

## LCD DRIVING WITH ST6240

### INTRODUCTION

LCD panels are more and more used in a wide range of applications. The reasons of this success are their low consumption, the enhanced ergonomy they provide to the application, and the possibility to use customized graphical patterns on the panel.

Using a LCD driver MCU allows to take profit of the LCD panels features without having to worry about the complex dynamic control interface they require. The LCD driver MCU takes care of the timing control, while the user software selects which of the segment is highlighted: the LCD panel is managed as a data array in RAM memory.

This application note describes the basic guidelines to achieve a fast and efficient LCD drive application development. The alphanumeric LCD panel of the ST624X Starter-kit is used as example and more general concerns are highlighted. Hardware and software issues are described to demonstrate the benefits brought about while using a ST62 LCD driver.

### LCD INTERFACING DESCRIPTION

#### ST6240 LCD Driver Overview

**Note:** This is a quick summary of the features of the ST6240 LCD driver. Refer to the ST6240 data book for more detailed information.

The ST6240 LCD driver consists of LCD control logic, a programmable prescaler, a 24 bytes wide dedicated LCD RAM, 45 segments and 4 common outputs. This allows direct driving of up to 180 LCD segments.

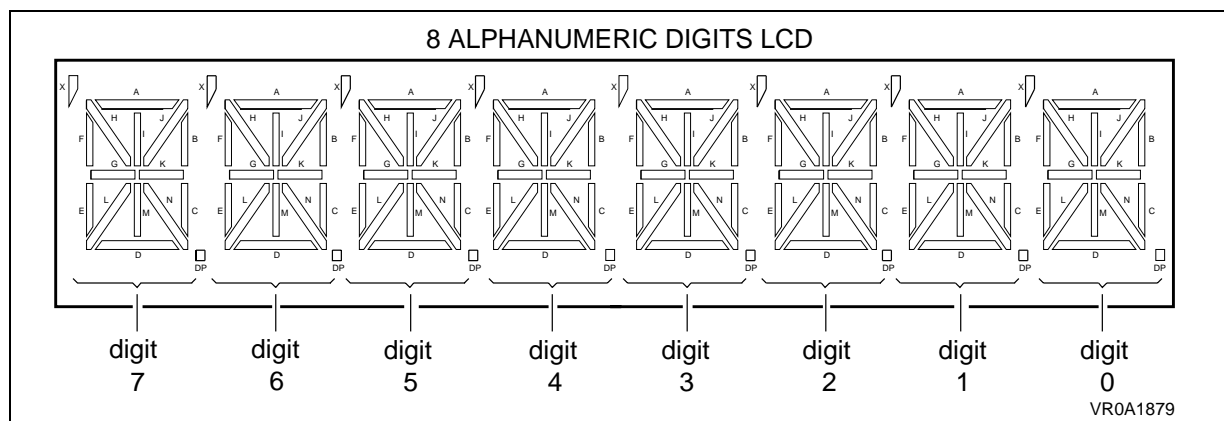
The LCD control logic operates automatically and without interruption of the processor.

Software configuration of the LCD driver is made by access to locations of the ST6240 DATA REGISTERS area:

**LMCR** (LCD Mode Control Register) allows the number of common backplanes (duty cycle) and the frame frequencies used by the LCD control to be set.

**LCD RAM** (24 bytes) is used to set each segment to "ON" or "OFF" state only by setting or resetting the corresponding bit. According to the contents of the LCD RAM, the drivers generate the segments and common signals which can directly drive an LCD panel.

**Figure 1. Display Layout**



**LCD INTERFACING DESCRIPTION (Continued)**

Hardware configuration requirements are reduced to minimum:

**VLCD** input pin must be fed with VLCD voltage (independently from  $V_{DD}$ ) according to the User's LCD panel specifications.

**VLCD1/3, VLCD2/3** pins are connected to intermediate VLCD voltages to enable external capacitive buffering or resistive shunting. An internal resistor network is implemented so that in most cases, no additional components are necessary.

This LCD control driver may be considered as a very easy to implement "black box", thus relieving User of hardware considerations.

