LAPIS
SEMICONDUCTOR

SQ003116E002

ML610Q400 Series

Sample Program AP Notes

For
RTC 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.

Copyright 2009 - 2011 LAPIS Semiconductor Co., Ltd.

Table of Contents

1. Overview

This document describes the application programming notes (hereafter called the AP notes) arranged to help customers develop software that performs time measurements on the ML610Q400 Series MCU (hereafter called the MCU).

APIs are provided for each function module. The AP notes describe the functions and operating conditions of each API and samples of use of those APIs.

In connection with the AP notes, a sample program is provided that actually operates using APIs on ML610Q400 Series Demo Kit.

◆ Related Documents
The following are the related documents. Read them as required.

- ML610Q400 Series Sample Program AP Notes For Sensor/Mesurement Application
- 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
- LCD Image Tool User's Manual

1.1. Software Configuration

Figure 1-1 shows the software configuration.

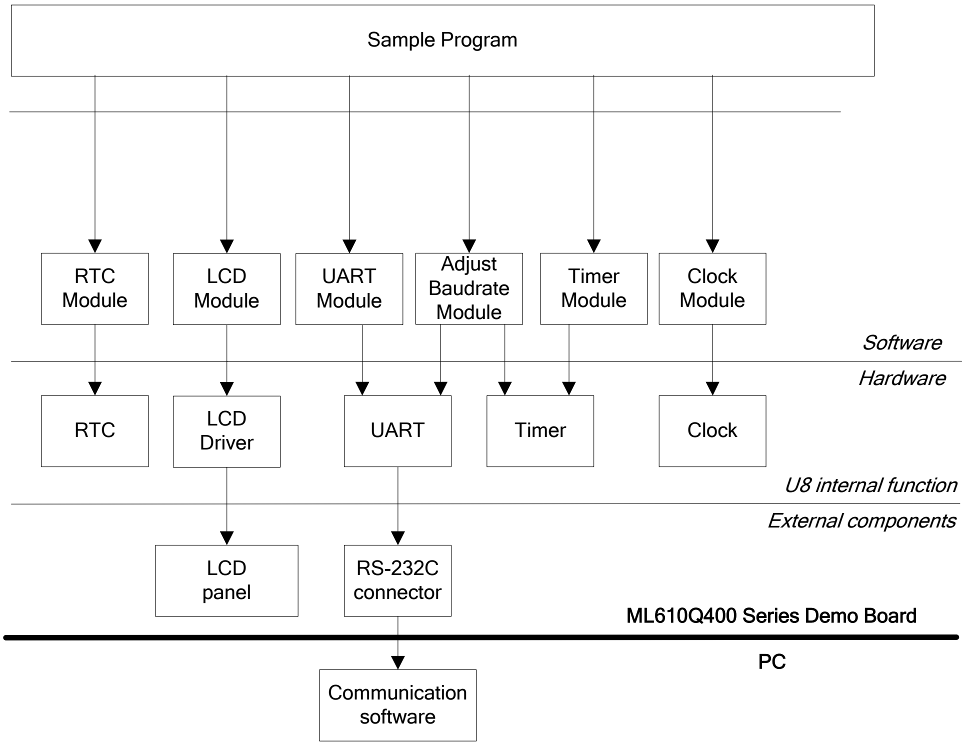


Figure 1-1 Software Configuration

1.2. List of Folders and Files

The folders and the files are as listed below.

```
[rtc]
├ [_output]                 … Build result output folder
│ ├ [_hex]
│ ├ [_lst]
│ ├ [_obj]
│ └ [_prn]
├ [adjustBaudrate]          … UART baud rate correction module folder
│ ├ adjustbaudrate.c
│ └ adjustBaudrate.h
├ [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.c
│ ├ LCD.h
│ ├ U8_Sample.tac
│ └ U8_Sample.tbc
├ [main]                    … Sample program main folder
│ ├ [mcu_large]
│ │ └ mcu.h
│ ├ [mcu_small]
│ │ └ mcu.h
│ ├ main.c
│ ├ main.h
│ ├ S610431SW.asm
│ └ S610435LW.asm
├ [output_uart]             … UART state control module folder
│ ├ output_uart.c
│ └ output_uart.h
├ [rtc]                     … Real time clock module folder
│ ├ rtc.c
│ └ rtc.h
├ [tbc]                     … Time base counter control module folder
│ ├ tbc.c
│ └ tbc.h
├ [timer]                   … Timer control module folder
│ ├ timer.c
│ └ timer.h
├ [uart]                    … UART communication control module folder
│ ├ uart.c
│ └ uart.h
├ readme.txt                … Description of compile options
├ U8_Rtc_Sample_Large.PID   … Project file for large model MCU
└ U8_Rtc_Sample_Small.PID   … Project file for large model MCU
```

1.3. 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_Rtc_Sample_Small.PID”. In the case of large model, the project file is “U8_Rtc_Sample_Large.PID”. Correspondence of MCU and PID file is shown below.

Table 1-1 Correspondence of MCU and PID file

	U8_Rtc_Sample_Small.PID	U8_Rtc_Sample_Large.PID
Supported MCU	ML610Q431/432 ML610Q421/422 ML610Q411/412/415 ML610Q482	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, define 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 1-1 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”

(5) If necessary, modify other macro definitions.
About the available macro definitions, see the “readme.txt” in the sample program folder.

- For ML610Q43X series MCU
  LCD_TYPE = 1
  FREQ_TIMER_MODE = 0
  _RTC_TYPE or _SOFTWARE_RTC
  _OUTPUT_UART (Please define, if you want to send the data of time and date via UART.)
- For ML610Q42X series MCU
  LCD_TYPE = 1
  FREQ_TIMER_MODE = 0 or 1
  _SOFTWARE_RTC
  _OUTPUT_UART (Please define, if you want to send the data of time and date via UART.)
- For ML610Q41X series MCU
  LCD_TYPE = 0
  FREQ_TIMER_MODE = 0 or 1
  (For ML610Q415, frequency measurement mode by hardware is not available on ML610Q415 because it does not have low-speed crystal oscillation clock. Please define FREQ_TIMER_MODE macro as 0.)
  _SOFTWARE_RTC
  _OUTPUT_UART (Please define, if you want to send the data of time and date via UART.)
- For ML610Q41X series MCU
  FREQ_TIMER_MODE = 0 or 1
  _SOFTWARE_RTC
  _OUTPUT_UART

③ Select the menu “Project” -> “Rebuild”. Then the build procssing for the sample program starts.

