GUI emWin Start Guide

V1.00.003

Publication Release Date: Sep. 2018

Support Chips:

N9H20 Series

Support Platforms:

Non-OS

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

Table of Contents

1. Introduction

1.1. Introduction

emWin is a graphic library with graphical user interface (GUI). It is designed to provide an efficient, processor- and display controller-independent graphical user interface (GUI) for any application that operates with a graphical display.

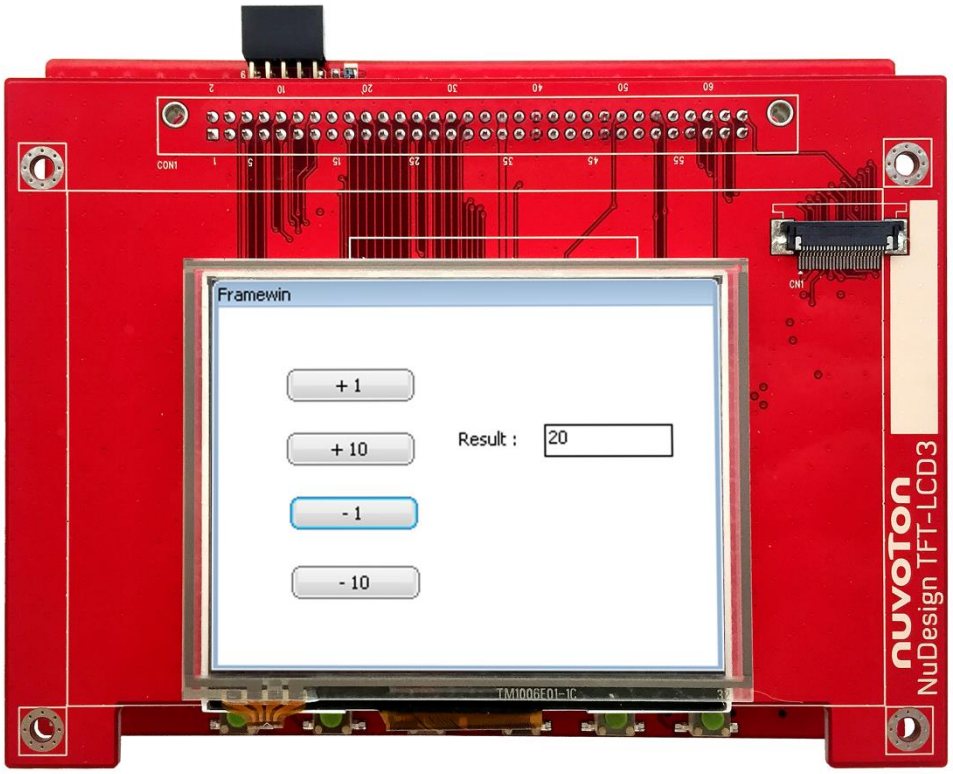

Figure 1.1-1 emWin runs on N9H20.

N9H20 BSP includes emWin related materials, e. g., sample codes, library, documents and tools. We can develop emWin applications on Keil MDK incudes IDE, compiler and debugger (embedded development tools for Arm) to build, modify and debug. We'll introduce the basic operations on section 2.4.

1.2. emWin Folder structure

N9H20 BSP contains emWin related materials and the folder structure is shown below:

Directory Name	Content
SampleCode \ emWin	Two emWin samples: 1. GUIDemo. 2. SimpleDemo.
ThirdParty \ emWin \ Config	emWin configuration files.
ThirdParty \ emWin \ Doc	Two emWin official documents: 1. AN03002_Custom_Widget_Type.pdf. 2. UM03001_emWin5.pdf.
ThirdParty \ emWin \ Include	emWin include files.
ThirdParty \ emWin \ Lib	emWin library: NUemWin_ARM9_Keil.lib.
ThirdParty \ emWin \ Tool	emWin tools.

2. Start emWin

2.1. Step 1: Open project

Double click “SimpleDemo.uvproj” (the path is in “SampleCode\emWin\SimpleDemo\KEIL”) to open project.

