



... the world's most energy friendly microcontrollers

EFM32 LCD Driver

AN0057 - Application Note

Introduction

The EFM32 microcontrollers include a self contained, ultra low power LCD driver with internal bias voltage circuit and boost converter to minimize external components. The LCD driver also supports autonomous animation and blinking in deep sleep without software intervention. This application note provides a description of how passive segment LCD displays work and how they can be interfaced with the EFM32.

This application note includes:

- This PDF document
- Source files (zip)
 - Example C-code
 - IDE workspaces with example projects



1 Introduction to LCD Segment Displays

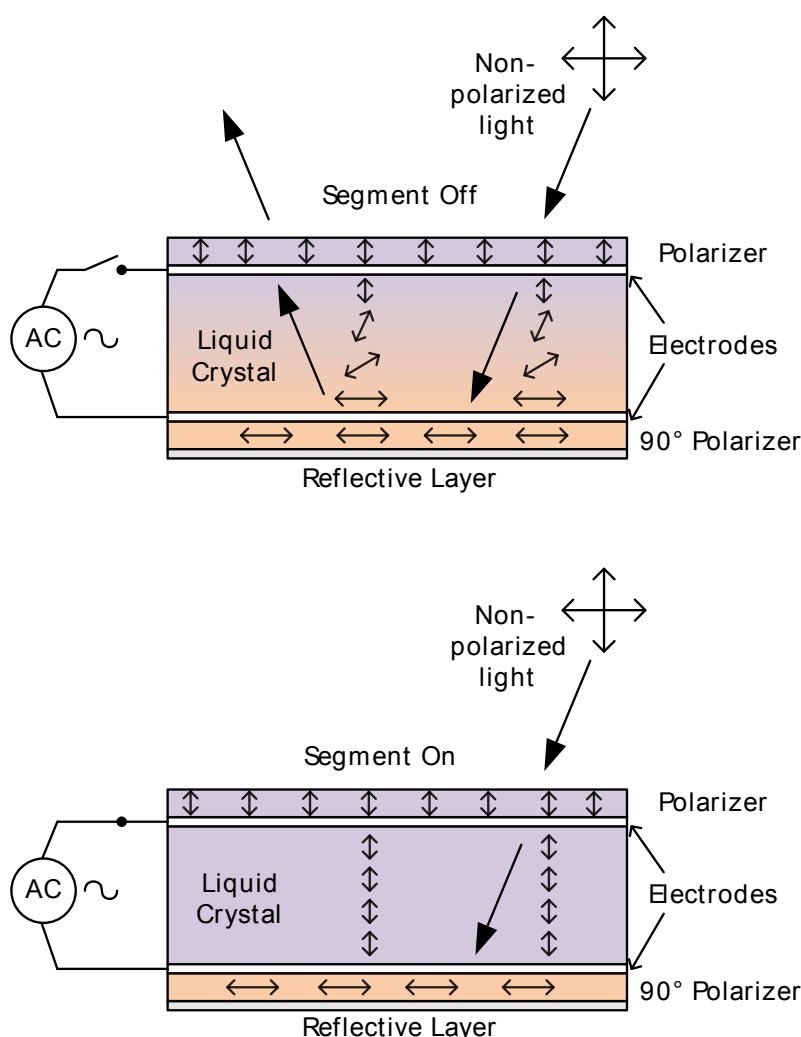
Segmented LCD displays are a common way to display information. The extreme low-power LCD driver in the EFM32 enables a lot of applications to utilize an LCD display even in energy critical systems. This document will both discuss how certain types of LCD displays work, how they are driven and how to minimize energy consumption doing so.

1.1 Passive LCD Displays

This document will only discuss passive segment LCD displays (Liquid Crystal Display). These displays are usually constructed by sandwiching the liquid crystal between two glass plates. By using the voltage dependent, polarizing properties of the liquid crystal material, light transmission through the LCD glass can be controlled. The display is usually built up by segments that can either block light or let it pass through depending on the voltage applied over the liquid crystal within that segment.

By having a reflective coating on one side of the glass, ambient light can either be reflected back to the user or blocked by the polarizer closest to the reflective layer. This blocking of light occurs because the liquid crystal in one state will change the polarization of the light to allow it to pass both polarizers. In the other state it will not affect the polarization, and the light is then blocked because the two polarizers are orthogonally oriented with respect to each other.

This document will further use the notion that a segment is "on" when it is blocking light and "off" when light can pass through. Some displays invert this notion by letting light pass through the "on" segments, and having the "off" segments block light. Figure 1.1 (p. 3) illustrates how light polarization is affected when the light passes through the polarizers and liquid crystal with and without a voltage applied over the liquid crystal.

Figure 1.1. One LCD segment with and without voltage applied.

When the voltage is applied, the liquid crystal does not change the polarization direction of the light, this causes it to be blocked by the second polarizer which is oriented at an orthogonal angle to the first polarizer. If no voltage is applied, the liquid crystal will change the polarization of the light so that it can pass through both polarizers.

