

XGATE Library: TN/STN LCD Driver

Driving Bare TN and STN LCDs Using GPIO Pins

by: Daniel Malik
MCD Applications, East Kilbride

1 Introduction

The XGATE LCD driver is a collection of header and source files, in the C programming language, that enable driving of bare twisted nematic (TN) and super twisted nematic (STN) liquid crystal displays (LCDs) with multiple backplane electrodes. The XGATE co-processor is used to generate all the waveforms required to drive the display. The driver is optimized for low-cost display driving solutions; it uses only GPIO pins and a few external resistors to generate the required waveforms.

2 TN LCDs

2.1 History

Properties of liquid crystal materials were first discovered in 1888 by an Austrian botanist, Friedrich Renitser. However, these materials were not suitable for commercial use. The first materials that retained their nematic properties even at room temperatures were developed in the 1960s, but the real breakthrough came

Table of Contents

| | | |
|-----|---|----|
| 1 | Introduction | 1 |
| 2 | TN LCDs | 1 |
| 2.1 | History | 1 |
| 2.2 | Construction | 2 |
| 2.3 | Principle of Operation | 2 |
| 2.4 | Organization of Display Segments | 4 |
| 2.5 | Voltage Waveforms for Driving LCDs | 5 |
| 2.6 | Voltages on LCD Cells Created by Different Drive Waveforms. | 9 |
| 2.7 | Selecting the Right Waveforms | 11 |
| 2.8 | Contrast Control. | 12 |
| 3 | Driving TN and STN LCDs Using General Purpose Microcontroller Pins | 15 |
| 3.1 | Selection of Waveforms for the GPIO Drive | 15 |
| 3.2 | Suitability of 1/2 Bias Ratio Waveforms. | 16 |
| 3.3 | Generating 1/2 Bias Ratio Waveforms | 16 |
| 4 | Driver Implementation Using XGATE | 17 |
| 4.1 | Structure of the Driver | 17 |
| 4.2 | Configuring the Driver | 18 |
| 4.3 | Using the Driver. | 21 |
| 4.4 | Driver Performance | 22 |
| 5 | References | 25 |

when the cyanobiphenyl materials were discovered. Properties of these materials make them almost ideal for the twist cell construction. Martin Schadt and Wolfgang Helfrich published their description of the twisted nematic cell in 1971 and triggered a revolution.

2.2 Construction

The most common LCD is called the twisted nematic (TN) display. This display consists of a nematic liquid crystal sandwiched between two glass plates. The inner glass surfaces are coated with a transparent metal oxide film, which acts as an electrode, and are used to apply voltages across the cells. A polymer alignment layer is placed on top of the electrodes, and microscopic grooves, which are created in it during the manufacturing process, force the crystal molecules to align in parallel with the grooves. A layer of polymer spacer beads is eventually applied to one of the glass plates to maintain a uniform gap between the two, the plates are placed together, and the edge is sealed with epoxy. A small hole is left in the epoxy and the liquid crystals are injected between the plates through this hole under a vacuum. After the hole is sealed, polarizers are applied to the top and bottom surfaces of the display. In TN displays, the polarizers and alignment layers are orientated perpendicular to each other.

2.3 Principle of Operation

The perpendicular orientation of the alignment grooves forces the crystal molecules to twist by 90 degrees under no-voltage conditions — see the left side of [Figure 1](#). The light entering the display from the top is polarized by the top polarizer and, as it travels through the liquid crystals, its polarization rotates with the molecules. When it emerges, its polarization has been changed by 90 degrees and it passes through the

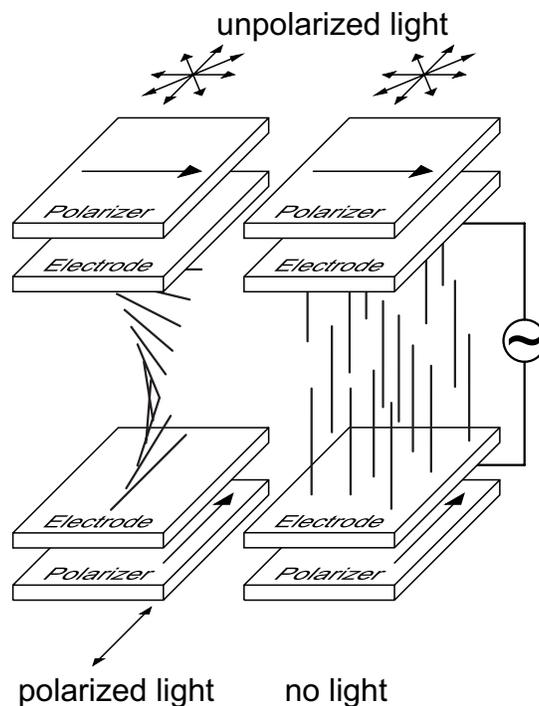


Figure 1. Principle of TN LCD Operation

bottom polarizer. However, when a voltage is applied to the electrodes, the majority of the molecules will arrange vertically — see the right side of [Figure 1](#). In this case, polarization of the light passing through the liquid crystals remains unchanged and the light beam is blocked by the bottom polarizer.

The voltage waveforms applied to the LCD electrodes must have no DC component. If DC voltage exists across the LCD cell, any impurity ions present will migrate towards the electrodes, under the action of the electric field, and may become embedded at the cell surfaces. Therefore, on removal of the stimulating voltage, an electric field across the cell may persist, due to the captured charges, and may hinder cell switching.

It may be expected that the AC field will cause the molecules to change their orientation with each polarity reversal. However, when a voltage is applied across the cell, the molecules are oriented by the applied electric field and interact with each other so that adjacent molecular dipoles are aligned anti-parallel. An individual molecule is therefore screened or shielded by the surrounding molecular dipoles; it is not directly subjected to the externally applied voltage, but only to the electric field created by the local environment. Therefore, polarity reversal of the driving waveforms will have no effect on the local environment of the molecules, and the performance of the device is dependent on the RMS voltage and not on the polarity of the external field.

During polarity reversal, the liquid crystal cell must first be discharged before being charged up again in the opposite sense, resulting in the removal of the voltage across the liquid crystal material for a short period of time. This enables partial molecular relaxation back to the helical stacking structure to occur, prior to the cell becoming fully charged again. The molecules are able to rock to-and-fro, resulting in oscillation of the overall optical transmittance of the device.

The human eye takes time to adapt to new illumination levels. If the intensity changes are faster than the adaptation period, the eye acts as an integrator perceiving the time-averaged luminance and preventing cell flickering from being observed. Since many artificial lamps operate at 50Hz/60Hz, it may be necessary to avoid these frequencies to prevent stroboscopic effects from occurring.

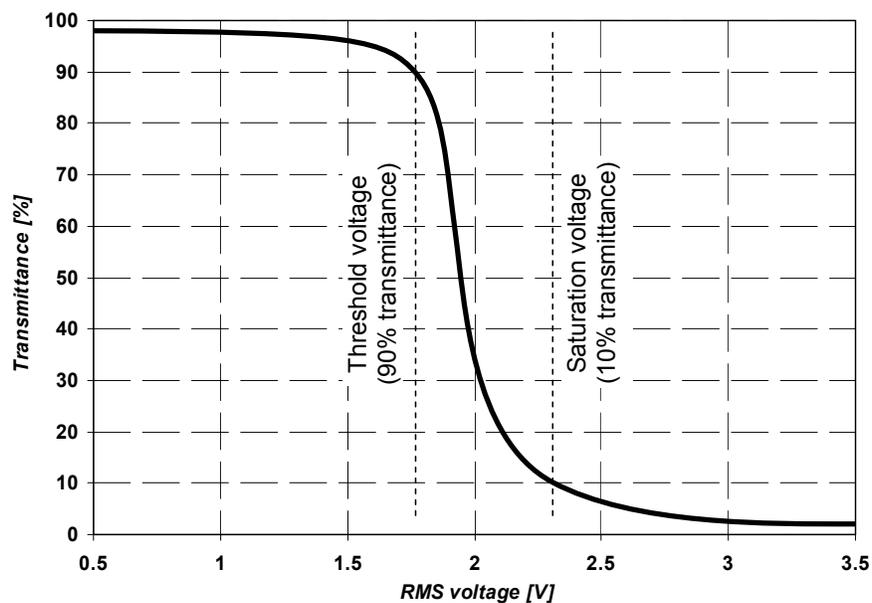


Figure 2. Example of TN LCD Cell Transmittance Characteristics

An example of the TN LCD cell optical characteristics can be seen in [Figure 2](#). The goal of the drive scheme is to keep the RMS voltage on invisible segments (V_{OFFRMS}) below the threshold voltage (V_T) and the RMS voltage on the visible segments (V_{ONRMS}) above the saturation voltage (V_S).

