

U16

ISP(In-system Programming)

Sample Program AP Note

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 2014 LAPIS Semiconductor Co., Ltd.

Table of contents

1. INTRODUCTION.....	4
1.1. RELATED DOCUMENTS.....	4
1.2. SUPPORT MCU	4
2. ISP PROGRAM OVERVIEW	5
2.1. HARDWARE CONFIGURATION.....	5
2.2. SOFTWARE CONFIGURATION	6
2.3. SPECIFICATION OF THE COMMUNICATION.....	9
2.4. MEMORY LOCATION	10
2.4.1. Remap terminal=Low	10
2.4.2. Remap terminal=High	11
2.4.3. Remap by the program.....	11
2.5. SAMPLE PROGRAM OPERATION	12
2.6. SOFTWARE CONFIGURATION FIGURE.....	13
2.6.1. Boot loader	13
2.6.2. ISP program	13
2.7. OPERATING THE SAMPLE PROGRAM	14
2.7.1. Sample program building	14
2.7.2. Sample program HEX combination.....	15
2.7.3. Sample program download	17
2.7.4. Binary file creation	19
2.7.5. File transfer by using XMODEM.....	20
3. TO START ISP PROGRAM FROM YOUR SYSTEM	22

1. Introduction

ISP (In system programming) is the software which rewrites the program of flash memory in U16. This AP notes describes the overview and the basic use of ISP program.

1.1. Related Documents

- Reference software for U16 API Manual
- nX-U16/100 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
- FWuEASE Flash Writer Host Program User's Manual

1.2. Support MCU

This system supports the the below MCUs.

- ML620Q154/Q155/Q156
- ML620Q504

2. ISP program Overview

2.1. Hardware Configuration

The hardware configuration of this system is as follows.

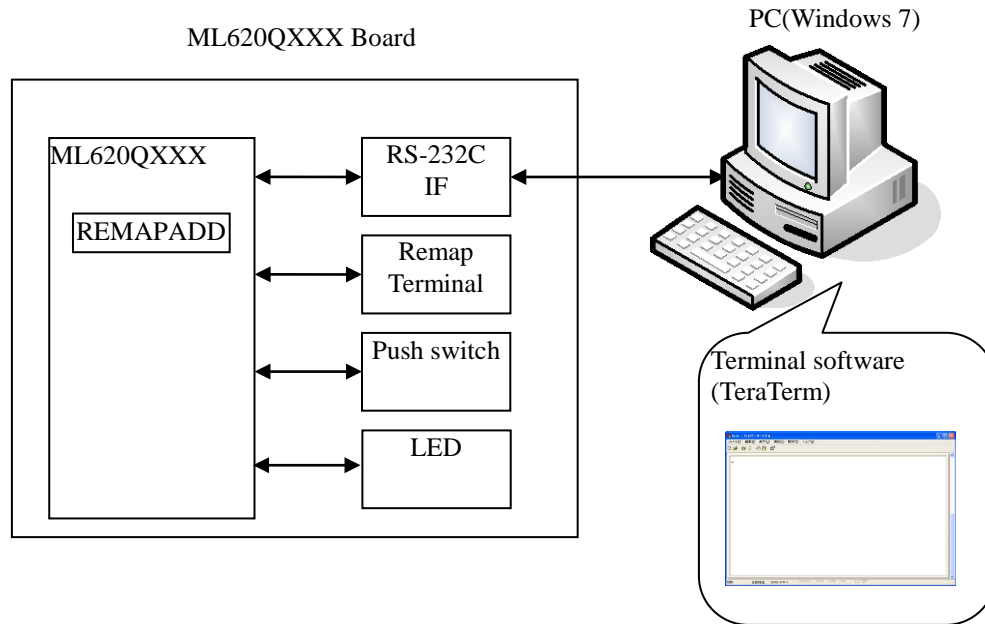


Figure 2-1 Hardware Configuration

Table 2-1 the specification of the hardware configuration

Item	number	Description
REMAPADD	1	REMAPADD is used to set the restart address of the flash memory space at the software reset (*1).
Switch	1	Switch is used to start ISP program by the sample program.
RS-232C interface	1	RS-232C interface is used to communicate with PC.
LED	1	LED lights up when an error occurs.
Remap terminal	1	Remap terminal is used to select whether to start the boot loader program or to start the user program at the hardware reset (*2).

*1: The software reset means to reset the U8 core by using BRK instruction in privilege level.

*2: Remap terminal has the case of not mounting, according to MCU. Refer to the following tables.

Table 2-2 Supported Remap terminal

MCU	Remap terminal	
	Available	Not available
ML620Q154/Q155/Q156		✓
ML620Q504	✓	

Table 2-3 Pins(ML620Q154/Q155/Q156)

Pin	Description
P02	UART RXD
P43	UART TXD
P20	LED
P00	Switch
*1	Remap terminal

*1 : ML620Q154/Q155/Q156 is not supporting the Remap terminal.

2.2. Software Configuration

The sample program consists of the below programs.

Table 2-4 sample program

No	Program Name	Description
1	Boot loader program	The boot loader program boots up the ISP program.
2	ISP program	The ISP program is able to program the user program which is received via UART to the flash memory.
3	User program	The user program is the program which operates on user area. In this sample program, this program boots up the ISP program when the switch is pushed..

The composition of this package is as follows.

