M480 emWin Display Text

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This example code utilizes API of emWin to show different drawing mode on the panel.
BSP Version	M480 Series BSP CMSIS V3.03.001
Hardware	NuMaker-PFM-M487 V3.0 & M487 Advance Ver4.0

1 Function Description

1.1 Introduction

Nuvoton has license emWin library from Segger. It is really simple to use API of emWin to develop human machine interface(HMI) on the product. This example code uses M480 MCU and utilizes API of emWin to show text on the panel. emWin can easily implement how to change a lot of drawing mode on the NuMaker-PFM-M487 V3.0 board.

For more information on the NuMaker-PFM-M487 board and NuMaker-M487-Advance board platform for emWin, please visit Nuvoton website to download the User Manual document UM_NuMaker-PFM_M487_User_Manual and UM_NuMaker_M487_Advance_User_Manual, more detail about emWin. Please download the UM_NuMaker-emWin-M487_User_Manual:

https://www.nuvoton.com/hq/products/microcontrollers/arm-cortex-m4-mcus/User-Manual/?__locale=en&resourcePage=Y&category=&pageIndex=3

NuMaker emWin M487 Board

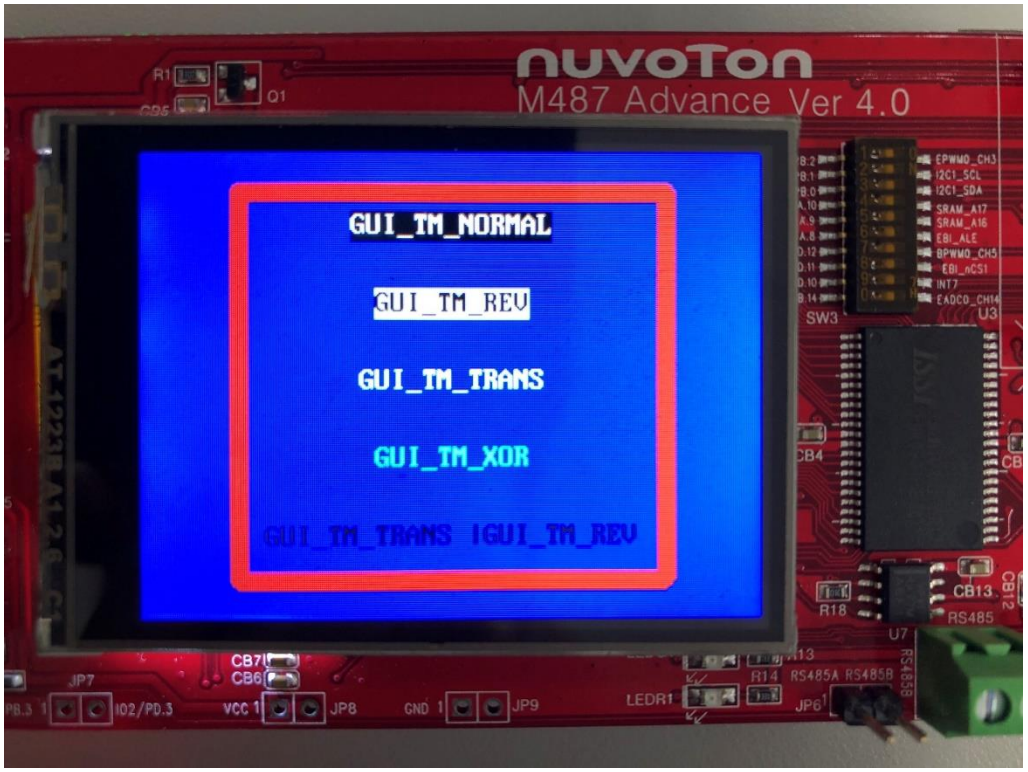
1.2 Principle

Text can be displayed just by calling GUI_DispString(). For example: GUI_DispString("Hello world!");. The example will display the text "Hello world" at the current text position. There are functions to display text using different fonts or at certain positions.
Text is displayed using the foreground and background color which are set using the functions GUI_SetColor() and GUI_SetBkColor(). In order to set a certain drawing mode to display strings or single characters the function GUI_SetTextMode() can be called using the flags described below:

Normal text
Displaying normal text is the default behavior. The characters are displayed using the foreground color. The background color is used to clear the background according to its width of the text and height of the currently selected font. Text can be displayed normally by specifying just GUI_TM_NORMAL or 0.
Reverse text
Text can be displayed in reverse mode by specifying GUI_TM_REV. This causes characters to be displayed using the background color and the background to be cleared using the foreground color.
Transparent text
Text can be displayed in transparent mode by specifying GUI_TM_TRANS. This causes characters to be displayed without the background to be cleared. In this case whatever was drawn before the text was displayed can still be seen.
XOR text
Text can be displayed in XOR mode by specifying GUI_TM_XOR. This causes characters to be displayed using the inverted colors of the background. This is done pixel wise. This is also a transparent drawing mode, so the background remains unchanged. This mode is often used to in 1bpp-configurations to ensure readability, since black is inverted to white and vice versa.

1.3 Demo Result

After executed this example code, user will be seeing five drawing mode on the panel.



2 Code Description

This main() function is used to initialize platform for emWin include PLL clock to 192MHz, Uart clock, EBI, Timer and multi-function pins.

```
/**
 * @brief       main
 *
 * @param[in]   None
 *
 * @return      None
 *
 * @details     This function is used to initialize platform for emWin
 */

void main(void)
{
    /* Init System, IP clock and multi-function I/O */
    _SYS_Init();

    /* Init UART to 115200-8n1 for print message */
    UART_Open(UART0, 115200);

    /* Enable Timer0 clock and select Timer0 clock source */
    CLK_EnableModuleClock(TMR0_MODULE);
    CLK_SetModuleClock(TMR0_MODULE, CLK_CLKSEL1_TMR0SEL_HXT, 0);

     /* Initial Timer0 to periodic mode with 1000Hz */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 1000);

     /* Enable Timer0 interrupt */
    TIMER_EnableInt(TIMER0);
    NVIC_SetPriority(TMR0_IRQn, 1);
    NVIC_EnableIRQ(TMR0_IRQn);

    /* Start Timer0 */
    TIMER_Start(TIMER0);
    /* Start application */
    GUI_Main();
    while(1);
```

This GUI_Main () function is used to emWin API to show text. First, GUI_Init() function is initialized for setting regarding emWin configuration and display driver link and emWin initialization in GUI_Main () function. Second, the background color of display is used to GUI_SetBkColor function then refresh panel need to use GUI_Clear() function.

```
void GUI_Main(void)
{
    GUI_Init();
    /*Set the current background color. */
    GUI_SetBkColor(GUI_BLUE);
    GUI_Clear();
    /*Sets the pen size in pixels. */
    GUI_SetPenSize(10);
    /*Set the current foreground color. */
    GUI_SetColor(GUI_RED);
    /*Draws a line from a specified start point to a pacified end point*/
    GUI_DrawLine(50, 20, 270, 20);
    GUI_DrawLine(50, 220, 270, 220);
    GUI_DrawLine(50, 20, 50, 220);
    GUI_DrawLine(270, 20, 270, 220);
    /*Sets the current font*/
    GUI_SetFont(&GUI_Font8x16);
    GUI_SetBkColor(GUI_BLACK);
    GUI_SetColor(GUI_WHITE);
    /*Sets the drawing mode. */
    GUI_SetTextMode(GUI_TM_NORMAL);
    /*Displays a string centered horizontally at the given position. */
    GUI_DispStringHCenterAt("GUI_TM_NORMAL" , 160, 30);

    GUI_SetTextMode(GUI_TM_REV);
    GUI_DispStringHCenterAt("GUI_TM_REV" , 160, 70);

    GUI_SetTextMode(GUI_TM_TRANS);
    GUI_DispStringHCenterAt("GUI_TM_TRANS" , 160, 110);

    GUI_SetTextMode(GUI_TM_XOR);
    GUI_DispStringHCenterAt("GUI_TM_XOR" , 160, 150);

    GUI_SetTextMode(GUI_TM_TRANS | GUI_TM_REV);
    GUI_DispStringHCenterAt("GUI_TM_TRANS |GUI_TM_REV", 160, 190);
}
```

GUI_SetBkColor()

Description

Sets the default background color.

Prototype

```
GUI_COLOR GUI_SetBkColor(GUI_COLOR Color);
```

Parameter	Description
Color	Color for background, 24-bit RGB value.

GUI_SetPenSize()

Description

Sets the pen size to be used for further drawing operations.

Prototype

```
U8 GUI_SetPenSize(U8 PenSize);
```

Parameter	Description
PenSize	Pen size in pixels to be used.