2.4 Organization of Display Segments

There are two basic types of TN LCDs, as far as the principle of their drive scheme is concerned: static and dynamic. Static LCDs have only one backplane electrode and one frontplane terminal for each segment they can display. Therefore, a 7-segment static display capable of displaying one digit will have eight terminals in total (see [Figure 3](#)).

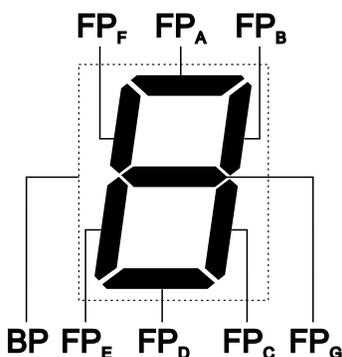


Figure 3. Example of Statically Driven LCD

Since the static displays have one terminal for each segment, the number of terminals rises very quickly when we try to add more digits on the same display. The higher number of terminals will increase the cost of both the LCD and the application where it is used. In some cases, there will not be enough space around the LCD glass edges for all the terminals.

The statically driven LCDs will always have better optical properties than their dynamically driven counterparts; however, the high number of terminals makes them economically unfavorable where a large number of segments is needed.

The solution to the problem of a high number of terminals is to organize the segments of the display into a matrix. Organization of such a matrix is very similar to the way keyboards with large numbers of keys are organized (in fact, the underlying reason for creating the matrix organization is exactly the same in both

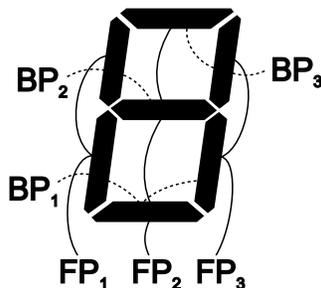


Figure 4. Example of Dynamically Driven LCD

cases). An example of a dynamically driven LCD with three frontplane and three backplane electrodes is shown in [Figure 4](#). The matrix organization of the LCD means that there are multiple segments connected to each pin. This complicates the driving scheme as we will see in [Section 2.5.2](#).

2.5 Voltage Waveforms for Driving LCDs

2.5.1 Statically Driven LCDs

There is nothing static about the waveforms required to drive the LCD, even though this type is called “statically driven”. This is because the LCD requires a pure AC voltage waveform to be applied to the electrodes to prevent damage caused by the impurity ion migration. Waveforms needed to drive this type of display are shown in [Figure 5](#).

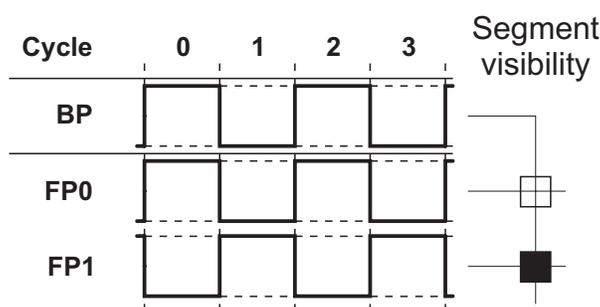


Figure 5. Waveforms for Statically Driven LCDs

When the same AC waveform is connected to both the backplane electrode and a frontplane electrode, the particular segment will be invisible, as there is no voltage across the cell. The frontplane electrodes connected to segments that should be visible receive the waveform shifted by 180 degrees.

The optimal frequency of the driving waveforms are usually specified by the LCD manufacturer. The LCD represents a capacitive load and, therefore, the current flowing through the cell rises with the waveform frequency. Therefore, the goal is to drive the display at the lowest possible frequency without affecting its optical parameters and long term reliability. The minimum frequency is around 30 Hz for most LCDs.

2.5.2 Dynamically Driven LCDs

The fact that the segments are organized into a matrix with multiple frontplane electrodes connected to a single pin makes the situation more complicated. For example, when displaying the digit “4” on the LCD shown in [Figure 4](#), one of the FP_1 segments must be visible and the other one must be invisible. Therefore, it is necessary to create a voltage greater than V_S between FP_1 and BP_2 , while keeping the voltage between FP_1 and BP_1 below V_T .

How the waveforms must look to achieve this is far from obvious. However, it is quite clear that the waveforms must have more than two discrete voltage levels. All manufacturers of LCDs and LCD drivers use the same or very similar terms to describe the properties of the waveforms. We will discuss these terms first.

The process of refreshing an LCD with n backplanes and m frontplanes is similar to the aforementioned keyboard operation. The driver will select one particular backplane BP_x (corresponding to a column on a keyboard) and drive the appropriate voltage levels to all frontplanes FP_1-FP_m (corresponding to keyboard rows). The remaining backplanes ($BP_1-BP_{x-1}, BP_{x+1}-BP_n$) are driven to a non-selected voltage level. This process is then repeated for all backplane electrodes of the display. Therefore, the full refresh cycle of the display will have n phases, one for each backplane electrode. The cells corresponding to the visible segments are driven to the “active” voltage level only in one (selected) phase of the refresh cycle and to an “inactive” voltage level in the remaining (non-selected) phases. This is why the number of phases of the refresh cycle (which equals to the number of backplane electrodes) is typically expressed as a **duty** ratio $1/n$.

The requirement that the LCD cells must be driven with a pure AC voltage is resolved by inverting the voltages in every other refresh cycle.

The voltage across an LCD cell during one of the inactive phases depends on how many discrete voltage levels the display driver uses. The voltage across the LCD cell during the inactive phases is usually expressed as a fraction of the supply voltage of the driving device and is called a **bias** (b). The number of discrete voltage levels used by the driver is normally equal to $b+1$. LCD manufacturers typically use the reciprocal of b and call it the bias ratio ($1/b$).

In this terminology, the waveforms in Figure 5 have a 1/1 duty ratio (because there is only one phase in every refresh cycle) and a 1/1 bias ratio (as there are two voltage levels used). Specifying the bias ratio does not make much practical sense, as there are no inactive phases in the refresh cycle. Note that the cell voltages are reversed in the odd and even refresh cycles.

The duty and bias parameters are properties of the driving waveforms. Each type of LCD display has its own optimal waveform parameters associated with it, but this does not mean that waveforms with different parameters would fail to work. The subject of the suitability of different waveforms for driving different types of LCDs is discussed in Section 2.7.

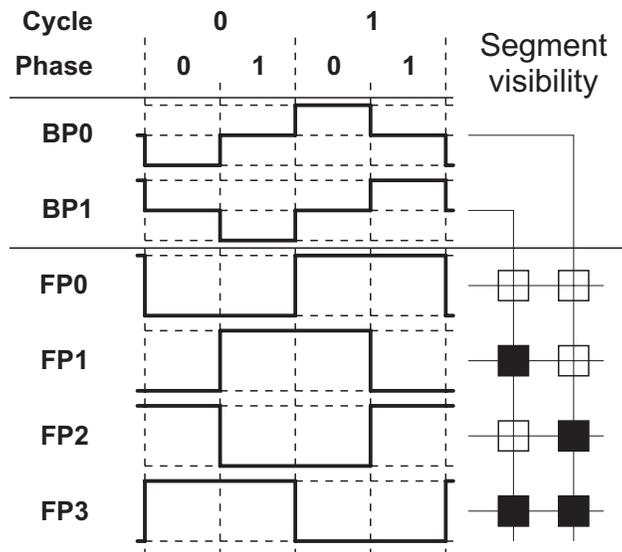


Figure 6. Waveforms for Dynamically Driven LCDs with 1/2 Duty and 1/2 Bias Ratios

It is possible to derive multiple different driving waveforms, which are equally good at driving the display. The waveforms presented in this application note represent a few of the many examples.

Figure 6 shows waveforms with 1/2 duty and 1/2 bias ratios. There are two backplane electrodes and each refresh cycle has two phases. The backplane electrode BP₀ has a “selected phase” in phase 0 and a “non-selected phase” in phase 1 (i.e. the duty ratio of 1/2). The voltage across the BP₀-FP_x cells in the non-selected phase 1 is equal to 1/2 of the supply voltage; hence, the bias ratio of 1/2.