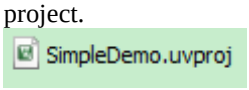

Figure 2.1-1 “SimpleDemo” project file.

“SimpleDemo” is a sample code to utilize emWin library to demonstrate interactive feature. It contains a frame window, four buttons, a text and a text editor. We can touch the GUI button and check the result that shown on the text editor.

The path of “SimpleDemo” is in “SampleCode\emWin”.

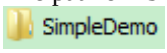

Figure 2.1-2 “SimpleDemo” sample folder is in “SampleCode\emWin”.

The structure of “SimpleDemo”:

- Blue part is related with BSP.
- Red part is related with emWin.

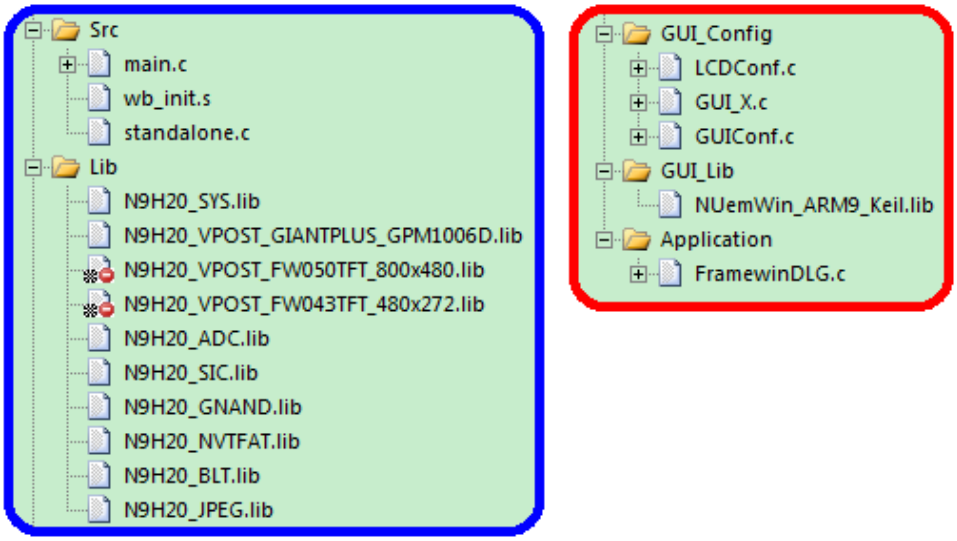

Figure 2.1-3 “SimpleDemo” project structure.

2.2. Step 2: BSP Initialization

In “main.c”, it contains the N9H20 start up flow, e.g., clock setting, timer, uart debug port, display output panel, vendor filesystem and resistor-type touch screen.

The path of “main.c” is in “SampleCode\emWin\SimpleDemo”:

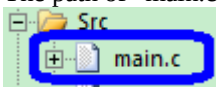

Figure 2.2-1 “main.c” contains N9H20 start up flow.

```
int main(void)
{
...
/* N9H20 start up here */
...

/* emWin start up here */
MainTask();

return 0;
}
```

2.3. Step 3: emWin Initialization

In “main.c”, called “MainTask()” to start up emWin GUI system.

“MainTask()” is in “SampleCode\emWin\SimpleDemo\main.c”:

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

2.4. Step 4: Build

To download and run the application, first, we need to utilize Keil to rebuild the application project.

Press “[F7]” to rebuild the application project or click “Rebuild”.

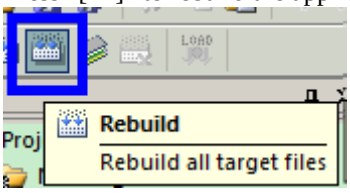

Figure 2.4-1 Rebuild application project.

2.5. Before download and run

IMPORTANT!!! We need to configurate ICE setting to download and run.

Press "[Alt + F7]" for project options.

Figure 2.5-1 Options for target.

Choose "Debug" page and select properly ICE, e. g., J-LINK, then press "Settings".