To implement LCD panel interfacing, the User only has to:

- Determine the operating LMCR register value according to the LCD panel specifications.
- Generate the correct VLCD voltage and, if needed, connect an external resistor network on VLCD1/3, VLCD2/3 pins.
- Determine the most suitable mapping between the LCD panel segments and the LCD RAM bits.

Once this is done, controlling the LCD panel becomes as easy as usual software bit modification operations.

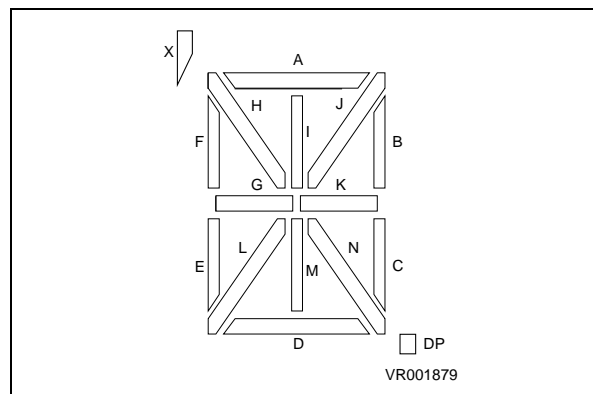
**Starter Kit LCD Panel Interfacing Description**

The following paragraphs describe in detail how the STARTER KIT LCD panel is interfaced to the ST6240 microcontroller.

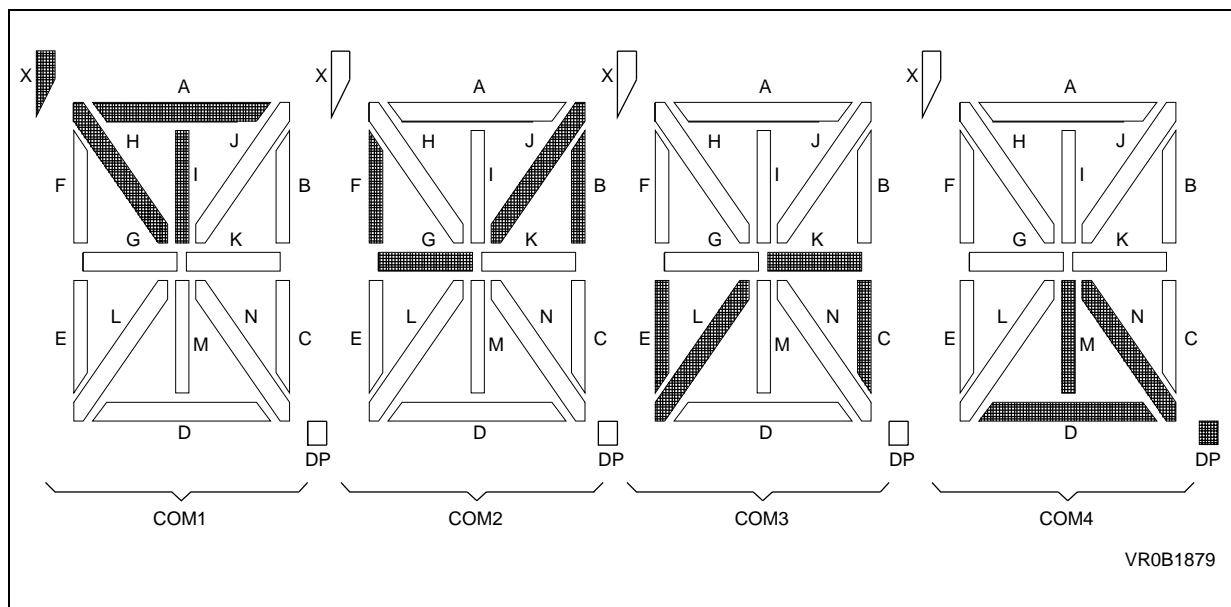
They may also be read as an example of simple LCD interfacing with information for User' own LCD control software development.

**"8 Alphanumeric Digits" LCD Panel Description**

**Figure 2. Digit Graphic Elements Layout**



**Figure 3. "ON" Graphic Segments According to Active COM**



**LCD INTERFACING DESCRIPTION** (Continued)

The LCD panel used on ST6240 STARTER KIT is an 8 alphanumeric digit display.

Each digit is based on a 16 segment matrix operating in multiplexed mode (1/4 duty) so that only 4 segments pins are required per digit.

