SHARP

Application Note

Sharp Memory LCD Technology

Ken Green, Sharp Microelectronics of the Americas

INTRODUCTION

Sharp's Memory LCDs represent a step forward in power-saving monochrome displays. The modules have an SPI interface and do not require the usual LCD controller. Most of the panels have a 10-conductor 0.5 mm pitch FFC that contains all of the signals needed, including power. This Application Note will touch briefly upon the Theory of Operation, as well as explain some general steps to programming the modules, and give some practical examples.

SEPTEMBER 2013

THEORY OF OPERATION

Sharp's Memory LCDs derive their name from the way the LCD pixel array is used to store image data. Each pixel in the LCD array forms a 1-bit write-only memory; therefore the frame buffer is internal to the panel. This feature relieves the processor and bus from the overhead of continuous data transfers to refresh the image, as the image only needs to be written once to be retained. The image is retained indefinitely as long as there is sufficient current to do so. The LCDs also do not exhibit image retention problems, so there is no worry about keeping a single image displayed for a very long time.

When a change to the displayed image is desired, there is no need to clear and rewrite the entire existing image; only the lines that are to be changed need to be written. The displays have a minimum addressable unit of one line.

Since the displays are write-only, there is no way to read back any information; so if part of a line is to be changed, a copy of it will have to be maintained elsewhere, such as in local memory. This is because the new data for the line must be inserted into the other, existing data for that line; since the minimum addressable unit is one line. After the data has been inserted, the line to be changed can be sent to the LCD. If major parts of the entire image are to be changed, then the entire frame should be duplicated in the processor memory.

Using two panels as an example for how to compute the necessary memory requirements, we get:

- 1.28-inch (128 × 128 pixels) = 128 × 128 / 8 = 2048 bytes
- 2.7-inch (400 × 240 pixels) = 400 × 240 / 8 = 12000 bytes

INTERFACE

The interface to most of the panels is through a 10-conductor FFC with 0.5 mm pitch. A sample connector is the Hirose FH19SC-10S-0.5SH(05). The signals are compatible with 3.3 V logic levels. The SPI clock will run at 2 MHz for 5 V parts; at 1 MHz for 3 V parts.

USING 5V PANELS IN 3V SYSTEMS

Because memory LCD panels are such low power, it is still easy to use panels that require 5 V in systems that only have 3 V available. A simple high efficiency charge pump circuit can be used to boost the 3 V to 5 V as shown in Figure 1.

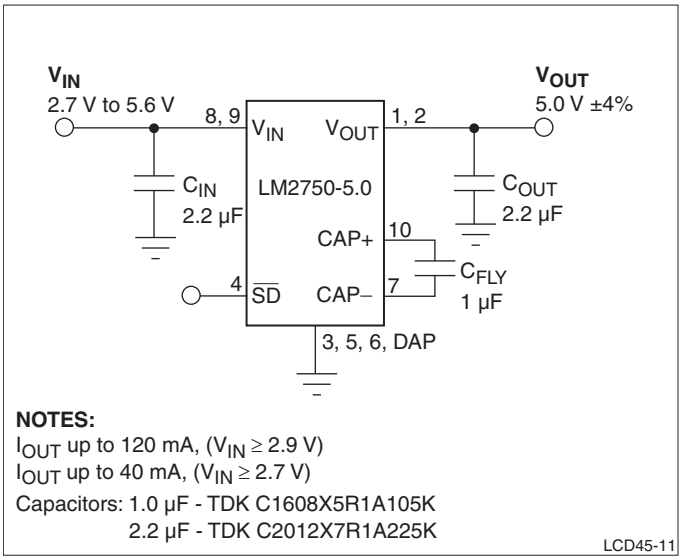

Figure 1. Charge Pump Voltage Boost Circuit

VCOM

When designing with a memory LCD panel, a decision must be made as to how VCOM will be generated. VCOM is an alternating signal that prevents a DC bias from being built up within the panel. Memory LCDs do not generate this signal internally. It can be supplied using one of two methods: software, or an external clock. The mode is selected by the EXTMODE pin on the interface.

If external clock is selected [EXTMODE = H], the clock should be supplied on the EXTCOMM pin. See Figure 2.