U16_ISP

```

|- License_e.pdf           Software License Agreement (English)
|- License_j.pdf           Software License Agreement (Japanese)
|- readme(E).txt           README(English)
|- readme(J).txt           README(Japanese)
|
|- DOC                     Document folder
|   |- FEXT_U16_ISP_FLASH_AP-01.pdf   ISP Sample Program AP Notes(English)
|   |- FJXT_U16_ISP_FLASH_AP-01.pdf   ISP Sample Program AP Notes(Japanese)
|
|- SampleSoft
|   |- BOOTLOADER           Bootloader
|       |- _output
|           |- _obj
|       |- main
|           |- main.c
|           |- vector_bootloader.h
|       |- bootloader_amall.PID         Project file(SMALL model)
|
|   |- common                 Folder for common
|       |- mcu_small
|           |- mcu.h
|       |- isp.h
|       |- led.h
|       |- remap.c
|       |- remap.h
|       |- sys_clock.c
|       |- sys_clock.h
|       |- sys_uart.c
|       |- sys_uart.h
|
|   |- driver                 Peripheral driver
|       |- clock
|           |- clock_i.h
|       |- common
|           |- rdwr_reg.h
|       |- flash
|           |- flash.c
|           |- flash.h
|       |- irq
|           |- irq_i.h
|       |- tbc
|           |- tbc.c
|           |- tbc.h
|       |- uart
|           |- uart_common_i.h
|           |- uart0_i.c
|           |- uart0_i.h
|       |- wdt
|           |- wdt.c
|           |- wdt.h
|
|- ISP                     ISP program
|   |- _output
|       |- _obj
|   |- main
|       |- crc.c
|       |- crc.h
|       |- isp.c
|       |- main.c

```

```

| | |- main.h
| | |- vector_isp.h
| | |- xmodem.c
| | |- xmodem.h
| |- isp_small.map
| |- isp_small.PID
|
| Project file (SMALL model)
|
|- obj
|   Object file
|   |- isp_small.res
|   |   Response file for HTU8 (ML620Q156)
|   |   Hex file after combination(Refer to chapter 2.7.2)
|   |- ML620Q154_000RA.HEX
|   |- ML620Q155_000RA.HEX
|   |- ML620Q156_000RA.HEX
|   |- ML620Q504_000RA.HEX
|   |
|   |   Binary file for downloading(Refer to chapter 2.7.4)
|   |- update_user_ML620Q154.bin
|   |- update_user_ML620Q155.bin
|   |- update_user_ML620Q156.bin
|   |- update_user_ML620Q504.bin
|   |- updated_user_small.res
|
|- SAMPLE_USER
|   User program
|   |- _output
|   |   |- _obj
|   |   |- main
|   |   |   |- main.c
|   |   |   |- main.h
|   |   |   |- vector_user.h
|   |- update_user_small.map
|   |- update_user_small.PID
|   |- user_small.map
|   |- user_small.PID
|
|- startup
|   Startup file
|   |- ML620154.ASM
|   |- ML620155.ASM
|   |- ML620156.ASM
|   |- ML620504.ASM
|   |- ML620504_BOOTLOADER.ASM
|
|- readme.txt

```


2.3. Specification of the communication

The specification of the communication with PC is as follows.

Table 2-5 communication parameter

Communication speed	115200bps
Data length	8bit
Parity bit	none
Stop bit	1bit
Flow control	none

Table 2-6 communication protocol

Protocol	XMODEM
Block size	128 bytes
Error detection	CRC16

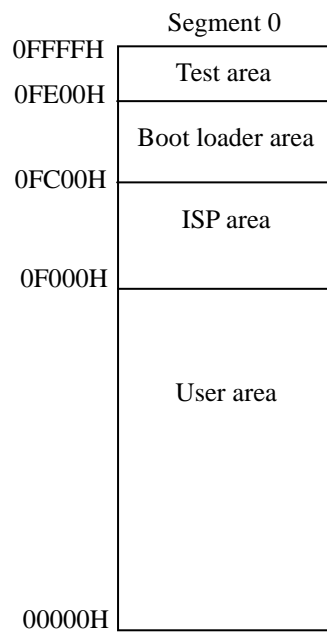
2.4. Memory location

2.4.1. Remap terminal=Low

The memory location when the hardware is reset after the remap terminal is low level is as follows. However, the start address of ISP area changes with target device. The range of ISP area of each device is shown in the following tables. This AP note is indicated about the case where the start address of ISP area is 0F000H. Please refer to the User's Manual of the device to use about the size of a boot loader area.

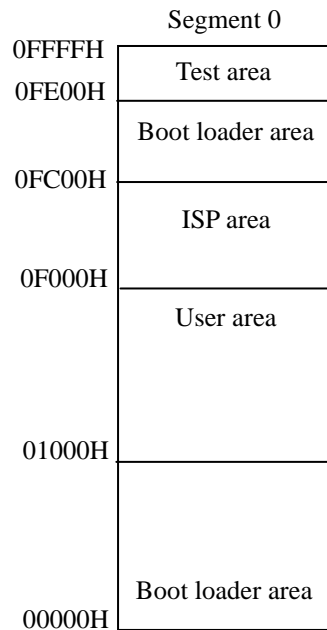
Table 2-7 Range of ISP area

MCU	Range of ISP area	note
ML620Q156 ML620Q504	0xF000 ~ 0xFBFF	
ML620Q155	0xB000 ~ 0xBBFF	
ML620Q154	0x7000 ~ 0x7BFF	



2.4.2. Remap terminal=High

The memory location when the hardware is reset after the remap terminal is high level is as follows.



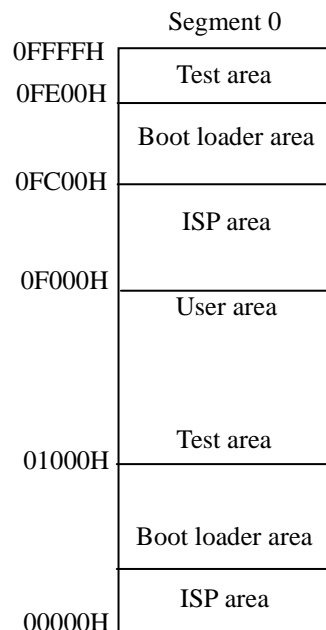
[Notes]

The following MCUs are not equipped with the Remap terminal.