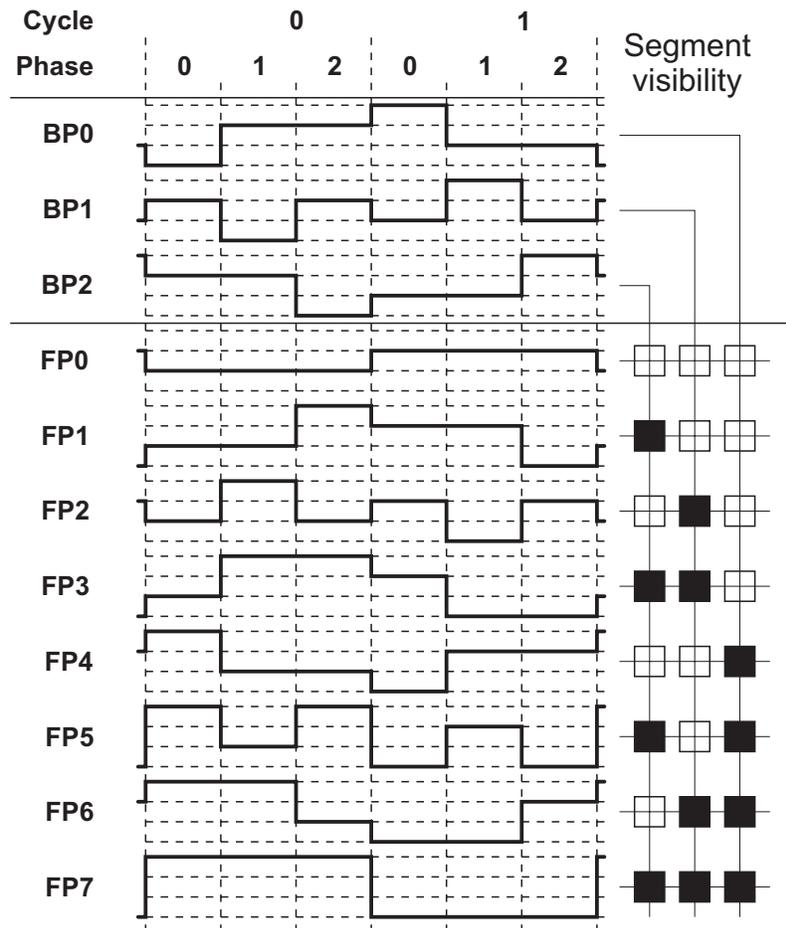


Figure 7. Waveforms for Dynamically Driven LCDs with 1/3 Duty and 1/3 Bias Ratios

Waveforms with 1/3 duty and 1/3 bias ratios are shown in Figure 7. The waveforms are now getting more complex as more voltage levels and phases are added. Note that the voltages across the LCD cells in the non-selected phases are now 1/3 of the supply voltage. When the waveforms in Figure 6 and Figure 7 are compared in deeper detail, some similarities start to emerge.

A simple algorithm exists for creating waveforms with any duty and bias ratios. The algorithm is outlined in a pseudo-C programming language in Listing 1. The voltage levels used by the algorithm range from 0 (corresponding to the lowest voltage) up to b (corresponding to the highest voltage). The interesting property of this algorithm is that the waveforms use only six voltage levels even for bias factors above 5.

The algorithm does not create any link between the duty factor n and the bias factor b . This means that it is possible to create waveforms with 1/4 duty ratio and only 1/2 bias ratio. On the other hand, it is also possible to drive an LCD with only two backplane electrodes (1/2 duty ratio) with waveforms that make use of many voltage levels (e.g. 1/5 bias ratio). The advantages and disadvantages of such driving schemes are discussed in [Section 2.7](#).

```

/* Even refresh cycle */
  for (phase=0;phase<n;phase++) {
    Drive electrode BPphase to voltage level 0
    Drive all other BP electrodes to voltage level b-1
    Drive FP electrodes of visible segments to voltage level b
    Drive FP electrodes of invisible segments to voltage level b-2
  }

/* Odd refresh cycle */
  for (phase=0;phase<n;phase++) {
    Drive electrode BPphase to voltage level b
    Drive all other BP electrodes to voltage level 1
    Drive FP electrodes of visible segments to voltage level 0
    Drive FP electrodes of invisible segments to voltage level 2
  }

```

Listing 1. Algorithm for Creating Waveforms with Arbitrary Duty and Bias Ratios

Waveforms “Type A” and “Type B”

The waveforms shown in [Figure 6](#) and [Figure 7](#) are of so called “type B”. Type B waveforms maintain zero DC component over the period of two refresh cycles. It is possible to change the visibility of the LCD segments only at the end of every second refresh period. If visibility of segments is altered more often, the DC component of the drive waveforms will cease to be zero and long term reliability of the LCD could be impacted.

It is possible to reorganize phases of the two consecutive refresh cycles to place phases, in which a particular backplane electrode is selected, next to each other. Such waveforms are of so called “type A” and are shown in [Figure 8](#). Type A waveforms maintain zero DC component over a period that is only two phases long. Therefore, it is possible to update the segment visibility much more often compared to type B waveforms. However, this fact is usually of minor importance, as the faster update rate is rarely required.

The main difference between the two types of waveform is in the frequencies of the voltages applied to the LCD cells. At first sight, it is apparent that type A waveforms contain many more edges than type B waveforms (compare [Figure 7](#) with [Figure 8](#)). The LCD cell voltages have a period equal to two refresh cycles in the case of type B waveforms and only of two phases in case of type A waveforms. This means that the LCD cells are under higher stress in terms of impurity ion migration when driven by type B waveforms. However, modern LCDs can typically handle the lower drive frequencies without any problems (see the LCD data sheet if in doubt).

The lower frequencies contained in the type B waveforms have one major advantage. Since the LCD represents a capacitive load, the drive current rises with frequency. Therefore, type B waveforms will

result in lower power consumption, which is an important parameter, especially in battery powered applications.

The length of the refresh cycle is the same for both types of waveform. Therefore, there are no differences between type A and type B waveforms in terms of refresh rate dependent optical parameters of the LCD segments (e.g., the possibility of flickering due to low refresh frequency).

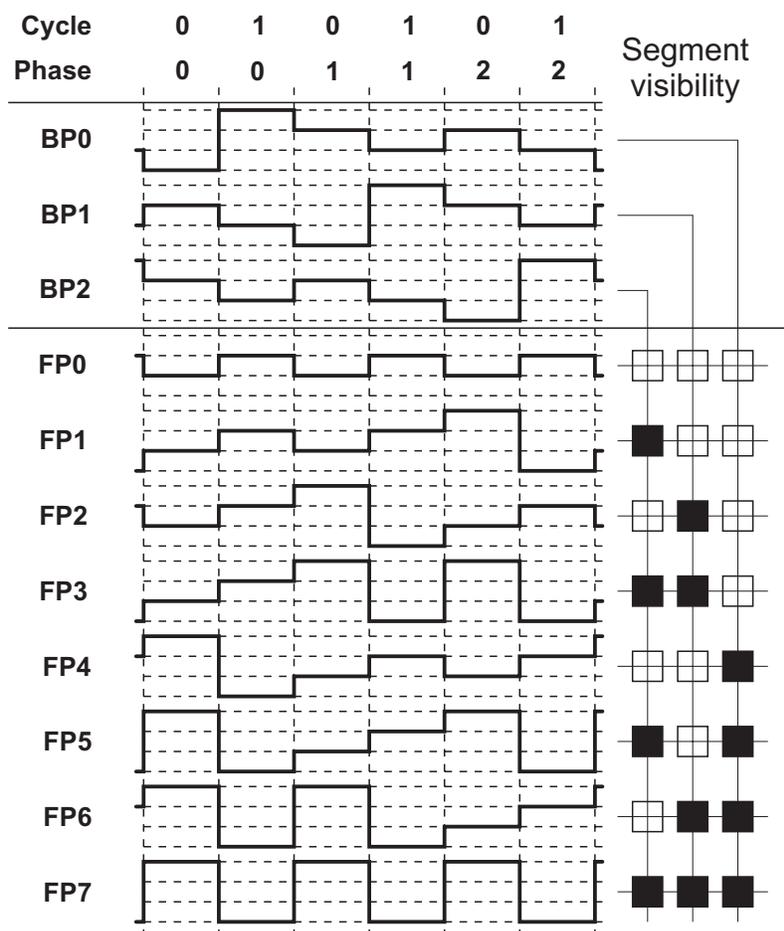


Figure 8. “Type A” Waveforms for Dynamically Driven LCDs with 1/3 Duty and 1/3 Bias Ratios

2.6 Voltages on LCD Cells Created by Different Drive Waveforms

As described in [Section 2.3](#), the apparent darkness of a segment on the display depends on the voltage across the LCD cell. To establish the optical effect of an arbitrary voltage waveform, we must calculate its RMS voltage:

$$V_{RMS} = \sqrt{\int_0^T V^2 dt} \quad \text{Eqn. 1}$$

The voltages across the LCD cells are constant during each phase. The equation can, therefore, be simplified to:

$$V_{RMS} = \sqrt{\frac{\sum_{i=0}^{n-1} V_{Phase}^2(i)}{n}} \quad \text{Eqn. 2}$$

As mentioned in [Section 2.5.2](#), the voltage across an LCD cell in the phases in which the corresponding backplane electrode is not selected is always the same (the “inactive” voltage level). This simplifies the equation even further:

$$V_{RMS} = \sqrt{\frac{V_{SelPh}^2 + (n-1) \cdot V_{nonSelPh}^2}{n}} \quad \text{Eqn. 3}$$

[Equation 3](#) is a general equation that can be used for any waveforms that keep the voltages across the cells constant in the non-selected phases. For waveforms generated according to the algorithm outlined in [Listing 1](#), we can substitute $V_{SelPh} = V_{dd}$ for visible segments, $V_{SelPh} = V_{dd} \cdot ((b-2)/b)$ for invisible segments (backplane selected) and $V_{nonSelPh} = V_{dd}/b$ (backplane not selected), where V_{dd} corresponds to the voltage level b in the algorithm description (i.e. the full supply voltage).