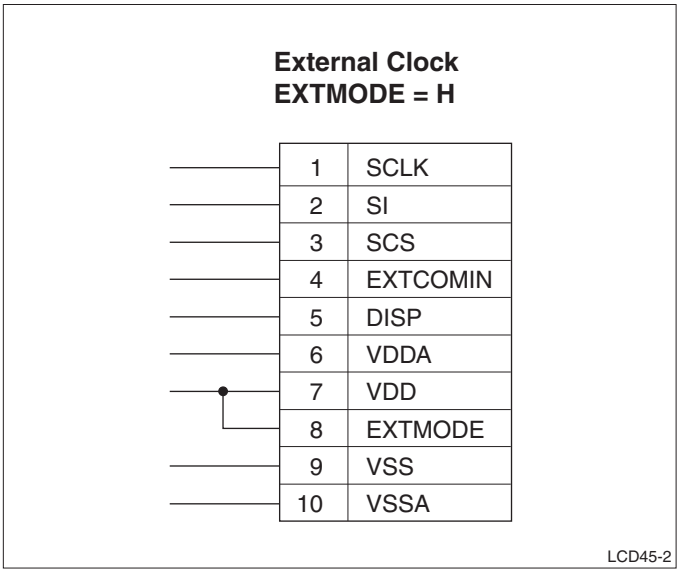

Figure 2. Selecting External Clock

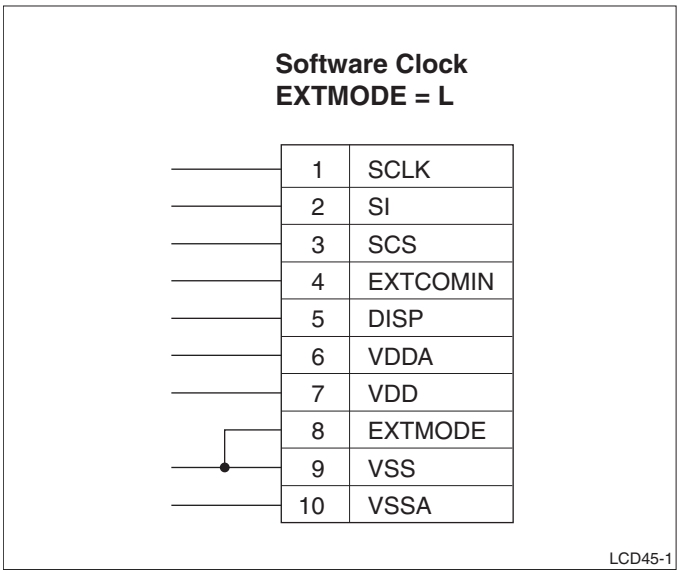

Figure 3. Selecting Internal Clock

If external clock is selected [EXTMODE = H], a clock must be supplied to the EXTCOMM pin. The clock is a pulse sent to the panel at a constant rate. Every time a pulse is sent, the internal level of VCOM toggles (much like a flip-flop). The rate is specified in each part's corresponding Specifications. In general, for panels that require a 3 V power supply, the frequency of EXTCOMM is 60 Hz. For 5 V panels the frequency is slower and more lenient, 1 – 20 Hz; again, check the Specifications. Higher frequency gives better contrast; lower frequency saves power.

When the software clock is selected [EXTMODE = L], bit V of the command bit string reflects the actual state of VCOM. See Figure 3. This bit must change states (by writing to the panel) at the frequency listed in the corresponding Specifications.

Any command can be used to change the state of the VCOM bit. If no update to the pixel memory is needed when it is time to change VCOM, the Change VCOM command can be used to change the state. It is important that the time intervals between VCOM level changes be as symmetrical as possible. In the descriptions below, the V bit represents the desired VCOM state if software VCOM control is selected. If external clock is selected, then the V-bit state doesn't matter.

LCD COMMANDS

There are four commands that can be sent to a memory LCD panel:

- Write Line
- Write Multiple Lines
- Change VCOM
- Clear Screen

The commands are explained below. The illustrations are in little-endian format with the LSB on the left and the MSB on the right.

DATA STRUCTURE

Data is sent to the panel in little-endian format; with the LSB first. The data width for the Write Line and Write Multiple Line commands depends on the horizontal resolution of the panel itself. Therefore, if you're working with a panel having a resolution of 400 × 240, then the data width for this panel will require a minimum of 400 bits of data (plus overhead).