ML620Q154/Q155/Q156

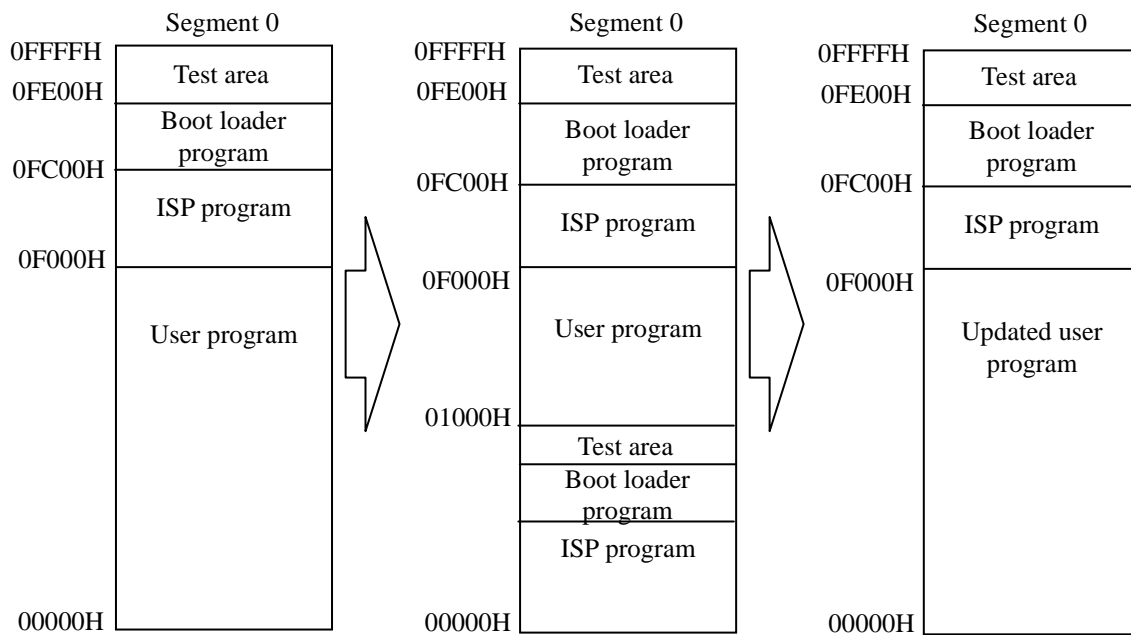
2.4.3. Remap by the program

The memory location when the software resets the hardware by controlling the REMAPADD register is as follows. The memory address (unit of 4K bytes) specified with REMAPADD register is remapped to the restart address. This figure is the example when the program specifies the ISP area (0F000H) to the REMAPADD register.



2.5. Sample program operation

This chapter describes the operation which the ISP program rewrites the user program in the user area.



1. Download the boot loader, ISP program, and user program by the FWuEASE Flash Writer. Refer to the chapter 2.7.1 and 2.7.3.

In this package, HEX file which combined bootloader, ISP program and User program is included.

SampleSoft¥obj¥ML620QXXX_000RA.hex (XXX is MCU name)

2. Push the hardware reset button after the remap terminal is low level.
User program starts and “Sample user program” is displayed on terminal software of PC.

Note: The following MCUs do not have remap terminal, so that user program starts immediately after resetting the hardware.

- ML620Q154/Q155/Q156

3. Push Switch
ISP program starts and “ISP” is displayed on terminal software of PC.
4. Transfer the binary file of the updating user program by XMODEM protocol from PC (Refer to the chapter 2.7.5).
 - (1) In this package, “Updated user program” is included.
SampleSoft¥obj¥update_user_ML620QXXX.bin (XXX is MCU name)
 - (2) Refer to the capter 2.7.4 for the method of creation of binary file
 - (3) When the following errors occur, LED is turned on and ISP program goes into an error condition (infinite loop).
 - The number of times of a retry by communication error (reception timeout is included) exceeded 10 times continuously.
 - The rewriting error to a flash memory occurred (a verification error or the writing more than flash memory size).
5. After updating the flash memory, the updating program starts and “Updated sample user program” is displayed on terminal software of PC.

2.6. Software Configuration figure

The software configuration figure is as follows.

2.6.1. Boot loader

The software configuration of the boot loader is as follows.

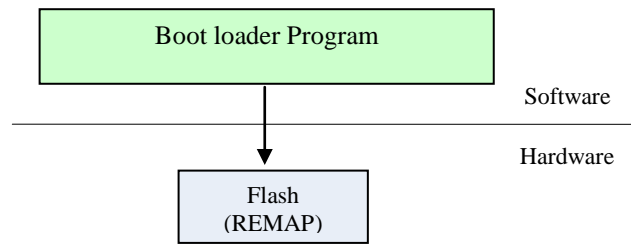


Figure 2-2 boot loader configuration

2.6.2. ISP program

The software configuration of the ISP program is as follows.

