**Freescale Semiconductor**
Application Note

Document Number: AN4179
Rev. 0, 7/2010

# How to Interface and Drive a 3V or 5V LCD Glass with MC9S08GW64

by: Saurabh Jhamb, Tanya Malik
Reference Design and Applications Engineering
Microcontroller Solutions Group

**Contents**

## 1 Introduction

This document describes a driver for the LCD. This LCD driver allows the user to customize glass requirements with the MCU’s LCD module. This driver is tested for the microcontroller demo board MC9S08GW64 (TWR-S08GW64).

The software architecture is designed to provide seamless migration between these devices. This document is intended to be used by all software development engineers and test engineers who has to use the MCUs with the LCD.

Figure 1 shows the block diagram for the LCD. The MC9S08GW64 contains 44 pins for LCD frontplane and backplane operation that are totally configurable. Therefore, any pin can be configured for use with either the frontplane or the backplane.

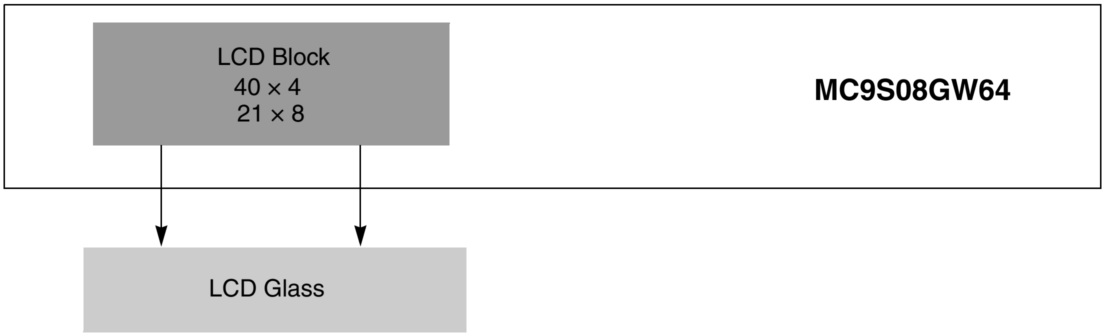


**Figure 1. LCD block diagram**

# 2 Hardware interface description

Table 1 describes the LCD pins and their directions.

**Table 1. LCD pin description**

| Pins | Direction | Description |
|---|---|---|
| LCD[43:0] | Output | LCD data pins<br>User can configure these pins as frontplane or backplane. |
| $V_{LL1}$ | Output | LCD bias voltage |
| $V_{LL2}$ | Output | LCD bias voltage |
| $V_{LL3}$ | Input | LCD bias voltage<br>See Section 5, “Options for $V_{LL3}$,” for more details. |
| $V_{LL3\_2}$ | Input | Provides current enhancement at $V_{LL3}$<br>**Note:** This pin must be shorted with $V_{LL3}$. |
| $V_{CAP1}$ | N/A | Provides storage capacitance for LCD operation on internal charge pump mode. |
| $V_{CAP2}$ | N/A | |

## 2.1 Frontplane and backplane pin connections

These pin connections identify the number of backplanes and frontplanes supported in the LCD glass and connect each LCD glass pin to the respective LCD pin at the MC9S08GW64 MCU.

**NOTE**

In the LCD software driver configuration, the number of LCD pins in a MCU that connect to the backplane of the LCD glass must be configured as backplane.

## 2.2 Voltage pin connections

Add a capacitor on the $V_{LL3}$ pin to filter noise from the $V_{DD}$ input and on $V_{LL1}$, $V_{LL2}$, $V_{CAP1}$, and $V_{CAP2}$ pins for the LCD operation in charge pump mode. No capacitor is required for the LCD operation in resistor bias mode on $V_{LL1}$, $V_{LL2}$, $V_{CAP1}$, and $V_{CAP2}$ pins

## 2.3 Frontplane and backplane configuration

### 2.3.1 LCD pin enable registers (LCDPENn)

These registers specify the number of pins used for the LCD display.

| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| LCDPEN0 | R<br>W | PEN7 | PEN6 | PEN5 | PEN4 | PEN3 | PEN2 | PEN1 | PEN0 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDPEN1 | R<br>W | PEN15 | PEN14 | PEN13 | PEN12 | PEN11 | PEN10 | PEN9 | PEN8 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDPEN2 | R<br>W | PEN23 | PEN22 | PEN21 | PEN20 | PEN19 | PEN18 | PEN17 | PEN16 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDPEN3 | R<br>W | PEN31 | PEN30 | PEN29 | PEN28 | PEN27 | PEN26 | PEN25 | PEN24 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDPEN4 | R<br>W | PEN39 | PEN38 | PEN37 | PEN36 | PEN35 | PEN34 | PEN33 | PEN32 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDPEN5 | R<br>W | ▨ | ▨ | ▨ | ▨ | PEN43 | PEN42 | PEN41 | PEN40 |
| | Reset | Indeterminate after reset | | | | | | | |

