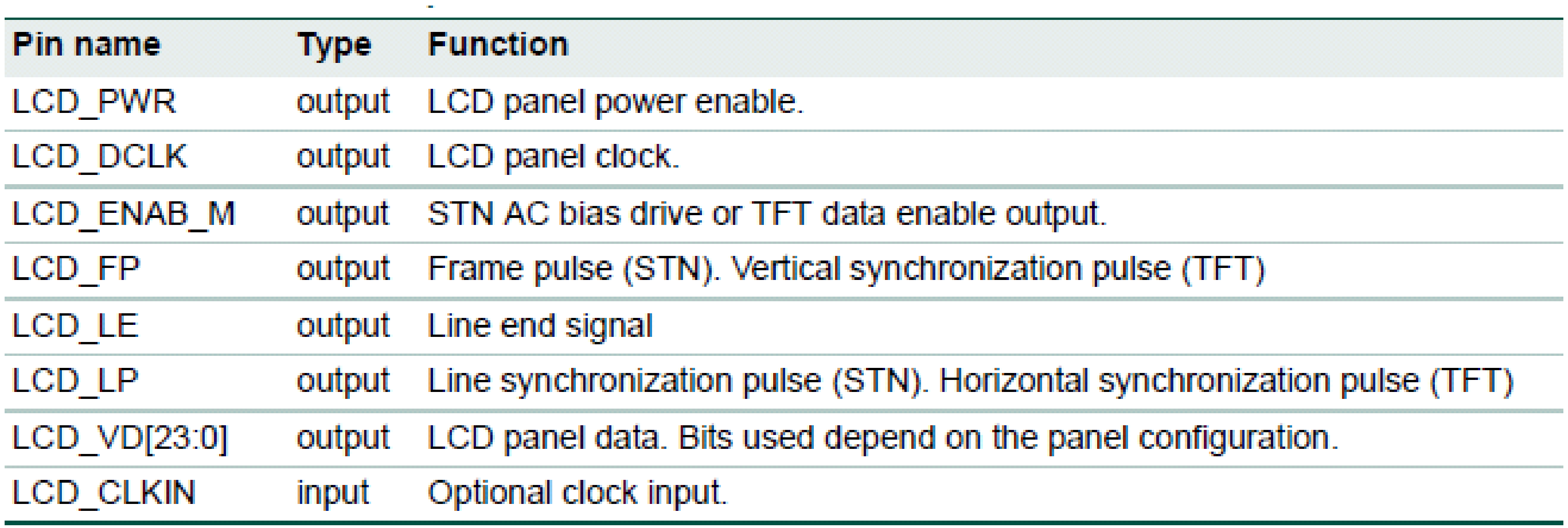
  pixel_clock_rate / [(vertical_resolution + vertical_front_porch + vertical_back_porch) * (pixel_clocks_per_data_line + horizontal_front_porch + horizontal_back_porch))]

- Example :
  - 6.5MHz pixel clock
  - vertical resolution=240 lines,
  - vertical front porch=5 lines,
  - vertical back porch=1 line,
  - pixel clocks per data line = 320 pixels,
  - horizontal front porch=20 clocks,
  - horizontal back porch=10 clocks

