Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



THIS SPEC IS OBSOLETE

Spec No: 002-05229

Spec Title: AN205229 - F2MC-8FX Family, MB95410H/470H Series LCDC_8COM.LIB

Replaced by: None

AN205229

F²MC-8FX Family, MB95410H/470H Series LCDC_8COM.Lib

This application note describes Cypress LCDC_8COM library, designed to operate the 8 COM segment LCD driver modules which is build up in MB95F410/470 H series MCU. Added to that is the descriptions of how to use LCDC_8COM library and some notices.

OBSOLETE

Contents

1 Introduction 1
1.1 Purpose 1
1.2 Document Overview 1
2 Segment LCD 2
2.1 LCD Driver Principle 2
2.2 Configuration of LCD Controller 3
2.3 LCD Controller Configure Flow 4
3 Library 5
3.1 Library Overview 5
3.2 Application Interface 6
3.3 How to Add Cypress LCDC_8COM.lib 8
4 LIB Usage Notice 11
5 Additional Information 11
5.1 Sample Code 12
6 Document History 14

1 Introduction

1.1 Purpose

This application note describes Cypress LCDC_8COM library, designed to operate the 8 COM segment LCD driver modules which is build up in MB95F410/470H series MCU. Added to that is the descriptions of how to use LCDC_8COM library and some notices.

1.2 Document Overview

The rest of document is organized as the following:

Chapter 2 explains the working principles of 8 COM segment LCD.

Chapter 3 explains how to use LCDC_8COM library.

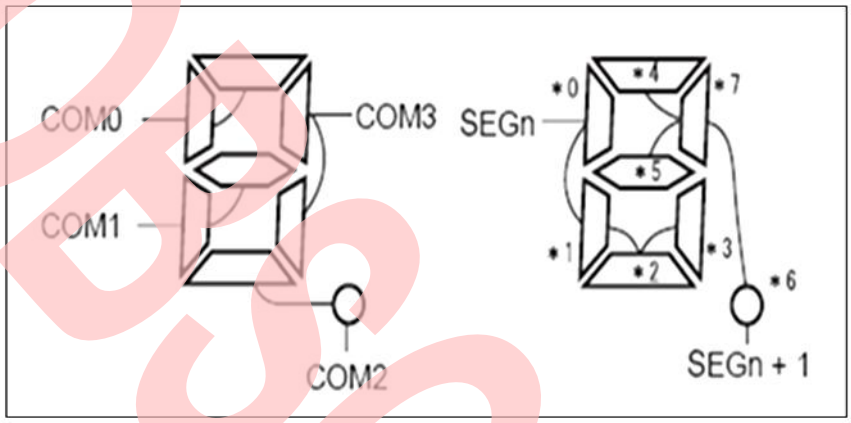
Chapter 4 explains LIB usage notice.