▨ Unimplemented or Reserved

**Figure 2. LCDPENn registers**

**Table 2. LCDPEN[5:0] field descriptions**

| Field | Description |
|---|---|
| PEN[43:0] | **LCD Pin Enable**—The PEN[43:0] bit enables the LCD[43:0] pin for the LCD operation. Each LCD[43:0] pin can be configured as a backplane or a frontplane based on the corresponding BPEN[n] bit in the Backplane Enable Register (LCDBPEN[5:0]). If LCDEN = 0, these bits have no effect on the state of the I/O pins. Set PEN[43:0] bits before LCDEN is set.<br>0 LCD operation disabled<br>1 LCD operation enabled |

## 2.3.2 Backplane enable registers

These registers specify the number of pins used as backplane. If the BPEN bits are cleared for a corresponding pin, then it acts as a frontplane pin.

| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| LCDBPEN0 | R<br>W | BPEN7 | BPEN6 | BPEN5 | BPEN4 | BPEN3 | BPEN2 | BPEN1 | BPEN0 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDBPEN1 | R<br>W | BPEN15 | BPEN14 | BPEN13 | BPEN12 | BPEN11 | BPEN10 | BPEN9 | BPEN8 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDBPEN2 | R<br>W | BPEN23 | BPEN22 | BPEN21 | BPEN20 | BPEN19 | BPEN18 | BPEN17 | BPEN16 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDBPEN3 | R<br>W | BPEN31 | BPEN30 | BPEN29 | BPEN28 | BPEN27 | BPEN26 | BPEN25 | BPEN24 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDBPEN4 | R<br>W | BPEN39 | BPEN38 | BPEN37 | BPEN36 | BPEN35 | BPEN34 | BPEN33 | BPEN32 |
| | Reset | Indeterminate after reset | | | | | | | |
| LCDBPEN5 | R<br>W | ▨ | ▨ | ▨ | ▨ | BPEN43 | BPEN42 | BPEN41 | BPEN40 |
| | Reset | Indeterminate after reset | | | | | | | |

▨ Unimplemented or Reserved

**Figure 3. LCDBPEN[5:0] registers**

**Table 3. LCDBPEN[5:0] field descriptions**

| Field | Description |
|---|---|
| LCDBPEN[43:0] | **Backplane Enable**—The BPEN[43:0] bit configures the LCD[43:0] pin to operate as an LCD backplane or LCD frontplane. If LCDEN = 0, these bits have no effect on the state of the I/O pins. It is recommended to set BPEN[43:0] bits before LCDEN is set.<br>0 Frontplane operation enabled<br>1 Backplane operation enabled |

## 2.3.3 LCD waveform register

| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| **LCDWF0** | R<br>W | BPHLCD | BPGLCD | BPFLCD | BPELCD | BPDLCD | BPCLCD | BPBLCD | BPALCD |
| | Reset | Indeterminate after reset | | | | | | | |

**Figure 4. LCDWF0 register**

**NOTE**

There are 64 LCD waveform registers LCDWF0 - LCDWF63. For more information, refer to the *MC9S08GW64 Reference Manual* (document MC9S08GW64RM).

**Table 4. LCDWF field descriptions**

| Field | Description |
|---|---|
| BP[x]LCD[y] | **Segment on frontplane operation** — If the LCD[x] pin is enabled and configured to operate as a frontplane, the BP[y]LCD[x] bits in the LCDWF registers controls the on/off state for the LCD segment connected between LCD[x] and BP[y]. BP[y] corresponds to LCD[43:0] pin enabled and configured to operate as a backplane that is active in phase[y]. Asserting BP[y]LCD[x] displays (turns on) the LCD segment connected between LCD[x] and BPp[y].<br>0 LCD segment off<br>1 LCD segment on<br>**Segment on backplane operation** — If the LCD[x] pin is enabled and configured to operate as a backplane, the BP[y] LCD[x] bit in the LCDWF registers controls the phase (A-H) in which the LCD[x] pin is active. Backplane phase assignment is done using the method.<br>0 LCD BP[x] inactive for LCD[y]<br>1 LCD BP[x] active for LCD[y] |

# 3 LCD Software

## 3.1 LCD software architecture

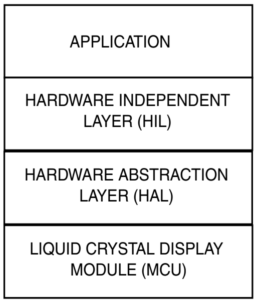

APPLICATION
HARDWARE INDEPENDENT LAYER (HIL)
HARDWARE ABSTRACTION LAYER (HAL)
LIQUID CRYSTAL DISPLAY MODULE (MCU)

**Figure 5. LCD software architecture**

### 3.1.1 Hardware abstraction layer

The hardware abstraction layer is defined as a collection of software components that directly access hardware resources. In this layer, the LCD driver defines macros and functions that configure the custom glass requirements in the LCD module registers.

### 3.1.2 Hardware independent layer

The hardware independent layer is defined as a collection of software components that access hardware resources through HAL. Peripheral drivers are implemented in this layer.

## 3.2 Hardware Abstraction Layer(HAL)

The HAL module defines the macros and functions needed to match the custom glass hardware specifications with the MCU LCD registers. This information is obtained from each custom glass specification. The changes to these macros are performed through modifications on the macro definitions located in the *lcd.h* header file. The user functions are located in *lcd.c*

The user is responsible for providing correct definitions for these hardware access macros. Descriptions of the individual macros and examples of their definitions are given below.