④ When the build processing is completed, .abs file is generated in the project folder and .hex file is generated in _output¥_hex folder.

1.4. Restrictions

1.4.1. 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 1-2 List of available functional modules

			Supported MCU			
			ML610Q43X	ML610Q42X	ML610Q41X	ML610Q48X
Functional modules	RTC Control Module	Hardware RTC	○	×	×	×
		Software RTC	○	○	○	○
	LCD Display Control Module *3		○	○ *1	○ *1	×
	UART Communication Control Module *3		○	○	○	○
	UART Baud Rate Correction Module		○	○	○	○
	Frequency measurement mode *3		×	○	○ *2	○
	Timer Control Module *3		○	○	○	○
	Clock Control Module *3		○	○	○	○

○ : 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: Frequency measurement mode by hardware is not available on ML610Q415 because it does not have low-speed crystal oscillation clock.
*3: For the details of these modules, please see the "ML610Q400 Series Sample Program AP Notes For Sensor/Mesurement Application".

1.4.2. About Display Area of LCD panel

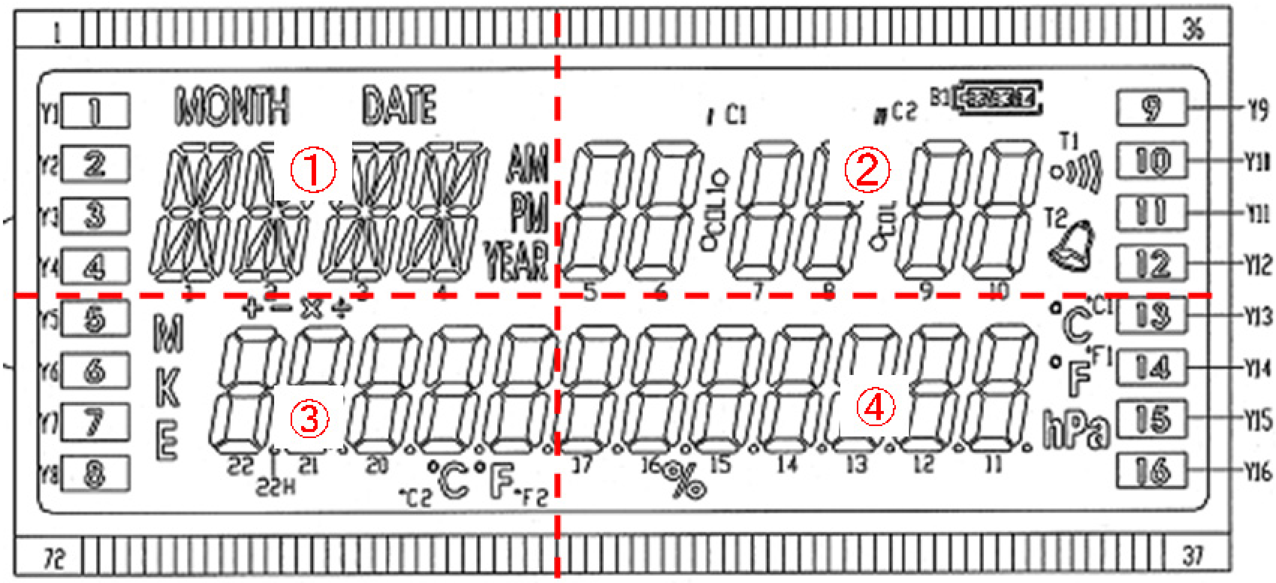
The display area of LCD panel is different by each MCU as follows, because of the specification difference of LCD driver.
* It is requred for displaying all areas of LCD panel that LCD driver supports 64seg×4com pins at least. The number of COM/SEG pin that LCD driver in each MCU supports is listed in parenthesis.

ML610Q43X: All area can be displayed.
(ML610Q431: 64seg×16com, ML610Q432: 64seg×24com)

ML610Q42X: Only the area of ①, ② and ④ can be displayed.
(ML610Q421: 50seg×8com, ML610Q422: 50seg×16com)

ML610Q41X: Only the area of ① and ② can be displayed.
(ML610Q411: 36seg×4com, ML610Q412: 44seg×4com, ML610Q415: 36seg×4com)

ML610Q48X: All area can not be displayed, because ML610Q48X does not have LCD driver.

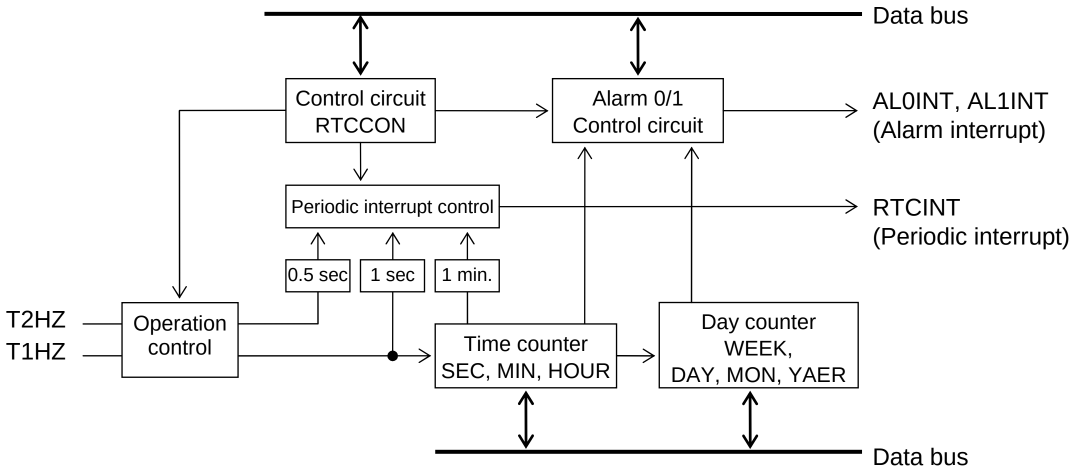


2. Description of Functional Modules

2.1. RTC Module

This sample program provides two methods to realize the real time clock (RTC). One method is the hardware RTC and another is the software RTC. The hardware RTC is enabled by default. The software RTC is enabled when the macro "_SOFTWARE_RTC" is defined. The software RTC can realize the RTC function even if MCU doesn't have the RTC function.