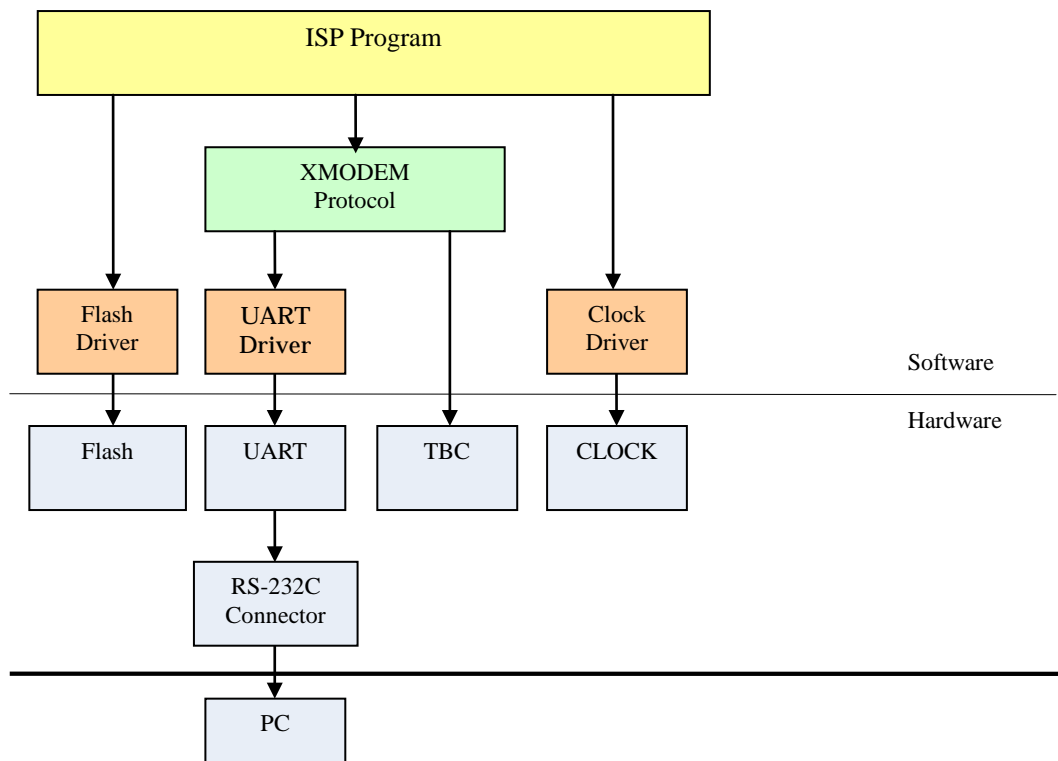


Figure 2-3 ISP program configuration

2.7. Operating the sample program

2.7.1. Sample program building

The each sample programs have a project file. You can build it by the project file. The hex file is created after building these. Refer to the chapter 2.7.4 to convert the hex file to the binary file

Table 2-8 project file

Program name	Project name
Boot loader program	bootloader_small.PID bootloader_large.PID
ISP program	isp_small.PID isp_large.PID
User program	user_small.PID user_large.PID
User program(for update)	update_user_small.PID update_user_large.PID

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 "XXX_Small.PID". In the case of large model, the project file is "XXX_Large.PID".
2. In the default setting in the small model, ML620Q156 is set as the target MCU.
If your target MCU is different, follow the procedure below to change the setting.
 - (1) Remove the startup file "ML620156.asm" registered in the file tree of IDEU8. Instead of that, register your target MCU's startup file. (In the case of ML620QXXX, it is S620XXX.asm.)
 - (2) Select the menu [Project] -> [Options] -> [Compiler/assembler].
 - (3) In the displayed window, select the target MCU from the "Target microcontroller" list in the "General" tab.
 - (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.

_ML620QXXX

 About the "XX" part, replace with the type number of MCU
 For example, if ML620Q156 is used, define the following macro.

_ML620Q156

 In the case that the macro other than the type number in the chapter 1.1 is defined, or the case that the memory model that is supported by PID file is different from the memory model of MCU, the compiler issues the following error message.
 Error : E2000 : #error : "Unknown target MCU"
3. Select the menu "Project" -> "Rebuild". Then the build processing 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 obj folder.

2.7.2. Sample program HEX combination

The allocations of three programs (Boot loader program, ISP program, and User program) are specified and one HEX file is created by HTU8. The usage of HTU8 is shown below.

1. In the case allocated as (User program to 0x0000 - 0EFFFH, ISP program to 0xF000 - 0FBFFH, boot loader program to 0xFC00 - 0FDDFH, code option data to 0xFDE0 - 0FDE1H), a response file is created as follows.

```
//Response file(sample.res)

USER_SMALL.HEX /AL(0H, 0EFFFH, 0H)           // 0 - 0EFFFH of the User program is allocated to 0H.
(1)
ISP_SMALL.HEX /AL(0H, 0BFFH, 0F000H)         // 0 - 0BFFH of the ISP program is allocated to 0F000H.
(2)      (3)
//BOOTLOADER_SMALL.HEX /AL(0H, 01DFH, 0FC00H) // 0 - 01DFH of the boot loader program is allocated to 0FDE0H.

USER_SMALL.HEX /AL(0FDE0H, 0FDE1H, 0FDE0H) // The code option data in a user program is arranged to 0FDE0H.
// (Unnecessary, when not specifying code option data)

/TML620156                                // Target device name (In the case of ML620Q156, it is /TML620Q156).

/FML620Q156_000                          // Output file name

/OH                                       // Output format (OH hex file)
```

[Notice]

ML620Q154/Q155/Q156 are not equipped with the remap terminal, so that the Boot loader program is commented out in above example.

Please change as follows (1)-(3) of a response file, since an address changes with device.