1.2 Driving a Display Segment

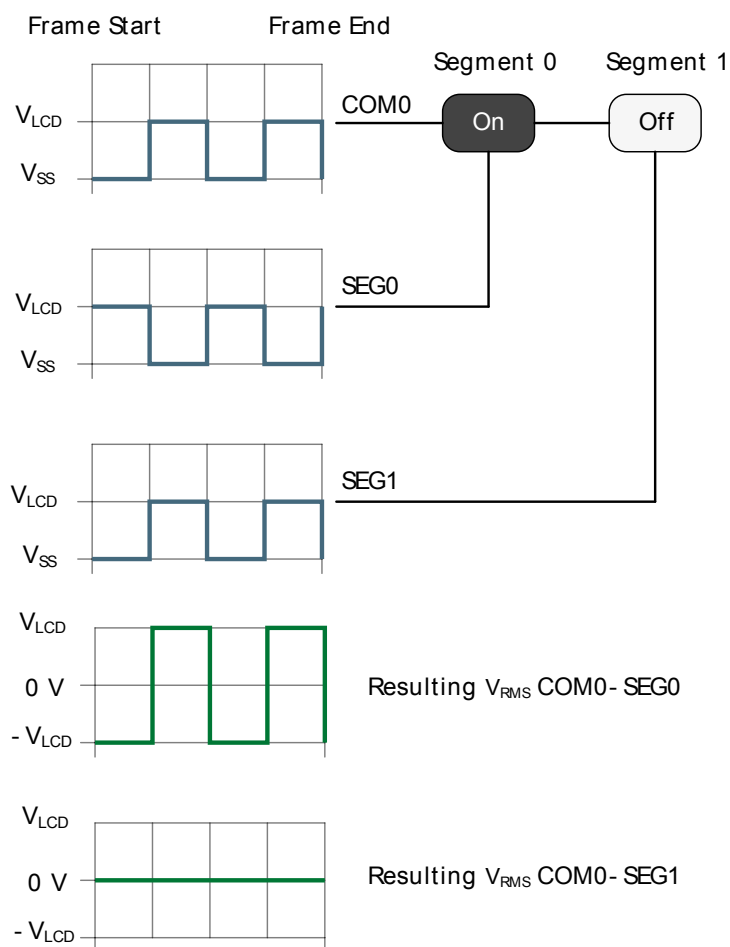
Segment LCD displays do not have any internal driving circuitry, all the display pins are directly connected to each side of the corresponding segment. The simplest LCD imaginable consist of only one segment with two electrical connections, one for each side of the segment. The segment is turned on or off by applying a voltage across it. The electric field, or voltage, between the top and bottom of the liquid crystal between the glass plates directly affects the polarization properties, see Figure 1.1 (p. 3) .

Just applying a constant voltage across the segment to turn it on would work, but not for a long time. The problem is that after a short time the LCD segment will be affected by the DC-current passing through it, mainly through electrolysis effects on the liquid crystal and electrodes. The solution is to drive the segment with a waveform that has an average 0 DC value. As long as the voltage is switched fast enough (30-100 Hz) it will mainly be the RMS-value of the voltage that affects the amount of polarization.

The simplest way of driving a "one segment" LCD with the correct zero DC bias would be to apply two identical and optionally phase shifted, square waveforms. By shifting the phase of one signal with respect

to the other by 180 degrees, the apparent RMS-voltage across the segment goes from 0 to two times the voltage of the two waveforms. See Figure 1.2 (p. 4) .

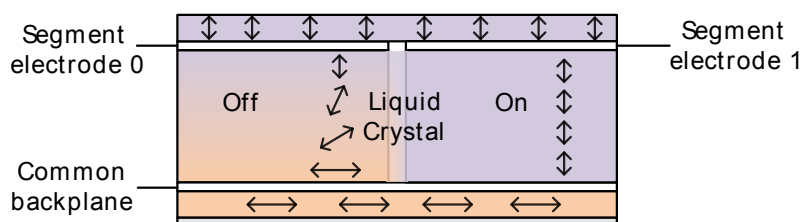
Figure 1.2. Static driving of two LCD segments, one On and one Off.



The reason for using two square waveforms, offset from 0 V is that this is simpler to achieve on a microcontroller with a single power supply. Negative voltages are rarely available, so connecting one side of the LCD segments to ground is not an option.

1.2.1 Driving Many Segments; Static Driving

An LCD display most often has more than one segment. Usually the segments are connected with one side common to many of the other segments, this is called the "common"-electrode. The other side of the segment has its own pin connection, the "segment"-electrode. See Figure 1.3 (p. 5)

Figure 1.3. Common backplane with two segments, one On and one Off.

