

AME-51 Lite Quick Starter Guide

Revision 1.04
2-Oct-05

1. Overview

This document briefly outlines the procedures for configuring the Oki ARM-Based Microcontroller AME-51 Lite Board and executing sample codes on the system. This CPU Board contains a complete development system with the hardware and software necessary for developing and executing applications on Oki ARM-based MCUs. It also provides a convenient platform for developers to evaluate and develop applications on Oki's ARM MCUs without any extra tools.

Note that from hereon, "AME-51 Lite" will be referred to as "CPU board." The corporate names and trade names used in this document are the trademarks or registered trademarks of each respective company.

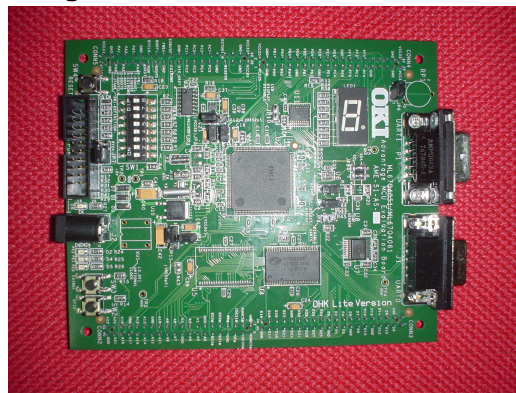


Figure1.1 show the AME-51 Lite CPU board

Package Component

This Development Package for Oki ARM-Based Microcontrollers ML67Q405x/6x contains the following major components.

- ◇ AME-51 Lite CPU Board
- ◇ RS232 serial cable
- ◇ Oki AME-51 Lite CD
- ◇ Quick Start Guide

2. System Requirements

The Software (AME-51GNU and Tera Term) runs on the following platforms.
 Pentium-based personal computer with
 32 megabytes RAM
 CD-ROM drive
 USB port
 Serial port (for programs using UART interfaces)

The package requires one of the following operating systems.
 Windows XP Professional
 Windows 2000 with Service Pack 1 installed
 Windows 98 Second Edition

3. Setting Up Software and Hardware

3.1 Hardware description

3.1.1 Description of Dip Switch

The AME-51 Lite board has one 8-position dip switch SW1.

Position	Name	Description	"ON" Position
1	JTAGE	JTAG enable	Enables JTAG mode
2	BOOTCLK	Boot clock	Enables internal ring oscillator
3	BOOT0	Boot select 0	
4	BOOT1	Boot select 1	Select Bank 0 memory
5	EXIROME	External memory access enable	Enables high speed bus interface
6	EXBUSE	External bus enable	Enables the external bus
7	ROMSEL	ROM select	SRAM set to external ROM Bank
8	FWJ	Flash Write JTAG mode	Enables JTAG flash write mode

Table 3.1 Description of SW1

Basically, there are 4 common setting of SW1

SW1	Setting
JTAG	X
BOOTCLK	X
BOOT0	ON
BOOT1	OFF
EXIROME	OFF
EXBUSE	X
ROMSEL	X
FWJ	OFF

Table 3.2
Serial
Flash Programming

SW1	Setting
JTAG	X
BOOTCLK	X
BOOT0	OFF
BOOT1	OFF
EXIROME	OFF
EXBUSE	ON
ROMSEL	ON
FWJ	OFF

Table 3.3
Standalone

SW1	Setting
JTAG	ON
BOOTCLK	X
BOOT0	OFF
BOOT1	ON
EXIROME	OFF
EXBUSE	ON
ROMSEL	ON
FWJ	OFF

Table 3.4
JTAG debug

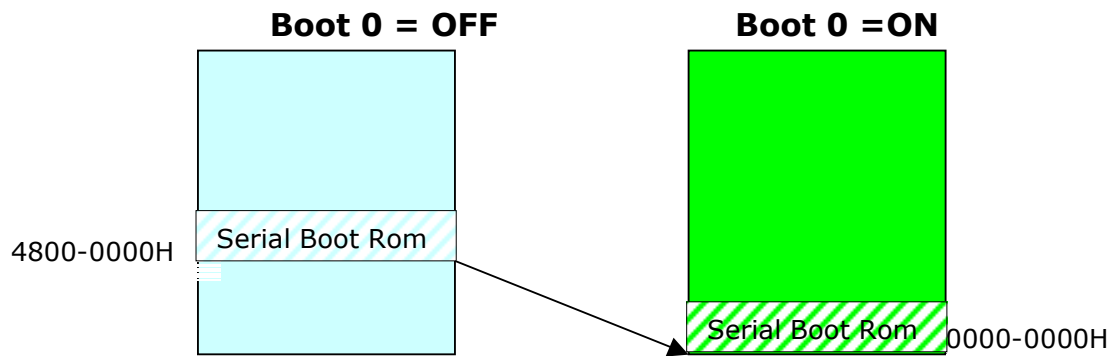
SW1	Setting
JTAG	X
BOOTCLK	X
BOOT0	OFF
BOOT1	ON
EXIROME	OFF
EXBUSE	ON
ROMSEL	ON
FWJ	OFF

