LAPIS SEMICONDUCTOR

SQ003116E004

ML610Q400 Series

Sample Program AP Notes For LCD Application

3rd Edition
Issue Date: April 16, 2010

NOTICE

No copying or reproduction of this document, in part or in whole, is permitted without the consent of LAPIS Semiconductor Co., Ltd.

The content specified herein is subject to change for improvement without notice.

The content specified herein is for the purpose of introducing LAPIS Semiconductor's products (hereinafter "Products"). If you wish to use any such Product, please be sure to refer to the specifications, which can be obtained from LAPIS Semiconductor upon request.

Examples of application circuits, circuit constants and any other information contained herein illustrate the standard usage and operations of the Products. The peripheral conditions must be taken into account when designing circuits for mass production.

Great care was taken in ensuring the accuracy of the information specified in this document. However, should you incur any damage arising from any inaccuracy or misprint of such information, LAPIS Semiconductor shall bear no responsibility for such damage.

The technical information specified herein is intended only to show the typical functions of and examples of application circuits for the Products. LAPIS Semiconductor does not grant you, explicitly or implicitly, any license to use or exercise intellectual property or other rights held by LAPIS Semiconductor and other parties. LAPIS Semiconductor shall bear no responsibility whatsoever for any dispute arising from the use of such technical information.

The Products specified in this document are intended to be used with general-use electronic equipment or devices (such as audio visual equipment, office-automation equipment, communication devices, electronic appliances and amusement devices).

The Products specified in this document are not designed to be radiation tolerant.

While LAPIS Semiconductor always makes efforts to enhance the quality and reliability of its Products, a Product may fail or malfunction for a variety of reasons.

Please be sure to implement in your equipment using the Products safety measures to guard against the possibility of physical injury, fire or any other damage caused in the event of the failure of any Product, such as derating, redundancy, fire control and fail-safe designs. LAPIS Semiconductor shall bear no responsibility whatsoever for your use of any Product outside of the prescribed scope or not in accordance with the instruction manual.

The Products are not designed or manufactured to be used with any equipment, device or system which requires an extremely high level of reliability the failure or malfunction of which may result in a direct threat to human life or create a risk of human injury (such as a medical instrument, transportation equipment, aerospace machinery, nuclear-reactor controller, fuel-controller or other safety device). LAPIS Semiconductor shall bear no responsibility in any way for use of any of the Products for the above special purposes. If a Product is intended to be used for any such special purpose, please contact a ROHM sales representative before purchasing.

If you intend to export or ship overseas any Product or technology specified herein that may be controlled under the Foreign Exchange and the Foreign Trade Law, you will be required to obtain a license or permit under the Law.

Table of Contents

1. Overview

This document describes the application programming notes (hereafter called the AP notes) arranged to help customers develop software that, by using the LCD driver, which are hardware that the ML610Q400 Series MCU (hereafter called the MCU) has, performs to display characters on LCD panel.

This AP note describes how to make program for LCD panel by using LCD Image Tool.

In connection with the AP notes, a sample program is provided that actually operates by using LCD Image Tool.

◆ Related Documents

The following are the related documents. Read them as required.

・LCD Image Tool User's Manual
・ML610Q400 Series Sample Program API Manual
・ML610Q431/ML610Q432 User's Manual
・ML610Q411/ML610Q412/ML610Q415 User's Manual
・ML610Q421/ML610Q422 User's Manual
・ML610Q482 User's Manual
・ML610Q435/ML610Q436 User's Manual
・ML610Q400 Series Demo Kit Hardware User's Manual
・nX-U8/100 Core Instruction Manual
・MACU8 Assembler Package User's Manual
・CCU8 User's Manual
・CCU8 Programming Guide
・CCU8 Language Reference
・DTU8 User's Manual
・IDEU8 User's Manual
・uEASE User's Manual
・uEASE Connection Manual ML610Qxxx
・FWuEASE Flash Writer Host Program User's Manual

1.1. Hardware Configuration

The operating environment of this sample program is composed of LCD panel and control MCU. Regarding LCD panel, LCD Image Tool, which generates a imaginary LCD panel on PC, is used. This tool can make LCD panel layout on PC and display the image data (DSPR data in MCU), which is outputted in a file by DTU8, as the LCD picture image.

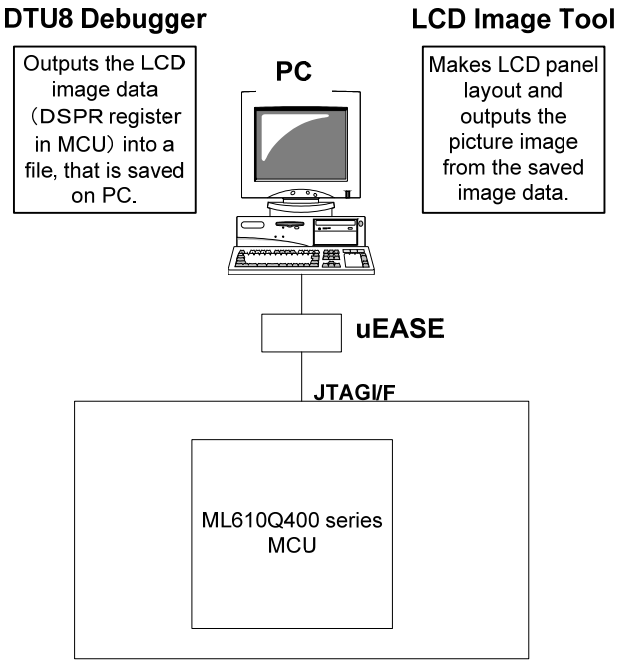


Figure 1.1_1 Hardware system configuration

1.2. Software Configuration

The folloing shows the software configuration.

