Application Note

Designing With MSPM0™ MCUs and Segment LCDs

Hao Mengzhen

ABSTRACT

Segment liquid crystal displays (LCDs) provide information to users in a wide variety of applications, from smart meters to electronic shelf labels (ESL) to medical equipment. New MSPM0™ microcontrollers include built-in, low-power LCD driver circuitry that allows the MSPM0 MCU to directly control the segmented LCD glass. This application note explains how segmented LCDs work, the features of the LCD module on the MSPM0 MCU, LCD hardware layout tips, and guidance for writing efficient and easy-to-use LCD driver software.

Table of Contents

Trademarks

MSPM0™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction: MSPM0 and LCD End Applications

There are a number of common applications where MSPM0 microcontrollers with built-in LCD drivers are a great fit. In particular, applications that require both a segmented LCD display, as well as limited battery life or current consumption. Examples include low-power LCD handhelds (like a watch or other device), blood glucose meters, appliances, water meters, electronic shelf labels, and one-time password tokens. The combination of rich analog and peripheral interfaces provided by MSPM0 devices, along with the built-in segment LCD display driver, enable a diverse array of applications with a compelling set of features all in one system on chip (SOC).

2 MSPM0 LCD Portfolio

Table 2-1 describes the LCD module features that are available on MSPM0 MCUs.

Table 2-1. MSPM0 LCD Module Feature Table

Parameter	LCD
Number of segments supported(1)	408/8-mux
Mux mode supported	8, 7, 6, 5, 4, 3, 2, Static
Segment functionality against port pin selection	Individual selection
Flexible configuration for COM and Segment pins	YES
LCD clock selection	LFCLK
LCD clock divider availability	1 to 32
Interrupt capabilities	YES (3 sources)
Whole display blinking	YES
Programmable blinking frequency	YES
Individual segment blinking capabilities with separate memory	Yes (Static and 2-4 Mux mode)
Dual memory display	YES
Bias mode	Static, 1/3 and 1/4
LCD bias generation using resistive network	External or Internal
Device protection against no connected capacitance on LOADCAP when charge pump is used	NO need for protection (A 0.1µF or larger capacitor must be connected from LOADCAP0 and LOADCAP1 pins)
Charge pump voltage with external voltage reference	3 or 4 × V_{ref} depends on bias mode
Low-power waveforms mode	YES

(1) LCD pin count varies with device and package. See device-specific data sheet for detailed information.

3 Segmented LCD Operation

The following sections explain the basic operation of all LCDs to create a background of MSPM0 LCD driver features.

3.1 LCD Structure (Simplified)

Figure 3-1 shows a simplified version of the structure of a segment LCD display. The display consists of two polarizers rotated 90 degrees from each other to polarize light coming into the display, liquid crystals between the polarizers with electrodes to apply a charge, and a reflective backing to reflect light that gets through all the layers of the display.

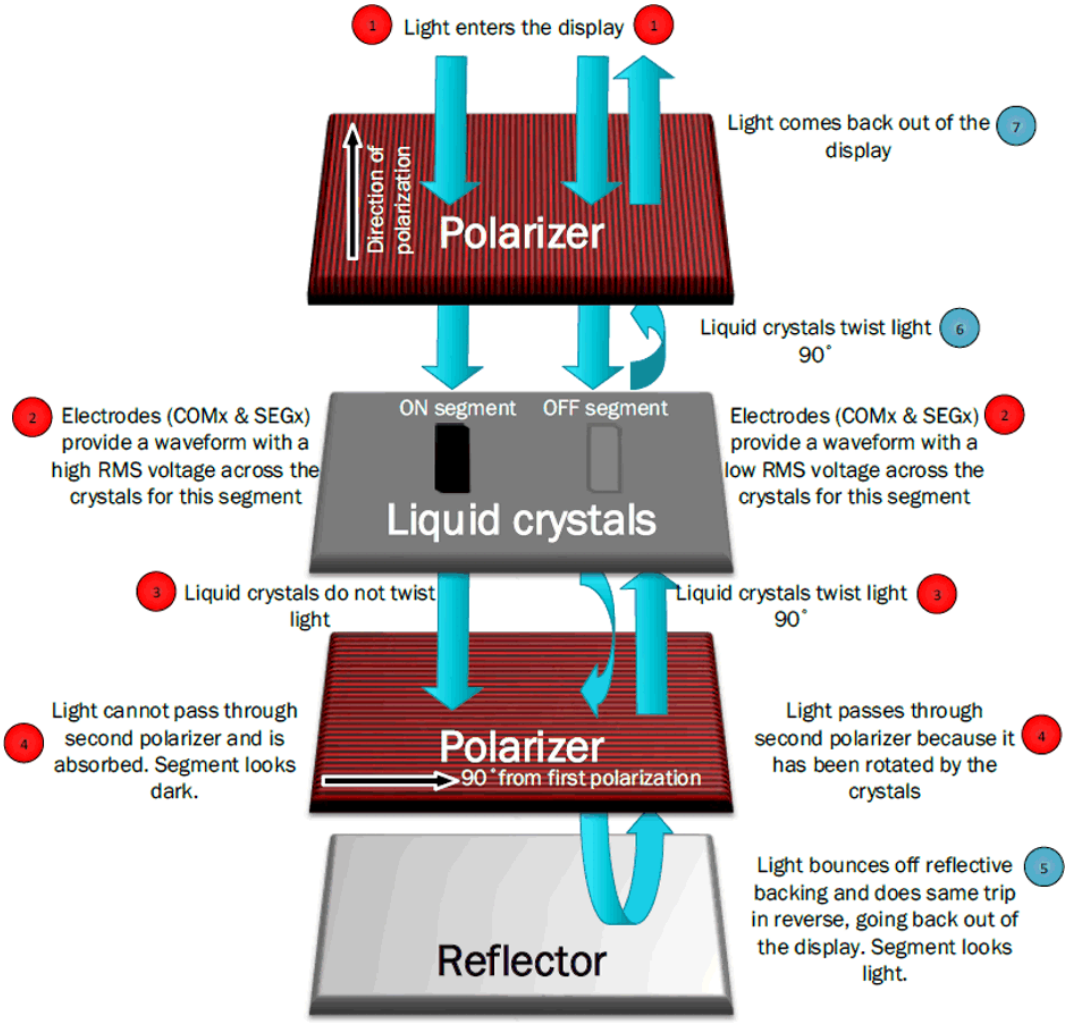


Figure 3-1. Segmented LCD Structure and Theory

When no charge is applied to the electrodes for a particular segment, the segment is "off," or gray. In this normal state, the liquid crystals have a twisted structure that turns the light 90 degrees. First, when no charge is applied, light comes in the first polarizer and emerges polarized in one direction. Then, the crystals turn the light 90 degrees as the crystals pass through the light - which allows the light to be able to pass through the second polarizer because, the second polarizer is rotated compared to the first polarizer. Finally, the light reflects off of the reflective backing and does the same path in reverse. Because the light is reflected back, the LCD segment appears light or gray.

When a charge is applied to the electrodes for a segment, the segment is "on," or black. In the charged state, the crystals untwist, so they do not turn the light and let the light passes through directly. First, when a charge is applied, light comes in the first polarizer and emerges polarized in one direction. Then, the crystals simply allow the light to pass straight through without turning the light. Because the second polarizer is at 90 degrees from the first one, the light is not able to pass through and is instead absorbed. This makes the segment appear dark in color.

3.2 LCD Drive Basics

LCDs must drive with AC signals. A DC level on an LCD segment damages the LCD, typically less than 50mV DC voltage is allowed. The MSPM0 LCD module generates these types of AC waveforms automatically so that the user only has to specify if a segment is on or off – the internal hardware does the rest.

LCD segments have a charge applied to the crystals between two electrodes – a COMx line and an Sx segment line. The potential difference applied by these two electrodes is the waveform seen by the LCD segment.

The RMS voltage presented on an LCD segment determines whether the LCD segment is on or off. The example waveforms in Figure 3-2 show resulting waveforms (combination of COMx and Sx pin signals) of both an on and off segment. The on segment has a much larger RMS voltage applied on the on segment, rather than the off segment. Note that both segments have waveforms that have net zero DC voltage, but the RMS voltage of the on segment is higher, which causes the segment to turn on and look dark.