Table 3.5
SRAM debug

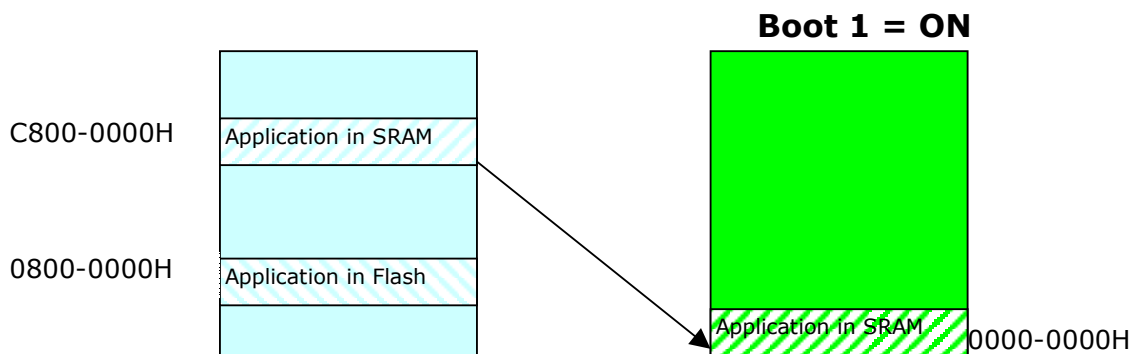
3.1.2 Explanation for memory map in different mode

CPU board use the re-map feature of IC to establish different operation mode. Below explain the memory map after reset on different operation mode. The memory map between JTAG mode and SRAM debug mode is the same, except the JTAG function is turned on.

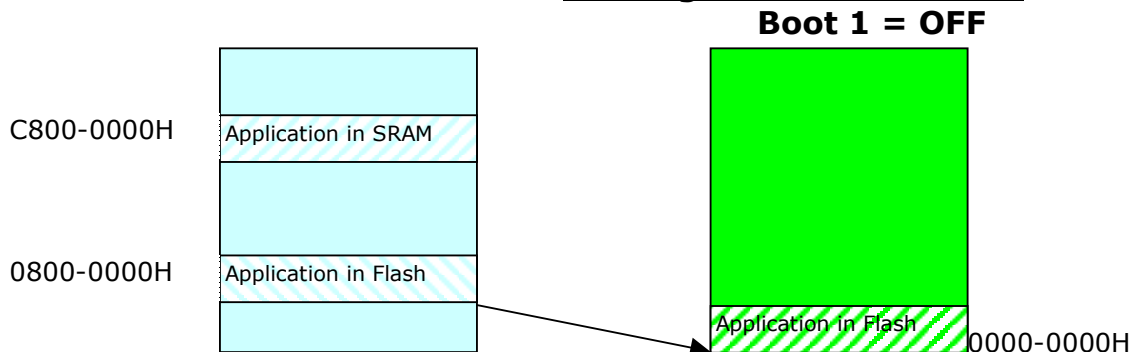
Serial Flash Programming :



SRAM debug when Boot 0 = OFF (ROMSEL=1) running on SRAM:



Standalone when Boot 0 = OFF running on Internal Flash:



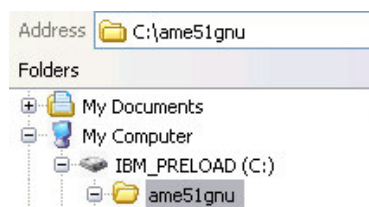
3.2 Software Setup

3.2.1 Installation of AME-51GNU

The ame51gnu software package is the main software part, including GNUARM compiler, SRAM loader firmware and sample programs.

To install this software, just click the "ame51setup.exe" on CD. The setup program will start. It will be installed by following the instructions. You should be reminded that the software must install to C:\. **If you change the drive and path, you need to amend the "ROOT" setting in the "makefiles".**

After you installed the program, you will find the "ame51gnu" directory in C:.