WRITE LINE

The minimum amount of data that can be written to the panel is one line. The actual width of the data written depends on the horizontal resolution of the panel itself. Therefore, a panel with a resolution of 400 × 240 will require a 400 bits of data (plus overhead). See Figure 4.

The command structure for Write Line is as follows:

- Command: 8 bits (see above regarding V-bit)
- Line address: 8 bits
- Data bits: leftmost pixel first, with data width depending on the resolution of the panel. For instance, a panel 400 pixels wide will require 400 data bits.
- Trailer: 16 bits. These clocks allow the panel time to transfer the data from the incoming latch to the pixel memory.

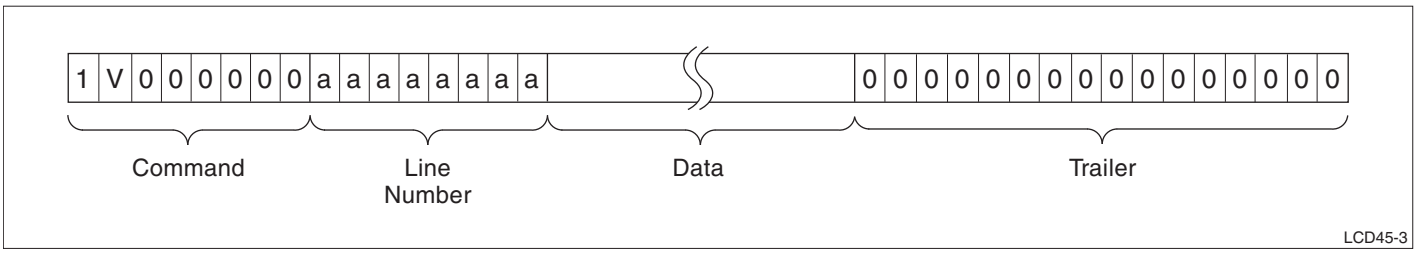

Figure 4. Write Line Data String

WRITE MULTIPLE LINES

Multiple lines can be updated quickly with this command. A line address is still used so the lines do not have to be successive.

This command begins the same as the Write Line command; using the same Write Line command bits and data bits. See Figure 5.

After the data bits follow 8 trailer bits (instead of 16), then the address of the next line (8 bits) to be written, followed by the data bits for that line, and so on until all of the desired lines are written. See Figure 6.

For the last line to be written, use 16 trailer bits (see Figure 7) instead of 8 bits and a line address.

CLEAR SCREEN

This command clears the screen to all white by writing 0's to all of the memory locations in the frame buffer. See Figure 8. The command structure is as follows:

- Command: 8 bits (including V-bit)
- Trailer: 8 bits (allows for latch transfer time)

CHANGE VCOM

This command is only used if [EXTMODE = L]. It is used to change the state of VCOM if no other command command (all commands have the ability to change the state of VCOM) is sent within the appropriate amount of time to maintain proper VCOM frequency. See Figure 9.

The command structure is:

- Command: 8 bits
- Trailer: 8 bits (allows for latch transfer time)