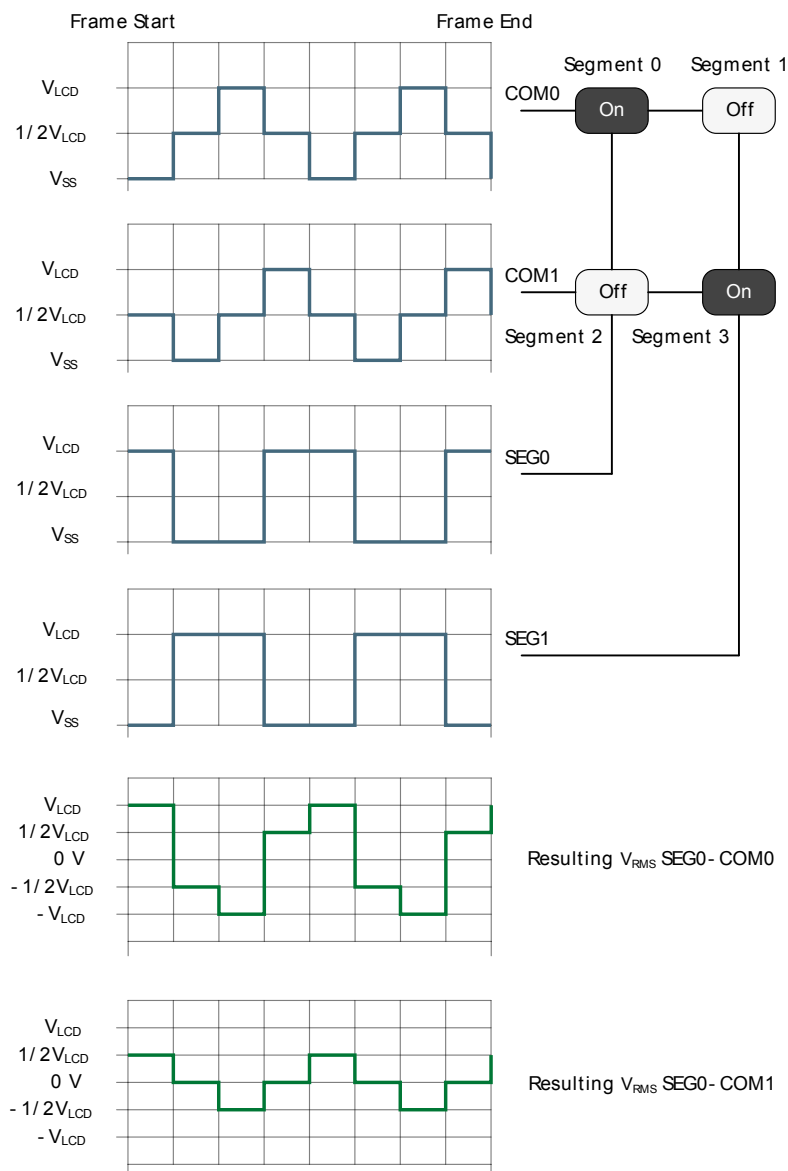
With a statically driven LCD display, there is only one common electrode and each segment has its own segment electrode. Both the common electrode and segments are driven with a square wave form. Segments that are "on" are driven with a phase shifted waveform to make the RMS voltage across these segment non-zero. See Figure 1.2 (p. 4) .

1.2.2 Driving Many Segments; Multiplexed Driving

With the static driving approach with one segment line for each segment, large displays with many segments would need a large number of the microcontroller pins just to drive the display. By multiplexing several common and segment lines, fewer pins can be used to drive more segments. The total number of segments that can be driven is the product of the number of common lines and segment lines. Usually the maximum contrast goes down, while the current consumption goes up with higher number of common lines.

It is the amplitude of the apparent RMS voltage across a segment that determines if it is on or off. A segment with a low RMS voltage amplitude applied, seems to be off even if the voltage is non-zero. The relationship between apparent RMS voltage across the segment, and the visual properties is non-linear. This non-linear property is utilized when multiplexing several segments on the same driving pins. A segment does not need to see zero RMS voltage to be perceived as completely off by the user.

Each common line and segment line is driven with waveforms consisting of more than two voltage levels as in the static driving case. The number of voltage levels are known as "bias"-levels. By carefully selecting the waveform of each segment line and common line, it is possible to make some segments "see" a low RMS-voltage, while others "see" a high RMS voltage, even if the segments share either their common or segment electrode with other segments. See Figure 1.4 (p. 6) .

Figure 1.4. Multiplexed driving of four LCD segments.

In the figure above there are two segment lines and two common lines. The bias is 1/2, which means there are 3 different voltage levels possible. The combination of the waveform on COM0 and SEG0 makes the segment between SEG0 and COM0 turn on, while the segment between SEG0 and COM1 is off. Notice the difference in RMS voltage across the two segments.

For a multiplexed display, the difference in RMS voltage between segments that are on and off will be smaller than for a statically driven display. This means that the visual quality and appearance such as contrast and viewing angles must be more carefully calibrated for a multiplexed display. The contrast can be calibrated in software by adjusting the bias-voltage levels within the LCD driver. Viewing angles and appearance of the display in different lighting conditions should be considered when selecting the actual LCD display.