  - REFRESH_RATE = 6,500,000 / [(240 + 5 + 1) * (320 + 20 + 10)] = 75.5Hz

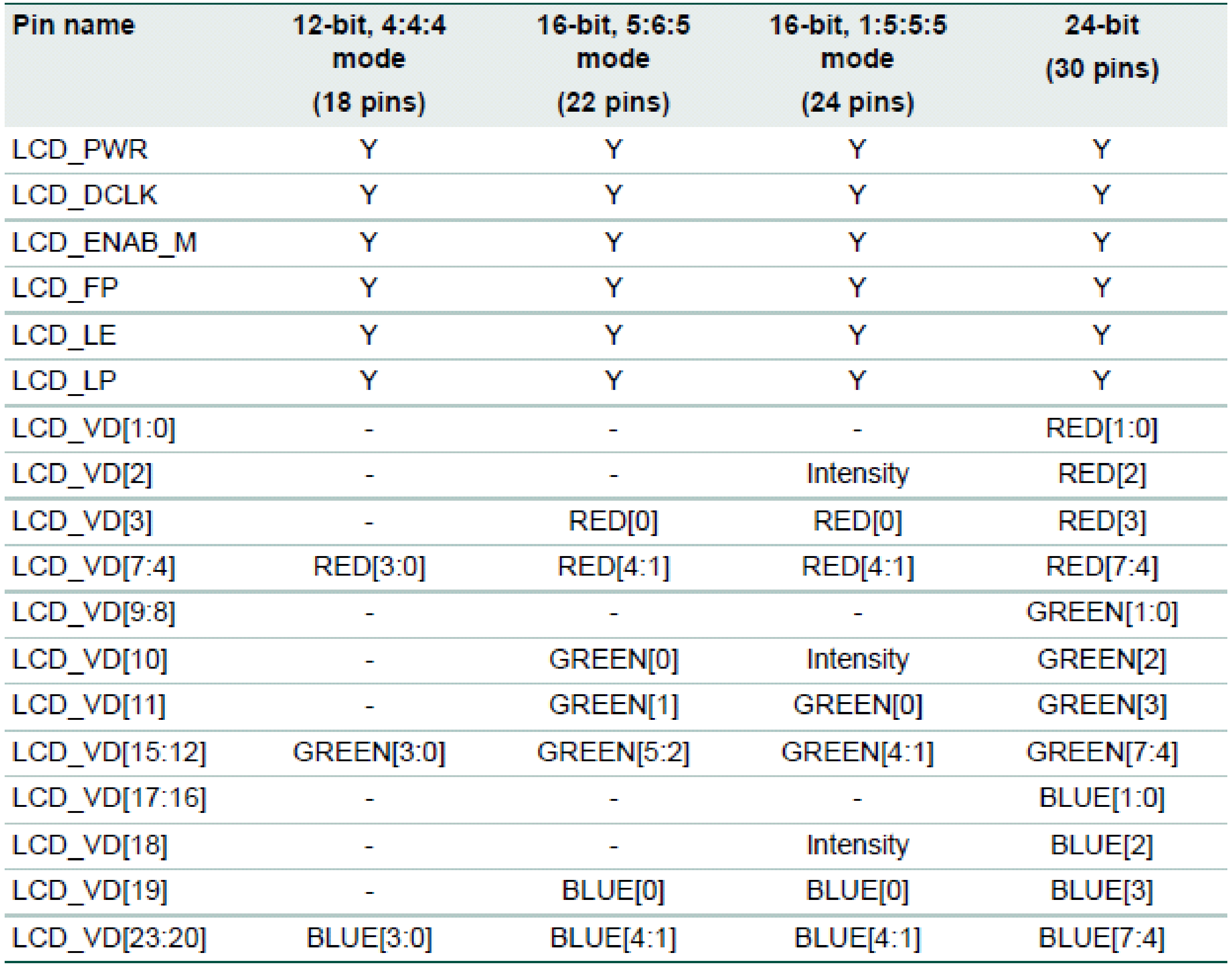
LCD Signals

- The largest configuration for the LCD controller uses 31 pins. There are many variants using as few as 10 pins for a monochrome STN panel.