This gives, for the 8 digits:

- 32 segments pins
- 4 commons pins

VLCD operating mode is 5V.

Digits are annotated from 0 to 7 respectively from right to left.

Each digit is composed by 16 graphic elements (A,B,C,D ....) as described in Figure 2.

Each elementary graphic segment can be set ON or OFF depending on the state of its dedicated Segment and Common lines, according to the boolean equation:

**graphic "ON" = Seg\_line "active" and Com\_line "active"**

Each DIGIT requires 4 SEG lines X 4 COM lines to be completely defined.

Table 3 describes the assignment between graphic elements set and the segment/common pins.

For example the elementary graphism 7J (J of digit 7) is driven by Segment line S2 (pin 2) and Common line C2 (pin 19). Other example: Pin 12 drives, for digit 2, the segments I,J,K,N when respectively commons C1,C2,C3,C4 are active.

**Table 1. Digit Matrix to Segment/COM Pins Mapping**

Pin	Segments				Pin	Segments				Pin	Segments			
1	7X	7F	7E	7D	13	1X	1F	1E	1D	25	2A	2B	2C	2DP
2	7I	7J	7K	7N	14	1I	1J	1K	1N	26	2H	2G	2L	2M
3	6X	6F	6E	6D	15	0X	0F	0E	0D	27	3A	3B	3C	3DP
4	6I	6J	6K	6N	16	0I	0J	0K	0N	28	3H	3G	3L	3M
5	5X	5F	5E	5D	17	.	.	.	CO4	29	4A	4B	4C	4DP
6	5I	5J	5K	5N	18	.	.	CO3	.	30	4H	4G	4L	4M
7	4X	4F	4N	4D	19	.	CO2	.	.	31	5A	5B	5C	5DP
8	4I	4J	4K	4N	20	CO1	.	.	.	32	5H	5G	5L	5M
9	3X	3F	3E	3D	21	0A	0B	0C	0DP	33	6A	6B	6C	6DP
10	3I	3J	3K	3N	22	0H	0G	0L	0M	34	6H	6G	6L	6M
11	2X	2F	2E	2D	23	1A	1B	1C	1DP	35	7A	7B	7C	7DP
12	2I	2J	2K	2N	24	1H	1G	1L	1M	36	7H	7G	7L	7M
<b>COM</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>COM</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>COM</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>

## LCD DRIVING WITH ST6240

### LCD INTERFACING DESCRIPTION (Continued)

#### Interfacing The LCD Panel with ST6240 LCD Driver - Characters Definition

No precise rule can be provided concerning the methodology that will lead to the LCD interfacing chart; it depends on the LCD panel used.