2 EFM32 LCD Driver

This chapter explains some of the features and possibilities of the EFM32 built in LCD driving peripheral. For a full set of features and limitations, please see the reference manual for the family of devices you are using.

2.1 Multiplexing

The EFM32 LCD driver can multiplex up to 8 common lines, of which 4 are dedicated common lines and 4 are part of the segment lines (not available on all parts). The multiplexing waveforms and bias voltage levels are automatically handled within the LCD driver peripheral. The software only needs to set segments on or off and adjust contrast if necessary.

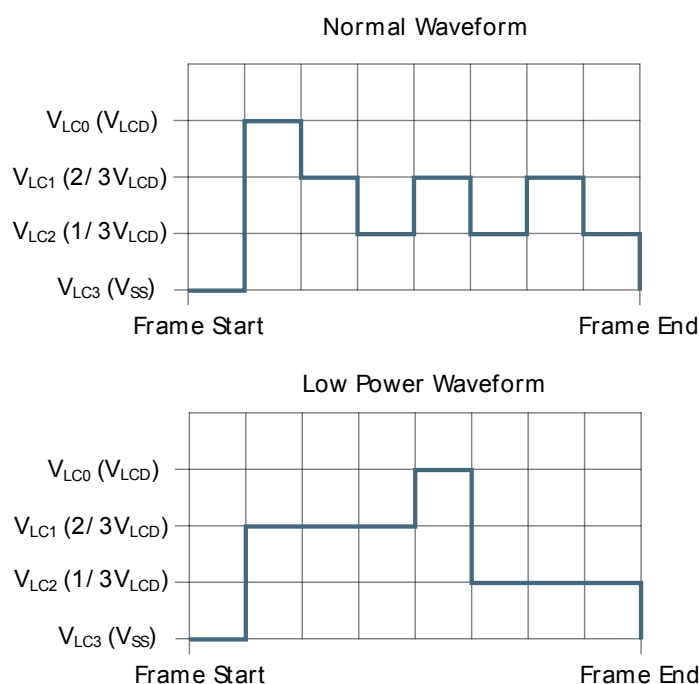
If multiplexing is not needed, static driving of the display is also possible. A statically driven display can have higher contrast than multiplexed displays. A static display with only one common line is usually lower power than multiplexed displays and in most cases current consumption go up with higher number of common lines.

2.2 Waveform and Bias Selection

When selecting multiplex-mode and number of bias-levels according to how many common lines the LCD display has, the waveforms are selected automatically by the EFM32 LCD peripheral. A normal waveform mode and a low power waveform mode is selectable, the difference is that the low power mode achieve the zero DC bias during several frames, in other words, the DC bias across LCD segments is not zero within one frame. This is ok for most LCD displays and applications, this mode is almost always recommended.

See Figure 2.1 (p. 7) for an illustration of the difference between low power and normal mode, notice that the normal mode contains more waveform changes within one frame. If flickering or contrast problems occur, the normal mode can be selected. A walk through of the different waveforms up to quad-plexed mode can be found in the reference manual.

Figure 2.1. Normal and low power waveform, only common signal shown.



Selecting the correct bias is necessary to provide the waveform generator with enough voltage levels to produce the necessary waveform for the chosen number of common lines. In general 5- to 8-plexed modes need 1/4 bias (5 voltage levels), 3- to 4-plexed modes need 1/3 bias (4 voltage levels), while 2-plexed need 1/2 bias (3 voltage levels) and static mode only need 2 voltage levels (named static mode in reference manual).

The following table illustrates the need for more bias levels when increasing the number of multiplexed common lines. The difference between V_{on} and V_{off} is essentially a measurement of the possible contrast. Notice that for the static driven display, the contrast is essentially infinite. Take a look at Figure 2.2 (p. 9) for an illustration of the non linear properties of the liquid crystal and why the difference between V_{on} and V_{off} is a measure of contrast.

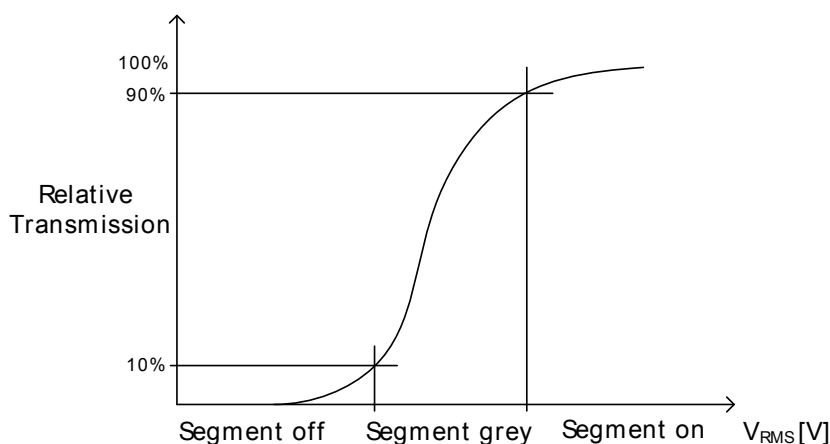
Table 2.1. LCD drive modes, bias setting and contrast characteristics.