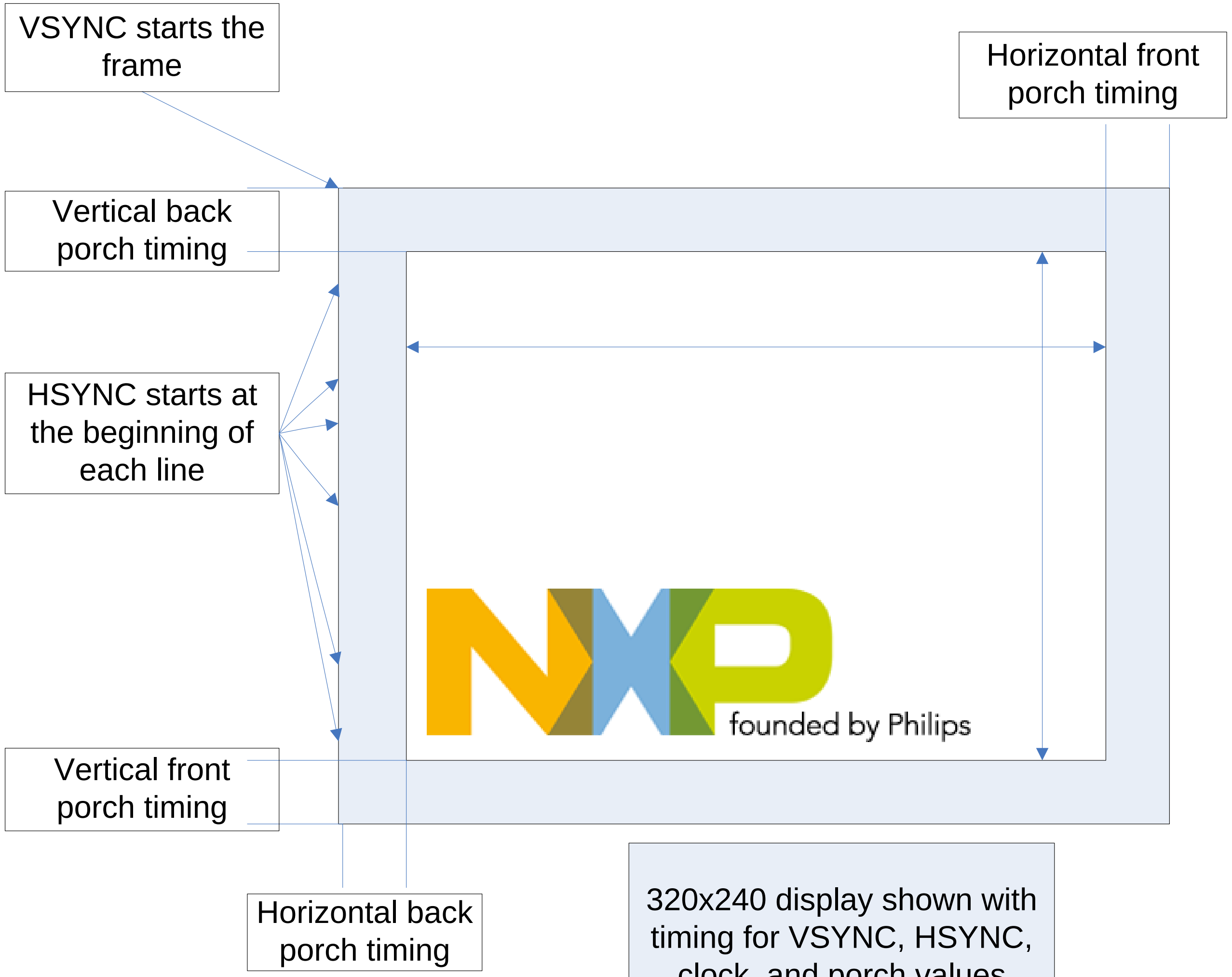
Pin name	Type	Function
LCD_PWR	output	LCD panel power enable.
LCD_DCLK	output	LCD panel clock.
LCD_ENAB_M	output	STN AC bias drive or TFT data enable output.
LCD_FP	output	Frame pulse (STN). Vertical synchronization pulse (TFT)
LCD_LE	output	Line end signal
LCD_LP	output	Line synchronization pulse (STN). Horizontal synchronization pulse (TFT)
LCD_VD[23:0]	output	LCD panel data. Bits used depend on the panel configuration.
LCD_CLKIN	input	Optional clock input.

LCD TFT Signals



Pin name	12-bit, 4:4:4 mode (18 pins)	16-bit, 5:6:5 mode (22 pins)	16-bit, 1:5:5:5 mode (24 pins)	24-bit (30 pins)
LCD_PWR	Y	Y	Y	Y
LCD_DCLK	Y	Y	Y	Y
LCD_ENAB_M	Y	Y	Y	Y
LCD_FP	Y	Y	Y	Y
LCD_LE	Y	Y	Y	Y
LCD_LP	Y	Y	Y	Y
LCD_VD[1:0]	-	-	-	RED[1:0]
LCD_VD[2]	-	-	Intensity	RED[2]
LCD_VD[3]	-	RED[0]	RED[0]	RED[3]
LCD_VD[7:4]	RED[3:0]	RED[4:1]	RED[4:1]	RED[7:4]
LCD_VD[9:8]	-	-	-	GREEN[1:0]
LCD_VD[10]	-	GREEN[0]	Intensity	GREEN[2]
LCD_VD[11]	-	GREEN[1]	GREEN[0]	GREEN[3]
LCD_VD[15:12]	GREEN[3:0]	GREEN[5:2]	GREEN[4:1]	GREEN[7:4]
LCD_VD[17:16]	-	-	-	BLUE[1:0]
LCD_VD[18]	-	-	Intensity	BLUE[2]
LCD_VD[19]	-	BLUE[0]	BLUE[0]	BLUE[3]
LCD_VD[23:20]	BLUE[3:0]	BLUE[4:1]	BLUE[4:1]	BLUE[7:4]

