



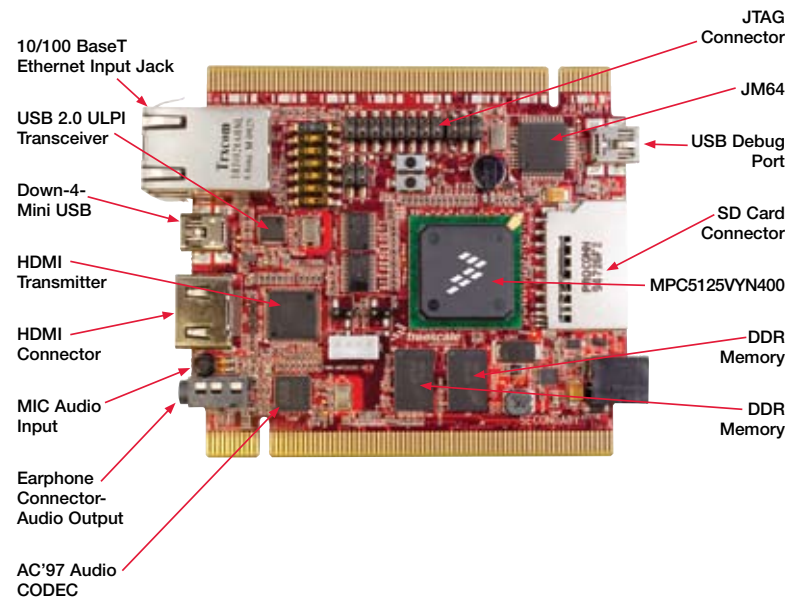
MQX RTOS
Lab tutorial



Table of Contents

- Lab 1: Simple digital sign, Freescale MQX RTOS
- Lab 2: Adding an FTP server, Freescale MQX RTOS with FTP server
- Lab 3: Adding a Web server, Freescale MQX RTOS with Web server
- Lab 4: Anatomy of the project

Get to know the MQX RTOS



Learn More: For more information about Freescale products, please visit www.freescale.com.

MQX RTOS Lab Tutorial—Digital Signage with MPC5125

Introduction

The following lab tutorial applies to the MPC5125 Tower Module, TWR-MPC5125. The following instructions have been prepared for a Windows® XP host computer.

Step by Step Instructions



Simple digital sign, Freescale MQX RTOS

This lab will guide you through the use of the Freescale MQX operating system under the CodeWarrior environment. A simple digital sign example application is presented which shows how information, advertising or other messages can be displayed on screens in retail stores, corporate buildings and public places.

Demonstrates

- MQX project in CodeWarrior IDE
- Project build, download and run in CodeWarrior IDE
- MQX frame buffer display driver for MPC5125

Step by Step Instructions

1. Install software
 - a. Install CodeWarrior for mobileGT v9.2
 - b. Install CodeWarrior patch for MPC5125
 - c. Install Freescale MQX RTOS v3.5.1
 - d. Install Freescale MQX patch for MPC5125
2. Connect the HDMI cable: Using the HDMI-to-DVI-D cable provided, connect the HDMI port on the TWR-MPC5125 to a DVI-D port on a display monitor. (Alternatively, connect the HDMI port to an HDMI port on a monitor. Cable not provided.)
Note: DVI-to-VGA is not supported.
3. Connect an Ethernet cable: Using a crossover Ethernet cable, make a direct connection between the host computer and the Ethernet RJ-45 connector on the board.
Note: Computers with a network interface that supports Auto-MDIX (automatic media-dependant interface crossover) can use either a straight-through or crossover cable. This is very common.
4. Connect the 5V power cable. (This step is optional because the Serial-to-USB connection in the next step provides power to the board.)

5. Connect the USB cable for the Serial-to-USB bridge. The Freescale MCS08JM60 Serial-to-USB bridge solution provides an RS-232 equivalent connection to the host computer through the USB communications device class. When plugged in and powered the USB connection will enumerate as a COM port on the PC.
To determine the COM port number for the connection, right click on the My Computer icon and select Manage. Click Device Manager. Find and expand ports (COM and LPT).