2 Segment LCD

Working principles of 8 COM segment LCD

2.1 LCD Driver Principle

Figure 1. COM and SEG signal layout Example in a LCD symbol

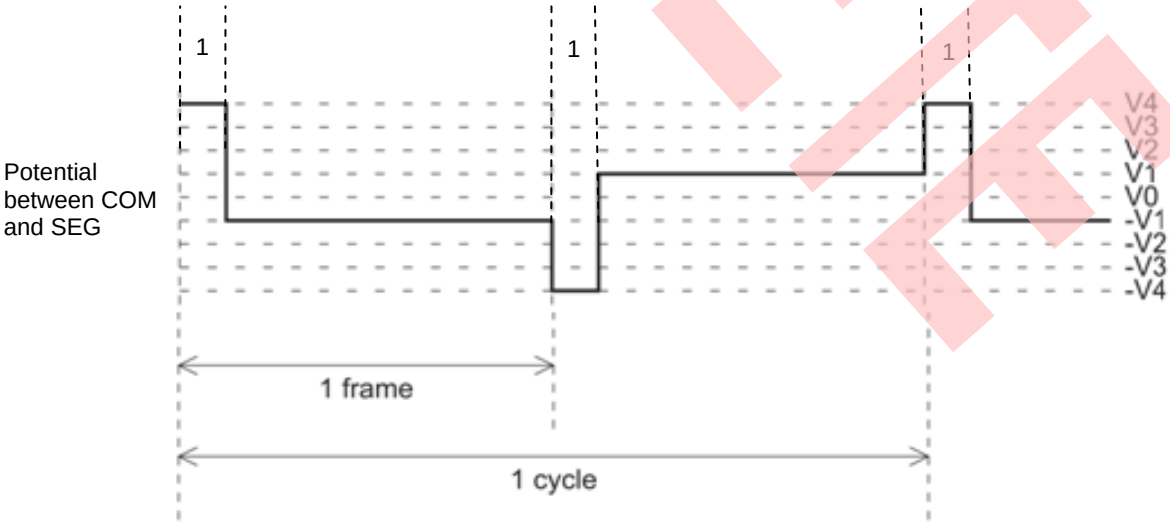


Each symbol on LCD is controlled by no less than 1 common signal and 1 segment signals. The ON or OFF status of which symbol is determined by both common and segment outputs.

2.1.1 LCD On Condition

Liquid crystal elements are turned "ON" to display that have the maximum potential difference between the common and segment outputs.

Figure 2. LCD On Condition Output Waveform Example (8 COM Mode, 1/3 Bias, 1/8 Duty)

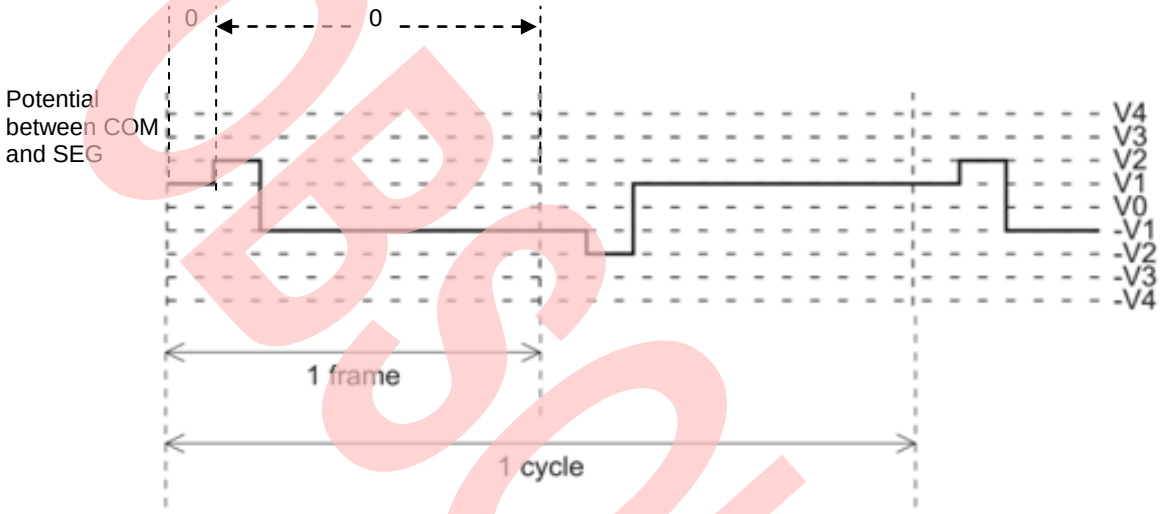


OBSOLETE

2.1.2 LCD Off Condition

Liquid crystal elements are turned "OFF" for display that haven't the maximum potential difference between the common and segment outputs.