The above substitutions can now be used to derive equations for V_{ONRMS} and V_{OFFRMS} across cells corresponding to the visible and invisible segments:

$$V_{ONRMS} = V_{dd} \cdot \sqrt{\frac{1 + (n-1) \cdot \left(\frac{1}{b}\right)^2}{n}} \quad \text{Eqn. 4}$$

$$V_{OFFRMS} = V_{dd} \cdot \sqrt{\frac{\left(\frac{b-2}{b}\right)^2 + (n-1) \cdot \left(\frac{1}{b}\right)^2}{n}} \quad \text{Eqn. 5}$$

[Equation 4](#) and [Equation 5](#) only apply to waveforms generated according to the algorithm shown in [Listing 1](#).

It can be seen that the difference between V_{ONRMS} and V_{OFFRMS} will get smaller as the number of backplane electrodes n increases (i.e., the duty ratio gets smaller). Therefore, it becomes increasingly

difficult to ensure that V_{ONRMS} is greater than the saturation voltage V_S and V_{OFFRMS} is lower than the threshold voltage V_T . This is why the basic TN LCDs can be used only down to a certain level of duty ratio (usually down to 1/32, but the limit is dependent on the liquid crystal material and the supply voltage available).

Higher levels of multiplexing require the use of STN displays. These differ from TN LCDs in the twist angle. Twist angles from 180 to 270 degrees are typical in STN displays (as opposed to 90 degrees in TN displays). The higher twist angles result in steeper voltage-transmittance characteristics. At 270 degrees of twist the LCD cell characteristic becomes extremely steep. Where displaying shades of grey is required, the twist angle is usually lowered (e.g. to 210 degrees) to make the characteristics less steep, simplifying the drive implementation. The minimum duty ratio for STN displays is typically around 1/240.

The principle of operation is slightly different for TN and STN displays; however, the same waveforms can be used to control both types. STN displays are typically more expensive to manufacture, but are easier to drive (since V_{ONRMS} and V_{OFFRMS} can be closer together) and, typically, have a wider viewing angle.

2.7 Selecting the Right Waveforms

When trying to assess whether a particular set of waveforms is suitable for driving a particular LCD display, we can calculate the V_{ONRMS}/V_{OFFRMS} ratio and compare it with the V_S/V_T ratio. If the V_{ONRMS}/V_{OFFRMS} ratio is greater than the V_S/V_T ratio, then the band between the voltages generated by the waveforms is greater than the band required by the LCD, and the waveforms can be used for driving the particular display.

Using results from [Equation 4](#) and [Equation 5](#), we can calculate:

$$\frac{V_{ONRMS}}{V_{OFFRMS}} = \sqrt{\frac{b^2 + n - 1}{(b - 2)^2 + n - 1}} \quad \text{Eqn. 6}$$

Calculating the ratio for a few combinations of n and b reveals the optimum drive waveform configurations for LCDs with different numbers of backplane electrodes n . These results are summarized in [Table 1](#).

Table 1. V_{ONRMS}/V_{OFFRMS} Ratio for Different Combinations of Duty and Bias

| $n \setminus b$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----------------|--------------|--------------|--------------|-------|-------|-------|-------|
| 2 | 2.236 | 2.236 | 1.844 | 1.612 | 1.475 | 1.387 | 1.325 |
| 3 | 1.732 | 1.915 | 1.732 | 1.567 | 1.453 | 1.374 | 1.318 |
| 4 | 1.528 | 1.732 | 1.648 | 1.528 | 1.433 | 1.363 | 1.311 |
| 5 | 1.414 | 1.612 | 1.581 | 1.494 | 1.414 | 1.352 | 1.304 |
| 6 | 1.342 | 1.528 | 1.528 | 1.464 | 1.397 | 1.342 | 1.297 |
| 7 | 1.291 | 1.464 | 1.483 | 1.438 | 1.382 | 1.332 | 1.291 |
| 8 | 1.254 | 1.414 | 1.446 | 1.414 | 1.367 | 1.323 | 1.285 |
| 9 | 1.225 | 1.374 | 1.414 | 1.393 | 1.354 | 1.314 | 1.279 |
| 10 | 1.202 | 1.342 | 1.387 | 1.374 | 1.342 | 1.306 | 1.274 |

Table 1. V_{ONRMS}/V_{OFFRMS} Ratio for Different Combinations of Duty and Bias (continued)

| $n \setminus b$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----------------|-------|-------|--------------|--------------|--------------|--------------|-------|
| 11 | 1.183 | 1.314 | 1.363 | 1.357 | 1.330 | 1.298 | 1.268 |
| 12 | 1.168 | 1.291 | 1.342 | 1.342 | 1.319 | 1.291 | 1.263 |
| 13 | 1.155 | 1.271 | 1.323 | 1.327 | 1.309 | 1.284 | 1.258 |
| 14 | 1.144 | 1.254 | 1.306 | 1.314 | 1.300 | 1.277 | 1.254 |
| 15 | 1.134 | 1.238 | 1.291 | 1.302 | 1.291 | 1.271 | 1.249 |
| 16 | 1.125 | 1.225 | 1.277 | 1.291 | 1.283 | 1.265 | 1.245 |
| 17 | 1.118 | 1.213 | 1.265 | 1.281 | 1.275 | 1.259 | 1.240 |
| 18 | 1.111 | 1.202 | 1.254 | 1.271 | 1.267 | 1.254 | 1.236 |
| 19 | 1.106 | 1.192 | 1.243 | 1.262 | 1.260 | 1.248 | 1.232 |
| 20 | 1.100 | 1.183 | 1.234 | 1.254 | 1.254 | 1.243 | 1.228 |
| 21 | 1.095 | 1.175 | 1.225 | 1.246 | 1.247 | 1.238 | 1.225 |
| 22 | 1.091 | 1.168 | 1.217 | 1.238 | 1.241 | 1.234 | 1.221 |
| 23 | 1.087 | 1.161 | 1.209 | 1.231 | 1.235 | 1.229 | 1.218 |
| 24 | 1.083 | 1.155 | 1.202 | 1.225 | 1.230 | 1.225 | 1.214 |
| 25 | 1.080 | 1.149 | 1.195 | 1.219 | 1.225 | 1.221 | 1.211 |
| 26 | 1.077 | 1.144 | 1.189 | 1.213 | 1.220 | 1.217 | 1.208 |
| 27 | 1.074 | 1.139 | 1.183 | 1.207 | 1.215 | 1.213 | 1.205 |
| 28 | 1.072 | 1.134 | 1.178 | 1.202 | 1.210 | 1.209 | 1.202 |
| 29 | 1.069 | 1.130 | 1.173 | 1.197 | 1.206 | 1.205 | 1.199 |
| 30 | 1.067 | 1.125 | 1.168 | 1.192 | 1.202 | 1.202 | 1.196 |
| 31 | 1.065 | 1.122 | 1.163 | 1.188 | 1.198 | 1.198 | 1.193 |
| 32 | 1.063 | 1.118 | 1.159 | 1.183 | 1.194 | 1.195 | 1.191 |

The highest value of the V_{ONRMS}/V_{OFFRMS} ratio available for a given number of backplane electrodes is highlighted in bold type on each line in [Table 1](#). Where two values are the same on a single line, the value corresponding to the lower number of voltage levels used in the drive scheme is highlighted.

While the table highlights the optimum driving scheme for a particular display, any other scheme on the same table row will work, provided the ratio indicated by the table is higher than the V_S/V_T ratio dictated by the LCD properties.