LCD drive mode configuration	LCD bias setting	Contrast = $V_{on(RMS)}/V_{off(RMS)}$
Static	Static	Infinite
1:2 multiplex	1/2	2.24
1:2 multiplex	1/3	2.24
1:2 multiplex	1/4	1.85
1:4 multiplex	1/2	1.53
1:4 multiplex	1/3	1.73
1:4 multiplex	1/4	1.65
1:6 multiplex	1/2	1.34
1:6 multiplex	1/3	1.53
1:6 multiplex	1/4	1.53
1:8 multiplex	1/2	1.25
1:8 multiplex	1/3	1.41
1:8 multiplex	1/4	1.45

Notice that for some of the multiplex configurations, the contrast actually goes down with higher number of bias levels. In general higher number of bias levels also means higher energy consumption, so for a given number of multiplexed common lines, the lowest possible number of bias levels should be chosen. The contrast might be acceptable even with a lower number of bias levels.

2.3 Contrast

Adjusting the contrast of the LCD is only relevant to multiplexed displays because the "off" segments in multiplexed mode still has a non-zero RMS voltage value across them. The idea of contrast adjustment is to get the RMS voltage of "off"-segments below a certain threshold so these segments are perceived as off, but still have high enough voltage across the "on"-segments for them to be perceived as on. The contrast properties varies between different LCD displays, and also changes with temperature and voltage, which makes contrast adjustments necessary. See Figure 2.2 (p. 9) for an illustration of the non-linear nature of the liquid crystal material with respect to applied voltage and transmission of light without changing the polarization.

Figure 2.2. Relative transmission of light through the LCD for changes in the applied RMS voltage.

2.3.1 Internal Voltage Boost

Some LCD displays require higher voltage waveforms to work. The EFM32 LCD peripheral includes a built in boost converter to generate voltage higher than the supply voltage of the EFM32. This boost generator can generate voltages up to 3.0 V to 3.6 V from supply voltages down to the minimum EFM32 V_{dd} voltage.

The usual sign that the boost converter is needed is that the LCD contrast goes down and the "on"-segments start to appear dim even with the contrast adjusted to its highest level. This can also happen because of reduced temperature of the LCD glass. When this happens, the LCD boost converter can be turned on by software, it is not necessary to have the boost converter enabled all the time, it can be enabled on the fly when needed. Remember that the boost converter significantly adds to the current consumption (several microamps typically).

Please see the reference manual for more information on the necessary external components for the boost converter and how to enable and adjust its output voltage.

2.3.2 Contrast Compensation

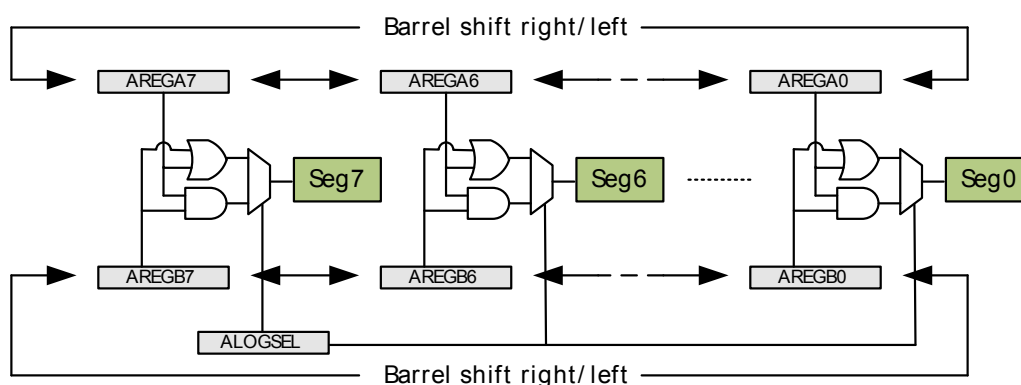
As mentioned earlier the LCD contrast is affected by both the EFM32 supply voltage and the temperature of the LCD glass. If the application is operated over a wide range of supply voltages and temperatures, the contrast, and optionally the boost converter, should be automatically adjusted according to predetermined thresholds.

The correct contrast adjustment varies between different displays and viewing angles. Exactly at which voltages and temperatures the boost converter is needed is also dependent on the LCD display itself. Some information can often be found in the LCD glass documentation.

2.4 Animation and Blinking

The EFM32 LCD driver includes special features to enable animation and blinking of specific segments without any software intervention. This is useful for displaying continuous animation to signal that a device is alive, without the need to wake up from deep sleep to update the segments.

The animation feature is available on segment 0 to 7 multiplexed with LCD_COM0. The animation is implemented as two programmable 8 bits registers that are shifted left or right every other Animation state for a total of 16 states. The animation state changes with each frame counter event. Please see the reference manual for more information on the animation feature and its interface. The animation registers and shift operation can be seen in Figure 2.3 (p. 10) .