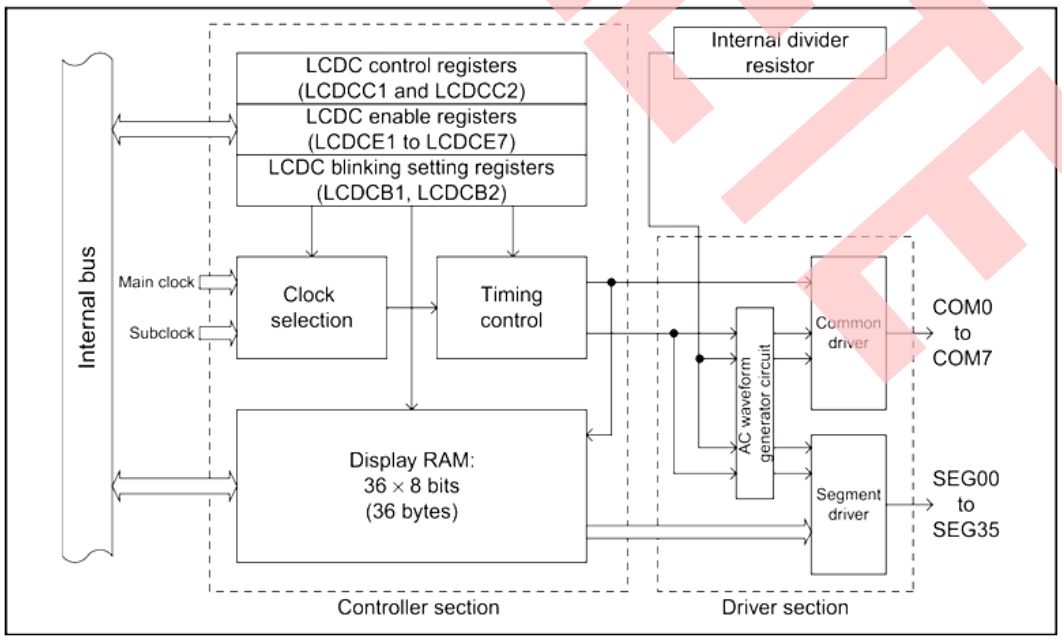
Figure 3. LCD Off Condition Output Waveform Example (8 COM Mode, 1/3 Bias, 1/8 Duty)



2.2 Configuration of LCD Controller

The LCD controller consists of the following blocks, which are divided functionally into a controller section that generates the segment and common signals based on the content of display RAM and a driver section that drives the LCD.

Figure 4. LCD Controller Block Diagram (8 COM Mode)

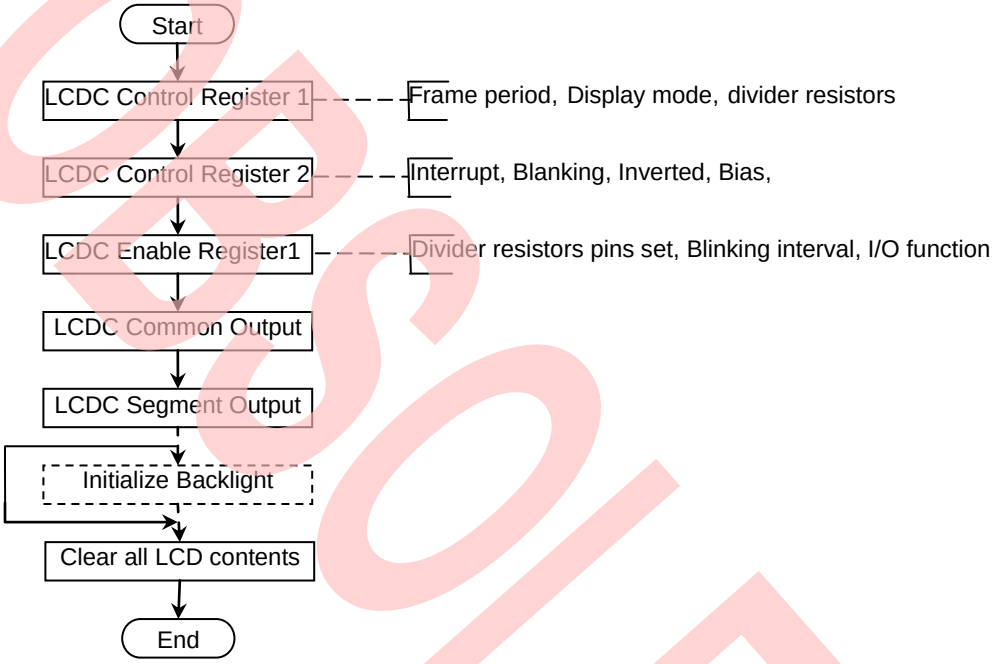


OBSOLETE

2.3 LCD Controller Configure Flow

It's necessary that configure the related registers before operate the LCD controller. There is a flow take set the LCD on MB2146-480-E EVB as an example:

Figure 5. Configure Flow chart



OBSOLETE

3 Library

This chapter introduces how to use LCDC_8COM library

3.1 Library Overview

There are parameters which user need to setup, and 4 functions as API for user's situations. All the parameters and functions are introduced as follow.