2.8 Contrast Control

When TN LCDs are viewed at an angle away from their optical axis, the contrast of the displayed image decreases. Typically, the angular optical properties (viewing characteristics) get worse with increasing multiplex rate (decreasing duty ratio).

The TN LCD voltage-transmittance characteristic also suffers from a negative temperature coefficient (the V_S and V_T voltages decrease with increasing temperature). This leads to lower contrast of the displayed image at extreme temperatures.

The viewing properties can be optimized and the temperature coefficient can be compensated for by adjusting the V_{ONRMS} and V_{OFFRMS} voltages. Figure 9 shows how the waveforms can be adjusted to fit the transmittance characteristics of a particular LCD (the temperature drift of the LCD properties is exaggerated for clarity). Note that the gap between V_{ONRMS} and V_{OFFRMS} changes as the waveforms are adjusted (the V_{ONRMS}/V_{OFFRMS} ratio remains constant).

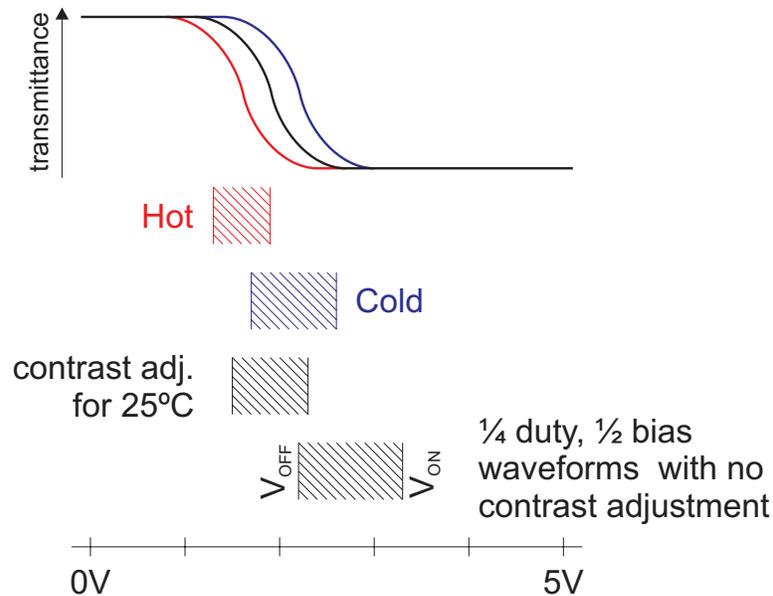


Figure 9. Adjusting the Drive Waveforms to Improve Contrast of the LCD Image

It is possible to use one of the contrast adjustment methods described in the following sections to adjust the waveform voltages downwards (as shown in Figure 9). If the natural waveform voltages (i.e., with no contrast adjustment) are lower than the voltages required by the LCD, the only possibilities are to increase the supply voltage of the driver chip or to increase the bias ratio.

2.8.1 Adjusting the Supply Voltage

Adjusting the supply voltage of the driver chip output buffers is the traditional method of shifting the V_{ONRMS} and V_{OFFRMS} voltages of the waveforms. Typically, this is done by using a potentiometer with a low enough impedance as a voltage divider. Several diodes or a thermistor may be added into the circuit if temperature drift of the LCD parameters has to be compensated for.

When general purpose pins of a microcontroller are used to drive the LCD, this method is still available. However, a typical microcontroller has only one supply voltage available for all the I/O pins, and therefore the application must tolerate the arbitrary choice of supply voltage. Many modern microcontrollers offer a wide range of possible supply voltages (typically from 3 V to 5 V). Choosing the right power supply voltage for the application may be a quick and simple solution, where temperature compensation is not required.

2.8.2 Adding Contrast Adjustment Phases

Another method of adjusting the V_{ONRMS} and V_{OFFRMS} voltages is to add another phase to the refresh cycle, which will contribute towards lowering the RMS voltage of the cell voltage. This can be done by keeping the voltage the same on all frontplane and backplane electrodes for the duration of this extra phase.

Theoretically, any single voltage level would be sufficient for all contrast adjustment phases. However, transitions of I/O voltages on different ports cannot happen at exactly the same time, when the waveforms are generated on general purpose I/O pins of a microcontroller. The time shift between updates of voltages on pins belonging to different ports causes an additional DC offset to appear across the LCD cells.

Compensation for this phenomenon can be achieved by introducing two opposite voltage levels for the contrast adjustment phases in odd and even refresh cycles. An example of waveforms with 1/2 duty and 1/2 bias ratios with the additional contrast adjustment phases is shown in [Figure 10](#).

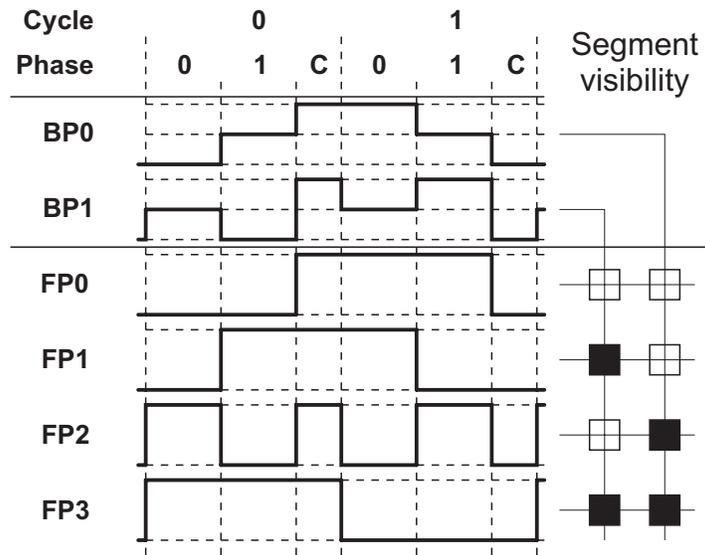


Figure 10. Introduction of Contrast Adjustment Phases

The contrast adjustment phases can have an arbitrary length. This enables very accurate adjustment of the waveform RMS voltages. LCD manufacturers typically recommend an optimal refresh period for their LCDs. Whenever the length of the contrast adjustment phase is changed, the lengths of the other phases must be adjusted to ensure that the refresh cycle length remains the same.

The equations for calculating V_{ONRMS} and V_{OFFRMS} voltages change, by the addition of the contrast adjustment phases, to:

$$V_{ONRMS} = V_{dd} \cdot \sqrt{\frac{t + (n - 1) \cdot \left(\frac{1}{b}\right)^2 \cdot t}{nt + c}} \quad \text{Eqn. 7}$$

$$V_{OFFRMS} = V_{dd} \cdot \sqrt{\frac{\left(\frac{b-2}{b}\right)^2 \cdot t + (n-1) \cdot \left(\frac{1}{b}\right)^2 \cdot t}{nt + c}} \quad \text{Eqn. 8}$$

where t is the length of the normal phases and c is the length of the contrast adjustment phases. It can be seen that the V_{ONRMS}/V_{OFFRMS} ratio remains the same.

3 Driving TN and STN LCDs Using General Purpose Microcontroller Pins

3.1 Selection of Waveforms for the GPIO Drive

Driving LCDs with a set of general purpose input/output (GPIO) pins on a microcontroller can result in a very cost effective and flexible solution. The microcontroller load resulting from the LCD drive is usually relatively low, and the cost of the specialized on-chip or off-chip LCD controller can be saved.

Waveforms for statically driven LCDs require only two discrete voltage levels (see [Figure 5](#)). General purpose outputs of a typical microcontroller can, therefore, be connected directly to the LCD. However, statically driven LCDs are not suitable in many applications because of the high number of connections required, and their associated cost.

Dynamically driven LCDs require more than two voltage levels to be present in the driving waveforms. In general, these are not trivial to obtain with the simple CMOS I/O structure used in the general purpose pins of modern microcontrollers. The additional voltage levels can be obtained by using several GPIO pins and/or external resistor networks.

Waveforms with 1/2 bias ratio ([Figure 6](#)) use three voltage levels for the backplane electrodes, and only two voltage levels for the frontplane electrodes. This means that the frontplane electrodes of the LCD can be connected directly to GPIO pins of the microcontroller. To keep the duty ratio favorable, dynamically driven LCDs typically feature more frontplane electrodes than backplane electrodes. Therefore, most LCD electrodes can be driven directly by the GPIO pins, and only the few backplane electrodes require some additional circuitry.