3.2.2 Installation of Tera Term

Tera Term (Pro) is a free software terminal emulator (communication program) for MS-Windows. It supports VT100 emulation, telnet connection, serial port connection, and so on. The Hyper Terminal program included with window can be used as well.

This package will include the Tera Term Pro ver. 2.3 for Windows 95/NT .

3.2.3 General Setup of Tera Term

When you open the Term Term, Figure 4.3.2 will be given out. You can choose serial and then select the COM port .

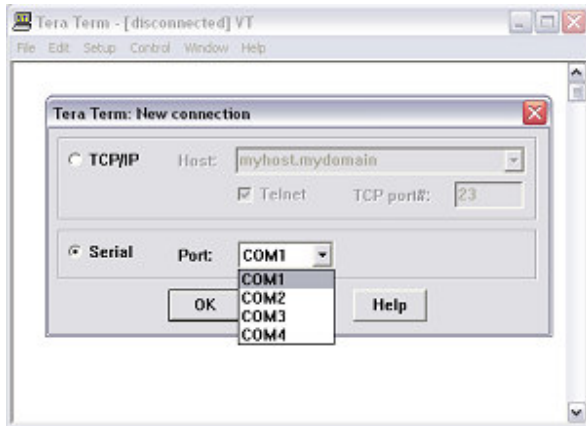


Figure 4.3.2 Selection of COM port

After selection of COM port, you are advised to set up the features of Serial Port. You can do this by click the *Setup\Serial port*. The Figure 4.4.3 will be given out.

You can adjust your required serial port transmission settings within this table.

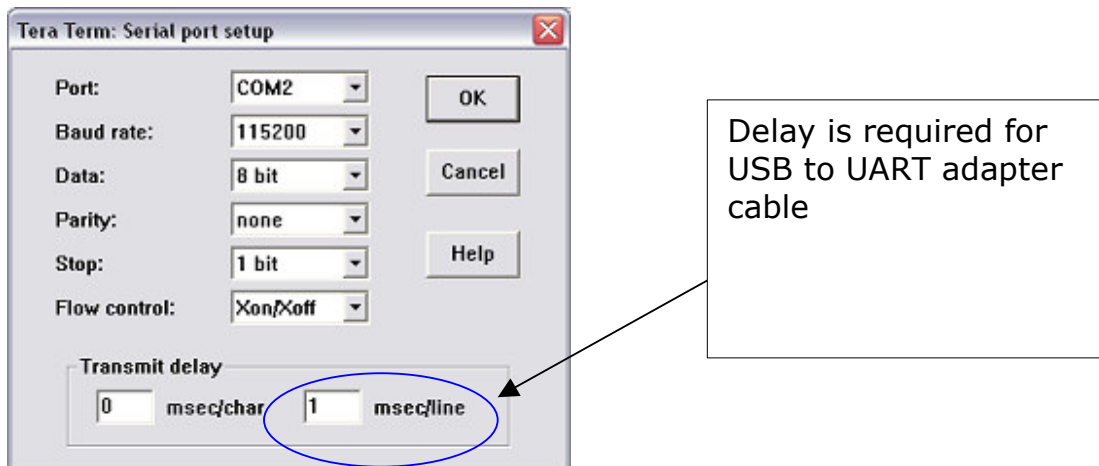


Figure 4.3.3 Setup Serial port features

3.3 Hardware Setup

The AME-51 Lite board has preinstalled SRAM downloader firmware in factory. Once connect, it can immediately start debugging.

1. We can connect the CPU board to PC by serial cable through RS232C1CN connector. The serial cable must provide a straight pin configuration, with no crossover.