Table 1. Parameters List

Name	Description	Remarks
LCD_BUFF_LEN	Define display buffer length	N/A
LCD_LoopDispBuff	Define loop display buffer	N/A
LCD_DispBuff	Define display buffer	N/A

Table 2. Functions List

Prototype	Function Description	Remarks
void LCD_Init(void)	Initialize LCD module	N/A
void LCD_DispLoop(unsigned char * DispBuff)	Display loop data on LCD screen	N/A
void LCD_PageDisplay(void)	Display various data page on LCD screen	N/A
void LCD_StepElectrovalency(unsigned char segId, unsigned char segState)	Display Multistep Electricity Expense LED ID	N/A
void LCD_TariffRate(unsigned char segId)	Set LCD Tariff Rate Display On/Off	N/A
void LCD_DatSend(unsigned char * dispDat, unsigned char size)	Send a string of display data to LCD controller module RAM area	N/A

OBSOLETE

3.2 Application Interface

All the functions supplied by the TSC.lib will be introduced below, include the function prototype, input parameter(s), return value(s), and the function description.

3.2.1 LCD_Init

Prototype	void TSCKey_Init(void)
Parameter	void
Return	void
Description	Initialize LCD module 1. LCDC Control Register 1. 2. LCDC Control Register 2Initialize pin status to discharge TSC pad 3. LCDC Enable Register1 4. LCDC Common Output 5. LCDC Segment Output 6. Clear all LCD contents
Remark	N/A

3.2.2 LCD_DispLoop

Prototype	void LCD_DispLoop(unsigned char* DispBuff)
Parameter	unsigned char DispBuff point to loop display data buffer
Return	void
Description	Display loop data on LCD screen.
Remark	N/A

3.2.3 LCD_PageDisplay

Prototype	void LCD_PageDisplay(void)
Parameter	void
Return	void
Description	Display various data page on LCD screen
Remark	N/A

3.2.4 LCD_StepElectrovalency

Prototype	void LCD_StepElectrovalency(unsigned char colonTag)
Parameter	unsigned char segId 1~4 to display “[1]" "[2]" ”[3]” "[4]” any other value - Hide all symbol display
Return	void
Description	Display Multistep Electricity Expense LED ID
Remark	N/A

OBSOLETE

3.2.5 LCD_TariffRate

Prototype	void void LCD_TariffRate(unsigned char segId)
Parameter	unsigned char Key_Num 1 - Display 1 2 - Display 2 any other value will hide relevant LCD contents
Return	void
Description	Set LCD Tariff Rate Display On/Off
Remark	N/A

3.2.6 LCD_DatSend

Prototype	void LCD_DatSend(unsigned char * dispDat, unsigned char size)
Parameter	unsigned char dispDat point to data buffer
	unsigned char size number of data bytes
Return	void
Description	Send a string of display data to LCD controller module RAM area
Remark	N/A