The third voltage level required for driving the backplane electrodes with 1/2 bias ratio waveforms can be obtained conveniently by making use of the input capability of the GPIO pins. A GPIO pin represents a high impedance load when configured as an input. In such a state, a simple resistor network can be used to determine the voltage level on the pin. Whenever the GPIO pin is configured as an output, it overdrives the resistor network and applies a high or low voltage level to the backplane electrode. The resistor network arrangement is shown in [Figure 11](#). The values of the resistors are not critical in this case. The voltage level imposed by the resistive divider will be influenced only by leakage currents of the microcontroller and the LCD, which are typically in the low μA range, or lower. Another factor influencing the choice of values for the resistors is the time constant of the circuit formed by the resistors and the stray capacitance of the connection between the microcontroller and the LCD. The stray capacitance can be significant in cases where the LCD is connected using long PCB tracks, a connector or

a flat cable. For the waveforms to maintain their properties, the time constant of the RC circuit must be small compared to the length of one phase (t).

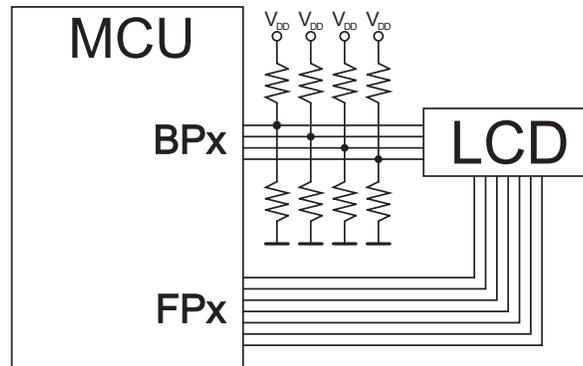


Figure 11. A Simple Resistor Divider Can Be Used to Generate an Additional Voltage Level

3.2 Suitability of 1/2 Bias Ratio Waveforms

The previous section explained the advantages of using the 1/2 bias ratio waveforms when driving an LCD using GPIO pins of a microcontroller. However, using the 1/2 bias waveforms is only optimal when driving an LCD with two backplane electrodes (1/2 duty ratio). For the best possible V_{ONRMS}/V_{OFFRMS} ratio, a higher number of discrete voltage levels must be used when driving LCDs with three and more backplane electrodes (see Table 1). When the 1/2 bias ratio waveforms are used for LCDs with three and more backplane electrodes, the V_{ONRMS}/V_{OFFRMS} ratio is lower, compared to the optimum waveforms. However, comparison of the values in Table 1 reveals that the V_{ONRMS}/V_{OFFRMS} ratio of the 1/2 bias waveforms is only slightly below the optimum value for high duty ratios. The voltage band between V_{ONRMS} and V_{OFFRMS} is reduced by 20% in the case of a 1/3 duty ratio, by 28% in the case of a 1/4 duty ratio, and by 32% in case of a 1/5 duty ratio.

Modern TN LCDs have relatively low V_S/V_T ratios and using the suboptimal 1/2 bias waveforms is usually possible for the higher duty ratios. However, the narrower band between V_{ONRMS} and V_{OFFRMS} may require a more accurate temperature compensation to maintain the best possible optical properties of the display over a wide temperature range. This depends on properties of the particular LCD — variation of V_S and V_T with temperature can usually be found in data sheets provided by LCD manufacturers.

3.3 Generating 1/2 Bias Ratio Waveforms

LCDs require relatively low-frequency waveforms for their drive. The algorithm to generate the waveforms can, therefore, be implemented as an interrupt service triggered by a periodic interrupt source (e.g., a timer). Since the frequency of interrupts is low, the overhead caused by processing the associated exceptions is usually insignificant.

The algorithm follows the general flow outlined in Listing 1, with some minor modifications to make it suitable for interrupt routine implementation. The modified algorithm is shown in Listing 2 in a pseudo-C programming language.

The algorithm in [Listing 2](#) does not include generation of the contrast adjustment phases. This is not complicated, but usually involves re-programming the interrupt rate of the associated timer to achieve fine adjustment of the waveform voltages. Practical implementation of this algorithm, including generation of contrast adjustment phases, is discussed in [Section 4](#).

```

if (refresh_cycle==EVEN) { /* even refresh cycle */
    Configure pin BP(phase) as output and drive it low
    Configure all other BP(x) pins as inputs /* mid-rail */
    Drive pins FP(x) corresponding to visible segments high
    Drive pins FP(x) corresponding to invisible segments low
} else { /* odd refresh cycle */
    Configure pin BP(phase) as output and drive it high
    Configure all other BP(x) pins as inputs /* mid-rail */
    Drive pins FP(x) corresponding to visible segments low
    Drive pins FP(x) corresponding to invisible segments high
}
Update the phase number and the refresh_cycle ODD/EVEN indicator

```

**Listing 2. Algorithm for Creating 1/2 Bias Ratio Waveforms on GPIO Pins
(for Repeated Execution in an Interrupt Service Routine)**

4 Driver Implementation Using XGATE

The flow of data, between an LCD driver and the application that uses the driver, is in one direction only. The application generates the display data (i.e., tells the driver which segments should be visible and invisible), and the driver generates the appropriate waveforms. All data is written by the application and read by the driver. This makes the XGATE co-processor on the S12X family an ideal platform for implementing the LCD driver. There is no opportunity for memory sharing violations; the XGATE concept ensures that the application is not interrupted by the waveform generation task and the application itself has only to update the display data in the system memory, as appropriate. From the application's point of view, the XGATE LCD driver behaves like a virtual LCD peripheral with a specific area of the system RAM used to share the display data.

Practical implementation of the LCD driver for the XGATE co-processor can be found in the zip file that accompanies this application note (AN3219SW.zip).

4.1 Structure of the Driver

The driver is split into several files:

1. "lcd.c" service functions executed by the S12X CPU
2. "lcd.cxgate" the main function of the LCD driver executed by the XGATE
3. "lcd.h" user defined macros for mapping the appropriate resources to the driver

The driver functions use macros to access hardware resources of the microcontroller. All the application dependent macros are located in the "lcd.h" header file. It is possible, therefore, to customize the driver by

editing the header file only; there is no need to edit the source files, unless a major change in functionality is required.

4.2 Configuring the Driver

All hardware specific configurations of the driver are performed through modification of the macro definitions located in the “lcd.h” header file. 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 in the following sections. The macro definitions used in the examples assume that the Periodic Interrupt Timer channel 0 is used to time the waveform generation process, the backplane resistor dividers are powered from pin PTB0, backplane LCD electrodes are connected to pins BP[0:3] (4 pins), and frontplane LCD electrodes are connected to PTB[1:7] and PTA[0:7] (15 pins).

4.2.1 LCD_BP_CNT

The LCD_BP_CNT macro defines the number of backplane electrodes of the LCD. The duty ratio of the waveforms generated by the driver is $1/\text{LCD_BP_CNT}$.

Example: `#define LCD_BP_CNT 4 /* number of backplane electrodes */`

4.2.2 LCD_FP_CNT

The LCD_FP_CNT macro defines the number of frontplane electrodes of the LCD. Together with LCD_BP_CNT, the macro defines the size of the array used to hold the display data.

Example: `#define LCD_FP_CNT 15 /* number of frontplane electrodes */`

4.2.3 LCD_FRAME_FREQUENCY

The LCD_FRAME_FREQUENCY macro defines the desired refresh cycle frequency for the display in Hertz. The CPU calculates periods of the normal and contrast adjustment phases based on the frequency specified by this macro. See [Section 4.3](#) for more details.

Example: `#define LCD_FRAME_FREQUENCY 64 /* desired refresh cycle frequency in Hz */`

4.2.4 LCD_IDLE_CNT

The LCD_IDLE_CNT macro defines the number of contrast adjustment phases per refresh cycle. Higher numbers of contrast adjustment phases per refresh cycle improves the contrast adjustment range, but increases the XGATE load (due to the need to generate additional phases) and decreases the contrast adjustment resolution.

Example: `#define LCD_IDLE_CNT 2 /* # of contrast adj. phases per cycle */`

4.2.5 LCD_INT_DIVISOR

The LCD_INT_DIVISOR macro defines the scaling factor between the bus frequency and the timer tick. It is used to make sure that the relatively long phase time fits into the timer period register, which can have a restricted size.

Example: `#define LCD_INT_DIVISOR 4 /* divisor of bus clock for the interrupt source */`

4.2.6 LCD_INT_FLAG_CLEAR()

The LCD_INT_FLAG_CLEAR() macro is used by the driver to clear the interrupt flag of the timer used to time the waveform generation process.

Example: `#define LCD_INT_FLAG_CLEAR() PIT.pitmf.byte = PTF0 /* clear the int flag */`

4.2.7 LCD_INT_PERIOD_SET(i)

The LCD_INT_PERIOD_SET(i) macro is used by the driver to re-program the interrupt rate of the timer. This happens twice during every refresh cycle, once to set the normal phase period, and once to set the contrast adjustment phase period.