Driving the LCD – various timings



VSYNC starts the frame

Horizontal front porch timing

Vertical back porch timing

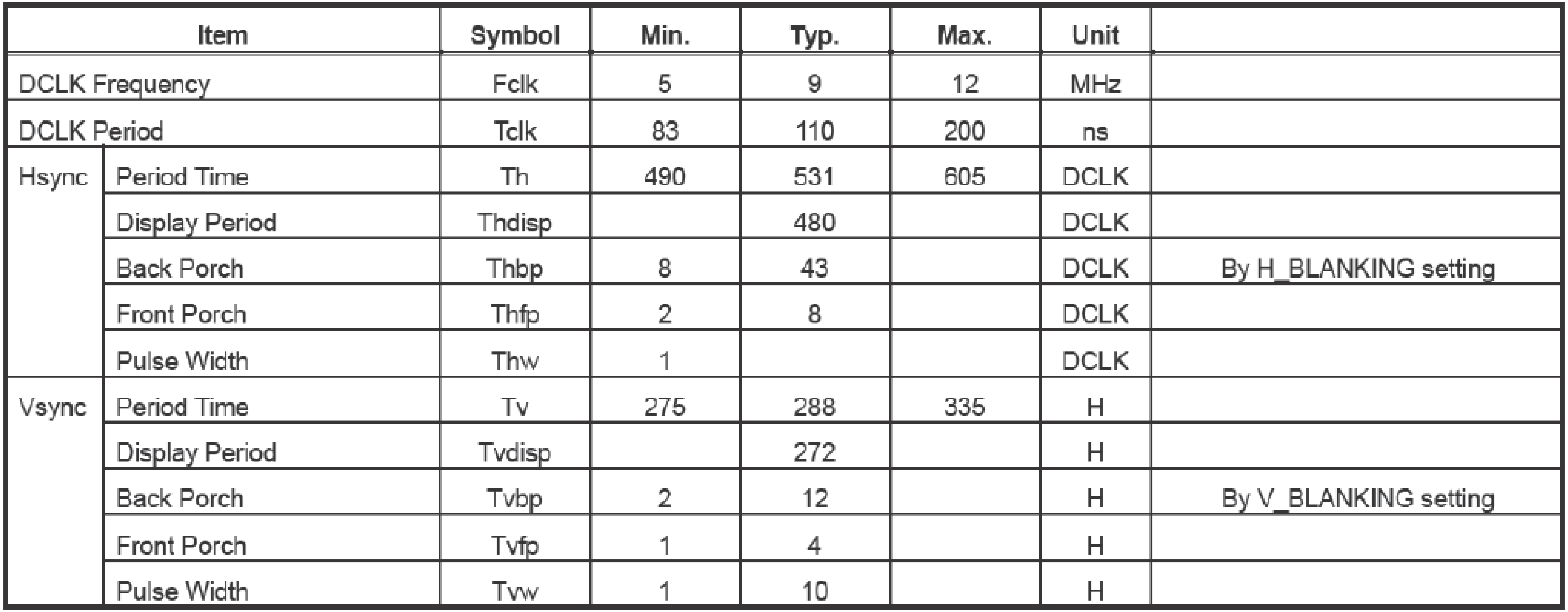
HSYNC starts at the beginning of each line