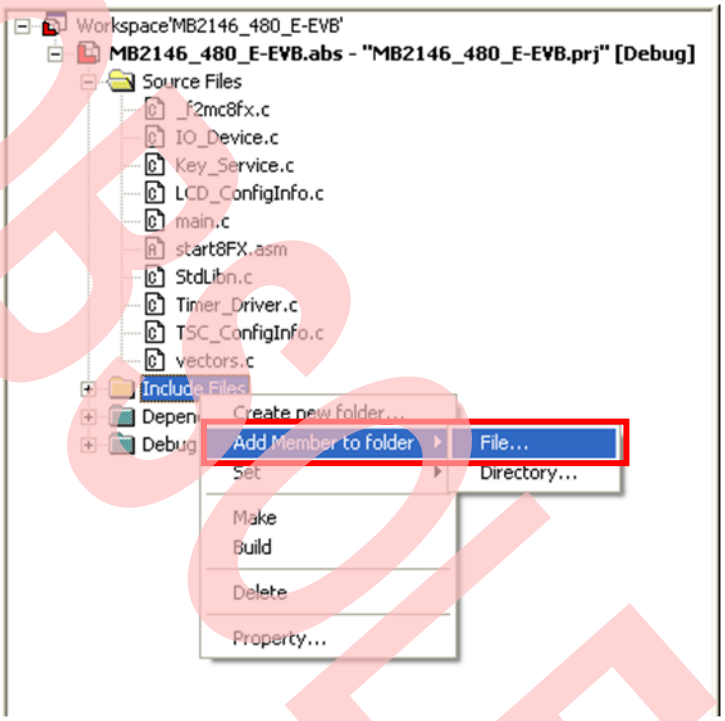
OBSOLETE

3.3 How to Add Cypress LCDC_8COM.lib

3.3.1 Add Cypress LCDC_8COM.lib to User's Project

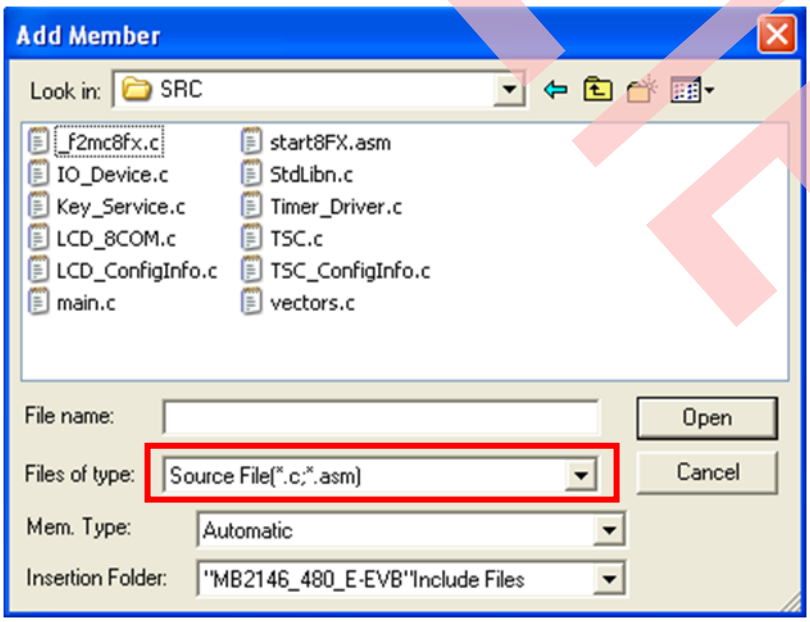
1. In Softune, Right click on folder *Include Files*→ select *Add member to folder* from the menu →select *File*.

Figure 6. Add member to folder



2. Because the default option of file type filters is *.c and * .asm, you can't found *LCDC_8COM.lib* in dialog box of *Add Member*.