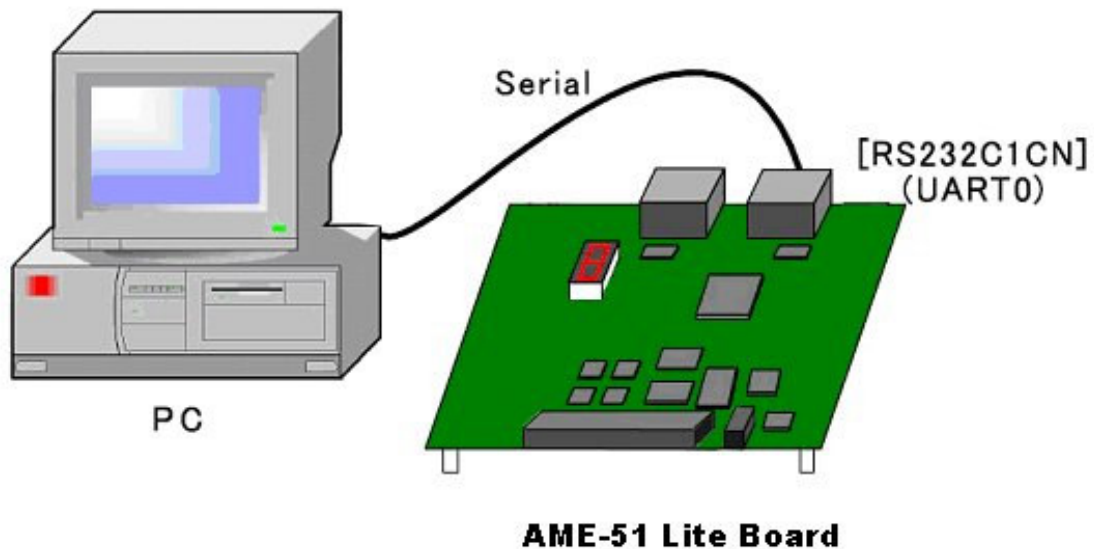


Figure 3.1 Serial connection of the CPU board

2. Set SW1 to Standalone Flash Mode.

SW1	Setting
JTAG	X
BOOTCLK	X
BOOT0	OFF
BOOT1	OFF
EXIROME	OFF
EXBUSE	ON
ROMSEL	ON
FWJ	OFF

3. Connect the power supply adapter to the CPU Board.

4. Apply the power and so it will display the pattern in Figure 3.2 on the 7-segment LED display.

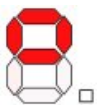


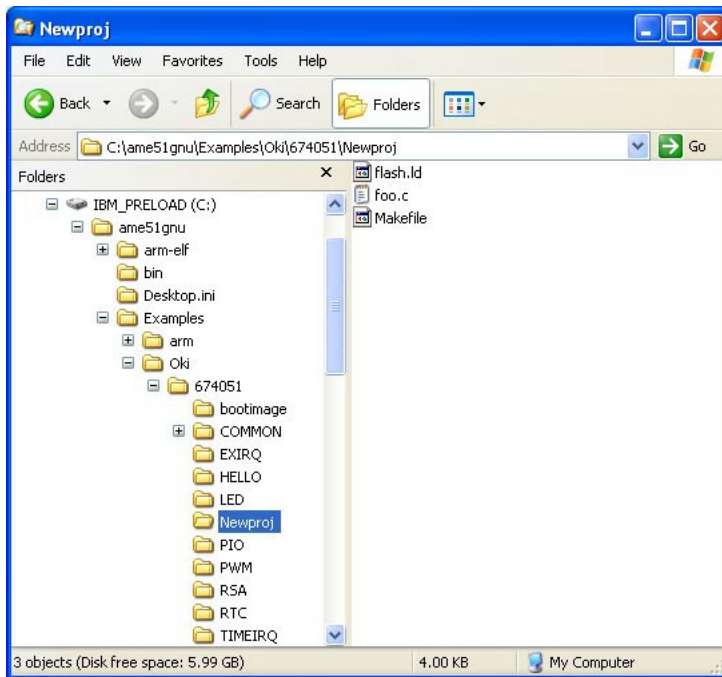
Figure 3.2 on the 7-segment LED display

4.GNUARM

To start up the CPU board, the most important step is to install the AME-51GNU package in your own PC. You should remind that the file must put in C:\. This chapter will discuss the file structure of GNUARM.

4.1 674051 Directory

There are at least 4 sub directories within 674051 directory. Figure 4.1 explain the meaning of each of this item



Name	Explanation
COMMON	Directory for common assembler and C sources
NewProj	Template project directory
Hello	“Hello world” sample program
bootimage	SRAM downloader binary

Figure 4.1 Explanation for 674051

4.2 COMMON directory

There are 2 more sub directory (1)INC and (2) SRC. INC for holding all the header file of source program. SRC for holding the common assembler and C source program.