### 3.2.1 LCD_NO_OF_BACK_PLANES

The `LCD_NO_OF_BACK_PLANES` macro defines the number of backplane electrodes on the custom glass. The duty ratio of the waveforms generated by the LCD module is 1/ (_LCDBACKPLANES). This module supports values from 1–8.

### 3.2.2 LCD_CLOCK

The `LCD_CLOCK` macro defines the LCD module clock source. The available clocks are the internal (also called alternate) clock or the external clock of 32.768 kHz. The supported clock range is from 30 kHz to 39.053 kHz.

- `LCD_CLOCK_EXT`—Selects external clock source
- `LCD_CLOCK_ALT`—Selects alternate clock source

**Example:** `#define LCD_CLOCK LCD_CLOCK_EXT`

### 3.2.3 LCD_VSUPPLY

The `LCD_VSUPPLY` macro defines whether the LCD module power supply is internal or external. These device sources change in each device. Supply sources for the MC9S08GW64 are:

- `LCD_VLL2_FROM_VDD`—Drive $V_{LL2}$ internally from $V_{DD}$
- `LCD_VLL3_FROM_VDD`—Drive $V_{LL3}$ internally from $V_{DD}$
- `LCD_VLL1_FROM_VLCD`—Drive $V_{LL1}$ internally from $V_{LCD}$
- `LCD_VLL3_FROM_EXT`—Drive $V_{LL3}$ externally Or $V_{IREG}$

**Example:** `#define LCD_VSUPPLY LCD_VLL3_FROM_VDD`

### 3.2.4 LCD_CLOCK_PRESCALER

This macro defines the LCD clock pre-scaler, which in combination with the number of backplanes of the LCD determines the LCD frame frequency.

The LCD frame frequency is the number of times the LCD is energized per second. The LCD module frame frequency must be selected to prevent the LCD display from flickering (LCD module frame frequency is too low) or ghosting (LCD module frame frequency is too high).To avoid these conditions, an LCD module frame frequency in the range of 28 Hz to 58 Hz is required.

Table 5 and Table 6 shows the LCD frame frequency calculations and have LCD clock inputs as ~32.786 kHz and ~ 39.063 kHz, respectively.

**Table 5. LCD frame frequency calculation at ~32.786kHz**

| Duty Cycle | 1/1 | 1/2 | 1/3 | 1/4 | 1/5 | 1/6 | 1/7 | 1/8 |
|---|---|---|---|---|---|---|---|---|
| Y | 16 | 8 | 5 | 4 | 3 | 3 | 2 | 2 |
| LCLK[2:0] | | | | | | | | |
| 0 | 64 | 64 | 68.3 | 64 | 68.3 | 56.9 | 73.1 | 64 |
| 1 | 51.2 | 51.2 | 54.6 | 51.2 | 54.6 | 45.5 | 58.5 | 51.2 |
| 2 | 42.7 | 42.7 | 45.5 | 42.7 | 45.5 | 37.9 | 48.8 | 42.7 |
| 3 | 36.6 | 36.6 | 39 | 36.6 | 39 | 32.5 | 41.8 | 36.6 |
| 4 | 32 | 32 | 34.1 | 32 | 34.1 | 28.4 | 36.6 | 32 |
| 5 | 28.4 | 28.4 | 30.3 | 28.4 | 30.3 | 25.3 | 32.5 | 28.4 |
| 6 | 25.6 | 25.6 | 27.3 | 25.6 | 27.3 | 22.8 | 29.3 | 25.6 |
| 7 | 23.3 | 23.3 | 24.8 | 23.3 | 24.8 | 20.7 | 26.6 | 23.3 |

**Table 6. LCD frame frequency calculation at ~39.063**

| Duty Cycle | 1/1 | 1/2 | 1/3 | 1/4 | 1/5 | 1/6 | 1/7 | 1/8 |
|---|---|---|---|---|---|---|---|---|
| Y | 16 | 8 | 5 | 4 | 3 | 3 | 2 | 2 |
| LCLK[2:0] | | | | | | | | |
| 0 | 76.3 | 76.3 | 81.4 | 76.3 | 81.4 | 67.8 | 87.2 | 76.3 |
| 1 | 61 | 61 | 65.1 | 61 | 65.1 | 54.3 | 69.8 | 61 |
| 2 | 50.9 | 50.9 | 54.3 | 50.9 | 54.3 | 45.2 | 58.1 | 50.9 |
| 3 | 43.6 | 43.6 | 46.5 | 43.6 | 46.5 | 38.8 | 49.8 | 43.6 |
| 4 | 38.1 | 38.1 | 40.7 | 38.1 | 40.7 | 33.9 | 43.6 | 38.1 |
| 5 | 33.9 | 33.9 | 36.2 | 33.9 | 36.2 | 30.1 | 38.8 | 33.9 |
| 6 | 30.5 | 30.5 | 32.6 | 30.5 | 32.6 | 27.1 | 34.9 | 30.5 |
| 7 | 27.7 | 27.7 | 29.6 | 27.7 | 29.6 | 24.7 | 31.7 | 27.7 |