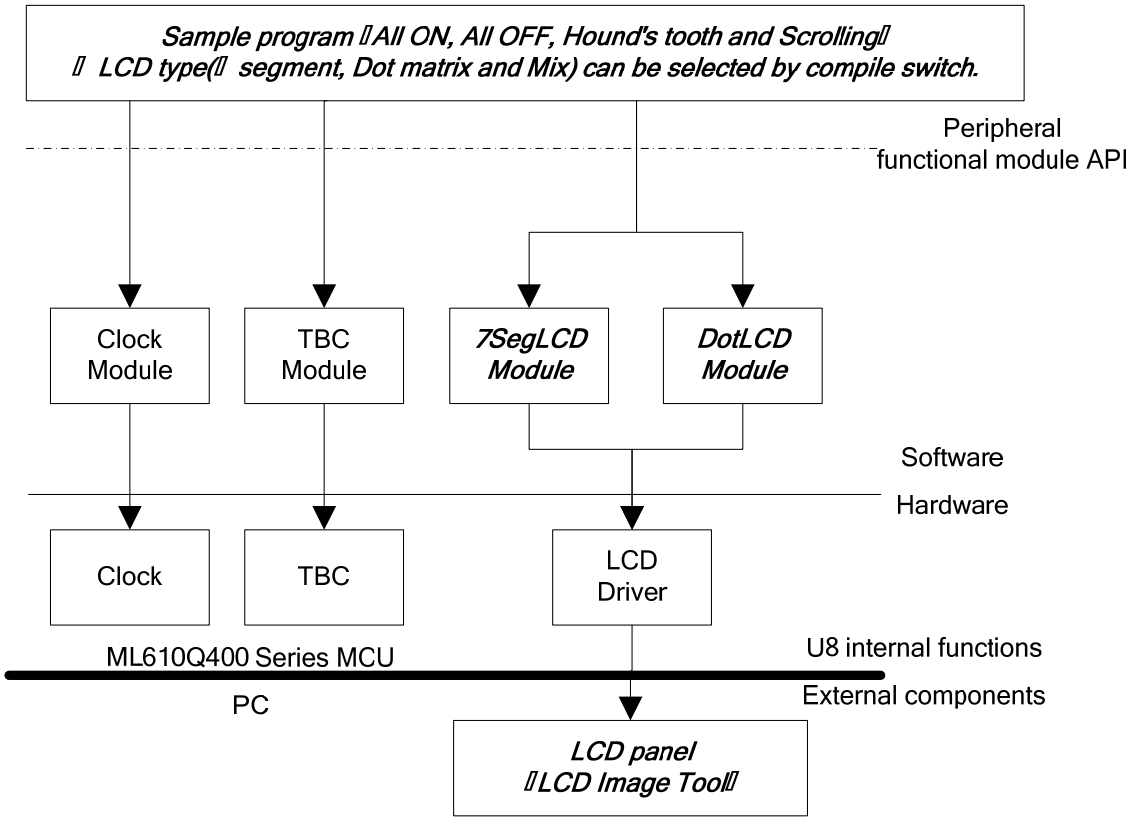


Figure 1.2_1 Software configuration

1.3. List of Folders and Files

The folders and the files are as listed below.

```
[lcd]
├ [_output]                  … Build result output folder
│  ├ [_hex]
│  ├ [_lst]
│  ├ [_obj]
│  └ [_prn]
├ [clock]                    … Clock control module folder
│  ├ clock.c
│  ├ clock.h
│  ├ clock_sysFunc.c
│  └ clock_sysFunc.h
├ [common]                   … General-purpose function module folder
│  ├ common.c
│  └ common.h
├ [irq]                            … Interrupt control module folder
│  ├ irq.c
│  └ irq.h
├ [lcd]                            … LCD display control module folder
│  ├ Lcd_drv.c
│  ├ Lcd_drv.h
│  ├ Lcd_FontTbl.c
│  ├ Lcd_FontTbl.h
│  ├ Lcd_ImgType1.c
│  ├ Lcd_ImgType1.tac
│  ├ Lcd_ImgType1.tbc
│  ├ Lcd_ImgType3.c
│  ├ Lcd_ImgType3.tac
│  ├ Lcd_ImgType3.tbc
│  ├ Lcd_main.c
│  ├ Lcd_main.h
│  ├ Lcd_setting.h
│  ├ Lcd_TP_Font.c
│  └ Lcd_TP_Font.h
├ [main]                     … Sample program main folder
│  ├ [mcu_large]
│  │  └ mcu.h
│  ├ [mcu_small]
│  │  └ mcu.h
│  ├ main.c
│  ├ main.h
│  ├ S610431SW.asm
│  └ S610435LW.asm
├ [tbc]                            … Time base counter control module folder
│  ├ tbc.c
│  └ tbc.h
├ [timer]                    … Timer control module folder
│  ├ timer.c
│  └ timer.h
├ lcdImgAtU8_L1.ld8
├ lcdImgAtU8_L2.ld8
├ lcdImgAtU8_L3.ld8
├ test_lcdsample.mcr
├ U8_Lcd_Sample_Large.PID    … Project file for large model MCU
└ U8_Lcd_Sample_Small.PID    … Project file for small model MCU
```

1.4. Build Procedure

① Start IDEU8, select the menu “Open” and open the project file (PID file). In the case that MCU memory model is small model, the project file is “U8_Lcd_Sample_Small.PID”. In the case of large model, the project file is “U8_Lcd_Sample_Large.PID”. Correspondence of MCU and PID file is shown below.

Table 1.4_1 Correspondence of MCU and PID file

	U8_Lcd_Sample_Small.PID	U8_Lcd_Sample_Large.PID
Supported MCU	ML610Q431/432	ML610Q435/436

② In the default setting, ML610Q431 is set as the target MCU.
If your target MCU is different, follow the procedure below to change the setting.

(1) Select the menu “Project” -> “Options” -> “Compiler/assembler”.
(2) In the displayed window, select the target MCU from the “Target microcontroller” list in the “General” tab.
(3) Remove the startup file “S610431SW.asm“ registered in the file tree of IDEU8. Instead of that, register your target MCU’s startup file. (In the case of ML610Q432, it is S610432SW.asm.)
(4) Define the macro that represents the target MCU.
Select the menu “Project” -> “Options” -> “Compiler/assembler” -> ”Macro”tab. In the displayed window, modify the macro like following name.

_ML610Q4XX

About the “XX” part, replace with the type number of MCU
For example, if ML610Q432 is used, define the following macro.

_ML610Q432

In the case that the macro other than the type number in the above Table 1.4_1 is defined, the case that macro such as above is not defined, or the case that the memory model that is supported by PID file is different from the memory model of MCU that is defined by the above macro, the compiler issues the following error at the beginning of the output messages.

Error : E2000 : #error : “Unknown target MCU”

③ If you want to run the sample program on the simulator, after this, operate according to the section “5 How to run the sample program”.

④ Select the menu “Project” -> “Options” -> “Compiler/assembler” -> ”Macro”tab. In the displayed window, modify the macro LCD_TYPE. In the case that the type of LCD is 1, 2 or 3, define LCD_TYPE as 0, 1, or 2, respectively. About the type of LCD, see the section “4.2 Definition”.

⑤ Select the menu “Project” -> “Rebuild”. Then the build procssing for the sample program starts.

⑥ When the build processing is completed, .abs file is generated in the project folder and .hex file is generated in _output¥_hex folder.

1.5. Restrictions

1.5.1. About Available Functional Modules

In the functional modules that compose this sample program, the available functional modules are different by target MCU, due to the difference of MCU peripherals. In the case that these functional modules are applied to user application, available functional modules on each MCU are shown below.

Table 1.5.1_1 List of available functional modules

		Supported MCU			
		ML610Q43X	ML610Q42X	ML610Q41X	ML610Q48X
Functional modules *3	LCD Display Control Module	○	○ *1	○ *1	×
	Clock Control Module	○	○	○	○
	Time Base Counter Control Module	○	○	○	○

○ : Available
× : Not available

*1: All display area of LCD panel can not be available, because the number of SEG pin that is connected to LCD panel is not enough.
*2: For the details of these modules, please see the "ML610Q400 Series Sample Program AP Notes For Sensor/Mesurement Application".

1.5.2. About Functions of Sample Program

This sample program is available on only ML610Q43X series MCU.

Note: This sample program is not available on ML610Q42X and ML610Q41X series MCU, because LCD driver does not support the number of LCD's COM/SEG pin which is required for operating the application. Also this sample program is not available on ML610Q48X series MCU, because it does not have LCD driver.

2. Mechanism of LCD Driver

This section describes the outline of LSI's operation to turn on and off patterns, displayed on LCD. In detail, please see the User's Manula of your target LSI.

The LCD driver provides the following types of contorol methods, depending on the specification of the LCD panel to be used.

■ **Fixed display allocation**
The mapping of the display register and the common/segment pin, that is described later, is already fixed.

■ **Programmable display allocation**
By using LSI's programmable display allocation function, a user defines the mapping of the display register and common/segment pin.

There are three types of registers which are used to control the turn-on/off for the displayed patterns on LCD.

▪ **Display register**
The display register is used to store the contents to be displayed to LCD as bit patterns.

▪ **Display allocation register**
The display allocation register is used to control the mapping of the display register for the LCD common/segment pin. This register can be used when the programmable display allocation is selected. If the fixed display alloction is selected, it can be used as data memory space.

▪ **Display control register group**
The display control register group is used to control output of the on waveform to the common/segment pin, bias, duty and contrast.

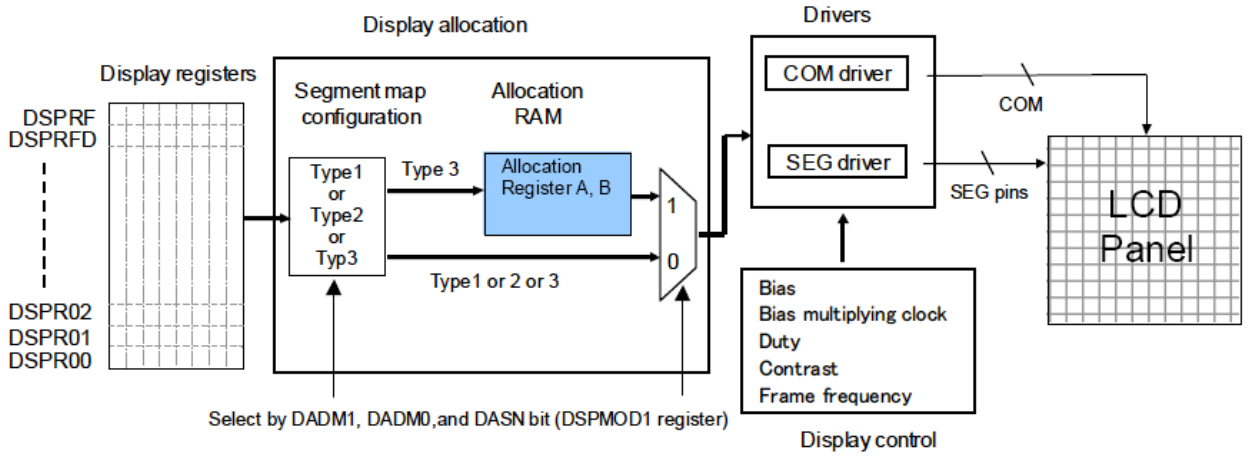


Figure 2-1 Configuration of LCD Display Function

To make the program which controls the turn-on/off for the displayed patterns on LCD, follow the procedure below.

