AN11244

emWin startup guide

Rev. 1 — 1 August 2012 Application note

Document information

Info	Content
Keywords	emWin, Graphical, LCD, BSP, LPC1769, LPC1788
Abstract	This application note describes the basics of using emWin on NXP microcontrollers.

Revision history

Rev	Date	Description
1	20120801	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

emWin is a software library from Segger which provides an efficient, processor- and LCD controller-independent graphical user interface (GUI) for use with graphical LCDs.

emWin allows the user to easily add graphics to their applications, ranging from:

- Basic drawing functions such as drawing lines, squares, circles and polygons.
- More complex functions such as managing windows, button widgets, listview widgets, edit widgets, etc.
- Displaying a variety of image formats (BMP, PNG, JPEG, etc.).
- Support for multiple displays.
- Support for multiple layers and transparency settings.
- Control of GUI by mouse and touch screen.
- Rapid development, even without targeted hardware, due to support for simulating the GUI in Microsoft Visual Studio.

This application note gives an introduction on using emWin on NXP microcontrollers. It covers the following topics:

- Where to get the software and documentation.
- How to import example project / BSPs, what their directory structure looks like and how to compile them.
- How to create your own GUI using Segger's GUIBuilder.

To give an impression of what is possible when using emWin, Fig 1 shows one of the examples provided by Segger running on Windows after compiling it in Visual Studio.

Fig 1. Example of emWin GUI when compiled for Visual Studio (for simulation / demonstration purposes)

NXP customers can use emWin for free on ARM based microcontrollers. The only limitation is that customers get a pre-compiled library, and not the original source code of emWin. If you do require the original emWin source code for your own project, Segger offers special pricing for NXP customers when upgrading from the NXP emWin library.

2. Where to get it?

As mentioned in the introduction, NXP offers both emWin libraries and emWin Board Support Packages (BSP).

The library packages only offer the compiled emWin library, source template files and the required emWin header files; the user needs to add specific source files to initialize the microcontroller and to allow the microcontroller to interface with the LCD. This option is recommended for advanced users only.

NXP also offers a number of BSPs. These BSPs contain the emWin library, emWin header files, and all source files required to run emWin on one specific development board, with one or more specific displays. These BSPs can be used for both evaluation and development; when porting emWin to another board, these BSPs can be used as a starting point.

All information on emWin and all libraries and BSPs provided by NXP can be found on the emWin page of LPCWare.com:

http://www.lpcware.com/content/project/emwin-graphics-library.

The emWin project page contains the following files / documents:

- Pre-compiled libraries for:
  - IAR Embedded Workbench, LPCXpresso, Keil µVision and Microsoft Visual Studio.
  - ARM7TDMI, ARM926, ARM Cortex M0, M3 & M4.
- A number of Board Support Packages. Supported boards:
  - IAR 1788-SK
  - Embedded Artists' LPC1788 Developer's Kit.
  - Keil MCB1700 (LPC1769).
- Segger's emWin User Manual.
- Useful external resources.
- Application examples.
- NXP Porting Guide, describing how existing BSPs can be ported to custom hardware.

3. Importing, compiling and customizing BSPs

The BSPs are ideal for evaluating emWin and as a starting point when porting to a custom board. This chapter shows how to import these BSPs into your IDE (LPCXpresso, Keil µVision or IAR Embedded Workbench).

3.1 The EA1788 BSP

The EA1788 BSP is the BSP which can be used with the Embedded Artists' LPC1788 Developer's Kit. Three different types of LCDs are supported within the BSP:

- 3.2" 240x320 pixels
- 4.3" 480x272 pixels
- 7" 800x480 pixels

The following is included in the EA1788 BSP:

- emWin library and header files.
- All other required source code.
- A number of sample applications and sample configurations.
- A number of tools which can be used with emWin, e.g. a GUI builder and an image converter.

Project files for LPCXpresso, Keil µVision, IAR Embedded Workbench and Microsoft Visual Studio.