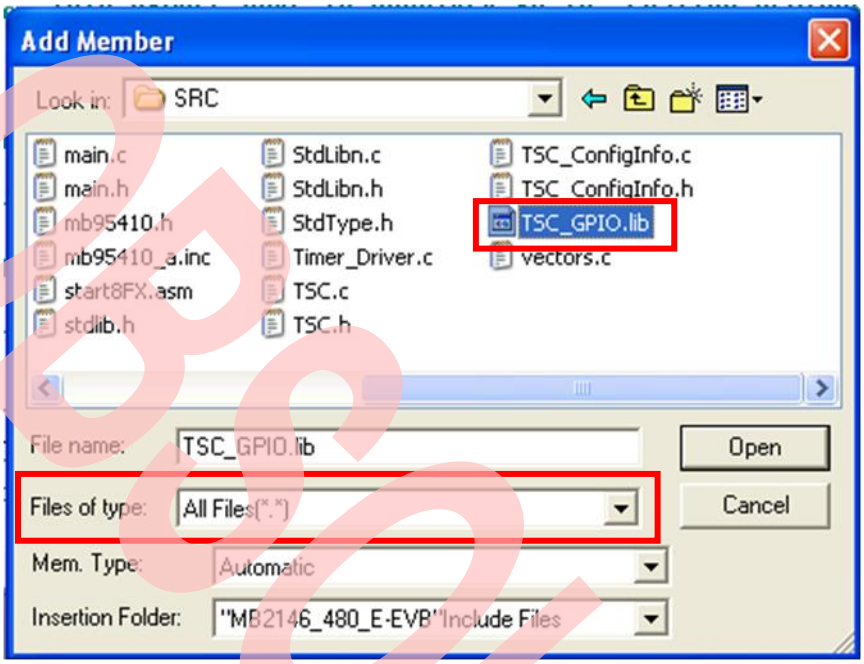
Figure 7. Popup Add Member dialog box



OBSOLETE

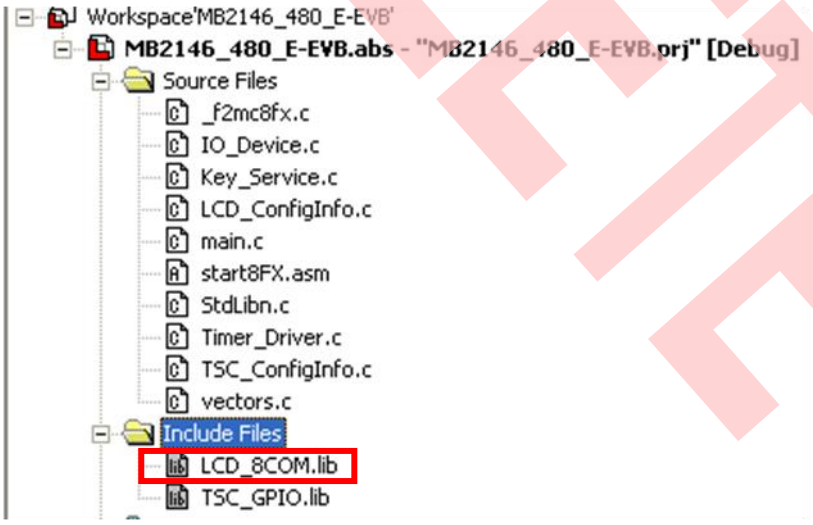
3. In *Add Member* dialog box, select 'ALL Files' from 'Files of Type', and then you will find the LCD_8COM.lib