① Determine the control method that meets the specification of the LCD panel to be used.
② Set the application specific configuration to the display control registers.
③ If the programmable display allocation, define the display allocation register.
④ Set the the contents to be displayed into the display register.
After this, change the setting in the procedure ② and ④, if necessary.

[Important]
In the procedure ③, define the display allocation register on the condition that DASN bit is 0. When DASN bit is 1, read/write operations from CPU are invalid.
Also, when using the programmable display allocation, select type 3 for the display register segment map.

3. LCD Operation Summary

This section introduces an example for the development of LCD control software by using LCD Image Tool. About the detail of each step that is described below, see the section 4.

3.1. Preparation

Determine the segment name, COM/SEG pin and DSPR which are mapped to each segment of LCD panel.
And write the result to a work sheet.

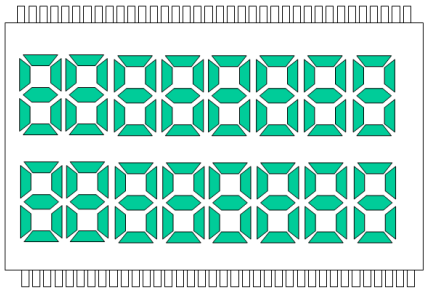
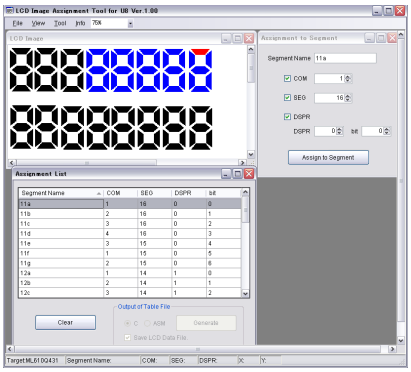


Table 3-1 Work sheet

Segment name	COM	SEG	DSPR	bit
11a	1	16	0	0
11b	2	16	0	1
11c	3	16	0	2
11d	4	16	0	3
11e	3	16	0	4
11f	2	16	0	5
11g	1	16	0	6

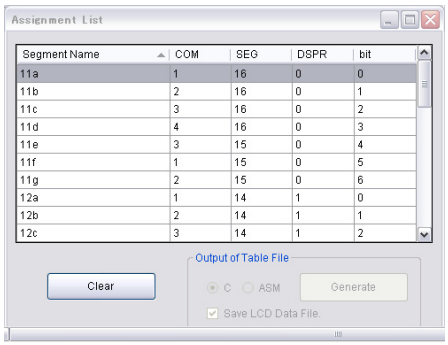
3.2. Definition

Input the contents of the work sheet into LCD Assignment Tool.



3.3. Confirmation

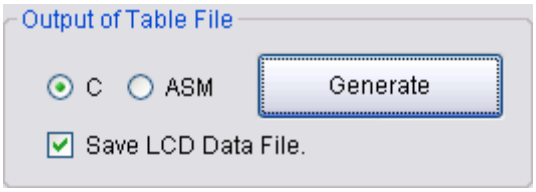
Confirm whether the input of LCD Assignment Tool is correct, compared with the work sheet.



Segment name	COM	SEG	DSPR	bit
11a	1	16	0	0
11b	2	16	0	1
11c	3	16	0	2
11d	4	16	0	3
11e	3	16	0	4
11f	2	16	0	5
11g	1	16	0	6

3.4. Programming

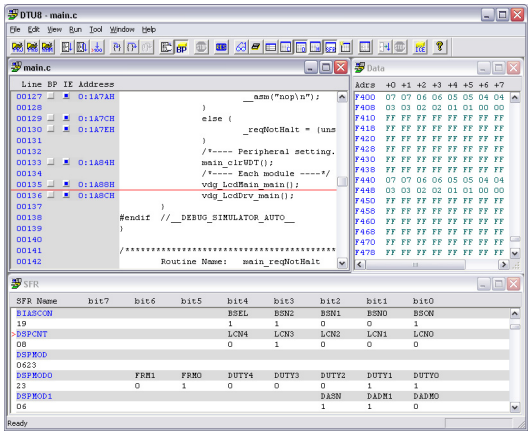
If the programmable display allocation is used, generate the table for the display allocation register.

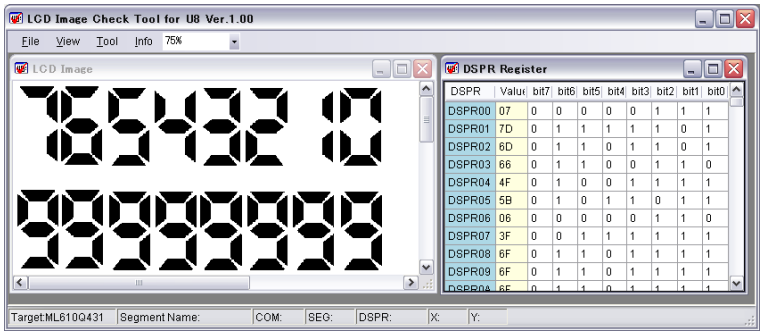


Clicking the “Generate” button, the table for the display allocation register A and B is automatically generated.

3.5. Debug

Working with DTU8 debugger, Confirm the displayed contents of LCD panel,





The result of program execution can be displayed to LCD Image Check Tool.

4. LCD Operation Details

This section describes the details of each step in the program development procedure that is introduced in the section 3.

4.1. Preparation

This subsection introduces the segment naming rule, the pin assignment and DSPR arrangement, which make programming easy, by giving the implementation of the sample program as a example.

4.1.1. How to determine the segment name

Make the naming rule for the segment.
In the sample program, the segment name is determined by the following naming rule.

- In the case of 7 segment, the display parts of a character are named clockwise a, b, c, …, g from the top part.
  This goes for also 16 segment, but outer parts are a, b, c, …, h and inner parts are i, j, k, …, p.
  * Name the segments for each clump of seguments, so that the segments can be identified easily when they are sorted by name.
- In the case of dot matrix, name each dot by its row and column values, from the upper left to the lower right.

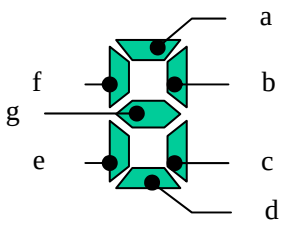


Figure 4.1.1_1 Segment name of 7SEG

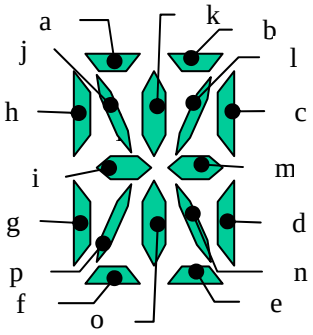


Figure 4.1.1_2 Segment name of 16SEG

4.1.2. How to determine COM/SEG pin

When using the fixed display allocation, the hardware allocates SEG pin to DSPR address, and allocates COM pin to a bit in DSPR address. Then, please allocate COM pin and SEG pin so that the display data is easily handled as a particular unit (e.g. 7 segment) by software. When using the programmable display allocation, we can determin the allocation for these pins as we like, so that we can wire the segments easily.(But, the number of available COM pin becomes 8.)

4.1.3. How to determine the DSPR arrangement

When using the fixed display allocation, SEG pin is allocated to DSPR address, and COM pin is allocated to a bit in DSPR address. When using the programmable display allocation, DSPR can be arrenged as we like, so that we can handle the display data easily by software. For example, we can collect display data as a clump (e.g. 7 segment or dot matrix) so that the display data can be handled easily, and arrange DSPR to the direction which the display character controled to.

When the segment name, COM/SEG pin and DSPR arrangement are determined, write the result into the work sheet like Table 4.1.3_1.

Table 4.1.3_1 Work sheet

Segment name	COM	SEG	DSPR	bit
11a	1	16	0	0
11b	2	16	0	1
11c	3	16	0	2
11d	4	16	0	3
11e	3	16	0	4
11f	2	16	0	5
11g	1	16	0	6

4.2. Definition

This sample program can display the following 3 types of LCD panel, by changing the definition of macro.

