



μTasker Document

- **Kirin3 M52259 demo user's guide**

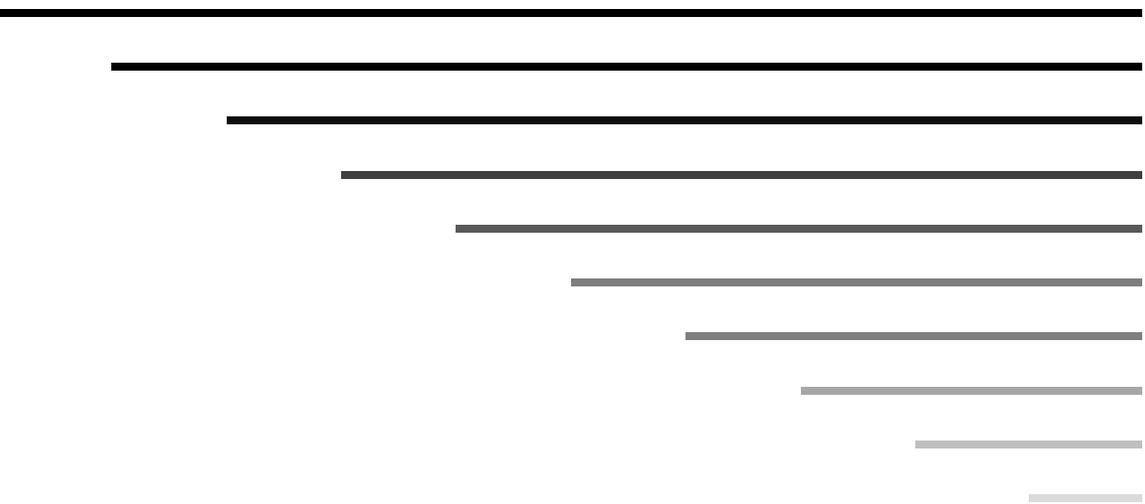


Table of Contents

1. Introduction.....	3
2. Preparing to start	3
3. Compiling the μTasker Demo using CodeWarrior	3
4. Compiling the Project in VisualStudio C++ and with GCC.....	4
5. Applying Power to the M52259DEMO Board	4
6. Applying Power to the M52259EVB	4
7. Loading the Software to the Board and Running.....	5
7.1. Running the M52259DEMO board without Codewarrior.....	5
7.2. Recommended Jumper Settings on the M52259EVB	6
8. USB CDC demo	7
9. Ethernet.....	8
10. Configuring the μTasker project to support firmware upload	9
11. Simulating the M52259DEMO with the SW demo pack executable.....	11
12. Conclusion	11

1. Introduction

The μTasker project for the Coldfire V2 MCU devices (M522XX) has been extended to include the Freescale Kirin3 M5225X family, featuring internal Ethernet MAC, CAN and USB-OTG. The Kirin3 was officially announced on 22nd January 2009 and the μTasker support prepared to be immediately available.

This guide offers step-by-step instructions to program and test the μTasker Ethernet and USB demos on the M52259DEMO and M52259EVB boards from Freescale.

2. Preparing to start

The M52259DEMO board contains an integrated background debugger (OSBDM), used for programming and testing, which is supported by CodeWarrior 7.1 with 7.1.1 patch. This software is delivered on the CD in the M52259DEMO board pack and should be installed before continuing (first the CW7.1, followed by the patch). This is a special edition version which allows you to develop software up to a maximum size of 128k. This is adequate for all μTasker demos and also for many complete application development projects using the μTasker. Visit also the Freescale web site to check whether there are newer versions of CodeWarrior available which will include the first patch and other improvements as standard. Furthermore the μTasker project is delivered with GCC bat and make files allowing also unrestricted code sizes using the GNU compiler (for example the Codesourcery g++ Lite edition).

The M52259EVB has a connector for an external Background Debug Module (BDM) as well as an integrated P&E Micro BDM module.

3. Compiling the μTasker Demo using CodeWarrior

The μTasker software can be obtained on request from www.uTasker.com/Licensing/request.html.

The project is free of charge for all non-commercial use and can be evaluated for a period of 30 days for commercial purposes. Full licensing details can be read at www.uTasker.com/Licensing/License.html

Precompiled target code is freely available of the software resource page: www.uTasker.com/software/software.html which can loaded to the board without the need to compile it.