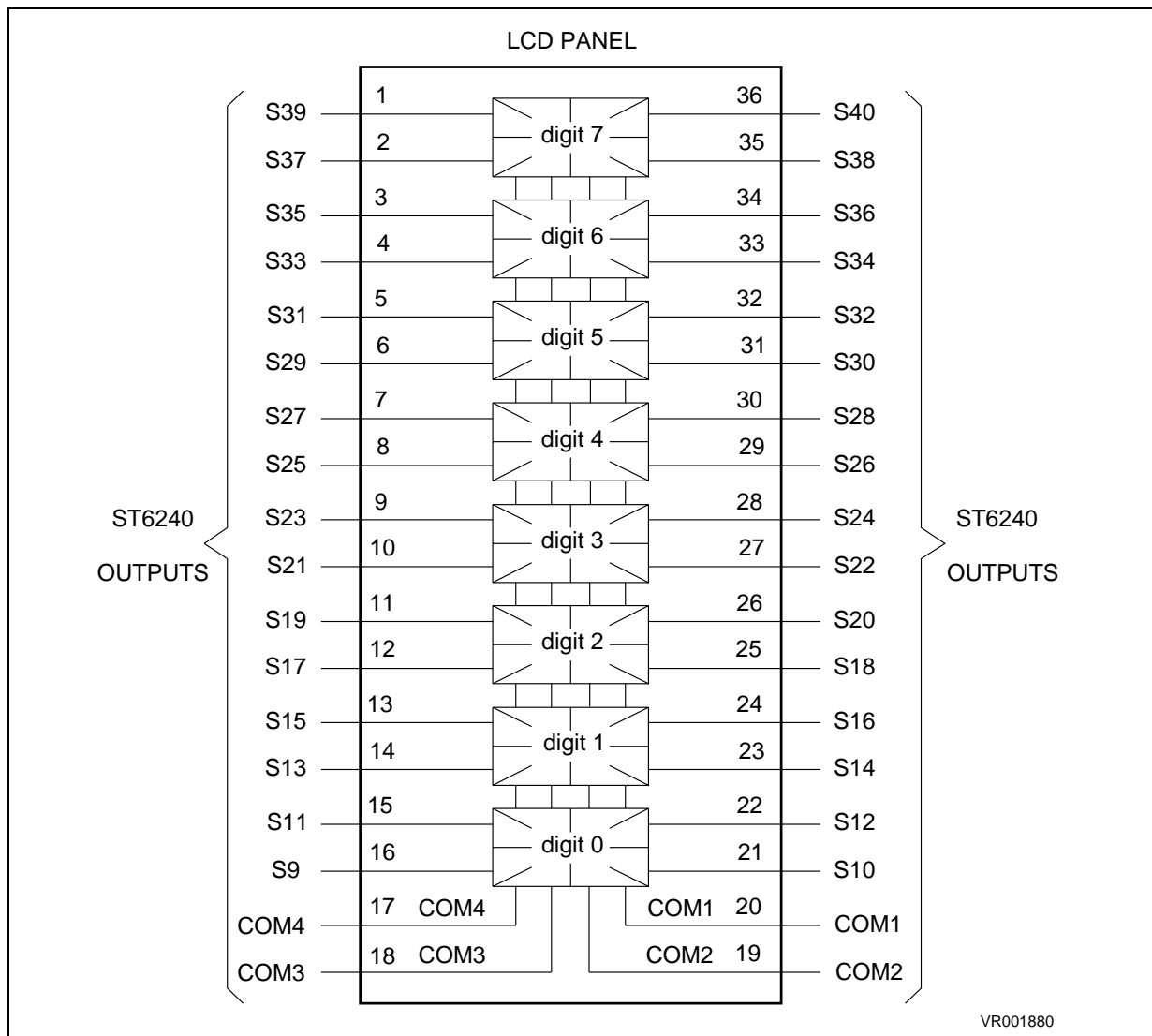
But beyond any other considerations, the User is strongly advised to think about the application software architecture before definitively deciding on the mapping of the LCD segments.

Assigning the LCD panel segments to the LCD RAM bits must be made with the aim of generating the most "user friendly" software description of the

graphics that will be displayed on the panel. For instance, for this LCD panel, the graphics are shown to be alphanumeric characters. The description map of each character in LCD RAM should be the same, whichever digit position it is displayed at. The contrary would increase the complexity of application software .

The LCD panel requires 32 segments lines multiplexed through 4 Common lines. This is less than the capabilities of the ST6240. S9 to S39 segments and COM1,2,3,4 outputs of ST6240 LCD driver are connected to the LCD panel inputs as described in Figure13 (see also the board scheme).

Figure 4. LCD Panel Pinout



**LCD INTERFACING DESCRIPTION** (Continued)

This connection array leads to the digit organisation in the LCD RAM described in Table 4. Notice that the memory is not entirely used.

Each digit is represented by 4 bits (MSB for odd digits, LSB for even digits) in each COM memory area.

These 4 bits x 4 coms = 16 bits are assigned to the 16 defined above as described in Table 5. This description is valid for each digit.

This table format can be useful to define the digit character set. Each character displayable on the LCD panel is completely defined by a pattern (0-1) of the 16 bit table.