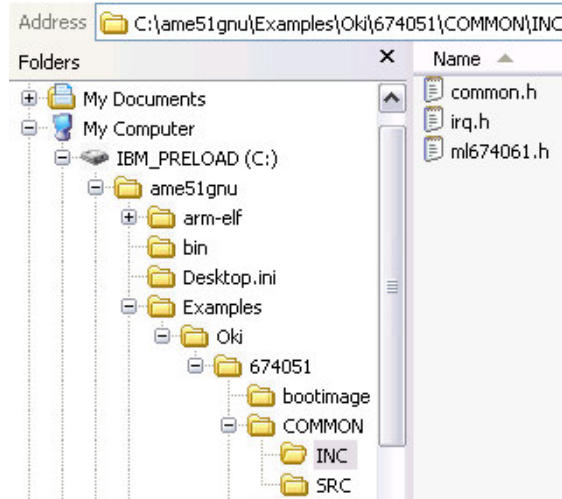


Figure 4.2a INC directory

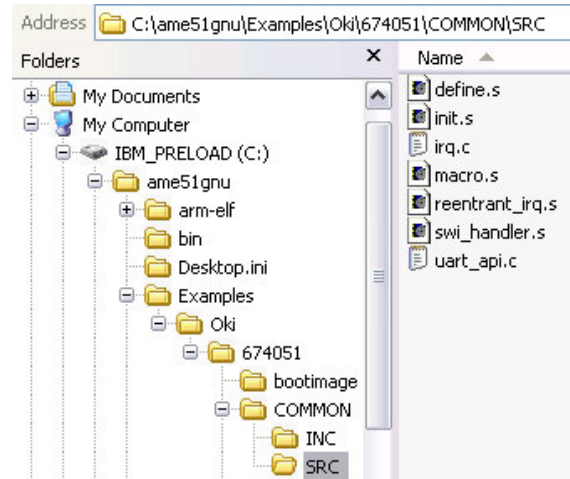


Figure 4.2b SRC

4.3 Project directory (e.g. Hello)

There are 3 files in this directory, including the main program of "hello.c", "flash.ld" and "Makefile".

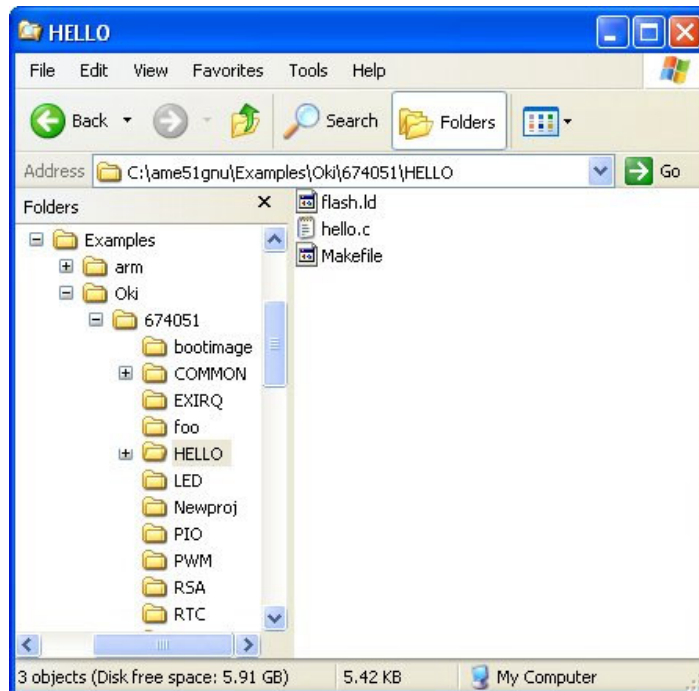


Figure 4.3 Hello directory

hello.c : main source program

Flash.ld : Linker script defines how the program places into the RAM and ROM of the LSI.

Makefile : It defines the compiler and linker setting. It automates the whole compiling process by just typing "**gnumake**"

5.Procedure to run user program

After you get the idea of file structure, you can develop your own program within AME-51GNU. This chapter will discuss the procedure to run your own program for the AME-51 Lite board. Here are the procedures to create a program FOO:

1. Copy your whole directory of "NewProj" and place it under ..\674051. Then rename it to "foo" as shown in Figure 5.1.

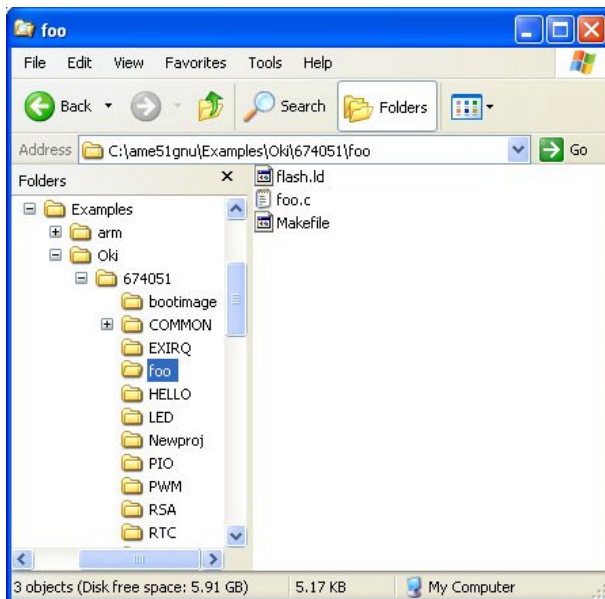


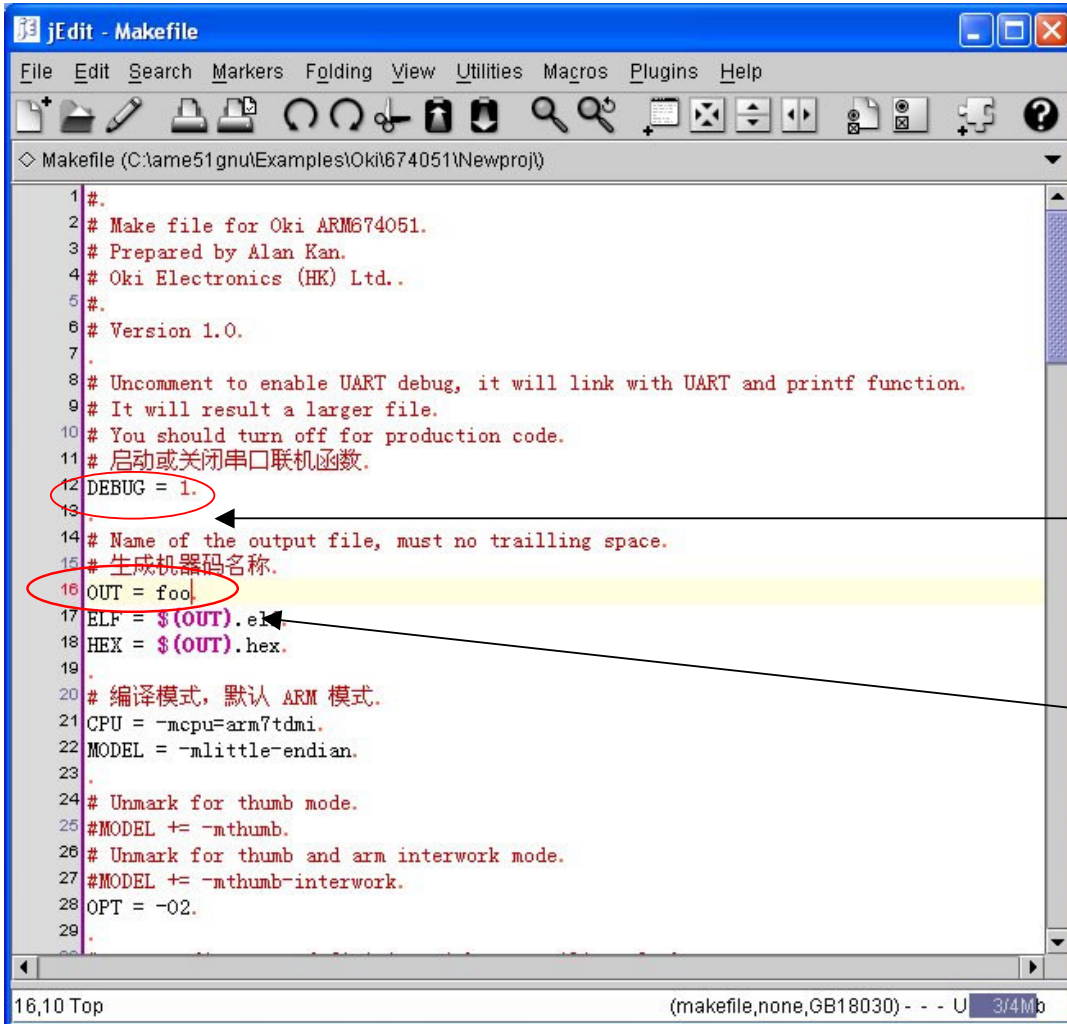
Figure5.1 Location of the USER program

2. Open the Makefile and change your output file name and set debug message as wish. As shown in Figure 5.2.

3. If any user assembler source is needed, put all of them under ..\COMMON\SRC with ".s" file extension. Make file will search all ".s" file in this directory and include them in assemble/link process automatically

4. Any user defined header file, should put under ..\COMMON\INC. Make file will include them in the compiling process automatically

5. All the user application C source code should place under the project directory. Make file will include all the ".c" file in compiling automatically



Allow the use of printf_uart when delete "#"