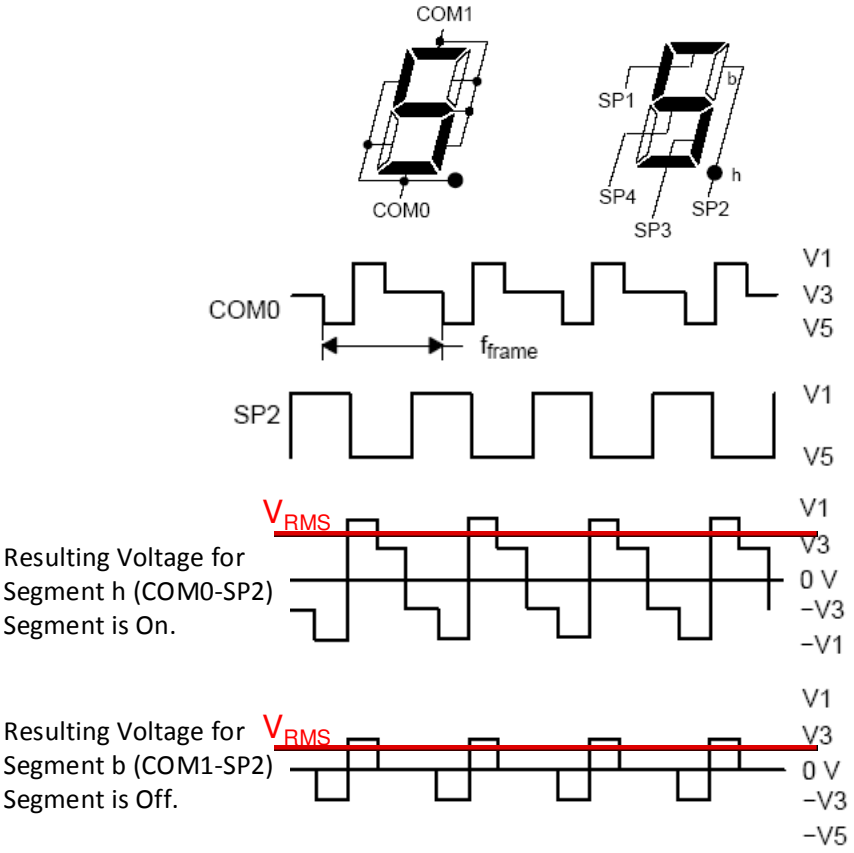


Figure 3-2. Example LCD AC Waveforms

4 MSPM0 LCD Features

This section explains the features available on MSPM0 LCD modules. Since the MSPM0 family is evolving rapidly, always make sure to check the data sheet for the particular device to see what LCD feature is present in the device and how many pins are available for LCD output.

4.1 Muxing

Segmented LCDs use multiplexing (muxing) to limit control pin count. Types of displays include static (no muxing) and 2 through 8-mux. The notation N-mux means each segment pin Sx drives N segments on the display – this also means there are N common (COMx) pins. Each LCD segment on the display is driven by the combination of a COMx pin and Sx pin, providing a difference in potential across the liquid crystals for that segment.

Muxing allows a limited number of pins to control a larger number of segments. If there is an 8-mux LCD display, then there are 8 COM pins and each segment (Sx) pin drives 8 segments. So when using an 8-mux capable MSPM0 with 51 Sx pins available (S0-S50), then MSPM0 is able to control 408 segments with only 8 (COMx) + 51 (Sx) = 59 pins.

MSPM0 devices with 8-mux mode support up to 408 segment displays. However, make sure to check the device-specific data sheet to see how many segments the particular device supports, the number of supported segments is limited not only by the muxing capability of the LCD module, but also by the number of LCD pins available on the particular device in a particular pin-package.

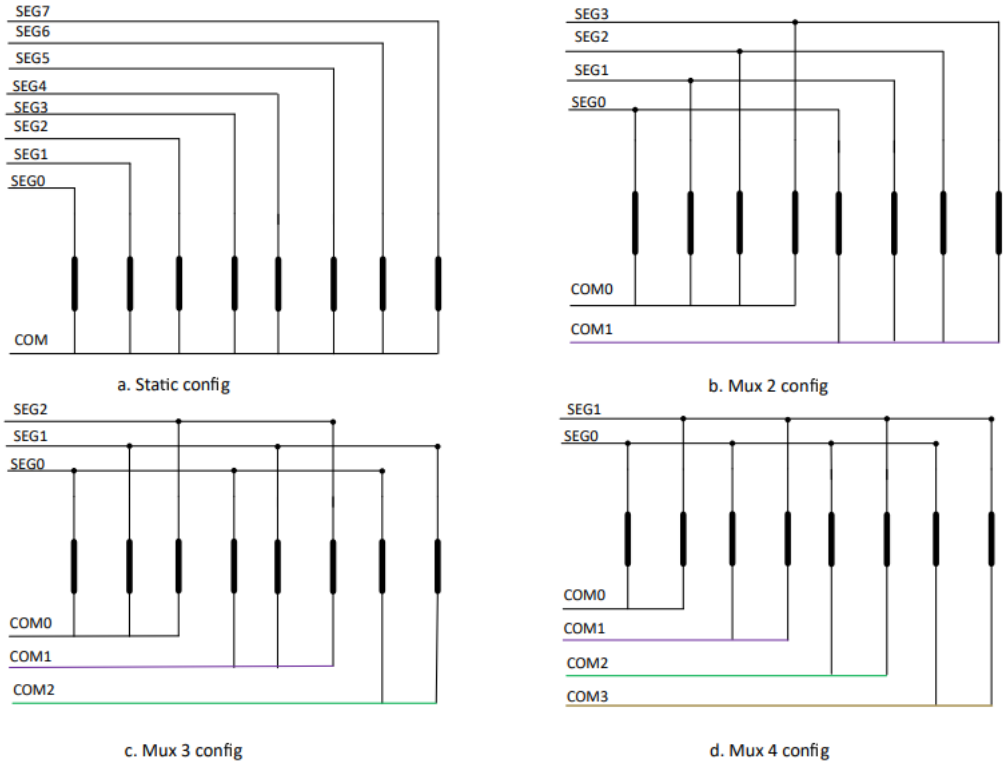


Figure 4-1. MUX Configurations

4.1.1 Muxing Example

Figure 4-2 shows a 4-mux 1/3 bias waveform. Each segment is controlled by signals on 2 pins – a COMx pin and an Sx segment pin. The signals shown are the waveforms applied on the electrodes for the segment, so the potential difference between the Sx and COMx signal is what is applied to the liquid crystals in that area. This potential difference is what is shown as the voltage in the resultant waveforms in Figure 4-2 (COM0-S0 and COM1-S1).

In this example, the COM0-S0 waveform has a high RMS voltage so the segment is on even though the waveform has a net zero DC voltage. The COM1-S1 waveform has a low RMS voltage, so the segment is off. While the waveforms look complex, they are generated automatically by the MSPM0 LCD module – the user only specifies the basic settings of the LCD and indicates which segments is on or off.