Open the project

```
\Applications\uTaskerV1.3\CodeWarrior_M5223X\uTaskerV1.3\uTaskerV1.3_CW7.mcp
```

 in CodeWarrior. Select the target M5225X_ROM and ensure that the target board is configured for the M52259DEMO board in the μTasker V1.3 project file config.h as follows:

```
#define _M5225X must be active to select the M5225X family  
and
```

```
#define M52259DEMO must be active to select the DEMO board rather than the EVB.
```

To compile for the M5225EVB, use the setting M52259EVB instead

Compile the project by pressing F7 (Project | make) or by clicking on the “Make” icon. The project should compile with neither errors nor warning. The target code size is about 80k – this is the full demo size with Ethernet (dynamic web server, FTP, TELNET, etc.), UART menu controlled settings and USB CDC device functionality. The project supports firmware updates via Ethernet (FTP or HTTP POST) and USB, which will be discussed in more detail later on in the guide.

4. Compiling the Project in VisualStudio C++ and with GCC

The µTasker simulator operates with VisualStudio C++ from version 6.0 and upwards, including the free VisualStudio Express editions. This operation is detailed in the document <http://www.utasker.com/docs/M5223X/uTaskerV1.3-Tutorial-M5225X.PDF>

The VisualStudio environment includes also a very powerful editor and the capabilities to add post-build steps which are suitable for cross-compiling to the target hardware. As detailed in the tutorial in the link above, a target is included which will automatically perform this stage to generate a standalone binary image using the GNU compiler, plus images suitable for downloading to the target with boot loader support.

As long as the GNU compiler is installed on the PC, the project can thus comfortably be built from the VisualStudio environment with GCC. In this case the target configuration “Win32 uTasker M5223X plus GNU build” should be selected.

5. Applying Power to the M52259DEMO Board

The M52259DEMO board is powered via the USB debug cable or via the miniAB-USB cable.

Start off by connecting the board via USB debug connector to a free USB host socket on your PC using a standard USB cable. The debug connector is the larger USB socket on the demo board (not the miniAB-USB one, which is for the processor’s USB interface).

When the cable is connected for the first time, the USB driver for the OSBDM will be installed. *This driver and support for CodeWarrior was added with the CW 7.1.1 patch.*

6. Applying Power to the M52259EVB

Start off by connecting the board via USB debug connector to a free USB host socket on your PC using a standard USB cable. The debug connector is the larger USB socket on the EVB (not the miniAB-USB one, which is for the processor’s USB interface).

When the cable is connected for the first time, the USB driver for the P&E will be installed.

The M52259EVB is easiest powered via the delivered 5V wall unit. This is connected to J5 where a board switch can be used to apply the power to the board when required. The P&E BDM is self-powered via USB and so doesn’t need the 5V main power for the installation step.

7. Loading the Software to the Board and Running

With CodeWarrior running, select the FLASH programmer from the Tools menu. In the Target Configuration setting of the Flash programmer dialog, select “Load Settings...” and the settings for MCF52259_INTFLASH.xml.

In the target configuration setting ensure that the debugger setting suits the board used:

- When using the M52259DEMO board the “Connection” setting must be selected for Coldfire v2-v4 JM60 OSBDM
- When using the M52259EVB the “Connection” setting must be selected for PEMICRO_USB

First perform an erase operation to ensure that the internal FLASH memory is completely cleared.

Then perform a program operation to load the software to the board.

If you are loading a pre-built target, select the path to the file to be loaded before starting the operation. The file in the SW demo pack is called uTaskerV1.3_M52259DEMO_USB_ETHERNET.S19 or uTaskerV1.3_M52259EVB_USB_ETHERNET.S19. In order to be able to load it you may need to have any other project open in Code Warrior, to ensure that the Flash Programmer tool is available.

To start the loaded software, command “Run” (click on the green triangle icon).

The board is operating correctly if LED1 (on the M52259DEMO) or D10 (on the M52259EVB) is blinking at a rate of 2.5Hz.