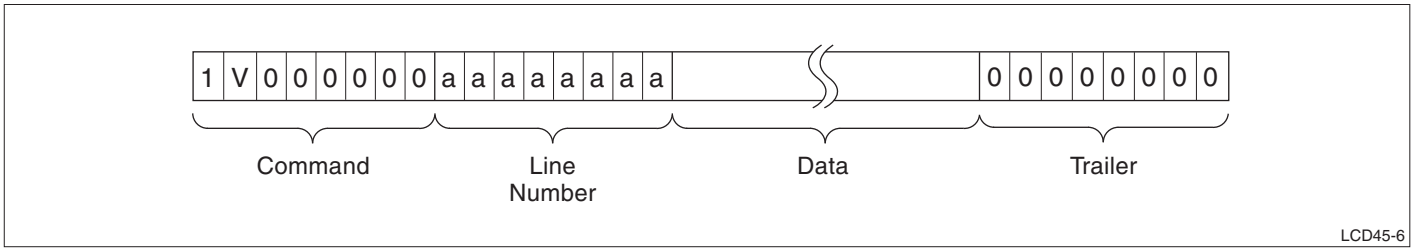

Figure 5. Write Multiple Lines Data String, First Line

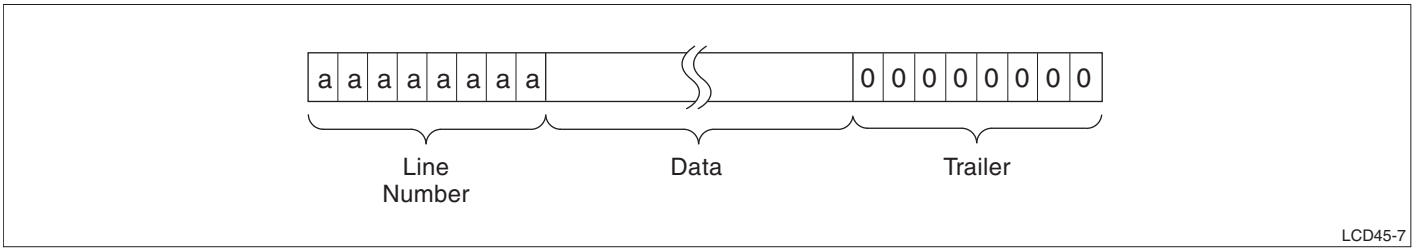

Figure 6. Write Multiple Lines Data String, Intermediate Line

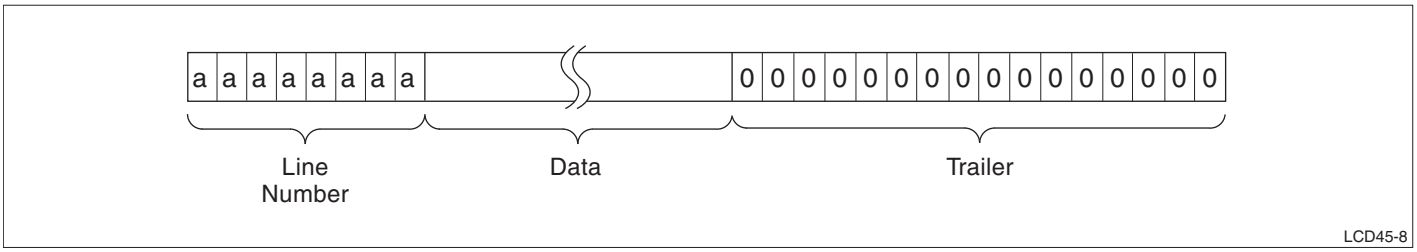

Figure 7. Write Multiple Lines Data String, Last Line

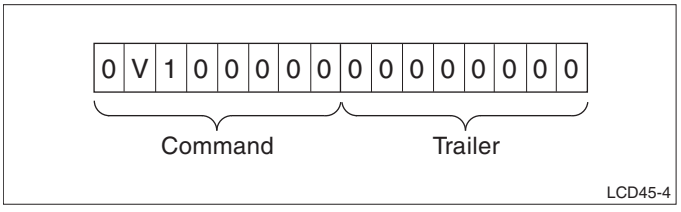

Figure 8. Clear Screen Data String

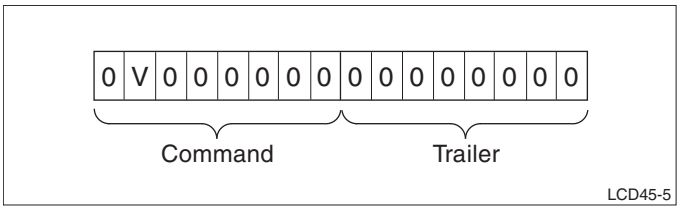

Figure 9. Toggle VCOM Data String

SOFTWARE IMPLEMENTATION

This section contains code for interfacing with the panel. The SPI port should be set up so that when in the idle state, the serial chip select (SCS) and the serial clock (SC) are LOW. Data is clocked on the rising edge of SC. Care should be taken to meet all of the timing such as Setup and Hold times and SC clock speed as given in your part's Specifications.