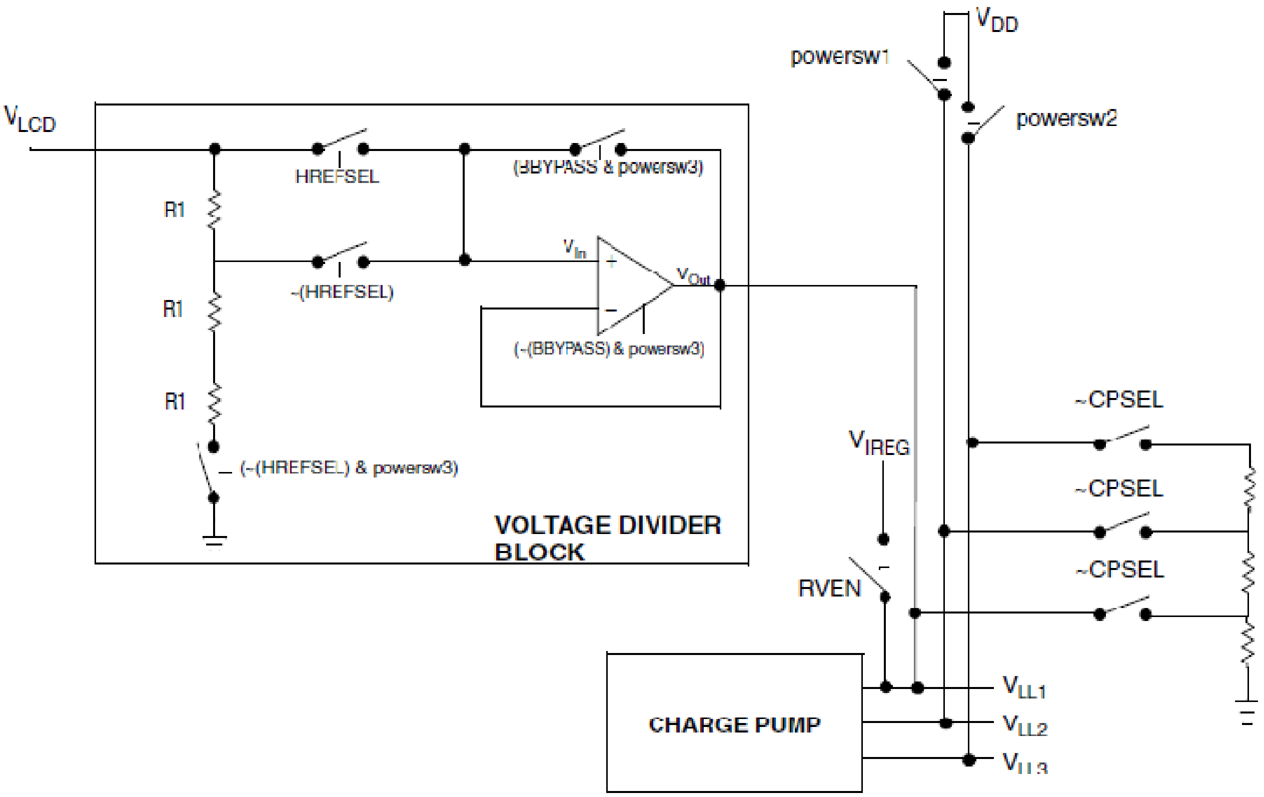
**Example:** `#define LCD_CLOCK_PRESCALER  0`

### 3.2.5 LCD_CHARGE_PUMP_SELECT

The LCD charge pump selector macro defines the type of supply for the LCD voltages $V_{LL1}$, $V_{LL2}$, and $V_{LL3}$. The LCD module provides two options for the supply voltage: resistor network or charge pump (Figure 6).

- `LCD_CHARGE_PUMP_DISABLE`—Selects the resistor network
- `LCD_CHARGE_PUMP_ENABLE`—Selects the charge pump

**Example:** `#define LCD_CHARGE_PUMP_SELECT LCD_CHARGE_PUMP_ENABLE`



| VSUPPLY[1:0] | Configuration | powersw1 | powersw2 | powersw3 |
|---|---|---|---|---|
| 00 | Drive $V_{LL2}$ internally from $V_{DD}$ | 1 | 0 | 0 |
| 01 | Drive $V_{LL3}$ internally from $V_{DD}$ | 0 | 1 | 0 |
| 10 | Drive $V_{LL1}$internally from the $V_{LCD}$ pin | 0 | 0 | 1 |
| 11 | Drive $V_{LL3}$ externally from $V_{DD}$<br>Or<br>Drive $V_{LL1}$ internally from $V_{IREG}$ | 0 | 0 | 0 |

**Figure 6. LCD Charge Pump and VLCD Voltage Divider Block Diagram**

### 3.2.6 LCD_LOAD_ADJUST

The load adjust macro configures the LCD module to manage a different LCD glass capacitance. The capacitance of the LCD depends on the custom glass. The value written in this macro is related to the type of voltage source selected (resistor network or charge pump). The results for the different possible combinations of `LCD_LOAD_ADJUST` and `LCD_CHARGE_PUMP_SELECT` (0—resistor network, 1—charge pump) for MC9S08GW64 are shown in Table 7.