Figure 8. Found the lib file



4. Double click LCD_8COM.lib, and then you can see it has been added in the folder *Include Files*

Figure 9. Add LCD_8COM.lib

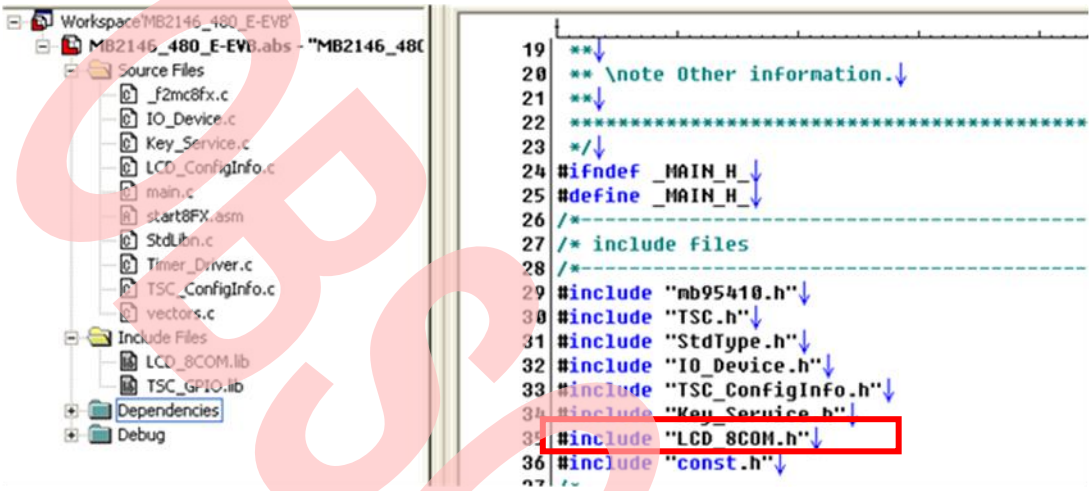


OBSOLETE

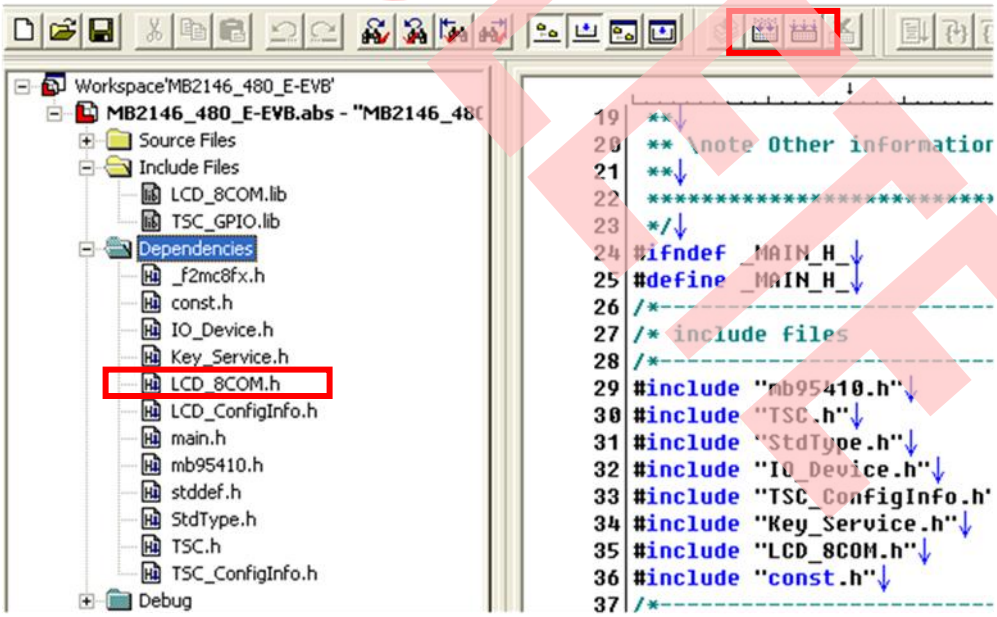
3.3.2 Include Header File

1. Add "#include "LCD_8COM.h "" in header file, such as in "main.h ".

Figure 10. Add include statement in C file



2. Compile the whole project, " LCD_8COM.h" will link LCD_8COM.lib to c file, so that user program can use API functions in LCD_8COM.lib.



OBSOLETE

4 LIB Usage Notice

This chapter introduces LIB usage notice.

- Machine clock

The machine clock should be set to 8M or above. If the machine clock is less than 8M, the sensor response is slow.

- Driver Resistors

The LCD controller has an LCD drive voltage divider resistor whose resistance value can be selected from 10kΩ to 100kΩ through software. An external divider resistor can also be used instead. The MB2146-480-EVB provide external resistor for customer evaluation. In the sample code project, we select external resistor as default, so if user need test the performance of internal resistors, please disassemble the external resistor to guarantee the LCD work right.

5 Additional Information

Please contact your local support team for any technical questions.

OBSOLETE

5.1 Sample Code

5.1.1 Main Function

Name: Main Function

Function: Initialize and configure.

main.c

```
/*!
 ******************************************************************************
 **
 ** \file main.c
 **
 ** $Id: main.c V2.0.0 2012.2.20 14:50 PM Lee.Song $
 **
 ** \brief .
 **
 ** Add here more detailed description if needed ...
 **
 ** (C) Copyright 200x-201x by Cypress Semiconductor Ltd. Asia
 **
*/
/*---------------------------------------------------------------------------*/
/* include files                                                             */
/*---------------------------------------------------------------------------*/
#include "main.h"
/*---------------------------------------------------------------------------*/
/* constants and macros                                                      */
/*---------------------------------------------------------------------------*/
volatile unsigned int LCD_DispTaskTime = 0;
RUN_FLAG run_flag;
/*---------------------------------------------------------------------------*/
/* local functions                                                           */
/*---------------------------------------------------------------------------*/
void Osc_Setup(void)
{
        #if EXClock_Used
                SYSC = 0xBF;
                SYCC = 0xF0;
                WATR = 0xF3;
                SYCC2 = 0xF4;
        #else
                SYCC = 0xF0;
                SYCC2 = 0xE5;
                CRTH_CRSEL0 = 1;
                CRTH_CRSEL1 = 0;
        #endif
        while(STBC_MRDY == 0);
}
/*---------------------------------------------------------------------------*/
/* local functions                                                           */
/*---------------------------------------------------------------------------*/
void RESET_WATCHDOG(void)
{
        WDTC = 0x35;
}
void main(void)
{
        unsigned char LoopIndex;
        CMR0_VCID = 1;          /* Set for use GPIO Disable CMP input */
        __DI();
        Osc_Setup();            /* Setup MCU main oscillator/FLL */
        InitIrqLevels();        /* Initialise Interrupt level and IRQ vector table */
        __EI();                 /* Enable system interrupt now */
```