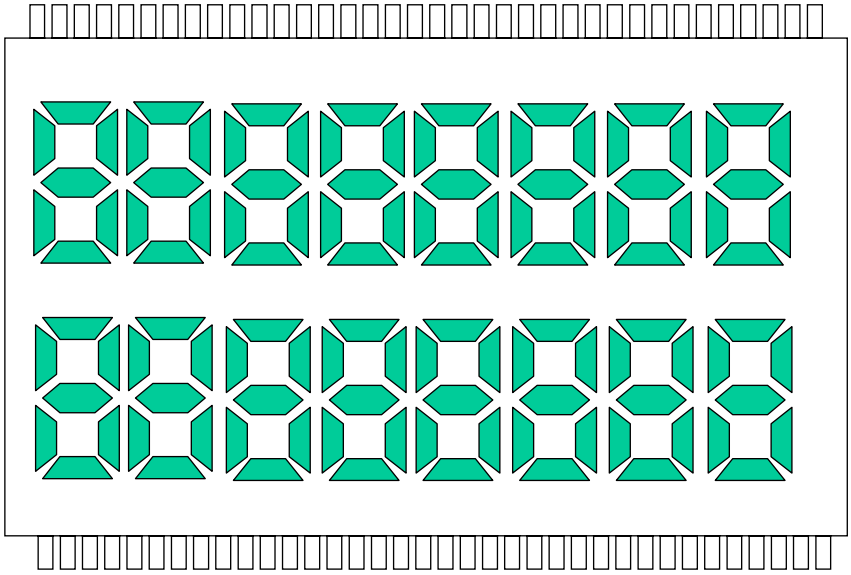
- Type 1 : 7 segment type LCD panel
- Type 2 : Dot matrix type layout
- Type 3 : Mixed type layout (mixture of segment and dot matrix)

This section shows the 3 types of LCD panel layout and mapping, which are prepared and defined for the sample program.

After the definitions are written in the work sheet according to the procedure that is introduced in the section 4.1, define them by using LCD Assignment Tool. About the usage of LCD Assignment Tool, refer the “LCD Image Tool User’s Manual”.

4.2.1. Type 1 : 7 segment type LCD panel

Type 1 is 7 segment type LCD panel and has 2 rows. The segments are named from the upper left.

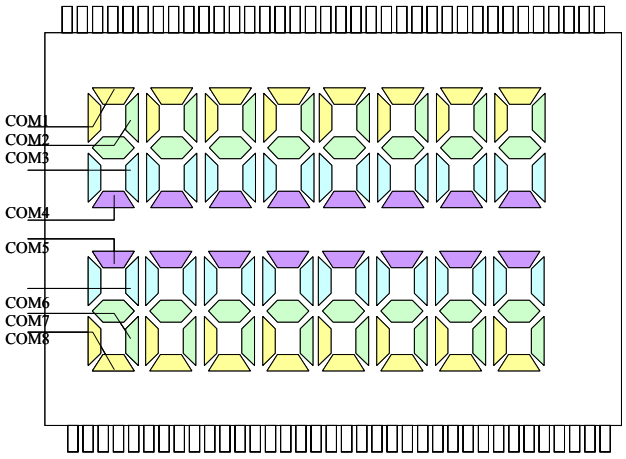


8COM, 16SEG : 7 SEG 16 Figures

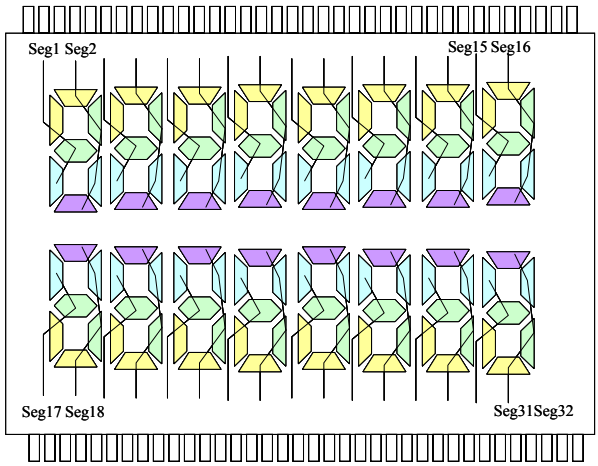
Use the display allocation function

DSPR is set by the display allocation function

7 segment type layout



Common Allocation



Segment Allocation

COM/SEG table of LCD panel

		SEG							
		0	1	2	3	4	5	6	7
Upper row	COM7								
	COM6	g	g	g	g	g	g	g	g
	COM5	f	f	f	f	f	f	f	f
	COM4	e	e	e	e	e	e	e	e
	COM3	d	d	d	d	d	d	d	d
	COM2	c	c	c	c	c	c	c	c
	COM1	b	b	b	b	b	b	b	b
	COM0	a	a	a	a	a	a	a	a
lower row	COM7								
	COM6	g	g	g	g	g	g	g	g
	COM5	f	f	f	f	f	f	f	f
	COM4	e	e	e	e	e	e	e	e
	COM3	d	d	d	d	d	d	d	d
	COM2	c	c	c	c	c	c	c	c
	COM1	b	b	b	b	b	b	b	b
	COM0	a	a	a	a	a	a	a	a
		8	9	10	11	12	13	14	15
		SEG							

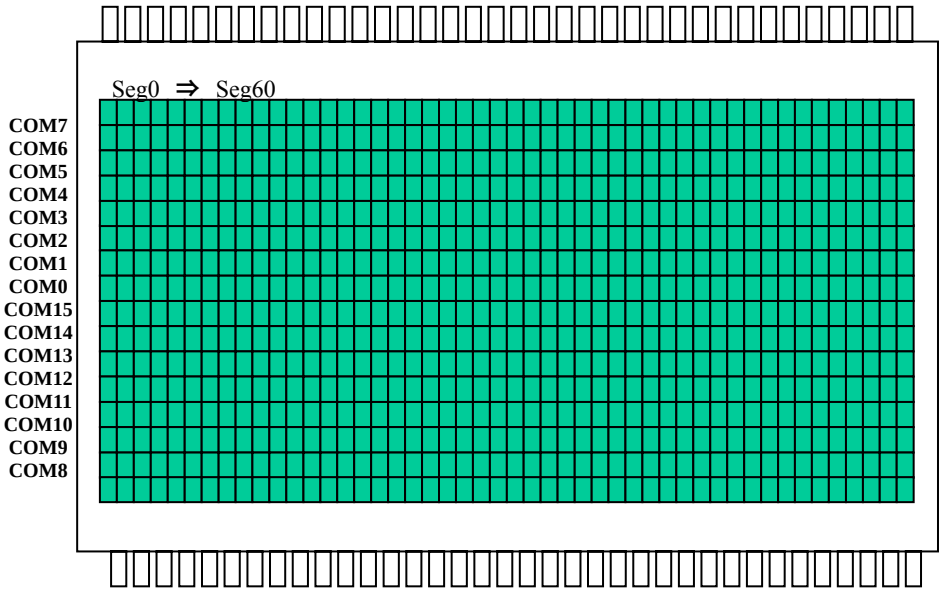
Display location⇔DSPR address correspondence table

Condition : COM = 8, Programmable display allocation, Segment mapping type 3

Display place		Display register	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Upper row	1	DSPR00		g	f	e	d	c	b	a
	2	DSPR01		g	f	e	d	c	b	a
	3	DSPR02		g	f	e	d	c	b	a
	4	DSPR03		g	f	e	d	c	b	a
	5	DSPR04		g	f	e	d	c	b	a
	6	DSPR05		g	f	e	d	c	b	a
	7	DSPR06		g	f	e	d	c	b	a
	8	DSPR07		g	f	e	d	c	b	a
lower row	1	DSPR08		g	f	e	d	c	b	a
	2	DSPR09		g	f	e	d	c	b	a
	3	DSPR0A		g	f	e	d	c	b	a
	4	DSPR0B		g	f	e	d	c	b	a
	5	DSPR0C		g	f	e	d	c	b	a
	6	DSPR0D		g	f	e	d	c	b	a
	7	DSPR0E		g	f	e	d	c	b	a
	8	DSPR0F		g	f	e	d	c	b	a

4.2.2. Type 2 : Dot matrix type layout

Type 2 is the full-dot type LCD panel that has 8COM×60SEG×2rows. The segments are named from the upper left.