**Table 2. Digit Locations in LCD RAM**

	LCD RAM Address	D7	D6	D5	D4	D3	D2	D1	D0
<b>COM1</b>	E1h	Digit 1				Digit 0			
	E2h	Digit 3				Digit 2			
	E3h	Digit 5				Digit 4			
	E4h	Digit 7				Digit 6			
<b>COM2</b>	E7h	Digit 1				Digit 0			
	E8h	Digit 3				Digit 2			
	E9h	Digit 5				Digit 4			
	EAh	Digit 7				Digit 6			
<b>COM3</b>	EDh	Digit 1				Digit 0			
	EEh	Digit 3				Digit 2			
	EFh	Digit 5				Digit 4			
	F0h	Digit 7				Digit 6			
<b>COM4</b>	F3h	Digit 1				Digit 0			
	F4h	Digit 3				Digit 2			
	F5h	Digit 5				Digit 4			
	F6h	Digit 7				Digit 6			

**Table 3. LCD RAM Bits Mapping**

	Bit 7/3	Bit 6/2	Bit 5/1	Bit 4/0
<b>COM1</b>	H	X	A	I
<b>COM2</b>	G	F	B	J
<b>COM3</b>	L	E	C	K
<b>COM4</b>	M	D	DP	N

LCD INTERFACING DESCRIPTION (Continued)

The following examples describe this method.

**Example 1: Character "A" definition**

Figure 5 represents the "A" character to be displayed on the LCD panel. The "ON" are shown darkened and the "OFF" ones are left blank.

Table 6 gives the corresponding bit pattern to be set into the LCD ram to see the character A displayed on the LCD panel. "0" bits locations are left blank, "1" bits are shadowed. The right column gives the corresponding hexadecimal value.

This description is valid for any number of digit. The position of the displayed character depends only on the addresses and the position (LS/MS nibble) where the 4 x 4 bits set are written as described in Table 4.

**Example 2: Character "3" definition**

In the same way, Figure 6 and Table 7 represents the definition of the "3" character .

This method can be used to define all the character set to be displayed on the LCD panel.

The set of 4 x 4 bits (representing the character set) can be stored into the ST624x DATA ROM and easily used by display software through indexed accesses.

A blank digit array and a bit map table are available on the next page to be used for personal character definition.