It is important to note which bit (MSB or LSB) gets shifted out of the SPI port first. The code in the example here is written for an ARM Cortex M3 processor which shifts the MSB out first. Because the memory LCD expects the LSB first, the bits within each byte need to be swapped before being sent. This is done using the "swap" routine.

Also, because the particular processor picked has a DMA controller, it relieves the processor load of writing individual bytes. The Cortex DMA unit disables itself after the transfer is complete, then generates an interrupt using the SPI vector. The SPI interrupt vector should point to the routine "show_frame". It is assumed that when show_frame is called from an interrupt, that the processor context has already been saved; and that it will be restored when show_frame exits.

The code can of course be modified if the processor picked doesn't have DMA or an SPI port.

```
// LCD commands - Note: the bits are reversed per the memory LCD data
// sheets because of the order the bits are shifted out in the SPI
// port.
#define MLCD_WR 0x80        //MLCD write line command
#define MLCD_CM 0x20        //MLCD clear memory command
#define MLCD_NO 0x00        //MLCD NOP command (used to switch VCOM)

//LCD resolution
#define MLCD_XRES 400   //pixels per horizontal line
#define MLCD_YRES 240   //pixels per vertical line
#define MLCD_BYTES_LINE  MLCD_XRES / 8 //number of bytes in a line
#define MLCD_BUF_SIZE    MLCD_YRES * MLCD_BYTES_LINE

//defines the VCOM bit in the command word that goes to the LCD
#define VCOM_HI 0x40
#define VCOM_LO 0x00

static char *frmbufter; //current address of buffer to be displayed
static char locbuf[MLCD_BYTES_LINE + 3];  //local line buffer
static char linenum;  //current line number being transmitted
static int stage = 0;  //there are 3 stages in transmitting a buffer:
                       //stage 0: first line (has command in it)
                       //stage 1: 2nd through last line (no command)
                       //stage 2: null byte trailer
extern char vcom;  //current state of vcom. This should alternate
                   //between VCOM_HI and VCOM_LO on a 1-30 second
                   //period.
```

```
////////////////////////////////////////////////////////////
//
//  This routine transmits the contents of the pixel memory in
//  a frame buffer to the memory LCD. It uses DMA to transfer
//  each individual line. The address of the frame buffer to
//  transfer is in the global variable show_frm.
//
//  NOTE: this routine also acts as the interrupt handler for SPI interrupts.
//
//  INPUTS:
//    The SPI and DMA units must have been previously initialized
//    show_frm:    pointer to the buffer to be displayed
//
////////////////////////////////////////////////////////////

void show_frame(char *show_frm) {
   int i;
   unsigned long sts;

   switch(stage) {
   case 0:   // stage 0: sending the first line. The command is
             // included in this line

      set_scs(HIGH);   //set SCS high

      frmbufptr = show_frm;   //init pointer to frame buffer
      linebuf = locbuf;   //init pointer to line buffer
      linenum = 1;   //init line address

      //first line of data is preceeded by a write command
      *linebuf++ = MLCD_WR | vcom;   //command (note: no need to swap)
      *linebuf++ = swap(linenum++);  //line number (address)

      for(i= 0; i < MLCD_BYTES_LINE; i++) *linebuf++ =
       swap(*frmbufptr++);  //swap the order of the bits
      *linebuf++ = 0;   //trailer

      //Setup the SPI DMA controller  (starting addr, size)
      TransferSetup(locbuf, linebuf - locbuf);

      //Start the xfer
      TransferStart;
```

```
      stage = 1;   //set to next stage
      break;

   case 1:   //Sending a line (other than the first line). At this
             //point the DMA transfer for the previous line is done
             //and the channel disabled.

      linebuf = locbuf;   //init pointer to line buffer
      *linebuf++ = swap(linenum++);  //line number

      for(i= 0; i < MLCD_BYTES_LINE; i++) *linebuf++ =
       swap(*frmbufptr++);  //swap the order of the bits
      *linebuf++ = 0;   //trailer

      //set up DMA control
      TransferSetup(locbuf, linebuf - locbuf);

      //Start the xfer
      TransferStart;

      if(linenum > MLCD_YRES) stage = 2;  //all lines sent, go to next
                                          //stage
      break;

   case 2:   //All lines sent, send a final null byte trailer. The DMA
             //transfer of the last line is finished and the channel
             //disabled. All that is left is to write a trailing null
             //byte. It's not worth using DMA to transfer 1 byte, so
             //it's done by directing writing to the SPI port.

      //Write the last (null) byte
      Write_to_SPI(0);

      //Since there is no interrupt on transmit buffer empty, loop
      //here until the byte is sent, then drop SCS.
      i = 0;
      while(SPI_BSY);  //wait until done

      set_scs(LOW);
      stage = 0;    //go back to stage 0 - sending out first line
   }
}
```