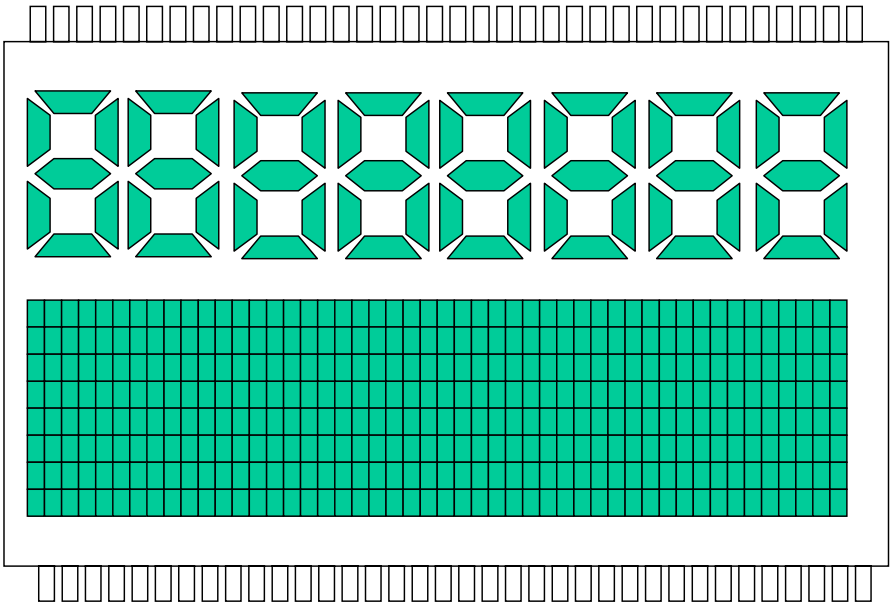
16COM, 60SEG : 8*60Dot (7×6Font 10characters)
Use the fixed display allocation

Display location⇔DSPR address correspondence table
Condition : COM = 16, Fixed display allocation

Display place		Display register	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Upper row	1	DSPR00	c8	c7	c6	c5	c4	c3	c2	c1
	2	DSPR01	c8	c7	c6	c5	c4	c3	c2	c1
	3	DSPR02	c8	c7	c6	c5	c4	c3	c2	c1
	4	DSPR03	c8	c7	c6	c5	c4	c3	c2	c1
	5	DSPR04	c8	c7	c6	c5	c4	c3	c2	c1
	6	DSPR05	c8	c7	c6	c5	c4	c3	c2	c1
		...								
	64	DSPR3F	c8	c7	c6	c5	c4	c3	c2	c1
lower row	1	DSPR40	c16	c15	c14	c13	c12	c11	c10	c9
	2	DSPR41	c16	c15	c14	c13	c12	c11	c10	c9
	3	DSPR42	c16	c15	c14	c13	c12	c11	c10	c9
	4	DSPR43	c16	c15	c14	c13	c12	c11	c10	c9
	5	DSPR44	c16	c15	c14	c13	c12	c11	c10	c9
	6	DSPR45	c16	c15	c14	c13	c12	c11	c10	c9
		...								
	64	DSPR7F	c16	c15	c14	c13	c12	c11	c10	c9

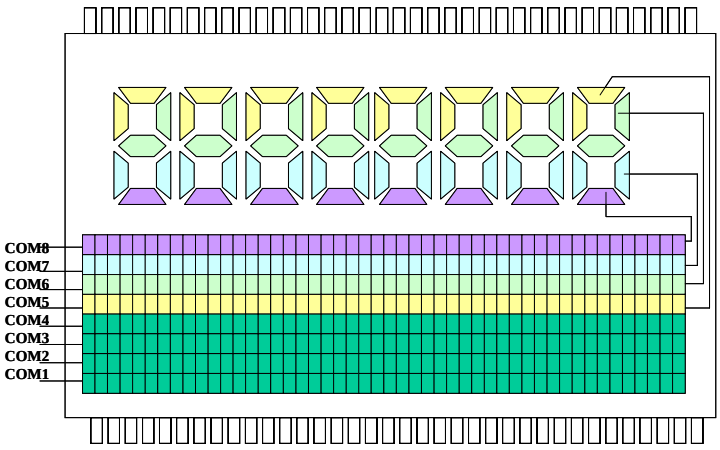
4.2.3. Type 3 : Mix type layout

Type 3 is the LCD panel that has 8COM. The upper row is 7SEG type and the lower row is dot matrix type. In both rows, segments are named from the upper left.

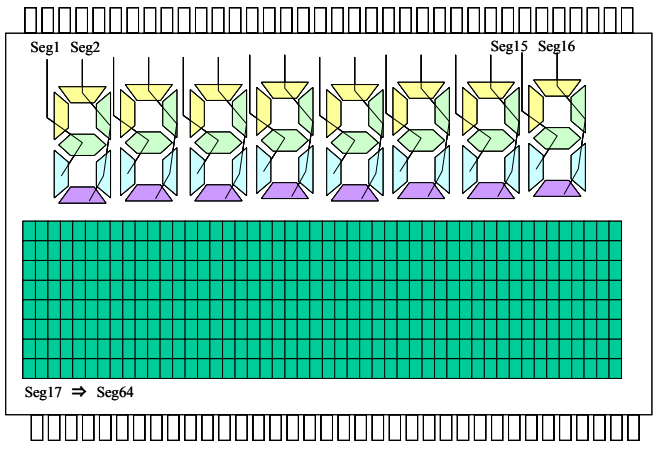


8 COM, 56SEG : 7SEG 8Figures, 8×48Dot (7×6Font 8characters)

Use the display allocation function



Common Allocation



Segment Allocation

COM/SEG table of LCD panel

		SEG							
		0	1	2	3	4	5	6	7
Upper row	COM8								
	COM7	g	g	g	g	g	g	g	g
	COM6	f	f	f	f	f	f	f	f
	COM5	e	e	e	e	e	e	e	e
	COM4	d	d	d	d	d	d	d	d
	COM3	c	c	c	c	c	c	c	c
	COM2	b	b	b	b	b	b	b	b
	COM1	a	a	a	a	a	a	a	a
Lower row	COM8								
	COM7								
	COM6								
	COM5								
	COM4								
	COM3								
	COM2								
	COM1								

8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61

SEG

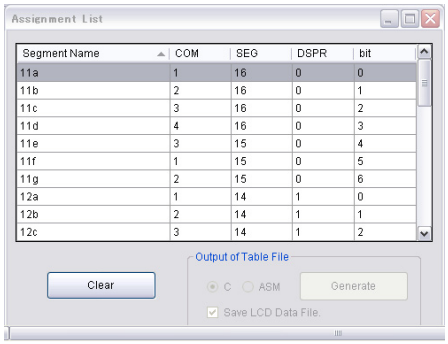
Display location⇔DSPR address correspondence table

Condition : COM = 8, Programmable display allocation

Display place		Display register	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Upper 7SEG	1	DSPR00		g	f	e	d	c	b	a
	2	DSPR01		g	f	e	d	c	b	a
	3	DSPR02		g	f	e	d	c	b	a
	4	DSPR03		g	f	e	d	c	b	a
	5	DSPR04		g	f	e	d	c	b	a
	6	DSPR05		g	f	e	d	c	b	a
	7	DSPR06		g	f	e	d	c	b	a
	8	DSPR07		g	f	e	d	c	b	a
Lower Dot	9	DSPR09	c8	c7	c6	c5	c4	c3	c2	c1
	10	DSPR0A	c8	c7	c6	c5	c4	c3	c2	c1
	11	DSPR0B	c8	c7	c6	c5	c4	c3	c2	c1
	12	DSPR0C	c8	c7	c6	c5	c4	c3	c2	c1
	13	DSPR0D	c8	c7	c6	c5	c4	c3	c2	c1
	14	DSPR0E	c8	c7	c6	c5	c4	c3	c2	c1
	...									
	64	DSPR7F	c8	c7	c6	c5	c4	c3	c2	c1

4.3. Confirmation

Confirm the contents of the definition of each segment and bit by using LCD Assignment Tool. Dragging with the left button of mouse pressing on Assignment List window, more than one segment can be selected.The selected segment is displayed with red color on Image window. So it would be better to confirm by selecting each clump of segment (e.g. 7SEG).