Note: On Linux host computers the USB CDC comes up in /dev/ACMO.

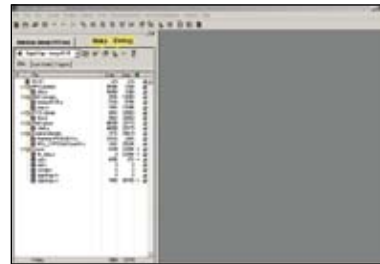
6. Open Microsoft® HyperTerminal or a terminal program of your choice.
Select File > New Connection. Type a name for the new connection. Pick the COM port number for the connection found in the last step. Choose the following port settings: Bits per Second: 115200, Data bits: 8, Parity: None, Stop bit: 1, Flow control: None.



Note: When the power is cycled on/off, the Serial-to-USB connection will lose contact with HyperTerminal. Therefore it is recommended that HyperTerminal be shut down and restarted each time the board is unplugged.

7. Open CodeWarrior for mobileGT v9.2.
8. Open the example project:
Select File > Open and browse for: C:\Program Files\Freescale\Freescale MQX 3.5\demo\digital_sign\cwmpc92\sign_twrmpc5125.mcp

Note: If you did not install MQX in the default C:\Program Files\Freescale MQX 3.5\ directory during installation, you must first recompile the MQX libraries to reflect the new path name. See the release notes for more information before continuing on with the lab.

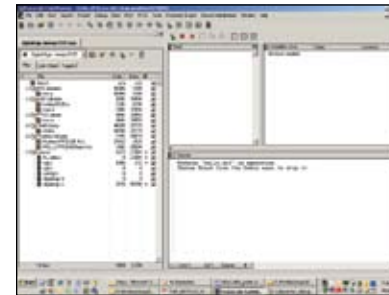
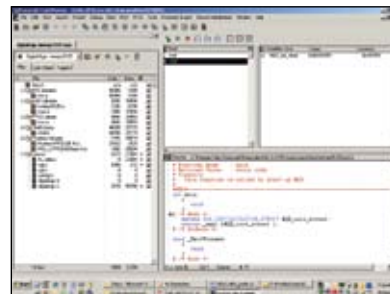


9. Compile the project by clicking the Make icon on the project pane toolbar.
There are two methods to download and execute the example application.

Method 1: Running directly out of SDRAM using the hardware debug interface

Note: Requires a hardware debug interface such as the CodeWarrior USB TAP, which can be purchased separately. This method takes full advantage of CodeWarrior debugging capabilities. The example project is preconfigured for the CodeWarrior USB TAP. To change to another interface, open the project settings window, select Remote Debugging, and choose the correct connection.

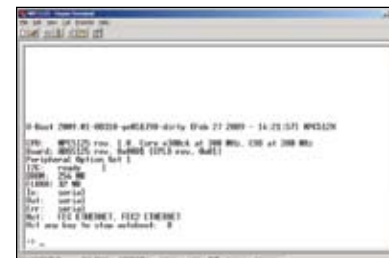
10. Connect a hardware debug interface such as the CodeWarrior USB TAP to the JTAG header J2.
11. Start the debugger and execute the application by clicking the Debug icon.
CodeWarrior initializes the board and loads the code into the SDRAM through the JTAG/COP debug interface.
12. Once the debug session starts, you will see the MQX entry-point function in the code window as seen in the figure below. Click the Debug icon again to run the code, which will look like the next figure.
Continue to the **Loading your own images** section.



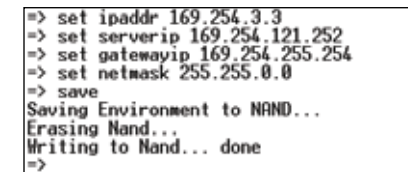
Method 2: Using u-boot bootloader to load and execute in SDRAM

Note: Requires u-boot flashed to the on-board nand flash and a network connection between the board and the host computer.
The board comes pre-programmed from the factory with u-boot bootloader. On power up, the processor boots from the nand flash and u-boot takes control. u-boot initializes the board and enables SDRAM. For information on u-boot see the board user's guide.