■ About MCU that has the real time clock (RTC) function
The following shows the configuration of the RTC in MCU.



Data bus
Control circuit RTCCON
Alarm 0/1 Control circuit
AL0INT, AL1INT (Alarm interrupt)
Periodic interrupt control
RTCINT (Periodic interrupt)
0.5 sec
1 sec
1 min.
T2HZ
T1HZ
Operation control
Time counter SEC, MIN, HOUR
Day counter WEEK, DAY, MON, YAER
Data bus

RTCCON: Real time clock control register

Figure 2-1 Configuration of the real time clock

* For details, refer to the chapter "Real Time Clock" of the User's Manual for your target MCU.

■ About MCU that does not have the real time clock (RTC) function

The RTC function is realized by the software RTC. The following shows the functional block diagram of the software RTC.

About the detail of each function, please see the secton “2.1.1 Function Overview”.

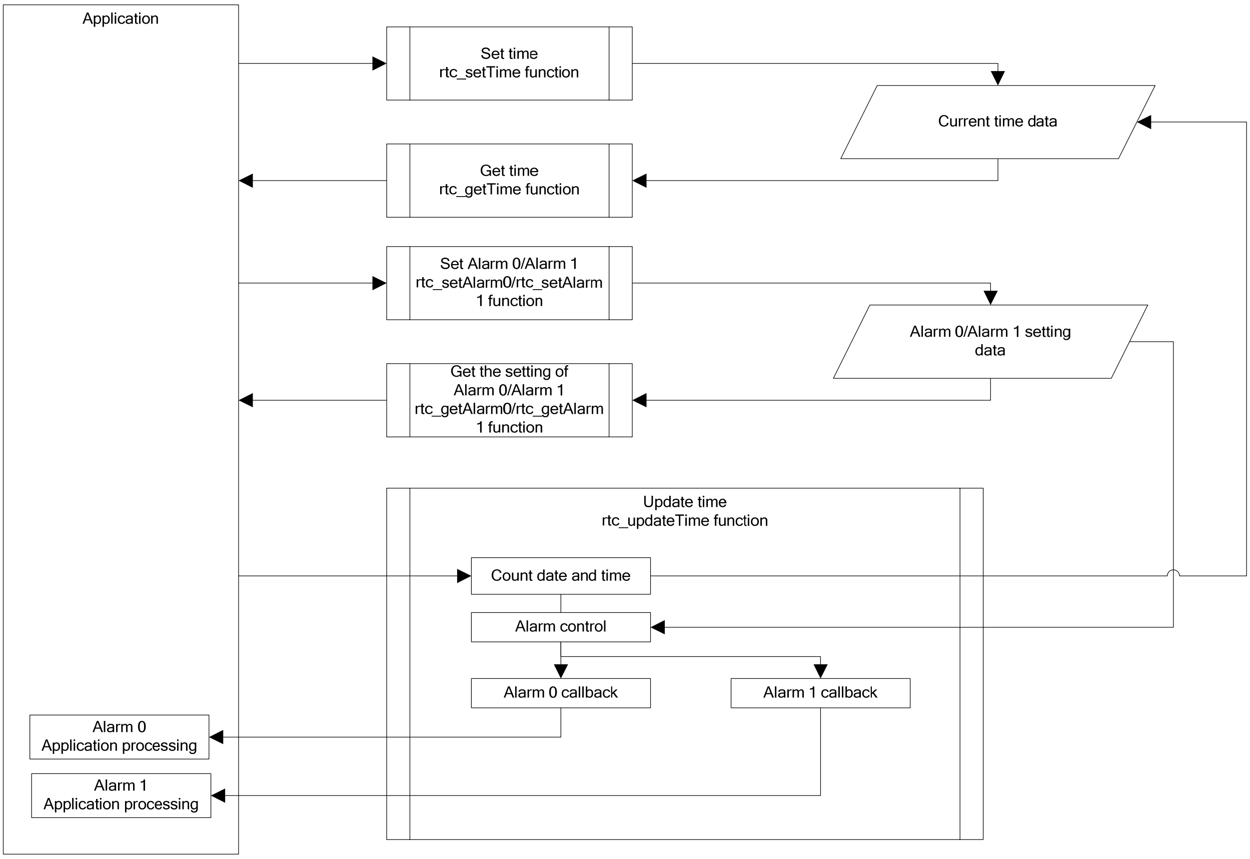


Figure 2-2 Functional block diagram of the software RTC

2.1.1. Function Overview

The RTC module mainly sets the settings for the counting functions of the real time clock of the MCU.
Table 3-20 lists the RTC module APIs that the sample program uses.

Table 2-1 List of APIs

Function name	Description	**Hardware RTC**	**Software RTC**
rtc_setTime function	Sets date (year, month, day, day of the week) and clock time (hour, minute, second).	✓	✓
rtc_getTime function	Obtains date (year, month, day, day of the week) and clock time (hour, minute, second).	✓	✓
rtc_start function	Starts RTC operation.	✓	(*2)
rtc_stop function	Stops RTC operation.	✓	(*2)
rtc_setRegularInt function	Selects the interval between periodic interrupts.	✓	(*2)
rtc_setAlarm0 function	Sets Alarm 0(day of the week, hour, minute).	✓	✓
rtc_setAlarm1 function	Sets Alarm 1(month, day, hour, minute).	✓	✓
rtc_getAlarm0 function	Obtains the setting of Alarm 0(day of the week, hour, minute).	✓	✓
rtc_getAlarm1 function	Obtains the setting of Alarm 1(month, day, hour, minute).	✓	✓
rtc_updateTime function	Updates date (year, month, day, day of the week) and clock time (hour, minute, second).	(*1)	✓
rtc_calcWeekday function	Calculates day of the week.	✓	✓

(*1) Not available for the hardware RTC.
(*2) No need to use for the software RTC. These functions do not execute any processing in the software RTC.

2.1.2. Operating Conditions

This section describes the operating conditions and valid range of this module. It also describes the restrictions on this module.

- The RTC is stopped immediately after an MCU reset.