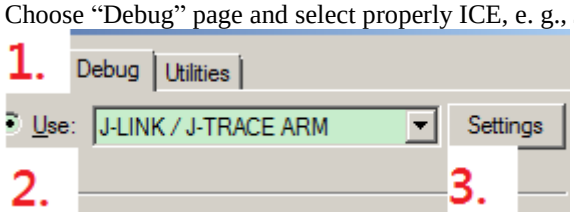

Figure 2.5-2 Select ICE to debug.

Set "Speed" (Auto or lower speed), disable "Debug Cache", then select "Reset Strategy as Hardware, halt after reset (normal)", finally, click "OK".

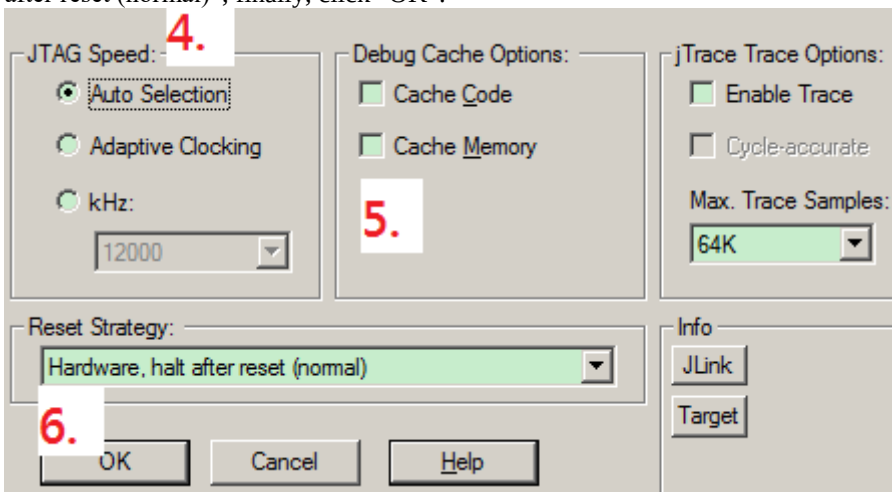

Figure 2.5-3 ICE environment setting.

2.6. Step 5: Download and run

Press “CTRL + [F5]” to download the application and run debug session. After downloaded, it will halt at main() and we should see the similar screenshow below.

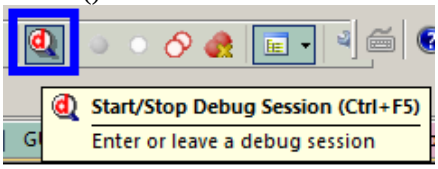

Figure 2.6-1 Download and run application.