13. Ensure switches are set correctly to allow for nand booting. See the board user's guide.
14. Open and configure a TFTP application on the host computer.
On the host computer, open a TFTP Server of your choice. If you do not have a TFTP Server, you can obtain one for free from Net Burner at www.netburner.com.
15. Set the TFTP directory to C:\tftp.
16. Copy C:\Program Files\Freescale\Freescale MQX 3.5\demo\digital_sign\cwmpc92\twrmpc5125\extram_d.bin to C:\tftp
Note: If extram_d.bin did not get generated in the make process, go to the target settings panel, click EPPC linker, and change the Binary File: drop down item to One.
17. Press Reset on the board.
18. Find the u-boot prompt on the terminal.
Quickly press a key (on the host computer keyboard) to stop automatic booting the pre-programmed operating system.



19. Set the network connection settings:
At the u-boot prompt type the following commands:
"set ipaddr 169.254.3.3"
"set serverip 169.254.xx" represents the IP address of your host computer
"set gatewayip 169.254.255.254"
"set netmask 255.255.0.0"
"save"

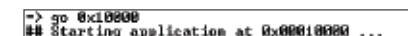


Note: Typically, when you connect your computer directly to the board, the host computer will default to an Auto-IP address on the subnet 169.254.x.x. By setting the board IP address to 169.254.3.3, no additional setup is required on the host computer. Be aware that the host computer will take a few minutes to default to the auto IP address and make the connection. However, if you have trouble connecting, you may configure the IP address of the computer manually. Select Start > Settings > Network Connections > Local Area Connection. Note your original TCP/IP settings, and then set your IP address to 169.254.3.4 and your subnet mask to 255.255.0.0.

20. Transfer the binary file to the on-board memory.
Type "tftp 10000 extram_d.bin."
With the TFTP server running on the host computer, u-boot will transfer the MQX application binary file to the on-board SDRAM.



21. To run the application, type "go 0x10000."



The MQX application will execute.
On the monitor, you will see the following images.



Loading your own images

- Note: This simple demonstration software hasn't been optimized to use a compressed picture format, therefore compile and load time can take several minutes. Recommended pictures sizes are 800 x 600 or less.
The project is set up to allocate about 4 MB for pictures. To add more space, increase the ROM size in the linker file and shift the other sections accordingly.
Pictures must be in 24-bit .bmp format.
22. Add your own 24-bit color bitmap pictures to C:\Program Files\Freescale\Freescale MQX 3.5\demo\digital_sign\web_pages\
 23. Double click mkfbs.bat in the directory C:\Program Files\Freescale\Freescale MQX 3.5\demo\digital_sign\
 24. A new ffs_data.c file will be created. This file contains MQX trivial file system data that is pre-loaded with the application.

continued on the reverse side

MQX RTOS—Lab Tutorials 2–4 (sheet 3 of 3)

continued from previous page

25. Open CodeWarrior and the example project if it is not already open.
Select File > Open and browse for:
C:\Program Files\Freescale\Freescale MQX 3.5\demo\digital_sign\cwmpc92\sign_twrmpc5125.mcp
26. Refresh the project by selecting Remove Object Code.
Select Project > Remove Object Code
27. Click the Make icon.

Note: This simple demonstration software hasn't been optimized to use a compressed picture format, therefore compile and load time can take several minutes.

28. Load and execute the application using method 1 or method 2 from above.

To change the default IP address of the board

Open the file: config.h.

Change the line:

```
#define ENET_IPADDR IPADDR(169,254,3,3)
```

Changing the speed of the images

With the Digital Sign project open in CodeWarrior, double click on the file digitalsign.c to open it.

Scroll down to the display_task function.

