SQ003116E003

ML610Q400 Series

Sample Program AP Notes

For
SSIO Application

2nd 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 synchronous serial port (SSIO), which is hardware that the ML610Q400 Series MCU (hereafter called the MCU) has, performs SPI communication.

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 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.

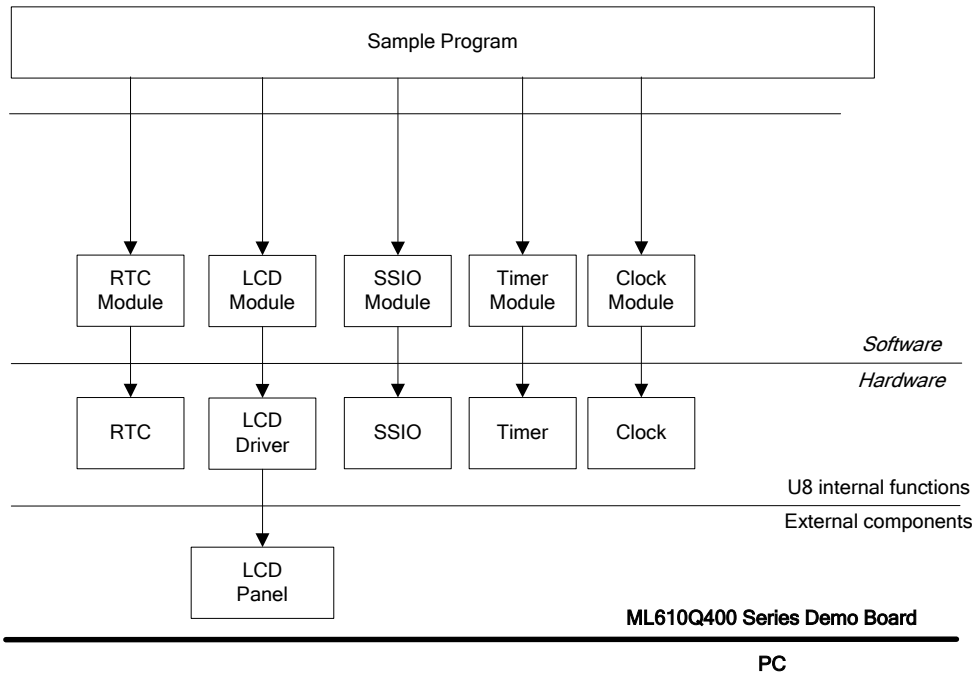


Figure 1-1 Software Configuration

1.2. List of Folders and Files

The folders and the files are as listed below.

```
[ssio]
├ [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
├ [rtc]                   …Real-time clock control module folder
│ ├ rtc.c
│ └ rtc.h
├ [ssio]                  …SSIO module folder
│ ├ ssio.c
│ └ ssio.h
├ [tbc]                   …Time base counter control module folder
│ ├ tbc.c
│ └ tbc.h
├ [timer]                 …Timer control module folder
├ timer.c
│ └ timer.h
├ readme.txt                       … Description of compile options
├ U8_Ssio_Sample_Large.PID         … Project file for large model MCU
└ U8_Ssio_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_Ssio_Sample_Small.PID”. In the case of large model, the project file is “U8_Ssio_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” -> “Option” -> “Compile/Assemble”.
(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” -> “Option” -> “Compile/Assemble” -> ”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
_RTC_TYPE or _SOFTWARE_RTC
_SSIO_P46_P45_P44 (Please define, if you want to use P46, P45 and P44 for SSIO port.)
_SSIO_MASTER_MODE (Please define, if you want to operate SSIO as the master mode.)
SSIO_TRANS_MODE_INI = 1 or 2 or 3
- For ML610Q42X series MCU
LCD_TYPE = 1
_SOFTWARE_RTC
_SSIO_P46_P45_P44 (Please define, if you want to use P46, P45 and P44 for SSIO port.)
_SSIO_MASTER_MODE (Please define, if you want to operate SSIO as the master mode.)
SSIO_TRANS_MODE_INI = 1 or 2 or 3
- For ML610Q41X series MCU
LCD_TYPE = 0
_SOFTWARE_RTC
_SSIO_P46_P45_P44 (Please define, if you want to use P46, P45 and P44 for SSIO port.)