Table 2-9 Correspondence address of each device

		(1)	(2)	(3)
MCU	ML620Q156	0EFFFH	0BFFH	0F000H
	ML620Q504			
	ML620Q155	0AFFFH		0B000H
	ML620Q154	06FFFH		07000H

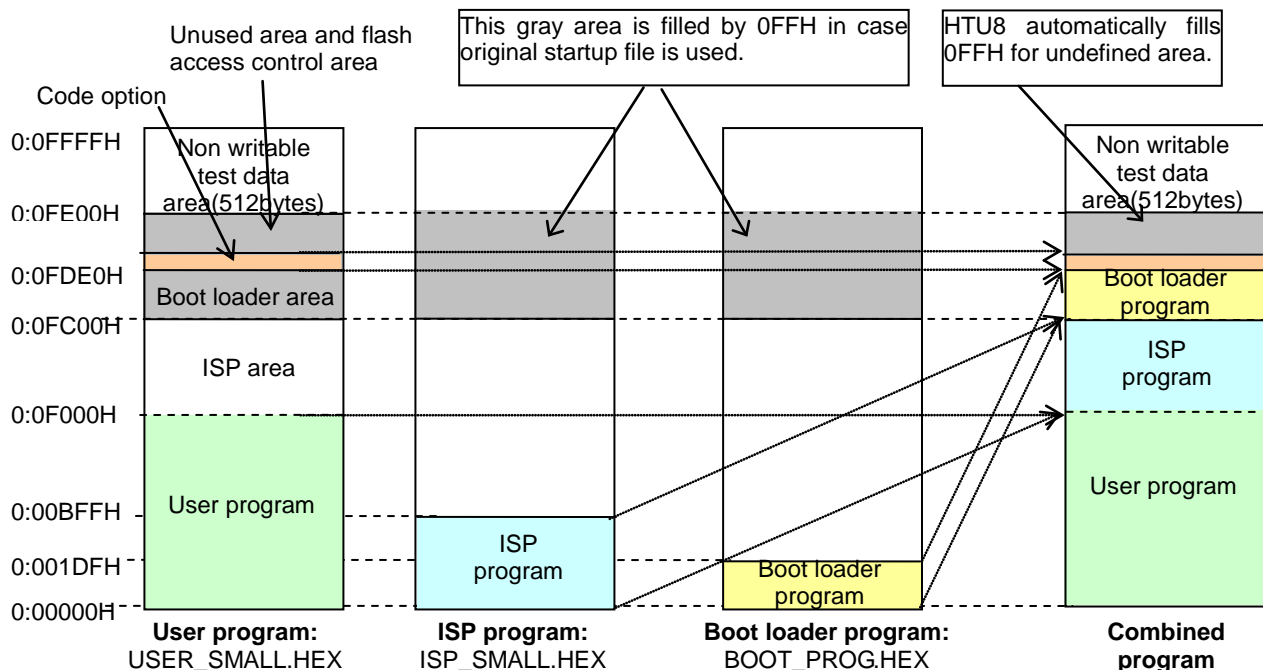


Figure 2-4 Combined memory map image

2. Select “Command-line environment” from [Start] -> [Programs] -> [U8 Tools] -> [nX-U8], and the command prompt for U8 is opened.
3. The input of a command prompt is inputted as follows.

HTU8 @sample.res

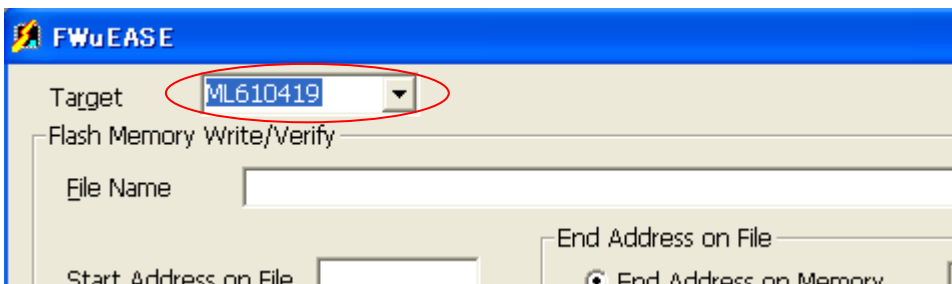
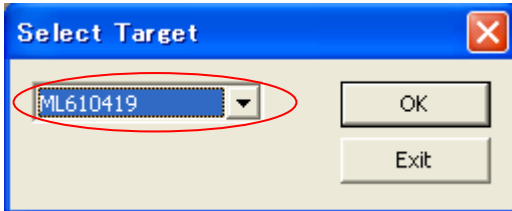
* When the conversion succeeds, HTU8 generates ML620QXXX_000RA.hex and ML620QXXX_000RB.hex. Since these two files have same contents, both files are available.

2.7.3. Sample program download

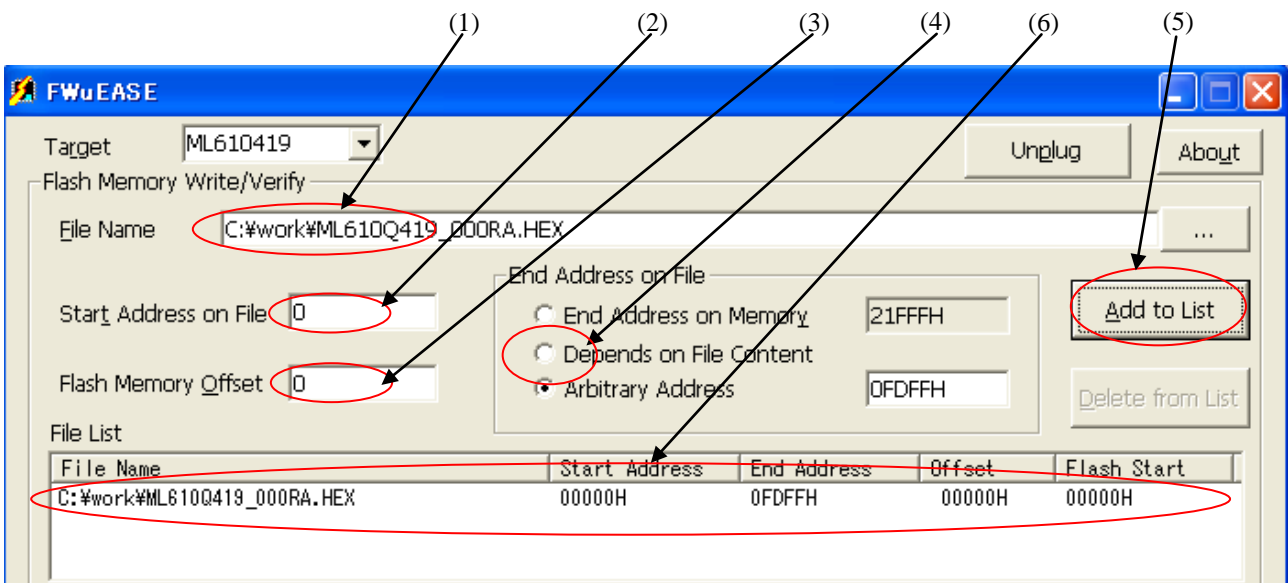
The program is downloaded to the flash memory by the FWuEASE Flash Writer.

The usage of downloading HEX file which is created in “Chapter 2.7.2” is shown below.