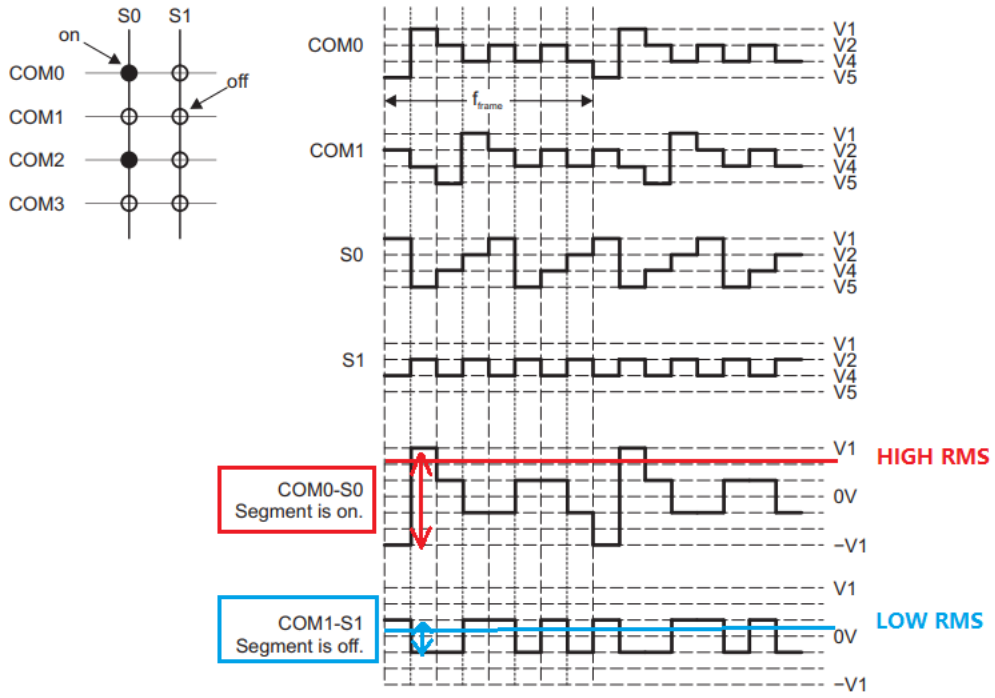


Figure 4-2. 4-Mux 1/3 Bias

4.2 Voltage Generation

There are 5 levels (V1, V2, V3, V4, V5) of voltages used to generate different waveforms on segment and common pins, depending on the bias mode. Check the voltage level in different bias mode in Table 4-1.

Table 4-1. Voltage Level in Different Bias Modes

Voltage Level	Static	1/3 bias mode	1/4 bias mode
V1(V_{LCD})	V_{LCD}	V_{LCD}	V_{LCD}
V2	NA	V_{LCD} × 2/3	V_{LCD} × 3/4
V3	NA	NA	V_{LCD} × 1/2
V4	NA	V_{LCD} × 1/3	V_{LCD} × 1/4
V5	0	0	0

The highest voltage level V1 is generated by V_{LCD}. V_{LCD} is sourced externally or from the internal charge pump, as discussed in Section 4.2.1. To produce the rest of the voltage levels in the LCD waveforms V2 through V5, the module produces bias voltages at fractions of V_{LCD}.

Divide the bias voltages V2 through V5 down from V_{LCD} internally or with an external resistor network. Customer can choose to use any combination of the source of V_{LCD} and the bias voltage generation method in their application. Table 4-2 shows an example of the possible internal and external bias options with different source of V_{LCD} in one of the LCD modules.

Table 4-2. Voltage Generation Mode

Mode	Description
0	Voltage generation from external reference and external resistor divider
1	Voltage generation from AVDD and external resistor divider
2	Voltage generation from external reference and internal resistor divider
3	Voltage generation from AVDD and internal resistor ladder
4	Voltage generation from charge pump with external supply
5	Voltage generation from charge pump with AVDD
6	Voltage generation from charge pump with external reference on R13
7	Voltage generation from charge pump with internal reference on R13

Generating bias voltages internally is simple because no external components are required – the module internally divides down the voltage. However, generating bias voltages externally instead may be lower power. External biasing requires the user to provide an external resistor divider to create the voltages V2 through V5 - the resistor divider used depends on the biasing mode – static, 1/3, or 1/4 bias, as shown in Figure 4-3. The resistors in the divider must all be the same value, but the size used may depend on the particular display used in the design.

Changing the external resistor values impacts both current consumption and contrast. Larger resistors cause less current consumption in the resistor ladder, saving power. However, if resistors are too large, the contrast may not be good or even for all segments. Experimentation with different sizes of resistor is usually needed in a design to find the best combination of performance versus current consumption.

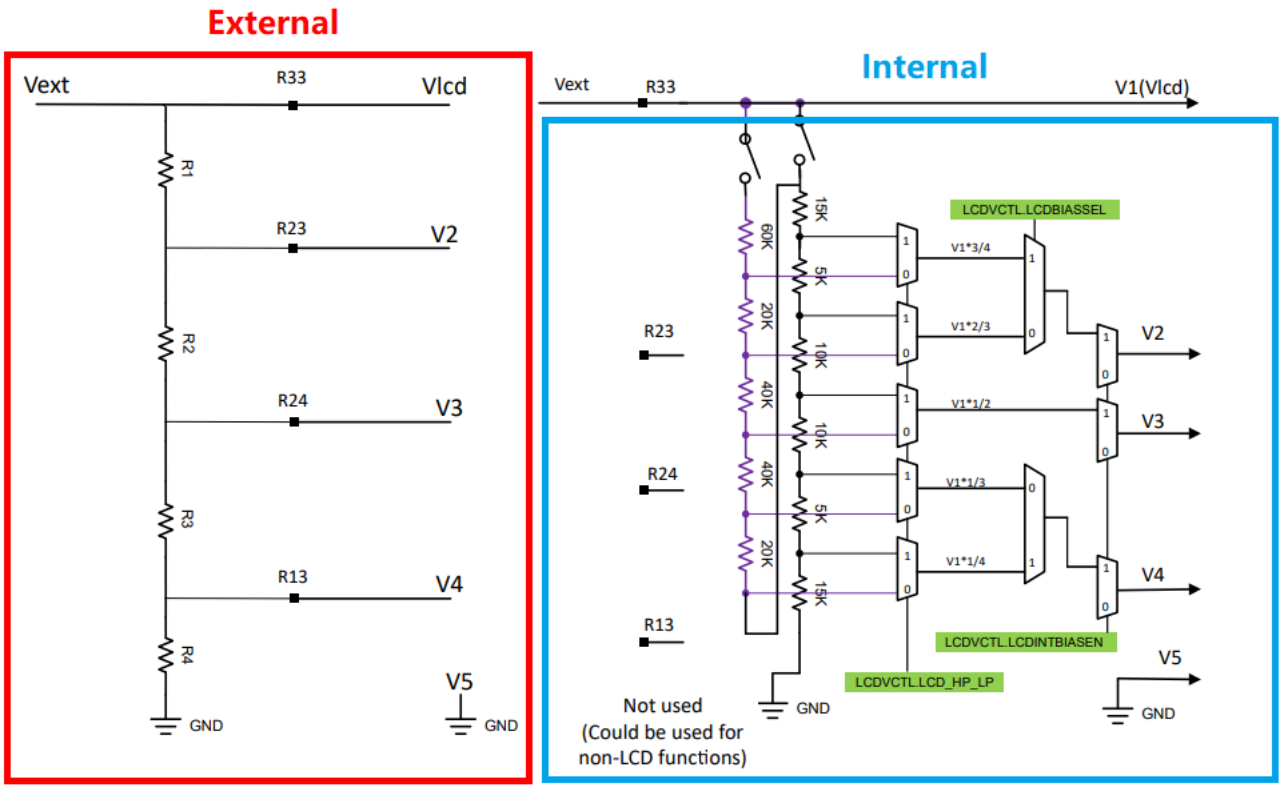


Figure 4-3. Voltage Generation Configurations

4.2.1 Charge Pump

The V_{LCD} voltage sets the voltage level of V1, the highest LCD voltage in the waveforms. This is set in software that is sourced from the AVDD, the internal charge pump, or an external source. MSPM0 LCD module includes a built-in charge pump.

There are several advantages of using a charge pump for generating the V_{LCD} voltage. Firstly, the charge pump provides a regulated voltage to the LCD to keep a stable voltage output for the display. Next, using a charge pump allows for the V_{LCD} to be set to a different voltage level that is independent of D_{VCC}. Therefore, the V_{LCD} is set at the best level for the particular LCD display, and keeps good contrast, even as the battery in the system drains.

The built-in charge pump has programmable voltage levels for use with different segmented displays. The allowed maximum operating voltage comes from the design of the particular LCD glass. Setting a different V_{LCD} changes the contrast ratio, so having software-configurable voltage levels from the charge pump allows contrast control through software. For example, using a lower V_{LCD} provides less contrast, but also has less current consumption; the user is able to experiment with this trade-off in the final design.