**Table 7. LCD load adjust values**

| LCD_LOAD_ADJUST | LCD_CHARGE_PUMP_SELECT=1 | LCD_CHARGE_PUMP_SELECT=0 |
|---|---|---|
| 0 | 8000pF | 2000pF |
| 1 | 6000pF | 2000pF |
| 2 | 4000pF | 8000pF |
| 3 | 2000pF | 8000pF |

For more information, refer to the *MC9S08GW64 Reference Manual* (document MC9S08GW64RM).

### 3.2.7 CONFIG_LCD_MODE

This macro configures the 29 pin LCD with 21 frontplanes and 8 backplanes. The user needs to change the configuration depending upon the LCD used and the connection of the LCD with the MCU.

```
#define CONFIG_LCD_MODE
{
     LCDPEN0 = 0xFC; /*Enables the respective pins of the MCU connected with the LCD glass */
     LCDPEN1  = 0x03;
     LCDPEN2  = 0xFC;
     LCDPEN3  = 0xFF;
     LCDPEN4  = 0x7F;
     LCDPEN5  = 0x00;
     LCDBPEN0  = 0xFC;  /*It enables the respective LCD pin as backplane*/
     LCDBPEN1  = 0x03;
     LCDBPEN2  = 0x00;
     LCDBPEN3  = 0x00;
     LCDBPEN4  = 0x00;
     LCDBPEN5  = 0x00;
     LCDPIN1  = 0x01;   /*It is used to enable the required segments*/
     LCDPIN2  = 0x02;
     LCDPIN3  = 0x04;
     LCDPIN4  = 0x08;
     LCDPIN5  = 0x10;
```

```
    LCDPIN6 = 0x20;
    LCDPIN7 = 0x40;
    LCDPIN8 = 0x80;
}
```

### 3.2.8 LCD_ALLFP_OFF

This macro is used to turn off all the segments of the custom glass.

### 3.2.9 LCD_ALLFP_ON

This macro is used to turn on all the segments of the custom glass.

## 3.3 LCD HAL functions

### 3.3.1 LCDFindChar()

**Description:**

Finds the position of the given character in the character table defined in *lcd.h*

**Prototype:**

```
unsigned char LCDFindChar(unsigned char character)
```

**Input parameters:**

`character`—The character whose position is to be found.

**Return value:**

None

**Typical usage:**

```
LCDFindChar(‘A’);
```

### 3.3.2 LCDPutChar()

**Description:**

It puts the character in the LCD RAM register and the character is displayed on the LCD glass at the position specified.

**Prototype:**

```
void LCDPutChar(unsigned char character, unsigned char digit)
```

**Input parameters:**

`character`—The character to be displayed.

`digit`—The position on the LCD glass.

**Return value:**

None

**Typical usage:**

```
LCDPutChar ('A',2);
```

### 3.3.3 LCDWriteSegment()

**Description:**

It displays the respective symbol on the LCD glass.

**Prototype:**

```
void LCDWriteSegment(enum symbol s, unsigned char num)
```

**Input parameters:**

`s`—The symbol to be displayed.

`num`—The position on the LCD glass.

**Return value:**

None

**Typical usage:**

```
LCDWriteSegment(TIME, 2);
```

### 3.3.4 LCDClearSegment()

**Description:**

It clears the respective symbol on the LCD glass.

**Prototype:**

```
void LCDClearSegment(enum symbol s, unsigned char num)
```

**Input parameters:**

`s`—The symbol to be cleared.

`num`—The position on the LCD glass.

**Return value:**

None

**Typical usage:**

```
LCDClearSegment(TIME,2);
```

### 3.3.5 LCDClearSegs()

**Description:**

It clears all the segments.

**Prototype:**

```
void LCDClearSegs(void)
```

**Input parameters:**

None

**Return value:**

None

**Typical usage:**

```
LCDClearSegs();
```

### 3.3.6 LCDClearDigit()

**Description:**

It clears all the digits.

**Prototype:**

```
void LCDClearDigit()
```

**Input parameters:**

None

**Return value:**

None

**Typical usage:**

```
LCDClearDigit();
```

### 3.3.7 LCDTest()

**Description:**

It enables all the segments on the LCD used to test the LCD.

**Prototype:**

```
void LCDTest (void)
```

**Input parameters:**

None

**Return value:**

None

**Typical usage:**

```
LCDTest();
```

# 4 Interfacing LCD glass with MC9S08GW64

This section describes an example of interfacing the LCD glass (21 x 8) with the MC9S08GW64. The LCD module of the MCU is configured in charge pump enable mode, and drives $V_{LL3}$ pin from $V_{DD}$ supply. Figure 7 shows the hardware connections for this example.