Figure 5. Character A

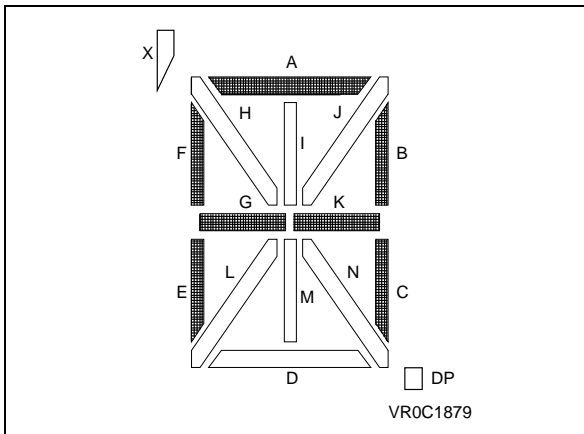


Figure 6. Character 3

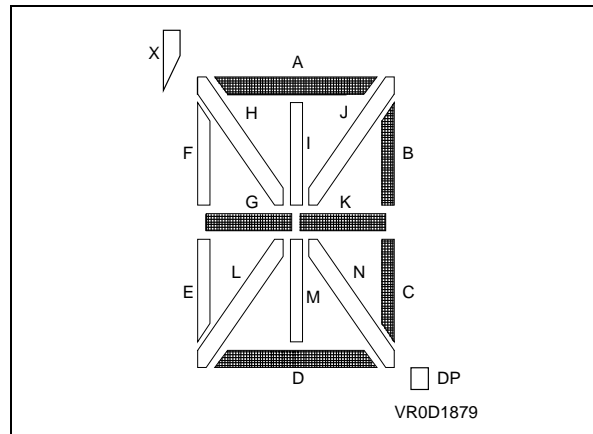


Table 4. Character A Definition Bit Map

	D3	D2	D1	D0	HEX VAL
COM1	H	X	A	I	2
COM2	G	F	B	J	E
COM3	L	E	C	K	7
COM4	M	D	DP	N	0

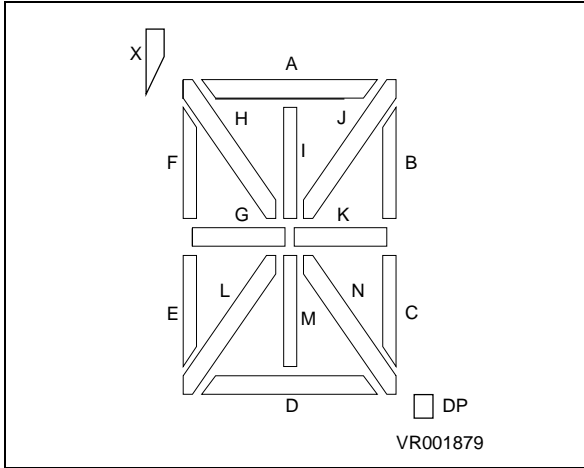
Table 5. Character 3 Definition Bit Map

	D3	D2	D1	D0	HEX VAL
COM1	H	X	A	I	2
COM2	G	F	B	J	A
COM3	L	E	C	K	3
COM4	M	D	DP	N	4

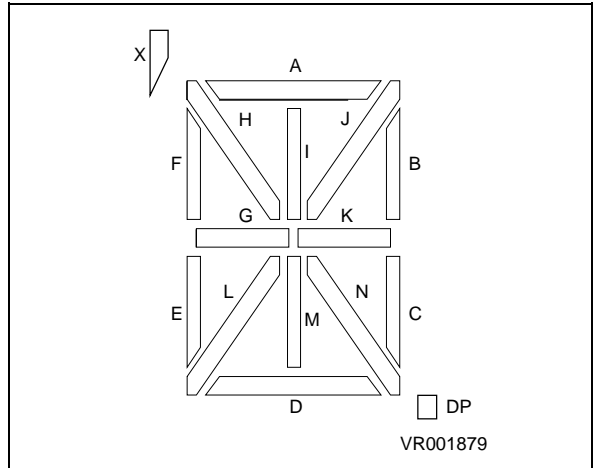
LCD INTERFACING DESCRIPTION (Continued)

User Character Definition Chart  
For LCD Provided with the ST6240 Starter Kit

User Character Definition:



User Character Definition:



	D3	D2	D1	D0	HEX VAL
COM1	H	X	A	I	
COM2	G	F	B	J	
COM3	L	E	C	K	
COM4	M	D	DP	N	

	D3	D2	D1	D0	HEX VAL
COM1	H	X	A	I	
COM2	G	F	B	J	
COM3	L	E	C	K	
COM4	M	D	DP	N	

Shade block corresponding to the segments used and enter the resulting hex number in the last column.

### LCD INTERFACING DESCRIPTION (Continued)

#### Starter Kit LCD Panel Character Set Software Model.

As described in the previous paragraphs, LCD connection and bit mapping definition of the characters set are driving a "software model" of the LCD displayable objects.

These objects are available for DISPLAY routines to simply perform LCD panel display during program execution.

As examples of display software organisation, please read the following lines. These methods are illustrated in the "ST6240 STARTER KIT EXERCISES".

#### 1- Direct Code LCD RAM Patching

This is the most simple way to perform the display of a digit. The bit map values of character are directly patched into the corresponding LCD RAM locations. The bit map value are contained in the immediate values of write instructions. (Example 1).

This method should be used only for very simple applications (few characters, displayed few times) or just for rapid evaluation.

#### 2- Indexed Data Rom Use

The bit-map values of characters set are defined into ST6240 DATA ROM AREA (refer to ST6240 and AST6/LST6 data books). The `.byte`, `.w`, `.d` directives of AST6/LST6 tools provide definition and access to DATA ROM area (refer to the appropriate documentation).

A generic display call subroutine can access to the character bit definition map through parameter from the calling program.

This, of course, is an illustration example of the principle. The described `disp_0` subroutine only displays the character at fixed digit 0 and at the same time clears digit 1 (MSB of the same LCD RAM locations).

#### Example 1. Display character A at digit 0 (see Figure 3 and Figure 1)

```
disp_A_0    ldi E1,x2          ; value 2 into digit0 COM1 add. (E1,LSB)
            ldi E7,xE          ; value E into digit0 COM2 add. (E7,LSB)
            ldi ED,x7          ; value 7 into digit0 COM3 add. (ED,LSB)
            ldi F3,x0          ; value 0 into digit0 COM4 add. (F3,LSB)
```

x is the value of MSB (digit 1)

#### Example 2. Display character A at digit 0 (see Figure 3 and Figure 1)

##### Definition in DATA ROM:

```
CHAR_A.    .byte 02
            .byte 0E
            .byte 07
            .byte 00
```