POWER CONSIDERATIONS

One of the prime attributes of the memory LCD is low power operation. The actual power usage is directly related to how often data is written to the panel, and how often VCOM is toggled. We'll look at two scenarios and calculate power draw.

POWER SCENARIO 1

Here all of the pixels will be written to the panel once per second. For the 2.7-inch panel, the quiescent current for the panel is 22 µA. Each write to the panel (assuming that the entire panel is written), takes 1.5 mA; and lasts for about 50 ms (assuming a 2 MHz SPI clock speed). See Figure 10.

If averaged over a second, this amounts to an "average" currant draw of approximately 75 µA. Therefore, the panel would have an average power draw of 485 µW (at 5 V). Note that a VCOM switch can be done whenever the panel is written to, including a data write.

POWER SCENARIO 2

Here, we write all the pixels to the panel once every 30 seconds. The current to write to the panel "averaged" over that time is approximately 3 µA. If VCOM is switched once a second (using the Change VCOM command), the additional power draw for this action (assuming that it is done 29 times) is an average of 3 µA. Therefore the average power draw over the 30-second period becomes 135 µW. See Figure 11.

Writing all pixels to the whole panel once a second takes an average of 485 µW. Writing all pixels once every 30 seconds (with a 1-second VCOM toggle) takes 135 µW.

