Freescale Semiconductor
Application Note

Document Number: AN4159
Rev. 0, 8/2010

LCD Driver for the MC9S08LGW64

by: Tanya Malik
Reference Design and Applications Group
Noida
India

1 Introduction

This document describes a driver for an Liquid Crystal Display (LCD). This allows customization of all the possible configurations for this peripheral.

The software architecture is designed to provide seamless migration between devices that contain the same peripheral module.

In this application note, the driver interfaces are explained. Various applications for the MC9S08GW64 can make use of this driver. The following sections describe the details and steps for creating an application using the LCD driver.

In the MC9S08GW64 series, the LCD module (LCD) controls up to 44 LCD pins to generate the waveforms necessary to drive a liquid crystal display. It can support a 3 V and 5 V LCD glass. The bus clock to the LCD can be gated on and off using the LCD bit in the SCGC2. This bit is cleared after any reset, which disables the bus clock to this module.

2 Software Driver Description

The LCD driver is provided as some C code files. You can add these files to your applications. With the integration of the LCD driver, you can call the LCD driver APIs to use the LCD functionality in your application.

There are five files associated with the LCD driver. The following is a brief description of them:

Contents

1 Introduction..1
2 Software Driver Description.................................1
2.1 lcd_functions.h...2
2.2 lcd_functions.c...2
2.2.1 LCDInit...2
2.2.2 LCDBlinkState.....................................2
2.2.3 lcd_StopBlinking..................................3
2.2.4 LCDOnOffState....................................3
2.2.5 LCDClear...3
2.2.6 LCDScrollNumbersAndAlphabetbets....4
2.2.7 LCDKey1Func.....................................4
2.2.8 LCDDiagnosticFunc.............................4
2.2.9 lcd_PrintString.....................................5
2.2.10 lcd_SlideString.....................................5
2.2.11 DispHexVal..5
2.2.12 DispDecVal...6
3 Assumptions...6
4 Use Case...6
5 Conclusion 6

- **lcd_functions.c**—It is the main file for the driver. It contains the various high level API definitions exposed to the applications for the LCD functionality.
- **lcd_functions.h**—This file contains the high level API declarations. This file is included in the application that intends to use the LCD driver.
- **lcd.c**—The LCD glass specific segment to the PIN mapping matrix is defined in this file.
- **lcd.h**—The LCD glass specific macros are defined in this file. This file contains the LCD configuration specific flags, that help the driver decide what LCD configuration is to be used. These flags are edited as per the application LCD requirements.

NOTE

The LCD driver supports a 21 x 8 mode. The driver can be used to support other modes by making the changes in the lcd.c and lcd.h (these files are specific to the glass used).

2.1 lcd_functions.h

This file contains the definitions of the high level functions that the user needs to call.

2.2 lcd_functions.c

This file contains the various high level functions that can be directly used.

2.2.1 LCDInit

Description:

This function is used to initialize the LCD. The LCD clock depends on the settings executed by the user in the lcd.h file. The user can select the clock source, enable or disable the charge pump, and select the source of the VLL1, VLL2, VLL3.

Prototype

void LCDInit(void);

Input parameters:

None

Output parameters:

None

Example:

LCDInit();

2.2.2 LCDBlinkState

Description:

It enables the LCD segment blink state mode. The function selects LCDRAM registers that control LCD segment blink states.

Prototype:

void LCDBlinkState(void);

Input parameters:

None

Output parameters:

None

Example:
LCDBlinkState();

2.2.3 lcd_StopBlinking

Description:
This function disables LCD segment blink state modes.

Prototype:
void lcd_StopBlinking(void);

Input parameters:
None

Output parameters:
None

Example:
lcd_StopBlinking();

2.2.4 LCDOnOffState

Description :
This function is used to enable LCD segments in the on and off state mode.

Prototype:
Void LCDOnOffState(void);

Input parameters:
None

Output parameters:
None

Example:
LCDOnOffState();

2.2.5 LCDClear

Description:
This function is used to clear LCDRAM registers.

Prototype:
void LCDClear(void);

Input parameters:
None

Output parameters:
None

Example:
LCDClear();

2.2.6 LCDScrollNumbersAndAlphabetbets

Description:

This function is used to scroll the digits and alphabets on the LCD screen at the positioned passed as an argument by the user and uses the functions LCDClearDigit(), and LCDPutChar().

Prototype:

void LCDScrollNumbersAndAlphabetbets(unsigned char digit);

Input parameters:

digit—The digit position on the LCD screen where the user wants to scroll the numbers and alphabets.

Output parameters:

None

Example:

LCDScrollNumbersAndAlphabetbets(2);

2.2.7 LCDKey1Func

Description:

Scrolls the digits and alphabets on the LCD screen at all positions, one by one. It uses the function LCDScrollNumbersAndAlphabetbets() that is used to scroll the digits and alphabets on the LCD screen at the the positioned passed as an argument by the user..

Prototype:

void LCDKey1Func(void);

Input parameters:

none

Output parameters:

none

Example:

LCDKey1Func();

2.2.8 LCDDiagnosticFunc

Description:

This function is used to test the LCD functionality by turning on the special LCD segments, then scrolling all the digits, and alphabets, one by one.

Prototype:

void LCDDiagnosticFunc(void);

Input parameters:

None

Output parameters:

None

Example:

LCDDiagnosticFunc();

2.2.9 lcd_PrintString

Description:

This function is used to print a string on the LCD starting from the first position on the LCD glass.

Prototype:

void lcd_PrintString(unsigned char *str);

Input parameters:

str—Enter the string you want displayed on the LCD.

Output parameters:

None

Example:

lcd_PrintString(“Hello”);

2.2.10 lcd_SlideString

Description:

This function slides strings on the LCD starting from the last position on the LCD glass.

Prototype:

void lcd_SlideString(unsigned char *str);

Input parameters:

str—Enter the string you want displayed on the LCD.

Output parameters:

None

Example:

lcd_SlideString('hello");

2.2.11 DispHexVal

Description:

This function is used to display the hex value of the character entered.

Prototype:

void DispHexVal(unsigned char val , unsigned char startloc);

Input parameters:

val—Character for the hex value desired.
startloc—Starting location where the hex value is displayed.

Output parameters:

None

Example:

DispHexVal(10, 2);

2.2.12 DispDecVal

Description:

This function displays the decimal value of the character entered.

Prototype:

void DispDecVal(unsigned char val, unsigned char startloc)

Input parameters:

val—Value for the desired decimal value.
startloc—Starting location where the decimal value needs to be displayed.

Output parameters:

None

Example:

DispDecVal(23, 2);

3 Assumptions

The descriptions in this document assume the person reading it has full knowledge of all the configuration registers of all the blocks in the MC9S08GW64, especially for LCD and Internal Clock Source (ICS) blocks.

4 Use Case

Step 1—Include the file lcd.h and lcd_functions.h in the main file. Make the configuration changes in lcd.h file according to the requirements and depending upon the LCD glass.

Step 2—Call the function:

LCD LCDInit();

It initializes the LCD with the configurations done in the lcd.h file

Step 3—To to turn on all the LCD segments call the function as shown:

LCD_ALLFP_ON

Step 4—To switch off all the segments call the function:

LCD_ALLFP_OFF

Step 5—To print a string on the LCD call the function:

lcd_PrintString("Hello");

5 Conclusion

This driver provides a software base for applications that need the implementation of LCDs.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and sofware implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor prodcuts are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

Document Number: AN4159
Rev. 0, 8/2010