■ Year values settable with the date counting function	00–99
■ Month values settable with the date counting function	01–12
■ Day values settable with the date counting function	01–31
■ Day of the week values settable with the date counting function	1–7
■ Hour values settable with the clock time counting function	00–23
■ Minute values settable with the clock time counting function	00–59
■ Second values settable with the clock time counting function	00–59
■ Year range in which day of the week can be calculated	2000–2099

Note:

- The parameters used in each of the APIs above must be specified by binary-code decimal (BCD) values, except for day of the week.
- Day of the week, which is calculated by rtc_calcWeekday function, is represented by the following values.

day of the week	value
Sunday	1
Monday	2
Tuesday	3
Wednesday	4
Thursday	5
Friday	6
Saturday	7

- The timing that the alarm occurs is different between the hardware RTC and the software RTC.
    - ➢ Hardware RTC: Alarm interrupt (AL0INT、AL1INT)
    - ➢ Software RTC: Main routine (rtc_updateTime function call)

2.1.3. Sample of Use

The subsection below describes the procedure for setting date and clock time using RTC module.

2.1.3.1. Date and Clock Time Setting Procedure (Hardware RTC)

The figure below shows the procedure for setting date (year, month, day, day of the week) and clock time (hour, minute, second) using the hardware RTC function in RTC module.

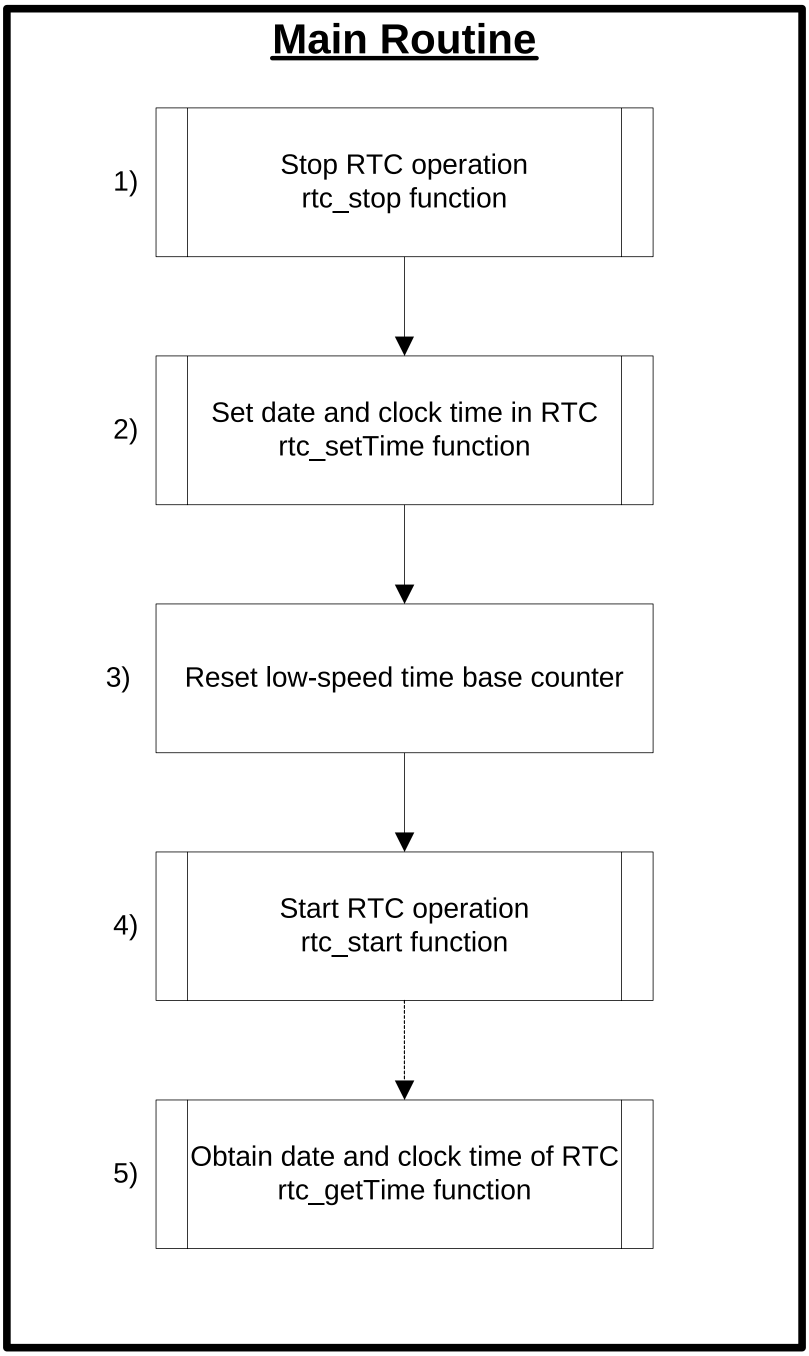


Figure 2-3 Date and Clock Time Setting Procedure (Hardware RTC)

[Main Routine]

1) Stop RTC operation (because date and clock time have to be set with the RTC stopped)

2) Set date and clock time
   - ➢ Stop RTC operation using the **rtc_stop function**.
   - ➢ Set date and clock time using the **rtc_setTime function**.
     ① Second data (0–59)
     ② Minute data (0–59)
     ③ Hour data (0–23)
     ④ Day-of-the-week data (1–7)
     ⑤ Day data (1–31)
     ⑥ Month data (1–12)
     ⑦ Year data (0–99)

3) Reset the low-speed time base counter
   - ➢ By resetting the low-speed time base counter, clear the internal counter used to count for less than one second and secure a count of 1 second after the start of RTC operation.

4) Start RTC operation
   - ➢ Start RTC operation using the **rtc_start function**.

5) Obtain the date and clock time of RTC
   - ➢ Obtain the current date (year, month, day, day of the week) and clock time (hour, minute, second) using the **rtc_getTime function**.

2.1.3.2. Date and Clock Time Setting Procedure (Software RTC)

The figure below shows the procedure for setting date (year, month, day, day of the week) and clock time (hour, minute, second) using the software RTC function in RTC module.