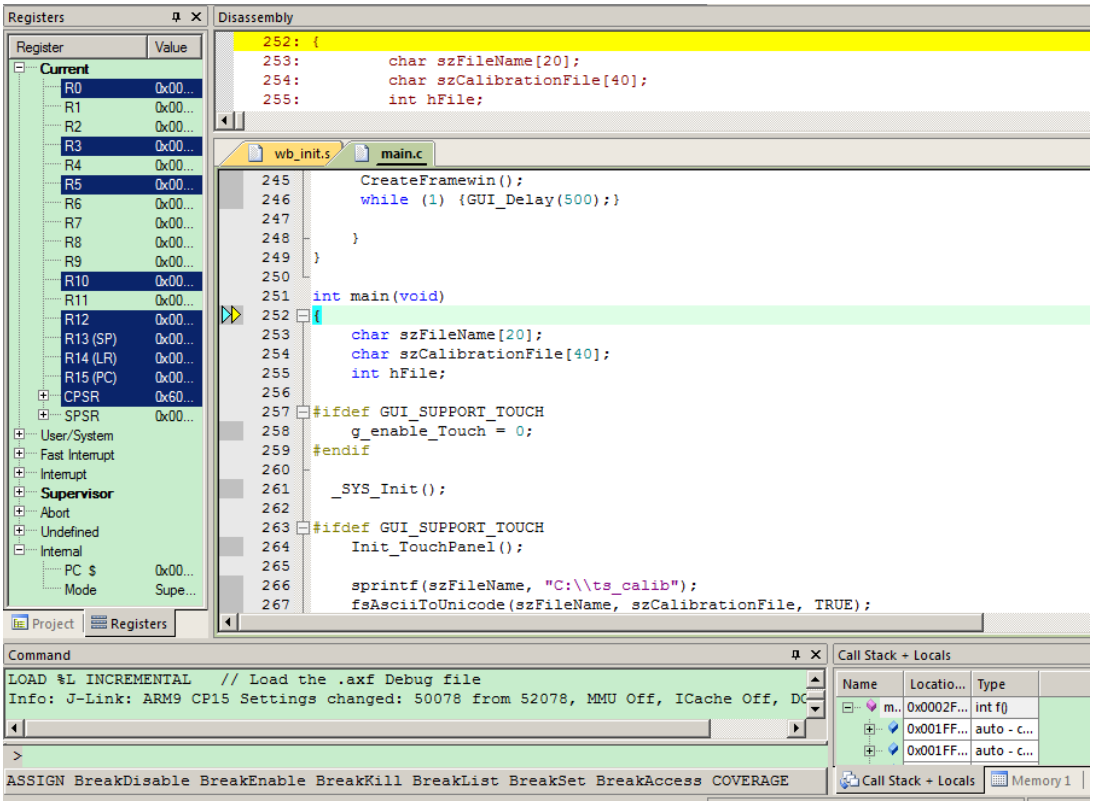

Figure 2.6-2 Debug session.

2.7. Touch screen

To control touch panel, we utilize N9H20 ADC library and open source library “tslib”. The touch calibration results store to a single file called “ts_calib”.

The path of “tslib” is in “SampleCode\emWin\SimpleDemo”:

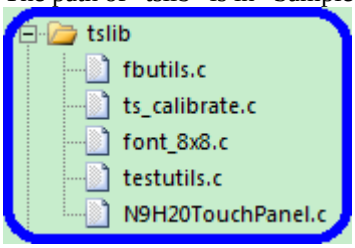

Figure 2.7-1 tslib structure.

3. Start emWin GUIBuilder

3.1. Step 1: Create widget

To create widget, we can utilize emWin “GUIBuilder” to arrange GUI layout and generate source file.

The path of “GUIBuilder” is in “ThirdParty\emWin\Tool”:

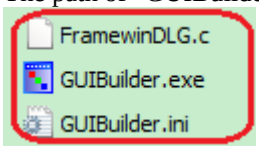

Figure 3.1-1 emWin GUIBuilder.

After finish GUI layout, then execute “File” → “Save…”, we can get the source file called “FramewinDLG.c”.