Example: `#define LCD_INT_PERIOD_SET(i) PIT.pitld0.word=i /* update period */`

4.2.8 LCD_INT_CONFIGURE()

The LCD_INT_CONFIGURE() macro is used to configure one of the on-chip timers to serve as a timebase for the waveform generation process. This operation is performed by the CPU during the driver startup.

Example: `#define LCD_INT_CONFIGURE() { /* macro to configure the timer */\
PIT.pitce.bit.pce0 = 1; /* enable PIT channel 0 */\
PIT.pitinte.bit.pinte0 = 1; /* enable interrupts from ch0 */\
PIT.pitmux.bit.pmux0 = 0; /* assign ch0 to microtimer 0 */\
PIT.pitmtld0.byte = (LCD_INT_DIVISOR-1); /* clock divider */\
PIT.pitld0.word = LCD_data.contrast; /* set int. period */\
PIT.pitcflmt.byte = PITE | PITFRZ | PFLMT0; /* enable the PIT */\
PIT.pitflt.bit.pflt0 = 1; /* force reload of counter 0 */\
Interrupt.int_cfaddr = (0x3D << 1) & 0xf0; /* XGATE, prio 1 */\
Interrupt.int_cfdata[0x3D & 0x07].byte = RQST|1;\
}`

4.2.9 LCD_INT_STOP()

The LCD_INT_STOP() macro is used during shutdown of the driver to disable interrupts from the timer.

Example: `#define LCD_INT_STOP() PIT.pitce.bit.pce0=0 /* stop the interrupt source */`

4.2.10 LCD_SET_FP_POS(data)

The LCD_SET_FP_POS(data) macro is used to drive data to the frontplane electrode pins. The parameter of the macro is a pointer to an unsigned int array. The size of the array depends on the number of frontplane

electrodes defined by the LCD_FP_CNT macro. One bit is needed for each frontplane electrode. The macro is used for driving the frontplane electrodes in even refresh cycles.

Example: `#define LCD_SET_FP_POS(data) { PORTB.byte=(data[0]<<1)|1;
PORTA.byte=(data[0]>>7); }`

4.2.11 LCD_SET_FP_NEG(data)

The LCD_SET_FP_NEG(data) macro is used to drive the one's complement of data to the frontplane electrode pins. The parameter of the macro is a pointer to an unsigned int array. The size of the array depends on the number of frontplane electrodes defined by the LCD_FP_CNT macro. One bit is needed for each frontplane electrode. The macro is used for driving the frontplane electrodes in odd refresh cycles.

Example: `#define LCD_SET_FP_NEG(data) { PORTB.byte=((~data[0])<<1)|1;
PORTA.byte=((~data[0])>>7); }`

4.2.12 LCD_ONE_BP_LOW(n)

The LCD_ONE_BP_LOW(n) macro is used to drive one backplane electrode low and configure all other backplane electrode pins as inputs. The parameter is the number of the electrode from 0 to LCD_BP_CNT-1. The macro is used for driving the backplane electrodes in even refresh cycles.

Example: `#define LCD_ONE_BP_LOW(n) { DDRP.byte&=~LCD_PTP_MASK;
PTP.byte&=~LCD_PTP_MASK;
DDRP.byte|=(1<<n); }`

4.2.13 LCD_ONE_BP_HIGH(n)

The LCD_ONE_BP_HIGH(n) macro is used to drive one backplane electrode high and configure all other backplane electrode pins as inputs. The parameter is the number of the electrode from 0 to LCD_BP_CNT-1. The macro is used for driving the backplane electrodes in odd refresh cycles.

Example: `#define LCD_ONE_BP_HIGH(n) { DDRP.byte&=~LCD_PTP_MASK;
PTP.byte|=LCD_PTP_MASK;
DDRP.byte|=(1<<n); }`

4.2.14 LCD_ALL_BP_LOW()

The LCD_ALL_BP_LOW() macro is used to drive all backplane electrodes low. The macro is used in the contrast adjustment phases at the end of odd refresh cycles.

Example: `#define LCD_ALL_BP_LOW() { PTP.byte&=~LCD_PTP_MASK;
DDRP.byte|=LCD_PTP_MASK;
PORTB.byte&=~0x01; }`

4.2.15 LCD_ALL_BP_HIGH()

The LCD_ALL_BP_HIGH() macro is used to drive all backplane electrodes high. The macro is used in the contrast adjustment phases at the end of even refresh cycles.

Example:

```
#define LCD_ALL_BP_HIGH()      { PTP.byte|=LCD_PTP_MASK;
                                   DDRP.byte|=LCD_PTP_MASK; }
```

4.2.16 LCD_ALL_FP_LOW()

The LCD_ALL_FP_LOW() macro is used to drive all frontplane electrodes low. The macro is used in the contrast adjustment phases at the end of odd refresh cycles.

Example:

```
#define LCD_ALL_FP_LOW()        { PORTB.byte=0x01; PORTA.byte=0x00; }
```

4.2.17 LCD_ALL_FP_HIGH()

The LCD_ALL_FP_HIGH() macro is used to drive all frontplane electrodes high. The macro is used in the contrast adjustment phases at the end of even refresh cycles.

Example:

```
#define LCD_ALL_FP_HIGH()       { PORTB.byte=0xff; PORTA.byte=0xff; }
```

4.2.18 LCD_ALL_CONFIGURE()

The LCD_ALL_CONFIGURE() macro is used to configure all LCD electrode pins as outputs. It is used during driver startup.

Example:

```
#define LCD_ALL_CONFIGURE()     { DDRP.byte|=LCD_PTP_MASK;
                                   DDRA.byte=0xff;
                                   DDRB.byte=0xff; }
```

4.2.19 LCD_ALL_PINS_LOW()

The LCD_ALL_PINS_LOW() macro is used to bring all LCD electrode pins low. It is used during driver startup and shutdown.

Example:

```
#define LCD_ALL_PINS_LOW()      { PORTB.byte=0;
                                   PORTA.byte=0;
                                   PTP.byte&=~LCD_PTP_MASK; }
```

4.3 Using the Driver

Apart from influencing the information displayed on the LCD by changing the shared display data, the application can manage operation of the driver through calling functions provided in the “lcd.c” source file. Behavior of these functions and their parameters are described in the following sections.

4.3.1 Function LCD_start

Function LCD_start is used to startup the LCD driver. It accepts two parameters:

- unsigned int bus_frequency — This parameter specifies the frequency used by the on-chip timer, which is used to time the waveform generation process. The function uses this parameter to calculate the number of “ticks” the timer must count to time the normal and the contrast adjustment phases.
- int contrast_adj — This parameter influences the length of the contrast adjustment phases. The contrast adjustment phases and the normal phases will be of the same length when contrast_adj is equal to zero. Values above zero extend the duration of the contrast adjustment phases; values below zero reduce it.

4.3.2 Function LCD_stop

This function immediately stops operation of the driver. The timer interrupt is disabled and all LCD pins are brought low.

4.3.3 Function LCD_contrast

Function LCD_contrast can be used to change the length of the contrast adjustment phases, without the need to shutdown and restart the driver. Parameters required by this function are identical to the parameters of function LCD_start. The new length of the normal and contrast adjustment phases will be applied at the end of the next odd cycle.

4.4 Driver Performance

This section describes results obtained after configuring the driver for an LCD with four backplane electrodes and a refresh cycle frequency of 64 Hz (as per the macro definition examples shown in [Section 4.2](#)).

4.4.1 Waveforms Generated by the Driver

All waveforms generated by the driver can be seen in the scope screen capture in [Figure 12](#). The horizontal timebase is 4 ms per division; the vertical resolution is 7 V per division (the microcontroller is powered by a 5 V power supply). There are two contrast adjustment phases in every refresh cycle to increase the effective range of the contrast adjustment. The driver is started with the contrast_adj parameter having a small negative value; therefore, the contrast adjustment period is slightly shorter than two normal phases.

The effect of a large negative and a large positive value of contrast_adj variable can be seen in [Figure 13](#) and [Figure 14](#). Note that function LCD_contrast recalculates the length of both the normal and the contrast adjustment phases so that the refresh cycle has the same length, irrespective of the contrast adjustment of the waveforms.