The charge pump requires an external capacitor for operation. The charge pump capacitor on the MSPM0 LCD module is connected between two pins instead of directly to ground, so the capacitor not being present is not damaged the MCU (though the charge pump does not work properly).

When charge pump is referenced from internal reference source, the internal reference circuit is able to operate in one of two modes, continuous or sampled. In the continuous mode, a continuous reference is generated if the LCDREFEN bit is set. In sampled mode, the voltage on the R13 pin is maintained by a capacitor for lower power consumption.

Reference the charge pump to follow an external source. Using an external source is useful if multiple MSPM0 MCUs are used together to control a single larger segmented display that is controlled with a single device.

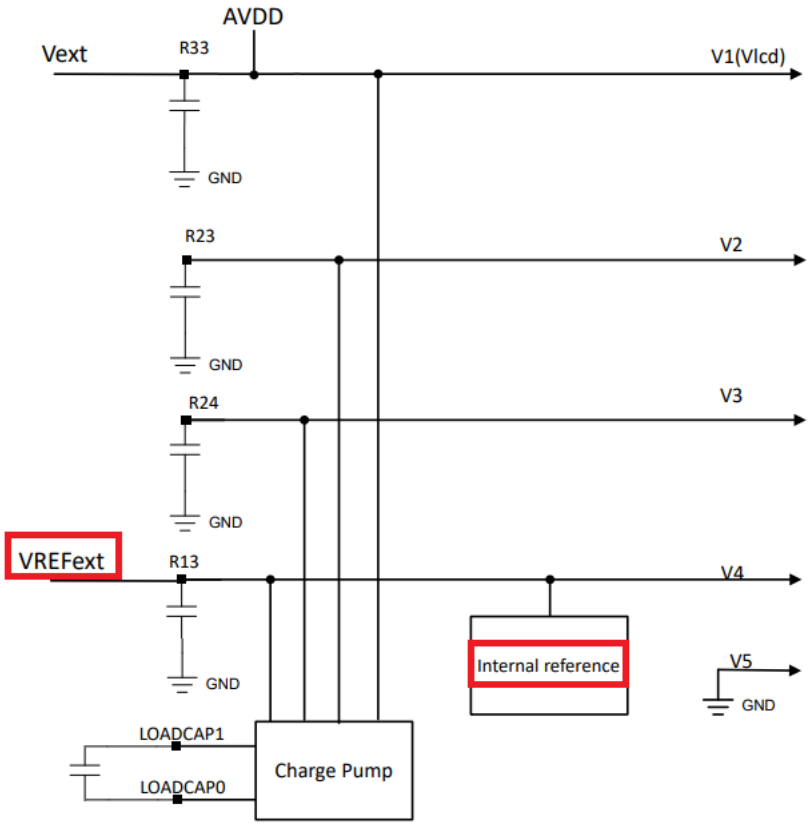


Figure 4-4. Charge Pump Reference

4.2.2 Contrast Control

As mentioned in Section 4.2.1, when using the charge pump, software controls the V_{LCD} of the display. The V_{LCD} allows the user to easily adjust contrast in software. Changing the V_{LCD} adjusts all of the other LCD voltages, regardless of internal or external biasing, because all of the other voltages are divided down from V_{LCD}.

When using an external bias resistor ladder for generating the bias voltages, the sizing of R impacts contrast. If contrast is not even across all segments, reduce the size of the resistors in the bias ladder at the tradeoff of additional current consumption (see Section 4.2 on biasing).

The different biasing modes and the particular LCD used also impacts contrast ratio. Seen in Table 4-3, the contrast ratio is represented as $V_{RMS,ON}/V_{RMS,OFF}$, or the RMS voltage from the waveforms for a segment that is on, divided by the RMS voltage from the waveforms for a segment that is off. The higher the contrast ratio, the greater the difference in appearance of an on-segment versus an off-segment. Table 4-3 shows that there is better or worse contrast depending on the bias configuration and muxing of the display – since these settings affect the characteristics of the output waveforms. As shown in Table 4-3, higher MUX rates tend to have lower contrast ratios, therefore performance is more sensitive to any tradeoffs that affect contrast. These ratios mean that a more sensitive LCD glass with a better threshold, or other factors to provide better contrast (such as higher V_{LCD}, smaller bias resistors, or the techniques from *Driving Large LCDs*), is needed for the desired LCD performance.

A typical approach to determine the V_{LCD} for good contrast, is to use the threshold voltage when the contrast is 10% and with the $V_{RMS,OFF}/V_{LCD}$ ratio from the user's guide table to calculate a recommended V_{LCD} using Equation 1.

$$V_{LCD} = V_{th,10\%} \div \left(V_{RMS,OFF}/V_{LCD}\right) \quad (1)$$

The $V_{th,10\%}$ is a characteristic of the fluid used in the LCD display which varies with the display. The display information provided by the manufacturer typically lists a visual threshold voltage for 10%.

Some configurations trade off a reduced contrast ratio for a reduction of the full-scale LCD voltage V_{LCD} used. For example, on some modules 1/3 bias give a better contrast, but the 1/3 bias mode may require a higher V_{LCD}. See the LCD module-specific section in the *MSPM0 L-Series 32MHz Microcontrollers* technical reference manual for more information pertaining to the particular LCD module and contrast ratio. User's guides have tables like Table 4-3 with information specific to that module's muxing and bias options.

Table 4-3. LCD Voltage and Biasing Effect on Contrast

Mode	Bias Configuration	Voltage Levels	$V_{RMS,OFF}/V_{LCD}$	$V_{RMS,ON}/V_{LCD}$	Contrast Ratio $V_{RMS,ON}/V_{RMS,OFF}$
Static	Static	V1, V5	0	1	1/0
2-mux	1/3	V1, V2, V4, V5	0.333	0.745	2.236
2-mux	1/4	V1, V2, V3, V4, V5	0.395	0.729	1.846
3-mux	1/3	V1, V2, V4, V5	0.333	0.638	1.915
3-mux	1/4	V1, V2, V3, V4, V5	0.356	0.612	1.719
4-mux	1/3	V1, V2, V4, V5	0.333	0.577	1.732
4-mux	1/4	V1, V2, V3, V4, V5	0.331	0.544	1.643

4.3 LCD Clocking

MSPM0 LCD modules include a internal timing generation that does not require timer modules. Figure 4-5 is the LCD clock module diagram in MSPM0 device.

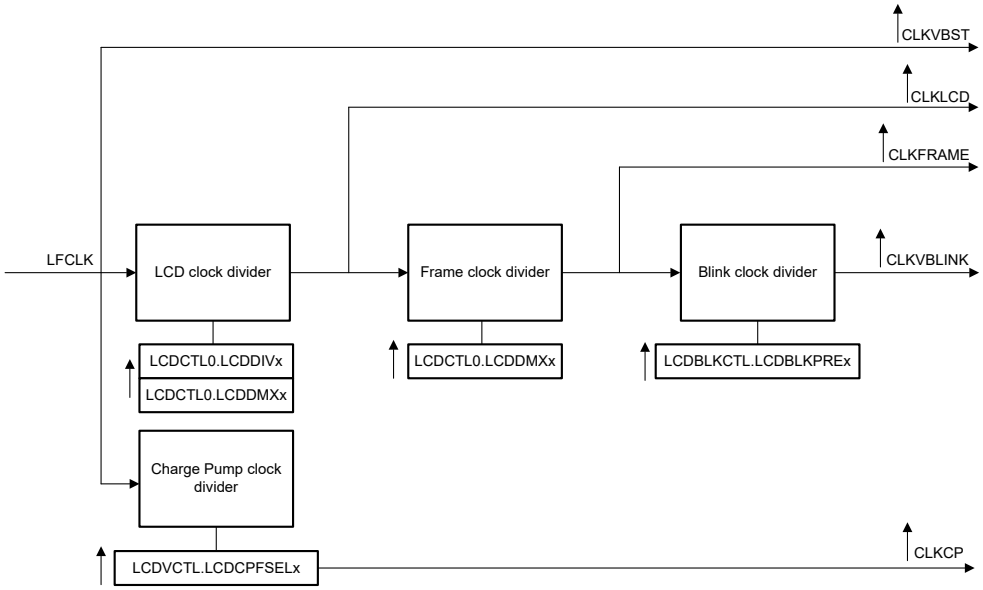


Figure 4-5. LCD Clock Module Diagram

The LCD module is sourced by the LFCLK. To achieve the desired frequency for f_{CLKLCD}, scale and divide within the module. f_{CLKLCD} frequencies are usually low (<1kHz), so typically the timing for the module is sourced from low-frequency LFCLK, this clock is also typically available in low-power modes (except SHUTDOWN mode). f_{CLKLCD} is the frequency that generates the timing for the common COMx and segment Sx signals. As shown in the below Table 4-4, when the MUX mode (LCDMXx) is set, the MUXDIVIDER is set automatically corresponding to the LCDMXx. Users are able to change the LCDCTL0.LCDDIVx to control the output frequency of the f_{CLKLCD}.

