M258 LCD Loading and Short Circuit Detection

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code uses charge pump timeout to calculate the current overall load value of the LCD, and then scans based on the timeout value to detect whether the LCD pins are abnormal.
BSP Version	M251/M252/M254/M256/M258 Series CMSIS BSP V3.01.002
Hardware	NuMaker-M258KE Ver1.1

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Overview

This example code "LCD_Loading_And_Short_Circuit_Detection" mainly uses the current maximum time of the charge pump timeout to infer the current overall LCD overall driving load. Then use the charge pump timeout time as the timeout time for scanning each Segment Pin and Common Pin to test whether the current LCD drive load is abnormal and whether there is a short circuit.

1.1 Principle

This example code mainly uses the charge pump timeout function as a load test tool to detect whether the Segment Pin and Common Pin are abnormal. First, keep trying to test until you find the maximum charge pump timeout time, which is the minimum boost time of the LCD current drive voltage. Use this timeout value as the reference value of the entire LCD drive load, and then use this reference value as the test value. With this test value, you can scan all Segment Pins and Common Pins for abnormal load. However, for the convenience of testing, this example code defines the "USING_TEST_CPT_THRD" threshold, which is the average value obtained by measuring several normal LCDs.

First, turn off all Segment Pin outputs. After that, scan and test one by one to confirm whether there is charge pump timeout. If it is, it means that the segment pin is not driving the load normally, and its line may be short-circuited. Otherwise, it means that the current LCD drive load is normal, that is, the output lines of all segment pins are normal. In the same way, the common pin scan detection method is the same as the segment pin.

Since this example code only provides a simple anomaly detection method, it only supports error checking for Segment Pin and Common Pin anomalies. If you want to continue scanning detection, you need to clear the current abnormality first, and then repeat the above-mentioned inspection method. Users can try to adjust the charge pump voltage level of the LCD, different transmission pulse types, different external pull-down resistors (10K~100K ohms or more), etc. Due to different settings, different charge pump timeout values will appear, which means that the size of the LCD drive load is changing. These changes can be used as a basis for users to find more accurate detection methods.

1.2 Demo Result

Burn the "LCD_Loading_And_Short_Circuit_Detection" project program into the NuMaker-M258KE Ver1.1 board, and confirm whether all hardware peripherals are as shown in Figure 3-1. Enter the Debug mode of the Keil project to enable the Semihost function, enable the Serial Window function to print test information, and send the corresponding characters according to the prompts.

Get the test information of the current LCD Panel charge pump timeout value, as shown in Figure 1-1.

```
LCD configurations:
 * Clock source is LIRC
 * 8 COM, 40 SEG and 1/4 Bias
 * Driving waveform is Type-A
 * Target frame rate is 64Hz

Working frame rate is 64Hz on Type-A.

 * Trigger the charge pump timeout threshold is 0x3
 * ------------------------------------------
 * [1]: Get the current charge pump timeout value(using active pulse)
 * [2]: Scan all segment pins for short circuit detection
 * [3]: Scan all common pins for short circuit detection
 Fail disable all pin & test again
 If work normaly, turn segment/common pin by pin
Test Item = 1




Get max charge pump time value = 0x3
If the test data is large, it is recommended to reduce it by 25%.
Suggest -50 percent is 0x2
Suggest -25 percent is 0x3
```

Figure 1-1 Get the Current Execution Result and Print Information of the LCD Charge Pump Timeout Value

LCD Segment 5 Pin external pull-down resistance test information is shown in Figure1- 2.

```
CPU @ 48000000 Hz
+----------------------------------------+
|    LCD Loading Detector Example Code   |
+----------------------------------------+

LCD configurations:
 * Clock source is LIRC
 * 8 COM, 40 SEG and 1/4 Bias
 * Driving waveform is Type-A
 * Target frame rate is 64Hz

Working frame rate is 64Hz on Type-A.

 * Trigger the charge pump timeout threshold is 0x3
 * ----------------------------------------
 * [1]: Get the current charge pump timeout value(using active pulse)
 * [2]: Scan all segment pins for short circuit detection
 * [3]: Scan all common pins for short circuit detection
 Fail disable all pin & test again
 If work normaly, turn segment/common pin by pin
Test Item = 2
 * Scan all segment pins and check for short circuits
 * LCD segment pin status is abnormal, and check again
 * One by one detect LCD segment pin!
 * Enable LCD SEG0 - OK
 * Enable LCD SEG1 - OK
 * Enable LCD SEG2 - OK
 * Enable LCD SEG3 - OK
 * Enable LCD SEG4 - OK
 * Enable LCD SEG5 - Short circuit detected!
```