Figure 2.3. Animation Function with barrel shift and logic operations

The included software example demonstrates how to enable and configure the blinking and animation features.

2.5 LCD Interrupts

The LCD peripheral contains one interrupt source, the frame counter interrupt. It can be used to time and execute display updates, animation changes or similar display effects. Since there are no limitations on when the LCD segment registers can be updated, there is no need to use the frame counter interrupt to time segment updates. The driver supplied with the kits and which is used in the software example with this application note does not use this interrupt. Please see the reference manual for more information.

2.6 Minimize Energy Consumption

The LCD driver is in itself a very low power peripheral that can operate in deep sleep mode (EM2) from the low frequency oscillator. When connecting an LCD display to the EFM32, the energy consumption will increase because of the capacitive load the LCD display represents. To minimize the energy consumption it is necessary to optimize a few parameters.

After the multiplex and bias settings are configured correctly, and low power waveform is selected, the display refresh rate should be configured to the lowest possible value that does not cause flickering. The refresh rate is configured by setting the frame counter top and prescale values.

The LCD boost converter should not be used unless absolutely needed. The contrast should be adjusted first to counteract falling supply voltage or lower temperatures. Remember that the boost converter significantly adds to the current consumption (several microamps typically).

3 Software Examples

This application note includes a software examples that demonstrate use of the LCD peripheral. The example is a simple "Hello World" example that exemplifies how to use the LCD display with display drivers for the EFM32 starter kits and development kits. It also implements more advanced features such as animation and the voltage boost feature.

3.1 Hello World

This example uses the BSP (Board Support Package) drivers to enable and write a message on the LCD display. It configures the LCD peripheral correctly for low current consumption without flickering on the different EFM32 starter kits. In addition it enables the animation feature of the EFM32 LCD driver to animate a rotating circle in deep sleep. Note that the rotating circle is only half a circle on some kits because not all of the circle segments are connected to the animation enabled segment and common lines.

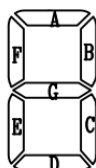
3.2 Porting the LCD Driver to other Displays

This section contains a description of the necessary steps needed to port the BSP LCD driver to other LCD displays with different segment and common line mapping. It will only describe how to port either 4-plexed or 8-plexed displays, since these are the only types of displays present on Energy Micro kits. It should however be fairly easy to port the driver to other multiplexed configurations after understanding the basics of how to port the display driver.

A segment LCD most often consist of up to three types of "information displaying modules":

- 1: Individual symbol segments that can be turned on/off, it can be anything from a low battery symbol to Celcius/Farhenheit temperature indication. The EFM32 kit LCD glass contains several symbols, most of them are located in the top left corner, including the animation ring, battery symbol, energy mode rings and the gecko.
- 2: 7 Segment numbers, each digit is built up of 7 segments that can be turned on or off to make up the numbers from 0 to 9. The EFM32 kit LCD glass contains 4 of these in the top right corner.

Figure 3.1. 7 segment digit with typical naming of segments.



- 3: 14 Segment alphanumeric letters, sometimes referred to as "star burst" or "union jack" display. Each letter or digit is built up of 14 segments that can be turned on or off to make up either any number from 0 to 9 or any letter from a to z, even capital letters are supported. The EFM32 kit LCD glass contains 7 of these in the lower part of the display.

Note that the driver only supports mapping for 14-segment star burst letters, some LCD's contain 15 or 16 segment letters, to support these, the mapping of letters to segments must also be changed in addition to the segment to pin mapping.

Figure 3.2. 14 segment alphanumeric letter with typical naming of segments.



Figure 3.3. EFM32 kit LCD glass segments.

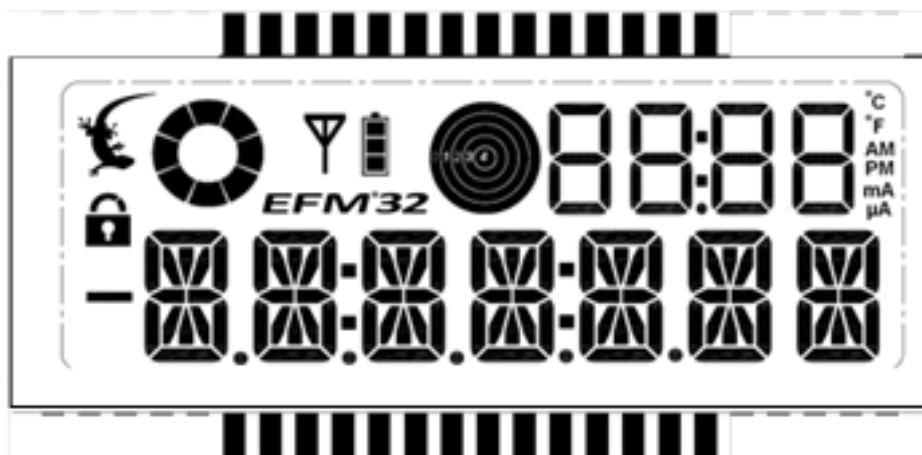


Figure 3.3 (p. 12) illustrates how the Energy Micro kit LCD glass is built up. Notice the three types of segment constellations, symbols, 7-segment digits and 14-segment alphanumerics. The three types of constellations are handled separately in the board support package LCD driver in the example projects.