Table 4-4. CLKLCD Calculation

Clock	Equation	
CLKLCD	$f_{CLKLCD} = f_{CLKLFCLK} \div (LCDCTL0.LCDDIVx + 1) \times MUXDIVIDER$	
	LCDMXx	MUXDIVIDER
	0	64
	1	32
	2	16
	3	16
	4	12
	5	8
	6	8
	7	8

$f_{CLKFRAME}$ is the frame frequency from LCD display's datasheet. The display has a range of allowed frame frequencies, which gives the user options when choosing an f_{CLKLCD}. Lower frequencies lead to lower current consumption, while higher frequencies result in less flickering on the display. Experiment with different f_{CLKLCD} frequencies to determine what setting yields an acceptable appearance on the LCD with the least current consumption. Calculate the required $f_{CLKFRAME}$ using Equation 2.

$$f_{CLKFRAME} = f_{CLKLCD} \div [(LCDCTL0.LCDMXx + 1) \times 2] \quad (2)$$

For example, Equation 3 and Equation 4 show the outcome with a 4-mux mode, where LCDCTL0.LCDMXx is 3, and LCDCTL0.LCDDIVx is 0.

$$f_{CLKLCD} = 32768 \div [(0 + 1) \times 16] = 2048h = Hz \quad (3)$$

$$f_{CLKFRAME} = 2048 \div [(3 + 1) \times 2] = 256Hz \quad (4)$$

f_{CLKBLK} is used to flash the segments on and off when blinking is enabled. Segments are turned off when CLKBLK is 0. The required f_{CLKBLK} is calculated using Equation 5.

$$f_{CLKBLK} = f_{CLKFRAME} \div \left[2^{(LCDBLKPREx + 1)}\right] \quad (5)$$

f_{CLKCP} is used to generate four phase clocks to the charge pump module, when the charge pump is enabled. The charge pump operates by transferring charge from load capacitor connected between LOADCAP0 and LOADCAP1 pins to other capacitors connected between R33, R23, R24, R13 and ground. The charge pump requires four non-overlapping clocks to control the switches. The required f_{CLKCP} is calculated using Equation 6.

$$f_{CLKCP} = f_{CLKLKCLK} \div (LCDVCTL.LCDCPFSELx + 1) \quad (6)$$

$f_{CLKVBST}$ is used to boost the voltage on control signals used to control the switches, its frequency is same as LFCLK.

4.4 LCD Memory and Blinking Mode

Software controls the segments by using two memory blocks, LCDMx and LCDBMx. As shown in Figure 4-6, these two memories are written from the VBUSP bus (system bus) and the contents are read using internal logic. The contents read from these two memory blocks are passed through the blinking override logic before the 64bit LCDVAL is passed to the IO buffers. Each of the 64 LCD pins are configured as either common line or segment line, using the LCDCSSELx registers.

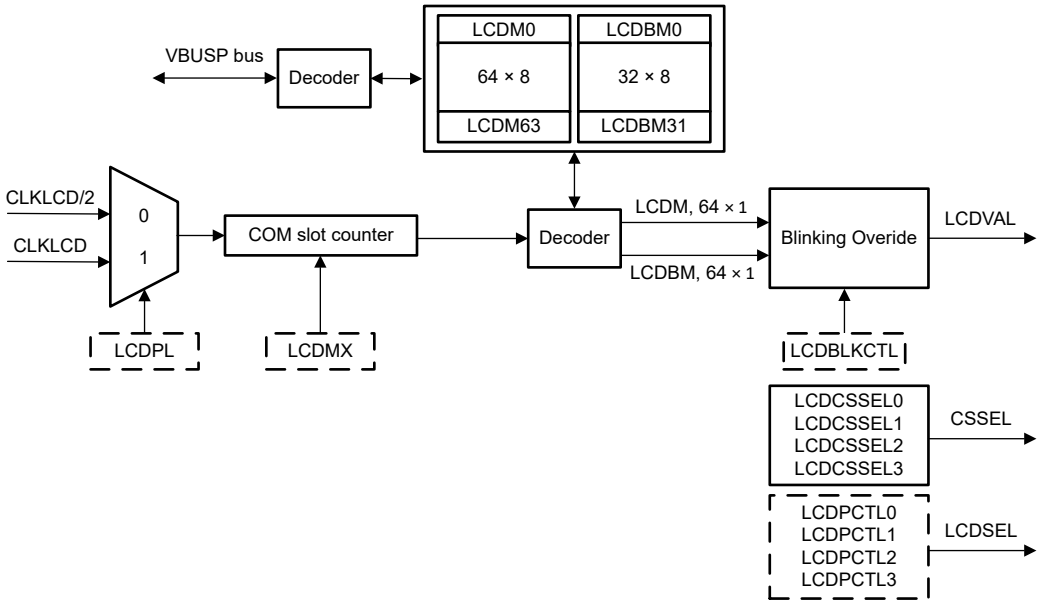


Figure 4-6. LCD Memory Diagram

4.4.1 LCD Memory Organization

User software selects which segments are on or off by using the LCD memory registers. Each bit represents a single LCD segment connected to a COMx and Sx pin pair. The row (or byte) corresponds to the Sx pin, and the columns (or each bit within the byte) correspond to the COMx pins. In 2-mux through 4-mux modes, the upper and lower nibbles of each row correspond to different Sx pins. In this case, only up to 4 bits in the byte are needed since there are only up to 4 COMx lines, so a single byte sets two Sx pins. In 5-mux through 8-mux modes, there are more than 4 COMx lines so the whole row (byte) is required for each segment Sx pin.

Figure 4-7 shows an example memory configuration for 4-mux mode. The L2 and L3 segment pins in this example correspond to the lower and upper nibbles of the byte at LCD memory offset 1. To control the segment connected to COM0 + L2 or COM0 + L3, the software sets the highlighted bits to either 1 or 0 to indicate the desired LCD segment state "on" or "off."

Memory Row	COM3 (MSB-7)	COM2	COM1	COM0	COM3	COM2	COM1	COM0 (LSB-0)
0	L1	L1	L1	L1	L0	L0	L0	L0
1	L3	L3	L3	L3	L2	L2	L2	L2
2	L5	L5	L5	L5	L4	L4	L4	L4
3	L7	L7	L7	L7	L6	L6	L6	L6
4	L9	L9	L9	L9	L8	L8	L8	L8
5	L11	L11	L11	L11	L10	L10	L10	L10
6	L13	L13	L13	L13	L12	L12	L12	L12
7	L15	L15	L15	L15	L14	L14	L14	L14
8	L17	L17	L17	L17	L16	L16	L16	L16
9	L19	L19	L19	L19	L18	L18	L18	L18
10	L21	L21	L21	L21	L20	L20	L20	L20
11	L23	L23	L23	L23	L22	L22	L22	L22
12	L25	L25	L25	L25	L24	L24	L24	L24
13	L27	L27	L27	L27	L26	L26	L26	L26
14	L29	L29	L29	L29	L28	L28	L28	L28
15	L31	L31	L31	L31	L30	L30	L30	L30
16	L33	L33	L33	L33	L32	L32	L32	L32
17	L35	L35	L35	L35	L34	L34	L34	L34
18	L37	L37	L37	L37	L36	L36	L36	L36
19	L39	L39	L39	L39	L38	L38	L38	L38
20	L41	L41	L41	L41	L40	L40	L40	L40
21	L43	L43	L43	L43	L42	L42	L42	L42
22	L45	L45	L45	L45	L44	L44	L44	L44
23	L47	L47	L47	L47	L46	L46	L46	L46
24	L49	L49	L49	L49	L48	L48	L48	L48
25	L51	L51	L51	L51	L50	L50	L50	L50
26	L53	L53	L53	L53	L52	L52	L52	L52
27	L55	L55	L55	L55	L54	L54	L54	L54
28	L57	L57	L57	L57	L56	L56	L56	L56
29	L59	L59	L59	L59	L58	L58	L58	L58
30	L61	L61	L61	L61	L60	L60	L60	L60
31	L63	L63	L63	L63	L62	L62	L62	L62

Figure 4-7. LCD Memory Map Example

4.4.2 Blinking

MSPM0 LCD module support blinking mode. Some or all the segments are configurable to flash on and off (blinking). Blinking modes are applicable only in MUX modes 1 to 4.