_SSIO_MASTER_MODE (Please define, if you want to operate SSIO as the master mode.)
SSIO_TRANS_MODE_INI = 1 or 2 or 3
- For ML610Q41X series MCU
_SOFTWARE_RTC
_SSIO_P46_P45_P44 (Please define, if you want to use P46, P45 and P44 for SSIO port.)
_SSIO_MASTER_MODE (Please define, if you want to operate SSIO as the master mode.)
SSIO_TRANS_MODE_INI = 1 or 2 or 3

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	SSIO Module		○	○	○	○
	RTC Control Module *2	Hardware RTC	○	×	×	×
		Software RTC	○	○	○	○
	LCD Display Control Module *3		○	○ *1	○ *1	×
	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: For the details of these modules, please see the “ML610Q400 Series Sample Program AP Notes For RTC Application”.
*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

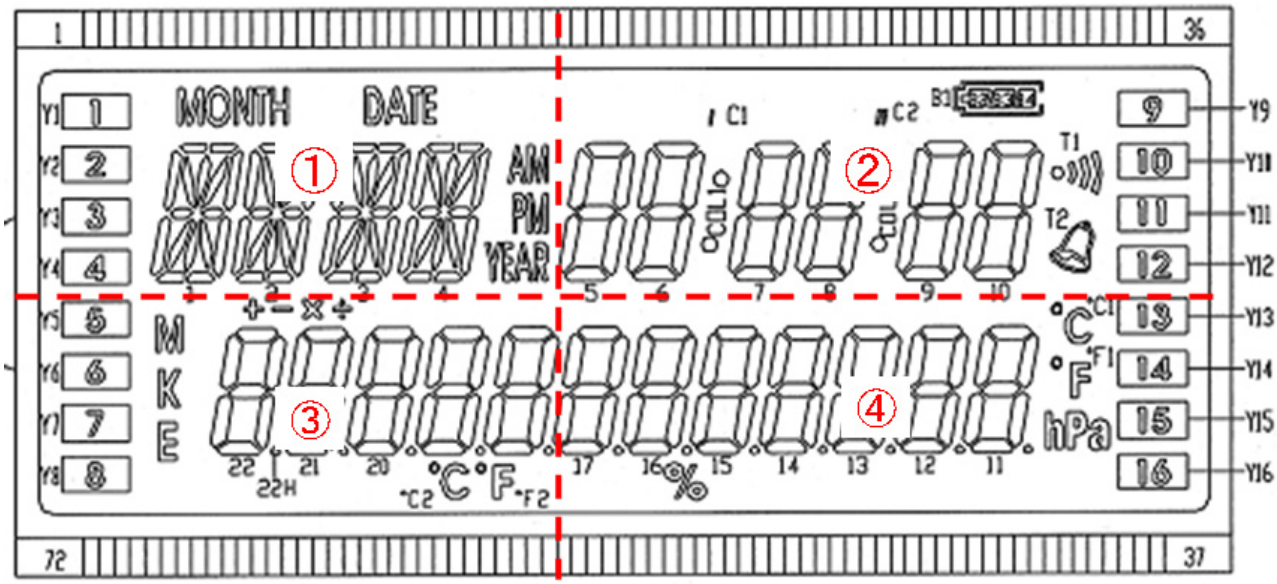
The display area of LCD panel is different by each MCU as follows, because of the specification difference of LCD driver.