7.1. Running the M52259DEMO board without Codewarrior

Once the software has been successfully loaded to the board it is also possible to power the board, and immediately start the target, by connecting a USB cable to the Coldfire’s USB connector (miniAB). This will avoid needing Codewarrior to start the software. This will however cause a PC host to enumerate with the demo software and request the driver – if a 5V power adapter with USB miniAB termination is available it will avoid this. It is generally best to complete the USB demo steps (see the following chapter), which explains the driver installation, before powering from a PC host via the miniAB USB connector.

7.2. Recommended Jumper Settings on the M52259EVB

The M52259EVB has various jumpers which allow it to be configured for flexible use. For use with the μTasker project and SW demo it is recommended that the jumpers are set to these positions:

J32 – no jumper	J30 – no jumper	J33 – U2RXD	J27 – U2RXD	J7 - HI
J22 – ON	J31 – USB_VBUS	J24 – U2CTS	J28 – U2TXD	J8 – no jumper
J23 – BDM	J24 – ON	J35 – U2TXD	H2 – on	J9 – FEC_MDC
J29 - PU	J21 - ON	H3 – no jumper	J12 – HI	J10 – FEC_MDIO
J11 – RCON	J13 – OSC	J14 - XTAL	H1 – no jumper	J16 – ON
J17 – P3V3	J6 – PSTCLK	J2 - ON	J18 – ON	

8. USB CDC demo

The USB CDC device demo is described in detail in the document:

http://www.utasker.com/docs/uTasker/uTaskerV1.3_USB_Demo.PDF

When asked for the USB driver, use the file `uTaskerFreescaleVirtualCOM.inf`. This is contained in the SW demo pack or in the project directory

`\Applications\uTaskerV1.3\USB`.

The demo document also explains loading new firmware via USB. For details about loading firmware via Web Browser and also configuring the project in CodeWarrior see the following sections of this document.

The guide explains also the µTasker simulator's capability to simulate the USB part of the project, where the following shows the simulator running with enumerated USB.