1. FWuEASE is started.
2. Target selection
Select MCU

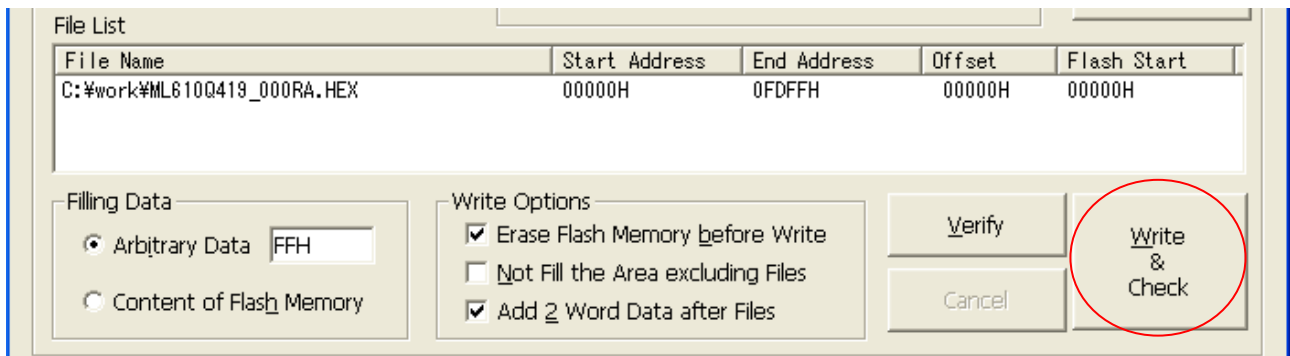


3. Writer-in program registration
 - (1) Select the HEX file that created in chapter 2.7.2
 - (2) Input '0' to [Start Address on File]
 - (3) Input '0' to [Flash Memory Offset]
 - (4) Check [Arbitrary Address] in [End Address on File], and input '0FDFFH'
 - (5) Click [Add to List]
 - (6) Confirm that HEX file is registration in [File List]

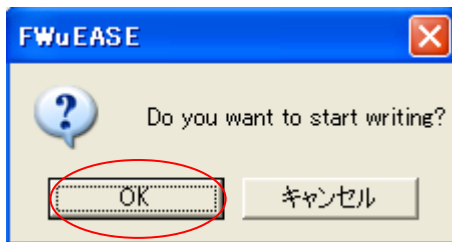


4. Download

Click [Write&Check]

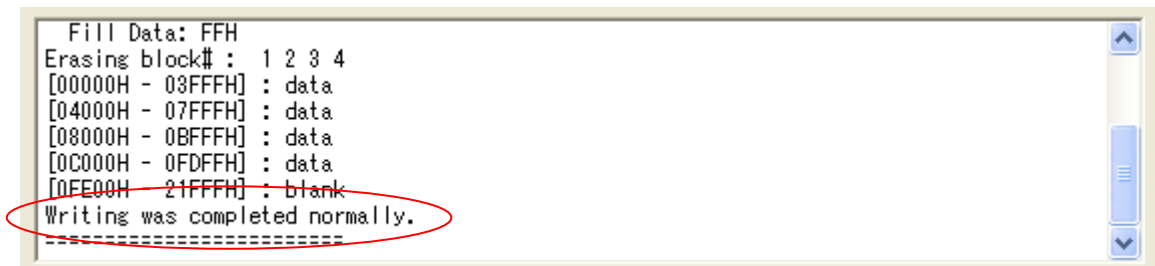


Click [OK] to start downloading



5. Download completion

If downloading is normally completed, then "Writing was completed normally." is displayed



2.7.4. Binary file creation

The format for updating program is supported only Binary file. The below is a description about conversion to binary file and the procedure of file division.

This system prepares "update_user_small.PID" which is changed the displaying characters to the terminal software. In this chapter, the usage of using this file is described.

1. Conversions from the abs file to the Hex file.
Select the menu [Project] -> [Options] -> [Target] -> [General] tab. In the displayed window, check "Create HEX file" and "Intel Hex" field. And start the build processing, then the Hex file will be generated. At this time, do not check "include debugging information" field.
2. Fill the space which the data of Hex file does not exist with 0FFFFh.
Select "Command-line environment" from [Start] -> [Programs] -> [U8 Tools] -> [nX-U8], and enter the below command line to command prompt for U8.

```
HTU8 UPDATE_USER_SMALL.HEX /TML620156 /OH
```

When the conversion succeed, HTU8 generates FLASH_USER_SMALLRA.hex and FLASH_USER_SMALLRB.hex. Since these two files have same contents, both files are available.

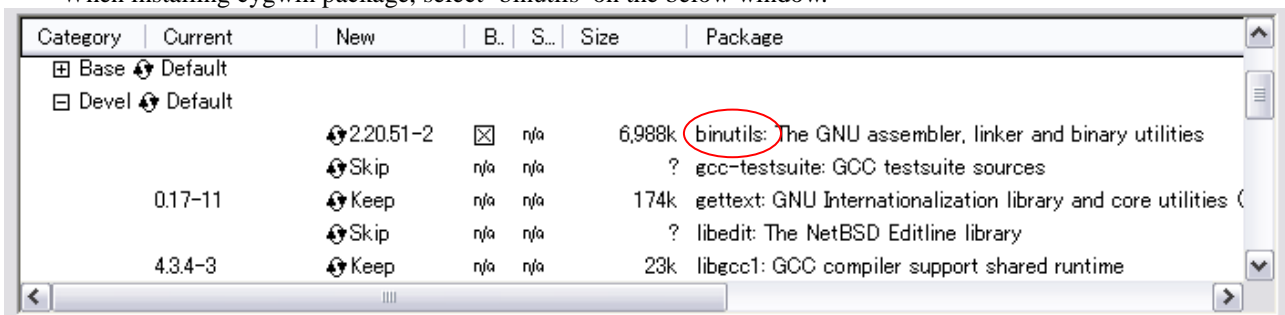
3. Conversion from the Hex file to the binary file.
You will be able to convert from the Hex file to the binary file using 'objcopy' which the cygwin environment provides.
The example of objcopy is as follows.

```
$ objcopy -Iihex -Obinary flashRA.hex flash.bin
```

You will be able to obtain cygwin environment from the below URL.

www.cygwin.com

When installing cygwin package, select 'binutils' on the below window.