* It is requred for displaying all areas of LCD panel that LCD driver supports 64seg×4com pins at least. The number of COM/SEG pin that LCD driver in each MCU supports is listed in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

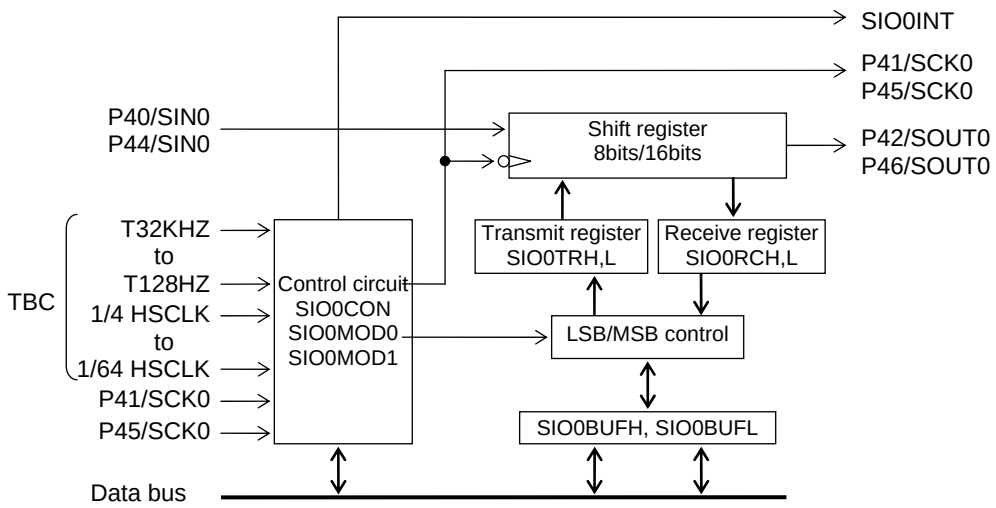
ML610Q48X: All area can not be displayed, because ML610Q48X does not have LCD driver.



2. Description of Functional Modules

2.1. SSIO Module

This LSI includes one channel of the 8/16-bit synchronous serial port (SSIO) and can also be used to control the device incorporated with the SPI interface by using one GPIO as the chip enable pin.



SIO0BUFL : Serial port transmit/receive buffer L
SIO0BUFH : Serial port transmit/receive buffer H
SIO0CON : Serial port control register
SIO0MOD0 : Serial port mode register 0
SIO0MOD1 : Serial port mode register 1

Figure 2-1 Configuration of Synchronous Serial Port

Table 2-1 List of Pins

Pin name	I/O	Description
P40/SIN0 P44/SIN0	I	Receive data input. Used for the tertiary function of the P40 and P44 pins.
P41/SCK0 P45/SCK0	I/O	Synchronous clock input/output. Used for the tertiary function of the P41 and P45 pins.
P42/SOUT0 P46/SOUT0	O	Transmit data output. Used for the tertiary function of the P42 and P46 pins.

* For details, refer to the chapter "Synchronous Serial Port" of the User's Manual for your target MCU.

2.1.1. Function Overview

The SSIO module controls the synchronous serial port (SSIO) of the MCU.
Table 3-9 lists the SSIO module APIs used in the sample program.

Table 2-2 List of APIs

Function name	Description
ssio_init function	Selects the transfer clock and mode (8/16-bit buffer length, clock output phase, LSB/MSB first, and so on).
ssio_start function	Executes synchronous serial communication start processing.
ssio_stop function	Executes synchronous serial communication stop processing.
ssio_checkIRQ function	Confirms whether a synchronous serial port interrupt occurs.
ssio_clearIRQ function	Clears a synchronous serial port interrupt request.
ssio_continue function	Executes synchronous serial communication continuation processing.

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.

- Transfer clock 8 types (LSCLK:32/16KHz, HSCLK:1/4 1/8 1/16 1/32, EXCLK:0/1)
- Bit order 2 types (LSB/MSB first)
- Bit length 2 types (8/16 bit)
- Transmission/reception mode 4 modes (Stop, Transmit, Receive, Transmit/Receive)

Notes:

- If HSCLK is selected as the transfer clock, it is necessary to set HSCLK configuration before the communication starts.
- SSIO ports use the same ports which are used by I2C and UART communication. In the case of using SSIO module with I2C module or UART module together, please be careful about these port assignments.
- In default, P42(SOUT0:data output), P41(SCK0:clock input/output) and P40(SIN0:data input) are enabled to use. But when the macro "_SSIO_P46_P45_P44" is defined, P46(SOUT0:data output), P45(SCK0:clock input/output) and P44(SIN0:data input) are enabled.

2.1.3. Sample of Use

The following subsections describe the procedure for performing data transmission/reception using the SSIO module.

2.1.3.1. Transmission Procedure

Shown below is the procedure for performing data transmission using the SSIO module.