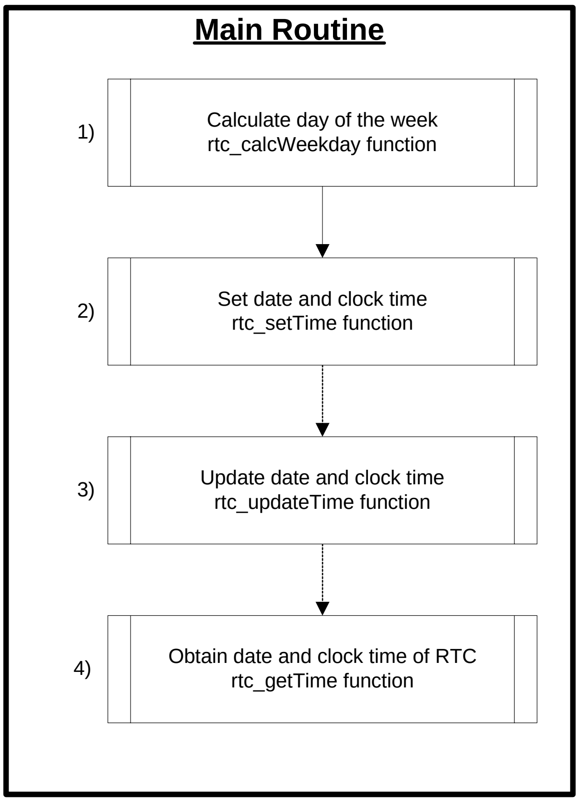


Figure 2-4 Date and Clock Time Setting Procedure (Software RTC)

[Main Routine]

1) Calculate day of the week
   - Calculate day of the week using the **rtc_calcWeekday function**.

2) Set date and clock time
   - Set date and clock time using the **rtc_setTime function**.
     ① Second data (0–59)
     ② Minute data (0–59)
     ③ Hour data (0–23)
     ④ Day-of-the-week data (caluculated value by **rtc_calcWeekday function**)
     ⑤ Day data (1–31)
     ⑥ Month data (1–12)
     ⑦ Year data (0–99)

3) Update date and clock time
   - Update the current date (year, month, day, day of the week) and clock time (hour, minute, second) by calling **rtc_updateTime function** at regular intervals (every second).

4) Obtain the date and clock time of RTC
   - Obtain the current date (year, month, day, day of the week) and clock time (hour, minute, second) using the **rtc_getTime function**.

2.1.3.3. Alarm 0 Setting Procedure (Software RTC)

The figure below shows the procedure for setting alarm 0 using the software RTC function in RTC module.

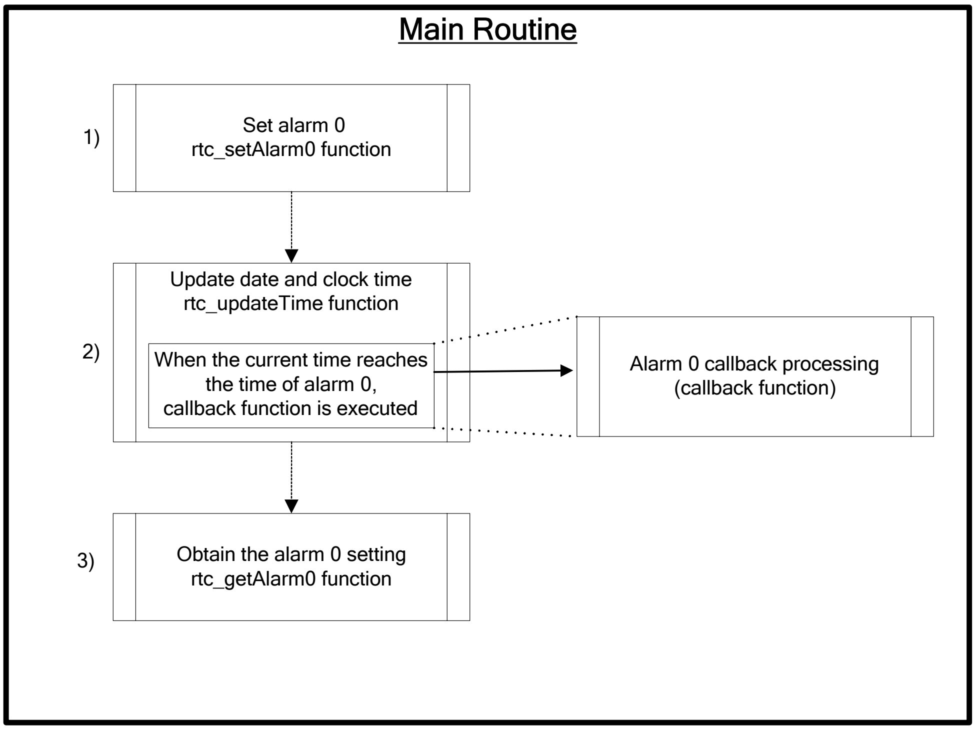

Main Routine
1)
Set alarm 0
rtc_setAlarm0 function
2)
Update date and clock time
rtc_updateTime function
When the current time reaches the time of alarm 0, callback function is executed
Alarm 0 callback processing
(callback function)
3)
Obtain the alarm 0 setting
rtc_getAlarm0 function

Figure 2-5 Alarm 0 Setting Procedure (Software RTC)

[Main Routine]

1) Set alarm 0
   - Set date and clock time using the **rtc_setAlarm0 function**.
     ① Minute data (0–59)
     ② Hour data (0–23)
     ③ Day-of-the-week data (1–7)
     ④ Address of callback function
   - If you do not use the day-of-the-week data as comparison data of alarm 0, set “0x00” as its data. For example, in the case of setting 8:30 in the morning to alarm 0, set “0x00” to the day-of-the-week data, then set “8(hour)” and “30(minute)” to the hour and minute data respectively.

2) Update date and clock time
   - Update the current date and clock time by calling **rtc_updateTime function**.
   - If the current time reaches the time that is set to alarm 0 (day of the week, hour, minute) when the second of current time overflows (from 59 to 00) and the minute is raised up, the callback function specified in above “Set alarm 0 (rtc_setAlarm0 function)” is executed.

3) Obtain the alarm 0 setting
   - Obtain the current alarm 0 setting data (day of the week, hour, minute, address of callback function) using the **rtc_getAlarm0 function**.

2.1.3.4. Alarm 1 Setting Procedure (Software RTC)

The figure below shows the procedure for setting alarm 1 using the software RTC function in RTC module.