Change the name of your output file

Figure 5.2 Makefile

6. Click the GNUARM Icon on Desktop will open a command prompt as follow:

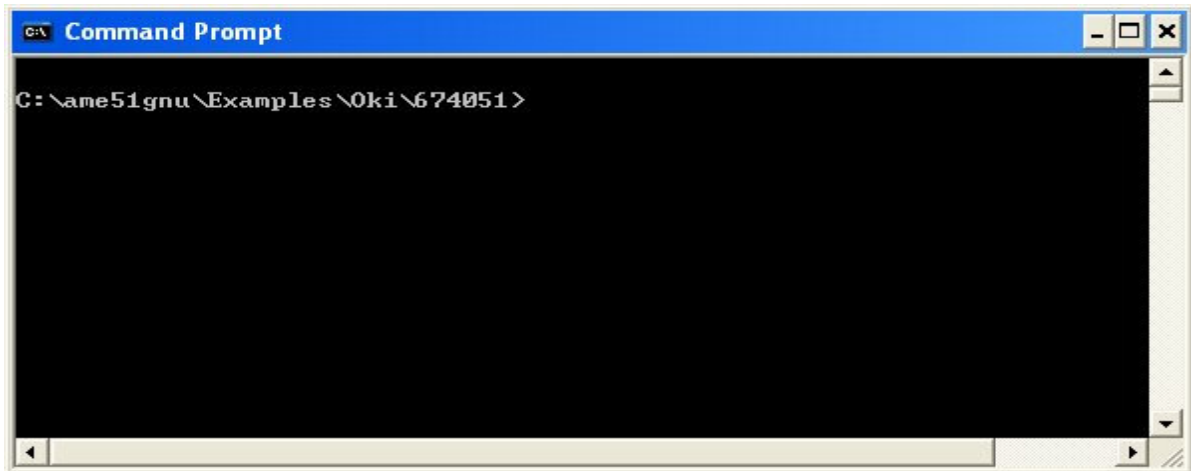


Figure 5.3 command prompt

7. Type "cd foo", change into foo directory

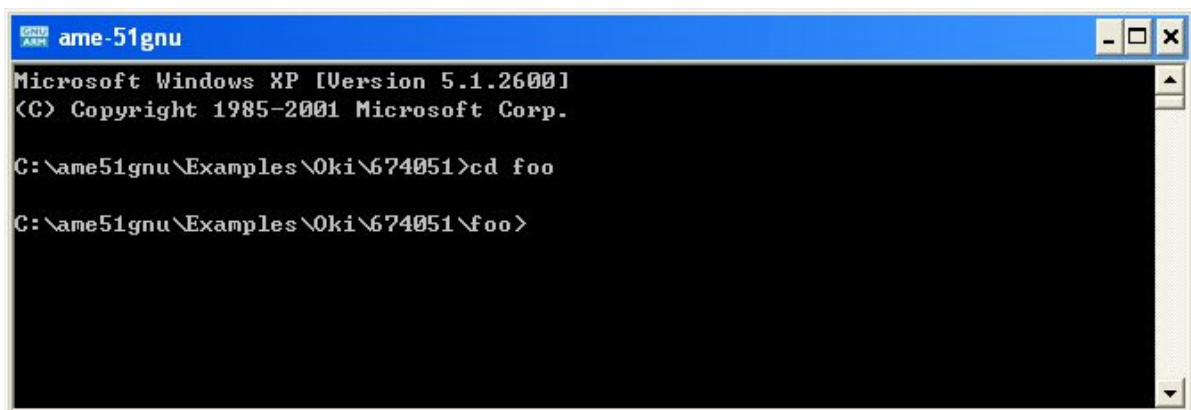


Figure 5.4

8. Type "**gnumake clean**" to clean up any intermediate files.

9. Type "**gnumake**" to generate HEX file. In general, "gnumake" will re-compile any modified source code. If you wish to re-compile all the source code, type "gnumake clean" before "gnumake".