Find the time delay function call, _time_delay(6000). Change the time value from 6000 to whatever you like. The time value argument is in milliseconds. For a 1 second interval, type 1000.

```
> @digital_sign\src\...
1 int main(int argc, char** argv) {
2     _time_delay(1000);
3 }

```

Programming the project to the nand flash

To flash your project to the on-board nand flash get back to the u-boot prompt.

```
fttp 0x10000 ram_d.bin
-transfer the binary file to the sdram
nand_e 0x80000 0x81400
-erase the nand flash blocks 0x80000 to 0x81400
nand_w 0x10000 0x80000 0x1400000
-write from sdram to the nand flash
Then make a command to boot mqx.
set mqxboot "nand_r 0x10000 0x80000
0x1400000; go 0x10000"
```

```
Save
=====
drom:
Bytes transferred = 505968 (bf5738 hex)
-> nand_e 0x1000 0x0000
-> nand_w 0x10000 0x1400 0x000000
-> set mqxboot "nand_r 0x10000 0x1400 0x000000; go 0x10000"
-> save
Saving Environment to NAND...
Erasing NAND...
Writing to NAND... done
```

To make the MQX application the auto-boot option, type "set bootcmd run MQXboot" and press Enter. Then, type "save" to save this change to the u-boot environment variables in the nand flash.

```
=> set bootcmd run mqxboot
=> save
Saving Environment to NAND...
Erasing NAND...
Writing to NAND... done
=>
```

Lab 2

Adding an FTP server, Freescale MQX RTCS with FTP server


1. Ensure the Ethernet cable is connected between the host computer and the board.
 2. Open an FTP client on your computer such as available on the DOS command line. Click the Start button, select Run and type cmd.
- At the command prompt, type "ftp 169.254.3.3" and press Enter.
3. No user name/password required. Press Enter.
 4. Take a 24-bit bitmap (bmp) file and put it in your FTP directory (current directory).
 5. Type "put file.bmp."



6. The file will transfer to the RAMDISK drive that that is mounted on the SDRAM.
7. The new image file will now display along with the preloaded images.
8. You can add up to 16 images, but do not exceed 5 MB memory. If you need more space you can adjust the RAM_DISK_SIZE in digitalsign.c.

Lab 3

Adding a Web server, Freescale MQX RTCS with Web server

1. Ensure the Ethernet cable is connected between the host computer and the board.
 2. Using the same method as Lab 2, transfer images to the board using FTP.
 3. On the host computer, open a Web browser such as Microsoft Internet Explorer.
 4. In the address toolbar, type: http://169.254.3.3.
- 
5. View the Web pages.
Click on the links to the left to view the network stats, system run time and the status of the display.



Creating your own Web page

1. Modify the html files in the directory C:\Program Files\Freescale\Freescale MQX 3.5\demo\digital_sign\web_pages\
2. Double click mkfts.bat in the directory C:\Program Files\Freescale\Freescale MQX 3.5\demo\digital_sign\
3. A new tfs_data.c file will be created. This file contains MQX trivial file system data that is pre-loaded with the application.
4. For Web pages making CGI requests, update cgi.c with functions to provide specific data.

Lab 4

Anatomy of the project

digitalsign.c—contains two auto-start tasks and a function

main task

initializes and starts the FTP server, Web server and ramdisk drive if the features are enabled

display task

initializes the display
locates and prepares bitmap images for display
displays the images

ramdisk function

installs the mfs file system and mounts an area of the SDRAM to a file system drive

digitalsign.h—configuration options for the digital sign example application

cgi.c—contain functions for sending data to the Web browser

cgi.h—header file for cgi.c

tfs_data.c—trivial file system data for pre-loaded images and Web pages. This file is automatically generated by the mkfts.bat script. Note: this file can get very large based on the size of the images.

twrmpc5125.a—TWR-MPC5125 Board Support Package (BSP) library

MQX.a—MQX library (real-time operating system)

rtcs.a—rtcs library (TCP/IP protocol)

mfs.a—mfs library (file system)