Segment name	COM	SEG	DSPR	bit
11a	1	16	0	0
11b	2	16	0	1
11c	3	16	0	2
11d	4	16	0	3
11e	3	16	0	4
11f	2	16	0	5
11g	1	16	0	6

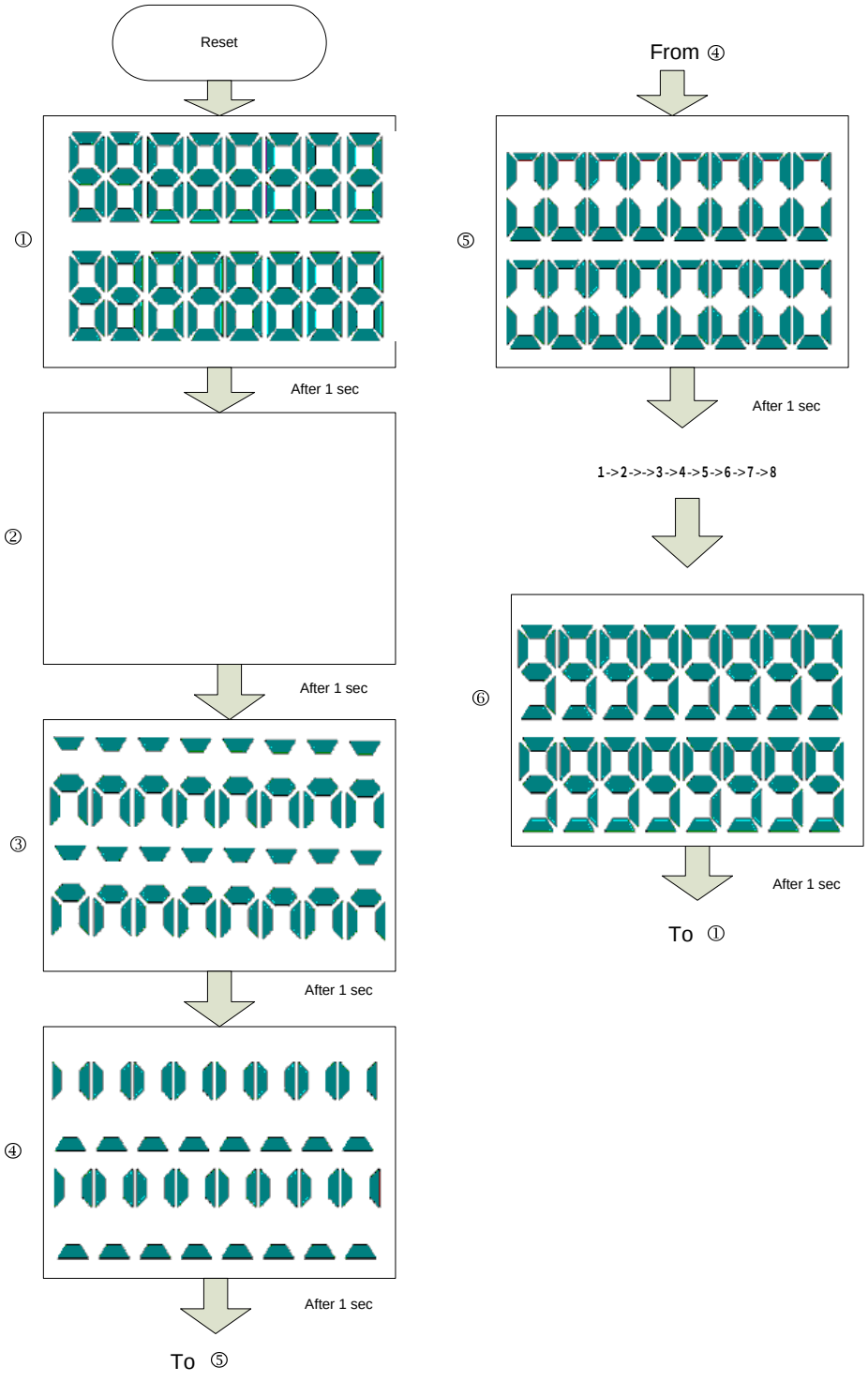
4.4. Program

The following describes the operation specification of the sample program for 3 types of LCD.

4.4.1. Type 1 Display operation of 7 segment type LCD

The display is updated each 1 second as follows.

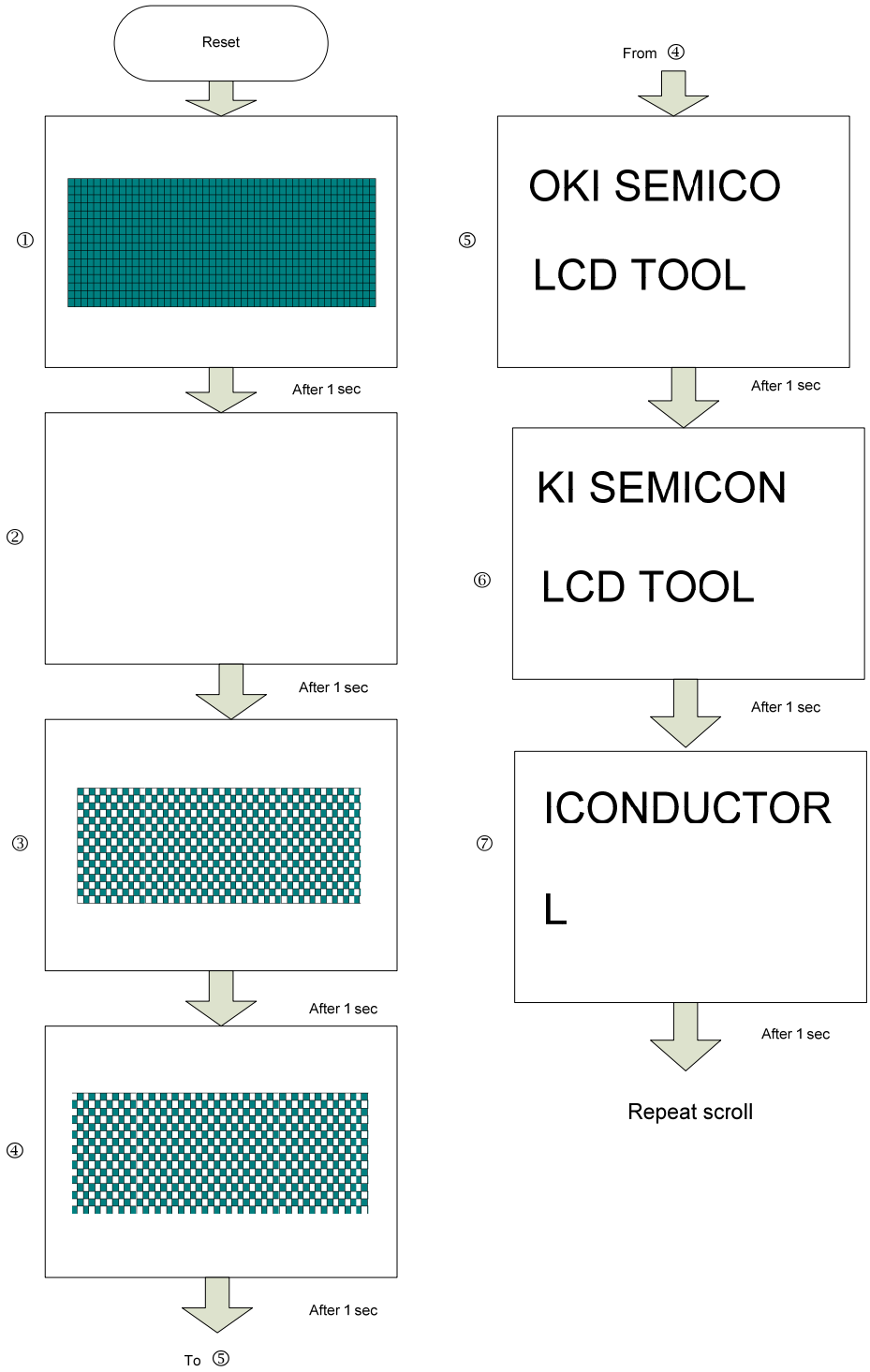
All ON -> All OFF -> Hound's tooth 1 -> Hound's tooth 2 -> Increment number from 0 to 9



4.4.2. Type 2 Display operation of Dot matrix type LCD panel

The display is updated each 1 second as follows.