3.2.1 Understanding the LCD Segment Map

All the mapping information on which segment and common line is connected to which LCD segment is contained in the *segmentlcdconfig.h* file which varies between the different starter kits and development kits. It is contained in the "config" folder within each kit software example folder.

To understand how the *segmentlcdconfig.h* is built up, it helps to also look at the LCD display mapping figure which is located in the kit schematic document and at the same time have the *segmentlcdconfig.h* file open. For better understanding of this section, the reader of this document should open it and have it open while reading the rest of this document.

As an example, consider the first alphanumeric letter in the "EFM_DISPLAY_DEF"-define in *segmentlcdconfig.h* for the STK3700. ".com[0]" and ".bit[0]" refers to the common and segment line of the first segment (denoted A in the 14-segment letter in Figure 3.2 (p. 12)). ".com[1]" and ".bit[1]" refers to the second segment (denoted B in the 14-segment letter in Figure 3.2 (p. 12)).

Note that in the case of the STK3700, the common line sequence is reversed and so is the numbering of the alphanumeric letters compared to the code. This causes the first alphanumeric letter in the *segmentlcdconfig.h* file to actually be the 7 alphanumeric letter in the LCD figure. Further the ".com[0] = 1" mapping is actually mapping to common line 6 in the LCD glass figure (Figure 3.4 (p. 14)).

Since the pin numbering of the LCD glass in the schematic begins at 1, the ".bit[0] = 13" mapping in *segmentlcdconfig.h* actually refers to PIN 14 in the LCD figure in the schematic. All of this information should make it clear that the ".com[0] = 1" and ".bit[0] = 13" results in mapping the segment named 7A to the first segment in the first alphanumeric letter in *segmentlcdconfig.h*.

3.2.2 Porting the LCD Segment Map

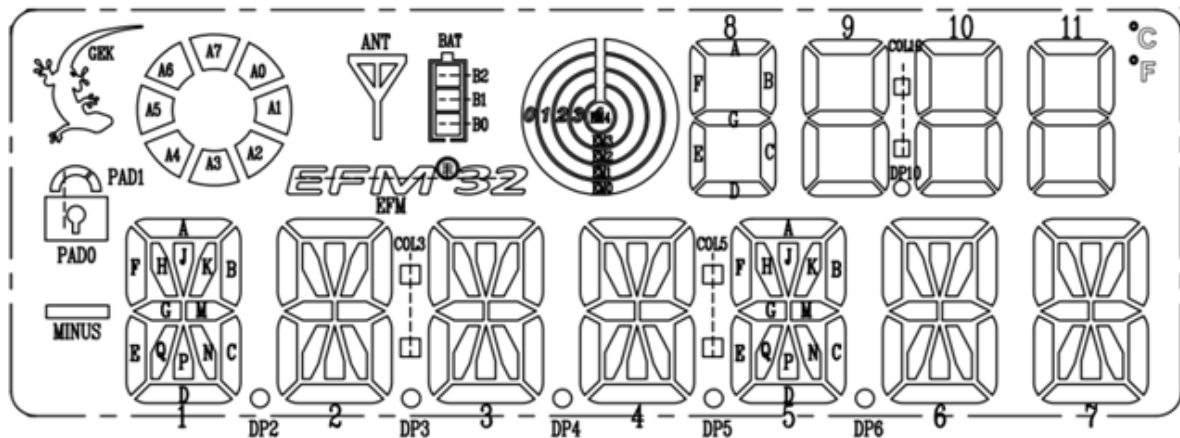
This section is an actual step by step description of the porting. Start with selecting a kit with the same number of multiplexed common lines as your display. Either select the g8xx starter kit for 4-plexed displays, or the STK3300/STK3700 for 8-plexed displays.

Open the *segmentlcdconfig.h* file for the selected kit. Start with the easiest part, the individual symbol segments, if any. These have their individual segment and common line defines at the start of header file. Change a couple of them and check if it is correct by turning on only those segments. Watch the LCD glass and check if your mapping was correct. This should indicate if you have connected the common lines/segment lines in the order you expect (it might be inverted or completely the wrong order because of errors in lcd documentation or PCB schematic/layout).

Next on the list is the 7-segment digits if any. These are a bit easier than the 14-segment alphanumerics because they contain fewer segments per digit. Remember that ".com[0]" and ".bit[0]" refers the common and segment line of the first segment (segment A) in the digit, ".com[1]" and ".bit[1]" is segment B and so on. These are most often named A-G in alphabetic order, clockwise around the 7-segment digit, starting at the top, see Figure 3.1 (p. 11) .