##### Display routine:

```
disp_A_0    ldi RWSR,CHAR_A.w    ; DATA ROM WINDOW register is set
            ldi X,CHAR_A.d        ; X index point to CHAR_A first byte.
            call disp_0           ; generic subroutine that displays at 0 digit

disp_0      ld E1,(X)             ; value 2 into digit0 COM1 add. (E1,LSB)
            inc X                 ; X points to second byte
            ldi E7,(X)           ; value E into digit0 COM2 add. (E7,LSB)
            inc X                 ; X points to third byte
            ldi ED,(X)           ; value 7 into digit0 COM3 add. (ED,LSB)
            inc X                 ; X points to fourth byte
            ldi F3,(X)           ; value 0 into digit0 COM4 add. (F3,LSB)
```



**LCD INTERFACING DESCRIPTION (Continued)**

An efficient generic display subroutine should manage the digit number (as input parameter) and perform the Read/Modify/Write only on the nibble concerned of the LCD RAM byte (MSB or LSB).

This is done by the “display” subroutine delivered with the ST6240 STARTER KIT LIBRARY.

(Please read the README file of the SK6240LIB directory).

**Note:** A character requires 4 x 4 bits words = 16 bits to be completely defined. To optimize the character set size in DATA ROM, it is useful to compact these 16 bits into 2 bytes, however this method would slightly increase the display routine complexity.

For ST6240 STARTER KIT demo routines, the character set is defined as follows (for example character “A”):

```
CHAR_A.      .byte 22
             .byte EE
             .byte 77
             .byte 00
```

4 bytes are used to define a character, with MSB and LSB equal to the 4 bit definition values. By this way, discrimination between MSB and LSB is easily done by using the AND instruction for masking the non-relevant nibble.

**3- Complete Message Display**

To define complete messages, it is interesting to use the **ASCIZ** directive of **AST6/LST6** software tools.

```
e.g.      MESS .ASCIZ “HELLO”
```

This directive returns the ASCII code pattern of the characters of message (ended by NULL).

In order to use this possibility, the ST6240 character set is implanted into DATA ROM so that it becomes simple to reach a definition bit-map through the character ASCII code.

**Principle:** One DATA ROM window contains 64 bytes. As a character definition bit map requires 4 bytes, 16 characters can be defined in a whole window. This corresponds to a page of ASCII codes (represented by the MSB of ASCII code).

For example characters @,A,B ...,O whose ASCII codes are respectively (Hex) 40,41,42 ...,4F represent the ASCII code page 4, and fill a complete DATA ROM window as in Example 3.

For example, the ASCII code of character A is 41 (hex):

- 4 is the number of the window
- 1 indexes the number of the character in the window

The displayable (not all the ASCII code Chart) CHARACTER SET delivered with the ST6240 STARTER KIT LIBRARY satisfies this principle. This allows the `asci_dis` subroutine to display a character defined by its ASCII code and `dis_mes` subroutine to display on LCD panel a complete message built with the ASCIZ AST6 directive.

Refer to SK6240LI.INI and SK6240LI.ASM files in the SK6240LIB directory for description of these subroutines.

**Example 3. ASCII Characters in Data ROM Window**

```

                                .ORG x00      ; beginning of a window
0      CHAR_@ .byte 22      ; Address 0 in the window
             .byte AA
             .byte BB
             .byte 44
1      CHAR_A .byte 22      ; Address 3 in the window
             .byte EE
             .byte 77
             .byte 00
.      .      .
.      .      .
.      .      .
F      CHAR_O .byte 22      ; Address 3C in the window
             .byte 66
             .byte 66
             .byte 44      ; Last address (3F) in the window

```

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of SGS-THOMSON Microelectronics.

© 1995 SGS-THOMSON Microelectronics - All rights reserved.

Purchase of I<sup>2</sup>C Components by SGS-THOMSON Microelectronics conveys a license under the Philips I<sup>2</sup>C Patent. Rights to use these components in an I<sup>2</sup>C system is granted provided that the system conforms to the I<sup>2</sup>C Standard Specification as defined by Philips.

SGS-THOMSON Microelectronics Group of Companies

Australia - Brazil - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands  
Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.