[Notice]

The size of the binary file is the same as the flash memory space specified with the procedure. You should delete the data exceeded the ROM size.

Example)

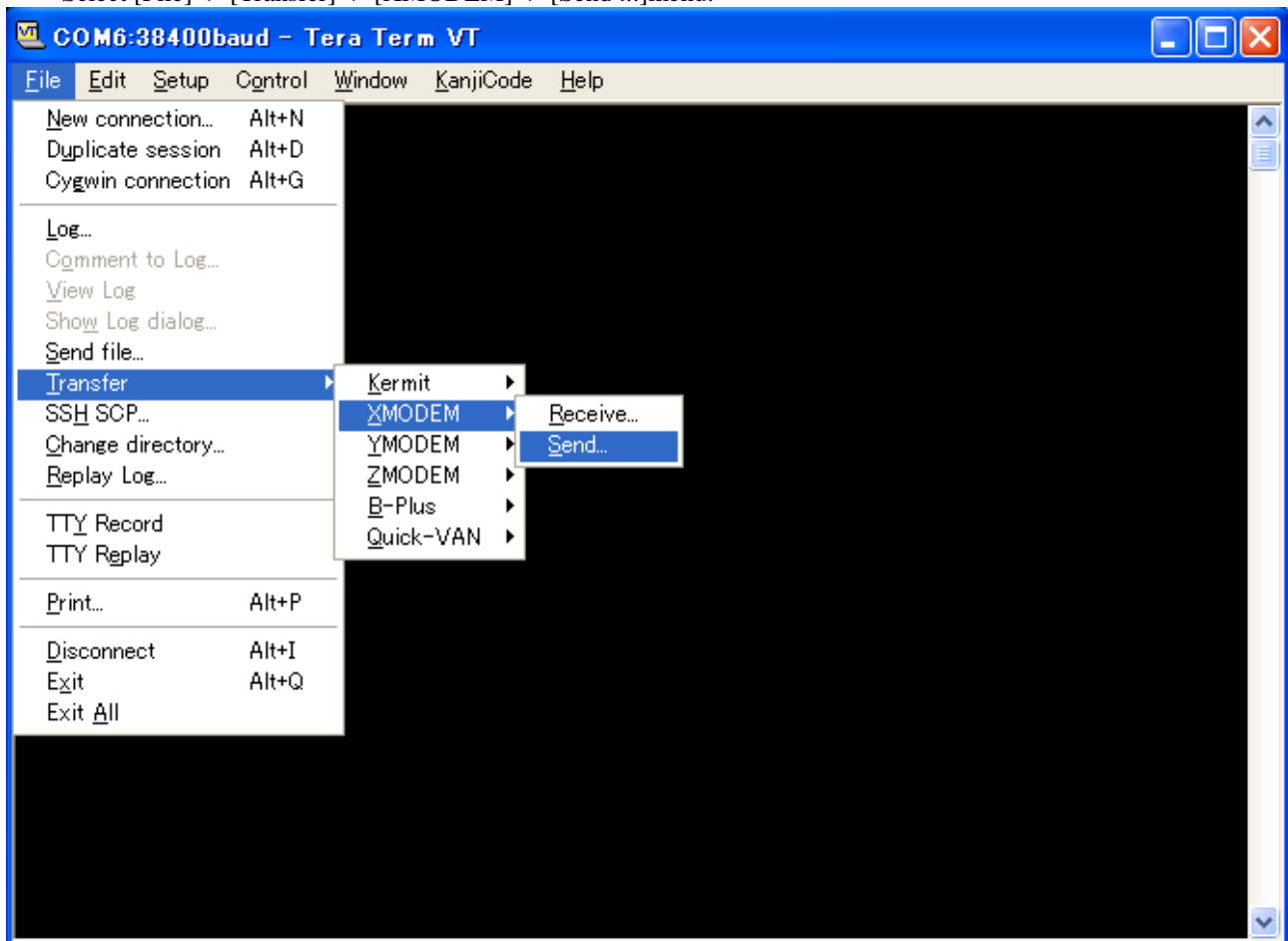
When you use ML620Q156, the ROM size of ML620Q156 is 64K Bytes, so that you should delete the data which exceeded the 64K bytes.

2.7.5. File transfer by using XMODEM

This system transfers the data by using XMODEM protocol.
The way to transfer the data by HyperTerminal is as follows.

1. File transfer protocol selection

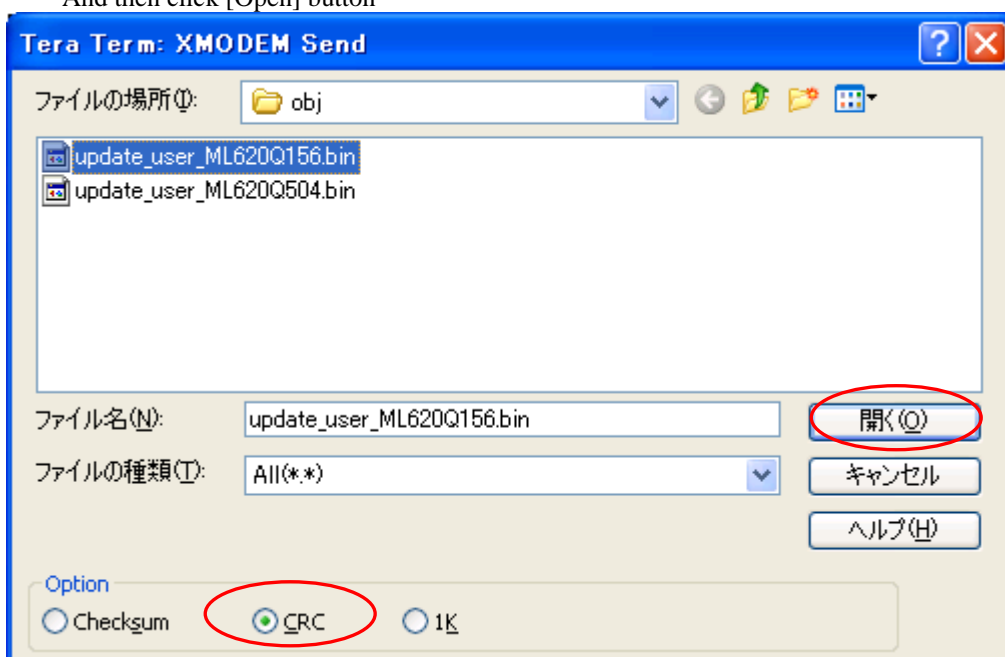
Select [File] -> [Transfer] -> [XMODEM] -> [Send ...] menu.