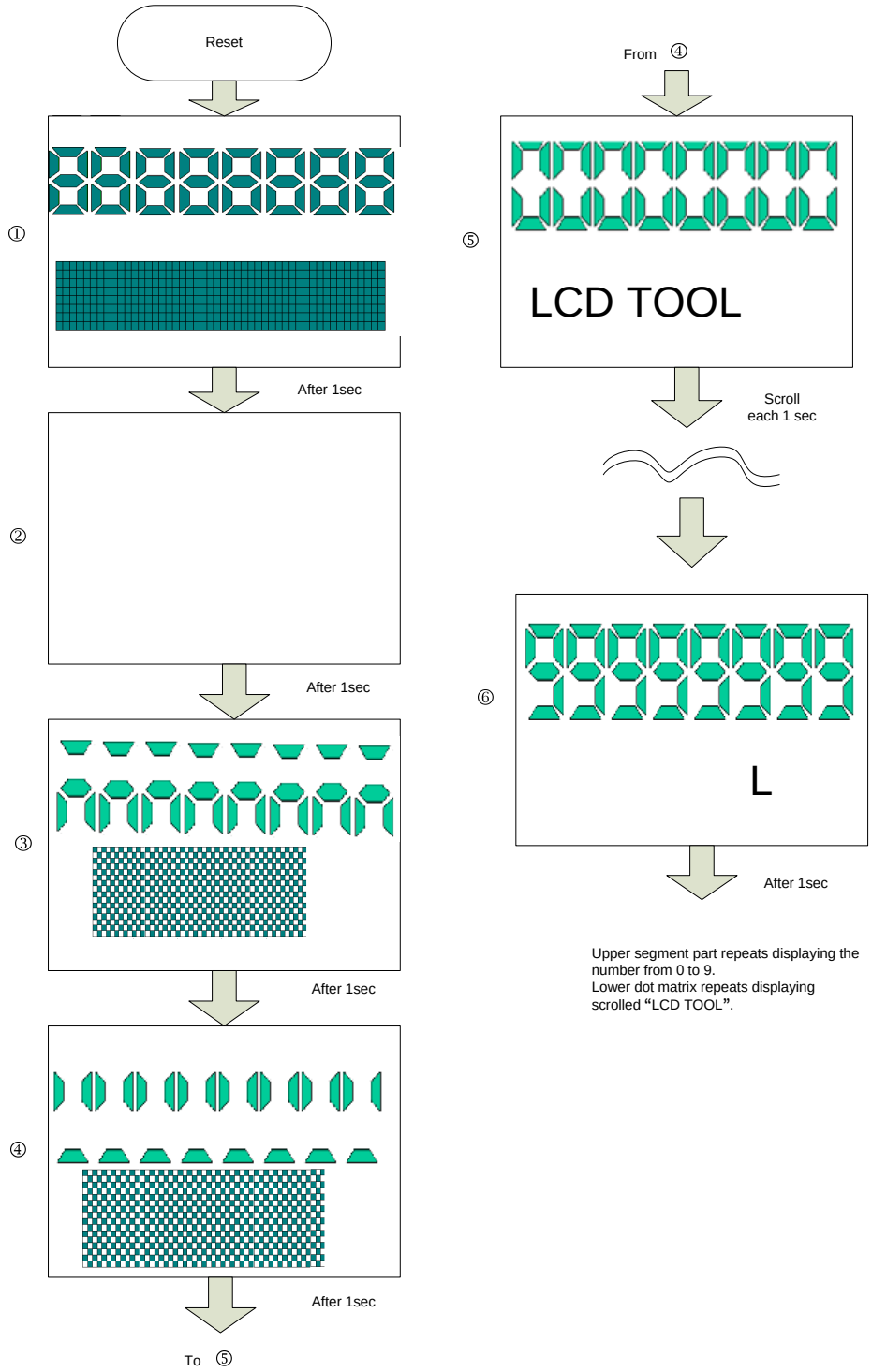
All ON -> All OFF -> Hound's tooth 1 -> Hound's tooth 2
-> Scroll the character (Upper : OKI SEMICONDUCTOR Lower : LCD TOOL)



4.4.3. Type 3 Display operation of Mix type LCD panel

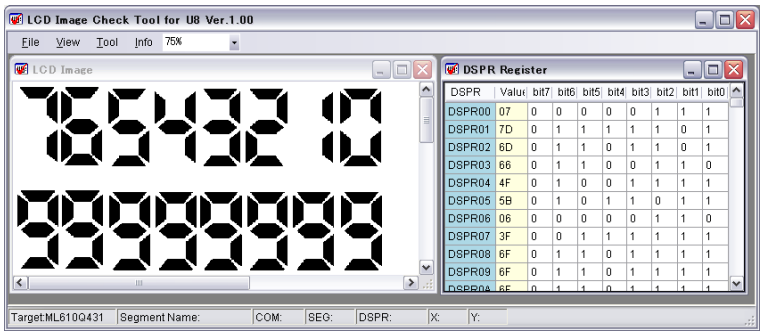
The display is updated each 1 second as follows.

All ON -> All OFF -> Hound's tooth 1 -> Hound's tooth 2
-> Increment number (from 0 to 9) at the upper row, and Scroll the character (LCD TOOL) at the lower row



4.5. Debug

Working with DTU8 debugger, Confirm the displayed contents of LCD panel,



There are two ways of displaying the contents.

- Snapshot
  Display the contents of DSPR to LCD Image Check Tool at any point of time.

- Auto update
  Display the contents of DSPR to LCD Image Check Tool at at regular intervals. It is useful to see the transition of the LCD panel image.

4.5.1. Snapshot

The following is the procedure to display the panel image by snapshot.

① Start DTU8 and load the program to be debugged.
② Set a breakpoint at the location (address) in the source window of DTU8, where you want to display DSPR, and execute the program.(Figure 4.5.1_1)
③ Select "Tool"->" Script command" from the menu of DTU8, and then save the contents of DSPR into a file by using "sfrsav" command. (Figure 4.5.1_2)
④ Start LCD Image Check Tool. Select the menu "File" -> "Load DSPR Data File" and specify the output file from ③. Then, LCD panel image is displayed in the Image window of LCD Image Check Tool.(Figure 4.5.1_3)

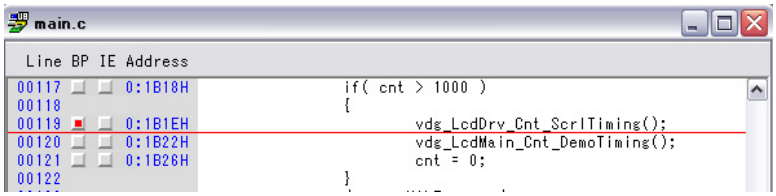


Figure 4.5.1_1 Set a breakpoint

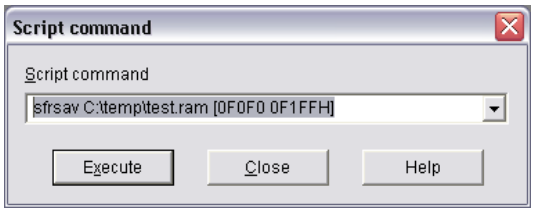


Figure 4.5.1_2 Execution of sfrsav command

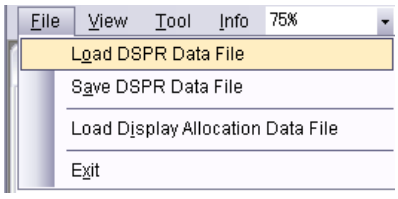


Figure 4.5.1_3 Load the contents of DSPR

[Format of sfrsav command]

sfrsav <filename> [<start address> <end address>]

<filename> : Specify the filename which the contents of DSPR are saved in.
If the extension is omitted, .ram is specified by default.

<start address> <end address> :
Specify the output range. The range from 0F0F0H to 0F1FFH, shown in Figure 4.5.1_2, is the range that the display control resisters and DSPR are located.
About the location of resisters, see the User's manual of LSI.

4.5.2. Auto update

The following is the procedure to display the panel image by auto update.

① Start DTU8 and load the program to be debugged.
② Confirm the location (address) in the source window of DTU8, where you want to display DSPR.(Figure 4.5.2_1)
③ Make the macro script for auto update.(Figure 4.5.2_2) Specify .mcr as the extention of the macro script file.
Here, assume the filename is test.mcr.
④ To update automatically on LCD Image Check Tool, select the menu "Tool" -> "Start Auto Update Mode" and specify the macro script file that is made in the procedure ③.
If this file does not exist, the message, like Figure 4.5.2_3, is issued. Then, press "Yes".
⑤ Select "Tool" -> "Run macro" from the menu of DTU8, and specify the file made in the procedure ③.
Then, LCD panel image is updated at at regular intervals on LCD Image Check Tool.
into a file by using "sfrsav" command. (Figure 4.5.1_2)
⑥ To stop the display, select the menu "File" -> "Stop Auto Update Mode" on LCD Image Check Tool and stop the macro execution on DTU8.

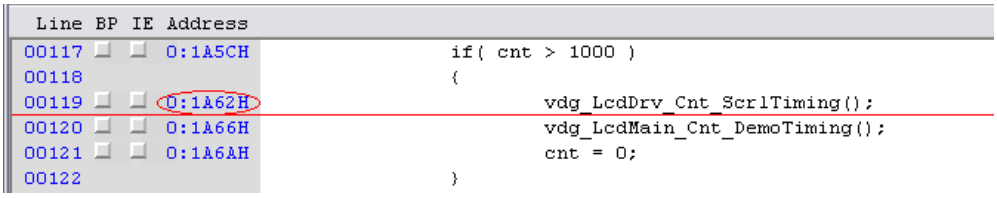


Figure 4.5.2_1 Confirm the brakepoint

```
SBP 0:1A62H
:label000
G
sfrsav C:¥temp¥test.ram [0F0F0h 0F1FFh]
GOTO label000
```

Figure 4.5.2_2 Contents of test.mcr

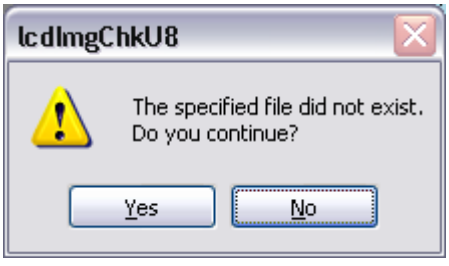


Figure 4.5.2_3 Message

5. How to run the sample program

This section introduces how to build the sample program and how to run on the DTU8 simulator.