For devices that support both full screen and individual segment blinking, the blinking happens automatically at a particular frequency. The blink frequency f_{CLKBLK} is configurable, but must be less than the frame frequency.

When using individual segment blinking, the blink memory controls whether a segment blinks. Software controls the segments by using two memory blocks, LCDMx and LCDBMx. As shown in Figure 4-8, these two memories are written from the VBUSP bus (system bus) and the contents are read using internal logic. The contents read from these two memory blocks are passed through the blinking override logic before the 64bit LCDVAL is passed to the IO buffers. The blink memory is used as a secondary display memory – the LCDMEMCTL.LCDDISP bit controls which memory is in use for the display.

There are three blinking modes supported on MSPM0 devices.

1. **Blink Selected:** The bits in the LCDBM memory determine the LCD outputs which participate in blinking. If a bit is set in LCDBM memory, the corresponding output is blinked, if the bit is 0, the corresponding bit is not blinked.
2. **Blink all:** All the segments participate in blinking.
3. **Blink toggle:** The state of a given segment alternates between values set in LCDM and LCDBM memory bits.

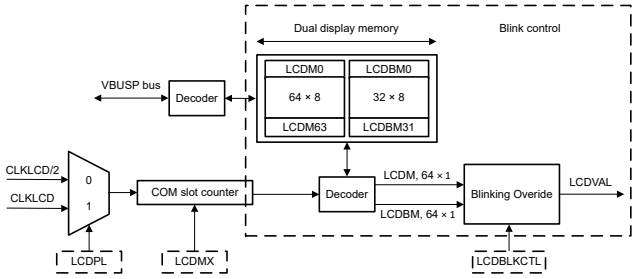


Figure 4-8. LCD Blinking and Dual Display Memory

4.5 LCD Output Pin Configuration

MSPM0 have LCD pins muxed with analogy and digital I/O functions. These pins are configurable in software for other functions when the application is not using all of the LCD pins. Each pin is individually configurable for either GPIO or LCD function, using a setting in the IOMUX registers.

On MSPM0 devices, each pin is configurable for GPIO or LCD function, and each pin set for LCD function is further configurable as either a COMx or Sx pin. Additional flexibility allows for completely configurable LCD pins, meaning users have the most flexible layout. This confirms that the LCD layout is achievable in a single layer. Since the COMx pins on the MSPM0 side are no longer fixed to a specific set of pins, these pins are configured to help accommodate where the COMx pins are located on the display side. Further, some layout mistakes are fixed in software instead of having to create a new PCB design. For more information regarding LCD layout, see Section 5.1. For more details about LCD flexible pin configuration, see the *MSPM0 L-Series 32-MHz Microcontrollers* technical reference manual.

4.6 Low Power Mode Feature

MSPM0 LCD modules are designed with low power as a key feature. In addition to some of the low power options discussed in the earlier sections (such as the adjustable charge pump voltage level, and options for external biasing), the charge pump is only turned on for a small percentage of the overall operation of the module. The charge pump runs with a low duty cycle so that its peak current is only seen for a very small portion of the overall time that the LCD is on; this means that the LCD keeps a very low average current.

The peak charge pump current is found in the data sheet for the particular device. There is also usually a specification for the time to charge the C_{LCD} charge pump cap when the cap is discharged, the peak current and the charge time help to determine an average current for the LCD module. The worst case is when the charge

pump is at the peak current $I_{CC,Peak,CP}$ for the time $t_{LCD,CP,on}$ when C_{LCD} has been discharged. The rest of the time, the module is in a much lower current state. In addition, using a low-leakage capacitor for C_{LCD} helps to reduce the energy consumption as the charge pump runs with a lower duty cycle.

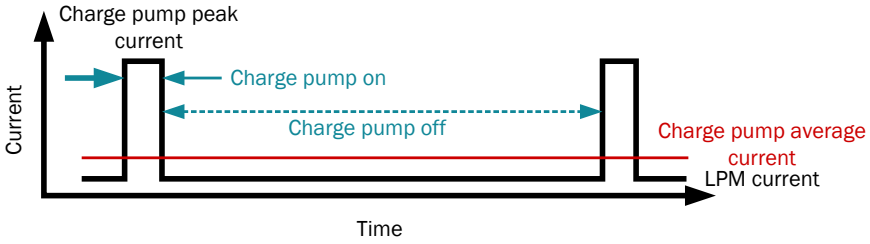


Figure 4-9. Low Charge Pump Duty Cycle

MSPM0 LCD module also has a setting for using lower power versions of the LCD waveforms. The lower power versions of the waveforms have the voltage sequence re-shuffled so that certain time slots are grouped together. This makes for fewer switching events on each pin and lower current consumption. By reducing the number of toggles on the switch controls in a given LCD frame, the power consumption is reduced. Figure 4-10 shows an example of the normal and lower power versions of waveforms for 8-mux mode.

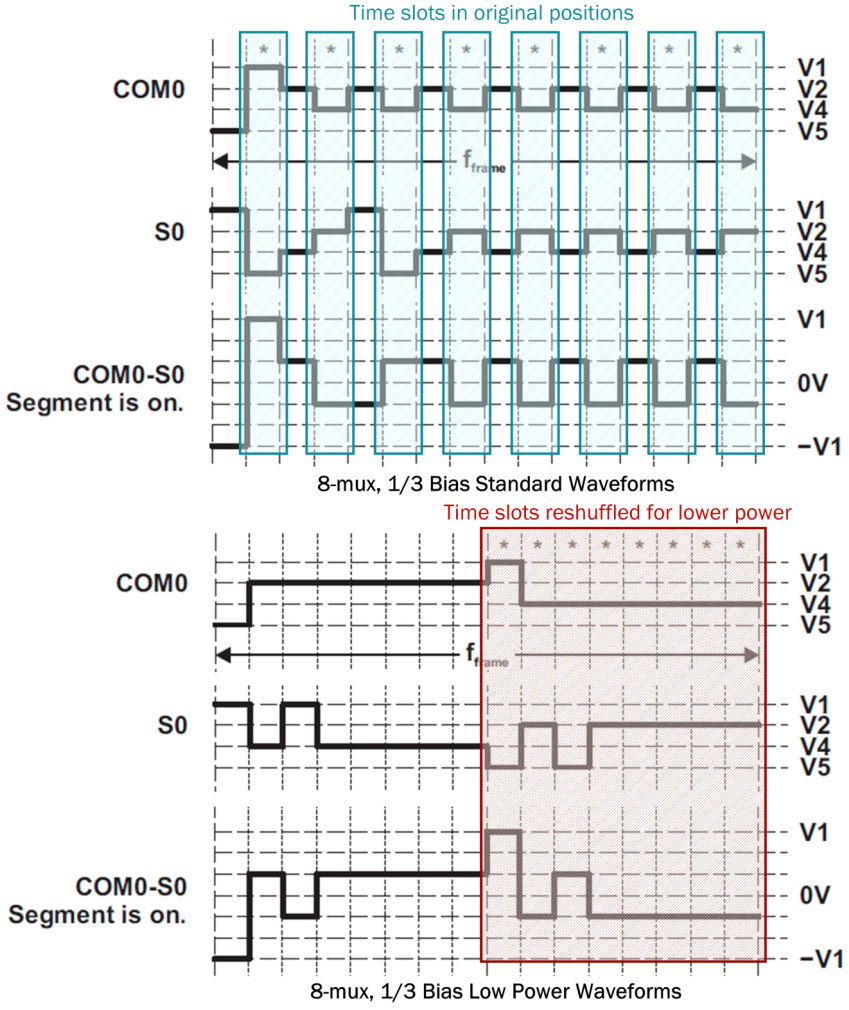


Figure 4-10. Low-Power Waveforms Example

The LCD module is operational in all power modes except SHUTDOWN mode, allowing for a new level of ultra-low power and enabling new LCD applications where the power supply is very limited. See the device-specific data sheet for more details about LCD current consumption in low-power modes.