3.2 Getting the EA1788 BSP

The EA1788 BSP can be downloaded from the emWin LPCware.com website. The direct link for downloading the BSP is:

http://www.lpcware.com/content/nxpfile/nxpemwin514bsp

The zip file does not need to be saved to the hard-disk; it is sufficient to simply open it from the web in WinZip or WinRAR.

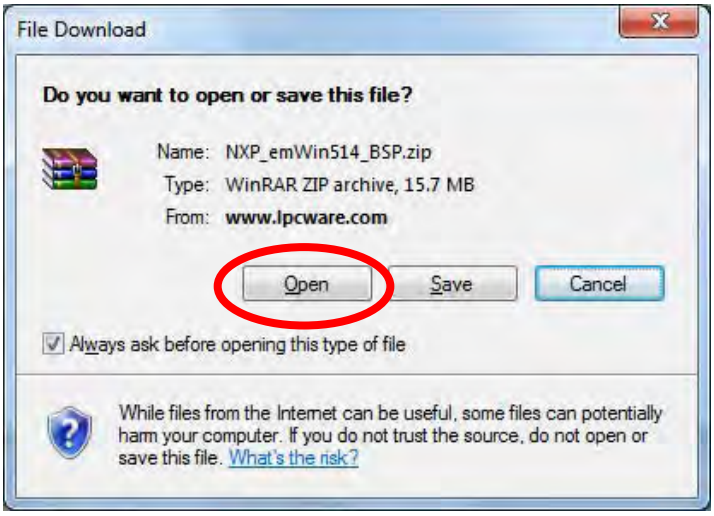

Fig 2. Opening EA1788 BSP

After opening the zip file, the installer can be run by double clicking the "NXP_emWin514_BSP.exe" file.

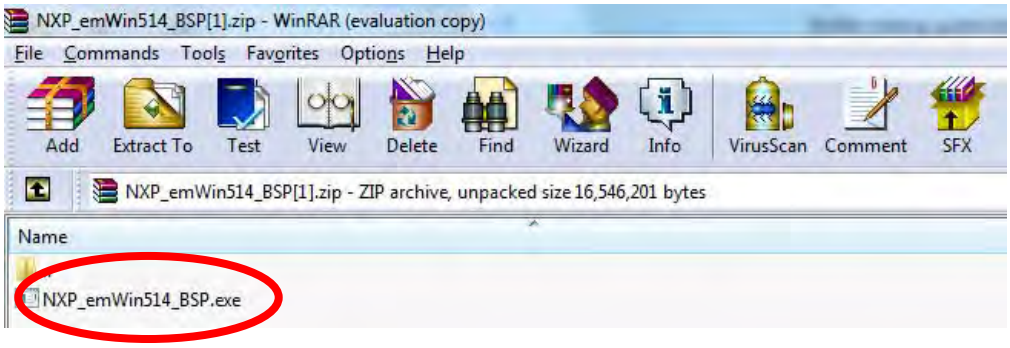

Fig 3. Running the installer

The installer installs a single zip file. By default, it will be installed to "C:\NXP\emWin". The final step is to extract the zip file.