GUI_SetColor()

Description

Sets the current foreground color.

Prototype

```
void GUI_SetColor(GUI_COLOR Color);
```

Parameter	Description
Color	Color for foreground, 24-bit RGB value.

GUI_DrawLine()

Description

Draws a line from a specified starting point to a specified endpoint in the current window (absolute coordinates).

Prototype

```
void GUI_DrawLine(int x0, int y0, int x1, int y1);
```

Parameter	Description
x0	X-starting position.
y0	Y-starting position.
x1	X-end position.
y1	Y-end position.

GUI_SetFont()

Description

Sets the font to be used for text output.

Prototype

```
const GUI_FONT * GUI_SetFont(const GUI_FONT * pNewFont);
```

Parameter	Description
pFont	Pointer to the font to be selected and used.

GUI_SetTextMode()

Description

Sets the text mode to the parameter specified.

Prototype

```
int GUI_SetTextMode(int TextMode);
```

Parameter	Description
TextMode	Text mode to set. May be any combination of the TEXTMODE flags.

GUI_DispStringHCenterAt()

Description

Displays the string passed as parameter horizontally centered at a specified position in the current window using the current font.

Prototype

```
void GUI_DispStringHCenterAt(const char * s, int x, int y);
```

Parameter	Description
s	String to display.
x	X-position to write to in pixels of the client window.
y	Y-position to write to in pixels of the client window.

3 Software and Hardware Environment

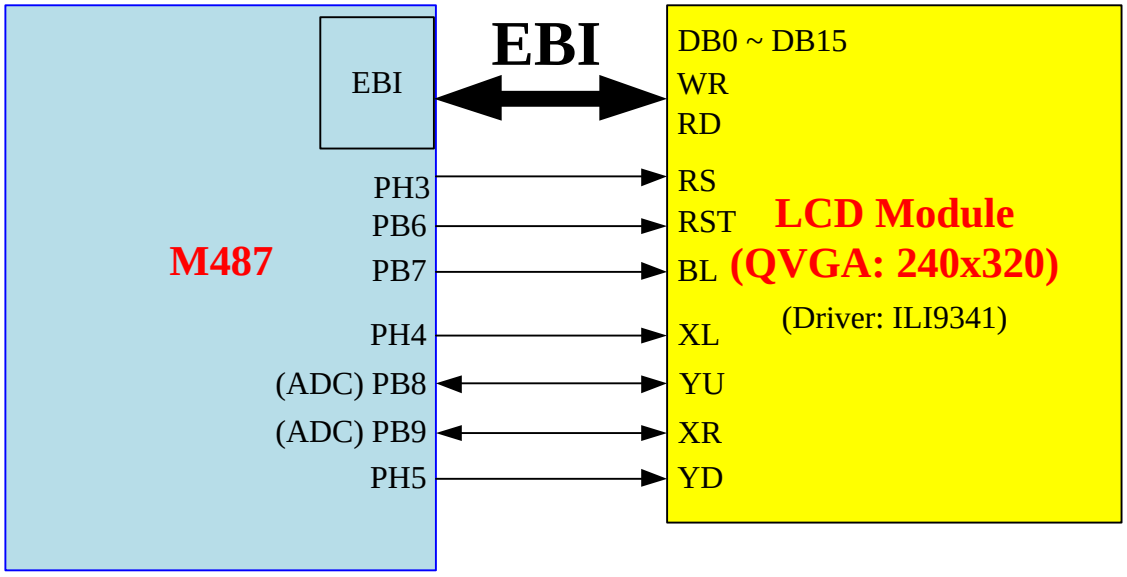
Software Environment

- BSP version
  - M480 Series BSP CMSIS V3.03.001
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuMaker-PFM-M487 V3.0 board
  - M487 Advance Ver4.0
- Diagram

The below diagram is to show how to connect LCD module with M487 including EBI interface for communication, determination data or command by RS pin, module reset by RST pin, backlight control by BL pin and two ADC pin for measuring x and y coordinate。



4 Directory Information

- EC_M480_emWin_SimpleDemo_DisplayText_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code
  - ThirdParty
    - emWin — emWin is designed to provide an efficient, processor- and display controller-independent graphical user interface for any application that operates with a graphical display.

5 How to Execute Example Code?

1. Browsing into sample code folder by Directory Information (section 4) and double click emWin_SimpleDemo.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Jul. 18, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.