10. After compiling the program, set the SW1 to Stand Alone Flash mode and press RESET (SW4)

SW1	Setting
JTAG	X
BOOTCLK	X
BOOT0	OFF
BOOT1	OFF
EXIROME	OFF
EXBUSE	ON
ROMSEL	ON
FWJ	OFF

11. Connect the CPU board to your PC by serial cable through RS232C1CN connector

12. Start and set up features of Tera Term as shown in Figure 5.5. (COM port setting depends on your PC setting).

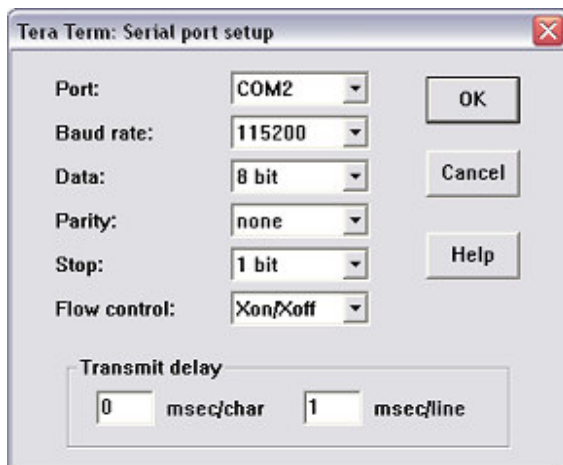


Figure 5.5 Setting for Tera Term

13. Press RESET on the board again, board should be connected and a window will be given out as shown in Figure 5.6. Press "Y" to start downloading Intel HEX file.

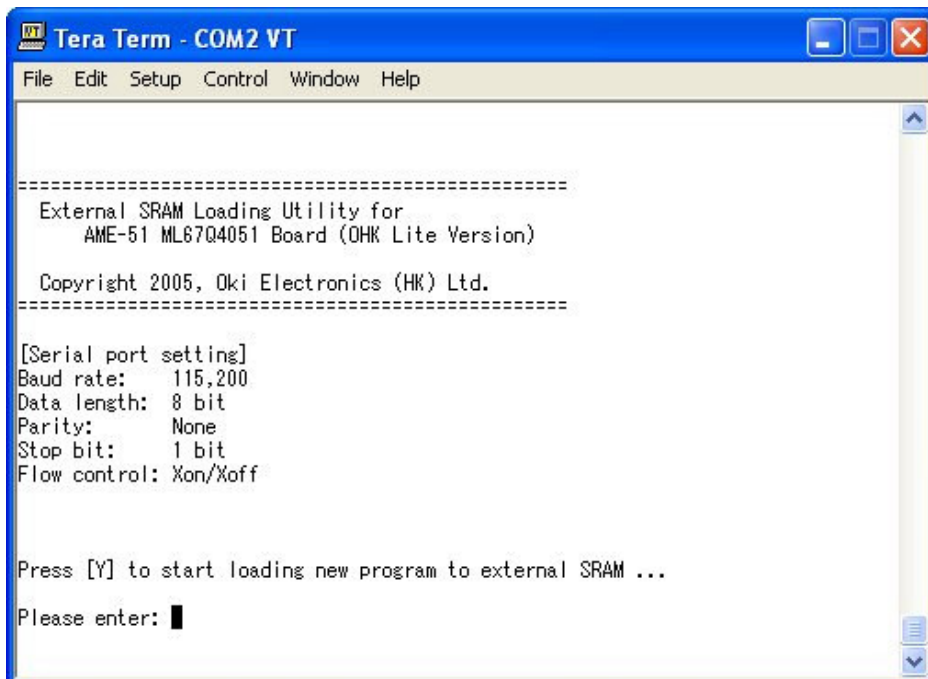


Figure 5.6

Now, you can download the program by choosing as Figure 5.7 , and then send the Intel HEX file that is created inside your program folder in C:\ame51gnu\Examples\Oki\674051\foo



Figure 5.7

14. After downloading the program, set BOOT1 of SW1 to ON (Standalone SRAM)

SW1	Setting
JTAG	X
BOOTCLK	X
BOOT0	OFF
BOOT1	ON
EXIROME	OFF
EXBUSE	ON
ROMSEL	ON
FWJ	OFF

15. Press RESET again, the program will start to run.

Appendix 1 Flash.ld

If user want to write a program to control the ML67Q4051 CPU board, beside Makefile ,flash.ld files may also be changed.

Flash.ld

```
MEMORY
{
    ROM      (rx) : ORIGIN = 0, LENGTH = 128K
    RAM      (!rx): ORIGIN = 0x10000000, LENGTH = 16K
}

SECTIONS
{
    /* All the readonly area place into ROM, it is normal place for program to put */
    .text : { init.o(.text)
            /* by putting at the beginning m
            *(.text)
            *(.rodata)
            *(.rodata*)
            *(.glue_7)
            *(.glue_7t)
            } >ROM

    /* All the variable or RW data will put into the RAM area */
    .data : { *(.data)
            } >RAM

    .bss : {
            *(.bss)
            *(COMMON)
            } >RAM
}
```

You may need to change the size of ROM

The initial address may also be change