Vertical front porch timing

Horizontal back porch timing

320x240 display shown with timing for VSYNC, HSYNC, clock, and porch values

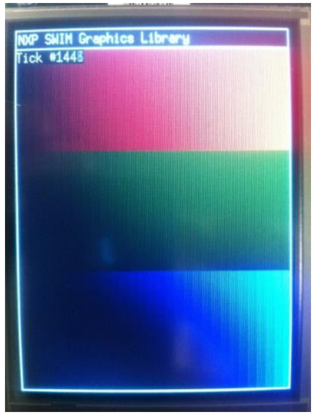
Example: Rocktech LCD on LPCXpresso54608

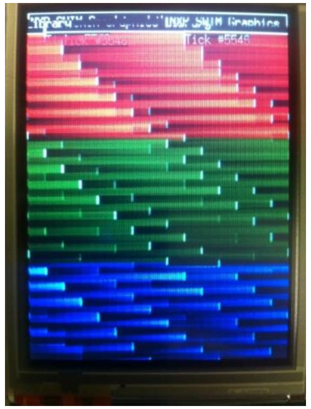


Item		Symbol	Min.	Typ.	Max.	Unit	
DCLK Frequency		Fclk	5	9	12	MHz	
DCLK Period		Tclk	83	110	200	ns	
Hsync	Period Time	Th	490	531	605	DCLK	
	Display Period	Thdisp		480		DCLK	
	Back Porch	Thbp	8	43		DCLK	By H_BLANKING setting
	Front Porch	Thfp	2	8		DCLK	
	Pulse Width	Thw	1			DCLK	
Vsync	Period Time	Tv	275	288	335	H	
	Display Period	Tvdisp		272		H	
	Back Porch	Tvbp	2	12		H	By V_BLANKING setting
	Front Porch	Tvfp	1	4		H	
	Pulse Width	Tvw	1	10		H	

LCD Tearing

- Tearing:

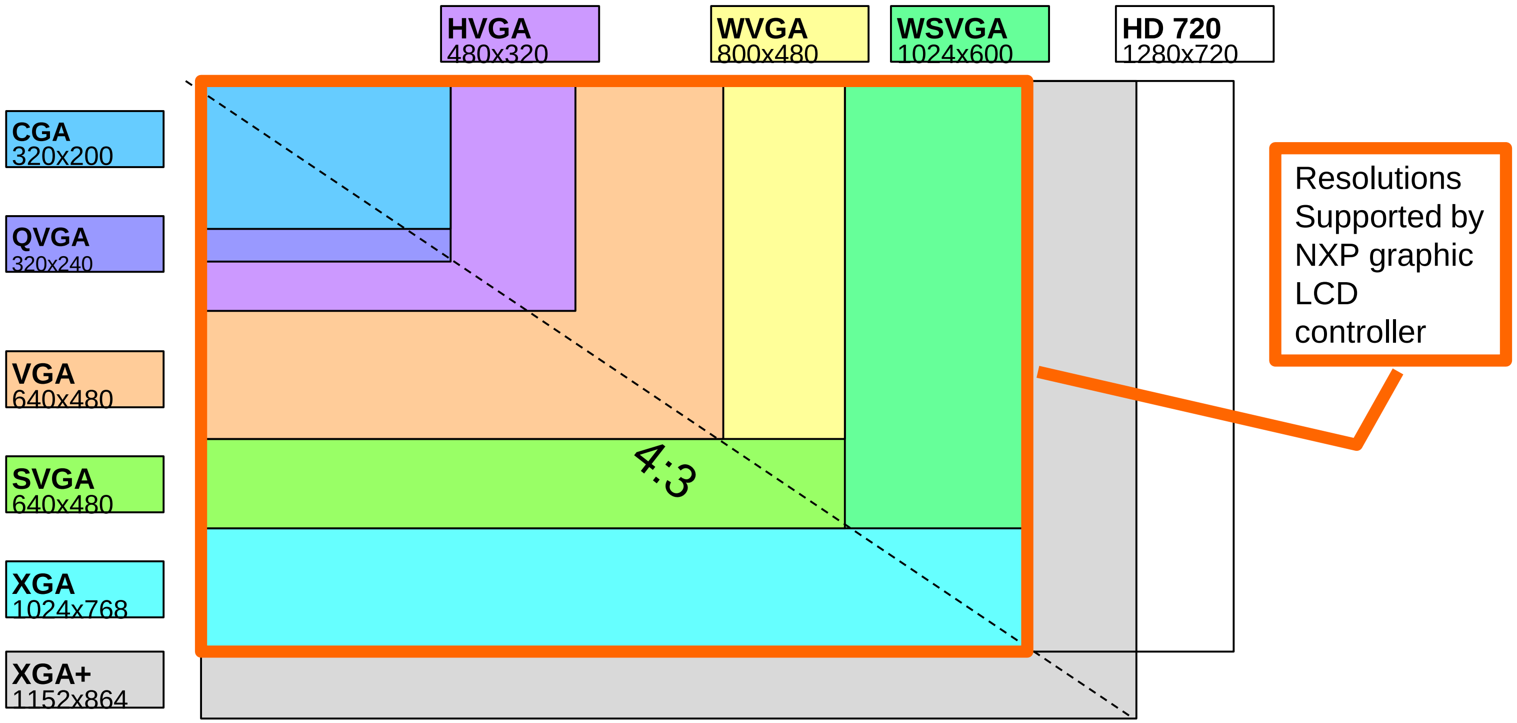




- Result of LCD DMA unable to service the LCD FIFO in time
- Use the FIFO Underflow to monitor for this
- Workarounds
- Change internal bus (AHB) priority
- Slow down frame refresh rate, pixel clock if possible
- Use 32-bit wide external memories
- Increase the SDRAM clock speed, use faster SRAM
- Profile code and move frequently accessed code to internal SRAM

SYSTEM DESIGN CONSIDERATIONS

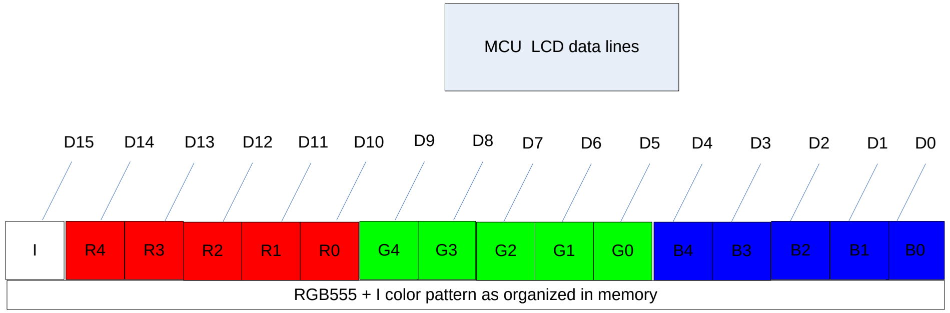
Choosing an LCD: Resolution:



Resolutions Supported by NXP graphic LCD controller

Choosing an LCD: Color Depth

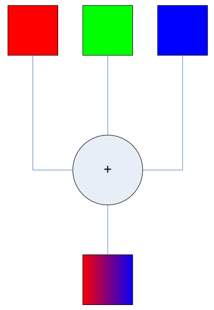
- Color depth or bits per pixel (bpp)



MCU LCD data lines

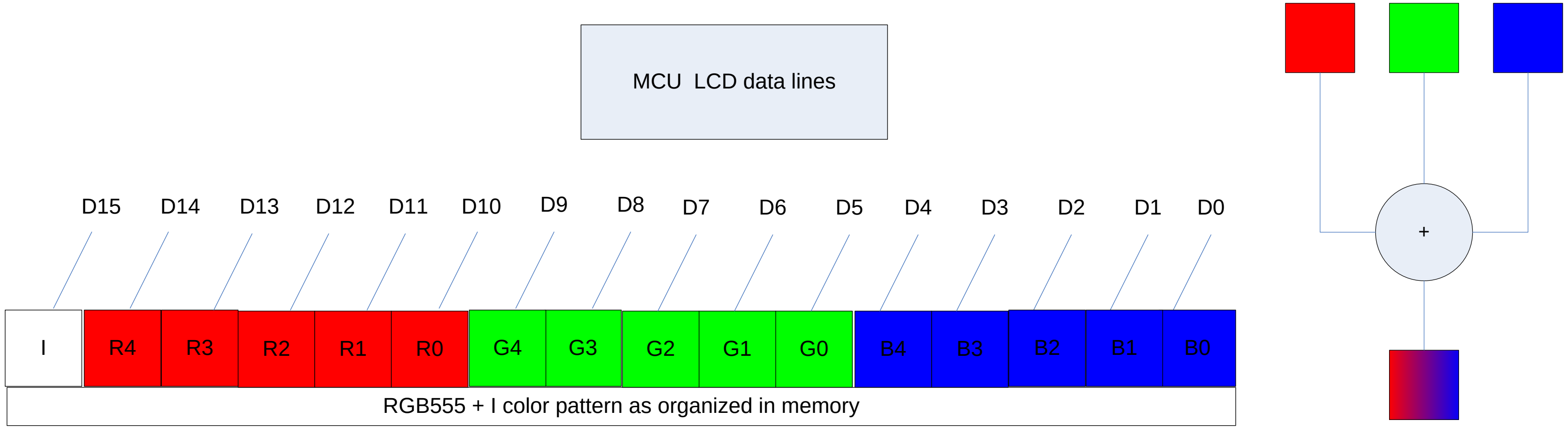
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
I	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