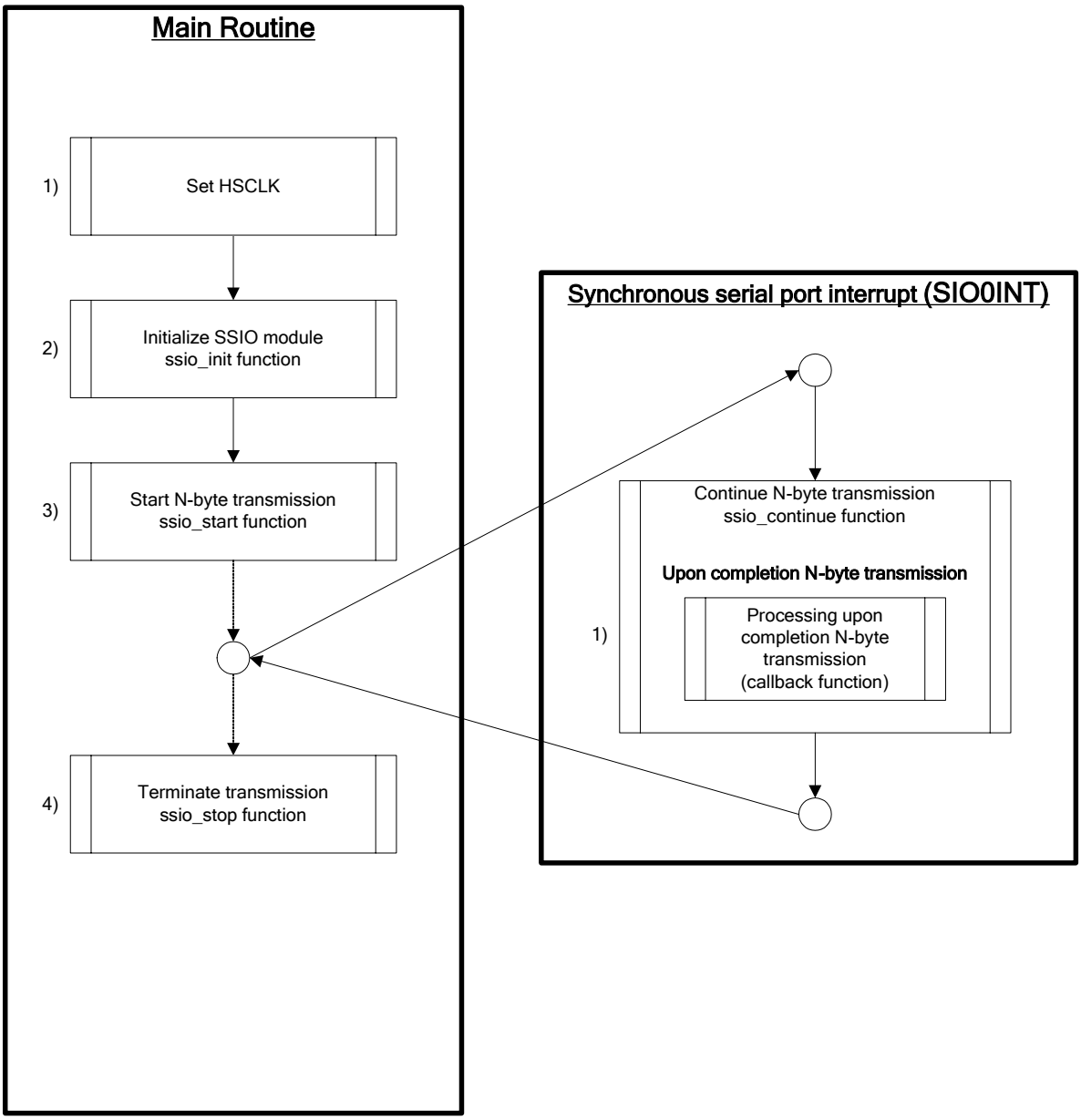


Figure 2-2 SSIO Data Transmission Procedure

[Main Routine]

1) Set HSCLK.
   - If HSCLK is selected as the transfer clock, it is necessary to set HSCLK configuration before SSIO module initialization.

2) Initialize the SSIO module.
   - Set the following communication conditions and initialize the SSIO module:
     ① Select the transfer clock from the table below.