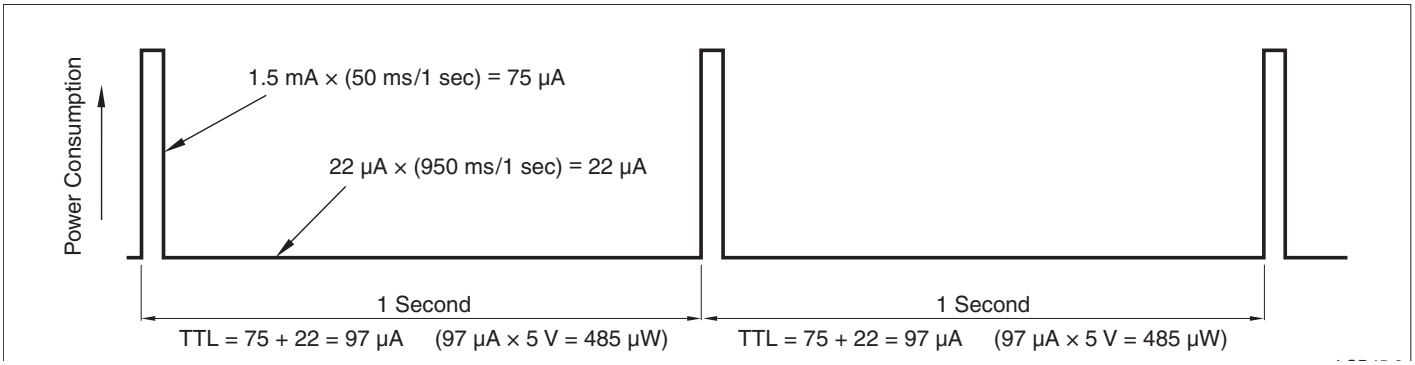

Figure 10. Writing All Pixels, Once per Second

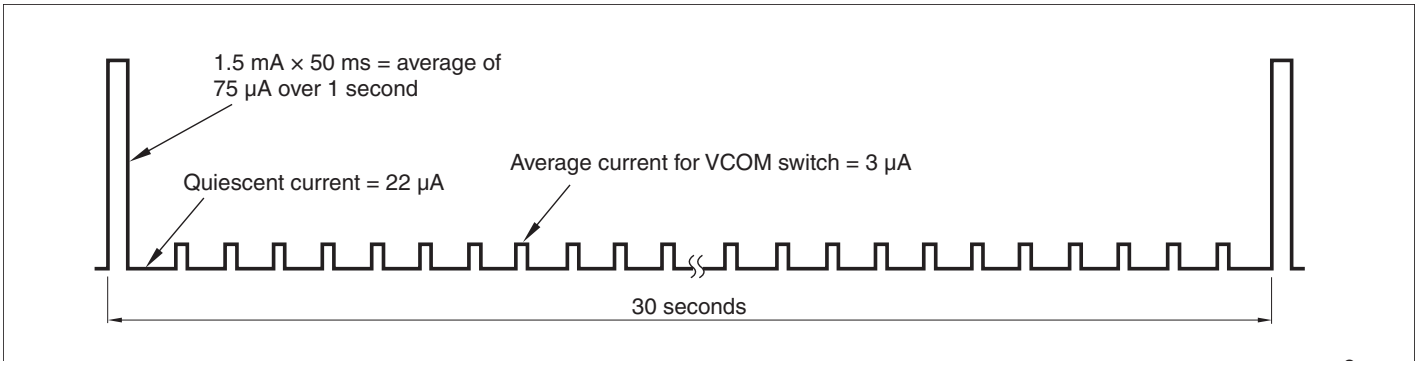

Figure 11. Writing All Pixels, Once Every 30 Seconds, Toggling VCOM Once Per Second

SHARP

SUMMARY

Sharp's Memory LCDs have a serial interface that makes them simple to program. The most challenging tasks for the programmer will be to ensure that VCOM is toggled periodically to maintain the lack of DC bias on the display; and that data is sent to the panel in the correct order.

NORTH AMERICA

SHARP Microelectronics of the Americas
5700 NW Pacific Rim Blvd.
Camas, WA 98607, U.S.A.

Phone: (1) 360-834-2500
Fax: (1) 360-834-8903

www.sharpsma.com

TAIWAN

SHARP Electronic Components
(Taiwan) Corporation
8F-A, No. 16, Sec. 4, Nanking E. Rd.
Taipei, Taiwan, Republic of China

Phone: (886) 2-2577-7341
Fax: (886) 2-2577-7326/2-2577-7328

CHINA

SHARP Microelectronics of China
(Shanghai) Co., Ltd.
28 Xin Jin Qiao Road King Tower 16F
Pudong Shanghai, 201206 P.R. China

Phone: (86) 21-5854-7710/21-5834-6056
Fax: (86) 21-5854-4340/21-5834-6057

Head Office:
No. 360, Bashen Road,
Xin Development Bldg. 22
Waigaoqiao Free Trade Zone Shanghai
200131 P.R. China

Email: smc@china.global.sharp.co.jp

EUROPE

SHARP Microelectronics Europe
Division of Sharp Electronics (Europe) GmbH
Sonninstrasse 3
20097 Hamburg, Germany

Phone: (49) 40-2376-2286
Fax: (49) 40-2376-2232

www.sharpsme.com

SINGAPORE

SHARP Electronics (Singapore) PTE., Ltd.
438A, Alexandra Road, #05-01/02
Alexandra Technopark,
Singapore 119967

Phone: (65) 271-3566
Fax: (65) 271-3855

KOREA

SHARP Electronic Components
(Korea) Corporation
RM 501 Geosung B/D, 541
Dohwa-dong, Mapo-ku
Seoul 121-701, Korea

Phone: (82) 2-711-5813 ~ 8
Fax: (82) 2-711-5819

JAPAN

SHARP Corporation
Electronic Components & Devices
22-22 Nagaike-cho, Abeno-Ku
Osaka 545-8522, Japan

Phone: (81) 6-6621-1221
Fax: (81) 6117-725300/6117-725301

www.sharp-world.com

HONG KONG

SHARP-ROXY (Hong Kong) Ltd.
3rd Business Division,
17/F, Admiralty Centre, Tower 1
18 Harcourt Road, Hong Kong

Phone: (852) 28229311
Fax: (852) 28660779

www.sharp.com.hk

Shenzhen Representative Office:
Room 13B1, Tower C,
Electronics Science & Technology Building
Shen Nan Zhong Road
Shenzhen, P.R. China

Phone: (86) 755-3273731
Fax: (86) 755-3273735