OBSOLETE

```
        /* Module initialization */
        GPIO_Init();
        LCD_Init();              /* LCD module initialization */
        TBT_Init();              /* Time Base Timer module initialization */
        TSCKey_Init(TSCKEY_KeyNum);       /* TSC Key module initialization */
        /* Load initial value */
        LCD_DispTaskTime = Now_Time;      /* Initial LCD display task time */
        run_flag.TSC_Play = TRUE;         /* Initial flag as TRUE */
        run_flag.LCD_PlayFirstLoop = TRUE;/* Initial first play flag as TRUE */
        run_flag.RunStatusLEDFilp = TRUE; /* Initial LED flip flag as TRUE */
        /* LCD display for check */
        LCD_AllDisplay(LCD_BUFF_LEN);/* Test the LCD hardware */
        /* LED display for check */
        LED1_On;
        LED2_On;
        /* Hold on display period of time for check */
        while((Now_Time - LCD_DispTaskTime) < LCD_DispTimeInterval);
        LCD_DispTaskTime = Now_Time;
        /* LCD shut down */
        LCD_PageDisplay();
        /* LED shut down */
        LED1_Off;
        LED2_Off;
        /* Beep ringing twice */
        PERL_Beep(2,300);
        /* Load loop display value to temp buffer */
        for(LoopIndex=0;LoopIndex<8;LoopIndex++)
        {
                LCD_DispBuffTemp[LoopIndex] = LCD_LoopDispBuff[LoopIndex];
        }
        /* Endless loop */
        while(1)
        {
                /* TSC key sample unit */
                TSCKey_SampleUnit(TSCKEY_SampleNumConst);
                /* TSC key fiter function */
                TSCKey_Filter(TSCKEY_KeyNum,TSCKEY_IIRShiftConstL4);
                /* Machanical key fiter function */
                MECHKey_Filter();
                /* Machanical key service */
                MECHKey_Service();
                /* TSC key service */
                TSCKey_Service();
                /* LCD display loop judgement */
                if(Now_Time - LCD_DispTaskTime >= LCD_DispTimeInterval)
                {
                        LCD_DispTaskTime = Now_Time;
                        if(run_flag.TSC_Play == TRUE)
                        {LCD_StepElectrovalency(((LCD_LoopIndex/LOGO_BUFF_LEN)
                                                        + 1),1);
                                LCD_DispLoop(LCD_DispBuffTemp);
                        }
                        /* System run status indication LED */
                        if(run_flag.RunStatusLEDFilp == TRUE)
                        {
                                run_flag.RunStatusLEDFilp = FALSE;
                                PERL_LED(1,LED_On);
                                PERL_LED(2,LED_Off);
                        }
                        else
                        {
                                run_flag.RunStatusLEDFilp = TRUE;
                                PERL_LED(1,LED_Off);
                                PERL_LED(2,LED_On);
                        }
                }
                /* LCD display */
                LCD_PageDisplay();        /* Display various data page on LCD screen */
        }
}
```

OBSOLETE

6 Document History

Document Title: AN205229 - F²MC-8FX Family, MB95410H/470H Series LCDC_8COM.LIB

Document Number: 002-05229

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	CHMA	02/23/2012	Initial release
*A	5267430	CHMA	06/17/2016	Migrated Spansion Application Note MCU-AN-500134-E-10 to Cypress format This app note is primarily about a code library, LCDC_8COM.Lib, that .lib file must be attached to this app note. But there is no .lib file on spansion.com, So this AN is for obsolete.

OBSOLETE

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

cypress.com/psoc

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums |Blogs | Video |Training

Technical Support

cypress.com/support

OBSOLETE

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2012-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.