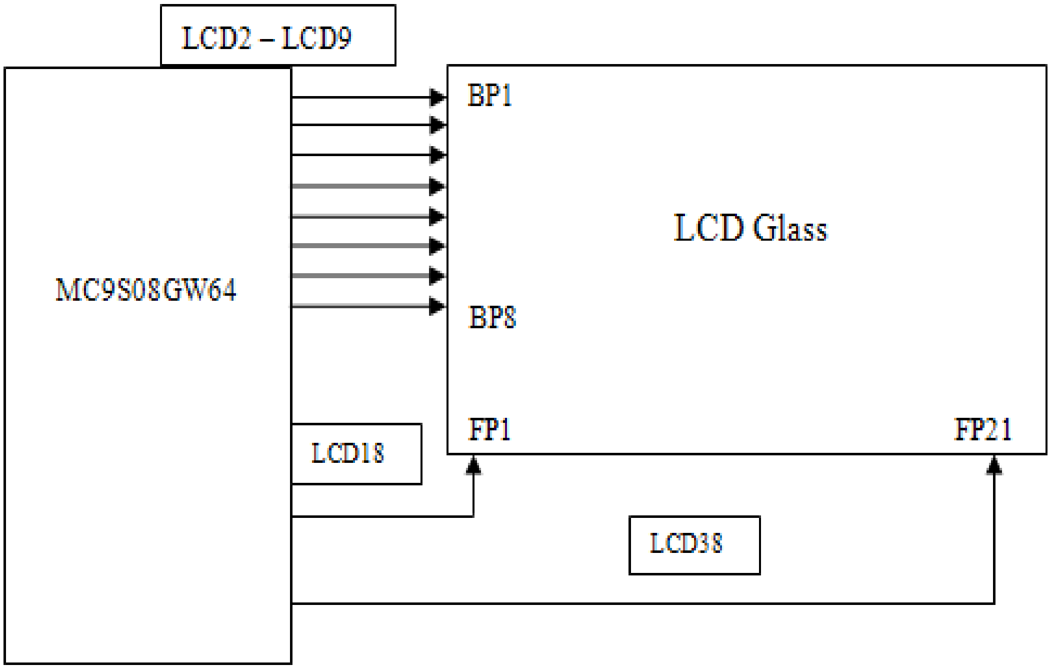


**Figure 7. Hardware connections for interfacing LCD glass with MC9S08GW64**

Follow the instructions below to configure the LCD block of the MC9S08GW64 to operate with an LCD glass, and to operate in charge pump mode with $V_{LL3}$ driven by $V_{DD}$.

1. Select the following configuration in *lcd.h*.

```
#define LCD_NO_OF_BACK_PLANES  8
#define LCD_CLOCK LCD_CLOCK_EXT
#define LCD_CLOCK_PRESCALER  0
#define LCD_CHARGE_PUMP_SELECT LCD_CHARGE_PUMP_ENABLE
#define LCD_LOAD_ADJUST  3
```

### NOTE

- For a 3V LCD glass select the configuration as `#define LCD_VSUPPLY LCD_VLL3_FROM_VDD`
- For a 5V LCD glass select the configuration as `#define LCD_VSUPPLY LCD_VLL2_FROM_VDD`

2. Depending upon the connection between the MCU LCD pins and the LCD glass pins, check the following defines.

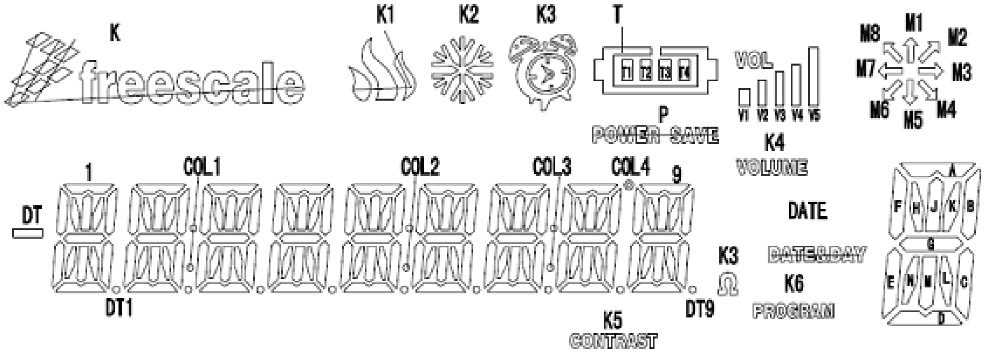
```
#define LCDPIN1   LCDWF2        /* Pin 1 of LCD glass connected to LCD2  */
#define LCDPIN2   LCDWF3        /* Pin 2 of LCD glass connected to LCD3   */
#define LCDPIN3   LCDWF4        /* Pin 3 of LCD glass connected to LCD4   */
#define LCDPIN4   LCDWF5        /* Pin 4 of LCD glass connected to LCD5   */
#define LCDPIN5   LCDWF6        /* Pin 5 of LCD glass connected to LCD6   */
#define LCDPIN6   LCDWF7        /* Pin 6 of LCD glass connected to LCD7   */
#define LCDPIN7   LCDWF8        /* Pin 7 of LCD glass connected to LCD8   */
#define LCDPIN8   LCDWF9        /* Pin 8 of LCD glass connected to LCD9   */
#define LCDPIN9   LCDWF18      /* Pin 9 of LCD glass connected to LCD18 */
#define LCDPIN10  LCDWF19     /* Pin 10 of LCD glass connected to LCD19 */
#define LCDPIN11  LCDWF20     /* Pin 11 of LCD glass connected to LCD20 */
#define LCDPIN12  LCDWF21     /* Pin 12 of LCD glass connected to LCD21 */
#define LCDPIN13  LCDWF22     /* Pin 13 of LCD glass connected to LCD22 */
#define LCDPIN14  LCDWF23     /* Pin 14 of LCD glass connected to LCD23 */
#define LCDPIN15  LCDWF24     /* Pin 15 of LCD glass connected to LCD24 */
#define LCDPIN16  LCDWF25     /* Pin 16 of LCD glass connected to LCD25 */
#define LCDPIN17  LCDWF26     /* Pin 17 of LCD glass connected to LCD26 */
#define LCDPIN18  LCDWF27     /* Pin 18 of LCD glass connected to LCD27 */
#define LCDPIN19  LCDWF28     /* Pin 19 of LCD glass connected to LCD28 */
#define LCDPIN20  LCDWF29     /* Pin 20 of LCD glass connected to LCD29 */
#define LCDPIN21  LCDWF30     /* Pin 21 of LCD glass connected to LCD30 */
#define LCDPIN22  LCDWF31     /* Pin 22 of LCD glass connected to LCD31 */
#define LCDPIN23  LCDWF32     /* Pin 23 of LCD glass connected to LCD32 */
#define LCDPIN24  LCDWF33     /* Pin 24 of LCD glass connected to LCD33 */
#define LCDPIN25  LCDWF34     /* Pin 25 of LCD glass connected to LCD34 */
#define LCDPIN26  LCDWF35     /* Pin 26 of LCD glass connected to LCD35 */
#define LCDPIN27  LCDWF36     /* Pin 27 of LCD glass connected to LCD36 */
#define LCDPIN28  LCDWF37     /* Pin 28 of LCD glass connected to LCD37 */
#define LCDPIN29  LCDWF38     /* Pin 29 of LCD glass connected to LCD38 */
```