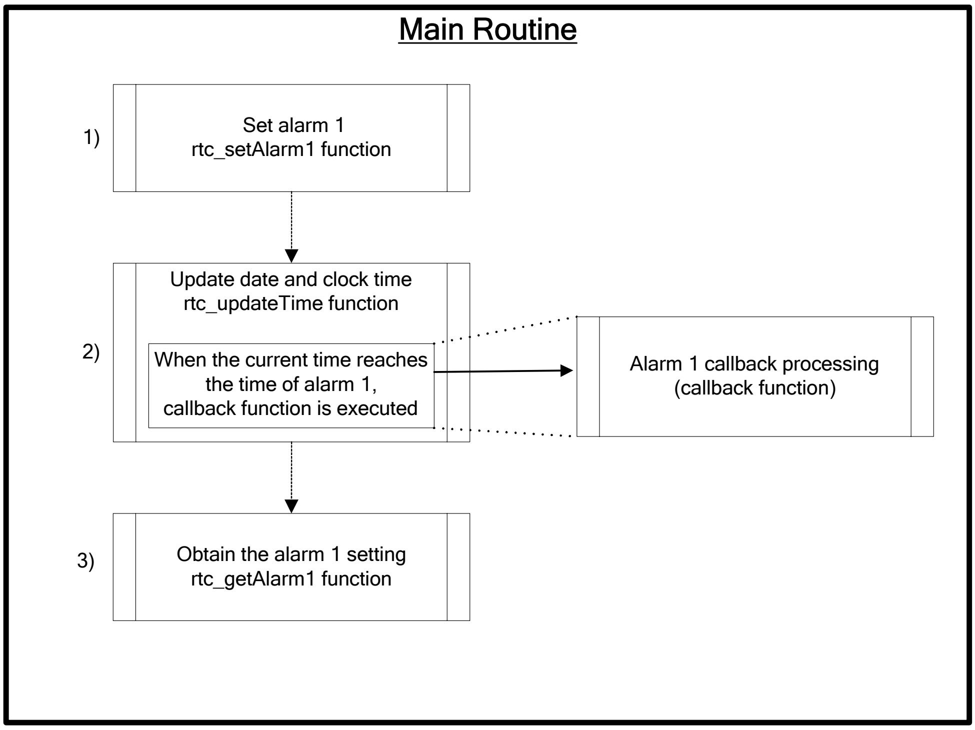

Main Routine
1)
Set alarm 1
rtc_setAlarm1 function
2)
Update date and clock time
rtc_updateTime function
When the current time reaches the time of alarm 1, callback function is executed
Alarm 1 callback processing
(callback function)
3)
Obtain the alarm 1 setting
rtc_getAlarm1 function

Figure 2-6 Alarm 1 Setting Procedure (Software RTC)

[Main Routine]

1) Set alarm 1
   - ➢ Set date and clock time using the **rtc_setAlarm1 function**.
     - ① Minute data (0–59)
     - ② Hour data (0–23)
     - ③ Month data (1–12)
     - ④ Day data (1–31)
     - ⑤ Address of callback function
   - ➢ If you do not use the month and day data as comparison data of alarm 1, set “0x00” as its data. For example, in the case of setting 8:30 in the morning to alarm 1, set “0x00” to the month and day data, then set “8(hour)” and “30(minute)” to the hour and minute data respectively.

2) Update date and clock time
   - ➢ Update the current date and clock time by calling **rtc_updateTime function**.
   - ➢ If the current time reaches the time that is set to alarm 1 (month, day, hour, minute) when the second of current time overflows (from 59 to 00) and the minute is raised up, the callback function specified in above “Set alarm 1 (rtc_setAlarm1 function)” is executed.

3) Obtain the alarm 1 setting
   - ➢ Obtain the current alarm 1 setting data (month, day, hour, minute, address of callback function) using the **rtc_getAlarm1 function**.

3. Description of the Sample Program

3.1. Operation conditions

1) System clock
   - SYSCLK=HSCLK (RC oscillation mode 500 kHz)
2) Timer
   - Channels 0, 16-bit mode, operating clock LSCLK
   - Overflow interval 0.015625 second. (It is multiplied by 64 in dorder to make 1 second interval.)
3) LCD driver
   - Bias voltage multiplying clock: 2 kHz, Bias: 1/4 bias, Duty: 1/4 duty
   - Frame frequency : 73 Hz
4) UART
   - 9600 bps, 8-bit, no parity, 1 Stop bit, positive logic, LSB first
     * To use RS232C interface mounted on ML610Q400 Series Demo Kit, it is necessary to set P42 and P43 as a secondary function by selection of a port function jumper switch (short-circuit between 2-1 pins) on ML610Q400 Series Demo Kit.

* About the other conditions and the peripheral circuit, please see "ML610Q400 Series Demo kit Hardware User's Manual".

3.2. Function Overview

This sample program displays date and clock time on LCD panel. If the macro "_OUTPUT_UART" is defined, the date and clock time data is transmitted via UART, not displayed on LCD panel.(*)
Date and clock time is updated every second from the pre-defined date and clock time.

(*) In the case of the following MCU, "_OUTPUT_UART" is defined by default.
ML610Q411/412/415
ML610Q421/422
ML610Q482

3.3. State Transition

The following shows the state transision diagram of this sample program.

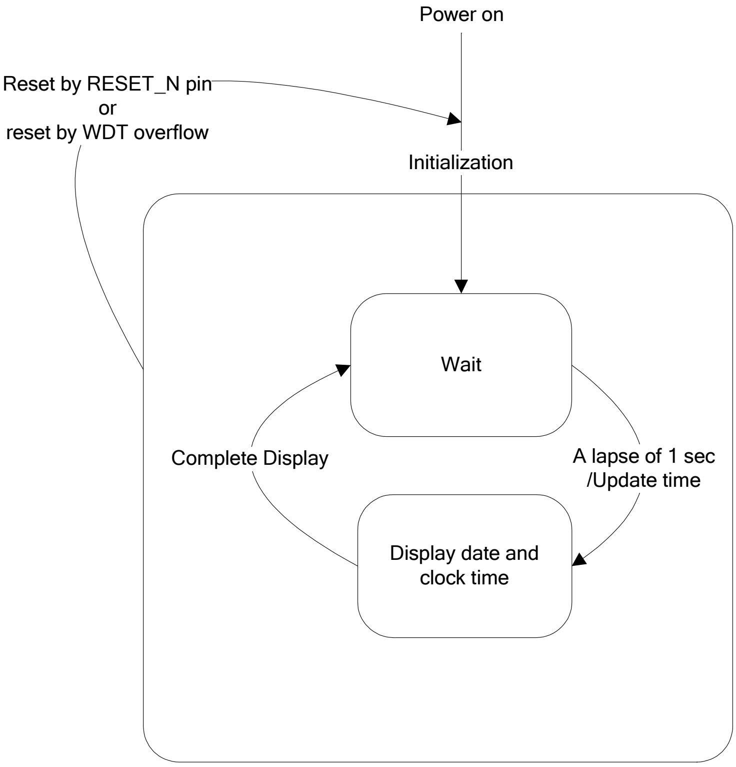


Figure 3-1 State Transition Diagram (main function)