2. Transfer file and protocol selection

Select the transfer file. Click [CRC]

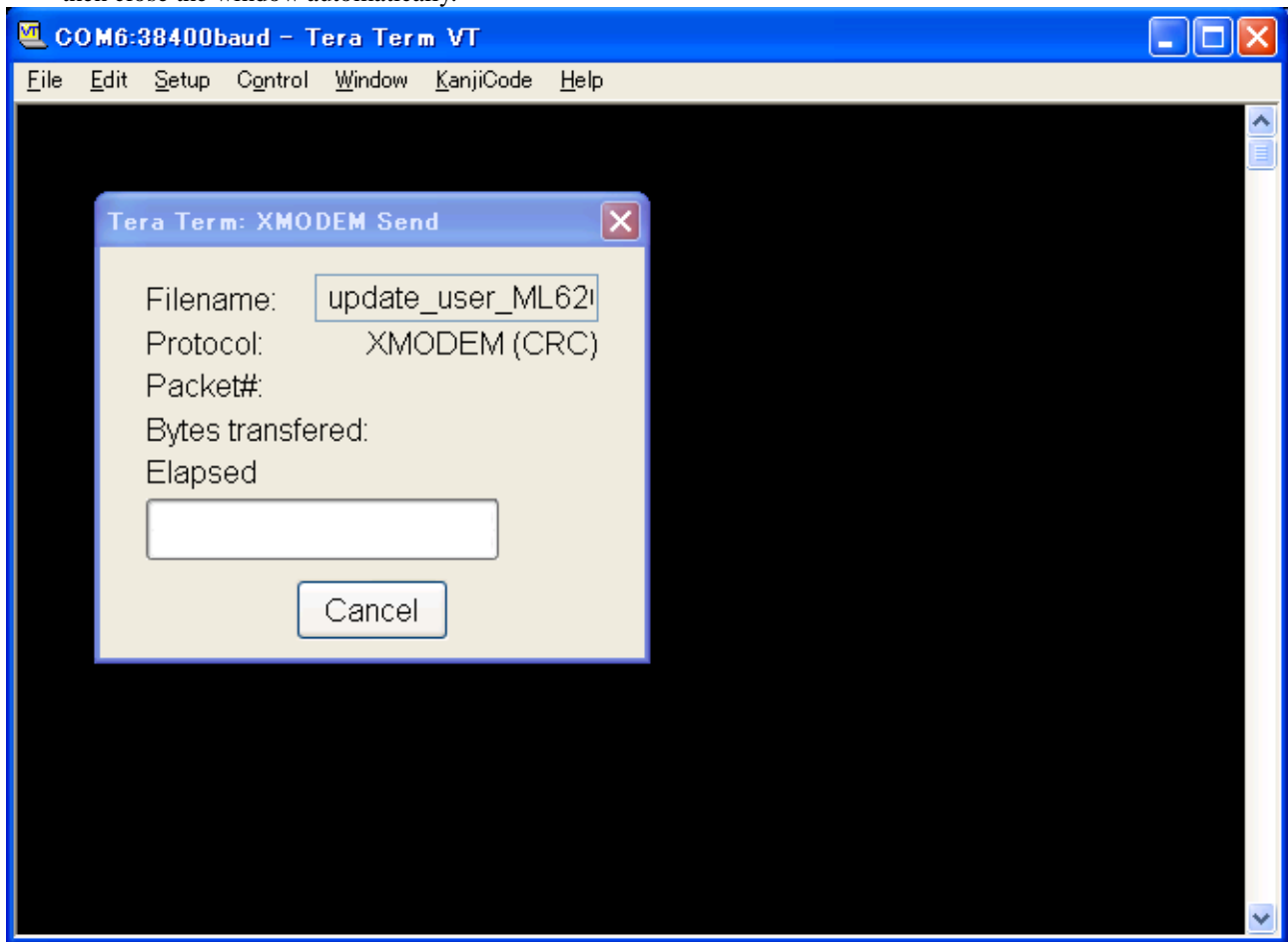
And then click [Open] button



3. Transfer waiting

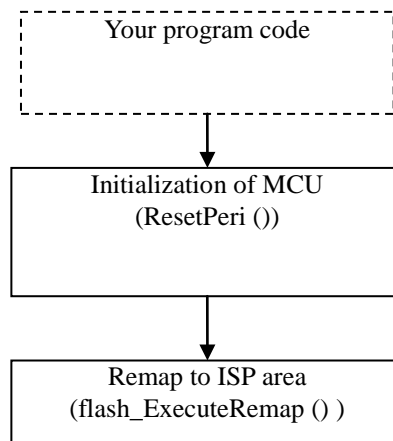
Wait to transfer the file by XMODEM protocol and display the following window.

Start to transfer when hyper terminal receives the request to send from the target board. If the transfer is completed, then close the window automatically.



3. To start ISP program from Your System

The way of change from your system to ISP program is as follows.



ResetPeri function is prepared as sample. Please customize this function according to your system.

Revision history

Document No.	Date	Page		Description
		Previous Edition	Previous Edition	
1	April. 21, 2014	–	–	First edition