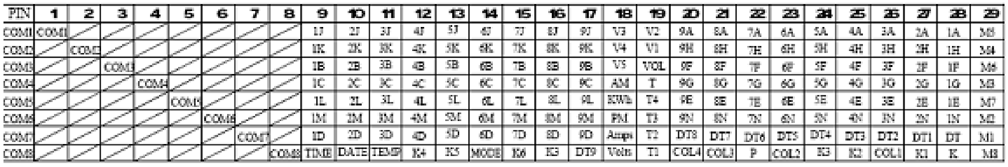
3. Depending upon the backplanes and frontplanes to be used, verify the settings of CONFIG_LCD_MODE. The macro is used to enable the LCD pins, enable the frontplanes and enable the backplanes as shown:

```
#define CONFIG_LCD_MODE
{
     LCDPEN0 = 0xFC; / *Enables the respective pins of the MCU connected with the
                        LCD glass */
     LCDPEN1 = 0x03;
```

```
            LCDPEN2  = 0xFC;
            LCDPEN3  = 0xFF;
            LCDPEN4  = 0x7F;
            LCDPEN5  = 0x00;
            LCDBPEN0  = 0xFC;   /*It enables the respective LCD pin as backplane*/
            LCDBPEN1  = 0x03;
            LCDBPEN2  = 0x00;
            LCDBPEN3  = 0x00;
            LCDBPEN4  = 0x00;
            LCDBPEN5  = 0x00;
            LCDPIN1  = 0x01;    /*It is used to enable the required segments*/
            LCDPIN2  = 0x02;
            LCDPIN3  = 0x04;
            LCDPIN4  = 0x08;
            LCDPIN5  = 0x10;
            LCDPIN6  = 0x20;
            LCDPIN7  = 0x40;
            LCDPIN8  = 0x80;
       }
```

4. Following is the snapshot of the data sheet of the LCD used in TWR-S08GW64.





| PIN | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM1 | COM1 | | | | | | | | 1J | 2J | 3J | 4J | 5J | 6J | 7J | 8J | 9J | V3 | V2 | 9A | 8A | 7A | 6A | 5A | 4A | 3A | 2A | 1A | M5 |
| COM2 | | COM2 | | | | | | | 1K | 2K | 3K | 4K | 5K | 6K | 7K | 8K | 9K | V4 | V1 | 9H | 8H | 7H | 6H | 5H | 4H | 3H | 2H | 1H | M4 |
| COM3 | | | COM3 | | | | | | 1B | 2B | 3B | 4B | 5B | 6B | 7B | 8B | 9B | V5 | VOL | 9F | 8F | 7F | 6F | 5F | 4F | 3F | 2F | 1F | M6 |
| COM4 | | | | COM4 | | | | | 1C | 2C | 3C | 4C | 5C | 6C | 7C | 8C | 9C | AM | T | 9G | 8G | 7G | 6G | 5G | 4G | 3G | 2G | 1G | M3 |
| COM5 | | | | | COM5 | | | | 1L | 2L | 3L | 4L | 5L | 6L | 7L | 8L | 9L | KWh | T4 | 9E | 8E | 7E | 6E | 5E | 4E | 3E | 2E | 1E | M7 |
| COM6 | | | | | | COM6 | | | 1M | 2M | 3M | 4M | 5M | 6M | 7M | 8M | 9M | PM | T3 | 9N | 8N | 7N | 6N | 5N | 4N | 3N | 2N | 1N | M2 |
| COM7 | | | | | | | COM7 | | 1D | 2D | 3D | 4D | 5D | 6D | 7D | 8D | 9D | Amps | T2 | DT8 | DT7 | DT6 | DT5 | DT4 | DT3 | DT2 | DT1 | DT | M1 |
| COM8 | | | | | | | | COM8 | TIME | DATE | TEMP | K4 | K5 | MODE | K6 | K3 | DT9 | Volts | T1 | COL4 | COL3 | P | COL2 | K3 | K2 | COL1 | K1 | K | M8 |