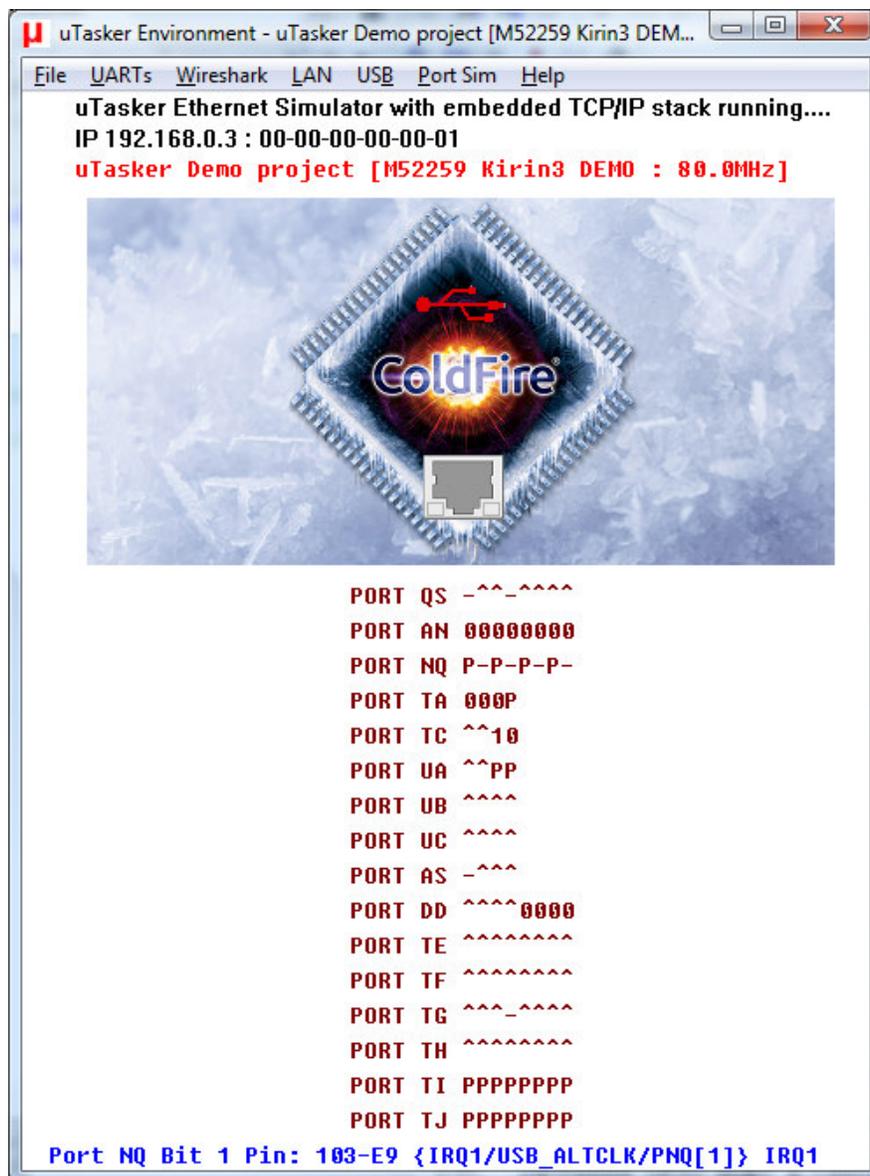


Figure 1 – uTasker simulator running M52259DEMO, showing enumerated USB state

9. Ethernet

The Ethernet part of the μTasker project is detailed in the M5225X Kirin3 tutorial:

<http://www.utasker.com/docs/M5223X/uTaskerV1.3-Tutorial-M5225X.PDF>

The tutorial contains a description of working with the μTasker simulator to fully test this functionality as well as compiling with CodeWarrior and GCC. The web pages are discussed in detail but not the method of configuring the project to work with firmware uploads. The following section contains details relevant to this task.

The web pages for the Ethernet demo are contained in the SW demo pack in the sub-directory WebPages. The default IP address is **192.168.0.3** with sub-net mask **255.255.255.0**. If this doesn't suit your test network it can be change in the command line interface via UART or USB:

- Enter the "Configure LAN Menu" by entering 1 followed by the ENTER key.
- With the command "show_config" the present configuration can be displayed.
- Use "set_ip_address 192.168.1.125" to change the IP address to 192.168.1.125 (example IP address – set to a value suitable for your network).
- With "save" this is committed to parameter FLASH, which will cause the board to reset and restart.
- Now test that you can ping the new address. If successful, enter into the "Configure LAN Menu" again and validate the new setting by commanding "validate". This will now commit this new address to be used permanently. Note that the new settings should be validated within a period of three minutes otherwise the temporarily saved ones will be deleted and original ones returned. See the μTasker tutorial for a full explanation of the reasons for this procedure.
- Note also that the board can be configured to obtain its IP configuration from a DHCP server "set_dhcp enable". As long as the sub-net settings match the network it is also possible to communicate with the board using its NetBIOS name "KIRIN3" (eg. "ping KIRIN3") without needing to know its IP address. The NetBIOS name can later be changed in the web server.
- To load the web pages to the target use the bat file `Copy_all.bat`, which is included in the SW demo pack or else can be found in the project directory `\Applications\uTaskerV1.3\WebPages\WebPagesM5225X`. If the IP address of the target has been changed, the IP address setting in the bat file will also need to be modified accordingly. The bat file will automatically transfer all web pages to the target via FTP, after which the board can be browsed using a normal web browser.

The Ethernet section of the demo contains the following main features:

- **TELNET** (command menu is accessible via TELNET and debug output is switched to it when connected)
- **Web Server** with demonstration of dynamic content generation, control of ports, email configuration and transmission via web command (SMTP). Display of user loaded images and firmware upload via HTTP POST.
- **FTP** server for loading web pages and firmware upload.
- **NetBIOS** - can be contacted by name **KIRIN3** (name configurable via web server)
- **DNS** (for SNMP provider resolution)
- Default IP address **192.168.0.3** (subnet mask **255.255.255.0**) – can be configured to other IP configuration via USB or UART menu, including **DHCP**.