Figure1- 2 Segment 5 Pin External Pull Low Resistance Execution Results, and Print Information

The LCD Common 1 Pin external pull low resistance is shown in Figure1- 3

```
LCD configurations:
 * Clock source is LIRC
 * 8 COM, 40 SEG and 1/4 Bias
 * Driving waveform is Type-A
 * Target frame rate is 64Hz

Working frame rate is 64Hz on Type-A.

 * Trigger the charge pump timeout threshold is 0x3
 * ----------------------------------------
 * [1]: Get the current charge pump timeout value(using active pulse)
 * [2]: Scan all segment pins for short circuit detection
 * [3]: Scan all common pins for short circuit detection
 Fail disable all pin & test again
 If work normaly, turn segment/common pin by pin
Test Item = 3
 * Scan all common pins and check for short circuits
 * LCD common pin status is abnormal, and check again
 * One by one detect LCD common pin!
 * Enable LCD COM0 - OK
 * Enable LCD COM1 - Short circuit detected!
```

Figure1- 3 LCD Common 1 Pin External Pull Low Resistance Execution Result, and Print Information

The test information of LCD Segment 10 Pin shorted to ICE's UART_Tx pin is shown in Figure1- 4

```
LCD configurations:
 * Clock source is LIRC
 * 8 COM, 40 SEG and 1/4 Bias
 * Driving waveform is Type-A
 * Target frame rate is 64Hz

Working frame rate is 64Hz on Type-A.

 * Trigger the charge pump timeout threshold is 0x3
 * ----------------------------------------
 * [1]: Get the current charge pump timeout value(using active pulse)
 * [2]: Scan all segment pins for short circuit detection
 * [3]: Scan all common pins for short circuit detection
 Fail disable all pin & test again
 If work normaly, turn segment/common pin by pin
Test Item = 3
 * Scan all common pins and check for short circuits
 * LCD common pin status is abnormal, and check again
 * One by one detect LCD common pin!
 * Enable LCD COM0 - OK
 * Enable LCD COM1 - Short circuit detected!
```

Figure1- 4 LCD Segment 10 Pin Short-circuited with the UART Tx Pin of the ICE to Execute the Result and Print Information

2 Code Description

The following is the key program of the "LCD_Loading_And_Short_Circuit_Detection" example code.

The pulse mode of the LCD drive load and the charge pump timeout threshold setting:

```
#define INACTIVE_PULSE              0
#define ACTIVE_PULSE                1
#define USING_TEST_PULSE            ACTIVE_PULSE
#define USING_TEST_CPT_THRD         3     //Charge pump timeout threshold
```

Calculate the current maximum LCD drive load time:

```
uint32_t LCD_GetMaxChargePumpTimeOutValue(void)
{
    uint32_t u32MaxCTOVal=0;
    uint32_t u32Loop;
    uint32_t u32CTVal;
    for( u32Loop = 0; u32Loop < 5 ; u32Loop++)
    {
        for(u32CTVal = 1; u32CTVal < 0xFFE; u32CTVal++)
        {
            //Send the 5 frames pulse
            LCD_SET_FRAME_COUNTING_VALUE(5);
            LCD_CLEAR_FRAME_COUNTING_END_FLAG();
            LCD_CLEAR_CHARGE_TIMEOUT_FLAG();
            LCD_SET_CHARGE_TIMEOUT_TIME(u32CTVal);
            while(LCD_GET_FRAME_COUNTING_END_FLAG()==0) {};
           //Check the charge pump timeout flag
            if(LCD_GET_CHARGE_TIMEOUT_FLAG()==0)
            {
                break;
            }
            if(u32CTVal%10 == 0)
                printf(">");
        }
        if(u32CTVal > u32MaxCTOVal)
            u32MaxCTOVal = u32CTVal;
        printf("\n");
    }
    return u32MaxCTOVal;
}
```