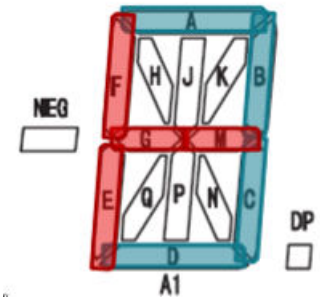
5 LCD Layout and Software Considerations

Choosing the right LCD and carefully choosing which MSPM0 pins to connect to particular pins of the display makes a big difference in the ease of use of code and code efficiency. This efficiency ties into how the display MUXes different areas together onto the same segment pin, in relation to what is displayed on the segment (for example, alphabet characters or numbers). While deciding on a layout, take into consideration the way that the MSPM0 LCD memory is structured in different muxing modes to provide efficient and easy to use software.

5.1 LCD Layout Tips

Depending on the muxing used, multiple Sx pins are required to display an entire digit or character on the screen. Which display areas are muxed together depends on the particular LCD glass and how the LCD has mapped these segments to drive pins. Choosing the right LCD display makes software less complex. For example, if a display has the Sx pins mapped to segments that allow the segment to display any numerical digit using only one or two Sx pins, the code is easier to write. The mapping of pins to LCD segments is found in the LCD data sheet.

Figure 5-1 shows some information from the data sheet for the LCD glass on the LP-MSPM0L2228 Launchpad Development Kit, which has a 4-mux display. This display allows for showing alphanumeric characters in addition to simple digits. The pin mapping found in the display data sheet, shows that the segments required to display any digit 0-9 require two Sx pins – pin 1 and 2 of the LCD. To display any alphabet character, use 4 segment pins.



PIN	1	2	3	4
COM0	A1M	A1D	A1DP	A1P
COM1	A1G	A1C	A1N	A1K
COM2	A1F	A1B	NEG	A1J
COM3	A1E	A1A	A1Q	A1H

Figure 5-1. 4-Mux Display Data Sheet Example

In addition to the mapping of the LCD glass, careful selection of which Sx segment pins on the MSPM0 to connect to which LCD segment pins on the LCD glass has a big impact on layout and software.

5.1.1 Hardware-Driven Layout

In a hardware-driven layout approach, one option is for pins to be connected to the closest LCD-capable pins on the MSPM0 to minimize crossings and layout the board in a single layer. However, the hardware-driven approach in layouts where the pins mapped to the MSPM0 LCD memory are scattered through memory, meaning more software work and overhead when writing the code.

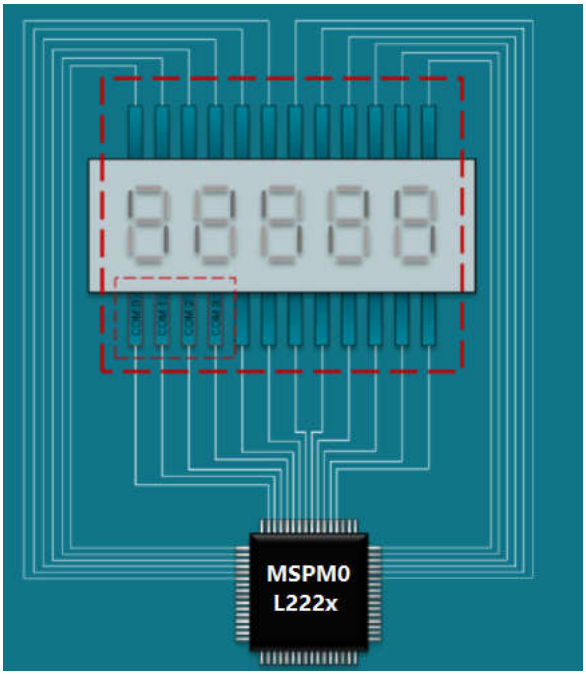
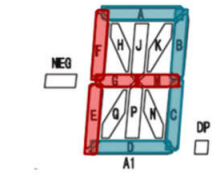


Figure 5-2. Example Layout Grouping LCD Lines Bus-Style in a Single Layer

5.1.2 Software-Driven Layout

Using a software-driven approach for pin selection allows the MSPM0 to set all required segments to display digits and characters using a single memory write of a byte or word. For example, for the 4-mux display shown previously in Figure 5-1, two pins control all the segments to display any digit 0-9, and four pins control all the segments to display any alphanumeric character. In 4-mux mode, the MSPM0 LCD memory has each byte controlling two Sx pins. Therefore, with careful connections between the MSPM0 and this LCD glass, set on a display is a full digit with a single byte access (writing two pins at once), or a full alphabet character is on the display with a half-word access (writing four pins at once). Figure 5-3 shows an example of choosing MSPM0 Sx pins within the same LCDMx memory register so that all segments for both Sx pins are set with a single byte access. It is also important to make sure that the same segments for each digit are assigned to the same ordering within the byte and are laid out in the same format in memory to confirm that the same function call is used, no matter the set digit of the display, saving greatly on software overhead.

However, depending on the application, the connections are more complex or require a multilayered board. On devices with LCD module, any pin is a COM pin or a segment pin helping to make layout easier which helps to mitigate the layout issue to an extent.



PIN	1	2	3	4
COM0	A1M	A1D	A1DP	A1P
COM1	A1G	A1C	A1N	A1K
COM2	A1F	A1B	NEG	A1J
COM3	A1E	A1A	A1Q	A1H

MSP430 Pin	LCD pin
L8	1
L9	2

Memory Row	COM3 (MSB-7)	COM2	COM1	COM0	COM3	COM2	COM1	COM0 (LSB-0)
0	L1	L1	L1	L1	L0	L0	L0	L0
1	L3	L3	L3	L3	L2	L2	L2	L2
2	L5	L5	L5	L5	L4	L4	L4	L4
3	L7	L7	L7	L7	L6	L6	L6	L6
4	L9	L9	L9	L9	L8	L8	L8	L8
5	L11	L11	L11	L11	L10	L10	L10	L10
6	L13	L13	L13	L13	L12	L12	L12	L12

Figure 5-3. Software-Driven Layout Pin Selection Example

5.1.3 General Layout Rules

LCD signal lines are constantly switching to keep the image on the display, keep them away from noise-sensitive lines (like the external crystal connections). Use guard rings to shield noise-sensitive lines, like the crystal connections or ADC inputs, from noise coupling. A ground plane underneath the LCD traces and guard traces also provide shielding.

One good practice is to keep all LCD signal traces (segment and common lines) together, similar to a data bus. Keeping the LCD layout in a single layer is helpful so that there are not LCD traces running over or under potentially sensitive traces. Keep the charge pump capacitor on the LOADCAP pin as close as possible to the MCU with a short trace.

Figure 5-4 shows a portion of the water meter design showing the LCD connections. The signals to the LCD at the top of the layout are grouped together and go around the crystal oscillators X1 and X2 rather than under to prevent noise from disturbing the crystals. The crystal circuitry also includes its own ground plane to help further shield them from noise from the LCD and other sources.