If your display contains star burst letters (14-segment letters) these can be ported in the same way as the 7-segment digits. The numbering is also here usually clockwise, starting from the top, note that the actual letters chosen by the LCD glass manufacturer to represent which segments can vary quite a bit. In any case the LCD glass of the efm32 kits combined with the *segmentlcdconfig.h* supplied for this LCD glass should make it possible to figure out the correct numbering of the segments. See Figure 3.4 (p. 14) for the mapping and naming of all the segments in the 8-plexed efm32 LCD glass.

Figure 3.4. EFM32 LCD segment to pin assignment.



PIN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
—	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
COM0	DP2	1 E	1 D	2 E	2 D	3 E	3 D	4 E	4 D	DP5	5 D	DP6	6 D	7 E
COM1	DP4	1 Q	1 N	2 Q	2 N	3 Q	3 N	4 Q	4 N	5 E	5 N	6 E	6 N	7 Q
COM2	DP3	1 P	1 C	2 P	2 C	3 P	3 C	4 P	4 C	5 Q	5 C	6 Q	6 C	7 P
COM3	COL3	1 G	1 M	2 G	2 M	3 G	3 M	4 G	4 M	5 P	5 M	6 P	6 M	7 G
COM4	MINUS	1 F	1 J	2 F	2 J	3 F	3 J	4 F	4 J	5 G	5 J	6 G	6 J	7 F
COM5	PAD1	1 H	1 K	2 H	2 K	3 H	3 K	4 H	4 K	5 F	5 K	6 F	6 K	7 H
COM6	GEK	1 A	1 B	2 A	2 B	3 A	3 B	4 A	4 B	5 H	5 B	6 H	6 B	7 A
COM7	A7	A6	A5	A4	A3	A2	A1	A0	EFM	5 A	COL5	6 A	ANT	BAT

PIN	15	16	17	18	19	20	21	22	23	24	25	26	27	28
—	S14	S15	S16	S17	S18	S19	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
COM0	7 D	11 A	10 A	9 A	8 A	EM2								COM0
COM1	7 N	11 F	10 F	9 F	8 F	EM4							COM1	
COM2	7 C	11 B	10 B	9 B	8 B	COL10						COM2		
COM3	7 M	11 G	10 G	9 G	8 G	DP10					COM3			
COM4	7 J	11 E	10 E	9 E	8 E	PAD0				COM4				
COM5	7 K	11 C	10 C	9 C	8 C	EM3			COM5					
COM6	7 B	11 D	10 D	9 D	8 D	EM1		COM6						
COM7	*C	*F	B1	B0	B2	EM0	COM7							

4 Revision History

4.1 Revision 1.03

2013-09-03

New cover layout

4.2 Revision 1.02

2013-05-08

Added software projects for ARM-GCC and Atollic TrueStudio.

4.3 Revision 1.01

2013-02-07

Fixed linker error for DK3550 project.

4.4 Revision 1.00

2013-02-06

Initial revision.

A Disclaimer and Trademarks

A.1 Disclaimer

Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

A.2 Trademark Information

Silicon Laboratories Inc., Silicon Laboratories, the Silicon Labs logo, Energy Micro, EFM, EFM32, EFR, logo and combinations thereof, and others are the registered trademarks or trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

B Contact Information

Silicon Laboratories Inc.

400 West Cesar Chavez

Austin, TX 78701

Please visit the Silicon Labs Technical Support web page:

<http://www.silabs.com/support/pages/contacttechnicalsupport.aspx>

and register to submit a technical support request.

Table of Contents

1. Introduction to LCD Segment Displays	2
1.1. Passive LCD Displays	2
1.2. Driving a Display Segment	3
2. EFM32 LCD Driver	7
2.1. Multiplexing	7
2.2. Waveform and Bias Selection	7
2.3. Contrast	8
2.4. Animation and Blinking	9
2.5. LCD Interrupts	10
2.6. Minimize Energy Consumption	10
3. Software Examples	11
3.1. Hello World	11
3.2. Porting the LCD Driver to other Displays	11
4. Revision History	15
4.1. Revision 1.03	15
4.2. Revision 1.02	15
4.3. Revision 1.01	15
4.4. Revision 1.00	15
A. Disclaimer and Trademarks	16
A.1. Disclaimer	16
A.2. Trademark Information	16
B. Contact Information	17
B.1.	17

List of Figures

1.1. One LCD segment with and without voltage applied.	3
1.2. Static driving of two LCD segments, one On and one Off.	4
1.3. Common backplane with two segments, one On and one Off.	5
1.4. Multiplexed driving of four LCD segments.	6
2.1. Normal and low power waveform, only common signal shown.	7
2.2. Relative transmission of light through the LCD for changes in the applied RMS voltage.	9
2.3. Animation Function with barrel shift and logic operations	10
3.1. 7 segment digit with typical naming of segments.	11
3.2. 14 segment alphanumeric letter with typical naming of segments.	12
3.3. EFM32 kit LCD glass segments.	12
3.4. EFM32 LCD segment to pin assignment.	14

List of Tables

2.1. LCD drive modes, bias setting and contrast characteristics. 8

silabs.com