10. Configuring the µTasker project to support firmware upload

The pre-compiled target files `uTaskerV1.3_M52259DEMO_USB_ETHERNET.S19` and `uTaskerV1.3_M52259EVB_USB_ETHERNET.S19` from the SW demo pack already contains the µTasker boot loader and the complete application. In this case firmware uploads can already be tested by using the following two files in the SW demo pack:

- `z_Upload_11.bin` (or `z_Upload_EVB_11.bin` for the EVB). This is the same application but with a version number **V1.3.011**. By checking the version number displayed in the web browser or via the command menus (UART, TELNET or USB) it is possible to verify that a firmware upload has been successful.
- `z_Upload_10.bin` (or `z_Upload_EVB_10.bin` for the EVB). This is the application contained in the original target file which can be reloaded after testing `z_Upload_11.bin` or (or `z_Upload_EVB_11.bin` for the EVB). This then displays the original version number **V1.3.010**.

In order to compile software to upload to the target a different CodeWarrior target is required. The corresponding target is `M5225X_BM_ROM` (Bare-minimum boot loader target). For full details about working with this and also the boot loader project see the boot loader documents

http://www.utasker.com/docs/uTasker/uTaskerBoot_003.PDF

and

http://www.utasker.com/docs/uTasker/BM-Booloader_for_M5223X.PDF

The target to select in the CodeWarrior boot loader project is `M5225X_ROM`.

To be noted are following settings for the boot loader project:

- in `config.h` the defines `#define KIRIN3` and `#define _M5225X` should be active. The first one sets the application start address to `0x1000` (after the first 4k FLASH block in the Kirin3 device) and the second configures the memory map accordingly. *Note that the define `KIRIN3` also automatically disables the assembler file workaround for the internal FLASH speculation address qualification incomplete errata of earlier M522XX devices.*

- The demo project uses the following memory map:
 - Boot loader occupies 4k (0..0xffff).
 - Application space is 128k (0x1000..0x16aff).
 - Parameter system occupies 2 x 4k sectors (0x21000..22fff)
 - The file system occupies 372k (0x23000..7ffff), whereby the last 128k (0x60000..0x7ffff) is intended for accepting secure firmware upload

When performing firmware uploads it is advisable to power the M52259DEMO board via the mini-AB USB connector rather than the debug USB connector. This avoids having to start the board using CodeWarrior and also avoids the upload/reset/programming process from disturbing the OSBDM driver. During the process the driver will otherwise lose contact with the Coldfire and report errors.

11. Simulating the M52259DEMO with the SW demo pack executable

In order to allow new users to quickly see how the µTasker simulator operates, an executable is included in the corresponding SW demo pack. This can be found in the sub-directory `\Simulator` and started by simply double-clicking on `uTaskerV1-3-M52259DEMO-Sim.exe`.

The simulator doesn't need any installation but does require WinPCap to be installed on the PC. If you already have WireShark installed (which also uses it) no further effort is required, otherwise this can be found at <http://www.wireshark.org/>

The simulator will be displayed as show in the screen shot earlier in this document. It has a default IP address of 192.168.0.1 and opens COM1 on the PC as UART0. This means that by connecting a cross-over cable to COM1 (or by using a virtual com port loop back to it) you can communicate with a terminal emulator (19'600 Baud) and change any settings if needed, as already described in this document and the other guides it links to. When the simulator program is terminated it will always save its complete simulated FLASH contents so that the new settings are valid the next time it is started.

You will need first to configure the PC's NIC (Network Interface Card) that it should use for mapping the M52259 Ethernet controller to – do this in the "LAN" menu with "Select Working NIC". This setting will also be saved and thus needs only be entered once.

Then use the bat file, as described in the chapter "Ethernet", to transfer the web pages to the simulated device via FTP. After this has been done, connect to its IP address using a standard browser and see the project in operation!

The simulator allows comfortable project development and testing in the VisualStudio environment and the complete project is included as part of the µTasker package.

12. Conclusion

This document has given a quick introduction to compiling and loading the demo software to, and running it on, the **M52259DEMO** and **M52259EVB** boards. As well as giving simple instructions as to how to verify that the software is correctly running is has given links to further documents containing detailed descriptions concerning the use of the USB and the Ethernet parts of the demo, as well as how the project can be configured to enable support of the firmware upload features via USB and Ethernet.

A short guide to using the simulator executable, as included in the SW demo pack, has also been given.

Any specific details concerning the **M52259DEMO** and **M52259EVB** boards, the **M52239** and its project target configurations which are not specified in the detailed documents has been included in this document.

Modifications:

- V0.0 21.01.2009 First version.
- V0.1 14.02.2009 Add M52259EVB details and GCC build reference