5.1. Modify source code

In the sample program, there are the program codes for the watch dog timer (WDT) clear procedure. Because the DTU8 simulator does not support WDT operation, if this WDT clear procedure runs on the simulator, it will not finish (that is, enter an eternal loop).
Therefore, only in the case that you run the sample program according to this section, please delete or comment out the WDT clear procedures, which are described at the following places in the source code.

[WDT clear procedure (1)]
Near the line 86 in the startup file "S610431SW.asm"

```
        mov     r12,    #05ah           ;; <---- from here
        mov     r13,    #0a5h
__wdtcounter_clear:
        st      r12,    WDTCON
        tb      WDP
        bz      __wdtcounter_clear
        st      r13,    WDTCON          ;; <---- to here
```

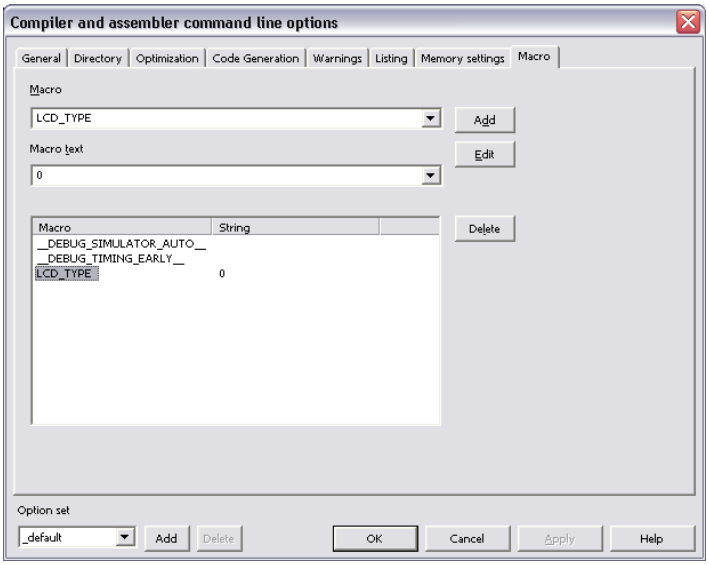
[WDT clear procedure (2)]
Near the line 167 in "main.c" (in main_clrWDT function)

```
void main_clrWDT( void )
{
        do {                                    /* <---- from here */
                WDTCON = (unsigned char)0x5Au;
        } while (WDP != 1);
        WDTCON = (unsigned char)0xA5u;          /* <---- to here */
}
```

5.2. Build

It is necessary to set the operation condition of the sample program by modifying the macro definition. To modify the macro definition, follow the procedure below on IDEU8.

① After loading the project file of the sample program (U8_Lcd_Sample.PID), select the menu "Project" -> "Options" -> "Compiler/assembler" -> "Macro" tab.
② After selecting "LCD_TYPE" in the list of macro, enter the value corresponding to the type of LCD in the "Macro text" field and press "Apply". The value that can be entered is as follows.
If the type of LCD is 1, enter 0.
If the type of LCD is 2, enter 1.
If the type of LCD is 3, enter 2.
③ Enter "__DEBUG_SIMULATOR_AUTO__" in the "Macro" field. And press "Add" button.
④ Next, enter "__DEBUG_TIMING_EARLY__" in the "Macro" field. And press "Add" button.
* If these macros have already been registered, the procedure ③④ are not needed.
Note: in the macro ③④, there are two '_' characters at the head and tail. But the number of '_' characters between other characters is 1.
⑤ Press "OK" button.
⑥ Select the menu "Project" -> "Rebuild".



5.3. Execution

① After the procedure ⑥”Rebuild” in the section 5.1, select the menu “Project” -> “Debug”.
② Select “SimU8 Simulator” from the “Target ICE” list on the dialog shown Figure 5.2_1, and press “OK”.

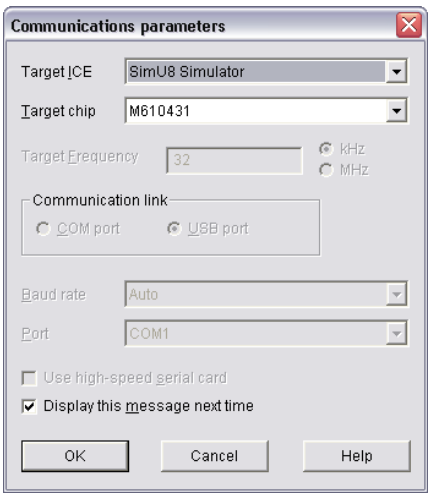


Figure 5.2_1 Dialog at DTU8 starting.

③ Select main.c and confirm the address that is indicated by the line in Figure 5.2_2. This address is used to set breakpoint later (④).

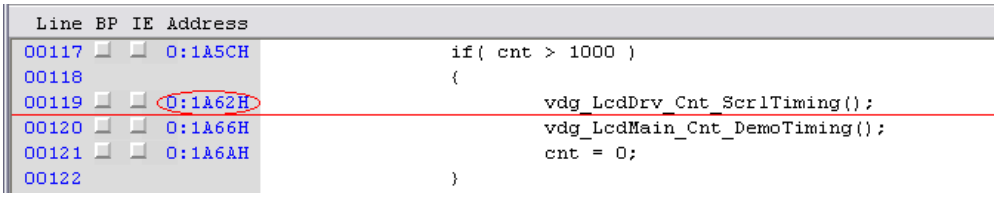


Figure 5.2_2 Check the location for the breakpoint setting later

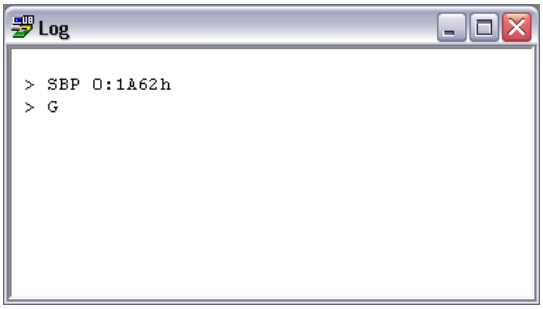
④ Edit the macro script “test_lcdsample.mcr”.
Change the address at line 1 to the address that is confirmed in the procedure ③.
Change the path of RAM file to an appropriate path.

```
SBP 0:1A62H
:label000
G
sfrsav C:¥temp¥test.ram [0F0F0h 0F1FFh]
GOTO label000
```

⑤ Start LCD Image Assignment Tool with the following setting. (Here, assume that the target device is ML610Q431.)

Type	Display Allocation Option	LCD file * This file is specified, selecting the menu ‘File’ -> ‘Load’
Type 1	Programmable	lcdImgAtU8_L1.ld8
Type 2	Fixed	lcdImgAtU8_L2.ld8
Type 3	Programmable	lcdImgAtU8_L3.ld8

⑥ Select the menu “Tool” -> “LCD Image Check Tool”, and then LCD Image Check Tool starts. If Type 2 was selected, the window to select Display Allocation Type is displayed. Please select Type 3 in this window.
⑦ According to the section “4.5.7 Auto update”, specify the RAM file that was specified in the procedure ④.
⑧ Select the menu of DTU8, “View” -> “Interrupt”. Then Interrupt window is displayed.
⑨ Select the menu of DTU8, “Tools” -> “Run macro” and specify the macro script that is edited in the procedure ④.
⑩ After the following message is outputted in the log window of DTU8, click TM2INT button in the Interrupt window.



After the above procedure, the display image of LCD panel is displayed in the Image window of LCD Image Check Tool.

Revision History

Revision History

Edition	Date	Page		Description
		Previous Edition	Current Edition	
1	2009.6.26	–	–	First Edition
2	2010.1.27	19	19	Description about the source code modification, which is needed to run the sample program on DTU8 simulator, is added.
3	2010.4.16	–	5-9	The following sections are added. 1.1 Hardware Configuration 1.2 Software Configuration 1.3 List of Folders and Files 1.4 Build Procedure 1.5 Restrictions