Check if the Charge Pump Timeout has occurred:

```
uint8_t LCD_CheckChargePumpTimeOut(uint32_t u32CTOVal)
{
    uint32_t u32Loop;
    for( u32Loop=0; u32Loop < 5 ; u32Loop++)
    {
       //Send the 5 frames pulse
        LCD_SET_FRAME_COUNTING_VALUE(5);
        LCD_CLEAR_FRAME_COUNTING_END_FLAG();
        LCD_CLEAR_CHARGE_TIMEOUT_FLAG();
        LCD_SET_CHARGE_TIMEOUT_TIME(u32CTOVal);
        while(LCD_GET_FRAME_COUNTING_END_FLAG()==0) {};
       //Check the charge pump timeout flag
        if(LCD_GET_CHARGE_TIMEOUT_FLAG())
        {
            return 1;
        }
    }
    return 0;
}
```

Scan each segment pin to see if it exceeds the LCD driver loading time to determine if there is a short circuit:

```
void LCD_SegmentPinShortCircuitScan(uint8_t u8PulseLeve,uint32_t u32CTval)
{
    printf(" * Scan all segment pins and check for short circuits \n");
    LCD_AllDataSet(u8PulseLeve);
    LCD_MultiFunctionPinSet();
    LCD_Segment_MultiFunctionPinOutputMode();
    LCD_Common_MultiFunctionPinOutputMode();
    printf(" * LCD segment pin status is abnormal, and check again \n");
    LCD_MultiFunctionPinClear();
    printf(" * One by one detect LCD segment pin!\n");
    uint8_t Pin_Cnt;
    sMFP_LCD *psPtr = (sMFP_LCD *) g_sLCD_SEG_PIN;

    for(Pin_Cnt=0; Pin_Cnt<sizeof(g_sLCD_SEG_PIN)/sizeof(sMFP_LCD); Pin_Cnt++)
    {
      printf(" * Enable LCD SEG%d -", Pin_Cnt);
      //Segment Pin Set
```

```
      Set_MFP((psPtr+Pin_Cnt)->u8PinID,1, (psPtr+Pin_Cnt)->u8FunVal );
      if(LCD_CheckChargePumpTimeOut(u32CTval))
      {
        printf(" Short circuit detected!\n");
        break;
      }
      else
        printf(" OK\n");
      //Segment Pin Clear
      Set_MFP((psPtr+Pin_Cnt)->u8PinID,0, (psPtr+Pin_Cnt)->u8FunVal );
    }
    printf("\n");
    LCD_Segment_MultiFunctionPinInputMode();
    LCD_Common_MultiFunctionPinInputMode();
}
```

Scan each common pin to see if it exceeds the LCD driver loading time to determine if there is a short circuit:

```
void LCD_CommonPinShortCircuitScan(uint8_t u8PulseLeve,uint32_t u32CTval)
{
    printf(" * Scan all common pins and check for short circuits \n");
    LCD_AllDataSet(u8PulseLeve);
    LCD_MultiFunctionPinSet();
    LCD_Segment_MultiFunctionPinOutputMode();
    LCD_Common_MultiFunctionPinOutputMode();

    printf(" * LCD common pin status is abnormal, and check again \n");
    LCD_MultiFunctionPinClear();
    printf(" * One by one detect LCD common pin!\n");
    uint8_t Pin_Cnt;

    sMFP_LCD *Ptr = (sMFP_LCD *) g_sLCD_COM_PIN;
    for(Pin_Cnt=0; Pin_Cnt<sizeof(g_sLCD_COM_PIN)/sizeof(sMFP_LCD); Pin_Cnt++)
    {
      printf(" * Enable LCD COM%d -", Pin_Cnt);
    // Common Pin Set
      Set_MFP((Ptr+Pin_Cnt)->u8PinID,1, (Ptr+Pin_Cnt)->u8FunVal);

     if(LCD_CheckChargePumpTimeOut(u32CTval))
      {
        printf(" Short circuit detected!\n");
        break;
      }
      else
        printf(" OK\n");
    // Common Pin Clear
    Set_MFP((Ptr+Pin_Cnt)->u8PinID,0, (Ptr+Pin_Cnt)->u8FunVal);

    }
    printf("\n");
    LCD_Segment_MultiFunctionPinInputMode();
    LCD_Common_MultiFunctionPinInputMode();
}
```