1	LSCLK（32KHz）
2	1/2 LSCLK（16KHz）
3	1/4 HSCLK（125KHz@500KHz）
4	1/8 HSCLK（62.5KHz@500KHz）
5	1/16 HSCLK（31.25KHz@500KHz）
6	1/32 HSCLK（15.765KHz@500KHz）
7	EXCLK 0（P41）
8	EXCLK 1（P45）

     * To initialize MCU as master mode, select one of the numbers from 1 to 6. To initialize as slave mode, select the number 7 (P42/P41/40 pin is used) or the number 8 (P46/P45/44 pin is used).

     ② Select the clock output phase (“H” or “L” for the default level).
     ③ Select the bit order (LSB first or MSB first).
     ④ Select the buffer length (8bit or 16bit).

3) Start N-byte transmission.
   - Specify the following transmit data information in the designated parameters of the **ssio_start** function and start transmission.
     ① Operation mode, that is, “Transmission”
     ② Initial address of the area that contains transmit data
     ③ Transmit data size (in bytes)
       * If the buffer length is 16 bit, specify the value by caluculating that 1 word equals 2 bytes.
       For example, in the case of 10 word data tramsmission, the transmit data size is 20 bytes.
     ④ Processing to be executed upon completion of transmission of N bytes of data (callback function specified)

4) Terminate transmission.
   - Terminate transmission using the **ssio_stop** function. Transmission can be terminated whether in the middle of N-byte transmission or after N-byte transmission completion.

[Synchronous serial port Interrupt (SIO0INT)]

1) Continue N-byte data transmission
   - Transmits data each time the **ssio_continue** function is executed based on the communication data information specified in step 3) above, “Start N-byte transmission”, by the ssio_start function. When 1 byte of data is transmitted, the Synchronous serial port Interrupt interrupt occurs again at the time of transmission termination of that 1-byte data. This will continue N-byte data transmission.
   - When N-byte data transmission is completed, the “Processing to be executed upon completion of transmission of N bytes of data (callback function)” specified in “Start N-byte transmission” (ssio_start function) above, is executed.

2.1.3.2. Reception Procedure

Shown below is the procedure for performing data reception using the SSIO module.

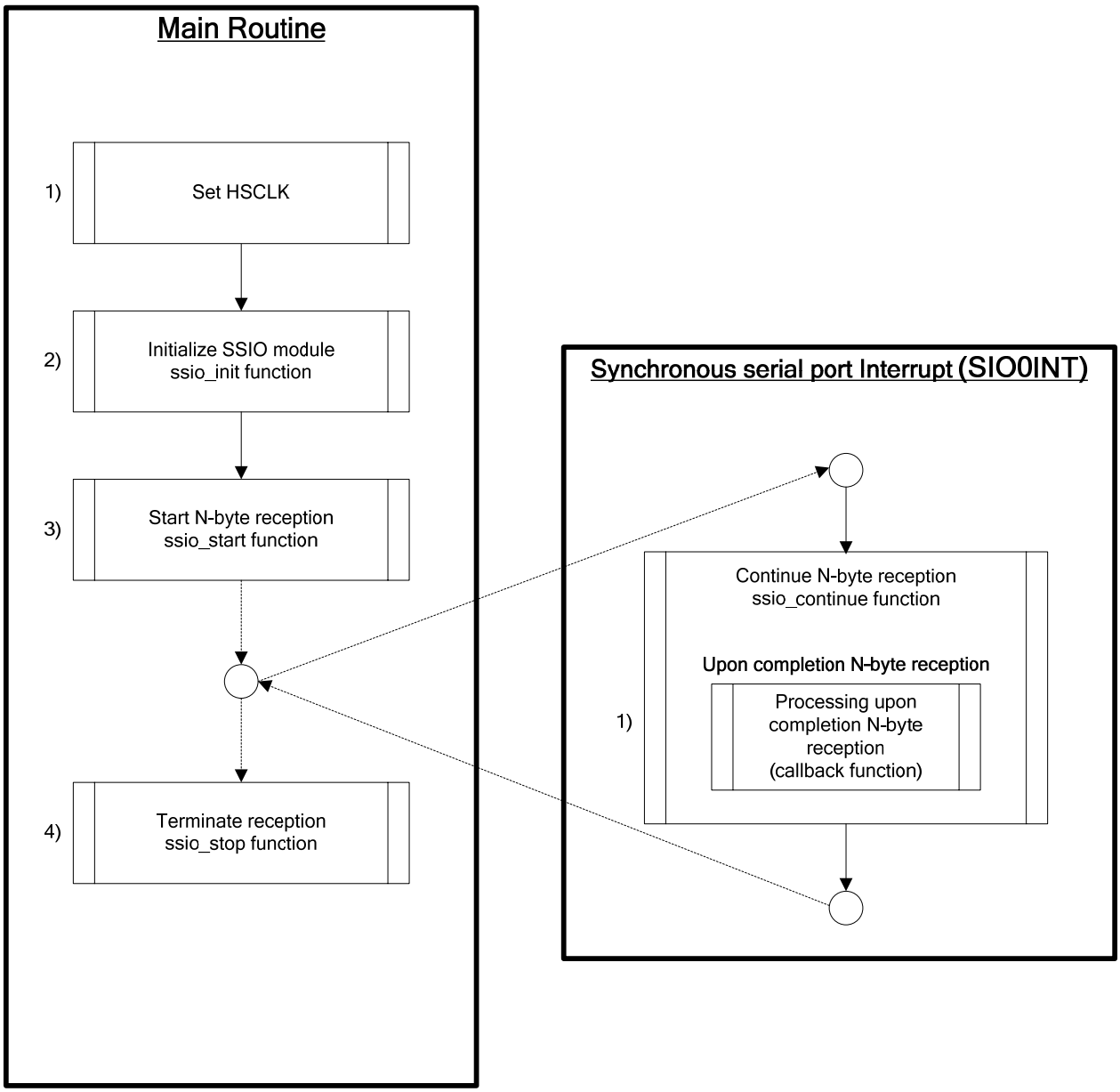


Figure 2-3 SSIO Data Reception Procedure

[Main Routine]