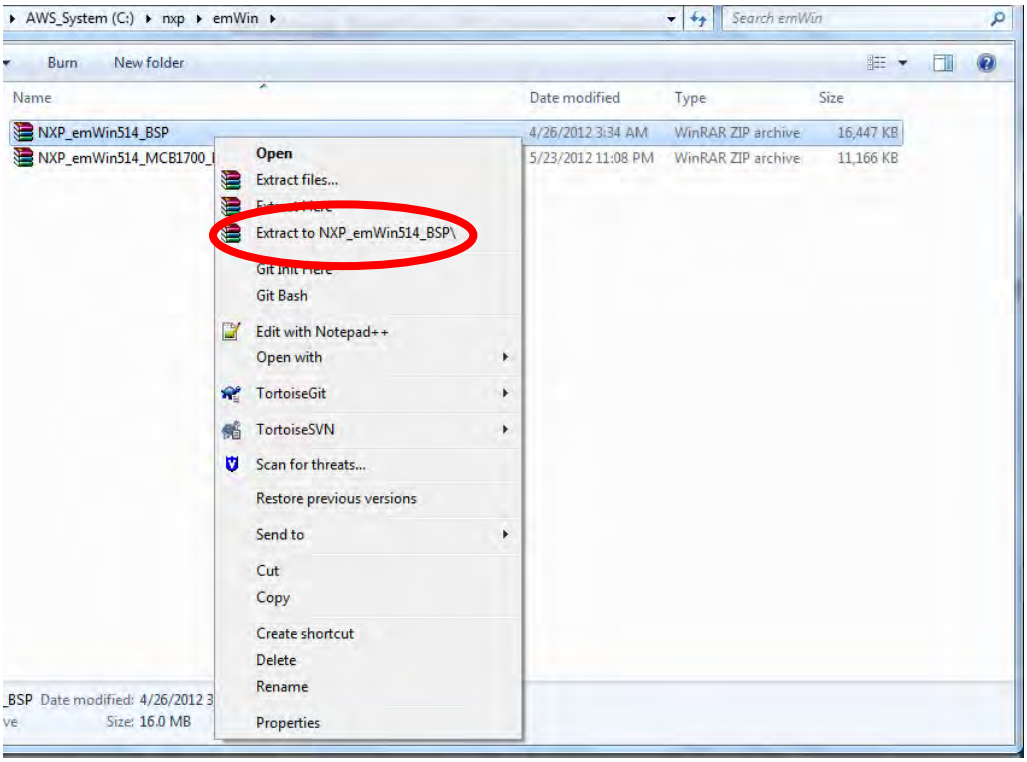

Fig 4. Extracting zip file

3.3 Importing, building and programming the BSP

After extracting the BSP, it can be imported into any of the supported IDEs (LPCXpresso, Keil µVision, IAR Embedded Workbench). Next, the project can be built to generate a binary and finally this binary can be loaded into the target.

3.3.1 LPCXpresso

1. The first step is to import the BSP into LPCXpresso.
   - Open a new workspace (File->Switch Workspace->Other).

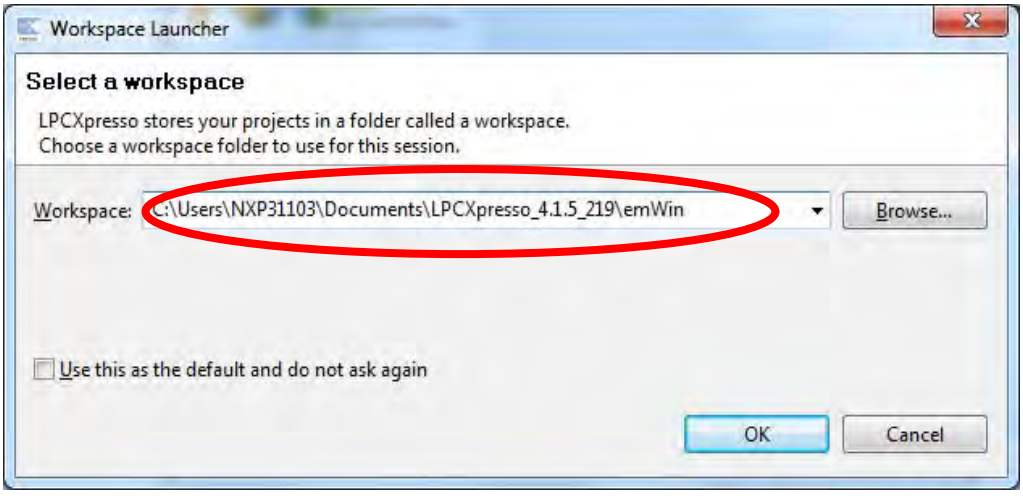

Fig 5. Opening a new workspace

- Import the Start folder of the BSP (“File->Import->General->Existing Projects into Workspace”, point “Select root directory” to the start folder of the BSP and press Finish).