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
  - M251/M252/M254/M256/M258 Series CMSIS BSP V3.01.002
- IDE version
  - Keil uVersion 5.26

3.2 Hardware Requirements

- Circuit components
  - NuMaker-M258KE Ver1.1
  - NuMaker – TNLCDSub_M258K_4.8V_V1.0
- Pin Connect

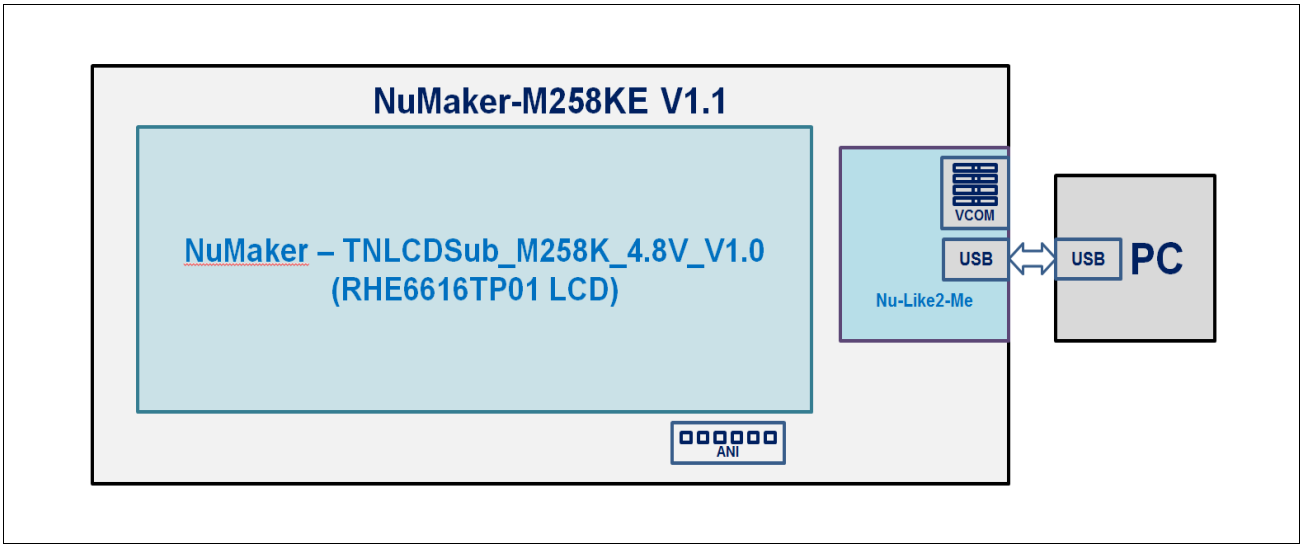

Figure 3-1 Schematic Diagram of Hardware Circuit Connection

In the first test of this example code, an external pull-down resistor is needed to increase the overall drive load and simulate the environment where the LCD pins are shorted. The user can choose the following 5 interfaces to connect to the pull-down resistor pins, as shown in Figure 3-2.