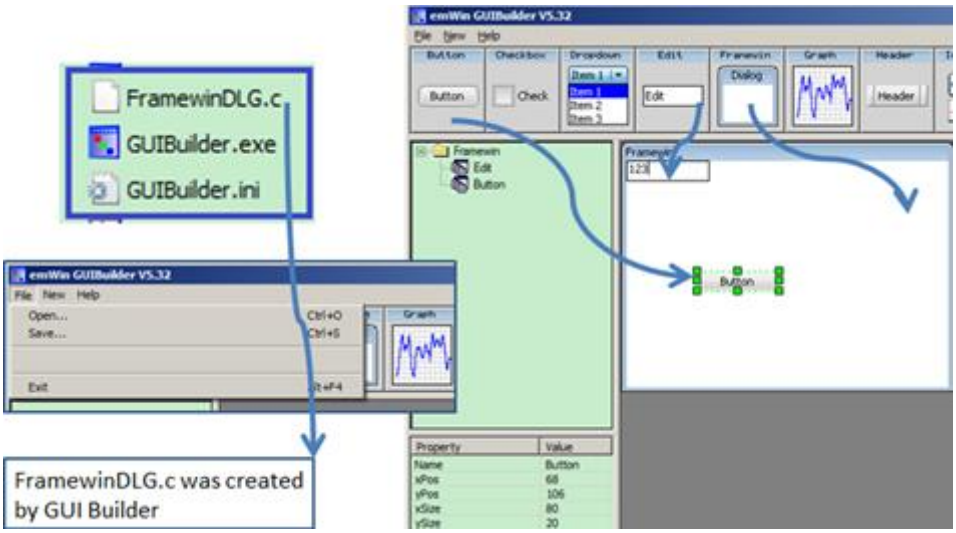

Figure 3.1-2 emWin GUIBuilder can arrange GUI layout and generate source file.

3.2. Step 2: Handle widget event

In “FramewinDLG.c”, we can handle widget event, e. g., button clicked, released or others and update text editor’s content when button released.

The path of “FramewinDLG.c” is in “SampleCode\emWin\SimpleDemo\Application”:

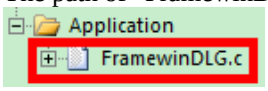

Figure 3.2-1 emWin GUI application source file.

```
...
switch(Id)
{
case ID_BUTTON_0: // Notifications sent by '+ 1'
switch(NCode)
{
case WM_NOTIFICATION_CLICKED:
// USER START (Optionally insert code for reacting on notification message)
// USER END
break;
case WM_NOTIFICATION_RELEASED:
// USER START (Optionally insert code for reacting on notification message)
value += 1;
sprintf(sBuf,"%d   ", value);
hItem = WM_GetDialogItem(pMsg->hWin, ID_EDIT_0);
EDIT_SetText(hItem, sBuf);
// USER END
break;
// USER START (Optionally insert additional code for further notification
handling)
// USER END
}
break;
...
```

4. How to change display panel

4.1. Step 1: emWin display

"LCDConf.c" defines emWin display configurations.

The path of "LCDConf.c" is in "ThirdParty\emWin\Config":

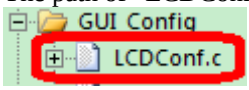

Figure 4.1-1 emWin display configurations.

- Display panel resolution:
  In "N9H20TouchPanel.h", we can modify the "XSIZE_PHYS" and "YSIZE_PHYS" for display panel width and height respectively. Please note that "XSIZE_PHYS" and YSIZE_PHYS" are defined in "SampleCode\emWin\SimpleDemo\tslib\N9H20TouchPanel.h".

```
...
#define XSIZE_PHYS 320
#define YSIZE_PHYS 240
...
```

- Display buffer address:
  In "LCDConf.c", we can assign display buffer address to emWin. Here, we utilize a "Sync-type LCD 320x240", the display buffer size in RGB565 is 320-width x 240-height x 2-byte-per-pixel = 153600Bytes = 150KB.

```
void LCD_X_Config(void)
{
...
/* assign display buffer address to emWin */
LCD_SetVRAMAddrEx(0, (void *)u8FrameBufPtr);
...
}
```

4.2. Step 2: BSP display

N9H20 can utilize VPOST library to output display data to “Sync-type LCD”. N9H20 BSP contains the default display library for 320 x 240 at 16-bit depth RGB565.

The path of the default display library is in “Library\IPLib\N9H20_GIANTPLUS_GPM1006D.lib“.

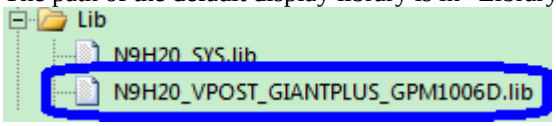

Figure 4.2-1 BSP default display library.

```
void LCD_X_Config(void)
{
...
/* assign display buffer address to emWin */
LCD_SetVRAMAddrEx(0, (void *)u8FrameBufPtr);
...
}
```

5. Revision History

Version	Date	Description
V1.00.003	Sep. 13, 2018	● Update source path and description.
V1.00.002	Aug. 17, 2018	● Update introduction for Keil MDK
V1.00.001	Mar. 30, 2018	● Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.