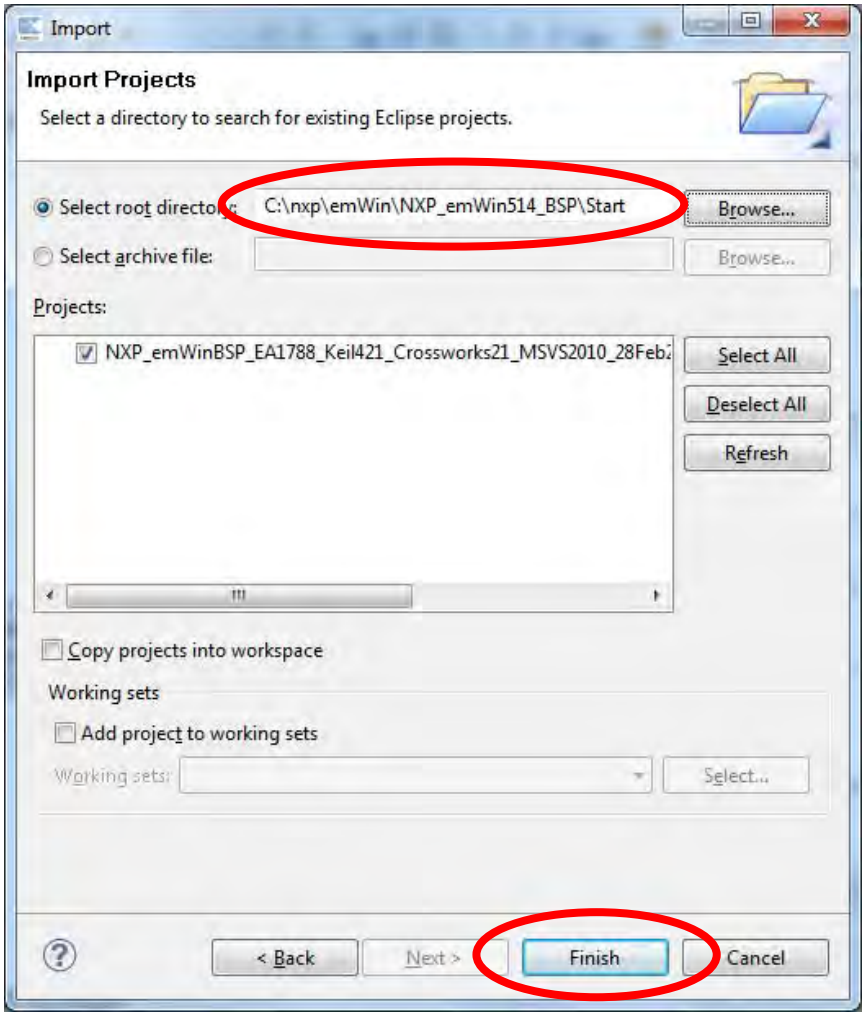

Fig 6. Importing the Start folder

2. After successfully importing the project, it can be compiled. This can be done by clicking Project->Build All, or by pressing Ctrl + B.
3. After successfully building the project, it can be flashed into the target. This can be done by clicking the Debug icon ( ).

3.3.2 Keil µVision

1. The first step is to import the BSP into Keil µVision. This is done by opening the file “NXP_emWin514_EA1788_Keil_CMSIS.uvproj”, which can be found in the Start folder of the BSP.
2. After successfully importing the project, it can be compiled by hitting the Build icon (), or by pressing F7.
3. After successfully building the project, it can be flashed into the target. This can be done by clicking the Start/Stop Debug Session icon ().

3.3.3 IAR Embedded Workbench

1. The first step is to import the BSP into IAR Embedded Workbench. This is done by opening the file “Start_LPC1788_IARLPC1788_IAR_CMSIS_V620a.eww”, which can be found in the Start folder of the BSP.
2. After successfully importing the project, it can be compiled by hitting the Make icon (), or by pressing F7.
3. After successfully building the project, it can be flashed into the target. This can be done by clicking the Download and Debug icon ().