State	Description
Wait	This state is wait sate for the clock time update. When the 1 sec interval interrupt occurs, It transits to“Display date and clock time” state after update date and clock time.
Display date and clock time	This state is the state for displaying the updated date and clock time. If the macro “_OUTPUT_UART” is defined, the date and clock time data is transmitted via UART, not displayed on LCD panel.*

* About the operation in transmitting via UART

In this case, it is necesary to correct UART baud rate by using UART Baud Rate Correction Module. Therefore, the state internally transits as follows.

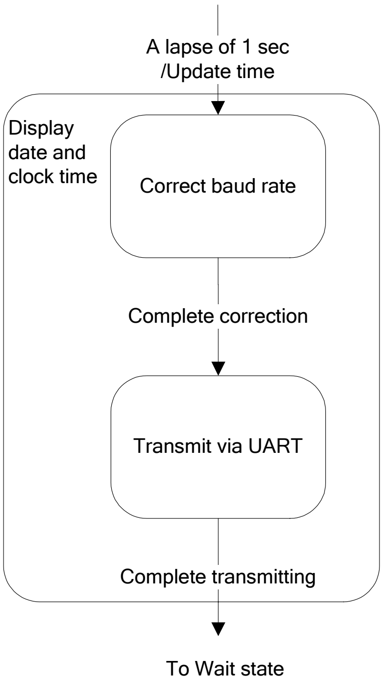


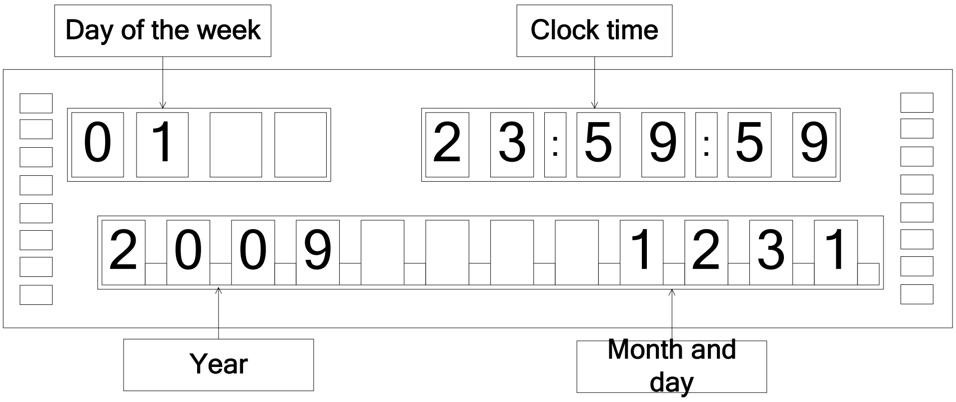
Figure 3-2 Internal transition in “Display date and clock time” state (when use UART)

About the procedure for correction of baud rate and UART communication, please see the section “UART Baud Rate Correction Module“ and “UART Communication Control Module”, in the “ML610Q400 Series Sample Program AP Notes for Sensor/Mesurement Application”

3.4. LCD Panel Image

This subsection describes the LCD panel configuration.

The LCD panel has two types of display patterns depending on the type of the LCD driver built into the MCU: one with the display allocation function and the other without it. This sample program assumes that the LCD panel is equipped with the display allocation function.



Name	Content to be displayed
Day of the week	Displays day of the week (from Monday:1 to Saturday:7) that corresponds to current date (year, month, day).
Clock time	Displays the current time in 24 hour format.
Year	Displays current year. Upper place of year is fixed to “20”.
Month and day	Displays current month and day. * These values are not zero-suppressed.

3.5. UART Data Formats

In this sample program, if the macro “_OUTPUT_UART” is defined, the date and clock time data is transmitted via UART, not displayed on LCD panel. The following shows the UART transmission data formats.