1) Set HSCLK.
   - If HSCLK is selected as the transfer clock, it is necessary to set HSCLK configuration before SSIO module initialization.

2) Initialize the SSIO module.
   - Set the following communication conditions and initialize the SSIO module:
     ① Select the transfer clock from the table below.

1	LSCLK（32KHz）
2	1/2 LSCLK（16KHz）
3	1/4 HSCLK（125KHz@500KHz）
4	1/8 HSCLK（62.5KHz@500KHz）
5	1/16 HSCLK（31.25KHz@500KHz）
6	1/32 HSCLK（15.765KHz@500KHz）
7	EXCLK 0（P41）
8	EXCLK 1（P45）

     * To initialize MCU as master mode, select one of the numbers from 1 to 6. To initialize as slave mode, select the number 7 (P42/P41/40 pin is used) or the number 8 (P46/P45/44 pin is used).

     ② Select the clock output phase (“H” or “L” for the default level).
     ③ Select the bit order (LSB first or MSB first).
     ④ Select the buffer length (8bit or 16bit).

3) Start N-byte reception.
   - Specify the following receive data information in the designated parameters of the **ssio_start** function and start transmission.
     ① Operation mode, that is, “Reception”
     ② Initial address of the area that contains receive data
     ③ Receive data size (in bytes)
       * If the buffer length is 16 bit, specify the value by caluculating that 1 word equals 2 bytes.
       For example, in the case of 10 word data reception, the receive data size is 20 bytes.
     ④ Processing to be executed upon completion of reception of N bytes of data (callback function specified)

4) Terminate reception.
   - Terminate reception using the **ssio_stop** function. Reception can be terminated whether in the middle of N-byte reception or after N-byte reception completion.

[Synchronous serial port Interrupt (SIO0INT)]

1) Continue N-byte data reception
   - Receives data each time the **ssio_continue** function is executed based on the communication data information specified in step 3) above, “Start N-byte reception”, by the ssio_start function.
   - When N-byte data reception is completed, the “Processing to be executed upon completion of reception of N bytes of data (callback function)” specified in “Start N-byte reception” (ssio_start function) above, is executed.

2.1.3.3. Transmission/Reception Procedure

Shown below is the procedure for performing data transmission/reception (Bi-directional comminucation) using the SSIO module.