Depending on the backplanes used to for different characters, verify the following defines.

```
#define LCD_SEG_NONE   0x00    /* no segment is selected */
#define LCD_SEG_J   0x01
#define LCD_SEG_K   0x02
```

```
#define LCD_SEG_B   0x04
#define LCD_SEG_C   0x08
#define LCD_SEG_L   0x10
#define LCD_SEG_M   0x20
#define LCD_SEG_D   0x40
#define LCD_SEG_SP   0x80
#define LCD_SEG_A   0x01
#define LCD_SEG_H   0x02
#define LCD_SEG_F   0x04
#define LCD_SEG_G   0x08
#define LCD_SEG_E   0x10
#define LCD_SEG_N   0x20
#define LCD_SEG_SP2   0x40
#define LCD_SEG_SP3   0x80
#define LCD_SEG_M5   0x01
#define LCD_SEG_M4   0x02
#define LCD_SEG_M6   0x04
#define LCD_SEG_M3   0x08
#define LCD_SEG_M7   0x10
#define LCD_SEG_M2   0x20
#define LCD_SEG_M1   0x40
#define LCD_SEG_M8   0x80
```

5. Verify the symbols to be used, which are declared as shown

```
enum symbol {  TIME, DT, DATE, FREESCALE, FIRE, TEMP, SNOWFLAKE, VOLUME, ALARM, T, P,
 CONTRAST, MODE, AM,PM, OM, KWH, AMPS, VOLTS, PROGRAM, VOL, DTx, COLx, Vx, Tx, Mx};
```

6. Open *lcd.c* and verify the structure `lcd_char_table[]`. This table defines the segments required to display the characters and digits.

**For example:** To display 0 on the LCD, the following setting is required in the LCD used in TWR-S08GW64

```
{
    LCD_SEG_B | LCD_SEG_C | LCD_SEG_D,
    LCD_SEG_E | LCD_SEG_F | LCD_SEG_A
},
```

7. This sample code can be used to test the driver.

/*Initializes the LCD */

```
LCDInit();
```

/*All segments are on*/

```
LCD_ALLFP_ON
```

/*All segments are off*/

```
LCD_ALLFP_OFF
lcd_PrintString("hello");
```

# 5 Options for $V_{LL3}$

Table 8 describes the configurations for $V_{LL3}$ in MC980SGW64.

**NOTE**

$V_{LL3}$ and $V_{DD}$ if different at power ramp, will cause current leakage.

**Table 8. $V_{LL3}$ configurations for LCD**

| Configuration | Conditions |
|---|---|
| $V_{LL3}$ connected to external supply | This configuration applies if the user:<br>• Uses any of the 44 LCD pins as a full complementary digital GPIO.<br>• Varies $V_{DD}$ through board regulator to enable the LCD contrast control feature.<br>• Selects register bias mode or charge pump operating mode. |
| $V_{LL3}$ connected to $V_{DD}$ internally | This configuration applies if the user:<br>• Uses any of the 44 LCD pins as a full complementary digital GPIO. GPIOs toggling as outputs must be configured as open drain. User can use digital input functions for these GPIOs.<br>• Varies $V_{DD}$ through board regulator to enable the LCD contrast control feature.<br>• Selects register bias mode or charge pump operating mode. |
| $V_{LL3}$ connected to an external independent source through potentiometer network | This configuration applies if the user:<br>• Uses all 44 pins as LCD pins. User can also use digital input functions for GPIOs.<br>• Varies $V_{DD}$ through the external potentiometer to control the LCD contrast.<br>• Selects the preferred charge pumped mode. If register bias mode is selected then the user must ensure that the potentiometer resistance is less than the register bias network values selected by setting LCDSUPPLY_LADJ bits. This is required to avoid a drop in $V_{LL3}$ if internal register bias is activated. |

# 6 Conclusion

This driver provides a software interface between the custom glass pinout and the LCD module in low-power Freescale microcontrollers. Minimal changes are required to customize an LCD application. The user must meet specified hardware requirements. These routines are reliable platforms to migrate between the various families and reduces development time when using Freescale products.

# 7 References

*MC9S08GW64 Reference Manual* (document: MC9S08GW64RM)

**How to Reach Us:**

**Home Page:**
www.freescale.com

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

**USA/Europe or Locations Not Listed:**
Freescale Semiconductor, Inc.
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 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor 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 and all 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 products 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 the 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 and hold 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 claim 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-compliant 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.

freescale™
semiconductor