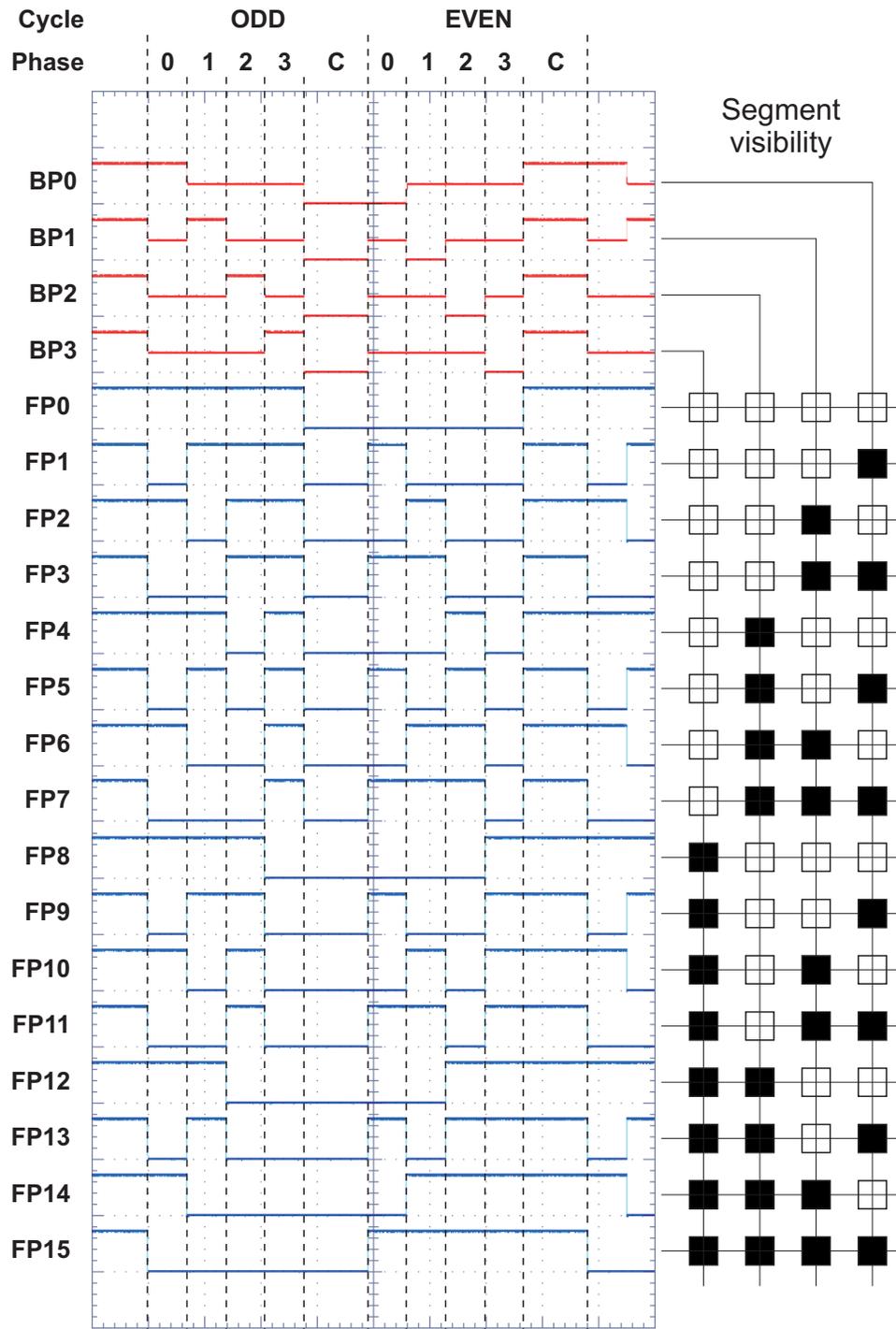


Figure 12. Waveforms with 1/4 Duty Ratio Generated by the Driver

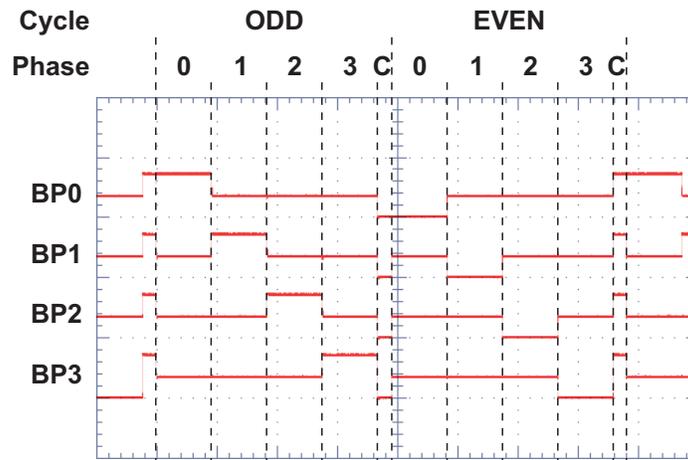


Figure 13. Very Short Contrast Adjustment Phase

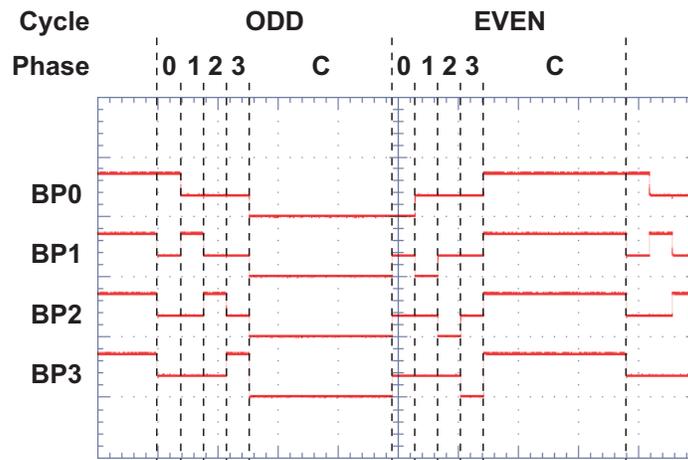


Figure 14. Very Long Contrast Adjustment Phase

4.4.2 Memory Size and Execution Time

The required memory size for both code and data, and XGATE performance requirements of the driver, are detailed in [Table 2](#). The frame frequency was set for 64 Hz with two contrast adjustment phases generated in every refresh cycle. All the measurements were performed using the CodeWarrior compiler/debugger version 4.5.

The code size includes only the driver code itself (executing on XGATE). It does not include the support functions executed on the S12X CPU.

The support functions are executed by the S12X CPU and have the following sizes:

- LCD_start - 111 bytes
- LCD_stop - 15 bytes

- LCD_contrast - 70 bytes

The very general implementation of function LCD_contrast imposes almost no restrictions on its parameters and the values of the user-defined macros. To allow this level of freedom, the calculations are performed in long integer arithmetic; therefore, the function needs two library functions for its execution:

- Long integer division - 176 bytes
- Long integer shift - 14 bytes

It would be possible to optimize this function for a smaller memory footprint and faster execution, providing some restrictions are imposed on the macro values and/or parameters.

The support functions are executed by the CPU on a very infrequent basis (startup, shutdown, and temperature compensation, or operator adjustment of the LCD contrast). The execution time of these functions is of little importance, therefore, when considering the overall microcontroller load caused by the LCD driver.

The XGATE executes the main body of the LCD driver all the time, so the load created by this function is important. Two aspects of this load must be considered. First is the average load over a long period of time; however, the driver performs different operations during different phases of each refresh cycle and, therefore, the load is uneven. The second, and more important factor, is the peak execution time, i.e., the longest time the LCD driver ever takes to execute. The peak execution time influences interrupt latency of other XGATE tasks.

**Table 2. Required Memory Size and Performance Details
(Execution on XGATE at 40 MHz Bus Frequency)**

| LCD configuration | Code Size (bytes) | Data Size (bytes) | Average Load (%) | Peak execution time (µs) |
|-------------------|-------------------|-------------------|------------------|--------------------------|
| 2 BP, 16 FP | 332 | 18 | 0.029 | 1.8 |
| 4 BP, 15 FP | 332 | 26 | 0.044 | 2.0 |
| 4 BP, 20 FP | 374 | 42 | 0.053 | 2.7 |
| 4 BP, 24 FP | 374 | 42 | 0.053 | 2.7 |
| 4 BP, 32 FP | 406 | 42 | 0.058 | 2.9 |

5 References

1. “Twisted nematic liquid crystal display”, Stephen Palmer, LC-TEC Displays AB, 2005
2. “100 Years of Commercial Liquid-Crystal Materials”, Werner Becker and Hans-Juergen Lemp, Merck & Co Inc., 2004
3. MC9S12XDP512 Data Sheet, Freescale Semiconductor Inc., 2005.

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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)
support@freescale.com

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 Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
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.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2006. All rights reserved.