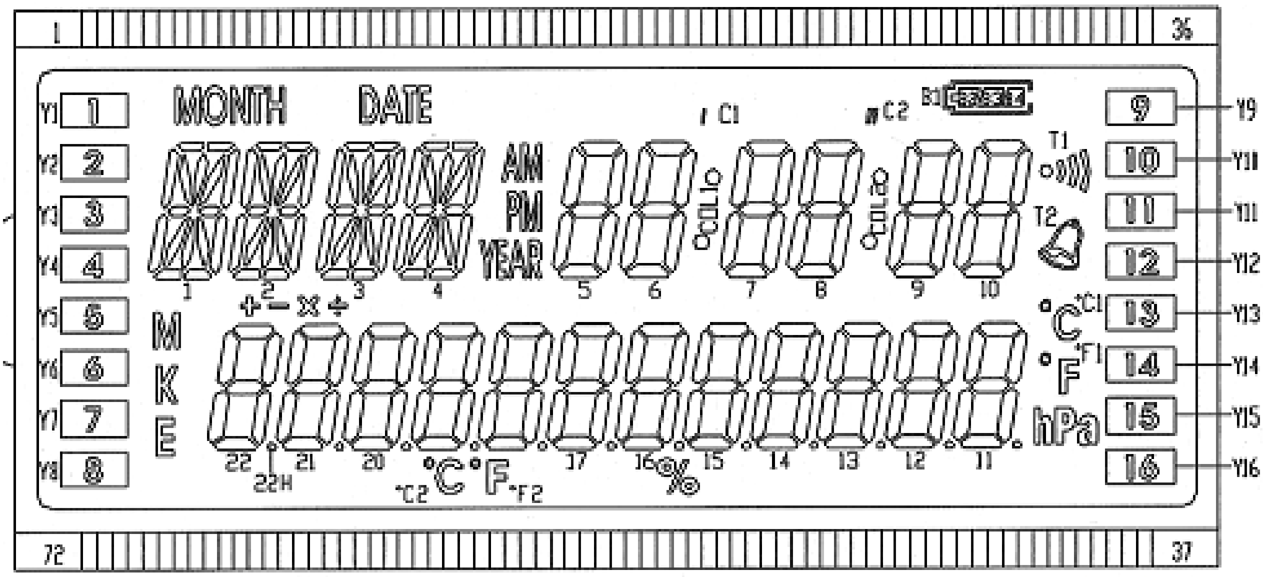
Table 3-1 UART transmission data format

Offset	Size (byte)	Content of data	Value of data	Remarks
0	4	Year data	“2000” - ”2099”	
4	1	Space	“ ”	Insert a space as a data delimiter.
5	2	Month data	“01” - ”12”	
7	1	/	“/”	Insert a slash “/” as a date delimiter.
8	2	Day data	“01” - ”31”	
10	1	Space	“ ”	Insert a space as a data delimiter.
11	1	Day of the week data	“1” - ”7”	From Monday:1 to Saturday:7
12	1	Space	“ ”	Insert a space as a data delimiter.
13	2	Hour data	“00” - ”23”	
15	1	“:”	“:”	Insert a colon “:” as a clock-time delimiter.
16	2	Minute data	“00” - ”59”	
18	1	“:”	“:”	Insert a colon “:” as a clock-time delimiter.
19	2	Second data	“00” - ”59”	
21	1	Line feed	“¥r”	Insert a linefeed code to indicate the termination of data.
Total	22			

* Use text data for the value of data.

4. Appendix

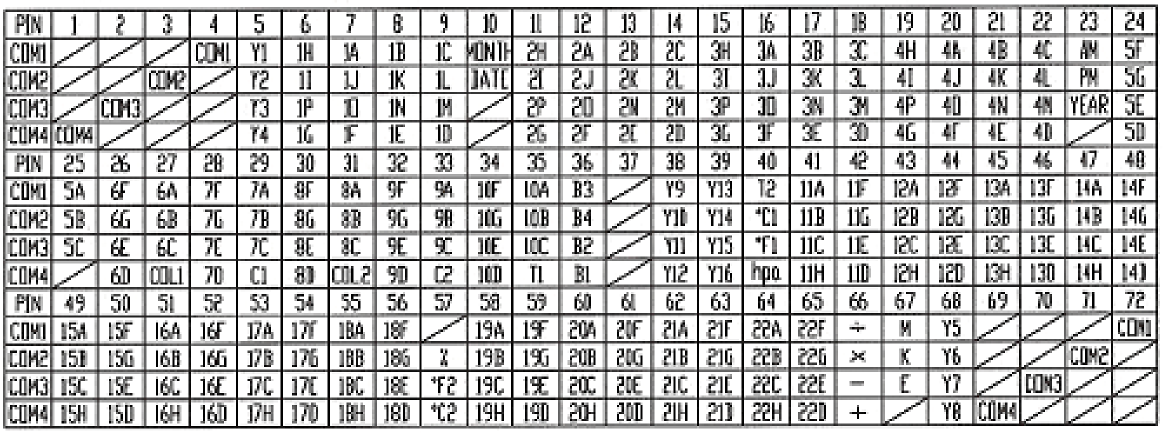
4.1. LCD Panel Specifications



16-segment characters: The 4 digits on the upper part of the panel
7-segment characters: The 6 digits on the upper part of the panel
8-segment characters: The 12 digits on the lower part of the panel
Marks for hand-held calculator: 7
Other marks: 32

Figure 4-1 Layout of the LCD Panel

Table 4-1 Pin Assignments (COM/SEG)



PIN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
COM1				COM1	Y1	1H	1A	1B	1C	MONTH	2H	2A	2B	2C	3H	3A	3B	3C	4H	4A	4B	4C	AM	5F
COM2			COM2		Y2	1I	1J	1K	1L	DATE	2I	2J	2K	2L	3I	3J	3K	3L	4I	4J	4K	4L	PM	5G
COM3		COM3			Y3	1P	1O	1N	1M		2P	2O	2N	2M	3P	3O	3N	3M	4P	4O	4N	4M	YEAR	5E
COM4	COM4				Y4	1G	1F	1E	1D		2G	2F	2E	2D	3G	3F	3E	3D	4G	4F	4E	4D		5D
PIN	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
COM1	5A	6F	6A	7F	7A	8F	8A	9F	9A	10F	10A	B3		Y9	Y13	T2	11A	11F	12A	12F	13A	13F	14A	14F
COM2	5B	6G	6B	7G	7B	8G	8B	9G	9B	10G	10B	B4		Y10	Y14	°C1	11B	11G	12B	12G	13B	13G	14B	14G
COM3	5C	6E	6C	7E	7C	8E	8C	9E	9C	10E	10C	B2		Y11	Y15	°F1	11C	11E	12C	12E	13C	13E	14C	14E
COM4		6D	COL1	7D	C1	8D	COL2	9D	C2	10D	T1	B1		Y12	Y16	hpa	11H	11D	12H	12D	13H	13D	14H	14D
PIN	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
COM1	15A	15F	16A	16F	17A	17F	18A	18F		19A	19F	20A	20F	21A	21F	22A	22F	÷	M	Y5				COM1
COM2	15B	15G	16B	16G	17B	17G	18B	18G	%	19B	19G	20B	20G	21B	21G	22B	22G	×	K	Y6			COM2	
COM3	15C	15E	16C	16E	17C	17E	18C	18E	°F2	19C	19E	20C	20E	21C	21E	22C	22E	−	E	Y7		COM3		
COM4	15H	15D	16H	16D	17H	17D	18H	18D	°C2	19H	19D	20H	20D	21H	21D	22H	22D	+		Y8	COM4			

Specifications of Operation

Clock for bias generation circuit voltage multiplication: 1/16 LSCLK (2 kHz)
Bias of the bias generation circuit: 1/4
Duty: 1/4 duty
Frame frequency: 73 Hz

Revision History

Revision History

Edition	Date	Page		Description
		Previous Edition	Current Edition	
1	2009.6.26	–	–	Initial Edition
2	2010.1.27	–	–	Update edition number to match with Japanese edition.
3	2010.4.16	3	3	List of Folders and Files is updated.
		4	4	Build procedure is updated.
		–	5-6	Description of Restrictions is added.