RGB555 + I color pattern as organized in memory



Resolution and Color Depth

- Resolution is not measured in inches!
  - QVGA 320 X 240
  - VGA 640 x 480
  - SVGA 800 X 600
  - Landscape or portrait orientation
- Color depth or bits per pixel (bpp)



D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
I	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

RGB555 + I color pattern as organized in memory

What is a Frame Buffer?

- **Contiguous** memory buffer containing a complete frame of data
- Consists of color values for every pixel
- Color values are commonly represented as
  - 1 bit (1 bpp): Monochrome
  - 2 bit (2 bpp): Palette based (4 colors)
  - 4 bit (4 bpp): Palette (16 colors, controller has a palette look-up table)
  - 8 bit (8 bpp): Palette (256 colors, controller has a palette look-up table)
  - 16 bit (16 bpp): High color format (5:5:5 - 32,768 colors; 5:6:5 - 65,536 colors)
  - 24 bit (24 bpp): True color format (16,777,216 colors)

Resolution x Color Depth = Memory Size

- Resolution x Color Depth = total bits needed (divide x8 for bytes)
- Framebuffer = memory buffer containing complete frame (bitmap) of data

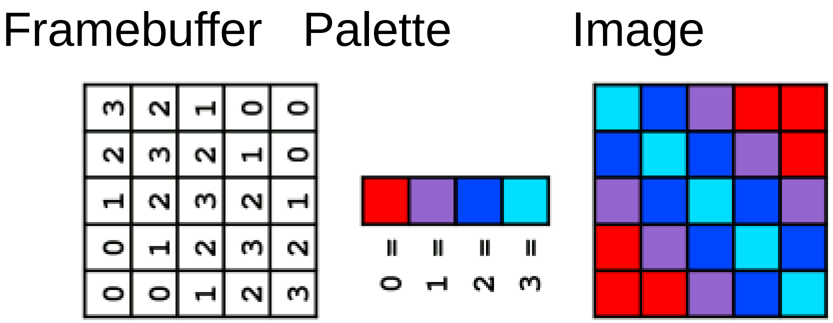
Resolution		1 bits/pixel	2 bits/pixel	4 bits/pixel	8 bits/pixel	16 bits/pixel	24 bits/pixel
XGA	1024x768	98,304	196,608	393,216	786,432	1,572,864	2,359,296
WVGA	800x480	48,000	96,000	192,000	384,000	768,000	1,152,000
VGA	640x480	38,400	76,800	153,600	307,200	614,400	921,600
WQVGA	480x272	16,320	32,640	65,280	130,560	261,120	391,680
QVGA	320x240	9,600	19,200	38,400	76,800	153,600	230,400
CGA	320x200	8,000	16,000	32,000	64,000	128,000	192,000

Example: 480 x 272 x 16bpp x 8bits/byte = 261,120 bytes needed

SEGGER
It simply works!

Palette Based Frame Buffer

- The frame buffer will contain an index value for each pixel
- Palette RAM is pre-filled with 16-bit color value for each index



- NXP microcontrollers have 256 entries to support
  - 1, 2, 4, or 8 bpp palletized color displays for color STN and TFT
  - 1, 2, or 4 bits-per-pixel (bpp) palletized displays for mono STN