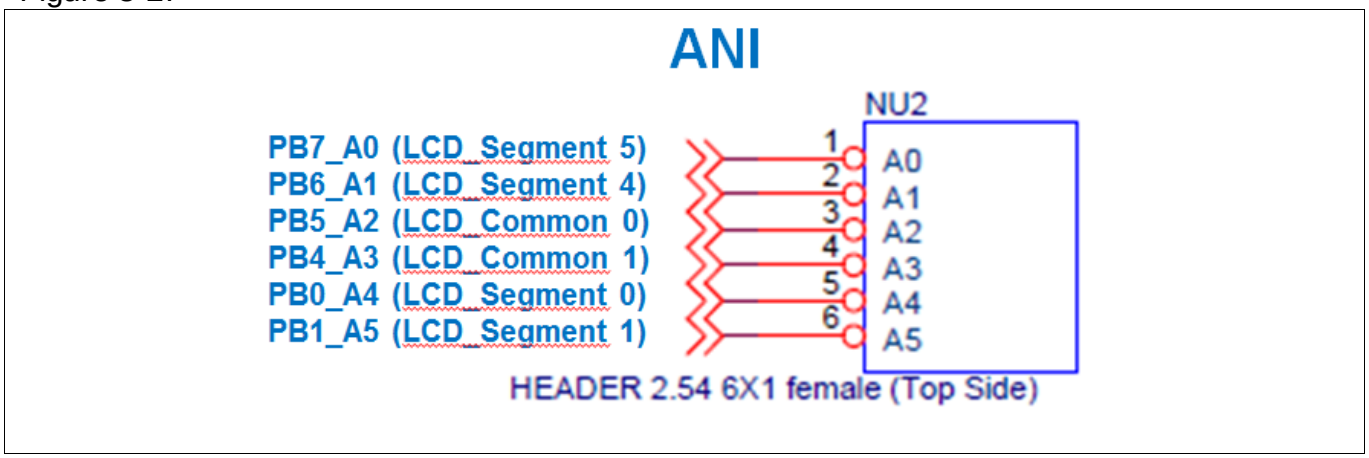

Figure 3-2 Connecting to the Pull-down Interface

In the second test, switch the connection between Switch and ICE_TX, use ICE_TX to increase the overall driving load, and simulate the LCD pin short-circuit environment, as shown in Figure 3-3.

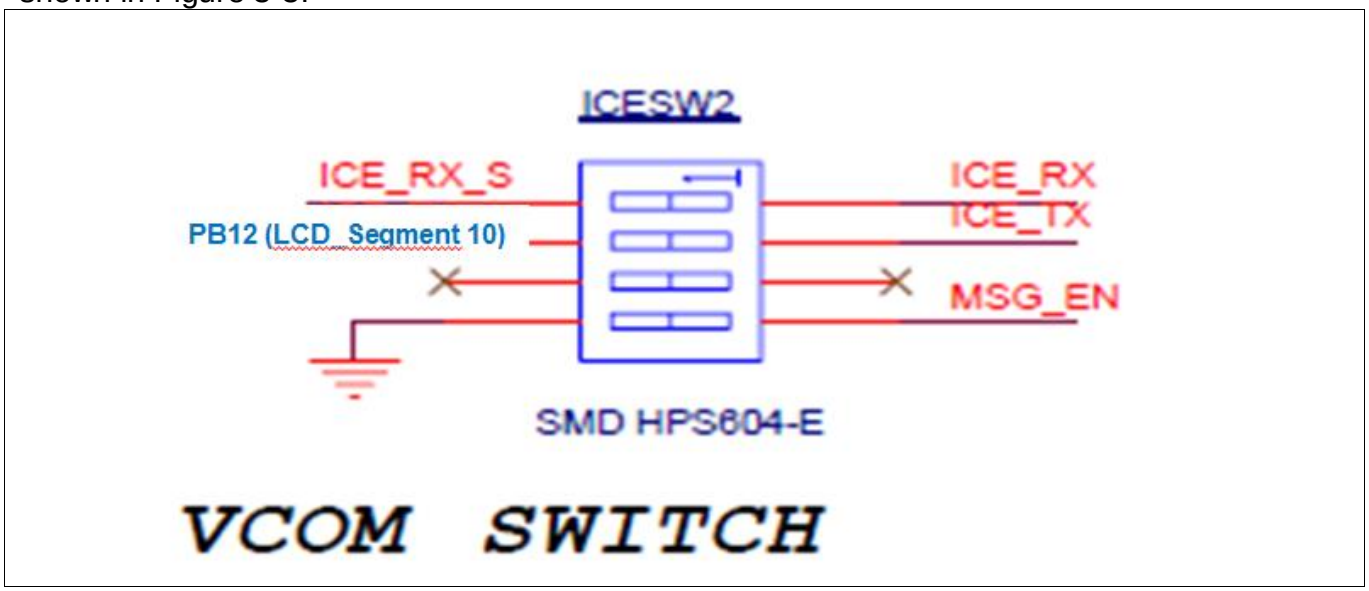

Figure 3-3 Connecting ICE_TX Circuit

4 Directory Information

The directory structure is shown below.

EC_M258_Loading_And_Short_Circuit_Detection_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

Figure 4-1 Directory Structure

5 Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click M258_Loading_And_Short_Circuit_Detection.uvprojx.
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session
3. Enter debug mode.
   - Run

6 Revision History

Date	Revision	Description
2021.10.28	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.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.