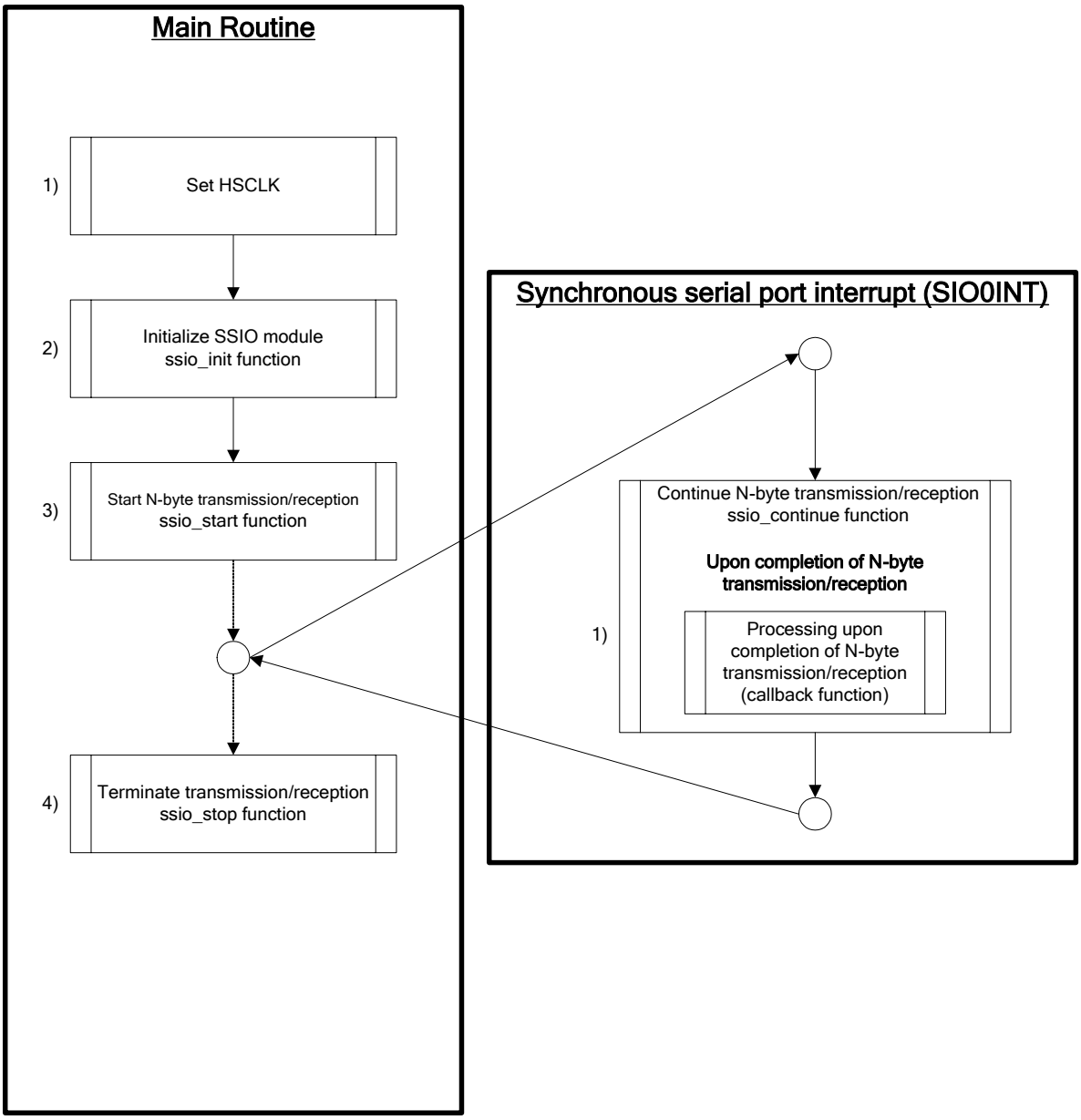


Figure 2-4 SSIO Data Transmission/Reception Procedure

[Main Routine]

1) Set HSCLK.
   - If HSCLK is selected as the transfer clock, it is necessary to set HSCLK configuration before SSIO module initialization.

2) Initialize the SSIO module.
   - Set the following communication conditions and initialize the SSIO module:
     ① Select the transfer clock from the table below.