3.3.4 Microsoft Visual Studio

Simulation libraries are provided to allow NXP customers to create emWin projects independently of the hardware. The BSPs can be imported into Microsoft Visual Studio and when completed, be given to others as “.exe” files. No hardware is needed to evaluate the graphics projects.

1. The first step is to import the BSP into Microsoft Visual Studio. This is done by opening the file “Simulation.sln”, which can be found in the Start folder of the BSP.
2. After successfully importing the project, it can be compiled by clicking Debug->Build Solution, or by pressing F7.
3. After successfully building the project, the simulation can be run by clicking the run icon (), or by pressing F5.

4. Understanding the BSPs

After successfully running the BSP with its sample application, you might like to change a few things and develop simple applications yourself.

4.1 Software organization of the EA1788 BSP

Before making any changes, it's important to know what files and folders are present in the BSP. All BSPs have two top-level folders: "Start" & "Doc". The Start folder contains all source- and project files, samples and tools. The Doc folder contains the emWin User Manual by Segger.

Fig 7 displays the files visible in LPCXpresso after importing, which is identical to the files which can be seen in Windows Explorer.

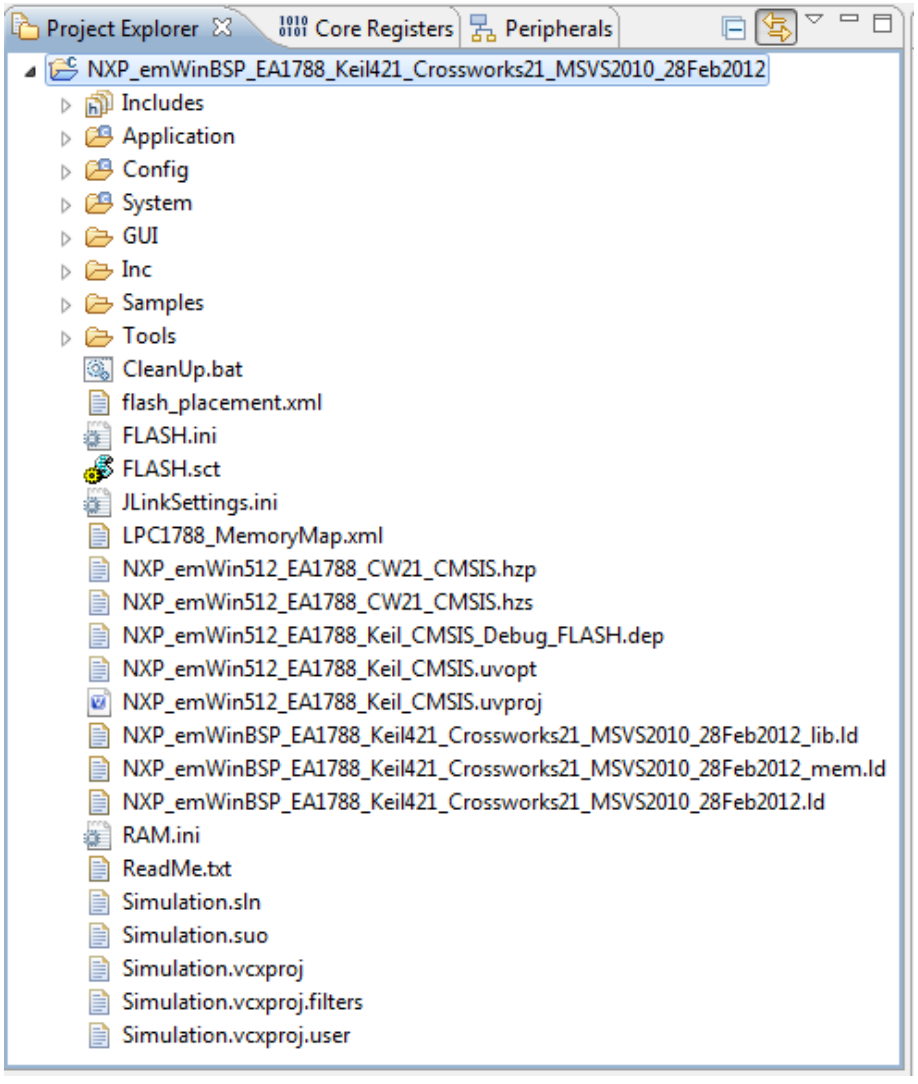

Fig 7. File- and directory structure of the EA1788 BSP

The following is a list of some folders of interest:

- **Application**. This folder contains a number of sample applications. The fairly simple “GraphXYDemo” is set as an active demo.
- **Config**. This folder holds several emWin configuration files to configure the emWin GUI and to control the LCD and to control the cursor (e.g. by touchscreen, joystick, mouse).
- **System**. All hardware specific functions and configurations are present in this folder. It contains the CMSIS files, start-up files and driver software required by the board.
- **GUI**. The GUI folder contains all emWin pre-compiled libraries and the emWin header files.
- **Samples**. This folder contains a large number of sample files. It contains sample applications, sample configurations and sample drivers.
- **Tools**. Contains a number of windows executable files which are useful when developing with emWin, e.g. several conversion programs, a GUI builder and a VNC viewer.

4.2 Setting another sample application as active application

In case of the EA1788 BSP, the GraphXYDemo is set as the standard active application. By changing the active source file in the Application folder in your IDE, another sample application can be loaded into the target. Details on this depend on the IDE. The following example sets the WM_RadialMenu as active application in LPCXpresso.

1. Right-click on file GUI_WIDGET_graphXYDemo.c and select “Resource Configurations -> Exclude from Build”

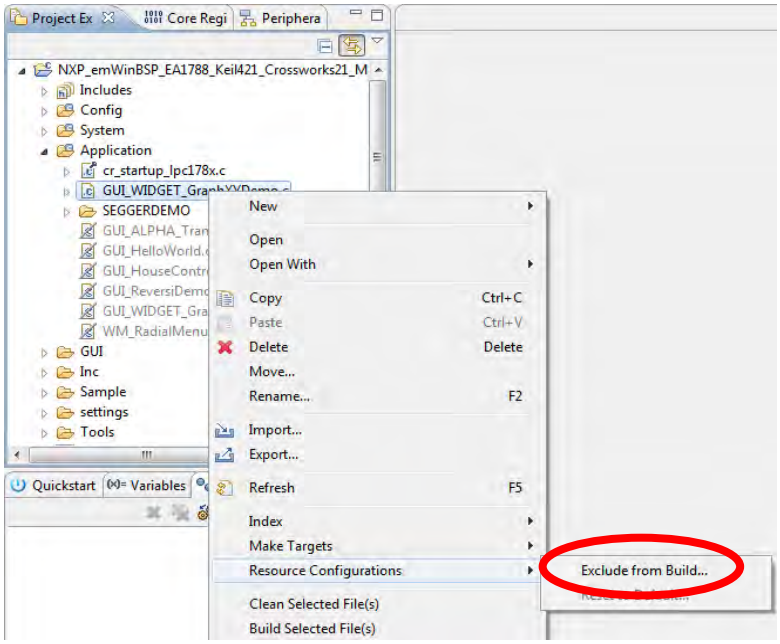

Fig 8. Setting active application

2. In the appearing dialog, click Select All and OK.

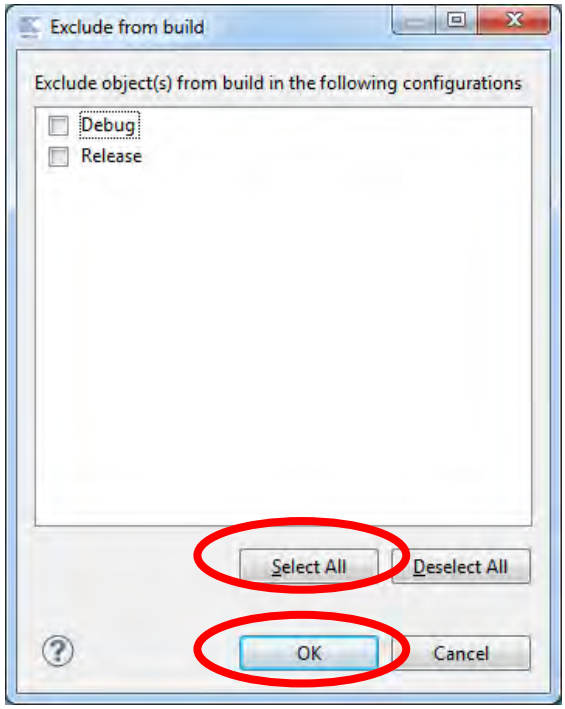

Fig 9. Selecting objects to exclude from build

3. Repeat Steps 1 & 2 for file WM_RadialMenu.c, but instead of choosing the option Select All, choose Deselct All.
4. Rebuild and program the target.

4.3 Using the GUI builder to design your own GUI

Creating your own GUI using emWin is made easy when using Segger's GUIBuilder. The GUIBuilder can be found in the Tools folder of the BSP and allows designing a GUI by simple Drag & Drop of Frames and widgets. Fig 10 shows the GUI builder, while building a simple GUI using emWin GUI builder 5.14.

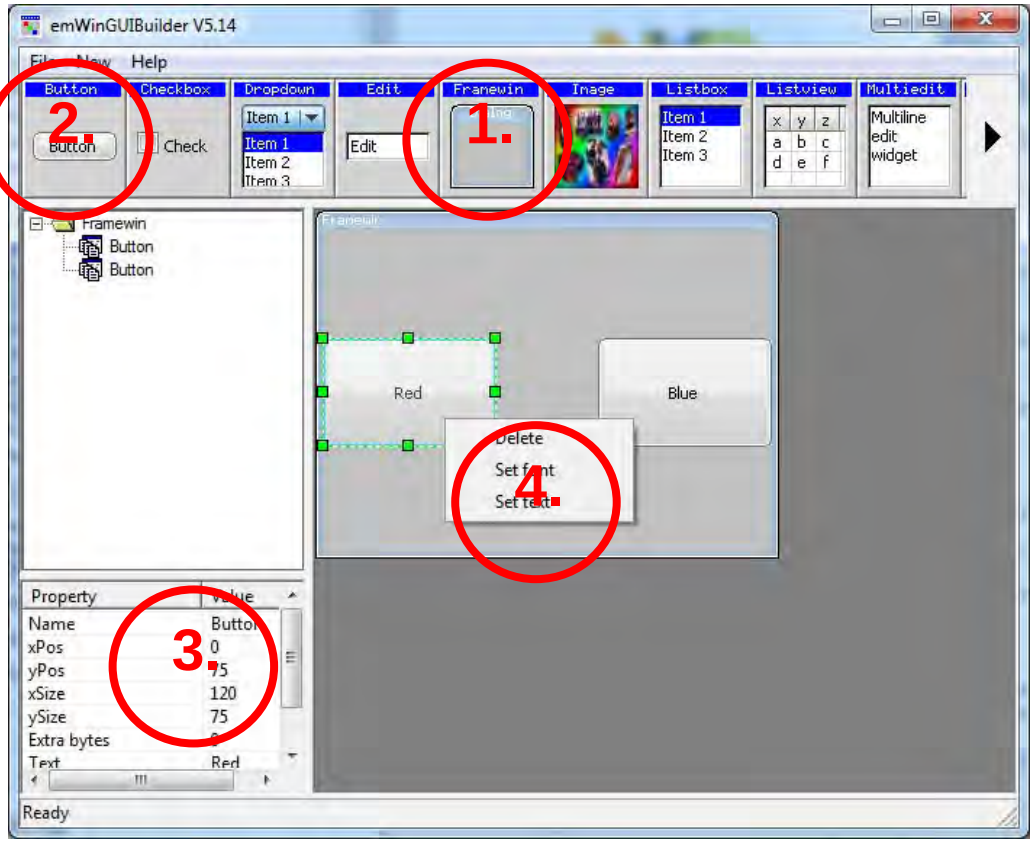

Fig 10. GUIBuilder screenshot while developing a simple GUI

The GUI of Fig 10 is created using the following steps:

1. Create the FrameWin object by clicking the FrameWin icon in the GUIBuilder [1.]. The default size is 320x240.
2. Add two buttons by clicking twice on the Button icon in the GUIBuilder [2.].
3. Change the size of both buttons to 120x75. Change the position of one button to (0, 75) and the other button to (190, 75) [3.].
4. Change the text of one button to “Red” and the text of the other button to “Blue” by right-clicking on the button and choosing the option “Set Text” [4.].
5. Save the GUI by clicking File->Save. This creates the file FramewinDLG.c in the Tools folder.

The newly created GUI can be used in the BSP as follows:

1. Disable the active sample application in the project (Chapter 4.2).
2. Add the FramewinDLG.c file to the project. In LPCXpresso this is done by simply copying the FramewinDLG.c file to the Application folder and refreshing the project (F5).

3. On the bottom of file FramewinDLG.c, in section “Public code”, add the following lines between “// USER START” and “//USER END”:

```
void MainTask(void);
void MainTask(void) {
  GUI_Init();
  CreateFramewin();
  while(1) {
    GUI_Delay(500);
  }
}
```

4. Rebuild the project and flash it into the target.

If the above steps are done correctly, the target should now display the designed GUI.

As a final step, let’s add functionality to the two buttons, e.g. change the color of the window red when the button labeled “Red” is pressed and change the color to blue when the button labeled “Blue” is pressed.

In file FramewinDLG.c, locate function “_cbDialog”. This function is a callback function, used for initializing the window and its widgets and for handling events, e.g. when a button is pressed. By adding lines of code at the “WM_NOTIFICATION_RELEASED” event of each button, the action of the buttons can be defined. Changing the color of the window can be done by using the FRAMEWIN_SetClientColor() function. Refer to the code below to see the implementation of adding this functionality to the buttons.

```
1    switch(Id) {
2    case ID_BUTTON_0: // Notifications sent by 'Button'
3      switch(NCode) {
4      case WM_NOTIFICATION_CLICKED:
5        // USER START (Optionally insert code for reacting on notification message)
6        // USER END
7        break;
8      case WM_NOTIFICATION_RELEASED:
9        // USER START (Optionally insert code for reacting on notification message)
10       FRAMEWIN_SetClientColor(pMsg->hWin, GUI_RED);
11       // USER END
12       break;
13     // USER START (Optionally insert additional code for further notification
     handling)
14     // USER END
15       }
16       break;
17   case ID_BUTTON_1: // Notifications sent by 'Button'
18     switch(NCode) {
19     case WM_NOTIFICATION_CLICKED:
20       // USER START (Optionally insert code for reacting on notification message)
21       // USER END
22       break;
23     case WM_NOTIFICATION_RELEASED:
24       // USER START (Optionally insert code for reacting on notification message)
25       FRAMEWIN_SetClientColor(pMsg->hWin, GUI_BLUE);
26       // USER END
27       break;
```

```
28      // USER START (Optionally insert additional code for further notification
      handling)
29      // USER END
30      }
31      break;
```

4.4 Porting: Next step when using emWin

NXP offers BSPs for a number of boards. However, emWin is often required to run on a custom board instead of any of the supported evaluation boards. This can be done by porting any of the already existing BSPs to the new target hardware. To aid you in doing so, a porting guide (AN11218) is available. This porting guide shows step-by-step how one BSP (EA1788) was ported to another BSP (MCB1700). The porting guide can be found on the LPCWare.com emWin page:

http://www.lpcware.com/content/project/emwin-graphics-library

5. References

[1] http://www.segger.com/emwin.html

[2] Segger emWin 5 User Manual, Rev. 1, 28 June 2012, http://www.segger.com/admin/uploads/productDocs/UM03001_emWin5.pdf

[3] http://www.lpcware.com/content/project/emwin-graphics-library

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

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

Date of release: 1 August 2012
Document identifier: AN11244