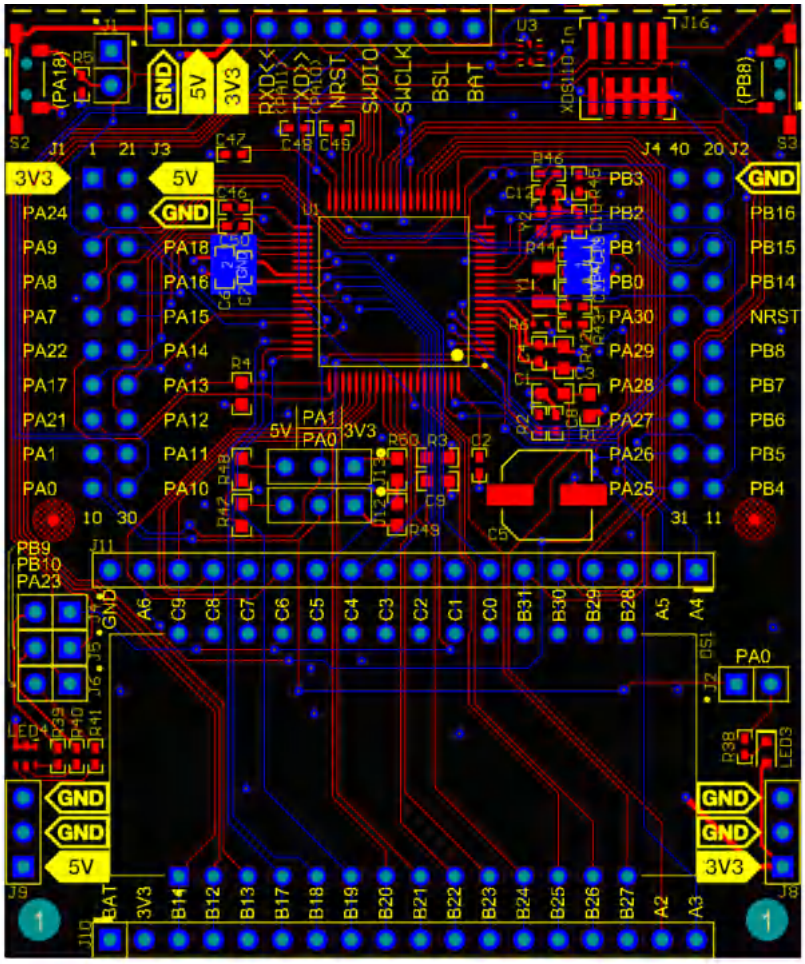


Figure 5-4. Portion of LP-MSPM0L2228 Design Showing LCD Layout

5.2 LCD Software Tips

Writing an LCD driver program seems daunting since there are so many hardware factors that the software must handle.

1. LCD display glass and its layout and properties.
2. Choice in connection between the LCD pins on the display and LCD pins on the MSPM0 MCU.
3. How the microcontroller maps the pins to display memory makes code confusing.
4. Hard to read without having the full picture and several data sheets handy.

This section offers some tips to help make LCD driver software easier to use and understand, as well as create efficient LCD code.

5.2.1 Create a Look-up Table

Creating a look-up table containing commonly displayed data, such as numbers, characters, or symbols, makes code easier to read. For example, if numbers are displayed on the LCD, create a look-up table containing the values to write into the LCD memory registers to display each digits 0-9. Using the look-up table in the code snippet below, a write to display a digit looks like: `DL_LCD_writeMemory(LCD, memIdx, displayData);`

```
//lookup table for digits on LP-MSPM0L2228 segmented LCD
const char digit[10][4] = {
    {0x07, 0x09, 0x08, 0x0A}, /* "0" LCD segments a+b+c+d+e+f+k+q */
    {0x00, 0x00, 0x00, 0x0A}, /* "1" */
    {0x03, 0x0A, 0x00, 0x0C}, /* "2" */
    {0x01, 0x0A, 0x00, 0x0E}, /* "3" */
    {0x04, 0x02, 0x00, 0x0E}, /* "4" */
    {0x05, 0x0A, 0x01, 0x00}, /* "5" */
    {0x07, 0x0A, 0x00, 0x06}, /* "6" */
    {0x00, 0x08, 0x00, 0x0A}, /* "7" */
    {0x07, 0x0A, 0x00, 0x0E}, /* "8" */
    {0x05, 0x0A, 0x00, 0x0E}  /* "9" */
};
DL_LCD_writeMemory(LCD, memIdx, displayData);
```

5.2.2 Use of #defines

Because the LCD memory on the MSPM0 MCU maps to particular MSPM0 LCD pins, which then are connected to different pins on the LCD display, the memory to pin map sometimes difficult to know which LCD memory to write to in code in order to display a particular character in a specific space on the display. An useful method is to create #defines for your LCD to reference the correct LCD memory by typing the name of the particular LCD display pins to set. For example, the entire digit for position one on the LCD data sheet is set using four Sx pins, due to the pin connections on the board. Using define gLCDPinPositionx, when the code needs to show digit on position one, software can directly write the digit in position gLCDPinPosition1 on the display and let engineer easier to write and understand the code.

```
//lookup table for digits on LP-MSPM0L2228 segmented LCD
typedef struct {
    uint32_t pin1;
    uint32_t pin2;
    uint32_t pin3;
    uint32_t pin4;
} LCD_pin;

/* Onboard LCD positions 1-6 */
LCD_pin gLCDPinPosition1;
LCD_pin gLCDPinPosition2;
LCD_pin gLCDPinPosition3;
LCD_pin gLCDPinPosition4;
LCD_pin gLCDPinPosition5;
LCD_pin gLCDPinPosition6;
```

5.2.3 Efficient Clearing of the LCD Memory

LCD modules support clearing the LCD memory using a single bit. The LCDCLRM (clears LCDM memory) or LCDCLRBM (clears LCDBM memory) bit is able to clear the entire LCD memory with a single instruction – all LCD display memory registers are cleared at the next frame boundary when LCDCLRM is set. After the memory registers are cleared, LCDCLRM is reset, so LCDCLRM is polled to see when the clear is completed. LCDCLRBM performs the same function and acts in the same way, however LCDCLRBM clears the blinking memory registers.

The clearing functionality is useful for efficiently clearing the whole screen and all memory registers. Therefore, preparation for updating the whole display as code must write only to memory registers that are used in the new display data, instead of also having to manually clear other registers for unused segments.

5.2.4 Double-buffering of the Display Buffer Using Dual Display Memory

As mentioned in Section 4.4.2, use a blink memory on LCD module as a secondary display memory when no blinking mode is selected. To select which memory (LCD memory or Blink memory) is currently being displayed simply set the LCDMEMCTL.LCDDISP bit to zero or one. The advantage of dual display memory feature performs an instant update of all LCD segments to display a new message or image on the screen. Over time, changes are made to the currently unused display memory without affecting the current display output. Once the full memory is populated with the desired data, change the display all at once by toggling the LCDMEMCTL.LCDDISP bit.

Using this methodology, use a slower MCLK (sometimes required to meet low peak current consumption), or the CPU is busy with other interrupts going on while populating this display buffer without any partial image showing up on the display.

A typical flow is:

1. Populate the display memory that is not currently being shown with desired data.
2. Toggle LCDMEMCTL.LCDDISP bit to change which memory is displayed.
3. Go to step one for next image.

Another feature that dual display memory provides is the ability for hardware to automatically toggle between the two display memories through setting the blinking mode LCDBLKCTL.LCDBLKMODx = 3. In blinking mode, the LCD toggles between the memories at the blinking frequency that are configured. Blinking mode is useful for displaying a long string of text or information that does not fit on the display – part of the string is loaded into each memory, and then blinking mode is set to automatically toggle between the two memories without software intervention.

For example, on a display that displays six alphanumeric characters, the message "Hello World" is displayed by loading the LCD memory registers to display "Hello" and loading the LCD Blink memory registers to display "World," and setting the LCDBLKCTL.LCDBLKMODx = 3, with the blink frequency configured to a slow frequency such as 1Hz to allow time for users to read the message. In this case, without any additional software intervention, the display continually shows "Hello" for 1 second and then "World" for 1 second.

6 Additional Resources

1. Texas Instruments, MSPM0 L-Series 32MHz Microcontrollers, technical reference manual
2. Texas Instruments, LP-MSPM0L2228 Launchpad
3. Texas Instruments, LP-MSPM0L2228 Software, out-of-box software
4. Texas Instruments, MSPM0-SDK, software development kit

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265