1	LSCLK（32KHz）
2	1/2 LSCLK（16KHz）
3	1/4 HSCLK（125KHz@500KHz）
4	1/8 HSCLK（62.5KHz@500KHz）
5	1/16 HSCLK（31.25KHz@500KHz）
6	1/32 HSCLK（15.765KHz@500KHz）
7	EXCLK 0（P41）
8	EXCLK 1（P45）

     * To initialize MCU as master mode, select one of the numbers from 1 to 6. To initialize as slave mode, select the number 7 (P42/P41/40 pin is used) or the number 8 (P46/P45/44 pin is used).

     ② Select the clock output phase (“H” or “L” for the default level).
     ③ Select the bit order (LSB first or MSB first).
     ④ Select the buffer length (8bit or 16bit).

3) Start N-byte transmission/reception.
   - Specify the following transmit/receive data information in the designated parameters of the **ssio_start** function and start transmission.
     ① Operation mode, that is, “Transmission/Reception”
     ② Initial address of the area that contains transmit data
     ③ Initial address of the area that contains receive data
     ④ Transmit/Receive data size (in bytes)
       * If the buffer length is 16 bit, specify the value by caluculating that 1 word equals 2 bytes.
       For example, in the case of 10 word data transmission/reception, the data size is 20 bytes.
     ⑤ Processing to be executed upon completion of transmission/reception of N bytes of data (callback function specified)

4) Terminate transmission/reception.
   - Terminate transmission/reception using the **ssio_stop** function. Transmission/reception can be terminated whether in the middle of N-byte transmission/reception or after N-byte transmission/reception completion.

[Synchronous serial port Interrupt (SIO0INT)]

1) Continue N-byte data transmission/reception
   - Transmits and receives data each time the **ssio_continue** function is executed based on the communication data information specified in step 3) above, “Start N-byte transmission/reception”, by the ssio_start function. When 1 byte of data is transmitted and received, the Synchronous serial port Interrupt interrupt occurs again at the time of transmission/reception termination of that 1-byte data. This will continue N-byte data transmission/reception.
   - When N-byte data transmission/reception is completed, the “Processing to be executed upon completion of transmission/reception of N bytes of data (callback function)” specified in “Start N-byte transmission/reception” (ssio_start function) above, is executed.

3. Description of the Sample Program

3.1. Operation conditions

1) System clock
   - SYSCLK=HSCLK (RC oscillation mode 500 kHz)

* 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 can be changed its operation mode, depending on the setting of communication parameters.

The communication parameters can be set by changing the compile options and the macro (#define) definitions. The example is shown below.

1. Change of master/slave
When a compile option defines "_SSIO_MASTER_MODE", the program runs as a master.

2. Specification in initial operation mode. (e.g. transmission or reception mode)
The definition of "_SSIO_TRANS_MODE_INI" in the compile option specifies the first operation mode. The following shows the value which is defined as "_SSIO_TRANS_MODE_INI" (it corresponds to the value which is set to S0MD1 and S0MD0.)

0:	The SSIO function cannot be used.
1:	The SSIO operation mode of 1(=receive) can be used.
2:	The SSIO operation mode of 2(=transmit) can be used.
3:	The SSIO operation mode of 3(=transmit/receive) can be used.

3. Communication setup
The following macro definitions in main.c are changed for communication setup.

Clock:	SSIO_SETTING_CLOCK
Clock phase:	SSIO_SETTING_CLOCK_TYPE
LSB/MSB first:	SSIO_SETTING_ENDIAN
Buffer mode:	SSIO_SETTING_BUFFER_MODE

The values, which can be used to define the above macro, are defined in ssio.h. (SSIO_CLK_HS4 etc.)

4. Transmitting and receiving data setting

The number of transmitting data :	Specified as the definition value of "SSIO_TX_SIZE". (In the case of the 16-bit mode, the number of WORD is specified.)
The number of receiving data :	Specified as the definition value of SSIO_RX_SIZE. (In the case of the 16-bit mode, the number of WORD is specified)
Transmitting data :	Specified as the data in _ssioTxBuf/_ssioTxWordBuf variable. It can be changed into arbitrary values.

Revision History

Revision History

Edition	Date	Page		Description
		Previous Edition	Current Edition	
1	2009.6.26	–	–	First edition
2	2010.4.16	5	5	List of Folders and Files is updated.
		6	6-7	Build procedure is updated.
		–	8-9	Description of Restrictions is added.